NIST Big Data Framework

1 Introduction
   1.1 Background
   1.2 Scope and Objectives of the Security and Privacy Subgroup
   1.3 Report Production
   1.4 Report Structure
   1.5 Future Work on this Volume

2 Big Data Security and Privacy
   2.1 Overview
   2.2 Effects of Big Data Characteristics on Security and Privacy
      2.2.1 Variety
      2.2.2 Volume
      2.2.3 Velocity
      2.2.4 Veracity
      2.2.5 Volatility
   2.3 Relation to Cloud

3 Example Use Cases for Security and Privacy
3.1 Retail/Marketing
   3.1.1 Consumer Digital Media Usage
   3.1.2 Nielsen Homescan: Project Apollo
   3.1.3 Web Traffic Analytics

3.2 Healthcare
   3.2.1 Health Information Exchange
   3.2.2 Genetic Privacy
   3.2.3 Pharma Clinical Trial Data Sharing[18]

3.3 Cybersecurity
   3.3.1 Network Protection

3.4 Government
   3.4.1 Military: Unmanned Vehicle Sensor Data
   3.4.2 Education: Common Core Student Performance Reporting

3.5 Industrial: Aviation
   3.5.1 Sensor Data Storage and Analytics

3.6 Transportation
   3.6.1 Cargo Shipping
      Figure 1: Cargo Shipping Scenario

4 Taxonomy of Security and Privacy Topics

4.1 Conceptual Taxonomy of Security and Privacy Topics
   Figure 2: Security and Privacy Conceptual Taxonomy
   4.1.1 Data Confidentiality
   4.1.2 Provenance
   4.1.3 System Health
   4.1.4 Public Policy, Social and Cross-Organizational Topics

4.2 Operational Taxonomy of Security and Privacy Topics
   Figure 3: Security and Privacy Operational Taxonomy
   4.2.1 Device and Application Registration
   4.2.2 Identity and Access Management
   4.2.3 Data Governance
   4.2.4 Infrastructure Management
   4.2.5 Risk and Accountability

4.3 Roles Related to Security and Privacy Topics
   4.3.1 Infrastructure Management
   4.3.2 Governance, Risk Management, and Compliance
   4.3.3 Information Worker

4.4 Relation of Roles to the Security and Privacy Conceptual taxonomy
   4.4.1 Data Confidentiality
4.4.2 Provenance
4.4.3 System Health management
4.4.4 Public Policy, Social, and Cross-Organizational Topics

4.5 Additional Taxonomy Topics
4.5.1 Provisioning, Metering, and Billing
4.5.2 Data Syndication

5 Security and Privacy Fabric

Figure 4: NIST Big Data Reference Architecture
5.1 Security and Privacy Fabric in the NBDRA

Figure 5: Notional Security and Privacy Fabric Overlay to the NBDRA
5.2 Privacy Engineering Principles
5.3 Relation of the Big Data Security Operational Taxonomy to the NBDRA

Table 1: Draft Security Operational Taxonomy Mapping to the NBDRA Components

6 Mapping Use Cases to NBDRA
6.1 Consumer Digital Media Use

Table 2: Mapping Consumer Digital Media Usage to the Reference Architecture
6.2 Nielsen Homescan: Project Apollo

Table 3: Mapping Nielsen Homescan to the Reference Architecture
6.3 Web Traffic Analytics

Table 4: Mapping Web Traffic Analytics to the Reference Architecture
6.4 Health Information Exchange

Table 5: Mapping HIE to the Reference Architecture
6.5 Genetic Privacy
6.6 Pharmaceutical Clinical Trial Data Sharing

Table 6: Mapping Pharmaceutical Clinical Trial Data Sharing to the Reference Architecture
6.7 Network Protection

Table 7: Mapping Network Protection to the Reference Architecture
6.8 Military: Unmanned Vehicle Sensor Data

Table 8: Mapping Military Unmanned Vehicle Sensor Data to the Reference Architecture
6.9 Education: Common Core Student Performance Reporting

Table 9: Mapping Common Core K–12 Student Reporting to the Reference Architecture
6.10 Sensor Data Storage and Analytics

6.11 Cargo Shipping

Table 10: Mapping Cargo Shipping to the Reference Architecture

Appendix A: Candidate Security and Privacy Topics for Big Data Adaptation
Appendix B: Internal Security Considerations within Cloud Ecosystems

Figure B-1: Composite Cloud Ecosystem Security Architecture[57]
Appendix C: Big Data Actors and Roles: Adaptation to Big Data Scenarios
Appendix D: Acronyms
Appendix E: References

Document References

[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
[9]
[10]
[11]
[12]
[13]
[14]
[15]
[16]
[17]
[18]
[19]
[20]
[21]
[22]
[23]
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
[33]
[34]
[35]
1. Introduction
   1.1 Background
1.2 Scope and Objectives of the Reference Architecture Subgroup

1.3 Report Production

1.4 Report Structure

1.5 Future Work on this Volume

2 Big Data Architecture Proposals Received

2.1 Bob Marcus
   2.1.1 General Architecture Description
   2.1.2 Architecture Model

       Figure 1: Components of the High Level Reference Model

   2.1.3 Key Components

       Figure 2: Description of the Components of the Low-Level Reference Model

2.2 Microsoft
   2.2.1 General Architecture Description
   2.2.2 Architecture Model

       Figure 3: Big Data Ecosystem Reference Architecture

   2.2.3 Key Components

2.3 University of Amsterdam
   2.3.1 General Architecture Description
   2.3.2 Architecture Model

       Figure 4: Big Data Architecture Framework

   2.3.3 Key Components

2.4 IBM
   2.4.1 General Architecture Description

       Figure 5: IBM Big Data Platform

   2.4.2 Architecture Model

   2.4.3 Key Components

2.5 Oracle
   2.5.1 General Architecture Description
   2.5.2 Architecture Model

       Figure 6: High level, Conceptual View of the Information Management Ecosystem

   2.5.3 Key Components

       Figure 7: Oracle Big Data Reference Architecture

2.6 Pivotal
   2.6.1 General Architecture Description
   2.6.2 Architecture Model

       Figure 8: Pivotal Architecture Model

   2.6.3 Key Components

       Figure 9: Pivotal Data Fabric and Analytics
2.7 SAP
   2.7.1 General Architecture Description
   2.7.2 Architecture Model
      Figure 10: SAP Big Data Reference Architecture
   2.7.3 Key Components
2.8 9Sight
   2.8.1 General Architecture Description
      Figure 11: 9Sight General Architecture
   2.8.2 Architecture Model
   2.8.3 Key Components
      Figure 12: 9Sight Architecture Model
2.9 LexisNexis
   2.9.1 General Architecture Description
      Figure 13: Lexis Nexis General Architecture
   2.9.2 Architecture Model
   2.9.3 Key Components
      Figure 14: Lexis Nexis High Performance Computing Cluster
3 Survey of Big Data Architectures
   3.1 Bob Marcus
      Table 1: Databases and Interfaces in the Layered Architecture from Bob Marcus
      Figure 15: Big Data Layered Architecture
   3.2 Microsoft
      Table 2: Microsoft Data Transformation Steps
   3.3 University of Amsterdam
   3.4 IBM
      Figure 16: Data Discovery and Exploration
   3.5 Oracle
   3.6 Pivotal
   3.7 SAP
   3.8 9Sight
   3.9 LexisNexis
   3.10 Comparative view of surveyed architectures
      Figure 17(a): Stacked View of Surveyed Architecture
      Figure 17(b): Stacked View of Surveyed Architecture (continued)
      Figure 17(c): Stacked View of Surveyed Architecture (continued)
4 Conclusions
   Figure 18: Big Data Reference Architecture
Appendix A: Acronyms
Appendix B: References

Document References

[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
[9]

Reference Architecture

Cover Page
Inside Cover Page
National Institute of Standards and Technology Special Publication 1500-6
Reports on Computer Systems Technology
Abstract
Acknowledgements
Notice to Readers
Table of Contents
Executive Summary

1 Introduction
  1.1 Background
  1.2 Scope and Objectives of the Reference Architectures Subgroup
  1.3 Report Production
  1.4 Report Structure
  1.5 Future Work on this Volume

2 High Level Reference Architecture Requirements
  2.1 Use Cases and Requirements
    Table 1: Mapping Use Case Characterization Categories to Reference Architecture Components and Fabrics
  2.2 Reference Architecture Survey
  2.3 Taxonomy
    Figure 1: NBDRA Taxonomy

3 NBDRA Conceptual Model
  Figure 2: NIST Big Data Reference Architecture (NBDRA)

4 Functional Components of the NBDRA
  4.1 System Orchestrator
4.2 Data Provider

4.3 Big Data Application Provider

4.3.1 Collection
4.3.2 Preparation
4.3.3 Analytics
4.3.4 Visualization
4.3.5 Access

4.4 Big Data Framework Provider

4.4.1 Infrastructure Frameworks
4.4.1.1 Networking
4.4.1.1.1 Software Defined Networks
4.4.1.1.2 Network Function Virtualization
4.4.1.2 Computing
4.4.1.3 Storage
4.4.1.4 Environmental Resources

4.4.2 Data Platform Frameworks

Figure 3: Data Organization Approaches
1.4.2.1 In-memory
1.4.2.2 File Systems
1.4.2.2.1 File System Organization
1.4.2.2.2 In File Data Organization
1.4.2.3 Indexed Storage Organization

Figure 4: Data Storage Technologies

4.4.3 Processing Frameworks

Figure 5: Information Flow
4.4.3.1 Batch Frameworks
Table 2: 13 Dwarfs—Algorithms for Simulation in the Physical Sciences
4.4.3.1.1 Map/Reduce
4.4.3.1.2 Bulk Synchronous Parallel
4.4.3.2 Streaming Frameworks
4.4.3.2.1 Event Ordering and Processing Guarantees
4.4.3.2.2 State Management
4.4.3.2.3 Partitioning and Parallelism

4.4.4 Messaging/Communications Frameworks

4.4.5 Resource Management Framework

4.5 Data Consumer

5 Management Fabric of the NBDRA

5.1 System Management
5.2 Big Data Lifecycle Management

6 Security and Privacy Fabric of the NBDRA

7 Conclusion

Appendix A: Deployment Considerations

Introduction

Figure A-1: Big Data Framework Deployment Options

Cloud Service Providers
Cloud Service Component
Resource Abstraction and Control Component
Security and Privacy and Management Functions

Physical Resource Deployments

Appendix B: Terms and Definitions

Appendix C: Examples of Big Data Organization Approaches

Relational Storage Models
Key-Value Storage Models
Columnar Storage Models

Figure B-1: Differences Between Row Oriented and Column Oriented Stores
Figure B-2: Column Family Segmentation of the Columnar Stores Model

Document
Graph

Figure B-3: Object Nodes and Relationships of Graph Databases

Appendix D: Acronyms

Appendix E: Resources and References

Document References

[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]

Standards Roadmap

Cover Page
Inside Cover Page
National Institute of Standards and Technology (NIST) Special Publication 1500-7
Reports on Computer Systems Technology
Abstract
Acknowledgements
Notice to Readers
Table of Contents
Executive Summary
1 Introduction
   1.1 Background
   1.2 NIST Big Data Public Working Group
   1.3 Scope and Objectives of the Technology Roadmap Subgroup
   1.4 Report Production
   1.5 Future Work on this Volume
2 Big Data Definition
   2.1 Big Data Definitions
   2.2 Data Science Definitions
3 Investigating the Big Data Ecosystem
   3.1 Use Cases
   3.2 Reference Architecture Survey
   3.3 Taxonomy
      Figure 1: NIST Big Data Reference Architecture Taxonomy
4 Big Data Reference Architecture
   4.1 Overview
      Table 1: Mapping of Use Case Categories to the NBDRA Components
   4.2 NBDRA Conceptual Model
      Figure 2: NBDRA Conceptual Model
5 Big Data Security and Privacy
6 Big Data Standards
   6.1 Existing Standards
      Table 2: Existing Big Data Standards
   6.2 Gap in Standards
   6.3 Pathway to Address Standards Gaps
Acronyms A: Acronyms
Appendix B: References
Document References
   [1]
   [2]
   [3]
DRAFT NIST Big Data Interoperability Framework:

Volume 4, Security and Privacy

Draft Version 1

NIST Big Data Public Working Group (NBD-PWG)

Security and Privacy Subgroup

National Institute of Standards and Technology

Gaithersburg, MD 20899

April 2015

U. S. Department of Commerce

Penny Pritzker, Secretary

National Institute of Standards and Technology

Dr. Willie E. May, Under Secretary of Commerce for Standards and Technology and Director

National Institute of Standards and Technology Special Publication 1500-4

71 pages (April 6, 2015)

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by Federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, Federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. All NIST Information Technology Laboratory publications, other than the ones noted above, are available at http://www.nist.gov/publication-portal.cfm.

Public comment period: April 6, 2015 through May 21, 2015

Comments on this publication may be submitted to Wo Chang

https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework
Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at NIST promotes the U.S. economy and public welfare by providing technical leadership for the Nation’s measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL’s responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in Federal information systems. This document reports on ITL’s research, guidance, and outreach efforts in Information Technology and its collaborative activities with industry, government, and academic organizations.

Abstract

Big Data is a term used to describe the deluge of data in our networked, digitized, sensor-laden, information-driven world. While great opportunities exist with Big Data, it can overwhelm traditional technical approaches and its growth is outpacing scientific and technological advances in data analytics. To advance progress in Big Data, the NIST Big Data Public Working Group (NBD-PWG) is working to develop consensus on important, fundamental questions related to Big Data. The results are reported in the *NIST Big Data Interoperability Framework* series of volumes. This volume, Volume 4, contains an exploration of security and privacy topics with respect to Big Data. This volume considers new aspects of security and privacy with respect to Big Data, reviews security and privacy use cases, proposes security and privacy taxonomies, presents details of the Security and Privacy Fabric of the NIST Big Data Reference Architecture (NBDRA), and begins mapping the security and privacy use cases to the NBDRA.

Keywords

Big Data security, Big Data privacy, Big Data taxonomy, use cases, Big Data characteristics, security and privacy fabric, Big Data risk management, cybersecurity, computer security, information assurance, information security frameworks, encryption standards, role-based access controls, Big Data forensics, Big Data audit

Acknowledgements

This document reflects the contributions and discussions by the membership of the NBD-PWG, co-chaired by Wo Chang of the NIST ITL, Robert Marcus of ET-Strategies, and Chaitanya Baru, University of California, San Diego Supercomputer Center.

The document contains input from members of the NBD-PWG Security and Privacy Subgroup, led by Arnab Roy (Fujitsu), Mark Underwood (Krypton Brothers), and Akhil Manchanda (GE); and the Reference Architecture Subgroup.
NIST SP1500-4, Version 1 has been collaboratively authored by the NBD-PWG. As of the date of this publication, there are over six hundred NBD-PWG participants from industry, academia, and government. Federal agency participants include the National Archives and Records Administration (NARA), National Aeronautics and Space Administration (NASA), National Science Foundation (NSF), and the U.S. Departments of Agriculture, Commerce, Defense, Energy, Health and Human Services, Homeland Security, Transportation, Treasury, and Veterans Affairs.

NIST would like to acknowledge specific contributions [a] to this volume by the following NBD-PWG members:

Pw Carey  Orit Levin  Sanjay Mishra
Compliance Partners, LLC  Microsoft  Verizon
Wo Chang  Yale Li  Ann Racuya-Robbins
NIST  Microsoft  World Knowledge Bank
Brent Comstock  Akhil Manchanda  Arnab Roy
Cox Communications  General Electric  Fujitsu
Michele Drgon  Marcia Mangold  Anh-Hong Rucker
Data Probity  General Electric  Jet Propulsion Laboratory
Roy D’Souza  Serge Mankovski  Paul Savitz
AlephCloud Systems, Inc.  CA Technologies  ATIS
Eddie Garcia  Robert Marcus  John Schiel
Gazzang, Inc.  ET-Strategies  CenturyLink, Inc.
David Harper  Lisa Martinez  Mark Underwood
Johns Hopkins University/ Applied Physics Laboratory  Northbound Transportation and Infrastructure, US  Krypton Brothers LLC
Pavithra Kenjige  William Miller  Alicia Zuniga-Alvarado
PK Technologies  MaCT USA  Consultant

The editors for this document were Arnab Roy, Mark Underwood, and Wo Chang.

[a] “Contributors” are members of the NIST Big Data Public Working Group who dedicated great effort to prepare and substantial time on a regular basis to research and development in support of this document.
Notice to Readers

NIST is seeking feedback on the proposed working draft of the NIST Big Data Interoperability Framework: Volume 4, Security and Privacy. Once public comments are received, compiled, and addressed by the NBD-PWG, and reviewed and approved by NIST internal editorial board, Version 1 of this volume will be published as final. Three versions are planned for this volume, with Versions 2 and 3 building on the first. Further explanation of the three planned versions and the information contained therein is included in Section 1.5 of this document.

Please be as specific as possible in any comments or edits to the text. Specific edits include, but are not limited to, changes in the current text, additional text further explaining a topic or explaining a new topic, additional references, or comments about the text, topics, or document organization. These specific edits can be recorded using one of the two following methods.

1. **TRACK CHANGES**: make edits to and comments on the text directly into this Word document using track changes

2. **COMMENT TEMPLATE**: capture specific edits using the Comment Template ([http://bigdatawg.nist.gov/_uploadfiles/SP1500-1-to-7_comment_template.docx](http://bigdatawg.nist.gov/_uploadfiles/SP1500-1-to-7_comment_template.docx)), which includes space for Section number, page number, comment, and text edits

Submit the edited file from either method 1 or 2 to SP1500comments@nist.gov with the volume number in the subject line (e.g., Edits for Volume 4.)

Please contact Wo Chang (wchang@nist.gov) with any questions about the feedback submission process.

Big Data professionals continue to be welcome to join the NBD-PWG to help craft the work contained in the volumes of the NIST Big Data Interoperability Framework. Additional information about the NBD-PWG can be found at [http://bigdatawg.nist.gov](http://bigdatawg.nist.gov).

---

**Table of Contents**

Executive Summary. viii

1 Introduction.. 1

1.1 Background. 1

1.2 Scope and Objectives of the Security and Privacy Subgroup. 2

1.3 Report Production. 3

1.4 Report Structure. 3

1.5 Future Work on this Volume. 3

2 Big Data Security and Privacy. 5

2.1 Overview.. 5
2.2 Effects of Big Data Characteristics on Security and Privacy. 7

2.2.1 Variety. 7

2.2.2 Volume. 8

2.2.3 Velocity. 8

2.2.4 Veracity. 8

2.2.5 Volatility. 9

2.3 Relation to Cloud. 9

3 Example Use Cases for Security and Privacy. 11

3.1 Retail/Marketing. 11

3.1.1 Consumer Digital Media Usage. 11

3.1.2 Nielsen Homescan: Project Apollo. 12

3.1.3 Web Traffic Analytics. 12

3.2 Healthcare. 13

3.2.1 Health Information Exchange. 13

3.2.2 Genetic Privacy. 14

3.2.3 Pharma Clinical Trial Data Sharing. 14

3.3 Cybersecurity. 15

3.3.1 Network Protection. 15

3.4 Government. 15

3.4.1 Military: Unmanned Vehicle Sensor Data. 15

3.4.2 Education: Common Core Student Performance Reporting. 16

3.5 Industrial: Aviation. 16

3.5.1 Sensor Data Storage and Analytics. 16

3.6 Transportation. 17

3.6.1 Cargo Shipping. 17
4 Taxonomy of Security and Privacy Topics. 19

4.1 Conceptual Taxonomy of Security and Privacy Topics. 19

4.1.1 Data Confidentiality. 19

4.1.2 Provenance. 20

4.1.3 System Health. 21

4.1.4 Public Policy, Social and Cross-Organizational Topics. 21

4.2 Operational Taxonomy of Security and Privacy Topics. 22

4.2.1 Device and Application Registration. 22

4.2.2 Identity and Access Management 23

4.2.3 Data Governance. 23

4.2.4 Infrastructure Management 24

4.2.5 Risk and Accountability. 25

4.3 Roles Related to Security and Privacy Topics. 26

4.3.1 Infrastructure Management 26

4.3.2 Governance, Risk Management, and Compliance. 26

4.3.3 Information Worker 27

4.4 Relation of Roles to the Security and Privacy Conceptual taxonomy. 27

4.4.1 Data Confidentiality. 27

4.4.2 Provenance. 27

4.4.3 System Health management 28

4.4.4 Public Policy, Social, and Cross-Organizational Topics. 29

4.5 Additional Taxonomy Topics. 29

4.5.1 Provisioning, Metering, and Billing. 29

4.5.2 Data Syndication. 29

5 Security and Privacy Fabric. 30
5.1 Security and Privacy Fabric in the NBDRA.. 31
5.2 Privacy Engineering Principles. 33
5.3 Relation of the Big Data Security Operational Taxonomy to the NBDRA.. 33

6 Mapping Use Cases to NBDRA.. 35
6.1 Consumer Digital Media Use. 35
6.2 Nielsen Homescan: Project Apollo. 36
6.3 Web Traffic Analytics. 37
6.4 Health Information Exchange. 38
6.5 Genetic Privacy. 40
6.6 Pharmaceutical Clinical Trial Data Sharing. 40
6.7 Network Protection. 41
6.8 Military: Unmanned Vehicle Sensor Data. 41
6.9 Education: Common Core Student Performance Reporting. 43
6.10 Sensor Data Storage and Analytics. 43
6.11 Cargo Shipping. 44

Appendix A: Candidate Security and Privacy Topics for Big Data Adaptation.. A-1
Appendix B: Internal Security Considerations within Cloud Ecosystems. B-1
Appendix C: Big Data Actors and Roles: Adaptation to Big Data Scenarios. C-1
Appendix D: Acronyms. D-1
Appendix E: References. E-1

Figures

Figure 1: Cargo Shipping Scenario. 18
Figure 2: Security and Privacy Conceptual Taxonomy. 19
Figure 3: Security and Privacy Operational Taxonomy. 22
Figure 4: NIST Big Data Reference Architecture. 31
Figure 5: Notional Security and Privacy Fabric Overlay to the NBDRA. 32

Figure B-1: Composite Cloud Ecosystem Security Architecture. B-1

Tables

Table 1: Draft Security Operational Taxonomy Mapping to the NBDRA Components. 33
Table 2: Mapping Consumer Digital Media Usage to the Reference Architecture. 35
Table 3: Mapping Nielsen Homescan to the Reference Architecture. 36
Table 4: Mapping Web Traffic Analytics to the Reference Architecture. 37
Table 5: Mapping HIE to the Reference Architecture. 38
Table 6: Mapping Pharmaceutical Clinical Trial Data Sharing to the Reference Architecture. 40
Table 7: Mapping Network Protection to the Reference Architecture. 41
Table 8: Mapping Military Unmanned Vehicle Sensor Data to the Reference Architecture. 41
Table 9: Mapping Common Core K–12 Student Reporting to the Reference Architecture. 43
Table 10: Mapping Cargo Shipping to the Reference Architecture. 44

Executive Summary

This NIST Big Data Interoperability Framework: Volume 4, Security and Privacy document was prepared by the NIST Big Data Public Working Group (NBD-PWG) Security and Privacy Subgroup to identify security and privacy issues that are specific to Big Data.

Big Data application domains include health care, drug discovery, insurance, finance, retail and many others from both the private and public sectors. Among the scenarios within these application domains are health exchanges, clinical trials, mergers and acquisitions, device telemetry, targeted marketing and international anti-piracy. Security technology domains include identity, authorization, audit, network and device security, and federation across trust boundaries.

Clearly, the advent of Big Data has necessitated paradigm shifts in the understanding and enforcement of security and privacy requirements. Significant changes are evolving, notably in scaling existing solutions to meet the volume, variety, velocity, and variability of Big Data and retargeting security solutions amid shifts in technology infrastructure, e.g., distributed computing systems and non-relational data storage. In addition, diverse datasets are becoming easier to access and increasingly contain personal content. A new set of emerging issues must be addressed, including balancing privacy and utility, enabling analytics and governance on encrypted data, and reconciling authentication and anonymity.

With the key Big Data characteristics of variety, volume, velocity, and variability in mind, the Subgroup gathered use cases from volunteers, developed a consensus-based security and privacy taxonomy, related the taxonomy to the NIST Big Data Reference Architecture (NBDRA), and validated the NBDRA by mapping the use cases to the NBDRA.
The NIST Big Data Interoperability Framework consists of seven volumes, each of which addresses a specific key topic, resulting from the work of the NBD-PWG. The seven volumes are as follows:

- Volume 1, Definitions
- Volume 2, Taxonomies
- Volume 3, Use Cases and General Requirements
- Volume 4, Security and Privacy
- Volume 5, Architectures White Paper Survey
- Volume 6, Reference Architecture
- Volume 7, Standards Roadmap

The NIST Big Data Interoperability Framework will be released in three versions, which correspond to the three stages of the NBD-PWG work. The three stages aim to achieve the following:

Stage 1: Identify the high-level Big Data reference architecture key components, which are technology, infrastructure, and vendor agnostic

Stage 2: Define general interfaces between the NBDRA components

Stage 3: Validate the NBDRA by building Big Data general applications through the general interfaces

Potential areas of future work for the Subgroup during stage 2 are highlighted in Section 1.5 of this volume. The current effort documented in this volume reflects concepts developed within the rapidly evolving field of Big Data.

1 Introduction

1.1 Background

There is broad agreement among commercial, academic, and government leaders about the remarkable potential of Big Data to spark innovation, fuel commerce, and drive progress. Big Data is the common term used to describe the deluge of data in today’s networked, digitized, sensor-laden, and information-driven world. The availability of vast data resources carries the potential to answer questions previously out of reach, including the following:

- How can a potential pandemic reliably be detected early enough to intervene?
- Can new materials with advanced properties be predicted before these materials have ever been synthesized?
- How can the current advantage of the attacker over the defender in guarding against cyber-security threats be reversed?

There is also broad agreement on the ability of Big Data to overwhelm traditional approaches. The growth rates for data volumes, speeds, and complexity are outpacing scientific and technological advances in data analytics, management, transport, and data user spheres.

Despite widespread agreement on the inherent opportunities and current limitations of Big Data, a lack of consensus on some important, fundamental questions continues to confuse potential users and stymie progress. These questions
include the following:

- What attributes define Big Data solutions?
- How is Big Data different from traditional data environments and related applications?
- What are the essential characteristics of Big Data environments?
- How do these environments integrate with currently deployed architectures?
- What are the central scientific, technological, and standardization challenges that need to be addressed to accelerate the deployment of robust Big Data solutions?

Within this context, on March 29, 2012, the White House announced the Big Data Research and Development Initiative.\[1\] The initiative’s goals include helping to accelerate the pace of discovery in science and engineering, strengthening national security, and transforming teaching and learning by improving the ability to extract knowledge and insights from large and complex collections of digital data.

Six federal departments and their agencies announced more than $200 million in commitments spread across more than 80 projects, which aim to significantly improve the tools and techniques needed to access, organize, and draw conclusions from huge volumes of digital data. The initiative also challenged industry, research universities, and nonprofits to join with the federal government to make the most of the opportunities created by Big Data.

Motivated by the White House initiative and public suggestions, the National Institute of Standards and Technology (NIST) has accepted the challenge to stimulate collaboration among industry professionals to further the secure and effective adoption of Big Data. As one result of NIST’s Cloud and Big Data Forum held on January 15–17, 2013, there was strong encouragement for NIST to create a public working group for the development of a Big Data Interoperability Framework. Forum participants noted that this roadmap should define and prioritize Big Data requirements, including interoperability, portability, reusability, extensibility, data usage, analytics, and technology infrastructure. In doing so, the roadmap would accelerate the adoption of the most secure and effective Big Data techniques and technology.

On June 19, 2013, the NIST Big Data Public Working Group (NBD-PWG) was launched with extensive participation by industry, academia, and government from across the nation. The scope of the NBD-PWG involves forming a community of interests from all sectors—including industry, academia, and government—with the goal of developing consensus on definitions, taxonomies, secure reference architectures, security and privacy, and from these a standards roadmap. Such a consensus would create a vendor-neutral, technology- and infrastructure-independent framework that would enable Big Data stakeholders to identify and use the best analytics tools for their processing and visualization requirements on the most suitable computing platform and cluster, while also allowing value-added from Big Data service providers.

The NIST Big Data Interoperability Framework consists of seven volumes, each of which addresses a specific key topic, resulting from the work of the NBD-PWG. The seven volumes are as follows:

- Volume 1, Definitions
- Volume 2, Taxonomies
- Volume 3, Use Cases and General Requirements
- Volume 4, Security and Privacy
- Volume 5, Architectures White Paper Survey
The *NIST Big Data Interoperability Framework* will be released in three versions, which correspond to the three stages of the NBD-PWG work. The three stages aim to achieve the following:

Stage 1: Identify the high-level Big Data reference architecture key components, which are technology, infrastructure, and vendor agnostic.

Stage 2: Define general interfaces between the NIST Big Data Reference Architecture (NBDRA) components.

Stage 3: Validate the NBDRA by building Big Data general applications through the general interfaces.

The NBDRA, created in Stage 1 and further developed in Stages 2 and 3, is a high-level conceptual model designed to serve as a tool to facilitate open discussion of the requirements, structures, and operations inherent in Big Data. It is discussed in detail in *NIST Big Data Interoperability Framework: Volume 6, Reference Architecture*. Potential areas of future work for the Subgroup during stage 2 are highlighted in Section 1.5 of this volume. The current effort documented in this volume reflects concepts developed within the rapidly evolving field of Big Data.

### 1.2 Scope and Objectives of the Security and Privacy Subgroup

The focus of the NBD-PWG Security and Privacy Subgroup is to form a community of interest from industry, academia, and government with the goal of developing consensus on a reference architecture to handle security and privacy issues across all stakeholders. This includes understanding what standards are available or under development, as well as identifying which key organizations are working on these standards.

The scope of the Subgroup’s work includes the following topics, some of which will be addressed in future versions of this Volume:

- Provide a context from which to begin Big Data-specific security and privacy discussions
- Gather input from all stakeholders regarding security and privacy concerns in Big Data processing, storage, and services
- Analyze/prioritize a list of challenging security and privacy requirements that may delay or prevent adoption of Big Data deployment
- Develop a Security and Privacy Reference Architecture that supplements the NBDRA
- Produce a working draft of this Big Data Security and Privacy document
- Develop Big Data security and privacy taxonomies
- Explore mapping between the Big Data security and privacy taxonomies and the NBDRA
- Explore mapping between the use cases and the NBDRA
While there are many issues surrounding Big Data security and privacy, the focus of this Subgroup is on the technology aspects of security and privacy with respect to Big Data.

1.3 Report Production

The NBD-PWG Security and Privacy Subgroup explored various facets of Big Data security and privacy to develop this document. The major steps involved in this effort included:

- Announce that the NBD-PWG Security and Privacy Subgroup is open to the public in order to attract and solicit a wide array of subject matter experts and stakeholders in government, industry, and academia
- Identify use cases specific to Big Data security and privacy
- Develop a detailed security and privacy taxonomy
- Expand the security and privacy fabric of the NBDRA and identify specific topics related to NBDRA components
- Begin mapping of identified security and privacy use cases to the NBDRA

This report is a compilation of contributions from the PWG. Since this is a community effort, there are several topics covered that are related to security and privacy. While an effort has been made to connect the topics, gaps may come to light that could be addressed in Version 2 of this document.

1.4 Report Structure

Following this introductory section, the remainder of this document is organized as follows:

- Section 2 discusses security and privacy issues particular to Big Data
- Section 3 presents examples of security and privacy related use cases
- Section 4 offers a preliminary taxonomy for security and privacy
- Section 5 introduces the details of a draft NIST Big Data security and privacy reference architecture in relation to the overall NBDRA
- Section 6 maps the use cases presented in Section 3 to the NBDRA
- Appendix A discusses special security and privacy topics
- Appendix B contains information about cloud technology
- Appendix C lists the terms and definitions appearing in the taxonomy
- Appendix D contains the acronyms used in this document
- Appendix E lists the references used in the document
1.5 Future Work on this Volume

The NBD-PWG Security and Privacy Subgroup plans to further develop several topics for the subsequent version (i.e., Version 2) of this document. These topics include the following:

- Examining closely other existing templates in literature: The templates may be adapted to the Big Data security and privacy fabric to address gaps and to bridge the efforts of this Subgroup with the work of others.
- Further developing the security and privacy taxonomy
- Enhancing the connection between the security and privacy taxonomy and the NBDRA components
- Developing the connection between the security and privacy fabric and the NBDRA
- Expanding the privacy discussion within the scope of this volume
- Exploring governance, risk management, data ownership, and valuation with respect to Big Data ecosystem, with a focus on security and privacy
- Mapping the identified security and privacy use cases to the NBDRA
- Contextualizing the content of Appendix B in the NBDRA
- Exploring privacy in actionable terms with respect to the NBDRA

Further topics and direction may be added, as warranted, based on future input and contributions to the Subgroup, including those received during the public comments period.

[b] There are multiple templates developed by others to adapt as part of a Big Data security metadata model. For instance, the subgroup has considered schemes offered in the NIST Preliminary Critical Infrastructure Cybersecurity Framework (CIICF) of October 2013, [http://1.usa.gov/1wQuti1](http://1.usa.gov/1wQuti1) (accessed January 9, 2015.).

2 Big Data Security and Privacy

The NBD-PWG Security and Privacy Subgroup began this effort by identifying a number of ways that security and Privacy in Big Data projects can be different from traditional implementations. While not all concepts apply all of the time, the following seven principles were considered representative of a larger set of differences:

1. Big Data projects often encompass heterogeneous components in which a single security scheme has not been designed from the outset.

2. Most security and privacy methods have been designed for batch or online transaction processing systems. Big Data projects increasingly involve one or more streamed data sources that are used in conjunction with data at rest, creating unique security and privacy scenarios.

3. The use of multiple Big Data sources not originally intended to be used together can compromise privacy, security, or both. Approaches to de-identify personally identifiable information (PII) that were satisfactory prior to Big Data may no longer be adequate.

4. An increased reliance on sensor streams, such as those anticipated with the Internet of Things (IoT; e.g., smart medical devices, smart cities, smart homes) can create vulnerabilities that were more easily managed before amassed
5. Certain types of data thought to be too big for analysis, such as geospatial and video imaging, will become commodity Big Data sources. These uses were not anticipated and/or may not have implemented security and privacy measures.

6. Issues of veracity, provenance, and jurisdiction are greatly magnified in Big Data. Multiple organizations, stakeholders, legal entities, governments, and an increasing amount of citizens will find data about themselves included in Big Data analytics.

7. Volatility is significant because Big Data scenarios envision that data is permanent by default. Security is a fast-moving field with multiple attack vectors and countermeasures. Data may be preserved beyond the lifetime of the security measures designed to protect it.

2.1 Overview

Security and privacy measures are becoming ever more important with the increase of Big Data generation and utilization and increasingly public nature of data storage and availability.

The importance of security and privacy measures is increasing along with the growth in the generation, access, and utilization of Big Data. Data generation is expected to double every two years to about 40,000 exabytes in 2020. It is estimated that over one third of the data in 2020 could be valuable if analyzed.[2] Less than a third of data needed protection in 2010, but more than 40% of data will need protection in 2020.[3]

Security and privacy measures for Big Data involve a different approach than traditional systems. Big Data is increasingly stored on public cloud infrastructure built by employing various hardware, operating systems, and analytical software. Traditional security approaches usually addressed small-scale systems holding static data on firewalled and semi-isolated networks. The surge in streaming cloud technology necessitates extremely rapid responses to security issues and threats.[4]

Big Data system representations that rely on concepts of actors and roles present a different facet to security and privacy. The Big Data systems should be adapted to the emerging Big Data landscape, which is embodied in many commercial and open source access control frameworks. These security approaches will likely persist for some time and may evolve with the emerging Big Data landscape. Appendix C considers actors and roles with respect to Big Data security and privacy.

Big Data is increasingly generated and used across diverse industries such as health care, drug discovery, finance, insurance, and marketing of consumer-packaged goods. Effective communication across these diverse industries will require standardization of the terms related to security and privacy. The NBD-PWG Security and Privacy Subgroup aims to encourage participation in the global Big Data discussion with due recognition to the complex and difficult security and privacy requirements particular to Big Data.

There is a large body of work in security and privacy spanning decades of academic study and commercial solutions. While much of that work is not conceptually distinct from Big Data, it may have been produced using different assumptions. One of the primary objectives of this document is to understand how Big Data security and privacy
requirements arise out of the defining characteristics of Big Data, and how these requirements are differentiated from traditional security and privacy requirements.

The following list is a representative—though not exhaustive—list of differences between what is new for Big Data and the requirements that informed previous big system security and privacy.

· **Big Data may be gathered from diverse end points.** Actors include more types than just traditional providers and consumers—data owners, such as mobile users and social network users, are primary actors in Big Data. Devices that ingest data streams for physically distinct data consumers may also be actors. This alone is not new, but the mix of human and device types is on a scale that is unprecedented. The resulting combination of threat vectors and potential protection mechanisms to mitigate them is new.

· **Data aggregation and dissemination must be secured inside the context of a formal, understandable framework.** The availability of data and transparency of its current and past use by data consumers is an important aspect of Big Data. However, Big Data systems may be operational outside formal, readily understood frameworks, such as those designed by a single team of architects with a clearly defined set of objectives. In some settings, where such frameworks are absent or have been unsystematically composed, there may be a need for public or walled garden portals and ombudsman-like roles for data at rest. These system combinations and unforeseen combinations call for a renewed Big Data framework.

· **Data search and selection can lead to privacy or security policy concerns.** There is a lack of systematic understanding of the capabilities that should be provided by a data provider in this respect.[c] A combination of well-educated users, well-educated architects, and system protections may be needed, as well as excluding databases or limiting queries that may be foreseen as enabling re-identification. If a key feature of Big Data is, as one analyst called it, “the ability to derive differentiated insights from advanced analytics on data at any scale,” the search and selection aspects of analytics will accentuate security and privacy concerns. [5]

[c] Reference to NBDRA Data Provider.

· **Privacy-preserving mechanisms are needed for Big Data, such as for Personally Identifiable Information (PII).** Because there may be disparate, potentially unanticipated processing steps between the data owner, provider, and data consumer, the privacy and integrity of data coming from end points should be protected at every stage. End-to-end information assurance practices for Big Data are not dissimilar from other systems but must be designed on a larger scale.

· **Big Data is pushing beyond traditional definitions for information trust, openness, and responsibility.** Governance, previously consigned to static roles and typically employed in larger organizations, is becoming an increasingly important intrinsic design consideration for Big Data systems.

· **Information assurance and disaster recovery for Big Data Systems may require unique and emergent practices.** Because of its extreme scalability, Big Data presents challenges for information assurance (IA) and disaster recovery (DR) practices that were not previously addressed in a systematic way. Traditional backup methods may be impractical for Big Data systems. In addition, test, verification, and provenance assurance for Big Data replicas may not complete in time to meet temporal requirements that were readily accommodated in smaller systems.
Big Data creates potential targets of increased value. The effort required to consummate system attacks will be scaled to meet the opportunity value. Big Data systems will present concentrated, high value targets to adversaries. As Big Data becomes ubiquitous, such targets are becoming more numerous—a new information technology scenario in itself.

Risks have increased for de-anonymization and transfer of PII without consent traceability. Security and privacy can be compromised through unintentional lapses or malicious attacks on data integrity. Managing data integrity for Big Data presents additional challenges related to all the Big Data characteristics, but especially for PII. While there are technologies available to develop methods for de-identification, some experts caution that equally powerful methods can leverage Big Data to re-identify personal information. For example, the availability of unanticipated data sets could make re-identification possible. Even when technology is able to preserve privacy, proper consent and use may not follow the path of the data through various custodians.

Emerging Risks in Open Data and Big Science. Data identification, metadata tagging, aggregation, and segmentation—widely anticipated for data science and open datasets—if not properly managed, may have degraded veracity because they are derived and not primary information sources. Retractions of peer-reviewed research due to inappropriate data interpretations may become more commonplace as researchers leverage third party Big Data.

2.2 Effects of Big Data Characteristics on Security and Privacy

Variety, volume, velocity, and variability are key characteristics of Big Data and commonly referred to as the Vs of Big Data. Where appropriate, these characteristics shaped discussions within the NBD-PWG Security and Privacy Subgroup. While the Vs provide a useful shorthand description, used in the public discourse about Big Data, there are other important characteristics of Big Data that affect security and privacy, such as veracity, validity, and volatility. These elements are discussed below with respect to their impact on Big Data security and privacy.

2.2.1 Variety

Variety describes the organization of the data—whether the data is structured, semi-structured, or unstructured. Retargeting traditional relational database security to non-relational databases has been a challenge. These systems were not designed with security and privacy in mind, and these functions are usually relegated to middleware. Traditional encryption technology also hinders organization of data based on semantics. The aim of standard encryption is to provide semantic security, which means that the encryption of any value is indistinguishable from the encryption of any other value. Therefore, once encryption is applied, any organization of the data that depends on any property of the data values themselves are rendered ineffective, whereas organization of the metadata, which may be unencrypted, may still be effective.

An emergent phenomenon introduced by Big Data variety that has gained considerable importance is the ability to infer identity from anonymized datasets by correlating with apparently innocuous public databases. While several formal models to address privacy preserving data disclosure have been proposed, in practice, sensitive data is shared after sufficient removal of apparently unique identifiers by the processes of anonymization and aggregation. This is an ad hoc process that is often based on empirical evidence and has led to many instances of de-anonymization in conjunction with publicly available data.
2.2.2 Volume

The volume of Big Data describes how much data is coming in. In Big Data parlance, this typically ranges from gigabytes to exabytes. As a result, the volume of Big Data has necessitated storage in multi-tiered storage media. The movement of data between tiers has led to a requirement of cataloging threat models and a surveying of novel techniques. The threat model for network-based, distributed, auto-tier systems includes the following major scenarios: confidentiality and integrity, provenance, availability, consistency, collusion attacks, roll-back attacks and recordkeeping disputes. [11]

A flip side of having volumes of data is that analytics can be performed to help detect security breach events. This is an instance where Big Data technologies can fortify security. This document addresses both facets of Big Data security.

2.2.3 Velocity

Velocity describes the speed at which data is processed. The data usually arrives in batches or is streamed continuously. As with certain other non-relational databases, distributed programming frameworks were not developed with security and privacy in mind.[12] Malfunctioning computing nodes might leak confidential data. Partial infrastructure attacks could compromise a significantly large fraction of the system due to high levels of connectivity and dependency. If the system does not enforce strong authentication among geographically distributed nodes, rogue nodes can be added that can eavesdrop on confidential data.

2.2.4 Veracity

Big Data veracity and validity encompass several subcharacteristics:

Provenance—or what some have called veracity in keeping with the V theme—is important for both data quality and for protecting security and maintaining privacy policies. Big Data frequently moves across individual boundaries to groups and communities of interest, and across state, national, and international boundaries. Provenance addresses the problem of understanding the data's original source, such as through metadata, though the problem extends beyond metadata maintenance. Various approaches have been tried, such as for glycoproteomics,[13] but no clear guidelines yet exist.

A common understanding holds that provenance data is metadata establishing pedigree and chain of custody, including calibration, errors, missing data (e.g., time stamp, location, equipment serial number, transaction number, and authority.)

Some experts consider the challenge of defining and maintaining metadata to be the overarching principle, rather than provenance. The two concepts, though, are clearly interrelated.

Veracity (in some circles also called Provenance, though the two terms are not identical) also encompasses information assurance for the methods through which information was collected. For example, when sensors are used, traceability, calibration, version, sampling, and device configuration is needed.

Curation is an integral concept which binds veracity and provenance to principles of governance as well as to data quality assurance, Curation, for example, may improve raw data by fixing errors, filling in gaps, modeling, calibrating values, ordering data collection.
Validity refers to the accuracy and correctness of data. Traditionally this is referred to data quality. In the Big Data security scenario, validity refers to a host of assumptions about data from which analytics are being applied. For example, continuous and discrete measurements have different properties. The field “gender” can be coded as 1=Male, 2=Female, but 1.5 does not mean halfway between male and female. In the absence of such constraints, an analytical tool can make inappropriate conclusions. There are many types of validity whose constraints are far more complex. By definition, Big Data allows for aggregation and collection across disparate data sets in ways not envisioned by system designers.

Several examples of “invalid” uses for Big Data have been cited. Click fraud, conducted on a Big Data scale, but which can be detected using Big Data techniques, has been cited as the cause of perhaps $11.6 billion in wasted advertisement spending. A software executive listed seven different types of online ad fraud, including non-human generated impressions, non-human generated clicks, hidden ads, misrepresented sources, all-advertising sites, malicious ad injections, and policy-violating content such as pornography or privacy violations.[14] Each of these can be conducted at Big Data scale and may require Big Data solutions to detect and combat.

Despite initial enthusiasm, some trend producing applications that use social media to predict the incidence of flu have been called into question. A study by Lazer et al.[15] suggested that one application overestimated the prevalence of flu for 100 of 108 weeks studied. Careless interpretation of social media is possible when attempts are made to characterize or even predict consumer behavior using imprecise meanings and intentions for “like” and “follow.”

These examples show that what passes for “valid” Big Data can be innocuously lost in translation, interpretation or intentionally corrupted to malicious intent.

2.2.5 Volatility

Volatility of data—how data management changes over time—directly affects provenance. Big Data is transformational in part because systems may produce indefinitely persisting data—data that outlives the instruments on which it was collected; the architects who designed the software that acquired, processed, aggregated, and stored it; and the sponsors who originally identified the project’s data consumers.

Roles are time-dependent in nature. Security and privacy requirements can shift accordingly. Governance can shift as responsible organizations merge or even disappear.

While research has been conducted into how to manage temporal data (e.g., in e-science for satellite instrument data),[16] there are few standards beyond simplistic timestamps and even fewer common practices available as guidance. To manage security and privacy for long-lived Big Data, data temporality should be taken into consideration.

2.3 Relation to Cloud

Many Big Data systems will be designed using cloud architectures. Any strategy to achieve proper access control and security risk management within a Big Data cloud ecosystem enterprise architecture for industry must address the complexities associated with cloud-specific security requirements triggered by cloud characteristics, including, but not limited to, the following:

- Broad network access
• Decreased visibility and control by consumer
• Dynamic system boundaries and commingled roles and responsibilities between consumers and providers
• Multi-tenancy
• Data residency
• Measured service
• Order-of-magnitude increases in scale (on demand), dynamics (elasticity and cost optimization), and complexity (automation and virtualization)

These cloud computing characteristics often present different security risks to an organization than the traditional information technology solutions, altering the organization’s security posture.

To preserve security when migrating data to the cloud, organizations need to identify all cloud-specific, risk-adjusted security controls or components in advance. It may be necessary in some situations to combat security or privacy violations. Each of these can be conducted on Big Data requests from the cloud service providers through contractual means and service-level agreements that all require security components and controls to be fully and accurately implemented.

A further discussion of internal security considerations within cloud ecosystems can be found in Appendix B. Future versions of this document will contextualize the content of Appendix B in the NBDRA.

3 Example Use Cases for Security and Privacy

There are significant Big Data challenges in science and engineering. Many of these are described in the use cases in NIST Big Data Interoperability Framework: Volume 3, Use Cases and General Requirements. However, these use cases focused primarily on science and engineering applications for which security and privacy were secondary concerns—if the latter had any impact on system architecture at all. Consequently, a different set of use cases was developed in the preparation of this document specifically to discover security and privacy issues. Some of these use cases represent inactive or legacy applications, but were selected because they demonstrate characteristic security / privacy design patterns.

The use cases selected for security and privacy are presented in the following subsections. The use cases included are grouped to organize this presentation, as follows: retail/marketing, healthcare, cybersecurity, government, industrial, aviation, and transportation. However, these groups do not represent the entire spectrum of industries affected by Big Data security and privacy.

The use cases were collected when the reference architecture was not mature. The use cases were collected to identify representative security and privacy scenarios thought to be suitably classified as particular to Big Data. An effort was made to map the use cases to the NBDRA. In Version 2, additional mapping of the use cases to the NBDRA and taxonomy will be developed. Parts of this document were developed in parallel and the connections will be strengthened in Version 2.

3.1 Retail/Marketing
3.1.1 Consumer Digital Media Usage

**Scenario Description:** Consumers, with the help of smart devices, have become very conscious of price, convenience, and access before they decide on a purchase. Content owners license data for use by consumers through presentation portals, such as Netflix, iTunes, and others.

Comparative pricing from different retailers, store location and/or delivery options, and crowd-sourced rating have become common factors for selection. To compete, retailers are keeping a close watch on consumer locations, interests, and spending patterns to dynamically create marketing strategies and sell products that consumers do not yet know they want.

**Current Security and Privacy:** Individual data is collected by several means, including smartphone GPS (global positioning system) or location, browser use, social media, and applications (apps) on smart devices.

- Privacy:

  - Most data collection means described above offer weak privacy controls. In addition, consumer unawareness and oversight allow third parties to legitimately capture information. Consumers can have limited to no expectation of privacy in this scenario.

- Security:

  - Controls are inconsistent and/or not established appropriately to achieve the following:

    § Isolation, containerization, and encryption of data

    § Monitoring and detection of threats

    § Identification of users and devices for data feed

    § Interfacing with other data sources

    § Anonymization of users: while some data collection and aggregation uses anonymization techniques, individual users can be re-identified by leveraging other public Big Data pools

    § Original digital rights management (DRM) techniques were not built to scale to meet demand for the forecasted use for the data. “DRM refers to a broad category of access control technologies aimed at restricting the use and copy of digital content on a wide range of devices.”[17] DRM can be compromised, diverted to unanticipated purposes, defeated, or fail to operate in environments with Big Data characteristics—especially velocity and aggregated volume

**Current Research:** There is limited research on enabling privacy and security controls that protect individual data (whether anonymized or non-anonymized).

3.1.2 Nielsen Homescan: Project Apollo

**Scenario Description:** Nielsen Homescan is a subsidiary of Nielsen that collects family-level retail transactions. Project
Apollo was a project designed to better unite advertising content exposure to purchase behavior among Nielsen panelists. Project Apollo did not proceed beyond a limited trial, but reflects a Big Data intent. The description is a best-effort general description and is not an official perspective from Nielsen, Arbitron or the various contractors involved in the project. The information provided here should be taken as illustrative rather than as a historical record.

A general retail transaction has a checkout receipt that contains all SKUs (stock keeping units) purchased, time, date, store location, etc. Nielsen Homescan collected purchase transaction data using a statistically randomized national sample. As of 2005, this data warehouse was already a multi-terabyte data set. The warehouse was built using structured technologies but was built to scale many terabytes. Data was maintained in house by Homescan but shared with customers who were given partial access through a private web portal using a columnar database. Additional analytics were possible using third party software. Other customers would only receive reports that include aggregated data, but greater granularity could be purchased for a fee.

**Then Current (2005-2006) Security and Privacy:**

- **Privacy:** There was a considerable amount of PII data. Survey participants are compensated in exchange for giving up segmentation data, demographics, and other information.
- **Security:** There was traditional access security with group policy, implemented at the field level using the database engine, component-level application security and physical access controls.
- **Opt-out data scrubbing was minimal.**

**3.1.3 Web Traffic Analytics**

**Scenario Description:** Visit-level webservers logs are high-granularity and voluminous. To be useful, log data must be correlated with other (potentially Big Data) data sources, including page content (buttons, text, navigation events), and marketing-level events such as campaigns, media classification, etc. There are discussions—if not deployment—of plans for traffic analytics using complex event processing (CEP) in real time. One nontrivial problem is segregating traffic types, including internal user communities, for which collection policies and security are different.

**Current Security and Privacy:**

- **Non-European Union (EU):** Opt-in defaults are relied upon to gain visitor consent for tracking. Internet Protocol (IP) address logging enables some analysts to identify visitors down to the level of a city block
- **Media access control (MAC) address tracking enables analysts to identify IP devices, which is a form of PII**
- **Some companies allow for purging of data on demand, but most are unlikely to expunge previously collected web server traffic**
- **The EU has stricter regulations regarding collection of such data, which is treated as PII. Such web traffic is to be scrubbed (anonymized) or reported only in aggregate, even for multinationals operating in the EU but based in the United States**
3.2 Healthcare

3.2.1 Health Information Exchange

**Scenario Description:** Health Information Exchanges (HIEs) facilitate sharing of healthcare information that might include electronic health records (EHRs) so that the information is accessible to relevant covered entities, but in a manner that enables patient consent.

HIEs tend to be federated, where the respective covered entity retains custodianship of its data. This poses problems for many scenarios, such as emergencies, for a variety of reasons that include technical (such as interoperability), business, and security concerns.

Cloud enablement of HIEs, through strong cryptography and key management, that meets the Health Insurance Portability and Accountability Act (HIPAA) requirements for protected health information (PHI)—ideally without requiring the cloud service operator to sign a business associate agreement (BAA)—would provide several benefits, including patient safety, lowered healthcare costs, and regulated accesses during emergencies that might include break-the-glass and Centers for Disease Control and Prevention (CDC) scenarios.

The following are some preliminary scenarios that have been proposed by the NBD PWG:

- **Break-the-Glass:** There could be situations where the patient is not able to provide consent due to a medical situation, or a guardian is not accessible, but an authorized party needs immediate access to relevant patient records. Cryptographically enhanced key life cycle management can provide a sufficient level of visibility and nonrepudiation that would enable tracking violations after the fact.

- **Informed Consent:** When there is a transfer of EHRs between covered entities and business associates, it would be desirable and necessary for patients to be able to convey their approval, as well as to specify what components of their EHR can be transferred (e.g., their dentist would not need to see their psychiatric records.) Through cryptographic techniques, one could leverage the ability to specify the fine-grain cipher text policy that would be conveyed. (For related standards efforts regarding consent, see NIST 800-53, Appendix J, Section IP-1), US DHS Health IT Policy Committee, Privacy and Security Workgroup) and Health Level Seven (HL7) International Version 3 standards for Data Access Consent, Consent Directives)

- **Pandemic Assistance:** There will be situations when public health entities, such as the CDC and perhaps other nongovernmental organizations that require this information to facilitate public safety, will require controlled access to this information, perhaps in situations where services and infrastructures are inaccessible. A cloud HIE with the right cryptographic controls could release essential information to authorized entities through authorization and audits in a manner that facilitates the scenario requirement.

**Project Current and/or Proposed Security and Privacy:**

- Security:
  - Lightweight but secure off-cloud encryption: There is a need for the ability to perform lightweight but secure off-cloud encryption of an EHR that can reside in any container that ranges from a browser to an enterprise server, and that...
leverages strong symmetric cryptography

- Homomorphic encryption

- Applied cryptography: Tight reductions, realistic threat models, and efficient techniques

- Privacy:

- Differential privacy: Techniques for guaranteeing against inappropriate leakage of PII

- HIPAA

### 3.2.2 Genetic Privacy

**Scenario Description:** A consortium of policy makers, advocacy organizations, individuals, academic centers, and industry has formed an initiative, Free the Data!, to fill the public information gap caused by the lack of available genetic information for the BRCA1 and BRCA2 genes. The consortium also plans to expand to provide other types of genetic information in open, searchable databases, including the National Center for Biotechnology Information’s database, ClinVar. The primary founders of this project include Genetic Alliance, the University of California San Francisco, InVitae Corporation, and patient advocates.

This initiative invites individuals to share their genetic variation on their own terms and with appropriate privacy settings in a public database so that their family, friends, and clinicians can better understand what the mutation means. Working together to build this resource means working toward a better understanding of disease, higher-quality patient care, and improved human health.

**Current Security and Privacy:**

- Security:

  - Secure Sockets Layer (SSL)-based authentication and access control. Basic user registration with low attestation level

  - Concerns over data ownership and custody upon user death

  - Site administrators may have access to data—strong encryption and key escrow are recommended

- Privacy:

  - Transparent, logged, policy-governed controls over access to genetic information

  - Full lifecycle data ownership and custody controls

### 3.2.3 Pharma Clinical Trial Data Sharing\[18\]

**Scenario Description:** Companies routinely publish their clinical research, collaborate with academic researchers, and
share clinical trial information on public websites, atypically at three different stages: the time of patient recruitment, after new drug approval, and when investigational research programs have been discontinued. Access to clinical trial data is limited, even to researchers and governments, and no uniform standards exist.

The Pharmaceutical Research and Manufacturers of America (PhRMA) represents the country’s leading biopharmaceutical researchers and biotechnology companies. In July 2013, PhRMA joined with the European Federation of Pharmaceutical Industries and Associations (EFPIA) in adopting joint Principles for Responsible Clinical Trial Data Sharing. According to the agreement, companies will apply these Principles as a common baseline on a voluntary basis, and PhRMA encouraged all medical researchers, including those in academia and government, to promote medical and scientific advancement by adopting and implementing the following commitments:

- Enhancing data sharing with researchers
- Enhancing public access to clinical study information
- Sharing results with patients who participate in clinical trials
- Certifying procedures for sharing trial information
- Reaffirming commitments to publish clinical trial results

**Current and Proposed Security and Privacy:**

PhRMA does not directly address security and privacy, but these issues were identified either by PhRMA or by reviewers of the proposal.

- Security:
  - Longitudinal custody beyond trial disposition is unclear, especially after firms merge or dissolve
  - Standards for data sharing are unclear
  - There is a need for usage audit and security
  - Publication restrictions: Additional security will be required to protect the rights of publishers; for example, Elsevier or Wiley

- Privacy:
  - Patient-level data disclosure—elective, per company
  - The PhRMA mentions anonymization (re-identification), but mentions issues with small sample sizes
  - Study-level data disclosure—elective, per company

**3.3 Cybersecurity**
3.3.1 Network Protection

**Scenario Description:** Network protection includes a variety of data collection and monitoring. Existing network security packages monitor high-volume data sets, such as event logs, across thousands of workstations and servers, but they are not yet able to scale to Big Data. Improved security software will include physical data correlates (e.g., access card usage for devices as well as building entrance/exit) and likely be more tightly integrated with applications, which will generate logs and audit records of previously undetermined types or sizes. Big Data analytics systems will be required to process and analyze this data to deliver meaningful results. These systems could also be multi-tenant, catering to more than one distinct company.

This scenario highlights two subscenarios:

- Security for Big Data
- Big Data for security

**Current Security and Privacy:**

- Security in this area is mature; privacy concepts less so.
  - Traditional policy-type security prevails, though temporal dimension and monitoring of policy modification events tends to be nonstandard or unaudited
  - Cybersecurity apps run at high levels of security and thus require separate audit and security measures
  - No cross-industry standards exist for aggregating data beyond operating system collection methods
  - Implementing Big Data cybersecurity should include data governance, encryption/key management, and tenant data isolation/containerization
  - Volatility should be considered in the design of backup and disaster recovery for Big Data cybersecurity. The useful life of logs may extend beyond the lifetime of the devices which created them

- Privacy:
  - Enterprise authorization for data release to state/national organizations
  - Protection of PII data

Currently vendors are adopting Big Data analytics for mass-scale log correlation and incident response, such as for security information and event management (SIEM).

3.4 Government

3.4.1 Military: Unmanned Vehicle Sensor Data

**Scenario Description:** Unmanned vehicles (or drones) and their onboard sensors (e.g., streamed video) can produce
petabytes of data that should be stored in nonstandard formats. These streams are often not processed in real time, but the U.S. Department of Defense (DOD) is buying technology to make this possible. Because correlation is key, GPS, time, and other data streams must be co-collected. The Bradley Manning leak situation is one security breach use case.

**Current Security and Privacy:**

- Separate regulations for agency responsibility apply.
  - For domestic surveillance: The U.S. Federal Bureau of Investigation (FBI)
  - For overseas surveillance: Multiple agencies, including the U.S. Central Intelligence Agency (CIA) and various DOD agencies
    - Not all uses will be military; for example, the National Oceanic and Atmospheric Administration
    - Military security classifications are moderately complex and determined on need to know basis
    - Information assurance practices are rigorously followed, unlike in some commercial settings

**Current Research:**

- Usage is audited where audit means are provided, software is not installed/deployed until ‘certified,’ and development cycles have considerable oversight; for example, the U.S. Army’s Army Regulation 25-2[19]
- Insider threats (e.g., Edward Snowden, Bradley Manning, and spies) are being addressed in programs such as the Defense Advanced Research Projects Agency’s (DARPA) Cyber-Insider Threat (CINDER) program. This research and some of the unfunded proposals made by industry may be of interest

### 3.4.2 Education: Common Core Student Performance Reporting

**Scenario Description:** Forty-five states have decided to unify standards for K–12 student performance measurement. Outcomes are used for many purposes, and the program is incipient, but it will obtain longitudinal Big Data status. The data sets envisioned include student-level performance across students’ entire school history and across schools and states, as well as taking into account variations in test stimuli.

**Current Security and Privacy:**

- Data is scored by private firms and forwarded to state agencies for aggregation. Classroom, school, and district identifiers remain with the scored results. The status of student PII is unknown; however, it is known that teachers receive classroom-level performance feedback. The extent of student/parent access to test results is unclear
- Privacy-related disputes surrounding education Big Data are illustrated by the reluctance of states to participate in the InBloom initiative[20]
- According to some reports, parents can opt students out of state tests, so opt-out records must also be collected and used to purge ineligible student records. [21]
Current Research:

- Longitudinal performance data would have value for program evaluators if data scales up
- Data-driven learning[22] will involve access to students’ performance data, probably more often than at test time, and at higher granularity, thus requiring more data. One example enterprise is Civitas Learning’s[23] predictive analytics for student decision making

3.5 Industrial: Aviation

3.5.1 Sensor Data Storage and Analytics

Scenario Description: Most commercial airlines are equipped with hundreds of sensors to constantly capture engine and/or aircraft health information during a flight. For a single flight, the sensors may collect multiple gigabytes of data and transfer this data stream to Big Data analytics systems. Several companies manage these Big Data analytics systems, such as parts/engine manufacturers, airlines, and plane manufacturers, and data may be shared across these companies. The aggregated data is analyzed for maintenance scheduling, flight routines, etc. One common request from airline companies is to secure and isolate their data from competitors, even when data is being streamed to the same analytics system. Airline companies also prefer to control how, when, and with whom the data is shared, even for analytics purposes. Most of these analytics systems are now being moved to infrastructure cloud providers.

Current and Proposed Security and Privacy:

- Encryption at rest: Big Data systems should encrypt data stored at the infrastructure layer so that cloud storage administrators cannot access the data
- Key management: The encryption key management should be architected so that end customers (e.g., airliners) have sole/shared control on the release of keys for data decryption
- Encryption in motion: Big Data systems should verify that data in transit at the cloud provider is also encrypted
- Encryption in use: Big Data systems will desire complete obfuscation/encryption when processing data in memory (especially at a cloud provider)
- Sensor validation and unique identification (e.g., device identity management)

Researchers are currently investigating the following security enhancements:

- Virtualized infrastructure layer mapping on a cloud provider
- Homomorphic encryption
- Quorum-based encryption
- Multi-party computational capability
3.6 Transportation

3.6.1 Cargo Shipping

The following use case outlines how the shipping industry (e.g., FedEx, UPS, DHL) regularly uses Big Data. Big Data is used in the identification, transport, and handling of items in the supply chain. The identification of an item is important to the sender, the recipient, and all those in between with a need to know the location of the item while in transport and the time of arrival. Currently, the status of shipped items is not relayed through the entire information chain. This will be provided by sensor information, GPS coordinates, and a unique identification schema based on the new International Organization for Standardization (ISO) 29161 standards under development within the ISO technical committee ISO JTC1 SC31 WG2. The data is updated in near real time when a truck arrives at a depot or when an item is delivered to a recipient. Intermediate conditions are not currently known, the location is not updated in real-time, and items lost in a warehouse or while in shipment represent a potential problem for homeland security. The records are retained in an archive and can be accessed for system-determined number of days.

4 Taxonomy of Security and Privacy Topics

A candidate set of topics from the Cloud Security Alliance Big Data Working Group (CSA BDWG) article, Top Ten Challenges in Big Data Security and Privacy Challenges, was used in developing these security and privacy taxonomies. Candidate topics and related material used in preparing this section are provided for reference in Appendix A.

A taxonomy for Big Data security and privacy should encompass the aims of existing, useful taxonomies. While many concepts surrounding security and privacy exist, the objective in the taxonomies contained herein is to highlight and refine new or emerging principles specific to Big Data.
The following subsections present an overview of each security and privacy taxonomy, along with lists of topics encompassed by the taxonomy elements. These lists are the results of preliminary discussions of the Subgroup and may be developed further in Version 2.

4.1 Conceptual Taxonomy of Security and Privacy Topics

The conceptual security and privacy taxonomy, presented in Figure 2, contains four main groups: data confidentiality; data provenance; system health; and public policy, social, and cross-organizational topics. The first three topics broadly correspond with the traditional classification of confidentiality, integrity, and availability (CIA), reoriented to parallel Big Data considerations.

4.1.1 Data Confidentiality

- Confidentiality of data in transit: For example, enforced by using Transport Layer Security (TLS)
- Confidentiality of data at rest
  - Policies to access data based on credentials
  - Systems: Policy enforcement by using systems constructs such as Access Control Lists (ACLs) and Virtual Machine (VM) boundaries
  - Crypto-enforced: Policy enforcement by using cryptographic mechanisms, such as PKI and identity/attribute-based encryption
  - Computing on encrypted data
  - Searching and reporting: Cryptographic protocols that support searching and reporting on encrypted data—any information about the plain text not deducible from the search criteria is guaranteed to be hidden
Homomorphic encryption: Cryptographic protocols that support operations on the underlying plain text of an encryption—any information about the plain text is guaranteed to be hidden

- Secure data aggregation: Aggregating data without compromising privacy
- Data anonymization
- De-identification of records to protect privacy
- Key management

As noted by Chandramouli and Iorga, cloud security for cryptographic keys, an essential building block for security and privacy, takes on "additional complexity," which can be rephrased for Big Data settings: (1) greater variety due to more cloud consumer-provider relationships, and (2) greater demands and variety of infrastructures "on which both the Key Management System and protected resources are located." [25]

Big Data systems are not purely cloud systems, but as is noted elsewhere in this document, the two are closely related. One possibility is to retarget the key management framework that Chandramouli and Iorga developed for cloud service models to the NBDRA security and privacy fabric. Cloud models would correspond to the NBDRA and cloud security concepts to the proposed fabric. NIST 800-145 provides definitions for cloud computing concepts, including infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) cloud service models [26]

Challenges for Big Data key management systems (KMS) reflect demands imposed by Big Data characteristics (i.e., volume, velocity, variety, and variability). For example, leisurely key creation and workflow associated with legacy—and often fastidious—data warehouse key creation is insufficient for Big Data systems deployed quickly and scaled up using massive resources. The lifetime for a Big Data KMS will likely outlive the period of employment of the Big Data system architects who designed it. Designs for location, scale, ownership, custody, provenance, and audit for Big Data key management is an aspect of a security and privacy fabric

4.1.2 Provenance

- End-point input validation: A mechanism to validate whether input data is coming from an authenticated source, such as digital signatures

Syntactic: Validation at a syntactic level

Semantic: Semantic validation is an important concern. Generally, semantic validation would validate typical business rules such as a due date. Intentional or unintentional violation of semantic rules can lock up an application. This could also happen when using data translators that do not recognize the particular variant. Protocols and data formats may be altered by a vendor using, for example, a reserved data field that will allow their products to have capabilities that differentiate them from other products. This problem can also arise in differences in versions of systems for consumer devices, including mobile devices. The semantics of a message and the data to be transported should be validated to verify, at a minimum, conformity with any applicable standards. The use of digital signatures will be important to provide assurance that the data from a sensor or data provider has been verified using a validator or data
checker and is, therefore, valid. This capability is important, particularly if the data is to be transformed or involved in the curation of the data. If the data fails to meet the requirements, it may be discarded, and if the data continues to present a problem, the source may be restricted in its ability to submit the data. These types of errors would be logged and prevented from being disseminated to consumers.

- Digital signatures will be very important in the Big Data system
  - Communication integrity: Integrity of data in transit, enforced, for example, by using TLS
  - Authenticated computations on data: Ensuring that computations taking place on critical fragments of data are indeed the expected computations

- Trusted platforms: Enforcement through the use of trusted platforms, such as Trusted Platform Modules (TPMs)

- Crypto-enforced: Enforcement through the use of cryptographic mechanisms
  - Granular audits: Enabling audit at high granularity
  - Control of valuable assets

- Life cycle management

- Retention and disposition

- DRM

### 4.1.3 System Health

- Security against denial-of-service (DoS)

- Construction of cryptographic protocols proactively resistant to DoS
  - Big Data for Security

- Analytics for security intelligence

- Data-driven abuse detection

- Big Data analytics on logs, cyberphysical events, intelligent agents

- Security breach event detection

- Forensics

- Big Data in support of resilience
4.1.4 Public Policy, Social and Cross-Organizational Topics

The following set of topics is drawn from an Association for Computing Machinery (ACM) grouping.[27] Each of these topics has Big Data security and privacy dimensions that could affect how a fabric overlay is implemented for a specific Big Data project. For instance, a medical devices project might need to address human safety risks, whereas a banking project would be concerned with different regulations applying to Big Data crossing borders. Further work to develop these concepts for Big Data is anticipated by the Subgroup.

- Abuse and crime involving computers
- Computer-related public / private health systems
- Ethics (within data science, but also across professions)
- Human safety
- Intellectual property rights and associated information management[d]
- Regulation
- Transborder data flows
- Use/abuse of power
- Assistive technologies for persons with disabilities (e.g., added or different security / privacy measures may be needed for subgroups within the population)
- Employment (e.g., regulations applicable to workplace law may govern proper use of Big Data produced or managed by employees)
- Social aspects of ecommerce
- Legal: Censorship, taxation, contract enforcement, forensics for law enforcement

[d] For further information, see the frameworks suggested by the Association for Information and Image Management (AIIM; http://www.aiim.org/) and the MIKE 2.0 Information Governance Association (http://mike2.openmethodology.org/wik...ce_Association)

4.2 Operational Taxonomy of Security and Privacy Topics

Current practice for securing Big Data systems is diverse, employing widely disparate approaches that often are not part of a unified conceptual framework. The elements of the operational taxonomy, shown in Figure 3, represent groupings of practical methodologies. These elements are classified as “operational” because they address specific vulnerabilities or risk management challenges to the operation of Big Data systems. At this point in the standards development process, these methodologies have not been incorporated as part of a cohesive security fabric. They are potentially valuable checklist-style elements that can solve specific security or privacy needs. Future work must better integrate these methodologies with risk management guidelines developed by others (e.g., NIST Special Publication 800-37 Guide for Applying the Risk Management Framework to Federal Information Systems[28] and COBIT Risk IT Framework[29].)

In the proposed operational taxonomy, broad considerations of the conceptual taxonomy appear as recurring features. For example, confidentiality of communications can apply to governance of data at rest and access management, but it is also part of a security metadata model.[30]

The operational taxonomy will overlap with small data taxonomies while drawing attention to specific issues with Big Data.[31] [32]
4.2.1 Device and Application Registration

- Device, User, Asset, Services, and Applications Registration: Includes registration of devices in machine to machine (M2M) and IoT networks, DRM-managed assets, services, applications, and user roles

- Security Metadata Model
  - The metadata model maintains relationships across all elements of a secured system. It maintains linkages across all underlying repositories. Big Data often needs this added complexity due to its longer life cycle, broader user community, or other aspects
  - A Big Data model must address aspects such as data velocity, as well as temporal aspects of both data and the life cycle of components in the security model

- Policy Enforcement
  - Environment build
  - Deployment policy enforcement
  - Governance model
  - Granular policy audit
  - Role-specific behavioral profiling

4.2.2 Identity and Access Management

- Virtualization layer identity (e.g., cloud console, platform as a service [PaaS])

- Trusted platforms
Application layer Identity

End-user layer identity management

Roles

Identity provider (IdP)

- An IdP is defined in the Security Assertion Markup Language (SAML). In a Big Data ecosystem of data providers, orchestrators, resource providers, framework providers, and data consumers, a scheme such as the SAML/Security Token Service (STS) or eXtensible Access Control Markup Language (XACML) is seen as a helpful—but not proscriptive—way to decompose the elements in the security taxonomy.

- Big Data may have multiple IdPs. An IdP may issue identities (and roles) to access data from a resource provider. In the SAML framework, trust is shared via SAML/web services mechanisms at the registration phase.

- In Big Data, due to the density of the data, the user “roams” to data (whereas in conventional virtual private network [VPN]-style scenarios, users roam across trust boundaries). Therefore, the conventional authentication/authorization (authn/authz) model needs to be extended because the relying party is no longer fully trusted—they are custodians of somebody else’s data. Data is potentially aggregated from multiple resource providers.

- One approach is to extend the claims-based methods of SAML to add security and privacy guarantees.

Additional XACML Concepts

- XACML introduces additional concepts that may be useful for Big Data security. In Big Data, parties are not just sharing claims, but also sharing policies about what is authorized. There is a policy access point at every data ownership and authoring location, and a policy enforcement point at the data access. A policy enforcement point calls a designated policy decision point for an auditable decision. In this way, the usual meaning of non-repudiation and trusted third parties is extended in XACML. Big Data presumes an abundance of policies, “points,” and identity issuers, as well as data.

  § Policy authoring points
  § Policy decision points
  § Policy enforcement point
  § Policy access points

4.2.3 Data Governance

However large and complex Big Data becomes in terms of data volume, velocity, variety, and variability, Big Data governance will, in some important conceptual and actual dimensions, be much larger. Big Data without Big Data governance may become less useful to its stakeholders. To stimulate positive change, data governance will need to persist across the data lifecycle—at rest, in motion, in incomplete stages, and transactions—while serving the security...
and privacy of the young, the old, individuals as organizations, and organizations as organizations. It will need to cultivate economic benefits and innovation but also enable freedom of action and foster individual and public welfare. It will need to rely on standards governing technologies and practices not fully understood while integrating the human element. Big Data governance will require new perspectives yet accept the slowness or inefficacy of some current techniques. Some data governance considerations are listed below.

**Big Data Apps to Support Governance:** The development of new applications employing Big Data principles and designed to enhance governance may be among the most useful Big Data applications on the horizon.

- Encryption and key management
  - At rest
  - In memory
  - In transit
- Isolation/containerization
- Storage security
- Data loss prevention and detection
- Web services gateway
- Data transformation
- Aggregated data management
- Authenticated computations
- Computations on encrypted data
- Data life cycle management
- Disposition, migration, and retention policies
- PII microdata as “hazardous” [34]
- De-identification and anonymization
- Re-identification risk management
  - End-point validation
  - DRM
  - Trust
4.2.4 Infrastructure Management

Infrastructure management involves security and privacy considerations related to hardware operation and maintenance. Some topics related to infrastructure management are listed below.

- Threat and vulnerability management
  - DoS-resistant cryptographic protocols
  - Monitoring and alerting
    - As noted in the Critical Infrastructure Cybersecurity Framework (CIICF), Big Data affords new opportunities for large-scale security intelligence, complex event fusion, analytics, and monitoring
  - Mitigation
    - Breach mitigation planning for Big Data may be qualitatively or quantitatively different
    - Configuration Management
      - Configuration management is one aspect of preserving system and data integrity. It can include the following:
        - Patch management
        - Upgrades
        - Logging
          - Big Data must produce and manage more logs of greater diversity and velocity. For example, profiling and statistical sampling may be required on an ongoing basis
          - Malware surveillance and remediation
            - This is a well-understood domain, but Big Data can cross traditional system ownership boundaries. Review of NIST’s “Identify, Protect, Detect, Respond, and Recover” framework may uncover planning unique to Big Data
            - Network boundary control
              - Establishes a data-agnostic connection for a secure channel

§ Shared services network architecture, such as those specified as “secure channel use cases and requirements” in the European Telecommunications Standards Institute (ETSI) TS 102 484 Smart Card specifications

§ Zones/cloud network design (including connectivity)
Resilience, Redundancy, and Recovery

Resilience

The security apparatus for a Big Data system may be comparatively fragile in comparison to other systems. A given security and privacy fabric may be required to consider this. Resilience demands are domain-specific, but could entail geometric increases in Big Data system scale.

Redundancy

Redundancy within Big Data systems presents challenges at different levels. Replication to maintain intentional redundancy within a Big Data system takes place at one software level. At another level, entirely redundant systems designed to support failover, resilience or reduced data center latency may be more difficult due to velocity, volume or other aspects of Big Data.

Recovery

Recovery for Big Data security failures may require considerable advance provisioning beyond that required for small data. Response planning and communications with users may be on a similarly large scale.

4.2.5 Risk and Accountability

Risk and accountability encompass the following topics:

- Accountability

  Information, process, and role behavior accountability can be achieved through various means, including:

  - Transparency portals and inspection points
  
  - Forward- and reverse-provenance inspection

- Compliance

  Big Data compliance spans multiple aspects of the security and privacy taxonomy, including privacy, reporting, and nation-specific law

- Forensics

  Forensics techniques enabled by Big Data

  Forensics used in Big Data security failure scenarios

- Business risk level

  Big Data risk assessments should be mapped to each element of the taxonomy. Business risk models can incorporate privacy considerations

https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework
Updated: Sat, 14 Sep 2019 05:26:25 GMT
Powered by mindtouch
4.3 Roles Related to Security and Privacy Topics

Discussions of Big Data security and privacy should be accessible to a diverse audience, including individuals who specialize in cryptography, security, compliance, or information technology. In addition, there are domain experts and corporate decision makers who should understand the costs and impact of these controls. Ideally, these documents would be prefaced by information that would help specialists find the content relevant to them. The specialists could then provide feedback on those sections.

Organizations typically contain diverse roles and workflows for participating in a Big Data ecosystem. Therefore, this document proposes a pattern to help identify the “axis” of an individual’s roles and responsibilities, as well as classify the security controls in a similar manner to make these more accessible to each class.

4.3.1 Infrastructure Management

Typically, the individual role axis contains individuals and groups who are responsible for technical reviews before their organization is on-boarded in a data ecosystem. After the on-boarding, they are usually responsible for addressing defects and security issues.

When infrastructure technology personnel work across organizational boundaries, they accommodate diverse technologies, infrastructures, and workflows and the integration of these three elements. For Big Data security, these include identity, authorization, access control, and log aggregation.

Their backgrounds and practices, as well as the terminologies they use, tend to be uniform, and they face similar pressures within their organizations to constantly do more with less. “Save money” is the underlying theme, and infrastructure technology usually faces pressure when problems arise.

4.3.2 Governance, Risk Management, and Compliance

Data governance is a fundamental element in the management of data and data systems. Data governance refers to administering, or formalizing, discipline (e.g., behavior patterns) around the management of data. Risk management involves the evaluation of positive and negative risks resulting from the handling of Big Data. Compliance encompasses adherence to laws, regulations, protocols, and other guiding rules for operations related to Big Data. Typically, governance, risk management, and compliance (GRC) is a function that draws participation from multiple areas of the organization, such as legal, human resources (HR), information technology (IT), and compliance. In some industries and agencies, there may be a strong focus on compliance, often in isolation from disciplines.

Professionals working in GRC tend to have similar backgrounds, share a common terminology, and employ similar processes and workflows, which typically influence other organizations within the corresponding vertical market or sector.

Within an organization, GRC professionals aim to protect the organization from negative outcomes that might arise from loss of intellectual property, liability due to actions by individuals within the organization, and compliance risks specific to its vertical market.

In larger enterprises and government agencies, GRC professionals are usually assigned to legal, marketing, or
accounting departments or staff positions connected to the CIO. Internal and external auditors are often involved.

Smaller organizations may create, own, or process Big Data, yet may not have GRC systems and practices in place, due to the newness of the Big Data scenario to the organization, a lack of resources, or other factors specific to small organizations. Prior to Big Data, GRC roles in smaller organizations received little attention.

A one-person company can easily construct a Big Data application and inherit numerous unanticipated related GRC responsibilities. This is a new GRC scenario.

A security and privacy fabric entails additional data and process workflow in support of GRC, which is most likely under the control of the System Orchestrator component of the NBDRA, as explained in Section 5.

### 4.3.3 Information Worker

Information workers are individuals and groups who work on the generation, transformation, and consumption of content. Due to the nascent nature of the technologies and related businesses in which they work, they tend to use common terms at a technical level within a specialty. However, their roles and responsibilities and the related workflows do not always align across organizational boundaries. For example, a data scientist has deep specialization in the content and its transformation, but may not focus on security or privacy until it adds effort, cost, risk, or compliance responsibilities to the process of accessing domain-specific data or analytical tools.

Information workers may serve as data curators. Some may be research librarians, operate in quality management roles, or be involved in information management roles such as content editing, search indexing, or performing forensic duties as part of legal proceedings.

Information workers are exposed to a great number of products and services. They are under pressure from their organizations to deliver concrete business value from these new Big Data analytics capabilities by monetizing available data, monetizing the capability to transform data by becoming a service provider, or optimizing and enhancing business by consuming third-party data.

### 4.4 Relation of Roles to the Security and Privacy Conceptual taxonomy

The next sections cover the four components of the conceptual taxonomy: data confidentiality, data provenance, system health, and public policy, social and cross-organizational topics. To leverage these three axes and to facilitate collaboration and education, a stakeholder can be defined as an individual or group within an organization who is directly affected by the selection and deployment of a Big Data solution. A ratifier is defined as an individual or group within an organization who is tasked with assessing the candidate solution before it is selected and deployed. For example, a third-party security consultant may be deployed by an organization as a ratifier, and an internal security specialist with an organization’s IT department might serve as both a ratifier and a stakeholder if tasked with ongoing monitoring, maintenance, and audits of the security.

The upcoming sections also explore potential gaps that would be of interest to the anticipated stakeholders and ratifiers who reside on these three new conceptual axes.
4.4.1 Data Confidentiality

IT specialists who address cryptography should understand the relevant definitions, threat models, assumptions, security guarantees, and core algorithms and protocols. These individuals will likely be ratifiers, rather than stakeholders. IT specialists who address end-to-end security should have an abbreviated view of the cryptography, as well as a deep understanding of how the cryptography would be integrated into their existing security infrastructures and controls.

GRC should reconcile the vertical requirements (e.g., HIPAA requirements related to EHRs) and the assessments by the ratifiers that address cryptography and security. GRC managers would in turn be ratifiers to communicate their interpretation of the needs of their vertical. Persons in these roles also serve as stakeholders due to their participation in internal and external audits and other workflows.

4.4.2 Provenance

Provenance (or veracity) is related in some ways to data privacy, but it might introduce information workers as ratifiers because businesses may need to protect their intellectual property from direct leakage or from indirect exposure during subsequent Big Data analytics. IWs would need to work with the ratifiers from cryptography and security to convey the business need, as well as understand how the available controls may apply.

Similarly, when an organization is obtaining and consuming data, information workers may need to confirm that the data provenance guarantees some degree of information integrity and address incorrect, fabricated, or cloned data before it is presented to an organization.

Additional risks to an organization could arise if one of its data suppliers does not demonstrate the appropriate degree of care in filtering or labeling its data. As noted in the U.S. Department of Health and Human Services (HHS) press release announcing the HIPAA final omnibus rule:

“The changes announced today expand many of the requirements to business associates of these entities that receive protected health information, such as contractors and subcontractors. Some of the largest breaches reported to HHS have involved business associates. Penalties are increased for noncompliance based on the level of negligence with a maximum penalty of $1.5 million per violation.”[38]

Organizations using or sharing health data among ecosystem partners, including mobile apps and SaaS providers, will need to verify that the proper legal agreements are in place to require data veracity and provenance.

4.4.3 System Health management

System health is typically the domain of IT, and IT managers will be ratifiers and stakeholders of technologies, protocols, and products that are used for system health. IT managers will also design how the responsibilities to maintain system health would be shared across the organizations that provide data, analytics, or services—an area commonly known as operations support systems (OSS) in the telecom industry, which has significant experience in syndication of services.

Security and cryptography specialists should scrutinize the system health to spot potential gaps in the operational architectures. The likelihood of gaps increases when a system infrastructure includes diverse technologies and
products.

System health is an umbrella concept that emerges at the intersection of information worker and infrastructure management. As with human health, monitoring nominal conditions for Big Data systems may produce Big Data volume and velocity—two of the Big Data characteristics. Following the human health analogy, some of those potential signals reflect defensive measures such as white cell count. Others could reflect compromised health, such as high blood pressure. Similarly, Big Data systems may employ applications like Security Information and Event Management (SIEM) or Big Data analytics more generally to monitor system health.

Volume, velocity, variety, and variability of Big Data systems health make it different from small data system health. Health tools and design patterns for existing systems are likely insufficient to handle Big Data—including Big Data security and privacy. At least one commercial web services provider has reported that its internal accounting and systems management tool uses more resources than any other single application. The volume of system events and the complexity of event interactions is a challenge that demands Big Data solutions to defend Big Data systems. Managing systems health—including security—will require roles defined as much by the tools needed to manage as by the organizational context. Stated differently, Big Data is transforming the role of the Computer Security Officer.

For example, one aspect motivated by the DevOps movement (i.e., move toward blending tasks performed by applications development and systems operations teams) is the rapid launch, reconfiguration, redeployment and distribution of Big Data systems. Tracking intended vs. accidental or malicious configuration changes is increasingly a Big Data challenge.

### 4.4.4 Public Policy, Social, and Cross-Organizational Topics

Roles in setting public policy related to security and privacy are established in the U.S. by federal agencies such as the Federal Trade Commission, the Food and Drug Administration or the DHHS Office of National Coordinator. DHS is responsible for aspects of domestic U.S. security through the activities of US-CERT. Social roles include the influence of NGO’s, interest groups, professional organizations and standards development organizations. Cross-organizational roles include design patterns employed across or within certain industries such as pharmaceuticals, logistics, manufacturing, distribution to facilitate data sharing, curation, and even orchestration. Big Data frameworks will impact, and are impacted by cross-organizational considerations, possibly industry-by-industry. Further work to develop these concepts for Big Data is anticipated by the Subgroup.

### 4.5 Additional Taxonomy Topics

Additional areas have been identified but not carefully scrutinized, and it is not yet clear whether these would fold into existing categories or if new categories for security and privacy concerns would need to be identified and developed. Some candidate topics are briefly described below.

#### 4.5.1 Provisioning, Metering, and Billing

Provisioning, metering and billing are elements in typically commercial systems used to manage assets, meter their use and invoice clients for that usage. Commercial pipelines for Big Data can be constructed and monetized more readily if these systems are agile in offering services, metering access suitably, and integrating with billing systems. While this
process can be manual for a small number of participants, it can become complex very quickly when there are many suppliers, consumers, and service providers. Information workers and IT professionals who are involved with existing business processes would be candidate ratifiers and stakeholders. Assuring privacy and security of provisioning and metering data may or may not have already been designed into these systems. The scope of metering and billing data will explode, so potential uses and risks have likely not been fully explored.

There are both veracity and validity concerns with these systems. GRC considerations, such as audit and recovery, may overlap with provisioning and metering.

4.5.2 Data Syndication

A feature of Big Data systems is that data is bought and sold as a valuable asset. That Google Search is free relies on users giving up information about their search terms on a Big Data scale. Google and Facebook can choose to repackage and syndicate that information for use by others for a fee.

Similar to service syndication, a data ecosystem is most valuable if any participant can have multiple roles, which could include supplying, transforming, or consuming Big Data. Therefore, a need exists to consider what types of data syndication models should be enabled; again, information workers and IT professionals are candidate ratifiers and stakeholders. For some domains, more complex models may be required to accommodate PII, provenance and governance. Syndication involves transfer of risk and responsibility for security and privacy.

5 Security and Privacy Fabric

Security and privacy considerations are a fundamental aspect of the NBDRA. Using the material gathered for this volume and extensive brainstorming among the NBD-PWG Security and Privacy Subgroup members and others, the following proposal for a security and privacy fabric was developed. [e]

[e] The concept of a “fabric” for security and privacy has precedent in the hardware world, where the notion of a fabric of interconnected nodes in a distributed computing environment was introduced. Computing fabrics were invoked as part of cloud and grid computing, as well as for commercial offerings from both hardware and software manufacturers.

Security and Privacy Fabric: Security and privacy considerations form a fundamental aspect of the NBDRA. This is geometrically depicted in Figure 4 by the Security and Privacy Fabric surrounding the five main components, since all components are affected by security and privacy considerations. Thus, the role of security and privacy is correctly depicted in relation to the components but does not expand into finer details, which may be more accurate but are best relegated to a more detailed security and privacy reference architecture. The Data Provider and Data Consumer are included in the Security and Privacy Fabric since, at the least, they should agree on the security protocols and mechanisms in place. The Security and Privacy Fabric is an approximate representation that alludes to the intricate interconnected nature and ubiquity of security and privacy throughout the NBDRA.

This pervasive dimension is depicted in Figure 4 by the presence of the security and privacy fabric surrounding all of the functional components. NBD-PWG decided to include the Data Provider and Data Consumer as well as the Big Data Application and Framework Providers in the Security and Privacy Fabric because these entities should agree on the security protocols and mechanisms in place. The NIST Big Data Interoperability Framework: Volume 6, Reference
Architecture document discusses in detail the other components of the NBDRA.

At this time, explanations as to how the proposed fabric concept is implemented across each NBDRA component are cursory—more suggestive than prescriptive. However, it is believed that, in time, a template will evolve and form a sound basis for more detailed iterations.

**Figure 4: NIST Big Data Reference Architecture**

Figure 4 introduces two new concepts that are particularly important to security and privacy considerations: information value chain and IT value chain.

**Information value chain:** While it does not apply to all domains, there may be an implied processing progression through which information value is increased, decreased, refined, defined, or otherwise transformed. Application of provenance-preservation and other security mechanisms at each stage may be conditioned by the state-specific contributions to information value.

**IT value chain** Platform-specific considerations apply to Big Data systems when scaled-up or -out. In the process of scaling, specific security, privacy, or GRC mechanism or practices may need to be invoked.

**5.1 Security and Privacy Fabric in the NBDRA**

Figure 5 provides an overview of several security and privacy topics with respect to some key NBDRA components and interfaces. The figure represents a beginning characterization of the interwoven nature of the Security and Privacy Fabric with the NBDRA components.

It is not anticipated that Figure 5 will be further developed for Version 2 of this document. However, the relationships between the Security and Privacy Fabric and the NBDRA and the Security and Privacy Taxonomy and the NBDRA will be investigated for Version 2 of this document.
The groups and interfaces depicted in Figure 5 are described below.

A. Interface between Data Providers Big Data Application Provider

Data coming in from data providers may have to be validated for integrity and authenticity. Incoming traffic may be maliciously used for launching DoS attacks or for exploiting software vulnerabilities on premise. Therefore, real-time security monitoring is useful. Data discovery and classification should be performed in a manner that respects privacy.

B. Interface between Big Data Application Provider Data Consumer

Data, including aggregate results delivered to data consumers, must preserve privacy. Data accessed by third parties or other entities should follow legal regulations such as HIPAA. Concerns include access to sensitive data by the government.

C. Interface between Application Provider Big Data Framework Provider

Data can be stored and retrieved under encryption. Access control policies should be in place to assure that data is only accessed at the required granularity with proper credentials. Sophisticated encryption techniques can allow applications to have rich policy-based access to the data as well as enable searching, filtering on the encrypted data, and computations on the underlying plaintext.

D. Internal to Big Data Framework Provider

Data at rest and transaction logs should be kept secured. Key management is essential to control access and keep track of keys. Non-relational databases should have a layer of security measures. Data provenance is essential to having proper context for security and function of the data at every stage. DoS attacks should be mitigated to assure availability of the data.

E. System Orchestrator
A System Orchestrator may play a critical role in identifying, managing, auditing, and sequencing Big Data processes across the components. For example, a workflow that moves data from a collection stage to further preparation may implement aspects of security or privacy.

System Orchestrators present an additional, attractive attack surface for adversaries. System Orchestrators often require permanent or transitory elevated permissions. System Orchestrators present opportunities to implement security mechanisms, monitor provenance, access systems management tools, provide audit points, and inadvertently subjugate privacy or other information assurance measures.

### 5.2 Privacy Engineering Principles

Big Data security and privacy should leverage existing standards and practices. In the privacy arena, a systems approach that considers privacy throughout the process is a useful guideline to consider when adapting security and privacy practices to Big Data scenarios. The Organization for the Advancement of Structured Information Standards (OASIS) Privacy Management Reference Model (PMRM), consisting of seven foundational principles, provides appropriate basic guidance for Big System architects. [39][40] When working with any personal data, privacy should be an integral element in the design of a Big Data system.

Other privacy engineering frameworks are also under consideration. [41] [42] [43] [44] [45] [46]

Related principles include identity management frameworks such as proposed in the National Strategy for Trusted Identities in Cyberspace (NSTIC) [47] and considered in the NIST Cloud Computing Security Reference Architecture. [48] Aspects of identity management that contribute to a security and privacy fabric will be addressed in future versions of this document.

Big Data frameworks can also be used for strengthening security. Big Data analytics can be used for detecting privacy breaches through security intelligence, event detection, and forensics.

### 5.3 Relation of the Big Data Security Operational Taxonomy to the NBDRA

Table 1 represents a preliminary mapping of the operational taxonomy to the NBDRA components. The topics and activities listed for each operational taxonomy element (Section 4.2) have been allocated to a NBDRA component under the Activities column in Table 1. The description column provides additional information about the security and privacy aspects of each NBDRA component.

<table>
<thead>
<tr>
<th>Activities</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Orchestrator</td>
<td>Several security functions have been mapped to</td>
</tr>
<tr>
<td>· Policy Enforcement</td>
<td></td>
</tr>
</tbody>
</table>

https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework
Updated: Sat, 14 Sep 2019 05:26:25 GMT
Powered by mindtouch™
### Activities

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>· Security Metadata Model</td>
</tr>
<tr>
<td>· Data Loss Prevention, Detection</td>
</tr>
<tr>
<td>· Data Lifecycle Management</td>
</tr>
<tr>
<td>· Threat and Vulnerability Management</td>
</tr>
<tr>
<td>· Mitigation</td>
</tr>
<tr>
<td>· Configuration Management</td>
</tr>
<tr>
<td>· Monitoring, Alerting</td>
</tr>
<tr>
<td>· Malware Surveillance and Remediation</td>
</tr>
<tr>
<td>· Resiliency, Redundancy and Recovery</td>
</tr>
<tr>
<td>· Accountability</td>
</tr>
<tr>
<td>· Compliance</td>
</tr>
<tr>
<td>· Forensics</td>
</tr>
<tr>
<td>· Business Risk Model</td>
</tr>
</tbody>
</table>

Aspects of these functionalities are strongly related to the Security Fabric and thus touch the entire architecture at various points in different forms of operational details. Such security functions include nation-specific compliance requirements, vastly expanded demand for forensics, and domain-specific, privacy-aware business risk models.

### Data Provider

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>· Device, User, Asset, Services, Applications Registration</td>
</tr>
<tr>
<td>· Application Layer Identity</td>
</tr>
<tr>
<td>· End User Layer Identity Management</td>
</tr>
<tr>
<td>· End Point Input Validation</td>
</tr>
<tr>
<td>· Digital Rights Management</td>
</tr>
<tr>
<td>· Monitoring, Alerting</td>
</tr>
</tbody>
</table>

Data Providers are subject to guaranteeing authenticity of data and in turn require that sensitive, copyrighted, or valuable data be adequately protected. This leads to operational aspects of entity registration and identity ecosystems.
<table>
<thead>
<tr>
<th>Activities</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Consumer</strong></td>
<td></td>
</tr>
<tr>
<td>· Application Layer Identity</td>
<td></td>
</tr>
<tr>
<td>· End User Layer Identity Management</td>
<td></td>
</tr>
<tr>
<td>· Web Services Gateway</td>
<td></td>
</tr>
<tr>
<td>· Digital Rights Management</td>
<td></td>
</tr>
<tr>
<td>· Monitoring, Alerting</td>
<td>Data Consumers exhibit a duality with Data Providers in terms of obligations and requirements -- only they face the access/visualization aspects of the Application Provider.</td>
</tr>
<tr>
<td><strong>Application Provider</strong></td>
<td></td>
</tr>
<tr>
<td>· Application Layer Identity</td>
<td></td>
</tr>
<tr>
<td>· Web Services Gateway</td>
<td>Application Provider interfaces between the Data Provider and Data Consumer. It takes part in all the secure interface protocols with these blocks as well as maintains secure interaction with the Framework Provider.</td>
</tr>
<tr>
<td>· Data Transformation</td>
<td></td>
</tr>
<tr>
<td>· Digital Rights Management</td>
<td></td>
</tr>
<tr>
<td>· Monitoring, Alerting</td>
<td></td>
</tr>
<tr>
<td><strong>Framework Provider</strong></td>
<td></td>
</tr>
<tr>
<td>· Virtualization Layer Identity</td>
<td></td>
</tr>
<tr>
<td>· Identity Provider</td>
<td>Framework Provider is responsible for the security of data/computations for a significant portion of the lifecycle of the data. This includes security of data at rest through encryption and access control; security of computations via isolation/virtualization; and security of communication with the Application Provider.</td>
</tr>
<tr>
<td>· Encryption and Key Management</td>
<td></td>
</tr>
<tr>
<td>· Isolation/Containerization</td>
<td></td>
</tr>
<tr>
<td>· Storage Security</td>
<td></td>
</tr>
<tr>
<td>· Network Boundary Control</td>
<td></td>
</tr>
<tr>
<td>· Monitoring, Alerting</td>
<td></td>
</tr>
</tbody>
</table>
6 Mapping Use Cases to NBDRA

In this section, the security and privacy related use cases presented in Section 3 are mapped to the NBDRA components and interfaces explored in Figure 5, Notional Security and Privacy Fabric Overlay to the NBDRA.

6.1 Consumer Digital Media Use

Content owners license data for use by consumers through presentation portals. The use of consumer digital media generates Big Data, including both demographics at the user level and patterns of use such as play sequence, recommendations, and content navigation.

Table 2: Mapping Consumer Digital Media Usage to the Reference Architecture

<table>
<thead>
<tr>
<th>NBDRA Component and Interfaces</th>
<th>Security and Privacy Topic</th>
<th>Use Case Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Provider → Application Provider</td>
<td>End-point input validation</td>
<td>Varies and is vendor dependent. Spoofing is possible. For example, protections afforded by securing Microsoft Rights Management Services. [49] Secure/Multipurpose Internet Mail Extensions (S/MIME)</td>
</tr>
<tr>
<td></td>
<td>Real-time security monitoring</td>
<td>Content creation security</td>
</tr>
<tr>
<td></td>
<td>Data discovery and classification</td>
<td>Discovery/classification is possible across media, populations, and channels</td>
</tr>
<tr>
<td></td>
<td>Secure data aggregation</td>
<td>Vendor-supplied aggregation services—security practices are opaque</td>
</tr>
<tr>
<td>Application Provider → Data Consumer</td>
<td>Privacy-preserving data analytics</td>
<td>Aggregate reporting to content owners</td>
</tr>
<tr>
<td></td>
<td>Compliance with regulations</td>
<td>PII disclosure issues abound</td>
</tr>
<tr>
<td></td>
<td>Government access to data and</td>
<td>Various issues; for example, playing terrorist</td>
</tr>
<tr>
<td><strong>NBDRA Component and Interfaces</strong></td>
<td><strong>Security and Privacy Topic</strong></td>
<td><strong>Use Case Mapping</strong></td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td></td>
<td>freedom of expression concerns</td>
<td>podcast and illegal playback</td>
</tr>
<tr>
<td>Data Provider ↔ Framework Provider</td>
<td>Data-centric security such as identity/policy-based encryption</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>Policy management for access control</td>
<td>User, playback administrator, library maintenance, and auditor</td>
</tr>
<tr>
<td></td>
<td>Computing on the encrypted data: searching/ filtering/ deduplicate/ fully homomorphic encryption</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>Audits</td>
<td>Audit DRM usage for royalties</td>
</tr>
<tr>
<td>Framework Provider</td>
<td>Securing data storage and transaction logs</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>Key management</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>Security best practices for non-relational data stores</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>Security against DoS attacks</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Data provenance</td>
<td>Traceability to data owners, producers, consumers is preserved</td>
</tr>
<tr>
<td>Fabric</td>
<td>Analytics for security intelligence</td>
<td>Machine intelligence for unsanctioned use/access</td>
</tr>
</tbody>
</table>
### Table 3: Mapping Nielsen Homescan to the Reference Architecture

<table>
<thead>
<tr>
<th>NBDRA Component and Interfaces</th>
<th>Security and Privacy Topic</th>
<th>Use Case Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Provider → Application Provider</td>
<td>End-point input validation</td>
<td>Device-specific keys from digital sources; receipt sources scanned internally and reconciled to family ID (Role issues)</td>
</tr>
<tr>
<td></td>
<td>Real-time security monitoring</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>Data discovery and classification</td>
<td>Classifications based on data sources (e.g., retail outlets, devices, and paper sources)</td>
</tr>
<tr>
<td></td>
<td>Secure data aggregation</td>
<td>Aggregated into demographic crosstabs. Internal analysts had access to PII</td>
</tr>
<tr>
<td>Application Provider → Data Consumer</td>
<td>Privacy-preserving data analytics</td>
<td>Aggregated to (sometimes) product-specific, statistically valid independent variables</td>
</tr>
</tbody>
</table>

6.2 Nielsen Homescan: Project Apollo

Nielsen Homescan involves family-level retail transactions and associated media exposure using a statistically valid national sample. A general description[50] is provided by the vendor. This project description is based on a 2006 Project Apollo architecture. (Project Apollo did not emerge from its prototype status.)
<table>
<thead>
<tr>
<th>NBDRA Component and Interfaces</th>
<th>Security and Privacy Topic</th>
<th>Use Case Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compliance with regulations</td>
<td>Panel data rights secured in advance and enforced through organizational controls</td>
<td></td>
</tr>
<tr>
<td>Government access to data and freedom of expression concerns</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Data-centric security such as identity/policy-based encryption</td>
<td>Encryption not employed in place; only for data-center-to-data-center transfers. XML (Extensible Markup Language) cube security mapped to Sybase IQ and reporting tools</td>
<td></td>
</tr>
<tr>
<td>Policy management for access control</td>
<td>Extensive role-based controls</td>
<td></td>
</tr>
<tr>
<td>Computing on the encrypted data: searching/filtering/deduplicate/fully homomorphic encryption</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Audits</td>
<td>Schematron and process step audits</td>
<td></td>
</tr>
<tr>
<td>Securing data storage and transaction logs</td>
<td>Project-specific audits secured by infrastructure team</td>
<td></td>
</tr>
<tr>
<td>Key management</td>
<td>Managed by project chief security officer (CSO). Separate key pairs issued for customers and internal users</td>
<td></td>
</tr>
<tr>
<td>Security best practices for non-relational data stores</td>
<td>Regular data integrity checks via XML schema validation</td>
<td></td>
</tr>
</tbody>
</table>
### 6.3 Web Traffic Analytics

Visit-level webserver logs are of high-granularity and voluminous. Web logs are correlated with other sources, including page content (buttons, text, and navigation events) and marketing events such as campaigns and media classification.

#### Table 4: Mapping Web Traffic Analytics to the Reference Architecture

<table>
<thead>
<tr>
<th>NBDRA Component and Interfaces</th>
<th>Security and Privacy Topic</th>
<th>Use Case Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Provider → Application Provider</td>
<td>End-point input validation</td>
<td>Device-dependent. Spoofing is often easy</td>
</tr>
<tr>
<td></td>
<td>Real-time security monitoring</td>
<td>Web server monitoring</td>
</tr>
<tr>
<td></td>
<td>Data discovery and classification</td>
<td>Some geospatial attribution</td>
</tr>
<tr>
<td>NBDRA Component and Interfaces</td>
<td>Security and Privacy Topic</td>
<td>Use Case Mapping</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td>Secure data aggregation</td>
<td>Aggregation to device, visitor, button, web event, and others</td>
</tr>
<tr>
<td>Application Provider → Data Consumer</td>
<td>Privacy-preserving data analytics</td>
<td>IP anonymizing and timestamp degrading. Content-specific opt-out</td>
</tr>
<tr>
<td></td>
<td>Compliance with regulations</td>
<td>Anonymization may be required for EU compliance. Opt-out honoring</td>
</tr>
<tr>
<td></td>
<td>Government access to data and freedom of expression concerns</td>
<td>Yes</td>
</tr>
<tr>
<td>Data Provider ↔ Framework Provider</td>
<td>Data-centric security such as identity/policy-based encryption</td>
<td>Varies depending on archivist</td>
</tr>
<tr>
<td></td>
<td>Policy management for access control</td>
<td>System- and application-level access controls</td>
</tr>
<tr>
<td></td>
<td>Computing on the encrypted data: searching/filtering/deduplicate/fully homomorphic encryption</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>Audits</td>
<td>Customer audits for accuracy and integrity are supported</td>
</tr>
<tr>
<td>Framework Provider</td>
<td>Securing data storage and transaction logs</td>
<td>Storage archiving—this is a big issue</td>
</tr>
<tr>
<td></td>
<td>Key management</td>
<td>CSO and applications</td>
</tr>
</tbody>
</table>
### NBDRA Component and Interfaces

<table>
<thead>
<tr>
<th>Security and Privacy Topic</th>
<th>Use Case Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security best practices for non-relational data stores</td>
<td>Unknown</td>
</tr>
<tr>
<td>Security against DoS attacks</td>
<td>Standard</td>
</tr>
<tr>
<td>Data provenance</td>
<td>Server, application, IP-like identity, page point-in-time Document Object Model (DOM), and point-in-time marketing events</td>
</tr>
<tr>
<td>Analytics for security intelligence</td>
<td>Access to web logs often requires privilege elevation</td>
</tr>
<tr>
<td>Event detection</td>
<td>Can infer; for example, numerous sales, marketing, and overall web health events</td>
</tr>
<tr>
<td>Forensics</td>
<td>See the SIEM use case</td>
</tr>
</tbody>
</table>

### 6.4 Health Information Exchange

Health information exchange (HIE) data is aggregated from various data providers, which might include covered entities such as hospitals and contract research organizations (CROs) identifying participation in clinical trials. The data consumers would include emergency room personnel, the CDC, and other authorized health (or other) organizations. Because any city or region might implement its own HIE, these exchanges might also serve as data consumers and data providers for each other.
## Table 5: Mapping HIE to the Reference Architecture

<table>
<thead>
<tr>
<th>NBDRA Component and Interfaces</th>
<th>Security and Privacy Topic</th>
<th>Use Case Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Provider → Application Provider</td>
<td>End-point input validation</td>
<td>Strong authentication, perhaps through X.509v3 certificates, potential leverage of SAFE (Signatures &amp; Authentication for Everything[^1]) bridge in lieu of general PKI</td>
</tr>
<tr>
<td></td>
<td>Real-time security monitoring</td>
<td>Validation of incoming records to assure integrity through signature validation and to assure HIPAA privacy through ensuring PHI is encrypted. May need to check for evidence of informed consent</td>
</tr>
<tr>
<td></td>
<td>Data discovery and classification</td>
<td>Leverage Health Level Seven (HL7) and other standard formats opportunistically, but avoid attempts at schema normalization. Some columns will be strongly encrypted while others will be specially encrypted (or associated with cryptographic metadata) for enabling discovery and classification. May need to perform column filtering based on the policies of the data source or the HIE service provider</td>
</tr>
<tr>
<td></td>
<td>Secure data aggregation</td>
<td>Clear text columns can be deduplicated, perhaps columns with deterministic encryption. Other columns may have cryptographic metadata for facilitating aggregation and deduplication. Retention rules are assumed, but disposition rules are not assumed in the related areas of compliance</td>
</tr>
<tr>
<td>Application Provider → Data Consumer</td>
<td>Privacy-preserving data analytics</td>
<td>Searching on encrypted data and proofs of data possession. Identification of potential adverse experience due to clinical trial participation.</td>
</tr>
</tbody>
</table>

[^1]: [SAFE](https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework)
<table>
<thead>
<tr>
<th>NBDRA Component and Interfaces</th>
<th>Security and Privacy Topic</th>
<th>Use Case Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Identification of potential professional patients. Trends and epidemics, and co-relations of these to environmental and other effects. Determination of whether the drug to be administered will generate an adverse reaction, without breaking the double blind. Patients will need to be provided with detailed accounting of accesses to, and uses of, their EHR data</td>
<td></td>
</tr>
<tr>
<td>Compliance with regulations</td>
<td>HIPAA security and privacy will require detailed accounting of access to EHR data. Facilitating this, and the logging and alerts, will require federated identity integration with data consumers</td>
<td></td>
</tr>
<tr>
<td>Government access to data and freedom of expression concerns</td>
<td>CDC, law enforcement, subpoenas and warrants. Access may be toggled based on occurrence of a pandemic (e.g., CDC) or receipt of a warrant (e.g., law enforcement)</td>
<td></td>
</tr>
<tr>
<td>Data Provider ↔ Provider Framework</td>
<td>Data-centric security such as identity/policy-based encryption</td>
<td>Row-level and column-level access control</td>
</tr>
<tr>
<td></td>
<td>Policy management for access control</td>
<td>Role-based and claim-based. Defined for PHI cells</td>
</tr>
<tr>
<td></td>
<td>Computing on the encrypted data: searching/filtering/deduplicate/fully homomorphic encryption</td>
<td>Privacy-preserving access to relevant events, anomalies, and trends for CDC and other relevant health organizations</td>
</tr>
<tr>
<td>Audits</td>
<td>Facilitate HIPAA readiness and HHS audits</td>
<td></td>
</tr>
<tr>
<td>NBDRA Component and Interfaces</td>
<td>Security and Privacy Topic</td>
<td>Use Case Mapping</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td><strong>Framework Provider</strong></td>
<td>Securing data storage and transaction logs</td>
<td>Need to be protected for integrity and privacy, but also for establishing completeness, with an emphasis on availability</td>
</tr>
<tr>
<td></td>
<td>Key management</td>
<td>Federated across covered entities, with the need to manage key life cycles across multiple covered entities that are data sources</td>
</tr>
<tr>
<td></td>
<td>Security best practices for non-relational data stores</td>
<td>End-to-end encryption, with scenario-specific schemes that respect min-entropy to provide richer query operations without compromising patient privacy</td>
</tr>
<tr>
<td></td>
<td>Security against distributed denial of Service (DDoS) attacks</td>
<td>A mandatory requirement: systems must survive DDoS attacks</td>
</tr>
<tr>
<td></td>
<td>Data provenance</td>
<td>Completeness and integrity of data with records of all accesses and modifications. This information could be as sensitive as the data and is subject to commensurate access policies</td>
</tr>
<tr>
<td><strong>Fabric</strong></td>
<td>Analytics for security intelligence</td>
<td>Monitoring of informed patient consent, authorized and unauthorized transfers, and accesses and modifications</td>
</tr>
<tr>
<td></td>
<td>Event detection</td>
<td>Transfer of record custody, addition/ modification of record (or cell), authorized queries, unauthorized queries, and modification attempts</td>
</tr>
<tr>
<td></td>
<td>Forensics</td>
<td>Tamper-resistant logs, with evidence of tampering events. Ability to identify record-level tampering</td>
</tr>
</tbody>
</table>

https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework
Updated: Sat, 14 Sep 2019 05:26:25 GMT
Powered by mindtouch™
6.5 Genetic Privacy

Mapping of genetic privacy is under development and will be included in future versions of this document.

6.6 Pharmaceutical Clinical Trial Data Sharing

Under an industry trade group proposal, clinical trial data for new drugs will be shared outside intra-enterprise warehouses.

<table>
<thead>
<tr>
<th>NBDRA Component and Interfaces</th>
<th>Security and Privacy Topic</th>
<th>Use Case Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Provider → Application Provider</td>
<td>End-point input validation</td>
<td>Opaque—company-specific</td>
</tr>
<tr>
<td></td>
<td>Real-time security monitoring</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>Data discovery and classification</td>
<td>Opaque—company-specific</td>
</tr>
<tr>
<td></td>
<td>Secure data aggregation</td>
<td>Third-party aggregator</td>
</tr>
<tr>
<td>Application Provider → Data Consumer</td>
<td>Privacy-preserving data analytics</td>
<td>Data to be reported in aggregate but preserving potentially small-cell demographics</td>
</tr>
<tr>
<td></td>
<td>Compliance with regulations</td>
<td>Responsible developer and third-party custodian</td>
</tr>
<tr>
<td>NBDRA Component and Interfaces</td>
<td>Security &amp; Privacy Topic</td>
<td>Use Case Mapping</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td>Government access to data and freedom of expression concerns</td>
<td>Limited use in research community, but there are possible future public health data concerns. Clinical study reports only, but possibly selectively at the study- and patient-levels</td>
</tr>
<tr>
<td>Data Provider ↔ Framework Provider</td>
<td>Data-centric security such as identity/policy-based encryption</td>
<td>TBD</td>
</tr>
<tr>
<td></td>
<td>Policy management for access control</td>
<td>Internal roles; third-party custodian roles; researcher roles; participating patients' physicians</td>
</tr>
<tr>
<td></td>
<td>Computing on the encrypted data: searching/filtering/deduplicate/fully homomorphic encryption</td>
<td>TBD</td>
</tr>
<tr>
<td></td>
<td>Audits</td>
<td>Release audit by a third party</td>
</tr>
<tr>
<td>Framework Provider</td>
<td>Securing data storage and transaction logs</td>
<td>TBD</td>
</tr>
<tr>
<td></td>
<td>Key management</td>
<td>Internal varies by firm; external TBD</td>
</tr>
<tr>
<td></td>
<td>Security best practices for non-relational data stores</td>
<td>TBD</td>
</tr>
<tr>
<td></td>
<td>Security against DoS attacks</td>
<td>Unlikely to become public</td>
</tr>
<tr>
<td></td>
<td>Data provenance</td>
<td>TBD—critical issue</td>
</tr>
</tbody>
</table>
### 6.7 Network Protection

SIEM is a family of tools used to defend and maintain networks.

Table 7: Mapping Network Protection to the Reference Architecture

<table>
<thead>
<tr>
<th>NBDRA Component and Interfaces</th>
<th>Security and Privacy Topic</th>
<th>Use Case Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Provider → Application Provider</td>
<td>End-point input validation</td>
<td>Software-supplier specific; refer to commercially available end point validation[52]</td>
</tr>
<tr>
<td></td>
<td>Real-time security monitoring</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>Data discovery and classification</td>
<td>Varies by tool, but classified based on security semantics and sources</td>
</tr>
<tr>
<td></td>
<td>Secure data aggregation</td>
<td>Aggregates by subnet, workstation, and server</td>
</tr>
<tr>
<td>Application Provider → Data Consumer</td>
<td>Privacy-preserving data analytics</td>
<td>Platform-specific</td>
</tr>
<tr>
<td></td>
<td>Compliance with regulations</td>
<td>Applicable, but regulated events are not readily</td>
</tr>
<tr>
<td>NBDRA Component and Interfaces</td>
<td>Security and Privacy Topic</td>
<td>Use Case Mapping</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>visible to analysts</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Government access to data and freedom of expression concerns</td>
<td>NSA and FBI have access on demand</td>
</tr>
<tr>
<td>Data Provider ↔ Framework</td>
<td>Data-centric security such as identity/policy-based encryption</td>
<td>Usually a feature of the operating system</td>
</tr>
<tr>
<td>Provider</td>
<td>Policy management for access control</td>
<td>For example, a group policy for an event log</td>
</tr>
<tr>
<td></td>
<td>Computing on the encrypted data: searching/filtering/deduplicate/fully homomorphic encryption</td>
<td>Vendor and platform-specific</td>
</tr>
<tr>
<td></td>
<td>Audits</td>
<td>Complex—audits are possible throughout</td>
</tr>
<tr>
<td>Framework Provider</td>
<td>Securing data storage and transaction logs</td>
<td>Vendor and platform-specific</td>
</tr>
<tr>
<td></td>
<td>Key management</td>
<td>Chief Security Officer and SIEM product keys</td>
</tr>
<tr>
<td></td>
<td>Security best practices for non-relational data stores</td>
<td>TBD</td>
</tr>
<tr>
<td></td>
<td>Security against DDoS attacks</td>
<td>Big Data application layer DDoS attacks can be mitigated using combinations of traffic analytics, correlation analysis</td>
</tr>
</tbody>
</table>
6.8 Military: Unmanned Vehicle Sensor Data

Unmanned vehicles (drones) and their onboard sensors (e.g., streamed video) can produce petabytes of data that should be stored in nonstandard formats. The U.S. government is pursuing capabilities to expand storage capabilities for Big Data such as streamed video. For more information, refer to the Defense Information Systems Agency (DISA) large data object contract for exabytes in the DOD private cloud. [53]

Table 8: Mapping Military Unmanned Vehicle Sensor Data to the Reference Architecture

<table>
<thead>
<tr>
<th>NBDRA Component and Interfaces</th>
<th>Security and Privacy Topic</th>
<th>Use Case Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Provider → Application Provider</td>
<td>End-point input validation</td>
<td>Need to secure the sensor (e.g., camera) to prevent spoofing/stolen sensor streams. There are new transceivers and protocols in the DOD pipeline. Sensor streams will include smartphone and tablet sources</td>
</tr>
<tr>
<td></td>
<td>Real-time security monitoring</td>
<td>Onboard and control station secondary sensor security monitoring</td>
</tr>
<tr>
<td>NBDRA Component and Interfaces</td>
<td>Security and Privacy Topic</td>
<td>Use Case Mapping</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td>Data discovery and classification</td>
<td>Varies from media-specific encoding to sophisticated situation-awareness enhancing fusion schemes</td>
</tr>
<tr>
<td></td>
<td>Secure data aggregation</td>
<td>Fusion challenges range from simple to complex. Video streams may be used[54] unsecured or unaggregated</td>
</tr>
<tr>
<td>Application Provider → Data Consumer</td>
<td>Privacy-preserving data analytics</td>
<td>Geospatial constraints: cannot surveil beyond Universal Transverse Mercator (UTM). Military secrecy: target and point of origin privacy</td>
</tr>
<tr>
<td></td>
<td>Compliance with regulations</td>
<td>Numerous. There are also standards issues</td>
</tr>
<tr>
<td></td>
<td>Government access to data and freedom of expression concerns</td>
<td>For example, the Google lawsuit over Street View</td>
</tr>
<tr>
<td>Data Provider ↔ Framework Provider</td>
<td>Data-centric security such as identity/policy-based encryption</td>
<td>Policy-based encryption, often dictated by legacy channel capacity/type</td>
</tr>
<tr>
<td></td>
<td>Policy management for access control</td>
<td>Transformations tend to be made within DOD/contractor-devised system schemes</td>
</tr>
<tr>
<td></td>
<td>Computing on the encrypted data: searching/filtering/deduplicate/fully homomorphic encryption</td>
<td>Sometimes performed within vendor-supplied architectures, or by image-processing parallel architectures</td>
</tr>
<tr>
<td></td>
<td>Audits</td>
<td>CSO and Inspector General (IG) audits</td>
</tr>
<tr>
<td>NBDRA Component and Interfaces</td>
<td>Security and Privacy Topic</td>
<td>Use Case Mapping</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Framework Provider</td>
<td>Securing data storage and transaction logs</td>
<td>The usual, plus data center security levels are tightly managed (e.g., field vs. battalion vs. headquarters)</td>
</tr>
<tr>
<td></td>
<td>Key management</td>
<td>CSO—chain of command</td>
</tr>
<tr>
<td></td>
<td>Security best practices for non-relational data stores</td>
<td>Not handled differently at present; this is changing in DOD</td>
</tr>
<tr>
<td></td>
<td>Security against DoS attacks</td>
<td>DOD anti-jamming e-measures</td>
</tr>
<tr>
<td></td>
<td>Data provenance</td>
<td>Must track to sensor point in time configuration and metadata</td>
</tr>
<tr>
<td>Fabric</td>
<td>Analytics for security intelligence</td>
<td>DOD develops specific field of battle security software intelligence—event driven and monitoring—that is often remote</td>
</tr>
<tr>
<td></td>
<td>Event detection</td>
<td>For example, target identification in a video stream infers height of target from shadow. Fuse data from satellite infrared with separate sensor stream</td>
</tr>
<tr>
<td></td>
<td>Forensics</td>
<td>Used for after action review (AAR)—desirable to have full playback of sensor streams</td>
</tr>
</tbody>
</table>

6.9 Education: Common Core Student Performance Reporting

Cradle-to-grave student performance metrics for every student are now possible—at least within the K-12 community, and probably beyond. This could include every test result ever administered.
Table 9: Mapping Common Core K–12 Student Reporting to the Reference Architecture

<table>
<thead>
<tr>
<th>NBDRA Component and Interfaces</th>
<th>Security and Privacy Topic</th>
<th>Use Case Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Provider → Application Provider</td>
<td>End-point input validation</td>
<td>Application-dependent. Spoofing is possible</td>
</tr>
<tr>
<td></td>
<td>Real-time security monitoring</td>
<td>Vendor-specific monitoring of tests, test-takers, administrators, and data</td>
</tr>
<tr>
<td></td>
<td>Data discovery and classification</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>Secure data aggregation</td>
<td>Typical: Classroom-level</td>
</tr>
<tr>
<td>Application Provider → Data Consumer</td>
<td>Privacy-preserving data analytics</td>
<td>Various: For example, teacher-level analytics across all same-grade classrooms</td>
</tr>
<tr>
<td></td>
<td>Compliance with regulations</td>
<td>Parent, student, and taxpayer disclosure and privacy rules apply</td>
</tr>
<tr>
<td></td>
<td>Government access to data and freedom of expression concerns</td>
<td>Yes. May be required for grants, funding, performance metrics for teachers, administrators, and districts</td>
</tr>
<tr>
<td>Data Provider ↔ Framework Provider</td>
<td>Data-centric security such as identity/policy-based encryption</td>
<td>Support both individual access (student) and partitioned aggregate</td>
</tr>
<tr>
<td></td>
<td>Policy management for access control</td>
<td>Vendor (e.g., Pearson) controls, state-level policies, federal-level policies; probably 20-50 different roles are spelled out at present</td>
</tr>
<tr>
<td></td>
<td>Computing on the encrypted data: searching/filtered/deduplicate/fully homomorphic</td>
<td>Proposed [55]</td>
</tr>
<tr>
<td>NBDRA Component and Interfaces</td>
<td>Security and Privacy Topic</td>
<td>Use Case Mapping</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td>encryption</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Audits</td>
<td>Support both internal and third-party audits by unions, state agencies, responses to subpoenas</td>
</tr>
<tr>
<td>Framework Provider</td>
<td>Securing data storage and transaction logs</td>
<td>Large enterprise security, transaction level controls—classroom to the federal government</td>
</tr>
<tr>
<td></td>
<td>Key management</td>
<td>CSOs from the classroom level to the national level</td>
</tr>
<tr>
<td></td>
<td>Security best practices for non-relational data stores</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>Security against DDoS attacks</td>
<td>Standard</td>
</tr>
<tr>
<td></td>
<td>Data provenance</td>
<td>Traceability to measurement event requires capturing tests at a point in time, which may itself require a Big Data platform</td>
</tr>
<tr>
<td>Fabric</td>
<td>Analytics for security intelligence</td>
<td>Various commercial security applications</td>
</tr>
<tr>
<td></td>
<td>Event detection</td>
<td>Various commercial security applications</td>
</tr>
<tr>
<td></td>
<td>Forensics</td>
<td>Various commercial security applications</td>
</tr>
</tbody>
</table>

### 6.10 Sensor Data Storage and Analytics

Mapping of sensor data storage and analytics is under development and will be included in future versions of this
6.11 Cargo Shipping

This use case provides an overview of a Big Data application related to the shipping industry for which standards may emerge in the near future.

Table 10: Mapping Cargo Shipping to the Reference Architecture

<table>
<thead>
<tr>
<th>NBDRA Component and Interfaces</th>
<th>Security and Privacy Topic</th>
<th>Use Case Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Provider → Application Provider</td>
<td>End-point input validation</td>
<td>Ensuring integrity of data collected from sensors</td>
</tr>
<tr>
<td></td>
<td>Real-time security monitoring</td>
<td>Sensors can detect abnormal temperature/environmental conditions for packages with special requirements. They can also detect leaks/radiation</td>
</tr>
<tr>
<td></td>
<td>Data discovery and classification</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>Secure data aggregation</td>
<td>Securely aggregating data from sensors</td>
</tr>
<tr>
<td>Application Provider → Data Consumer</td>
<td>Privacy-preserving data analytics</td>
<td>Sensor-collected data can be private and can reveal information about the package and geographical information. The revealing of such information needs to preserve privacy</td>
</tr>
<tr>
<td></td>
<td>Compliance with regulations</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>Government access to data and freedom of expression concerns</td>
<td>The U.S. Department of Homeland Security may monitor suspicious packages moving into/out of the country</td>
</tr>
<tr>
<td>Data Provider ↔ Data Consumer</td>
<td>Data-centric security such as</td>
<td>---</td>
</tr>
<tr>
<td>NBDRA Component and Interfaces</td>
<td>Security and Privacy Topic</td>
<td>Use Case Mapping</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Framework Provider</td>
<td>identity/policy-based encryption</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Policy management for access control</td>
<td>Private, sensitive sensor data and package data should only be available to authorized individuals. Third-party commercial offerings may implement low-level access to the data</td>
</tr>
<tr>
<td></td>
<td>Computing on the encrypted data: searching/filtering/deduplicate/fully homomorphic encryption</td>
<td>See above section on “Transformation”</td>
</tr>
<tr>
<td></td>
<td>Audits</td>
<td>---</td>
</tr>
<tr>
<td>Framework Provider</td>
<td>Securing data storage and transaction logs</td>
<td>Logging sensor data is essential for tracking packages. Sensor data at rest should be kept in secure data stores</td>
</tr>
<tr>
<td></td>
<td>Key management</td>
<td>For encrypted data</td>
</tr>
<tr>
<td></td>
<td>Security best practices for non-relational data stores</td>
<td>The diversity of sensor types and data types may necessitate the use of non-relational data stores</td>
</tr>
<tr>
<td></td>
<td>Security against DoS attacks</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>Data provenance</td>
<td>Metadata should be cryptographically attached to the collected data so that the integrity of origin and progress can be assured. Complete preservation of provenance will sometimes mandate a separate Big Data application</td>
</tr>
</tbody>
</table>
## Appendix A: Candidate Security and Privacy Topics for Big Data Adaptation

The following set of topics was initially adapted from the scope of the CSA BDWG charter and organized according to the classification in CSA BDWG’s *Top 10 Challenges in Big Data Security and Privacy*. Security and privacy concerns are classified in four categories:

- **Infrastructure Security**
- **Data Privacy**
- **Data Management**
- **Integrity and Reactive Security**

NBD-PWG Security and Privacy Subgroup identified the Big Data topics below for possible inspection during the preparation of Version 2 of this document. A complete rework of these topics is beyond the scope of this document. This material may be refined and organized if needed in future versions of this document.

### Infrastructure Security

- Review of technologies and frameworks that have been primarily developed for performance, scalability, and availability, massively parallel processing (MPP) databases, and others.
- High-availability
  - Use of Big Data to enhance defenses against DDoS attacks.
- DevOps Security

### Data Privacy

<table>
<thead>
<tr>
<th>NBDRA Component and Interfaces</th>
<th>Security and Privacy Topic</th>
<th>Use Case Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fabric</td>
<td>Analytics for security intelligence</td>
<td>Anomalies in sensor data can indicate tampering/fraudulent insertion of data traffic</td>
</tr>
<tr>
<td></td>
<td>Event detection</td>
<td>Abnormal events such as cargo moving out of the way or being stationary for unwarranted periods can be detected</td>
</tr>
<tr>
<td></td>
<td>Forensics</td>
<td>Analysis of logged data can reveal details of incidents after they occur</td>
</tr>
</tbody>
</table>
• System architects should consider the impact of the social data revolution on the security and privacy of Big Data implementations. Some systems not designed to include social data could be connected to social data systems by third parties, or by other project sponsors within an organization.

  ◦ Unknowns of innovation: When a perpetrator, abuser, or stalker misuses technology to target and harm a victim, there are various criminal and civil charges that might be applied to ensure accountability and promote victim safety. A number of U.S. federal and state, territory, or tribal laws might apply. To support the safety and privacy of victims, it is important to take technology-facilitated abuse and stalking seriously. This includes assessing all ways that technology is being misused to perpetrate harm, and considering all charges that could or should be applied.

  ◦ Identify laws that address violence and abuse
    • Stalking and cyberstalking (e.g., felony menacing by, via electronic surveillance)
    • Harassment, threats, and assault
    • Domestic violence, dating violence, sexual violence, and sexual exploitation
    • Sexting and child pornography: electronic transmission of harmful information to minors, providing obscene material to a minor, inappropriate images of minors, and lascivious intent
    • Bullying and cyberbullying
    • Child abuse
  
  ◦ Identify possible criminal or civil laws applicable related to Big Data technology, communications, privacy, and confidentiality
    • Unauthorized access, unauthorized recording/taping, illegal interception of electronic communications, illegal monitoring of communications, surveillance, eavesdropping, wiretapping, and unlawful party to call
    • Computer and internet crimes: fraud and network intrusion
    • Identity theft, impersonation, and pretexting
    • Financial fraud and telecommunications fraud
    • Privacy violations
    • Consumer protection laws
    • Violation of no contact, protection, and restraining orders
    • Technology misuse: Defamatory libel, slander, economic or reputational harms, and privacy torts
    • Burglary, criminal trespass, reckless endangerment, disorderly conduct, mischief, and obstruction of justice

• Data-centric security may be needed to protect certain types of data no matter where it is stored or accessed (e.g., attribute-based encryption and format-preserving encryption). There are domain-specific particulars that should be considered when addressing encryption tools available to system users.

• Big data privacy and governance
  ◦ Data discovery and classification
  ◦ Policy management for accessing and controlling Big Data
    • Are new policy language frameworks specific to Big Data architectures needed?
  ◦ Data masking technologies: Anonymization, rounding, truncation, hashing, and differential privacy
    • It is important to consider how these approaches degrade performance or hinder delivery all together—for Big Data systems in particular. Often these solutions are proposed and then cause an outage at the time of the release, forcing the removal of the option.
  ◦ Compliance with regulations such as the Health Insurance Portability and Accountability Act (HIPAA), European Union (EU) data protection regulations, Asia-Pacific Economic Cooperation (APEC) Cross-Border Privacy Rules (CBPR) requirements, and country-specific regulations
- Regional data stores enable regional laws to be enforced
  - Cybersecurity Executive Order 1998—assumed data and information would remain within the region
- People-centered design makes the assumption that private-sector stakeholders are operating ethically and respecting the freedoms and liberties of all Americans.
  - Litigation, including class action suits, could follow increased threats to Big Data security, when compared to other systems
    - People before profit must be revisited to understand the large number of Executive Orders overlooked
    - People before profit must be revisited to understand the large number of domestic laws overlooked
- Indigenous and aboriginal people and the privacy of all associated vectors and variables must be excluded from any Big Data store in any case in which a person must opt in
  - All tribal land is an exclusion from any image capture and video streaming or capture
  - Human rights
- Government access to data and freedom of expression concerns
  - Polls show that U.S. citizens are less concerned about the loss of privacy than Europeans are, but both are concerned about data misuse and their inability to govern private- and public-sector use
- Potentially unintended/unwanted consequences or uses
  - Appropriate uses of data collected or data aggregation and problem management capabilities must be enabled
  - Mechanisms for the appropriate secondary or subsequent data uses, such as filtered upon entry processed and presented in the inbound framework
- Issues surrounding permission to collect data, consent, and privacy
  - Differences between where the privacy settings are applied in web services and the user’s perception of the privacy setting application
  - Permission based on clear language and not forced by preventing users to access their online services
  - People do not believe the government would allow businesses to take advantage of their rights
- Data deletion: Responsibility to purge data based on certain criteria and/or events
  - Examples include legal rulings that affect an external data source. For example, if Facebook were to lose a legal challenge and required to purge its databases of certain private information. Is there then a responsibility for downstream data stores to follow suit and purge their copies of the same data? The provider, producer, collector or social media supplier, or host absolutely must inform and remove all versions. Enforcement? Verification?
- Computing on encrypted data
  - Deduplication of encrypted data
  - Searching and reporting on the encrypted data
  - Fully homomorphic encryption
  - Anonymization of data (no linking fields to reverse identify)
  - De-identification of data (individual centric)
  - Non-identifying data (individual and context centric)
- Secure data aggregation
- Data loss prevention
Fault tolerance—recovery for zero data loss
- Aggregation in end-to-end scale of resilience, record, and operational scope for integrity and privacy in a secure or better risk management strategy
- Fewer applications will require fault tolerance with clear distinction around risk and scope of the risk

Data Management
- Securing data stores
  - Communication protocols
- Database links
- Access control list (ACL)
- Application programming interface (API)
- Channel segmentation
  - Attack surface reduction
- Key management and ownership of data
  - Providing full control of the keys to the data owner
  - Transparency of data life cycle process: Acquisition, uses, transfers, dissemination, and destruction
- Maps to aid non-technical people determine who is using their data and how their data is being used, including custody over time

Integrity and Reactive Security
- Big Data analytics for security intelligence (identifying malicious activity) and situational awareness (understanding the health of the system)
  - Large-scale analytics
- Need assessment of the public sector
  - Streaming data analytics
- This could require, for example, segregated virtual machines and secure channels
- This is a low-level requirement
- Roadmap
- Priority of security and return on investment must be done to move to this degree of maturity
- Event detection
  - Respond to data risk events trigger by application-specific analysis of user and system behavior patterns
  - Data-driven abuse detection
- Forensics
- Security of analytics results

Appendix B: Internal Security Considerations within Cloud Ecosystems

Many Big Data systems will be designed using cloud architectures. Any strategy to implement a mature security and privacy framework within a Big Data cloud ecosystem enterprise architecture must address the complexities associated
with cloud-specific security requirements triggered by the cloud characteristics. These requirements could include the following:

- Broad network access
- Decreased visibility and control by consumer
- Dynamic system boundaries and com mingled roles/responsibilities between consumers and providers
- Multi-tenancy
- Data residency
- Measured service
- Order-of-magnitude increases in scale (on demand), dynamics (elasticity and cost optimization), and complexity (automation and virtualization)

These cloud computing characteristics often present different security risks to an agency than the traditional information technology solutions, thereby altering the agency’s security posture.

To preserve the security-level after the migration of their data to the cloud, organizations need to identify all cloud-specific, risk-adjusted security controls or components in advance. The organizations must also request from the cloud service providers, through contractual means and service-level agreements, to have all identified security components and controls fully and accurately implemented.

The complexity of multiple interdependencies is best illustrated by Figure B-1.

**Figure B-1: Composite Cloud Ecosystem Security Architecture[57]**

When unraveling the complexity of multiple interdependencies, it is important to note that enterprise-wide access controls fall within the purview of a well thought out Big Data and cloud ecosystem risk management strategy for end-to-end enterprise access control and security (AC&S), via the following five constructs:

1. Categorize the data value and criticality of information systems and the data custodian’s duties and responsibilities
to the organization, demonstrated by the data custodian’s choice of either a discretionary access control policy or a mandatory access control policy that is more restrictive. The choice is determined by addressing the specific organizational requirements, such as, but not limited to the following:

1. GRC
2. Directives, policy guidelines, strategic goals and objectives, information security requirements, priorities, and resources available (filling in any gaps)
3. Select the appropriate level of security controls required to protect data and to defend information systems
4. Implement access security controls and modify them upon analysis assessments
5. Authorize appropriate information systems
6. Monitor access security controls at a minimum of once a year

To meet GRC and confidentiality, integrity, and availability regulatory obligations required from the responsible data custodians—which are directly tied to demonstrating a valid, current, and up-to-date AC&S policy—one of the better strategies is to implement a layered approach to AC&S, comprised of multiple access control gates, including, but not limited to, the following infrastructure AC&S via:

- Physical security/facility security, equipment location, power redundancy, barriers, security patrols, electronic surveillance, and physical authentication
- Information Security and residual risk management
- Human resources (HR security, including, but not limited to, employee codes of conduct, roles and responsibilities, job descriptions, and employee terminations
- Database, end point, and cloud monitoring
- Authentication services management/monitoring
- Privilege usage management/monitoring
- Identify management/monitoring
- Security management/monitoring
- Asset management/monitoring

The following section revisits the traditional access control framework. The traditional framework identifies a standard set of attack surfaces, roles, and tradeoffs. These principles appear in some existing best practices guidelines. For instance, they are an important part of the Certified Information Systems Security Professional (CISSP) body of knowledge. This framework for Big Data may be adopted during the future work of the NBD-PWG. CISSP is a professional computer security certification administered by (ISC)2. (https://www.isc2.org/cissp/default.aspx)

Access Control

Access control is one of the most important areas of Big Data. There are multiple factors, such as mandates, policies, and laws that govern the access of data. One overarching rule is that the highest classification of any data element or
string governs the protection of the data. In addition, access should only be granted on a need-to-know/-use basis that is reviewed periodically in order to control the access.

Access control for Big Data covers more than accessing data. Data can be accessed via multiple channels, networks, and platforms—including laptops, cell phones, smart phones, tablets, and even fax machines—that are connected to internal networks, mobile devices, the internet, or all of the above. With this reality in mind, the same data may be accessed by a user, administrator, another system, etc., and it may be accessed via a remote connection/access point as well as internally. Therefore, visibility as to who is accessing the data is critical in protecting the data. The trade-offs between strict data access control versus conducting business requires answers to questions such as the following.

- How important/critical is the data to the lifeblood and sustainability of the organization?
- What is the organization responsible for (e.g., all nodes, components, boxes, and machines within the Big Data/cloud ecosystem)?
- Where are the resources and data located?
- Who should have access to the resources and data?
- Have GRC considerations been given due attention?

Very restrictive measures to control accounts are difficult to implement, so this strategy can be considered impractical in most cases. However, there are best practices, such as protection based on classification of the data, least privilege\[ii\], and separation of duties that can help reduce the risks.

The following measures are often included in Best Practices lists for security and privacy. Some, and perhaps all, of the measure require adaptation or expansion for Big Data systems.

- Least privilege—access to data within a Big Data/cloud ecosystem environment should be based on providing an individual with the minimum access rights and privileges to perform his/her job
- If one of the data elements is protected because of its classification (e.g., PII, HIPAA, payment card industry [PCI]), then all of the data that it is sent with it inherits that classification, retaining the original data’s security classification. If the data is joined to and/or associated with other data that may cause a privacy issue, then all data should be protected. This requires due diligence on the part of the data custodian(s) to ensure that this secure and protected state remains throughout the entire end-to-end data flow. Variations on this theme may be required for domain-specific combinations of public and private data hosted by Big Data applications.
- If data is accessed from, transferred to, or transmitted to the cloud, internet, or another external entity, then the data should be protected based on its classification.
- There should be an indicator/disclaimer on the display of the user if private or sensitive data is being accessed or viewed. Openness, trust, and transparency considerations may require more specific actions, depending on GRC or other broad considerations of how the Big Data system is being used
- All system roles (“accounts”) should be subjected to periodic meaningful audits to check that they are still required
· All accounts (except for system-related accounts) that have not been used within 180 days should be deactivated.

· Access to PII data should be logged. Role-based access to Big Data should be enforced. Each role should be assigned the fewest privileges needed to perform the functions of that role.

  · Roles should be reviewed periodically to check that they are still valid and that the accounts assigned to them are still appropriate.

User Access Controls

· Each user should have his or her personal account. Shared accounts should not be the default practice in most settings.

  · A user role should match the system capabilities for which it was intended. For example, a user account intended only for information access or to manage an Orchestrator should not be used as an administrative account or to run unrelated production jobs.

System Access Controls

· There should not be shared accounts in cases of system-to-system access. "Meta-accounts" that operate across systems may be an emerging Big Data concern.

· Access for a system that contains Big Data needs to be approved by the data owner or his/her representative. The representative should not be infrastructure support personnel (e.g., a system administrator), because that may cause a separation of duties issue.

  · Ideally, the same type of data stored on different systems should use the same classifications and rules for access controls to provide the same level of protection. In practice, Big Data systems may not follow this practice, and different techniques may be needed to map roles across related but dissimilar components or even across Big Data systems.

Administrative Account Controls

· System administrators should maintain a separate user account that is not used for administrative purposes. In addition, an administrative account should not be used as a user account.

  · The same administrative account should not be used for access to the production and non-production (e.g., test, development, and quality assurance) systems.

Appendix C: Big Data Actors and Roles: Adaptation to Big Data Scenarios

Service-oriented architectures (SOA) were a widely discussed paradigm through the early 2000’s. While the concept is employed less often, SOA has influenced systems analysis processes, and perhaps to a lesser extent, systems design. As noted by Patig and Lopez-Sanz et al., actors and roles were incorporated into Unified Modeling Language so that these concepts could be represented within and well as across services. [59] [60] Big Data calls for further adaptation of these concepts. While actor/role concepts have not been fully integrated into the proposed security fabric, the Subgroup felt it important to emphasize to Big Data system designers how these concepts may need to be adapted from legacy systems.
and SOA usage.

Similar adaptations from Business Process Execution Language, Business Process Model and Notation frameworks offer additional patterns for Big Data security and privacy fabric standards. Ardagna et al. [61] suggest how adaptations might proceed from SOA, but Big Data systems offer somewhat different challenges.

Big Data systems can comprise simple machine-to-machine actors, or complex combinations of persons and machines that are systems of systems.

A common meaning of actor assigns roles to a person in a system. From a citizen’s perspective, a person can have relationships with many applications and sources of information in a Big Data system.

The following list describes a number of roles as well as how roles can shift over time. For some systems, roles are only valid for a specified point in time. Reconsidering temporal aspects of actor security is salient for Big Data systems, as some will be architected without explicit archive or deletion policies.

- A retail organization refers to a person as a consumer or prospect before a purchase; afterwards, the consumer becomes a customer
- A person has a customer relationship with a financial organization for banking services
- A person may have a car loan with a different organization or the same financial institution
- A person may have a home loan with a different bank or the same bank
- A person may be "the insured" on health, life, auto, homeowners, or renters insurance
- A person may be the beneficiary or future insured person by a payroll deduction in the private sector, or via the employment development department in the public sector
- A person may have attended one or more public or private schools
- A person may be an employee, temporary worker, contractor, or third-party employee for one or more private or public enterprises
- A person may be underage and have special legal or other protections
- One or more of these roles may apply concurrently

For each of these roles, system owners should ask themselves whether users could achieve the following:

- Identify which systems their PII has entered
- Identify how, when, and what type of de-identification process was applied
- Verify integrity of their own data and correct errors, omissions, and inaccuracies
- Request to have information purged and have an automated mechanism to report and verify removal
- Participate in multilevel opt-out systems, such as will occur when Big Data systems are federated
- Verify that data has not crossed regulatory (e.g., age-related), governmental (e.g., a state or nation), or expired ("I am no longer a customer") boundaries

Opt-In Revisited

While standards organizations grapple with frameworks such as the one developed here, and until an individual's privacy and security can be fully protected using such a framework, some observers believe that the following two
simple “protocols” ought to govern PII Big Data collection in the meantime.

**Suggested Protocol one:** An individual can only decide to opt-in for inclusion of their personal data manually, and it is a decision that they can revoke at any time.

**Suggested Protocol number two:** The individual's privacy and security opt-in process should enable each individual to modify their choice at any time, to access and review log files and reports and establish a self-destruct timeline (similar to the EU’s “right to be forgotten”.)

---

**Appendix D: Acronyms**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC&amp;S</td>
<td>access control and security</td>
</tr>
<tr>
<td>ACLs</td>
<td>Access Control Lists</td>
</tr>
<tr>
<td>AuthN/AuthZ</td>
<td>Authentication/Authorization</td>
</tr>
<tr>
<td>BAA</td>
<td>business associate agreement</td>
</tr>
<tr>
<td>CDC</td>
<td>U.S. Centers for Disease Control and Prevention</td>
</tr>
<tr>
<td>CEP</td>
<td>complex event processing</td>
</tr>
<tr>
<td>CIA</td>
<td>U.S. Central Intelligence Agency</td>
</tr>
<tr>
<td>CIICF</td>
<td>Critical Infrastructure Cybersecurity Framework</td>
</tr>
<tr>
<td>CINDER</td>
<td>DARPA Cyber-Insider Threat</td>
</tr>
<tr>
<td>CMS</td>
<td>U.S. Centers for Medicare &amp; Medicaid Services</td>
</tr>
<tr>
<td>CoP</td>
<td>communities of practice</td>
</tr>
<tr>
<td>CSA</td>
<td>Cloud Security Alliance</td>
</tr>
<tr>
<td>CSA BDWG</td>
<td>Cloud Security Alliance Big Data Working Group</td>
</tr>
<tr>
<td>CSP</td>
<td>Cloud Service Provider</td>
</tr>
<tr>
<td>DARPA</td>
<td>Defense Advanced Research Projects Agency’s</td>
</tr>
<tr>
<td>DDoS</td>
<td>distributed denial of Service</td>
</tr>
<tr>
<td>DOD</td>
<td>U.S. Department of Defense</td>
</tr>
<tr>
<td>DoS</td>
<td>denial of service</td>
</tr>
<tr>
<td>DRM</td>
<td>digital rights management</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>EFPIA</td>
<td>European Federation of Pharmaceutical Industries and Associations</td>
</tr>
<tr>
<td>EHRs</td>
<td>electronic health records</td>
</tr>
<tr>
<td>EU</td>
<td>European Union</td>
</tr>
<tr>
<td>FBI</td>
<td>U.S. Federal Bureau of Investigation</td>
</tr>
<tr>
<td>FTC</td>
<td>Federal Trade Commission</td>
</tr>
<tr>
<td>GPS</td>
<td>global positioning system</td>
</tr>
<tr>
<td>GRC</td>
<td>governance, risk management, and compliance</td>
</tr>
<tr>
<td>HIEs</td>
<td>Health Information Exchanges</td>
</tr>
<tr>
<td>HIPAA</td>
<td>Health Insurance Portability and Accountability Act</td>
</tr>
<tr>
<td>HITECH Act</td>
<td>Health Information Technology for Economic and Clinical Health Act</td>
</tr>
<tr>
<td>HR</td>
<td>human resources</td>
</tr>
<tr>
<td>IdP</td>
<td>Identity Provider</td>
</tr>
<tr>
<td>IoT</td>
<td>internet of things</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IT</td>
<td>information technology</td>
</tr>
<tr>
<td>LHNCBC</td>
<td>Lister Hill National Center for Biomedical Communications</td>
</tr>
<tr>
<td>M2M</td>
<td>machine to machine</td>
</tr>
<tr>
<td>MAC</td>
<td>media access control</td>
</tr>
<tr>
<td>NBD-PWG</td>
<td>NIST Big Data Public Working Group</td>
</tr>
<tr>
<td>NBDRA</td>
<td>NIST Big Data Reference Architecture</td>
</tr>
<tr>
<td>NBDRA-SP</td>
<td>NIST Big Data Security and Privacy Reference Architecture</td>
</tr>
<tr>
<td>NIEM</td>
<td>National Information Exchange Model</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>NSA</td>
<td>U.S. National Security Agency</td>
</tr>
<tr>
<td>OSS</td>
<td>operations systems support</td>
</tr>
</tbody>
</table>
PaaS  platform as a service

PHI  protected health information

PII  personally identifiable information

PKI  public key infrastructure

SAML  Security Assertion Markup Language

SIEM  Security Information and Event Management

SKUs  stock keeping units

SLAs  Service Level Agreements

STS  Security Token Service

TLS  Transport Layer Security

VM  virtual machine

VPN  virtual private network

WS  web services

XACML  eXtensible Access Control Markup Language

---

**Appendix E: References**

**General Resources**


**Document References**


[29]


[30]


[31]


[32]


[33]


[34]


[35]


[36]


[37]


[38]


Architecture White Paper Survey

Source: http://bigdatawg.nist.gov/_uploadfil...656223932.docx (Word)

Cover Page

NIST Special Publication 1500-5

DRAFT NIST Big Data Interoperability Framework:

Volume 5, Architectures White Paper Survey

NIST Big Data Public Working Group

Reference Architecture Subgroup
planning and transition purposes, Federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. All NIST Information Technology Laboratory publications, other than the ones noted above, are available at http://www.nist.gov/publication-portal.cfm.

Public comment period: April 6, 2015 through May 21, 2015

Comments on this publication may be submitted to Wo Chang

National Institute of Standards and Technology

Attn: Wo Chang, Information Technology Laboratory

100 Bureau Drive (Mail Stop 8900) Gaithersburg, MD 20899-8930

Email: SP1500comments@nist.gov

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at NIST promotes the U.S. economy and public welfare by providing technical leadership for the Nation’s measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL’s responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in Federal information systems. This document reports on ITL’s research, guidance, and outreach efforts in Information Technology and its collaborative activities with industry, government, and academic organizations.

Abstract

Big Data is a term used to describe the deluge of data in our networked, digitized, sensor-laden, information-driven world. While great opportunities exist with Big Data, it can overwhelm traditional technical approaches and its growth is outpacing scientific and technological advances in data analytics. To advance progress in Big Data, the NIST Big Data Public Working Group (NBD-PWG) is working to develop consensus on important, fundamental questions related to Big Data. The results are reported in the *NIST Big Data Interoperability Framework* series of volumes. This volume, Volume 5, presents the results of the reference architecture survey. The reviewed reference architectures are described in detail, followed by a summary of the reference architecture comparison.

Keywords

Big Data, Big Data management, Big Data storage, Big Data analytics, application interfaces, Big Data infrastructure, reference architecture, architecture survey
Acknowledgements

This document reflects the contributions and discussions by the membership of the NBD-PWG, co-chaired by Wo Chang of the NIST Information Technology Laboratory, Robert Marcus of ET-Strategies, and Chaitanya Baru, University of California San Diego Supercomputer Center.

The document contains input from members of the NBD-PWG Reference Architecture Subgroup, led by Orit Levin (Microsoft), Don Krapohl (Augmented Intelligence), and James Ketner (AT&T).

NIST SP1500-5, Version 1 has been collaboratively authored by the NBD-PWG. As of the date of this publication, there are over six hundred NBD-PWG participants from industry, academia, and government. Federal agency participants include the National Archives and Records Administration (NARA), National Aeronautics and Space Administration (NASA), National Science Foundation (NSF), and the U.S. Departments of Agriculture, Commerce, Defense, Energy, Health and Human Services, Homeland Security, Transportation, Treasury, and Veterans Affairs.

NIST would like to acknowledge the specific contributions to this volume by the following NBD-PWG members:

- Milind Bhandarkar
  - **EMC/Pivotal**

- Wo Chang
  - **National Institute of Standards and Technology**

- Yuri Demchenko
  - **University of Amsterdam**

- Barry Devlin
  - **9sight Consulting**

- Harry Foxwell
  - **Oracle Press**

- James Kobielus
  - **IBM**

- Orit Levin
  - **Microsoft**

- Robert Marcus
  - **ET-Strategies**

- Tony Middleton
  - **LexisNexis**

- Sanjay Mishra
  - **Sanjay Patil**

- Sanjay Patil
  - **SAP**

The editors for this document were Sanjay Mishra and Wo Chang.
Notice to Readers

NIST is seeking feedback on the proposed working draft of the NIST Big Data Interoperability Framework: Volume 5, Architectures White Paper Survey. Once public comments are received, compiled, and addressed by the NBD-PWG, and reviewed and approved by NIST internal editorial board, Version 1 of this volume will be published as final. Three versions are planned for this volume, with Versions 2 and 3 building on the first. Further explanation of the three planned versions and the information contained therein is included in Section 1.5 of this document.

Please be as specific as possible in any comments or edits to the text. Specific edits include, but are not limited to, changes in the current text, additional text further explaining a topic or explaining a new topic, additional references, or comments about the text, topics, or document organization. These specific edits can be recorded using one of the two following methods.

1. **TRACK CHANGES**: make edits to and comments on the text directly into this Word document using track changes
2. **COMMENT TEMPLATE**: capture specific edits using the Comment Template
   ([http://bigdatawg.nist.gov/_uploadfiles/SP1500-1-to-7_comment_template.docx](http://bigdatawg.nist.gov/_uploadfiles/SP1500-1-to-7_comment_template.docx)), which includes space for Section number, page number, comment, and text edits

Submit the edited file from either method 1 or 2 to SP1500comments@nist.gov with the volume number in the subject line (e.g., Edits for Volume 5.)

Please contact Wo Chang (wchang@nist.gov) with any questions about the feedback submission process.

Big Data professionals continue to be welcome to join the NBD-PWG to help craft the work contained in the volumes of the NIST Big Data Interoperability Framework. Additional information about the NBD-PWG can be found at [http://bigdatawg.nist.gov](http://bigdatawg.nist.gov).

Table of Contents

Executive Summary. viii

1 Introduction. 1
   1.1 Background. 1
   1.2 Scope and Objectives of the Reference Architecture Subgroup. 2
   1.3 Report Production. 3
   1.4 Report Structure. 3
   1.5 Future Work on this Volume. 3

2 Big Data Architecture Proposals Received. 4
   2.1 Bob Marcus. 4
2.1.1 General Architecture Description. 4
2.1.2 Architecture Model 4
2.1.3 Key Components. 5

2.2 Microsoft. 7
2.2.1 General Architecture Description. 7
2.2.2 Architecture Model 8
2.2.3 Key Components. 8

2.3 University of Amsterdam.. 9
2.3.1 General Architecture Description. 9
2.3.2 Architecture Model 10
2.3.3 Key Components. 11

2.4 IBM... 11
2.4.1 General Architecture Description. 11
2.4.2 Architecture Model 12
2.4.3 Key Components. 14

2.5 Oracle. 14
2.5.1 General Architecture Description. 14
2.5.2 Architecture Model 14
2.5.3 Key Components. 15

2.6 Pivotal. 16
2.6.1 General Architecture Description. 16
2.6.2 Architecture Model 16
2.6.3 Key Components. 17

2.7 SAP. 19
2.7.1 General Architecture Description. 19
Figure 1: Components of the High Level Reference Model. 4

Figure 2: Description of the Components of the Low-Level Reference Model. 5

Figure 3: Big Data Ecosystem Reference Architecture. 8

Figure 4: Big Data Architecture Framework. 10

Figure 5: IBM Big Data Platform. 12

Figure 6: High level, Conceptual View of the Information Management Ecosystem. 15

Figure 7: Oracle Big Data Reference Architecture. 15

Figure 8: Pivotal Architecture Model. 16

Figure 9: Pivotal Data Fabric and Analytics. 17

Figure 10: SAP Big Data Reference Architecture. 20

Figure 11: 9Sight General Architecture. 21

Figure 12: 9Sight Architecture Model. 22

Figure 13: Lexis Nexis General Architecture. 23

Figure 14: Lexis Nexis High Performance Computing Cluster. 24

Figure 15: Big Data Layered Architecture. 28

Figure 16: Data Discovery and Exploration. 30

Figure 17(a): Stacked View of Surveyed Architecture. 37

Figure 17(b): Stacked View of Surveyed Architecture (continued) 38

Figure 17(c): Stacked View of Surveyed Architecture (continued) 39

Figure 18: Big Data Reference Architecture. 41

Tables

Table 1: Databases and Interfaces in the Layered Architecture from Bob Marcus. 26

Table 2: Microsoft Data Transformation Steps. 28
Executive Summary

This document, NIST Big Data Interoperability Framework: Volume 5, Architectures White Paper Survey, was prepared by the NIST Big Data Public Working Group (NBD-PWG) Reference Architecture Subgroup to facilitate understanding of the operational intricacies in Big Data and to serve as a tool for developing system-specific architectures using a common reference framework. The Subgroup surveyed currently published Big Data platforms by leading companies or individuals supporting the Big Data framework and analyzed the material. This effort revealed a remarkable consistency of Big Data architecture. The most common themes occurring across the architectures surveyed are outlined below.

Big Data Management

- Structured, semi-structured, and unstructured data
- Velocity, variety, volume, and variability
- SQL and NoSQL
- Distributed file system

Big Data Analytics

- Descriptive, predictive, and spatial
- Real-time
- Interactive
- Batch analytics
- Reporting
- Dashboard

Big Data Infrastructure

- In-memory data grids
- Operational database
- Analytic database
- Relational database
- Flat files
- Content management system
The NIST Big Data Interoperability Framework consists of seven volumes, each of which addresses a specific key topic, resulting from the work of the NBD-PWG. The seven volumes are as follows:

- Volume 1, Definitions
- Volume 2, Taxonomies
- Volume 3, Use Cases and General Requirements
- Volume 4, Security and Privacy
- Volume 5, Architectures White Paper Survey
- Volume 6, Reference Architecture
- Volume 7, Standards Roadmap

The NIST Big Data Interoperability Framework will be released in three versions, which correspond to the three stages of the NBD-PWG work. The three stages aim to achieve the following:

Stage 1: Identify the high-level NIST Big Data Reference Architecture (NBDRA) key components, which are technology, infrastructure, and vendor agnostic

Stage 2: Define general interfaces between the NBDRA components

Stage 3: Validate the NBDRA by building Big Data general applications through the general interfaces

Potential areas of future work for the Subgroup during stage 2 are highlighted in Section 1.5 of this volume. The current effort documented in this volume reflects concepts developed within the rapidly evolving field of Big Data.

1 Introduction

1.1 Background

There is broad agreement among commercial, academic, and government leaders about the remarkable potential of Big Data to spark innovation, fuel commerce, and drive progress. Big Data is the common term used to describe the deluge of data in today’s networked, digitized, sensor-laden, and information-driven world. The availability of vast data resources carries the potential to answer questions previously out of reach, including the following:

- How can a potential pandemic reliably be detected early enough to intervene?
- Can new materials with advanced properties be predicted before these materials have ever been synthesized?
- How can the current advantage of the attacker over the defender in guarding against cyber-security threats be reversed?

There is also broad agreement on the ability of Big Data to overwhelm traditional approaches. The growth rates for data...
volumes, speeds, and complexity are outpacing scientific and technological advances in data analytics, management, transport, and data user spheres.

Despite widespread agreement on the inherent opportunities and current limitations of Big Data, a lack of consensus on some important, fundamental questions continues to confuse potential users and stymie progress. These questions include the following:

- What attributes define Big Data solutions?
- How is Big Data different from traditional data environments and related applications?
- What are the essential characteristics of Big Data environments?
- How do these environments integrate with currently deployed architectures?
- What are the central scientific, technological, and standardization challenges that need to be addressed to accelerate the deployment of robust Big Data solutions?

Within this context, on March 29, 2012, the White House announced the Big Data Research and Development Initiative. The initiative’s goals include helping to accelerate the pace of discovery in science and engineering, strengthening national security, and transforming teaching and learning by improving the ability to extract knowledge and insights from large and complex collections of digital data.

Six federal departments and their agencies announced more than $200 million in commitments spread across more than 80 projects, which aim to significantly improve the tools and techniques needed to access, organize, and draw conclusions from huge volumes of digital data. The initiative also challenged industry, research universities, and nonprofits to join with the federal government to make the most of the opportunities created by Big Data.

Motivated by the White House initiative and public suggestions, the National Institute of Standards and Technology (NIST) has accepted the challenge to stimulate collaboration among industry professionals to further the secure and effective adoption of Big Data. As one result of NIST’s Cloud and Big Data Forum held on January 15–17, 2013, there was strong encouragement for NIST to create a public working group for the development of a Big Data Interoperability Framework. Forum participants noted that this framework should define and prioritize Big Data requirements, including interoperability, portability, reusability, extensibility, data usage, analytics, and technology infrastructure. In doing so, the framework would accelerate the adoption of the most secure and effective Big Data techniques and technology.

On June 19, 2013, the NIST Big Data Public Working Group (NBD-PWG) was launched with extensive participation by industry, academia, and government from across the nation. The scope of the NBD-PWG involves forming a community of interests from all sectors—including industry, academia, and government—with the goal of developing consensus on definitions, taxonomies, secure reference architectures, security and privacy, and a standards roadmap. Such a consensus would create a vendor-neutral, technology- and infrastructure-independent framework that would enable Big Data stakeholders to identify and use the best analytics tools for their processing and visualization requirements on the most suitable computing platform and cluster, while also allowing value-added from Big Data service providers.
The *NIST Big Data Interoperability Framework* consists of seven volumes, each of which addresses a specific key topic, resulting from the work of the NBD-PWG. The seven volumes are as follows:

- Volume 1, Definitions
- Volume 2, Taxonomies
- Volume 3, Use Cases and General Requirements
- Volume 4, Security and Privacy
- Volume 5, Architectures White Paper Survey
- Volume 6, Reference Architecture
- Volume 7, Standards Roadmap

The *NIST Big Data Interoperability Framework* will be released in three versions, which correspond to the three stages of the NBD-PWG work. The three stages aim to achieve the following:

Stage 1: Identify the high-level NIST Big Data Reference Architecture (NBDRA) key components, which are technology, infrastructure, and vendor agnostic

Stage 2: Define general interfaces between the NBDRA components

Stage 3: Validate the NBDRA by building Big Data general applications through the general interfaces

Potential areas of future work for the Subgroup during stage 2 are highlighted in Section 1.5 of this volume. The current effort documented in this volume reflects concepts developed within the rapidly evolving field of Big Data.

### 1.2 Scope and Objectives of the Reference Architecture Subgroup

Reference architectures provide "an authoritative source of information about a specific subject area that guides and constrains the instantiations of multiple architectures and solutions."[2] Reference architectures generally serve as a reference foundation for solution architectures, and may also be used for comparison and alignment purposes. This volume was prepared by the NBD-PWG Reference Architecture Subgroup. The effort focused on developing an open reference Big Data architecture that achieves the following objectives:

- Provides a common language for the various stakeholders
- Encourages adherence to common standards, specifications, and patterns
- Provides consistent methods for implementation of technology to solve similar problem sets
- Illustrates and improve understanding of the various Big Data components, processes, and systems, in the context of vendor- and technology-agnostic Big Data conceptual model
- Provides a technical reference for U.S. government departments, agencies, and other consumers to understand,
discuss, categorize, and compare Big Data solutions

- Facilitates the analysis of candidate standards for interoperability, portability, reusability, and extendibility

The NBDRA is intended to facilitate the understanding of the operational intricacies in Big Data. It does not represent the system architecture of a specific Big Data system, but rather is a tool for describing, discussing, and developing system-specific architectures using a common framework. The reference architecture achieves this by providing a generic, high-level conceptual model, which serves as an effective tool for discussing the requirements, structures, and operations inherent to Big Data. The model is not tied to any specific vendor products, services, or reference implementation, nor does it define prescriptive solutions for advancing innovation.

The NBDRA does not address the following:

- Detailed specifications for any organizations’ operational systems
- Detailed specifications of information exchanges or services
- Recommendations or standards for integration of infrastructure products

As a precursor to the development of the NBDRA, the NBD-PWG Reference Architecture Subgroup surveyed the currently published Big Data platforms by leading companies supporting the Big Data framework. All the reference architectures provided to the NBD-PWG are listed and the capabilities of each surveyed platform are discussed in this document.

### 1.3 Report Production

A wide spectrum of Big Data architectures were explored and developed from various industries, academic, and government initiatives. The NBD-PWG Reference Architecture Subgroup produced this report through the four steps outlined below.

1. Announced that the NBD-PWG Reference Architecture Subgroup is open to the public to attract and solicit a wide array of subject matter experts and stakeholders in government, industry, and academia
2. Gather publicly available Big Data architectures and materials representing various stakeholders, different data types, and different use cases
3. Examine and analyze the Big Data material to better understand existing concepts, usage, goals, objectives, characteristics, and key elements of the Big Data, and then document the findings using NIST’s Big Data taxonomies model (presented in *NIST Big Data Interoperability Framework: Volume 2, Taxonomies*)
4. Produce this report to document the findings and work of the NBD-PWG Reference Architecture Subgroup

### 1.4 Report Structure

Following the introductory material presented in Section 1, the remainder of this document is organized as follows:

- Section 2 contains the reference architectures submitted to the NBD-PWG
1.5 Future Work on this Volume

As presented in this document, information about existing reference architectures was collected and analyzed by the NBD-PWG. The knowledge gained from the surveyed architectures was used in the NBDRA development. Additional work is not anticipated for the reference architecture survey.

2 Big Data Architecture Proposals Received

To begin the survey, NBD-PWG members were asked to provide information about Big Data architectures. The architectures described in this section were discussed at length by the NBD-PWG. The NBD-PWG Reference Architecture Subgroup sincerely appreciates contributions of the respective Big Data architectures from the following individuals and organizations. This survey would not have been possible without their contributions,

1. Robert Marcus, ET Strategies
2. Orit Levin, Microsoft
3. Yuri Demchenko, University of Amsterdam
4. James Kobielus, IBM
5. Harry Foxwell, Oracle
6. Milind Bhandarkar, EMC/Pivotal
7. Sanjay Patil, SAP
8. Barry Devlin, 9sight Consulting
9. Tony Middleton, LexisNexis

2.1 Bob Marcus

2.1.1 General Architecture Description

The High Level, Layered Reference Model and detailed Lower Level Reference Architecture in this section are designed to support mappings from Big Data use cases, requirements, and technology gaps. The Layered Reference Model is at a similar level to the NBDRA. The Lower Level Reference Architecture (Section 2.1.3) is a detailed drill-down from the High Level, Layered Reference Model (Section 2.1.2).

2.1.2 Architecture Model

The High Level, Layered Reference Model in Figure 1 gives an overview of the key functions of Big Data architectures.
Figure 1: Components of the High Level Reference Model

<table>
<thead>
<tr>
<th>A. External Data Sources and Sinks</th>
</tr>
</thead>
<tbody>
<tr>
<td>B. Stream and ETL Processing</td>
</tr>
<tr>
<td>C. Highly Scalable Foundation</td>
</tr>
<tr>
<td>D. Operational and Analytics Databases</td>
</tr>
<tr>
<td>E. Analytics and Database Interfaces</td>
</tr>
<tr>
<td>F. Applications and User Interfaces</td>
</tr>
<tr>
<td>G. Supporting Services</td>
</tr>
</tbody>
</table>

A. External Data Sources and Sinks

This feature provides external data inputs and output to the internal Big Data components.

B. Stream and Extract, Transform, Load (ETL) Processing

This processing feature filters and transforms data flows between external data resources and internal Big Data systems.

C. Highly Scalable Foundation

Horizontally scalable data stores and processing form the foundation of Big Data architectures.

D. Operational and Analytics Databases

Databases are integrated into the Big Data architecture. These can be horizontally scalable databases or single platform databases with data extracted from the foundational data store.

E. Analytics and Database Interfaces

These are the interfaces to the data stores for queries, updates, and analytics.

F. Applications and User Interfaces

These are the applications (e.g. machine learning) and user interfaces (e.g. visualization) that are built on Big Data components.

G. Supporting Services

These services include the components needed for the implementation and management of robust Big Data systems.
2.1.3 Key Components

The Lower Level Reference Architecture in Figure 2 expands on some of the layers in the High Level Layered Reference Model and shows some of the data flows.

Figure 2: Description of the Components of the Low-Level Reference Model

The Lower Level Reference Architecture (Figure 2) maps to the High Level Layered Reference Model (Figure 1) as outlined below.

A. External Data Sources and Sinks

| 4. Data Sources and Sinks | These components clearly define interfaces to Big Data horizontally scalable internal data stores and applications. |

B. Stream and ETL Processing

| 5. Scalable Stream Processing | This is processing of “data in movement” between data stores. It can be used for filtering, transforming, or routing data. For Big Data streams, the stream processing should be scalable to support distributed and/or pipelined processing. |

C. Highly Scalable Foundation

| 1. Scalable Infrastructure | To support scalable Big Data stores and processing, the infrastructure should be able to support the easy addition of new resources. Possible platforms |
include public and/or private clouds.

<table>
<thead>
<tr>
<th>2. Scalable Data Stores</th>
</tr>
</thead>
<tbody>
<tr>
<td>This is the essence of Big Data architecture. Horizontal scalability using less expensive components can support the unlimited growth of data storage. However, there should be fault tolerance capabilities available to handle component failures.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3. Scalable Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>To take advantage of scalable distributed data stores, the scalable distributed parallel processing should have similar fault tolerance. In general, processing should be configured to minimize unnecessary data movement.</td>
</tr>
</tbody>
</table>

### D. Operational and Analytics Databases

<table>
<thead>
<tr>
<th>6. Analytics Databases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytics databases are generally highly optimized for read-only interactions (e.g. columnar storage, extensive indexing, and denormalization). It is often acceptable for database responses to have high latency (e.g. invoke scalable batch processing over large data sets).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. Operational Databases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation databases generally support efficient write and read operations. NoSQL databases are often used in Big Data architectures in this capacity. Data can be later transformed and loaded into analytic databases to support analytic applications.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8. In Memory Data Grids</th>
</tr>
</thead>
<tbody>
<tr>
<td>These high performance data caches and stores minimize writing to disk. They can be used for large-scale, real-time applications requiring transparent access to data.</td>
</tr>
</tbody>
</table>

### E. Analytics and Database Interfaces

<table>
<thead>
<tr>
<th>9. Batch Analytics and Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>These interfaces use batch scalable processing (e.g. Map-Reduce) to access data in scalable data stores (e.g., Hadoop File System). These interfaces can be SQL-like (e.g., Hive) or programmatic (e.g., Pig).</td>
</tr>
</tbody>
</table>
10. Interactive Analytics and Interfaces

These interfaces directly access data stores to provide interactive responses to end-users. The data stores can be horizontally scalable databases tuned for interactive responses (e.g., HBase) or query languages tuned to data models (e.g., Drill for nested data).

11. Real-Time Analytics and Interfaces

Some applications require real-time responses to events occurring within large data streams (e.g. algorithmic trading). This complex event processing uses machine-based analytics requiring very high performance data access to streams and data stores.

F. Applications and User Interfaces

12. Applications and Visualization

The key new capability available to Big Data analytic applications is the ability to avoid developing complex algorithms by utilizing vast amounts of distributed data (e.g. Google statistical language translation). However, taking advantage of the data available requires new distributed and parallel processing algorithms.

G. Supporting Services

13. Design, Develop, and Deploy Tools

High level tools are limited for the implementation of Big Data applications (e.g. Cascading). This should change to lower the skill levels needed by enterprise and government developers.

14. Security

Current Big Data security and privacy controls are limited (e.g. only Kerberos authentication for Hadoop, Knox). They should be expanded in the future by commercial vendors (e.g. Cloudera Sentry) for enterprise and government applications.

15. Process Management

Commercial vendors are supplying process management tools to augment the initial open source implementations (e.g. Oozie).


Open Source data governance tools are still immature (e.g. Apache Falcon). These will be augmented in the future by commercial vendors.
Open source systems management tools are also immature (e.g. Ambari). Fortunately robust system management tools are commercially available for scalable infrastructure (e.g. cloud-based).

2.2 Microsoft

2.2.1 General Architecture Description

This Big Data ecosystem reference architecture is a high level, data-centric diagram that depicts the Big Data flow and possible data transformations from collection to usage.

2.2.2 Architecture Model

The Big Data ecosystem is comprised of four main components: Sources, Transformation, Infrastructure and Usage, as shown in Figure 3. Security and Management are shown as examples of additional supporting, crosscutting subsystems that provide backdrop services and functionality to the rest of the Big Data ecosystem.

Figure 3: Big Data Ecosystem Reference Architecture

2.2.3 Key Components

Data Sources

Typically, the data behind Big Data is collected for a specific purpose, and is created in a form that supports the known use at the data collection time. Once data is collected, it can be reused for a variety of purposes, some potentially unknown at the collection time. Data sources are classified by four main characteristics that define Big Data (i.e., volume, variety, velocity, and variability)[3], and are independent of the data content or context.
Data Transformation

As data circulates through the ecosystem, it is being processed and transformed in different ways to extract value from the information. For the purpose of defining interoperability surfaces, it is important to identify common transformations that are implemented by independent modules, systems, or deployed as stand-alone services. The transformation functional blocks shown in Figure 3 can be performed by separate systems or organizations, with data moving between those entities, such as the case with the advertising ecosystem. Similar and additional transformational blocks are being used in enterprise data warehouses, but typically they are closely integrated and rely on a common database to exchange the information.

Each transformation function may have its specific pre-processing stage including registration and metadata creation; may use different specialized data infrastructure best fitted for its requirements; and may have its own privacy and other policy considerations. Common data transformations shown in Figure 3 are:

- **Collection:** Data can be collected in different types and forms. At the initial collection stage, sets of data (e.g., data records) from similar sources and of similar structure are collected (and combined) resulting in uniform security considerations, policies, etc. Initial metadata is created (e.g., subjects with keys are identified) to facilitate subsequent aggregation or lookup method(s).

- **Aggregation:** Sets of existing data collections with easily correlated metadata (e.g., identical keys) are aggregated into a larger collection. As a result, the information about each object is enriched or the number of objects in the collection grows. Security considerations and policies concerning the resulting collection are typically similar to the original collections.

- **Matching:** Sets of existing data collections with dissimilar metadata (e.g., keys) are aggregated into a larger collection. For example, in the advertising industry, matching services correlate HTTP cookies’ values with a person’s real name. As a result, the information about each object is enriched. The security considerations and policies concerning the resulting collection are subject to data exchange interfaces design.

- **Data Mining:** According to DBTA[4], “[d]ata mining can be defined as the process of extracting data, analyzing it from many dimensions or perspectives, and then producing a summary of the information in a useful form that identifies relationships within the data. There are two types of data mining: descriptive, which gives information about existing data; and predictive, which makes forecasts based on the data.”

Data Infrastructure:

Big Data infrastructure is a bundle of data storage or database software, servers, storage, and networking used in support of the data transformation functions and for storage of data as needed. Data infrastructure is placed to the right of the data transformation, to emphasize the natural role of data infrastructure in support of data transformations. Note that the horizontal data retrieval and storage paths exist between the two, which are different from the vertical data paths between them and data sources and data usage.

To achieve high efficiencies, data of different volume, variety, and velocity would typically be stored and processed using computing and storage technologies tailored to those characteristics. The choice of processing and storage technology is also dependent on the transformation itself. As a result, often the same data can be transformed (either
sequentially or in parallel) multiple times using independent data infrastructure.

Examples of conditioning include de-identification, sampling, and fuzzing.

Examples of storage and retrieval include NoSQL and SQL Databases with various specialized types of data load and queries.

Data Usage:

The results can be provided in different formats, (e.g., displayed or electronically encoded, with or without metadata, at rest or streamed), different granularity (e.g., as individual records or aggregated), and under different security considerations (e.g., public disclosure vs. internal use).

2.3 University of Amsterdam

2.3.1 General Architecture Description

This Big Data Architecture Framework (BDAF) supports the extended Big Data definition presented in the Systems and Network Engineering (SNE) technical report[5] and reflects the main components and processes in the Big Data Ecosystem (BDE). The BDAF, shown in Figure 4, is comprised of the following five components that address different aspects of the SNE Big Data definition:

- **Data Models, Structures, and Types**: The BDAF should support a variety of data types produced by different data sources. These data must be stored and processed and will, to some extent, define the Big Data infrastructure technologies and solutions.

- **Big Data Management Infrastructure and Services**: The BDAF should support Big Data Lifecycle Management, provenance, curation, and archiving. Big Data Lifecycle Management should support the major data transformations stages: collection, registration, filtering, classification, analysis, modeling, prediction, delivery, presentation, and visualization. Big Data Management capabilities can be partly addressed by defining scientific or business workflows and using corresponding workflow management systems.

- **Big Data Analytics and Tools**: These specifically address required data transformation functionalities and related infrastructure components.

- **Big Data Infrastructure (BDI)**: This component includes storage, computing infrastructure, network infrastructure, sensor networks, and target or actionable devices.

- **Big Data Security**: Security should protect data in rest and in motion, ensure trusted processing environments and reliable BDI operation, provide fine grained access control, and protect users’ personal information.

2.3.2 Architecture Model

Figure 4 illustrates the basic Big Data analytics capabilities as a part of the overall cloud-based BDI.
In addition to the general cloud-based infrastructure services (e.g., storage, computation, infrastructure or VM management), the following specific applications and services will be required to support Big Data and other data centric applications:

- High-Performance Cluster systems
- Hadoop related services and tools; distributed file systems
- General analytics tools/systems: batch, real-time, interactive
- Specialist data analytics tools: logs, events, data mining
- Databases: operational and analytics; in-memory databases; streaming databases; SQL, NoSQL, key-value storage, etc.
- Streaming analytics and ETL processing
- Data reporting and visualization

Big Data analytics platforms should be vertically and horizontally scalable, which can be naturally achieved when using cloud-based platforms and Intercloud integration models and architecture. [6]

### 2.3.3 Key Components

Big Data infrastructure, including the general infrastructure for general data management, is typically cloud-based. Big Data analytics will use the High-Performance Computing (HPC) architectures and technologies, as shown in Figure 4. General BDI includes the following capabilities, services and components to support the whole Big Data lifecycle:

- General cloud-based infrastructure, platform, services and applications to support creation, deployment and operation of Big Data infrastructures and applications (using generic cloud features of provisioning on-demand, scalability, measured services)
Big Data Management services and infrastructure, which includes data backup, replication, curation, provenance
Registries, indexing/search, metadata, ontologies, namespaces
Security infrastructure (access control, policy enforcement, confidentiality, trust, availability, accounting, identity management, privacy)
Collaborative environment infrastructure (groups management) and user-facing capabilities (user portals, identity management/federation)

Big Data Infrastructure should be supported by broad network access and advanced network infrastructure, which will play a key role in distributed heterogeneous BDI integration and reliable operation:

- Network infrastructure interconnects BDI components. These components are typically distributed, increasingly multi-provider BDI components, and may include intra-cloud (intra-provider) and Intercloud network infrastructure. HPC clusters require high-speed network infrastructure with low latency. Intercloud network infrastructure may require dedicated network links and connectivity provisioned on demand.
- FADI is presented in Figure 4 as a separate infrastructure/structural component to reflect its importance, though it can be treated as a part of the general Intercloud infrastructure of the BDI. FADI combines both Intercloud network infrastructure and corresponding federated security infrastructure to support infrastructure components integration and users federation.

Heterogeneous multi-provider cloud services integration is addressed by the Intercloud Architecture Framework (ICAF), and also, especially, the Intercloud Federation Framework (ICFF) being developed by the authors.[7] [8] [9] ICAF provides a common basis for building adaptive and on-demand provisioned multi-provider cloud based services.

FADI is an important component of the overall cloud and Big Data infrastructure that interconnects all the major components and domains in the multi-provider Intercloud infrastructure, including non-cloud and legacy resources. Using a federation model for integrating multi-provider heterogeneous services and resources reflects current practice in building and managing complex infrastructures and allows for inter-organizational resource sharing and identity federation.

2.4 IBM

2.4.1 General Architecture Description

A Big Data platform must support all types of data and be able to run all necessary computations to drive the analytics.

The IBM reference architecture, as shown in Figure 5 outlines the architecture model. Section 2.4.2 provides additional details.
2.4.2 Architecture Model

To achieve these objectives, any Big Data platform should address six key imperatives:

**Data Discovery and Exploration:** The process of data analysis begins with understanding data sources, figuring out what data is available within a particular source, and getting a sense of its quality and its relationship to other data elements. This process, known as data discovery, enables data scientists to create the right analytic model and computational strategy. Traditional approaches required data to be physically moved to a central location before it could be discovered. With Big Data, this approach is too expensive and impractical. To facilitate data discovery and unlock resident value within Big Data, the platform should be able to discover data ‘in place.’ It should be able to support the indexing, searching, and navigation of different sources of Big Data. It should be able to facilitate discovery of a diverse set of data sources, such as databases, flat files, content management systems—pretty much any persistent data store that contains structured, semi structured, or unstructured data. The security profile of the underlying data systems should be strictly adhered to and preserved. These capabilities benefit analysts and data scientists by helping them to quickly incorporate or discover new data sources in their analytic applications.

**Extreme Performance: Run Analytics Closer to the Data:** Traditional architectures decoupled analytical environments from data environments. Analytical software would run on its own infrastructure and retrieve data from back-end data warehouses or other systems to perform complex analytics. The rationale behind this was that data environments were optimized for faster access to data, but not necessarily for advanced mathematical computations. Hence, analytics were treated as a distinct workload that had to be managed in a separate infrastructure. This
architecture was expensive to manage and operate, created data redundancy, and performed poorly with increasing data volumes. The analytic architecture of the future should run both data processing and complex analytics on the same platform. It should deliver petabyte scale performance throughput by seamlessly executing analytic models inside the platform, against the entire data set, without replicating or sampling data. It should enable data scientists to iterate through different models more quickly to facilitate discovery and experimentation with a “best fit” yield.

**Manage and Analyze Unstructured Data:** For a long time, data has been classified on the basis of its type—structured, semi-structured, or unstructured. Existing infrastructures typically have barriers that prevented the seamless correlation and holistic analysis of this data (e.g., independent systems to store and manage these different data types.) Hybrid systems have also emerged, which have often not performed as well as expected because of their inability to manage all data types. However, organizational processes don’t distinguish between data types. To analyze customer support effectiveness, structured information about a customer service representative (CSR) conversation (e.g., call duration, call outcome, customer satisfaction, survey response) is as important as unstructured information gleaned from that conversation (e.g., sentiment, customer feedback, verbally expressed concerns). Effective analysis should factor in all components of an interaction, and analyze them within the same context, regardless of whether the underlying data is structured or not. A game-changing analytics platform should be able to manage, store, and retrieve both unstructured and structured data. It also should provide tools for unstructured data exploration and analysis.

**Analyze Data in Real Time:** Performing analytics on activity as it unfolds presents a huge untapped opportunity for the analytic enterprise. Historically, analytic models and computations ran on data that was stored in databases. This worked well for transpired events from a few minutes, hours, or even days back. These databases relied on disk drives to store and retrieve data. Even the best performing disk drives had unacceptable latencies for reacting to certain events in real time. Enterprises that want to boost their Big Data IQ need the capability to analyze data as it is being generated, and then to take appropriate action. This allows insight to be derived before the data gets stored on physical disks. This type of data is referred to as streaming data, and the resulting analysis is referred to as analytics of data in motion. Depending on the time of day, or other contexts, the volume of the data stream can vary dramatically. For example, consider a stream of data carrying stock trades in an exchange. Depending on trading activity, that stream can quickly swell from 10 to 100 times its normal volume. This implies that a Big Data platform should not only be able to support analytics of data in motion, but also should scale effectively to manage increasing volumes of data streams.

**A Rich Library of Analytical Functions and Tool Sets:** One of the key goals of a Big Data platform should be to reduce the analytic cycle time—the amount of time that it takes to discover and transform data, develop and score models, and analyze and publish results. As noted earlier, when a platform empowers the user to run extremely fast analytics, a foundation is provided on which to support multiple analytic iterations and speed up model development (the snowball gets bigger and rotates faster). Although this is the desired end goal, there should be a focus on improving developer productivity. By making it easy to discover data, develop and deploy models, visualize results, and integrate with front-end applications, the organization can enable practitioners, such as analysts and data scientists, to be more effective in their respective jobs. This concept is referred to as the art of consumability. Most companies do not have hundreds (if not thousands) of developers on hand, who are skilled in new age technologies. Consumability is key to democratizing Big Data across the enterprise. The Big Data platform should be able to flatten the time-to-analysis curve with a rich set of accelerators, libraries of analytic functions, and a tool set that accelerates the development and visualization process. Because analytics is an emerging discipline, it’s not uncommon to find data scientists who have their own preferred mechanisms for creating and visualizing models. They might use packaged applications, emerging
open source libraries, or build the models using procedural languages. Creating a restrictive development environment curtails their productivity. A Big Data platform should support interaction with the most commonly available analytic packages, with deep integration that facilitates pushing computationally intensive activities from those packages, such as model scoring, into the platform. It should have a rich set of "parallelizable" algorithms that have been developed and tested to run on Big Data. It has to have specific capabilities for unstructured data analytics, such as text analytics routines and a framework for developing additional algorithms. It must also provide the ability to visualize and publish results in an intuitive and easy-to-use manner.

**Integrate and Govern All Data Sources:** Recently, the information management community has made enormous progress in developing sound data management principles. These include policies, tools, and technologies for data quality, security, governance, master data management, data integration, and information lifecycle management. They establish veracity and trust in the data, and are extremely critical to the success of any analytics program.

### 2.4.3 Key Components

The technological capabilities of the IBM framework that address these key strategic imperatives are as follows:

- **Tools:** These components support visualization, discovery, application development, and systems management.

- **Accelerators:** This component provides a rich library of analytical functions, schemas, tool sets, and other artifacts for rapid development and delivery of value in Big Data projects.

- **Hadoop:** This component supports managing and analyzing unstructured data. To support this requirement, IBM InfoSphere BigInsights and PureData System for Hadoop support are required.

- **Stream Computing:** This component supports analyzing in-motion data in real time.

- **Data Warehouse:** This component supports business intelligence, advanced analytics, data governance, and master data management on structured data.

- **Information Integration and Governance:** This component supports integration and governance of all data sources. Its capabilities include data integration, data quality, security, lifecycle management, and master data management.

### 2.5 Oracle

#### 2.5.1 General Architecture Description

Oracle’s Reference Architecture for Big Data provides a complete view of related technical capabilities, how they fit together, and how they integrate into the larger information ecosystem. This reference architecture helps to clarify Oracle’s Big Data strategy and to map specific products that support that strategy.

Oracle offers an integrated solution to address enterprise Big Data requirements. Oracle’s Big Data strategy is centered on extending current enterprise information architectures to incorporate Big Data. Big Data technologies, such as Hadoop and Oracle NoSQL database, run alongside Oracle data warehouse solutions to address Big Data requirements.
for acquiring, organizing, and analyzing data in support of critical organizational decision-making.

2.5.2 Architecture Model

Figure 6 shows a high level view of the Information Management ecosystem:

Figure 6: High level, Conceptual View of the Information Management Ecosystem

2.5.3 Key Components

Figure 7 presents the Oracle Big Data reference architecture, detailing infrastructure services, data sources, information provisioning, and information analysis components.

Figure 7: Oracle Big Data Reference Architecture

2.6 Pivotal
2.6.1 General Architecture Description

Pivotal, a spinoff from EMC and VMware, believes that Hadoop Distributed File System (HDFS) has emerged as a standard interface between the data and the analytics layer. Marching on that path of providing a unified platform, Pivotal integrated the Greenplum Database technology to work directly on HDFS. It now has the first standards-compliant interactive SQL query engine, HAWQ, to store and query both structured and semi-structured data on HDFS. Pivotal is also in the late stages of integrating real-time, in-memory analytics platforms with HDFS. Making HDFS a standard interface results in storage optimization—only a single copy of the data is stored—and gives developers freedom to use an appropriate analytics layer for generating insights for data applications.

2.6.2 Architecture Model

Pivotal Big Data architecture is composed of three different layers: infrastructure, data ingestion and analytics, and data-enabled applications. These layers, or fabrics, have to seamlessly integrate to provide a frictionless environment for users of Big Data systems. Figure 8 implicitly corresponds to the three layers. The infrastructure layer is the cloud (i.e., virtual) layer in the grey box. The black colored boxes represent the data ingestion and analytics layer. The PaaS (Platform as a Service) is included as an example of an application layer.

Figure 8: Pivotal Architecture Model

Infrastructure

- Most of the early Big Data platforms instances assumed a networked bare metal infrastructure. Speed to market drove the early adopters to offer this platform as a service on the bare metal platforms. Monitoring and management capabilities were built around managing a bare metal cluster at scale. The numbers of Big Data platform nodes at some of the early adopters are in the many tens of thousands. Custom management tools were developed for implemented limited elasticity by adding nodes, retiring nodes and monitoring and alerting in case of infrastructure failures.

- Since then, virtualization has been widely adopted in enterprises, since it offers enhanced isolation, utilization, elasticity, and management capabilities inherent in most virtualization technologies.

- Customers now expect a platform that will enable them to take advantage of their investments in virtualized infrastructure, whether deployed on premise, such as with VMWare vSphere, or in public clouds, such as Amazon AWS.

Data Storage and Analytics
- Big Data platforms deal with the gravity of data (i.e., difficulty of moving large amounts of data over existing network capabilities) by moving processing very close to the storage tier. As a result, the analytics and data storage tiers were fused together in the Pivotal architecture. Every attempt was made to move analytics closer to the data layer. Now, high-speed data center interconnects (with 10+ Gbps bandwidth) are becoming more affordable, making it possible to keep the processing within the same subnet and still deliver acceptable performance.

- Virtualization technology is also innovating to add data locality awareness to the compute node selection, thereby speeding up analytics workloads over large amounts of data, while offering the benefits of resource isolation, better security, and improved utilization, compared to the underlying bare metal platform.

- Pivotal finds Data Lake is one of the most prevalent emerging usage patterns of the Big Data platforms. Data Lake—a design pattern where streams of data are stored and available for historical and operational analytics on the Big Data platform—is the starting point for most of the enterprises. Once the data is available in one place, business users experiment with use cases and solutions to business problems, resulting in second order benefits.

### Big Data Applications

- Pivotal considers extracting insight from a Big Data platform’s data as a Big Data application. Excel spreadsheets, and PowerPoint presentations are used for early demonstrations of these insights, and once the business cases are approved, applications need to be built to act on these insights. Developers are looking for flexibility to integrate these new Big Data applications with existing applications or build completely new applications. A good platform will make it easier for the developers to do both at the same time.

- In addition, the operations team for the application requires a platform to easily deploy, manage and scale the application as the usage increases. These operability requirements are very critical for enterprise customers.

#### 2.6.3 Key Components

**Figure 9: Pivotal Data Fabric and Analytics**

![Diagram of Pivotal Data Fabric and Analytics](https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework)

Analytics is all about distilling information (e.g., structured, unstructured across varying latencies) to generate insights. Various stages in the distillation have distinctly different platform needs. Figure 9 illustrates Pivotal data and analytical products operating in the three layers of Pivotal architecture model.

Hypothesis: Historical structured and unstructured access pattern
Analytics problems typically start with a hypothesis that is tested on the historical data. This requires analyzing massive amounts of historical data, and combining various genres of data, to extract features relevant to the problem at hand. The ability to query and mix and match large amounts of data and performance are the key requirements at this phase. Customers use the Hadoop interfaces when working with 'truly' unstructured data (e.g., video, voice, images). For semi-unstructured data (e.g., machine generated, text analytics, and transactional data) customers prefer structured interfaces, such as SQL, for analytics.

Modeling: Model building requires interactive response times when accessing structured data

Once features and relevant attributes are clearly defined and understood, models need to be built and interactive response times become one of the most important requirements. Customers get tremendous performance gains by using in-database analytics techniques, which optimize for reducing data movements. HAWQ delivers interactive response time by parallelizing analytical computation. Most of the enterprise customers have built internal teams of business analysts who are well-versed with SQL. By leveraging the SQL interfaces, customers extend the life of their resource investment, continuing business innovation at a much faster pace.

Model Operationalization: Run models in operational systems

Hypothesis validation and analytics finally result in analytical models that need to be operationalized. Operationalizing is a process where the coefficients from the models are used in Big Data applications to generate insights and scores for decision making. Some customers are building real-time applications that generate insights and alerts based on the computed features.

Requirements from the customers across the three stages of analytics are as follows:

- Interface compatibility across the three stages—SQL is a common standard for accessing data. Pivotal Data Fabric is focused on providing a single consistent interface for all data on HDFS.
- Quality of Service—Support varying response time for access to same data. Customers are looking to get batch, interactive and real-time access to same datasets. The window of data access drives the expected response times.
- Ability to manage and monitor the models. Advanced customers are using the Application Fabric technologies to build data applications.

Pivotal Data Fabric treats HDFS as the storage layer for all the data—low latency data, structured interactive data, and unstructured batch data. This improves the storage efficiency and minimizes friction, as the same data can be accessed via varying interfaces at varying latencies. Pivotal data fabric comprises the following three core technologies:

1. **Pivotal HD**: for providing the HDFS capabilities along with the Hadoop interfaces to access data (e.g., Pig, Hive, HBase and Map/Reduce).
2. **HAWQ**: for interactive analytics for the data stored on HDFS.
3. **GemFire/SQLFire**: for real-time access to the data streaming in and depositing the data on HDFS.

The Data Fabric can also run on a bare metal platform or in the public/private clouds. The deployment flexibility allows...
enterprise customers to select the ideal environment for managing their analytical environments without worrying about changing the interfaces for their internal users. As the enterprise needs change, the deployment infrastructure can be changed accordingly.

2.7 SAP

2.7.1 General Architecture Description

The data enterprises needed for managing data has grown exponentially in recent years. In addition to traditional transactional data, businesses are finding it advantageous to accumulate all kinds of other data such as weblogs, sensor data, and social media—generally referred to as Big Data—and leverage it to enrich their business applications and gain new insights.

SAP Big Data architecture (Figure 10) provides businesses with a platform to handle the high volumes, varieties and velocities of Big Data, as well as to rapidly extract and utilize business value from Big Data in the context of business applications and analytics.

SAP HANA platform for Big Data helps relax the traditional constraints of creating business applications.

Traditional disk-based systems suffer from performance hits not just due to the necessity of moving data to and from the disk, but also from the inherent limitations of disk-based architectures, such as the need to create and maintain indices and materialized aggregates of data. SAP HANA platform largely eliminates such drawbacks and enables the business to innovate freely and in real time.

Business applications are typically modeled to capture the rules and activities of real world business processes. SAP HANA platform for Big Data makes a whole new set of data available for consumption and relaxes the dependency on static rules in creating business applications. Applications can now leverage real-time Big Data insights for decision making, which vastly enhances and expands their functionality.

Traditionally, exploration and analytics of data is based on the knowledge of the structures and formats of the data. With Big Data, it is often required to first investigate the data for what it contains, whether it is relevant to the business, how it relates to the other data, and how it can be processed. SAP HANA platform for Big Data supports tools and services for data scientists to conduct such research of Big Data and enable its exploitation in the context of business applications.

2.7.2 Architecture Model

SAP Big Data architecture provides a platform for business applications with features such as the ones referenced above. The key principles of SAP Big Data architecture include:

- An architecture that puts in-memory technology data at its core and maximizes computational efficiencies by bringing the compute and data layers together.

- Support for a variety of data processing engines (such as transactional, analytical, graph, and spatial) operating directly on the same data set in memory.
- Interoperability/integration of best of breed technologies such as Hadoop/Hive for the data layer, including data storage and low level processing.

- The ability to leverage these technologies to transform existing business applications and build entirely new ones that were previously not practical.

- Comprehensive native support for predictive analytics, as well as interoperability with popular libraries such as R, for enabling roles such as data scientists to uncover and predict new business potentialities.

In a nutshell, SAP Big Data architecture is not limited to handling large amounts of data, but instead is meant to enable enterprises to identify and realize the business value of real-time Big Data.

**Figure 10: SAP Big Data Reference Architecture**

![SAP HANA Platform for Big Data](https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework)

### 2.7.3 Key Components

SAP Big Data architecture (Figure 10) enables an end-to-end platform and includes support for ingestion, storage, processing and consumption of Big Data.

The ingestion of data includes acquisition of structured, semi-structured, and unstructured data from a variety of sources to include traditional back end systems, sensors, social media, and event streams. Managing data quality through stewardship and governance and maintaining a dependable metadata store are key aspects of the data ingestion phase.

SAP Big Data architecture brings together transactional and analytical processing that directly operate on the same copy of the enterprise data that is held entirely in memory. This architecture helps by eliminating the latency between transactional and analytical applications, as there is no longer a need to copy transactional data into separate systems for analytical purposes.

With in-memory computing, important application functionalities such as planning and simulation can be executed in real time. SAP Big Data architecture includes a dedicated engine for planning and simulation as a first class component, making it possible to iterate through various simulation and planning cycles in real time.

SAP Big Data architecture includes a Graph engine. The elements of Big Data are typically loosely structured. With the constant addition of new types of data, the structure and relationship between the data is constantly evolving. In such environments, it is inefficient to impose an artificial structure (e.g. relational) on the data. Modeling the data as graphs of
complex and evolving interrelationships provides the needed flexibility in capturing dynamic, multi-faceted data.

An ever increasing number of business applications are becoming location aware. For example, businesses can now send promotions to the mobile device of a user walking into a retail store, which has a much higher chance of capturing the user’s attention and generating a sale than traditional marketing. Recognizing this trend, SAP Big Data architecture includes a spatial data processing engine to support location aware business applications. For similar reasons, inherent capabilities for text, media, and social data processing are also included.

SAP Big Data architecture supports complex event processing throughout the entire stack. Event streams (e.g., sensor data, update from capital markets) are not just an additional source of Big Data; they also require sophisticated processing of events such as processing on the fly (e.g., ETL) and analytics on the fly.

SAP Big Data architecture also enables customers to use low cost data storage and low level data processing solutions such as Hadoop. By extending the SAP HANA platform for Big Data with Hadoop, customers can bring the benefits of real-time processing to data in Hadoop systems. Scenarios for extracting high value data from Hadoop, as well as federating data processing in SAP HANA with Hadoop / Hive computational engines into a single SQL query, are fully supported.

SAP Big Data architecture enables applying common approaches, such as modeling, life cycle management, landscape management, and security, across the platform.

In summary, SAP Big Data architecture takes full advantage of the SAP HANA platform, helping businesses use Big Data in ways that will fundamentally transform their operations.

2.8 9Sight

2.8.1 General Architecture Description
This simple picture (Figure 11) sets the overall scope for the discussion and design between business and IT of systems supporting modern business needs, which include Big Data and real-time operation in a biz-tech ecosystem. Each layer is described in terms of three axes or dimensions as follows:

- **Timeliness/Consistency**: the balance between these two demands commonly drives layering of data, e.g. in data warehousing.
- **Structure/Context**: an elaboration of structured/unstructured descriptions that defines the transformation of information to data.
- **Reliance/Usage**: information trustworthiness based on its sourcing and pre-processing.

The typical list of Big Data characteristics is subsumed in these characteristic dimensions.

### 2.8.2 Architecture Model

The REAL (Realistic, Extensible, Actionable, Labile) architecture supports building an IT environment that can use Big Data in all business activities. The term “business” encompasses all social organizations of people with the intention of pursuing a set of broadly related goals, including both for-profit and nonprofit enterprises and governmental and nongovernmental stakeholders. The REAL architecture covers all information and all processes that occur in such a business, but does not attempt to architect people. The architecture model is shown in Figure 12.

### 2.8.3 Key Components

Business applications or workflows, whether operational, informational, or collaborative, with their business focus and wide variety of goals and actions, are gathered together in a single component, **utilization**.
Three information processing components are identified. **Instantiation** is the means by which measures, events, and messages from the physical world are represented as, or converted to, transactions or instances of information within the enterprise environment. **Assimilation** creates reconciled and consistent information, using ETL tools and data virtualization tools, before users have access to it. **Reification**, which sits between all utilization functions and the information itself, provides a consistent, cross-pillar view of information according to an overarching model. Reification allows access to the information in real time, and corresponds to data virtualization for “online” use. Modern data warehouse architectures use such functions extensively, but the naming is often overlapping and confusing, hence the unusual function names used here.

The Service Oriented Architecture (SOA) process- and services-based approach uses an underlying **choreography** infrastructure, which coordinates the actions of all participating elements to produce desired outcomes. There are two subcomponents: adaptive workflow management and an extensible message bus. These functions are well known in standard SOA work.

Finally, the **organization** component covers all design, management, and governance activities relating to both processes and information.

Information/data is represented in pillars for three distinct classes as described below.

- **Human-sourced information**: Information originates from people, because context comes only from the human intellect. This information is the highly subjective record of human experiences and is now almost entirely digitized and electronically stored everywhere from tweets to movies. Loosely structured and often ungoverned, this information may not reliably represent for the business what has happened in the real world.
- **Process-mediated data**: Business processes record well-defined, legally binding business events. This process-mediated data is highly structured and regulated, and includes transactions, reference tables and relationships, and the metadata that sets its context. Process-mediated data includes operational and BI systems and was the vast majority of what IT managed in the past. It is amenable to information management and to storage and manipulation in relational database systems.

- **Machine-generated data**: Sourced from the sensors, computers, etc. used to record events and measures in the physical world, such data is well-structured and usually reliable. As the Internet of Things grows, well-structured machine-generated data is of growing importance to business. Some claim that its size and speed is beyond traditional RDBMS, mandating NoSQL stores. However, high-performance RDBMSs are also often used.

Context setting information (metadata) is an integral part of the information resource, spanning all pillars.

### 2.9 LexisNexis

#### 2.9.1 General Architecture Description

The High Performance Computing Cluster (HPCC) Systems platform is designed to handle massive, multi-structured datasets ranging from hundreds of terabytes to tens of petabytes, serving as the backbone for both LexisNexis online
applications and programs within the U.S. Federal Government alike. The technology has been in existence for over a
decade, and was built from the ground up to address internal company requirements pertaining to scalability, flexibility,
agility and security. Prior to the technology being released to the open source community in June 2011, the HPCC had
been deployed to customer premises as an appliance (software fused onto a preferred vendor’s hardware), but has
since become hardware-agnostic in an effort to meet the requirements of an expanding user base. The Lexis Nexis
general architecture is shown in Figure 13.

2.9.2 Architecture Model

The HPCC is based on a distributed, shared-nothing architecture and contains two cluster types—one optimized for
“data refinery” activities (i.e., THOR cluster) and the other for “data delivery” (i.e., ROXIE cluster). The nodes comprising
both cluster types are homogenous, meaning all processing, memory and disk components are the same and based on
commercial-off-the-shelf (COTS) technology.

In addition to compute clusters, the HPCC environment also contains a number of system servers which act as a
gateway between the clusters and the outside world. The system servers are often referred to collectively as the HPCC
“middleware,” and include the following:

- **The Enterprise Control Language (ECL): compiler, executable code generator and job server (ECL Server):** Serves as the code generator and compiler that translate ECL code.

- **System data store (Dali):** Used for environment configuration, message queue maintenance, and enforcement of
  LDAP security restrictions.

- **Archiving server (Sasha):** Serves as a companion ‘housekeeping’ server to Dali.

- **Distributed File Utility (DFU Server):** Controls the spraying and despraying operations used to move data onto
  and out of THOR.

- **The inter-component communication server (ESP Server):** The inter-component communication server that
  allows multiple services to be “plugged in” to provide various types of functionality to client applications via multiple
  protocols.

2.9.3 Key Components

Core components of the HPCC include the THOR data refinery engine, ROXIE data delivery engine, and an implicitly
parallel, declarative programming language, ECL. Each component is outlined below in further detail and shown in
Figure 14.

- **THOR Data Refinery:** THOR is a massively parallel ETL engine that can be used for performing a variety of tasks
  such as massive: joins, merges, sorts, transformations, clustering, and scaling. Essentially, THOR permits any problem
  with computational complexities $O(n^2)$ or higher to become tractable.

- **ROXIE Data Delivery:** ROXIE serves as a massively parallel, high throughput, structured query response engine.
  It is suitable for performing volumes of structured queries and full text ranked Boolean search, and can also operate in
highly available (HA) environments due to its read-only nature. ROXIE also provides real-time analytics capabilities, to address real-time classifications, prediction, fraud detection and other problems that normally require handling processing and analytics on data streams.

File:/C:\DOCUME~1\user\LOCALS~1\Temp\msohtmlclip1\01\clip_image014.png
Figure 14: Lexis Nexis High Performance Computing Cluster

- **The Enterprise Control Language (ECL):** ECL is an open source, data-centric programming language used by both THOR and ROXIE for large-scale data management and query processing. ECL's declarative nature enables users to solely focus on what they need to do with their data, while leaving the exact steps for how this is accomplished within a massively parallel processing architecture (MPP) to the ECL compiler.

As multi-structured data is ingested into the system and sprayed across the nodes of a THOR cluster, users can begin to perform a multitude of ETL-like functions, including the following:

- Mapping of source fields to common record layouts used in the data
- Splitting or combining of source files, records, or fields to match the required layout
- Standardization and cleaning of vital searchable fields, such as names, addresses, dates, etc.
- Evaluation of current and historical timeframe of vital information for chronological identification and location of subjects
- Statistical and other direct analysis of the data for determining and maintaining quality as new sources and updates are included
- Mapping and translating source field data into common record layouts, depending on their purpose
- Applying duplication and combination rules to each source dataset and the common build datasets, as required

THOR is capable of operating either independently or in tandem with ROXIE; when ROXIE is present it hosts THOR results and makes them available to the end-user through a web service API.
3 Survey of Big Data Architectures

Based on the Big Data platforms surveyed, a remarkable consistency in the core of the Big Data reference platforms was observed. Each of the surveyed platforms minimally consists of the following components in its architecture:

- Data Management
- Analytics
- Infrastructure

In the subsections below, each of the nine surveyed architectures is discussed and key functionality for each are highlighted. The main features of the submitted architectures are described in the sections below.

3.1 Bob Marcus

Mr. Marcus, an individual contributor, presented a layered architecture model for Big Data. This model proposes the following six components:

A. Data Sources and Sinks

This component provides external data inputs and output to the internal Big Data components.

B. Application and User Interfaces

These are the applications (e.g., machine learning) and user interfaces (e.g., visualization) built on Big Data components.

C. Analytics Databases and Interfaces

The framework proposes integration of databases into the Big Data architecture. These can be horizontally scalable databases or single platform databases, with data extracted from the foundational data store. The framework outlines the following databases and interfaces (Table 1.)

Table 1: Databases and Interfaces in the Layered Architecture from Bob Marcus

<table>
<thead>
<tr>
<th>Category</th>
<th>Database Types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Databases</td>
<td>Analytics Databases</td>
<td>In general, these are highly optimized for read-only interactions and typically acceptable for database responses to have high latency (e.g., invoke scalable batch processing over large data sets).</td>
</tr>
<tr>
<td>Category</td>
<td>Database Types</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Operational Databases</td>
<td></td>
<td>In general, these support efficient write and read operations. NoSQL databases are often used in Big Data architectures in this capacity. Data can later be transformed and loaded into analytic databases to support analytic applications.</td>
</tr>
<tr>
<td>In Memory Data Grids</td>
<td></td>
<td>These high performance data caches and stores minimize writing to disk. They can be used for large scale real-time applications requiring transparent access to data.</td>
</tr>
<tr>
<td>Analytics and Database Interfaces</td>
<td>Batch Analytics and Database Interfaces</td>
<td>These interfaces use batch scalable processing (e.g., Map-Reduce) to access data in scalable data stores (e.g., Hadoop File System). These interfaces can be SQL-like (e.g., Hive) or programmatic (e.g., Pig).</td>
</tr>
<tr>
<td></td>
<td>Interactive Analytics and Interfaces</td>
<td>These interfaces avoid direct access data stores to provide interactive responses to end users. The data stores can be horizontally scalable databases tuned for interactive responses (e.g., HBase) or query languages tuned to data models (e.g., Drill for nested data).</td>
</tr>
<tr>
<td></td>
<td>Real-Time Analytics and Interfaces</td>
<td>Some applications require real-time responses to events occurring within large data streams (e.g., algorithmic trading). This complex event processing uses machine-based analytics, which require very high performance data access to streams and data stores.</td>
</tr>
</tbody>
</table>

D. Scalable Stream and Data Processing

This component provides filters and transforms data flows between external data resources and internal Big Data systems.

E. Scalable Infrastructure

The framework specifies scalable infrastructure that can support easy addition of new resources. Possible platforms include public and/or private clouds and horizontal scaling data stores, as well as distributed scalable data processing platforms.
F. Supporting Services

The framework specifies services needed for the implementation and management of robust Big Data systems. The subcomponents specified within the supporting services are described below.

- **Design, Develop and Deploy Tools:** The framework cautions that high level tools are limited for the implementation of Big Data applications. This should change to lower the skill levels needed by enterprise and government developers.

- **Security:** The framework also notes lack of standardized and adequate support to address data security and privacy. The framework cites that only Kerberos authentication for Hadoop, Knox exists. Capabilities should be expanded in the future by commercial vendors for enterprise and government applications.

- **Process Management:** The framework notes that commercial vendors are supplying process management tools to augment the initial open source implementations (e.g., Oozie).

- **Data Resource Management:** The author notes that open source data governance tools are still immature (e.g., Apache Falcon). These will be augmented in the future by commercial vendors.

- **System Management:** The framework notes that open source systems management tools are also immature (e.g., Ambari). However, robust system management tools are commercially available for scalable infrastructure (e.g., cloud-based).

Figure 15 provides diagrammatic representation of the architecture.

**Figure 15: Big Data Layered Architecture**

3.2 Microsoft

Microsoft defines a Big Data reference architecture that would have the four key functional capabilities, as summarized below.

A. Data Sources

According to Microsoft, “data behind Big Data” is collected for a specific purpose, creating the data objects in a form that supports the known use at the time of data collection. Once data is collected, it can be reused for a variety of purposes,
some potentially unknown at the collection time. Microsoft also explains that data sources can be classified by four characteristics that are independent of the data content and context: volume, velocity, variability, and variety.

B. Data Transformation

The second component of the Big Data reference architecture Microsoft describes is Data Transformation. Microsoft defines this as the stage where data is processed and transformed in different ways to extract value from the information. Each transformation function may have its specific pre-processing stage, including registration and metadata creation; may use different specialized data infrastructure best fitted for its requirements; and may have its own privacy and other policy considerations and interoperability. Table 2 lists the common transformation functions defined by Microsoft.

**Table 2: Microsoft Data Transformation Steps**

<table>
<thead>
<tr>
<th>Data Transformation Steps</th>
<th>Functional Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Collection</td>
<td>Data can be collected for different types and forms; similar sources and structure resulting in uniform security considerations, policies, and allows creation of an initial metadata</td>
</tr>
<tr>
<td>Aggregation</td>
<td>This is defined by Microsoft as where sets of existing data is collected to form an easily correlated metadata (e.g., identical keys) and then aggregated into a larger collection thus enriching number of objects as the collection grows.</td>
</tr>
<tr>
<td>Matching</td>
<td>This is defined as where sets of existing data collections with dissimilar metadata (e.g., keys) are aggregated into a larger collection. Similar to aggregation this stage also enhances information about each object.</td>
</tr>
<tr>
<td>Data Mining</td>
<td>Microsoft refers this as a process where data, analyzing it from many dimensions or perspectives, then producing a summary of the information in a useful form that identifies relationships within the data. There are two types of data mining: descriptive, which gives information about existing data; and predictive, which makes forecasts based on the data</td>
</tr>
</tbody>
</table>

C. Data Infrastructure
Microsoft defines Big Data infrastructure as a collection of data storage or database software, servers, storage, and networking used in support of the data transformation functions and for storage of data as needed. Furthermore, to achieve higher efficiencies, Microsoft defines infrastructure as a medium where data of different volume, variety, variability, and velocity would typically be stored and processed using computing and storage technologies tailored to those characteristics. The choice of processing and storage technology is also dependent on the transformation itself. As a result, often the same data can be transformed (either sequentially or in parallel) multiple times using independent data infrastructure.

D. Data Usage

The last component of the Big Data architecture framework is the data usage. After data has cycled through a given infrastructure, the end-result can be provided in different formats, different granularity and under different security considerations.

3.3 University of Amsterdam

University of Amsterdam (UVA) proposes a five-part Big Data framework as part of the overall cloud-based BDI. Each of the five subsections is explored below.

A. Data Models

The UVA Big Data architecture framework includes data models, structures and types that support a variety of data types produced by different data sources and need to be stored and processed.

B. Big Data Analytics

In the UVA Big Data reference architecture, Big Data analytics is envisioned as a component that will use HPC architectures and technologies. Additionally, in the proposed model, Big Data analytics is expected to be scalable vertically and horizontally. This can be naturally achieved when using cloud-based platform and Intercloud integration models/architecture. The architecture outlines the following analytics capabilities supported by high-performance computing architectures and technologies:

- Refinery, linking, and fusion
- Real-time, interactive, batch, and streaming
- Link analysis, cluster analysis, entity resolution, and complex analysis

C. Big Data Management

The UVA architecture describes Big Data management services with these components:

- Data backup, replication, curation, provenance
- Registries, indexing/search, metadata, ontologies, namespaces

D. Big Data Infrastructure
The UVA architecture defines Big Data infrastructure that requires broad network access and advanced network infrastructure to integrate distributed heterogeneous BDI integration offering reliable operation. This includes:

- General cloud-based infrastructure, platform, services, and applications to support creation, deployment, and operation of Big Data infrastructures and applications (using generic cloud features of provisioning on-demand, scalability, measured services).

- Collaborative environment infrastructure (e.g., group management) and user-facing capabilities (e.g., user portals, identity management/federation).

- Network infrastructure that interconnects typically distributed and increasingly multi-provider BDI components that may include intra-cloud (intra-provider) and Intercloud network infrastructure.

- FADI, which is to be treated as a part of the general Inter-cloud infrastructure of the BDI. FADI combines both inter-cloud network infrastructure and corresponding federated security infrastructure to support infrastructure components integration and users federation. FADI is an important component of the overall cloud and Big Data infrastructure that interconnects all the major components and domains in the multi-provider inter-cloud infrastructure including non-cloud and legacy resources. Using federation model for integrating multi-provider heterogeneous services and resources reflects current practice in building and managing complex infrastructures and allows for inter-organizational resource sharing and identity federation.

E. Big Data Security

The UVA Big Data reference architecture describes Big Data security that should protect data at rest and in motion, ensure trusted processing environments and reliable BDI operation, provide fine-grained access control, and protect users’ personal information.

3.4 IBM

IBM’s Big Data reference model proposes a Big Data framework that can be summarized in four key functional blocks, which are described below.

A. Data Discovery and Exploration (Data Source)

IBM explains data analysis as first understanding data sources, what is available in those data sources, the quality of data, and its relationship to other data elements. IBM describes this process as data discovery, which enables data scientists to create the right analytic model and computational strategy. Data discovery also supports indexing, searching, and navigation. The discovery is independent of data sources that include relational databases, flat files, and content management systems. The data store supports structured, semi-structured, or unstructured data.

**Figure 16: Data Discovery and Exploration**

My Note: Colors Missing
B. Data Analytics

IBM’s Big Data platform architecture recommends running both data processing and complex analytics on the same platform, as opposed to the traditional approach where analytics software runs on its own infrastructure and retrieves data from back-end data warehouses or other systems. The rationale is that data environments were customarily optimized for faster access to data, but not necessarily for advanced mathematical computations. Therefore, analytics were treated as a distinct workload that had to be managed in a separate infrastructure.

Within this context, IBM recommends the following:

- **Manage and analyze unstructured data**: IBM notes that a game-changing analytics platform must be able to manage, store, and retrieve both unstructured and structured data. It also has to provide tools for unstructured data exploration and analysis.

- **Analyze data in real time**: Performing analytics on activity as it unfolds presents a huge untapped opportunity for the analytics enterprise.

In above context, IBM’s Big Data platform breaks down Big Data analytics capabilities as follows:

- BI/Reporting
- Exploration/Virtualization
- Functional App
- Industry App
- Predictive Analytics
- Content Analytics

C. Big Data Platform Infrastructure

IBM notes that one of the key goals of a Big Data platform should be to reduce the analytic cycle time (i.e., the amount of time that it takes to discover and transform data, develop and score models, and analyze and publish results.) IBM further emphasizes the importance of compute intensive infrastructure with this statement, “A Big Data platform needs to support interaction with the most commonly available analytic packages, with deep integration that facilitates pushing computationally intensive activities from those packages, such as model scoring, into the platform. It needs to have a
rich set of “parallelizable” algorithms that have been developed and tested to run on Big Data. It has to have specific capabilities for unstructured data analytics, such as text analytics routines and a framework for developing additional algorithms. It must also provide the ability to visualize and publish results in an intuitive and easy-to-use manner.” IBM outlines the tools that support Big Data infrastructure as follows:

- **Hadoop:** This component supports managing and analyzing unstructured data. To support this requirement, IBM InfoSphere, BigInsights, and PureData System for Hadoop support are required.

- **Stream Computing:** This component supports analyzing in-motion data in real time.

- **Accelerators:** This component provides a rich library of analytical functions, schemas, tool sets and other artifacts for rapid development and delivery of value in Big Data projects.

D. Information Integration and Governance

The last component of IBM’s Big Data framework is integration and governance of all data sources. This includes data integration, data quality, security, lifecycle management, and master data management.

### 3.5 Oracle

The architecture as provided by Oracle contains the following four components:

A. Information Analytics

The information analytics component has two major areas: (1) descriptive analytics, and (2) predictive analytics, with subcomponents supporting those two analytics.

- **Descriptive Analytics**
  - Reporting
  - Dashboard

- **Predictive Analytics (In-Database)**
  - Statistical Analysis
  - Semantic Analysis
  - Data Mining
  - Text Mining
  - In-DB Map/Reduce
  - Spatial

B. Information Provisioning
The information-provisioning component performs discovery, conversion, and processing of massive structured, unstructured, and streaming data. This is supported by both the operational database and data warehouse.

C. Data Sources

The Oracle Big Data reference architecture supports the following data types:

- Distributed file system
- Data streams
- NoSQL/Tag-value
- Relational
- Faceted unstructured
- Spatial/relational

D. Infrastructure Services

The following capabilities support the infrastructure services:

- Hardware
- Operating system
- Storage
- Security
- Network
- Connectivity
- Virtualization
- Management

3.6 Pivotal

Pivotal’s Big Data architecture is composed of three different layers:

- Infrastructure
- Data ingestion and analytics
- Data-enabled applications

These layers, or fabrics, have to seamlessly integrate to provide a frictionless environment for users of Big Data
systems. Pivotal believes the ability to perform timely analytics largely depends on the proximity between the data store and where the analytics are being performed. This led Pivotal to fuse analytics and the data storage tier together. Additionally, Pivotal sees virtualization technology innovating to add data locality awareness to the compute node selection, thereby speeding up analytics workloads over large amounts of data, while offering the benefits of resource isolation, better security, and improved utilization, compared to the underlying bare metal platform. The following subsections further describe the capabilities:

Pivotal Data Fabric and Analytics

Pivotal uses its HAWQ analytics platform to support structured and unstructured data across varying latencies. The architecture recognizes the ability to query and mix and match large amounts of data and while supporting data performance. Pivotal supports Hadoop interfaces when working with “truly” unstructured data (e.g., video, voice, images). For semi-unstructured data (e.g., machine-generated, text analytics, and transactional data), it supports structured interfaces (e.g., SQL) for analytics. According to Pivotal, the data fabric is architected to run on bare metal platform or in the public/private clouds.

Pivotal data fabric treats HDFS as the storage layer for all data—low latency data, structured interactive data, and unstructured batch data. Pivotal states that this improves the storage efficiency and minimizes friction as the same data can be accessed via the varying interfaces at varying latencies. Pivotal data fabric comprises the following three core technologies:

- Pivotal HD for providing the HDFS capabilities, through Greenplum Database technology, to work directly on HDFS along with the Hadoop interfaces to access data (e.g., Pig, Hive, HBase and Map/Reduce).
- HAWQ for interactive analytics for the data stored on HDFS—store and query both structured and semi-structured data on HDFS.
- GemFire/SQLFire for real-time access to the data streaming in and depositing the data on HDFS.

3.7 SAP

SAP’s Big Data reference architecture relies on three basic functions that support data ingestion, data storage, and data consumption. These three functions are then supported by vertical pillar functions that include data lifecycle management, infrastructure management, and data governance and security. The subsections below provide additional details to the three basic functions and the supporting pillar functions.

A. Data Ingestion

The data ingestion function provides the foundation for the Big Data platform. The SAP HANA platform supports several data types, including structured, semi-structured, and unstructured data, which can come from a variety of sources, such as traditional back-end systems, sensors, and social media and events streams.

B. Data Storage and Processing

In the SAP architecture, analytical, and transactional processing are closely tied to the data store that eliminates latency. Such architecture is seen as favorable when considering efficiency. Also, the architecture supports in-memory
computing in real time. The SAP architecture also has features such as graph engine and spatial data processing. These components address complex and varying "multi-faceted" data and location-aware data, respectively. The architecture also has built-in support for processing distributed file system via an integrated Hadoop platform.

C. Data Consumption

The data consumption functional block supports both analytics and applications. The analytics provides various functions, including exploration, dashboards, reports, charting, and visualization. The Application component of this functional block supports machine learning and predictive and native HANA applications and services.

D. Data lifecycle management

The data lifecycle management provides common core functionality for the life of the data object that is managed by HANA platform.

E. Infrastructure Management

The Infrastructure management provides management of data objects and security for the data objects within the HANA platform.

F. Data Security and Governance

A host of functions vertically supports the above three functional blocks. These vertical services include data modeling, life cycle management, and data governance issues, including security, compliance, and audits.

3.8 9Sight

9Sight proposes a REAL architecture, aimed at IT, which supports building an IT environment capable of supporting Big Data in all business activities. The architecture model is described in the following six blocks:

- **Utilization** is a component that gathers operational, informational, or collaborative business applications/workflows with their business focus and wide variety of goals and actions.

- **Instantiation** is the process by which measures, events, and messages from the physical world are represented as, or converted to, transactions or instances of information within the enterprise environment.

- **Assimilation** is a process that creates reconciled and consistent information, using ETL and data virtualization tools, before users have access to it.

- **Reification**, which sits between all utilization functions and the information itself, provides a consistent, cross-pillar view of information according to an overarching model. Reification allows access to the information in real time, and corresponds to data virtualization for ‘online’ use.

- **Choreography** is a component that supports both the SOA process and services-based approach to delivering output. Choreography coordinates the actions of all participating elements to produce desired outcomes. There are two subcomponents: *adaptive workflow management* and an *extensible message bus*. These functions are well known in standard SOA work.
· **Organization** is a component that covers all design, management, and governance activities relating to both processes and information.

### 3.9 LexisNexis

LexisNexis submitted a HPCC Systems platform that is designed to handle massive, multi-structured datasets ranging from hundreds of terabytes to tens of petabytes. This platform was built to address requirements pertaining to scalability, flexibility, agility, and security, and was built for LexisNexis’ internal company use and the U.S. federal government. Prior to the technology being released to the open source community in June 2011, the HPCC had been deployed to customer premises as an appliance (software fused onto a preferred vendor’s hardware), but has since become hardware-agnostic in an effort to meet the requirements of an expanding user base.

**HPCC Architecture Model**

The HPCC is based on a distributed, shared-nothing architecture, and contains two cluster types:

1. **Data Refinery** is a massively parallel ETL engine used for performing a variety of tasks such as massive: joins, merges, sorts, transformations, clustering, and scaling. The component permits any problem with computational complexities \(O(n^2)\) or higher to become tractable. This component supports Big Data in structured, semi-structured, or unstructured data forms.

2. **Data Delivery** component serves as a massively parallel, high throughput, structured query response engine. It is suitable for performing volumes of structured queries and full text ranked Boolean search, and can operate in HA environments due to its read-only nature. This component also provides real-time analytics capabilities to address real-time classifications, prediction, fraud detection, and other problems that normally require handling processing and analytics on data streams.

The above two cluster types are supported by system servers that act as a gateway between the clusters and the outside world. The system servers are often referred to collectively as the HPCC “middleware” and include the following components.

- **ECL compiler** is an executable code generator and job server (ECL Server) that translates ECL code. The ECL is an open source, data-centric programming language that supports data refinery and data delivery components for large-scale data management and query processing.

- **System Data Store** is used for environment configuration, message queue maintenance, and enforcement of LDAP security restrictions.

- **Archiving server** is used for housekeeping purposes.

- **Distributed File Utility** moves data in/out of Data Refinery component.

- **The inter-component communication server (ESP Server)** allows multiple services to be “plugged in” to provide various types of functionality to client applications via multiple protocols.
3.10 Comparative view of surveyed architectures

Figure 17 was developed to visualize the commonalities and differences among the surveyed architectures. The components of each architecture were aligned horizontally. The architectures were stacked vertically to facilitate comparison of the components between the surveyed architectures. Three common components were observed in many of the surveyed architectures. These common components are as follows:

- Big Data Management and Storage
- Big Data Analytics and Application Interfaces
- Big Data Infrastructure

This alignment of surveyed architecture helps in the development of a reference architecture along common areas.

To facilitate visual comparison the following color scheme was used in Figure 17:

- Big Data Management and Storage: Light Blue
- Big Data Analytics and Application Interfaces: Light Green
- Big Data Infrastructure: Light Brown
- Other parts of the architecture: Light Grey

**Figure 17(a): Stacked View of Surveyed Architecture**
4 Conclusions

Through the collection, review, and comparison of Big Data architecture implementations, numerous commonalities were identified. These commonalities between the surveyed architectures aided in the development of the NBDRA. Even though each Big Data system is tailored to the needs of the particular implementation, certain key components appear in most of the implementations. Three general components were observed in the surveyed architectures as outlined below with common features listed below the components.

- Big Data Management and Storage
  - Structured, semi-structured and unstructured data
  - Volume, variety, velocity, and variability
  - SQL and NoSQL
  - Distributed file system

- Big Data Analytics and Application Interfaces
  - Descriptive, predictive and spatial
  - Real-time
Most of the surveyed architectures provide support for data users/consumers and orchestrators and capabilities such as systems management, data resource management, security, and data governance. The architectures also show a general lack of standardized and adequate support to address data security and privacy. Additional data security and privacy standardization would strengthen the Big Data platform.

Figure 18 is a simplified view of a Big Data reference architecture using the common components and features observed in the surveyed architectures. The components in Figure 18 are as follows:

- **Big Data Management and Storage**: The data management component provides Big Data capabilities for structured and unstructured data.
- **Big Data Analytics and Application Interfaces**: This component includes data ETL, operational databases, data analytics, data visualization, and data governance.
- **Big Data Infrastructure**: The Big Data platform provides infrastructure and systems support with capabilities to integrate with internal and external components and to interact with Big Data platform users.

Together these components and features formed the basis for development of the NBDRA as developed and detailed in the NIST Big Data Interoperability Framework: Volume 6, Reference Architecture document.

**Figure 18: Big Data Reference Architecture**

![Big Data Reference Architecture](https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework)
Appendix A: Acronyms

BDAF  Big Data Architecture Framework
BDE  Big Data Ecosystem
BDI  Big Data Infrastructure
COTS  commercial-off-the-shelf
CSR  customer service representative
DFU  Distributed File Utility
ECL  Enterprise Control Language
ETL Extract, Transform, and Load
ETL extract/transform/load
FADI Federated Access and Delivery Infrastructure
HDFS Hadoop Distributed File System
HPCC High Performance Computing Cluster
HA highly available
HPC High-Performance Computing
ITL Information Technology Laboratory
ICAF Intercloud Architecture Framework
ICFF Intercloud Federation Framework
MPP massively parallel processing architecture
NASA National Aeronautics and Space Administration
NARA National Archives and Records Administration
NIST National Institute of Standards and Technology
NSF National Science Foundation
NBD-PWG NIST Big Data Public Working Group
NBDRA NIST Big Data Reference Architecture
PaaS Platform as a Service
REAL Realistic, Extensible, Actionable, and Labile
SOA Service Oriented Architecture
SNE Systems and Network Engineering
UVA University of Amsterdam

Appendix B: References

Document References

https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework
Updated: Sat, 14 Sep 2019 05:26:25 GMT
Powered by
153


Reference Architecture

Source: http://bigdatawg.nist.gov/_uploadfil...395481670.docx (Word)

Cover Page

NIST Special Publication 1500-6

DRAFT NIST Big Data Interoperability Framework:

Volume 6, Reference Architecture

NIST Big Data Public Working Group

Reference Architecture Subgroup

Draft Version 1

April 6, 2015

http://dx.doi.org/10.6028/NIST.SP.1500-6

Inside Cover Page

NIST Special Publication 1500-6

Information Technology Laboratory

DRAFT NIST Big Data Interoperability Framework:

Volume 6, Reference Architecture

Draft Version 1

NIST Big Data Public Working Group (NBD-PWG)

Reference Architecture Subgroup

National Institute of Standards and Technology

Gaithersburg, MD 20899

https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework

Updated: Sat, 14 Sep 2019 05:26:25 GMT

Powered by mindtouch 155
April 2015

U. S. Department of Commerce

Penny Pritzker, Secretary

National Institute of Standards and Technology

Dr. Willie E. May, Under Secretary of Commerce for Standards and Technology and Director

National Institute of Standards and Technology Special Publication 1500-6

60 pages (April 6, 2015)

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by Federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, Federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. All NIST Information Technology Laboratory publications, other than the ones noted above, are available at http://www.nist.gov/publication-portal.cfm.

Public comment period: April 6, 2015 through May 21, 2015

Comments on this publication may be submitted to Wo Chang

National Institute of Standards and Technology

Attn: Wo Chang, Information Technology Laboratory

100 Bureau Drive (Mail Stop 8900) Gaithersburg, MD 20899-8930

Email: SP1500comments@nist.gov

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at NIST promotes the U.S. economy and public welfare by providing
technical leadership for the Nation’s measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL’s responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in Federal information systems. This document reports on ITL’s research, guidance, and outreach efforts in Information Technology and its collaborative activities with industry, government, and academic organizations.

Abstract

Big Data is a term used to describe the large amount of data in the networked, digitized, sensor-laden, information-driven world. While opportunities exist with Big Data, the data can overwhelm traditional technical approaches and the growth of data is outpacing scientific and technological advances in data analytics. To advance progress in Big Data, the NIST Big Data Public Working Group (NBD-PWG) is working to develop consensus on important, fundamental concepts related to Big Data. The results are reported in the NIST Big Data Interoperability Framework series of volumes. This volume, Volume 6, summarizes the work performed by the NBD-PWG to characterize Big Data from an architecture perspective, presents the NIST Big Data Reference Architecture (NBDRA) conceptual model, and discusses the components and fabrics of the NBDRA.

Keywords

Big Data, reference architecture, System Orchestrator, Data Provider, Application Provider, Framework Provider, Data Consumer, Security and Privacy Fabric, Management Fabric, use cases, Big Data characteristics

Acknowledgements

This document reflects the contributions and discussions by the membership of the NBD-PWG, co-chaired by Wo Chang of the NIST ITL, Robert Marcus of ET-Strategies, and Chaitanya Baru, University of California San Diego Supercomputer Center.

The document contains input from members of the NBD-PWG: Reference Architecture Subgroup, led by Orit Levin (Microsoft), Don Krapohl (Augmented Intelligence), and James Ketner (AT&T); Technology Roadmap Subgroup, led by Carl Buffington (Vistronix), David Boyd (InCadence Strategic Solutions), and Dan McClary (Oracle); Definitions and Taxonomies Subgroup, led by Nancy Grady (SAIC), Natasha Balac (SDSC), and Eugene Luster (R2AD); Use Cases and Requirements Subgroup, led by Geoffrey Fox (University of Indiana) and Tsegereda Beyene (Cisco); Security and Privacy Subgroup, led by Arnab Roy (Fujitsu) and Akhil Manchanda (GE).

NIST SP1500-6, Version 1 has been collaboratively authored by the NBD-PWG. As of the date of publication, there are over six hundred NBD-PWG participants from industry, academia, and government. Federal agency participants include the National Archives and Records Administration (NARA), National Aeronautics and Space Administration (NASA), National Science Foundation (NSF), and the U.S. Departments of Agriculture, Commerce, Defense, Energy, Health and Human Services, Homeland Security, Transportation, Treasury, and Veterans Affairs.

NIST acknowledges the specific contributions to this volume by the following NBD-PWG members:
Chaitan Baru
*University of California, San Diego, Supercomputer Center*

Janis Beach
*Information Management Services, Inc.*

David Boyd
*InCadence Strategic Solutions*

Scott Brim
*Internet2*

Gregg Brown
*Microsoft*

Carl Buffington
*Vistronix*

Yuri Demchenko
*University of Amsterdam*

Jill Gemmill
*Clemson University*

Nancy Grady
*SAIC*

Ronald Hale
*ISACA*

Keith Hare
*JCC Consulting, Inc.*

Richard Jones

Pavithra Kenjige
*PK Technologies*

James Kobielski
*IBM*

Donald Krapohl
*Augmented Intelligence*

Orit Levin
*Microsoft*

Eugene Luster
*DISA/R2AD*

Serge Manning
*Huawei USA*

Robert Marcus
*ET-Strategies*

Gary Mazzaferro
*AlloyCloud, Inc.*

Shawn Miller

Felix Njeh
*U.S. Department of the Army*

Gururaj Pandurangi
*Avyan Consulting Corp.*

Linda Pelekoudas
*Strategy and Design Solutions*

Dave Raddatz
*Silicon Graphics International Corp.*

John Rogers
*HP*

Arnab Roy
*Fujitsu*

Michael Seablom
*NASA*

Rupinder Singh
*McAfee, Inc.*

Anil Srivastava
*Open Health Systems Laboratory*

Glenn Wasson
*SAIC*

Timothy Zimmerlin
*Automation Technologies Inc.*

Alicia Zuniga-Alvarado

https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework

Updated: Sat, 14 Sep 2019 05:26:25 GMT

Powered by mindtouch
Notice to Readers

NIST is seeking feedback on the proposed working draft of the NIST Big Data Interoperability Framework: Volume 6, Reference Architecture. Once public comments are received, compiled, and addressed by the NBD-PWG, and reviewed and approved by NIST internal editorial board, Version 1 of this volume will be published as final. Three versions are planned for this volume, with versions 2 and 3 building on the first. Further explanation of the three planned versions and the information contained therein is included in Section 1.5 of this document.

Please be as specific as possible in any comments or edits to the text. Specific edits include, but are not limited to, changes in the current text, additional text further explaining a topic or explaining a new topic, additional references, or comments about the text, topics, or document organization. These specific edits can be recorded using one of the two following methods.

1. **TRACK CHANGES**: make edits to and comments on the text directly into this Word document using track changes.
2. **COMMENT TEMPLATE**: capture specific edits using the Comment Template ([http://bigdatawg.nist.gov/_uploadfiles/SP1500-1-to-7_comment_template.docx](http://bigdatawg.nist.gov/_uploadfiles/SP1500-1-to-7_comment_template.docx)), which includes space for Section number, page number, comment, and text edits.

Submit the edited file from either method 1 or 2 to SP1500comments@nist.gov with the volume number in the subject line (e.g., Edits for Volume 6.)

Please contact Wo Chang ([wchang@nist.gov](mailto:wchang@nist.gov)) with any questions about the feedback submission process.

Big Data professionals continue to be welcome to join the NBD-PWG to help craft the work contained in the volumes of the NIST Big Data Interoperability Framework. Additional information about the NBD-PWG can be found at [http://bigdatawg.nist.gov](http://bigdatawg.nist.gov).

Table of Contents

Executive Summary. vii

1. Introduction. 1

   1.1. Background. 1

   1.2. Scope and Objectives of the Reference Architectures Subgroup. 2

   1.3. Report Production. 3

   1.4. Report Structure. 3
1.5 Future Work on this Volume. 4

2 High Level Reference Architecture Requirements. 5
   2.1 Use Cases and Requirements. 5
   2.2 Reference Architecture Survey. 7
   2.3 Taxonomy. 7

3 NBDRA Conceptual Model. 10

4 Functional Components of the NBDRA. 12
   4.1 System Orchestrator. 12
   4.2 Data Provider. 12
   4.3 Big Data Application Provider. 14
      4.3.1 Collection. 15
      4.3.2 Preparation. 15
      4.3.3 Analytics. 15
      4.3.4 Visualization. 16
      4.3.5 Access. 16
   4.4 Big Data Framework Provider. 16
      4.4.1 Infrastructure Frameworks. 17
      4.4.2 Data Platform Frameworks. 19
      4.4.3 Processing Frameworks. 23
      4.4.4 Messaging/Communications Frameworks. 28
      4.4.5 Resource Management Framework. 28
   4.5 Data Consumer. 29

5 Management Fabric of the NBDRA. 31
   5.1 System Management. 31
   5.2 Big Data Lifecycle Management. 32
6 Security and Privacy Fabric of the NBDRA.. 34

7 Conclusion.. 35

Appendix A: Deployment Considerations. A-1

Appendix B: Terms and Definitions. B-1

Appendix C: Examples of Big Data Organization Approaches. C-1

Appendix D: Acronyms. D-1

Appendix E: Resources and References. E-1

Figures

Figure 1: NBDRA Taxonomy. 8

Figure 2: NIST Big Data Reference Architecture (NBDRA) 10

Figure 3: Data Organization Approaches. 20

Figure 4: Data Storage Technologies. 23

Figure 5: Information Flow.. 24

Figure A-1: Big Data Framework Deployment Options. A-1

Figure B-1: Differences Between Row Oriented and Column Oriented Stores. B-3

Figure B-2: Column Family Segmentation of the Columnar Stores Model. B-3

Figure B-3: Object Nodes and Relationships of Graph Databases. B-6

Tables

Table 1: Mapping Use Case Characterization Categories to Reference Architecture Components and Fabrics. 5

Table 2: 13 Dwarfs—Algorithms for Simulation in the Physical Sciences. 25

Executive Summary

The NIST Big Data Public Working group (NBD-PWG) Reference Architecture Subgroup prepared this NIST Big Data Interoperability Framework: Volume 6, Reference Architecture to provide a vendor-neutral, technology- and infrastructure-agnostic conceptual model and examine related issues. The conceptual model, referred to as the NIST Big Data Reference Architecture (NBDRA), was crafted by examining publicly available Big Data architectures representing various approaches and products. Inputs from the other NBD-PWG subgroups were also incorporated into the creation of the NBDRA. It is applicable to a variety of business environments, including tightly-integrated enterprise
systems, as well as loosely-coupled vertical industries that rely on cooperation among independent stakeholders. The NBDRA captures the two known Big Data economic value chains: information, where value is created by data collection, integration, analysis, and applying the results to data-driven services; and, the information technology (IT), where value is created by providing networking, infrastructure, platforms, and tools in support of vertical, data-based applications.

The NIST Big Data Interoperability Framework consists of seven volumes, each of which addresses a specific key topic, resulting from the work of the NBD-PWG. The seven volumes are as follows:

- Volume 1, Definitions
- Volume 2, Taxonomies
- Volume 3, Use Cases and General Requirements
- Volume 4, Security and Privacy
- Volume 5, Architectures White Paper Survey
- Volume 6, Reference Architecture
- Volume 7, Standards Roadmap

The NIST Big Data Interoperability Framework will be released in three versions, which correspond to the three stages of the NBD-PWG work. The three stages aim to achieve the following with respect to the NBDRA:

Stage 1: Identify the high-level Big Data reference architecture key components, which are technology, infrastructure, and vendor agnostic

Stage 2: Define general interfaces between the NBDRA components

Stage 3: Validate the NBDRA by building Big Data general applications through the general interfaces

Potential areas of future work for the Subgroup during stage 2 are highlighted in Section 1.5 of this volume. The current effort documented in this volume reflects concepts developed within the rapidly evolving field of Big Data.

1 Introduction

1.1 Background

There is broad agreement among commercial, academic, and government leaders about the remarkable potential of Big Data to spark innovation, fuel commerce, and drive progress. Big Data is the common term used to describe the deluge of data in today's networked, digitized, sensor-laden, and information-driven world. The availability of vast data resources carries the potential to answer questions previously out of reach, including the following:

- How can a potential pandemic reliably be detected early enough to intervene?
- Can new materials with advanced properties be predicted before these materials have ever been synthesized?
- How can the current advantage of the attacker over the defender in guarding against cyber-security threats be reversed?

There is also broad agreement on the ability of Big Data to overwhelm traditional approaches. The growth rates for data
volumes, speeds, and complexity are outpacing scientific and technological advances in data analytics, management, transport, and data user spheres.

Despite widespread agreement on the inherent opportunities and current limitations of Big Data, a lack of consensus on some important, fundamental questions continues to confuse potential users and stymie progress. These questions include the following:

- What attributes define Big Data solutions?
- How is Big Data different from traditional data environments and related applications?
- What are the essential characteristics of Big Data environments?
- How do these environments integrate with currently deployed architectures?
- What are the central scientific, technological, and standardization challenges that need to be addressed to accelerate the deployment of robust Big Data solutions?

Within this context, on March 29, 2012, the White House announced the Big Data Research and Development Initiative.[1] The initiative’s goals include helping to accelerate the pace of discovery in science and engineering, strengthening national security, and transforming teaching and learning by improving the ability to extract knowledge and insights from large and complex collections of digital data.

Six federal departments and their agencies announced more than $200 million in commitments spread across more than 80 projects, which aim to significantly improve the tools and techniques needed to access, organize, and draw conclusions from huge volumes of digital data. The initiative also challenged industry, research universities, and nonprofits to join with the federal government to make the most of the opportunities created by Big Data.

Motivated by the White House initiative and public suggestions, the National Institute of Standards and Technology (NIST) has accepted the challenge to stimulate collaboration among industry professionals to further the secure and effective adoption of Big Data. As one result of NIST’s Cloud and Big Data Forum held on January 15–17, 2013, there was strong encouragement for NIST to create a public working group for the development of a Big Data Interoperability Framework. Forum participants noted that this roadmap should define and prioritize Big Data requirements, including interoperability, portability, reusability, extensibility, data usage, analytics, and technology infrastructure. In doing so, the roadmap would accelerate the adoption of the most secure and effective Big Data techniques and technology.

On June 19, 2013, the NIST Big Data Public Working Group (NBD-PWG) was launched with extensive participation by industry, academia, and government from across the nation. The scope of the NBD-PWG involves forming a community of interests from all sectors—including industry, academia, and government—with the goal of developing consensus on definitions, taxonomies, secure reference architectures, security and privacy, and a standards roadmap. Such a consensus would create a vendor-neutral, technology- and infrastructure-independent framework that would enable Big Data stakeholders to identify and use the best analytics tools for their processing and visualization requirements on the most suitable computing platform and cluster, while also allowing value-added from Big Data service providers.

The **NIST Big Data Interoperability Framework** consists of seven volumes, each of which addresses a specific key topic, resulting from the work of the NBD-PWG. The seven volumes are as follows:

- Volume 1, Definitions
The **NIST Big Data Interoperability Framework** will be released in three versions, which correspond to the three stages of the NBD-PWG work. The three stages aim to achieve the following with respect to the NBDRA:

Stage 1: Identify the high-level Big Data reference architecture key components, which are technology, infrastructure, and vendor agnostic

Stage 2: Define general interfaces between the NBDRA components

Stage 3: Validate the NBDRA by building Big Data general applications through the general interfaces

Potential areas of future work for the Subgroup during stage 2 are highlighted in Section 1.5 of this volume. The current effort documented in this volume reflects concepts developed within the rapidly evolving field of Big Data.

### 1.2 Scope and Objectives of the Reference Architectures Subgroup

Reference architectures provide “an authoritative source of information about a specific subject area that guides and constrains the instantiations of multiple architectures and solutions.” [2] Reference architectures generally serve as a foundation for solution architectures and may also be used for comparison and alignment of instantiations of architectures and solutions.

The goal of the NBD-PWG Reference Architecture Subgroup is to develop an open reference architecture for Big Data that achieves the following objectives:

- Provides a common language for the various stakeholders
- Encourages adherence to common standards, specifications, and patterns
- Provides consistent methods for implementation of technology to solve similar problem sets
- Illustrates and improves understanding of the various Big Data components, processes, and systems, in the context of a vendor- and technology- agnostic Big Data conceptual model
- Provides a technical reference for U.S. government departments, agencies, and other consumers to understand, discuss, categorize, and compare Big Data solutions
- Facilitates analysis of candidate standards for interoperability, portability, reusability, and extendibility

The NIST Big Data Reference Architecture (NBDRA) is a high-level conceptual model crafted to serve as a tool to facilitate open discussion of the requirements, design structures, and operations inherent in Big Data. The NBDRA is
intended to facilitate the understanding of the operational intricacies in Big Data. It does not represent the system architecture of a specific Big Data system, but rather is a tool for describing, discussing, and developing system-specific architectures using a common framework of reference. The model is not tied to any specific vendor products, services, or reference implementation, nor does it define prescriptive solutions that inhibit innovation.

The NBDRA does not address the following:

- Detailed specifications for any organization’s operational systems
- Detailed specifications of information exchanges or services
- Recommendations or standards for integration of infrastructure products

1.3 Report Production

A wide spectrum of Big Data architectures have been explored and developed as part of various industry, academic, and government initiatives. The development of the NBDRA and material contained in this volume involved the following steps:

1. Announce that the NBD-PWG Reference Architecture Subgroup is open to the public to attract and solicit a wide array of subject matter experts and stakeholders in government, industry, and academia

2. Gather publicly-available Big Data architectures and materials representing various stakeholders, different data types, and diverse use cases [3]

3. Examine and analyze the Big Data material to better understand existing concepts, usage, goals, objectives, characteristics, and key elements of Big Data, and then document the findings using NIST’s Big Data taxonomies model (presented in NIST Big Data Interoperability Framework: Volume 2, Taxonomies)

4. Develop a technology-independent, open reference architecture based on the analysis of Big Data material and inputs received from other NBD-PWG subgroups

1.4 Report Structure

The organization of this document roughly corresponds to the process used by the NBD-PWG to develop the NBDRA. Following the introductory material presented in Section 1, the remainder of this document is organized as follows:

- Section 2 contains high-level, system requirements in support of Big Data relevant to the design of the NBDRA and discusses the development of these requirements
- Section 3 presents the generic, technology-independent NBDRA conceptual model
- Section 4 discusses the five main functional components of the NBDRA
- Section 5 describes the system and lifecycle management considerations related to the NBDRA management fabric
Section 6 briefly introduces security and privacy topics related to the security and privacy fabric of the NBDRA.

Appendix A summarizes deployment considerations.

Appendix B lists the terms and definitions in this document.

Appendix C provides examples of Big Data logical data architecture options.

Appendix D defines the acronyms used in this document.

Appendix E lists several general resources that provide additional information on topics covered in this document and specific references in this document.

1.5 Future Work on this Volume

The NIST Big Data Interoperability Framework will be released in three versions, which correspond to the three stages of the NBD-PWG work. The three stages aim to achieve the following with respect to the NBDRA:

Stage 1: Identify the common reference architecture components of Big Data implementations and formulate the technology-independent NBDRA.

Stage 2: Define general interfaces between the NBDRA components.

Stage 3: Validate the NBDRA by building Big Data general applications through the general interfaces.

This document (Version 1) presents the overall NBDRA components and fabrics with high-level description and functionalities.

Version 2 activities will focus on the definition of general interfaces between the NBDRA components by performing the following:

- Select use cases from the 62 (51 general and 11 security and privacy) submitted use cases or other, to be identified, meaningful use cases.

- Work with domain experts to identify workflow and interactions among the NBDRA components and fabrics.

- Explore and model these interactions within a small scale, manageable, and well-defined confined environment.

- Aggregate the common data workflow and interactions between NBDRA components and fabrics and package them into general interfaces.

Version 3 activities will focus on validation of the NBDRA through the use of the defined NBDRA general interfaces to build general Big Data applications. The validation strategy will include the following:

- Implement the same set of use cases used in Version 2 by using the defined general interfaces.

- Identify and implement a few new use cases outside the Version 2 scenarios.
- Enhance general NBDRA interfaces through lessons learned from the implementations in Version 3 activities

The general interfaces developed during Version 2 activities will offer a starting point for further refinement by any interested parties and is not intended to be a definitive solution to address all implementation needs.

2 High Level Reference Architecture Requirements

The development of a Big Data reference architecture requires a thorough understanding of current techniques, issues, and concerns. To this end, the NBD-PWG collected use cases to gain an understanding of current applications of Big Data, conducted a survey of reference architectures to understand commonalities within Big Data architectures in use, developed a taxonomy to understand and organize the information collected, and reviewed existing technologies and trends relevant to Big Data. The results of these NBD-PWG activities were used in the development of the NBDRA and are briefly described in this section.

2.1 Use Cases and Requirements

To develop the use cases, publicly available information was collected for various Big Data architectures in nine broad areas, or application domains. Participants in the NBD-PWG Use Case and Requirements Subgroup and other interested parties provided the use case details via a template, which helped standardize the responses and facilitate subsequent analysis and comparison of the use cases. However, submissions still varied in levels of detail, quantitative data, or qualitative information. The NIST Big Data Interoperability Framework: Volume 3, Use Cases and General Requirements document presents the original use cases, an analysis of the compiled information, and the requirements extracted from the use cases.

The extracted requirements represent challenges faced in seven characterization categories (Table 1) developed by the Subgroup. Requirements specific to the use cases were aggregated into high level, generalized requirements, which are vendor and technology neutral.

The use case characterization categories were used as input in the development of the NBDRA and map directly to NBDRA components and fabrics as shown in Table 1.

<table>
<thead>
<tr>
<th>Use Case Characterization Categories</th>
<th>Corresponds To</th>
<th>Reference Architecture Components and Fabrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data sources</td>
<td>→</td>
<td>Data Provider</td>
</tr>
<tr>
<td>Data transformation</td>
<td>→</td>
<td>Big Data Application Provider</td>
</tr>
</tbody>
</table>
Use Case Characterization

<table>
<thead>
<tr>
<th>Categories</th>
<th>Corresponds To</th>
<th>Reference Architecture Components and Fabrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capabilities</td>
<td>→</td>
<td>Big Data Framework Provider</td>
</tr>
<tr>
<td>Data consumer</td>
<td>→</td>
<td>Data Consumer</td>
</tr>
<tr>
<td>Security and privacy</td>
<td>→</td>
<td>Security and Privacy Fabric</td>
</tr>
<tr>
<td>Lifecycle management</td>
<td>→</td>
<td>System Orchestrator; Management Fabric</td>
</tr>
<tr>
<td>Other requirements</td>
<td>→</td>
<td>To all components and fabrics</td>
</tr>
</tbody>
</table>

The high-level, generalized requirements are presented below. The development of these generalized requirements is presented in the *NIST Big Data Interoperability Framework: Volume 3, Use Cases and Requirements* document.

**Data Provider Requirements**
- **DSR-1**: Reliable, real-time, asynchronous, streaming, and batch processing to collect data from centralized, distributed, and cloud data sources, sensors, or instruments
- **DSR-2**: Slow, bursty, and high throughput data transmission between data sources and computing clusters
- **DSR-3**: Diversified data content ranging from structured and unstructured text, documents, graphs, web sites, geospatial, compressed, timed, spatial, multimedia, simulation, and instrumental (i.e., system managements and monitoring) data

**Big Data Application Provider Requirements**
- **TPR-1**: Diversified, compute-intensive, analytic processing and machine learning techniques
- **TPR-2**: Batch and real time analytic processing
- **TPR-3**: Processing large diversified data content and modeling
- **TPR-4**: Processing data in motion (e.g., streaming, fetching new content, data tracking, traceability, data change management, and data boundaries)

**Big Data Framework Provider Requirements**
• CPR-1: Legacy software and advanced software packages
• CPR-2: Legacy and advanced computing platforms
• CPR-3: Legacy and advanced distributed computing clusters, co-processors, input/output (I/O) processing
• CPR-4: Advanced networks (e.g., Software Defined Networks [SDN]) and elastic data transmission, including fiber, cable, and wireless networks (e.g., local area network [LAN], wide area network [WAN], metropolitan area network [MAN], Wi-Fi)
• CPR-5: Legacy, large, virtual, and advanced distributed data storage
• CPR-6: Legacy and advanced programming executables, applications, tools, utilities, and libraries

Data Consumer Requirements

• DCR-1: Fast searches (~0.1 seconds) from processed data with high relevancy, accuracy, and recall
• DCR-2: Diversified output file formats for visualization, rendering, and reporting
• DCR-3: Visual layout for results presentation
• DCR-4: Rich user interface for access using browser, visualization tools
• DCR-5: High resolution, multi-dimension layer of data visualization
• DCR-6: Streaming results to clients

Security and Privacy Requirements

• SPR-1: Protect and preserve security and privacy of sensitive data
• SPR-2: Support sandbox, access control, and multi-tenant, multi-level, policy-driven authentication on protected data and ensure these are in line with accepted governance, risk, and compliance (GRC) and confidentiality, integrity, and availability (CIA) best practices

Management Requirements

• LMR-1: Data quality curation, including pre-processing, data clustering, classification, reduction, and format transformation
• LMR-2: Dynamic updates on data, user profiles, and links
• LMR-3: Data lifecycle and long-term preservation policy, including data provenance
• LMR-4: Data validation
• LMR-5: Human annotation for data validation
- LMR-6: Prevention of data loss or corruption
- LMR-7: Multi-site (including cross-border, geographically dispersed) archives
- LMR-8: Persistent identifier and data traceability
- LMR-9: Standardization, aggregation, and normalization of data from disparate sources

Other Requirements
- OR-1: Rich user interface from mobile platforms to access processed results
- OR-2: Performance monitoring on analytic processing from mobile platforms
- OR-3: Rich visual content search and rendering from mobile platforms
- OR-4: Mobile device data acquisition and management
- OR-5: Security across mobile devices and other smart devices such as sensors

2.2 Reference Architecture Survey

The NBD-PWG Reference Architecture Subgroup conducted a survey of current reference architectures to advance the understanding of the operational intricacies in Big Data and to serve as a tool for developing system-specific architectures using a common reference framework. The Subgroup surveyed currently published Big Data platforms by leading companies or individuals supporting the Big Data framework and analyzed the collected material. This effort revealed a consistency between Big Data architectures that served in the development of the NBDRA. Survey details, methodology, and conclusions are reported in *NIST Big Data Interoperability Framework: Volume 5, Architectures White Paper Survey*.

2.3 Taxonomy

The NBD-PWG Definitions and Taxonomy Subgroup focused on identifying Big Data concepts, defining terms needed to describe the new Big Data paradigm, and defining reference architecture terms. The reference architecture taxonomy presented below provides a hierarchy of the components of the reference architecture. Additional taxonomy details are presented in the *NIST Big Data Interoperability Framework: Volume 2, Taxonomy* document.

Figure 1 outlines potential actors for the seven roles developed by the NBD-PWG Definition and Taxonomy Subgroup. The blue boxes contain the name of the role at the top with potential actors listed directly below.
System Orchestrator

The System Orchestrator provides the overarching requirements that the system must fulfill, including policy, governance, architecture, resources, and business requirements, as well as monitoring or auditing activities to ensure the system complies with those requirements. The System Orchestrator role provides system requirements, high-level design, and monitoring for the data system. While the role pre-dates Big Data systems, some related design activities have changed within the Big Data paradigm.

Data Provider

A Data Provider makes data available to itself or to others. In fulfilling its role, the Data Provider creates an abstraction of various types of data sources (such as raw data or data previously transformed by another system) and makes them available through different functional interfaces. The actor fulfilling this role can be part of the Big Data system, internal to the organization in another system, or external to the organization orchestrating the system. While the concept of a Data Provider is not new, the greater data collection and analytics capabilities have opened up new possibilities for providing valuable data.

Big Data Application Provider

The Big Data Application Provider executes the manipulations of the data lifecycle to meet requirements established by the System Orchestrator. This is where the general capabilities within the Big Data framework are combined to produce the specific data system. While the activities of an application provider are the same whether the solution being built concerns Big Data or not, the methods and techniques have changed because the data and data processing is parallelized across resources.

Big Data Framework Provider

The Big Data Framework Provider has general resources or services to be used by the Big Data Application Provider in the creation of the specific application. There are many new components from which the Big Data Application Provider can choose in using these resources and the network to build the specific system. This is the role that has seen the most significant changes because of Big Data. The Big Data Framework Provider consists of one or more instances of
the three subcomponents: infrastructure frameworks, data platforms, and processing frameworks. There is no requirement that all instances at a given level in the hierarchy be of the same technology and, in fact, most Big Data implementations are hybrids combining multiple technology approaches. These provide flexibility and can meet the complete range of requirements that are driven from the Big Data Application Provider. Due to the rapid emergence of new techniques, this is an area that will continue to need discussion.

Data Consumer

The Data Consumer receives the value output of the Big Data system. In many respects, it is the recipient of the same type of functional interfaces that the Data Provider exposes to the Big Data Application Provider. After the system adds value to the original data sources, the Big Data Application Provider then exposes that same type of functional interfaces to the Data Consumer.

Security and Privacy Fabric

Security and privacy issues affect all other components of the NBDRA. The Security and Privacy Fabric interacts with the System Orchestrator for policy, requirements, and auditing and also with both the Big Data Application Provider and the Big Data Framework Provider for development, deployment, and operation. The *NIST Big Data Interoperability Framework: Volume 4, Security and Privacy* document discusses security and privacy topics.

Management Fabric

The Big Data characteristics of volume, velocity, variety, and variability demand a versatile system and software management platform for provisioning, software and package configuration and management, along with resource and performance monitoring and management. Big Data management involves system, data, security, and privacy considerations at scale, while maintaining a high level of data quality and secure accessibility.

3 NBDRA Conceptual Model

As discussed in Section 2, the NBD-PWG Reference Architecture Subgroup used a variety of inputs from other NBD-PWG subgroups in developing a vendor-neutral, technology- and infrastructure-agnostic, conceptual model of Big Data architecture. This conceptual model, the NBDRA, is shown in Figure 2 and represents a Big Data system comprised of five logical functional components connected by interoperability interfaces (i.e., services). Two fabrics envelop the components, representing the interwoven nature of management and security and privacy with all five of the components.

The NBDRA is intended to enable system engineers, data scientists, software developers, data architects, and senior decision makers to develop solutions to issues that require diverse approaches due to convergence of Big Data characteristics within an interoperable Big Data ecosystem. It provides a framework to support a variety of business environments, including tightly-integrated enterprise systems and loosely-coupled vertical industries, by enhancing understanding of how Big Data complements and differs from existing analytics, business intelligence, databases, and systems.
The NBDRA is organized around two axes representing the two Big Data value chains: the information (horizontal axis) and the IT (vertical axis). Along the information axis, the value is created by data collection, integration, analysis, and applying the results following the value chain. Along the IT axis, the value is created by providing networking, infrastructure, platforms, application tools, and other IT services for hosting of and operating the Big Data in support of required data applications. At the intersection of both axes is the Big Data Application Provider component, indicating that data analytics and its implementation provide the value to Big Data stakeholders in both value chains. The names of the Big Data Application Provider and Big Data Framework Provider components contain “providers” to indicate that these components provide or implement a specific technical function within the system.

The five main NBDRA components, shown in Figure 2 and discussed in detail in Section 4, represent different technical roles that exist in every Big Data system. These functional components are as follows:

- System Orchestrator
- Data Provider
- Big Data Application Provider
- Big Data Framework Provider
- Data Consumer

The two fabrics shown in Figure 2 encompassing the five functional components are the following:

- Management
- Security and Privacy

These two fabrics provide services and functionality to the five functional components in the areas specific to Big Data and are crucial to any Big Data solution.
The “DATA” arrows in Figure 2 show the flow of data between the system’s main components. Data flows between the components either physically (i.e., by value) or by providing its location and the means to access it (i.e., by reference). The “SW” arrows show transfer of software tools for processing of Big Data in situ. The “Service Use” arrows represent software programmable interfaces. While the main focus of the NBDRA is to represent the run-time environment, all three types of communications or transactions can happen in the configuration phase as well. Manual agreements (e.g., service-level agreements [SLAs]) and human interactions that may exist throughout the system are not shown in the NBDRA.

The components represent functional roles in the Big Data ecosystem. In system development, actors and roles have the same relationship as in the movies, but system development actors can represent individuals, organizations, software, or hardware. According to the Big Data taxonomy, a single actor can play multiple roles, and multiple actors can play the same role. The NBDRA does not specify the business boundaries between the participating actors or stakeholders, so the roles can either reside within the same business entity or can be implemented by different business entities. Therefore, the NBDRA is applicable to a variety of business environments, from tightly-integrated enterprise systems to loosely-coupled vertical industries that rely on the cooperation of independent stakeholders. As a result, the notion of internal versus external functional components or roles does not apply to the NBDRA. However, for a specific use case, once the roles are associated with specific business stakeholders, the functional components would be considered as internal or external—subject to the use case’s point of view.

The NBDRA does support the representation of stacking or chaining of Big Data systems. For example, a Data Consumer of one system could serve as a Data Provider to the next system down the stack or chain.

4 Functional Components of the NBDRA

As outlined in Section 3, the five main functional components of the NBDRA represent the different technical roles within a Big Data system. The functional components are listed below and discussed in subsequent subsections.

- **System Orchestrator**: Defines and integrates the required data application activities into an operational vertical system
- **Data Provider**: Introduces new data or information feeds into the Big Data system
- **Big Data Application Provider**: Executes a data lifecycle to meet security and privacy requirements as well as System Orchestrator-defined requirements
- **Big Data Framework Provider**: Establishes a computing framework in which to execute certain transformation applications while protecting the privacy and integrity of data
- **Data Consumer**: Includes end users or other systems that use the results of the Big Data Application Provider

4.1 System Orchestrator

The System Orchestrator role includes defining and integrating the required data application activities into an operational vertical system. Typically, the System Orchestrator involves a collection of more specific roles, performed by one or more actors, which manage and orchestrate the operation of the Big Data system. These actors may be human.

https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework

Updated: Sat, 14 Sep 2019 05:26:25 GMT
Powered by mindtouch™
The function of the System Orchestrator is to configure and manage the other components of the Big Data architecture to implement one or more workloads that the architecture is designed to execute. The workloads managed by the System Orchestrator may be assigning/provisioning framework components to individual physical or virtual nodes at the lower level, or providing a graphical user interface that supports the specification of workflows linking together multiple applications and components at the higher level. The System Orchestrator may also, through the Management Fabric, monitor the workloads and system to confirm that specific quality of service requirements are met for each workload, and may actually elastically assign and provision additional physical or virtual resources to meet workload requirements resulting from changes/surges in the data or number of users/transactions.

The NBDRA represents a broad range of Big Data systems, from tightly coupled enterprise solutions (integrated by standard or proprietary interfaces) to loosely coupled vertical systems maintained by a variety of stakeholders bounded by agreements and standard or standard-de-facto interfaces.

In an enterprise environment, the System Orchestrator role is typically centralized and can be mapped to the traditional role of system governor that provides the overarching requirements and constraints, which the system must fulfill, including policy, architecture, resources, or business requirements. A system governor works with a collection of other roles (e.g., data manager, data security, and system manager) to implement the requirements and the system’s functionality.

In a loosely coupled vertical system, the System Orchestrator role is typically decentralized. Each independent stakeholder is responsible for its own system management, security, and integration, as well as integration within the Big Data distributed system using the interfaces provided by other stakeholders.

### 4.2 Data Provider

The Data Provider role introduces new data or information feeds into the Big Data system for discovery, access, and transformation by the Big Data system. New data feeds are distinct from the data already in use by the system and residing in the various system repositories. Similar technologies can be used to access both new data feeds and existing data. The Data Provider actors can be anything from a sensor, to a human inputting data manually, to another Big Data system.

One of the important characteristics of a Big Data system is the ability to import and use data from a variety of data sources. Data sources can be internal or public records, tapes, images, audio, videos, sensor data, web logs, system and audit logs, HTTP cookies, and other sources. Humans, machines, sensors, online and offline applications, Internet technologies, and other actors can also produce data sources. The roles of Data Provider and Big Data Application Provider often belong to different organizations, unless the organization implementing the Big Data Application Provider owns the data sources. Consequently, data from different sources may have different security and privacy considerations. In fulfilling its role, the Data Provider creates an abstraction of the data sources. In the case of raw data sources, the Data Provider can potentially cleanse, correct, and store the data in an internal format that is accessible to the Big Data system that will ingest it.

The Data Provider can also provide an abstraction of data previously transformed by another system (i.e., legacy system, another Big Data system.) In this case, the Data Provider would represent a Data Consumer of the other system. For example, Data Provider 1 could generate a streaming data source from the operations performed by Data
Provider 2 on a dataset at rest.

Data Provider activities include the following, which are common to most systems that handle data:

- Collecting the data
- Persisting the data
- Providing transformation functions for data scrubbing of sensitive information such as Personally Identifiable Information (PII)
- Creating the metadata describing the data source(s), usage policies/access rights, and other relevant attributes
- Enforcing access rights on data access
- Establishing formal or informal contracts for data access authorizations
- Making the data accessible through suitable programmable push or pull interfaces
- Providing push or pull access mechanisms
- Publishing the availability of the information and the means to access it

The Data Provider exposes a collection of interfaces (or services) for discovering and accessing the data. These interfaces would typically include a registry so that applications can locate a Data Provider, identify the data of interest it contains, understand the types of access allowed, understand the types of analysis supported, locate the data source, determine data access methods, identify the data security requirements, identify the data privacy requirements, and other pertinent information. Therefore, the interface would provide the means to register the data source, query the registry, and identify a standard set of data contained by the registry.

Subject to Big Data characteristics (i.e., volume, variety, velocity, and variability) and system design considerations, interfaces for exposing and accessing data would vary in their complexity and can include both push and pull software mechanisms. These mechanisms can include subscription to events, listening to data feeds, querying for specific data properties or content, and the ability to submit a code for execution to process the data in situ. Because the data can be too large to economically move across the network, the interface could also allow the submission of analysis requests (e.g., software code implementing a certain algorithm for execution), with the results returned to the requestor. Data access may not always be automated, but might involve a human role logging into the system and providing directions where new data should be transferred (e.g., establishing a subscription to an email based data feed).

The interface between the Data Provider and Big Data Application Provider typically will go through three phases: initiation, data transfer, and termination. The initiation phase is started by either party and often includes some level of authentication/authorization. The phase may also include queries for metadata about the source or consumer, such as the list of available topics in a publish/subscribe (pub/sub) model and the transfer of any parameters (e.g., object count/size limits or target storage locations). Alternatively, the phase may be as simple as one side opening a socket connection to a known port on the other side.
The data transfer phase may be a push from the Data Provider or a pull by the Big Data Application Provider. It may also be a singular transfer or involve multiple repeating transfers. In a repeating transfer situation, the data may be a continuous stream of transactions/records/bytes. In a push scenario, the Big Data Application Provider must be prepared to accept the data asynchronously but may also be required to acknowledge (or negatively acknowledge) the receipt of each unit of data. In a pull scenario, the Big Data Application Provider would specifically generate a request that defines through parameters of the data to be returned. The returned data could itself be a stream or multiple records/units of data and the data transfer phase may consist of multiple request/send transactions.

The termination phase could be as simple as one side simply dropping the connection or could include checksums, counts, hashes, or other information about the completed transfer.

### 4.3 Big Data Application Provider

The Big Data Application Provider role executes a specific set of operations along the data lifecycle to meet the requirements established by the System Orchestrator, as well as meeting security and privacy requirements. The Big Data Application Provider is the architecture component that encapsulates the business logic and functionality to be executed by the architecture. The Big Data Application Provider activities include the following:

- Collection
- Preparation
- Analytics
- Visualization
- Access

These activities are represented by the subcomponents of the Big Data Application Provider as shown in Figure 2. The execution of these activities would typically be specific to the application and, therefore, are not candidates for standardization. However, the metadata and the policies defined and exchanged between the application's subcomponents could be standardized when the application is specific to a vertical industry.

While many of these activities exist in traditional data processing systems, the data volume, velocity, variety, and variability present in Big Data systems radically change their implementation. Traditional algorithms and mechanisms of traditional data processing implementations need to be adjusted and optimized to create applications that are responsive and can grow to handle ever growing data collections.

As data propagates through the ecosystem, it is being processed and transformed in different ways in order to extract the value from the information. Each activity of the Big Data Application Provider can be implemented by independent stakeholders and deployed as stand-alone services.

The Big Data Application Provider can be a single instance or a collection of more granular Big Data Application Providers, each implementing different steps in the data lifecycle. Each of the activities of the Big Data Application Provider may be a general service invoked by the System Orchestrator, Data Provider, or Data Consumer, such as a Web server, a file server, a collection of one or more application programs, or a combination. There may be multiple and
differing instances of each activity or a single program may perform multiple activities. Each of the activities is able to interact with the underlying Big Data Framework Providers as well as with the Data Providers and Data Consumers. In addition, these activities may execute in parallel or in any number of sequences and will frequently communicate with each other through the messaging/communications element of the Big Data Framework Provider. Also, the functions of the Big Data Application Provider, specifically the collection and access activities, will interact with the Security and Privacy Fabric to perform authentication/authorization and record/maintain data provenance.

Each of the functions can run on a separate Big Data Framework Provider or all can use a common Big Data Framework Provider. The considerations behind these different system approaches would depend on potentially different technological needs, business and/or deployment constraints (including privacy), and other policy considerations. The baseline NBDRA does not show the underlying technologies, business considerations, and topological constraints, thus making it applicable to any kind of system approach and deployment.

For example, the infrastructure of the Big Data Application Provider would be represented as one of the Big Data Framework Providers. If the Big Data Application Provider uses external/outsourced infrastructures as well, it or they will be represented as another or multiple Big Data Framework Providers in the NBDRA. The multiple grey blocks behind the Big Data Framework Providers in Figure 2 indicate that multiple Big Data Framework Providers can support a single Big Data Application Provider.

4.3.1 Collection

In general, the collection activity of the Big Data Application Provider handles the interface with the Data Provider. This may be a general service, such as a file server or web server configured by the System Orchestrator to accept or perform specific collections of data, or it may be an application specific service designed to pull data or receive pushes of data from the Data Provider. Since this activity is receiving data at a minimum, it must store/buffer the received data until it is persisted through the Big Data Framework Provider. This persistence need not be to physical media but may simply be to an in-memory queue or other service provided by the processing frameworks of the Big Data Framework Provider. The collection activity is likely where the extraction portion of the Extract, Transform, Load/Extract, Load, Transform (ETL/ELT) cycle is performed. At the initial collection stage, sets of data (e.g., data records) of similar structure are collected (and combined), resulting in uniform security, policy, and other considerations. Initial metadata is created (e.g., subjects with keys are identified) to facilitate subsequent aggregation or lookup methods.

4.3.2 Preparation

The preparation activity is where the transformation portion of the ETL/ELT cycle is likely performed, although analytics activity will also likely perform advanced parts of the transformation. Tasks performed by this activity could include data validation (e.g., checksums_hashes, format checks), cleansing (e.g., eliminating bad records/fields), outlier removal, standardization, reformatting, or encapsulating. This activity is also where source data will frequently be persisted to archive storage in the Big Data Framework Provider and provenance data will be verified or attached/associated. Verification or attachment may include optimization of data through manipulations (e.g., deduplication) and indexing to optimize the analytics process. This activity may also aggregate data from different Data Providers, leveraging metadata keys to create an expanded and enhanced data set.
4.3.3 Analytics

The analytics activity of the Big Data Application Provider includes the encoding of the low-level business logic of the Big Data system (with higher level business process logic being encoded by the System Orchestrator). The activity implements the techniques to extract knowledge from the data based on the requirements of the vertical application. The requirements specify the data processing algorithms for processing the data to produce new insights that will address the technical goal. The analytics activity will leverage the processing frameworks to implement the associated logic. This typically involves the activity providing software that implements the analytic logic to the batch and/or streaming elements of the processing framework for execution. The messaging/communication framework of the Big Data Framework Provider may be used to pass data or control functions to the application logic running in the processing frameworks. The analytic logic may be broken up into multiple modules to be executed by the processing frameworks which communicate, through the messaging/communication framework, with each other and other functions instantiated by the Big Data Application Provider.

4.3.4 Visualization

The visualization activity of the Big Data Application Provider prepares elements of the processed data and the output of the analytic activity for presentation to the Data Consumer. The objective of this activity is to format and present data in such a way as to optimally communicate meaning and knowledge. The visualization preparation may involve producing a text-based report or rendering the analytic results as some form of graphic. The resulting output may be a static visualization and may simply be stored through the Big Data Framework Provider for later access. However, the visualization activity frequently interacts with the access activity, the analytics activity, and the Big Data Framework Provider (processing and platform) to provide interactive visualization of the data to the Data Consumer based on parameters provided to the access activity by the Data Consumer. The visualization activity may be completely application implemented, leverage one or more application libraries, or may use specialized visualization processing frameworks within the Big Data Framework Provider.

4.3.5 Access

The access activity within the Big Data Application Provider is focused on the communication/interaction with the Data Consumer. Similar to the collection activity, the access activity may be a generic service such as a web server or application server that is configured by the System Orchestrator to handle specific requests from the Data Consumer. This activity would interface with the visualization and analytic activities to respond to requests from the Data Consumer (who may be a person) and uses the processing and platform frameworks to retrieve data to respond to Data Consumer requests. In addition, the access activity confirms that descriptive and administrative metadata and metadata schemes are captured and maintained for access by the Data Consumer and as data is transferred to the Data Consumer. The interface with the Data Consumer may be synchronous or asynchronous in nature and may use a pull or push paradigm for data transfer.

4.4 Big Data Framework Provider

The Big Data Framework Provider typically consists of one or more hierarchically organized instances of the components in the NBDRA IT value chain (Figure 2). There is no requirement that all instances at a given level in the hierarchy be of the same technology. In fact, most Big Data implementations are hybrids that combine multiple
technology approaches in order to provide flexibility or meet the complete range of requirements, which are driven from
the Big Data Application Provider.

Many of the recent advances related to Big Data have been in the area of frameworks designed to scale to Big Data
needs (e.g., addressing volume, variety, velocity, and variability) while maintaining linear or near linear performance.
These advances have generated much of the technology excitement in the Big Data space. Accordingly, there is a great
deal more information available in the frameworks area compared to the other components and the additional detail
provided for the Big Data Framework Provider in this document reflects this imbalance.

The Big Data Framework Provider is comprised of the following three subcomponents (from the bottom to the top):

- Infrastructure Frameworks
- Data Platform Frameworks
- Processing Frameworks

4.4.1 Infrastructure Frameworks

This Big Data Framework Provider element provides all of the resources necessary to host/run the activities of the other
components of the Big Data system. Typically, these resources consist of some combination of physical resources,
which may host/support similar virtual resources. These resources are generally classified as follows:

- Networking: These are the resources that transfer data from one infrastructure framework component to another
- Computing: These are the physical processors and memory that execute and hold the software of the other Big
  Data system components
- Storage: These are resources which provide persistence of the data in a Big Data system
- Environmental: These are the physical plant resources (power, cooling) that must be accounted for when
  establishing an instance of a Big Data system

While the Big Data Framework Provider component may be deployed directly on physical resources or on virtual
resources, at some level all resources have a physical representation. Physical resources are frequently used to deploy
multiple components that will be duplicated across a large number of physical nodes to provide what is known as
horizontal scalability. Virtualization is frequently used to achieve elasticity and flexibility in the allocation of physical
resources and is often referred to as Infrastructure as a Service (IaaS) within the cloud computing community.
Virtualization is typically found in one of three basic forms within a Big Data Architecture.

- Native: In this form, a hypervisor runs natively on the bare metal and manages multiple virtual machines consisting
  of Operating Systems (OS) and applications.
- Hosted: In this form, an OS runs natively on the bare metal and a hypervisor runs on top of that to host a client OS
  and applications. This model is not often seen in Big Data architectures due to the increased overhead of the extra
  OS layer.
- Containerized: In this form, hypervisor functions are embedded in the OS, which runs on bare metal. Applications
are run inside containers, which control or limit access to the OS and physical machine resources. This approach has gained popularity for Big Data architectures because it further reduces overhead since most OS functions are a single shared resource. It may not be considered as secure or stable since in the event that the container controls/limits fail, one application may take down every application sharing those physical resources.

The following subsections describe the types of physical and virtual resources that comprise Big Data infrastructure.

### 4.4.1.1 Networking

The connectivity of the architecture infrastructure should be addressed, as it affects the velocity characteristic of Big Data. While, some Big Data implementations may solely deal with data that is already resident in the data center and does not need to leave the confines of the local network, others may need to plan and account for the movement of Big Data either into or out of the data center. The location of Big Data systems with transfer requirements may depend on the availability of external network connectivity (i.e., bandwidth) and the limitations of Transmission Control Protocol (TCP) where there is low latency (as measured by packet Round Trip Time) with the primary senders or receivers of Big Data. To address the limitations of TCP, architects for Big Data systems may need to consider some of the advanced non-TCP based communications protocols available that are specifically designed to transfer large files such as video and imagery.

Overall availability of the external links is another infrastructure aspect relating to the velocity characteristic of Big Data that should be considered in architecting external connectivity. A given connectivity link may be able to easily handle the velocity of data while operating correctly. However, should the quality of service on the link degrade or the link fail completely, data may be lost or simply back up to the point that it can never recover. Use cases exist where the contingency planning for network outages involves transferring data to physical media and physically transporting it to the desired destination. However, even this approach is limited by the time it may require to transfer the data to external media for transport.

The volume and velocity characteristics of Big Data often are driving factors in the implementation of the internal network infrastructure as well. For example, if the implementation requires frequent transfers of large multi-gigabyte files between cluster nodes, then high speed and low latency links are required. Depending on the availability requirements, redundant and fault tolerant links may be required. Other aspects of the network infrastructure include name resolution (e.g., Domain Name Server [DNS]) and encryption along with firewalls and other perimeter access control capabilities. Finally, the network infrastructure may also include automated deployment, provisioning capabilities, or agents and infrastructure wide monitoring agents that are leveraged by the management/communication elements to implement a specific model.

Two concepts, Software Defined Networks (SDN) and Network Function Virtualization (NFV), have recently been developed in support of scalable networks and scalable systems using them.

### 4.4.1.1.1 Software Defined Networks

Frequently ignored, but critical to the performance of distributed systems and frameworks, and especially critical to Big Data implementations, is the efficient and effective management of networking resources. Significant advances in network resource management have been realized through what is known as SDN. Much like virtualization frameworks
manage shared pools of CPU/memory/disk, SDNs (or virtual networks) manage pools of physical network resources. In contrast to the traditional approaches of dedicated physical network links for data, management, I/O, and control, SDNs contain multiple physical resources (including links and actual switching fabric) that are pooled and allocated as required to specific functions and sometimes to specific applications. This allocation can consist of raw bandwidth, quality of service (QOS) priority, and even actual data routes.

4.4.1.2 Network Function Virtualization

With the advent of virtualization, virtual appliances can now reasonably support a large number of network functions that were traditionally performed by dedicated devices. Network functions that can be implemented in this manner include routing/routers, perimeter defense (e.g., firewalls), remote access authorization, and network traffic/load monitoring. Some key advantages of NFV include elasticity, fault tolerance, and resource management. For example, the ability to automatically deploy/provision additional firewalls in response to a surge in user or data connections and then un-deploy them when the surge is over can be critical in handling the volumes associated with Big Data.

4.4.1.2 Computing

The logical distribution of cluster/computing infrastructure may vary from a dense grid of physical commodity machines in a rack, to a set of virtual machines running on a cloud service provider, or to a loosely coupled set of machines distributed around the globe providing access to un-used computing resources. Computing infrastructure also frequently includes the underlying operating systems and associated services used to interconnect the cluster resources via the networking elements.

4.4.1.3 Storage

The storage infrastructure may include any resource from isolated local disks to Storage Area Networks (SANs) or network attached storage.

Two aspects of storage infrastructure technology that directly influence their suitability for Big Data solutions are capacity and transfer bandwidth. Capacity refers to the ability to handle the data volume. Local disks/file systems are specifically limited by the size of the available media. Hardware or software (HW/SW) redundant array of independent disks (RAID) solutions—in this case local to a processing node—help with scaling by allowing multiple pieces of media to be treated as a single device. However, this approach is limited by the physical dimension of the media and the number of devices the node can accept. SAN and network-attached storage (NAS) implementations—often known as shared disk solutions—remove that limit by consolidating storage into a storage specific device. By consolidating storage, the second aspect—transfer bandwidth—may become an issue. While both network and I/O interfaces are getting faster and many implementations support multiple transfer channels, I/O bandwidth can still be a limiting factor. In addition, despite the redundancies provided by RAID, hot spares, multiple power supplies, and multiple controllers, these boxes can often become I/O bottlenecks or single points of failure in an enterprise. Many Big Data implementations address these issues by using distributed file systems within the platform framework.
4.4.1.4 Environmental Resources

Environmental resources, such as power and heating, ventilation, and air conditioning (HVAC), are critical to the Big Data Framework Provider. While environmental resources are critical to the operation of the Big Data system, they are not within the technical boundaries and are, therefore, not depicted in Figure 2, the NBDRA conceptual model.

Adequately sized infrastructure to support application requirements is critical to the success of Big Data implementations. The infrastructure architecture operational requirements range from basic power and cooling to external bandwidth connectivity (as discussed above). A key evolution that has been driven by Big Data is the increase in server density (i.e., more CPU/memory/disk per rack unit). However, with this increased density, infrastructure specifically power and cooling may not be distributed within the data center to allow for sufficient power to each rack or adequate air flow to remove excess heat. In addition, with the high cost of managing energy consumption within data centers, technologies have been developed that actually power down or idle resources not in use to save energy or to reduce consumption during peak periods.

4.4.2 Data Platform Frameworks

Data Platform Frameworks provide for the logical data organization and distribution combined with the associated access application programming interfaces (APIs) or methods. The frameworks may also include data registry and metadata services along with semantic data descriptions such as formal ontologies or taxonomies. The logical data organization may range from simple delimited flat files to fully distributed relational or columnar data stores. The storage mediums range from high latency robotic tape drives, to spinning magnetic media, to flash/solid state disks, or to random access memory. Accordingly, the access methods may range from file access APIs to query languages such as SQL. Typical Big Data framework implementations would support either basic file system style storage or in-memory storage and one or more indexed storage approaches. Based on the specific Big Data system considerations, this logical organization may or may not be distributed across a cluster of computing resources.

In most aspects, the logical data organization and distribution in Big Data storage frameworks mirrors the common approach for most legacy systems. Figure 3 presents a brief overview of data organization approaches for Big Data.

![Figure 3: Data Organization Approaches](https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework)

Many Big Data logical storage organizations leverage the common file system concept where chunks of data are organized into a hierarchical namespace of directories as their base and then implement various indexing methods within the individual files. This allows many of these approaches to be run both on simple local storage file systems for
testing purposes or on fully distributed file systems for scale.

1.4.2.1 In-memory

The infrastructure illustrated in the NBDRA (Figure 2) indicates that physical resources are required to support analytics. However, such infrastructure will vary (i.e., will be optimized) for the Big Data characteristics of the problem under study. Large, but static, historical datasets with no urgent analysis time constraints would optimize the infrastructure for the volume characteristic of Big Data, while time-critical analyses such as intrusion detection or social media trend analysis would optimize the infrastructure for the velocity characteristic of Big Data. Velocity implies the necessity for extremely fast analysis and the infrastructure to support it—namely, very low latency, in-memory analytics.

In-memory database technologies are increasingly used due to the significant reduction in memory prices and the increased scalability of modern servers and operating systems. Yet, an in-memory element of a velocity-oriented infrastructure will require more than simply massive random-access memory (RAM). It will also require optimized data structures and memory access algorithms to fully exploit RAM performance. Current in-memory database offerings are beginning to address this issue.

Traditional database management architectures are designed to use spinning disks as the primary storage mechanism, with the main memory of the computing environment relegated to providing caching of data and indexes. Many of these in-memory storage mechanisms have their roots in the massively parallel processing and super computer environments popular in the scientific community.

These approaches should not be confused with solid state (e.g., flash) disks or tiered storage systems that implement memory-based storage which simply replicate the disk style interfaces and data structures but with faster storage medium. Actual in-memory storage systems typically eschew the overhead of file system semantics and optimize the data storage structure to minimize memory footprint and maximize the data access rates. These in-memory systems may implement general purpose relational and other NoSQL style organization and interfaces or be completely optimized to a specific problem and data structure.

Like traditional disk-based systems for Big Data, these implementations frequently support horizontal distribution of data and processing across multiple independent nodes—although shared memory technologies are still prevalent in specialized implementations. Unlike traditional disk-based approaches, in-memory solutions and the supported applications must account for the lack of persistence of the data across system failures. Some implementations leverage a hybrid approach involving write-through to more persistent storage to help alleviate the issue.

The advantages of in-memory approaches include faster processing of intensive analysis and reporting workloads. In-memory systems are especially good for analysis of real time data such as that needed for some complex event processing of streams. For reporting workloads, performance improvements can often be on the order of several hundred times faster—especially for sparse matrix and simulation type analytics.

1.4.2.2 File Systems

Many Big Data processing frameworks and applications access their data directly from underlying file systems. In almost all cases, the file systems implement some level of the Portable Operating System Interface (POSIX) standards for
permissions and the associated file operations. This allows other higher-level frameworks for indexing or processing to operate with relative transparency as to whether the underlying file system is local or fully distributed. File-based approaches consist of two layers, the file system organization and the data organization within the files.

1.4.2.2.1 File System Organization

File systems tend to be either centralized or distributed. Centralized file systems are basically implementations of local file systems that are placed on a single large storage platform (e.g., SAN or NAS) and accessed via some network capability. In a virtual environment, multiple physical centralized file systems may be combined, split, or allocated to create multiple logical file systems.

Distributed file systems (also known as cluster file systems) seek to overcome the throughput issues presented by the volume and velocity characteristics of big data combine I/O throughput across multiple devices (spindles) on each node, with redundancy and failover mirroring or replicating data at the block level across multiple nodes. The data replication is specifically designed to allow the use of heterogeneous commodity hardware across the Big Data cluster. Thus, if a single drive or an entire node should fail, no data is lost because it is replicated on other nodes and throughput is only minimally affected because that processing can be moved to the other nodes. In addition, replication allows for high levels of concurrency for reading data and for initial writes. Updates and transaction style changes tend to be an issue for many distributed file systems because latency in creating replicated blocks will create consistency issues (e.g., a block is changed but another node reads the old data before it is replicated.) Several file system implementations also support data compression and encryption at various levels. One major caveat is that, for distributed block based file systems, the compression/encryption must be splittable and allow any given block to be decompressed/ decrypted out of sequence and without access to the other blocks.

Distributed object stores (DOSs) (also known as global object stores) are a unique example of distributed file system organization. Unlike the approaches described above, which implement a traditional file system hierarchy namespace approach, DOSs present a flat name space with a globally unique identifier (GUID) for any given chunk of data. Generally, data in the store is located through a query against a metadata catalog that returns the associated GUIDs. The GUID generally provides the underlying software implementation with the storage location of the data of interest. These object stores are developed and marketed for storage of very large data objects, from complete data sets to large individual objects (e.g., high resolution images in the tens of gigabytes [GBs] size range). The biggest limitation of these stores for Big Data tends to be network throughput (i.e., speed) because many require the object to be accessed in total. However, future trends point to the concept of being able to send the computation/application to the data versus needing to bring the data to the application.

From a maturity perspective, two key areas where distributed file systems are likely to improve are (1) random write I/O performance and consistency, and (2) the generation of de facto standards at a similar or greater level as the Internet Engineering Task Force (IETF) Requests for Comments (RFC), such as those currently available for the network file system (NFS) protocol. DOSs, while currently available and operational from several commercial providers and part of the roadmap for large organizations such as the National Geospatial Intelligence Agency (NGA), currently are essentially proprietary implementations. For DOSs to become prevalent within Big Data ecosystems, there should be: some level of interoperability available (i.e., through standardized APIs); standards-based approaches for data discovery; and, most importantly, standards-based approaches that allow the application to be transferred over the grid.

https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework
Updated: Sat, 14 Sep 2019 05:26:25 GMT
Powered by mindtouch™
185
and run locally to the data versus transferring the data to the application.

1.4.2.2 In File Data Organization

Very little is different in Big Data for in file data organization. File based data can be text, binary data, fixed length
records, or some sort of delimited structure (e.g., comma separated values [CSV], Extensible Markup Language [XML]).
For record oriented storage (either delimited or fixed length), this generally is not an issue for Big Data unless individual
records can exceed a block size. Some distributed file system implementations provide compression at the volume or
directory level and implement it below the logical block level (e.g., when a block is read from the file system, it is
decompressed/decrypted before being returned). Because of their simplicity, familiarity, and portability, delimited files are
frequently the default storage format in many Big Data implementations. The tradeoff is I/O efficiency (i.e., speed). While
individual blocks in a distributed file system might be accessed in parallel, each block still needs to be read in sequence.
In the case of a delimited file, if only the last field of certain records is of interest with perhaps hundreds of fields, a lot of
I/O and processing bandwidth is wasted.

Binary formats tend to be application or implementation specific. While they can offer much more efficient access due to
smaller data sizes (i.e., integers are two to four bytes in binary while they are one byte per digit in ASCII [American
Standard Code for Information Interchange]), they offer limited portability between different implementations. At least
one popular distributed file system provides its own standard binary format, which allows data to be portable between
multiple applications without additional software. However, the bulk of the indexed data organization approaches
discussed below leverage binary formats for efficiency.

1.4.2.3 Indexed Storage Organization

The very nature of Big Data (primarily the volume and velocity characteristics) practically drives requirements to some
form of indexing structure. Big Data volume requires that specific data elements be located quickly without scanning
across the entire dataset. Big Data velocity also requires that data can be located quickly either for matching (e.g.,
incoming data matches something in an existing data set) or to know where to write/update new data.

The choice of a particular indexing method or methods depends mostly on the data and the nature of the application to
be implemented. For example, graph data (i.e., vertices, edges, and properties) can easily be represented in flat text
files as vertex-edge pairs, edge-vertex-vertex triples, or vertex-edge list records. However, processing this data
efficiently would require potentially loading the entire data set into memory or being able to distribute the application and
data set across multiple nodes so a portion of the graph is in memory on each node. Splitting the graph across nodes
requires the nodes to communicate when graph sections have vertices that connect with vertices on other processing
nodes. This is perfectly acceptable for some graph applications such as shortest path especially when the graph is
static. Some graph processing frameworks operate using this exact model. However, this approach is infeasible for large
scale graphs requiring a specialized graph storage framework, where the graph is dynamic or searching or matching to
a portion of the graph is needed quickly.

Indexing approaches tend to be classified by the features provided in the implementation, specifically: the complexity of
the data structures that can be stored; how well they can process links between data; and, how easily they support
multiple access patterns as shown in Figure 4. Since any of these features can be implemented in custom application
code, the values portrayed represent approximate norms. For example, key-value (KV) stores work well for data that is only accessed through a single key, whose values can be expressed in a single flat structure, and where multiple records do not need to be related. While document stores can support very complex structures of arbitrary width and tend to be indexed for access via multiple document properties, they do not tend to support inter-record relationships well.

It is noted that the specific implementations for each storage approach vary significantly enough that all of the values for the features represented here are really ranges. For example, relational data storage implementations are supporting increasingly complex data structures and on going work aims to add more flexible access patterns natively in BigTable columnar implementations. Within Big Data, the performance of each of these features tends to drive the scalability of that approach depending on the problem being solved. For example, if the problem is to locate a single piece of data for a unique key, then KV stores will scale really well. However, if a problem requires general navigation of the relationships between multiple data records, a graph storage model will likely provide the best performance.

File:/C:\DOCUME~1\user\LOCALS~1\Temp\mshtmlclip1\01\clip_image002.png

Figure 4: Data Storage Technologies

4.4.3 Processing Frameworks

The processing frameworks for Big Data provide the necessary infrastructure software to support implementation of applications that can deal with the volume, velocity, variety, and variability of data. Processing frameworks define how the computation and processing of the data is organized. Big Data applications rely on various platforms and technologies to meet the challenges of scalable data analytics and operation.

Processing frameworks generally focus on data manipulation, which falls along a continuum between batch and streaming oriented processing. However, depending on the specific data organization platform, and actual processing requested, any given framework may support a range of data manipulation from high latency to NRT processing. Overall, many Big Data architectures will include multiple frameworks to support a wide range of requirements.

Typically, processing frameworks are categorized based on whether they support batch or streaming processing. This categorization is generally stated from the user perspective (e.g., how fast does a user get a response to a request).
However, Big Data processing frameworks actually have three processing phases: data ingestion, data analysis, and data dissemination, which closely follow the flow of data through the architecture. The Big Data Application Provider activities control the application of specific framework capabilities to these processing phases. The batch-streaming continuum, illustrated in the processing subcomponent in Figure 2, can be applied to the three distinct processing phases. For example, data may enter a Big Data system at high velocity and the end user must quickly retrieve a summary of the prior day’s data. In this case, the ingestion of the data into the system needs to be near real time (NRT) and keep up with the data stream. The analysis portion could be incremental (e.g., performed as the data is ingested) or could be a batch process performed at a specified time, while retrieval (i.e., read visualization) of the data could be interactive. Specific to the use case, data transformation may take place at any point during its transit through the system. For example, the ingestion phase may only write the data as quickly as possible, or it may run some foundational analysis to track incrementally computed information such as minimum, maximum, average. The core processing job may only perform the analytic elements required by the Big Data Application Provider and compute a matrix of data or may actually generate some rendering like a heat map to support the visualization component. To permit rapid display, the data dissemination phase almost certainly does some rendering, but the extent depends on the nature of the data and the visualization.

For the purposes of this discussion, most processing frameworks can be described with respect to their primary location within the information flow illustrated in Figure 5.

**Figure 5: Information Flow**

The green shading in Figure 5 illustrates the general sensitivity of that processing phase to latency, which is defined as the time from when a request or piece of data arrives at a system until its processing/delivery is complete. For Big Data, the ingestion may or may not require NRT performance to keep up with the data flow. Some types of analytics (specifically those categorized as complex event processing [CEP]) may or may not require NRT processing. The Data Consumer generally is located at the far right of Figure 5, depending upon the use case and application batch responses (e.g., a nightly report is emailed) may be sufficient. In other cases, the user may be willing to wait minutes for the results of a query to be returned, or they may need immediate alerting when critical information arrives at the system. In general, batch analytics tend to better support long term strategic decision making, where the overall view or direction is not affected by the latest small changes in the underlying data. Streaming analytics are better suited for tactical decision making, where new data needs to be acted upon immediately. A primary use case for streaming analytics would be electronic trading on stock exchanges where the window to act on a given piece of data can be measured in microseconds. Messaging and communication provides the transfer of data between processing elements and the buffering necessary to deal with the deltas in data rate, processing times, and data requests.

Typically, Big Data discussions focus around the categories of batch and streaming frameworks for analytics. However, frameworks for retrieval of data that provide interactive access to Big Data are becoming a more prevalent. It is noted that the lines between these categories are not solid or distinct, with some frameworks providing aspects of each.
category.

4.4.3.1 Batch Frameworks

Batch frameworks, whose roots stem from the mainframe processing era, are some of the most prevalent and mature components of a Big Data architecture because the historically long processing times for large data volumes. Batch frameworks ideally are not tied to a particular algorithm or even algorithm type, but rather provide a programming model where multiple classes of algorithms can be implemented. Also, when discussed in terms of Big Data, these processing models are frequently distributed across multiple nodes of a cluster. They are routinely differentiated by the amount of data sharing between processes/activities within the model.

In 2004, a list of algorithms for simulation in the physical sciences was developed that became known as the “Seven Dwarfs”.\[4\] The list was recently modified and extended to the 13 algorithms (Table 2), based on the following definition: “A dwarf is an algorithmic method that computes a pattern of computation and communication.”\[5\]

<table>
<thead>
<tr>
<th>Dense Linear Algebra*</th>
<th>Combinational Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse Linear Algebra*</td>
<td>Graph Traversal</td>
</tr>
<tr>
<td>Spectral methods</td>
<td>Dynamic Programming</td>
</tr>
<tr>
<td>N-Body Methods</td>
<td>Backtrack and Branch-and-Bound</td>
</tr>
<tr>
<td>Structured Grids*</td>
<td>Graphical Models</td>
</tr>
<tr>
<td>Unstructured Grids*</td>
<td>Finite State Machines</td>
</tr>
<tr>
<td>Map/Reduce</td>
<td></td>
</tr>
</tbody>
</table>

Notes:
* Indicates one of the original seven dwarfs. The recent list modification removed three of the original seven algorithms: Fast Fourier Transform, Particles, and Monte Carlo.

Many other algorithms or processing models have been defined over the years two of the best-known models in the Big Data space, Map/Reduce (M/R) and Bulk Synchronous Parallel (BSP), are described in the following subsections.
4.4.3.1 Map/Reduce

Several major Internet search providers popularized the M/R model as they worked to implement their search capabilities. In general, M/R programs follow five basic stages:

1. Input preparation and assignment to mappers
2. Map a set of keys and values to new keys and values: Map(k1,v1) → list(k2,v2)
3. Shuffle data to each reducer and each reducer sorts its input—each reducer is assigned a set of keys (k2)
4. Run the reduce on a list(v2) associated with each key and produce an output: Reduce(k2, list(v2)) → list(v3)
5. Final output the lists(v3) from each reducer are combined and sorted by k2

While there is a single output, nothing in the model prohibits multiple input data sets. It is extremely common for complex analytics to be built as workflows of multiple M/R jobs. While the M/R programming model is best suited to aggregation (e.g., sum, average, group-by) type analytics, a wide variety of analytic algorithms have been implemented within processing frameworks. M/R does not generally perform well with applications or algorithms that need to directly update the underlying data. For example, updating the values for a single key would require the entire data set be read, output, and then moved or copied over the original data set. Because the mappers and reducers are stateless in nature, applications that require iterative computation on parts of the data or repeated access to parts of the data set do not tend to scale or perform well under M/R.

Due to its shared nothing approach, the usability of M/R for Big Data applications has made it popular enough that a number of large data storage solutions (mostly those of the NoSQL variety) provide implementations within their architecture. One major criticism of M/R early on was that the interfaces to most implementations were at too low of a level (written in Java or JavaScript.) However many of the more prevalent implementations now support high level procedural and declarative language interfaces and even visual programming environments are beginning to appear.

4.4.3.1.2 Bulk Synchronous Parallel

The BSP programming model, originally developed by Leslie Valiant[6], combines parallel processing with the ability of processing modules to send messages to other processing modules and explicit synchronization of the steps. A BSP algorithm is composed of what are termed “supersteps”, which are comprised of the following three distinct elements.

- Bulk Parallel Computation: Each processor performs the calculation/analysis on its local chunk of data.
- Message Passing: As each processor performs its calculations it may generate messages to other processors. These messages are frequently updates to values associated with the local data of other processors but may also result in the creation of additional data.
- Synchronization: Once a processor has completed processing its local data it pauses until all other processors have also completed their processing.

This cycle can be terminated by all the processors “voting to stop”, which will generally happen when a processor has generated no messages to other processors (e.g., no updates). All processors voting to stop in turn indicates that there are no new updates to any processors’ data and the computation is complete. Alternatively, the cycle may be terminated after a fixed number of supersteps have been completed (e.g., after a certain number of iterations of a Monte Carlo
The advantage of BSP over Map/Reduce is that processing can actually create updates to the data being processed. It is this distinction that has made BSP popular for graph processing and simulations where computations on one node/element of data directly affect values or connections with other nodes/elements. The disadvantage of BSP is the high cost of the synchronization barrier between supersteps. Should the distribution of data or processing between processors become highly unbalanced then some processors may become overloaded while others remain idle.

Numerous extensions and enhancements to the basic BSP model have been developed and implemented over the years, many of which are designed to address the balancing and cost of synchronization problems.

### 4.4.3.2 Streaming Frameworks

Streaming frameworks are built to deal with data that requires processing as fast or faster than the velocity at which it arrives into the Big Data system. The primary goal of streaming frameworks is to reduce the latency between the arrival of data into the system and the creation, storage, or presentation of the results. Complex Event Processing (CEP) is one of the problem domains frequently addressed by streaming frameworks. CEP uses data from one or more streams/sources to infer or identify events or patterns in NRT.

Almost all streaming frameworks for Big Data available today implement some form of basic workflow processing for the streams. These workflows use messaging/communications frameworks to pass data objects (often referred to as events) between steps in the workflow. This frequently takes the form of a directed execution graph. The distinguishing characteristics of streaming frameworks are typically organized around the following three characteristics: event ordering and processing guarantees, state management, and partitioning/parallelism. These three characteristics are described below.

#### 4.4.3.2.1 Event Ordering and Processing Guarantees

This characteristic refers to whether stream processing elements are guaranteed to see messages or events in the order they are received by the Big Data System, as well as how often a message or event may or may not be processed. In a non-distributed and single stream mode, this type of guarantee is relatively trivial. Once distributed and/or multiple streams are added to the system, the guarantee becomes more complicated. With distributed processing, the guarantees must be enforced for each partition of the data (partitioning and parallelism as further described below.) Complications arise when the process/task/job dealing with a partition dies. Processing guarantees are typically divided into the following three classes:

- **At-most-once delivery**: This is the simplest form of guarantee and allows for messages or events to be dropped if there is a failure in processing or communications or if they arrive out of order. This class of guarantee is applicable for data where there is no dependence of new events on the state of the data created by prior events.

- **At-least-once delivery**: Within this class, the frameworks will track each message or event (and any downstream messages or events generated) to verify that it is processed within a configured time frame. Messages or events that are not processed in the time allowed are re-introduced into the stream. This mode requires extensive state management by the framework (and sometimes the associated application) to track which events have been processed by which stages.
of the workflow. However, under this class, messages or events may be processed more than once and also may arrive
out of order. This class of guarantee is appropriate for systems where every message or event must be processed
regardless of the order (e.g., no dependence on prior events), and the application either is not affected by duplicate
processing of events or has the ability to de-duplicate events itself.

- Exactly-once delivery: This class of framework processing requires the same top level state tracking as At-least-
onece delivery but embeds mechanisms within the framework to detect and ignore duplicates. This class often
guarantees ordering of event arrivals and is required for applications where the processing of any given event is
dependent on the processing of prior events. It is noted that these guarantees only apply to data handling within the
framework. If data is passed outside the framework processing topology, then by an application then the application
must ensure the processing state is maintained by the topology or duplicate data may be forwarded to non-framework
elements of the application.

In the latter two classes, some form of unique key must be associated with each message or event to support de-
duplication and event ordering. Often, this key will contain some form of timestamp plus the stream ID to uniquely
identify each message in the stream.

4.4.3.2.2 State Management

A critical characteristic of stream processing frameworks is their ability to recover and not lose critical data in the event
of a process or node failure within the framework. Frameworks typically provide this state management through
 persistence of the data to some form of storage. This persistence can be: local, allowing the failed process to be
restarted on the same node; a remote or distributed data store, allowing the process to be restarted on any node; or,
local storage that is replicated to other nodes. The tradeoff between these storage methods is the latency introduced by
the persistence. Both the amount of state data persisted and the time required to assure that the data is persisted
contribute to the latency. In the case of a remote or distributed data store, the latency required is generally dependent on
the extent to which the data store implements ACID (Atomicity, Consistency, Isolation, Durability) or BASE (Basically
Available, Soft state, Eventual consistency) style consistency. With replication of local storage, the reliability of the state
management is entirely tied to the ability of the replication to recover in the event of a process or node failure.
Sometimes this state replication is actually implemented using the same messaging/communication framework that is
used to communicate with and between stream processors. Some frameworks actually support full transaction
semantics, including multi-stage commits and transaction rollbacks. The tradeoff is the same one that exists for any
transaction system is that any type of ACID-like guarantee will introduce latency. Too much latency at any point in the
stream flow can create bottlenecks and, depending on the ordering or processing guarantees, can result in deadlock or
loop states especially when some level of failure is present.

4.4.3.2.3 Partitioning and Parallelism

This streaming framework characteristic relates to the distribution of data across nodes and worker tasks to provide the
horizontal scalability needed to address the volume and velocity of Big Data streams. This partitioning scheme must
interact with the resource management framework to allocate resources. The even distribution of data across partitions
is essential so that the associated work is evenly distributed. The even data distribution directly relates to selection of a
key (e.g., user ID, host name) that can be evenly distributed. The simplest form might be using a number that
increments by one and then is processed with a modulus function of the number of tasks/workers available. If data dependencies require all records with a common key be processed by the same worker, then assuring an even data distribution over the life of the stream can be difficult. Some streaming frameworks address this issue by supporting dynamic partitioning where the partition of overloaded workers is split and allocated to existing workers or newly created workers. To achieve success especially with a data/state dependency related to the key, it is critical that the framework have state management, which allows the associated state data to be moved/transitioned to the new/different worker.

4.4.4 Messaging/Communications Frameworks

Messaging and communications frameworks have their roots in the High Performance Computing (HPC) environments long popular in the scientific and research communities. Messaging/Communications Frameworks were developed to provide APIs for the reliable queuing, transmission, and receipt of data between nodes in a horizontally scaled cluster. These frameworks typically implement either a point-to-point transfer model or a store-and-forward model in their architecture. Under a point-to-point model, data is transferred directly from the sender to the receivers. The majority of point-to-point implementations do not provide for any form of message recovery should there be a program crash or interruption in the communications link between sender and receiver. These frameworks typically implement all logic within the sender and receiver program space, including any delivery guarantees or message retransmission capabilities. One common variation of this model is the implementation of multicast (i.e., one-to-many or many-to-many distribution), which allows the sender to broadcast the messages over a “channel”, and receivers in turn listen to those channels of interest. Typically, multicast messaging does not implement any form of guaranteed receipt. With the store-and-forward model, the sender would address the message to one or more receivers and send it to an intermediate broker, which would store the message and then forward it on to the receivers. Many of these implementations support some form of persistence for messages not yet delivered, providing for recovery in the event of process or system failure. Multicast messaging can also be implemented in this model and is frequently referred to as a pub/sub model.

4.4.5 Resource Management Framework

As Big Data systems have evolved and become more complex, and as businesses work to leverage limited computation and storage resources to address a broader range of applications and business challenges, the requirement to effectively manage those resources has grown significantly. While tools for resource management and “elastic computing” have expanded and matured in response to the needs of cloud providers and virtualization technologies, Big Data introduces unique requirements for these tools. However, Big Data frameworks tend to fall more into a distributed computing paradigm, which presents additional challenges.

The Big Data characteristics of volume and velocity drive the requirements with respect to Big Data resource management. Elastic computing (i.e., spawning another instance of some service) is the most common approach to address expansion in volume or velocity of data entering the system. CPU and memory are the two resources that tend to be most essential to managing Big Data situations. While shortages or over-allocation of either will have significant impacts on system performance, improper or inefficient memory management is frequently catastrophic. Big Data differs and becomes more complex in the allocation of computing resources to different storage or processing frameworks that are optimized for specific applications and data structures. As such, resource management frameworks will often use data locality as one of the input variables in determining where new processing framework elements (e.g., master...
nodes, processing nodes, job slots) are instantiated. Importantly, because the data is big (i.e., large volume), it generally is not feasible to move data to the processing frameworks. In addition, while nearly all Big Data processing frameworks can be run in virtualized environments, most are designed to run on bare metal commodity hardware to provide efficient I/O for the volume of the data.

Two distinct approaches to resource management in Big Data frameworks are evolving. The first is intra-framework resource management, where the framework itself manages allocation of resources between its various components. This allocation is typically driven by the framework’s workload and often seeks to “turn off” unneeded resources to either minimize overall demands of the framework on the system or to minimize the operating cost of the system by reducing energy use. With this approach, applications can seek to schedule and request resources that are managed through scheduling queues and job classes.

The second approach is inter-framework resource management, which is designed to address the needs of many Big Data systems to support multiple storage and processing frameworks that can address and be optimized for a wide range of applications. With this approach, the resource management framework actually runs as a service that supports and manages resource requests from frameworks, monitoring framework resource usage, and in some cases manages application queues. In many ways, this approach is like the resource management layers common in cloud/virtualization environments, and there are efforts underway to create hybrid resource management frameworks that handle both physical and virtual resources.

Taking these concepts further and combining them is resulting in the emerging technologies built around what is being termed software-defined data centers (SDDCs). This expansion on elastic and cloud computing goes beyond the management of fixed pools of physical resources as virtual resources to include the automated deployment and provisioning of features and capabilities onto physical resources. For example, automated deployment tools that interface with virtualization or other framework APIs can be used to automatically stand up entire clusters or to add additional physical resources to physical or virtual clusters.

4.5 Data Consumer

Similar to the Data Provider, the role of Data Consumer within the NBDRA can be an actual end user or another system. In many ways, this role is the mirror image of the Data Provider, with the entire Big Data framework appearing like a Data Provider to the Data Consumer. The activities associated with the Data Consumer role include the following:

- Search and Retrieve
- Download
- Analyze Locally
- Reporting
- Visualization
- Data to Use for Their Own Processes

The Data Consumer uses the interfaces or services provided by the Big Data Application Provider to get access to the information of interest. These interfaces can include data reporting, data retrieval, and data rendering.

This role will generally interact with the Big Data Application Provider through its access function to execute the analytics
and visualizations implemented by the Big Data Application Provider. This interaction may be demand-based, where the Data Consumer initiates the command/transaction and the Big Data Application Provider replies with the answer. The interaction could include interactive visualizations, creating reports, or drilling down through data using business intelligence functions provided by the Big Data Application Provider. Alternately, the interaction may be stream- or push-based, where the Data Consumer simply subscribes or listens for one or more automated outputs from the application. In almost all cases, the Security and Privacy fabric around the Big Data architecture would support the authentication and authorization between the Data Consumer and the architecture, with either side able to perform the role of authenticator/authorizer and the other side providing the credentials. Like the interface between the Big Data architecture and the Data Provider, the interface between the Data Consumer and Big Data Application Provider would also pass through the three distinct phases of initiation, data transfer, and termination.

5 Management Fabric of the NBDRA

The Big Data characteristics of volume, velocity, variety, and variability demand a versatile management platform for storing, processing, and managing complex data. Management of Big Data systems should handle both system and data related aspects of the Big Data environment. The Management Fabric of the NBDRA encompasses two general groups of activities: system management and Big Data lifecycle management. System management includes activities such as provisioning, configuration, package management, software management, backup management, capability management, resources management, and performance management. Big Data lifecycle management involves activities surrounding the data lifecycle of collection, preparation/curation, analytics, visualization, and access.

As discussed above, the NBDRA represents a broad range of Big Data systems—from tightly-coupled enterprise solutions integrated by standard or proprietary interfaces to loosely-coupled vertical systems maintained by a variety of stakeholders or authorities bound by agreements, standard interfaces, or de facto standard interfaces. Therefore, different considerations and technical solutions would be applicable for different cases.

5.1 System Management

The characteristics of Big Data pose system management challenges on traditional management platforms. To efficiently capture, store, process, analyze, and distribute complex and large datasets arriving or leaving with high velocity, a resilient system management is needed.

As in traditional systems, system management for Big Data architecture involves provisioning, configuration, package management, software management, backup management, capability management, resources management, and performance management of the Big Data infrastructure, including compute nodes, storage nodes, and network devices. Due to the distributed and complex nature of the Big Data infrastructure, system management for Big Data is challenging, especially with respect to the capability for controlling, scheduling, and managing the processing frameworks to perform the scalable, robust, and secure analytics processing required by the Big Data Application Provider. The Big Data infrastructure may contain SAN or NAS storage devices, cloud storage spaces, NoSQL databases, M/R clusters, data analytics functions, search and indexing engines, and messaging platforms. The supporting enterprise computing infrastructure can range from traditional data centers, cloud services, and dispersed computing nodes of a grid. To manage the distributed and complex nature of the Big Data infrastructure, system management relies on the following:
Standard protocols such as SNMP which are used to transmit status about resources and fault information to the management fabric components

Deployable agents or management connectors which allow the management fabric to both monitor and also control elements of the framework.

These two items aid in monitoring the health of various types of computing resources and coping with performance and failures incidents while maintaining the quality of service (QoS) levels required by the Big Data Application Provider. Management connectors are necessary for scenarios where the cloud service providers expose management capabilities via APIs. It is conceivable that the infrastructure elements contain autonomic, self-tuning, and self-healing capabilities, thereby reducing the centralized model of system management. In large infrastructures with many thousands of computing and storage nodes, the provisioning of tools and applications should be as automated as possible. Software installation, application configuration, and regular patch maintenance should be pushed out and replicated across the nodes in an automated fashion, which could be done based on the topology knowledge of the infrastructure. With the advent of virtualization, the utilization of virtual images may speed up the recovery process and provide efficient patching that can minimize downtime for scheduled maintenance.

In an enterprise environment, the management platform would typically provide enterprise-wide monitoring and administration of the Big Data distributed components. This includes network management, fault management, configuration management, system accounting, performance management, and security management.

In a loosely-coupled vertical system, each independent stakeholder is responsible for its own system management, security, and integration. Each stakeholder is responsible for integration within the Big Data distributed system using the interfaces provided by other stakeholders.

5.2 Big Data Lifecycle Management

Big Data lifecycle management (BDLM) faces more challenges compared to traditional data lifecycle management (DLM), which may require less data transfer, processing, and storage. However, BDLM still inherits the DLM phases in terms of data acquisition, distribution, use, migration, maintenance, and disposition—but at a much bigger processing scale. Big Data Application Providers may require much more computational processing for collection, preparation/curation, analytics, visualization, and access to be able to use the analytic results. In other words, the BDLM activity includes verification that the data are handled correctly by other NBDRA components in each process within the data lifecycle—from the moment they are ingested into the system by the Data Provider, until the data are processed or removed from the system.

The importance of BDLM to Big Data is demonstrated through the following considerations:

- Data volume can be extremely large that may overwhelm the storage capacity, or make storing incoming data prohibitively expensive
- Data velocity, the rate at which data can be captured and ingested into the system, can overwhelm available storage space at a given time. Even with the elastic storage service provided by cloud computing for handling dynamic storage needs, uncontrolled data management may also be unnecessarily costly for certain application requirements.
Different Big Data applications will likely have different requirements for the lifetime of a piece of data. The differing requirements have implications on how often data must be refreshed so that processing results are valid and useful. In data refreshment, old data are dispositioned and not fed into analytics or discovery programs. At the same time, new data is ingested and taken into account by the computations. For example, real-time applications will need very short data lifetime but a market study of consumers' interest in a product line may need to mine data collected over a longer period of time.

Because the task of BDLM can be distributed among different organizations and/or individuals within the Big Data computing environment, coordination of data processing between NBDRA components has greater difficulty in complying with policies, regulations, and security requirements. Within this context, BDLM may need to include the following subactivities:

- **Policy Management**: Captures the requirements for the data lifecycle that allows old data to be dispositioned and new data to be considered by Big Data applications; maintains the migration and disposition strategies that specify the mechanism for data transformation and dispositioning, including transcoding data, transferring old data to lower tier storage for archival purpose, removing data, or marking data as in situ.

- **Metadata Management**: Enables BDLM, since metadata are used to store information that governs the management of the data within the system. Essential metadata information includes persistent identification of the data, fixity/quality, and access rights. The challenge is to find the minimum set of elements to execute the required BDLM strategy in an efficient manner.

- **Accessibility Management**: Change of data accessibility over time: For example, census data can be made available to the public after 72 years. BDLM is responsible for triggering the accessibility update of the data or sets of data according to policy and legal requirements. Normally, data accessibility information is stored in the metadata.

- **Data Recovery**: BDLM can include the recovery of data that were lost due to disaster or system/storage fault. Traditionally, data recovery can be achieved using regular backup and restore mechanisms. However, given the large volume of Big Data, traditional backup may not be feasible. Instead, replication may have to be designed within the Big Data ecosystem. Depending on the tolerance of data loss, each application has its own tolerance level. Replication strategies have to be designed. The replication strategy includes the replication window time, the selected data to be replicated, and the requirements for geographic disparity. Additionally, in order to cope with the large volume of Big Data, data backup and recovery should consider the use of modern technologies within the Big Data Framework Provider.

- **Preservation Management**: The system maintains data integrity so that the veracity and velocity of the analytics process are fulfilled. Due to the extremely large volume of Big Data, preservation management is responsible for disposition-aged data contained in the system. Depending on the retention policy, these aged data can be deleted or migrated to archival storage. In the case where data must be retained for years, decades, and even centuries, a preservation strategy will be needed so the data can be accessed by the provider components if required. This will invoke long-term digital preservation that can be performed by Big Data Application Providers using the resources of the Big Data Framework Provider.

In the context of Big Data, BDLM contends with the Big Data characteristics of volume, velocity, variety, and variability.
As such, BDLM and its subactivities interact with other components of the NBDRA as shown in the following examples:

- **System Orchestrator**: BDLM enables data scientists to initiate any combination of processing including accessibility management, data backup/recovery, and preservation management. The process may involve other components of the NBDRA, such as Big Data Application Provider and Big Data Framework Provider. For example, data scientists may want to interact with the Big Data Application Provider for data collection and curation, invoke the Big Data Framework Provider to perform certain analysis, and grant access to certain users to access the analytic results from the Data Consumer.

- **Data Provider**: BDLM manages ingestion of data and metadata from the data source(s) into the Big Data system, which may include logging the entry event in the metadata by the Data Provider.

- **Big Data Application Provider**: BDLM executes data masking and format transformations for data preparation or curation purpose.

- **Big Data Framework Provider**: BDLM executes basic bit-level preservation and data backup and recovery according to the recovery strategy.

- **Data Consumer**: BDLM ensures that relevant data and analytic results are available with proper access control for consumers and software agents to consume within the BDLM policy strategy.

- **Security and Privacy Fabric**: Keeps the BDLM up to date according to new security policy and regulations.

The Security and Privacy Fabric also uses information coming from BDLM with respect to data accessibility. The Security and Privacy Fabric control access to the functions and data usage produced by the Big Data system. This data access control can be informed by the can use metadata, which is managed and updated by BDLM.

### 6 Security and Privacy Fabric of the NBDRA

Security and privacy considerations form a fundamental aspect of the NBDRA. This is geometrically depicted in Figure 2 by the Security and Privacy Fabric surrounding the five main components, indicating that all components are affected by security and privacy considerations. Thus, the role of security and privacy is correctly depicted in relation to the components but does not expand into finer details, which may be more accurate but are best relegated to a more detailed security and privacy reference architecture. The Data Provider and Data Consumer are included in the Security and Privacy Fabric since, at the least, they may often nominally agree on security protocols and mechanisms. The Security and Privacy Fabric is an approximate representation that alludes to the intricate interconnected nature and ubiquity of security and privacy throughout the NBDRA. Additional details about the Security and Privacy Fabric are included in the *NIST Interoperability Framework: Volume 4, Security and Privacy* document.

### 7 Conclusion

The NBD-PWG Reference Architecture Subgroup prepared this *NIST Big Data Interoperability Framework: Volume 6, Reference Architecture* to provide a vendor-neutral, technology- and infrastructure-agnostic conceptual model and examine related issues. The conceptual model, referred to as the NIST Big Data Reference Architecture (NBDRA), was a collaborative effort within the Subgroup and with the other NBD-PWG subgroups. The goal of the NBD-PWG...
Reference Architecture Subgroup is to develop an open reference architecture for Big Data that achieves the following objectives:

- Provides a common language for the various stakeholders
- Encourages adherence to common standards, specifications, and patterns
- Provides consistent methods for implementation of technology to solve similar problem sets
- Illustrates and improves understanding of the various Big Data components, processes, and systems, in the context of a vendor- and technology- agnostic Big Data conceptual model
- Provides a technical reference for U.S. government departments, agencies, and other consumers to understand, discuss, categorize, and compare Big Data solutions
- Facilitates analysis of candidate standards for interoperability, portability, reusability, and extendibility

This document presents the results of the first stage of NBDRA development. The NIST Big Data Interoperability Framework will be released in three versions, which correspond to the three stages of the NBD-PWG work. The three stages aim to achieve the following:

Stage 1: Identify the common reference architecture components of Big Data implementations and formulate the technology-independent NBDRA

Stage 2: Define general interfaces between the NBDRA components

Stage 3: Validate the NBDRA by building Big Data general applications through the general interfaces

This document (Version 1) presents the overall NBDRA components and fabrics with high-level description and functionalities.

Version 2 activities will focus on the definition of general interfaces between the NBDRA components by achieving the following:

- Select use cases from the 62 (51 general and 11 security and privacy) submitted use cases or other, to be identified, meaningful use cases
- Work with domain experts to identify workflow and interactions from the System Orchestrator to the rest of the NBDRA components
- Implement small scale, manageable, and well-defined confined environment and model interaction between NBDRA components and fabrics
- Aggregate the common data workflow and interaction between NBDRA components/fabrics and package them into general interfaces

Version 3 activities will focus on validation of the NBDRA through the use of the defined NBDRA interfaces to build
general Big Data applications. The validation strategy will include the following:

- Implement the same set of use cases used in Version 2 by using the defined general interfaces
- Identify and implement a few new use cases outside the Version 2 scenarios
- Enhance general NBDRA interfaces through lessons learned from the implementations in Version 3 activities

The general interfaces developed during Version 3 activities will offer a starting point for further refinement by any interested parties and is not intended to be a definitive solution to address all implementation needs.

Appendix A: Deployment Considerations

Introduction

The NIST Big Data Reference Architecture is applicable to a variety of business environments and technologies. As a result, possible deployment models are not part of the core concepts discussed in the main body of this document. However, the loosely coupled and distributed natures of Big Data Framework Provider functional components allows it to be deployed using multiple infrastructure elements as described in Section 4.4.1. The two most common deployment configurations are directly on physical resources or on top of an IaaS cloud computing framework. The choices between these two configurations are driven by needs of efficiency/performance and elasticity. Physical infrastructures are typically used to obtain predictable performance and efficient utilization of CPU and I/O bandwidth since it eliminates the overhead and additional abstraction layers typical in the virtualized environments for most IaaS implementations. IaaS cloud based deployments on are typically used when elasticity is needed to support changes in workload requirements. The ability to rapidly instantiate additional processing nodes or framework components allows the deployment to adapt to either increased or decreased workloads. By allowing the deployment footprint to grow or shrink based on workload demands this deployment model can provide cost savings when public or shared cloud services are used and more efficient use and energy consumption when a private cloud deployment is used. Recently, a hybrid deployment model known as Cloud Bursting has become popular. In this model a physical deployment is augmented by either public or private IaaS cloud services. When additional processing is needed to support the workload additional the additional framework component instances are established on the IaaS infrastructure and then deleted when no longer require.
In addition to providing IaaS support cloud providers are now offering Big Data Frameworks under a PaaS model. Under this model the system implementer is freed from the need to establish and manage the complex configuration and deployment typical of many Big Data Framework components. The implementer simply needs to specify the size of the cluster required and the cloud provider manages the provisioning, configuration, and deployment of all the framework components. There are even some nascent offerings for specialized SaaS Big Data applications appearing in the market that implement the Big Data Application Provider functionality within the cloud environment. Figure A-1 illustrates how the components of the NBDRA might align onto the NIST Cloud Reference architecture[7]. The following sections describe some of the high level interactions required between the Big Data Architecture elements and the Cloud Service Provider elements.

Cloud Service Providers

Recent data analytics solutions use algorithms that can utilize and benefit from the frameworks of the cloud computing systems. Cloud computing has essential characteristics such as rapid elasticity and scalability, multi-tenancy, on-demand self-service and resource pooling, which together can significantly lower the barriers to the realization of Big Data implementations.

The Cloud Service Provider (CSP) implements and delivers cloud services. Processing of a service invocation is done by means of an instance of the service implementation, which may involve the composition and invocation of other services as determined by the design and configuration of the service implementation.

Cloud Service Component

The cloud service component contains the implementation of the cloud services provided by a CSP. It contains and controls the software components that implement the services (but not the underlying hypervisors, host operating systems, device drivers, etc.)

Cloud services can be described in terms of service categories.

Cloud services are also grouped into categories, where each service category is characterized by qualities that are
common between the services within the category. The NIST Cloud Computing Reference Model defines the following cloud service categories:

- Infrastructure as a Services (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

**Resource Abstraction and Control Component**

The Resource Abstraction and Control component is used by cloud service providers to provide access to the physical computing resources through software abstraction. Resource abstraction needs to assure efficient, secure, and reliable usage of the underlying physical resources. The control feature of the component enables the management of the resource abstraction features.

The Resource Abstraction and Control component enables a cloud service provider to offer qualities such as rapid elasticity, resource pooling, on-demand self-service and scale-out. The Resource Abstraction and Control component can include software elements such as hypervisors, virtual machines, virtual data storage, and time-sharing.

The Resource Abstraction and Control component enables control functionality. For example, there may be a centralized algorithm to control, correlate, and connect various processing, storage, and networking units in the physical resources so that together they deliver an environment where IaaS, PaaS or SaaS cloud service categories can be offered. The controller might decide which CPUs/racks contain which virtual machines executing which parts of a given cloud workload, and how such processing units are connected to each other, and when to dynamically and transparently reassign parts of the workload to new units as conditions change.

**Security and Privacy and Management Functions**

In almost all cases the Cloud Provider will provide elements of the Security, Privacy, and Management functions. Typically the provider will support high level security/privacy functions that control access to the Big Data Applications and Frameworks while the frameworks themselves must control access to their underlying data and application services. Many times the Big Data specific functions for security and privacy will depend on and must interface with functions provided by the cloud service provider. Similarly, management functions are often split between the Big Data implementation and the Cloud Provider implementations. Here the cloud provider would handle the deployment and provisioning of Big Data architecture elements within its IaaS infrastructure. The cloud provider may provide high level monitoring functions to allow the Big Data implementation to track performance and resource usage of its components. In, many cases the Resource Management element of the Big Data Framework will need to interface to the Cloud Service Provider’s management framework to request additional resources.

**Physical Resource Deployments**

As stated above deployment on physical resources is frequently used when performance characteristics are paramount. The nature of the underlying physical resource implementations to support Big Data requirements has evolved significantly over the years. Specialized, high performance super computers with custom approaches for sharing resources (memory, CPU, storage) between nodes has given way to shared nothing computing clusters built from
commodity servers. The custom super computing architectures almost always required custom development and components to take advantage of the shared resources. The commodity server approach both reduced the hardware investment and allowed the Big Data frameworks to provide higher level abstractions for the sharing and management of resources in the cluster. The Recent trends now involve density, power, cooling optimized server form factors that seek to maximize the available computing resources while minimizing size, power and/or cooling requirements. This approach retains the abstraction and portability advantages of the shared nothing approaches while providing improved efficiency.

Appendix B: Terms and Definitions

NBDRA Components

- **Big Data Engineering**: Advanced techniques that harness independent resources for building scalable data systems when the characteristics of the datasets require new architectures for efficient storage, manipulation, and analysis.

- **Data Provider**: Organization or entity that introduces information feeds into the Big Data system for discovery, access, and transformation by the Big Data system.

- **Big Data Application Provider**: Organization or entity that executes a generic vertical system data lifecycle, including: (a) data collection from various sources, (b) multiple data transformations being implemented using both traditional and new technologies, (c) diverse data usage, and (d) data archiving.

- **Big Data Framework Provider**: Organization or entity that provides a computing fabric (such as system hardware, network, storage, virtualization, and computing platform) to execute certain Big Data applications, while maintaining security and privacy requirements

- **Data Consumer**: End users or other systems that use the results of data applications.

- **System Orchestrator**: Organization or entity that defines and integrates the required data transformations components into an operational vertical system.

Operational Characteristics

- **Interoperability**: The capability to communicate, to execute programs, or to transfer data among various functional units under specified conditions.

- **Portability**: The ability to transfer data from one system to another without being required to recreate or reenter data descriptions or to modify significantly the application being transported.

- **Privacy**: The assured, proper, and consistent collection, processing, communication, use and disposition of data associated with personal information (PI) and personally-identifiable information (PII) throughout its lifecycle.

- **Security**: Protecting data, information, and systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide:
  o **Integrity**: guarding against improper data modification or destruction, and includes ensuring data nonrepudiation and
authenticity

- Confidentiality: preserving authorized restrictions on access and disclosure, including means for protecting personal privacy and proprietary data

- Availability: ensuring timely and reliable access to and use of data

- Elasticity: The ability to dynamically scale up and down as a real time response to the workload demand. Elasticity will depend on the Big Data system, but adding or removing "software threads" and "virtual or physical servers" are two widely used scaling techniques. Many types of workload demands drive elastic responses, including Web based users, software agents, and periodic batch jobs.

- Persistence: The placement/storage of data in a medium design to allow its future access.

Provisioning Models

- Infrastructure as a Service (IaaS): The capability provided to the consumer to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls).

- Platform as a Service (PaaS): The capability provided to the consumer to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations. (Source: NIST CC Definition)

- Software as a Service (SaaS): The capability provided to the consumer to use applications running on a cloud infrastructure. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings. (Source: NIST CC Definition)

Appendix C: Examples of Big Data Organization Approaches

This appendix provides an overview of several common Big Data Organization Approaches. The reader should keep in mind that new and innovative approaches are emerging regularly and that some of these approaches are hybrid models that combine features of several indexing techniques (e.g., relational and columnar, or relational and graph).

Relational Storage Models

This model is perhaps the most familiar to folks as the basic concept has existed since the 1950s and the Structured Query Language (SQL) is a mature standard for manipulating (search, insert, update, delete) relational data. In the relational model, data is stored as rows with each field representing a column organized into Table based on the logical data organization. The problem with relational storage models and Big Data is the join between one or more tables.
While the size of 2 or more tables of data individually might be small the join (or relational matches) between those tables will generate exponentially more records. The appeal of this model for organizations just adopting Big Data is its familiarity. The pit falls are some of the limitations and more importantly the tendency to adopt standard RDBMS practices (high normalization, detailed and specific indexes) and performance expectations

Big data implementations of relational storage models are relatively mature and have been adopted by a number of organizations. They are also maturing very rapidly with new implementations focusing on improved response time. Many Big Data implementations take a brute force approach to scaling relational queries. Essentially, queries are broken into stages but more importantly processing of the input tables is distributed across multiple nodes (often as a Map/Reduce job). The actual storage of the data can be flat files (delimited or fixed length) where each record/line in the file represents a row in a table. Increasingly however these implementations are adopting binary storage formats optimized for distributed file systems. These formats will often use block level indexes and column oriented organization of the data to allow individual fields to be accessed in records without needing to read the entire record. Despite this, most Big Data Relational storage models are still “batch oriented” systems designed for very complex queries which generate very large intermediate cross-product matrices from joins so even the simplest query can required 10s of seconds to complete. There is significant work going on and emerging implementations that are seeking to provide a more interactive response and interface.

Early implementations only provided limited data types and little or no support for indexes. However, most current implementations have support for complex data structures and basic indexes. However, while the query planners/optimizers for most modern RDBMS systems are very mature and implement cost-based optimization through statistics on the data the query planners/optimizers in many Big Data implementations remain fairly simple and rule-based in nature. While for batch oriented systems this generally acceptable (since the scale of processing the Big Data in general can be orders of magnitude more an impact) any attempt to provide interactive response will need very advanced optimizations so that (at least for queries) only the most likely data to be returned is actually searched. This of course leads to the single most serious draw back with many of these implementations. Since distributed processing and storage are essential for achieving scalability these implementations are directly limited by the CAP (Consistency, Availability, and Partition Tolerance) theorem. Many in fact provide what is generally referred to as a t-eventual consistency which means that barring any updates to a piece of data all nodes in the distributed system will eventually return the most recent value. This level of consistency is typically fine for Data Warehousing applications where data is infrequently updated and updates are generally done in bulk. However, transaction oriented databases typically require some level of ACID (Atomicity, Consistency, Isolation, Durability) compliance to insure that all transactions are handled reliably and conflicts are resolved in a consistent manner. There are a number of both industry and open source initiatives looking to bring this type of capability to Big Data relational storage frameworks. One approach is to essentially layer a traditional RDBMS on top of an existing distributed file system implementation. While vendors claim that this approach means that the overall technology is mature a great deal of research and implementation experience is needed before the complete performance characteristics of these implementations are known.

**Key-Value Storage Models**

Key-value stores are one of the oldest and mature data indexing models. In fact, the principles of key value stores underpin all the other storage and indexing models. From a Big Data perspective, these stores effectively represent random access memory models. While the data stored in the values can be arbitrarily complex in structure all the handling of that complexity must be provided by the application with the storage implementation often providing back
just a pointer to a block of data. Key-Value stores also tend to work best for 1-1 relationships (e.g., each key relates to a single value) but can also be effective for keys mapping to lists of homogeneous values. When keys map multiple values of heterogeneous types/structures or when values from one key need to be joined against values for a different or the same key then custom application logic is required. It is the requirement for this custom logic that often prevents Key-Value stores from scaling effectively for certain problems. However, depending on the problem certain processing architectures can make effective use of distributed key-value stores. Key-value stores generally deal well with updates when the mapping is one to one and the size/length of the value data does not change. The ability of key-value stores to handle inserts is generally dependent on the underlying implementation. Key-value stores also generally require significant effort (either manual or computational) to deal with changes to the underlying data structure of the values.

Distributed key-value stores are the most frequent implementation utilized in Big Data applications. One problem that must always be addressed (but is not unique to key-value implementations) is the distribution of keys across over the space of possible key values. Specifically, keys must be chosen carefully to avoid skew in the distribution of the data across the cluster. When data is heavily skewed to a small range it can result in computation hot spots across the cluster if the implementation is attempting to optimize data locality. If the data is dynamic (new keys being added) for such an implementation then it is likely that at some point the data will require rebalancing across the cluster. Non-locality optimizing implementations employ various sorts of hashing, random, or round-robin approaches to data distribution and don’t tend to suffer from skew and hot spots. However, they perform especially poorly on problems requiring aggregation across the data set.

**Columnar Storage Models**

Much of the hype associated with Big Data came with the publication of the Big Table paper in 2006 (Chang, et al., 2006) but column oriented storage models like BigTable are not new to even Big Data and have been stalwarts of the data warehousing domain for many years. Unlike traditional relational data that store data by rows of related values, columnar stores organize data in groups of like values. The difference here is subtle but in relational databases an entire group of columns are tied to some primary-key (frequently one or more of the columns) to create a record. In columnar, the value of every column is a key and like column values point to the associated rows. The simplest instance of a columnar store is little more than a key-value store with the key and value roles reversed. In many ways, columnar data stores look very similar to indexes in relational databases. Figure 5 below shows the basic differences between row oriented and column oriented stores.

**Figure B-1: Differences Between Row Oriented and Column Oriented Stores**

![Figure B-1: Differences Between Row Oriented and Column Oriented Stores](https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework)
In addition, implementations of columnar stores that follow the BigTable model introduce an additional level of segmentation beyond the table, row, and column model of the relational model. That is called the column family. In those implementations rows have a fixed set of column families but within a column family each row can have a variable set of columns. This is illustrated in Figure 6 below.

**Figure B-2: Column Family Segmentation of the Columnar Stores Model**

The key distinction in the implementation of columnar store over relational stores is that data is high de-normalized for column stores and that while for relational stores every record contains some value (perhaps NULL) for each column, in columnar store the column is only present if there is data for one or more rows. This is why many column oriented stores are referred to as sparse storage models. Data for each column family is physically stored together on disk sorted by rowed, column name and timestamp. The last (timestamp) is there because the BigTable model also includes the concept of versioning. Every, RowKey, Column Family, Column triple is stored with either a system generated or user provided Timestamp. This allows users to quickly retrieve the most recent value for a column (the default), the specific value for a column by timestamp, or all values for a column. The last is most useful because it permits very rapid temporal analysis on data in a column.

Because data for a given column is stored together two key benefits are achieved. First aggregation of the data in that column requires only the values for that column to be read. Conversely in a relational system the entire row (at least up to the column) needs to be read (which if the row is long and the column at the end it could be lots of data). Secondly, updates to a single column do not require the data for the rest of the row to be read/written. Also, because all the data in a column is uniform, data can be compressed much more efficiently. Often only a single copy of the value for a column is stored followed by the row keys where that value exists. And while deletes of an entire column is very efficient, deletes of an entire record are extremely expensive. This is why historically column oriented stores have been applied to OLAP style applications while relational stores were applied to OLTP requirements.

Recently, security has been a major focus of existing column implementations primarily due to the release by the National Security Agency (NSA) of its BigTable implementation to the open source community. A key advantage of the NSA implementation and other recently announced implementations is the availability of security controls at the individual cell level. With these implementations a given user might have access to only certain cells in group based potentially based on the value of those or other cells.

There are several very mature distributed column oriented implementations available today from both open source groups and commercial foundations. These have been implemented and operational across a wide range of businesses and government organizations. Emerging are hybrid capabilities that implement relational access methods (e.g., SQL) on top of BigTable/Columnar storage models. In addition, relational implementations are adopting columnar oriented...
physical storage models to provide more efficient access for Big Data OLAP like aggregations and analytics.

Document

Document storage approaches have been around for some time and popularized by the need to quickly search large amounts of unstructured data. Modern, document stores have evolved to include extensive search and indexing capabilities for structured data and metadata and why they are often referred to as semi-structured data stores. Within a document-oriented data store each “document” encapsulates and encodes the metadata, fields, and any other representations of that record. While somewhat analogous to a row in a relational table one-reason document stores evolved and have gained in popularity is that most implementations do not enforce a fixed or constant schema. While best practices hold that groups of documents should be logically related and contain similar data there is no requirement that they be alike or that any two documents even contain the same fields. That is one reason that document stores are frequently popular for data sets which have sparsely populated fields since there is far less overhead normally than traditional RDBMS systems where null value columns in records are actually stored. Groups of documents within these types of stores are generally referred to as collections and like key-value stores some sort of unique key references each document.

In modern implementations documents can be built of arbitrarily nested structures and can include variable length arrays and in some cases executable scripts/code (which has significant security and privacy implications). Most document-store implementations also support additional indexes on other fields or properties within each document with many implementing specialized index types for sparse data, geospatial data, and text.

When modeling data into document-stores the preferred approach is to de-normalize the data as much as possible and embed all one-to-one and most one-to-many relationships within a single document. This allows for updates to documents to be atomic operations which keep referential integrity between the documents. The most common case where references between documents should be use is when there are data elements that occur frequently across sets of documents and whose relationship to those documents is static. As an example, the publisher of a given book edition does not change and there are far fewer publishers than there are books. It would not make sense to embed all the publisher information into each book document. Rather the book document would contain a reference to the unique key for the publisher. Since for that edition of the book the reference will never change and so there is no danger of loss of referential integrity. Thus information about the publisher (address for example) can be updated in a single atomic operation the same as the book. Where this information embedded it would need to be updated in every book document with that publisher.

In the Big Data realm document stores scale horizontally through the use of partitioning or sharding to distribute portions of the collection across multiple nodes. This partitioning can be round-robin based insuring an even distribution of data or content/key based so that data locality is maintained for similar data. Depending on the application required the choice of partitioning key like with any data base can have significant impacts on performance especially where aggregation functions are concerned.

There are no standard query languages for document store implementations with most using a language derived from their internal document representation (e.g., JSON, XML).
Graph

While social networking sites like Facebook and LinkedIn have certainly driven the visibility of and evolution of graph stores (and processing as discussed below), graph stores have been a critical part of many problem domains from military intelligence and counter terrorism to route planning/navigation and the semantic web for years. Graph stores represent data as a series of nodes, edges, and properties on those. Analytics against graph stores include very basic shortest path and page ranking to entity disambiguation and graph matching.

Graph databases typically store two types of objects nodes and relationships as show in Figure 7 below. Nodes represent objects in the problem domain that are being analyzed be they people, places, organizations, accounts, or other objects. Relationships describe those objects in the domain relate to each other. Relationships can be nondirectional/bidirectional but are typically expressed as unidirectional in order to provide more richness and expressiveness to the relationships. Hence, between two people nodes where they are father and son there would be two relationships. One "is father of" going from the father node to the son node, and the other from the son to the father of "is son of". In addition, nodes and relationships can have properties or attributes. This is typically descriptive data about the element. For people it might be name, birthdate, or other descriptive quality. For locations it might be an address or geospatial coordinate. For a relationship like a phone call it could be the date, time of the call, and the duration of the call. Within graphs relationships are not always equal or have the same strength. Thus relationship often has one or more weight, cost, or confidence attributes. A strong relationship between people might have a high weight because they have known each other for years and communicate every day. A relationship where two people just met would have a low weight. The distance between nodes (be it a physical distance or a difficulty) is often expressed as a cost attribute on a relation in order to allow computation of true shortest paths across a graph. In military intelligence applications, relationships between nodes in a terrorist or command and control network might only be suspected or have not been completely verified so those relationships would have confidence attributes. Also, properties on nodes may also have confidence factors associated with them although in those cases the property can be decomposed into its own node and tied with a relationship. Graph storage approaches can actually be viewed as a specialized implementation of a document storage scheme with two types of documents (nodes and relationships). In addition, one of the most critical elements in analyzing graph data is locating the node or edge in the graph where the analysis is to begin. To accomplish this most graph databases implement indexes on the node or edge properties. Unlike, relational and other data storage approaches most graph databases tend to use artificial/pseudo keys or guides to uniquely identify nodes and edges. This allows attributes/properties to be easily changed due to both actual changes in the data (someone changed their name) or as more information is found out (e.g., a better location for some item or event) without needing to change the pointers two/from relationships.
The problem with graphs in the Big Data realm is that they grow to be too big to fit into memory on a single node and by their typically chaotic nature (few real world graphs follow well defined patterns) makes their partitioning for a distributed implementation problematic. While distance between or closeness of nodes would seem like a straightforward partitioning approach there are multiple issues which must be addressed. First, would be balancing of data. Graphs often tend to have large clusters of data very dense in a given area thus leading to essentially imbalances and hot spots in processing. Second, no matter how the graph is distributed, there are connections (edges) that will cross the boundaries. That typically requires that nodes know about or how to access the data on other nodes and requires inter-node data transfer or communication. This makes the choice of processing architectures for graph data especially critical. Architectures that do not have inter-node communication/messaging tend not to work well for most graph problems. Typically, distributed architectures for processing graphs assign chunks of the graph to nodes then the nodes use messaging approaches to communicate changes in the graph or the value of certain calculations along a path.

Even small graphs quickly elevate into the realm of Big Data when one is looking for patterns or distances across more than one or two degrees of separation between nodes. Depending on the density of the graph, this can quickly cause a combinatorial explosion in the number of conditions/patterns that need to be tested.

A specialized implementation of a graph store known as the Resource Description Framework (RDF) is part of a family
of specifications from the World Wide Web Consortium (W3C) that is often directly associated with Semantic Web and associated concepts. RDF triples as they are known consist of a Subject (Mr. X), a predicate (lives at), and an object (Mockingbird Lane). Thus a collection of RDF triples represents and directed labeled graph. The contents of RDF stores are frequently described using formal ontology languages like OWL or the RDF Schema (RDFS) language, which establish the semantic meanings and models of the underlying data. To support better horizontal integration (Smith, Malyuta, Mandirck, Fu, Parent, & Patel, 2012) of heterogeneous data sets extensions to the RDF concept such as the Data Description Framework (DDF) (Yoakum-Stover & Malyuta, 2008) have been proposed which add additional types to better support semantic interoperability and analysis.

Graph data stores currently lack any form of standardized APIs or query languages. However, the W3C has developed the SPARQL query language for RDF which is currently in a recommendation status and there are several frameworks such as Sesame which are gaining popularity for working with RDF and other graph oriented data stores.

---

**Appendix D: Acronyms**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>APIs</td>
<td>application programming interface</td>
</tr>
<tr>
<td>CIA</td>
<td>confidentiality, integrity, and availability</td>
</tr>
<tr>
<td>CRUD</td>
<td>create/read/update/delete</td>
</tr>
<tr>
<td>CSP</td>
<td>Cloud Service Provider</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name Server</td>
</tr>
<tr>
<td>GRC</td>
<td>governance, risk, and compliance</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transfer Protocol</td>
</tr>
<tr>
<td>I/O</td>
<td>Input/Output Processing</td>
</tr>
<tr>
<td>IaaS</td>
<td>Infrastructure as a Service</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MAN</td>
<td>Metropolitan Area Network</td>
</tr>
<tr>
<td>NaaS</td>
<td>Network as a Service</td>
</tr>
<tr>
<td>NBD-PWG</td>
<td>NIST Big Data Public Working Group</td>
</tr>
<tr>
<td>NBDRA</td>
<td>NIST Big Data Reference Architecture</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
</tbody>
</table>

---

https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework

Updated: Sat, 14 Sep 2019 05:26:25 GMT

Powered by mindtouch™
Appendix E: Resources and References

General Resources

The following resources provide additional information related to Big Data architecture.


Document References


Standards Roadmap

Source: http://bigdatawg.nist.gov/_uploadfil...449826642.docx (Word)

Cover Page

NIST Special Publication 1500-7

DRAFT NIST Big Data Interoperability Framework:

Volume 7, Standards Roadmap

NIST Big Data Public Working Group

Technology Roadmap Subgroup

Draft Version 1

April 6, 2015

http://dx.doi.org/10.6028/NIST.SP.1500-7

Inside Cover Page

NIST Special Publication 1500-7

Information Technology Laboratory

DRAFT NIST Big Data Interoperability Framework:

Volume 7, Standards Roadmap

Draft Version 1

NIST Big Data Public Working Group (NBD-PWG)

Technology Roadmap Subgroup

National Institute of Standards and Technology

Gaithersburg, MD 20899

April 2015

U. S. Department of Commerce

Penny Pritzker, Secretary
Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by Federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, Federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. All NIST publications are available at [http://www.nist.gov/publication-portal.cfm](http://www.nist.gov/publication-portal.cfm).

**Public comment period:** April 6, 2015 through May 21, 2015

**Comments on this publication may be submitted to Wo Chang**

National Institute of Standards and Technology

Attn: Wo Chang, Information Technology Laboratory

100 Bureau Drive (Mail Stop 8900) Gaithersburg, MD 20899-8930

Email: [SP1500comments@nist.gov](mailto:SP1500comments@nist.gov)

**Reports on Computer Systems Technology**

The Information Technology Laboratory (ITL) at NIST promotes the U.S. economy and public welfare by providing technical leadership for the Nation’s measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology (IT). ITL’s responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in Federal information systems. This document reports on ITL’s research, guidance, and outreach efforts in IT and its collaborative activities with industry, government, and academic organizations.
Abstract

Big Data is a term used to describe the large amount of data in the networked, digitized, sensor-laden, information-driven world. While opportunities exist with Big Data, the data can overwhelm traditional technical approaches and the growth of data is outpacing scientific and technological advances in data analytics. To advance progress in Big Data, the NIST Big Data Public Working Group (NBD-PWG) is working to develop consensus on important, fundamental concepts related to Big Data. The results are reported in the NIST Big Data Interoperability Framework series of volumes. This volume, Volume 7, contains summaries of the work presented in the other six volumes and an investigation of standards related to Big Data.

Keywords

Big Data, reference architecture, System Orchestrator, Data Provider, Big Data Application Provider, Big Data Framework Provider, Data Consumer, Security and Privacy Fabric, Management Fabric, Big Data taxonomy, use cases, Big Data characteristics, Big Data standards

Acknowledgements

This document reflects the contributions and discussions by the membership of the NBD-PWG, co-chaired by Wo Chang of the NIST ITL, Robert Marcus of ET-Strategies, and Chaitanya Baru, University of California San Diego Supercomputer Center.

The document contains input from members of the NBD-PWG: Technology Roadmap Subgroup, led by Carl Buffington (Vistronix), David Boyd (InCadence Strategic Solutions), and Dan McClary (Oracle); Definitions and Taxonomies Subgroup, led by Nancy Grady (SAIC), Natasha Balac (SDSC), and Eugene Luster (R2AD); Use Cases and Requirements Subgroup, led by Geoffrey Fox (University of Indiana) and Tsegereda Beyene (Cisco); Security and Privacy Subgroup, led by Arnab Roy (Fujitsu) and Akhil Manchanda (GE); and Reference Architecture Subgroup, led by Orit Levin (Microsoft), Don Krapohl (Augmented Intelligence), and James Ketner (AT&T).

NIST SP1500-7, Version 1 has been collaboratively authored by the NBD-PWG. As of the date of this publication, there are over six hundred NBD-PWG participants from industry, academia, and government. Federal agency participants include the National Archives and Records Administration (NARA), National Aeronautics and Space Administration (NASA), National Science Foundation (NSF), and the U.S. Departments of Agriculture, Commerce, Defense, Energy, Health and Human Services, Homeland Security, Transportation, Treasury, and Veterans Affairs.

NIST acknowledges the specific contributions to this volume by the following NBD-PWG members:

Chaitan Baru
University of California, San Diego, Supercomputer Center

Pavithra Kenjige
PK Technologies

Brenda Kirkpatrick

Shawn Miller
U.S. Department of Veterans Affairs

David Boyd
Hewlett-Packard

William Miller
InCadence Strategic Services

Carl Buffington
Vistronix
Luca Lepori

Pw Carey
Data Hold
Orit Levin

Compliance Partners, LLC
Wo Chang
Microsoft

NIST
Jan Levine

Yuri Demchenko
kloudtrack

University of Amsterdam
Serge Mankovski

Nancy Grady
CA Technologies

SAIC
Robert Marcus

Keith Hare
ET-Strategies

JCC Consulting, Inc.
Gary Mazzaferro

Bruno Kelpsas
AlloyCloud, Inc.

Microsoft Consultant

MaCT USA
Sanjay Mishra
Verizon
Quyen Nguyen
NARA
John Rogers
HP
Doug Scrimager
Slalom Consulting
Cherry Tom
IEEE-SA
Timothy Zimmerlin
Automation Technologies Inc.

The editors for this document were David Boyd, Carl Buffington, and Wo Chang

[a] “Contributors” are members of the NIST Big Data Public Working Group who dedicated great effort to prepare and substantial time on a regular basis to research and development in support of this document.

Notice to Readers

NIST is seeking feedback on the proposed working draft of the *NIST Big Data Interoperability Framework: Volume 7, Standards Roadmap*. Once public comments are received, compiled, and addressed by the NBD-PWG, and reviewed and approved by NIST internal editorial board, Version 1 of this volume will be published as final. Three versions are planned for this volume, with Versions 2 and 3 building on the first. Further explanation of the three planned versions and the information contained therein is included in Section 1.5 of this document.

Please be as specific as possible in any comments or edits to the text. Specific edits include, but are not limited to, changes in the current text, additional text further explaining a topic or explaining a new topic, additional references, or comments about the text, topics, or document organization. These specific edits can be recorded using one of the two
following methods.

1. **TRACK CHANGES**: make edits to and comments on the text directly into this Word document using track changes
2. **COMMENT TEMPLATE**: capture specific edits using the Comment Template ([http://bigdatawg.nist.gov/_uploadfiles/SP1500-1-to-7_comment_template.docx](http://bigdatawg.nist.gov/_uploadfiles/SP1500-1-to-7_comment_template.docx)), which includes space for Section number, page number, comment, and text edits

Submit the edited file from either method 1 or 2 to SPF0comment@nist.gov with the volume number in the subject line (e.g., Edits for Volume 7.)

Please contact Wo Chang (wchang@nist.gov) with any questions about the feedback submission process.

Big Data professionals continue to be welcome to join the NBD-PWG to help craft the work contained in the volumes of the NIST Big Data Interoperability Framework. Additional information about the NBD-PWG can be found at [http://bigdatawg.nist.gov](http://bigdatawg.nist.gov).

---

**Table of Contents**

Executive Summary. vii

1. Introduction. 1
   1.1 Background. 1
   1.2 NIST Big Data Public Working Group. 2
   1.3 Scope and Objectives of the Technology Roadmap Subgroup. 3
   1.4 Report Production. 3
   1.5 Future Work on this Volume. 3

2. Big Data Definition. 5
   2.1 Big Data Definitions. 5
   2.2 Data Science Definitions. 5

3. Investigating the Big Data Ecosystem. 7
   3.1 Use Cases. 7
   3.2 Reference Architecture Survey. 9
   3.3 Taxonomy. 9

4. Big Data Reference Architecture. 11
Executive Summary

To provide a common Big Data framework, the NIST Big Data Public Working Group (NBD-PWG) is creating vendor-neutral, technology- and infrastructure-agnostic deliverables, which include the development of consensus based definitions, taxonomies, reference architecture, and roadmap. This document, *NIST Interoperability Framework: Volume 7, Standards Roadmap*, summarizes the deliverables of the other NBD-PWG subgroups (presented in detail in the other volumes of this series) and presents the work of the NBD-PWG Technology Roadmap Subgroup. In the first phase of development, the NBD-PWG Technology Roadmap Subgroup investigated existing standards that relate to Big Data and recognized general categories of gaps in those standards.

The *NIST Big Data Interoperability Framework* consists of seven volumes, each of which addresses a specific key topic, resulting from the work of the NBD-PWG. The seven volumes are as follows:

- Volume 1, Definitions
- Volume 2, Taxonomies
The NIST Big Data Interoperability Framework will be released in three versions, which correspond to the three stages of the NBD-PWG work. The three stages aim to achieve the following with respect to the NIST Big Data Reference Architecture (NBDRA.)

Stage 1: Identify the high-level Big Data reference architecture key components, which are technology, infrastructure, and vendor agnostic

Stage 2: Define general interfaces between the NBDRA components

Stage 3: Validate the NBDRA by building Big Data general applications through the general interfaces

Potential areas of future work for the Subgroup during stage 2 are highlighted in Section 1.5 of this volume. The current effort documented in this volume reflects concepts developed within the rapidly evolving field of Big Data.

1 Introduction

1.1 Background

There is broad agreement among commercial, academic, and government leaders about the remarkable potential of Big Data to spark innovation, fuel commerce, and drive progress. Big Data is the common term used to describe the deluge of data in today’s networked, digitized, sensor-laden, and information-driven world. The availability of vast data resources carries the potential to answer questions previously out of reach, including the following:

- How can a potential pandemic reliably be detected early enough to intervene?
- Can new materials with advanced properties be predicted before these materials have ever been synthesized?
- How can the current advantage of the attacker over the defender in guarding against cyber-security threats be reversed?

There is also broad agreement on the ability of Big Data to overwhelm traditional approaches. The growth rates for data volumes, speeds, and complexity are outpacing scientific and technological advances in data analytics, management, transport, and data user spheres.

Despite widespread agreement on the inherent opportunities and current limitations of Big Data, a lack of consensus on some important, fundamental questions continues to confuse potential users and stymie progress. These questions include the following:
• What attributes define Big Data solutions?
• How is Big Data different from traditional data environments and related applications?
• What are the essential characteristics of Big Data environments?
• How do these environments integrate with currently deployed architectures?
• What are the central scientific, technological, and standardization challenges that need to be addressed to accelerate the deployment of robust Big Data solutions?

Within this context, on March 29, 2012, the White House announced the Big Data Research and Development Initiative.[1] The initiative’s goals include helping to accelerate the pace of discovery in science and engineering, strengthening national security, and transforming teaching and learning by improving the ability to extract knowledge and insights from large and complex collections of digital data.

Six federal departments and their agencies announced more than $200 million in commitments spread across more than 80 projects, which aim to significantly improve the tools and techniques needed to access, organize, and draw conclusions from huge volumes of digital data. The initiative also challenged industry, research universities, and nonprofits to join with the federal government to make the most of the opportunities created by Big Data.

Motivated by the White House initiative and public suggestions, the National Institute of Standards and Technology (NIST) has accepted the challenge to stimulate collaboration among industry professionals to further the secure and effective adoption of Big Data. As one result of NIST’s Cloud and Big Data Forum held on January 15–17, 2013, there was strong encouragement for NIST to create a public working group for the development of a Big Data Interoperability Framework. Forum participants noted that this roadmap should define and prioritize Big Data requirements, including interoperability, portability, reusability, extensibility, data usage, analytics, and technology infrastructure. In doing so, the roadmap would accelerate the adoption of the most secure and effective Big Data techniques and technology.

On June 19, 2013, the NIST Big Data Public Working Group (NBD-PWG) was launched with extensive participation by industry, academia, and government from across the nation. The scope of the NBD-PWG involves forming a community of interests from all sectors—including industry, academia, and government—with the goal of developing consensus on definitions, taxonomies, secure reference architectures, security and privacy requirements, and from these a standards roadmap. Such a consensus would create a vendor-neutral, technology- and infrastructure-independent framework that would enable Big Data stakeholders to identify and use the best analytics tools for their processing and visualization requirements on the most suitable computing platform and cluster, while also allowing value-added from Big Data service providers.

The NIST Big Data Interoperability Framework consists of seven volumes, each of which addresses a specific key topic, resulting from the work of the NBD-PWG. The seven volumes are as follows:

• Volume 1, Definitions
• Volume 2, Taxonomies
• Volume 3, Use Cases and General Requirements
• Volume 4, Security and Privacy
The NIST Big Data Interoperability Framework will be released in three versions, which correspond to the three stages of the NBD-PWG work. The three stages aim to achieve the following with respect to the NIST Big Data Reference Architecture (NBDRA.)

Stage 1: Identify the high-level Big Data reference architecture key components, which are technology, infrastructure, and vendor agnostic

Stage 2: Define general interfaces between the NBDRA components

Stage 3: Validate the NBDRA by building Big Data general applications through the general interfaces

The NBDRA, created in Stage 1 and further developed in Stages 2 and 3, is a high-level conceptual model designed to serve as a tool to facilitate open discussion of the requirements, structures, and operations inherent in Big Data. It is discussed in detail in NIST Big Data Interoperability Framework: Volume 6, Reference Architecture. Potential areas of future work for the Subgroup during stage 2 are highlighted in Section 1.5 of this volume. The current effort documented in this volume reflects concepts developed within the rapidly evolving field of Big Data.

1.2 NIST Big Data Public Working Group

The focus of the NBD-PWG is to form a community of interest from industry, academia, and government, with the goal of developing consensus-based Big Data definitions, taxonomies, reference architectures, and standards roadmap. The aim is to create vendor-neutral, technology- and infrastructure-agnostic deliverables to enable Big Data stakeholders to select the best analytic tools for their processing and visualization requirements on the most suitable computing platforms and clusters while allowing value-added from Big Data service providers and flow of data between the stakeholders in a cohesive and secure manner.

To achieve this goal, five subgroups were formed to address specific issues and develop the deliverables. These subgroups are as follows:

- NIST Big Data Definitions and Taxonomies Subgroup
- NIST Big Data Use Case and Requirements Subgroup
- NIST Big Data Security and Privacy Subgroup
- NIST Big Data Reference Architecture Subgroup
- NIST Big Data Technology Roadmap Subgroup

This volume and its companions were developed based on the following guiding principles:
• Deliverables are technologically agnostic
• The audience is multi-sector, comprised of industry, government, and academia
• Findings from all subgroups are aligned
• Deliverables represent the culmination of concepts from all subgroups

1.3 Scope and Objectives of the Technology Roadmap Subgroup

The NBD-PWG Technology Roadmap Subgroup focused on forming a community of interest from industry, academia, and government, with the goal of developing a consensus vision with recommendations on how Big Data should move forward. The Subgroup’s approach was to perform a gap analysis through the materials gathered from all other subgroups. This included setting standardization and adoption priorities through an understanding of what standards are available or under development as part of the recommendations. The goals of the Subgroup will be realized throughout the three planned phases of the NBD-PWG work, as outlined in Section 1.1. The primary tasks of the NBD-PWG Technology Roadmap Subgroup include the following:

• Gather input from NBD-PWG subgroups and study the taxonomies for the actors’ roles and responsibility, use cases and general requirements, and secure reference architecture
• Gain understanding of what standards are available or under development for Big Data
• Perform a gap analysis and document the findings
• Identify what possible barriers may delay or prevent adoption of Big Data
• Document vision and recommendations

1.4 Report Production

The *NIST Big Data Interoperability Framework: Volume 7, Standards Roadmap* is one of seven volumes in the document, whose overall aims are to define and prioritize Big Data requirements, including interoperability, portability, reusability, extensibility, data usage, analytic techniques, and technology infrastructure in order to support secure and effective adoption of Big Data. The *NIST Big Data Interoperability Framework: Volume 7, Standards Roadmap* is dedicated to developing a consensus vision with recommendations on how Big Data should move forward specifically in the area of standardization. In the first phase, the Subgroup focused on the identification of existing standards relating to Big Data and inspection of gaps in those standards.

Following the introductory material presented in Section 1, the remainder of this document is organized as follows:

• Section 2 summarizes the Big Data definitions presented in the *NIST Interoperability Framework: Volume 1, Definitions* document
• Section 3 summarizes the assessment of the Big Data ecosystem, which was used to develop the NBDRA and this roadmap
Section 4 presents an overview of the NBDRA

Section 5 presents an overview of the security and privacy fabric of the NBDRA

Section 6 investigates the standards related to Big Data and the gaps in those standards

1.5 Future Work on this Volume

The NIST Big Data Interoperability Framework will be released in three versions, which correspond to the three stages of the NBD-PWG work, as outlined in Section 1.1.

Version 2 activities will focus on the following:

- Continue to build and refine the gap analysis and document the findings
- Identify where standards may accelerate the adoption and interoperability of Big Data technologies
- Document recommendations for future standards activities
- Further map standards to NBDRA components and the interfaces between them.

2 Big Data Definition

There are two fundamental concepts in the emerging discipline of Big Data that have been used to represent multiple concepts. These two concepts, Big Data and data science, are broken down into individual terms and concepts in the following subsections. As a basis for discussions of the NBDRA and related standards and measurement technology, associated terminology is defined in subsequent subsections. NIST Big Data Infrastructure Framework: Volume 1, Definitions contains additional details and terminology.

2.1 Big Data Definitions

Big Data refers to the inability of traditional data architectures to efficiently handle the new datasets. Characteristics of Big Data that force new architectures are volume (i.e., the size of the dataset) and variety (i.e., data from multiple repositories, domains, or types), and the data in motion characteristics of velocity (i.e., rate of flow) and variability (i.e., the change in other characteristics). These characteristics—volume, variety, velocity, and variability—are known colloquially as the ‘Vs’ of Big Data and are further discussed in Section 3. Each of these characteristics influences the overall design of a Big Data system, resulting in different data system architectures or different data lifecycle process orderings to achieve needed efficiencies. A number of other terms are also used, several of which refer to the analytics process instead of new Big Data characteristics. The following Big Data definitions have been used throughout the seven volumes of the NIST Big Data Interoperability Framework and are fully described in Volume 1.

**Big Data** consists of extensive datasets—primarily in the characteristics of volume, variety, velocity, and/or variability—that require a scalable architecture for efficient storage, manipulation, and analysis.

The **Big Data paradigm** consists of the distribution of data systems across horizontally coupled, independent resources to achieve the scalability needed for the efficient processing of extensive datasets.
Veracity refers to accuracy of the data

Value refers to the inherent wealth, economic and social, embedded in any data set

Volatility refers to the tendency for data structures to change over time

Validity refers to appropriateness of the data for its intended use

2.2 Data Science Definitions

In its purest form, data science is the fourth paradigm of science, following theory, experiment, and computational science. The fourth paradigm is a term coined by Dr. Jim Gray in 2007 to refer to the conduct of data analysis as an empirical science, learning directly from data itself. Data science as a paradigm would refer to the formulation of a hypothesis, the collection of the data—new or pre-existing—to address the hypothesis, and the analytical confirmation or denial of the hypothesis (or the determination that additional information or study is needed.) As in any experimental science, the end result could in fact be that the original hypothesis itself needs to be reformulated. The key concept is that data science is an empirical science, performing the scientific process directly on the data. Note that the hypothesis may be driven by a business need, or can be the restatement of a business need in terms of a technical hypothesis.

Data science is the empirical synthesis of actionable knowledge from raw data through the complete data lifecycle process.

The data science paradigm is extraction of actionable knowledge directly from data through a process of discovery, hypothesis, and hypothesis testing.

While the above definition of the data science paradigm refers to learning directly from data, in the Big Data paradigm this learning must now implicitly involve all steps in the data lifecycle, with analytics being only a subset. Data science can be understood as the activities happening in the data layer of the system architecture to extract knowledge from the raw data.

The data lifecycle is the set of processes that transform raw data into actionable knowledge.

Traditionally, the term analytics has been used as one of the steps in the data lifecycle of collection, preparation, analysis, and action.

Analytics is the synthesis of knowledge from information.

3 Investigating the Big Data Ecosystem

The development of a Big Data reference architecture involves a thorough understanding of current techniques, issues, concerns, and other topics. To this end, the NBD-PWG collected use cases to gain an understanding of current applications of Big Data, conducted a survey of reference architectures to understand commonalities within Big Data architectures in use, developed a taxonomy to understand and organize the information collected, and reviewed existing Big Data relevant technologies and trends. From the collected information, the NBD-PWG created the NBDRA, which is a high-level conceptual model designed to serve as a tool to facilitate open discussion of the requirements, structures, and operations inherent in Big Data. These NBD-PWG activities were used as input during the development of the entire
3.1 Use Cases

A consensus list of Big Data requirements across stakeholders was developed by the NBD-PWG Use Cases and Requirements Subgroup. The development of requirements included gathering and understanding various use cases from the nine diversified areas, or application domains, listed below.

- Government Operation
- Commercial
- Defense
- Healthcare and Life Sciences
- Deep Learning and Social Media
- The Ecosystem for Research
- Astronomy and Physics
- Earth, Environmental, and Polar Science
- Energy

Participants in the NBD-PWG Use Cases and Requirements Subgroup and other interested parties supplied publically available information for various Big Data architecture examples from the nine application domains, which developed organically from the 51 use cases collected by the Subgroup.

After collection, processing, and review of the use cases, requirements within seven Big Data characteristic categories were extracted from the individual use cases. Requirements are the challenges limiting further use of Big Data. The complete list of requirements extracted from the use cases is presented in the document *NIST Big Data Interoperability Framework: Volume 3, Use Cases and General Requirements*.

The use case specific requirements were then aggregated to produce high-level, general requirements, within seven characteristic categories. The seven categories were as follows:

- **Data sources** (e.g., data size, file formats, rate of growth, at rest or in motion)
- **Data transformation** (e.g., data fusion, analytics)
- **Capabilities** (e.g., software tools, platform tools, hardware resources such as storage and networking)
- **Data consumer** (e.g., processed results in text, table, visual, and other formats)
- **Security and privacy**
Lifecycle management (e.g., curation, conversion, quality check, pre-analytic processing)

Other requirements

The general requirements, created to be vendor neutral and technology agnostic, as listed below.

Data Source Requirements (DSR)

- DSR-1: Needs to support reliable real-time, asynchronous, streaming, and batch processing to collect data from centralized, distributed, and cloud data sources, sensors, or instruments.
- DSR-2: Needs to support slow, bursty, and high-throughput data transmission between data sources and computing clusters.
- DSR-3: Needs to support diversified data content ranging from structured and unstructured text, document, graph, web, geospatial, compressed, timed, spatial, multimedia, simulation, and instrumental data.

Transformation Provider Requirements (TPR)

- TPR-1: Needs to support diversified compute-intensive, analytic processing, and machine learning techniques.
- TPR-2: Needs to support batch and real-time analytic processing.
- TPR-3: Needs to support processing large diversified data content and modeling.
- TPR-4: Needs to support processing data in motion (e.g., streaming, fetching new content, tracking)

Capability Provider Requirements (CPR)

- CPR-1: Needs to support legacy and advanced software packages (software).
- CPR-2: Needs to support legacy and advanced computing platforms (platform).
- CPR-3: Needs to support legacy and advanced distributed computing clusters, co-processors, input output processing (infrastructure).
- CPR-4: Needs to support elastic data transmission (networking).
- CPR-5: Needs to support legacy, large, and advanced distributed data storage (storage).
- CPR-6: Needs to support legacy and advanced executable programming: applications, tools, utilities, and libraries (software).

Data Consumer Requirements (DCR)

- DCR-1: Needs to support fast searches (~0.1 seconds) from processed data with high relevancy, accuracy, and recall.
DCR-2: Needs to support diversified output file formats for visualization, rendering, and reporting.

DCR-3: Needs to support visual layout for results presentation.

DCR-4: Needs to support rich user interface for access using browser, visualization tools.

DCR-5: Needs to support high-resolution, multi-dimension layer of data visualization.

DCR-6: Needs to support streaming results to clients.

Security and Privacy Requirements (SPR)

SPR-1: Needs to protect and preserve security and privacy of sensitive data.

SPR-2: Needs to support sandbox, access control, and multi-level, policy-driven authentication on protected data.

Lifecycle Management Requirements (LMR)

LMR-1: Needs to support data quality curation including pre-processing, data clustering, classification, reduction, and format transformation.

LMR-2: Needs to support dynamic updates on data, user profiles, and links.

LMR-3: Needs to support data lifecycle and long-term preservation policy, including data provenance.

LMR-4: Needs to support data validation.

LMR-5: Needs to support human annotation for data validation.

LMR-6: Needs to support prevention of data loss or corruption.

LMR-7: Needs to support multi-site archives.

LMR-8: Needs to support persistent identifier and data traceability.

LMR-9: Needs to support standardizing, aggregating, and normalizing data from disparate sources.

Other Requirements (OR)

OR-1: Needs to support rich user interface from mobile platforms to access processed results.

OR-2: Needs to support performance monitoring on analytic processing from mobile platforms.

OR-3: Needs to support rich visual content search and rendering from mobile platforms.

OR-4: Needs to support mobile device data acquisition.

OR-5: Needs to support security across mobile devices.
Additional information about the Subgroup, use case collection, analysis of the use cases, and generation of the use case requirements are presented in the *NIST Big Data Interoperability Framework: Volume 3, Use Cases and General Requirements* document.

### 3.2 Reference Architecture Survey

The NBD-PWG Reference Architecture Subgroup conducted the reference architecture survey to advance understanding of the operational intricacies in Big Data and to serve as a tool for developing system-specific architectures using a common reference framework. The Subgroup surveyed currently published Big Data platforms by leading companies or individuals supporting the Big Data framework and analyzed the collected material. This effort revealed a remarkable consistency between Big Data architectures. Survey details, methodology, and conclusions are reported in *NIST Big Data Interoperability Framework: Volume 5, Architectures White Paper Survey*.

### 3.3 Taxonomy


Figure 1 outlines potential actors for the seven roles developed by the NBD-PWG Definition and Taxonomy Subgroup. The dark blue boxes contain the name of the role at the top with potential actors listed directly below.

**Figure 1: NIST Big Data Reference Architecture Taxonomy**

---

**4 Big Data Reference Architecture**

### 4.1 Overview

The goal of the NBD-PWG Reference Architecture Subgroup is to develop a Big Data, open reference architecture that facilitates the understanding of the operational intricacies in Big Data. It does not represent the system architecture of a specific Big Data system, but rather is a tool for describing, discussing, and developing system-specific architectures using a common framework of reference. The reference architecture achieves this by providing a generic high-level
conceptual model that is an effective tool for discussing the requirements, structures, and operations inherent to Big Data. The model is not tied to any specific vendor products, services, or reference implementation, nor does it define prescriptive solutions that inhibit innovation.

The design of the NBDRA does not address the following:

- Detailed specifications for any organization’s operational systems
- Detailed specifications of information exchanges or services
- Recommendations or standards for integration of infrastructure products

Building on the work from other subgroups, the NBD PWG Reference Architecture Subgroup evaluated the general requirements formed from the use cases, evaluated the Big Data Taxonomy, performed a reference architecture survey, and developed the NBDRA conceptual model. The *NIST Big Data Interoperability Framework: Volume 3, Use Cases and General Requirements* document contains details of the Subgroup’s work.

The NBD-PWG Use Case Subgroup developed requirements in seven categories, which correspond to the reference architecture components as shown in Table 1. The requirements from each category were used as input for the development of the corresponding NBDRA component.

**Table 1: Mapping of Use Case Categories to the NBDRA Components**

<table>
<thead>
<tr>
<th>Use Case Characterization Categories</th>
<th>Corresponds To</th>
<th>Reference Architecture Components And Fabrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data sources</td>
<td>→</td>
<td>Data Provider</td>
</tr>
<tr>
<td>Data transformation</td>
<td>→</td>
<td>Big Data Application Provider</td>
</tr>
<tr>
<td>Capabilities</td>
<td>→</td>
<td>Big Data Framework Provider</td>
</tr>
<tr>
<td>Data consumer</td>
<td>→</td>
<td>Data Consumer</td>
</tr>
<tr>
<td>Security and privacy</td>
<td>→</td>
<td>Security and Privacy Fabric</td>
</tr>
<tr>
<td>Lifecycle management</td>
<td>→</td>
<td>System Orchestrator; Management Fabric</td>
</tr>
</tbody>
</table>
4.2 NBDRA Conceptual Model

The NBD-PWG Reference Architecture Subgroup used a variety of inputs from other NBD-PWG subgroups in developing a vendor-neutral, technology- and infrastructure-agnostic, conceptual model of Big Data architecture. This conceptual model, the NBDRA, is shown in Figure 2 and represents a Big Data system comprised of five logical functional components connected by interoperability interfaces (i.e., services). Two fabrics envelop the components, representing the interwoven nature of management and security and privacy with all five of the components.

The NBDRA is intended to enable system engineers, data scientists, software developers, data architects, and senior decision makers to develop solutions to issues that require diverse approaches due to convergence of Big Data characteristics within an interoperable Big Data ecosystem. It provides a framework to support a variety of business environments, including tightly-integrated enterprise systems and loosely-coupled vertical industries, by enhancing understanding of how Big Data complements and differs from existing analytics, business intelligence, databases, and systems.

Figure 2: NBDRA Conceptual Model

The NBDRA is organized around two axes representing the two Big Data value chains: the information (horizontal axis) and the Information Technology (IT) (vertical axis). Along the information axis, the value is created by data collection, integration, analysis, and applying the results following the value chain. Along the IT axis, the value is created by providing networking, infrastructure, platforms, application tools, and other IT services for hosting of and operating the Big Data in support of required data applications. At the intersection of both axes is the Big Data Application Provider component, indicating that data analytics and its implementation provide the value to Big Data stakeholders in both value chains. The names of the Big Data Application Provider and Big Data Framework Provider components contain

https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework
Updated: Sat, 14 Sep 2019 05:26:25 GMT
Powered by mindtouch™
“providers” to indicate that these components provide or implement a specific technical function within the system.

The five main NBDRA components, shown in Figure 2 and discussed in detail in Section 4, represent different technical roles that exist in every Big Data system. These functional components are as follows:

- System Orchestrator
- Data Provider
- Big Data Application Provider
- Big Data Framework Provider
- Data Consumer

The two fabrics shown in Figure 2 encompassing the five functional components are the following:

- Management
- Security and Privacy

These two fabrics provide services and functionality to the five functional components in the areas specific to Big Data and are crucial to any Big Data solution.

The “DATA” arrows in Figure 2 show the flow of data between the system’s main components. Data flows between the components either physically (i.e., by value) or by providing its location and the means to access it (i.e., by reference). The “SW” arrows show transfer of software tools for processing of Big Data in situ. The “Service Use” arrows represent software programmable interfaces. While the main focus of the NBDRA is to represent the run-time environment, all three types of communications or transactions can happen in the configuration phase as well. Manual agreements (e.g., service-level agreements [SLAs]) and human interactions that may exist throughout the system are not shown in the NBDRA.

The components represent functional roles in the Big Data ecosystem. In system development, actors and roles have the same relationship as in the movies, but system development actors can represent individuals, organizations, software, or hardware. According to the Big Data taxonomy, a single actor can play multiple roles, and multiple actors can play the same role. The NBDRA does not specify the business boundaries between the participating actors or stakeholders, so the roles can either reside within the same business entity or can be implemented by different business entities. Therefore, the NBDRA is applicable to a variety of business environments, from tightly-integrated enterprise systems to loosely-coupled vertical industries that rely on the cooperation of independent stakeholders. As a result, the notion of internal versus external functional components or roles does not apply to the NBDRA. However, for a specific use case, once the roles are associated with specific business stakeholders, the functional components would be considered as internal or external—subject to the use case’s point of view.

The NBDRA does support the representation of stacking or chaining of Big Data systems. For example, a Data Consumer of one system could serve as a Data Provider to the next system down the stack or chain.

The five main components and the two fabrics of the NBDRA are discussed in the NIST Big Data Interoperability Framework: Volume 6, Reference Architecture and Volume 4, Security and Privacy.
5 Big Data Security and Privacy

Security and privacy measures for Big Data involve a different approach than traditional systems. Big Data is increasingly stored on public cloud infrastructure built by various hardware, operating systems, and analytical software. Traditional security approaches usually addressed small scale systems holding static data on firewalled and semi-isolated networks. The surge in streaming cloud technology necessitates extremely rapid responses to security issues and threats.\[2\]

Security and privacy considerations are a fundamental aspect of Big Data and affect all components of the NBDRA. This comprehensive influence is depicted in Figure 2 by the grey rectangle marked “Security and Privacy” surrounding all of the reference architecture components. At a minimum, a Big Data reference architecture will provide verifiable compliance with both governance, risk management, and compliance (GRC) and confidentiality, integrity, and availability (CIA) policies, standards, and best practices. Additional information on the processes and outcomes of the NBD PWG Security and Privacy Subgroup are presented in NIST Big Data Interoperability Framework: Volume 4, Security and Privacy.

6 Big Data Standards

Big Data has generated interest in a wide variety of multi-stakeholder, collaborative organizations, including those involved in the de jure standards process, industry consortia, and open source organizations. These organizations may operate differently and focus on different aspects, but they all have a stake in Big Data. Integrating additional Big Data initiatives with ongoing collaborative efforts is a key to success. Identifying which collaborative initiative efforts address architectural requirements and which requirements are not currently being addressed is a starting point for building future multi-stakeholder collaborative efforts. Collaborative initiatives include, but are not limited to the following:

- Subcommittees and working groups of American National Standards Institute (ANSI)
- Accredited standards development organizations (SDOs; the de jure standards process)
- Industry consortia
- Reference implementations
- Open source implementations

Some of the leading SDOs and industry consortia working on Big Data related standards include:

- International Committee for Information Technology Standards (INCITS) and International Organization for Standardization (ISO)—de jure standards process
- Institute of Electrical and Electronics Engineers (IEEE)—de jure standards process
- International Electrotechnical Commission (IEC)
- Internet Engineering Task Force (IETF)
· World Wide Web Consortium (W3C)—Industry consortium
· Open Geospatial Consortium (OGC®)—Industry consortium
· Organization for the Advancement of Structured Information Standards (OASIS)—Industry consortium
· Open Grid Forum (OGF)—Industry consortium

The organizations and initiatives referenced in this document do not from an exhaustive list. It is anticipated that as this document is more widely distributed, more standards efforts addressing additional segments of the Big Data mosaic will be identified.

There are a number of government organizations that publish standards relative to their specific problem areas. The US Department of Defense alone maintains hundreds of standards. Many of these are based on other standards (e.g., ISO, IEEE, ANSI) and could be applicable to the Big Data problem space. However, a fair, comprehensive review of these standards would exceed the available document preparation time and may not be of interest to the majority of the audience for this report. Readers interested in domains covered by the government organizations and standards, are encouraged to review the standards for applicability to their specific needs.

Open source implementations are providing useful new technology that is being used either directly or as the basis for commercially supported products. These open source implementations are not just individual products. One needs to integrate an eco-system of products to accomplish ones goals. Because of the ecosystem complexity, and because of the difficulty of fairly and exhaustively reviewing open source implementations, such implementations are not included in this section. However, it should be noted that those implementations often evolve to become the de facto reference implementations for many technologies.

### 6.1 Existing Standards

This section presents a list of existing standards from the above listed organizations that are relevant to Big Data and the NBDRA. Determining the relevance of standards to the Big Data domain is challenging since almost all standards in some way deal with data. Whether a standard is relevant to Big Data is generally determined by impact of Big Data characteristics (i.e., volume, velocity, variety, and veracity) on the standard or, more generally, by the scalability of the standard to accommodate those characteristics. A standard may also be applicable to Big Data depending on the extent to which that standard helps to address one or more of the Big Data characteristics. Finally, a number of standards are also very domain or problem specific and, while they deal with or address Big Data, they support a very specific functional domain and developing even a marginally comprehensive list of such standards would require a massive undertaking involving subject matter experts in each potential problem domain, which is beyond the scope of the NBD-PWG.

In selecting standards to include in Table 2, the working group focused on standards that would do the following:

- Facilitate interfaces between NBDRA components
- Facilitate the handling of data with one or more Big Data characteristics
- Represent a fundamental function needing to be implemented by one or more NBDRA components
Table 2 represents a portion of potentially applicable standards from a portion of contributing organizations working in Big Data domain.

As most standards represent some form of interface between components Table 2 is annotated with whether the NBDRA component would be an Implementer or User of the standard. For the purposes of this table the following definitions were used for Implementer and User.

**Implementer:** A component is an implementer of a standard if it provides services based on the standard (e.g., a service that accepts Structured Query Language [SQL] commands would be an implementer of that standard) or encodes or presents data based on that standard.

**User:** A component is a user of a standard if it interfaces to a service via the standard or if it accepts/consumes/decodes data represented by the standard.

While the above definitions provide a reasonable basis for some standards the difference between implementation and use may be negligible or non-existent.

The NBDRA components are abbreviated in the table header as follows:

- SO = System Orchestrator component
- DP = Data Provider component
- DC = Data Consumer component
- BDAP = Big Data Application Provider component
- BDFP = Big Data Framework Provider component
- S&P = Security and Privacy Fabric
- M = Management Fabric

Table 2: Existing Big Data Standards

<table>
<thead>
<tr>
<th>Standard Name/Number</th>
<th>Description</th>
<th>NBDRA Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO/IEC 9075-*</td>
<td>ISO/IEC 9075 defines SQL. The scope of SQL is the definition of data structure and the</td>
<td>SO DP DC BDAP BDFP S&amp;P M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I I/U U I/U U U</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>ISO/IEC 11179-*</td>
<td>The 11179 standard is a multipart standard for the definition and implementation of Metadata Registries. The series includes the following parts:</td>
<td>I I/U I/U U</td>
</tr>
<tr>
<td></td>
<td>· Part 1: Framework</td>
<td></td>
</tr>
<tr>
<td></td>
<td>· Part 2: Classification</td>
<td></td>
</tr>
<tr>
<td></td>
<td>· Part 3: Registry metamodel and basic attributes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>· Part 4: Formulation of data definitions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>· Part 5: Naming and identification principles</td>
<td></td>
</tr>
<tr>
<td></td>
<td>· Part 6: Registration</td>
<td></td>
</tr>
<tr>
<td>ISO/IEC 10728-*</td>
<td>Information Resource Dictionary System Services Interface</td>
<td></td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td>------------------</td>
</tr>
<tr>
<td>ISO/IEC 13249-*</td>
<td>Database Languages – SQL Multimedia and Application Packages</td>
<td>SO DP DC BDAP BDFP S&amp;P M</td>
</tr>
<tr>
<td></td>
<td>I I/U U I/U</td>
<td></td>
</tr>
<tr>
<td>ISO/IEC TR 19075-*</td>
<td>This is a series of TRs on SQL related technologies. · Part 1: Xquery · Part 2: SQL Support for Time-Related Information · Part 3: Programs Using the Java Programming Language · Part 4: Routines and Types Using the Java Programming Language</td>
<td>SO DP DC BDAP BDFP S&amp;P M</td>
</tr>
<tr>
<td></td>
<td>I I/U U I/U</td>
<td></td>
</tr>
<tr>
<td>ISO/IEC 19503</td>
<td>Extensible Markup Language (XML) Metadata Interchange (XMI)</td>
<td>SO DP DC BDAP BDFP S&amp;P M</td>
</tr>
<tr>
<td></td>
<td>I I/U U I/U</td>
<td></td>
</tr>
<tr>
<td>ISO/IEC 19773</td>
<td>Metadata Registries Modules</td>
<td>SO DP DC BDAP BDFP S&amp;P M</td>
</tr>
<tr>
<td>ISO/IEC TR 20943</td>
<td>Metadata Registry Content Consistency</td>
<td>SO DP DC BDAP BDFP S&amp;P M</td>
</tr>
<tr>
<td></td>
<td>I I/U U U</td>
<td></td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td>The 19763 standard is a multipart standard that includes the following parts:</td>
<td>SO</td>
</tr>
<tr>
<td></td>
<td>· Part 1: Reference model</td>
<td></td>
</tr>
<tr>
<td></td>
<td>· Part 3: Metamodel for ontology registration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>· Part 5: Metamodel for process model registration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>· Part 6: Registry Summary</td>
<td></td>
</tr>
<tr>
<td></td>
<td>· Part 7: Metamodel for service registration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>· Part 8: Metamodel for role and goal registration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>· Part 9: On Demand Model Selection (ODMS) TR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>· Part 10: Core model and basic mapping</td>
<td></td>
</tr>
<tr>
<td></td>
<td>· Part 12: Metamodel for information model registration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>· Part 13: Metamodel for forms registration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>· Part 14: Metamodel for dataset registration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>· Part 15: Metamodel for data provenance registration</td>
<td></td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>IEEE 21451-*</td>
<td>Information Technology—Smart transducer interface for sensors and actuators</td>
<td>SO: I, DP: U, DC:</td>
</tr>
<tr>
<td></td>
<td>- Part 1: Network Capable Application Processor (NCAP) information model</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Part 2: Transducer to microprocessor communication protocols and Transducer Electronic Data Sheet (TEDS)</td>
<td></td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SO</td>
</tr>
<tr>
<td>formats</td>
<td></td>
<td></td>
</tr>
<tr>
<td>· Part 4: Mixed-mode communication protocols and TEDS formats</td>
<td></td>
<td></td>
</tr>
<tr>
<td>· Part 7: Transducer to radio frequency identification (RFID) systems communication protocols and TEDS formats</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td>------------------</td>
</tr>
<tr>
<td>JSR (Java Specification Request) 221</td>
<td>JDBC™ 4.0 Application Programming Interface (API)</td>
<td>I/U</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td>------------------</td>
</tr>
<tr>
<td>(developed by the Java Community Process)</td>
<td>Specification</td>
<td>SO</td>
</tr>
<tr>
<td>W3C Resource Description Framework (RDF)</td>
<td>The RDF is a framework for representing information in the Web. RDF graphs are sets of subject-predicate-object triples, where the elements are used to express descriptions of resources.</td>
<td>I</td>
</tr>
<tr>
<td>W3C JavaScript Object Notation (JSON)-LD 1.0</td>
<td>JSON-LD 1.0 A JSON-based Serialization for Linked Data W3C Recommendation 16 January 2014</td>
<td>I</td>
</tr>
<tr>
<td>W3C Document Object Model (DOM) Level 1 Specification</td>
<td>This series of specifications define the DOM, a platform- and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of HyperText Markup Language (HTML) and XML documents.</td>
<td>I</td>
</tr>
<tr>
<td>W3C XQuery 3.0</td>
<td>The XQuery specifications describe a query language called XQuery, which is</td>
<td>I</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td>designed to be broadly applicable across many types of XML data sources.</td>
<td></td>
</tr>
<tr>
<td>W3C XProc</td>
<td>This specification describes the syntax and semantics of XProc: An XML Pipeline Language, a language for describing operations to be performed on XML documents.</td>
<td>I     I   U   I/U  I/U</td>
</tr>
<tr>
<td>W3C XML Encryption</td>
<td>This specification covers a process for encrypting data and representing the result in XML.</td>
<td>I     U   I/U</td>
</tr>
<tr>
<td>Syntax and Processing</td>
<td>Version 1.1</td>
<td></td>
</tr>
<tr>
<td>W3C XML Signature</td>
<td>This specification covers XML digital signature processing rules and syntax. XML Signatures provide integrity, message authentication, and/or signer authentication services for data of any type, whether located within the XML that includes the signature or elsewhere.</td>
<td>I     U   I/U</td>
</tr>
<tr>
<td>Syntax and Processing</td>
<td>Version 1.1</td>
<td></td>
</tr>
<tr>
<td>W3C XPath 3.0</td>
<td>XPath 3.0 is an expression language that allows the processing of values conforming to the data model defined in [XQuery and XPath Data Model (XDM) 3.0]. The data model provides a tree representation of XML documents as well as</td>
<td>I     U   I/U  I/U</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td>atomic values and sequences that may contain both references to nodes in an XML document and atomic values.</td>
<td></td>
</tr>
<tr>
<td>W3C XSL Transformations (XSLT) Version 2.0</td>
<td>This specification defines the syntax and semantics of XSLT 2.0, a language for transforming XML documents into other XML documents.</td>
<td>I U I/U I/U</td>
</tr>
<tr>
<td>W3C Efficient XML Interchange (EXI) Format 1.0 (Second Edition)</td>
<td>This specification covers the EXI format. EXI is a very compact representation for the XML Information Set that is intended to simultaneously optimize performance and the utilization of computational resources.</td>
<td>I U I/U</td>
</tr>
<tr>
<td>W3C RDF Data Cube Vocabulary</td>
<td>The Data Cube vocabulary provides a means to publish multi-dimensional data, such as statistics on the Web using the W3C RDF standard.</td>
<td>I U I/U I/U</td>
</tr>
<tr>
<td>W3C Data Catalog Vocabulary (DCAT)</td>
<td>DCAT is an RDF vocabulary designed to facilitate interoperability between data catalogs published on the Web. This document defines the schema and provides examples</td>
<td>I U I/U</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td>------------------</td>
</tr>
<tr>
<td><strong>W3C HTML5 A vocabulary and associated APIs for HTML and XHTML</strong></td>
<td>This specification defines the 5th major revision of the core language of the World Wide Web—HTML.</td>
<td>SO DP DC BDAP BDFP S&amp;P M</td>
</tr>
<tr>
<td><strong>W3C Internationalization Tag Set (ITS) 2.0</strong></td>
<td>The ITS 2.0 specification enhances the foundation to integrate automated processing of human language into core Web technologies and concepts that are designed to foster the automated creation and processing of multilingual Web content.</td>
<td>SO DP DC BDAP BDFP S&amp;P M</td>
</tr>
<tr>
<td><strong>W3C OWL 2 Web Ontology Language</strong></td>
<td>The OWL 2 Web Ontology Language, informally OWL 2, is an ontology language for the Semantic Web with formally defined meaning.</td>
<td>SO DP DC BDAP BDFP S&amp;P M</td>
</tr>
<tr>
<td><strong>W3C Platform for Privacy Preferences (P3P) 1.0</strong></td>
<td>The P3P enables Web sites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents.</td>
<td>SO DP DC BDAP BDFP S&amp;P M</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td>------------------</td>
</tr>
<tr>
<td><strong>W3C Protocol for Web Description Resources (POWDER)</strong></td>
<td>POWDER—the Protocol for Web Description Resources—provides a mechanism to describe and discover Web resources and helps the users to make a decision whether a given resource is of interest.</td>
<td>SO I  DP U  DC I/U  BDAP I/U  BDFP I/U  S&amp;P I/U  M</td>
</tr>
<tr>
<td><strong>W3C Provenance</strong></td>
<td>Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness. The Provenance Family of Documents (PROV) defines a model, corresponding serializations and other supporting definitions to enable the inter-operable interchange of provenance information in heterogeneous environments such as the Web.</td>
<td>SO I  DP U  DC I/U  BDAP I/U  BDFP I/U  S&amp;P I/U  M</td>
</tr>
<tr>
<td><strong>W3C Rule Interchange Format (RIF)</strong></td>
<td>RIF is a series of standards for exchanging rules among rule systems, in particular among Web rule engines.</td>
<td>SO I  DP U  DC I/U  BDAP I/U  BDFP I/U  S&amp;P I/U  M</td>
</tr>
<tr>
<td><strong>W3C Service Modeling Language (SML) 1.1</strong></td>
<td>This specification defines the SML, Version 1.1 used to model complex services and systems, including their structure,</td>
<td>SO I/U  DP I/U  DC I/U  BDAP I/U  BDFP I/U  S&amp;P I/U  M</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td>constraints, policies, and best practices.</td>
<td></td>
</tr>
<tr>
<td>W3C Simple Knowledge Organization System Reference (SKOS)</td>
<td>This document defines the SKOS, a common data model for sharing and linking knowledge organization systems via the Web.</td>
<td>I</td>
</tr>
<tr>
<td>W3C Simple Object Access Protocol (SOAP) 1.2</td>
<td>SOAP is a protocol specification for exchanging structured information in the implementation of web services in computer networks.</td>
<td>I</td>
</tr>
<tr>
<td>W3C SPARQL 1.1</td>
<td>SPARQL is a language specification for the query and manipulation of linked data in a RDF format.</td>
<td>I</td>
</tr>
<tr>
<td>W3C Web Service Description Language (WSDL) 2.0</td>
<td>This specification describes the WSDL Version 2.0, an XML language for describing Web services.</td>
<td>U</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td><strong>W3C XML Key Management Specification (XKMS) 2.0</strong></td>
<td>This standard specifies protocols for distributing and registering public keys, suitable for use in conjunction with the W3C Recommendations for XML Signature [XML-SIG] and XML Encryption [XML-Enc]. The XKMS comprises two parts — the XML Key Information Service Specification (X-KISS) and the XML Key Registration Service Specification (X-KRSS).</td>
<td>SO   DP  DC  BDAP BDFP S&amp;P M</td>
</tr>
<tr>
<td><strong>OGC® OpenGIS® Catalogue Services Specification 2.0.2 - ISO Metadata Application Profile</strong></td>
<td>This series of standard covers Catalogue Services based on ISO19115/ISO19119 are organized and implemented for the discovery, retrieval and management of data metadata, services metadata and application metadata.</td>
<td>I     U   I/U</td>
</tr>
<tr>
<td><strong>OGC® OpenGIS® GeoAPI</strong></td>
<td>The GeoAPI Standard defines, through the GeoAPI library, a Java language API including a set of types and methods which can be used for the manipulation of geographic information structured following the specifications adopted by the Technical Committee 211 of the ISO and by the OGC®.</td>
<td>I     U   I/U I/U</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>OGC® OpenGIS® GeoSPARQL</td>
<td>The OGC® GeoSPARQL standard supports representing and querying geospatial data on the Semantic Web. GeoSPARQL defines a vocabulary for representing geospatial data in RDF, and it defines an extension to the SPARQL query language for processing geospatial data.</td>
<td>I</td>
</tr>
<tr>
<td>OGC® OpenGIS® Geography Markup Language (GML) Encoding Standard</td>
<td>The GML is an XML grammar for expressing geographical features. GML serves as a modeling language for geographic systems as well as an open interchange format for geographic transactions on the Internet.</td>
<td>I</td>
</tr>
<tr>
<td>OGC® Geospatial eXtensible Access Control Markup Language (GeoXACML) Version 1</td>
<td>The Policy Language introduced in this document defines a geo-specific extension to the XACML Policy Language, as defined by the OASIS standard eXtensible Access Control Markup Language (XACML), Version 2.0.</td>
<td>I</td>
</tr>
<tr>
<td>OGC® network Common Data Form (netCDF)</td>
<td>netCDF is a set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of data.</td>
<td>I</td>
</tr>
</tbody>
</table>

https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework
Updated: Sat, 14 Sep 2019 05:26:25 GMT
Powered by mindtouch™ 250
<table>
<thead>
<tr>
<th>Standard Name/Number</th>
<th>Description</th>
<th>NBDRA Components</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>array-oriented scientific data.</td>
<td></td>
</tr>
<tr>
<td>OGC® Open Modelling</td>
<td>The purpose of the OpenMI is to enable the runtime exchange of data between process simulation models and also between models and other modelling tools such as databases and analytical and visualization applications.</td>
<td>I</td>
</tr>
<tr>
<td>Interface Standard</td>
<td></td>
<td>U</td>
</tr>
<tr>
<td>(OpenMI)</td>
<td></td>
<td>I/U</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I/U</td>
</tr>
<tr>
<td>OGC® OpenSearch Geo</td>
<td>This OGC standard specifies the Geo and Time extensions to the OpenSearch query protocol. OpenSearch is a collection of simple formats for the sharing of search results.</td>
<td>I</td>
</tr>
<tr>
<td>and Time Extensions</td>
<td></td>
<td>U</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I/U</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I</td>
</tr>
<tr>
<td>OGC® Web Services</td>
<td>The OGC® OWS Context was created to allow a set of configured information resources (service set) to be passed between applications primarily as a collection of services.</td>
<td>I</td>
</tr>
<tr>
<td>Context Document</td>
<td></td>
<td>U</td>
</tr>
<tr>
<td>(OWS Context)</td>
<td></td>
<td>I/U</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I</td>
</tr>
<tr>
<td>OGC® Sensor Web</td>
<td>This series of standards support interoperability interfaces and metadata encodings that enable real time integration of heterogeneous sensor webs. These standards include a modeling language (SensorML),</td>
<td>I</td>
</tr>
<tr>
<td>Enablement (SWE)</td>
<td></td>
<td>U</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I/U</td>
</tr>
</tbody>
</table>

https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework
Updated: Sat, 14 Sep 2019 05:26:25 GMT
Powered by mindtouch®
<table>
<thead>
<tr>
<th>Standard Name/Number</th>
<th>Description</th>
<th>NBDRA Components</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OGC® OpenGIS® Simple Features Access (SFA)</strong></td>
<td>The common data model, and sensor observation, planning, and alerting service interfaces.</td>
<td>SO   DP   DC   BDAP   BDFP   S&amp;P   M</td>
</tr>
<tr>
<td><strong>OGC® OpenGIS® Georeferenced Table Joining Service (TJS) Implementation Standard</strong></td>
<td>Describes the common architecture for simple feature geometry and is also referenced as ISO 19125. It also implements a profile of the spatial schema described in ISO 19107:2003.</td>
<td>I    U    I/U  I/U</td>
</tr>
<tr>
<td><strong>OGC® OpenGIS® Web Coverage Processing Service Interface (WCPS) Standard</strong></td>
<td>This standard is the specification for a TJS that defines a simple way to describe and exchange tabular data that contains information about geographic objects.</td>
<td>I    U    I/U  I/U</td>
</tr>
<tr>
<td><strong>OGC® OpenGIS® Web Coverage Service (WCS)</strong></td>
<td>Defines a protocol-independent language for the extraction, processing, and analysis of multi-dimensional gridded coverages representing sensor, image, or statistics data.</td>
<td>I    U    I/U  I</td>
</tr>
<tr>
<td></td>
<td>This document specifies how a WCS offers multi-dimensional coverage data for access over the Internet. This document specifies a core set of</td>
<td>I    U    I/U  I</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td><strong>OGC® Web Feature Service (WFS) 2.0 Interface Standard</strong></td>
<td>The WFS standard provides for fine-grained access to geographic information at the feature and feature property level. This International Standard specifies discovery operations, query operations, locking operations, transaction operations and operations to manage stored, parameterized query expressions.</td>
<td>I U I/U I</td>
</tr>
<tr>
<td><strong>OGC® OpenGIS® Web Map Service (WMS) Interface Standard</strong></td>
<td>The OpenGIS® WMS Interface Standard provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases.</td>
<td>I U I/U I</td>
</tr>
<tr>
<td><strong>OGC® OpenGIS® Web Processing Service (WPS) Interface Standard</strong></td>
<td>The OpenGIS® WPS Interface Standard provides rules for standardizing how inputs and outputs (requests and responses) for geospatial processing services, such as polygon overlay. The standard also defines how a client can request the execution of a process, and how the output</td>
<td>I U I/U I</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td></td>
<td>from the process is handled. It defines an interface that facilitates the publishing of geospatial processes and clients’ discovery of and binding to those processes.</td>
<td></td>
</tr>
<tr>
<td>OASIS AS4 Profile of ebMS 3.0 v1.0</td>
<td>Standard for business to business exchange of messages via a web service platform.</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>U</td>
</tr>
<tr>
<td>OASIS Advanced Message Queuing Protocol (AMQP) Version 1.0</td>
<td>The AMQP is an open internet protocol for business messaging. It defines a binary wire-level protocol that allows for the reliable exchange of business messages between two parties.</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>U</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I</td>
</tr>
<tr>
<td>OASIS Application Vulnerability Description Language (AVDL) v1.0</td>
<td>This specification describes a standard XML format that allows entities (such as applications, organizations, or institutes) to communicate information regarding web application vulnerabilities.</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>U</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>U</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>OASIS Content Management Interoperability Services (CMIS)</td>
<td>The CMIS standard defines a domain model and set of bindings that include Web Services and ReSTful AtomPub that can be used by applications to work with one or more Content Management repositories/systems.</td>
<td>SO: I, DP: U, DC: I/U, BDAP: I</td>
</tr>
<tr>
<td>OASIS Digital Signature Service (DSS)</td>
<td>This specification describes two XML-based request/response protocols - a signing protocol and a verifying protocol. Through these protocols a client can send documents (or document hashes) to a server and receive back a signature on the documents; or send documents (or document hashes) and a signature to a server, and receive back an answer on whether the signature verifies the documents.</td>
<td>SO: I, DP: U, DC: I/U</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td>------------------</td>
</tr>
<tr>
<td><strong>OASIS Directory Services Markup Language (DSML) v2.0</strong></td>
<td>The DSML provides a means for representing directory structural information as an XML document methods for expressing directory queries and updates (and the results of these operations) as XML documents</td>
<td>I</td>
</tr>
<tr>
<td><strong>OASIS ebXML Messaging Services</strong></td>
<td>These specifications define a communications-protocol neutral method for exchanging electronic business messages as XML.</td>
<td>I</td>
</tr>
<tr>
<td><strong>OASIS ebXML RegRep</strong></td>
<td>ebXML RegRep is a standard defining the service interfaces, protocols and information model for an integrated registry and repository. The repository stores digital content while the registry stores metadata that describes the content in the repository.</td>
<td>I</td>
</tr>
<tr>
<td><strong>OASIS ebXML Registry Information Model</strong></td>
<td>The Registry Information Model provides a blueprint or high-level schema for the ebXML Registry. It provides implementers with the type of metadata that is stored in the Registry as well as the relationships among metadata Classes.</td>
<td>I</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>OASIS ebXML Registry Services Specification</td>
<td>An ebXML Registry is an information system that securely manages any content type and the standardized metadata that describes it. The ebXML Registry provides a set of services that enable sharing of content and metadata between organizational entities in a federated environment.</td>
<td>SO  DP  DC  BDAP  BDFP  S&amp;P  M</td>
</tr>
<tr>
<td>OASIS eXtensible Access Control Markup Language (XACML)</td>
<td>The standard defines a declarative access control policy language implemented in XML and a processing model describing how to evaluate access requests according to the rules defined in policies.</td>
<td>I    U    I/U</td>
</tr>
<tr>
<td>OASIS Message Queuing Telemetry Transport (MQTT)</td>
<td>MQTT is a Client Server publish/subscribe messaging transport protocol for constrained environments such as for communication in Machine to Machine and Internet of Things contexts where a small code footprint is required and/or network bandwidth is at a premium.</td>
<td>I    U    I/U</td>
</tr>
<tr>
<td>OASIS Open Data (OData) Protocol</td>
<td>The OData Protocol is an application-level protocol for interacting with data via RESTful interfaces. The protocol supports</td>
<td>I    U    I/U    I/U</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SO</td>
</tr>
<tr>
<td>NBDRA Components</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SO</td>
<td>the description of data models and the editing and querying of data according to those models.</td>
<td>I</td>
</tr>
<tr>
<td>DC</td>
<td>The OASIS SWS initiative defines a generic protocol for the interaction required between a client and server for performing searches. SWS define an Abstract Protocol Definition to describe this interaction.</td>
<td>I</td>
</tr>
<tr>
<td>BDAP</td>
<td>The SAML defines the syntax and processing semantics of assertions made about a subject by a system entity. This specification defines both the structure of SAML assertions, and an associated set of protocols, in addition to the processing rules involved in managing a SAML system.</td>
<td>I</td>
</tr>
<tr>
<td>BDFP</td>
<td>This specification defines a binding of SOAP to user datagrams, including message patterns, addressing requirements, and security considerations.</td>
<td>I</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>OASIS Solution Deployment Descriptor Specification v1.0</td>
<td>This specification defines schema for two XML document types: Package Descriptors and Deployment Descriptors. Package Descriptors define characteristics of a package used to deploy a solution. Deployment Descriptors define characteristics of the content of a solution package, including the requirements that are relevant for creation, configuration and maintenance of the solution content.</td>
<td>U</td>
</tr>
<tr>
<td>OASIS Symptoms Automation Framework (SAF) Version 1.0</td>
<td>This standard defines reference architecture for the Symptoms Automation Framework, a tool in the automatic detection, optimization, and remediation of operational aspects of complex systems,</td>
<td>I/U</td>
</tr>
<tr>
<td>OASIS Topology and Orchestration Specification for Cloud Applications Version 1.0</td>
<td>The concept of a “service template” is used to specify the “topology” (or structure) and “orchestration” (or invocation of management behavior) of IT services. This specification introduces the formal description of Service Templates, including their structure, properties, and behavior.</td>
<td>I/U U I</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td>------------------</td>
</tr>
<tr>
<td><strong>OASIS Universal Business Language (UBL) v2.1</strong></td>
<td>The OASIS UBL defines a generic XML interchange format for business documents that can be restricted or extended to meet the requirements of particular industries.</td>
<td>I U I/U U</td>
</tr>
<tr>
<td><strong>OASIS Universal Description, Discovery and Integration (UDDI) v3.0.2</strong></td>
<td>The focus of UDDI is the definition of a set of services supporting the description and discovery of (1) businesses, organizations, and other Web services providers, (2) the Web services they make available, and (3) the technical interfaces which may be used to access those services.</td>
<td>I U I/U U</td>
</tr>
<tr>
<td><strong>OASIS Unstructured Information Management Architecture (UIMA) v1.0</strong></td>
<td>The UIMA specification defines platform-independent data representations and interfaces for text and multi-modal analytics.</td>
<td>U I</td>
</tr>
<tr>
<td><strong>OASIS Unstructured Operation Markup Language (UOML) v1.0</strong></td>
<td>UOML is interface standard to process unstructured document; it plays the similar role as SQL to structured data. UOML is expressed with standard XML.</td>
<td>I U I/U I</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>OASIS/W3C WebCGM v2.1</td>
<td>Computer Graphics Metafile (CGM) is an ISO standard, defined by ISO/IEC 8632:1999, for the interchange of 2D vector and mixed vector/raster graphics. WebCGM is a profile of CGM, which adds Web linking and is optimized for Web applications in technical illustration, electronic documentation, geophysical data visualization, and similar fields.</td>
<td>I U I/U I</td>
</tr>
<tr>
<td>OASIS Web Services Business Process Execution Language (WS-BPEL) v2.0</td>
<td>This standard defines a language for specifying business process behavior based on Web Services. WS-BPEL provides a language for the specification of Executable and Abstract business processes.</td>
<td>U I</td>
</tr>
<tr>
<td>OASIS/W3C - Web Services Distributed Management (WSDM): Management Using Web Services (MUWS) v1.1</td>
<td>MUWS defines how an IT resource connected to a network provides manageability interfaces such that the IT resource can be managed locally and from remote locations using Web services technologies.</td>
<td>U I I U U</td>
</tr>
<tr>
<td>OASIS WSDM: Management of Web Services (MOWS) v1.1</td>
<td>This part of the WSDM specification addresses management of the Web</td>
<td>U I I U U</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td><strong>OASIS Web Services Dynamic Discovery (WS-Discovery) v1.1</strong></td>
<td>This specification defines a discovery protocol to locate services. The primary scenario for discovery is a client searching for one or more target services.</td>
<td>SO  DP  DC  BDAP  BDFP  S&amp;P  M</td>
</tr>
<tr>
<td><strong>OASIS Web Services Federation Language (WS-Federation) v1.2</strong></td>
<td>This specification defines mechanisms to allow different security realms to federate, such that authorized access to resources managed in one realm can be provided to security principals whose identities and attributes are managed in other realms.</td>
<td>SO  DP  DC  BDAP  BDFP  S&amp;P  M</td>
</tr>
<tr>
<td><strong>OASIS Web Services Notification (WSN) v1.3</strong></td>
<td>WSN is a family of related specifications that define a standard Web services approach to notification using a topic-based publish/subscribe pattern.</td>
<td>SO  DP  DC  BDAP  BDFP  S&amp;P  M</td>
</tr>
<tr>
<td><strong>IETF Simple Network Management Protocol (SNMP) v3</strong></td>
<td>SNMP is a series of IETF sponsored standards for remote management of system/network resources and transmission of status regarding network resources. The standards include definitions of standard</td>
<td>SO  DP  DC  BDAP  BDFP  S&amp;P  M</td>
</tr>
<tr>
<td>Standard Name/Number</td>
<td>Description</td>
<td>NBDRA Components</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td>management objects along with security controls.</td>
<td>SO</td>
</tr>
<tr>
<td>IETF Extensible Provisioning Protocol (EPP)</td>
<td>This IETF series of standards describes an application-layer client-server protocol for the provisioning and management of objects stored in a shared central repository. Specified in XML, the protocol defines generic object management operations and an extensible framework that maps protocol operations to objects.</td>
<td>U</td>
</tr>
</tbody>
</table>

**Table Notes:**

SO = System Orchestrator component  
DP = Data Provider component  
DC = Data Consumer component  
BDAP = Big Data Application Provider component  
BDFP = Big Data Framework Provider component  
S&P = Security and Privacy Fabric  
M = Management Fabric

**6.2 Gap in Standards**

The potential gaps in Big Data standardization are provided in this section to describe broad areas that may be of interest to SDOs, consortia, and readers of this document. The list provided below was produced by an ISO/IEC Joint
Technical Committee 1 (JTC1) Study Group on Big Data to serve as a potential guide to ISO in their establishment of Big Data standards activities. The potential Big Data standardization gaps, identified by the study group, described broad areas that may be of interest to this community. These gaps in standardization activities related to Big Data are in the following areas:

1. Big Data use cases, definitions, vocabulary and reference architectures (e.g., system, data, platforms, online/offline)
2. Specifications and standardization of metadata including data provenance
3. Application models (e.g., batch, streaming)
4. Query languages including non-relational queries to support diverse data types (e.g., XML, RDF, JSON, multimedia) and Big Data operations (e.g., matrix operations)
5. Domain-specific languages
6. Semantics of eventual consistency
7. Advanced network protocols for efficient data transfer
8. General and domain specific ontologies and taxonomies for describing data semantics including interoperability between ontologies
9. Big Data security and privacy access controls.
10. Remote, distributed, and federated analytics (taking the analytics to the data) including data and processing resource discovery and data mining
11. Data sharing and exchange
12. Data storage (e.g., memory storage system, distributed file system, data warehouse)
13. Human consumption of the results of big data analysis (e.g., visualization)
14. Energy measurement for Big Data
15. Interface between relational (i.e., SQL) and non-relational (i.e., Not Only or No Structured Query Language [NoSQL]) data stores
16. Big Data quality and veracity description and management

6.3 Pathway to Address Standards Gaps

Standards often evolve from implementation of best practices and approaches which are proven against real world applications or from theory that is tuned to reflect additional variables and conditions uncovered during implementation. In the case of Big Data, most standards are evolving from existing standards modified to address the unique characteristics of Big Data. Like many terms that have come into common usage in the current information age, Big Data has many possible meanings depending on the context from which it is viewed. Big Data discussions are complicated by the lack of accepted definitions, taxonomies, and common reference views. The products of the NBD-PWG are designed to specifically address the lack of consistency. Recognizing this lack of a common framework on which to build standards, ISO/IEC JTC1 has specifically charted a working group, which will first focus on developing common definitions and a reference architecture. Once established, the definitions and reference architecture will form the basis for evolution of existing standards to meet the unique needs of Big Data and evaluation of existing implementations and practices as candidates for new Big Data related standards. In the first case, existing standards efforts may address these gaps by either expanding or adding to the existing standard to accommodate Big Data characteristics or developing Big Data unique profiles within the framework of the existing standards. The exponential
growth of data is already resulting in the development of new theories addressing topics from synchronization of data across large distributed computing environments to addressing consistency in high volume and velocity environments. As actual implementations of technologies are proven, reference implementations will evolve based on community accepted open source efforts.

**Acronyms A: Acronyms**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMQP</td>
<td>Advanced Message Queuing Protocol</td>
</tr>
<tr>
<td>ANSI</td>
<td>American National Standards Institute</td>
</tr>
<tr>
<td>API</td>
<td>application programming interface</td>
</tr>
<tr>
<td>AVDL</td>
<td>Application Vulnerability Description Language</td>
</tr>
<tr>
<td>BDAP</td>
<td>Big Data Application Provider component</td>
</tr>
<tr>
<td>BDFP</td>
<td>Big Data Framework Provider component</td>
</tr>
<tr>
<td>BIAS</td>
<td>Biometric Identity Assurance Services</td>
</tr>
<tr>
<td>CGM</td>
<td>Computer Graphics Metafile</td>
</tr>
<tr>
<td>CIA</td>
<td>confidentiality, integrity, and availability</td>
</tr>
<tr>
<td>CMIS</td>
<td>Content Management Interoperability Services</td>
</tr>
<tr>
<td>CPR</td>
<td>Capability Provider Requirements</td>
</tr>
<tr>
<td>DC</td>
<td>Data Consumer component</td>
</tr>
<tr>
<td>DCAT</td>
<td>Data Catalog Vocabulary</td>
</tr>
<tr>
<td>DCR</td>
<td>Data Consumer Requirements</td>
</tr>
<tr>
<td>DOM</td>
<td>Document Object Model</td>
</tr>
<tr>
<td>DP</td>
<td>Data Provider component</td>
</tr>
<tr>
<td>DSML</td>
<td>Directory Services Markup Language</td>
</tr>
<tr>
<td>DSR</td>
<td>Data Source Requirements</td>
</tr>
<tr>
<td>DSS</td>
<td>Digital Signature Service</td>
</tr>
<tr>
<td>EPP</td>
<td>Extensible Provisioning Protocol</td>
</tr>
<tr>
<td>EXI</td>
<td>Efficient XML Interchange</td>
</tr>
</tbody>
</table>

https://semanticommunity.info/Data_Science/Data_Science_for_NIST_Big_Data_Framework/NIST_Big_Data_Framework

Updated: Sat, 14 Sep 2019 05:26:25 GMT

Powered by mindtouch
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeoXACML</td>
<td>Geospatial eXtensible Access Control Markup Language</td>
</tr>
<tr>
<td>GML</td>
<td>Geography Markup Language</td>
</tr>
<tr>
<td>GRC</td>
<td>governance, risk management, and compliance</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language</td>
</tr>
<tr>
<td>IEC</td>
<td>International Electrotechnical Commission</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>INCITS</td>
<td>International Committee for Information Technology Standards</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>IT</td>
<td>information technology</td>
</tr>
<tr>
<td>ITL</td>
<td>Information Technology Laboratory</td>
</tr>
<tr>
<td>ITS</td>
<td>Internationalization Tag Set</td>
</tr>
<tr>
<td>JPEG</td>
<td>Joint Photographic Experts Group</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>JSR</td>
<td>Java Specification Request</td>
</tr>
<tr>
<td>JTC1</td>
<td>Joint Technical Committee 1</td>
</tr>
<tr>
<td>LMR</td>
<td>Lifecycle Management Requirements</td>
</tr>
<tr>
<td>M</td>
<td>Management Fabric</td>
</tr>
<tr>
<td>MFI</td>
<td>Metamodel Framework for Interoperability</td>
</tr>
<tr>
<td>MOWS</td>
<td>Management of Web Services</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Picture Experts Group</td>
</tr>
<tr>
<td>MQTT</td>
<td>Message Queuing Telemetry Transport</td>
</tr>
<tr>
<td>MUWS</td>
<td>Management Using Web Services</td>
</tr>
<tr>
<td>NARA</td>
<td>National Archives and Records Administration</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>NASA</td>
<td>National Aeronautics and Space Administration</td>
</tr>
<tr>
<td>NBD-PWG</td>
<td>NIST Big Data Public Working Group</td>
</tr>
<tr>
<td>NCAP</td>
<td>Network Capable Application Processor</td>
</tr>
<tr>
<td>netCDF</td>
<td>network Common Data Form</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>NoSQL</td>
<td>Not Only or No Structured Query Language</td>
</tr>
<tr>
<td>NSF</td>
<td>National Science Foundation</td>
</tr>
<tr>
<td>OASIS</td>
<td>Organization for the Advancement of Structured Information Standards</td>
</tr>
<tr>
<td>OData</td>
<td>Open Data</td>
</tr>
<tr>
<td>ODMS</td>
<td>On Demand Model Selection</td>
</tr>
<tr>
<td>OGC</td>
<td>Open Geospatial Consortium</td>
</tr>
<tr>
<td>OpenMI</td>
<td>Open Modelling Interface Standard</td>
</tr>
<tr>
<td>OR</td>
<td>Other Requirements</td>
</tr>
<tr>
<td>OWS Context</td>
<td>Web Services Context Document</td>
</tr>
<tr>
<td>P3P</td>
<td>Platform for Privacy Preferences Project</td>
</tr>
<tr>
<td>PICS</td>
<td>Platform for Internet Content Selection</td>
</tr>
<tr>
<td>POWDER</td>
<td>Protocol for Web Description Resources</td>
</tr>
<tr>
<td>RDF</td>
<td>Resource Description Framework</td>
</tr>
<tr>
<td>RFID</td>
<td>radio frequency identification</td>
</tr>
<tr>
<td>RIF</td>
<td>Rule Interchange Format</td>
</tr>
<tr>
<td>S&amp;P</td>
<td>Security and Privacy Fabric</td>
</tr>
<tr>
<td>SAF</td>
<td>Symptoms Automation Framework</td>
</tr>
<tr>
<td>SAML</td>
<td>Security Assertion Markup Language</td>
</tr>
<tr>
<td>SDOs</td>
<td>standards development organizations</td>
</tr>
<tr>
<td>SFA</td>
<td>Simple Features Access</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SKOS</td>
<td>Simple Knowledge Organization System Reference</td>
</tr>
<tr>
<td>SLAs</td>
<td>service-level agreements</td>
</tr>
<tr>
<td>SML</td>
<td>Service Modeling Language</td>
</tr>
<tr>
<td>SNMP</td>
<td>Simple Network Management Protocol</td>
</tr>
<tr>
<td>SO</td>
<td>System Orchestrator component</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>SPR</td>
<td>Security and Privacy Requirements</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SWE</td>
<td>Sensor Web Enablement</td>
</tr>
<tr>
<td>SWS</td>
<td>Search Web Services</td>
</tr>
<tr>
<td>TEDS</td>
<td>Transducer Electronic Data Sheet</td>
</tr>
<tr>
<td>TJS</td>
<td>Table Joining Service</td>
</tr>
<tr>
<td>TPR</td>
<td>Transformation Provider Requirements</td>
</tr>
<tr>
<td>TR</td>
<td>Technical Report</td>
</tr>
<tr>
<td>UBL</td>
<td>Universal Business Language</td>
</tr>
<tr>
<td>UDDI</td>
<td>Universal Description, Discovery and Integration</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UIMA</td>
<td>Unstructured Information Management Architecture</td>
</tr>
<tr>
<td>UOML</td>
<td>Unstructured Operation Markup Language</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>WCPS</td>
<td>Web Coverage Processing Service Interface</td>
</tr>
<tr>
<td>WCS</td>
<td>Web Coverage Service</td>
</tr>
<tr>
<td>WFS</td>
<td>Web Feature Service</td>
</tr>
<tr>
<td>WMS</td>
<td>Web Map Service</td>
</tr>
<tr>
<td>WPS</td>
<td>Web Processing Service</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>WS-BPEL</td>
<td>Web Services Business Process Execution Language</td>
</tr>
<tr>
<td>WS-Discovery</td>
<td>Web Services Dynamic Discovery</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Services Description Language</td>
</tr>
<tr>
<td>WSDM</td>
<td>Web Services Distributed Management</td>
</tr>
<tr>
<td>WS-Federation</td>
<td>Web Services Federation Language</td>
</tr>
<tr>
<td>WSN</td>
<td>Web Services Notification</td>
</tr>
<tr>
<td>XACML</td>
<td>eXtensible Access Control Markup Language</td>
</tr>
<tr>
<td>XDM</td>
<td>XPath Data Model</td>
</tr>
<tr>
<td>X-KISS</td>
<td>XML Key Information Service Specification</td>
</tr>
<tr>
<td>XKMS</td>
<td>XML Key Management Specification</td>
</tr>
<tr>
<td>X-KRSS</td>
<td>XML Key Registration Service Specification</td>
</tr>
<tr>
<td>XMI</td>
<td>XML Metadata Interchange</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>XSLT</td>
<td>XSL Transformations</td>
</tr>
</tbody>
</table>

### Appendix B: References

#### General Resources

- Institute of Electrical and Electronics Engineers (IEEE). [https://www.ieee.org/index.html](https://www.ieee.org/index.html)
Document References

