**NIST Big Data**

**Reference Architecture**

DRAFT

**Version 0.5**

**Reference Architecture Subgroup**

**NIST Big Data Working Group (NBD-WG)**

**September, 2013**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Changes abstract | References | Editor |
| 0.1 | 9/6/13 | Outline and Diagram | M0202v1 | Orit |
| 0.2 | 9/11/13 | Text from input docs | M0142v3, M0189v1, M0015v1, M0126v4, M0231v1 | Orit |
| 0.3 |  | - Additions to the diagram based on a weekly conference call  - Clarification to the RA model based on email conversation  - Intro text is added | M0230v2, M0239v1 | Orit |
| 0.4 |  | - Security and Management are shown as the fabrics around all blocks  - Transformation sub-blocks are shown with same color  - Editorial changes to Section 3 (as an input to roadmap) | Weekly conf call. M0230v3 | Orit |
| 0.5 |  | - Multiple RA views for discussion (open the Figure as an object)  - Appendices A and C  - Additions to Executive Summary  - New: Management Lifecycle | M0243v2, M0247v1, M0249v1  Input from Gary Mazzaferro | Orit |
|  |  |  |  |  |
|  |  |  |  |  |

[Executive Summary 4](#_Toc367825792)

[1 Introduction 4](#_Toc367825793)

[1.1 Background 4](#_Toc367825794)

[1.2 Objectives 5](#_Toc367825795)

[1.3 How This Report Was Produced 6](#_Toc367825796)

[1.4 Structure of This Report 6](#_Toc367825797)

[2 Big Data System Requirements 7](#_Toc367825798)

[3 Conceptual Model 7](#_Toc367825799)

[4 Main Components 9](#_Toc367825800)

[4.1 Data Provider 9](#_Toc367825801)

[4.1.1 Data Service Abstraction 9](#_Toc367825802)

[4.2 Transformation Provider 10](#_Toc367825803)

[4.2.1 System Service Abstraction 11](#_Toc367825804)

[4.2.2 Usage Service Abstraction 11](#_Toc367825805)

[4.3 Capabilities Provider 11](#_Toc367825806)

[4.3.1 Capabilities Service Abstraction 11](#_Toc367825807)

[4.4 Data Consumer 12](#_Toc367825808)

[4.5 System Orchestrator 12](#_Toc367825809)

[5 Management 13](#_Toc367825810)

[5.1 System Management 13](#_Toc367825811)

[5.2 Lifecycle Management 13](#_Toc367825812)

[6 Security and Privacy 14](#_Toc367825813)

[7 Big Data Taxonomy 14](#_Toc367825814)

[Appendix A: Terms and Definitions 14](#_Toc367825815)

[Appendix B: Acronyms 16](#_Toc367825816)

[Appendix C: References 16](#_Toc367825817)

# Executive Summary

# Introduction

## Background

Big Data is the common term used to describe the deluge of data in our networked, digitized, sensor-laden, information driven world. The availability of vast data resources carries the potential to answer questions previously out of reach. Questions like: How do we reliably detect a potential pandemic early enough to intervene? Can we predict new materials with advanced properties before these materials have ever been synthesized? How can we reverse the current advantage of the attacker over the defender in guarding against cyber-security threats?

Within this context, on 29 March, 2012 The White House announced the Big Data Research and Development Initiative[[1]](#footnote-1). The initiative’s goals were to help accelerate the pace of discovery in science and engineering, strengthen the national security, and transform teaching and learning by improving our 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 80+ projects – that aimed to significantly improve the tools and techniques needed to access, organize, and glean discoveries from huge volumes of digital data. The initiative also challenged industry, research universities, and non-profits to join with the Federal government to make the most of the opportunities created by Big Data.

Despite the widespread agreement on the opportunities inherent to Big Data, a lack of consensus on some important, fundamental questions continues to confuse potential users and hold back progress. What are the attributes that define Big Data solutions? How is Big Data different from the traditional data environments and related applications we have encountered thus far? 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?

The NIST Big Data program was formally launched on 13 June, 2012 to help answer some of the questions surrounding Big Data and to support the federal government effort to incorporate Big Data as a replacement for, or enhancement to, traditional data analysis systems and models where appropriate.

**[Editor’s Note:** Need some transition verbiage here. How did the first conference lead to the BD-PWG?]

On 19 June, 2013 NIST hosted the Big Data Public Working Group (BD-PWG) kickoff meeting to begin addressing those questions. The Group was charged with developing a consensus definition, taxonomy, reference architecture, and technology roadmap for Big Data that can be embraced by all sectors.

These efforts will help define and prioritize requirements for *interoperability*, *portability*, *reusability*, and *extendibility* for Big Data analytic techniques and technology infrastructure in order to support secure and effective adoption of Big Data.

The aim is to create vendor-neutral, technology and infrastructure agnostic deliverables to enable Big Data stakeholders to pick-and-choose best analytics 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.

Within the BD-PWG, the following working groups were chartered in order to provide a technically-oriented strategy and standards-based guidance for the federal Big Data implementation effort:

* Definitions and Taxonomies
* General Requirements
* Security and Privacy Requirements
* Reference Architectures
* Technology Roadmap

## Objectives

In general terms, a reference architecture provides “an authoritative source of information about a specific subject area that guides and constrains the instantiations of multiple architectures and solutions”[[2]](#footnote-2). Reference architectures generally serve as a reference foundation for solution architectures and may also be used for comparison and alignment purposes.

The broad goal of the Reference Architecture working group is to develop a Big Data open reference architecture that:

1. Provides a common language for the various stakeholders
2. Encourages adherence to common standards, specifications, and patterns
3. Provides consistency of implementation of technology to solve similar problem sets

The reference architecture 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; instead it is a tool for describing, discussing, and developing system-specific architectures using a common framework of reference.

It provides 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 NIST Big Data reference architecture serves the following objectives:

1. To illustrate and understand the various Big Data components, processes, and systems, in the context of an overall Big Data conceptual model;
2. To provide a technical reference for U.S. Government departments, agencies and other consumers to understand, discuss, categorize and compare Big Data solutions; and
3. To facilitate the analysis of candidate standards for interoperability, portability, reusability, and extendibility.

The design of the Big Data reference architecture does not address the following:

1. Detailed specifications for any organizations’ operational systems;
2. Detailed specifications of information exchanges or services; or
3. Recommendations or standards for integration of infrastructure products.

It is important to note that at this time, the Big Data reference architecture is not complete. Many sections of this document are still under development.

## How This Report Was Produced

The approach for developing this document involved four steps:

1. The first step was to announce a Big Data Reference Architecture Working Group open to the public in order to attract and solicit a wide array of subject matter experts and stakeholders in government, industry, and academia.
2. The second step was to gather available Big Data architectures and materials representing various stakeholders, different data types, and different use cases.
3. The third step was to examine and analyze the Big Data materiel to better understand existing concepts of Big Data, what it is used for, its goals, objectives, characteristics, and key elements, and then document the these using the Big Data taxonomies model.
4. The fourth step was to develop an open reference architecture based on the analysis of Big Data material and the inputs from the other NIST Big Data working groups.

## Structure of This Report

The remainder of this document is organized as follows:

Section 2 contains high level requirements relevant to the design of the Reference Architecture.

Section 3 represents a generic big data system comprised of technology-agnostic function blocks interconnected by interoperability surfaces.

Section 4 describes the main components of the generic system.

Section 5 addresses security and privacy.

Section 6 contains the Big Data taxonomy.

Appendix A lists the terms and definitions appearing in the taxonomy.

Appendix B contains the acronyms used in this document.

Appendix C lists the references used in the document.

# Big Data System Requirements

**[This section contains high level requirements relevant to the design of the Reference Architecture. This section will be further developed by the NIST BDWG Requirements SG.]**

The “big data” ecosystem is an evolution and a superset of a “traditional data” system exhibiting any or all of the following characteristics or requirements:

* Data sources are diverse in their security and privacy considerations and their business relationships with the data system integrators
* Data imported into the system vary in its structure, exhibit large volume, velocity, variety, and other complex properties
* The nature and the order of data transformations varies between vertical systems; it is not prearranged, and is evolving for a given system
* Storage technologies and databases are tailored to specific transformation needs and their scalability properties scale horizontally, vertically, or both
* Innovative analytic functions continuously emerge; proven technologies get enhanced and abstracted, resulting in frequent updates and outsourcing practices
* Data usage vary in structure and format; new use cases can be easily introduced to the system

# Conceptual Model

The NIST Big Data Reference Architecture (RA) shown on Figure 1 represents a generic business-neutral big data system comprised of technology-agnostic blocks, each representing a defined functional role in the Big Data ecosystem. The functional blocks are interconnected by interoperability interfaces.

According to the big data taxonomy, a single actor can play multiple roles, and multiple actors can play the same role. This RA doesn’t specify the business boundaries between the participating stakeholders or actors, indicating that each two roles can reside within the same business entity or can be implemented by different business entities. As such, the RA is applicable to a variety of business environments including tightly-integrated enterprise systems, as well as loosely-coupled vertical industries that rely on cooperation of independent stakeholders.

Note that as a result, the notion of internal vs. external functional blocks or roles doesn’t apply to this RA. However, for a specific use case, once the roles are associated with specific business stakeholders, the functional blocks would be considered as internal or external - subject to the use case’s point of view.



Figure 1: Big Data Reference Architecture

The RA is organized around two axes representing the two big data value chains: the information flow (along the vertical axis) and the IT integration (along the horizontal axis). Along the information flow 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 and operating of the big data in support of data transformations required for implementation of a specific application or a vertical. Note that the transformation block is at the intersection of both axes indicating that data analytics and its implementation are of special value to big data stakeholders in both value chains.

The five main RA blocks represent different technical roles that exist in every big data system: “Data Provider”, “Data Consumer”, “Transformation Provider”, “Capabilities Provider”, and “System Orchestrator”. The two additional “Security & Privacy” and “Management” blocks are shown as the fabrics enclosing all sub-systems thus providing services and functionality to the rest of the system components in the areas specific to “big data”.

Note that this RA supports the representation of stacking or chaining of big data systems, in a sense that a Data Consumer of one system could serve as a Data Provider to the next system down the stack or chain.

The “DATA” arrows show the flow of data between the system’s main blocks. The data flows between the components either physically (a.k.a., by value) or by providing its location and the means to access it (a.k.a., by reference). The “SW” arrows show transfer of software tools for processing of big data *in situ*. The “Service Abstraction” blocks represent software programmable interfaces representing functional abstractions. Manual agreements (e.g., SLAs) and human interactions that may exist throughout the system are not shown in the RA.

# Main Components

## Data Provider

Data Provider is the role of introducing new information feeds into the big data system for discovery, access, and transformation by the big data system.

Note that new data feeds are distinct from the data already being in use by the system and residing in the various system repositories although similar technologies can be used to access both.

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 and public records, online or offline applications, tapes, images/audio and videos, sensor data, Web logs, HTTP cookies, etc. Data sources can be produced by humans, machines, sensors, Internet technologies, and so on.

In its role, Data Provider creates an abstraction of data sources. In case of raw data sources, Data provider can potentially cleanse, correct, and store the data an internal format that is accessible to the big data system that will ingest it.

Frequently, the role of Data Provider and Transformation Provider would belong to different authorities, unless the authority implementing the Transformation Provider owns the data sources. Consequently, data from different sources may have different security and privacy considerations.

Data Provider can also provide an abstraction of data being transformed earlier by another system that can be either a legacy system or another big data system. In this case, Data Provider would represent a Data Consumer of that other system. For example, a (big) streaming data source could be generated by another system operation on (big) data at rest.

Data Provider activities include:

* Creating the metadata describing the data source(s), usage policies/access rights, and other relevant attributes
* Publishing the availability of the information and the means to access it
* Making the data accessible by other RA components using suitable programmable interface.
* Enforcing access rights on data access

### Data Service Abstraction

The Data Service Abstraction would typically include a registry so that the transformation functions can locate a Data provider, identify what useful comparable data it contains, understand what types of access are allowed, understand what types of analysis are supported, where the data source is located, how to access the data, security requirements for the data, privacy requirements for the data, etc. As such, the interface would include the means for registering the data source, for querying the registry, and a standard set of data contained by the registry.

Because the data can be too large to economically move across the network, the interface could also allow the submission of analysis requests (as a software code implementing a certain algorithm for execution) with the results returned to the requestor.

Subject to data characteristics (such as, volume, velocity, and variety) 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.*

Note that not all use of data service abstraction would always be automated, but rather might involve a human role logging into the system and providing directions where new data should be transferred (for example, via FTP).

## Transformation Provider

Transformation Provider is the role of executing a generic “vertical system” data life cycle including data collection from various sources, multiple data transformations being implemented using both traditional and new technologies, and diverse data usage.

The Transformation functions would typically be specific to the vertical application and therefore are not candidates for standardization. However, the metadata and the policies defined and exchanged between the transformation blocks could be standardized.

As the data propagates through the ecosystem, it is being processed and transformed in different ways in order to extract the value from the information. Transformation sub-components can be implemented by independent stakeholders and deployed as stand-alone services.

Each transformation function can use different specialized data infrastructure or capabilities best suited for its requirements, and can have its own privacy and other policy considerations.

In its role, Transformation Provider typically executes the manipulations of the data lifecycle of a specific vertical system to meet the requirements or instructions established by the System Orchestrator.

**[Editor’s Note: Listed activities need to be aligned with the sub-components shown on the diagram.]**

Transformation Provider activities include:

* Data Collection (connect, transport, stage): obtains connection to Data Provider APIs to collect into local system, or to access dynamically when requested. At the initial collection stage, sets of data (e.g., data records) 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).
* Data Curation: provides cleansing, outlier removal, standardization for the ingestion and storage processes.
* Data Aggregation: Data from different data providers with easily correlated metadata (e.g., identical keys) is aggregated into a single data set. 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 resultant data set are typically similar to the original data.
* Data Matching: Data from different data providers with disparate metadata (e.g., keys) are aggregated into a single data set. As a result, the information about each object is enriched. The security considerations and policies concerning the resultant data set may differ from the original policies.
* Data Optimizion (Pre-analytics): determines the appropriate data manipulations and indexes to optimize subsequent transformation processes.
* Data Analysis: implements the techniques to extract knowledge from the data based on the requirements of the data scientist, who has specified the algorithms to process the data to produce new insights that will address the technical goal.
* Data Transfer: facilitates secure transfer of data between different repositories and/or between the Transformation and the Capabilities RA blocks.

While many of these tasks have traditionally existed in data processing systems, the scale, velocity and variety present in big data systems radically changes their implementation. The algorithms and mechanisms need to be re-written and optimized to create applications that are responsive and can grow to handle ever growing data collections.

### System Service Abstraction

TBD

### Usage Service Abstraction

The Transformation Provider capabilities will comprise a collection of application-specific services for processing the data that resides in the big data system. These services, as defined by the Usage Service Abstraction, can be called and composed by 3rd party applications that have the permissions to consume the data. While the specific of this service will necessarily be application-specific, some commonality will exist in a number of areas, including:

Identity Management and Authorization: Individual vertical Transformation Providers will implement their own schemes for usage of their services. Identity management enables Transformation Providers to implement charging schemes, provide levels of differentiated service and protect confidential services from unauthorized access.

Discovery: Data consumers require a directory that defines the services that a Transformation Provider can support.

Code execution services: A Transformation Provider may allow data consumers to push analytics in the form of a code to execute on the big data system. The Usage Services will define the precise form that these requests support, for example the software languages that can be used, the constraints (eg execution time) on the code that is provided to the service, and how the results will be delivered to the data consumer.

Charging: Transformation Providers may implement charging schemes to generate revenue from data consumers. The Usage Services will enable users to discover the amounts they are being charged, monitor usage and make payments.

## Capabilities Provider

Capabilities Provider is the role of providing a computing fabric (such as system hardware, network, storage, virtualization, and computing platform) in order to execute certain transformation applications, while protecting the privacy and integrity of data. The computing fabric facilitates a mix-and-match of traditional and future computing features from software, platforms, and infrastructures based on application needs.

Capabilities are abstracted functionalities that exist in order to support big data transformation functions. Capabilities include infrastructures (e.g., VM clusters), platforms (e.g., databases), and applications (e.g., analytic tools).

**[Editor’s Note: Activities need to be listed and described after the agreement is reached on the sub-components presented on the diagram.]**

### Capabilities Service Abstraction

Big data applications, implemented with the Transformation Provider, require the capabilities of many core platforms and technologies to meet their challenges of scalable data analytics and management. The specific capabilities vary widely and different vertical application domains will utilize a variety of technologies to meet their functional and cost requirements.

A broad categorization of the capabilities services that will be supported in a big data system are as follows:

Data Services: A big data system will expose its data resources through a set of services that can be invoked by a Transformation Provider. The nature and granularity of these services will be application-specific, but generally provide standard CRUD (create/read/update/delete) functionality. The services should be designed to efficiently support application requests, and commonly one service will invoke a cascade of internal capability service invocations to access a multitude of individual big data collections. As big data is often replicated, data services may expose functions that enable an application to explicitly trade-off consistency and latency to more efficiently satisfy a request at the risk of obtaining stale data or performing inconsistent updates.

Security Services: The capability provider must expose services to perform identity management and provide authentication and authorization of the data and processing resources that are encompassed. This ensures resources are protected from unauthorized access and protects from tampering. This can be a particularly challenging area for big data systems that integrate heterogeneous data resources and/or execute on cloud platforms.

Management Services: A big data system will have many ‘moving parts’ that must be managed. Automation is a fundamental principle in building big data systems, and hence management services are an integral component of the capabilities service abstraction. Management capabilities range from VM deployment and recovery, to fine grained monitoring of system performance and detecting and diagnosing faults.

Test Services: A unique characteristic of big data systems is that it is impossible to fully test application changes before deployment, as the scale of the data and processing environment precludes exhaustive testing in an isolated environment. For this reason, big data systems must provide services to support testing of new features in a production environment. Techniques such as canary testing and A/B testing are widely used, and require the ability to reconfigure the big data platform to direct percentages of live requests to test components, and provide detailed information and logging from the components under test.

Processing Services: Supporting in situ processing allows Transformation Providers to push analytics to be performed by Capability Providers. To achieve this, services must be provided to accept the code for the analytics, execute the code in a protected environment, and return the results to the user. The latter is typically achieved asynchronously as many such analytics will be long running tasks that process many millions of data items.

## Data Consumer

Data Consumer is the role performed by end users or other systems in order to use the results of data transformation. Data Consumer uses the Usage Service Abstraction interface to get access to the information of its interest. Usage Services Abstraction can include data reporting, data retrieval, and data rendering.

Data Consumer activities can include:

* Exploring data using data visualization software
* Ingesting data into their own system
* Putting data to work for the business, for example to convert knowledge produced by the transformers into business rule transformation

Data Consumer can play the role of the Data Provider to the same system or to another system. Data Consumer can provide requirements to the System Orchestrator as a user of the output of the system, whether initially or in a feedback loop.

## System Orchestrator

System Orchestrator is the role of defining and integrating the required data transformations components into an operational vertical system. Typically, System Orchestrator would represent a collection of more specific roles performed by one or more actors, which manages and orchestrates the operation of the Big Data System.

The Big Data RA represents a broad range of big data systems: from tightly-coupled enterprise solutions (integrated by standard or proprietary interfaces) to loosely-coupled verticals maintained by a variety of stakeholders or authorities 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, business requirements, etc. System Governor works with a collection of other roles (such as Data Manager, Data Security, and System Manager) to implement the requirements and the system’s functionality.

In a loosely-coupled vertical, the System Orchestrator role is typically decentralized. Each independent stakeholder is responsible for its system management, security, and integration. In this situation, each stakeholder is responsible for integration within the big data distributed system using service abstractions provided by other stakeholders.

In both cases (i.e., tightly and loosely coupled), the role of the System Orchestrator can include the responsibility for

* Translating business goal(s) into technical requirements.
* Supplying and integrating with both external and internal Providers.
* Overseeing evaluation of data available from Data Providers.
* Directing the Transformation Provider by establishing requirements for the collection, curation, analysis of data, etc.
* Overseeing transformation activities for compliance with requirements.

# Management

## System Management

TBD.

## Lifecycle Management

Lifecycle Management is responsible of managing data coming into the system, residing within the system, and going out of the system for application usage. In other words, the role of Lifecycle Management is to ensure that the data are accessible by other Provider Components throughout the lifecycle of the data, since the moment they are ingested into the system by the Data Provider, and until the data are dispositioned. Moreover, this accessibility has to comply with policies, regulations, and security requirements. In the context of Big Data, Lifecycle Management has to deal with the three V characteristics: Volume, Velocity and Variety. As such, Lifecycle Management and its components will have to interact with other components of the Big Data Reference Architecture, such as Capability Provider, Transformation Provider, and System Orchestrator.

Lifecycle Management activities include:

* Metadata Management. Metadata Management is the enabler of Lifecycle Management, since metadata are used to store information that governs the lifecycle management of the data within the system. Metadata also contains critical information such as persistent identification of the data, the fixity, and the access rights.
* Accessibility Management
  + Data Masking for security privacy. Privacy information has to be anonymized prior to the data analytics process. For instance, demographic data can be aggregated and analyzed to reveal data trends, but specific personal identifiable information (PII) with names and social security numbers have to be masked. This masking managed by Lifecycle Management depends on the type of application usage and the authorization usage specified by Security and Privacy.
  + Accessibility of data may change over time. For instance, Census data can be made available to the public after 75 years. In that case, Lifecycle Management is responsible of triggering the update of the accessibility of the data or sets of data according to the policy and legal requirements. Normally, data accessibility information is stored in the metadata.
  + Data Recovery. Data management should also include recovering data that were lost due to disaster, or system/storage fault. Traditionally, this data recovery can be achieved using backup and restore mechanisms. But, in order to cope with the large volume of Big Data, this should be embedded in the architectural design, and the exploitation of modern technologies within the Big Data Capability Provider.

Preservation Management. At the basic level, the system needs to ensure the integrity of the data 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 to disposition aged data contained in the system. Depending on the retention policy, these aged data can be deleted or migrated to archival storage. On the other hand, in the case where data need to be retained for years, decades and even centuries, there will be a need to have a preservation strategy so they can be accessed by the Provider Components if required. This will invoke the so-called long-term digital preservation that can be performed by Transformation Provider using the resources in Capability Provider.

In order to perform its activities, Lifecycle Management will interact with the other Provider Components of the Big Data Reference Architecture:

* Data Provider to manage the metadata from the entry of data into the Big Data system;
* Transformation Provider to perform data masking and format transformations for preservation purpose;
* Capabilities Provider to perform basic bit-level preservation and data recovery;

Security and Privacy to keep the data management up to date according to new security policy and regulations. In the other direction, Security and Privacy also utilizes information coming from Lifecycle Management with respect to data accessibility. Assuming that Security and Privacy controls access to the functions and data usage produces by the Big Data system, this data access control can be informed by the metadata managed and updated by Lifecycle Management.

# Security and Privacy

**[This section will be prepared by the NIST BDWG Security and Privacy SG and will contain high level security and privacy considerations relevant to the design of the Reference Architecture.]**

# Big Data Taxonomy

**[This section will be prepared by the NIST BDWG Def&Tax SG and will contain high level taxonomy relevant to the design of the Reference Architecture.]**

# Appendix A: Terms and Definitions

===============================================================

**First Level Terms:**

Big data - 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.

Transformation Provider – Organization or entity that executes a generic “vertical system” data life cycle, 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.

System Orchestrator – Organization or entity that defines and integrates the required data transformations components into an operational vertical system.

Capabilities Provider – Organization or entity that provides a computing fabric (such as system hardware, network, storage, virtualization, and computing platform) to execute certain transformation applications, while maintaining security and privacy requirements.

Data Consumer - End users or other systems that use the results of data transformations.

**===============================================================**

**Second Level Terms:**

Data Service Abstraction – The interface for both registering data sources and querying the registry so that transformation functions can locate a data provider, identify what comparable data it contains, understand what types of access is allowed, understand what types of analysis are supported, where the data source is located, how to access the data, security requirements for the data, privacy requirements for the data, etc.

Usage Service Abstraction –

System Service Abstraction –

Capabilities Service Abstraction –

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.

Reusability –

Extendability -

Security –Protecting data, information, and systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide:

(a) integrity: guarding against improper data modification or destruction, and includes ensuring data nonrepudiation and authenticity;

(b) confidentiality: preserving authorized restrictions on access and disclosure, including means for protecting personal privacy and proprietary data;

(c) availability: ensuring timely and reliable access to and use of data.

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 life cycle.

**===============================================================**

**Third Level Terms:**

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)

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)

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). (Source: NIST CC Definition)

# Appendix B: Acronyms

# Appendix C: References

The lists below provide examples of resources that may be helpful.

[1] White House Press Release, “Obama Administration Unveils “Big Data” Initiative”, 29 March 2012, <http://www.whitehouse.gov/sites/default/files/microsites/ostp/big_data_press_release_final_2.pdf>

[2] White House, “Big Data Across The Federal Government”, 29 March 2012, <http://www.whitehouse.gov/sites/default/files/microsites/ostp/big_data_fact_sheet_final_1.pdf>

[3] NIST, Big Data Workshop, 13 June 2012, <http://www.nist.gov/itl/ssd/is/big-data.cfm>

[4] NIST, Big Data Public Working Group, 26 June 2013, <http://bigdatawg.nist.gov/home.php>

[5] National Science Foundation, “Big Data R&D Initiative”, June 2012, <http://www.nist.gov/itl/ssd/is/upload/NIST-BD-Platforms-05-Big-Data-Wactlar-slides.pdf>

[6] Gartner, “3D Data Management: Controlling Data Volume, Velocity, and Variety”, <http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf>

[7] Gartner, “The Importance of 'Big Data': A Definition”, <http://www.gartner.com/DisplayDocument?id=2057415&ref=clientFriendlyUrl>

[8] Hilbert, Martin and Lopez, Priscilla, “The World’s Technological Capacity to Store, Communicate, and Compute Information”, Science, 01 April 2011

[9] Department of Defense, “Reference Architecture Description”, June 2010, <http://dodcio.defense.gov/Portals/0/Documents/DIEA/Ref_Archi_Description_Final_v1_18Jun10.pdf>

[10] Rechtin, Eberhardt, “The Art of Systems Architecting”, CRC Press; 3rd edition, 06 January 2009

[11] ISO/IEC/IEEE 42010 Systems and software engineering — Architecture description, 24 November 2011, <http://www.iso.org/iso/catalogue_detail.htm?csnumber=50508>



1. http://www.whitehouse.gov/blog/2012/03/29/big-data-big-deal [↑](#footnote-ref-1)
2. Department of Defense Reference Architecture White Paper, June 2010 [↑](#footnote-ref-2)