**NIST Big Data**

**Reference Architecture**

DRAFT

**Version 1.2**

**Reference Architecture Subgroup**

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

**December, 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 | 9/18/13 | - 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 | 9/22/13 | - 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 | 9/24/13 | - 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 |
| 0.6 | 9/25/13 | - Editorial changes  - Interfaces description  - Diagram update | M0252v1 | Orit |
| 0.7 | 9/26/13 | - Diagram update  - New: Deployment Appendix | M0251v1 | Orit |
| 1.0 | 9/29/13 | - Security Fabric description  - High level requirements  - BD Application Provider revision | Input from Arnab Roy  Input from the Reqs subgroup  M0258v1 | Orit |
| 1.1 | 10/15/13 | - Editorial changes | Including the “System Management” baseline text. | Felix Njeh |
| 1.1 | 11/22/13 | - Clarifications based on the mailing list discussions | Replies to Bob Marcus | Orit |
| 1.2 | 12/8/13 | - Editorial changes |  | Pw Carey, Nancy & Orit |

**Table of Contents**

[Executive Summary 4](#_Toc376786279)

[1 Introduction 4](#_Toc376786280)

[1.1 Background 4](#_Toc376786281)

[1.2 Objectives 5](#_Toc376786282)

[1.3 How This Report Was Produced 6](#_Toc376786283)

[1.4 Structure of This Report 6](#_Toc376786284)

[2 Big Data General Requirements 7](#_Toc376786285)

[3 Conceptual Model 8](#_Toc376786286)

[4 Main Components 10](#_Toc376786287)

[4.1 Data Provider 10](#_Toc376786288)

[4.2 Big Data Application Provider 11](#_Toc376786289)

[4.3 Big Data Framework Provider 14](#_Toc376786290)

[4.4 Data Consumer 15](#_Toc376786291)

[4.5 System Orchestrator 16](#_Toc376786292)

[5 Management 16](#_Toc376786293)

[5.1 System Management 16](#_Toc376786294)

[5.2 Lifecycle Management 17](#_Toc376786295)

[6 Security and Privacy 18](#_Toc376786296)

[7 Big Data Taxonomy 18](#_Toc376786297)

[Appendix A: Terms and Definitions 18](#_Toc376786298)

[Appendix B: Acronyms 20](#_Toc376786299)

[Appendix C: References 20](#_Toc376786300)

[Appendix D: Deployment Considerations 21](#_Toc376786301)

[1 Big Data Framework Provider 21](#_Toc376786302)

[1.1 Traditional On-Premise Frameworks 22](#_Toc376786303)

[1.2 Cloud Service Providers 22](#_Toc376786304)

*[1.2.1](#_Toc376786305)* [Cloud Service Component 22](#_Toc376786305)

*[1.2.2](#_Toc376786306)* [Resource Abstraction & Control 23](#_Toc376786306)

*[1.2.3](#_Toc376786307)* [Physical Resources 23](#_Toc376786307)

# 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*, *usability and* *reusability*, and *extendibility* for Big Data analytic techniques and technology infrastructure in order to facilitate the adoption of Big Data.

The aim is to create a 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 functionalities 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, 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:

* Provides a common language for the various stakeholders
* Encourages adherence to common standards, specifications, and patterns
* 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:

* To illustrate and understand the various Big Data components, processes, and systems, in the context of an overall Big Data conceptual model;
* To provide a technical reference for U.S. Government departments, agencies and other consumers to understand, discuss, categorize and compare Big Data solutions; and
* 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:

* Detailed specifications for any organizations’ operational systems;
* Detailed specifications of information exchanges or services; or
* 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 material 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 describes the system and data management considerations.

Section 6 addresses security and privacy.

Section 7 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 General Requirements

There is a two-step process involved with requirement extraction. The first step is to extract specific requirements based on each application’s characteristics, which includes detailed information on

(a) data sources (data size, file formats, rate of grow, at rest or in motion, etc.),

(b) data lifecycle management (curation, conversion, quality check, pre-analytic processing, etc.),

(c) data transformation (data fusion, analytics),

(d) big data framework and infrastructure (software tools, platform tools, hardware resources such as storage and networking), and

(e) data usage (processed results in text, table, visual, and other formats).

The second step is to aggregate each application’s specific requirements into high-level generalized requirements which are vendor-neutral and technology agnostic. For complete use cases’ characteristics and requirements analysis, please refer Big Data Use Cases and Requirements, M0245.

The following are the high-level big data general requirements.

**Data Provider Requirements (DPR)**

DPR-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.

DPR-2: Needs to support slow, bursty and high throughput data transmission between data sources and computing clusters.

DPR-3: Needs to support diversified data content ranging from structured and unstructured text, documents, graphs, web sites, geospatial, compressed, timed, spatial, multimedia, simulation, and instrumental (a.k.a., system managements and monitoring) data.

**Big Data Application Provider Requirements (APR)**

APR-1: Needs to support diversified compute intensive, analytic processing and machine learning techniques

APR-2: Needs to support batch and real time analytic processing

APR-3: Needs to support processing large diversified data content and modeling

APR-4: Needs to support processing data in motion (streaming, fetching new content, data tracking, traceability, data change management, data boundaries, etc.)

**Big Data Framework Provider Requirements (FPR)**

FPR-1: Needs to support legacy software and advanced software packages (software)

FPR-2: Needs to support legacy and advance computing platforms (platform)

FPR-3: Needs to support legacy and advanced distributed computing clusters, co-processors, I/O processing (infrastructure)

FPR-4: Needs to support advanced networks (ex. Software Defined Networks) and elastic data transmission including fiber, cable, and wireless networks, LAN, WAN, MAN and WiFi (networking)

FPR-5: Needs to support legacy, large, virtual and advanced distributed data storage (storage)

FPR-6: Needs to support legacy and advanced programming executable, 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 high 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 & Privacy Requirements (SPR)**

SPR-1: Needs to protect and preserve security and privacy on sensitive data

SPR-2: Needs to support multi-tenant, multi-level policy-driven, sandbox, access control, authentication on protected data in-line with accepted GRC (Governance, Risk & Compliance) and CIA (Confidentiality, Integrity & Availability) best practices

**Lifecycle Management Requirements (LMR)**

LMR-1: Needs to support data quality curation including pre-processing, data clustering, classification, reduction, 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 (including cross-border, geographically dispersed) archival

LMR-8: Needs to support persistent identifier and data traceability

LMR-9: Needs to support standardize, aggregate, and normalize 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 and management

OR-5: Needs to support security across mobile devices and other smart devices such as sensors

# Conceptual Model

The NIST Big Data Reference Architecture (RA) shown on Figure 1 represents an agnostic big data system comprised of logical functional blocks interconnected by interoperability interfaces (a.k.a., services). The blocks represent functional roles in the Big Data ecosystem and are called “Providers” to indicate that they provide or implement a specific technical function within the system.

According to the big data taxonomy, a single actor can play multiple roles, and multiple actors can play the same role. This functional RA doesn’t specify the business boundaries between the participating stakeholders or actors, indicating that such 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 the cooperation by independent stakeholders.

Note: 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. For examples, please turn to Appendix D: Deployment Considerations.



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 applications required for implementation of a vertical. Note that the Big Data Application Provider 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”, “Big Data Application Provider”, “Big Data Framework 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”. These two key functions are crucial and hence are integrated into any Big Data solution.

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 Use” arrows represent software programmable interfaces. While the main focus of the RA 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., 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 data or 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, system and audit 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 Big Data Application Provider would belong to different authorities, unless the authority implementing the Big Data Application 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:

* Collecting the data
* Persisting the data
* 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.
* Providing push or pull access mechanisms
* Enforcing access rights on data access
* Establishing formal or informal contract for data access authorizations
* Providing transformation functions for data scrubbing of implicit or explicit Personally Identifiable Information

The Data Provider will expose a collection of interfaces (or services) for discovering and accessing the data. These services would typically include a registry so that the applications can locate a Data Provider, identify what data of interest 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 data access may not 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).

## Big Data Application Provider

Big Data Application Provider is the role of executing a specific set of data life cycle to meet the requirements established by the System Orchestrator, as well as the Security and Privacy requirements. These processes include data collection from various sources, multiple data transformations being implemented using both traditional and new technologies, and diverse data usage. The Big Data Application Provider activities would typically be specific to the application and therefore are not candidates for standardization. However, especially when the application represents a vertical industry, the metadata and the policies defined and exchanged between the application’s sub-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. Each activity within the Big Data Application Provider can be implemented by independent stakeholders and deployed as stand-alone services. As such, the "Application Provider" can be a single or a collection of more granulate "Application Providers", each implementing different steps in the "data lifecycle".

Each of the steps can run on a separate "Framework Provider" or all can use a common "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. This baseline RA doesn’t show the underlying technologies, business considerations, and topology constraints thus making it applicable for any kind of system approaches and deployments.

For example, the Application Provider's "own" infrastructure would be represented as one of the "Data Framework Providers". If the Application Provider uses "external" / "outsourced" infrastructure(s) as well, it (they) will be represented as another (or multiple) "Data Framework Provider(s)" in the RA. "Framework Providers" are shown as "many" in the RA indicating that here can be as many as are being used for a single "Application Provider".

In its role, Big Data Application 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: The activities below need to be synchronizes with the Def&Tax document.]**

The Big Data Application Provider activities include:

* Collection
  + Obtains connection to Data Provider APIs to connect 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).
* Preparation
  + Prepares the data through cleansing, outlier removal, and standardization for the ingestion and storage processes.
  + Aggregates data from different data providers with easily correlated metadata (e.g., identical keys) 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.
  + Matches data from different data providers with disparate metadata (e.g., keys) 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.
  + Optimizes data by determining the appropriate data manipulations and indexing to optimize subsequent transformation processes.
* Analysis and Analytics
  + 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.
* Visualization
  + Ensures relevant visualization tools are integrated into data life cycle system.
  + Formats and presents data in such a way as to optimally communicate meaning and knowledge.
  + Develops appropriate statistical charts and diagrams to reflect analysis results.
* Access
  + Identifies and stores data in persistent repositories for use, sharing, and re-use.
  + Ensures descriptive, administrative, and preservation metadata and metadata schemes are developed and used.
  + Ensures secure access to data and information.
  + Conducts appropriate curation activities to account for changes in storage media, file formats, etc., over time in order to ensure data access for specified periods.

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.

The Big Data Application Provider will expose a collection of interfaces (or services) for consuming the results of the data processing performed by the big data system. These interfaces can be called and composed by 3rd party applications that have the permissions to consume the data. While the specifics of this interface may be application-specific, commonality will exist in a number of areas, including:

Data Transfer: Facilitates secure transfer of data between different repositories and/or between the Big Data Application Provider and the Big Data Framework Provider RA blocks.

Identity Management and Authorization: Individual vertical Big Data Application Providers may implement their own schemes for usage of their services. Identity management enables Big Data Application 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 Big Data Application Provider can support.

Code execution services: A Big Data Application 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 (e.g., execution time) on the code that is provided to the service, and how the results will be delivered to the Data Consumer.

Charging: Big Data Application 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.

## Big Data Framework Provider

Big Data Framework 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 state-of-art computing features from software, platforms, and infrastructures based on application needs.

**[Editor’s Note: The activities below need to be synchronizes with the Def&Tax document.]**

Big data applications need to rely on various platforms and technologies to meet their challenges of scalable data analytics and operation. The specific technological approaches 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 IT 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 Big Data Application Provider. The nature and granularity of these services may vary, but generally will 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 IT 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 Big Data Framework 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.

Automation Services: A big data system will have many ‘moving parts’ that must be operated dynamically. Automation is a fundamental principle in building big data systems, and hence automation services are an integral component of the provided services. Automation services 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 Big Data Application Providers to push analytics to be performed by Big Data Framework 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 Big Data Application Provider. Data Consumer uses the interfaces (or services) exposed by Big Data Application Provider to get access to the information of its interest. These services can include data reporting, data retrieval, and data rendering.

Data Consumer activities can include:

* Data search, query, retrieval
* Exploring data using data visualization software
* Creating reports and organized drill-down using business intelligence software
* Ingesting data into their own system
* Putting data to work for the business, for example to convert knowledge produced by the big data applications into business rule transformation
* Conversion of data into additional data-derived products

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 applications’ activities 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 the interfaces 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 Data Providers.
* Overseeing evaluation of data available from Data Providers.
* Defining requirements for the collection, preparation, analysis of data, etc.
* Establishing System architecture requirements
* Auditing data applications’ activities for compliance with requirements.
* Defining data dictionaries and data storage models

# Management

## System Management

System Management is responsible for managing Big Data infrastructure including all physical and virtual resources. A solid and efficient management platform is required to effectively store, manage and protect Big Data resources.

## Lifecycle Management

Lifecycle Management is responsible for 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 Big Data Framework Provider, Big Data Application 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 may need 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 Framework 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 Big Data Big Data Application Provider using the resources in Big Data Framework 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;
* Big Data Application Provider to perform data masking and format transformations for preservation purpose;
* Big Data Framework 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

Security and Privacy considerations form a fundamental aspect of the Big Data Reference Architecture. This is geometrically depicted by having a Security and Privacy fabric around the reference architecture components, since it touches all of the components. At a minimum, a big data reference architecture will provide and ensure verifiable compliance with both GRC (Governance, Risk & Compliance) and CIA (Confidentiality, Integrity & Availability) regulations, standards and best practices.

Please see the NIST Big Data Security Reference Architecture document for additional guidance and best practices regarding these critical and sensitive areas of individual privacy and corporate security.

This way the role of S&P is depicted in the right relation to the components and at the same time does not explode into finer details, which may be more accurate but are best relegated to a more detailed Security Reference Architecture.

In addition to the Application and Framework Providers, we also decided to include the Data Provider and Data Consumer into the fabric since at the least they have to agree on the security protocols and mechanisms in place.

# Big Data Taxonomy

**[Editor’s Note: 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.

Big Data Application 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.

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.

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

**Second Level Terms:**

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

[Editor’s Note: TBD]

# 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>

# Appendix D: Deployment Considerations

# Big Data Framework Provider



Figure 2: Big Data Framework Deployment Options

The Big Data Framework Provider implements various functionalities in order to support big data applications. These include infrastructures (e.g., VM clusters, storage, networking), platforms (e.g., programming languages and runtimes, databases), and applications (e.g., analytic tools).

Traditional Data Frameworks have been implemented as dedicated, custom-made on-premise systems. Recently, Big Data analytics and processing has been flourishing thanks to the adoption and use of cloud computing. Figure 2 illustrates these two options.

## Traditional On-Premise Frameworks

Traditional frameworks needed to support data analytics require custom-designed computing resources in order to satisfy the heavy demand on computing, storage and database systems. Such on-premise systems usually use large, high performing custom-design databases designed to address unique demands of the data analytics workloads.

Such traditional solutions offer software, platform and cluster frameworks that can be used separately, or in conjunction with each other. The frameworks could be implemented in a layered fashion, or could be built vertically. Regardless, the implementations of such frameworks will require physical resources such as CPUs, storage devices, networking equipment, cooling and electrical facilities, etc.

## 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 facilitate the realization of big data implementations.

In Cloud Computing systems 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 in turn 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.)

The **cloud Capabilities Types** inside the cloud service component are significant. The cloud service component offers its services via a service interface whose capabilities is in turn graphically represented by the inverted L diagrams. Each L represents the service interface(s) associated with the respective **Cloud Capability Type** that is implemented by the **Cloud Service Category** (e.g. IaaS, PaaS and SaaS)**.** The relative positioning of the L diagrams are also significant. The implication of the L shapes is that application capabilities can be implemented using platform capabilities or not (at the choosing of the cloud service provider) and that platform capabilities can be implemented using infrastructure capabilities or not.

Cloud services can be described in terms of the cloud capabilities which they offer, based on the resources provided by the cloud service. There are three cloud capabilities types:

• Application capabilities

• Platform capabilities

• Infrastructure capabilities

Cloud services are also grouped into categories, where each category of services has some qualities that are common between the services. The services in these categories may include capabilities from one or more than one of the capabilities types above. Some common cloud service categories include:

• Infrastructure as a Services (IaaS)

• Platform as a Service (PaaS)

• Software as a Service (SaaS)

• Network as a Service (NaaS)

### Resource Abstraction & Control

The Resource Abstraction & Control component is used by cloud service providers to provide access to the physical computing resources through software abstraction. Resource abstraction needs to ensure 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 & 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 & Control component can include software elements such as hypervisors, virtual machines, virtual data storage, and time-sharing.

The Resource Abstraction & 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.

### Physical Resources

This layer represents physical resources needed for supporting the cloud computing system. Example of such physical resources are CPUs, storage devices, networking equipment, cooling and electrical facilities, etc.



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)