**NIST Big Data**

**Taxonomies**

**Version 0.3**

**Definitions & Taxonomies Subgroup**

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

**March, 2014**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Changes abstract | References | Editor |
| 0.1 | 2/4/14 | Extraction from Definitions Document | M0142v6 Def and Tax | Nancy |
| 0.2 | 3/4/14 | Completion from mindmap | M0226v10 Ref Arch | Nancy |
| 0.3 | 3/16/14 | Guidance on what activities and components have changed because of big data | With additions by John Rogers | Nancy |

Executive Summary 4

1 Introduction 4

1.1 Objectives 4

1.2 How This Report Was Produced 5

1.3 Structure of This Report 5

2 Actors and Roles 5

3 Data Provider 7

3.1 Data Capture from Sources 8

3.2 Data Persistence 9

3.3 Data Scrubbing 9

3.4 Data Annotation / Metadata creation 9

3.5 Access Rights Management 9

3.6 Access Policy Contracts 10

3.7 Data Distribution APIs 10

3.8 Capabilities Hosting 10

3.9 Data Availability Publication 10

4 Data Consumer 10

5 System Orchestrator 12

5.1 Business Ownership Requirements and Monitoring 12

5.2 Governance Requirements and Monitoring 13

5.3 Data Science Requirements and Monitoring 13

5.4 System Architecture Requirements and Monitoring 14

6 Big Data Application Provider 14

6.1 Data Collection Processes 14

6.2 Data Preparation Processes 15

6.3 Data Analytics Processes 15

6.4 Visualization 16

6.5 Access 16

7 Big Data Framework Provider 17

7.1 Infrastructures 17

7.2 Platforms 18

7.3 Processing Frameworks 19

8 Security and Privacy 20

9 Management 20

# Executive Summary

The Big Data Paradigm shift has changed the architecture of big data and analytics systems. This shift has introduced many new tools and techniques to obtain scaling in big data systems. In addition, what used to be one-server end-to-end data systems has now been distributed across a number of resources, and even across a number of organizations. To facilitate better communication and understanding across the participants in this field, this taxonomy document expands the functional components of the reference architecture. The top-level roles of the taxonomy are Data Provider, System Orchestrator, Data Consumer, Big Data Application Provider Big Data Framework Provider, Security and Privacy, and Management. The activities within each of these roles will be specified to see where the functional components reside. Then components and subcomponents will be listed within each activity. This taxonomy is not meant to be an exhaustive list, but only necessary to describe what is new in big data systems. In some cases this requires also listing current practices and technologies in the same category. This taxonomy is a work in progress, and is expected to evolve through the continuing efforts of the NIST Big Data Working Group as the use cases and requirements are analyzed against the reference architecture.

# Introduction

## Objectives

The Definitions and Taxonomy subgroup focused on identifying the concepts involved in big data, and defining terms in both the concepts needed to describe this new paradigm, and to define the terms used in the reference architecture. This taxonomy provides a hierarchy of the components of the reference architecture.

For *managers* the terms will distinguish the categorization of techniques needed to understand this changing field

For *procurement officers* this will provide the framework for discussing organizational needs, and distinguishing among offered approaches

For *marketers* this document will provide the means to promote the characteristics of solutions and innovations

For the *technical* community it will provide a common language to better differentiate the specific offerings

## How This Report Was Produced

The document derives from the discussions in the Definitions and Taxonomy Subgroup of the NIST Big Data Working Group. This subgroup produced two reports, the Definitions document that provides terms and definitions for the important new concepts in big data, and this document that provides a taxonomy of technologies that comprise big data systems. This taxonomy was developed using a mindmap representation, which provided a mechanism for multiple inputs and easy editing.

It is difficult to describe the new components of in big data systems without fully describing the context in which they reside. In the subgroup we attempted to describe only what has changed in the shift to the new big data paradigm, and only as many of the other components as are needed to clarify the new technologies. There is for example no attempt to expand out analytics techniques, as these pre-date “Big Data”. This taxonomy needs to remain a work in progress to mature as better categorizations are developed for detail in the reference architecture to align with the different use cases.

## Structure of This Report

This document provides a taxonomy for the reference architecture, providing the terminology and definitions for the components of technical systems that implement these technologies. The taxonomy will first describe the actors, or the entities that fulfill the different roles in the reference architecture. Then the roles will then be described in turn, with the activities that are performed. The components and subcomponents within a given activity are listed underneath that activity in a hierarchical fashion.

The architectural components are more fully described in the NIST Big Data Reference Architecture and the NIST Big Data Security and Privacy documents. Comparing the related sections in these two documents will give the reader a more complete picture of the consensus of the working groups.

For descriptions of where big data is going and how to get started to make use of these technologies, the reader is referred to the NIST Big Data Roadmap. Finally, to understand how these systems are architected to meet users’ needs, the reader is referred to the NIST Big Data Use Cases and Requirements document, to be viewed along with section 4

# Actors and Roles

Actors and roles have the same relationship as in the movies; except in system development the Actors can represent individuals, organizations, software or hardware. Each element in the taxonomy can potentially be executed by a different actor. Examples of actors include:

* Sensors
* Applications
* Software agents
* Individuals
* Organizations
* Hardware Resources
* Service abstractions

While in the past, data systems tended to be hosted, developed, and deployed with single resources of only one organization. Now roles may be distributed, in analogy to the way cloud has spurred on a diversity of actors within a given solution. We are mindful that in big data systems actors can likewise come from multiple organizations.

The roles are the parts the actors play in the overall system. One actor can perform multiple roles. A role can potentially have multiple actors, in the sense that a team of independents entities, perhaps from independent organizations, may be used to satisfy the end-to-end system requirements.

The functional components of the Reference Architecture are given in Figure 1.



Figure : Big Data Reference Architecture

The architecture provides the roles that make up the disparate functionality of the overall end-to-end system. These roles consist of the

* Data Provider
* Data Consumer
* System Orchestrator
* Big Data Application Provider
* Big Data Framework Provider
* Security and Privacy
* System Management

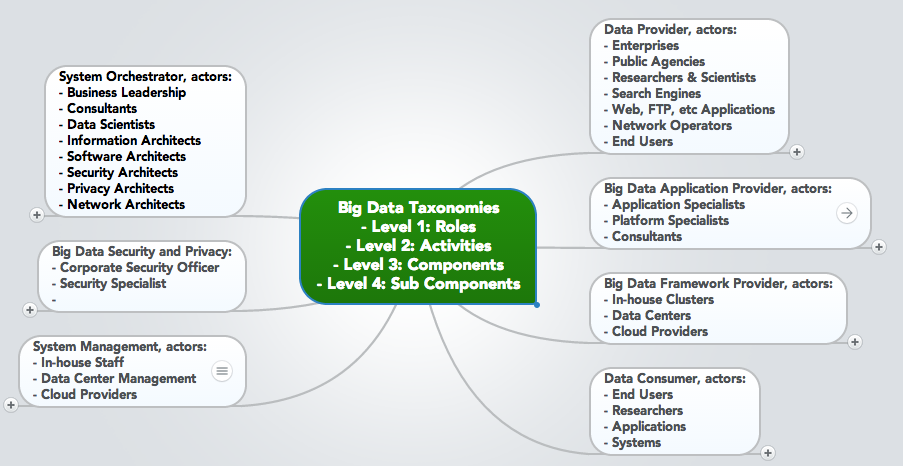


Figure: Roles and a sampling of actors

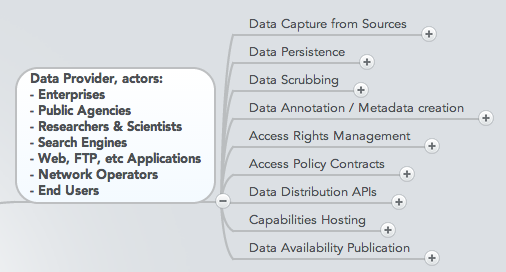
These roles will each be explored in turn, with their activities.

# Data Provider

A **Data Provider** makes data available to themselves or to others. The actor fulfilling this role can be part of the big data system, internal to the organization in another system, or external to the organization orchestrating the system. Once the data is within the local system, requests to retrieve the needed data will be made by the Big Data Application Provider and routed to the Big Data Framework Provider.

Actors can include:

* Enterprises
* Public Agencies
* Researchers & Scientists
* Search Engines
* Web, FTP, etc Applications
* Network Operators
* End Users
* Smartphones



While the concept of a Data Provider is not new, the greater data collection and analytics capability have opened up a number of new possibilities for providing valuable data. The Open Data Initiative of the U.S. Federal government pushes for agencies of the federal government as stewards of public data to also serve the role of Data Provider.

## Data Capture from Sources

The Data Provider captures the data from their own sources or others. This activity could be described as the capture from a data producer, whether it is a sensor or an organizational process. Examples of data sources include online sources:

* Web Browsers
* Sensors
* Deep Packet Inspection Devices (e.g.Bridge, Router, Border Controller)
* Mobile devices

and offline sources:

* Public Records
* Internal Records

While perhaps not different theoretically from what has been in use before, this is an area that is exploding in the new Big Data Paradigm. Devices are being instrumented as sensors providing not only a number of sources of data, but data in large quantities. Examples of new devices as sensors include smartphones, personal wearable devices such as exercise monitors, and household electric meters. In addition, technologies such as RFID are sources of data for the location of shipped items. Collectively all the data producing sensors are known as the “Internet of Things”. The subset of personal information devices are often referred to as “Wearable Tech”, with the resulting data sometimes referred to as “Digital Exhaust”.

## Data Persistence

The Data Provider persists the data into a repository from which the data can be extracted to make available to others. The data can be persisted in

* Internal hosting
* External hosting
* Cloud hosting (a different hosting model whether internal or external)

And would be subject to a

* Data Retention Policy

Hosting models have expanded through the use of cloud computing. In addition often the data persistence is accessed through mechanisms such as web services that hide the specifics of the underlying storage. Data-as-a-Service is a term used for this kind of data persistence that is accessed through specific interfaces.

## Data Scrubbing

Some datasets contain data elements naturally collected as a part of the data production process that contain sensitive information. Whether for regulatory compliance or sensitivity, such data elements may be altered or removed. As one example for Personally Identifiable Information (PII), the Provider can

* Remove Personally Identifiable Information
* Randomization (for implicit PII)

The latter obscures the PII to eliminate any possibility of tracing the data back to an individual. In the era of big data, this is an area that requires greater diligence. While individual sources may not contain PII, when combined with other data sources there is the risk that individuals can now be identified from the integrated data.

## Data Annotation / Metadata creation

The Data Provider would maintain information about the data in their repository, as well as maintaining the data itself. The metadata, or data annotation, would provide information of the provenance of the data, or the history of the data, in sufficient detail to enable anyone using the data to understand how to properly use and interpret the data. The metadata can be encoded in

* Ontology – a semantic description of the elements of the data
* Within a data file – in any number of formats

With the push for open data, it has become even more critical that information *about* the data be encoded to clarify the data’s origins. While the actors that collected the data will have a clear understanding of the processes, re-purposing data for other uses is open to misinterpretations when other actors use the data at a later date.

## Access Rights Management

The Data Provider will determine the different mechanisms that will be used to define the rights of access, which can be specified separately by:

* Data Sources – the collection of datasets from a specific source
* Data producer – the collection of datasets from a given producer
* Personal Identifier (PII) access rights - as an example of restrictions on data elements

## Access Policy Contracts

The Data Provider will define contracts by which others will be allowed access to the data, or will be allowed to retrieve the data. These contracts specify:

* Policy for Primary and Secondary Rights
* Agreements

## Data Distribution APIs

Technical protocols are defined for different types of data access, which can include:

* File Transfer Protocol (FTP) or Streaming
* Compression techniques (single compressed file, Split compressed file)
* Authentication
* Authorization

## Capabilities Hosting

In addition to offering data downloads, the Data Provider could

* Provide query access without transferring the data
* Allow analytic tools to be sent to operate on the data sets

For large volumes of data it become impractical to move the data to another location for processing. This is often described as moving the processing to the data, rather than the data to the processing.

## Data Availability Publication

The Data Provider makes available the information needed to know what data or data services they offer. Such publication can consist of:

* Web description
* Services Catalog
* Data dictionaries
* Advertising

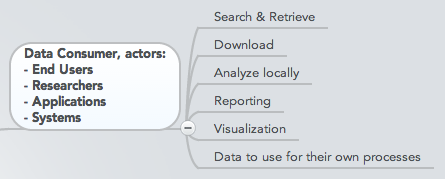
In addition there are now a number of third-party locations that publish available datasets, such as data.gov for the U.S. Federal Government.

# Data Consumer

The Data Consumerreceives the value output of the big data system. In many respects they are the recipients of the same functionality that the Data Provider brings to the Big Data Application Provider. The Application Provider then offers that same functionality to the Data Consumer following the value that the system adds to the original data sources.

Actors fulfilling the role of Data Consumer can consist of:

* End Users
* Researchers
* Applications – Systems



The Data Consumer can pursue the following sample activities:

* Search & Retrieve
* Download
* Analyze locally
* Reporting
* Visualization
* Data to use for their own processes

## Current Application as Data Provider

These activities are explicit to the Data Consumer role within a data system. If the consumer is in fact a follow-on application, then the Data Consumer would look to the Application Provider for the activities of any other Data Provider. The follow-on application’s System Orchestrator would negotiate with this application’s System Orchestrator for the types of data wanted, access rights, etc. The Big Data Application Provider would thus need to serve as the Data Provider, from the perspective of the follow-on application.

# System Orchestrator

The **System Orchestrator** provides the overarching requirements which the implementation of the system must fulfill, including policy, architecture, resources, business requirements, etc., as well as the monitoring or auditing activities to ensure the compliance of the system with respect to those requirements.

System Orchestrator, actors:

* Business Leadership
* Consultants
* Data Scientists
* Information Architects
* Software Architects
* Security Architects
* Privacy Architects
* Network Architects



This role provides the system requirements, high-level design, and monitoring for the data system. While the roles pre-date big data systems, there are design activities that have changed within the big data paradigm.

## Business Ownership Requirements and Monitoring

As the business owner of the system, the System Orchestrator oversees the business context within which the system operates, including specifying the:

* Business Goals
* Targeted Business Action
* Data Provider contracts and SLAs
* Data Consumer contracts and SLAs
* Capabilities Provider Negotiation
* Make/Buy Cost Analysis

A number of new business models have been created for Big Data systems, including data-as-a-service, where a business contracts to provide a process in the Big Data Application Provider role as a service to other actors. In this case the business model is to process data received from a data provider and provide the transformed data to the contracted data consumer.

## Governance Requirements and Monitoring

The System Orchestrator establishes all policies and regulations to be followed throughout the data lifecycle.

* Policy compliance requirements and monitoring
* Change management process definition
* Data Stewardship and Ownership
* Requirements for change management processes

Big Data systems potentially interact with processes and data being provided by other organizations, requiring more detailed governance and monitoring between the components of the overall system.

## Data Science Requirements and Monitoring

The System Orchestrator establishes the detailed requirements for the functional performance of the analytics for the end-to-end system, translating the business goal into data and analytics design, including:

* Data Source Selection
  + Data Description
  + Data Location
  + File Types
  + File Attributes
  + Data provenance evaluation
* Data Collection Requirements and Monitoring
* Data Preparation Requirements and Monitoring
* Data Analysis Requirements and Monitoring
* Analytical Model Choice
* Data Visualization Requirements and Monitoring
* Application Type Specification
  + ex. Streaming
  + ex. Aggregation
  + ex. Integration
  + ex. Transfer
  + ex. Search
  + ex. Statistics
  + ex. RT Analytics
  + ex. Batch Analytics
  + ex. Interactive Annotation
  + ex. Others

A number of the design activities have changed in the new paradigm. In particular there are more choices of data models to be considered than just the relational model. The choice of which non-relational model can depend on the choice of data analysis needs, and the best choice for the data element to use for the splitting of the data across the storage nodes can sometimes only be determined operationally based on the quantity of data.

## System Architecture Requirements and Monitoring

The System Orchestrator establishes the detailed architectural requirements for the data system.

* Data Process requirements
* Software component determination
* Hardware component determination
* Logical Data Modeling and Partitioning
* Data export requirements
* Scaling Requirements

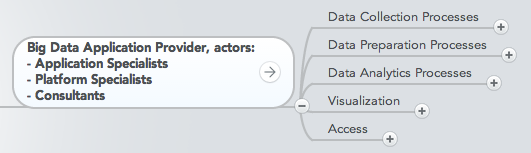
The system architecture has changed in the big data paradigm due to the interplay of a number of independent components, potentially provided by different actors. In addition there are a number of additional communications and inter-connectivity requirements among the components. Maintaining the needed performance can lead to a very different architecture from what would have been used prior to the new distribution of data across system nodes.

# Big Data Application Provider

The Big Data Application Provider executes the manipulations of the data lifecycle to meet the requirements established by the System Orchestrator. This is where the general capabilities within the Big Data Framework are combined to produce the specific data system.

Actors:

* Application Specialists
* Platform Specialists
* Consultants



## Data Collection Processes

The Application Provider must establish the mechanisms to capture data from the Data Provider.

* Protocol (link)
* Network
* Secure transport

While these transport mechanisms predate big data, the resources to handle the large volumes or velocities do result in changes in the way the processes are resourced.

## Data Preparation Processes

Whether after the storage of raw data, or before the storage of the data as organized information, a number of processes are used in the data preparation activity, analogous to all that is done now for data systems. New Big Data preparation processes include partitioning the data, and new processes for aggregation or summarization.

* Data Cleansing
* Data Conversion
* Calculated field creation
* Data summarization
* Data partition implementation
* Data Storage preparation
* Data Virtualization Layer
* Schema-on-read

Just as data collection may require a number of resources to handle the load, likewise the preparation may also require new resources or new techniques. For large data volumes, data collection is often followed by storage of the data in its raw form. This pushes the data preparation processes until after the storage, and to be handled by the application code. This is what is meant by the expression “Schema-on-Read”.

## Data Analytics Processes

* Metadata Read Processes
* Human-in-the-loop rapid analytics
  + discovery
  + hypothesis
  + hypothesis testing
  + alternate methods
* Machine Learning implementation
  + Correlation
  + Classification
  + Data Fusion
  + Data Integration
  + Data Mining
  + Artificial Intelligence
  + Pattern Recognition
  + Predictive Modeling
  + Regression
  + Cluster Analysis
  + Spatial Analysis
  + Time Series Analysis
  + Audio Analysis
  + Visual Analysis
  + Textual Analysis (e.g. Structured, Unstructured)

The term Data Science is used in many ways. While this term refers to the end-to-end data lifecycle, the most common usage focuses on the discovery, or rapid hypothesis-test cycle for finding value in big volume datasets. This rapid analytics cycle is also described as “Agile Analytics”, which starts with quick correlation or trending analysis, with greater effort spent on hypothesis that appear most promising.

The analytics processes for structured and unstructured data have been maturing for many years. There is now greater emphasis on the analytics of unstructured data because of the greater quantities now available. The knowledge that valuable information resides in unstructured data promotes a greater attention to the analysis of this type of data.

While the analytics methods have not radically changed with Big Data, their implementation has changed to accommodate for example the techniques that break the overall analytics into tasks to scatter to the independent data nodes, and the subsequent gathering of the results to achieve the final full-dataset analysis.

## Visualization

There are multiple types of visualization, and usages of visualization.

* Exploratory Data Visualization
  + Multidimension (ex. 2D/3D)
  + Multiresolution
  + Interaction
  + Animation
  + Simulation
  + Statistical Graphics
  + Surface Rendering
  + Volume Rendering
* Knowledge or Explanatory Visualization
  + Reports
  + Business Intelligence
  + Customer summarization presentation

Data Science relies on the full dataset type of discovery or exploration visualization, from which the Data Scientist would form a hypothesis. While clearly predating Big Data, there is a greater emphasis on this type of exploratory visualization to begin to understand large volumes of re-purposed data.

Explanatory visualization is the creation of a simplified, digestible visual representation of the results, suitable for assisting a decision or communicating the knowledge gained. Again, while this technique has long been in use, there is greater emphasis now to “tell the story”. Given the large volumes and varieties of data, and the data’s potentially complex relationships, the communication of the results of the analytics to a non-analyst audience requires careful visual representation to communicate the results in a way that can be easily consumed.

## Access

The Application Provider gives the Data Consumer access to the results of the data system.

* Data export API Processes
  + API
  + Protocol
  + Query Language
* Data Charging Mechanisms
* Consumer Analytics Hosting
* Analytics-as-a-service hosting

This activity should in fact mirror everything in the Data Provider, since the Data Consumer may view this system as the Data Provider for their follow-on tasks.

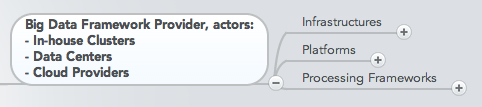
Many of these tasks have changed, as algorithms have been re-written to accommodate and be optimized for the horizontally distributed resources.

# Big Data Framework Provider

The Big Data Framework Provider has general resources or services to be used by the Application Provider in the creation of the specific application. There are many new components here from which the Application Provider can choose in utilizing these resources and the network to build the specific system.

Actors:

* In-house Clusters
* Data Centers
* Cloud Providers
* Storage Vendors
* Software Vendors



## Infrastructures

* Cluster Implementation
  + Operating System
    - Linux
    - Machintosh
    - Windows
    - Etc.
  + Physical Resource Implementation
    - Server
    - Cluster
    - Multi-site Cluster
  + Logical Distribution
    - Peer-to-Peer
    - Master-Slave
    - Multiple masters with slave nodes
  + Virtual Resource Implementation
  + System Administration
* Storage Implementation
* Local Disks
* HW/SW RAID
* SAN
* NAS
* Distributed Filesystems
* Distributed Object Stores
* Storage Administration
* Networking Implementation
* Physical Network
* Protocol
* Fault Tolerance
* Security/Encryption
* Virtual Network
  + Protocol
  + Fault Tolerance
  + Security/Encryption
* Network Administration

The biggest change under the Big Data Paradigm is of course the cooperation of horizontally-scaled independent resources to achieve the desired performance.

## Platforms

* Logical Storage
  + File Systems
    - Delimited
    - Fixed Length
    - Binary
  + Indexed
    - Relational
    - Key-Value
    - Tables
      * Row-oriented
      * Column-oriented
    - Document Store
      * JSON
      * BSON
      * XML
    - Graph
      * Property Graph
        + Non-Native (global index)
        + Native (index-free-adjacency)
      * Hyper-graph
      * Triple-Stores
    - Queues
* Data Query Implementation
  + SQL
    - JDBC
    - SQL/CLI (ODBC)
  + API
  + Distributed and Parallel
    - Standard
    - Parallel
    - Cluster-based
    - Distributed
    - Local Federated
  + Custom
* Data Domains
  + Data Registry
  + Semantic Representation
  + Ontology
* Data Surety
  + Available, Consistent, Isolated, Durable (ACID)
  + Basically Available, Soft State, Eventually Consistent (BASE)
    - Sesstion Affinity
    - Logical Consistency
    - Eventual Consistency
    - Replication Durability
    - CAP tradeoffs
    - Write Quorum
  + Business Transactions
    - Compare and Set (CAS)
    - Version Stamp
      * Counter
      * Random GUID
      * Content Hash
      * Time Stamp
    - Vector Stamp
      * Version Clock
      * Version Vector
* Database Administration

The logical storage paradigm has expanded beyond the “flat file” and relational model paradigms to develop new non-relational models. This has implications for the concurrency of the data across nodes within the non-relational model. Transaction support in his context refers to the completion of an entire data update sequence and the maintenance of eventually consistency across data nodes. This is an area that needs more exploration and categorization.

## Processing Frameworks

* Data Type Processing Services
  + Textual
  + Documents
  + Spatial
  + Images
  + Aerial
  + Faces
  + Fingerprints
  + Astronomical
  + Etc.
  + Video
  + Etc.
* Schema Information (metadata)
  + Queryable
  + Schema on demand
  + Pre-knowledge
* Application Framework
  + automation services
  + test services
  + analytics hosting services
  + data quality servce
  + workflow services
* Batch Processing Frameworks
  + MapReduce
  + Linear Algebra
  + Sparse
  + Dense
  + Spectral methods
  + N-Body methods
  + Grids
  + Structured
  + Unstructured
  + Combinational Logic
  + Graph Traversal
  + Backtrack and Branch-and-Bound
  + Dynamic Programming
  + Graphical Models
  + Finite State Machines
* Stream Processing Frameworks
  + Complex Event Processing
* Resource Managment Frameworks
* Processing Framework Administration

Both Transformation and Capability Provider activities have changed because of big data. Now the interchange between these two roles operates over a set of independent yet coupled resources. It is in this interchange that the new methods for data distribution over a cluster have developed. Just as simulations went through a process of parallelization (or horizontal scaling) to harness massive numbers of independent process to coordinate them to a single analysis, now Big Data Services perform the orchestration of data processes over horizontally scaled resources.

# Security and Privacy

Security and privacy are important activities in both the System Orchestrator for policy, requirements, and auditing; but are also present in the development, deployment and operation both for the Application Provider and the Framework Provider. These ubiquitous activities are described in the NIST Big Data Working Group Security and Privacy document.

Actors:

* Corporate Security Officer
* Security Specialist



## Security and Privacy Policy Requirements

* International
* National
* Enterprise

## Security and Privacy Monitoring

### Security Protection Requirements and Monitoring

* Security against Denial of Service (Cryptographic Protocols proactively resistant to DoS)
* Big Data for Security
  + Analytics for Security Intelligence
  + Data-driven Abuse Detection
  + Event Detection
  + Forensics
* Security negotiation with actors
* Security Compliance Analysis

### Data Provenance Monitoring

* End-point Input Validation
* Communication Integrity
* Authenticated Computations
  + Trusted Platforms
  + Cryptographically Enforced
* Granular Audits
* Control of Valuable Assets
  + Lifecycle Management
  + Retention, Disposition, Hold
  + Digital Rights Management

### Data Privacy Requirements and Compliance Monitoring

* Communication Privacy
* Confidentiality
  + Access Policies
  + Systems Enforced
  + Cryptographically Enforced
* Computing on Encrypted Data
  + Searching and Reporting
  + Outsourcing of Computation
* Secure Data Aggregation
* Key Management
* Implicit PII determination

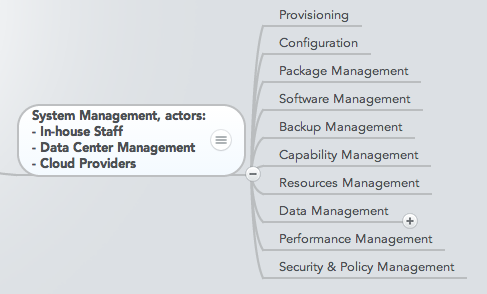
[Needs notes on what is new because of the new Big Data Paradigm vs security as we’ve always done it]

# Management

In the Reference Architecture, Management is the overarching control of the execution of the system, its deployment of the system and its operational maintenance.

Actors:

* In-house Staff
* Data Center Mangement
* Cloud Providers



Activities include:

* Provisioning
* Configuration
* Package Management
* Software Management
* Backup Management
* Capability Management
* Resources Management
* Data Management
  + Big Data Lifecycle Management
  + Administration
* Performance Management
* Security & Policy Management

[Needs a better description of what is new or different in managing big data systems]

# Future Directions

This taxonomy must be considered a work in progress. As the NIST Big Data Working Group explores the mapping of use cases to the reference architecture, new components will be identified, or perhaps the components will be better organized into a different hierarchy.

While this taxonomy seeks to identify the components, there is no exploration here for the interactions between the components. For example, how does the specific instance of an analytics process in the Application Provider interact with a distributed data framework? How does a specific non-relational database load or process data that has been distributed across a storage cluster.

The interactions between components would be two-fold, first a description of instructions, and second of the return of data or acknowledgement from an instruction. Descriptions of the interplay between components would perhaps be better described in terms of a link-node structure, with the links representing these types of interactions.

In future activities of the NIST Big Data Working group, we look forward to additional clarification not only in the components of this taxonomy, but of the interactions between components.