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

**NBD-WD-2013/M0126, v4**

**Source: Reference Architecture Subgroup>**

**Status: Proposal**

**Title: RA Proposal based from Earlier Proposals, Use Cases’ Requirements and Feedback**

**Author: Members from Requirements and RA Subgroups**

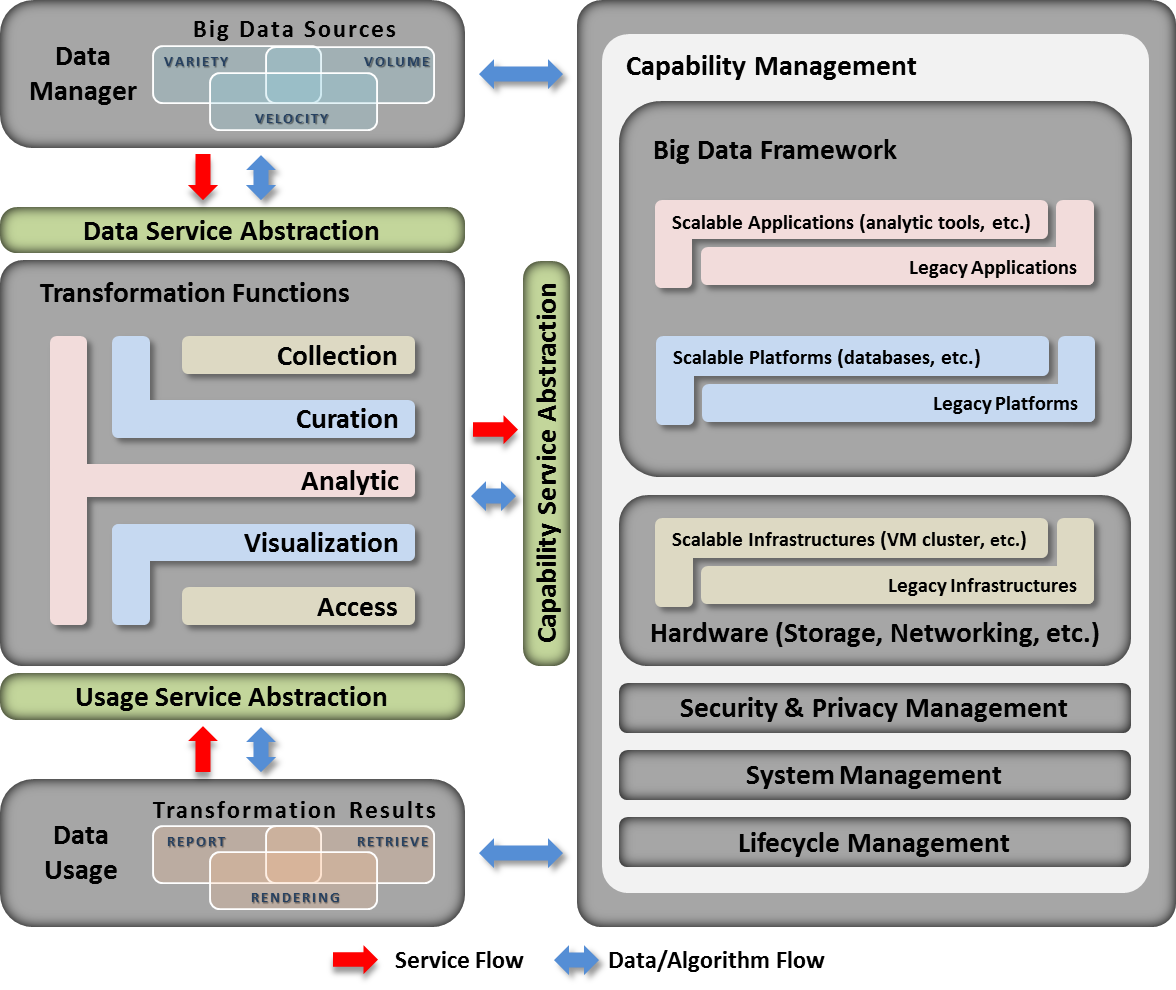
***Abstract***

This document combined the ideas from proposals submitted to the Reference Architecture Subgroup, extracted key requirements from the Requirements Subgroup, and other personal feedback to present a combined view of reference architecture for review and discussion.

|  |  |
| --- | --- |
| **Revision, Date** | **Description** |
| V4, 09-04-13 | Changed service flow direction from Data Usage to Transformation; changed “Data/Signal” to “Data/Algorithm) |
| V3, 08-28-13 | Further refine the RA diagram with better descriptions on the four key components and the interfaces in between. |
| V2, 08-21-13 | Combine the combine the “Legacy Framework” and “Cloud C. Framework” into Big Data Framework which should cover both the traditional approaches as well as the future approaches to handle big data issues. |
| V1, 08-15-13 | As stated in the Abstract |

1. **Background**

Based from the proposals submitted to the NBD Reference Architecture Subgroup such as the M0059 (DataFlow#1, DataFlow#2, Stack/Layer), M0055 (Big Data Architecture Framework) and the requirements extraction from the NBD Requirements Subgroup on M0096 along with constructive feedback from Felix Njeh (U.S. Army Information Technology Agency), Vivek Navale (NARA), Quyen Nguyen (NARA), and Gary Mazzaferro (Consultant), Figure-1 shows a revised RA diagram to further review and discussion.



**Figure-1: Combined Contribution Reference Architecture**

Key points are to meet the use cases’ requirements:

* Be able to support diversified data sources with variety, volume, and velocity of data
* Be able to support various transformation functions such as collect, curate, analysis, visual, and access for a given dataset.
* Be able to support mixed of legacy/traditional and cloud computing framework with the additional management of hardware, security and Privacy, systems, and lifecycle management.

**Roles and Responsibilities of RA Components**

**Data Manager (DM)** [machine or person] is responsible for processing given data using certain methodologies and placing the processed results into certain formats. Their main responsibility is to provide data source description (location of data which can be centralized or distributed, data at rest or data in motion, file types and attributes, etc.) to the TF and request one or more data services for the given data. Data services can include (a) collecting data from one or more data sources, (b) deciding what curation process should be performed, (c) determining how data should be analyzed, (d) choosing what methods and formats the processed results should be stored as, and (e) deciding how and where processed results should be pushed to (end users or systems, if any).

**Transformation Functions (TF)** is responsible for executing service requests from DM and DU. DM data service responsibilities include the execution of transformation functions such as collection, curation, and analytics, using CM as the computing fabric infrastructure. The processed results can then be stored in certain reporting formats and made available for access. DU usage service responsibilities include search and retrieval of processed data for reporting and visualization rendering.

**Capabilities Management (CM)** is responsible for providing a computing fabric such as system hardware, network, storage, virtualization, computing platform, and security in order to execute certain applications while protecting the privacy and integrity of data. The computing fabric infrastructure facilitates a mix-and-match of traditional and future computing features from softwares, platforms, and infrastructures based on application needs.

**Data Usage (DU)** is responsible for assisting end users and systems in formulating search queries and expected outputs for the TF, and to request one or more usage services from the available processed results.

**Communication between components**

Two types of communication will occur between each component:

1. Service Flow
   1. DM--> TF
      1. DM may request one or more services for TF to perform
   2. TF --> CM
      1. TF may request one or more computing capabilities from CM
   3. TF <--> DU
      1. TF may push processed results to end users or systems at the request of DM
      2. DU, on behalf of end users or systems, may request processed results from TF
2. Data Flow
   1. DM <--> TF
      1. DM may continue updating data source descriptions to TF
      2. TF may send signals to DM for additional info and send processing status to DM
   2. TF <--> CM
      1. TF may send capabilities request either sequentially or in parallel to CM
      2. CM may send processing status to TF
   3. CM <--> DM
      1. CM may send initiation signals such as collect/transfer data to DM
      2. DM may respond to signals by pumping data directly to CM after CM’s initiation
   4. TF <--> DU
      1. TF may send processed results and processing status to DU
      2. DU may send continuous requested signals to TF
   5. CM <--> DU
      1. CM may send initiation signals such as visualization data to DU
      2. DU may send signals such as interactive inputs to CM

**Location of components**

Each component may either reside at different systems separate from each other, be a mix of components at one system, or multiple systems for a single component, or all components may be stationed in a single system. It is all depends on application needs and the computing environment.

**Service Abstractions (SAs)**

NBD use cases submission listed multi-stages complex data and analytic processing needs on data across many diversified vertical applications. In order to support such a wide spectrum of processing needs while fulfilling the NBD-PWG scope of vendor neutral technology agnostic, the BD RA introduced the concept of Service Abstraction (SA) in order to aggregate such multi-stage processing between top-level components.

An SA defines the interaction between certain key components as listed above. The interaction forms the relationship as *Requestor* and *Provider* between key given components. The SA contains several factors:

* A set of services such as collection, curation, etc. that the Requestor would like to get Provider to perform
* A complete, specific set of instructions (steps (e.g., how to initiate data transfer; which methods to perform curation; how and where to store the processed results, etc.) to carry out specific services) for each requested service
* As needed, detailed signaling operations such as handshake between Requestor and Provider per each instruction
* As needed: detailed data flow for each operation
* A set of metrics to determine whether the Provider performed the requested services

There are three types of service abstractions:

***Data Service Abstraction*** – enables DM to subscribe to one or more data services at the TF so that data can be collected, curated, analyzed, visualized, and accessed. The processed results can be used by the DM and DU in real time or at a later time.

***Capability Service Abstraction*** – enables TM to subscribe to one or more capability services at the CM so that given datasets and algorithms can be securely transferred, stored, and executed by the specific computing fabrics using data storage, networking, and computing cluster.

***Usage Service Abstraction*** – enables DU to subscribe to one or more protected usage services from the TF so that the processed results can be retrieved, reported, and rendered for visualization. The DU can be the same users from DM or other consumers or systems.

**Appendix A – High-level Requirements Extraction from M0125**

**Data Source Requirements:**

DSR-1: needs to support reliable real time, streaming, and batch processing to collect data from centralized and distributed data sources, sensors, or instruments. (6: 1, 3, 4, 5, 7, 8)

DSR-2: needs to support slow and high throughput data transmission between data sources and computing clusters. (2: 3,5)

DSR-3: needs to support diversified data content ranging from text to multimedia to instrumental data. (5: 3, 4, 5, 6, 7)

**Transformation Requirements:**

TR-1: needs to support diversified analytic processing and machines learning techniques (6: 1, 2, 3, 5, 6, 8,)

TR-2: needs to support batch and real time analytic processing (4: 2, 4, 7, 8)

TR-3: needs to support processing diversified data content (2: 1, 6)

TR-4: needs to support processing data in motion (streaming, fetching new content, tracking, etc.) (6: 1, 2, 4, 5, 7, 8)

TR-5: needs to support legacy and advance programming executable and libraries (5: 1, 2, 3, 5, 6)

**Data Infrastructure Requirements (to enable Transformation processing):**

DIR-1: needs to support legacy and advance software packages (subcomponent: SaaS) (5: 1, 2, 3, 5, 6)

DIR-2: needs to support legacy and advance computing platforms (subcomponent: PaaS) (3: 2, 5, 6)

DIR-3: needs to support legacy and advance distributed computing cluster (subcomponent: IaaS) (6: 1, 2, 3, 5, 6, 7)

DIR-4: needs to support elastic data transmission (subcomponent: networking) (3: 6, 7, 8)

DIR-5: needs to support legacy and advance distributed data storage (subcomponent: storage) (6: 1, 2, 3, 4, 5, 6,)

**Data Usage Requirements:**

DUR-1: needs to support fast search (~0.1 seconds) from processed data (1: 4)

DUR-2: needs to support diversified output file formats for rendering (7: 1, 2, 3, 4, 5, 6, 7)

DUR-3: needs to support visual layout for results presentation (2: 3, 4)

DUR-4: needs to support rich user interface (2: 3, 6)

DUR-5: needs to support streaming results to clients (1: 2)

**Security & Privacy Requirements:**

SnPR-1: needs to support security and privacy on protected data (7: 1, 2, 3, 4, 5, 6, 8)

SnPR-2: needs to support multi-level access control on protected data (6: 1, 3, 4, 5, 6, 8)

**Lifecycle Management Requirements:**

LMR-1: needs to support data quality curation (3: 1, 3, 4)

LMR-2: needs to support dynamic updates on data and user profiles (1: 2)

LMR-3: needs to support data lifecycle policy (1: 4)

LMR-4: needs to support data validation (1: 7)

LMR-5: needs to support human annotation for data validation (1: 6)

**System Management Requirements:**

SMR-1: needs to support rich user interface from mobile platforms to access processed results (1: 2, 5)

SMR-2: needs to support performance monitoring on analytic processing from mobile platforms (1: 3)

SMR-3: needs to support rich visual content rendering from mobile platforms (4: 1, 4, 5, 6)