NIST Special Publication 1500-5

# NIST Big Data Interoperability Framework: Volume 5, Architectures White Paper Survey

Final Version 1

NIST Big Data Public Working Group
Reference Architecture Subgroup

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.1500-5

National Institute of
Standards and Technology
U.S. Department of Commerce

# NIST Special Publication 1500-5

# NIST Big Data Interoperability Framework: Volume 5, Security and Privacy

## Final Version 1

NIST Big Data Public Working Group (NBD-PWG)
Reference Architecture Subgroup
*Information Technology Laboratory*

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.1500-5

September 2015

U. S. Department of Commerce
*Penny Pritzker, Secretary*

National Institute of Standards and Technology
*Willie May, Under Secretary of Commerce for Standards and Technology and Director*

**National Institute of Standards and Technology (NIST) Special Publication 1500-5**
53 pages (September 16, 2015)

NIST Special Publication series 1500 is intended to capture external perspectives related to NIST standards, measurement, and testing-related efforts. These external perspectives can come from industry, academia, government, and others. These reports are intended to document external perspectives and do not represent official NIST positions.

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

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

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. All NIST publications are available at http://www.nist.gov/publication-portal.cfm.

**Comments on this publication may be submitted to Wo Chang**

National Institute of Standards and Technology
Attn: Wo Chang, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8900) Gaithersburg, MD 20899-8930
Email: SP1500comments@nist.gov

## Reports on Computer Systems Technology

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

## Abstract

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

## Keywords

application interfaces; architecture survey; Big Data; Big Data analytics; Big Data infrastructure; Big Data management; Big Data storage; reference architecture.

# Acknowledgements

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

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

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

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

Milind Bhandarkar
*EMC/Pivotal*

Wo Chang
*National Institute of Standards and Technology*

Yuri Demchenko
*University of Amsterdam*

Barry Devlin
*9sight Consulting*

Harry Foxwell
*Oracle Press*

James Kobielus
*IBM*

Orit Levin
*Microsoft*

Robert Marcus
*ET-Strategies*

Tony Middleton
*LexisNexis*

Sanjay Mishra
*Verizon*

Sanjay Patil
*SAP*

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


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

# Table of Contents

## Figures



## Tables

# Executive Summary

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

Big Data Management

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

Big Data Analytics

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

Big Data Infrastructure

- In-memory data grids
- Operational database
- Analytic database
- Relational database
- Flat files
- Content management system
- Horizontal scalable architecture

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

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

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

Stage 1: Identify the high-level Big Data reference architecture key components, which are technology-, infrastructure-, and vendor-agnostic.
Stage 2: Define general interfaces between the NBDRA components.
Stage 3: Validate the NBDRA by building Big Data general applications through the general interfaces.

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

# 1 INTRODUCTION

## 1.1 BACKGROUND

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

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

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

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

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

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

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

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

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

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

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

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

Stage 1: Identify the high-level NBDRA key components, which are technology-, infrastructure-, and vendor-agnostic.
Stage 2: Define general interfaces between the NBDRA components.
Stage 3: Validate the NBDRA by building Big Data general applications through the general interfaces.

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

## 1.2 SCOPE AND OBJECTIVES OF THE REFERENCE ARCHITECTURE SUBGROUP

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

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

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

The NBDRA does not address the following:

- Detailed specifications for any organizations' operational systems;
- Detailed specifications of information exchanges or services; and
- Recommendations or standards for integration of infrastructure products.

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

## 1.3 REPORT PRODUCTION

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

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

## 1.4 REPORT STRUCTURE

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

- Section 2 contains the reference architectures submitted to the NBD-PWG.
- Section 3 discusses the surveyed reference architectures and highlights key functionalities.
- Section 4 presents several conclusions gained from the evaluation of the reference architectures.
- Appendix A contains acronyms used in this document.
- Appendix B lists references provided throughout this document.

## 1.5 FUTURE WORK ON THIS VOLUME

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

# 2 BIG DATA ARCHITECTURE PROPOSALS RECEIVED

The charter for the NBD-PWG Reference Architecture Subgroup identified the comparison of architectures as an important task. After the open forum NBD-PWG was formed, to fulfill the Reference Architecture Subgroup scope, participating members were asked submit Big Data architectures via the NBD-PWG reflector and the NBD-PWG website. The architectures included in Volume 5 reflect submissions from those that chose to participate and decided to provide architecture information. The NBD-PWG fully recognizes that the survey responses received does not cover the gamut of popular and available Big Data architectures. Additional resources were leveraged as input to the other documents in the *NIST Big Data Interoperability Framework*, such as those referenced in the Use Case document. The architectures described in this section were discussed at length by the NBD-PWG. The NBD-PWG Reference Architecture Subgroup sincerely appreciates contributions of the respective Big Data architectures from the following individuals and organizations. This survey would not have been possible without their contributions.

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

The architectures are presented in this section with the information originally submitted. The original content has not been modified. Specific vendor solutions and technologies are mentioned in the architecture solutions. However, the listing of these solutions and technologies does not constitute endorsement from the NBD-PWG. The front matter (page ii) contains a general disclaimer.

## 2.1 ET STRATEGIES

### 2.1.1 GENERAL ARCHITECTURE DESCRIPTION

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

### 2.1.2 ARCHITECTURE MODEL

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

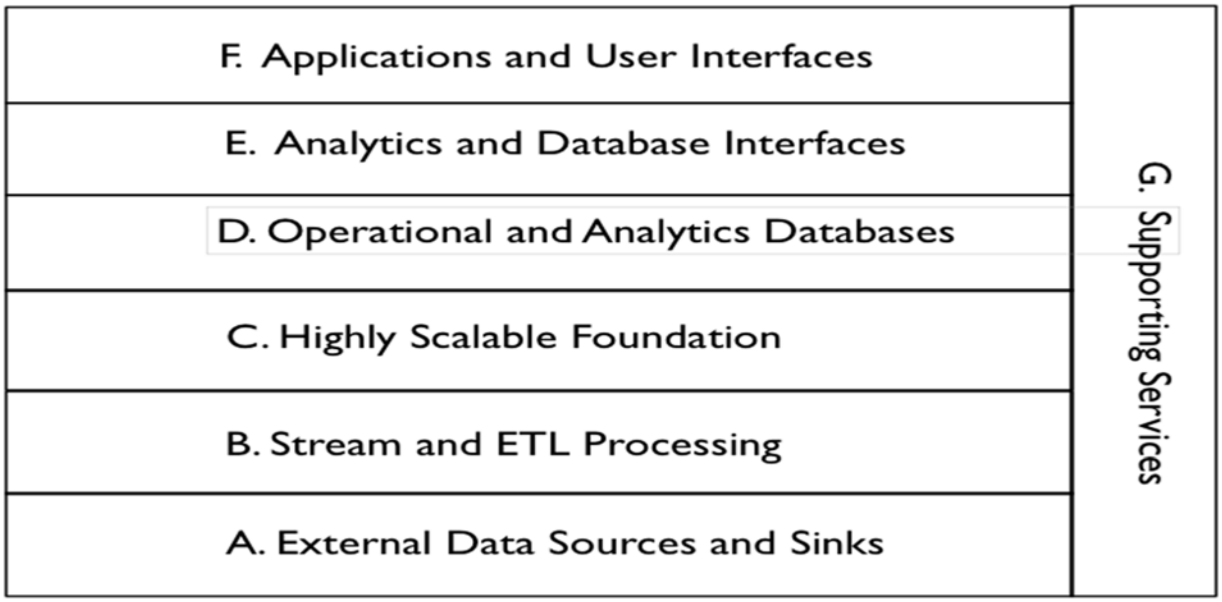



***Figure 1: Components of the High Level Reference Model***

#### A. External Data Sources and Sinks

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

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

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

#### C. Highly Scalable Foundation

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

#### D. Operational and Analytics Databases

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

#### E. Analytics and Database Interfaces

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

#### F. Applications and User Interfaces

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

#### G. Supporting Services

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

### 2.1.3 KEY COMPONENTS

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

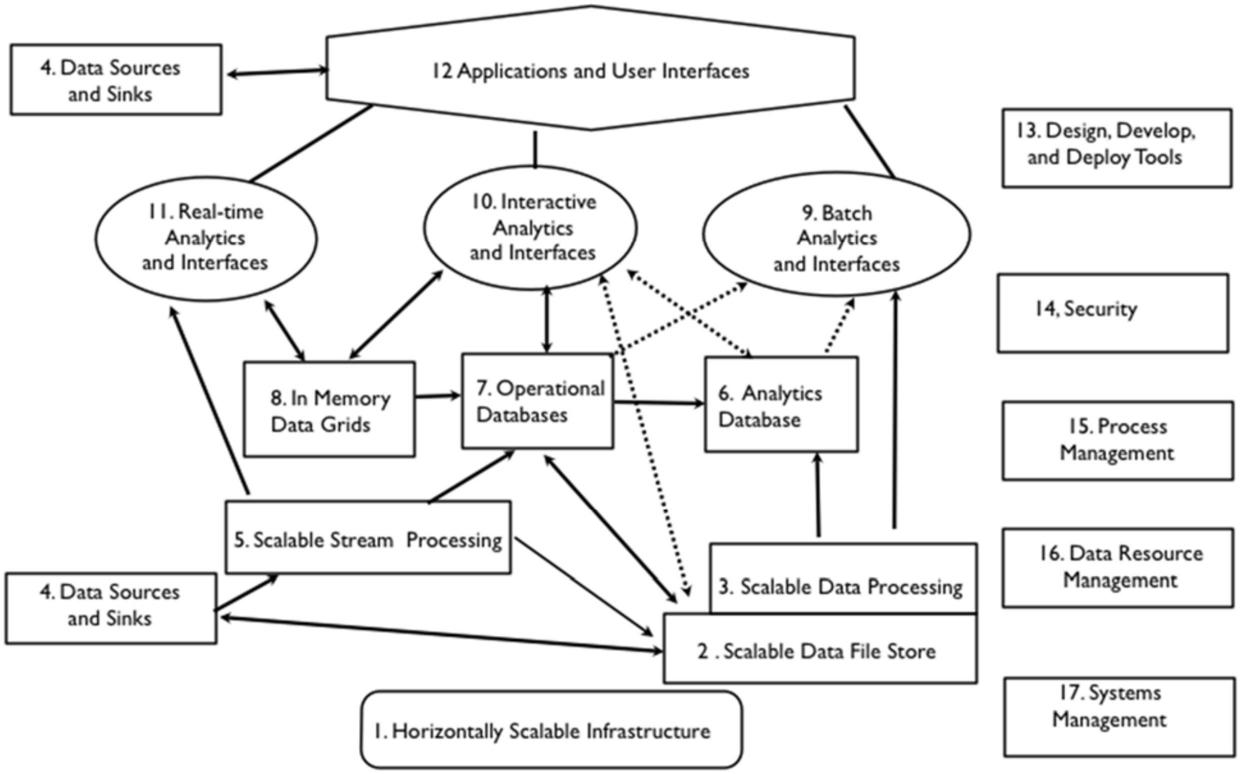



*Figure 2: Description of the Components of the Low Level Reference Model*

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

#### A. External Data Sources and Sinks

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

#### B. Stream and ETL Processing

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

## C. Highly Scalable Foundation

| | |
|---|---|
| **1. Scalable Infrastructure** | To support scalable Big Data stores and processing, the infrastructure should be able to support the easy addition of new resources. Possible platforms include public and/or private clouds. |
| **2. Scalable Data Stores** | This is the essence of Big Data architecture. Horizontal scalability using less expensive components can support the unlimited growth of data storage. However, there should be fault tolerance capabilities available to handle component failures. |
| **3. Scalable Processing** | To take advantage of scalable distributed data stores, the scalable distributed parallel processing should have similar fault tolerance. In general, processing should be configured to minimize unnecessary data movement. |

## D. Operational and Analytics Databases

| | |
|---|---|
| **6. Analytics Databases** | Analytics databases are generally highly optimized for read-only interactions (e.g., columnar storage, extensive indexing, and denormalization). It is often acceptable for database responses to have high latency (e.g., invoke scalable batch processing over large data sets). |
| **7. Operational Databases** | Operation databases generally support efficient write and read operations. NoSQL databases are often used in Big Data architectures in this capacity. Data can be later transformed and loaded into analytic databases to support analytic applications. |
| **8. In Memory Data Grids** | These high performance data caches and stores minimize writing to disk. They can be used for large-scale, real-time applications requiring transparent access to data. |

## E. Analytics and Database Interfaces

| | |
|---|---|
| **9. Batch Analytics and Interfaces** | These interfaces use batch scalable processing (e.g., Map/Reduce) to access data in scalable data stores (e.g., Hadoop File System). These interfaces can be SQL-like (e.g., Hive) or programmatic (e.g., Pig). |
| **10. Interactive Analytics and Interfaces** | These interfaces directly access data stores to provide interactive responses to end users. The data stores can be horizontally scalable databases tuned for interactive responses (e.g., HBase) or query languages tuned to data models (e.g., Drill for nested data). |
| **11. Real-Time Analytics and Interfaces** | Some applications require real-time responses to events occurring within large data streams (e.g., algorithmic trading). This complex event processing uses machine-based analytics requiring very high performance data access to streams and data stores. |

### F. Applications and User Interfaces

| | |
|---|---|
| **12. Applications and Visualization** | The key new capability available to Big Data analytic applications is the ability to avoid developing complex algorithms by utilizing vast amounts of distributed data (e.g., Google statistical language translation). However, taking advantage of the data available requires new distributed and parallel processing algorithms. |

### G. Supporting Services

| | |
|---|---|
| **13. Design, Develop, and Deploy Tools** | High-level tools are limited for the implementation of Big Data applications (e.g., Cascading). This should change to lower the skill levels needed by enterprise and government developers. |
| **14. Security** | Current Big Data security and privacy controls are limited (e.g., only Kerberos authentication for Hadoop, Knox). They should be expanded in the future by commercial vendors (e.g., Cloudera Sentry) for enterprise and government applications. |
| **15. Process Management** | Commercial vendors are supplying process management tools to augment the initial open source implementations (e.g., Oozie). |
| **16. Data Resource Management** | Open Source data governance tools are still immature (e.g., Apache Falcon). These will be augmented in the future by commercial vendors. |
| **17. System Management** | Open source systems management tools are also immature (e.g., Ambari). Fortunately robust system management tools are commercially available for scalable infrastructure (e.g., cloud-based). |

## 2.2 MICROSOFT

### 2.2.1 GENERAL ARCHITECTURE DESCRIPTION

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

### 2.2.2 ARCHITECTURE MODEL

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

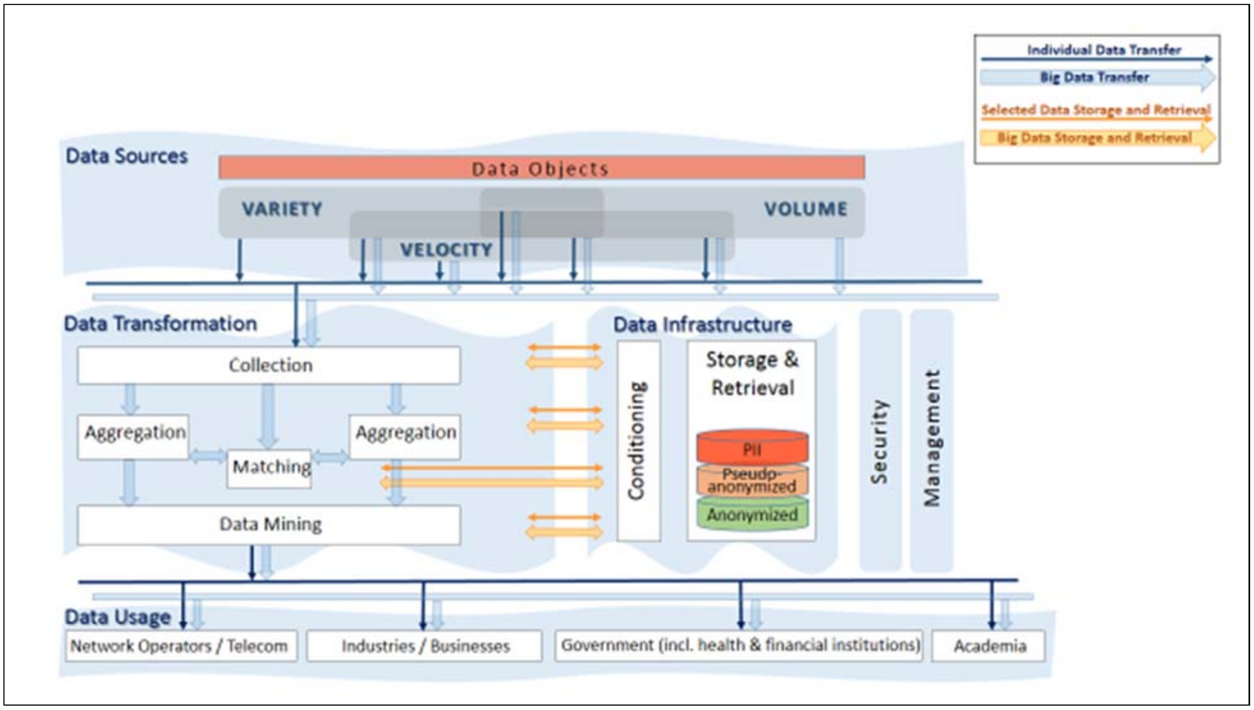



*Figure 3: Big Data Ecosystem Reference Architecture*

### 2.2.3 KEY COMPONENTS

#### Data Sources

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

#### Data Transformation

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

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

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

- **Aggregation:** Sets of existing data collections with easily correlated metadata (e.g., identical keys) are aggregated into a larger collection. As a result, the information about each object is enriched or the number of objects in the collection grows. Security considerations and policies concerning the resulting collection are typically similar to the original collections.
- **Matching:** Sets of existing data collections with dissimilar metadata (e.g., keys) are aggregated into a larger collection. For example, in the advertising industry, matching services correlate HTTP cookies' values with a person's real name. As a result, the information about each object is enriched. The security considerations and policies concerning the resulting collection are subject to data exchange interfaces design.
- **Data Mining:** According to DBTA,[4] "[d]ata mining can be defined as the process of extracting data, analyzing it from many dimensions or perspectives, and then producing a summary of the information in a useful form that identifies relationships within the data. There are two types of data mining: descriptive, which gives information about existing data; and predictive, which makes forecasts based on the data."

#### Data Infrastructure:

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

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

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

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

#### Data Usage:

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

## 2.3 UNIVERSITY OF AMSTERDAM

### 2.3.1 GENERAL ARCHITECTURE DESCRIPTION

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

- **Data Models, Structures, and Types:** The BDAF should support a variety of data types produced by different data sources. These data must be stored and processed and will, to some extent, define the Big Data infrastructure technologies and solutions.
- **Big Data Management Infrastructure and Services:** The BDAF should support Big Data Life Cycle Management, provenance, curation, and archiving. Big Data Life Cycle Management should support the major data transformations stages: collection, registration, filtering, classification, analysis, modeling, prediction, delivery, presentation, and visualization. Big Data

Management capabilities can be partly addressed by defining scientific or business workflows and using corresponding workflow management systems.

- **Big Data Analytics and Tools:** These specifically address required data transformation functionalities and related infrastructure components.
- **Big Data Infrastructure (BDI):** This component includes storage, computing infrastructure, network infrastructure, sensor networks, and target or actionable devices.
- **Big Data Security:** Security should protect data in rest and in motion, ensure trusted processing environments and reliable BDI operation, provide fine-grained access control, and protect users' personal information.

### 2.3.2 ARCHITECTURE MODEL

Figure 4 illustrates the basic Big Data analytics capabilities as a part of the overall cloud-based BDI.

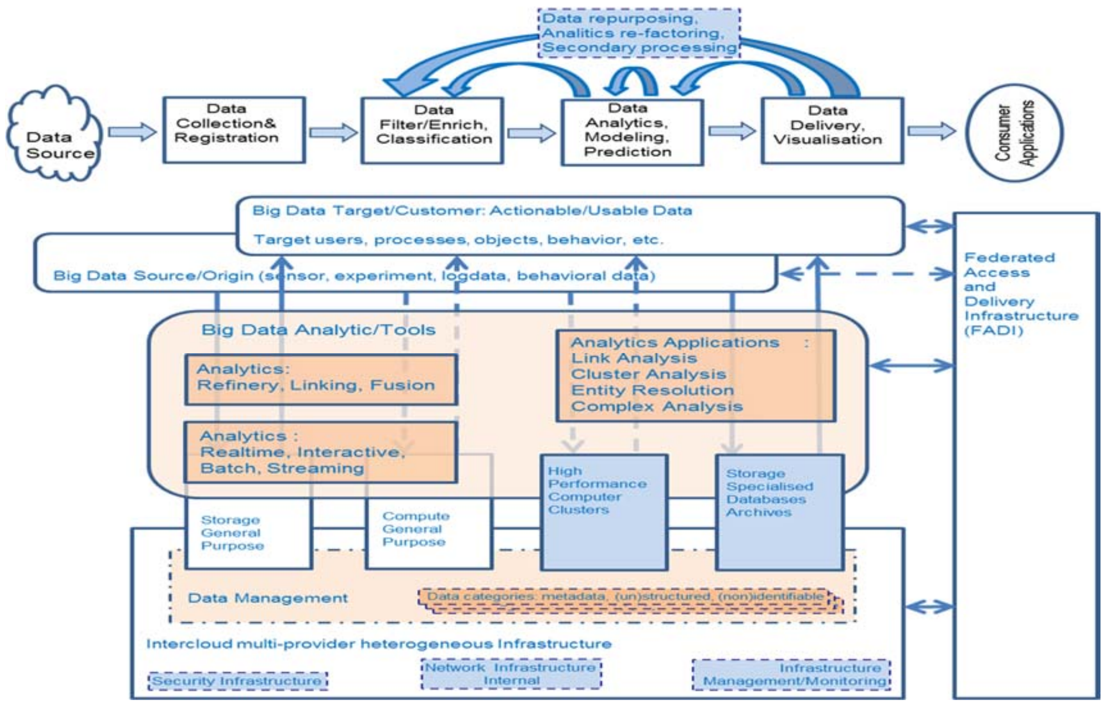



*Figure 4: Big Data Architecture Framework*

In addition to the general cloud-based infrastructure services (e.g., storage, computation, infrastructure or VM management), the following specific applications and services will be required to support Big Data and other data centric applications:

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

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

### 2.3.3 KEY COMPONENTS

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

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

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

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

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

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

## 2.4 IBM

### 2.4.1 GENERAL ARCHITECTURE DESCRIPTION

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

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

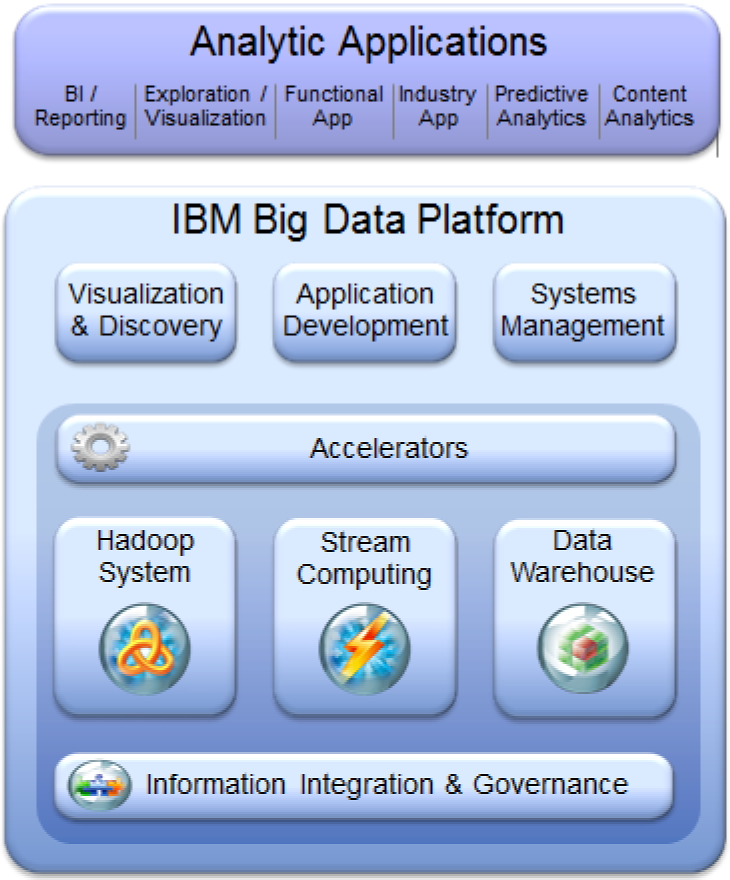



***Figure 5: IBM Big Data Platform***

### 2.4.2 ARCHITECTURE MODEL

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

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

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

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

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

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

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

### 2.4.3 KEY COMPONENTS

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

- **Tools:** These components support visualization, discovery, application development, and systems management.
- **Accelerators:** This component provides a rich library of analytical functions, schemas, tool sets, and other artifacts for rapid development and delivery of value in Big Data projects.
- **Hadoop:** This component supports managing and analyzing unstructured data. To support this requirement, IBM InfoSphere BigInsights and PureData System for Hadoop support are required.
- **Stream Computing:** This component supports analyzing in-motion data in real time.
- **Data Warehouse:** This component supports business intelligence, advanced analytics, data governance, and master data management on structured data.
- **Information Integration and Governance:** This component supports integration and governance of all data sources. Its capabilities include data integration, data quality, security, life cycle management, and master data management.

## 2.5 ORACLE

### 2.5.1 GENERAL ARCHITECTURE DESCRIPTION

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

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

### 2.5.2 ARCHITECTURE MODEL

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

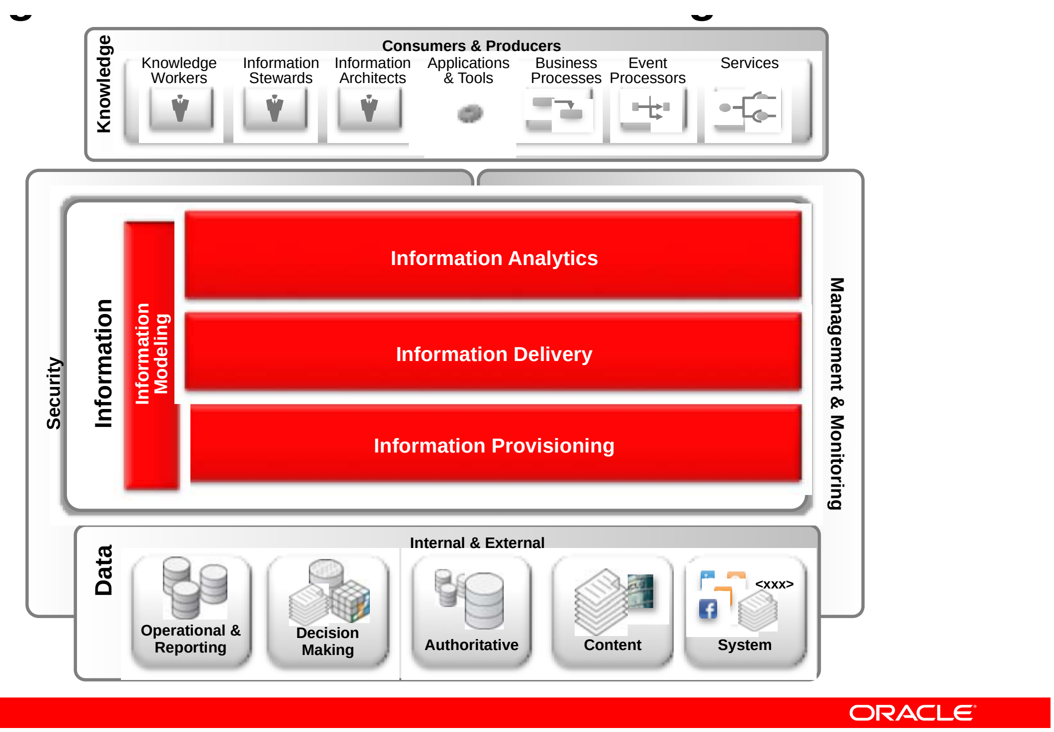



*Figure 6: High level, Conceptual View of the Information Management Ecosystem*

### 2.5.3 KEY COMPONENTS

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

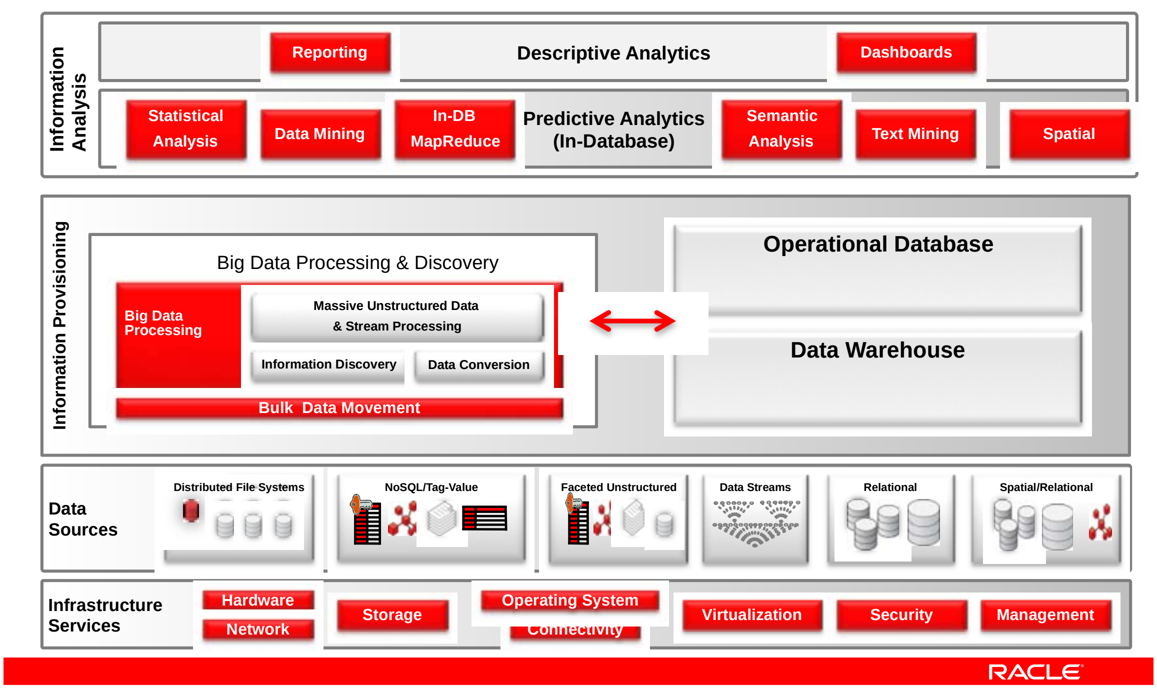



*Figure 7: Oracle Big Data Reference Architecture*

## 2.6 PIVOTAL

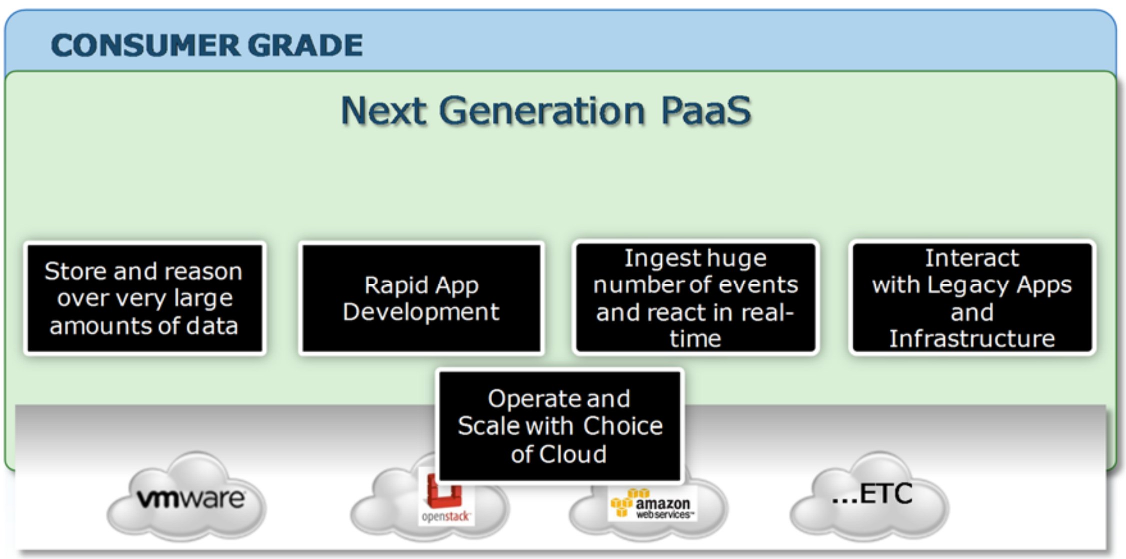
### 2.6.1 GENERAL ARCHITECTURE DESCRIPTION

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

### 2.6.2 ARCHITECTURE MODEL

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

***Figure 8: Pivotal Architecture Model***




**Infrastructure**

- Most of the early Big Data platforms instances assumed a networked bare metal infrastructure. Speed to market drove the early adopters to offer this PaaS on the bare metal platforms. Monitoring and management capabilities were built around managing a bare metal cluster at scale. The numbers of Big Data platform nodes at some of the early adopters are in the many tens of thousands. Custom management tools were developed for implemented limited elasticity by adding nodes, retiring nodes and monitoring and alerting in case of infrastructure failures.
- Since then, virtualization has been widely adopted in enterprises, since it offers enhanced isolation, utilization, elasticity, and management capabilities inherent in most virtualization technologies.

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

**Data Storage and Analytics**

- Big Data platforms deal with the gravity of data (i.e., difficulty of moving large amounts of data over existing network capabilities) by moving processing very close to the storage tier. As a result, the analytics and data storage tiers were fused together in the Pivotal architecture. Every attempt was made to move analytics closer to the data layer. Now, high-speed data center interconnects (with 10+ Gbps bandwidth) are becoming more affordable, making it possible to keep the processing within the same subnet and still deliver acceptable performance.
- Virtualization technology is also innovating to add data locality awareness to the compute node selection, thereby speeding up analytics workloads over large amounts of data, while offering the benefits of resource isolation, better security, and improved utilization, compared to the underlying bare metal platform.
- Pivotal finds Data Lake is one of the most prevalent emerging usage patterns of the Big Data platforms. Data Lake—a design pattern where streams of data are stored and available for historical and operational analytics on the Big Data platform—is the starting point for most of the enterprises. Once the data is available in one place, business users experiment with use cases and solutions to business problems, resulting in second order benefits.

**Big Data Applications**

- Pivotal considers extracting insight from a Big Data platform's data as a Big Data application. Excel spreadsheets, and PowerPoint presentations are used for early demonstrations of these insights, and once the business cases are approved, applications need to be built to act on these insights. Developers are looking for flexibility to integrate these new Big Data applications with existing applications or build completely new applications. A good platform will make it easier for the developers to do both at the same time.
- In addition, the operations team for the application requires a platform to easily deploy, manage and scale the application as the usage increases. These operability requirements are very critical for enterprise customers.

### *2.6.3 Key Components*

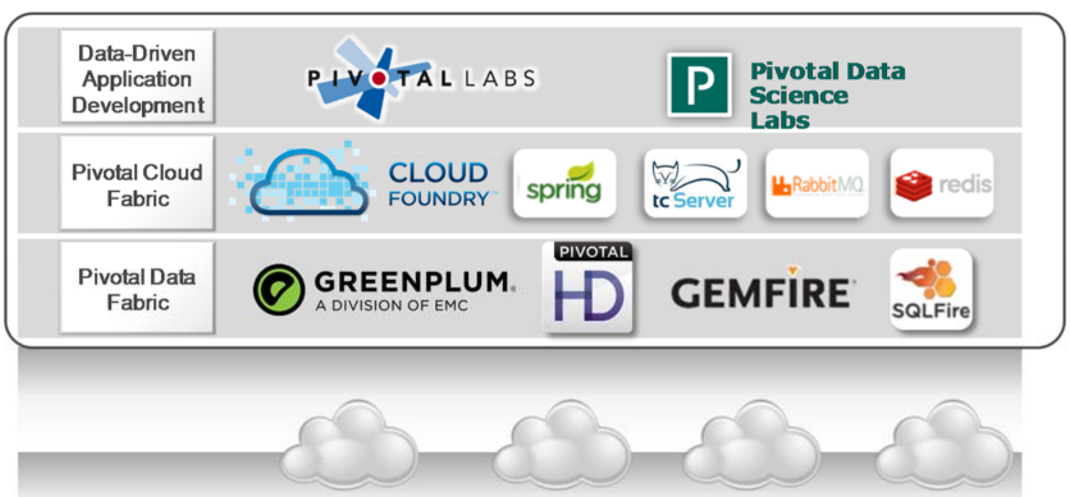



***Figure 9: Pivotal Data Fabric and Analytics***

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

#### Hypothesis: Historical structured and unstructured access pattern

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

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

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

#### Model Operationalization: Run models in operational systems

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

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

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

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

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

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

## 2.7 SAP

### 2.7.1 GENERAL ARCHITECTURE DESCRIPTION

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

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

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

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

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

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

### 2.7.2 ARCHITECTURE MODEL

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

- An architecture that puts in-memory technology data at its core and maximizes computational efficiencies by bringing the compute and data layers together.
- Support for a variety of data processing engines (such as transactional, analytical, graph, and spatial) operating directly on the same data set in memory.
- Interoperability/integration of best of breed technologies such as Hadoop/Hive for the data layer, including data storage and low level processing.
- The ability to leverage these technologies to transform existing business applications and build entirely new ones that were previously not practical.
- Comprehensive native support for predictive analytics, as well as interoperability with popular libraries such as R, for enabling roles such as data scientists to uncover and predict new business potentialities.

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

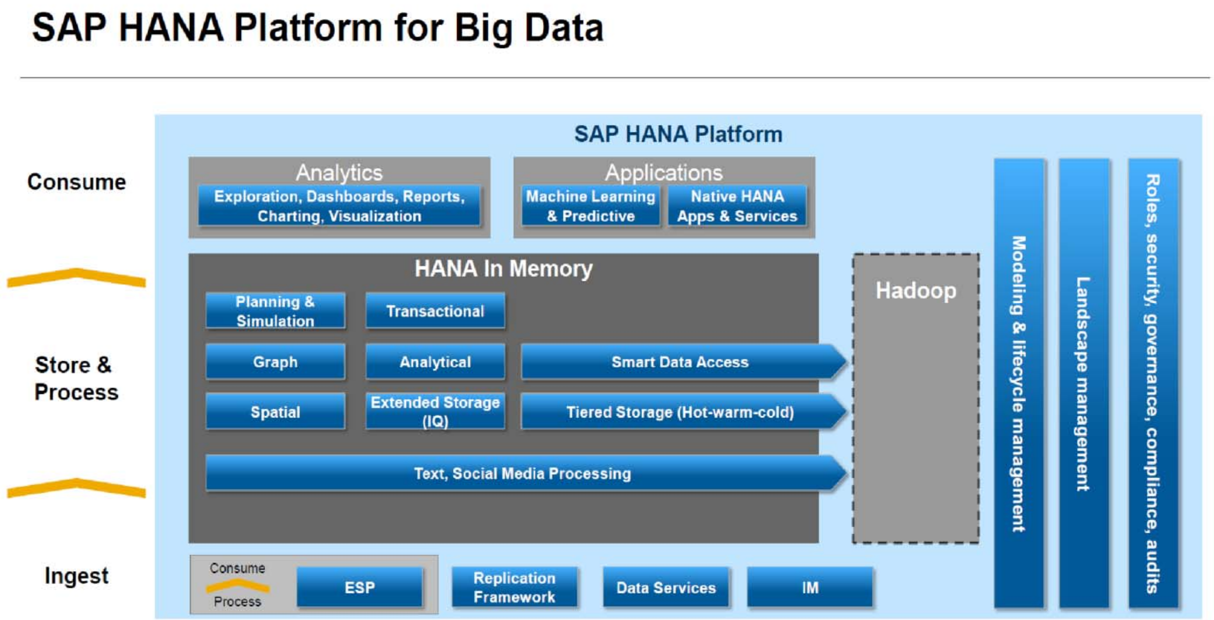



*Figure 10: SAP Big Data Reference Architecture*

### 2.7.3 KEY COMPONENTS

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

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

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

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

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

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

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

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

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

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

## 2.8 9SIGHT

### 2.8.1 GENERAL ARCHITECTURE DESCRIPTION

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

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

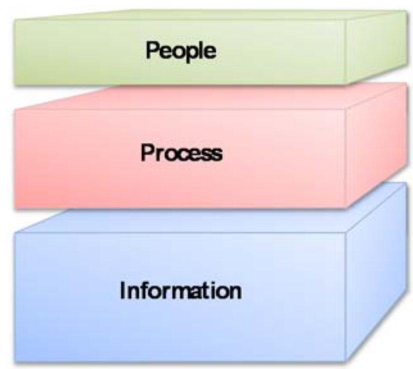



*Figure 11: 9Sight General Architecture*

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

### 2.8.2 ARCHITECTURE MODEL

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

### 2.8.3 KEY COMPONENTS

Business applications or workflows, whether operational, informational, or collaborative, with their business focus and wide variety of goals and actions, are gathered together in a single component, **utilization**.

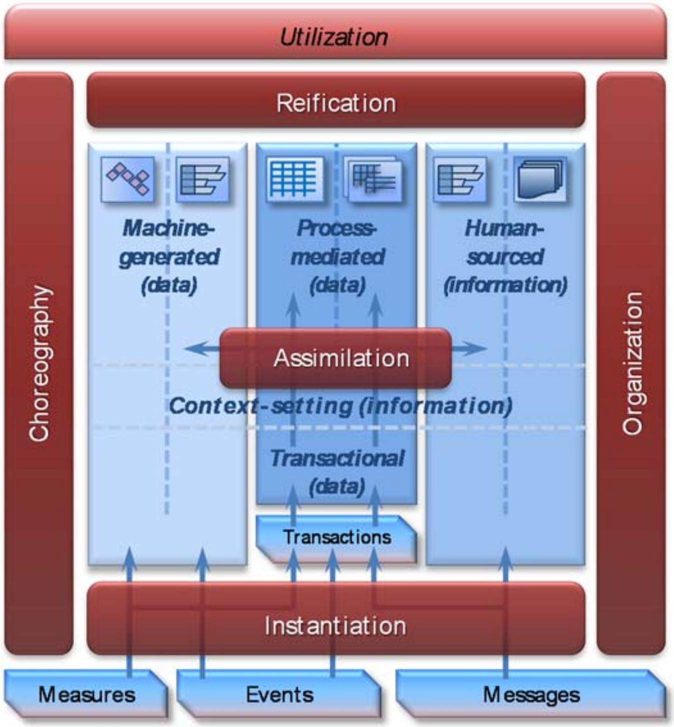



*Figure 12: 9Sight Architecture Model*

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

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

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

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

- **Human-sourced information:** Information originates from people, because context comes only from the human intellect. This information is the highly subjective record of human experiences and is now almost entirely digitized and electronically stored everywhere from tweets to movies. Loosely structured and often ungoverned, this information may not reliably represent for the business what has happened in the real world.
- **Process-mediated data:** Business processes record well-defined, legally binding business events. This process-mediated data is highly structured and regulated, and includes transactions, reference tables and relationships, and the metadata that sets its context. Process-mediated data includes operational and BI systems and was the vast majority of what IT managed in the past. It is amenable to information management and to storage and manipulation in relational database systems.
- **Machine-generated data:** Sourced from the sensors, computers, etc. used to record events and measures in the physical world, such data is well-structured and usually reliable. As the Internet of Things grows, well-structured machine-generated data is of growing importance to business. Some claim that its size and speed is beyond traditional RDBMS, mandating NoSQL stores. However, high-performance RDBMSs are also often used.

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

## 2.9 LEXISNEXIS

### 2.9.1 GENERAL ARCHITECTURE DESCRIPTION

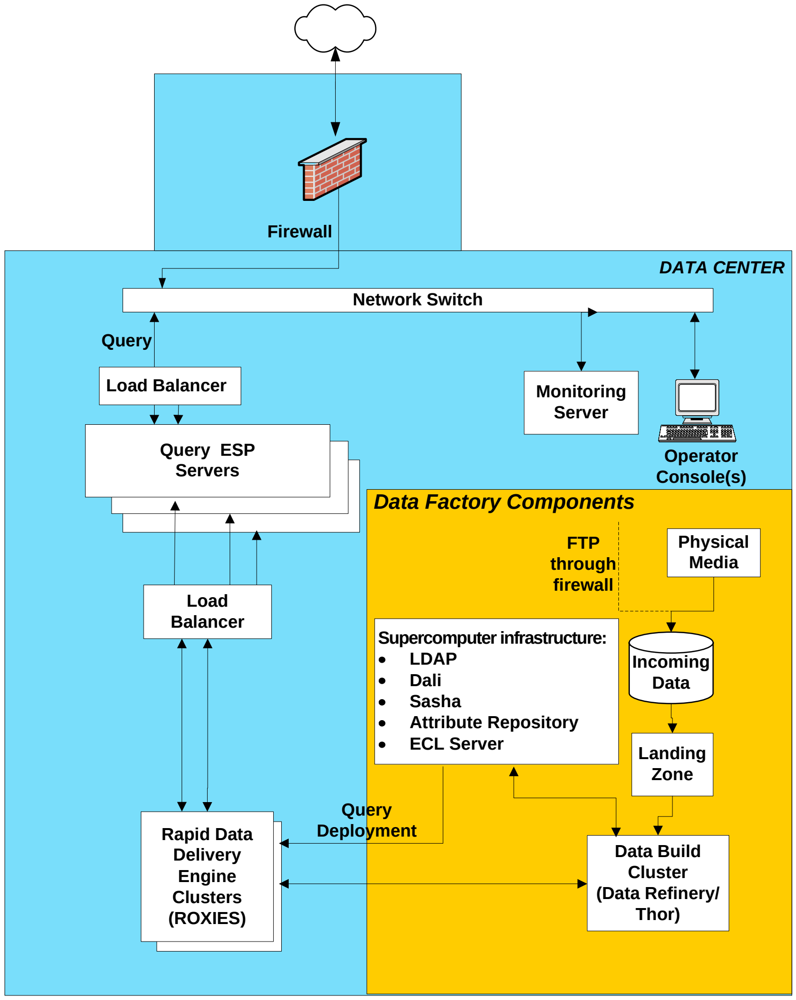



*Figure 13: Lexis Nexis General Architecture*

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

### 2.9.2 ARCHITECTURE MODEL

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

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

- **The Enterprise Control Language (ECL): compiler, executable code generator and job server (ECL Server):** Serves as the code generator and compiler that translate ECL code.
- **System data store (Dali):** Used for environment configuration, message queue maintenance, and enforcement of LDAP security restrictions.
- **Archiving server (Sasha):** Serves as a companion 'housekeeping' server to Dali.
- **Distributed File Utility (DFU Server):** Controls the spraying and despraying operations used to move data onto and out of THOR.
- **The inter-component communication server (ESP Server):** The inter-component communication server that allows multiple services to be "plugged in" to provide various types of functionality to client applications via multiple protocols.

### 2.9.3 KEY COMPONENTS

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

- **THOR Data Refinery:** THOR is a massively parallel ETL engine that can be used for performing a variety of tasks such as massive: joins, merges, sorts, transformations, clustering, and scaling. Essentially, THOR permits any problem with computational complexities $O(n^2)$ or higher to become tractable.
- **ROXIE Data Delivery:** ROXIE serves as a massively parallel, high throughput, structured query response engine. It is suitable for performing volumes of structured queries and full text ranked Boolean search, and can also operate in highly available (HA) environments due to its read-only nature. ROXIE also provides real-time analytics capabilities, to address real-time classifications, prediction, fraud detection and other problems that normally require handling processing and analytics on data streams.

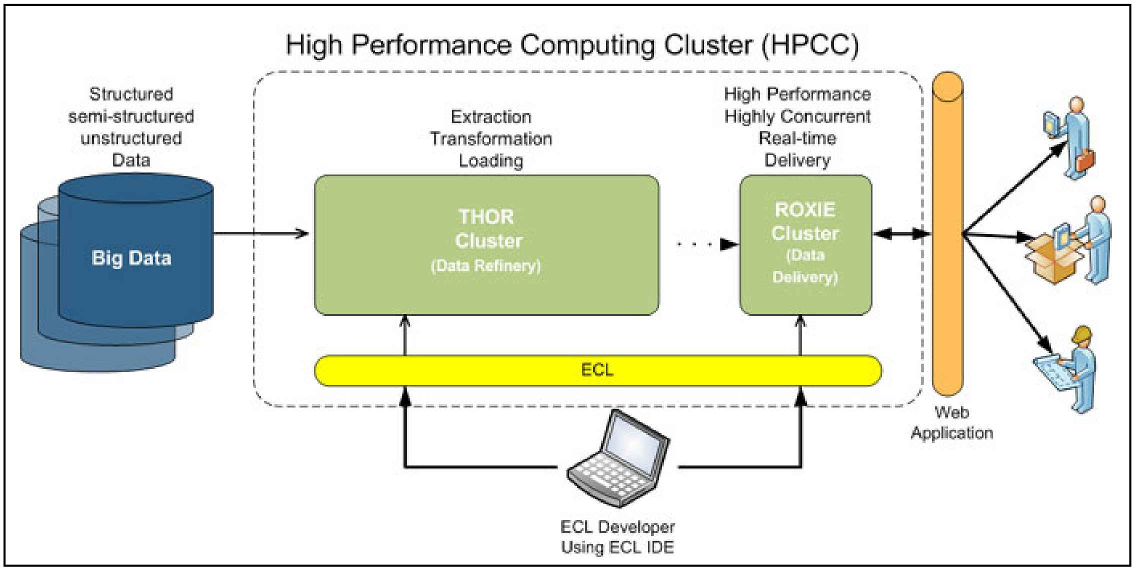



*Figure 14: Lexis Nexis High Performance Computing Cluster*

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

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

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

THOR is capable of operating either independently or in tandem with ROXIE; when ROXIE is present it hosts THOR results and makes them available to the end-user through a web service application programming interface.

# 3 SURVEY OF BIG DATA ARCHITECTURES

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

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

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

## 3.1 BOB MARCUS

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

### A. Data Sources and Sinks

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

### B. Application and User Interfaces

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

### C. Analytics Databases and Interfaces

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

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

| | Database Types | Description |
|---|---|---|
| Databases | Analytics Databases | In general, these are highly optimized for read-only interactions and typically acceptable for database responses to have high latency (e.g., invoke scalable batch processing over large data sets). |
| | Operational Databases | In general, these support efficient write and read operations. NoSQL databases are often used in Big Data architectures in this capacity. Data can later be transformed and loaded into analytic databases to support analytic applications. |
| | In Memory Data Grids | These high performance data caches and stores minimize writing to disk. They can be used for large scale real-time applications requiring transparent access to data. |

| | Database Types | Description |
|---|---|---|
| Analytics and Database Interfaces | Batch Analytics and Database Interfaces | These interfaces use batch scalable processing (e.g., Map/Reduce) to access data in scalable data stores (e.g., Hadoop File System). These interfaces can be SQL-like (e.g., Hive) or programmatic (e.g., Pig). |
| | Interactive Analytics and Interfaces | These interfaces avoid direct access data stores to provide interactive responses to end users. The data stores can be horizontally scalable databases tuned for interactive responses (e.g., HBase) or query languages tuned to data models (e.g., Drill for nested data). |
| | Real-Time Analytics and Interfaces | Some applications require real-time responses to events occurring within large data streams (e.g., algorithmic trading). This complex event processing uses machine-based analytics, which require very high performance data access to streams and data stores. |

### D. Scalable Stream and Data Processing

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

### E. Scalable Infrastructure

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

### F. Supporting Services

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

- **Design, Develop and Deploy Tools:** The framework cautions that high level tools are limited for the implementation of Big Data applications. This should change to lower the skill levels needed by enterprise and government developers.
- **Security:** The framework also notes lack of standardized and adequate support to address data security and privacy. The framework cites that only Kerberos authentication for Hadoop, Knox exists. Capabilities should be expanded in the future by commercial vendors for enterprise and government applications.
- **Process Management:** The framework notes that commercial vendors are supplying process management tools to augment the initial open source implementations (e.g., Oozie).
- **Data Resource Management:** The author notes that open source data governance tools are still immature (e.g., Apache Falcon). These will be augmented in the future by commercial vendors.
- **System Management:** The framework notes that open source systems management tools are also immature (e.g., Ambari). However, robust system management tools are commercially available for scalable infrastructure (e.g., cloud-based).

Figure 15 provides diagrammatic representation of the architecture.

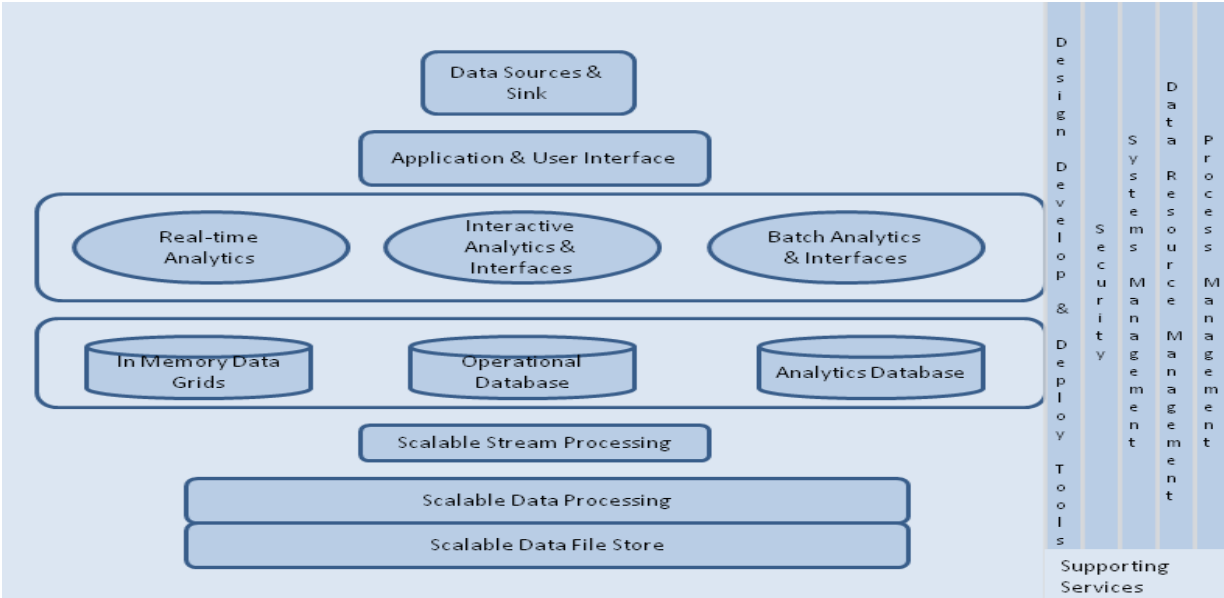



*Figure 15: Big Data Layered Architecture*

## 3.2 MICROSOFT

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

### A. Data Sources

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

### B. Data Transformation

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

***Table 2: Microsoft Data Transformation Steps***

| Data Transformation Steps | Functional Description |
|---|---|
| Data Collection | Data can be collected for different types and forms; similar sources and structure resulting in uniform security considerations, policies, and allows creation of an initial metadata |
| Aggregation | This is defined by Microsoft as where sets of existing data is collected to form an easily correlated metadata (e.g., identical keys) and then aggregated into a larger collection thus enriching number of objects as the collection grows. |

| Data Transformation Steps | Functional Description |
|---|---|
| Matching | This is defined as where sets of existing data collections with dissimilar metadata (e.g., keys) are aggregated into a larger collection. Similar to aggregation this stage also enhances information about each object. |
| Data Mining | Microsoft refers this as a process where data, analyzing it from many dimensions or perspectives, then producing a summary of the information in a useful form that identifies relationships within the data. There are two types of data mining: descriptive, which gives information about existing data; and predictive, which makes forecasts based on the data |

### C. Data Infrastructure

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

### D. Data Usage

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

## 3.3 UNIVERSITY OF AMSTERDAM

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

### A. Data Models

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

### B. Big Data Analytics

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

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

### C. Big Data Management

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

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

### D. Big Data Infrastructure

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

- General cloud-based infrastructure, platform, services, and applications to support creation, deployment, and operation of Big Data infrastructures and applications (using generic cloud features of provisioning on-demand, scalability, measured services.
- Collaborative environment infrastructure (e.g., group management) and user-facing capabilities (e.g., user portals, identity management/federation).
- Network infrastructure that interconnects typically distributed and increasingly multi-provider BDI components that may include intra-cloud (intra-provider) and Intercloud network infrastructure.
- FADI, which is to be treated as a part of the general Inter-cloud infrastructure of the BDI. FADI combines both inter-cloud network infrastructure and corresponding federated security infrastructure to support infrastructure components integration and users federation. FADI is an important component of the overall cloud and Big Data infrastructure that interconnects all the major components and domains in the multi-provider inter-cloud infrastructure including non-cloud and legacy resources. Using federation model for integrating multi-provider heterogeneous services and resources reflects current practice in building and managing complex infrastructures and allows for inter-organizational resource sharing and identity federation.

### E. Big Data Security

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

## 3.4 IBM

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

### A. Data Discovery and Exploration (Data Source)

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

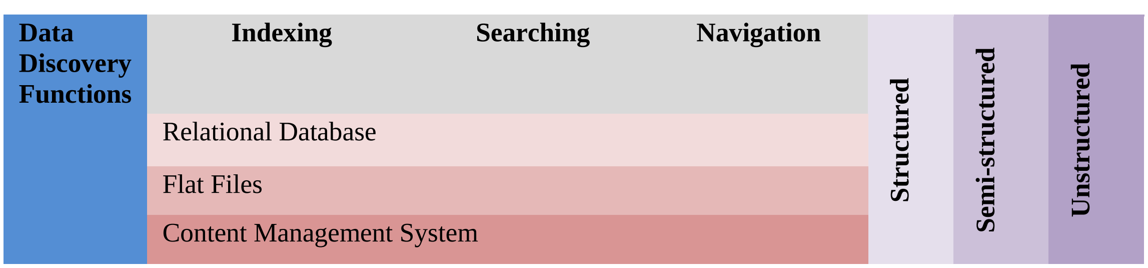



***Figure 16: Data Discovery and Exploration***

### B. Data Analytics

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

Within this context, IBM recommends the following:

- **Manage and analyze unstructured data:** IBM notes that a game-changing analytics platform must be able to manage, store, and retrieve both unstructured and structured data. It also has to provide tools for unstructured data exploration and analysis.
- **Analyze data in real time:** Performing analytics on activity as it unfolds presents a huge untapped opportunity for the analytics enterprise.

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

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

### C. Big Data Platform Infrastructure

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

- **Hadoop:** This component supports managing and analyzing unstructured data. To support this requirement, IBM InfoSphere, BigInsights, and PureData System for Hadoop support are required.
- **Stream Computing:** This component supports analyzing in-motion data in real time.
- **Accelerators:** This component provides a rich library of analytical functions, schemas, tool sets and other artifacts for rapid development and delivery of value in Big Data projects.

### D. Information Integration and Governance

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

## 3.5 ORACLE

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

### A. Information Analytics

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

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

### B. Information Provisioning

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

### C. Data Sources

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

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

### D. Infrastructure Services

The following capabilities support the infrastructure services:

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

## 3.6 PIVOTAL

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

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

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

### Pivotal Data Fabric and Analytics

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

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

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

## 3.7 SAP

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

### A. Data Ingestion

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

### B. Data Storage and Processing

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

#### C. Data Consumption

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

#### D. Data life cycle management

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

#### E. Infrastructure Management

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

#### F. Data Security and Governance

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

## 3.8 9SIGHT

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

- **Utilization** is a component that gathers operational, informational, or collaborative business applications/workflows with their business focus and wide variety of goals and actions.
- **Instantiation** is the process by which measures, events, and messages from the physical world are represented as, or converted to, transactions or instances of information within the enterprise environment.
- **Assimilation** is a process that creates reconciled and consistent information, using ETL and data virtualization tools, before users have access to it.
- **Reification**, which sits between all utilization functions and the information itself, provides a consistent, cross-pillar view of information according to an overarching model. Reification allows access to the information in real time, and corresponds to data virtualization for ‘online’ use.
- **Choreography** is a component that supports both the SOA process and services-based approach to delivering output. Choreography coordinates the actions of all participating elements to produce desired outcomes. There are two subcomponents: *adaptive workflow management* and an *extensible message bus*. These functions are well known in standard SOA work.
- **Organization** is a component that covers all design, management, and governance activities relating to both processes and information.

## 3.9 LEXISNEXIS

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

### HPCC Architecture Model

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

1. **Data Refinery** is a massively parallel ETL engine used for performing a variety of tasks such as massive: joins, merges, sorts, transformations, clustering, and scaling. The component permits any problem with computational complexities $O(n^2)$ or higher to become tractable. This component supports Big Data in structured, semi-structured, or unstructured data forms.
2. **Data Delivery** component serves as a massively parallel, high-throughput, structured query response engine. It is suitable for performing volumes of structured queries and full text-ranked Boolean search, and can operate in HA environments due to its read-only nature. This component also provides real-time analytics capabilities to address real-time classifications, prediction, fraud detection, and other problems that normally require handling processing and analytics on data streams.

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

- **ECL compiler** is an executable code generator and job server (ECL Server) that translates ECL code. The ECL is an open source, data-centric programming language that supports data refinery and data delivery components for large-scale data management and query processing.
- **System Data Stor**e is used for environment configuration, message queue maintenance, and enforcement of LDAP security restrictions.
- **Archiving server** is used for housekeeping purposes.
- **Distributed File Utility** moves data in/out of Data Refinery component.
- **The inter-component communication server (ESP Server)** allows multiple services to be "plugged in" to provide various types of functionality to client applications via multiple protocols.

## 3.10 COMPARATIVE VIEW OF SURVEYED ARCHITECTURES

The working group members were allowed to submit architectures as they existed within their implementation without any constraints or a set template. The submitted architectures were not intended to align well with one another. There was no requirement from the NBD-PWG to provide the architecture in any specific format. Therefore, Figure 17 was developed to visualize the commonalities and differences among the surveyed architectures. The architectures were stacked vertically to facilitate comparison of the components between the surveyed architectures. The components within each architecture were aligned horizontally. Three common components were observed in many of the surveyed architectures and are as follows:

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

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

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

| | |
|---|---|
| Big Data Management and Storage | |
| Big Data Analytics and Application Interfaces | |
| Big Data Infrastructure | |
| Other parts of the architecture | |

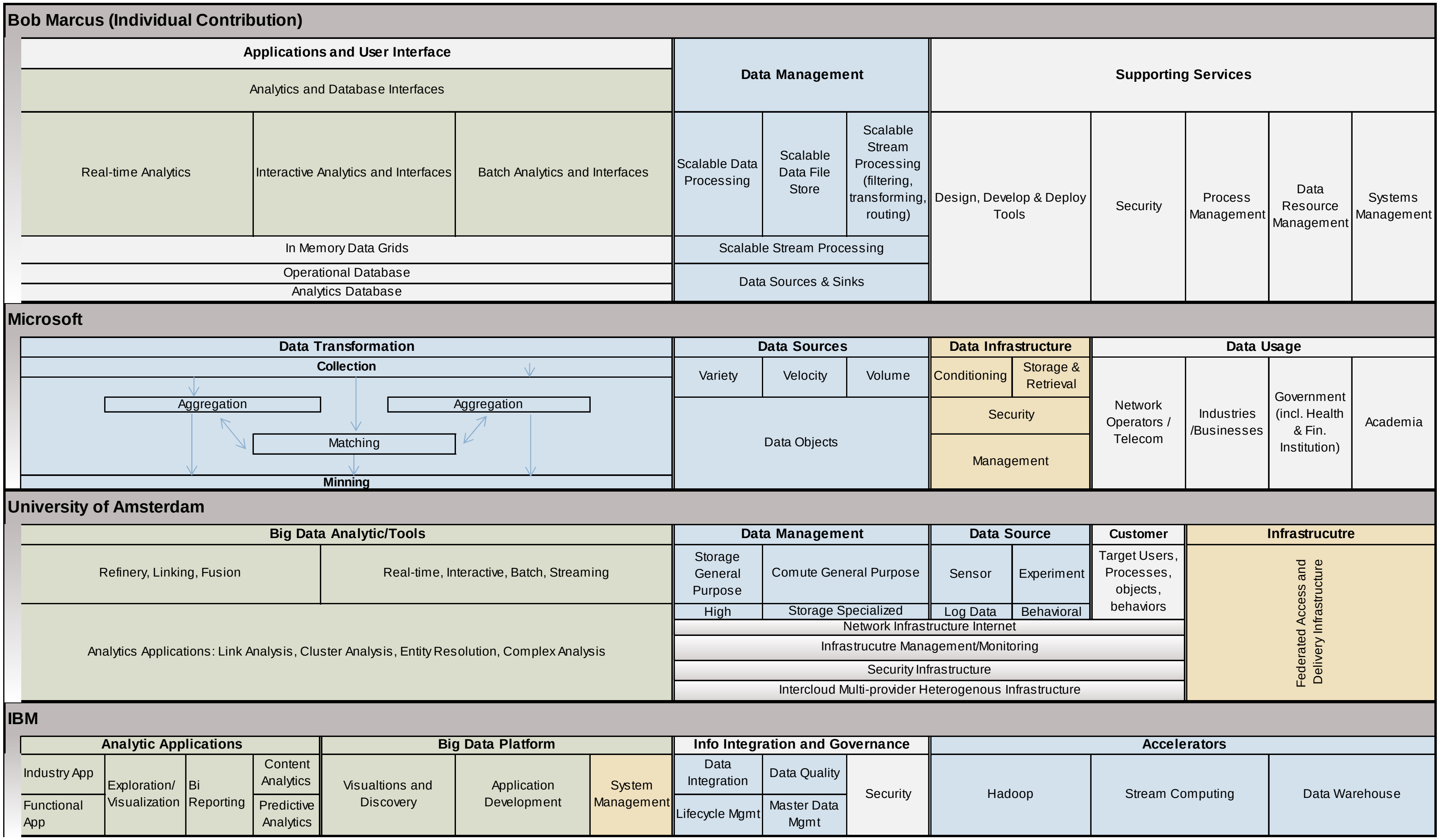



*Figure 17(a): Stacked View of Surveyed Architecture*

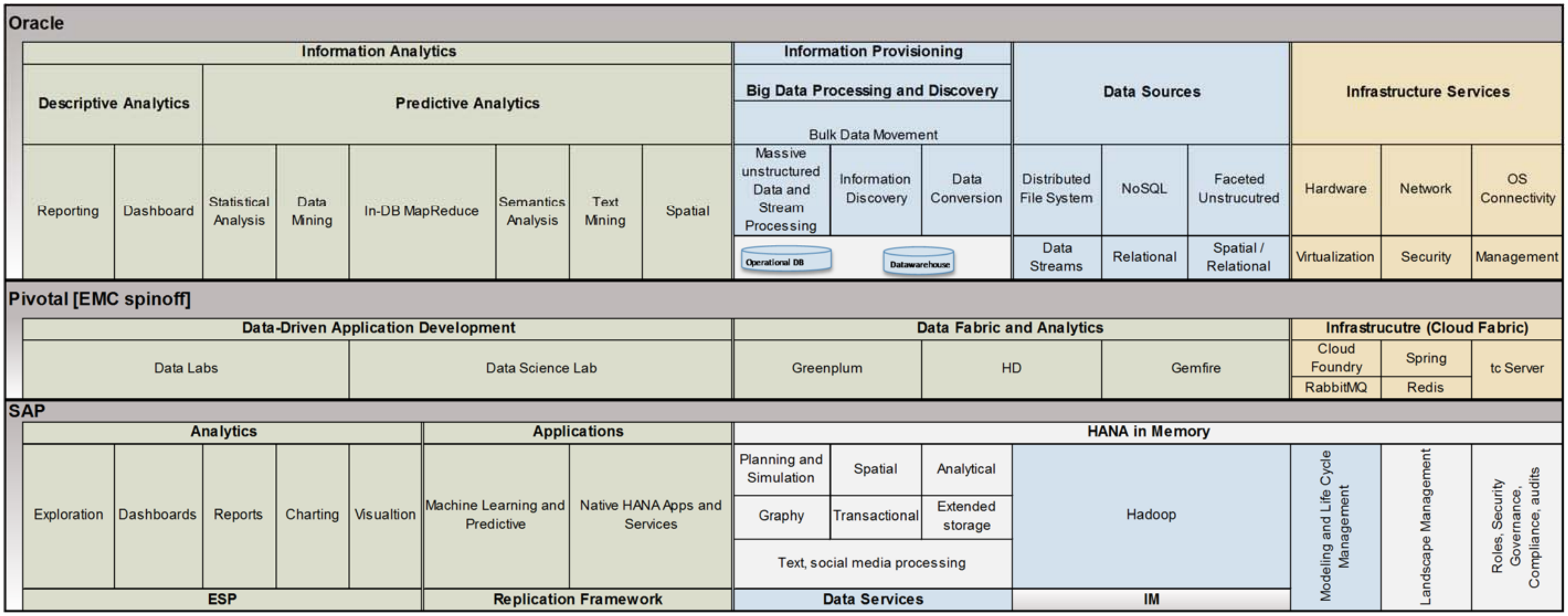



*Figure 17(b): Stacked View of Surveyed Architecture (continued)*

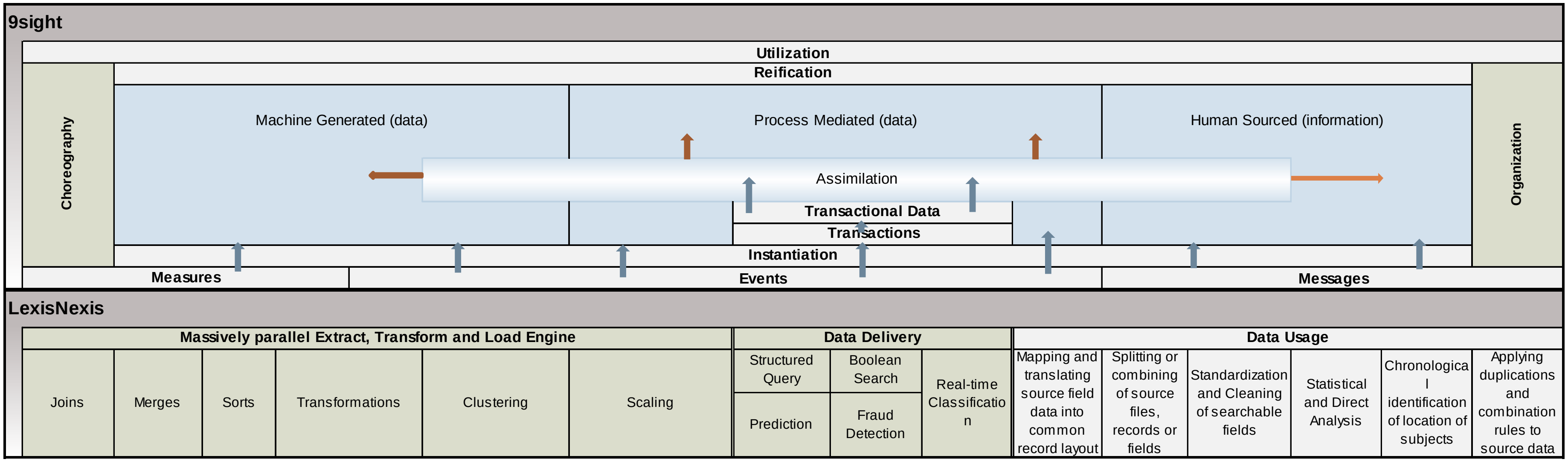



*Figure 17(c): Stacked View of Surveyed Architecture (continued)*

# 4 CONCLUSIONS

Through the collection, review, and comparison of Big Data architecture implementations, numerous commonalities were identified. These commonalities between the surveyed architectures aided in the development of the NBDRA. Even though each Big Data system is tailored to the needs of the particular implementation, certain key components appear in most of the implementations. Three general components were observed in the surveyed architectures as outlined below with key considerations listed with each component. Figure 18 contains additional information about the three general components.

- Big Data Management and Storage
    - Structured, semi-structured and unstructured data
    - Volume, variety, velocity, and variability
    - SQL and NoSQL
    - Distributed file system
- Big Data Analytics and Application Interfaces
    - Descriptive, predictive and spatial
    - Real-time
    - Interactive
    - Batch analytics
    - Reporting
    - Dashboard
- Big Data Infrastructure
    - In memory data grids
    - Operational database
    - Analytic database
    - Relational database
    - Flat files
    - Content management system
    - Horizontal scalable architecture

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

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

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

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

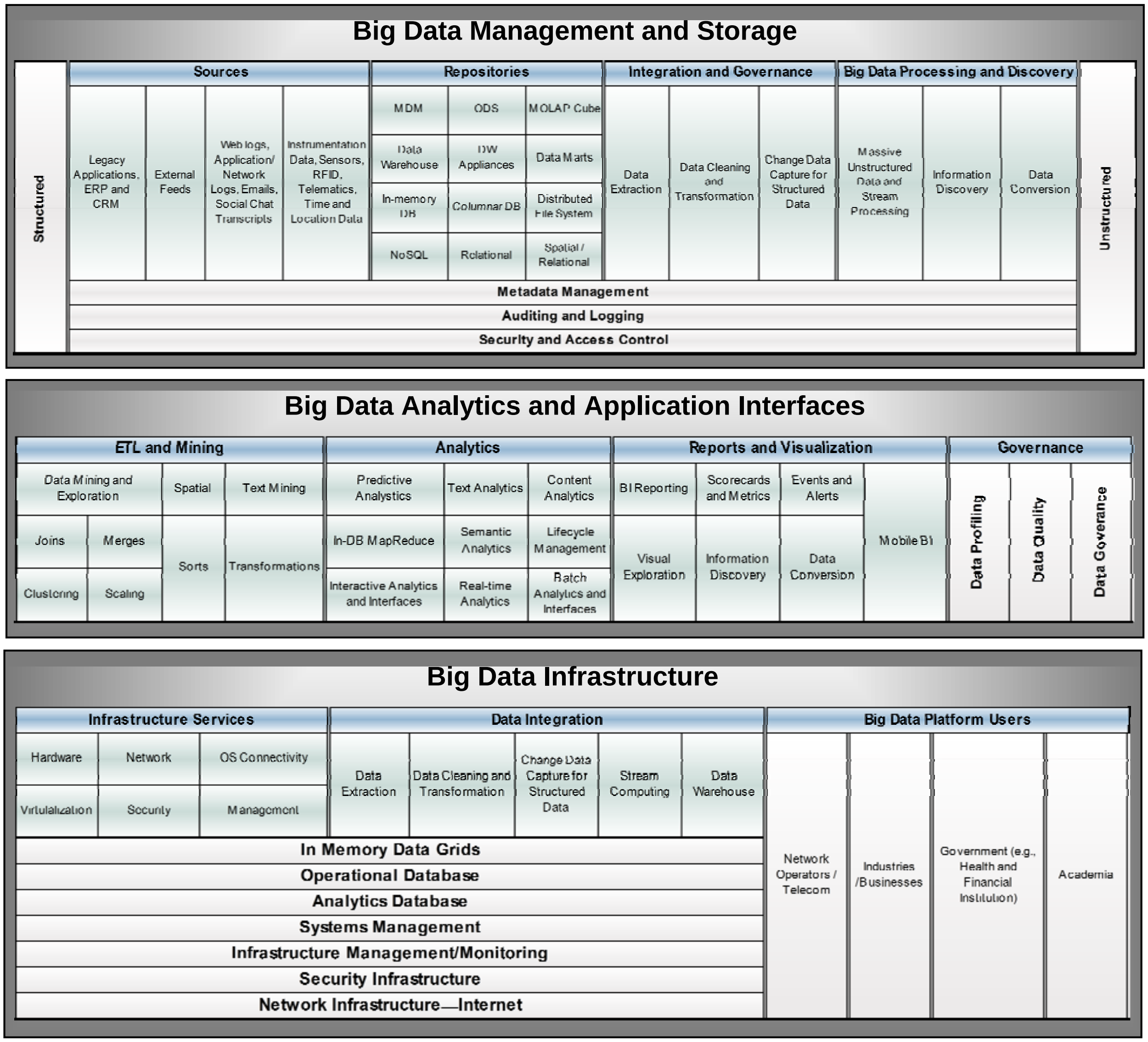



***Figure 18: Big Data Reference Architecture***

# Appendix A: Acronyms

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

# Appendix B: References

## DOCUMENT REFERENCES


[1] The White House Office of Science and Technology Policy, "Big Data is a Big Deal," OSTP Blog, accessed February 21, 2014, http://www.whitehouse.gov/blog/2012/03/29/big-data-big-deal.

[2] Department of Defense Chief Information Officer, "Reference Architecture Description," *Department of Defense*, June 2010, http://dodcio.defense.gov/Portals/0/Documents/DIEA/Ref_Archi_Description_Final_v1_18Jun10.pdf.

[3] Gartner Press Release, "Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data," *Gartner, Inc.*, June 27, 2011, http://www.gartner.com/it/page.jsp?id=1731916.

[4] Lynn Greiner, "What is Data Analysis and Data Mining?" *Data Base Trends and Applications*, January 7, 2011, http://www.dbta.com/Articles/Editorial/Trends-and-Applications/What-is-Data-Analysis-and-Data-Mining-73503.aspx.

[5] Yuri Demchenko, Canh Ngo, and Peter Membrey, "Architecture Framework and Components for the Big Data Ecosystem Draft Version 0.2," *System and Network Engineering Group, UVA*, September 12, 2013, http://www.uazone.org/demch/worksinprogress/sne-2013-02-techreport-bdaf-draft02.pdf.

[6] Y. Demchenko, M. Makkes, R. Strijkers, C. Ngo, and C. de Laat, "Intercloud Architecture Framework for Heterogeneous Multi-Provider Cloud based Infrastructure Services Provisioning," *The International Journal of Next-Generation Computing (IJNGC)*, Volume 4, Issue 2 (2013).

[7] Y. Demchenko, M. Makkes, R. Strijkers, C. Ngo, and C. de Laat, "Intercloud Architecture Framework for Heterogeneous Multi-Provider Cloud based Infrastructure Services Provisioning," *The International Journal of Next-Generation Computing (IJNGC)*, Volume 4, Issue 2 (2013).

[8] B. Khasnabish, J. Chu, S. Ma, N. So, P. Unbehagen, M. Morrow, M. Hasan, Y. Demchenko, and Y. Meng, "Cloud Reference Framework – Draft," *Internet Engineering Task Force*, July 2, 2013, http://www.ietf.org/id/draft-khasnabish-cloud-reference-framework-05.txt.

[9] Marc Makkes, Canh Ngo, Yuri Demchenko, Rudolf Strijkers, Robert Meijer, and Cees de Laat, "Defining Intercloud Federation Framework for Multi-provider Cloud Services Integration," *The Fourth International Conference on Cloud Computing, GRIDs, and Virtualization (CLOUD COMPUTING 2013)*, May 27 - June 1, 2013,Valencia, Spain.