# Big Data Working Group Overview

Brainstorming  Presentation showing how subgroup deliverables might be tied together in an overview for September meeting.  Purely for illustrative purposes.

# Outline

- Big Data Definition
- Big Data Working Group Process
- Reference Architecture as a Focal Point
- Reference Architecture
- Taxonomy mapped to Reference Architecture
- Actors Mapped to Reference Architecture
- Security
- Technology Improvements mapped to Reference Architecture
- Building Blocks for Use Case with Requirements, Gap Analysis, and Best Practice Recommendations mapped to Reference Architecture
- Conclusions

# Big Data Definition

- **Big Data Definition -** *“Big Data refers to the new technologies and applications introduced to handle increasing Volume, Velocity and Variety of data while enhancing data utilization capabilities such as Variability, Veracity, and Value.”*

The large Volume of data available forces horizontal scalability of storage and processing and has implications for all the other attributes.

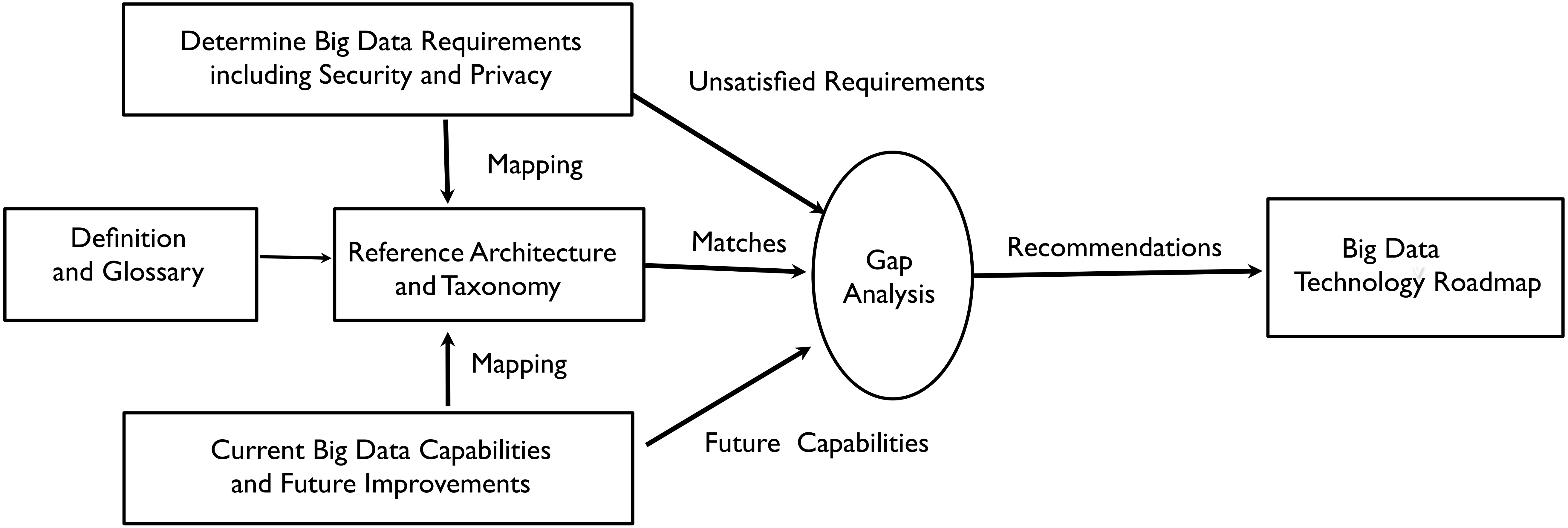
The increasing Velocity of data ingestion and change implies the need for stream processing, filtering, and processing optimizations.

The Variety of data types  (e.g. multimedia) being generated requires the use of non-traditional data stores and processing.

# Some changes driven by the V-attributes

- *Volume* - Driving the requirement for robust horizontal scalability of storage and processing
- *Velocity* - Driving optimization such as parallel stream processing and performance optimized databases
- *Variety* - Driving move to non-relational data models (e.g. key-value)
- *Variability* - Driving need for adaptive infrastructure
- *Value* - Driving need for new querying and analytics tools
- *Veracity* - Driving need for ensuring trust in the accuracy, relevance, and security of Big Data storage and processing

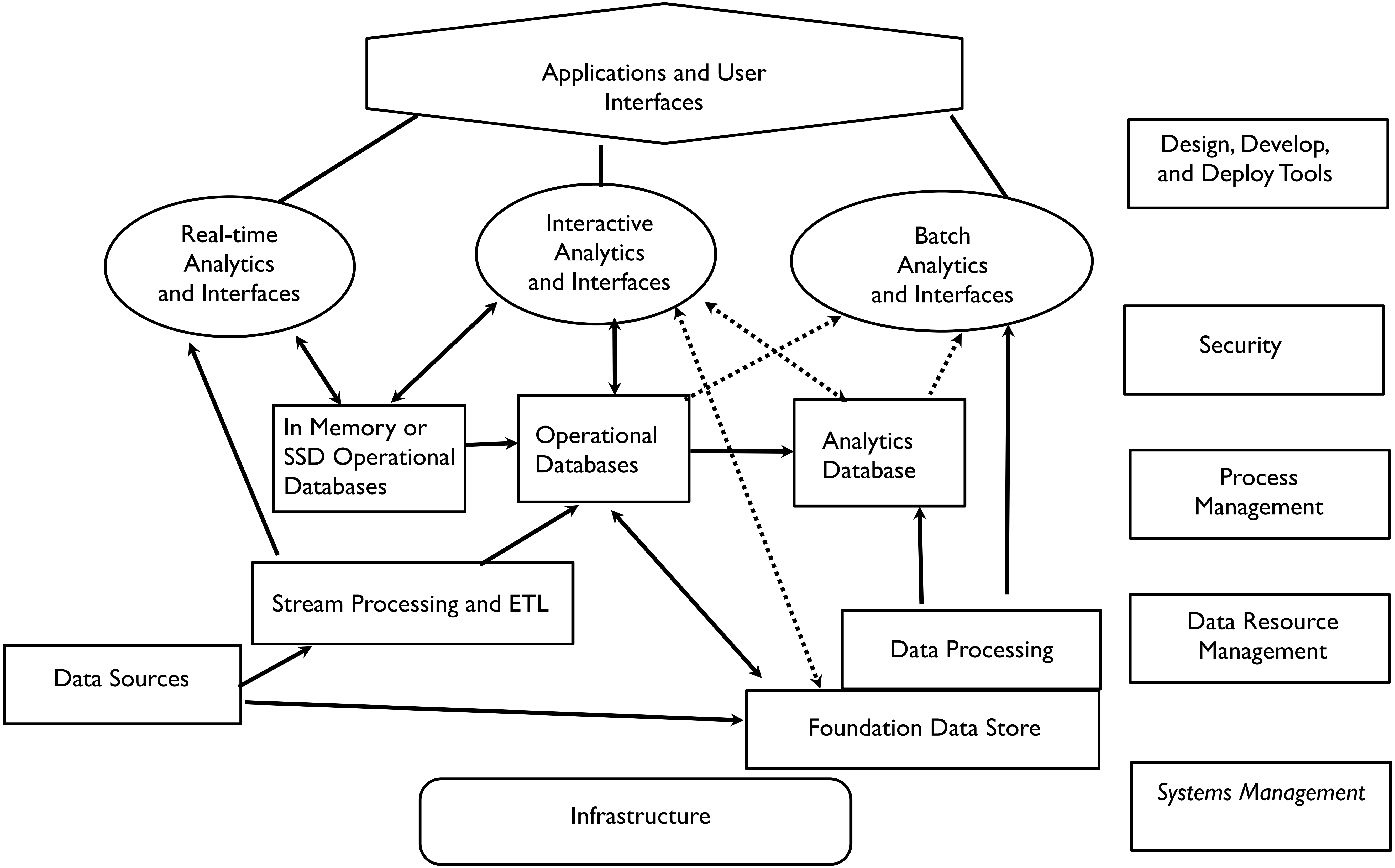
# Big Data Working Group Process


Determine Big Data Requirements including Security and Privacy
Unsatisfied Requirements
Mapping
Definition and Glossary
Reference Architecture and Taxonomy
Matches
Gap Analysis
Recommendations
Big Data Technology Roadmap
Mapping
Current Big Data Capabilities and Future Improvements
Future Capabilities

# Reference Architecture as a Focal Point

- The Big Data Reference Architecture will be the focal point of the Process
- A Taxonomy are defined on the components of the Reference Architecture
- Use Case actors are mapped to the Reference Architecture components
- Current and future technology is mapped to Reference Architecture components
- Requirements based on Use Cases are also mapped to these components
- Comparing current technology with requirements enables gap analysis
- Matching gaps with future technology provides roadmap recommendations

# Big Data Reference Architecture


Applications and User Interfaces
Real-time Analytics and Interfaces
Interactive Analytics and Interfaces
Batch Analytics and Interfaces
Design, Develop, and Deploy Tools
Security
In Memory or SSD Operational Databases
Operational Databases
Analytics Database
Process Management
Stream Processing and ETL
Data Processing
Data Resource Management
Data Sources
Foundation Data Store
Infrastructure
*Systems Management*

# Taxonomy Extending the Reference Architecture

**1. Applications**

**2. Design, Develop, Deploy Tools**

**3. Security and Privacy**

**4. Analytics and Interfaces**

4.1 Complex Analytics

4.1.1 Real-time

4.1.1.1 Complex Event Processing

4.1.2 Interactive

4.1.3 Batch

4.1.3.1 Machine Learning

4.2 Interfaces

4.2.1Non-SQL

4.2.2 SQL

4.2.2.1 To Relational Databases

4.2.2.2 To Filesystem

4.2.2.3 To NoSQL Database

4.3 Visualization

4.4 Business Intelligence

**5. System and Process Management**

5.1 Systems Management

5.2 Process Management

**6. Data Processing within the Architecture Framework**

6.1 ETL

6.2 Data Serialization

**7. Data Resource Management**

7.1 Data Governance

7.2 Metadata Management

**8. Data Stores**

8.1 File Systems

8.2 Databases

8.2.1 Operational

8.2.1.1NoSQL Databases

8.2.1.1.1 Column-oriented

8.2.1.2.2 Document

8.2.1.1 .3Graphical

8.2.1.1.4 Key-Value

8.2.1.2 Relational Databases

8.2.1.2.1 NewSQL Databases

8.2.2 Analytic

8.2.2.1 EDW

8.2.3 In Memory or Solid State Drive (SSD) resident data bases

**9. IO External to Architecture Framework and Stream Processing -**

**10. Infrastructure**

10.1 Appliances

10.2 Internal Server Farm

10.3 Data Grids and Fabrics

10.4 Cloud-based

# Actors mapped to Reference Architecture (in red)

From http://www.smartplanet.com/blog/bulletin/7-new-types-of-jobs-created-by-big-data/682 based on an article
by Tam Harbert of Computerworld http://www.computerworld.com/s/article/9231445/Big_data_big_jobs_?taxonomyId=221&pageNumber=2

1. **Data scientists:** This emerging role is taking the lead in processing raw data and determining what types of analysis would deliver the best results. Typical backgrounds,include math and statistics, as well as artificial intelligence and natural language processing. (Reference Architecture: Analytics)

2. **Data architects:** Organizations managing Big Data need professionals who will be able to build a data model, and plan out a roadmap of how and when various data sources and analytical tools will come online, and how they will all fit together. (Reference Architecture: Design, Develop, Deploy Tools)

3. **Data visualizers:** These days, a lot of decision-makers rely on information that is presented to them in a highly visual format — either on dashboards with colorful alerts and "dials," or in quick-to-understand charts and graphs. Organizations need professionals who can "harness the data and put it in context, in layman's language, exploring what the data means and how it will impact the company says Tam Harbert of Computerwold. (Reference Architecture: Applications)

4. **Data change agents:** Every forward-thinking organization needs "change agents" — usually an informal role — who can evangelize and marshal the necessary resources for new innovation and ways of doing business. They need to be good communicators, and a Six Sigma background — meaning they know how to apply statistics to improve quality on a continuous basis — also helps. (Not applicable to Reference Architecture)

5. **Data engineer/operators:** These are the people that make the Big Data infrastructure hum on a day-to-day basis. "They develop the architecture that helps analyze and supply data in the way the business needs, and make sure systems are performing smoothly" says Tam Harbert of Computerworld (Reference Architecture: Data Processing and Data Stores)

6. **Data stewards:** Essential to any analytics-driven organization, is the emerging role of data steward. Every bit and byte of data across the enterprise should be owned by someone — ideally, a line of business. Data stewards ensure that data sources are properly accounted for, and may also maintain a centralized repository as part of a Master Data Management approach, in which there is one "gold copy" of enterprise data to be referenced. (Reference Architecture: Data Resource Management)

7. **Data virtualization/cloud specialists:** Databases themselves are no longer as unique as they use to be. What matters now is the ability to build and maintain a virtualized data service layer that can draw data from any source and make it available across organizations in a consistent, easy-to-access manner. Sometimes, this is called "Database-as-a-Service." No matter what it's called, organizations need professionals that can also build and support these virtualized layers or clouds." (Reference Architecture: Infrastructure)

# Security and Privacy (Placeholder): CSA 10 Top Big Data Challenges

1. Secure computations in distributed programming frameworks
2. Security best practices for non-relational data stores
3. Secure data storage and transactions logs
4. End-point input validation/filtering
5. Real-time security monitoring
6. Scalable and composable privacy-preserving data mining and analytics
7. Cryptographically enforced data centric security
8. Granular access control
9. Granular audits
10. Data provenance

# Technology Improvements mapped to Reference Architecture

# Technology Improvements Sets 1-2

**1. Processing Performance Improvements**
**(Reference Architecture: Data Processing)**

Data in memory or stored on Solid State Drive (SSD)
http://www3.weforum.org/docs/GITR/2012/GITR_Chapter1.7_2012.pdf
http://www.datanami.com/datanami/2012-02-13/big_data_and_the_ssd_mystique.htm'

Enhancements to first generation Map-Reduce
http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html
http://incubator.apache.org/mesos/

Use of GPUs
http://www.networkworld.com/news/tech/2013/062413-hadoop-gpu-271194.html

**2. Application Development Improvements**
**(Reference Architecture: Development Tools)**

Big Data PaaS and data grids
http://searchsoa.techtarget.com/feature/Look-out-Big-Data-In-memory-data-grids-start-to-go-mainstream
http://aws.amazon.com/elasticmapreduce/

Visual design, development, and deploy tools
http://www.pentahobigdata.com/overview

Unified interfaces using data virtualization
http://www.compositesw.com/company/pages/composite-software-next-generation-data-virtualization-platform-composite-6/
http://www.marklogic.com/solutions/data-virtualization/

# Technology Improvements Sets 3-4

**3. Complex Analytics Improvements**
**(Reference Architecture: Analytics)**

Embedded analytics
http://www.slideshare.net/InsideAnalysis/embedded-analytics-the-next-megawave-of-innovation

Stream analytics, filtering, and complex event processing
http://www.sqlstream.com/

Integrated data ingestion, processing, storage, and analytics
www.teradata.com/products-and-services/unified-data-architecture/

4. **Interoperability Improvements**
**(Reference Architecture: integration across components)**

Data sharing among multiple Hadoop tools and external tools (e.g. using HCatalog)
http://hortonworks.com/hdp/hdp-hcatalog-metadata-services/

Queries across Hadoop and legacy databases (e.g. EDW)
http://hadapt.com/product/

Data exchanges and ETL among diverse data stores
http://sqoop.apache.org/
http://www.talend.com/products/data-integration

# Technology Improvements Sets 5-6

**5. Possible Alternative Deployment Improvements**
**(Reference Architecture: Infrastructure)**

Cloud
http://www.cloudstandardscustomercouncil.org/031813/agenda.htm

HPC clusters
http://insidehpc.com/2013/06/19/cray-launches-complete-lustre-storage-solution-across-hpc-and-big-data-computing-markets/

Appliances
http://nosql.mypopescu.com/post/15729871938/comparing-hadoop-appliances-oracles-big-data

**6. Applications**
**(Reference Architecture: Applications)**

Internet of Things
http://en.wikipedia.org/wiki/Internet_of_Things

Big Data for Vertical Applications (e.g. science, healthcare)
http://jameskaskade.com/?p=2708

Big Data Society Applications and Issues
www.parl.gc.ca/HousePublications/Publication.aspx?DocId=6094136&Language=E&Mode=1&Parl=41&Ses=1

# Technology Improvements Set 7

**7. Interface Improvement**s
(**Reference Architecture: Interfaces)**

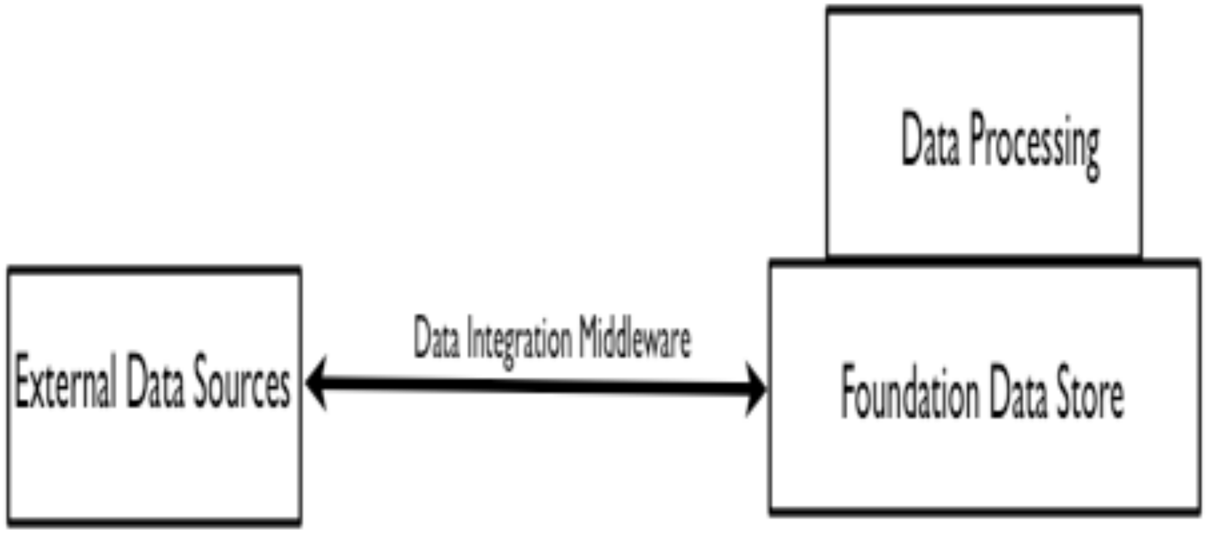
SQL interfaces to NoSQL databases
http://qconsf.com/sf2012/dl/qcon-sanfran-2012/slides/MaryHolstege_and_StephenBuxton_TheDesignOfASQLInterfaceForANoSQLDatabase.pdf

Performance optimizations for querying (e.g. columnar storage)
http://searchdatamanagement.techtarget.com/definition/columnar-database

Querying and analytics interfaces for end-user
http://www.tableausoftware.com/

# Building Blocks for Use Cases with Requirements, Gap Analysis, and Best Practice Recommendations mapped to subsets of the Reference Architecture

# 1. Data input and output to Big Data File System (ETL, ELT)



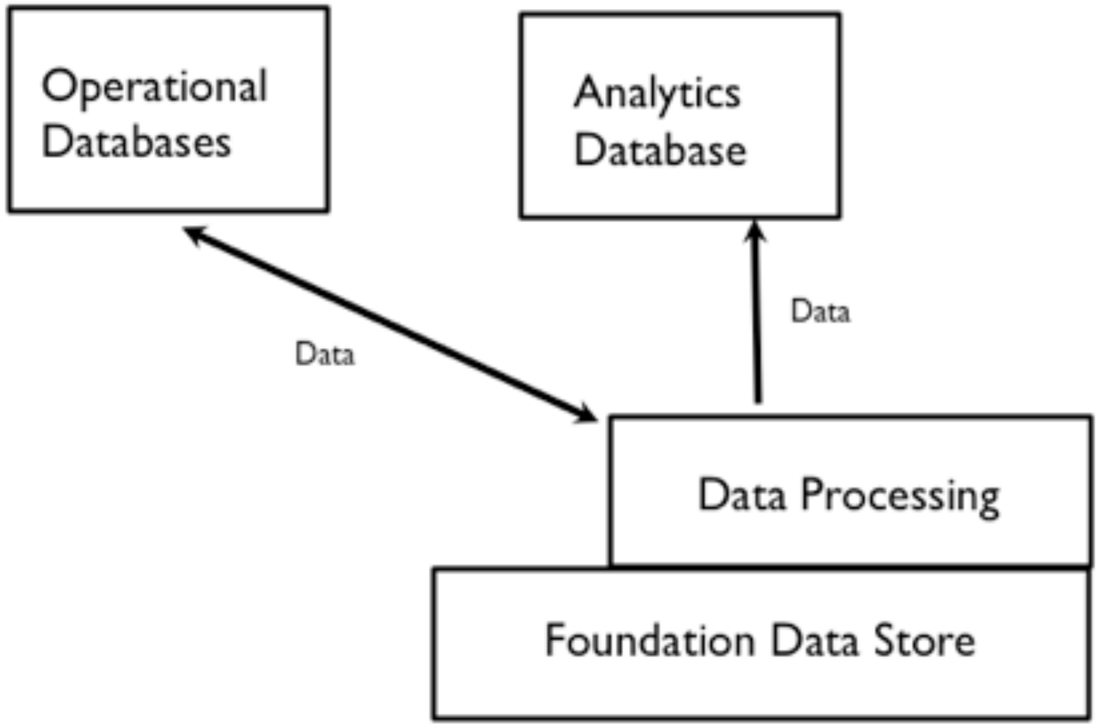
***Description:*** *The Foundation Data Store can be used as a repository for very large amounts of data (structured, unstructured, semi-structured). This data can be imported and exported to external data sources using data integration middleware.*

***Possible Requirements:*** *The data integration middleware should be able to do high performance extraction, transformation and load operations for diverse data models and formats.*

***Gap Analysis:*** *The technology for fast ETL to external data sources (e.g Apache Flume, Apache Sqoop) is available for most current data flows. There could be problems in the future as the size of data flows increases (e.g. LHC). This may require some filtering or summation to avoid overloading storage and processing capabilities*

***Suggested Best Practices:*** *Use packages that support data integration. Be aware of the possibilities for Extract-Load-Transform (ELT) where transformations can be done using data processing software after the raw data has been loaded into the data store e.g, Map-Reduce processing on top of HDFS.*

# 2. Data exported to Databases from Big Data **File**



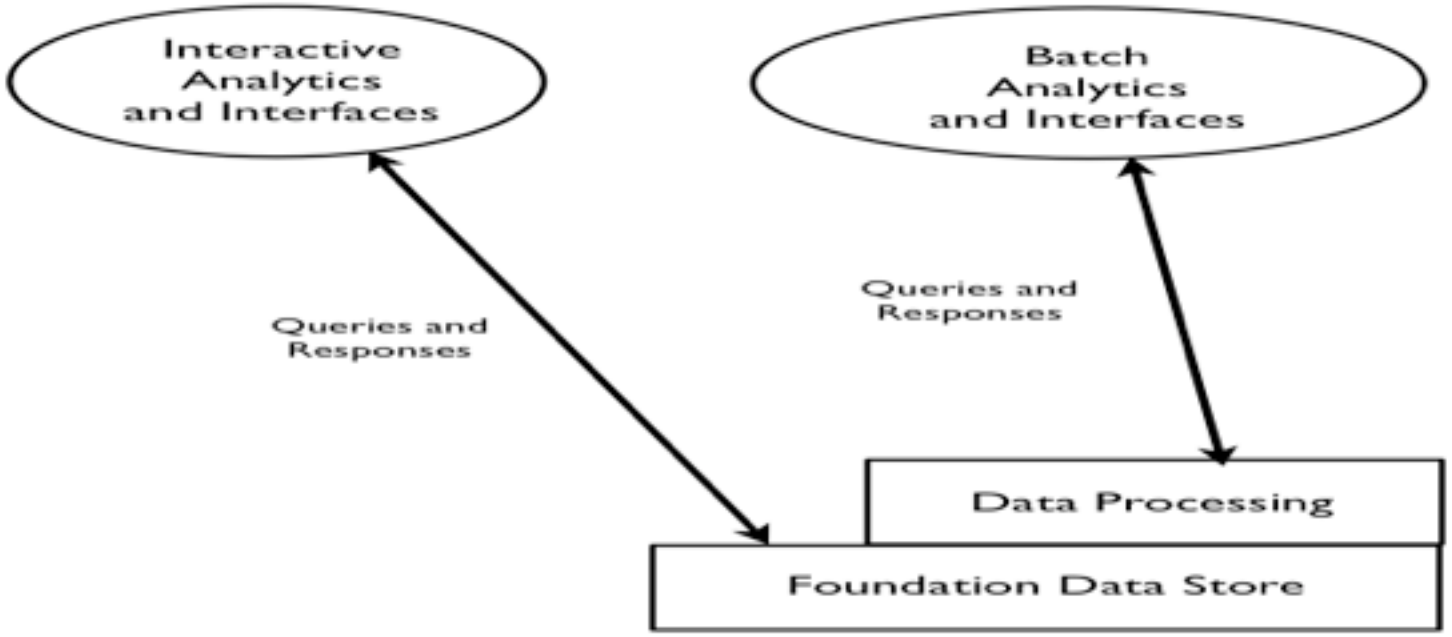
***Description:*** *A data processing system can extract, transform, and transmit data to operational and analytic databases.*

***Possible Requirements:*** *For good through-put performance on very large data sets, the data processing system will require multi-stage parallel processing*

***Gap Analysis:*** *Technology for ETL is available (e.g. Apache Sqoop for relational databases, MapReduce processing of files). However if high performance multiple passes through the data are necessary, it will be necessary to avoid rewriting intermediate results to files as is done by the original implementations of MapReduce.*

***Suggested Best Practices:*** *Consider using data processing that does not need to write intermediate results to files e.g.*

# 3. Big Data File Systems as a Data Resource for Queries



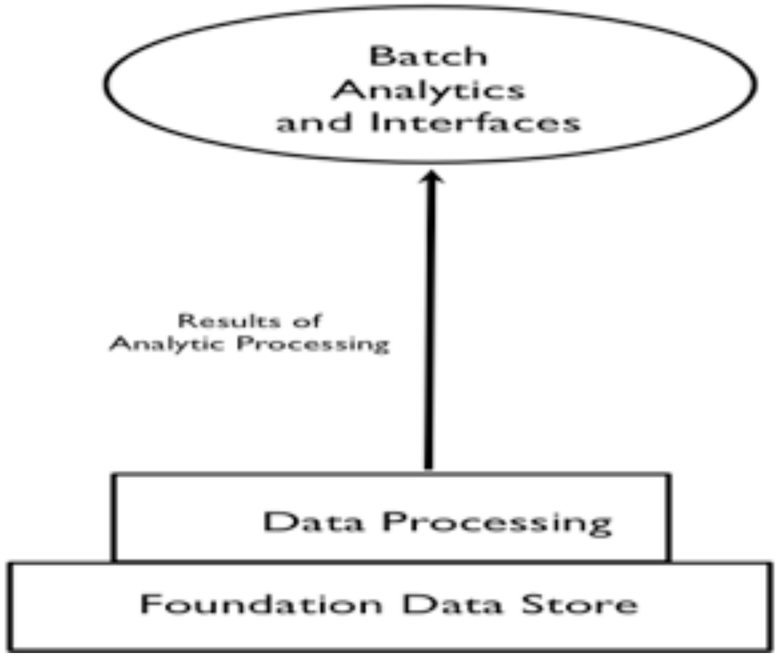
***Description:*** *The foundation data store can be queried through interfaces using batch data processing or direct foundation store access.*

***Possible Requirements:*** *The interfaces should provide good throughput performance for batch queries and low latency performance for direct interactive queries.*

***Gap Analysis:*** *Optimizations will be necessary in the internal format for file storage to provide high performance (e.g. Hortonworks ORC files, Cloudera Parquet)*

***Suggested Best Practices:*** *If performance is required, use optimizations for file formats within the foundation data store. If multiple processing steps are required, data processing packages that retain intermediate values in memory.*

# 4. Batch Analytics on Files using Parallel Processing


Batch Analytics and Interfaces
Results of Analytic Processing
Data Processing
Foundation Data Store

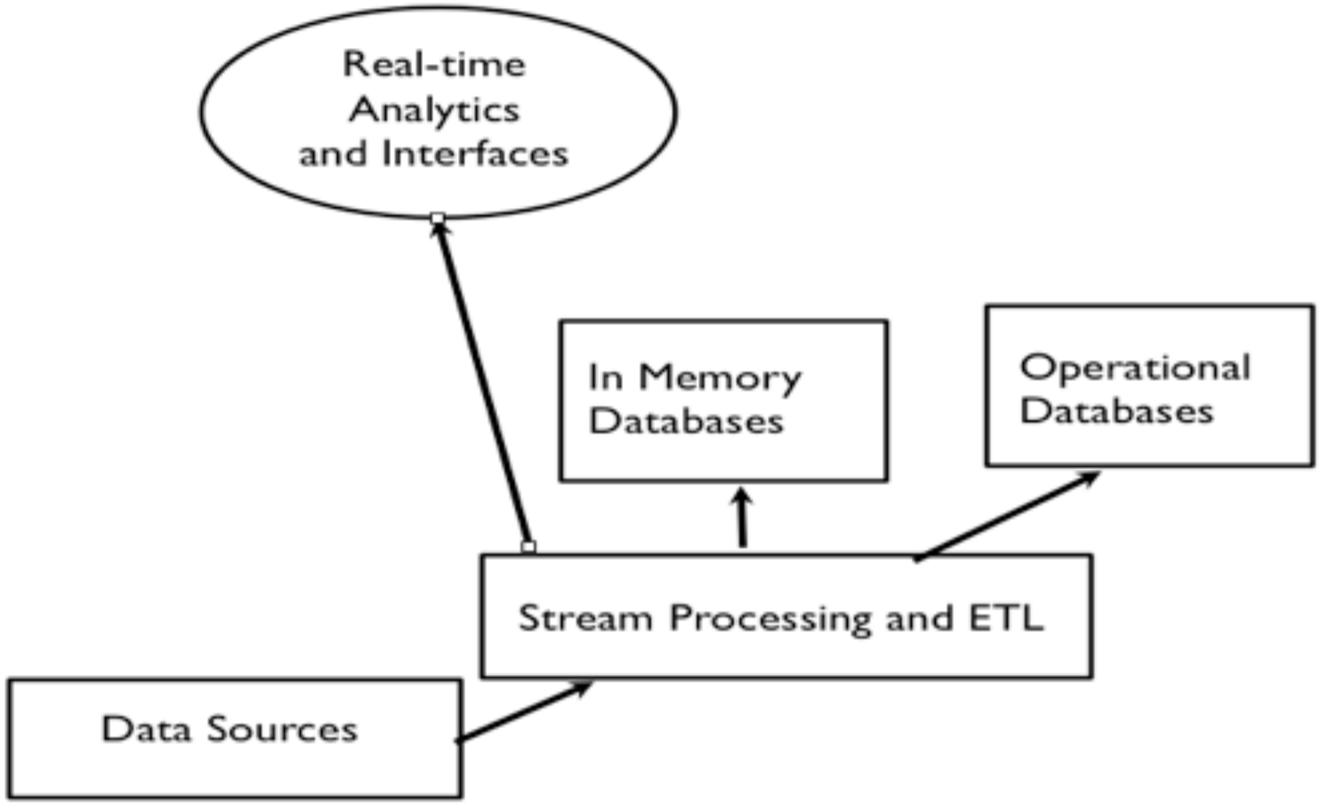
***Description:*** *A data processing system augmented by user defined functions can perform batch analytics on data sets stored in the foundation data store.*

***Possible Requirements:*** *High performance data processing is needed for efficient analytics.*

***Gap Analysis:*** *Analytics will often use multiple passes through the data. High performance will require the processing engine to avoid writing intermediate results to files as is done in the original version of MapReduce*

***Suggested Best Practices:*** *If possible, intermediate results of iterations should be kept in memory. Consider moving data to be analyzed into memory or an analytics optimized database.*

# 5. Stream Processing and ETL



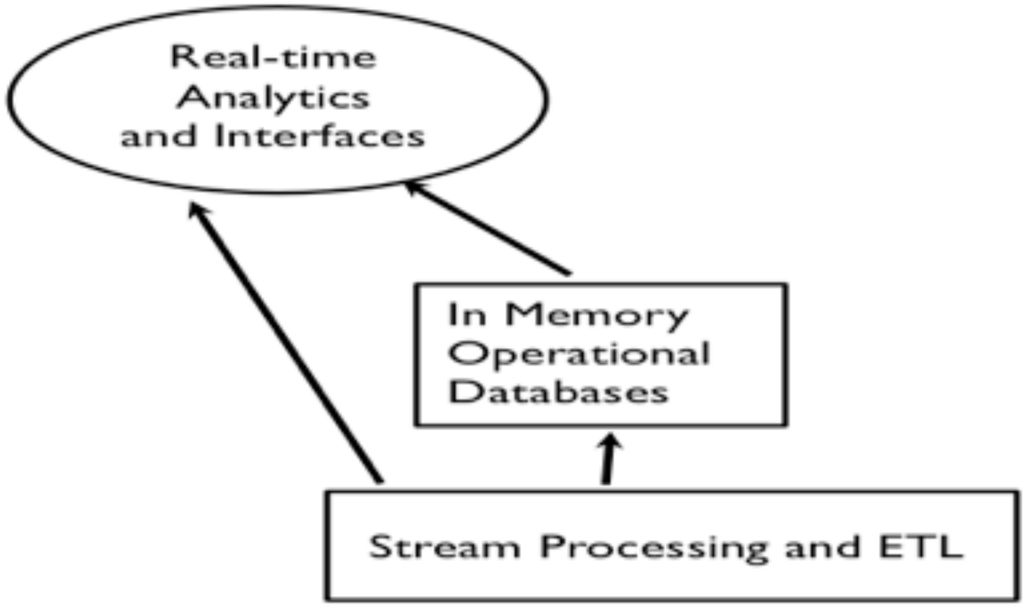
***Description:*** *Stream processing software can transform, process, and route data to databases and real time analytics*

***Possible Requirements:*** *The stream processing software should be capable of high performance processing of large high velocity data streams.*

***Gap Analysis:*** *Many stream processing solutions are available. In the future, complex analytics will be necessary to enable stream process to perform accurate filtering and summation of very large data streams.*

***Suggested Best Practices:*** *Parallel processing is necessary for good performance on large data streams.*

# 6. Real Time Analytics (e.g. Complex Event Processing)


Real-time Analytics and Interfaces
In Memory Operational Databases
Stream Processing and ETL

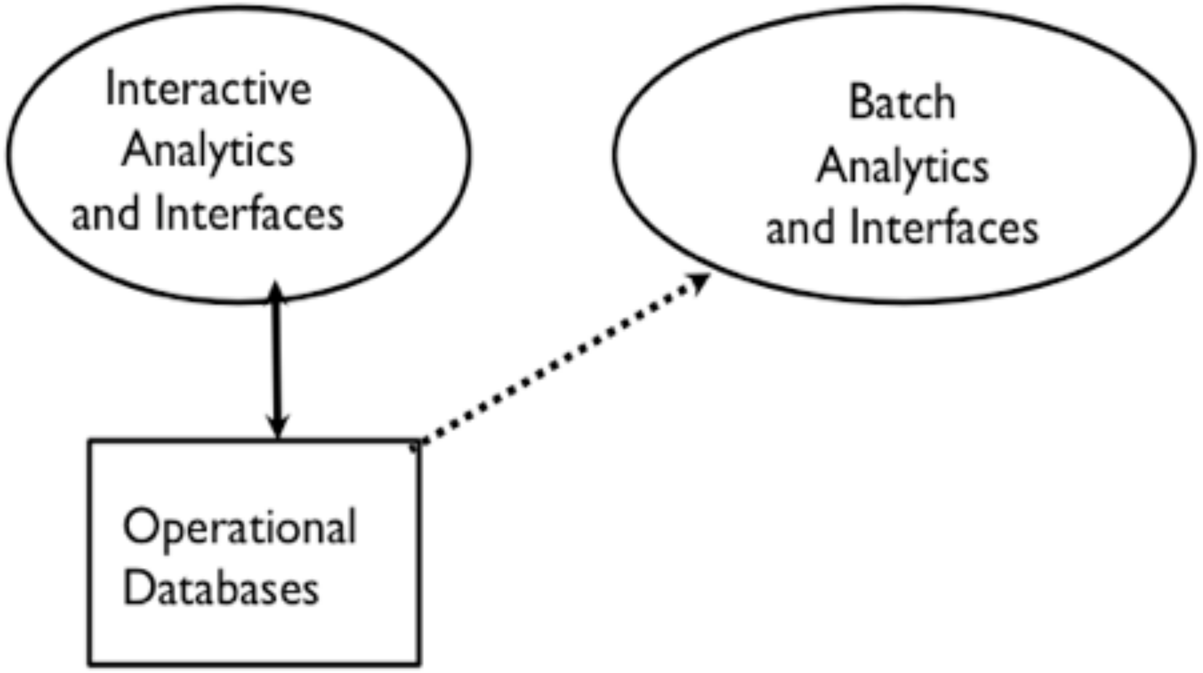
***Description:*** *Large high velocity data streams and notifications from in memory operational databases can be analyzed to detect pre-determined patterns, discover new relationships, and provide predictive analytics.*

***Possible Requirements:*** *Efficient algorithms for pattern matching and/or machine learning are necessary.*

***Gap Analysis:*** *There are many solutions available for complex event processing. It would be useful to have standards for describing event patterns to enable portability.*

***Suggested Best Practices****: Evaluate commercial packages to determine the best fit for your application.*

# 7. NoSQL (and NewSQL) DBs as operational


Interactive Analytics and Interfaces
Batch Analytics and Interfaces
Operational Databases

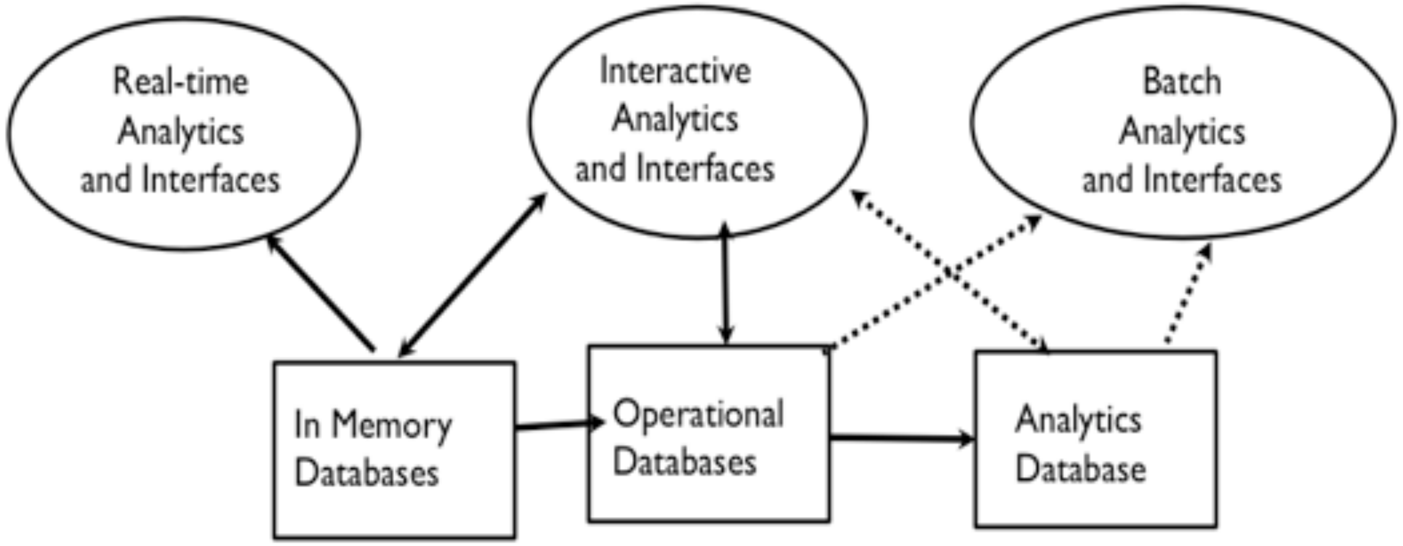
***Description:*** *Non-relational databases can be used for high performance for large data volumes (e.g. horizontally scaled). New SQL databases support horizontal scalability within the relational model.*

***Possible Requirements:*** *It is necessary to decide on the level of consistency vs. availability is needed since the CAP theorem demonstrates that both can not be achieved in horizontally scaled systems.*

***Gap Analysis:*** *The first generation of horizontal scaled databases emphasized availability over consistency. The current trend seems to be toward increasing the role of consistency. In some cases (e.g. Apache Cassandra), it is possible to adjust*

***Suggested Best Practices:*** *Horizontally scalable databases are experiencing rapid advances in performance and functionality. Choices should be based on application requirements and evaluation testing. Be very careful about choosing a*

# 8. NoSQL DBs for storing diverse data types


Real-time Analytics and Interfaces
Interactive Analytics and Interfaces
Batch Analytics and Interfaces
In Memory Databases
Operational Databases
Analytics Database

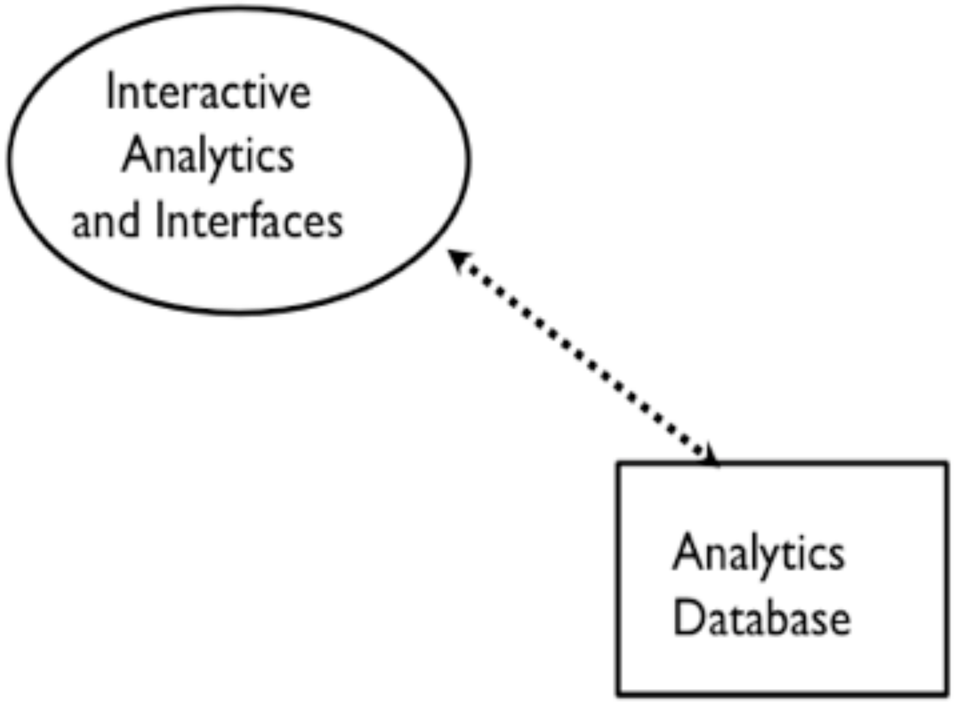
***Description:*** *Non-relational databases can store diverse data types (e.g. documents, graphs, heterogeneous rows) that can be retrieved by key or queries.*

***Possible Requirements:*** *The data types to be stored depend on application data usage requirements and query patterns.*

***Gap Analysis:*** *In general, the NoSQL databases are not tuned for analytic applications.*

***Suggested Best Practices:*** *There is a trade off when using non-relational databases. Usually some functionality is given up (e.g. joins, referential integrity) in exchange for some advantages (e.g. higher availability, better performance). Be sure that the trade-off meets application requirements.*

# 9. Databases optimized for complex ad hoc queries



:

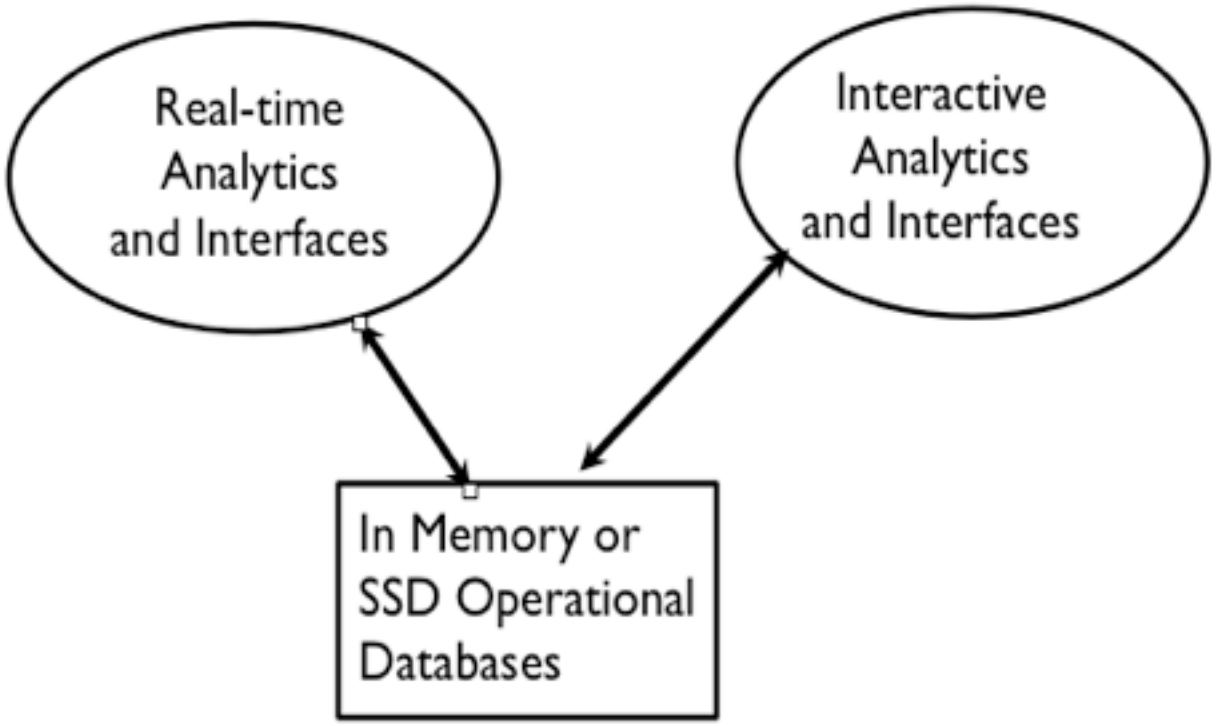
***Description:*** *Interactive ad hoc queries and analytics to specialized databases are key Big Data capabilities*

***Possible Requirements:*** *Analytic databases need high performance on complex queries which require optimizations such as columnar storage, in memory caches, and star schema data models.*

***Gap Analysis:*** *There is a need for embedded analytics and/or specialized databases for complex analytics applications.*

***Suggested Best Practices****: Use databases that have been optimized for analytics and/or support embedded analytics. It will often be necessary to move data from operational databases and/or foundation data stores using ETL tools.*

# 10. Databases optimized for rapid updates and retrieval


Real-time Analytics and Interfaces
Interactive Analytics and Interfaces
In Memory or SSD Operational Databases

***Description:*** *Very high performance operational databases are necessary for some large-scale applications.*

***Possible Requirements:*** *Very high performance will often require in memory databases and/or solid state drive (SSD) storage.*

***Gap Analysis:*** *Data retrieval from disk files is extremely slow compared in memory, cache, or SSD access. There will be increased need for these faster options as performance requirements increase.*

***Suggested Best Practices****: In the future, disk drives will be used for archiving or for non-performance oriented applications. Evaluate the use of data stores that can reside in memory, caches, or on SSDs.*

# Conclusions (Sample Placeholder)

- Big Data Technology is at an early stage of development
- The volume and velocity of data being processed is rapidly increasing
- The next few years will see major enhancements to technology capabilities in performance and functionality
- The eventual importance of interoperability standards and tools (e.g. UIMA, HCatalog) is still undecided
- It will be very valuable for governments to play a supportive and coordination role in Big Data technology initiatives
- Future collaboration will be necessary internationally as well as among US organizations (e.g. NIST, NITRD)