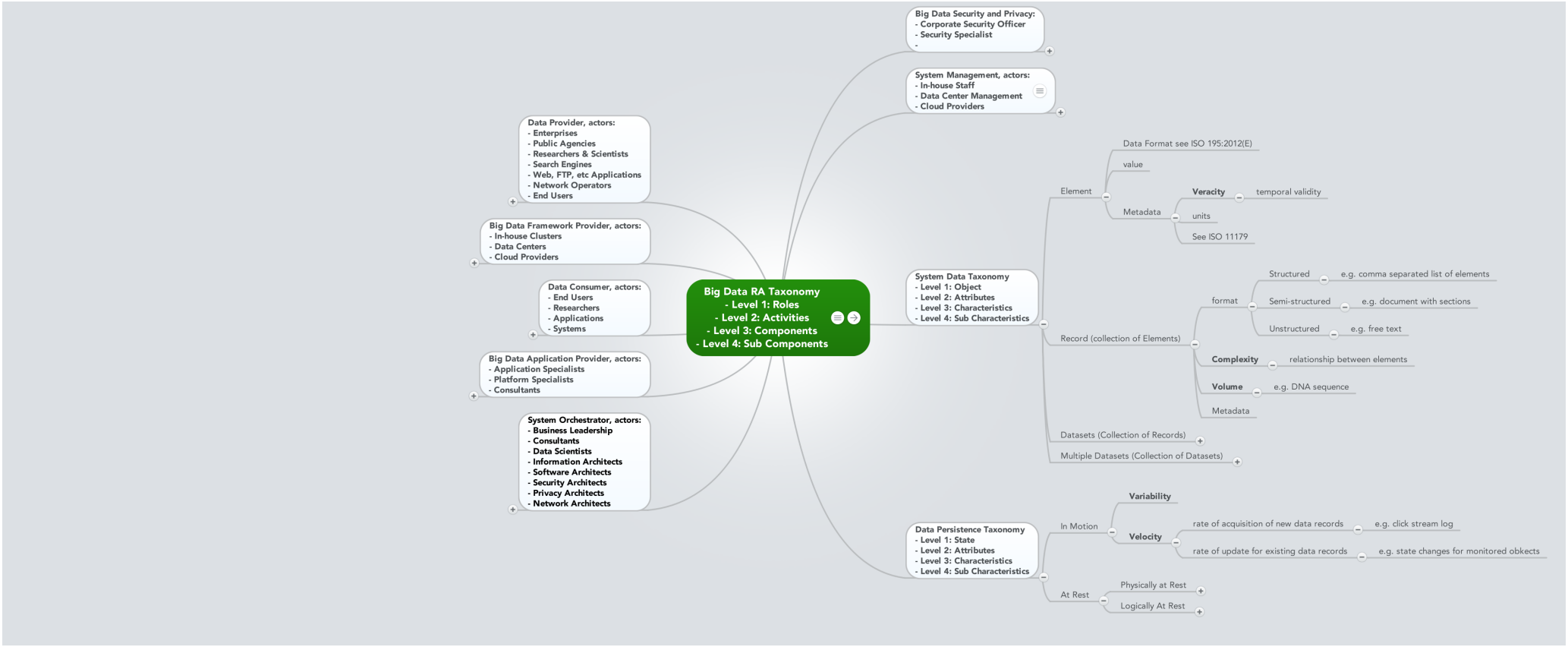
Big Data RA Taxonomy
- Level 1: Roles
- Level 2: Activities
- Level 3: Components
- Level 4: Sub Components
Data Provider, actors:
- Enterprises
- Public Agencies
- Researchers & Scientists
- Search Engines
- Web, FTP, etc Applications
- Network Operators
- End Users
Big Data Framework Provider, actors:
- In-house Clusters
- Data Centers
- Cloud Providers
Data Consumer, actors:
- End Users
- Researchers
- Applications
- Systems
Big Data Application Provider, actors:
- Application Specialists
- Platform Specialists
- Consultants
System Orchestrator, actors:
- Business Leadership
- Consultants
- Data Scientists
- Information Architects
- Software Architects
- Security Architects
- Privacy Architects
- Network Architects
Big Data Security and Privacy:
- Corporate Security Officer
- Security Specialist
-
System Management, actors:
- In-house Staff
- Data Center Management
- Cloud Providers
System Data Taxonomy
- Level 1: Object
- Level 2: Attributes
- Level 3: Characteristics
- Level 4: Sub Characteristics
Element
Data Format see ISO 195:2012(E)
value
Metadata
Veracity
temporal validity
units
See ISO 11179
Record (collection of Elements)
format
Structured
e.g. comma separated list of elements
Semi-structured
e.g. document with sections
Unstructured
e.g. free text
Complexity
relationship between elements
Volume
e.g. DNA sequence
Metadata
Datasets (Collection of Records)
Multiple Datasets (Collection of Datasets)
Data Persistence Taxonomy
- Level 1: State
- Level 2: Attributes
- Level 3: Characteristics
- Level 4: Sub Characteristics
In Motion
Variability
Velocity
rate of acquisition of new data records
e.g. click stream log
rate of update for existing data records
e.g. state changes for monitored obkects
At Rest
Physically at Rest
Logically At Rest

Big Data RA Taxonomy - Level 1: Roles - Level 2: Activities - Level 3: Components - Level 4: Sub Components

http://www.mindmeister.com/322462463#

Characteristics?
Processes?

Storage Implementation

The Roadmap document breaks this out into more granularity as:

Local Disks/Filesystems
HW/SW RAID
SAN
NAS
Distributed Filesystems
Distributed Object Stores

Block and Object stores are generally more logical implementations.

Also it should either be under cluster implementation or data services.

System Management, actors: - In-house Staff - Data Center Management - Cloud Providers

I really see this as a component of the orchestration aspect.

Queues

Can someone give me an example of a straming logical storage organization. By definition to me streaming is data in motion not at rest. I can store streamed data in any number of logical models. If that is the case we should add FIFO and LIFO buffers here.

Data Query Implementation

This is just way too limiting in nature and very focused on SQL style interfaces. What about query languages like the Lucene query syntax, or the MONGODB JSON query language.

This also ignores the HUGE variety of APIs used for bigtable and other DB interfaces (Hibernate's object store API for example).

Human-in-the-loop rapid analytics

Isn't this really Workflow support? I like that better than human in the loop as some of it automated for a user.

Operating System

To me Operating System is an attribute of the cluster implementation - not of the framework itself. Ideally, the framework shields the external actors from knowing the underlying OS and cluster implementation. This is certainly true for Hadoop.

Schema Information (metadata)

How is this different than information in a Data Registry, or in a semantic catalog tied to the ontology. I would combine the Data Domains and this area into one at this level.

Batch Processing Frameworks

In 2004, Phillip Colella working on DARPA's High Productivity Computing Systems (HPCS) program developed a list of algorithms for simulation in the physical sciences that became known as the "Seven Dwarfs"<!--[if supportFields]> CITATION Col04 \l 1033 <![endif]--> (Colella, 2004)<!--[if supportFields]><![endif]-->. More recently David Patterson and Katherine Yelick of the University of California – Berkley modified and extended this list to 13 shown in the table below based on the definition where "A dwarf is an algorithmic method that computes a pattern of computation and communication"<!--[if supportFields]> CITATION Pat \l 1033 <![endif]--> (Patterson & Yelick)<!--[if supportFields]><![endif]-->