NIST Big Data Public Working Group

Overview of NIST Big Data Interoperability Framework Volume 6

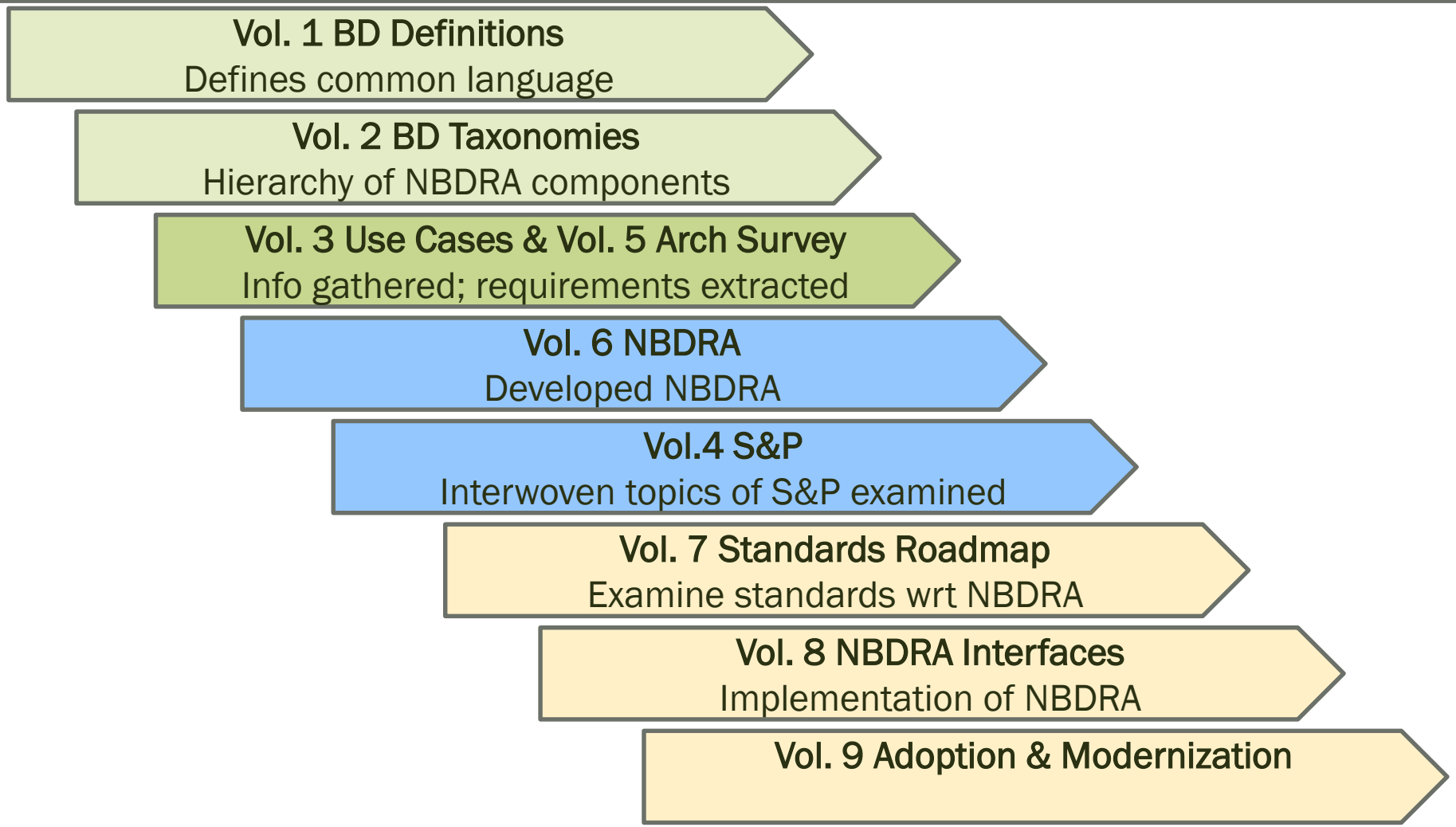
David Boyd
VP of Data Solutions
InCadence Strategic Solutions

NIST Campus
Gaithersburg, Maryland
June 1, 2017

Presentation Overview

- Volume Presentation Outline
- Volume 1, Definitions (Nancy Grady, SAIC)
- Volume 2, BD Taxonomies (Nancy Grady, SAIC)
- Volume 3, Use Cases and General Requirements (Geoffrey Fox, Indiana University)
- **Volume 6, Reference Architecture (David Boyd, InCadence Corp.)**
- Volume 4, Security and Privacy (Arnab Roy, Fujitsu; Mark Underwood, AVP, Strategic Initiatives, Controls and Countermeasures)
- Volume 8, Reference Architecture Interface (Gregor von Laszewski, Indiana University)
- Reference Architecture Software Implementation Environment and Demonstration (Gregor von Laszewski, Indiana University)
- Volume 7, Standards Roadmap (Russell Reinsch, Center for Government Interoperability)
- Volume 9, Adoption and Modernization (Russell Reinsch, Center for Government Interoperability)

NBDIF Volume Overview



Volume Presentation Outline

- For each volume
  - Scope of the volume
  - Brief recap of version 1
  - Highlights of version 2 accomplishments
  - Summary of version 2 areas needing contributions
  - Topics that could be considered for version 3

Volume 6, Reference Architecture

Document Scope

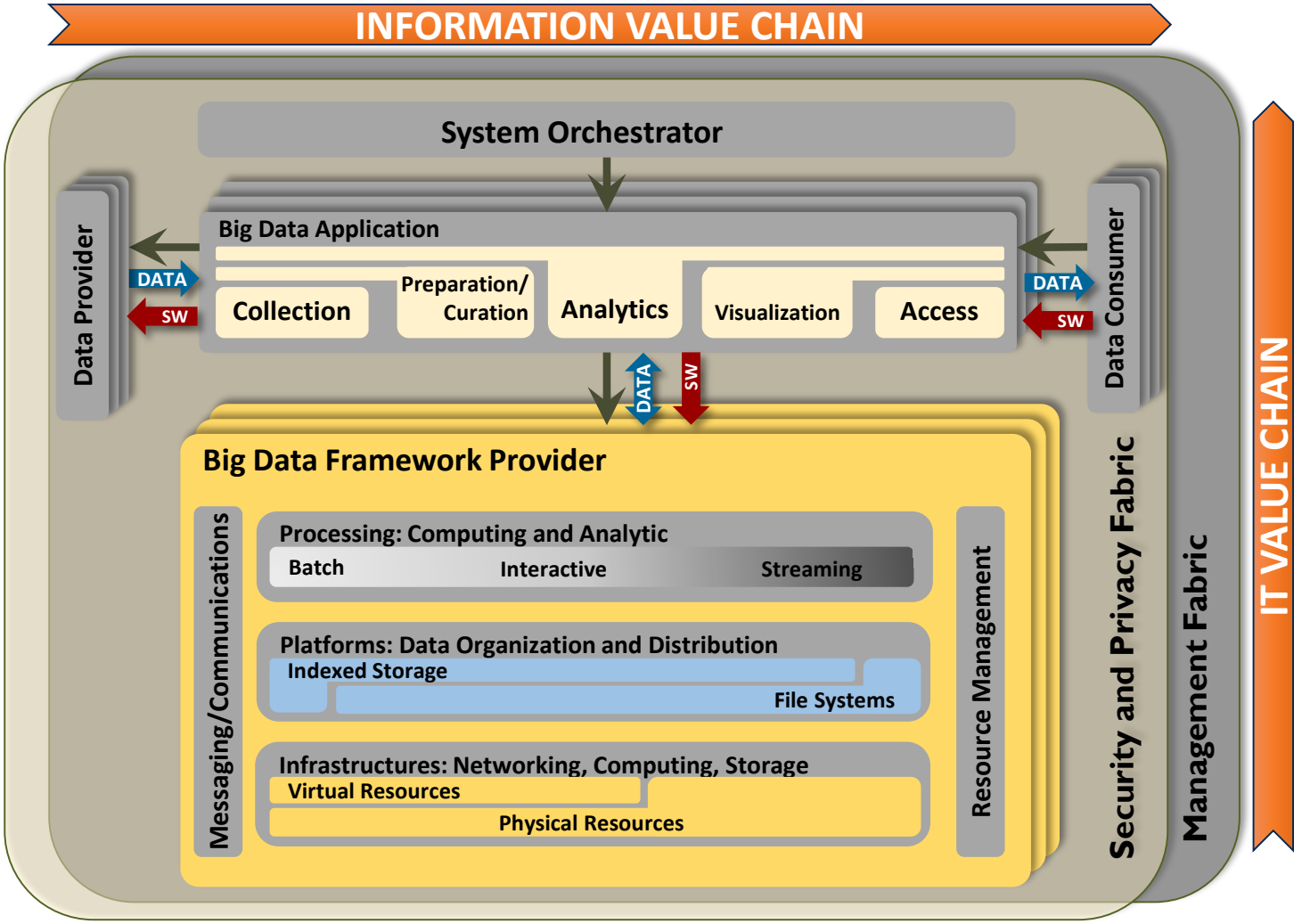
- Develop an open reference architecture for Big Data that achieves the following objectives:
  - Provides a common language
  - Encourages adherence to common standards, specifications, and patterns
  - Provides consistent technology implementation methods for similar problem sets
  - Illustrates various Big Data components, processes, and systems, in the context of a vendor- and technology-agnostic Big Data conceptual model
  - Provides a technical reference to discuss compare BD solutions
  - Facilitates analysis of candidate standards for interoperability, portability, reusability, and extendibility.

Volume 6, Reference Architecture

Version 1 Overview

- Collaborated with other subgroups to construct an understanding of Big Data requirements
- Developed a vendor- and technology-agnostic conceptual model with five components and two fabrics:
  - System Orchestrator
  - Data Provider
  - Big Data Application Provider
  - Big Data Framework Provider
  - Data Consumer
  - Security and Privacy Fabric
  - Management Fabric

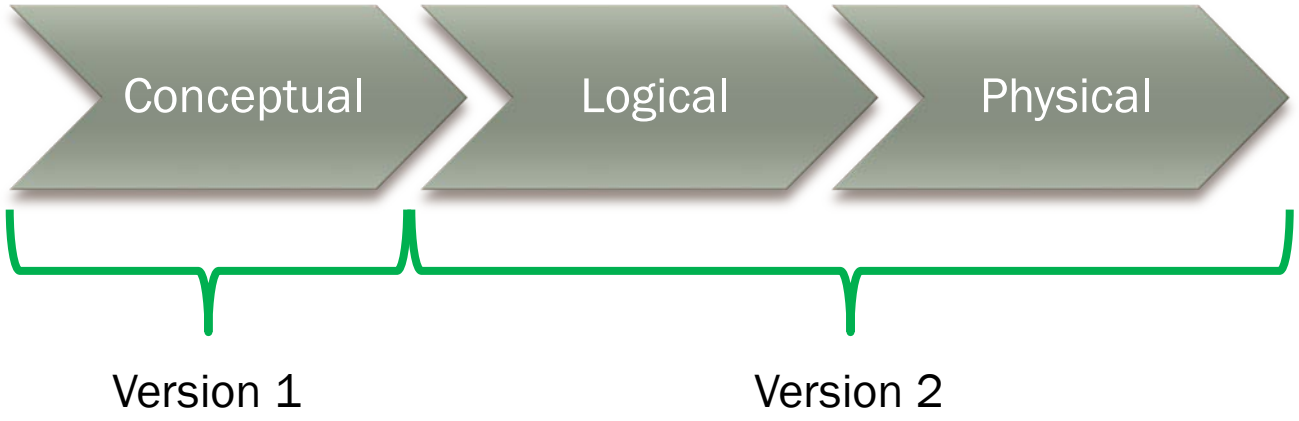
NIST Big Data Reference Architecture



Volume 6, Reference Architecture

Version 2 Accomplishment

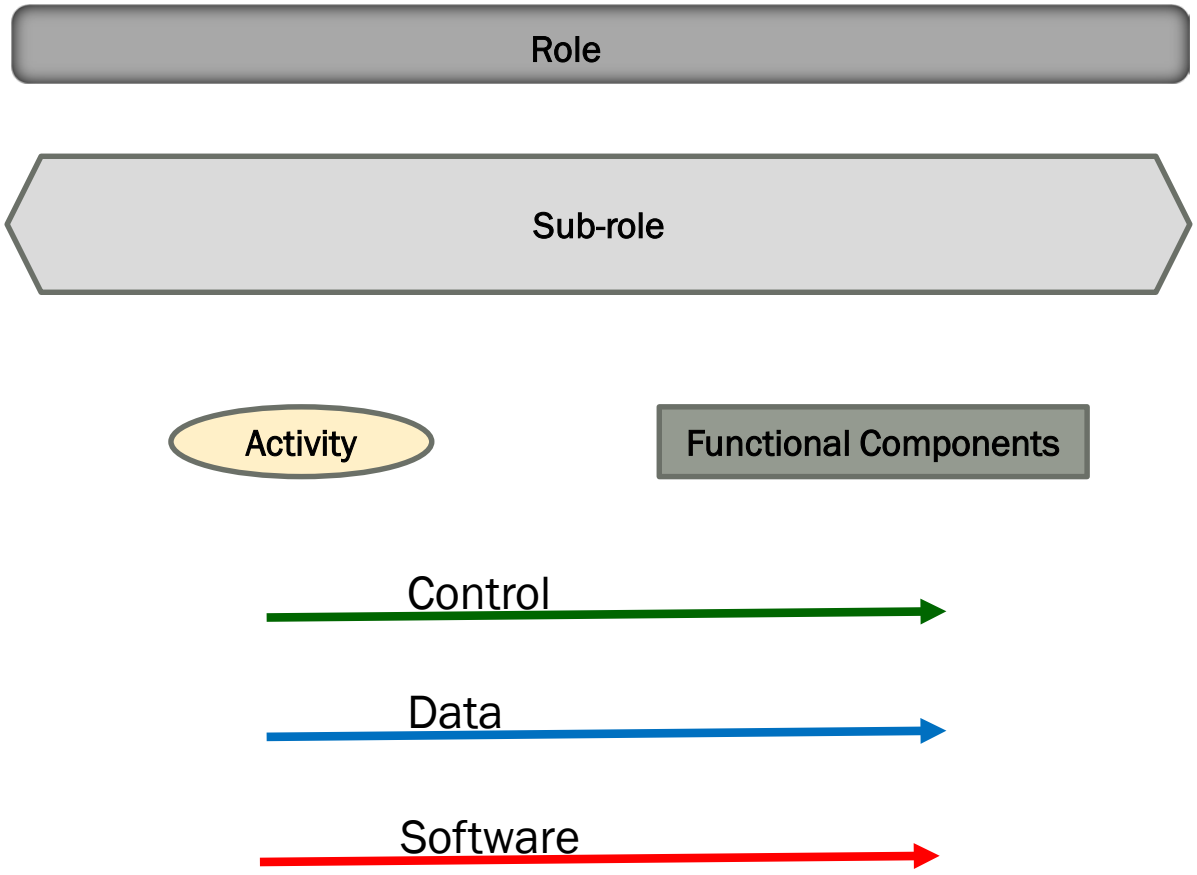
- Primary Goal: Develop a more rigourous set of architecture views
- Decided on two initial views:
  - Activity – What activities take place within the Roles and Sub-roles of the BDRA
  - Functional Component – What functional components are needed to accomplish the activities within the Roles and Sub-roles



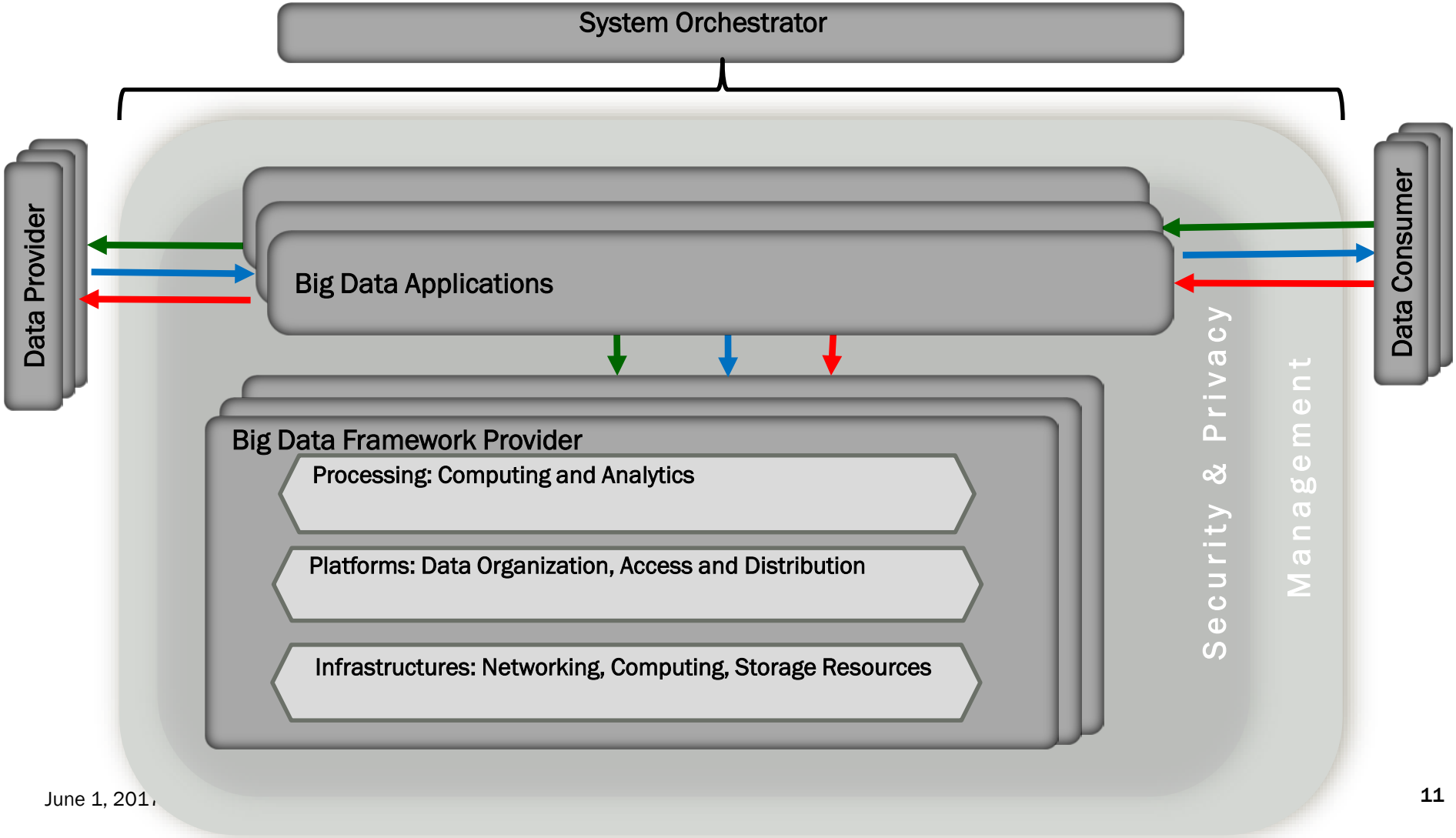
Developing Views - Definitions

- Role: A related set of functions performed by one or more actors.
- Fabric: A role which touches upon and supports multiple other roles
- Activity: A class of functions performed to full fill the needs of one or more roles.
  - Example: Data Collection is a class of activities through which a Big Data Application obtains data. Instances of such would be web crawling, FTP site, web services, database queries, etc.
- Functional Component: A class of physical items which support one or more activities within a role.
  - Example: Stream Processing Frameworks are a class of computing frameworks which implement processing of streaming data. Instances of such frameworks would include SPARK and STORM

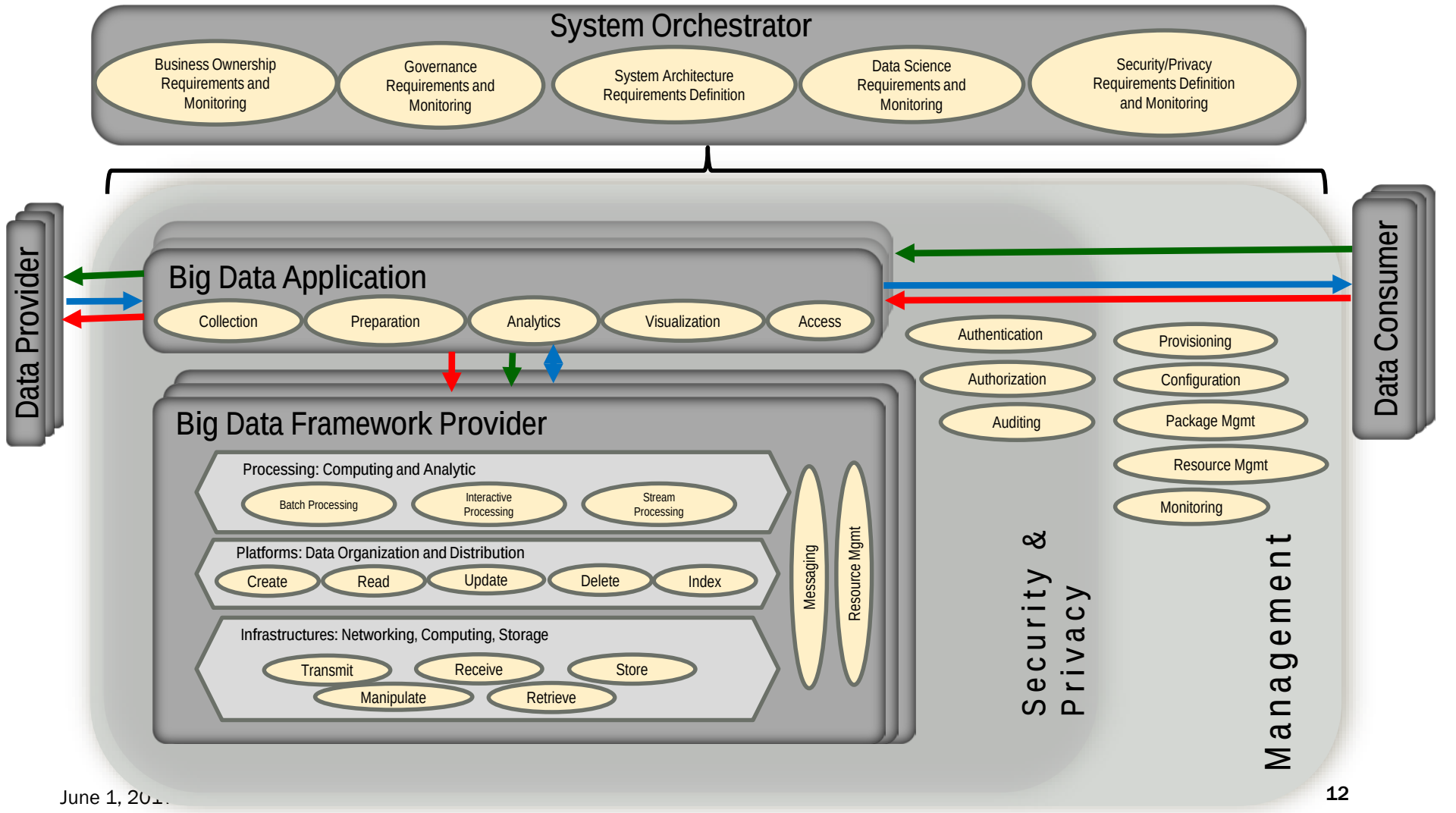
Developing Views - Notation



Developing Views – View Template



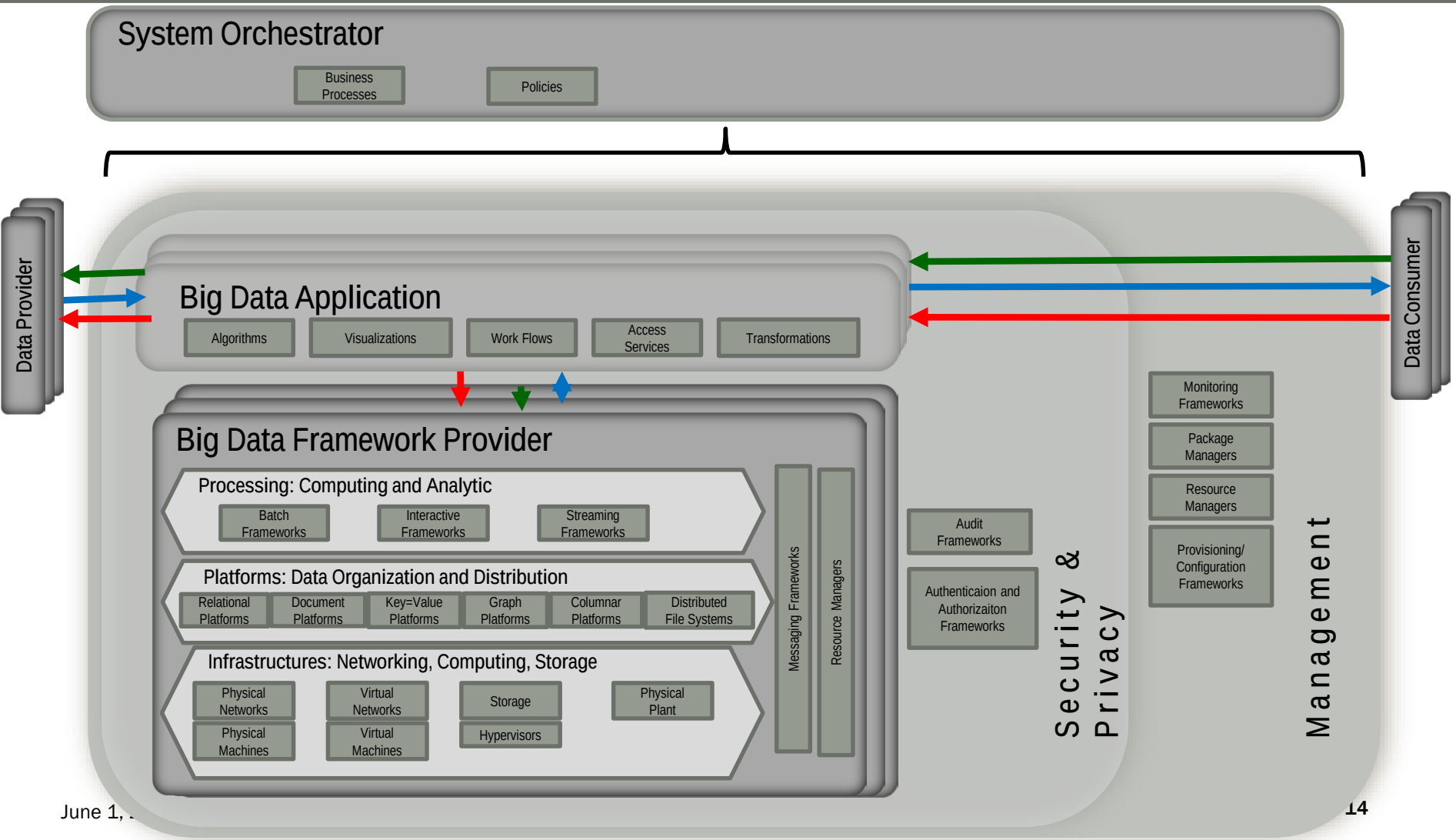
Activity View (Initial)



Activities Defined in View

- Reference Architecture: Top Level Activity Classes
  - Collection: In general, the collection activity of the Big Data Application handles the interface with the Data Provider. This may be a general service, such as a file server or web server configured by the System Orchestrator to accept or perform specific collections of data, or it may be an application-specific service designed to pull data or receive pushes of data from the Data Provider. Since this activity is receiving data at a minimum, it must store/buffer the received data until it is persisted through the Big Data Framework Provider. This persistence need not be to physical media but may simply be to an in-memory queue or other service provided by the processing frameworks of the Big Data Framework Provider. The collection activity is likely where the extraction portion of the Extract, Transform, Load (ETL)/Extract, Load, Transform (ELT) cycle is performed. At the initial collection stage, sets of data (e.g., data records) of similar structure are collected (and combined), resulting in uniform security, policy, and other considerations. Initial metadata is created (e.g., subjects with keys are identified) to facilitate subsequent aggregation or look-up methods.
- Reference Architecture Implementation: Specific Activities
  - Log Data Collection: Accept incoming data from log services as files. Store data on local file system for ingestion processing

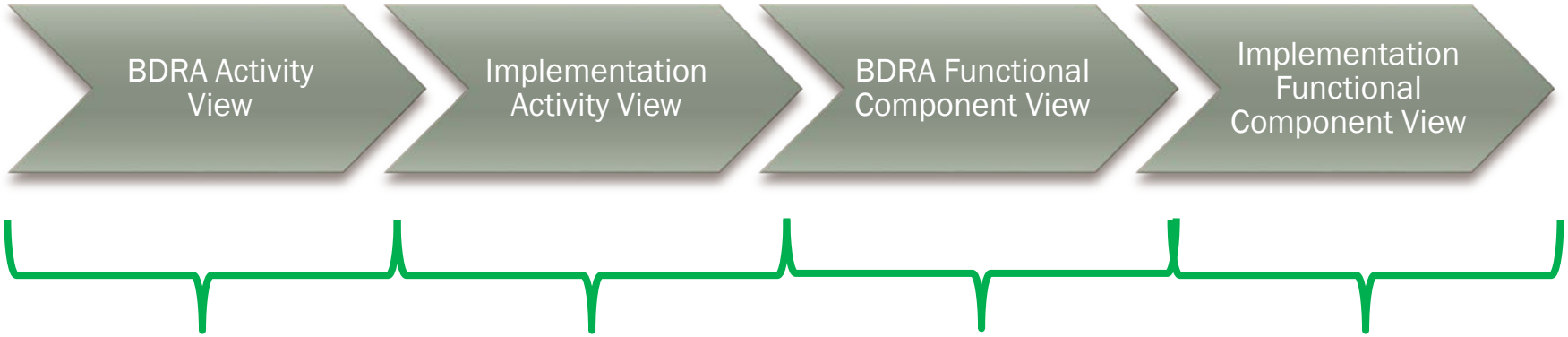
Functional Components View (Initial)



Functional Components Defined in View

- Reference Architecture: Top Level Component Classes
  - Graph Platforms : Graph databases typically store two types of objects nodes and relationships as show in Figure 7 below. Nodes represents objects in the problem domain that are being analyzed be they people, places, organizations, accounts, or other objects. Relationships describe those objects in the domain relate to each other. Relationships can be non-directional/bidirectional but are typically expressed as unidirectional in order to provide more richness and expressiveness to the relationships. Hence, between two people nodes where they are father and son, there would be two relationships. One “is father of” going from the father node to the son node, and the other from the son to the father of “is son of”. In addition, nodes and relationships can have properties or attributes. This is typically descriptive data about the element. For people it might be name, birthdate, or other descriptive quality. For locations it might be an address or geospatial coordinate. ….
- Reference Architecture Implementation: Specific Components
  - Neo4J configured as a causal cluster of 8 nodes (3 core, 5 read replicas) so that client applications enjoy read-your-own-writes semantics.

Applying the BDRA – Developing Implementation Views



Volume 6, Reference Architecture

Version 2 Opportunities for Contribution

- Activity View
  - Review classes of activities initially defined for completeness
  - As required develop zoomed in views
  - Develop text descriptions of each activity class
- Functional Component View
  - Review classes of functional components initially defined for completeness
  - As required develop zoomed in views
  - Develop text descriptions of each functional component class
- Align Functional Component view to Vol 8 Interfaces
- Define and describe a process for developing implementation activity and functional component views
- Add a discussion of the reference architecture in terms of a system of system (Section 4)

Possible Version 3 Topics

- Mapping of Activity and Functional Component classes to specific standards (may be better in Volume 7)
- Develop process for generating functional and system requirements from Activity and Functional Component views
- Refine and expand deployment models to cover containerization
- Link views and deployment models to Dev Ops standards (IEEE P2675)