UPDM and SysML for Systems Engineers

Graham Bleakley, IBM
graham.bleakley@uk.ibm.com

Rational. Software

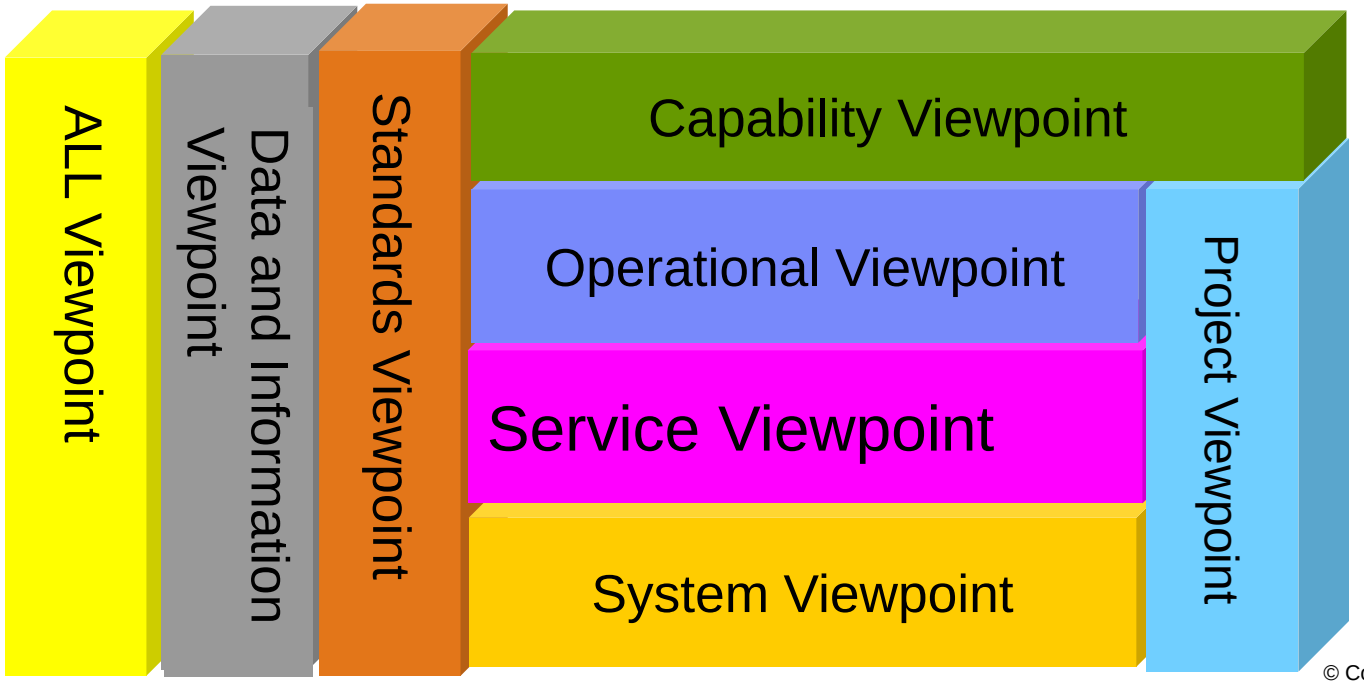
Agenda

This presentation will cover

- Describe how the views are related
- Identify the main viewpoints relevant for the Enterprise Architecture and Systems Engineering phases of development
- Show the patterns across the views and Viewpoints
- Discuss a generic workflow
- Points for discussion on UPDM/SysML modelling

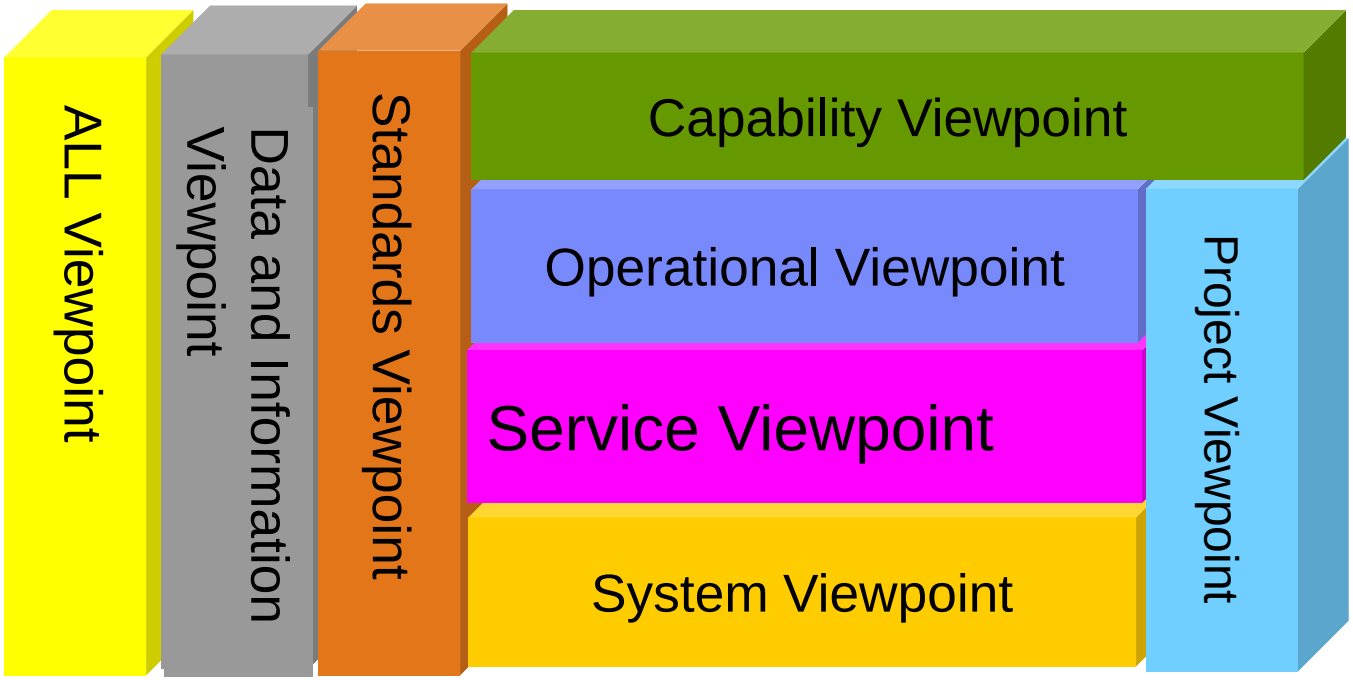
DoDAF 2.0 Viewpoints

- Capability Viewpoint:- Capture the capabilities that the enterprise is expected to realise and shows how they are deployed to organizations (prefix CV)
- Operational Viewpoint:- Contains the views required to describe the Operational and high level functional aspects of the architecture (prefix OV)
- Service Viewpoint:- Captures the views required to specify of the Services (i.e. interface, no implementation) required to support the Operational objectives of the architecture (prefix SvcV)



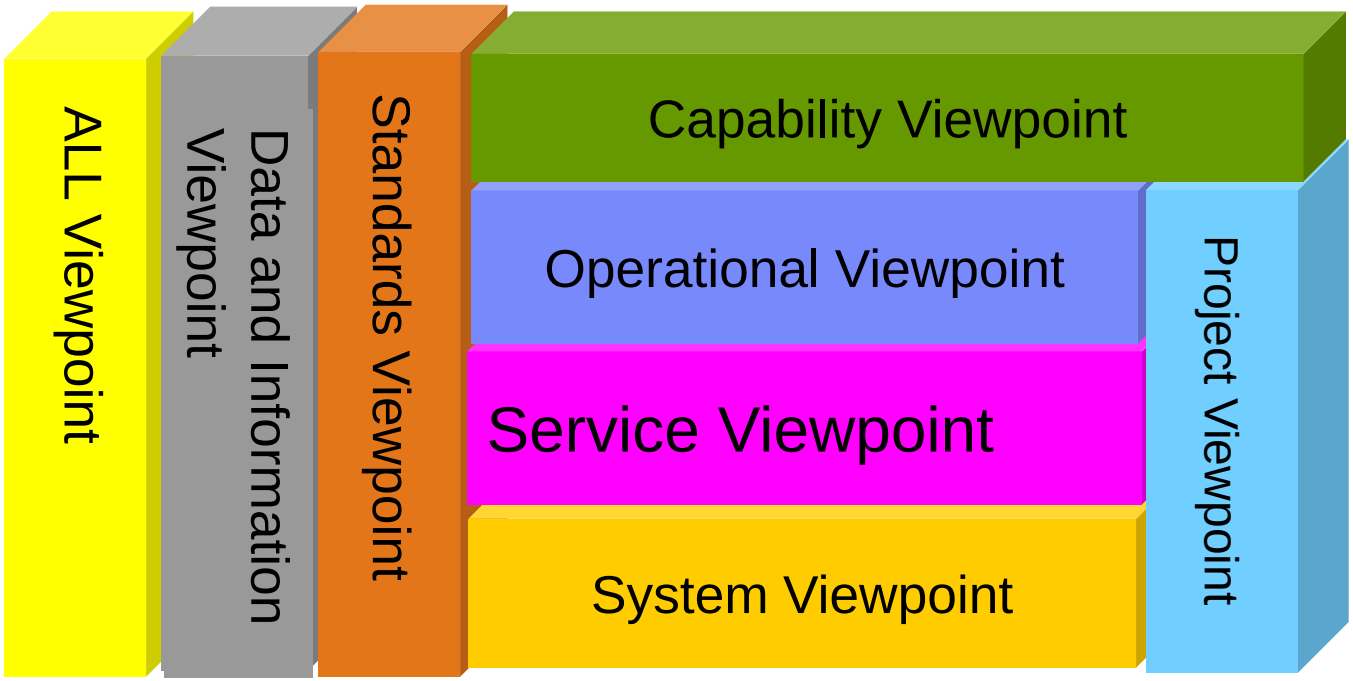
DoDAF 2.0 Viewpoints

- Systems Viewpoint:-Captures the specification of the Systems that are required to be implemented or that already exist that help achieve the operational objectives (prefix SV)
- Project Viewpoint:- Maps the enterprises to the projects and organisations that will realise the capabilities through the development of systems and services (prefix PV)



DoDAF 2.0 Viewpoints

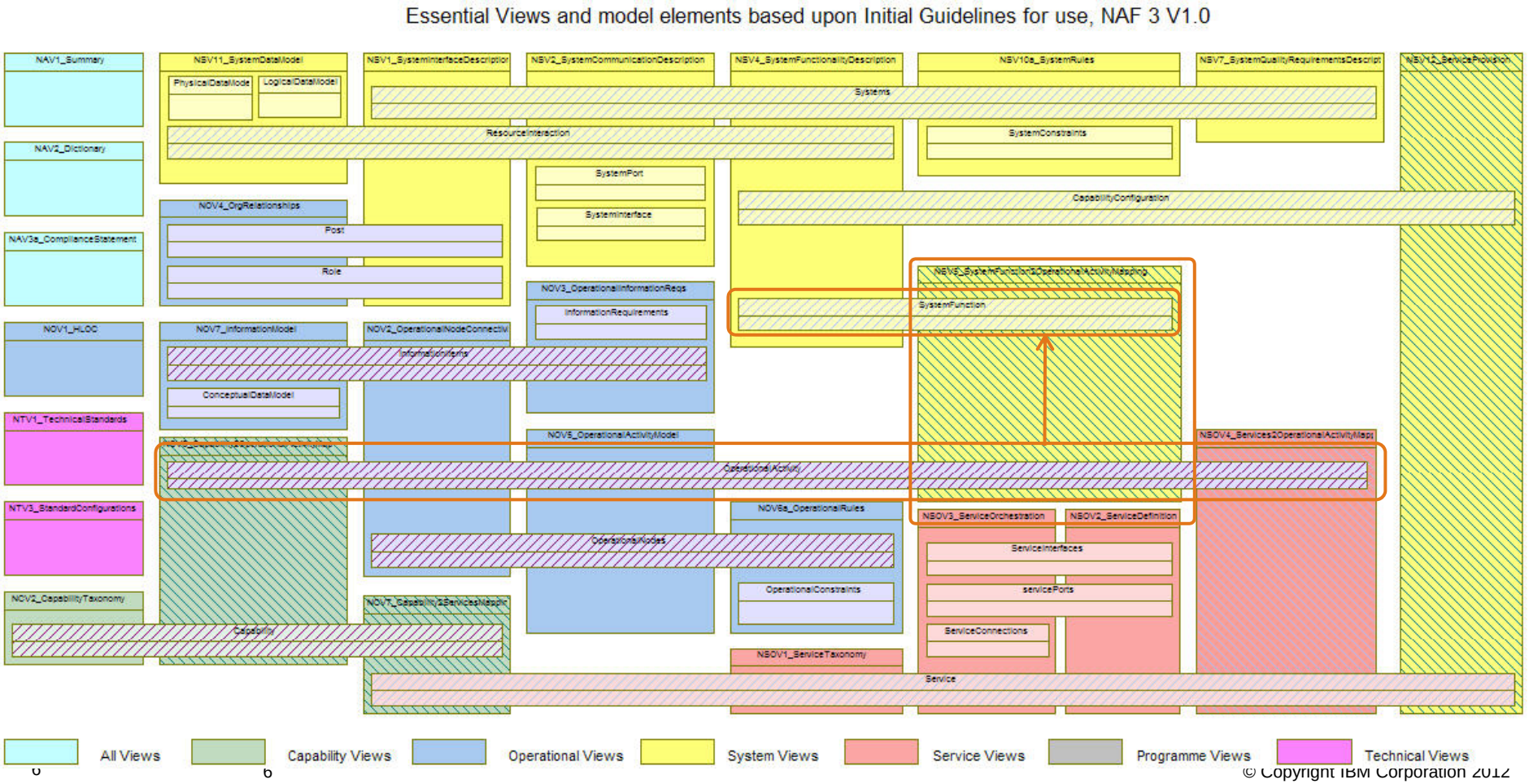
- Standards Viewpoint:-Contains all the information relating to the standards that constrain the other 3 viewpoints (prefix by StdV)
- All Viewpoint:- Contains the summary information about the architecture being developed including the data dictionary (prefix by AV)



Model Element interaction Across Views

Separation of Elements and Views/Viewpoints

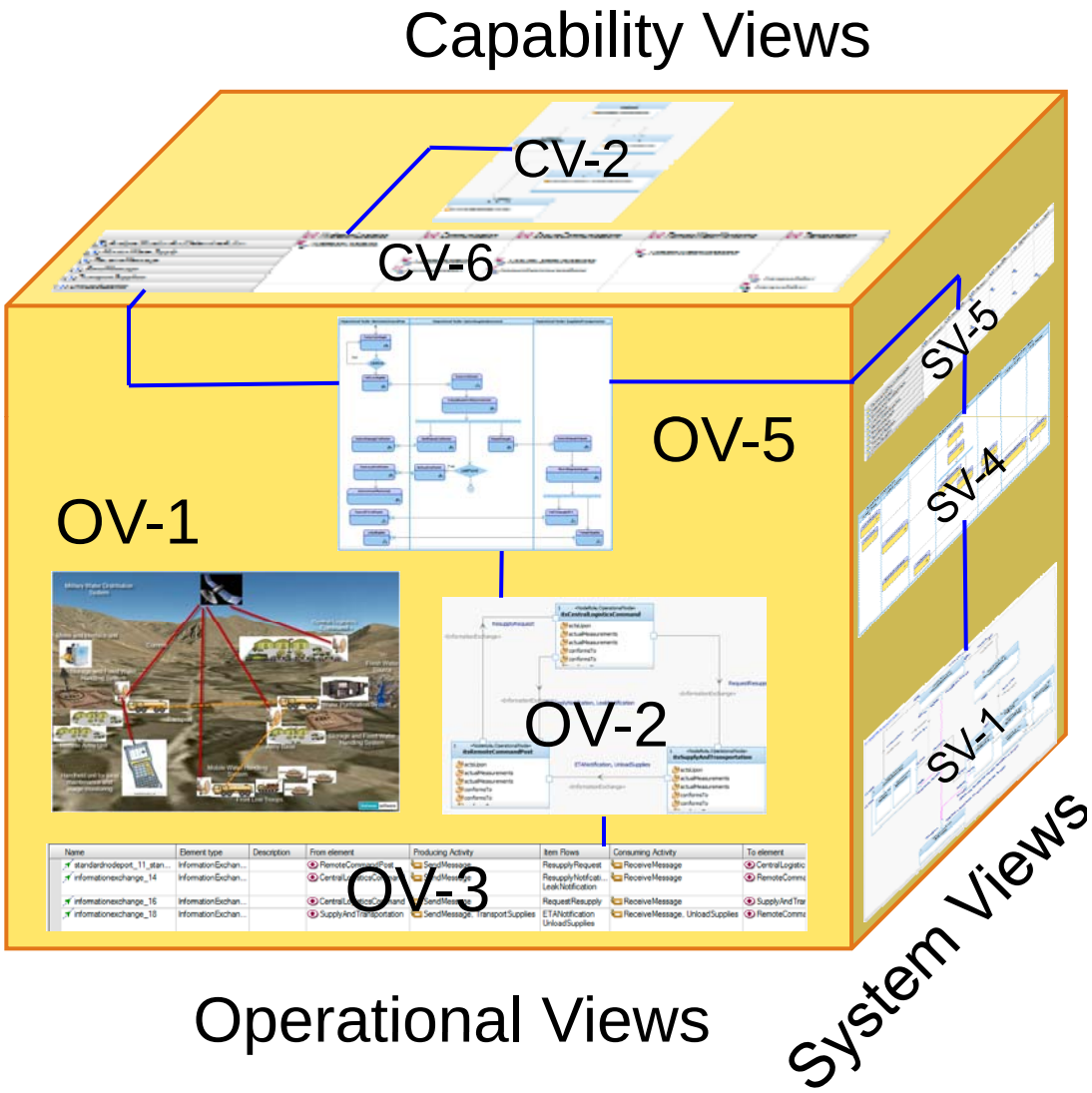
- Some elements seen in multiple views
- Some views ties the viewpoints together e.g. SV-5



6

View and Viewpoint interrelationshhips

- The Viewpoints are not independent but are heavily interrelated
- Each face of Cube represents a viewpoint
- Each window a separate view or product
- Model Elements internal to cube used by multiple views
- Views can act as
  - Filters on the information in the architecture (OV-3, SV-5)
  - Diagrams allowing you to create the information that populates the architecture (SV-1, OV-2)

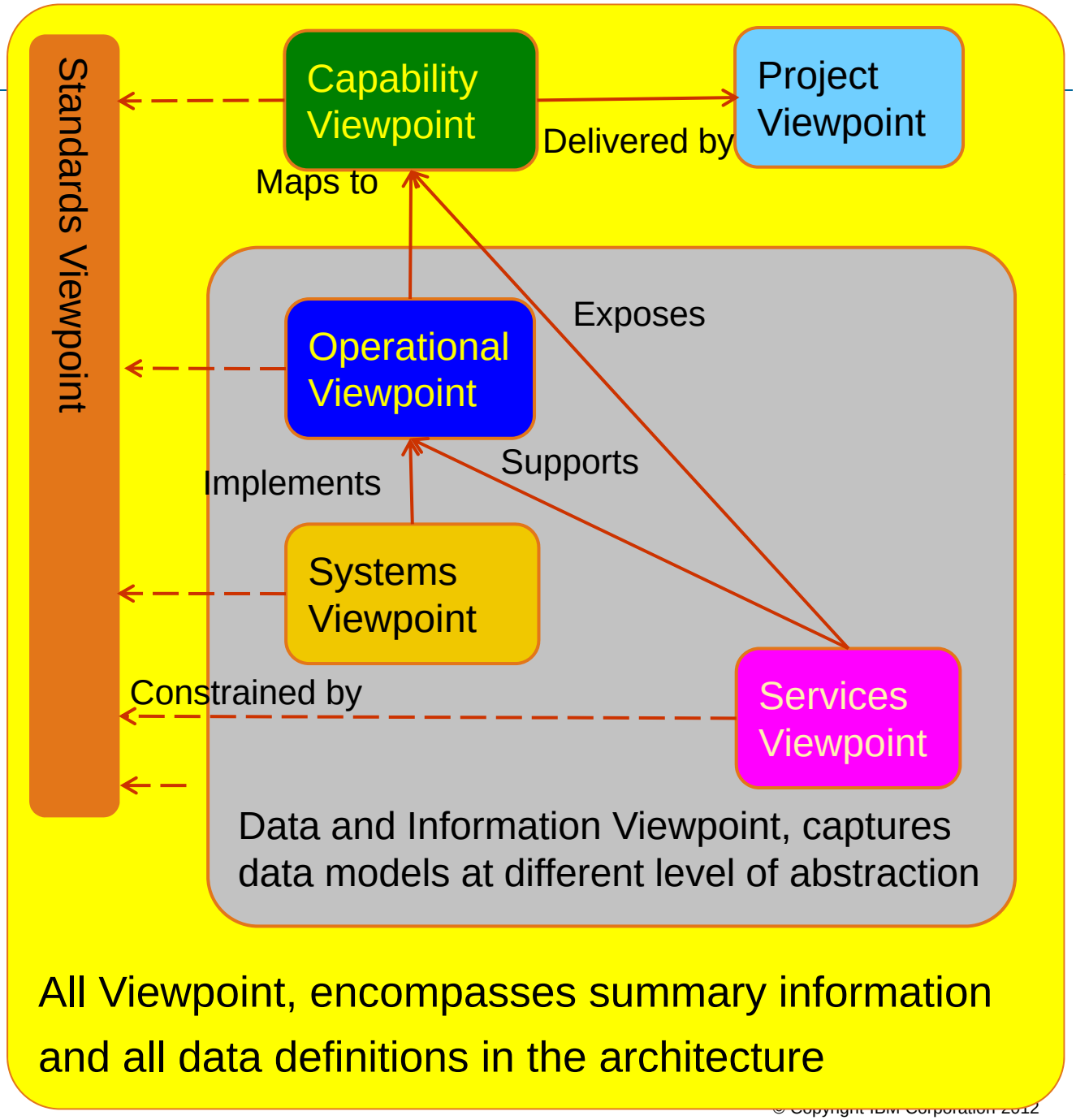


Viewpoint relationships

UPDM DoDAF is a set of traceability matrices

- Systems and Services support and implement Operational elements
- Services expose capabilities (service as an interface)
- Operational elements (activities) map to capabilities
- Capabilities are delivered by Projects
- Systems and Services are the realized by projects
- Everything constrained by standards

These relationships are captured in the various matrix views



Enterprise View of UPDM DoDAF

Used by

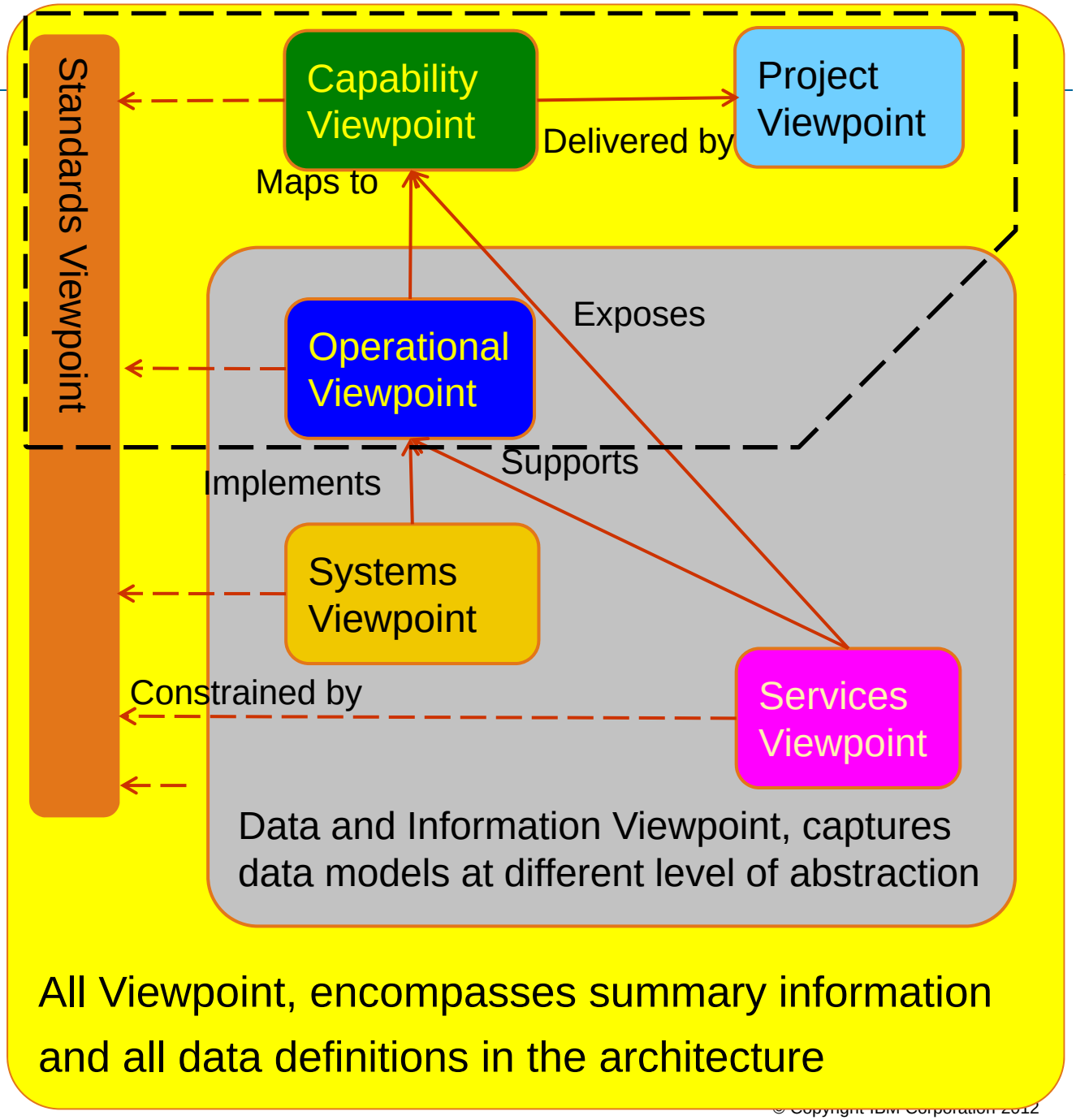
- Planning
- JCIDS
- Operations
- Portfolio management

For

- Capability management
- Operations Planning
- Develop High level requirements for prime suppliers

Still need the

- All view,
- Standards view
- Parts of the Data and Information view



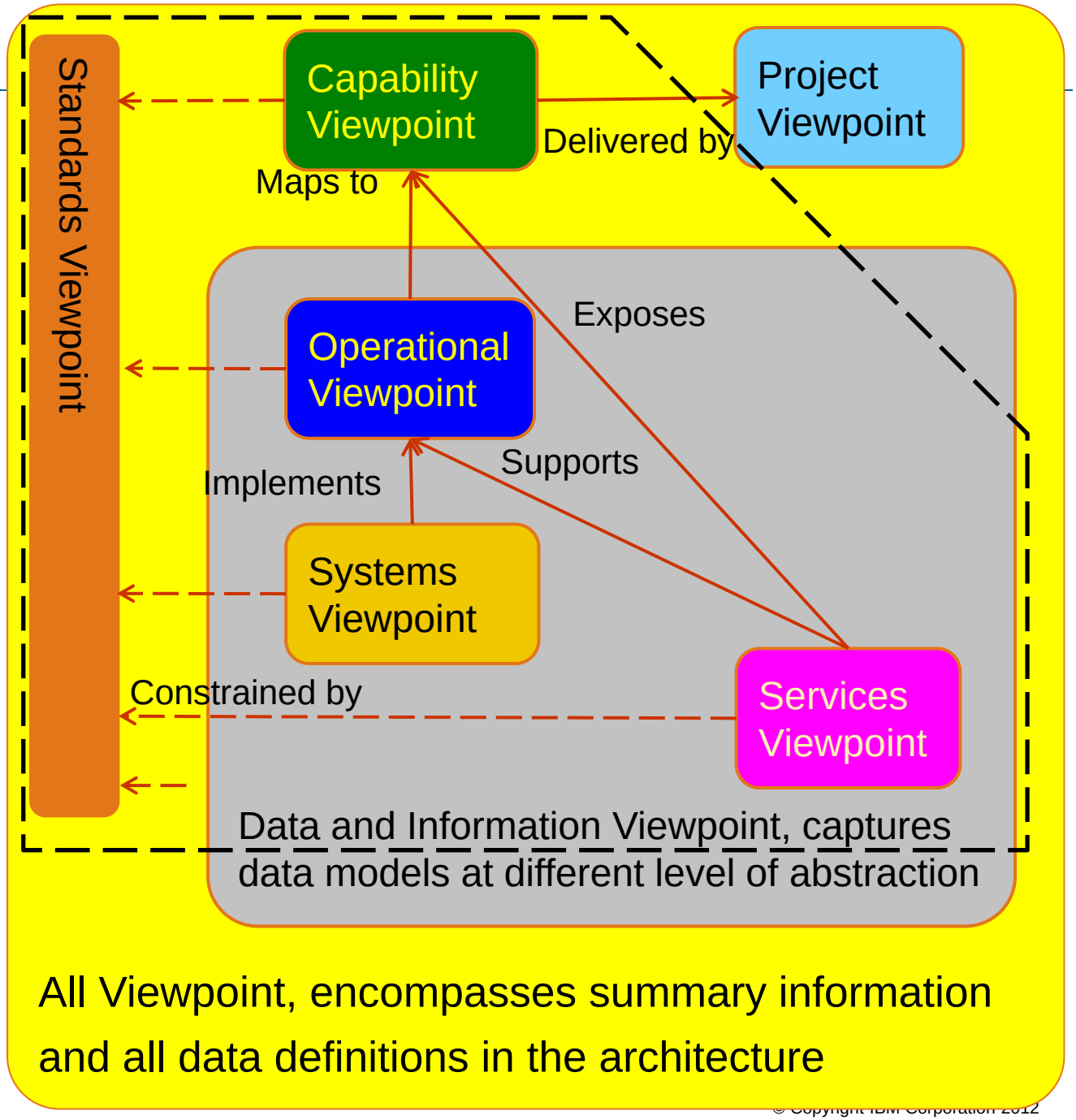
The Systems Engineering View of UPDM DoDAF

Used by

- Portfolio management
- Operations planning
- Defense Acquisition System
- Systems Engineering teams
  - In forces
  - Primes/Tier ones
- Provides requirements to engineering teams

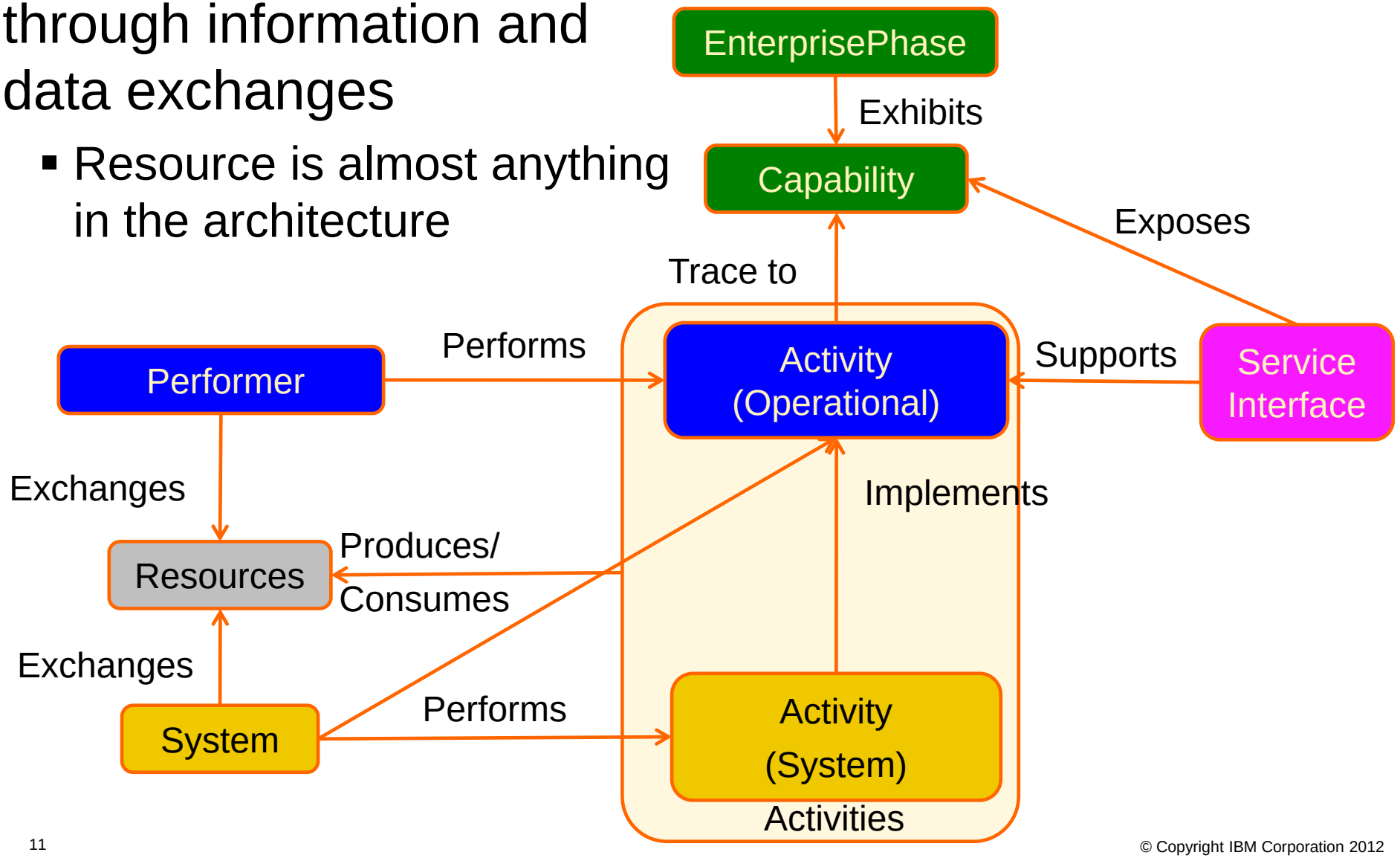
Still need the

- All view,
- Standards view
- Parts of the Data and

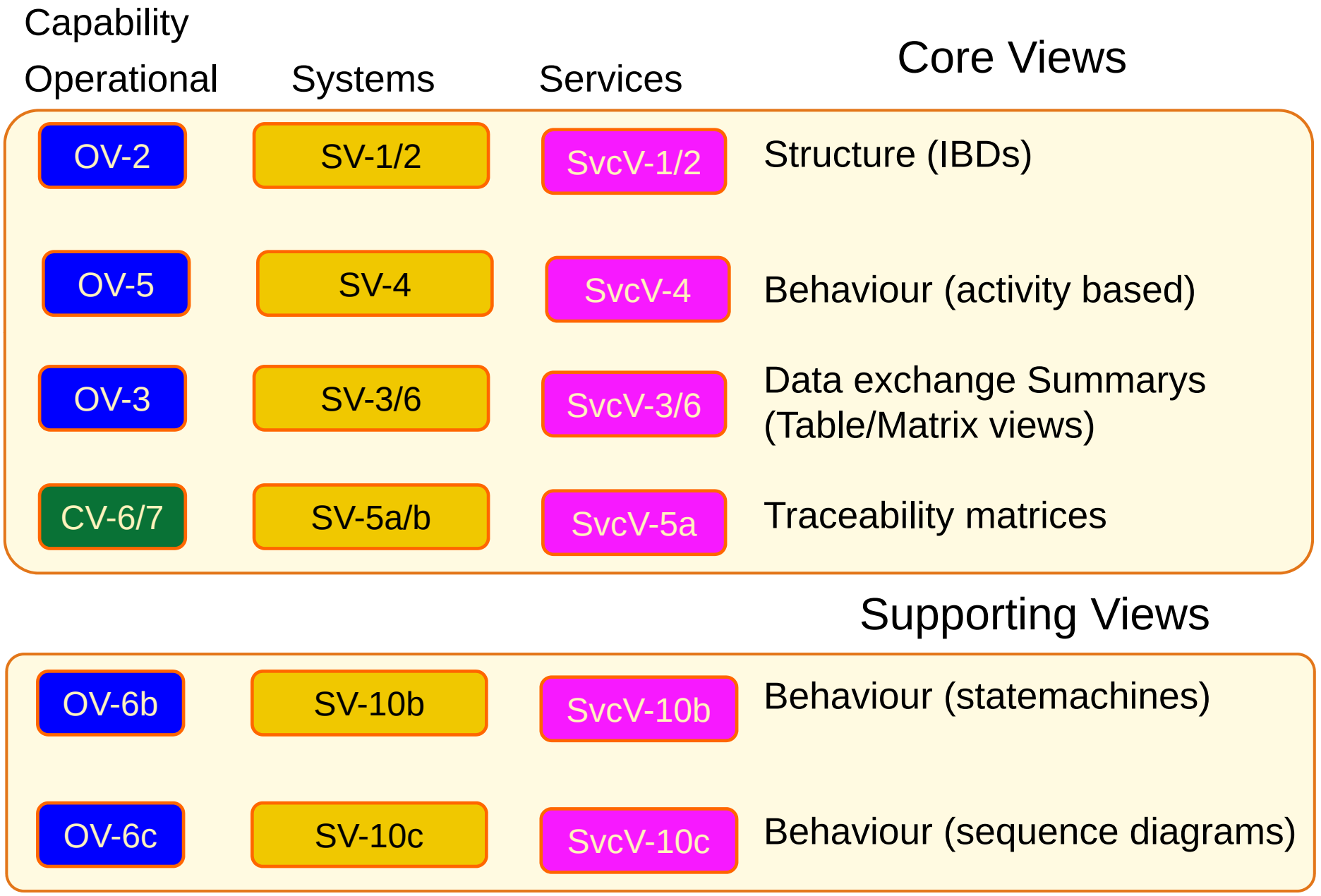


Key relationships for Systems Engineering

- Resources exchanged through information and data exchanges
  - Resource is almost anything in the architecture

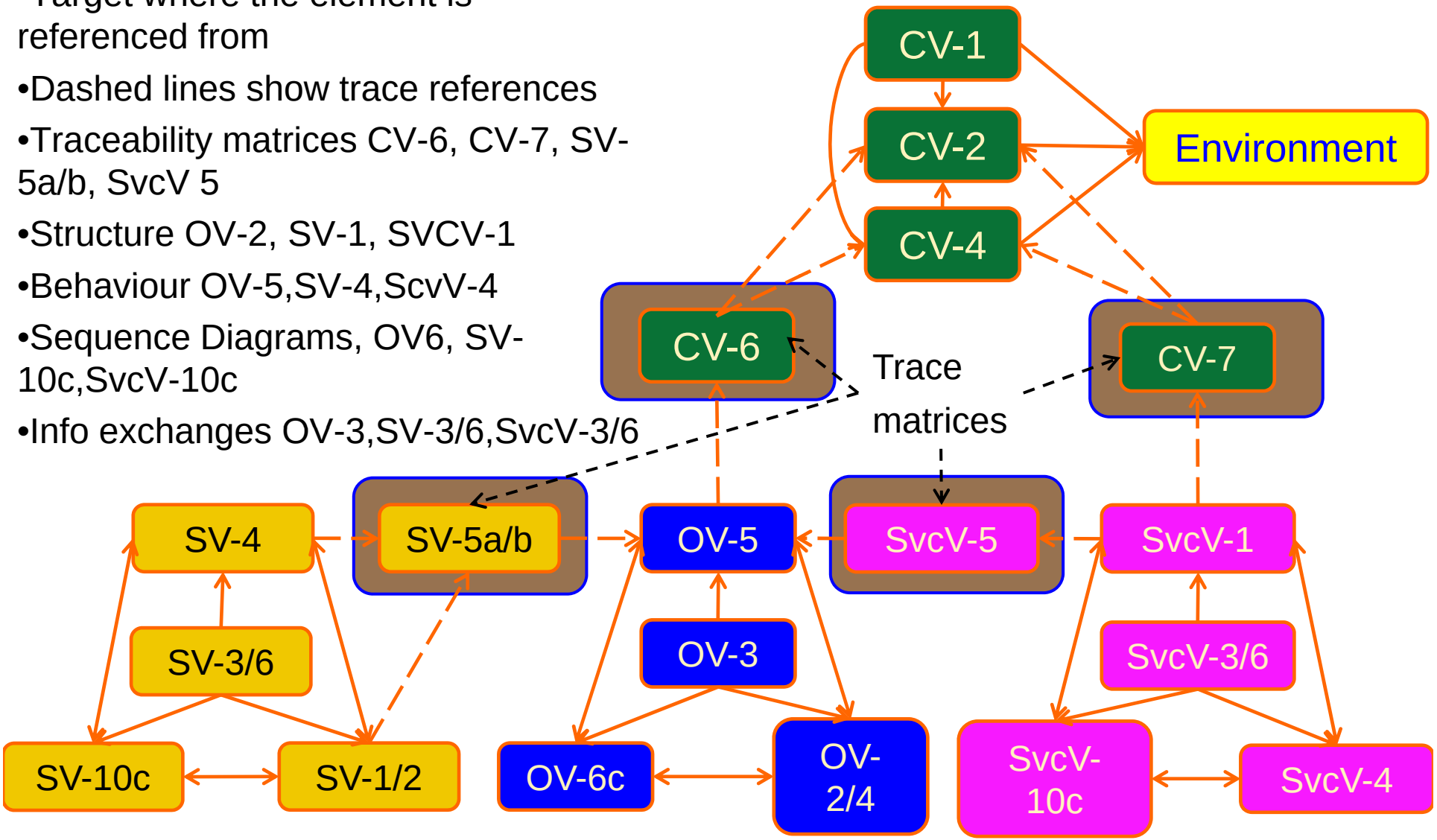


Patterns in the Framework



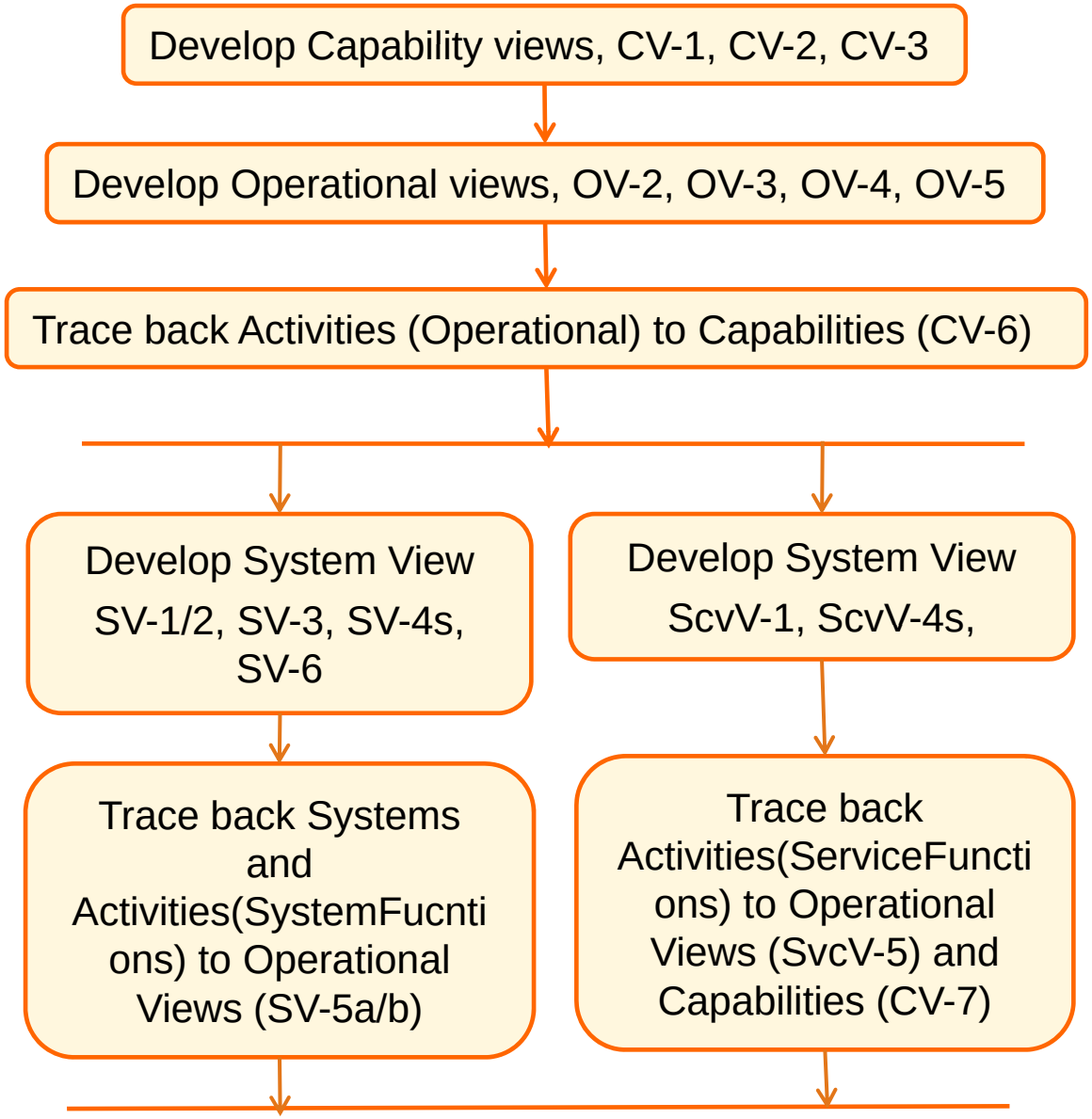
Key View dependencies for systems engineering

- Source where element is shown
- Target where the element is referenced from
- Dashed lines show trace references
- Traceability matrices CV-6, CV-7, SV-5a/b, SvcV 5
- Structure OV-2, SV-1, SVCV-1
- Behaviour OV-5,SV-4,ScvV-4
- Sequence Diagrams, OV6, SV-10c,SvcV-10c
- Info exchanges OV-3,SV-3/6,SvcV-3/6



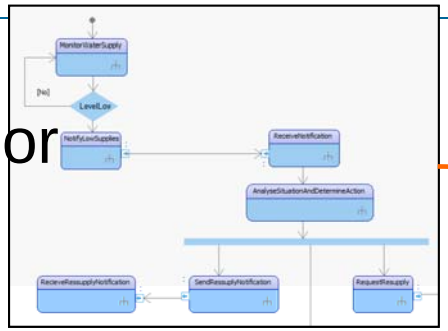
High level workflow (core)

- Develop capability views
  - Enterprise visions
  - Capability hierarchy and dependencies
- Develop Operational views
  - Behavior, structure,
  - Derive data exchange matrices
  - Trace back to capabilities
- Develop systems/service views
  - Behavior, structure,
  - Derive data exchange matrices
  - Trace back
    - Systems to operations that they implement
  - Trace back Services to the
    - Operations they support
    - Capabilities they expose



The generic workflow applied to the OVs

Step-1 BlackBox behavior



Step-2 Define Structu

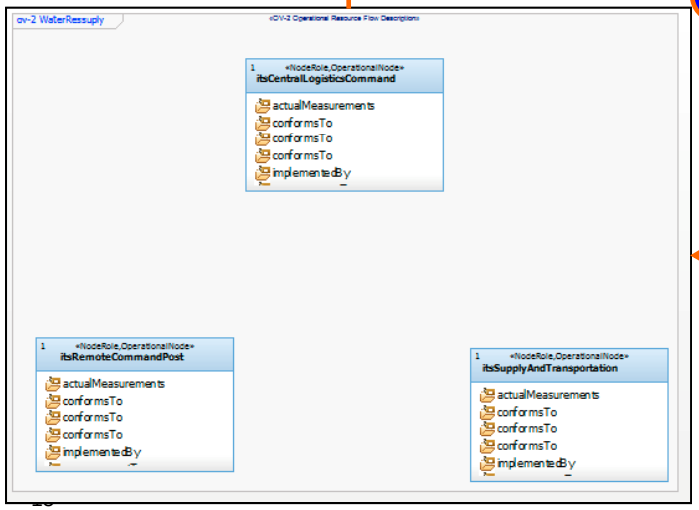
OV-5

OV-3

OV-2/4

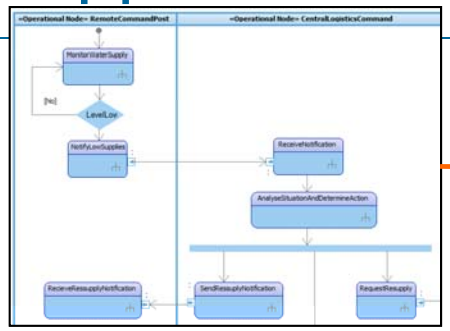
OV-6c

Step-3 Allocate behavior to structure (WhiteBox)

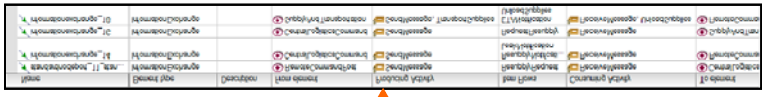


The generic workflow applied to the OVs

Step-3 Allocate behaviour to structure (WhiteBox)



Step-6 generate OV-3 Information exchange summary



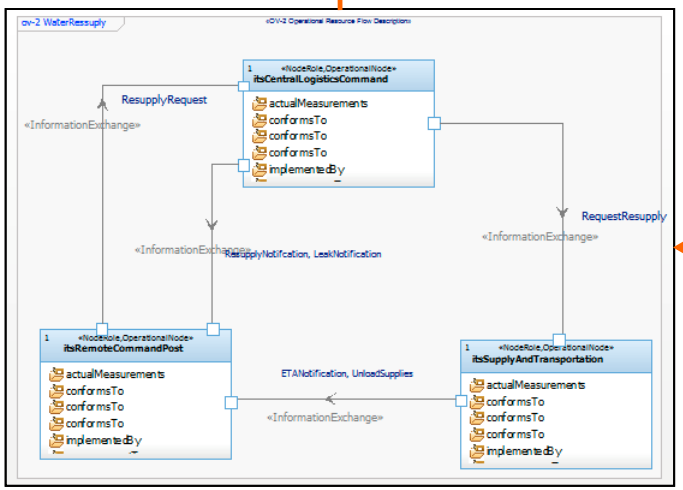
OV-5

OV-3

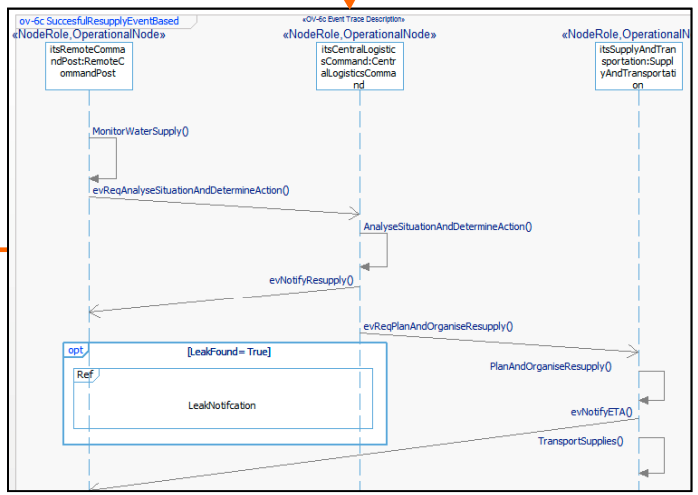
OV-2/4

OV-6c

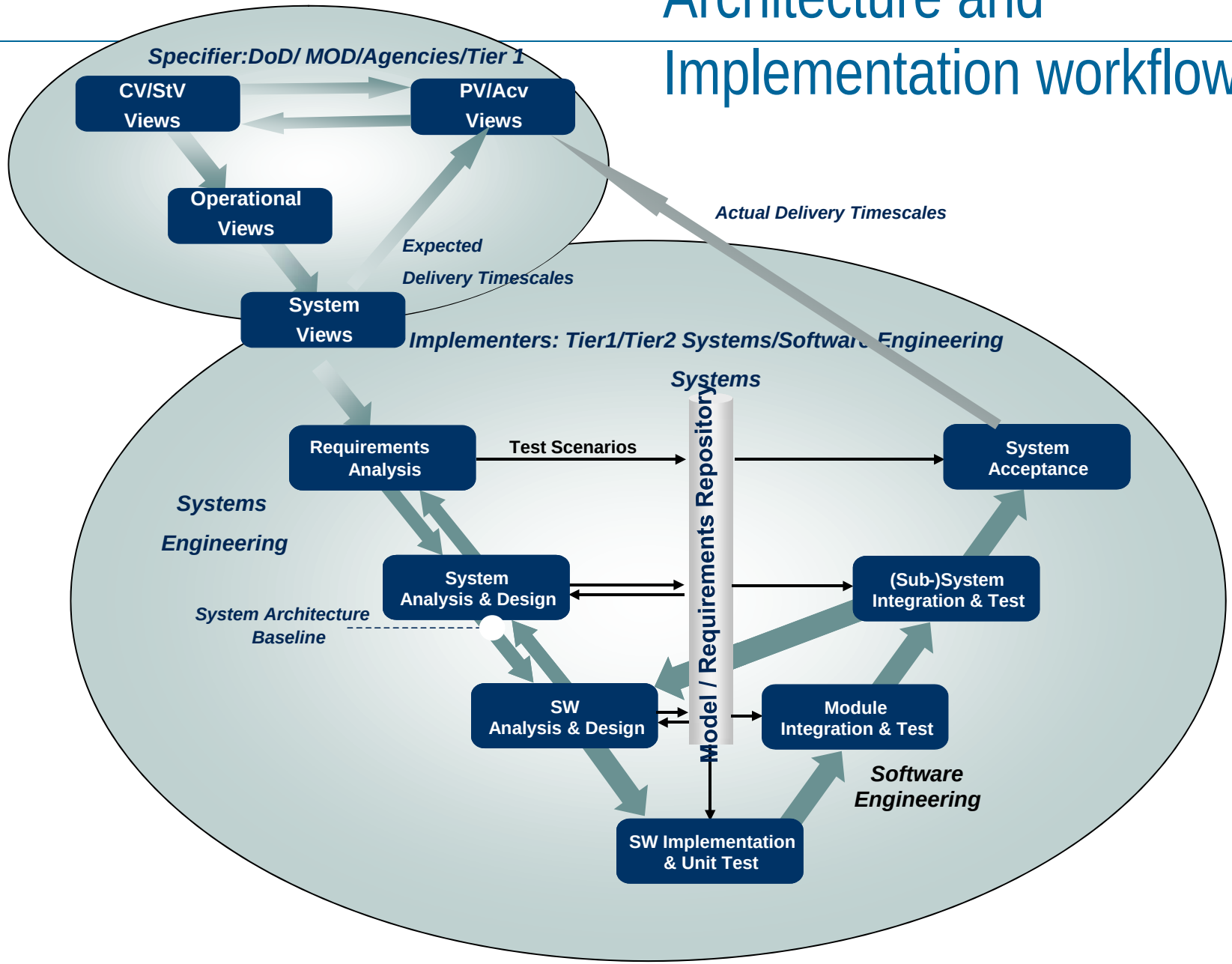
Step-4 Define Scenarios Optional



Step-5 Define Information Exchanges



Architecture and Implementation workflow

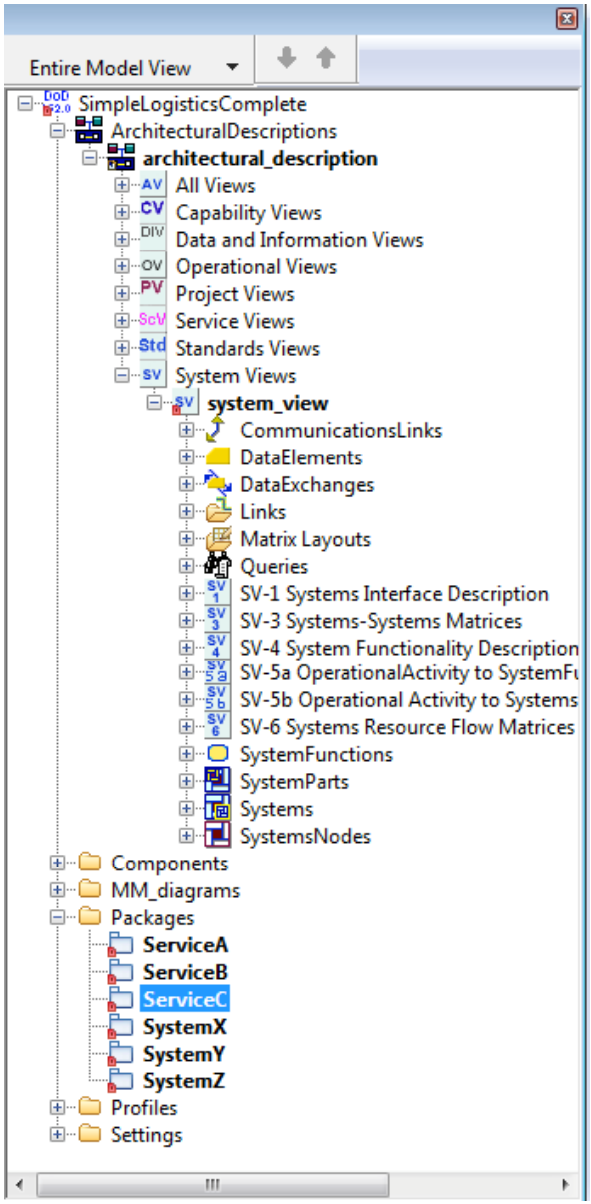


Relating UPDM and SysML models Discussion

- Handling requirements
  - In UPDM requirements driven from the level above
  - Non functional requirements captured as constraints
    - Also work with Parametric diagrams
  - It is possible to import requirements into UPDM models and tie them to Capabilities
- 3 main options
  - Keep SysML models as part of UPDM models
  - Import parts of UPDM models directly into a SysML model
  - Reference UPDM elements from SysML models
- Options as part of these are
  - Embed SysML elements directly in the UPDM elements
    - Ties the elements too closely into UPDM but I think there will be issues
  - Separate out SysML models and trace to UPDM elements
    - Keep to the patterns in UPDM re traceability to the upper levels
- 4 th option
  - Separate UPDM and SysML models and trace through third party tool (i.e. DOORS)

Relating UPDM and SysML models Discussion

- Keep SysML models as part of UPDM models
  - Models become very large and monolithic
  - Hard to maintain and manage

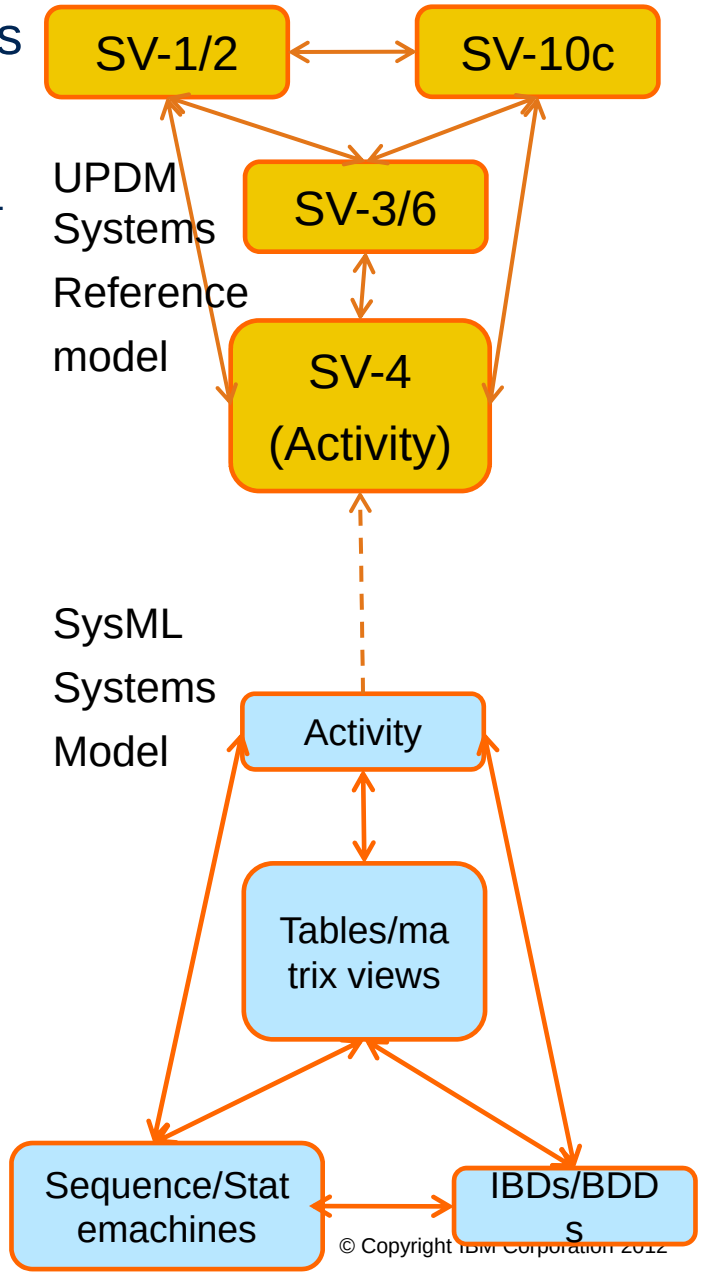


Relating UPDM and SysML models Discussion

- ┬Import parts of UPDM models directly into a SysML model
  - Evolve the model with SysML elements
- End up with separate models
- Disconnect between the high level specification of a System in the UPDM model and the ┬SysML
- Issues with the need to maintain consistency between the views
- Issues with pushing more detail up into the UPDM views as there will then be inconsistency in the detail and you may lose relations/dependencies to the elements at the higher level
  - **Interchange issue as you are replacing elements you may lose ownership of relationships (might be OK in the same tool, but XMI ?)**
  - **Need Diff-Merge capability for XMI ?**

Relating UPDM and SysML models Discussion

- Reference UPDM elements from SysML models
  - Maintain the consistency of the UPDM model
  - Maintain the patterns of the UPDM structure in the SysML model
  - Trace between SysML and UPDM easily
  - End up with a number of Branch SysML models that reflect the lower level system structure
  - Cannot use the UPDM elements as they are read only (Advantage)
- Initial proposal
  - Combination of taking UPDM models by reference and importing the lower levels of the UPDM architecture and removing UPDM stereotypes
    - Change to SysML elements
    - Trace back to reference elements
    - Gives the possibility of reuse and also shows the reference to original elements



Summary

- Brief review of Viewpoints
- How elements are related across the viewpoints
- Identification of patterns in the view points
- Generic Workflow
- UPDM and SysML