" Sikkerhet ved endring"
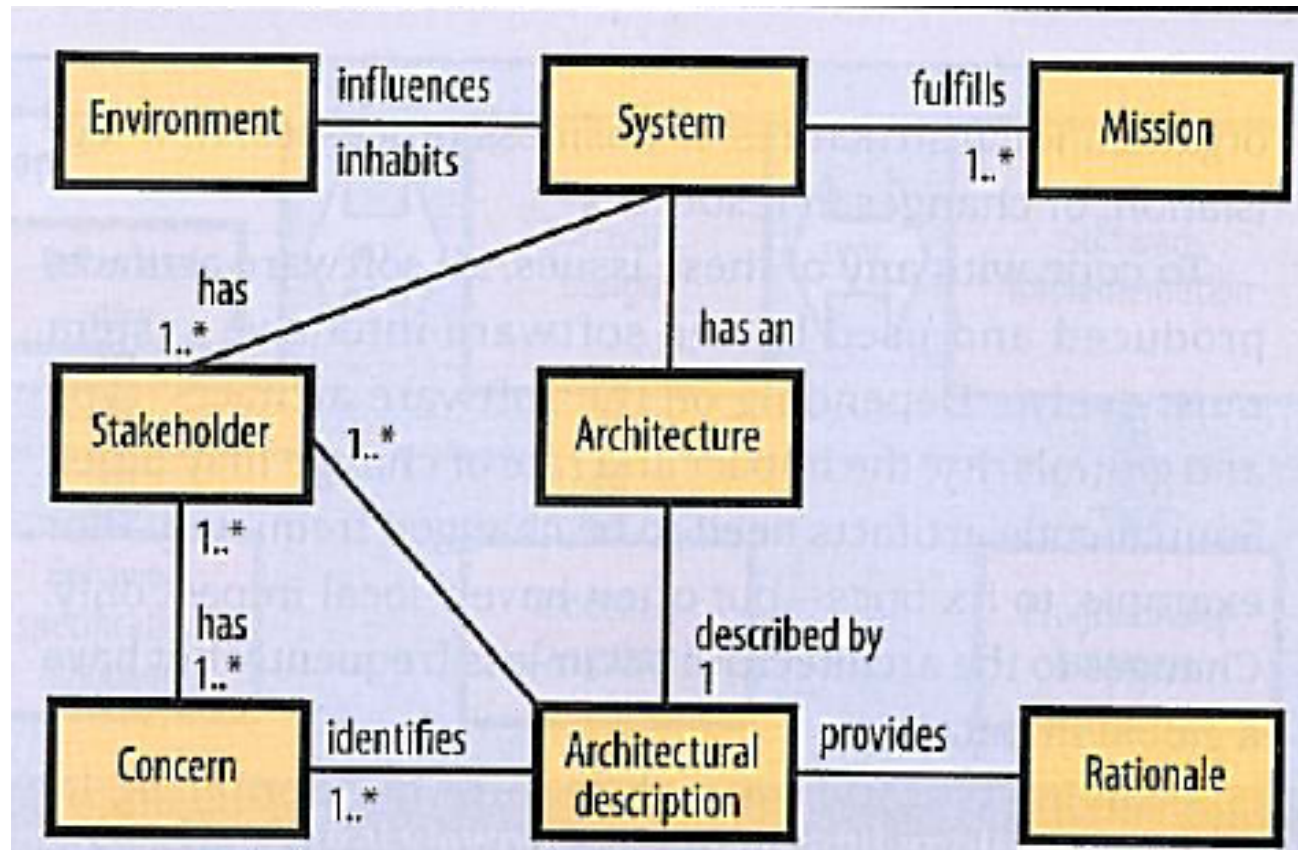# - *Fra en teknologs (arkitekts) ståsted* –

Arne-Jørgen Berre, SINTEF og Institutt for informatikk, UiO
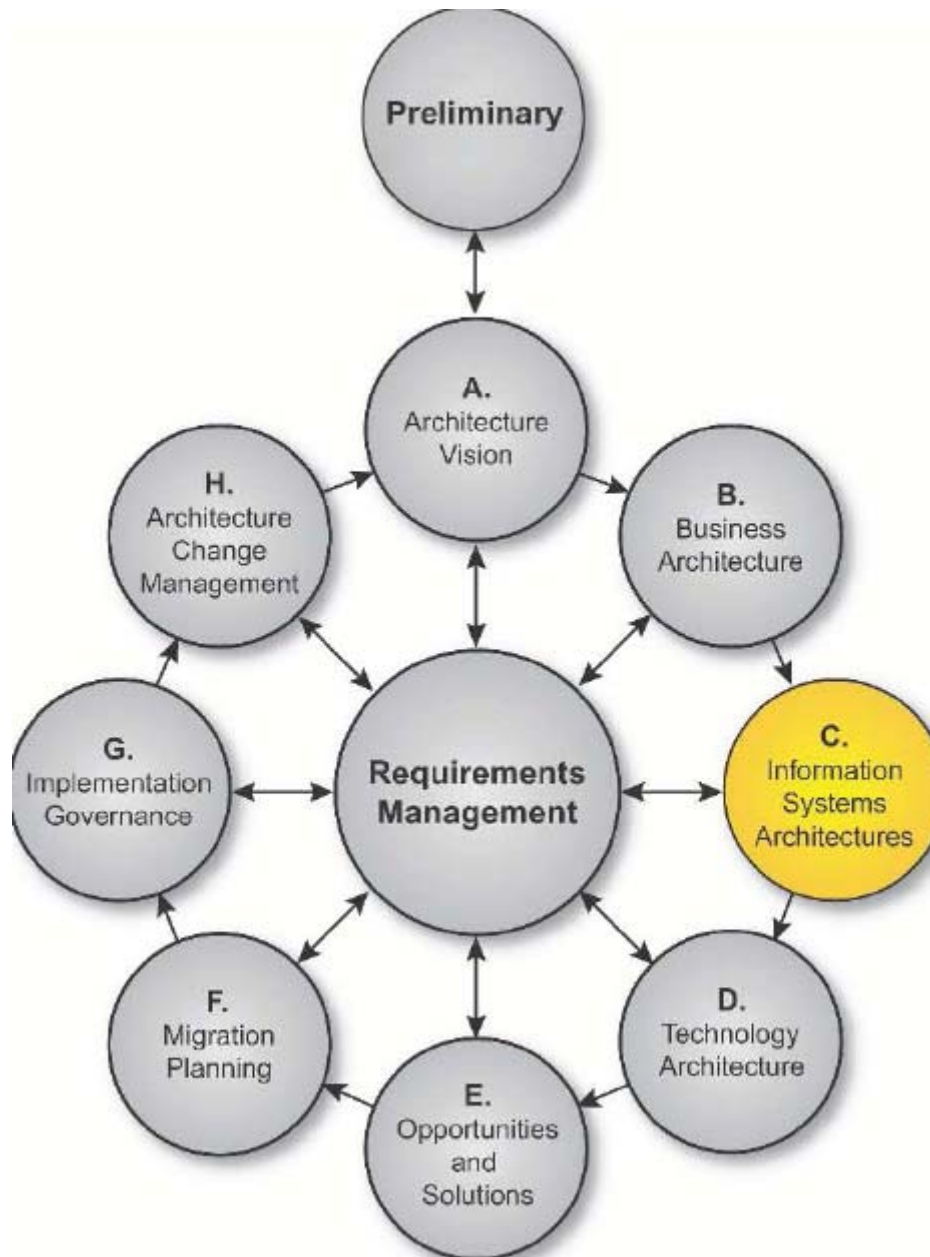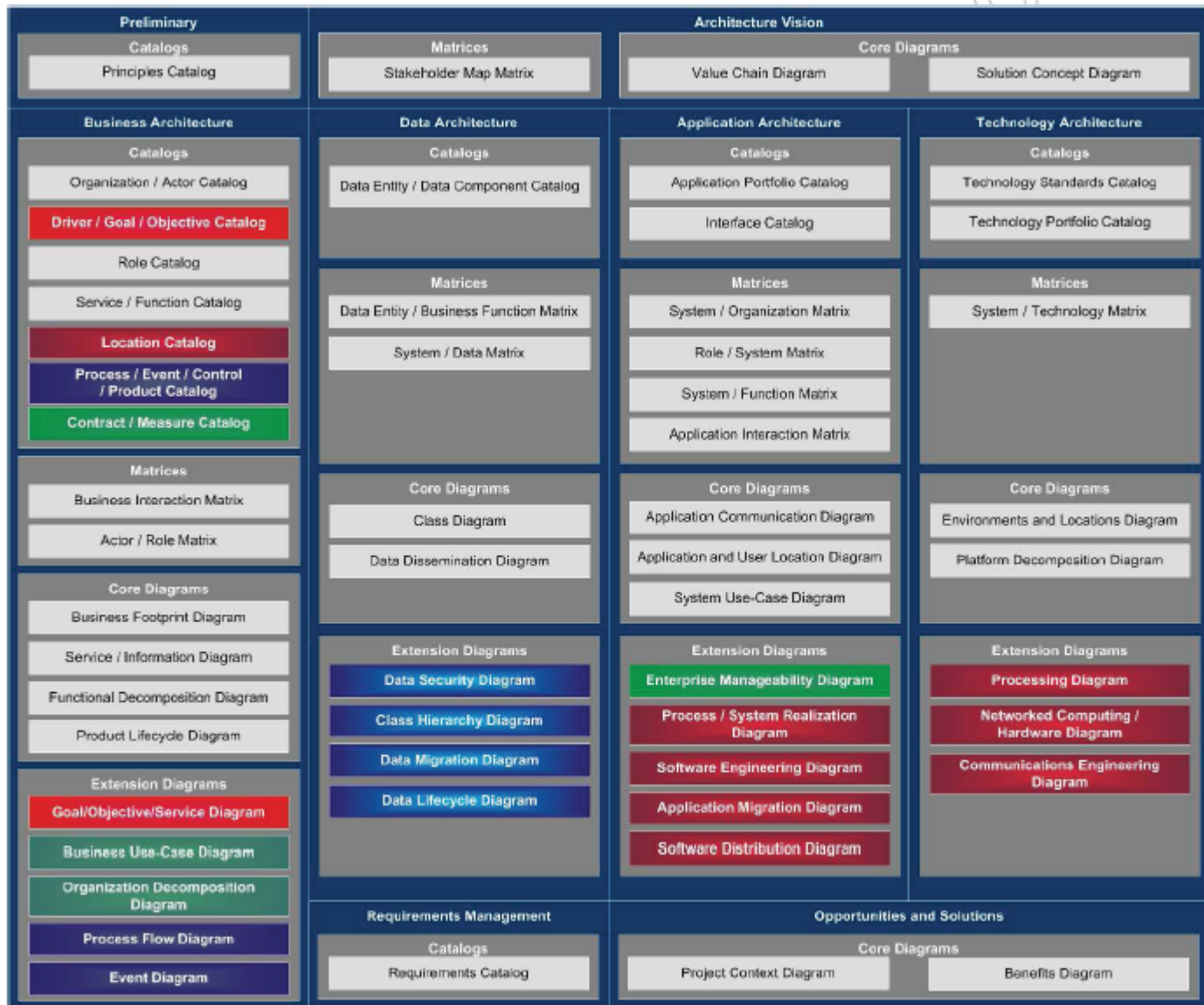
Oslo, 24. juni, 2010

Innhold

- Sikkerhet ved Endring

- Fra en Arkitekts ståsted (Arkitekturperspektiv)

- Arkitekturmodeller / Arkitekturrammeverk (IEEE/ 1471/ISO 42010, ADL, UML 2.x, TOGAF, UPDM (DODAF/MODAF), SoaML (*SHAPE*), DSLs)

- Sikkerhet og Modeller (SecureUML, ObjectSecurity,DSML-Security, AOM)

- Systemer og modeller (ADM og MDA) (*REMICS)*

- Modeller og Endring/Evolusjon (Subclassing, parameterisation, templates, Role models/Coll.models (SoaML), CVL (Common Variability Language) (*MoSiS*), Traceability, Service Variability, Resemblance & Replacement)
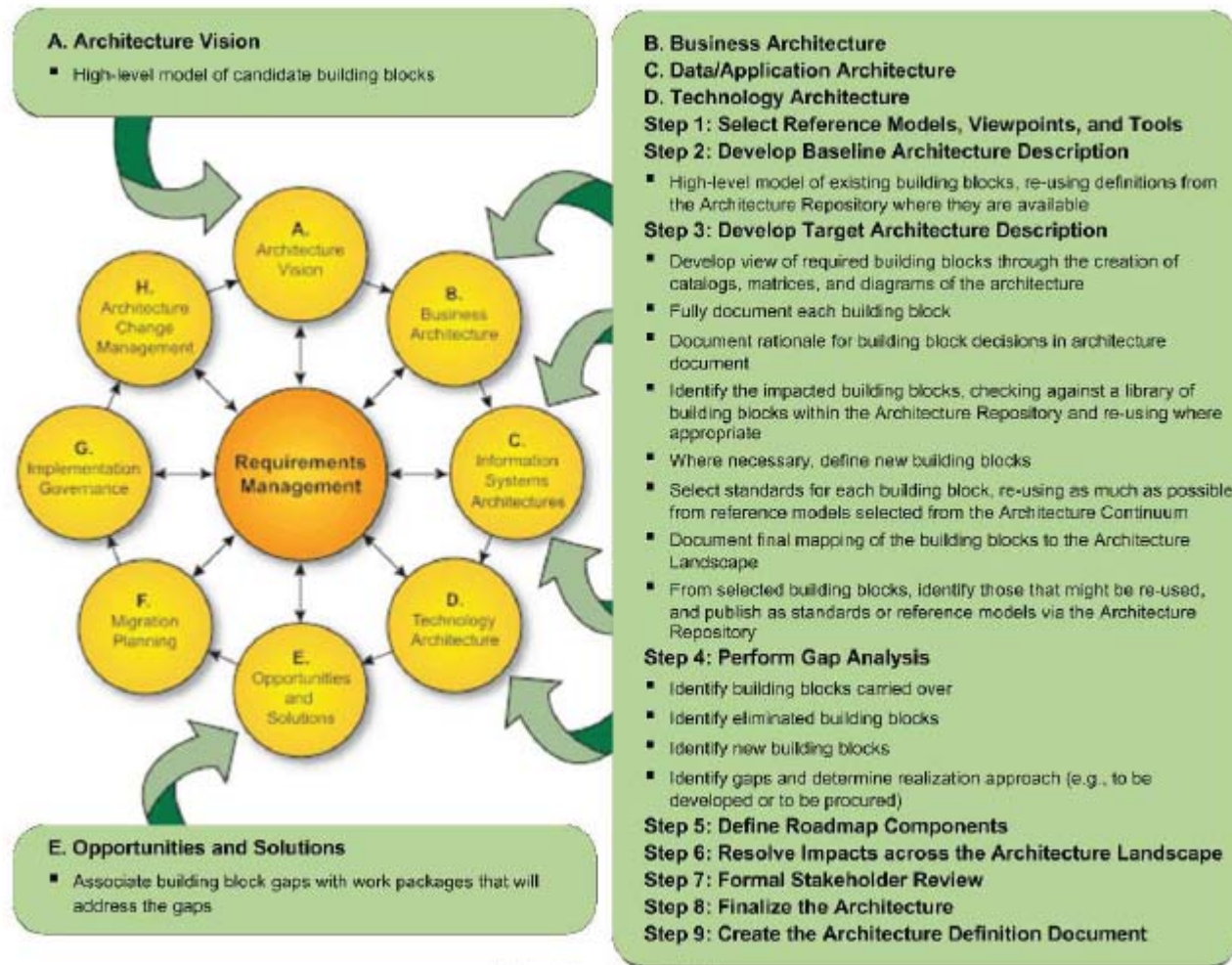
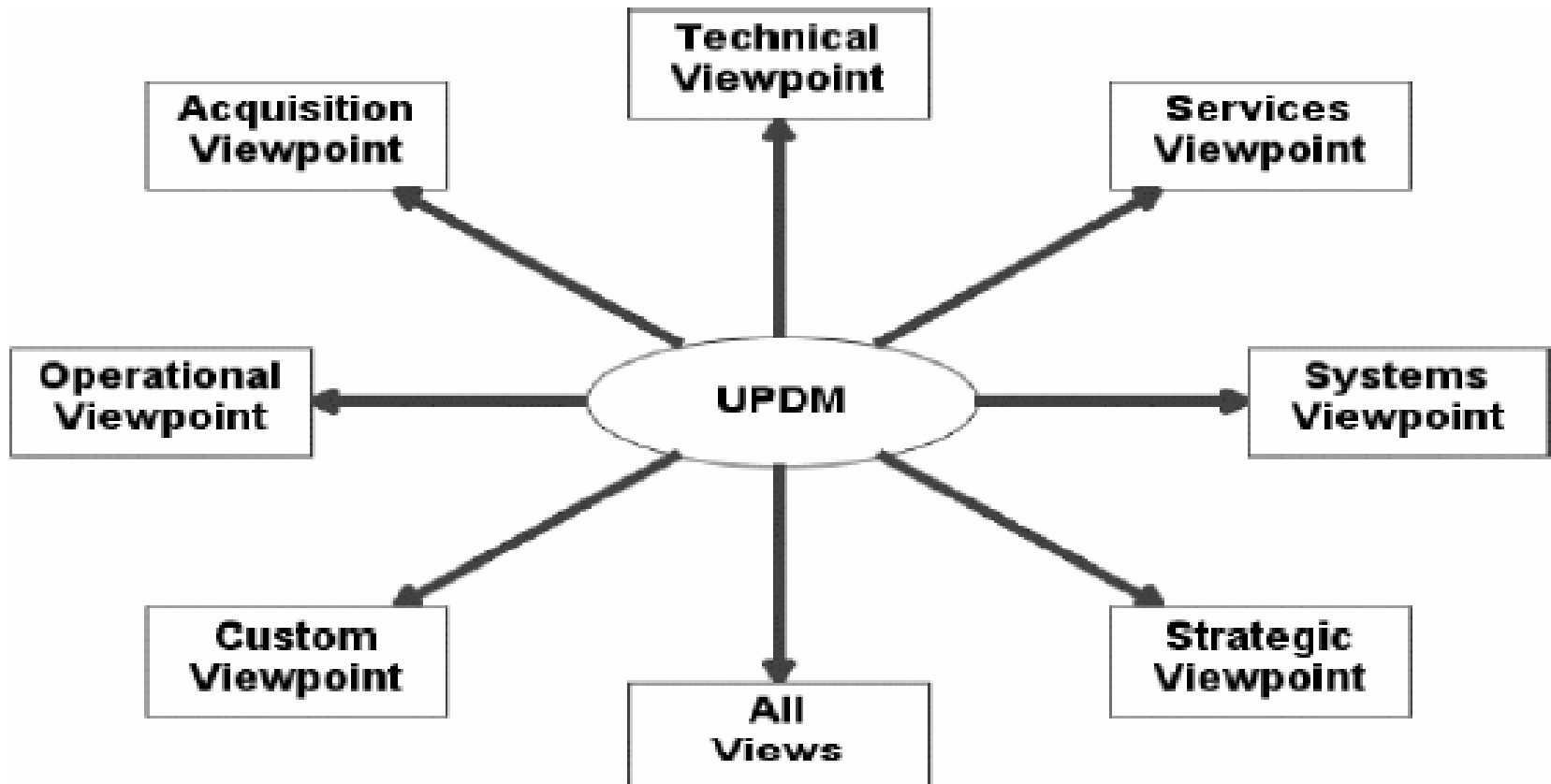# IEEE 1471, ISO 42010

Open Group ADM

Preliminary	Architecture Vision		
Catalogs	**Matrices**	**Core Diagrams**	
Principles Catalog	Stakeholder Map Matrix	Value Chain Diagram	Solution Concept Diagram

Business Architecture	Data Architecture	Application Architecture	Technology Architecture
Catalogs	**Catalogs**	**Catalogs**	**Catalogs**
Organization / Actor Catalog	Data Entity / Data Component Catalog	Application Portfolio Catalog	Technology Standards Catalog
Driver / Goal / Objective Catalog		Interface Catalog	Technology Portfolio Catalog
Role Catalog			
Service / Function Catalog	**Matrices**	**Matrices**	**Matrices**
Location Catalog	Data Entity / Business Function Matrix	System / Organization Matrix	System / Technology Matrix
Process / Event / Control / Product Catalog	System / Data Matrix	Role / System Matrix	
Contract / Measure Catalog		System / Function Matrix	
		Application Interaction Matrix	
Matrices			
Business Interaction Matrix	**Core Diagrams**	**Core Diagrams**	**Core Diagrams**
Actor / Role Matrix	Class Diagram	Application Communication Diagram	Environments and Locations Diagram
	Data Dissemination Diagram	Application and User Location Diagram	Platform Decomposition Diagram
Core Diagrams		System Use-Case Diagram	
Business Footprint Diagram			
Service / Information Diagram	**Extension Diagrams**	**Extension Diagrams**	**Extension Diagrams**
Functional Decomposition Diagram	Data Security Diagram	Enterprise Manageability Diagram	Processing Diagram
Product Lifecycle Diagram	Class Hierarchy Diagram	Process / System Realization Diagram	Networked Computing / Hardware Diagram
	Data Migration Diagram	Software Engineering Diagram	Communications Engineering Diagram
Extension Diagrams	Data Lifecycle Diagram	Application Migration Diagram	
Goal/Objective/Service Diagram		Software Distribution Diagram	
Business Use-Case Diagram			
Organization Decomposition Diagram	**Requirements Management**	**Opportunities and Solutions**	
Process Flow Diagram	**Catalogs**	**Core Diagrams**	
Event Diagram	Requirements Catalog	Project Context Diagram	Benefits Diagram

Legend:
- Infrastructure Consolidation Extension
- Governance Extension
- Motivation Extension
- Process Modeling Extension
- Data Modeling Extension
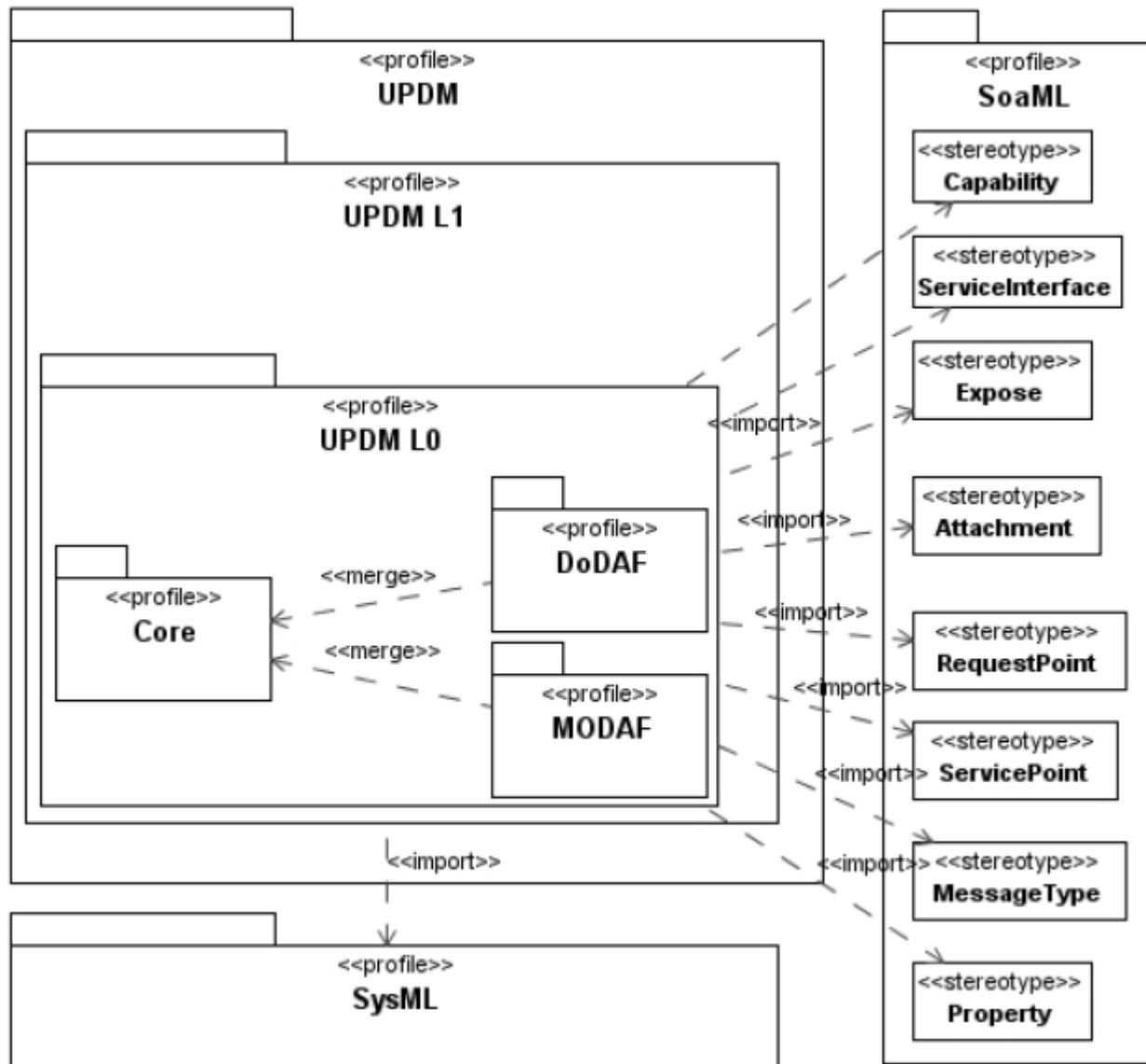- Services Extension
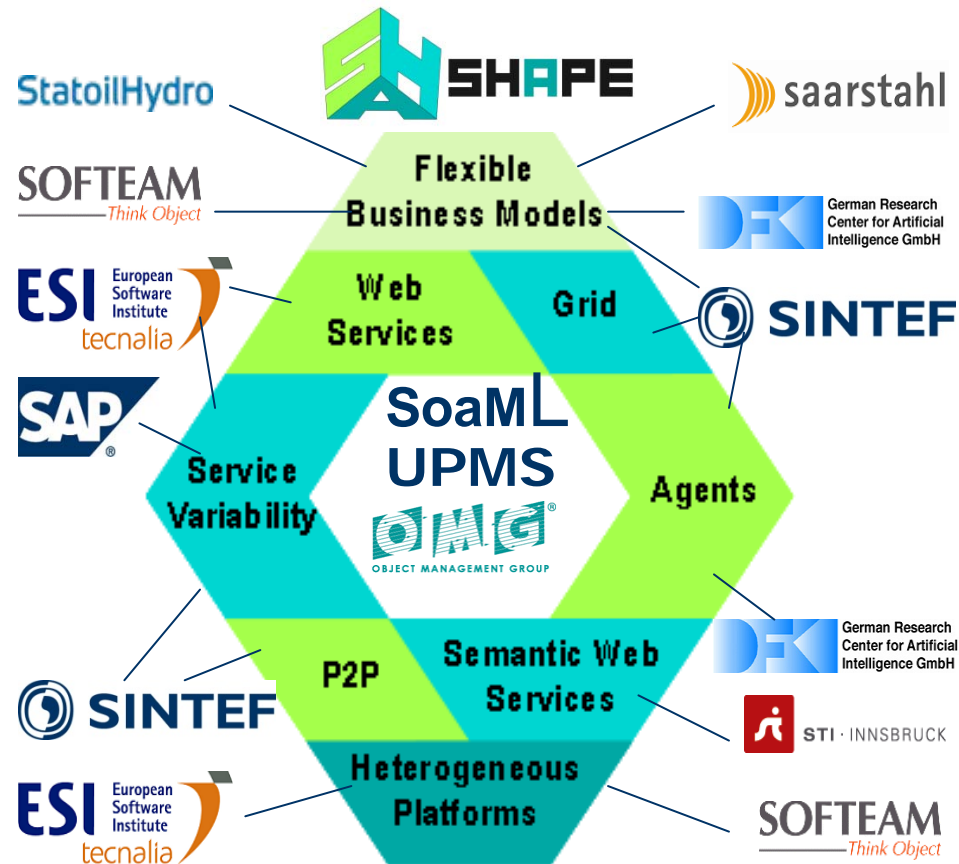- Core Content

Building block evolution

A. Architecture Vision
- High-level model of candidate building blocks

B. Business Architecture
C. Data/Application Architecture
D. Technology Architecture
Step 1: Select Reference Models, Viewpoints, and Tools
Step 2: Develop Baseline Architecture Description
- High-level model of existing building blocks, re-using definitions from the Architecture Repository where they are available

Step 3: Develop Target Architecture Description
- Develop view of required building blocks through the creation of catalogs, matrices, and diagrams of the architecture
- Fully document each building block
- Document rationale for building block decisions in architecture document
- Identify the impacted building blocks, checking against a library of building blocks within the Architecture Repository and re-using where appropriate
- Where necessary, define new building blocks
- Select standards for each building block, re-using as much as possible from reference models selected from the Architecture Continuum
- Document final mapping of the building blocks to the Architecture Landscape
- From selected building blocks, identify those that might be re-used, and publish as standards or reference models via the Architecture Repository

Step 4: Perform Gap Analysis
- Identify building blocks carried over
- Identify eliminated building blocks
- Identify new building blocks
- Identify gaps and determine realization approach (e.g., to be developed or to be procured)

Step 5: Define Roadmap Components
Step 6: Resolve Impacts across the Architecture Landscape
Step 7: Formal Stakeholder Review
Step 8: Finalize the Architecture
Step 9: Create the Architecture Definition Document

E. Opportunities and Solutions
- Associate building block gaps with work packages that will address the gaps

UPDM - – Unified Model for DODAF and MODAF

UPDM – Unified Model for DODAF and MODAF

SHAPE project and SoaML

CIM to PIM to PSM

CIM – PIM - PSM

CIM
Business Models

BPMN   BPDM   BMM   EPC   …

SoaML-SHA

Core

SoaML

Service Variability

PIM
System Models

PIM4 WS-A   PIM4 SWS   PIM4 Agents   P2P/Grid/ Components

PIMs for different Architectural Styles

WSDL, WSMO, OWŁS, JACK, JADE, JXTA, OGSA, J2EE, CORBA

PSM
Implementation Models

J2EE, NetWeaver, .Net, …

Realization Technologies

SINTEF

CIM-PIM-PSM Reference Matrix

Aspect / Level		Information	Service	Process	Rules	Events	Organization	Goals	NFA
CIM	MM	Ontologies - ODM	(BPMN2) (SoaML)	BPMN EPC	SBVR	EPC BPMN (EMP)	OSM	BMM	OMG MM for performance / security / quality
	Tool	Objecteering, WSMT	Objecteering	CIMFlex	CIMFlex	CIMFlex	CIMFlex, Objecteering	CIMFlex, Objecteering	Objecteering, WSMT
	Meth	OE Methodologies, GERAM,ARIS, EUP, COMET-S, ESIM, SCM, SM, ISE, ESOA	GERAM,ARIS, EUP COMET-S, OGSOA, ESIM, SM, SCM, SMART, SOMA, ISE, ESOA	GERAM,ARIS, EUP, COMET-S, OGSOA, ESIM, SAE,SCM, SM, SMART, SOAD, SOMA, ISE, ESOA	GERAM, EUP, ESIM, SM, SOMA, ISE, ESOA, Cyc	GERAM, EUP	GERAM,ARIS, EUP, ESIM, SAE, SM, SMART, SOMA, ISE, ESOA	GERAM,ARIS, EUP, COMET-S, ESIM, SM, SMART, SOMA, ISE, ESOA	GERAM, ESIM, SCM, SM, SOMA, ISE, ESOA
CIM2PIM	Tool								
	Meth	COMET-S	COMET-S	COMET-S					
PIM	MM	UML Class diagram ODM, IMM	SoaML	UML Behaviour (BPMN)	(BPR)	EMP	SoaML Participant, UML Deploym. Element	(Agent Goals), (WSMO Goals)	OMG MM for performance, security, quality
	Tool	WSMT	Objecteering, PIM4Agents,WSMT	Objecteering	WSMT	CIMFlex	Objecteering	PIM4Agents, WSMT	Objecteering, WSMT
	Meth	COMET-S, OASIS, ESIM, SCM, SM, SMART, SOMA, ISE, ESOA	COMET-S, OASIS, OGSOA, ESIM, SAE, SCM, SOAD, SMART, SOMA, ISE, ESOA	OASIS, OGSOA, ESIM, SAE, SCM, SMART, SOAD, SOMA, ISE, ESOA	SMART, ISE, ESOA	OASIS	SMART, ESOA	SMART	OASIS, ESIM, SCM, SMART, SOMA, ISE, ESOA
PIM2PSM	Tool	[automated model transformation]	[automated model transformation]	[automated model transformation]	[automated model transformation]	[automated model transformation]	[automated model transformation]	[automated model transformation]	[automated model transformation]
	Meth	COMET-S	ESOA, COMET-S		ESOA	ESIM, ESOA	ESIM	ESIM	ESIM
PSM	WS	XML	WSDL	BPEL	RTF	-	-	-	WS*-standards
	Agent	Jack: Data Jade: Classes	-	Jack: Plans Jade: Behaviors	Jack: Plans Jade: -	Jack: Events Jade: Messages	Jack: Team Jade: Agent/Organ.	Jack: Goals Jade: -	- / -
	SWS	OWL WSML	OWL-S WSMO	OWL-S WSMO	SWRL WSML	-	-	WSMO Goals	WSMO NFP
	P2P	-	JXTA	JXTA	-	(JXTA)	-	-	-
	Grid	Grid Resource Ontologies	OGSA (Open Grid Service Architect.)	OGSA	-	-	OGSA (Virtual Organizat. Management)	JSDL (Job Submission Description Lang.)	Grid Security Infrastructure (GSI)

SoaML Historikk
(Service oriented architecture modeling language)

- OMG RFP – September 2006
- 3 initial submissions – June 2007
- Merge process in 2008 and 2009
- SoaML 1.0 ferdigstilt desember 2009
- SoaML 1.0 adopteres av OMG i mars 2010
- FTF chairs: Arne J. Berre, SINTEF og Jim Amsden, IBM

- http://www.soaml.org

Service Architecture Modeling with SoaML collaboration models

Services Architecture
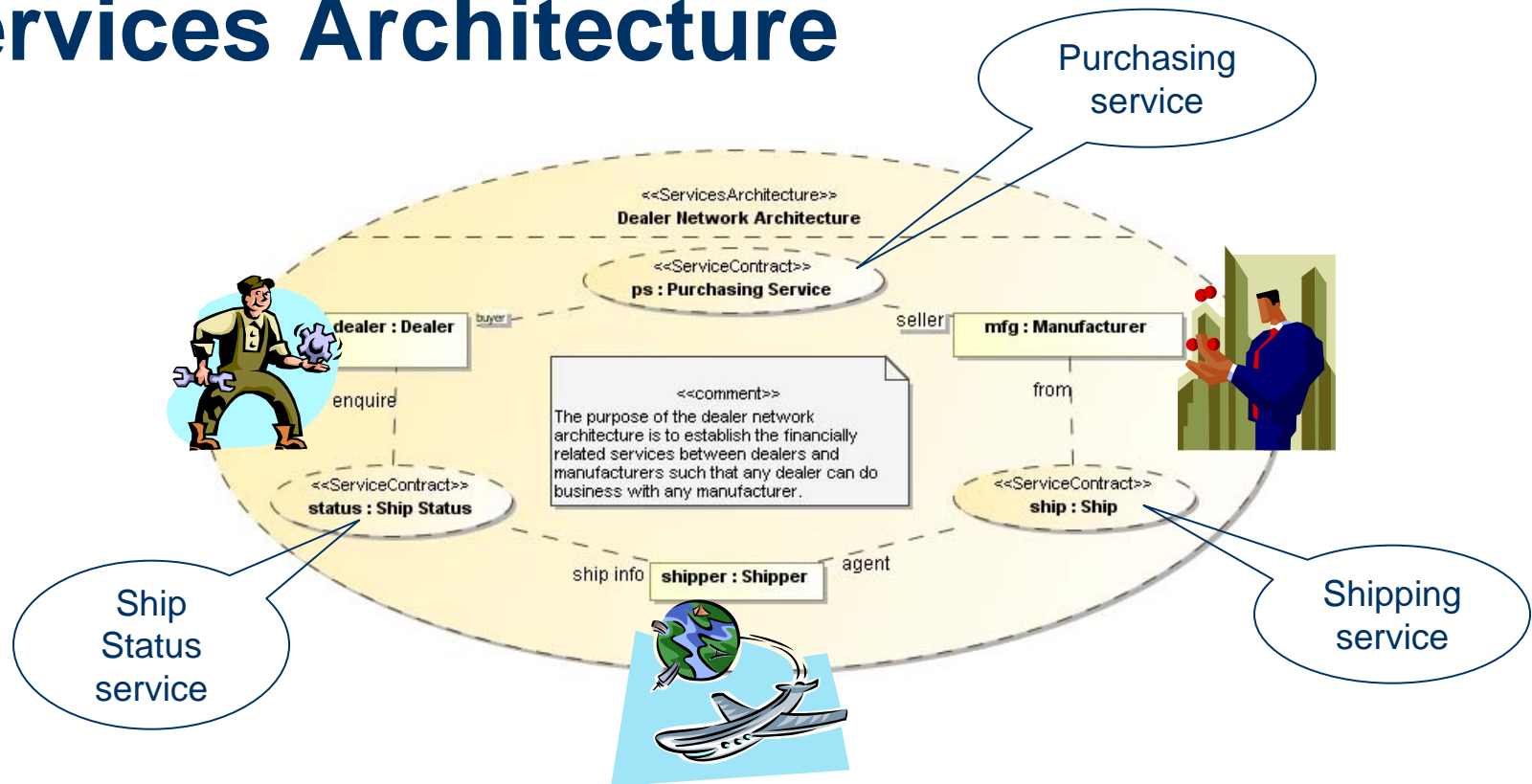
A ServicesArchitecture (or SOA) is a network of participant roles *providing* and *consuming services* to fulfill a purpose. The services architecture defines the requirements for the types of participants and service realizations that fulfill those roles.

The services architecture puts a set of services in context and shows how participants work together for a community or organization without required process management.

Inside the Seller/Manufacturer

Order

Conformation

Shipped

Ship Req

Shipped

Delivered

Order Processing

Service

Accounting
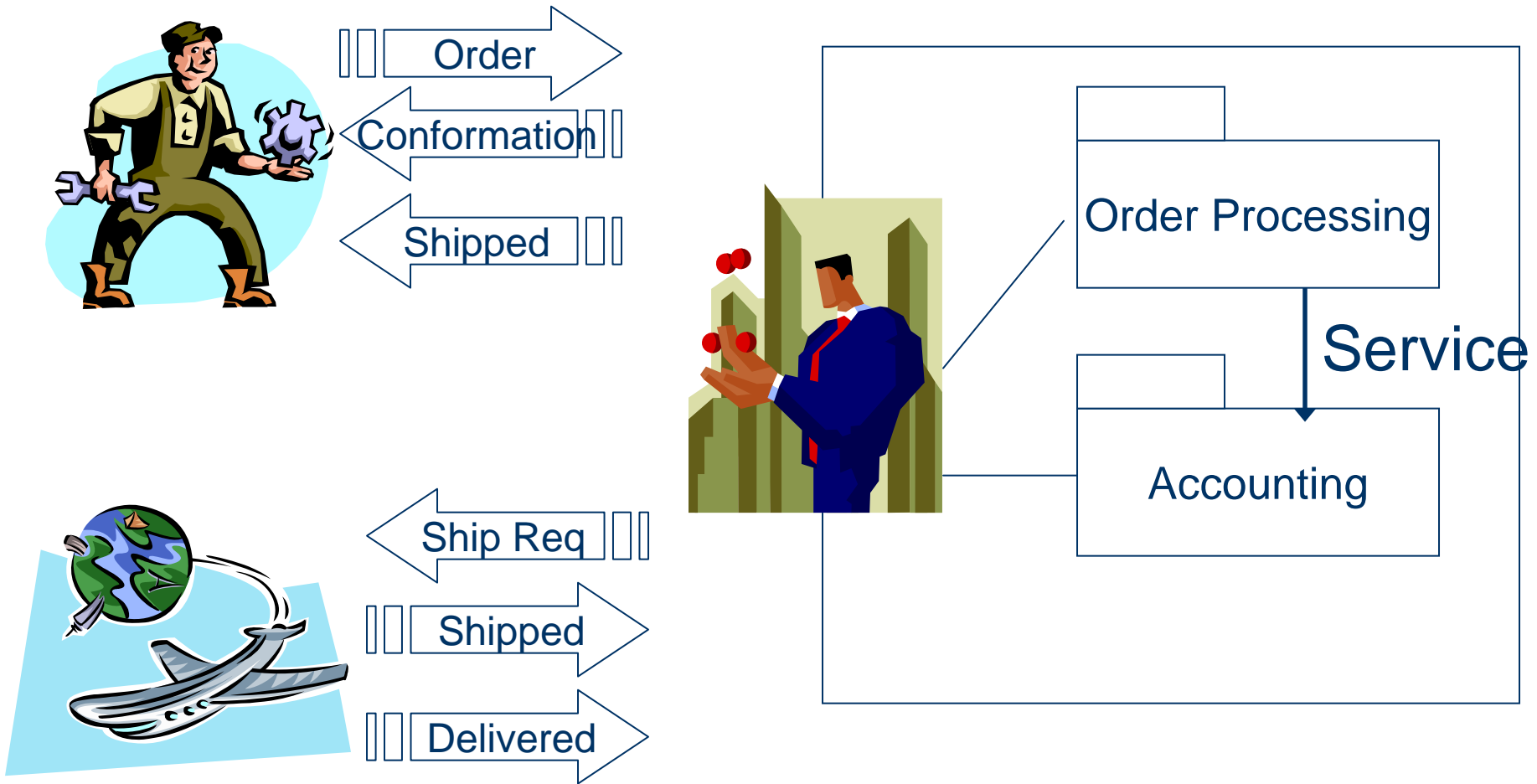
ServiceContract

A ServiceContract defines the terms, conditions, interfaces and choreography that interacting participants must agree to (directly or indirectly) for the service to be enacted - the full specification of a service which includes all the information, choreography and any other "terms and conditions" of the service. A ServiceContract is binding on *both* the providers and consumers of that service. The basis of the service contract is also a UML collaboration that is focused on the interactions involved in providing a service. A participant plays a role in the larger scope of a ServicesArchitecture and also plays a role as the provider or user of services specified by ServiceContracts

Service Contract
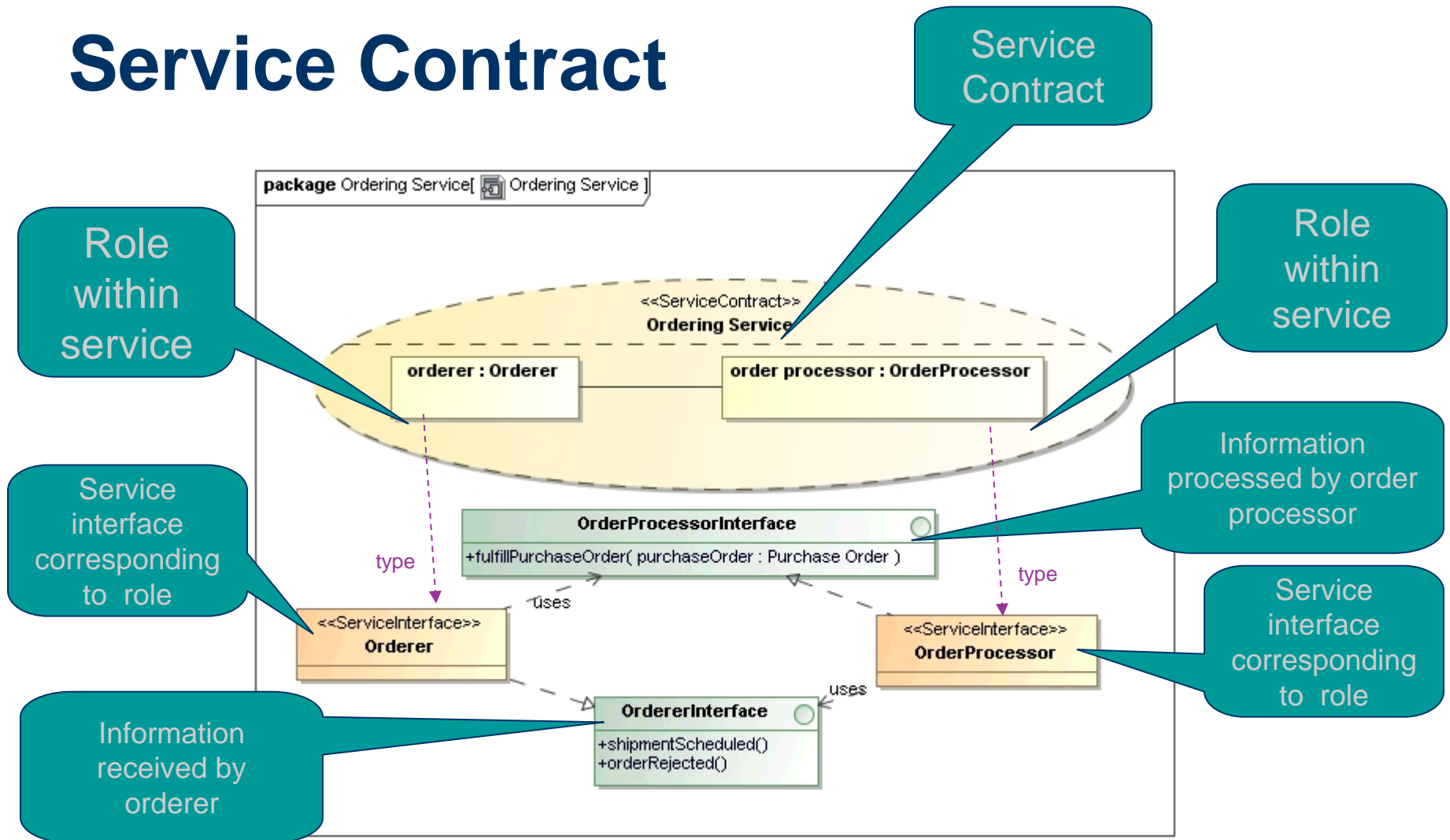
The service contract specifies the details of the service – what information, assets and responsibilities are exchanged and under what rules

SINTEF

ICT

Simple Protocol Choreography for Ordering Service Contract
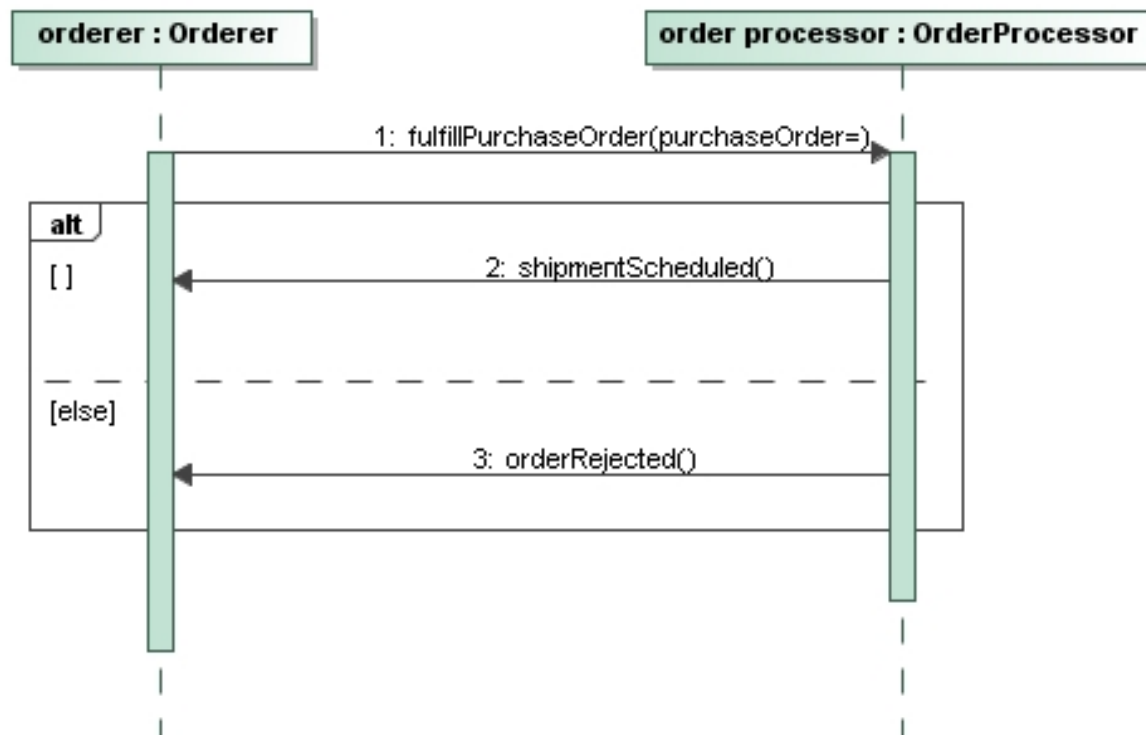
Could also be specified in BPMN, in principle

Service Modelling with SoaML Port/Connector models – extending UML 2.0 composite structure models

Service ports and Service Participants
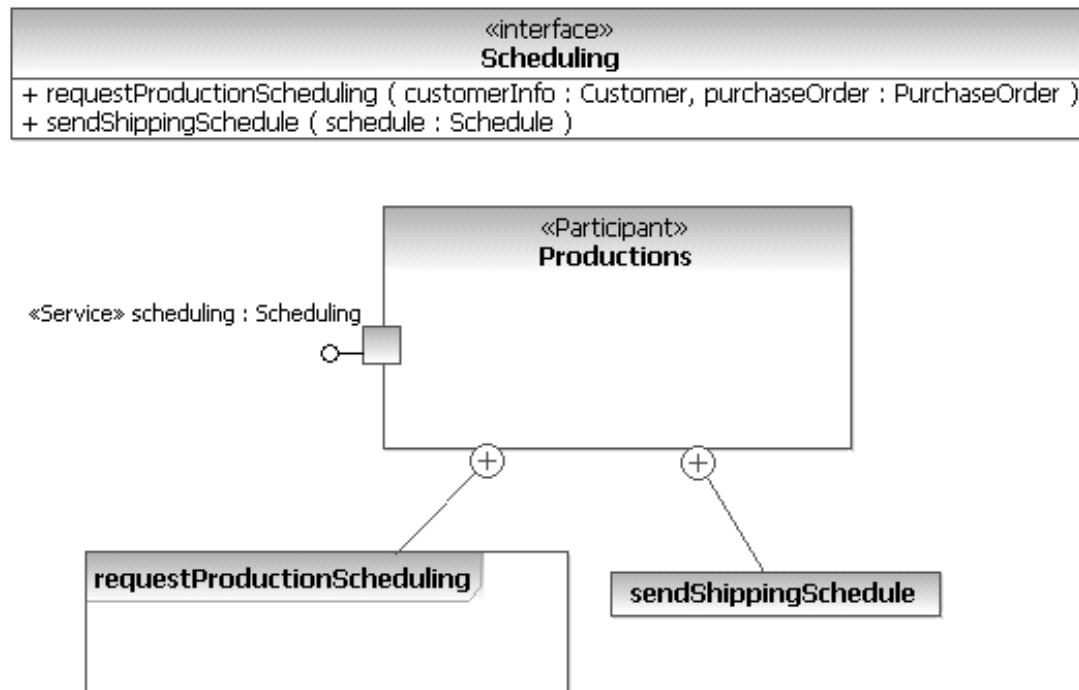
A Service port s the offer of a service by one participant to others using well defined terms, conditions and interfaces. A Service port defines the connection point through which a Participant offers its capabilities and provides a service to clients.
A Service port is a mechanism by which a provider Participant makes available services that meet the needs of consumer requests as defined by ServiceInterfaces, Interfaces and ServiceContracts. A Service port is represented by a UML Port on a Participant stereotyped as a «Service, .

ServiceInterface

a ServiceInterface can be the type of a service port. The service interface has the additional feature that it can specify a bi-directional service – where both the provider and consumer have responsibilities to send and receive messages and events. The service interface is defined from the perspective of the service provider using three primary sections: the provided and required Interfaces, the ServiceInterface class and the protocol Behavior.
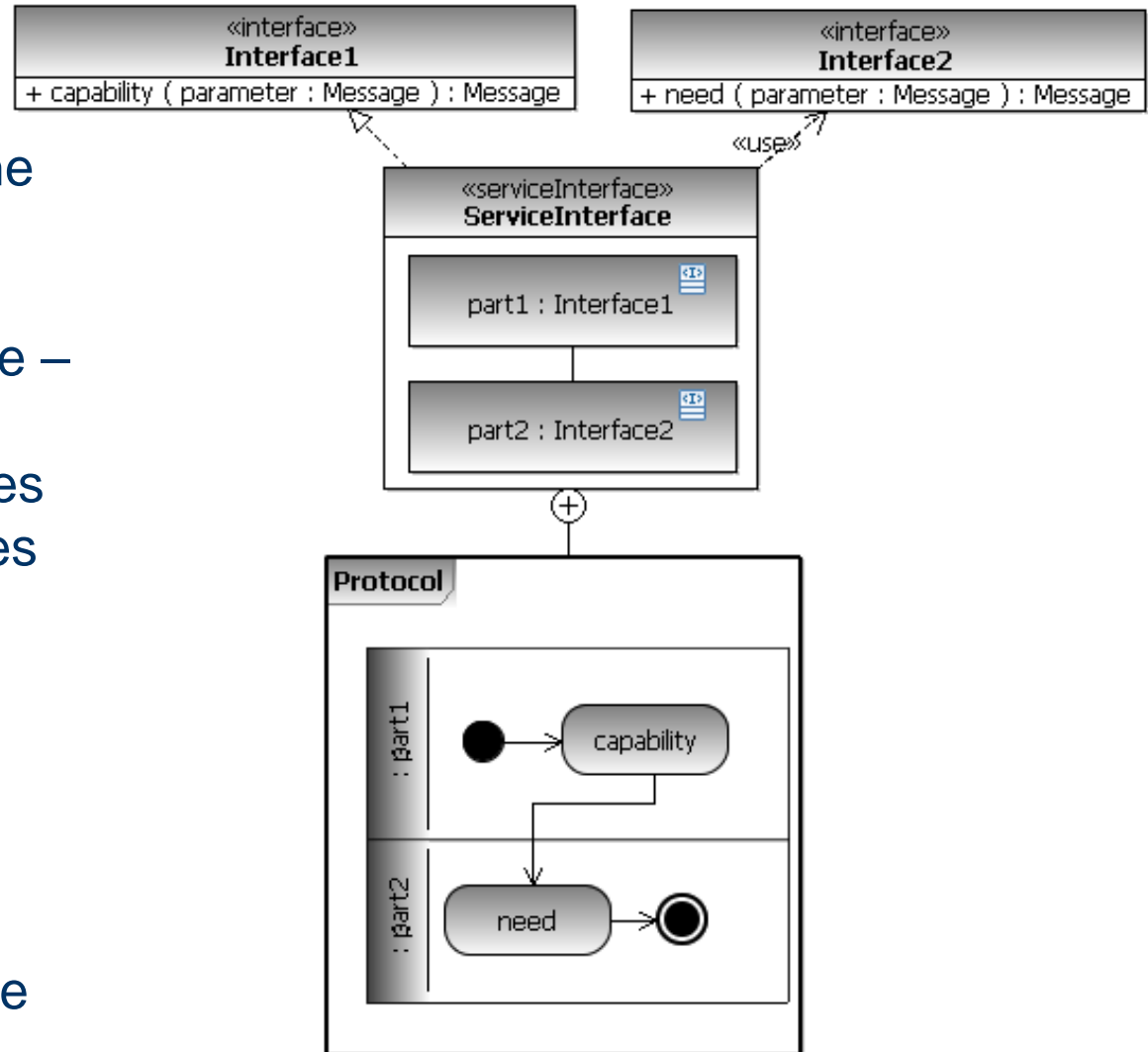
Participant with Service and Request ports
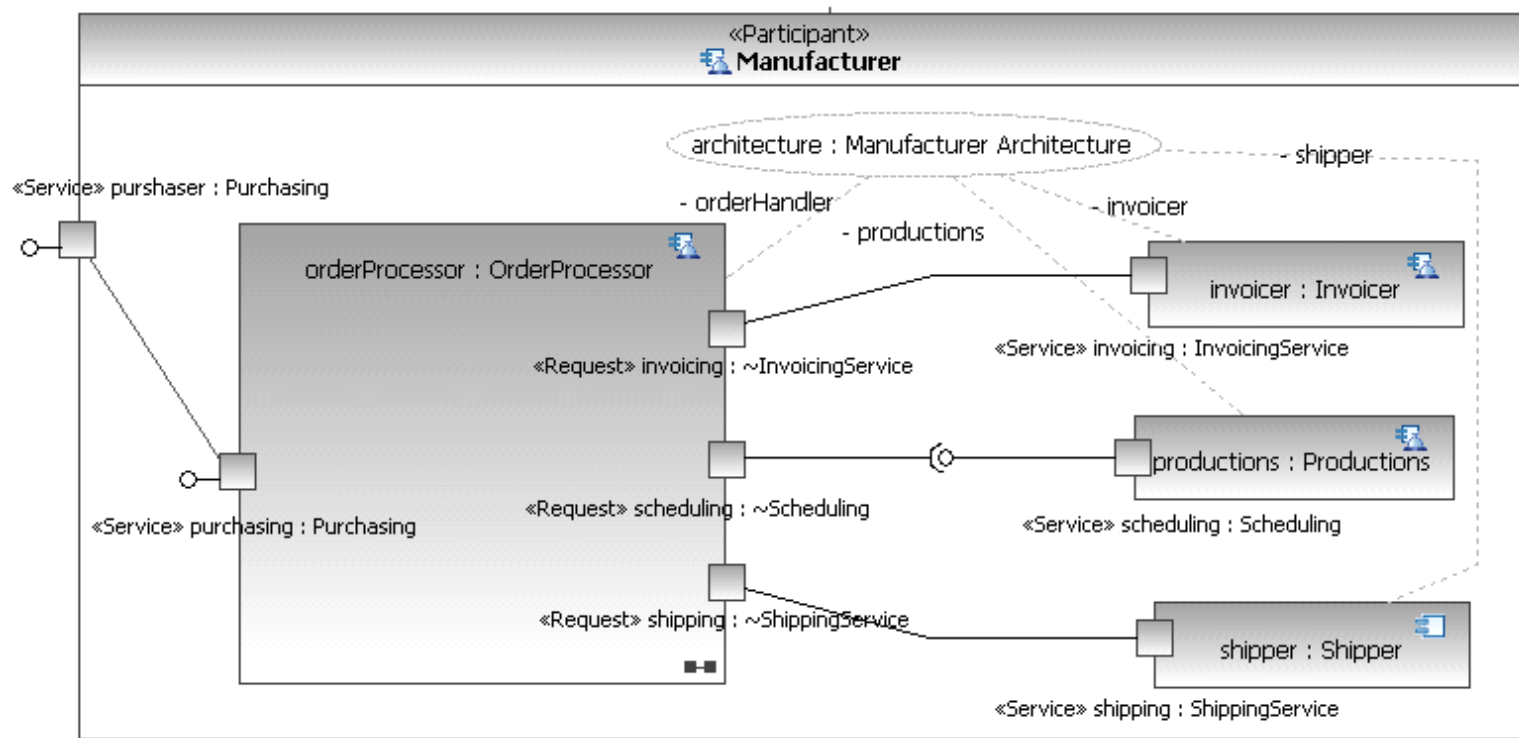
The type of a Request port is also a ServiceInterface, or UML Interface, as it is with a Service port The Request port is the conjugate of a Service port in that it defines the use of a service rather than its provision. This will allow us to connect service providers and consumers in a Participant.

SHAPE Service Variability Process
Overview
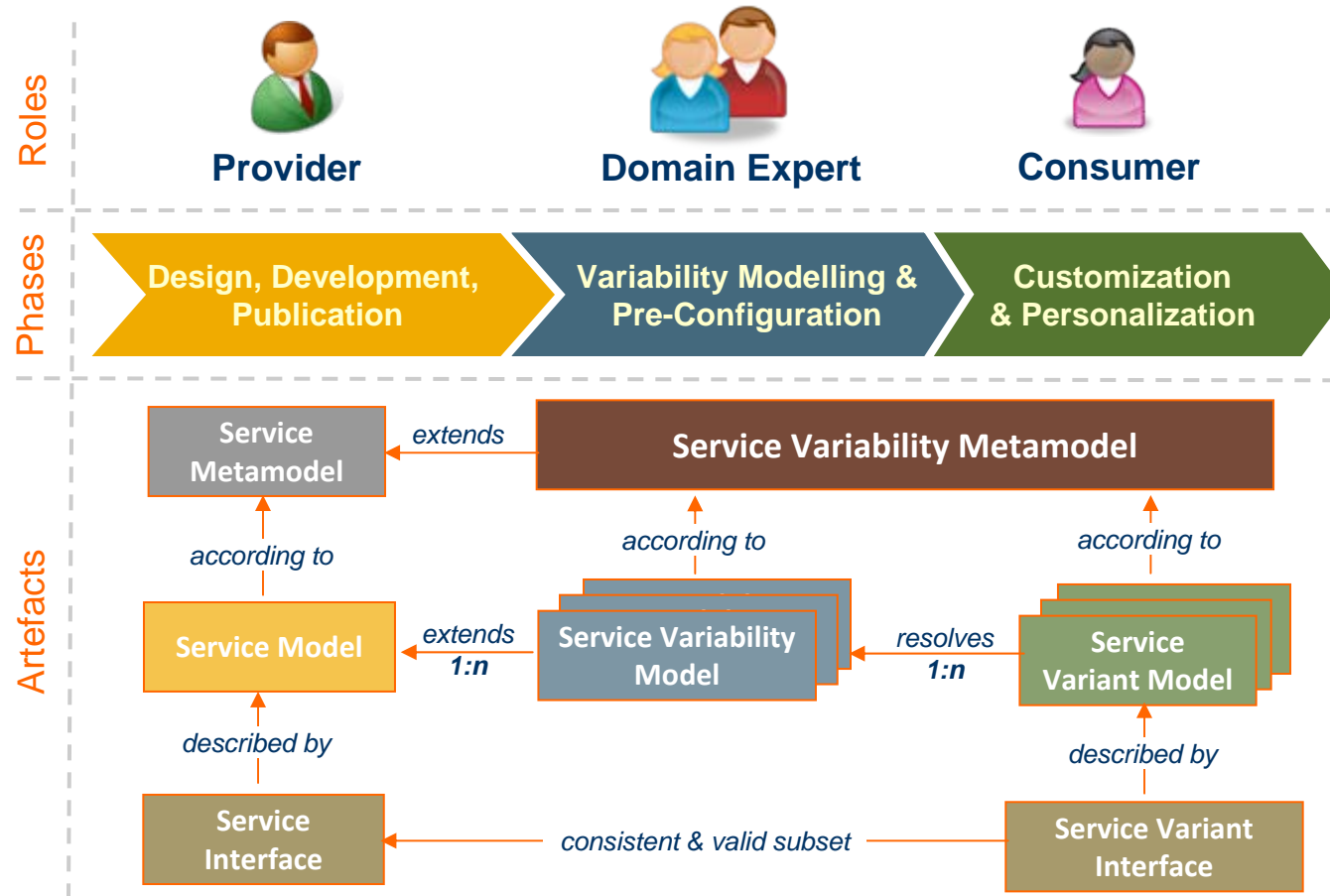
REuse and Migration of legacy applications to Interoperable Cloud Services

REMICS Overall process – The ADM, (Architecture Driven Modernisation), "SEI" Horse shoe
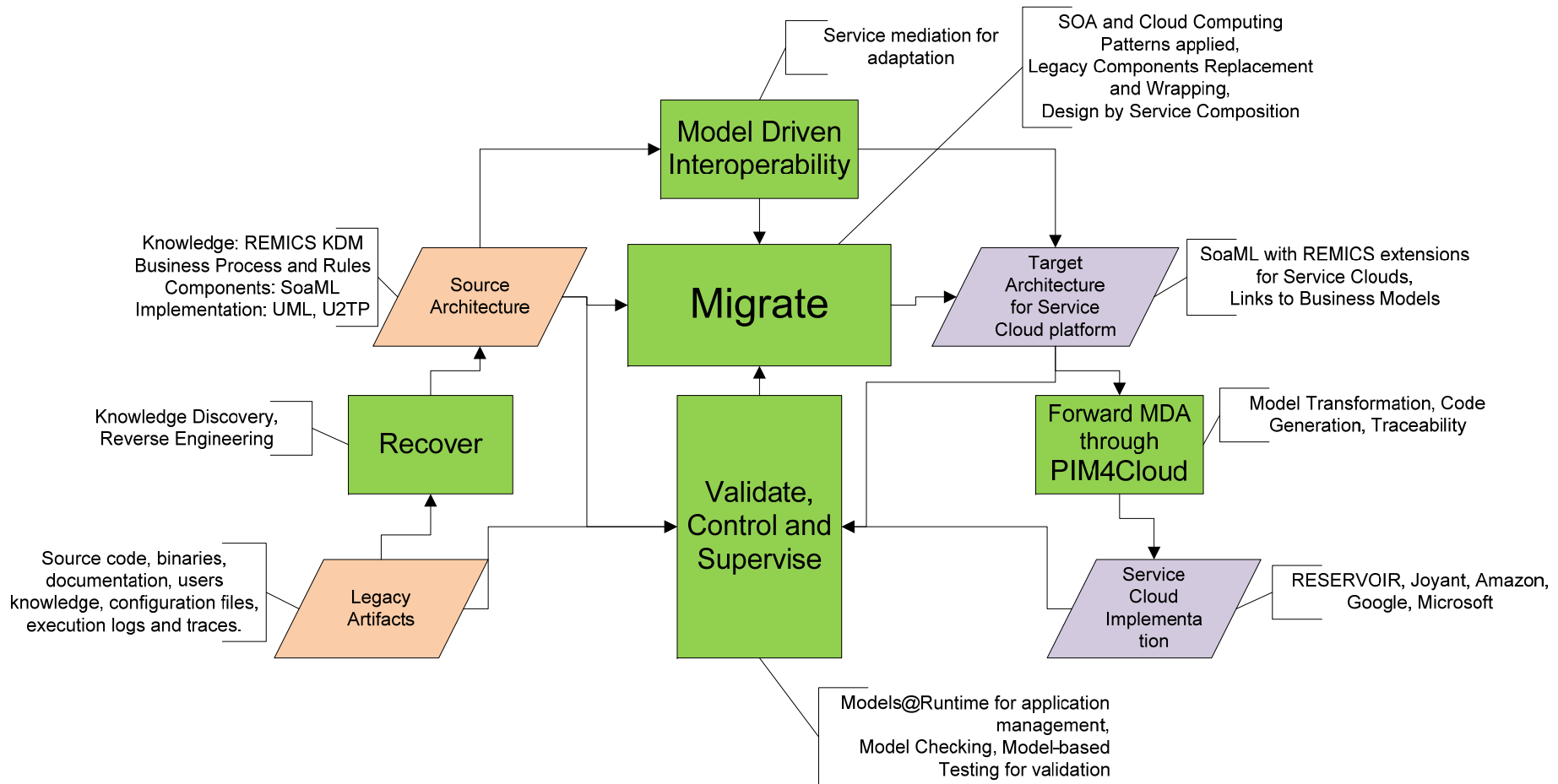
REMICS Metamodel extensions
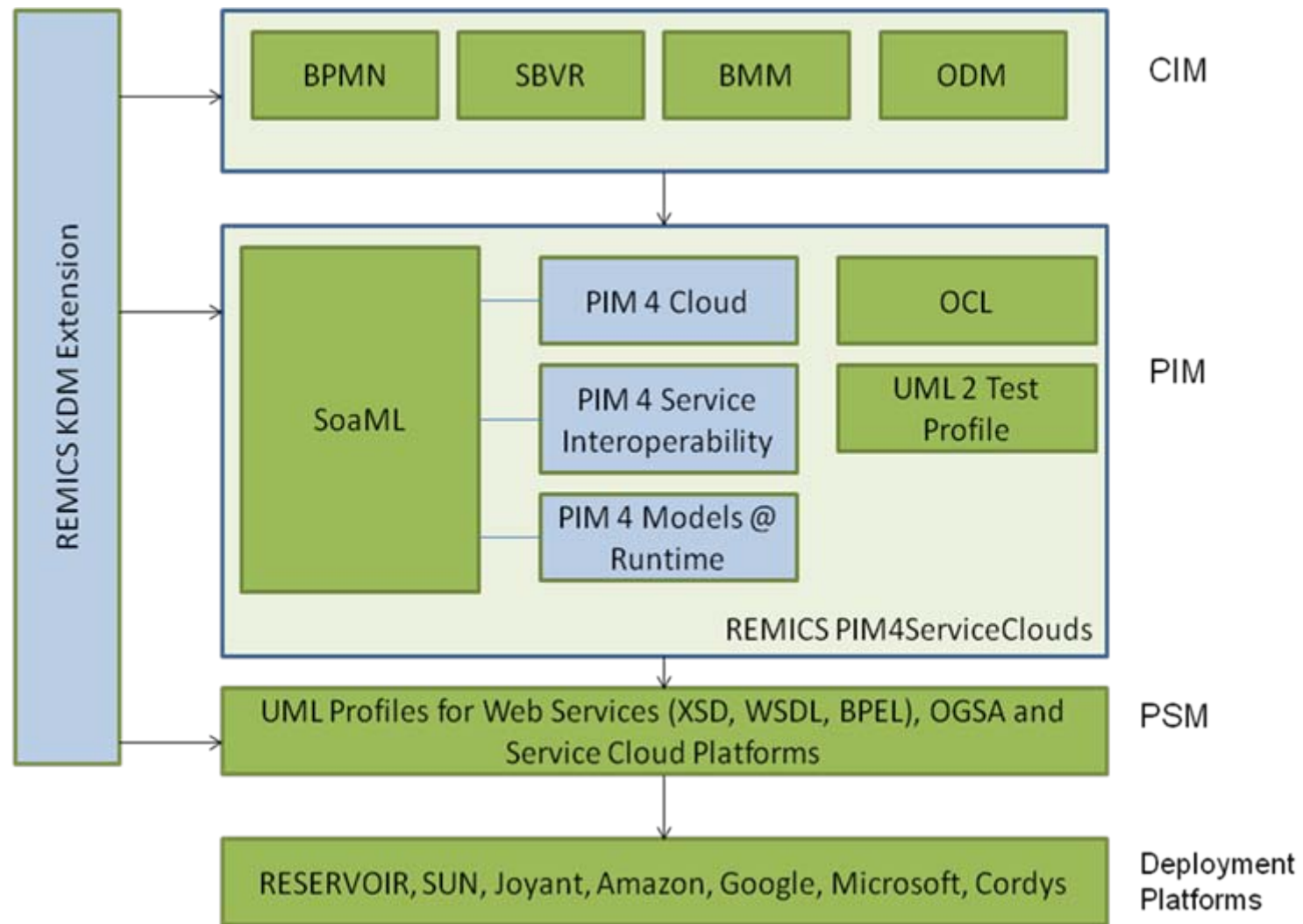
Model Evolution and Variability

Architecture model of a simple database server
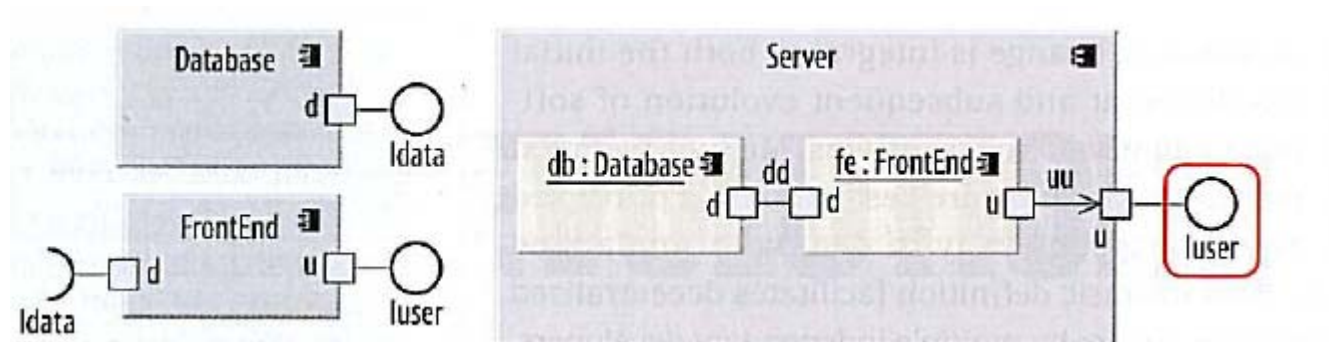
From article: Evolving Architectural Descriptions of Critical Systems, Mens/Magee/Rumpe, IEEE Computer, May 2010

Resemblance: Architecture description of managed database server
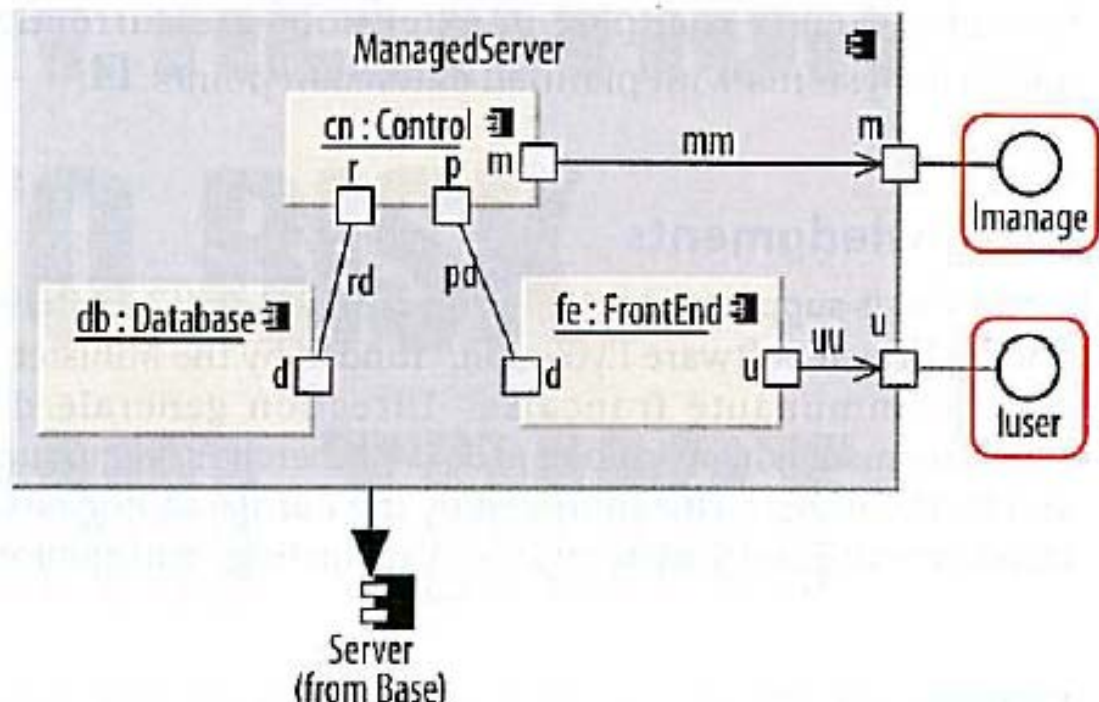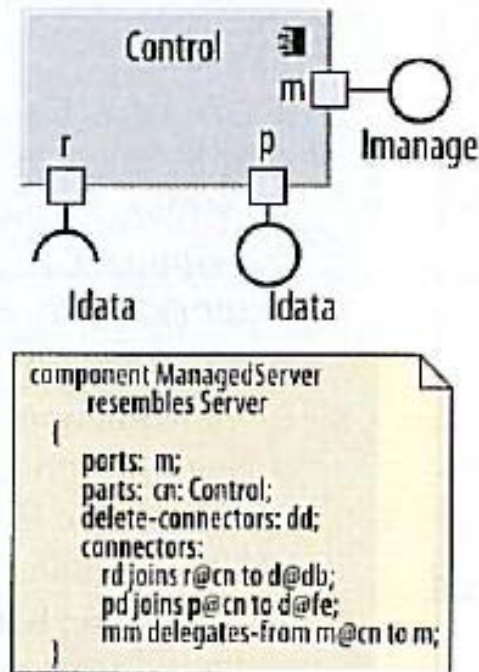
Replacement: Replacing the Database component

component Database' implementation-class ImprovedDatabase
resembles Database replaces Database
{ }

Database
(replaces
Base :: Database)   d

Idata

Server
(c)   (from Base)

SINTEF
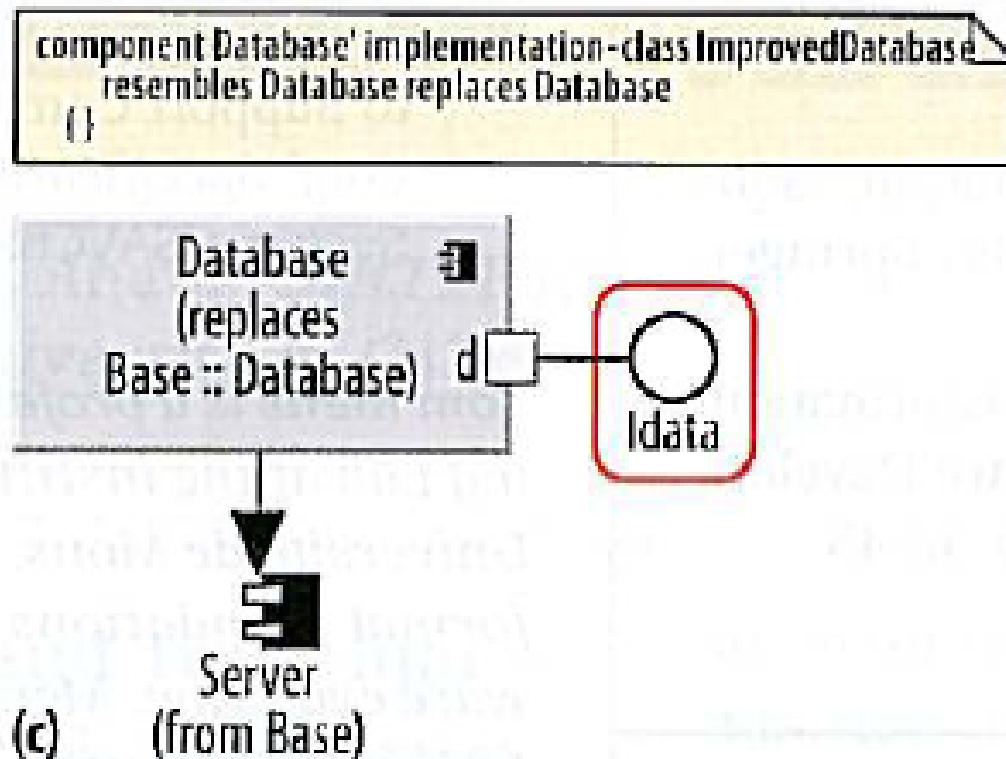
CVL overview and terms
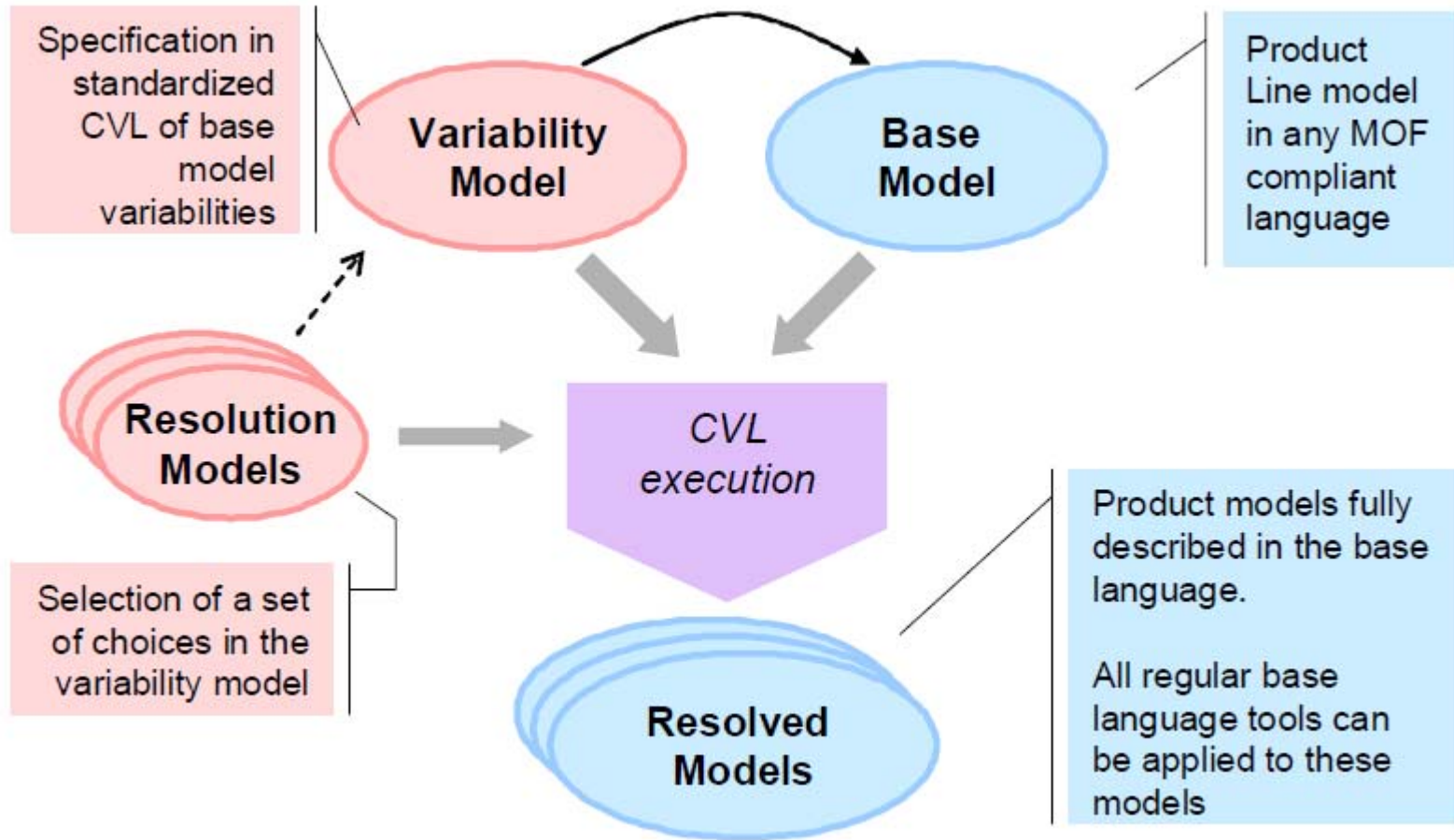
Applying CVL

variabilitymodeling.org

SINTEF Variability modeling tool

DIVA Adaptation model

CVL RFP Timetable

Event or Activity	Actual Date
Preparation of RFP by TF	*September 2009*
RFP placed on OMG document server	*September 2009*
Approval of RFP by Architecture Board *Review by TC*	*December 2009*
TC votes to issue RFP	*December 2009*
LOI to submit to RFP due	*May 2010*
Initial Submissions due and placed on OMG document server ("Three week rule")	*August 2010*
Voter registration closes	
Initial Submission presentations	*September 2010*
Preliminary evaluation by TF	*October 2010*
Revised Submissions due and placed on OMG document server ("Three week rule")	*February 2011*
Revised Submission presentations	*March 2011*
Final evaluation and selection by TF *Recommendation to AB and TC*	*June 2011*
Approval by Architecture Board *Review by TC*	*June 2011*
TC votes to recommend specification	*June 2011*
BoD votes to adopt specification	*June 2011*