Towards an Interoperability Framework for Model-Driven Development of Software Systems

INTEROP-ESA'05
Session IT1 : Industrial Track

Brian Elvesæter[1], Axel Hahn[2], **Arne-Jørgen Berre**[1], Tor Neple[1]

[1] SINTEF ICT, P. O. Box 124 Blindern, N-0314 Oslo, Norway
{brian.elvesater, arne.j.berre, tor.neple}@sintef.no
[2] Wirtschaftsinformatik Universität Oldenburg, D-26111 Oldenburg, Germany
hahn@wi-ol.de

Background

- Research sponsored by
  - ATHENA IP
    - http://www.athena-ip.org/
  - INTEROP NoE
    - http://www.interop-noe.org/

- Challenges
  - To address interoperability for enterprise applications and software.
  - Research focus on Model-Driven Development (MDD), in particular OMG's Model Driven Architecture® (MDA®)
- Interoperability Framework for MDD
  - To provide guidance on how MDD should be applied to address interoperability.

Motivation

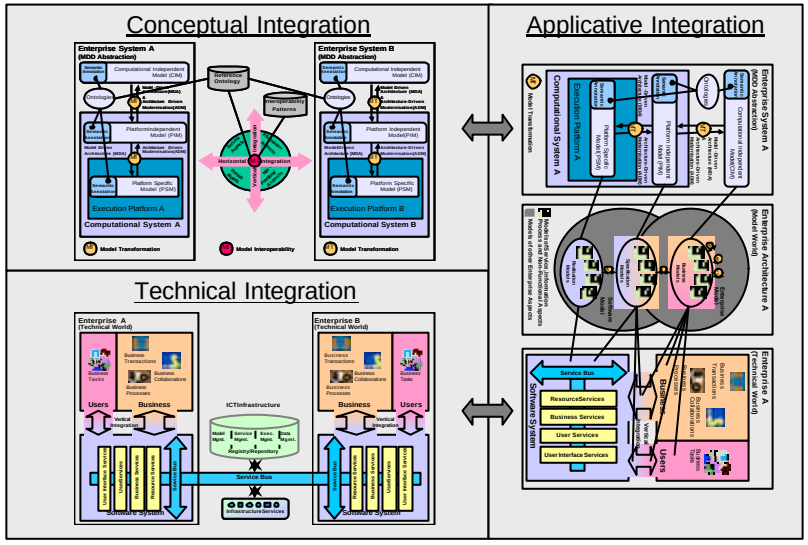
- We believe that there is a need for an interoperability framework that provides guidance on how MDD should be applied to address interoperability.

- The interoperability framework integrates principles of model-driven, service-oriented and adaptive architectures:
  - Model-driven architectures focus on design-time aspects of system engineering. Model-driven development methodologies describe how to develop and utilise (visual) models as an active aid in the analysis, specification, design and implementation phases of an ICT system.
  - Service-Oriented Architecture (SOA) specifies systems composed of services offered by various service providers, which provides the basis for supporting new business models, such as “virtual organisations”.
  - Adaptive interoperability architectures focus on run-time aspects of system engineering. Agent and P2P technologies enrich an ICT system with dynamic and adaptive qualities.

Interoperability Framework

- The interoperability framework itself is structured according to three main integration areas defined in ATHENA:

1. **Conceptual integration** which focuses on concepts, meta-models, languages and model relationships to systemise software model interoperability.
2. **Technical integration** which focuses on the software development and execution environments.
3. **Applicative integration** which focuses on methodologies, standards and domain models. It provides us with guidelines, principles and patterns that can be used to solve software interoperability issues.

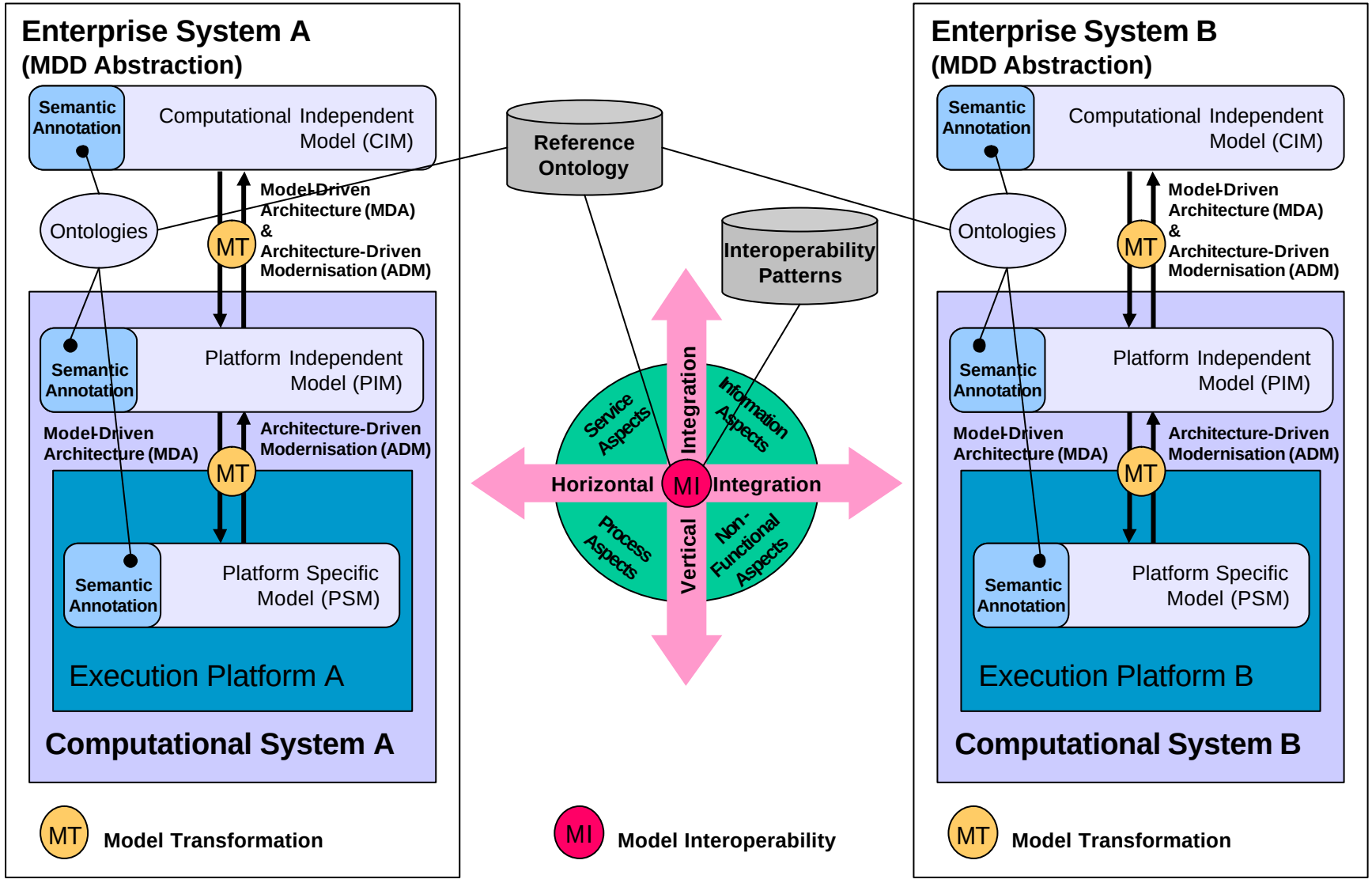


- For each of these three areas we developed a reference model to describe and support the application of model-driven development of software systems.

Conceptual Integration – Overview

- Developed from a MDD point of view focusing on the enterprise applications and software system.
- A Computation independent model (CIM) corresponds to a view defined by a computation independent viewpoint. It describes the business context and business requirements for the software system(s).
- A platform independent model (PIM) corresponds to a view defined by a platform independent viewpoint. It describes software specifications independent of execution platforms.
- A platform specific model (PSM) corresponds to a view defined by a platform specific viewpoint. It describes the realisation of software systems.

Conceptual Integration: Reference Model



Conceptual Integration: System Aspects

- We have identified four categories of system aspects where specific software interoperability issues can be addressed. These four aspects can be addressed at all three CIM, PIM and PSM levels.

1. **Service aspects:** Services are an abstraction and an encapsulation of the functionality provided by an autonomous entity,.
2. **Information aspects:** Information aspects are related to the messages or structures exchanged, processed and stored by software systems or software components.
3. **Process aspects:** Processes describe sequencing of work in terms of actions, control flows, information flows, interactions, protocols, etc.
4. **Non-functional aspects:** Extra-functional qualities that can be applied to services, information and processes.

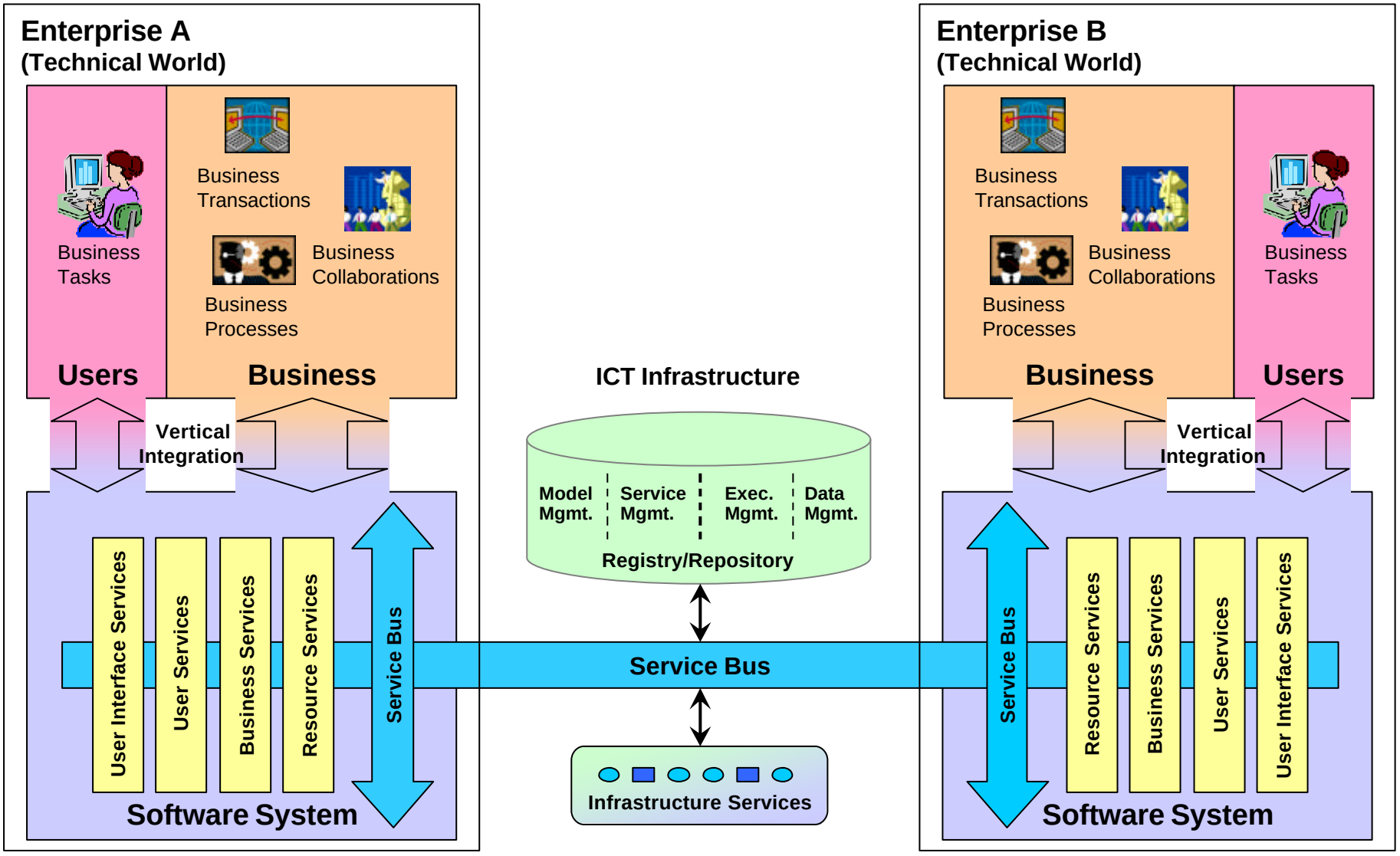
- We have been studying the following metamodels.

- CIM level
  - Business Process Definition Metamodel (BPDM)
  - Software Process Engineering Metamodel (SPEM)
  - POP* (being defined within ATHENA)
- PIM level
  - EDOC – UML Profile for Enterprise Distributed Object Computing Specification
  - QoS – UML Profile for Modelling Quality of Services and Fault Tolerance Characteristics and Mechanisms
- PSM level
  - Web Services
  - Web Services Business Process Execution Language (WSBPEL)
  - Agents Architectures
  - Peer2Peer Architectures

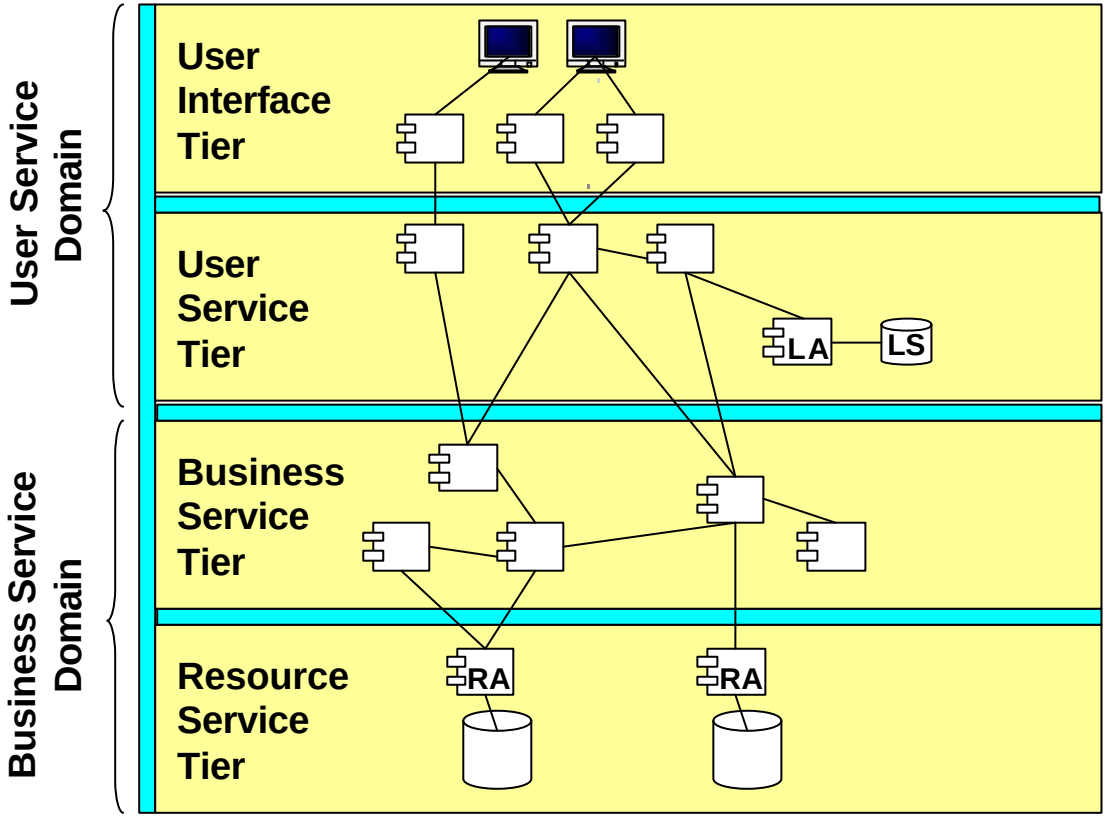
Technical Integration – Overview

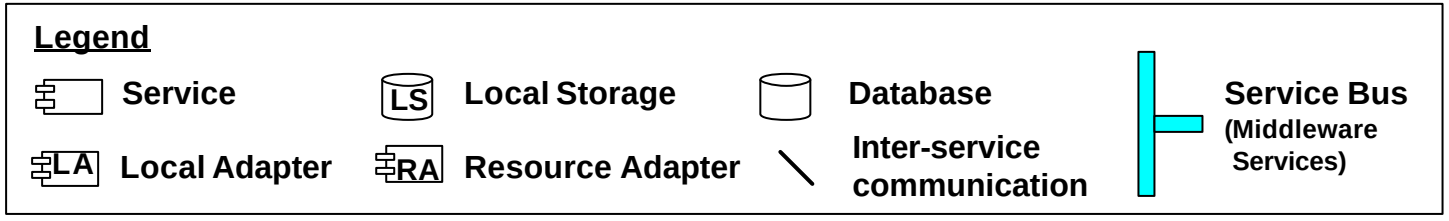
- Developed from a service-oriented point of view where a software system provides a set of services required by the businesses and users of the enterprise.
- The architecture of the enterprise applications and software systems can be described according to a 4-tier reference architecture where each tier provides different software services required by the enterprise.
- The software system itself is coupled to a service bus that provides the necessary communication infrastructure.
- Infrastructure services such as composition, mediation, matchmaking and transformation that enables interoperability between software systems should be provided.
  - *Model repository* for managing models of various kinds.
  - *Service registry* for managing naming, directory and location of services
  - *Execution repository* for managing information and state needed in the execution of software services and processes
  - *Data repository* for managing results and traces of the executions.

Technical Integration: Reference Model



Technical Integration: 4-tier reference architecture

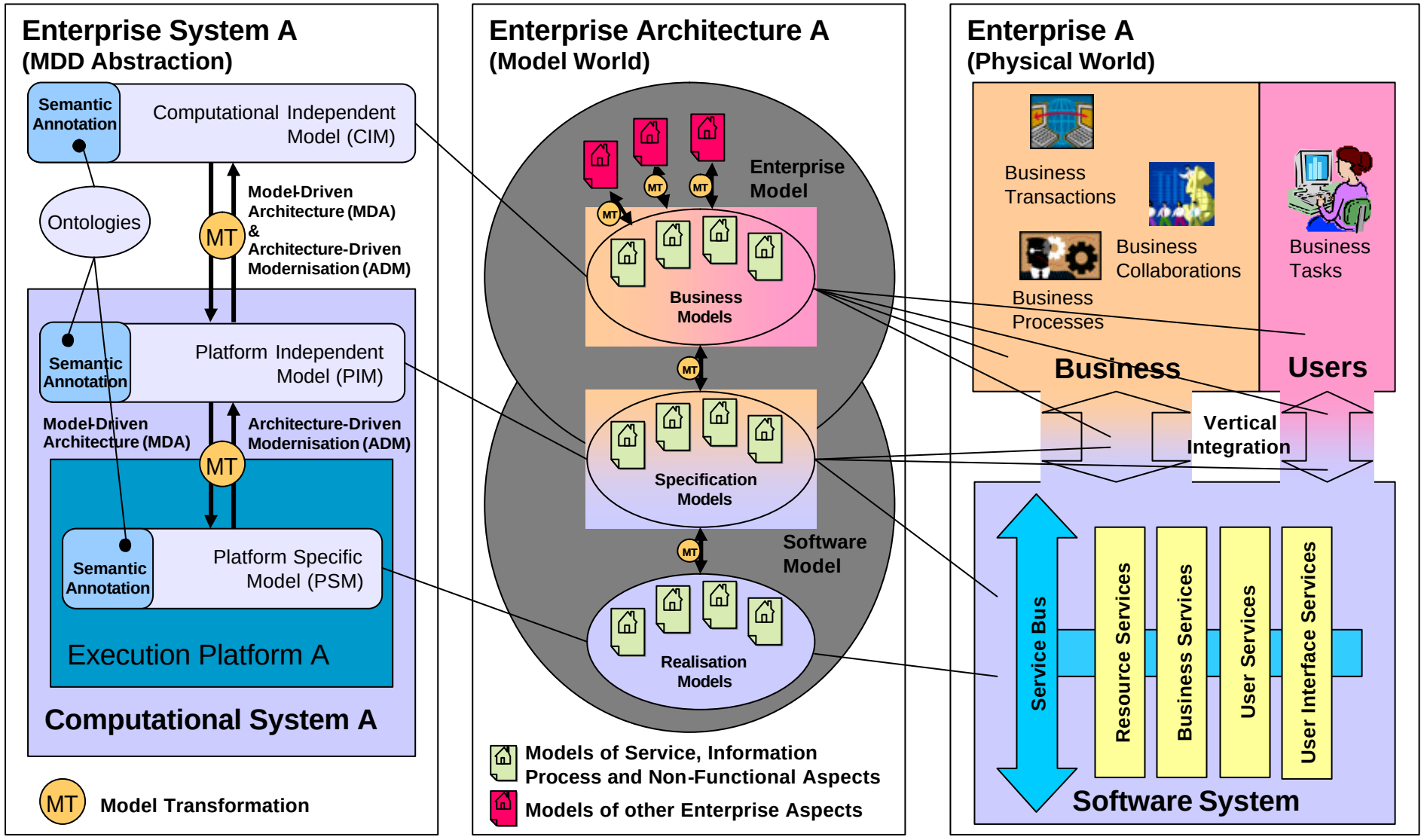




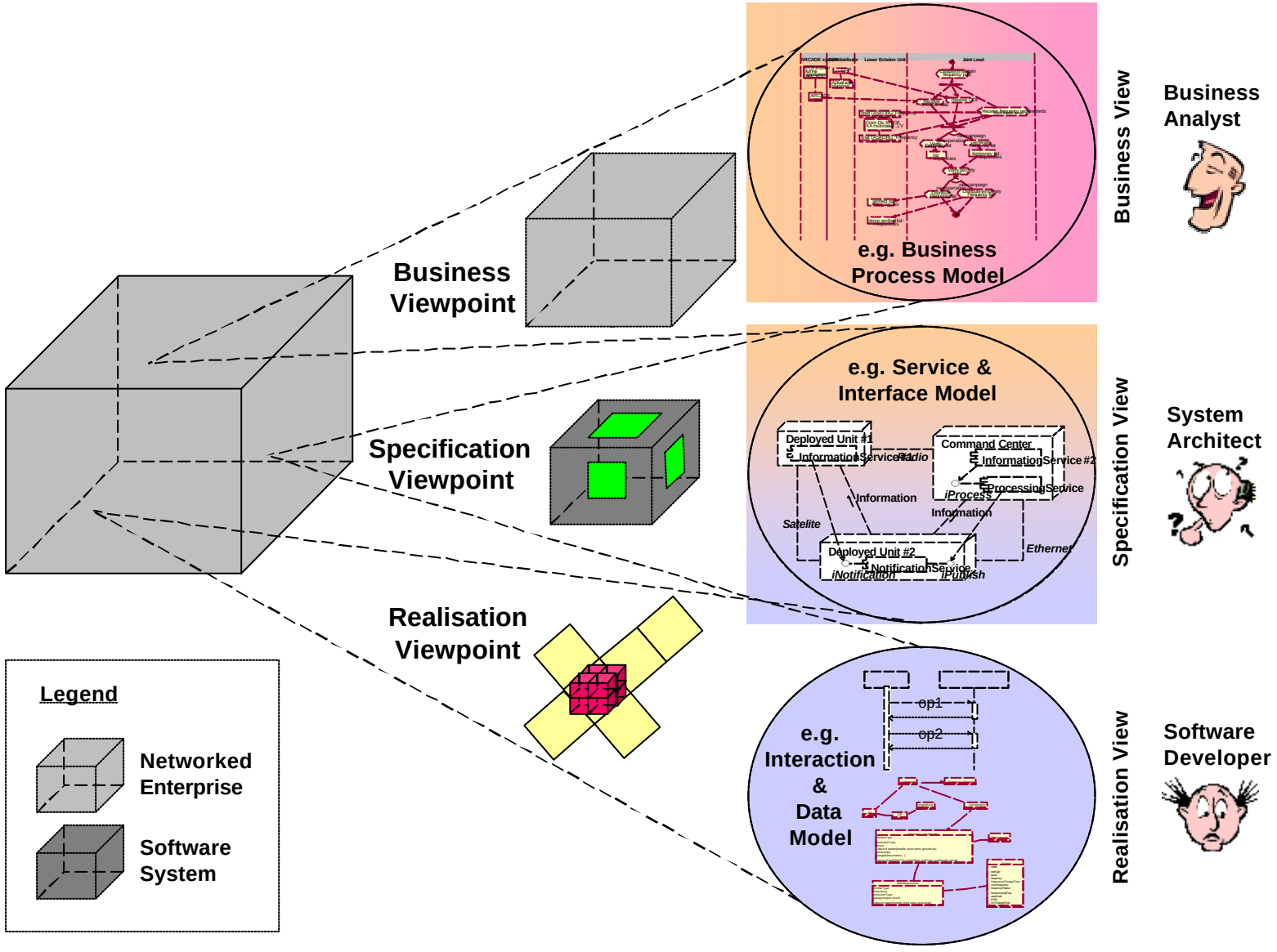
Applicative Integration – Overview

- Developed based on work related to enterprise architecture frameworks and software architecture frameworks.
- Enterprise and software models can be related in a holistic view, regardless of modelling language formalisms, by the use of meta-models.
- The MDD methodology needs to follow a structured approach where interoperability requirements from business operations in a networked enterprise drive the development of software solutions.
- An enterprise model describes a set of enterprise aspects, which includes descriptions of the business operations of the business models.
- Software models describe how software systems are used to support the businesses of an enterprise.
- All these models should include descriptions of the four system aspects identified in the reference model for conceptual integration.

Applicative Integration: Reference Model



Applicative Integration: Three Basic Viewpoints



Future Work (1/2)

Conceptual Integration

- We will develop a ***metamodel for Service-Oriented Architectures*** and a corresponding ***UML 2.0 Profile*** at the PIM level.
- The metamodel will address the four system aspects identified:

1. **Service aspects**
2. **Information aspects**
3. **Process aspects**
4. **Non-functional aspects**

Applicative Integration

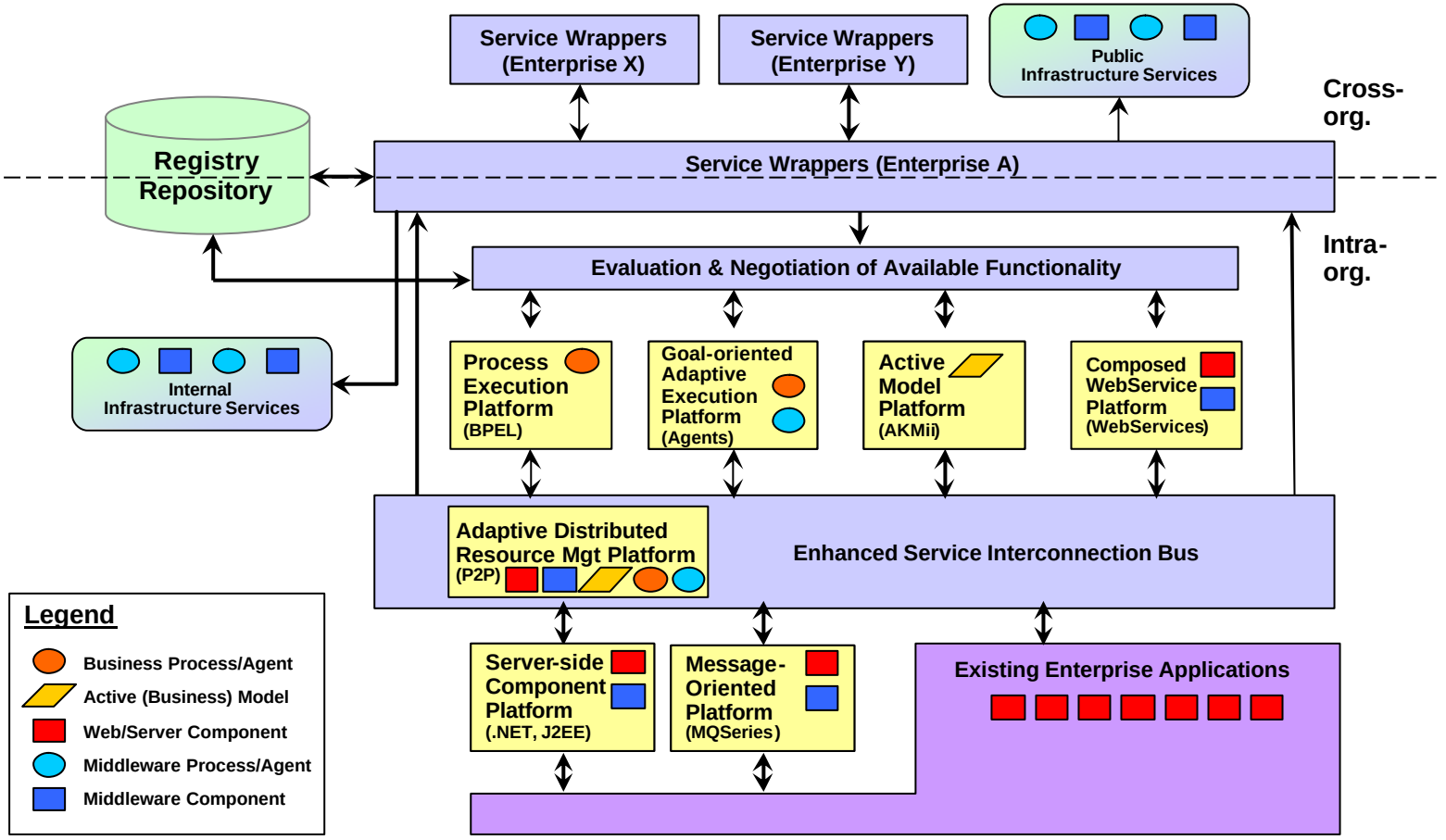
- We will develop ***interoperability profiles*** that provide specific guidelines for how to apply model-driven development of software systems in four different business domains:

1. **Supply chain management** where stable supply chains and dynamic supply networks will be considered.
2. **Collaborative product development** in which cross-functional and cross-organisational teams collaborate in product development.
3. **e-Procurement** focusing on electronic purchasing and selling of goods and services.
4. **Portfolio management** focusing on project classifications, selection, prioritisation, and resource allocation.

Future Work (2/2)

Technical Integration

- We will develop an integrated execution infrastructure that enables different platforms to interoperate in a Service-Oriented Architecture.



Questions?