Model-Driven Architecture, Processes and Methodology from the Perspective of the "Modeling Discipline"

MDA™ Implementers' Workshop:
Succeeding with Model Driven Systems
May 12th 2003
Orlando, Florida

Background for Mathet Consulting, Inc.

- Enterprise (and Federation) / Software, Architecture / Engineering
- Government, Financial, Pharmaceutical, Transportation
- Large scale modeling system development

Agenda

- Introduction
- A Problem Domain
- Model Driven Architecture (Focus on the MDA Framework – CIM/PIM/PSM, and Context and Contracts)
- Holistic Conceptualization of MDA Implementation in Enterprise, Federated and Internet Systems
- Putting It All Together

Introduction: Purpose, Goals and Objectives, Strategies for this Tutorial

Purpose / Goals / Objectives

Purpose: To illustrate and characterize the implementation and application of Model Driven Architecture in creating information systems in the context of the "Modeling Discipline", and in the context of Enterprise, Federated, and Internet Systems

Goals:

- Identify the different domains, theories and systems that make up the "Modeling Discipline"
- Identify the usages of these domains, theories and systems in enterprises, federations and the Internet
- Illustrate how MDA fits into the "Modeling Discipline" and its use in creating enterprise, federated and Internet Systems
- Stimulate questions as to whether MDA tool sets (e.g. the Reference Model) is adequately supporting modeling needs

Strategies

Scenario:

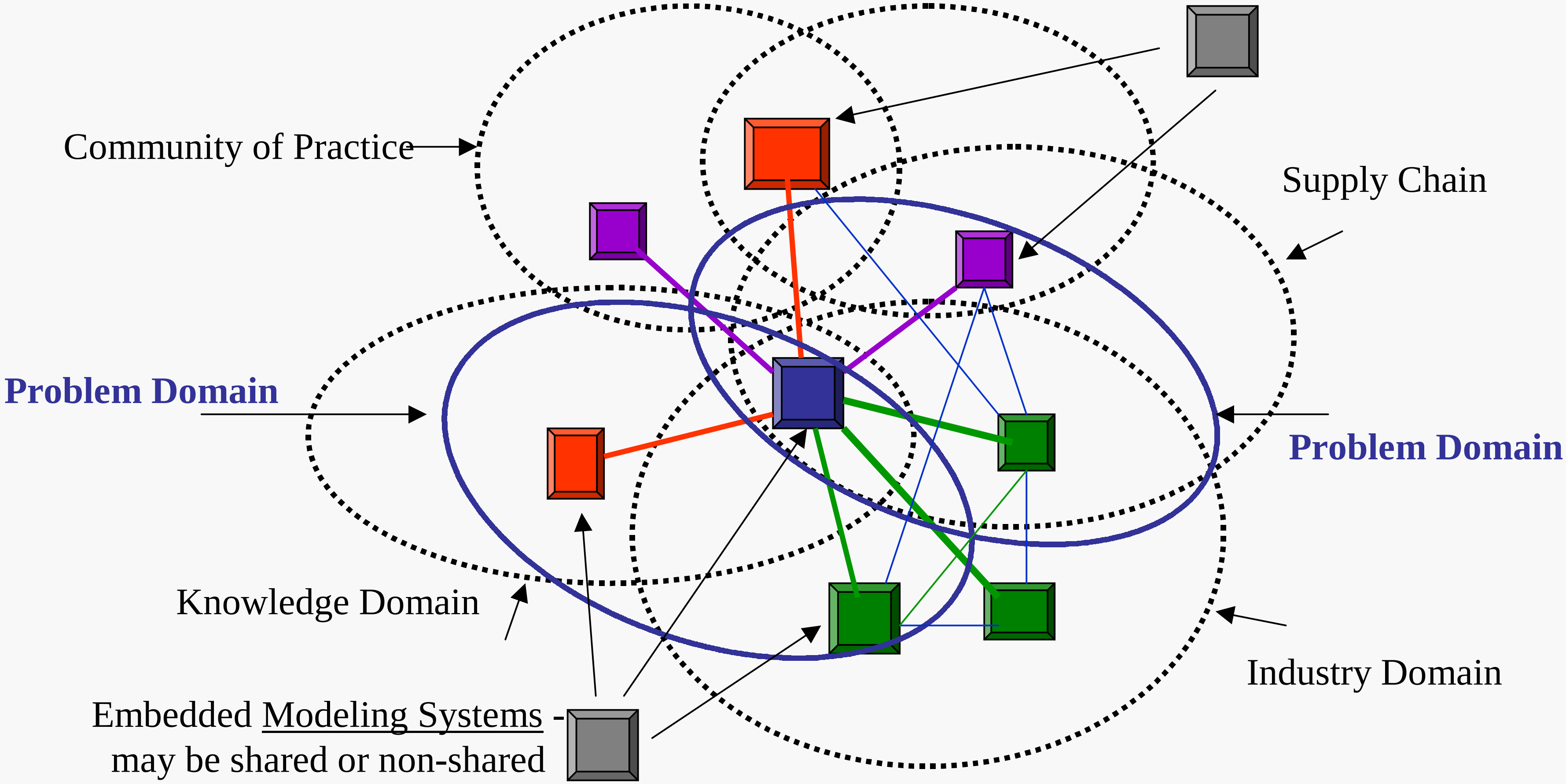
- Provide a holistic view of modeling in general and MDA in particular in the context of a creating a modeling system (<u>implementing MDA</u>) embedded in Enterprise, Federation, and Internet Systems
- Illustrate the application of the "Modeling Discipline"
- Illustrate the application of MDA in the context of the "Modeling Discipline

Expectation: The MDA architecture (e.g. the Reference Model, Language set) needs to support <u>modeling requirements</u> of modeling systems that drive systems architecture through the use of models (the application of MDA)

Expectation: The tool set which realizes and implements the "Architecture of MDA" is one component of a modeling system which is used to create new systems in a technology/platform independent way through the creation of models

A Problem Domain

The Multi-context Integration, Interoperability and Collaboration Problem Within and Between Enterprise, Federation and Internet Systems



The Multi-context Integration, Interoperability and Collaboration Problem Highlights

- Multiple contexts must be reasoned with
- New solutions increase overall complexity by increasing the number of contexts
- Systems address multiple problem/solution domains concurrently
- Problem domains have cross dependencies
- Systems are emergent, adaptive

Model Driven Architecture (MDA)™

(Focus on the MDA Framework – CIM/PIM/PSM, Reference Model, and Context and Contracts)

Model Driven Architecture

What it isn't

Process, Methodology

Technology

What it is

A general approach, strategies, to architect and engineer (information) systems

"An approach to IT system specification that separates the specification of functionality from the specification of the implementation of that functionality on a specific technology platform"
(Draft MDA Guide: www.omg.org/docs/ab/03-01-03)

An emerging knowledge domain, based in the "modeling discipline", whose universe of discourse is the engineering of information systems through models and modeling.

Model Driven Architecture -Semantics

"Model Driven Architecture"

Software
"Model Driven ∨ Architecture"

Enterprise, Federated, Internet
"Model Driven ∨ Architecture"

IT System
"Model Driven ∨ Architecture"

"Model Driven (IT System) Architecture"

May be distributed at the Enterprise, Federated, Internet Level

Sense 1: Architecture for MDA Tool Sets (a "platform" for creating system architectures)

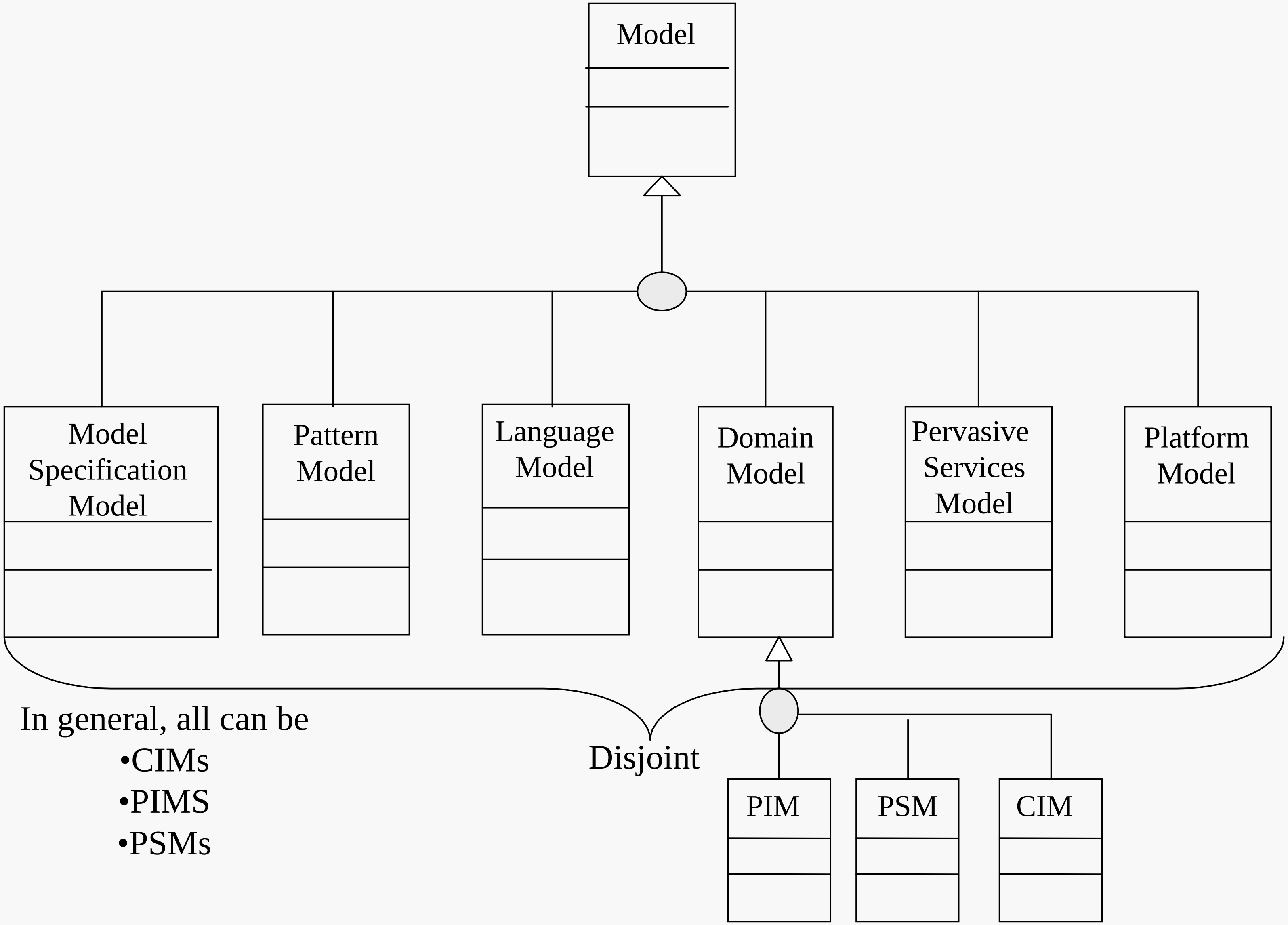
Sense 2: An (IT) system architecture created using the MDA approach and MDA tool sets

Model Driven Architecture

- Basic Terminology
  - Model: A representation/specification, expressed in one or more formal languages with defined semantics, of a part of the function, structure, and / or behavior of a system
  - Computation Independent Model (CIM): a model that is independent of computation representations
  - Platform: "a set of (sub)systems/technologies that provide a coherent set of functionality through interfaces and specified usage patterns that any sub(system) depends on the platform can use without concern for the details of how the functionality provided by the platform is implemented"
  - Platform Independent Model (PIM): a model that is computationally complete and independent of technology / platform (e.g. independent of CORBA, J2EE, etc.)
  - Platform Specific Model (PSM): a model specific to a platform
  - Platform Model (PM): A model of a platform
  - *Business Domain Model (BDM) (not part of MDA terminology)*
  - *Business Domain Independent "Platform" Model (BPM) (not part of MDA terminology)*

All definitions are taken from, or modified, from the MDA Guide (draft)
www.omg.org/docs/ab/03-01-03

Relationship between Reference Model and CIM/PIM/PSM Framework

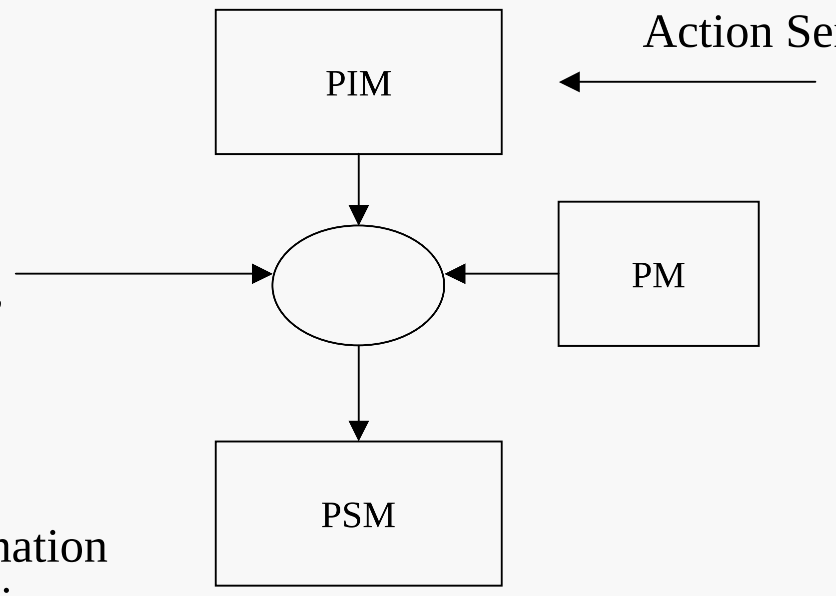


In general, all can be

- CIMs
- PIMS
- PSMs

MDA Pattern

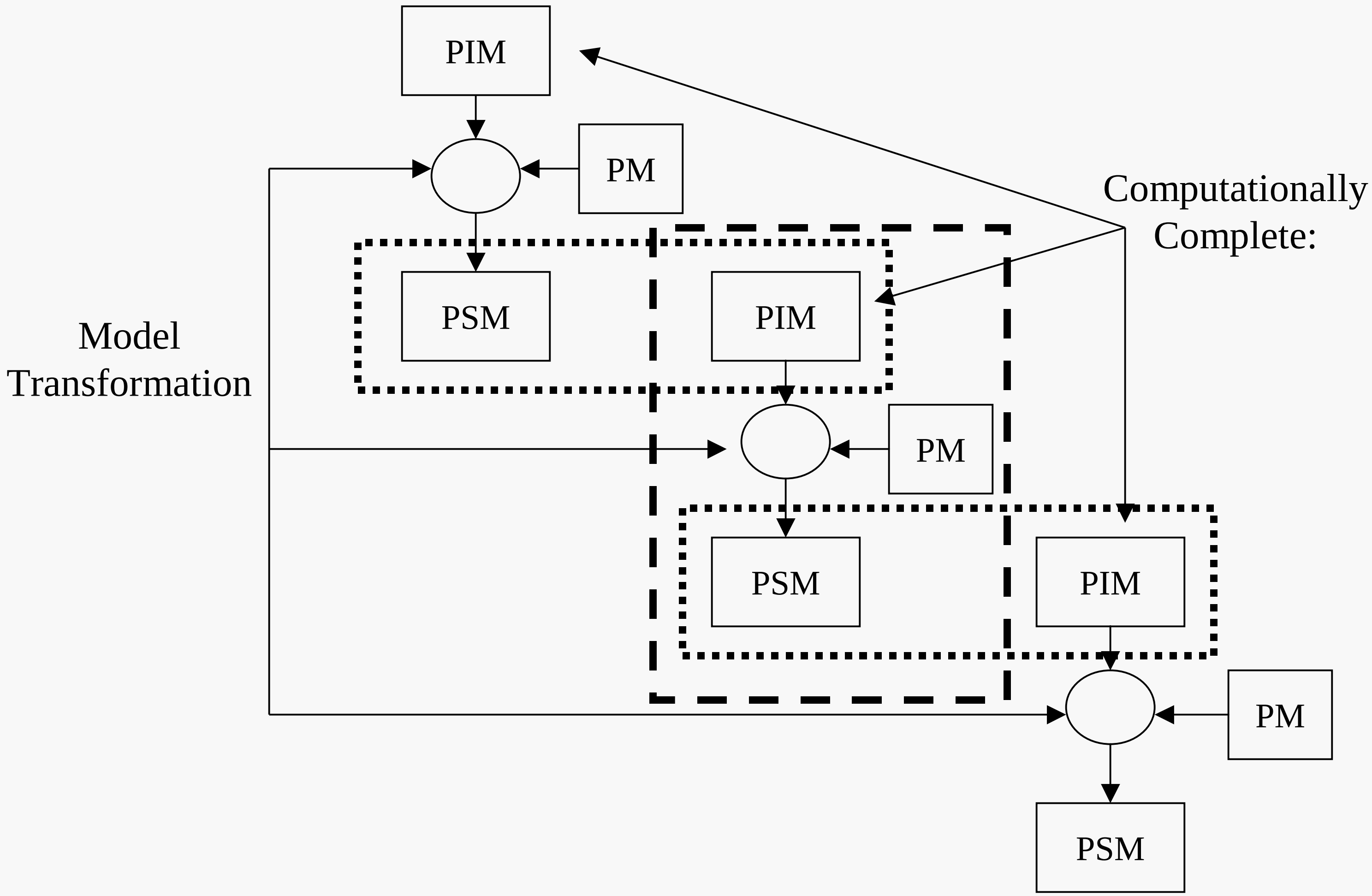
Computationally Complete:
Can be executed
(code generation / interpretation
Action Semantics / OCL)



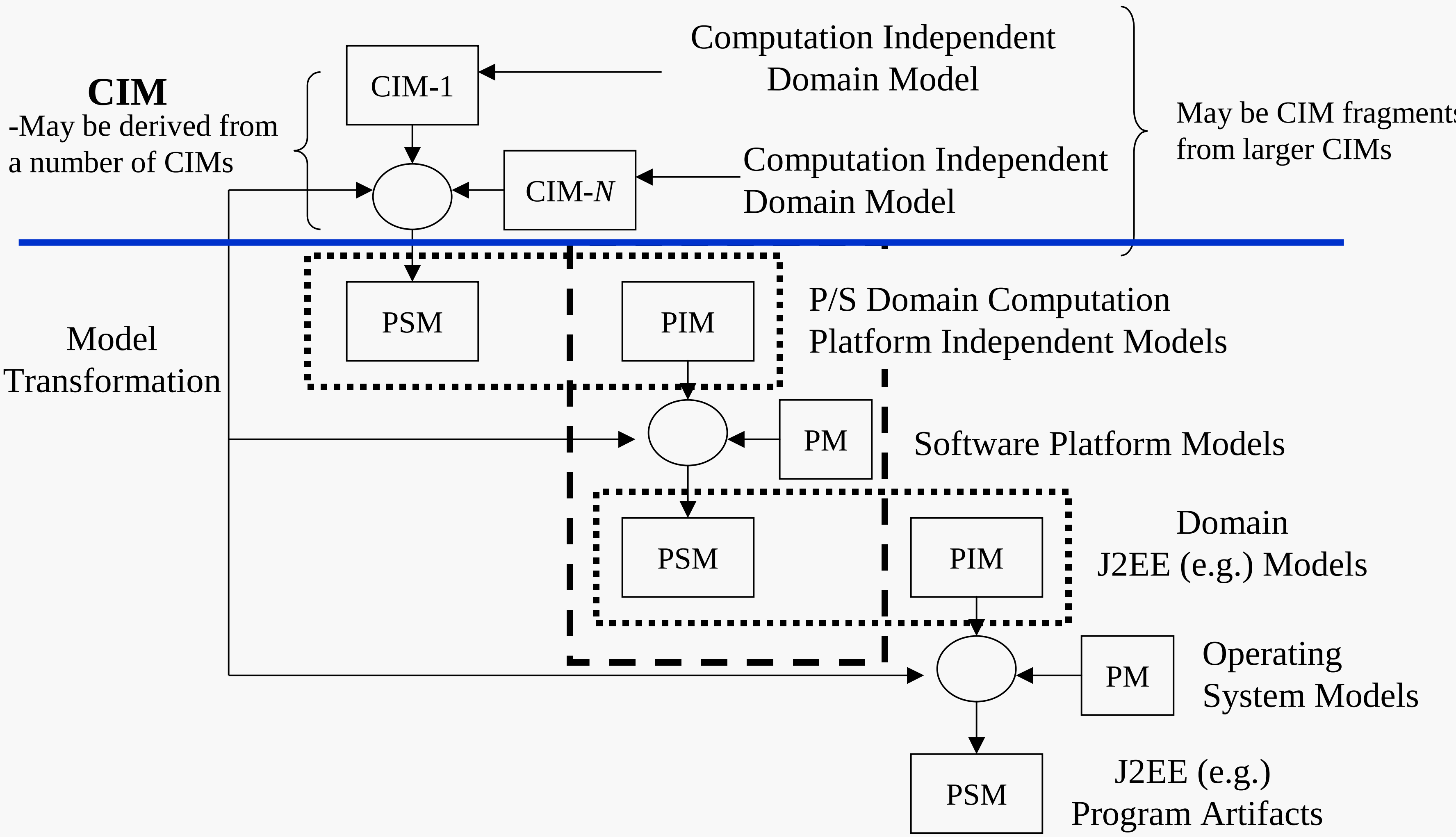
“Model
Transformation”

Approaches:
Marking
Metamodel Transformation
Model Transformation
Pattern Application
Model Merging

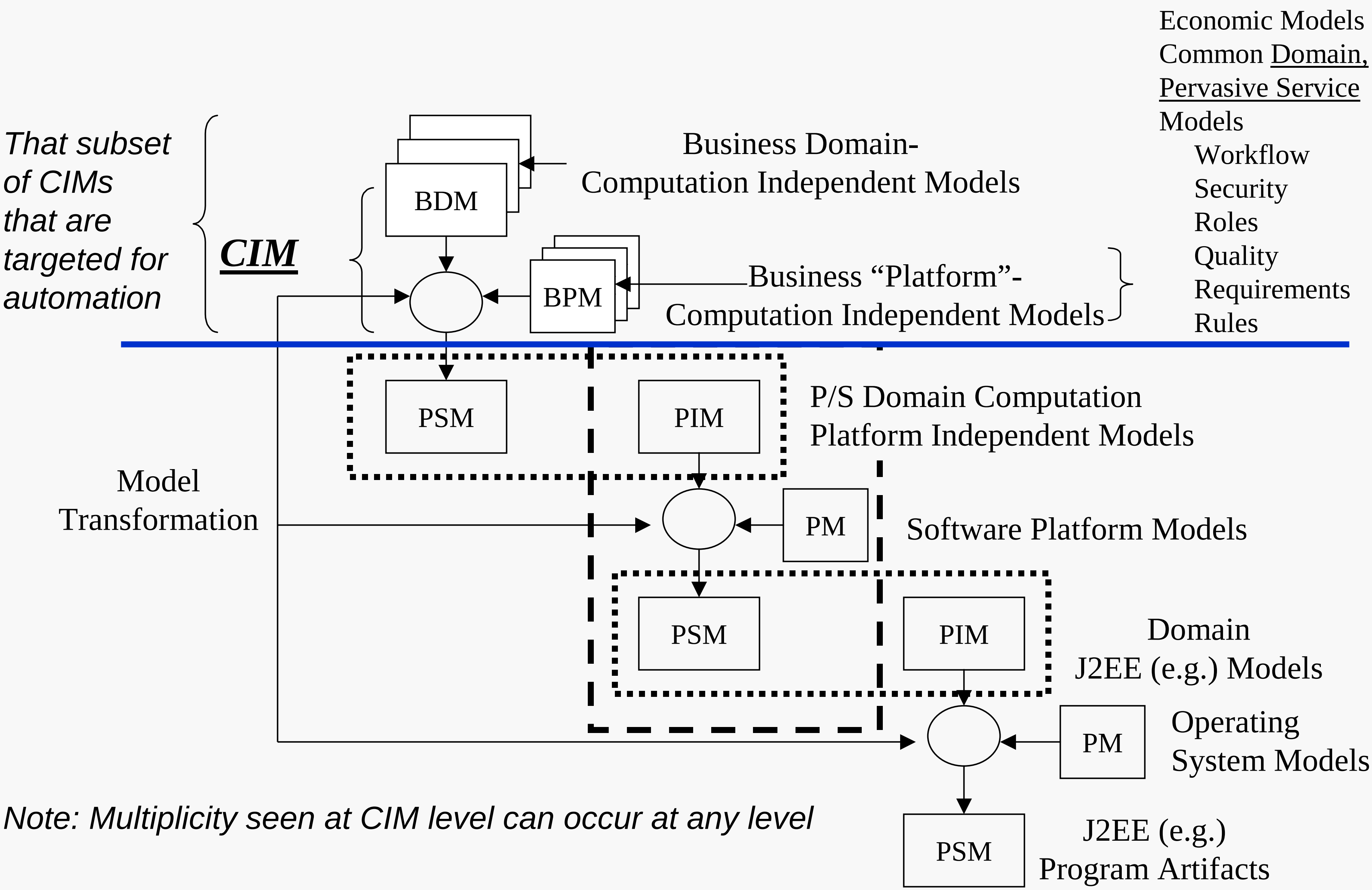
MDA Pattern is Relative



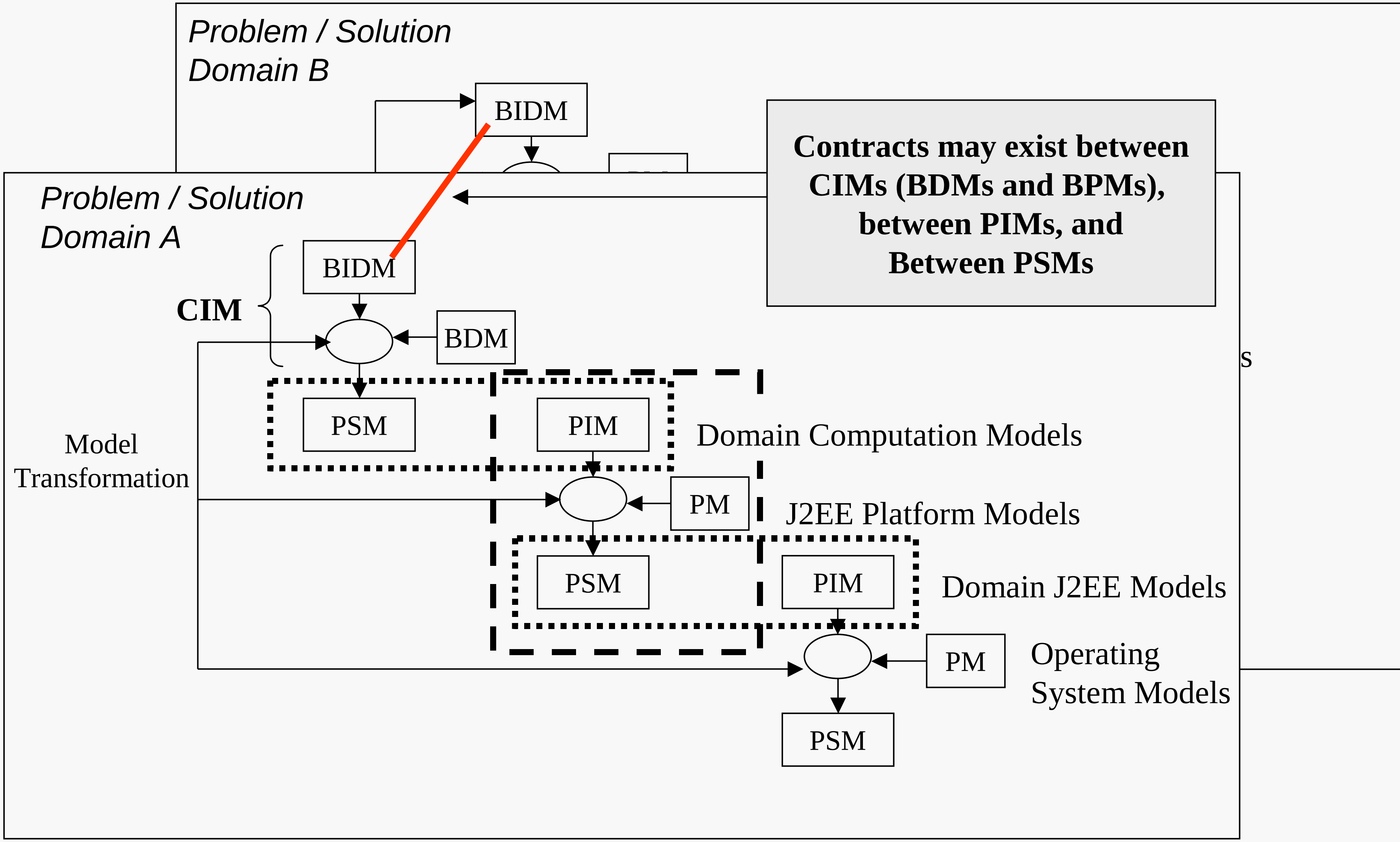
A PIM is Derived From a CIM



CIM May Consist of BDMs and BPMs



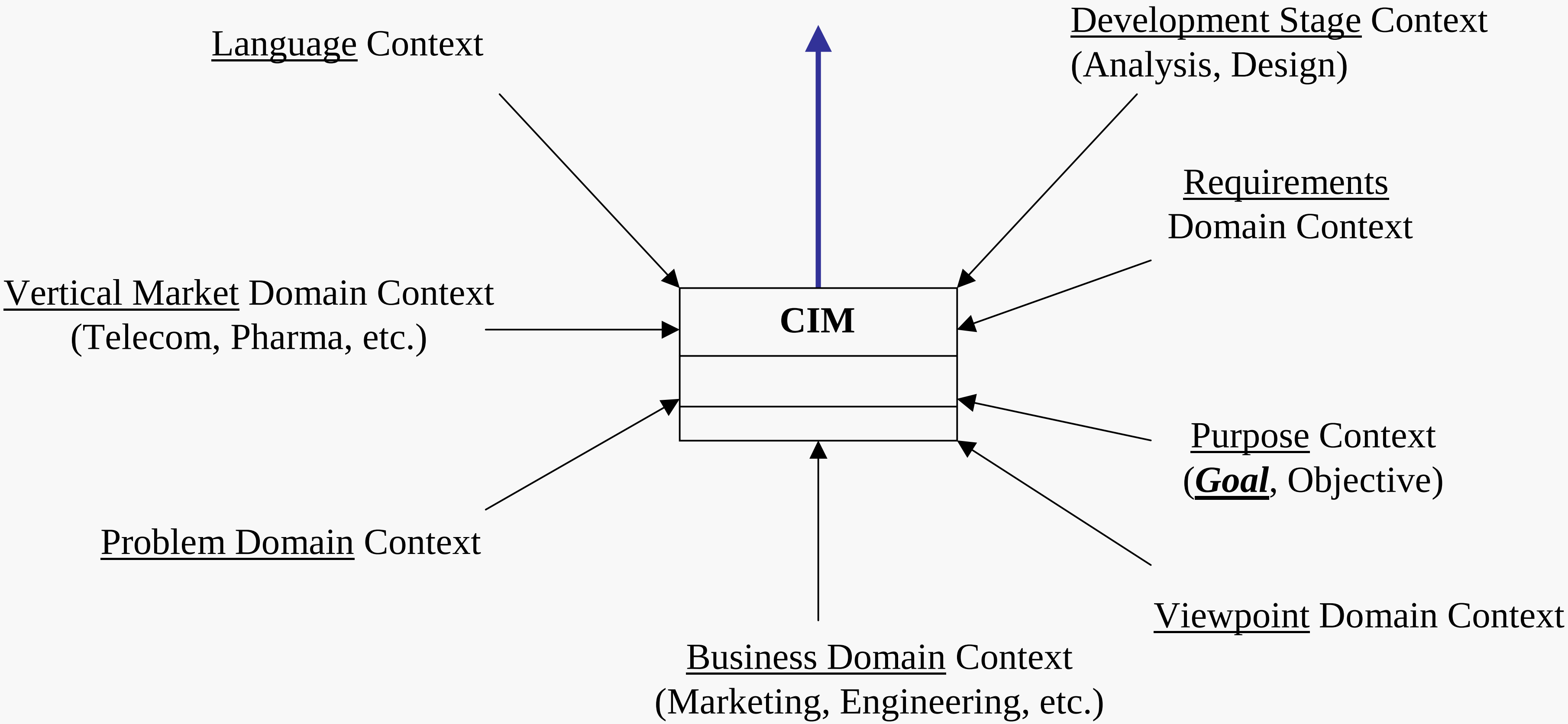
CIMs, PIMs, PSMs May Integrate



Model and Model Fragments are Created in an Instance of a *Context*

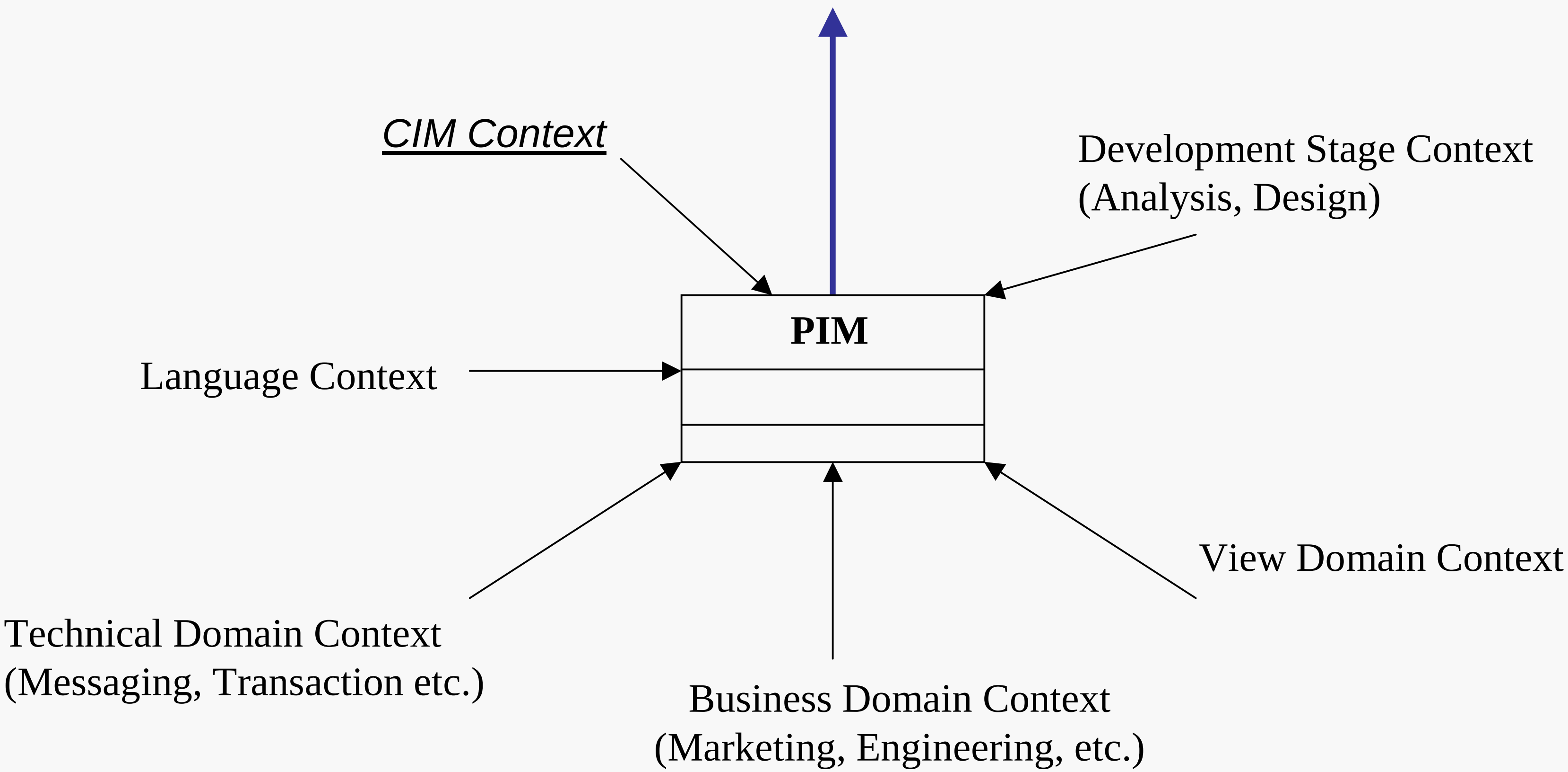
A representation / specification of some part of:

- The current world (of interest)
- A future world (of interest)

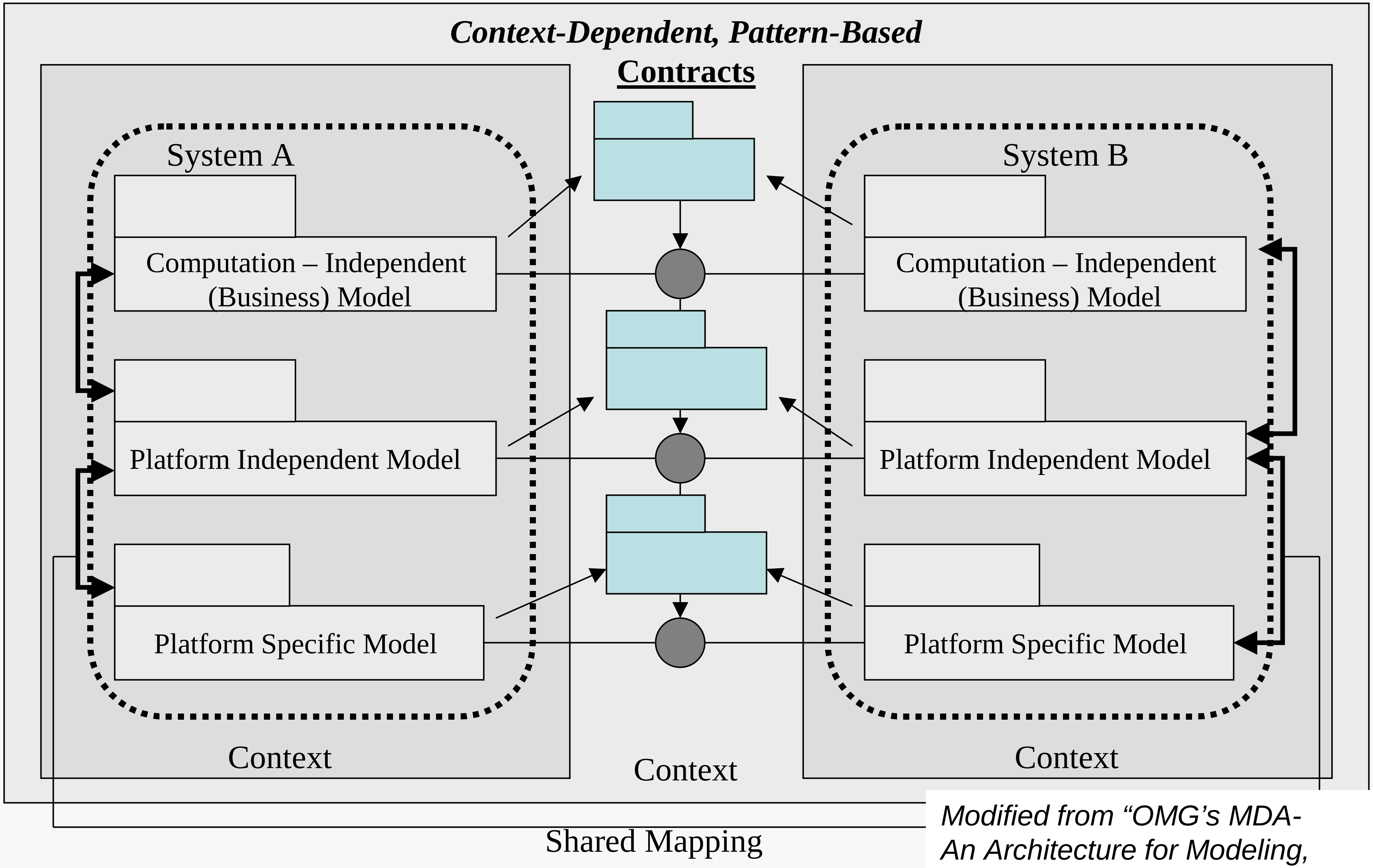


Model and Model Fragments are Created in an Instance of a *Context*

A representation / specification of some part of:
- The current world (of interest)
- A future world (of interest)



Fragment Composition Via Contracts May be Along Context Dimensions



Modified from "OMG's MDA- An Architecture for Modeling, Enabling Model-Driven Integration" Desmond DSousa, Kinetium

CIM/PIM/PSM for Contract May Capture Information Along Context Dimensions (Communication Boundaries)

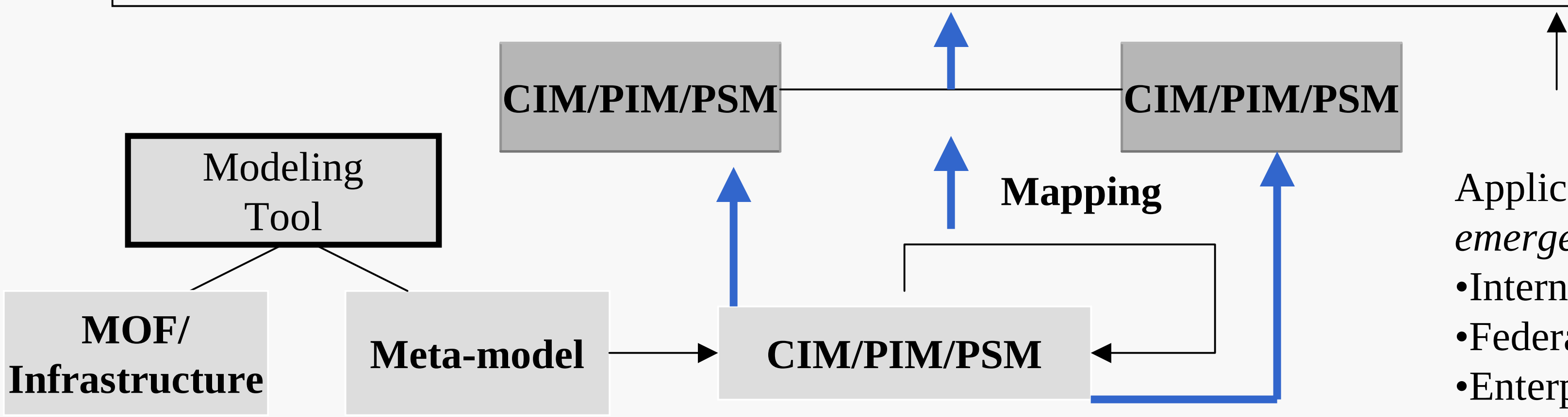
Contract

Context-Dependent, Pattern-Based Contract Contents for Integration, Interoperability and Collaboration

Examples

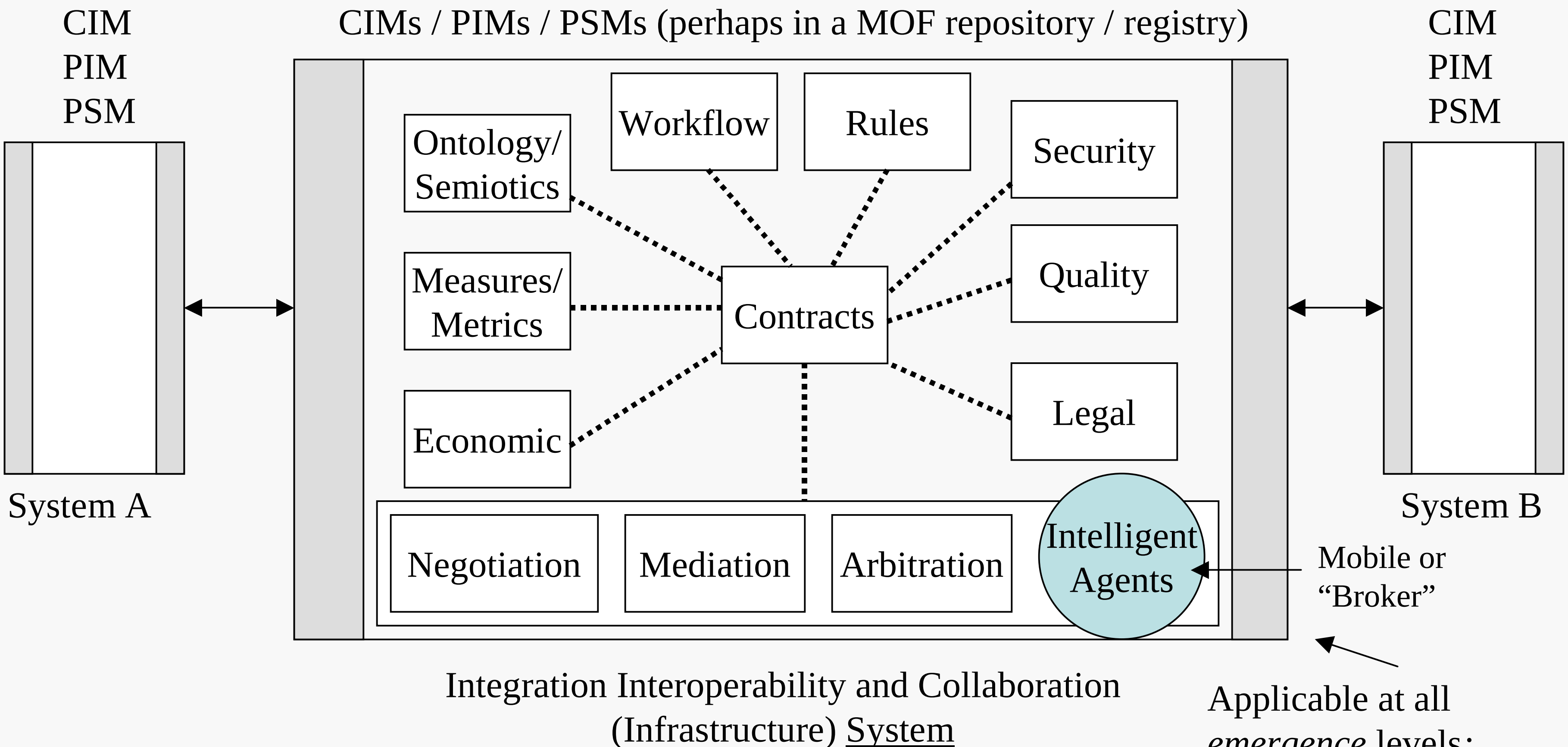
•Ontology (e.g. domain and task)
•/ Semiotics (e.g. semantics and pragmatics)
•Computation (
/ Measures and Metrics (e.g. coupling, cohesion)
•Standards and Best Practices
/ Policies and Procedures

Process, Workflow, Rules
Intelligence (Reasoning, Learning, Adaptability)
Knowledge, Information, Data
Security
Legal
Communications (e.g. Language)
Economic (Competition)



Applicable at all *emergence* levels:
•Internet
•Federation
•Enterprise

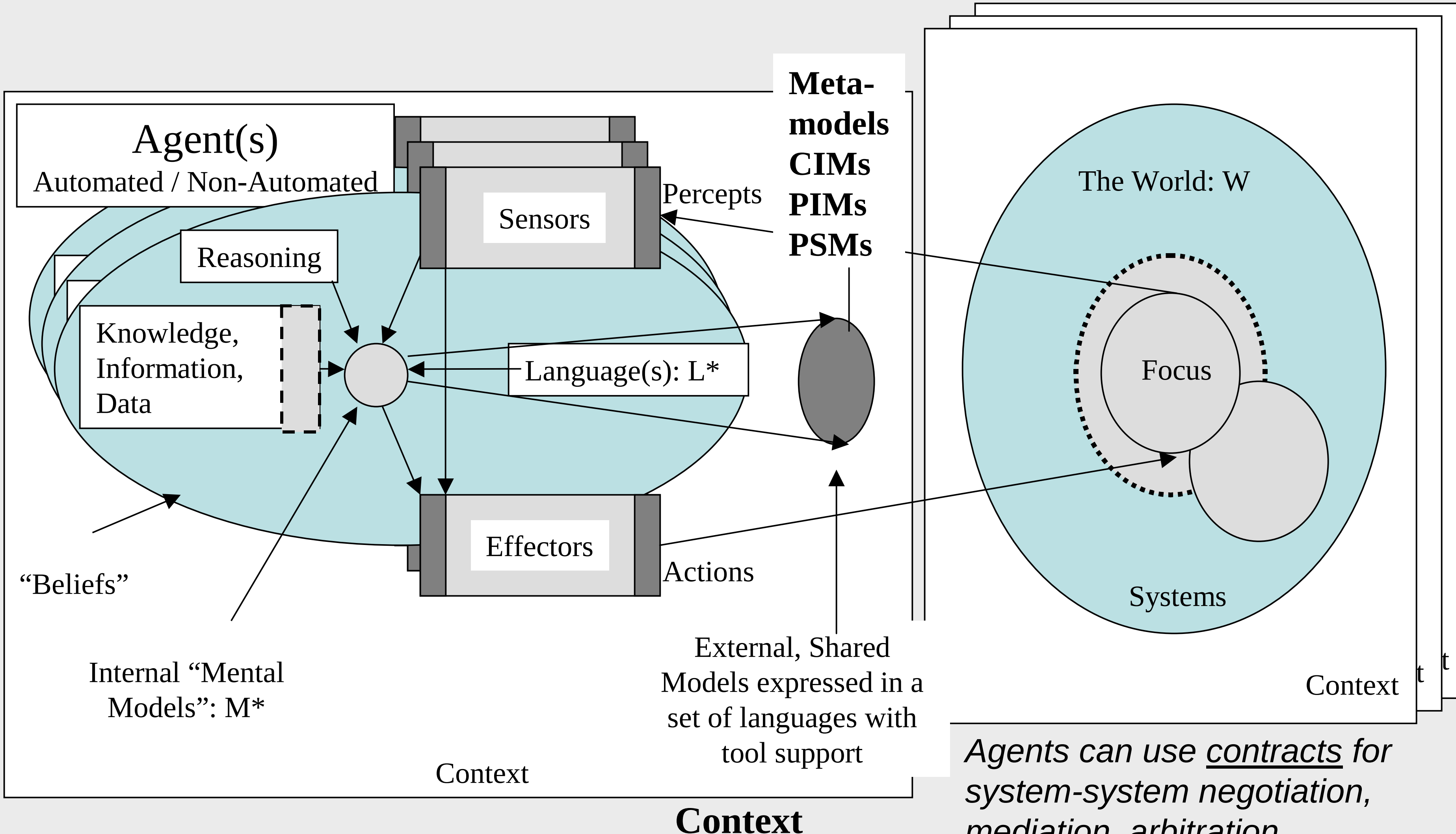
Context Dimensions (Communication Boundaries) May be Subsystems of an Integration, Interoperability and Collaboration Infrastructure System



Applicable at all *emergence* levels*:*

- Internet
- Federation
- Enterprise

Intelligent Agents: Mobile or "Broker"



Agents can use contracts for system-system negotiation, mediation, arbitration

Holistic Conceptualization of MDA Implementation in Enterprise, Federated and Internet Systems

MDA in Context with the "Modeling Discipline" and Enterprise, Federated and Internet Systems

Domain and Systems of Modeling

The “Modeling Discipline” provides a framework – a unified set of primitives, principles, axioms, etc. that provides a cohesive integrated modeling foundation that is derived from each of the science and engineering disciplines that is concerned with modeling.

Scope: Covers modeling and models that occur in all other domains, fields, disciplines, and systems:

Set Theory, Category Theory, Model Theory, Systems Theory, Context Theory, Linguistics, Cognition, Semiotics, Ontology, Logics, etc.

The Modeling Discipline / Domain is a Framework for modeling in all other domains of interest (e.g. Enterprise Domain, Software Domain, Business Domain, Rules Domain, Economics Domain, Security Domain, Quality Domain, Problem/Solution, etc. Each of these domains of interest may have its own framework.

The “Modeling Discipline” is concerned with the ***meaning/interpretation of models*** – to the modelers that created the models, and the users of the models

High-Level Examples of Modeling Principles

- The modeling system should not be biased. The tool should not bias the modeler
- The modeling language should have sufficient syntax and semantics to model problem and solution domains
- Tests for model validity, sufficiency, necessity, completeness, correctness, fitness, etc. are against the requirements and / or the world (including context).
- Viewpoints are filters on the world
- Language is a filter on the world
- The meaning of a model is a function of the interpreter and the context the model and interpreter are embedded in

“System” Sense

- “System”
  - Refers to a group/collection of units/elements combined in such a way as to form a whole and to operate in unison
- “Natural System”
  - Refers to some part of a current or future world
  - E.g. a human / ”agent” construct which has an implied or explicit closure / boundary
- “Formal System”
  - Refers to systems of logic/reasoning

“System” is taken in the system-theoretic sense

All natural systems are taken to be “emergent” systems (the whole is greater than the sum of its parts) unless proven/designed to be otherwise.

Systems in the logistic sense are also seen as “emergent” as in the case of emerging knowledge domains

Systems in the logistic sense may “drive” the behaviors/actions of systems in the natural system sense.

“Model” Sense

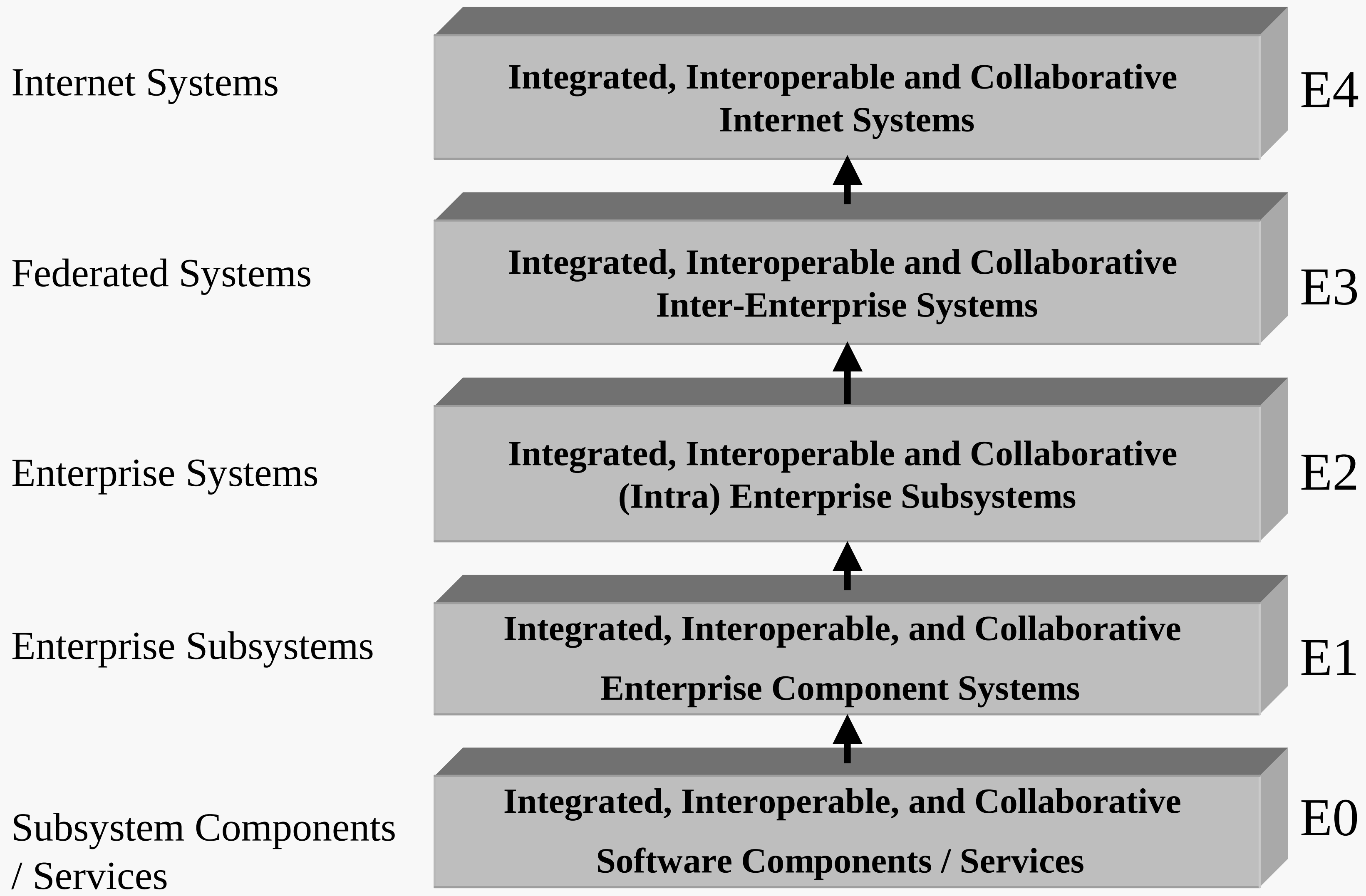
- “Model”: A representation/specification, expressed in one or more formal languages (with defined semantics), of a part of the function, structure, and / or behavior of a <u>system</u>
- “Natural System Model”
  – Refers to a model of a natural (current or future) system
- “Formal System Model”
  – Refers to a model of a formal system
  – Is a basis for theory construction

Models may model other models, formal or natural - metamodeling

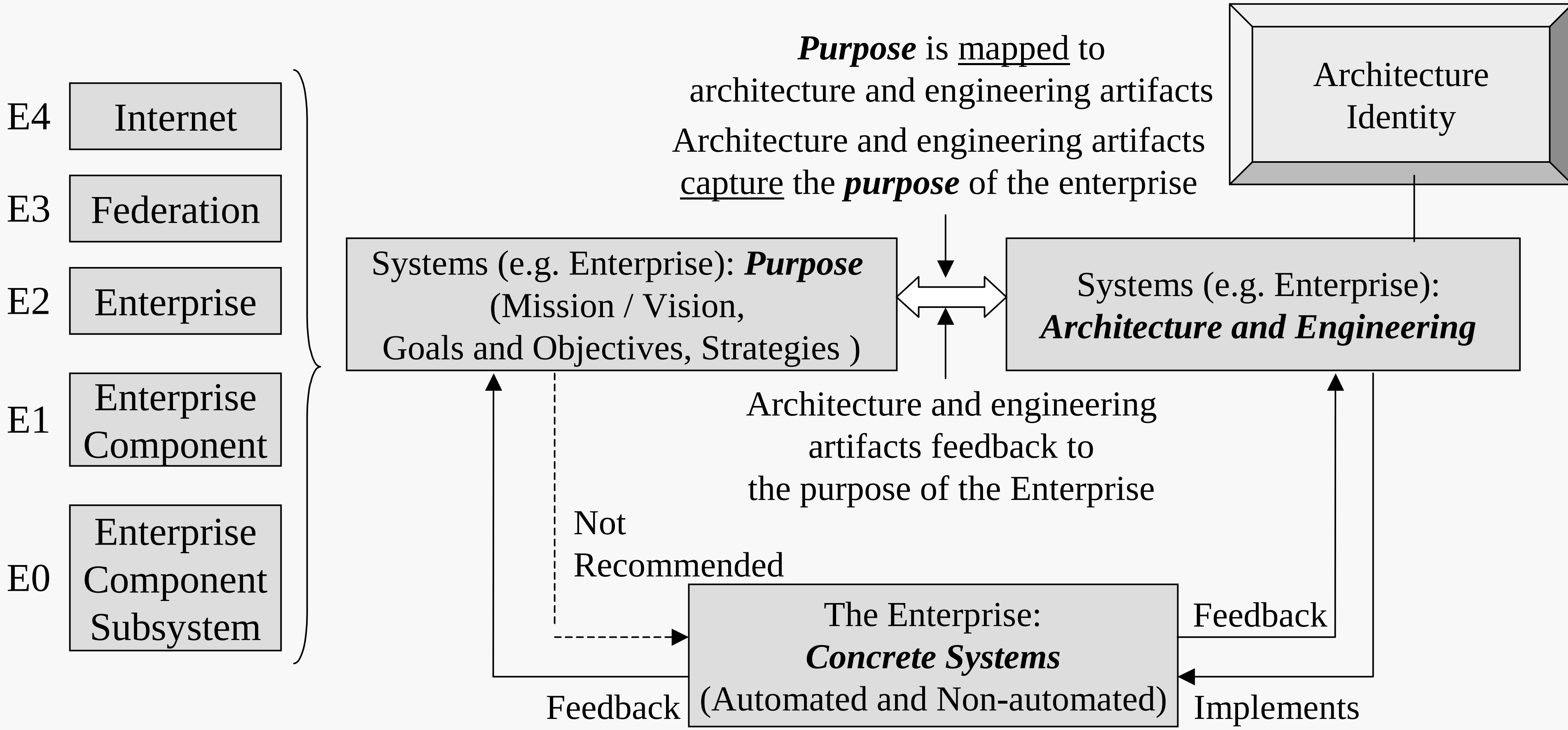
Integrated, Interoperable and Collaborative Systems: Architecture and Engineering (IICSAE™)

- A system (framework and methodology) based in, and ‘implements” the “Modeling Discipline”
- Is used as a basis for software, enterprise, federation, and Internet system modeling
- MDA (approach and tools) is embedded in it
- Focus is on the domain and systems of modeling
- Each “component”/subsystem is based on some parts of different disciplines. For example, Measurement Theory and Category Theory make up parts of the Measures and Metrics system, Ontology Theory makes up part of the Ontology System, Shannon’s Information Theory makes up part of the Communications System, etc.
- All systems, current and future, modeling and modeled, are to some degree considered to be emergent, complex adaptive systems

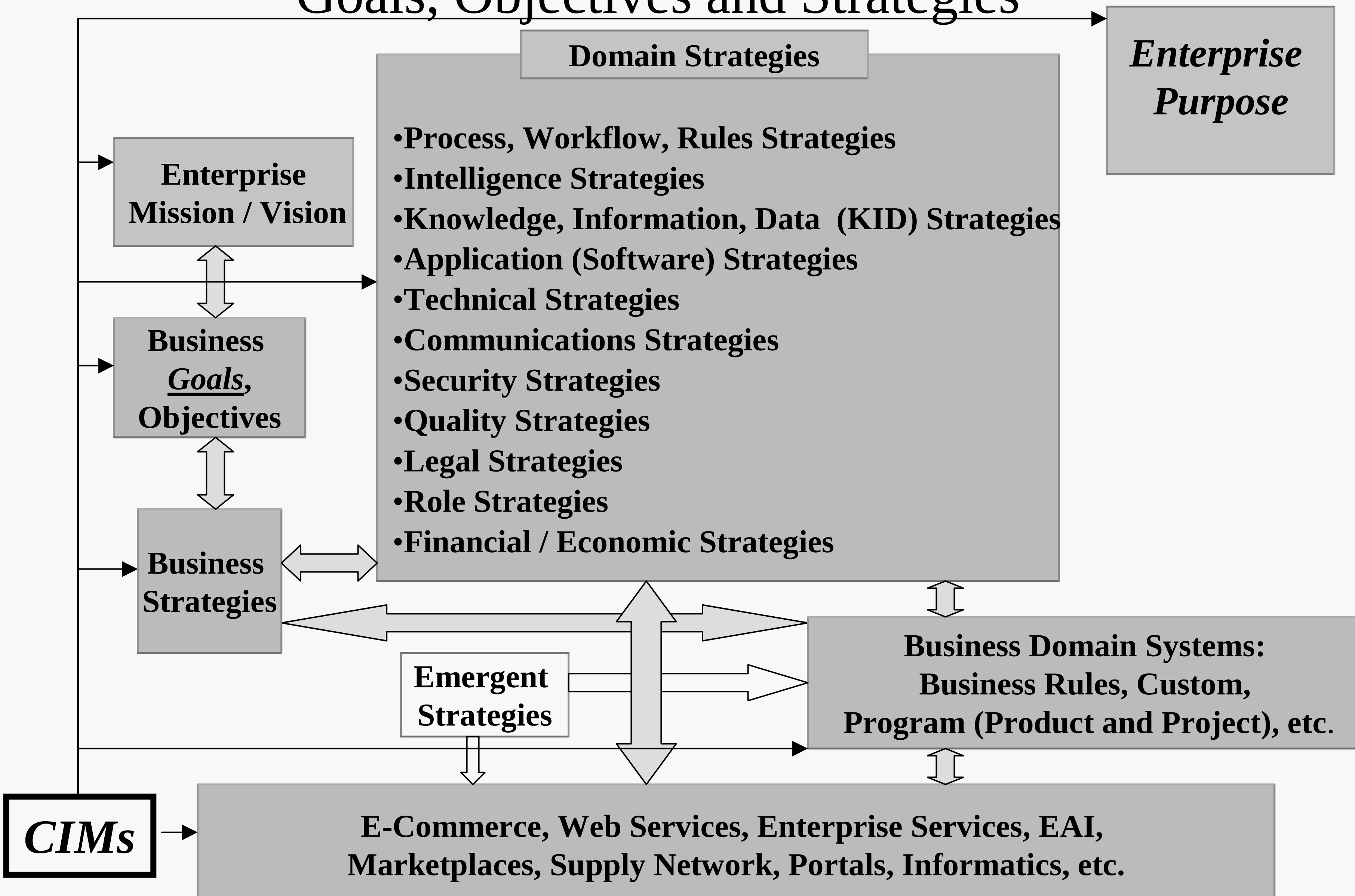
Emergence Layers for IICSAE



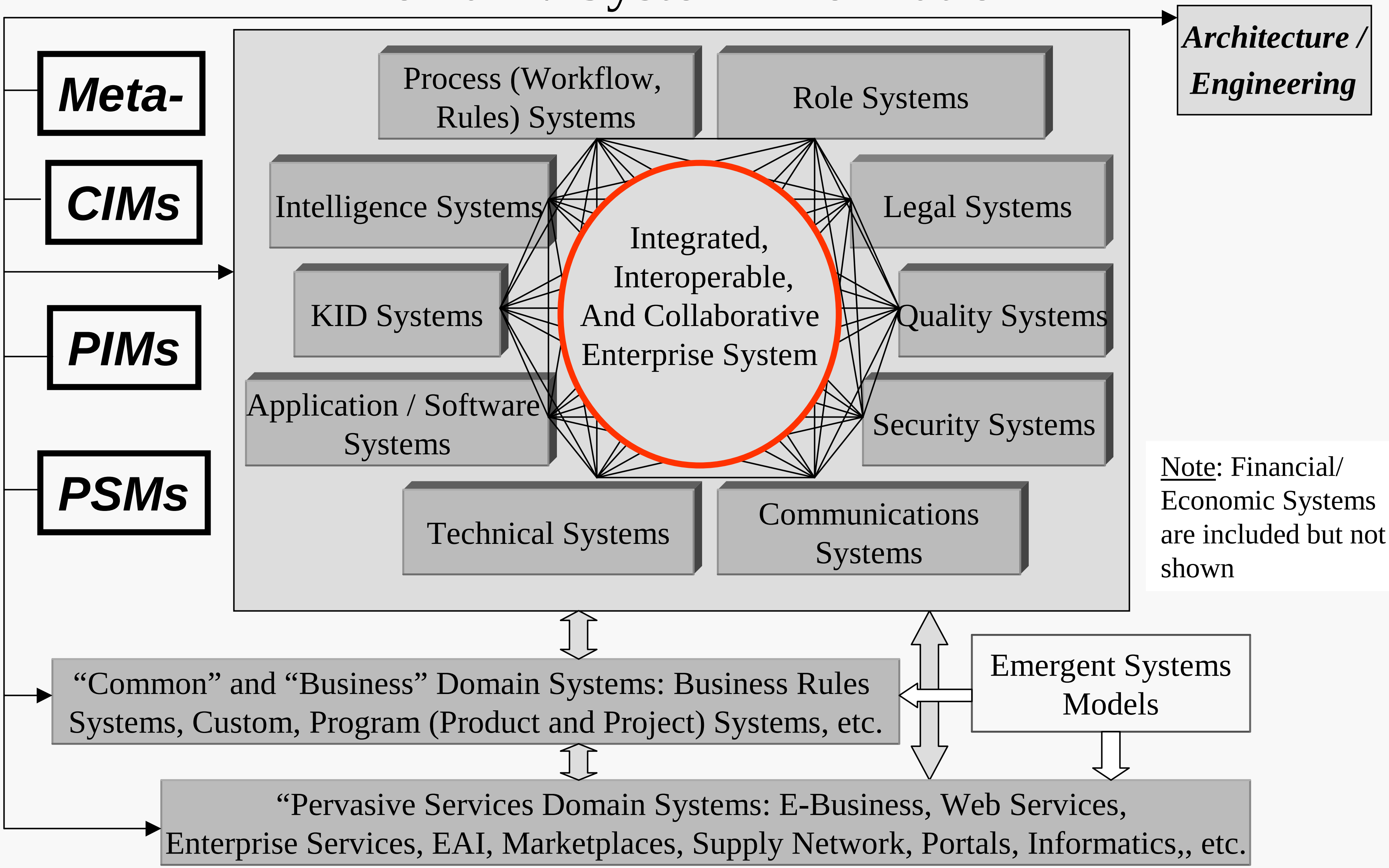
Architecture and Engineering Artifacts (e.g. Models) are Positioned between System Mission, Goals and Objectives and Concrete Systems



CIMs May Reflect System Mission, Vision, Goals, Objectives and Strategies

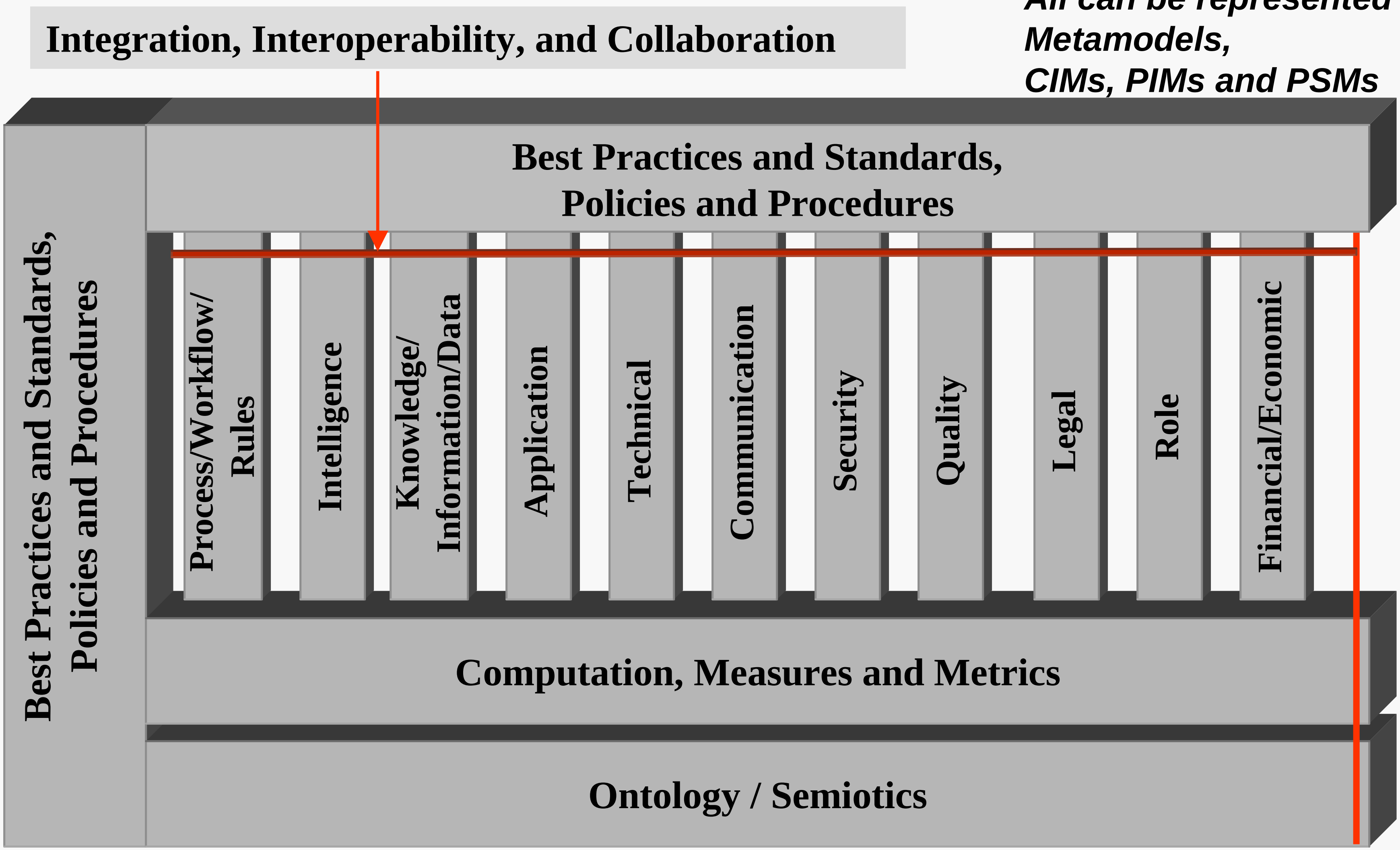


CIMs, PIMs and PSMs May Capture Specific Domain / System Information



CIMs, PIMs and PSMs May Capture Specific Domain / System Information

All can be represented in Metamodels, CIMs, PIMs and PSMs

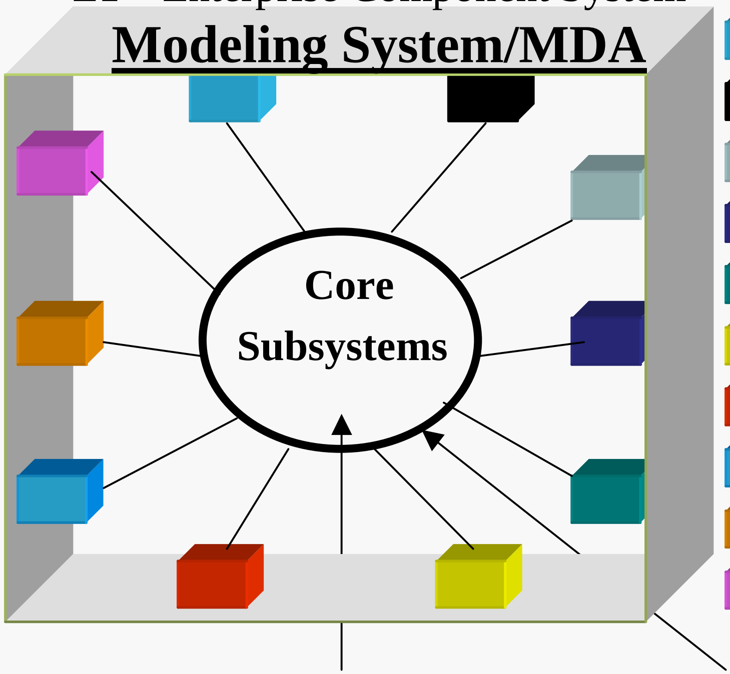


Conceptual Illustrations of a Modeling System with MDA Embedded in Enterprise, Federated and Internet Systems

Emergence of Enterprise Component Systems From the "Merging" of CIMs, PIMs and PSMs of Common Domain Systems: A Modeling System

All can be CIMs PIMs PSMs Meta-models

L1 – Enterprise Component System



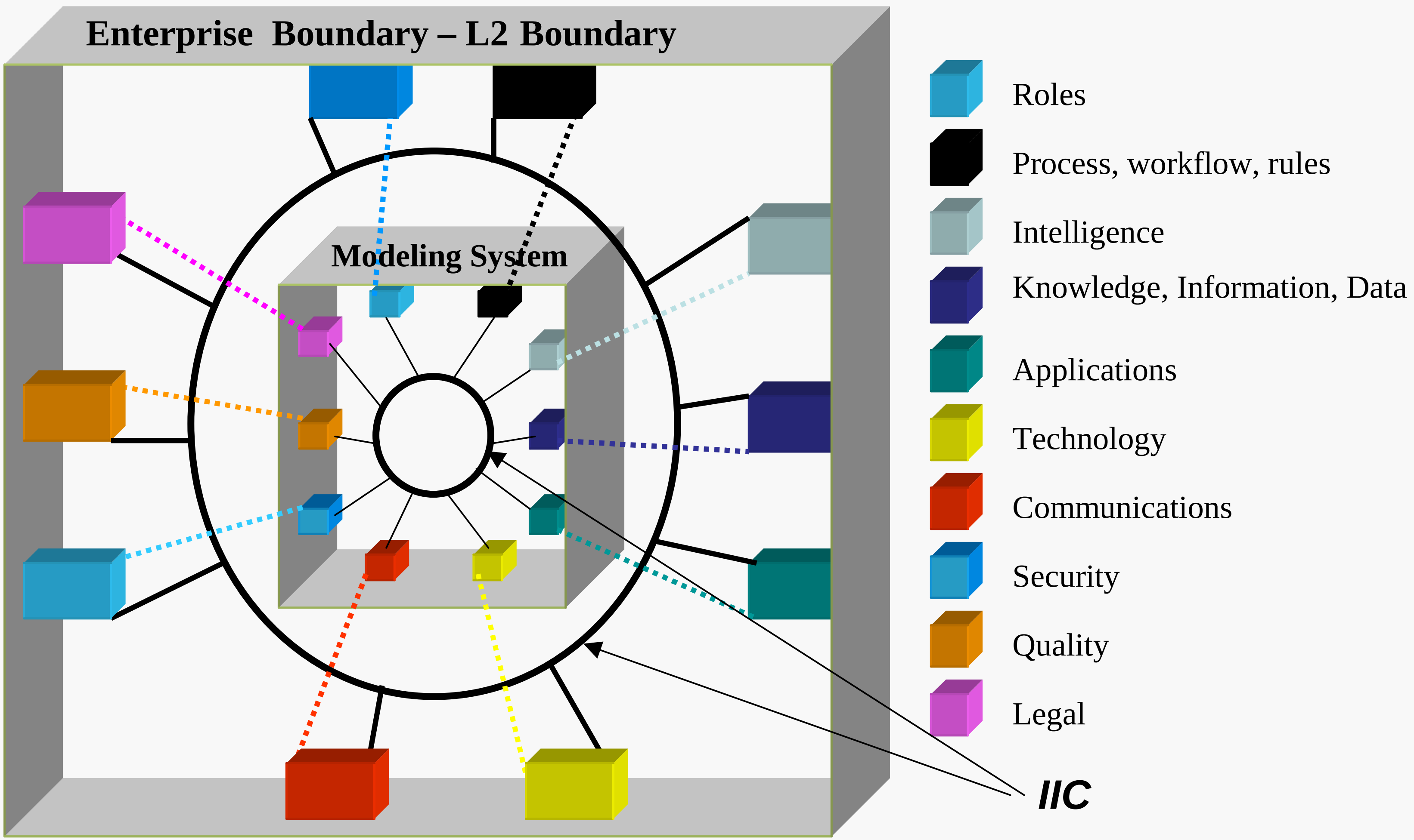
Foundation Subsystems of the Modeling System

- Roles System
- Process, Workflow, Rules System
- Intelligence System
- Knowledge, Information, Data System
- Applications System
- Technology System
- Communications System
- Security System
- Quality System
- Legal System

Integration, Interoperability, Collaboration via
•Ontology and Semiotics System
•Computation, Measures and Metrics System
guided by
•Best Practices and Standards, and Policies and Procedures

•Integration, Interoperability and Collaboration System (IIC):
Interfaces
Contracts,
Bindings,
Mappings,
Context
"Broker"

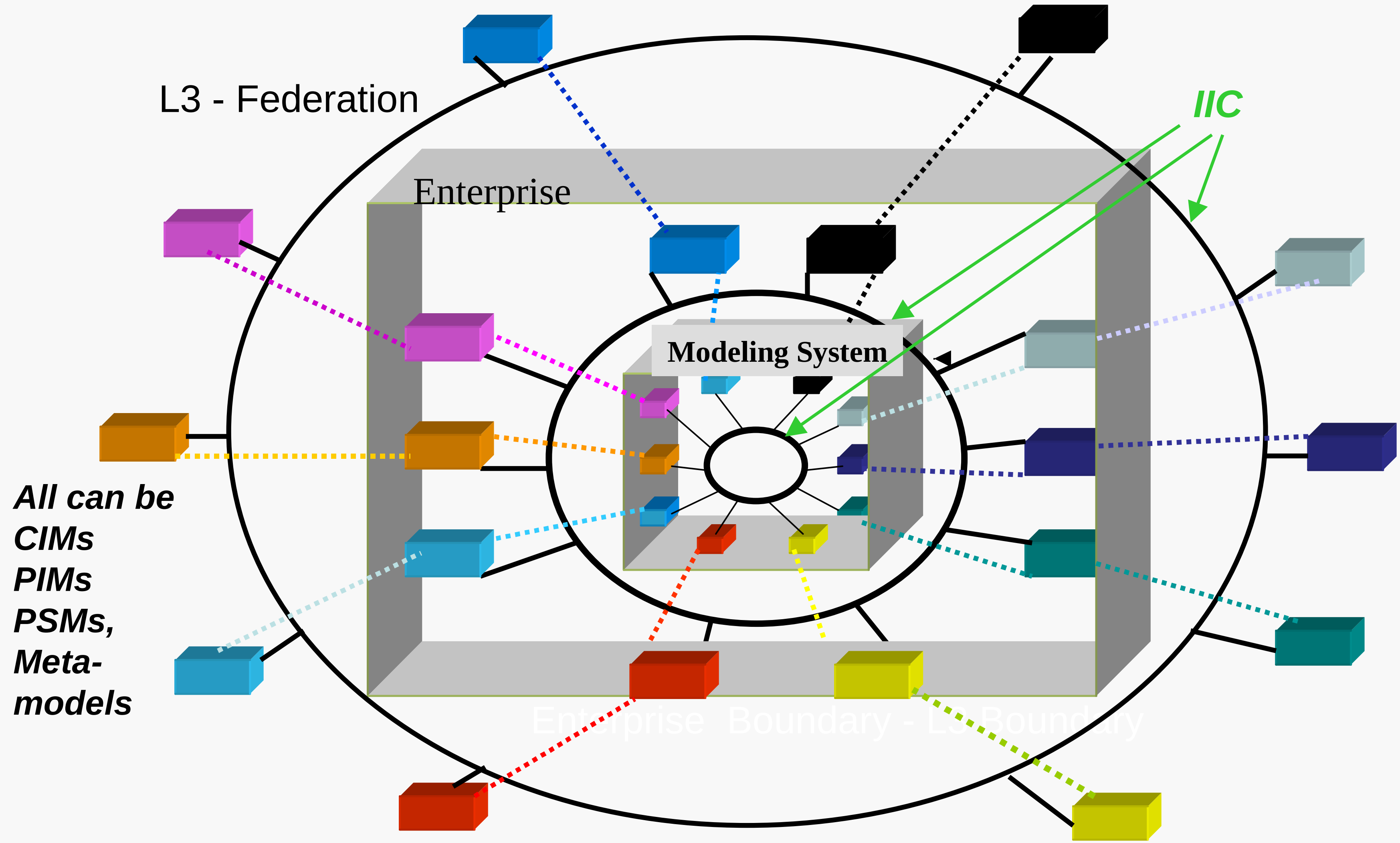
A Modeling System Embedded in an Enterprise Context



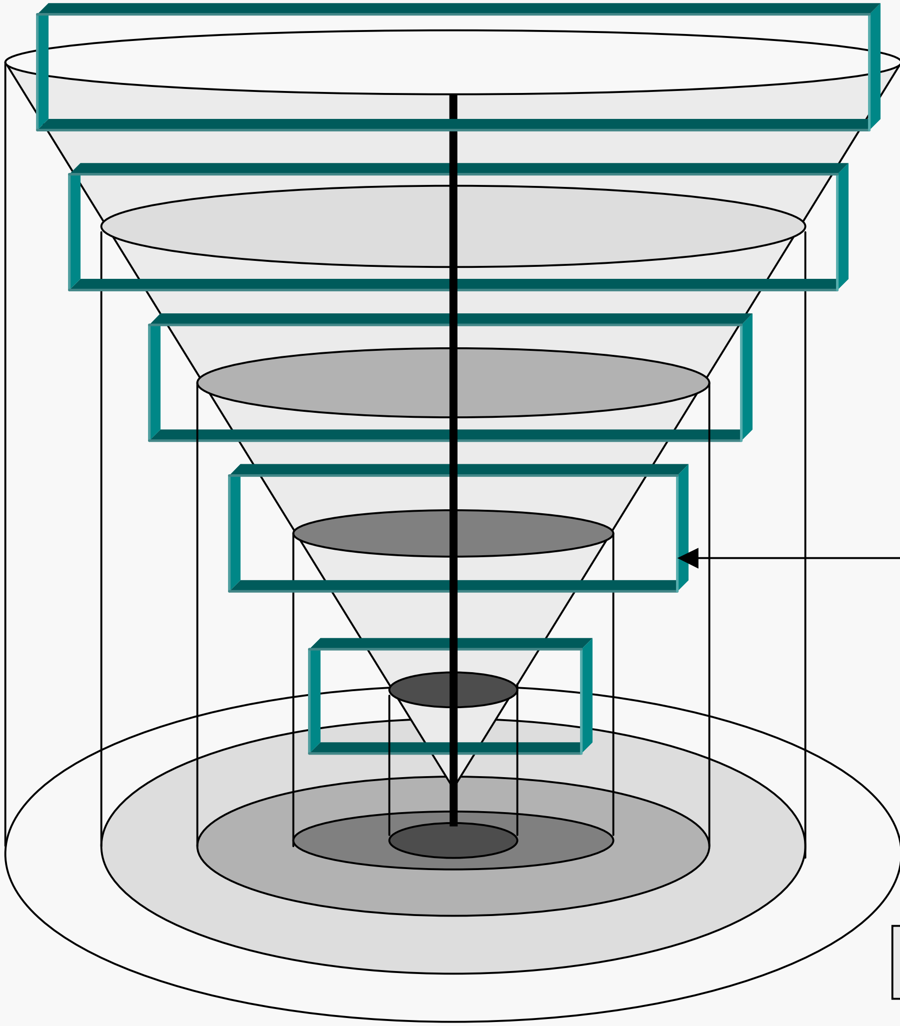
All can be CIMs PIMs PSMs Meta-models

May 12, 2003

A Modeling System Embedded in a Federation



A Modeling System in Context with the Internet



E4 – Internet Systems

E3 – Federated Systems

E2 – Enterprise Systems

E1 – Enterprise Component Systems
(e.g. Modeling System – embedded in Internet Systems)

E0 – Enterprise Component Subsystems

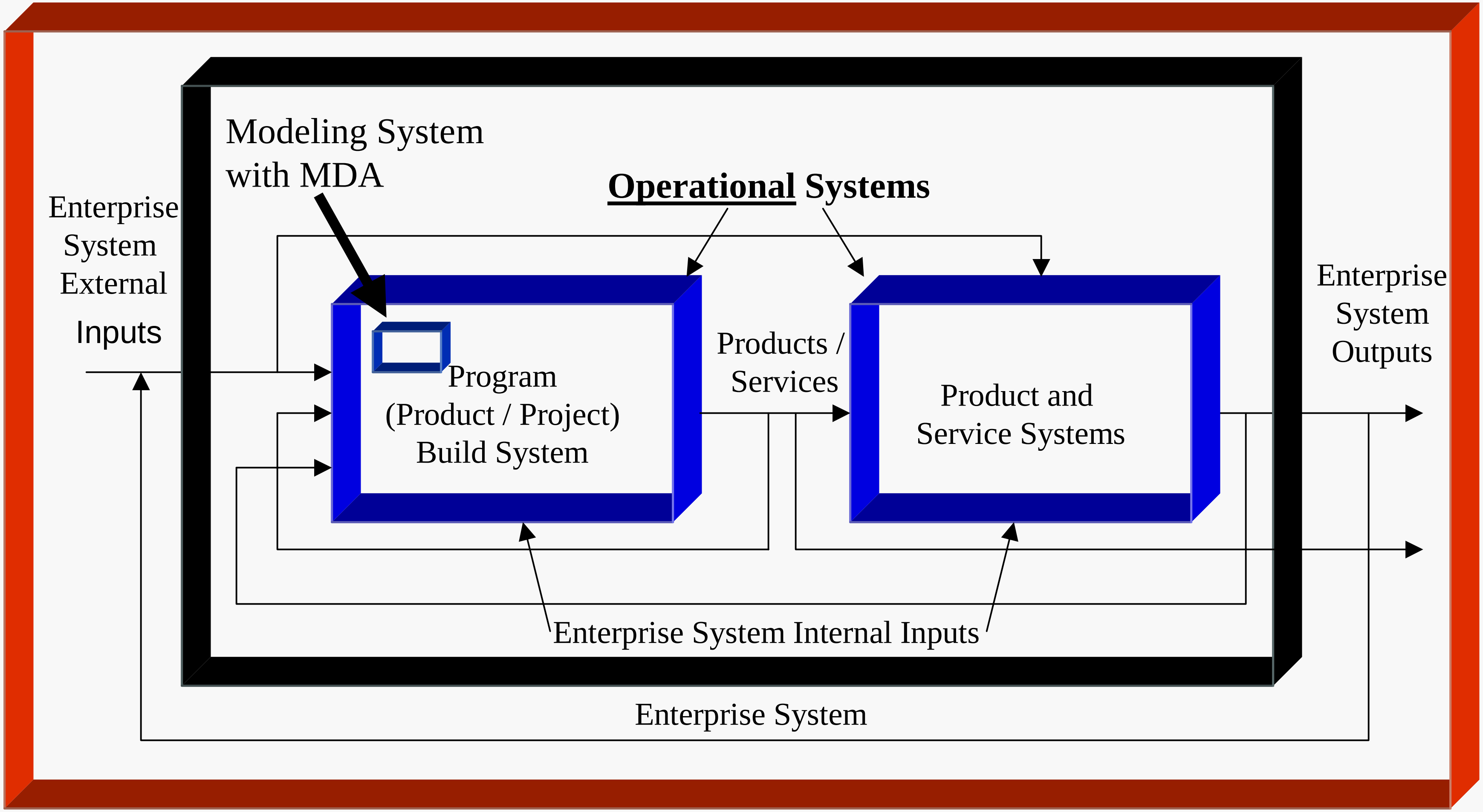
The scope of the core at a higher level contains the system at the lower level

Emergent Systems Cone™ ←→Universal IIC Cone™

System – Context – Domain
Emergent Integration, Interoperability and Collaboration

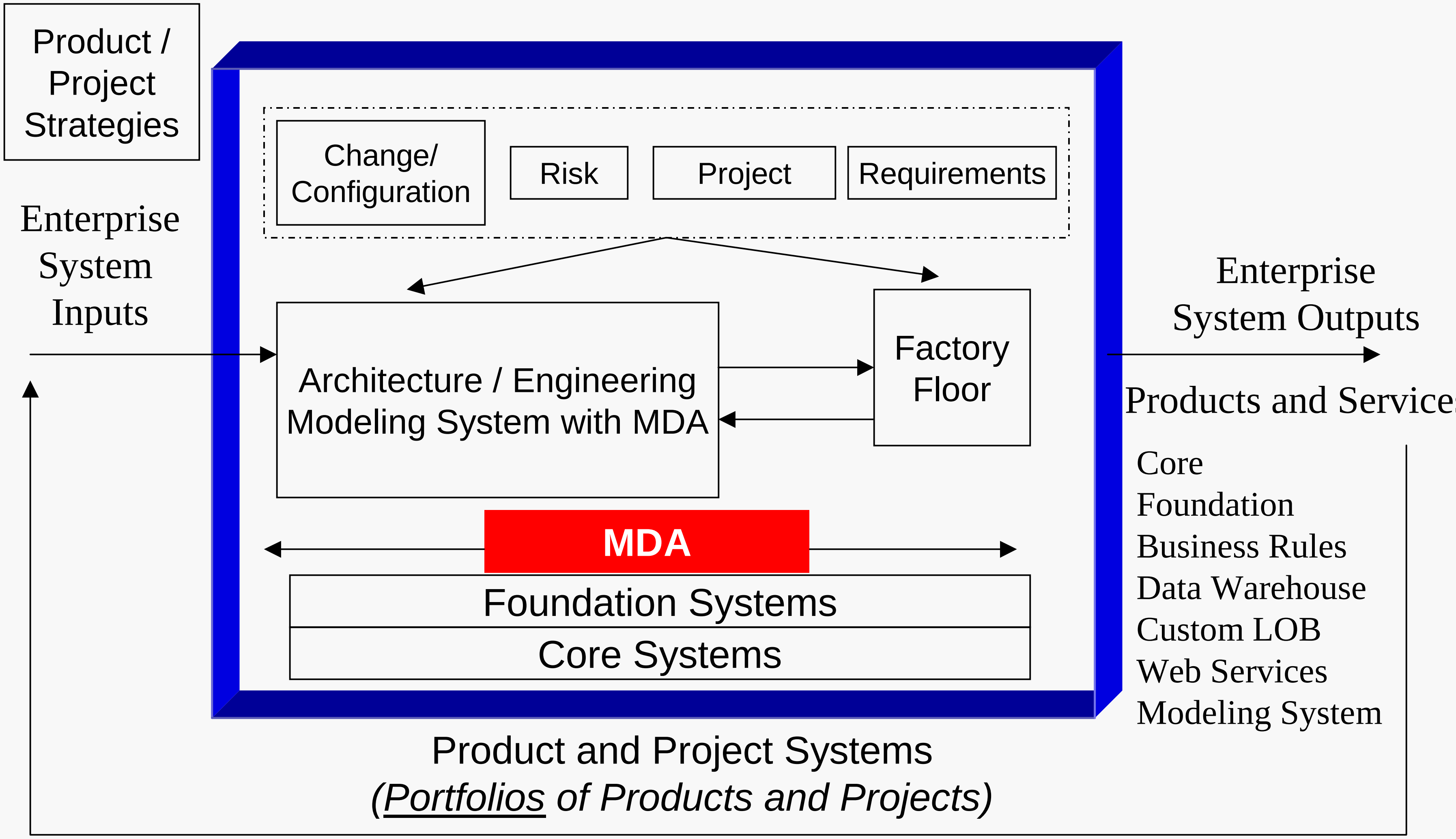
Conceptual Representation of a Modeling System with MDA at the Enterprise Level

The Modeling System is Part of a Architecture and Engineering System Embedded in a Product and Project System – Factory Motif

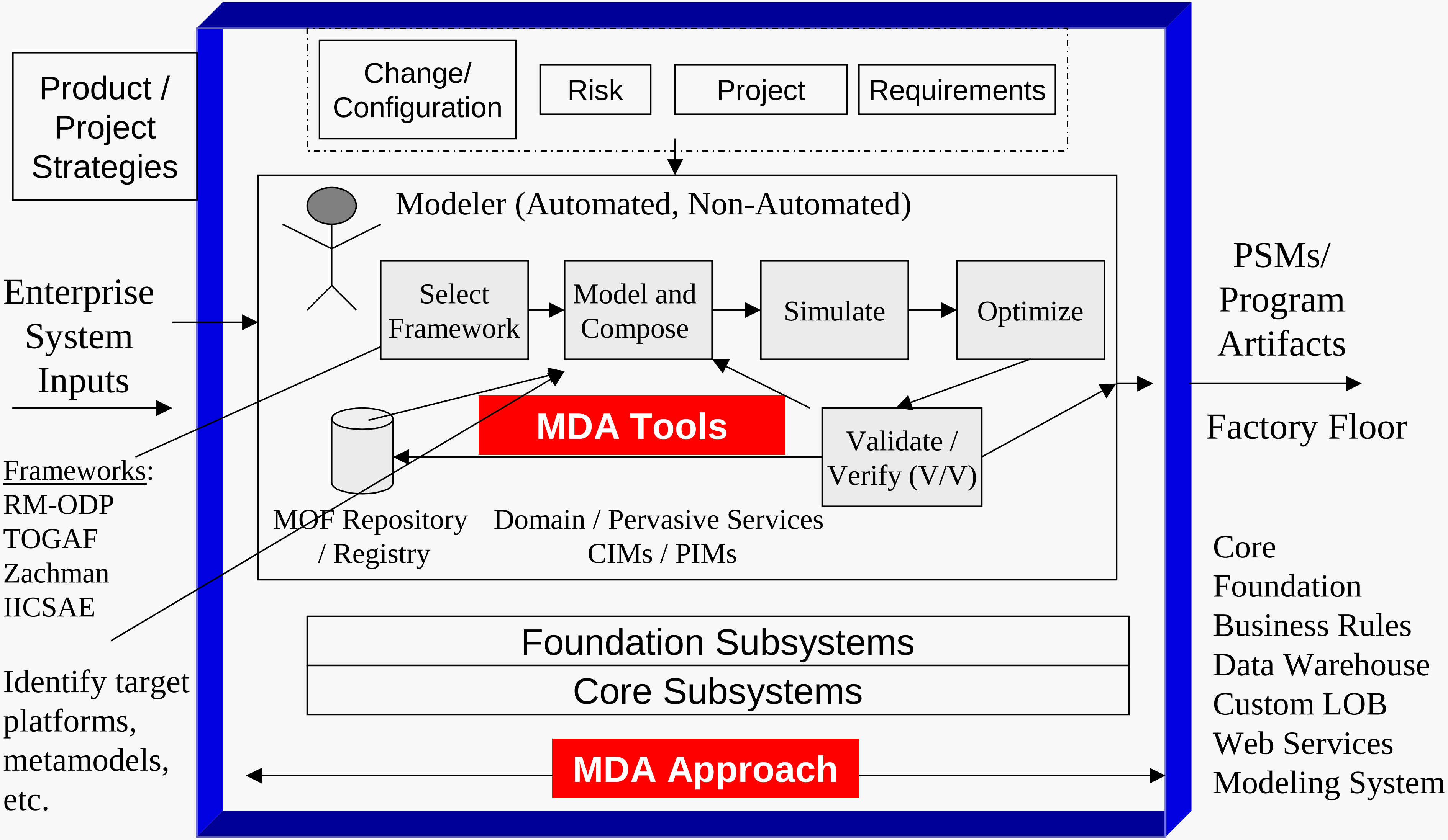


Enterprise Context and Environment

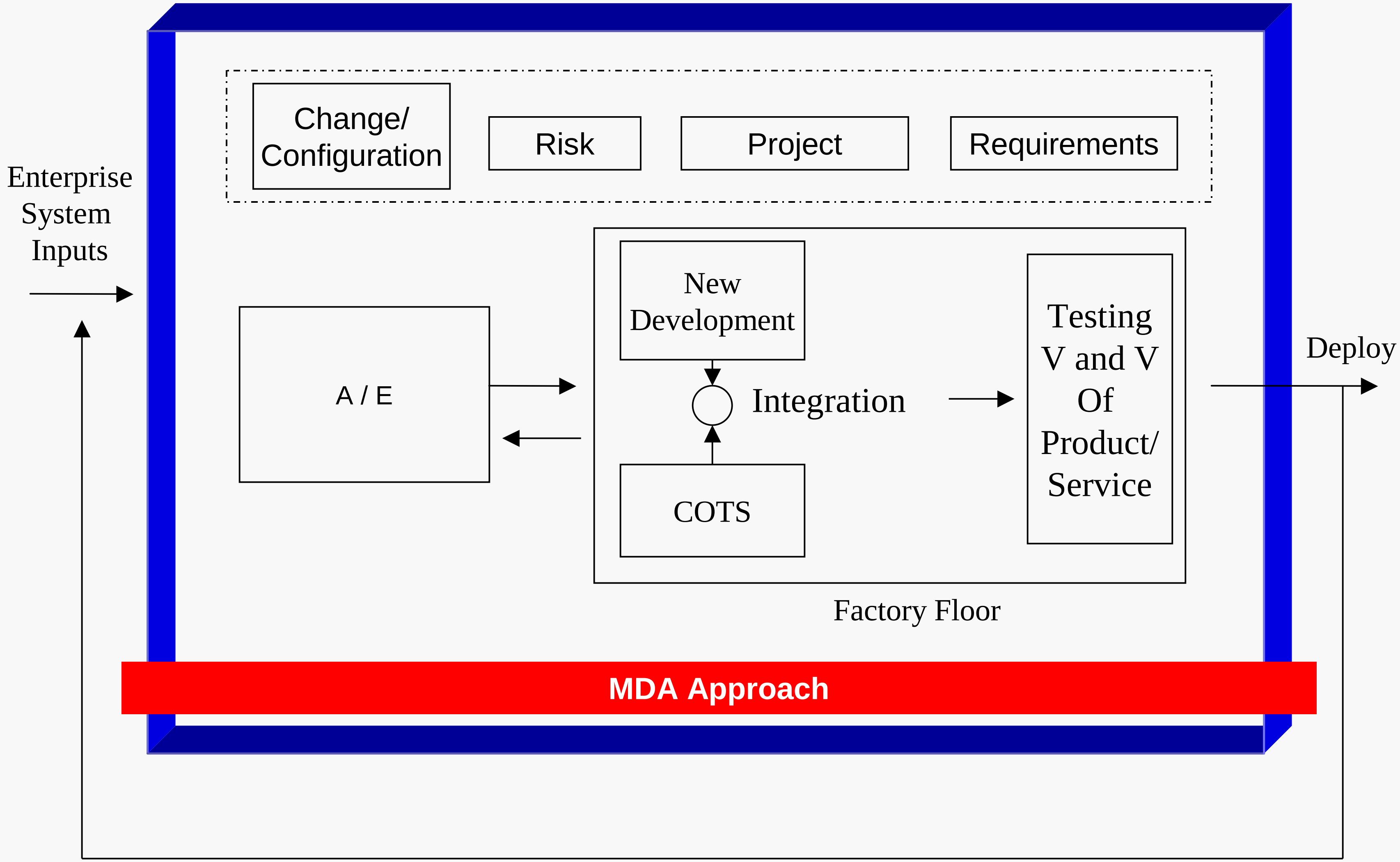
The “Build” Environment – Factory Motif



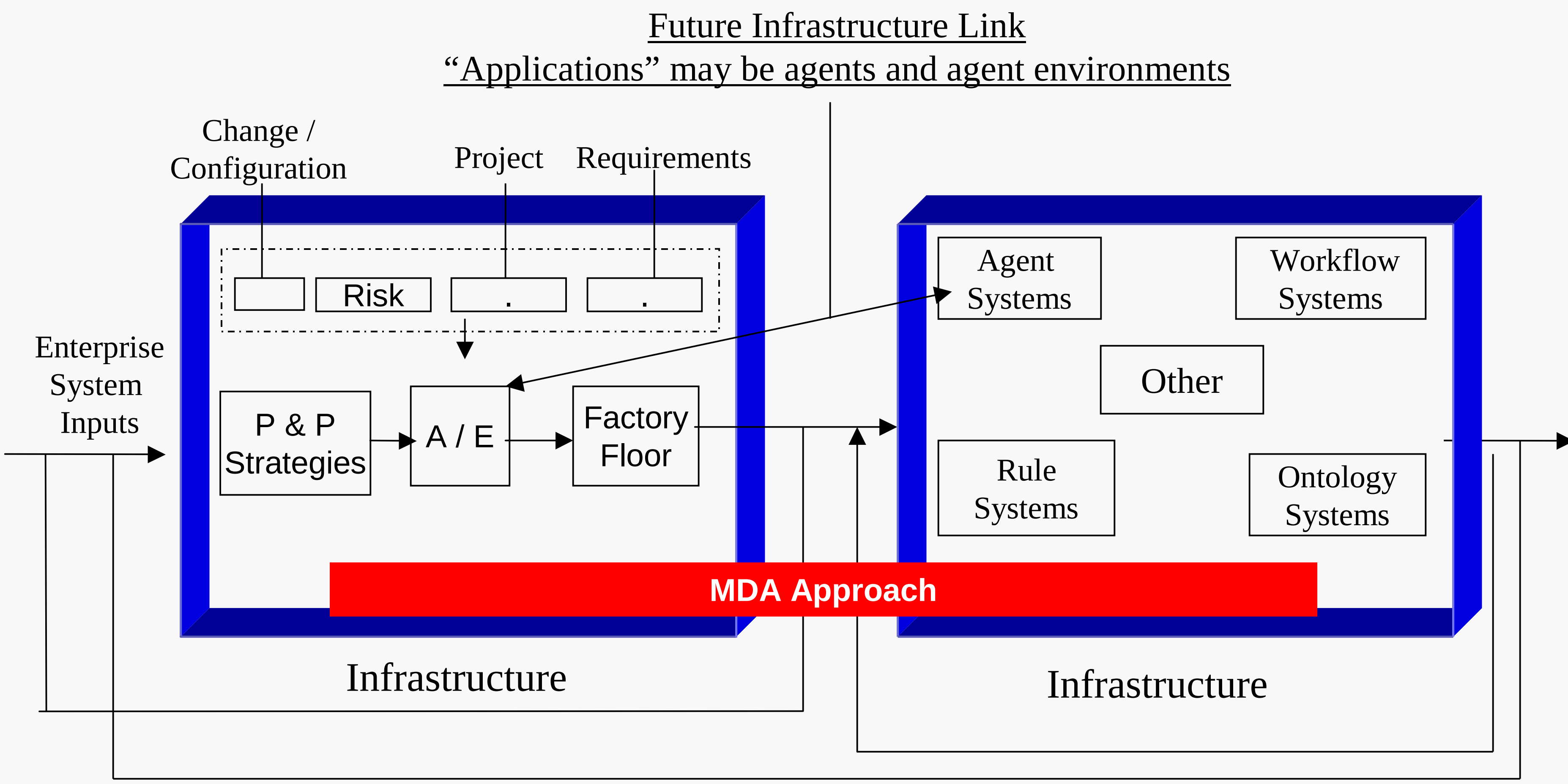
The Architecture and Engineering (A/E) System (Modeling System with MDA)



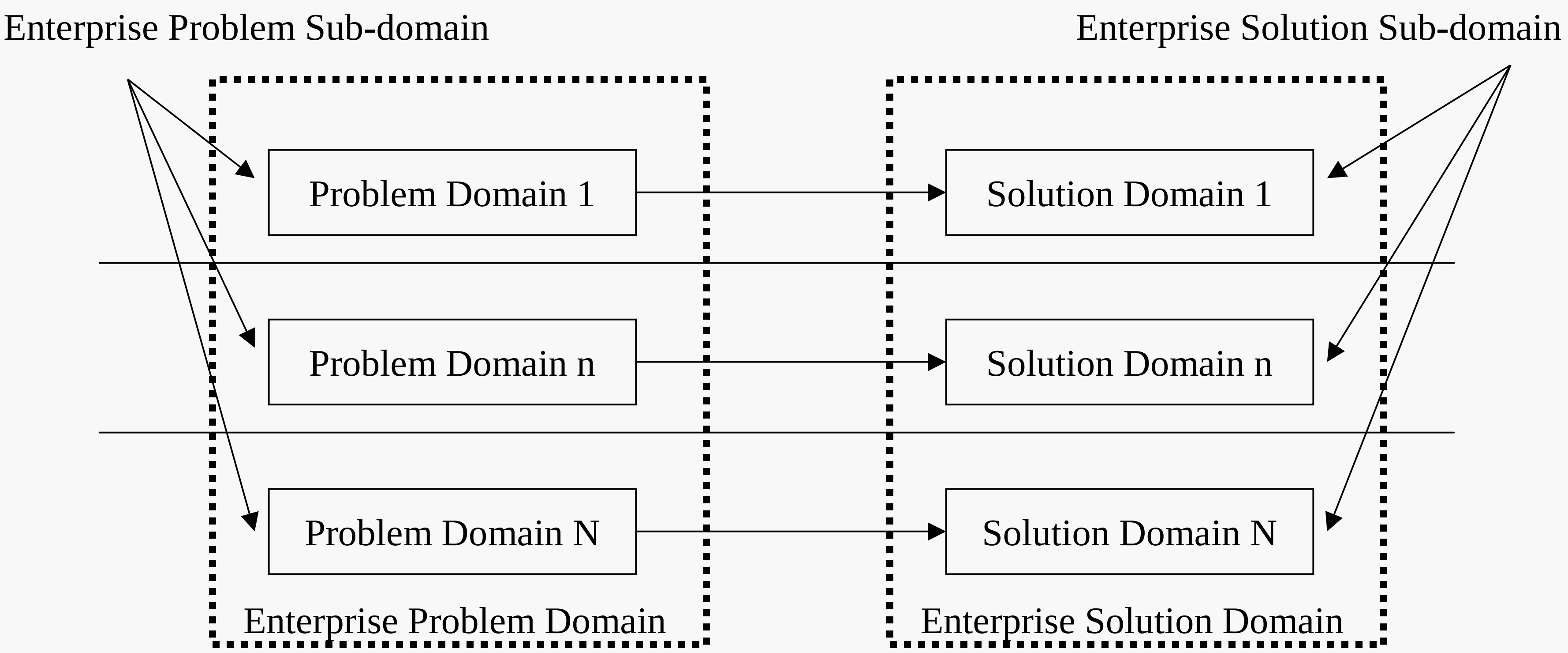
The Modeling System and the Factory Floor



The Modeling System in Operational Systems Feedback and Feed-forward Information and Control

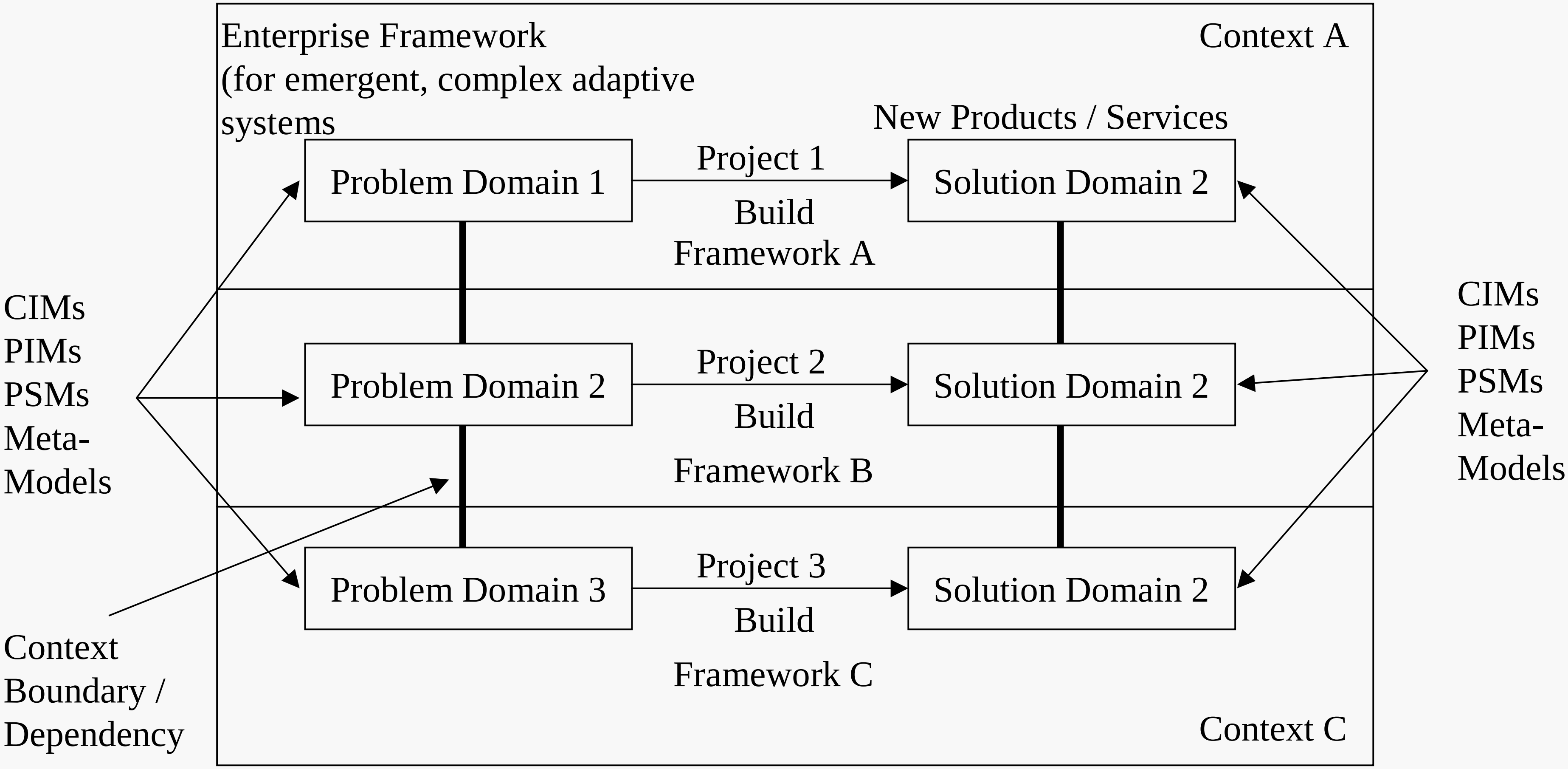


The Modeling System (with MDA) Addresses the Enterprise Problem/Solution Domain Enterprise Product/Project Programs/Portfolios



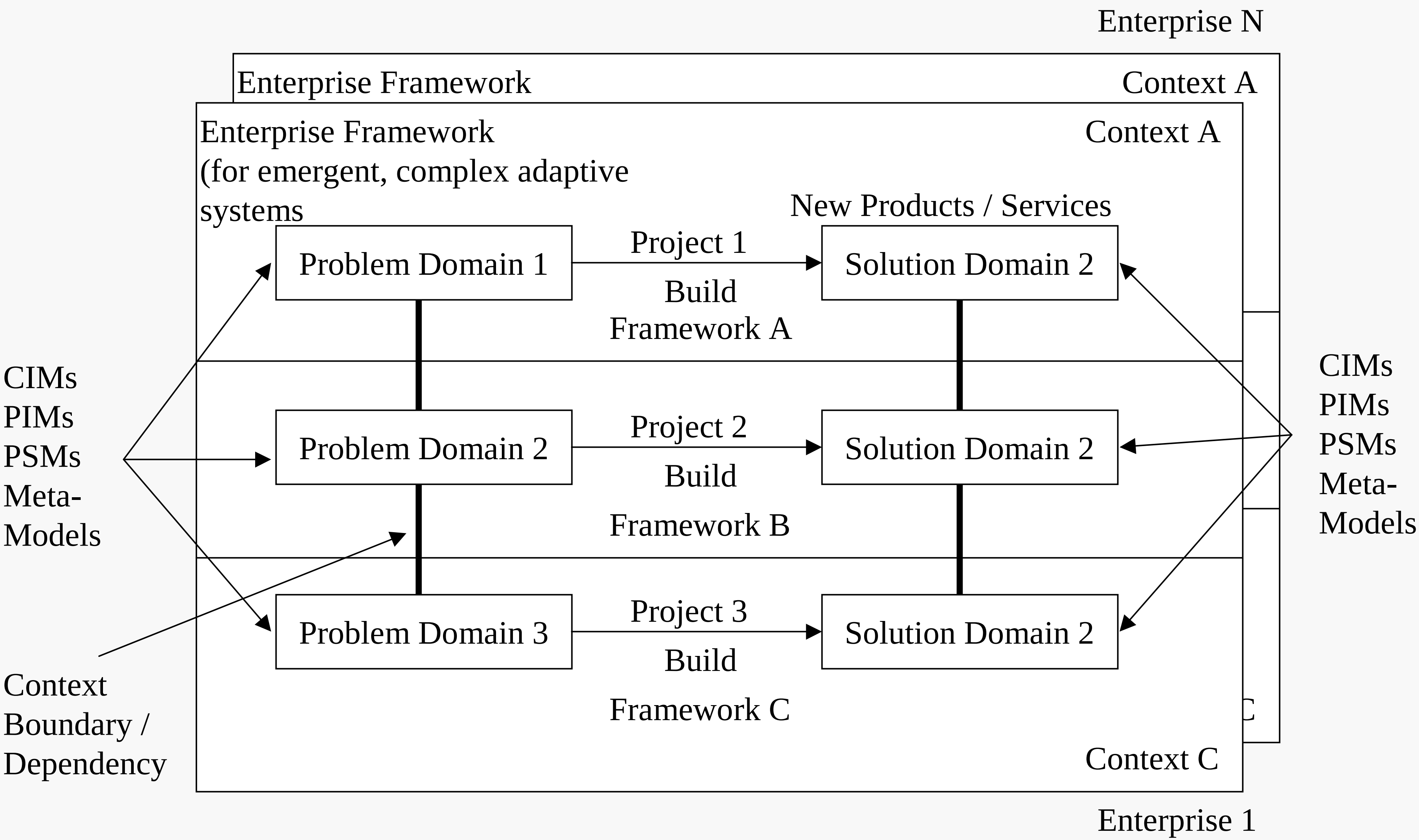
Problem / Solution Sub-domains are addressed concurrently

The Modeling System and Context Boundaries and Dependencies Between Different Problem / Solution Domains



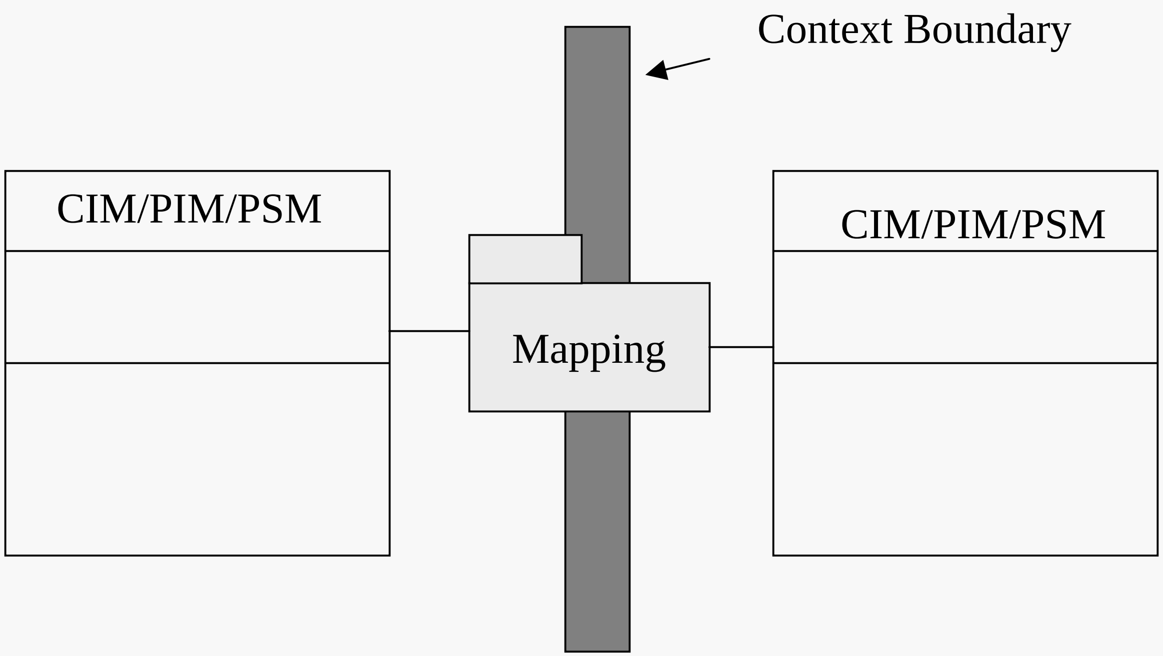
Products and Services, and the projects that build products and services, generally are “instantiated by different teams with different time frames and may utilize different frameworks

The Modeling System and the Number of Context Boundaries and Dependencies Magnified with Federations and the Internet



Cross-Context Model Mapping / Merging Requires Dealing with Context

- CIM-CIM Mappings
- PIM-PIM Mappings
- PSM-PSM Mappings



Advantages of Cross-Context Mapping / Merging

- Ensures consistency within an enterprise, federation
- Allows Validation and Verification at the different levels abstraction – information at lower levels do not confound evaluation at higher levels
- Allows new products / projects to be properly scoped and defined
- Allows the enterprise / federation, as a system to be emergent and adaptable
- Allows for the evaluation of discrepancies in model meaning between contexts
- Etc.

Characteristics/Subsystems of Embedded Modeling Systems in Enterprise, Federated and Internet Systems to Implement the “Modeling Discipline”

- ***Formal*** Modeling Language(s) (may be graphical) that address aspects of different “modeling domains” – e.g. Ontology, Logics, Knowledge, Process, Measures and Metrics, etc.
- “Composition” Tools
- Simulation Tools
- Prediction Tools
- Visualization Tools
- Optimization Tools

Characteristics/Subsystems of Embedded Modeling Systems in Enterprise, Federated and Internet Systems to Implement the "Modeling Discipline" - continued

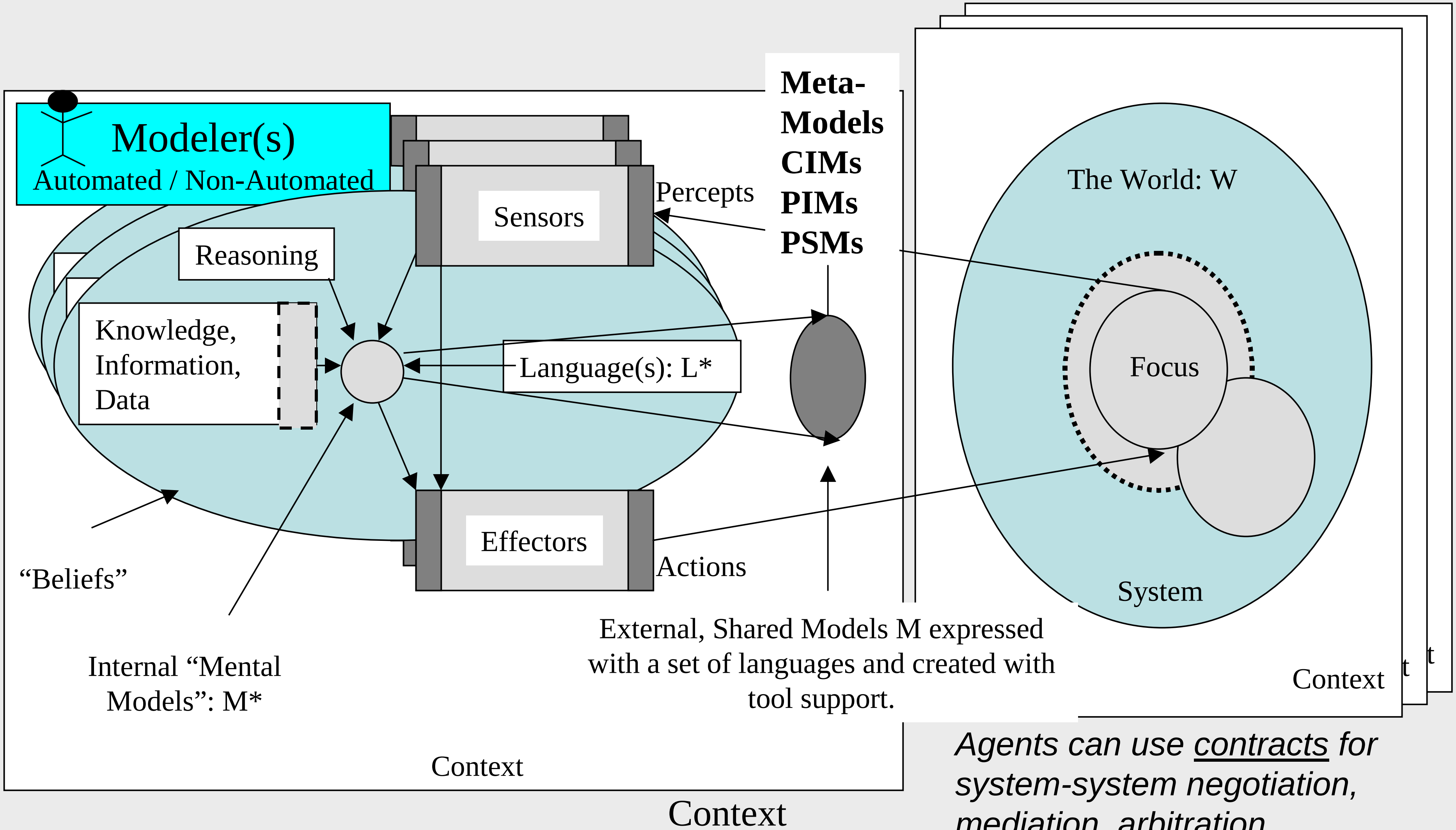
- Evaluation tools (Measures and Metrics)
  - *Semantic Distance, Semantic Cohesion*
  - *Coupling/Cohesion*
  - *Stability / Stationarity*
  - *Fitness, Correctness, Validity*
  - *Scalability, Extensibility, Reliability, Flexibility, Adaptability, etc.*
  - *Trade-off analysis, Return-On-Investment*
  - *Etc.*
- Validation and Verification Tools
- Configuration Tools
- Change Management Tools
- Risk Tools
- Etc.

Characteristics of Embedded Modeling Systems in Enterprise, Federated and Internet Systems

- Integrated with the rest of the Enterprise, Federated, Internet Systems
- Exhibits emergent behavior
- Is extensible, flexible, adaptable, etc.
- Components could be “real-time”
- Components could function as a “model broker”
- Is capable of working with and representing models originating in multiple domains such as Economic, Business, Security, Pervasive Services, Workflow, Ontology, Logic, etc.

Formal System(s) of the modeling system is based on formal systems of those disciplines that are applicable to the creation of models in the context the modeling system is embedded in (Set Theory, Model Theory, System Theory, Ontology, Semiotics, Communications, Context, Logics, Category, Computation, Cognition, Linguistics, etc.)

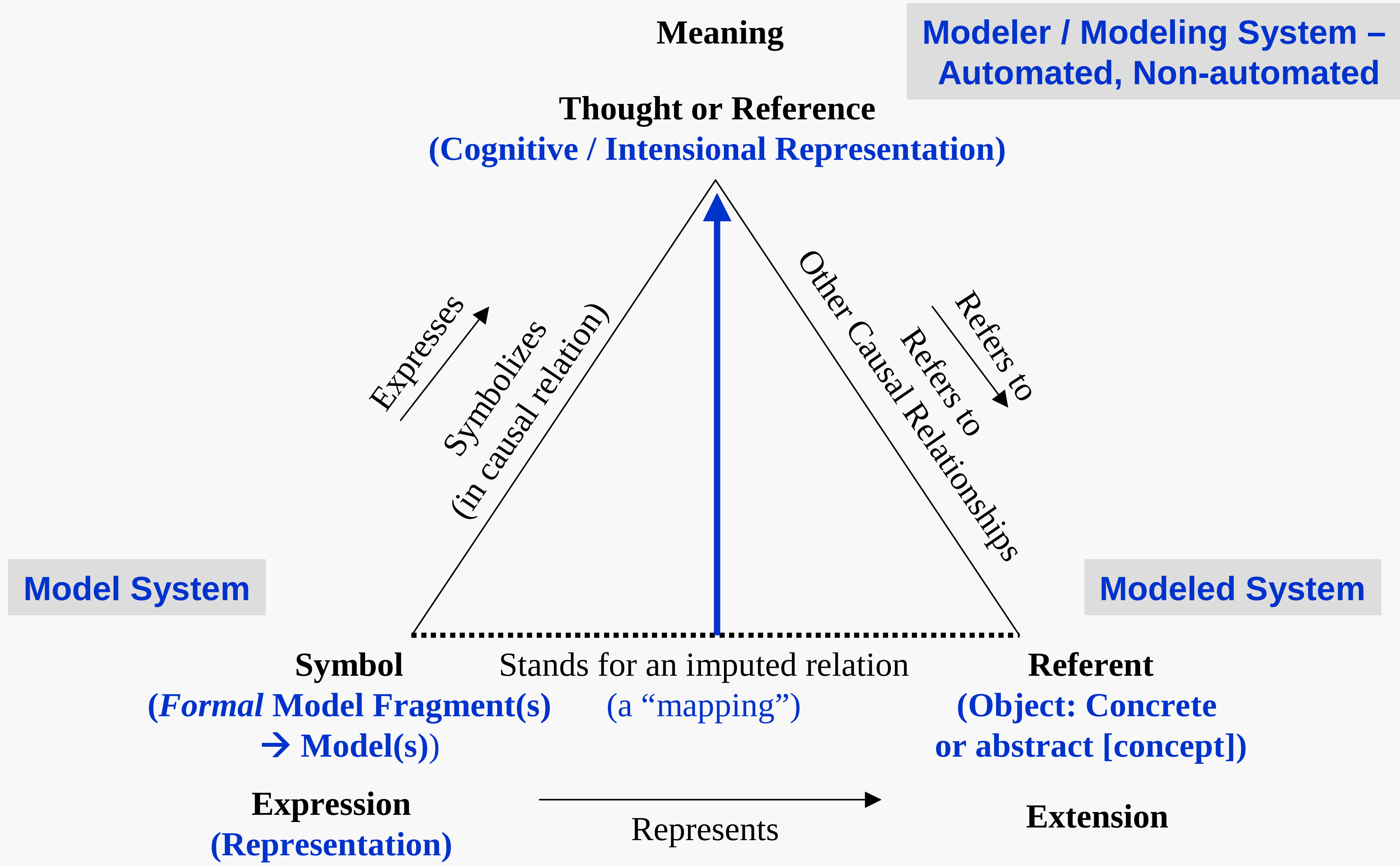
The Modeler is the "Modeling Engine" – The "Core" of the Modeling System"



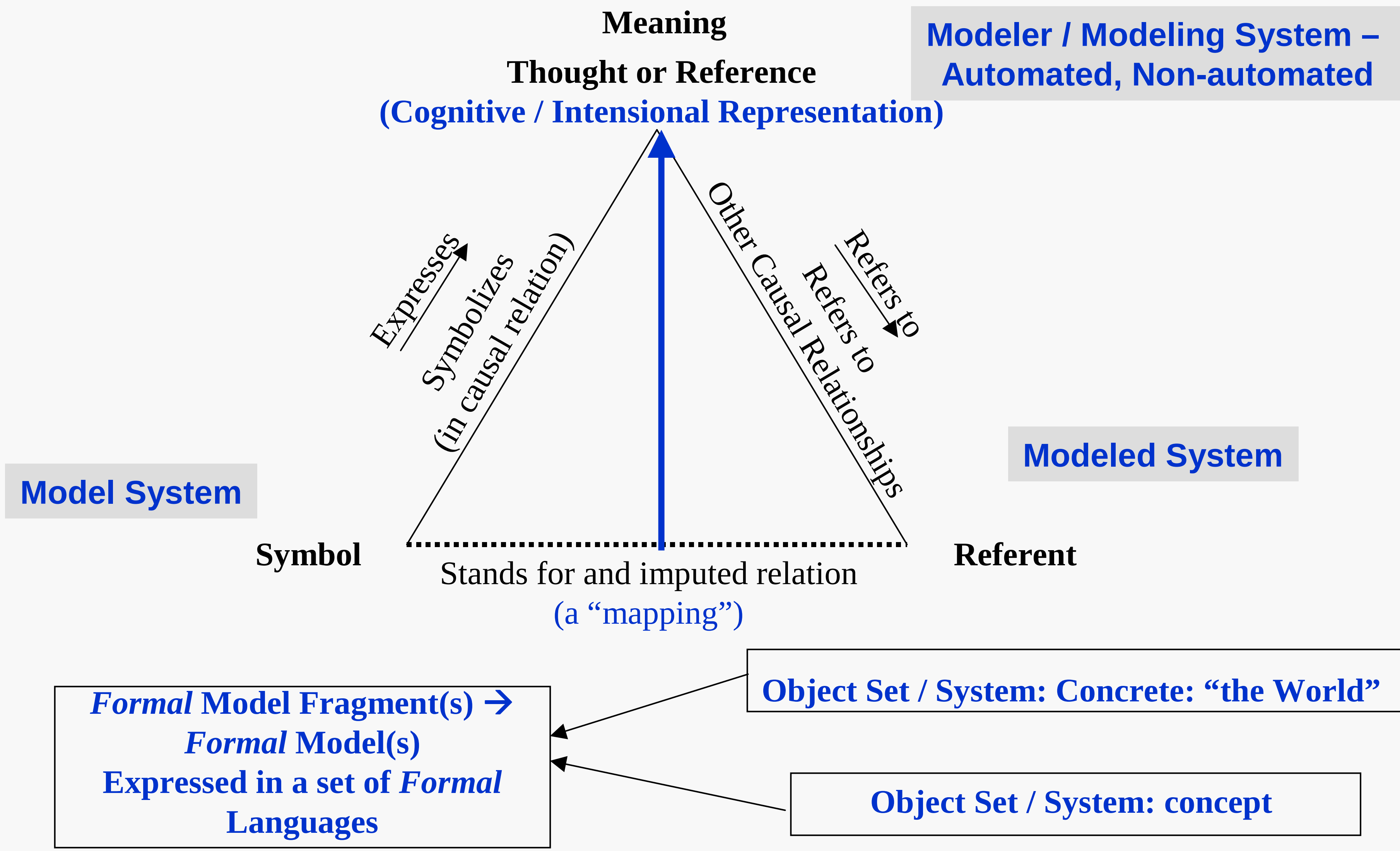
Agents can use contracts for system-system negotiation, mediation, arbitration

Putting It All Together

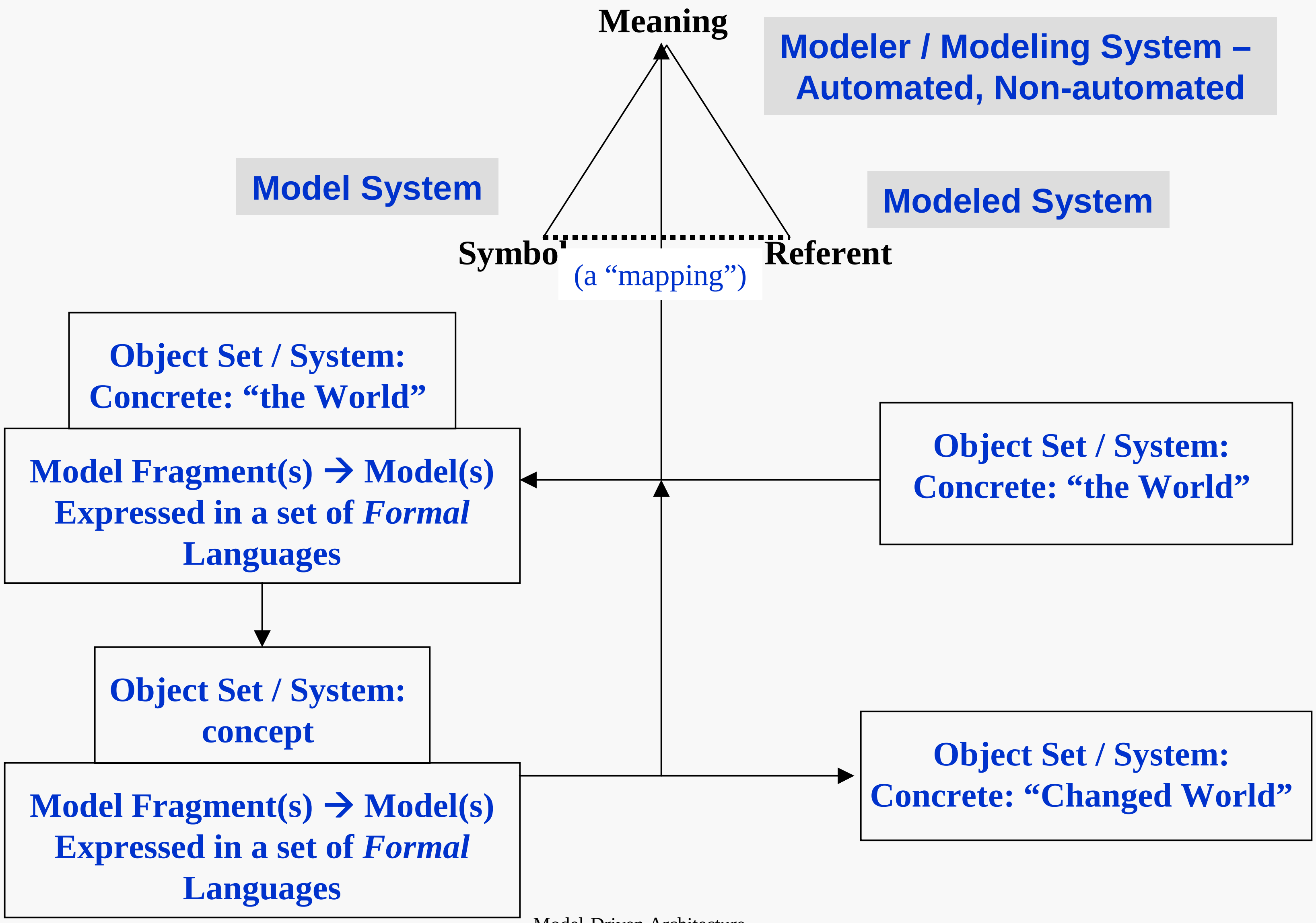
Semiotic Triangle
(“Ogden’s Triangle / Meaning Triangle)



Semiotic Triangle
(“Ogden’s Triangle / Meaning Triangle)



Semiotic Triangle
(“Ogden’s Triangle / Meaning Triangle)



Reference for the Following Slides

$\mathbf{M} = \langle S_W, S_M, R \rangle$

$S_W = \langle W, A \rangle$

$W = \{w_i, C_W\}$ $\quad$ $A: W \rightarrow S_W$

$S_M = \langle M, F \rangle$

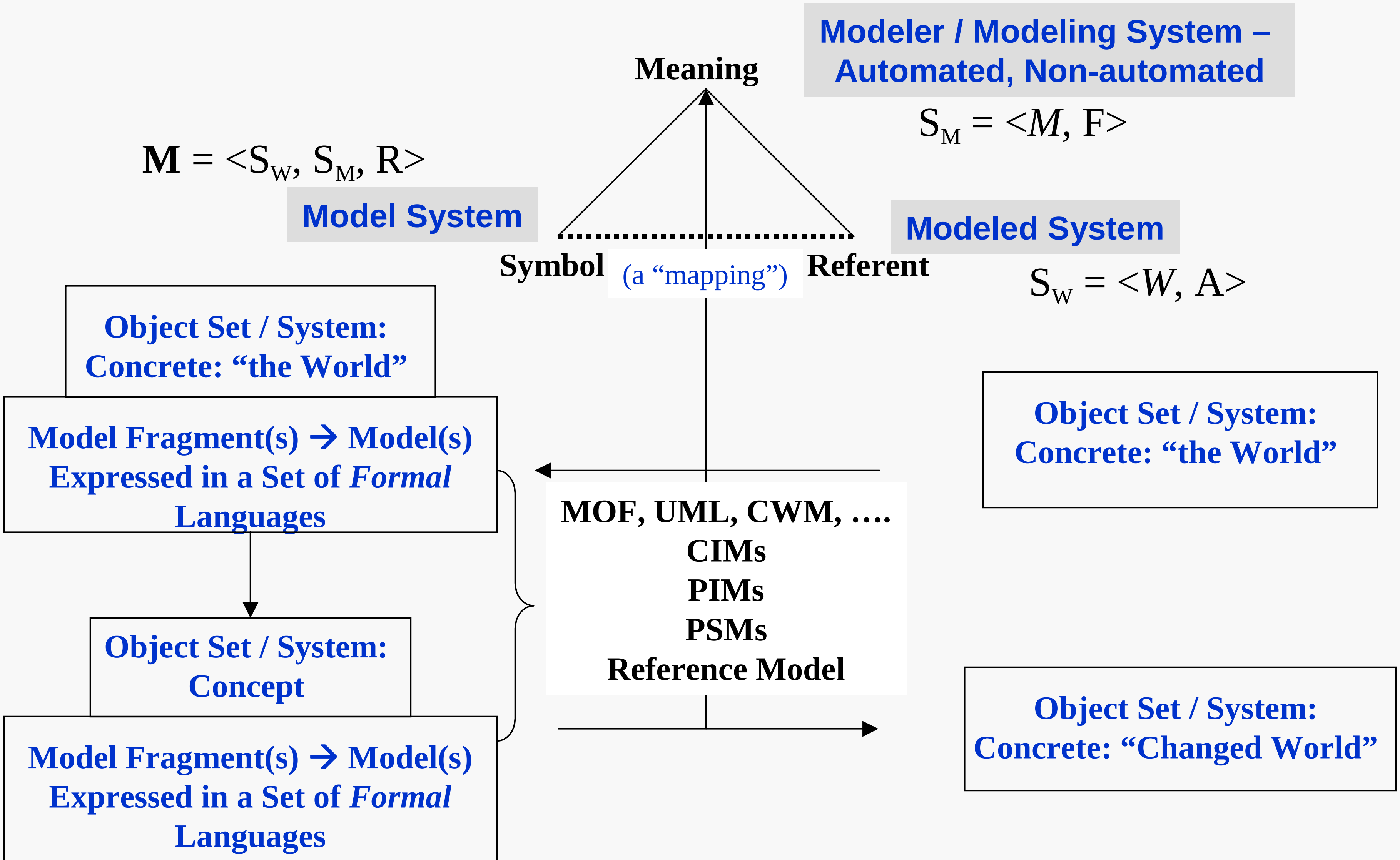
$M = \{m_i, C_M\}$ $\quad$ $F: M \rightarrow S_M$

$R: S_W \rightarrow F$

M: a model
S_W : a modeled system, the world (current or future) of interest
$W = \{w_i, C_W\}$: set of world states as a function of context
A: actions (behavior) of the world
S_M : modeling system
$M = \{m_i\}$: set of <u>internal</u> model states (representations) as a function of context
F: modeling function
R: representation function (e.g. perception, measurement)

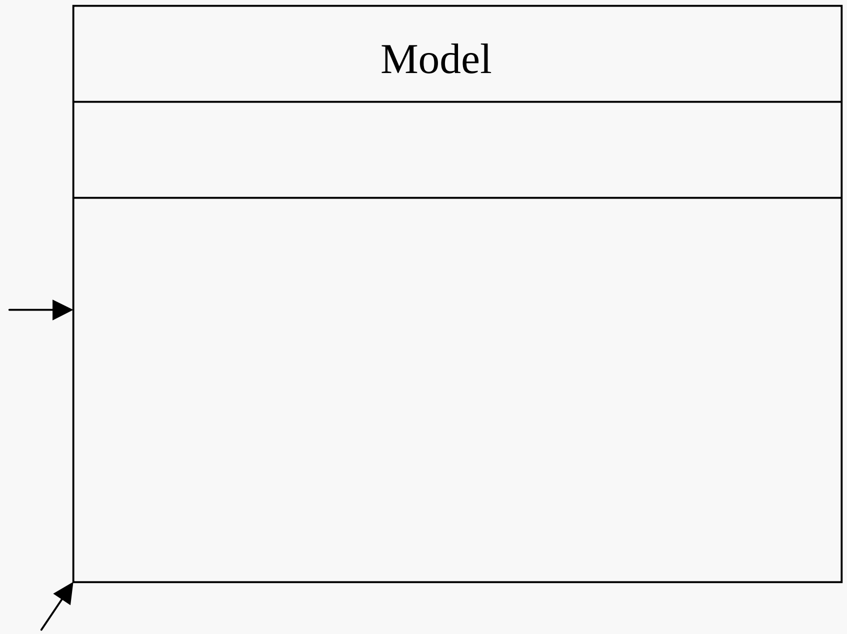
When A, F, and R commute then M is a good model, and the modeling system M can predict the behavior (from observation, or simulation) of the world (modeled system).

Formulizing the Semiotic Triangle



Putting it all Together

Model in Context:
- Goal
- Language
- Viewpoint
- Requirements
- Vertical Domain
- Community of Practice
- Business Domain
- Development Stage
- Problem Domain
- Etc.



$\mathbf{M} = <S_W, S_M, R>$

C → M
M → C

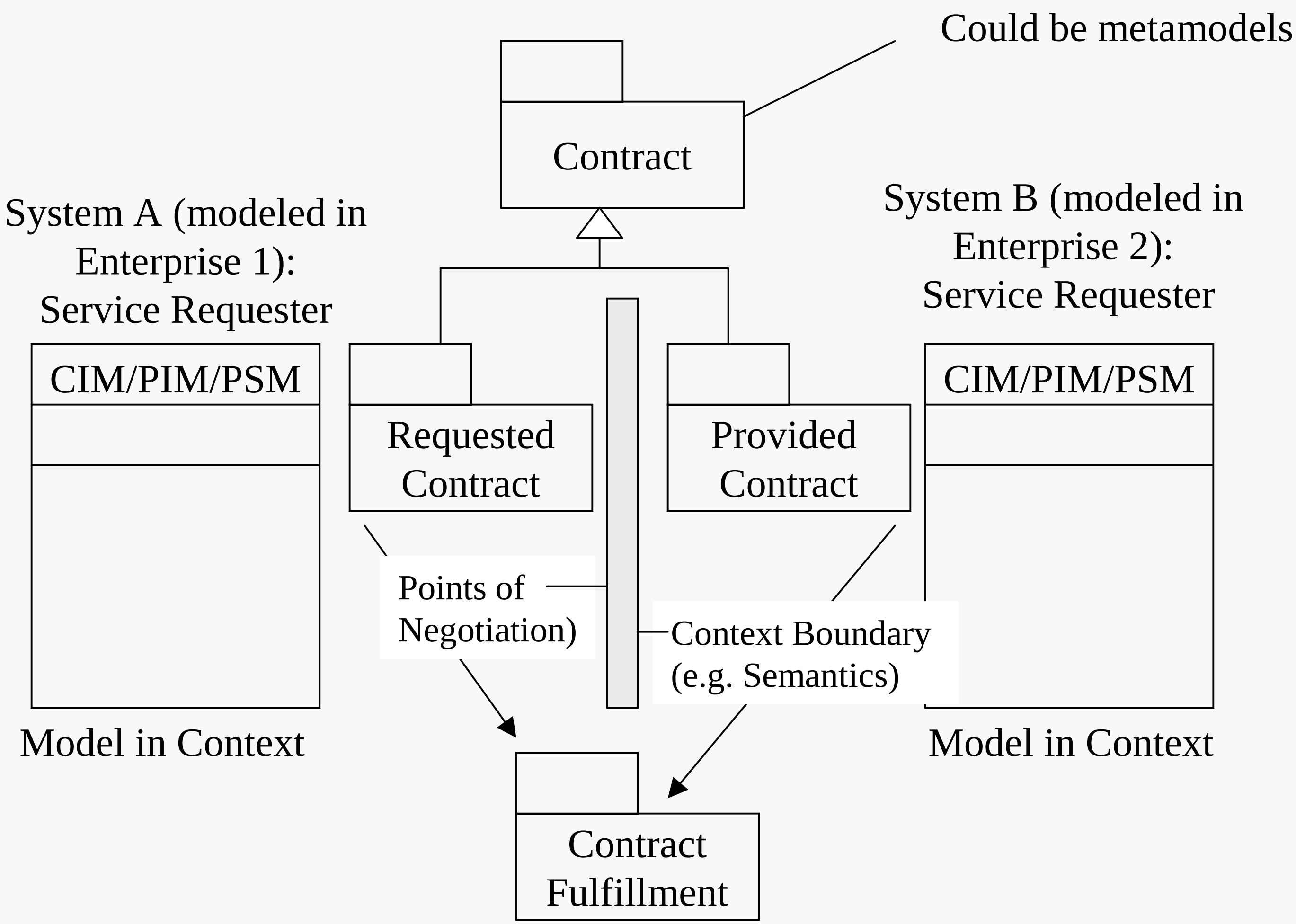
If any attribute is context dependent, then it is possible the meaning of the model may change if the context changes.

A UML Class: e.g. an algebra where $\mathbf{C} = <AT, f_i: i = I>$
Where C – class, AT – set of Attributes (each attributes has a set of values),
F_i – family of operations.

State-space is a subset of the cross product of the attribute domain sets:
$SS\ \underline{C}\ A^n,\ A^n = a_1\ X\ a_2\ X\ \ldots.\ A_n,\ a_i\ C\ A$ (any of which may be <u>context dependent)</u>

Note: This representation assumes that normal sub-setting relation with parent and child classes. If the sub-setting relation is disjoint, for example, the class is better represented with partial functions

Putting it all Together – Contracts in Context



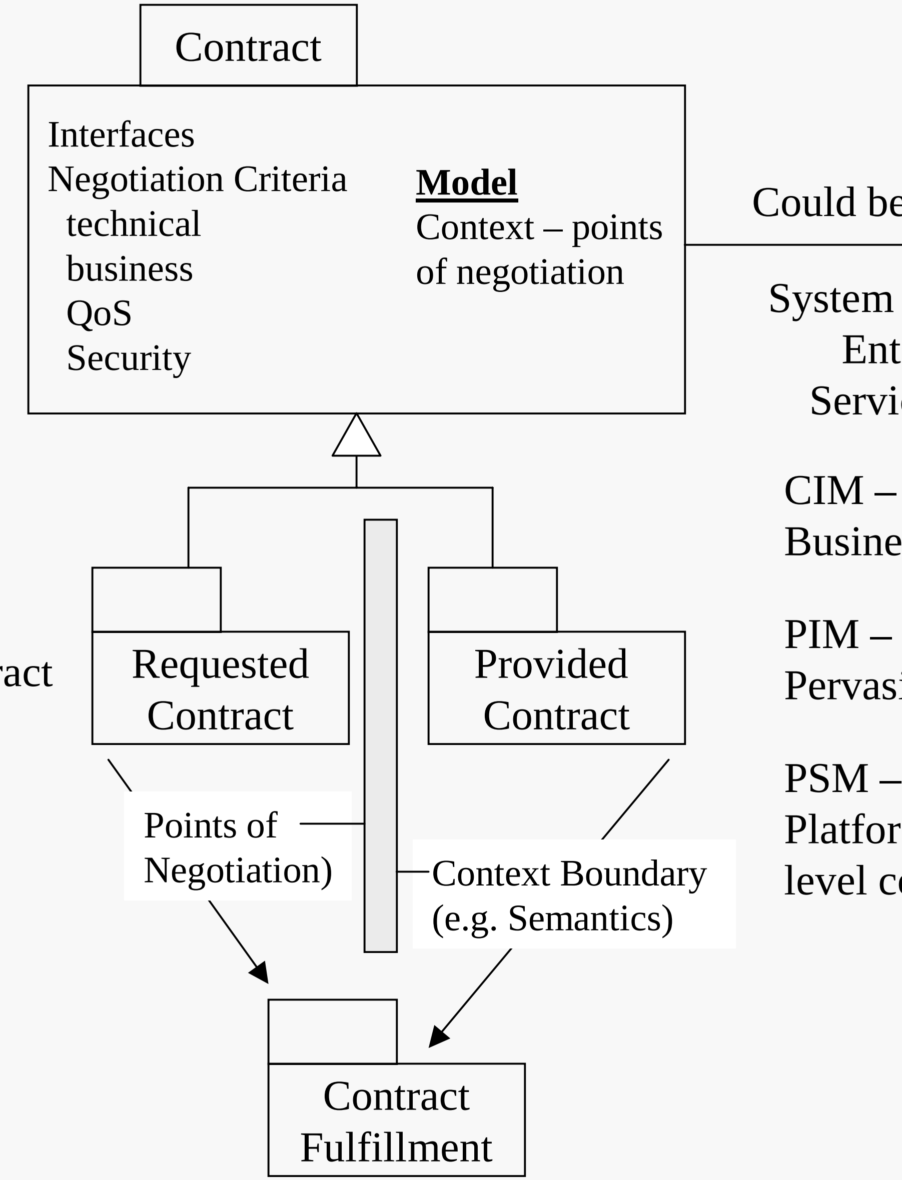
Putting it all Together – Contracts in Context

System A (modeled in Enterprise 1): Service Requester

CIM –
Business level contract

PIM –
Pervasive Service contract

PSM –
Platform / Technology level contract



Could be metamodels

System B (modeled in Enterprise 2): Service Requester

CIM –
Business level contract

PIM –
Pervasive Service contract

PSM –
Platform / Technology level contract

Putting it all Together – Contracts in Context

- Note that there are no explicit model dependencies. If a set of attributes are context dependent, and the context changes, the contract will change (e.g. new negotiation criteria)
- This helps to protect both service requester and service provider when the meaning of their models change as a function of context.

Summary

- Modeling systems embedded in enterprise, federation and Internet Systems should be grounded in the “Modeling Discipline”
- MDA which is embedded in enterprise, federation and Internet Modeling Systems should support the various aspects of the “Modeling Discipline” that is applied to these systems
- These aspects include Ontology and Semiotic Theory Measurement Theory, Logics, Communication Theory, Category Theory, etc.
- The tool set for MDA should support these aspects by providing a set of languages to support these aspects, simulation and prediction capabilities, measures and metrics capabilities, etc.
- Modeling systems embedded in enterprise, federation and Internet systems should be able to handle multiple problem / solution domains concurrently. This includes the ability to trace, validate and verify across these domains to help ensure that the enterprise, federation, Internet is emergent, adaptable, etc.
- Meaning of models is context dependent – context needs to be explicitly addressed
- Contracts are a good way to buffer the effects of changing context

Contact Information

Matthew K. Hettinger, President and Chief Architect/Engineer
Mathet Consulting, Inc. PMB 14004
1450 E. American Lane
Schaumburg, IL 60173
847-330-6375
hettinge@mindspring.com