

OMG's MDA TM

An Architecture for Modeling



Enabling Model-Driven Integration TM

Desmond DSouza
President

Kinetium

desmond@kinetium.com



www.kinetium.com

About the Speaker and Kinetium

Desmond D'Souza is President of Kinetium. He is co-author and developer of the *CATALYSIS* method, published by Addison Wesley in 1998, and is a respected authority and speaker at companies and conferences internationally. He was previously senior vice president of component-based development at Platinum Technology and at Computer Associates, working on methods, tools, and architectures for component-based development. He founded ICON Computing, an object and component technology methods and services company that was acquired by Platinum in 1998. Mr. D'Souza has worked with object and component technology since 1985.

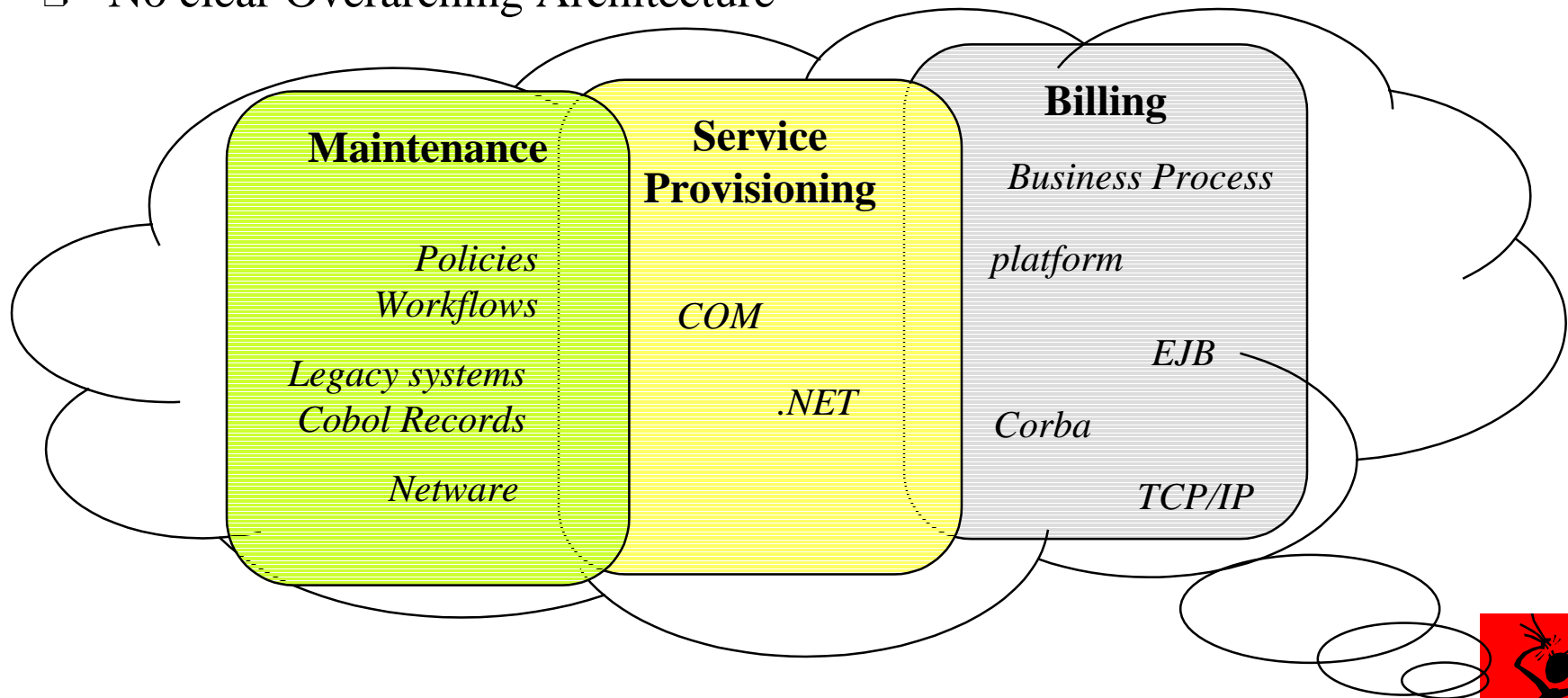
Kinetium is an Austin, Texas startup providing solutions for model-driven component-based development and integration architectures, methods, and tools. To learn more about the strategies, methods, modeling, architecture, and technology of component-based development and e-Business, and how Kinetium can help your company, you can contact Mr. D'Souza at **desmond@kinetium.com**

Copyright and Usage Terms

- These materials are © 2001 by Kinetium. If you wish, you may select slides from here and use them within your presentations for non-commercial purposes, provided that (a) you use those selected slides in their entirety, without any modification, including the copyright notice and URL, and (b) you include the following among your references;
“OMG’s MDA – An Architecture for Modeling, Enabling Model-Driven Integration”, Desmond DSouza, Kinetium, www.kinetium.com
- OMG MDA™, MDA™, OMG Model Driven Architecture™, Model Driven Architecture™, "The Architecture of Choice for a Changing World™" and Unified Modeling Language™ are trademarks of the Object Management Group.

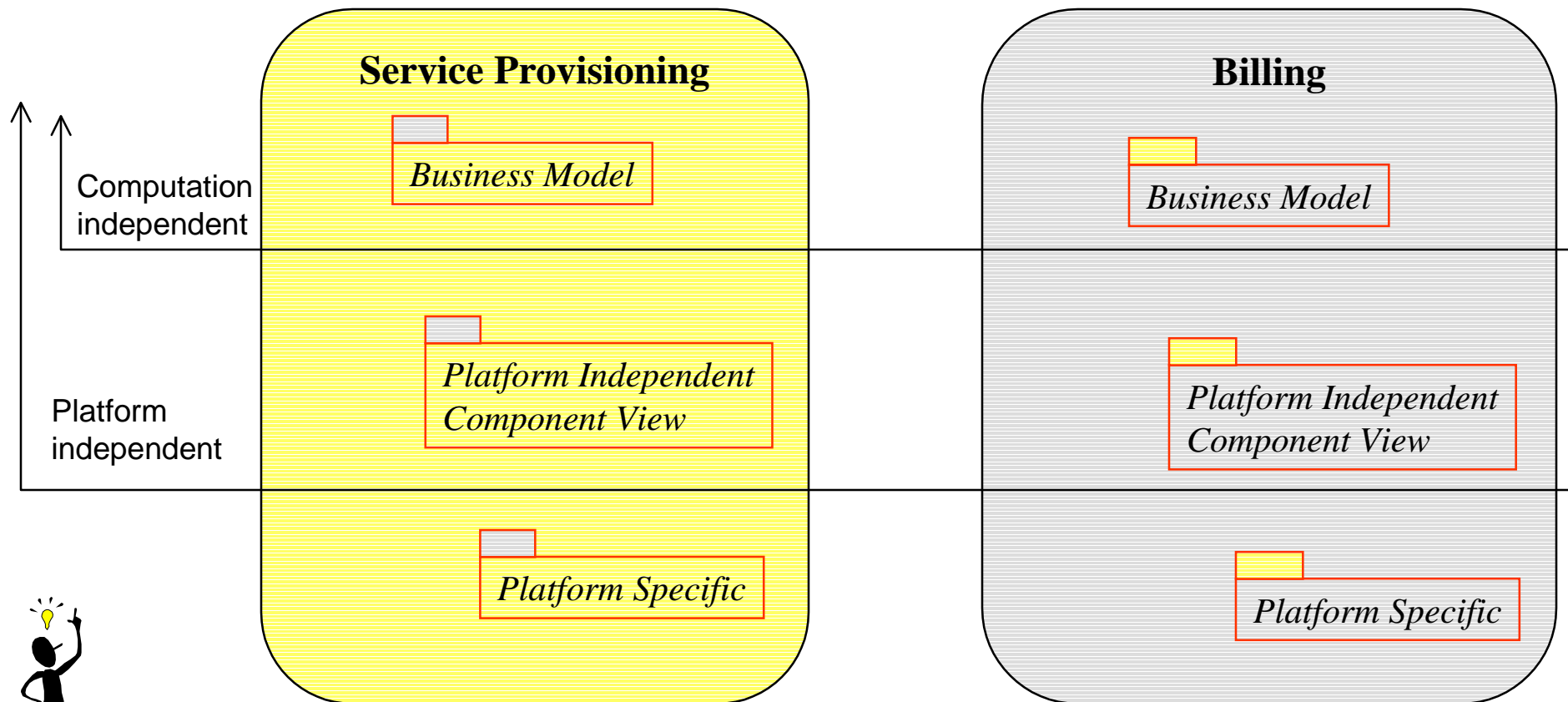
UML™, Corba®, .NET, J2EE, ebXML, and all

- ☐ There are a huge amount of software descriptions – UML, IDL, Java, XML, ...
 - Many different subject areas or domains
 - Many levels of detail, separating and mixing of concerns, diverse languages
 - Many overlaps and relationships between these
- ☐ No clear Overarching Architecture



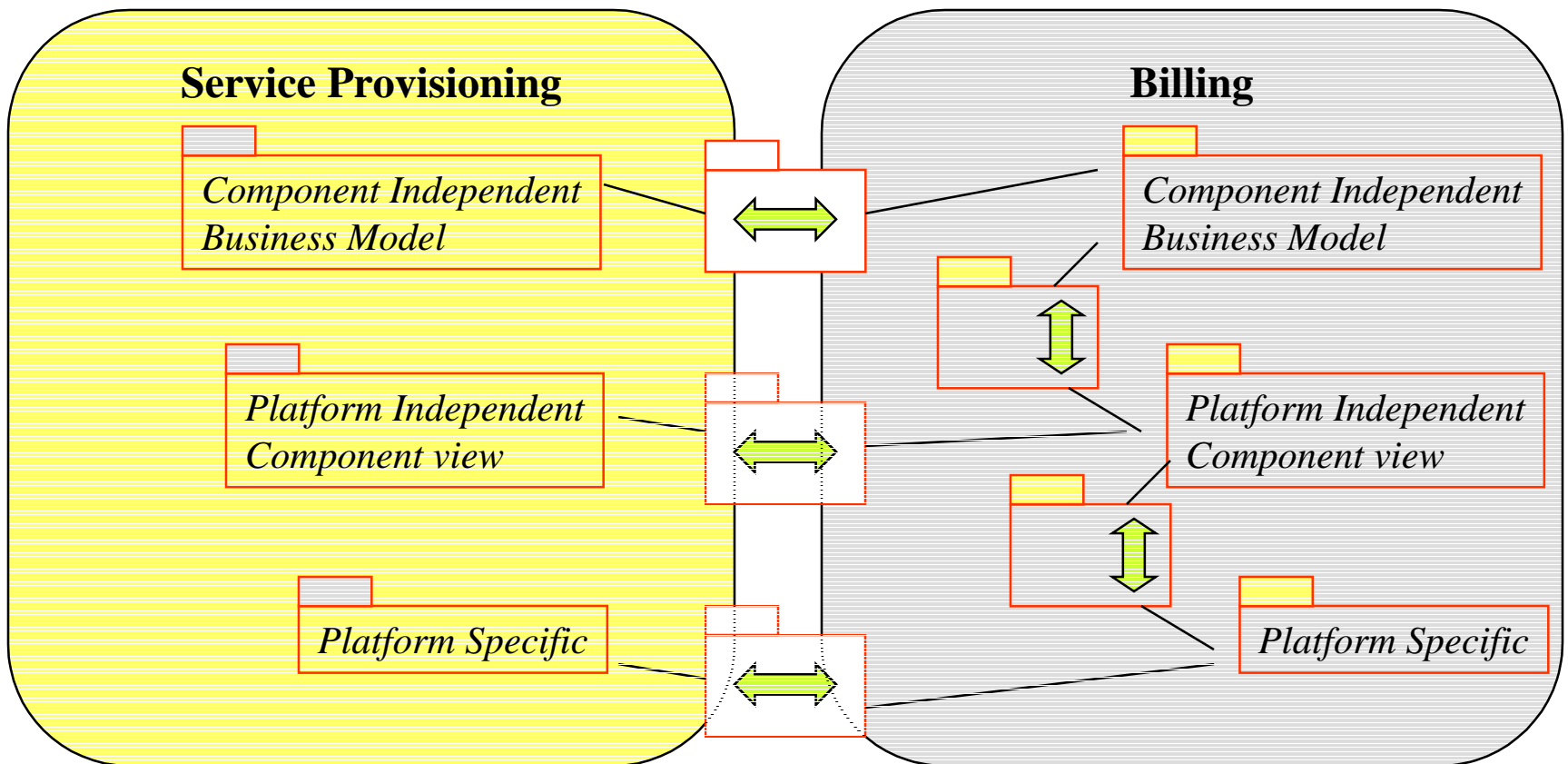
MDA brings Consistent Separation

- Clear separation of different kinds of computation- and platform-independent models
 - There can be more than one viewpoint or degree of detail with each level
 - PIM – platform independent component model; PSM – platform-specific component model



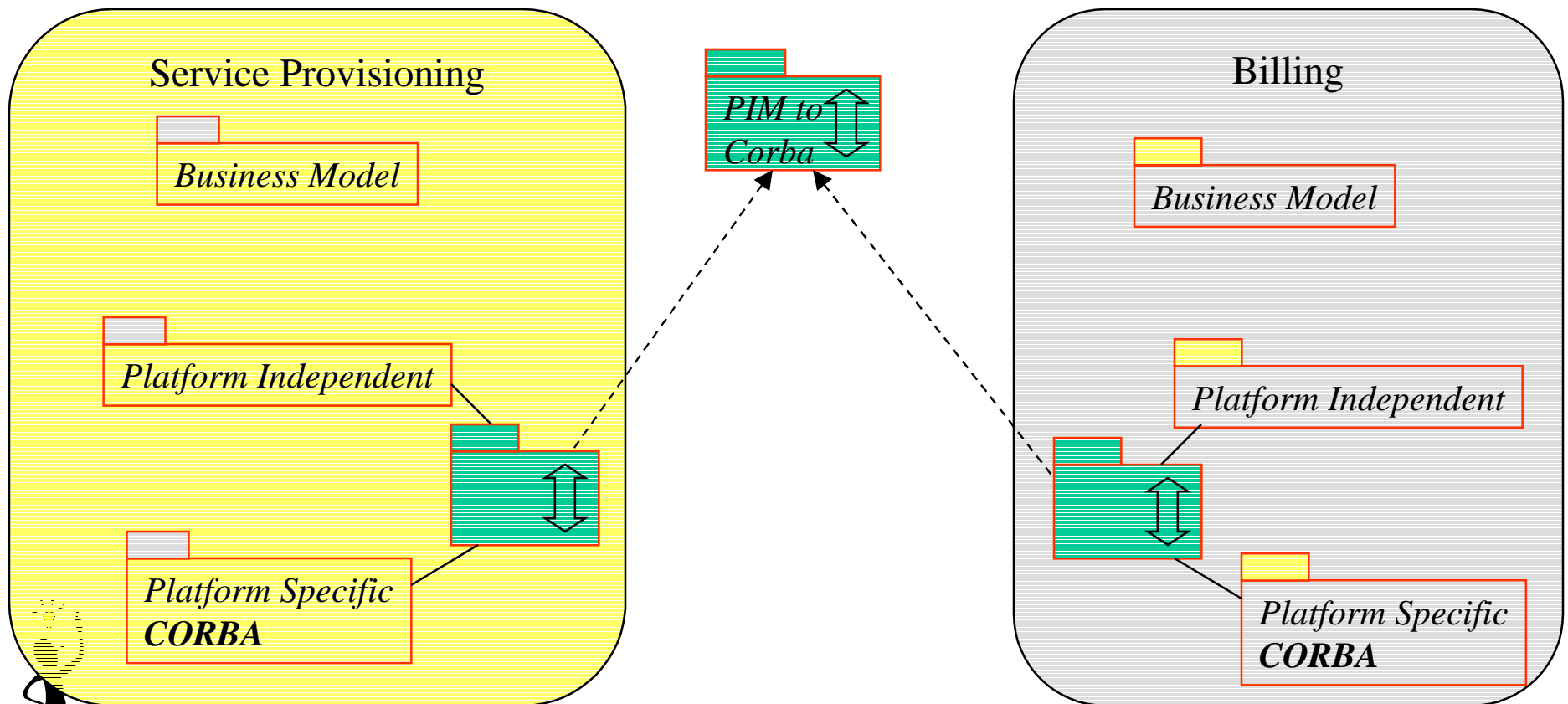
MDA brings Consistent Relationships

- Clear relationship relating different models
 - Platform independent components to platform specific models – a refinement (“vertical”) mapping
 - Business model to Platform independent components – a refinement mapping
 - Business area A to Business area B to model the overlap – a “horizontal” model integration
 - Interface A to interface B of the same component – a “horizontal” model integration



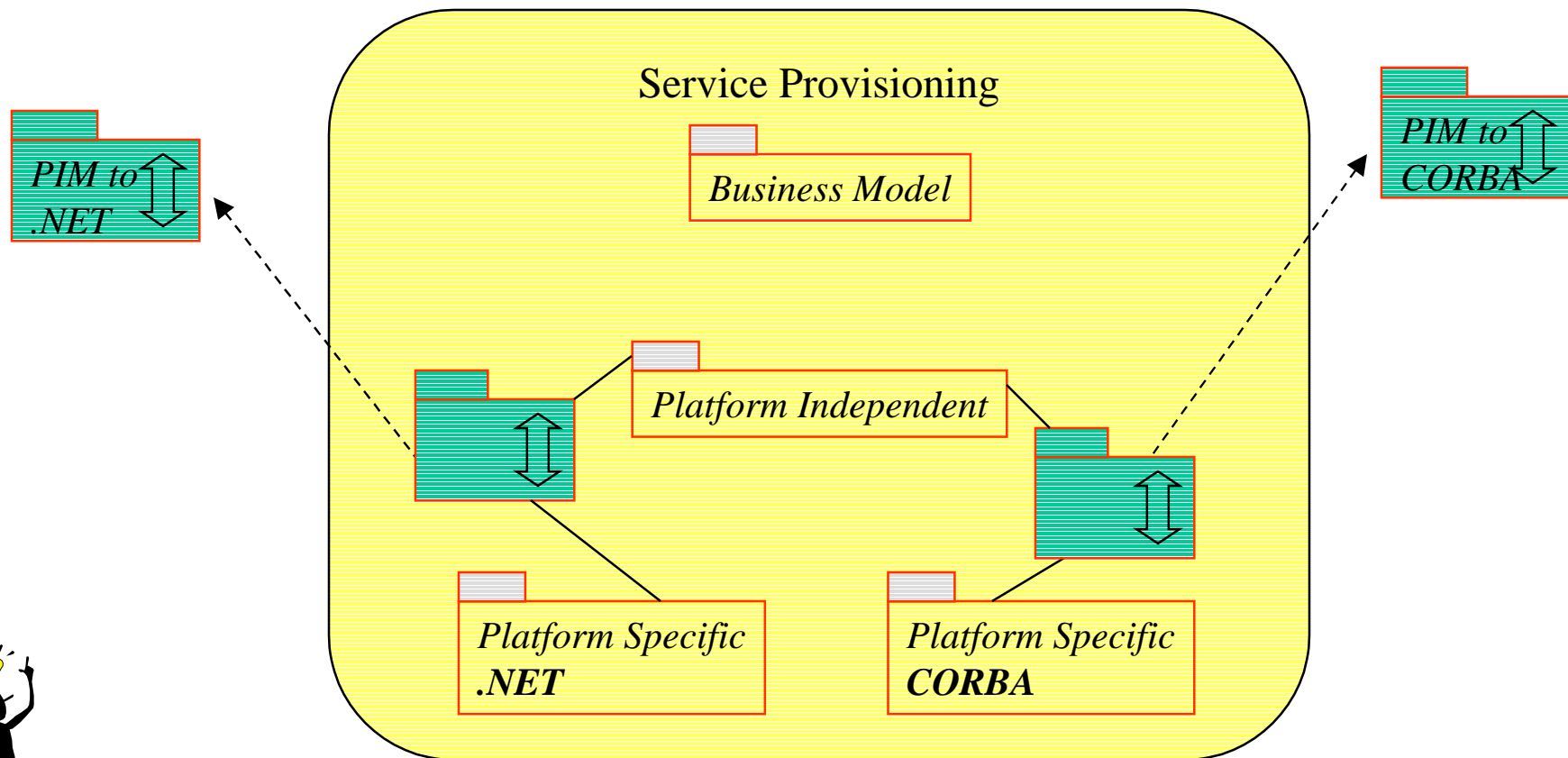
MDA brings Shared Standard Mappings

- Standardized mappings and patterns for platform-specific model generators
 - These generic mappings and patterns are defined in shared packages
 - A specific application uses (perhaps is even generated from) the shared package



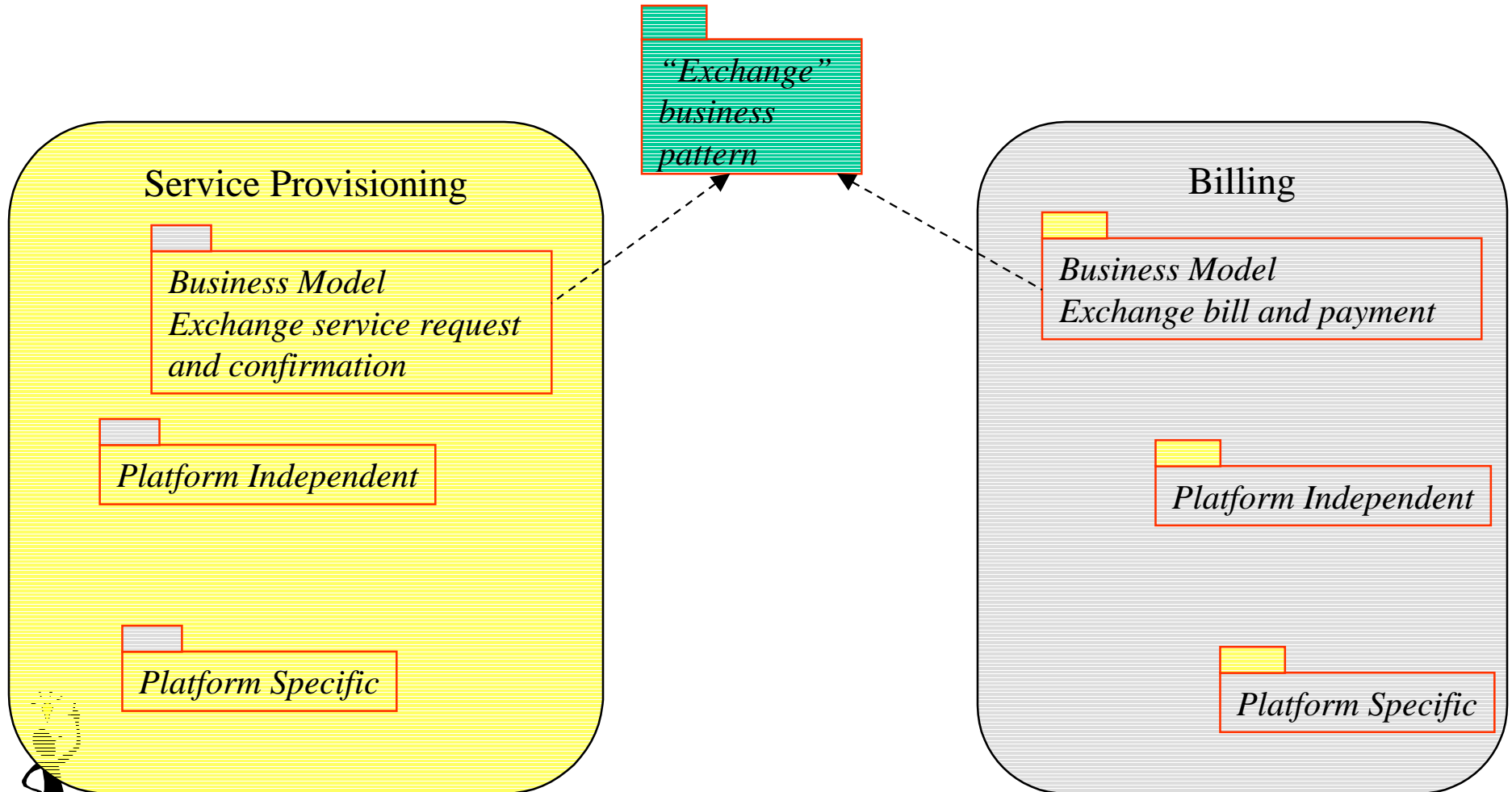
MDA brings Improved Portability

- Simpler to re-target platform-independent to different platform targets
 - Common mappings to different target platforms



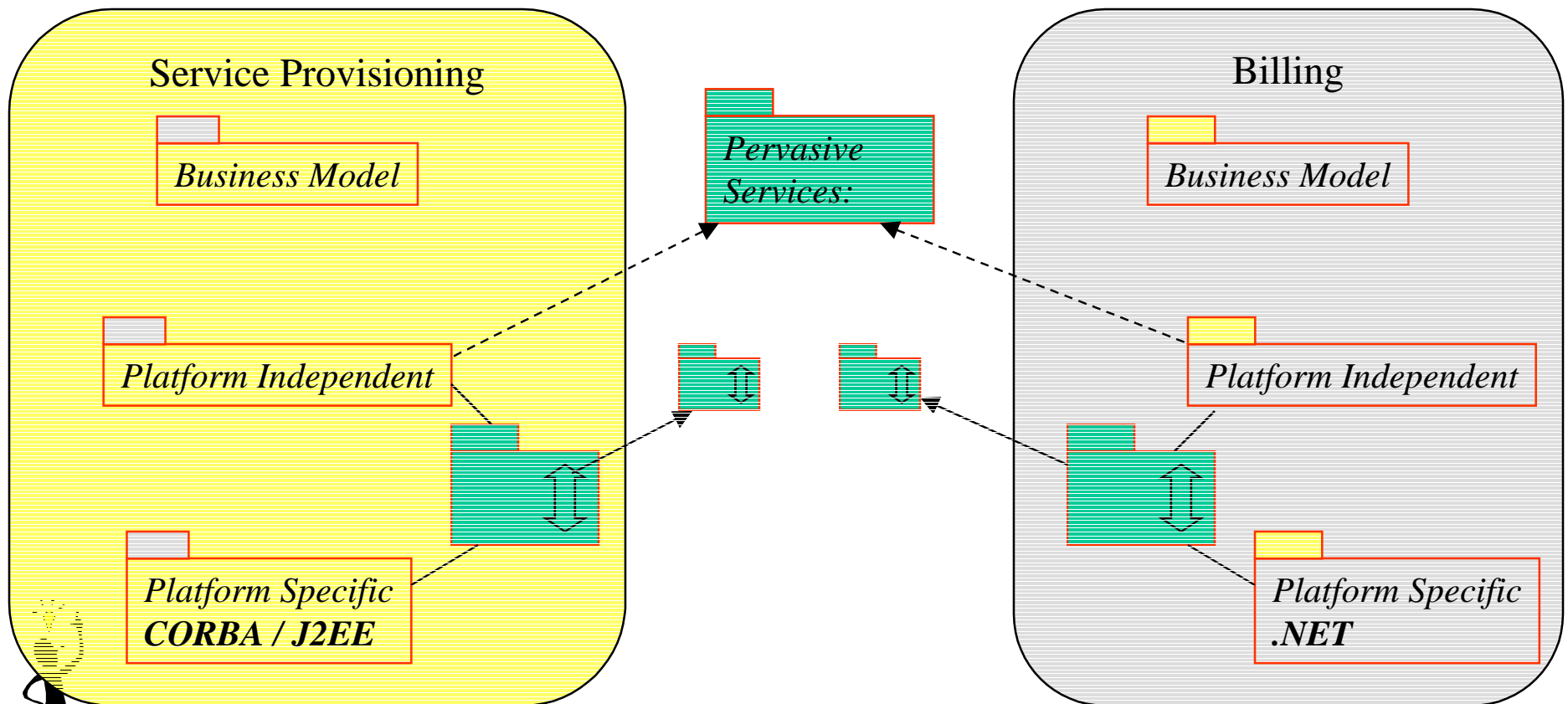
MDA brings Shared Standard Patterns

- Common patterns at any level of modeling
 - The idea of common patterns applies to models of business, components, platforms, mappings



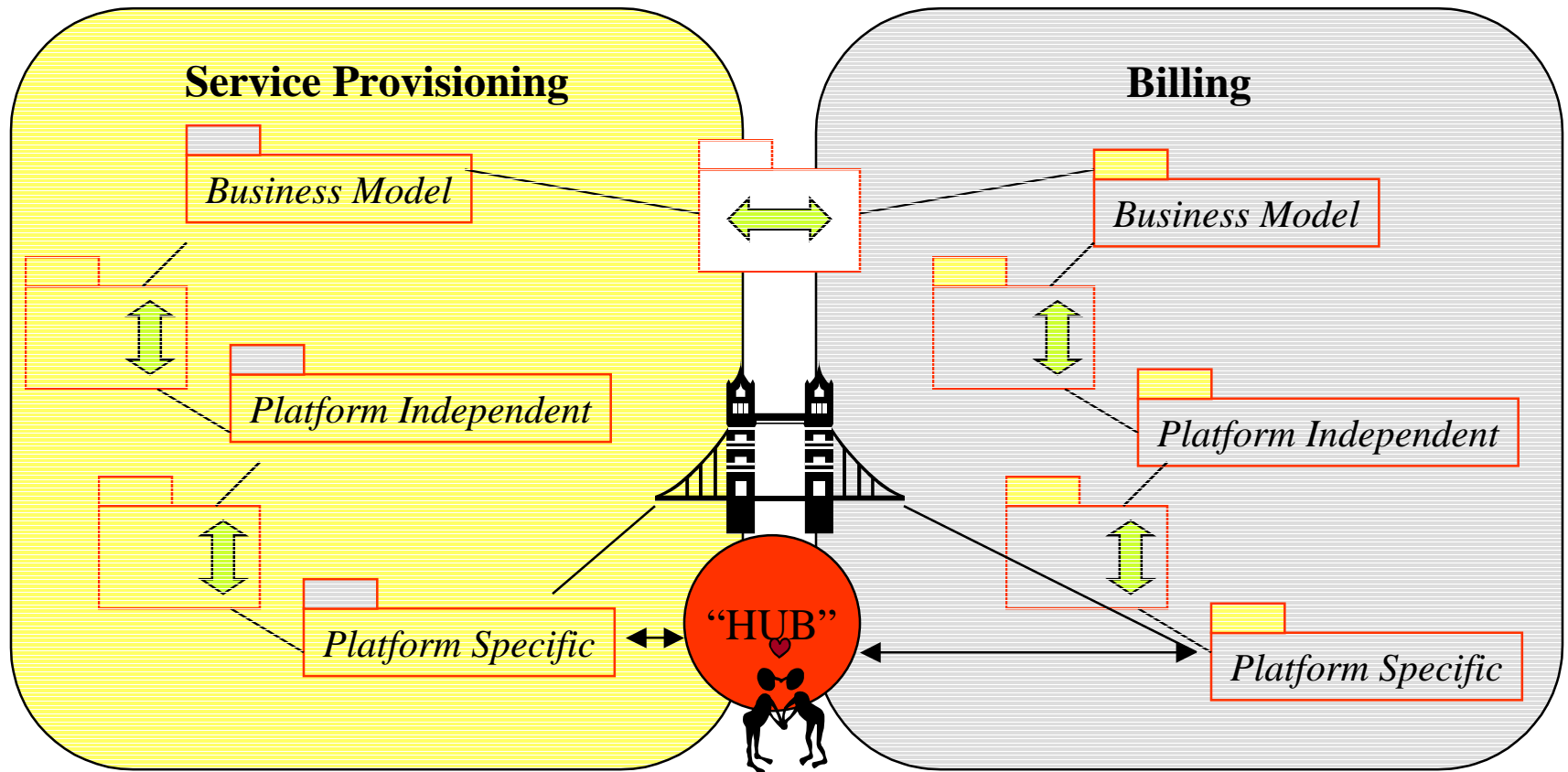
MDA Integrates Across Computing Platform

- Pervasive Services = Common platform-independent model of platform services
- Leverages shared standard mappings to platform-specific realizations of services



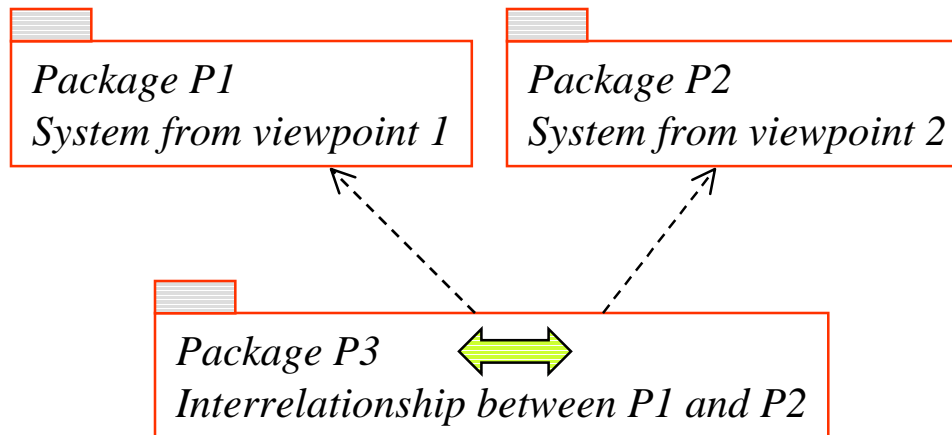
MDA enables Model-Driven Integration

- Design-time or Run-time mediator has access to model structure
 - Utilize inter-model refinement mappings from concrete up to more abstract levels
 - Works with given inter-model relationships or “horizontal” mappings at abstract level
 - Generate data, protocol, service, and business process adaptors

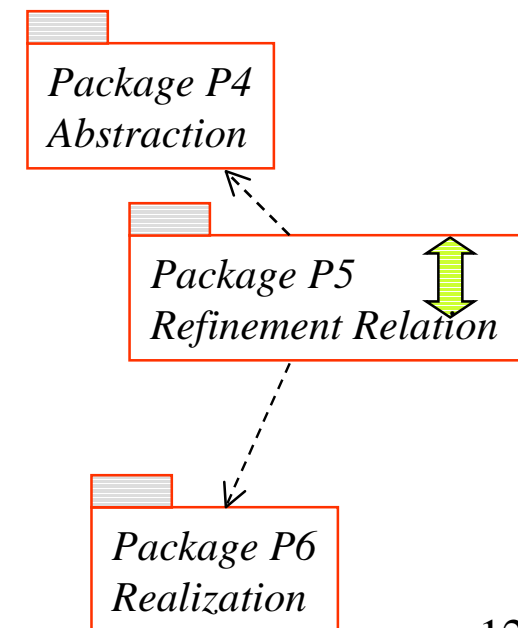


Foundation – Fractal Inter-Model Relationships

- Two different viewpoints on the same system or component
 - Their overlap is modeled – common objects, events, attributes, etc.

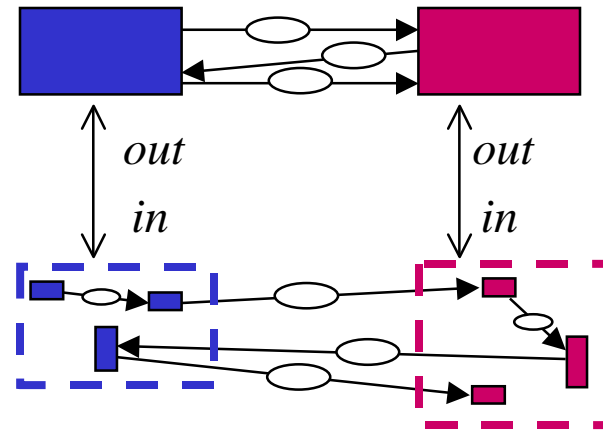


- Two different levels of abstraction of the same system
 - The refinement relationship between levels is modeled



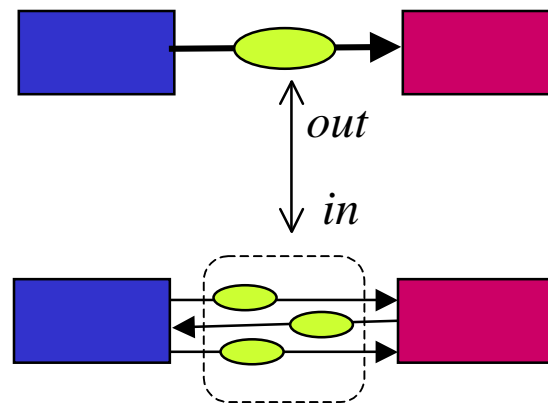
Foundation – Fractal “Zooming” in and out

- Abstract multiple objects as a single larger-grained object



(a) Zooming in/out – objects

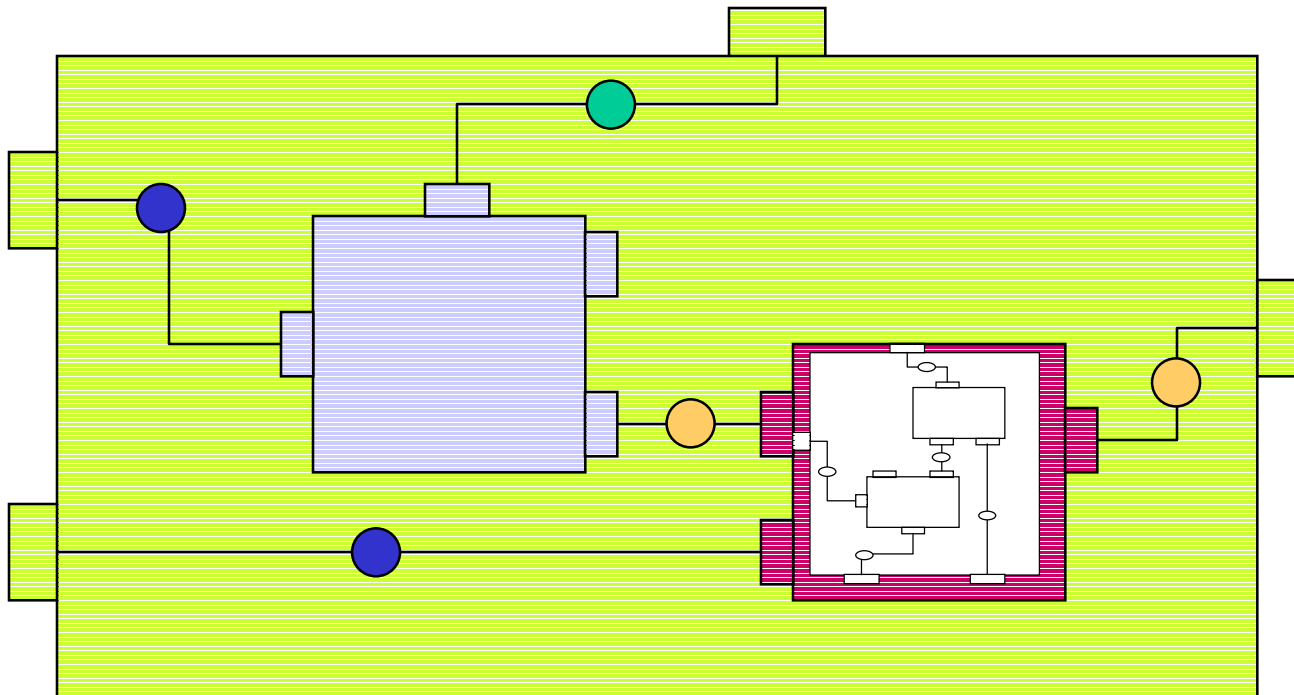
- Abstract detailed interaction protocols as a single action



(b) Zooming in/out – interactions

Foundation – Fractal Assembly by Composition

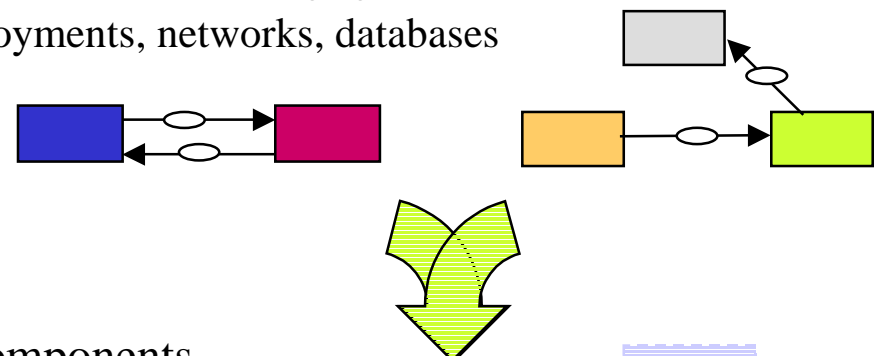
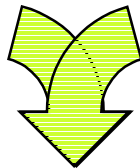
- Uniform construction of assemblies from other parts
 - Component, port, connector, assembly for all variety of design artifacts
 - Assembly has separate external specification of ports Vs. its internal design



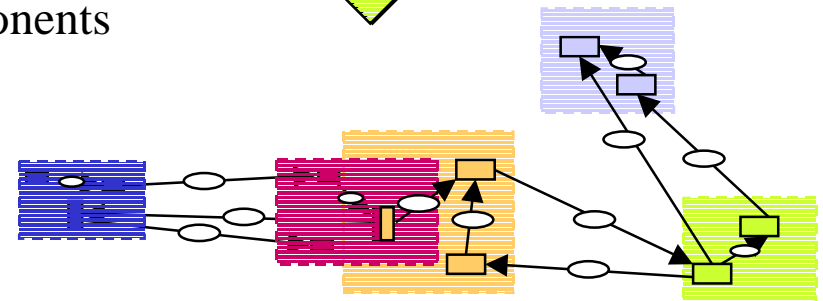
Next – Compose Business Specifications?

Example inspired by Kevin Tyson

- Precise patterns of domains and processes
 - Commodities trading – orange juice, pork bellies, ... electricity, drinking water?
 - Production – auto parts, ... orange juice, pork bellies, ... electricity, water water?
 - Degradation – auto parts, orange juice, pork bellies, ... electricity, drinking water?
- Compose models, business specifics process and rules, patterns
 - Futures Trading [commodity → electricity, payment → drinking water futures]
AND
 Consumption [consumable → futures electricity purchase]
- Compose patterns of refinements / transformation / architectural styles
 - Styles can combine aspect-based code weaving, reflection, schema merging, ...
- Compose directory services information on deployments, networks, databases



- Implicitly integrate and evolve software components
 - Specialize and Configure
 - Generate plug-ins, bridges, adaptors
 - Connect together
 - Migrate data





Kinetium's Endorsement of MDA

- Kinetium is a new company in Austin, Texas, focused on
 - Model-Driven Integration™
 - Architecture-centric development
 - Shared software assets at all levels
 - Light-weight versions of precisely defined development methods
 - Solutions through consulting, training, development process, and products
 - Contact us for more information at info@kinetium.com

“MDA is an important step for enterprise software systems. It will support a new generation of full-lifecycle model-driven tools and methods. At Kinetium we have a focus on Model-Driven Integration and Synthesis, and our light-weight yet precise approach to exploit models to develop and integrate heterogeneous enterprise and cross-enterprise systems will complement the OMG's MDA very well. We are excited about MDA.”

*Desmond D'Souza
President, Kinetium*