Model Driven Solutions

Where Business Meets Technology

A division of Data Access Technologies, Inc.

Architectures and Ontologies for Business Value

Ontology Summit 2011 – Making a case for ontologies

Cory Casanave

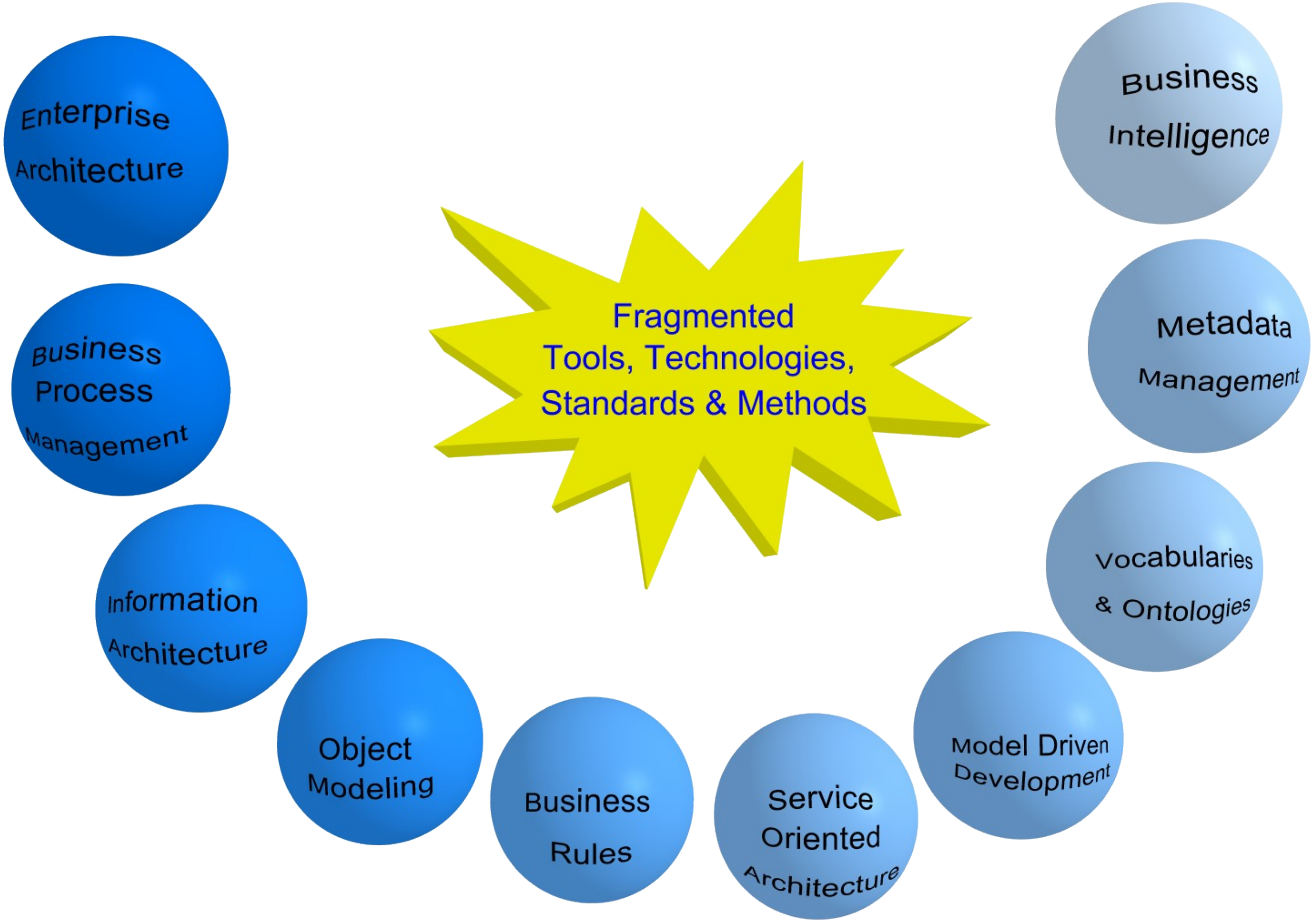
cory-c (at) modeldriven.com

Febuary 2011

The Issues

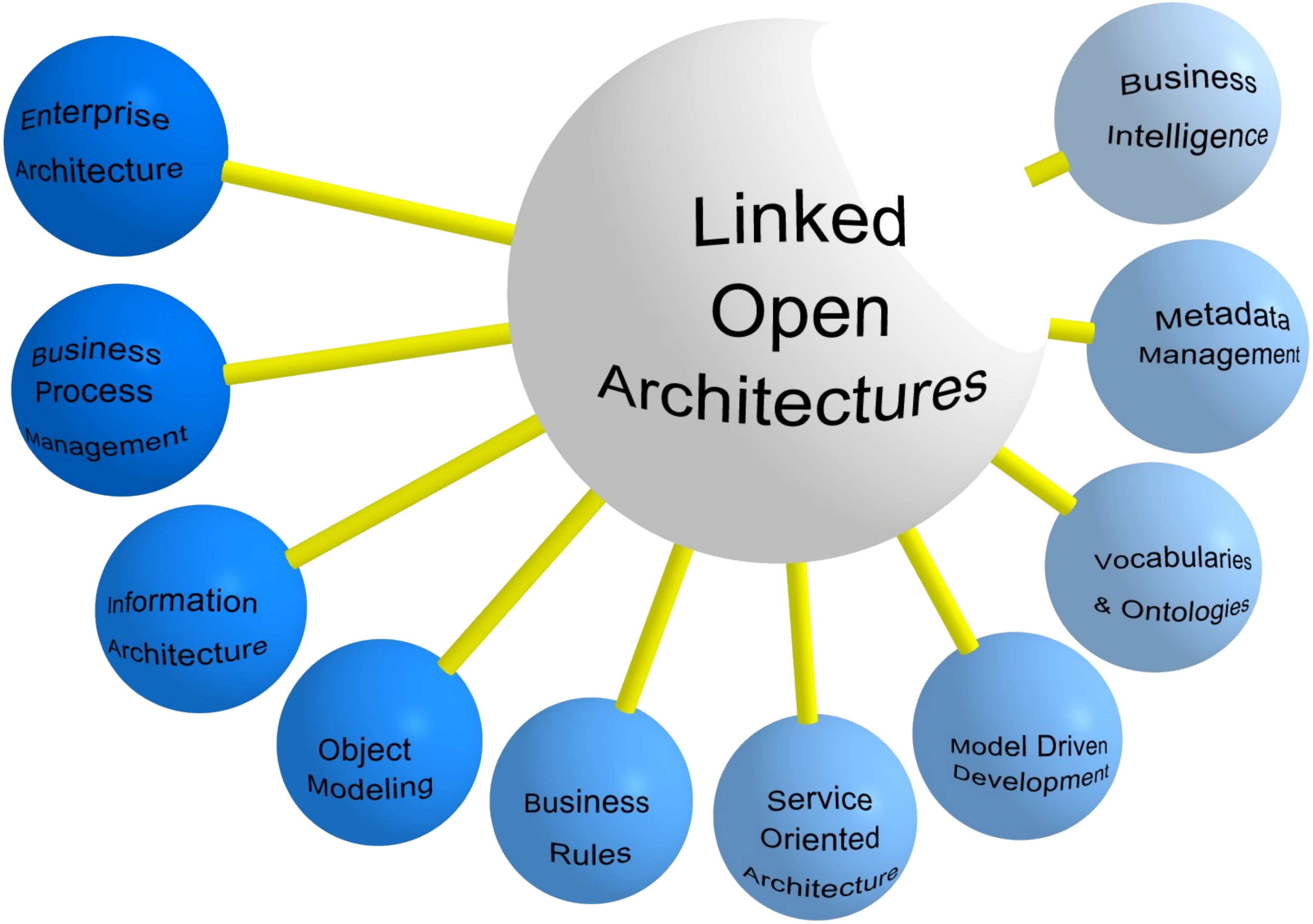
- Organizations are complex entities involving human, organizational and technological elements that, to be effective, must work together.
- Business and systems concerns are often misaligned and do not work together effectively
- An architecture provides a view for how these elements work and work together as an integrated system – yet the architectures are disjoint
- The entire architecture must evolve as necessary to meet both new business requirements (e.g., market changes, regulation changes, etc.) and new technical approaches (e.g., Web-based delivery, service-oriented architecture, etc.).

Fragmented Architecture Domains



Fragmented
Tools, Technologies,
Standards & Methods

Unified Architectures



A Strategic Opportunity

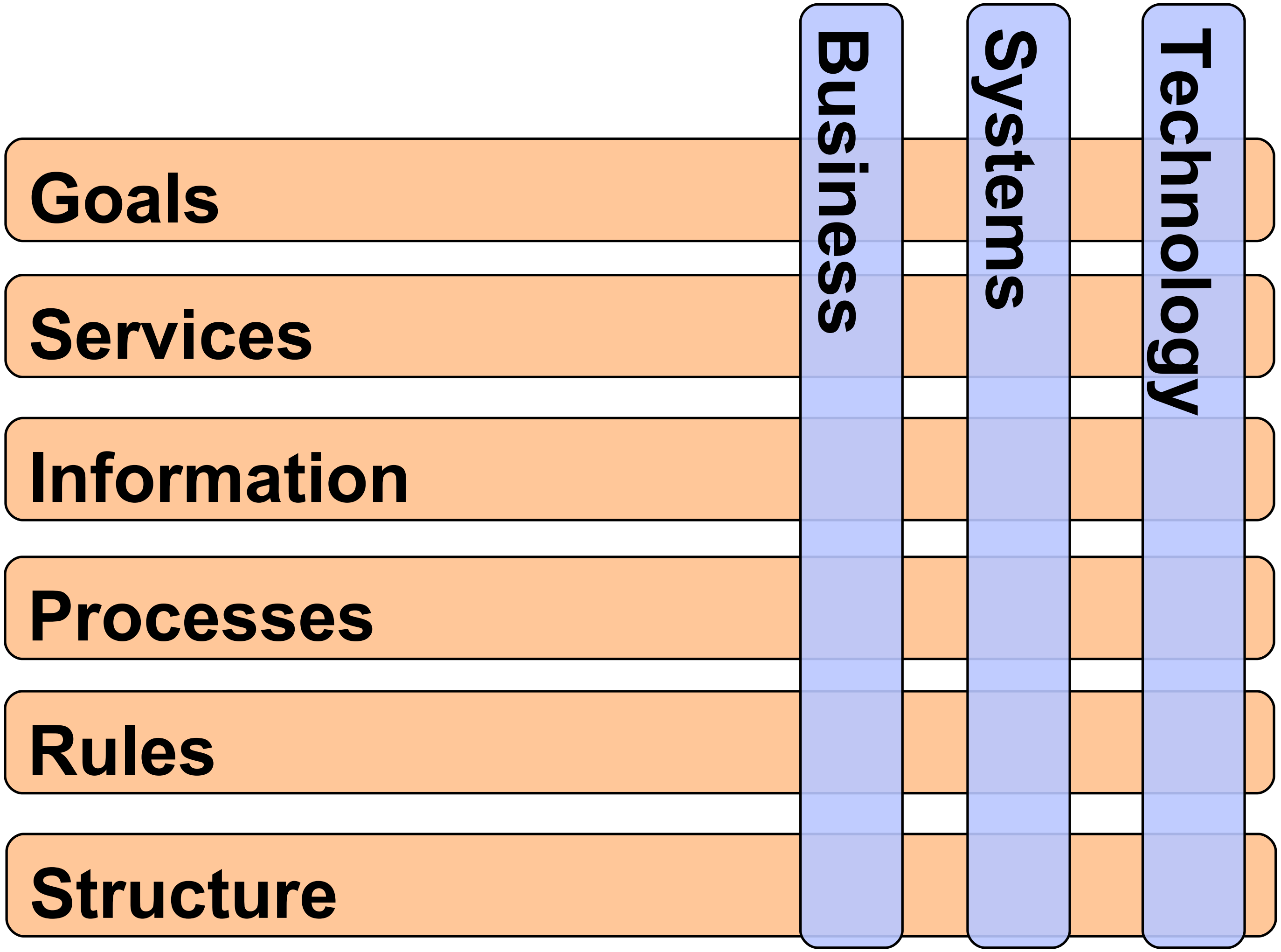
- Today, modeling, architecture, vocabularies and enterprise information are closed and siloed
- There is an opportunity
  - To help federate information for and about the enterprise and enterprise systems
  - To enable architecture as an open and collaborative experience, tuned to the needs of stakeholders
  - To discover and reconcile concepts, entities and architectures throughout the enterprise and beyond.
  - To unify the knowledge in multiple tools, infrastructures and information resources

  - To enable the transformations, agility, efficiency, collaboration and automation we have been promising for years

Case Study

- Business and systems financial architecture for a government agency
- Understand the business needs in terms of business processes, information and business services
- Specify the data, technology processes and SOA services of the systems to meet business needs

Architectural viewpoints



Business

Systems

Technology

Goals

Services

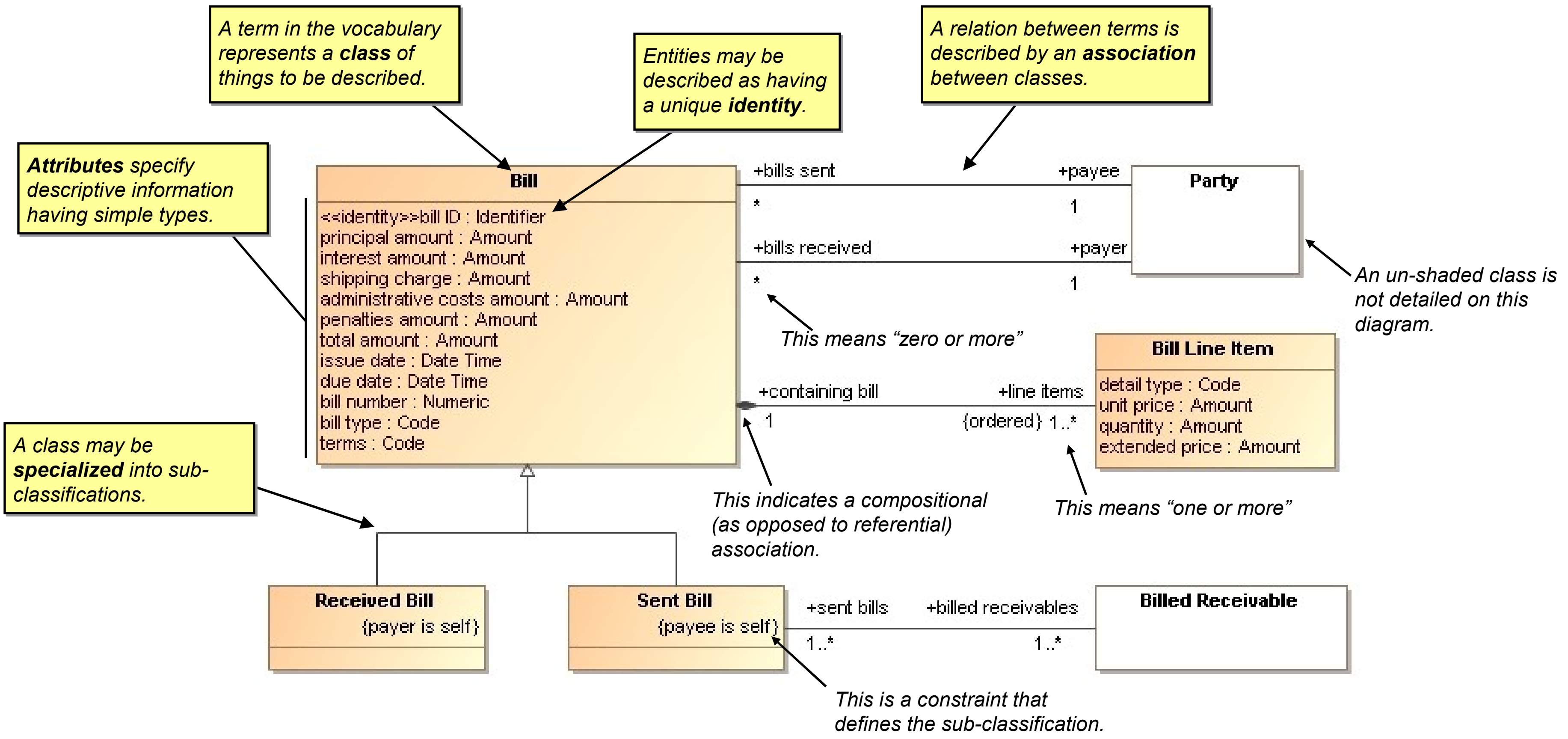
Information

Processes

Rules

Structure

Example Information Viewpoint



A term in the vocabulary represents a ***class*** of things to be described.

Entities may be described as having a unique ***identity***.

A relation between terms is described by an ***association*** between classes.

Attributes specify descriptive information having simple types.

An un-shaded class is not detailed on this diagram.

This means "zero or more"

This indicates a compositional (as opposed to referential) association.

This means "one or more"

A class may be ***specialized*** into sub-classifications.

This is a constraint that defines the sub-classification.

Information viewpoint as Ontology

- The information viewpoint corresponds most closely to what is typically done in an ontology – it establishes the concepts and vocabulary of the domain
- While not quite as powerful as ontological languages, UML can be used ontologically to describe the domain
- Information from UML can be used to populate the entities, relations and constraints of an ontology – or visa-versa
- We take an ontological approach to the high level model – the intent is to capture the concepts of the domain without technology concerns

Other Viewpoints as Ontologies

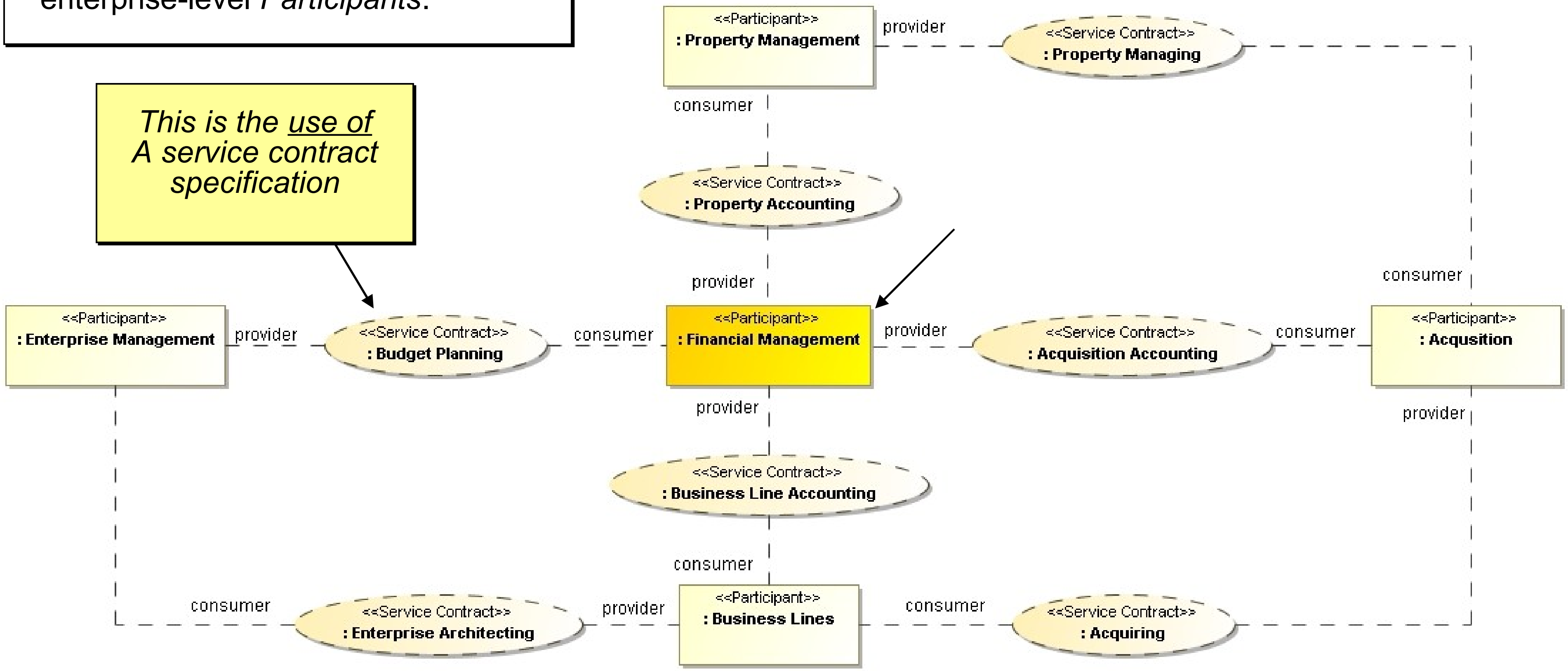
- Requirements, processes & services are less often captured as ontologies
- Yet the ontology of a domain must include these viewpoints
- Better support for other viewpoints with architecturally focused ontologies would provide increased value
- Links between architectural an ontological tools provides a bridge between these related approaches

Financial Management Enterprise Context – Value Chain Services

- The service-oriented business architecture of an enterprise is modeled as a *Collaboration* of enterprise-level *Participants*.

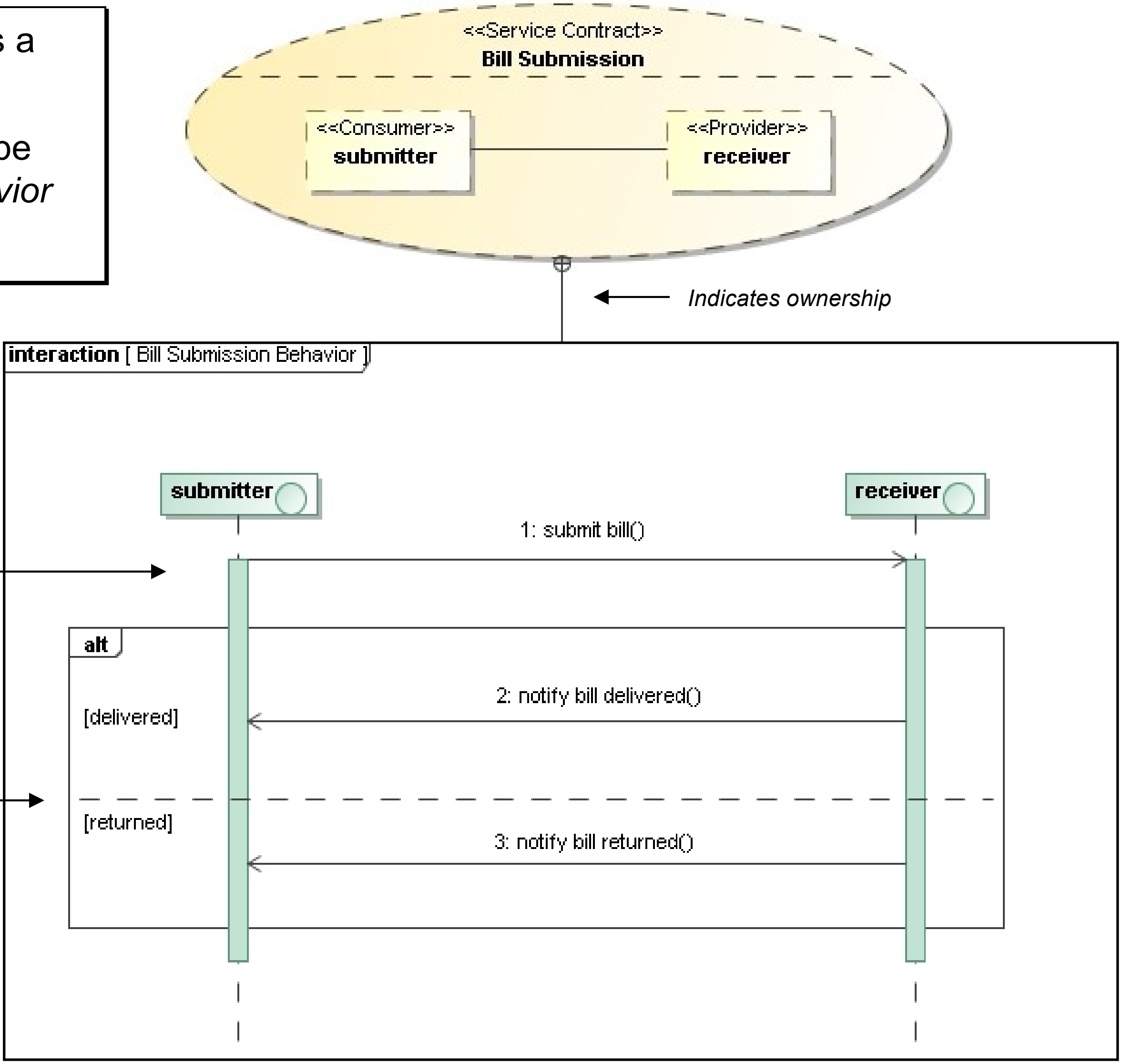
External enterprise level participants

This is the use of ***A service contract specification***



Simple Bill Submission Service Contract

- A service contract is modeled as a *UML Collaboration*.
- The required conversation may be specified using an *Owned Behavior* (e.g., Interaction or Activity)



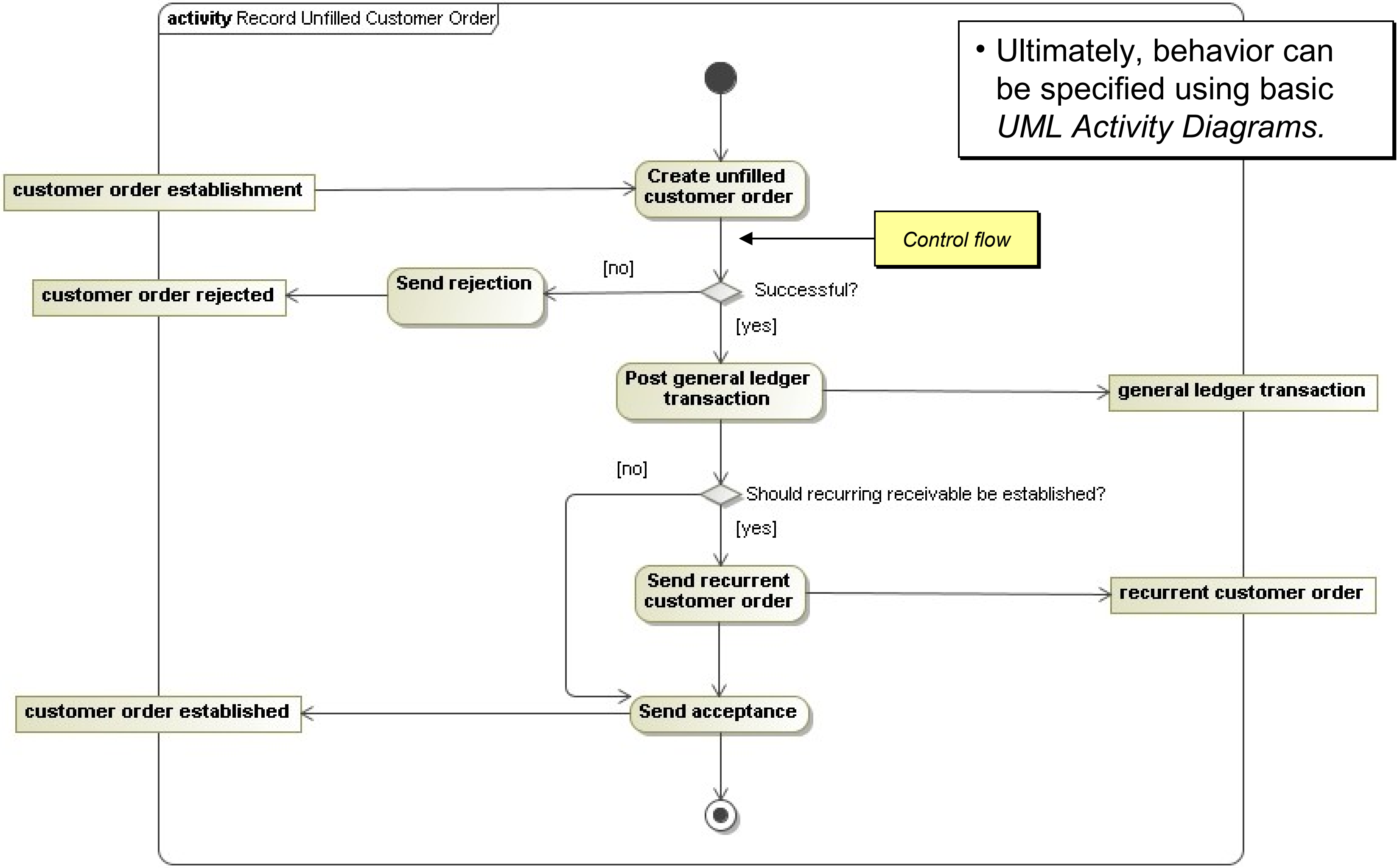
Indicates ownership

Note that, while one Participant requests the service and the other responds, information may flow both ways during the interaction.

First the submitter submits a bill to the receiver…

…then either *the bill is successfully delivered or it is returned.*

Process Viewpoint

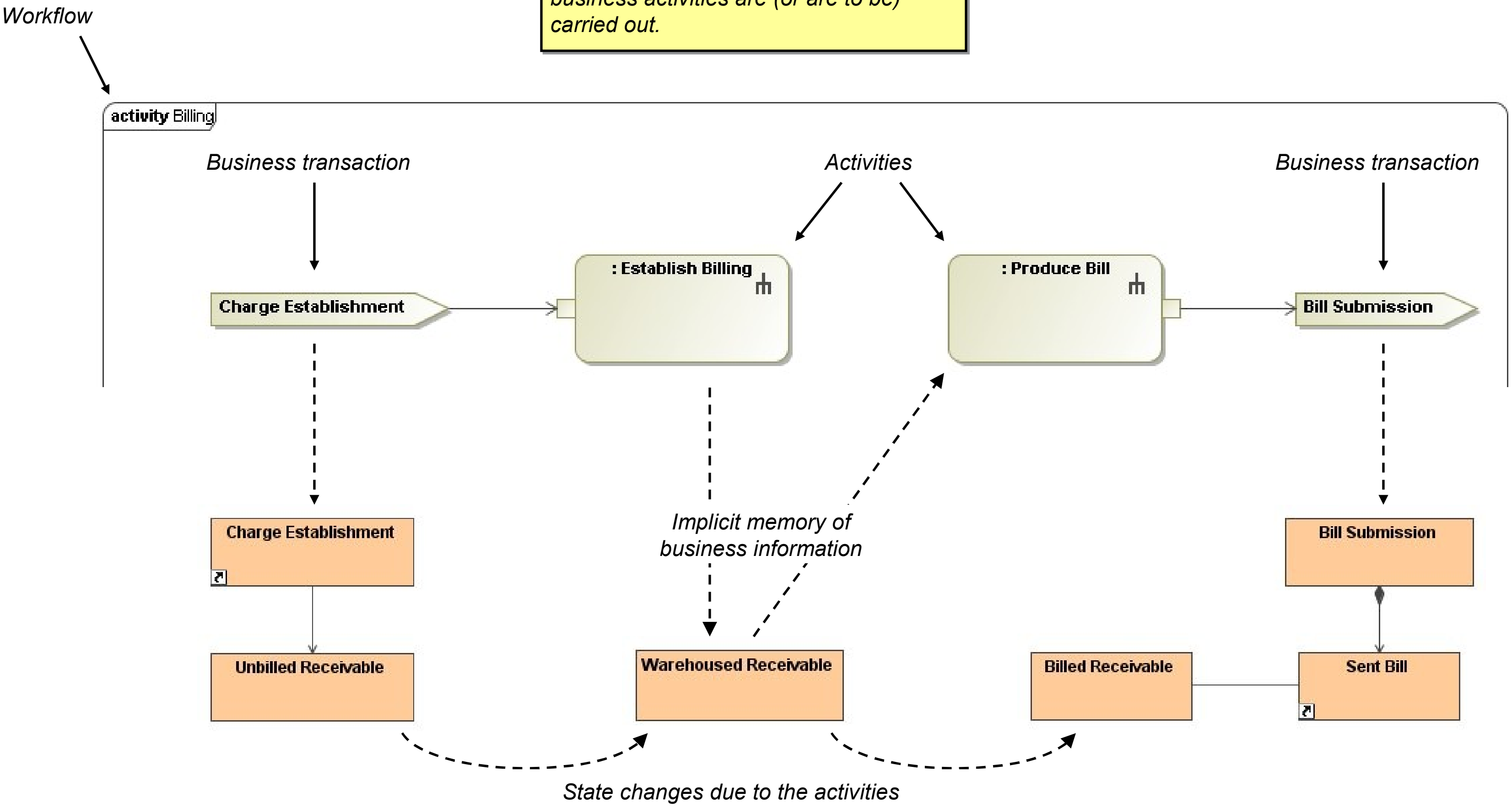


activity Record Unfilled Customer Order
customer order establishment
Create unfilled customer order
Control flow
[no]
Successful?
Send rejection
customer order rejected
[yes]
Post general ledger transaction
general ledger transaction
[no]
Should recurring receivable be established?
[yes]
Send recurrent customer order
recurrent customer order
Send acceptance
customer order established

- Ultimately, behavior can be specified using basic *UML Activity Diagrams.*

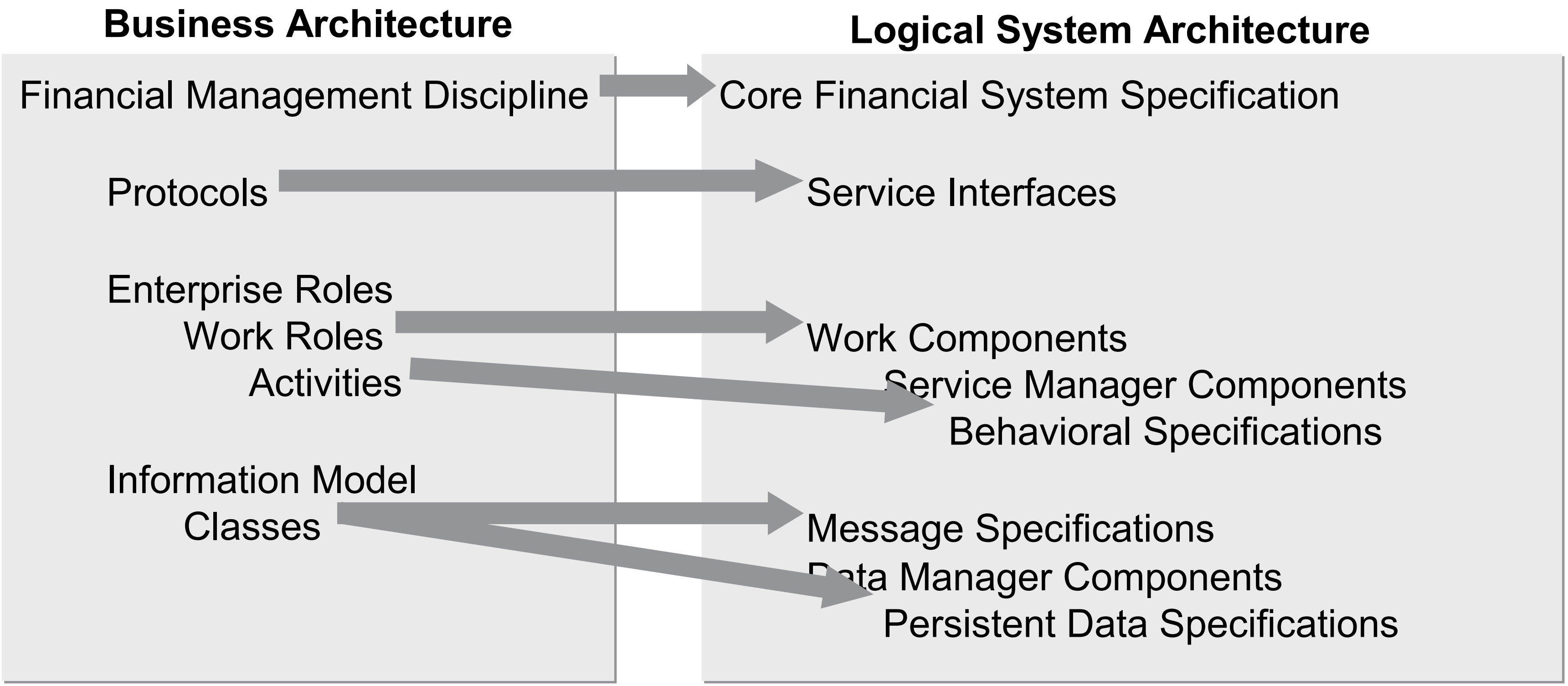
Information connects processes

*The **process model** describes how business activities are (or are to be) carried out.*



*The **information model** details the vocabulary of the business entities and transactions used in the process model.*

From Business Architecture to System Architecture



Business Architecture

Financial Management Discipline

Protocols

Enterprise Roles
Work Roles
Activities

Information Model
Classes

Logical System Architecture

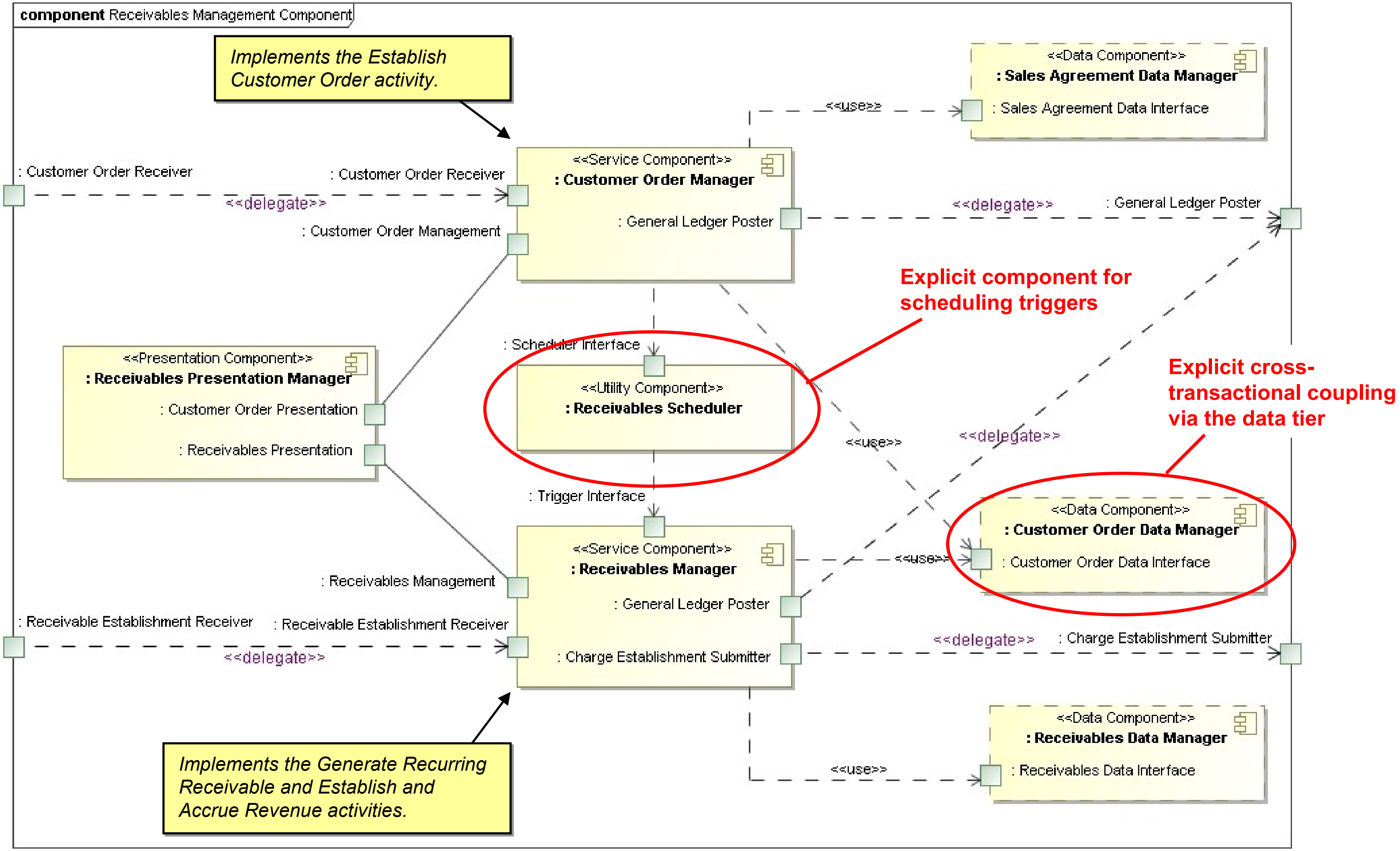
Core Financial System Specification

Service Interfaces

Work Components
Service Manager Components
Behavioral Specifications

Message Specifications
Data Manager Components
Persistent Data Specifications

Receivables Management Component Architecture


component Receivables Management Component
<<Data Component>>
: Sales Agreement Data Manager
: Sales Agreement Data Interface
<<use>>
<<Service Component>>
: Customer Order Manager
: Customer Order Receiver
: Customer Order Receiver
<<delegate>>
: Customer Order Management
: General Ledger Poster
<<delegate>>
: General Ledger Poster
<<Presentation Component>>
: Receivables Presentation Manager
: Customer Order Presentation
: Receivables Presentation
: Scheduler Interface
<<Utility Component>>
: Receivables Scheduler
<<use>>
<<delegate>>
: Trigger Interface
<<Data Component>>
: Customer Order Data Manager
: Customer Order Data Interface
<<use>>
<<Service Component>>
: Receivables Manager
: Receivables Management
: General Ledger Poster
: Receivable Establishment Receiver
: Receivable Establishment Receiver
<<delegate>>
: Charge Establishment Submitter
<<delegate>>
: Charge Establishment Submitter
<<Data Component>>
: Receivables Data Manager
: Receivables Data Interface
<<use>>

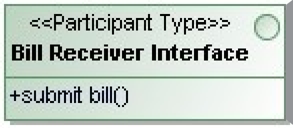
Implements the Establish Customer Order activity.

Explicit component for scheduling triggers

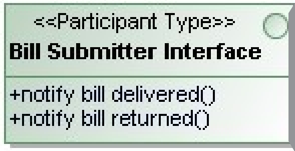
Explicit cross-transactional coupling via the data tier

Implements the Generate Recurring Receivable and Establish and Accrue Revenue activities.

Example Web Services Generation

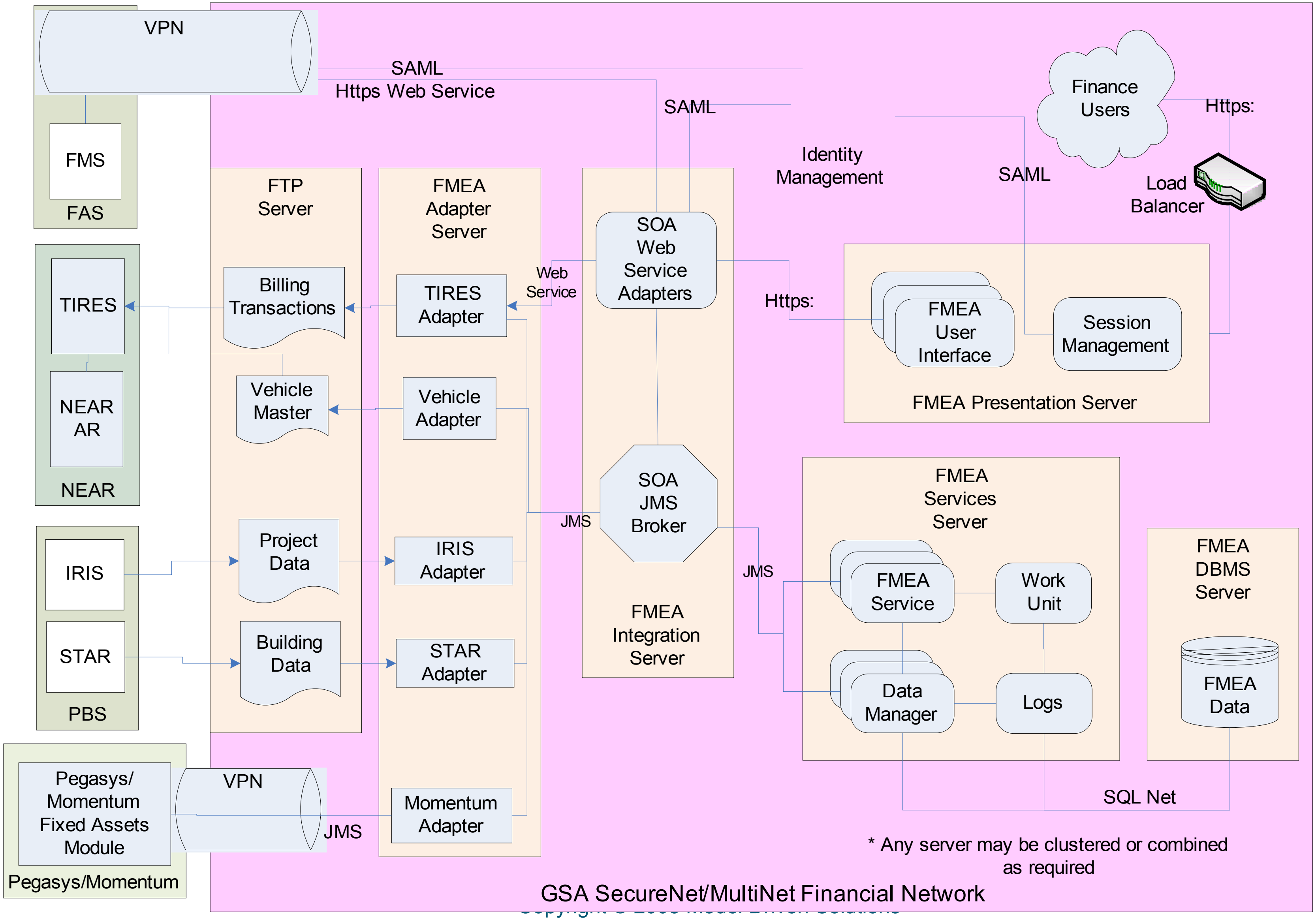


```
<wsdl:portType name=“BillSubmission.BillSubmissionReceiverInterface">
   <wsdl:operation name=“submitBill">
     <wsdl:input message="tns:BillSubmissionCluster“
                 name=“billSubmission">
     </wsdl:input>
   </wsdl:operation>
 </wsdl:portType>
```



```
<wsdl:portType name=“BillSubmission.BillSubmissionSubmitterInterface">
  <wsdl:operation name=“notifyBillDelivered">
    <wsdl:input message="tns:BillDeliveredCluster“
                name=“billDelivered">
    </wsdl:input>
  </wsdl:operation>
  <wsdl:operation name=“notifyBillReturned">
    <wsdl:input message="tns:BillReturnedCluster“
                name=“billReturned">
    </wsdl:input>
  </wsdl:operation>
</wsdl:portType>
```

On this infrastructure



* Any server may be clustered or combined as required

Summary

- Architectures and ontologies are mutually supportive
- Ontological precision and the ability to federate ontologies brings value to architecture
- Architectural tools can provide a more friendly way to express ontological information to stakeholders
- Automating parts of systems from models and ontologies using MDA (model driven architecture) provides the much of the value without runtime overhead
- The strategic opportunity is to bring all of this information into focus for the enterprise – we are only starting to do so.