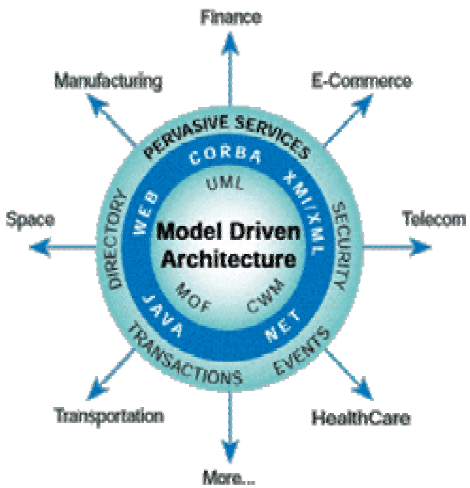
Implementing SOA/ESB using Enterprise MDA



VCA
MDA
SOA/ESB

EDOC

UNIFIED MODELING LANGUAGE™

Introductions

DataAccessTechnologies

Where Business Meets Technology

Cory Casanave

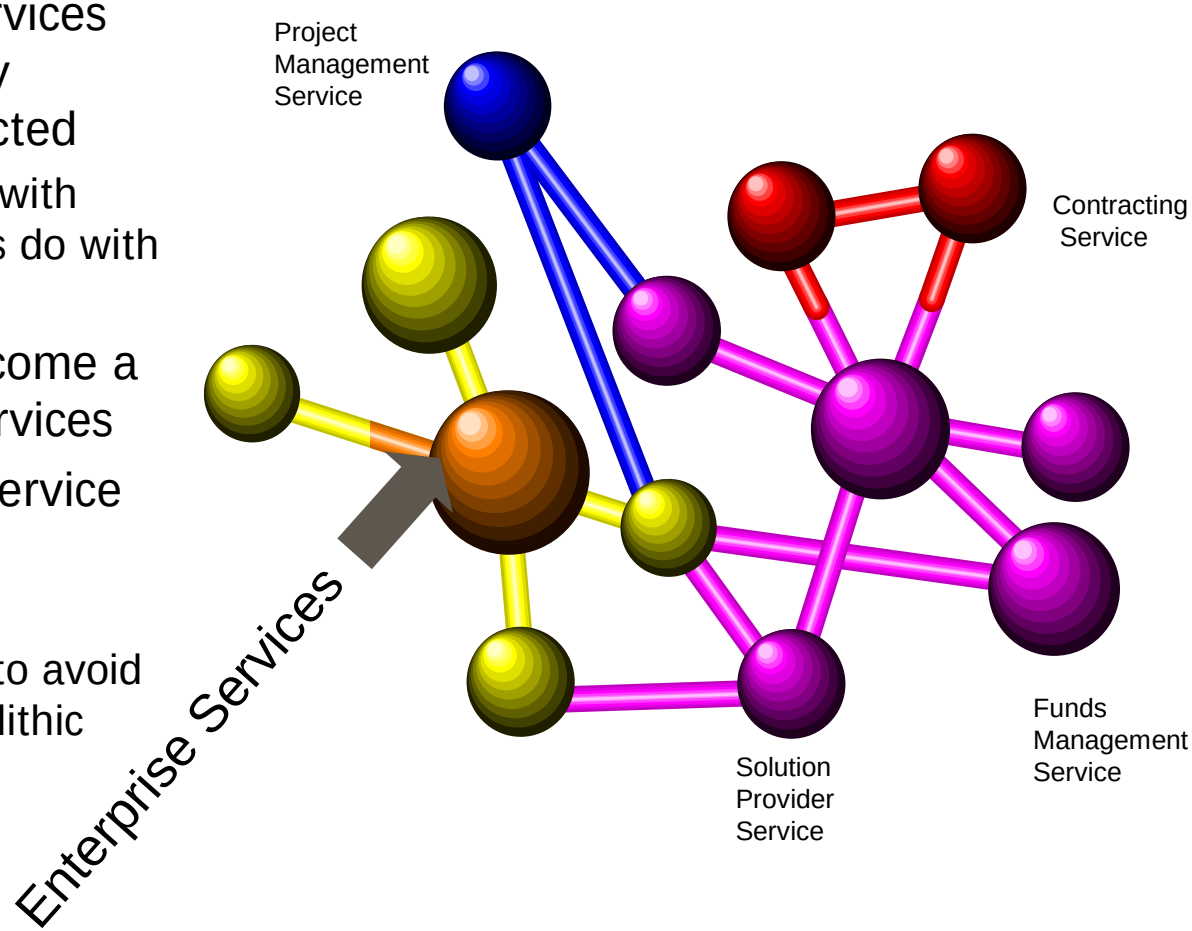
cory-c@enterprisecomponent.com

Primary author of “CCA” in EDOC

Enterprise Service Bus to Enable Target State

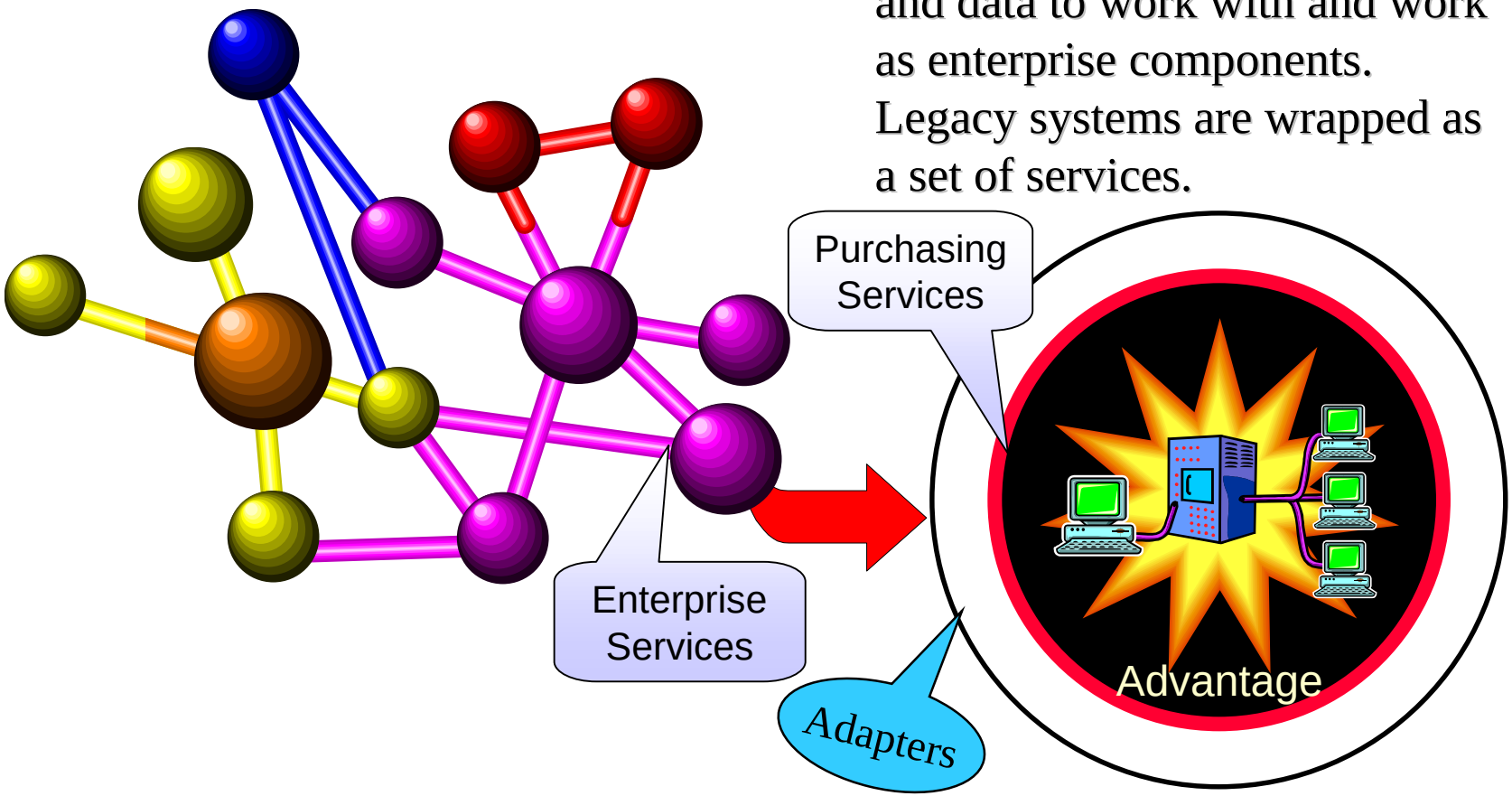
One-GSA Business Model

- Services driven from the business model
- Reusable Enterprise Services are independent & easily adapted and interconnected
  - Services communicate with each other like humans do with email
- Information systems become a lattice of cooperating services
- Provide an “Enterprise Service Bus” using commercial standards
  - Industry best practice to avoid developing large monolithic applications

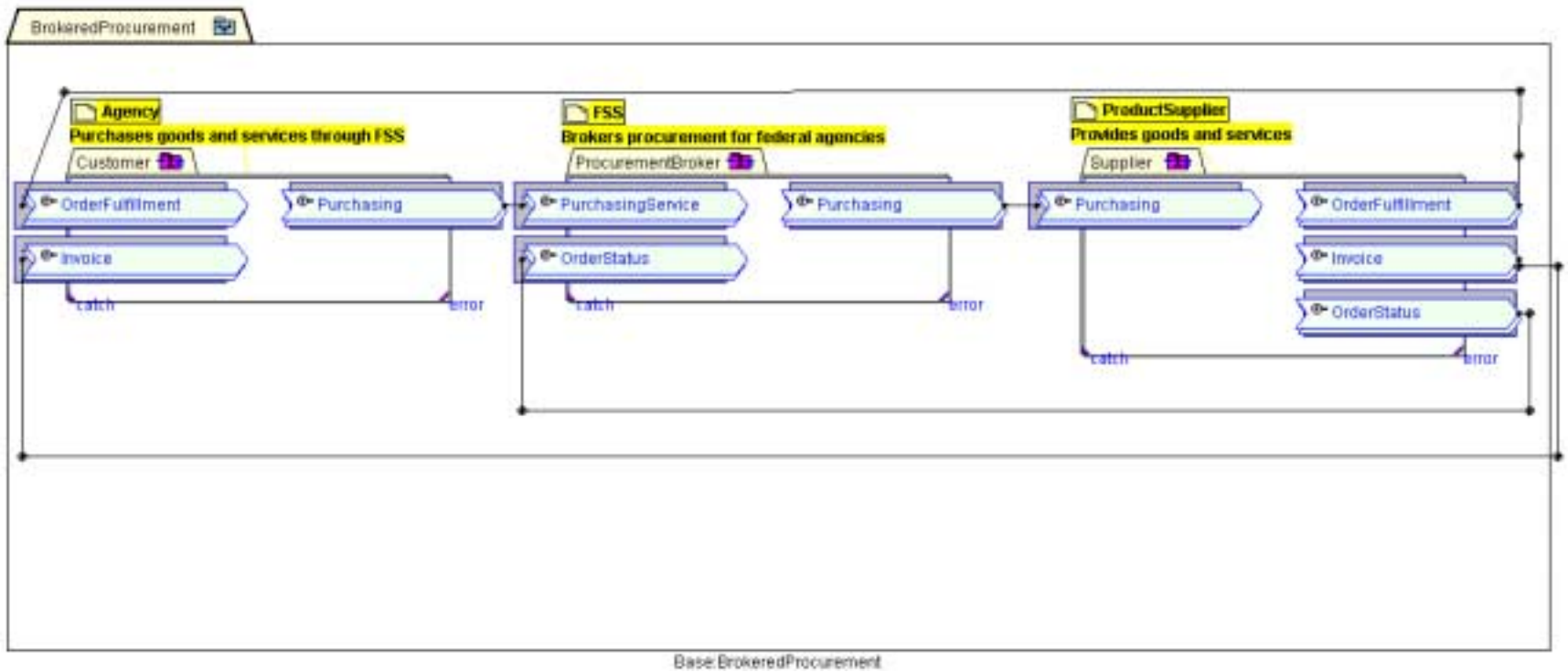


Legacy “Wrapping”

Wrapping allows existing programs and data to work with and work as enterprise components. Legacy systems are wrapped as a set of services.

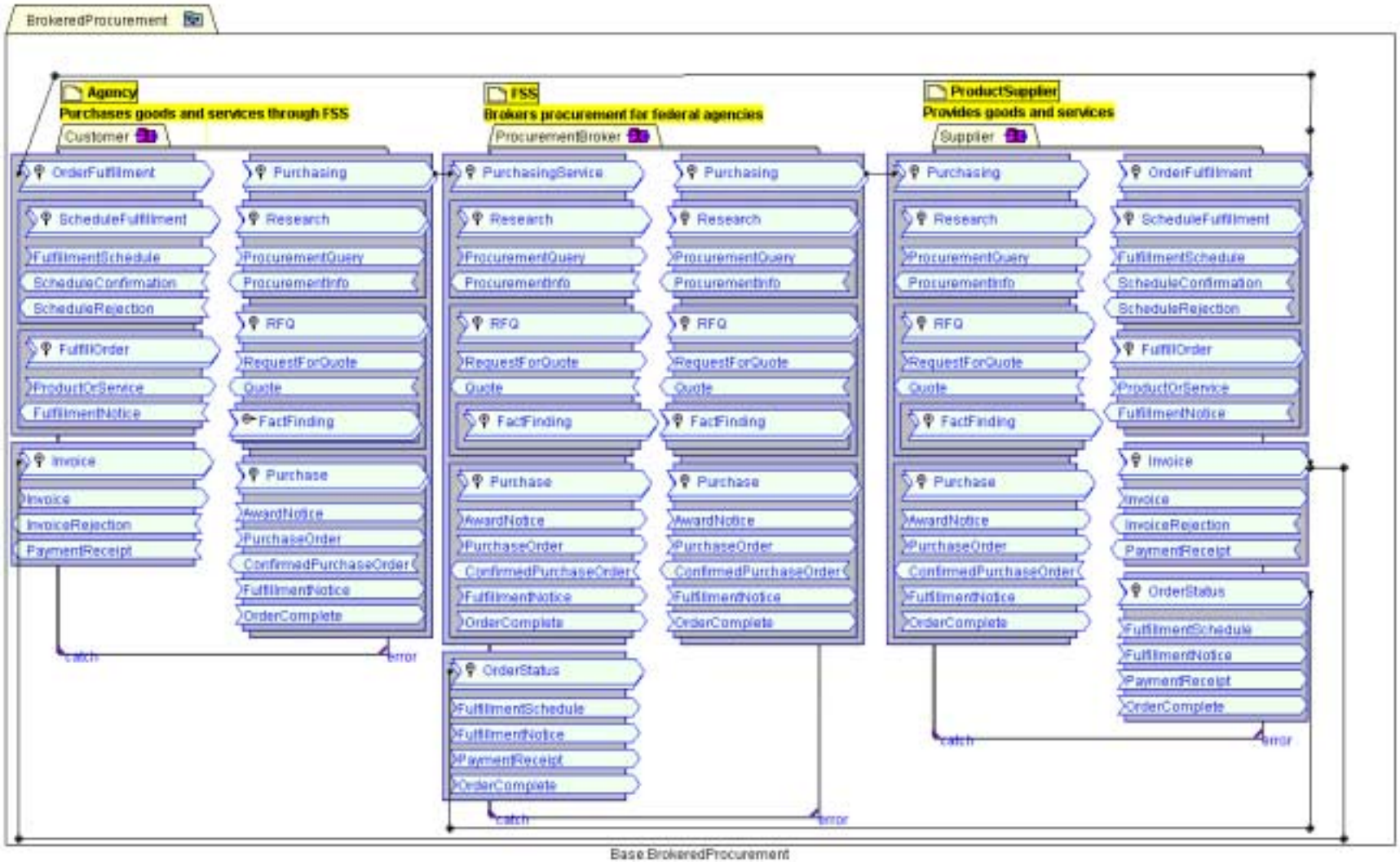


Summary Top Level Collaboration

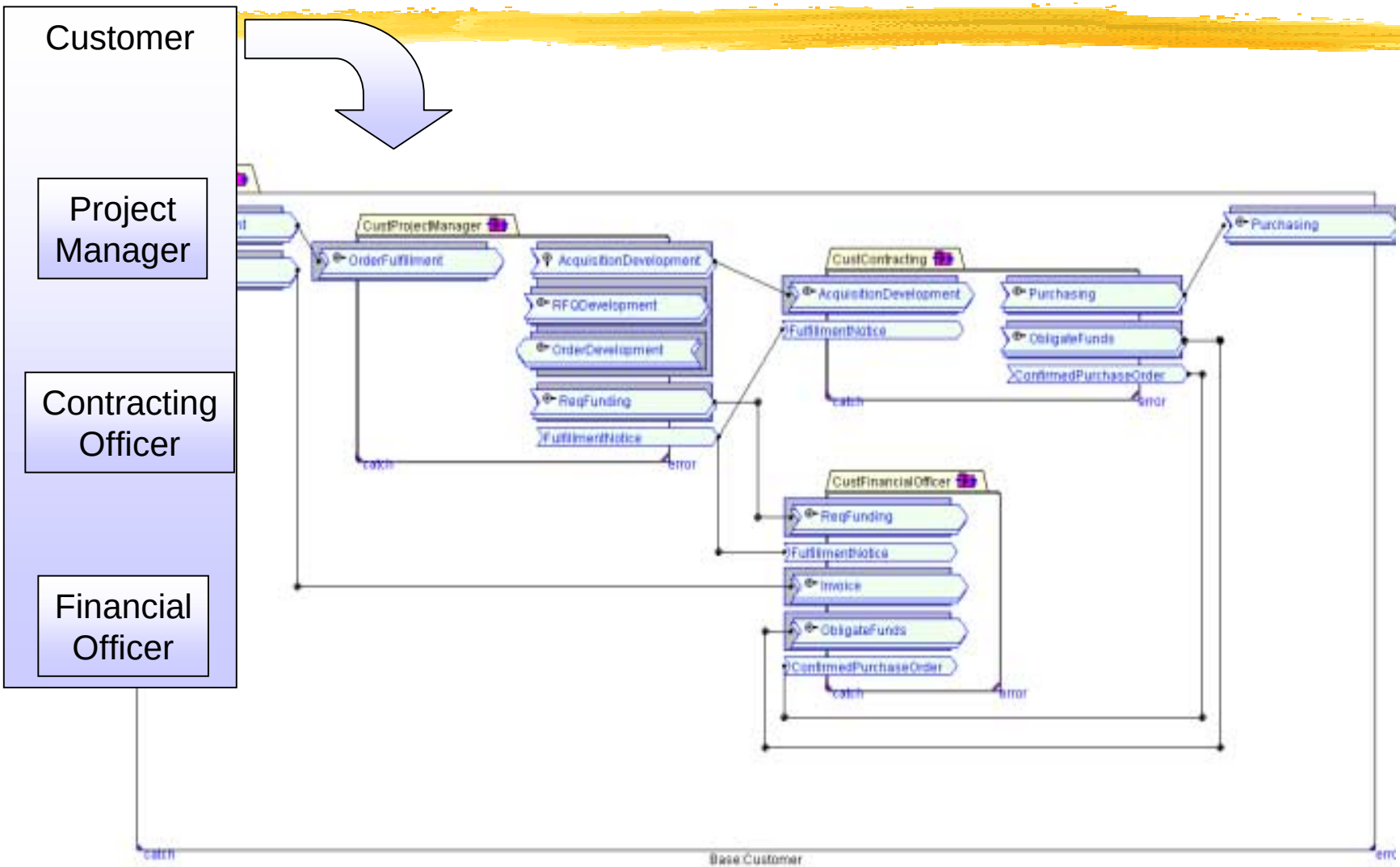


Simplified View - Level of detail is optional

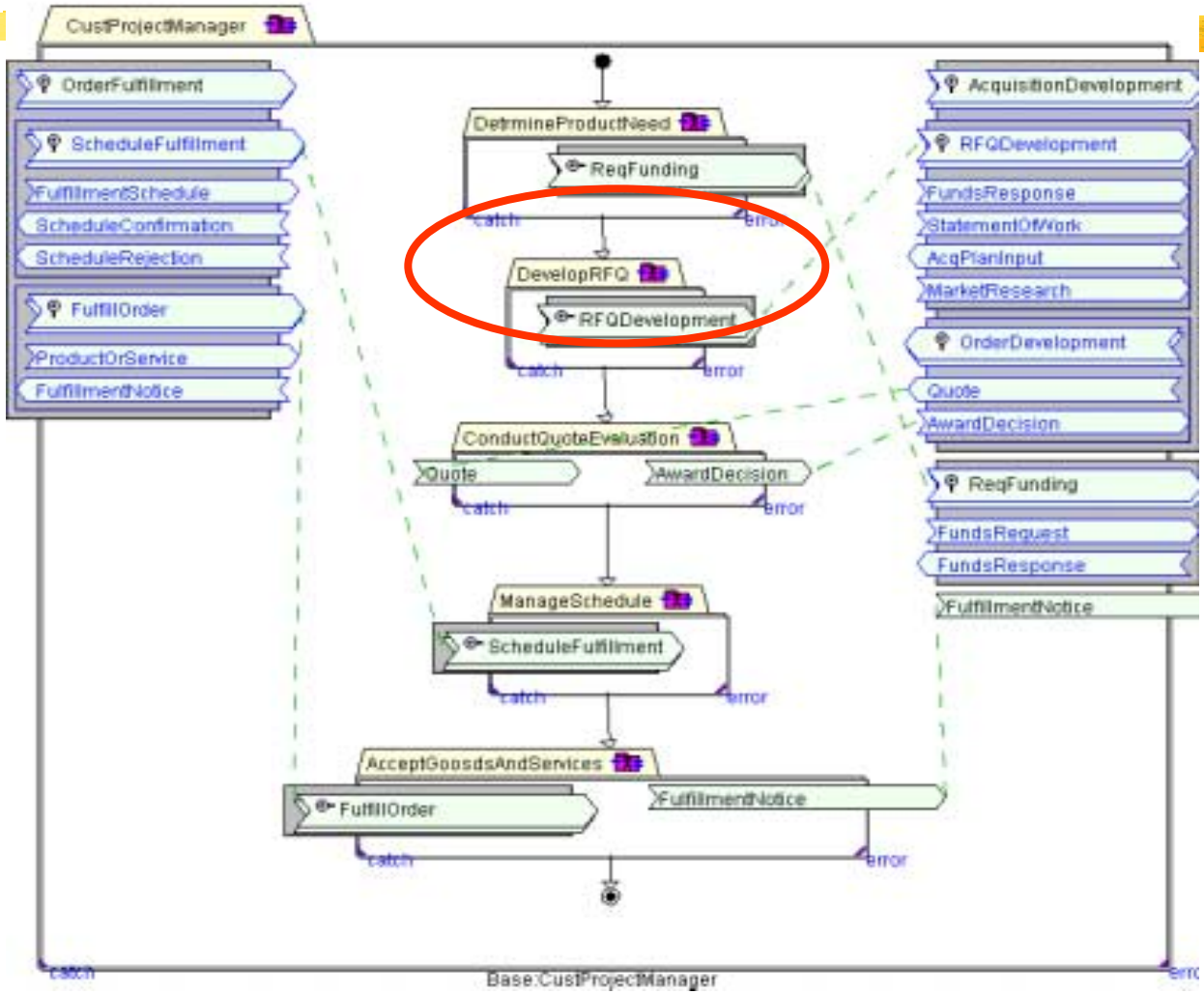
Full Services Collaboration



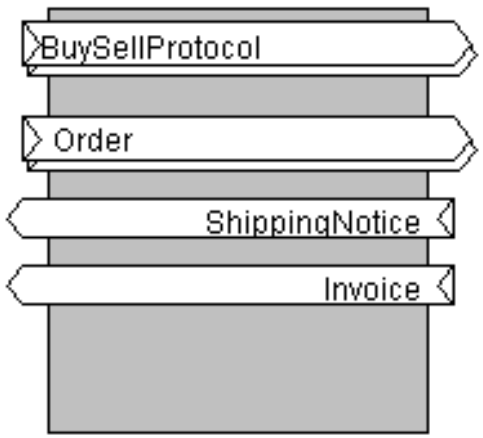
Drilling Down into Customer Detail

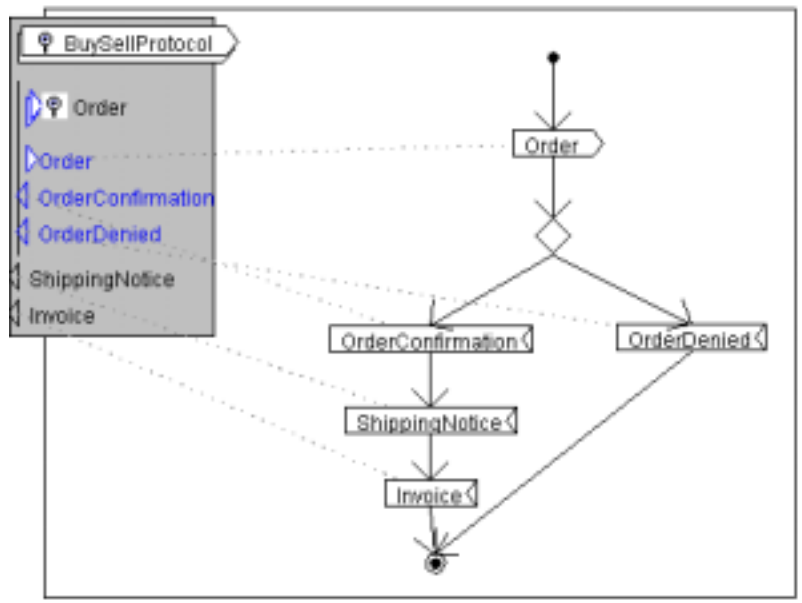


Choreography of Process



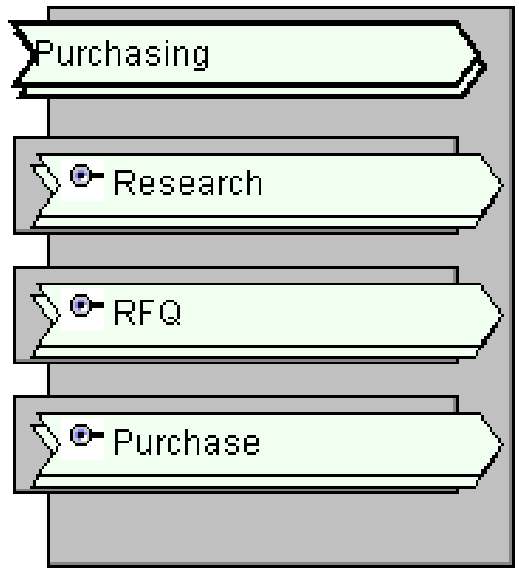
Organize into protocols





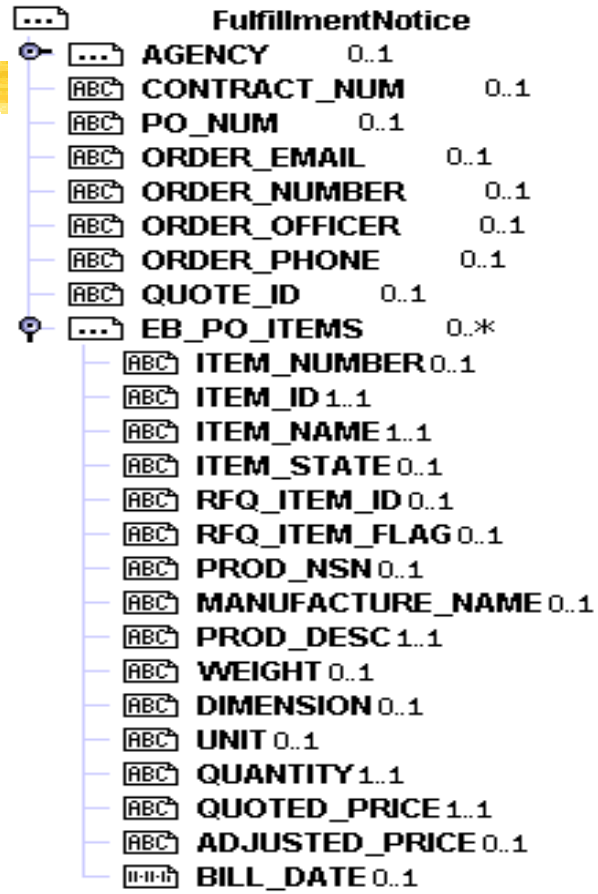
Inner Protocols

- Protocols represent conversations between roles
- Conversations frequently have sub-conversations, detail about a specific subject
- These sub-conversations are inner protocols
- Inner protocols can also be reused in other protocols or even as top-level protocols
- Protocols can “nest” to any level of detail

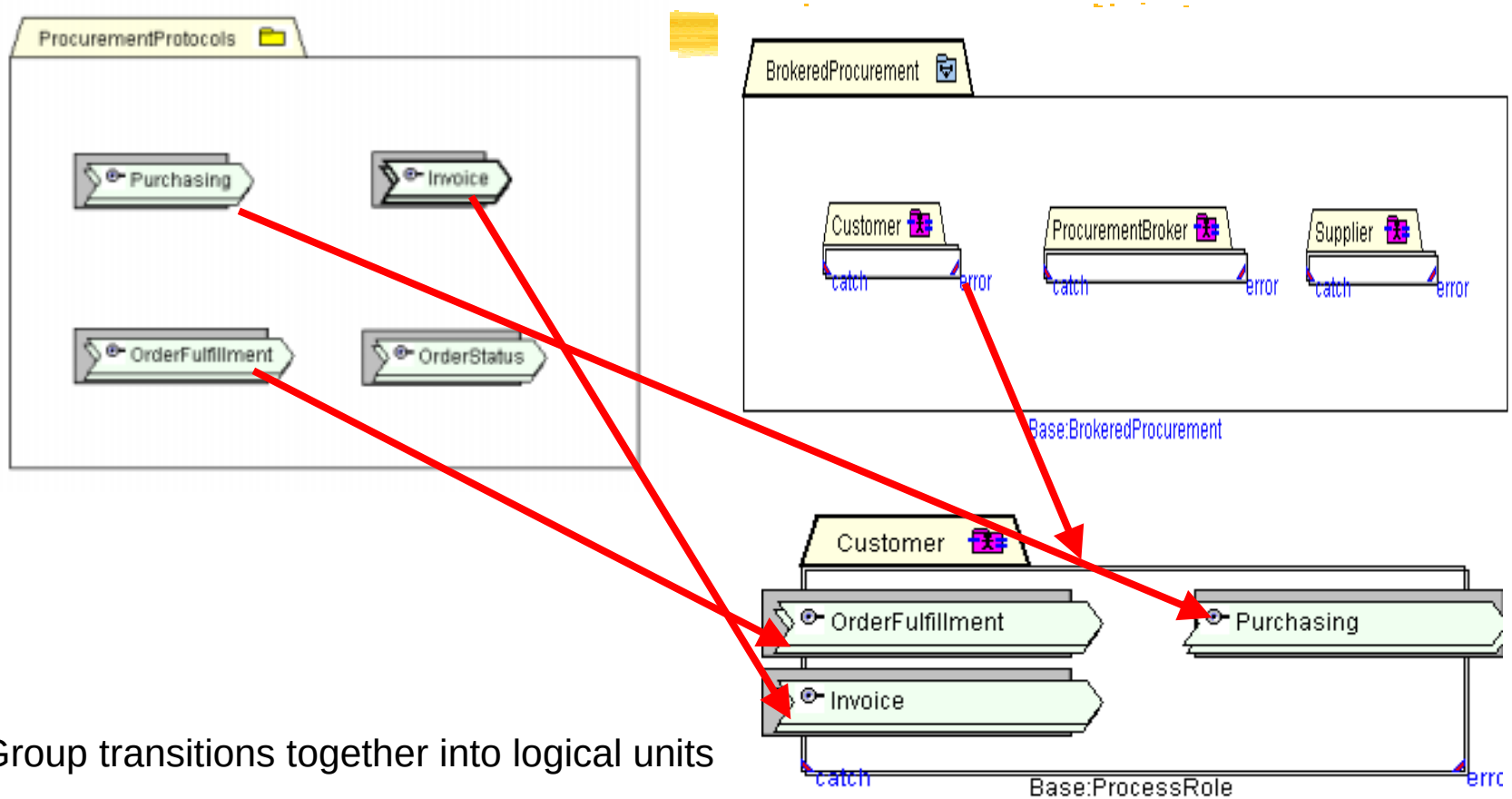


Modeling Collaboration Documents

- Fill in details of the documents
- Focus on business information - not technology
- Collaboration - Not an information model
- May be derived from existing sources
- Helps in creating technology mappings - E.G. Web Services
- Includes
  - Composition
  - Type
  - Cardinality



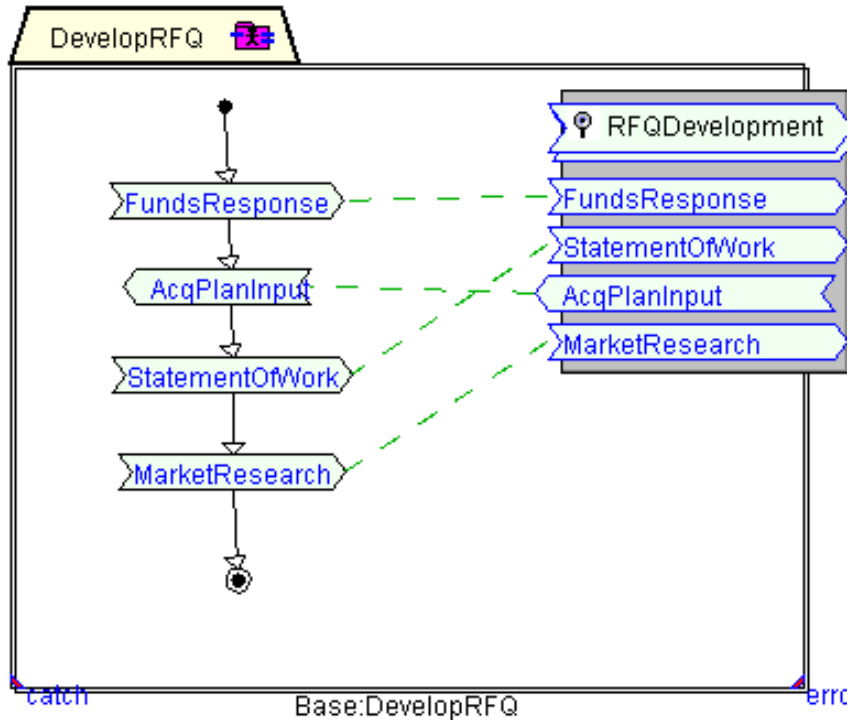
Attach Protocols to Roles as "Ports"



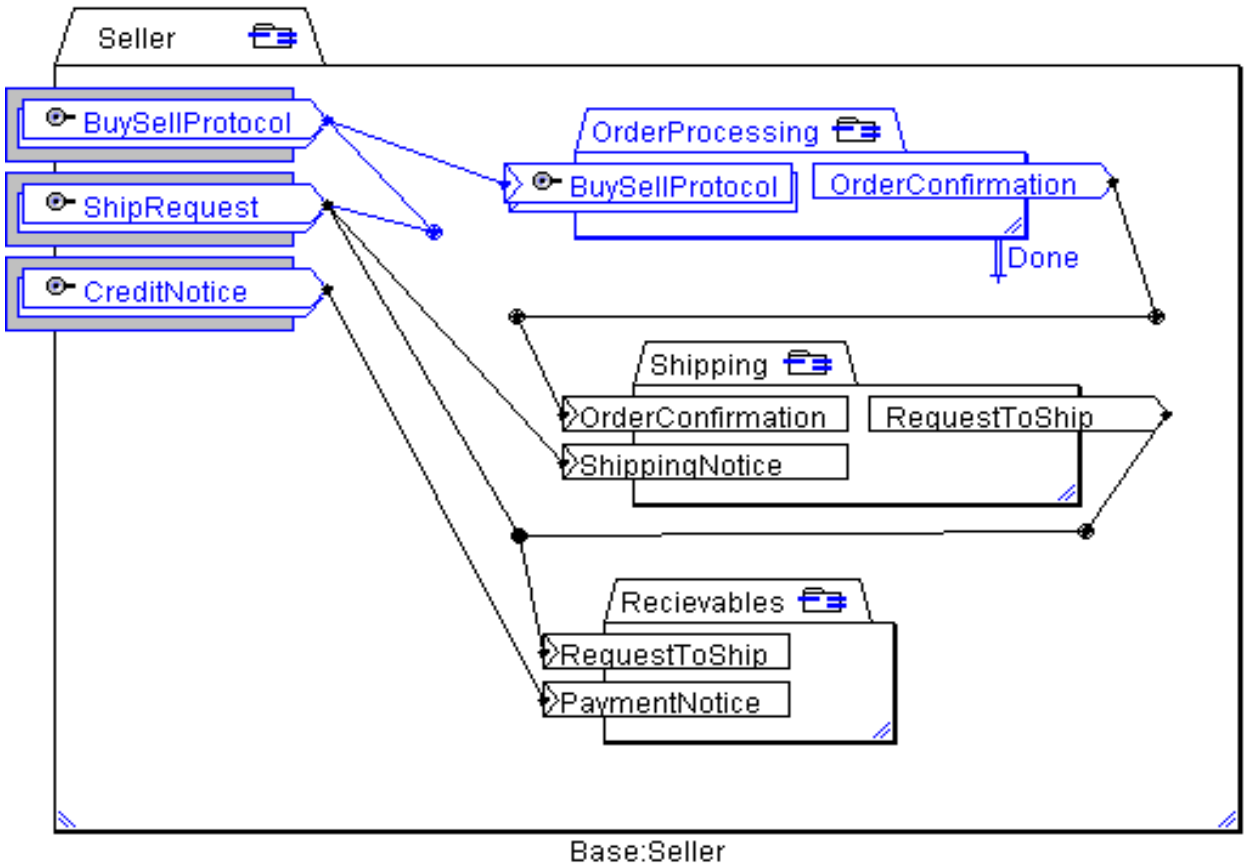
Group transitions together into logical units

Detailed Information Flows

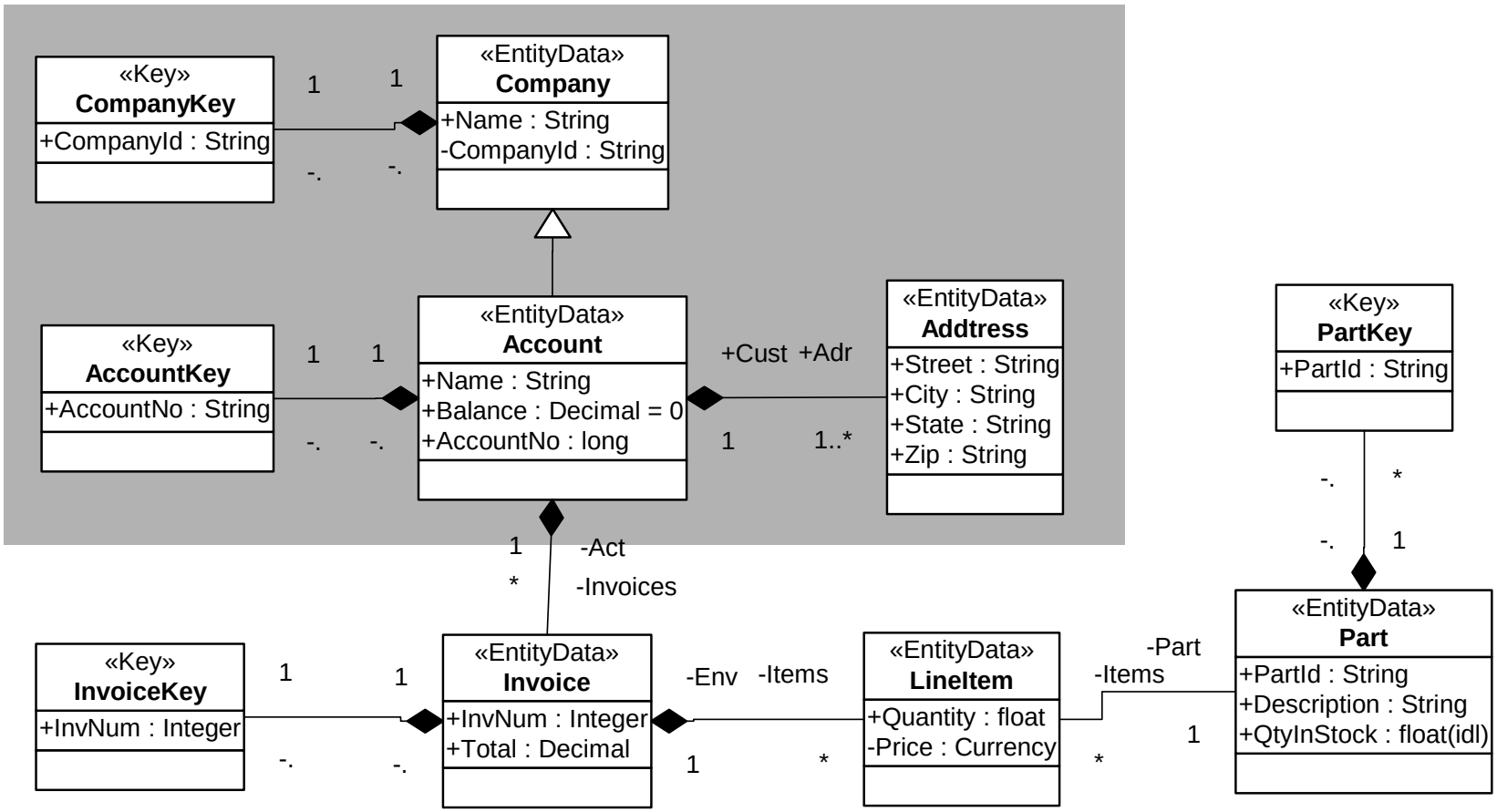
- Inside the activities we can drill down to inner activities or detailed document flows - sending and receiving information
- This is used for the simulation, to validate the the model is correct and ultimately to test the implemented components.



Drill-down

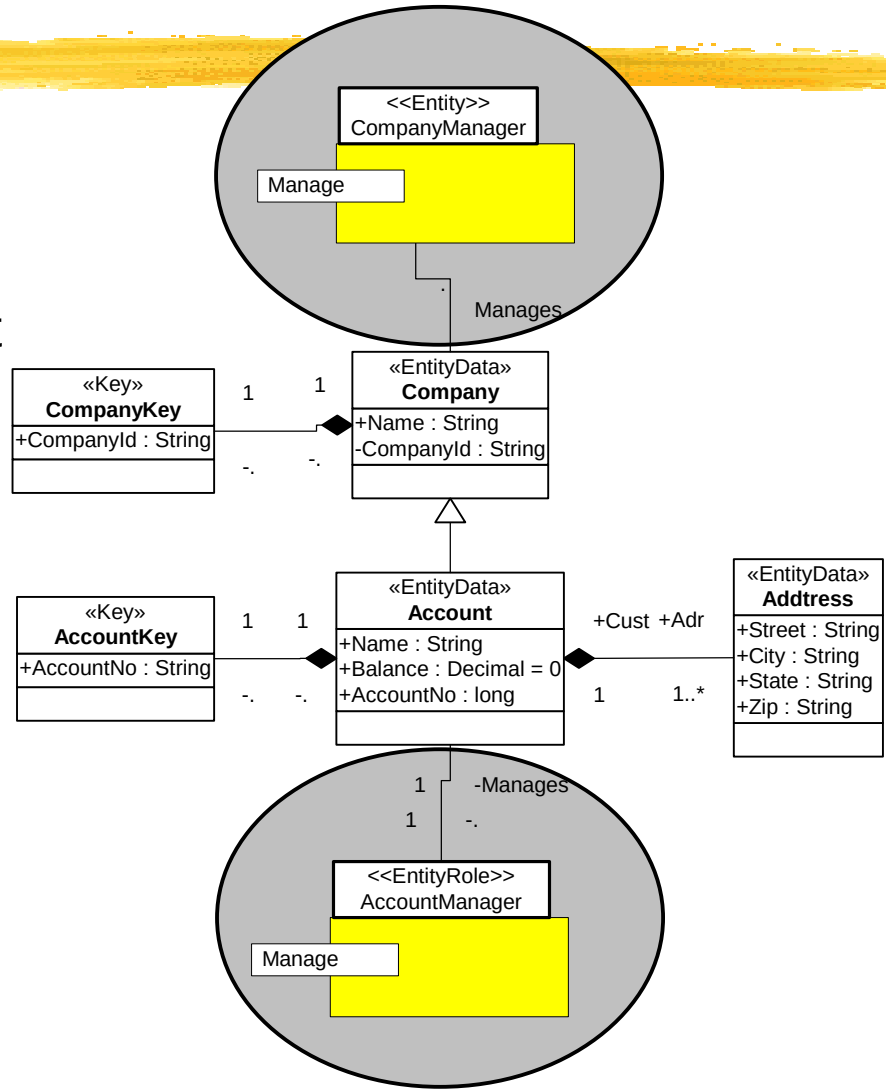


Sample Information Model

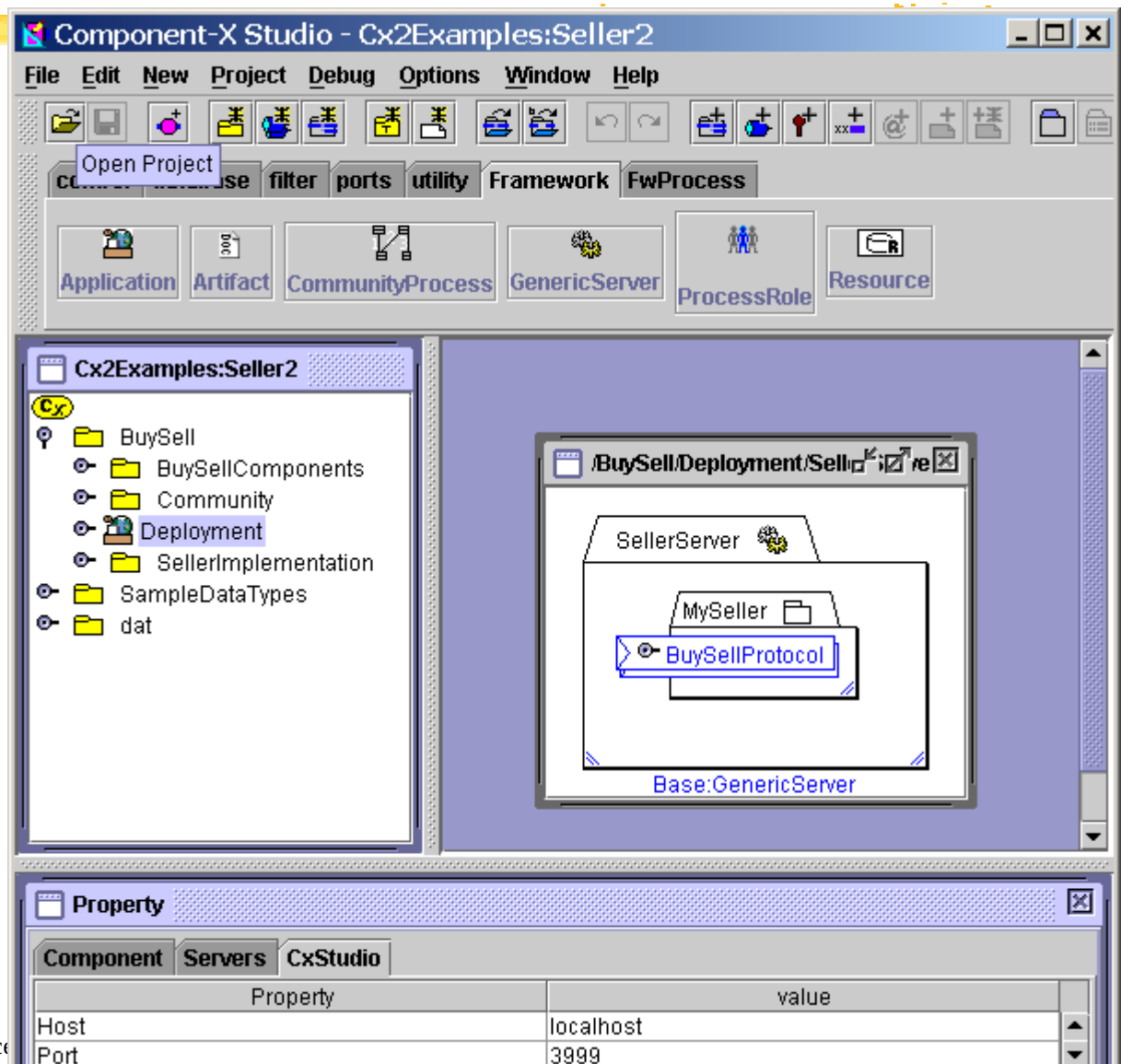


Adding Entities

- Entities are added to manage entity data
- Entity Roles are managers that provides a view of the same identity in another context
- The Entities have ports for managing and accessing the entities
- Non-entities which are owned by (aggregate into) an entity are managed by the entity



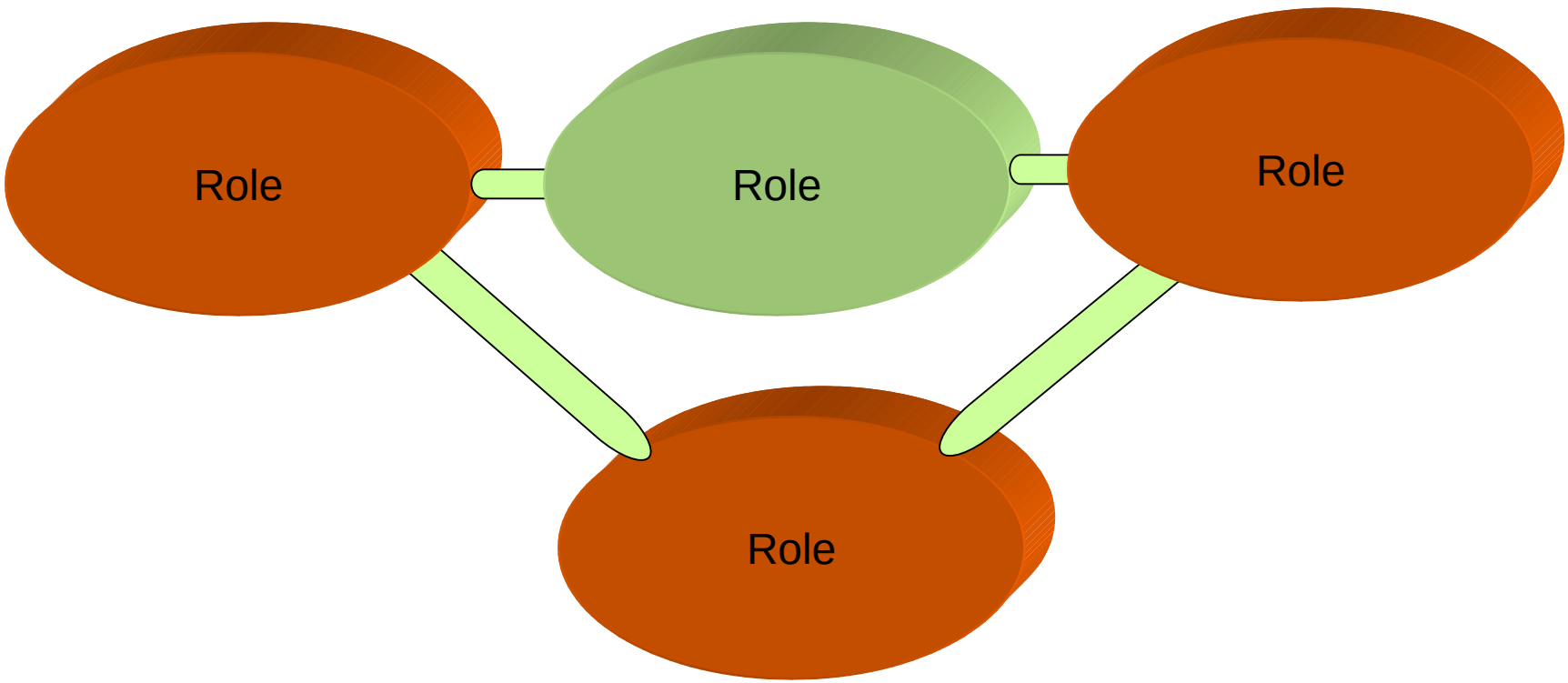
Add technology specifics for deployment



Realizing a Business Architecture

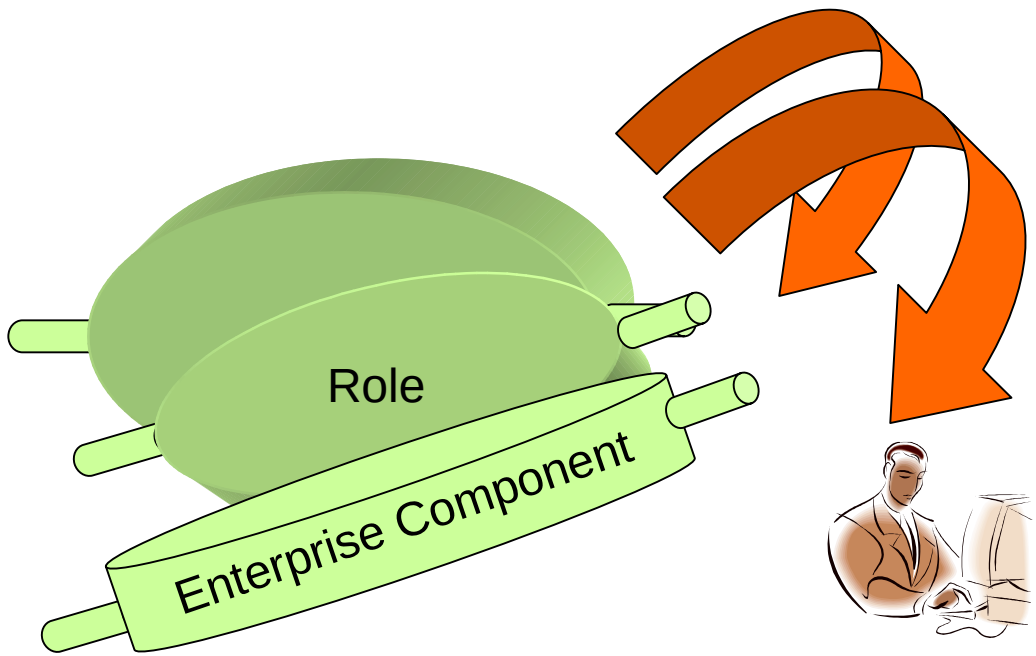
Roles to Enterprise Components & Services

Business (CIM) view - Collaborating Roles



“Upper” PIM View - Enterprise Component

“Rotate” to look
At other aspects
of the component



People, organizations
And/or enterprise components
play roles in Business
Processes.

The "Enterprise Digital Assistant"

Business Process

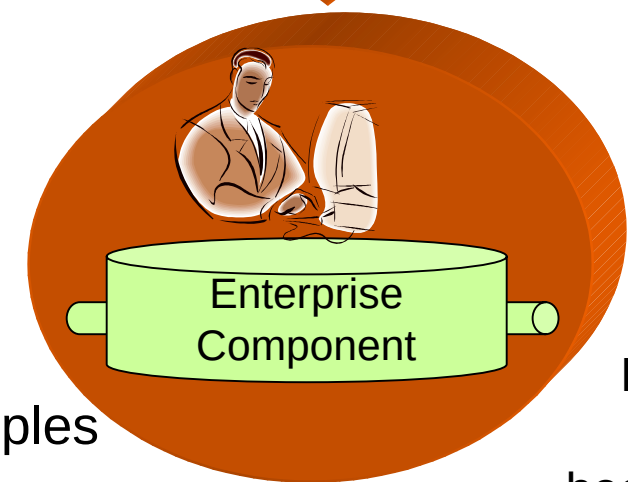
People, Organizations
And systems play roles

Role

Components frequently
help people play these roles

Enterprise components help people
and organizations play roles
by automating and monitoring
The business process

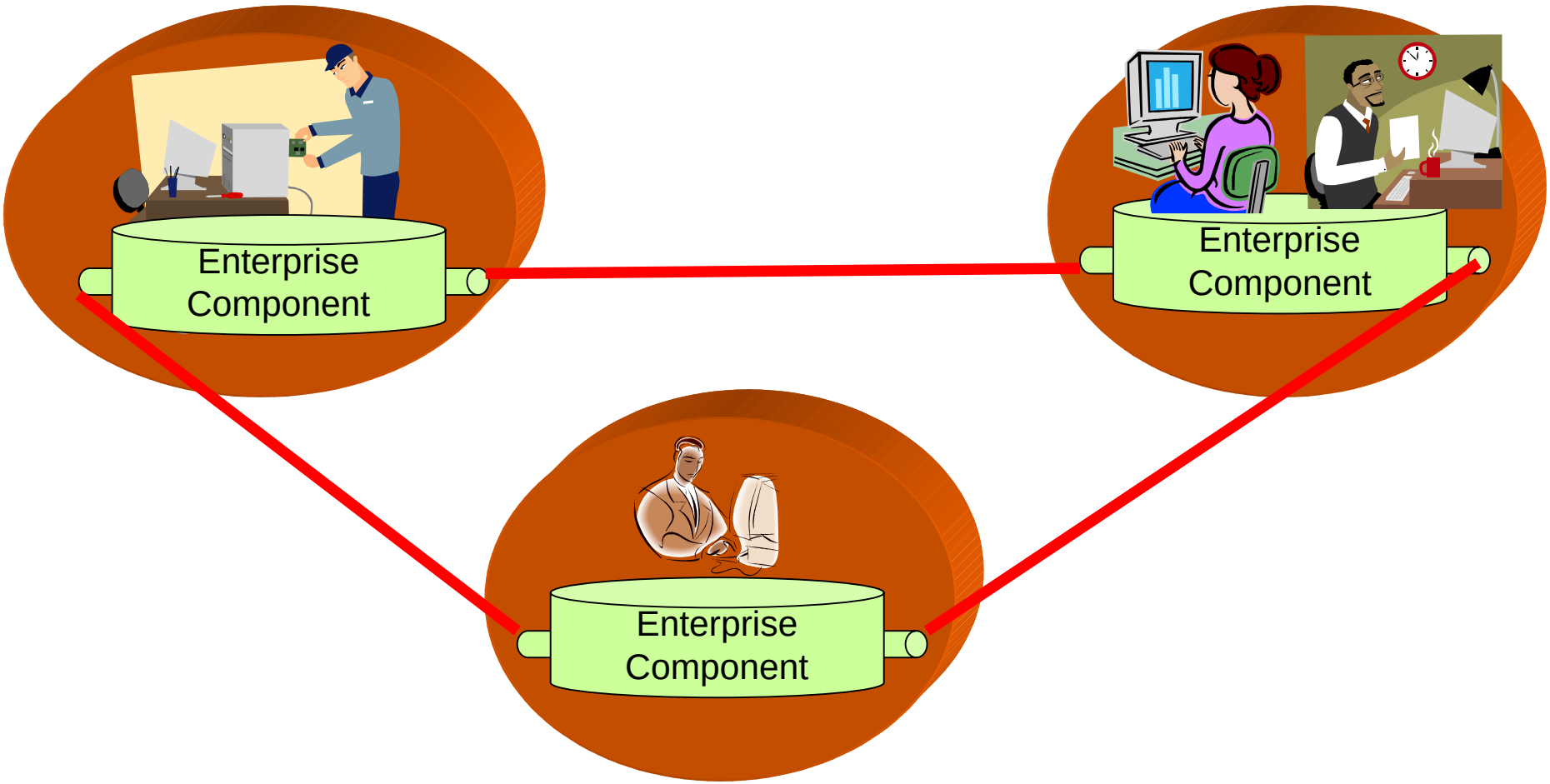
People, organizations
and systems
components work
together to realize roles



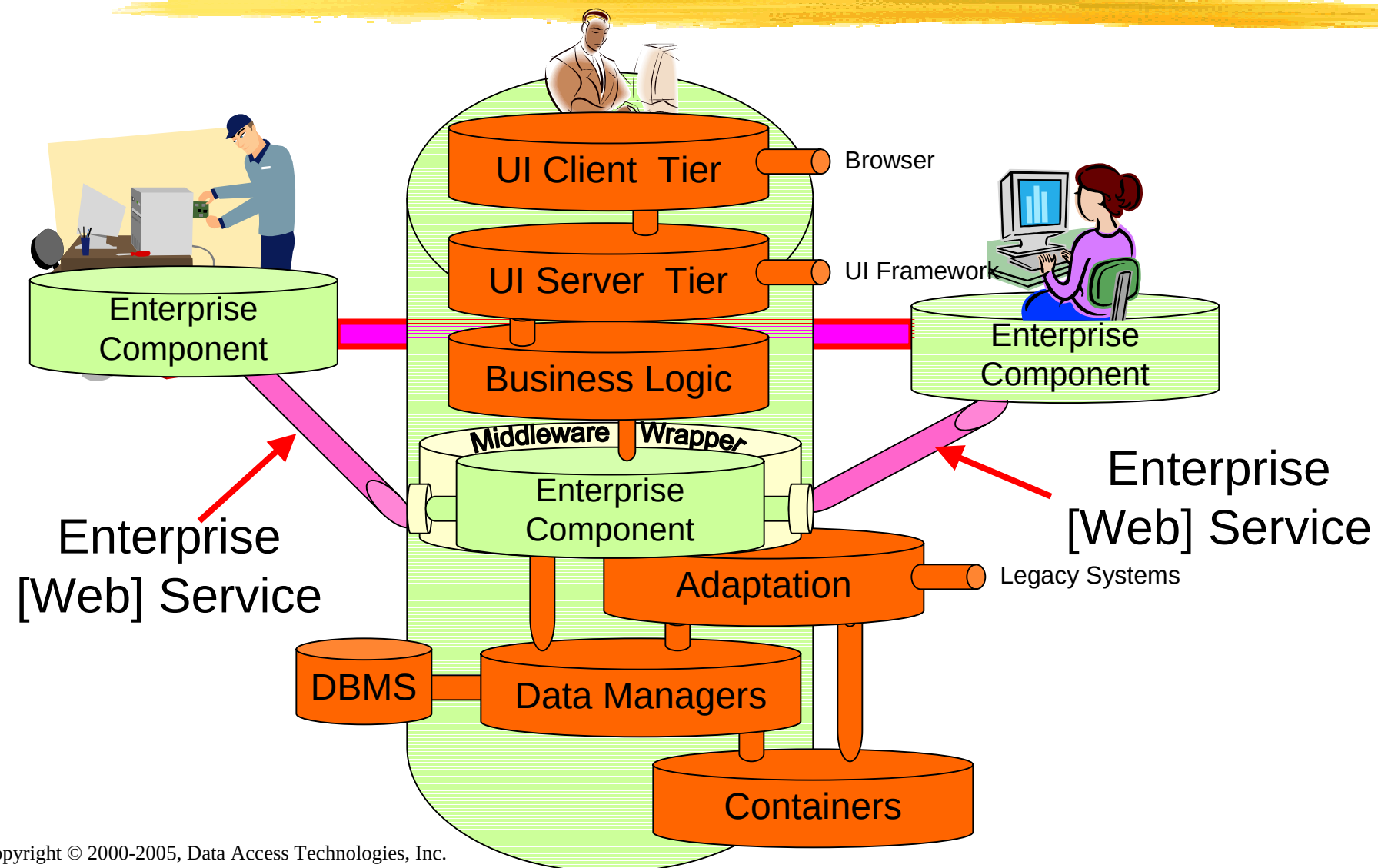
Components are the peoples
Automated assistant

From the system perspective.
People and organizations
become part of the implementation
Of the role

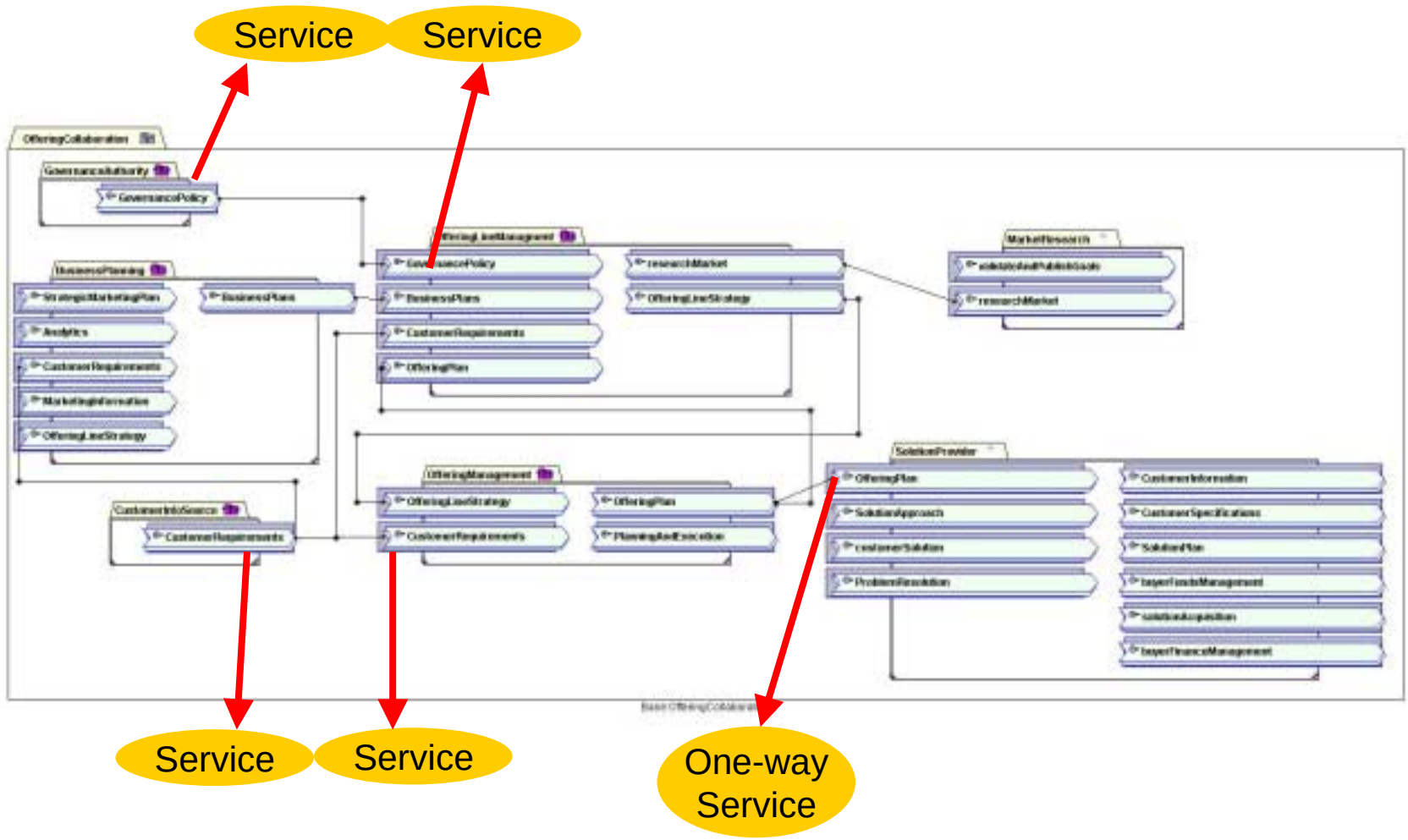
People, Components & Organizations Collaborating



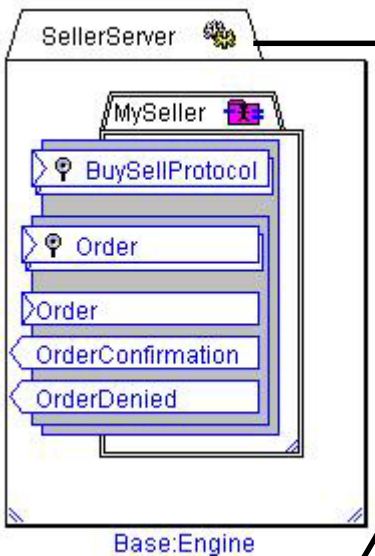
“Lower” PIM View - Enterprise Component Internals



PSM View - Mapping to [web] Services



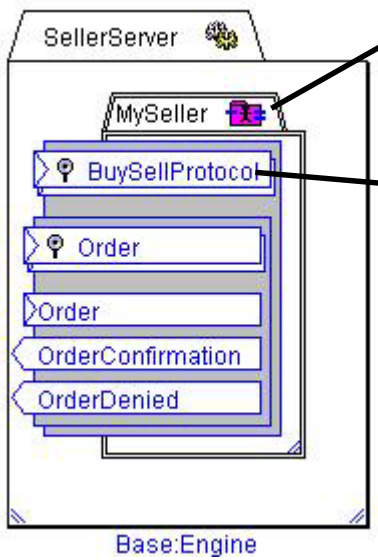
Mapping of a WSDL Engine



- <definitions xmlns="http://schemas.xmlsoap.org/ws
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soa
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mi
xmlns:http="http://schemas.xmlsoap.org/wsdl/http
ENC="http://schemas.xmlsoap.org/soap/encoding/
xmlns:xs2000="http://www.w3.org/1999/XMLSchem
xmlns:xs2001="http://www.w3.org/2001/XMLSchem
targetNamespace="urn:SellerServer" xmlns:tns="urn:S
xmlns:CoreTypes="urn:CoreTypes" xmlns:Ordering="ur
- <!--
definitions obtained from component /BuySell/Deployment/Selle

Aspects
WSDL
WSDL-SOAP

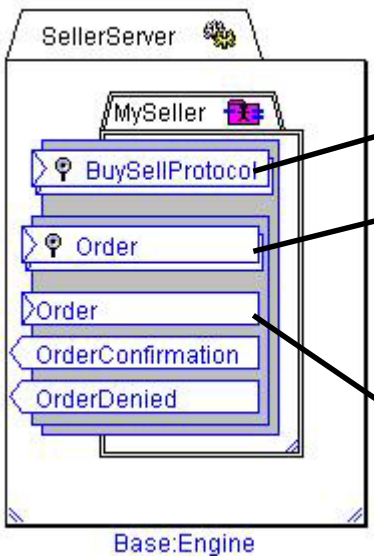
Mapping of an Enterprise Component



Aspects
WSDL
WSDL-SOAP

```
- <service name="MySeller">
- <!--
    implemented service role
/BuySell/Deployment/SellerServer/MySeller    -->
    <documentation><p> </p></documentation>
  - <port name="BuySellProtocol"
binding="tns:BuySellProtocol">
  - <!--
    original service port was
/BuySell/Deployment/SellerServer/MySeller/BuySellProtoc
ol (extending Component
&lt;/BuySell/SellerImplementation/MySeller/BuySellProtoc
ol&gt; )    -->
    <soap:address
location="http://localhost:8080/cx/app/BuyS
ell/Deployment/SellerServer/MySeller/BuyS
ellProtocol" />
    </port>
```

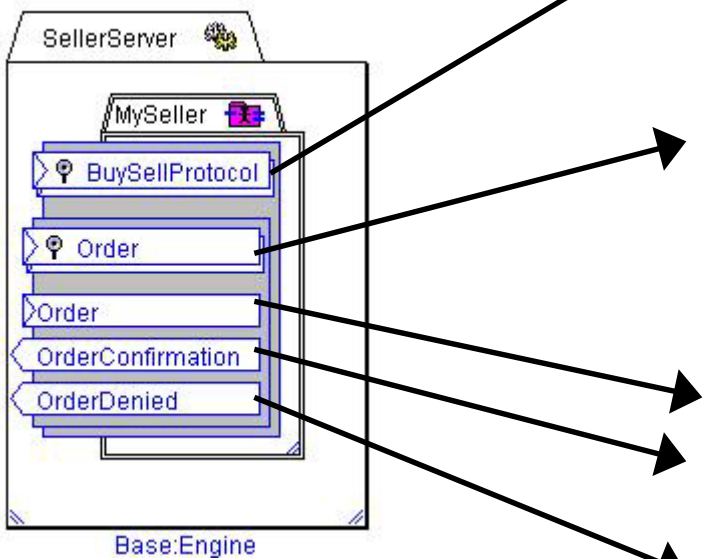
Mapping of a protocol binding



Aspects
WSDL
WSDL-SOAP

```
<binding name="BuySellProtocol"
 type="tns:BuySellProtocol">
<soap:binding
      transport="http://schemas.xmlsoap.org/soap/h
      style="rpc" />
<operation name="Order">
<soap:operation
      soapAction="urn:/BuySell/Community/BuySellP
      col/Order" style="rpc" />
<input name="Order">
<soap:body use="encoded" namespace="urn:SellerServ
      encodingStyle="http://schemas.xmlsoap.org/so
      encoding/" />
```

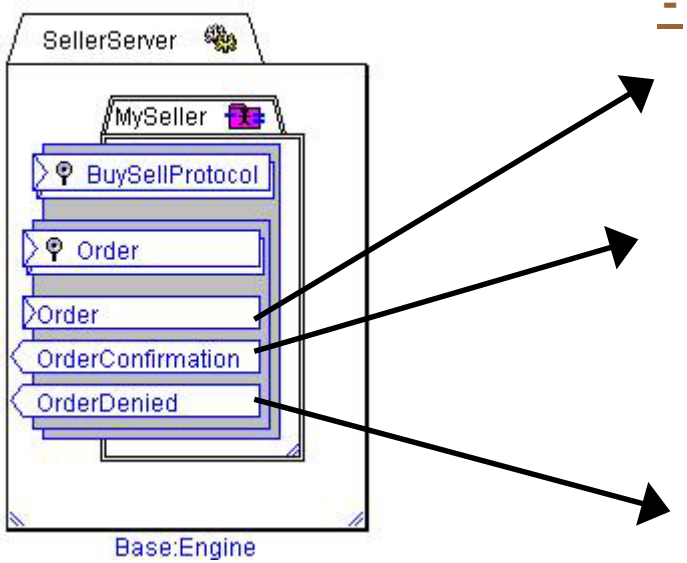
Mapping of a protocol




```
- <portType name="BuySellProtocol">
- <!--
  original cx operation =
/BuySell/Community/BuySellProtocol/Order    -->
- <operation name="Order">
- <!--
  original cx flow port =
/BuySell/Community/BuySellProtocol/Order/Order    -->
  <input name="Order" message="tns:Order" />
  <output name="OrderConfirmation"
message="tns:OrderConfirmation" />
  <fault name="OrderDenied"
message="tns:OrderDenied" />
  </operation>
  </portType>
```

Aspects
WSDL
WSDL-SOAP

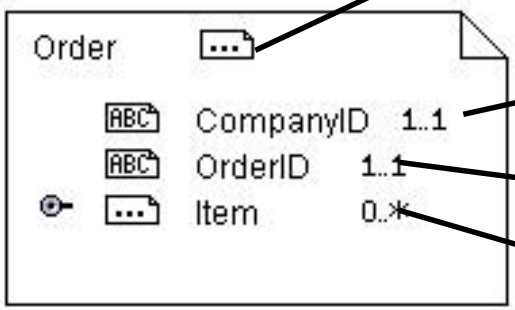
Mapping of message types



Aspects
WSDL
WSDL-SOAP

```
- <message name="Order">
    <part name="Order" type="Ordering:Order"
<message name="OrderConfirmation">
    <part name="OrderConfirmation"
  type="Ordering:OrderConfirmation" />
    </message>    </message>
- <message name="OrderDenied">
    <part name="OrderDenied"
  type="Ordering:OrderDenied" />
    </message>
```

Mapping of data types



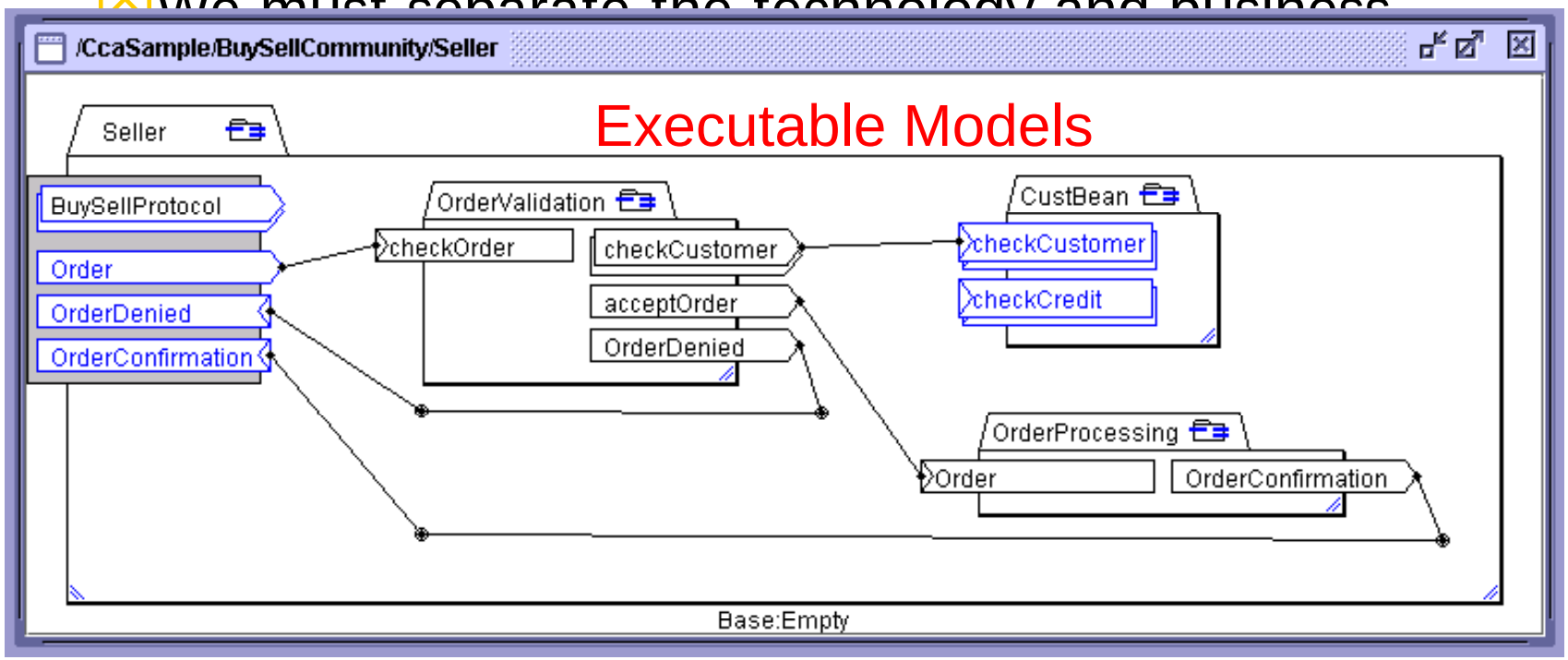

```
- <xs2001:complexType name="Order">

- <xs2001:sequence>

  <xs2001:element minOccurs="1"
maxOccurs="1" name="CompanyID"
type="CoreTypes:CompanyID" />

  <xs2001:element minOccurs="1"
maxOccurs="1" name="OrderID"
type="Ordering:OrderID" />

  <xs2001:element minOccurs="0"
maxOccurs="unbounded" name="Item"
type="Ordering:Item" />

  </xs2001:sequence>

  </xs2001:complexType>
```

High level tooling & infrastructure

- MUST BE SIMPLE!
  - We must be able to create better applications faster
  - We must separate the technology and business concerns, enable the user
- Tooling + Infrastructure
  - Executable models are source code
  - Tooling must be technology aware
  - Infrastructure must support tooling, not manual techniques
- Model based component architectures

High level tooling & infrastructure

- MUST BE SIMPLE!
  - We must be able to create better applications faster
  - We must separate the technology and business



Net effect

- Using these open standards and automated techniques we can;
  - 80% Reduction in complexity (Conservative)
  - Achieve the strategic advantage of an open and flexible enterprise
  - Produce and/or integrate these systems FASTER and CHEAPER than could be done with legacy techniques
  - Provide a lasting software asset that will outlive the technology of the day

Contact

Cory Casanave

Data Access Technologies

www.enterprisecomponent.com

cory-c@EnterpriseComponent.com