



The Leading-Edge Software Solutions

Enterprise Processes Automation Technologies

Domenico Rotondi

(domenico.rotondi@txt.it)

Corporate Research Division

Txt e-solutions SpA

Presentation Content



- Introduction
- Business Processes
- Business Documents
- SOA
- Enterprise Service Bus
- Enterprise Integration patterns (EIP)
- Business Rules Engine
- Demo

Where are we?

- Introduction
- Business Processes
- Business Documents
- SOA
- Enterprise Service Bus
- Enterprise Integration patterns (EIP)
- Business Rules Engine
- Demo

- **Presentation focus:**

- Quick overview of technologies that **offload** business logic from business applications

- **Why:**

- Flexibility
- Auditability of Enterprise Activities (e.g. **US Sarbanes–Oxley Act**)
- Application's cost cutting (SW reuse)
- HW/SW opportunities
- ...

- **Context:**

- Enterprise internal procedures
- Enterprise-2-Enterprise procedures

Where are we?

- Introduction
- **Business Processes**
- Business Documents
- SOA
- Enterprise Service Bus
- Enterprise Integration patterns (EIP)
- Business Rules Engine
- Demo

■ Definition:

Business Process



set of related, organised tasks to produce something (service or product) for someone (customer) according to an enterprise's strategies/needs

■ Relevant Elements:

- process **Objective** (produce something)
- defined **Outcome** (something)
- defined **Recipient** (customer)
 - The customer can be: internal (user) or external (customer/client, supplier, ...)
- enterprise **Strategy/Need**: add value to the enterprise
- identified and organised **tasks** (workflow)
- identified **roles/subjects** for each task
- as a **defined start** and an **end**

Types of business processes:

1. **Management Processes:** finalised to govern the system

- i.e.: overlook other business processes, pursuing of enterprise's objectives, ...

2. **Operational Processes:** devoted to the enterprise's core business

- Primary processes to create enterprise added value
- examples: Sales, Production, ...

3. **Supporting Processes:** provide specific support to other processes

- Finalised to optimize uses of resources (human/machines/money/...)
- examples: Purchasing, Human Resources Management, ICT, ...

Scope (extent) of business processes:

1. Internal Processes: process's tasks involve only internal subjects

- example: holiday authorization

2. External Processes: involvement of external subjects. External processes are normally further classified as:

- **Business-to-Business Processes (B2B):** involved subjects act within and on behalf of enterprises
 - Examples: Supply Chain Processes, Personnel Travel Booking, ...
- **Business-to-Consumer Processes (B2C):** at least one of the subject acts as a physical subject
 - Examples: end-users' sales, customers' support service, ...

Business Process Modeling:

■ Definition:

- **Representation/formalization** of business processes *(does not imply IT support)*

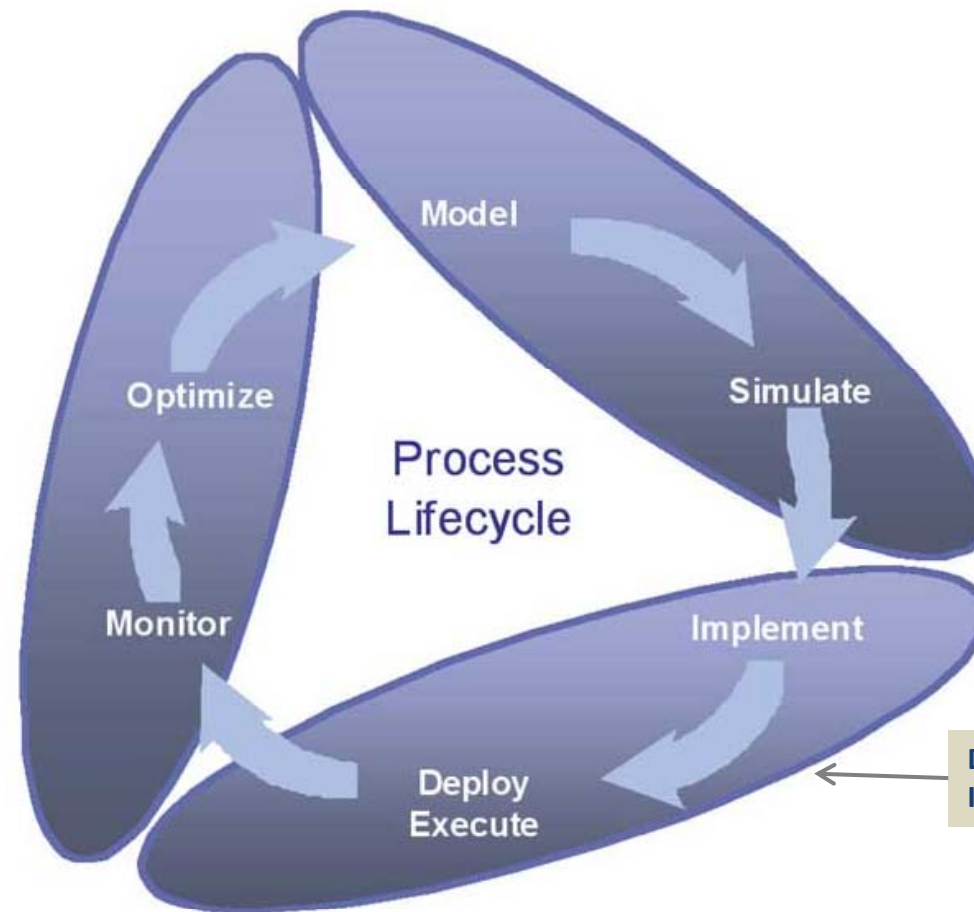
■ Objectives:

- have an **objective** (*unbiased*) representation of the business processes
- have a clear **identification** of processes' steps, information, involved subjects
- Have the possibility to perform **process simulations**
- have the possibility to easily **revise/improve** the processes

■ Actors:

- **Business Analysts**
- **Managers**

Business Process Lifecycle



Does not automatically imply
ICT technologies

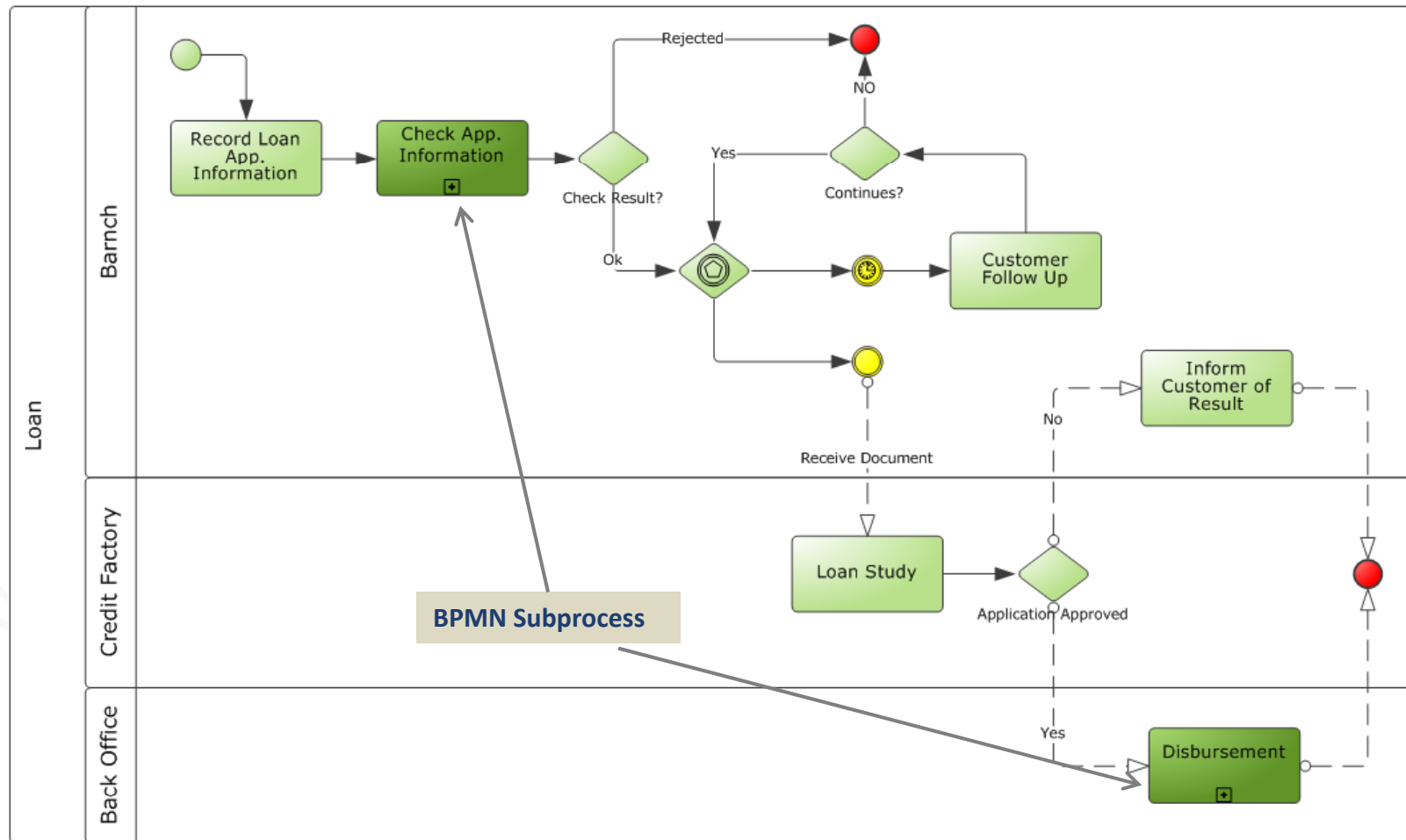
Business Process Modeling techniques:

■ Business Process Modeling Notation (BPMN):

- OMG (Object Management Group) standard (<http://www.bpmn.org/>)
- provides a graphical notation to structure business processes
- provides a mapping between notation elements and execution languages (e.g. **BPEL-Business Process Execution Language** - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel)
- Current version 1.2 (Jan 2009)
- V2.0 currently under development

■ UML, IDEF0, xBML (*Extended Business Modeling Language*), ...


BPMN Process Example (Loan Management)



Where are we?

- Introduction
- Business Processes
- **Business Documents**
- SOA
- Enterprise Service Bus
- Enterprise Integration patterns (EIP)
- Business Rules Engine
- Demo

■ Definition:

Business Document  A structured set of information,
(exchanged within business processes) to
be electronically moved between
business partners

■ Relevant Organisations & standards:

- UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business):
 - **UN/EDIFACT** (EDI standard)
 - **CCTS-Core** Component Technical Specification (syntax-neutral and technology-independent building blocks)
- OASIS (Organization for the Advancement of Structured Information Standards):
 - **UBL (Universal Business Language)**: library of electronic business documents structured as XML Schema Files
 - UBL based on **UN/CEFACT CCTS-Core**
 - Currently at **V2.0** (**V2.1** under ballot)

■ UBL 2.0:

- approved October 2006
- 31 business documents
- some localisation available (Italian, Spanish, Chinese, ...)
- URL: <http://docs.oasis-open.org/ubl/os-UBL-2.0/>

■ UBL 2.1:

- under ballot
- adds 33 new business documents
- new documents support: e-tendering / CPFR (collaborative planning, forecasting, and replenishment) / vendor managed inventory / intermodal freight management
- Specification URL: <http://docs.oasis-open.org/ubl/UBL-2.1.pdf>

Where are we?

- Introduction
- Business Processes
- Business Documents
- **SOA**
- Enterprise Service Bus
- Enterprise Integration patterns (EIP)
- Business Rules Engine
- Demo

■ What is SOA:

- application systems' design principles:
 - a SOA compliant system integrates a set of *distinct packages* (called **services**) providing specific functionalities
 - a **service** can be used by many different application systems (**service reuse**)
 - a **service** has a well define **service interface**, possibly formally defined (e.g. WSDL)
 - service's functionalities are accessed via 1+ calling mechanism (**service invocation**)

■ SOA protocols & standards:

- Web Services (HTTP/SOAP, RESTful-Representational State Transfer, ...): WS-*, ...
- SCA (Service Component Architecture): see OASIS SCA Home Page (<http://www.oasis-opencsa.org/sca>), Apache Tuscany (<http://tuscany.apache.org/home.html>)
- OSGi (??): see OSGi Home Page (<http://www.osgi.org/>)
- RPC ...

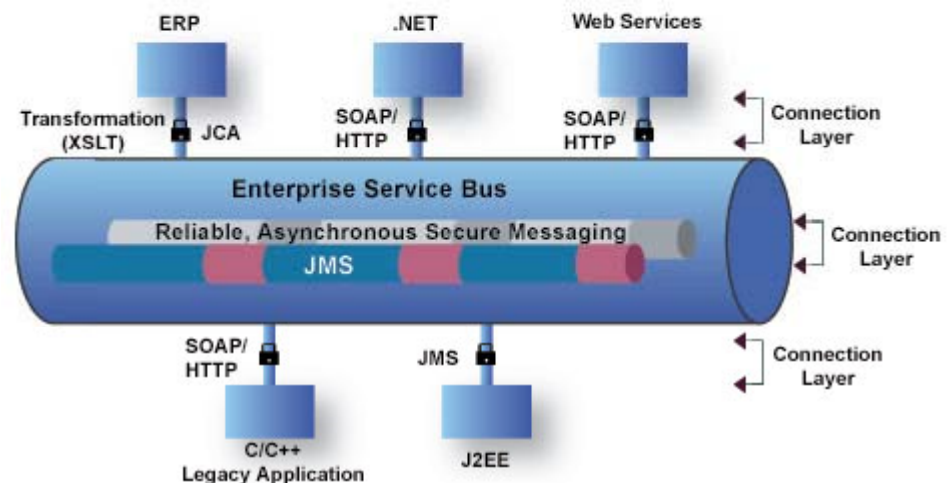
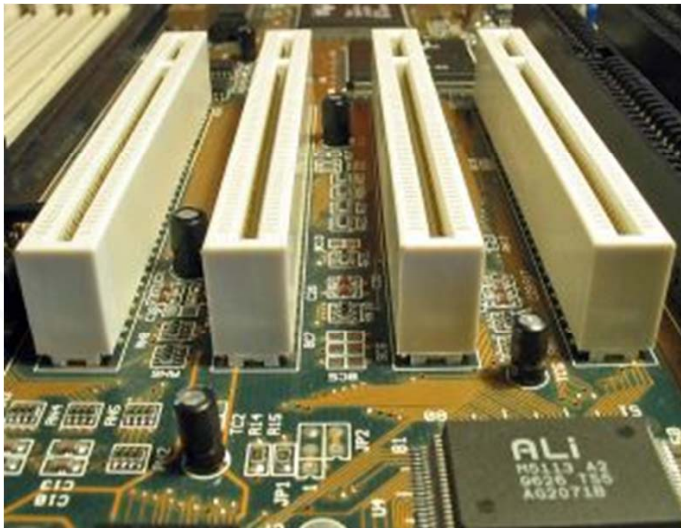
Where are we?

- Introduction
- Business Processes
- Business Documents
- SOA
- **Enterprise Service Bus**
- Enterprise Integration patterns (EIP)
- Business Rules Engine
- Demo

Enterprise Service Bus (1)

■ ESB???

- “A Web-services-capable infrastructure that supports intelligently directed communication and mediated relationships among loosely coupled and decoupled biz components” (Gartner Group definition)
- “a software architecture construct which provides fundamental services for complex architectures via an event-driven and standards-based messaging engine (the bus)” (Wikipedia definition)
- ...



■ Typical features:

- **Service Invocation:** synchronous and asynchronous service invocation + service mapping (locating and binding)
- **Routing:** messages are provided to the intended recipients (static/deterministic routing, content-based routing, rules-based routing, policy-based routing, ...)
- **Mediation:** decouple messages from the sender's to the receiver's (transport) protocol
- **Message Transformation:** transforms messages into a **Normalised Message Format**
- **Process choreography:** implementation of complex business processes
- **Service orchestration:** coordination of multiple implementation services exposed as a single, aggregate service
-

■ Typical Components:

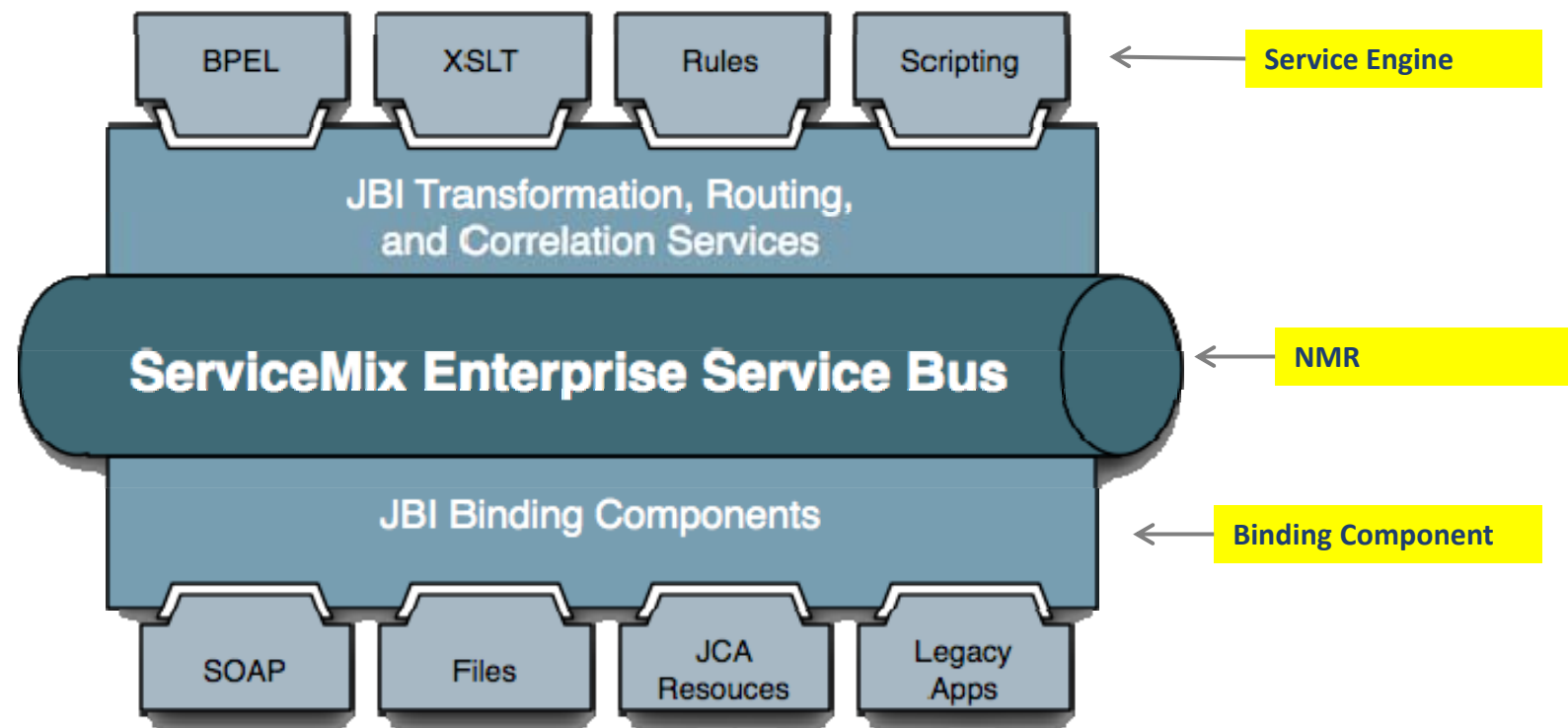
- **Normalised Message Router (NMR)**: is the core element in charge of message routing, service calling, ...
- **Binding Components (BC)**: pluggable components which connect external services to the NMR. Are in charge of:
 - supporting the external transport protocol (FTP, HTTP, SMTP, ...)
 - transforming the message from its external representation to the *Normalised Message Format*
- **Service Engines (SE)**: are service components that receive & send messages in *Normalised Message Format* on which they perform some specialised processing (e.g.: XSLT, rules application, ...)

■ Standards (Java World):

- **Java Business Integration (JBI)**: Java Community Process (JCP) **JSR 208**

■ Implementations:

- Commercial: TIBCO, IBM WebSphere ESB, Oracle ESB, Microsoft BizTalk, ...
- Open Source: Apache ServiceMix, Mule ESB, OpenESB, ...

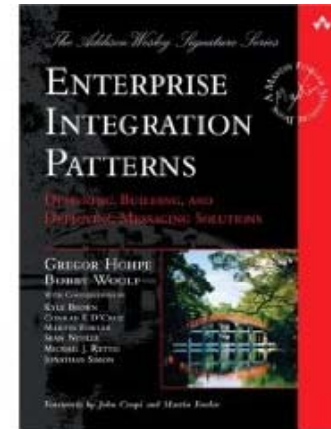


Where are we?

- Introduction
- Business Processes
- Business Documents
- SOA
- Enterprise Service Bus
- **Enterprise Integration patterns (EIP)**
- Business Rules Engine
- Demo

■ Enterprise Integration Patterns (EIPs)???

- Initially formalised in the book “**Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions**” by Gregor Hohpe and Bobby Woolf in 2003
- EIPs are **design patterns** normally present in enterprise application integration and message-oriented middleware
- **Design Pattern:** “In software engineering, a design pattern is a general reusable solution to a commonly occurring problem in software design. A design pattern is not a finished design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations” (Wikipedia)
- More details in the “**Patterns and Best Practices for Enterprise Integration**” <http://www.eaipatterns.com/>



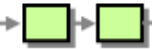
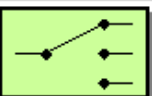
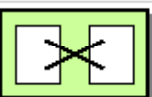
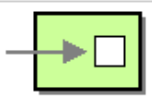


■ EIPs Categories:

- **Integration Styles:** document ways of applications integration (e.g.: File Transfer, RPC, shared DB, ...)
- **Channel Patterns:** fundamental attributes of a messaging system (e.g.: Point-2Point Channel, Pub-Sub Channel, Guaranteed Delivery Channel, ...)
- **Message Construction Patterns:** detail intent, form and content of messages (e.g.: Document Message, Event Message, Message Sequence, ...)
- **Routing Patterns:** identify mechanisms for routing messages from its sender to the correct receiver(s) (e.g.: Content-Based Routing, Filter, Splitter, ...)
- **Transformation Patterns:** identify mechanisms that change the information content (*body*) of a message (e.g.: Content Enricher, Normalizer, ...)
- **Endpoint Patterns:** behavior of messaging system clients (e.g.: Messaging Gateway, Polling Consumer, ...)
- **System Management Patterns:** identify features to keep a message-based system running (e.g.: Wire Tap, Message Store, Control Bus, ...)

■ EIPs Examples *(from Apache CAMEL documentation):*



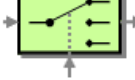


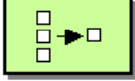
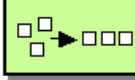


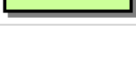
Messaging Systems

	Message Channel
	Message
	Pipes and Filters
	Message Router
	Message Translator
	Message Endpoint

Messaging Channels

	Point to Point Channel
	Publish Subscribe Channel
	Dead Letter Channel
	Guaranteed Delivery
	Message Bus

Message Routing

	Content Based Router
	Message Filter
	Dynamic Router
	Recipient List
	Splitter
	Aggregator
	Resequencer
	Composed Message Processor
	Scatter-Gather
	Routing Slip

Where are we?

- Introduction
- Business Processes
- Business Documents
- SOA
- Enterprise Service Bus
- Enterprise Integration patterns (EIP)
- **Business Rules Engine**
- Demo

■ Business Rules Engine???

- a software component that applies **business rules** on messages in a runtime environment (e.g. within an ESB based integration environment)
- **business rule**: is a statement that defines or constrains some aspect of a business process (e.g.: set upper threshold for expenses)

■ Why Business Rule Engines???

- business rule systems enable to offload company business policies and other operational decisions from application code
- make possible to define, test, execute, document and maintain rules independently from the application code
- More easy and cheaper to maintain
- ...

■ Types of Business Rules Engine:

- **Forward Chaining** engines: start with same data (message, event, ...), identify applicable rules and generates new data (messages, events, ...). Subcategories:
 - **production/inference rule engines**: rules like *"IF condition THEN action"*
 - **Reaction rule engines**: rules specify event patterns that the engine tries to identify within the event set (these can be event streams or more general collection of events)
- **Backward Chaining** engines: starts with a list of goals and works backwards from the consequent to the antecedent to see if there is data

■ Business Rules Engine standards???

- Rules definition & Rule Engine interfacing: practically inexistent
- Java World: **Java Rule Engine API (JSR 94)**

■ Implementations:

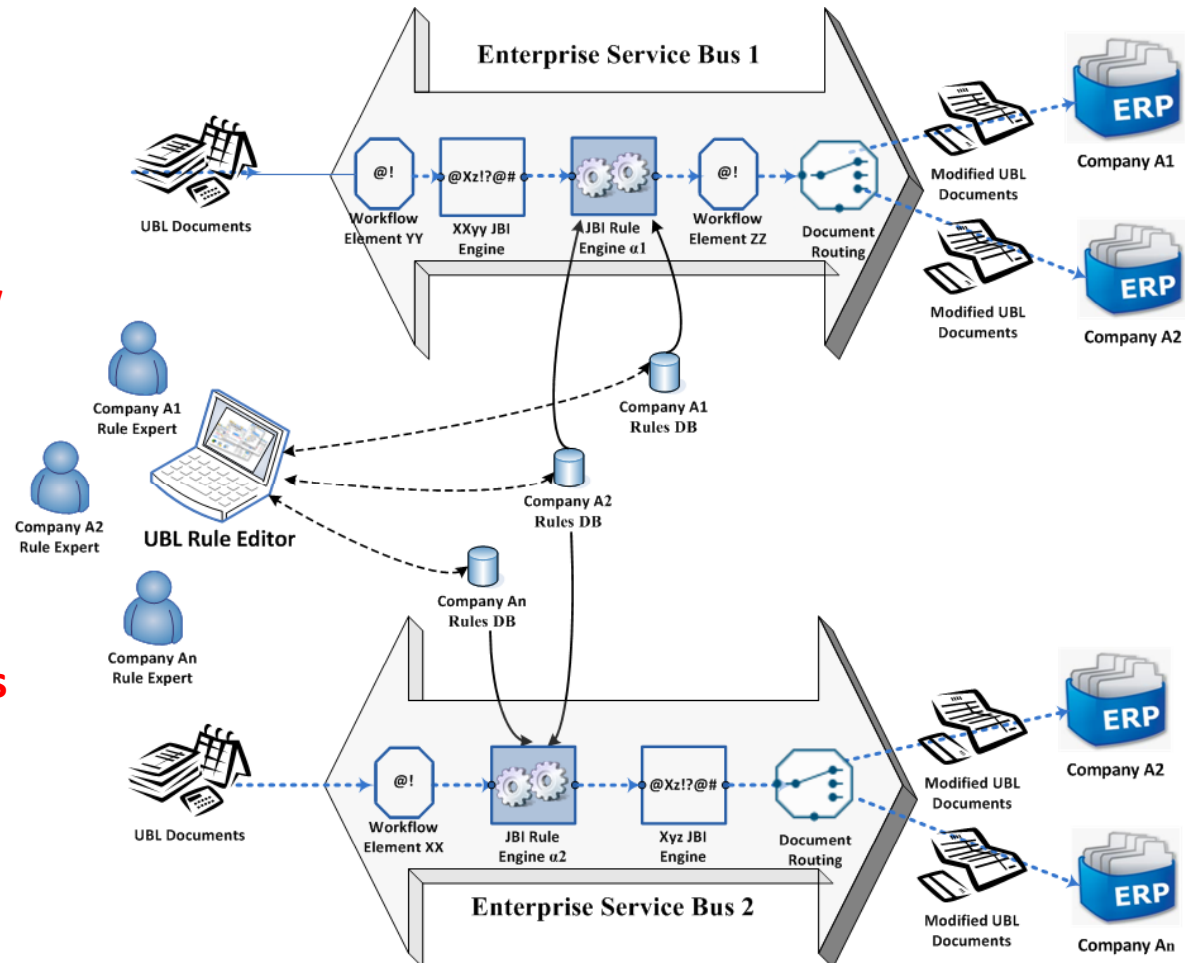
- Commercial: TIBCO BusinessEvents, IBM WebSphere ILOG BRMS, JBoss Enterprise BRMS, ...
- Open Source: Drools, OpenRules, JRuleEngine, JLisa, ...
 - **Drools**: <http://www.jboss.org/drools/>
 - **OpenRules**: <http://openrules.com/>
 - **JRuleEngine**: <http://jruleengine.sourceforge.net/>
 - **JSLisa**: <http://jlisa.sourceforge.net/>
 - ...

Where are we?

- Introduction
- Business Processes
- Business Documents
- SOA
- Enterprise Service Bus
- Enterprise Integration patterns (EIP)
- Business Rules Engine
- **Demo**

■ What is demoed?

- **OSGi**
- **OpenESB Fuji**
- **EIPs usage is workflow**
- **A UBL document rule editor**
- **Simple Business process demo**
- **SOAP Business process demo**



Thanks for your attention!

Questions????