JUDCon

JBoss Users & Developers Conference

2012:Boston

Deep dive into your business processes using jBPM

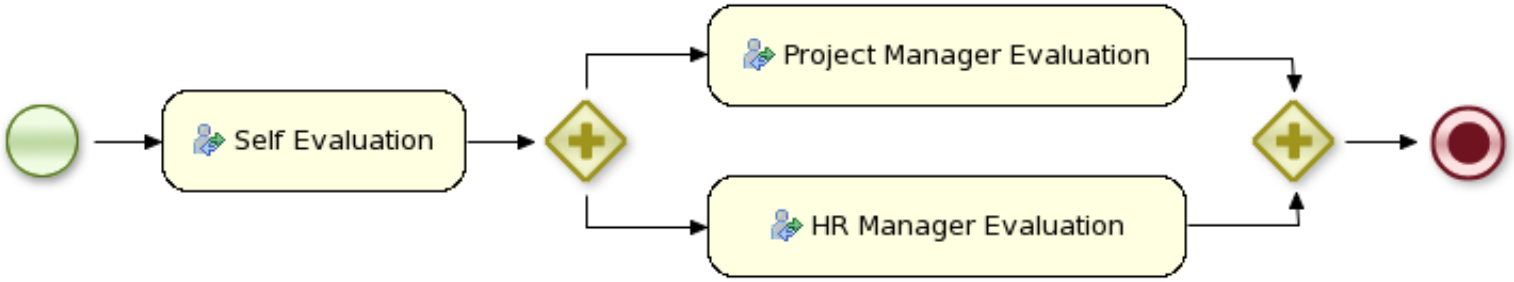
Kris Verlaenen
jBPM Project Lead

jBPM Overview

- jBPM Project
- jBPM walkthrough
- Demo
- Adaptive processes
- Roadmap

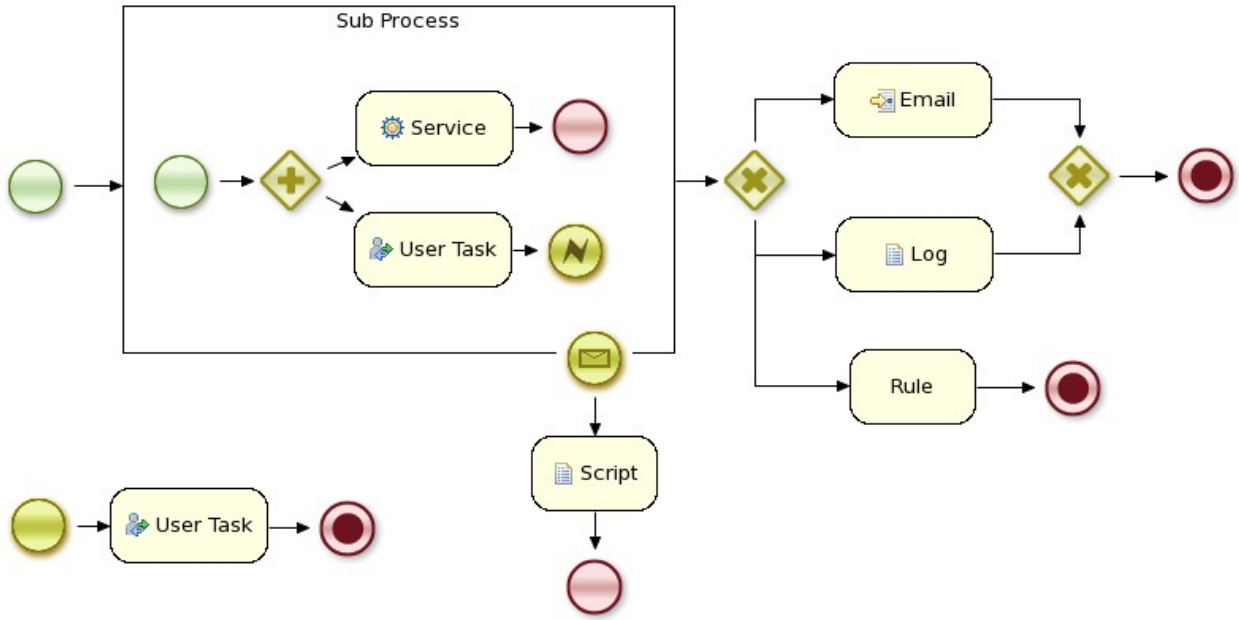
What is BPM?

Business Process Management



A ***business process*** is a process that describes the order in which a series of steps need to be executed, using a flow chart.

Why BPM?

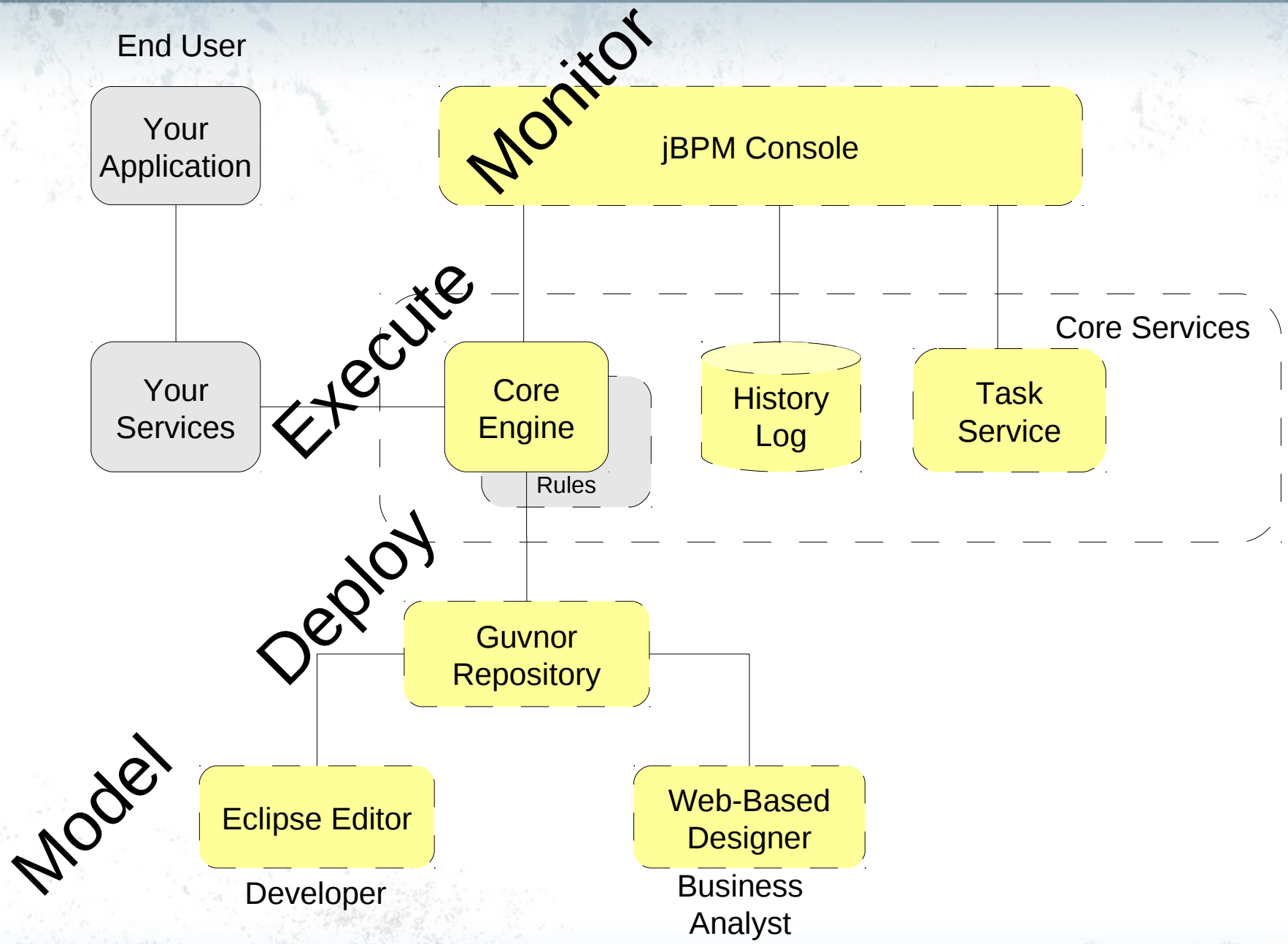


- Visibility
- Monitoring
- Higher-level

- Continuous improvement
- Speed of development
- Increased agility

jBPM Project

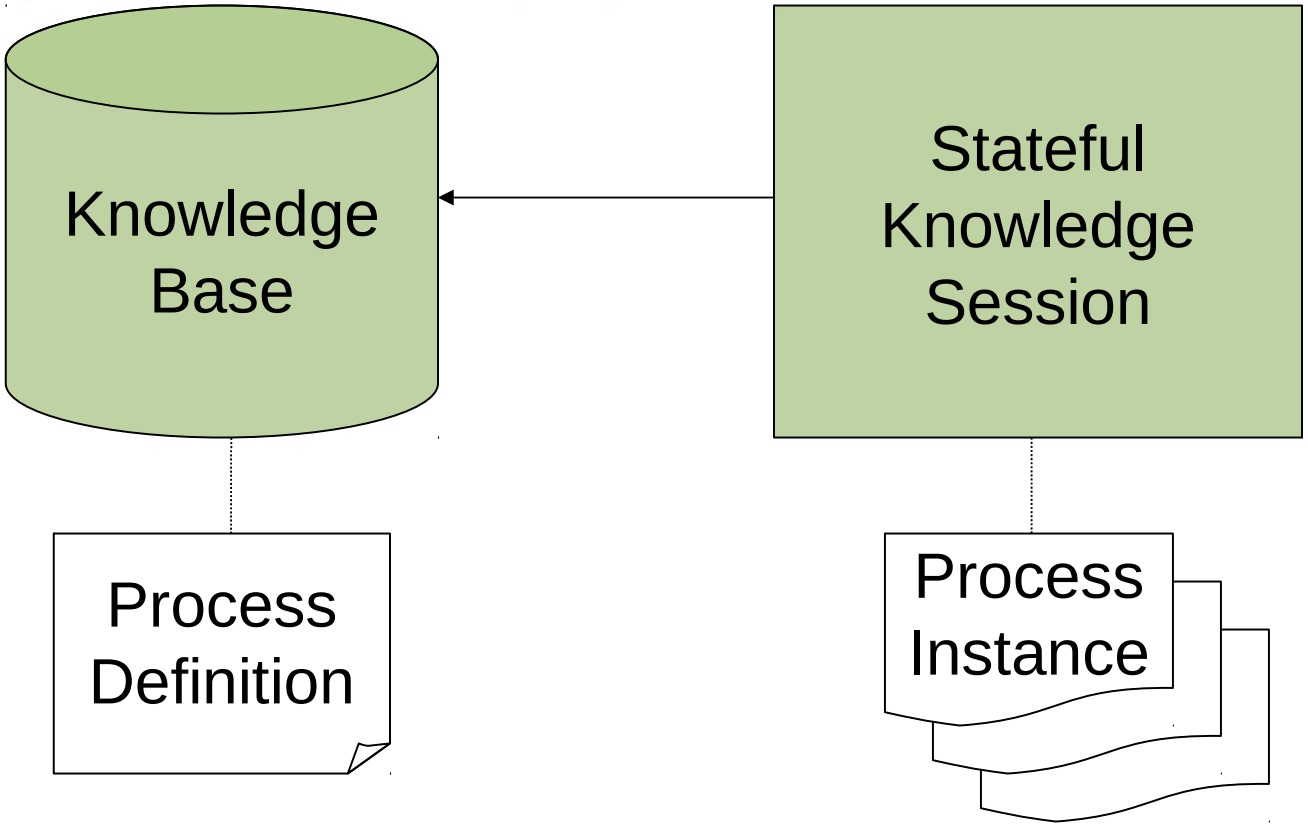
- (Executable) business processes
  - light-weight, native BPMN2 engine
  - from embedded to as a service
- Entire life cycle
- Both developers and business users
- Advanced, adaptive processes


End User
Your Application
Monitor
jBPM Console
Execute
Core Services
Your Services
Core Engine
Rules
History Log
Task Service
Deploy
Guvnor Repository
Model
Eclipse Editor
Developer
Web-Based Designer
Business Analyst

jBPM Walkthrough

- Core Engine
- BPMN2
- Domain-specific processes
- Installer + examples

Core Engine


Knowledge Base
Stateful Knowledge Session
Process Definition
Process Instance

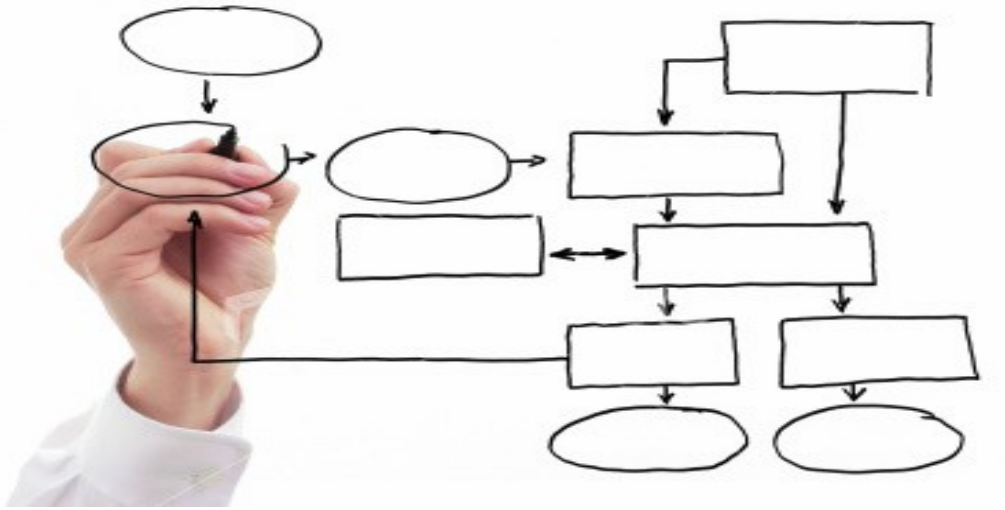
Java Interface

ProcessRuntime interface

- startProcess(processId)
- startProcess(processId, parameters)
- signalEvent(type, event)
- signalEvent(type, event, instanceId)
- abortProcessInstance(instanceId)
- getProcessInstance(instanceId)
- …

Java Example

```
// (1) Create knowledge base and add process definition
KnowledgeBuilder kbuilder = ...
kbuilder.add( ..., "sample.bpmn", ResourceType.BPMN2);
KnowledgeBase kbase = kbuilder.newKnowledgeBase();
// (2) Create new stateful knowledge session
StatefulKnowledgeSession ksession =
      kbase.newStatefulKnowledgeSession();
// (3) Start a new process instance
ksession.startProcess(“com.sample.bpmn.hello”);
```



BPMN 2.0 as the new standard

Activities

Task

Een **Task** is een afzonderlijke activiteit, een task die uitgevoerd moet worden. Met een ⊞ symbool wordt een Sub-Process aangegeven, waarmee een taskinvulling nader gespecificeerd kan worden.

Transaction

Een **Transaction** is een set activiteiten die een logisch geheel vormt; het kan een gespecificeerd transactieprotocol volgen.

Event Sub-Process

Een **Event Sub-Process** wordt in een Process of Sub-Process geplaatst. Het wordt geactiveerd als het start event ervan getriggered wordt en kan het proces op een hoger niveau onderbreken of parallel lopen zonder het proces te beïnvloeden.

Call Activity

Een **Call Activity** omvat een globaal gedefinieerd Sub-Process of Task die in het huidige proces wordt hergebruikt.

Activity Markers

Markers tonen het uitvoergedrag van een task:

- Sub-Process Marker
- Loop Marker
- Parallel MI Marker
- Sequential MI Marker
- Ad Hoc Marker
- Compensation Marker

Task Types

Types specificeren welke actie wordt uitgevoerd:

- Send Task
- Receive Task
- User Task
- Manual Task
- Business Rule Task
- Service Task
- Script Task

Sequence Flow

Definieert de volgorde waarin activiteiten uitgevoerd worden.

Default Flow

Is de standaard aftakking die gekozen wordt als alle andere voorwaarden negatief zijn.

Conditional Flow

Heeft een toegekende voorwaarde die definieert wanneer er wel of niet gebruikt gemaakt wordt van deze vertakking.

Gateways

Exclusive Gateway

Bij het splitsen van een stro exact één uitgaande stroom stromen bij elkaar komen, totdat er één input binnen

Event-based Gateway

Wordt altijd gevolgd door ‘ ‘ontvang taken’. De volgorc naar de eerstvolgende task

Parallel Gateway

Als de inkomende stroom g uitgaande stromen simultaa stromen bij elkaar komen, pas in gang gezet als alle in zijn.

Inclusive Gateway

Bij een splitsing worden één of meerdere stromen geactiveerd. Bij een samenvoeging van stromen moeten alle inkomende takken klaar zijn voordat de stroom verder gaat.

Exclusive Event-based Gateway (instantiate)

Er wordt een nieuwe proces

Events

	Start: Top-Level	Start: Event Sub-Process Interrupting	Start: Event Sub-Process Non-Interrupting	Intermediate: Catching	Intermediate: Boundary Interrupting	Intermediate: Boundary Non-Interrupting	Intermediate: Throwing	End
None: ongetypeerde gebeurtenis, zoals een startpunt of statusverandering.	○						◎	●
Message: verzenden en ontvangen van berichten.	✉	✉	✉	✉	✉	✉	✉	✉
Timer: cyclische timer gebeurtenissen, punten in de tijd, tijdspanne of time-out.	🕒	🕒	🕒	🕒	🕒	🕒		
Escalation: escaleren naar een hoger niveau van verantwoordelijkheid.		▲	▲		▲	▲	▲	▲
Conditional: reactie op veranderende procescondities of integratie van BusinessRules.	▤	▤	▤	▤	▤	▤		
Link: twee corresponderende link events zijn gelijk aan een ‘Sequence Flow’.				⇨			➔	
Error: opvangen of werpen van gedefinieerde fouten.		⚡			⚡			⚡
Cancel: reactie op geannuleerde transacties of					⊗			⊗
					◀◀		◀◀	◀◀
				◆	◆	◆	▲	▲
				⬟	⬟	⬟	⬟	⬟
				✚	✚	✚		
								●

Data

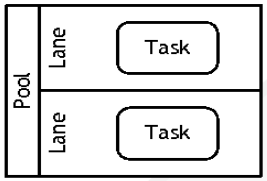
Een **Data Input** is een externe input voor het gehele proces. Het kan door een Activity gelezen worden.
Een **Data Output** is een variabele aanwezig als resultaat van het gehele proces.

Een **Data Object** representeert informatie die door het proces stroomt, zoals zakelijke documenten, e-mail of brieven.

Een **Collection Data Object** representeert een verzameling informatie, bijvoorbeeld een lijst van bestelde goederen.

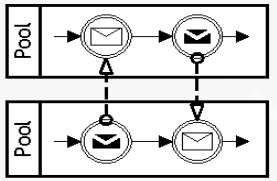
Een **Data Store** is een plaats waar het proces data heen kan schrijven en kan uitlezen, bijvoorbeeld een database. Het blijft bestaan na

Swimlanes



Pools (Participants) en **Lanes** representeren verantwoordelijken voor activiteiten in een proces. Een pool of een lane kan een organisatie, een rol of een systeem zijn. Pools worden op hiërarchische wijze door Lanes onderverdeeld.

Message Flow symboliseert informatiestromen tussen organisaties. Een Message Flow kan worden verbonden aan Pools, Activities of Message Events.

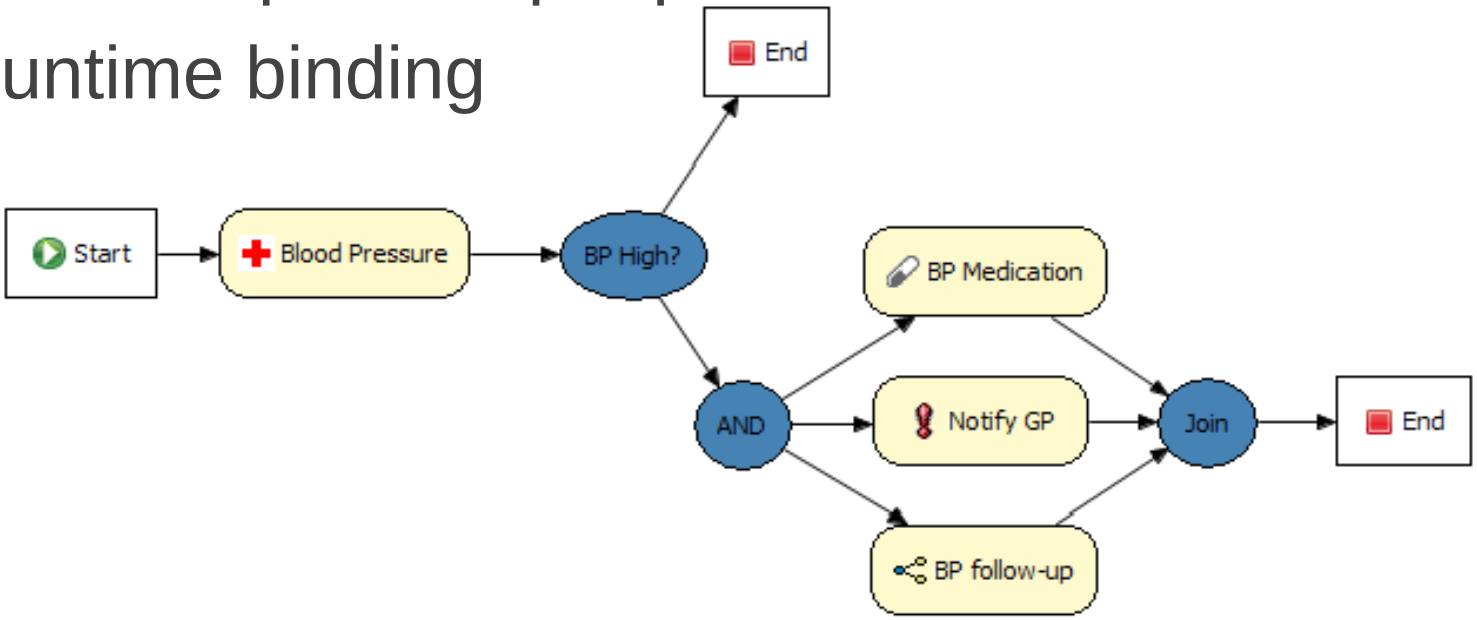


De volgorde van berichtuitwisseling kan worden gespecificeerd door de combinatie van Message Flow en Sequence Flow.

JBoss

Domain-specific Processes

- Extend palette with domain-specific, declarative service nodes
  - define input / output parameters
  - runtime binding



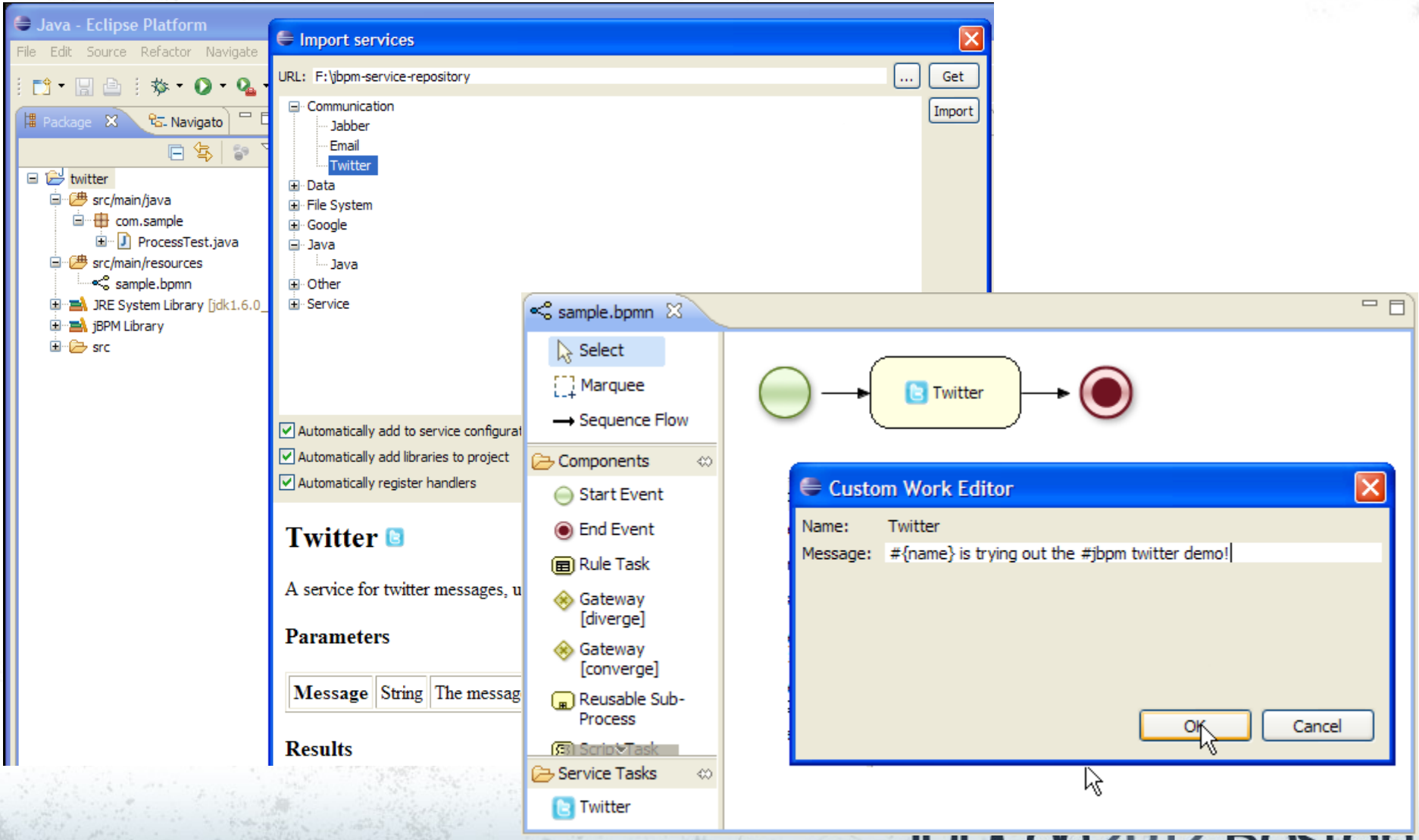
Domain-specific Processes

- Why?
  - Domain-specific
  - Declarative
  - High-level
  - Context-defined

Repository !

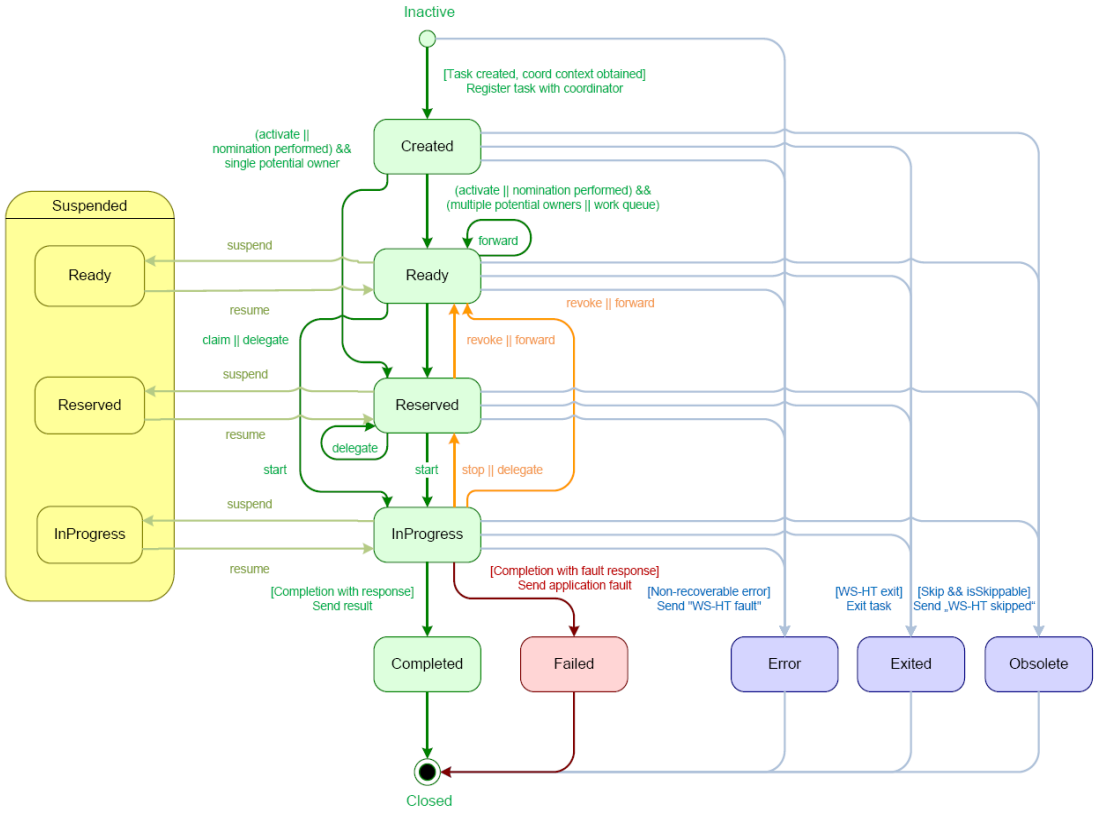
- Other examples
  - Human task
  - Java method
  - WS, Rest
  - Email, Twitter
  - FTP, RSS, Jabber
  - Finder, Exec, Archive
  - Google Calendar
  - *YOUR SERVICES !*

Service Repository



Human task service

- User task
- Human task service (WS-HT)
  - Task lists
  - Task life cycle
- Task clients
  - Task forms



Extra Features

Extra features

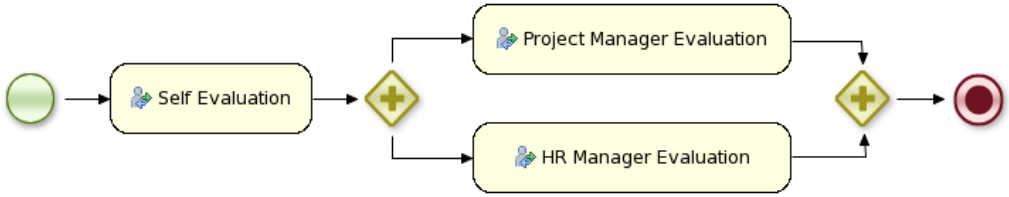
- Process instance migration
- JUnit testing
- Spring
- OSGi
- Migration from jBPM3(/4)

Installer

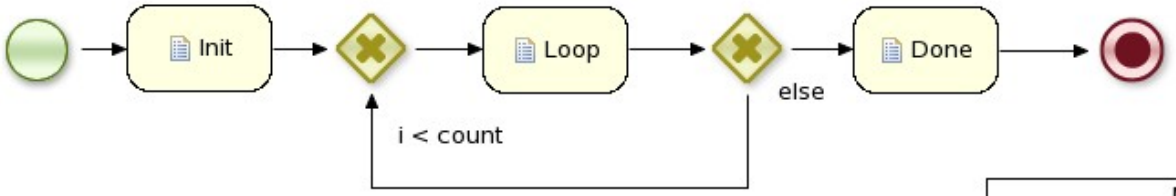
- Out-of-the-box working environment
- Sample process
- Components
  - Eclipse (jBPM + Drools)
  - Guvnor + Designer
  - jbpm-console
  - Services
    - H2 DB, human task service, history log

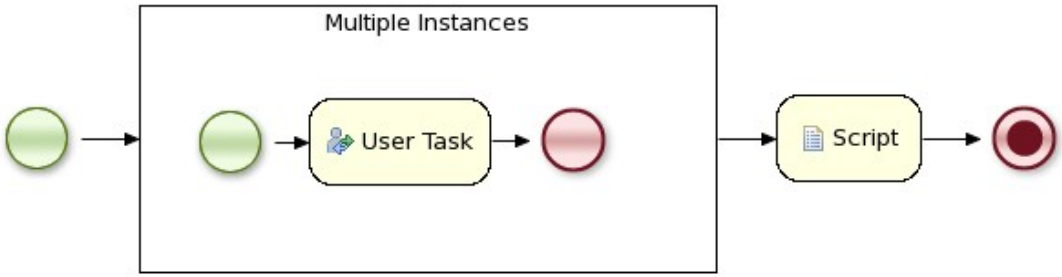
Examples

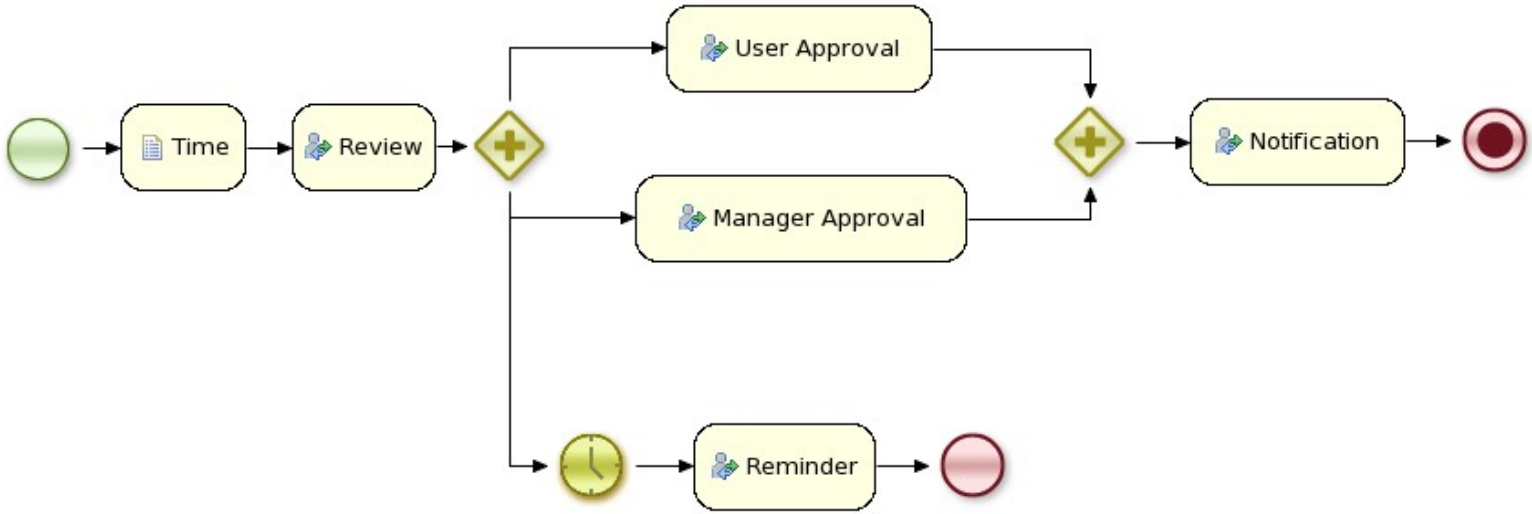
- Evaluation

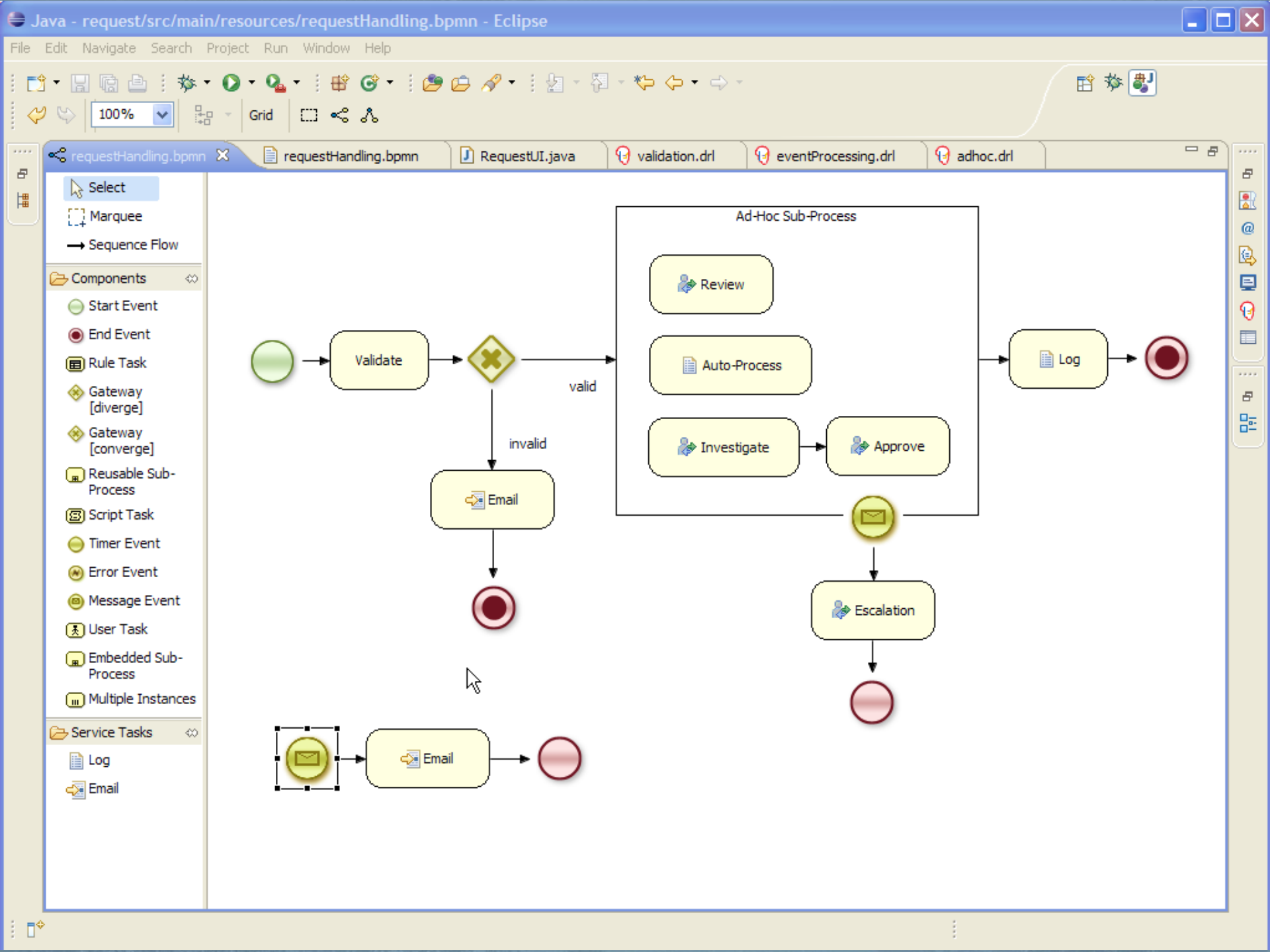


- Examples module
  - Looping
  - Multi-instances
  - Human tasks (including user / group assignment, delegation, forms, etc.)
  - Process + rules integration


Init
Loop
Done
else
i < count


Multiple Instances
User Task
Script


User Approval
Time
Review
Notification
Manager Approval
Reminder


Java - request/src/main/resources/requestHandling.bpmn - Eclipse
File Edit Navigate Search Project Run Window Help
100%
Grid
requestHandling.bpmn
requestHandling.bpmn
RequestUI.java
validation.drl
eventProcessing.drl
adhoc.drl
Select
Marquee
Sequence Flow
Components
Start Event
End Event
Rule Task
Gateway [diverge]
Gateway [converge]
Reusable Sub-Process
Script Task
Timer Event
Error Event
Message Event
User Task
Embedded Sub-Process
Multiple Instances
Service Tasks
Log
Email
Ad-Hoc Sub-Process
Review
Auto-Process
Investigate
Approve
Validate
valid
invalid
Email
Log
Escalation
Email

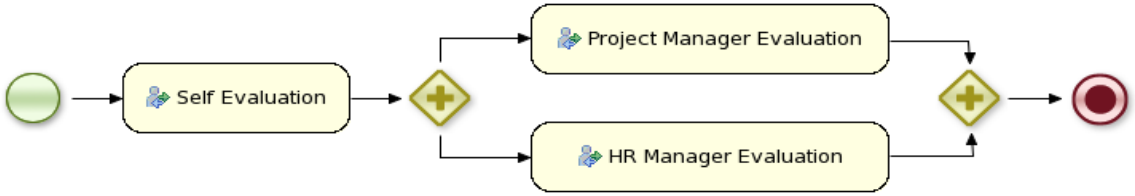
JBoss AS 7 + OpenShift

- AS 7
  - Blazingly fast
  - Lightweight
  - Module core
- OpenShift
  - AS 7 in the cloud

Demo

Ready for the future?

Traditional BPM systems have problems with change, complexity, flexibility, data-intensive applications, etc.



- Adaptive Case Management (ACM)
- Unstructured, non-lineair or flexible processes
- Event-driven BPM (edBPM)

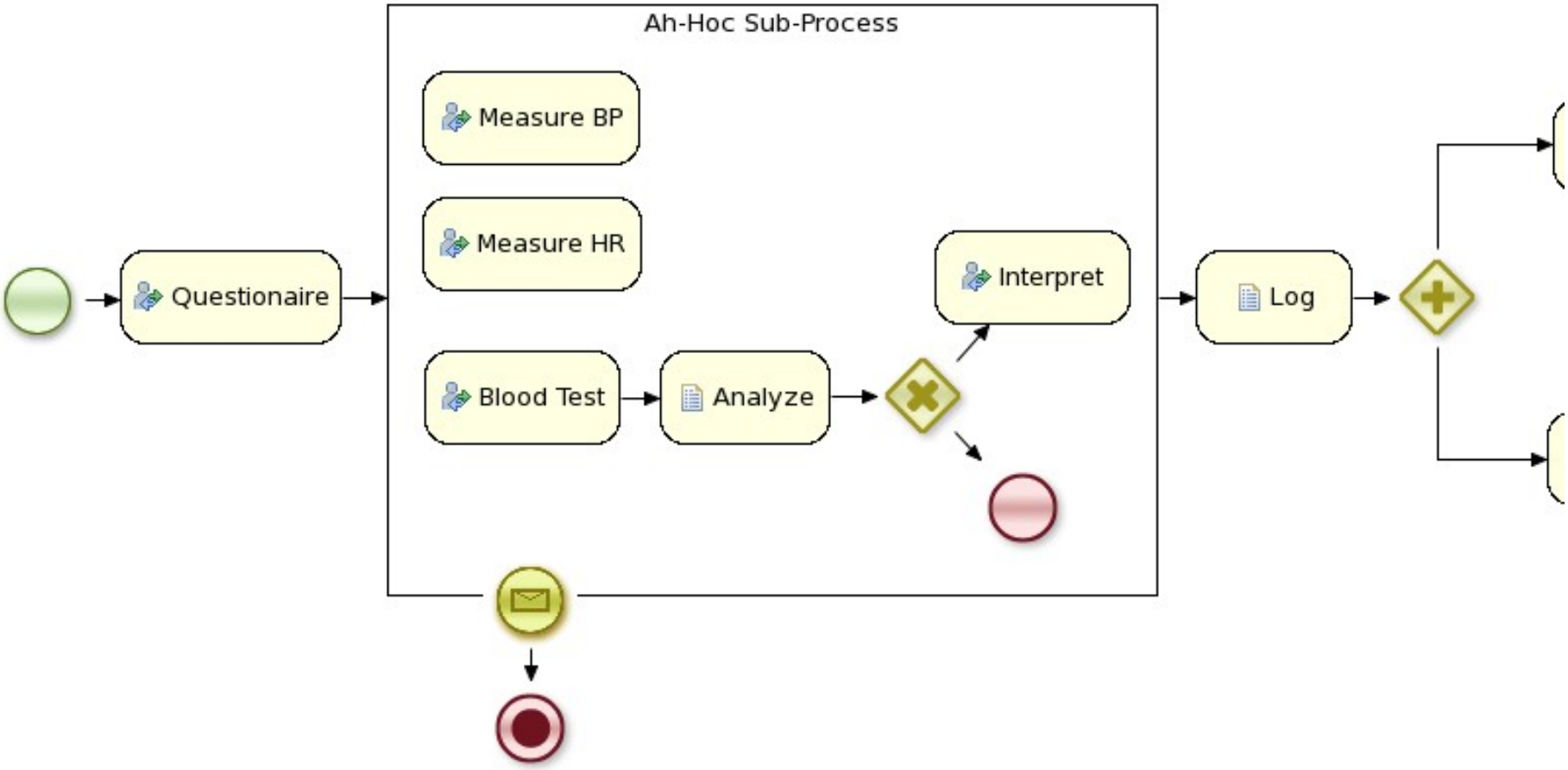
Processes

Rules

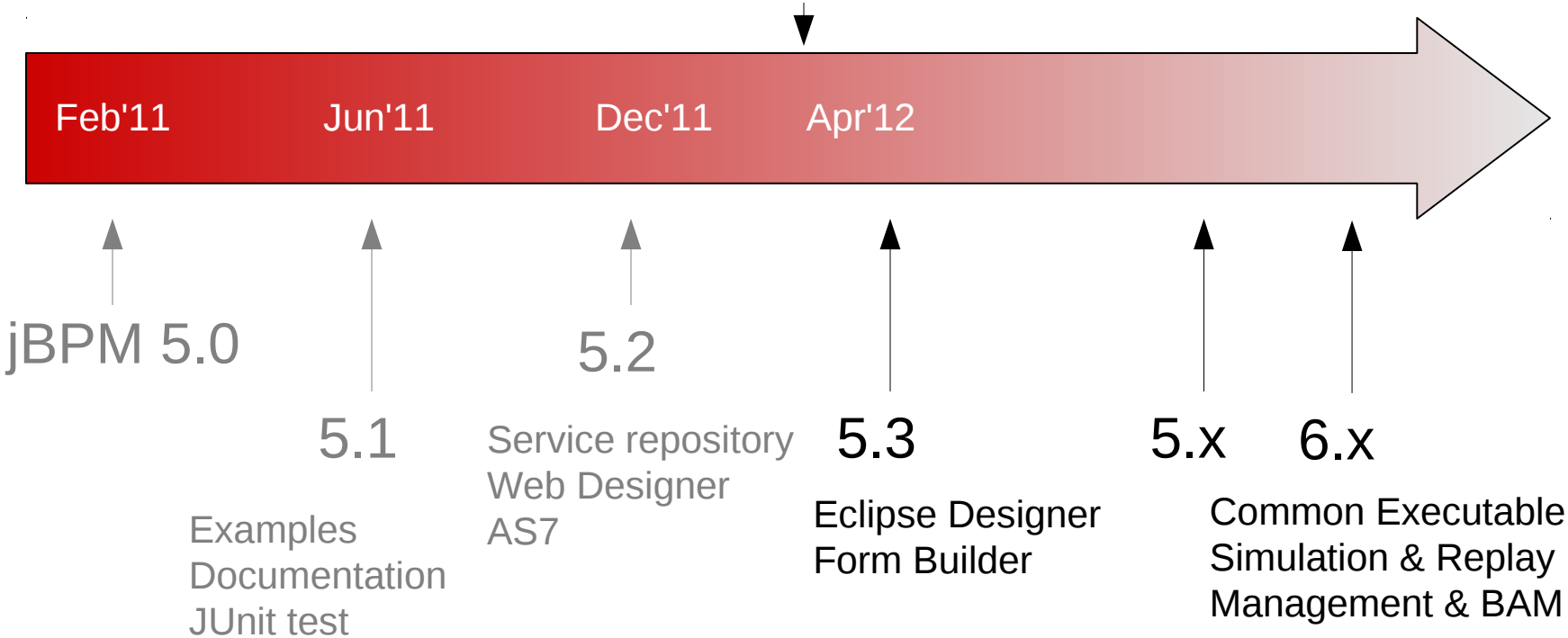
Events

Business Logic

Example: Clinical DSS

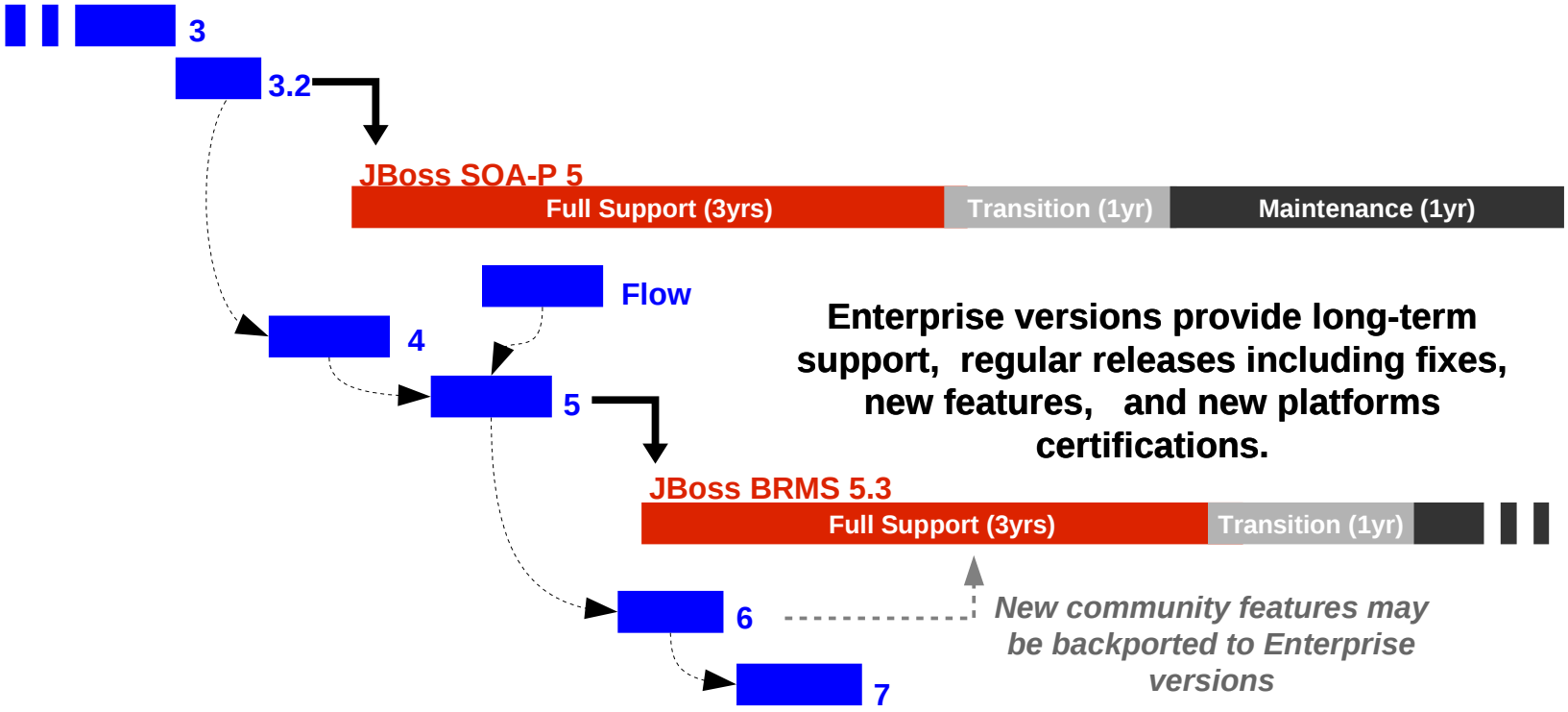


Roadmap


Feb'11
Jun'11
Dec'11
Apr'12
jBPM 5.0
5.1
Examples
Documentation
JUnit test
5.2
Service repository
Web Designer
AS7
5.3
Eclipse Designer
Form Builder
5.x
6.x
Common Executable
Simulation & Replay
Management & BAM

JBoss Enterprise Product Delivery Model

Example: jBPM Community and Enterprise Product



While **community projects** continue to rapidly evolve,
enterprise middleware products focus on long term stability.

jBPM: What, where?

- jBPM home page
- Source http://github.com/droolsjbpm/jbpm
- Hudson http://hudson.jboss.org/hudson/job/jBPM
- Blog http://kverlaen.blogspot.com/
- #jbpm on irc.codehaus.org
- jbpm-dev@jboss.org
- jBPM user forum

Drools & jBPM
All Day Drop in Centre

Friday June 29th

Room 105

Hynes Convention Centre

Deep dive into your business processes using jBPM

Kris Verlaenen
jBPM Project Lead