



JBoss Enterprise Business Rules

Business Rules Management System (BRMS)

Eric D. Schabell

JBoss Senior Solution Architect Benelux

COLLABORATION  WORK

Agenda

- **Introduction**
- **JBoss BRMS**
 - Overview
 - Future directions
- ***Demo***
- ***Questions***

JBoss BRMS v5.2

- **Declarative Business Rules**
 - Avoids hard-coding business rules in business logic
 - Rules can change without Java code changes or re-compilation
 - Expert system allowing solution of complex problems
- **Multiple Authoring Options:** DRL, DSL, CSV and Programmatic Rules Definition
 - Application objects (facts) mixed with conditions (rules) – similar to HQL
 - POJO and Declarative Fact Model
 - Decision Tables in Excel/Open Office
 - DSL Natural Language Extensions
- **Rule Repository**
 - Versioning of business rule-related artifacts (e.g. Fact models, enumerations, functions, DSL definitions, rules, tests, etc.)
- **Rule Manager**
 - RIA for creation & maintenance of business rule artifacts

JBoss BRMS

What is a rule engine?

- **A rule engine at its core is an environment/shell for capturing knowledge**
 - Applying that knowledge to specific data (facts)
 - Uses production rules
 - IF <conditions> THEN <actions>
 - rules express logic
- **Has roots in AI research**
 - Success of “expert” systems in the past triggered popularity of rule engines

JBoss BRMS

What is it going to do for me?

- **Externalize business logic**
 - Logic that is complex
 - Logic that changes often
 - Logic that just doesn't fit neatly in code
 - Logic that means more to domain experts than it does programmers
- **Keep application and user interface logic in your application**
 - Business logic in the rule engine
- **Use domain objects as interface to rule engine**

JBoss BRMS

When should I use a rules engine?

- When there is no satisfactory “traditional” solution
- The problem becomes too complex to express using traditional methods
- No known algorithms for solving the problem traditionally
- Too hard
- Too “fluid”

JBoss BRMS

What is a rule?

```
when
    <conditions>
then
    <actions>
```

- **In short, a rule is an premise and conclusion**
 - *If it is raining, then the ground must be wet*
- **A rule is made of conditions and actions. When all the conditions are met, a rule may “fire”.**
- **The conditions are collectively referred to as the LHS (left hand side) and the actions are referred to as the RHS (right hand side, or consequence).**
- **A rule operates on facts (data). In our case, these facts are instances of objects in our application.**

JBoss BRMS

What is inferencing?

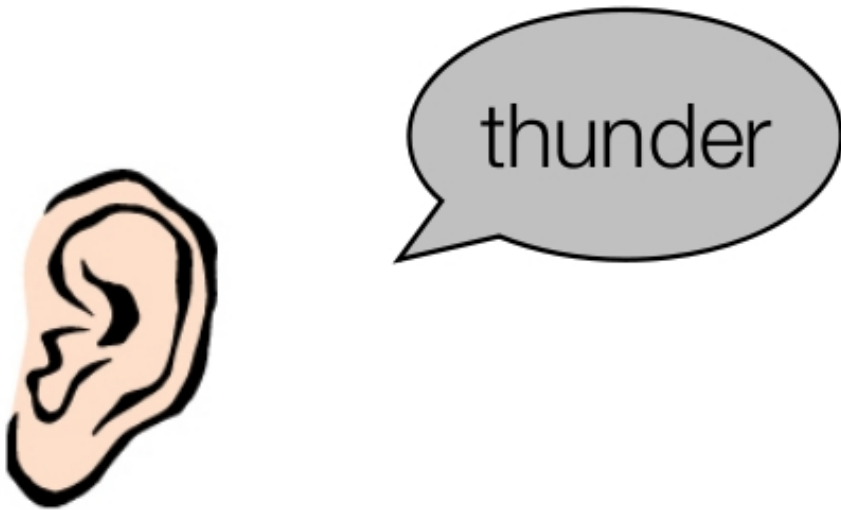
- We want to take all of our data (facts) and combine it with our knowledge (rules) and come up with some conclusions (inferencing)



JBoss BRMS

What is inferencing?

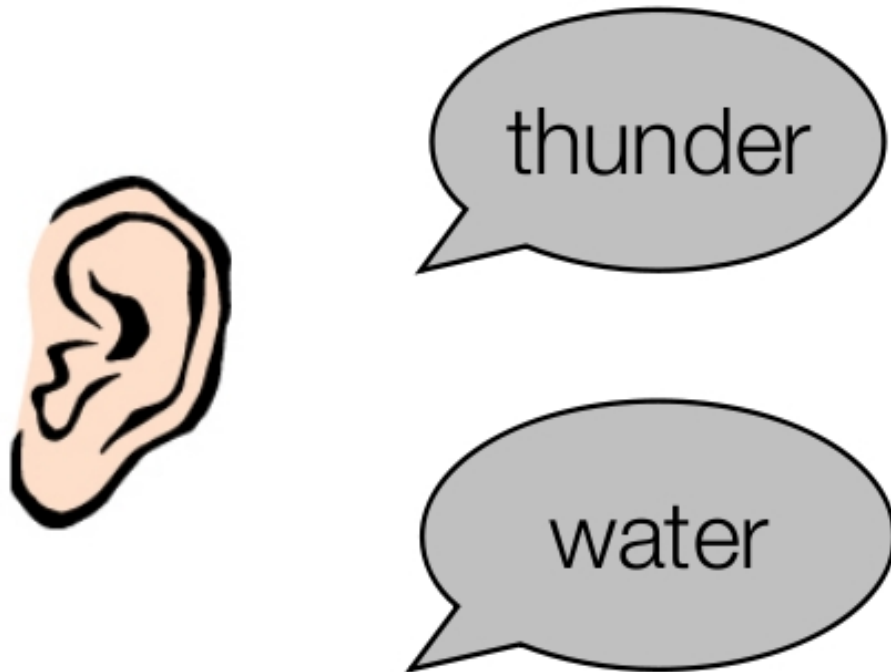
- We want to take all of our data (facts) and combine it with our knowledge (rules) and come up with some conclusions (inferencing)



JBoss BRMS

What is inferencing?

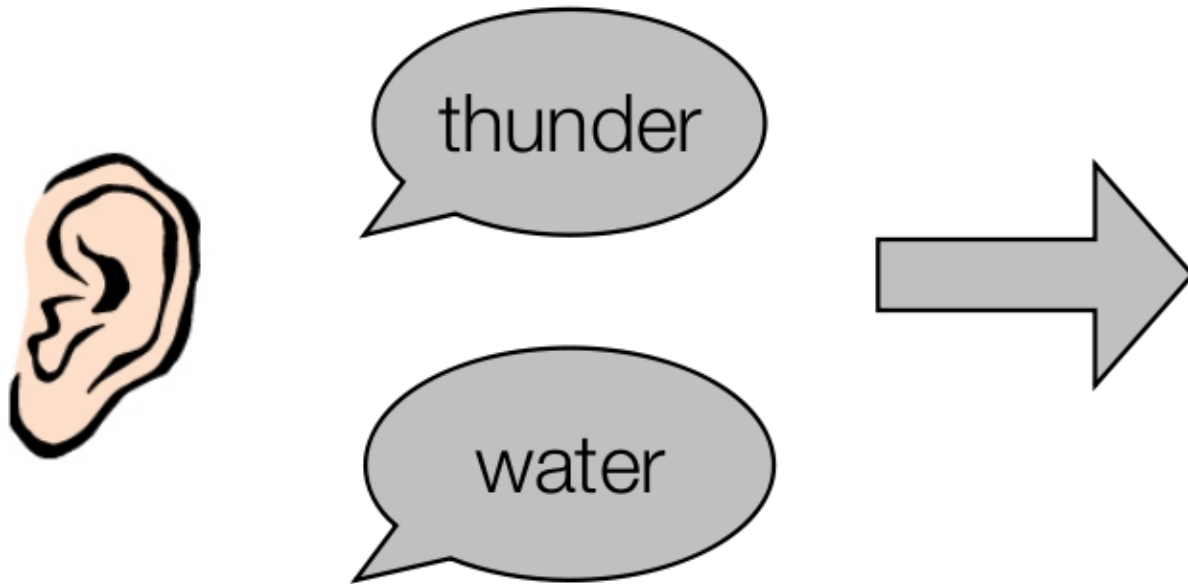
- We want to take all of our data (facts) and combine it with our knowledge (rules) and come up with some conclusions (inferencing)



JBoss BRMS

What is inferencing?

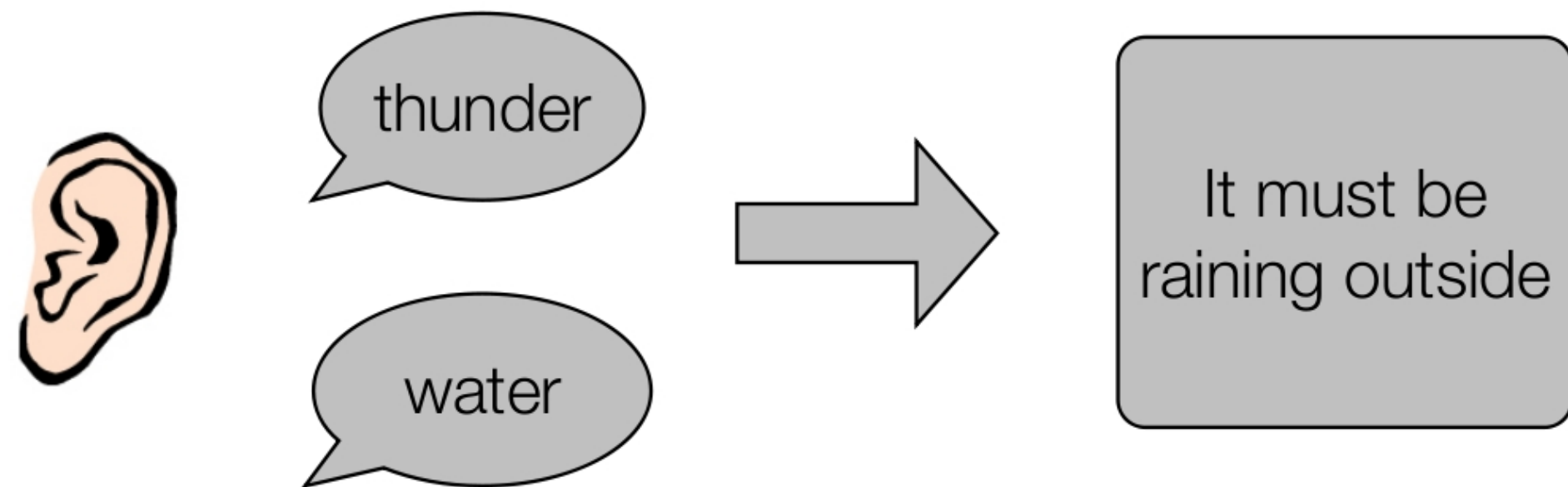
- We want to take all of our data (facts) and combine it with our knowledge (rules) and come up with some conclusions (inferencing)



JBoss BRMS

What is inferencing?

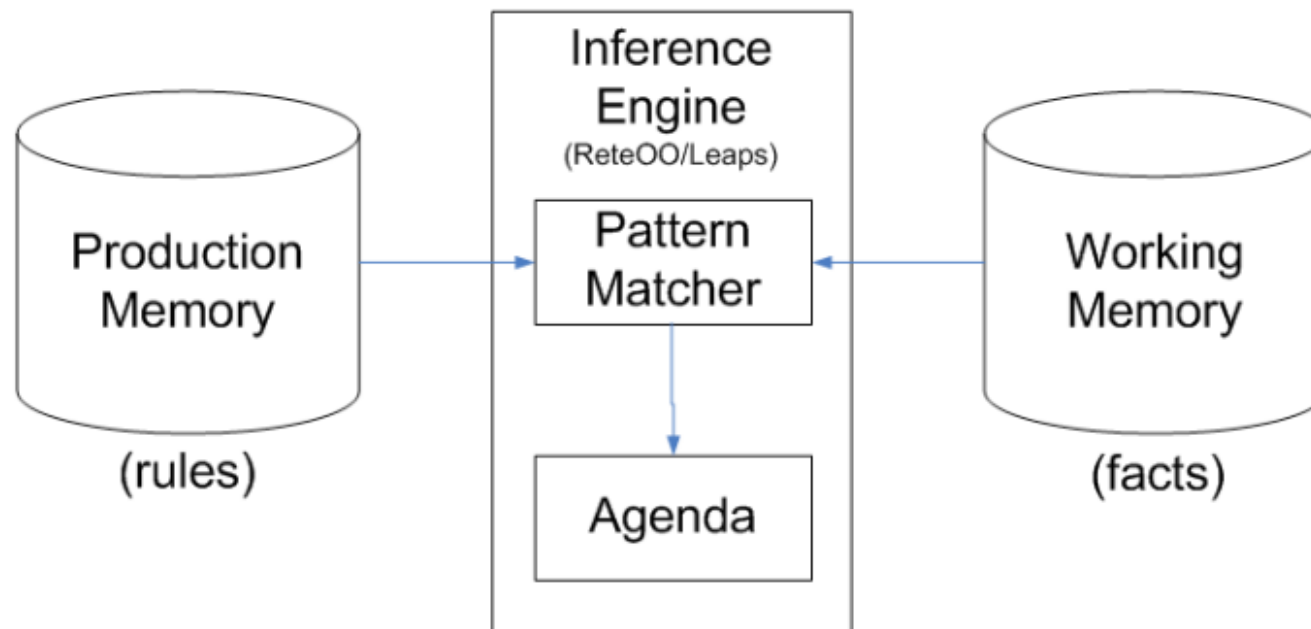
- We want to take all of our data (facts) and combine it with our knowledge (rules) and come up with some conclusions (inferencing)



JBoss BRMS

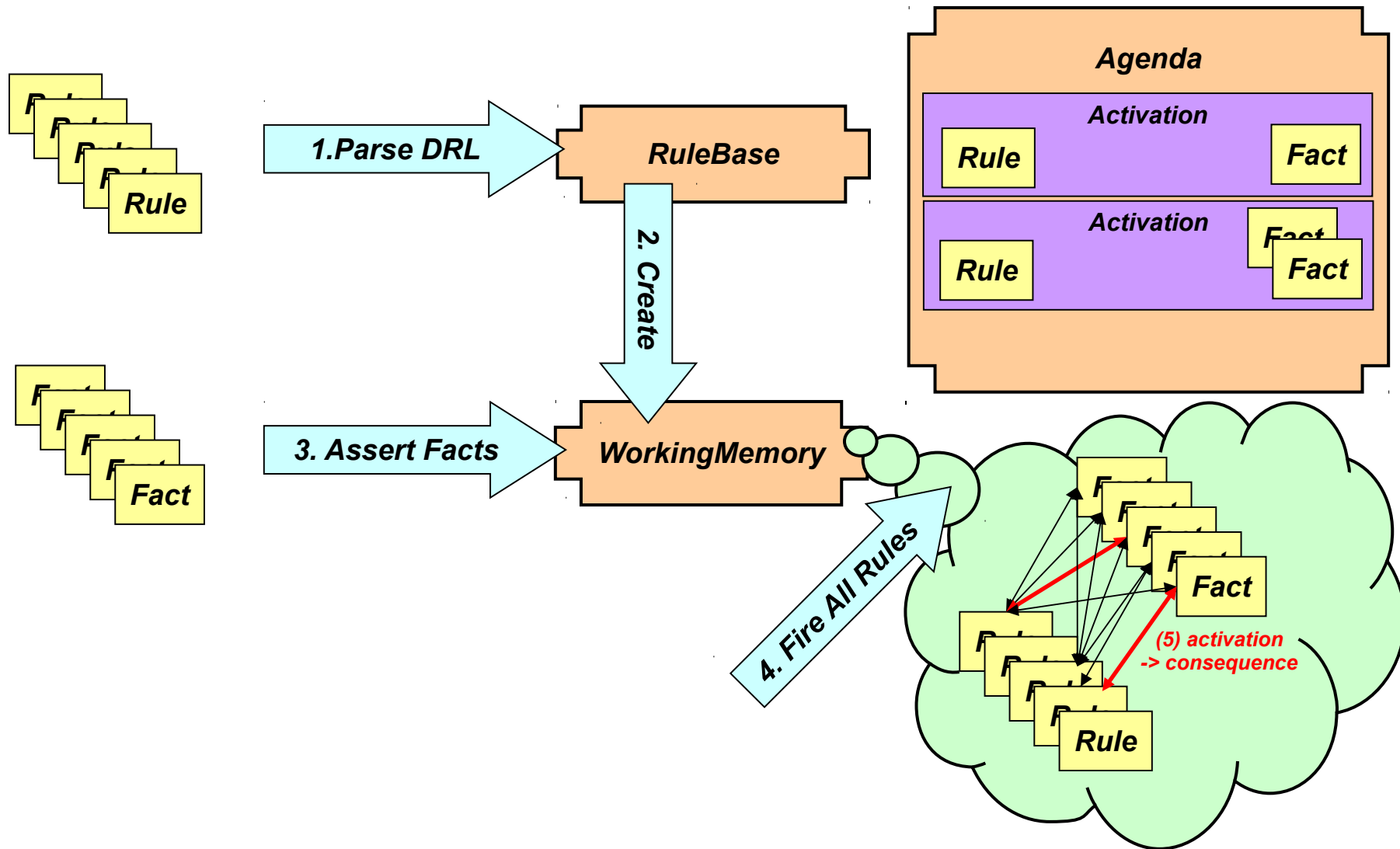
Inference Engine

- The brain of a Production Rule System is an Inference Engine which matches facts against Production Rules.
- When matches are found, the rules actions are fired.
- Actions most often change the state of the facts, or perform some external action on the application.

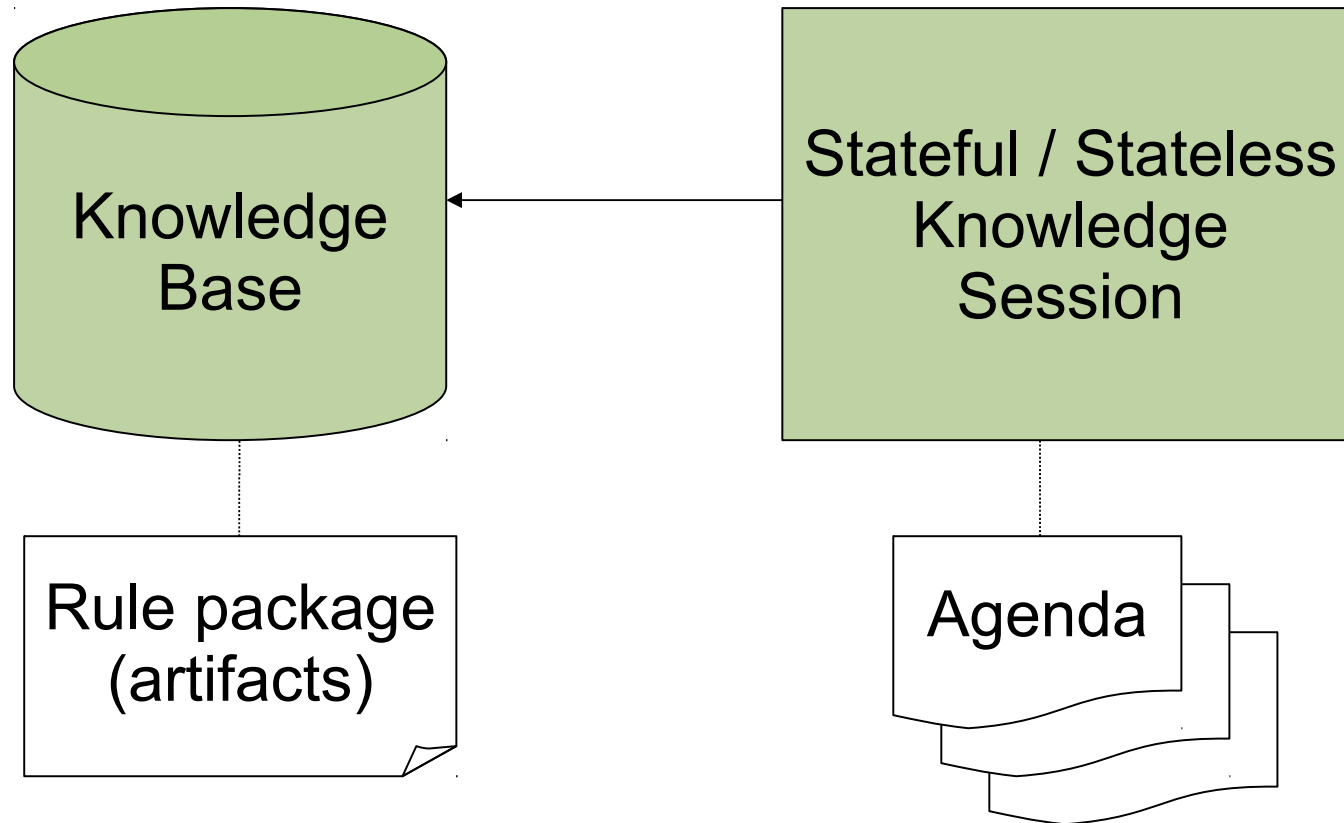


JBoss BRMS

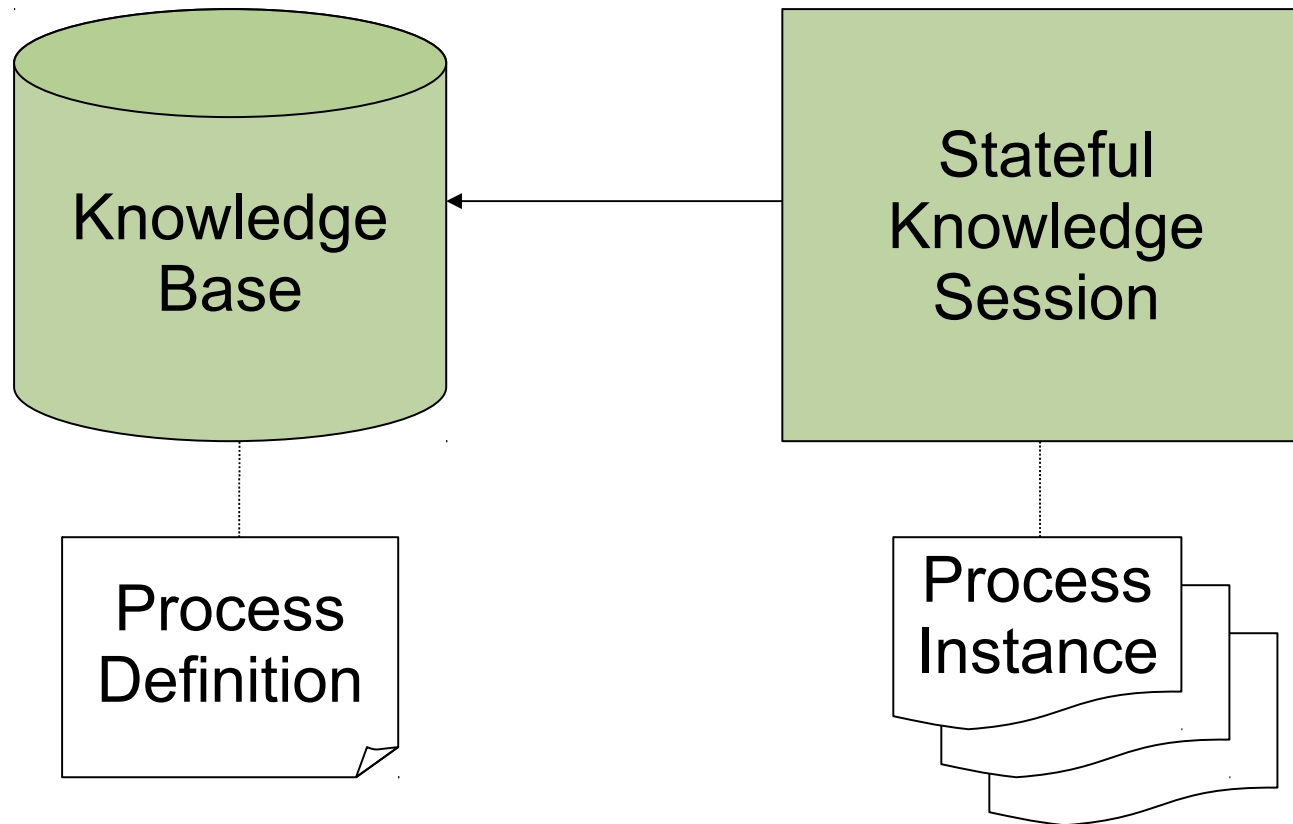
How it Works



Conceptual example



Example: real BPM



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");
```

BRMS – Rich UI for Business Rules

The screenshot displays the JBoss BRMS (Business Rules Management System) interface, which is a rich UI for managing business rules. The interface is divided into several sections:

- Left Panel (Navigation):** Contains a tree view of the project structure, including Packages, Assets, Business rule assets, Technical rule assets, Functions, DSL configurations, Model, Rule Flows, Enumerations, Test Scenarios, XML, Properties, Other assets, documentation, mortgages, and order_processing.
- Top Panel (Model Editor):** Shows the 'Simple Insurance Model' with fields for Driver (locationRiskProfile, eyeColor, age, priorClaims, name) and Policy (basePrice, approved, type, discountPercent).
- Right Panel (Test Results):** Displays the results of a test scenario. It shows a summary of the test, including the number of rules fired (2 rules fired in 0ms) and the overall result (FAILURE). It also lists the test scenarios and their results, such as 'Adv scoring sanity test' (1 failure out of 23 expectations) and 'Sydney advertiser ratings' (10 failures out of 3).

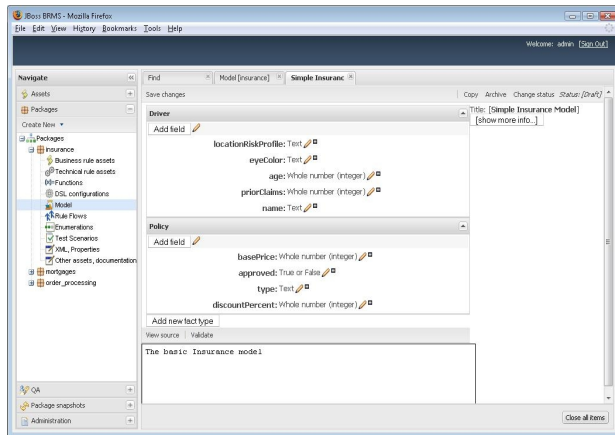
The main area of the interface shows the rule editor, where users can define rules using a visual editor. The rule is defined as follows:

```
THEN
  Set value of policy basePrice 455
  (options)
```

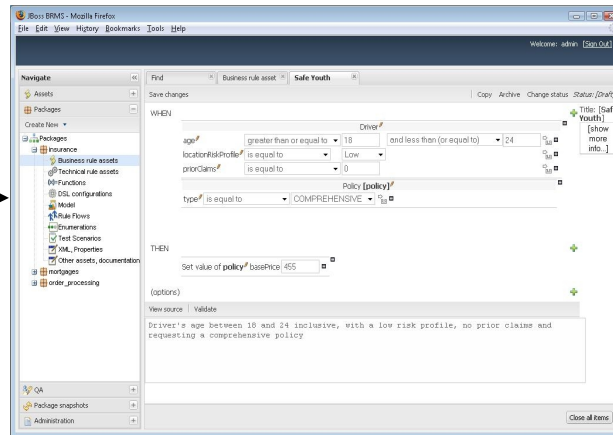
The rule is titled 'Policy [policy]' and is set to 'COMPREHENSIVE'. The rule is triggered when the driver's age is greater than or equal to 18 and less than (or equal to) 24, the location risk profile is equal to 'Low', and the prior claims are equal to 0.

Analysis Perspective

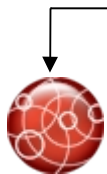
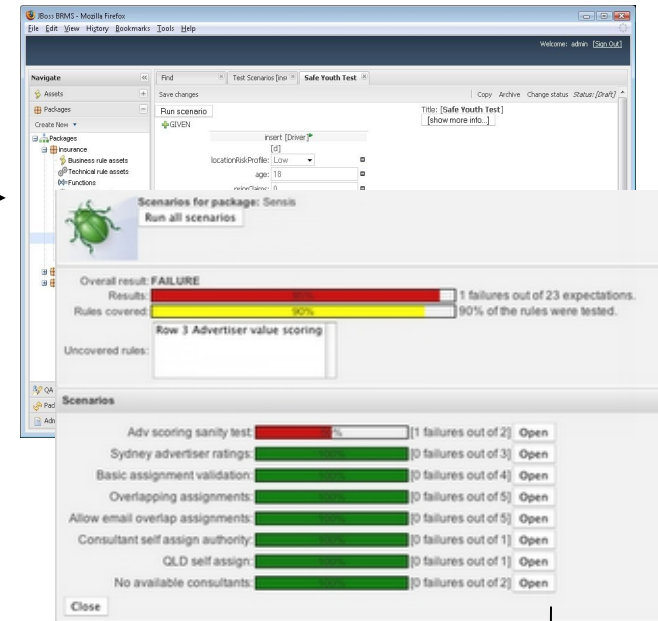
Test the Rules



JBoss BRMS Web Manager
Model the Facts

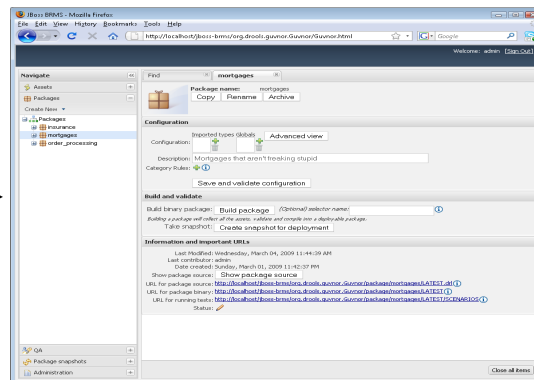


Prototype the Rules



JBoss Developer Studio

Check-Out Artifacts from Repository
Build & Test Application
Check-In changes to Repository



JBoss BRMS Web Manager
Create Package

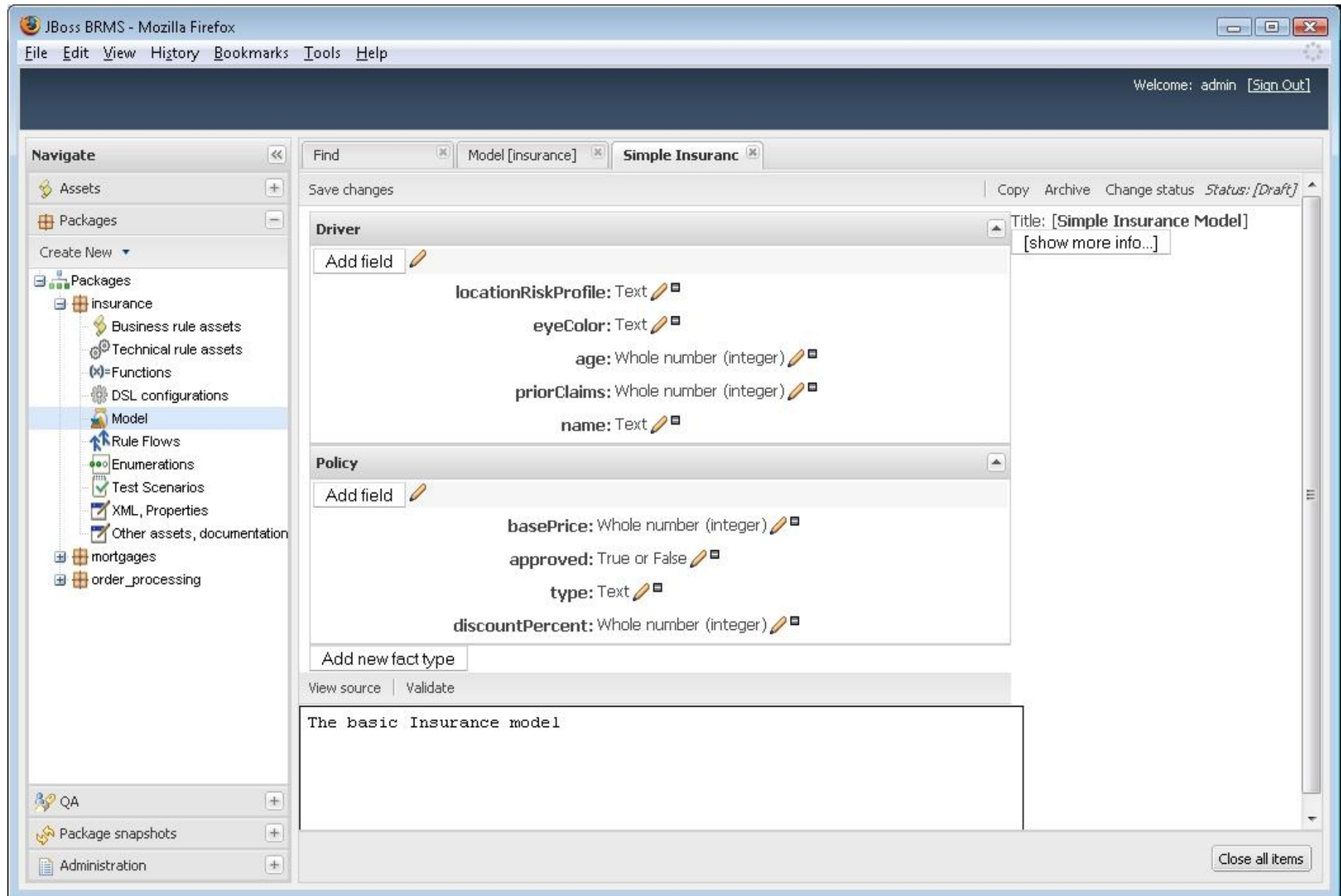
Deployment

J2SE

EAP-WS,WAR,EAR

SOA-P, Mediated Services

Fact Model Editor



Guided Rule Editor (Web)

The screenshot shows the JBoss BRMS Web Guided Rule Editor interface. The browser window title is "JBoss BRMS - Mozilla Firefox". The menu bar includes File, Edit, View, History, Bookmarks, Tools, and Help. The top right corner displays "Welcome: admin [Sign Out]".

Navigate Panel: Located on the left, it contains a tree view with the following structure:

- Assets (+)
- Packages (-)
 - Create New ▾
 - Packages
 - insurance
 - Business rule assets (selected)
 - Technical rule assets
 - Functions
 - DSL configurations
 - Model
 - Rule Flows
 - Enumerations
 - Test Scenarios
 - XML, Properties
 - Other assets, documentation
 - mortgages
 - order_processing
- QA (+)
- Package snapshots (+)
- Administration (+)

Main Editor Area:

- Find:** Business rule asset | Safe Youth
- Save changes:** Copy | Archive | Change status | Status: [Draft]
- WHEN:**
 - Driver:**
 - age: greater than or equal to 18 and less than (or equal to) 24
 - locationRiskProfile: is equal to Low
 - priorClaims: is equal to 0
 - Policy [policy]:**
 - type: is equal to COMPREHENSIVE
- THEN:**
 - Set value of policy basePrice 455
- (options):**
- View source | Validate**
- Text Area:** Driver's age between 18 and 24 inclusive, with a low risk profile, no prior claims and requesting a comprehensive policy

Right Panel: Title: [Safe Youth] [show more info...]

Bottom Right: Close all items

Guided Rule Editor (Web)

The screenshot shows the JBoss BRMS Web Guided Rule Editor interface. The browser window title is "JBoss BRMS - Mozilla Firefox". The menu bar includes File, Edit, View, History, Bookmarks, Tools, and Help. The top right corner displays "Welcome: admin [Sign Out]".

Navigate Panel: Located on the left, it contains a tree view with the following structure:

- Assets (+)
- Packages (-)
 - Create New ▾
 - Packages
 - insurance
 - Business rule assets (selected)
 - Technical rule assets
 - Functions
 - DSL configurations
 - Model
 - Rule Flows
 - Enumerations
 - Test Scenarios
 - XML, Properties
 - Other assets, documentation
 - mortgages
 - order_processing
- QA (+)
- Package snapshots (+)
- Administration (+)

Main Editor Area:

- Find:** Business rule asset | Safe Youth
- Save changes:** Copy | Archive | Change status | Status: [Draft]
- WHEN:**
 - Driver:**
 - age: greater than or equal to 18 and less than (or equal to) 24
 - locationRiskProfile: is equal to Low
 - priorClaims: is equal to 0
 - Policy [policy]:**
 - type: is equal to COMPREHENSIVE
- THEN:**
 - Set value of policy basePrice 455
- (options):**
- View source | Validate**
- Text Area:** Driver's age between 18 and 24 inclusive, with a low risk profile, no prior claims and requesting a comprehensive policy

Right Panel: Title: [Safe Youth] [show more info...]

Bottom Right: Close all items

Web Decision Table Editor

JBoss BRMS - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Welcome: admin [Sign Out]

Navigate

- Assets
- Packages
- Create New
- Packages
 - insurance
 - Business rule assets
 - Technical rule assets
 - Functions
 - DSL configurations
 - Model
 - Rule Flows
 - Enumerations
 - Test Scenarios
 - XML, Properties
 - Other assets, documentation
 - mortgages
 - Business rule assets
 - Technical rule assets
 - Functions
 - DSL configurations
 - Model
 - Rule Flows
 - Enumerations
 - Test Scenarios
 - XML, Properties
- QA
- Package snapshots
- Administration

Find Pricing loans

Save changes

Decision table

Modify...

	Des	amount min	amount max	period	income	deposit max	Loan approved	LMI
income: Asset (1 Item)								
3		131000	200000	30	Asset	20000	true	0
income: Job (2 Items)								
1		10000	100000	20	Job	2000	true	0
2		100001	130000	20	Job	3000	true	10

Close all items

Test Scenario Editor

The screenshot shows the JBoss BRMS Test Scenario Editor interface. The browser window title is "JBoss BRMS - Mozilla Firefox". The menu bar includes File, Edit, View, History, Bookmarks, Tools, and Help. The top right corner displays "Welcome: admin [Sign Out]".

Navigate sidebar (left):

- Assets (+)
- Packages (-)
- Create New ▾
- Packages
 - insurance
 - Business rule assets
 - Technical rule assets
 - Functions
 - DSL configurations
 - Model
 - Rule Flows
 - Enumerations
 - Test Scenarios**
 - XML, Properties
 - Other assets, documentation
 - mortgages
 - order_processing
- QA (+)
- Package snapshots (+)
- Administration (+)

Find tabs: Test Scenarios [ins], **Safe Youth Test**

Safe Youth Test details:

- Save changes | Copy | Archive | Change status | Status: [Draft]
- Run scenario
- Title: [Safe Youth Test] [show more info...]
- + GIVEN**
 - insert [Driver] [d]
 - locationRiskProfile: Low
 - age: 18
 - priorClaims: 0
 - name: Joe Blogs
 - insert [Policy] [p]
 - type: COMPREHENSIVE
- + EXPECT**
 - Use real date and time
 - Expect rules
 - Safe Youth: fired at least once
 - Policy [p] has values:
 - basePrice: equals 450
 - More... (configuration) All rules may fire
 - (globals)
- Verifies that a Safe Youth is correctly priced

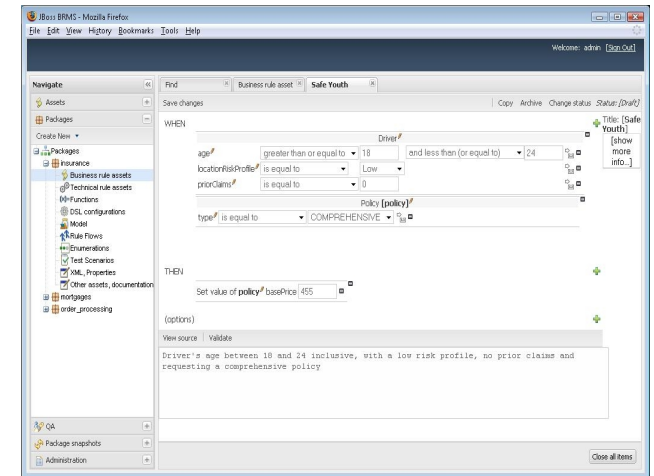
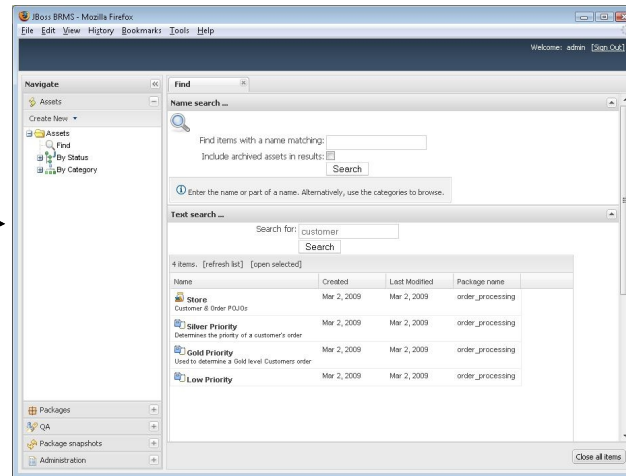
Close all items

Developer Perspective

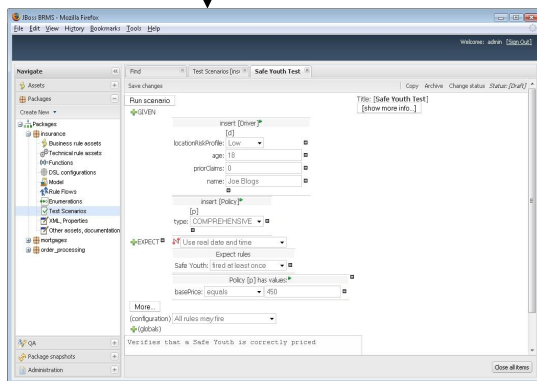


JBoss Developer Studio
Fact Model
DSL Template

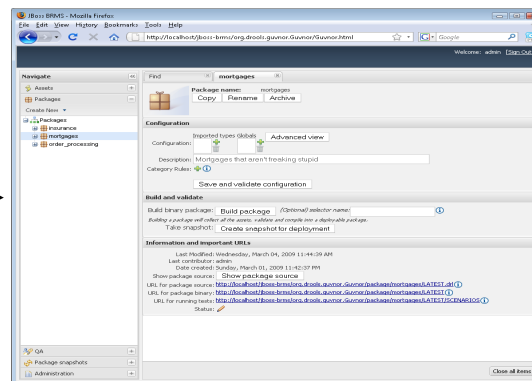
Check-In Artifacts to
Repository



JBoss BRMS Web
Manager
Use Facts & DSL
to create Business Rules



JBoss BRMS Web
Manager
Create Test Scenarios



JBoss BRMS Web
Manager
Create Package

Deployment

J2SE

EAP-
WS,WAR,EAR

SOA-P,
Mediated
Services

Eclipse – Developer Perspective

The screenshot displays the Eclipse IDE in the Developer Perspective for a Drools application. The main components visible are:

- Text Editor:** Shows the `StateExampleUsingSaliency.drl` file with the following code:

```
import org.drools.examples.State;

rule Bootstrap
when
  a : State(name == "A", state == State.NOTRUN)
then
  System.out.println(a.getName() + " finished");
  a.setState(State.FINISHED);
end

rule "A to B"
when
  State(name == "A", state == State.FINISHED)
  b : State(name == "B", state == State.NOTRUN)
then
  b.setState(State.FINISHED);
  System.out.println(b.getName() + " finished");
end

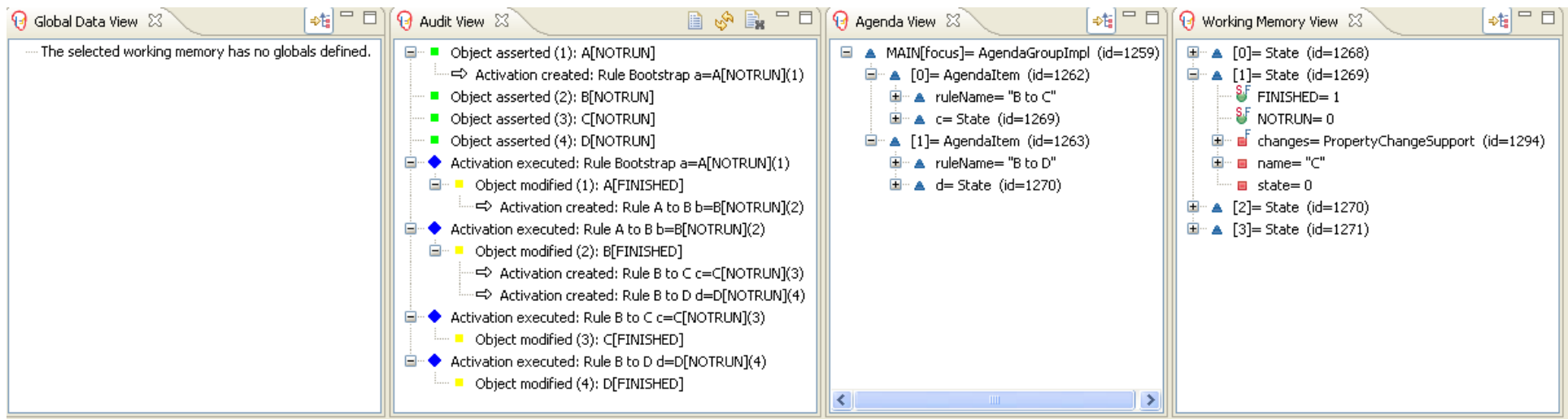
rule "B to C"
salience 10
when
  State(name == "B", state == State.FINISHED)
  c : State(name == "C", state == State.NOTRUN)
then
  System.out.println(c.getName() + " finished");
end
```
- Rete Tree:** A visual representation of the rule network, showing nodes for states A, B, and C, and their transitions.
- Variables:** A table showing the current state of variables:

Name	Value
b	State (id=1268)
changes	PropertyChangeSupport (id=1297)
name	"B"
state	1
- Properties:** A table showing the properties of the selected object (State):

Property	Value
General	
Constraint	[LiteralConstraint fieldExtra]
Evaluator	Integer ==
Field Name	state
Name	Alpha BaseVertex
Value	1
- Outline:** A tree view showing the project structure, including the `org.drools.examples` package and its contents.
- Global Data View:** Shows the selected working memory has no globals defined.
- Audit View:** A log of events, including object assertions, modifications, and rule activations. For example, it shows the activation of the `Bootstrap` rule and the subsequent activation of the `"A to B"` rule.
- Agenda View:** Shows the current state of the agenda, including the `MAIN` focus and the `"B to C"` rule.
- Working Memory View:** Shows the current state of the working memory, including the `State` objects and their properties.

Debugging

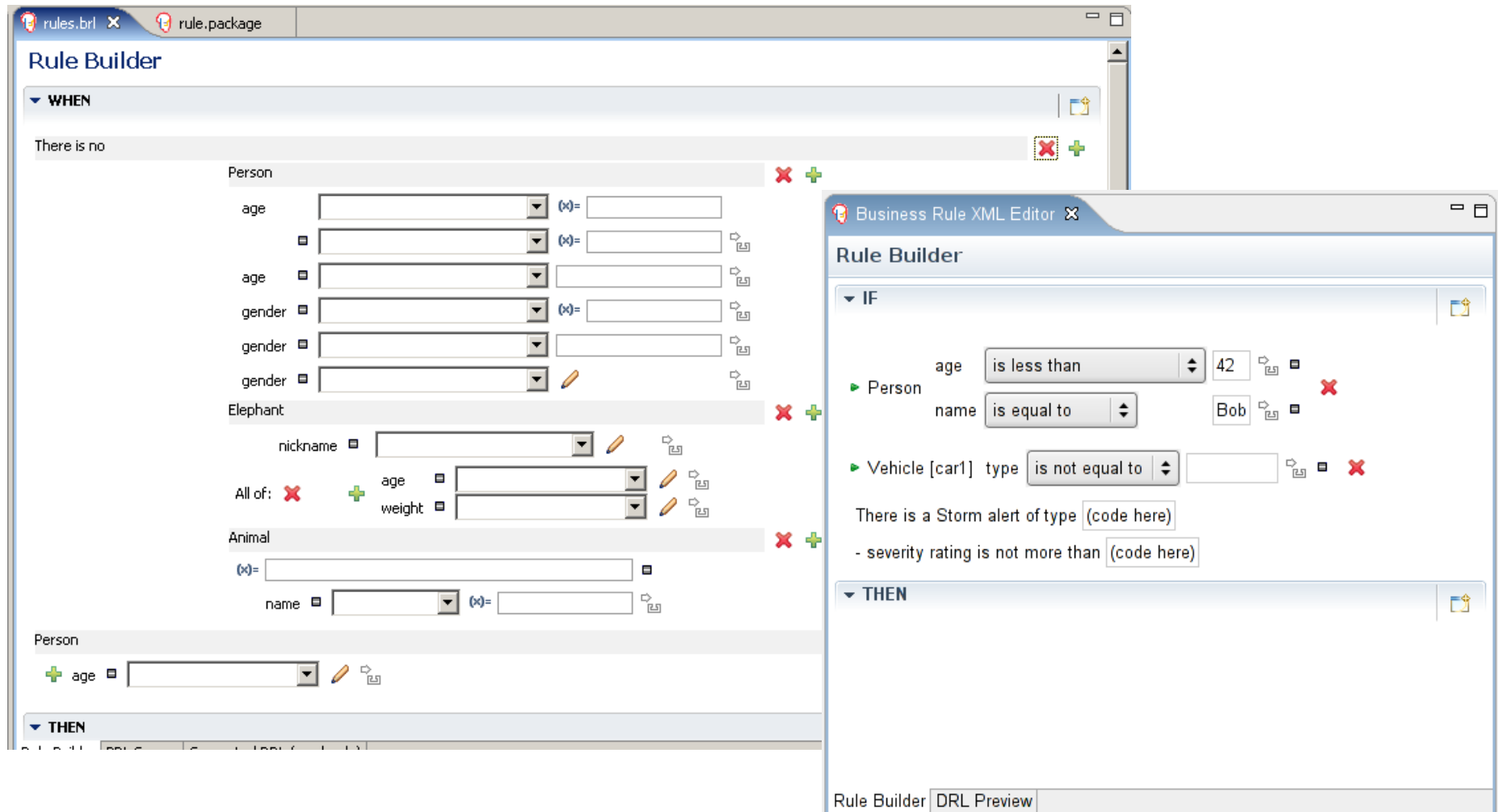
Debug Views



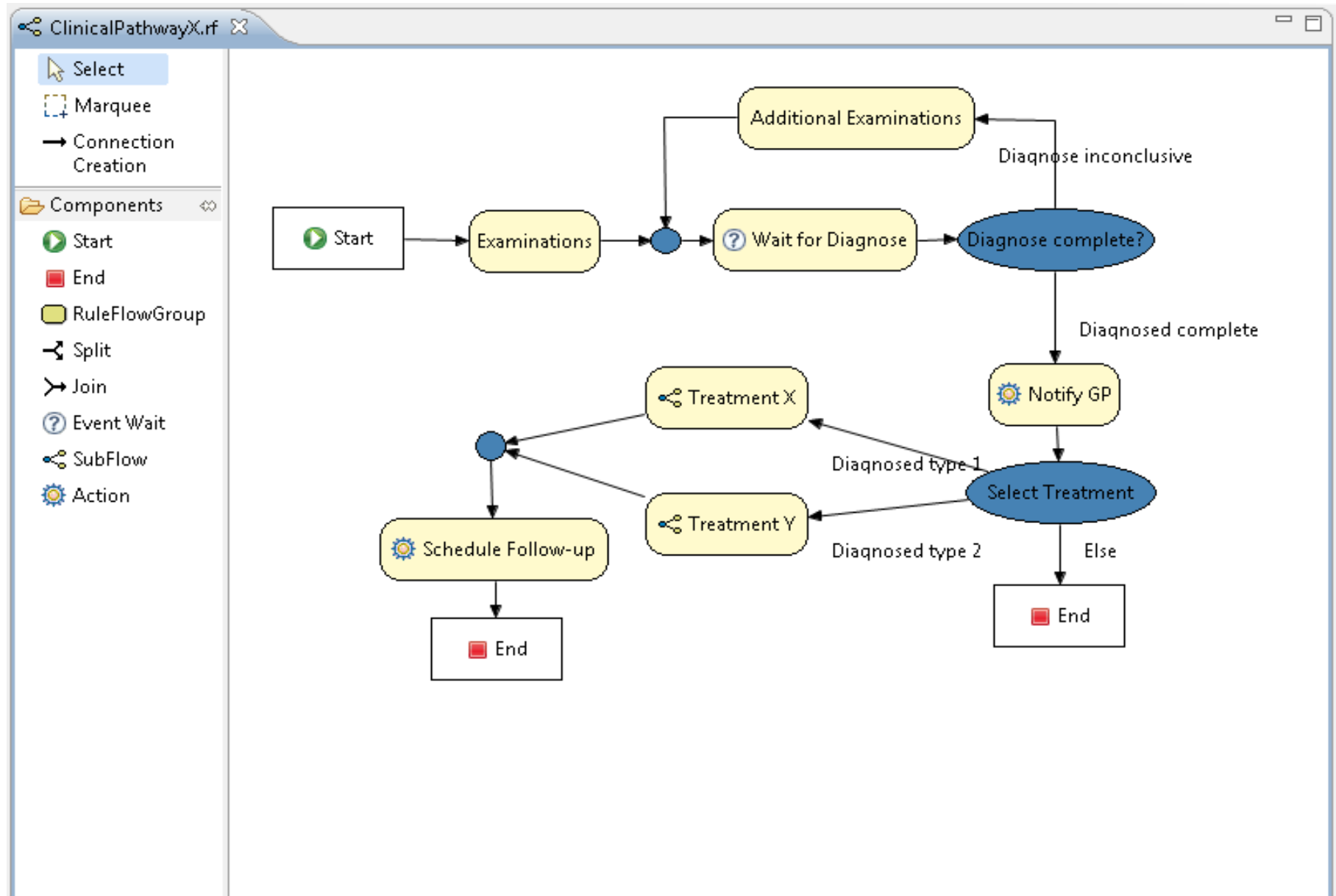
- New rule “perspective” to configure IDE for rules as needed

Guided Rule Editor (Eclipse)

Eclipse Guided Editor



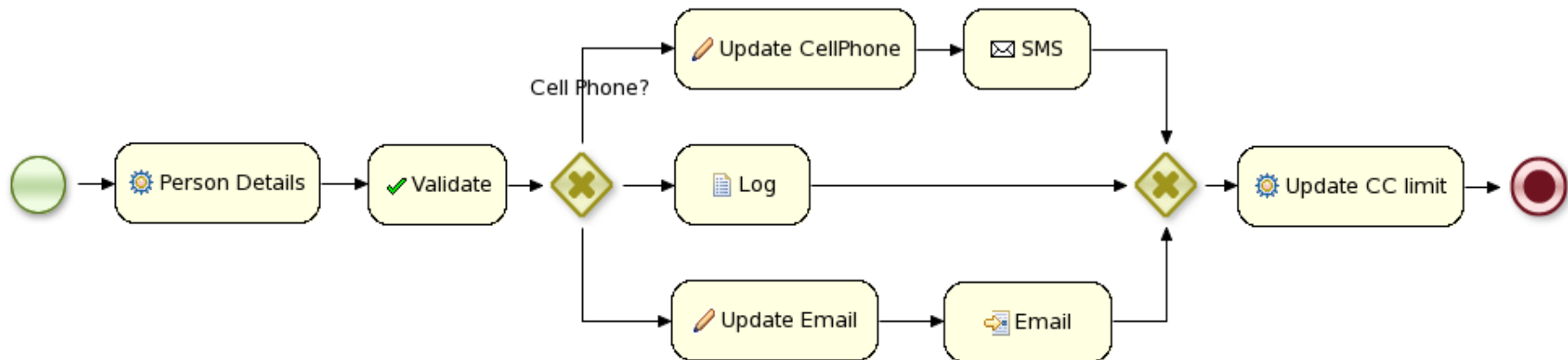
Rule Flow



JBoss BPM

Core
Engine

- Core engine is a workflow engine in pure Java
 - state transitions
 - lightweight
 - embeddable



Decision Tables

Decision Tables – Excel/Open Office

	B	C	D	E	F	G	H
1							
4							
9	Base pricing rules	Age Bracket	Location risk profile	Number of prior claims	Policy type applying for	Base \$ AUD	Record Reason
10	Young safe package	18, 24	LOW	1	COMPREHENSIVE	450	
11			MED		FIRE_THEFT	200	Priors not relevant
12			MED	0	COMPREHENSIVE	300	
13			LOW		FIRE_THEFT	150	
14			LOW	0	COMPREHENSIVE	150	Safe driver discount
15	Young risk	18,24	MED	1	COMPREHENSIVE	700	
16		18,24	HIGH	0	COMPREHENSIVE	700	Location risk
17		18,24	HIGH		FIRE_THEFT	550	Location risk
18		25,30		0	COMPREHENSIVE	120	Cheapest possible
19	Mature drivers	25,30		1	COMPREHENSIVE	300	
20		25,30		2	COMPREHENSIVE	590	
21		25,35		3	THIRD PARTY	800	High risk

Eclipse to BRMS Synchronization

JBoss Developer Studio

BRMS 5

Eclipse IDE (Left): Guvnor Repository Exploring - String-based file: MortgageModel.model.drl (Read only) - JBoss Developer Studio

BRMS 5 Web Interface (Right): JBoss Guvnor - Windows Internet Explorer

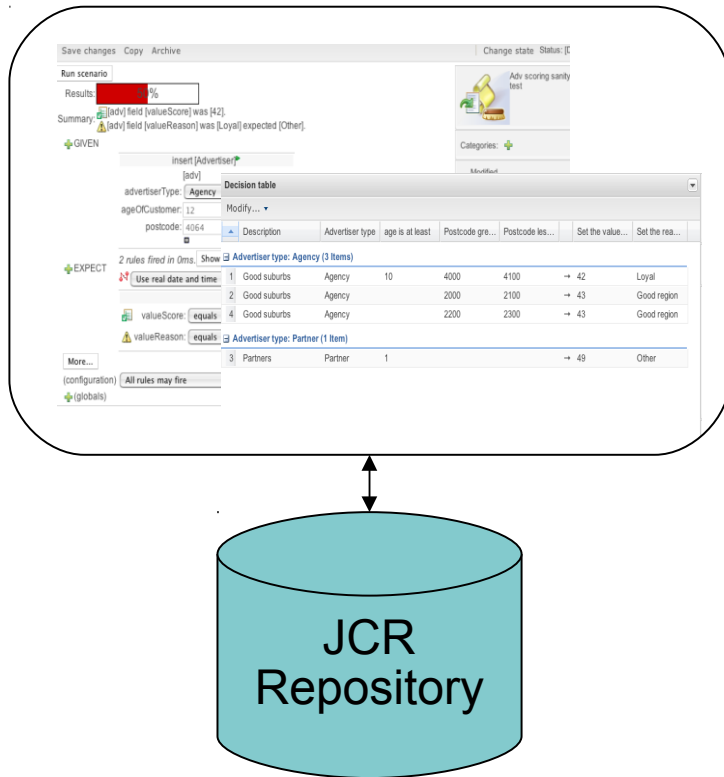
Guvnor Resource History:

Revision	Date	Author	Comment
3	2008-07-15T15:37:34	john	<from webide>
2	2008-07-15T15:32:03	john	
1	2008-07-15T15:28:35	john	<from webide>

JCR Repository: A cylinder icon labeled "JCR Repository" connected to the BRMS 5 interface.

BRMS/Rules Deployment Scenario

BRMS



RuleAgent

J2SE

JBoss Rules Runtime

RuleAgent

EAP 4.3

J2EE use cases
Seam use cases
Clustered

RuleAgent

SOA-P 5.0 (future)

Stateless Rule Services
Stateful Rule Services
CBR & CBF
Federated Hot Deployment

Delivery of proper version of a rule package via the ruleagent to any of the above “containers”.

EE and SOA containers allow for hot deployment/reload of the changed rules without component restart and no loss of transactions/messages.

BRMS v5: Summary of Features

Graphical User Interface and repository for Business Rule Editing, Versioning & Deployment management that is usable by the non-Java Programmer and non-System Administrator.

Manages the following artifacts:

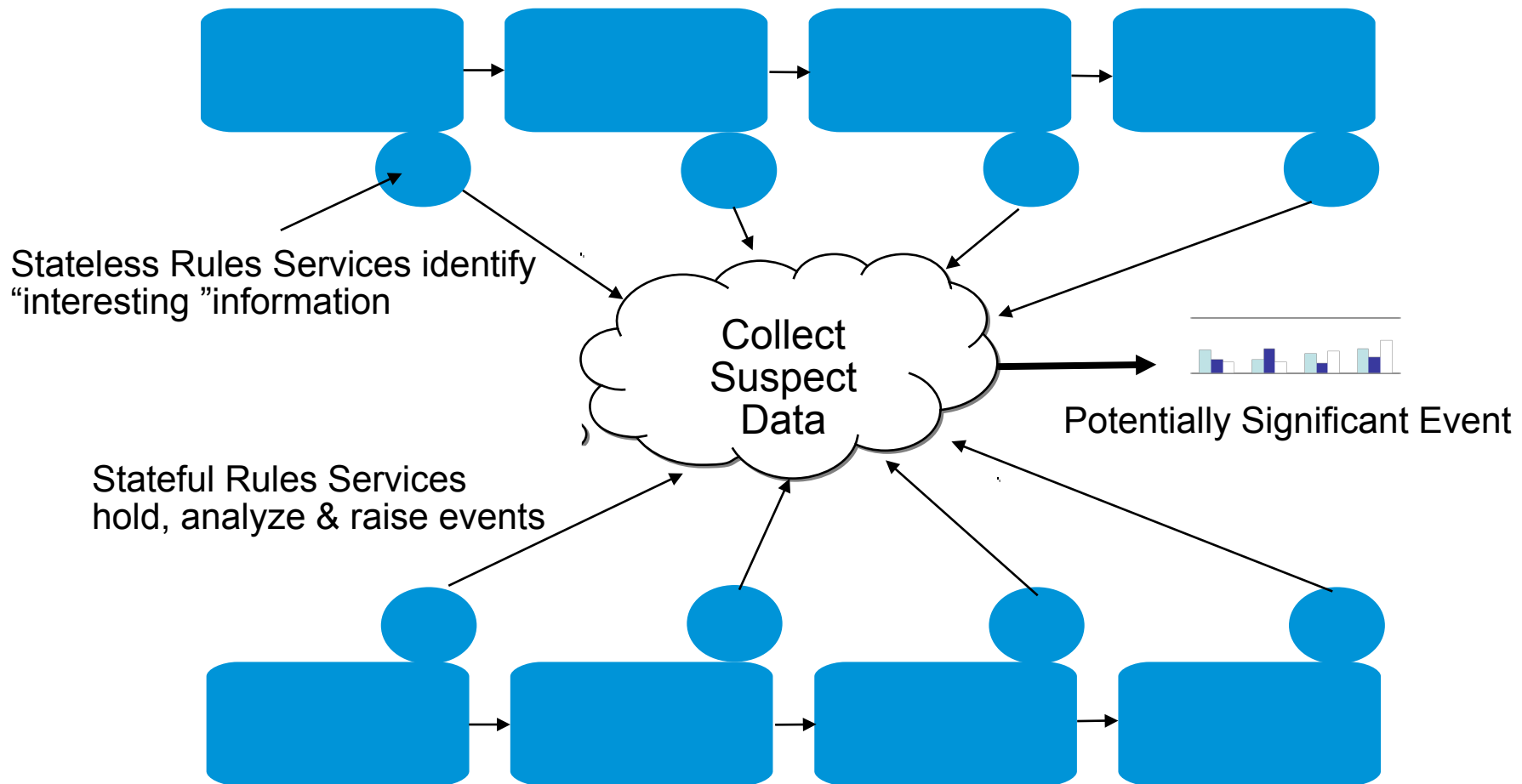
- Rule Packages
- Business Rules (via Guided Rule Editor)
- Business Fact Model
- Technical Rules (sync with Eclipse and/or BRMS UI)
- Technical POJO Fact Model
- Domain Specific Languages – DSL
- Web-based Decision Tables
- Spreadsheet-based Decision Tables
- Rule Categories (for searching)
- Users & Roles associated with rule life cycle states
- Rule Promotion (Dev, QA, Production)
- Test Scenarios

JBoss BRMS, the future is now



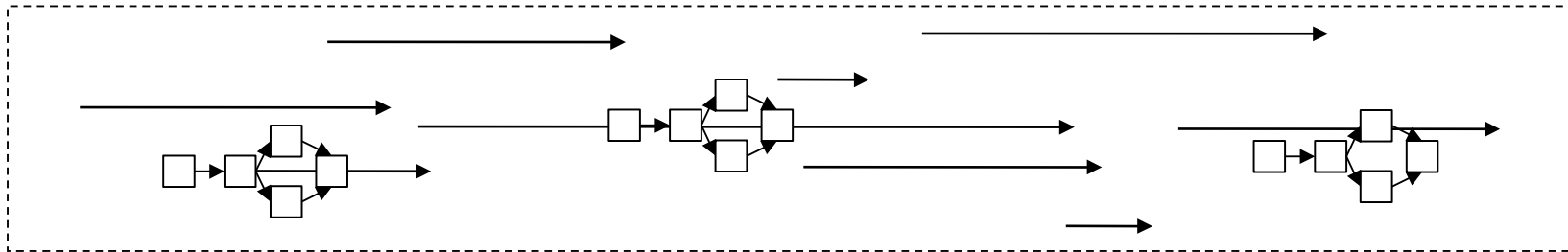
Future: Event Stream Processing

Some Process: Order Fulfillment, Replenishment, Customer Service, Financial Analysis



Other Process also capturing interesting data points and routing them to the "stateful" rules service for keeping and analysis.

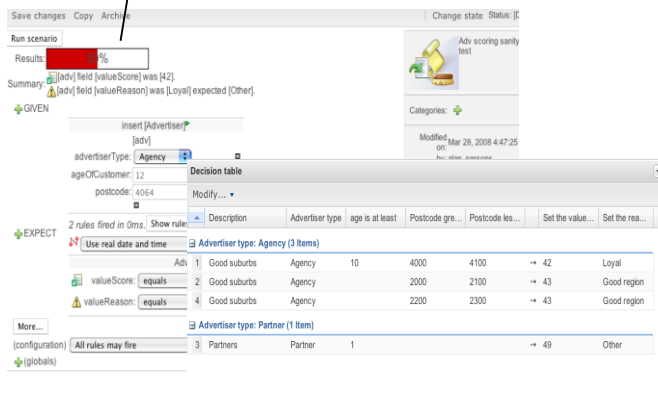
Features: ESP + CEP via ESB, Rules & JON



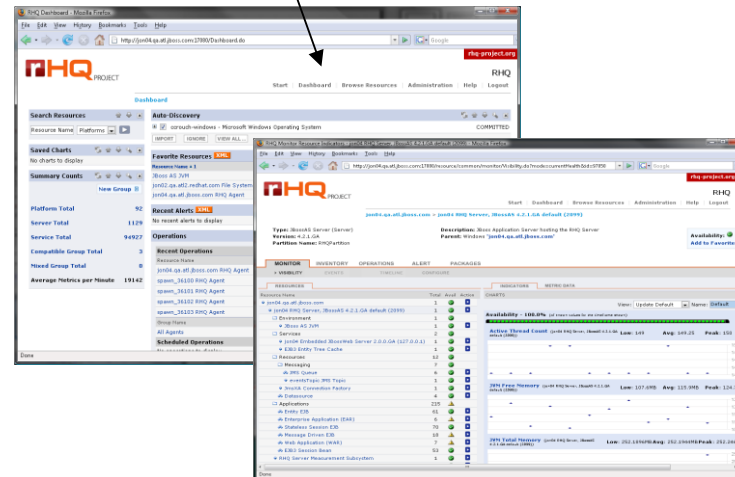
ESB: consumption, capture, transformation, routing, orchestration
Rules: selection, aggregation, correlation, generation and publication



Governance
Tools



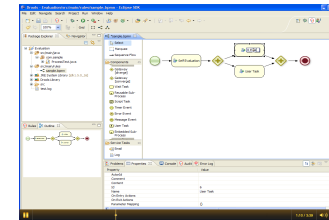
Repository: for editing, versioning, testing and publishing rules and SOA artifacts



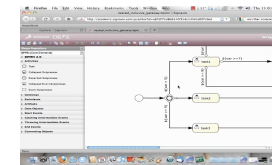
JON: start/stop services, monitor and alert on service and action-level performance, monitor and alert on business metric values

BPMS future at JBoss

Pluggable
&
Standards



Eclipse
BPMN2
Editor

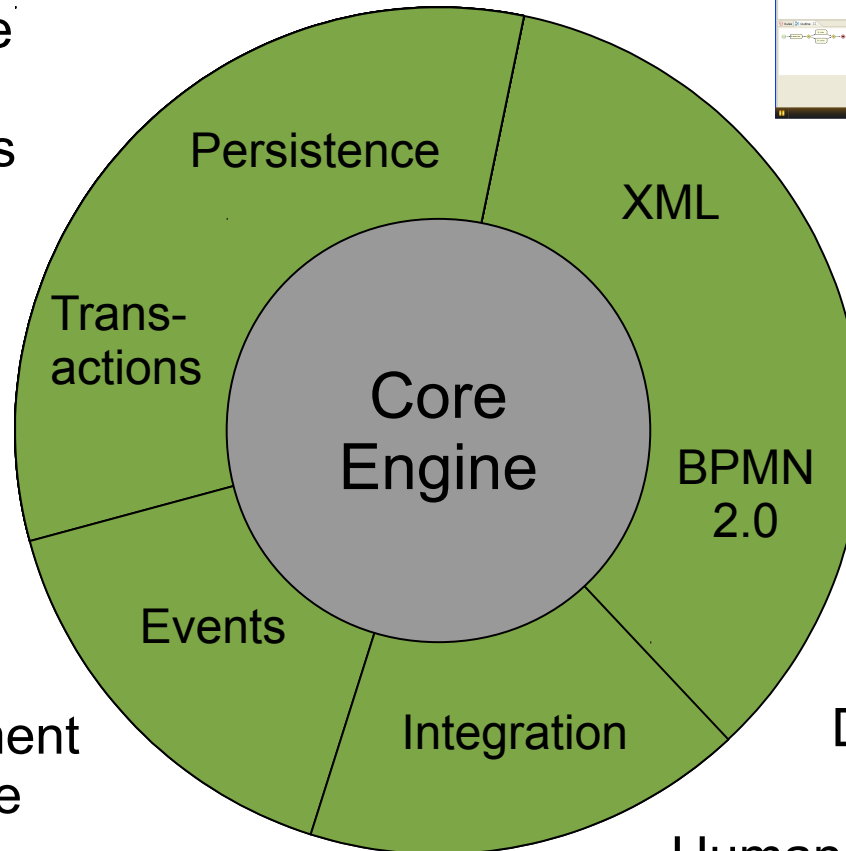


Oryx Web-
Based Editor

History
Log

INSTANCEID	STATUS	DATE	DATE
1	on-heap	2009-09-09	2009-09-09
2	on-heap	2009-09-09	2009-09-09
3	on-heap	2009-09-09	2009-09-09

1	2	3	4	5	6	7	8	9	10
1	new	0	3	1	on-heap	2009-09-09	2009-09-09	2009-09-09	2009-09-09
2	new	3	2	2	on-heap	2009-09-09	2009-09-09	2009-09-09	2009-09-09
3	new	0	3	2	on-heap	2009-09-09	2009-09-09	2009-09-09	2009-09-09



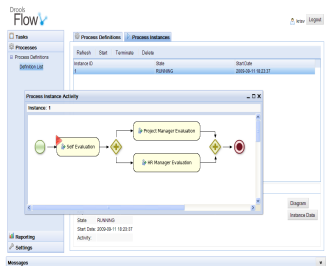
Domain-specific
Processes

Human Task
Service

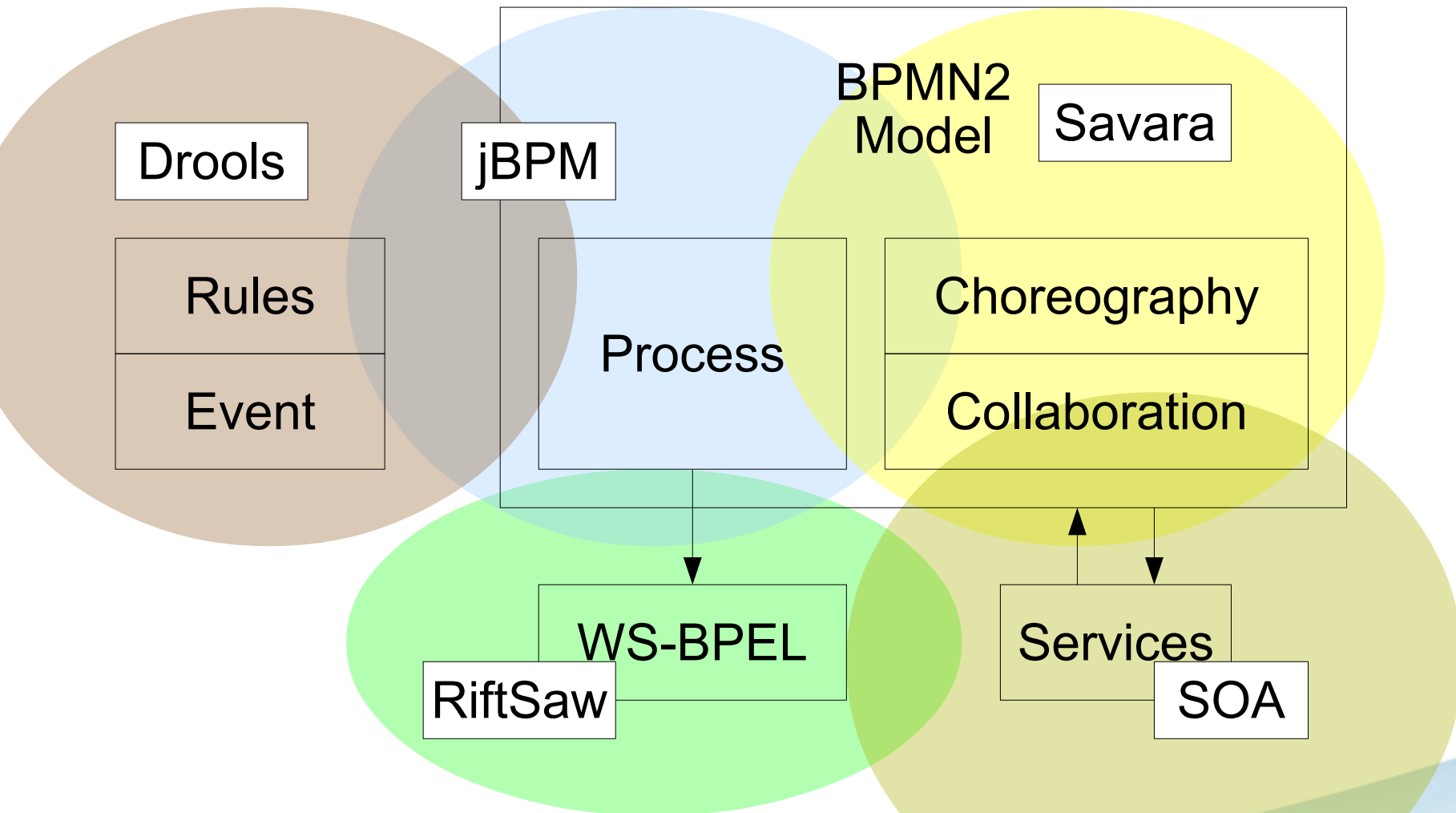
ESB

BAM

Management
Console



BPMN2 and Projects



Agenda

- Introduction
- JBoss BRMS
 - Overview
 - Future directions
- *Demo*
- *Questions*