Business Logic Integration Platform

D.Sottara, PhD
OMG Technical Meeting – Spring 2013, Reston, VA

Outline

- Part I – The Consolidated Past : Drools 5.x
  - **Drools Expert**
    - Object-Oriented, Production Rule engine
  - **Drools Flow**
    - Complex Event Processing engine
  - **JBPM**
    - BPMNv2-compliant process engine
  - **Decision Tables**
    - Waiting for DMN
  - **Drools Planner (new! Rebranded and released as OptaPlanner)**
    - Metaheuristics Constraint Optimization
  - **Guvnor**
    - Repository and Authoring BRMS

Outline

- Part II – The Upcoming Future : Drools 6.x
  - **KIE**
    - Knowledge Integration and Execution Environment
  - **Drools – PMML**
    - Predictive Analytics
  - **Drools Chance**
    - Fuzzy and other “non-boolean” reasoning styles
  - **Drools Shapes**
    - Semantic Web Technology integration
  - **Drools MAS**
    - FIPA-compliant agent implementation

About me

- Davide Sottara, PhD
  - Department of Biomedical Informatics

    Arizona State University, Scottsdale (AZ)
  - Background :
    - AI
    - Hybrid Systems
    - Decision Support Systems
  - Drools community member, developer and contributor since 2006

- Community Member - not a JBoss employee
  - *The opinions presented here are personal and may not reflect the actual intentions of the Drools development team, or JBoss, inc. their employer*
  - ***Thanks to the members of the Drools team for providing part of the material presented!***

Resources

- Main Web Sites
  - www.jboss.org/drools
  - http://www.jboss.org/jbpm/
  - http://www.optaplanner.org/
- Documentation & Demos
  - www.jboss.org/drools/documentation
- More introductory material
  - http://www.jboss.org/drools/presentations.html
- License
  - Open source, ASL 2.0
- Source code
  - www.github.com/droolsjbpm

Drools Expert

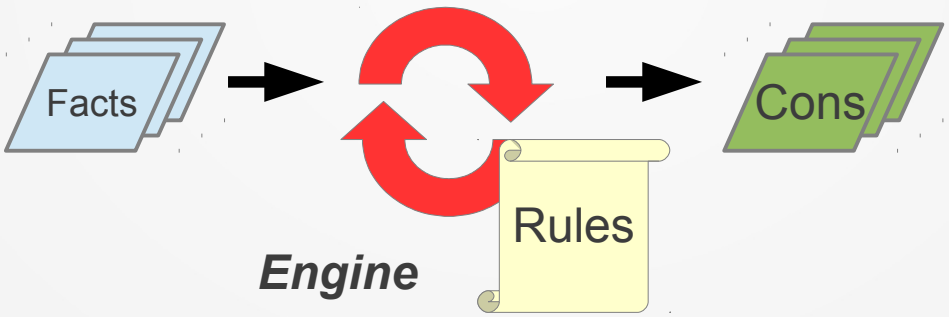
Drools Expert

- Heir of the “golden age” of RETE Production Rule Systems:
  - OPSx
  - CLIPS
  - Jess
  - …

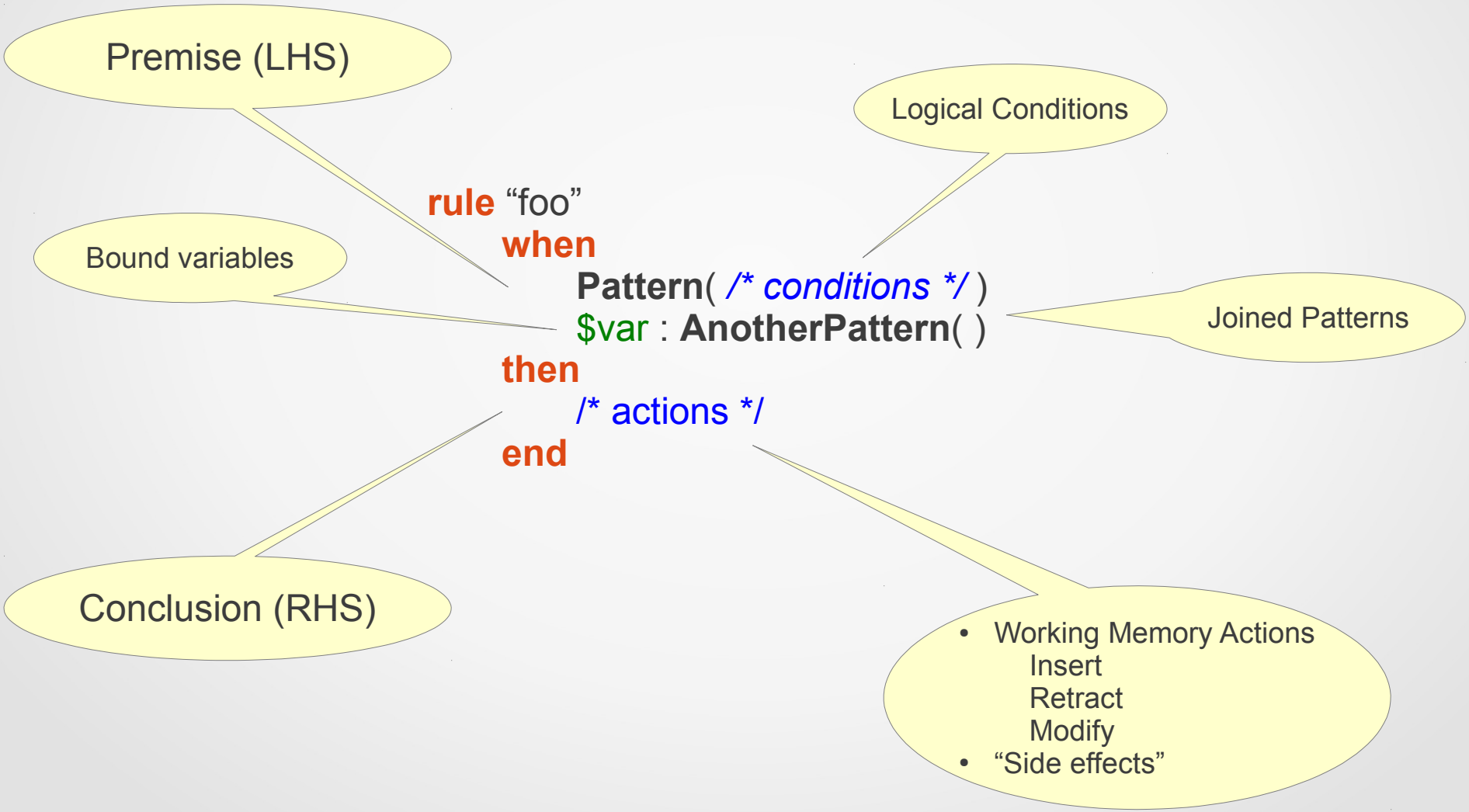
Started as a “friday afternoon” project, sometimes considered just the open source “clone” of Jess, but...

Drools Expert

- Traditional “forward chaining” architecture



Drools Expert



Drools Expert

- **Native Java implementation**
  - Object-Oriented engine
  - Object-Oriented rules and facts
    - (Java) Classes, *Interfaces* and properties can be used directly
      - As facts
      - As patterns
- **Declarative model**
  - Runtime code generation and compilation
- **Model Import**
  - **XSD** (via JaxB / xjc)
  - **UML** (missing.. anyone?)
  - **OWL**

```
declare Person
        name    : String
        age     : int
end

declare Patient extends Person
        mrn     : String
end
```

Drools Expert - Advanced features

- **Complex operators**
  - **Quantifiers**: exists, forall, not*
  - **Aggregators** : sum, avg, collect, ...
- **Truth maintenance**
  - Deep retraction
- “In-place” **modifications**
  - Property reactivity (limited refraction)
- **Queries** → backward chaining
  - Limited unification
- **Free-form constraints**
  - Use expressions to filter patterns

PRR Compliance

Drools Expert - Demo

- Some basic examples...
  Demo from https://github.com/droolsjbpm/drools/tree/master/drools-examples
- Wumpus world !
  Demo https://github.com/droolsjbpm/drools/tree/master/drools-examples/src/main/java/org/drools/games

BRMS : Guvnor

- Centralized Knowledge Repository
  - Web Application
    - Access Control and Security
  - Versioning Repository
  - Assisted deployment and distribution
    - “Hot-swap” knowldege base updates
  - Assisted authoring
  - Integrated testing

Guvnor tooling : guided editor

- “Stylesheet” for technical rule language
- Controlled authoring

Guvnor tooling : Decision Tables

- Compact notation for rule logic
  - Generate DRL from templates

	#	Description	salience	name	age	age
	1		1	Bill	30	12345
	2		2	Ben	<otherwise>	12345
	6		6	Weed	40	12345
	7		7	<otherwise>	50	

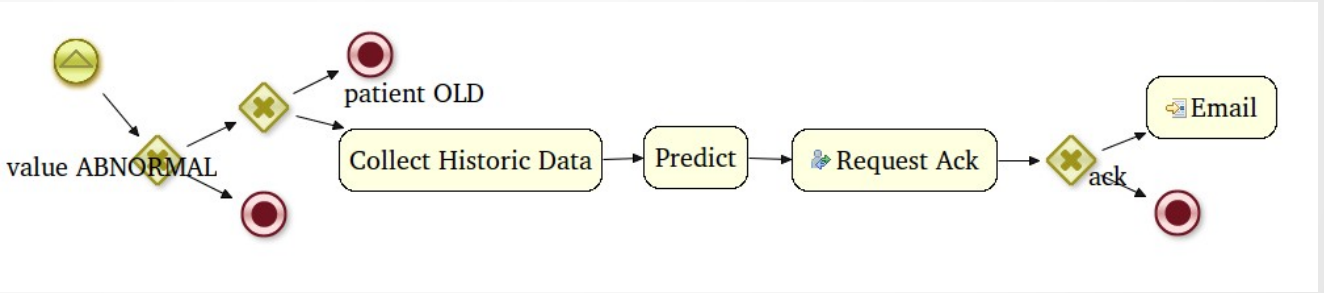
Waiting for the **DMN** standard....

JBPM

(previously : Drools Flow)

jBPM

- Business Process Engine
  - BPMNv2 compliant
    - Only a covers a subset of the features
  - Native **Rule** support
    - Branching logic
    - “Rule tasks”
    - Fact sharing



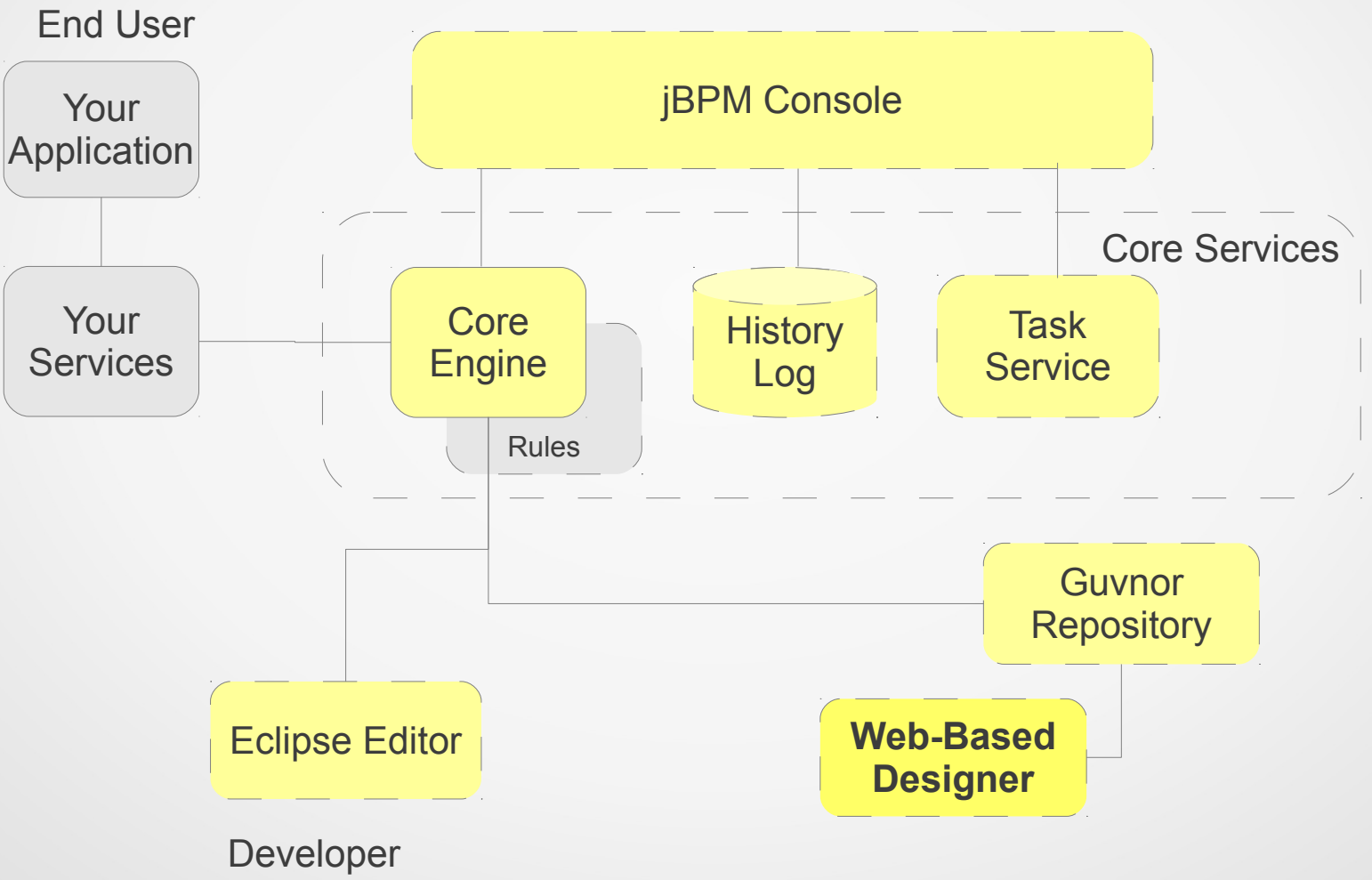
  - Native Event support
  - Native Human Task support
    - Based on WS-HumanTask specification

jBPM - Integration

- Management console
  - Process Instance management
  - Active Task Lists
  - Human Task Form management
  - Reporting

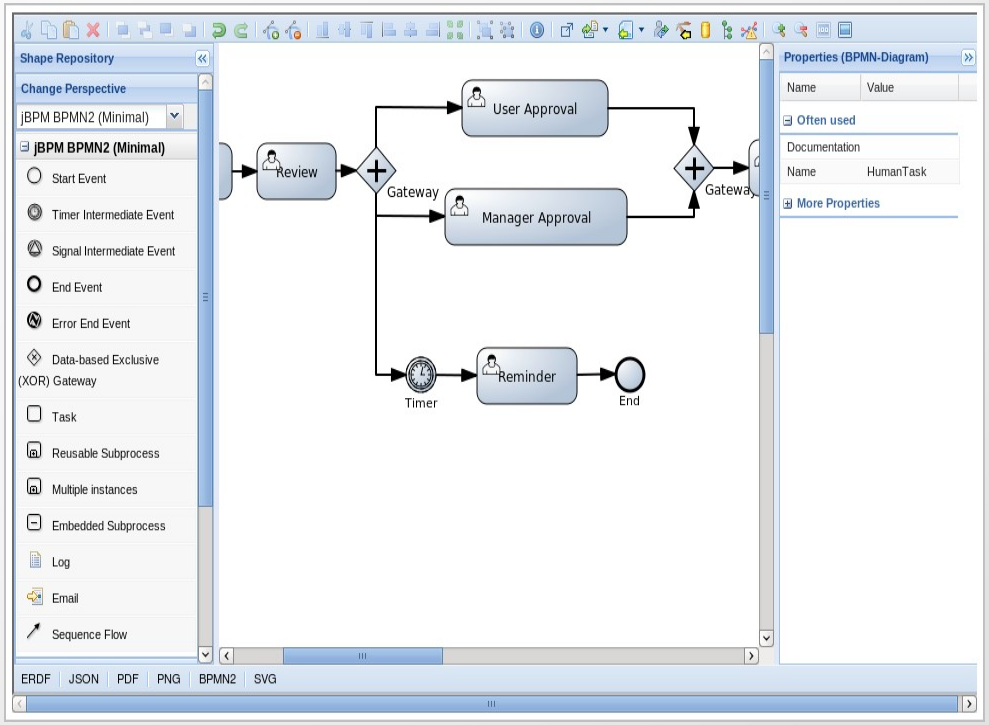
- Pluggable persistence : JPA
- Pluggable transactions : JTA
- Pluggable human task service : WS-HumanTask
- Pluggable process repository (optional)
- History logging (for querying / monitoring / analysis)
- Framework support : Seam, Spring, OSGi, ...

jBPM - Architecture



jBPM Designer

- BPMNv2 authoring environment
  - Guided editor
    - *Workflow patterns*
  - Process Simulation



Drools Fusion

Drools Fusion

- Enables "*temporal awareness*" in Drools Expert
  - Event semantics
    - ***Event*** : significant state change at a given time
  - ***Complex*** Event Processing
    - Detection
    - Processing
      - Aggregation
      - Correlation
      - Abstraction
    - Reaction

```
declare Encounter
        @Role( event )
        @Timestamp( dateTime )

        patientId       : String
        providerId      : String
        dateTime        : Date
end
```

Drools Fusion - enabled rules

- Temporal Rules
  - Sliding Windows
    - Temporal
    - Length
  - Temporal Constraints
    - Allen temporal logic

```
rule “fusion”
when
        ...
        $dx : Diagnosis( code == “1.2.3” )
        accumulate (  $e : Encounter( this after $dx )
                        over window:length[ 30d ],
                $num : count( $e );
                $num < 3
        )
then
        /* inadequate follow-up */
end
```

Drools fusion – enabled rules

- Scheduling and delaying

```
rule "fusion"
        timer( 1h )
        when
                $dx : Diagnosis( code == "1.2.3" )
                not Encounter( this after[0, 15d] $dx )
        then
                /* wait up to 15d + 1hr, then alert! */
        end
```

- Scalability optimizations
  - Automatic fact retraction
  - Stream processing

Drools Core

Expert + Fusion + jBPM

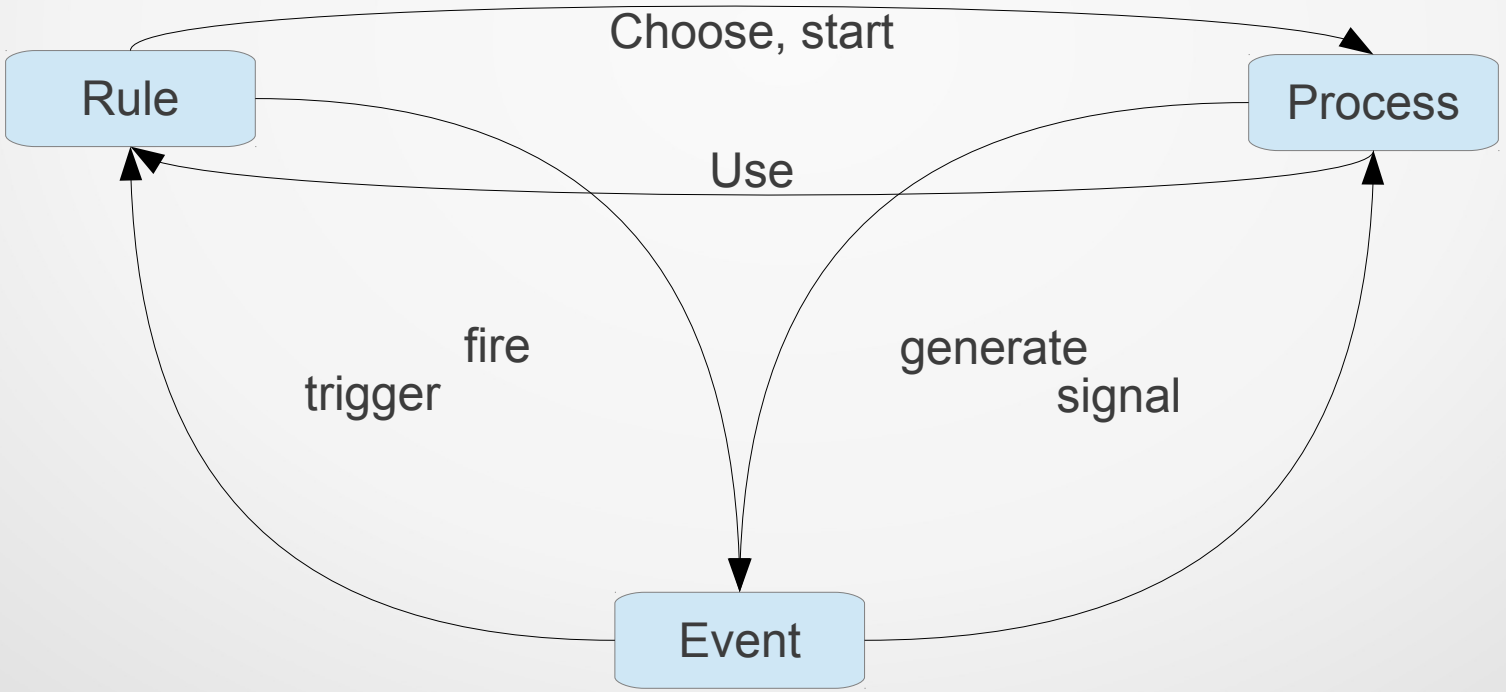
- Hybrid SOA / EDA architectures

	SOA	EDA
Coupling	Loose	Decoupled
Interaction	Synchronous	Asynchronous
Coordination	Scheduling	Reaction
Trigger	Consumer	Producer
Management	Orchestration	Pub/Sub
Communication	One-to-one	Many-to-Many

Drools Core

Expert + Fusion + jBPM

- Hybrid SOA / EDA architectures



OptaPlanner

(previously Drools Planner)

Constrained Optimization Problems

- Problem : Maximize/Minimize a ***goal...***
  - Assigning values to *variables*
    - *Degrees of freedom*
  - With *limited* *resources*
    - *(CPU) Time, (Memory) Space, Money, etc..*
  - Under a number of *constraints*
    - Not all variable values, or combinations thereof, are admissible
  - *Optimum vs Optimal* solution(s)
    - Potentially many solutions satisfy the constraints
    - Some are better than others w.r.t. the goal

Constrained Optimization Problems

- Maximize a ***goal...***
  - ***when the problem is difficult***
  - NP-Complete Problems
    - Do not “scale well” → impossible to solve *in practice*
    - *But a “good enough”* solution may be acceptable

Drools Planner

- Define a problem
  - Annotated (OO) **domain model**
    - Planning variables
      - Admissible domain
      - Owning planning entity
    - Solution
      - Collection of entities with assigned variables
    - Constraints
      - As Drools rules!
        - Scalable engine performance
        - Incremental, Differential evaluation

Drools Planner Solver

- Explores the “solution space”
  - Highly configurable
  - Uses the constraints as guides
  - **Meta-heuristic N-phase optimization:**
    - Initialization (Global search)
      - Brute Force
      - Best Fit, First Fit, Last Fit
    - Refinement (Local search)
      - Hill Climbing
      - Simulated annealing
      - Taboo search

- Planner Demo
  - Nurse Rostering
  - Routing

  - Examples can be downloaded and executed from http://www.optaplanner.org/