Full Scale STP with jBPM

Use case about jBPM at SNS Bank

Eric D. Schabell
RedHat, Solution Architect

Maurice de Château
SNS IT, System Specialist (Java)

Agenda

- **SNS Bank STP Strategy**
- Current STP Products
- How does jBPM fit in?
- 'Legacy' tooling and way of working
- Current improvements
- Future directions

About SNS Bank

- 4th largest bank in the Netherlands
- Origin in savings banks
- Aims mainly at private individuals and small and medium-sized businesses
- Financial conglomerate with REAAL Insurances since 1997
- Several sublabels (ASN, BLG, ZwitserLeven)

STP Strategy

- **S**traight **T**hrough **P**rocessing
- 5 clicks to purchase new products (2010)
- Central focus: customer experience
- Transparent, quick and simple
- Effective and efficient, eliminating handwork
- Paperless (as much as legally possible)

Agenda

- SNS Bank STP Strategy
- **Current STP Products**
- How does jBPM fit in?
- 'Legacy' tooling and way of working
- Current improvements
- Future directions

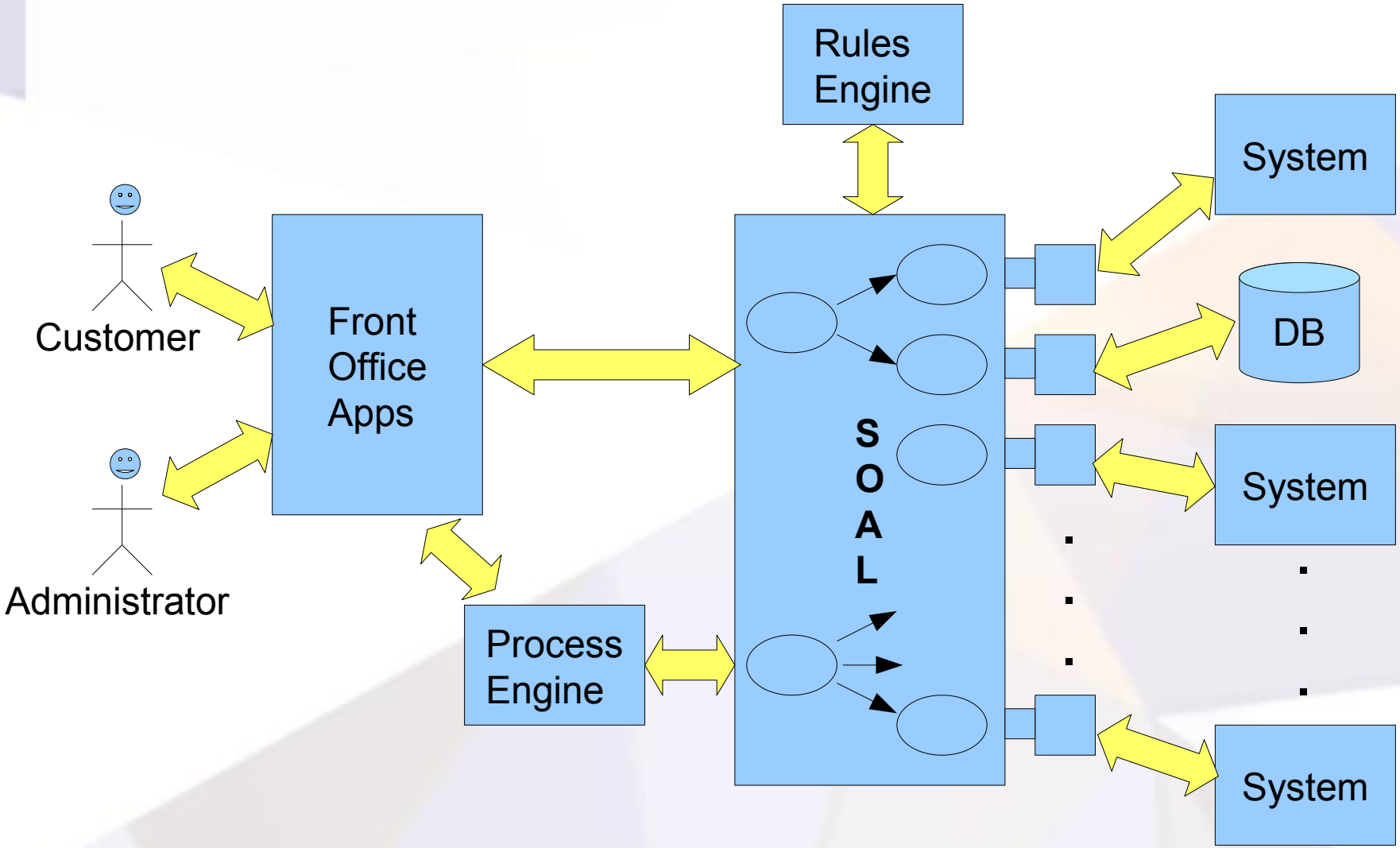
STP Products

- Savings accounts (Jaarsparen, Internetsparen, Maxisparen, Spaarmix, Plussparen)
- Deposit accounts (Depositosparen (interest per year or per month), Klimrente, Varivast, Rendementsparen)
- Service processes (among others Change of address, Change of contra account, Temporary increase of debit card limit, and many more to follow...)

Agenda

- SNS Bank STP Strategy
- Current STP Products
- **How does jBPM fit in?**
- 'Legacy' tooling and way of working
- Current improvements
- Future directions

Simplified System Overview



System Restrictions

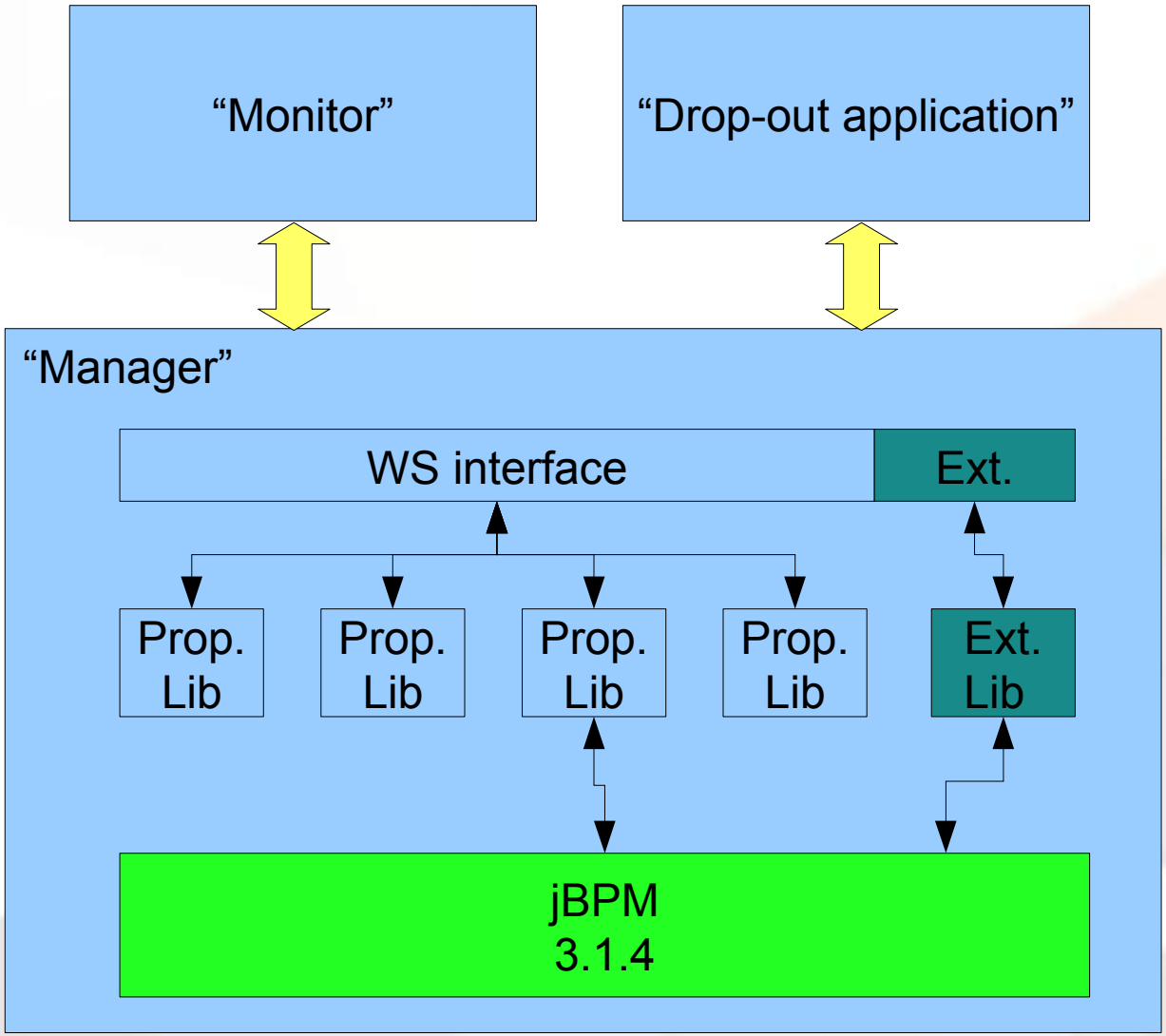
- No ESB
- No integrated rules engine
- Some (prominent) back ends don't support transactionality/asynchronicity/events

- And...

Agenda

- SNS Bank STP Strategy
- Current STP Products
- How does jBPM fit in?
- **'Legacy' tooling and way of working**
- Current improvements
- Future directions

Tooling


"Monitor"
"Drop-out application"
"Manager"
WS interface
Ext.
Prop. Lib
Prop. Lib
Prop. Lib
Prop. Lib
Ext. Lib
jBPM 3.1.4

Tooling

open chain

Atos Origin

Home

Beheer Afmelden

Algemene eigenschappen

Technisch id : 1708209

Naam : VerwerkenKlantEnProspectCon

Versie : 5

Aantallen

Totaal aantal instanties : 121

Aantal draaiende instanties : 120

Aantal gepauseerde instanties : 0

Aantal beeindigde instanties : 0

Aantal instanties met problemen : 1

Procesinstanties | Grafisch | Definitiedetails | Misgelopen timers

Aantallen per node

Nodenaam	Aantallen		
	✓	⚠	Σ
Zet aanvraag in behandeling			
Toevoegen verkoopkans			
validatie akkoord			
klantnummer beschikbaar			
Match klanten of prospects			

Alle procesinstanties

id	flowid	node	start	node date
1711144	67761	BackendMelding	2009.04.14 - 15:24:44	2009.04.14 - 15:29:04
1711122	67761	BackendMelding	2009.04.14 - 15:14:40	2009.04.14 - 15:15:38
1711104	67761	Start	2009.04.14 - 15:12:12	
1711082	67761	BackendMelding	2009.04.14 - 15:11:37	2009.04.14 - 15:12:12

Acties

De onderstaande acties gelden voor alle aangevinkte procesinstanties:

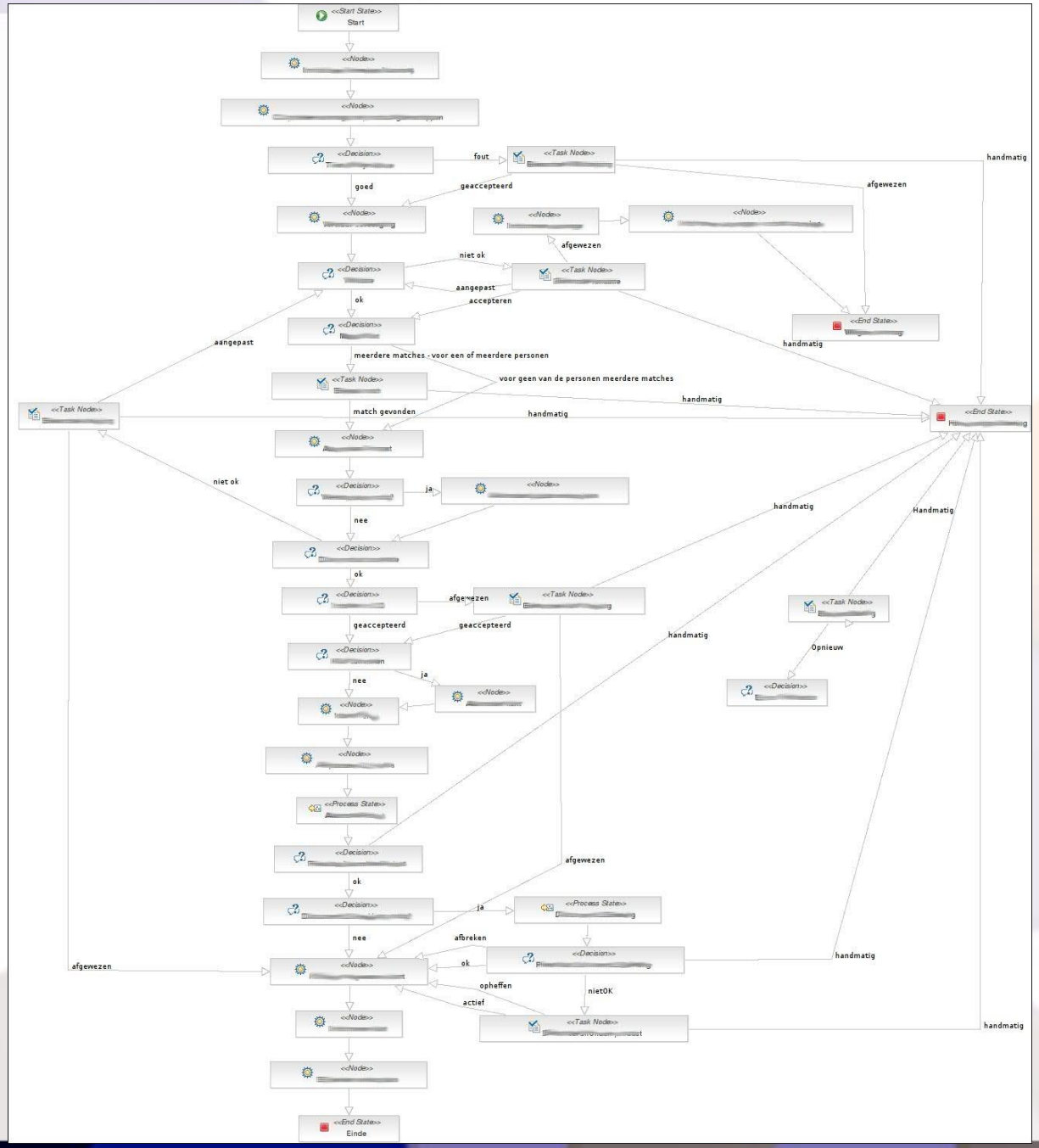
Verwijderen:

Way of working (1)

- The only node types used:
  - Node (for all automated tasks, containing all necessary business logic)
  - Decision
  - Task-node (for the human tasks)
- Synchronous execution
  - Entire process runs in the thread it is started with (so no intermediate persisting)

Way of working (2)

- Synchronous scheduler
  - Starts a list of process instances one after another, so...
- Java exception handling by using one top-level jBPM exception-handler
  - Control is passed to the drop-out application by jumping to a loose task
- Hibernate configured to auto-commit


<<Start State>>
Start
<<Node>>
<<Node>>
<<Decision>>
fout
<<Task Node>>
handmatig
goed
geaccepteerd
afgewezen
<<Node>>
<<Node>>
<<Node>>
afgewezen
niet ok
<<Decision>>
<<Task Node>>
aangepast
ok
accepteren
<<Decision>>
<<End State>>
aangepast
handmatig
meerdere matches - voor een of meerdere personen
<<Task Node>>
voor geen van de personen meerdere matches
handmatig
<<Task Node>>
match gevonden
handmatig
<<End State>>
<<Node>>
niet ok
<<Decision>>
ja
<<Node>>
handmatig
Handmatig
nee
<<Decision>>
ok
<<Decision>>
afgewezen
<<Task Node>>
<<Task Node>>
geaccepteerd
geaccepteerd
handmatig
<<Decision>>
Opnieuw
ja
nee
<<Node>>
<<Decision>>
<<Node>>
<<Node>>
<<Process State>>
<<Decision>>
afgewezen
ok
<<Decision>>
ja
<<Process State>>
nee
afbreken
<<Decision>>
handmatig
afgewezen
<<Node>>
ok
opheffen
nietOK
actief
<<Node>>
<<Task Node>>
handmatig
<<Node>>
<<End State>>
Einde

Funny (?) Code Example (1)

```
public class BackendExceptionHandler implements ActionHandler {
    public void execute(ExecutionContext context) throws Exception {
        Token token = context.getProcessInstance().getRootToken();
        String originatingNode = token.getNode().getName();
        if (!BackendExceptionHandler.BACKENDERROR_REDIRECTING_NODE.equals(originatingNode)) {
            ExecutionContextHelper.setVariable(context, BACKENDERROR_DROPOUT_NODE,
               token.getNode().getName());
        }

        token.setNode(context.getProcessDefinition().getNode(BACKENDERROR_NODE_NAME));
        token.signal();

        throw new Exception("Functional error on back end.");
    }
}
```

Funny (?) Code Example (2)

```
public class SaveProcessInstanceHandler implements ActionHandler {
    public void execute(ExecutionContext context) throws Exception {
        context.getJbpmContext().getConnection().setAutoCommit(false);
        context.getJbpmContext().getConnection().commit();
        context.getJbpmContext().getConnection().setAutoCommit(true);

        // Do something with context and changes will be saved, this call results
        // in an empty list Remark: not every call to context will result in a DB update.
        // This one does...
        context.getJbpmContext().getTaskList();

        [further processing...]
    }
}
```

Agenda

- SNS Bank STP Strategy
- Current STP Products
- How does jBPM fit in?
- 'Legacy' tooling and way of working
- **Current improvements**
- Future directions

Improvements

- Most business logic moved to services
  - At most one WS call to a back end
  - Lack of transactionality less of a problem
- “State Proxy” solution
  - Makes the back end calls asynchronous
  - Allows for the use of proper state nodes

Agenda

- SNS Bank STP Strategy
- Current STP Products
- How does jBPM fit in?
- 'Legacy' tooling and way of working
- Current improvements
- **Future directions**

Wish List

- Asynchronous (parallel) scheduling
- Implementing multi-process solutions
- Removal of the legacy tooling
- Upgrade to jBPM 3.2.x
- Proper use of swimlanes and the possibilities of tasks in the jBPM console

Questions?