

# PDI + Jare Ruleengine

## Pentaho PDI and the Jare Ruleengine

Enhanced Agility - Enhanced Quality

# We are good developers !

Our skills:

- ♦ Pentaho tools, SQL, Scripting
- ♦ Source/Target Systems, Interfaces
- ♦ Other technologies
- ♦ Automation, Optimization, Standardization
- ♦ Programming languages
- ♦ Different data formats
- ♦ and more...

# What works against us:

Good – Fast – Cheap: Pick only **two** !



# Good - Fast - Cheap

- ♦ Good + Fast = Expensive

Choose good and fast and we will postpone every other job, cancel all appointments and stay up 25-hours a day just to get your job done. But, don't expect it to be cheap.

- ♦ Good + Cheap = Slow

Choose good and cheap and we will do a great job for a discounted price, but be patient until we have a free moment from paying clients.

- ♦ Fast + Cheap = Inferior

Choose fast and cheap and expect an inferior job delivered on time. You truly get what you pay for, and in our opinion this is the least favorable choice of the three.

# Other “disturbing” factors

- ♦ ever-changing business
- ♦ changing teams/skills
- ♦ incomplete/late requirements
- ♦ changing requirements
- ♦ inadequate resource planning
- ♦ timeline defined without us

# That's life

... but that's life !



# Let's make life easier

... so let's do something about it,

on our end !



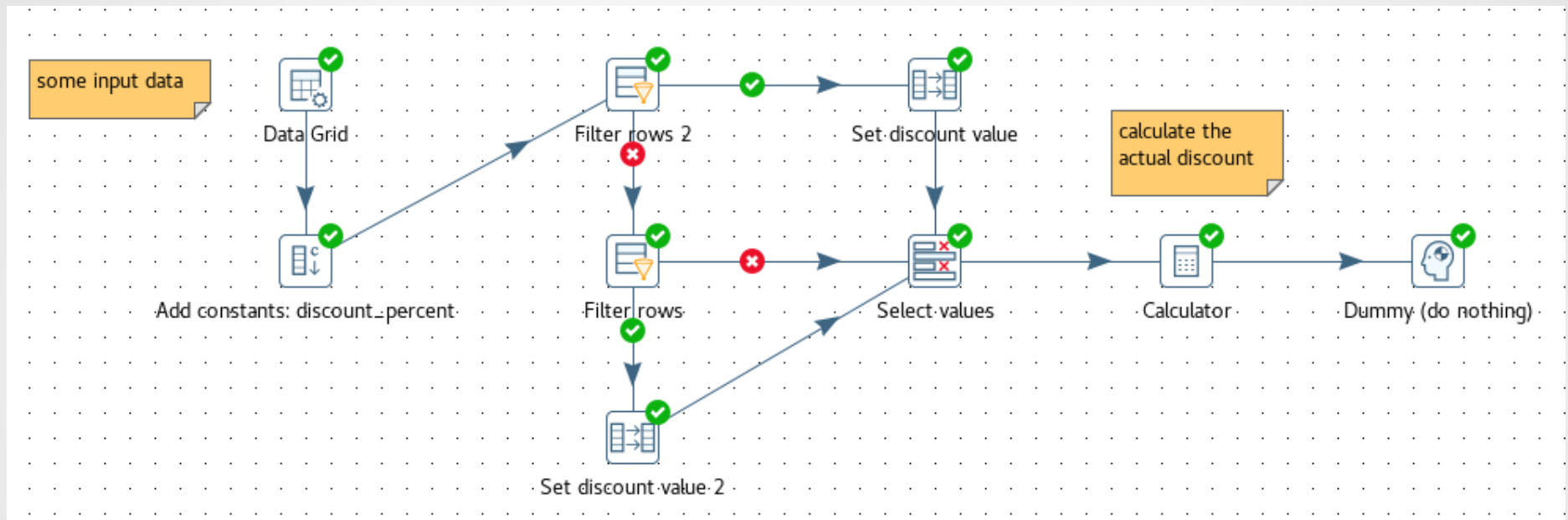
# Business Logic: What's the problem?

- we are responsible for Business Logic **and** ETL or code
- same business requirements in different places (Good+Fast+Cheap)
- in-transparent for business users
- business requirements clutter our ETL flow

Result: bad for quality, bad for agility

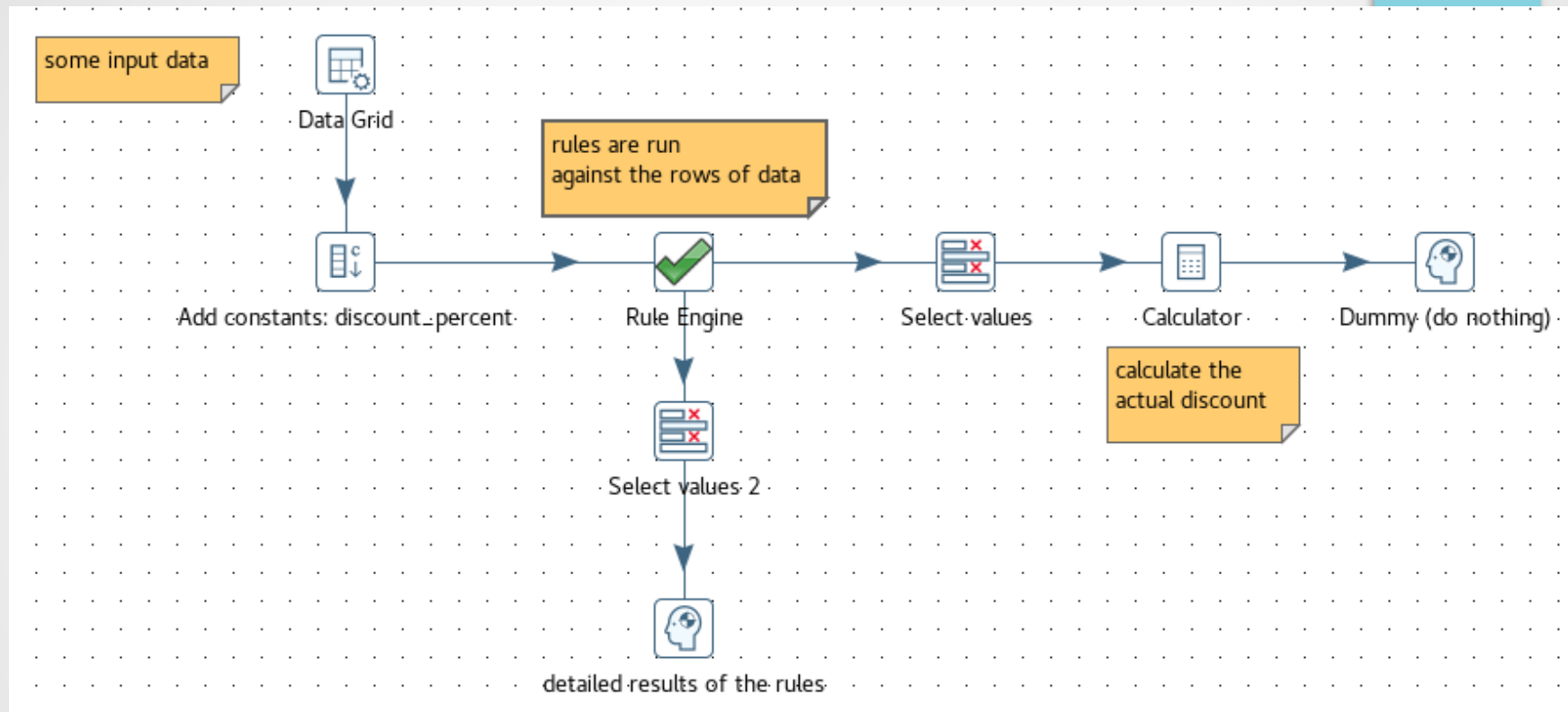


# PDI Sample without Ruleengine



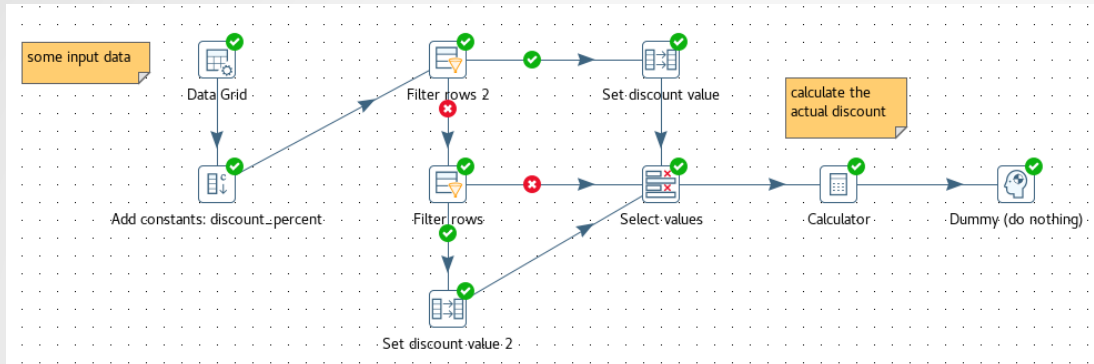
- ♦ over time: more complexity = more clutter = slowdown
- ♦ difficult to explain to business user (in-transparent)

# PDI Sample with Ruleengine



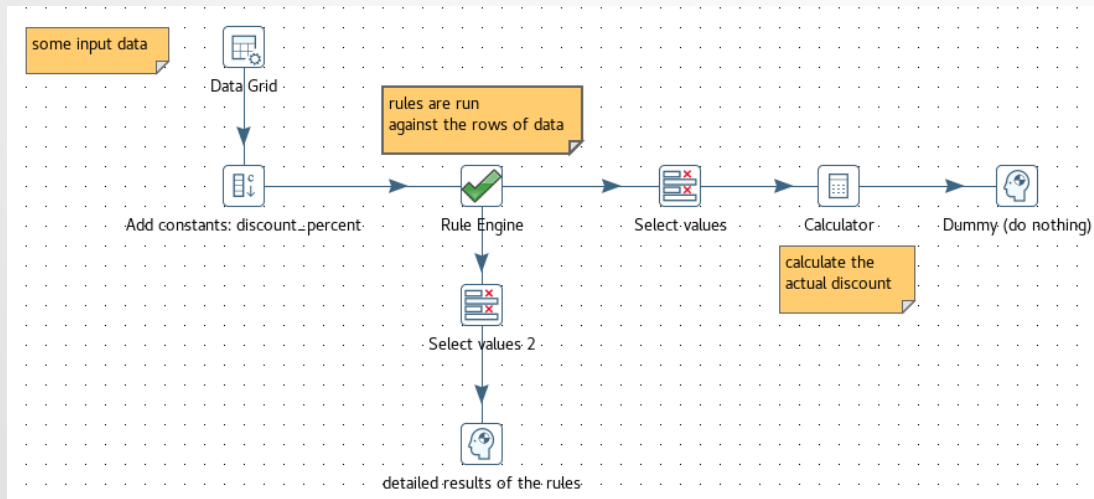
- ♦ less complexity = less clutter = more agile
- ♦ Business rules are managed in a web application

# New/changing Requirements



## Without Ruleengine:

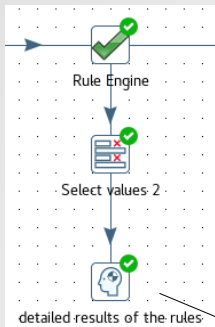
- change in ETL
- changes elsewhere??
- clutter, complexity



## With Ruleengine:

- No change in ETL
- all changes externally
- less clutter – less complex

# Ruleengine detailed results



Examine preview data

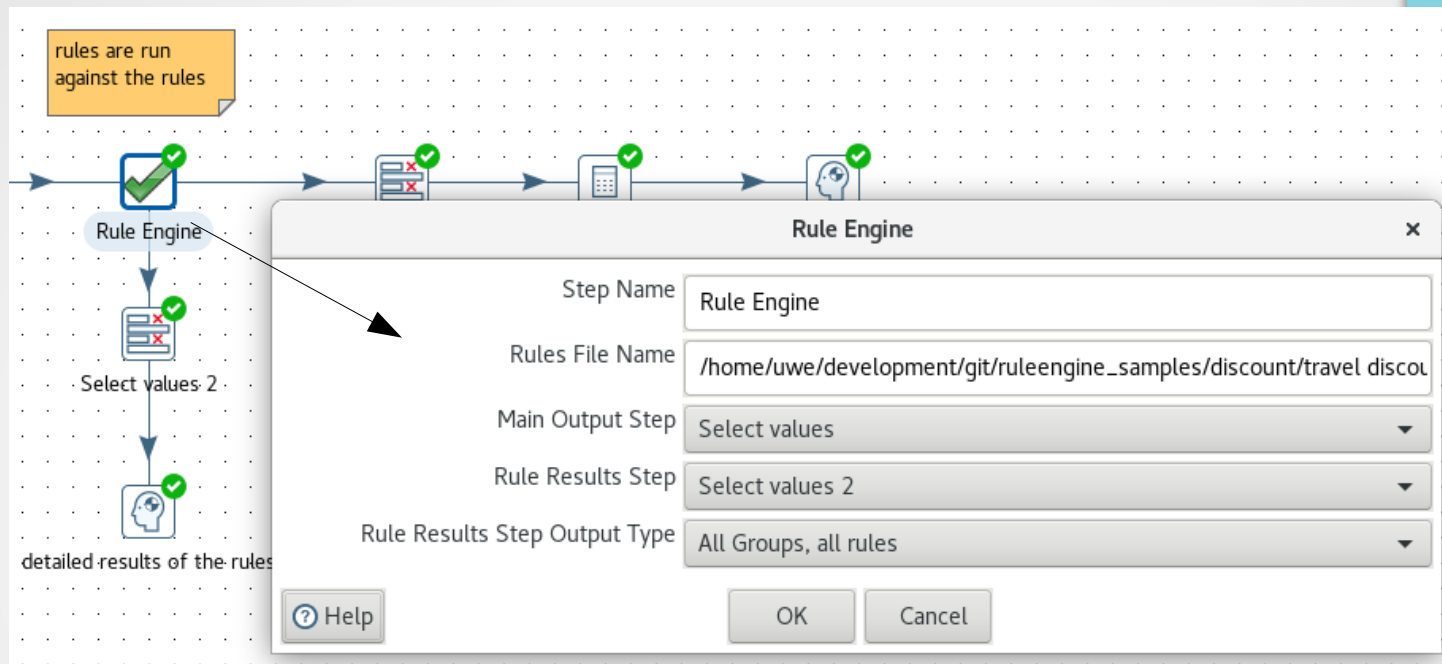
Rows of step: detailed results of the rules (36 rows)

▲	id	age	destination_rejior	price	discount_percent	ruleengine_group_faile	ruleengine_rule	ruleengine_rule_faile	ruleengine_message
1	1	1	Asia	100.0	100	0	Check Age	0	Field age [1] correctly is smaller than [3]
2	1	1	Asia	100.0	100	1	Check Age 50+	1	Field Age [1] incorrectly not is greater or equal than [50]
3	1	1	Asia	100.0	100	1	Destination Europe	1	Field Destination Region [Asia] incorrectly not [Europe]
4	2	12	America	150.0	0	1	Check Age	1	Field age [12] incorrectly not is smaller than [3]
5	2	12	America	150.0	0	1	Check Age 50+	1	Field Age [12] incorrectly not is greater or equal than [50]
6	2	12	America	150.0	0	1	Destination Europe	1	Field Destination Region [America] incorrectly not [Europe]
7	3	33	Africa	120.0	0	1	Check Age	1	Field age [33] incorrectly not is smaller than [3]
8	3	33	Africa	120.0	0	1	Check Age 50+	1	Field Age [33] incorrectly not is greater or equal than [50]
9	3	33	Africa	120.0	0	1	Destination Europe	1	Field Destination Region [Africa] incorrectly not [Europe]
10	4	24	Europe	200.0	0	1	Check Age	1	Field age [24] incorrectly not is smaller than [3]
11	4	24	Europe	200.0	0	1	Check Age 50+	1	Field Age [24] incorrectly not is greater or equal than [50]
12	4	24	Europe	200.0	0	1	Destination Europe	0	Field Destination Region [Europe] correctly [Europe]

debugging inclusive !

>> explains what happened (for each rule and row)

# Ruleengine Plugin properties



- ♦ specify:
  - ♦ file path and name containing business logic
  - ♦ main output step (1 row in, 1 row out)
  - ♦ output step for detailed results (1 result per rule and row)
  - ♦ which details to output (failed, passed, all)

# Ruleengine Technical Details

## Ruleengine:

- ♦ filter and/or update data
- ♦ execute one to many rules (per rulegroup)
- ♦ execute one to many actions (per rulegroup)
- ♦ rulegroups, rules, actions in XML format
- ♦ project file: zip file containing all rulegroups (logic)
- ♦ rules and actions instantiated and executed using Java Reflection (classnames, methodnames)
  - >> instantiate any Java object
  - >> execute on: any data format: CSV, XML, JSON, database resultsets, ...

# Business Logic Maintenance: Project

The screenshot displays the 'Business Rules Maintenance Tool' interface in a Mozilla Firefox browser window. The browser's address bar shows the URL: `localhost:8081/rule_maintenance/process?action=bsh&ro=1&scriptname=listprojects`. The page header includes the 'DATAMELT' logo on the left and a user profile picture on the right. A sidebar on the left contains navigation links: START PAGE, PASSWORD, LOGOUT, PROJECTS, HISTORY, ACTIVITY, SEARCH, EXPORT, IMPORT, CHECKS, ACTIONS, DOCUMENTATION, and ABOUT. The main content area is titled 'Business Rules Maintenance Tool' and features a 'Projects List (1)' section. Below this, there is a 'Filter:' input field and a project entry for 'Travel Discount'. The project entry shows 'Travel Discount' as the name, 'group: User Group 1, owner: admin user' as details, and 'updated: 2017-08-20 13:02' as the timestamp. The project is marked as 'private' and has a document icon next to it.

Business Rules Maintenance Tool | Projects List - Mozilla Firefox

Business Rules Maintainer

localhost:8081/rule\_maintenance/process?action=bsh&ro=1&scriptname=listprojects

Fedora Project Free Content Understanding MapRed... Google Maps Nifi Ricoh InfraPort Pike Suggester - decin... Cockpit

DATAMELT

Business Rules Maintenance Tool

Projects List (1)

Filter:

project (private)

Travel Discount

Travel Discount

group: User Group 1, owner: admin user

updated: 2017-08-20 13:02

# Business Logic Maintenance: Rule groups

The screenshot displays the 'Business Rules Maintenance Tool' interface in a Mozilla Firefox browser window. The browser's address bar shows the URL: `localhost:8081/rule_maintenance/process?action=bsh&ro=1&scriptname=listrulegroups&projectid=51`. The tool's header includes the 'DATAMELT' logo, the title 'Business Rules Maintenance Tool', and a user profile picture. A left-hand navigation menu contains links for 'START PAGE', 'PASSWORD', 'LOGOUT', 'PROJECTS', 'HISTORY', 'ACTIVITY', 'SEARCH', 'EXPORT', 'IMPORT', 'CHECKS', 'ACTIONS', 'DOCUMENTATION', and 'ABOUT'. The main content area is titled 'Rule Groups List (2) - Project: Travel Discount' and features a 'Filter:' input field. Below this, a 'project (private)' card for 'Travel Discount' is shown. Two 'rulegroup' cards are displayed: 'Age under 3=100%' with a description 'Age under three 100% Discount' and a validity period of '2016-11-01/2017-12-31'; and 'Europe 50+ 25%' with a description 'Discount 25% with Destination Europe and Age 50+' and a validity period of '2016-06-01/2019-12-31'. Each rule group card includes icons for document, edit, and delete actions.



# Business Logic Maintenance: Rule groups

The screenshot displays the 'Business Rules Maintenance Tool' interface in a Mozilla Firefox browser window. The browser's address bar shows the URL: `localhost:8081/rule_maintenance/process?action=bsh&ro=1&scriptname=listrulesubgroups&rulegroupid=110`. The tool's header includes the 'DATAMELT' logo, the title 'Business Rules Maintenance Tool', and a user profile picture. A left-hand navigation menu contains links for 'START PAGE', 'PASSWORD', 'LOGOUT', 'PROJECTS', 'HISTORY', 'ACTIVITY', 'SEARCH', 'EXPORT', 'IMPORT', 'CHECKS', 'ACTIONS', 'DOCUMENTATION', and 'ABOUT'. The main content area, titled 'Rules List - Project: Travel Discount - Rule Group: Europe 50+ 25%', displays a hierarchical view of the rule structure. It includes a 'project' box for 'Travel Discount', a 'rulegroup' box for 'Europe 50+ 25%' (valid from 2016-06-01 to 2099-12-31), a 'subgroup' box for 'subgroup\_1', and two 'rule' boxes: 'Check Age 50+' (checking if age is greater than or equal to 50) and 'Destination Europe' (checking if the destination region is Europe). A 'rules connector' box shows the connector is 'and'. At the bottom, an 'action' box is titled 'Set 25% discount', with the description 'Set 25% discount' and the update logic 'update field [discount\_percent] set value (number) from: parameter [25]'. The execution condition is 'execute: when rulegroup passed'.

maintenance by the business user (expert) – not IT !

# Business Logic Maintenance: Live Demo

## **Live Demo** Business Rules Maintenance Tool

if you missed it, there's a video on YouTube:

# Business Logic Maintenance: Features

- ♦ copy projects, groups, rules, actions
- ♦ groups of rules have an effective date and expiry date
  - ♦ plan changes ahead
  - ♦ automatic activation and expiry of logic
- ♦ logic can be tested within the web application
- ♦ history of changes, activities, search
- ♦ project export and import
- ♦ users, roles >> security
- ♦ generates documentation of the logic

# Business Logic Maintenance: Advantages

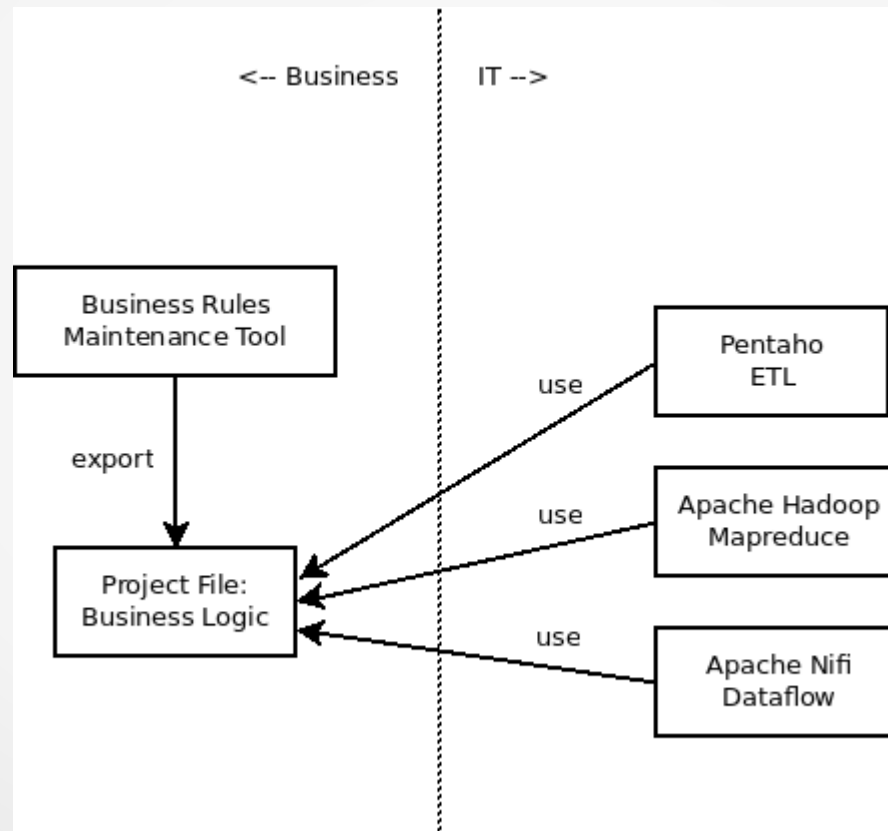
- ♦ Business logic is outside of PDI
  - ♦ less clutter in ETL
  - ♦ less duplication
  - ♦ less complex ETL
  - ♦ change to logic does not require change to ETL
- ♦ Business logic located in a central place
  - ♦ Web application for Business Users
  - ♦ if people come and go, better for continued service
  - ♦ Review easier + documentation of logic

# Business Logic Maintenance: Results

- ♦ Business logic is outside of PDI
  - >> agility, quality for IT
  - >> future ETL changes are easier to implement
- ♦ Business logic is in a central place
  - >> agility, quality for Business User
  - >> more transparent for Business User
  - >> proper separation of responsibilities between IT/Business

# Ruleengine: Central Logic

use Ruleengine in other environments:



# Ruleengine: Advantages

- ♦ PDI Step + Web application
  - ♦ PDI step: install from Marketplace
  - ♦ production ready
  - ♦ open source Apache license, documented
  - ♦ rules and actions extendable
- ♦ use Ruleengine in other environments (logic still centrally!):
  - ♦ Hadoop map-reduce
  - ♦ Nifi (processor available)
  - ♦ Kafka
  - ♦ use in your own java project



Thank you

**Thank you for your precious time !**

... and please send feedback  
or let's discuss



# Blog Posts

- Why a rule engine helps IT
- Video: Calculate discount using Pentaho PDI and Business Rule Maintenance Tool
- Ruleengine now with client/server mode
- Rule Engine: Different Types of checks
- Apache Drill - Java Code sample
- Apache Drill - Easy access to old data
- Apache Drill - CSV Query Sample
- Apache Drill - CSV to Parquet files
- Apache Drill - Querying Hadoop (HDFS)
- Ruleengine - Division of Responsibilities
- Apache Nifi - Starter
- Apache Nifi combined with Apache Velocity
- Twitter Tweets - Apache Nifi combined with MongoDB, Groovy, Velocity and Highcharts
- Ruleengine - Project Overview
- Ruleengine - Discount example
- Business Rules Maintenance Tool - Test your rules
- Ruleengine: dynamically process files with Metadata Injection
- Ruleengine: Use the ruleengine from code
- Hadoop - Raspberry PI Cluster
- Hadoop and the Ruleengine - example
- Hadoop and the Ruleengine
- Nifi Processor: RuleEngine
- Nifi Processor: GenerateData
- Nifi ExecuteRuleEngine processor: Run sample explanations & screenshot
- Nifi: RuleEngine Use Case
- Nifi ExecuteRuleEngine processor: Flow Screenshots
- Apache Nifi with the Ruleengine - Screencast

<http://datamelt.weebly.com>

# Contact

**Uwe Geercken**

email: [uwe.geercken@web.de](mailto:uwe.geercken@web.de)

twitter: [uweeegeeee](#)

github: <http://github.com/uwegeercken>

blog: <http://datamelt.weebly.com>