CAMUNDA
CON
2020.2

holisticon

DMN
on Steroids 2.0

To Decision Tables and beyond

Who am I ?

Holisticon AG is a Management and IT consulting company based in Hamburg, Germany.

- *Founded 2007 // 13+ years*
- *Offices in Hamburg, Hannover, Kiel*
- *About 70+ handpicked consultants*
- *Camunda Advanced Partner*

holisticon

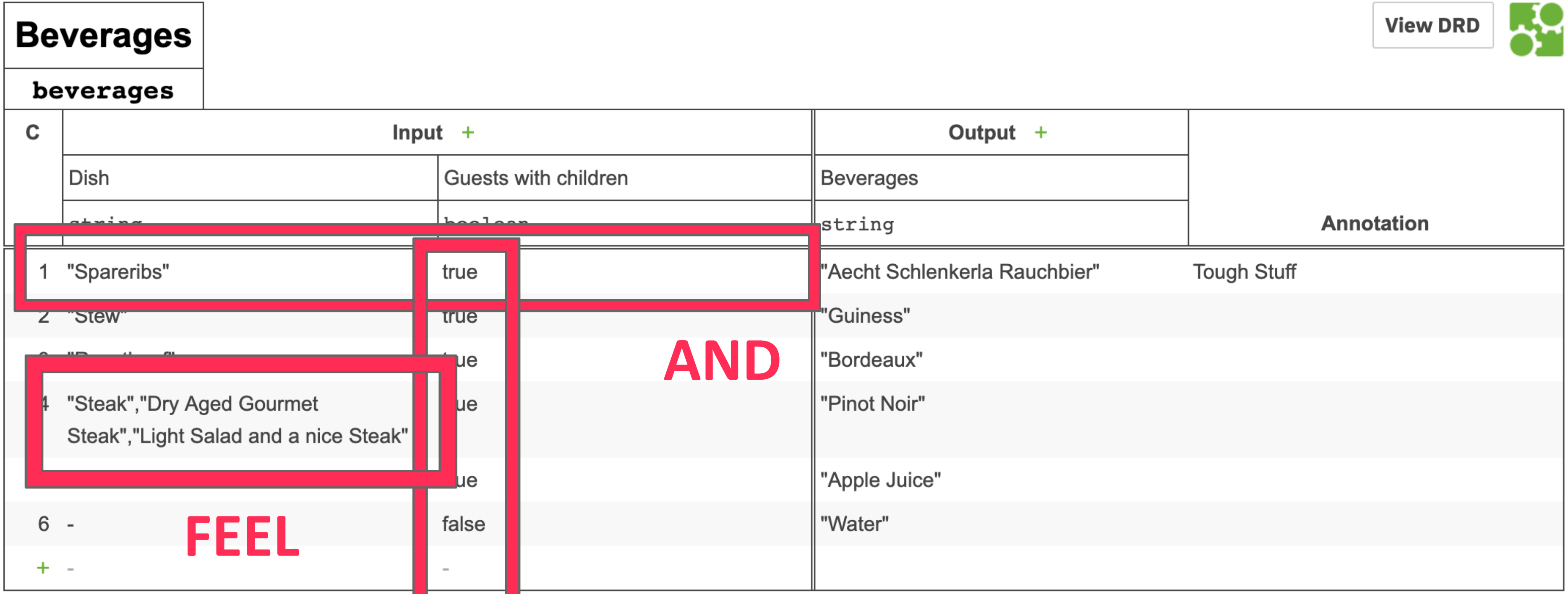
Jan Galinski

- *~ 15 years Workflow Management*
- *~ 8 years with Camunda*
- *Contributor and (Co-)Author of spring-boot-starters, mockito extension, reactor extension, …*
- *Camunda User Group Hamburg*
- *BPM Craftsman, Consultant, Developer*

https://about.me/jangalinski

What is DMN ?

What is DMN?$_1$ – Decision Table



Beverages

beverages

C	Input +		Output +	
	Dish	Guests with children	Beverages	
	string	boolean	string	Annotation
1	"Spareribs"	true	"Aecht Schlenkerla Rauchbier"	Tough Stuff
2	"Stew"	true	"Guiness"	
3	"Roastbeef"	true	"Bordeaux"	
4	"Steak","Dry Aged Gourmet Steak","Light Salad and a nice Steak"	true	"Pinot Noir"	
		true	"Apple Juice"	
6	-	false	"Water"	
+	-	-		

AND

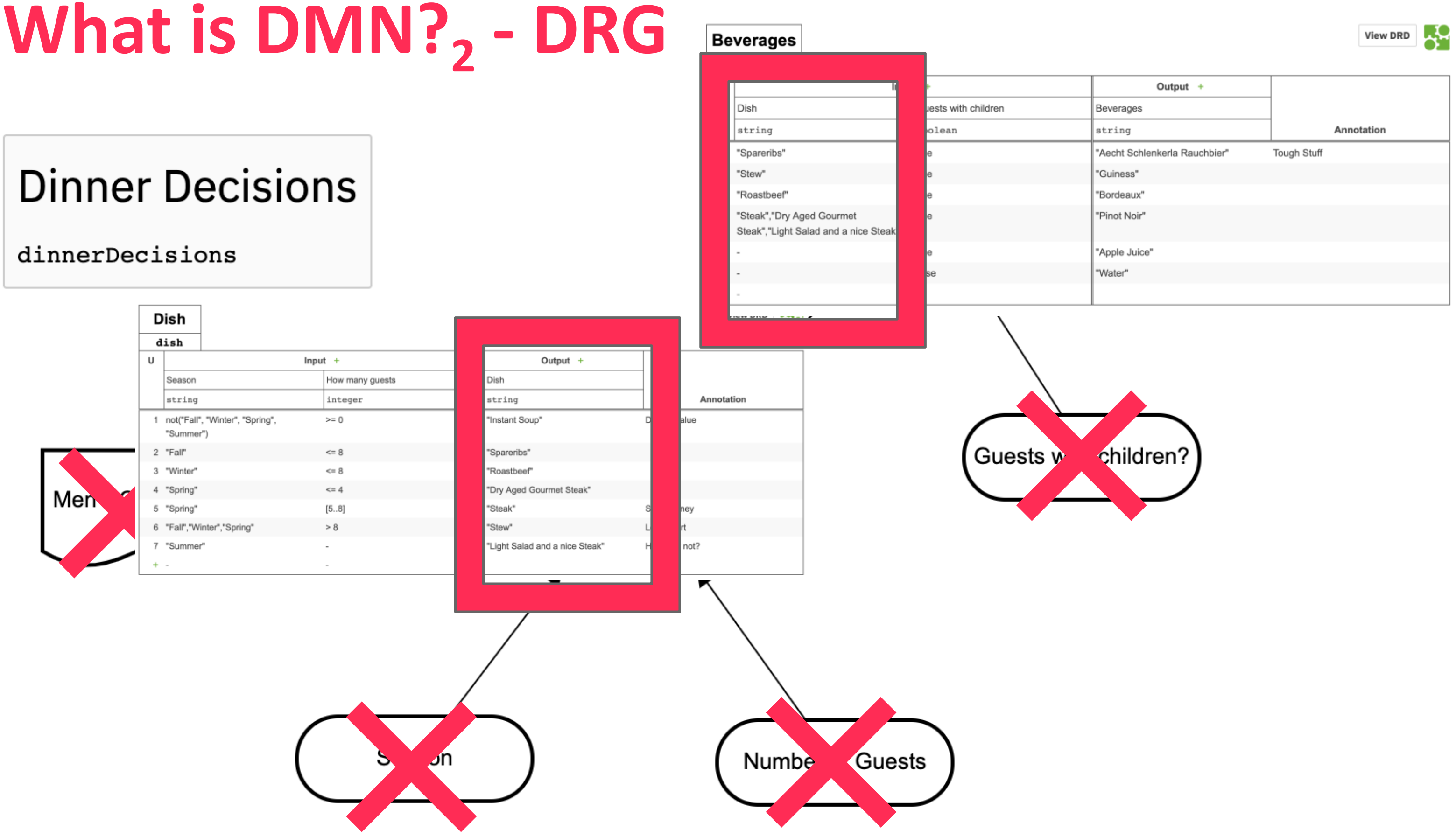
FEEL

OR

holisticon

https://camunda.com/dmn/simulator/

What is DMN?$_2$ - DRG



Dinner Decisions

dinnerDecisions

Beverages

	I	+	Output +	
	Dish	uests with children	Beverages	
	string	olean	string	Annotation
	"Spareribs"	e	"Aecht Schlenkerla Rauchbier"	Tough Stuff
	"Stew"	e	"Guiness"	
	"Roastbeef"	e	"Bordeaux"	
	"Steak","Dry Aged Gourmet Steak","Light Salad and a nice Steak	e	"Pinot Noir"	
	-	e	"Apple Juice"	
	-	se	"Water"	
	-			

Dish

dish

U	Input +		Output +	
	Season	How many guests	Dish	
	string	integer	string	Annotation
1	not("Fall", "Winter", "Spring", "Summer")	>= 0	"Instant Soup"	D alue
2	"Fall"	<= 8	"Spareribs"	
3	"Winter"	<= 8	"Roastbeef"	
4	"Spring"	<= 4	"Dry Aged Gourmet Steak"	
5	"Spring"	[5..8]	"Steak"	S ney
6	"Fall","Winter","Spring"	> 8	"Stew"	L rt
7	"Summer"	-	"Light Salad and a nice Steak"	H not?
+	-	-		

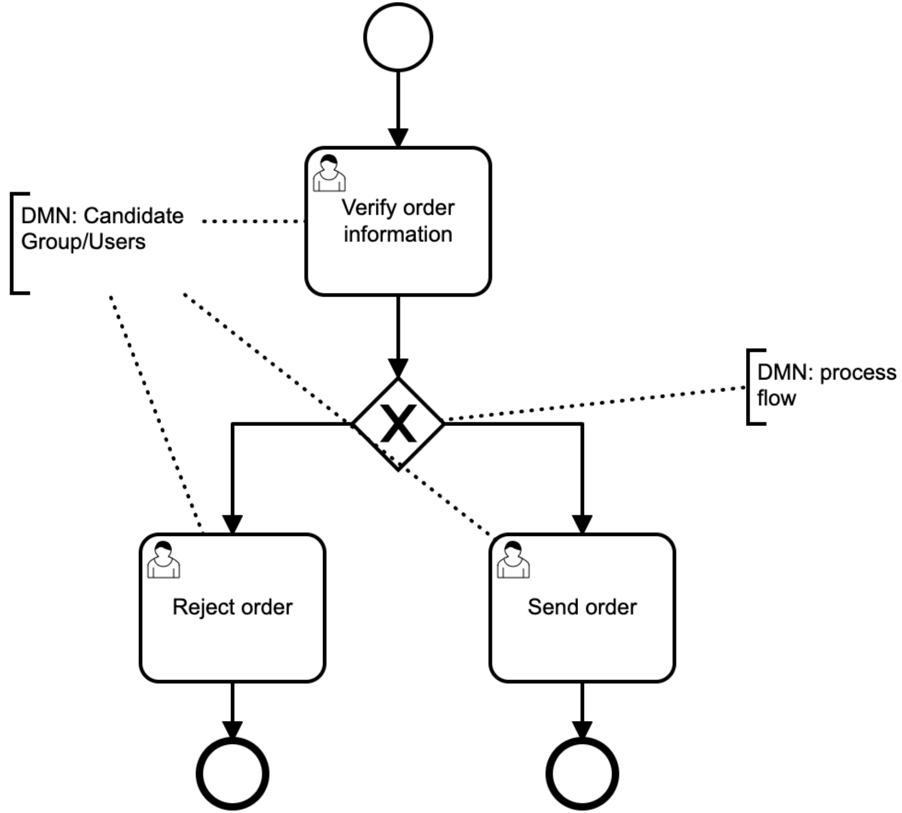
https://camunda.com/dmn/simulator/

Benefits of DMN

OMG Standard	No vendor lock-in
Business IT Alignment	Business can read and write decisions
Executable	No separation of build and runtime model
Transparency	What decision was made based on which input and when?
Flexibility	Change how your decisions are made, without downtime

Why this talk?

Project: BPMN

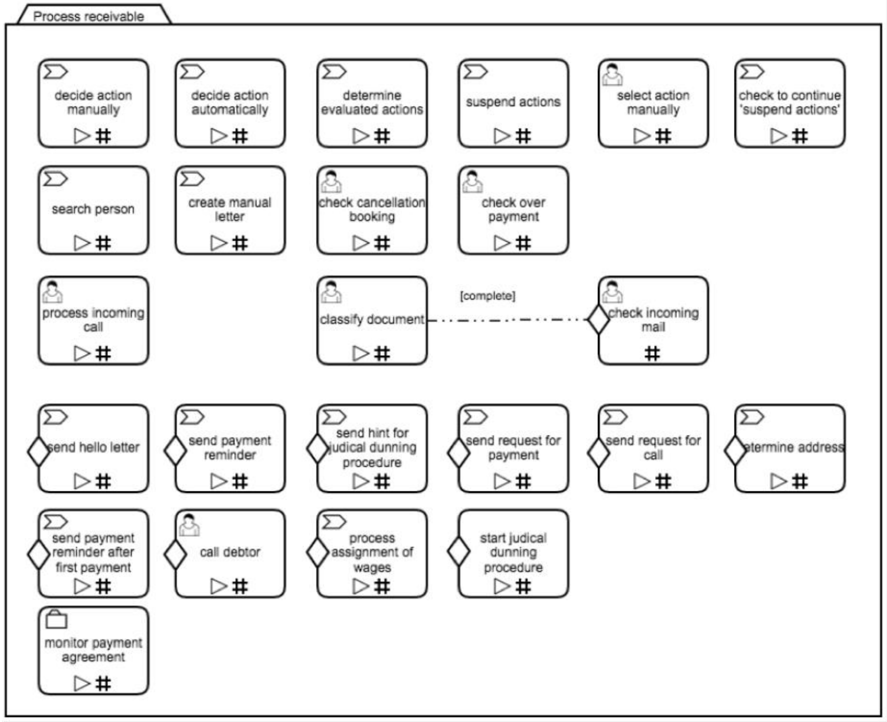


Motivation

- On Task create: evaluate DMN table
- On gateway decision: evaluate DMN table
- On Business Data change: reevaluate task assignment
- ~ 50 DMN tables (one per task)*
- Up to 1000 rows/table
- ~ 10k tasks per day*
- ~ 10 reassignments per day*

CamundaCon 2017
https://www.youtube.com/watch?v=5t_R6pFGk10
https://github.com/holunda-io/example-task-assignment-with-dmn

Project: CMMN



- ◇ SENTRY:
  Boolean Function (aka Predicate)
- Implemented by DMN
- ~ 100 Decision Tables
- Ø 20 inputs, 20 rows
- ~ 35M executions per day
- Multi-Tenant
- Experimental constraints

CamundaCon 2018
https://www.youtube.com/watch?v=QmacEUt7y_I&t=766s

5

Rules of Enterprise DMN

1

Make your rules explicit!

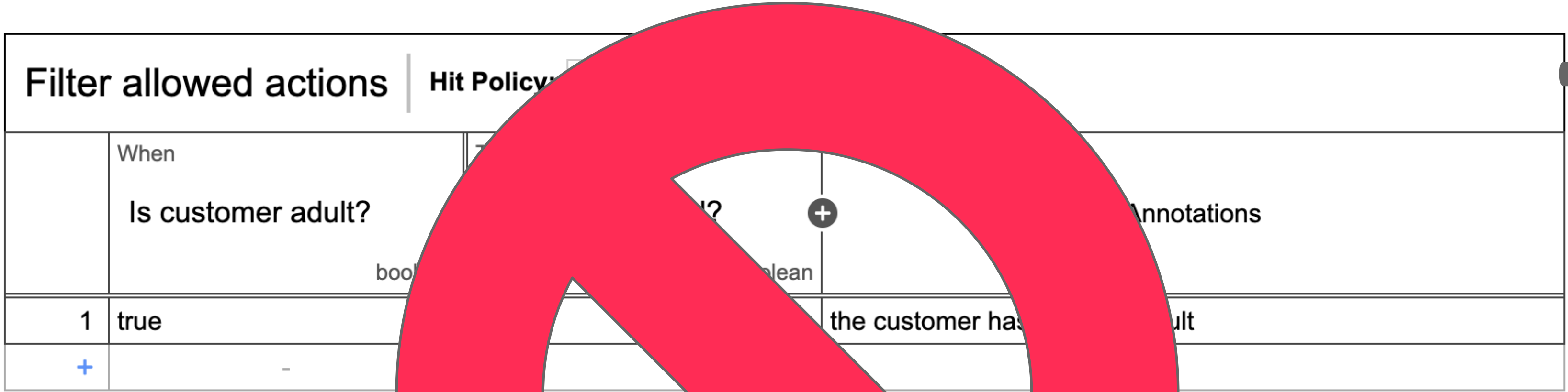
Make your rules explicit$_1$

1

- Requirement Engineering
  - □ ***„We must only execute this action, if the customer is an adult!“***
    - □ *„What does that mean?“*
  - □ *„Well, in Germany, this means, the person has to be at least 18 years old!“*
    - □ *„I thought 21?“*
  - □ *„No, they changed that 50 years ago.“*
    - □ *„OK, will do!“*

Make your rules explicit$_2$

1



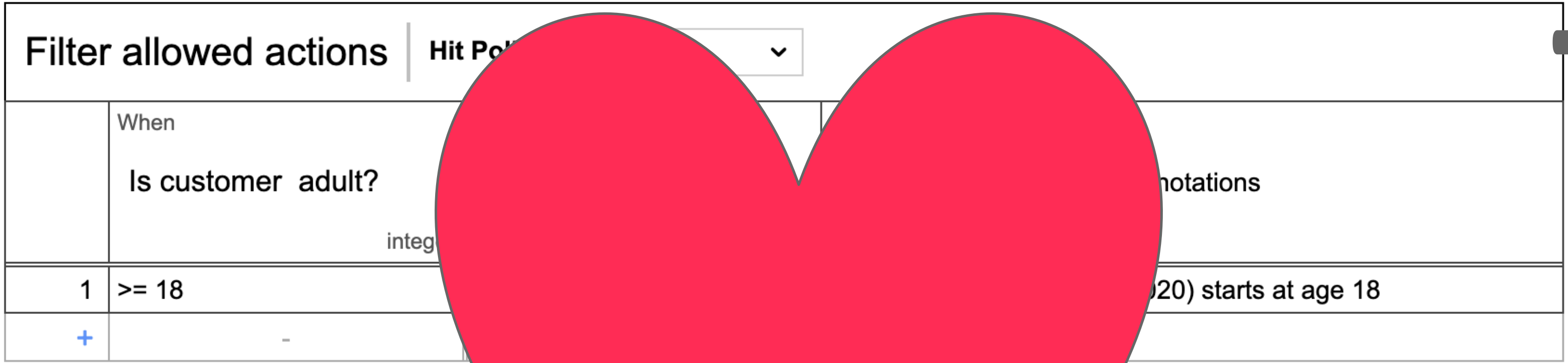
Filter allowed actions | Hit Policy:

	When Is customer adult? boo	? lean	Annotations
1	true		the customer has ult
+	-		

- Input: interpreted
- Business relevant c ison done b ce
- Runtime: *„Hey Jan, w wed?“* – *„I will have to check with e isAdultSubComponent!“* – *„Stay until this is solved!“*

Make your rules explicit$_3$

1



Filter allowed actions | Hit Po ⌄

	When		notations
	Is customer adult? integ		
1	>= 18		20) starts at age 18
+	-		

- Input: age of custome
- Comparison done by dec
- Runtime: *„Hey Jan, why was* *ot allowed?“ –* *„Well the rule said, he had to be 1* *ut he was only 17!“ –* *„Thanks, Jan! Have a nice weekend!“*

2

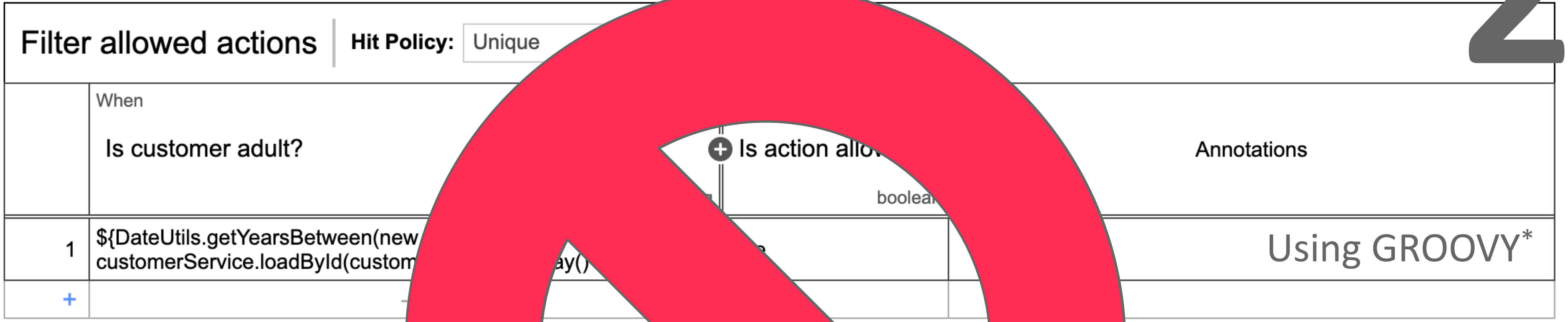
Use FEEL!

Use FEEL$_1$

2

- Requirement Engineering
  - □ ***„We must only execute this action, if the customer is an adult!“***
    - □ *„I have an idea! Camunda supports multiple expression languages in decision tables!“*
  - □ *„I trust your judgement“*

Use FEEL$_2$



Filter allowed actions | **Hit Policy:** Unique

	When Is customer adult?	Is action allo boolea		Annotations
1	${DateUtils.getYearsBetween(new customerService.loadById(custom ay()			Using GROOVY*
+				

- Input: customerI eed for a l ser
- Business relevant c ison done b e in decision table
- Runtime: *„Hey Jan, why ? It clearly says: 18“ – „Hm. Some Bug with our D s birthday is ok! “ – „Stay until this is solved!“*

Use FEEL$_3$ - Benefits

2

Simple	No complex calculations, only comparison
Business IT alignment	Business can **read*** and write* decisions
Execution	Pure functions – DMN as „black box“

holisticon

3

DMN != <code>

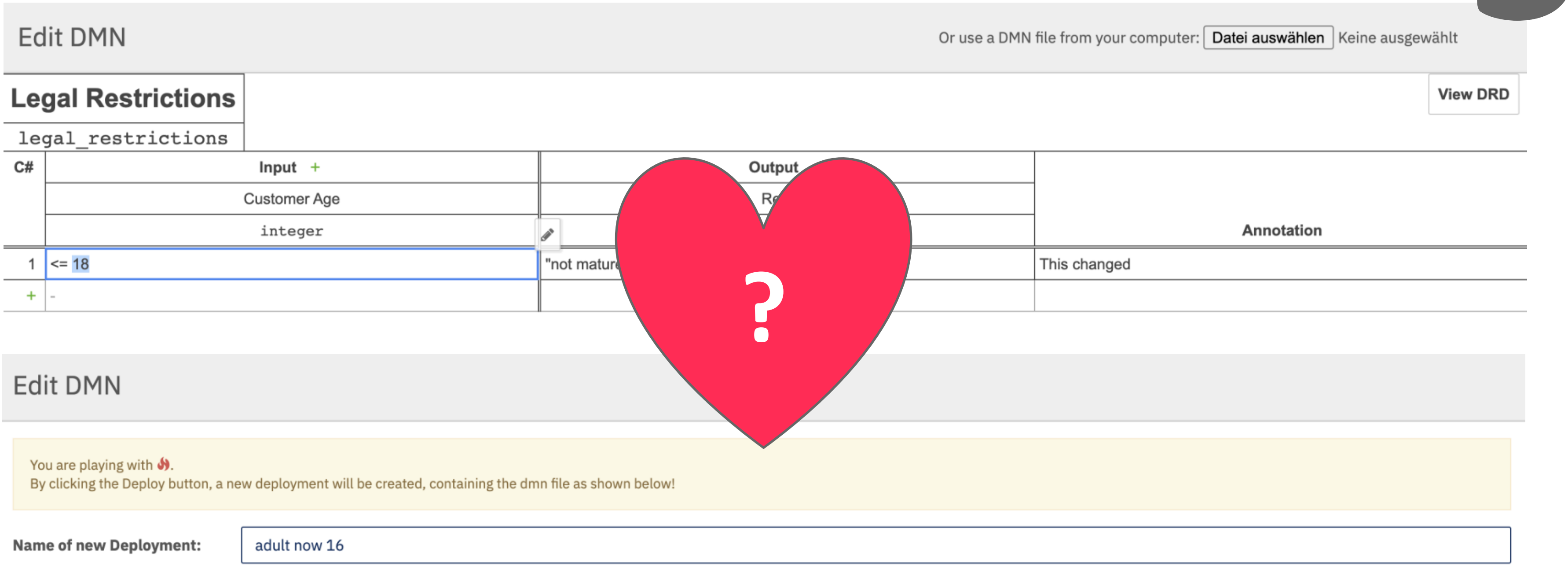
DMN != <code>$_1$

3

- Requirement Engineering
  - □ ***„We have a new government. Adulthood now starts at age 16! What do we do“***
    - □ *„ Too bad. We have our DMN files in `src/main/recources`, like our BPMN files. We will have to alter them and do a deployment.“*
  - □ *„Can you do it, please?“*
    - □ *„Of course! File a jira issue, we have a deployment scheduled for next Thursday anyway!“*
  - □ *„I need it now!“*
    - □ *„Wait a minute … we got an enterprise license, right?“*

DMN != <code>$_2$

3



Edit DMN

Or use a DMN file from your computer: Datei auswählen Keine ausgewählt

Legal Restrictions

`legal_restrictions`

View DRD

C#	Input +	Output	
	Customer Age	R	
	integer		Annotation
1	<= 18	"not matur	This changed
+	-		

Edit DMN

You are playing with 🔥.
By clicking the Deploy button, a new deployment will be created, containing the dmn file as shown below!

Name of new Deployment: adult now 16

DMN != <code>$_3$

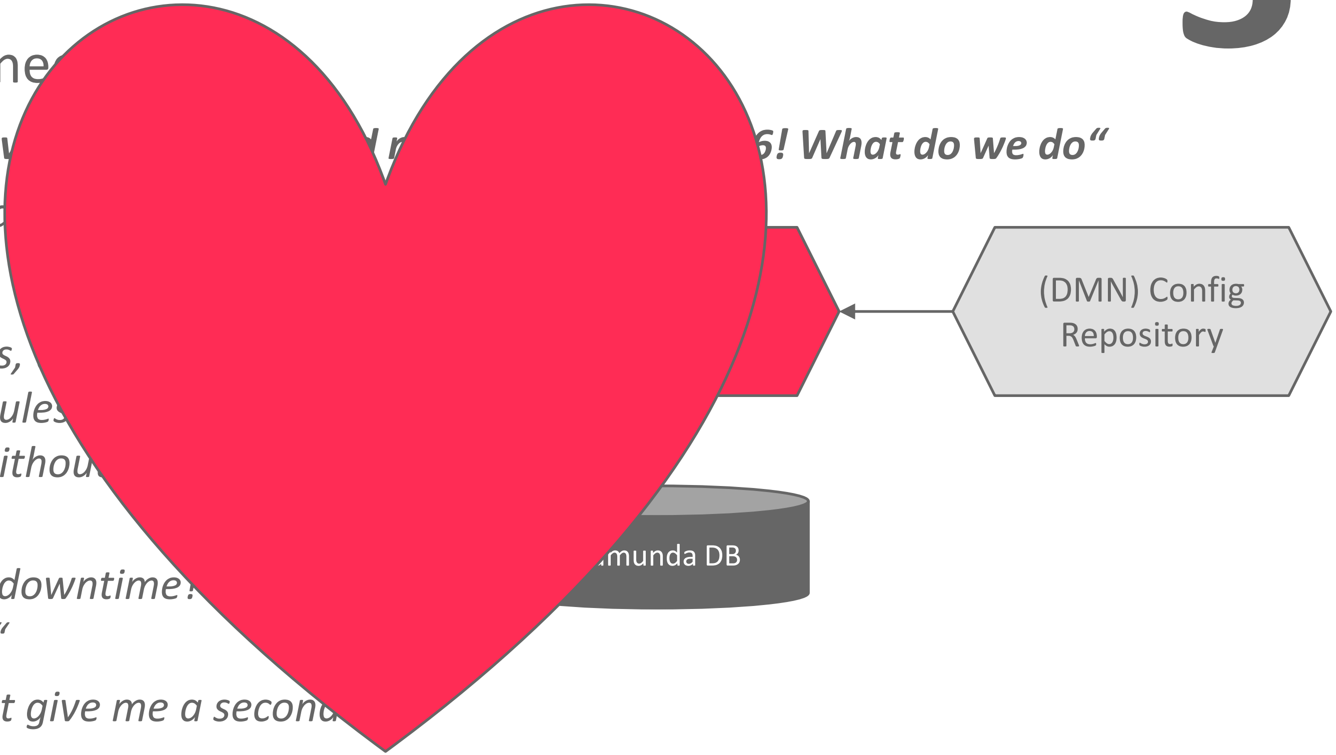
3

- Next Thursday
  - □ *„Major incident! You h* *8 again to be* *Vhat happened!“*
    - □ *„We did the sche* *this morning.*
      And as I said: th *les* *f our code b* *e deployed them over your hot fixes!“*
  - □ *„That‘s going to be* *y, my frie*

DMN != <code>$_4$

3

- Requirement Engine
  - *„We have a new gov l r 6! What do we do“*
    - *„Did the required*
  - *„No!“*
    - *„Well, good news,*
      and deploy our rules
      configuration, withou
      core app.“
  - *„Will that require a downtime?*
    inform operations?“
    - *„ No need to, just give me a second*
  - *„Well done, Jan!“*



holisticon

4

Generate DMN!

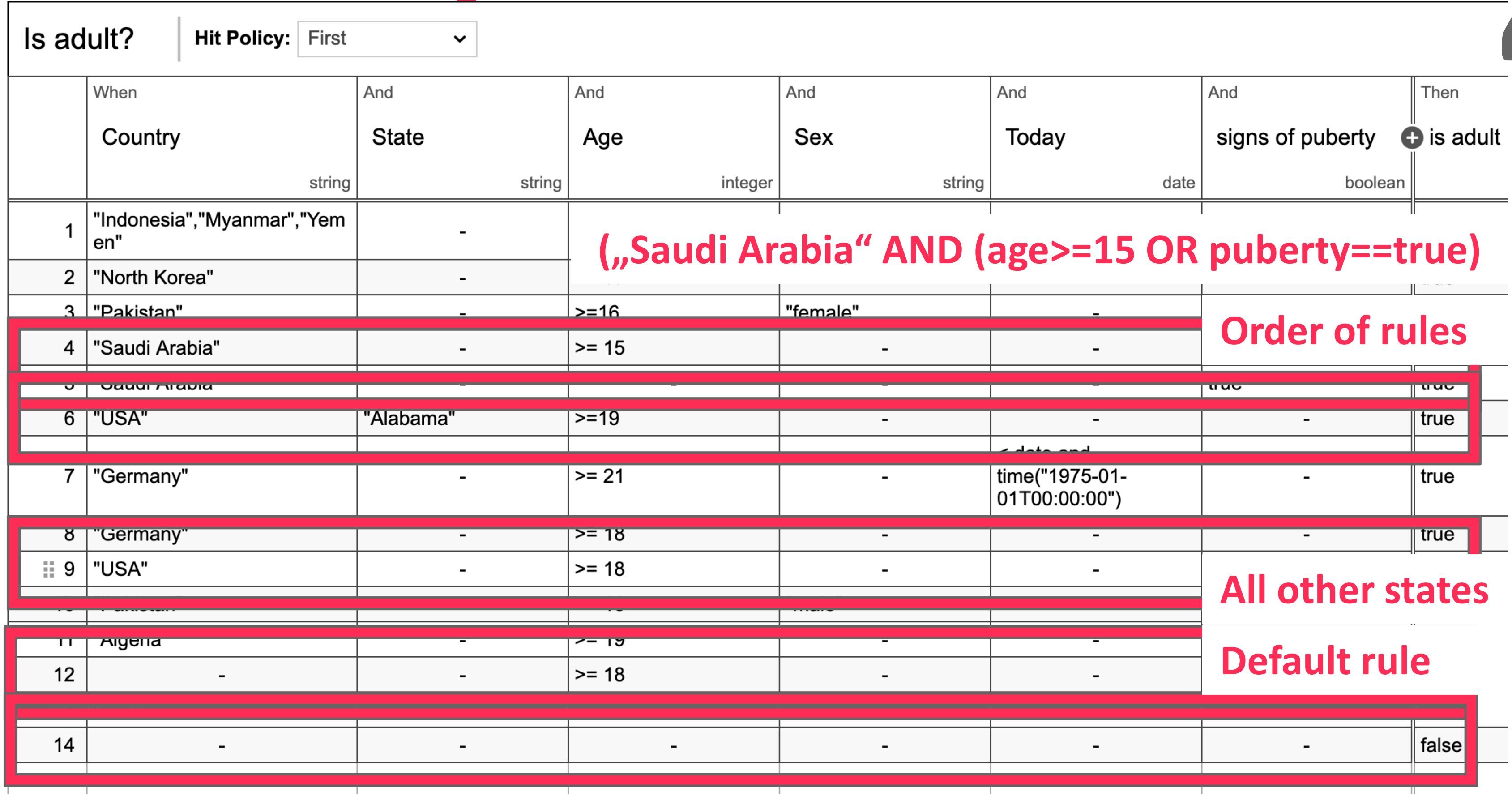
Generate DMN$_1$

4

- Requirement Engineering
  - ***„Remember that little DMN project we set up last year?“***
    - *„Of course! Great success, right? “*
  - *„Well, kind of … we expanded our business, now look at this …“*

Generate DMN_2

4



Is adult? **Hit Policy:** First

	When Country string	And State string	And Age integer	And Sex string	And Today date	And signs of puberty boolean	Then is adult
1	"Indonesia","Myanmar","Yemen"	-					
2	"North Korea"	-					
3	"Pakistan"	-	>=16	"female"	-		
4	"Saudi Arabia"	-	>= 15	-	-		
5	"Saudi Arabia"	-	-	-	-	true	true
6	"USA"	"Alabama"	>=19	-	-	-	true
7	"Germany"	-	>= 21	-	< date and time("1975-01-01T00:00:00")	-	true
8	"Germany"	-	>= 18	-	-	-	true
9	"USA"	-	>= 18	-	-		
10	"Pakistan"		18	"male"			
11	"Algeria"	-	>= 19	-	-		
12	-	-	>= 18	-	-		
14	-	-	-	-	-	-	false

(„Saudi Arabia“ AND (age>=15 OR puberty==true)

Order of rules

All other states

Default rule

https://en.wikipedia.org/wiki/Age_of_majority

Generate DMN$_3$

4

Business IT alignment

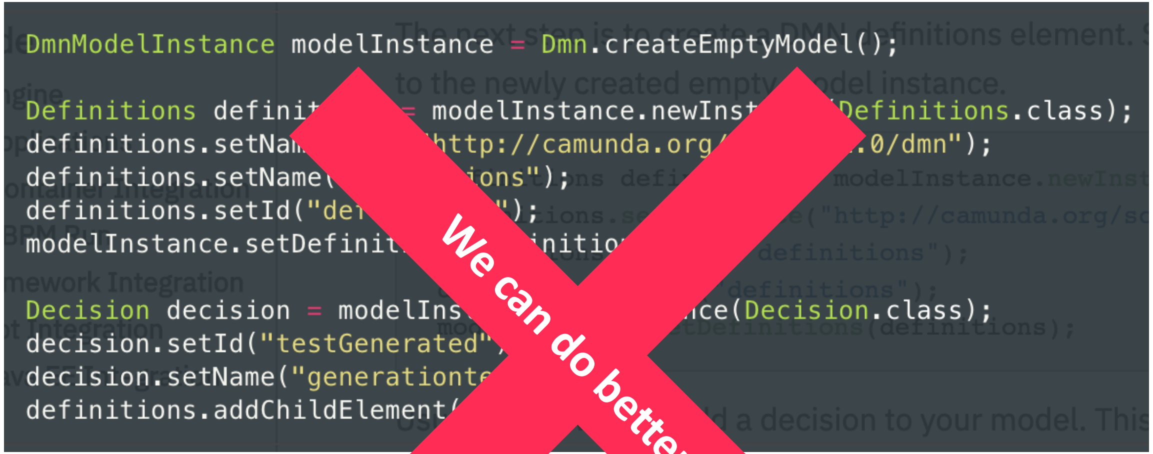
Business can read and write decisions ?

trivial

- Complex tables?
- Large tables?
- Complex boolean logic?
- Hit Policies?
- Rule ordering?

Generate DMN$_4$

4



```
DmnModelInstance modelInstance = Dmn.createEmptyModel();

Definitions definit          = modelInstance.newInst      Definitions.class);
definitions.setNa             http://camunda.org/        .0/dmn");
definitions.setName(              ions");
definitions.setId("de              ");
modelInstance.setDefinit          initio

Decision decision = modelIns               ce(Decision.class);
decision.setId("testGenerated")
decision.setName("generationte
definitions.addChildElement(
```

We can do better

- Camunda can generate d   on tables
  - □ But not decision graphs
- Unintuitive direct XML API

5

Divide and conquer

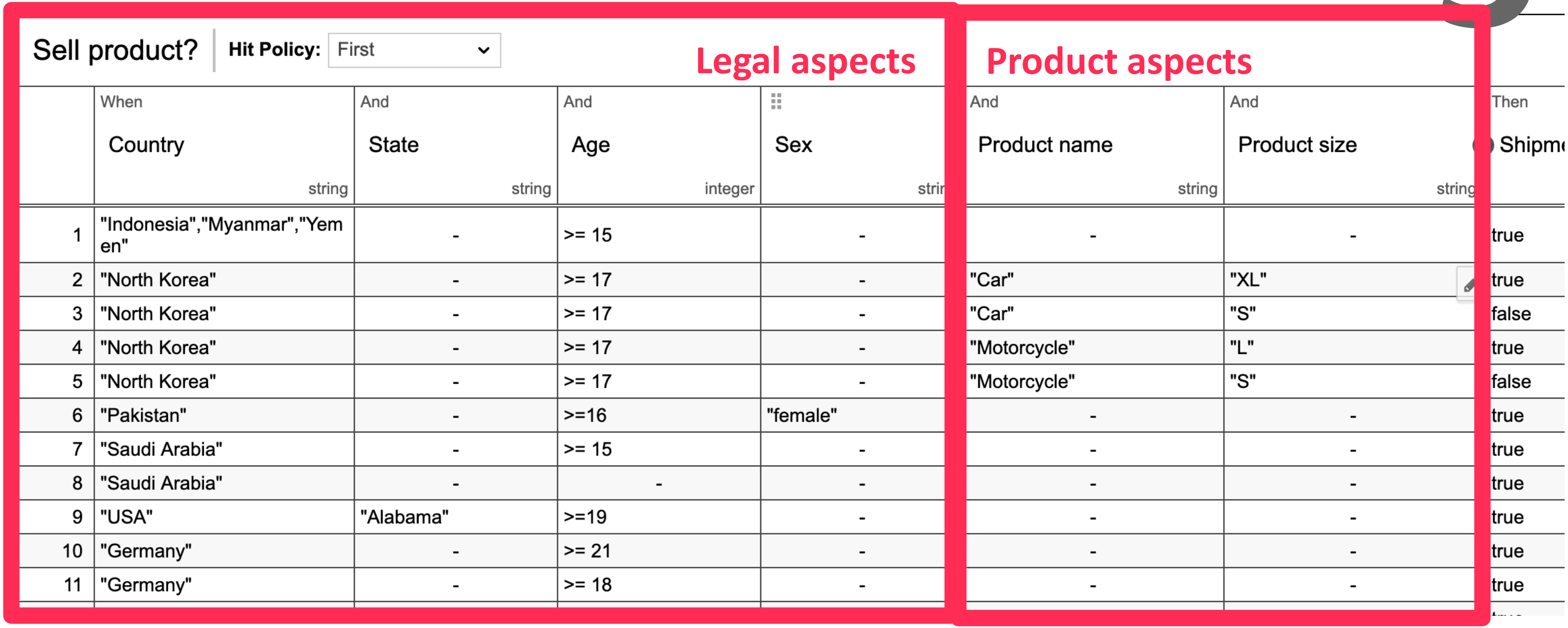
Divide and conquer$_1$

5

- Requirement Engineering
  - □ ***„OK, we got generators in place, but it‘s still hard to manage the DMN tables!“***
    - □ *„Why so?“*
  - □ *„Well, turned out is customer adult was not our only concern …“*

Divide and conquer$_2$

5



Sell product? | Hit Policy: First

Legal aspects

Product aspects

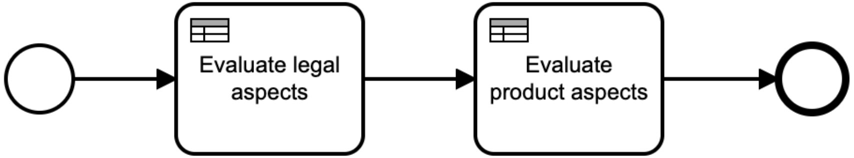
	When Country string	And State string	And Age integer	Sex stri	And Product name string	And Product size string	Then Shipme
1	"Indonesia","Myanmar","Yemen"	-	>= 15	-	-	-	true
2	"North Korea"	-	>= 17	-	"Car"	"XL"	true
3	"North Korea"	-	>= 17	-	"Car"	"S"	false
4	"North Korea"	-	>= 17	-	"Motorcycle"	"L"	true
5	"North Korea"	-	>= 17	-	"Motorcycle"	"S"	false
6	"Pakistan"	-	>=16	"female"	-	-	true
7	"Saudi Arabia"	-	>= 15	-	-	-	true
8	"Saudi Arabia"	-	-	-	-	-	true
9	"USA"	"Alabama"	>=19	-	-	-	true
10	"Germany"	-	>= 21	-	-	-	true
11	"Germany"	-	>= 18	-	-	-	true

Divide and conquer$_3$

5

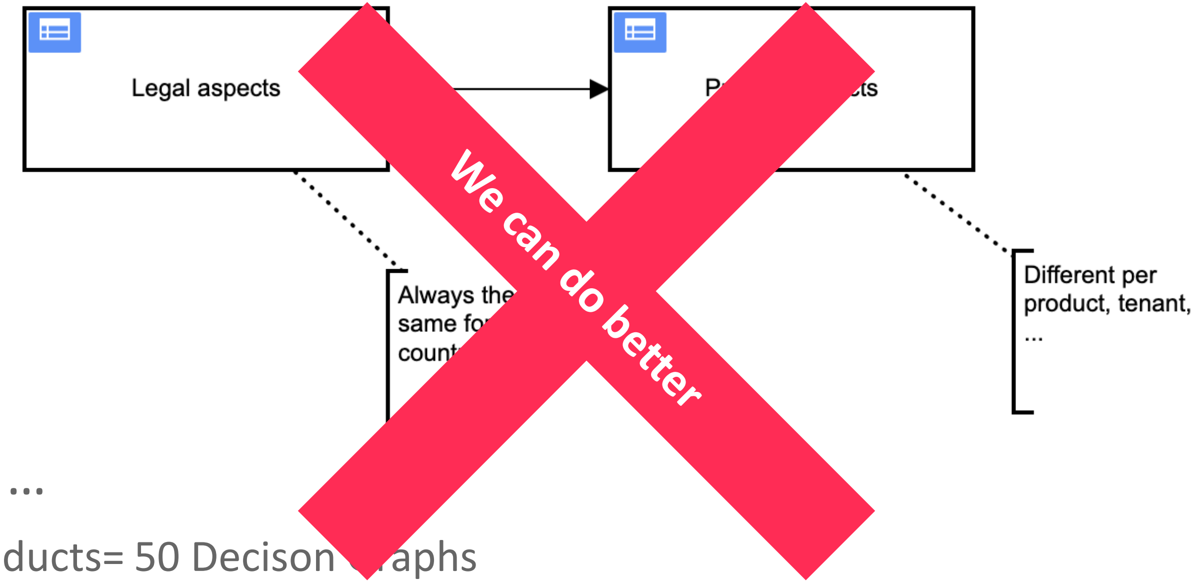
- Options to split large tables into smaller ones
  - □ One table per region (`sellproduct_germany.dmn`, `sellproduct_korea.dmn`)
  - □ Graph of multiple tables: first legal aspects, then product aspects
  - □ Combined approach

- Options to execute chained tables
  - □ Create a custom BPMN evaluation process
  - □ Use DMN Decision Requirement Graphs (DRG)



Divide and conquer$_4$

5



- OK, but …
  - □ 50 Products= 50 Decison Graphs
  - □ Camunda does not support table references = 50 copies of the legal aspects table
  - □ When legal aspects change, you have to edit 50 tables

Ramp up

Ramp up

- Make your rules explicit
  - Context just delivers data
  - DMN does the comparison
- Use Feel
  - Keep it simple
- DMN != <code>
  - Do not put *.dmn files in `src/main/resources`
- Generate DMN
  - Computers are better at modifying large tables

- Divide and Conquer
  - Multiple small tables are easier to handle than one large table
  - Use separate files for separate concerns
  - Generate DRGs

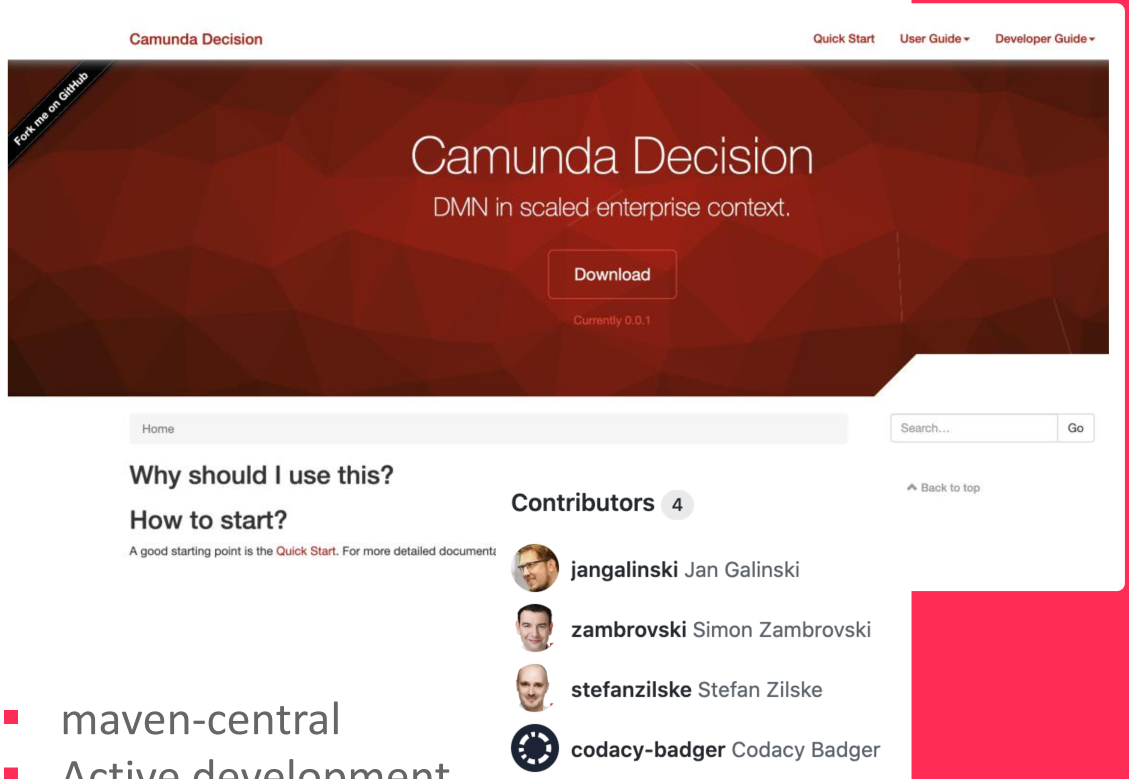
- Bonus
  - Decision History grows fast. Use TTL, think about clean up strategies
  - Consider how you want to test your DMNs …

Demo

extension: camunda-decision

holunda
happy paths with heart

camunda-decision



Camunda Decision

DMN in scaled enterprise context.

Why should I use this?

How to start?

A good starting point is the Quick Start. For more detailed documenta

Contributors 4

- **jangalinski** Jan Galinski
- **zambrovski** Simon Zambrovski
- **stefanzilske** Stefan Zilske
- **codacy-badger** Codacy Badger

- maven-central
- Active development
- Contributions welcome

SCAN ME

https://www.holunda.io/camunda-decision/

What next?

Roadmap

Buildtime

- Property based testing
  ensure the rules are correct for all possible inputs
- Validation
  using dmnCheck
- Schema based Query/Result Objects
  Becoming full type safe, generate code
- Support complex input objects
  currently limited to primitives

Runtime

- Caching
  do not recalculate when nothing changed
- Distributed Systems
  event driven architecture/axon
- Simulation
  How would the system behavior change?
- Auditing/Cockpit
  better tracing/tracking of decision evaluations

Documentation, Documentation, Documentation

Have a look at …

dmn-check

- DMN File validation
  - □ Duplicate rules
  - □ Conflicting rules
  - □ Shadowed rules
  - □ Types of the expressions
  - □ Correct use of enumerations
  - □ Correctly connected requirement graphs

https://github.com/red6/dmn-check

Star

dmnmgr

- Camunda Modeler Extension
  - □ Define input/output via OpenAPI
  - □ Git repo as DMN store
  - □ Validation (using dmn-check)
  - □ Simulation
  - □ Client/Server architecture

https://github.com/davidibl/dmnmgr-server

CAMUNDA
CON
2020.2

Questions?