Red Hat
Process Automation

Software and Services for Business Automation

PV207 BPM

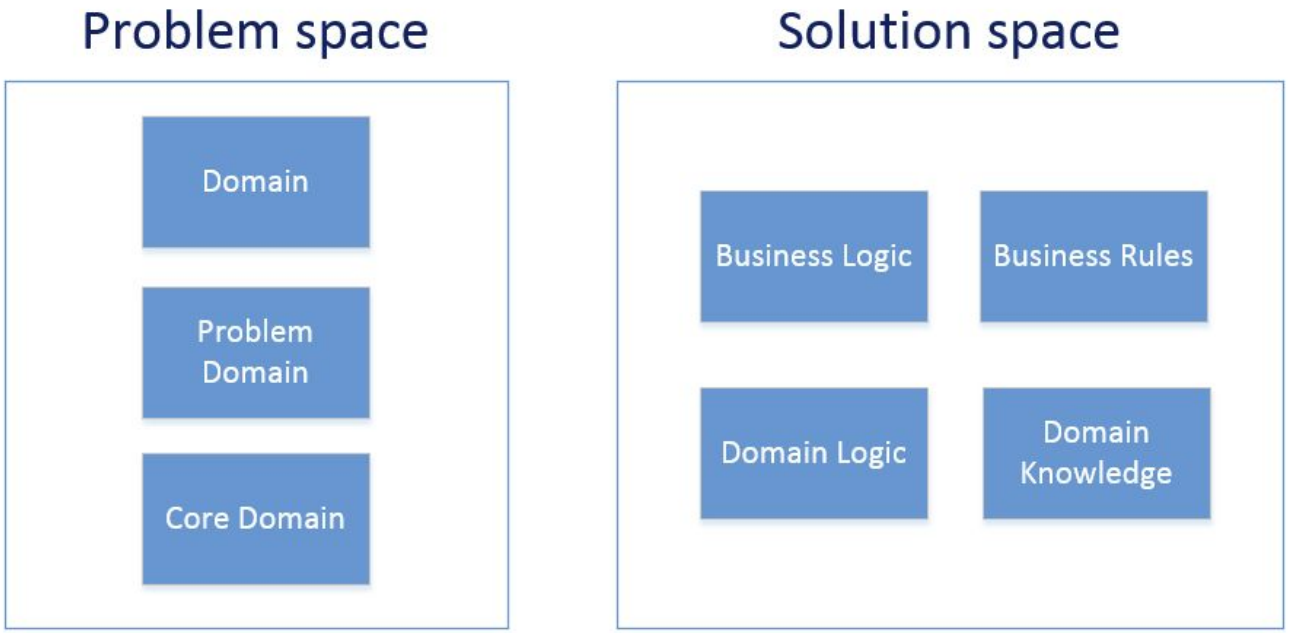
Mgr. Marian Macik
Senior Software Quality Engineer

April 2023

Red Hat

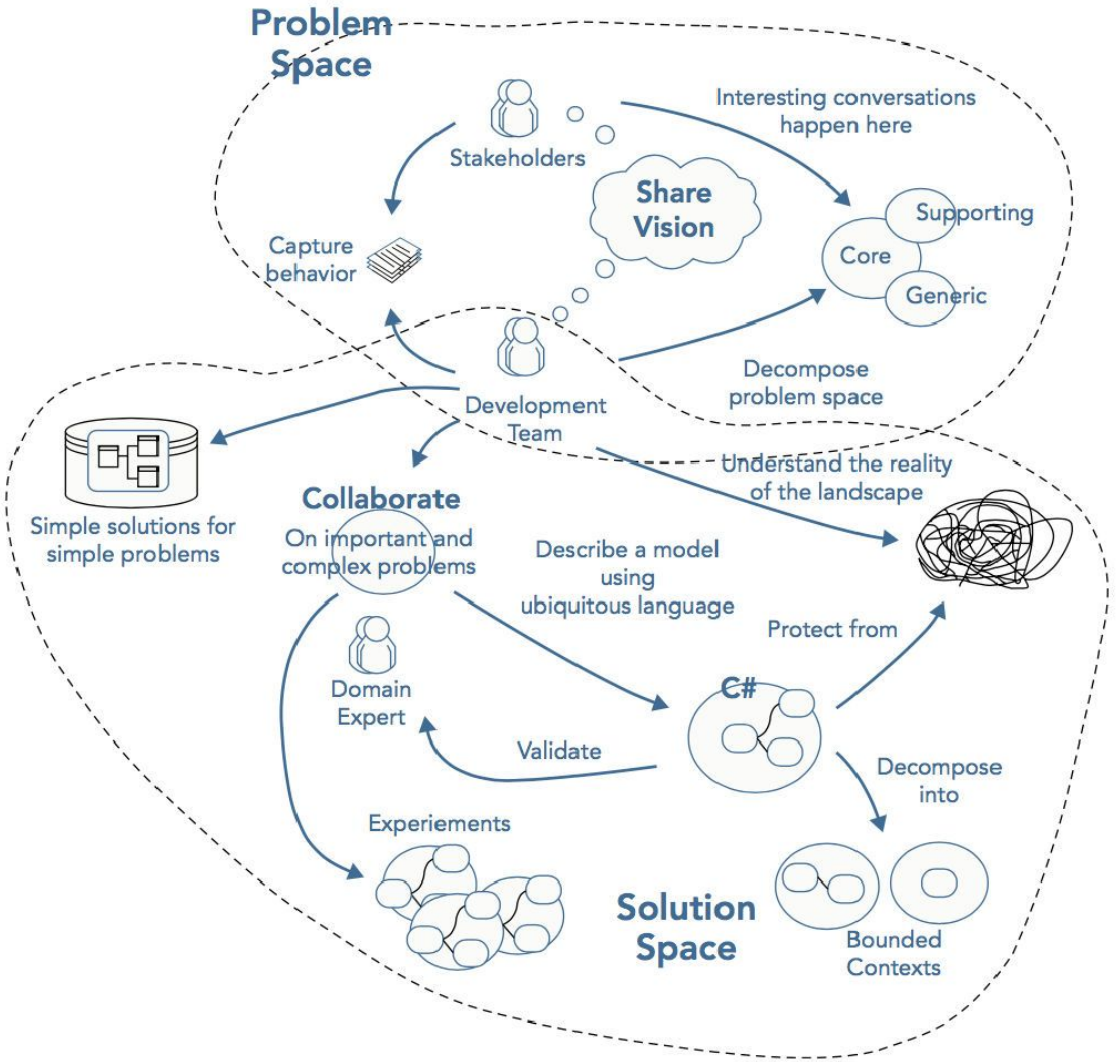
DOMAIN-DRIVEN DESIGN (DDD)

Domain-driven design is a concept that the structure and language of software code should match the business domain.



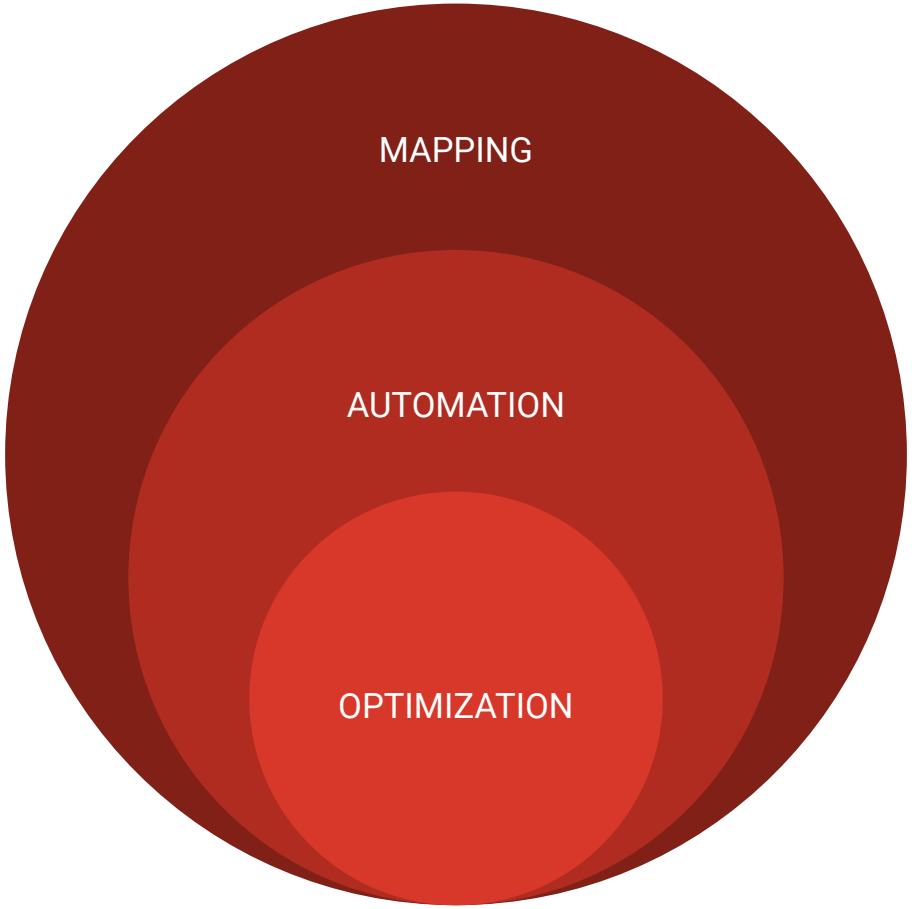
- Eases communication
- Improves flexibility
- Emphasizes domain over interface
- Requires robust domain expertise
- Encourages iterative practices
- Ill-suited for highly technical projects

DOMAIN-DRIVEN DESIGN (DDD)



Red Hat

BUSINESS AUTOMATION



Red Hat

MARKET

Low Code Platforms

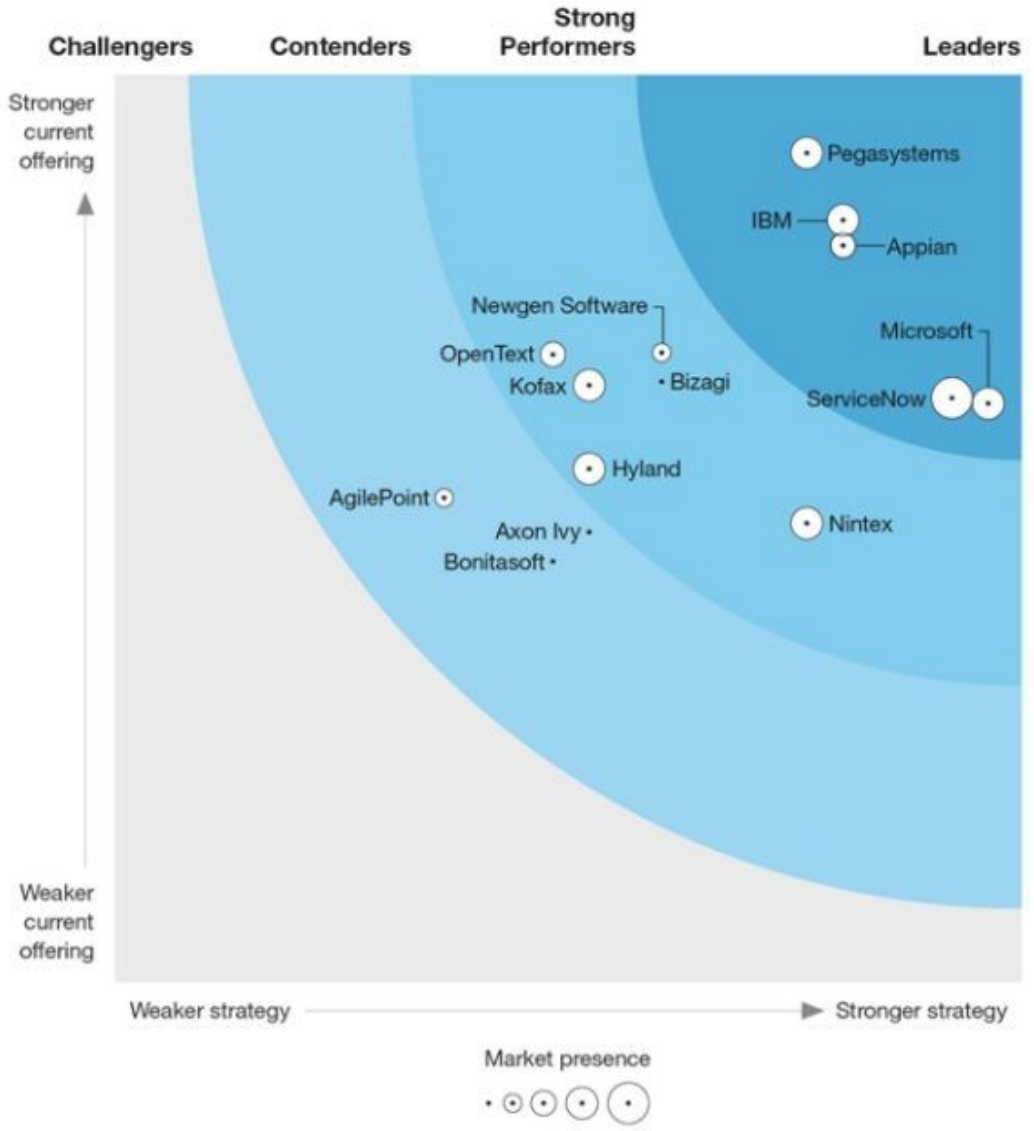
DPA Platforms

Cloud Services

Communities

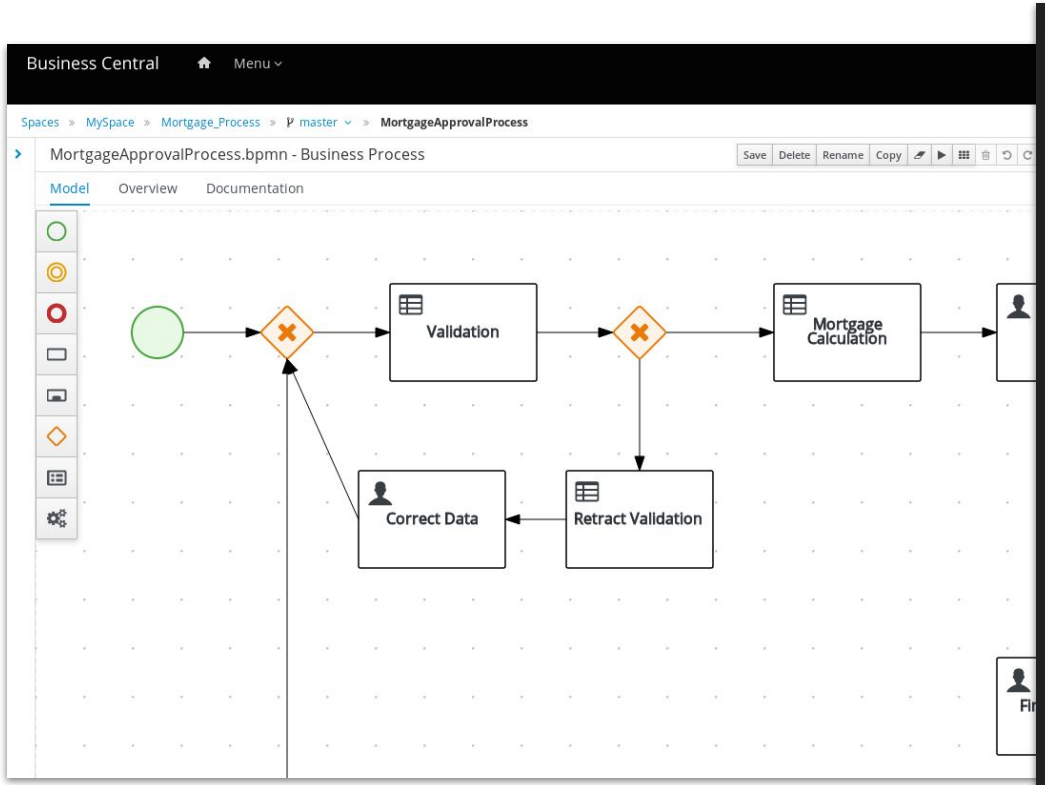
(Object Management Group, Cloud Native Computing Foundation, Kubernetes, Knative, KIE, Kogito)

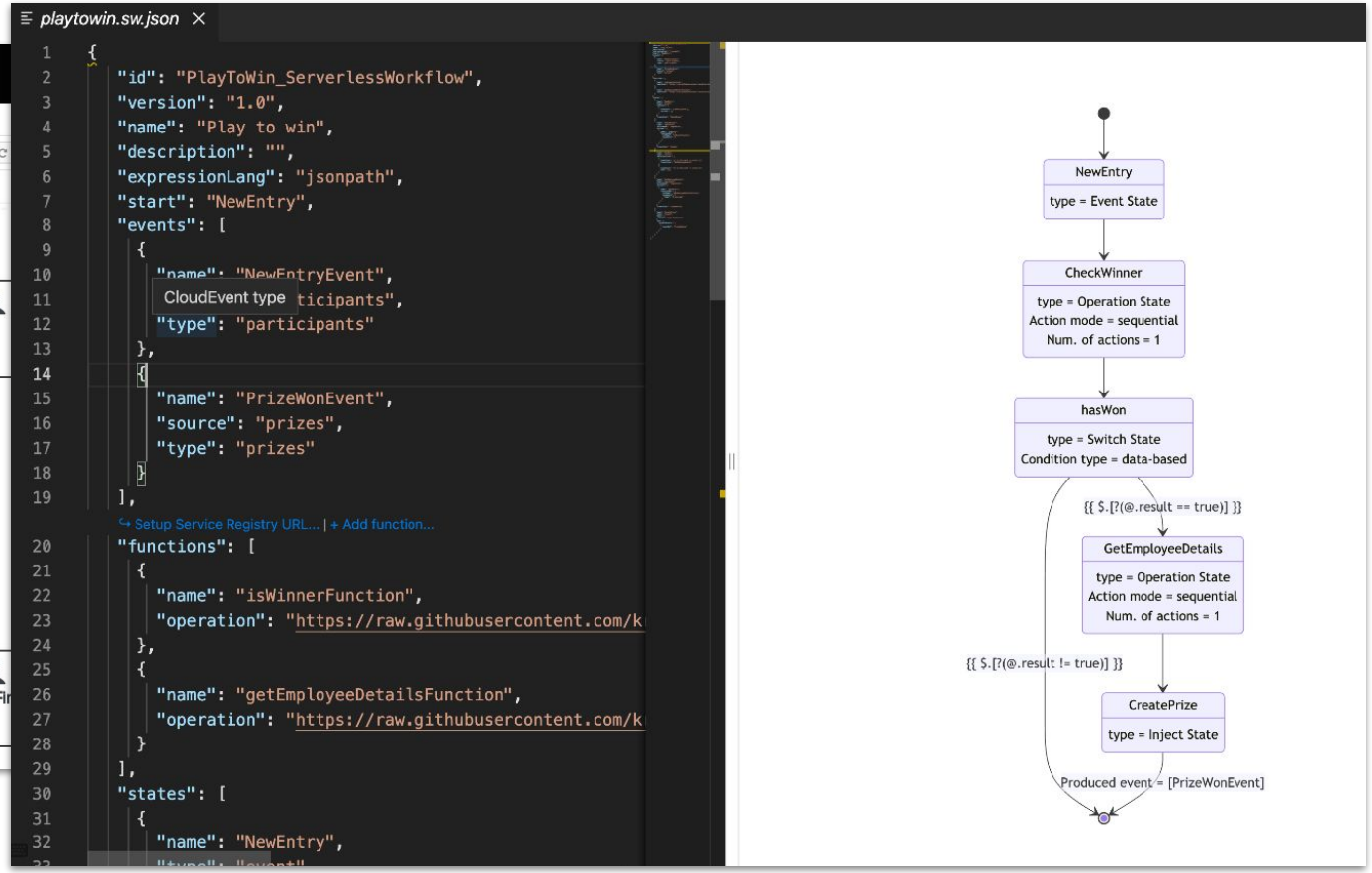
DPA SOFTWARE FROM FORRESTER ANALYST REPORT ‘21



Red Hat

AUTOMATION OF WORKFLOWS



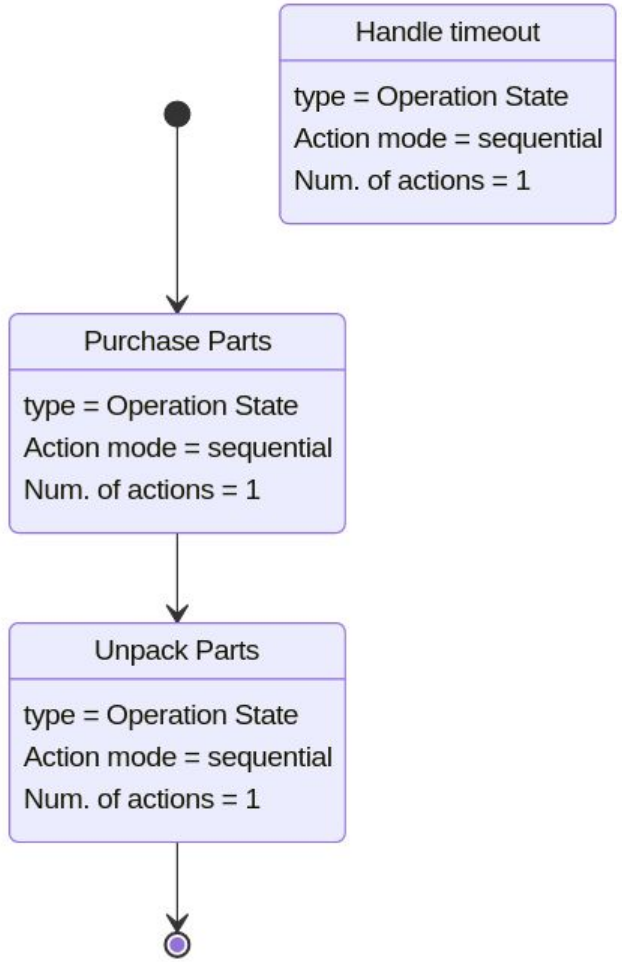


Red Hat

SERVERLESS WORKFLOW

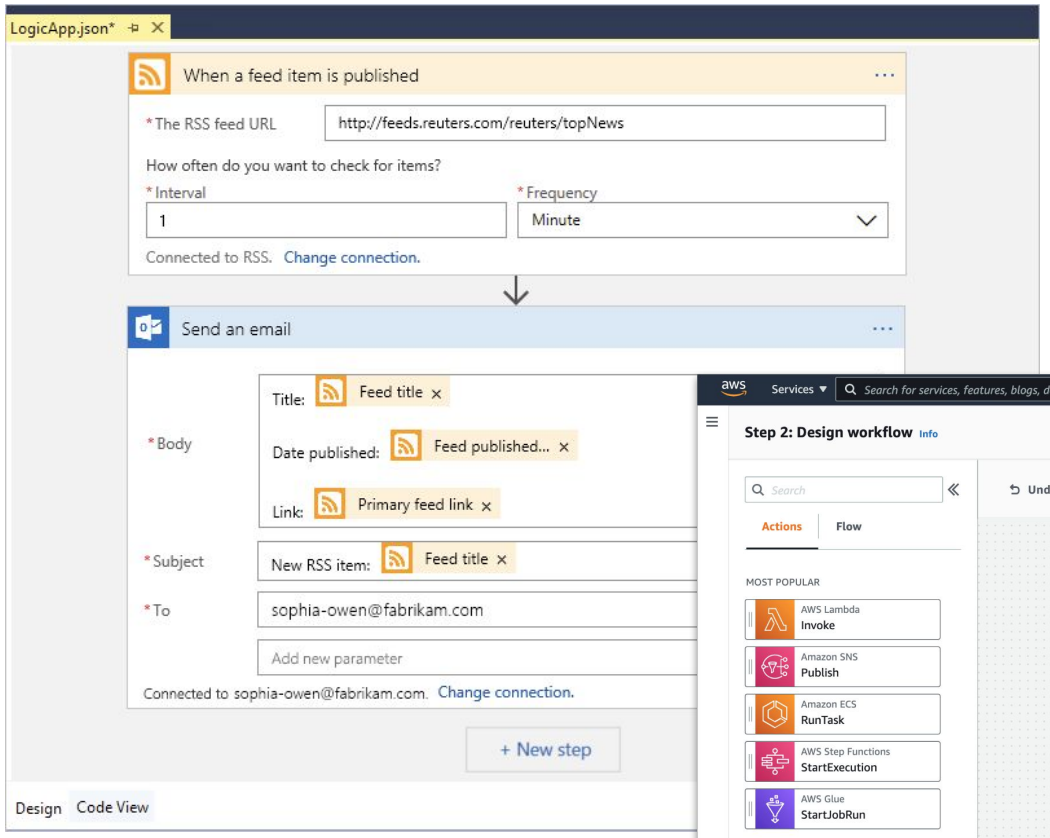
- Similar specification to BPMN but aimed more at developers
- More lightweight, no human tasks, targeted at machine interaction
- Workflow is “modelled” directly by writing the code (JSON/YAML), currently doesn’t define graphical notation
- Higher level than BPM, focuses more on orchestration of services or infrastructure in the cloud, self-healing infrastructure
- Background service, end users don’t interact directly with it (as with BPM in loan approval, order processing…)
- https://serverlessworkflow.io/
- BPMN vs. SWF

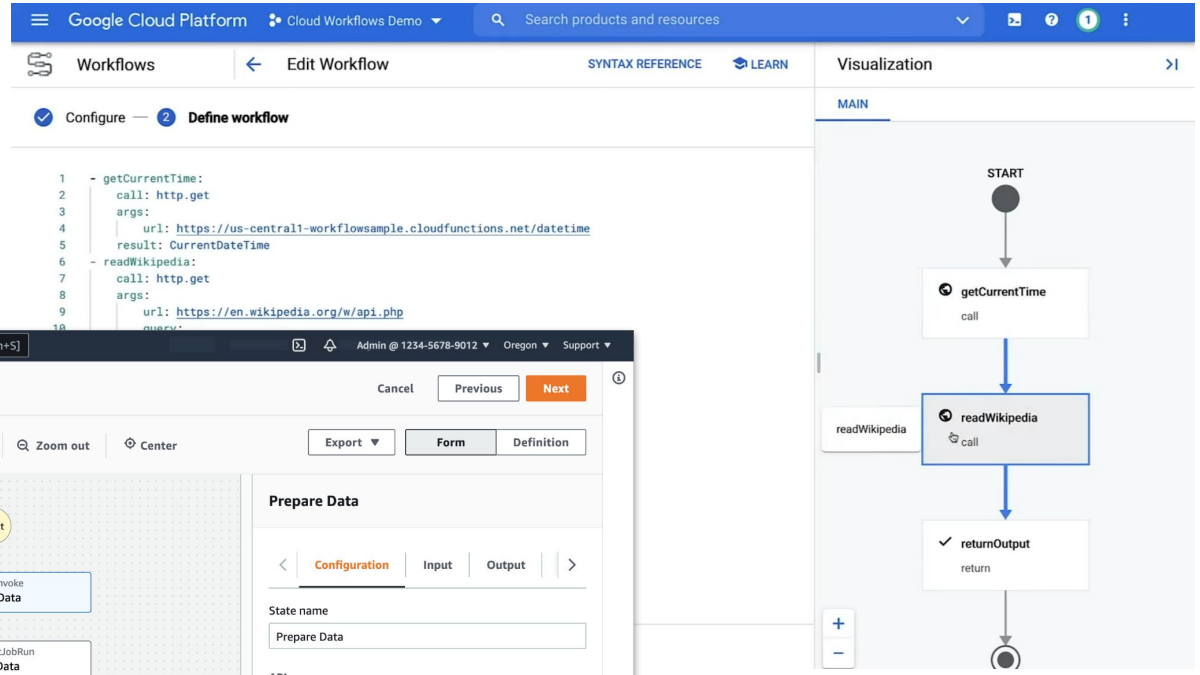
```yaml
id: executiontimeout
name: Execution Timeout Workflow
version: '1.0.0'
specVersion: '0.8'
start: Purchase Parts
timeouts:
  workflowExecTimeout:
    duration: PT7D
    interrupt: true
    runBefore: Handle timeout
states:
- name: Purchase Parts
  type: operation
  actions:
  - functionRef: purchasePartsFunction
  transition: Unpack Parts
- name: Unpack Parts
  type: operation
  actions:
  - functionRef: unpackPartsFunction
  end: true
- name: Handle timeout
  type: operation
  actions:
  - functionRef: handleTimeoutFunction
functions:
- name: purchasePartsFunction
  operation: file://myservice.json#purchase
- name: unpackPartsFunction
  operation: file://myservice.json#unpack
- name: handleTimeoutFunction
  operation: file://myservice.json#handle
```

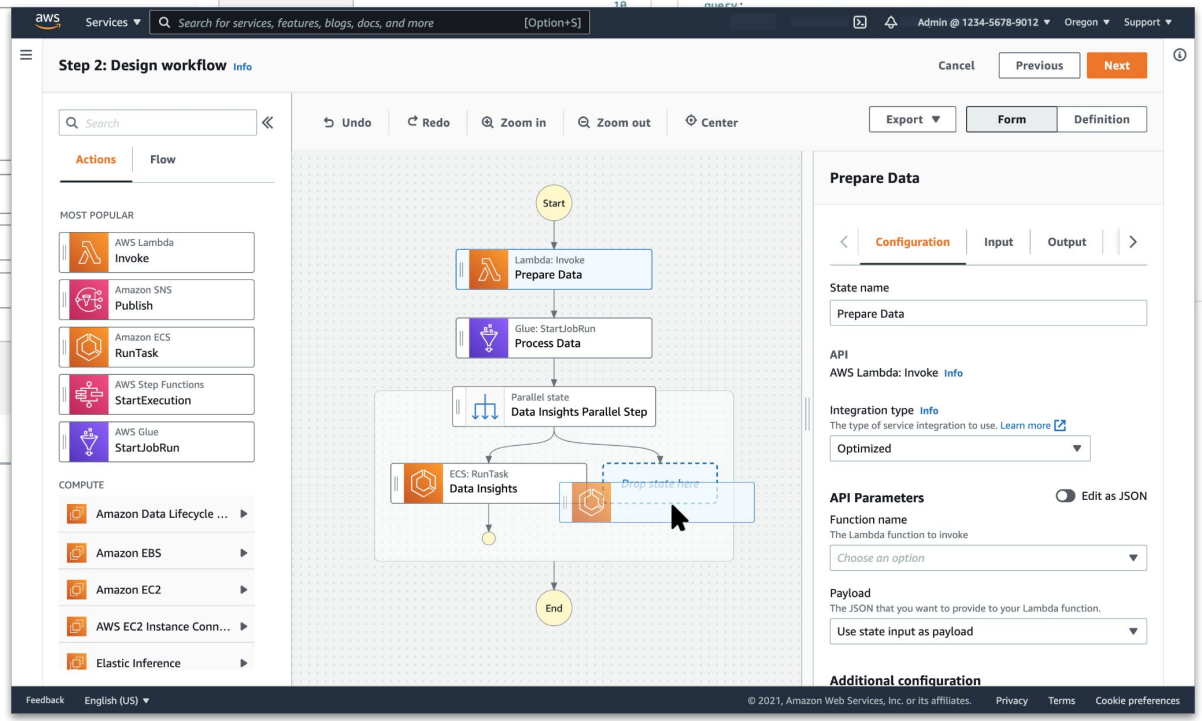


Red Hat

AUTOMATION OF WORKFLOWS

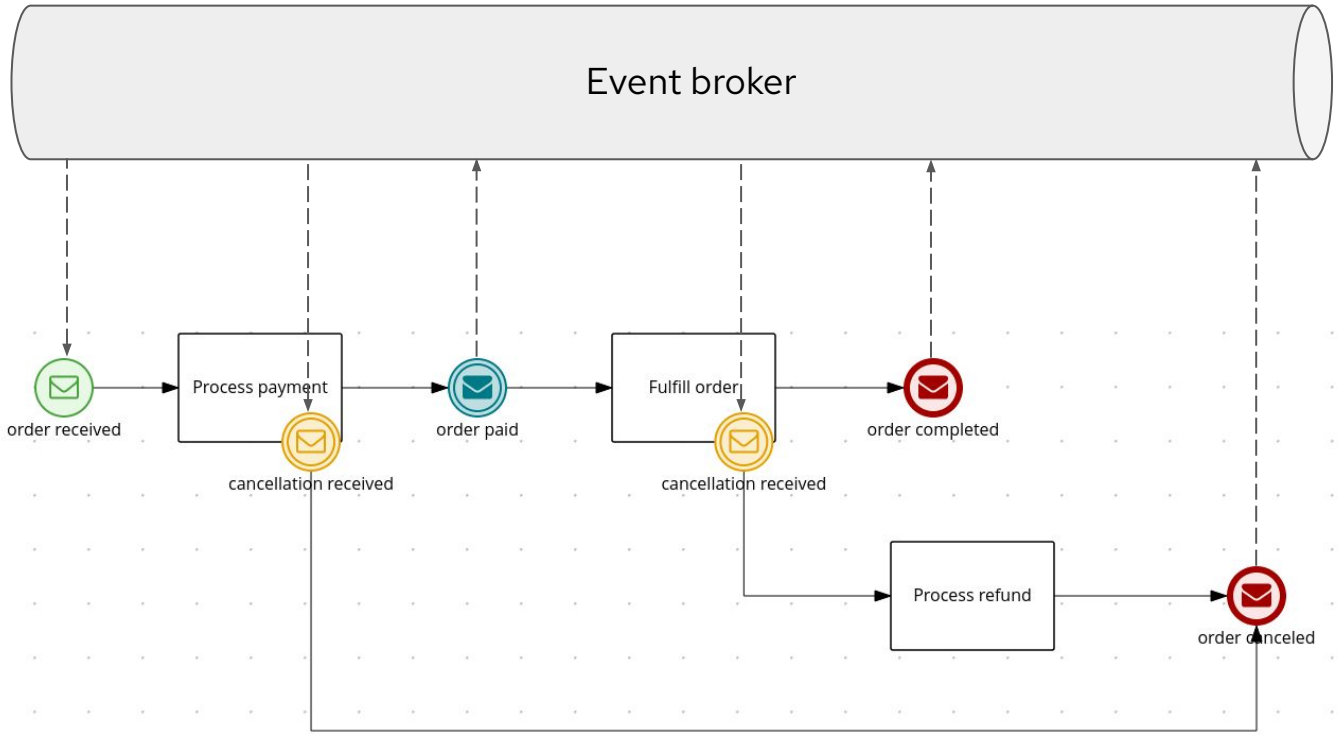






Red Hat

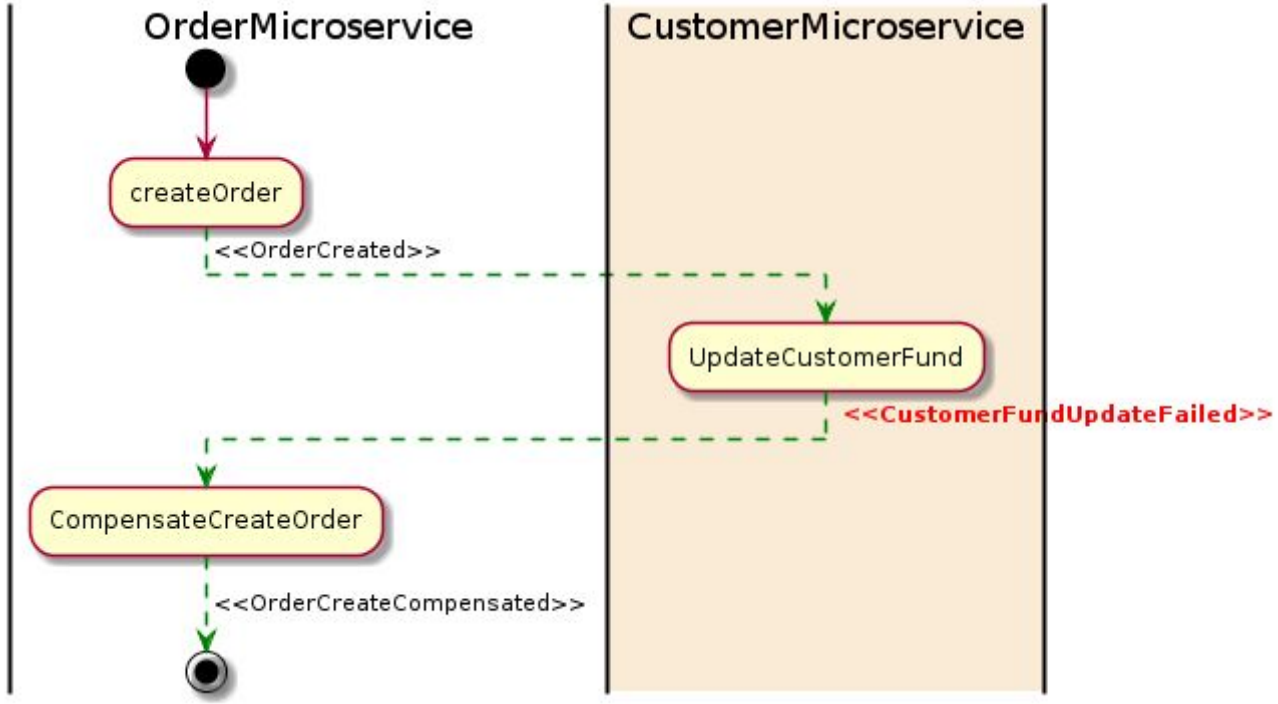
AUTOMATION OF EVENT-DRIVEN WORKFLOWS



Red Hat

AUTOMATION OF LONG-LIVED TRANSACTIONS

Saga pattern for microservice architectures



If any microservice fails to complete its local transaction, the other microservices will run **compensation** transactions to rollback the changes.

Advantages

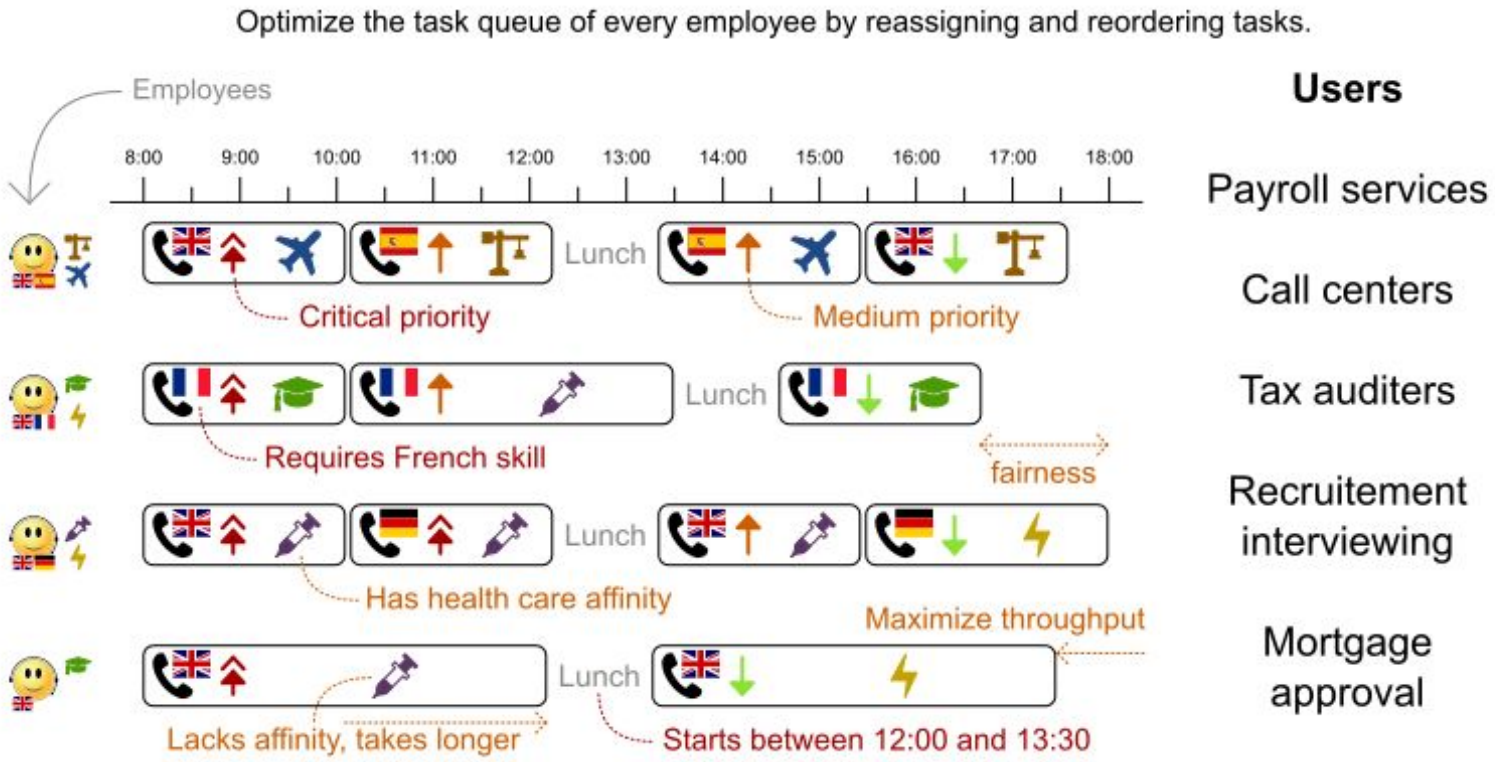
- support for long-lived transactions
- other microservices are not blocked if a microservice is running for a long time
- there is no lock on any object

Disadvantages

- difficult to debug
- difficult to maintain if the system gets complex
- does not have read isolation

Adding a **process manager** addresses the complexity issue of the Saga pattern when it becomes responsible for listening to events and triggering endpoints.

AUTOMATION OF TASK ASSIGNMENTS

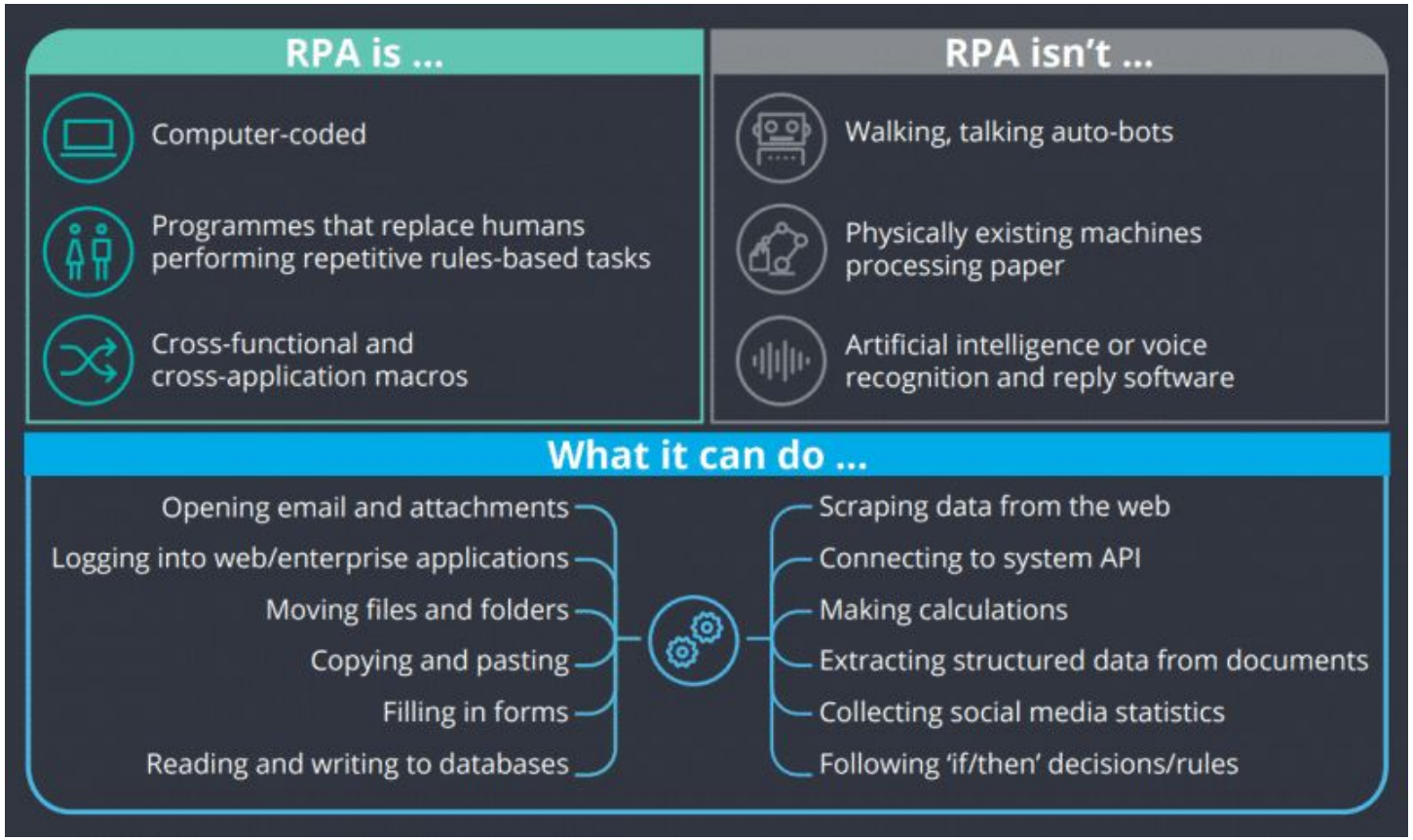


[KIELive#45] Optimizing user tasks assignment in business processes with Kogito and OptaPlanner

Red Hat

AUTOMATION OF HUMAN ACTIONS INTERACTING WITH SW

Robotic Process Automation (RPA)



Red Hat

AUTOMATION OF BUSINESS RULES

Kogito DRL Rules language

IBM Action rules express business policy statements using a predefined business vocabulary that can be interpreted by a computer.

Policy

"change customers in the `Gold` category to the `Platinum` category when they spend more than `$1,500` in a single transaction"

Action Rule

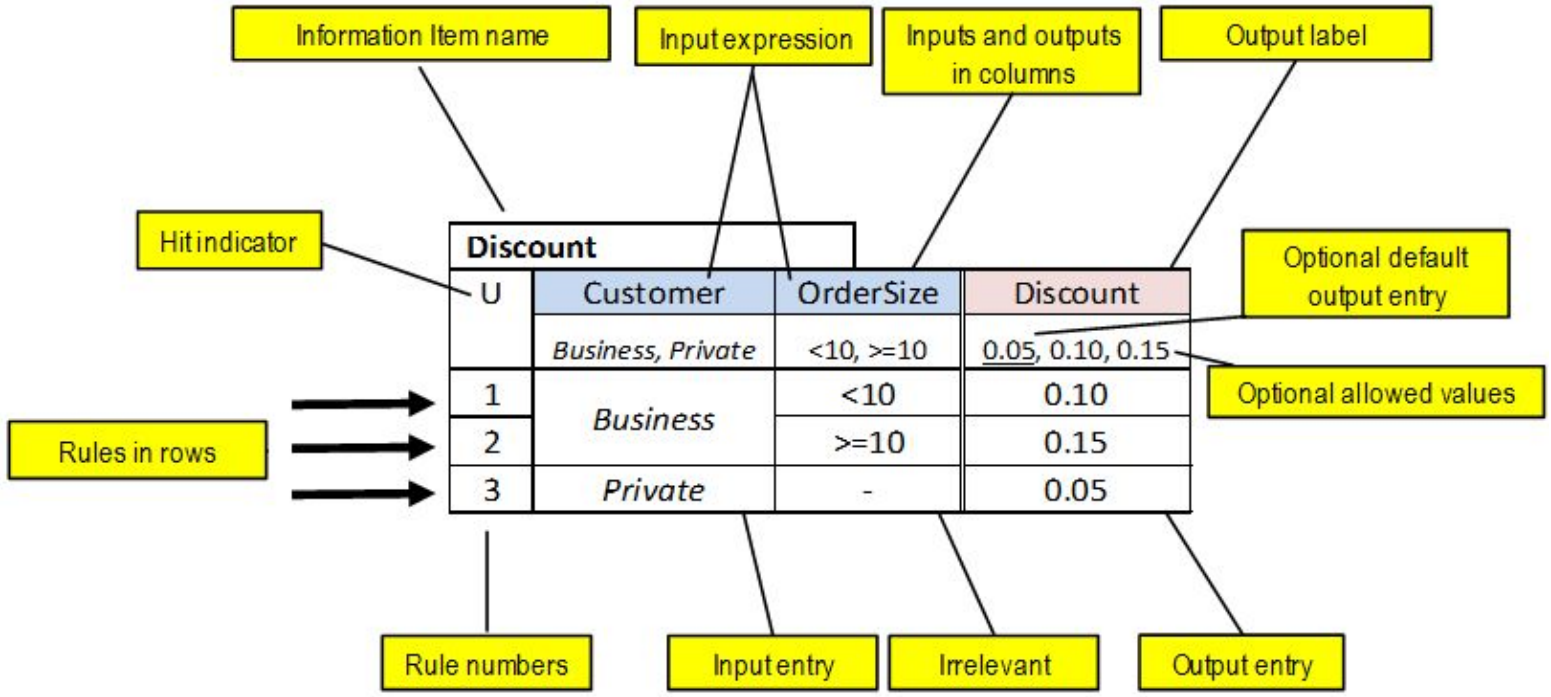
If All of the following conditions are true:

- the customer category is `Gold`
- the value of the shopping cart is more than `$ 1,500`

Then Change the customer category to `Platinum`

Red Hat

AUTOMATION OF DECISION TABLES



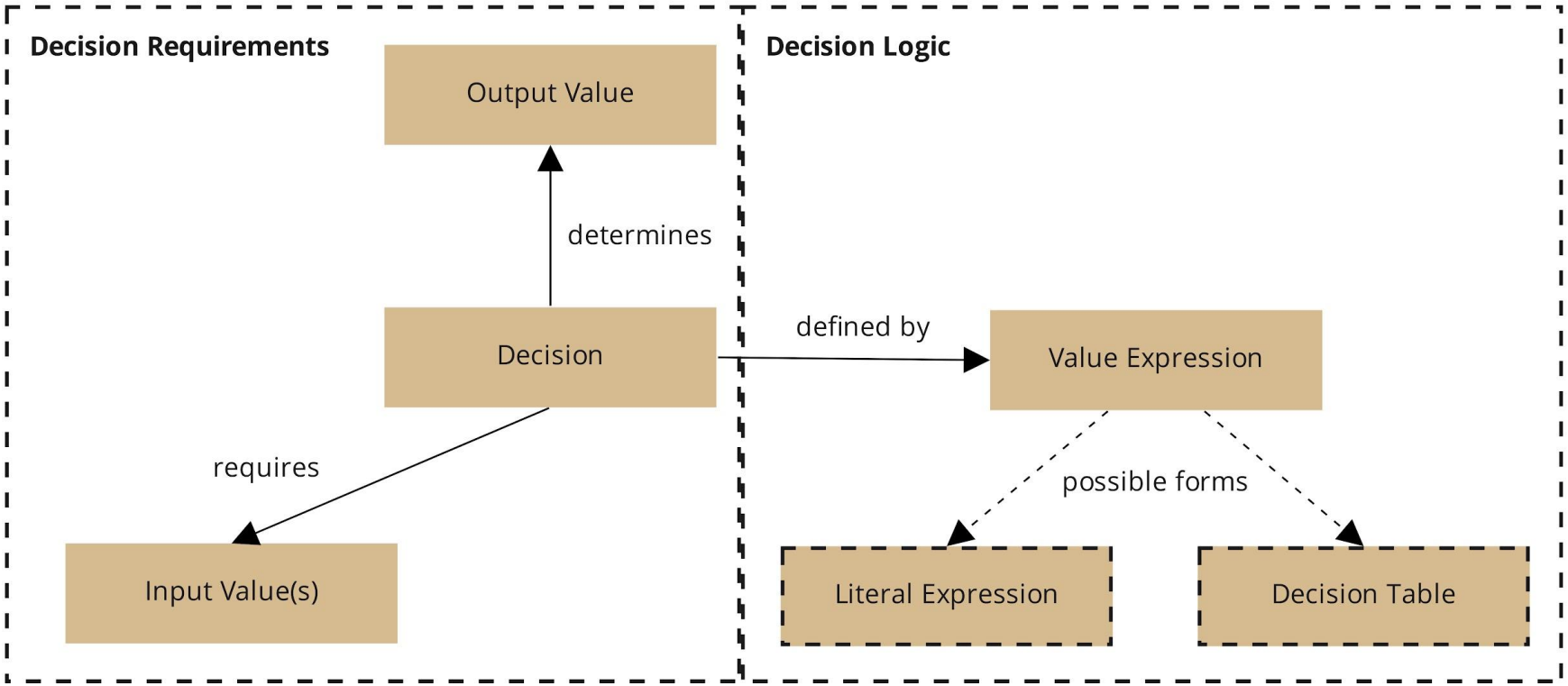
Types of decisions

- Selection (routing)
- Scoring
- Categorizing

The decision doesn't take an action (no side effect), just determines a data value

Red Hat

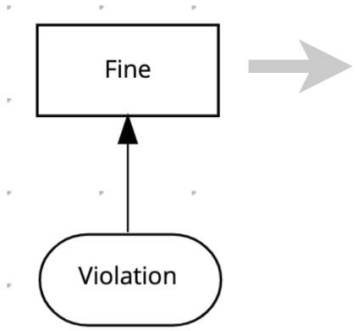
AUTOMATION OF DECISION MODELS



Red Hat

AUTOMATION OF DECISION MODELS

Executing decision logic in DMN models



Fine *(Decision Table)*

U	Violation.Type *(string)*	Violation.Actual Speed - Violation.Speed Limit *(number)*	Fine *(tFine)*		Description
			Amount *(number)*	Points *(number)*	
1	"speed"	[10..30)	500	3	
2	"speed"	>= 30	1000	7	
3	"parking"	-	100	1	
4	"driving under the influence"	-	1000	5	

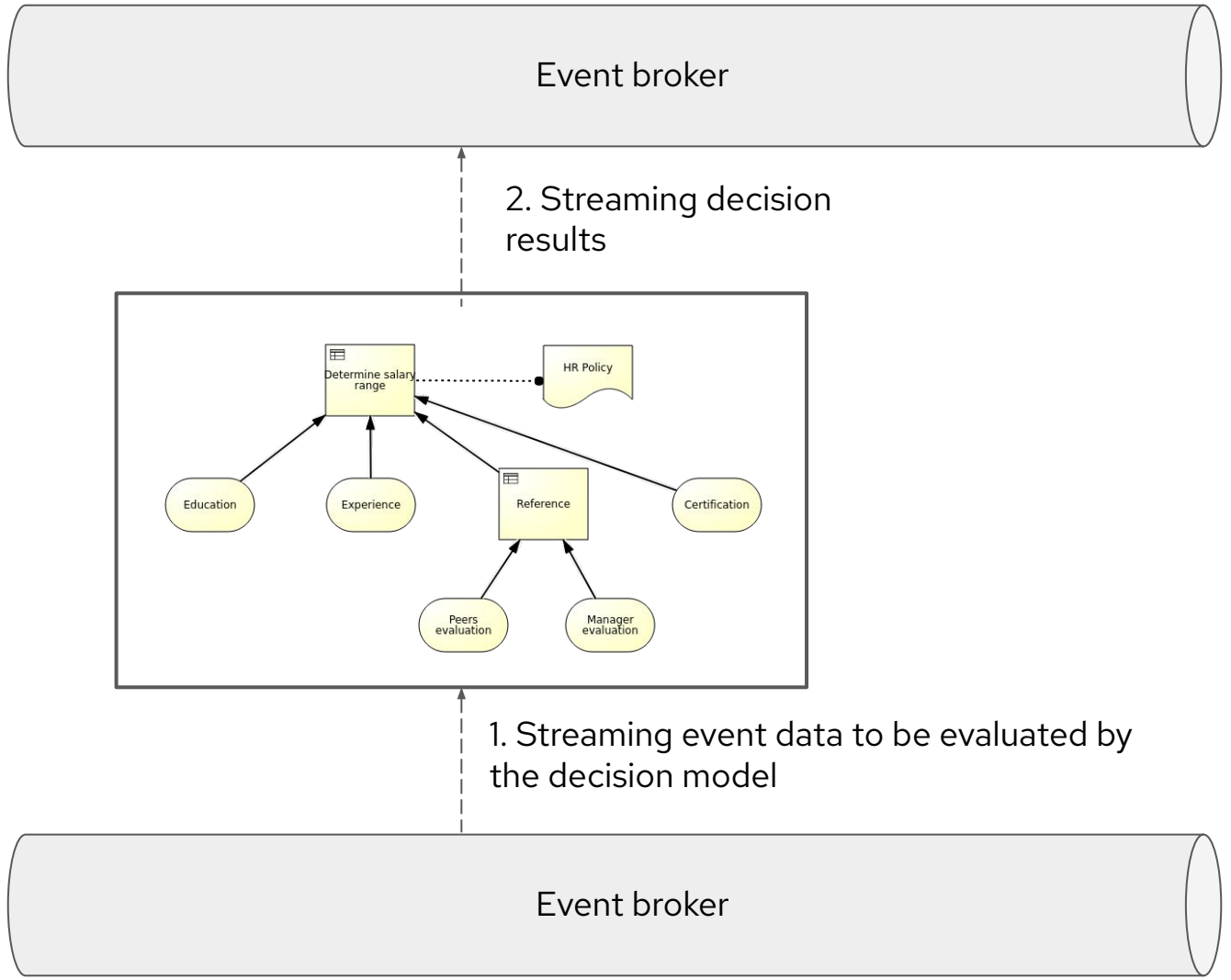
```
/* = Actual input data = */
"Violation": {
  "Type": "speed",
  "Speed Limit": 60,
  "Actual Speed": 100
}
```

```
/* = Expected output data = */
"Fine": {
  "Points": 7,
  "Amount": 1000
}
```

DMN with Kogito on Quarkus

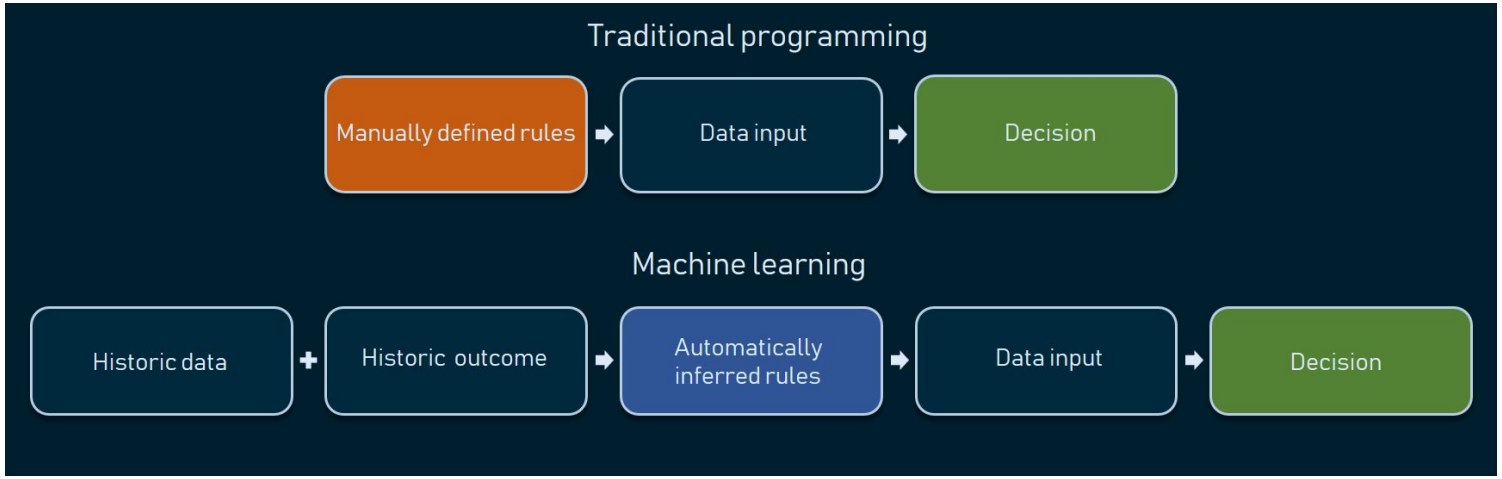
DMN FEEL Cheatsheet

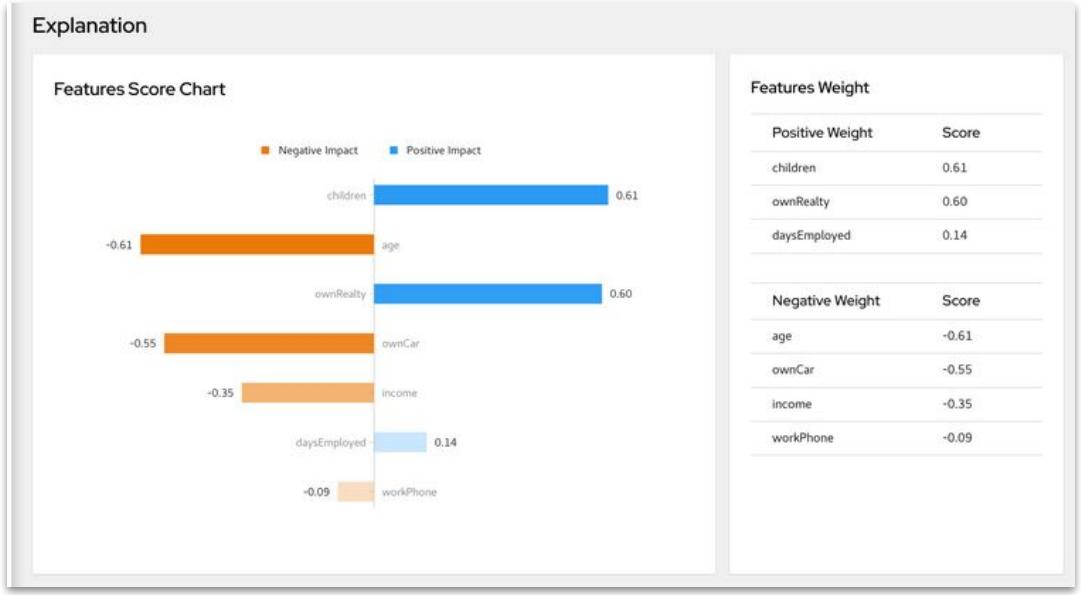
AUTOMATION OF DECISION MAKING IN STREAMS



Red Hat

AUTOMATION OF DECISION MAKING WITH AI





[TrustyAI]

[Can you trust AI? - presentation]

Red Hat

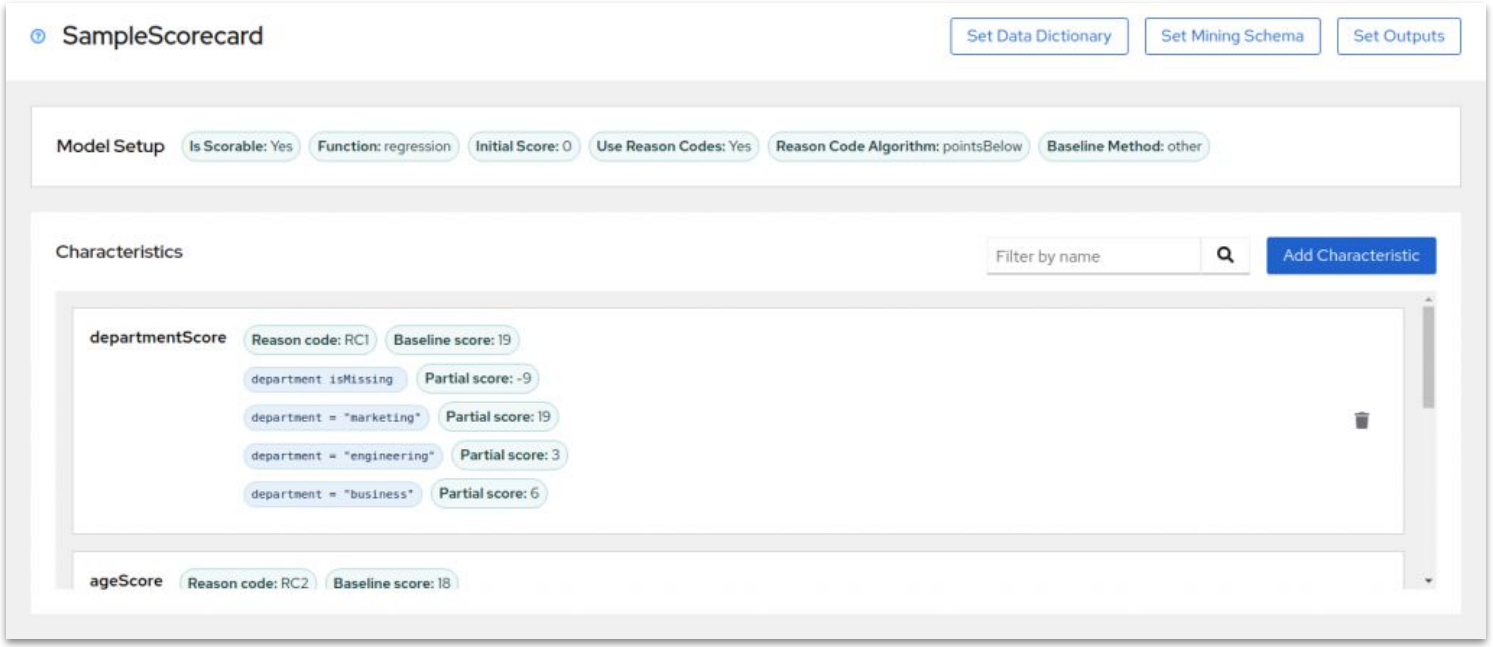
AUTOMATION OF SCORING

Scorecard is a risk management tool used mostly by banks to calculate the risk they take by selling you one of their products.

How risky is a customer?
What are the changes the customer might default on payment?

Based on the customer score we may adapt the product offer - a better interest rate, a higher credit limit, etc.

PMML Scorecard Editor in VS Code



Red Hat

AUTOMATION OF HUMAN-LIKE CONVERSATIONS

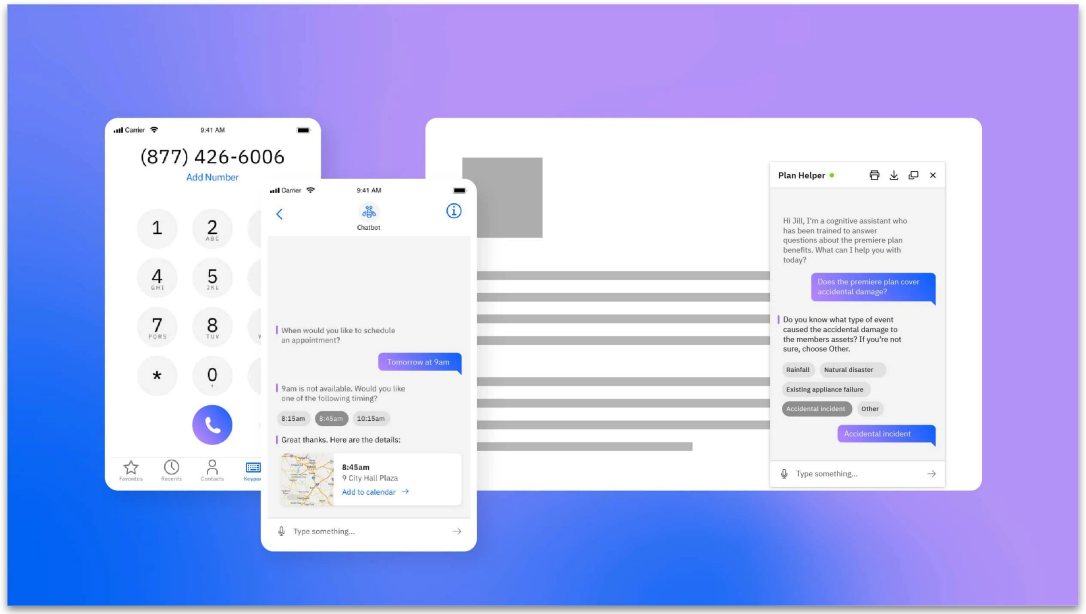
Chatbots

A chatbot is software that simulates human-like conversations with users via text messages on chat or text-to-speech.

- Customer self-service - call centres, scheduling doctor appointments
- Employee self-service - HR assistants, meeting and scheduling, expenses, people finder

Omnichannel - communication is done on all kinds of channels - phone calls, chat and email on computers, sms messages, ...

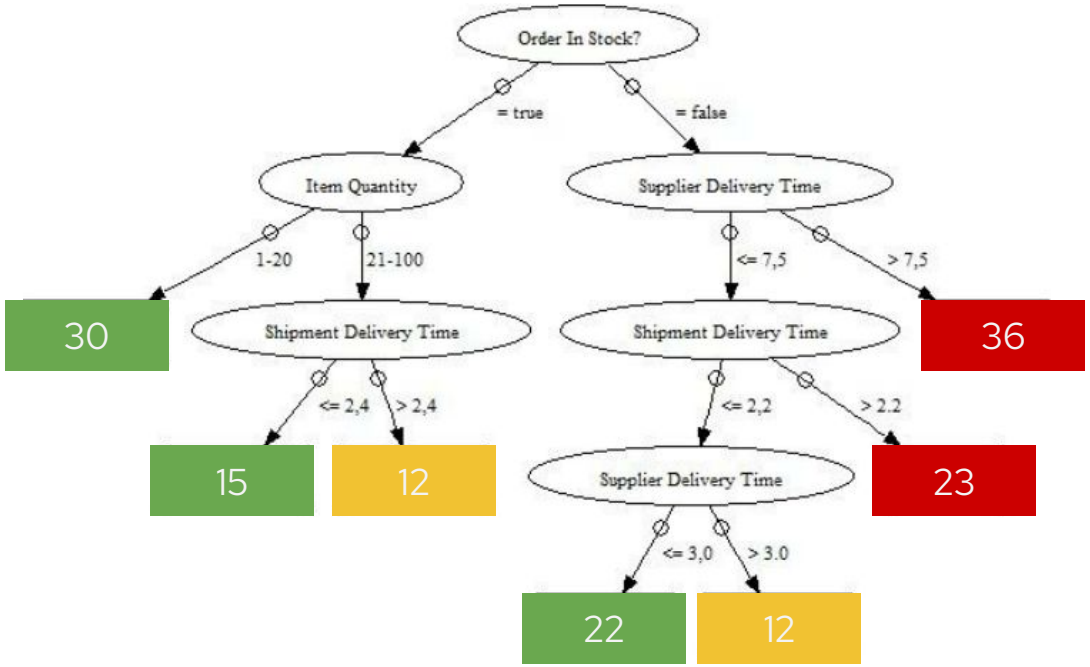
Watson Assistant - COVID-19 response automation



Red Hat

AUTOMATION OF BUSINESS MONITORING

KPI Dependency Tree



Alerts, triggers, ...

Red Hat

Choose the right software and services to fit your needs

Kogito

Cloud-native business automation for building intelligent applications, backed by battle-tested capabilities.

Red Hat

KOGITO

https://kogito.kie.org/

- Domain-Driven Development
- Generated Domain-specific APIs from your BPMN and DMN models
- Lightweight orchestration microservices
- Event-driven business logic
- Dev-mode hot reload
- Distributed sagas for microservices
- Serverless workflows
- Polyglot programming

More info at

- Blogs
- Youtube

KOGITO DEMO

- Available on GitHub
  - git clone https://github.com/MarianMacik/kogito-intro.git
  - cd kogito-intro-quarkus
- Contains BPMN and DMN integration
- Development mode with hot reload
  - mvn clean compile quarkus:dev
- Compile
  - mvn clean install
- Domain-specific API available at http://localhost:8080/q/swagger-ui/
  - OpenAPI specification

Thank you, questions?