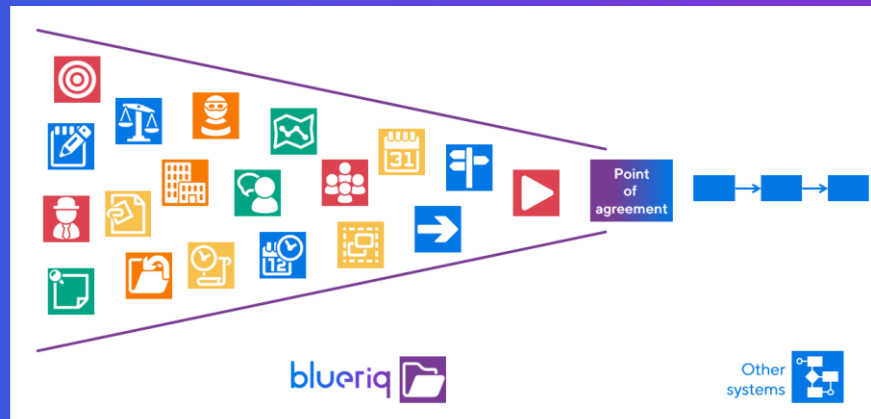


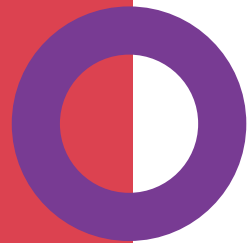
# Dynamic Case Management

A deepdive into Blueriq's proposition of DCM, the Foundation that supports this proposition and how to model with the Foundation in Studio



blueriq

# Goals



## Why DCM, what is it, and how to model this with Blueriq

Understand what Dynamic Case Management is

Understand the proposition by Blueriq

Get to know all DCM concepts by means of a demo

Plot the DCM concepts onto Blueriq concepts

Get to know the Blueriq process engine

Understand Persistency management

Being able to transform BPMN processes to DCM processes

Understand the DCM foundation template by means of a standard solution

Being able to make changes in the case and the process of this solution

Being able to make sound choices when adjusting the solution template with regards to maintainability and understandability

# What is Dynamic Case Management?



Case Management focusses on the **case centric** (not process centric) handling of customer's needs to achieve the customer's goal.



The term **Dynamic** implies that the customer's needs do not adhere to a predesigned order of activities, tasks and events, but that the information about the customer and a set of rules determines the next steps.



DCM has a few principles:

A case has one **goal** and a **result**.

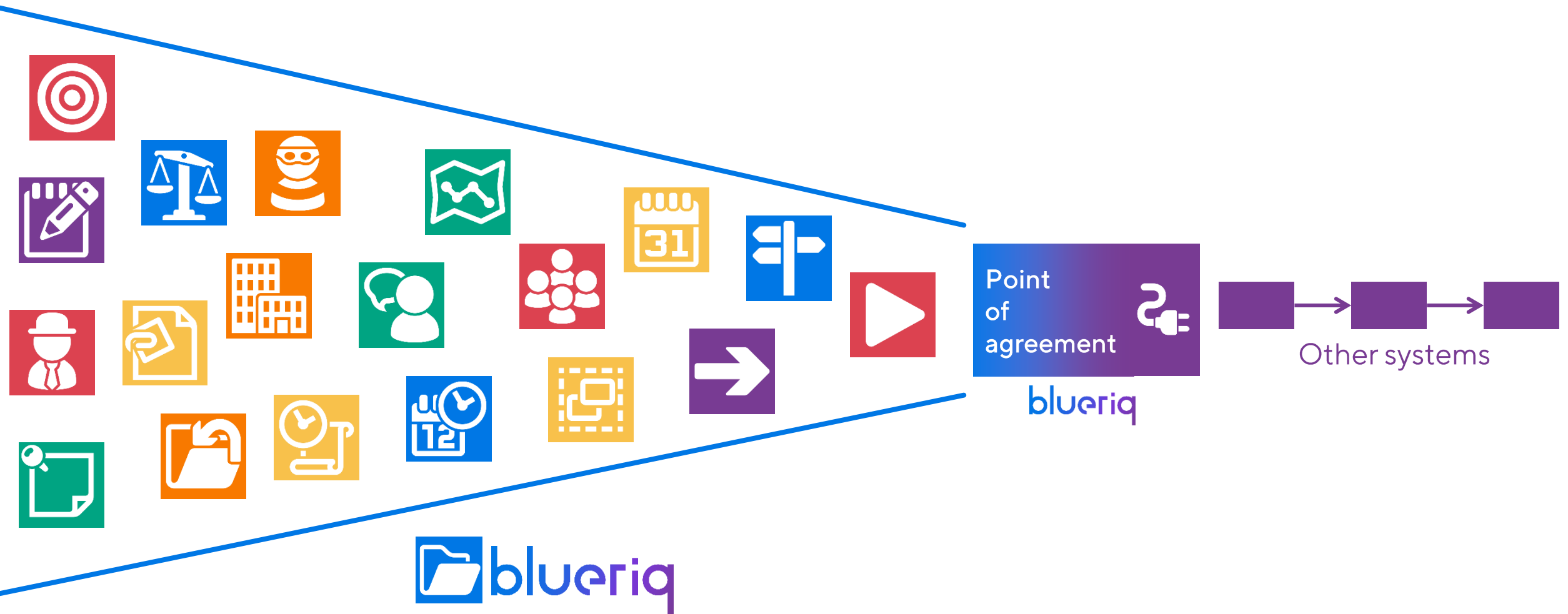
**A case has a start and an end.**

**Cooperation** by all involved to reach that goal.

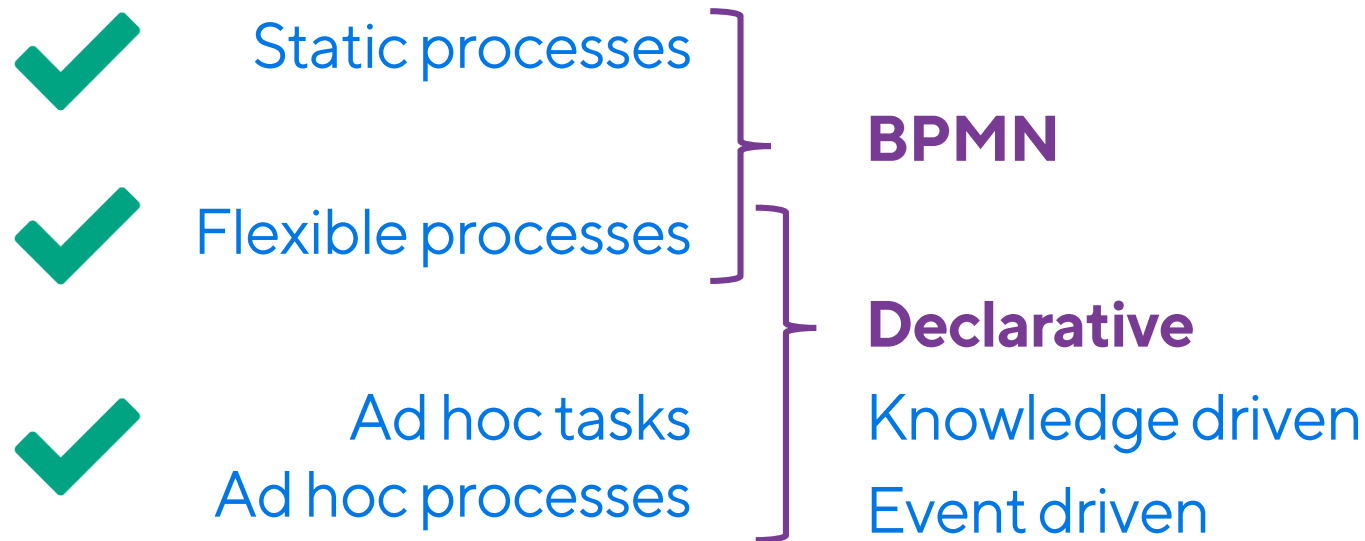
A case provides a **central view** of all relevant information, decisions and open tasks for all case participants. Principle need to know, need to do.

A case is handled **dynamically**, completely based on declarative **rules**.

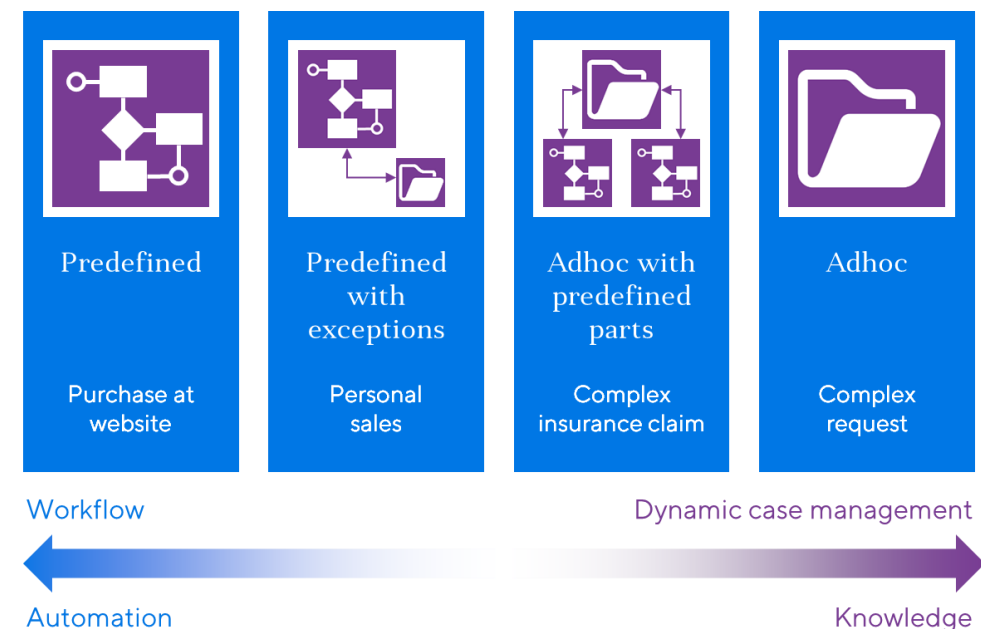
# Point of agreement



# Process Dynamics



✗ Adaptive processes  
 Designtime = runtime  
 Enduser editors



# Benefits of a declarative process



## Flexible

- Goal driven
- Those involved are in control, not the process



## Deal with change

- No knowledge in the process
- Rules guide the process, based on the customer's context
- Changes do not have to interfere with processes
- The process does not change that often, but the rules do



## Reusable

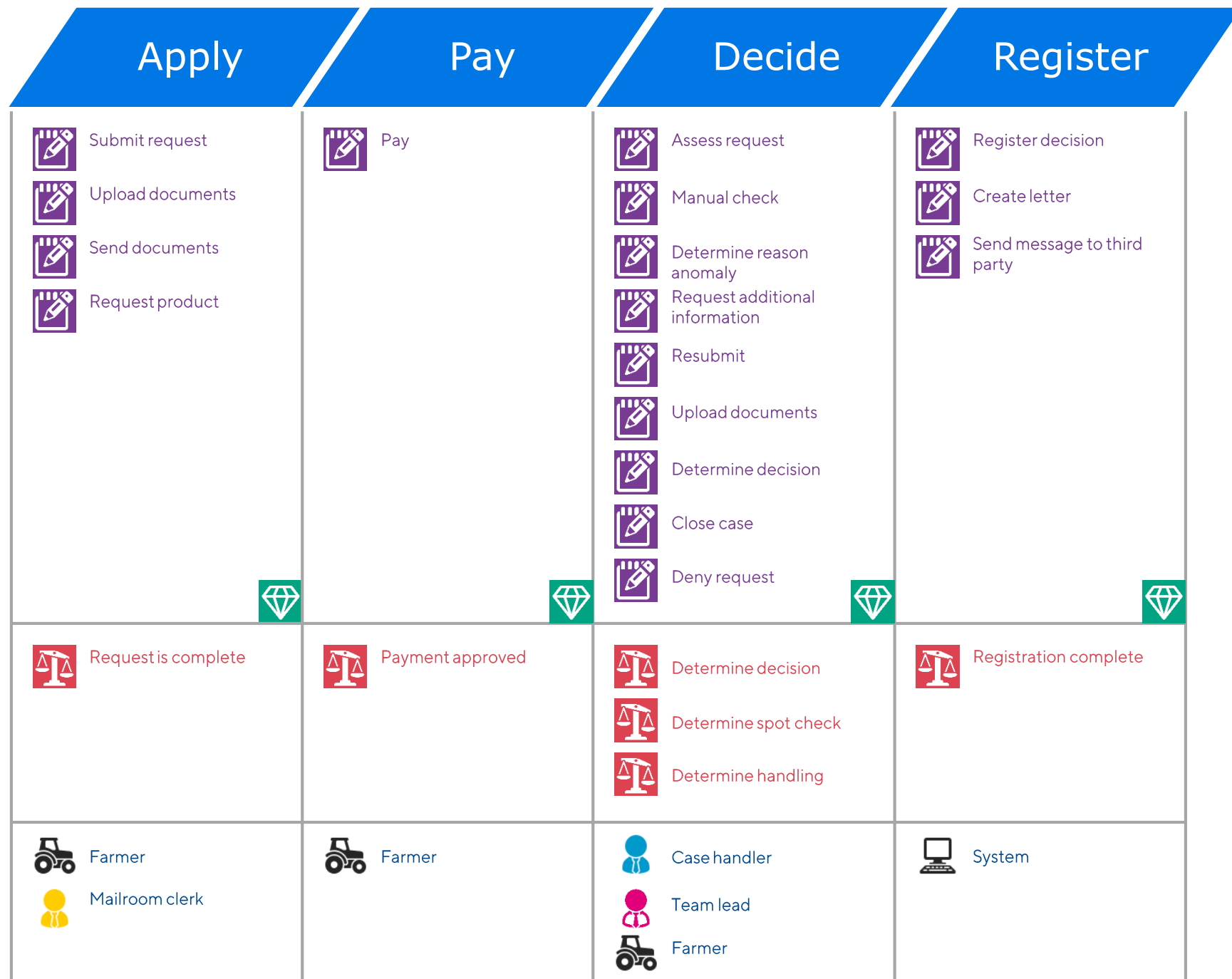
- Template process parts customizable for different products and services
- Manual and automated (STP) case handling



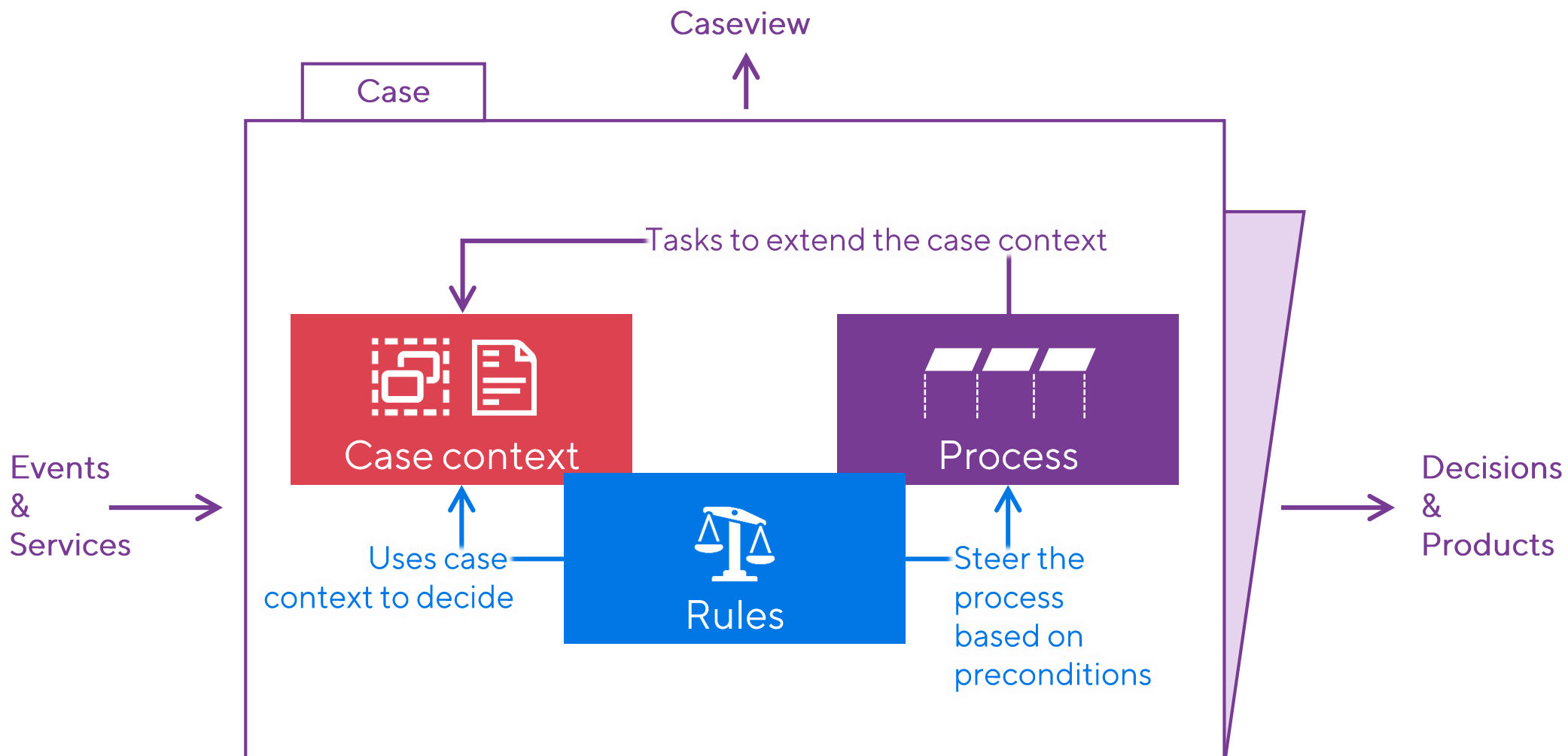
## Maintainable

- One process, many rules

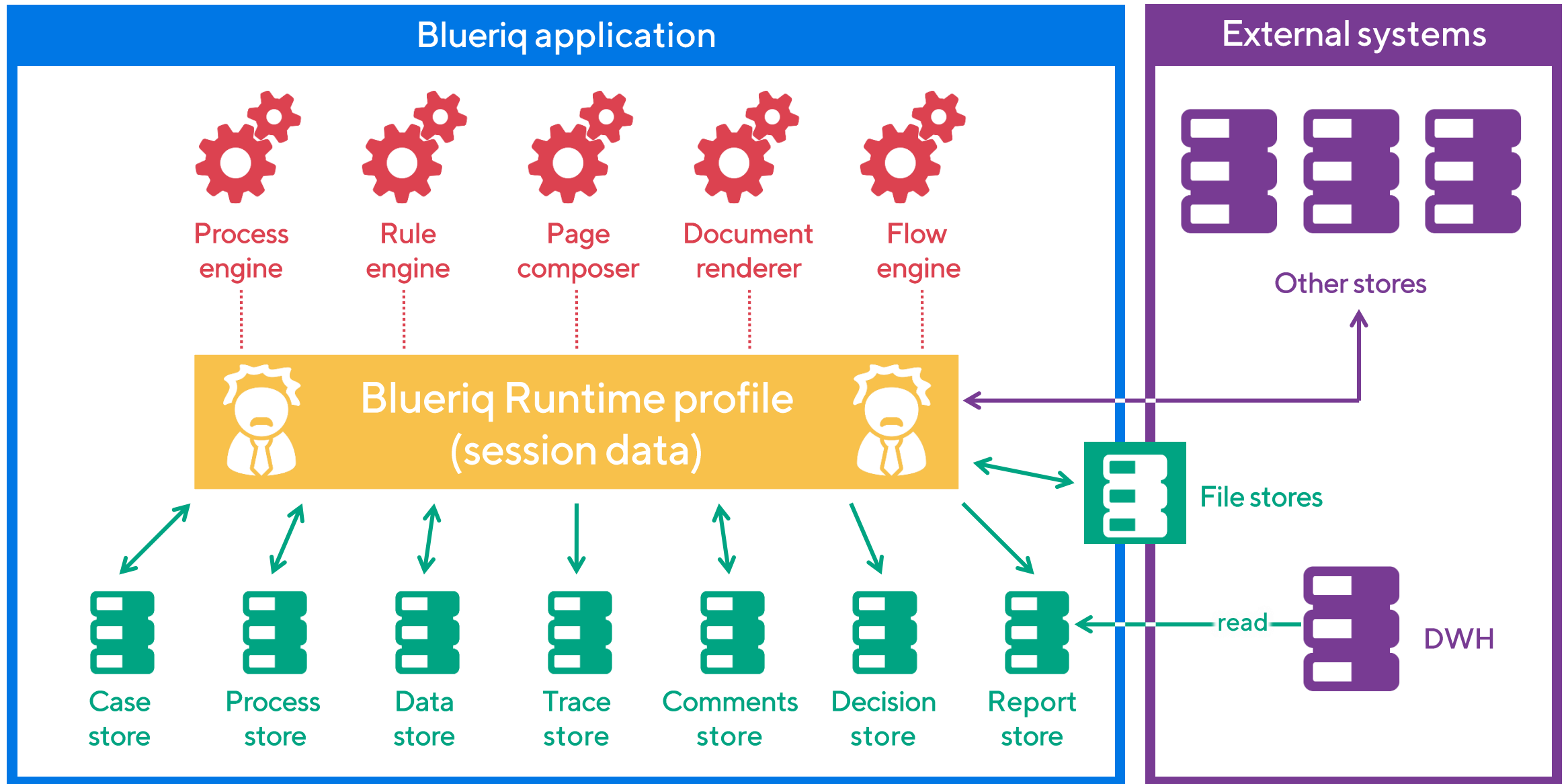
# Case centric design



# Recap demo



# Stores

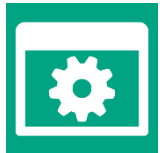


# Types of tasks



Tasks can be

- Manual or Automatic ⚙️
- One-time or Repeatable ↺️
- Optional or Mandatory ✖️
- Planned or Ad hoc ~



Ad hoc tasks have a precondition stating when they may be executed



Necessity (mandatory) can also be specified by means of a precondition





































The four dimensions can be combined, with one exception:


A task cannot be **Automatic** and **Repeatable** at the same time


Create a repeatable subprocess with the automatic task in it instead


# Case


Apply	Pay	Decide	Register
<div><div> Submit request</div><div> Upload documents</div><div> Send documents</div><div> Request product</div></div> <div></div>	<div><div> Pay</div></div> <div></div>	<div><div> Assess request</div><div> Manual check</div><div> Determine reason anomaly</div><div> Request additional information</div><div> Resubmit</div><div> Upload documents</div><div> Determine decision</div><div> Close case</div><div> Deny request</div></div> <div></div>	<div><div> Register decision</div><div> Create letter</div><div> Send message to third party</div></div> <div></div>
<div> Request is complete</div>	<div> Payment approved</div>	<div> Determine decision</div> <div> Determine spot check</div> <div> Determine handling</div>	<div> Registration complete</div>
<div><div> Farmer</div><div> Mailroom clerk</div></div>	<div><div> Farmer</div></div>	<div><div> Case handler</div><div> Team lead</div><div><div> Farmer</div></div></div>	<div><div> System</div></div>


Case





 Goal


 Decisions


 Processes


 Milestones


 Phases


 States


 Data


 Tasks


 Related cases


 Risk profile


 Notes


 Appointments


 Events


 Documents in


 Documents out


 Involvements

 Teams

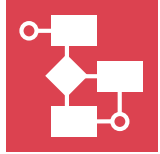
 Organizations

 Roles

 Time mgmt

 History

# Process engine benefits



Modeling **Time limits**

**STP handling**, automated tasks, automatic cases

Automatic **tracing** of all actions of the process engine

Modeling **4-eye** decision making a.k.a. **2-man rule**

**Overview** of the process in a model

**Event handling** over multiple processes

Modeling **locking** of tasks

**State** of the process, cooperation/parallelization

Provide tasks with **data**

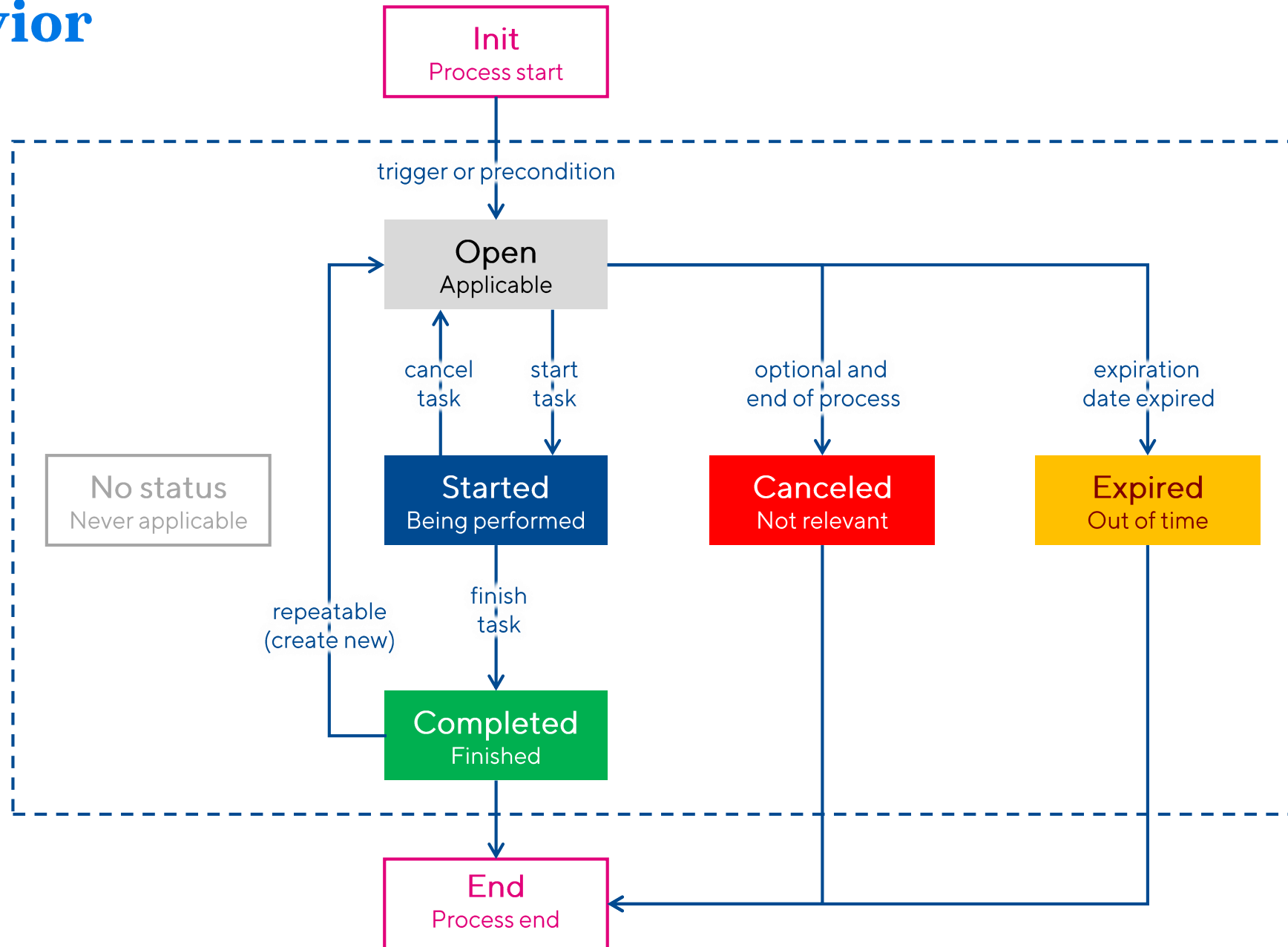
Authorization

Hooks for tasks priority and routing

**Repeat** loops, repeating tasks

**Timeline** for an overview of who did what when

# Task behavior



# Decoupling and mapping

