Slice your business into functions and events

Maciej Swiderski

who am I?

- independent software engineer and consultant
- workflow enthusiast in the field for more than 15 years
- creator of Automatiko project
- write blog posts to share knowledge
- occasionally tweets about workflows and software development

@SwiderskiMaciek

Events

Your data contract

Data that is exchanged between services and functions that can be both inputs and outputs

Events should be in well defined format as that in many cases will be considered as a contract between services and functions

Relying on standards can actually make it more portable and gain better understanding on the data being send over the wire

Functions

Your business logic pieces

Functions are meant to encapsulate business logic as executable action. They should be small and self contained to allow fast and efficient scale (up and down) to respond to demands

Functions should follow certain principles

- make functions to be specific - perform one (business) action
- avoid functions to call another function
- keep it small but meaningful

Knative Eventing

Workflows

Your complete business logic

Workflows in general are meant to capture business logic in a structured way.

That is done either graphically (e.g. like BPMN does with form of flow charts) or declaratively

Main benefits of workflows are:

- visibility
- isolation
- domain data contract

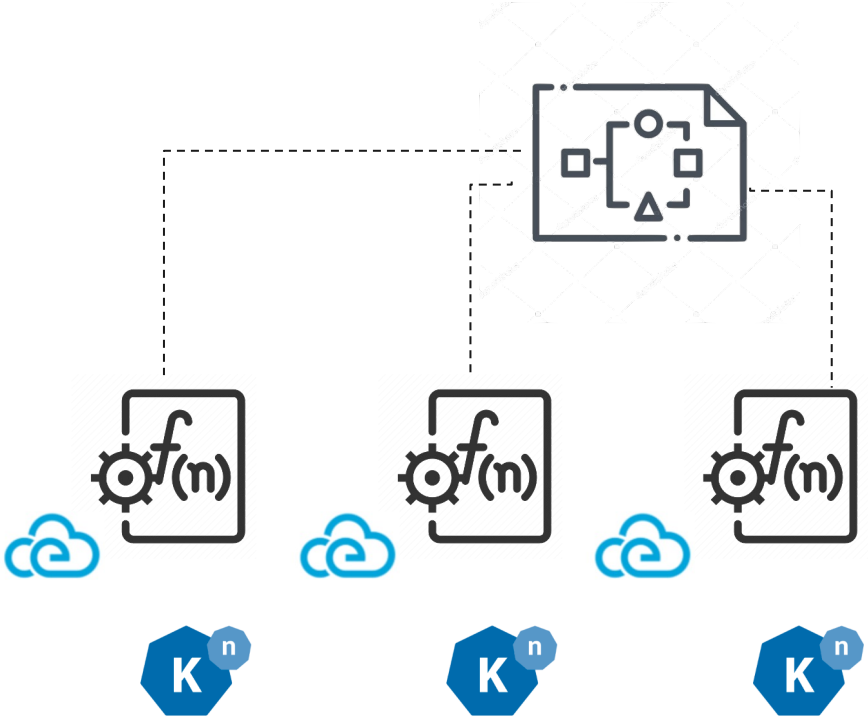
Workflow as a function flow

Your complete business logic executed as functions

Workflow as a function flow allows to define complete business logic that in most of the cases is more complex than just a single function but execute it as set of functions that are

- self contained
- independent
- invokable at any time
- scalable

Functions in the workflow



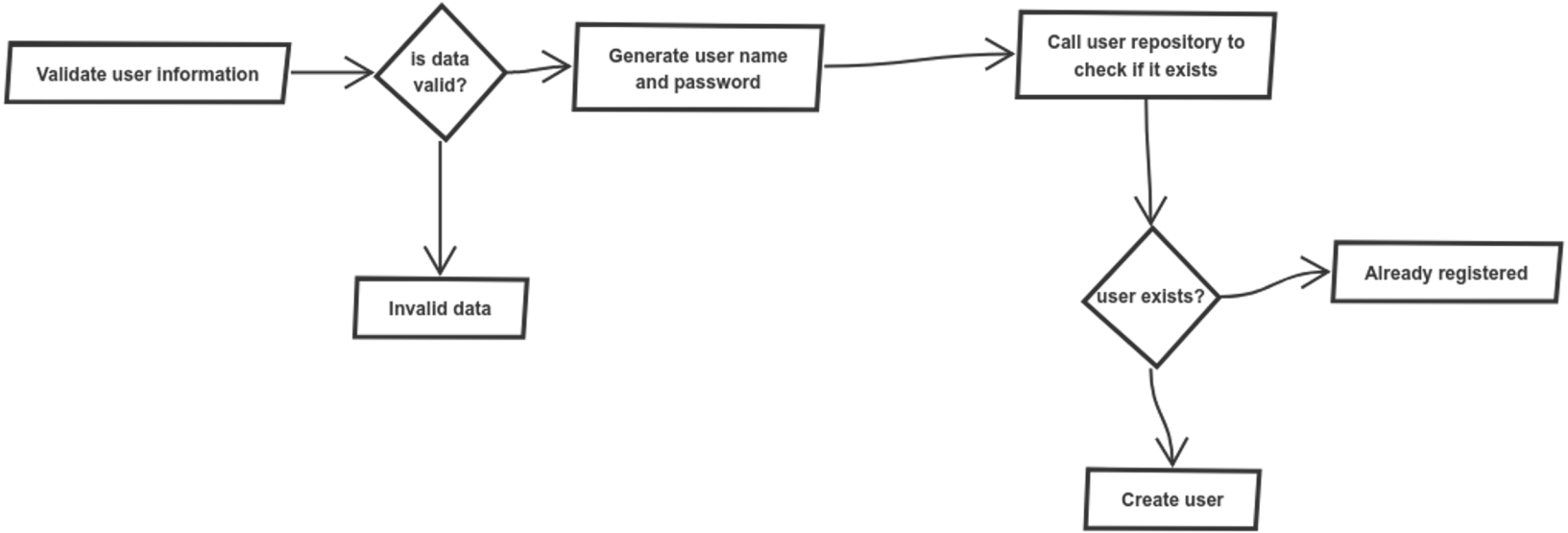
- Workflow definition is sliced into functions that
  - will be a **dedicated entry point**
  - will have **input**
  - can produce **one or more outputs**
    - becomes a sink
  - will have Knative **trigger** associated with it

- All of these steps are performed based on **workflow definition at build time** (generates both code and manifest file)

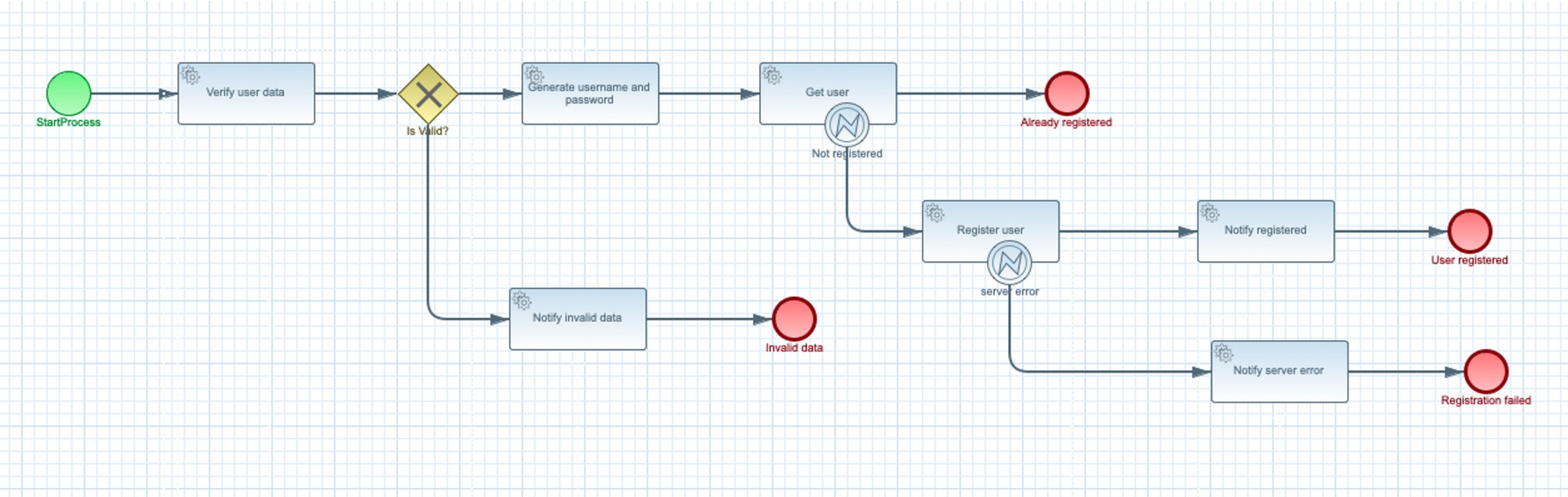
Let’s look at example

User registration use case

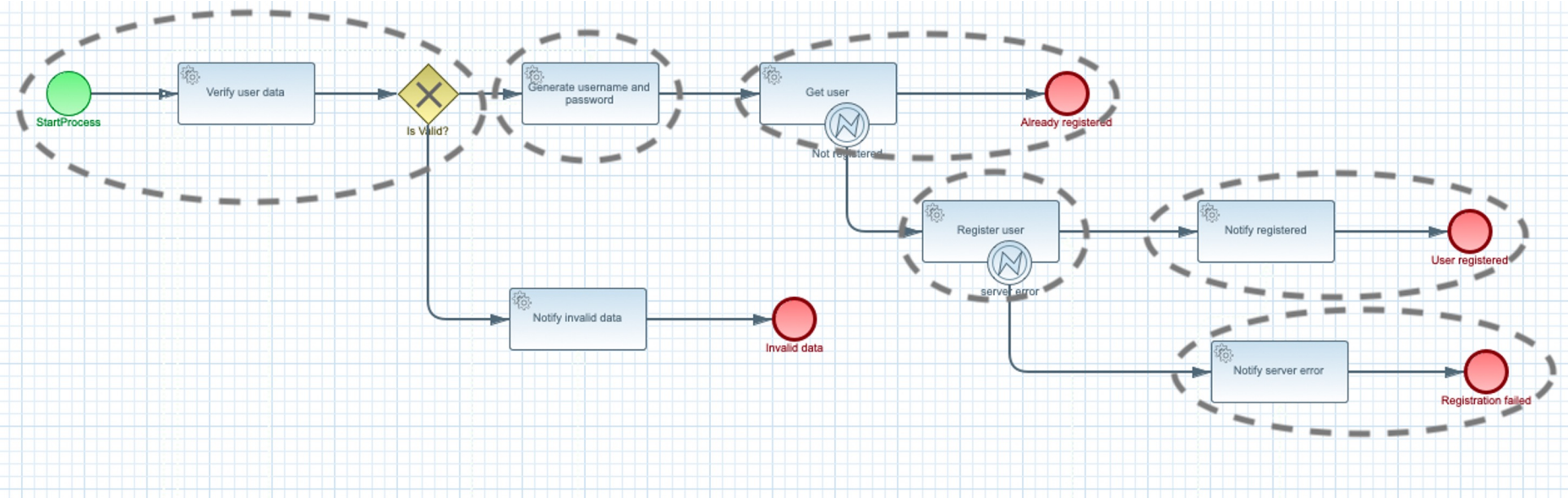
User registration use case



User registration workflow



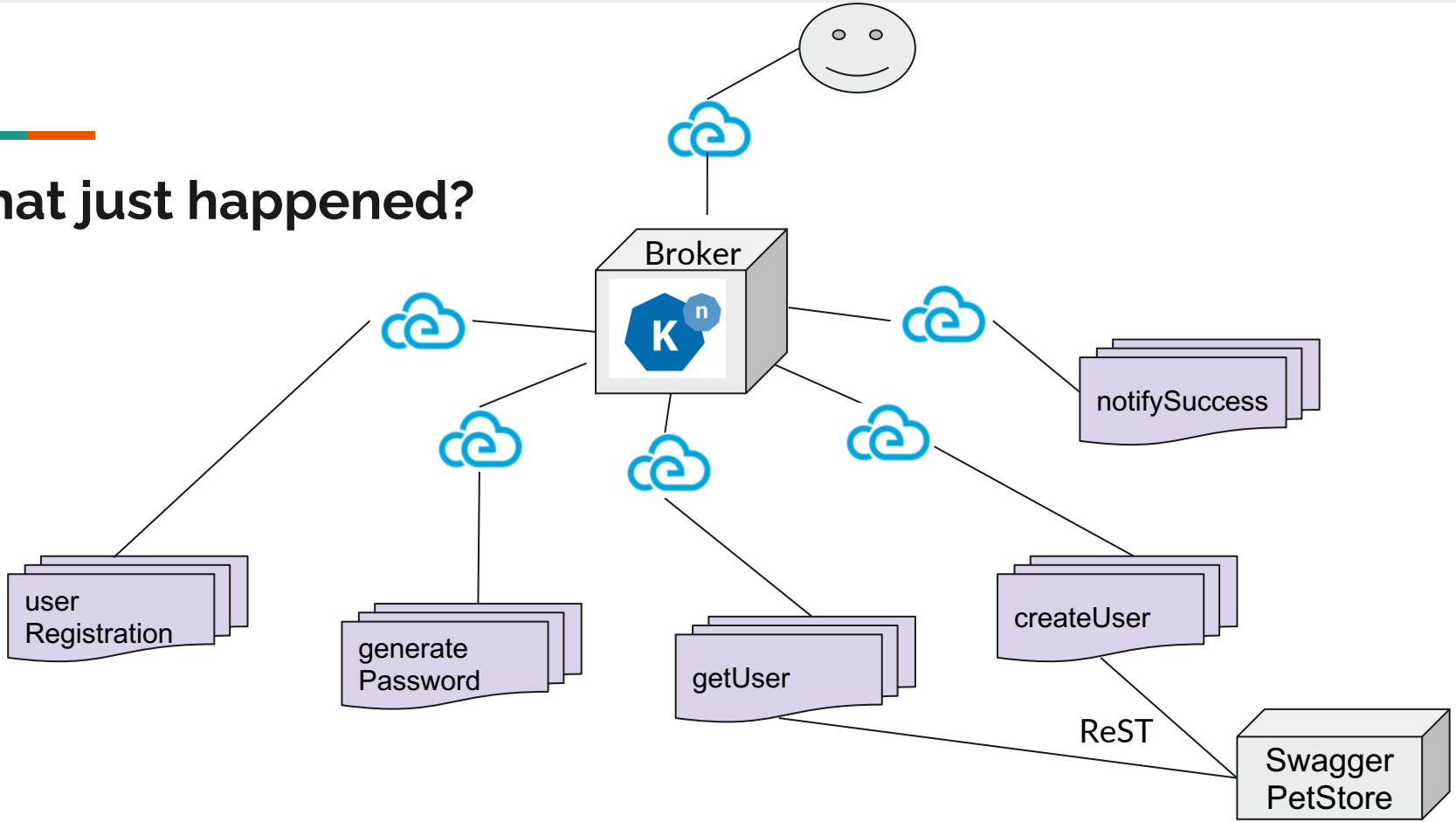
User registration workflow as function flow



Let’s see it in action

User registration as function flow

What just happened?



What is it built with?

- Workflow definitions created with BPMN2 format (graphical flow chart)
- Backed by Automatiko project
  - Open source project (Apache2 license)
  - Implemented in Java
  - Relies on Quarkus and Microprofile specifications - Quarkus Funqy for functions
  - Compiles to native executable via GraalVM
  - Packaged as container images
- Deployed to Kubernetes cluster
  - With Knative Eventing as the backbone of the function invocation and cloud events for data format

Want to know more?

- Get in touch on twitter **@SwiderskiMaciek @automatiko_io**
- Visit website
  - https://cloudevents.io
  - https://knative.dev
  - https://automatiko.io
  - https://quarkus.io

Questions?