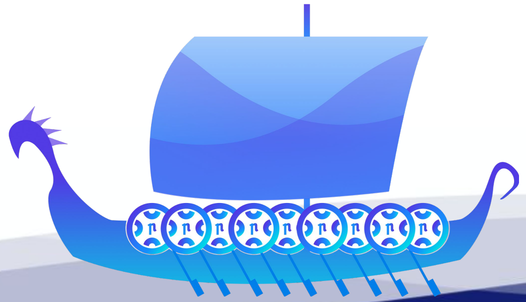
# Rönd

## A lightweight sidecar for security enforcement



# What is Rönd?

# What is Rönd?

**Rönd** is a lightweight container that distributes security policy enforcement throughout your application.

Rönd allows you to **define your own security policies** and distribute them across your application in order to:

Prevent undesired **API accesses**

**Generate queries** to be applied to protect your data

**Modify responses** to remove sensitive information

## Why did we create it?

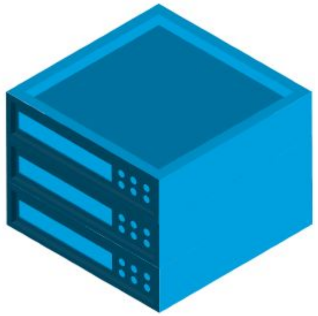
We needed to introduce strict security enforcement over what users can do but we had several constraints:

Avoid the introduction of a single point of failure

No unnecessary changes to existing codebase

# Why did we create it?

We needed to introduce strict security enforcement over what users can do but we had several constraints:



Build something flexible

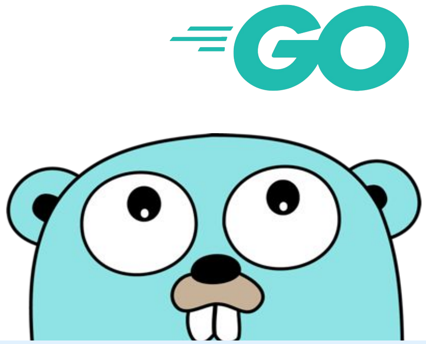
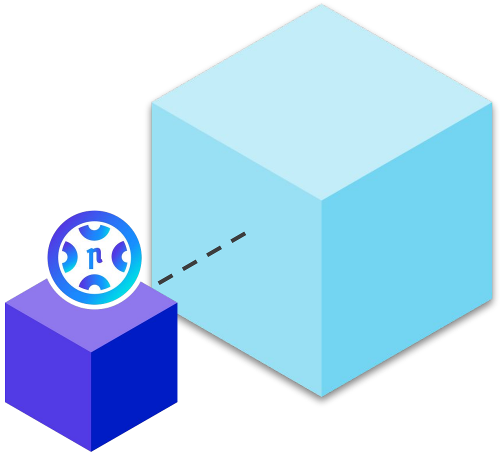
Extracting only authorized data from the db

Build something to define RBAC rules on-top

## How we addressed these constraints

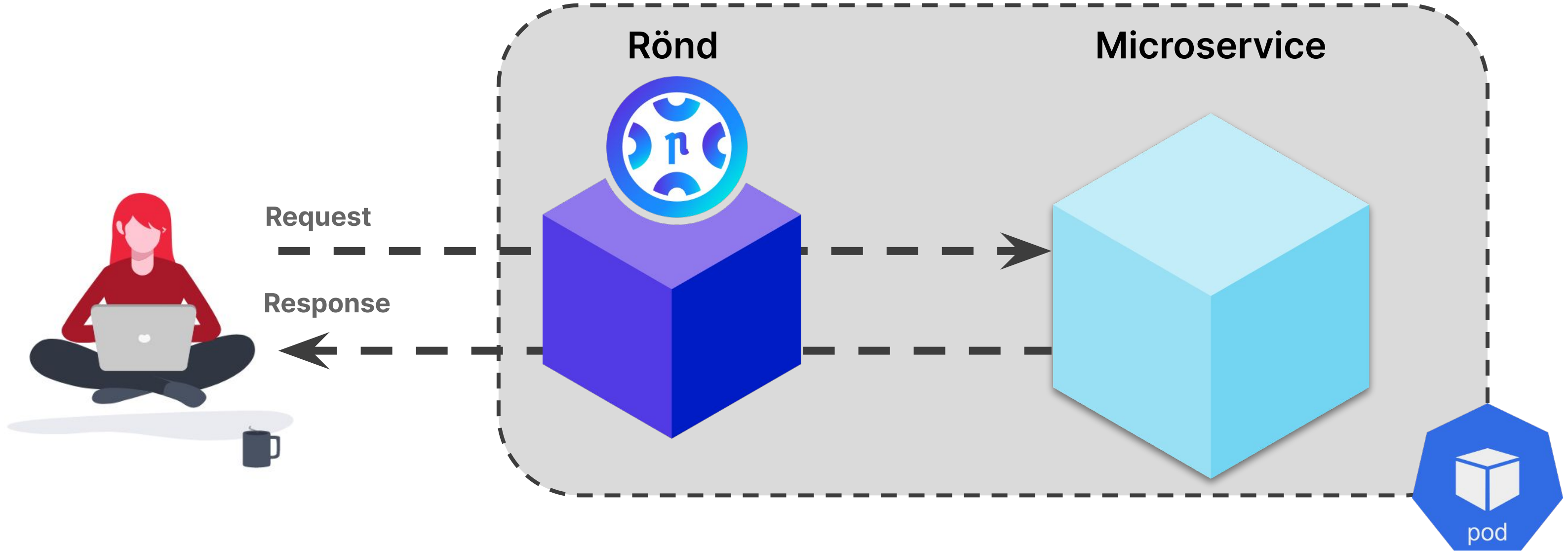
## The solution

- Leverages **Open Policy Agent** (OPA) and **REGO** Language
- Adopts an **agnostic** design
- **No single point of failure** introduced

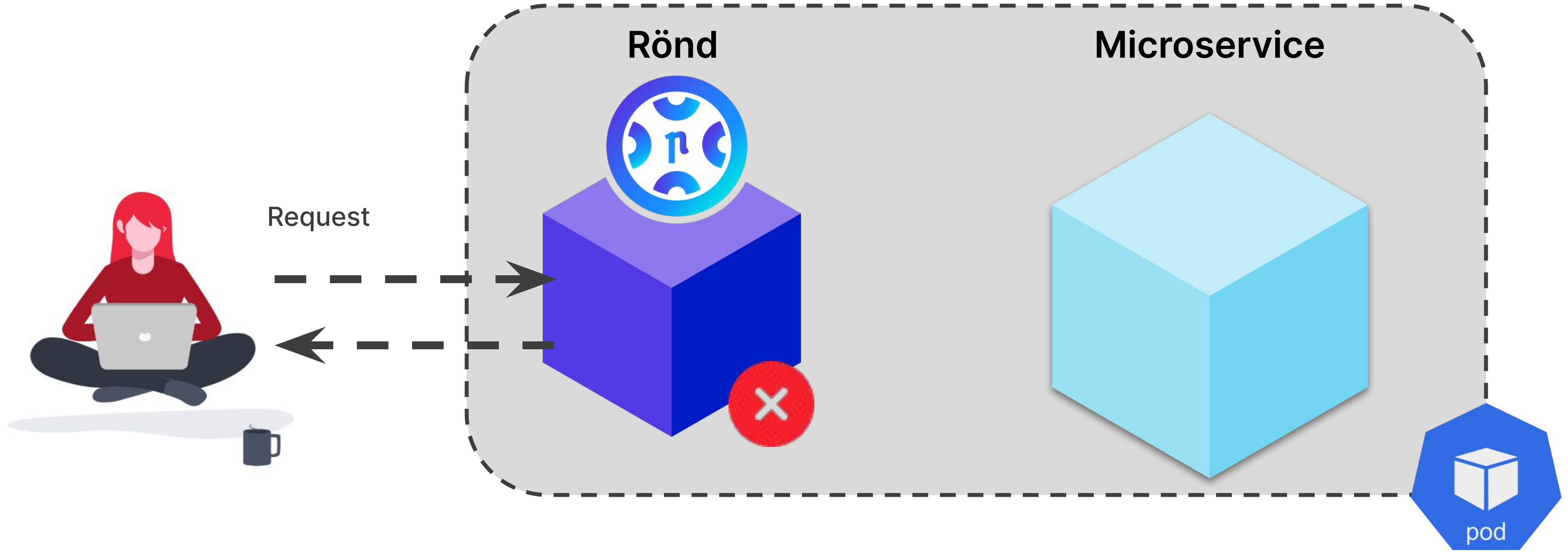
# Sidecar Container Pattern

Thanks to the Sidecar Container running in each Pod, no Single Point Of Failure is introduced

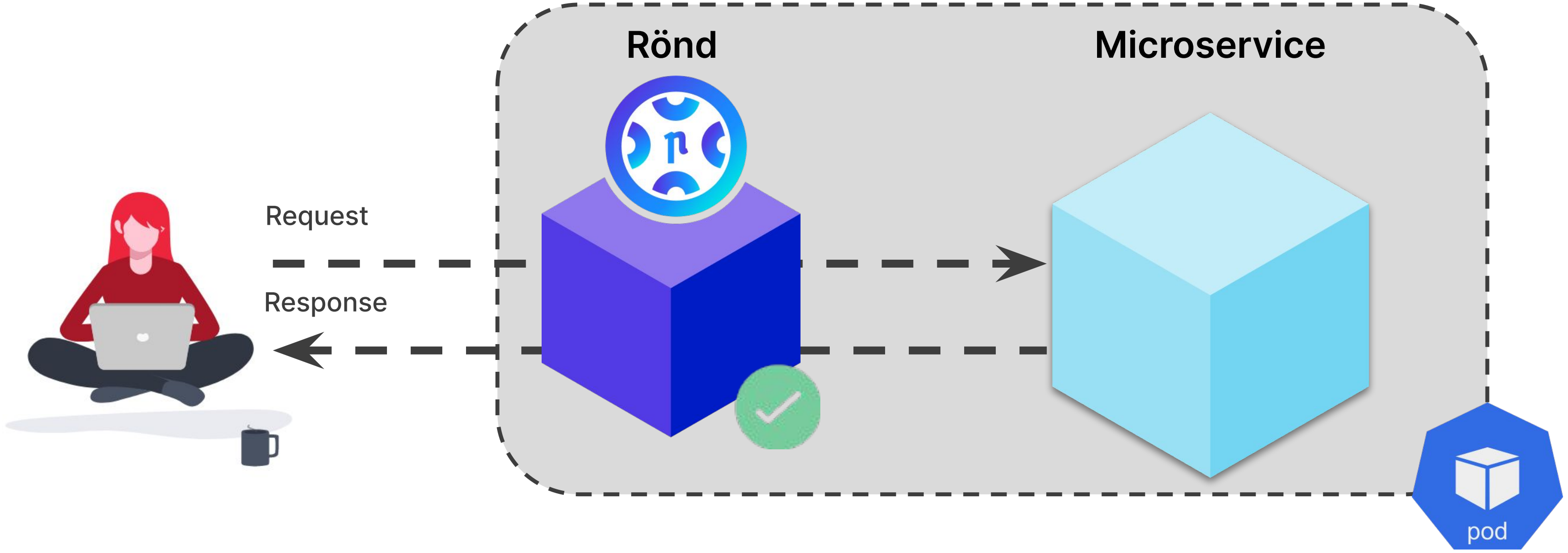


## The Features

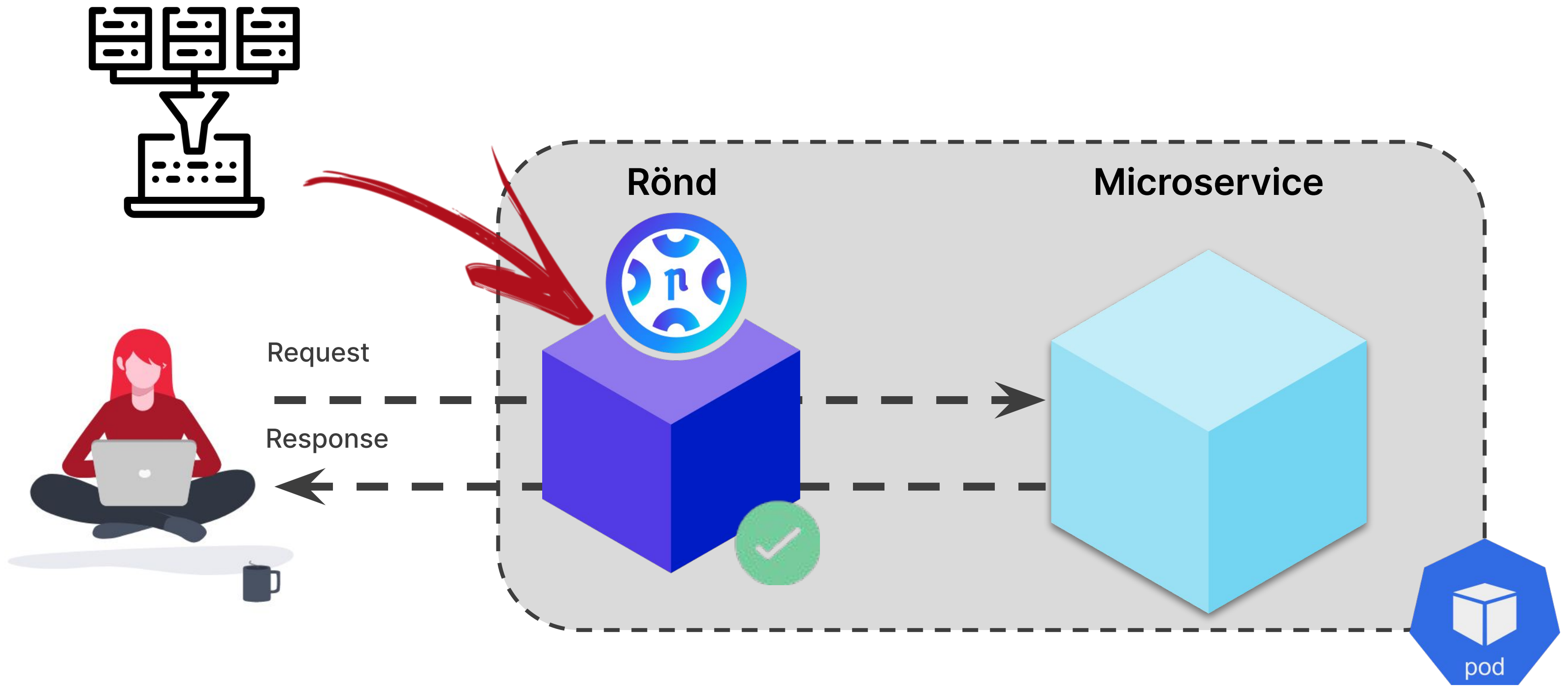
# Rönd flow rejecting the user request



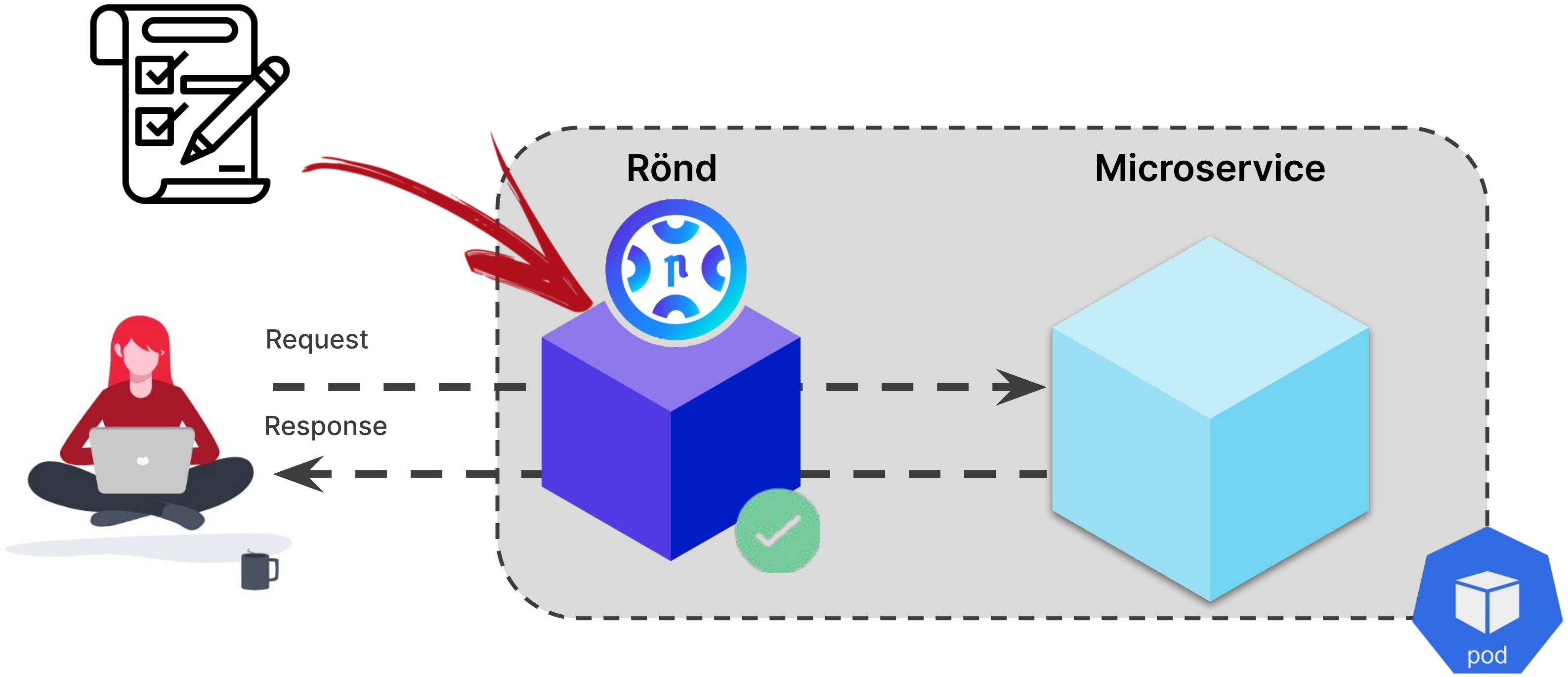
# Rönd flow authorizing the user request



## Rönd flow generating queries



## Rönd flow that modifies the results



## Write configuration and Policies

## What does the configuration looks like?

```
{
  "paths": {
    "/example": {
      "get": {
        "x-rond-config": {
          "requestFlow": {
            "policyName": "greetings_read",
            "generateQuery": true,
            "queryOptions": {
              "headerName": "x-acl-rows"
            }
          }
        }
      },
      "responseFlow": {
        "policyName": "filter_response_example"
      },
      "options": {}
    }
  }
}
```

## What does the configuration look like?

**Automatic configuration**: if your service exposes its own **OpenAPI Specification** documentation, Rönd can fetch it and dynamically route the APIs towards the corresponding policy

## How to write a policy (Rego 101)

Let’s dive a bit into how you write a Rego policy that works with **Rönd**!

The examples are taken from the github.com/rond-authz/example repository

**https://bit.ly/3p9hWu6**

## How to write an allow policy

```
verify_authorization {
  authorization := input.request.headers.authorization[0]
  authorization == "bearer 123456"
}

verify_authorization {
  input.request.headers.authorization[_]
}
```

```
"request": {
    "headers": {
      "authorization": [
        "bearer FALSE",
        "bearer FALSE",
        "bearer FALSE",
        ...............
        "bearer 123456"
      ]
    }
  }
```

## How to write an allow policy

```
verify_authorization_and_filter {
  input.request.headers.authorization[_] == "bearer 123456"

  resource := data.resources[_]
  resource.name = "Mich"
  resource.surname = "Murabito"

}
```

```
      // It's like

      // WHERE name = 'Mich' and surname = "Murabito"
```

## How it works

```
{
  "paths": {
    "/example": {
      "get": {
        "x-rond-config": {
          "requestFlow": {
            "policyName": "verify_authorization_and_filter",
            "generateQuery": true,
            "queryOptions": {
              "headerName": "x-acl-rows"
            }
          }
        }
      },
      "responseFlow": {
        "policyName": "basic_parser"
      }
    }
  }
}
```

## Final thoughts

# Final thoughts

Thanks to **Open Policy Agent** we managed to build a blazing fast, general purpose, microservice that can be easily introduced in any application with little,
to none, changes in your codebase.

Rönd is **actively maintained** as we are constantly improving it and keeping it up to date. We would love to hear from you, so don’t hesitate to let us know what do you think about!

## Useful resources

- https://rond-authz.io
- https://github.com/rond-authz/rond
- https://www.openpolicyagent.org/
- https://blog.mia-platform.eu/en/how-why-adopted-role-based-access-control-rbac
- https://blog.mia-platform.eu/en/announcing-rond-new-open-source-security-enforcement-over-your-apis