Building Kubernetes Operators using Jakarta EE and MicroProfile

RedBridge

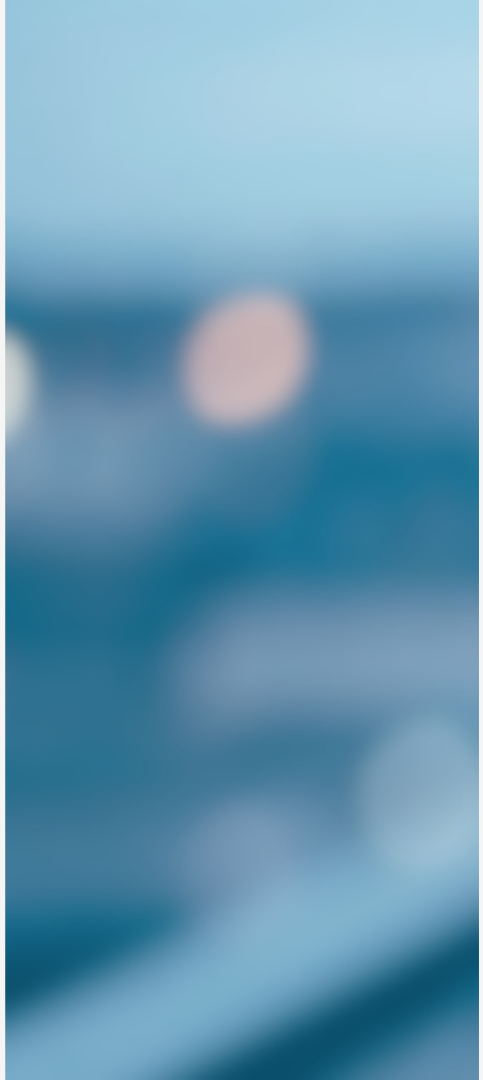
Who is that guy?

Daniel Pfeifer

Principal Consultant - Application Engineering
RedBridge Technology AB (part of Binero Group)

daniel@redbridge.se
linkedin.com/in/danpfe

RedBridge

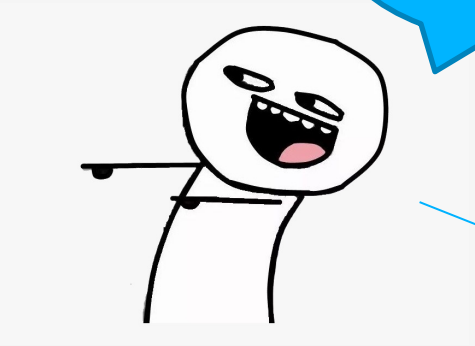


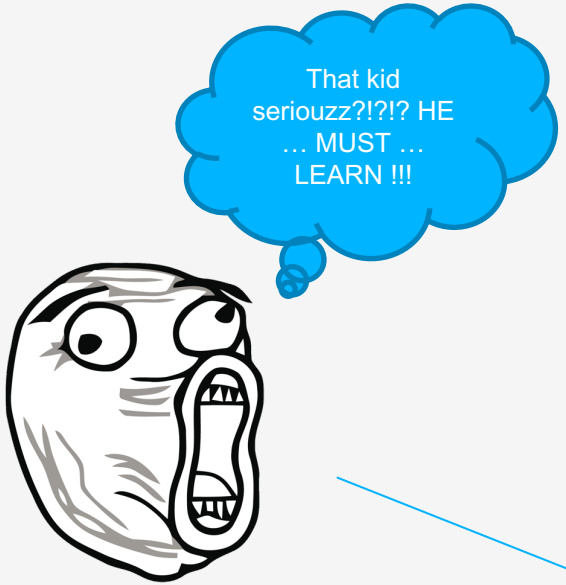
What I'll talk about

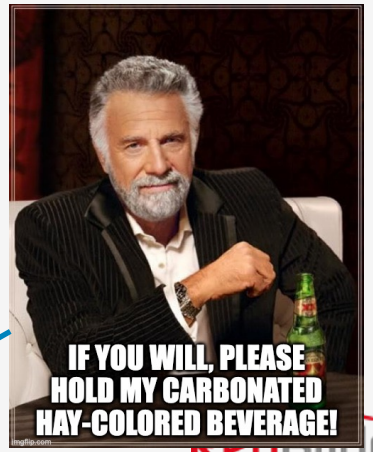
- Why an operator?
- The parts of an operator
  - Reconciler
  - Validation
  - Self-documenting
  - HA
  - Ownership
  - Custom API
- Why Java combined with Jakarta EE and MicroProfile?
- Demo
- Wrapping up

RedBridge

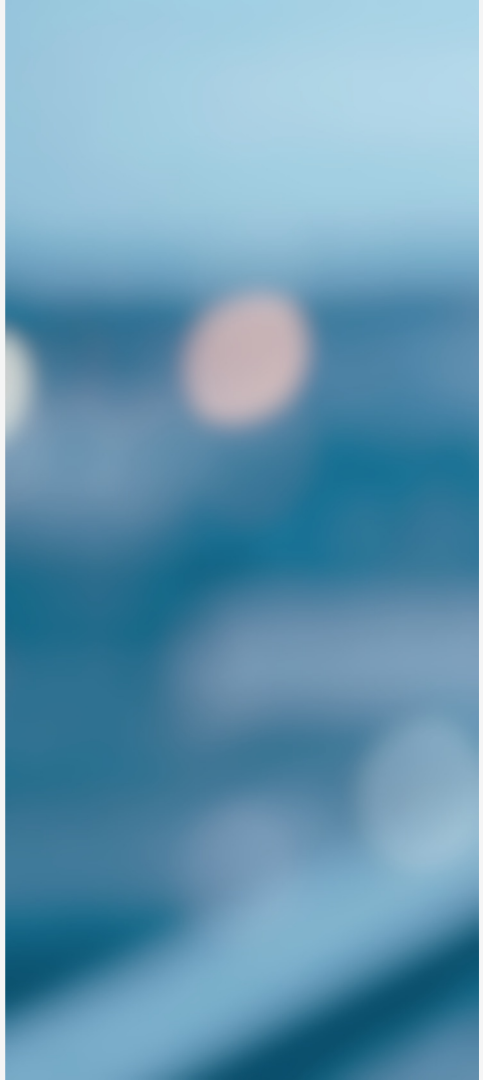
Why th







The more composed version of me

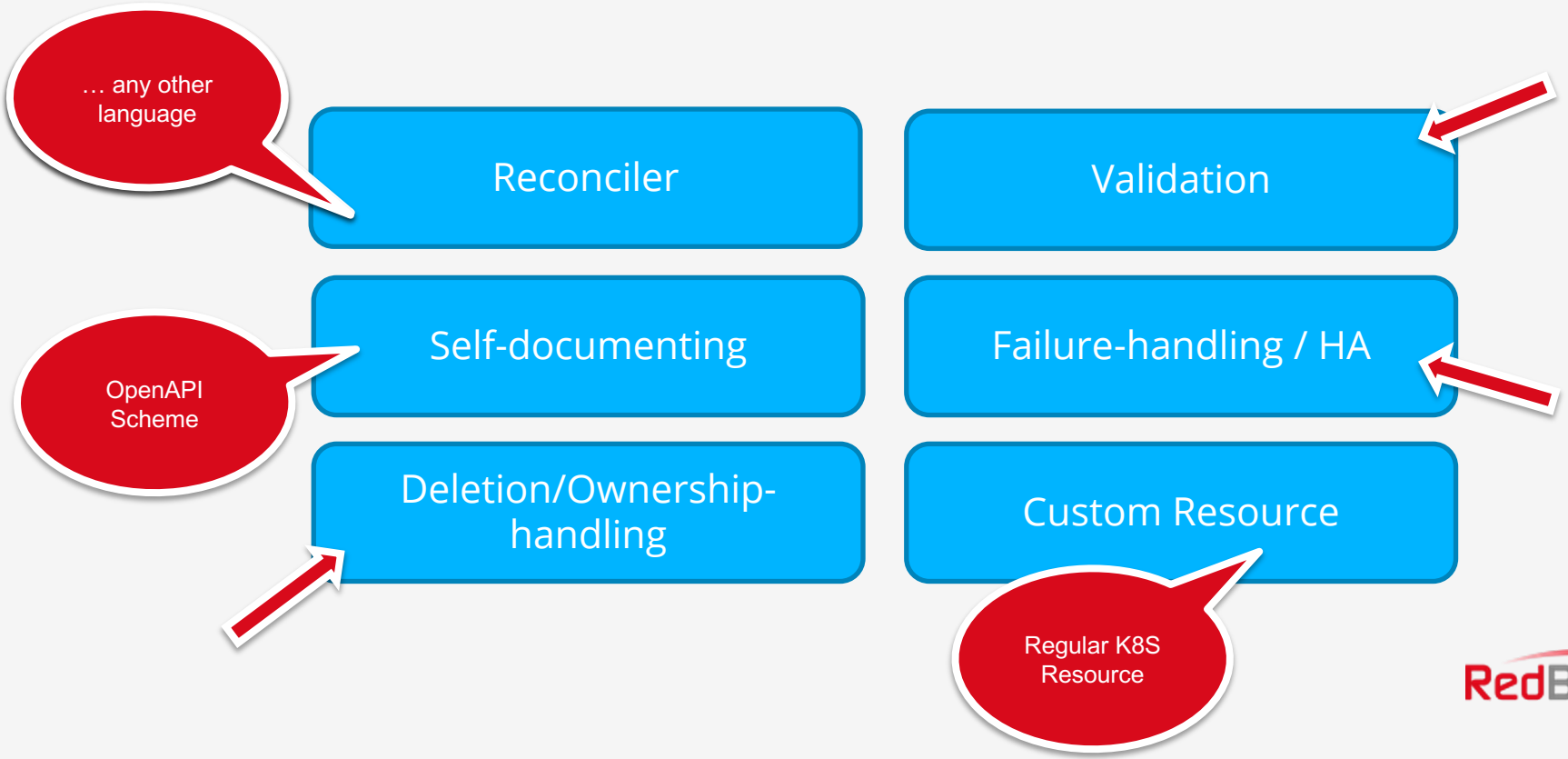


Why an operator?

- Augments Kubernetes with new resources.
- Enables the automation of manual and repetitive tasks through a well-defined API.
- Examples:
  - Infrastructure-as-Code through Kubernetes Resources (i.e. create a client in an externally located Single-Sign-On-service).
  - Set-up and manage the lifecycle of in-cluster middleware (setup a database and schedule daily backups).
  - Get external data and populate Kubernetes-native resources with the result (i.e. fetch the current stock price and store it in a ConfigMap for all your Pods to easily read).

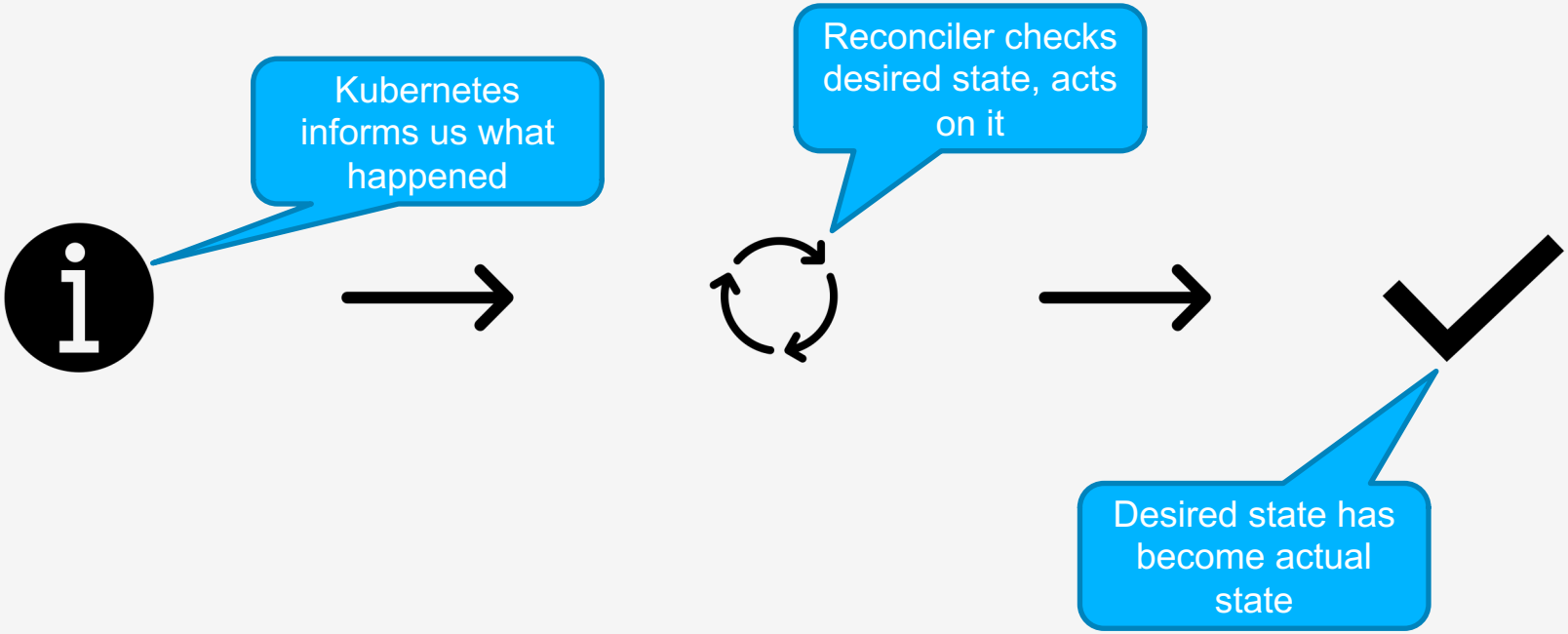
RedBridge

The parts of a production-ready operator



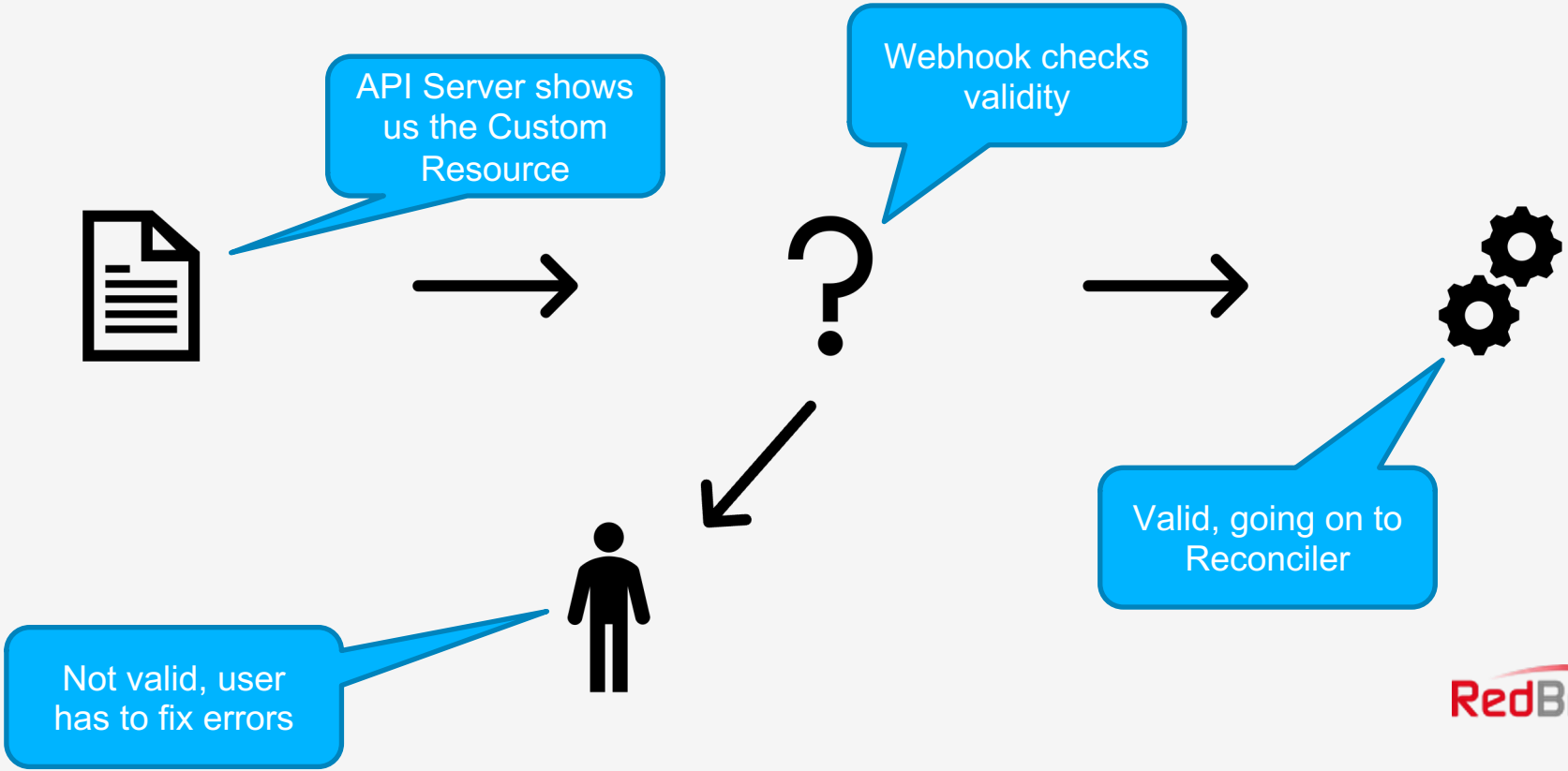
RedBridge

Reconciler



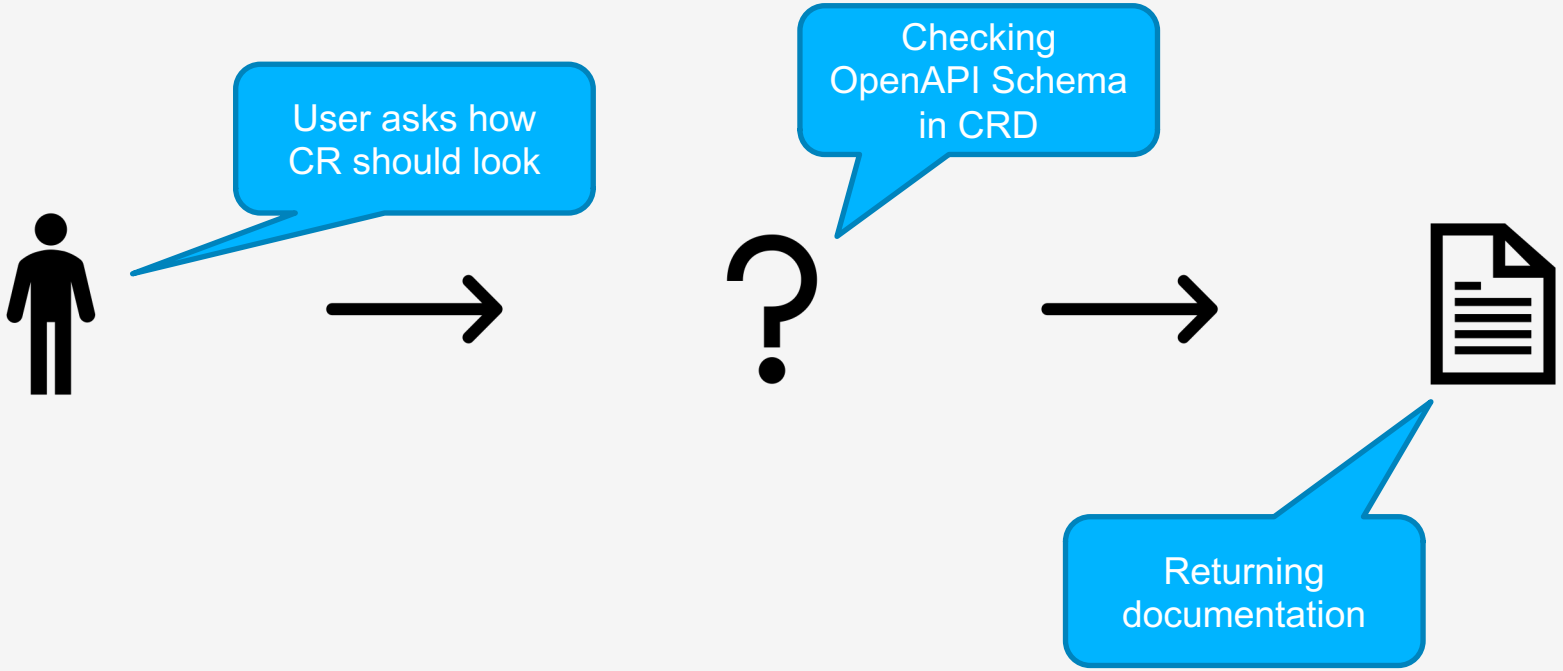
RedBridge

Validation



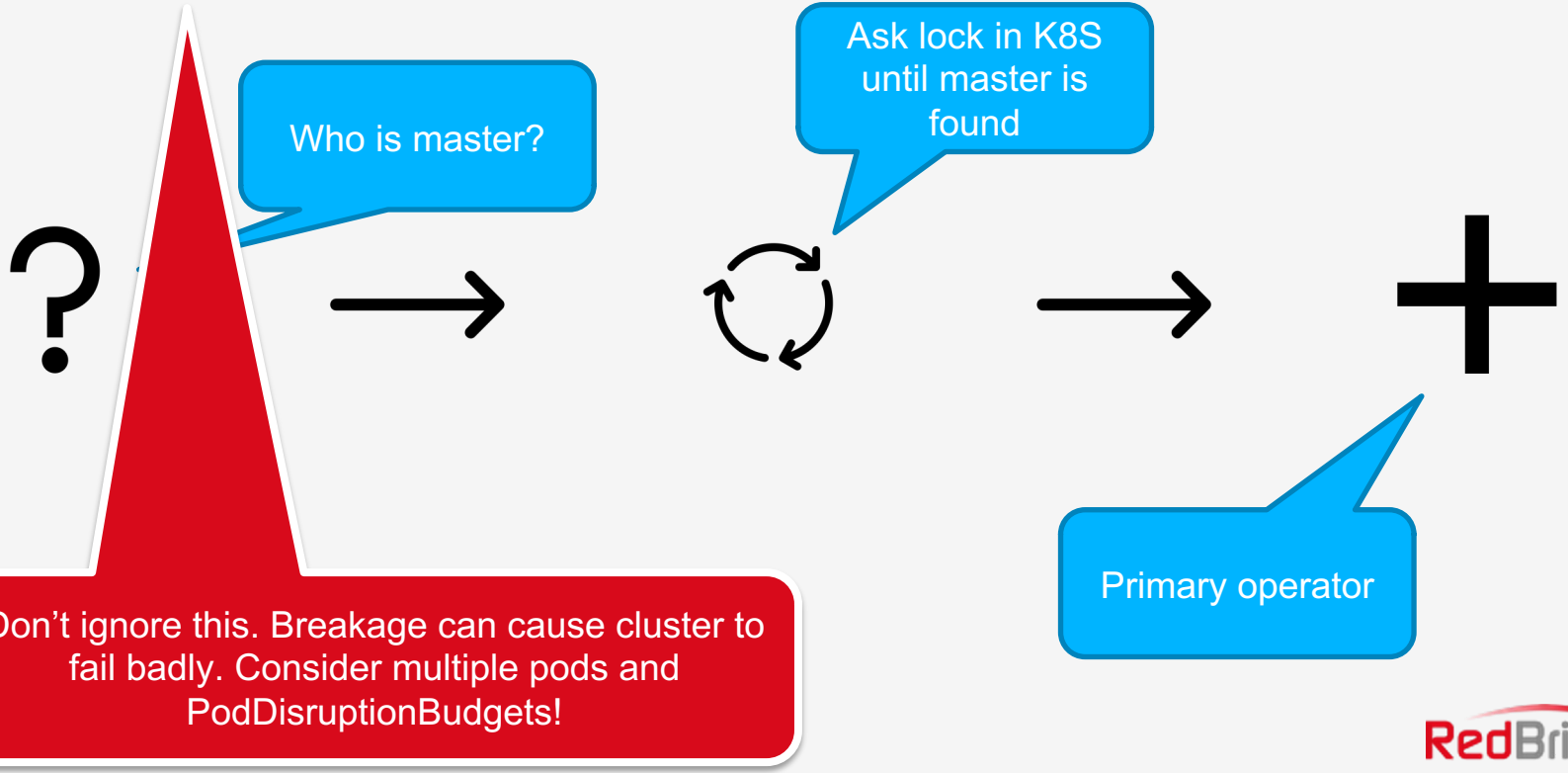
RedBridge

Self-documenting



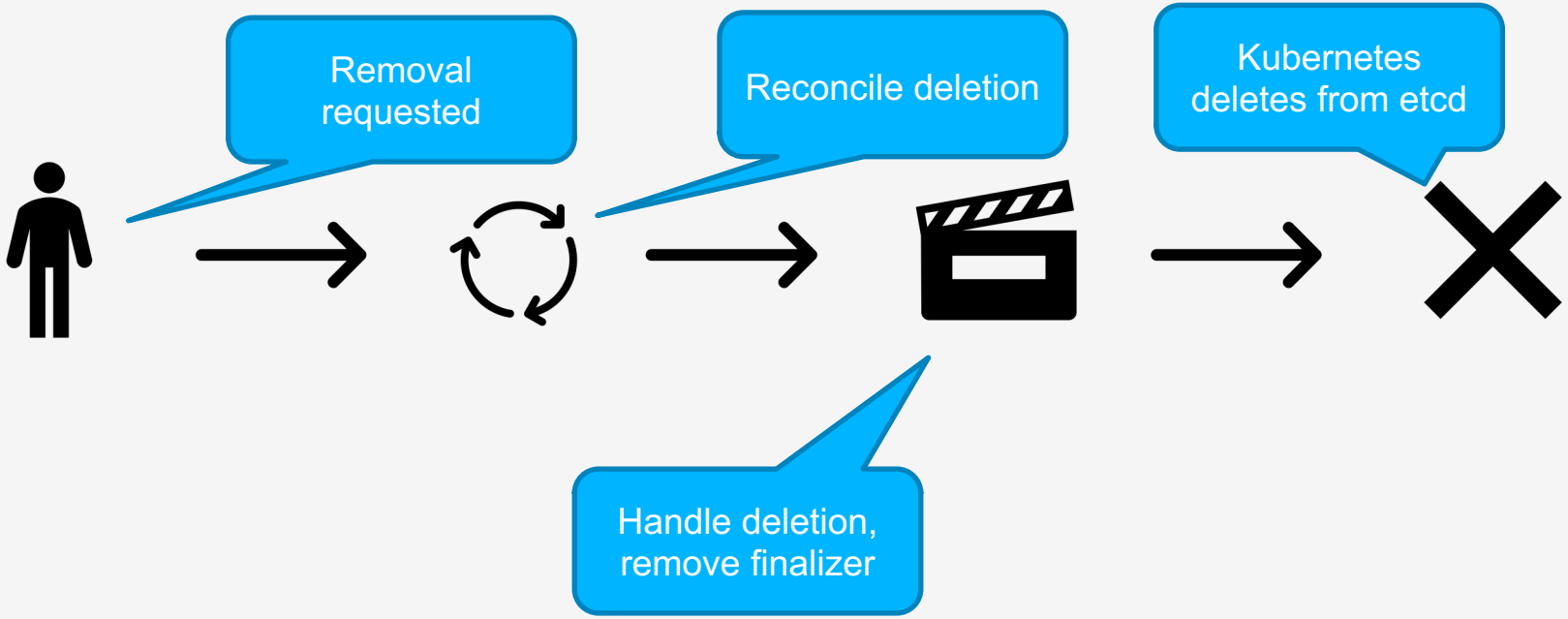
RedBridge

High Availability



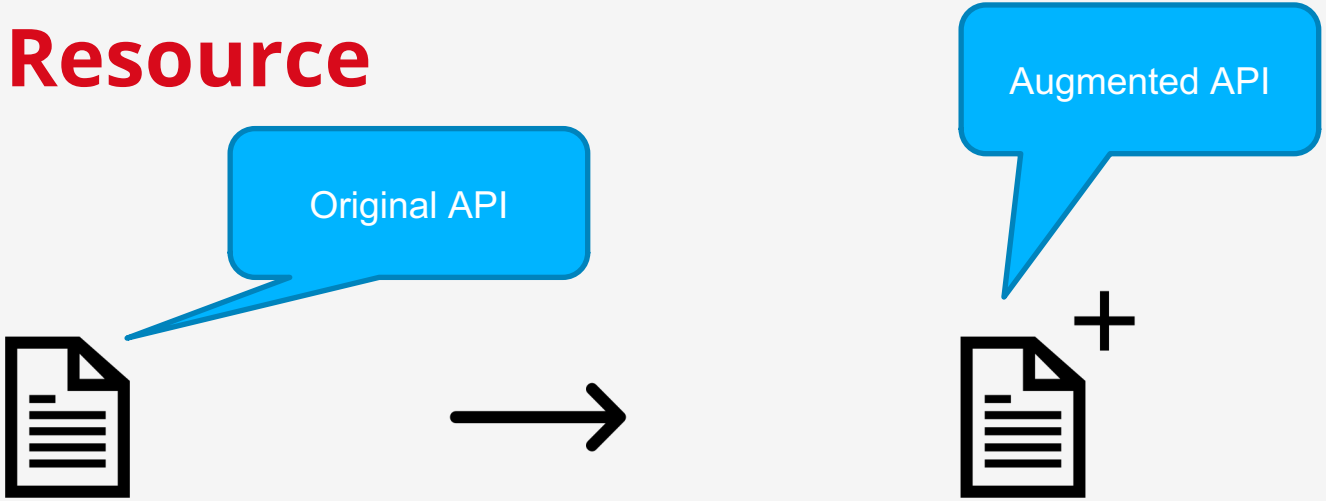
RedBridge

Ownership



RedBridge

Custom Resource



RedBridge

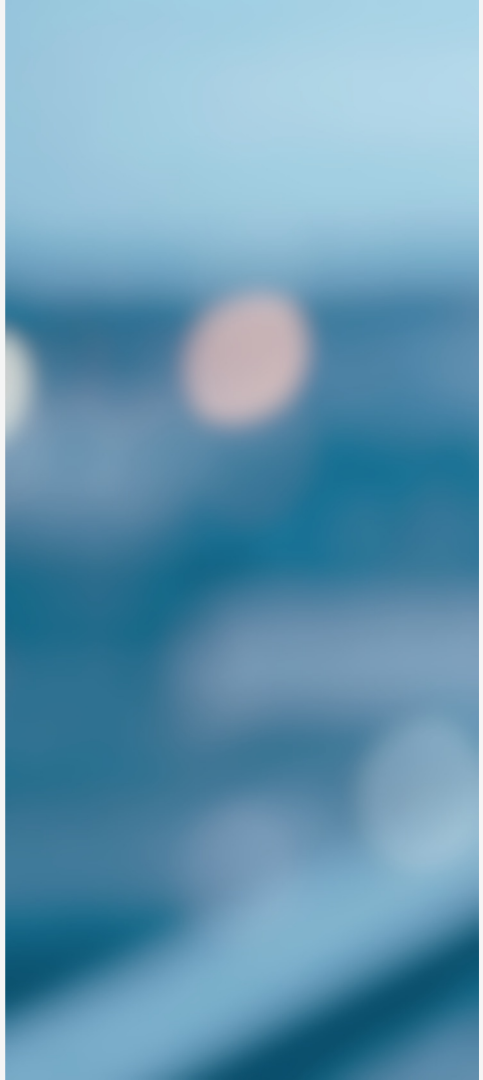
Why Jakarta EE and MicroProfile

- **Familiar, few new skills required.**
- **Light-weight and fast (and IMHO it's nice to have many competing open source implementations to chose from).**
- **Contains specs we need (JSON, REST) and provides plenty of additional capabilities to use outside services if we need to integrate (like Pub/Sub, Queues, Databases and so forth).**
- **Out-of-the-box features to integrate with Kubernetes.**

RedBridge

Walkthrough of an operator

RedBridge

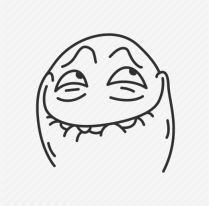


Wrapping up

- An operator and its' extras are really just a fancy REST client/server, why not in a framework and language you know?
- Java (and Jakarta EE+MicroProfile) fits in a container-world, perhaps even better than many other languages because us Java-devs have had a plethora of parameters to optimize our apps to specific workloads for many years.
- Kubernetes as our god-OS is not as impenetrable as you think, it's primarily dealing with JSON and HTTP, dare to bend it to your needs!

RedBridge

Thank you!



Good links to check:

- **https://redbridge.se**
  My employer, a provider of consulting services within development, observability, infrastructure … and offers a EU-based Public Cloud. Ask me for a voucher if you want to try.
- **https://github.com/danpfe/eclipsecon**
  The demo code.
- **https://linkedin.com/in/danpfe**
  My LinkedIn.
- **https://home.robusta.dev/blog/stop-using-cpu-limits/**
  Some easily agreeable thoughts on K8S CPU Limits.
- **https://ee.kumuluz.com**
  The Jakarta EE / MP-framework used in the demo code.
- **https://github.com/fabric8io/kubernetes-client**
  A nice Java K8S Client for Kubernetes.

RedBridge