KUEHNE+NAGEL

AsyncAPI For Platform Self-Service: A GitOps Tale

AACoT'23 apidays PARIS

Hello world! 🤚

João Dias
Software Engineer @ Kuehne+Nagel
Invited Assistant Professor @ University of Porto

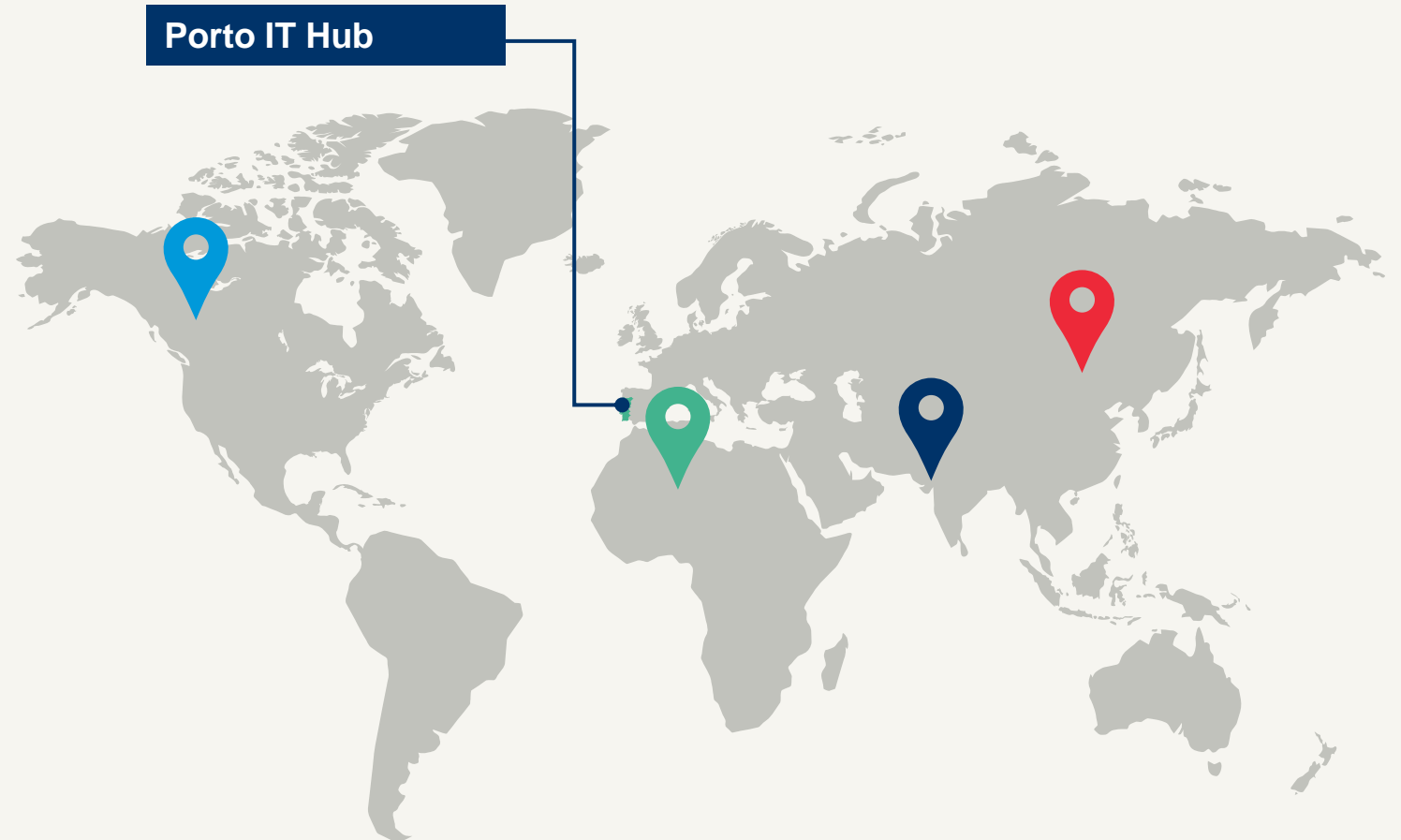📮 joao.dias@kuehne-nagel.com

📓 in/joaopdias

🔗 https://jpdias.me

Rui Eusébio
Senior Software Architect @ Kuehne+Nagel

📮 rui.eusebio@kuehne-nagel.com

📓 in/rui-eusebio

KUEHNE+NAGEL

Porto IT Hub

The Kuehne+Nagel Group at a Glance

400 000

customers trust us to manage their logistics

Nearly 1 300

offices worldwide, so that we are close to our customers

100

countries, connected by our network

Over 80 000

logistics and supply chain professionals who give their best everyday.

No. 1

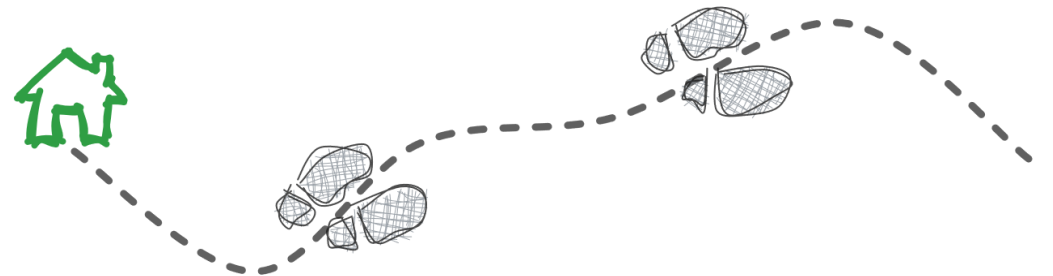sea freight and air freight forwarder worldwide.

130 years
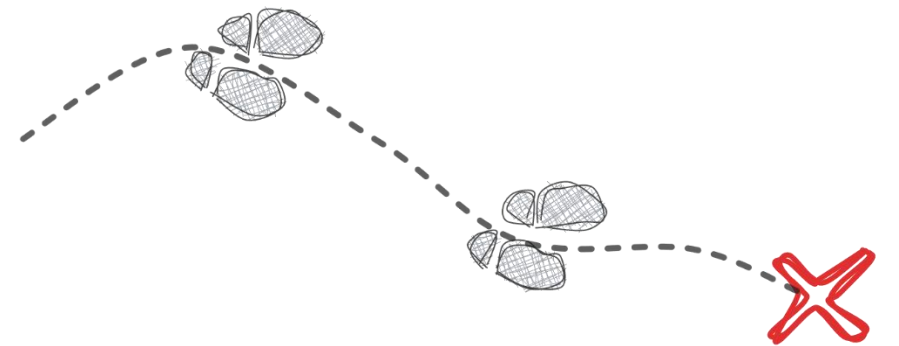
founded in 1890 by August Kuehne and Friedrich Nagel

Providing Kafka for Kuehne+Nagel teams

We provides an *Internal Developer Platform (IDP)* for the Kafka ecosystem…

…and we are becoming overwhelmed with operational tasks to meet users' needs in a timely fashion.

The solution?
**Empower users
to manage their
own resources.**

KUEHNE+NAGEL

KUEHNE+NAGEL

The trade-off?

Giving more autonomy to platform users relies on automated validations to maintain governance at scale.

Governance: the Missing Link?

- **We oversee and manage the IDP usage by:**
  - Providing Guidelines and enforce their fulfilment.
  - Architectural and development support/guidance.
  - Tutorials and best-practices documentation.
  - Requiring API specifications with AsyncAPI (with automatic and manual reviews).
  - Enforce usage policies across environments.
  - ⧖ Creation of service accounts.
  - ⧖ Access control management (i.e., permissions for topics).
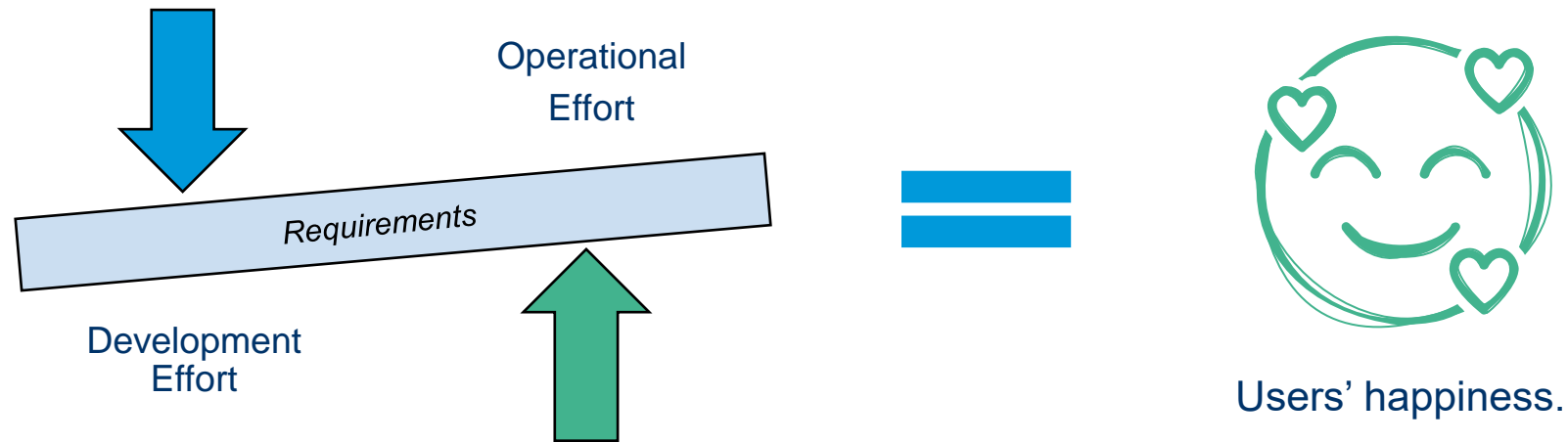 - ⧖ Facilitating agreements between teams.

Moving forward

Reduce intermediary efforts while leveraging documentation
as a first-class citizen:
API contract-first approach for Kafka

The Paradox of Decision-Making

KUEHNE+NAGEL

Operational Effort

Requirements

Development Effort

=

Users' happiness.

Choosing the best approach is hard given that technology is constantly changing; and **we want to keep evolving with minor impact to our users.**
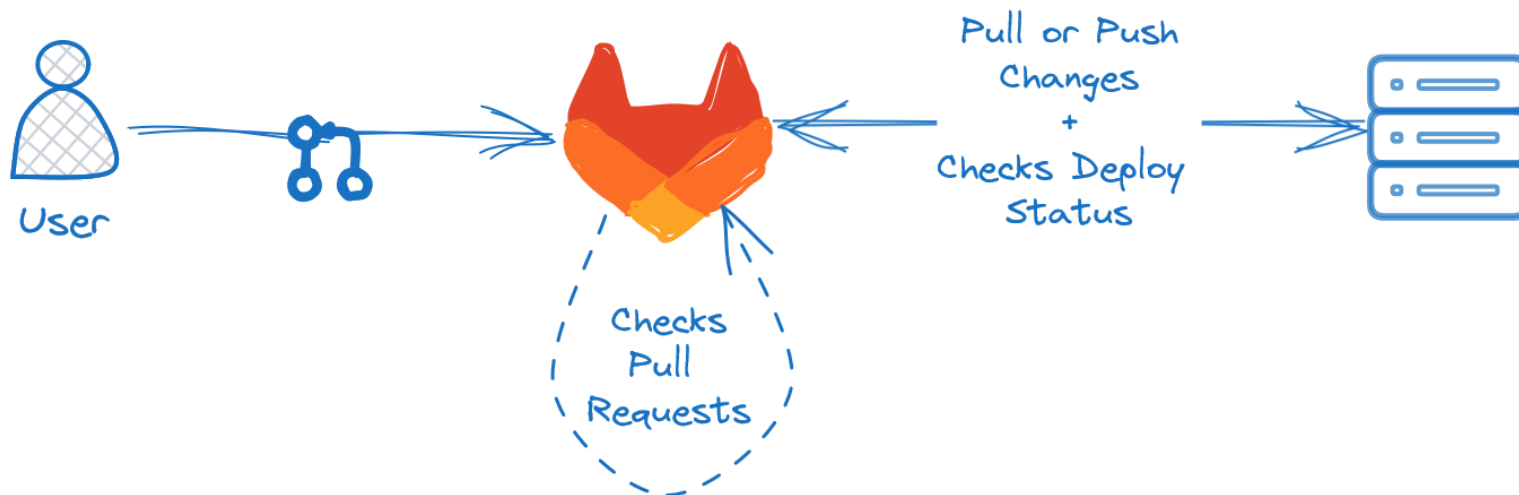
Expected Outcome

Automatically transform user's specifications into infrastructure configurations fostering the reduction of operational burden by the IDP team, without requiring added effort from users.
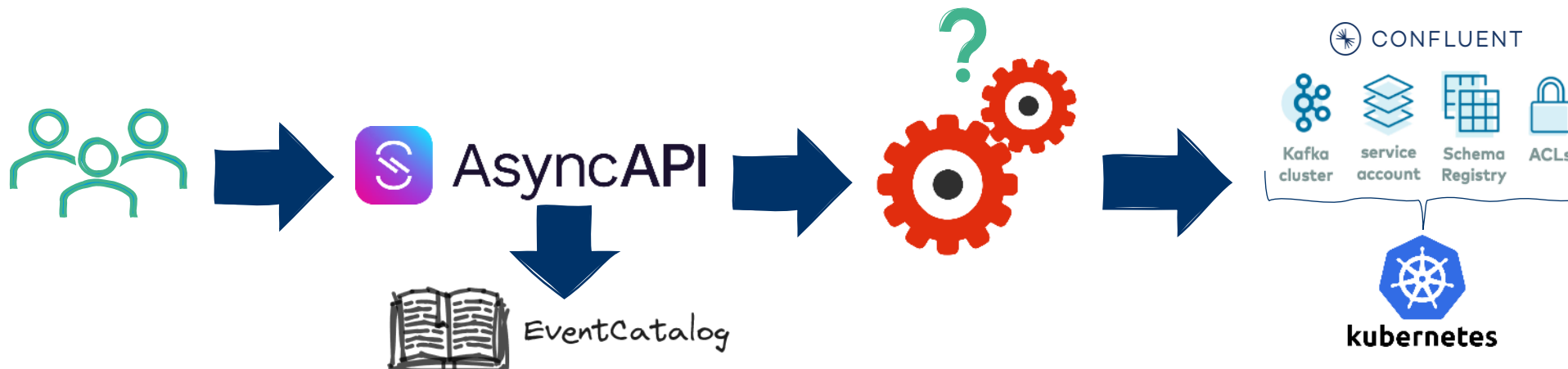
# Embracing Uncertainty with AsyncAPI and GitOps

GitOps

"GitOps is an operational framework that takes DevOps best practices used for application development such as **version control**, **collaboration**, **compliance**, and **CI/CD**, and applies them to infrastructure automation."

– What is GitOps?, https://about.gitlab.com/topics/gitops/

User

Checks
Pull
Requests

Pull or Push
Changes
+
Checks Deploy
Status

Continuous Integration and Deployment

Keeping the **continuous** in the CI/CD **but segregate the responsibilities** of integration from deployment.
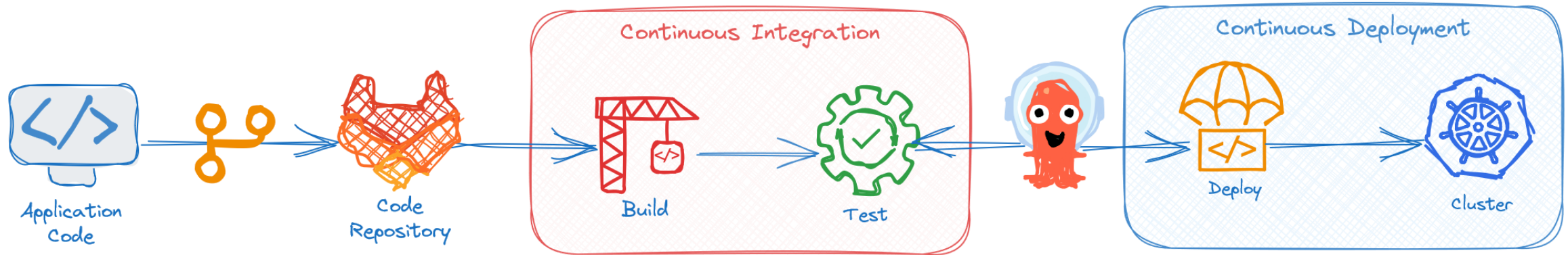
User Interaction Flow

- **Pull requests are opened by users:**
  - Automatically checked for syntax errors and common semantic mistakes.
  - Reviewed by an IDP team member when needed.
- **Collaboration and documentation process:**
  - Subscriptions to API can be allowed/denied by API owners.
  - Documentation reflects the state of the cluster.
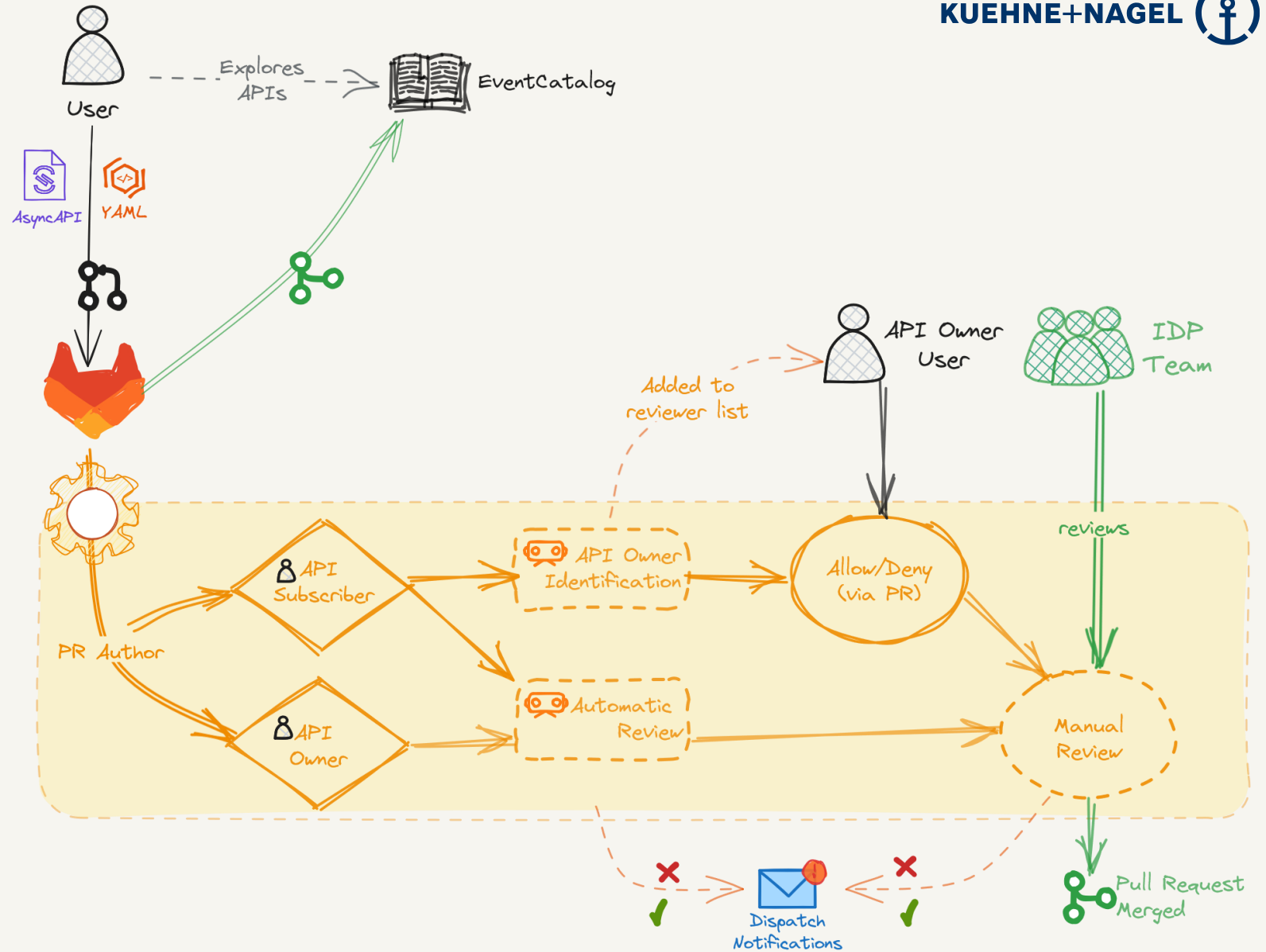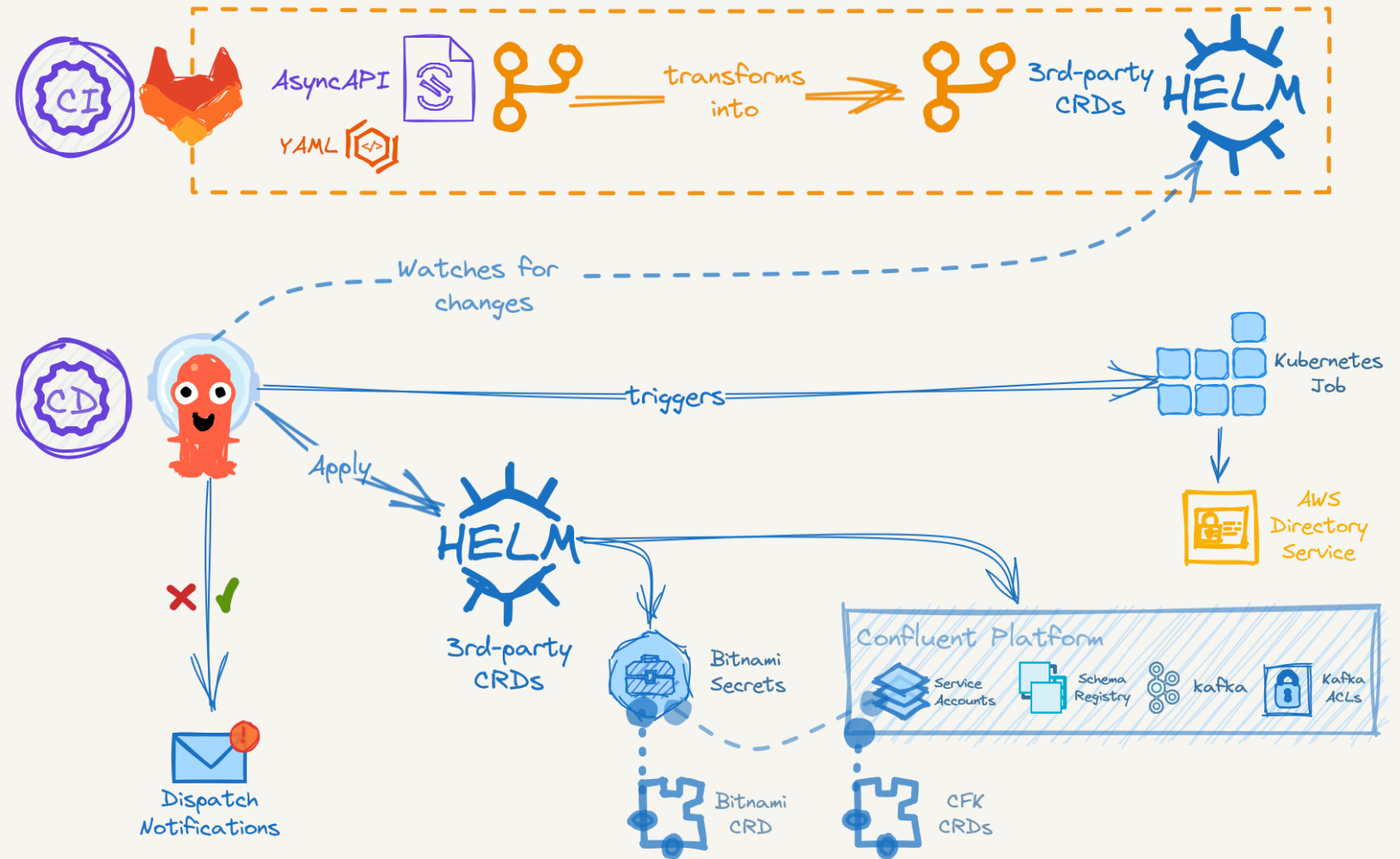
CI/CD
External Provider Version

- **On merge:**
  - ArgoCD **pulls** changes from git main branch.
  - ArgoCD automatically applies the HELM charts in the cluster.
- **Key takeaways:**
  - Out-of-the-box deployment failure handling and notifications support.
  - More modularity but less extensibility (3rd-party).
  - More "fire-and-forget" approach due to jobs.
 - Everything should be defined in git (default way to interact with cluster).

CI/CD
Operator Version

- **On merge:**

  - ArgoCD **pulls** changes from git.

  - Custom CRDs are applied to the cluster and deployed/managed by the operator.

- **Key takeaways:**

  - Out-of-the-box deployment failure handling and notifications.

  - Centralized cluster operation via Kubernetes operator.

  - Extensability to other interfaces beyond Git via operator API.

KUEHNE+NAGEL
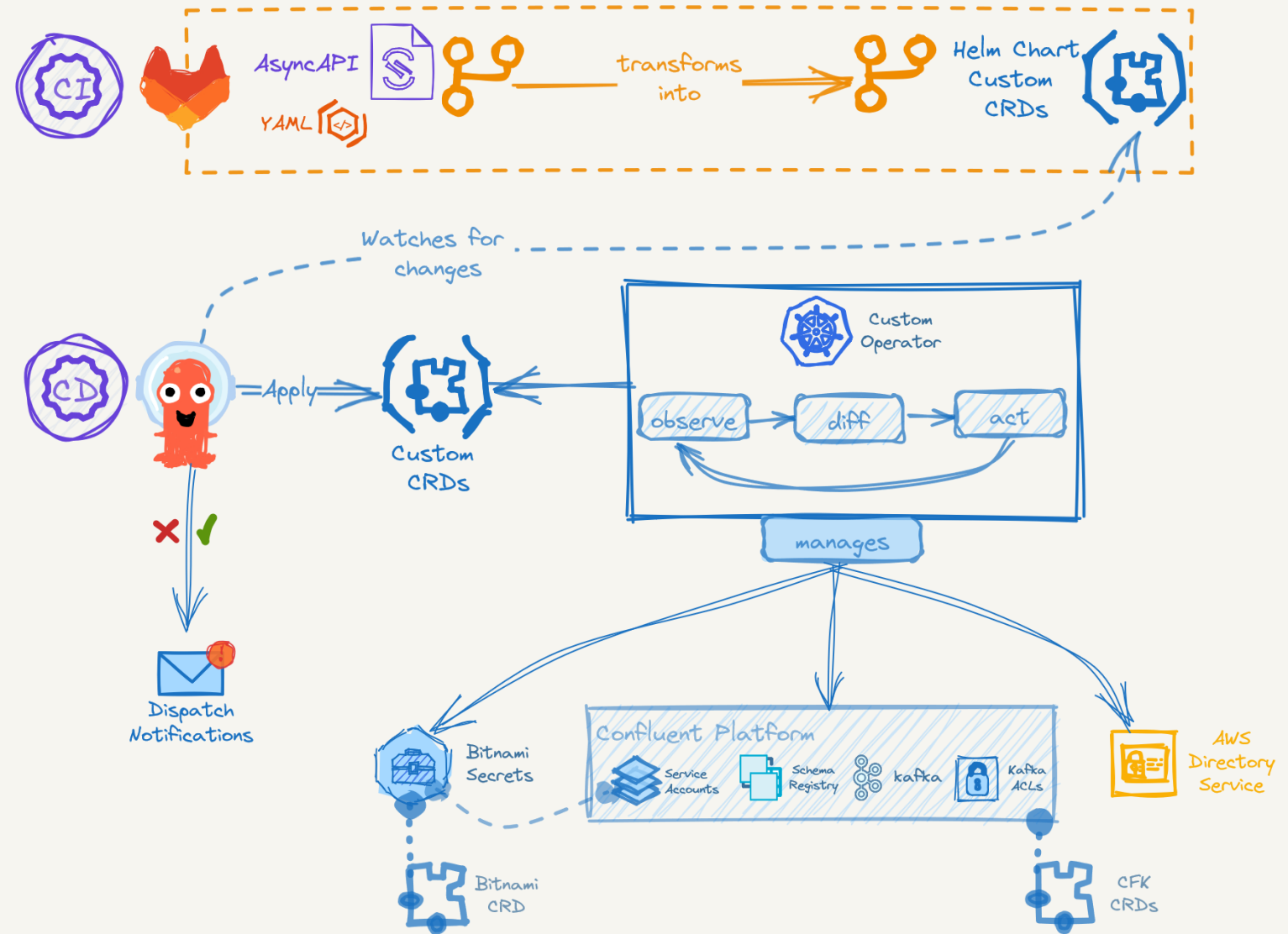
"The more significant the audience, the more catastrophic the failure"
– *Murphy Law Corollary*

Demo

🖥 **https://github.com/RuiEusebio/confluent-selfservice**

Key Takeaways

KUEHNE+NAGEL

- **Standardization of processes via contract-first approach (AsyncAPI)**
  - Allows better automatic validations (fewer one-off mistakes).
  - Pull requests are manually reviewed only when needed.
- **Built-in version control and deployment management**
  - Easier to revert mistakes (git) and better failure handling (rollback)
- **Improve collaboration and reduce manual handshakes**
  - When required, API Owners are automatically added as reviewers.
- **Future-proof internal specification for resources**
  - Custom YAML helm chart that is created by transformation of AsyncAPI.
  - More resilient to AsyncAPI version changes (*v3 is HERE!*).
  - Some implementation effort on the "transformation" from spec to charts.

Next Steps

- ⏩ Improve AsyncAPI validator to reduce manual review steps and provider better documentation for users.
- ⏩ Complete the development of the main building blocks, initiate the beta testing, and encourage some teams to start using the new functionality (feedback-loop).
- 🔄 Enhance *EventCatalog* navigation by visually displaying the information about API owners and providing links to the corresponding AsyncAPI files.
- ⏩ Close the gap between the AsyncAPI specification and the deployed infrastructure.
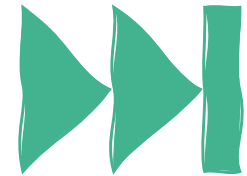
Closing Remarks

- **Empower platform users' autonomy with self-service**
  - Users can create service accounts and grant topic access to other users.
  - Reducing the necessity for IDP team to act as intermediaries.
- **Cohesive approach but open for extension**
  - AsyncAPI as the default entrypoint for usage of the IDP.
- **Breaking down silos by ensuring that documentation is consistently generated and updated through AsyncAPI**
  - Users can use EventCatalog to explore currently available APIs as well as the message schemas, headers and other relevant docs available on AsyncAPI.
  - Users can quickly jump from EventCatalog to request topic access via Git PR.
- **Reducing operations burden to free up the team's time**
  - Team can focus on core features and on providing advanced customer support.
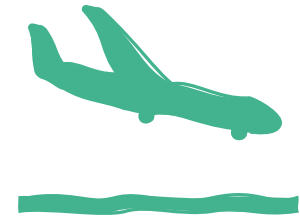
Thank you!

KUEHNE+NAGEL

Inspire. Empower. Deliver.

KUEHNE+NAGEL