PUBLIC

Red Hat Summit ‘23 – NS Case study

How simplification boosts adoption of a hybrid cloud integration platform

Jack Fleuren (Lead Product Owner)

Taco Nieuwenhuis (Solution Architect)

Floris Alfverink (Platform engineer)

CPO Centrale Platform Organisatie

Let us introduce ourselves

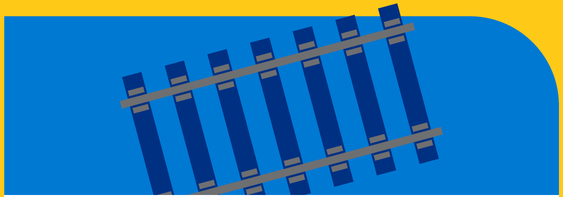
Jack Fleuren
Lead Product Owner

Taco Nieuwenhuis
Solution Architect

Floris Alfverink
Platform Engineer

We are the largest passenger rail transport company of the Netherlands

Facts & Figures



2.100 kilometers of railway

1 million travelers per day

Approximately 4.800 train journeys per day

IT and data exchanges are critical in running daily train operations…

IT applications

Train sensors for fleet management and health checks during operations

Displaying current departure times on platform displays

Inventory tracking of retail stores and sales data from POS systems

Execution of the train schedule and/or respond to deviations

Application domains at Dutch Railways

- Develop, sell and maintain **commercial propositions**
- Develop and execute **surveillance** and **enforcement** on trains and stations
- Develop, plan and adjust **personnel** and **train schedules**
- Develop and **maintain trains** and other company assets
- Development of **train stations** and surrounding areas
- Operations of train stations and **mobility services**
- **Retail activities**

Traditional datacenter /
Private cloud

Public cloud hosting

SaaS hosting

Integration Landscape

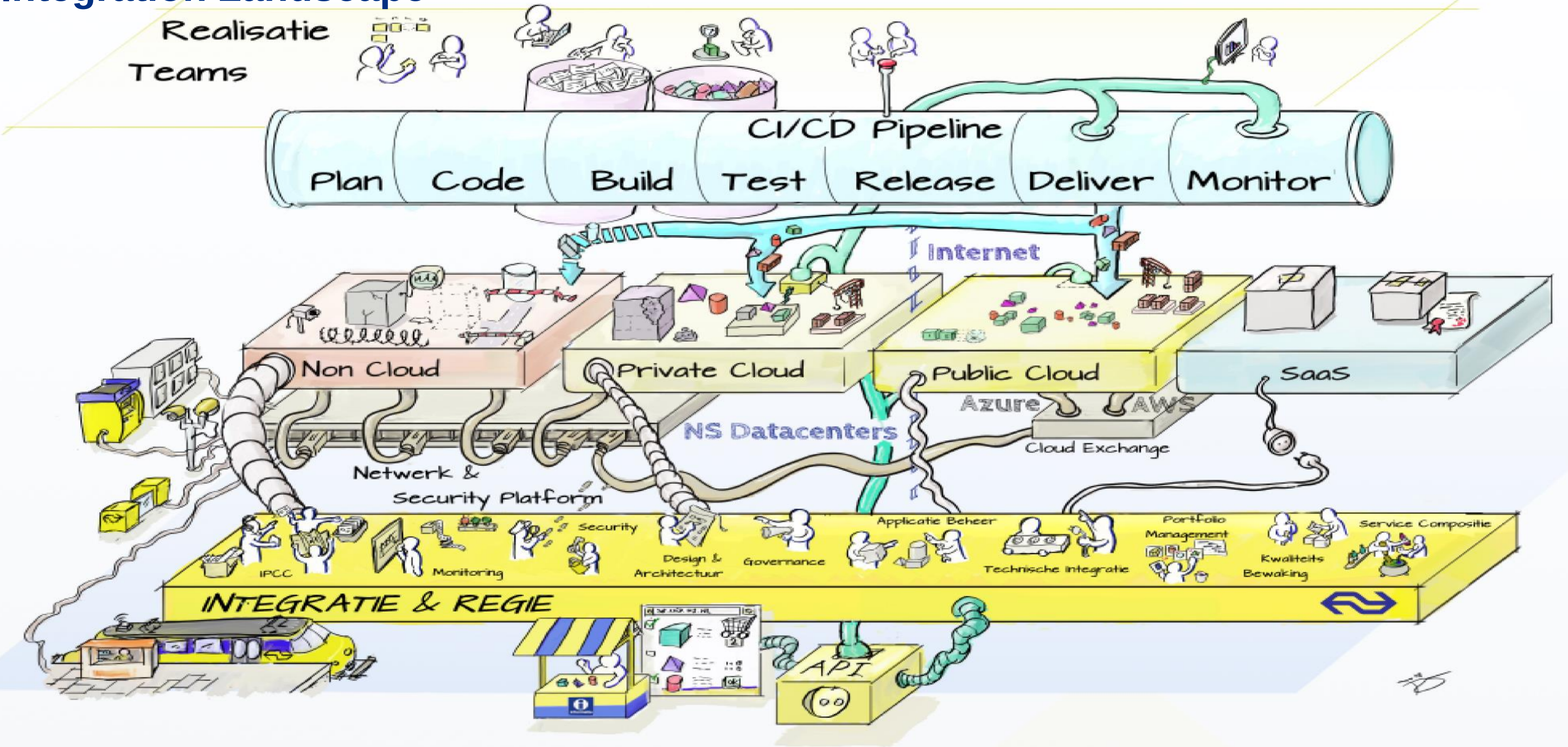


Illustration: Remco Schellekens

Our situation

Current integration solution is reaching the end of its product lifecycle

Our challenge

Complete migration in time

Change to Devops operating model

Change to Devops operating model

Learning curve & tech stack

Complexity

Ops responsibility

Hybrid Integration Platform: triple-P demarcation

Three Protocols:

- HTTPs
- AMQPs
- MQTTs

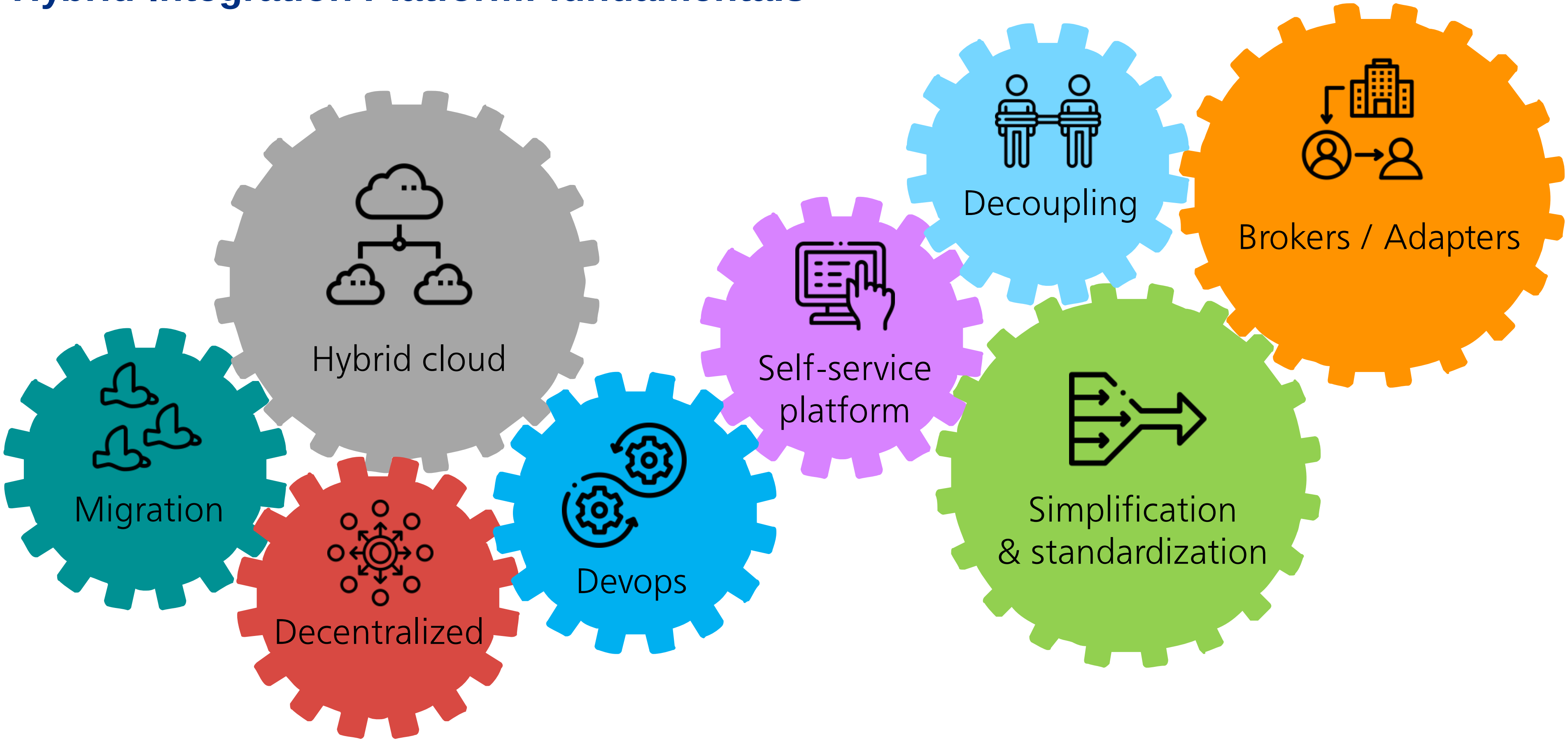
Three Product Suites:

- Azure API Management
- Red Hat AMQ Integration Suite
- Apache Nifi

Three Priorities

1. Synchronous REST / SOAP APIs
2. Asynchronous Messaging / Events
3. Workflow Management

Hybrid Integration Platform: fundamentals



Hybrid cloud

Migration

Decentralized

Devops

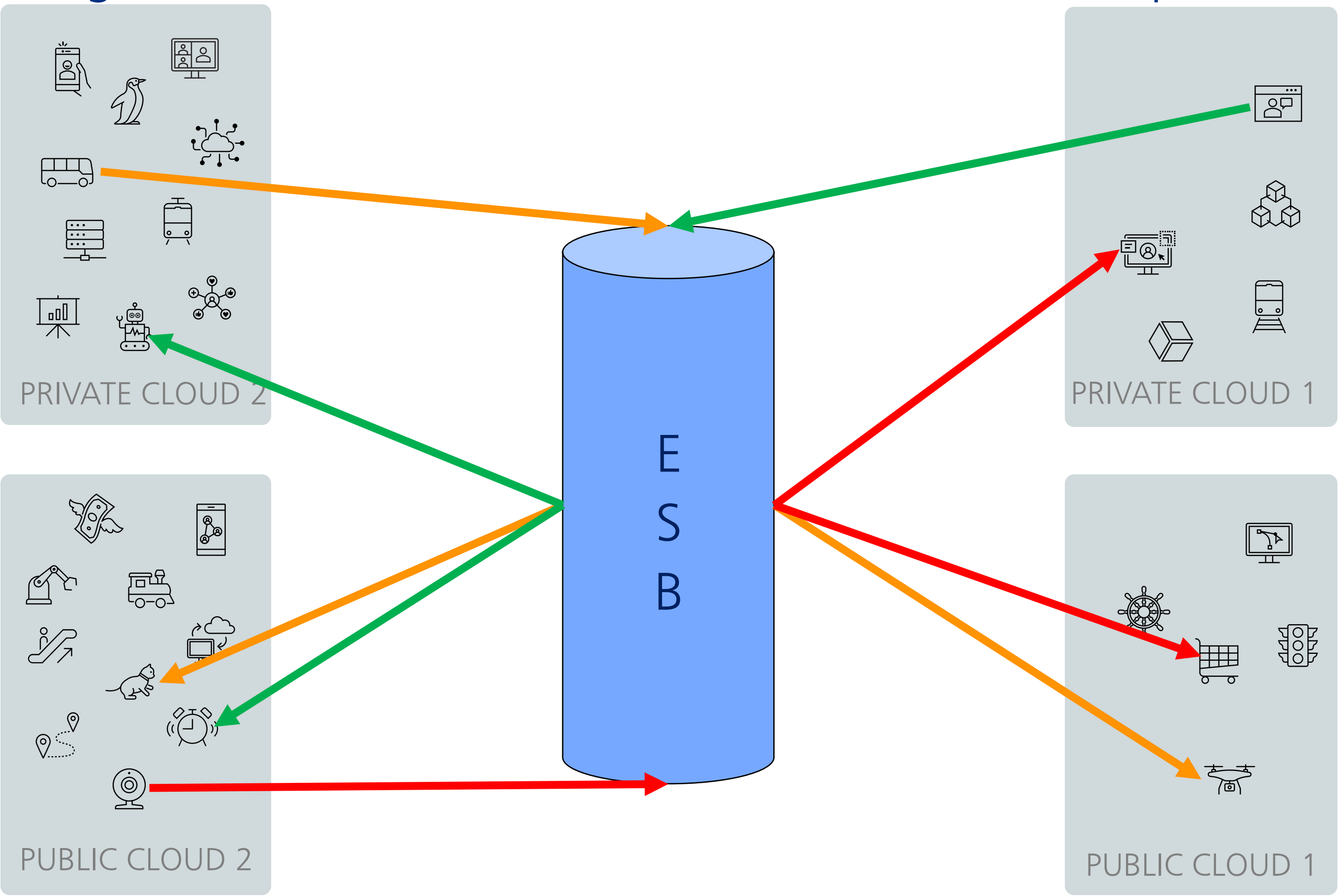
Self-service platform

Decoupling

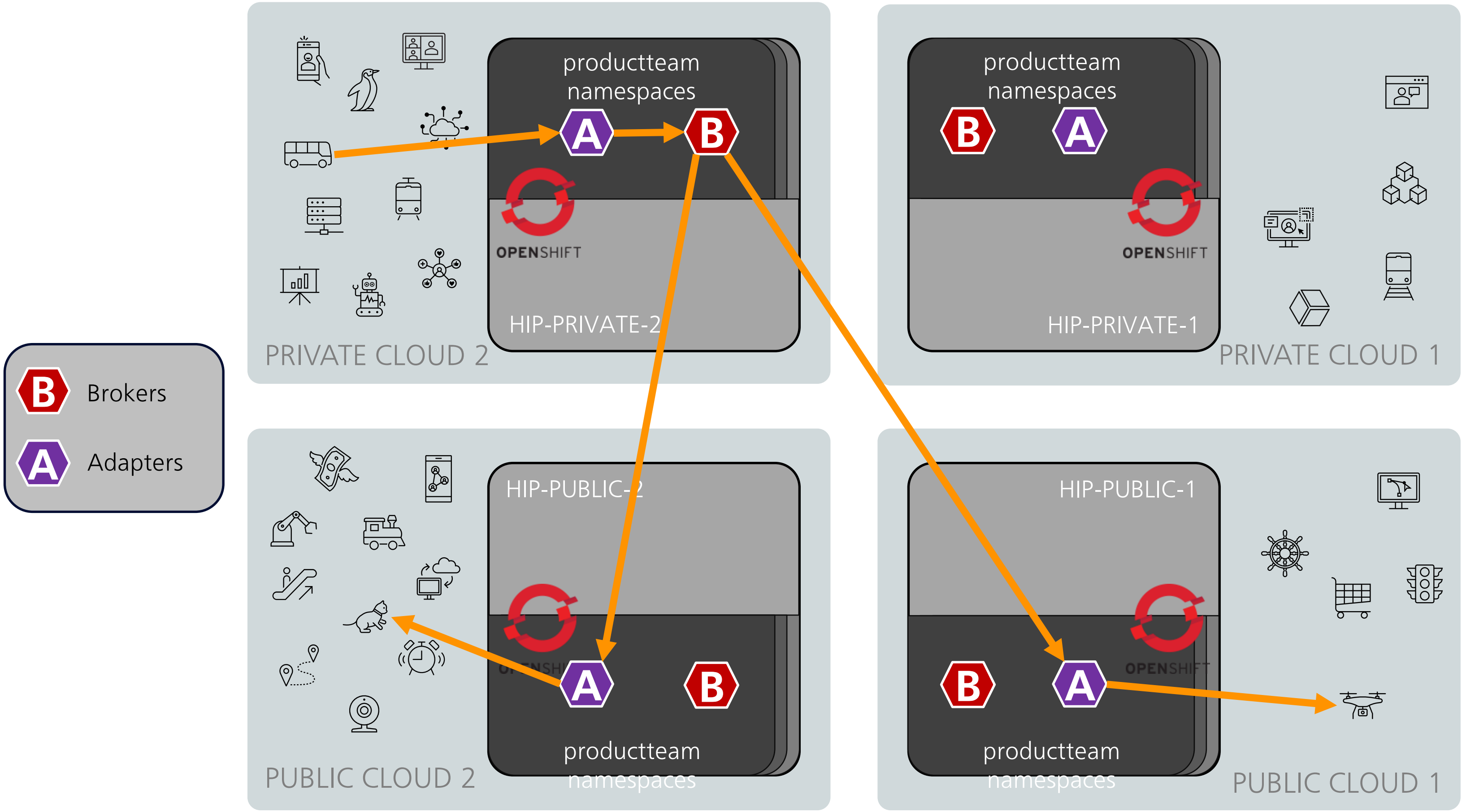
Brokers / Adapters

Simplification & standardization

Old integration architecture: centralized and lots of dependencies



HIP Phase 1: decentralisation, self-service and standardisation



Which objectives have been accomplished?

Enabling of migration away from ESB

Local brokers become mini-ESBs, adapters offer all possible enterprise integration patterns

Decentralized architecture

No centralized integration function anymore but distributed architecture equiped with CI/CD pipelines to enable teams to build and run their own integrations.

Self-service platform

NS-wide iPaaS offering connected to central functions like logging, monitoring, security, IAM under central Life Cycle and Resource Management.

Standardised building blocks

Preconfigured and prevalidated integration components to increase flexibility and minimise dependences. Reuse of technology and expertise.

Which adoption challenges remain?

Learning curve and tech stack

Brokers and adapters are complex components. Not every team has the expertise or ambition to adopt their technology.

Architectural complexity

Obligation to use stateful integration brokers is not always feasible, adapters as bridges are sometimes overkill. Transparancy can be improved.

Ops responsibility

Not all devops teams have the capacity of organisation to add new integration components to their 24/7 ops catalogue.

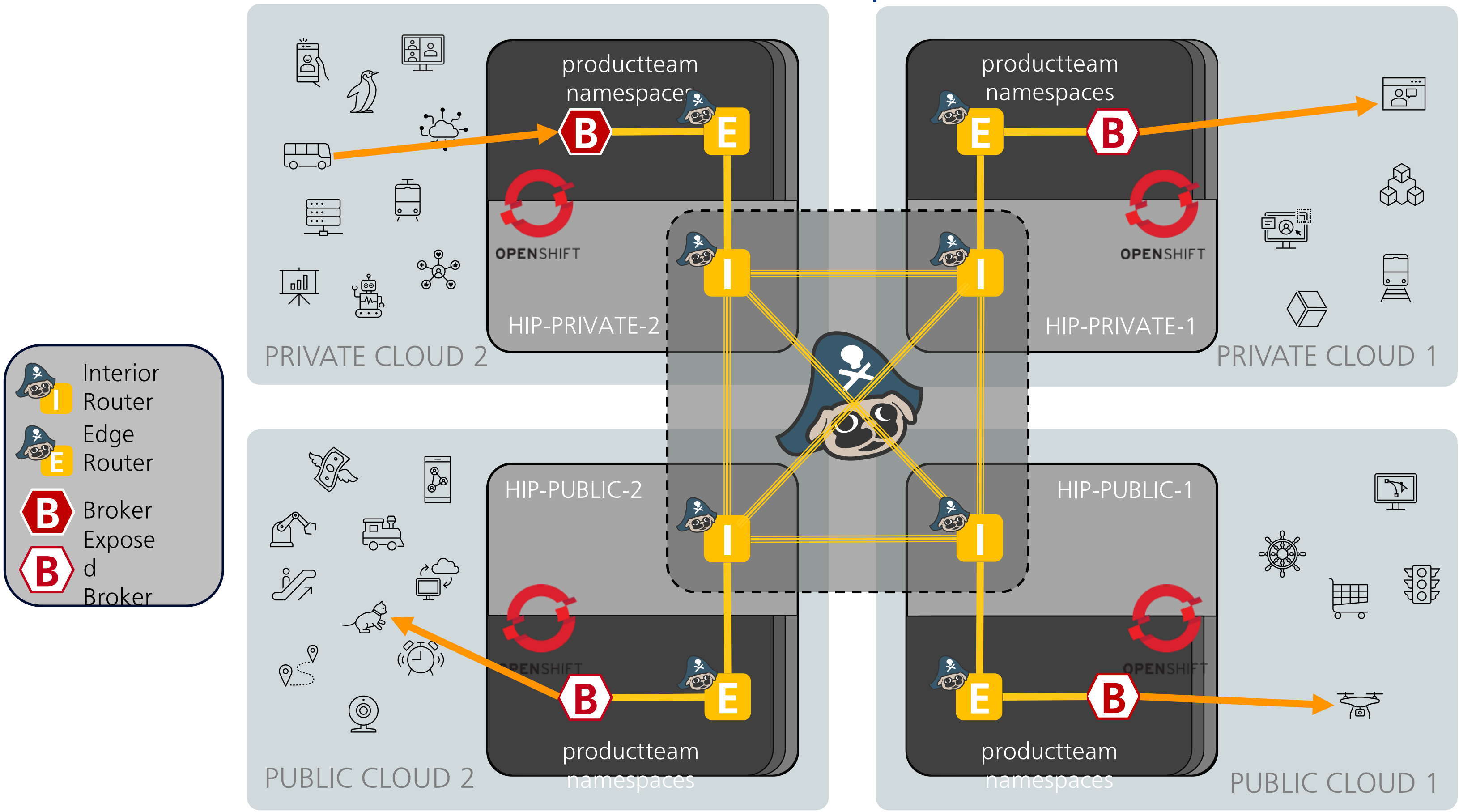
Which objectives have been accomplished?

- **Enabling of migration away from ESB**
- **Decentralized architecture**
- **Self-service platform**
- **Standardised building blocks**

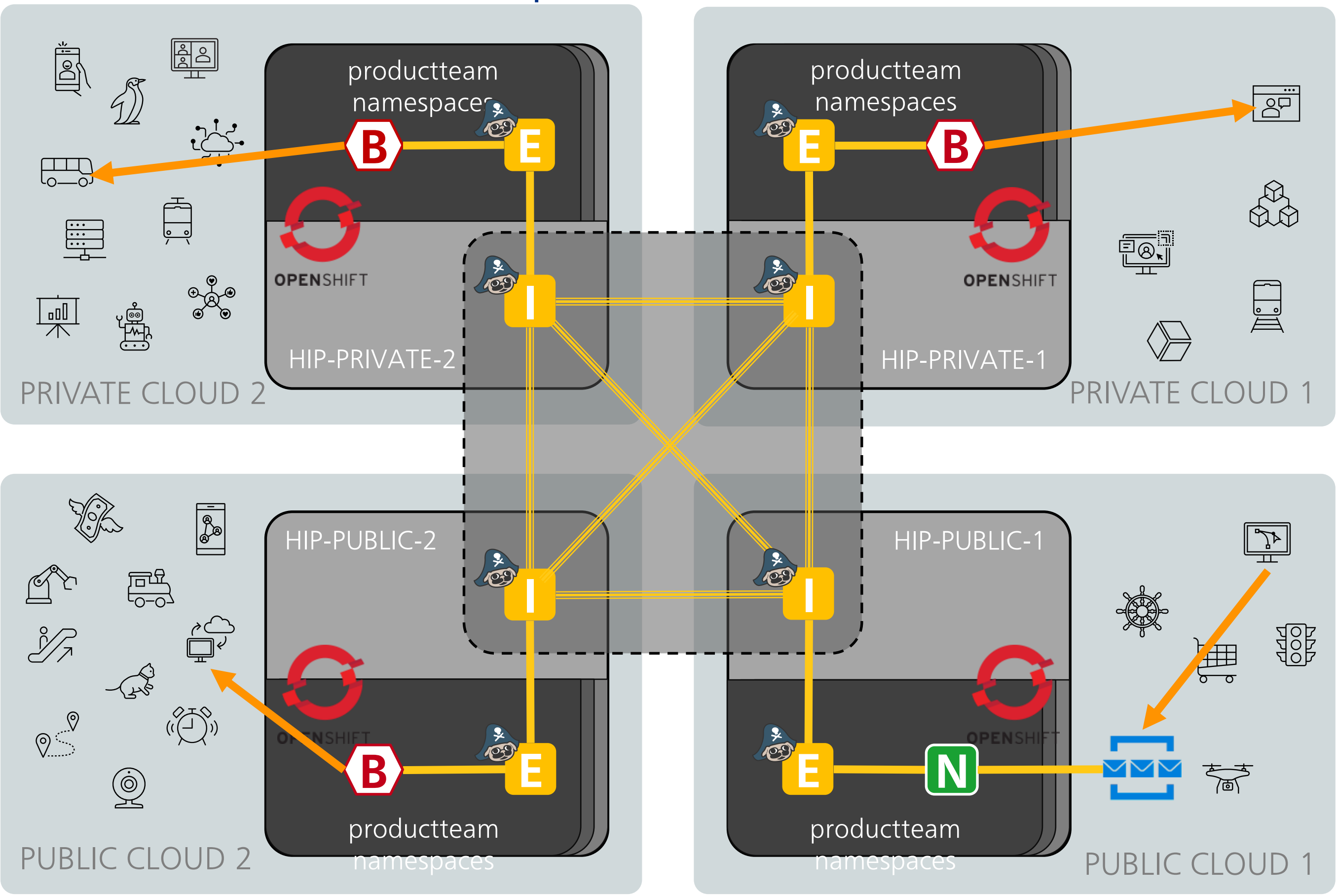
Red Hat Service Interconnect

- Simplifies application interconnectivity across the hybrid cloud by creating a L7 Virtual Application Network. This allows applications and services to communicate with each other as if they were running on the same site.
- L7 smart routing for traffic management, increased redundancy and automatic failover.
- Routes are secured via mutual TLS. VPNs and firewall rules are not needed.
- GA since July '23
- Aka Application Interconnect, based on https://skupper.io/ and Qpid Routers. Not a follow-up from AMQ Interconnect.
- Our challenges:
  1. Transparant interconnect
  2. Expose services / brokers from outside the OCPs.

HIP Phase 2: add transparent interconnect



HIP Phase 2: expose cloud-native services



I Interior Router

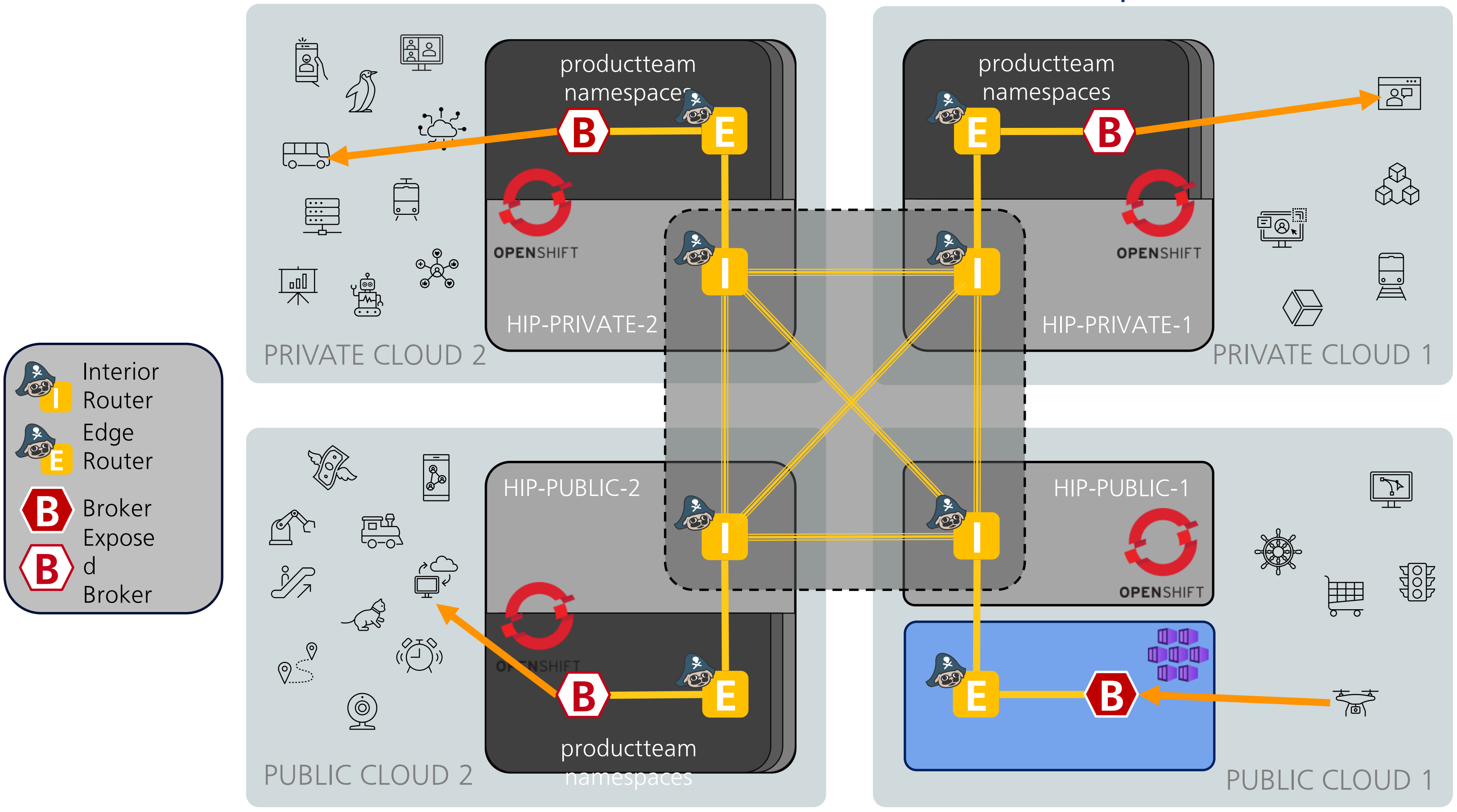
E Edge Router

Cloud-native Broker

B Exposed Broker

N Nginx

HIP Phase 2: interconnect other K8s platforms



Simplification drives platform adoption

Learning curve and tech stack

Simplification: although brokering remains necessary, it may now also be run in the application context. Same holds for adapter functions.

Architectural complexity

Simplification: stateless and transparent HIP mode is possible. Adapters no longer used as interconnect bridges.

Ops responsibility

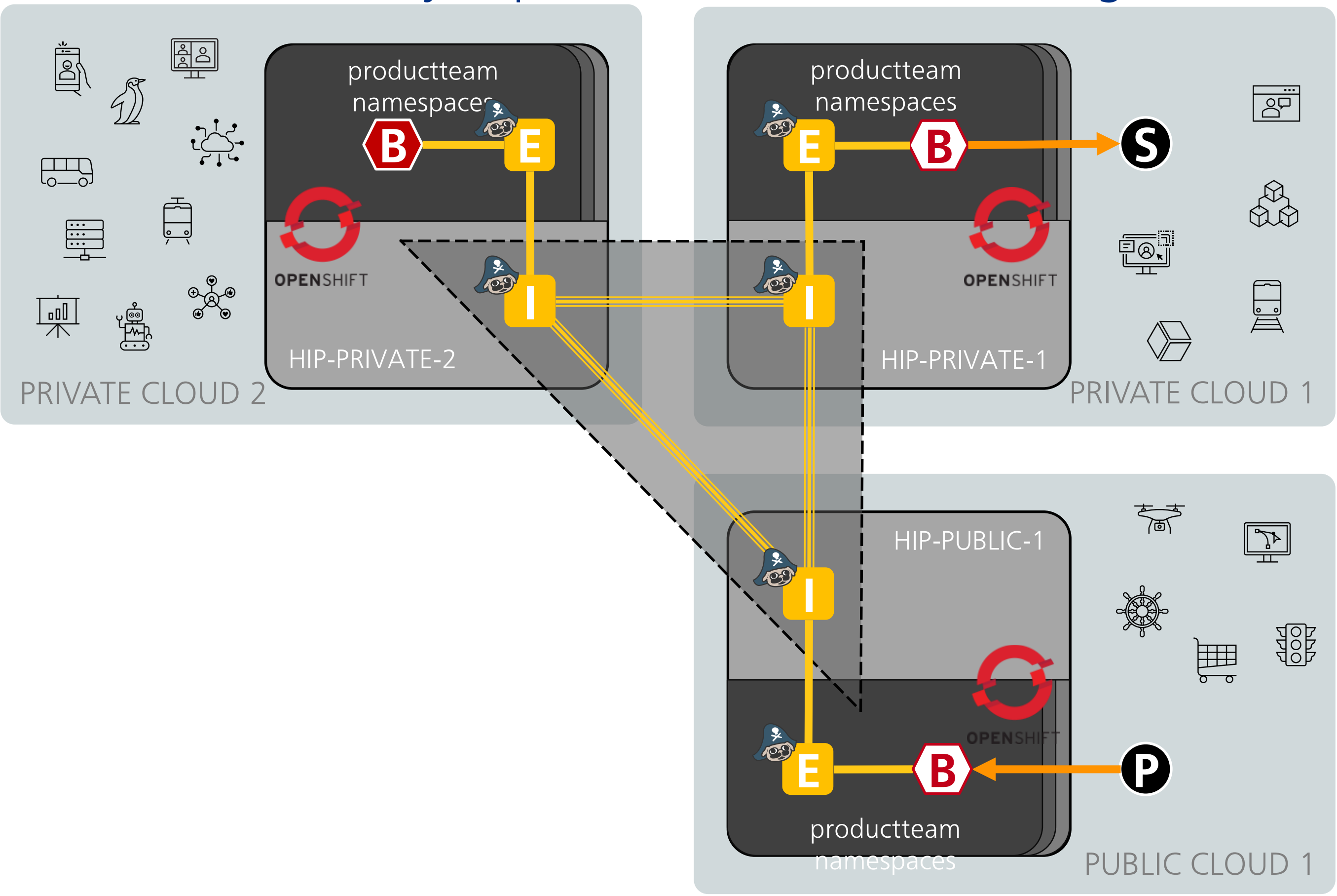
Simplification: Service Interconnect is based on a static configuration and its operational burden is therefore minimal.

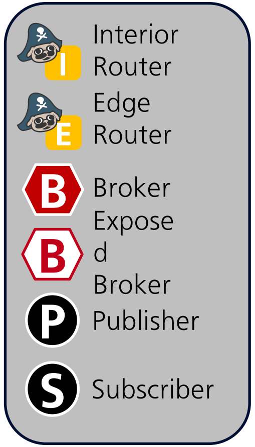
Which objectives were already accomplished?

- Enabling of migration away from ESB
- Decentralized architecture
- Self-service platform
- Standardised building blocks

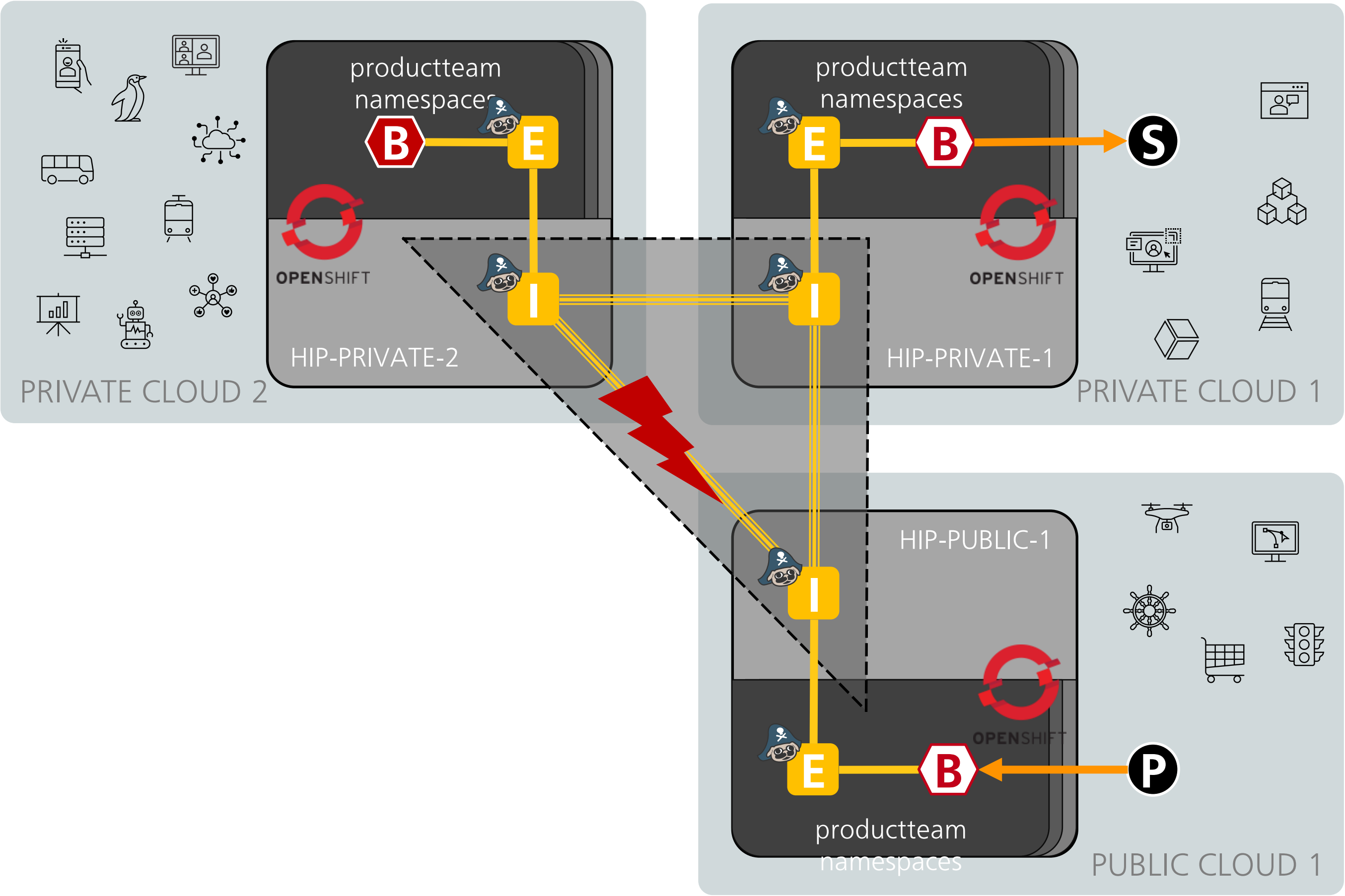
Demo

Demo 1: virtually expose broker across landingzones





Demo 2: interconnect failover



I Interior Router
E Edge Router
B Broker
B Exposed Broker
P Publisher
S Subscriber

Three takeaways

One size does not necessarily fit all

While in flux, simple solutions are the most attractive ones

Make sure that the best fitting solution is also the easiest solution (or vice versa)

Questions?

Jack Fleuren

Product Owner – CCI

✉ jack.fleuren@ns.nl
in /jackfleuren

Let's connect

Taco Nieuwenhuis

Solution Architect

✉ taco.nieuwenhuis@ns.nl
in /taconieuwenhuis

Let's connect

Floris Alfverink

Platform Engineer

✉ floris.alfverink@ns.nl
in /floris-alfverink-52767b1b

Let's connect