



Future of Continuous Delivery & Deployment @ Swisscom

MDS, MSRO



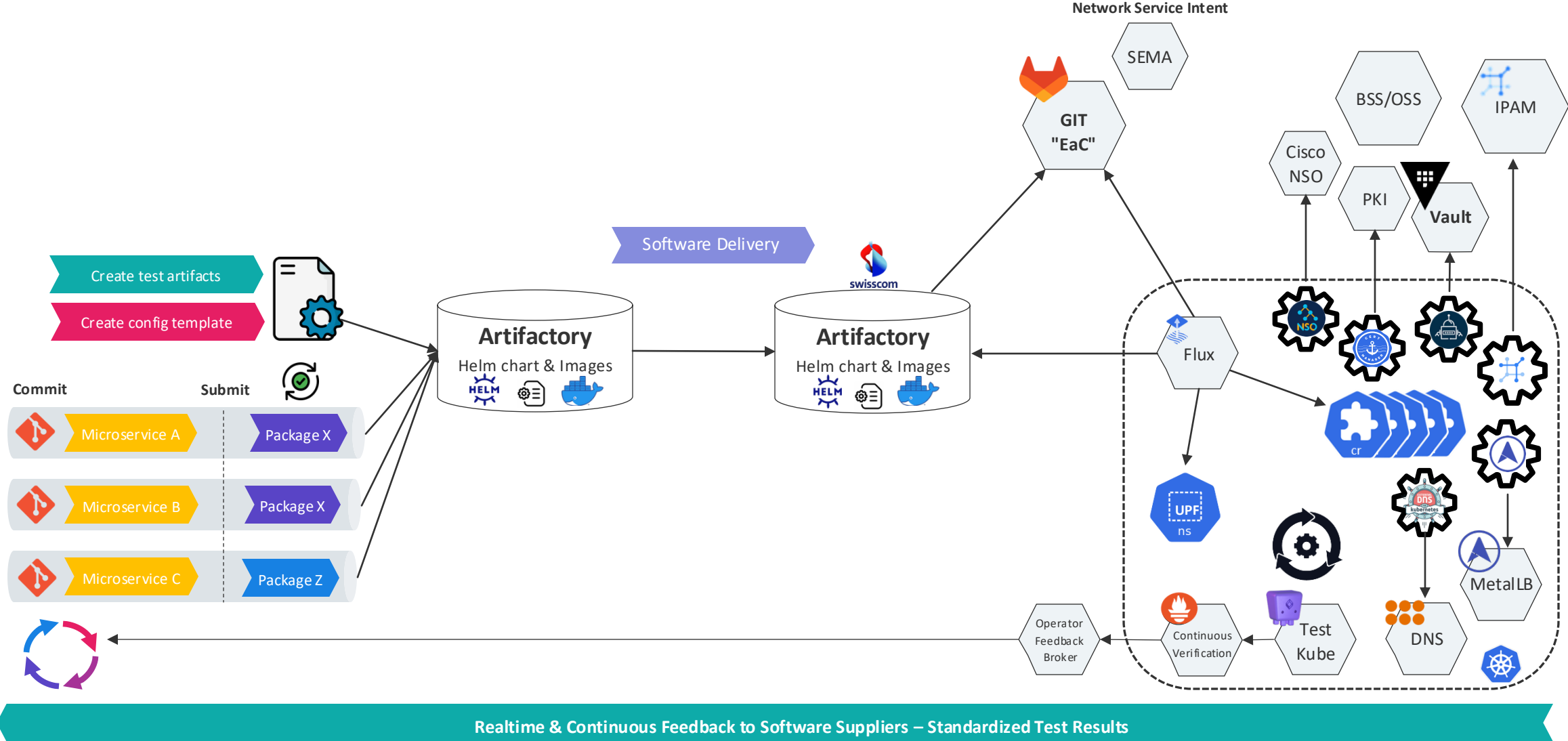
Problem Statement

The challenge is to develop an autonomous network service capable of interpreting high-level intent and ensuring these requirements are continuously and automatically met, providing reliable service to our customers. Achieving the necessary end-to-end automation requires a framework that guarantees rapid and reliable delivery of updates through continuous delivery pipelines.



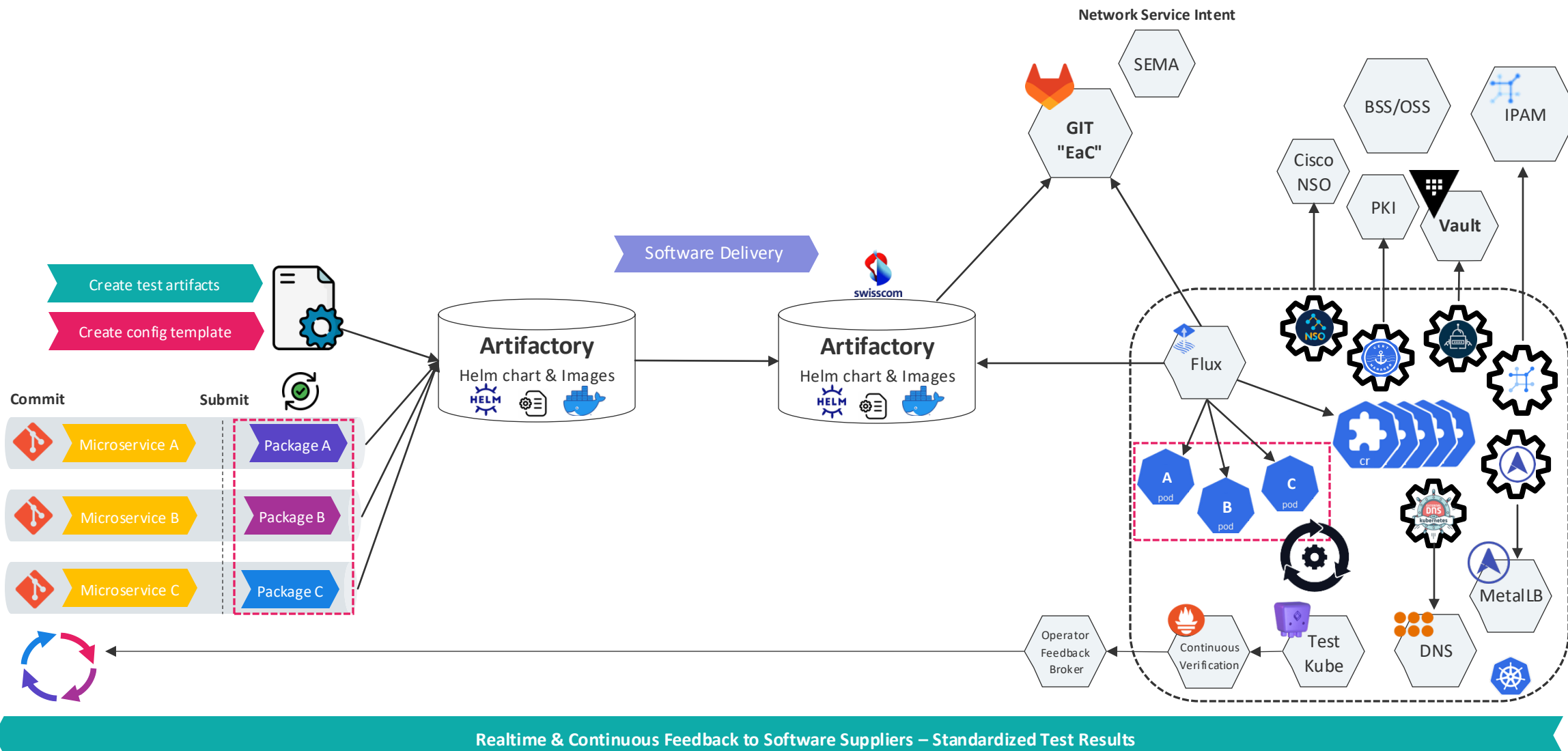
CNF Continuous Delivery Framework

Continuous Deployment and Automation at Swisscom, C2 General





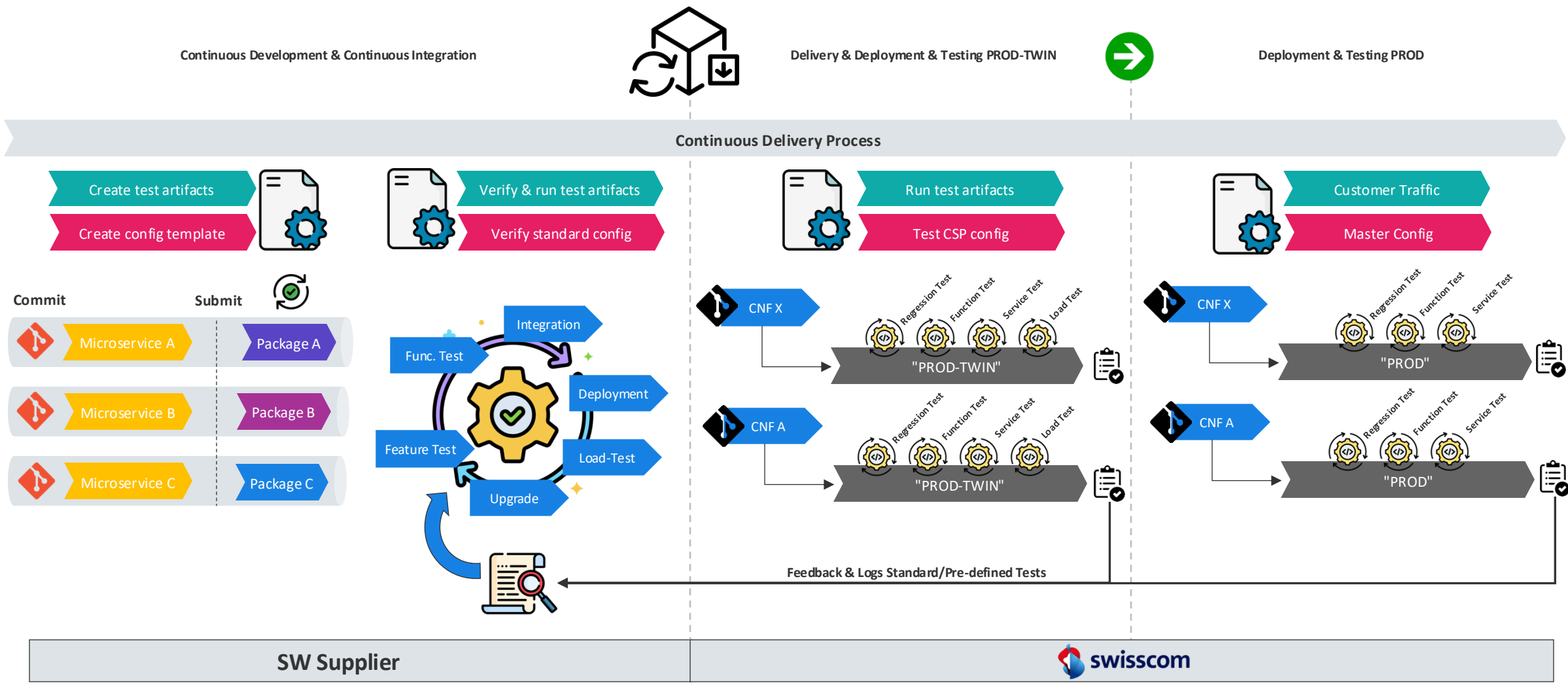
CNF Continuous Delivery Framework further Evolution





CNF Continuous Testing

Continuous Deployment and Automation at Swisscom, C2 General



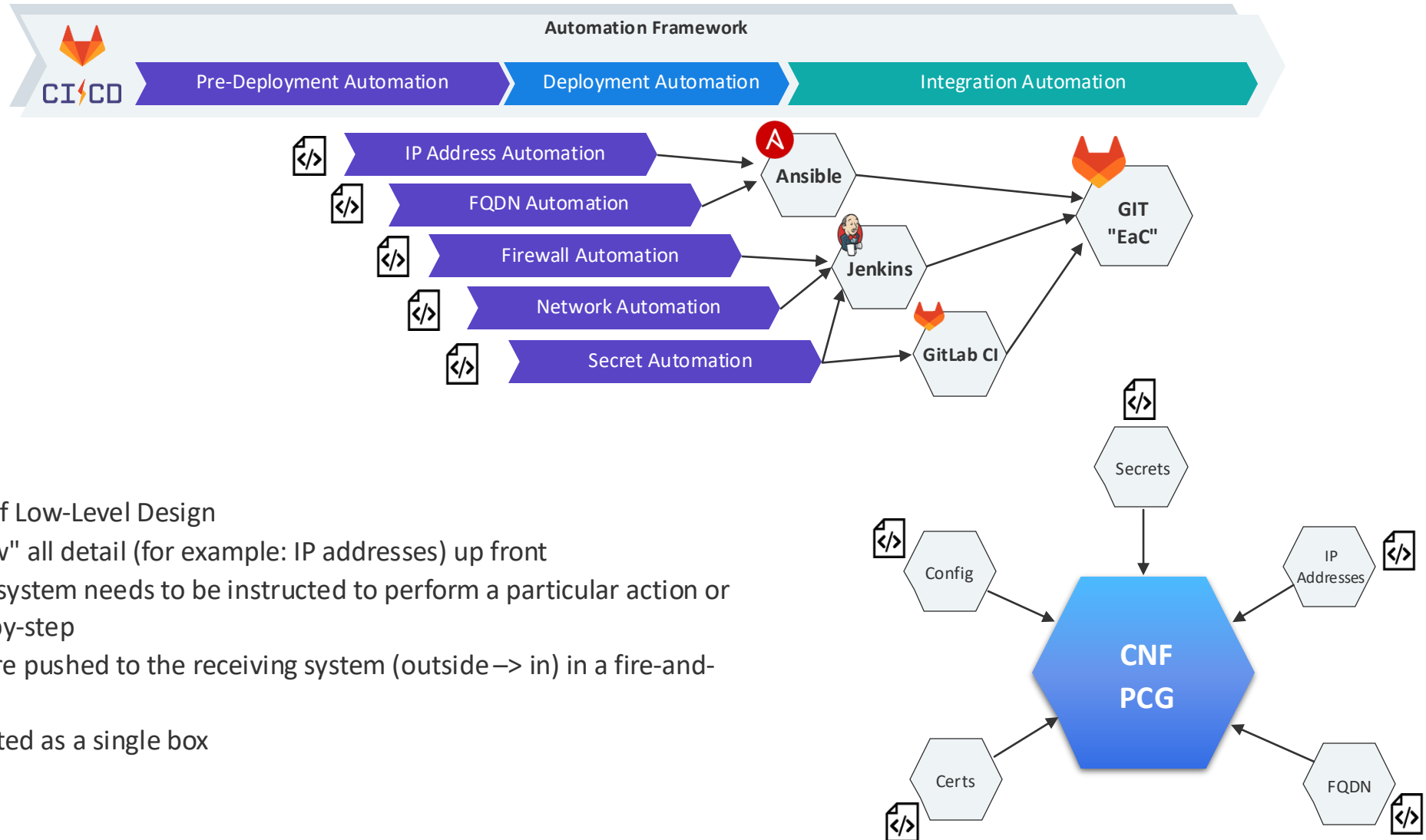


Design and Continuous Deployment

Swisscom aims to enhance the processes for designing, deploying, and operating network functions, including all Mobile Core applications that provide various services. Swisscom seeks to achieve a seamless and continuous process for managing applications, along with the necessary resources (PaaS).



Imperative pre-created design artifacts (actual state)



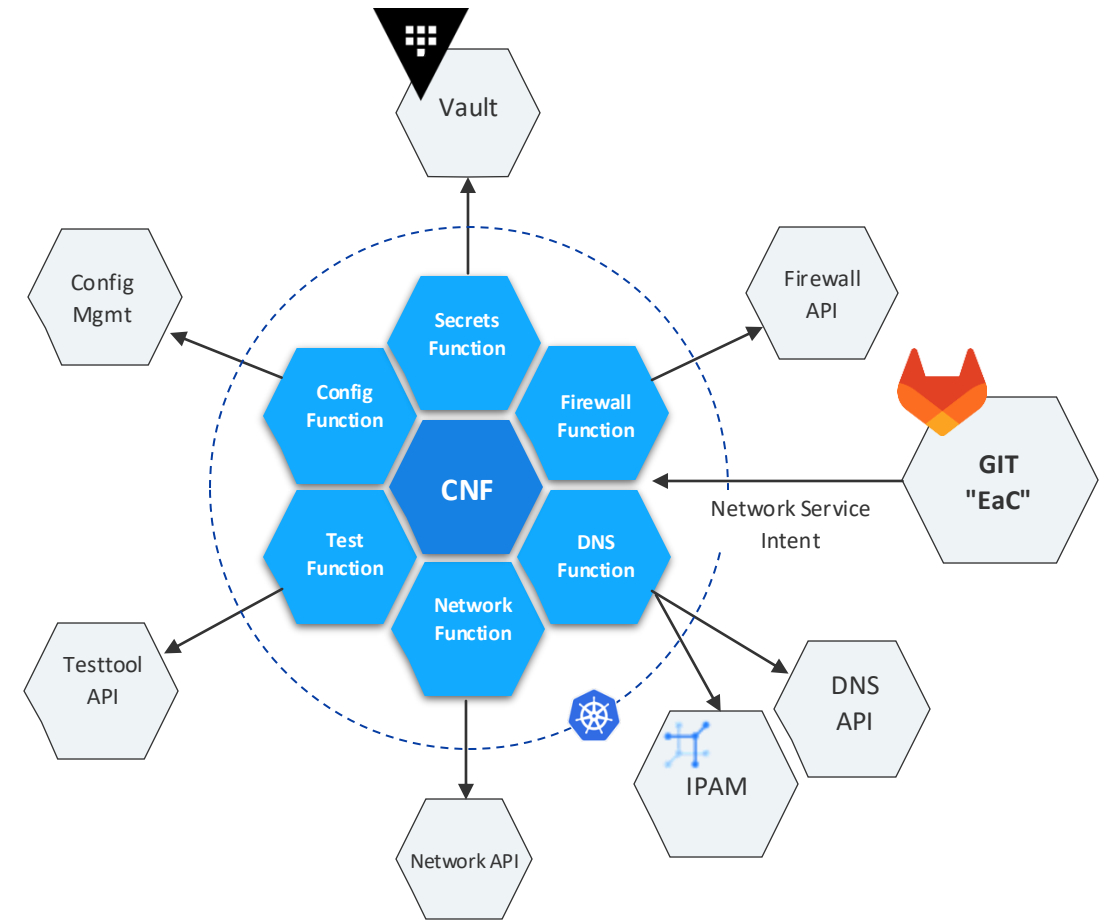
- Pre-creation of Low-Level Design
- "Need to know" all detail (for example: IP addresses) up front
- The receiving system needs to be instructed to perform a particular action or process step-by-step
- Instructions are pushed to the receiving system (outside → in) in a fire-and-forget mode
- System is treated as a single box



What If?

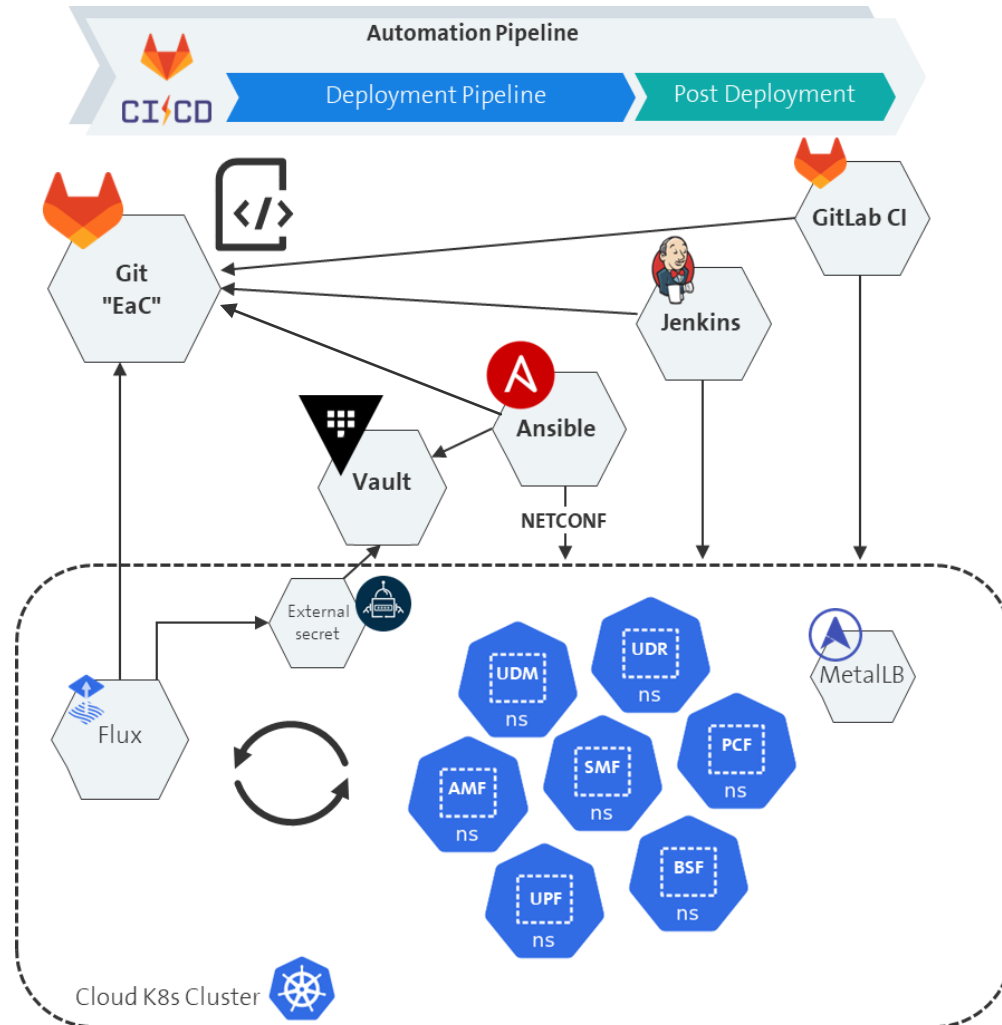
*Instead of telling the system **how to do everything**, we tell system **what we want**, and **system figures out the best possible way** to make it happen.*

- CNF receives the **intent** to move into an operational state **without step-by-step instructions**
- **Dynamic config assembly** based on **required resources** during runtime
- **Inside → out**
System requests relevant information automatically from external sources
- **Extend Kubernetes API** to bring Kubernetes-native capabilities





1. Current State (**not preferred**)



2. Future State

