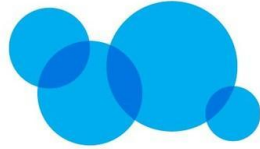




# LF NETWORKING

Developer & Testing Forum



NEPHIO

# SIG Automation

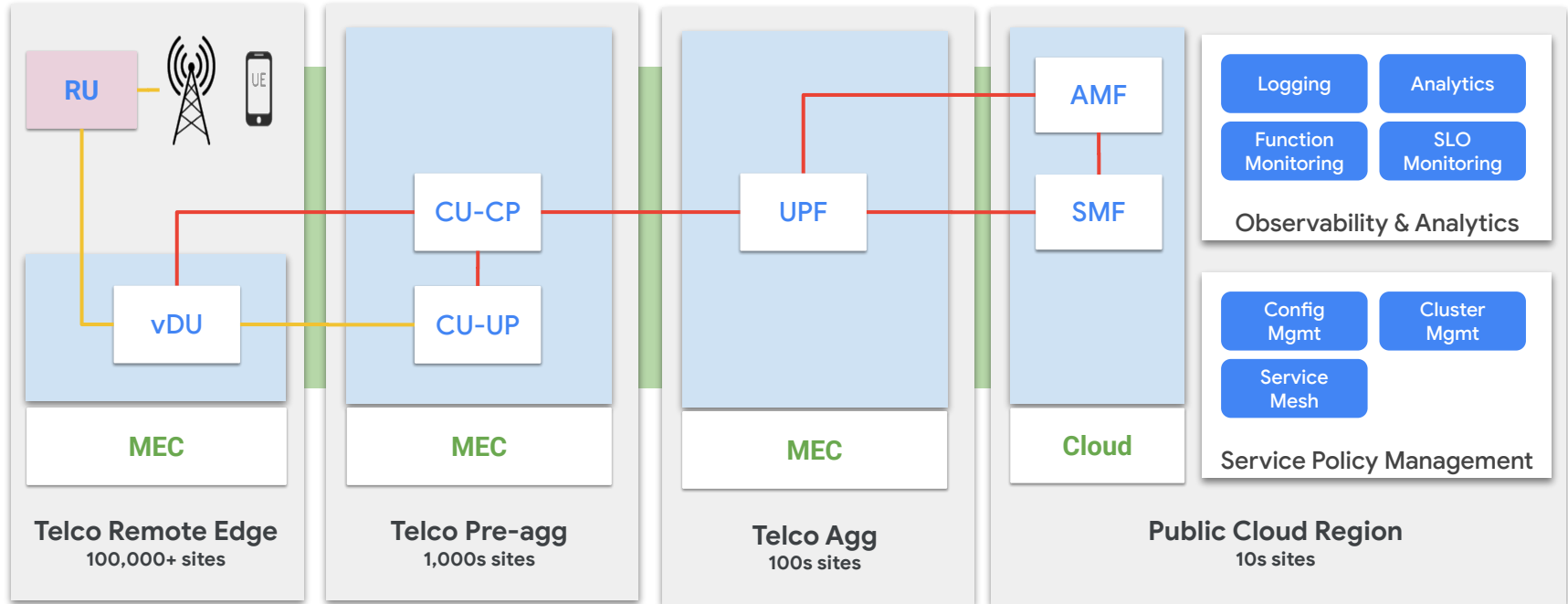
**John Belamaric** Sr Staff Software Engineer *Google*  
(*Chair SIG Automation*)

**Wim henderickx** Head of Technology & Architecture *Nokia*  
(*Vice-Chair SIG Automation*)

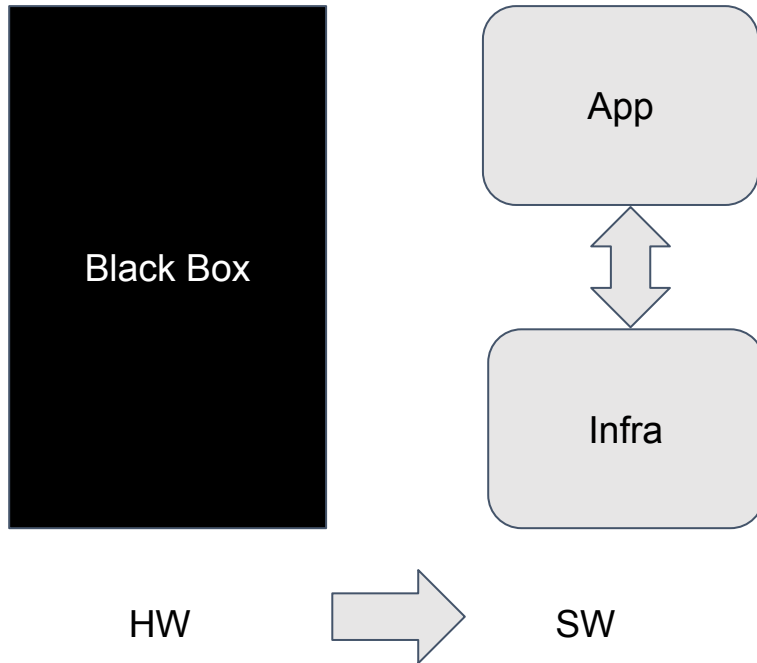
<https://lfnetworking.org>



# The Telco landscape



# Tightly coupled



- High Performance
- Mission critical
- Closed systems
- Different culture/mindset
- ...

# Specialization versus Generalization



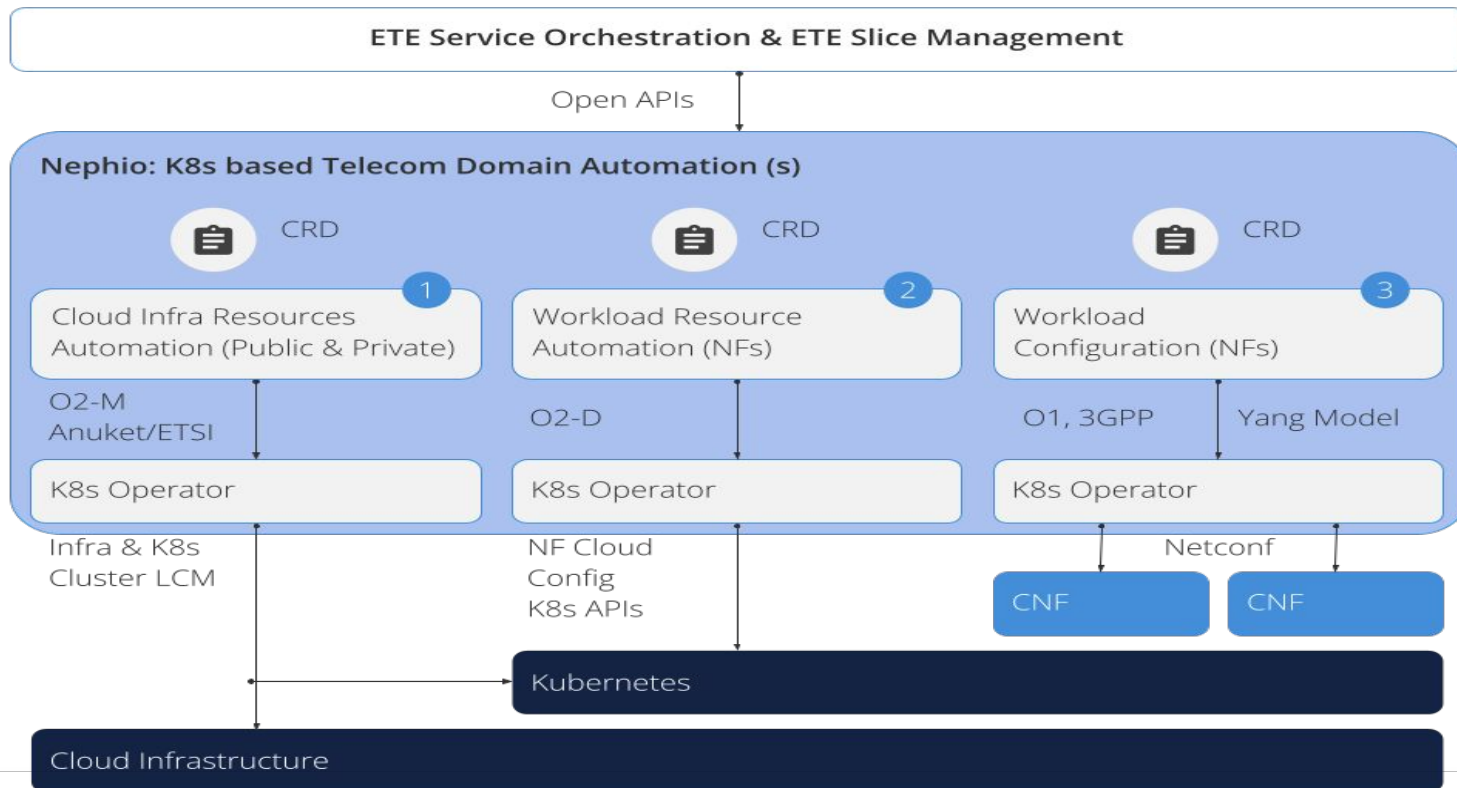
**Domain A**



**Domain B**



**Domain C**



# Goal

## Intent Driven

Continuously reconciling systems are more robust at scale than imperative, fire-and-forget tools.

## Distributed Actuation

A must for large-scale edge deployments. Triggering all actions from a centralized location is not reliable and does not scale, especially for edge deployments.

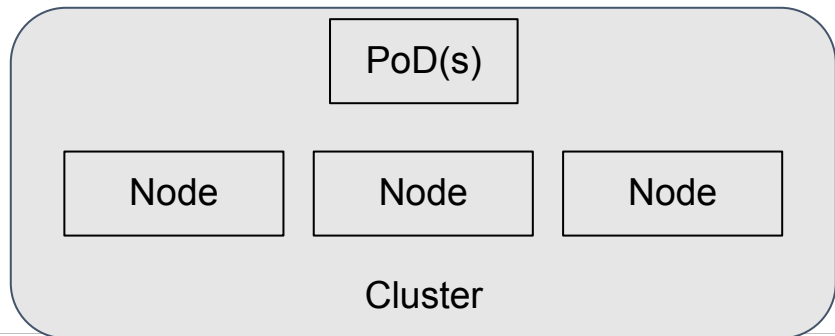
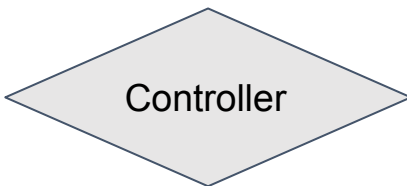
## Uniformity in systems

Reduce human errors, easier to maintain and allow for pre-validation at scale

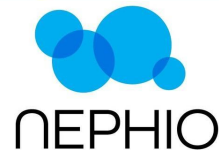
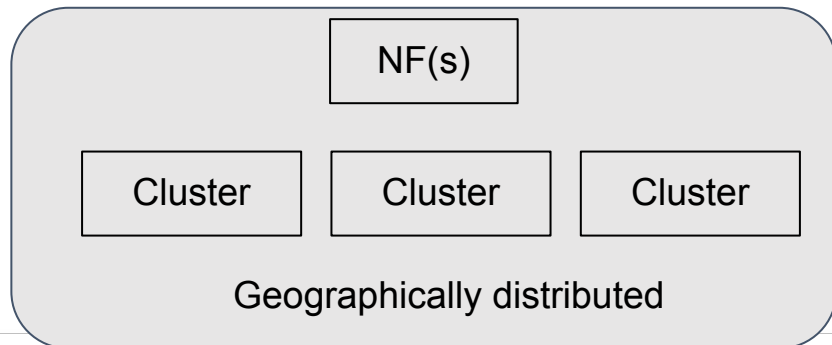
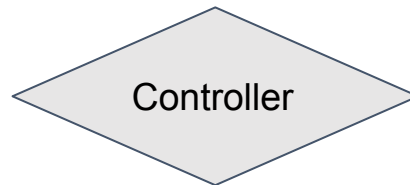
# Analogy



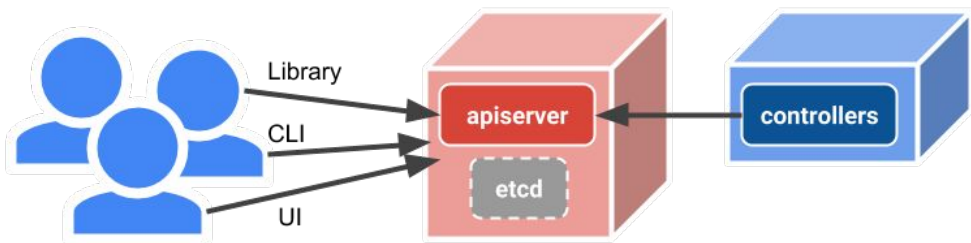
Deployment, Daemonset, StatefulSet



Package Deployment



- API centric
- Declarative control
  - desired state
  - observed state
- Standard metadata (indexing)
- Extendable (CRD)
- Event driven (Watch)
- Resource Semantics and lifecycle
- Eventual consistency
- Huge eco-system



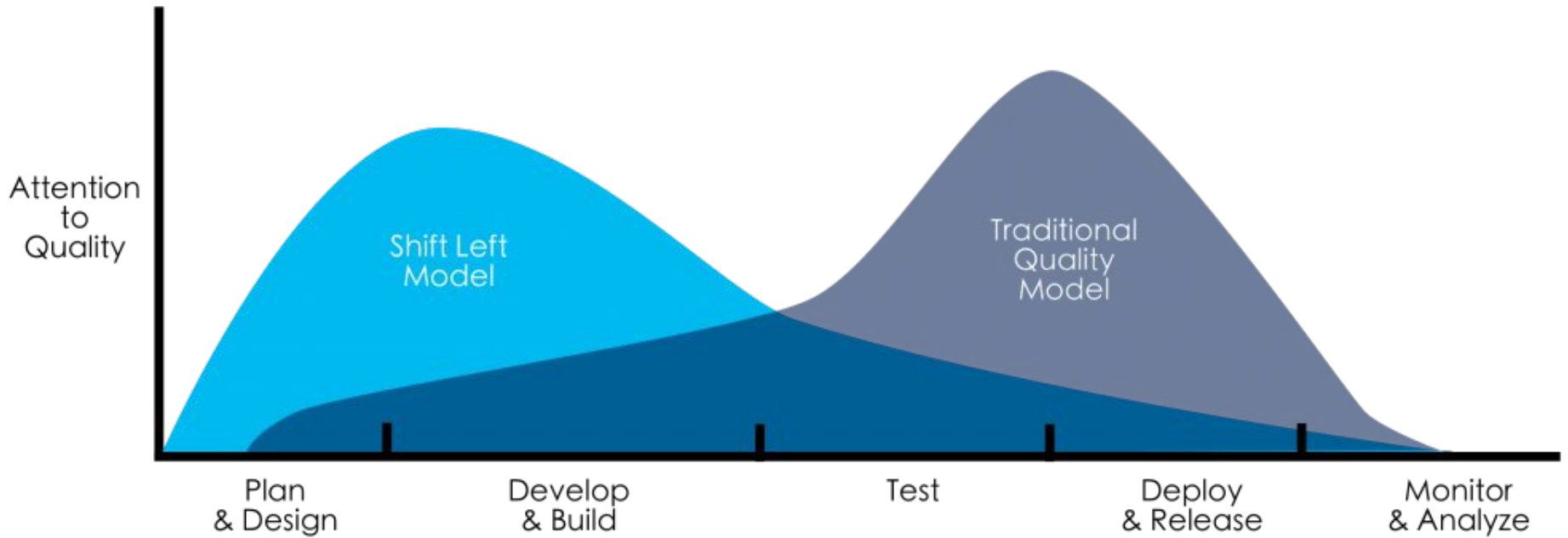
<https://github.com/kubernetes/design-proposals-archive/blob/main/architecture/resource-management.md>



# Configuration-as-Data (CAD)

- A new approach to configuration management
  - Represent config in a well-defined, structured data model (KRM!)
  - Configuration lives in versioned storage, separate from the live state
  - Tools operate on the config - do not intermingle code and configuration
  - Clients interact with config via APIs, not directly on storage
- Machine manageable configurations
- Enables iterative, multi-actor workflows to operate and validate configurations
- Automated changes, bulk operations, and human-initiated modifications co-exist peacefully
- Automatic system validation of configuration before applying to live state
- Reusable, well-tested functions operate on configuration rather than embedding code inside the configuration
- Implemented in open source projects kpt, Porch, and Config Sync

# Shift Left

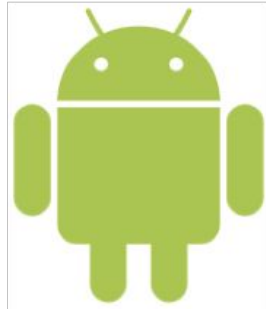


# The app model (KRM)

Specialization



Generalization



# High Level Nephio architecture

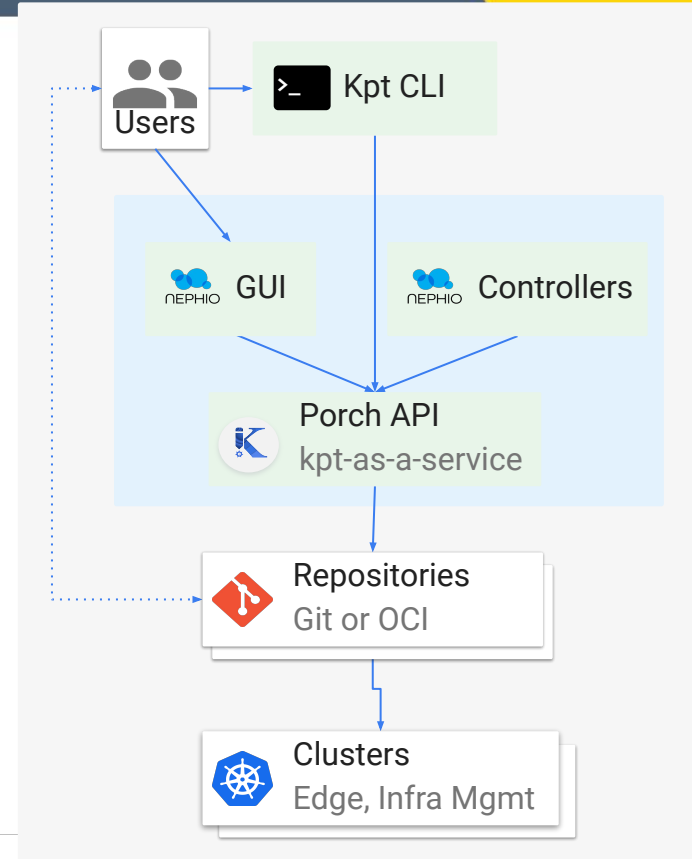
Platform enabling users and automation agents to cooperatively interact with and deploy configuration

Central Nephio K8s cluster houses GUI service, Porch APIs, and Nephio Controllers

Users manipulate config packages which Porch pushes to Git or OCI repositories

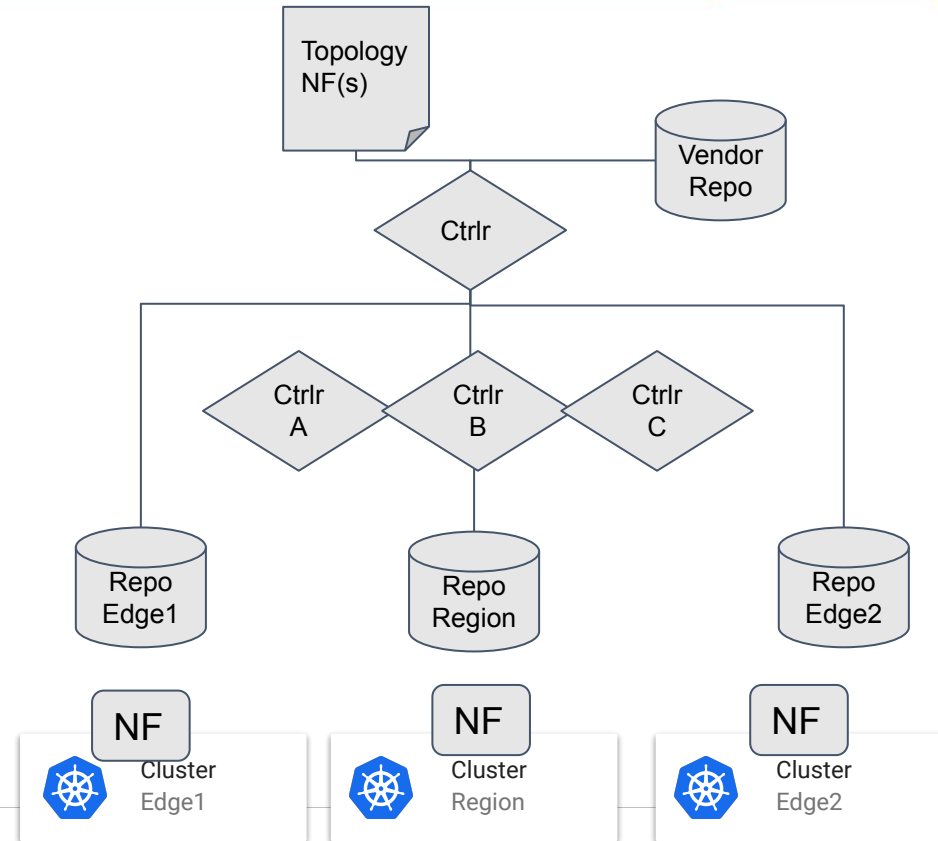
Downstream clusters consume those via Config Sync, which applies them to the K8s API server

Same process for infrastructure - using for example KCC to create clusters, as for workloads



# Decoupling/hydration

- Business intent (abstract definition of what we want to achieve)
- E.g i wants a set of NF(s) based on some criteria's
- Result:
  - A set of NF(s) get deployed on a set of selected clusters
- How?
  - Controllers, Functions
  - Hydration
  - Loose coupling
  - Reusable components
  - Late binding
  - Event driven



# Demo time

# Demo

- Installation the POC Nov 22
  - Details how to install the Nephio POC
  - <https://github.com/nephio-project/one-summit-22-workshop/tree/main/nephio-ansible-install>
  -
- Workshop POC Nov 22
  - Details how to go through the demo
  - <https://github.com/nephio-project/one-summit-22-workshop>

## Project Resources

- Website - <https://nephio.org/> , <https://nephio.org/about/>
- Wiki - <https://wiki.nephio.org/>
- Blog Postings - <https://nephio.org/blog/>
- Project Github - <https://github.com/nephio-project> (Please note “nephio-project is right one”)
- More links at <https://github.com/nephio-project/docs>
- Project email distro - <https://lists.nephio.org>
  - [nephio-tsc](#) (for TSC members and interested parties)
  - [nephio-dev](#) (for all)
- SIG lists: [sig-netarch](#), [sig-automation](#), [sig-release](#)