Deliver Applications Efficiently with Cloud-Native Platforms

CNCF TAG App Delivery

Jennifer Strejevitch (Chair)

Thomas Schuetz (Tech Lead)

Josh Gavant (Tech Lead)

Agenda

- **TAG** App Delivery: Mission and opportunities
- **Platforms** enable efficient app delivery
- **Operators** enable apps and platforms
- What is a cloud-native **application**?

CNCF TAG App Delivery

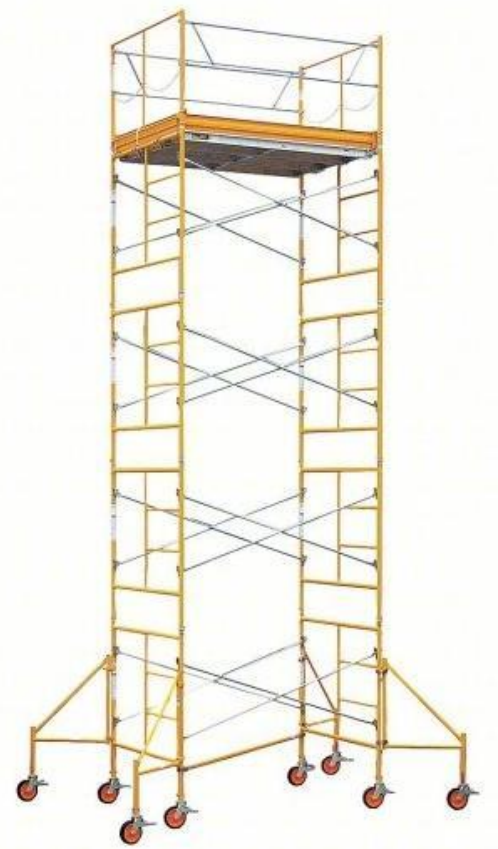
- Objectives: Enable CNCF projects and guide CNCF end users
- How to contribute?
  - Share your cloud-native application experiences
  - Share your open source or CNCF project
  - Build inter-project and inter-user connections
  - Develop prototypes and patterns
- Active work
  - Working Groups: Platforms, Operators, Artifacts, GitOps
  - Your passion!

Platforms

Platforms enable efficient app delivery

What are platforms and how do they facilitate app delivery?

- **Demo**: deliver app on a platform
- **Capabilities**: functional components
- **Attributes**: non-functional characteristics

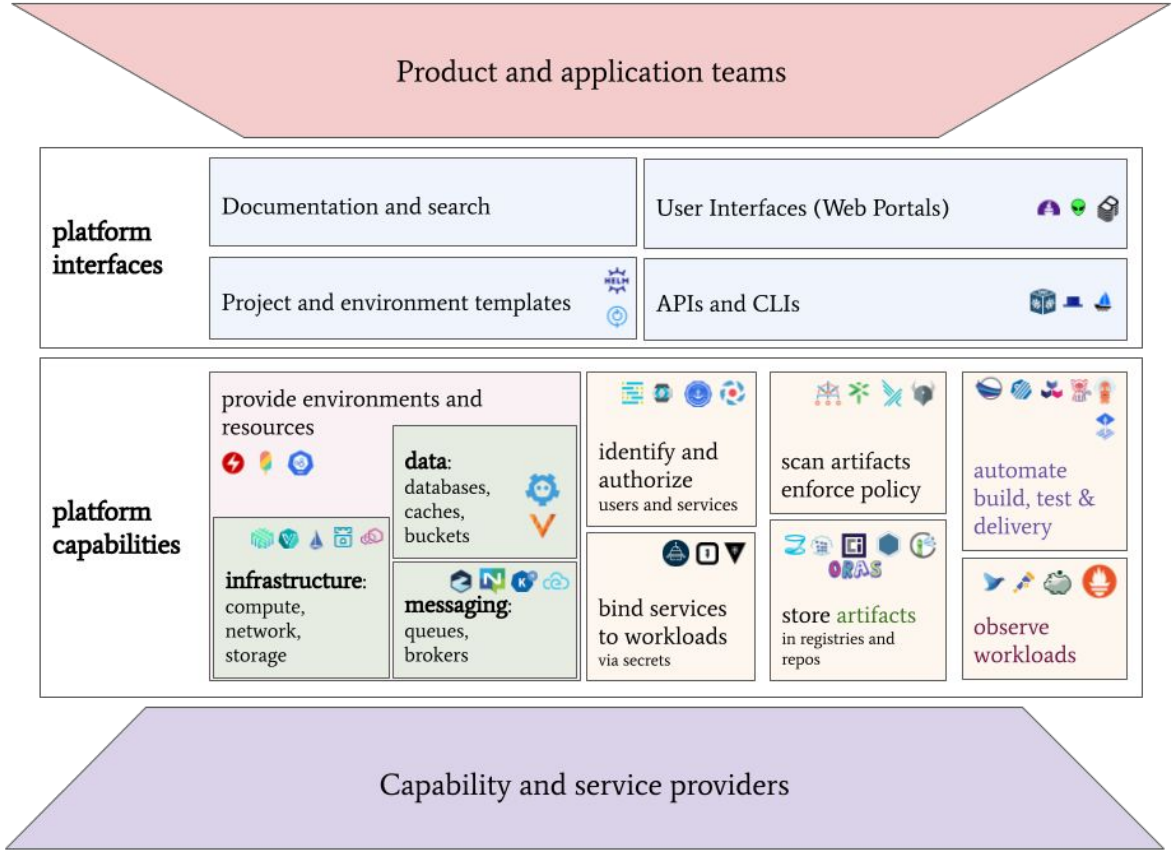


What and why?

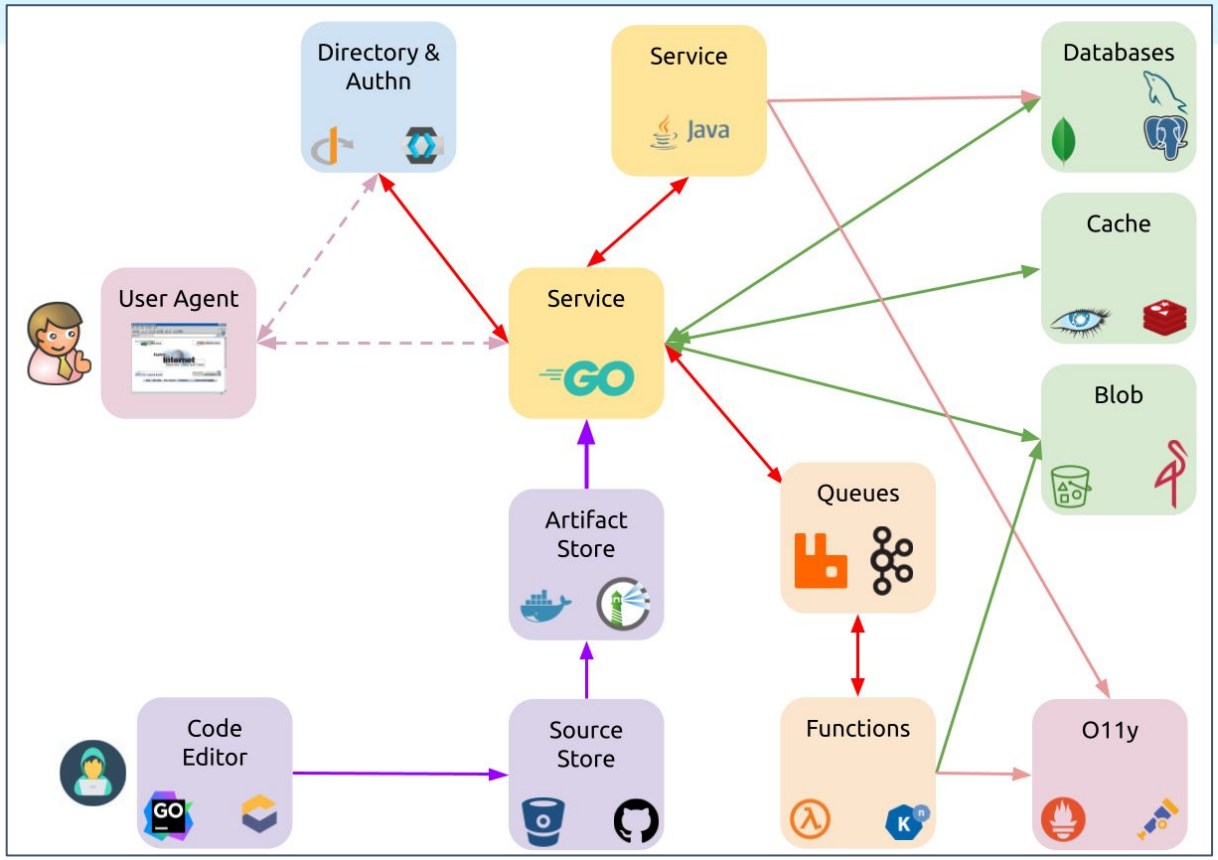
From https://tag-app-delivery.cncf.io/whitepapers/platforms/

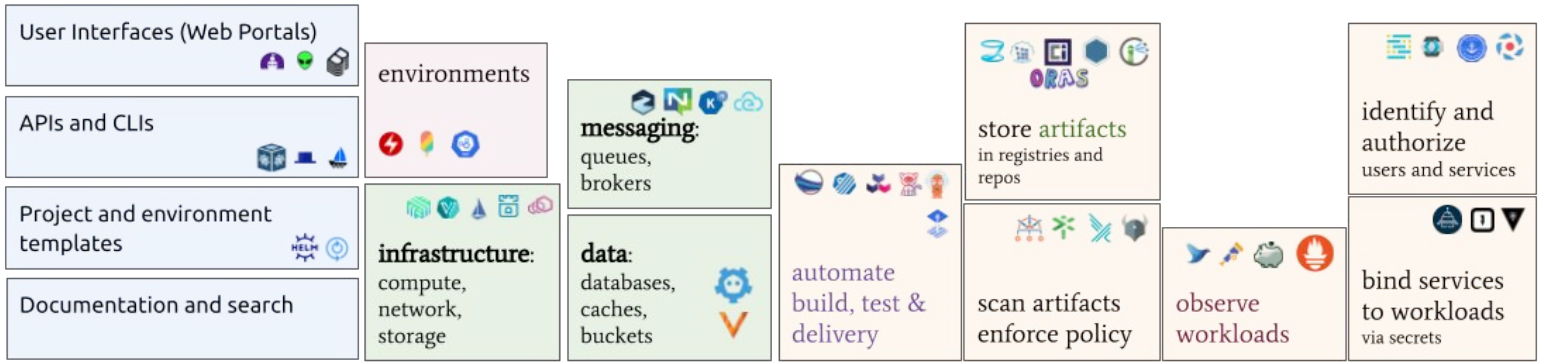
"A platform is ... an integrated collection of capabilities defined and presented according to the needs of the platform's users"

"In cloud-like environments resources and capabilities are often managed independently and integrated with custom business components."



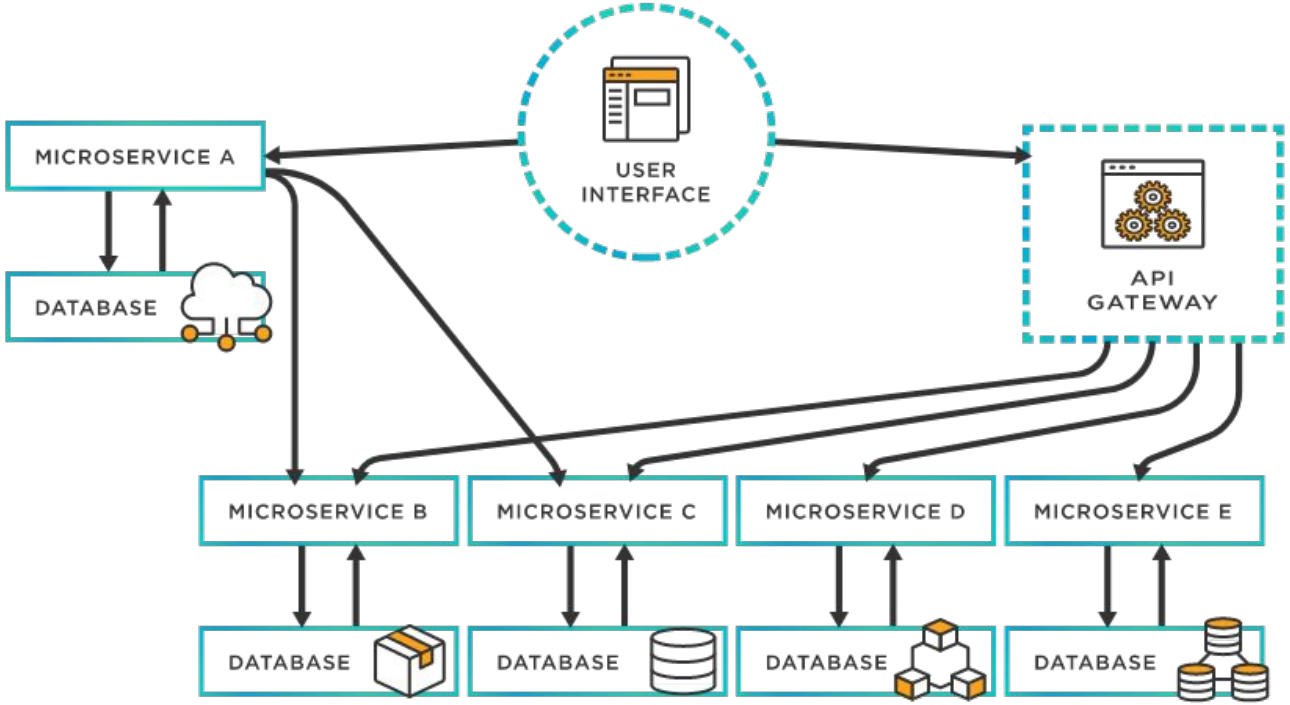
Cloud-Native App & Platform





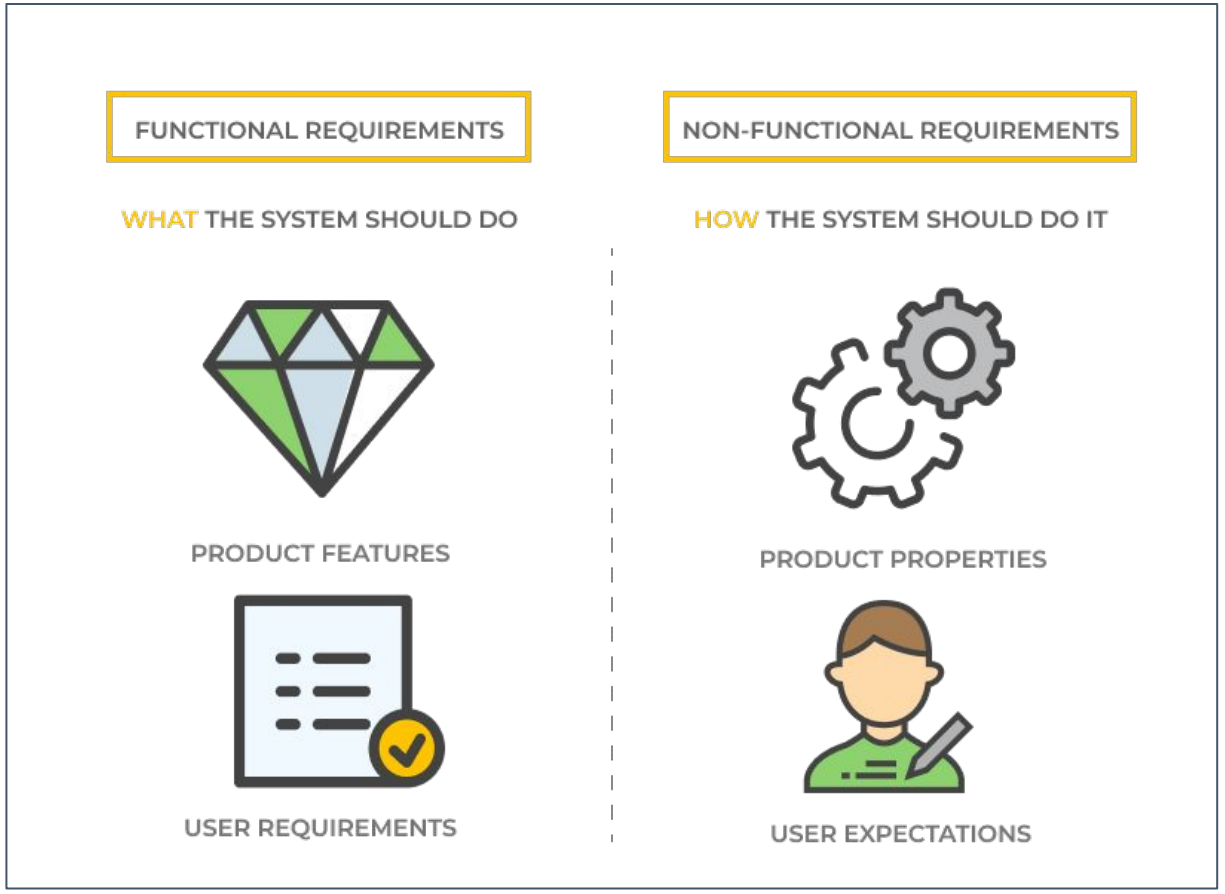
Demo

https://github.com/joshgav/devenv/tree/main/apps/apiserver/base-openshift



Platform Attributes

- Product Mindset
- Developer Experience
- Documentation
- Self-service
- Reduce cognitive load
- Optional and composable
- Secure by default



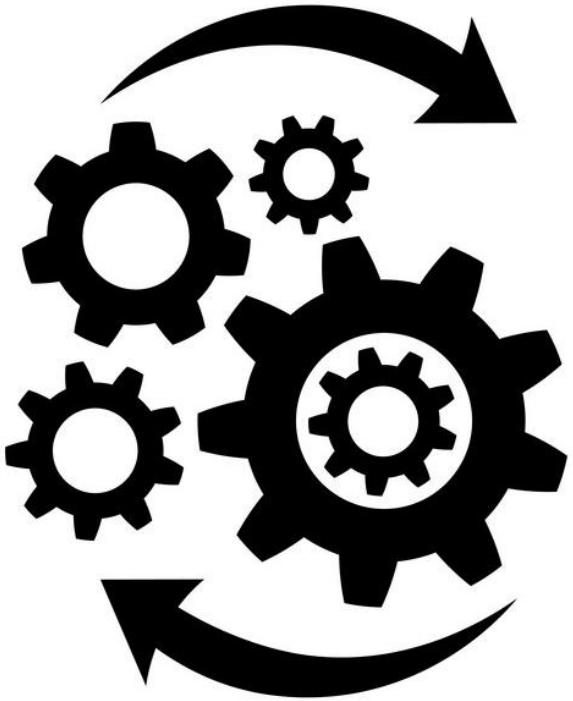
from https://upplabs.com/blog/the-importance-of-functional-and-non-functional-requirements-in-software-development/

Platform Capabilities

- Portals and APIs
- Templates and Docs
- Automation for Build, Test, and Delivery
- Development Environments
- Observability
- Infrastructure: Compute, Network, Storage
- Data and Messaging
- Identity and Secrets
- Security and Policy
- Artifact Storage

https://landscape.cncf.io/

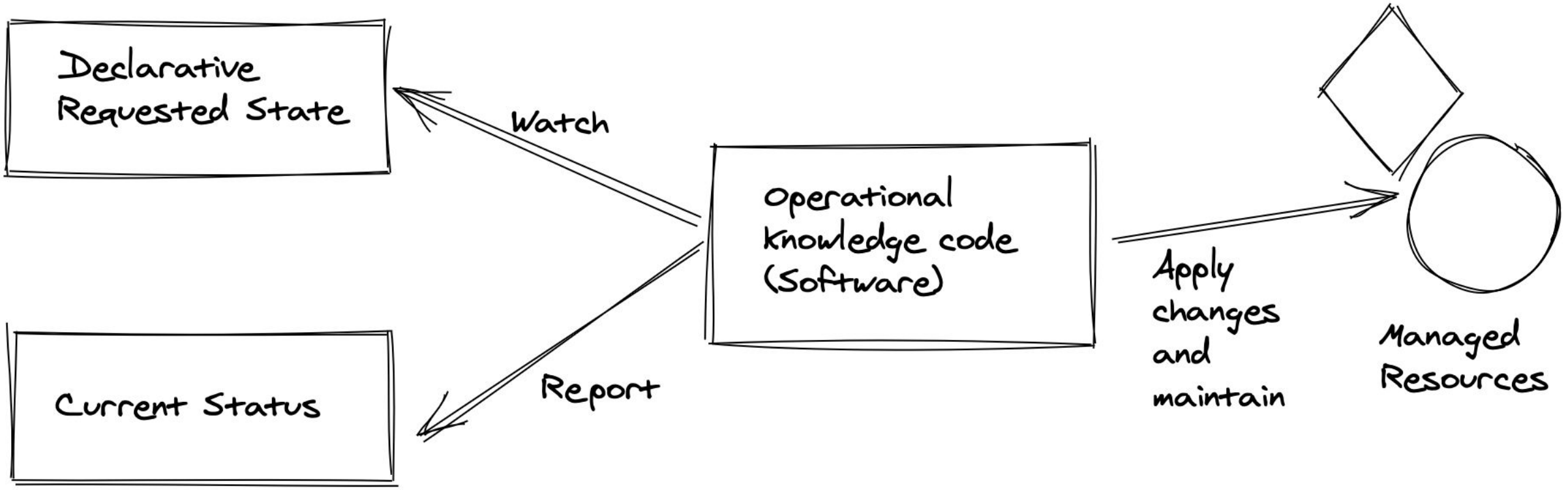
Operators



Operators enable applications and platforms

- Operators aid application lifecycle management beyond Kubernetes primitives.
- Operators abstract a lot of the platform toil and provide foundations for platform tooling interfaces and self-service.
- Operator patterns have matured beyond Kubernetes itself.

Basic Operator reconciliation



Operators make platforms more product-like

- Manage your platform as a product
- Kubernetes Operators make infrastructure management more application-like by allowing the management of complex infrastructure resources to be automated and abstracted away from low-level details.
- That allows developers to request dependencies without worrying about the underlying infrastructure provider for example.

Example: Crossplane

- **CRD**: (Human) User interface (which can be abstracted to GUI/CLIs, etc.)
- **Controllers**: Watch and perform programed tasks on change of resources to achieve reconciliation.
- **Platform Agnostic Infrastructure**: Operators can be developed to talk to different cloud providers or on premises APIs

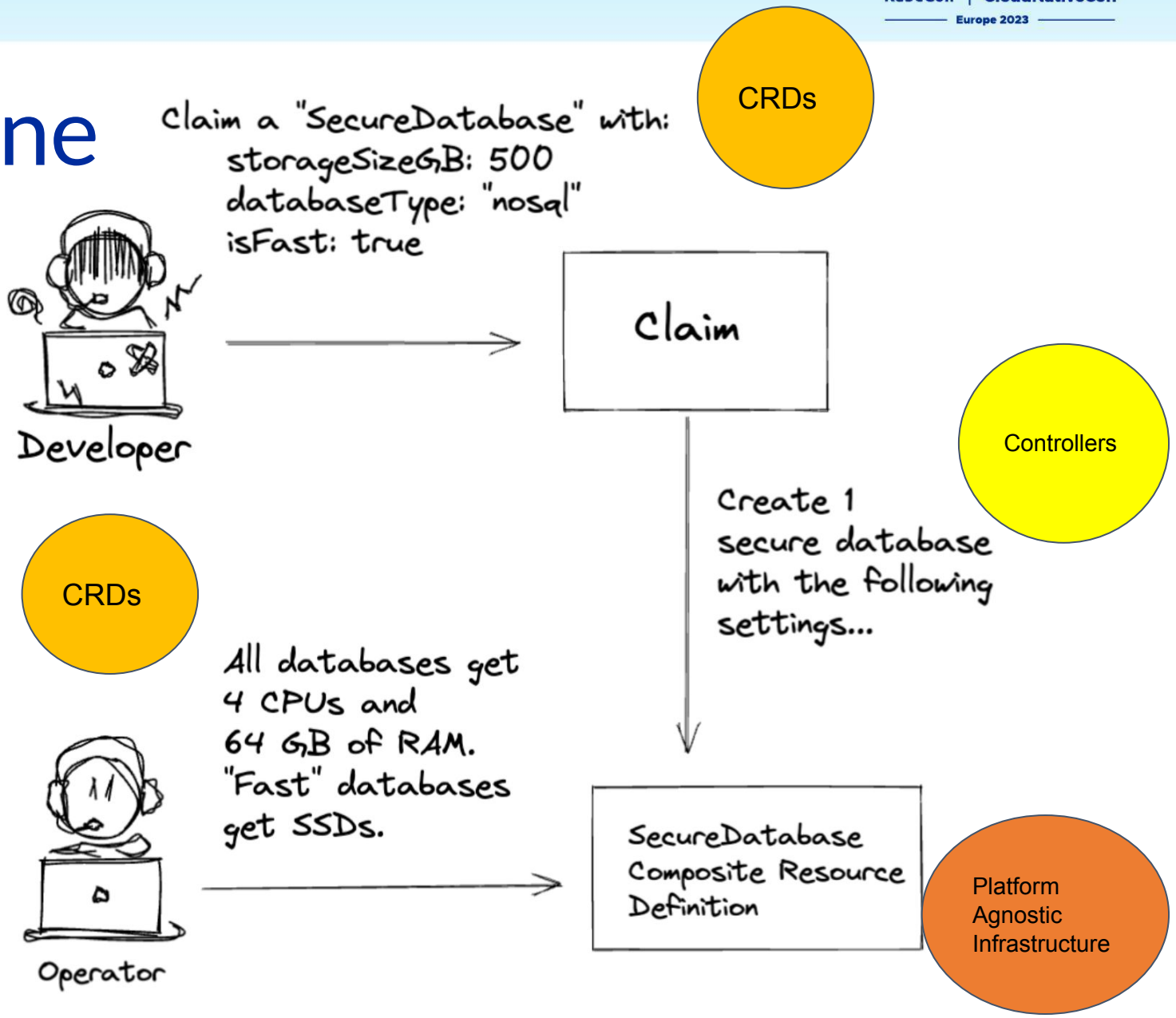
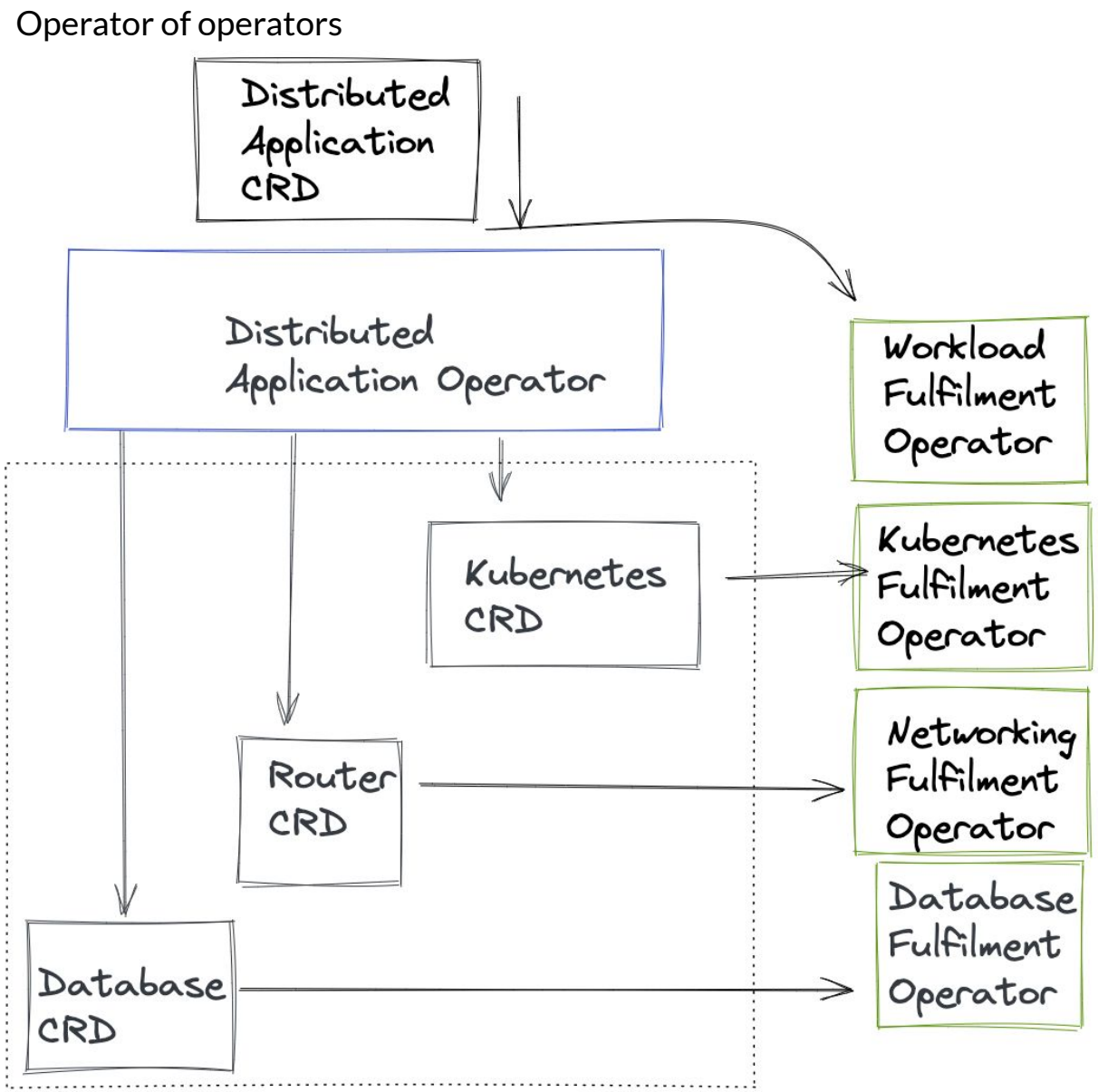


Image source: https://blog.upbound.io/crossplane-operators-new-bff/

Operator Patterns

- Operator of Operators
- OLM - Operator Lifecycle Manager
- Operator SDK



Popular Operator Frameworks

- kubebuilder
- CNCF Operator Framework

Emerging Operator Frameworks

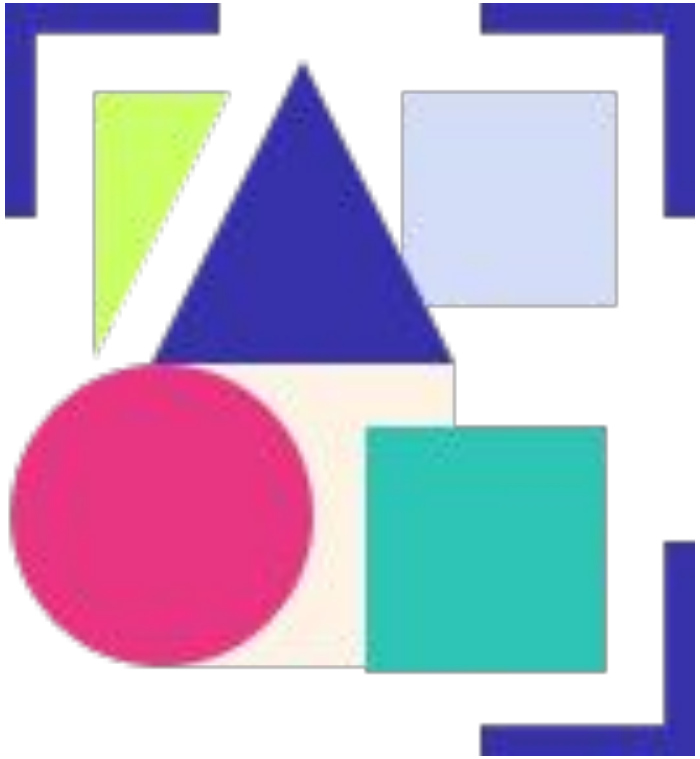
- Metacontroller - Lightweight Kubernetes controller as a service
- Juju - Model Driven Operator Framework

Operator Whitepaper and Operators evolution

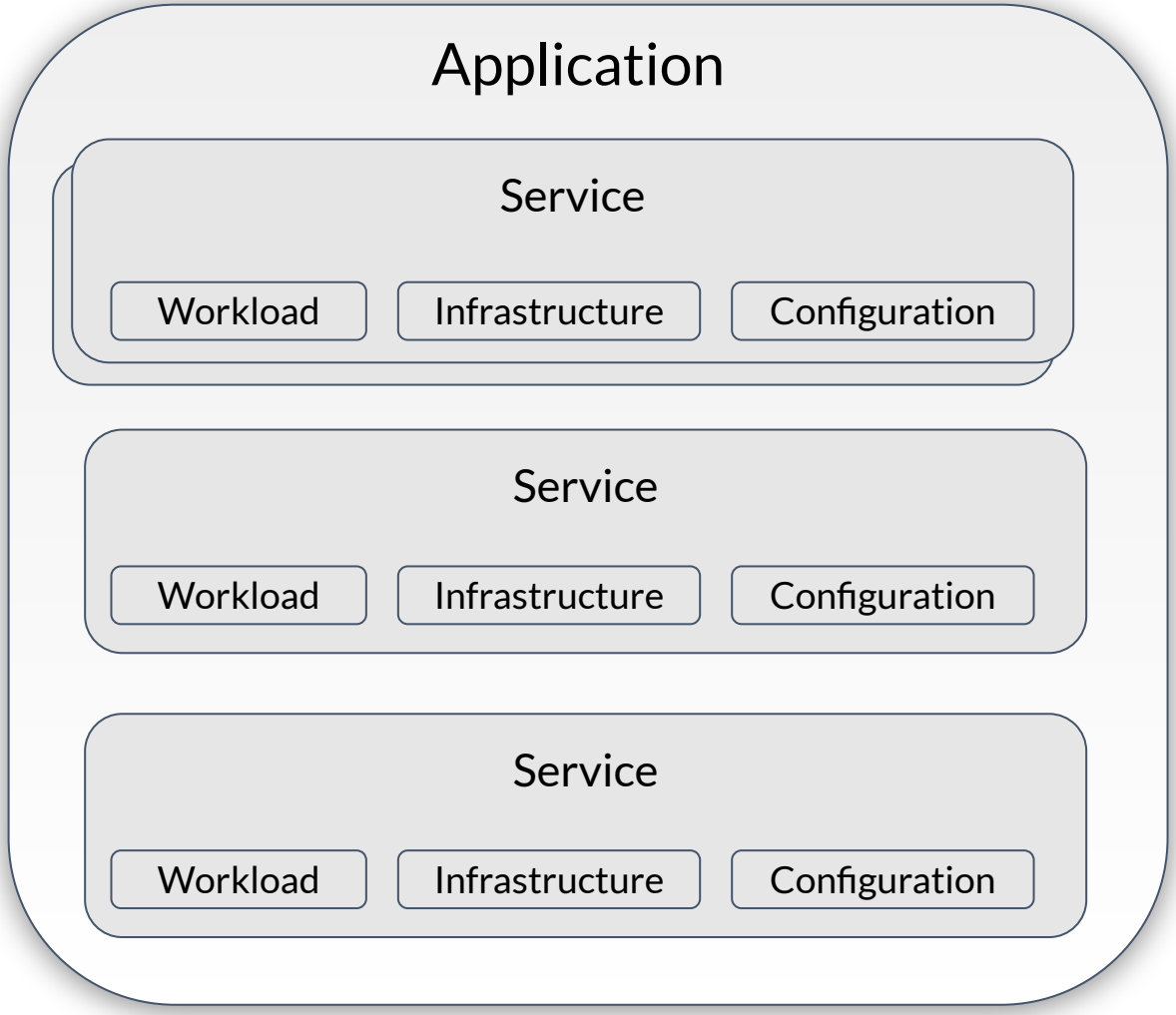
Operators increased adoption in the ecosystem reveal new patterns to be explored further

- Operator Whitepaper v2
- Call for contributions
- See more in https://tag-app-delivery.cncf.io
- Slack: #tag-app-delivery-operator-whitepaper
- https://github.com/cncf/tag-app-delivery/tree/main/operator-whitepaper/latest

Applications



What is a cloud application?



- Single or multiple Services
- Services scale independently
- Infrastructure and Configuration can be seen as part of the service

What is the problem?

Services are not always developed and tested individually

Knowing when an application deployment is finished

Workload (Deployment, StS, DS) Health Checks vs. Application Health

Standardization - How to switch between tools?

Application Definitions

How Projects are achieving this at the moment

Whole Application is defined in Resource

Reference to Source is stored in Resource

Reference to Objects is stored in Resource

Showcasing App Definitions

- ArgoCD
- KubeVela / OAM
- Keptn

Application Definitions - Next Steps

- Are you interested in this topic?
  => Reach out to us!
- Know more app definitions?
  => Provide examples for the PodTatoHead
- Expected outcome
  - Maybe a proposal for a standard
  - Blog Posts / Whitepaper in a WG
  - Examples

CNCF TAG App Delivery

tag-app-delivery.cncf.io

CLOUD NATIVE APP DELIVERY

About Whitepapers Blog

Search this site...

CNCF TAG App Delivery

TAG App Delivery supports projects and initiatives related to delivering cloud-native applications, including building, packaging, deploying, managing, and operating them.

Edit this page

Create issue

The TAG produces guidance for and gathers feedback from cloud app users and developers and provides guidance and coordination to CNCF projects in the TAG's technical domains.

- **TAG Charter**
- **Community events**
- Slack channel: **#tag-app-delivery**
  - **Invite yourself to the CNCF Slack**
- **Mailing list**

Call to Action!

- Share experiences and ideas in TAG meetings
- Website: https://tag-app-delivery.cncf.io
- Provide examples to the podtato-head
  - https://github.com/podtato-head/podtato-head-delivery
- Support the TAG's technical domains:
  - Build prototypes and patterns
  - Facilitate collaboration amongst projects
  - Share end user experiences and requirements

Please scan the QR Code above
to leave feedback on this session


KubeCon
CloudNativeCon
Europe 2023
TiKV