
Giant Swarm
Platform Reality Has a Surprising Amount of Detail
27/09/2023
KUBERNETES

Giant Swarm

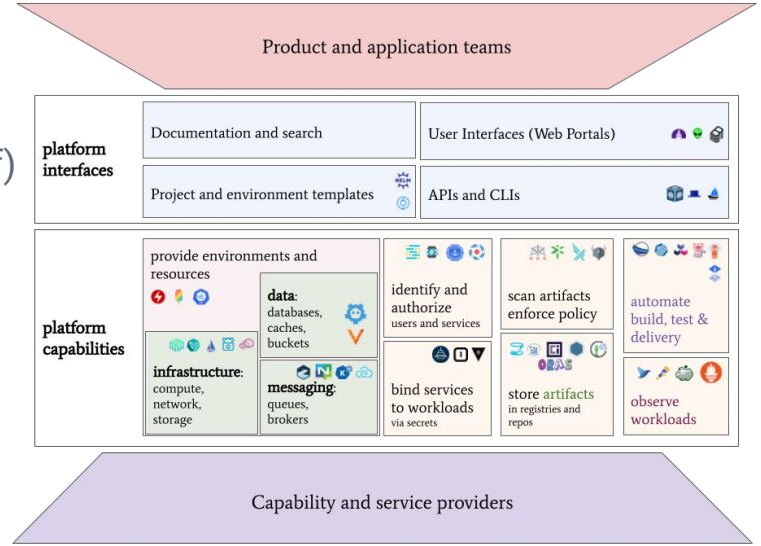
Recovering Academic Data Scientist, Ex-Developer-Advocate, CKA, CNCF Ambassador

KubeCon China | 2023

Puja Abbassi
VP Product
@puja108

Everyone is Talking about Platforms and Platform Engineering

- Many KubeCon talks and even some recent Keynotes like the one from Mercedes Benz
- What feels like almost all vendors in our ecosystem
- There's Platforms WG under CNCF TAG App Delivery
  - Platforms White Paper
  - Platform Maturity Model (coming soon)
  - Platform-as-a-Product White Paper (just kicking off)



Developer Platforms Should Enable Developers

To focus on the developers you need a product mindset

The platform whitepaper states the three core jobs of a platform team as follows:

Research platform user requirements and plan a feature roadmap.

Market, evangelize and advocate for the platform's proposed values.

Manage and develop interfaces for using and observing capabilities and services, including portals, APIs, documentation and templates, and CLI tools.

Platform Engineering Reality

Source: David Bell on X

Refocusing on the Developer Means Adopting Their PoV

- Capabilities need to be from the user perspective not tech or platform capabilities
  - Not “I want Grafana”, also not “I need Observability”, but rather “I need to debug my application and I need data to help me with that”
- Jobs-to-Be Done
  - Not “I want a drill”, but rather “I want a hole in the wall”
- User Journey Mapping

Example High-Level Capability: Progressive Rollouts

JTBD: I want to roll out my (iteration of an) application to production
Release engineering capabilities to build and deploy the software
Connectivity capabilities to dynamically route traffic to the new service
Observability capabilities to observe our new version and have automated feedback loops to roll back or go forward with the rollout to production
We want the whole software supply chain that facilitates this to be secure and trusted


Giant Swarm
How many cloud native tools do you need to integrate to achieve the above “simple” high-level capability in an automated and secure way?
KUBERNETES

Progressive Rollouts Based on Standards

GitOps (Flux & Flagger)  **Trigger Canary Deployment**

Gateway API (Linkerd & Contour)  **Dynamically Route Traffic to Canary**

Prometheus  **Observe Service**

sigstore & Kyverno  **Secure Software Supply Chain**

   Additional Base Infra Tools & Standards

Risks When Building Platforms

- PaaS-like abstraction leads to limitations, which can lead to (ugly) workarounds
- Bespoke Platforms risk getting stuck with maintenance and evolution
- The more we build ourselves, the more we risk getting overtaken by OSS (also applies to forks)
- The more we lock ourselves into specific tools, the more we risk a tool change blocking us from delivering value

Source: Nick Youngson CC BY-SA 3.0 Pix4free.org

The Solution is Betting on OSS and Community Standards

Risk Reduction Strategies:

- Multi-vendor OSS
- CNCF (or other foundation membership)
- Community Standards (Cluster API, Gateway API, sigstore,...)
- Make abstractions "poke-able"

Source: Nicole C. Baratta on Flickr

Stand on the Shoulders of Giants & Build Great Platforms

- Developer Platforms should enable Developers
- Focus needs to be on the developers' capabilities and their JTBD
- Do not underestimate the effort needed to build a developer platform
- Build on top of OSS and community standards
- Stay true to the core of open source
- Strive to make developers happy
- Stay happy as successful and appreciated platform teams

Source: Dylan Baugh on Flickr