



PRESENTER

Andreas Grabner
CNCF Ambassador
@grabnerandi

Successful Platform Engineering

Reality vs Hype

*From the viewpoint of a CNCF Ambassador
With a focus on „The Role of Observability“*

With inspiration from <https://ksick.dev>



OpenFeature



Houston: It seems we have a problem!



~40%

of engineering time
is productive

~36%

of developers leave because of
bad developer experiences

What happened?

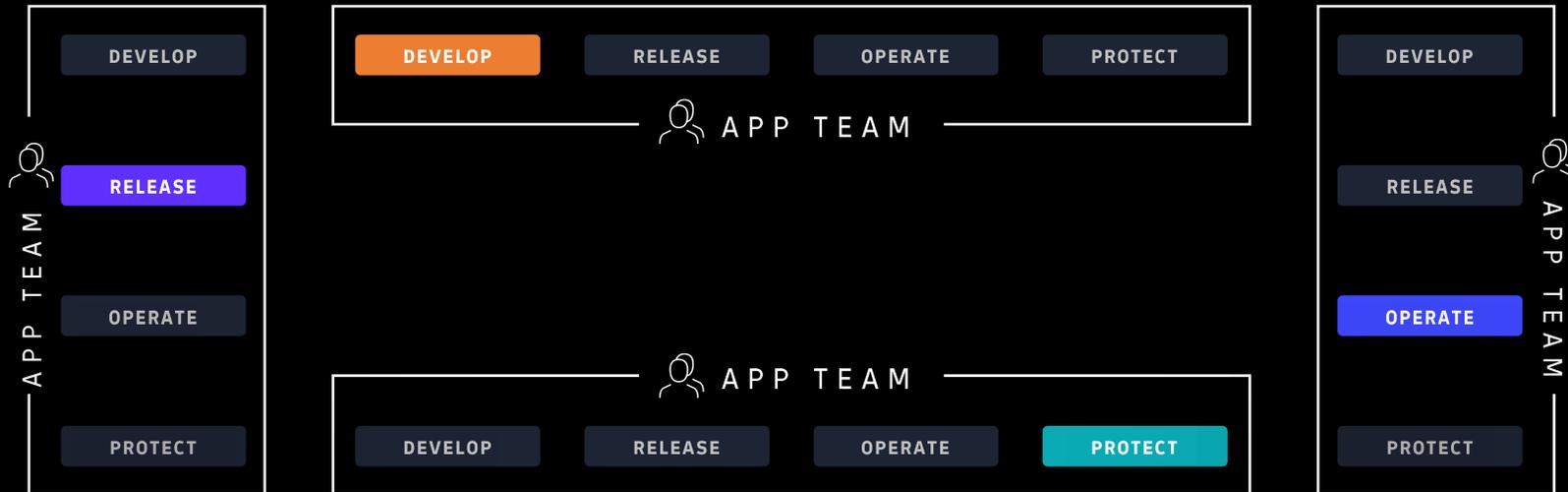


Shifting left!

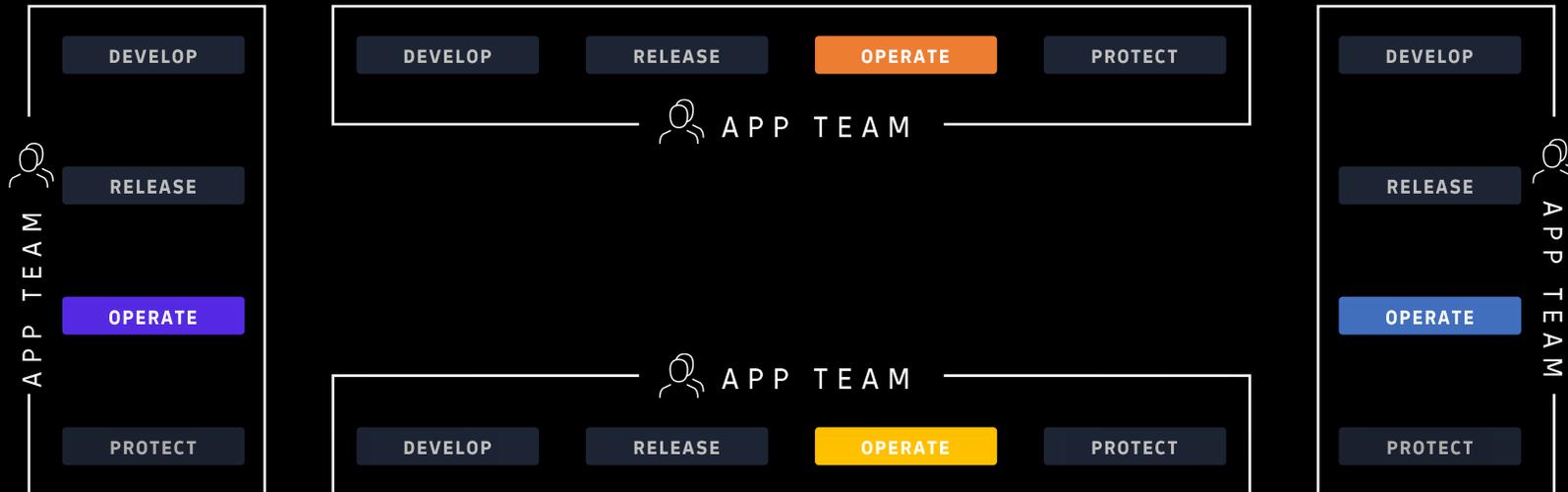
Teams taking on more responsibility



When Organizations grow - Teams start do things “their way”!



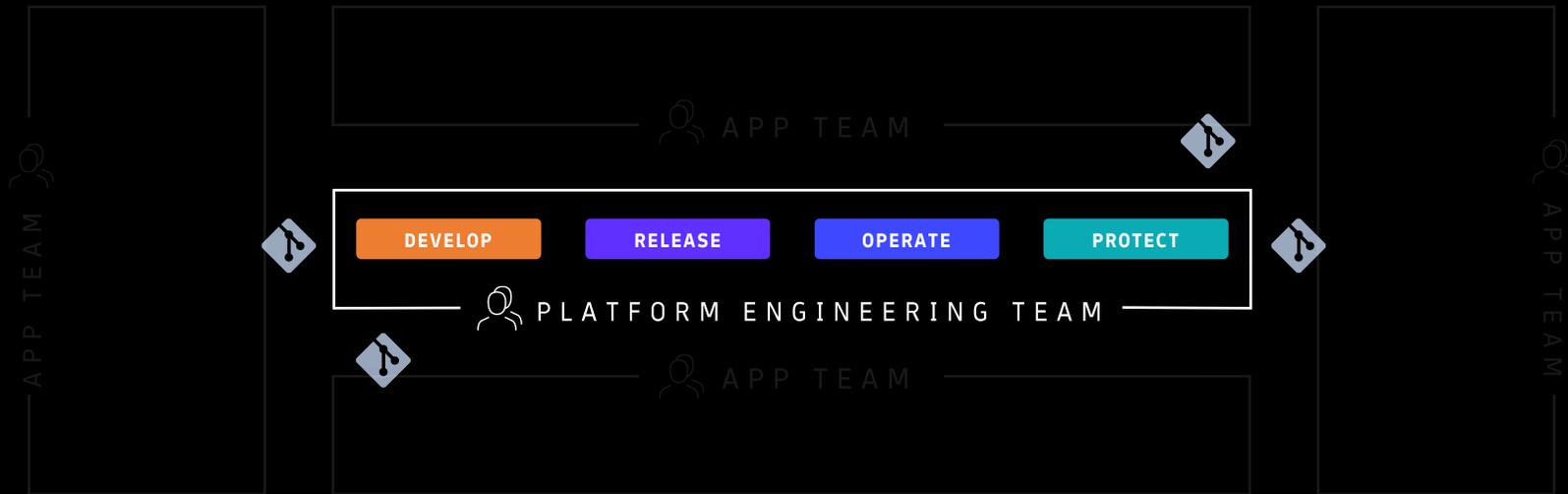
When Organizations grow - Teams start do things “their way”!

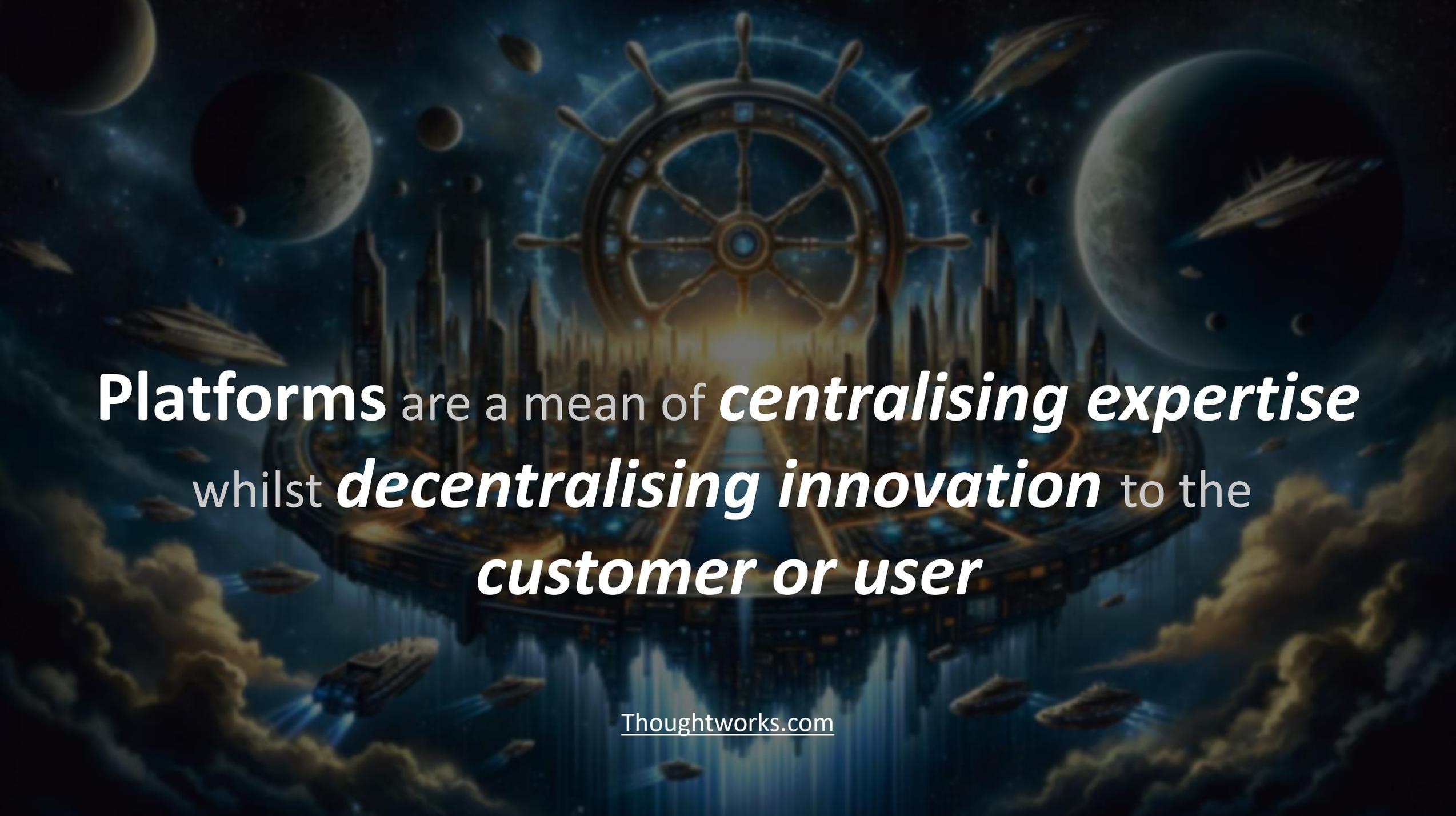


Central Services became the “Patch Solution”



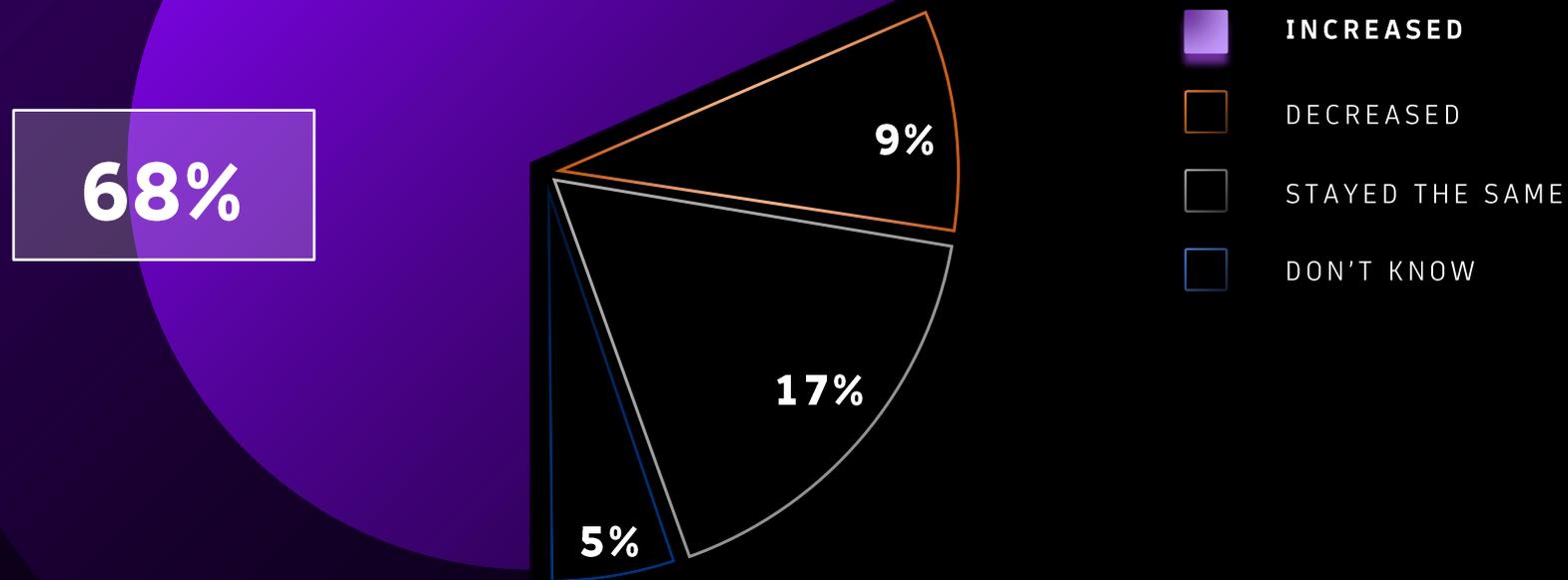
Leading to Platform Engineering: Git* as one Interface for Self-Service





Platforms are a mean of *centralising expertise*
whilst *decentralising innovation* to the
customer or user

Developer Productivity as Key Measure of Success to Platform Engineering!



More tangible KPIs to measure success

Business benefits achieved and lessons learned

Outcomes

83% of developer time is spend in productive activities

50K active projects with **12.7K** active users

42.7M pipeline runs by 2023 (*up from 3M, 2019*)

324K user stories by 2023 (*up from 22K in FY20*)

99% of MTT Instrument reduction (days down to minutes)

Significant reduction in incident count with major improvements in code quality

Lessons learned

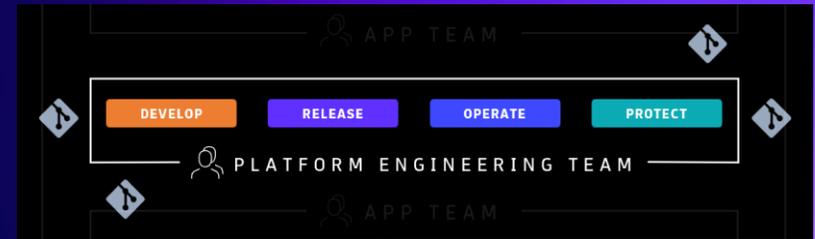
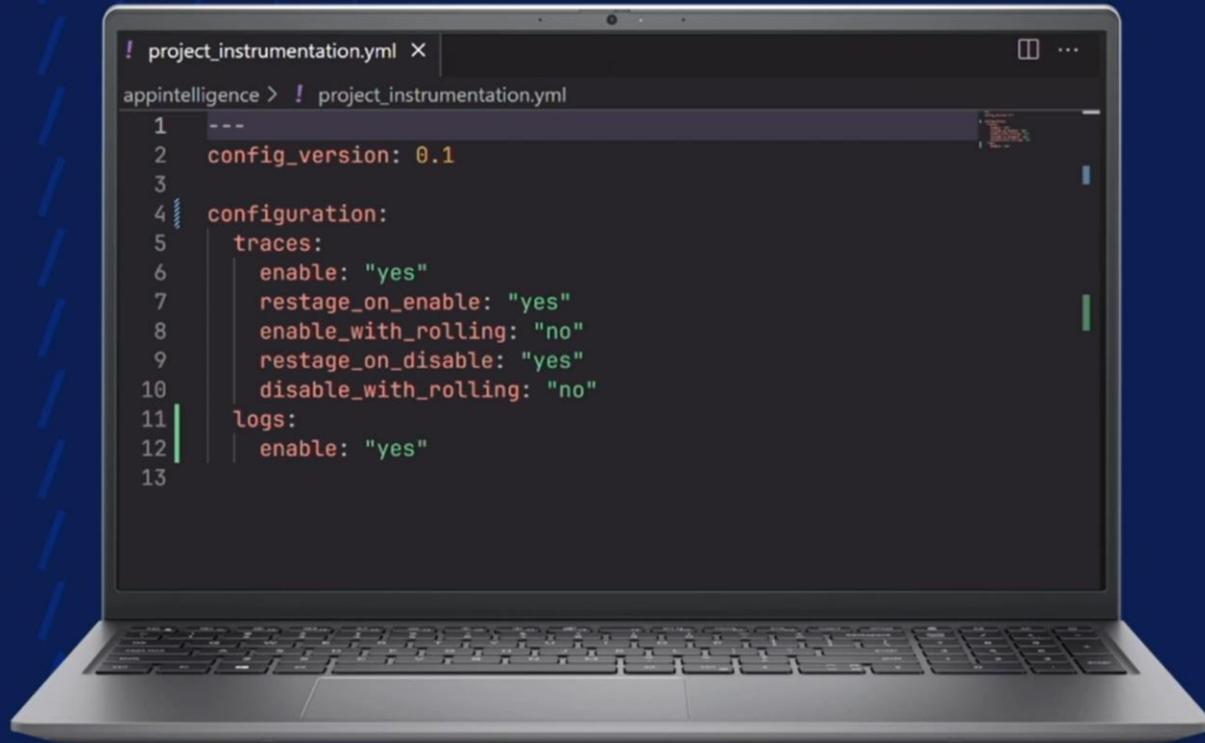
- ✓ Understand the developer journey and how they spend their time
- ✓ Automating standards enables developers to code quickly & with increased security
- ✓ Provide flexibility, but with guardrails
- ✓ Use metrics to measure progress
- ✓ Put the customer at the center (consider product model)

Dell Digital Proven 

 Dell Technologies

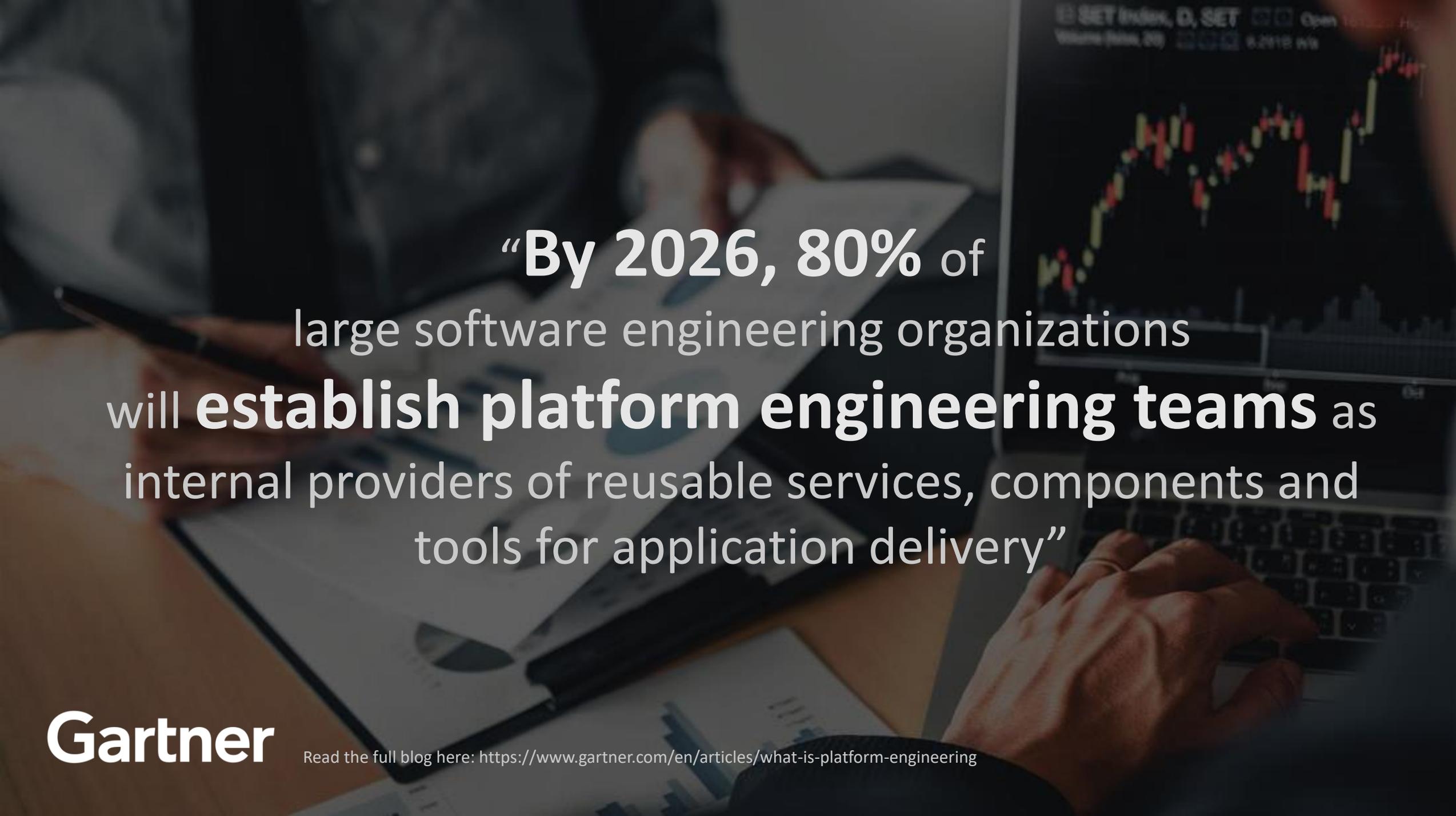


Git enables “Observability as Self-Service”



Dell Digital Proven

Dell Technologies



“By 2026, 80% of large software engineering organizations will **establish platform engineering teams** as internal providers of reusable services, components and tools for application delivery”

Gartner

Read the full blog here: <https://www.gartner.com/en/articles/what-is-platform-engineering>



“80% of all organizations
indicate expanding use, using, or piloting an
IDP (Internal Developer Platform) –
compared to
55,9% in the previous year.”



Read the full blog here: <https://www.idc.com/getdoc.jsp?containerId=US51622924>

**So – that's it!
Everyone is
happy!**

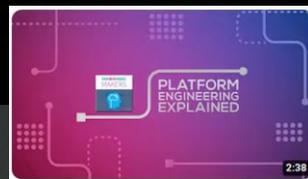


My job is done here 😊

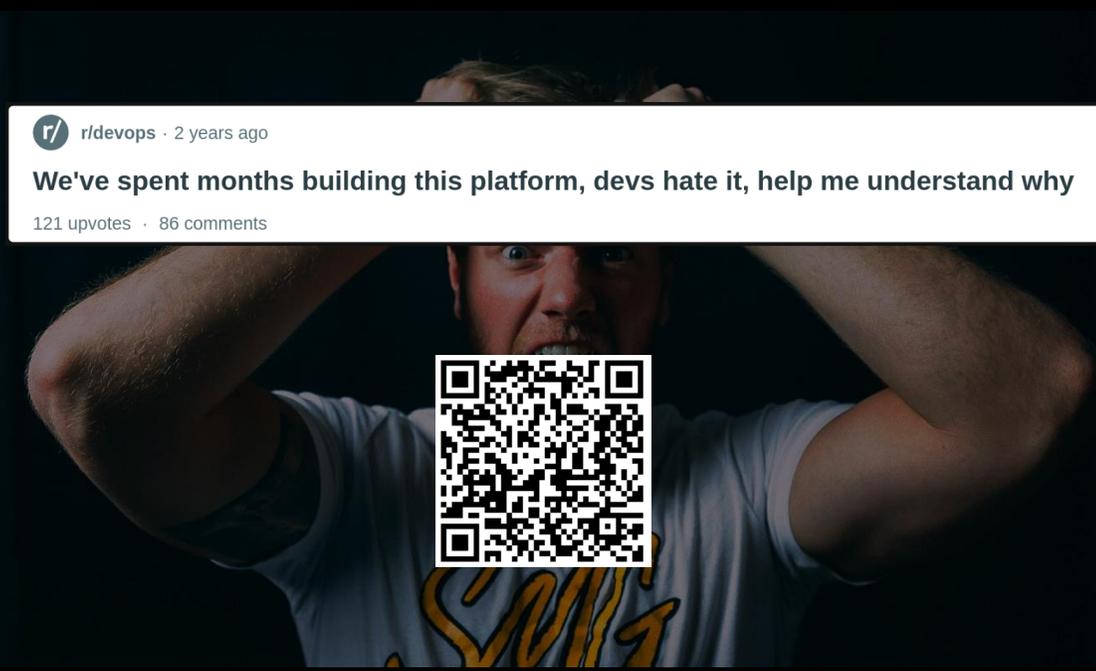
Hype

vs

Reality



 r/devops · 2 years ago
We've spent months building this platform, devs hate it, help me understand why
121 upvotes · 86 comments



How to Build a Successful Platform?



Find the **pain points** and **needs** of your users



Have a clear **mission** with measurable **KPIs**



Design for a good **User/Dev Experience**



Start with the **TVP (Thinnest Viable Platform)**



Make everything **observable**

Advice: Build your Platform like a Product!

SELF SERVICE PLATFORM END-USER FEATURES

Dev DevOps Security SRE Support

USER INTERFACE(S) TAILORED TO THE NEEDS OF YOUR USERS

Command Line REST API Git Repo Dev Portal Something else

```
1 ---
2 config_version: 0.1
3
4 configuration:
5   traces:
6     enable: "yes"
7     restage_on_enable: "yes"
8     enable_with_rolling: "no"
```

service ☆

DTP Application Onboarding

THIS TEMPLATE IS WORK IN PROGRESS

This template is used to onboard a new Apps

Request ♡

Database as a Service for MSSQL

Create a Database on SQL Managed Instance or the...

CORE PLATFORM TAILORED TO FIT THE REQUIRMENTS

Delivery Services

Platform Services

Platform

X-as-Code

Observability

Availability, Resiliency, Security

Observed SUCCESS KPIS

Active Users
NPS SPACE

SLOs
DORA

FinOps
Utilization

Use Case: Self-Service Observability @ Bank



Devs uses template: git commit!

Journey

Owner

SLO

```
{
  "deployTemplate": "java-service",
  "dtServicePriority": "false",
  "manifestDirPath": "BOOT-INF/classes",
  "AksConfigsDirPath": "BOOT-INF/classes",
  "journey": [
    "HOME", "OMNI-PLATFORM"
  ],
  "internalAutoScaleConfig": "true",
  "setEurekaHostname": true,
  "appType": "apptype-spring-boot",
  "squad": [
    "BLUE-STEEL",
    "SPRINTENDO",
    "SQUAD-RABBITHOLE"
  ],
  "appName": "home-client-api",
  "dynatraceConfig": {
    "dtEntityConfig": [{
      "dtEntity": "ClientController",
      "dtEntityType": "SERVICE",
      "dtTargetSLO": 99.98,
      "dtWarningSLO": 99.99,
      "dtSLOType": "Availability"
    },
    {
      "dtEntity": "ClientContactDetailsController",
      "dtEntityType": "SERVICE",
      "dtTargetSLO": 99.97,
      "dtWarningSLO": 99.98,
      "dtSLOType": "Availability"
    }
  ]
}
```

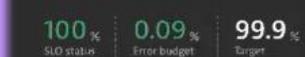
SLO



Automated Observability and Targeted Alerting to Owners

UE90 Service Level Objectives

BANK WITH CONFIDENCE



★ UE90 OMNI OVERVIEW (PROD)

OVERVIEW



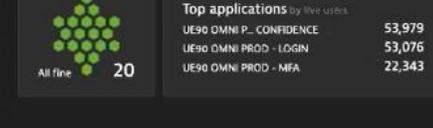
SERVICE LEVEL OBJECTIVES

RELEASE MONITORING

REAL USER MONITORING



OMNI PLATFORM



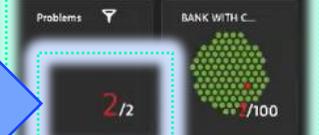
PLATFORM



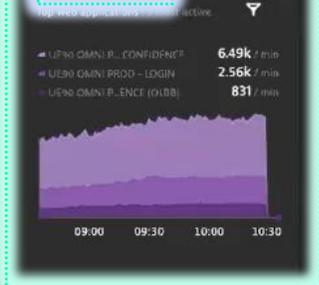
Journey

Ownership

BANK WITH CONFIDENCE



CHOOSE A BANK



PCF HOSTS



DATABASES



Tip: Follow Cloud Native Best Practices

Community providing practices on releases / apps



Labels

In order to take full advantage of using these labels, they should be applied on every resource object.

Key	Description
app.kubernetes.io/name	The name of the application
app.kubernetes.io/instance	A unique name identifying the instance of an application
app.kubernetes.io/version	The current version of the application (e.g., a SemVer 1.0 , revision hash, etc.)
app.kubernetes.io/component	The component within the architecture
app.kubernetes.io/part-of	The name of a higher level application this one is part of
app.kubernetes.io/managed-by	The tool being used to manage the operation of an application

```
kind: Deployment
metadata:
  name: accountservice
spec:
  selector:
    matchLabels:
      app: accountservice
  template:
    metadata:
      labels:
        app: accountservice
        app.kubernetes.io/version: 44e1878
        app.kubernetes.io/part-of: easytrade
        dt.owner: team-workshop
```



Devs: Label your CRDs



SREs:
Get release-aware
observability

Name	Version	Stage	Product
contentcreator.easytrade	44e1877	easytrade	easytrade
simplenode.simplenode-dev	3.0.2	simplenode-	simplenode-product
offerservice.easytrade	44e1877	easytrade	easytrade
accountservice.easytrade	44e1878	easytrade	easytrade
engine.easytrade	44e1877	easytrade	easytrade
aggregator-service-*.easytrade	44e1877	easytrade	easytrade

Use Case: Self-Service Feature Flagging



Developer adds **OpenFeature**

```
OpenFeature.setProvider(new MyProvider());  
const featureFlags = OpenFeature.getClient();  
  
const withCows = await featureFlags.getBooleanValue("with-cows", false);  
if (withCows) {  
  res.send(cowsay.say({ text: "Hello, world!" }));  
} else {  
  res.send("Hello, world!");  
}
```



DevOps control rollout of new features via Git, e.g: flagd provider

The screenshot shows the Argo CD web interface. On the left, a sidebar lists 'Applications', 'Settings', 'User Info', and 'Documentation'. The main area displays the 'feature-flags' application details for the 'gartner-demo' namespace. The 'SUMMARY' tab is active, showing the following information:

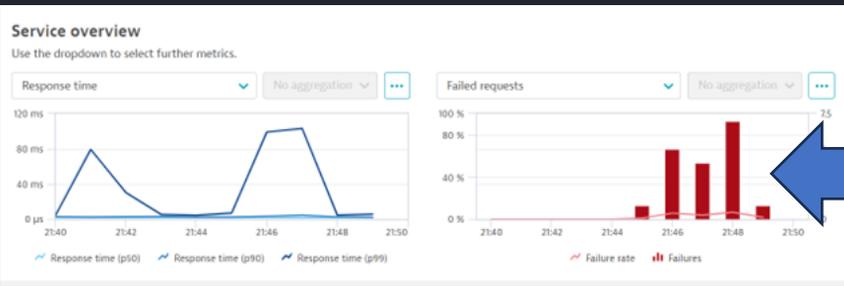
- KIND: FeatureFlag
- NAME: feature-flags
- NAMESPACE: gartner-demo
- CREATED AT: 02/29/2024 17:33:41 (4 hours ago)
- STATUS: OutOfSync

Below this, there are sections for 'LIVE MANIFEST' and 'DESIRED MANIFEST' with a 'DIFF' view. The diff shows a change in the 'defaultVariant' for the 'enable-new-feature' flag:

LINE	KEY	VALUE	LINE	KEY	VALUE
64	red: '#b01c1c'		64	red: '#b01c1c'	
65	enable-new-feature:		65	enable-new-feature:	
66	defaultVariant: 'false'		66	defaultVariant: 'true'	
67	state: ENABLED		67	state: ENABLED	
68	variants:		68	variants:	



Developer gets immediate feedback



Use Case: Automated FinOps @ Bank



Devs provision resource

Request Request for Windows and Linux Virtual Machines

Azure Virtual Machine Self-Service

Request for Windows and Linux Virtual Machines

Tag Details

Options

- Is this request for a Technical Service?
- Alphabet Application in draft / planning state?

*Application Id

Nedbank Business Authenticator

*Maintenance Slot

After Business Hours only

*Environment

Quality Assurance

*Business Hours

Next Business Day (Mon-Fri)

*Snoozing Options

Please Select

- Option-0 (No Snooze)
- Option-1 (Mon-Fri 7.00pm to 6.00am)
- Option-2 (Mon-Fri 11.00pm to 6.00am)
- Option-3 (Mon-Sun 7.00pm to 6.00am)**
- Option-4 (Mon-Sun 11.00pm to 6.00am)
- Option-5 (Fri-Mon 7.00pm to 6.00am)



SRE / FinOps have observability to and optimize Carbon & Costs across all environments



Emissions in the last 7 days by Hypervisor - gCO2eq

Hypervisor	gCO2eq
VMWARE	15,914.7
AWS_NITRO	75,566.13
LPAR	227,477.44

60%

Cost Reduction Target

Use Case: Self-Service Onboarding @ Dynatrace



The collage shows four screenshots from different tools:

- Jira:** A workflow action configuration for 'Send email' with details like 'Type: ValueIncrement', 'Priority: Undefined', and 'Component/s: Cloud Automation'.
- Dynatrace:** A 'Create a new component' screen for 'DTP Application Onboarding' with a list of tasks like 'Create catalog-info.yaml in the app source code'.
- Bitbucket:** A repository view for 'email-app' showing a 'changeset' in the 'src' directory.
- Gitpod:** A 'New Workspace' creation screen for 'email-app' with options for 'VS Code' and 'Standard' environment.



Tip: Platform Observability from git to prod



Backstage

Jenkins

HARBOR

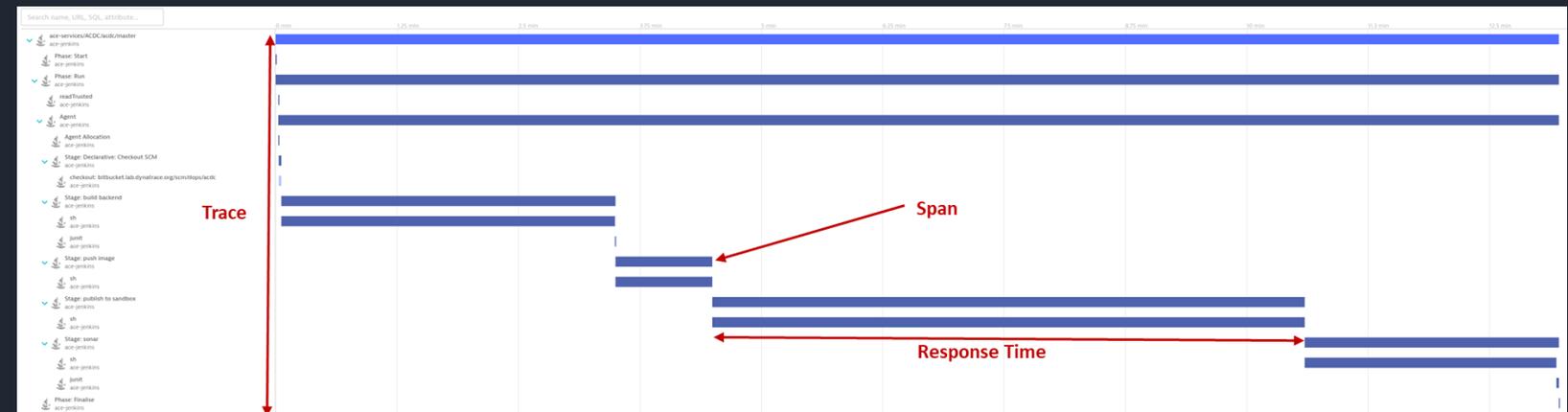
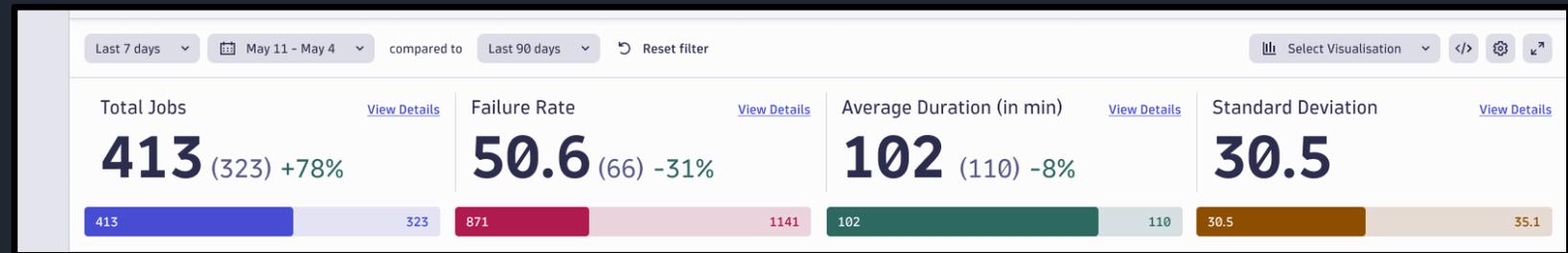
Kyverno

Argo CD

Crossplane



Platform Engineering: Do we have any issues and bottlenecks in our pipelines?



Tip: Platform Observability from git to prod



git commit



Platform Engineering: Is our container registry running smoothly? Which teams use it?



Tip: Platform Observability from git to prod

git commit

Backstage

Jenkins

HARBOR

Kyverno

Argo CD

Crossplane



Platform Engineering: No vulnerabilities in prod? All policies enforced?

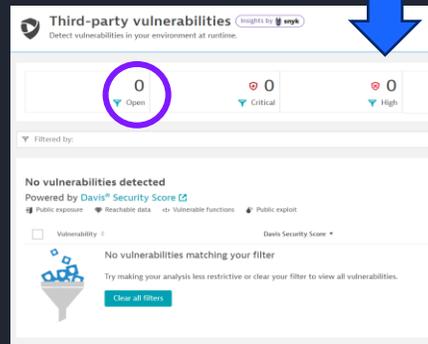
DEV



HARDENING



PROD



Policy insights

K8S with/without admission controller



k8s clusters without admission controller

entity.name

K8s Cloud Automation Dev claus-dev-01-us-east-1-cluster
SPINE Dev
doaks-dev-dok-daily
doaks-dev-e2e-private
doaks-dev-priv-ajoskowski
doaks-dev-readinnesse2e-westeu
dogke-dev-dok-daily

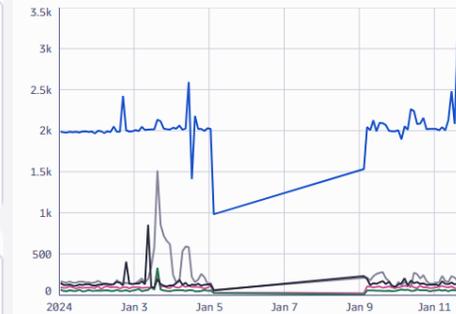
k8s clusters with admission controller

entity.name	k8s.admissioncontroller.entity.name
dok-gitops-argo-dev	kyverno-admission-controller
dok-gitops-argo-sandbox	kyverno-admission-controller
ntp-dev-csc-central-services	kyverno-admission-controller
ntp-dev1-grail	kyverno-admission-controller
ntp-dev2-apigw	kyverno-admission-controller
ntp-dev2-apprt	kyverno-admission-controller

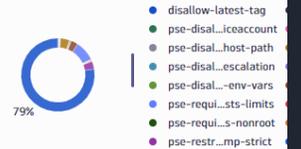
Policy violations by cluster



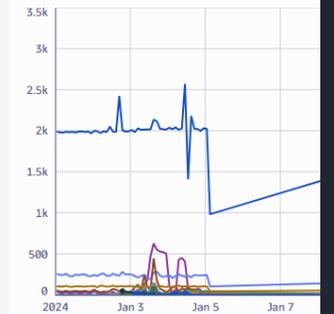
Violations by cluster (trend)



Violations by policy



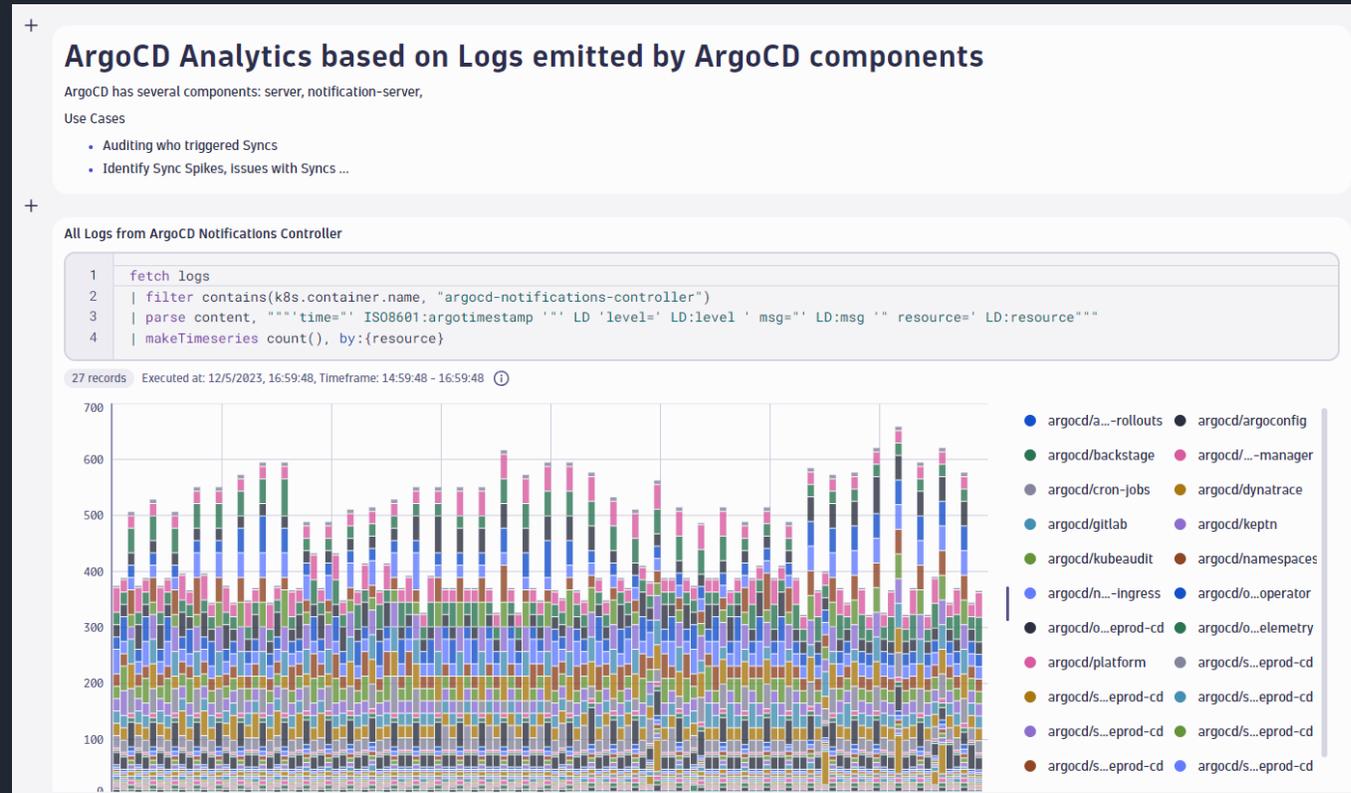
Violations by policy (trend)



Tip: Platform Observability from git to prod



Platform Engineering: How many apps are onboarded? How well does it run?



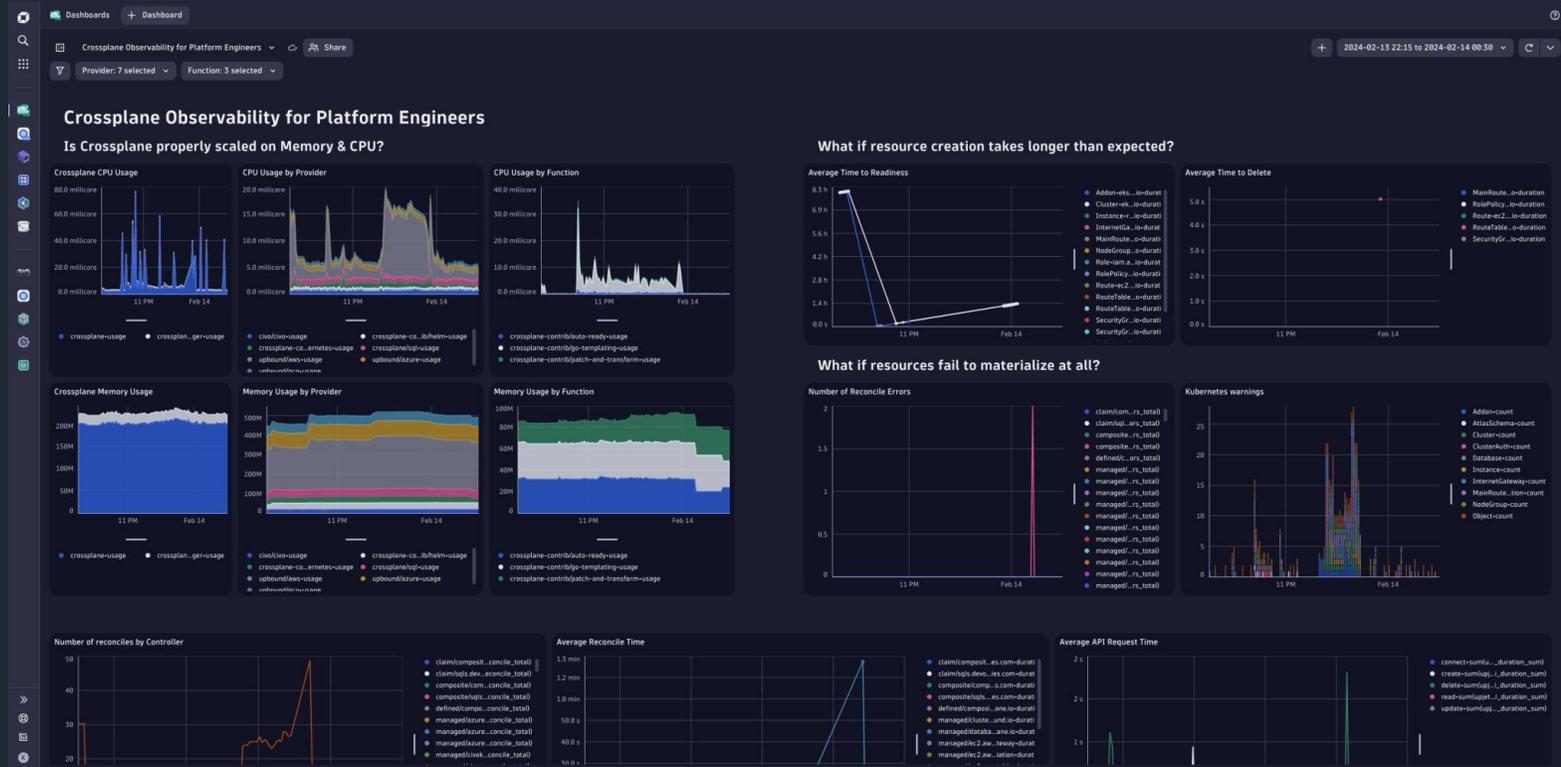
Tip: Platform Observability from git to prod



git commit



Platform Engineering: Can we deploy in-time or do we have issues in stages?



Recap: Build your Platform like a Product!

SELF SERVICE
PLATFORM END-USER FEATURES



Dev



DevOps



Security



SRE



Support

USER INTERFACE(S)

TAILORED TO THE NEEDS OF YOUR USERS



Command Line



REST API



Git Repo



Dev Portal



Something else

```
1 ---
2 config_version: 0.1
3
4 configuration:
5   traces:
6     enable: "yes"
7     restage_on_enable: "yes"
8     enable_with_rolling: "no"
```

service ☆

DTP Application Onboarding

THIS TEMPLATE IS WORK IN PROGRESS

This template is used to onboard a new Apps

Request ♡

Database as a Service for MSSQL

Create a Database on SQL Managed Instance or the...

CORE PLATFORM

TAILORED TO FIT THE REQUIRMENTS

Delivery Services



Platform Services



Platform



X-as-Code



Cloud Monit or APIs

Observability

Availability, Resiliency, Security

Observed
SUCCESS KPIS

Active Users
NPS SPACE

SLOs
DORA

FinOps
Utilization

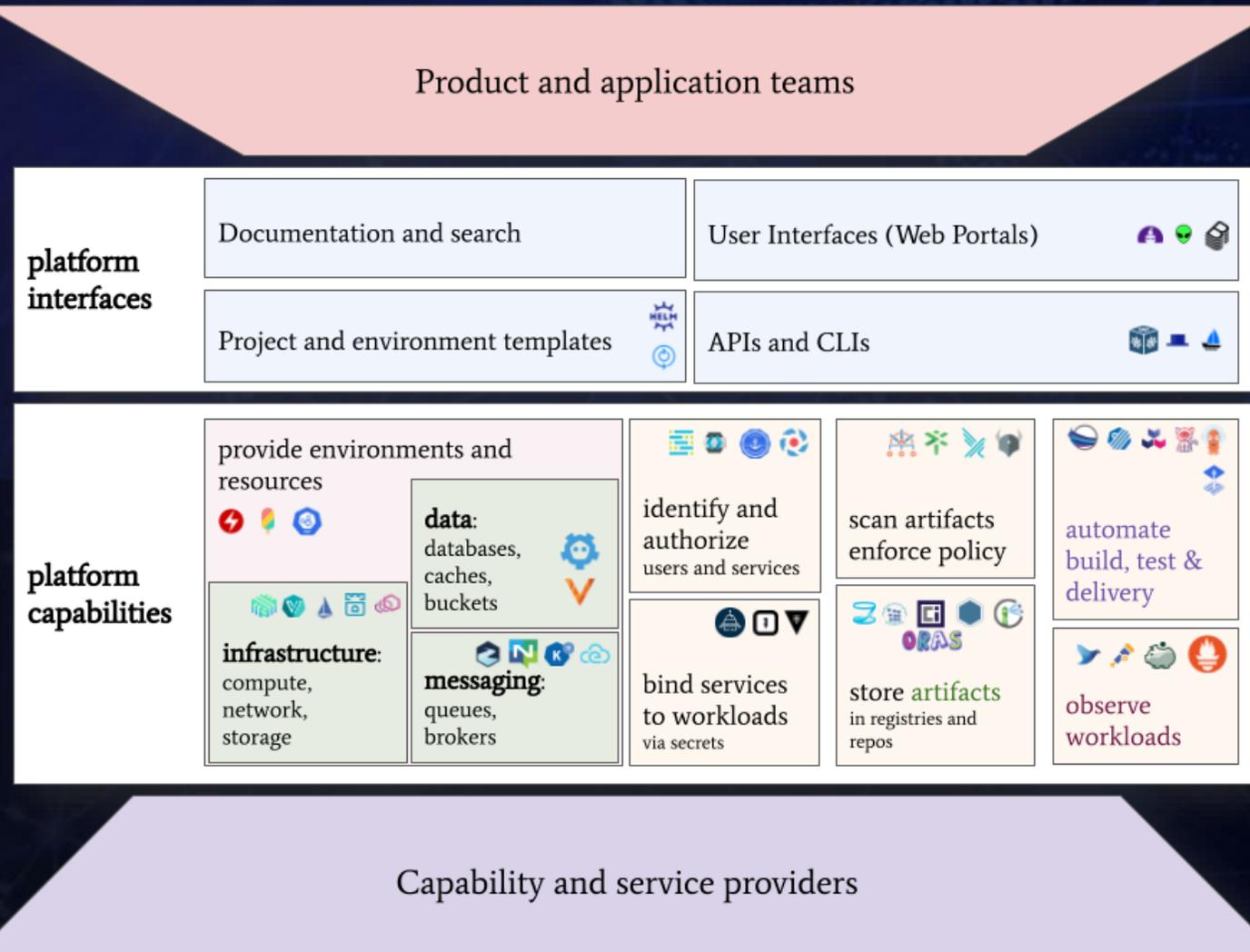
Additional Resource: CNCF White Paper



CNCF Platforms White Paper:
<https://tag-app-delivery.cncf.io/whitepapers/platforms>



cncf.io | info@cncf.io



Additional Resource: OpenGitOps.dev



<https://opengitops.dev/>



CLOUD NATIVE
COMPUTING FOUNDATION

cncf.io | info@cncf.io

What is OpenGitOps

OpenGitOps is a set of open-source standards, best practices, and community-focused education to help organizations adopt a structured, standardized approach to implementing GitOps.

GET INVOLVED

[GitOps Principles](#) [Upcoming Events](#)

Additional Resource: GitHub Tutorial



<https://dt-url.net/devrel-PE-demo>

Developer Self-Service: Onboarding an app
with automated observability & security



Open
Feature



Platform Engineering White Paper



<https://dt-url.net/ebook-plateng-angr>

FREE EBOOK

Driving DevOps and platform engineering for digital transformation

Amid the ever-evolving landscape of software development, the debate surrounding DevOps vs. platform engineering has ignited. Which is the preferred approach to software delivery? Are these two disciplines interlinked?

Download the free ebook to discover the following:

- The nuances, best practices, and tangible benefits that DevOps and platform engineering bring to organizations.
- The challenges and complexities that organizations face in adopting and optimizing these methodologies.
- How a unified observability platform helps organizations unlock transformative potential with their platform engineering use cases.

What's inside

CHAPTER ONE

What are DevOps and platform engineering?

CHAPTER TWO

Four ways DevOps and platform engineering complement one another

CHAPTER THREE

Four benefits of DevOps and platform engineering

CHAPTER FOUR

Four challenges of DevOps and platform engineering

CHAPTER FIVE

Four ways platform engineering scales DevOps in a cloud-native world

CHAPTER SIX

Four evolving practices in DevOps and platform engineering

CONCLUSION

Unified observability and security for DevOps and platform engineering

Successful Platform Engineering: Hype vs Reality!

Status Quo

~40%

of engineering time
is productive

~36%

of developers leave because of
bad developer experiences

Lessons learned on how to get there



Find the **pain** points



Have a clear **mission**



Design for **DevExperience**



Start with the **TVP**



Make everything **observable**

What we see

~83%

of engineering time is
productive

~66%

report increased dev efficiency /
dev experience

All the resources in one slide



Report

"State of DevOps 2023"



CNCF White Paper

"WG on Platform Engineering"



GitHub Tutorial

<https://dt-url.net/devrel-PE-demo>



OpenGitOps

<https://opengitops.dev/>



Katharina Sick

<https://ksick.dev/>

Thank You

From DevOps to Platform Engineering and Beyond!



PRESENTER

Andreas Grabner
CNCF Ambassador
@grabnerandi



OpenFeature

