

Platform Engineering for Software Developers and Architects (Redux) - Daniel Bryant, Syntasso

KubeCon CloudNativeCon Europe 2025

Platform Engineering for Software Developers and Architects (Redux)

Daniel Bryant, Syntasso

Watch on YouTube

Building an AI-Powered, Paved Road Platform with Cloud-Native OSS - Todd Ekenstam & Avni Sharma

Platform Engineering Day EUROPE

Building an AI-Powered, Paved Road Platform with Cloud-Native OSS

Todd Ekenstam & Avni Sharma, Intuit

Watch on YouTube

Platform Engineering Day Zero, The Origin Story - Murriel McCabe, Google

KubeCon CloudNativeCon North America 2025

Platform Engineering: Day Zero, The Origin Story

Murriel McCabe, Google

Watch on YouTube

PLATFORM BLUEPRINTS

Build golden paths for day 50, not only day 1!

Kaspar von Grünberg
CEO
Humanitec

PLATFORM CON23



KubeCon CloudNativeCon North America 2025

Economics of Platforms: Building Marketplaces Beyond Golden Paths

Atulpriya Sharma, InfraCloud Technologies



KubeCon CloudNativeCon Europe 2023

Paved Paths Leading the Way to Compliance

Kasper Borg Nissen & Brian Nielsen, Lunar



Find Your Golden Path



2:10

Blueprints SPONSORED TALK

Weaving intents into reality: Defining your own intent-driven golden path

Zibo He
Senior Technical Expert
Ant Group

PLATFORM CON24



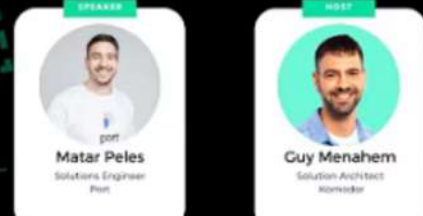
THE PLATFORMERS

GOLDEN PATHS THROUGH THE DEVELOPMENT LIFECYCLE

WEDNESDAY 26TH MARCH 06:00 PM CET 1:00 PM ET

Speakers: Matar Peles, Guy Menahem

58:12



You went **Back**
You **Stared** at scripts

Today we **Fix** that

The bottom of the page features a decorative graphic consisting of multiple thin, overlapping wavy lines in shades of gray. These lines form a series of peaks and valleys, resembling a stylized landscape or a series of sound waves. The lines are more densely packed in some areas, creating a sense of depth and movement.

Build Your Golden Path

*A Maturity-First Implementation
Approach*

March 23, 2026

Atulpriya Sharma


 Hyderabad, India

 Sr. Developer Advocate, Improving (formerly InfraCloud)

 Co-Chair, KubeCon India'25, CNCF Ambassador, CNCG Hyderabad

 Platform Engineering, DevOps, Developer Relations, GenAI

   @TheTechMaharaj

 Atulpriya Sharma



atulmaharaj [Follow](#) [Message](#) [...](#)

1,042 posts 4,064 followers 2,279 following

Atulmaharaj
 Blogger
 🌐 Hyderabad Food, Travel & Lifestyle Blogger
 ✍️ 10+ Years Blogging
 🎯 6M+ Blog Views
 🏆 Best Blogger Award
 ⚠️ Insta Is Not Blogging
 #Food #Travel #Blogger
 🌐 socialmaharaj.com

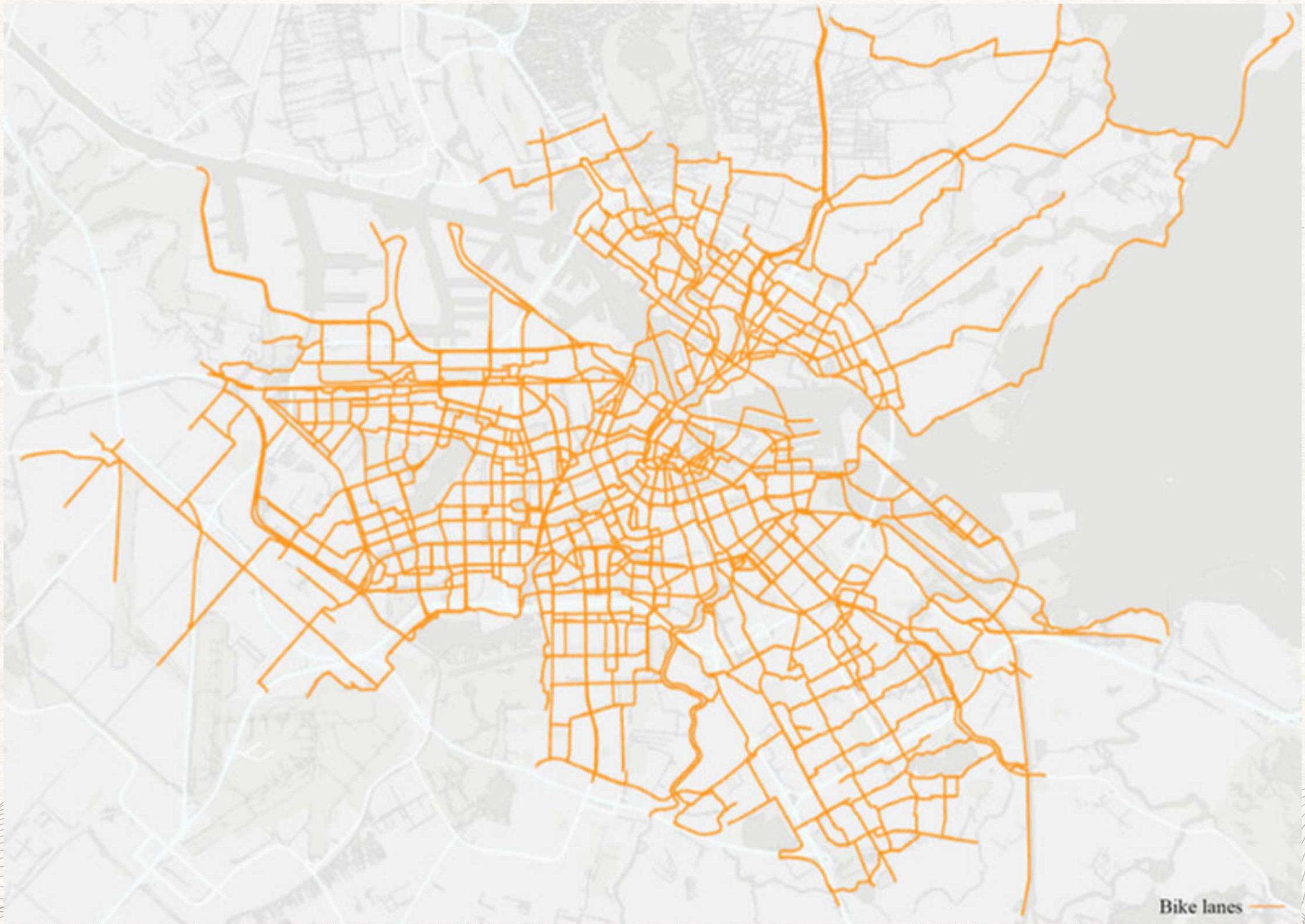
NAV 🇹🇷 '25 IST 🇹🇷 '25 LHR 🇬🇧 '25 EDI 🇳🇴 '25 FRA 🇩🇪 '24 LAX 🇺🇸 '24



  @Atulmaharaj

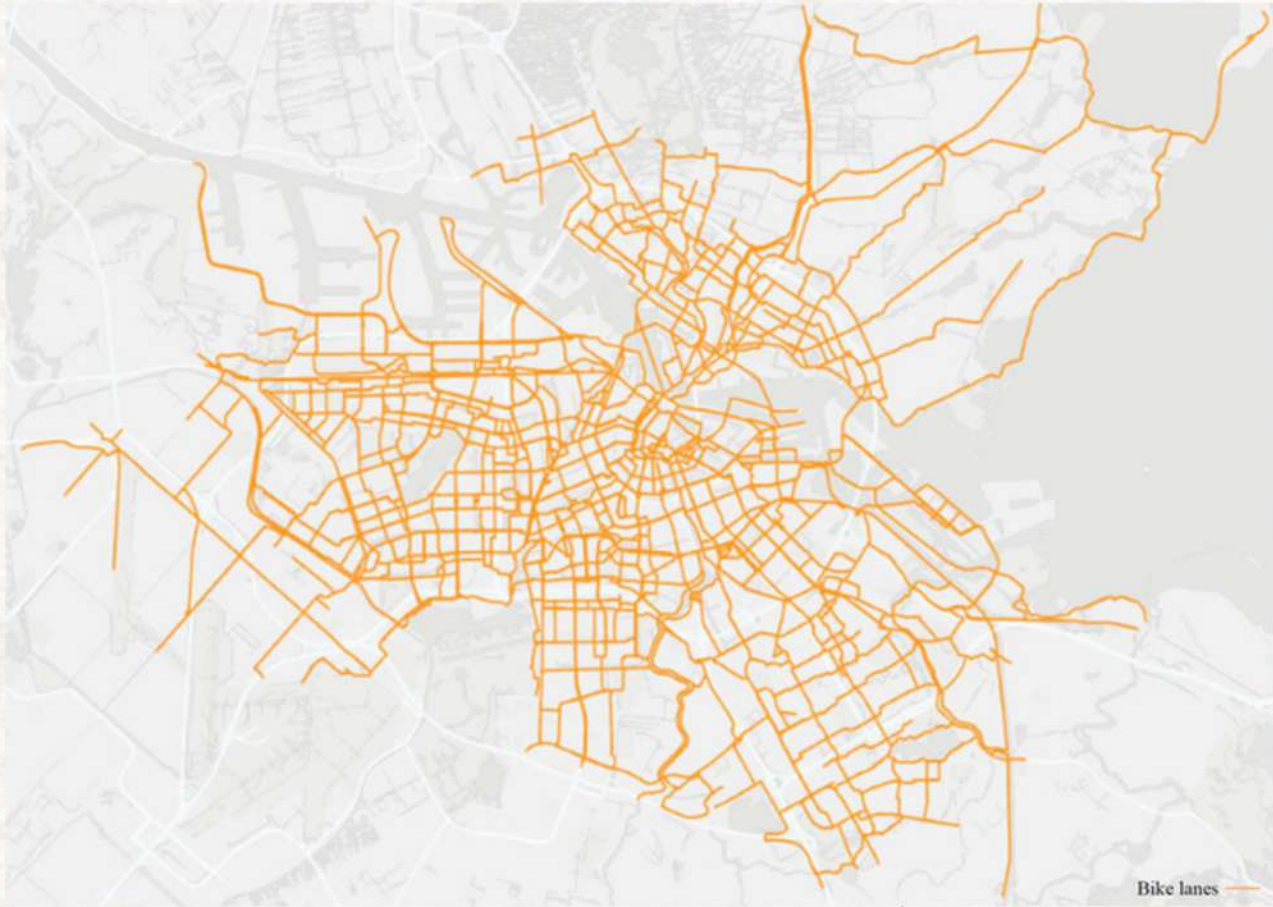
 <https://socialmaharaj.com>





Bike lanes

Amsterdam's **Golden Path**

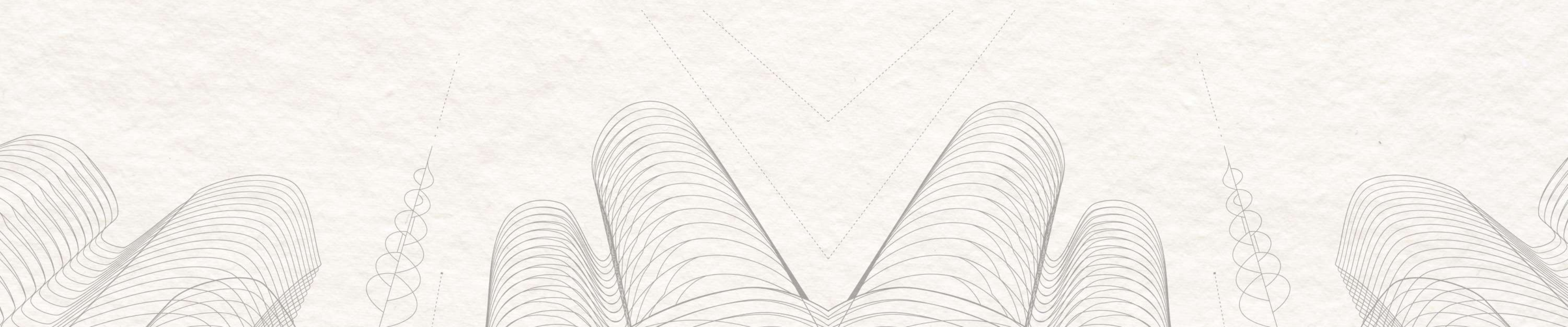


- Opinionated
- Self-service
- Safe by default
- Reduces cognitive load
- Built progressively

Golden Path - 101

- A supported, opinionated way to build and deploy software
- Makes the right way the easiest way
- Not a mandate - a product
- Reduces decision fatigue for developers
- Platform team's job: pave it, maintain it, evolve it

Where Do You
Actually **Start?**



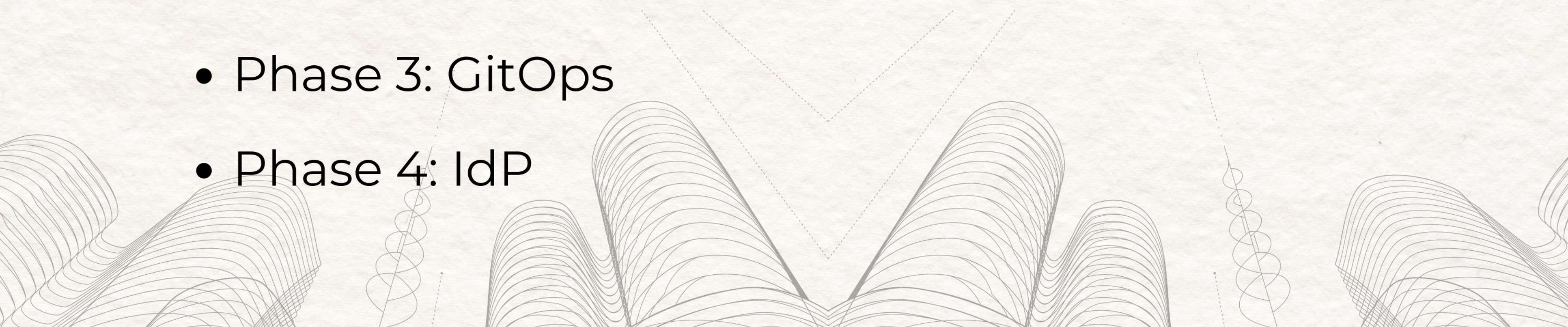
ASPECT		PROVISIONAL	OPERATIONAL	SCALABLE	OPTIMIZING
Investment	<i>How are staff and funds allocated to platform capabilities?</i>	Voluntary or temporary	Dedicated team	As product	Enabled ecosystem
Adoption	<i>Why and how do users discover and use internal platforms and platform capabilities?</i>	Erratic	Extrinsic push	Intrinsic pull	Participatory
Interfaces	<i>How do users interact with and consume platform capabilities?</i>	Custom processes	Standard tooling	Self-service solutions	Integrated services
Operations	<i>How are platforms and their capabilities planned, prioritized, developed and maintained?</i>	By request	Centrally tracked	Centrally enabled	Managed services
Measurement	<i>What is the process for gathering and incorporating feedback and learning?</i>	Ad hoc	Consistent collection	Insights	Quantitative and qualitative

The Platform Maturity Model tells you where you are.

Read more about the **Platform Maturity Model**



The Maturity-First Construction Framework

- Phase 0: Chaos
 - Phase 1: Standardize
 - Phase 2: Validate
 - Phase 3: GitOps
 - Phase 4: IdP
- 

Time to **Build**

The Maturity-First Construction Framework



GitHub Repo

Phase 0 - The Chaos

- Every team has their own deployment script
- Same goal, zero consistency, no standards
- latest tags, missing limits, broken health checks



\$ team-alpha-deploy.sh

\$ team-beta-deploy.sh



phase1-chaos > \$ team-alpha-deploy.sh

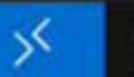
```
10 spec:
15   template:
19     spec:
22       image: nginx:latest           # X latest tag
23       ports:
24         - containerPort: 80
25         # X no resource limits
26         # X no health checks
27         # X no namespace
28 EOF
```



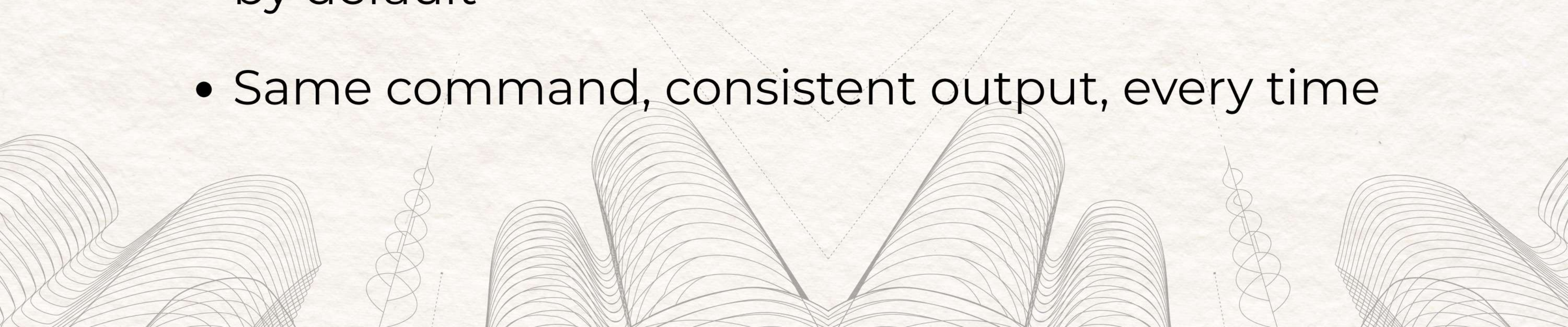
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

zsh - phase1-chaos

atulpriyasharma@Atulpriyas-MacBook-Air phase1-chaos %



Phase 1 - **Standardize**

- One script, one template, every team
 - Resource limits, health checks, labels - enforced by default
 - Same command, consistent output, every time
- 

! template.yaml × \$ deploy.sh

phase2-standardize > ! template.yaml > {} metadata > {} labels > abc app

```

1  io.k8s.api.core.v1.Service (v1@service.json) | io.k8s.api.apps.v1.Deployment (v1@deployment.json)
2  apiVersion: apps/v1
3  kind: Deployment
4  metadata:
5    name: ${APP_NAME}
6    namespace: ${NAMESPACE}
7    labels:
8      app: ${APP_NAME}
9      managed-by: platform-team
10 spec:
11   replicas: 1
12   selector:

```



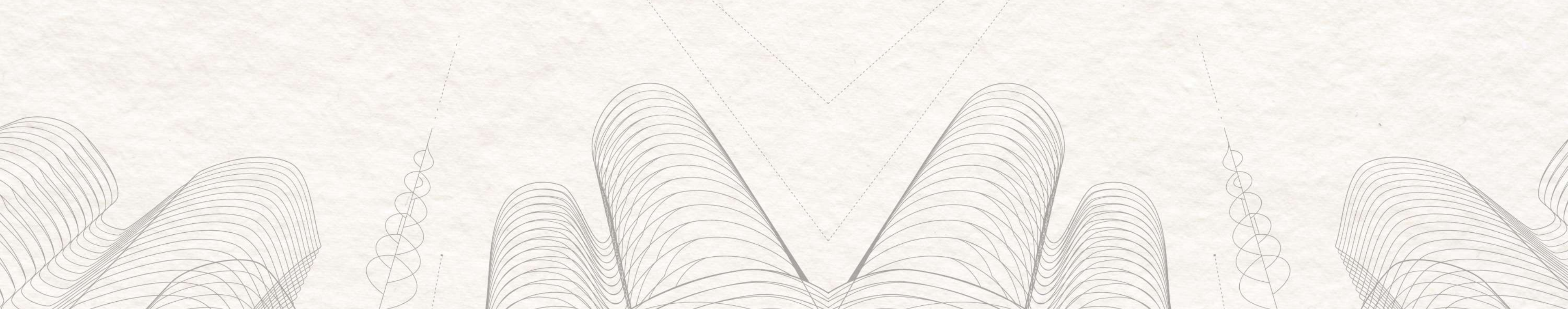
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

zsh - phase2-standardize + ▾ 🗑️ ⋮ | 🗑️ ×

atulpriyasharma@Atulpriyas-MacBook-Air phase2-standardize %



Phase 2 - **Scalable**

- Config-driven - declarative YAML, not CLI arguments
 - Bad inputs are rejected before they touch the cluster
 - Environment-aware defaults - dev, staging, prod
- 

Phase 3 - **GitOps**

- Same validation, same manifest - one change: *git push* instead of *kubectl apply*
- Argo CD watches the repo and deploys automatically
- Every deployment is a commit - full audit trail, rollback is a revert

argo v3.3.2

- Applications
- Settings
- User Info
- Documentation

Application filters

- Favorites Only

SYNC STATUS

- Unknown 0
- Synced 1
- OutOfSync 0

HEALTH STATUS

- Progressing 1
- Suspended 0
- Healthy 0
- Degraded 0
- Missing 0
- Unknown 0

LABELS

PROJECTS

CLUSTERS

Applications

+ NEW APP

calculat

Project: c

Labels:

Status: c

Reposito...

Target R...

Path: c

Destinat...

Namesp...

Created ...

Last Sync: c

SYNC

CREATE CANCEL

EDIT AS YAML

GENERAL

Application Name

Project Name

SYNC POLICY

Manual

SET DELETION FINALIZER

SYNC OPTIONS

SKIP SCHEMA VALIDATION AUTO-CREATE NAMESPACE

PRUNE LAST APPLY OUT OF SYNC ONLY

RESPECT IGNORE DIFFERENCES SERVER-SIDE APPLY

PRUNE PROPAGATION POLICY: foreground

Loading...

RETRY

SOURCE

Repository URL

GIT

argo v3.3.2

- Applications
- Settings
- User Info
- Documentation

Application filters

- Favorites Only

SYNC STATUS

- Unknown 0
- Synced 1
- OutOfSync 0

HEALTH STATUS

- Progressing 1
- Suspended 0
- Healthy 0
- Degraded 0
- Missing 0
- Unknown 0

LABELS

PROJECTS

CLUSTERS

Applications

+ NEW APP

calculat

Project: c

Labels:

Status: c

Reposito...

Target R...

Path: c

Destinat...

Namesp...

Created ...

Last Sync: c

SYNC

CREATE CANCEL

EDIT AS YAML

GENERAL

Application Name

Project Name

SYNC POLICY

Manual

SET DELETION FINALIZER

SYNC OPTIONS

SKIP SCHEMA VALIDATION AUTO-CREATE NAMESPACE

PRUNE LAST APPLY OUT OF SYNC ONLY

RESPECT IGNORE DIFFERENCES SERVER-SIDE APPLY

PRUNE PROPAGATION POLICY: foreground

Loading...

RETRY

SOURCE

Repository URL

GIT

Phase 4 - **IdP, Self Service**

- Same validation, same Git push, same ArgoCD sync - underneath
- Developer fills a form - no YAML, no terminal, no kubectl
- Any developer can deploy. The platform enforces everything.

Deploy an Application

Platform-enforced deployment via the Golden Path. No YAML. No kubectl.

`platform` `golden-path` `kubernetes`

Application Details

Fill in the fields below. The platform handles validation, manifest generation, and deployment.

1 Application Identity

Name and image tag for your application.

App Name * — lowercase, DNS-compatible

Must start with a letter. Only lowercase letters, numbers, and hyphens.

Image Tag * — no :latest

Must include a specific version tag.

2 Ownership

Which team owns this deployment?

Team *

Platform Standards

Namespace	demo
Managed by	platform-golden-path
GitOps engine	ArgoCD
Sync policy	automated
Image policy	no :latest
Health checks	enforced

What happens when you deploy

- 1 Validation** — platform rejects bad configs before anything runs
- 2 Manifest generated** — standards enforced, limits set per environment
- 3 Git push** — manifest committed to apps/ folder

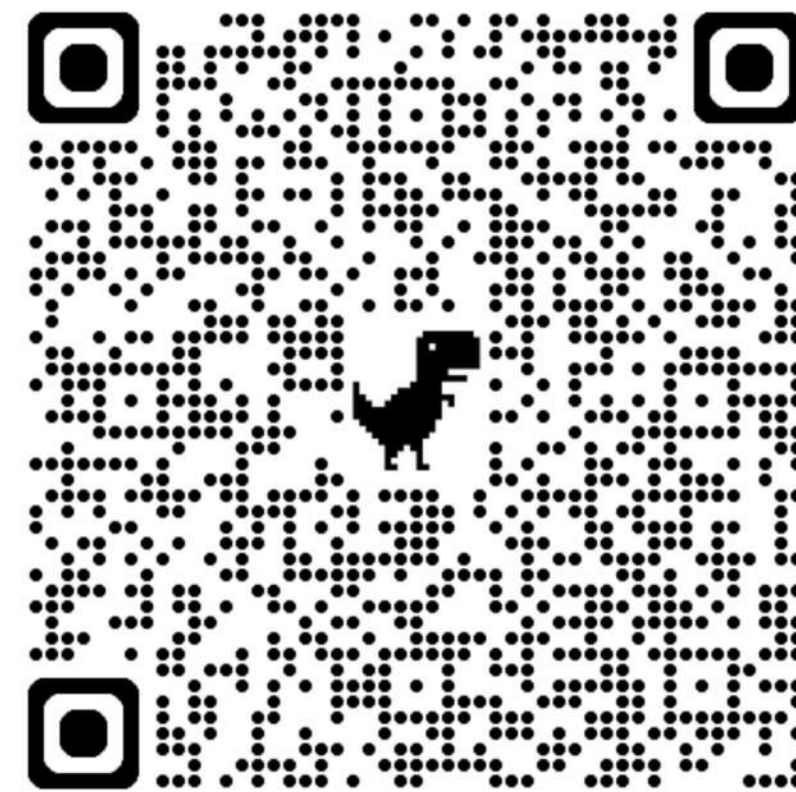
Mapping the Journey

Phase	Maturity Level	What Changed
Phase 1: Standardize	Provisional	Nothing - this is your starting point
Phase 2: Validate	Operational	Consistent tooling, standard interfaces
Phase 3: GitOps	Scalable	Self-service, guardrails, config-driven
Phase 4: IdP	Optimizing	Integrated services, audit trail, fully automated

What Next?

- Start with what you have - one script is enough for Phase 2
- Standardize before you automate
- Let the platform carry the knowledge, not the developer
- Git is your audit trail - not an afterthought
- The interface changes. The logic doesn't.

Thank You.



Read more about platforms and security