

The logo consists of a solid purple square. Inside the square, the word "zühlke" is written in a white, lowercase, sans-serif font. Below it, the tagline "empowering ideas" is written in a smaller, white, lowercase, sans-serif font.

zühlke
empowering ideas

AI-Augmented DevOps with Platform Engineering

Romano Roth | 05.09.2024 | Conf42 Platform Engineering 2024

My passion is helping companies bringing people, processes and technology together so that they can continuously deliver value to their customers



Romano Roth

Chief of DevOps & Partner

✉ romano.roth@zuehlke.com

🏠 <https://www.romanoroth.com/>

🐦 [@RomanoRoth](https://twitter.com/RomanoRoth)

🌐 [Romanoroth](https://www.linkedin.com/company/Romanoroth)

📺 [RomanoRoth](https://www.youtube.com/channel/UCRomanoRoth)

📄 [DevOps Meetup Zürich](#)

🧠 [DevOpsDays Zürich](#)

» AI is a game changer!

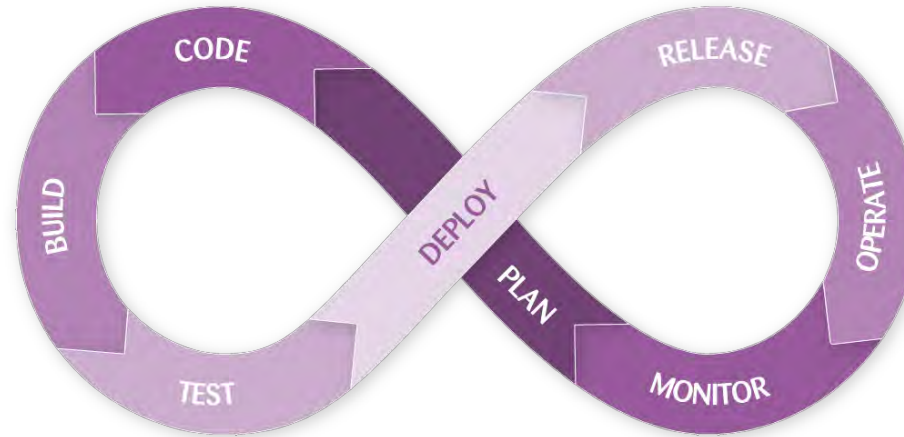
Business

» Why?

- » ■ **Faster time to market**
- **More value for money**
- **Higher quality**

Business

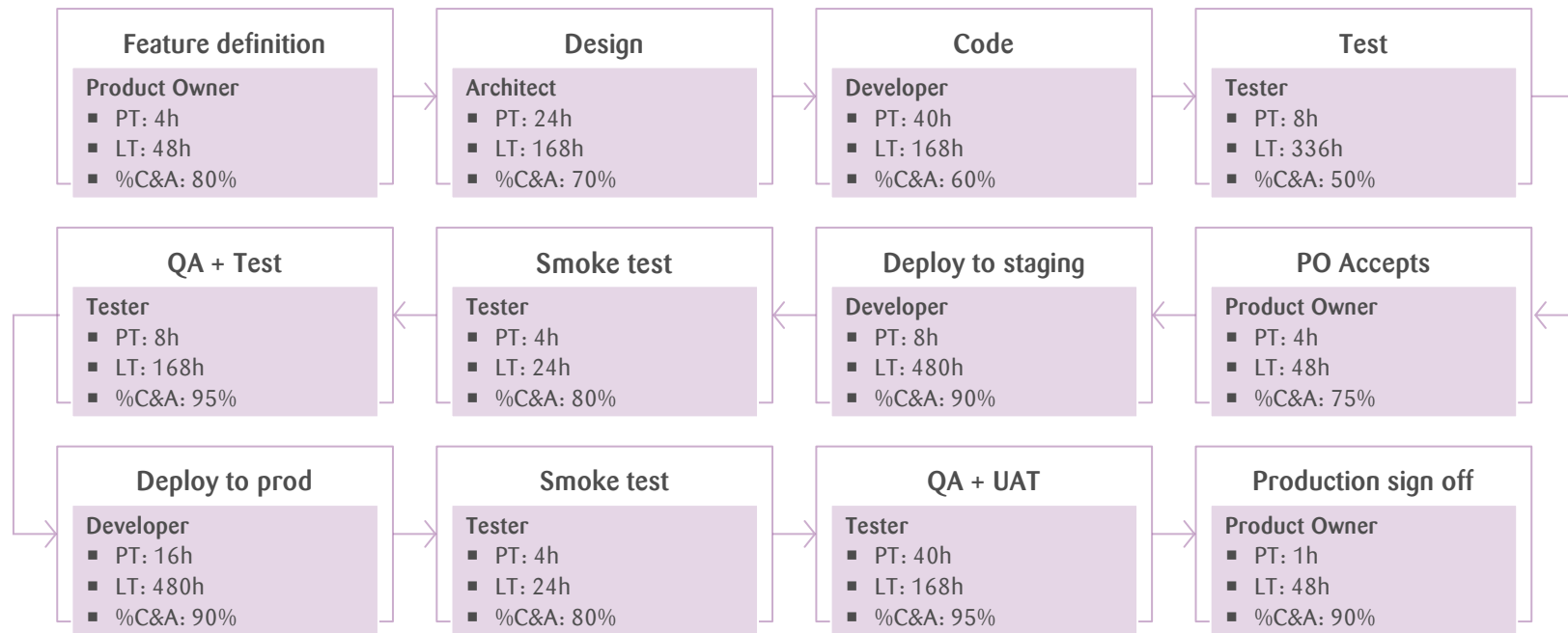
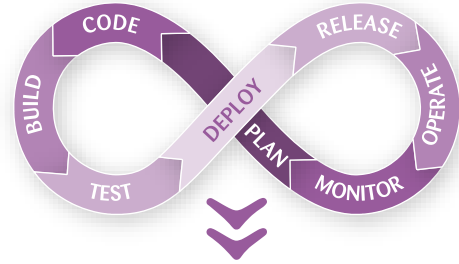
Modern Software Development is a continuous process across the value stream



Plan	Code	Build	Test	Deploy	Release	Operate	Monitor
<ul style="list-style-type: none"> ▪ Prioritize requirements ▪ Ideate ▪ Triage backlog ▪ Refine requirement ▪ Create user stories ▪ Acceptance test creation ▪ Create release plan ▪ Threat modeling 	<ul style="list-style-type: none"> ▪ Architect ▪ Design ▪ Code ▪ Search/discover ▪ Test ▪ Debug ▪ Refactor ▪ Run static code analysis ▪ Code review and merge changes 	<ul style="list-style-type: none"> ▪ Continuous Integration ▪ Software Composition Analysis ▪ License Compliance ▪ SAST ▪ Container Scanning ▪ Secret Detection ▪ Container Registry 	<ul style="list-style-type: none"> ▪ Write build pipeline code ▪ Run unit tests ▪ Run security scans ▪ Run “fitness functions” ▪ Debug and fix build errors ▪ Build artifacts ▪ Provision test environments ▪ Deploy artifacts ▪ Runtime checks ▪ E2E Test 	<ul style="list-style-type: none"> ▪ Scheduled Pipelines ▪ Environment Management ▪ Deployment automation ▪ Staging environment ▪ Continuous Delivery / Deployment ▪ Production Testing 	<ul style="list-style-type: none"> ▪ Feature management ▪ Change impact analysis ▪ Acceptance tests ▪ Production readiness tests ▪ Performance and chaos tests ▪ Test backup and recovery ▪ Deploy/verify changes ▪ Approvals 	<ul style="list-style-type: none"> ▪ Incident response ▪ Monitoring service levels ▪ Analyzing logs, metrics, traces ▪ Root cause analysis ▪ Triage alert ▪ Recover from failure ▪ Cloud operations 	<ul style="list-style-type: none"> ▪ Continuous Monitoring ▪ Full-stack telemetry ▪ Observability ▪ On-Call Schedule Mgmt. ▪ Incident Mgmt. ▪ Service desk ▪ Software Bill of Material

Value Stream Mapping

Identify bottlenecks in your development process and AI Use Cases



Total PT = 161 Hours

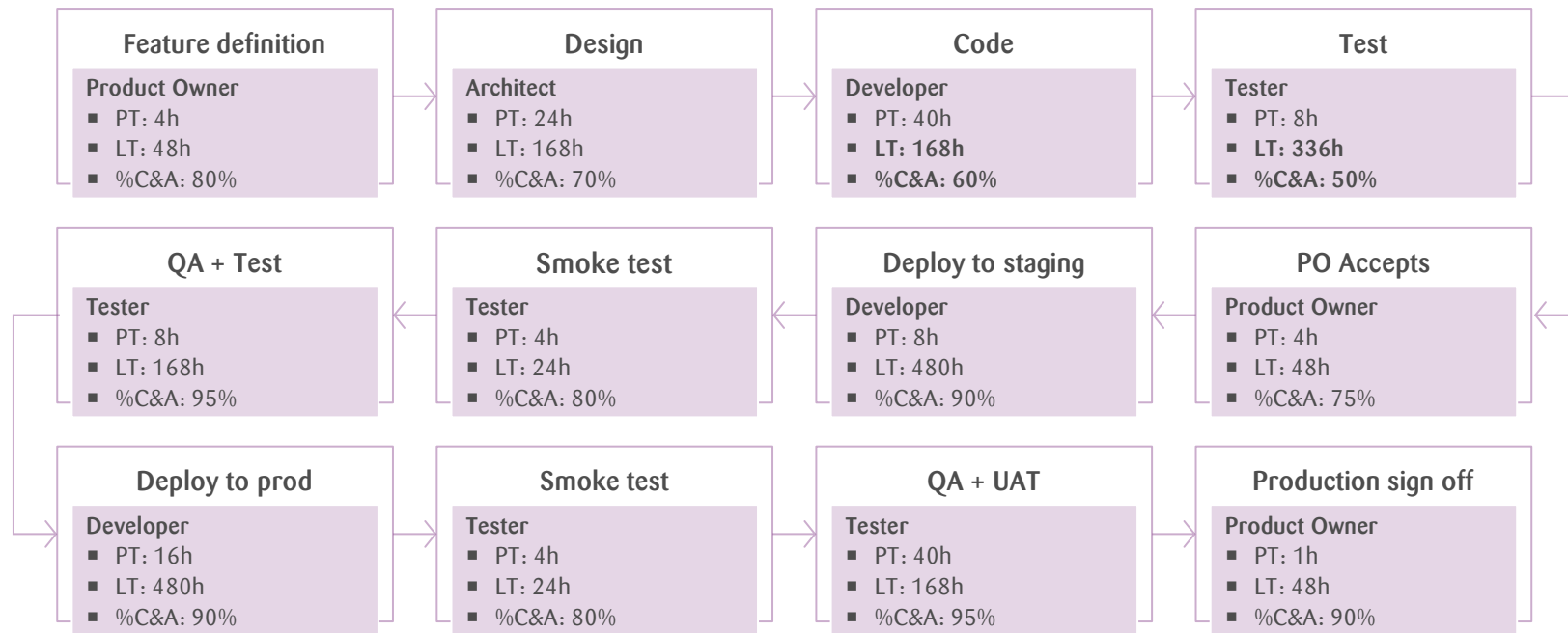
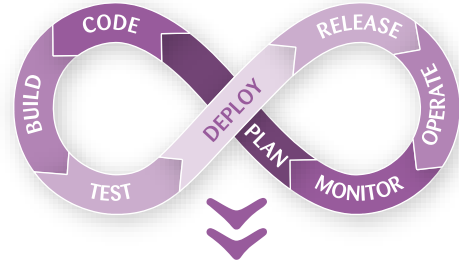
Total LT = 2160 Hours

Activity RATIO = 7%

Rolled % C& A = 5%

Value Stream Mapping

Identify bottlenecks in your development process and AI Use Cases



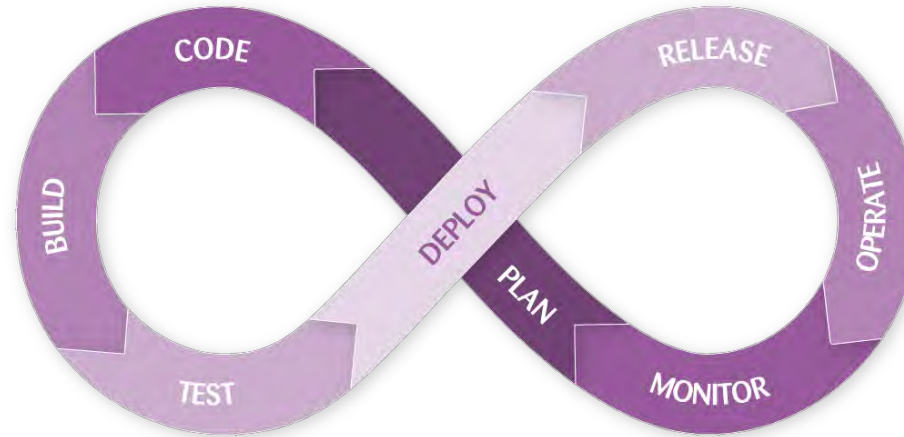
Total PT = 161 Hours

Total LT = 2160 Hours

Activity RATIO = 7%

Rolled % C& A = 5%

Iteratively Resolve the Greatest Constraint your Value Stream with AI



Plan	Code	Build	Test	Deploy	Release	Operate	Monitor
<ul style="list-style-type: none"> ▪ Prioritize requirements ▪ Ideate ▪ Triage backlog ▪ Refine requirement ▪ Create user stories ▪ Acceptance test creation ▪ Create release plan ▪ Threat modeling 	<ul style="list-style-type: none"> ▪ Architect ▪ Design ▪ Code ▪ Search/discover ▪ Test ▪ Debug ▪ Refactor ▪ Run static code analysis ▪ Code review and merge changes 	<ul style="list-style-type: none"> ▪ Continuous Integration ▪ Software Composition Analysis ▪ License Compliance ▪ SAST ▪ Container Scanning ▪ Secret Detection ▪ Container Registry 	<ul style="list-style-type: none"> ▪ Write build pipeline code ▪ Run unit tests ▪ Run security scans ▪ Run “fitness functions” ▪ Debug and fix build errors ▪ Build artifacts ▪ Provision test environments ▪ Deploy artifacts ▪ Runtime checks ▪ E2E Test 	<ul style="list-style-type: none"> ▪ Scheduled Pipelines ▪ Environment Management ▪ Deployment automation ▪ Staging environment ▪ Continuous Delivery / Deployment ▪ Production Testing 	<ul style="list-style-type: none"> ▪ Feature management ▪ Change impact analysis ▪ Acceptance tests ▪ Production readiness tests ▪ Performance and chaos tests ▪ Test backup and recovery ▪ Deploy/verify changes ▪ Approvals 	<ul style="list-style-type: none"> ▪ Incident response ▪ Monitoring service levels ▪ Analyzing logs, metrics, traces ▪ Root cause analysis ▪ Triage alert ▪ Recover from failure ▪ Cloud operations 	<ul style="list-style-type: none"> ▪ Continuous Monitoring ▪ Full-stack telemetry ▪ Observability ▪ On-Call Schedule Mgmt. ▪ Incident Mgmt. ▪ Service desk ▪ Software Bill of Material

AI-Augmented DevOps use cases to optimize the Value Stream

- » AI can analyze historical project data to predict risks, resource needs, and delivery timelines, optimizing sprint planning
- » Summarize minutes of meeting
- » Automated generation of threat models
- » Prioritize and label issues based on descriptions

- » Generate, refactor, debug and explain code
- » Generate unit tests from code and language
- » Convert UX designs to code
- » Generate comments and documentation
- » Explain the security vulnerability

- » Auto Remediation of security vulnerabilities
- » Intelligent test selection to reduce CI build times
- » Summarize PR changes and review comments
- » Synthetic test data generation

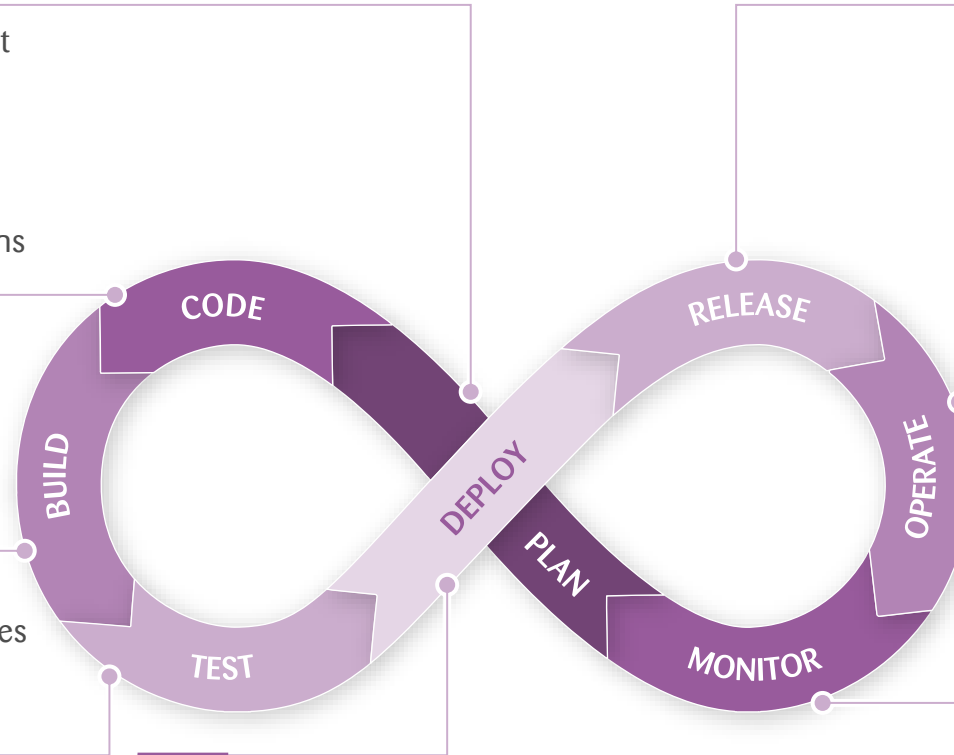
- » Impact analysis
- » Intelligent test selection
- » Prioritize test execution based on risk
- » Analyze and identify gaps in test coverage
- » Predict test failures using historical data
- » Auto-remediate flaky or unstable tests

- » AI predicts the impact of deployments
- » Monitors deployment health and auto triggers rollbacks
- » Suggesting the best time to deploy based on historical data
- » Automation of rollback strategies
- » Orchestrate blue-green deployments
- » Selects the optimal subset of users for canary releases

- » Continuous release verification
- » Change impact analysis
- » Auto enable/disable feature flags based on health checks
- » Predict change failures by assessing release readiness

- » Detect and fix configuration drift
- » Takes proactive actions such as auto-scaling, self-healing applications, and dynamic resource allocation.
- » Predicts operational incidents

- » Pattern recognition, anomaly detection, event correlation, root-cause analysis, self-healing systems
- » Intelligent workload optimization (optimize cost, reliability and sustainability, and achieve trade-offs)
- » Predictive and prescriptive analytics



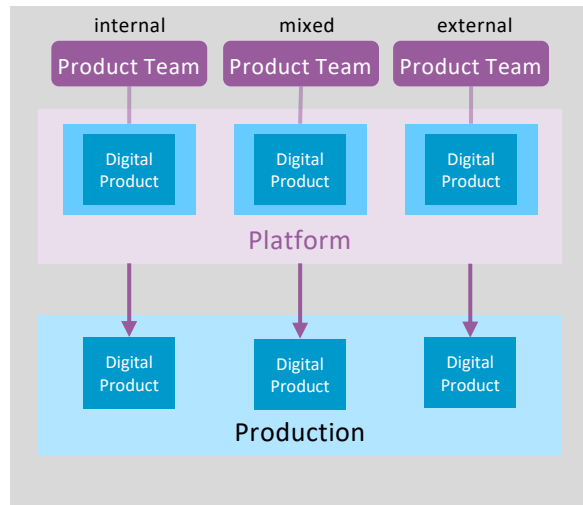
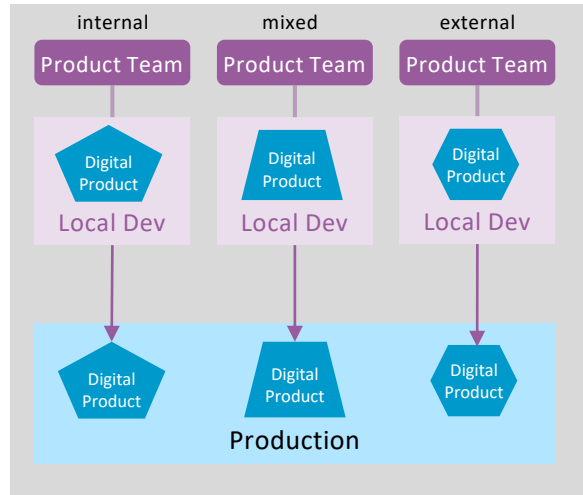
» But, to do that we need the right foundation.

Industrialization of Software Engineering

Distributed and heterogeneous Development Environments
internal and with partners

from
↓
to

Platform
internal and with partners



Source: CNCF

Huge Tool Landscape

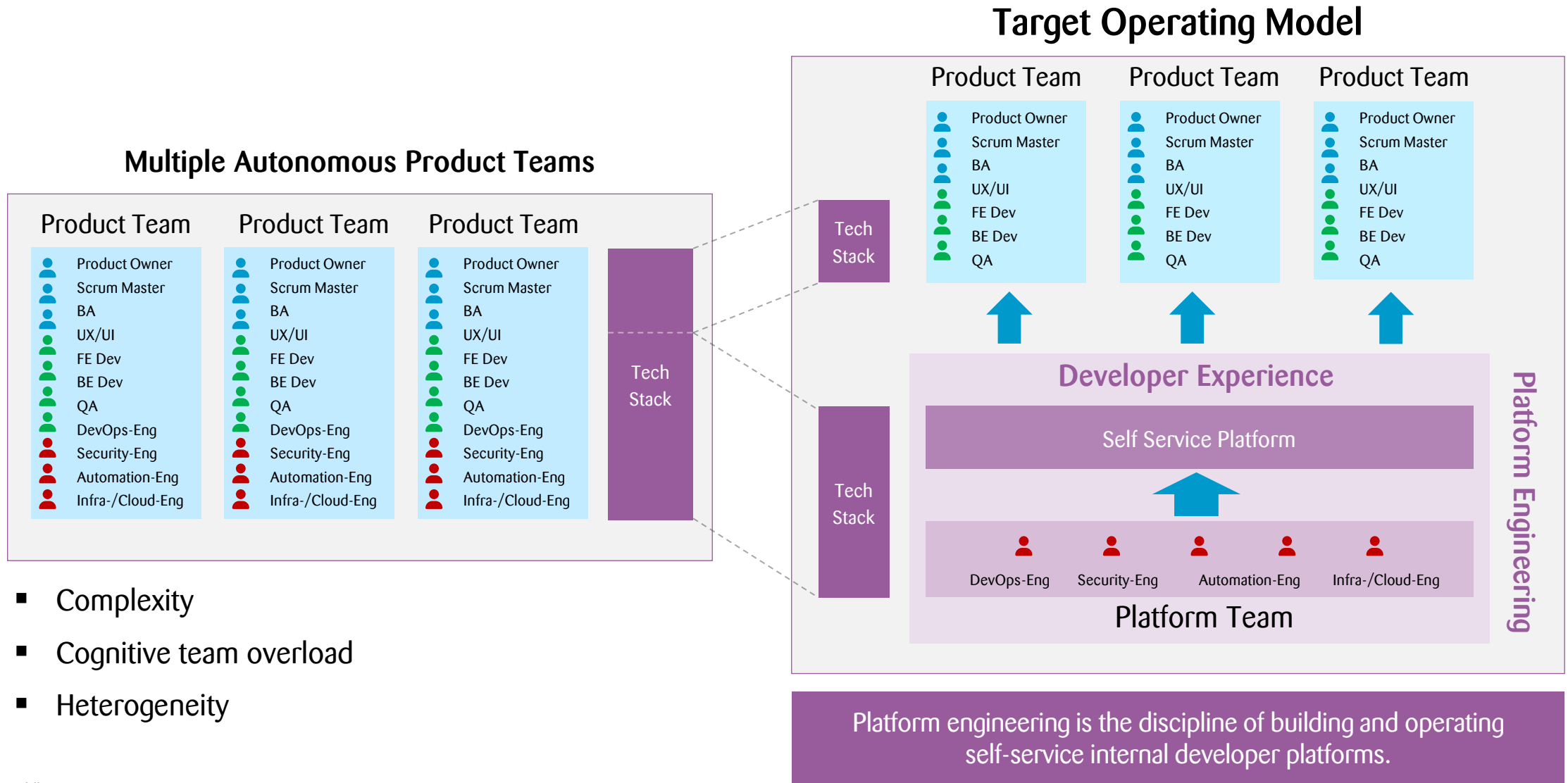
from
↓
to

Standards of preselected Services and Products (paved paths)

- Services**
- Kubernetes (AKS)
 - Crossplane
 - Kafka
 - Ingress NGINX
 - MQ: RabbitMQ
 - API Gateway: Tyk, Gloo
 - DB: MS SQL, MySQL, Oracle, PostgreSQL, MongoDB, MariaDB, Redis, ElasticSearch, Neo4j
 - S3: MinIO
 - IDP: Azure Entra, Keycloak

- Products**
- GitLab
 - JFrog
 - ArgoCD
 - Hashicorp Vault
 - Prometheus
 - Grafana
 - Open Telemetry
 - Trivy
 - Kyverno
 - Tempo
 - Calico
 - Aqua
 - SonarQube
 - Jenkins
 - Cypress
 - Playwright

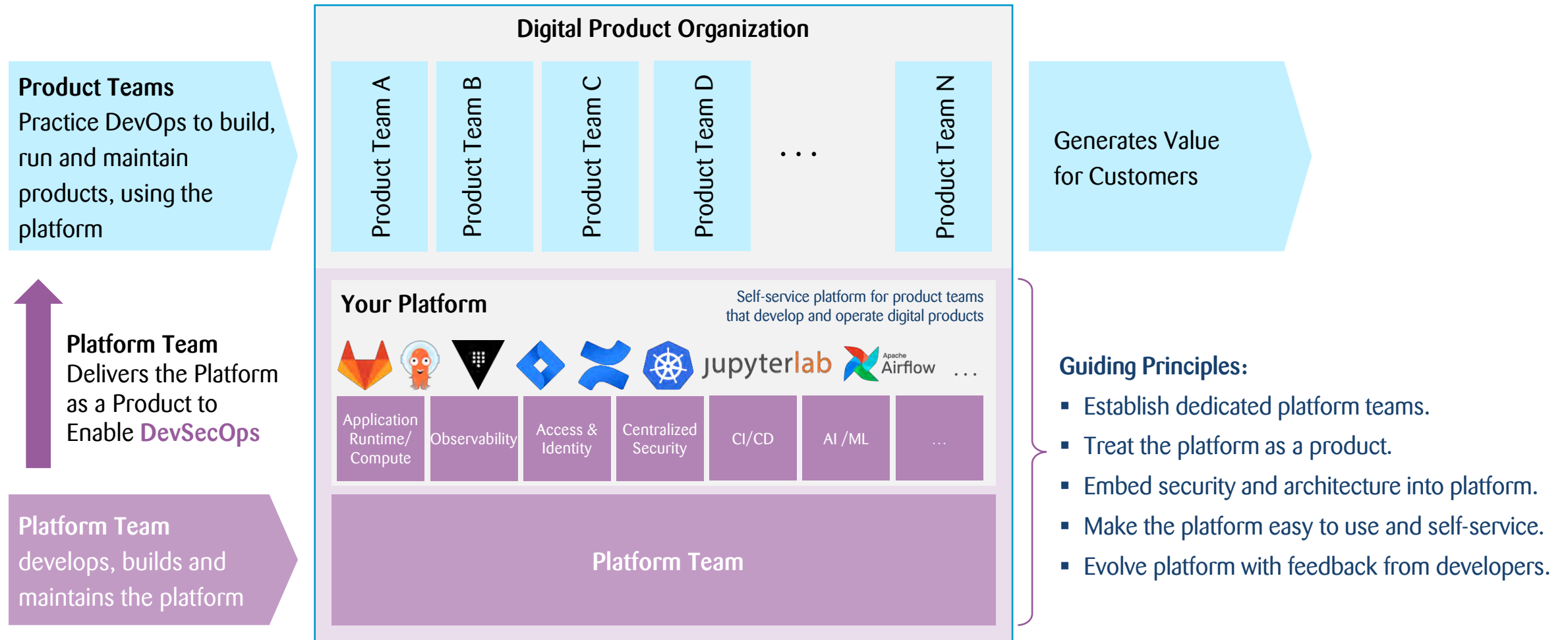
Target Operating Model with Platform Engineering and Platform Team



- Complexity
- Cognitive team overload
- Heterogeneity

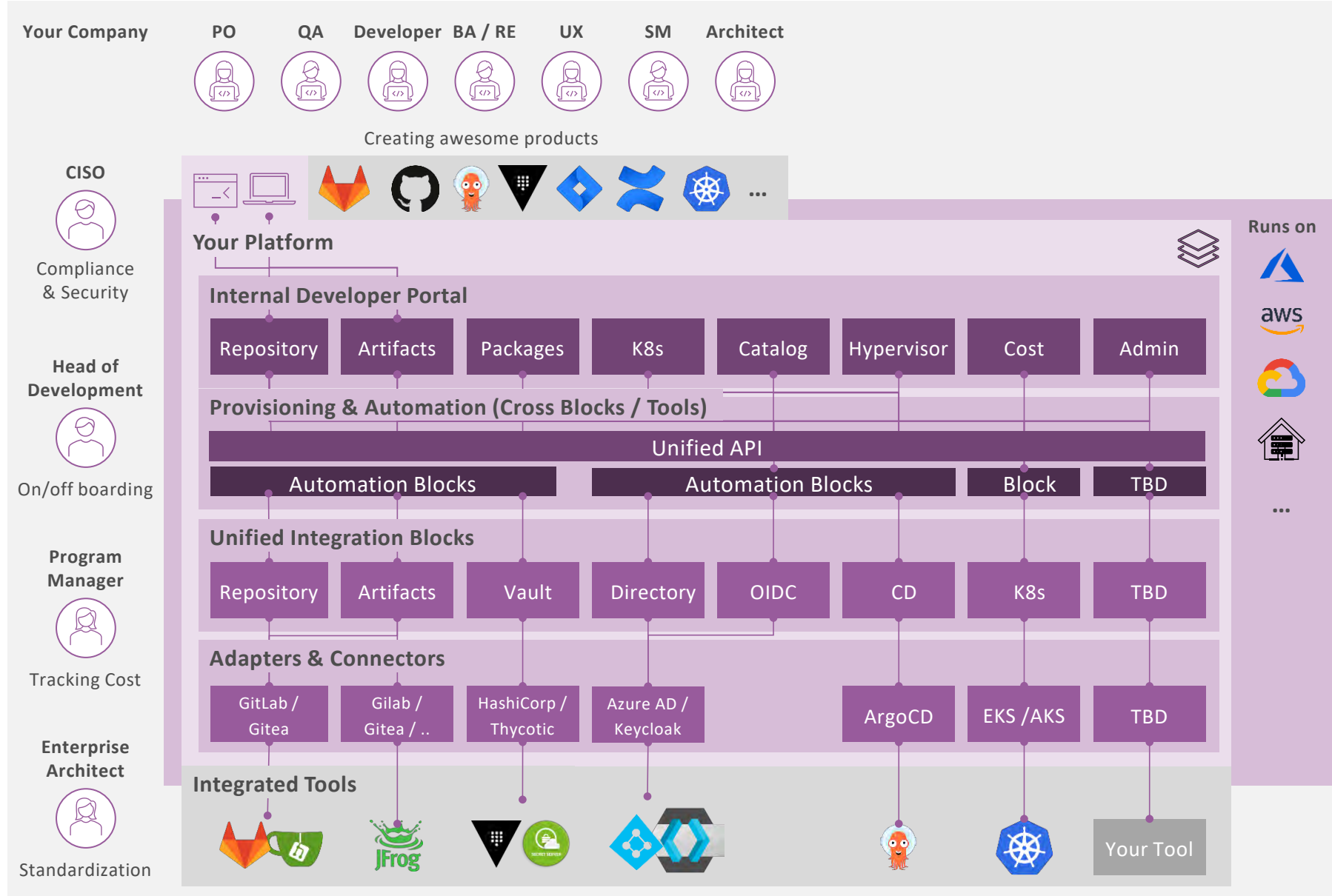
Digital Product Organization

Platform Team provides a Developer Platform to Product Teams, which build and run Digital Products using self services



High Level Architecture of a Platform

Public

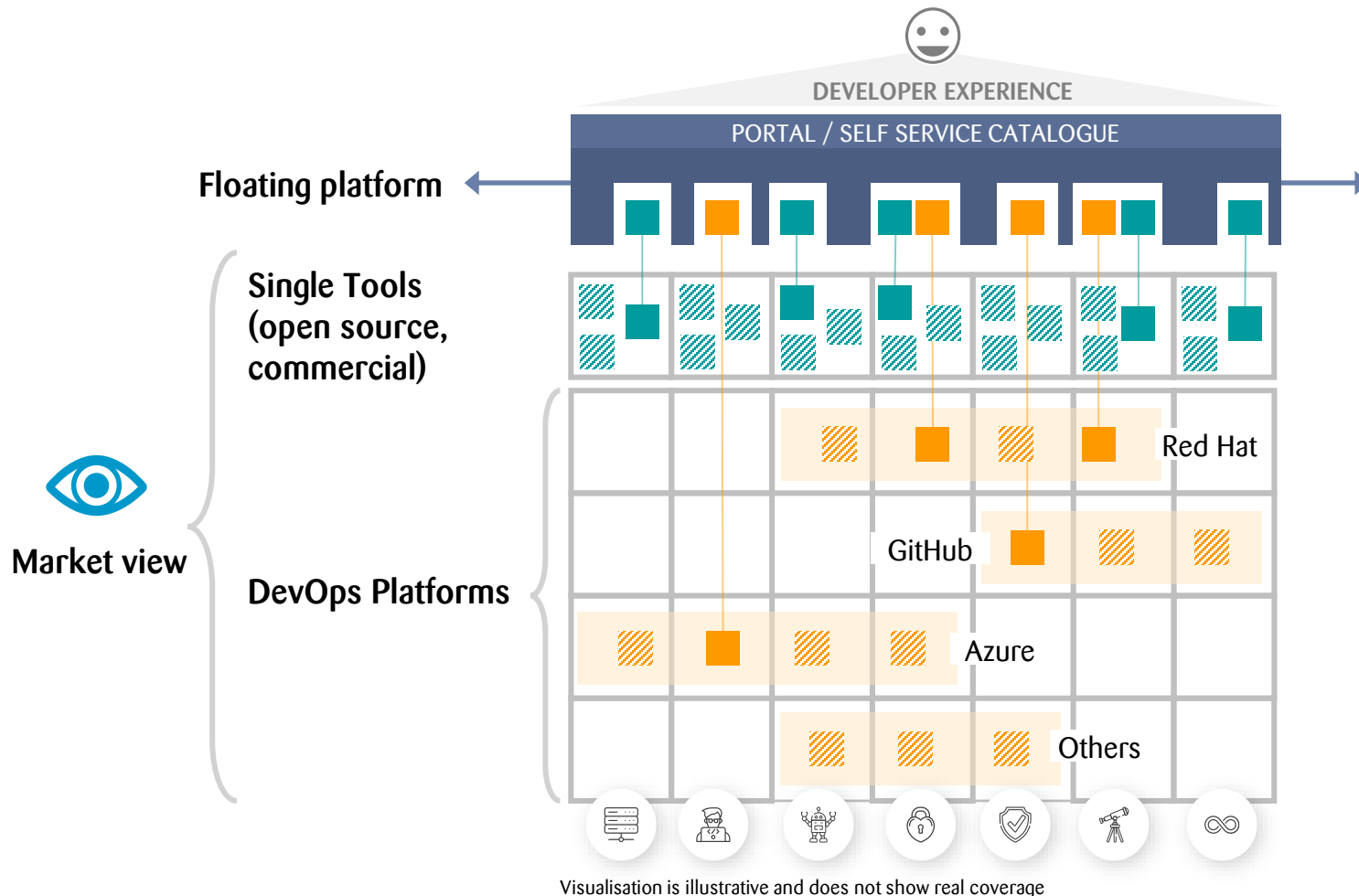


- Application Runtime/Compute
- Developer Experience
- AI
- Automatic DevSecOps
- Access & Identity
- GitOps
- Observability
- Centralized Security

- Faster Time to market
- More value for money
- Better quality

Floating platform

A Platform integrates best-in-class tool chains for software product development in a highly efficient way

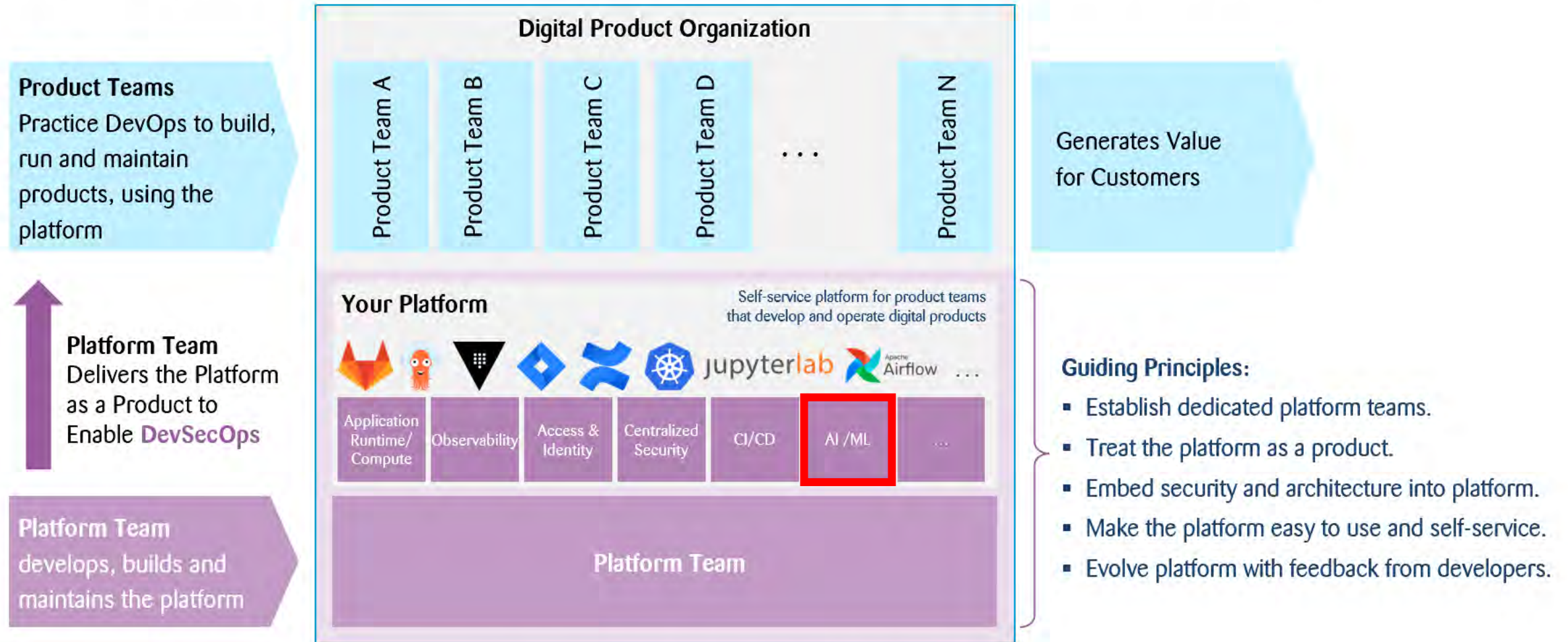


Characteristics of a Platform

- Standing platform team and community able to navigate a rapidly changing tool environment and select the best tools
- Highly efficient, standards-based integration of commercial and open-source tools into the platform
- Covering the entire range of functions across all relevant subject areas
- Excellent developer experience through easy use via portal and self-service catalog
- Supports tool life-cycle management via a managed service catalog
- Utilizes the innovative power and speed of the entire tool and platform provider ecosystem, without lock-in risks

Digital Product Organization

Platform Team provides a Developer Platform to Product Teams, which build and run Digital Products using self services



Platform Teams Offer GenAI Capabilities “as a Service” and also use them



Product Development Teams

Platform Interfaces



Developer Portal



Conversational



CLI



API

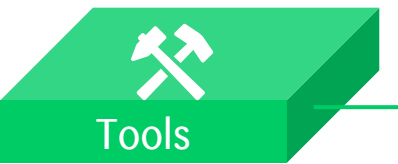
Common generative AI Platform capabilities

Reusable GenAI components (eg. Summarization, content synthesis and analysis, prediction)



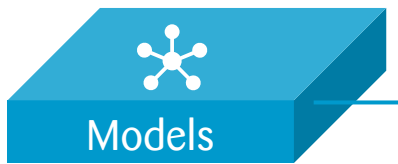
Apps

Chatbots	AI coding assistants	Generative design tools
Synthetic data tools	Knowledge management	Productivity assistants



Tools

Prompt Engineering	RAG and fine-tuning tools	GenAI application frameworks
Vector databases	Model life cycle management (DataOps, MLOps, ModelOps)	AI TRiSM (AI trust risk and security management) tools



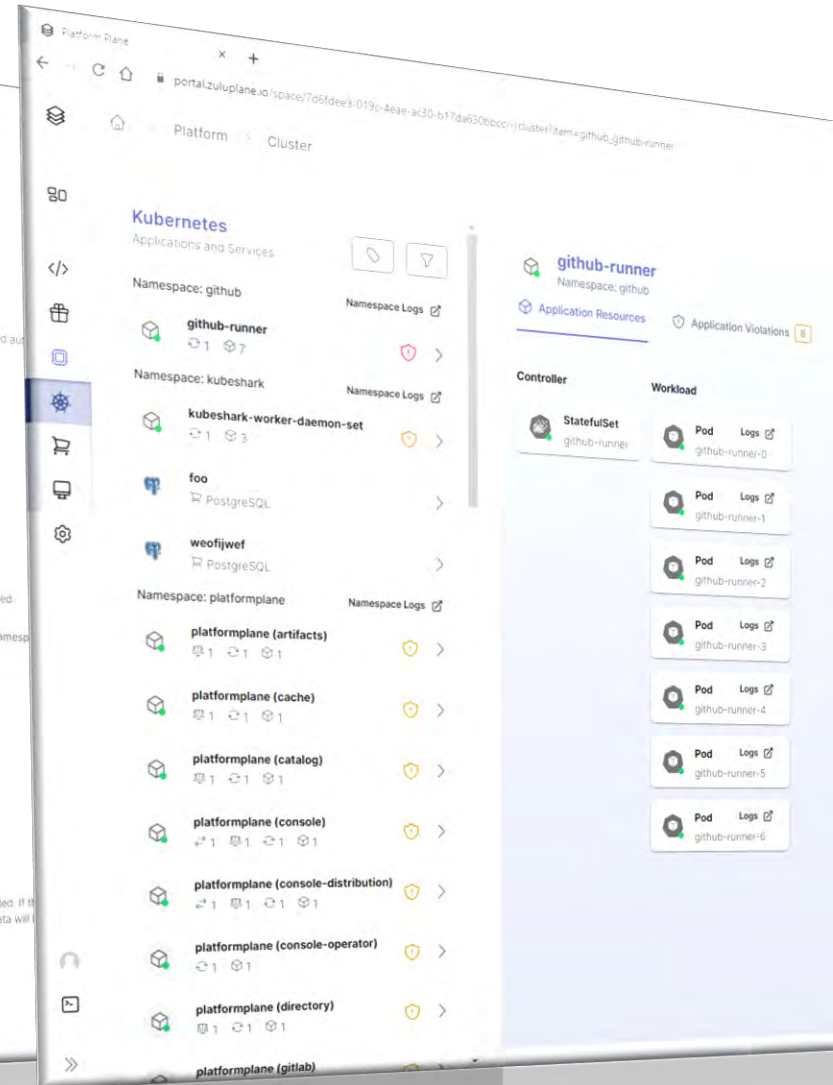
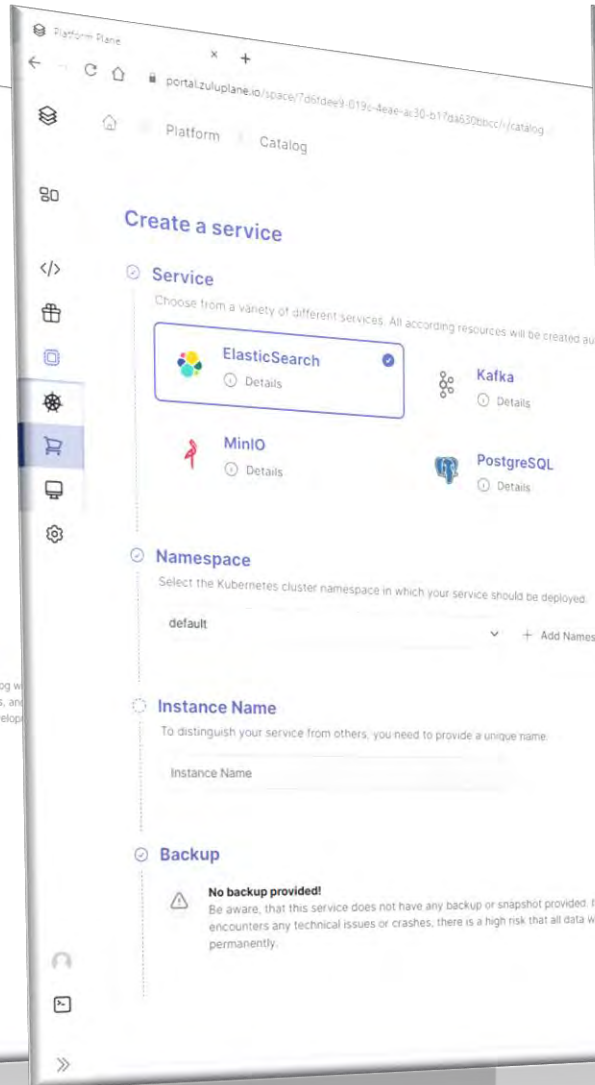
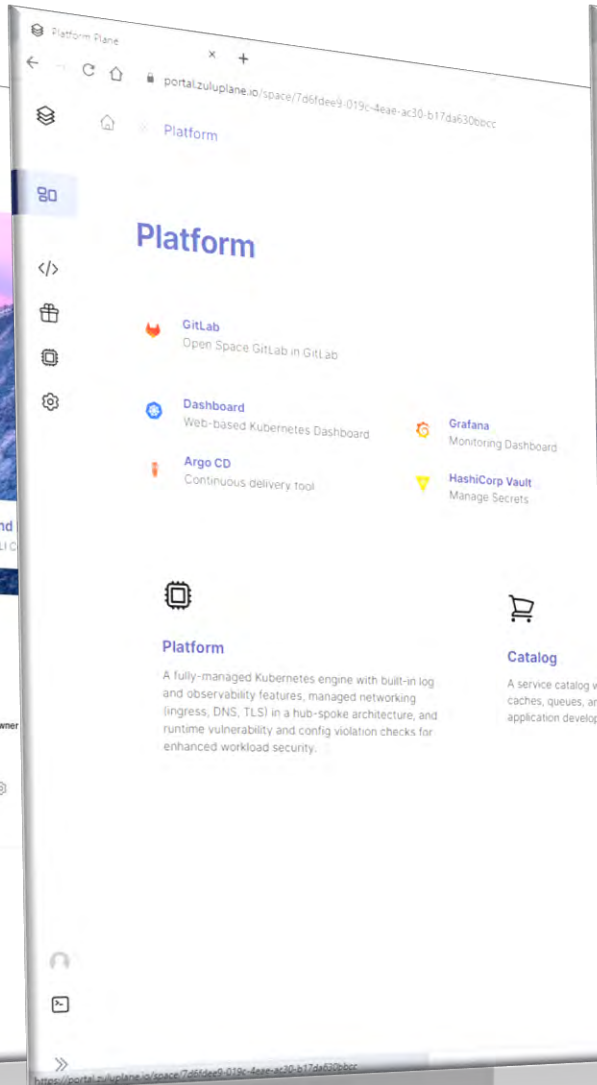
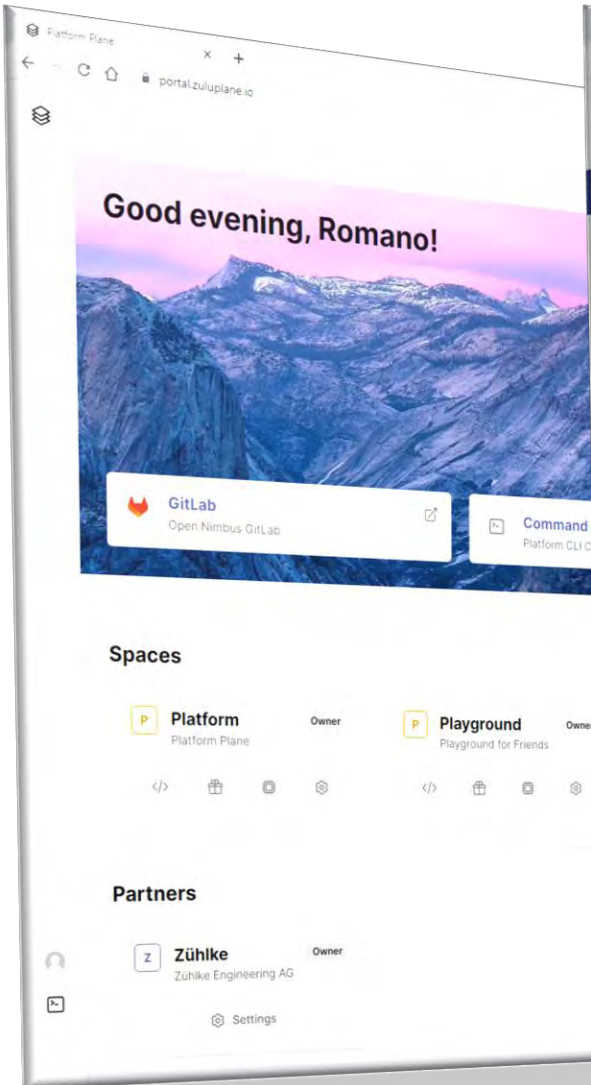
Models

Model hubs (app store for models)	Visibility into model usage and cost	Sandbox environments for experimentation
Open-Source Models	Enterprise-specific models	Domain-specific models



GenAI infrastructure components – compute, network, storage

Demo



Summary

AI-Augmented DevOps with Platform Engineering

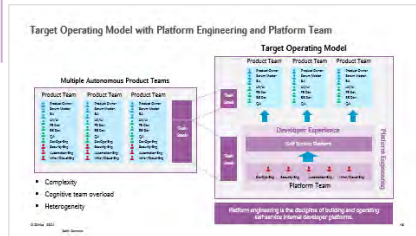
00

Value Stream Mapping



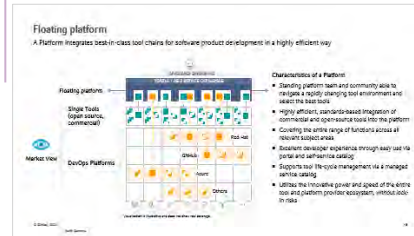
01

TOM with Platform Engineering



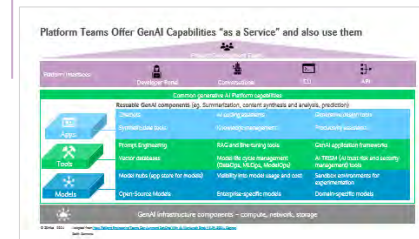
02

Build a floating Platform



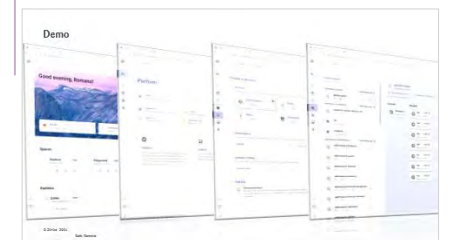
03

AI Capabilities as a service



04

Everyone uses these AI Services



We are entering the age of industrialization of Software Development

Platform Teams builds your Platform which enables teams to do AI-Augmented DevOps.

This is the future of **DevOps**: faster, smarter, and more resilient software development through AI-driven innovation.

