Breaking down the principles of Platform Engineering

How to empower developer experience

Cedric Clyburn
Senior Developer Advocate
Red Hat

Valentina Rodriguez Sosa
Principal Architect
Red Hat

CONF42

Today's Schedule

- Developer Persona & Platform Engineering

- Platform as a Product

- Internal Developer Portals

  - Demo (with AI + Virtualization)

- Best Practices and Reference Architectures

CONF42

Let's build an app!

I'll just grab my editor and compiler…

...and Podman
(ok, probably Docker)
...and a container registry
...and a place to run containers

...and CD to deploy it
...and a LOT of YAML
...and a place to store the code (and the YAML)

CONF42

Developers are under pressure: And their productivity is slowing down.

Organizations are looking for a platform to reduce toil and increase productivity.

39%
Increasing workload / demand from other teams lead to increased overhead.[1]

76%
More than three-quarters of organizations say the **cognitive load is so high that it is a source low productivity.**[1]

37%
Adapting to the pressures of digital transformation is third biggest challenge.[1]

35%
Learning skills to adapt to new technologies and approaches [1]

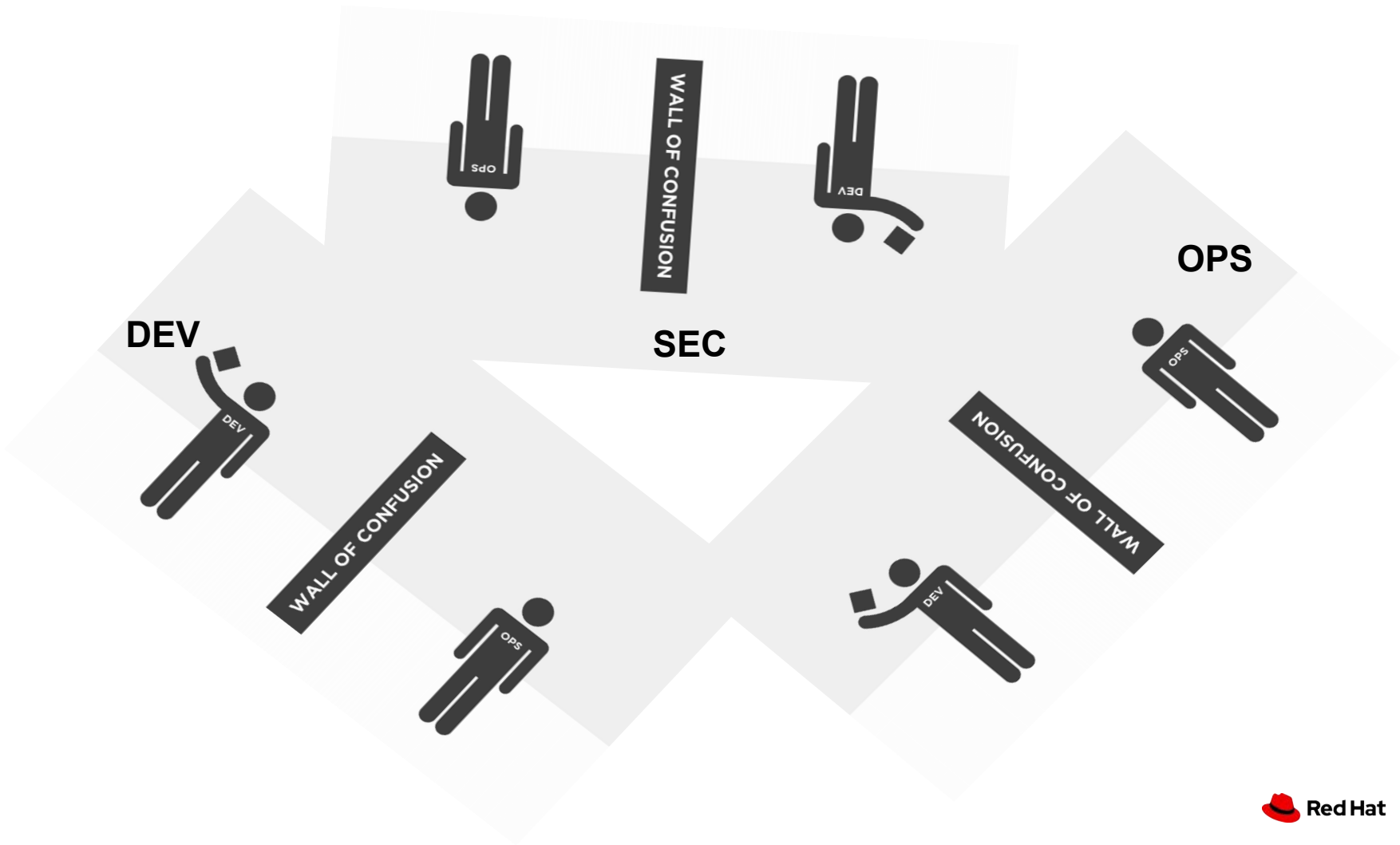[1] Salesforce productivity survey
[2] Stackoverflow happiness survey

CØNF42

DEV OPS

WALL OF CONFUSION

OPS

DEV

SEC

WALL OF CONFUSION

WALL OF CONFUSION

WALL OF CONFUSION

Red Hat

An alternative platform approach

Traditional delivery	Product operating model
Maintaining a stable state	Continuous improvement
Requirements defined up front	Continuous review of business and user needs
Problems are remediated as a response	Problems are pre-emptively engineered out
Requests are handled via tickets	Requests are managed automatically and on-demand
Slow to adapt to changing needs	Greater agility and incremental improvement aligned with user needs

Benefits of platform with a product mindset
Faster to value, reduced waste, lower risk, greater agility

CONF42

Observability and Monitoring

Compliance & Security

App Platform

Platform Engineers

Full Stack Developer

Backend Developer

Frontend Developer

Enterprise App Devs

Developer Tools

Enhance Application Capabilities

Data Eng
Data Scientist

Designs the bespoke platform
Offers a **self-service portal** to access the platform
Evangelizes the platform
Constantly **improves the platform**

Python Anaconda Jupyter Notebook

CONF42

Platform Engineering
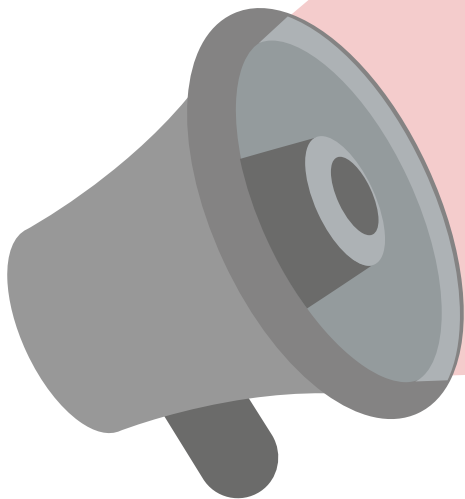
Gartner named Platform Engineering as one of the **top 10** strategic technology trends for 2024[1]

IDC predicts that by 2027, **70%** of enterprises will adopt a **composable, internal-facing app platform empowering developers** to source and personalize discovery of development-related products, services, and infrastructure"[2]

CONF42

Platform-as-a-Product

What is a **Platform**?

"A **digital platform** is a foundation of self-service APIs, tools, services, knowledge and support which are arranged as a compelling internal product. "

Evan Bottcher - 2018

CONF42

"A **digital platform** is a foundation of self-service APIs, tools, services, knowledge and support which are arranged as a compelling internal product. "

Evan Bottcher - 2018

CONF42
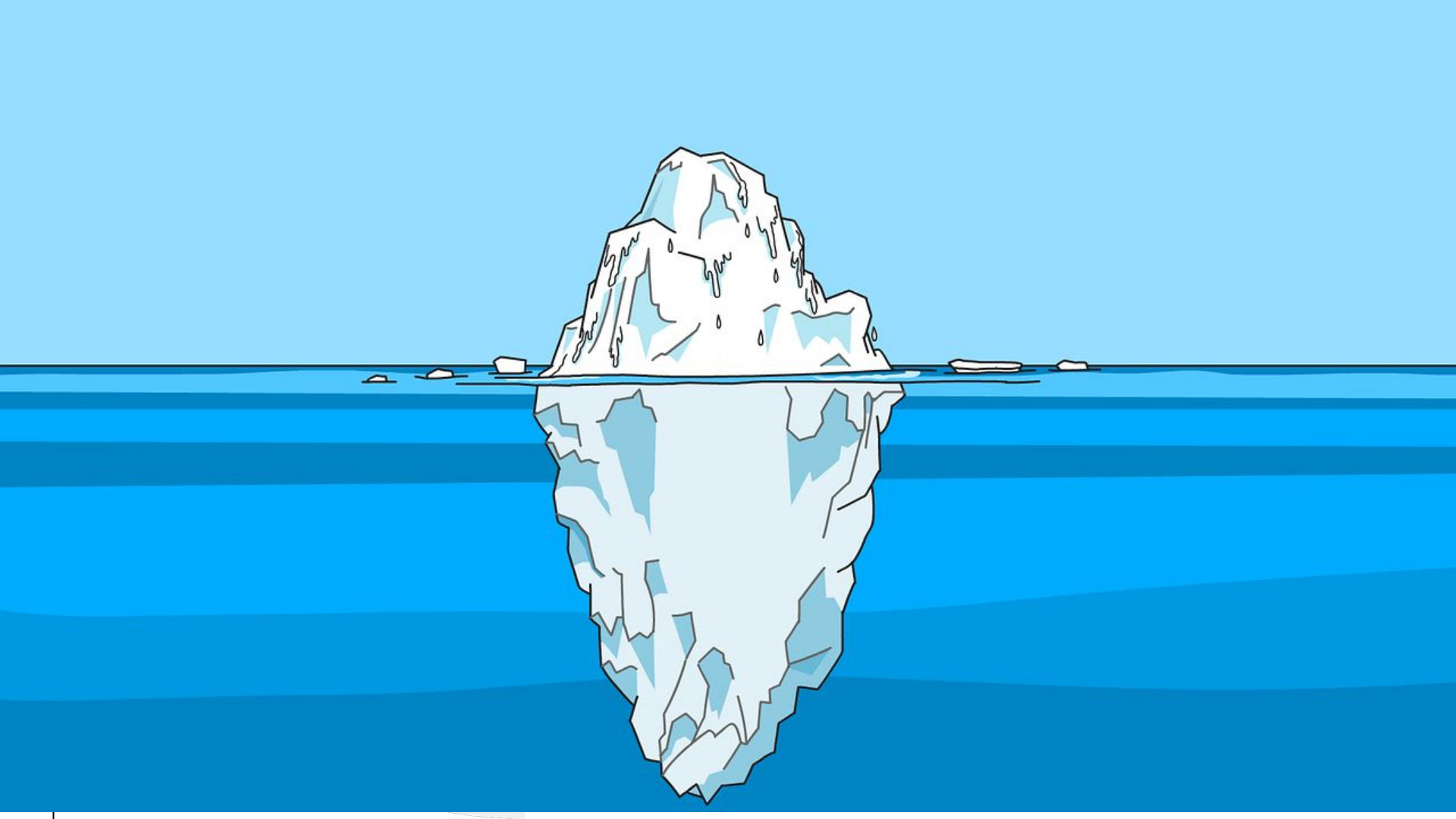
Internal Developer Platform (Backstage)

Software Templates (Golden Path Templates)

DevX: Collaboration, Tutorials, Videos, Slack

Developer Environments

DBaaS, Kafka-as-a-Service, Cache-as-a-Service, SSO-as-a-Service, etc

CLI (kubectl, helm, git, oc), API

Observability (Ops/SRE view, Dev view, AppOps view, PE view)

Continuous Delivery

Artifact Storage

Security: Runtime scanning, build-time scanners

Continuous Integration Engine & Task Runners

Artifact Builder, Image Builder

Source Control

AuthN/AuthZ/RBAC

Compute/Network/Storage, Namespace-as-a-Service/Cluster-as-a-Service, Configuration Management

"A **digital platform** is a foundation of self-service APIs, tools, services, knowledge and support which are arranged as a compelling internal product. "

Evan Bottcher - 2018

CONF42

Pillars of an Internal Development Platform

Onboarding

Streamlined process for developers to get their team, application, or component **set up on the platform**

Includes access management, documentation, and guidance

Code Time

Optimizes the developer's local environment and coding workflow

Focuses on quick workstation setup and a **fast, reliable inner loop** (code, build, test, debug)

Build Time

Encompasses the CI/CD process for building, testing, and packaging code

Ensures **consistent and reliable promotion of code changes** to various environments

Run Time

Provides a **stable, scalable, and secure foundation** for running applications in production

Includes **monitoring, logging,** and troubleshooting capabilities

CONF42

Backstage: A open platform for building developer portals

CLOUD NATIVE
COMPUTING FOUNDATION

Allowing developers to focus on what they want to ... coding, rather than navigating to all the different tools

Resulting in lowering the cognitive load and unlocking developer productivity

Developer portals function as access points to the IDP and provide a user interface for developers to discover the platform's capabilities.

Happy developers make happy code!

CONF42

Backstage Adopters

CONF42

Backstage Architecture

	Centralized Software Catalog	Keep track of application ownership and metadata for all the software in your organization (services, websites, libraries, data pipelines, etc.)
	Plugins	Extend functionality and provide customization
	Software Templates (Golden Path Templates)	Quickly spin up new projects and standardize your tooling with your organization's best practices
	Tech Docs	Easily create, maintain, and find technical documentation using a "docs as code" approach
	Search	Find the right information across your entire Backstage ecosystem

CONF42

Backstage Plugins

Plugins add **additional functionality** to make your Backstage useful for your organization. Plugins can be specific to a company or open sourced and reusable.

190+ community plugins:

- ▸ SCM
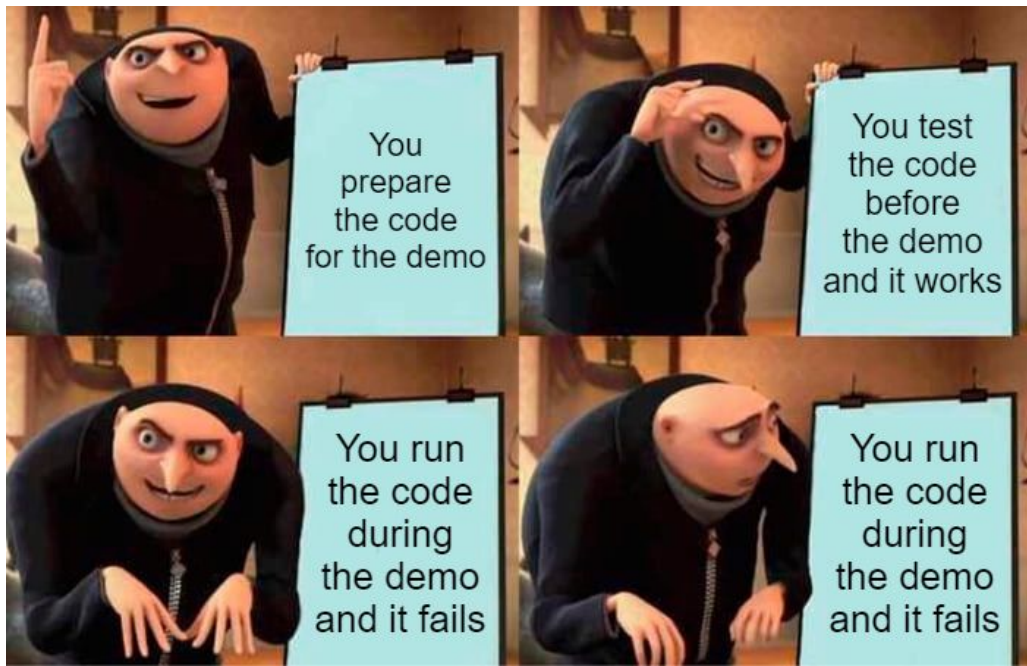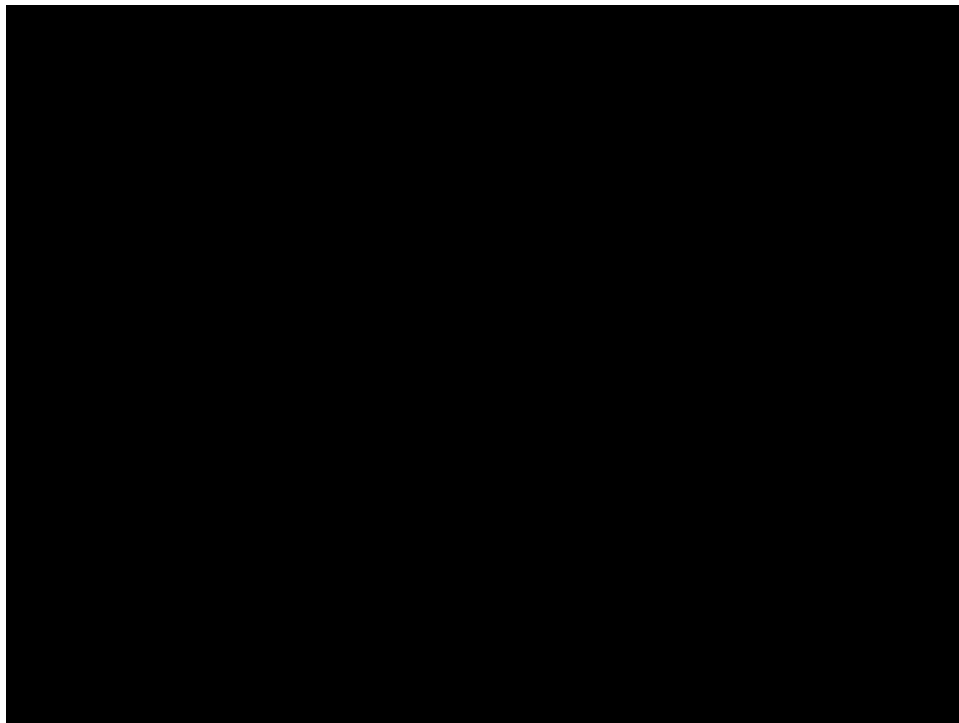- ▸ CI/CD
- ▸ Monitoring
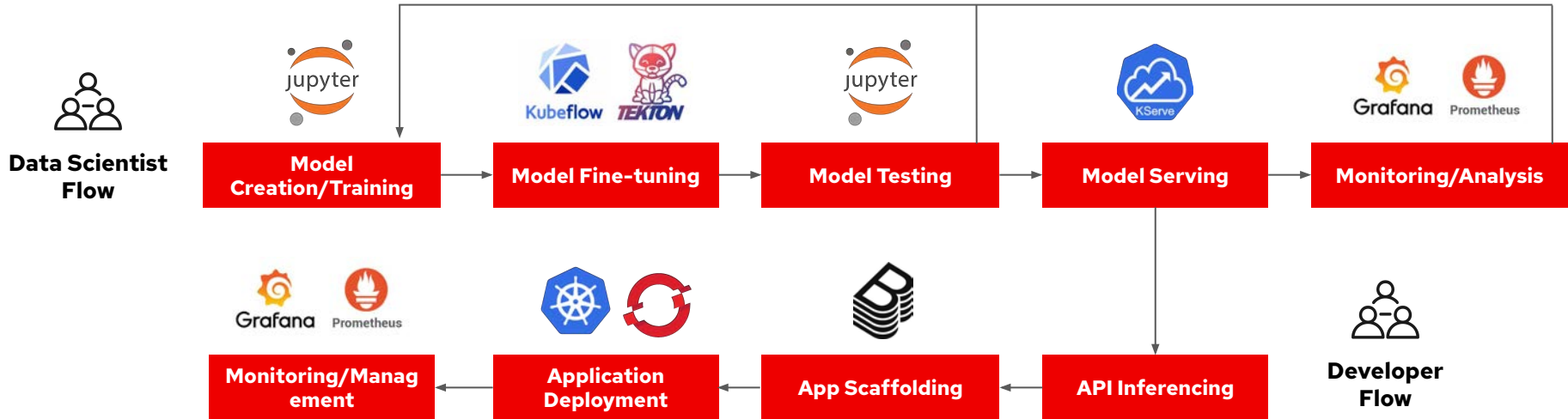- ▸ Issue tracking
- ▸ Code quality

Backstage Intro & Overview

Templates & Gen AI App Dev

Reference Architecture

AI Model to Production Application Diagram

Keycloak
Single sign-on with identity access management

Backstage
Developer portal with golden path

Developer

Sigstore
Securely sign and verify software artifacts

KubeLinter
Static analysis tool for Kubernetes YAML files

Source

GUAC
Aggregate software security metadata for actionable insights

Open Policy Agent
Policy-based controls as-code in cloud native dev

KONFLUX

Tekton Chains & Konflux
Abstract underlying CI/CD tools & processes

Build

Dependencies

StackRox
Build, deploy and runtime container security with risk analysis and vulnerability scanning

ArgoCD
Monitor actual to declared state for configuration drift

Clair
Visibility into security status of container images

Package

Project Quay
Store and protect data, with image vulnerability scan

Falco
Streaming detection at runtime

Consumer

CONF42

Best Practices

Shift Left Security

Trust is critical in platform engineering, from automated composition to signing and attestation

Find Your Personas

Find your champions to bring adoption

Start Small

Start with TechDocs

Evolve into Software Templates

Keep One Portal

Ensure consistency across your organization

CONF42

Thank you!

Connect Via LinkedIn

Slides

red.ht/platform-eng-slides

Solution Patterns

solutionpatterns.io

& be sure visit us at the Red Hat booth!

CONF42

Resources!

Sign up at developers.redhat.com

Find out more about Red Hat's projects and products, and what it offers devs/ops

Learn more about [Platform Engineering](https://...)

Try the [Developer Sandbox for OpenShift](https://...)

Deploy and test applications for free with OpenShift, Red Hat's enterprise distribution of Kubernetes

Check out [Red Hat Developer Hub](https://...)

Get hands on with Developer Hub, an enterprise-grade level IDP created from contributions to Backstage

CONF42