

Accelerate Your Modern Apps To Production

Robert Jensen
Lead Systems Engineer @Vmware

 @rhjensen / jensenr@vmware.com

9/5/2023



Why You Want To Accelerate Your Apps To Production ?

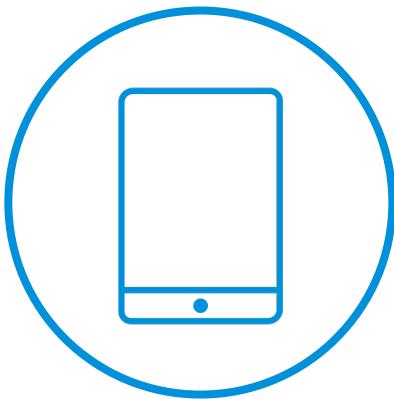


Apps are Everywhere

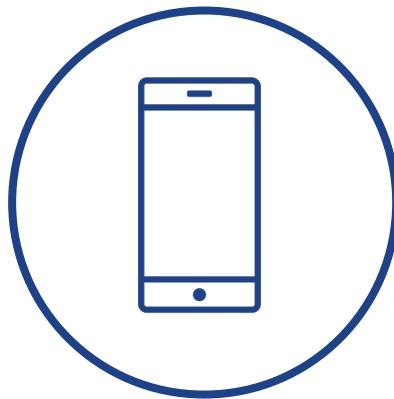
The “Why”



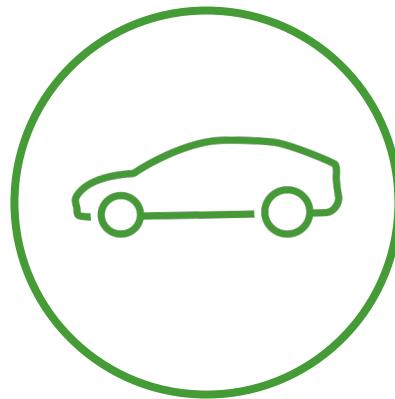
Website



Tablet



Phone



Car

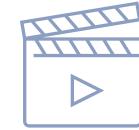
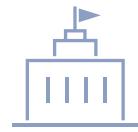


House

The Why

needs to be a custom
^

Every Company ~~is~~ a Software Company



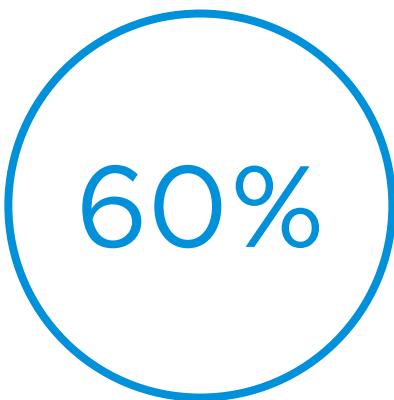
Good Software Drives Business Outcomes

Top performers compared to peers

The Why



Increased
Revenue
Growth



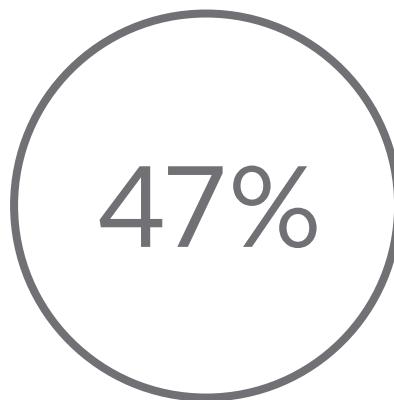
Higher
Shareholder
Returns



Higher
Operating
Margins



Increased
Innovation



Higher
Retention
Rates

Not just for the Netflix and Spotifys of the world

Dinero.dk

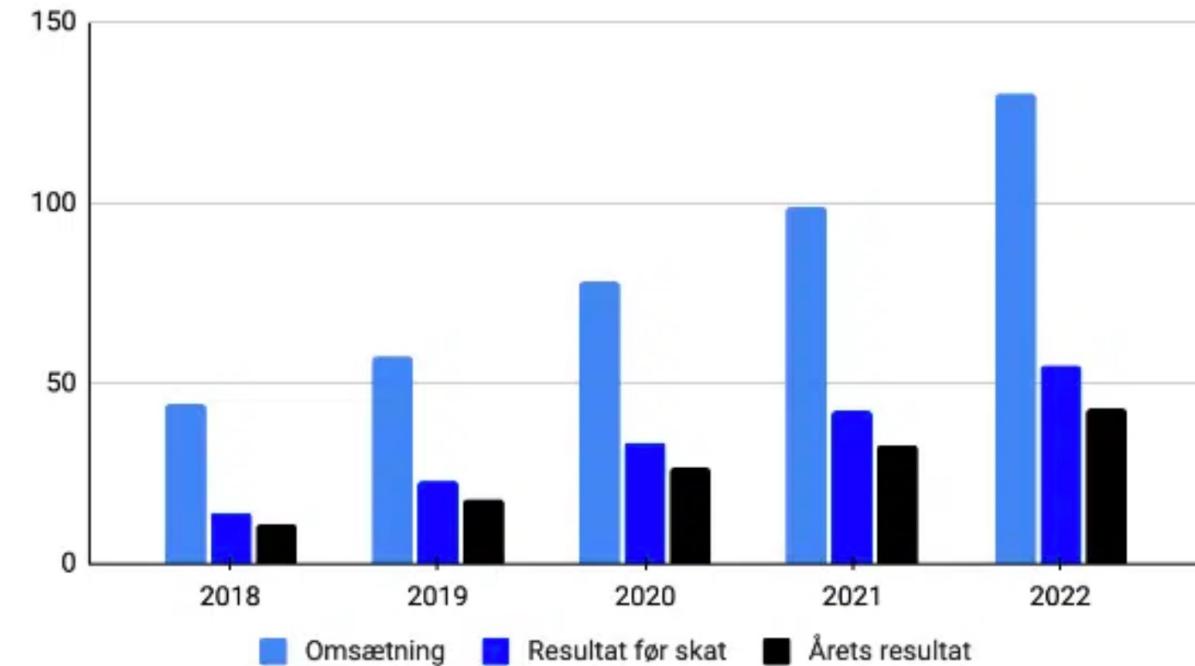


Martin Buch Thorborg
@thorborg

Hvor ofte tror du vi lægger en ny version af Dinero online?

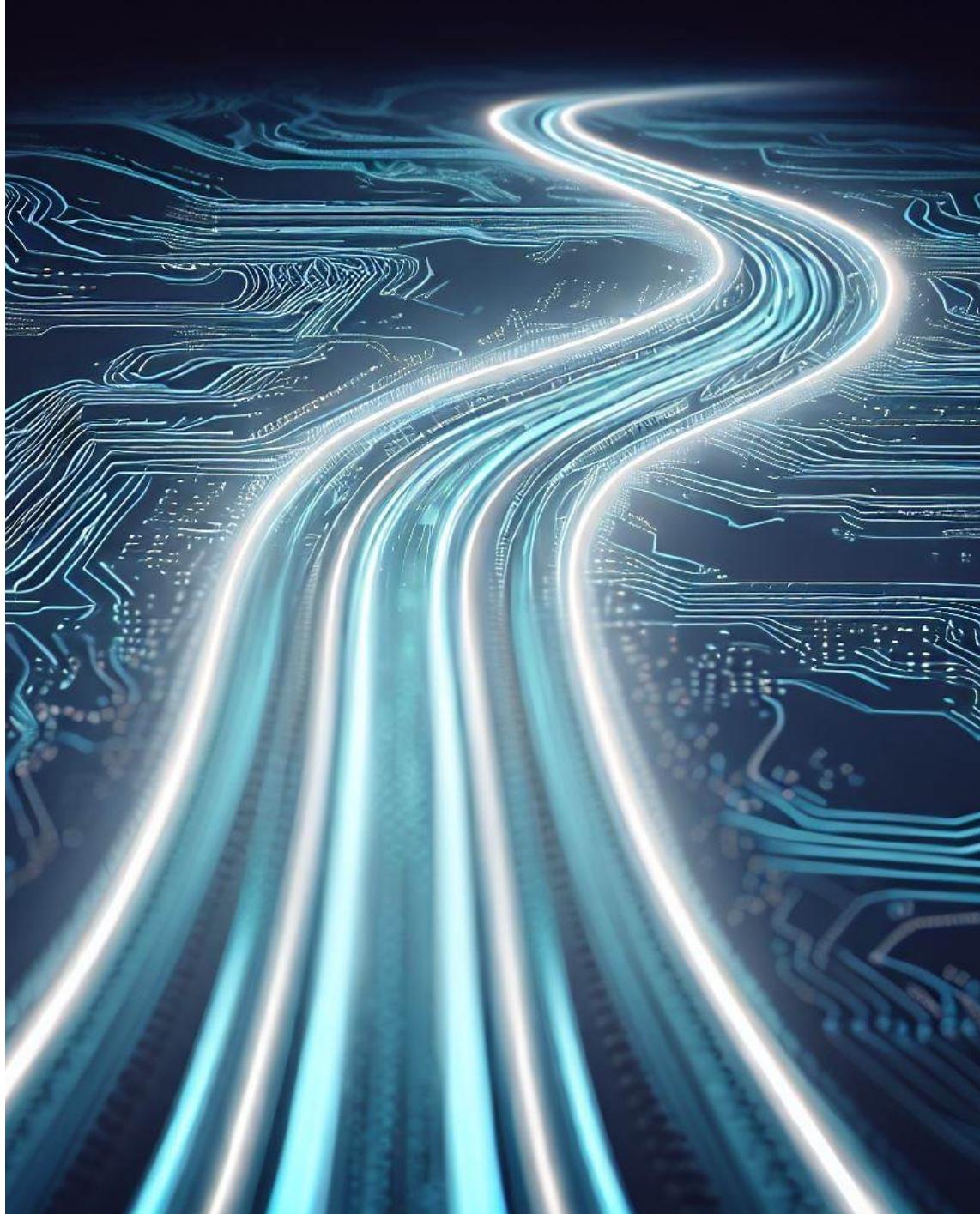
19.00 · 23/01/2023

Nøgletal - i mio. kroner



Reference

How do you get there ?



Supply-chain Levels for Software Artifacts, or SLSA ("salsa").

It's a security framework, a checklist of standards and controls to prevent tampering, improve integrity, and secure packages and infrastructure. It's how you get from "safe enough" to being as resilient as possible, at any link in the chain.

SLSA Levels

Level	Description	Example
1	Documentation of the build process	Unsigned provenance
2	Tamper resistance of the build service	Hosted source/build, signed provenance
3	Extra resistance to specific threats	Security controls on host, non-falsifiable provenance
4	Highest levels of confidence and trust	Two-party review + hermetic builds

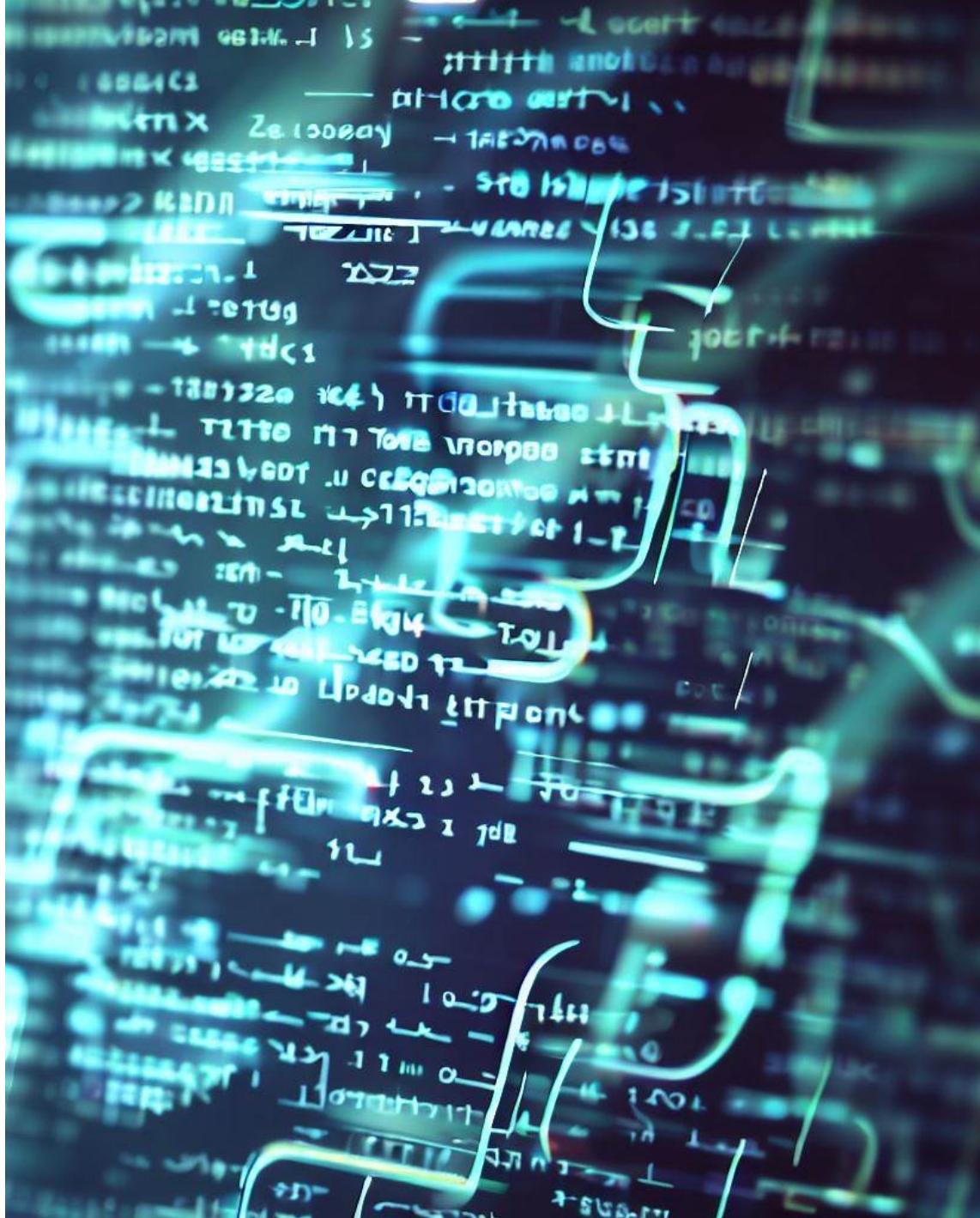
<https://slsa.dev/spec/v0.1/levels>

Level Requirements

Requirement	SLSA 1	SLSA 2	SLSA 3	SLSA 4
Source - Version controlled		✓	✓	✓
Source - Verified history			✓	✓
Source - Retained indefinitely			18 mo.	✓
Source - Two-person reviewed				✓
Build - Scripted build	✓	✓	✓	✓
Build - Build service		✓	✓	✓
Build - Build as code			✓	✓
Build - Ephemeral environment			✓	✓
Build - Isolated			✓	✓
Build - Parameterless				✓
Build - Hermetic				✓
Build - Reproducible				○

<https://slsa.dev/spec/v0.1/requirements>

Scripted Builds



What Tools To Use ?

Remember “As Code” & “Idempotence”



- Terraform
- Ansible
- Bash
- Puppet
- SaltStack (Salt)
- Powershell
- Python
- Golang
- Others ?

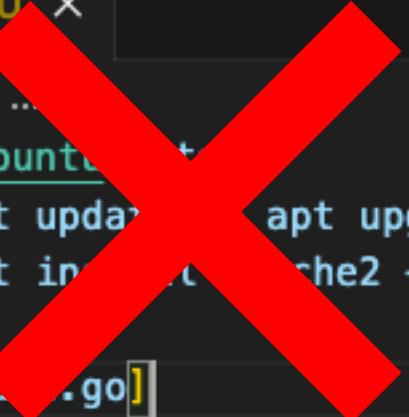
Idempotence

Get the same result every time



SLA Level 1

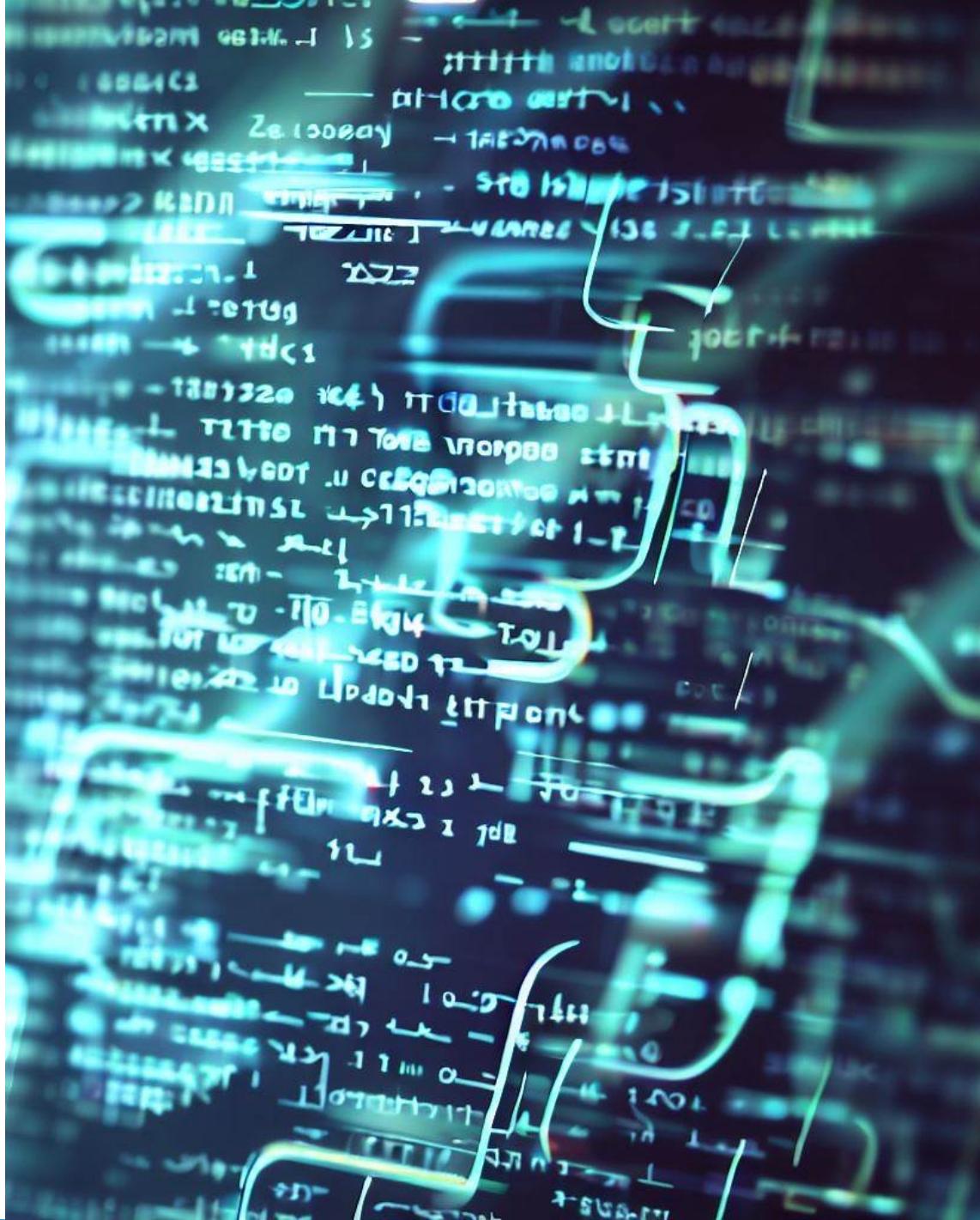
BUILD... 1/3 Best effort
SOURCE... 1/3 Best effort
DEPS.... 1/3 Best effort



```
Dockerfile 1, U X
Dockerfile > ...
1 FROM ubuntu
2 RUN apt update && apt upgrade -y
3 RUN apt install apache2 -y
4 COPY .
5 CMD [main.go]
```

```
Dockerfile U X
Dockerfile > ...
1 FROM ubuntu:jammy-20230425
2 RUN apt update && apt install apache2=2.4.52-1ubuntu4.5 -y
3 COPY .
4 CMD [main.go]
```

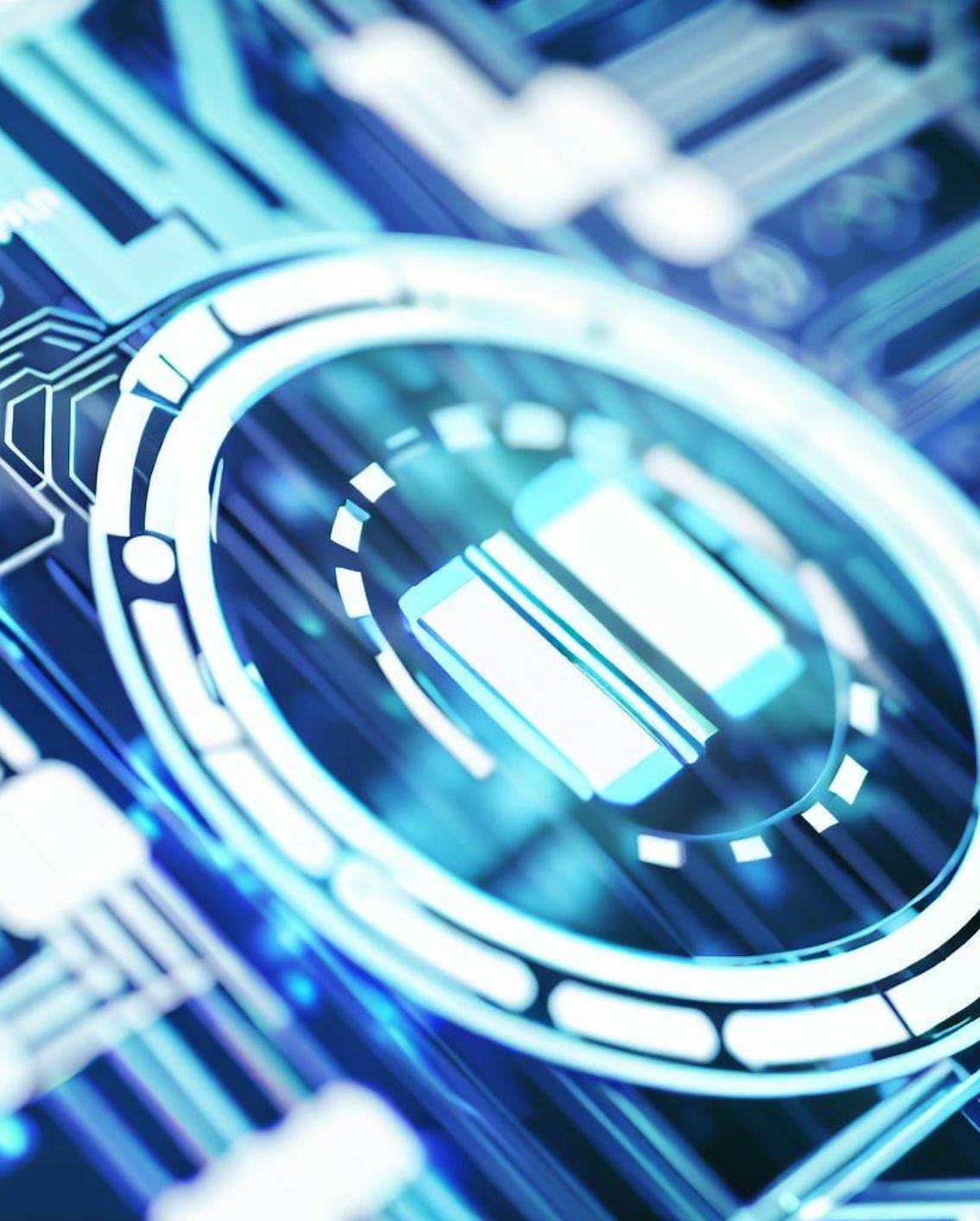
Version Controlled



The power of Git

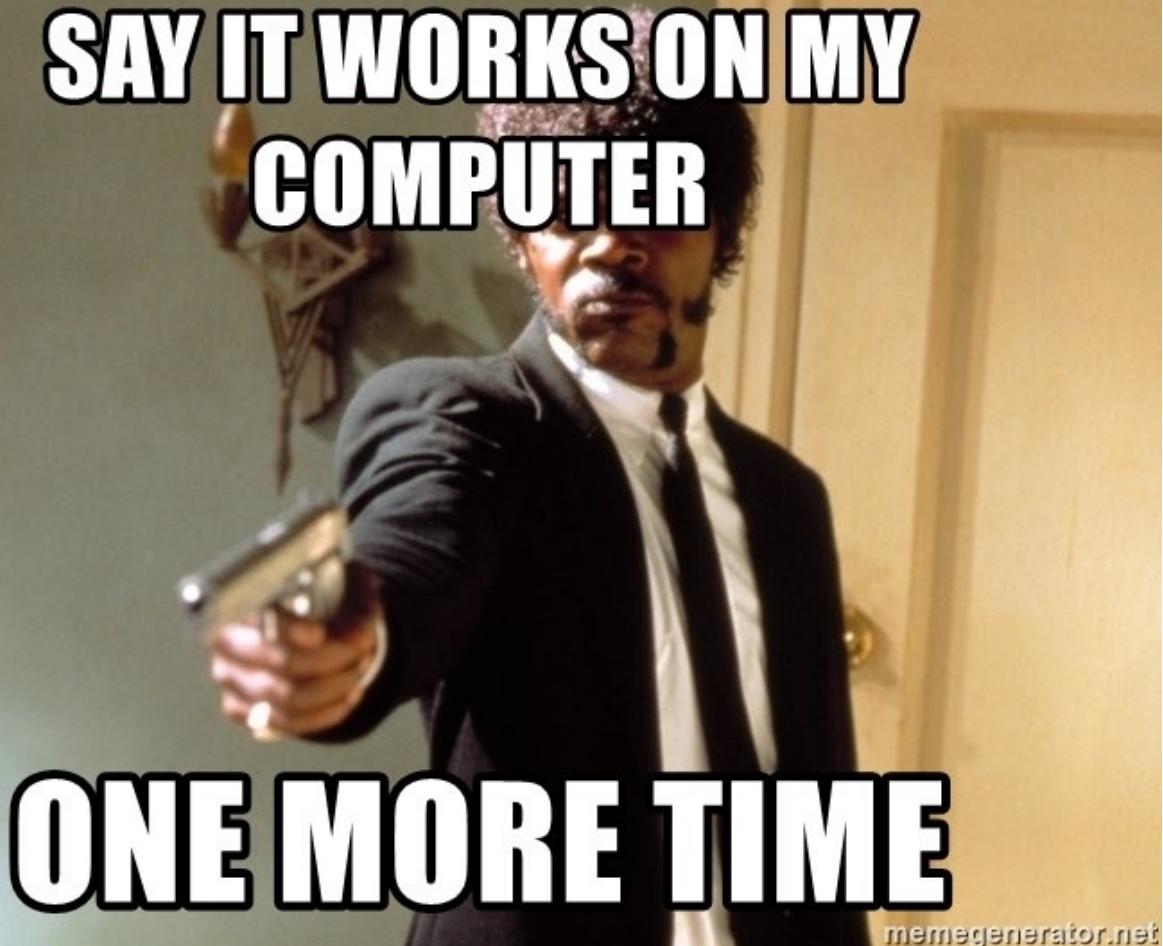
Gra...	Description	Date	Author	Commit
o	o p master 2	28 Apr 2023 10:54	Robert Jensen	9afab749
	Commit: 9afab74969bf80bee349cdd4804a8ae6d742bf18 Parents: 12bad7610d317d01aa18a7f270a96f24a3a05011			x
	main ▾			
-o	Commits on Apr 27, 2023			
	minor update rjhensen79 committed 13 hours ago			
	working with new login feature rjhensen79 committed 13 hours ago			
	fix rjhensen79 committed yesterday			
o	initial commit	28 Apr 2023 10:52	Robert Jensen	538da353

Build Service



Build Service

Which one to use ?



- Github Action
- Jenkins
- Travis CI
- Aria Pipelines
- Others



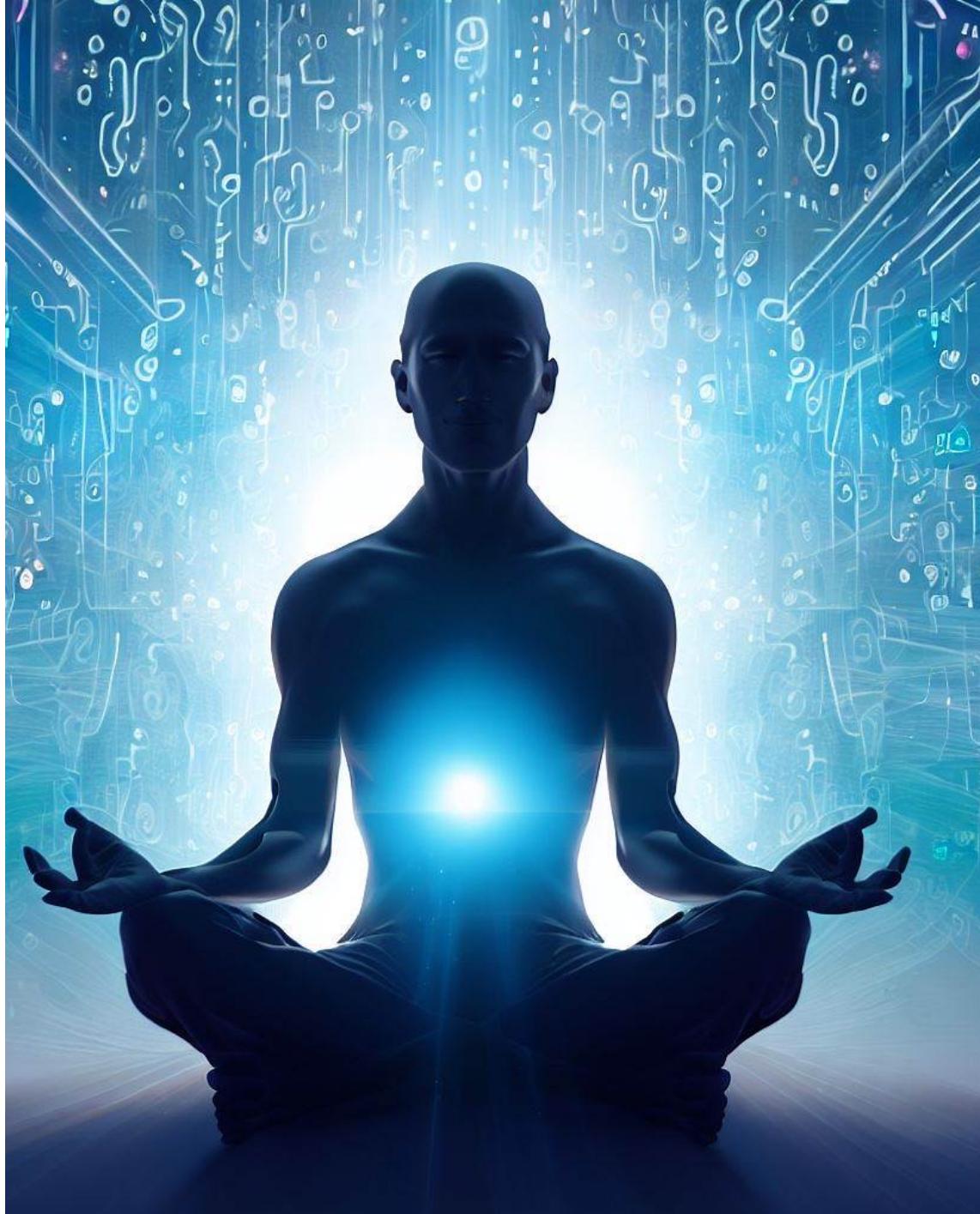
SLSA Level 2

BUILD....2/3 Credible
SOURCE...1/3 Best effort
DEPS.....1/3 Best effort

Simple Github Actions example of building code, at every push.

```
1  name: Go
2  on:
3    push:
4      branches: [ "main" ]
5  jobs:
6    build:
7      runs-on: ubuntu-22.04
8      steps:
9        - uses: actions/checkout@v3
10       - name: Set up Go
11         uses: actions/setup-go@v3
12         with:
13           go-version: 1.19
14       - name: Build
15         run: go build -v ./...
16       - name: Test
17         run: go test -v ./...
```

What is the/one solution ?



Backstage





Backstage

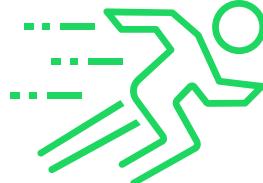
«Happy developers make happy code»

The Spotify Journey

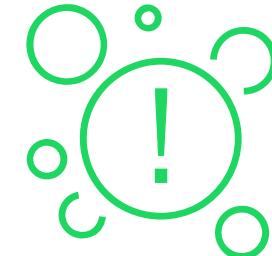
.. back in 2016



hyper-growth
mode



speed-to-
market



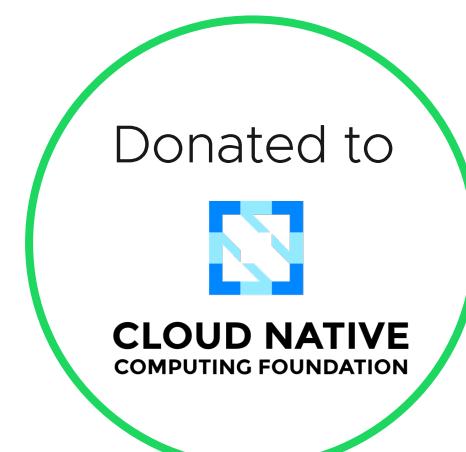
developer
effectiveness



Backstage is an open platform for building developer portals.



2016



2020

Attributes Of Developer Portal

Lowers cognitive load
for developers and
boosts Developer
Productivity by
creating



Provides a developer centric view



Abstracts away underlying technology

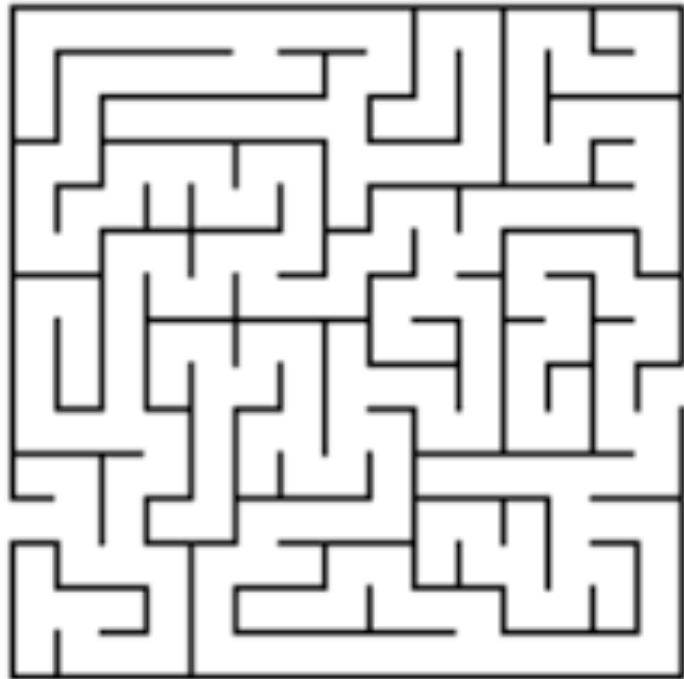


Provides a pluggable framework

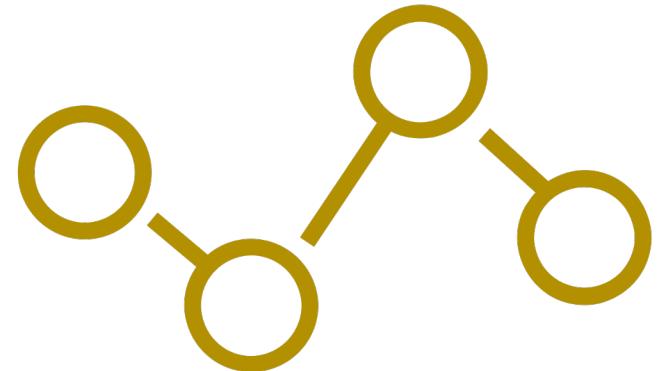


Maximizes Time to Value

Finding the golden path | path to production



This is the way



Golden Path

Backstage plugin ecosystem

Customizable and extensible plugin architecture

- Built with modern technologies and common frameworks
- Makes it easy to develop for and contribute to your dev portal
- Cloud-agnostic and vendor-neutral



 Azure Sites by FRISS <small>Infrastructure</small> Azure Sites (Apps & Functions) support for a given entity. View the current status of the site, quickly jump to site's Overview page, or Log Stream page. Explore	 backstage-plugin-api-linter by Zalando <small>Linting</small> API Linter is a quality assurance tool that checks the compliance of API's specifications to Zalando's API rules. Explore	 Badges by Andreas Stenius <small>Discovery</small> The badges plugin offers a set of badges that can be used outside of Backstage, showing information related to data from the catalog. Explore
 Bazaar by Axis Communications AB <small>Discovery</small> A marketplace where engineers can propose projects suitable for inner sourcing Explore	 Bitrise by SDA SE <small>CI/CD</small> View Bitrise builds and download the build artifacts within Backstage. Explore	 Bugsnag by roadie.io <small>Monitoring</small> View and monitor Bugsnag errors. Explore
 Buildkite by roadie.io <small>CI/CD</small> View Buildkite CI builds for your service in Backstage. Explore	 Catalog Graph by SDA SE <small>Discovery</small> Extend the Backstage Software Catalog with a graph that shows all entities and their relationships providing an easier way to discover the ecosystem. Explore	 CircleCI by Spotify <small>CI/CD</small> Automate your development process with CI hosted in the cloud or on a private server. Explore
 Cloud Carbon Footprint by Thoughtworks <small>Metrics</small> View your cloud carbon footprint by estimating energy use (kilowatt-hours) and carbon emissions (metric tons CO2e) from public cloud usage. Explore	 Cloudify by Cloudify <small>Orchestration</small> Cloudify provides a remote execution and environment management backend for Kubernetes, Terraform, Ansible, etc. Explore	 CodeScene by CodeScene <small>Quality</small> CodeScene is a multi-purpose tool bridging code, business and people. See hidden risks and social patterns in your code. Prioritize and reduce technical debt. Explore

Backstage in open source



**CLOUD NATIVE
COMPUTING FOUNDATION**



1,500+
contributors



3,200+
project forks



400+
adopters



7,300+
members



13,000+
contributions

Backstage is building a proven track record across industries



NETFLIX

ROKU

 **PELOTON**

 **Expedia**

 **zalando**

 **American
Airlines**

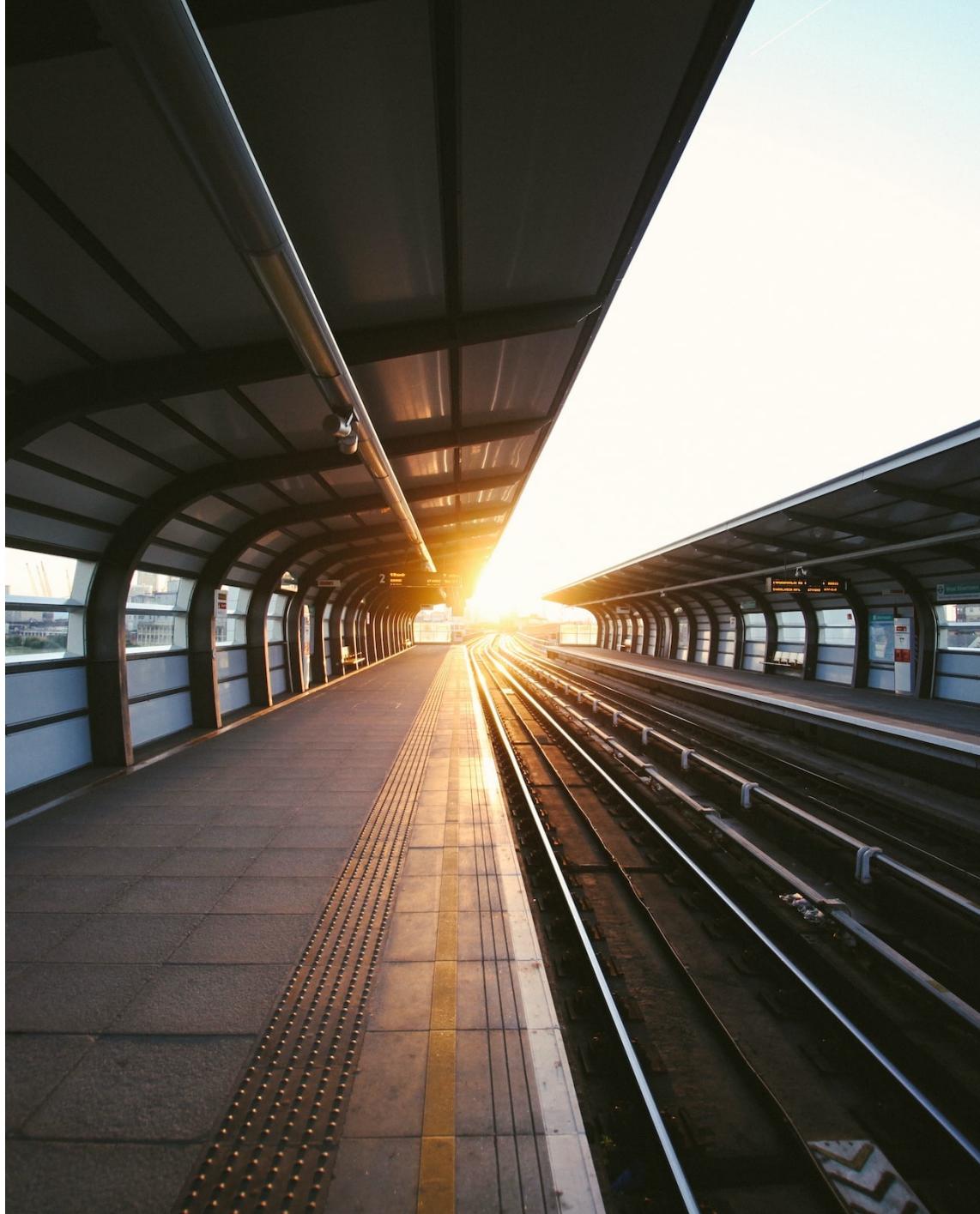
 **TELUS**

 **TOYOTA**

Currently ~~75 224~~ 241 public adopters

<https://github.com/backstage/backstage/blob/master/ADOPTERS.md>

Tanzu Application Platform (TAP)



Based on Backstage, but with it's own opinions (and focus)

The screenshot shows the Tanzu Application Platform interface. At the top, there's a search bar and a sidebar for 'Accelerators' containing items like 'Hello Fun', 'Helloytt', and 'New Accelerator'. Below this is the 'Your Organization Catalog' section, which lists components such as 'backstage-component', 'petclinic-app', and 'tanzu-java-web-app'. On the right, a detailed view of the 'petclinic-app' service is shown, including tabs for 'OVERVIEW', 'DEPENDENCIES', and 'WORKLOADS'. The 'WORKLOADS' tab displays status conditions like 'Ready', 'ConfigurationsReady', and 'RoutesReady', all of which are currently true. Annotations and labels for the service are also visible.



Pre installed with

➤ Docs

➤ Accelerators

➤ API's

➤ Supply Chain

➤ Security Analytics

Only runs on K8S

Overview

The screenshot shows the VMware Tanzu Application Platform interface. The top navigation bar includes a logo, a search icon, and a user profile icon. Below the navigation is a header bar with the title "Your Organization Catalog". On the left side, there is a sidebar with sections for "PERSONAL" (Owned: 0, Starred: 0), "YOUR ORGANIZATION" (All: 4), and "OWNER" (dropdown menu). At the bottom of the sidebar are "TAGS" and a dropdown menu. The main content area displays a table titled "All (4)" with columns for "Name", "Owner", "Description", "Tags", and "Actions". The table lists four entities: "cncf" owned by "denmark-team" with description "CNCF Demo" and tags "quiz", "postgres"; "demo" owned by "denmark-team" with description "Demo Appl..." and tags "quiz", "postgres"; "tanzuquiz" owned by "denmark-team" with description "Tanzu Quiz..." and tags "quiz", "postgres"; and "tanzutrends" owned by "denmark-team" with description "Tanzu..." and tags "quiz", "postgres". Each row has an "Actions" column with edit and star icons.

- Owners
- Systems
- Groups
- Dependencies
- Resources

Documentation

The screenshot shows the VMware Tanzu Application Platform interface. At the top, it displays the application name "tanzutrends" with a star icon. Below the title, there are two tabs: "Overview" (selected) and "Diagram".

The "Overview" tab contains the following information:

- About:** Includes "VIEW SOURCE" and "VIEW TECHDOCS" buttons.
- DESCRIPTION:** Tanzu Trends Demo Application.
- OWNER:** denmark-team.
- DOMAIN:** tap-gui-domain.
- TAGS:** No Tags.

The "Relations" section displays a graph diagram showing the relationships between components:

```
graph TD; group[denmark-team] -- ownerOf / ownedBy --> system[tanzutrends]; system -- hasPart / partOf --> componentFrontend[component:frontend]; system -- hasPart / partOf --> componentScrape[component:scrape]; system -- hasPart / partOf --> resourceDb[resource:tanzutrends-db]
```

A "View graph →" button is located below the graph.

The "Has components" section lists the following components:

Name	Owner	Type	Lifecycle	Description
frontend	denmark-team	website	production	The frontend for the Tanzu...
scrape	denmark-team	service	production	The scraper for the Tanzu...

The "APIs" section states: "This system does not contain any APIs. Learn how to change this."

- Based on Mkdocs
- Written in markdown
- Documentation & Code in same repo.
- Build when you push

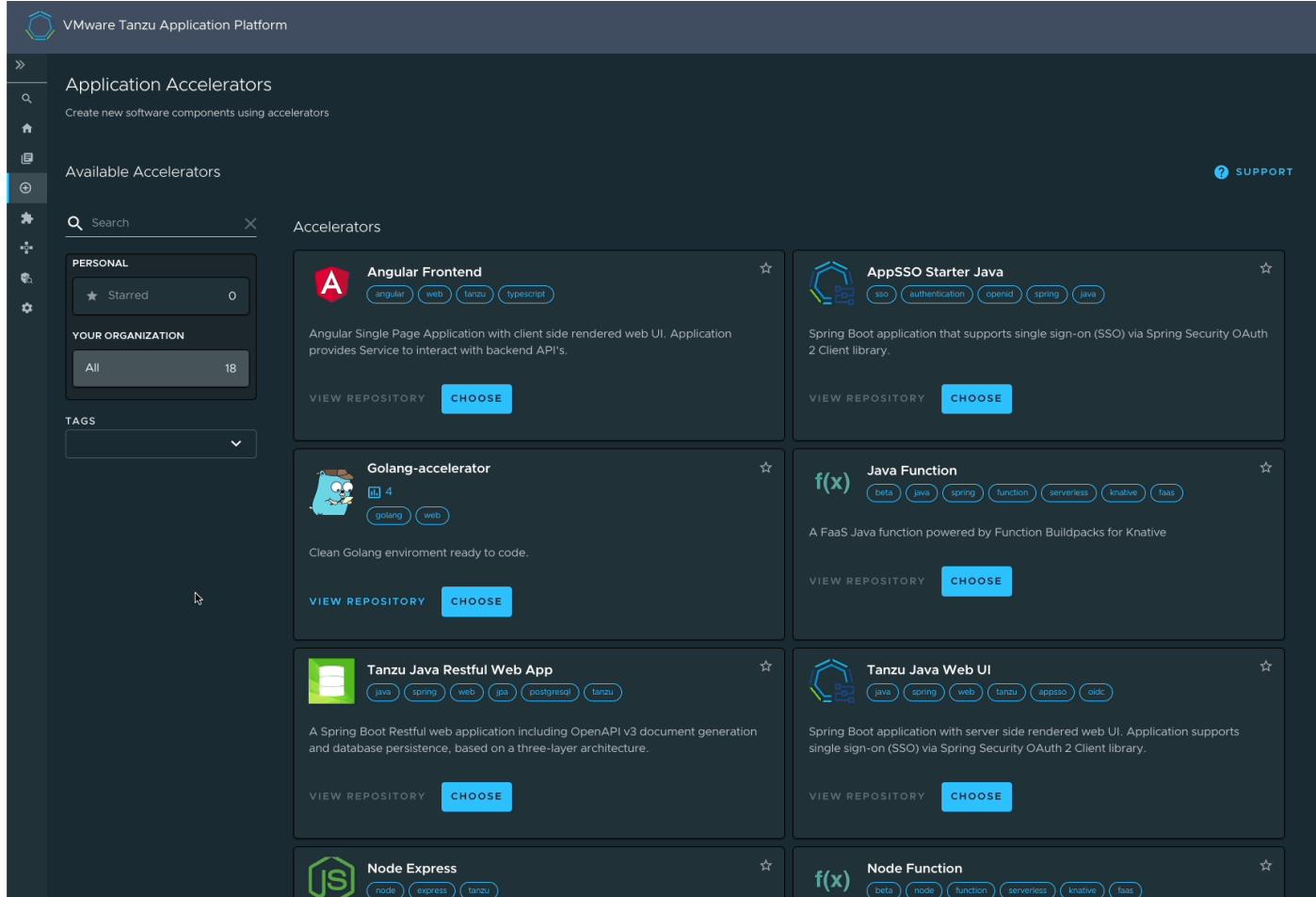
API

The screenshot shows the VMware Tanzu Application Platform API Explorer. The left sidebar has a navigation menu with icons for Home, APIs, Catalog, Pipelines, and Support. The main area is titled "APIs" and "Your Organization API Explorer". It features a search bar and filter dropdowns for "Type" (All), "OWNER" (All), and "LIFECYCLE" (All). A sidebar on the left shows metrics for "PERSONAL" (Owned: 0, Starred: 0) and "YOUR ORGANIZATION" (All: 1). The central table lists one API entry:

Name	System	Owner	Type	Lifecycle	Description	Tags	Actions
fastapi-test-vmlab.tanzu.dk	demo	denmark-team	openapi	vmlab.tanzu.dk	A set of API...		

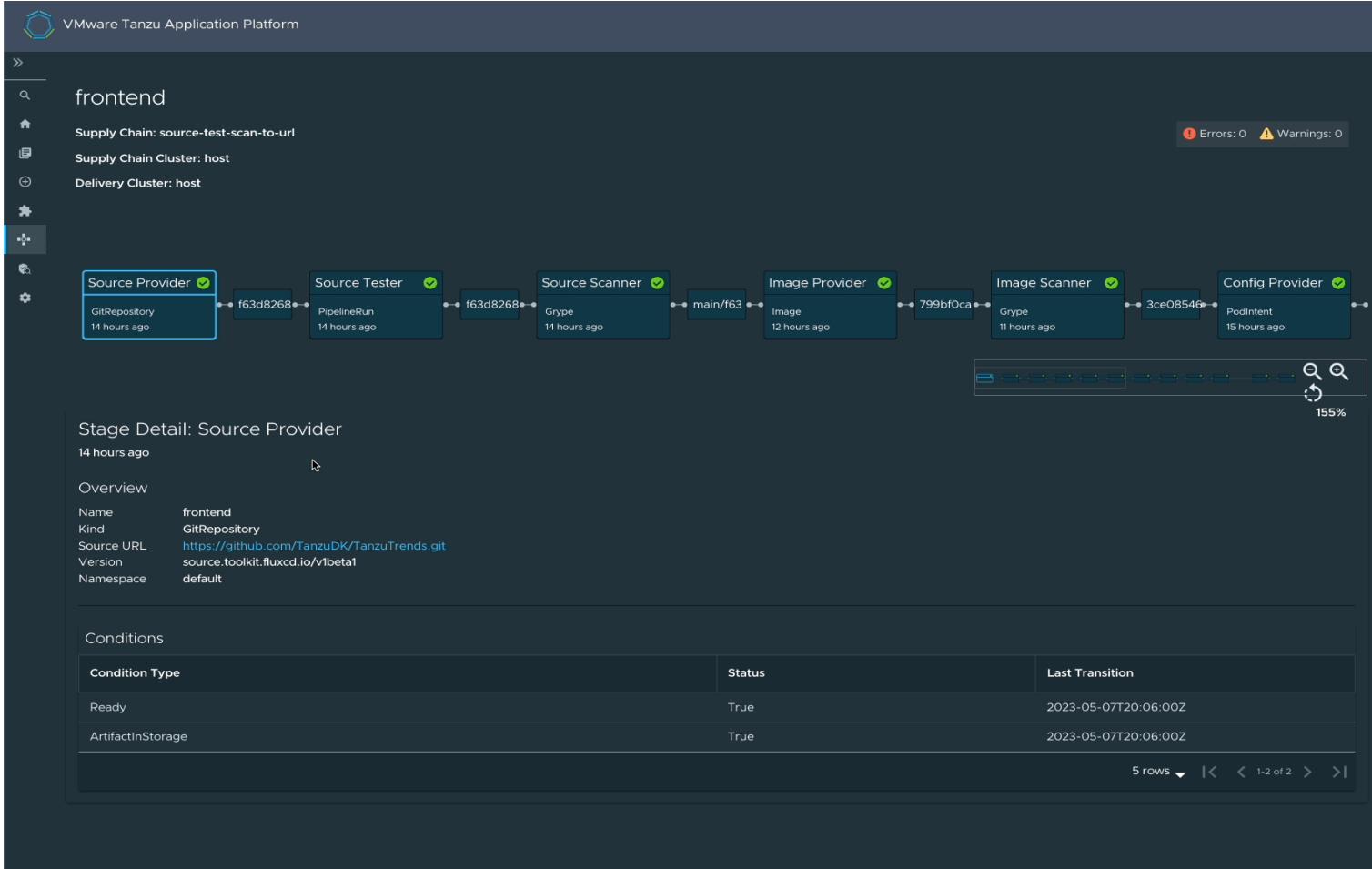
- Based on OpenAPI
- Auto updated with app
- Dependencies
- Internal / External Api's
- Swagger interface

Application Accelerators



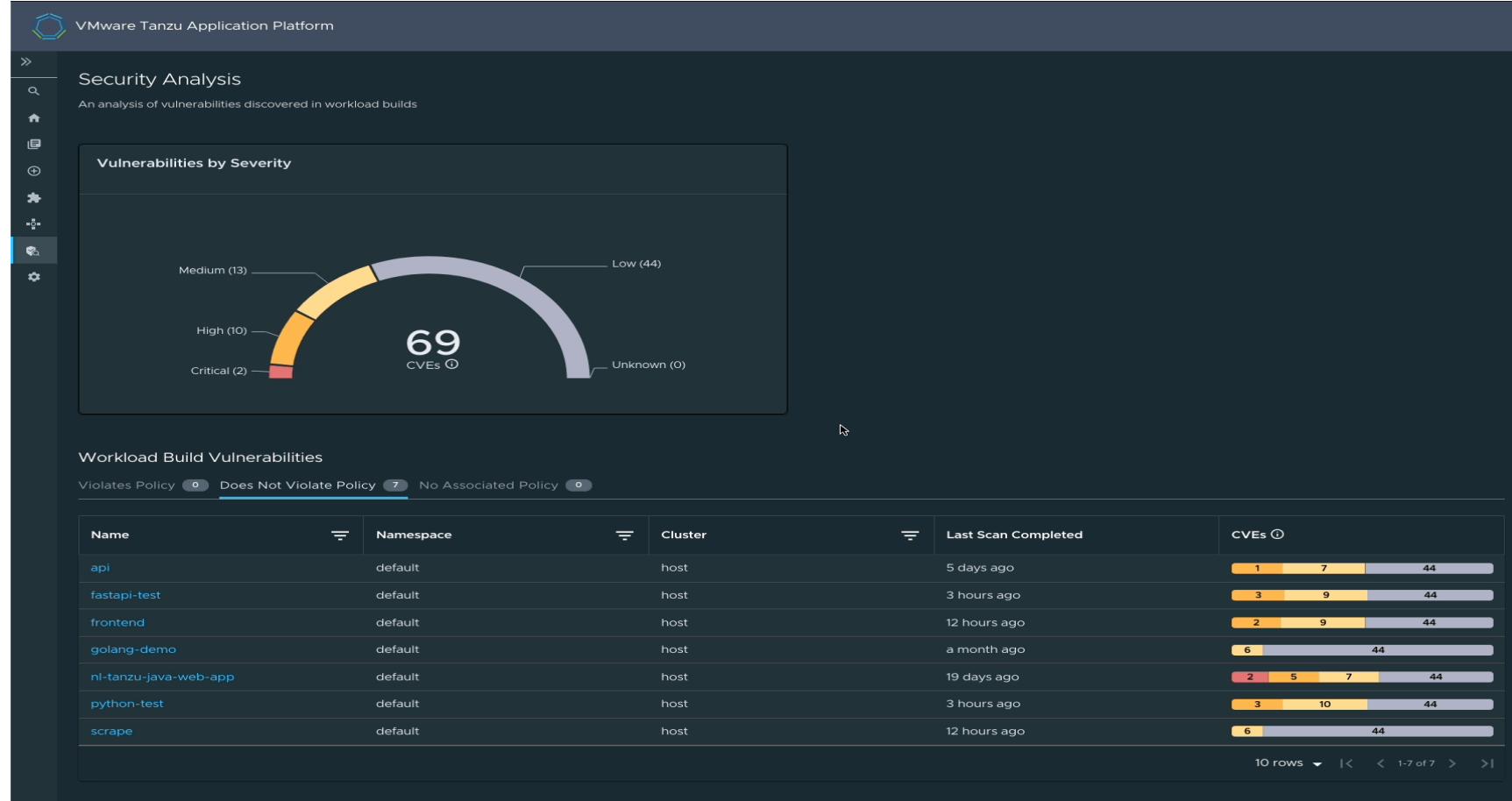
- Templates for Apps, Baselines etc.
- Can be "anything"
- Takes input
- Customizable using Sed, YTT, etc.
- “Golden Path” build in.

Supply Chain



- Supply Chain UI
- CVE info
- Test info
- Overview

Security Analysis



- Security overview

Run

VMware Tanzu Application Platform

Your Organization Catalog

Components ▾

Type All

PERSONAL

- Owned 0
- Starred 0

YOUR ORGANIZATION

- All 7

OWNER

LIFECYCLE

TAGS

Name	System	Owner	Type	Lifecycle	Description	Tags	Actions
api	tanzuquiz	denmark-team	api	production	The api for...	golang	<input type="checkbox"/> <input type="pen"/> <input type="star"/>
fastapi-test	demo	denmark-team	service	experimental	FastAPI fr...	python fastapi api	<input type="checkbox"/> <input type="pen"/> <input type="star"/>
frontend	tanzutrends	denmark-team	website	production	The...	python web	<input type="checkbox"/> <input type="pen"/> <input type="star"/>
golang-demo	cncf	denmark-team	service	experimental	A golang...	golang web	<input type="checkbox"/> <input type="pen"/> <input type="star"/>
ni-tanzu-java-web-app	java-web-app	dutch-team	service	experimental	Tanzu Java...	app-accelerator java spring web tanzu	<input type="checkbox"/> <input type="pen"/> <input type="star"/>
python-test	demo	denmark-team	service	experimental	A python...	python streamlit	<input type="checkbox"/> <input type="pen"/> <input type="star"/>
scrape	tanzutrends	denmark-team	service	production	The scraper...	golang	<input type="checkbox"/> <input type="pen"/> <input type="star"/>

- Run the app, on the TAP cluster (or a dedicated run cluster)

Built with an open source-first mindset

Innovative, interoperable, scalable and secure solutions

Tanzu Application Platform is backed by some of the most mature and popular open-source projects available today

In addition to Backstage Backstage

Garnering 200+ plugins Backstage has gained tremendous traction by helping organizations build self-service developer portals

Carvel CARVEL

Developers build, deploy, and manage their own apps and package them so they are more easily distributable

Cartographer CARTOGRAPHER

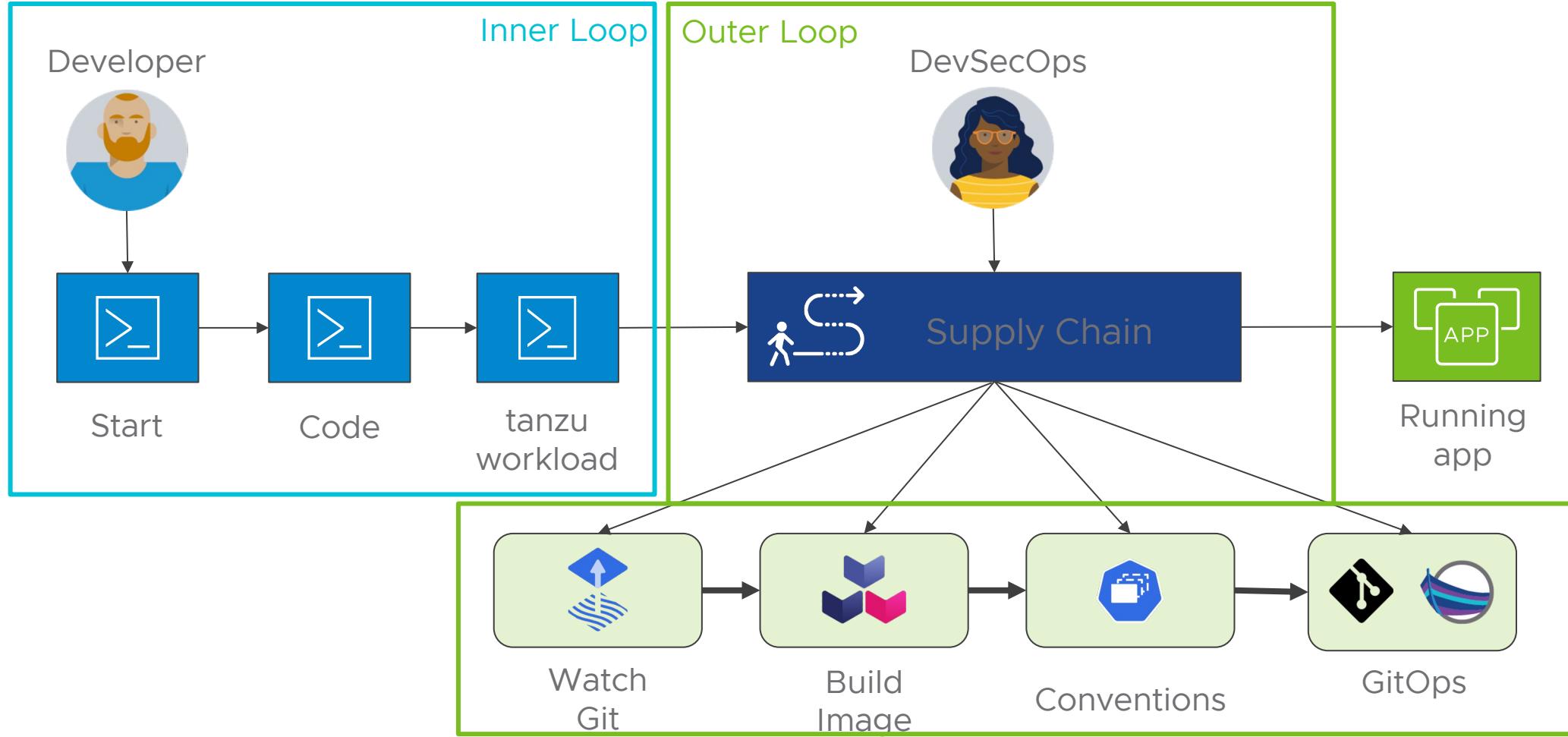
Operator teams create secure, reusable supply chains that define all of application CI and CD in one place

And many more....

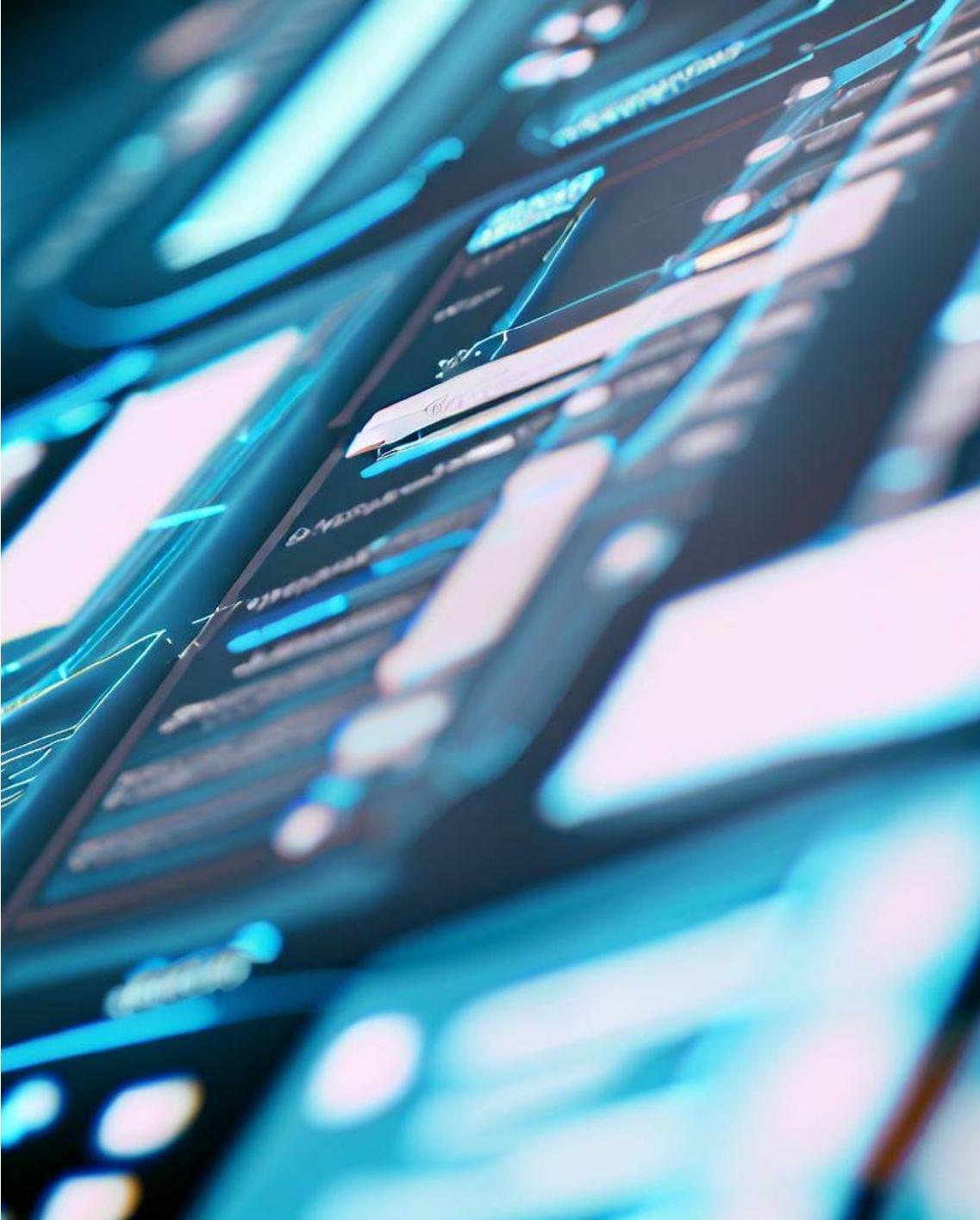


Building open-source software and contributing to communities is at the core of VMware's engineering spirit

Deploying an App with Tanzu Application Platform



Come by the booth, for a
“real” demo and a talk.



Thank You

