JFrog

Future-Proofing the DevSecOps Pipeline from Code to Cloud

BETTER Together 2024

About the speaker

A solution engineering veteran with a wealth of experience and expertise in IT. His tenure at NGINX and F5 Networks honed his acumen in network optimisation and application security, playing an instrumental role in safeguarding the operations of multinational enterprises and helping him cultivate a nuanced understanding of the dynamic threat landscape confronting modern businesses.

Today, as a Senior Solutions Engineer at JFrog, Mike remains steadfast in his advocacy for application security and operating a robust software development lifecycle (SDLC).

And fan of the Emmy Award winning TV show, The Bear.

Mike Holland
Senior Solutions Engineer

Email: mikeho@jfrog.com

Agenda

- Metrics of a High-Performing DevOps Team
- Common Problems in the SDLC
- The Way of the Frog!
- EveryOps with JFrog!

[Metrics] DevOps Team's Performance

Metric	What it measures	Why it is important
Deployment Frequency	How many times you successfully change production.	Identify problems with inefficient process, staff shortage, need for longer testing time. Track a team's velocity to delivery.
Lead time for Changes	How long it takes to go from code committed to code successfully running in production.	Helps analyse the teams response time to needs and fixes. Indicates efficiency of the process, code complexity, and team's capacity.
Time to Restore Service	How long it takes to restore service when an incident occurs.	Indicates a team's response time and development of efficient processes. Reveals inadequate team structure. Helps ensure availability of software.
Change Failure Rate	The percentage of deployments causing failure in production	Depicts time spent on fixing bugs vs. delivering new code. Indicates the team's capabilities, process efficiency, and the lack of testing before deployment.

Indie hackers, repeat after me!
Adding more features is better than a real marketing plan.

Common Problems

BETTER Together 2024

Common Problems

- Balancing Speed & Quality

  [Confidence to deploy quicker]

- Strategic Misalignment

  [R&D-Product conflict, Lack of visibility, Monitoring, feedback loops]

- Resources and Tool options

  [So few vs so many dilemma, Integration issues, Tools overload]

- Multiple Security Solutions

  [Lack of consistent coverage]

- Deployment failures

  [Testing results, Lack of evidence]

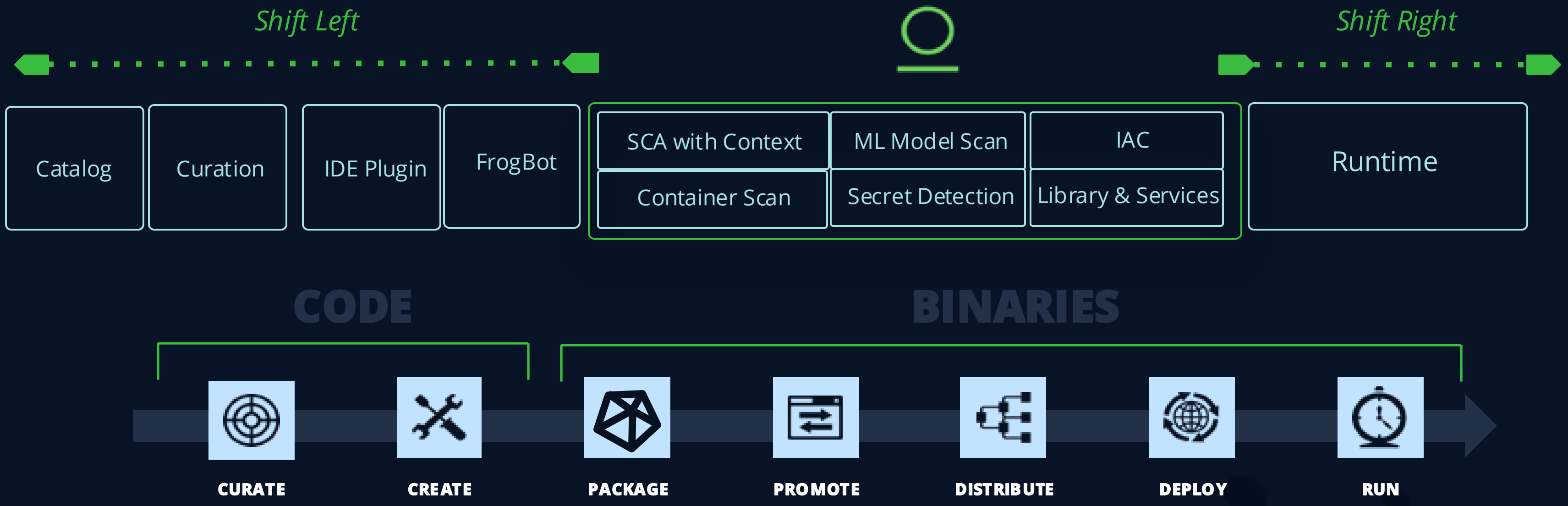
BETTER Together

JFrog

The Way of the Frog!

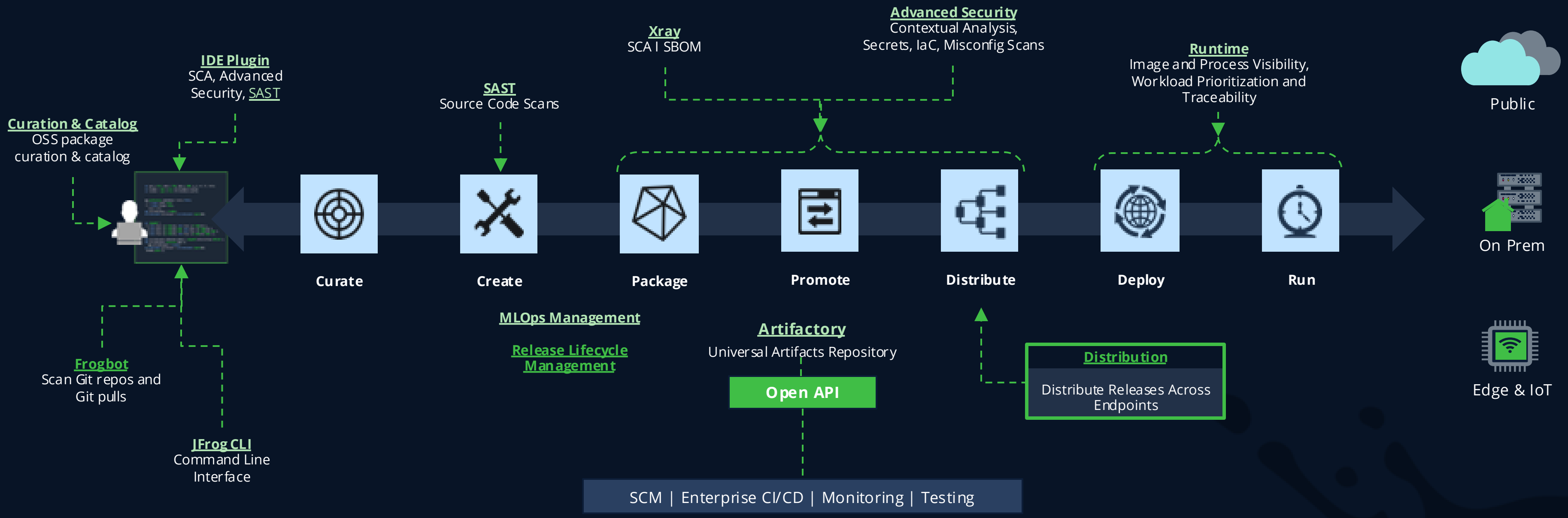
BETTER Together 2024

Software Supply Chain Security



Complete End-to-End Security from Left to Right

The JFrog Software Supply Chain Platform


Curation & Catalog
OSS package curation & catalog
IDE Plugin
SCA, Advanced Security, SAST
SAST
Source Code Scans
Xray
SCA I SBOM
Advanced Security
Contextual Analysis, Secrets, IaC, Misconfig Scans
Runtime
Image and Process Visibility, Workload Prioritization and Traceability
Public
On Prem
Edge & IoT
Curate
Create
Package
Promote
Distribute
Deploy
Run
Frogbot
Scan Git repos and Git pulls
JFrog CLI
Command Line Interface
MLOps Management
Release Lifecycle Management
Artifactory
Universal Artifacts Repository
Open API
Distribution
Distribute Releases Across Endpoints
SCM | Enterprise CI/CD | Monitoring | Testing

JFrog

BETTER Together

Siloed Solutions v/s Consolidated Approach

Cost Optimization

Single Source of Truth

Automate Everything

Single point of Support

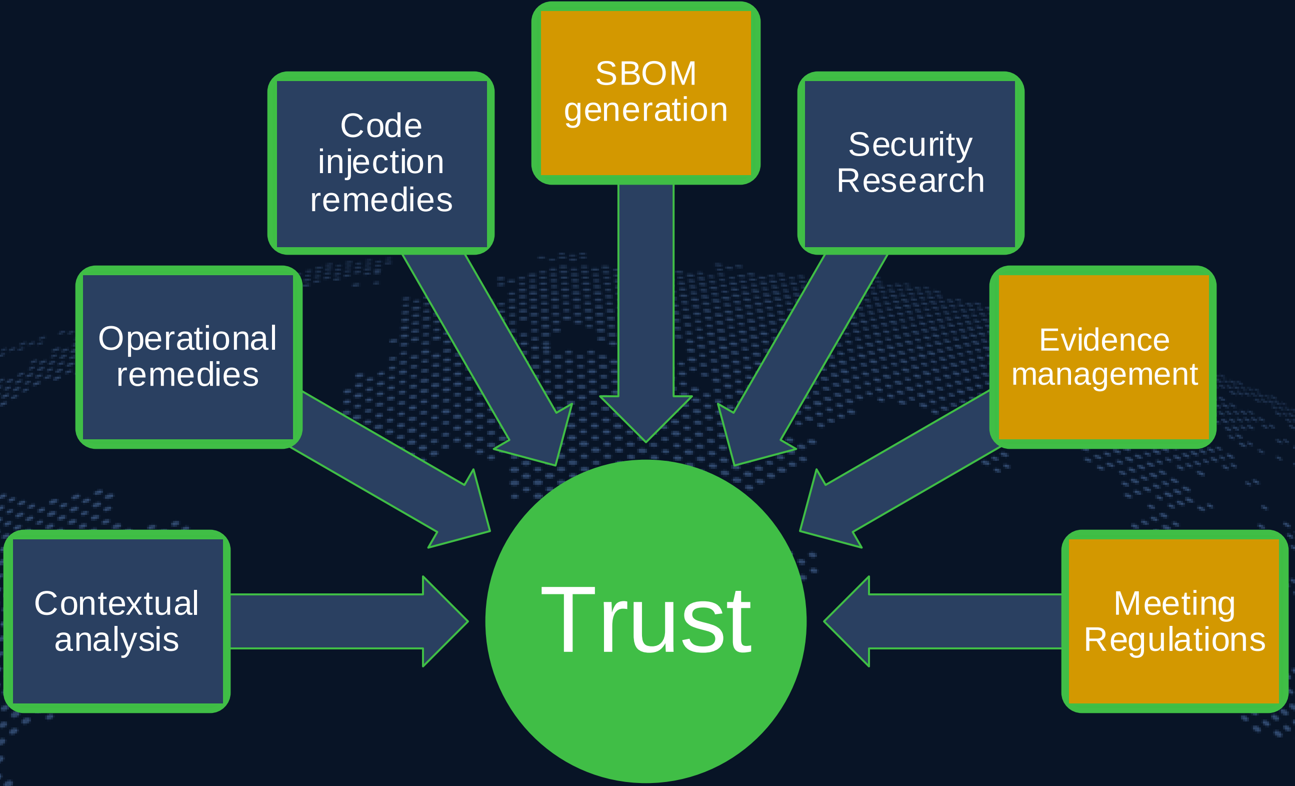
Right Data Capture

Enhanced Depth and Breadth

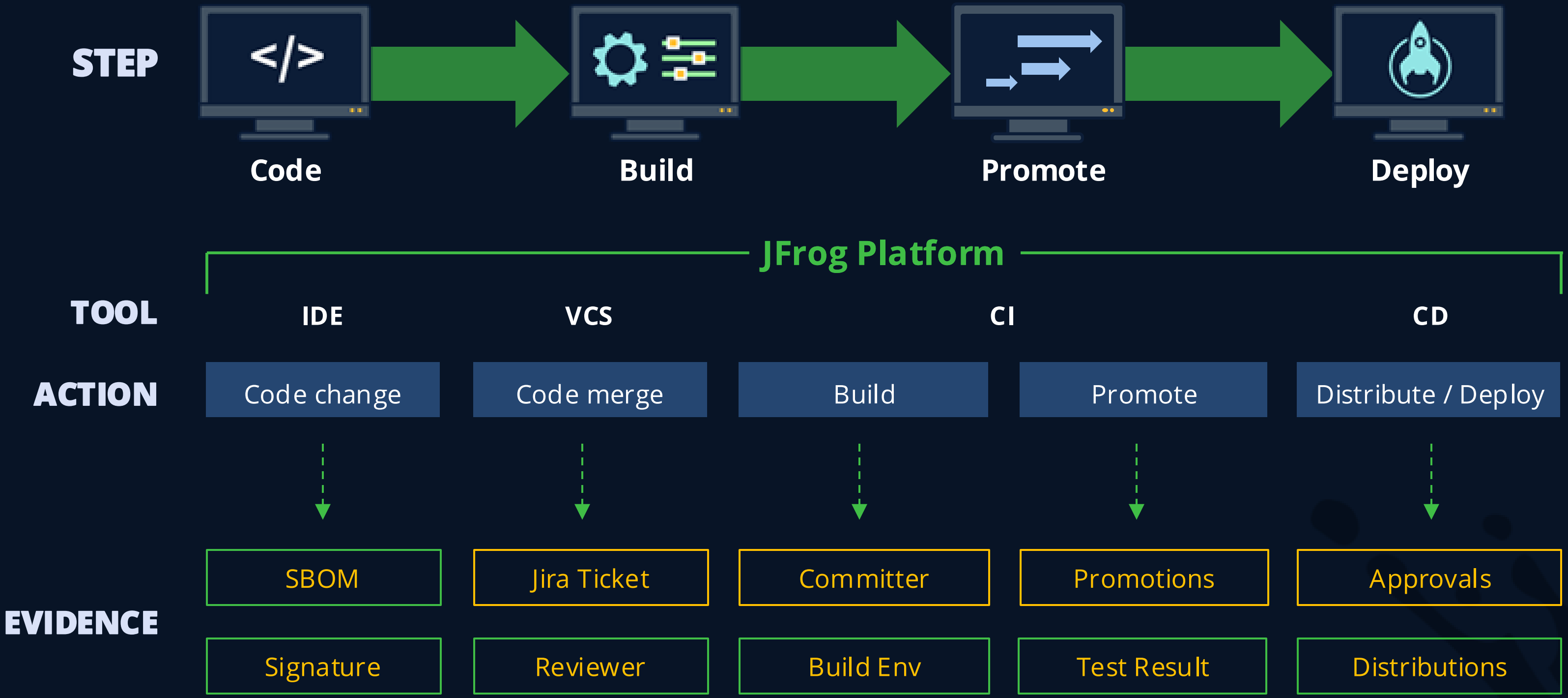
me putting last nights chickem mcnuggets in the air fryer that hasn't been cleaned since june 2021

BETTER Together

HOW CAN WE RELEASE FAST WITH **TRUST?**



SDLC End-To-End Evidence

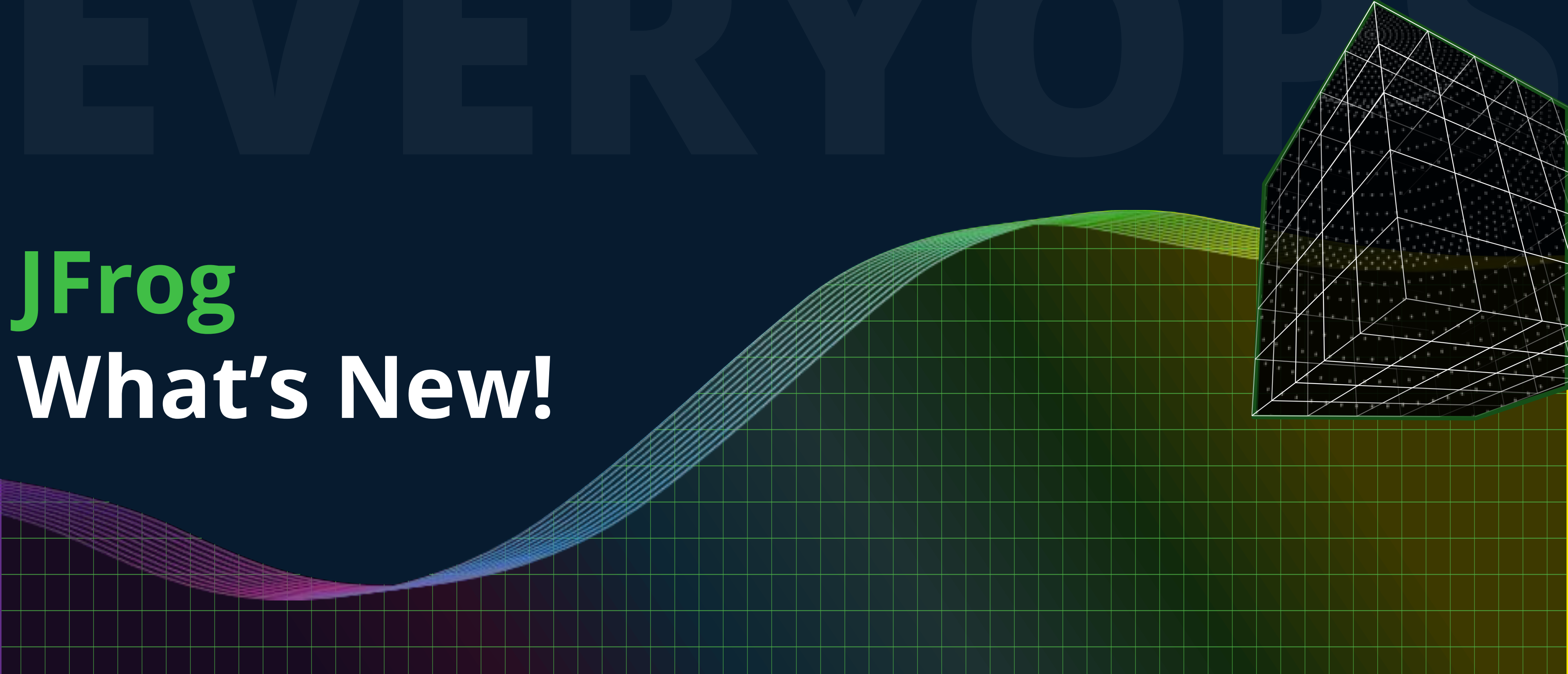


JFrog

BETTER Together

EVERYOPS

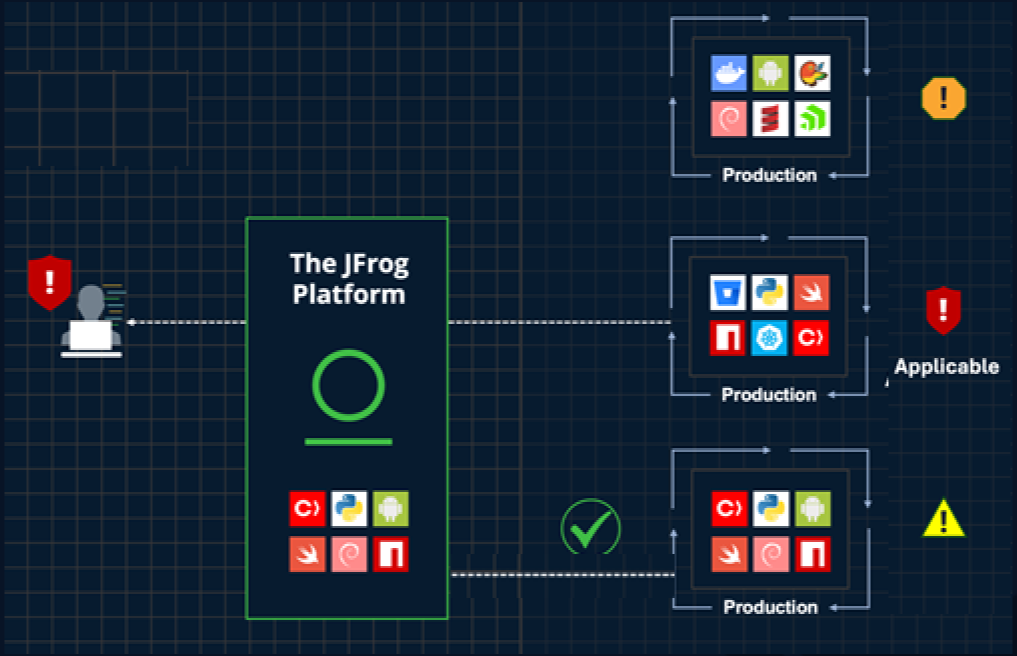
JFrog
What’s New!



JFROG Runtime

Fast Discovery & Remediation. From Code to Production

- **Get Real-Time Visibility**
  Into runtime environments
- **Verify Image Integrity**
  In runtime, with images from Artifactory
- **Prioritize & Focus on What Matters Most**
  Running images, with critical/high severity CVEs that are applicable
- **Bi-Directional Lineage**
  Triage and mitigate risk Immediately



BETTER Together

JFrog Runtime Uncover Hidden Risks

Runtime Integrity

- Image visibility
- Image integrity
- Image risk prioritization and traceability
- Workload visibility

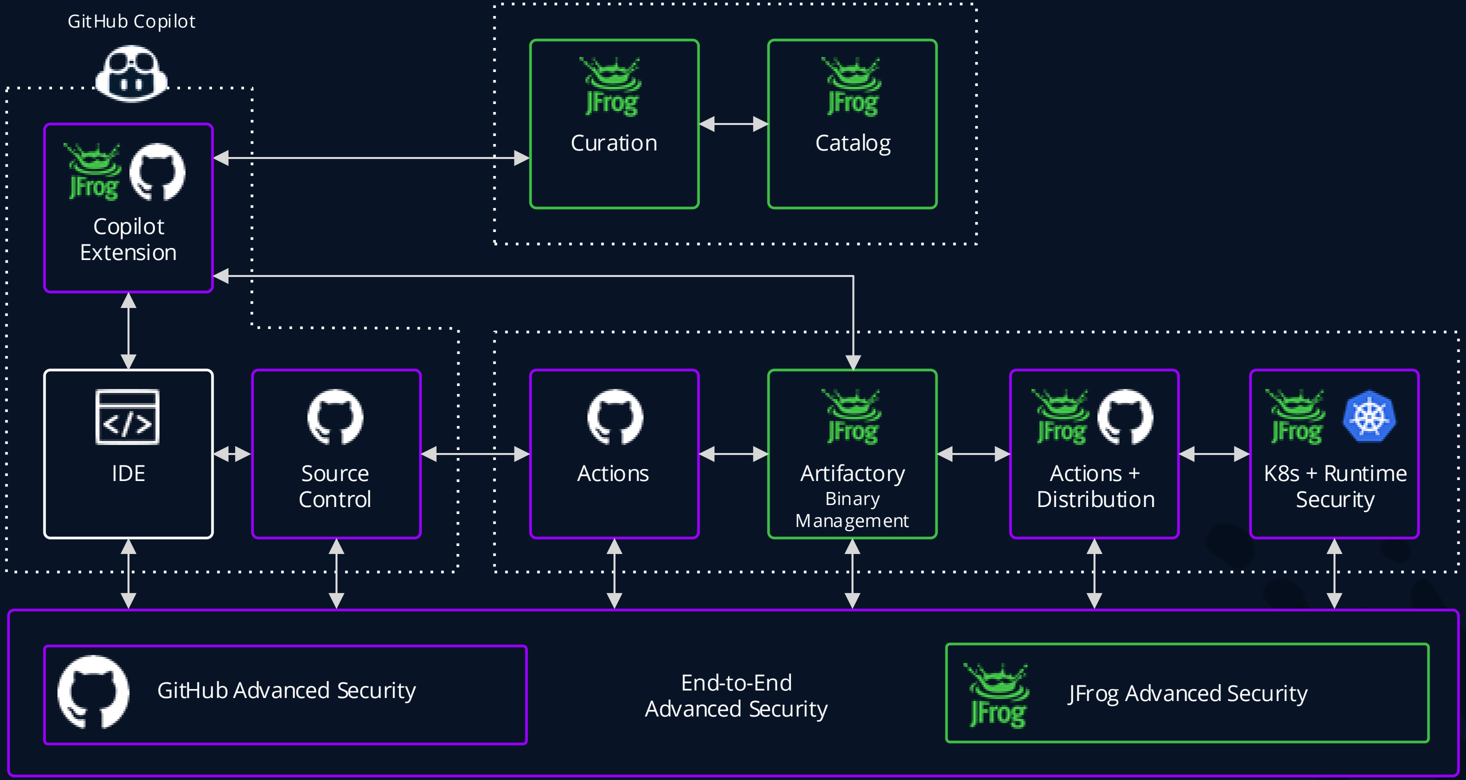
Runtime Impact

- Workload risk prioritization and traceability
- Process visibility
- Process risk prioritization and traceability

Support

- Cloud (AWS, Azure, GCP), REST API, and SH Kubernetes deployments
- Tech/Language: OS Binaries, Go and Java

JFrog & Github Integration



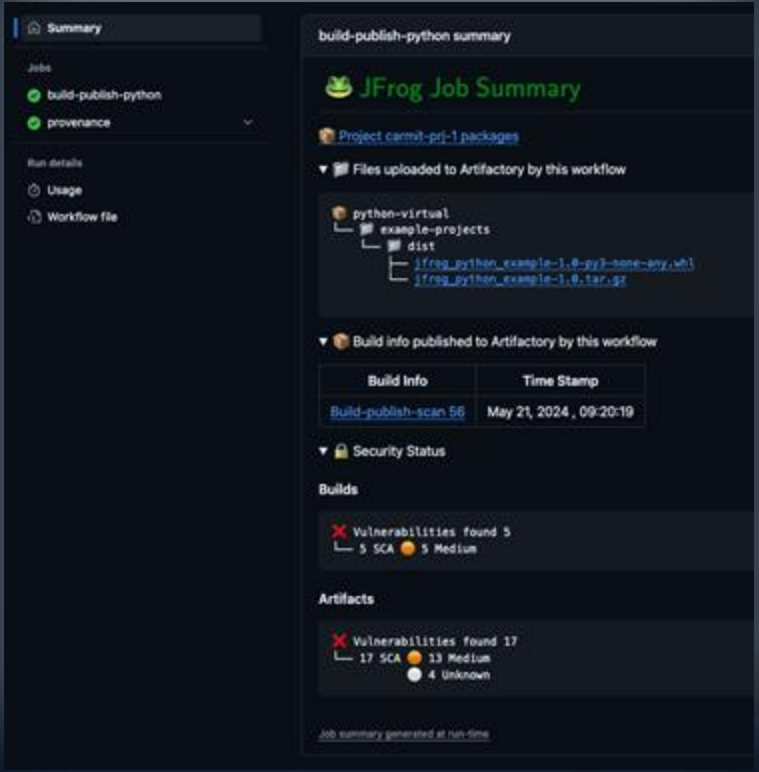
Seamless Code & Binary Development Experience

- **Unified Software Supply Chain Experience**
  Unify SSO, roles, and projects across platforms - swift context switching

- **Bridge Code, Actions, and Binaries**
  Trace production binaries back to their code source, simplifying compliance and provenance

- **Code to Cloud Contextualized Security**
  Code and binary security results in one place with unified security scans shared between platforms

- **AI-Powered Software Supply Chain**
  Extend Copilot interactions to the package and binary level across the SDLC including security insights (Private Beta)



Summary
Jobs
build-publish-python
provenance
Run details
Usage
Workflow file

build-publish-python summary

JFrog Job Summary

Project carmit-prj-1 packages

▼ Files uploaded to Artifactory by this workflow

```
python-virtual
└── example-projects
    └── dist
        ├── jfrog_python_example-1.0-py3-none-any.whl
        └── jfrog_python_example-1.0.tar.gz
```

▼ Build info published to Artifactory by this workflow

Build Info	Time Stamp
Build-publish-scan 56	May 21, 2024 , 09:20:19

▼ Security Status

Builds

```
❌ Vulnerabilities found 5
└── 5 SCA 🟠 5 Medium
```

Artifacts

```
❌ Vulnerabilities found 17
└── 17 SCA 🟠 13 Medium
           ⚪ 4 Unknown
```

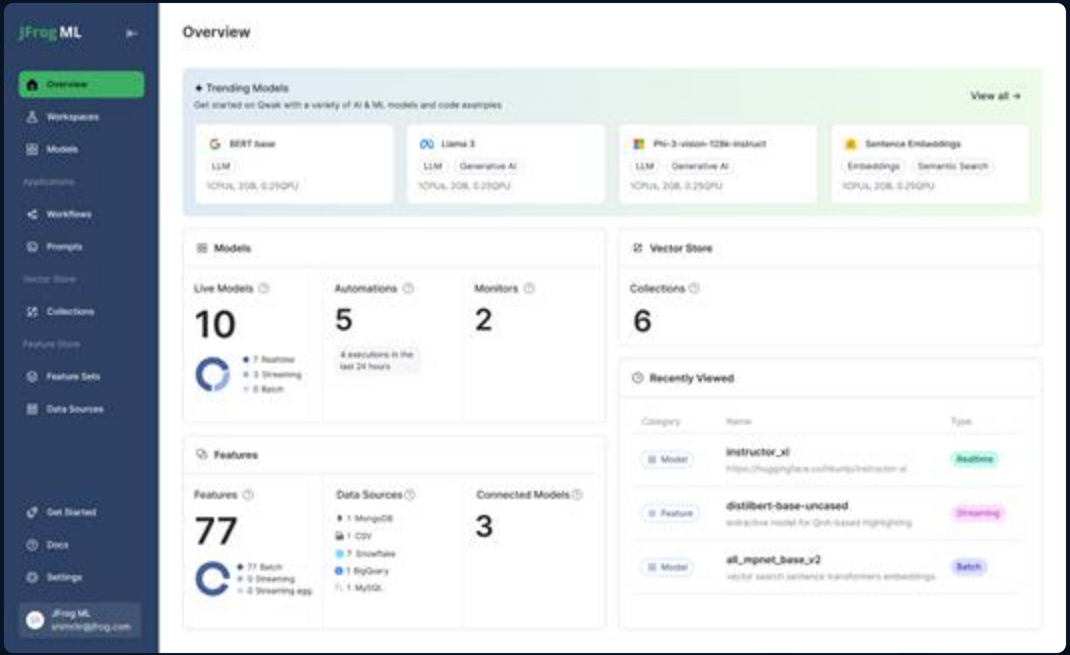
Job summary generated at run-time

JFROG ML

Full E2E Model Workflow

- from development to runtime -

connected to the JFrog Platform



JFrog ML

Overview
Workspaces
Models
Applications
Workflows
Prompts
Vector Store
Collections
Feature Store
Feature Sets
Data Sources
Get Started
Docs
Settings
JFrog ML
shimrite@jfrog.com

Overview

✦ Trending Models

Get started on Qwak with a variety of AI & ML models and code examples

View all →

Model	Tags	Resources
BERT base	LLM	1CPUs, 2GB, 0.25GPU
Llama 3	LLM, Generative AI	1CPUs, 2GB, 0.25GPU
Phi-3-vision-128k-instruct	LLM, Generative AI	1CPUs, 2GB, 0.25GPU
Sentence Embeddings	Embeddings, Semantic Search	1CPUs, 2GB, 0.25GPU

Models

Live Models: 10
- 7 Realtime
- 3 Streaming
- 0 Batch

Automations: 5
4 executions in the last 24 hours

Monitors: 2

Vector Store

Collections: 6

Features

Features: 77
- 77 Batch
- 0 Streaming
- 0 Streaming agg

Data Sources:
- 1 MongoDB
- 1 CSV
- 7 Snowflake
- 1 BigQuery
- 1 MySQL

Connected Models: 3

Recently Viewed

Category	Name	Type
Model	instructor_xl https://huggingface.co/hkunlp/instructor-xl	Realtime
Feature	distilbert-base-uncased extractive model for QnA based highlighting	Streaming
Model	all_mpnet_base_v2 vector search sentence transformers embeddings	Batch

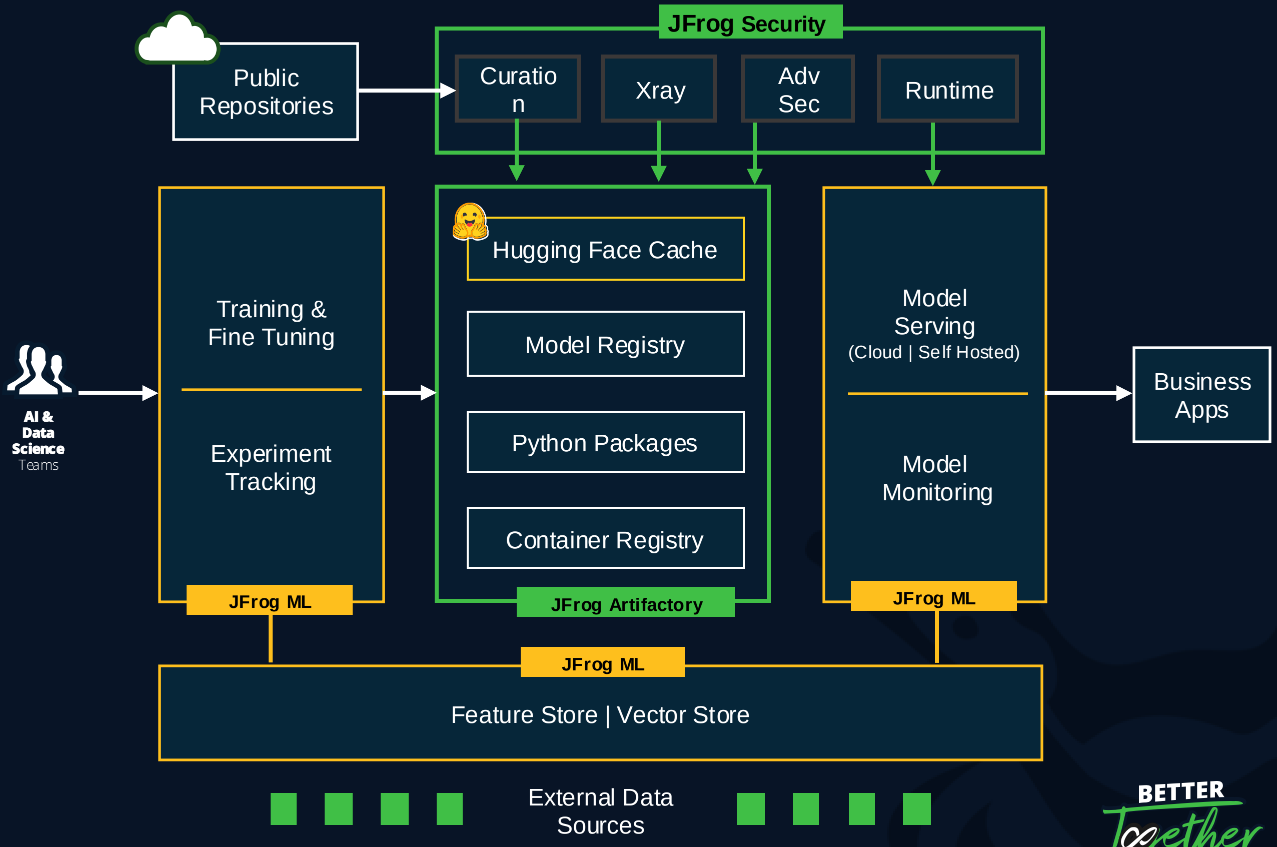
JFrog

BETTER *Together*

JFROG ML: Unified Solution

Take Trusted AI/ML to Production Fast!

- ✓ Model training, deployment, and monitoring
- ✓ Fine tune models and build from scratch
- ✓ LLM applications and prompt engineering
- ✓ Feature lifecycle management



Summary

- It's important to **consolidate** DevSecOps toolset
- Security is a **must** at every step
- Automation of promotion is key to building pipeline **confidence**
- DevSecOps is not one tool, not one approach... it is a **mindset**
- **Come meet us at booth #3 and get a demo**

Start with JFrog

JFrog

Thank you!

Questions?

BETTER Together 2024