



The **LIQUID SOFTWARE** Company

Introduction to JFrog Platform

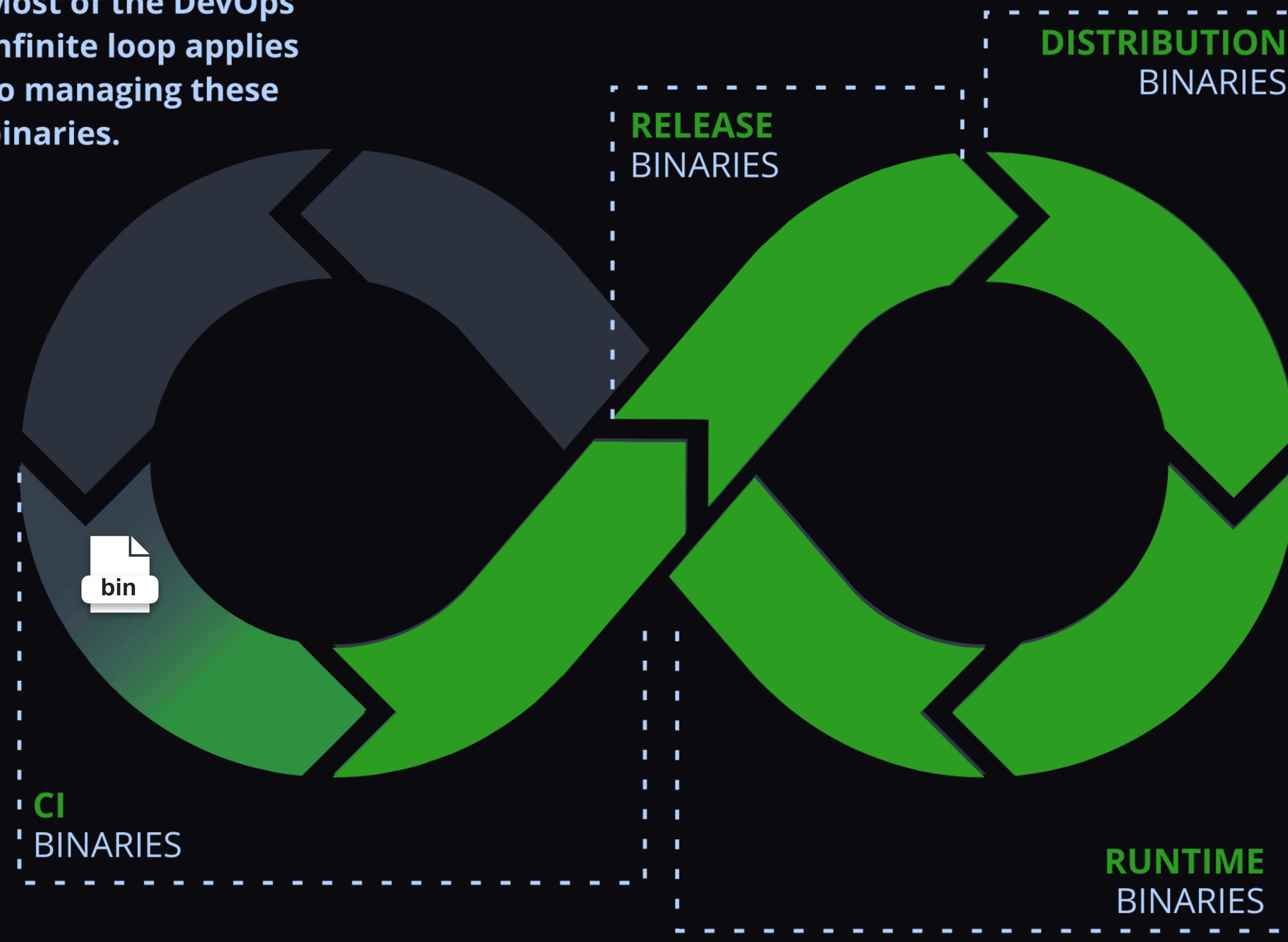


Carlos Moya | Senior Solutions Engineer | cmoya@jfrog.com

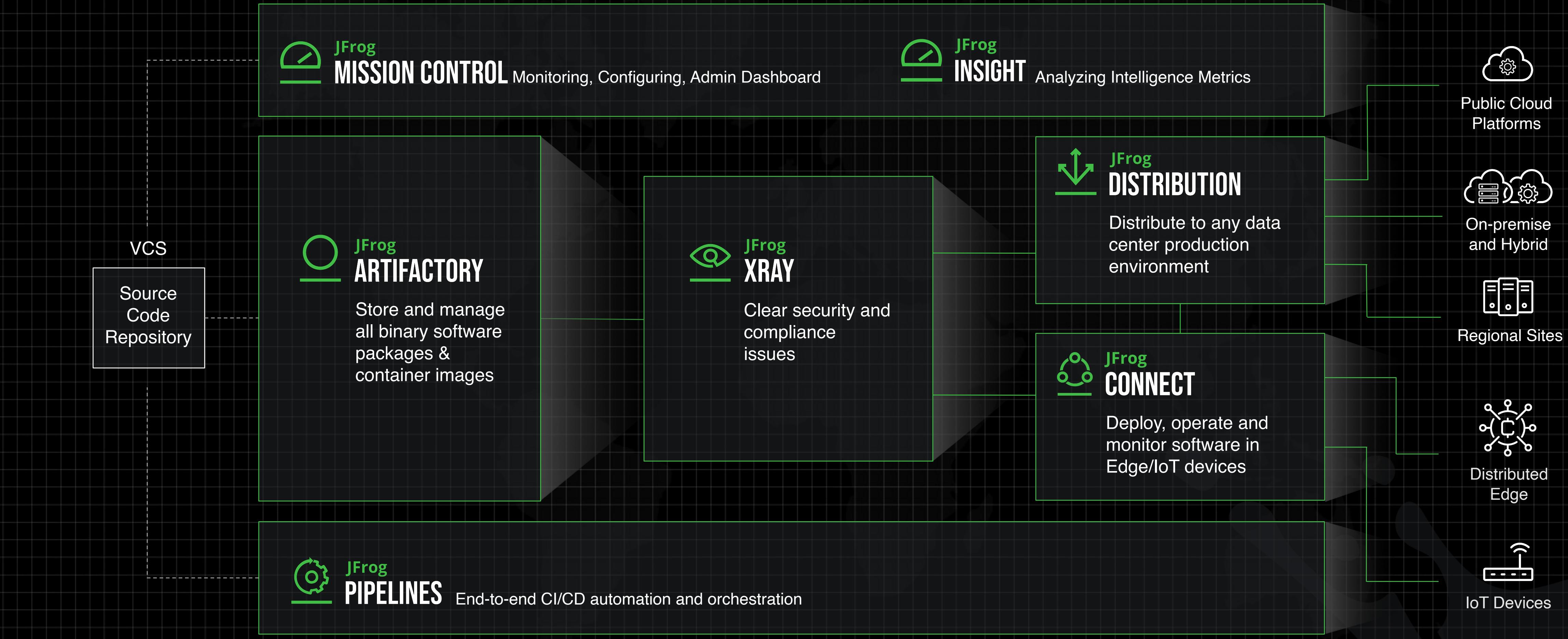
A Bit Of Context

A binary-centric approach to DevOps

Most of the DevOps infinite loop applies to managing these binaries.



Overview of the JFrog Platform



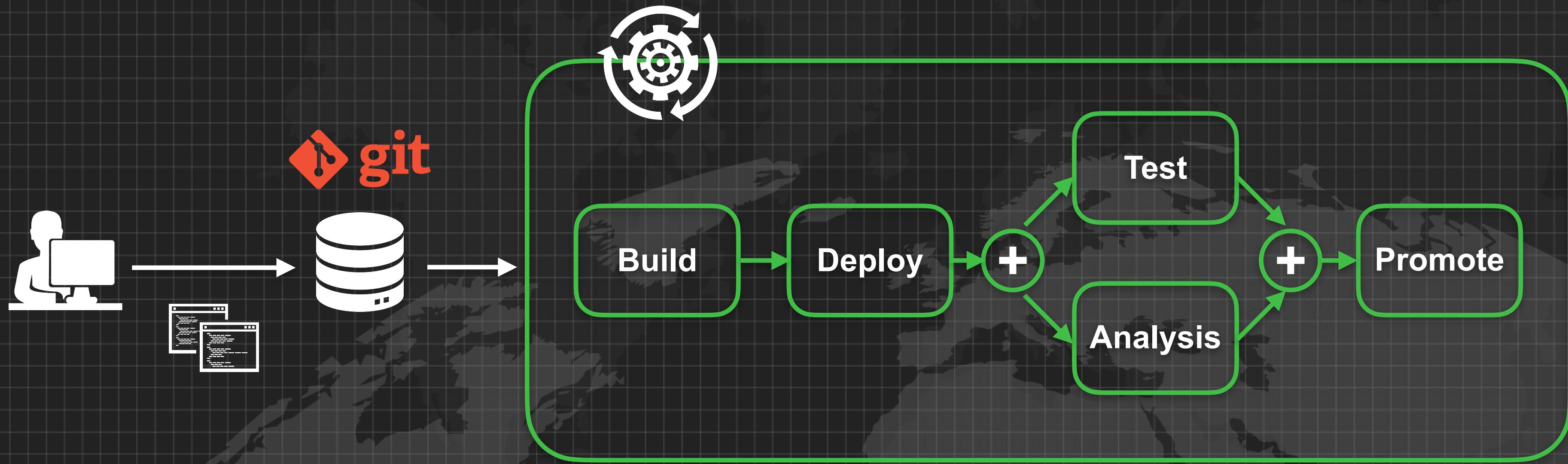
Addressing common challenges with Artifactory

Basic

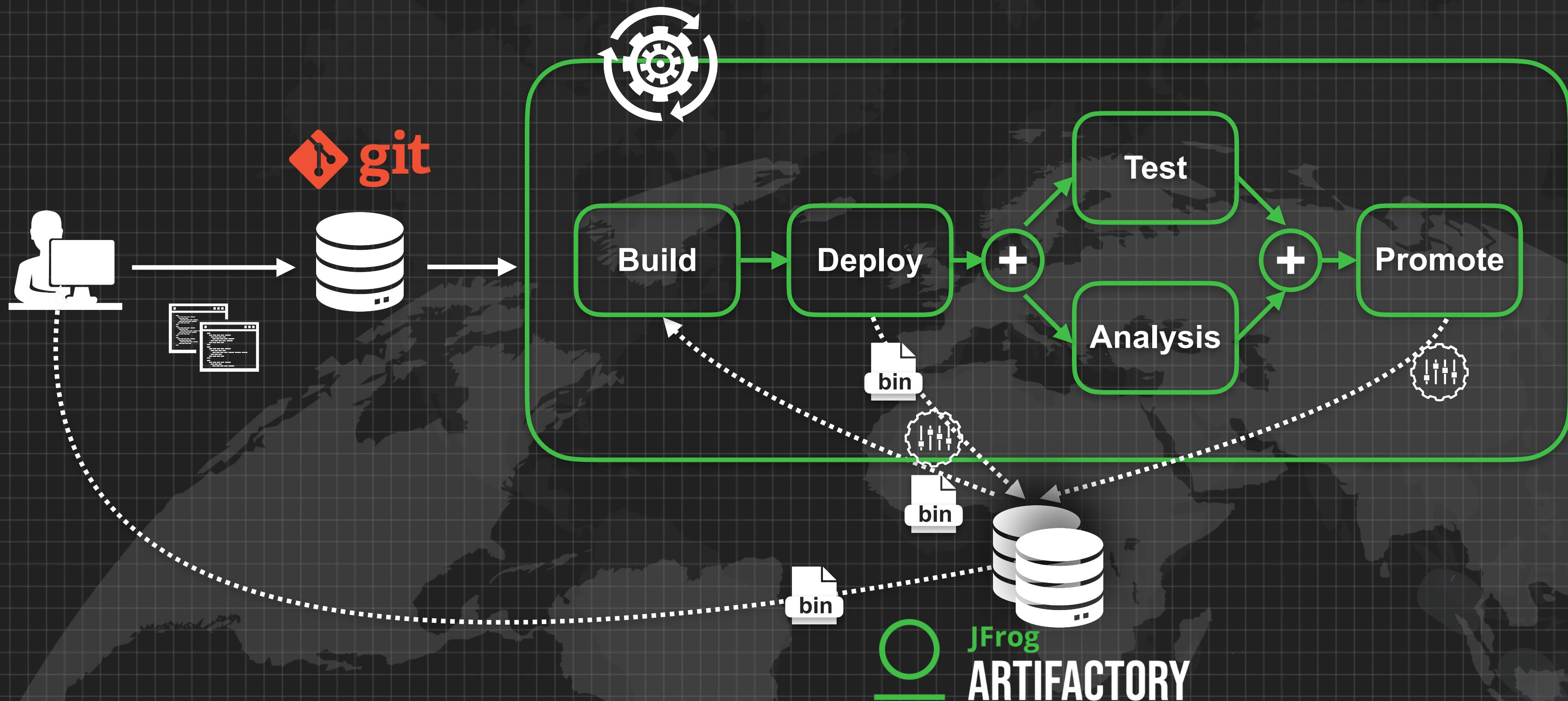
Single Development Team, single location



Single Development Team, single location



Single Development Team, single location



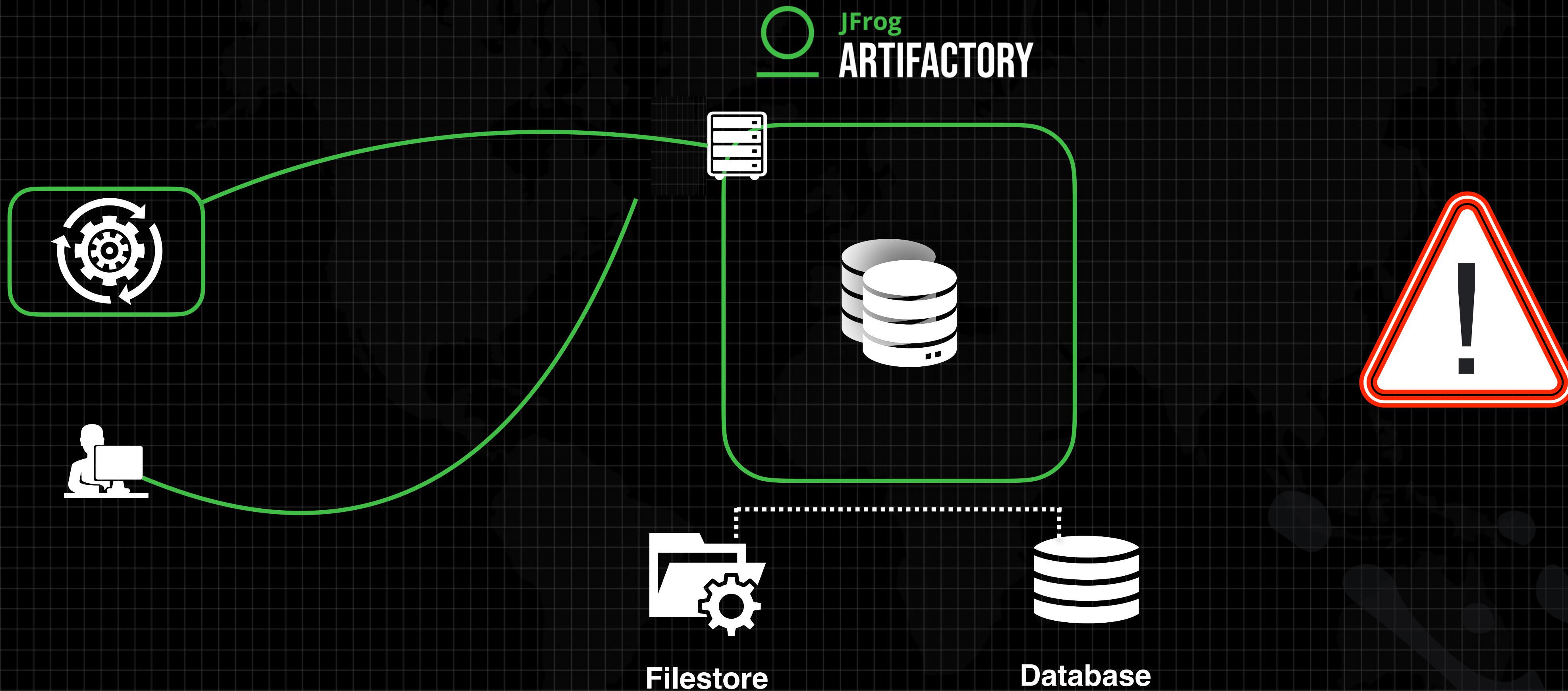
Key features

- Checksum based storage
- Integrates with 30+ package managers
- Full metadata for all supported package formats
- REST API, CLI, AQL, User Plugins
- Built in integrations with automation tools for CI/CD
- Local / Remote / Virtual repositories
- Projects

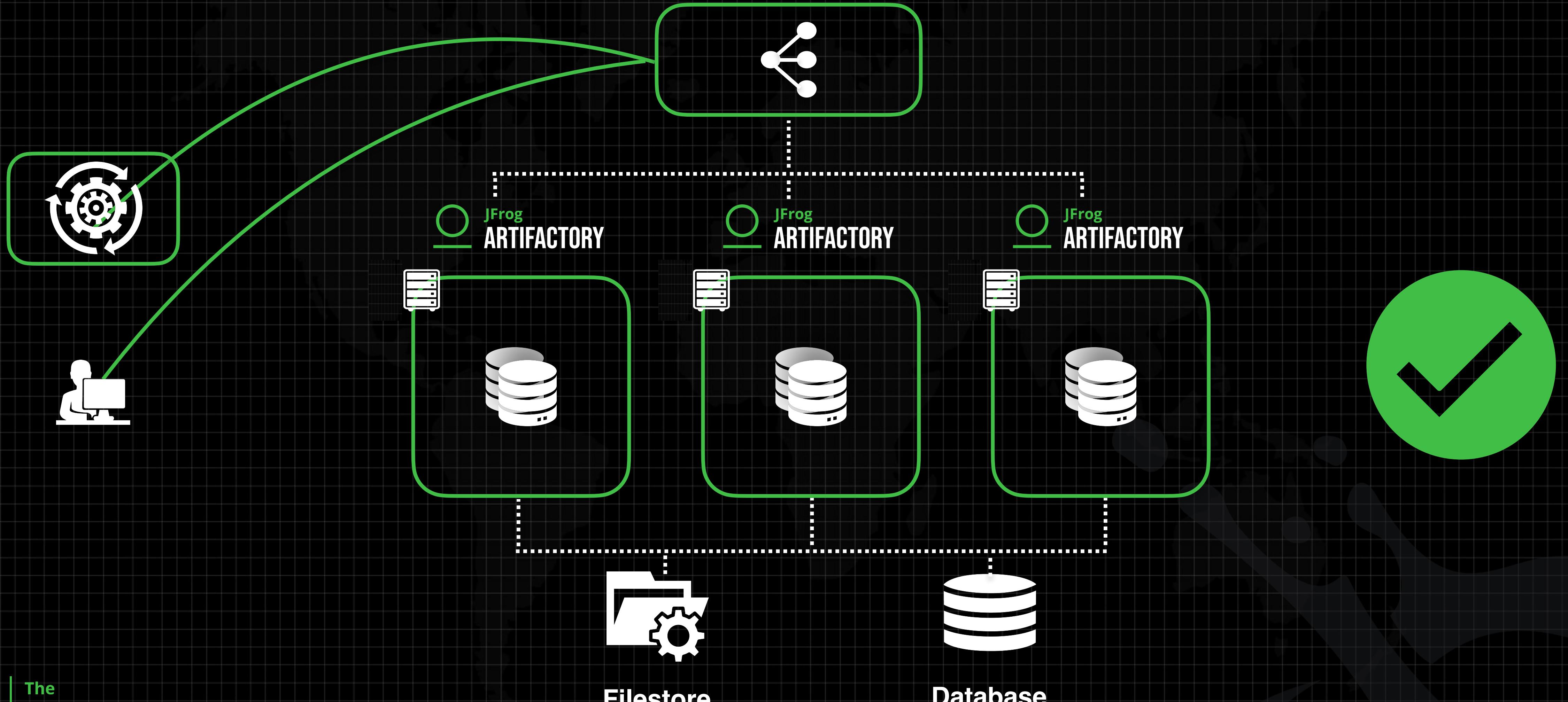
Build Info



Binary management as critical service



Binary management as critical service



Key features

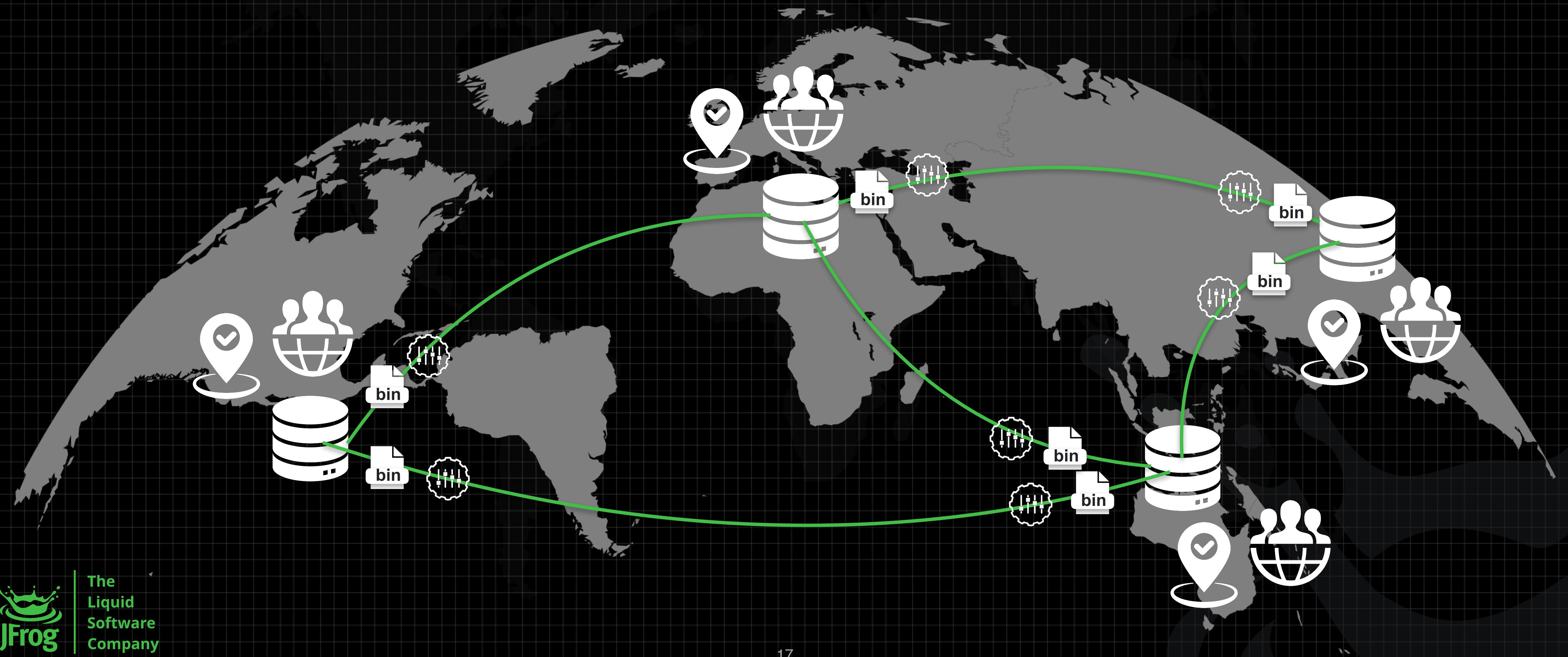
- Zero Downtime Upgrades | Resiliency | Performance
- Read and Write caches
- Filestore sharding
- Cloud storage support

Advanced

Many Development Teams, spread across the globe



Many Development Teams, spread across the globe

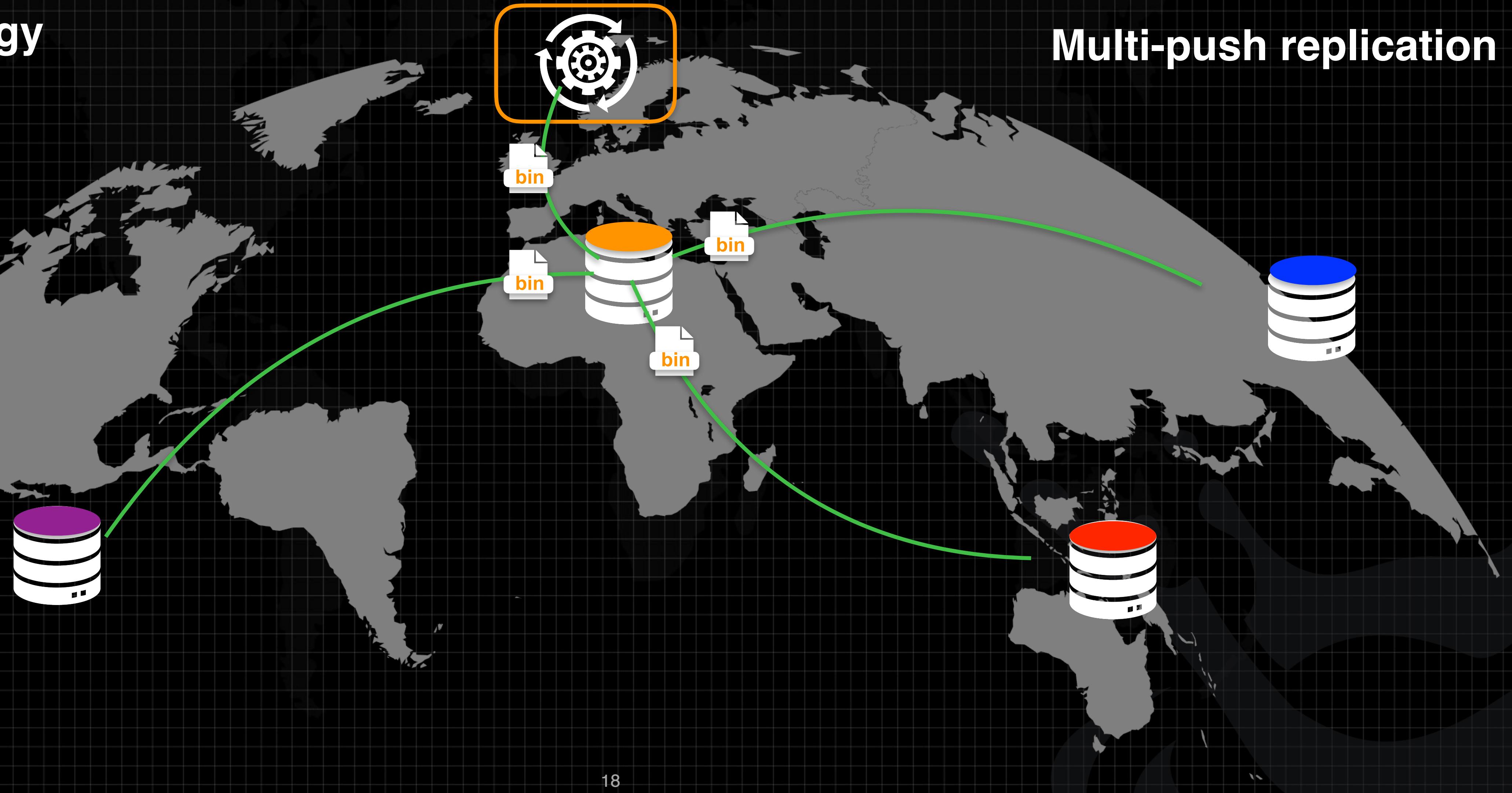


Many Development Teams, spread across the globe

One central CI / CD Server

Star Topology

Multi-push replication

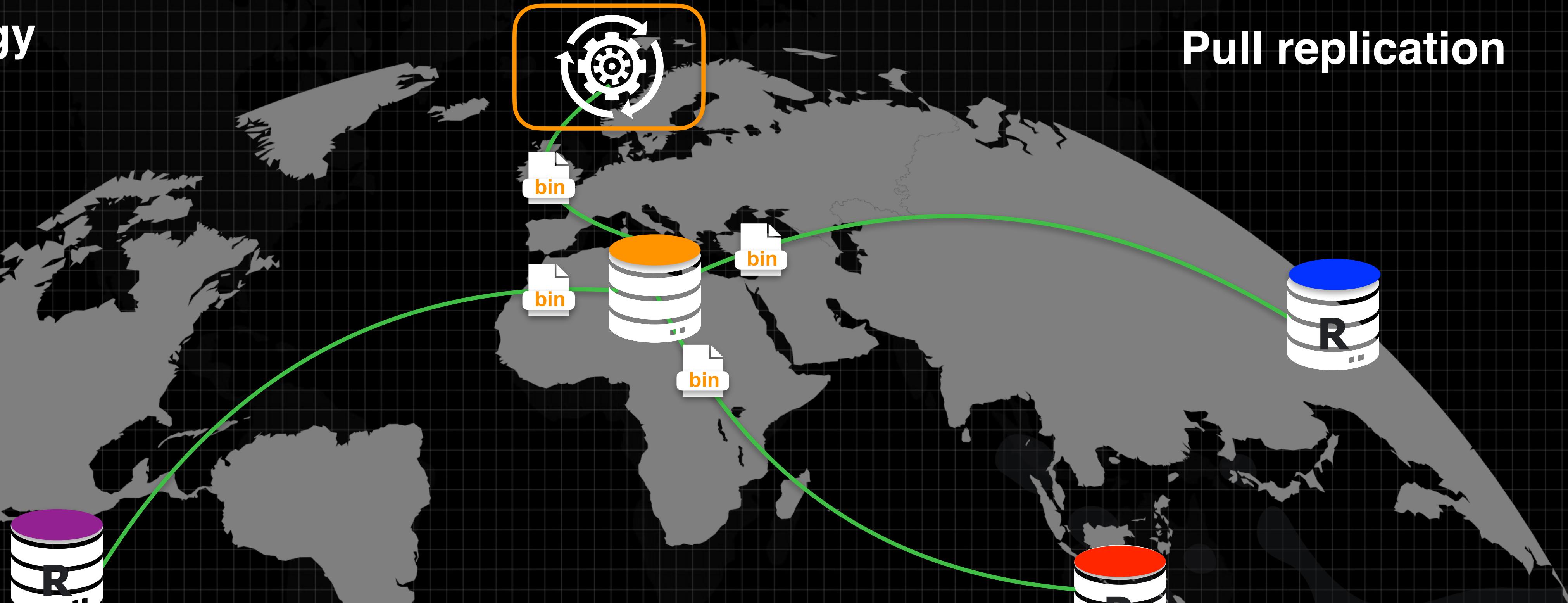


Many Development Teams, spread across the globe

One central CI / CD Server

Star Topology

- Local
- Remote



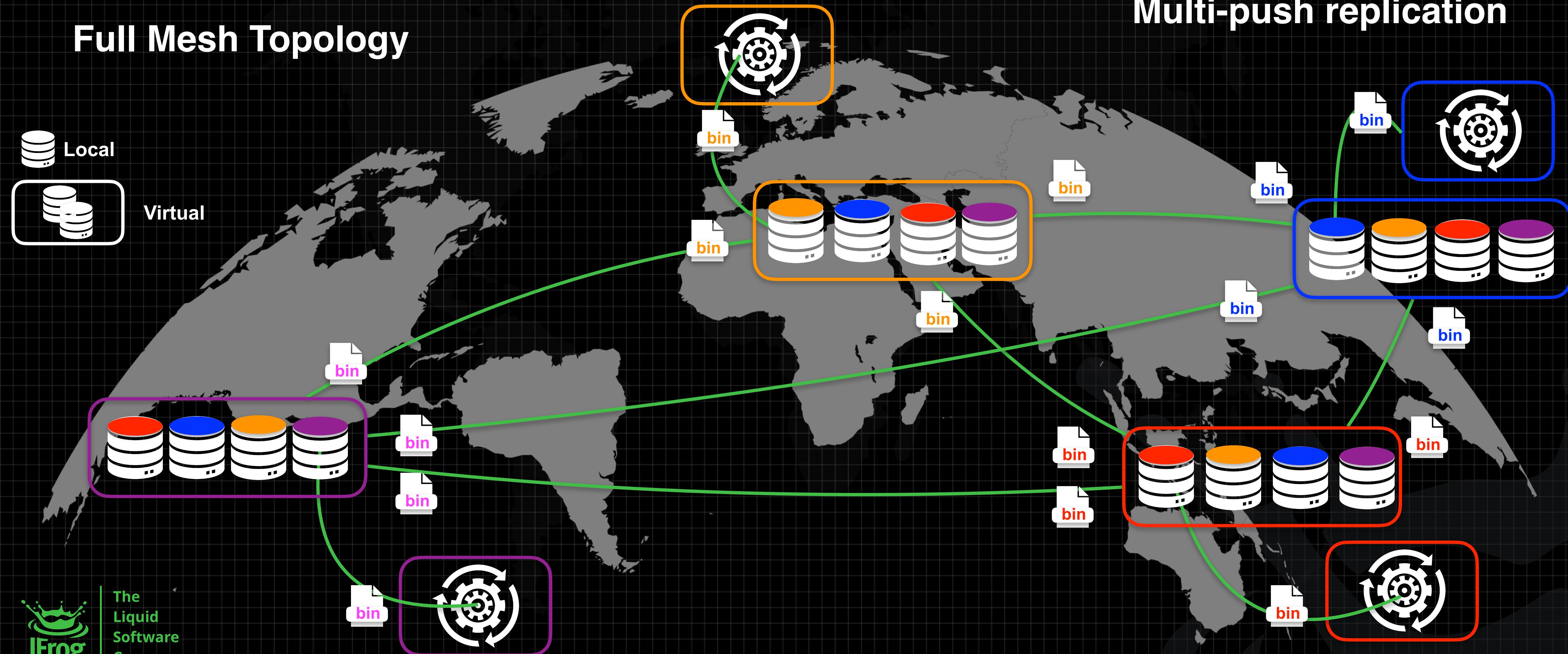
Pull replication

Many Development Teams, spread across the globe

Multiple CI servers

Full Mesh Topology

Multi-push replication



Many Development Teams, spread across the globe

Multiple CI servers

Full Mesh Topology

Local

Remote

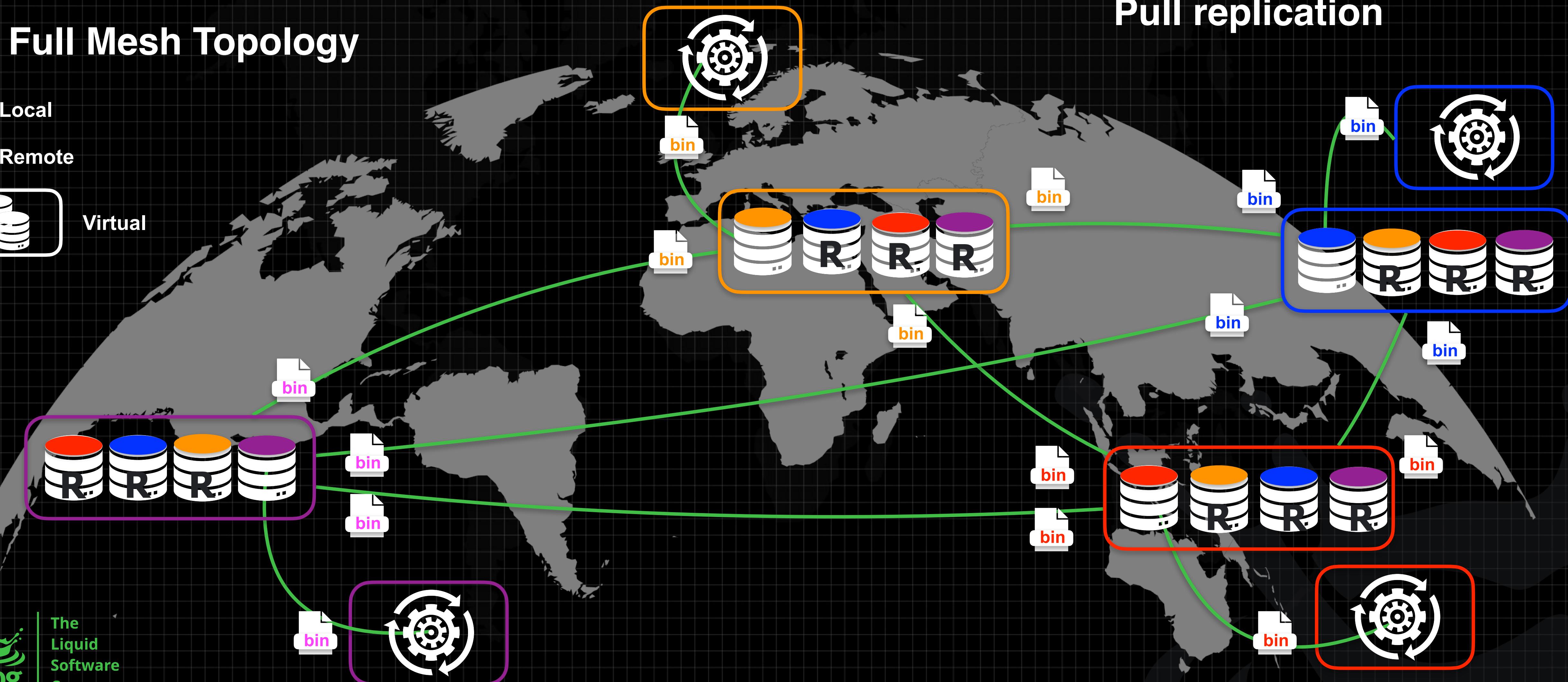
Virtual

R

The
Liquid
Software
Company

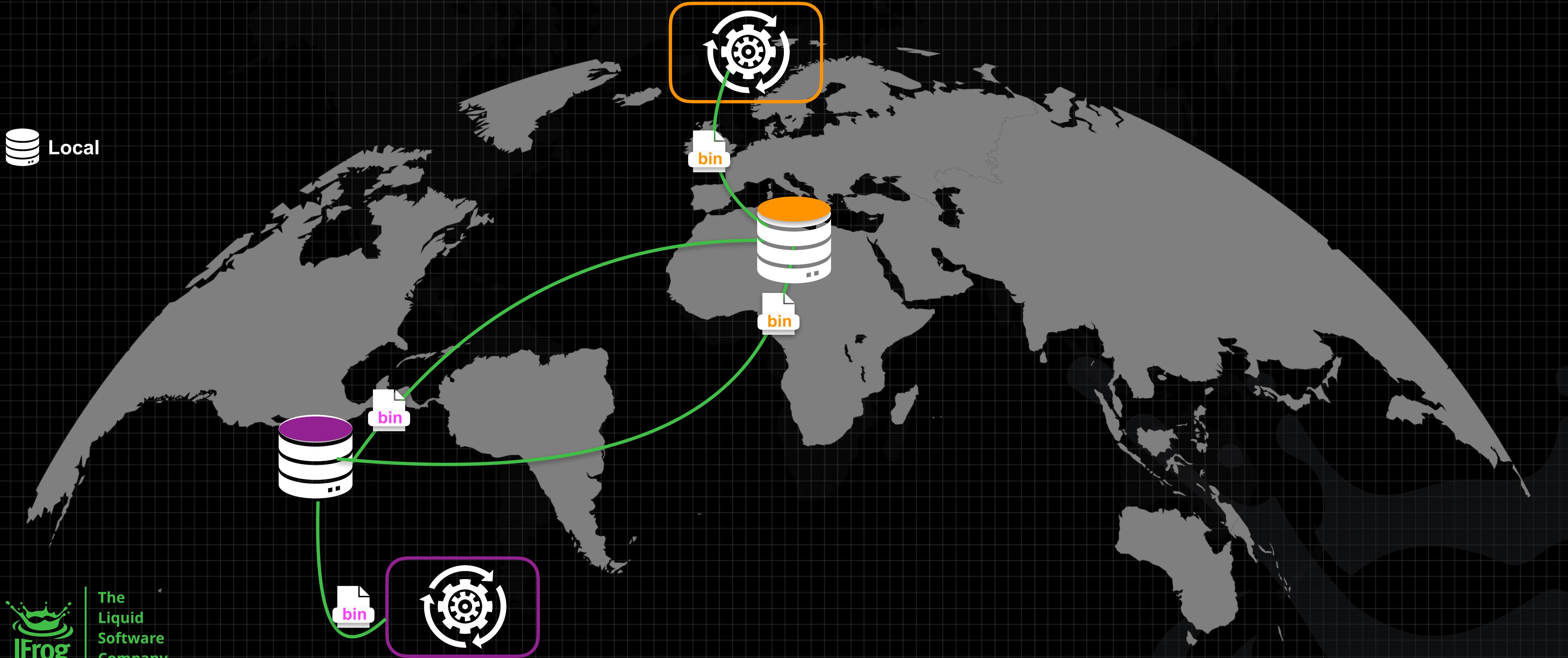
JFrog

Pull replication



Many Development Teams, spread across the globe

Federated repositories



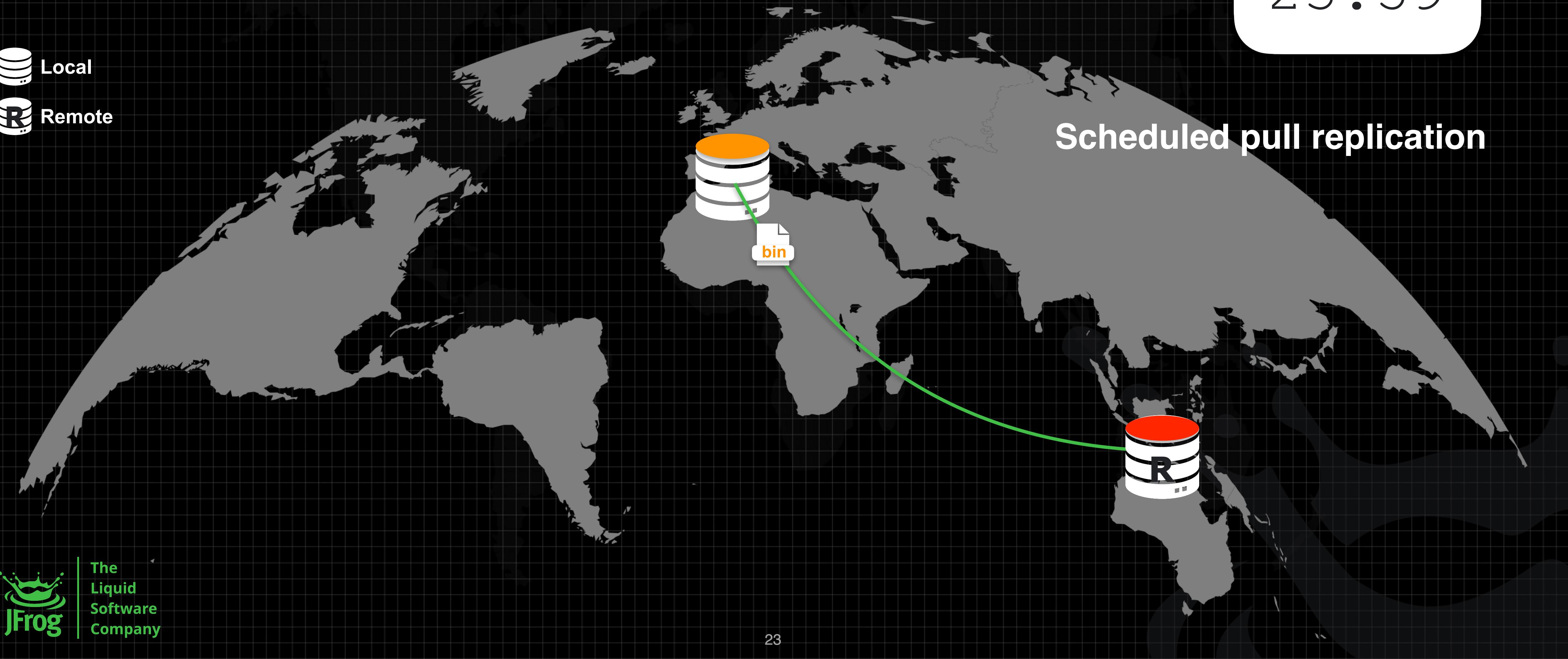
Many Development Teams, spread across the globe

Limited bandwidth

23 : 59

- Local
- Remote

Scheduled pull replication

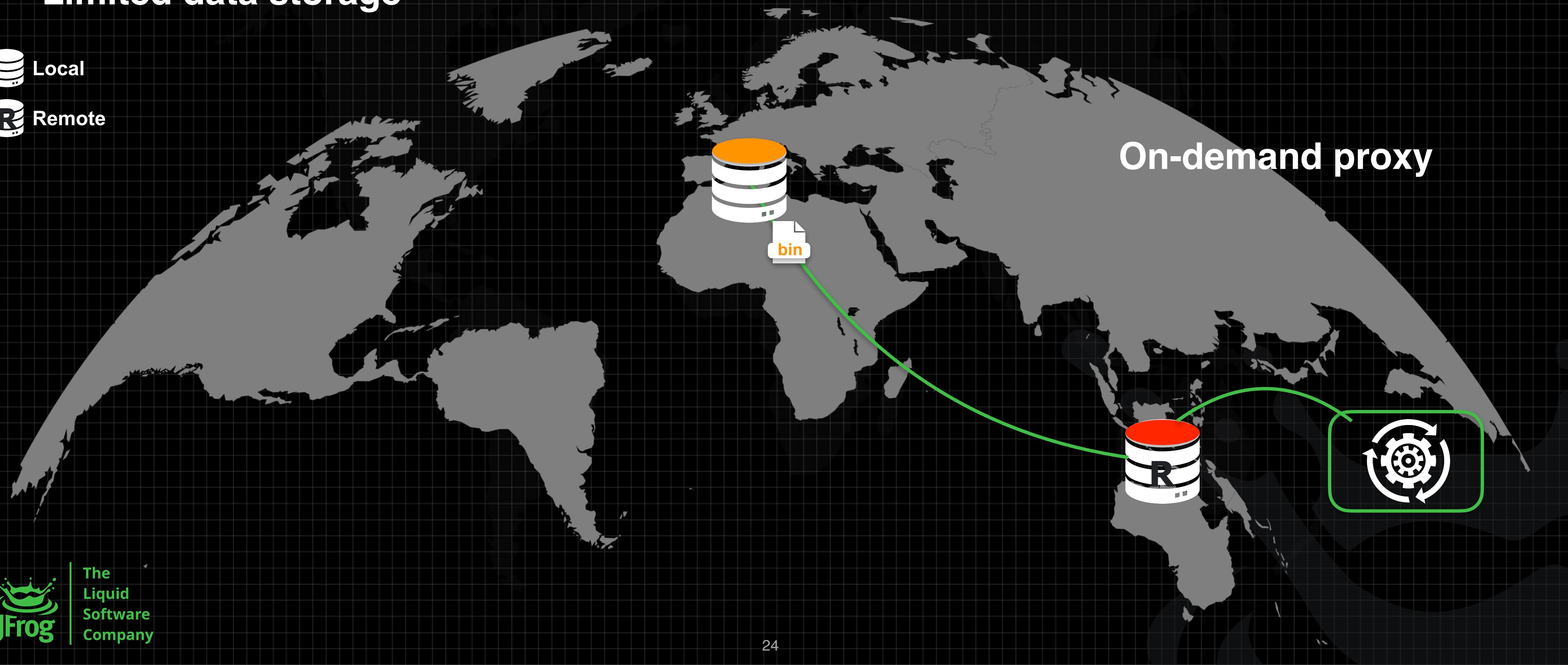


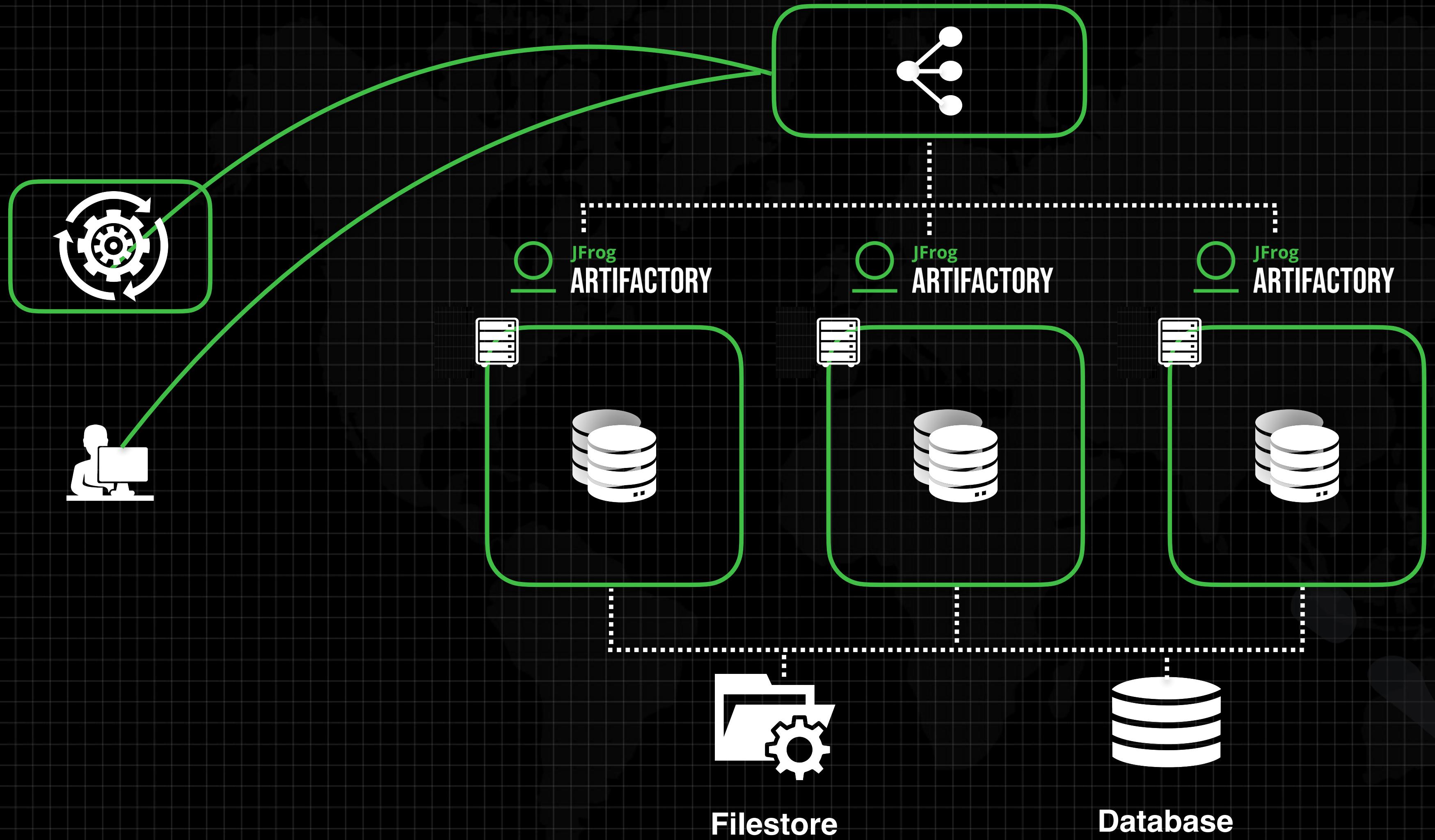
Many Development Teams, spread across the globe

Limited data transfer

Limited data storage

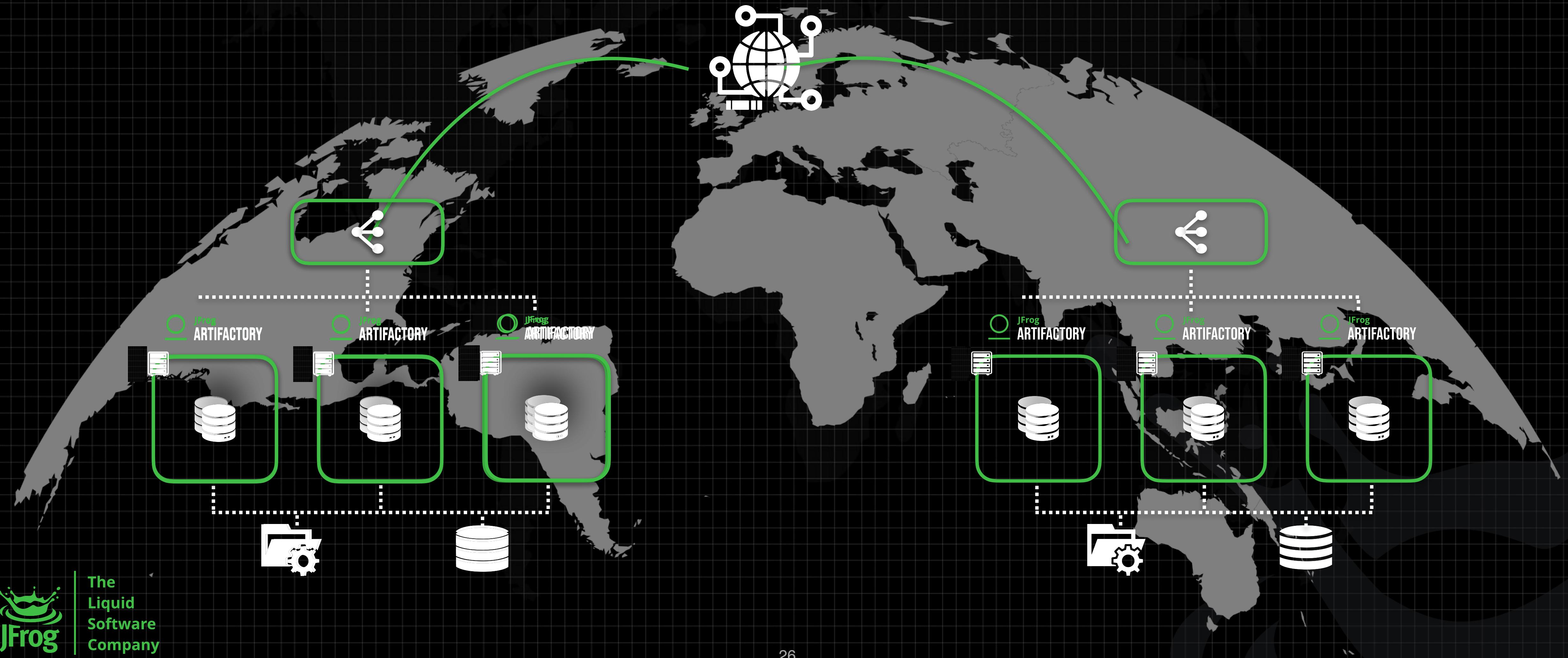
- Local
- Remote





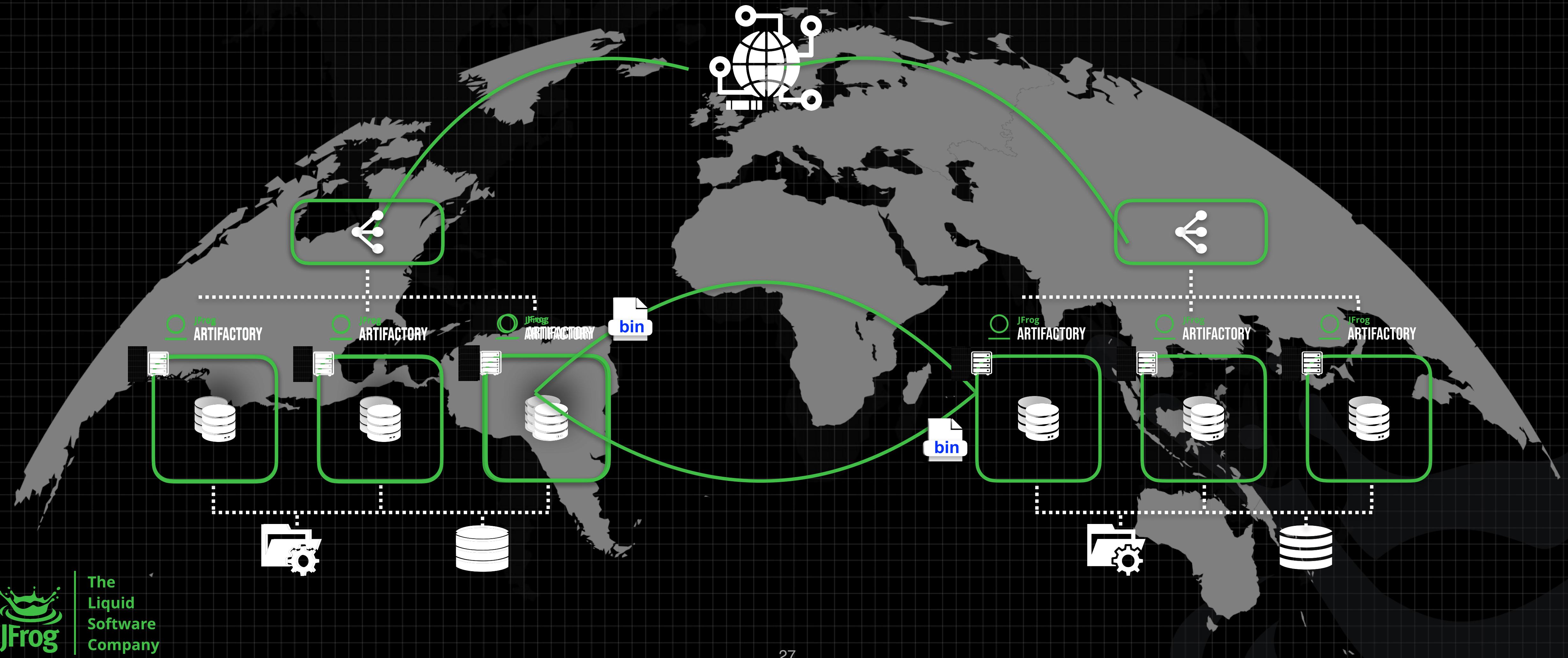
Many Development Teams, spread across the globe

Geo DNS Architecture



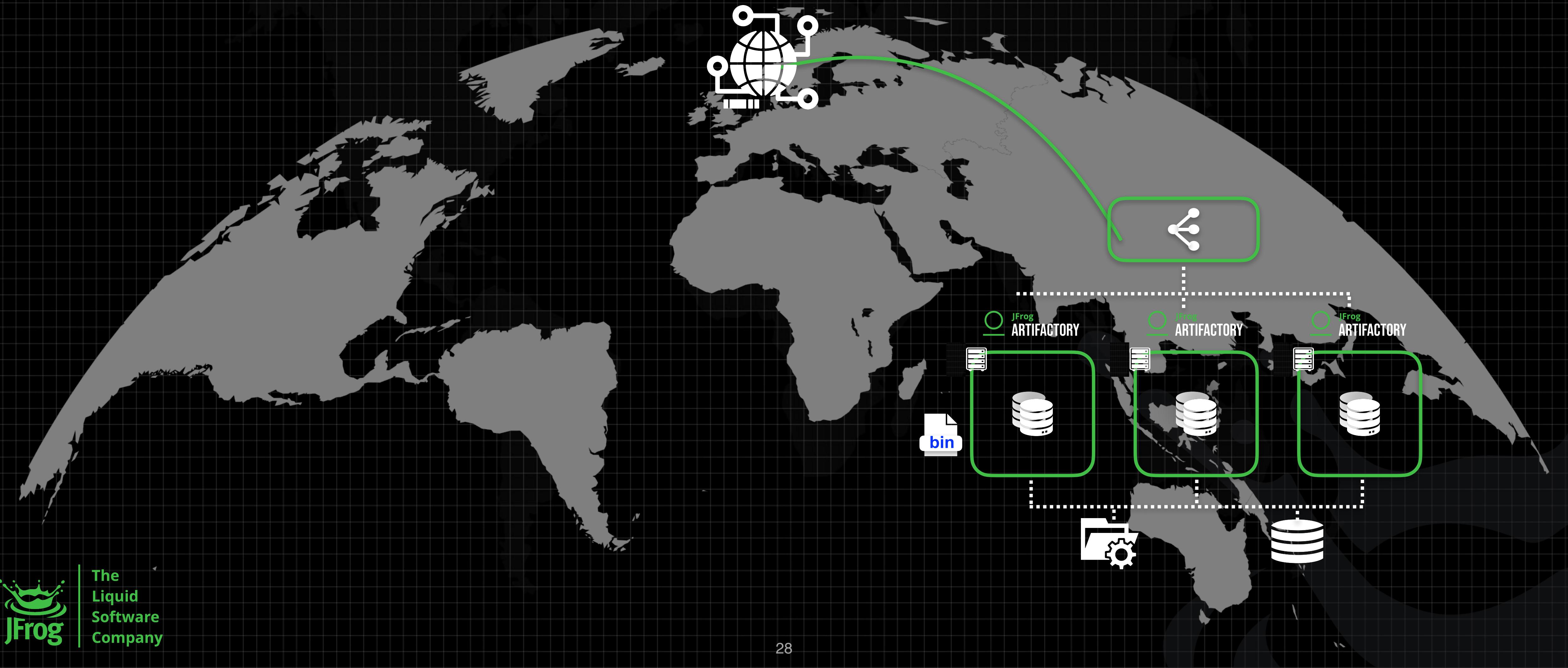
Many Development Teams, spread across the globe

Geo DNS Architecture



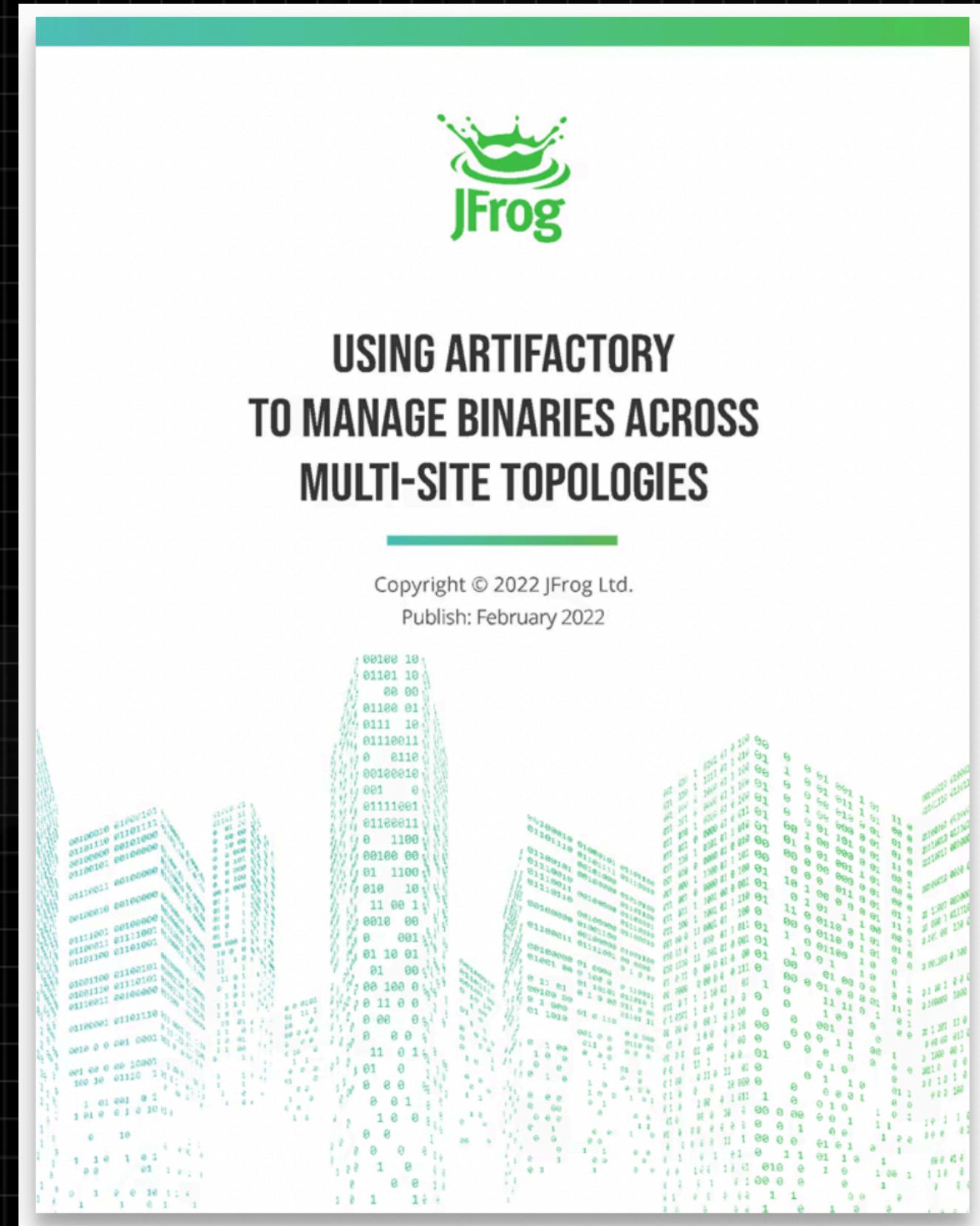
Many Development Teams, spread across the globe

Geo DNS Architecture



Useful links

<https://jfrog.com/whitepaper/replication-using-artifactory-to-manage-binaries-across-multi-site-topologies/>

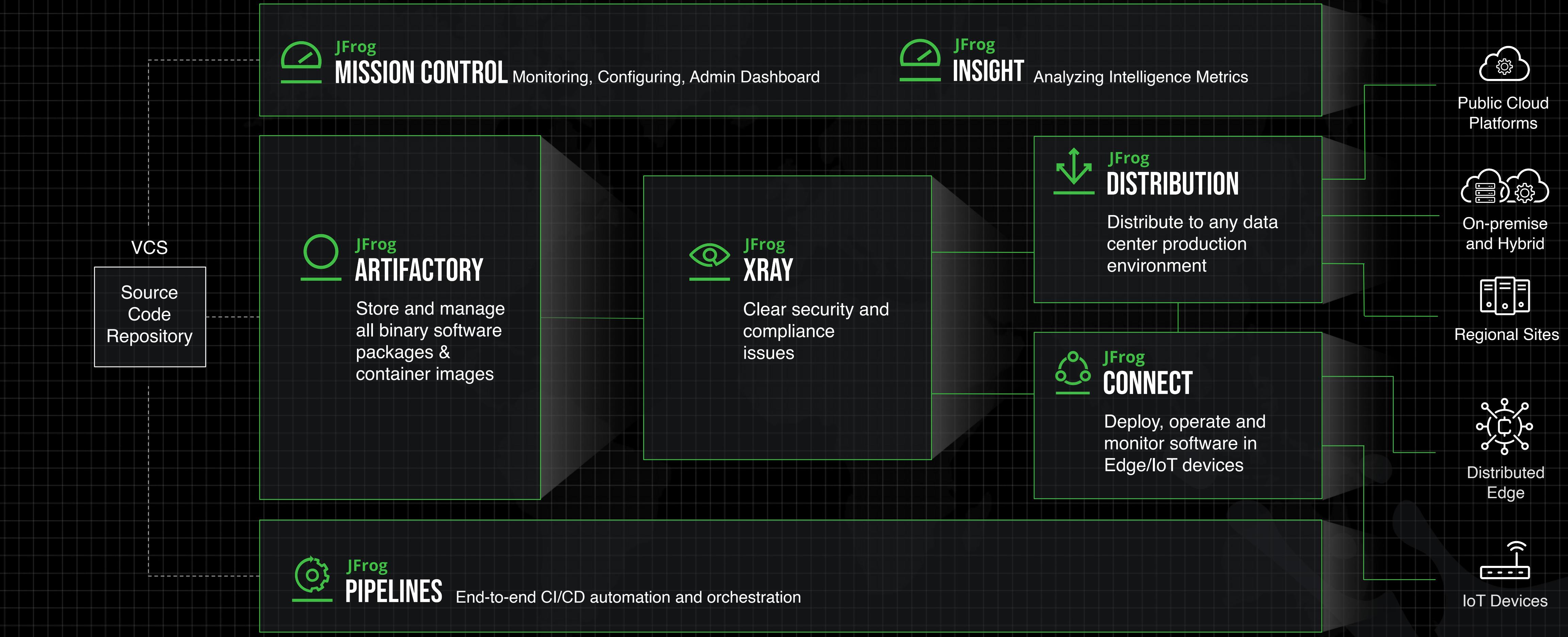




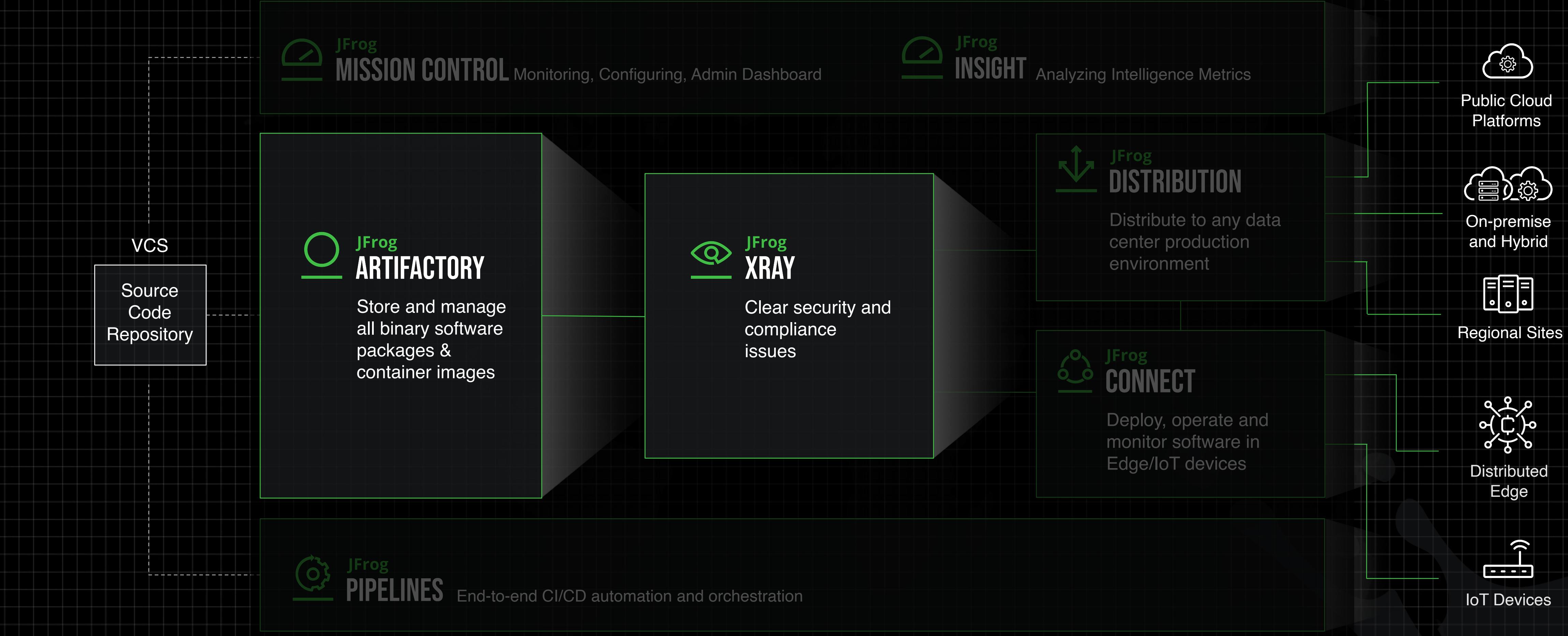
Securing your binary supply chain with Xray

Introduction Xray

Overview of the JFrog Platform

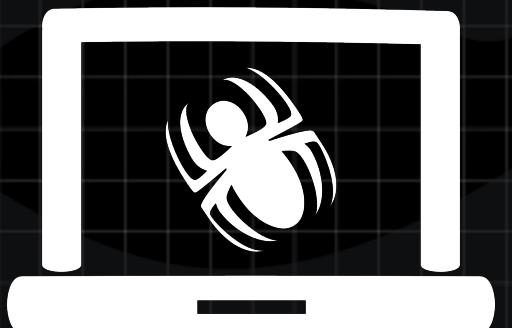
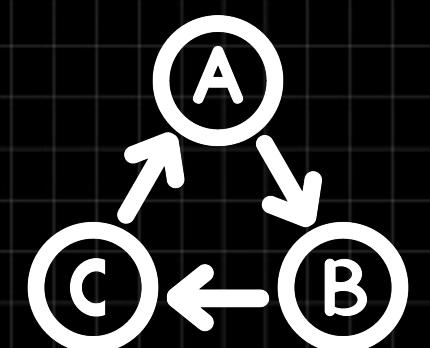


Overview of the JFrog Platform

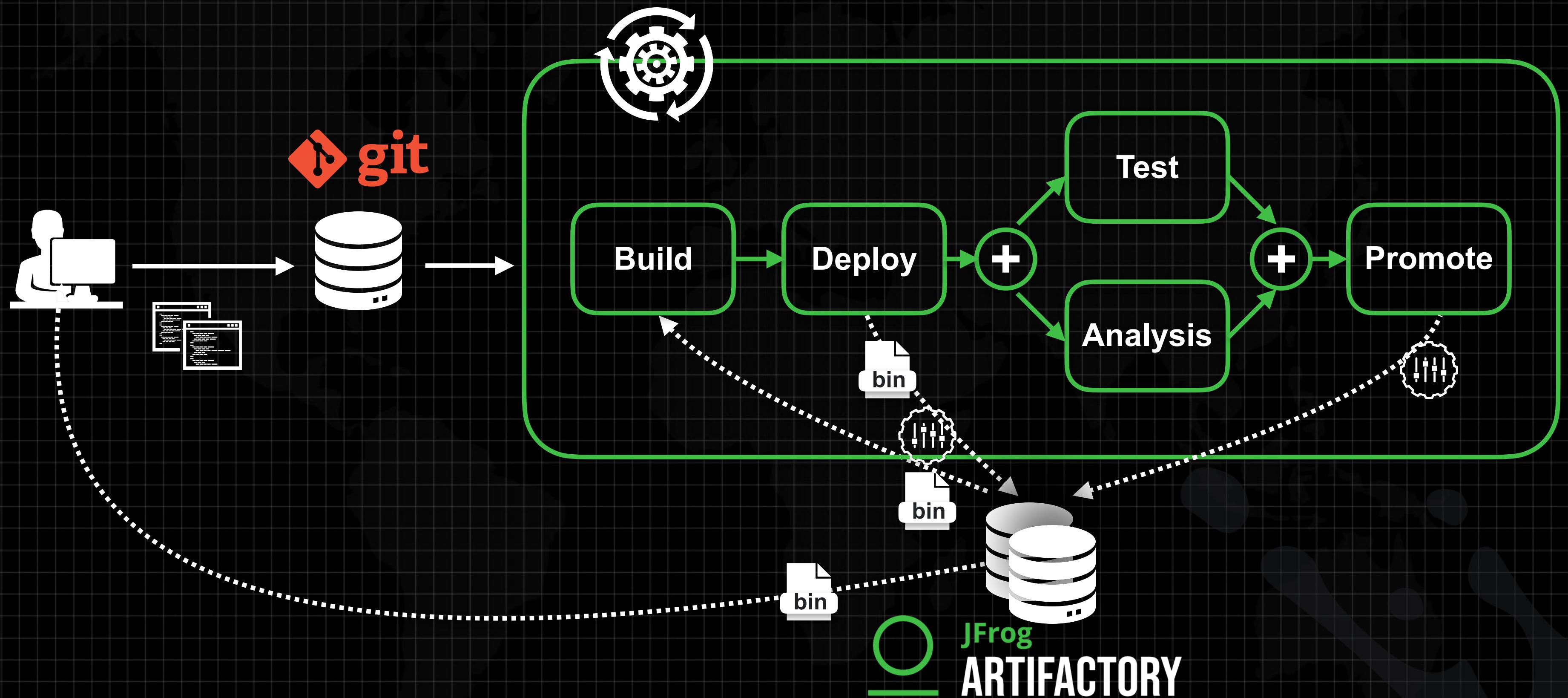


What is the need

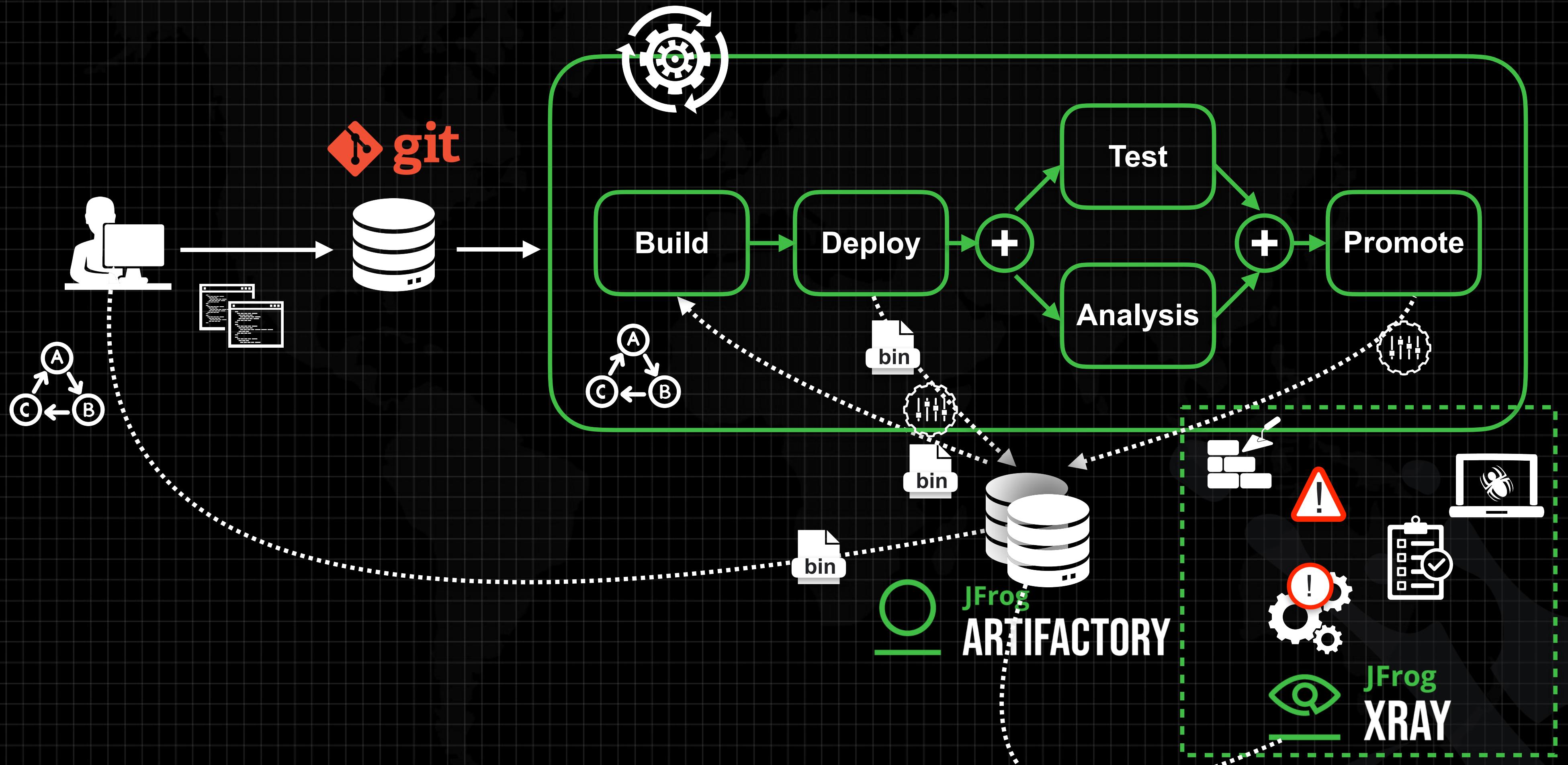
- “ In an average application, **85 -90%** of the codebase is open source
- “ **49%** of codebases analysed had at least one component with a **high-risk vulnerability**
- “ **90%** of applications used at least one **open source** component that was **out-of-date** by four or more years, or was **abandoned**



What is the need



What is the need



Top Notch Security Research At Your Fingertips

- Certified CNA
- In-Depth Research
- Discover and disclose new OSS vulnerabilities
- Analyze novel attack methods
- Help the community and customers with OSS Tools promptly

<https://research.jfrog.com/>

MALICIOUS PACKAGES IN PYPI PART II: GETTING STEALTHIER
LEARN MORE >

HEAD-TO-HEAD:
PENETRATION TESTING VS VULNERABILITY SCANNING
~1,000 FINDINGS PUBLISHED
LEARN MORE >

LARGE SCALE NPM ATTACK TARGETS AZURE DEVELOPERS WITH MALICIOUS PACKAGES
LEARN MORE >

7 NEW VULNERABILITIES FOUND IN CLICKHOUSE DBMS
LEARN MORE >
1,430+ MALICIOUS PACKAGES DISCOVERED

NEW HTTP SMUGGLING VULNERABILITY IN HAProxy
LEARN MORE >
500+ ZERO-DAY VULNERABILITIES DISCOVERED

5 NEW VULNERABILITIES DISCOVERED IN PJSIP
LEARN MORE >
17 OSS SECURITY TOOLS RELEASED

HOW TO PREVENT THE NEXT LOG4J STYLE ZERO-DAY VULNERABILITY
LEARN MORE >

MALICIOUS PACKAGES IN NPM REGISTRY EXPOSE DISCORD TOKENS
LEARN MORE >

NPM SUPPLY CHAIN ATTACK TARGETING GERMANY-BASED COMPANIES WITH BACKDOOR MALWARE
LEARN MORE >

MALICIOUS PACKAGES IN NPM REGISTRY: MALWARE AUTHORS TARGETED

NETTY VULNERABILITIES EXPOSE APPS TO DOS ATTACK
LEARN MORE >

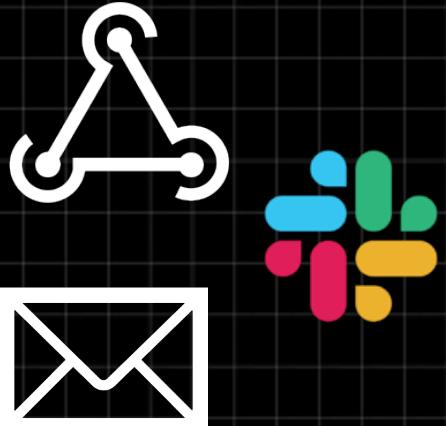
Spring Shell 0-DAY VULNERABILITY ALL YOU NEED TO KNOW
READ MORE >



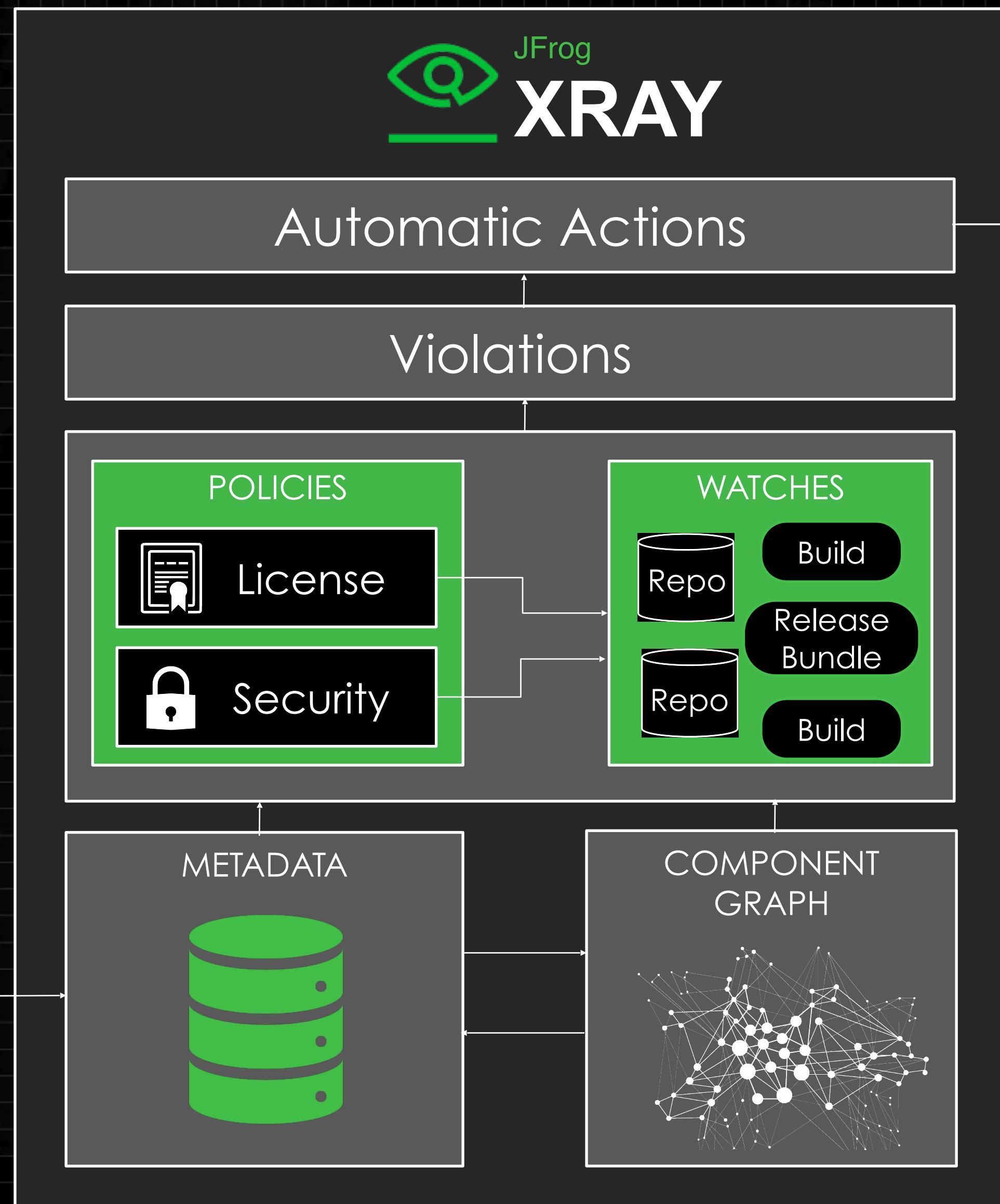
The
Liquid
Software
Company

Xray overview

Web Hooks, Slack, Emails



Prevent
Download



JFrog Xray | Shift left to empower developers



IntelliJ Idea



VSCode



Eclipse



Visual Studio



JFrog CLI



Docker Desktop



The
Liquid
Software
Company

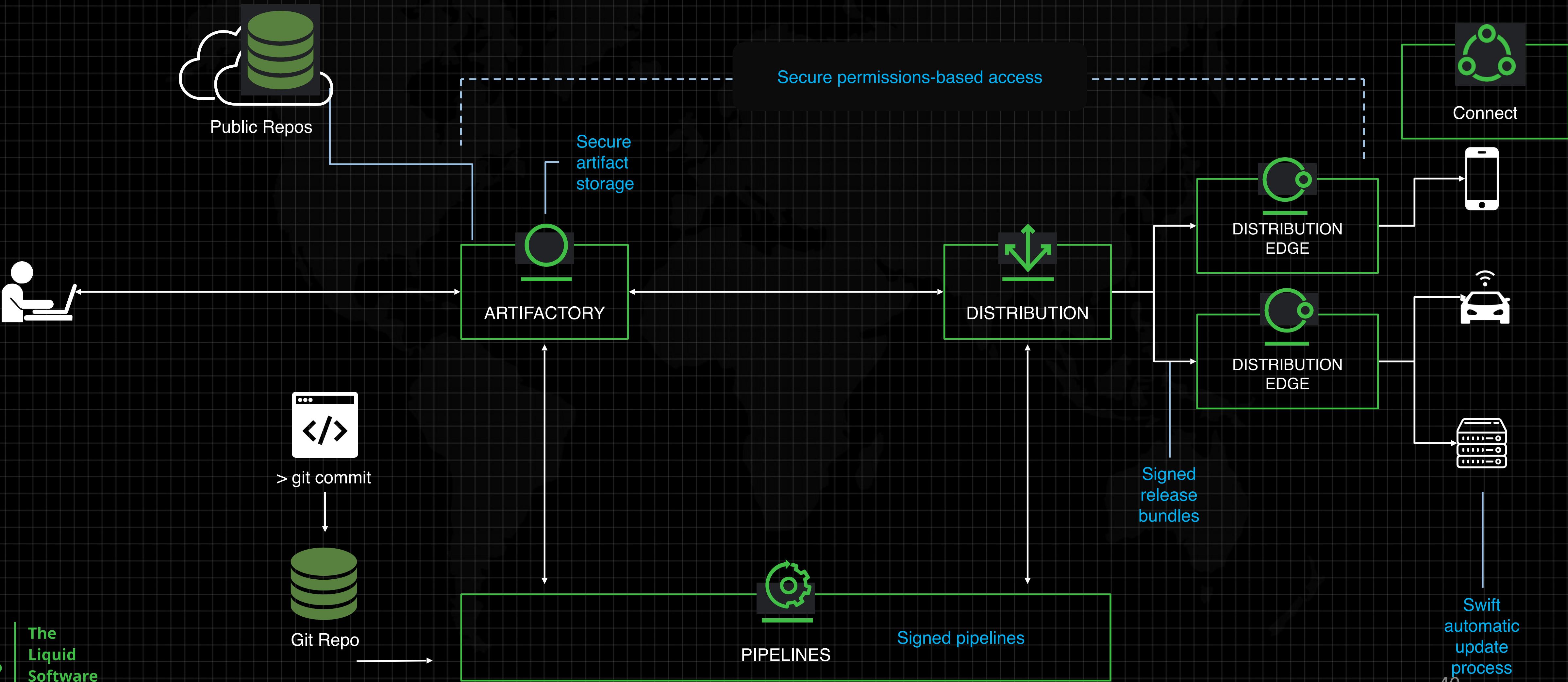
SEVERITY	IMPACTED PACKAGE	IMPACTED PACKAGE VERSION	TYPE	FIXED VERSIONS	COMPONENT	COMPONENT VERSION	CVE	CVSS V2	CVSS V3	ISSUE ID
Critical	json-schema	0.2.3	npm	[0.4.0]	npm	6.14.15	CVE-2021-3807	9.8	7.5	1
High	http-proxy	1.18.1	npm	(,0.0.0)	browser-sync	2.27.5	CVE-2021-3807	9.8	10.0	2
High	ansi-regex	3.0.0	npm	[5.0.1] [6.0.1]	npm	6.14.15	CVE-2021-3807	9.8	7.5	3
High	ansi-regex	4.1.0	npm	[5.0.1] [6.0.1]	npm	6.14.15	CVE-2021-3807	9.8	7.5	4
High	engine.io	3.5.0	npm	[4.0.0]	browser-sync	2.27.5	CVE-2021-3807	9.8	10.0	5
Medium	tar	4.4.19	npm	[6.1.4]	npm	6.14.15	CVE-2021-3807	9.8	7.5	6
Medium	debug	4.1.1	npm	[4.3.1] [3.2.7] [3.1.0] [2.6.9]	browser-sync	2.27.5	CVE-2021-3807	9.8	7.5	7
Medium	tar	4.4.19	npm	[6.1.4]	npm	6.14.15	CVE-2021-3807	9.8	7.5	8

The screenshot shows the JFrog Xray Scan interface. On the left, there's a sidebar with options like Home, Containers, Images, Volumes, Dev Environments (PREVIEW), Extensions (BETA), and JFrog. The main area has tabs for Docker and JFrog. Under Docker, it says "Shift left and run a deep recursive scan for vulnerabilities through all the layers of an image". A dropdown menu shows "local/mypetclinic:latest" and a "Scan" button. To the right, there's a "Vulnerabilities" section with a donut chart showing 489 total vulnerabilities across four severity levels: Critical (red), High (orange), Medium (yellow), and Low (green). Below this is a "Image Scan Results" table with columns: Severity, Impacted Package, Version, Type, Fix Versions, CVE, CVSS 3.0, and CVSS 2.0. The table lists several vulnerabilities found in the Docker image. At the bottom, there's a "Give Feedback" button.

This screenshot shows a detailed view of JFrog Xray's UI. On the left, a "COMPONENTS TREE" panel shows a hierarchical structure of dependencies for an npm project. It includes components like "npm 6.14.15", "browser-sync 2.27.5", and various socket.io and http-proxy versions. To the right, a "COMPONENT DETAILS" panel for "npm" version 6.14.15 provides information such as artifact type, version, scopes, and issue counts. It also lists "Licenses" and "Component Issues Details". The "Component Issues Details" section highlights two specific vulnerabilities: "json-schema lib/validate.js Prototype Pollution Unspecified R..." and "ansi-regex is vulnerable to Inefficient Regular Expression Co...". Each issue is categorized by severity (High) and provides details like component name, CVE, and fix versions.

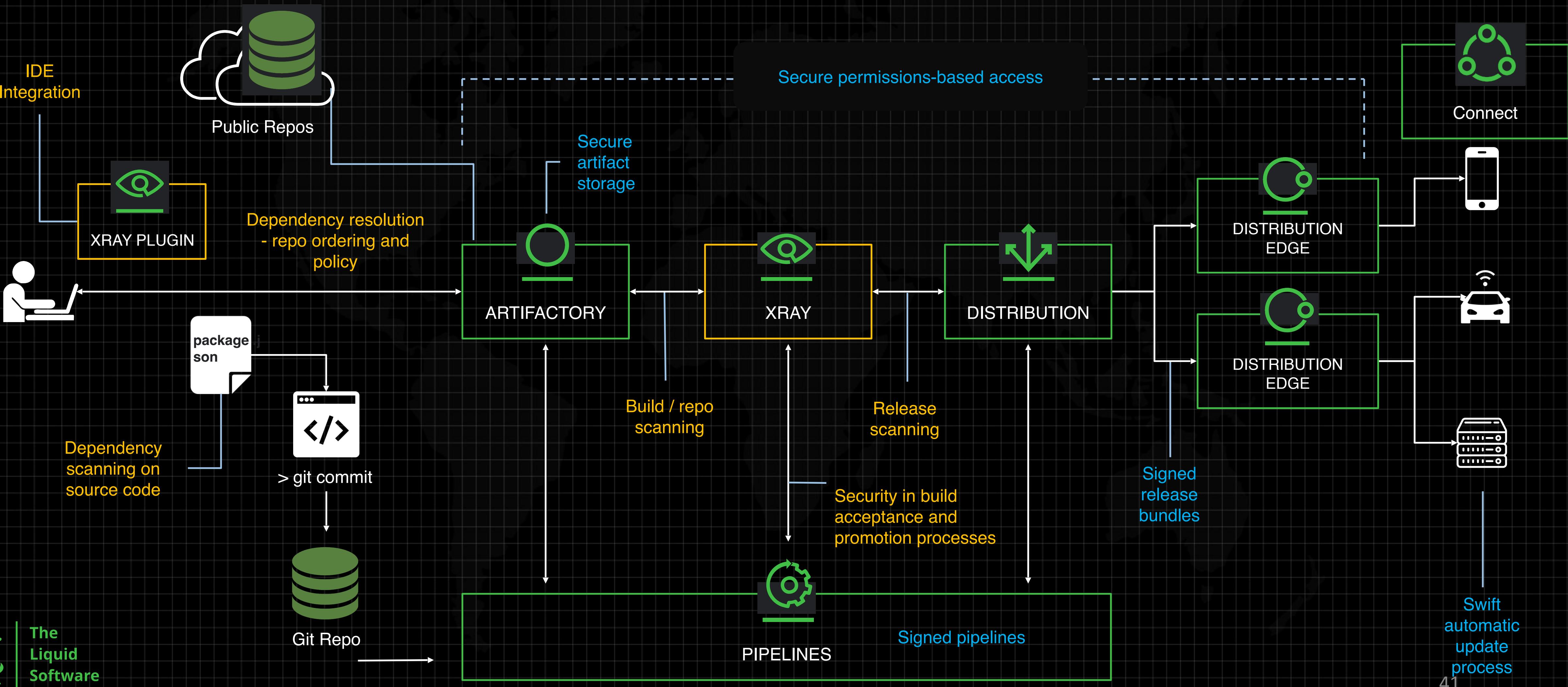
End-to-end security with the JFrog Platform

Continuously securing the software supply chain



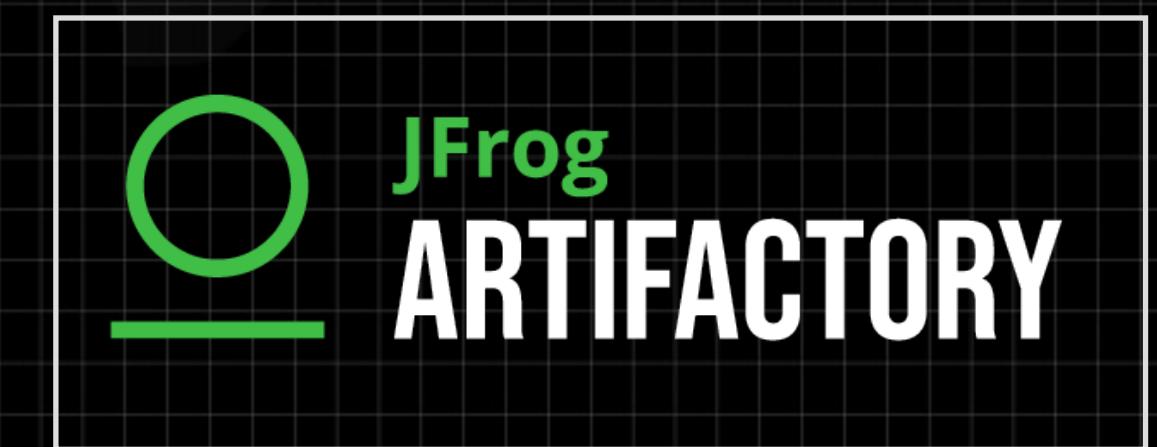
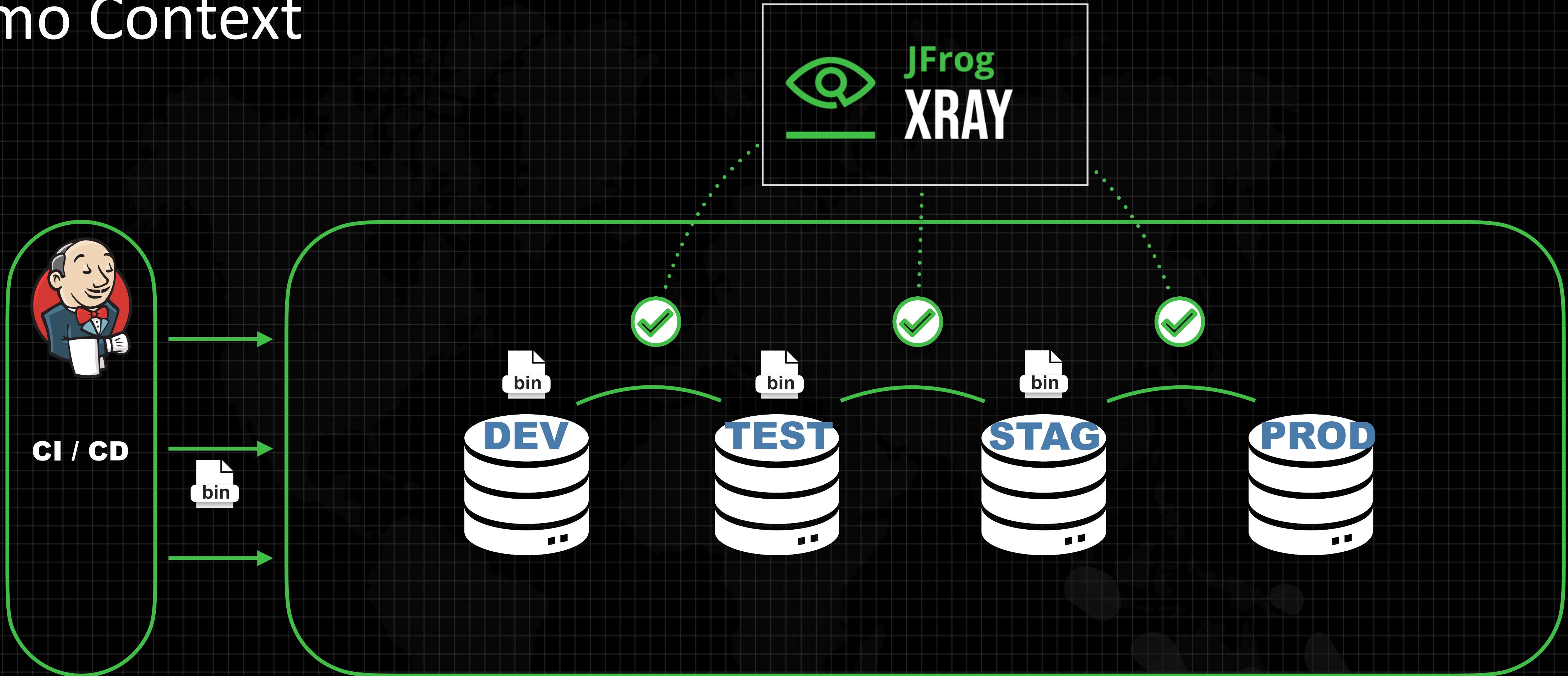
End-to-end security with the JFrog Platform

Continuously securing the software supply chain



Demo Xray

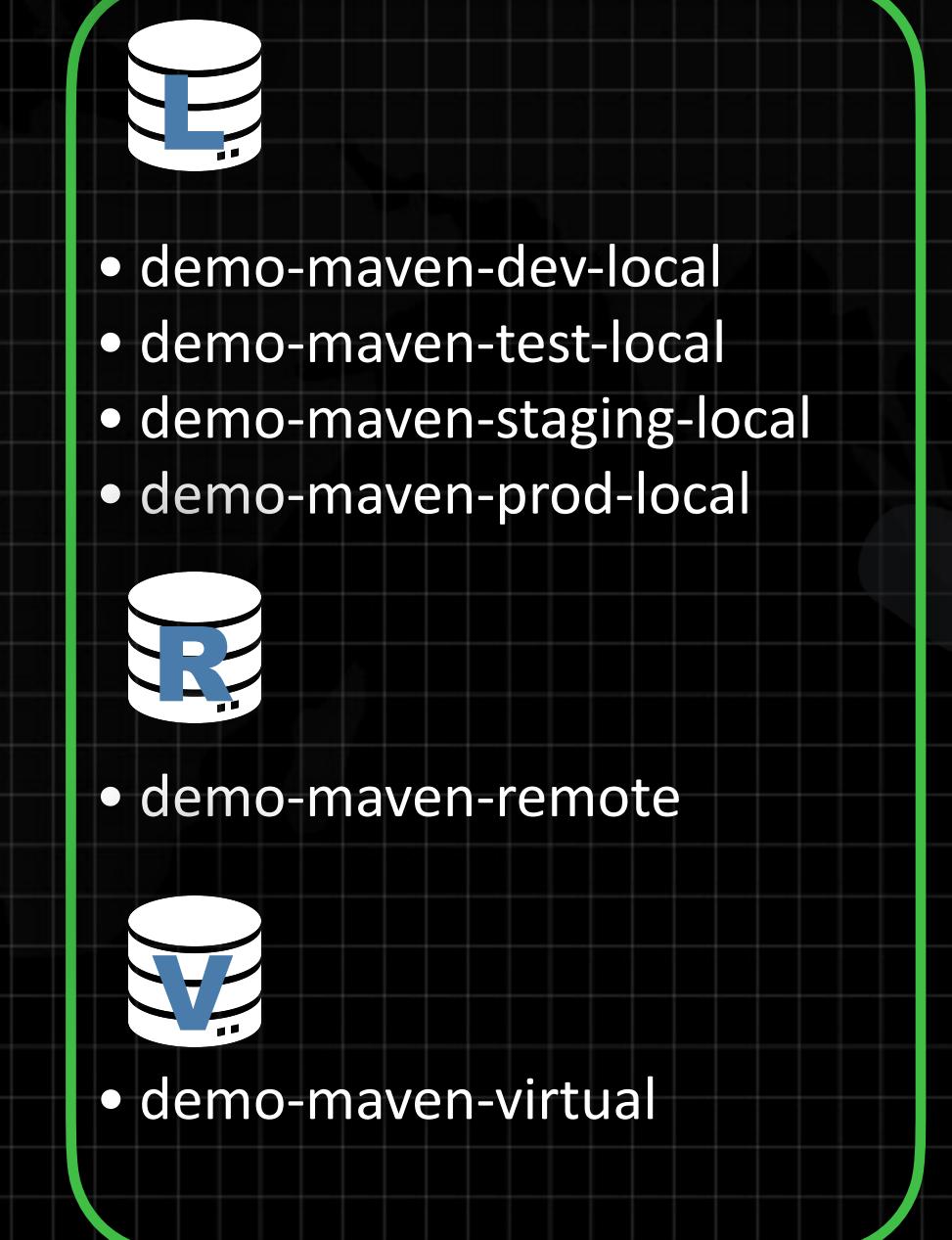
Demo Context



Demo Environment



- GSBoot-DEV
- GSBoot-Promo-Test
- GSBoot-Promo-Staging
- GSBoot-Promo-Prod
- GSBoot-Deploy



BuildWatch

GS_SPRING_BOOT_main_mvn

RepoWatch

demo-maven-prod-local

BuildPolicy

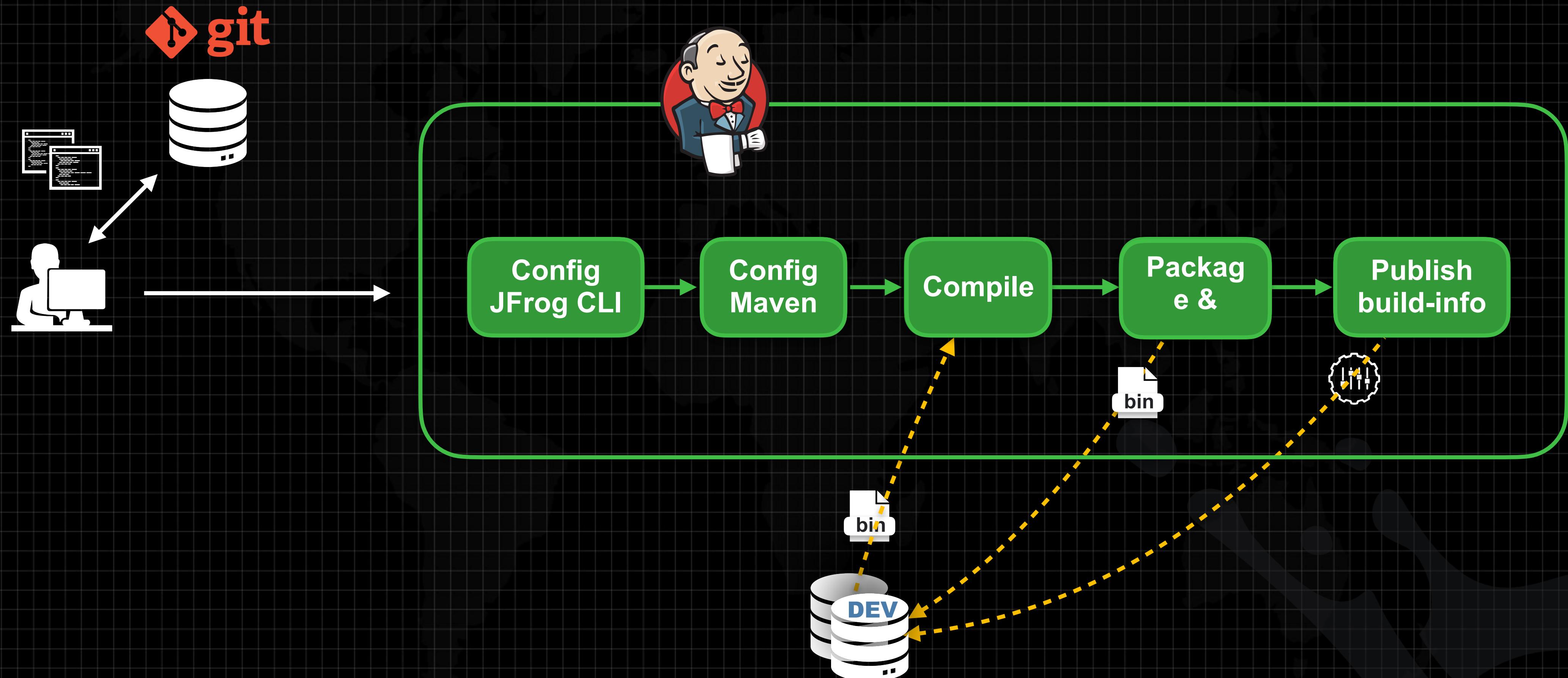
Critical $9 \leq CVSS \leq 10$
Generate violation
Fail Build

High $7 \leq CVSS < 9$
Generate violation

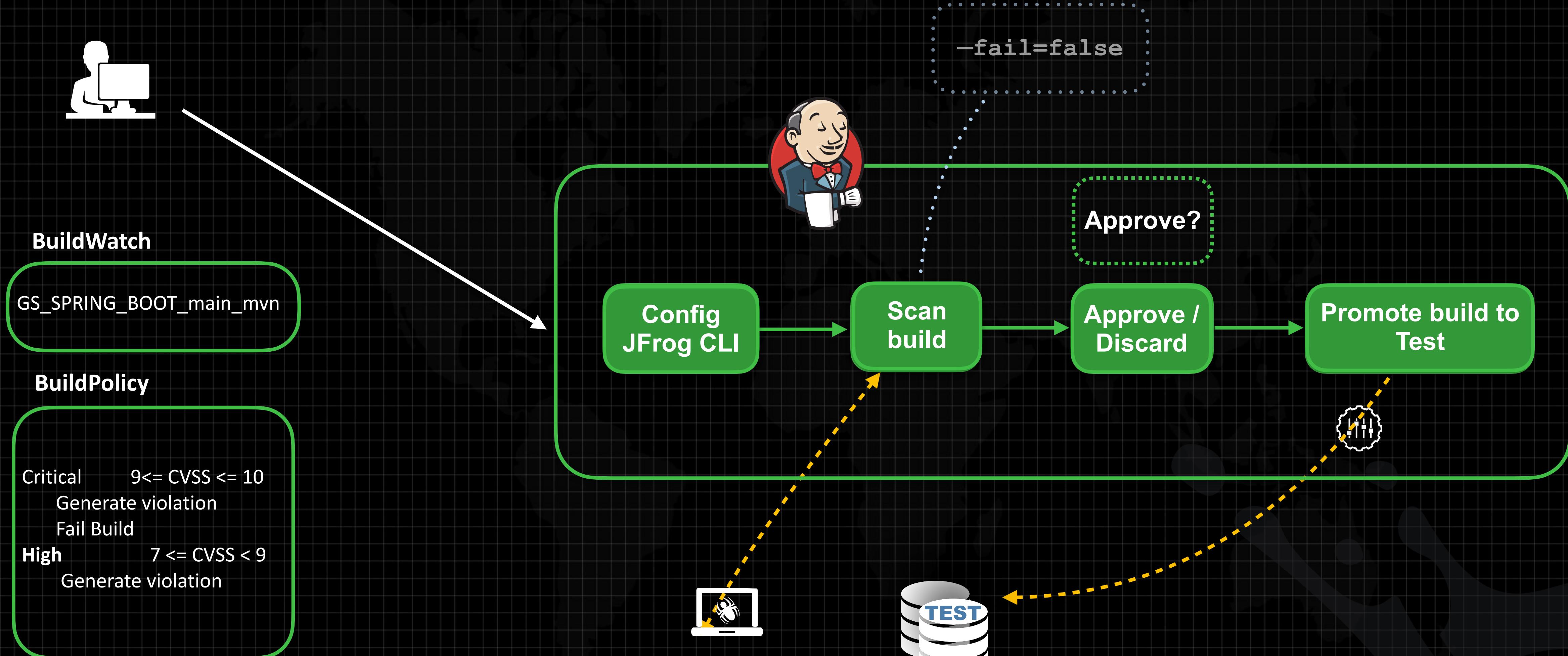
RepoPolicy

Critical $9 \leq CVSS < 10$
Block download

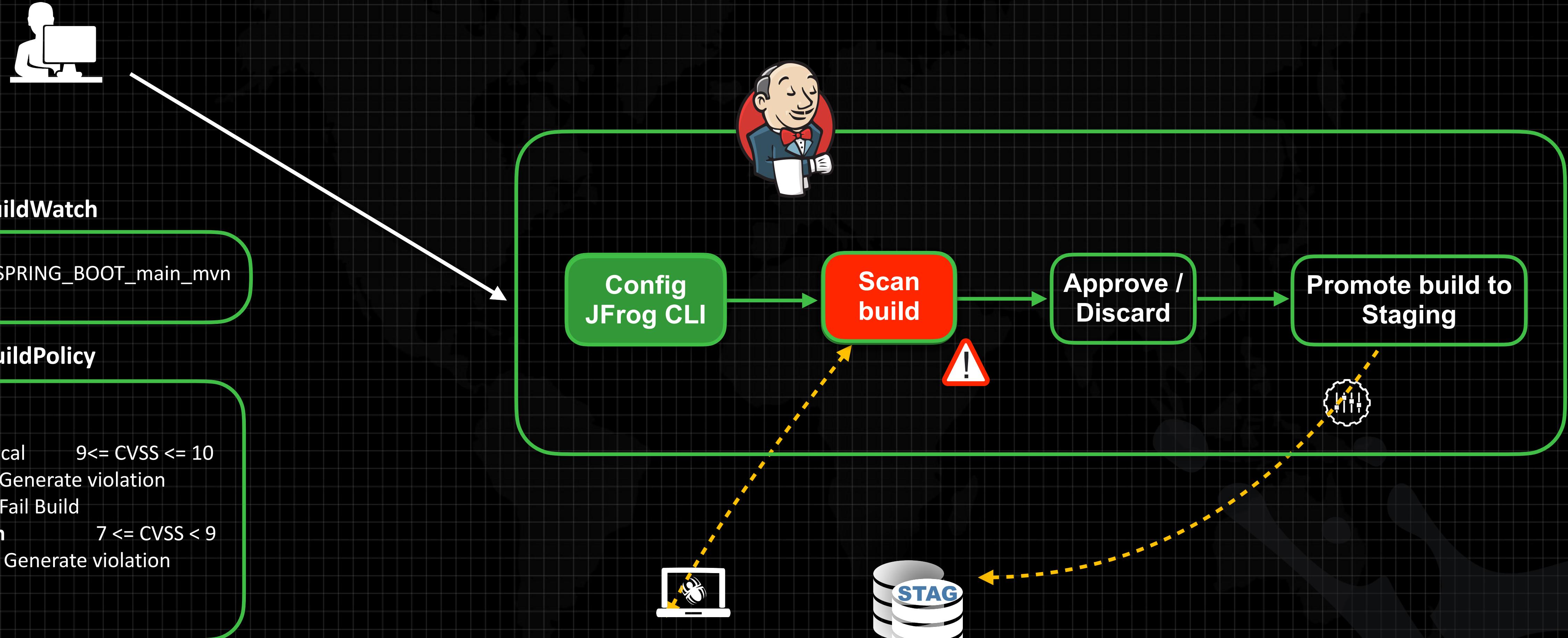
Development Pipeline



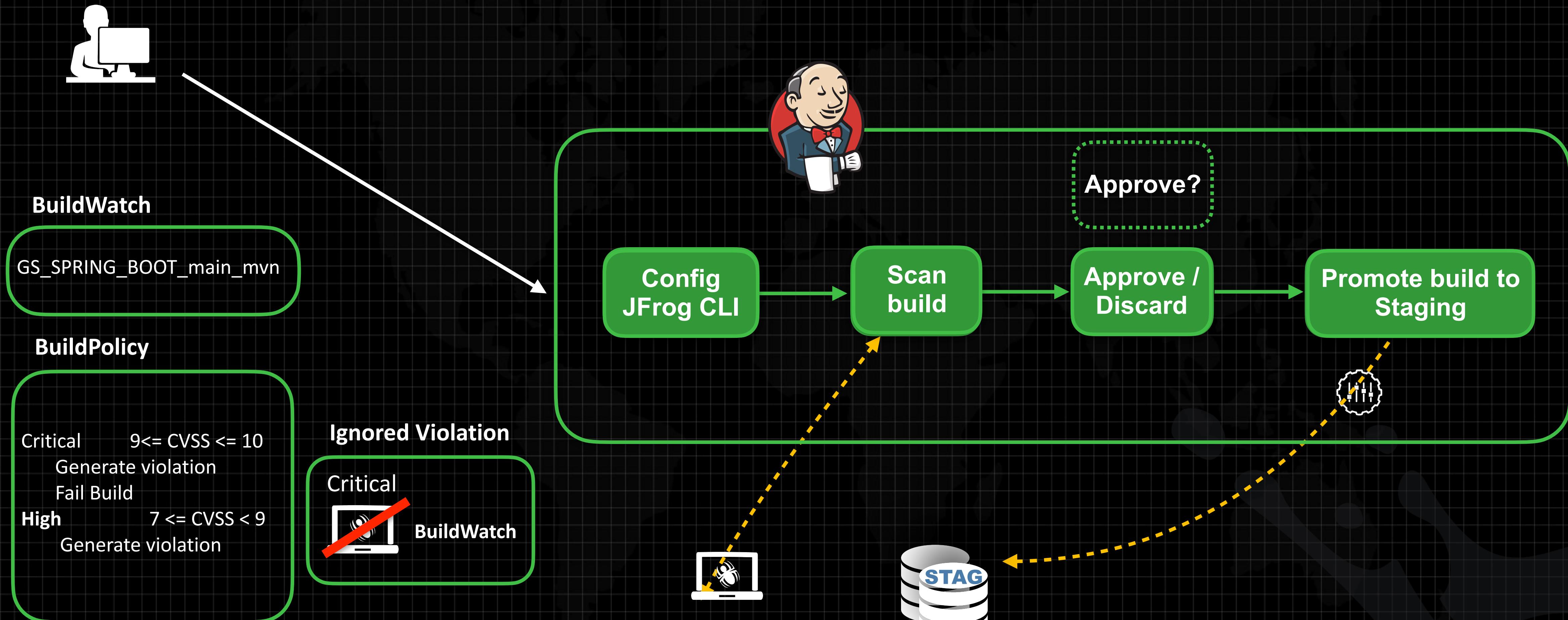
Test Promotion Pipeline



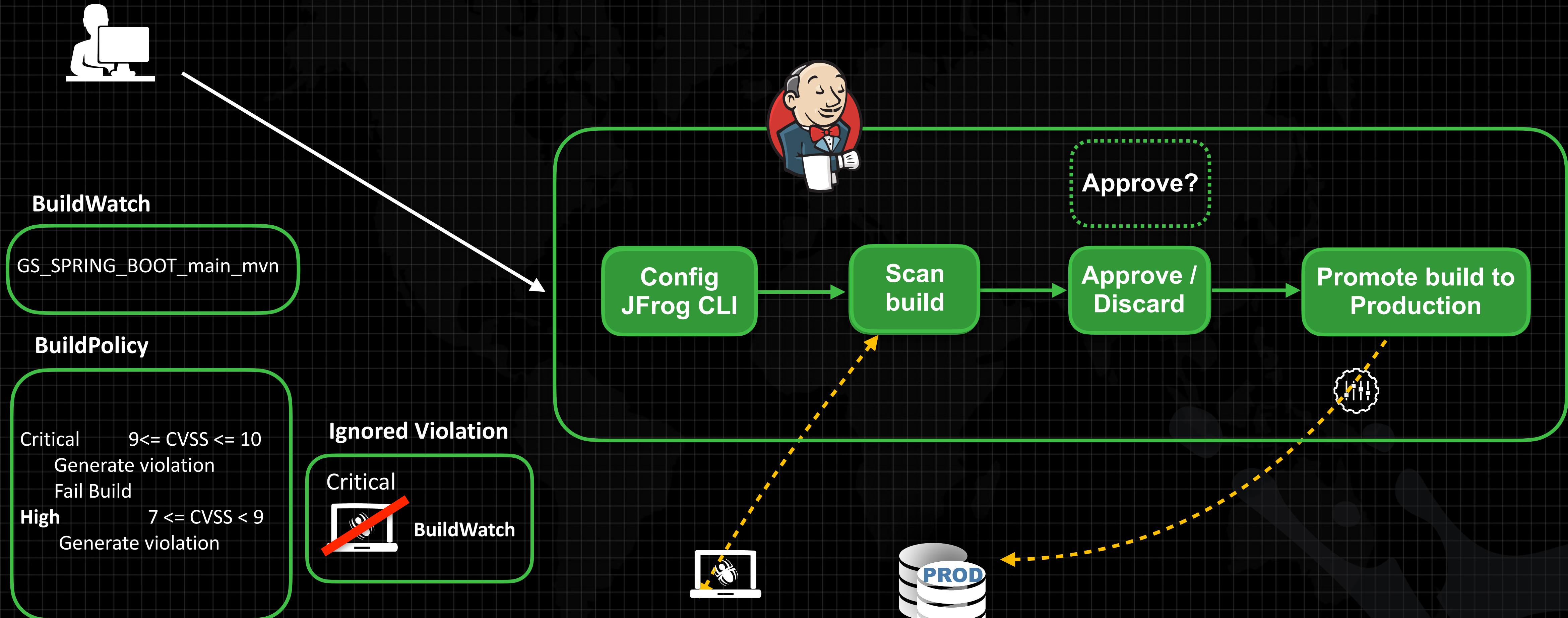
Staging Promotion Pipeline (I)



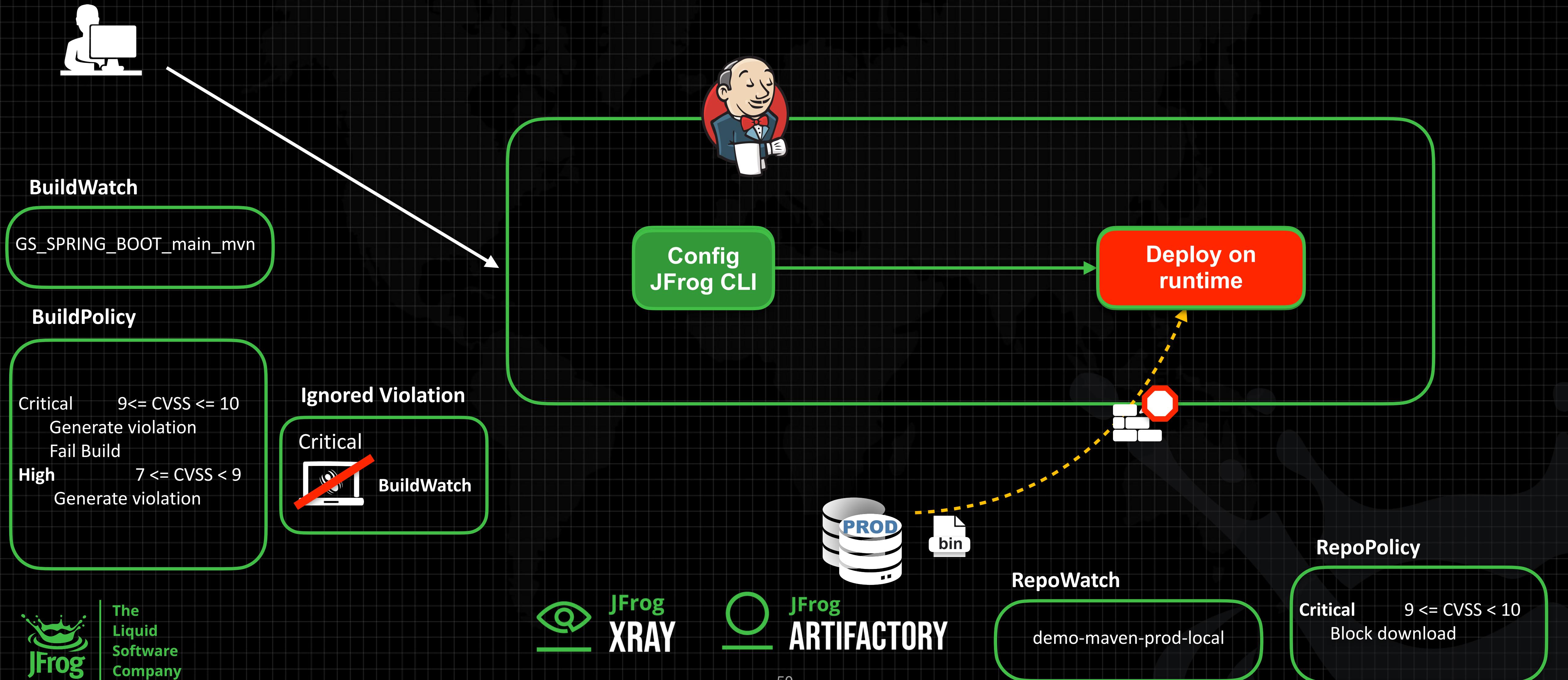
Staging Promotion Pipeline (II)



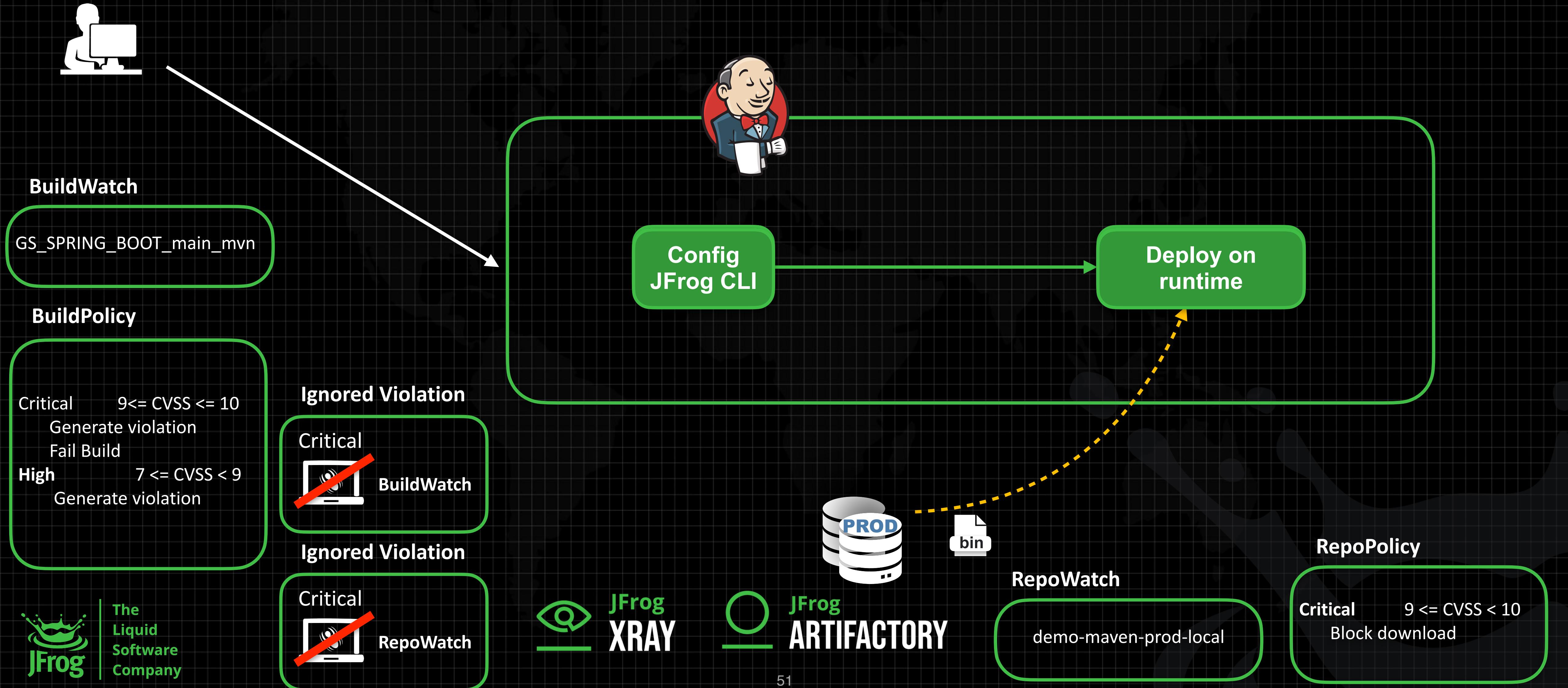
Production Promotion Pipeline



Deploy on Production Runtime Pipeline (I)



Deploy on Production Runtime Pipeline (II)



Shift Left



Dependencies Scan

Supported: Npm | Maven | Gradle | Pip | Go

```
[gs-spring-boot/complete] $ jf audit --mvn
```

On-demand Scan

(Stored under Xray / On-demand Scanning)

```
[gs-spring-boot/complete] $ jf s target/*.jar --watches="BuildWatch"
```

Docker scan

(Existing image in local registry)

```
$ jf docker scan [repo] / [image] : [tag]
```

BuildWatch

GS_SPRING_BOOT_main_mvni

RepoWatch

demo-maven-prod-local

BuildPolicy

Critical 9 <= CVSS <= 10
Generate violation

Fail Build

High 7 <= CVSS < 9
Generate violation

RepoPolicy

Critical 9 <= CVSS < 10
Block download

Shift Left (cont.)



IDE Plugin

The screenshot shows the JFrog IDE plugin integrated into an IntelliJ IDEA environment. The project tree on the left shows a 'gs-spring-boot' project with various files and folders. The central panel displays a dependency scan results for the 'gs-spring-boot' project. The 'Dependencies (Issues #)' section lists several dependencies with their versions and severity levels (e.g., Critical, High, Medium, Low). The 'Vulnerabilities' section shows a detailed view of a critical vulnerability for 'org.springframework:spring-web:5.3.21'. The 'More Info' section provides details about the vulnerability, including its severity (Critical), CVE number (CVE-2016-1000027), and a summary of the issue. The 'Impact path' section shows the dependency chain from 'org.springframework.boot:spring-boot-starter-web:2.7.1' down to 'org.springframework:spring-web:5.3.21'. The bottom of the screen shows the IntelliJ IDEA status bar and a JFrog logo.



BuildWatch

GS_SPRING_BOOT_main_mvn

RepoWatch

demo-maven-prod-local

BuildPolicy

Critical $9 \leq CVSS \leq 10$

Generate violation

Fail Build

High $7 \leq CVSS < 9$

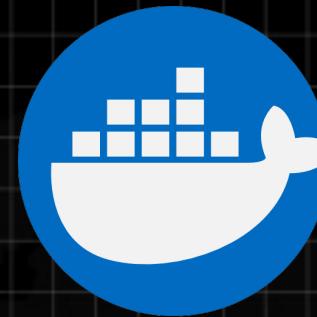
Generate violation

RepoPolicy

Critical $9 \leq CVSS < 10$

Block download

Shift Left (cont.)



Docker Desktop

Home

Containers

Images

Volumes

Dev Environments **PREVIEW**

Extensions **BETA**

JFrog

Add Extensions

JFrog Xray Scan

Shift left and run a deep recursive scan for vulnerabilities through all the layers of an image [Learn more](#)

Select local image for scanning

local/mypetclinic:latest

Scan

Vulnerabilities

Critical : 45
High : 162
Medium : 209
Low : 72
Unknown : 1

Image Scan Results

Severity	Impacted Package	Version	Type	Fix Versions	CVE	CVSS 3.0	CVSS 2.0
Critical	3.9:sqlite-libs	3.26.0-r3	Alpine	N/A	CVE-2019-19317	9.8	7.5
Critical	com.h2database:h2	1.4.200	Maven	[2.0.206]	CVE-2021-42392	9.8	10.0
Critical	3.9:musl-utils	1.1.20-r4	Alpine	[1.1.20-r5]	CVE-2019-14697	9.8	7.5
Critical	3.9:musl	1.1.20-r4	Alpine	[1.1.20-r5]	CVE-2019-14697	9.8	7.5
Critical	com.h2database:h2	1.4.200	Maven	[2.1.210]	CVE-2022-23221	9.8	10.0
Critical	com.h2database:h2	1.4.200	Maven	[2.0.202]	CVE-2021-23463	9.1	6.4
Critical	3.9:freetype	2.9.1-r2	N/A		CVE-2022-27404	9.8	7.5

The LiquiSoft Company

Give Feedback



BuildWatch

GS_SPRING_BOOT_main_mvn

RepoWatch

demo-maven-prod-local

BuildPolicy

Critical 9<= CVSS <= 10
Generate violation
Fail Build
High 7 <= CVSS < 9
Generate violation

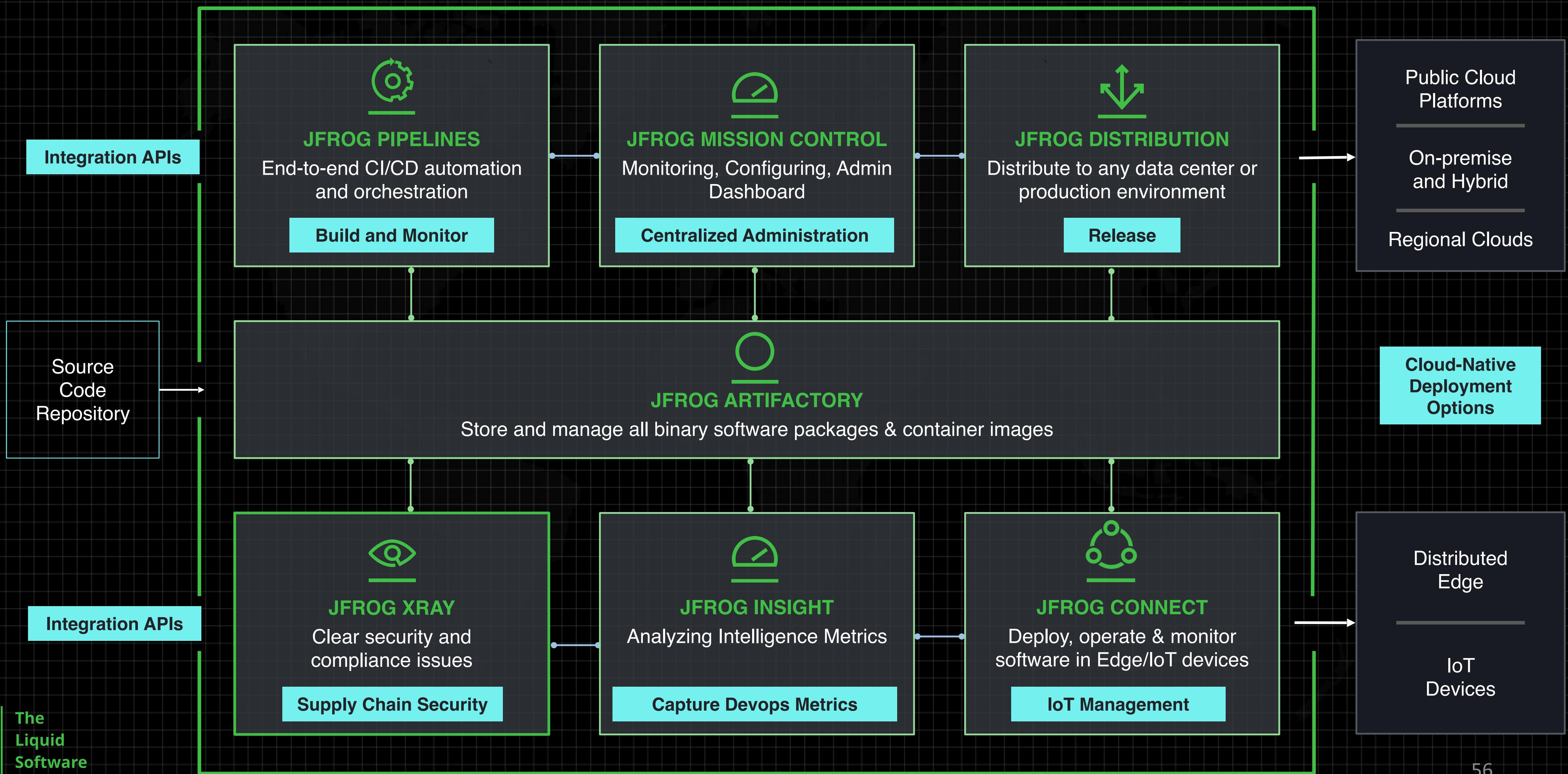
RepoPolicy

Critical 9 <= CVSS < 10
Block download

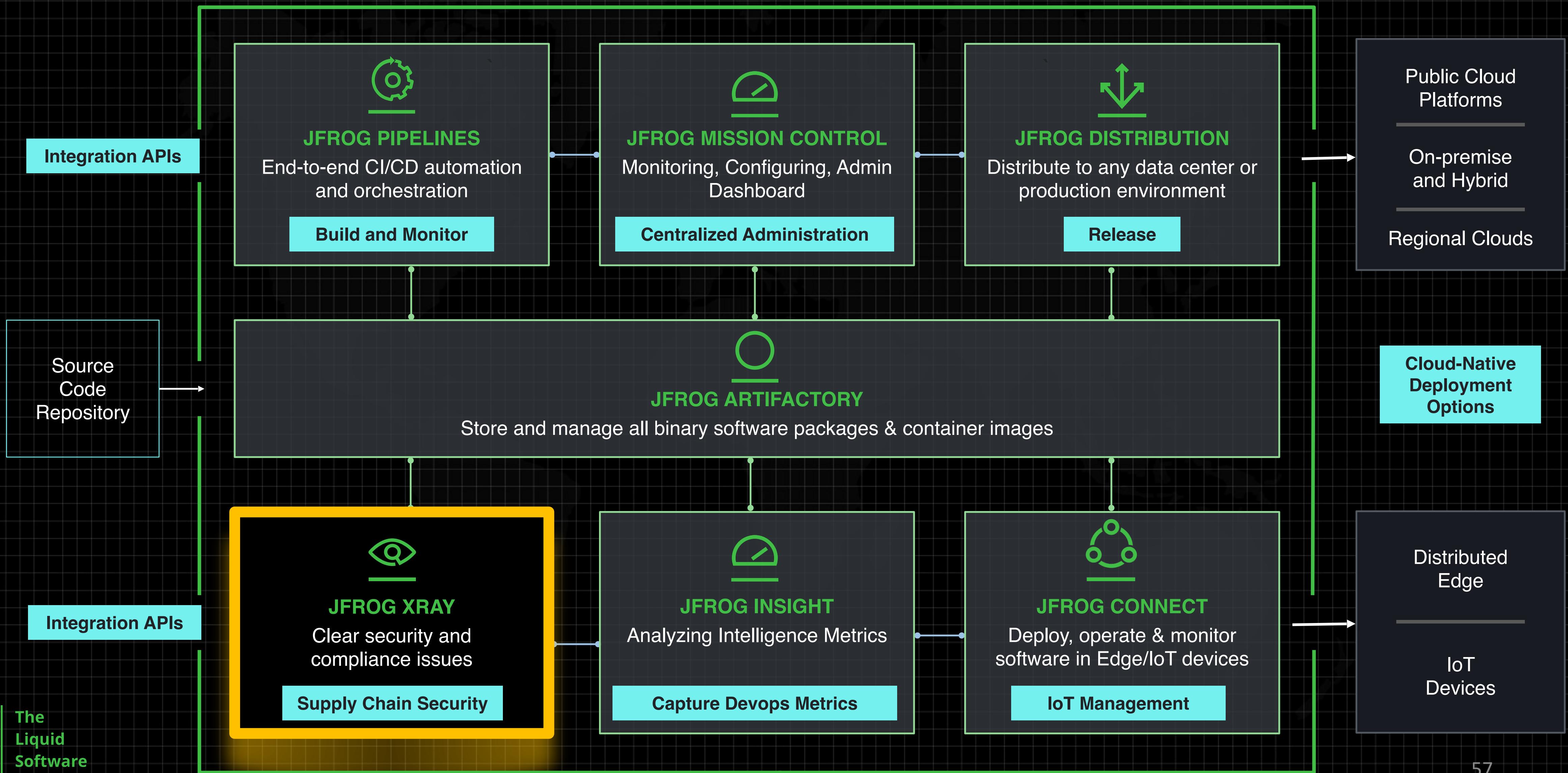


JFrog Advanced Security

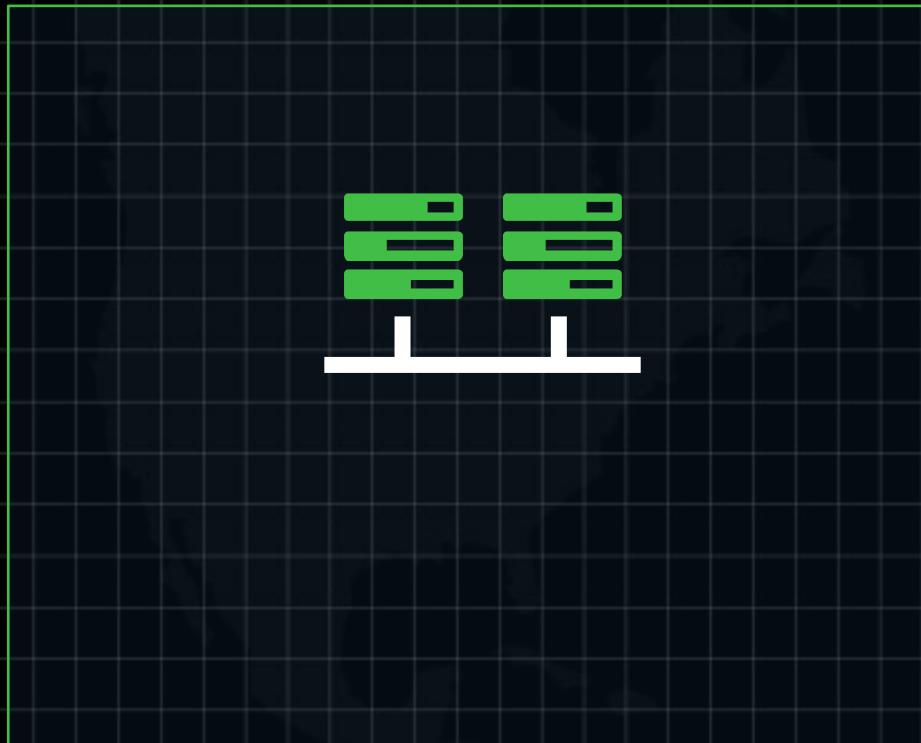
THE JFROG DEVOPS PLATFORM



THE JFROG DEVOPS PLATFORM



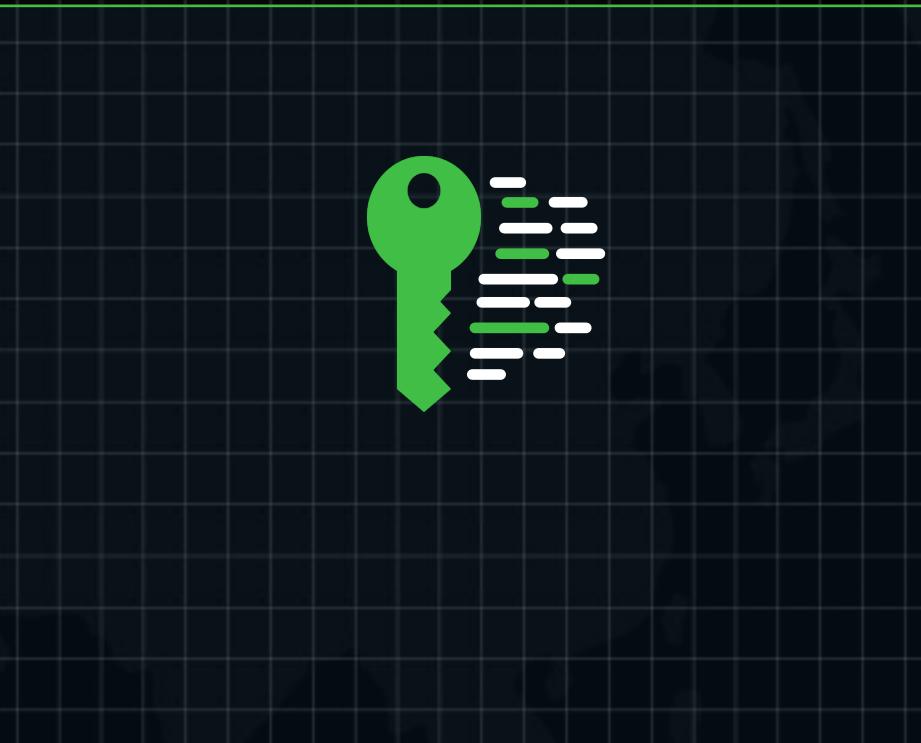
Software Supply Chain Attack Trends



Known vulnerabilities
in 3rd Party software
("CVEs"), OSS or
Commercial



Unknown
vulnerabilities in
1st and 3rd party
code ("Zero-
days")

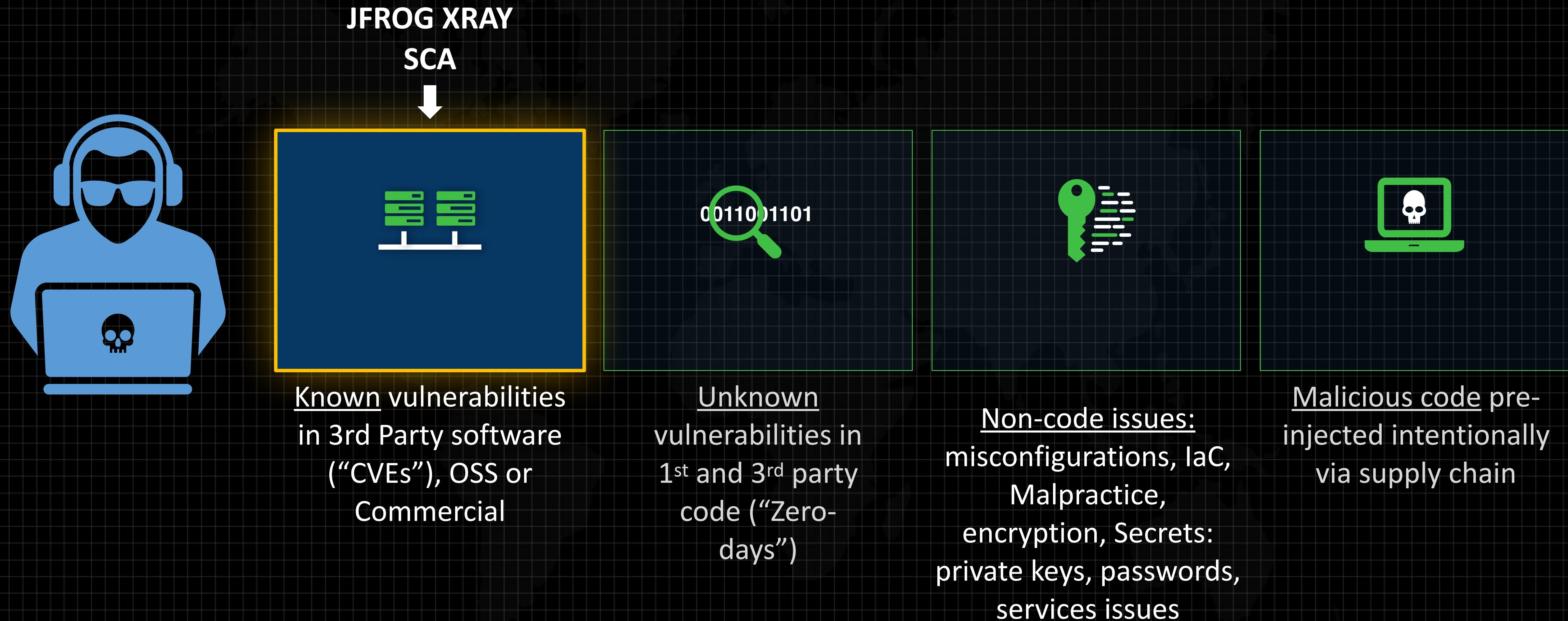


Non-code issues:
misconfigurations, IaC,
Malpractice,
encryption, Secrets:
private keys, passwords,
services issues

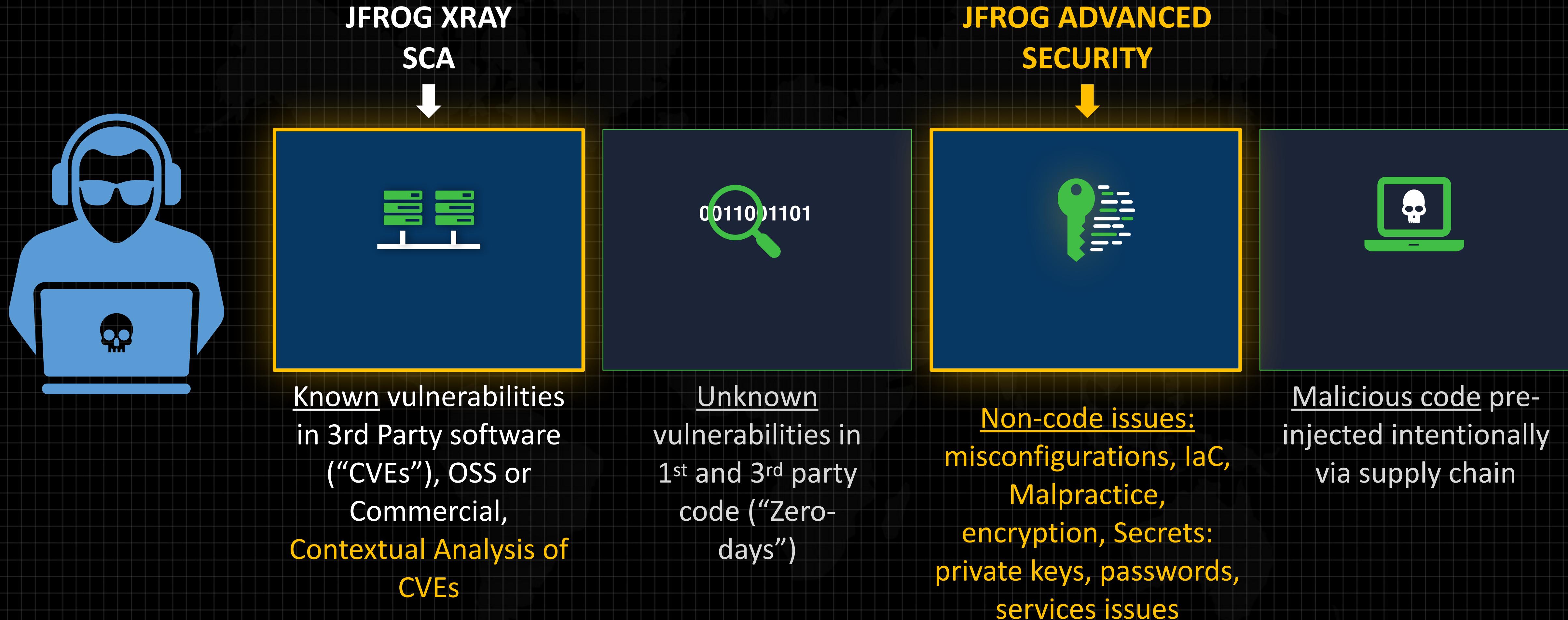


Malicious code pre-
jected intentionally
via supply chain

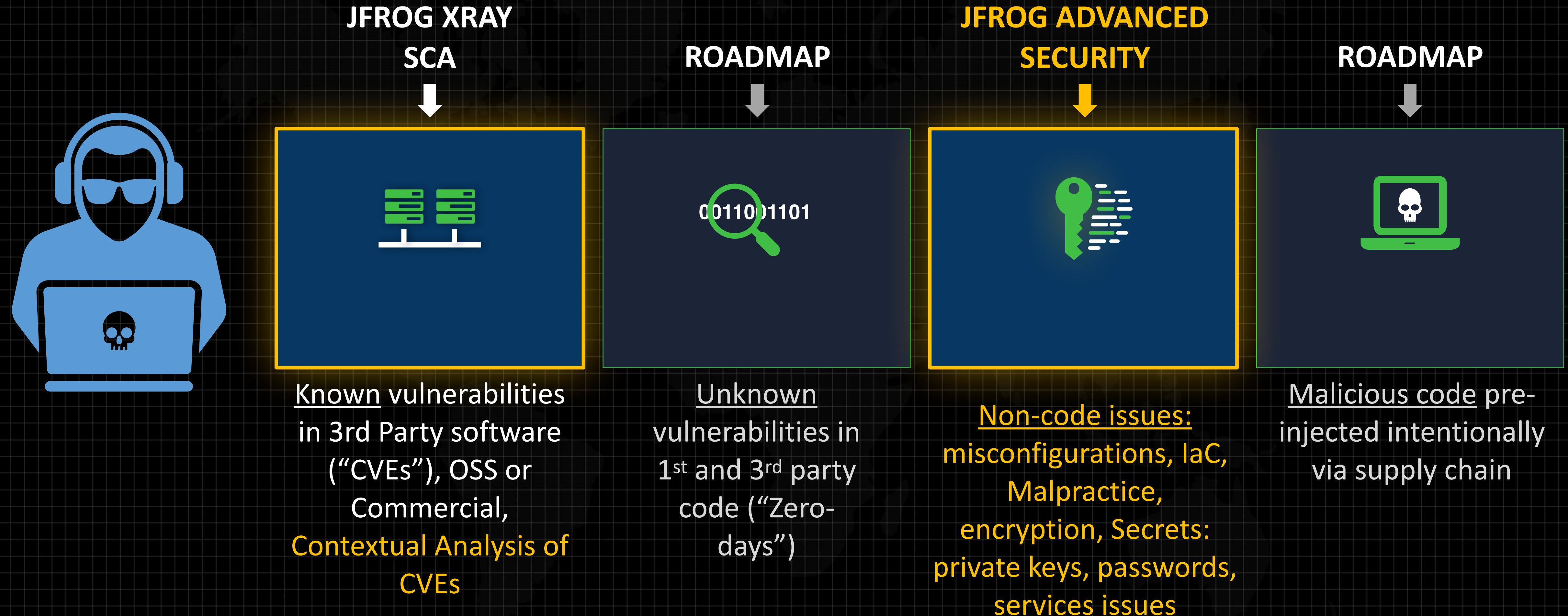
Software Supply Chain Attack Trends



Software Supply Chain Attack Trends



Software Supply Chain Attack Trends



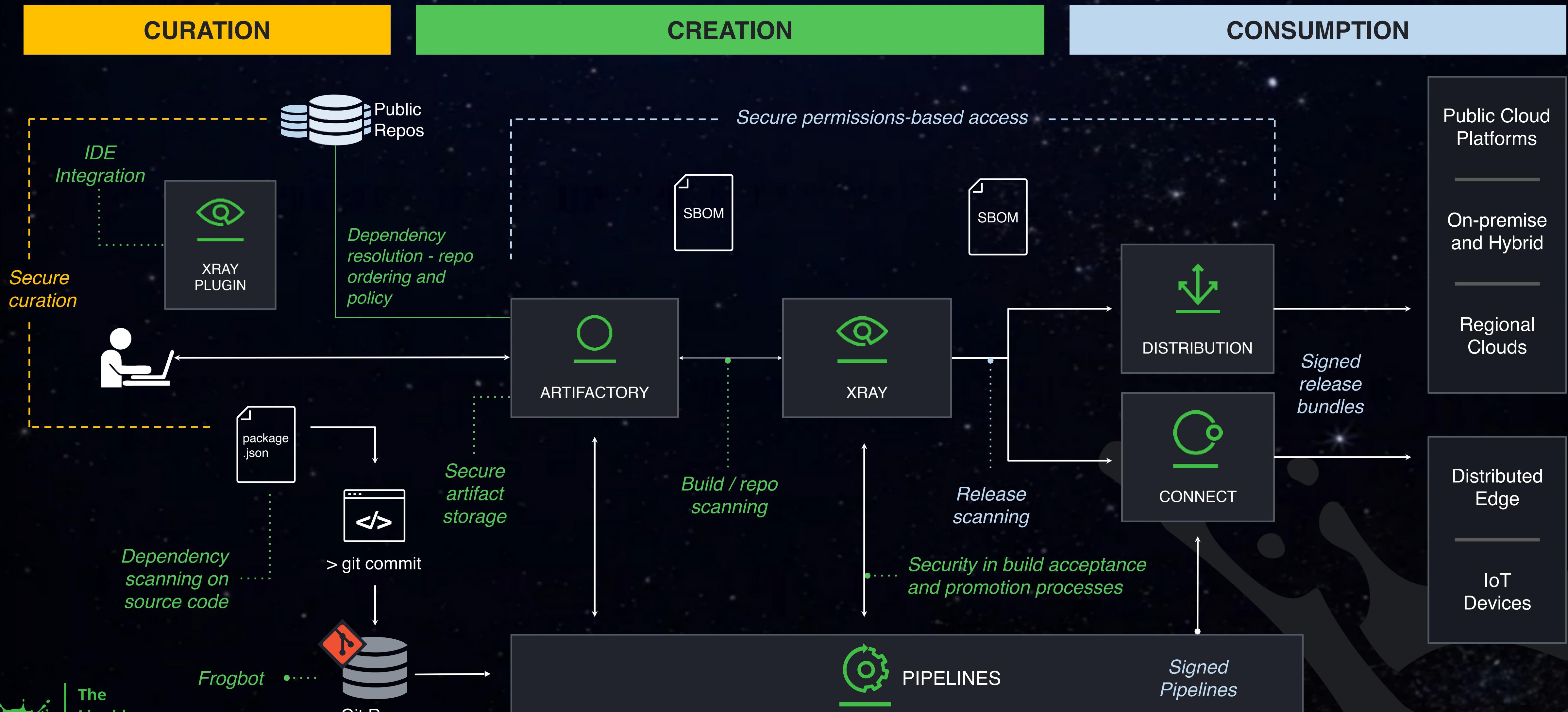
Reinforcing Xray's Supply Chain Security Capabilities



ADVANCED SUPPLY CHAIN-WIDE SECURITY CAPABILITIES



Security & Compliance: Infused Throughout The SSC



JFrog Advanced Security Capabilities



CONTEXTUAL ANALYSIS

Scan containers to identify and prioritise whether OSS vulnerabilities are actually exploitable.



SECRETS DETECTION

Detects any secrets left exposed in any containers stored in Artifactory to stop any accidental leak of internal tokens or credentials.



APPLICATION EXPOSURE

Find configuration issues, security malpractices, and insecure usage of popular OSS libraries related to your application framework



SERVICES EXPOSURE

Detect configuration issues and security malpractices for specific services and daemons included in your artifacts



IAC SECURITY ANALYSIS

Infrastructure-as-Code (IaC) Security Analysis
Scans IaC files stored in Artifactory for early detection of cloud and infrastructure misconfigurations to prevent attacks and data leak.

Contextual Analysis Coverage

JFrog Research Extended CVE
Information

1100+ **CVEs**

JAS Contextual Analysis
Scanner

400+ **CVEs**

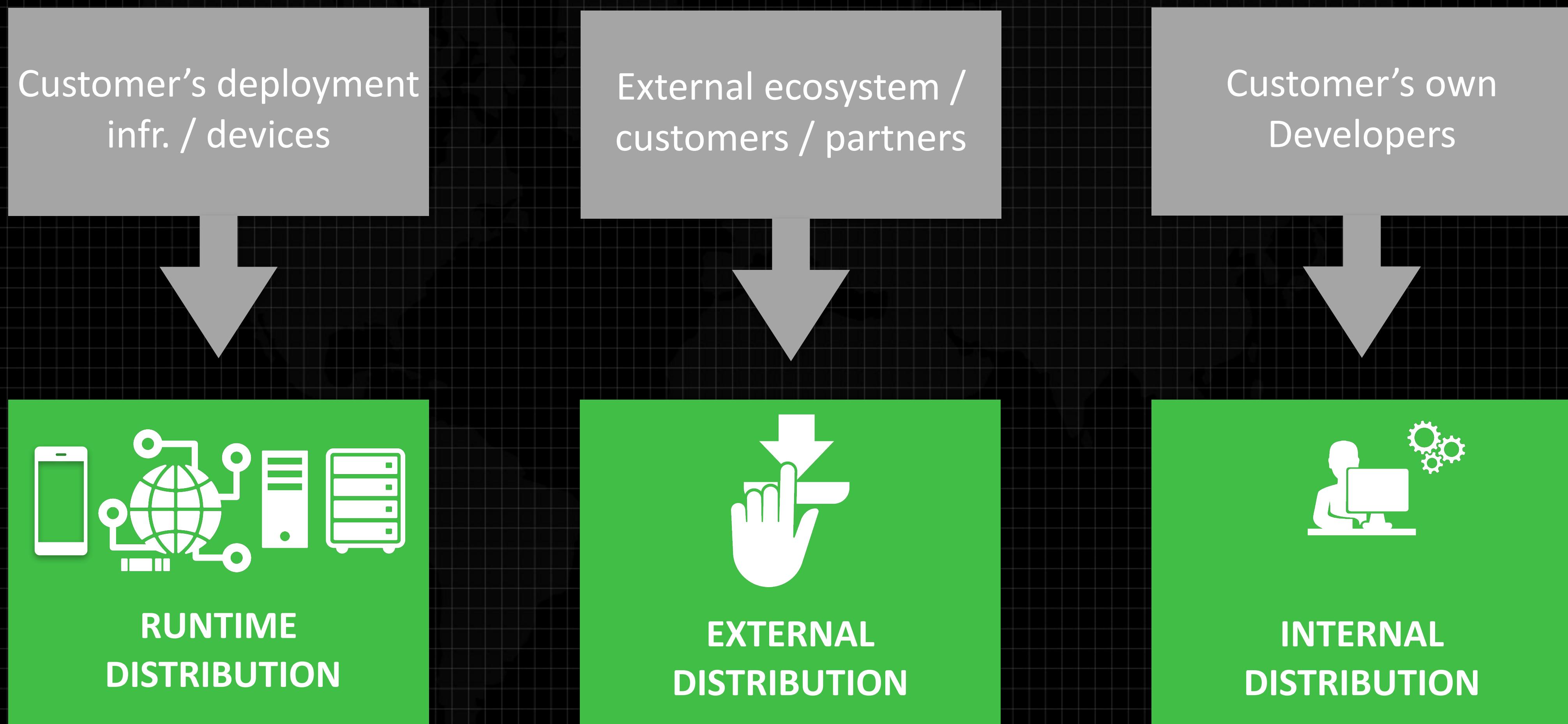
40

Contextual Scanners
Monthly



Distribute your software with JFrog Distribution

Who is consuming the distributed software

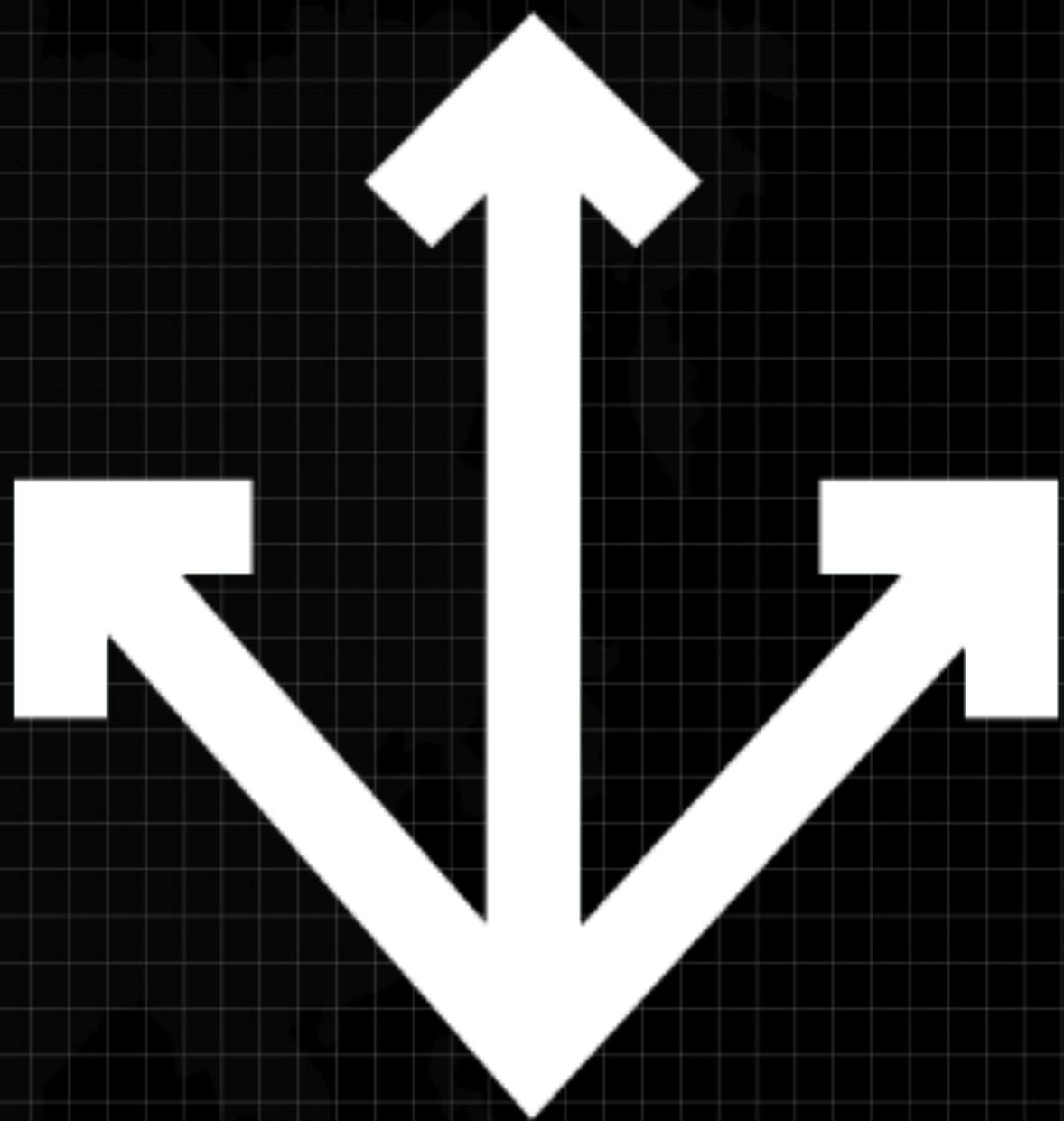


Possible Solutions

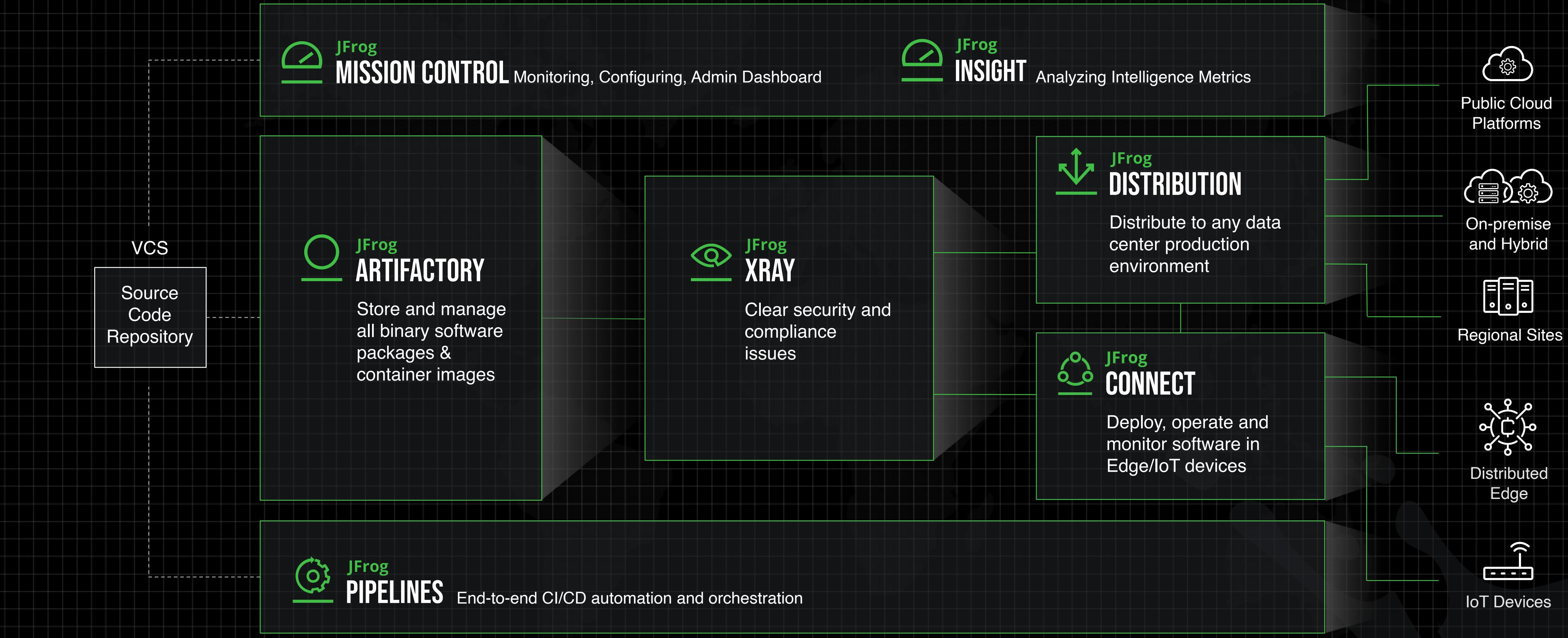
- Cloud Storage Hosted Solution
- CDN
- Download Centers
- Replication
- Federated Repositories
- JFrog Distribution

Possible Solutions

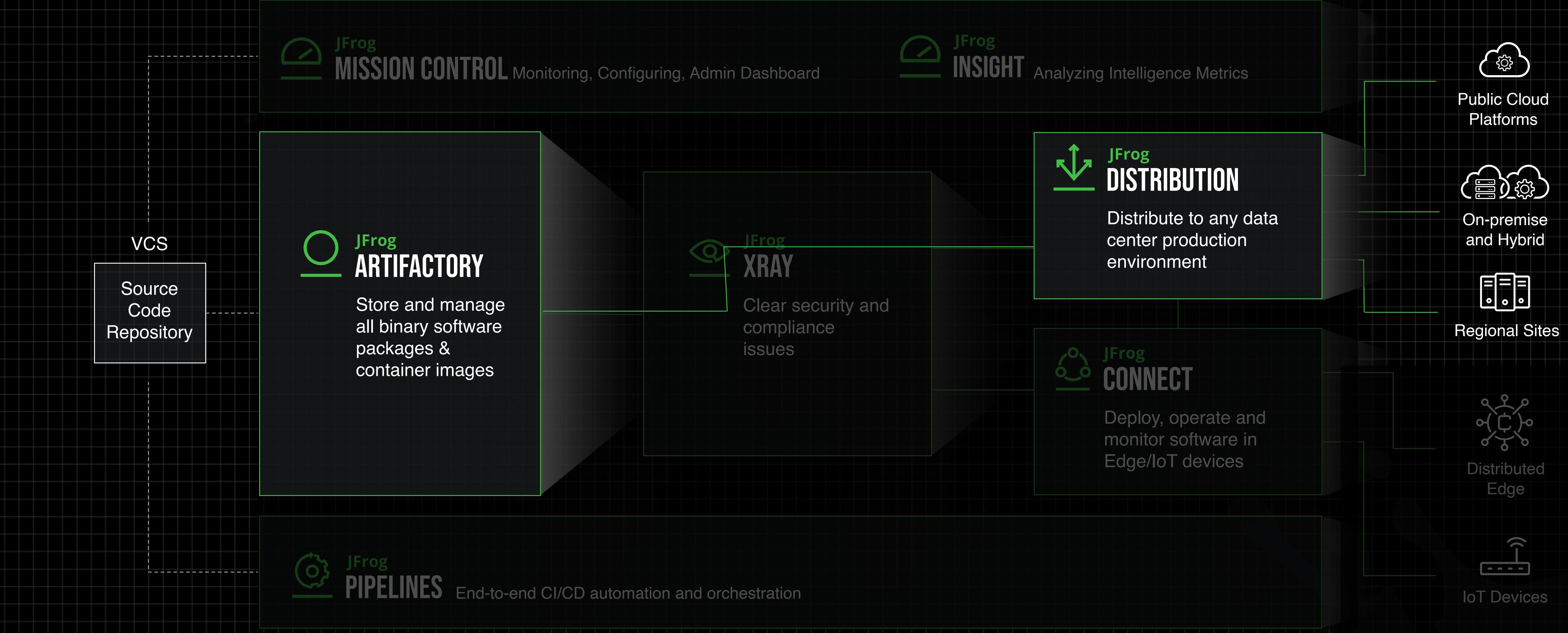
- Cloud Storage Hosted Solution
- CDN
- Download Centers
- Replication
- Federated Repositories
- JFrog Distribution



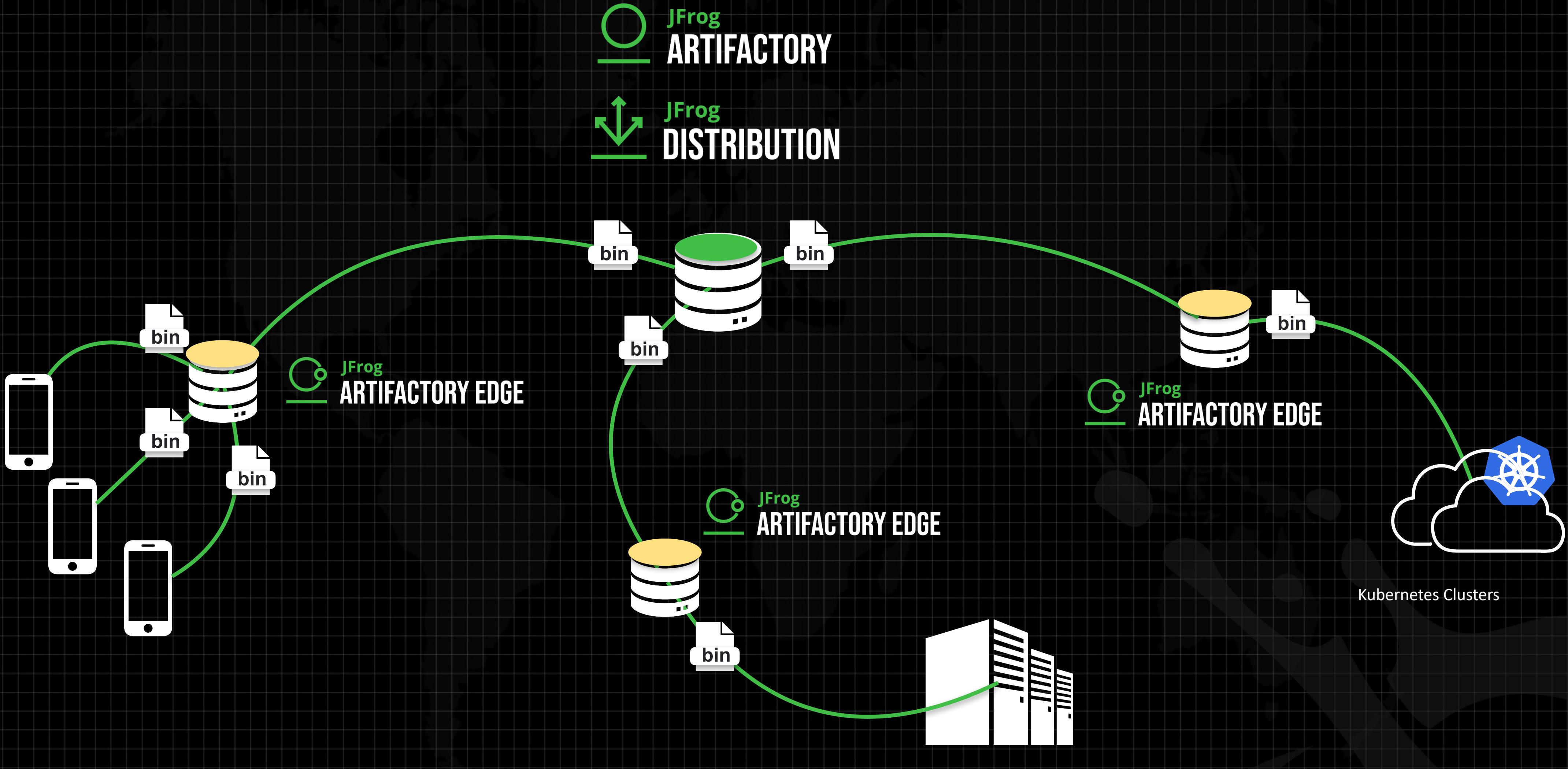
Overview of the JFrog Platform



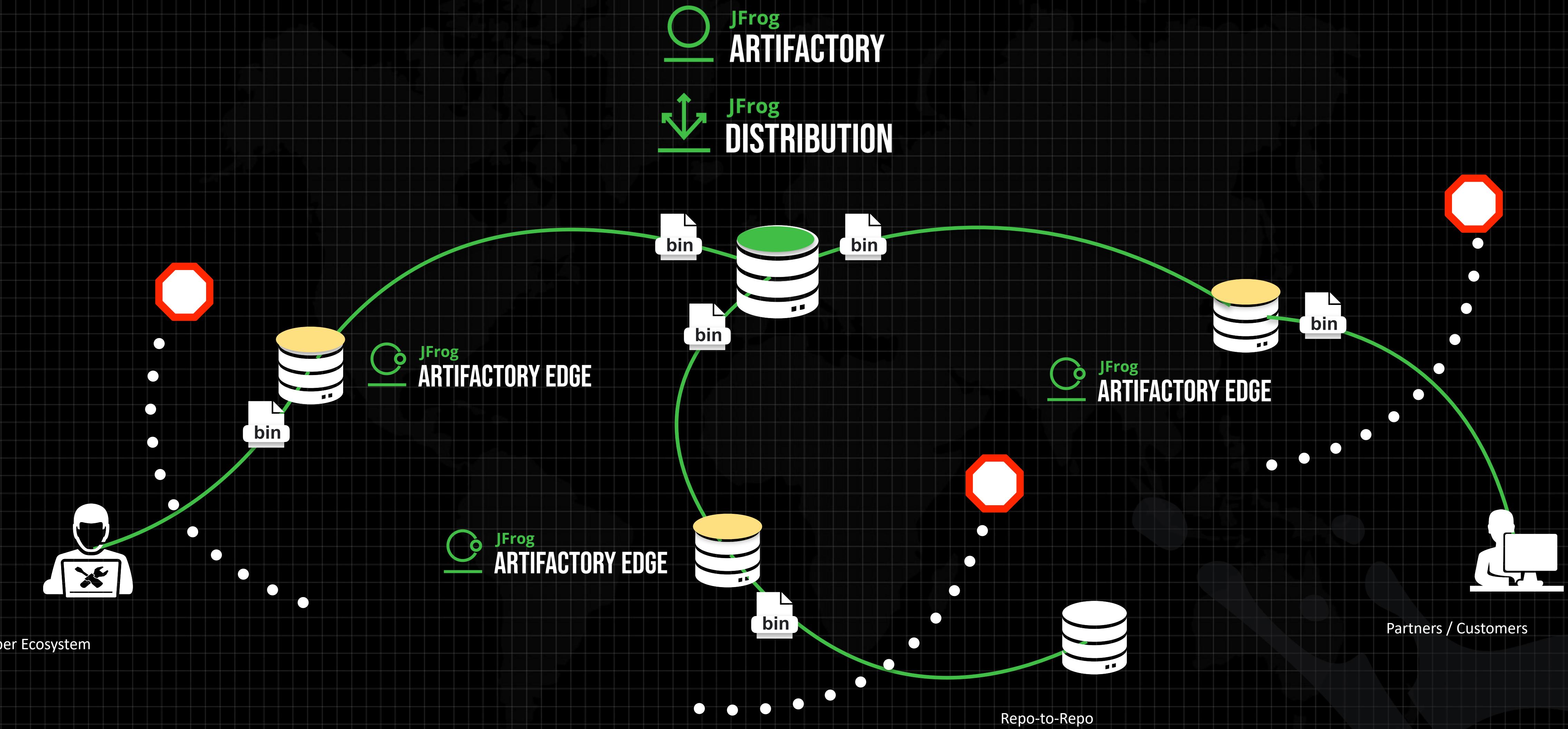
Overview of the JFrog Platform



Runtime & Infrastructure EDGE Distribution



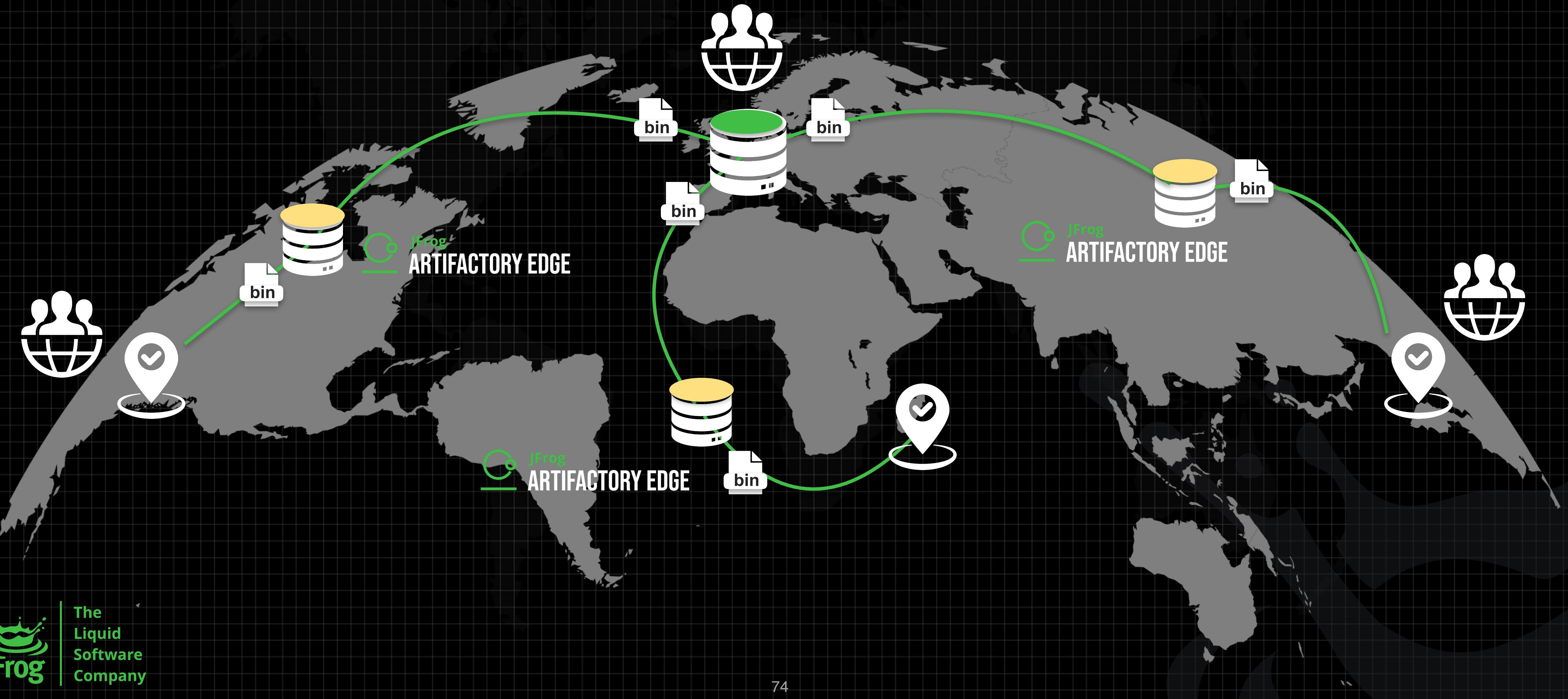
External Distribution



Internal Distribution

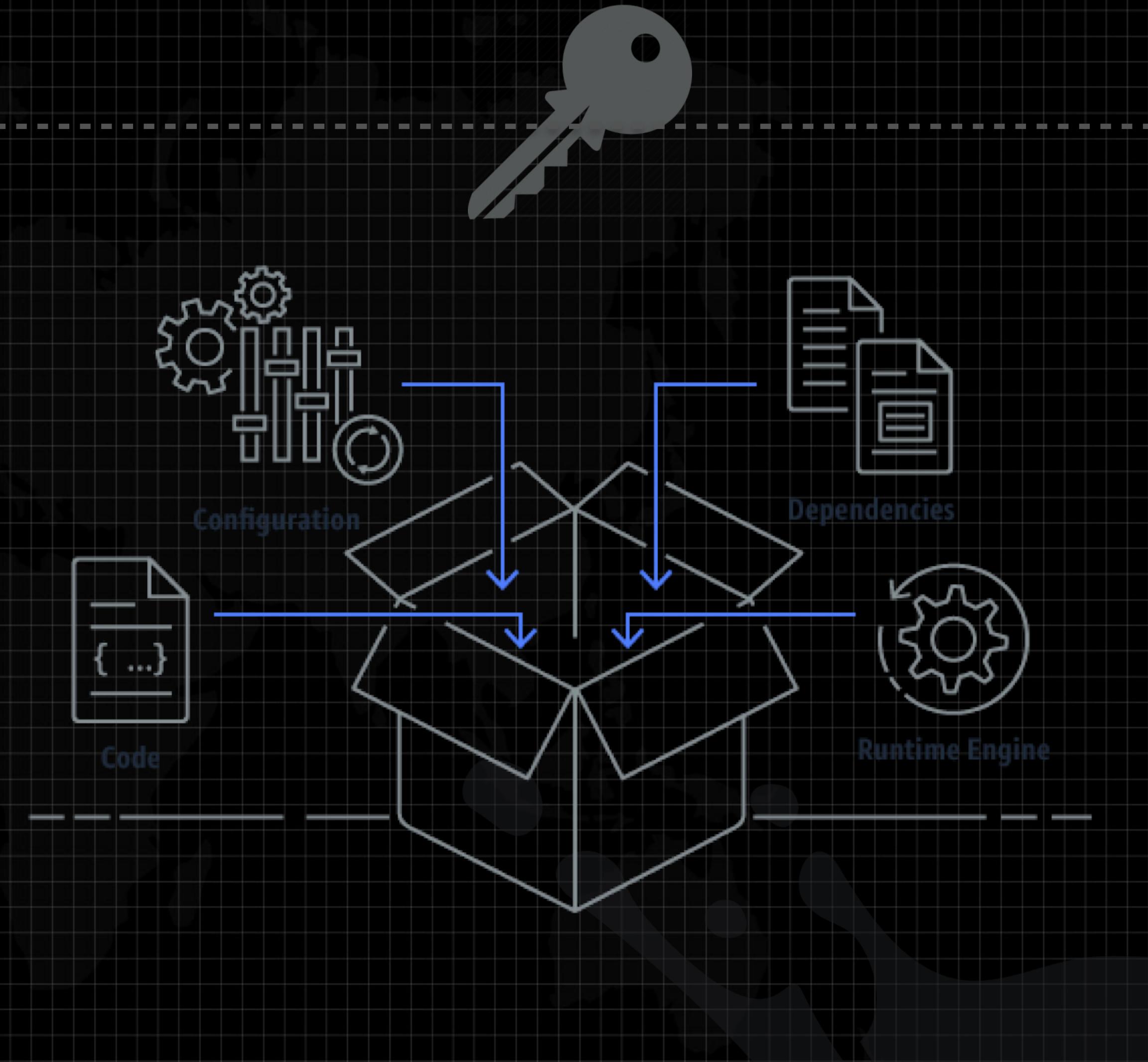
JFrog
ARTIFACTORY

↑↓ JFrog
DISTRIBUTION



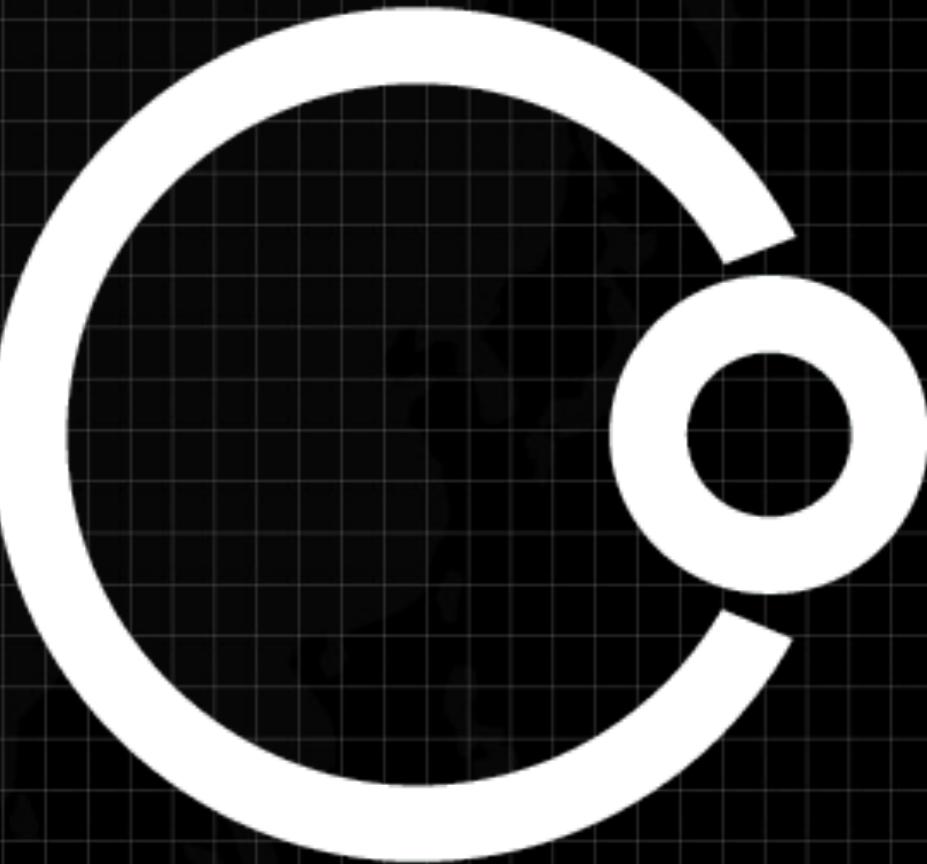
Release Bundle

- A Release Bundle is a collection of artifacts grouped together
- This can be for example a Docker image together with a Helm chart
- Release Bundles can be created using REST API, CLI or through the Distribution UI
- Creating a Release Bundle requires Release Bundle writing permissions
- Secured and immutable



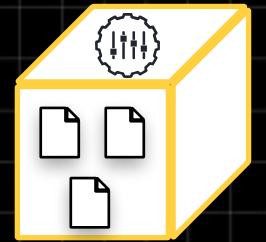
Edge Node

- “Read-only” artifactory
- Deployed physically close to the final destination of the artifact
- Direct upload of artifacts to an Edge node is not possible (except for a designated generic repository)
- No Remote repos (only smart remote repositories)
- No Build Integration
- No Xray scans allowed



Distributing software across the globe

Push distribution



Release Bundle

Distributing software across the globe

Pull Distribution



Artifact / Binary



Smart Remote Repository

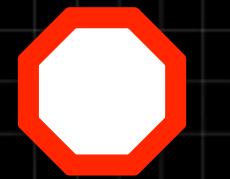


The
Liquid
Software
Company



Distributing software across the globe

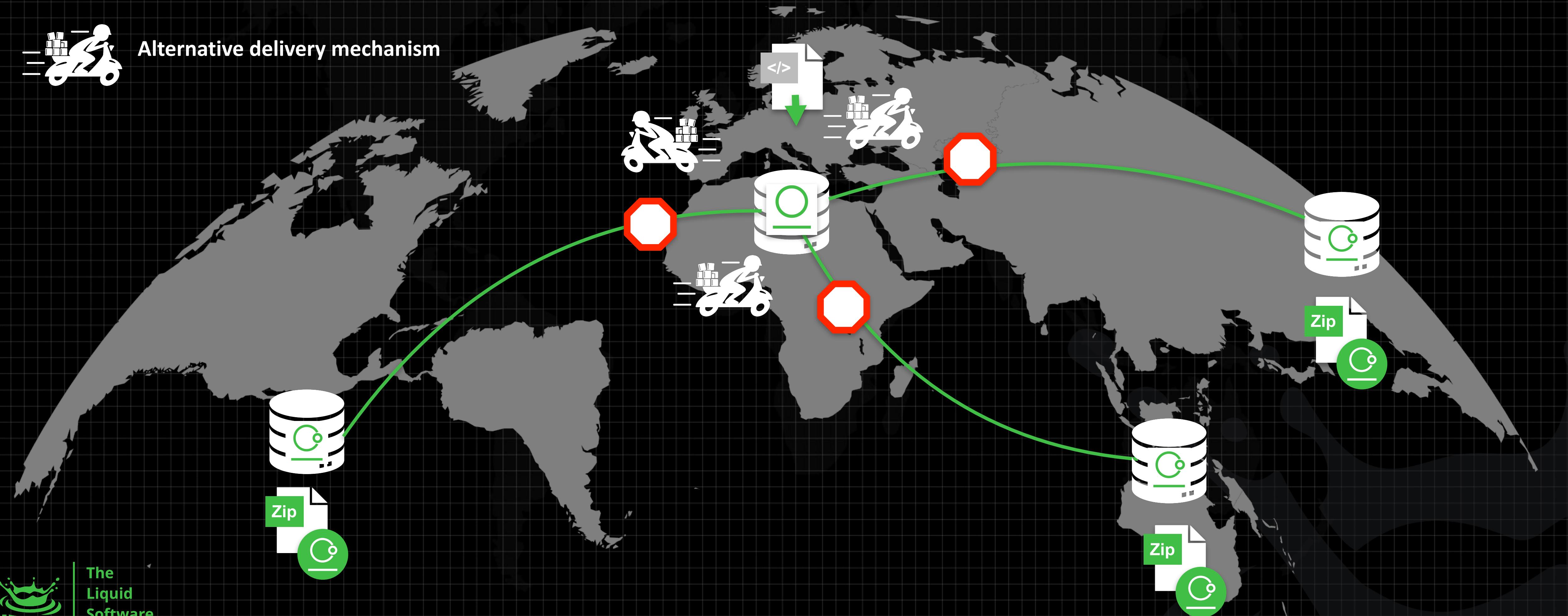
Air Gap Distribution



Firewall rules



Alternative delivery mechanism



End-to-end security with the JFrog Platform

Continuously securing the software supply chain

