DevOpsSchool
Lets Learn, Share & Practice DevOps

A Jfrog Artifactory

An Introduction

DevOpsSchool
Lets Learn, Share & Practice DevOps

Rajesh Kumar

www.RajeshKumar.xyz

DevOps@RajeshKumar.xyz

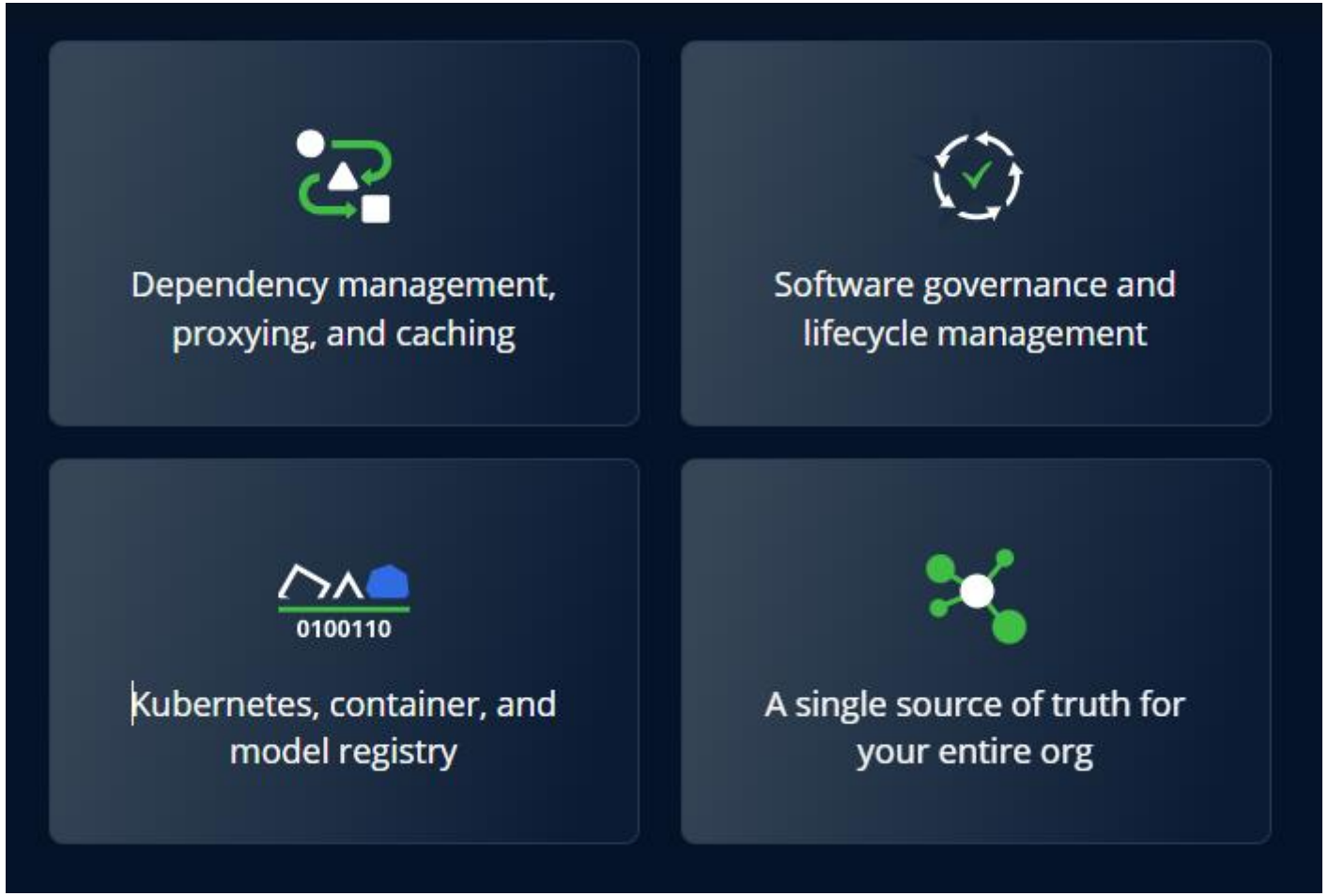
What is Artifactory?

A Repository Management Systems with Registry Systems From

What is Artifactory?

JFrog Artifactory is the single solution for housing and managing all the software artifacts, AI/ML models, binaries, packages, files, containers, components, and releases used in and generated across your organization's software supply chain. Artifactory serves as your central hub for DevOps and developers - integrating with your tools and processes to improve automation, capture attestation, ensure release integrity, and provide unrivaled visibility across your development processes.

What is Artifactory?



What is Artifactory?

Universal | Scalable | Secure | Automated | Performance

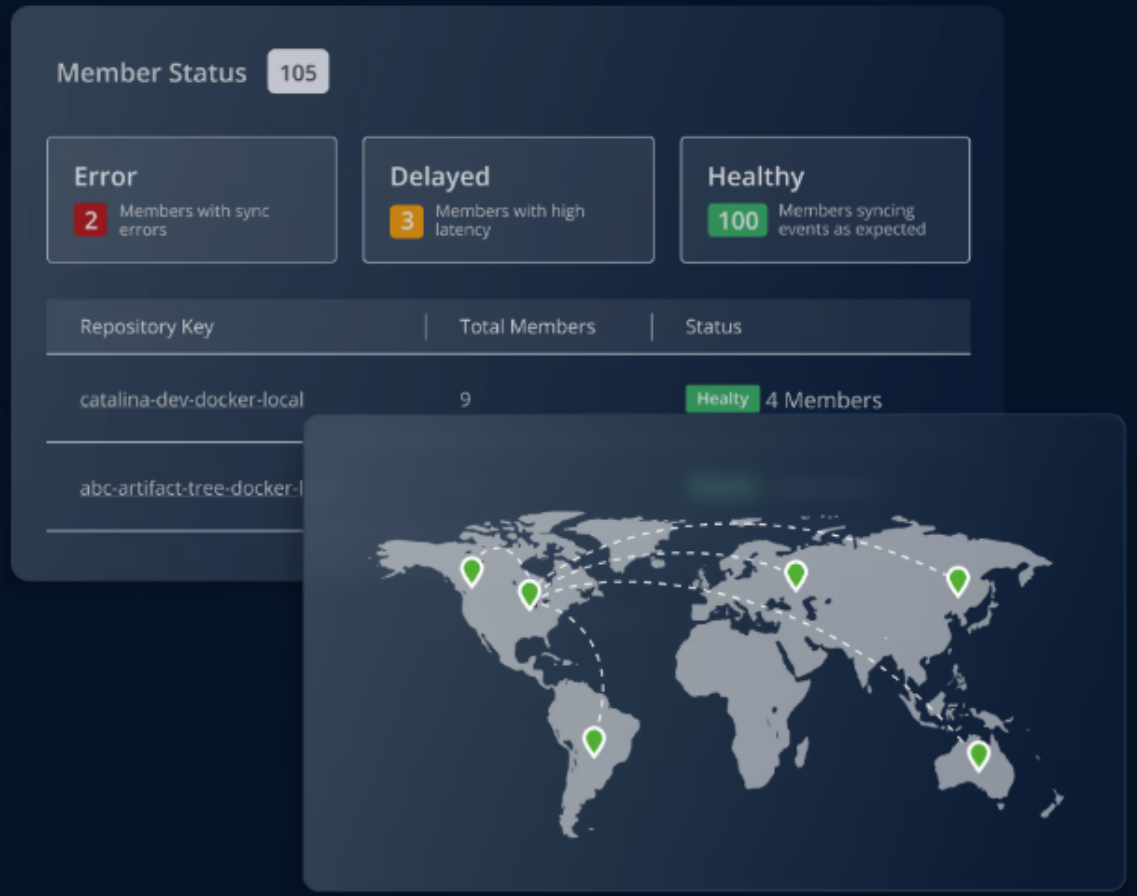
Manage every artifact and release **in a single place**

Eliminate silos and sprawl with the artifact manager that sets the bar for universality.

- ✓ 40+ natively supported package and file types
- ✓ 100s of ecosystem integrations
- ✓ Multiple installation and deployment options

What is Artifactory?

Universal | Scalable | Secure | Automated | Performance



undefined

Effortlessly scale your SDLC across multiple sites

Deploy across any topology and environment strategy to support growing global development and deliver ultimate uptime & resiliency.

- ✓ Automated bi-directional repository sync
- ✓ Control access to all services from one location
- ✓ Project-based resource management
- ✓ Managed, multi-cloud, hybrid, and Self-Hosted deployment support

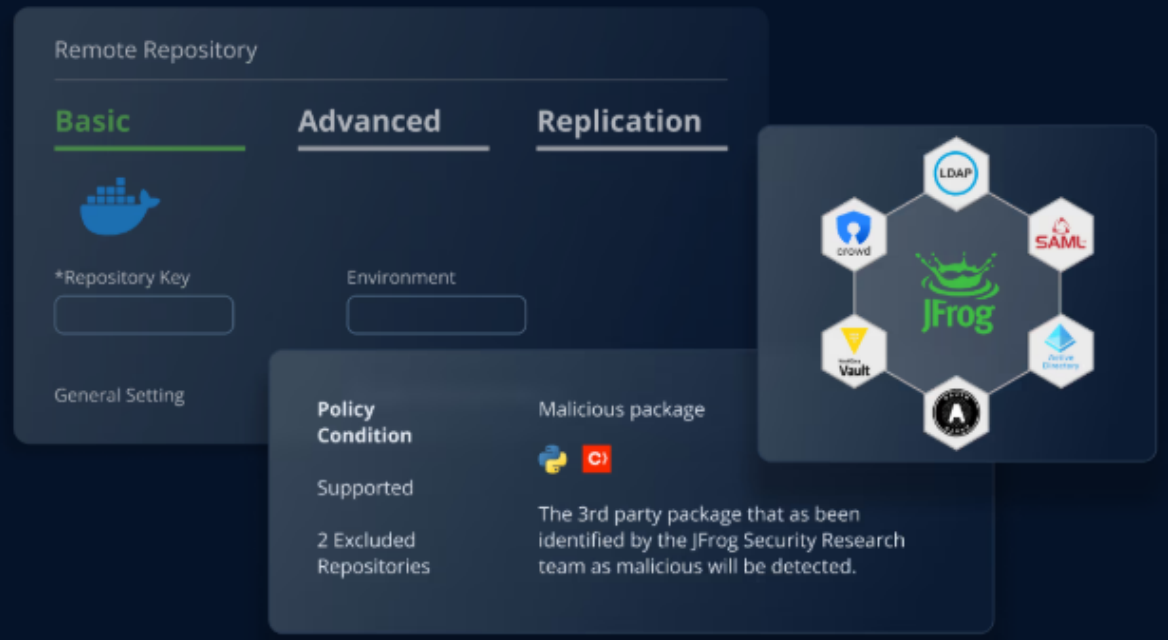
What is Artifactory?

Universal | Scalable | Secure | Automated | Performance

Lay the foundation for a secure software supply chain

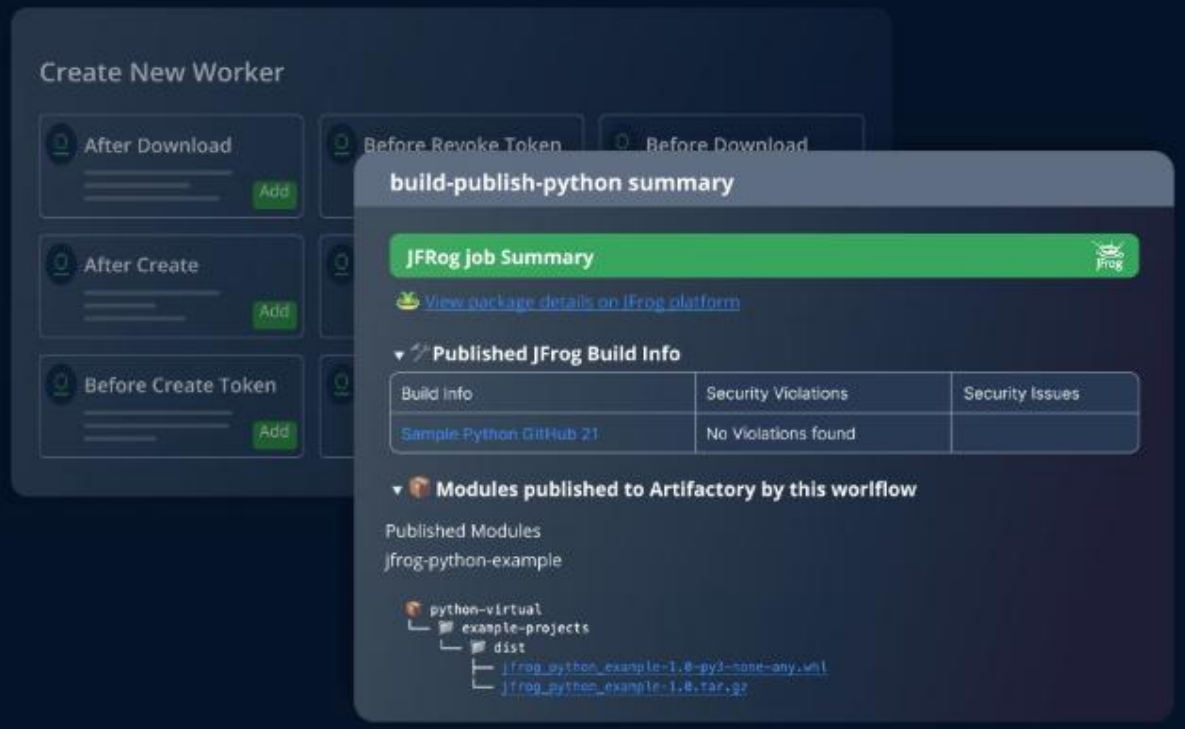
Secure your supply chain from a single place and control how packages are brought in, used, and accessed across your organization.

- ✓ Proxy public registries, preventing direct download from internet
- ✓ Robust RBAC and governance policies
- ✓ Enterprise platform security (SSO, PrivateLink, etc.)
- ✓ Identify and block vulnerable packages from use



What is Artifactory?

Universal | Scalable | Secure | Automated | Performance



Improve build speeds and **accelerate CI/CD pipelines**

Cache dependencies and builds close to their consumption points to ensure fast availability. Take advantage of multiple options to power workflow automation.

- ✓ Dependency and build caching
- ✓ Open-source JFrog CLI
- ✓ Rest APIs for nearly everything
- ✓ Custom plugins with robust event triggers

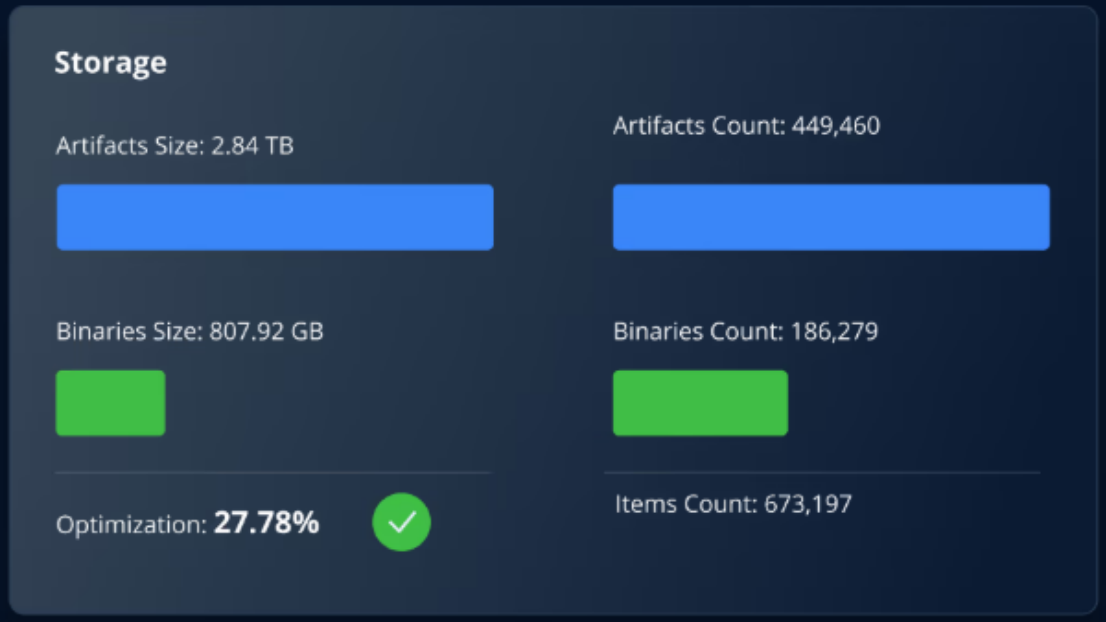
What is Artifactory?

Universal | Scalable | Secure | Automated | Performance

Deliver enterprise performance while optimizing costs

Streamline your workflows by preventing duplicates, removing unneeded artifacts, and archiving non-operational assets kept for regulatory purposes.

- ✓ Checksum optimized storage and replication
- ✓ Automated cleanup and asset archival
- ✓ Location based DNS routing



Over 7,500 DevOps Teams Trust JFrog

Serving over 80% of the Fortune 100

Mercedes
Monster
Redbox
Yunex Traffic
Cars.com
FFF Enterprises
Workiva
Deloitte
Spot

"We wanted to figure out what can we really use instead of having five, or six different applications. Is there anything we can use as a single solution? And Artifactory came to the rescue. It turned out to be a one-stop shop for us. It provided everything that we need."

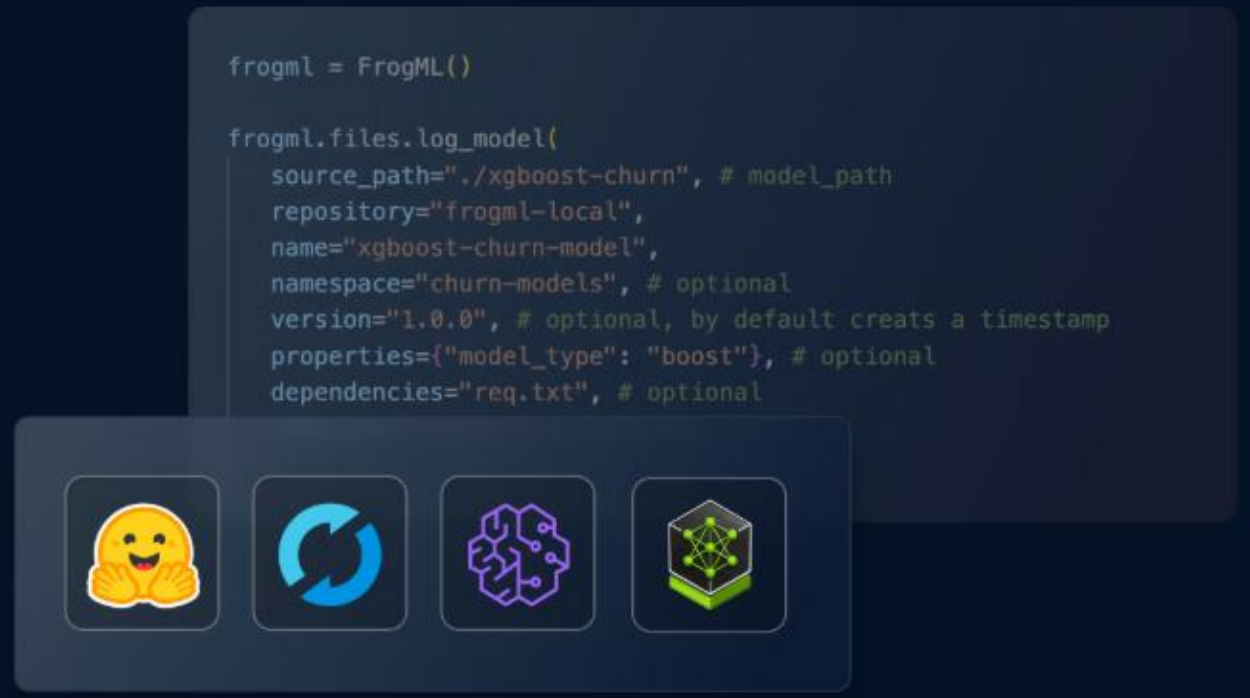
Keith Kreissl
Principal Developer, Cars.com

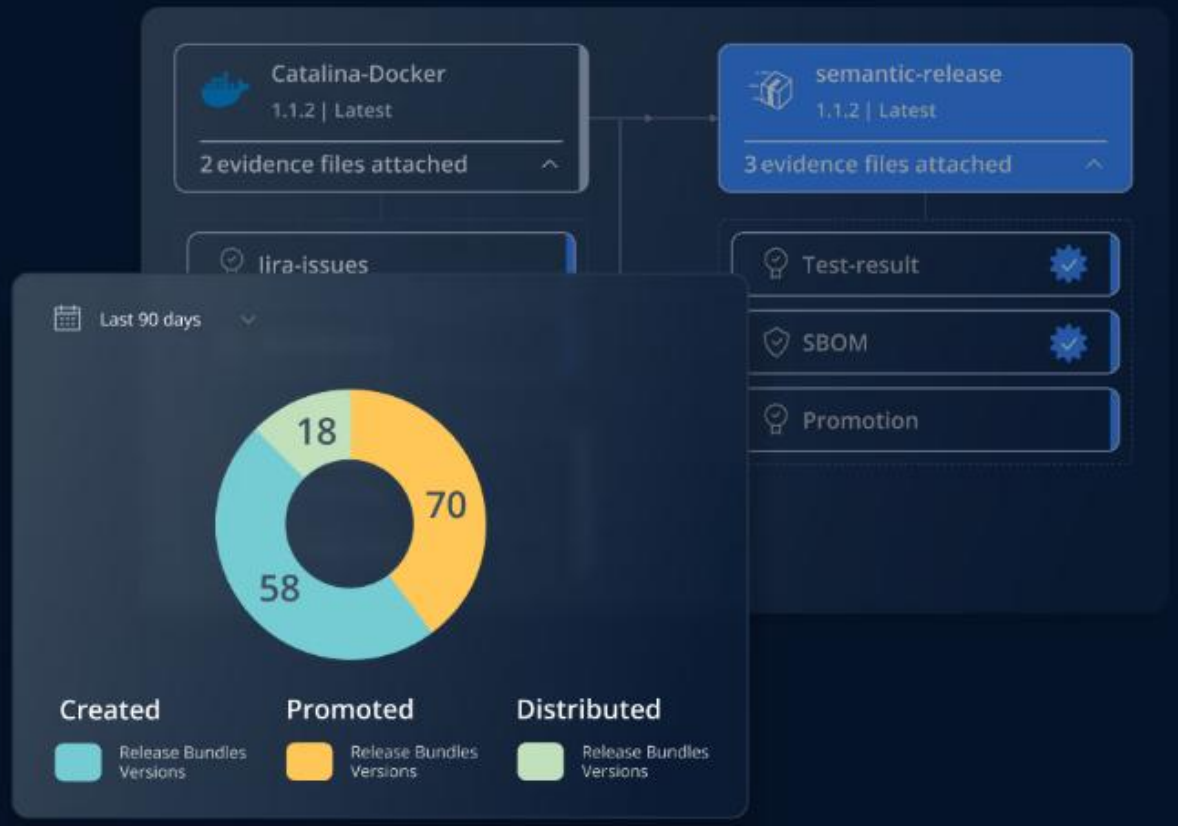
Bring ML models into your secure software supply chain

Treat models like a package to version, manage and secure AI/ML models in a way that makes sense for stakeholders across your org.

- Proxy Hugging Face and Nvidia NGC
- Manage private models with a simple SDK
- Model scanning via JFrog Xray

Explore JFrog For MLOps





Simplify auditing and governance across the SDLC

Manage, collect, and attest every action taken to bring a piece of software to life - from coding, to testing, to sign offs.

- Immutable release context, tracking, and auditing
- Security policy and promotion gating
- Attestation and evidence capture

Control Releases | **Collect Evidence**

Supercharge your developer experience

Empower developers to find the best packages for their projects, live in their IDE, and not have to change anything about the way that they work.

Find packages for use in projects

IDE plugins and native package command support

Single URL resolution

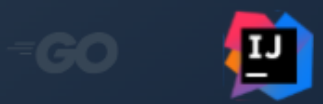
undefined

Search for an open source packages and CVEs

You can search for over 7 million packages and CVEs worldwide










Powering enterprise software development lifecycles

Artifactory serves as a mission-critical piece of infrastructure to accelerate, secure and create visibility into your development teams.

The Trusted Source for Curated Artifacts

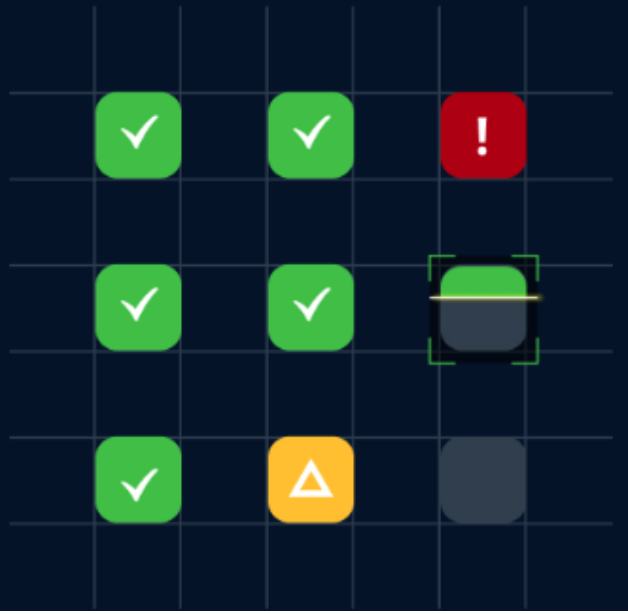
Enhance the Integrity, Reliability, and Speed of CI/CD

Ensure Constant Shared Access for Distributed Teams

Serve Components to Production Runtimes

Curate an approved set of artifacts available for use by global development teams. Mitigate risk by proactively identifying and preventing open source security issues before development even starts. Improve productivity by keeping a locally cached set of artifacts always available to the developers who need to access them.

Learn More



Accelerate AI & Software Delivery with Built-In Security

Explore Other JFrog Products

JFrog Distribution

Distribute securely to any data center or production environment

JFrog Curation

Seamlessly curate software packages and ML models

JFrog Xray

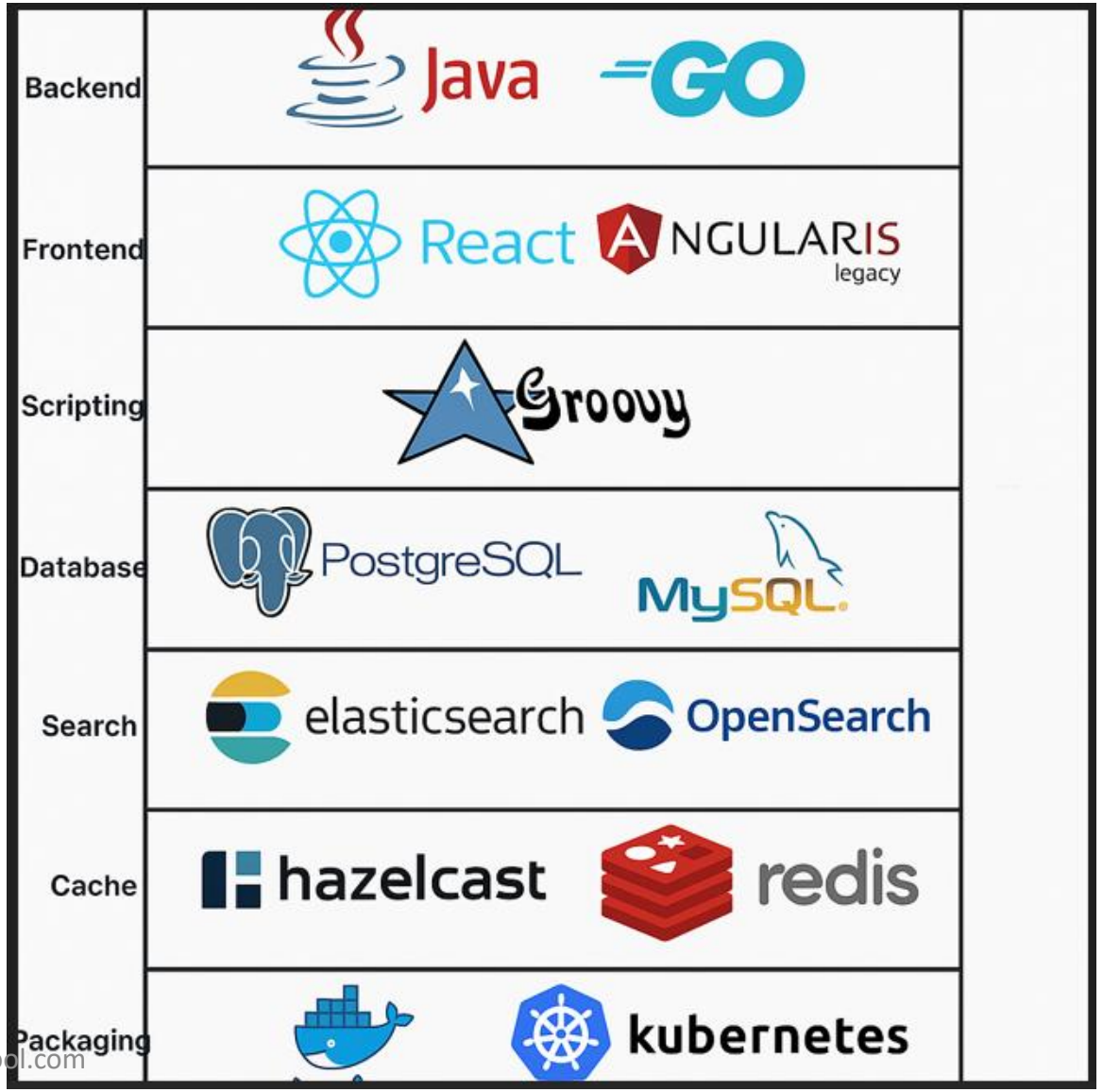
Integrated SCA for software artifacts and ML models

JFrog ML

Build, Train, Secure, Deploy, Serve and Monitor ML Models and GenAI

Artifactory Technology

- **Backend:** Java (Spring Boot), Go (for microservices)
- **Frontend:** React + TypeScript (UI), AngularJS (legacy)
- **Scripting:** Groovy
- **Database:** PostgreSQL (primary), MySQL/Oracle supported
- **Search:** Elasticsearch/OpenSearch
- **Cache:** Hazelcast/Redis
- **Packaging:** Docker + Kubernetes for deployment\



Types of Offering

- JFrog Artifactory Container Registry Community Editon (C/C++)
- JFrog Artifactory OSS
- JFrog Artifactory Pro
- JFrog Artifactory Pro X
- JFrog Artifactory Enterprise X
- JFrog Artifactory Enterprise+

Types of Offering & Licenses

- JFrog Artifactory Container Registry Community Editon (C/C++)
- JFrog Artifactory OSS
- JFrog Artifactory Pro
- JFrog Artifactory Pro X
- JFrog Artifactory Enterprise X
- JFrog Artifactory Enterprise+

Types of Offering & Licenses

EDITION	LICENSE TYPE	FEATURES	RESTRICIONS
Community Edition (C/C++)	Open Source	C/C+ package types, Generic repositiories	No database support
OSS	Open Source	Package support Database support	
Pro	Commercial	Package support Database support	
Pro X	Commercial	Security & compliance feature set	
Enterprise X	Commercial	Advanced storage capabilities Advanced security	
Enterprise+	Commercial	Advanced distribution capabilities	

PRO

25 GB Base Consumption | Community Support

Starting at:

$150 / Month Buy on Marketplace >

Buy Now

- Universal Binary Repository
- Release Lifecycle Management
- Unlimited Docker Hub Pulls
- Complete Container Registry
- Cloud-native Managed Platform

Additional Consumption Available >

ENTERPRISE X

125 GB + 1000 ML Credits Base Consumption | SLA Support | 99.9% Uptime

Starting at:

$950 / Month Buy on Marketplace >

Get Started

Everything in Pro, plus:

- Globally Federated Repositories
- Enterprise Access Control (SSO)
- Enhanced SCA & Model Security
- AI/ML Lifecycle Management
- AI/ML Serving
- One Platform Experience with GitHub

Additional Consumption Available >

Featured Add-Ons

Most Popular

ENTERPRISE +

Custom Consumption | High Touch Support | Strategic Multi-year Contracts

Custom Pricing:

Let's Talk

Contact Us

Everything in Enterprise X, plus:

- Global Access Federation
- Software Distribution
- Distributed Edge Nodes
- Advanced Traffic Management
- Third Party Evidence Collection
- Multi-cloud and Shadow AI/ML Deployments
- Streaming Feature Store for ML

Featured Add-Ons

Container Registryy Community idition (C/C++)	OSS	Pro	Pro X	Enterprise X
Edition	Free	Open Source	Commercial	Enterprise+
License Type	Supports Docker/OCI ind. C/C++ Binaries	30+ package t. Local, remote and Virtual	Commerciaal	All Enterprise X Ex tectures
Key Features	Basic Repository Management	Local, Remote, and Virtual Repositories	All Pro features	All Enterprise X for Large organizations with global teams
Best For	Supports Dacv/OCI images a contaí- ner registry or C/C++ package	Basic CI/CD integrations	All Pro X features • HA clustering • Multi-site replli- cation	Large organizations with global teams and productiongrade requirements
Best For	Small teams o projects that ' a container rég7- stry or C/C++ package store	Small-to-mid-size teams requiring multi-format artifact repo aot	Teams neending security scanning into their arti- fact repositóry	Enterprise managing software supply cha- ins at scale with full DevOps-DevSeoS

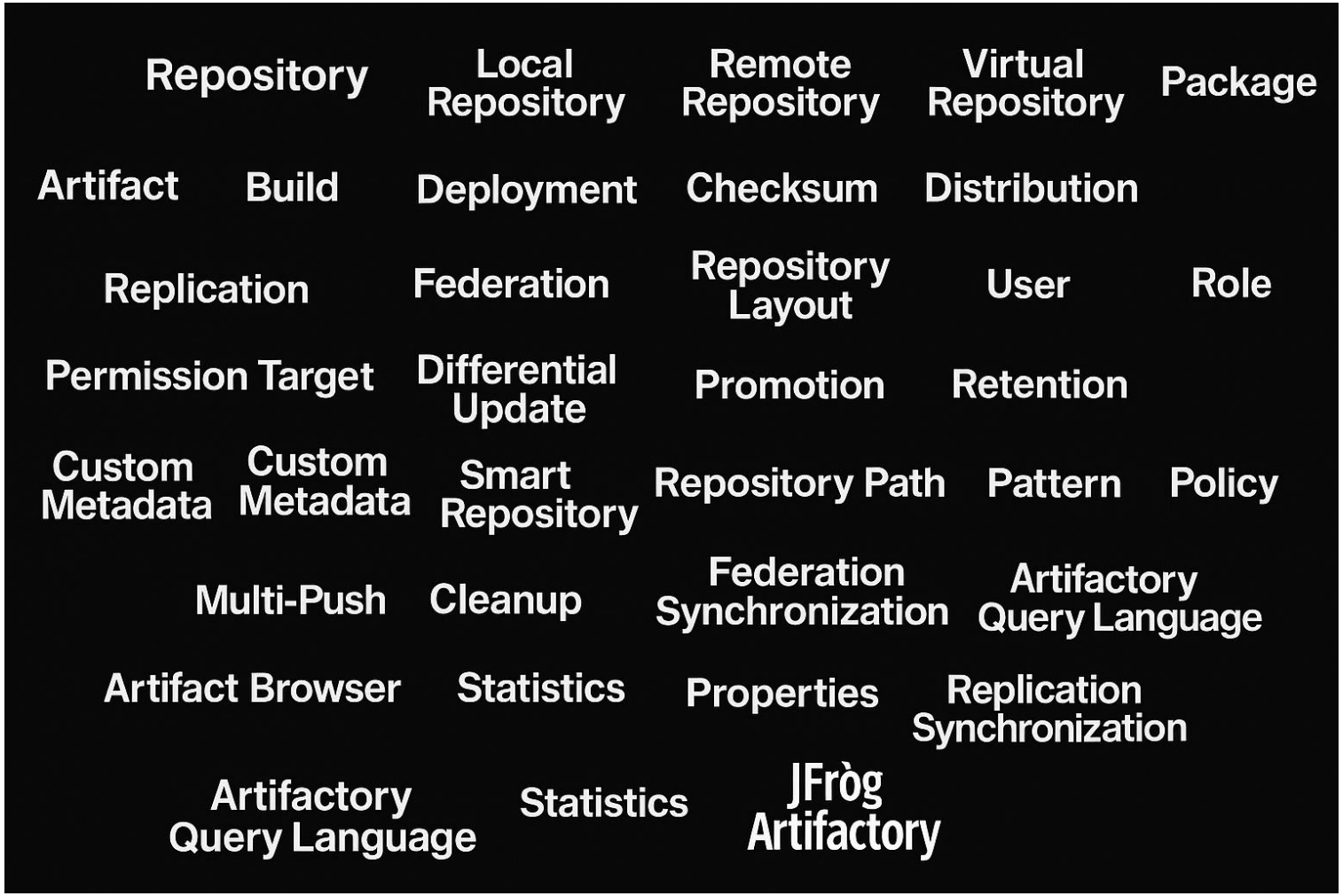
https://www.devopsschool.com/blog/artifactory-jfrog-artifactory-editions-and-their-license-types/

Latest Version

Latest Releases:

Self-Hosted	Cloud
7.117.7	7.118.2

ARTIFACTORY TERMINOLOGY



Terminology: Core Concepts

Artifact: Any binary file produced during build or development (e.g., .jar, .war, .apk, .docker).

Repository: A storage location in Artifactory where artifacts are kept.

Package Type: Defines the technology the repository supports (e.g., Maven, npm, Docker, PyPI).

Build: A collection of artifacts and metadata generated by a CI/CD process.

Artifact Metadata: Additional information (version, build info, properties) attached to artifacts.

Terminology: Repository Types

Local Repository: Hosts internally created artifacts.

Remote Repository: Acts as a caching proxy for external repositories (e.g., Maven Central, npm registry).

Virtual Repository: Aggregates multiple local and remote repositories into a single endpoint.

Federated Repository: Synchronizes repositories across multiple Artifactory instances.

Terminology: Repository Layout

Repository Path: The full URL/path to access artifacts within a repository.

Checksum: A hash value (SHA1, MD5) used to verify artifact integrity.

Immutable Artifacts: Artifacts that cannot be overwritten once deployed.

Terminology: Access & Permissions

RBAC (Role-Based Access Control): System for controlling user permissions.

Users & Groups: Entities with access to repositories based on roles.

Access Token: Token-based authentication method for API and automation.

Terminology: Metadata & Querying

Properties: Key-value metadata pairs attached to artifacts.

AQL (Artifactory Query Language): Query language for searching and automating tasks within Artifactory.

Build Info: Metadata about builds (git commit, build number, environment).

Terminology: Security & Compliance

Xray: JFrog’s security scanning tool for vulnerabilities and license compliance.

SCA (Software Composition Analysis): Analyzing dependencies for security issues.

CVE (Common Vulnerabilities and Exposures): Database of known security vulnerabilities.

Terminology: Distribution

Replication: Syncing repositories across instances.

Promotion: Moving artifacts from one repository (e.g., dev) to another (e.g., prod).

Distribution Repository: Repository used for distributing signed, immutable releases.

Terminology: Storage & Infrastructure

Build Pipeline: Automated sequence of steps for building and deploying software.

JFrog CLI: Command-line tool to automate interaction with Artifactory.

REST API: Provides programmatic access to Artifactory functions.

Terminology: Advanced Concepts

Filestore: The physical storage backend for artifacts (e.g., local disk, NFS, cloud object storage).

Checksum Storage: Deduplicates artifacts by storing only unique binaries identified by checksum.

HA (High Availability): Clustered Artifactory instances for redundancy and load balancing.

Terminology: Distribution

Edge Node: Remote Artifactory node for distributed teams/sites.

CDN Distribution: Use of Content Delivery Network for artifact delivery at scale.

Immutable Release Bundle: Signed, versioned artifact collection for distribution.

Artifactory Repository Types

Local

Internal artifacts.

Remote

External proxy with caching.

Virtual

Unified endpoint combining multiple repos.

Federated

Cross-instance synchronization for multi-site deployments.

Artifactory Repository Types: Local Repository

1. Local Repository

• **Definition:**

A repository hosted within your Artifactory instance for **internally created artifacts**.

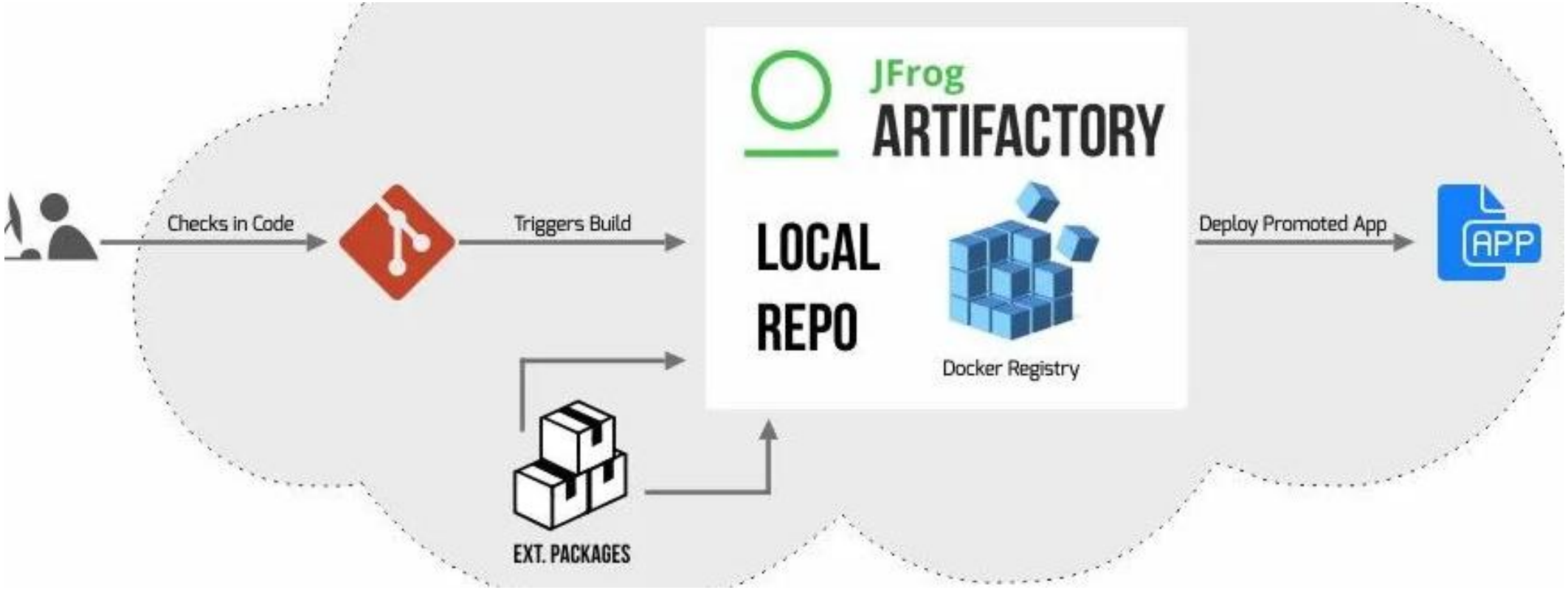
•**Usage:**

- Stores build outputs, binaries, and internal packages.
- Ideal for artifacts you produce and want to manage securely.

•**Example:**

- Hosting your organization's Maven or npm packages for internal use.

Artifactory Repository Types: Local Repository



Artifactory Repository Types: Remote Repository

2. Remote Repository
Definition:
Acts as a caching proxy for external repositories such as Maven Central, npm registry, PyPI, etc.

Usage:

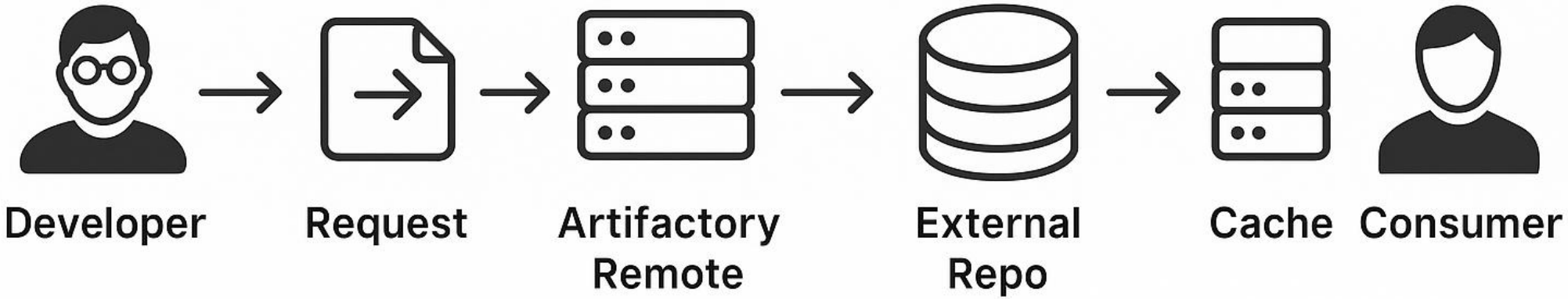
Fetches artifacts from remote sources and caches them locally.

Improves build speed and provides offline access.

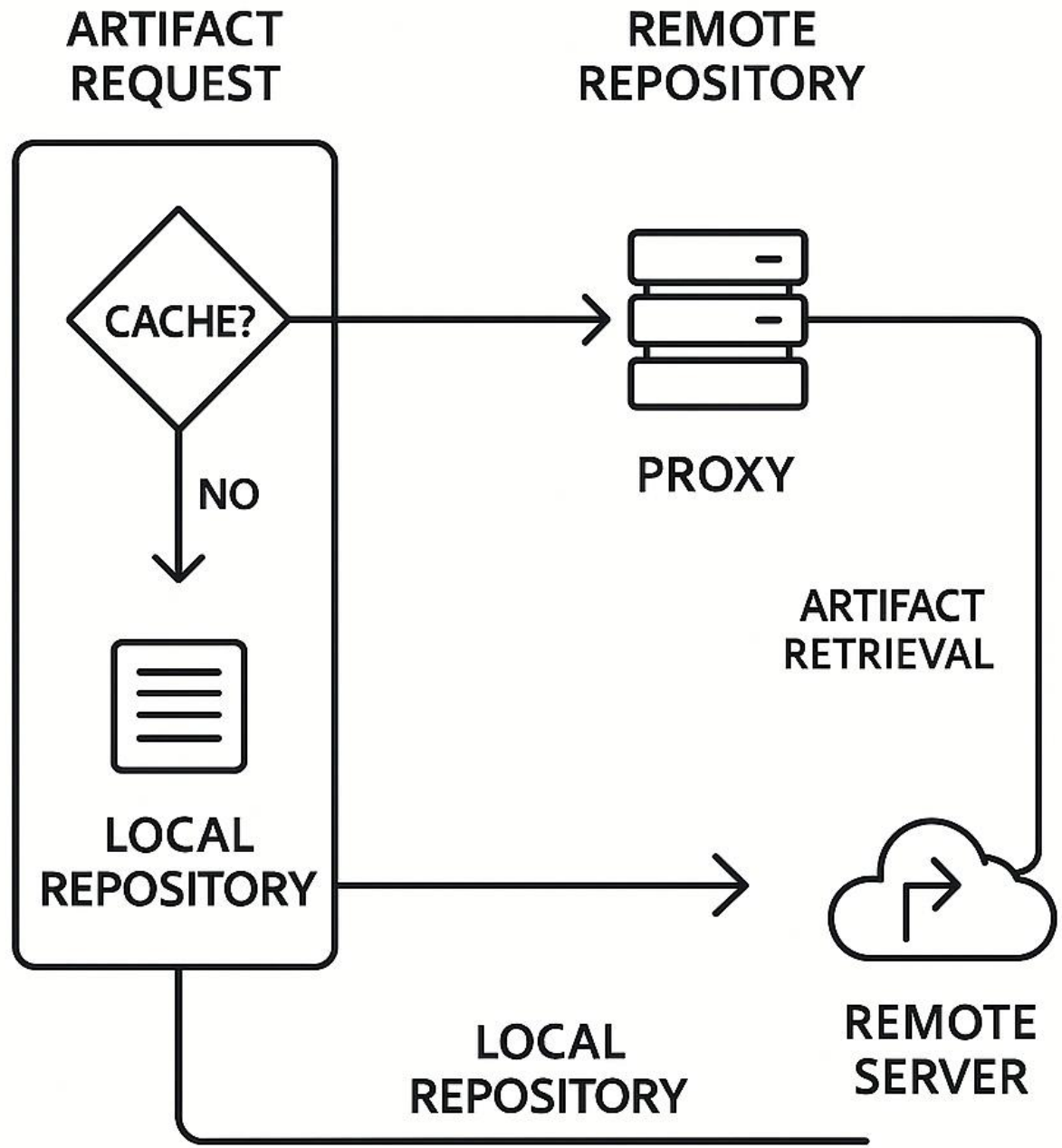
Example:

Remote repository for https://repo.maven.apache.org/maven2.

Remote Repository



Remote Repository



DevOpsSchool
Lets Learn, Share & Practice DevOps

Artifactory Local Repository vs. Remote Repository

Aspect	Local Repository	Remote Repository
Definition	A repository hosted inside your Artifactory instance to store **internally created artifacts**.	A proxy repository in Artifactory that caches **external artifacts** from remote repositories (e.g., Maven Central, npm Registry).
Purpose	Manage and version control artifacts you produce in-house.	Provide faster access and caching for third-party dependencies while acting as a single access point.
Artifact Source	Artifacts are **deployed directly** to Artifactory (e.g., via CI/CD pipelines or manual uploads).	Artifacts are **fetched from external repositories** when requested and cached locally.
Use Case	- Hosting your own Maven, npm, PyPI, Docker images.- Storing internal builds and release candidates.- Controlling distribution of internal software.	- Speed up builds by caching external libraries.- Provide offline access to external dependencies.- Control external dependency versions and security.
Caching	Not applicable – stores artifacts you push to it.	Automatic caching: First request downloads from external repo, stores locally for future use.
Access Pattern	Developers and CI/CD pipelines **upload** and **download** artifacts directly.	Developers and CI/CD pipelines **request** external artifacts; Artifactory fetches and caches them.
Examples	- libs-release-local (Maven internal releases)- docker-local (private Docker images)- npm-local (internal npm packages)	- maven-central-remote- npmjs-remote- pypi-remote
Security Control	Full control over access, metadata, and versioning since artifacts are internal.	Provides control over external dependencies via caching, access rules, and checksum validation.
Promotion	Supports promotion between repositories (e.g., dev → staging → prod).	Typically used as a dependency source; not for promoting internal builds.
Offline Availability	Always available since artifacts are local.	Becomes available offline **after initial fetch** due to caching.
Best For	Internal builds, proprietary software, custom packages.	External dependencies, reducing internet calls, ensuring reproducibility of builds.

Artifactory Repository Types: Virtual Repository

3. Virtual Repository
Definition:
A logical repository that aggregates multiple local and remote repositories under a single URL endpoint.

Usage:

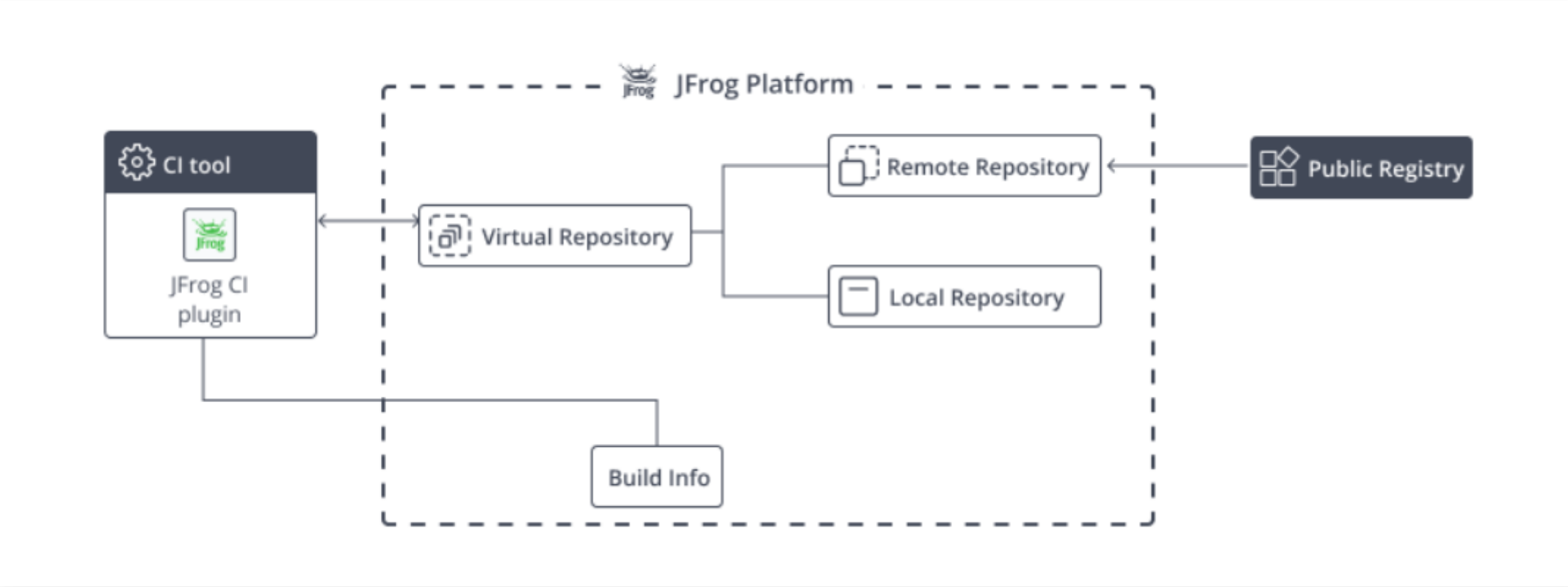
Simplifies dependency resolution by exposing a single URL to developers and CI/CD pipelines.

Controls repository priority and routing.

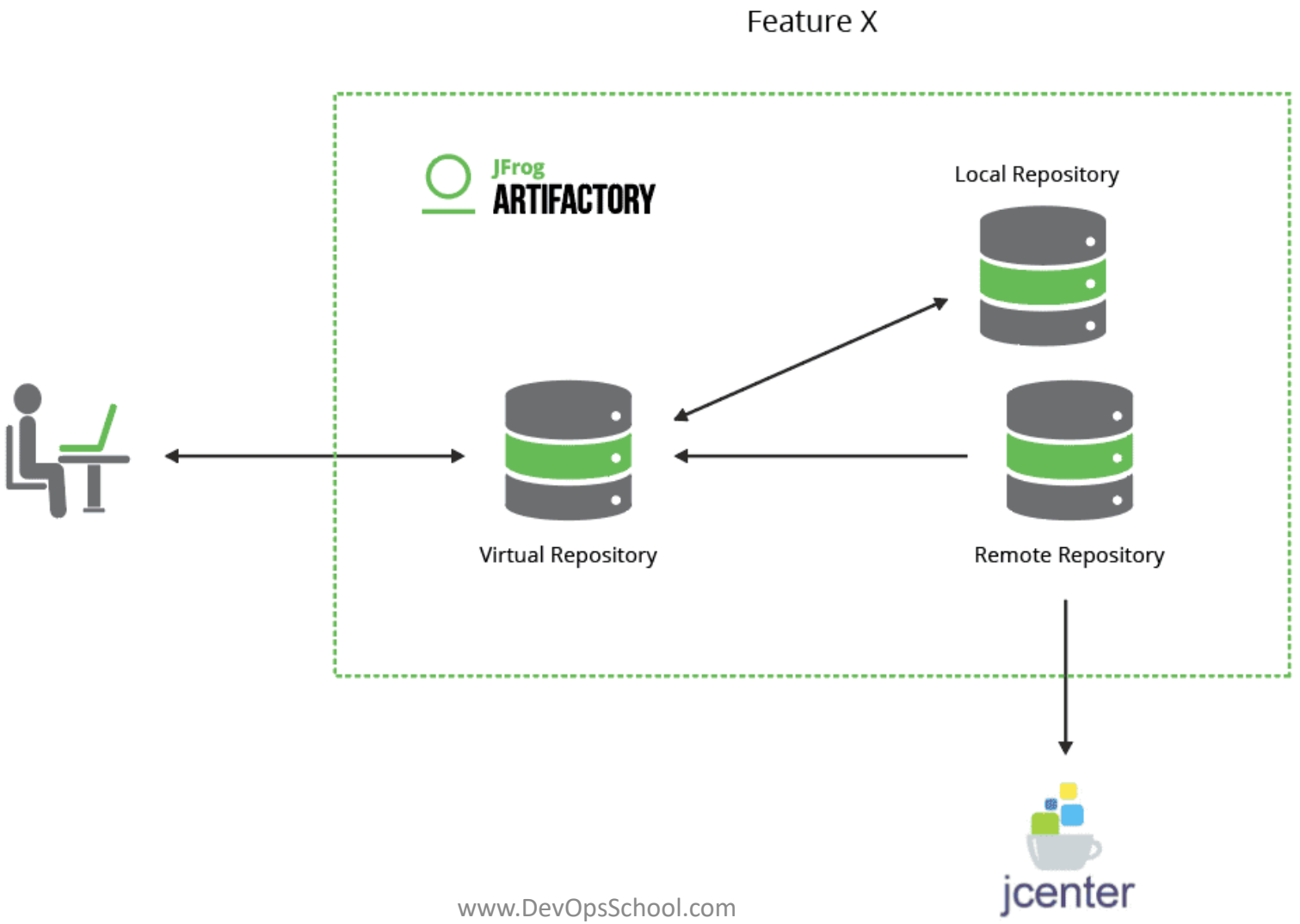
Example:

A virtual repository combining internal Maven packages + Maven Central proxy.

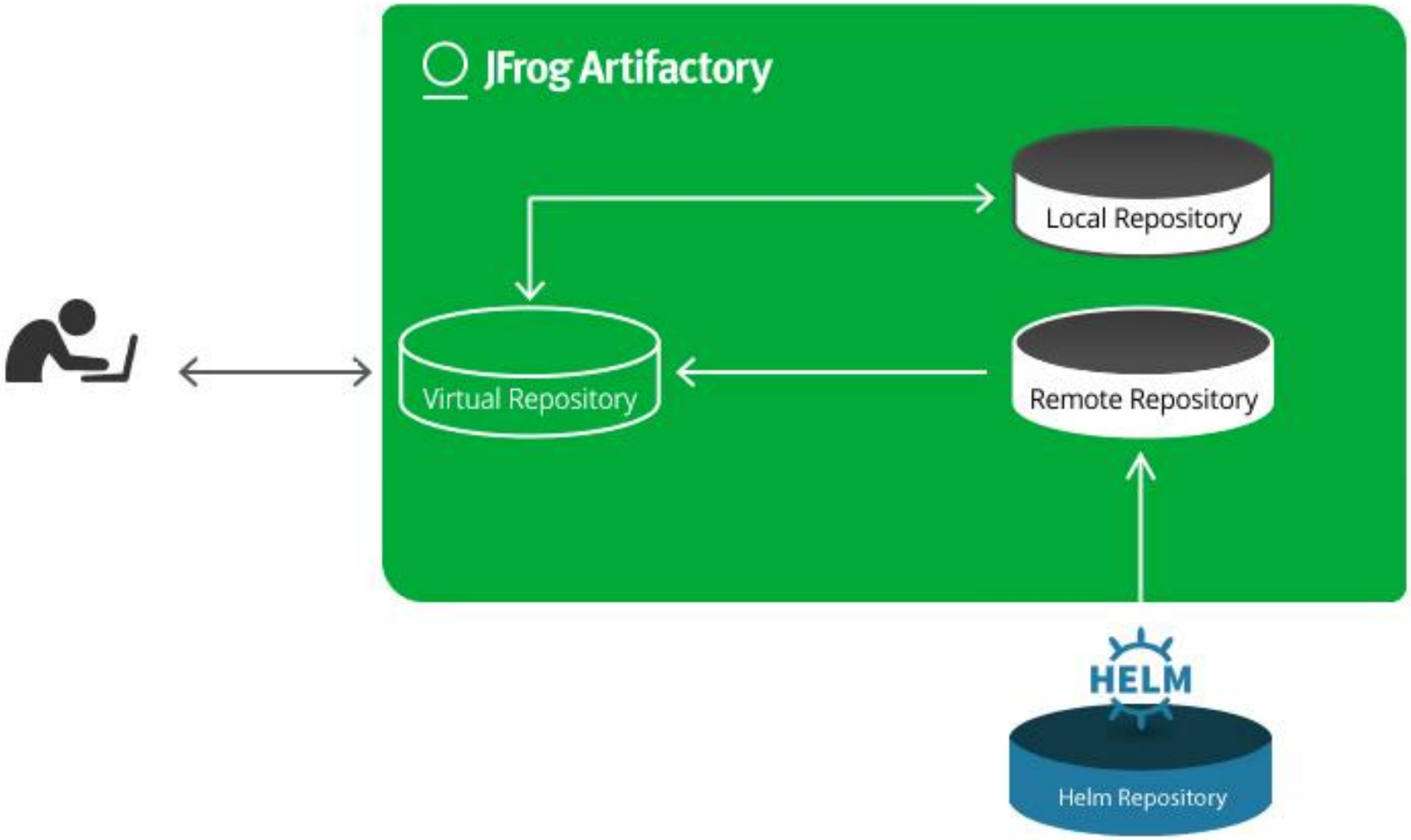
Local & Remote & Virtual



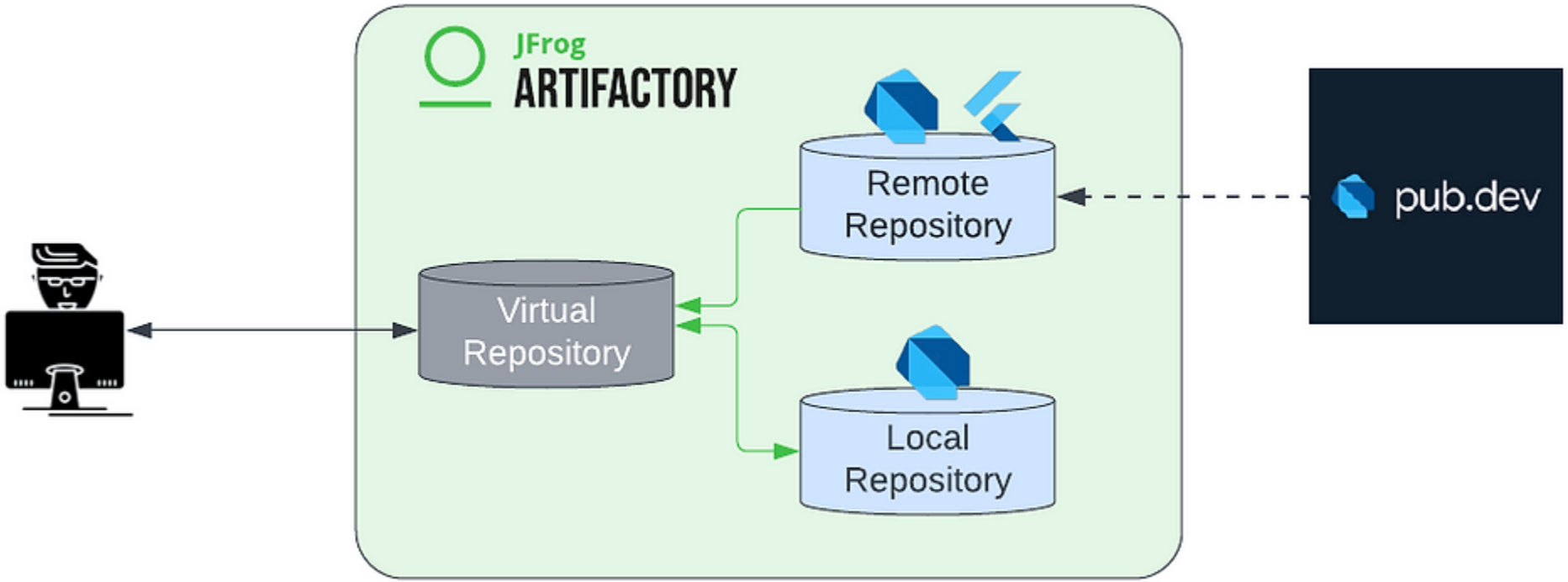
Local & Remote & Virtual



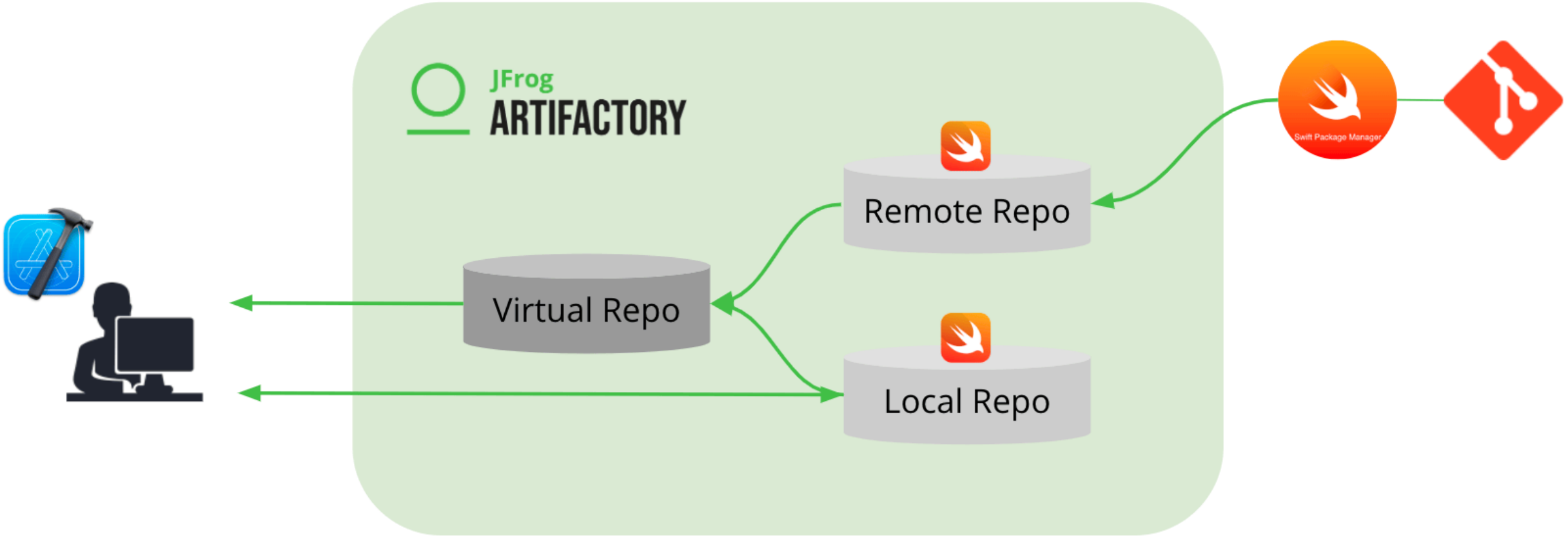
Local & Remote & Virtual



Local & Remote & Virtual



Local & Remote & Virtual



DevOpsSchool
Lets Learn, Share & Practice DevOps

Local & Remote & Virtual

Aspect	Local Repository	Remote Repository	Virtual Repository
Definition	Hosts **internally created artifacts** within Artifactory.	Acts as a **caching proxy** for external repositories (e.g., Maven Central, npm).	Provides a **unified endpoint** that aggregates multiple local and remote repositories.
Purpose	Store and version artifacts you build in-house.	Provide cached access to external dependencies and reduce build time.	Simplify dependency resolution by exposing a single URL for multiple repositories.
What It Can Do	✅ Upload artifacts manually or via CI/CD pipelines. ✅ Store proprietary/internal packages. ✅ Full control over metadata, access, and promotion. ✅ Always available since artifacts are internal.	✅ Cache external artifacts for offline use. ✅ Control external dependency versions. ✅ Speed up builds by reducing external calls. ✅ Verify artifact integrity via checksums.	✅ Combine multiple repos under one endpoint. ✅ Provide consistent repository URL to developers and pipelines. ✅ Control repository priority/order for resolution. ✅ Hide complexity of multiple repositories.
What It Cannot Do	❌ Cannot fetch artifacts from external repositories automatically. ❌ Cannot act as a proxy.	❌ Cannot host internally built artifacts (only caches what it fetches). ❌ Cannot promote artifacts between environments.	❌ Cannot store artifacts itself (only routes to local/remote repos). ❌ Cannot cache external artifacts directly.
Artifact Source	Internal builds and manual uploads.	External upstream repositories.	Combination of both local and remote sources.
Usage Example	Hosting your organization's Maven or npm internal packages.	Caching dependencies from Maven Central or npmjs registry.	A single "all-in-one" endpoint for devs combining internal and external Maven repos.
Best For	Internal builds and proprietary packages.	External dependency caching and reproducibility.	Simplifying access for developers and CI/CD pipelines.

Artifactory Repository Types: Federated Repository

Definition:
Allows repositories to synchronize content across multiple Artifactory instances, enabling multi-site and hybrid cloud setups.

Usage:

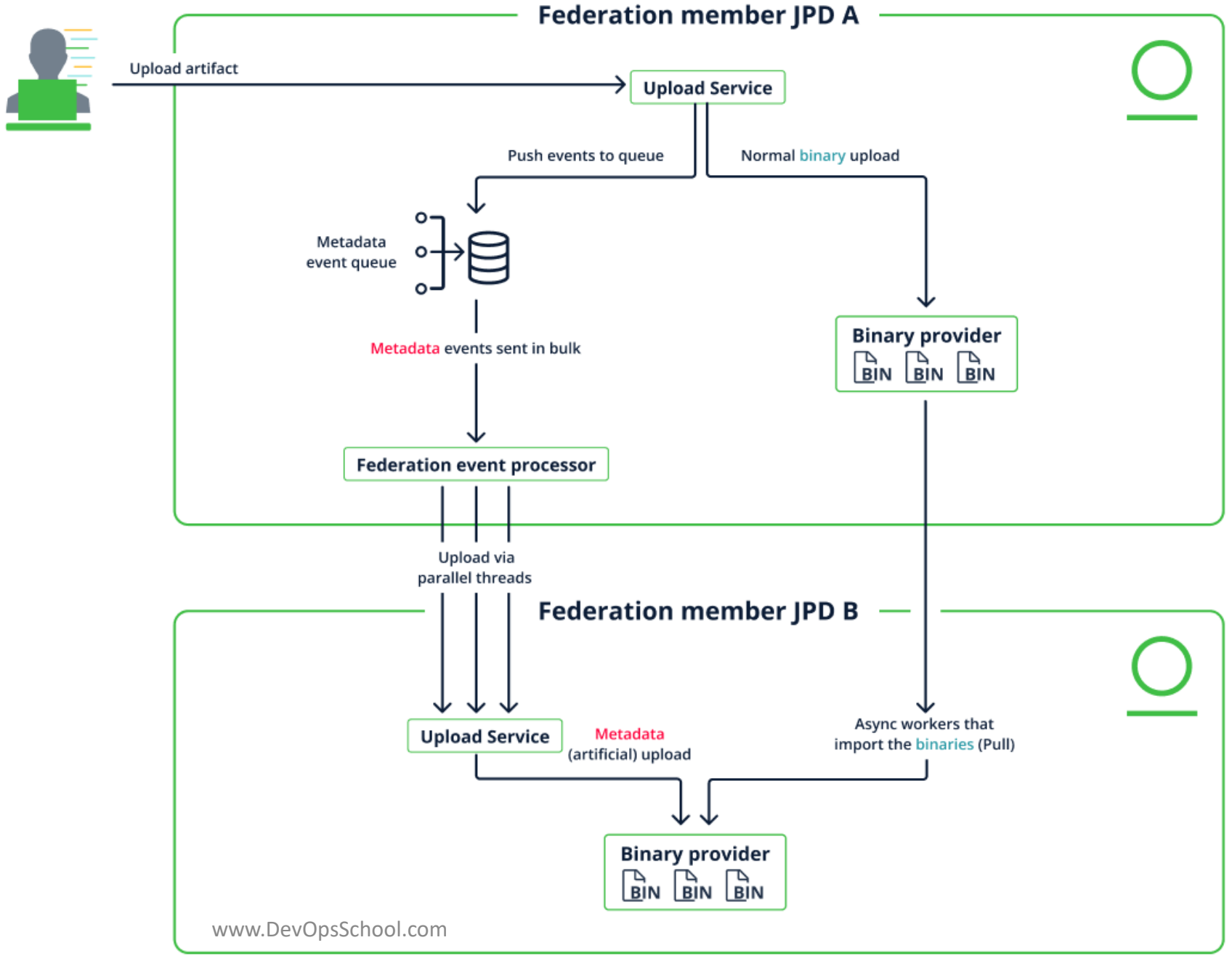
Ideal for global enterprises with distributed teams.

Ensures artifact availability and consistency across locations.

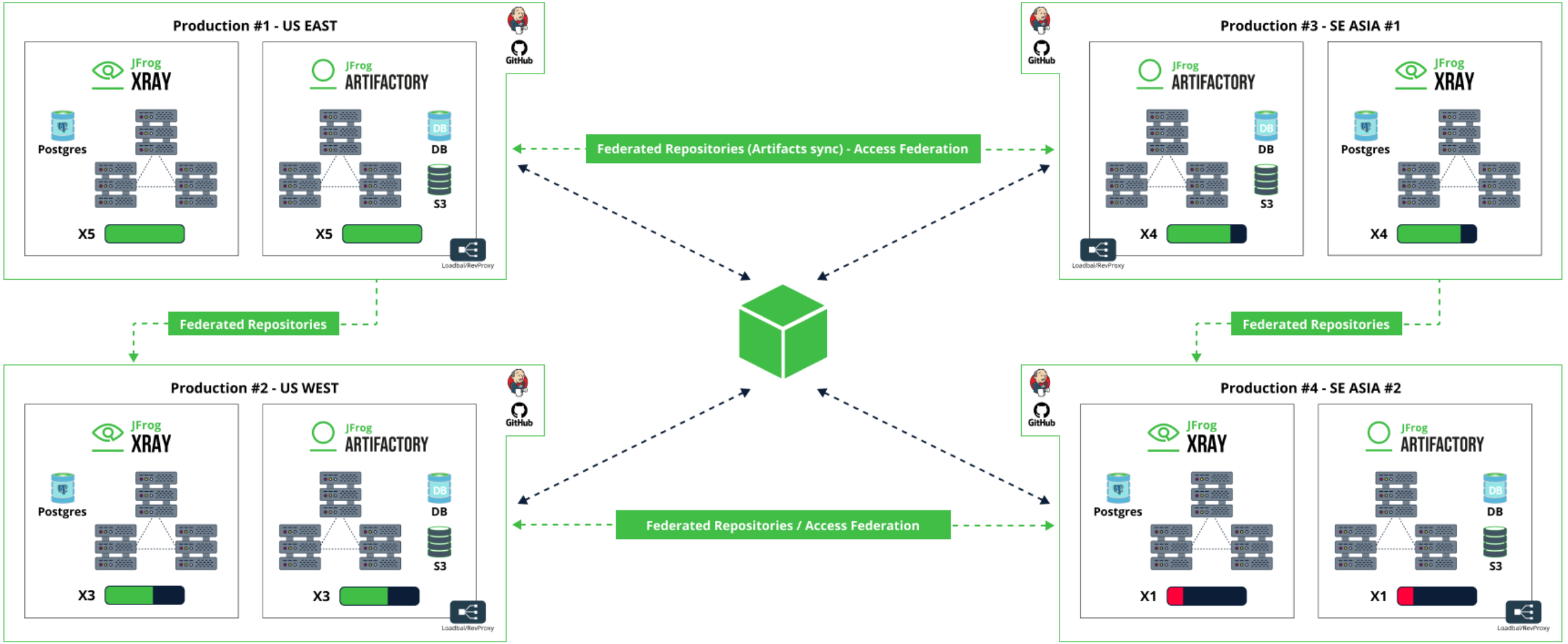
Example:

Syncing an enterprise's libs-release repository across multiple data centers.

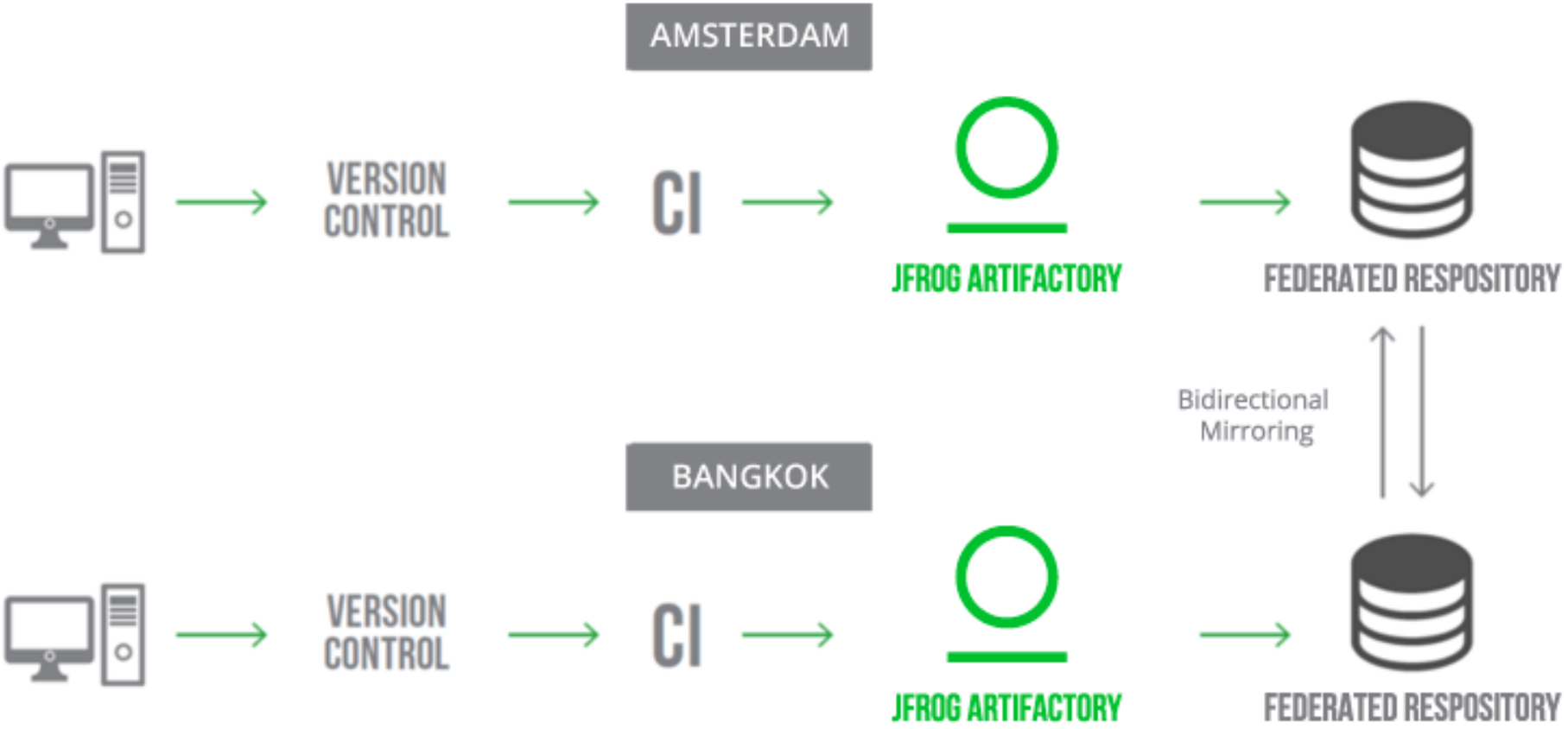
Artifactory Repository Types: Federated Repository



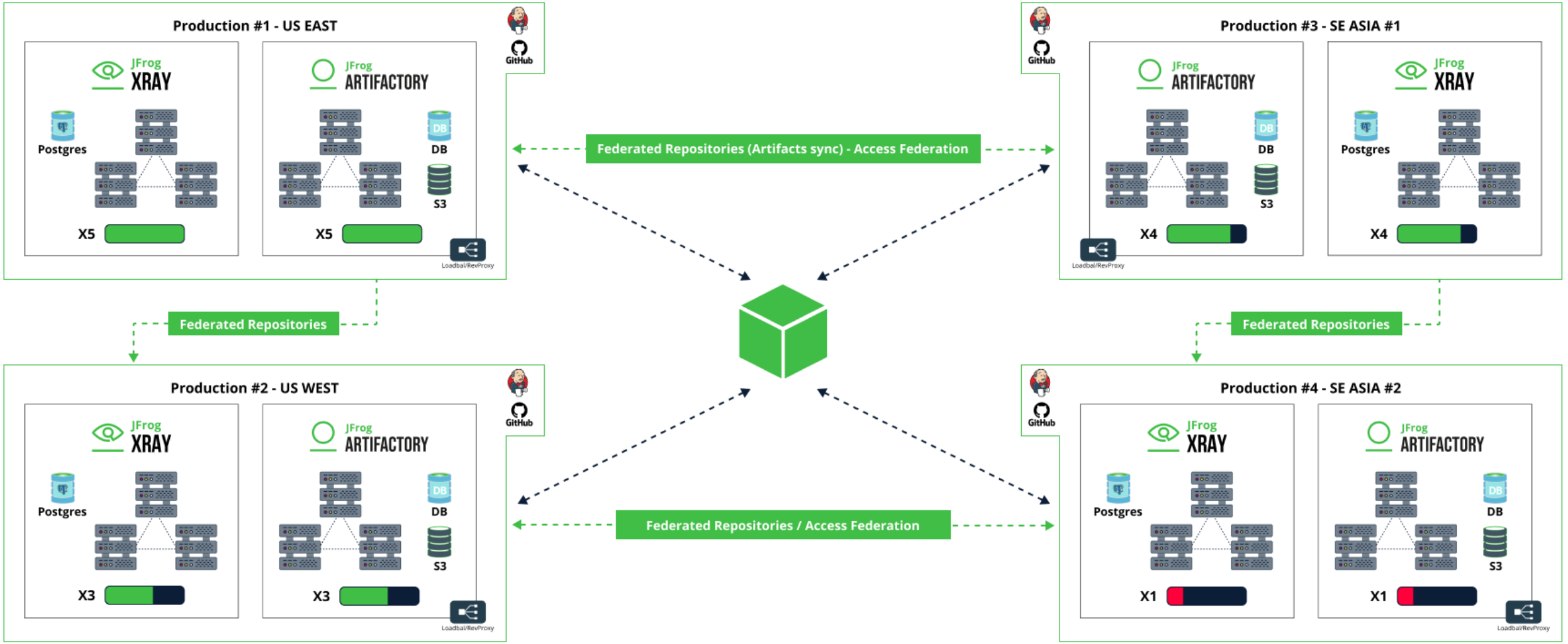
Artifactory Repository Types: Federated Repository



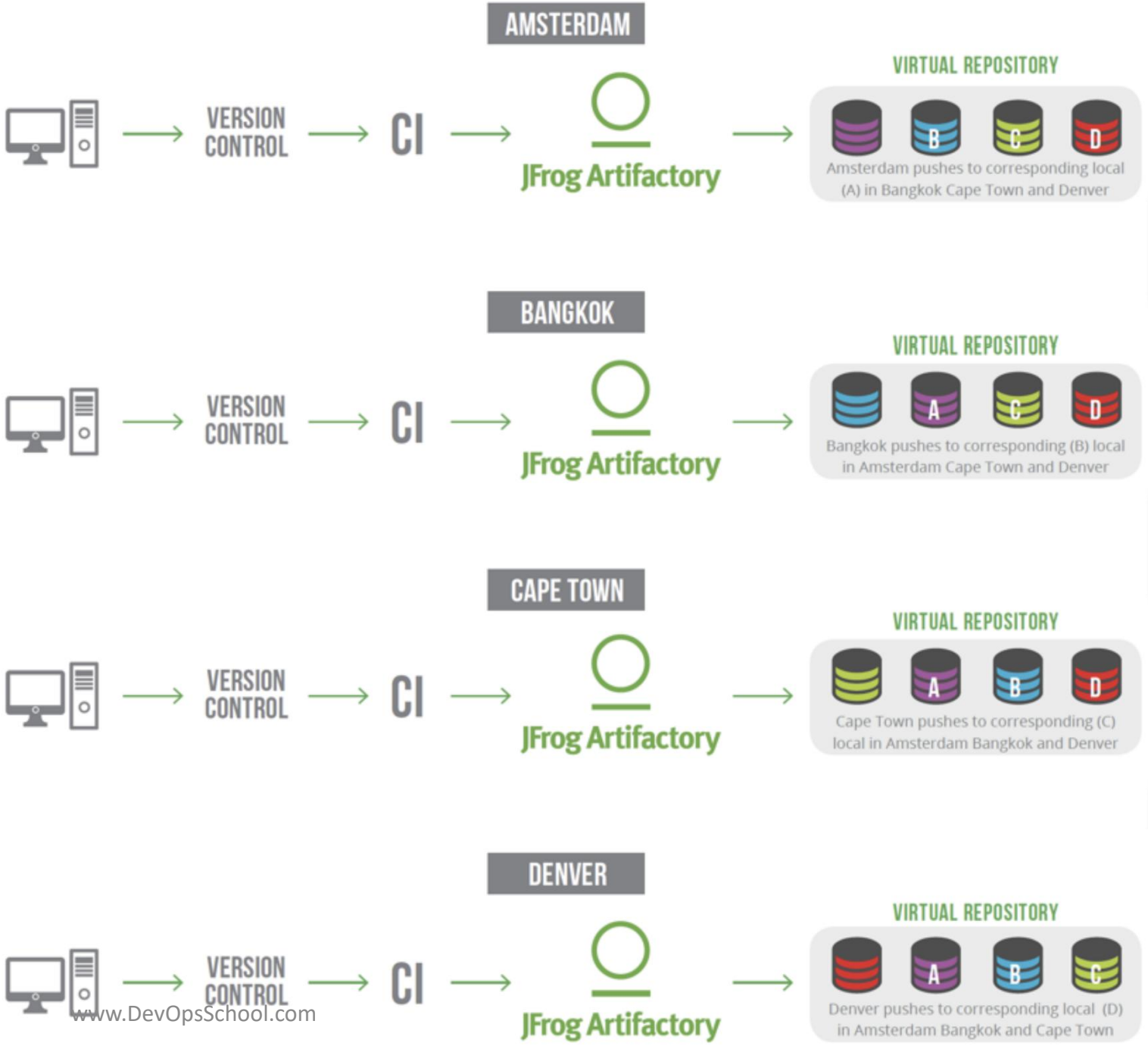
Artifactory Repository Types: Federated Repository



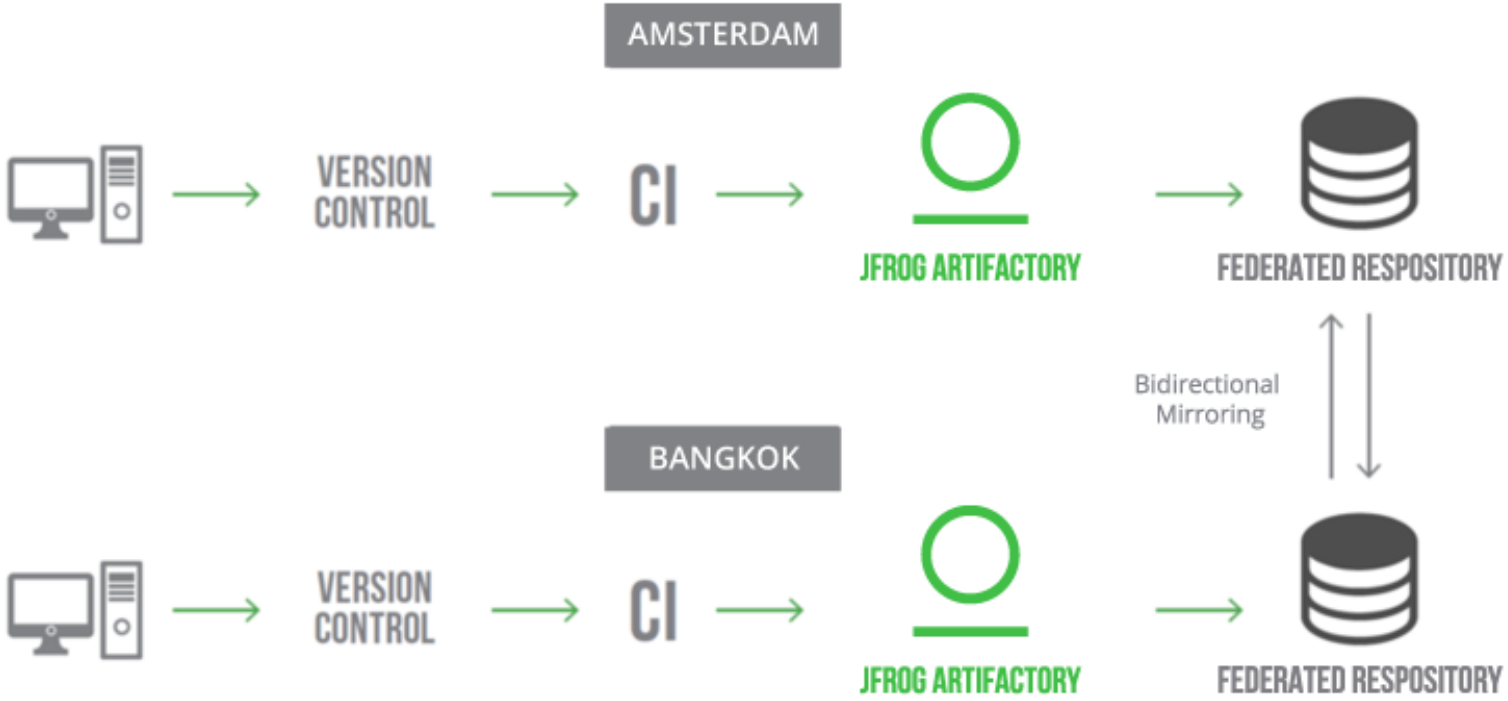
Artifactory Repository Types: Federated Repository



Federated Repositories: REPLICATION



Federated Repositories: BIDIRECTIONAL SYNCHRONIZATION



Artifactory Package Types

Category	Supported Package Types
🔹 **Language & Application Packages**	- Maven, Gradle, Ivy, SBT (Scala) - npm (Node.js) - PyPI / Poetry / Twine (Python) - NuGet (C#/.NET) - RubyGems (Ruby) - Cargo (Rust) - Composer (PHP) - CocoaPods (iOS/macOS) - Pub/Dart - Hex (Elixir) - CRAN (R) - Go modules - Bower (web libraries) - Chef cookbooks - Puppet modules
🔹 **Container & Cloud Native**	- Docker / OCI images (including Helm OCI charts) - Helm Charts (OCI-based) - ORAS artifacts via OCI (e.g., WASM to OCI, BuildKit/BuildX packaged formats
🔹 **Infrastructure & VM**	- Alpine Linux packages (`.apk`) - Debian/YUM RPM packages - Opkg (embedded Linux) - Vagrant boxes - Terraform / OpenTofu (module/provider and backend registries) - P2 (Eclipse plugins) - VCS repositories
🔹 **Generic & Versioned Files**	- Generic repository (untyped files) - Git LFS (Large File Storage) - Machine Learning artifacts (Hugging Face models, scikit-learn, ONNX, etc.)

	Ansible	Bower	Buildkit/Buildctl (using Docker or OCI repositories)	BuildX (using Docker or OCI repositories)	Cargo
Chef	Chocolatey (using NuGet repositories)	CocoaPods	Conan	Conda	CRAN
Debian	Docker	Generic	Git LFS	Go Registry	Gradle
Helm (Helm OCI)	Hex	Hugging Face Models & Data Sets	Machine Learning Repositories	Maven	npm
NuGet (.NET)	OCI	OpenTofu (using Terraform repositories)	Opkg	ORAS (using Docker or OCI repositories)	P2 (Eclipse)
PHP Composer	podman (using Docker or OCI repositories)	Poetry (using PyPI repositories)	Powershell (using NuGet repositories)	Pub/Dart	Puppet
PyPI (Python)	RPM (yum)	RubyGems	SBT	Swift	Terraform
Twine (using PyPI repositories)	Vagrant	VCS	WASM (using OCI repositories)	Yarn (using npm repositories)	

Thank you!

DevOpsSchool Team

 www.DevOpsSchool.com

 contact@devopsschool.com