swampUP 2022 CITY TOUR!

Introduction to DevSecOps with JFrog Xray

Maharshi Patel, Solutions Engineer, JFrog

JFROG TEAM

Maharshi Patel
Solutions
Engineer

Eric Rosa
Solutions
Engineer

AGENDA

01:05–02:45	DevOps & DevSecOps Overview
	Introduction To JFrog Xray
	Policies and Watches
	Impact Analysis
02:45-3:00	Break
03:00-04:50	Shift Left Strategy
	CI / CD Integration
	Scan Reports
	New Features
04:45–05:05	Summary, Quiz, Survey & Q/A

CLASS LOGISTICS

- Coffee Break
- Restrooms
- Course Materials
- Hands-on labs on your computer
- Survey
- Quiz
- Q&A

CLASS PREREQUISITES

- Personal
  - Basic understanding of the Software Development Lifecycle (SDLC)
  - Basic understanding of JFROG Artifactory
- Labs
  - Since a significant part of the training involves hands-on exercises. So a laptop with a browser and IDE or Terminal client.
  - Git Client installed (https://git-scm.com/downloads)
  - Clone Github repository from **https://github.com/jfrog/SwampUp2022**
    - We will be using **SU003-Intro-to-DevSecOps-with-JFrog-Xray** for today
  - Install JFrog CLI (https://jfrog.com/getcli/)
  - Docker daemon + client (https://docs.docker.com/get-docker)
  - Maven client (https://maven.apache.org/install.html)
  - NPM client (https://docs.npmjs.com/downloading-and-installing-node-js-and-npm)

LAB SETUP INFORMATION

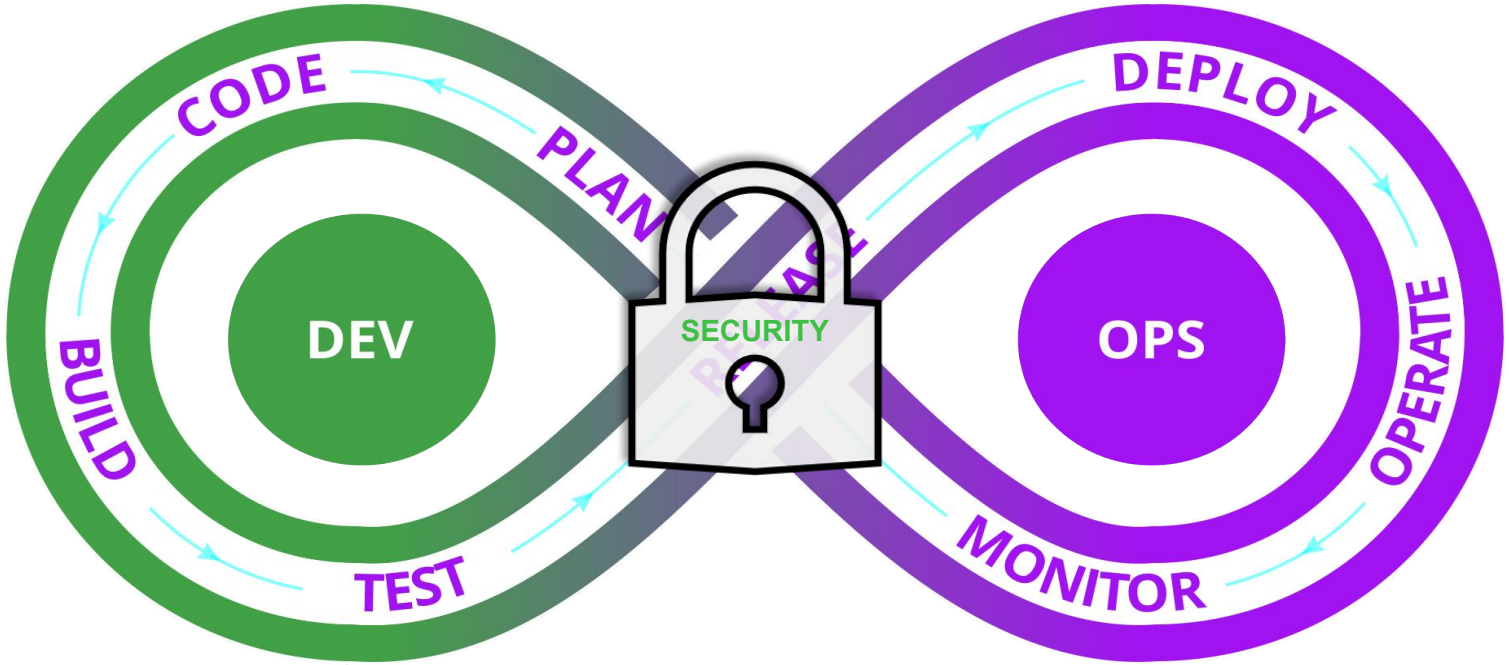
- A hands-on lab environment has already been provisioned for you to use in this class, based on your Last Name and First Initial:
- Login Information:
    - URL: https://sup009epsu**YY**.jfrog.io/
      Username: fabienlsup009epsu**YY**@jfrog.com
      Password: SwampUp2022!
    - **YY** = Student number, for example 03 or 05 or 32.
- Each user in the class will be given a student number.

swampUP 2022 CITY TOUR!

DevSecOps

JFrog
The Liquid Software Company

WHAT IS DEVSECOPS?



The philosophy of integrating security practices within the DevOps process.
#SecurityFirst culture!

How? Introducing security earlier in the life cycle of application development

WHY DEVSECOPS?

Developer Options:

- Write from scratch
- Leverage OSS/3rd party libraries

RISKS with OSS:

- Security vulnerabilities
- License compliance issues
- Performance issues

Example:

- **Equifax** - usage of Apache Struts
- **Cisco** - usage of GPL license (General Public License)
- **Log4Shell**
- **Spring4Shell**
- many more...

JFROG

The Liquid Software Company

JFROG AT A GLANCE

1,000+
EMPLOYEES

10
OFFICES
WORLDWIDE

3M+ ENTERPRISE
DEVELOPERS
DAILY

85%
OF THE
FORTUNE 100[1]

~7,000
CUSTOMERS
GLOBALLY

>300
COMMUNITY
EVENTS ANNUALLY

DEVELOPER ECOSYSTEM
JAVA/ JAVASCRIPT, C++, .NET,
VISUAL STUDIO, PYTHON ...

$207M
ANNUAL
REVENUE

DEVOPS DOZEN DEVSECOPS 2021 & 2022

DUN's 100 LIST 2021

Deloitte 2021 Technology Fast 500 Winner

SF Bay Area Best Places to Work 2021

swampUP 2022 CITY TOUR!

THE JFROG APPROACH

END-TO-END DEVOPS PLATFORM
Unify, Accelerate & Secure Your Software Delivery, From Development To Distribution

ULTIMATE SCALABILITY
Flexible, Open And Expandable By Design, That Grows With Your Scaling Needs

UNIVERSAL TECHNOLOGY
Supports 30+ Package Types and Any Tool in Your DevOps Ecosystem

CONTINUOUS SECURITY
Integrated Security Across Your Pipeline to Achieve Security and Compliance at DevOps Speed

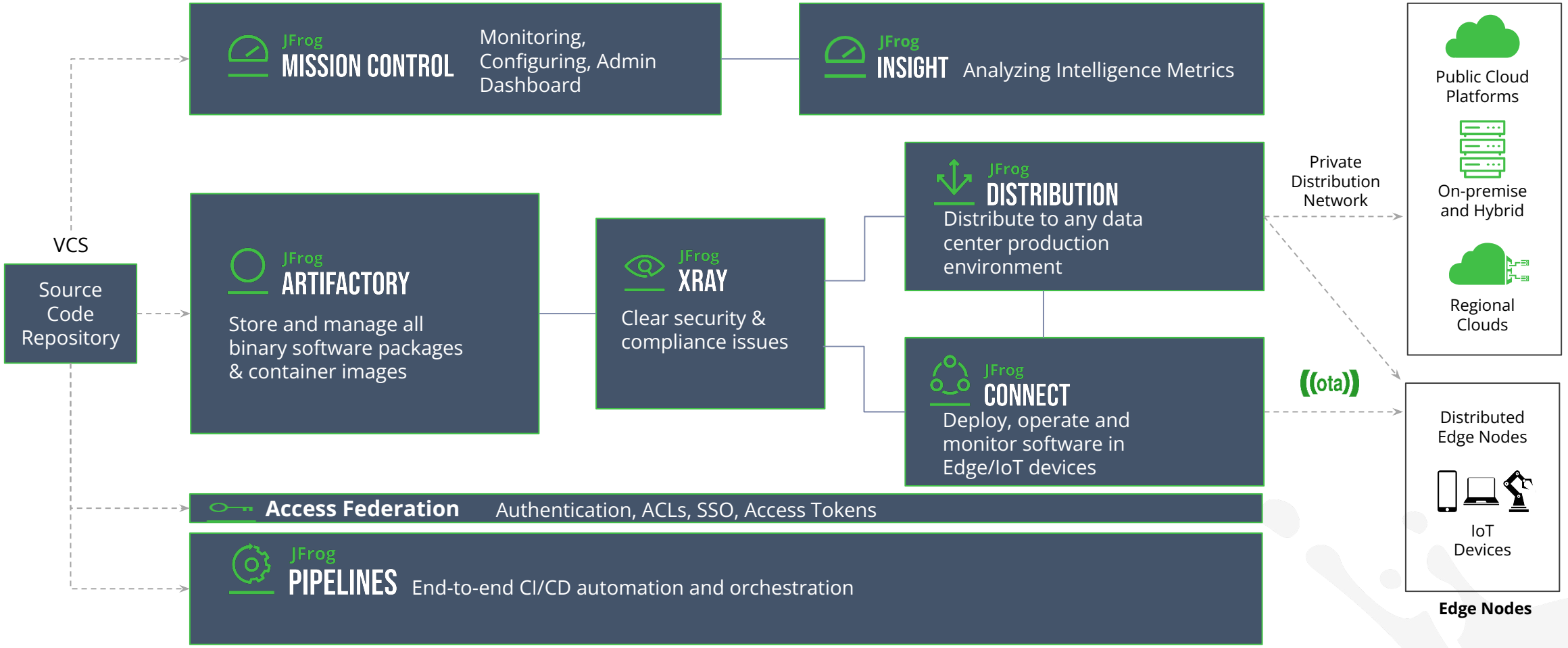
CLOUD, MULTI-CLOUD & HYBRID
Deployment Flexibility Supports Any Development Topology Including Cloud Native

INTEGRATED ECOSYSTEM
Extensive Technology, ISV and Cloud Partner Integrations, Out-of-the-Box

swampUP 2022 CITY TOUR!

THE JFROG DEVOPS PLATFORM

Unify, accelerate & secure your software delivery, from development to distribution



swampUP 2022 CITY TOUR!

JFROG XRAY

Universal Software Composition Analysis

(SCA)

JFROG XRAY | AUTOMATED SECURITY & COMPLIANCE

SCAN ANY SOFTWARE

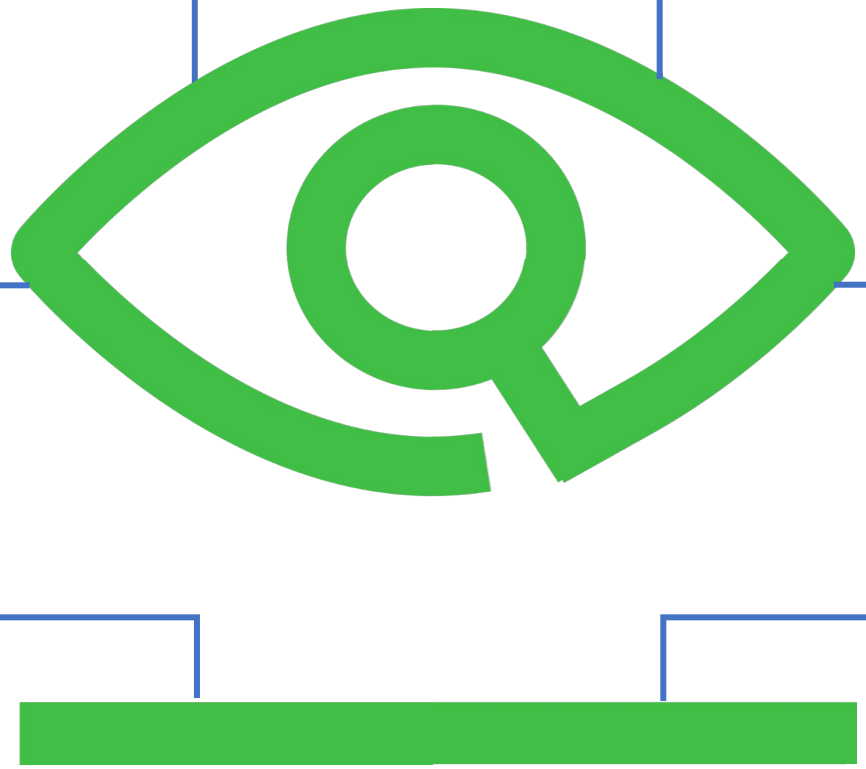
Many supported package types
From single binaries to applications
Containers

FIX WHAT MATTERS, FAST

Impact analysis

INTEGRATE IN YOUR ENVIRONMENT

Rich integrations, REST API, CLI
Automated security in CI/CD
From development to runtime



GAIN FULL VISIBILITY

Supply chain full visibility / SBOM
Vulnerabilities in all dependencies

CONTROL YOUR SECURITY POSTURE

Context-based policy
Platform management & security

DEPLOY ANYWHERE

Self-hosted (on-prem / cloud)
SaaS
Multi-Cloud
Hybrid

swampUP 2022 CITY TOUR!

XRAY SUPPORTED PACKAGE FORMATS

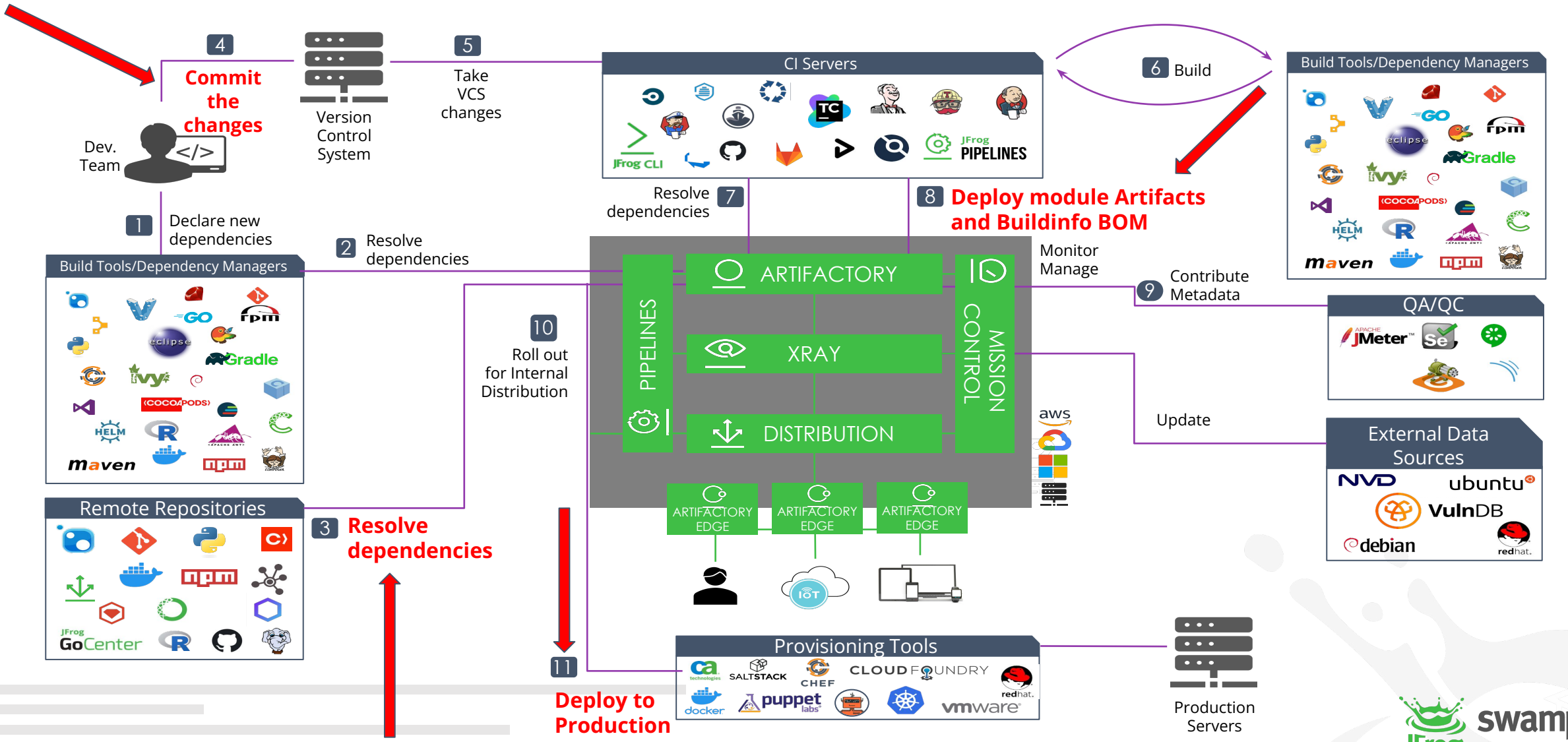
Google Distroless Images

XRAY SETUP

- Download JFrog Xray

NOTE: Applicable for Self-Managed only

WHERE DOES XRAY FIT?



XRAY
ARCHITECTURE

The Liquid Software Company

JFROG XRAY TERMINOLOGY

INDEXING — The process of Analyzing the Artifact and listing the components and Artifact layers

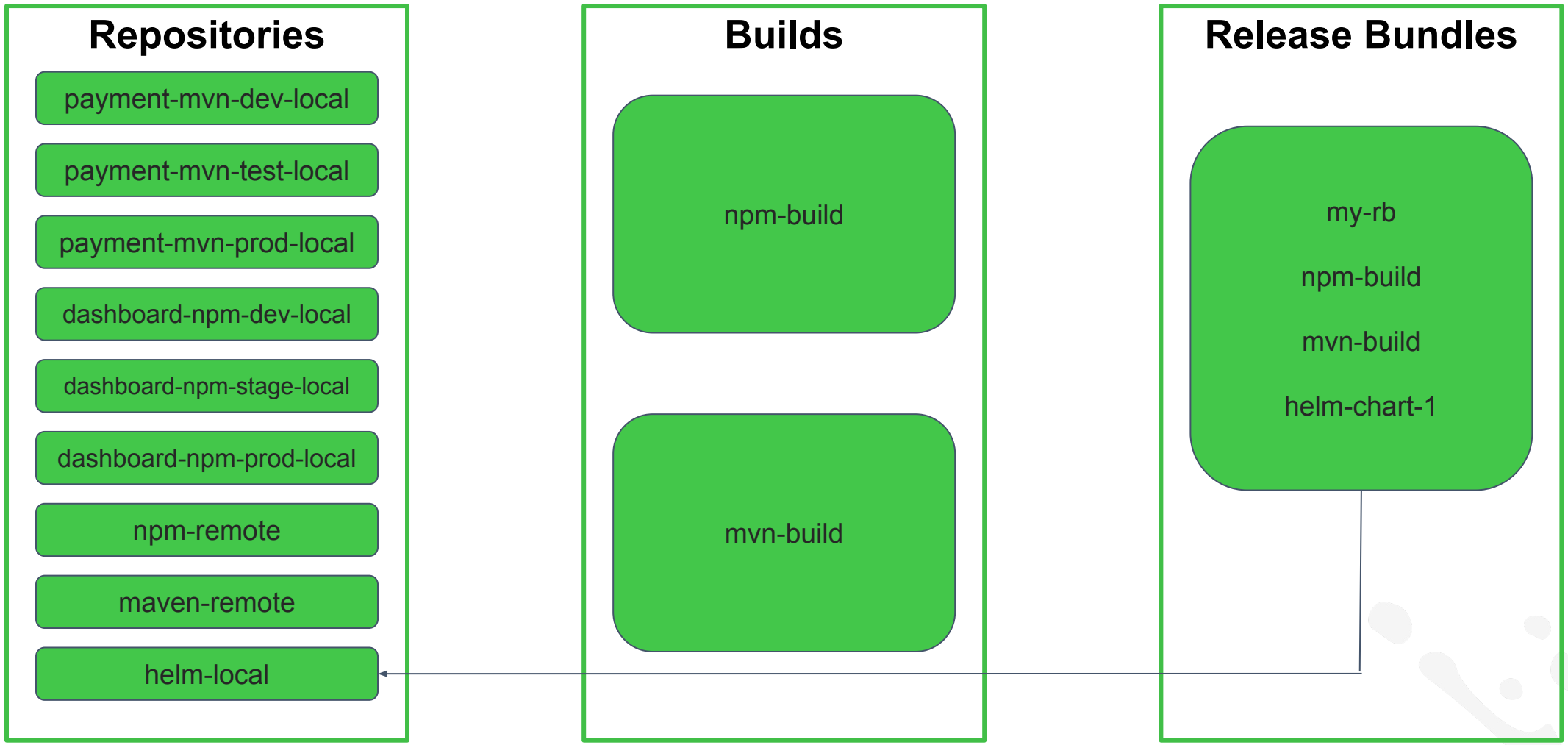
RESOURCES — Repositories, Builds and Release bundles

POLICIES — Contextless set of rules with set of automatic actions

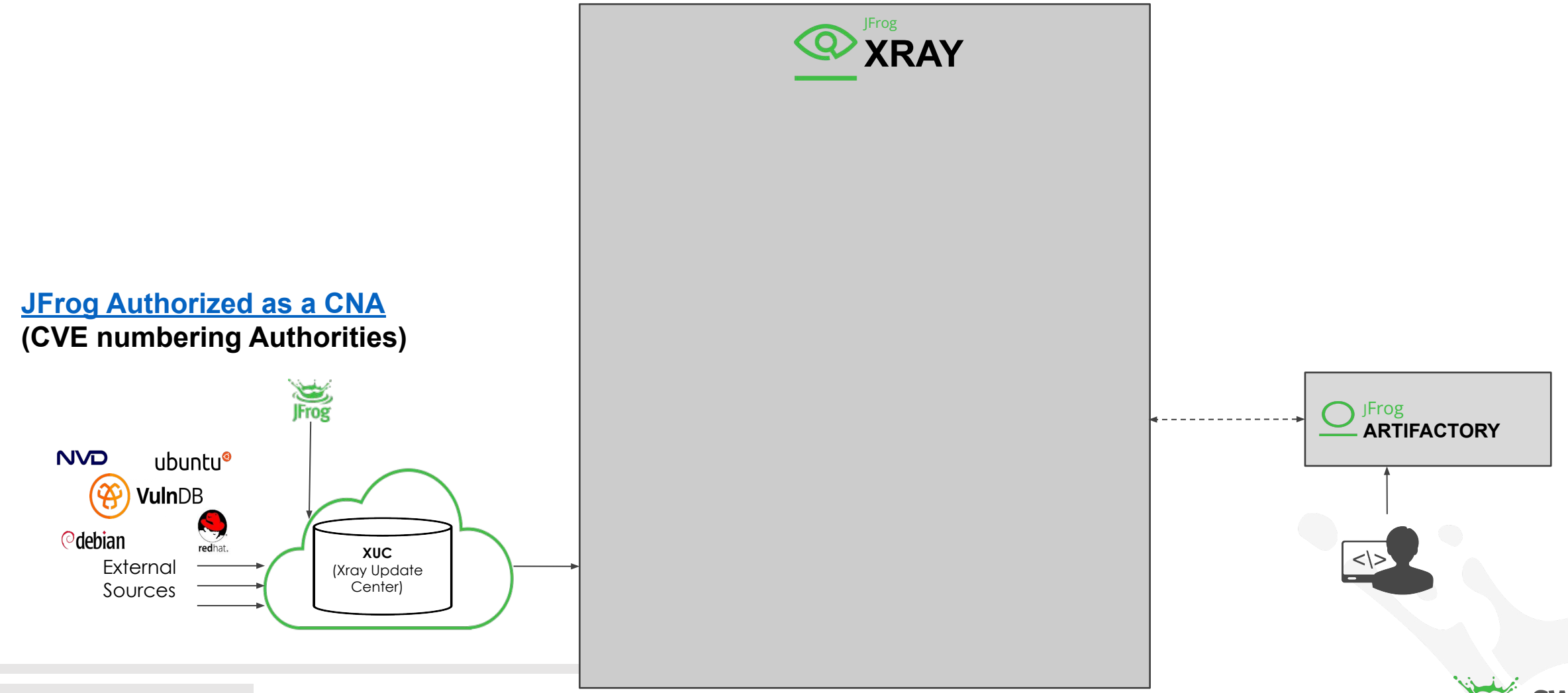
RULES — Consists of Security, Licensing and Operation Risks criteria with corresponding Automatic actions

WATCHES — Define the scope of resources and enforce the policies assigned to them

JFROG PLATFORM RESOURCES

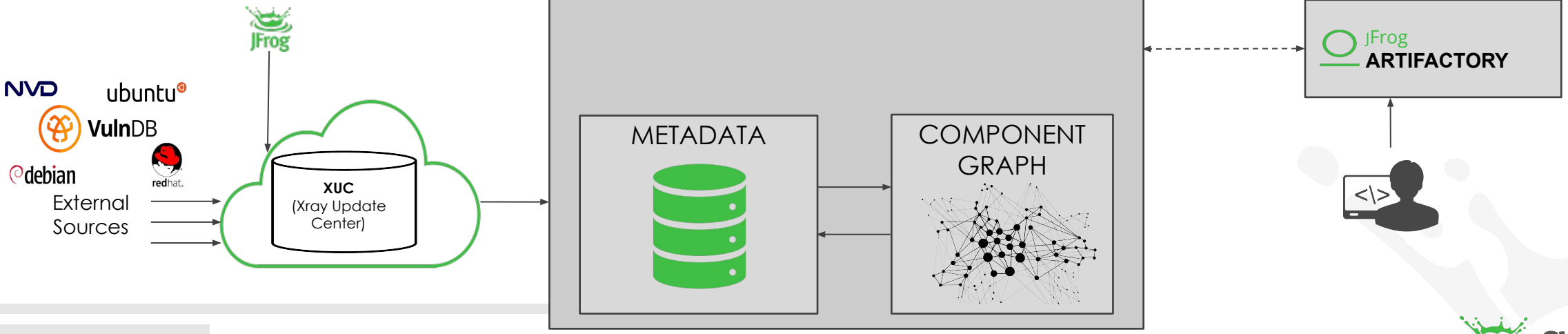


XRAY OVERVIEW



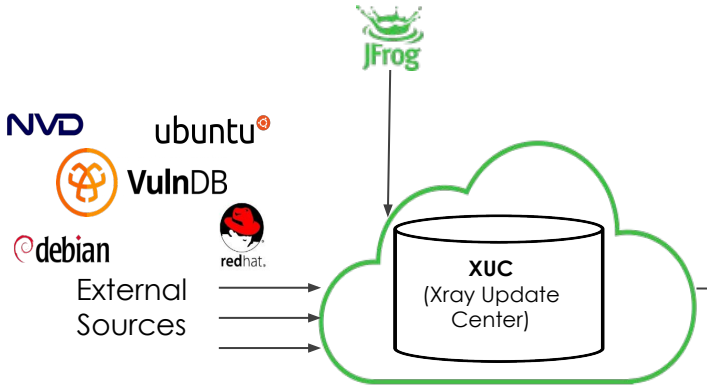
XRAY OVERVIEW

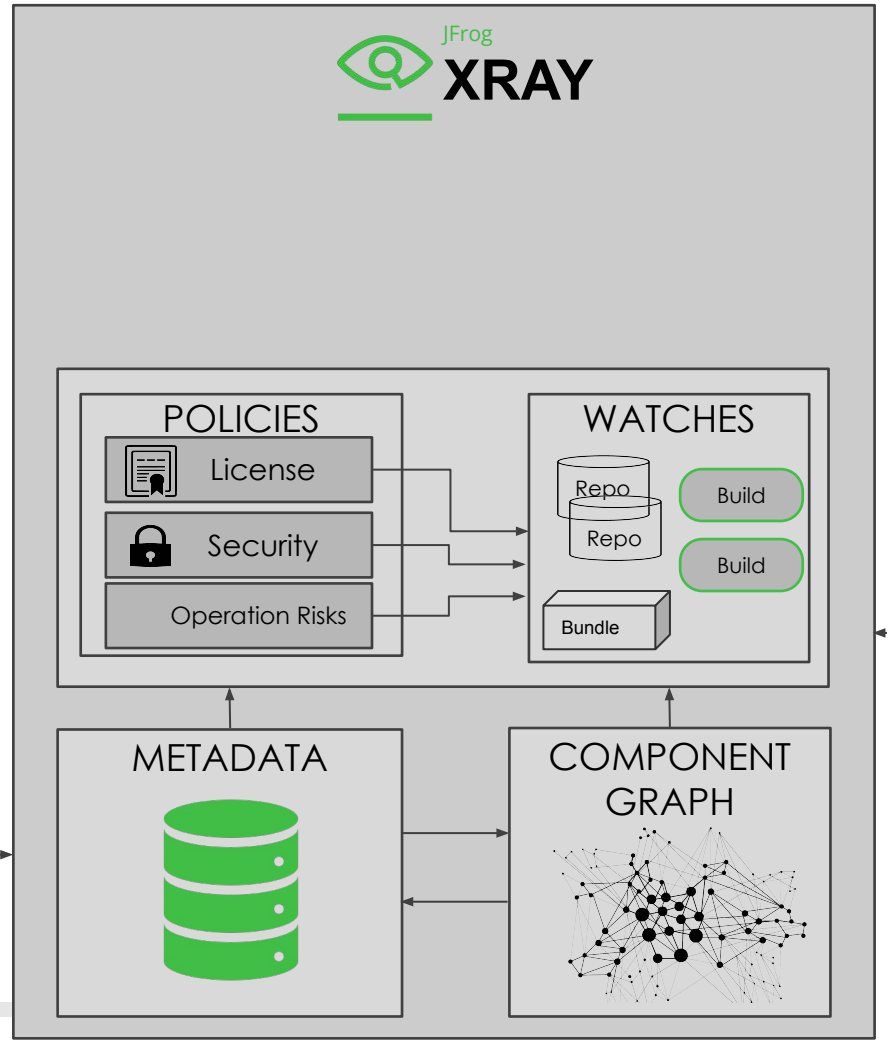
JFrog Authorized as a CNA
(CVE numbering Authorities)

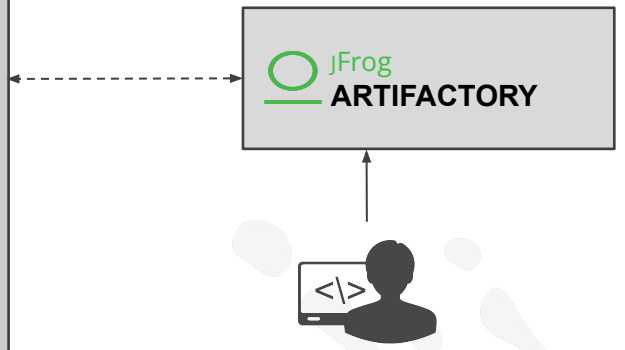


XRAY OVERVIEW

JFrog Authorized as a CNA
(CVE numbering Authorities)

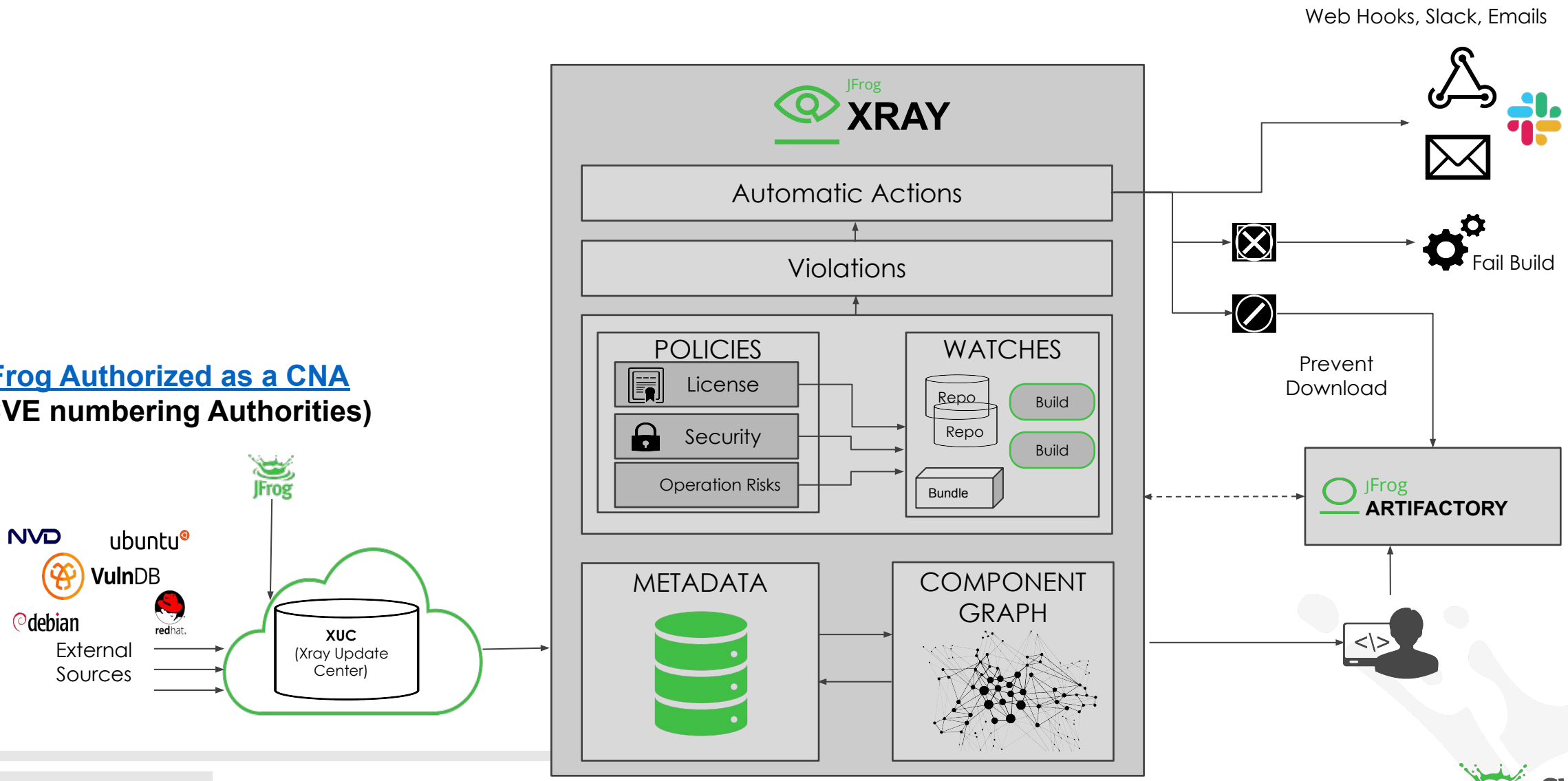




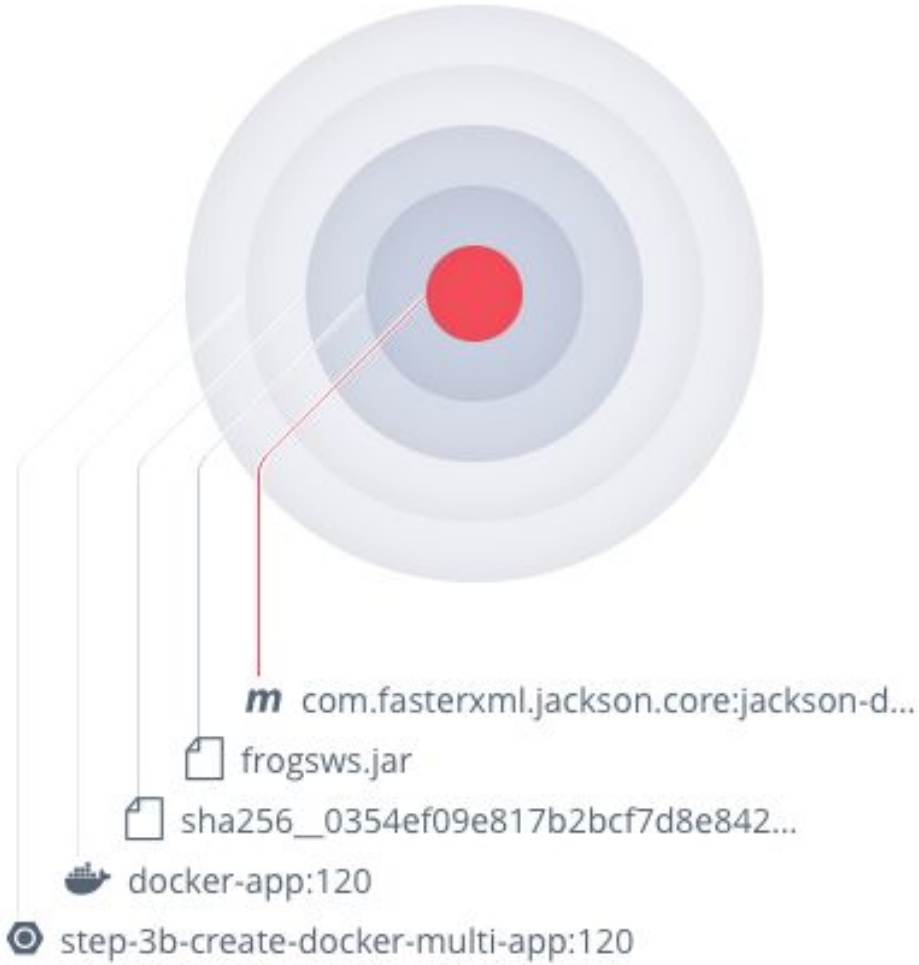


swampUP
JFrog 2022 CITY TOUR!

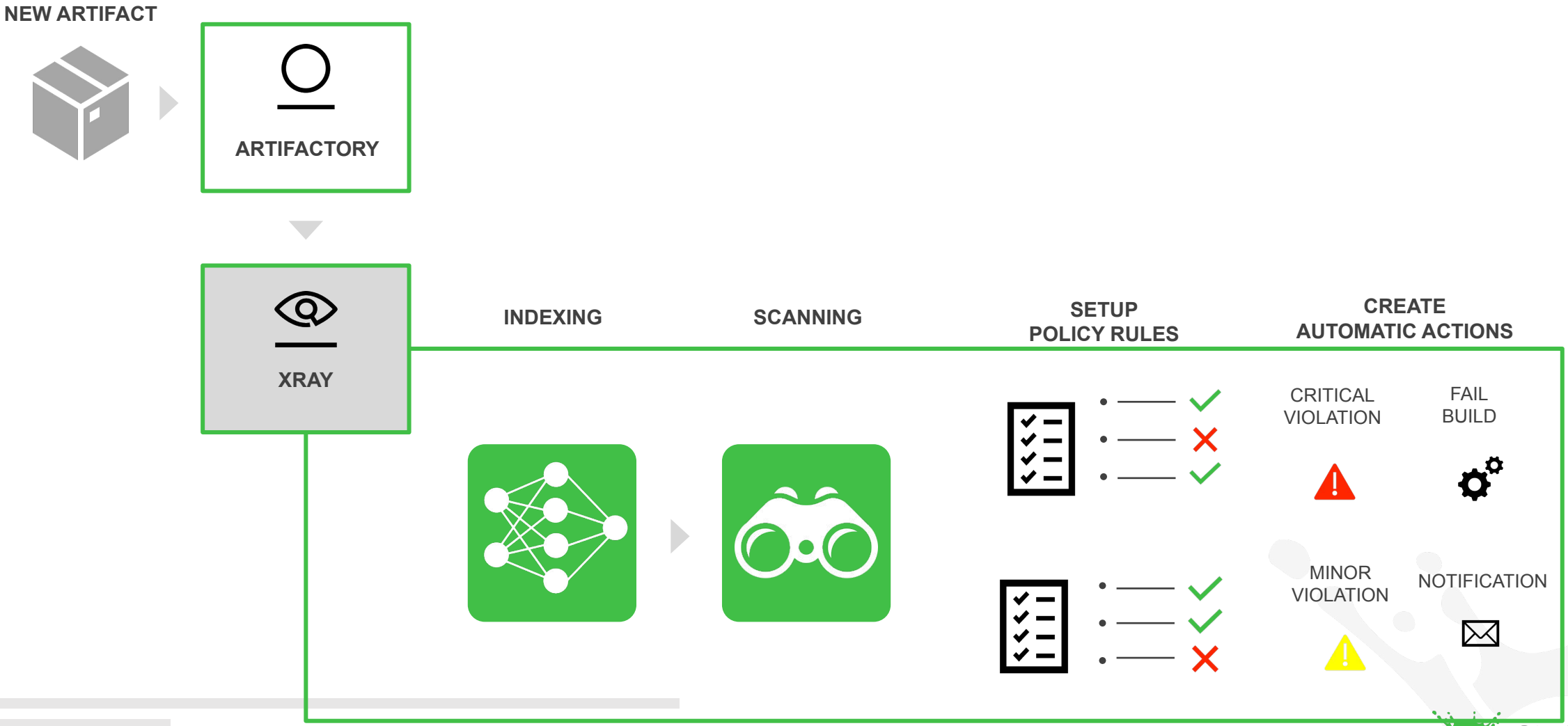
XRAY OVERVIEW



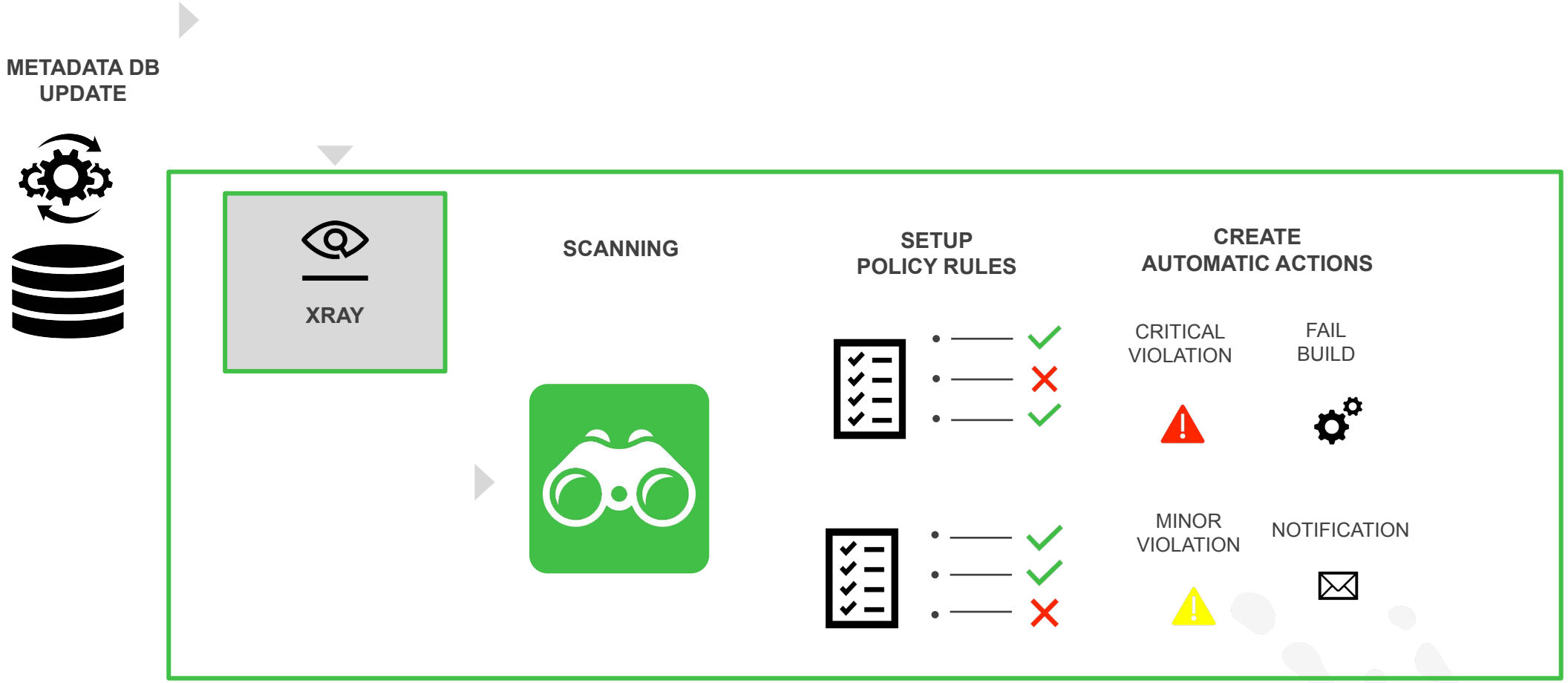
DEEP RECURSIVE SCAN



CONTINUOUS ANALYSIS ON NEW ARTIFACTS

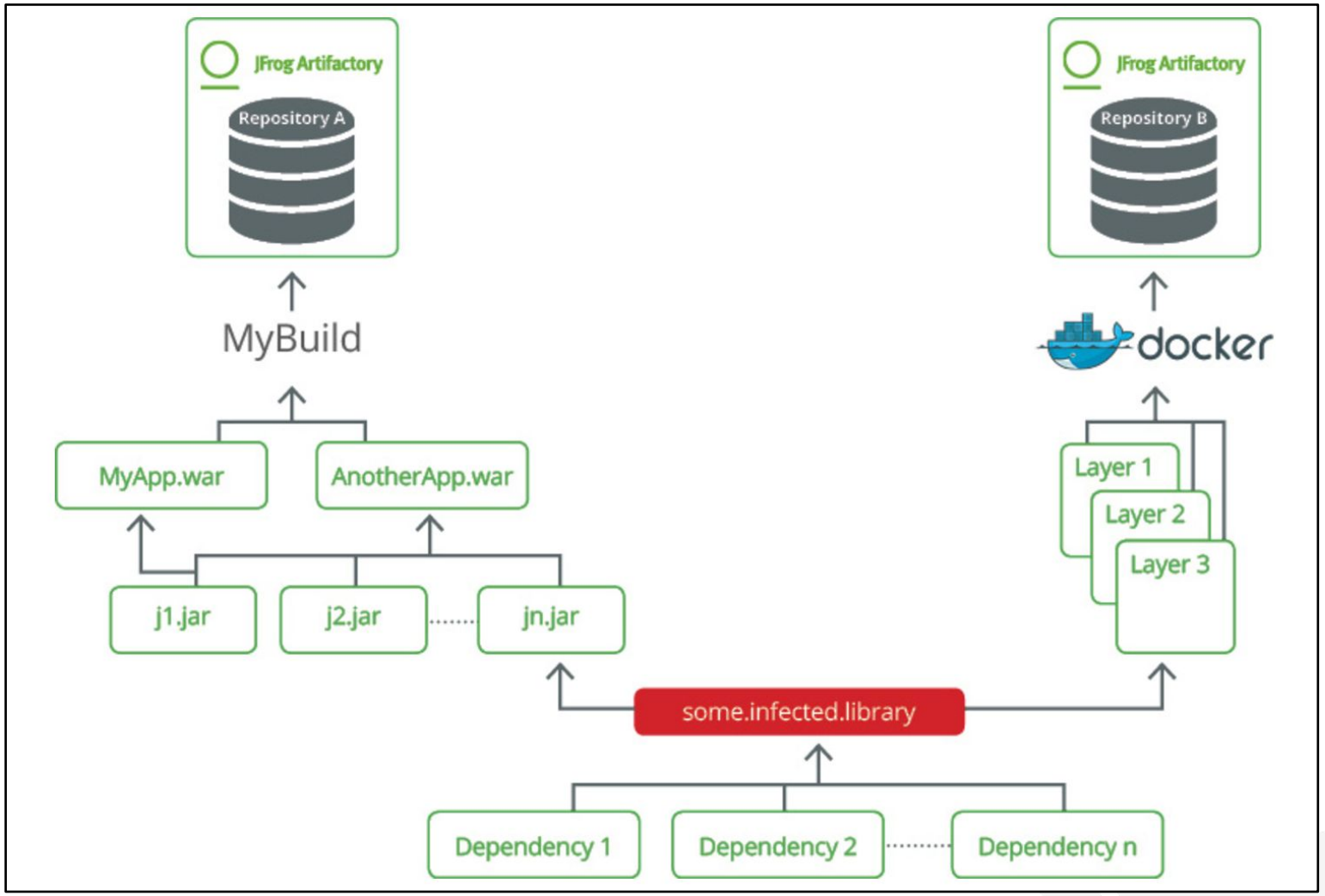


CONTINUOUS ANALYSIS ON METADATA UPDATE



IMPACT ANALYSIS

- Relationships between all the components
- Once a component has a vulnerability, Xray understands the impact it has on other components
- Xray displays a graph showing the relationship between the vulnerable component and all others that are affected



DEMO

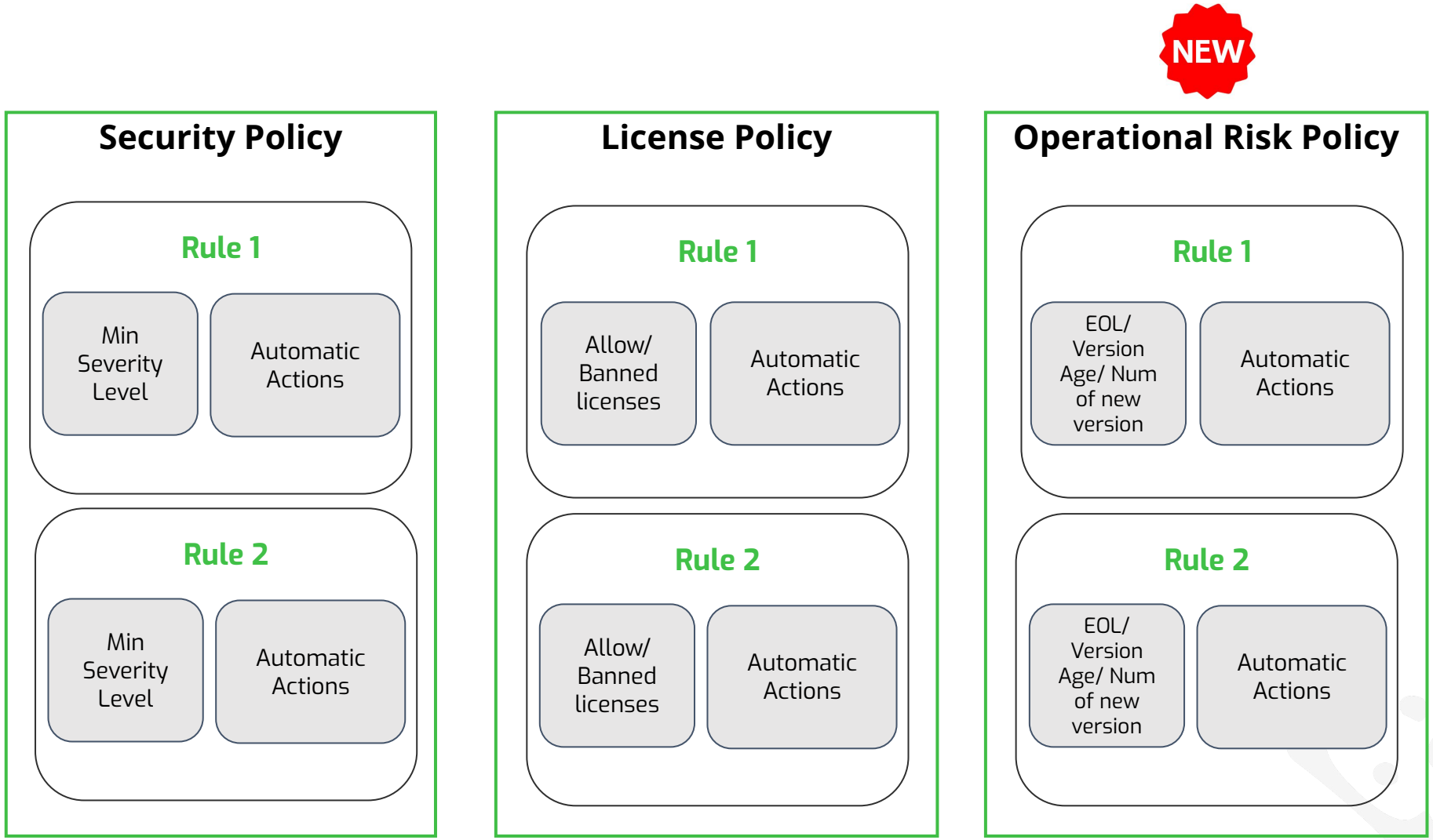
POLICY

- A policy is :
  - contextless, which means that it only defines what to enforce and not what to enforce it on
  - composed of at least 1 rule
- Each rule is defined by :
  - a license/security/operational risk criteria
  - a set of automatic actions according to your needs
    - Generate Violations, Notify by Email, Notify Deployer, Notify Watch Recipients, Create JIRA ticket, Trigger Webhook, Block Download, Fail Build

The Rule priority list sets the order in which the rules are processed.

Once a rule is met, the subsequent rules in the list will not be applied.

XRAY POLICIES



COMPONENTS OPERATIONAL RISK

- Makes you aware of the risks (EOL/Version Age/Number of New Versions/Health of Open Source Project) of using outdated or inactive OSS components in your project.
- Tracks the following:
  - EOL: OSS components in use is obsolete or declared EOL by author.
  - Version Age: Age of OSS component's age in use in months.
  - Number of new versions: Number of new versions released after current version
  - Health of OSS projects: boolean value, depending on the following:
    - Release cadence/year
    - Number of commits/year
    - Number of commiters/year

ORGANIZE POLICIES

- Configure at
  - Global level
  - Project level
- Generally managed by the security team / project admin
  - requires the “Manage Policies” roles set at the user or group level
  - provides create / update / delete actions on ALL policies !!
- Global to the organization ⇔ can be referenced in any watch
- For dedicated policies per project, different solutions :
  - grant the “Manage Policies” to the watch owners
  - automate the policies creation process using Xray Rest API

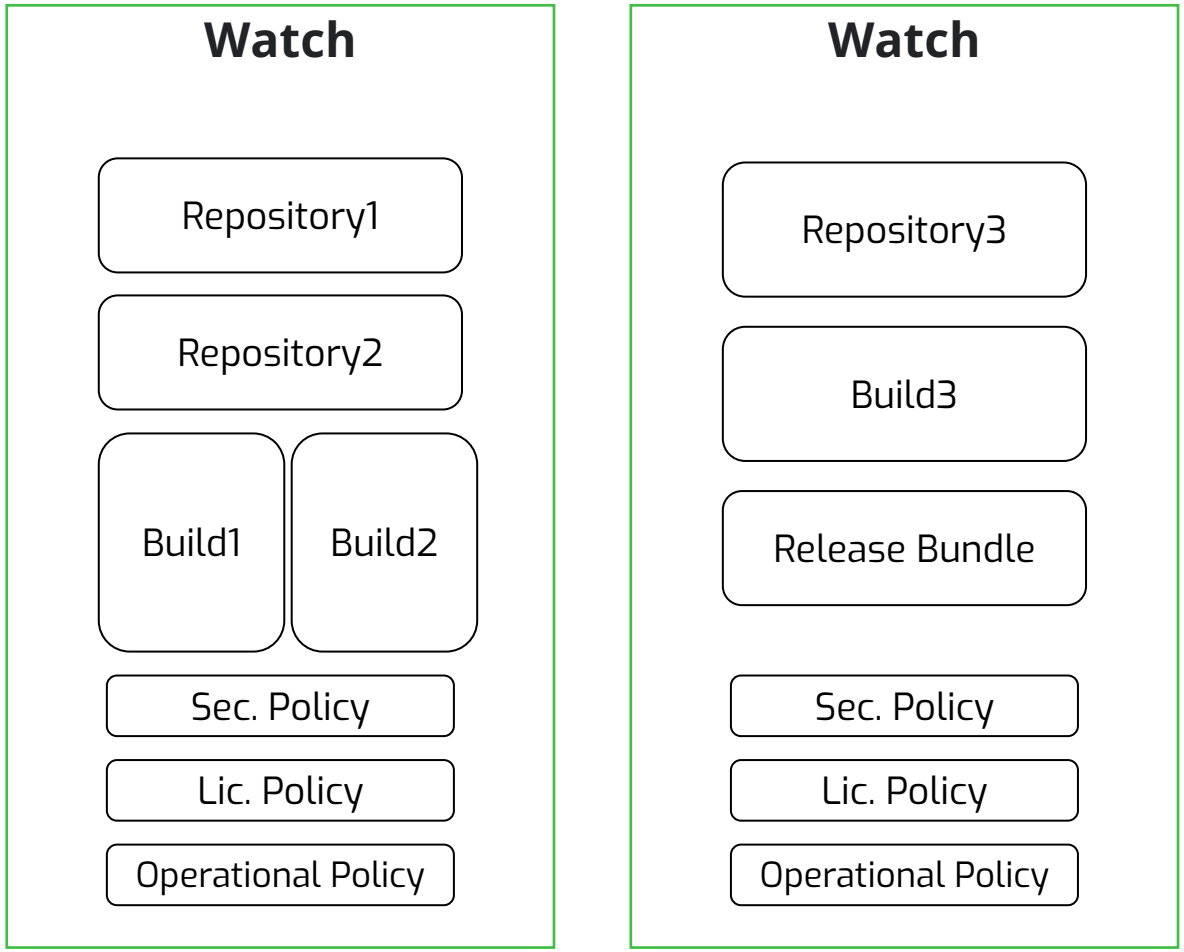
LAB 1

[SwampUp2022/SUP003-Intro_to_DevSecOps_with_JFrog_Xray/lab-1]

WATCHES

- Define the **scope of the resources** you want to watch - repositories, builds, release bundles.
- Enforce the policies assigned to them on the artifacts by **generating violations**.
- **Filters** in a watch determine on which binaries the policies will be applied.
- Even if a rule in one policy is applied, the other policies will be applied as well. Only inside a policy we stop at the first rule.

XRAY WATCHES



ORGANIZE WATCHES

- Configure at
  - Global level
  - Project level
- Managed by a group of “trusted” people such as Release managers, Project managers, Technical leads, …
  - requires the “Manage Watches” role set at the user or group level
  - provides create / update / delete actions on ALL watches !!

VIOLATION CONCEPT

- A violation is raised when a criteria, set in a license or security or operational risk policy rule, matches the license of a component and respectively the vulnerability level in the scanned artifact.
- A violation can be ignored temporary or permanently
  - the scope is tied to :
    - ❏ watch, build, component, vulnerability

DEMO

LAB 2

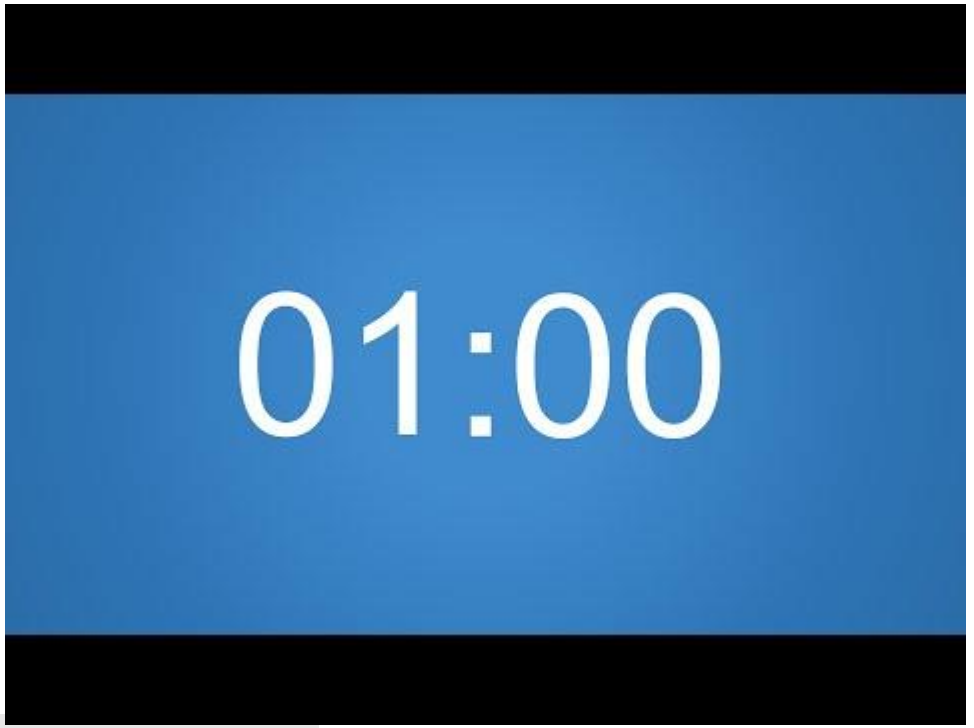
[SwampUp2022/SUP003-Intro_to_DevSecOps_with_JFrog_Xray/lab-2]

COFFEE BREAK

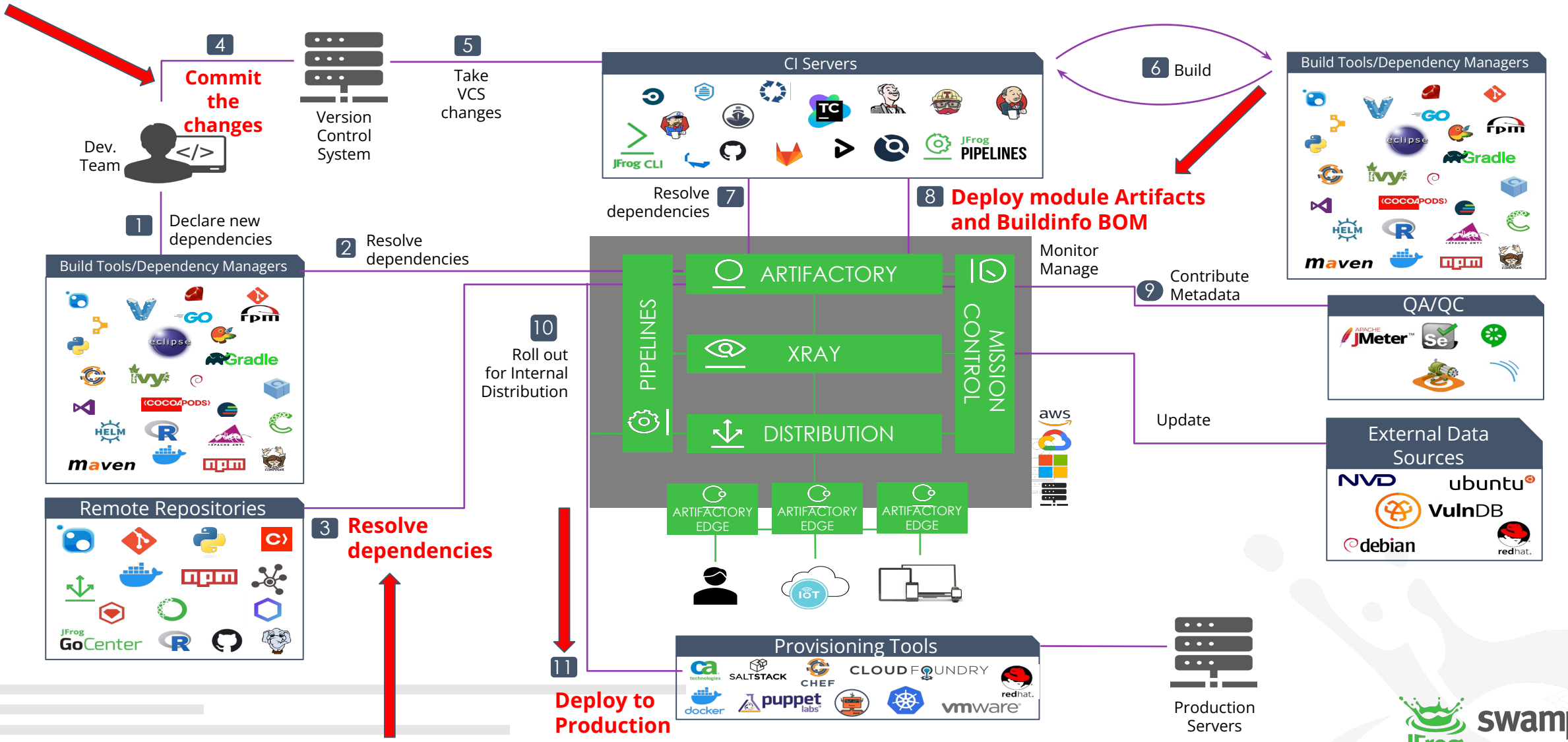
AGENDA

01:05–02:45	DevOps & DevSecOps Overview
	Introduction To JFrog Xray
	Policies and Watches
	Impact Analysis
02:45-3:00	Break
03:00-04:50	Shift Left Strategy
	CI / CD Integration
	Scan Reports
	New Features
04:45–05:05	Summary, Quiz, Survey & Q/A

POLL



WHERE DOES XRAY FIT?



INTERACT WITH XRAY

- **Shift Left Strategy**
  - IDE Integration
  - Dependency Scanning
  - On Demand Binary Scan
  - Docker Desktop Extension
  - Frogbot
- JFrog CLI
- CI Plugins
- REST API
- GUI

swampUP 2022 CITY TOUR!

IDE
INTEGRATIONS

JFrog
The Liquid Software Company

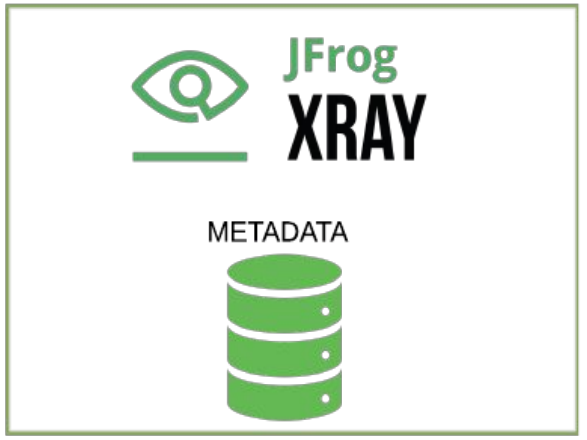
SHIFT LEFT STRATEGY AT IDE

- Cost of **remediating a vulnerability** is akin to the cost of fixing a bug
- Earlier you remediate a vulnerability in the release cycle, the **lower the cost**
- Bringing Xray's issue discovery one step earlier, to **development time**
- Flags components when vulnerabilities are discovered
- Developers will access the **component tree**, **vulnerability data** and **license details**

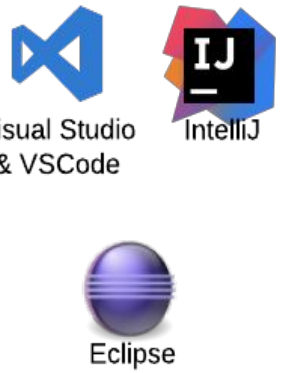
SECURING AT DEV (SHIFT LEFT STRATEGY)

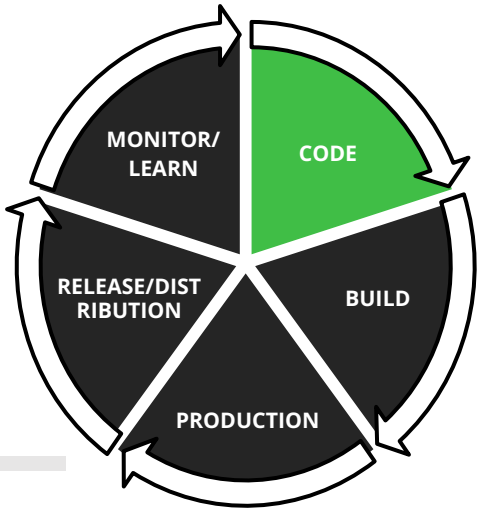
Xray Cloud DB

Daily updates
Sync



Update
Security &
License data





SUPPORTED TECHNOLOGIES BY IDE PLATFORMS

Project Dependencies	Visual Studio Code	IntelliJ	WebStorm	PyCharm	GoLand	Android Studio	Visual Studio	Eclipse	Theia
Maven	✓	✓						✓	✓
Gradle		✓				✓		✓	
Python	✓	✓		✓					✓
Go	✓	✓			✓				✓
NPM	✓	✓	✓					✓	✓
NuGet	✓						✓		✓

DEPENDENCY SCANNING

- Scan your source dependencies for security or license violations using JFrog CLI
- Scan can be done **before a developer checks-in the code**.
- Run on source directory, ANYWHERE/ANYTIME, without the need to compile, test or deploy to Artifactory.
- **Supported** for Maven, Gradle, NPM, Pip, Pipenv, GO, NuGet, .NET Core packages, yarn2
- JFrog Cli 2.40 and above for Dependency Scanning.

- **jf audit**

  or

- **jf aud**

NOTE:

--watches - A comma separated list of Xray watches, to enable Xray to determine violations accordingly.

--licenses - List of licenses to be displayed if it's set to true.

--format - Defines the output format. Table or JSON

--repo-path - Artifactory repository path to enable Xray to determine violations accordingly.

Command Options

DEPENDENCY SCANNING

Security Violations

SEVERITY	IMPACTED PACKAGE	IMPACTED PACKAGE VERSION	TYPE	FIXED VERSIONS	COMPONENT	COMPONENT VERSION	CVE	CVSS V2	CVSS V3	ISSUE ID
Critical	commons-io:commons-io	2.6	Maven	[2.7]	commons-io:commons-io	2.6		5.0		XRAY-78200
High	commons-io:commons-io	2.6	Maven	[2.8.0]	commons-io:commons-io	2.6		7.1	7.5	XRAY-125253
Medium	org.apache.httpcomponents:httpclient	4.5.6	Maven	[4.5.13]	org.apache.httpcomponents:httpclient	4.5.6	CVE-2020-13956	5.0	5.3	XRAY-129495
Medium	commons-io:commons-io	2.6	Maven	[2.7]	commons-io:commons-io	2.6	CVE-2021-29425	5.0	5.3	XRAY-172728
Medium	commons-codec:commons-codec	1.10	Maven	[1.13]	commons-codec:commons-codec	1.10		5.0		XRAY-87486
Medium	org.postgresql:postgresql	42.2.0	Maven	[42.2.13]	org.postgresql:postgresql	42.2.0	CVE-2020-13692	6.8	7.7	XRAY-146611
Medium	org.postgresql:postgresql	42.2.0	Maven	[42.2.5]	org.postgresql:postgresql	42.2.0	CVE-2018-10936	6.8	8.1	XRAY-77885
Medium	log4j:log4j	1.2.17	Maven		log4j:log4j	1.2.17	CVE-2020-9488	4.3	3.7	XRAY-96751
Low	log4j:log4j	1.2.17	Maven		log4j:log4j	1.2.17		6.9		XRAY-87321

License Compliance Violations

LICENSE	SEVERITY	IMPACTED PACKAGE	IMPACTED PACKAGE VERSION	TYPE	COMPONENT	COMPONENT VERSION
BSD 2-Clause	High	com.ongres.scram:common	1.0.0-beta.2	Maven	com.ongres.scram:common	1.0.0-beta.2
BSD 2-Clause	High	org.postgresql:postgresql	42.2.0	Maven	org.postgresql:postgresql	42.2.0

ON DEMAND BINARY SCAN

- Build software securely during development, without trying to find and fix vulnerabilities after your code is compiled.
- Use JFrog CLI:
  - run ad-hoc scans for security purposes **without uploading to Artifactory** first.
  - point to a binary in your local file system and receive a report that contains a list of vulnerabilities and licenses for that binary.
- Get the results in Tabular or JSON format.

- **jf scan "{{file_path}}"** / **jf docker scan {{image_name:tag}}**

  or

- **jf s "{{file_path}}"**

Command Options

ON DEMAND BINARY SCAN

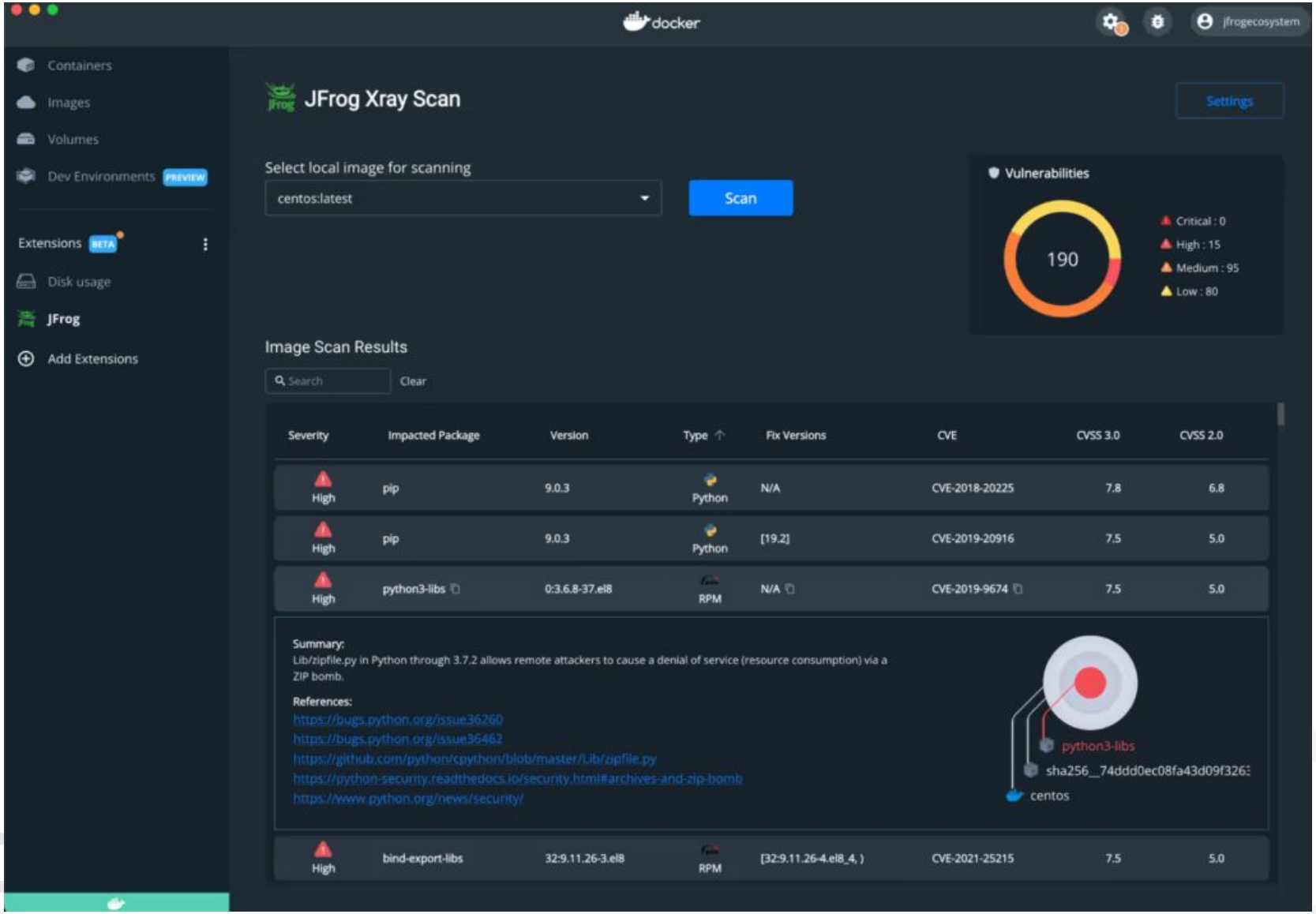
Security Violations

SEVERITY	IMPACTED PACKAGE	IMPACTED PACKAGE VERSION	TYPE	FIXED VERSIONS	COMPONENT	COMPONENT VERSION	CVE	CVSS V2	CVSS V3	ISSUE ID
Critical	commons-io:commons-io	2.6	Maven	[2.7]	commons-io:commons-io	2.6		5.0		XRAY-78200
High	commons-io:commons-io	2.6	Maven	[2.8.0]	commons-io:commons-io	2.6		7.1	7.5	XRAY-125253
Medium	org.apache.httpcomponents :httpclient	4.5.6	Maven	[4.5.13]	org.apache.httpcomponents :httpclient	4.5.6	CVE-2020-13956	5.0	5.3	XRAY-129495
Medium	commons-io:commons-io	2.6	Maven	[2.7]	commons-io:commons-io	2.6	CVE-2021-29425	5.0	5.3	XRAY-172728
Medium	commons-codec:commons-cod ec	1.10	Maven	[1.13]	commons-codec:commons-cod ec	1.10		5.0		XRAY-87486
Medium	org.postgresql:postgresql	42.2.0	Maven	[42.2.13]	org.postgresql:postgresql	42.2.0	CVE-2020-13692	6.8	7.7	XRAY-146611
Medium	org.postgresql:postgresql	42.2.0	Maven	[42.2.5]	org.postgresql:postgresql	42.2.0	CVE-2018-10936	6.8	8.1	XRAY-77885
Medium	log4j:log4j	1.2.17	Maven		log4j:log4j	1.2.17	CVE-2020-9488	4.3	3.7	XRAY-96751
Low	log4j:log4j	1.2.17	Maven		log4j:log4j	1.2.17		6.9		XRAY-87321

License Compliance Violations

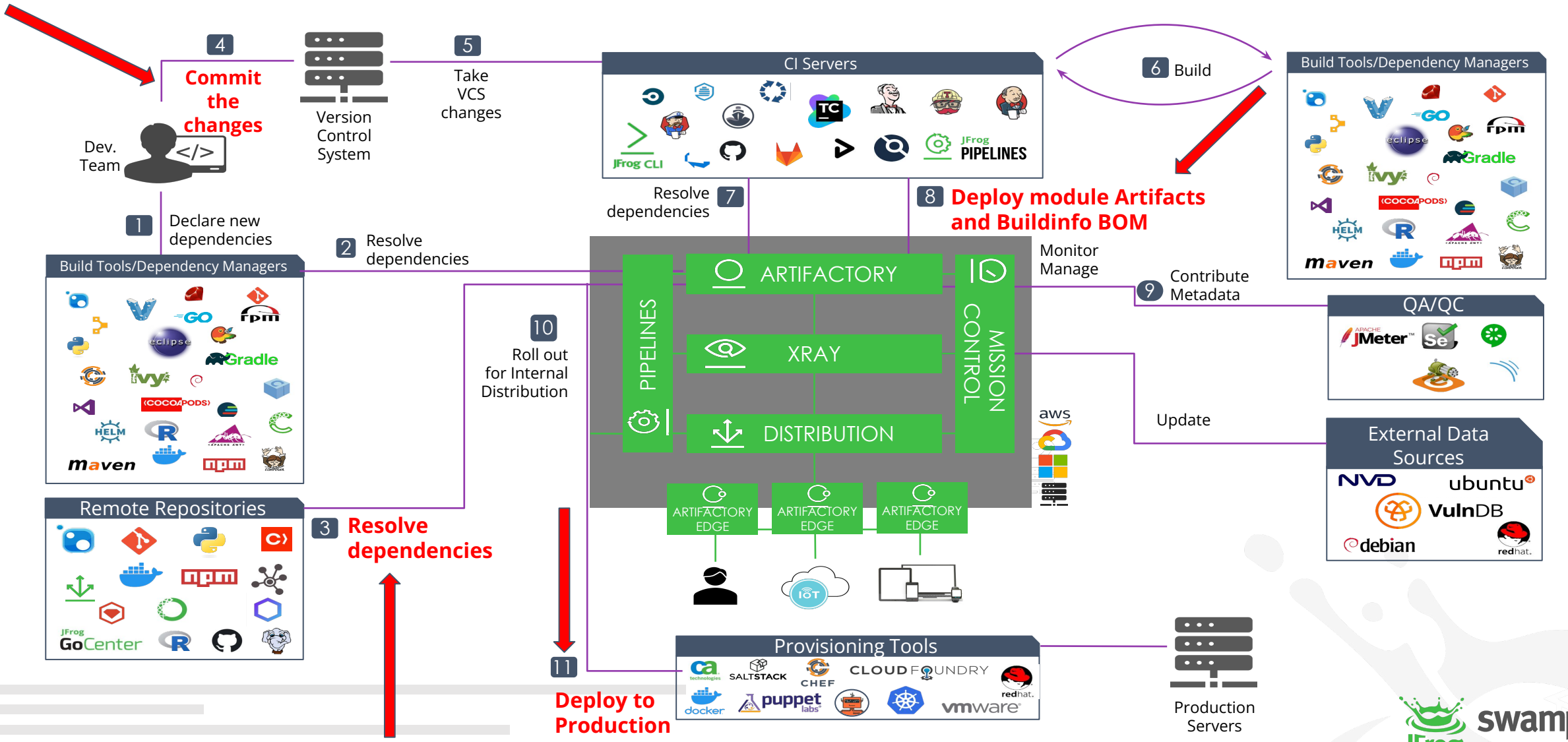
LICENSE	SEVERITY	IMPACTED PACKAGE	IMPACTED PACKAGE VERSION	TYPE	COMPONENT	COMPONENT VERSION
BSD 2-Clause	High	com.ongres.scram:common	1.0.0-beta.2	Maven	com.ongres.scram:common	1.0.0-beta.2
BSD 2-Clause	High	org.postgresql:postgresql	42.2.0	Maven	org.postgresql:postgresql	42.2.0

DOCKER DESKTOP JFROG XRAY EXTENSION



DEMO

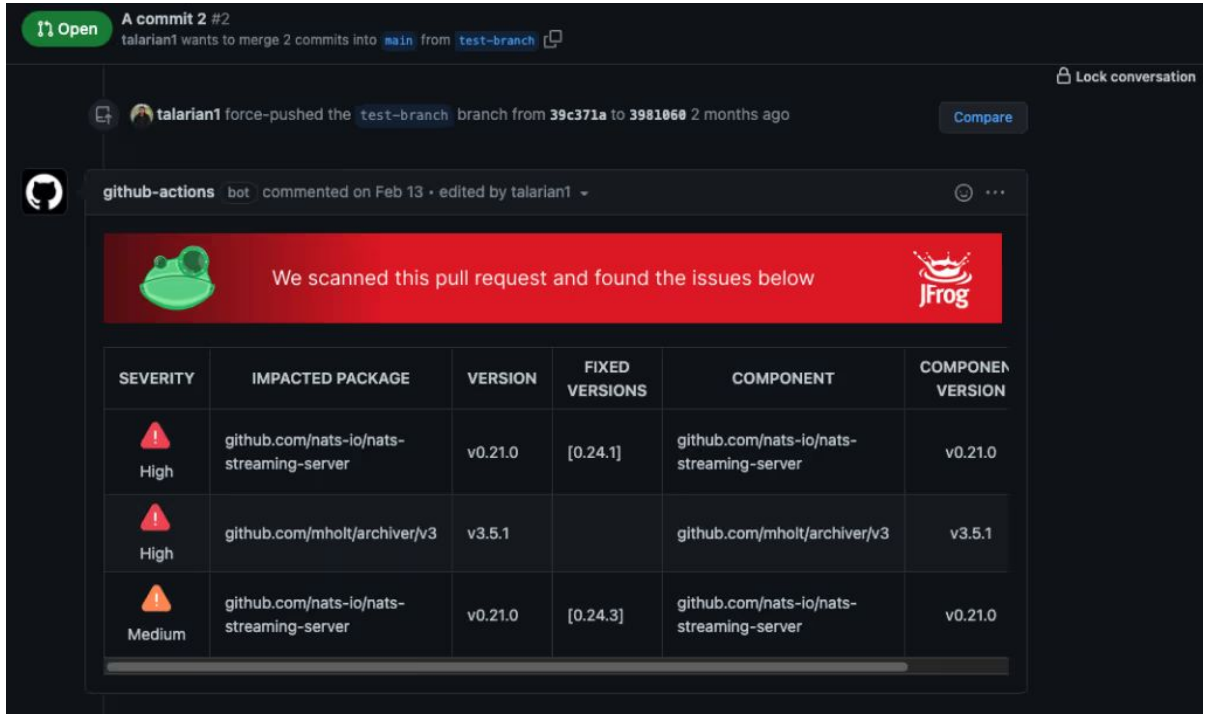
WHERE DOES XRAY FIT?

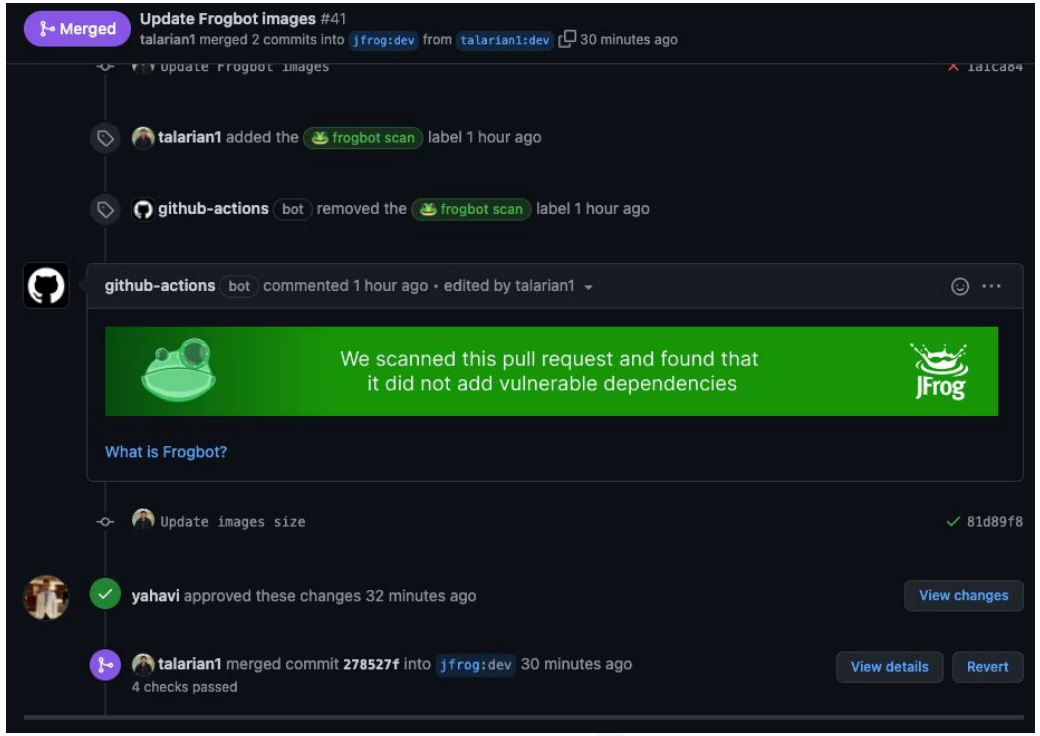


FROGBOT

- FrogBot is available for FREE
- Protect your git projects as they are being developed.
- **Scans every Pull Request** created, for security vulnerabilities with JFrog Xray.
  - This is to ensure that new PR's don't add new security vulnerabilities to your code base.
- **Automatically** opens pull requests for upgrading vulnerable dependencies to a version with a fix.
- What's **supported**
  - Github, Gitlab and Bitbucket are supported.
  - NPM, Maven, Gradle, Go, Pip, Pipenv, NuGet, .Net, Yarn2

FROGBOT





DEMO

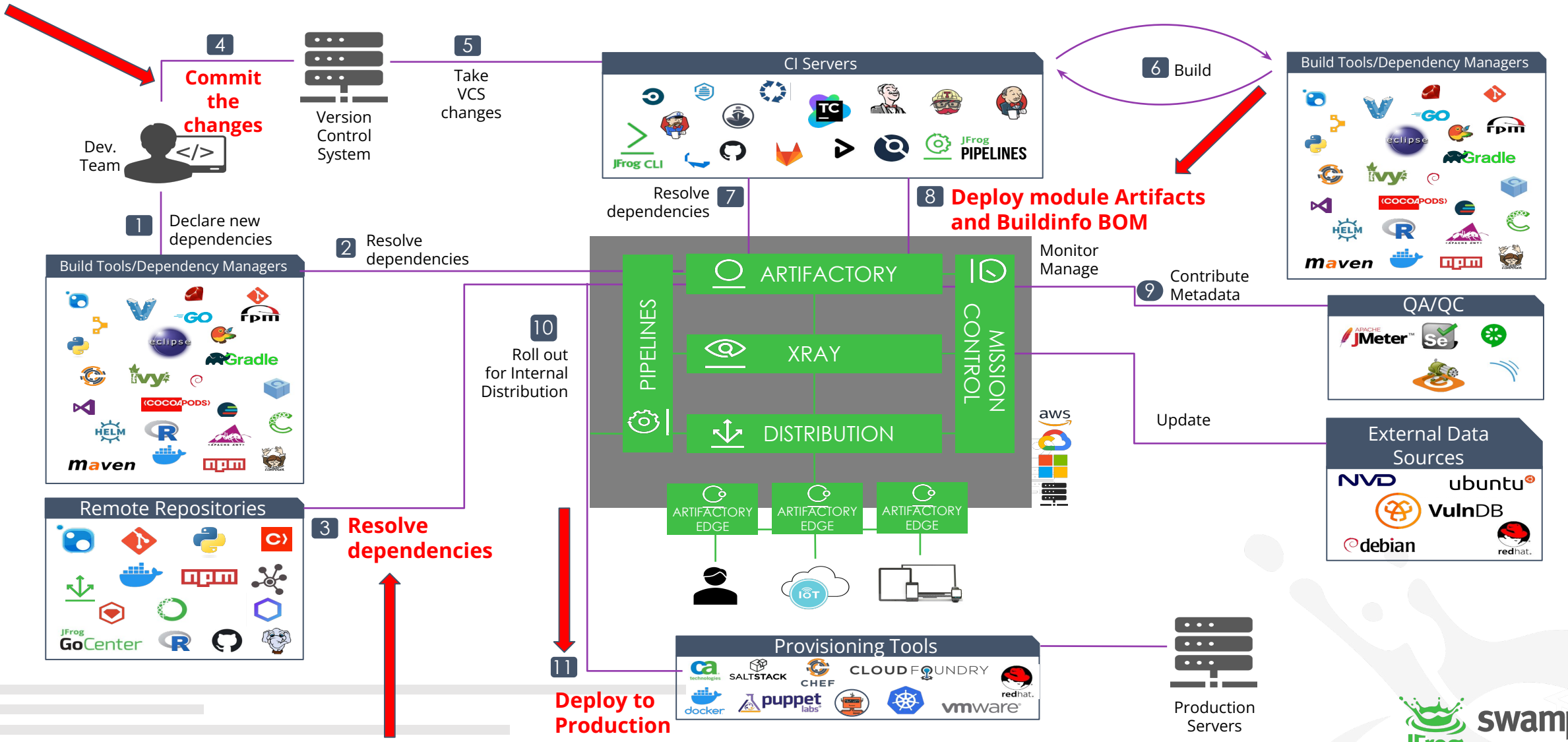
LAB 3

[SwampUp2022/SUP003-Intro_to_DevSecOps_with_JFrog_Xray/lab-3]

CI/CD
INTEGRATION

The Liquid Software Company

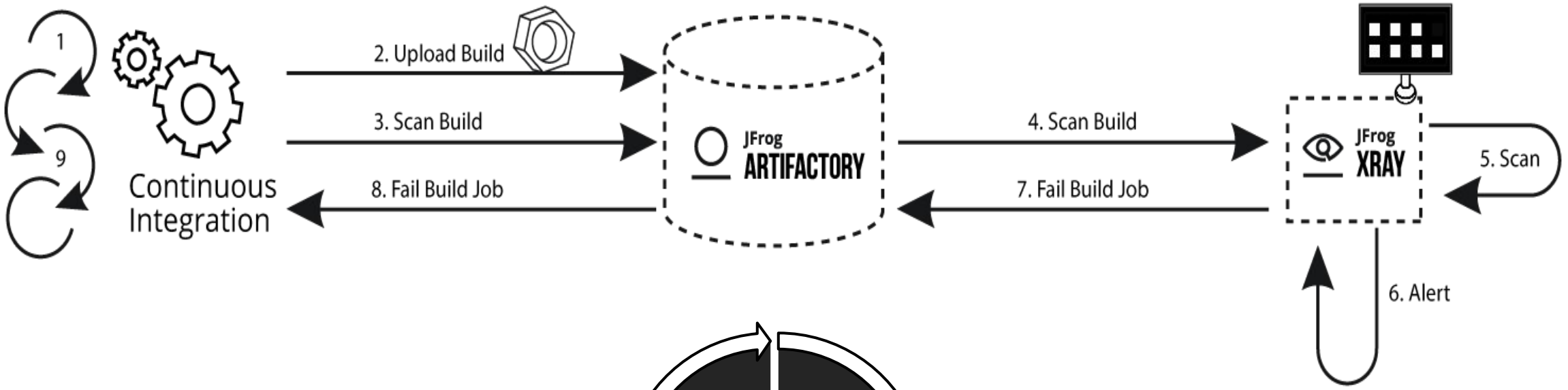
WHERE DOES XRAY FIT?

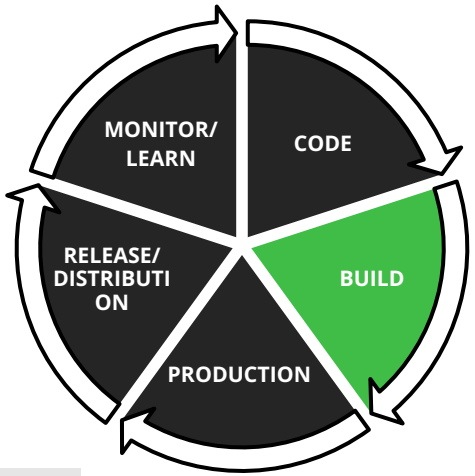


BUILD INFO

- System record for fully reproducible builds
- Dependency traceability per build
- Build info promotion
- System & Environment Variables
- Modules produced by Builds including Artifacts and Dependencies
- Build Project details - Produced by & Used By
- Difference between the Builds

SECURING AT BUILD TIME-CI



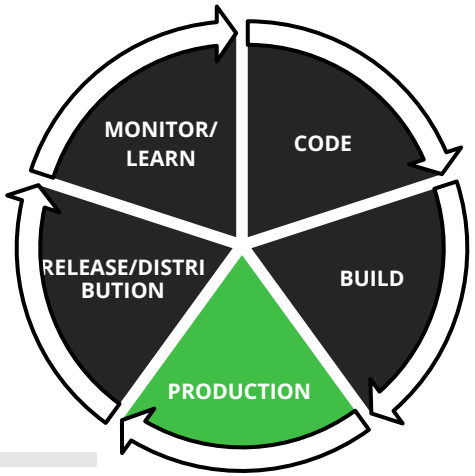


JFROG BUILD OFFERING BY PACKAGE TYPE

CI Servers								
Package Type	**JFrog Pipelines**	**JFrog CLI**	**Jenkins**	**Bamboo**	**TeamCity**	**Azure DevOps**	**GitHub Actions**	**Bitbucket Pipelines**
Generic	✓	✓	✓	✓	✓	✓	✓	✓
Conan			✓			✓		
Docker	✓	✓	✓			✓	✓	✓
Go	✓	✓	✓			✓	✓	✓
Gradle	✓	✓	✓	✓	✓		✓	✓
Ivy			✓	✓	✓			
Maven	✓	✓	✓	✓	✓	✓	✓	✓
npm	✓	✓	✓	✓		✓	✓	✓
NuGet	✓	✓				✓	✓	✓
Python		✓					✓	✓
Terraform		✓					✓	
Podman		✓						
Kaniko		✓	✓					

DEMO

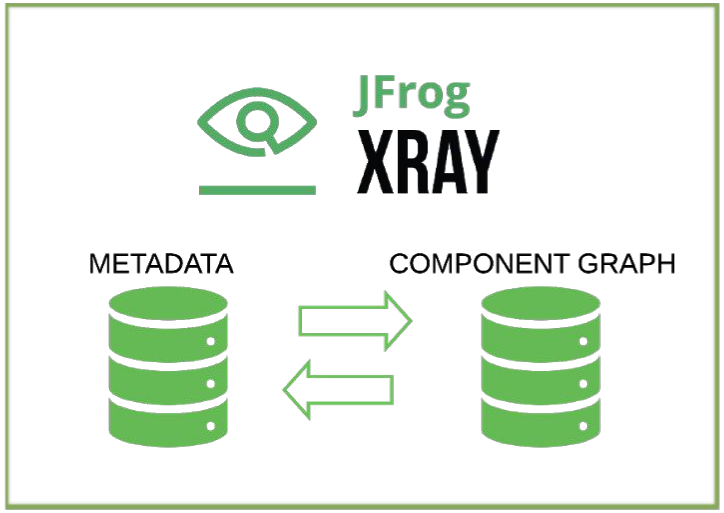
SECURING AT PRODUCTION



CONTINUOUS ANALYSIS AT PRODUCTION

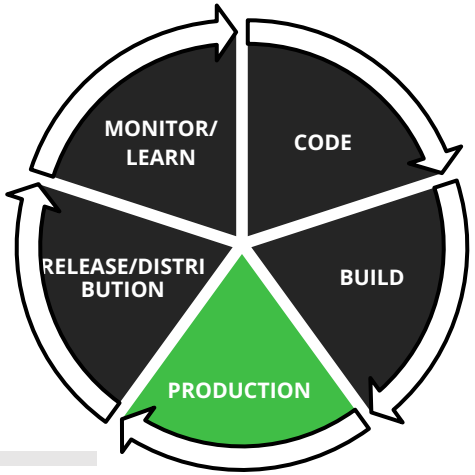
Xray Cloud DB

Daily updates
Sync



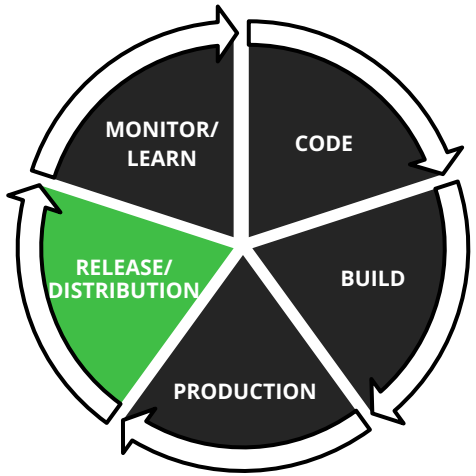
Update
Violations &
Actions

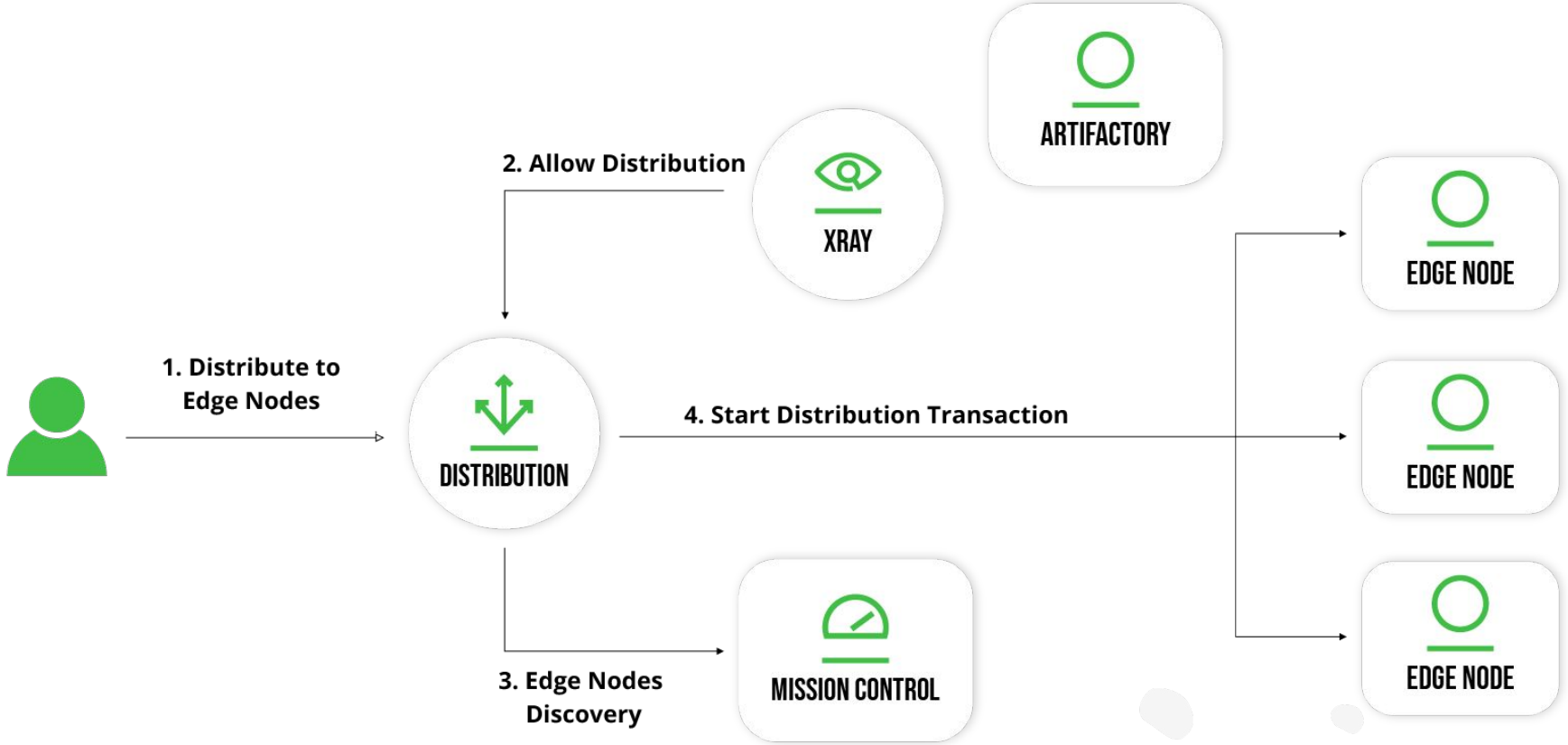
JFrog
ARTIFACTORY



swampUP
JFrog 2022 CITY TOUR!

SECURING DISTRIBUTION RELEASE BUNDLES





DEMO

REPORTS

The Liquid Software Company

REPORT TYPES

- Vulnerabilities Report
  - Vulnerabilities in Artifacts, Builds, Release Bundles, JFrog Project
- Licenses Due Diligence Report
  - List of Components and Artifacts and their relevant Licenses such as unknown licenses and unrecognized licenses found in component
- Operational Risk Reports
  - Data on OSS components that will help you gain insight into risk level of the the components in use such as EOL, version Age, and so on.
- Violations Report
  - Security and license violations for each component in the selected scope such as type of violation, impacted artifacts, and severity

SOFTWARE BILLS OF MATERIALS (SBOM) REPORT

- Has become a critical DevSecOps piece.
- SBOM is a readable inventory of software components and dependencies which make up a piece of software.
- Supported SBOM formats:
  - SPDX & CycloneDX

Sec. 4. Enhancing Software Supply Chain Security
(vii) providing a purchaser a Software Bill of Materials (SBOM) for each product directly or by publishing it on a public website;

BRIEFING ROOM

Executive Order on Improving the Nation's Cybersecurity

MAY 12, 2021 • PRESIDENTIAL ACTIONS

By the authority vested in me as President by the Constitution and the laws of the United States of America, it is hereby ordered as follows:

Section 1. Policy. The United States faces persistent and increasingly sophisticated malicious cyber campaigns that threaten the public sector, the private sector, and ultimately the American people's security and privacy. The Federal Government must improve its efforts to identify, deter, protect against, detect, and respond to these actions and actors. The Federal Government must also carefully examine what occurred during any major cyber incident and apply lessons learned. But cybersecurity requires more

LAB 4

[SwampUp2022/SUP003-Intro_to_DevSecOps_with_JFrog_Xray/lab-4]

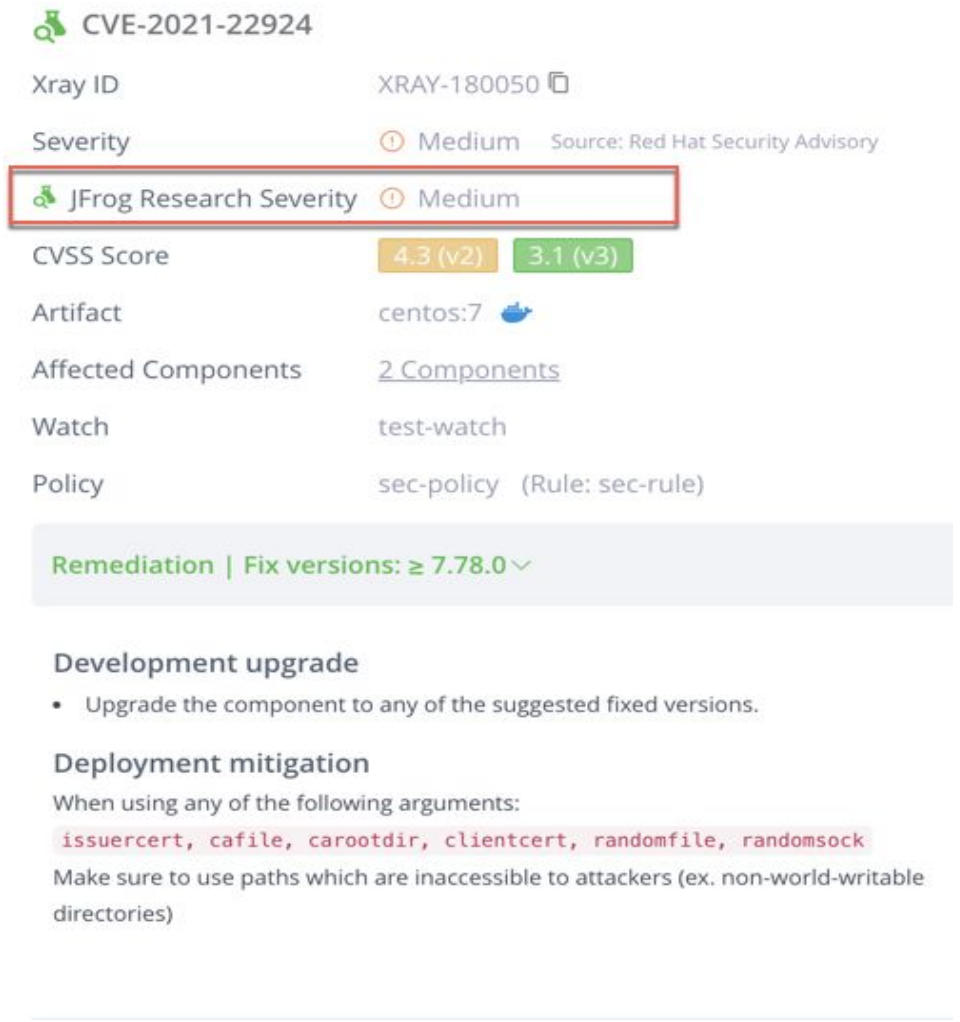
RECENTLY INTRODUCED FEATURES

The Liquid Software Company

NEW FEATURES SUMMARY

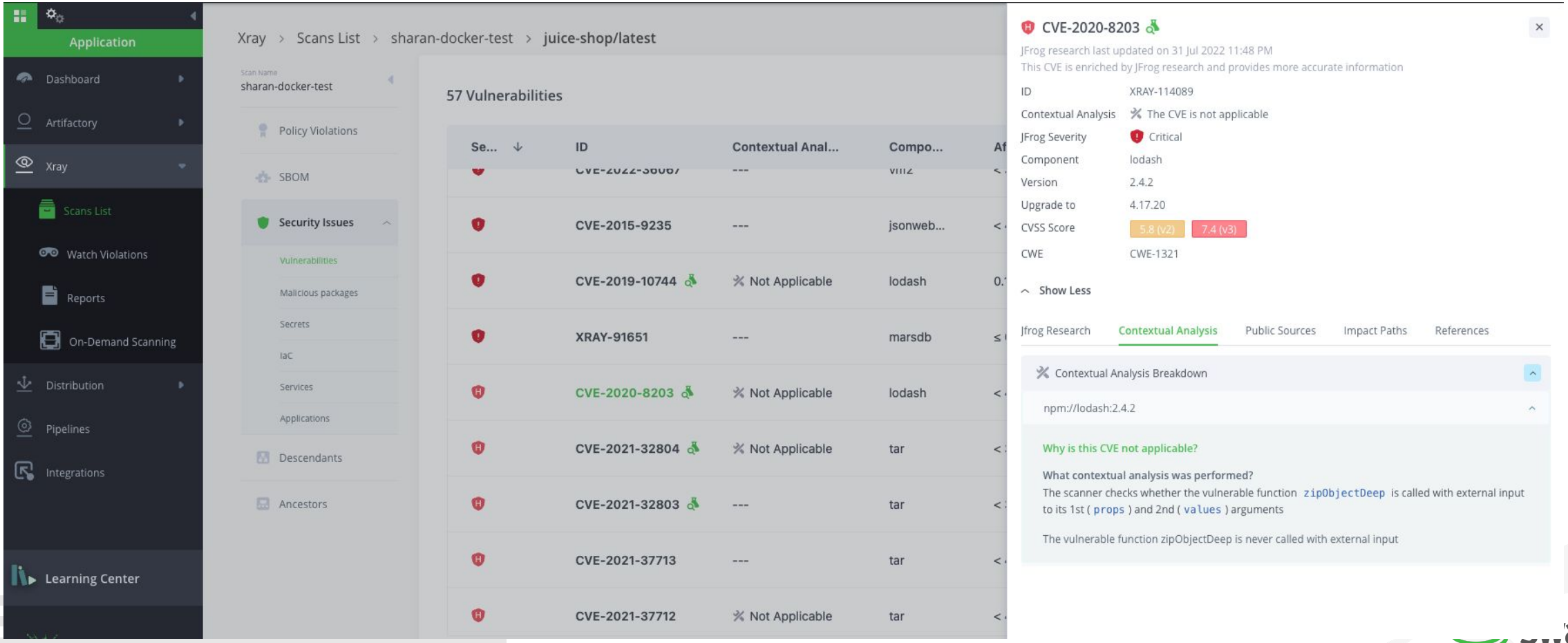
- Security CVE Research and Enrichment of data
- Contextual Analysis / Applicability Scanner
- Secrets Detection
- Application Exposures
- Service Exposures
- IaC Security Analysis

RESEARCH AND ENRICHMENT OF CVE DATA



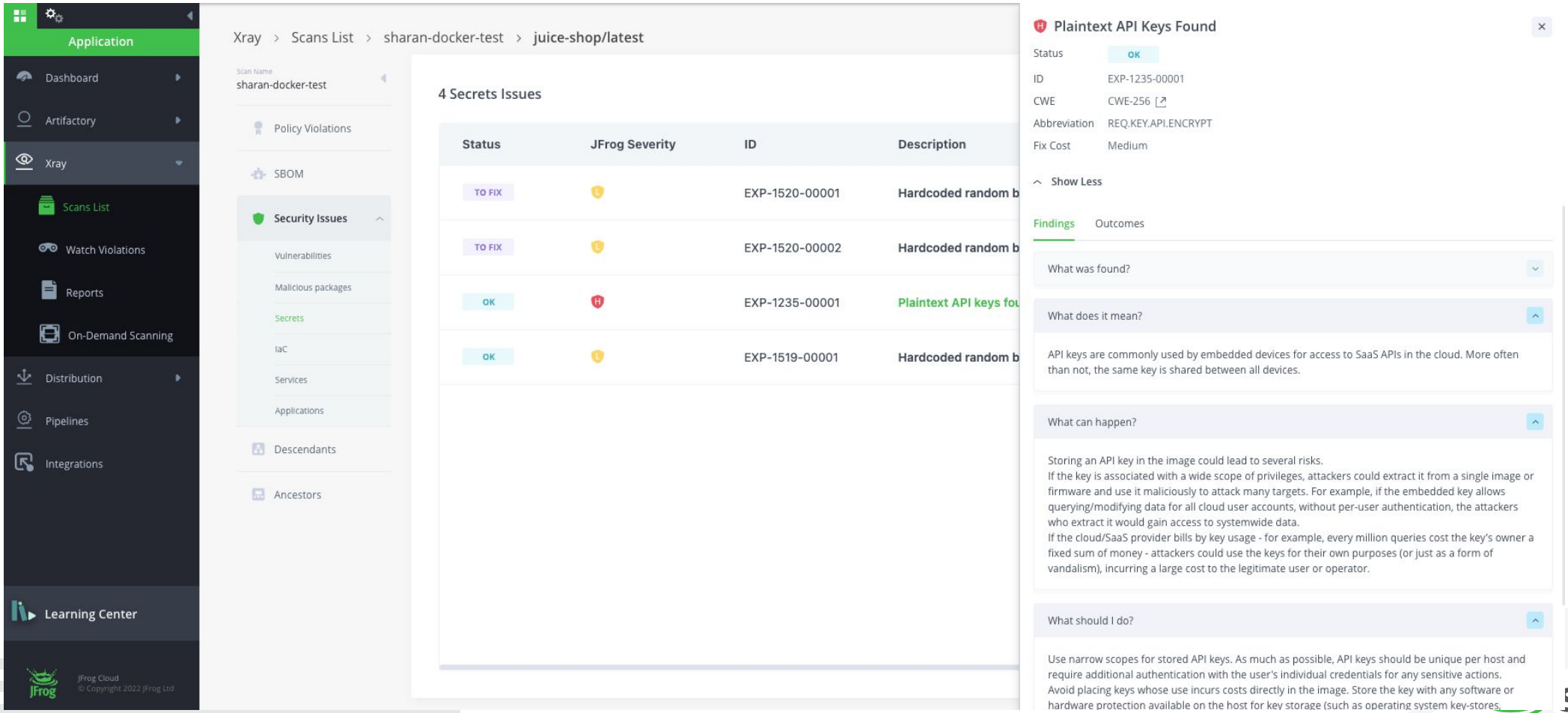
CONTEXTUAL ANALYSIS / APPLICABILITY SCANNER

- Applicability of a CSV to you application, **through analysis of reachable** paths and configurations at the binary level.



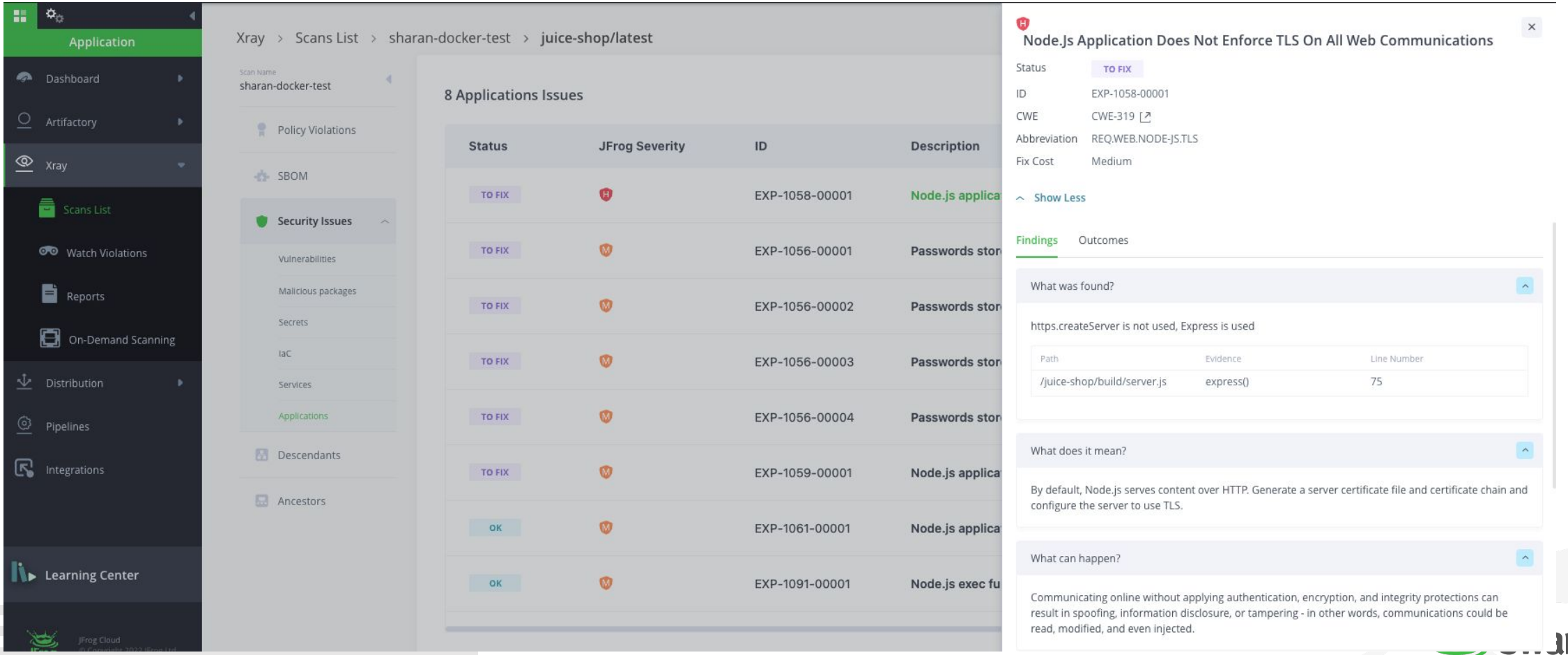
SECRETS DETECTION

- Discover secrets such as **passwords**, **access tokens**, **API keys** that can leaked through config files, IaC modules and binaries.



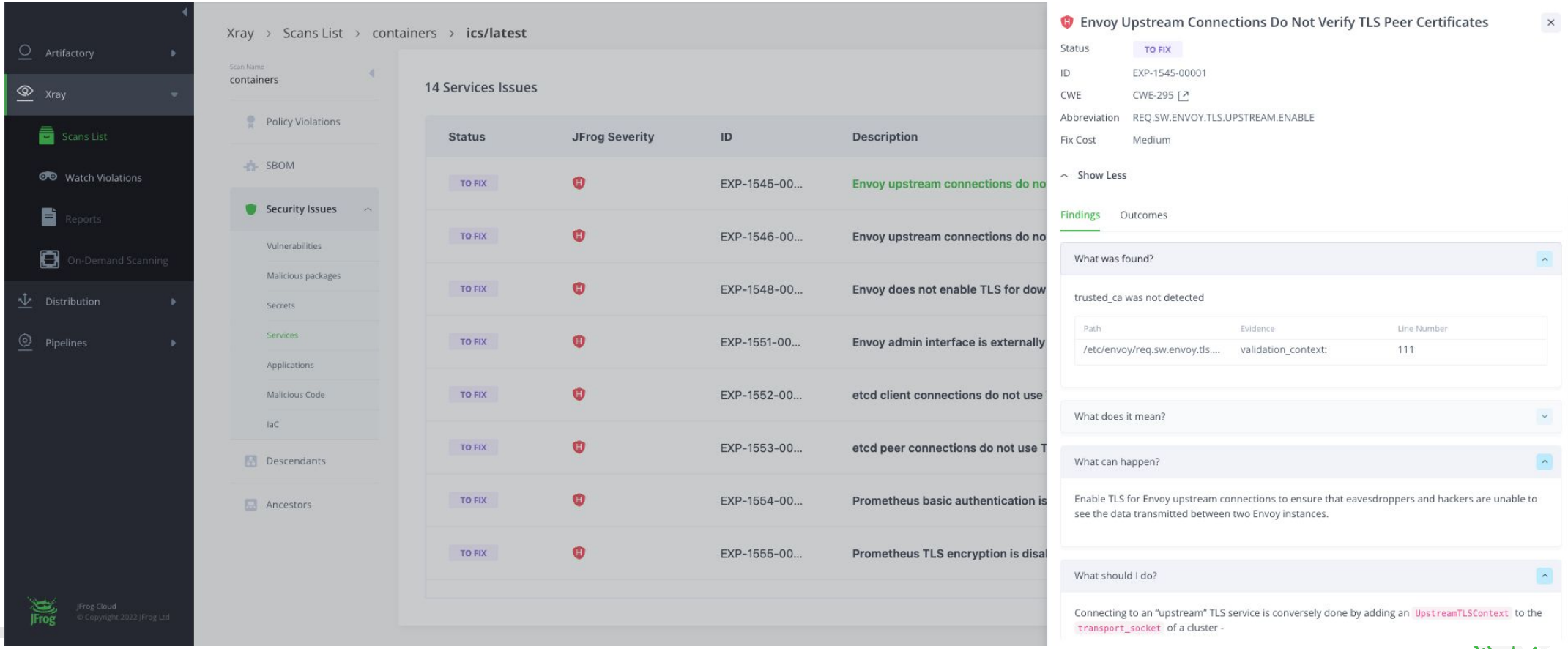
APPLICATION EXPOSURE

- find **configuration issues**, **security malpractices** and **insecure usage of popular libraries** related to your application framework.



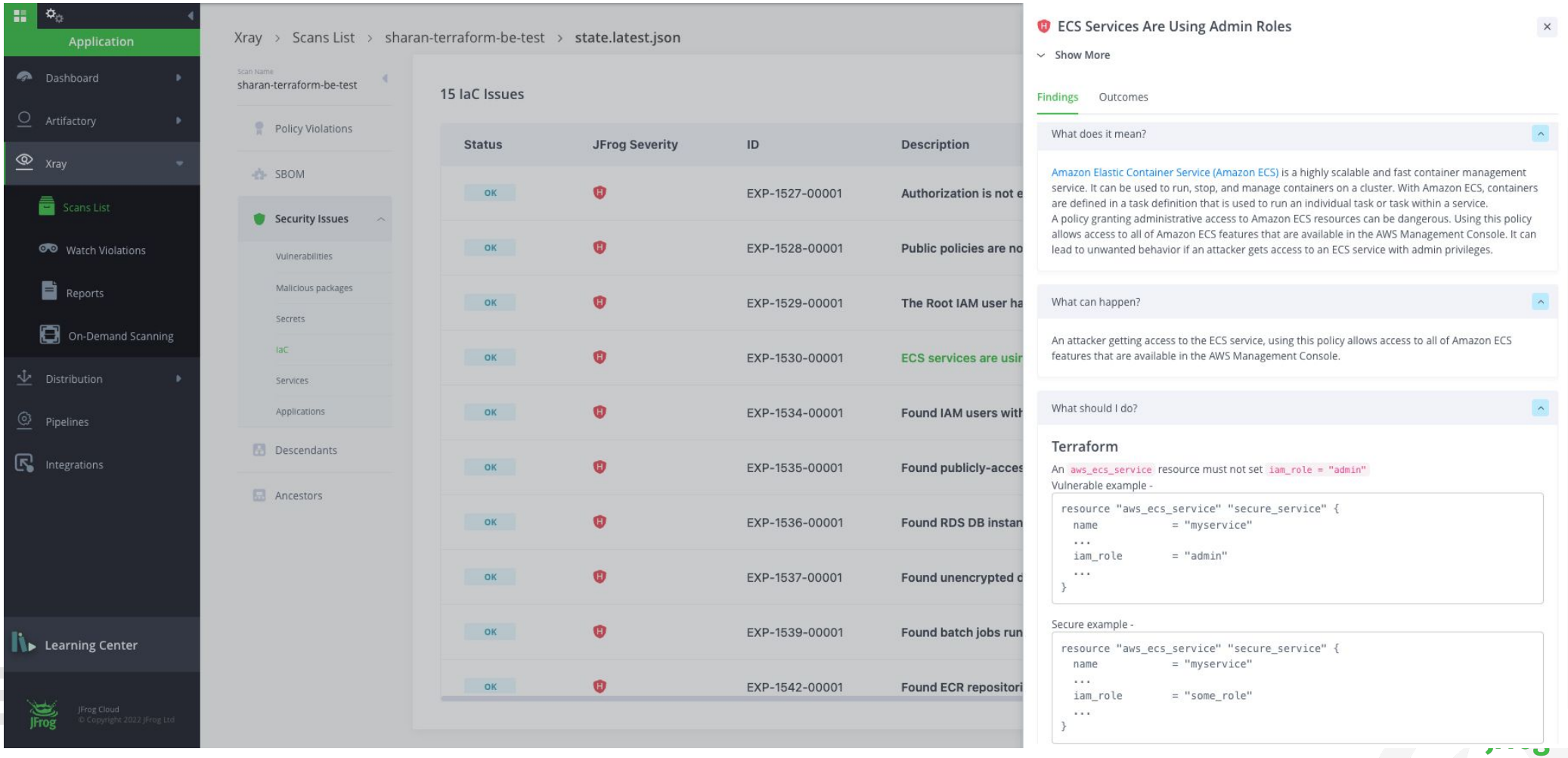
SERVICE EXPOSURE

- detect **configuration issues** and **security malpractices** for specific services and daemons included in your artifacts.



IaC SECURITY ANALYSIS

- Scan your **Terraform** state and module files in Artifactory for **configuration issues** such as **insecure use of credentials**, and **insufficient access restriction** to services.



DEMO

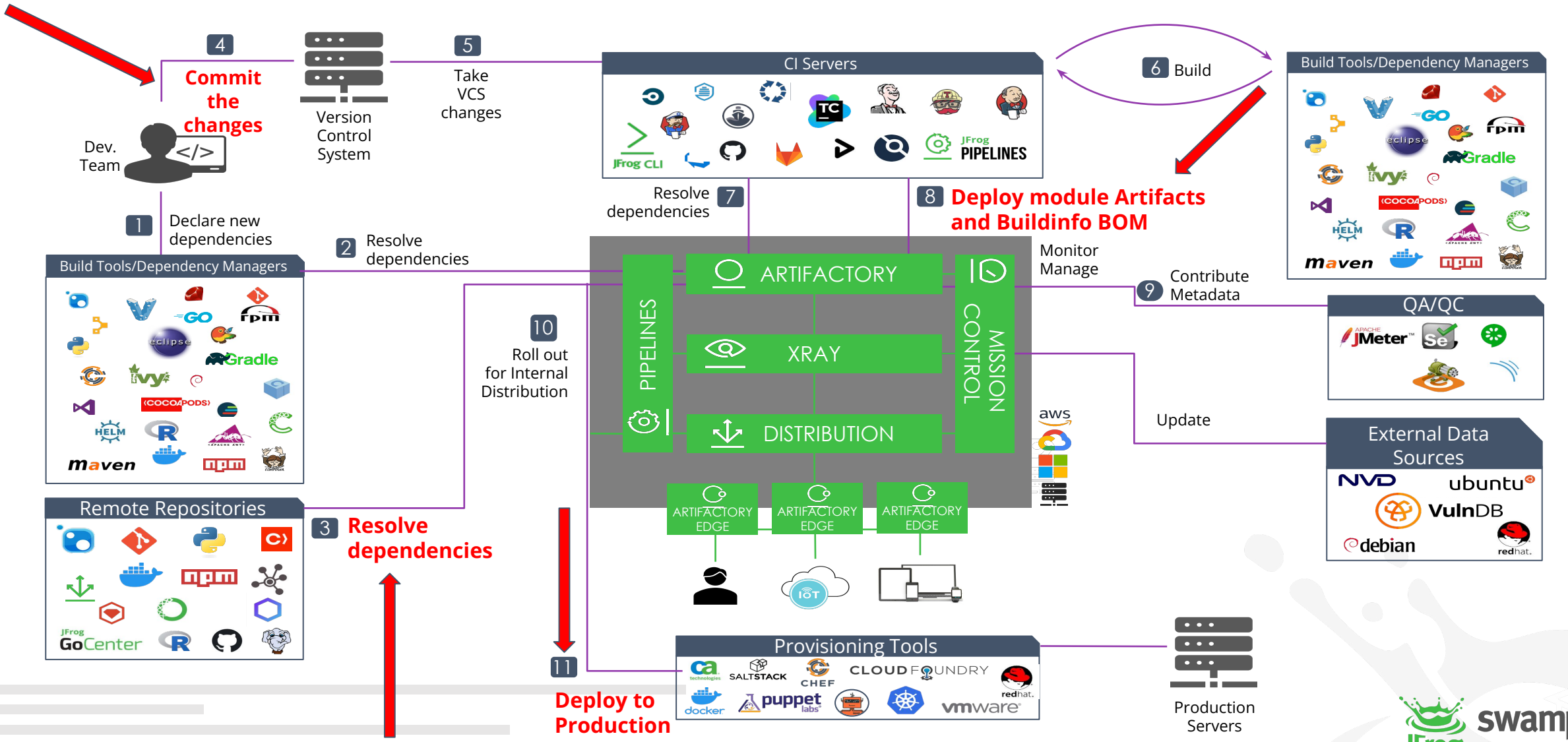
ADDITIONAL FEATURES IN ROADMAP

- Support for Conda Packages scan
- Support for OCI Containers scan
- Archive file scan support : AAR and XZ files
- SPDX and CycloneDX, SBOM has CPE string for identifying software artifact
    - cpe:/ <part>:<vendor>:<product>:<version>:<update>:<edition>:<language>
- Policy framework for Adv. Security Features
- IaC scan
    - For k8s
    - Scan in IDE

CONCLUSION

The Liquid Software Company

WHERE DOES XRAY FIT?



AGENDA

09:00–10:25	DevOps & DevSecOps Overview
	Introduction To JFrog Xray
	Policies and Watches
	Impact Analysis
10:25-10:40	Break
10:40-11:50	IDE Plugin Integration
	CI / CD Integration
	Scan Reports
	New additions/features
11:50–12:00	Summary, Quiz, Survey & Q/A

swampUP
2022 CITY TOUR!

CLASS SURVEY

https://www.surveymonkey.com/r/swampup22NYCtraining

JFrog
The Liquid Software Company

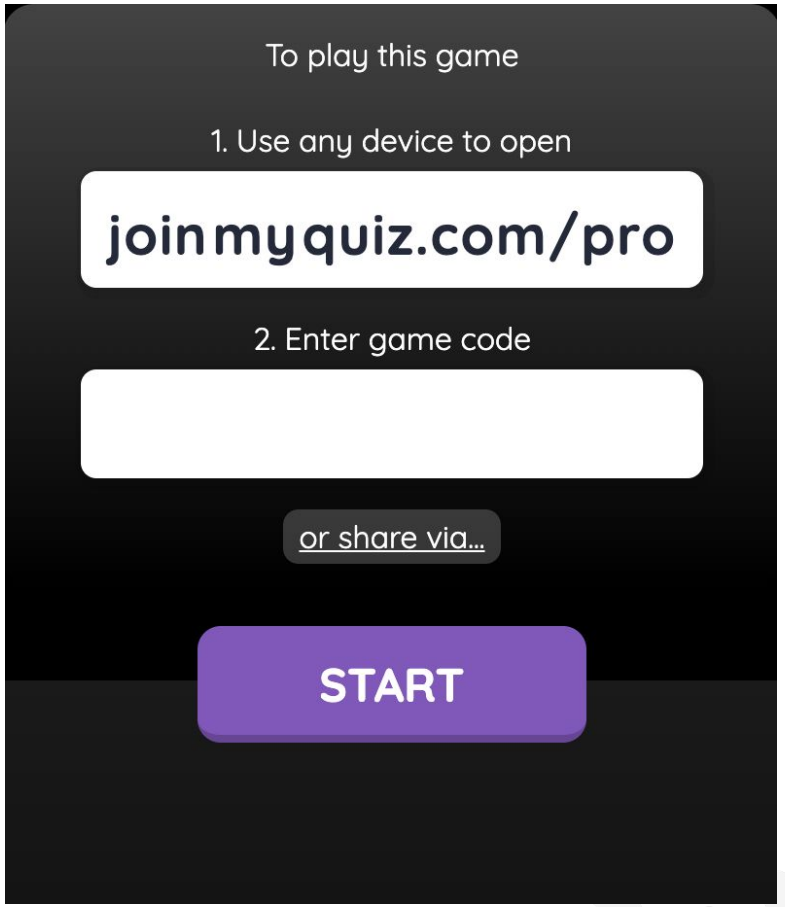
CLASS QUIZ

Please join our quiz!

Game URL will be shared in Class

Enter your Game Code to begin

https://quizizz.com/pro/join?gc=456733



COURSE RESOURCES

- Course material: GitHub
- JFrog Platform Overview
- JFrog Xray Developer Guide
- JFrog Xray REST APIs
- JFrog Xray IDE Integrations
- JFrog Xray CLI
- JFrog Frogbot
- JFrog Security Research
- New Advanced Security Features - wiki
- JFrog Academy: Online Training
- JFrog Free Tier: Start Free
- Adv Security Features Test Drive

JFrog ACADEMY

Q & A

THANK YOU!