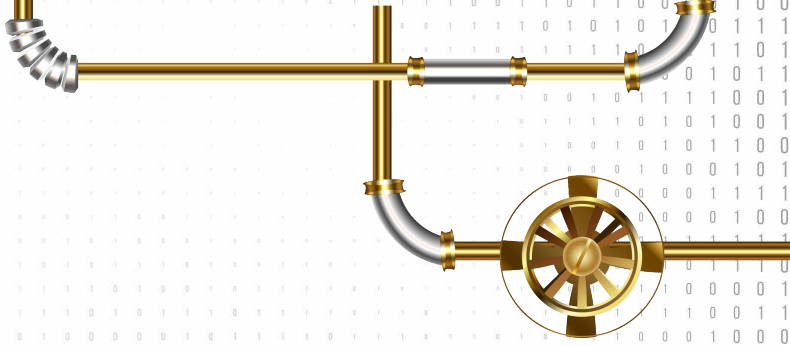
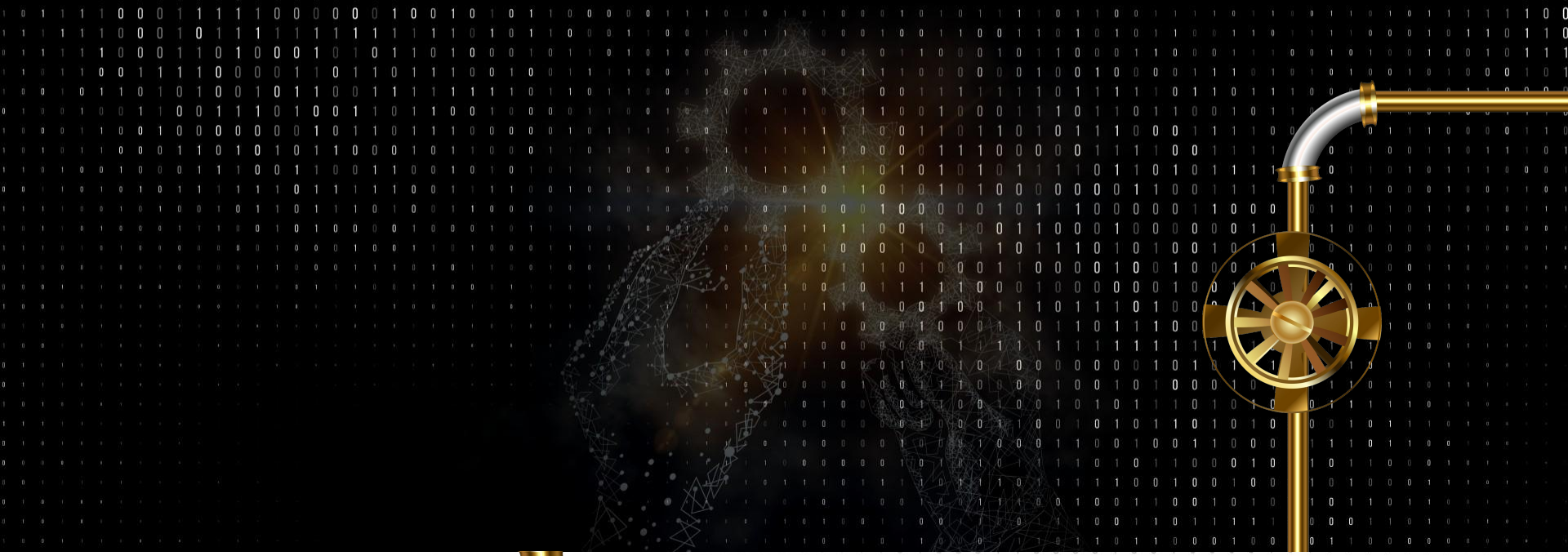
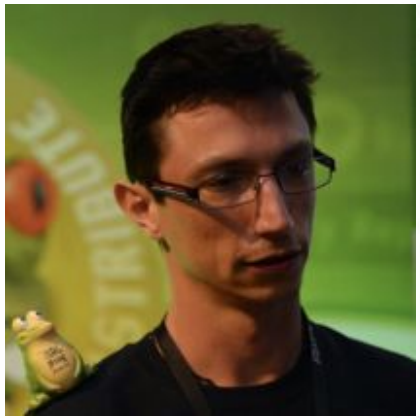


JFrog Xray



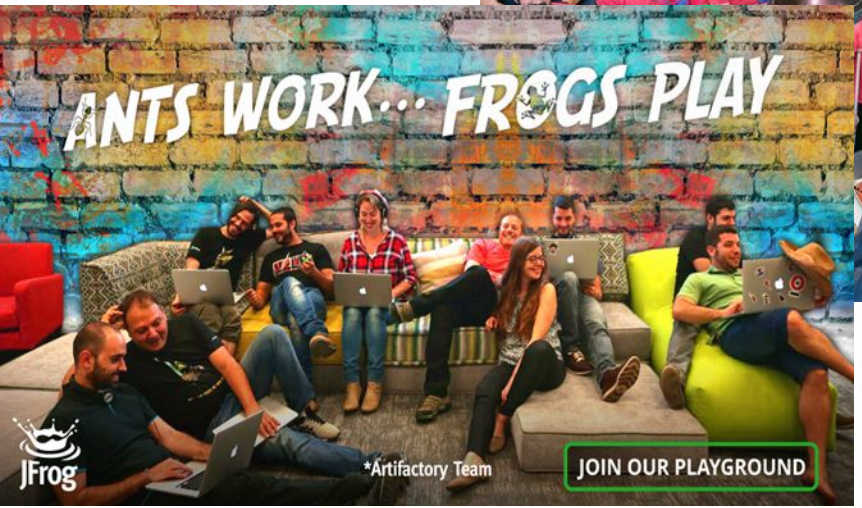
INTRODUCTION

WHO AM I?



Jonathan ROQUELAURE
Senior Solution Architect - Squad Leader
jonathanr@jfrog.com
@roquelaurej

FROGS?



JFROG IN A NUTSHELL

2008
Founded

5,000
Customers

400
Employees

FORTUNE 100 COMPANY
Clients include
70%
of the
Fortune 100

$230M
Raised to date

All Hybrid
From OSS to Multicloud

6 Products
from Git to K8S

9 locations
7 countries

TECHNOLOGY LEADERSHIP

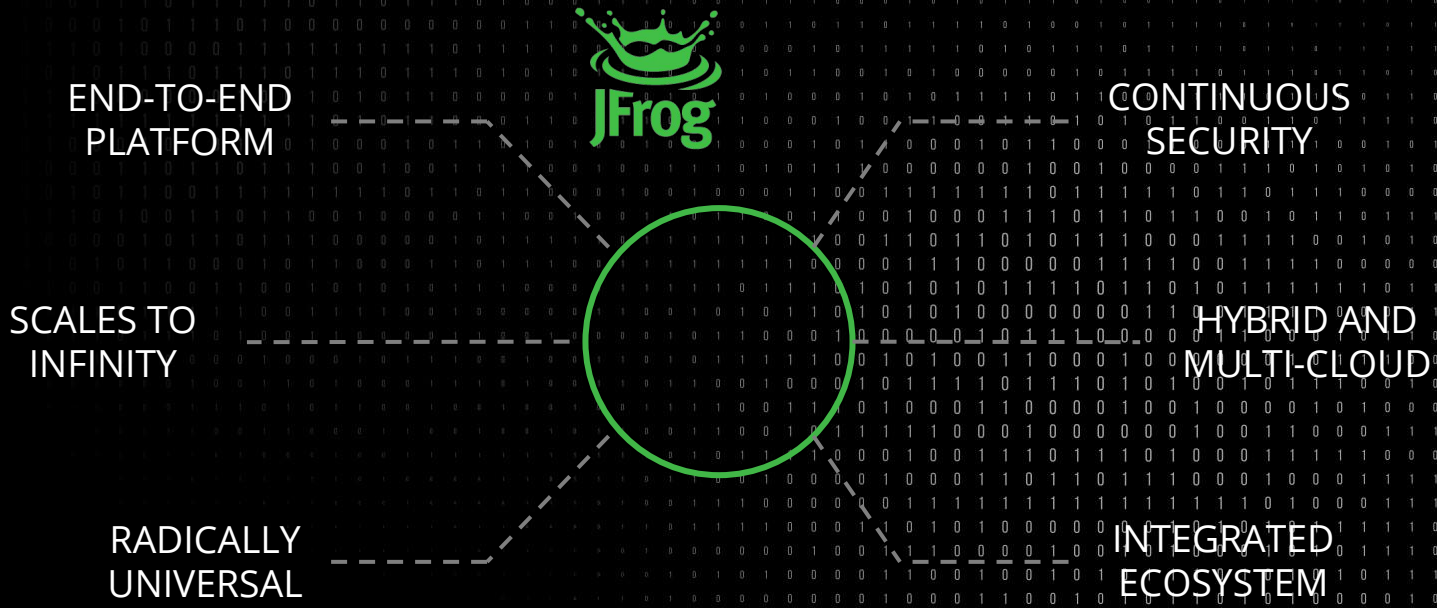
Forbes CLOUD 100 LIST
Forbes CLOUD 100 LIST

Deloitte 2018 Technology Fast 500 Winners
Deloitte Technology Fast 500

The 2018 SD Times 100 Award
SD Times

JFrog

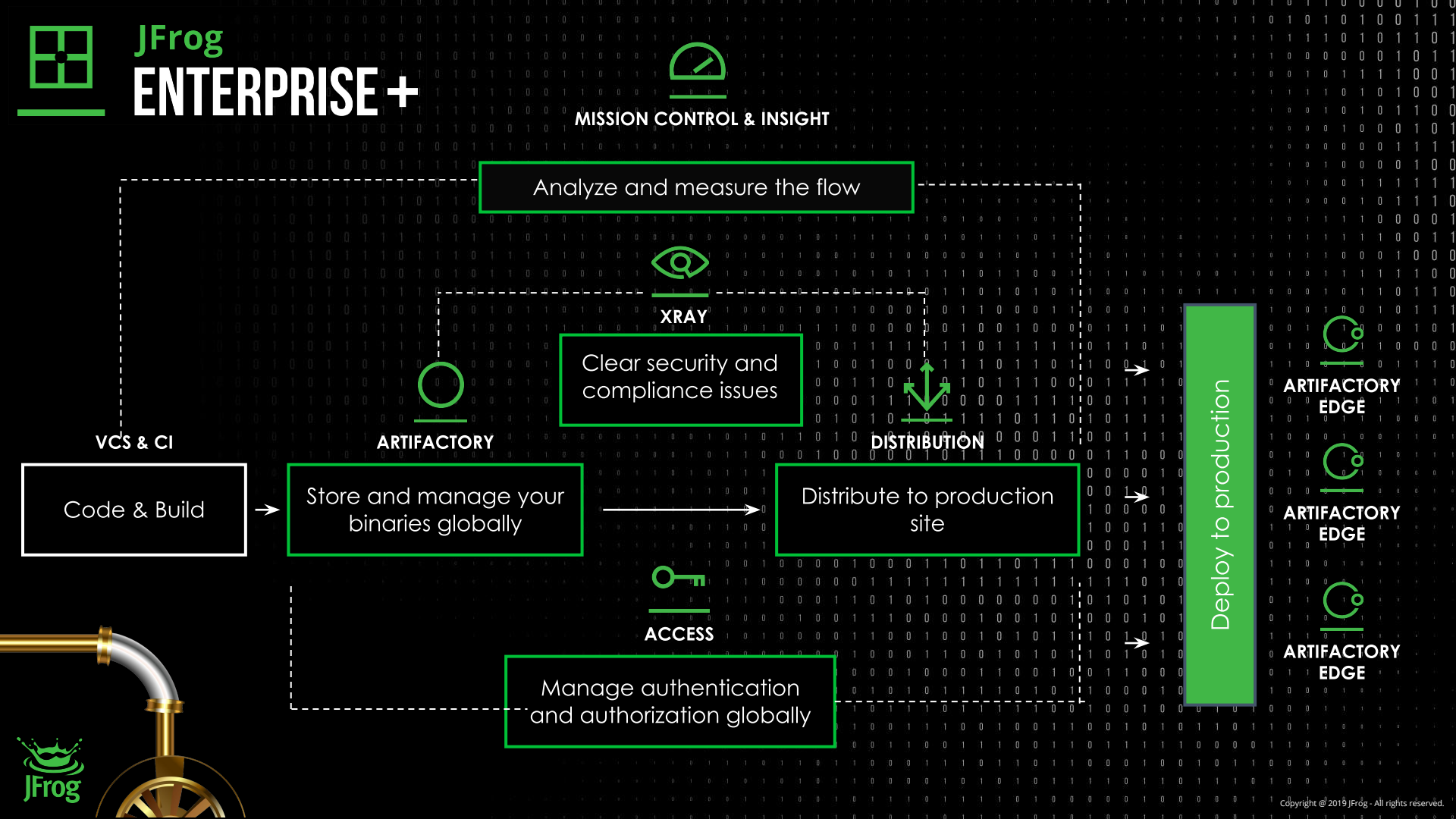
THE FROG PHILOSOPHY

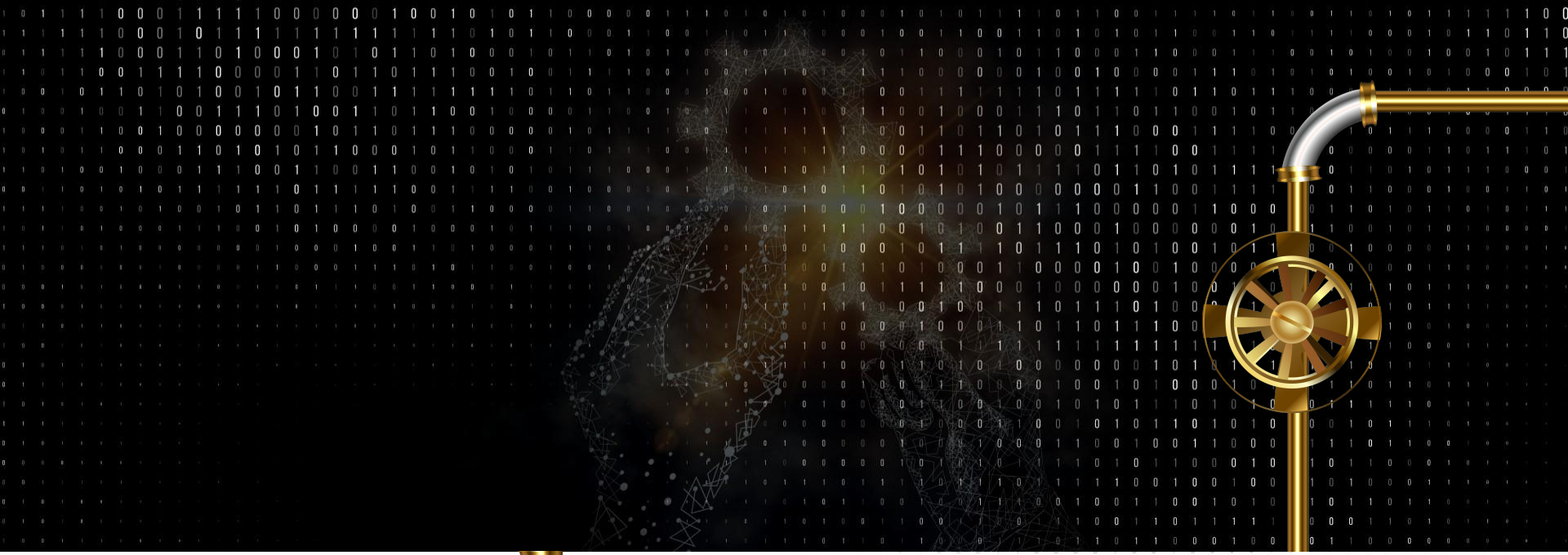


HONORED TO LEAD

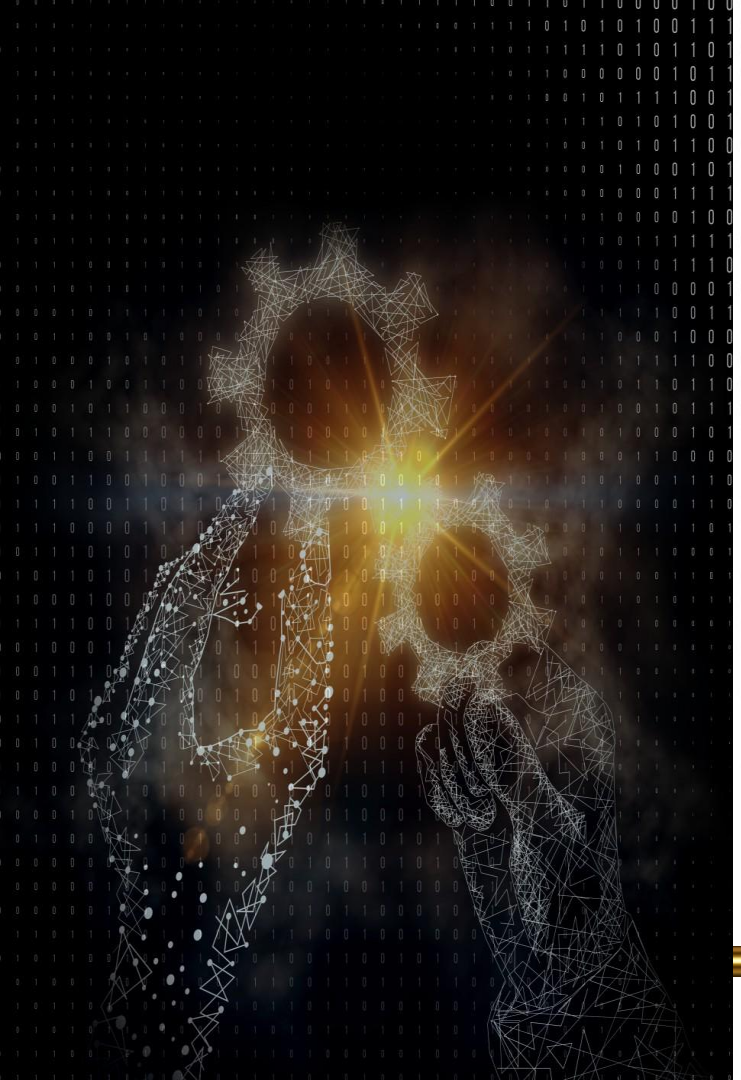
Internet & Software	Google, vmware, amazon, IBM, salesforce, ORACLE, YAHOO!, slack
Technology & Electronics	LinkedIn, intel, cisco, EMC², CHEF, box, ERICSSON
Banking & Finance	BARCLAYS, Fidelity INVESTMENTS, CHASE, citi, Bank of America, CREDIT SUISSE, WELLS FARGO
Engineering & Aerospace	ANSYS, GM, NASA, AIRBUS DEFENCE & SPACE, BOSCH, HALLIBURTON, GE, BOEING
Retail & Consumer	COSTCO WHOLESALE, NETFLIX, GAP OUTLET, VOLVO, macy's, Riot GAMES
Education & Research	Yale, OHIO UNIVERSITY, Stanford University, University of San Diego, OICR, MIT

JFrog





SETUP ENV



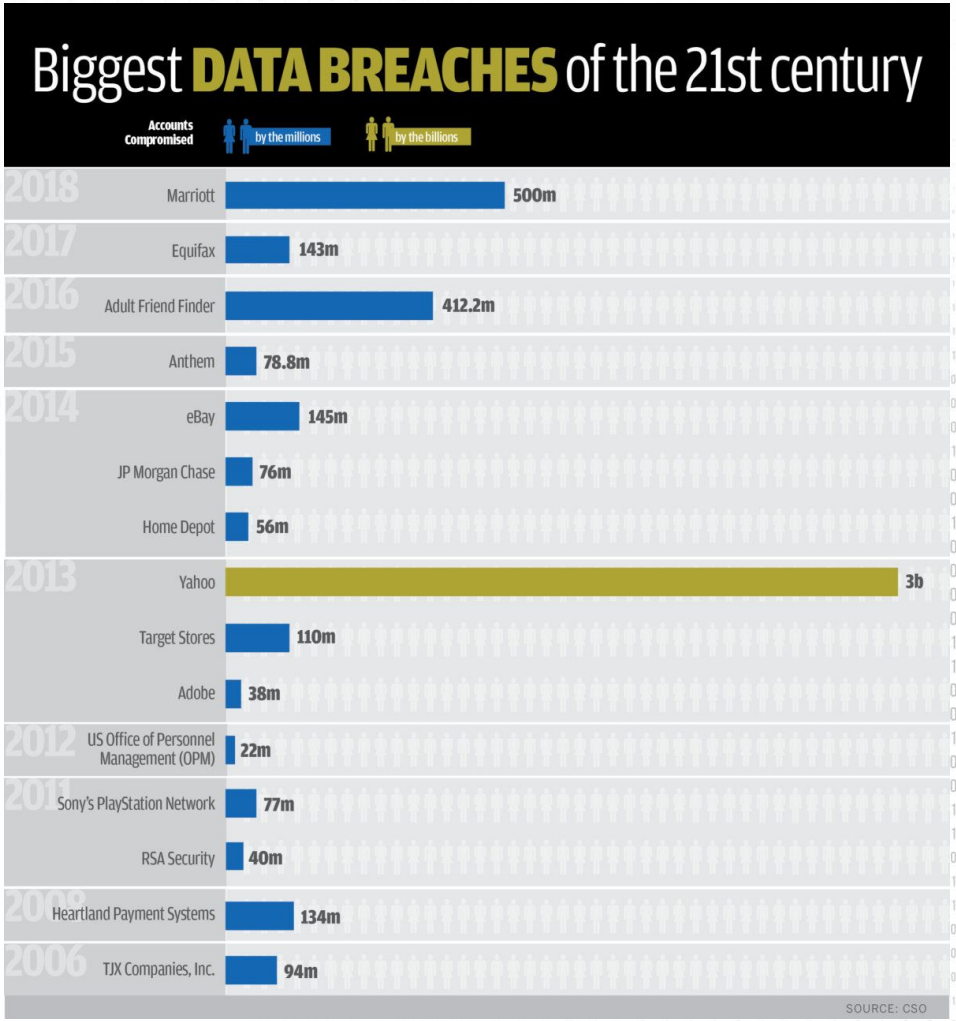
AGENDA

- DevSecOps & General UCs
- Overview & Architecture
- Rules & Policies
- Watches & Violations
- Components & Views
- Use Cases and Integrations
- Administration & Configuration
- Summary and Q&A

Training Objectives

- Get familiar with JFrog Xray architecture
- Comprehend the main features of JFrog Xray and how to use them for different use cases
- Administrate JFrog Xray using these features
- Discover advanced features of JFrog Xray
- Understand which automation tools are available for JFrog Xray and how to use them
- Leverage those tools in your DevSecOps environment

DEVSECOPS & GENERAL USE CASES



What is DevSecOps?

Recap - DevOps

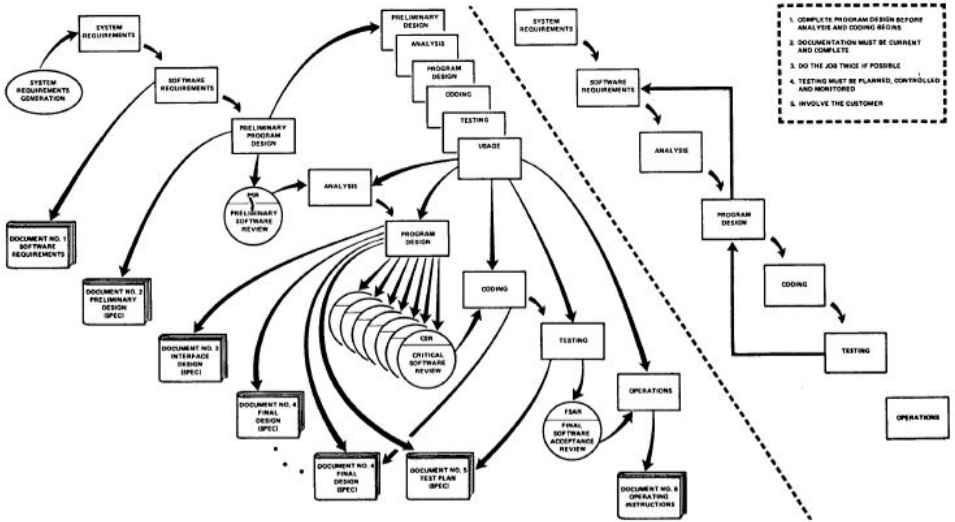


Figure 10. Summary

"I believe in this concept, but the implementation described above is risky and invites failure." -Royce

Recap - DevOps

Figure 10. Summary

"I believe in this concept, but the implementation described above is risky and invites failure." -Royce

Recap - DevOps

Figure 10. Summary

"I believe in this concept, but the implementation described above is risky and invites failure." -Royce

Recap - DevOps

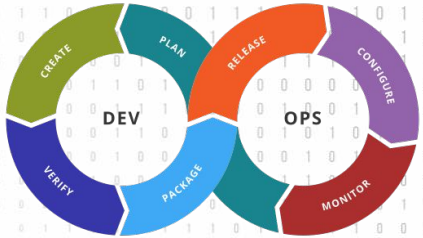
Figure 10. Summary

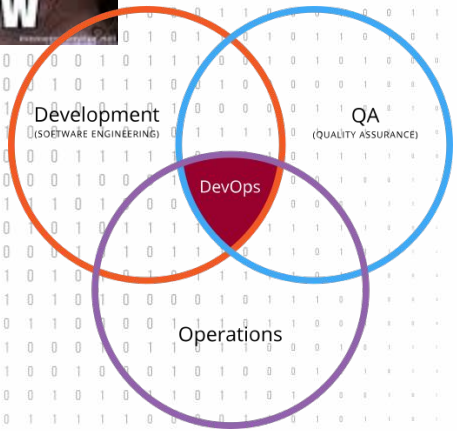
"I believe in this concept, but the implementation described above is risky and invites failure." -Royce

Recap - DevOps

Figure 10. Summary

"I believe in this concept, but the implementation described above is risky and invites failure." -Royce





Recap - DevOps

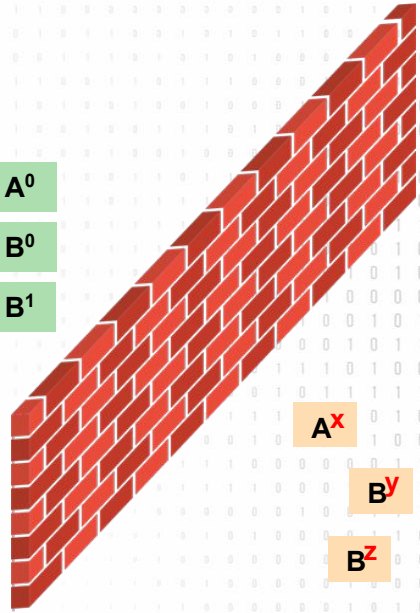
Dev

Plan → Code → Build → Test

A^0

B^0

B^1



Ops

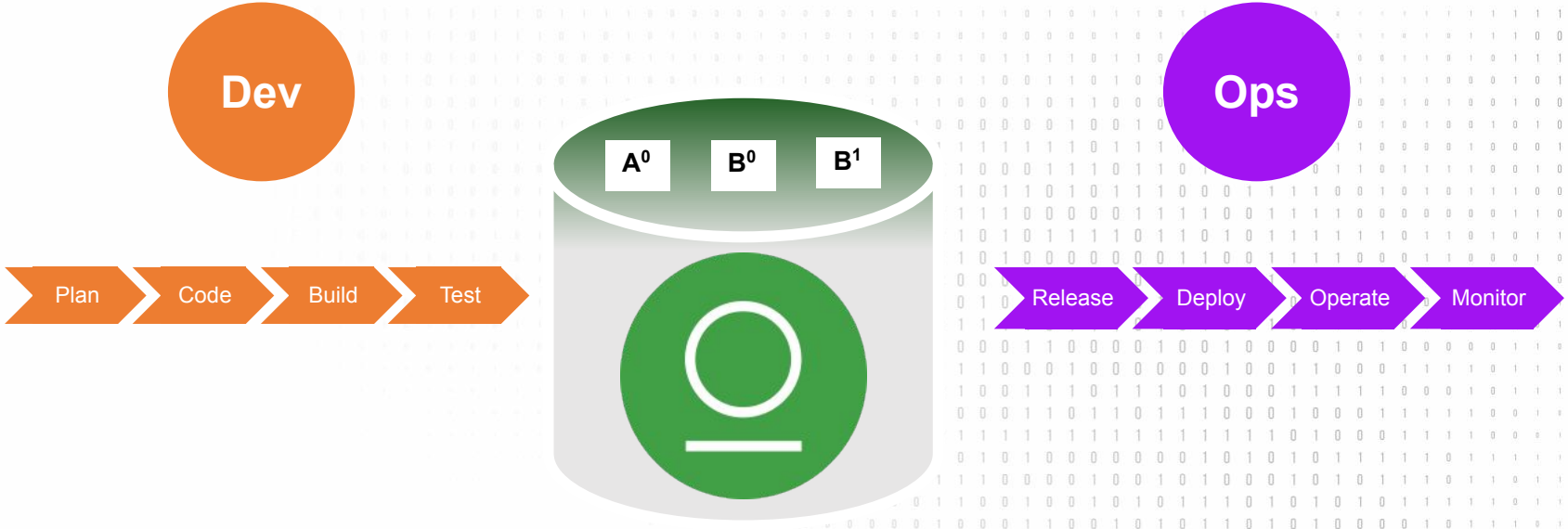
A^x

B^y

B^z

Release → Deploy → Operate → Monitor

Recap - DevOps



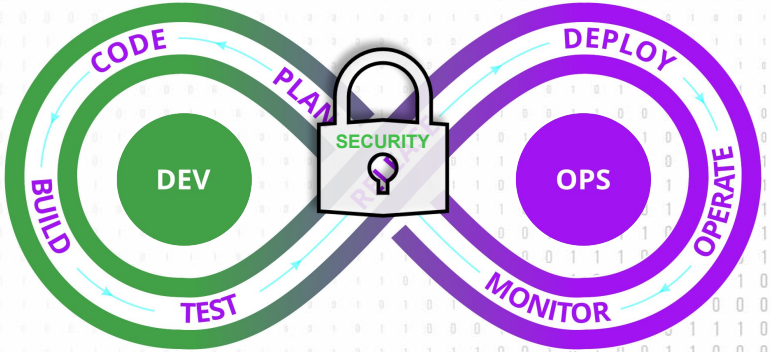
Busting the Myths: DevSecOps

- Myth: Security can't fit into DevOps
- DevSecOps is an Abomination!
- DevOps = DevSecOps
- Security is an inhibitor to DevOps agility
- Adopting DevSecOps = "giving up control"

Busting the Myths: DevSecOps

- DevSecOps is all about speed, letting us deploy anytime, anywhere and in any way
- We'll need to hire "super" developers to implement DevSecOps
- DevSecOps is a capability, let's buy it!!!
- And many more ……..

What is DevSecOps?



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

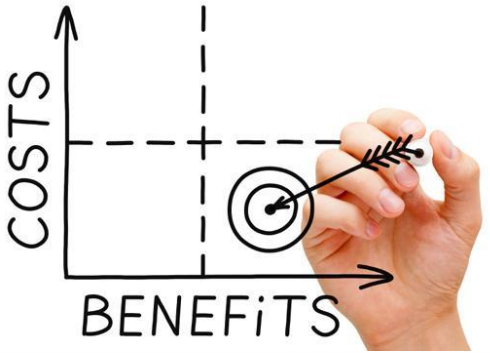
How? Introducing security earlier in the life cycle of application development

What is DevSecOps?

- DevSecOps aims to embed security in every part of the application lifecycle – runtime, build time and even development time.
- It means developing more secure applications faster while refusing to accept that the two (secure & fast) are mutually exclusive!

DevSecOps Benefits

- Increase Production
  - ✔ Keep quality & stability
- Cost Benefit
- Respond to Change Rapidly
- Stay Competitive
- Gain Consistency in Compliance



Creating a *#securityfirst* Culture!

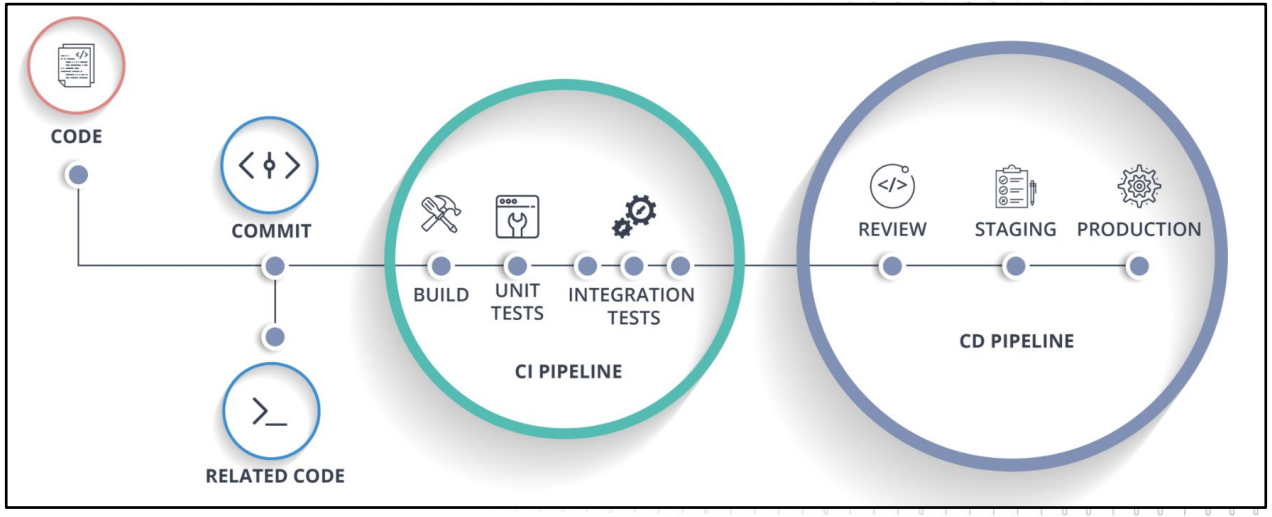
The purpose and result of DevSecOps is to:

- Instill that "everyone is responsible for security”
- Holistic team responsibility v.s 1 team 1 task

  ✔ Focus on team ownership and responsibility
- Bridge gaps between IT and security
- Break down silos and maximize transparency

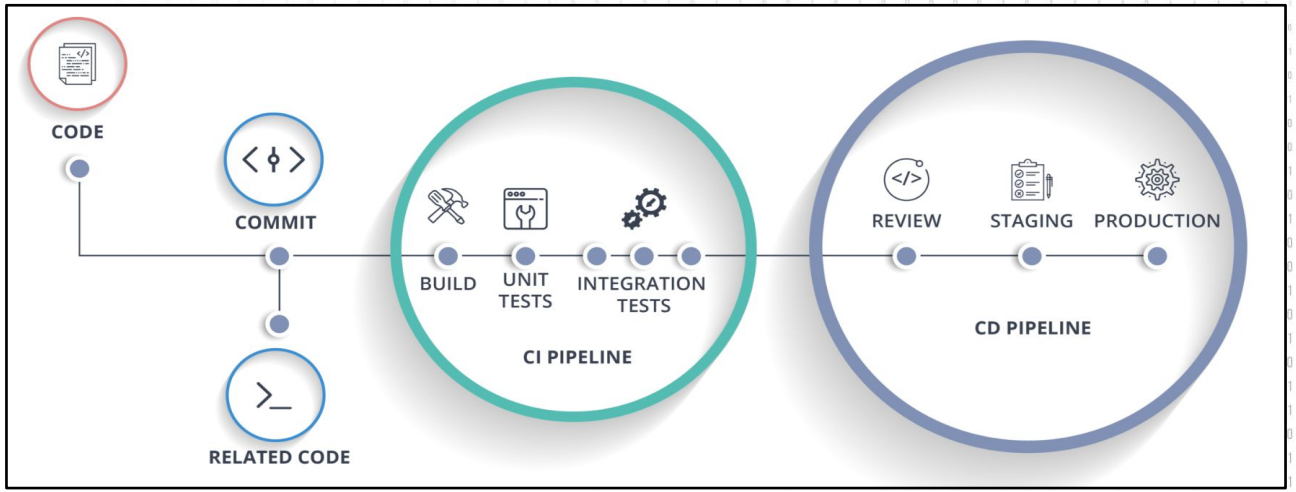
Achieving True DevSecOps

Integrating security practices, tooling, and automation throughout the CI/CD pipeline.



Achieving True DevSecOps

Moving security left in the application development lifecycles



Achieving True DevSecOps

- Inclusion. Adding trained, security minded, and skilled members within the team = metrics + goals are shared by everyone
- Integration. Security professionals are embedded with application development teams. Use them.

Implementation

1. Understand what's at stake if you don't get security right
2. Know what tools you're using
3. Drop what you don't need
4. Find & fix current vulnerabilities
5. Monitor for new vulnerabilities

Stay alert!

Implementation examples

1. **No secret exposures**: works to guarantee that no one password, passphrase, certificate chain, private key will be leaked
2. **TLS is mandatory**: Employ a tool to monitor certificate expiration
3. **No library vulnerability:** scans the projects dependencies, o.s. libraries, o.s. services and o.s. utilities
4. **No code vulnerability**: statically scans the source code for vulnerabilities and bugs
5. **OWASP Top 10:** scans app for vulnerabilities

JFrog & Security…

Open Source is Awesome

Share Your Work

Reuse What Others Built

Focus on Creating Your Own New Thing

Open Source Usage Has Exploded

78% of Enterprises Use Open Source

Is Security a Concern When Adopting OSS?

Number 1 concern: 13%

Number 2 concern: 29%

Number 3 concern: 21%

(Total: 63%)

Source: Wipro

Open Source != Closely Inspected

Open Source != Secure

Open Source != Insecure Either!

Newly Disclosed Vulnerabilities Are Found On Old Code

Open Source is Less Tested for Security

OS Project Owners not aware/budgeted for security

OS consumers not engaged/aware of code

Attackers are Targeting Open Source

One vulnerability, many victims

How Do We Consume OSS?

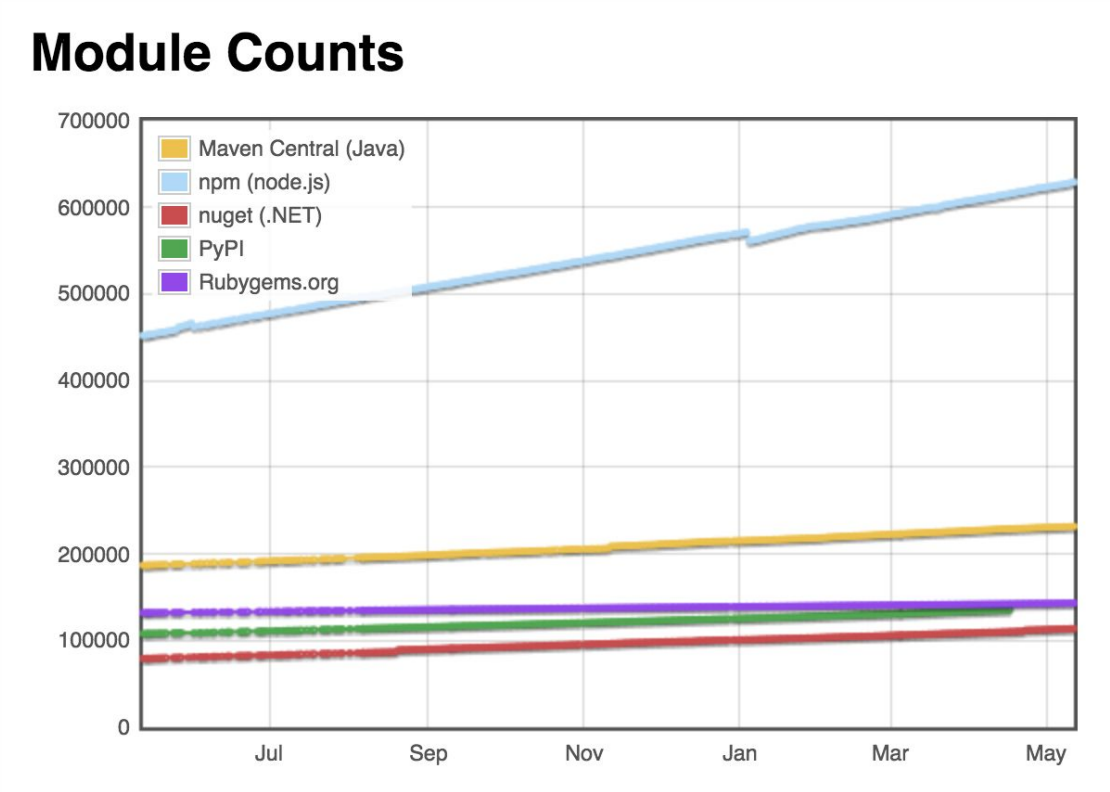
2000:
Select Open Source Providers

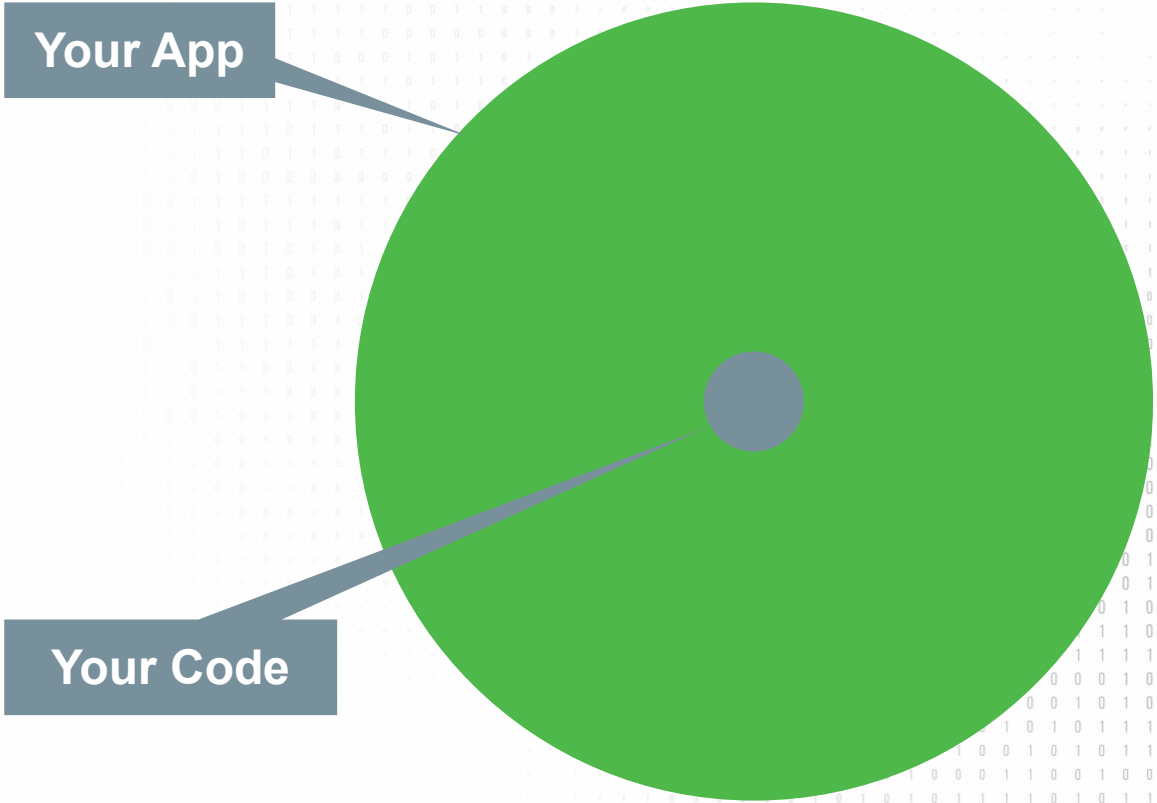
Apache, Linux, IBM, OpenSSL…

2015:
Open Source Marketplaces

Everybody is a provider

Available Open Source Package growth in the last year




Your App
Your Code

Each Dependency is a Security Risk

Do You Know Which Dependencies You Have?

Do you know, for

EVERY SINGLE DEPENDENCY

if its Developers have any

SECURITY EXPERTISE?

Do you know, for

EVERY SINGLE DEPENDENCY

if it went through any

SECURITY TESTING?

Do you know, for

EVERY SINGLE DEPENDENCY

if it has

KNOWN VULNERABILITIES?

~30%
of Docker Hub images carry Known Vulnerabilities

High Priority known vulnerabilities, to be exact

Source: BanyanOps Analysis

Do You Have Known Vulnerabilities in Your Code?

Do you even know?

What Can You Do?

~~Not Use Third Parties~~

Third Party

BINARIES

Third Party

CODE

Test for Known Vulnerabilities in Build (CI) & Deploys (CD)

1 Know What You're Using

2 Drop What You Don't Need

3 Find & Fix Current Vulns

4 Monitor For New Vulns

Register to Security Alerts

Platform Specific

Ubuntu

Node.js

OpenSSL

(your vendor sec list)

Broad Lists

US-CERT

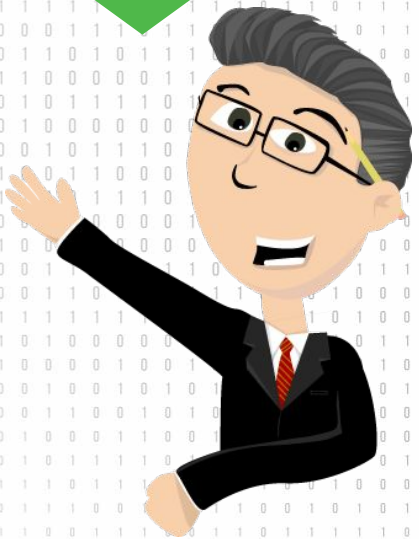
NVD

OSVDB


BASED ON A
TRUE STORY

Day 1

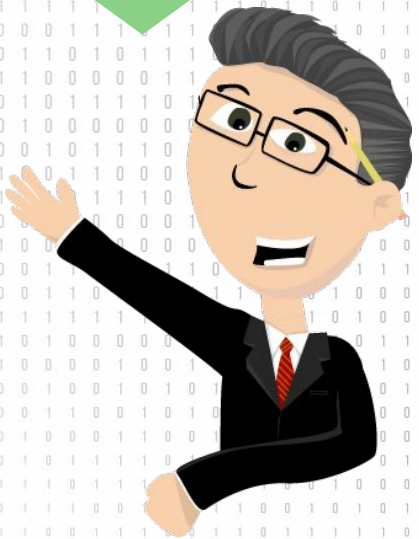
A new CVE was just made public. What is the impact?



Ok .. .. .. ☹.

Day 1.5

A new CVE was just made public. What is the impact?

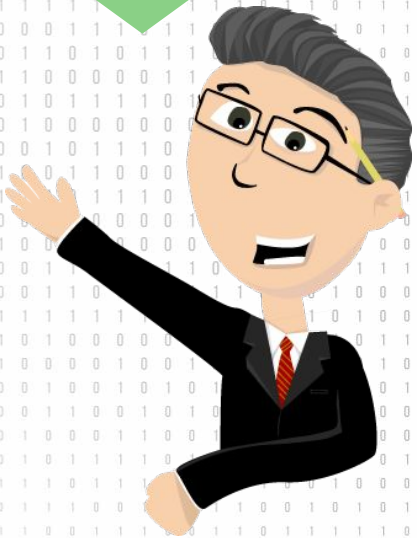


I looked at the internet, the issue is in a jar file.

Day 2

Oh crap, I just moved towards containerized microservices architecture.

A new CVE was just made public. What is the impact?



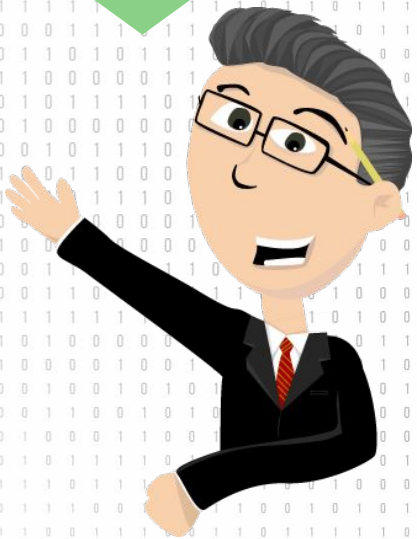

USE DOCKER THEY SAID
IT'LL BE FUN THEY SAID
memegenerator.net


Honest Status Page
@honest_update
Follow
We replaced our monolith with micro services so that every outage could be more like a murder mystery.
RETWEETS
1,987
LIKES
1,435
1:10 AM - 8 Oct 2015

Day 2

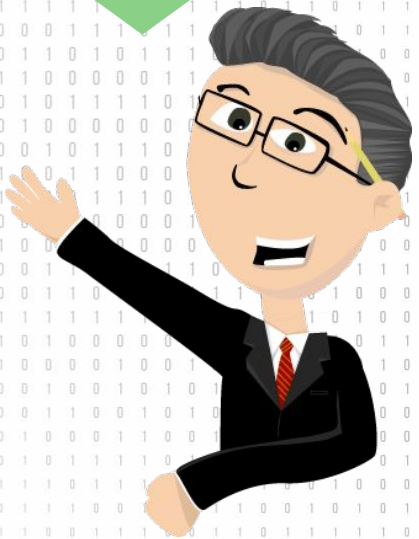
We are polyglot.
And there are a lot of binaries

A new CVE was just made public. What is the impact?



Day 4

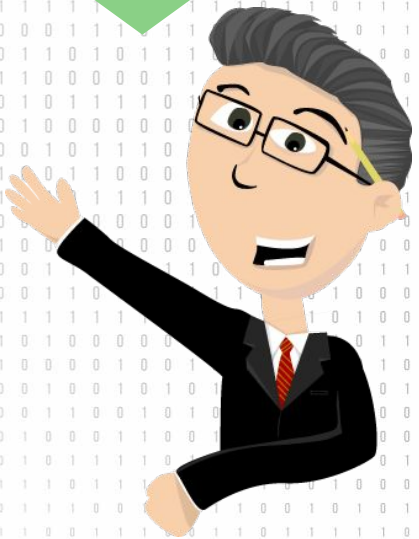
I haven’t heard back. Which business units are impacted?



Broadcasted about the vulnerable library. Still working on it .. .. .. ☹.

Day 4.5

I haven’t heard back. Which business units are impacted?

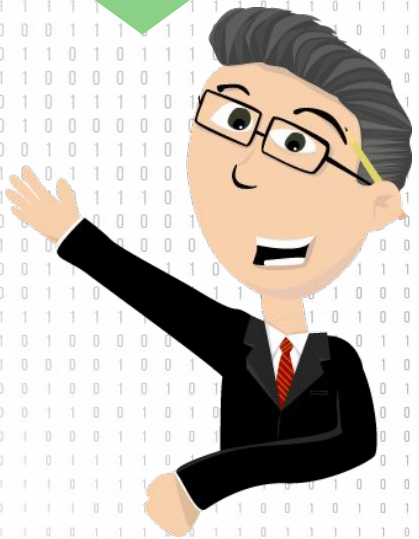


This jar file can be in a war file, zip file, Docker image.

Day 5

I need to find the CI jobs responsible to produce such binaries.

I haven't heard back. Which business units are impacted?

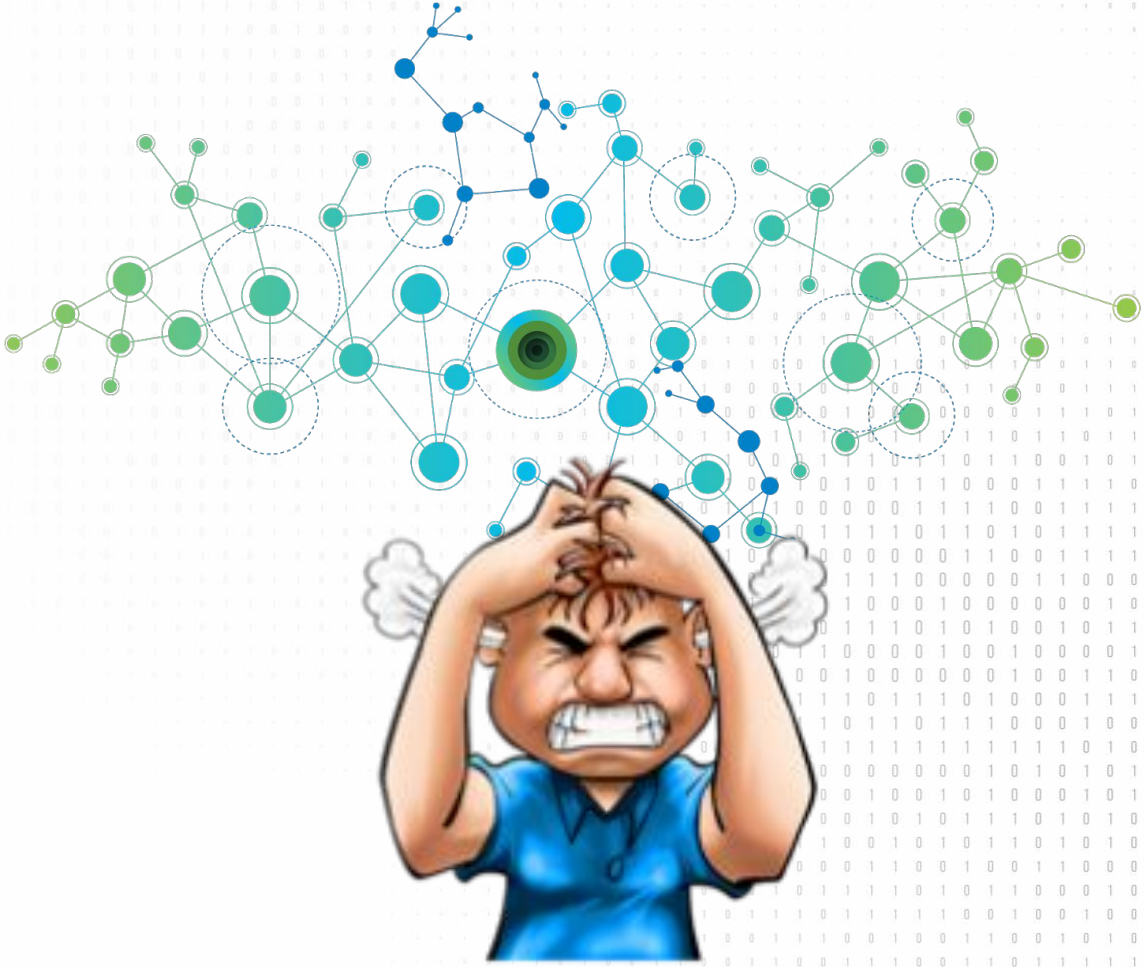


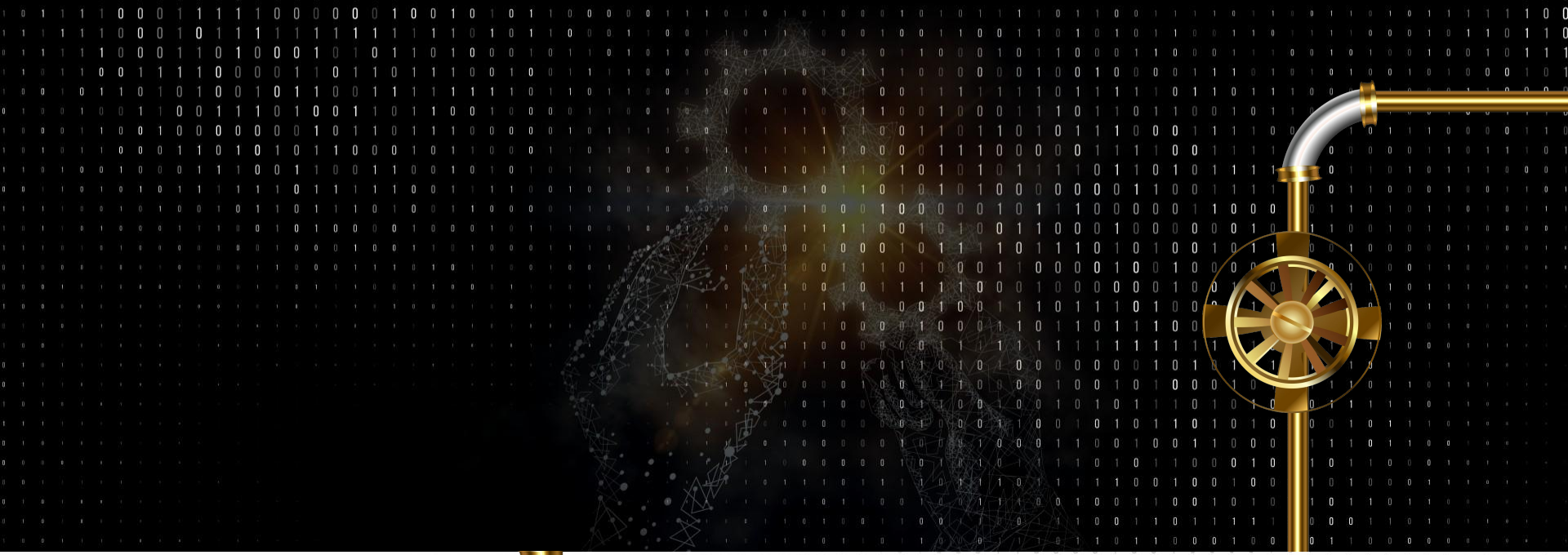
Day 6

Are we running this in production?

I haven’t heard back. Which business units are impacted?

Day 7

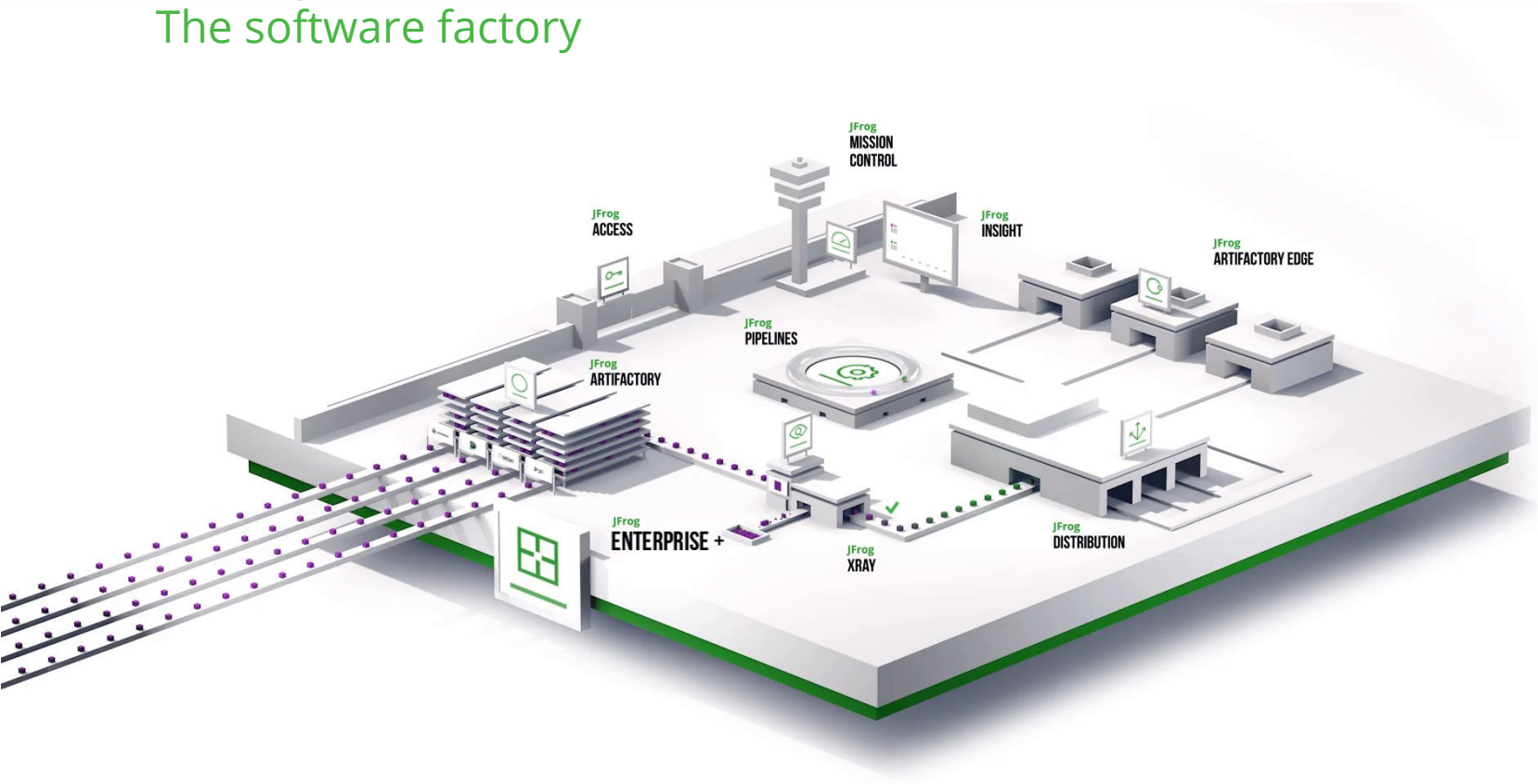




OVERVIEW & ARCHITECTURE

JFrog Platform

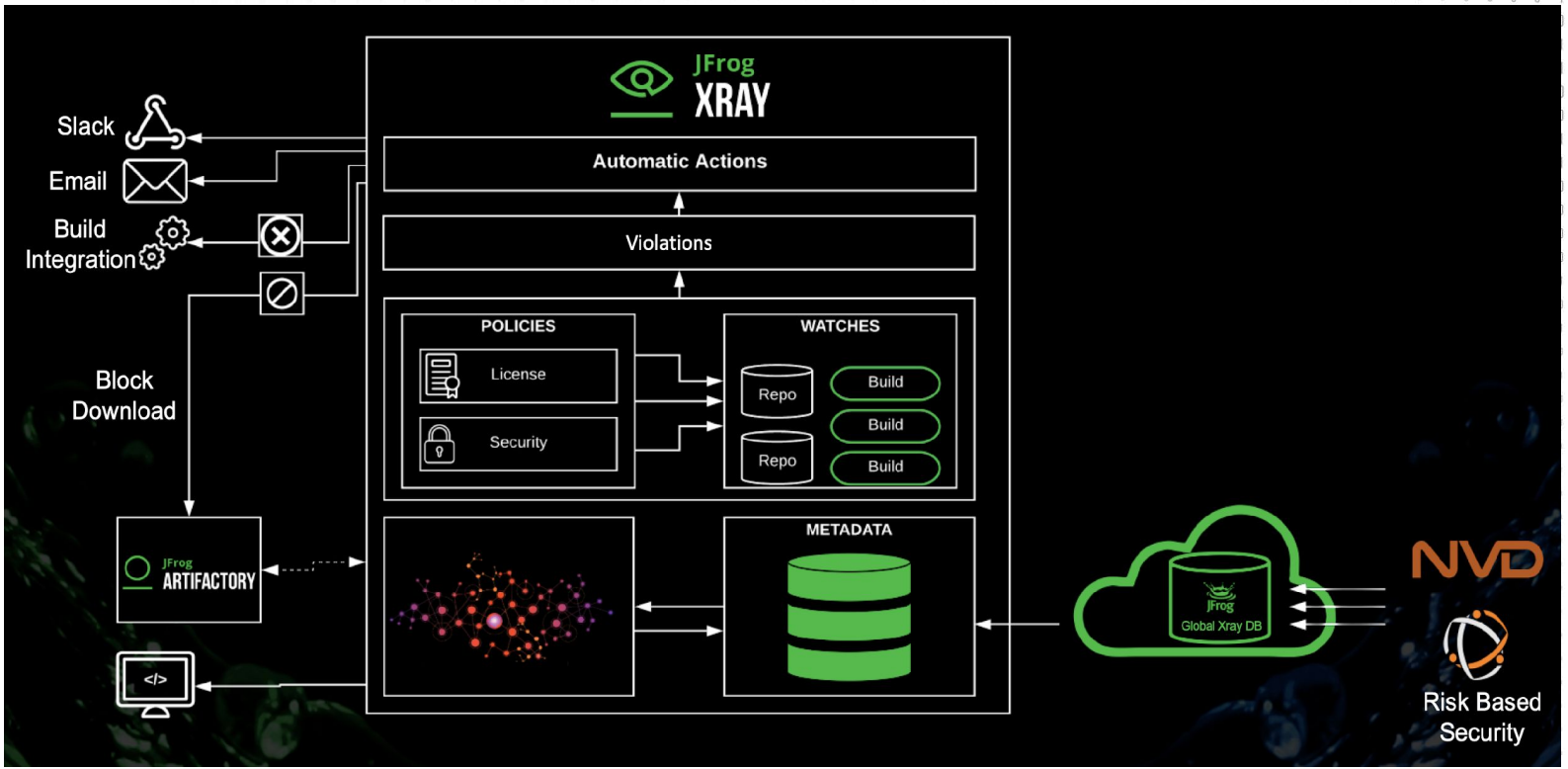
The software factory



What is Jfrog Xray

- Static binary scanning tool
- Not stand alone
- Part of the JFrog products suite
- Recursive scanning tool
- Provides impact analysis
- Provides continuous scanning

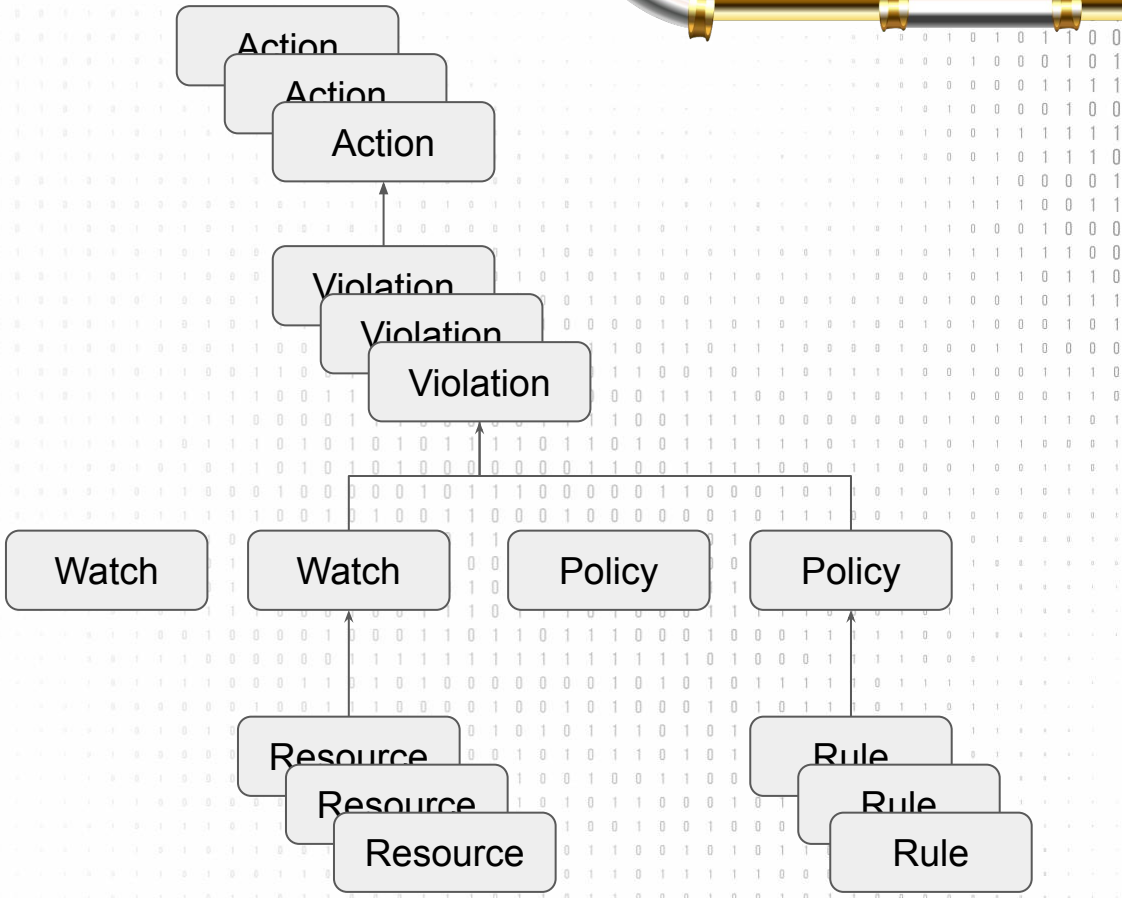
High Level Architecture



Xray terminology

Logical Entities

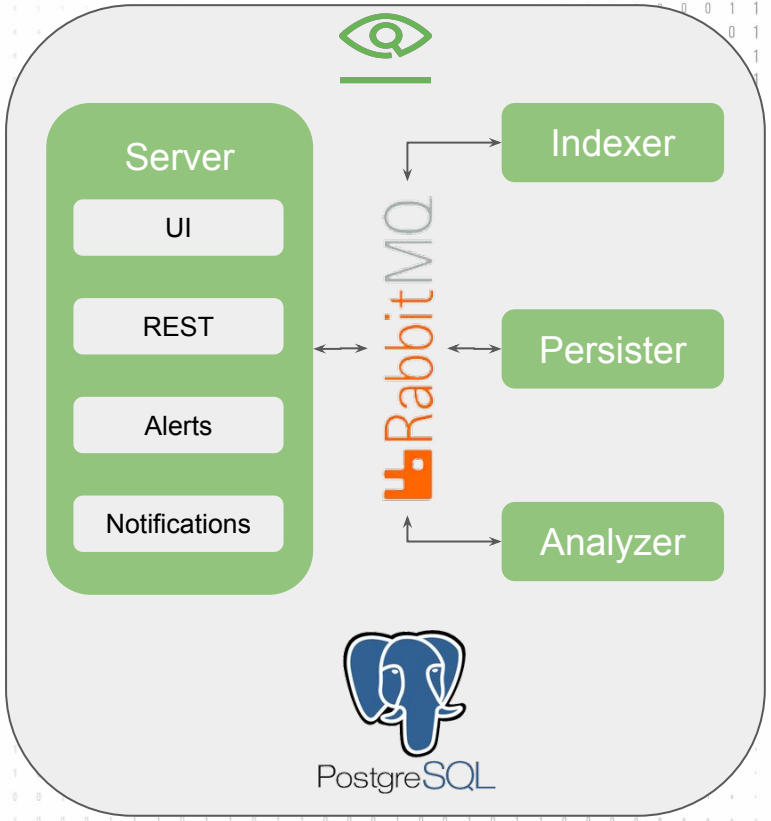
- Rules
- Policies
- Resources
- Watches
- Violations
- Actions



Xray Flows

High Level

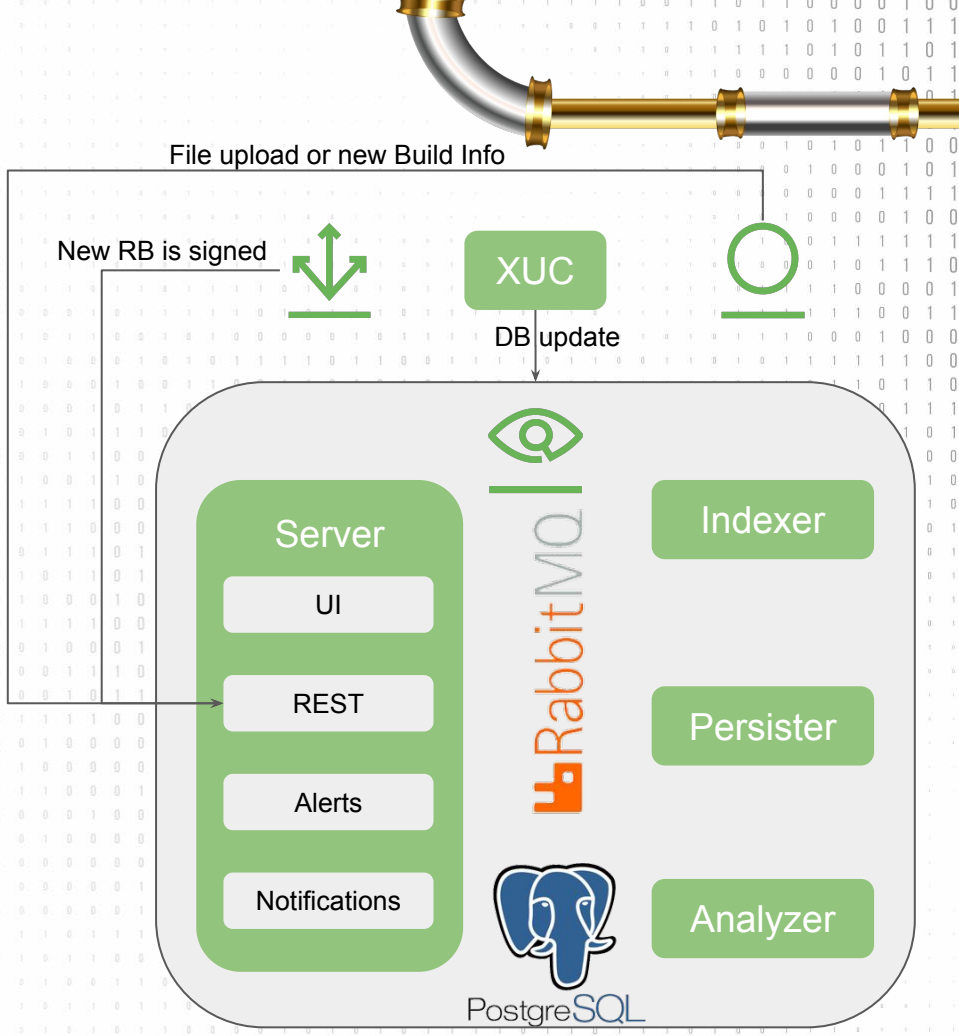
- Written in GO
- 6 different microservices
  - Indexer - fetches the component and creates the graph
  - Persist - Save the Graph
  - Analysis - Scanning & Impact
  - Server - Multiple roles
  - 3rd party services
- MQ based Internal communication
- PostgreSQL DB



Xray Flow

High Level - external

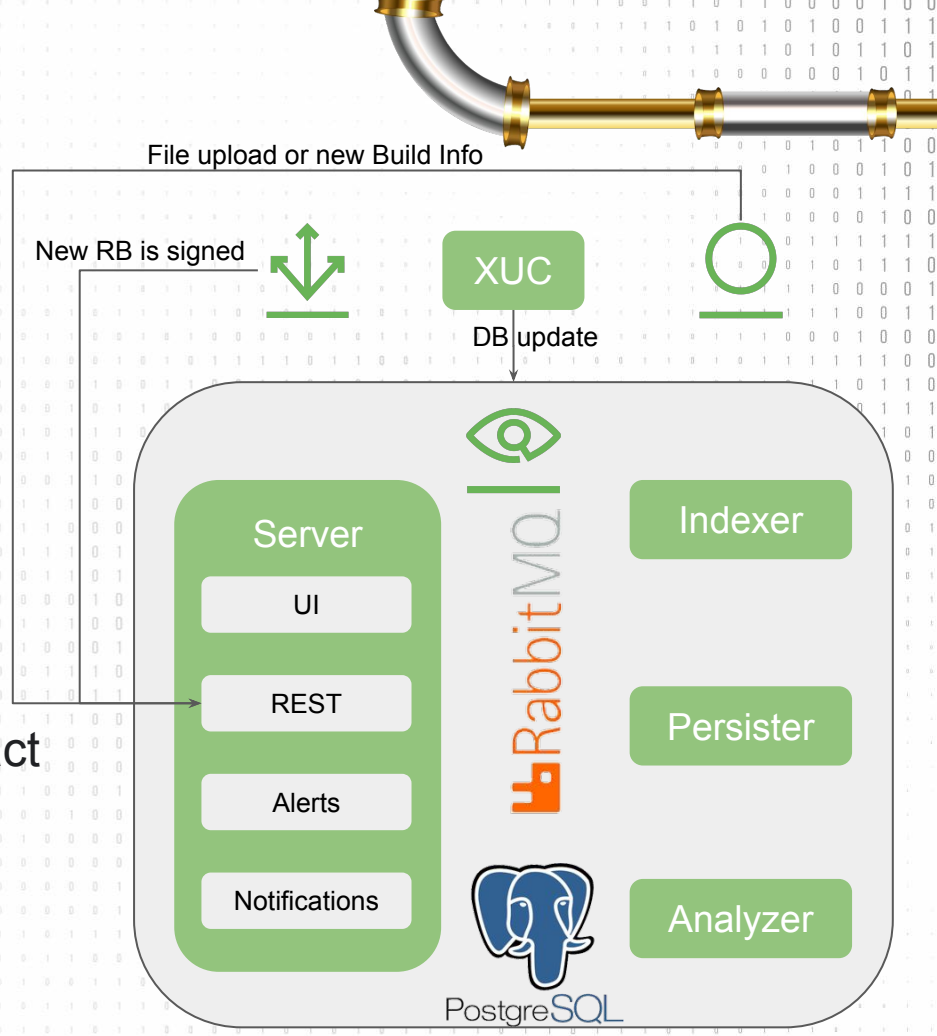
- File upload onto artifactory
- New Build info published (based on any CI/CD integration)
- New release bundle is created and signed in Distribution
- Xray DB is updated



Xray Flow

High Level - internal

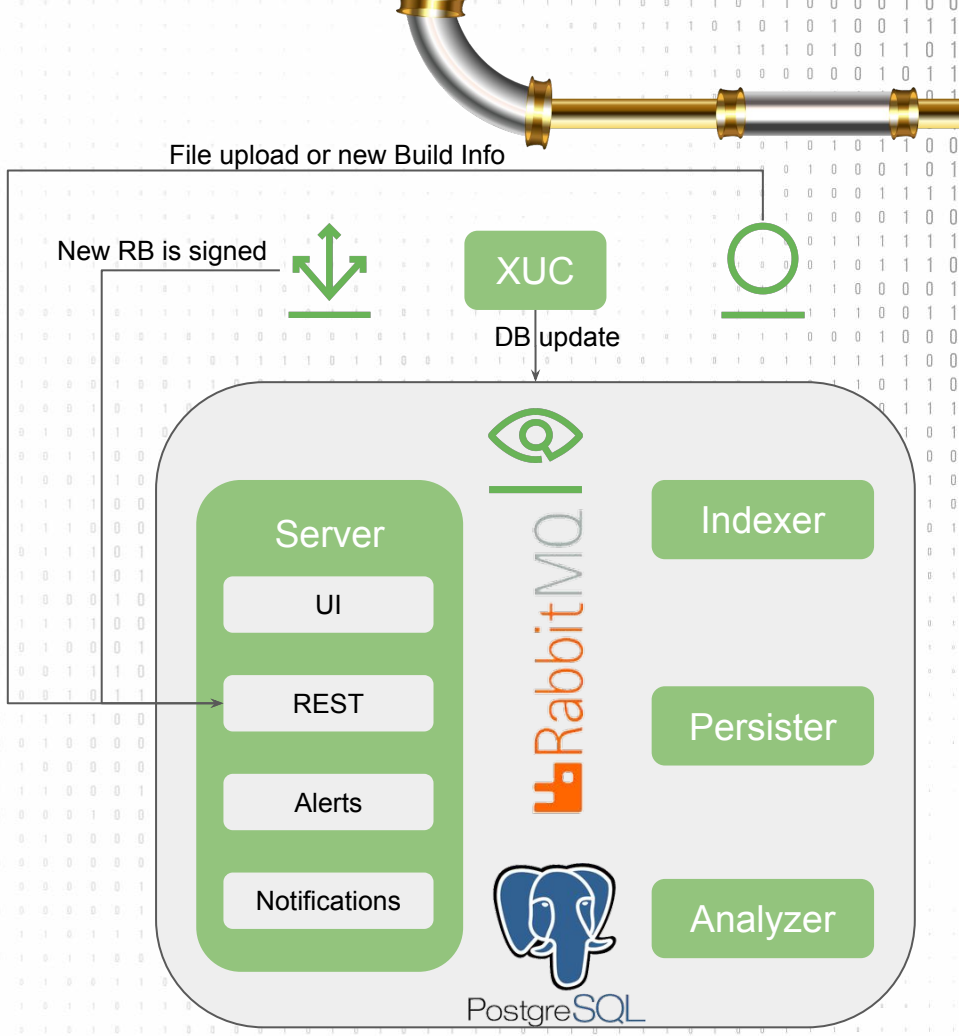
- The initial message (index or impact) is placed on MQ
- If index -
  - Indexer will fetch the artifact and prepare the graph
  - Persister will store it in DB
  - Analyzer scan for issues
- If impact - Analyzer will calculate impact
- Alerts service will generate Violations
- Notifier will trigger Actions



Xray Flow

High Level -

- What will trigger indexing?
  - File Upload
  - Build Info publish
  - Release Bundle Signing
- What will trigger a scan?
  - Previous indexing
  - Manual triggering
- What will trigger a Impact Analysis?
  - XUC update

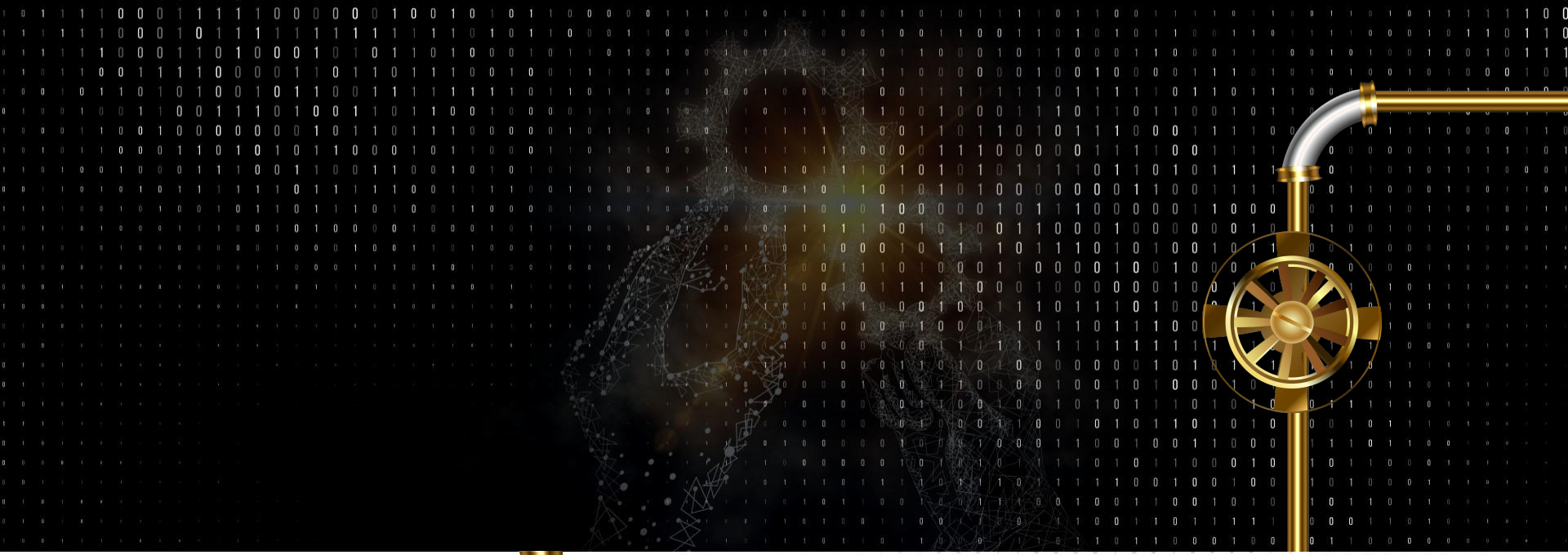


Xray Supported Technologies

Always changing.....

- Maven, Gradle, Ivy
- NPM
- PyPi
- NuGet
- RubyGems
- Docker
- Debian
- RPM
- Alpine
- SBT

https://www.jfrog.com/confluence/display/XRAY/Supported+Technologies



RULES & POLICIES

Xray rules

- Has a name
- Security rules
  - Specify severity OR CVSS Score range
- License rules
  - List of Allowed licenses OR Banned licences
- Attached to a specific policy
- Defines the automatic actions

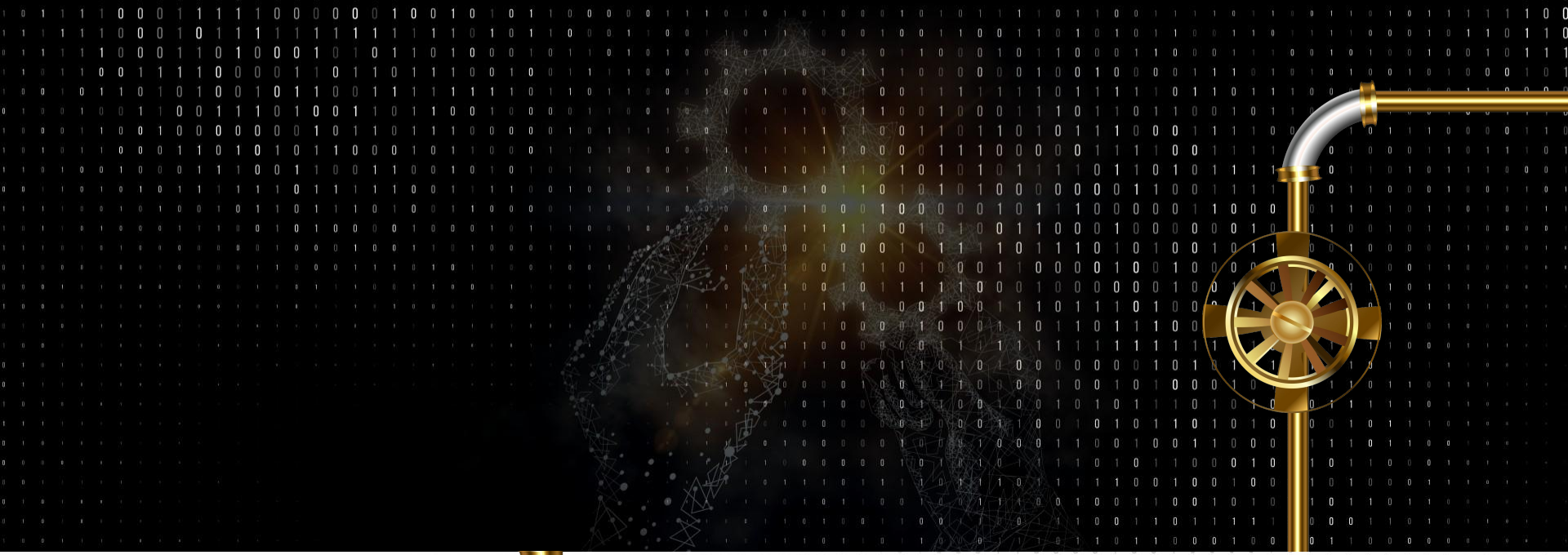
Xray policies

- Has a name and description
- Has a type - security or license
- 1:n rules attached to it
- Can be attached to multiple Watches
- Is contextless

  (defines “what” to enforce but not “where” to enforce it)

Xray Actions

- Defined on xray rules
- Generate violations
- Trigger a webhook (Jira, Slack, etc.)
- Email notification
- Block download (including unscanned)
- Fail build



WATCHES AND VIOLATIONS

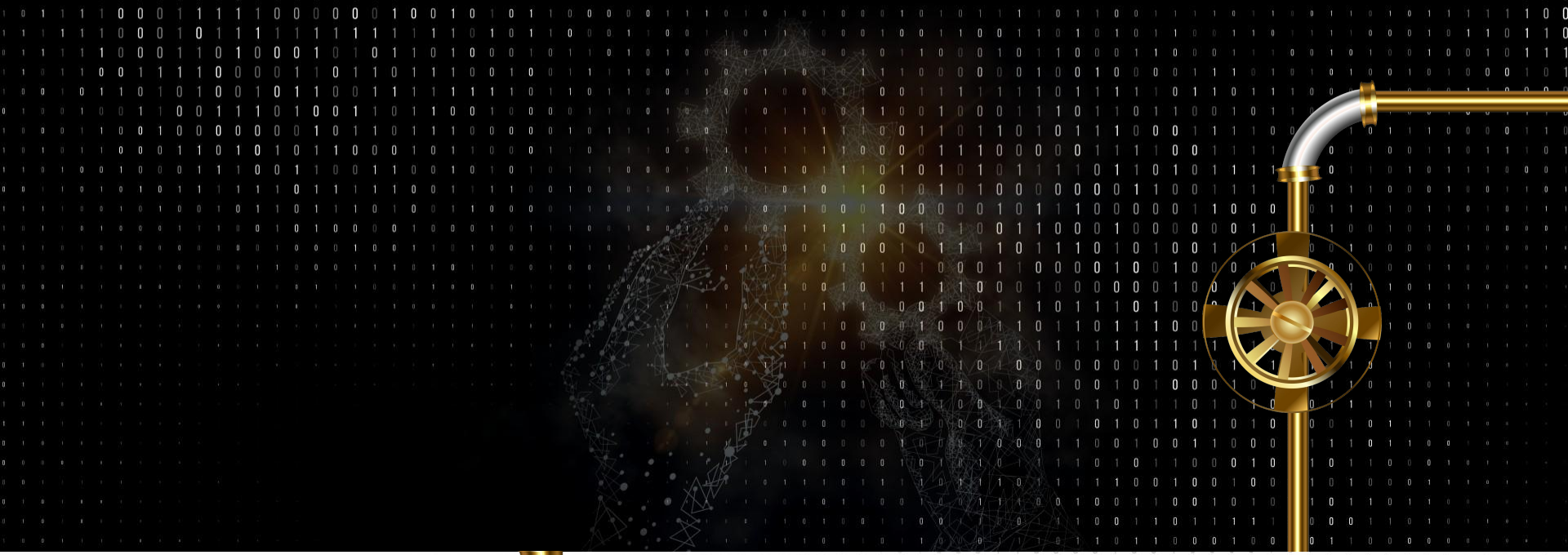
Xray Watches

- Has a name and description
- Enabled or Disabled
- Holds one or more Resources
- Holds one or more Policies (m:n relation)
- Shows the list of generated violations
- This list can be filtered
- Shows permanent ignore rules

Xray Violations

- The product of policies and watches

  (Based on their resources and rules)
- Severity
  - For security rules - based on the issue
  - For licences - based on rule configuration
  - For custom issues - based on custom issue definition
- Can be marked for ignore



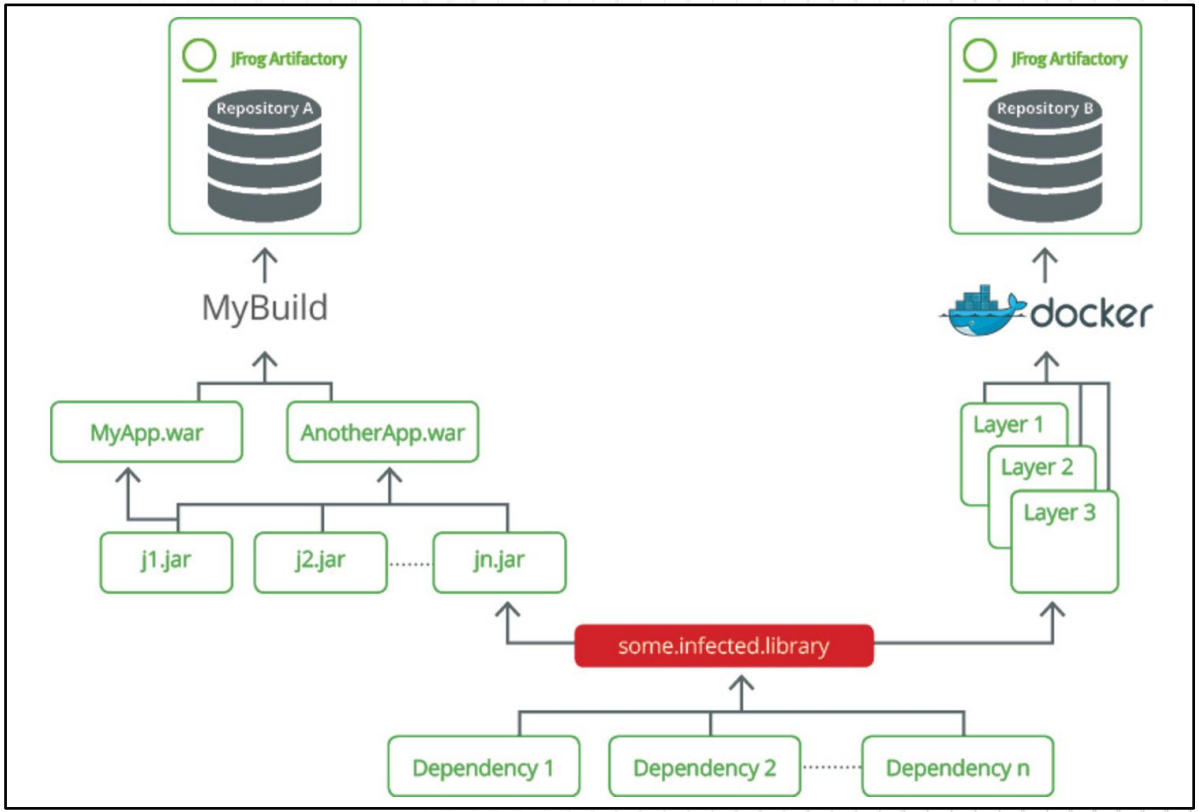
COMPONENTS & VIEWS

Xray Components

- A general name for “Violation Targets”
  - Known Package’s files
  - Unknown package’s files
  - Builds
- Extensive filtered Search
- Drill down
  - Violations for each version
  - Security & License issues
  - Location within Artifactory
  - Descendants and Ancestors - impact analysis

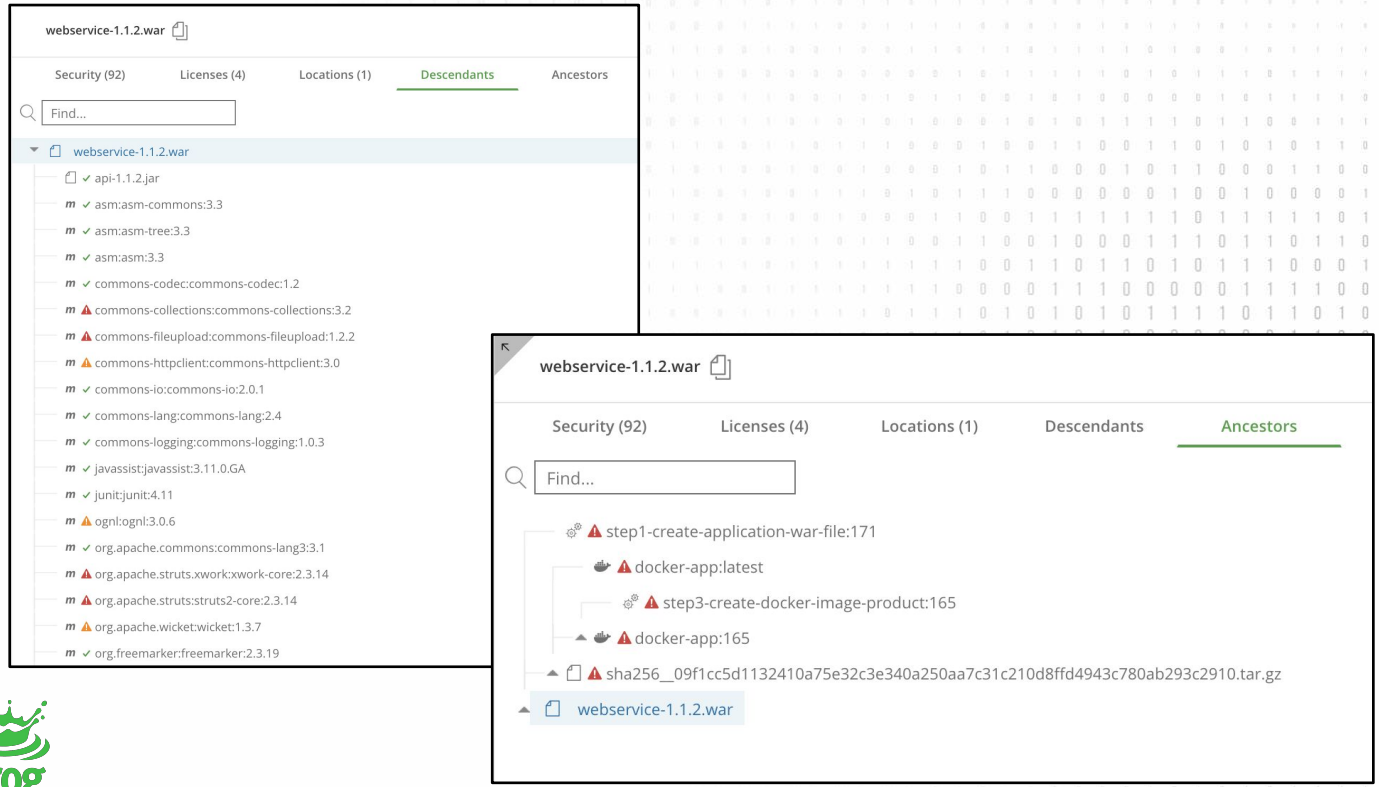
Graph metadata

The recursive nature of xray scanning



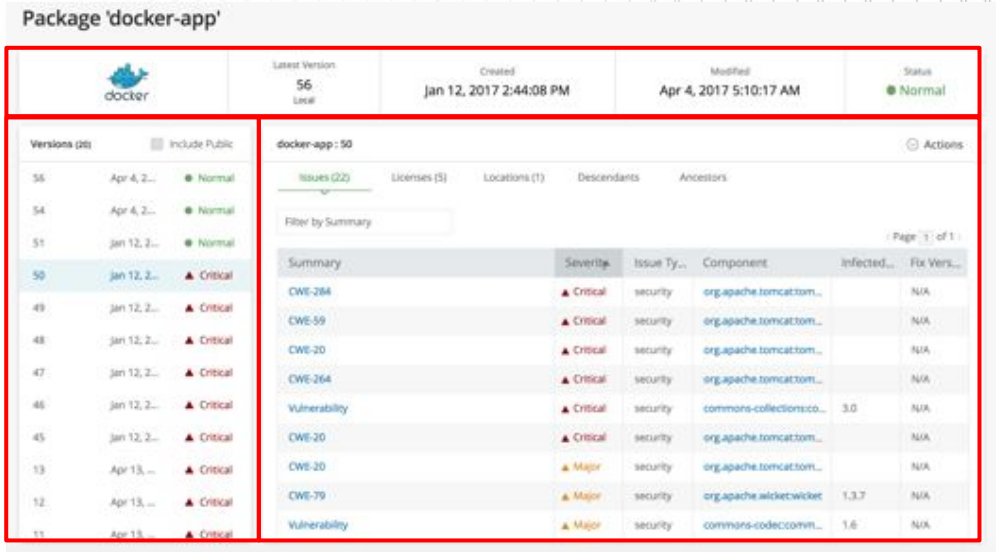
Graph metadata

The recursive nature of xray scanning



Xray Components - cont.

- Drill down view
    - General info - varies a little based of component
    - Versions of component
    - Details view

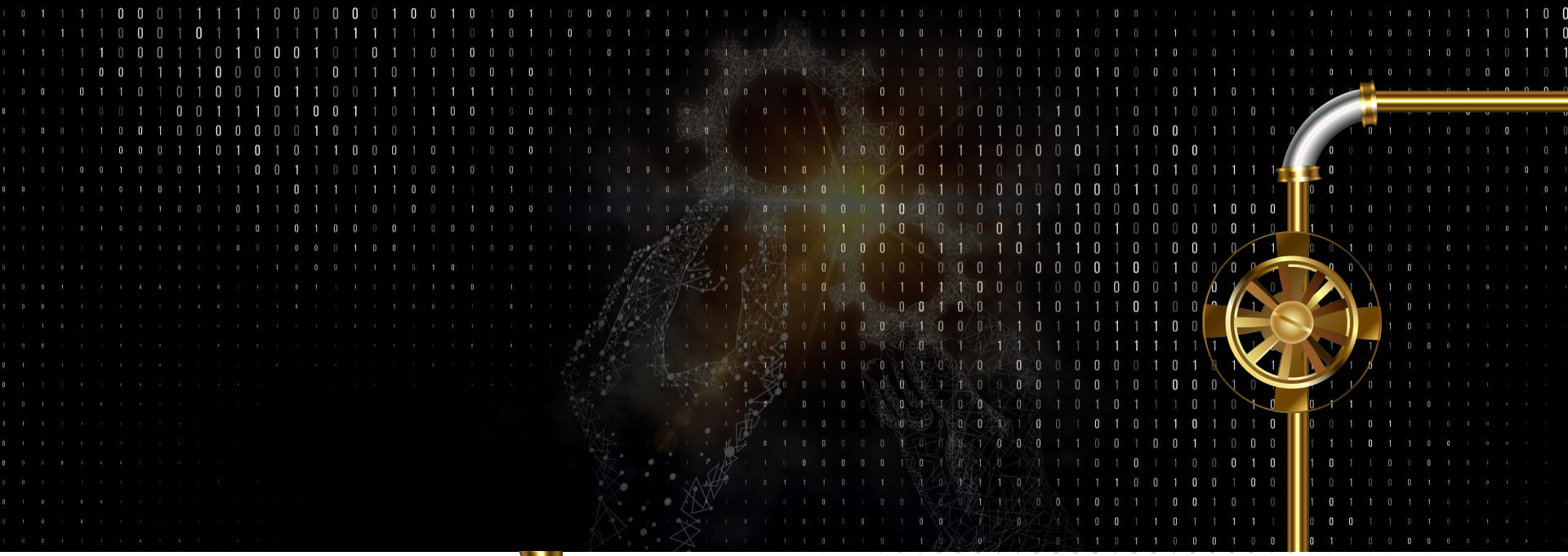


Xray Components - cont.

- Drill down view
  - General info - varies a little based of component
  - Versions
  - Details view
- More capabilities
  - Assign custom issue
  - Assign custom license
  - Export data

Reports

- License report
  - License type distribution across the organization
  - Validity break down based on all license policies
- Security report
  - General status of the system in several charts
- Some drill down capabilities
- Runs automatically every 4 hours in the background
- Reports are going to be re-born.....



Lab Exercise 2

Policies & Watches

LAB 2 - Policies & Watches

Security vulnerabilities

Congrats!! Following our course and after hearing about the recent breaches in the market, Your company has decided to embrace DevSecOps practices!!

YOU were chosen to design and implement the security and compliance policies.

- Make sure developers can't download artifacts from Artifactory that have security vulnerabilities of medium or higher severity.
- Try to download an artifact with a high severity security vulnerability. Did it work? Why?
- Try to download an artifact with a low severity security vulnerability. Did it work? Why?

LAB 2 - Policies & Watches

License Compliance

After a few legal issues with OSS software, the legal team has decided to set a strict policy for only allowing the use of artifacts with the MIT license.

- Enforce the new policy by making sure that no one can use artifacts with licenses other than the MIT license.
- Try to download an artifact that has a license other than the MIT license. Did it work? Why?
- Try to download an artifact that has the MIT license? Did it work? Why?
- Try to download an artifact that has no discovered license. Did it work? Why?

LAB 2 - Policies & Watches

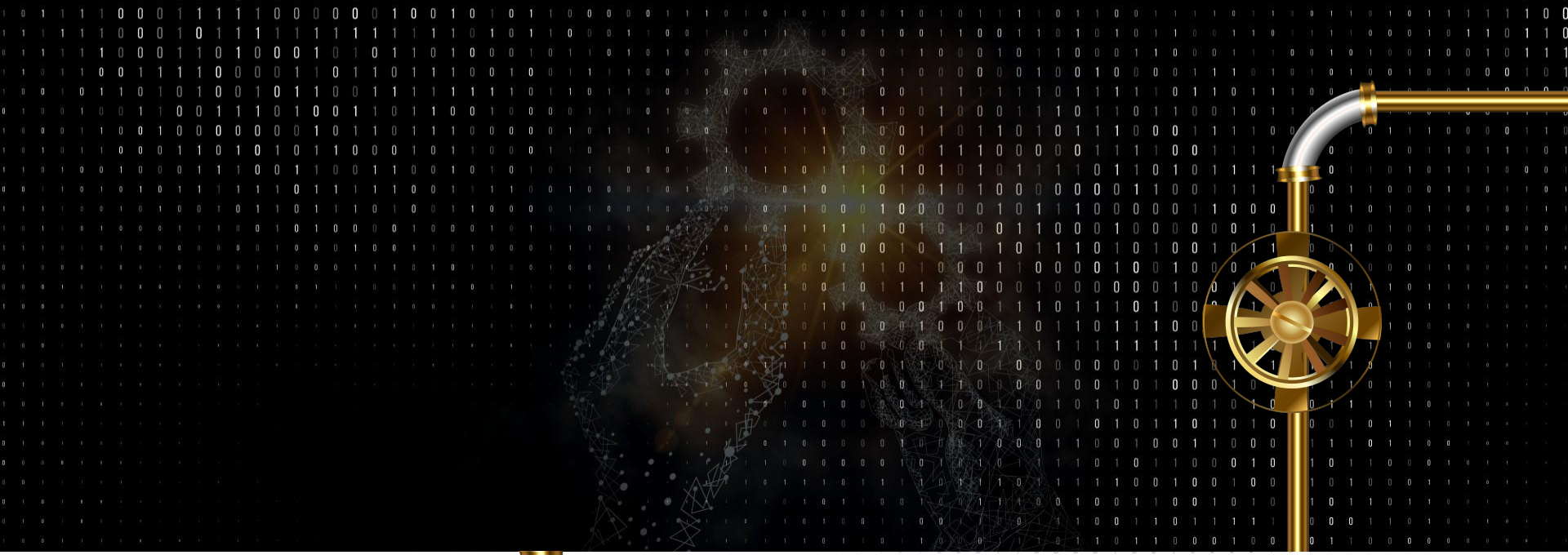
Custom issues

After intensive debugging and troubleshooting, the QA team found a performance issue with a specific version of one of the widely used packages in Artifactory.

- Make sure no one can use this version of the package (Hint: Use custom issues)
- Add information about why you have decided to block the of use this package.

Your company wants to mark all internal software packages with a new proprietary type of license due to legal concerns.

- Assign that license to a proprietary artifact. (Hint: Create the new license first)
- Since this is a proprietary license, the legal team are fine with developers using this license. Allow the use of this new license in addition to the MIT license.



USE CASES & INTEGRATIONS

General

Means of Integrations

- Internal capabilities

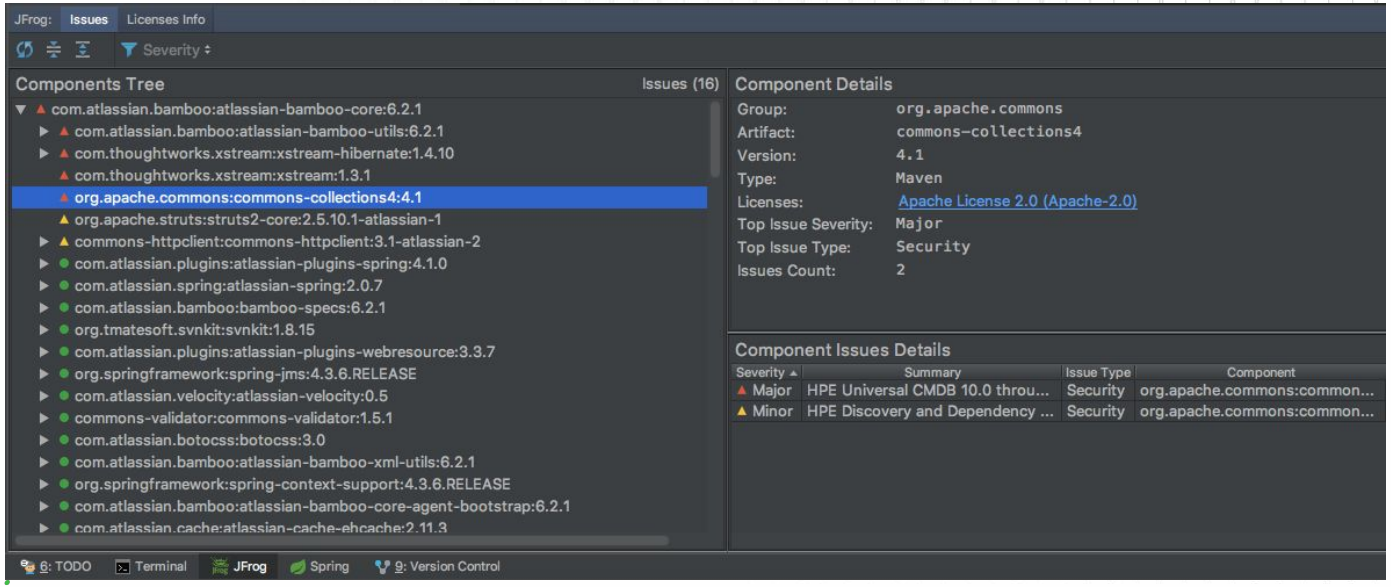
  (like Distribution -> Xray connectivity)
- Existing plugin's support

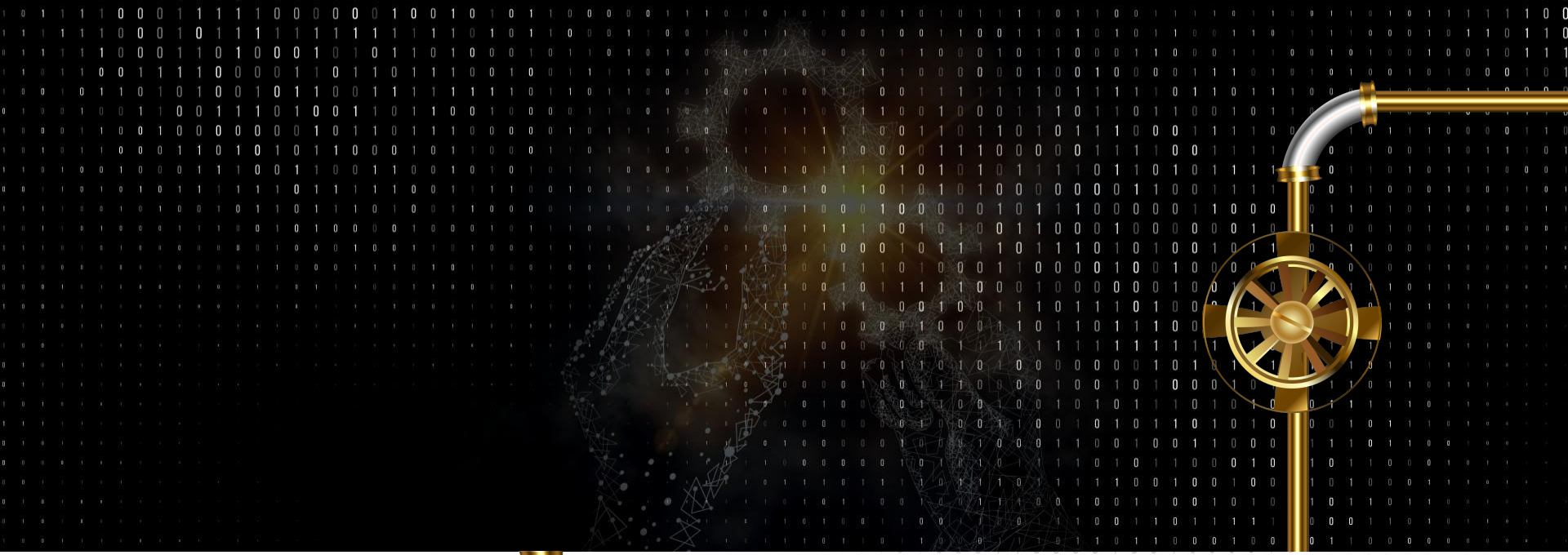
  (like the scanConfig object and the xrayScan API)
- REST API

  (As you can see here)
- JFrog CLI

IDE Integration

Available for IntelliJ, Eclipse, Visual Studio





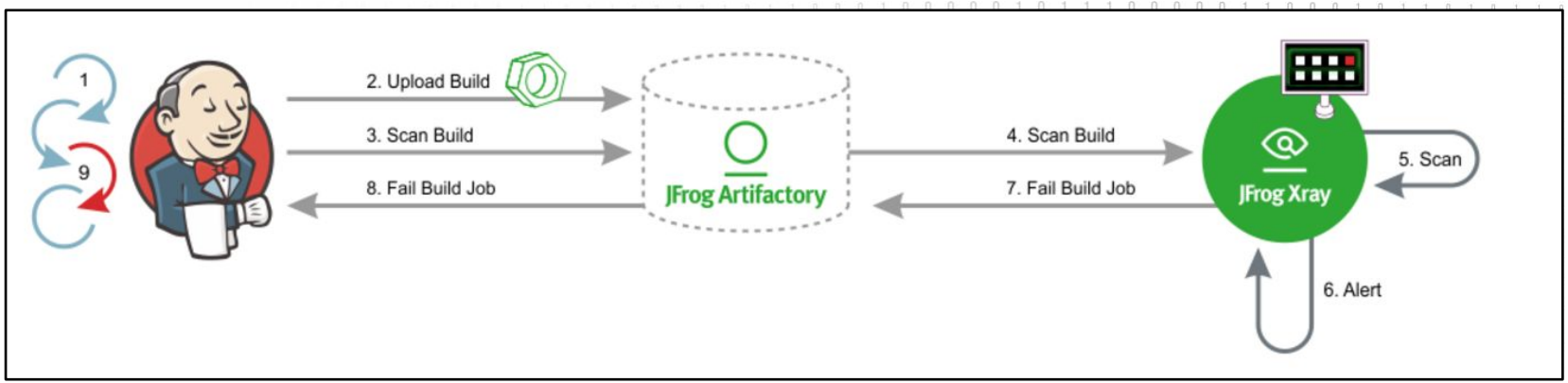
Demo - IDE integration

You can’t have any vulnerabilities in your builds

CI/CD Integration

Using plugins, Xray REST API or JFrog CLI

- Integrate into an organization's CI/CD pipeline
- Ensure the build jobs are stopped early in the process if they violate your assigned policies



Configuration

- Xray – Xray supports CI/CD integration from version 1.6
- Configure a Watch with the right filters that specify which artifacts and vulnerabilities should trigger an alert
- Set a Fail Build Job Action for that Watch
- Jenkins – Xray CI/CD integration is supported with Jenkins Artifactory Plugin v2.9.0 and above
- Create a scanConfig instance and pass it to the xrayScan method in the Jenkins Pipeline

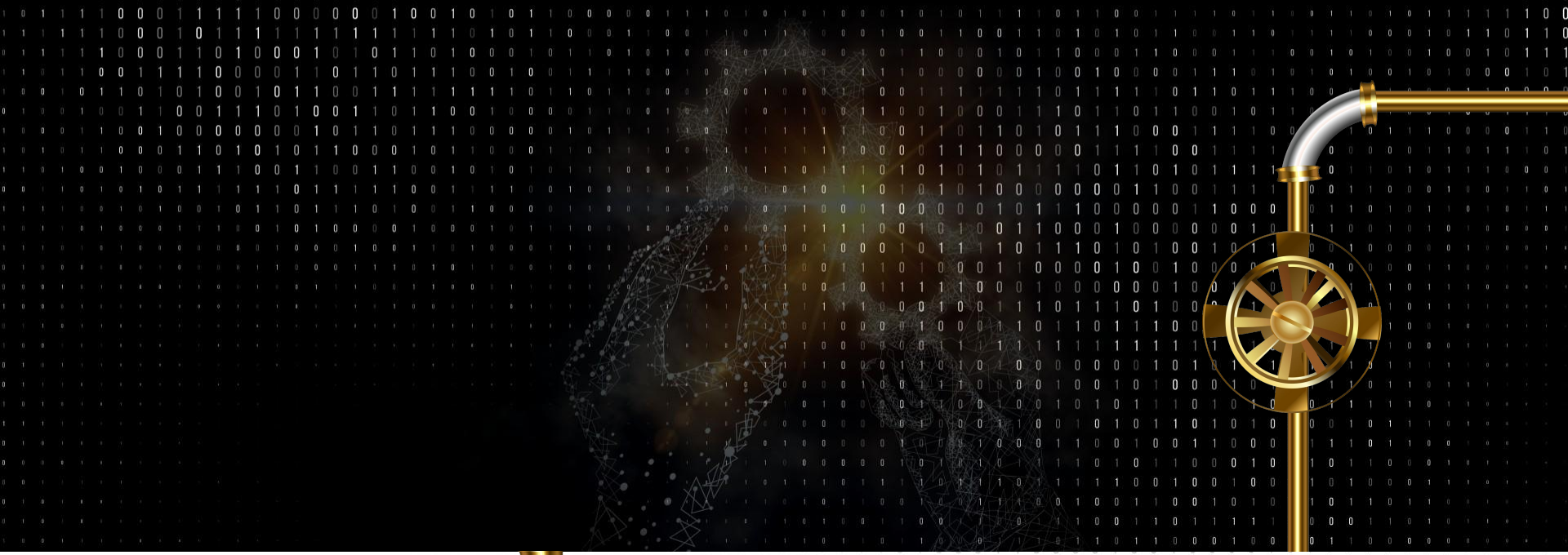
Jenkinsfile

```
node('promote-template') {
    stage('Xray') {
        if (XRAY_SCAN == "YES") {
            java.util.LinkedHashMap<java.lang.String, java.lang.Boolean> xrayConfig = [
                    'buildName' : env.JOB_NAME,
                    'buildNumber' : env.BUILD_NUMBER,
                    'failBuild' : false
            ]
            def xrayResults = server.xrayScan xrayConfig

            if (xrayResults.isFoundVulnerable()) {
                error('Stopping early… got Xray issues ')
            }
        } else {
            println "No Xray scan performed. To enable set XRAY_SCAN = YES"
        }
    }
```

Jenkinsfile - another way

```
stage('Xray Scan'){
    when {
        expression { return params.XRAY_SCAN }
    }
    steps {
        script {
            xrayConfig = [
                'buildName'      : buildInfo.name,
                'buildNumber'    : buildInfo.number,
                'failBuild'      : "${params.FAIL_BUILD}".toBoolean()
            ]
            xrayResults = rtServer.xrayScan xrayConfig
            echo xrayResults as String
        }
    }
}
```



Lab Exercise 3

CI/CD Integration

LAB 3 - CI/CD Integration

The Frog and the Butler - A Match Made in Heaven

Good job! Developers have stopped using infected artifacts due to this new setup, however vulnerable builds were still being promoted to production.

- Avoid breach and liability risk by making sure the vulnerable builds can't make it to production.
- Locate the jenkins URL (use kubectl get svc)
- Trigger the gradle-app-demo and the npm-app-demo builds and wait until they have finished successfully.
- Trigger the docker-app-demo build. Make sure you set the value of the XRAY_SCAN parameter to YES
- Did the build pass? Why?
- Look for the docker-app-demo build in the Xray UI. Can you see what is/are the infected component(s)?
- How can we fix these security vulnerabilities?

LAB 3 - CI/CD Integration

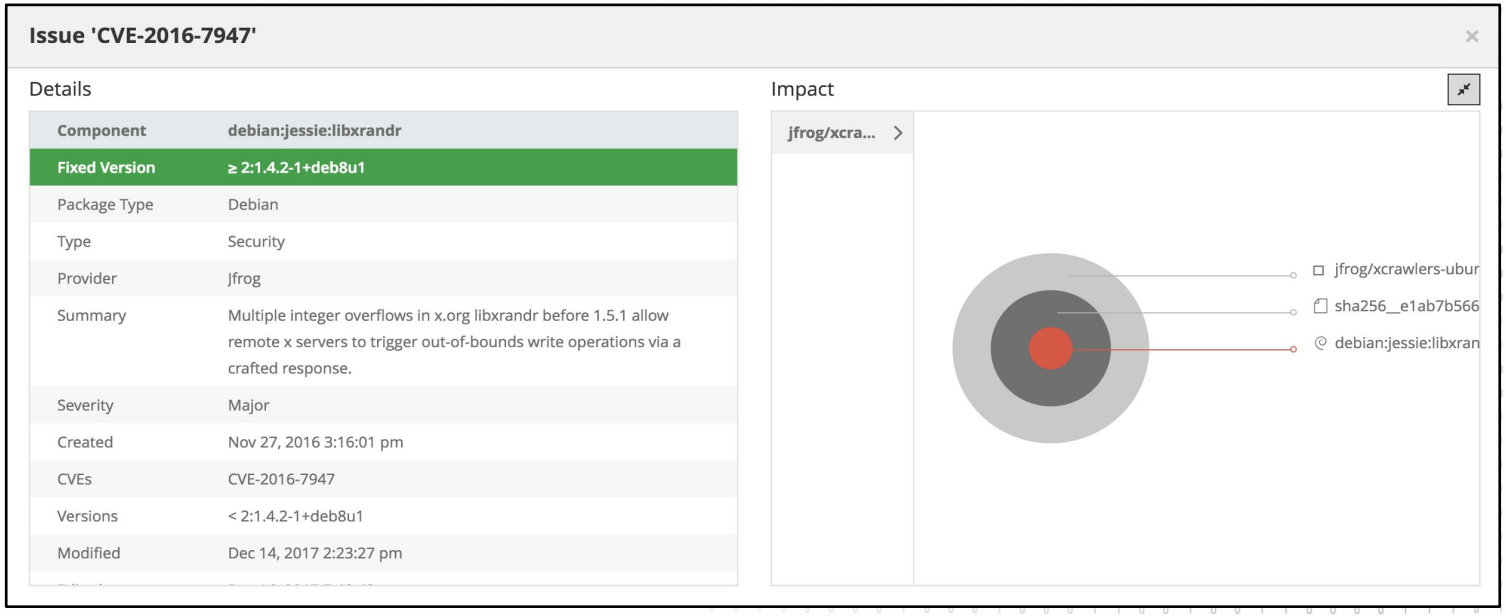
Optional - if time allows

Your company is starting to look into using CI servers other than Jenkins, like CircleCI. As you know, CircleCI is a non-pluggable CI server.

- How can we integrate Xray into our CI/CD pipelines with CircleCI? 🤔🤔🤔
- Scan the gradle-app-demo build using the JFrog CLI
- Look at the gradle-app-demo build in the Xray UI. Are there any vulnerabilities? Try to locate the infected component(s).
- How can we get rid of those vulnerabilities?

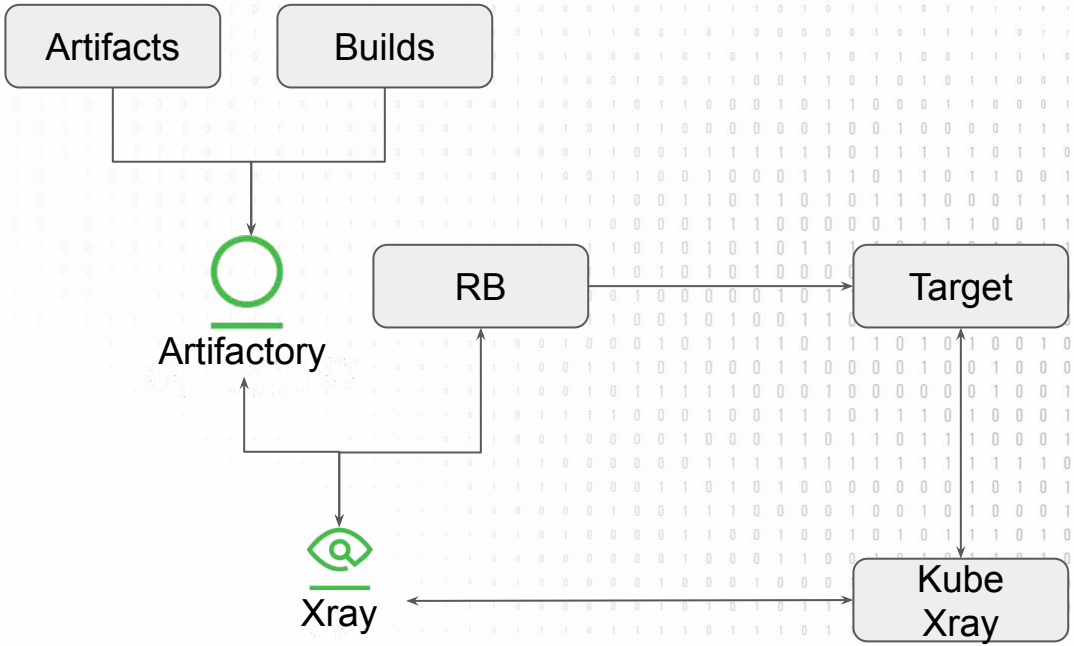
Deploy/Runtime Integration

Impact analysis on deployed/running components



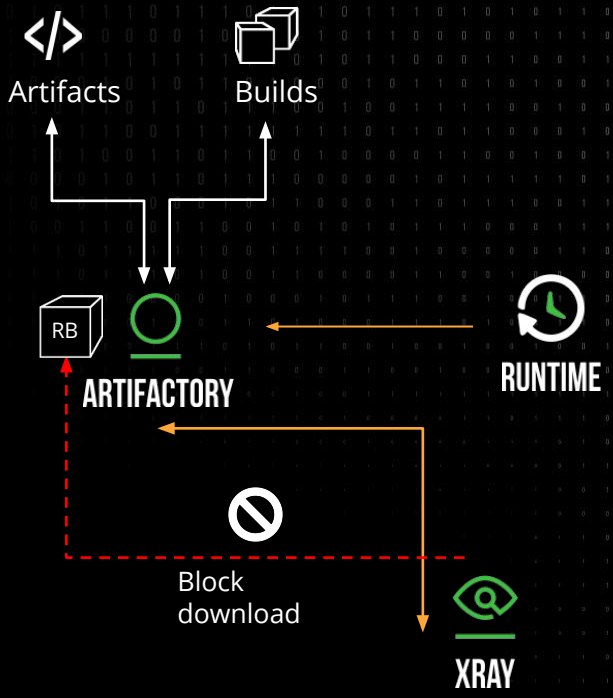
Deploy/Runtime Integration

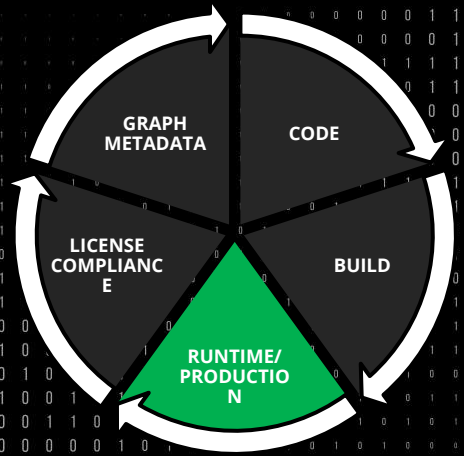
Based on Release bundle support or Kube Xray



Xray Use Cases

3A. Xray and Runtime/Production binary scanning

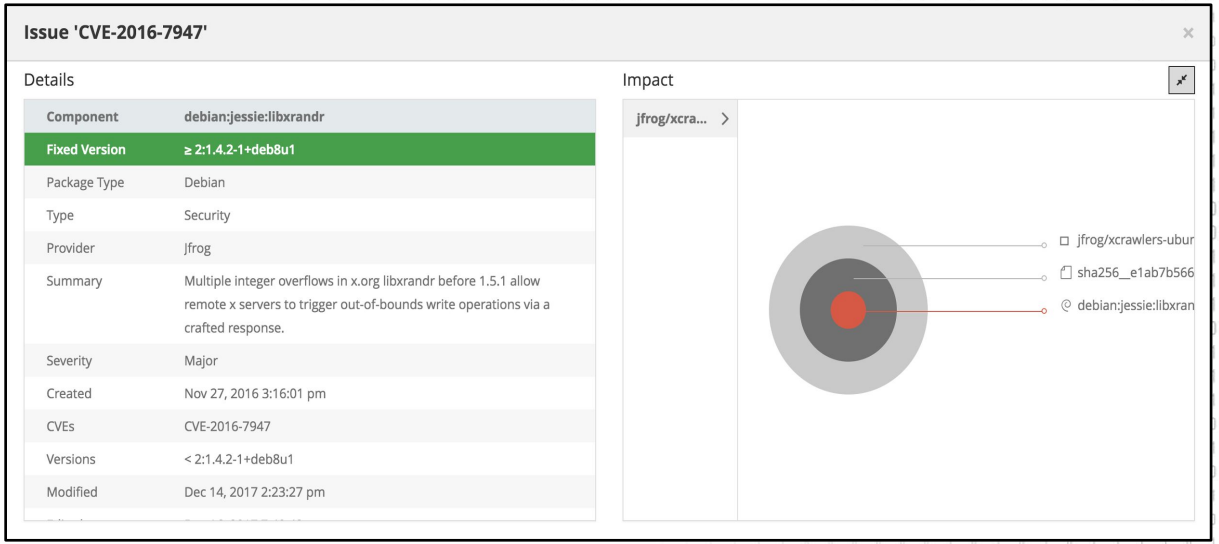


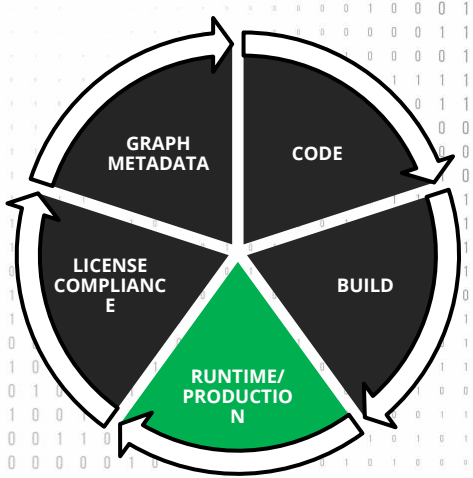


RB Release Binary

Xray Use Cases

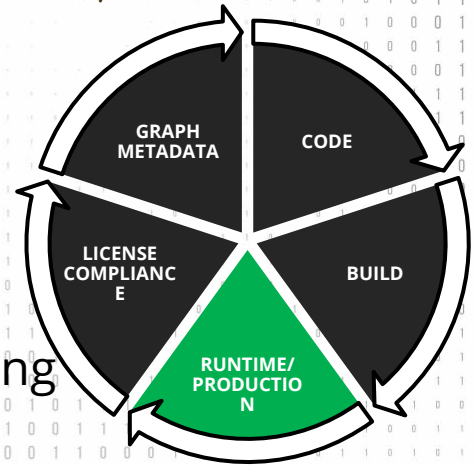
3B. Runtime / Production binary scanning





Xray Use Cases

3C. Runtime / Production binary scanning

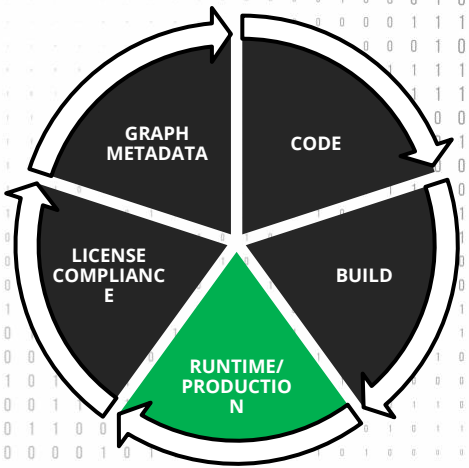


- KubeXray
  - ✔ An open source software project
  - ✔ Extends security of Xray to the applications running (or about to run) in Kubernetes pods.
  - ✔ KubeXray can enforce policies on what has already been deployed using the metadata that Xray generates by scanning container images.
  - ✔ KubeXray monitors events from the Kubernetes server and from Xray, and enforces current security policy for all of the pods Kubernetes runs

Xray Use Cases

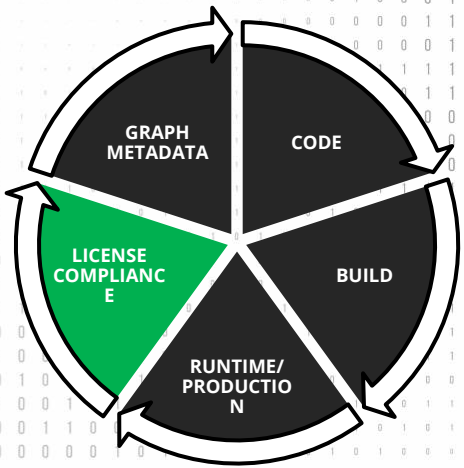
3C. Runtime / Production binary scanning

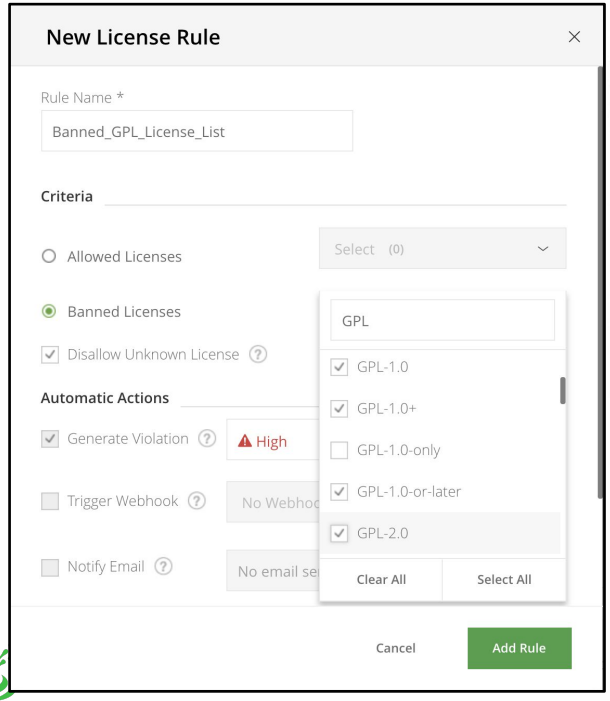
- KubeXray
  - ✔ KubeXray listens to event streams
    - Deployment of a new service
    - Upgrade of an existing service
    - A new license policy
    - A new security issue
  - ✔ KubeXray policy action - set in values.yaml file
    - Scaledown to 0
    - Delete
    - Ignore
  - ✔ Configure policy actions for the following conditions
    - Unscanned
    - Security
    - License

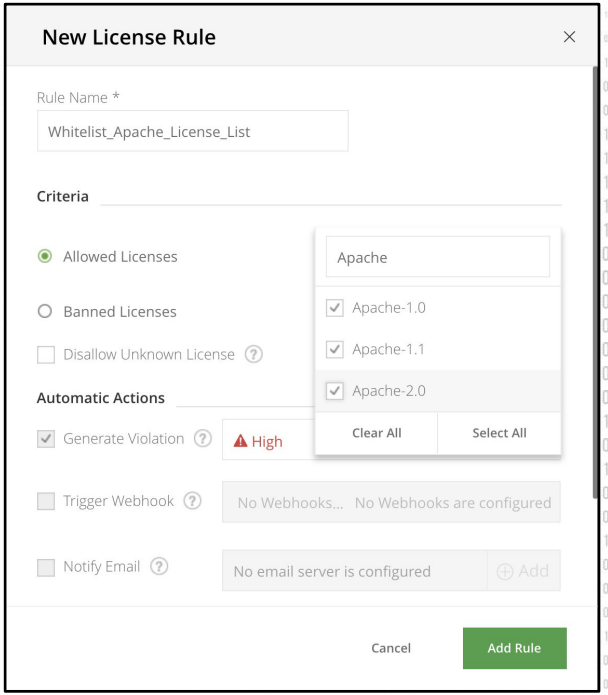


Xray Use Cases

4. License Compliance

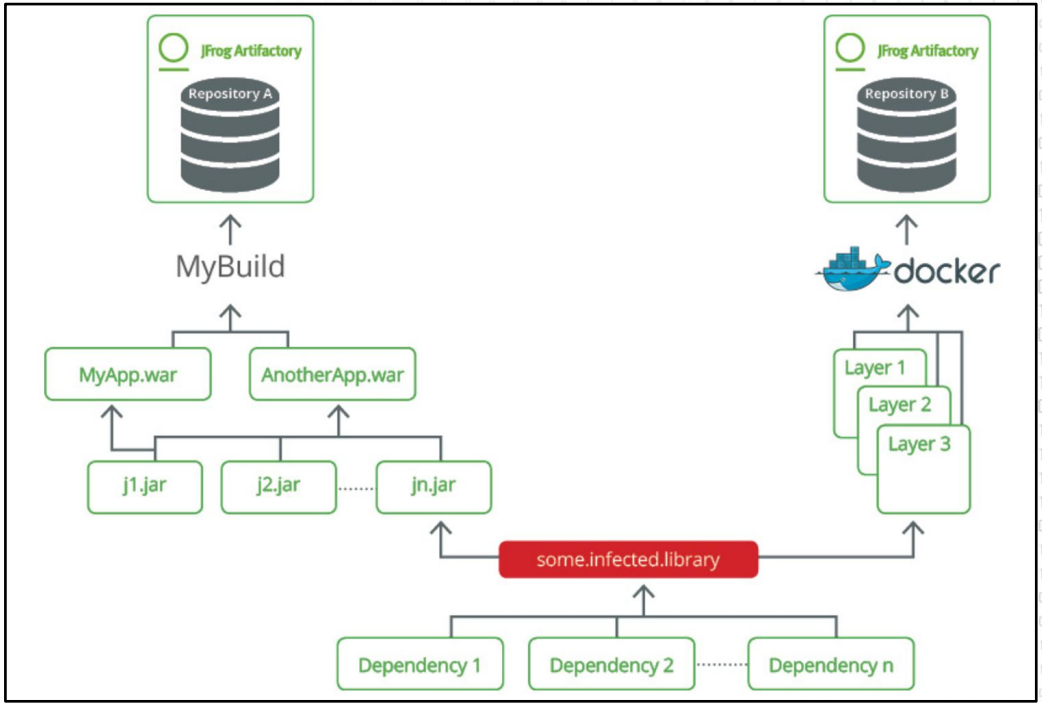


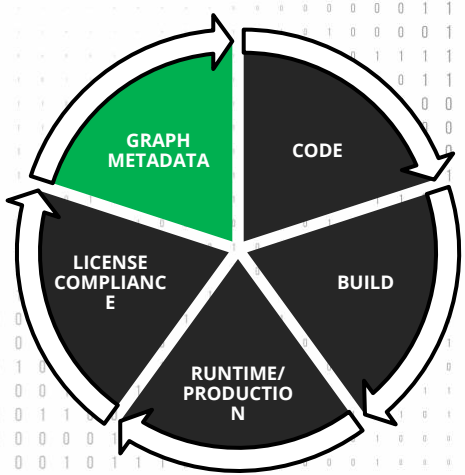




Xray Use Cases

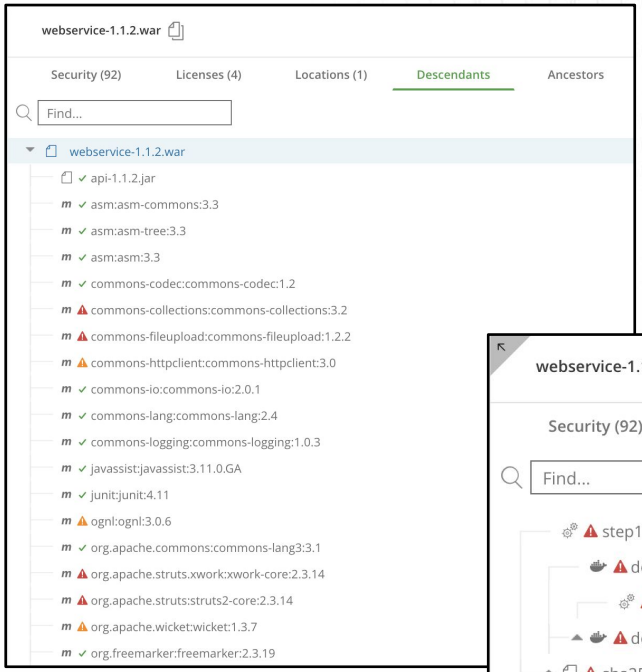
5. Graph Metadata

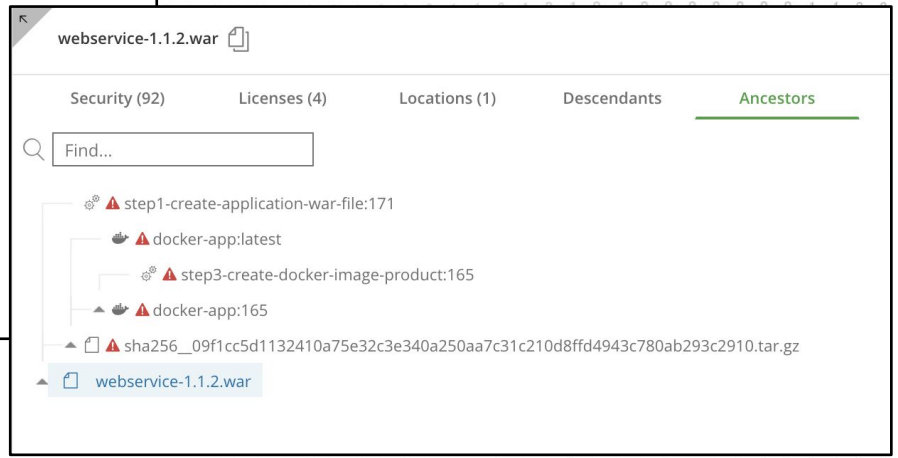


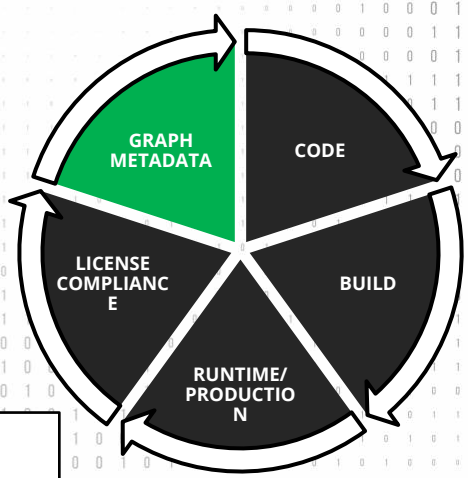


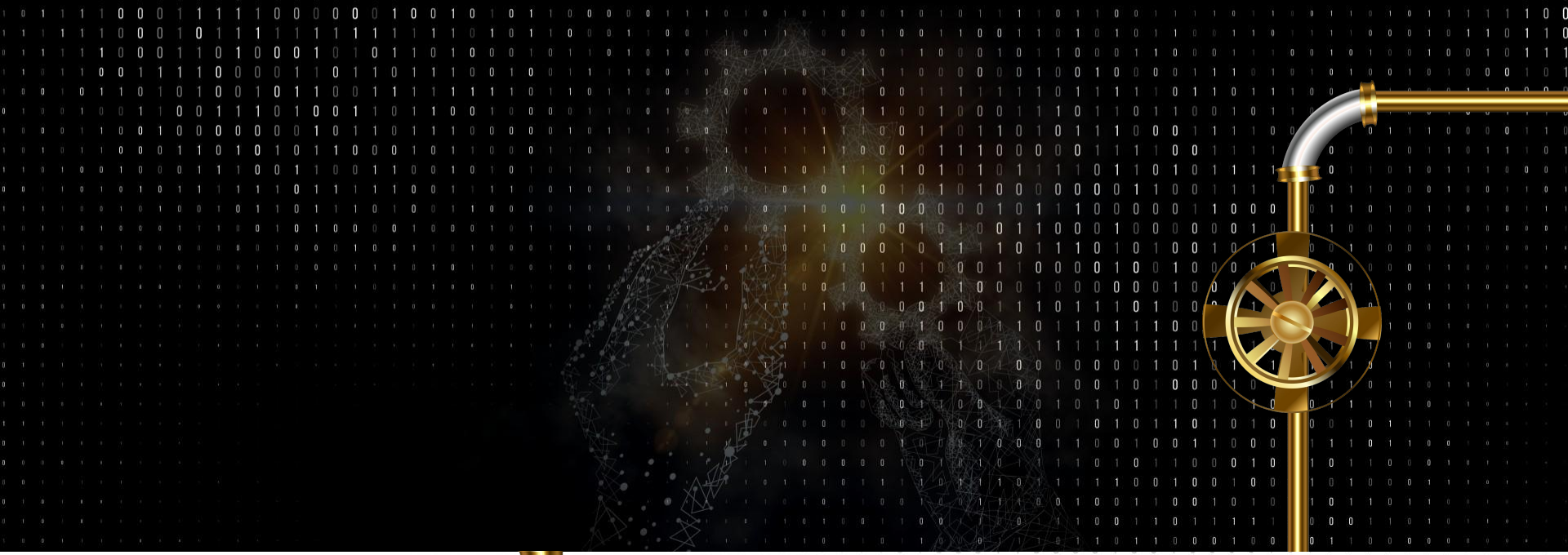
Xray Use Cases

5. Graph Metadata









Lab Exercise 4

Xray Dependency Graphs

LAB 4 - Xray Dependency Graph

Impact analysis

Woot Woot! 🚀 The application we built by running the docker-app-demo build job was deployed to production! That was quick ha? That's DevOps for you 😉

Oh wait, but not everything is hunky dory, a new vulnerability was found in the **jackson-databind** package.

- Find out which components are affected by the newly discovered vulnerability.
- How can know which of the infected components have made it to production? (Hint: Examine the Locations tab)
- What can we do with this information? How can we fix it?

In the meantime, the legal team has reached out to you saying they are considering to ban the use of the CDDL-1.0 license.

- Tell the legal team which components use the CDDL-1.0 license
- If the legal team indeed decides to ban the use of that license, how can you enforce it?

Xray For You

JFrog Xray is offered on-prem or in the cloud as a SaaS offering

On-prem

Self-managed.

Install, Manage, Maintain on a hardware or host in the cloud itself

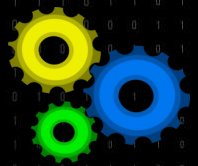
Cloud
(SaaS offering)

We manage, maintain and scale the infrastructure

Automated server backups with free updates and guaranteed uptime

Xray for Small Business

Single DC/Site



Centralized CI

Few Users/Developers

Self-hosted

Cloud/SaaS

JFrog ARTIFACTORY + JFrog XRAY

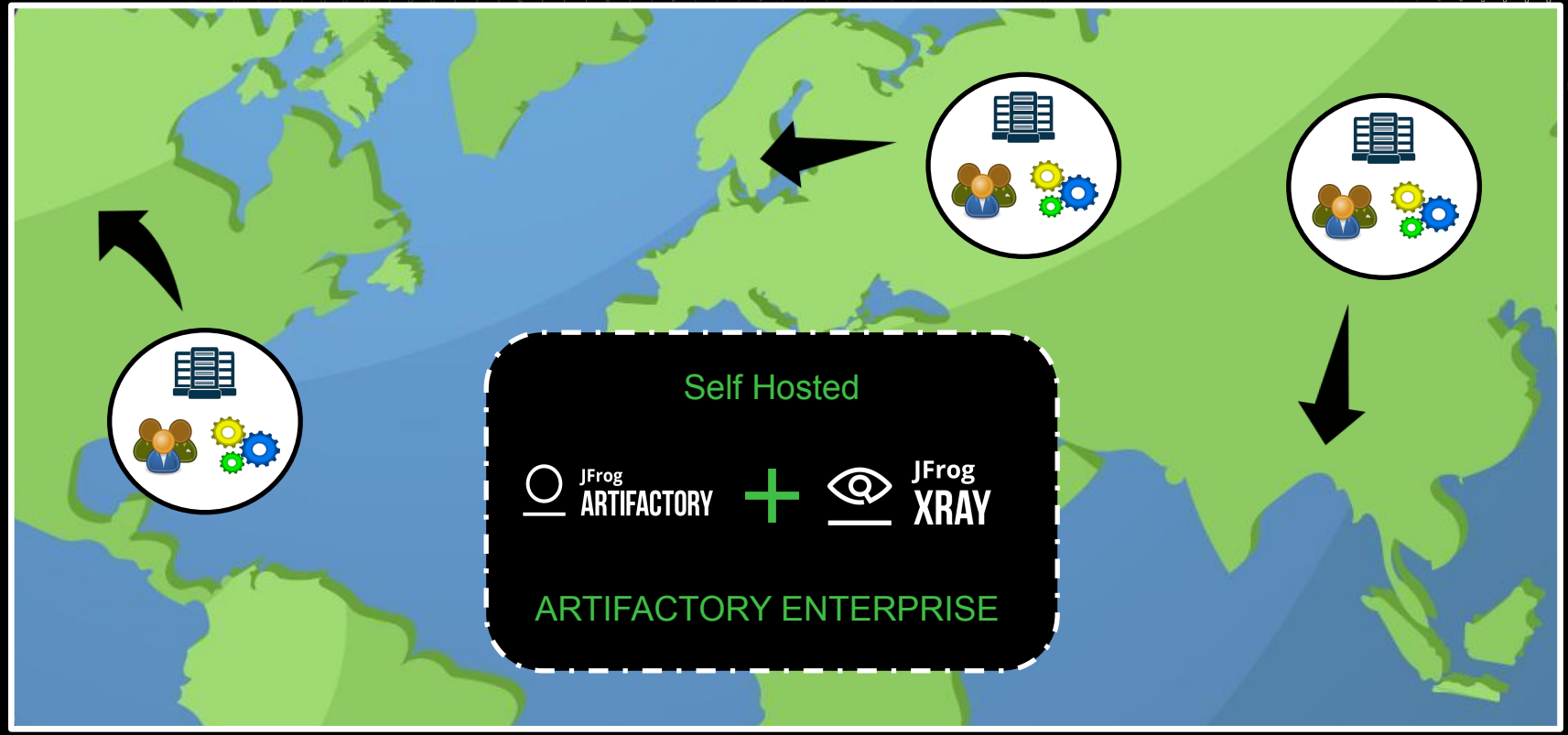
ARTIFACTORY PRO X

JFrog ARTIFACTORY + JFrog XRAY

ARTIFACTORY CLOUD PRO X

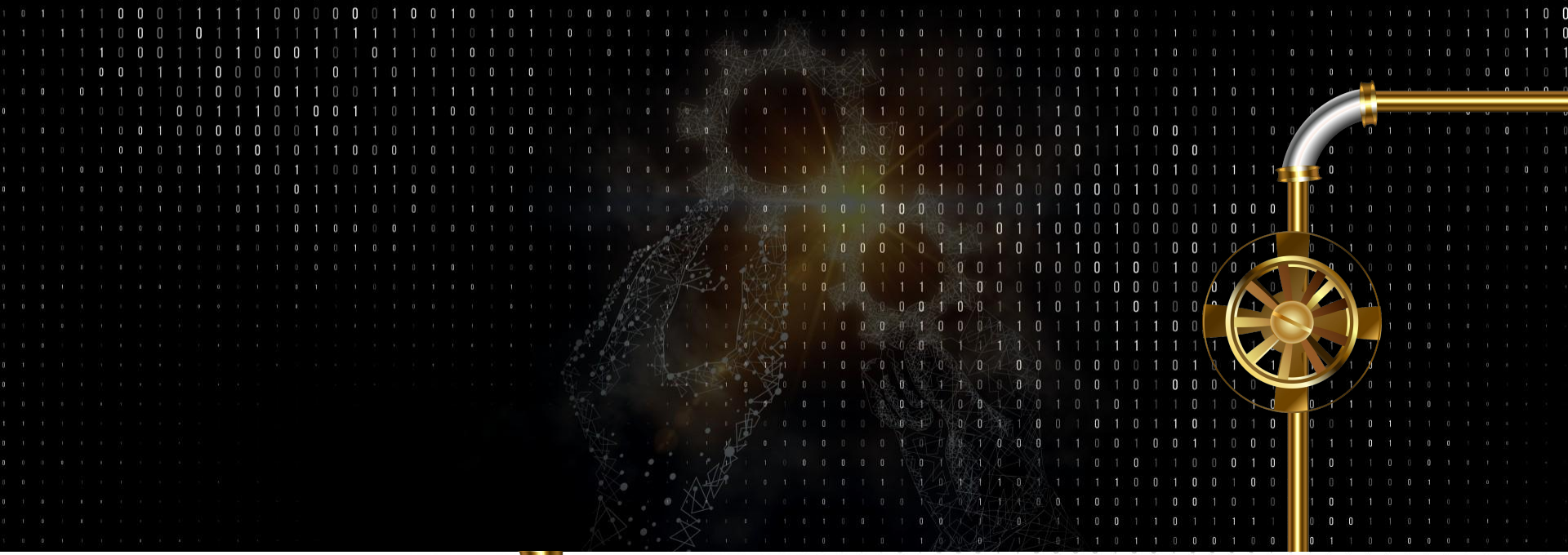
JFrog

Xray for Enterprise Business



The Future of DevSecOps

- No cutting corners when it comes to security.
- Accountability = Action
- DevSecOps Demand Comes from:
  - ✔ New technologies
  - ✔ Rising security breaches
  - ✔ Awareness about DevSecOps platforms
  - ✔ Need to improve SDLC
- Adding Security to DevOps is here!



ADMINISTRATION & CONFIGURATION

Xray Configuration

On-Boarding

- Base URL
- License
- Connection to Artifactory
- Selecting initial resources
- Database sync

Xray Configuration

Additional Configuration

- Managing users - internally or via Authentication provider
- Mail Server
- Configuring Web Hooks
- Managing indexed resources
- Number of workers per service
- License management
- HTTP and Proxy settings

Xray Administration

Monitoring

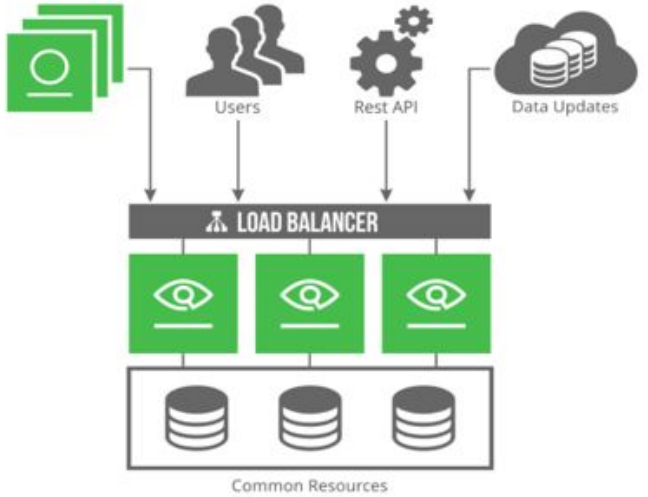
- System Logs
- System Messages
- System Monitoring
- Backup & Restore
- HA status
- Support Zone

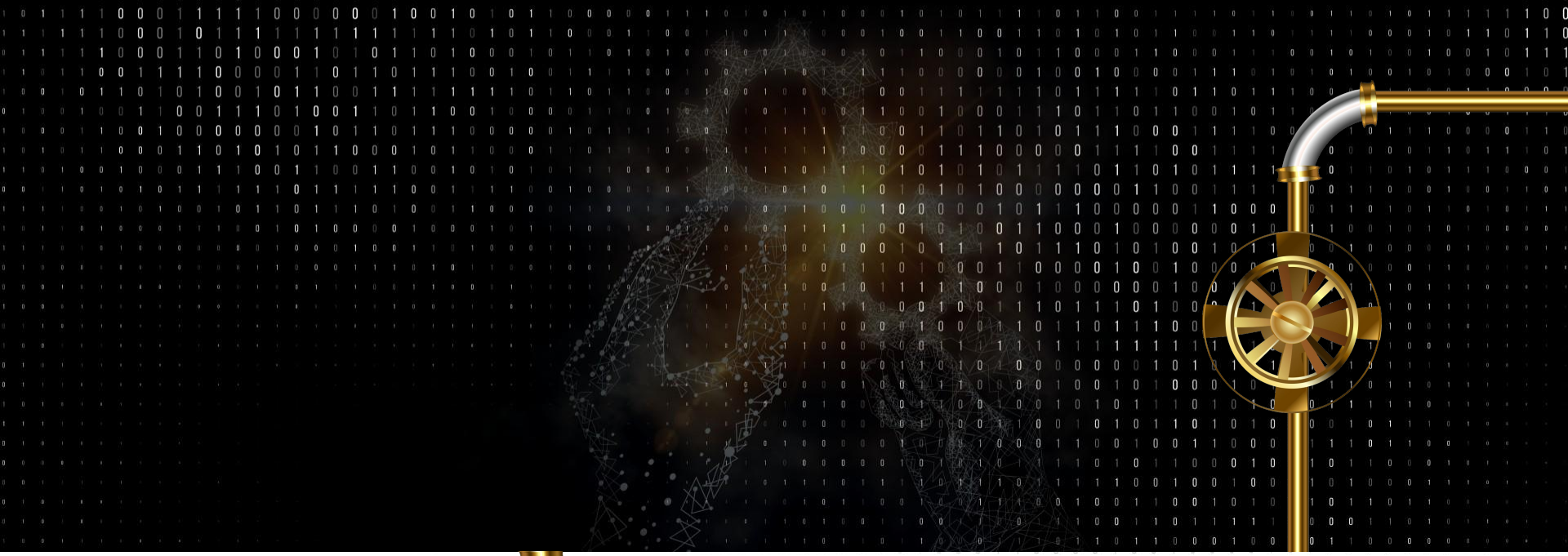
- https://www.jfrog.com/confluence/display/XRAY/System+Maintenance+and+Monitoring

Scaling Out

High Availability - Xray

- Each node runs 4 microservices: Indexer, Persist, Analyzer and Server
- Three Common Resources: MongoDB, PostgreSQL and RabbitMQ
- The storage used by Xray is **NOT** a common resource
  - Only node specific files, such as configuration and temporary files, are saved to the disk
- Xray synchronizes the following across nodes:
  - Critical and Temporary Data (using common DB)
  - Cached Objects (via RabbitMQ)
  - Scheduled Jobs (via RabbitMQ)





SUMMARY and Q&A

THANK YOU ALL