



Release Management with Artifactory

Yoav Landman & Frederic Simon



About us



★ Yoav Landman

- Creator of Artifactory, JFrog's CTO

★ Frederic Simon

- JFrog's Chief Architect

★ 10+ years experience in commercial enterprise build and development environments

★ Serving the community since 2006 with OSS tools (Artifactory, Jade Maven Plugins and AnnoMojo, Dependency Analyzer, Stellarium for Java, IDEA plugins...)



Agenda



- ★ **Intro**
- ★ **Configuration tips**
- ★ **Searches**
- ★ **Reproducible builds**
- ★ **Advanced features**
- ★ **Smart staging and promotion**
- ★ **Web Start and JavaFX**
- ★ **Gradle integration**



What is a repository manager?



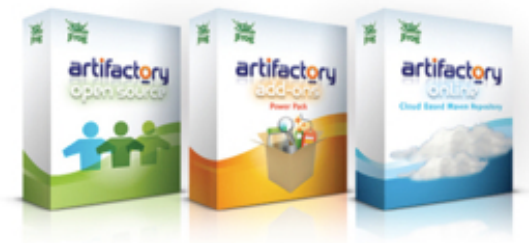
- ★ Artifacts storage and proxy
- ★ Avoid hitting public remote repositories
- ★ Inefficient, unreliable, content quality, non-secure...
- ★ Deploy, manage and share local artifacts
- ★ Full control over artifacts resolution and delivery



What is artifactory ?



- ★ Advanced binaries repository - the binaries' SCM
- ★ JavaFX and WebStart repository
- ★ Supports REST, Maven*, Gradle*, Ivy/Ant*, Buildr
* native support
- ★ First “real” web-driven repository manager (2006)
- ★ Upload through UI, indexed searches...
- ★ +80,000 downloads (02/2010)
- ★ OSS | Enterprise PowerPack | Cloud hosting



Configuration tips - demo



- ★ Remote repositories sharing via REST
 - Reuse configuration
- ★ Automatically generate the client configuration
 - Maven - settings.xml
 - Gradle plug-in
- ★ Centrally controlled encrypted password

```
<credentials host="localhost" realm="Artifactory Realm"  
  username="admin"  
  passwd="{DESede}OyxbjkRcS9JxKmi2Rm8RcA=="  
</>
```



DEMO TIME



Searches



★ Search types

- Quick search (wildcard part of path)
- GAVC
- Properties
- Class/Jar resource
 - ▶ See the actual class found!
 - ▶ View source + syntax highlighting
- XPath - also search inside POM, Ivy.xml content!



★ Grouping support

- E.g. group by repository, groupId etc.

★ Always possible to locate the results in the repo tree browser



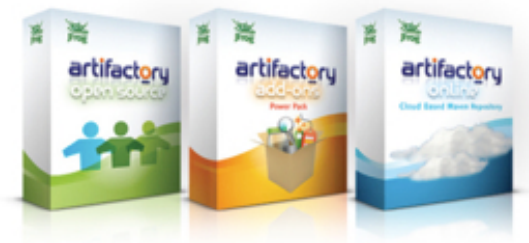
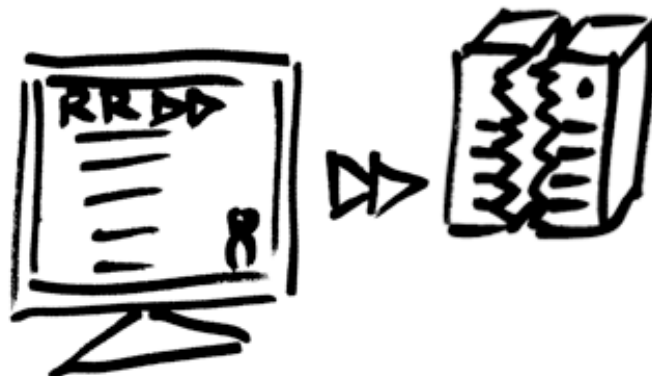
DEMO TIME



Reproducible Builds



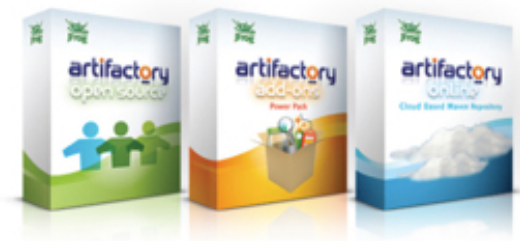
- ★ Sources have a reproducible context - compilation
 - Reproducible via SCM tagging.
- ★ Binaries also have a context - packaging and publishing
- ★ A lot of things are only resolved at build time:
 - Version ranges
 - Dynamic property values
 - Latest snapshot and release versions (latest.integration/latest.release)
 - ▶ For both Dependencies and Plugins



The Role of the CI Server



- ★ Captures all the information needed to reproduce the build
- ★ **Published module artifacts**
- ★ **Resolved dependencies of all scopes**
- ★ **General Build Environment**
 - JVM
 - Architecture
 - CI server version
 - General properties
 - Build statistics
 - User who executed the build
 - Etc.



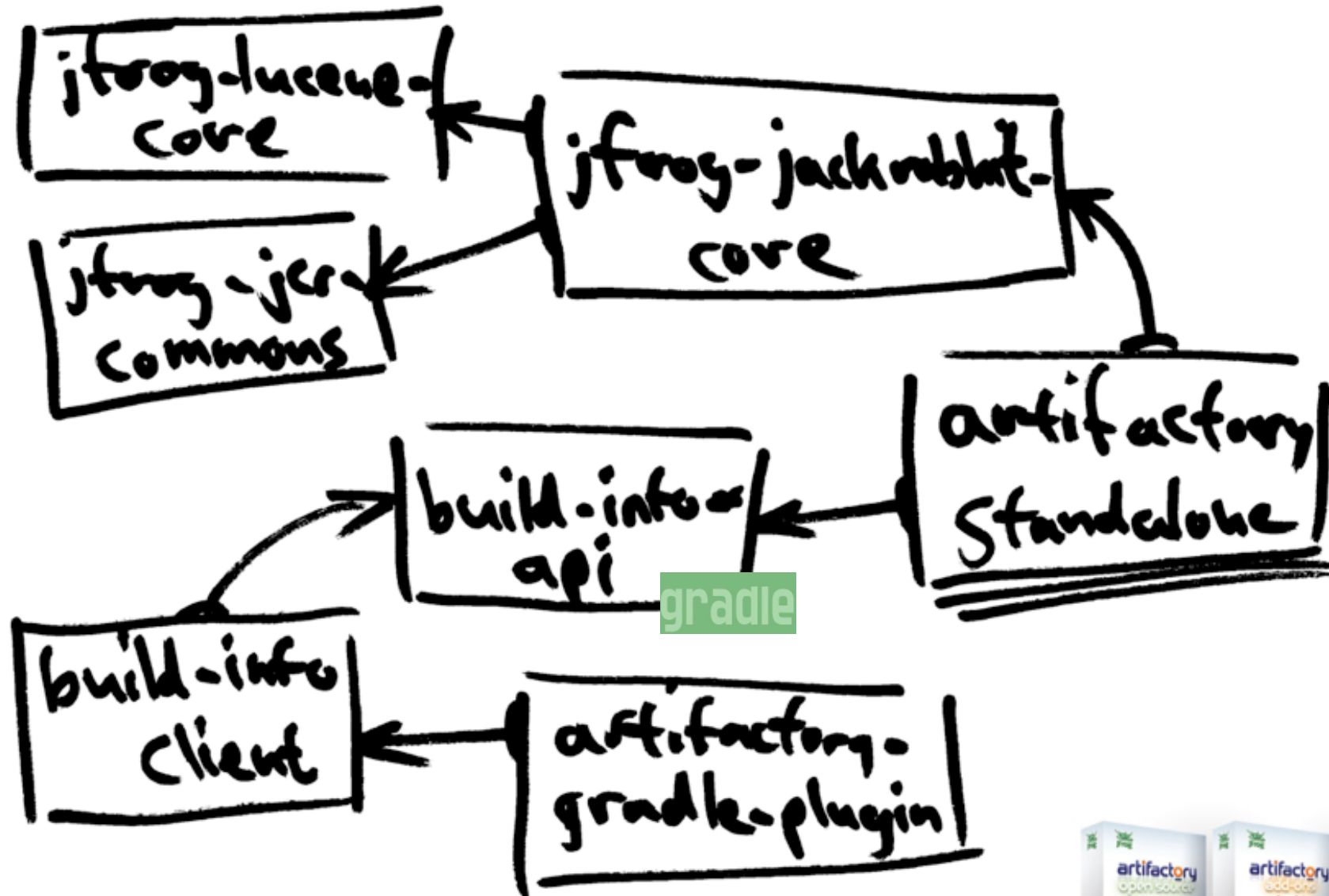
Artifactory and Hudson



- ★ Deploy from Hudson to Artifactory
- ★ Navigate the builds in Artifactory
- ★ Link back to Hudson builds
- ★ Relate artifacts to build
- ★ Export/promote/manipulate build artifacts
- ★ More efficient multi-module deployment



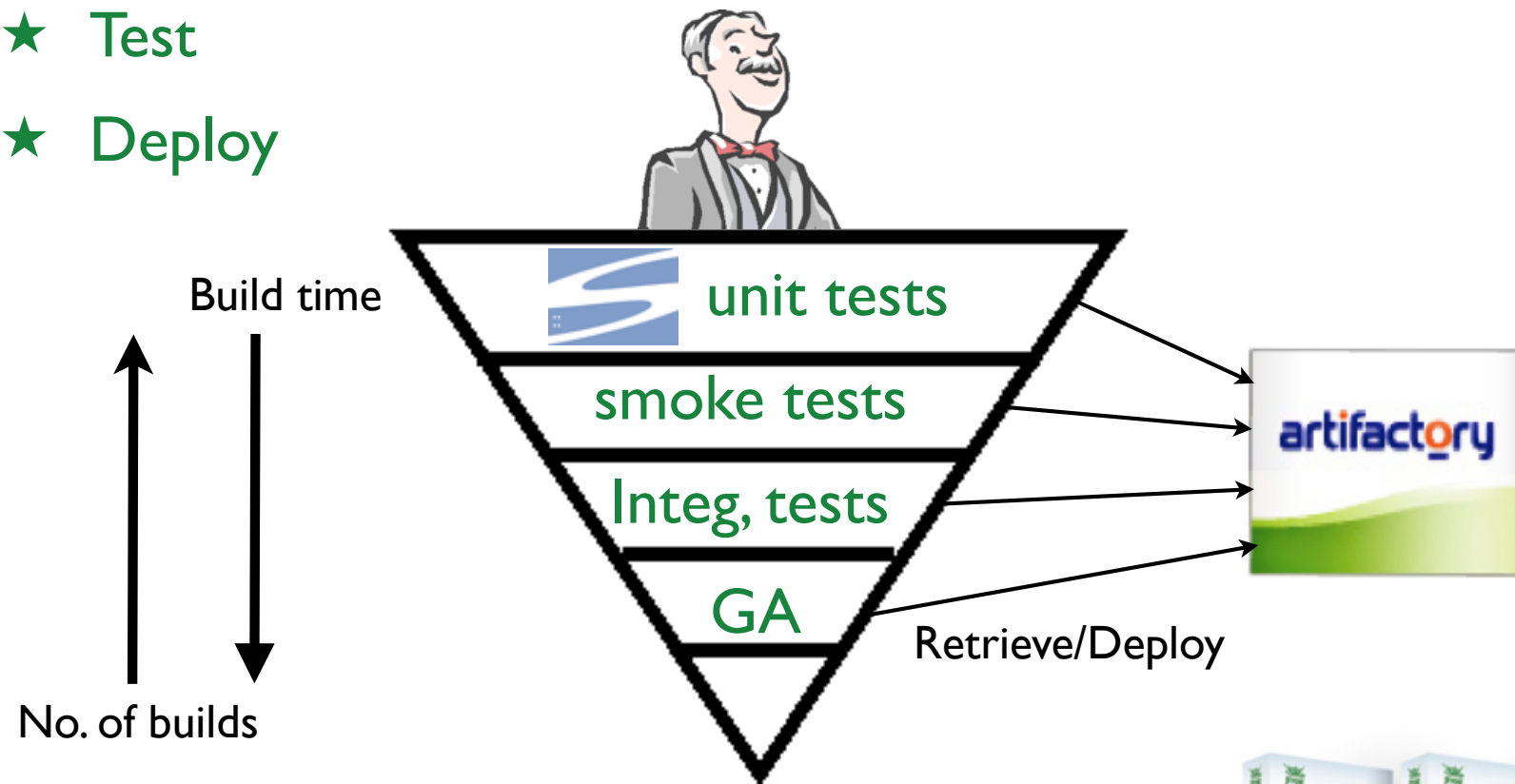
Build Scenario



The build jobs pyramid



- ★ SCM update
- ★ Retrieve
- ★ Test
- ★ Deploy



DEMO TIME



Advanced repository features



- ★ Checksum-based storage
- ★ Handling download bursts
- ★ Verifying uploaded artifacts
- ★ Locking
- ★ Cleanup bad remote repo references
- ★ Built-in metadata



Checksum-based storage



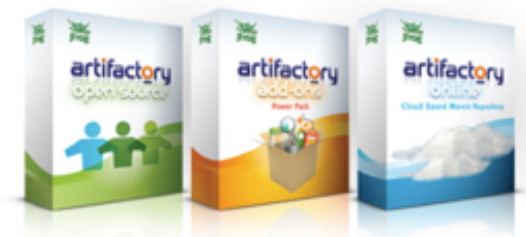
- ★ Many identical artifacts are produced and stored numerous times in the repositories
 - Unique snapshots that are exactly the same between subsequent builds
 - Other artifacts that are copied
 - ▶ Mainly needed for more natural security control
- ★ Artifactory uses a checksum-based storage
- ★ Identical artifacts are stored on the server exactly once!
 - No matter how many references are there
- ★ Copy and move are very cheap
 - Pointer operations





Concurrent downloads/ request bursts

- ★ New snapshot dependency available/ POM updated with a new dependency version
- ★ Dependency can be as big as hundreds of megs
 - Assemblies
- ★ All clients download at once
 - Network blockage (DOSing)
- ★ Artifactory will identify this
 - Queue all incoming request until the first one finishes
 - Others will get the cached version



Checksum for uploaded artifacts



- ★ No way to verify uploaded artifacts
- ★ Maven approach:
 - The repository is passive
 - All calculations done on client
 - Repository to accept and store client checksum
- ★ Artifactory
 - Compare client checksum with the one calculated on the repository
 - If in conflict return 409 until a good checksum is found
 - This behavior is configurable



Locking



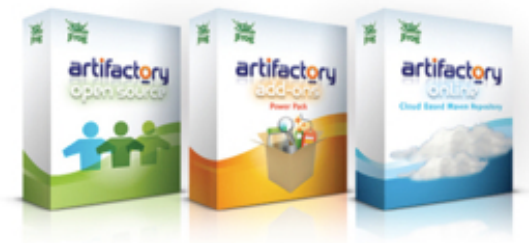
- ★ Artifactory applies RWLocks on all items
- ★ Avoid concurrent writes & dirty reads
- ★ Spans to metadata as well - cannot create metadata conflicts
- ★ Built-in utility for debugging lock contention in runtime (zero overhead)



POM cleanup



- ★ Many common third party POMs contain remote repository references making controlled resolution a nightmare
- ★ Global mirroring is not a solution
 - Forces a unified repository for releases/snapshots/plugins
- ★ Artifactory can facade POMs through a Virtual Repository
 - Can configure remote repo references to be removed
 - Original POM is intact



Metadata backed into the core



- ★ Every repository element can hold metadata definitions
 - Both files and folders
- ★ Metadata is any XML document
 - Can be queried using XPath
 - All exposed via UI and REST API
- ★ Properties tagging
 - Similar to SVN props
 - Internally stored as XML
 - “Strongly typed” props can be defined via UI
 - ▶ Open/closed lists, multi-select, single-select etc.
 - Applied via UI or REST
 - Also on deploy time with zero build tool tweaking!



Smart staging and promotion



- ★ Repository has two main roles
 - Proxy remote artifacts
 - Host deployed artifacts
 - ▶ Artifacts (should) come from CI server
- ★ Promotion of artifacts starts with a build
- ★ *The Binaries Repository & the CI Server are always interconnected*



Search-based management



- ★ Makes bulk artifact management a lot easier
- ★ Shopping-cart of artifacts
- ★ Fill-up the cart by searching and saving the search results - any search type!
- ★ Do other searches and add/subtract the results from the original
- ★ Can tweak the result manually
 - E.g. remove sources
- ★ Once done move/promote/copy/export the result
- ★ Does not enforce narrow concepts to support only specific limited use cases
 - E.g. promotion



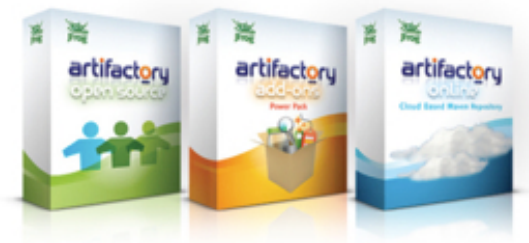
DEMO TIME



JavaFX



- ★ JavaFX/JNLP plugin
 - Compiles JavaFX sources
 - Uses standard Maven resources for classpath resolution
 - Creates Web Start JNLP files - standalone and/or browser
- ★ WebStart Virtual Repository
 - Sign jars automatically
 - Transform JNLP file href
 - Provision JNLP files and dependencies



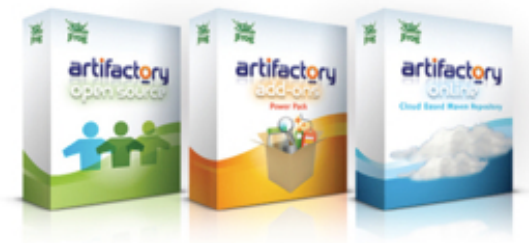
DEMO TIME



Gradle integration



- ★ Artifactory and Gradle started a strong technical collaboration
- ★ Zero configuration plugin to deploy and resolve from Artifactory when using Gradle in enterprise env.
- ★ Extract BuildInfo on the fly
 - Done!
 - Integrating with the Hudson Gradle plug-in
- ★ Gradle shows great potential
 - Zero intrusiveness
 - Ivy for mature resolution and deployments
 - Flexible control
 - Terse and easy to understand configuration
 - Ideal for integrating with a repo manager





Thank You!

Q&A

