

adaptTo()

EUROPE'S LEADING AEM DEVELOPER CONFERENCE
28th – 30th SEPTEMBER 2020

AEM Developer's best friend - Gradle AEM Plugin

Damian Mierzwiński, Krystian Panek
Cognifide a Wunderman Thompson Company



Agenda

1. Why we are here
2. Gradle basics
3. Gradle AEM Plugin basics
4. Local/Remote instance management
5. Dispatcher configuration development
6. Premiere
7. Q&A



adaptTo()

Why we are here

Developer Experience



Photo by ETA+ on Unsplash



Photo by Patrick Ward on Unsplash



Speed up development





Gradle AEM Plugin matured





Many new features





Why Gradle?

Why Gradle?





Gradle basics

- **Adaptable automation tool**
 - No plugin development required
 - Powerful DSL thanks to Kotlin (or Groovy)
- **Well supported and widely used**
 - JVM, Android, C++, Swift





Maven – software build tool

Pros

- Great conventions
- Well defined lifecycle
- Project Object Model
- Dependency management

Cons

- Configuring via XML
- Difficult to customize
- Unoptimized performance

Maven™



Gradle – adaptable automation tool

Pros

- Uses Maven goodness
- Easy customization
- Expressive DSL
- Great performance
(cache, daemon, DAG)

Cons

- Configuring via code/DSL
- Steeper learning curve



Gradle AEM Plugin(s)





Gradle AEM Plugin(s) - GAP





GAP use cases

- Building & deploying application
- Managing remote AEM instances
- Setting up local instances
- Setting up complete local AEM environment



Building & deploying application

- Awaiting instance stability
- Configuring package structure
- Leveraging cache

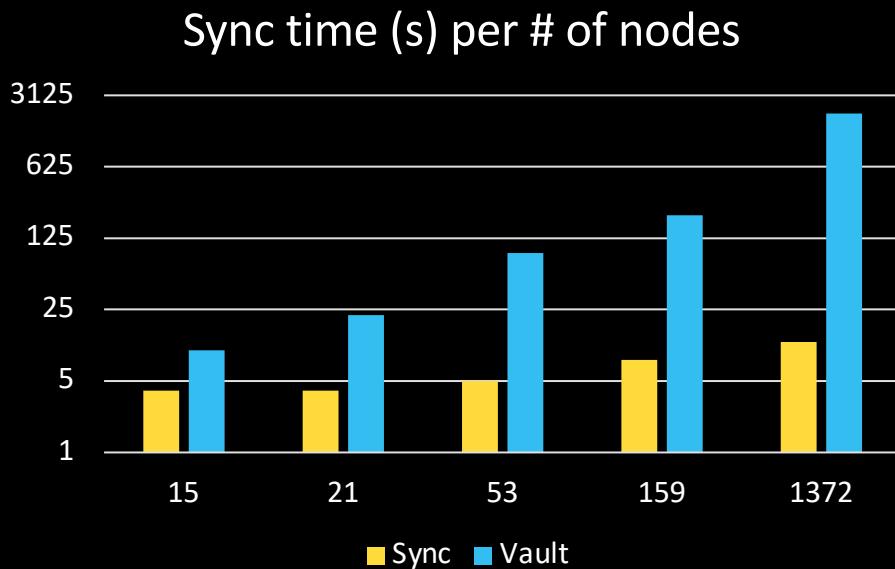


Gradle AEM Plugin(s)

Demo: Building & deploying application

Downloading JCR content

- Fast content transfer without Vault
- Normalizing content after transfer



[Documentation](#)



Gradle AEM Plugin(s)

Demo: Downloading JCR content



Downloading OSGi configuration

- Automatic creation of XML files
- Ready to store in VCS
- Actual values grabbed from instance
- No place for errors when filling via dialog

[Documentation](#)



Gradle AEM Plugin(s)

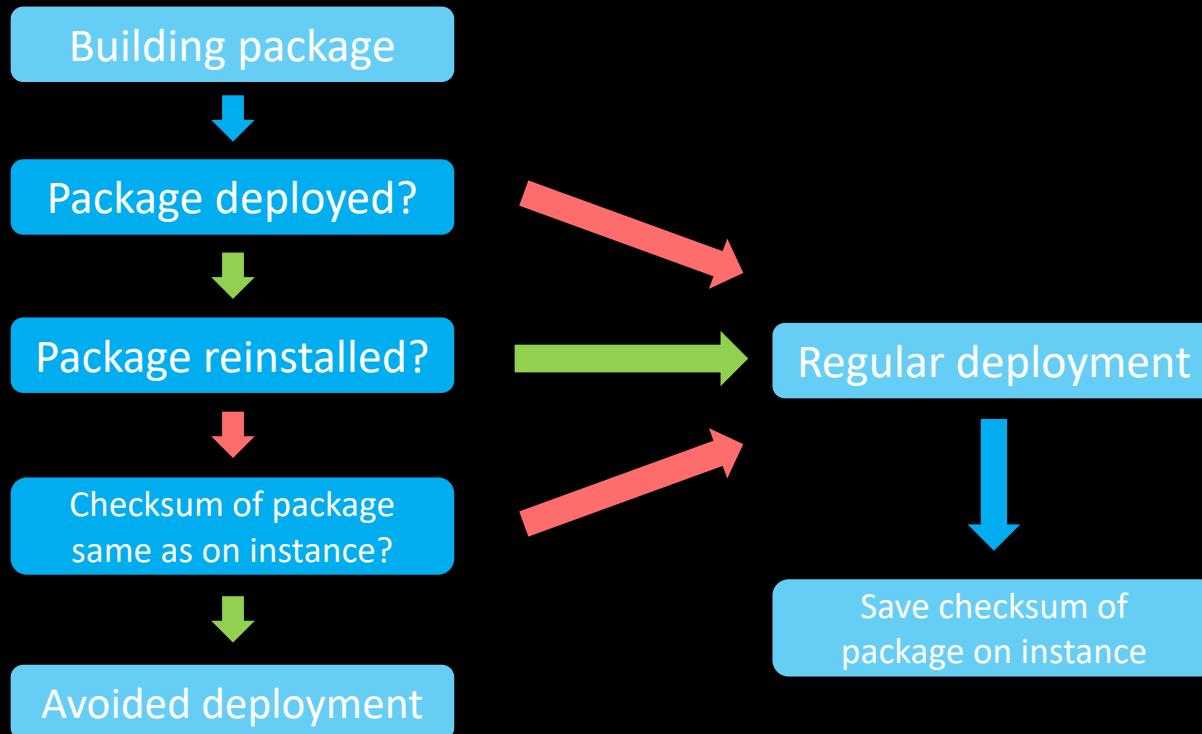
Demo: Downloading OSGi configuration



Deployment avoidance

- Redeploying unchanged package
 - Wasted time
 - Wasted resources
 - Long feedback

Deployment avoidance





Setting up local AEM instances

Setting up local instances





Setting up local instances

- Good performance
- Customized instance creation
 - Easy to change: HTTP ports, run modes, admin password
 - Debug port opened by default
- Providing extra files from remote sources (HTTP/SFTP/SMB)
(to directories: crx-quickstart/[install/config])



Provisioning instances

- Pre-defined steps for configuring e.g.
 - Deploy packages
 - Enable CRX/DE
 - Configure replication agents
- Implement custom steps using Gradle AEM DSL
- Awaiting instance stability after each step



Backing up local instances

- Archives local instance(s) into ZIP
- Automatic remote backups
- Restoring is much faster and reliable than creating complex instances

[Documentation](#)



Tailing logs

- Tailing multiple instances in parallel
- Smart notifications
 - Handling error cannonades
 - Muting non-issues
- Handling time zone differences in timestamps

[Documentation](#)



Setting up local instances

For more details...
Check out Damian's post!

<https://tech.cognifide.com/>

[blog/2020/aem-instance-setup-using-gradle-aem-plugin/](https://tech.cognifide.com/blog/2020/aem-instance-setup-using-gradle-aem-plugin/)





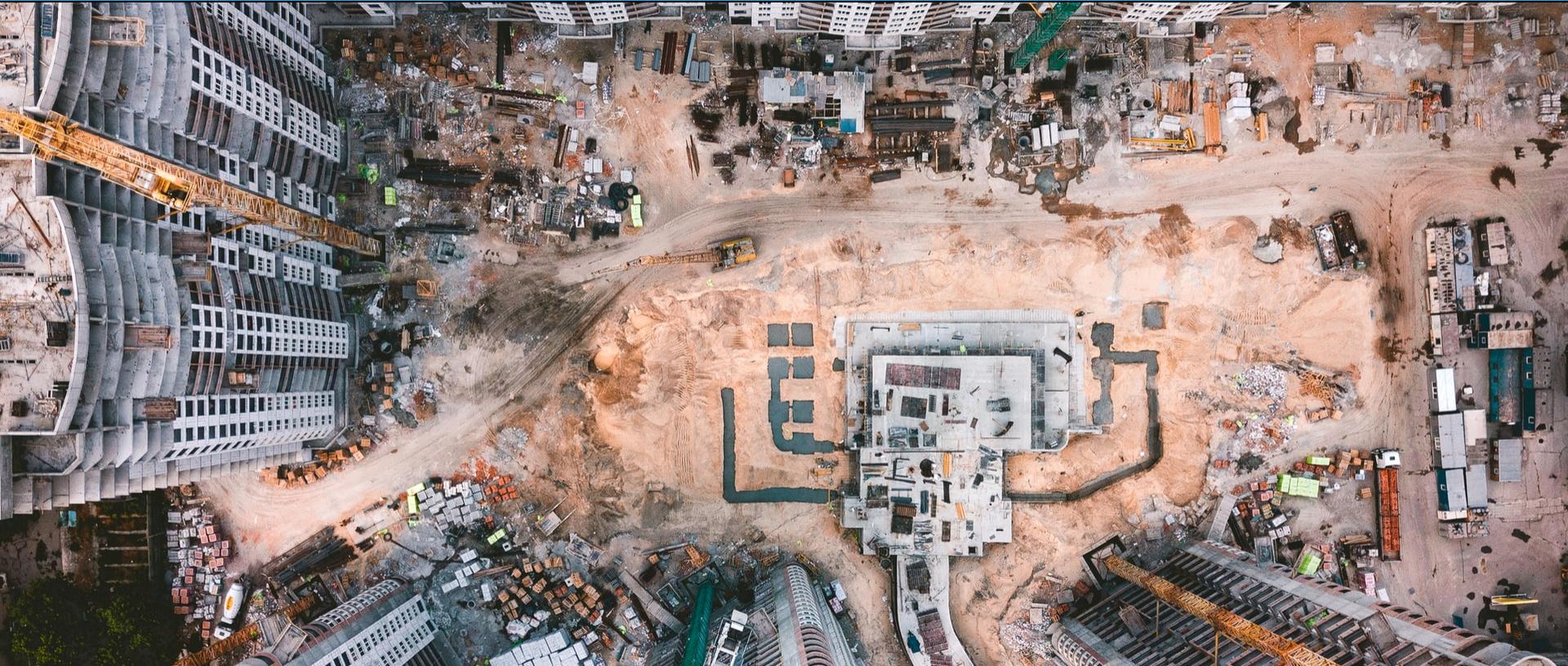
Gradle AEM Plugin(s)

Demo: Setting up local instance



Setting up full AEM environment

Setting up full AEM environment



Setting up full AEM environment





Setting up full AEM environment

- Complete - Author, Publish and Dispatcher
- Repeatable and extendable
- Developing with live reload & health checks



Gradle AEM Plugins

Demo: Setting up environment



Premiere: Gradle-powered AEM archetypes



Gradle-powered AEM archetypes





Gradle-powered AEM archetypes

All the best from Maven archetype...

- Best practices for AEM applications
- AEM Cloud Manager support



Gradle-powered AEM archetypes

...supplemented by:

- Fully automated & complete AEM env setup
- GAP build performance improvements



Gradle-powered AEM archetypes

- 2 independent build configurations
 - *.build.gradle.kts files
 - *.pom files
- Up-to-date with Adobe AEM Archetype



Gradle-powered AEM archetypes

Dual-build



Gradle

Maven™



Dual-build

- Gradle used in local development
- Maven used for integration
- Requires maintenance
of 2 separate build systems



Dual-build

- Best build performance
- Uses all Gradle goodness
(daemon, caching, parallelism)
- Modules built (bundles, packages)
 - only when changes detected
 - in parallel (if possible)

Gradle-powered AEM archetypes

Hybrid-build



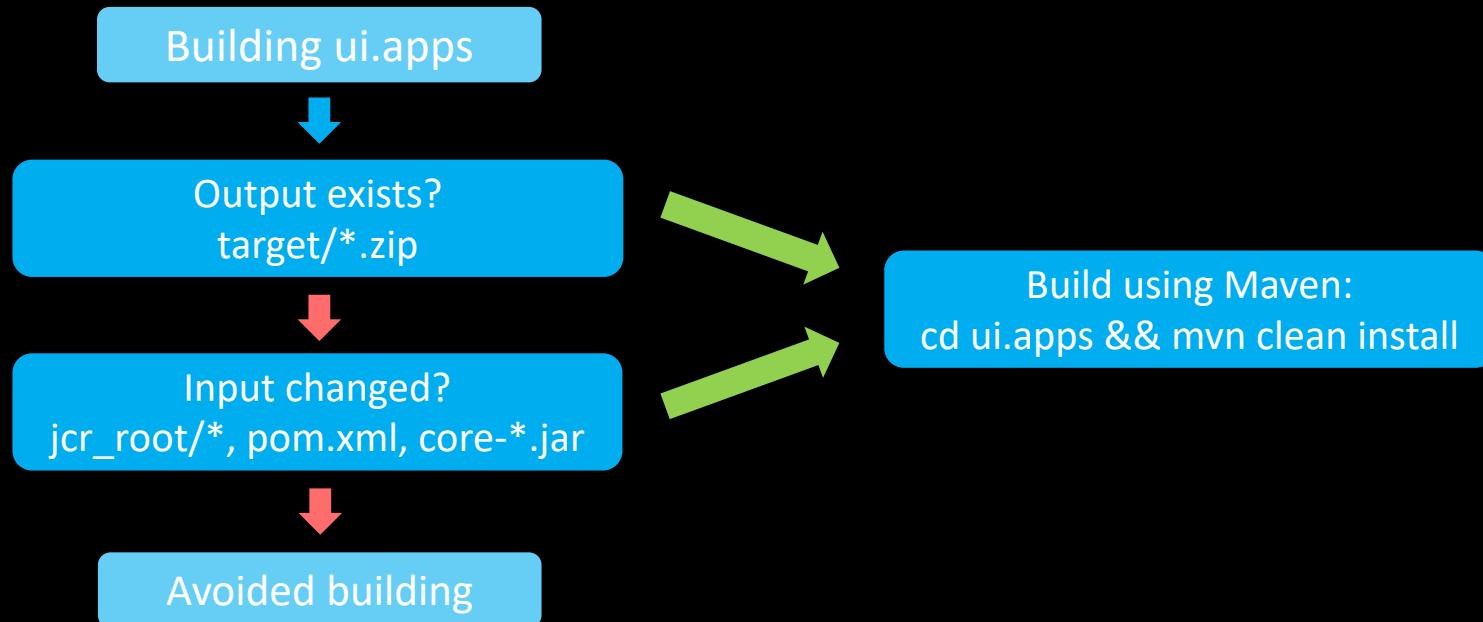


Hybrid-build

- **Gradle**
 - launches Maven one module after another
 - is only aware of build command, inputs & outputs
- **Changes usually done only in POM files**
 - Differences in built package not possible



Hybrid-build

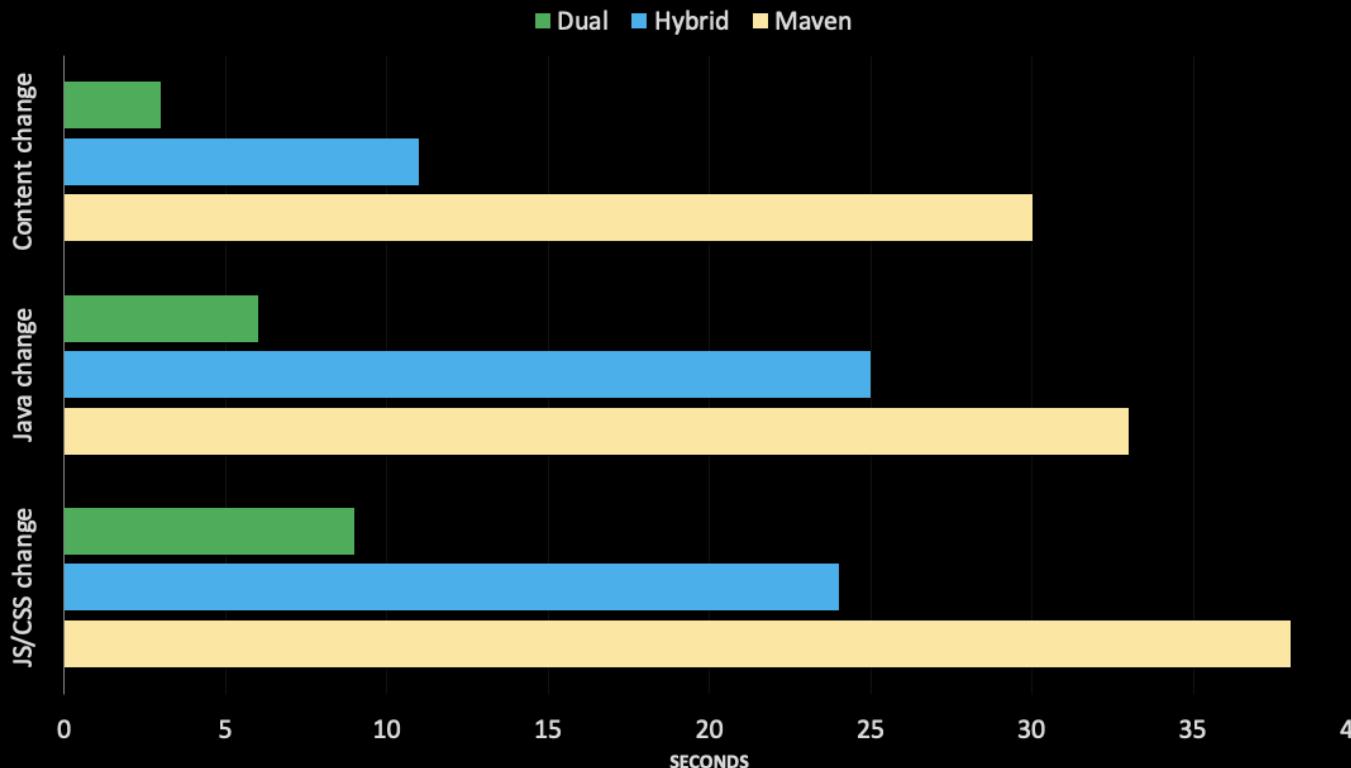




Hybrid-build

- Improved build performance
- Uses some of Gradle goodness
- Modules built (bundles, packages)
 - only when changes detected
 - in parallel (if possible)

Gradle AEM archetypes - build time





Gradle-powered AEM archetypes

For more details...
Checkout Krystian's post!

[https://tech.cognifide.com
/blog/2020/gradle-powered-aem-archetypes/](https://tech.cognifide.com/blog/2020/gradle-powered-aem-archetypes/)





Gradle AEM Plugin(s)

Demo: Gradle-powered AEM archetypes



adaptTo()

Summary

Gradle AEM Plugin(s)





Available now: Gradle-powered archetypes



[https://tech.cognifide.com
/blog/2020/gradle-powered-aem-archetypes/](https://tech.cognifide.com/blog/2020/gradle-powered-aem-archetypes/)



GAP is open-source project

<https://github.com/Cognifide>

/gradle-aem-plugin

<https://tech.cognifide.com/>





adaptTo()

Q&A



Thank you!



Appendix



Materials

- Gradle AEM Plugin open-source project: <https://github.com/Cognifide/gradle-aem-plugin>
- Gradle Environment Plugin open-source project: <https://github.com/Cognifide/gradle-environment-plugin>
- AEM project archetypes:
 - Dual-build: <https://github.com/Cognifide/aem-project-archetype#dual-gradlemaven-build>
 - Hybrid-build: <https://github.com/Cognifide/aem-project-archetype#hybrid-gradlemaven-build>
 - Multi-module: <https://github.com/Cognifide/gradle-aem-multi>
 - Local instance setup: <https://github.com/Cognifide/gradle-aem-boot>
- Learning materials, some of which will be covered in live coding:
 - <https://github.com/mierzwid/gap-workshop>
 - <https://tech.cognifide.com/tag/gradle-aem-plugin/>