# AEM Developer Tools for Eclipse

Stefan Egli, Software Developer, Adobe Basel

# AEM Developer Tools for Eclipse

Mission Statement

Make AEM application development

- easier
- more straight forward
- and aligned with best-practices

Target User

(Java) Developers

- implementing Business Logic
- based on AEM-stack

# Under the hood

99% Open Source : 1% to become Open Source too

- 95,74% based on Sling IDE Tooling

  https://sling.apache.org/documentation/development/ide-tooling.html

  Predecessor: SlingClipse by Antonio Sanso

  Developed into Sling IDE Tooling by Robert Munteanu & Stefan Egli

- 3,26% AEM specific extensions

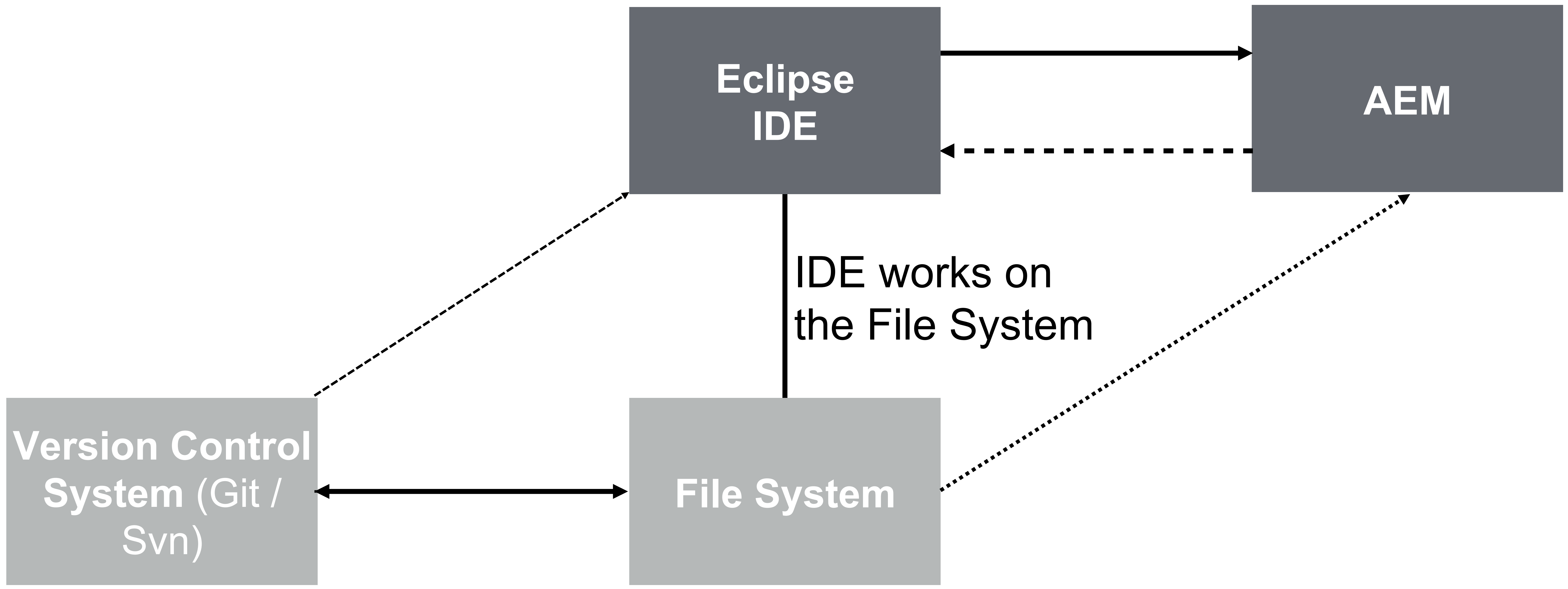
  https://github.com/Adobe-Marketing-Cloud/aem-eclipse-developer-tools

- 1% sample archetype: source to be published

  http://repo.adobe.com/nexus/content/groups/public/com/adobe/granite/archetypes/sample-project-archetype/

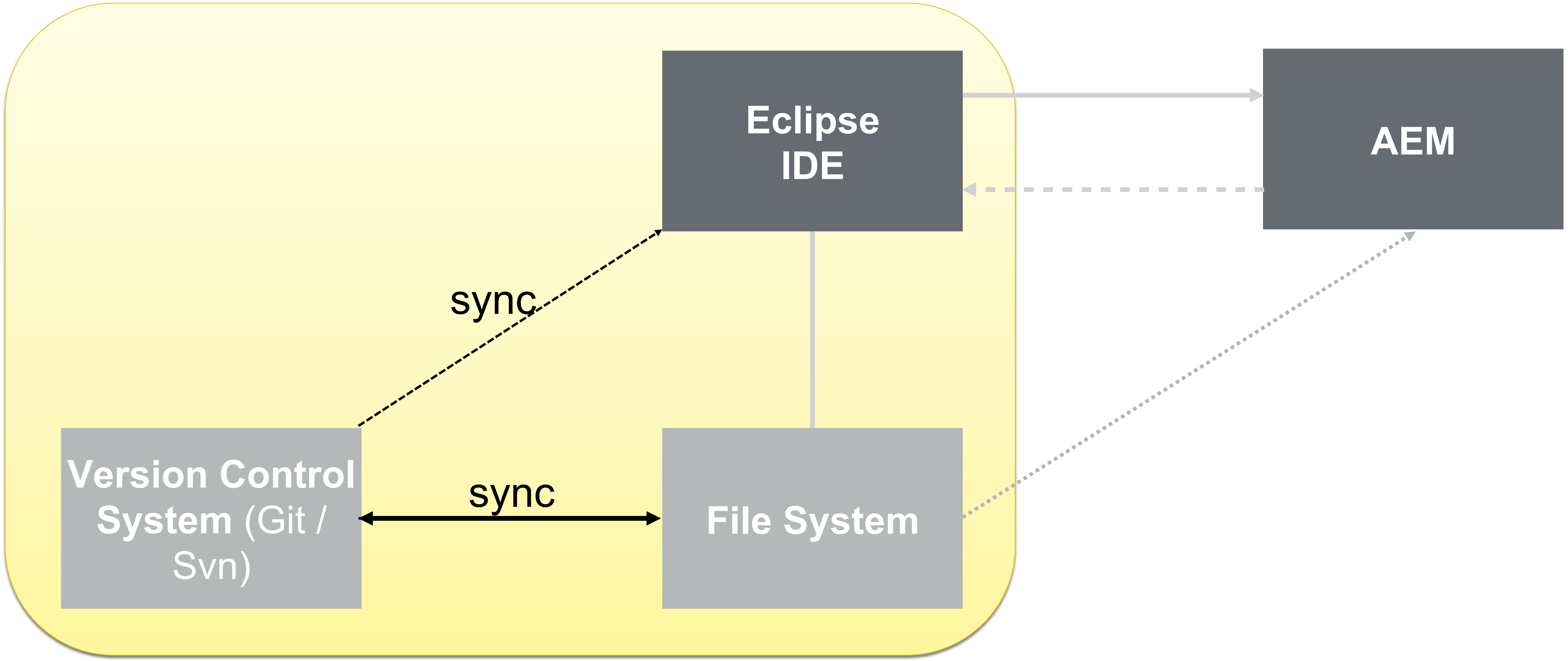
# AEM Application Development

Work on file system, team-sync + AEM-sync


Eclipse IDE
AEM
IDE works on the File System
Version Control System (Git / Svn)
File System

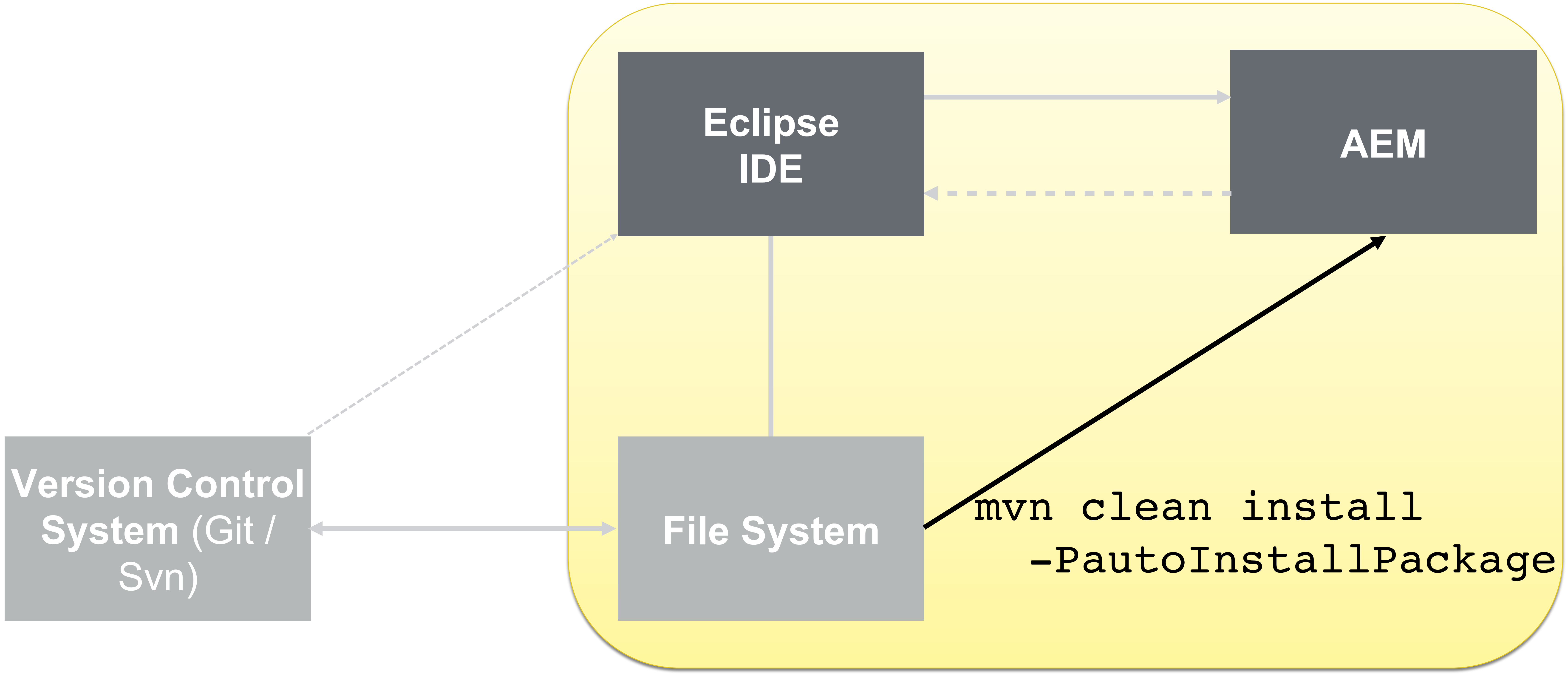
# AEM Application Development

## Team Synchronization



# AEM Application Development

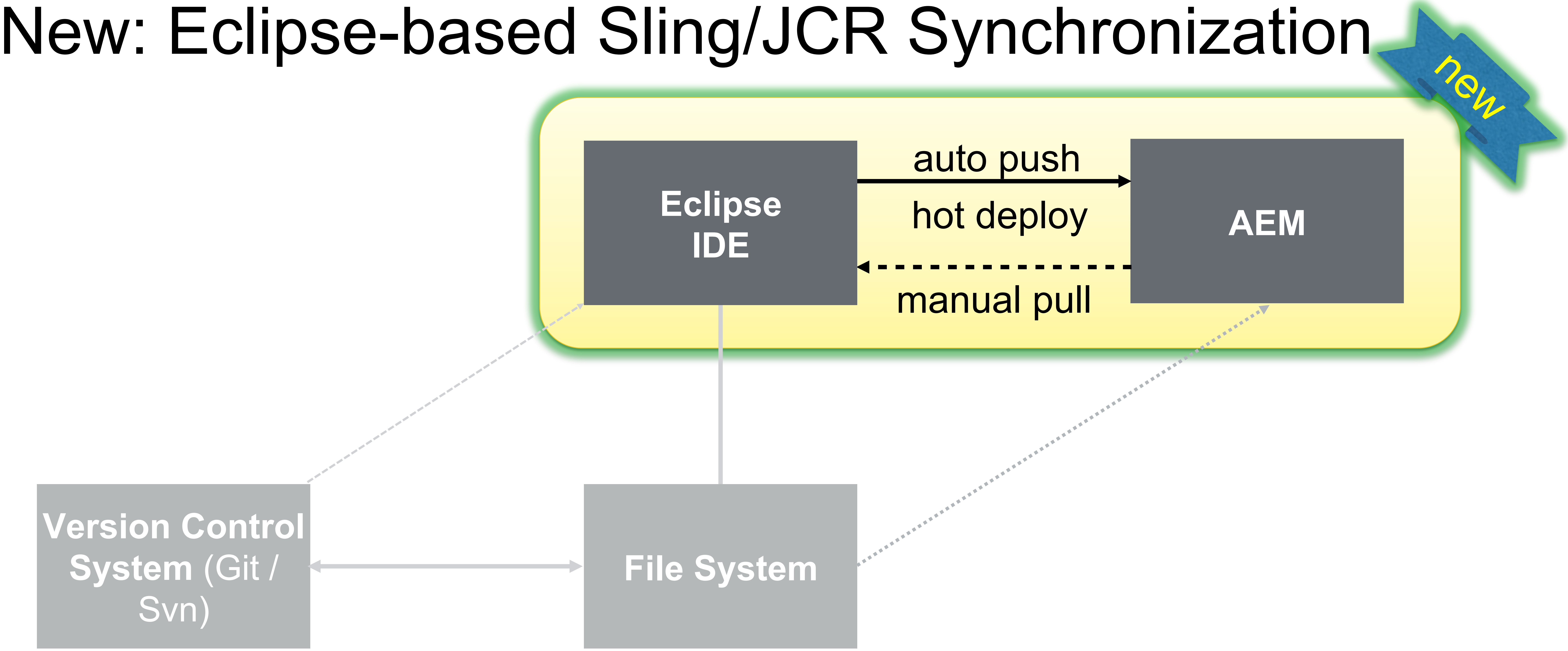
## Maven-based Sling/JCR Synchronization



```
mvn clean install
  -PautoInstallPackage
```

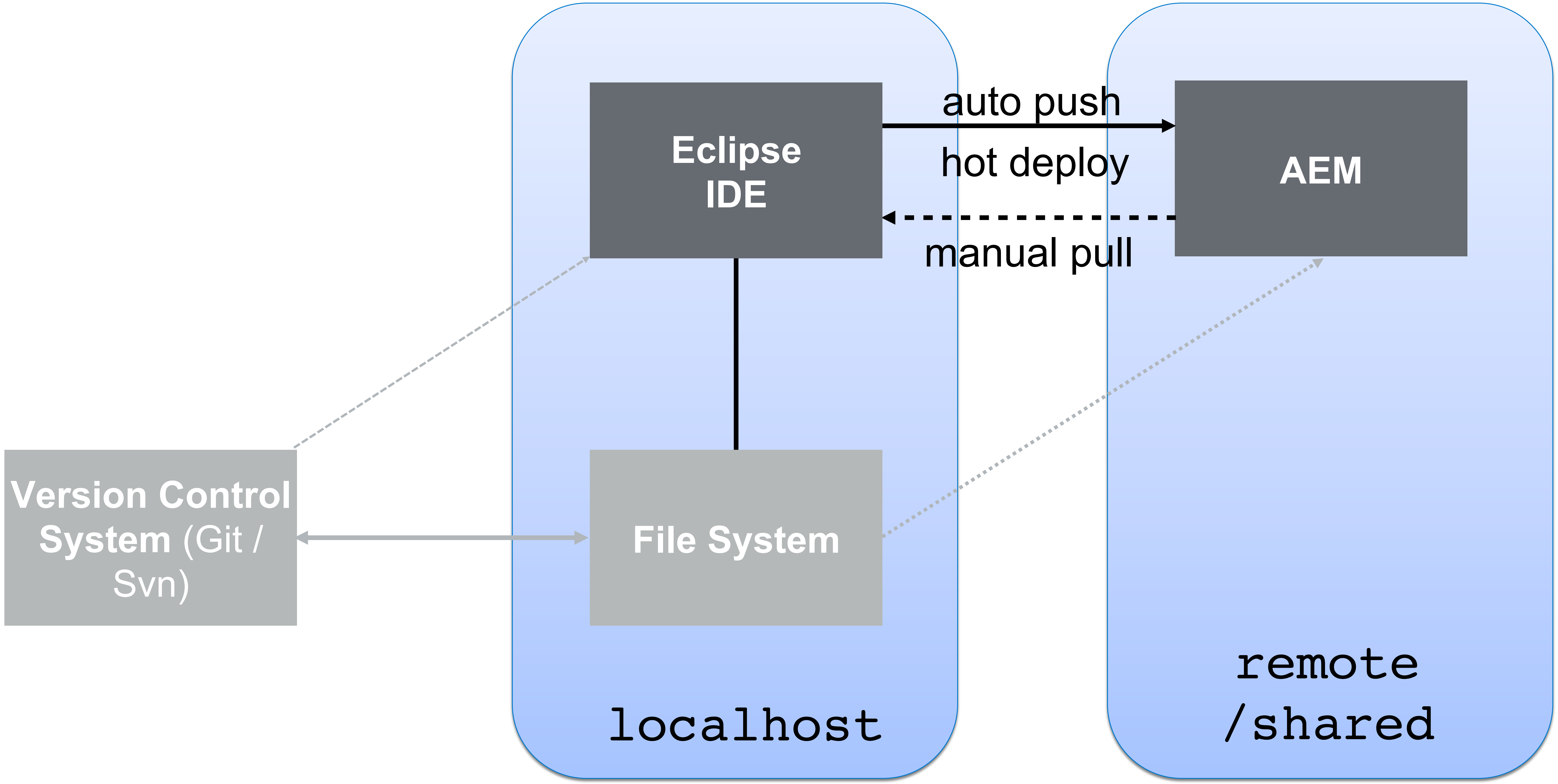
# AEM Application Development

## New: Eclipse-based Sling/JCR Synchronization



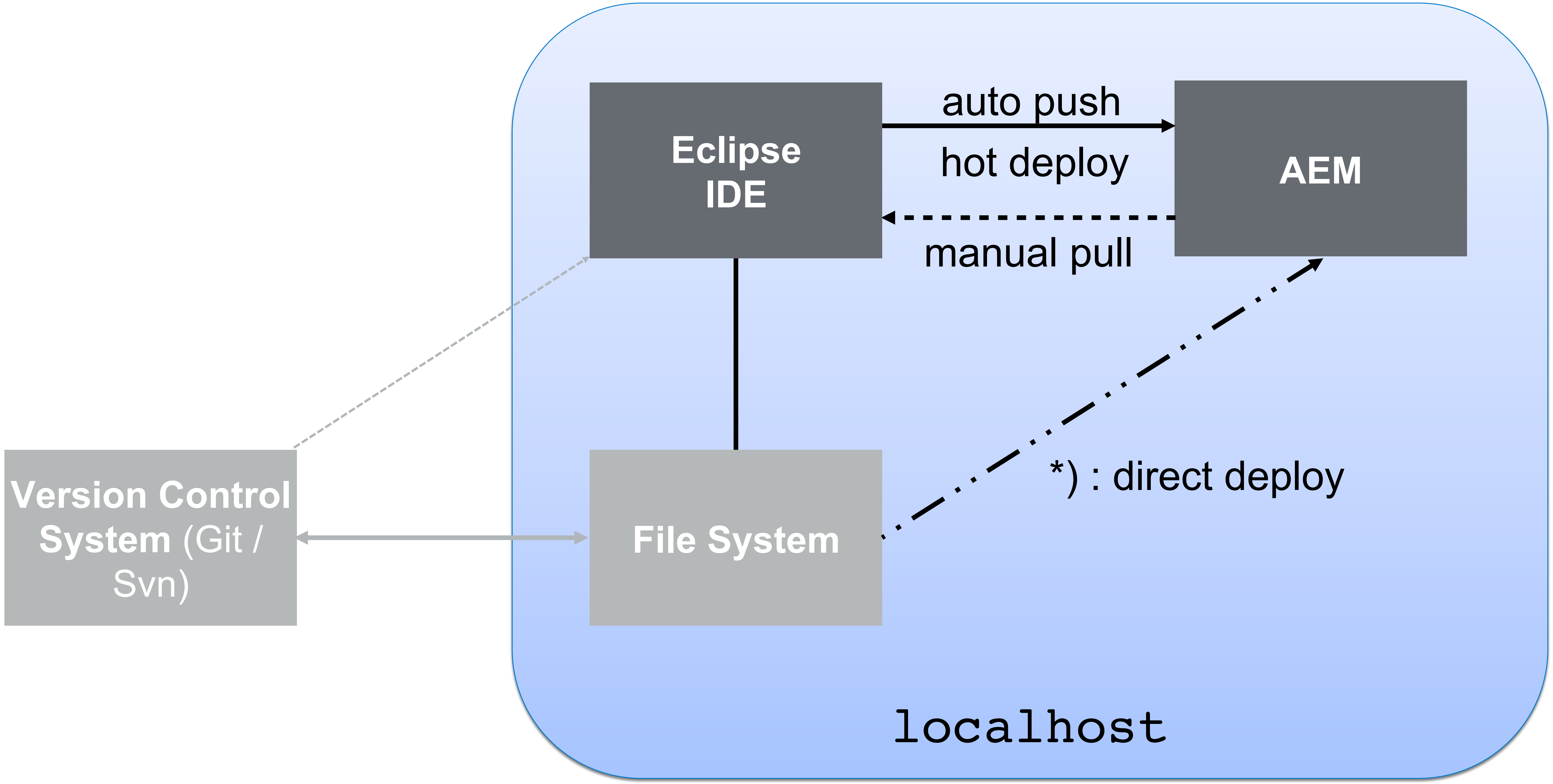
# Installation Scenarios

## Remote server



# Installation Scenarios

## localhost *)

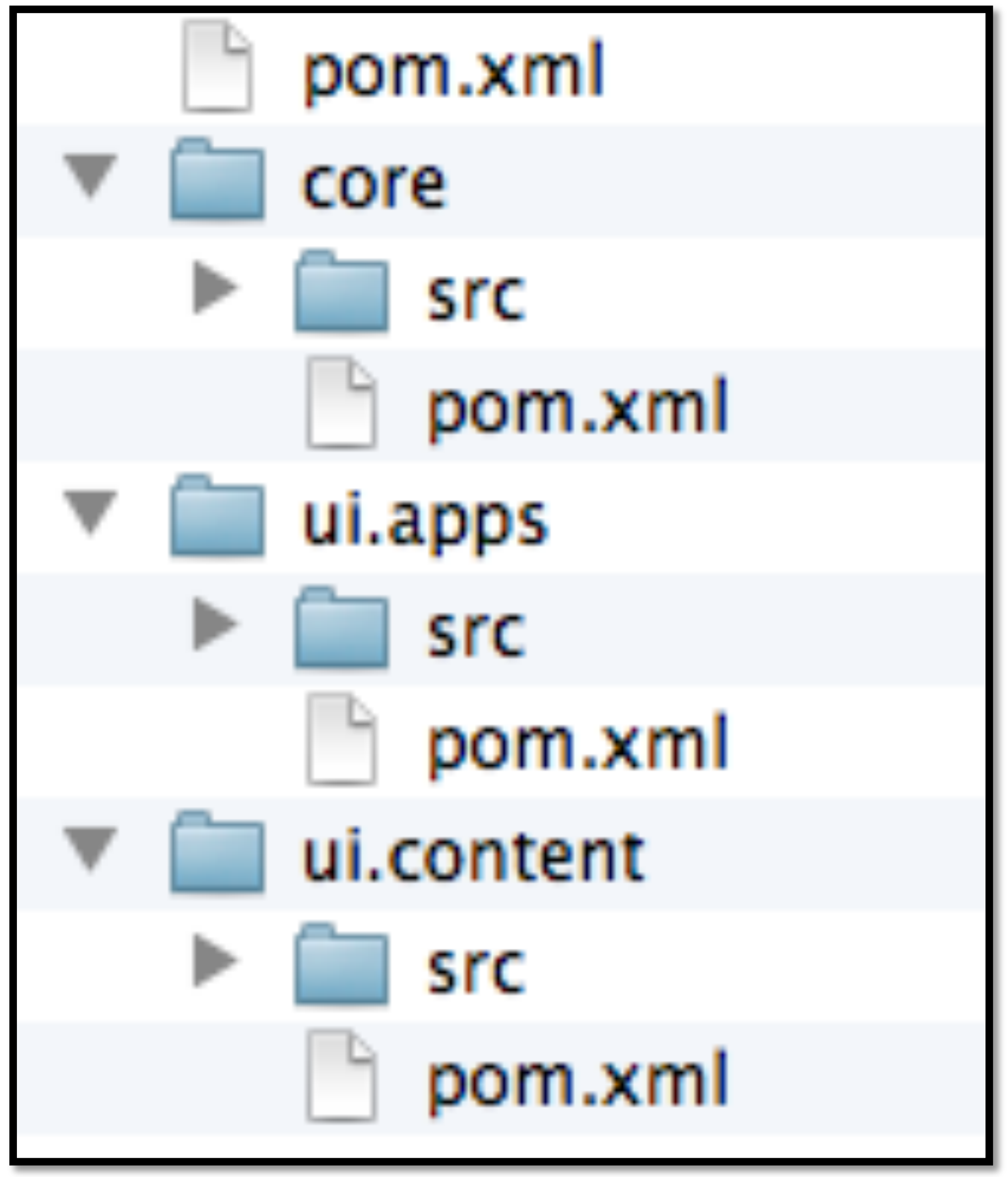


# Anatomy of an AEM Project

Parent & reactor pom

OSGi bundle
(Java code)

JCR content
(/apps and sample/test content)



```
pom.xml
▼ core
    ▶ src
      pom.xml
▼ ui.apps
    ▶ src
      pom.xml
▼ ui.content
    ▶ src
      pom.xml
```

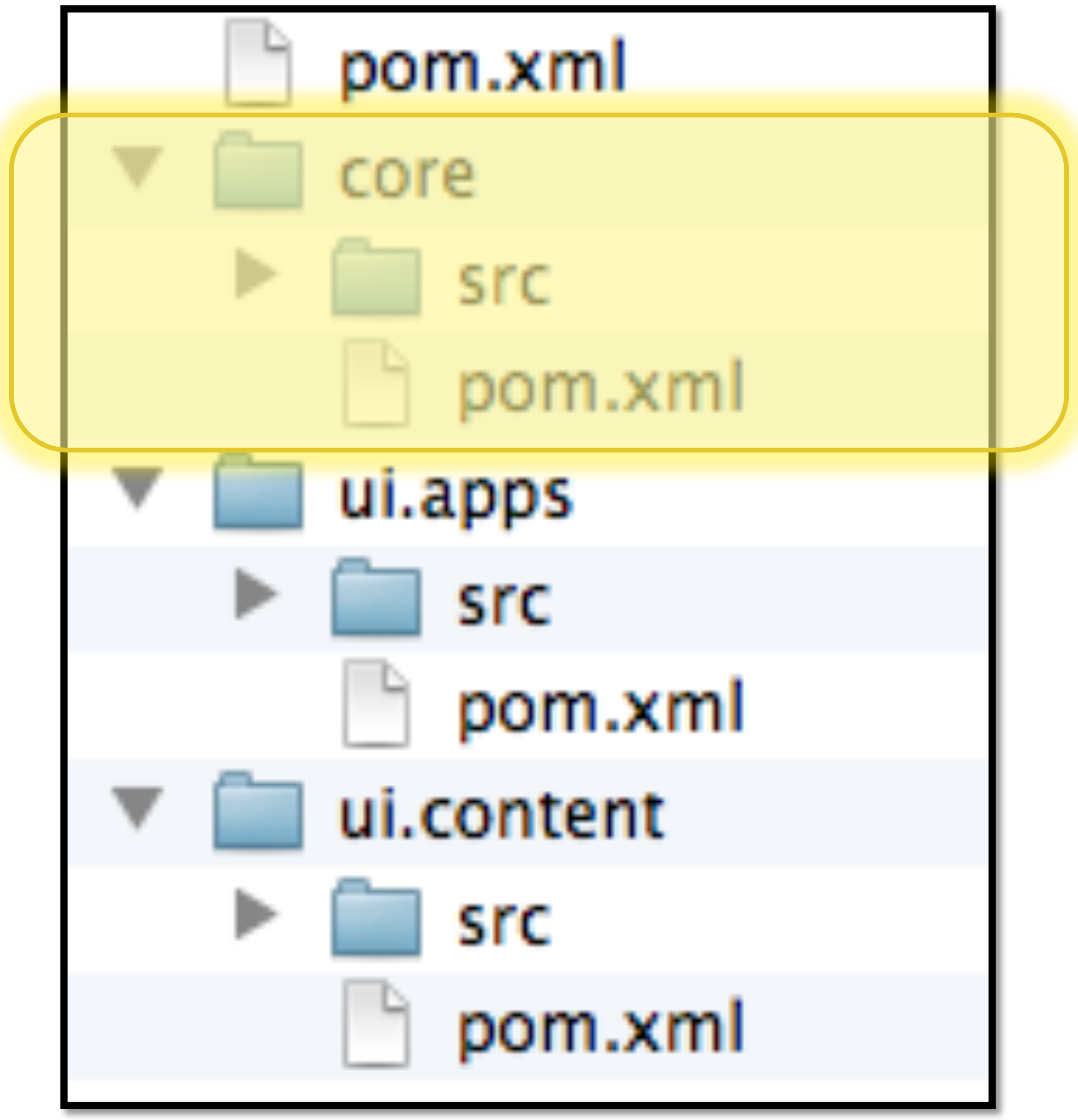
# Working with OSGi Bundles

- Install via mvn:

```
mvn clean install -PautoInstallBundle
```

new

- Install while developing in Eclipse:
  - Automatically on save, after a delay
  - Manually
  - Based on new Bundle module Facet
- Hot code replacement when in Debugger



```
pom.xml
▼ core
  ▶ src
    pom.xml
▼ ui.apps
  ▶ src
    pom.xml
▼ ui.content
  ▶ src
    pom.xml
```

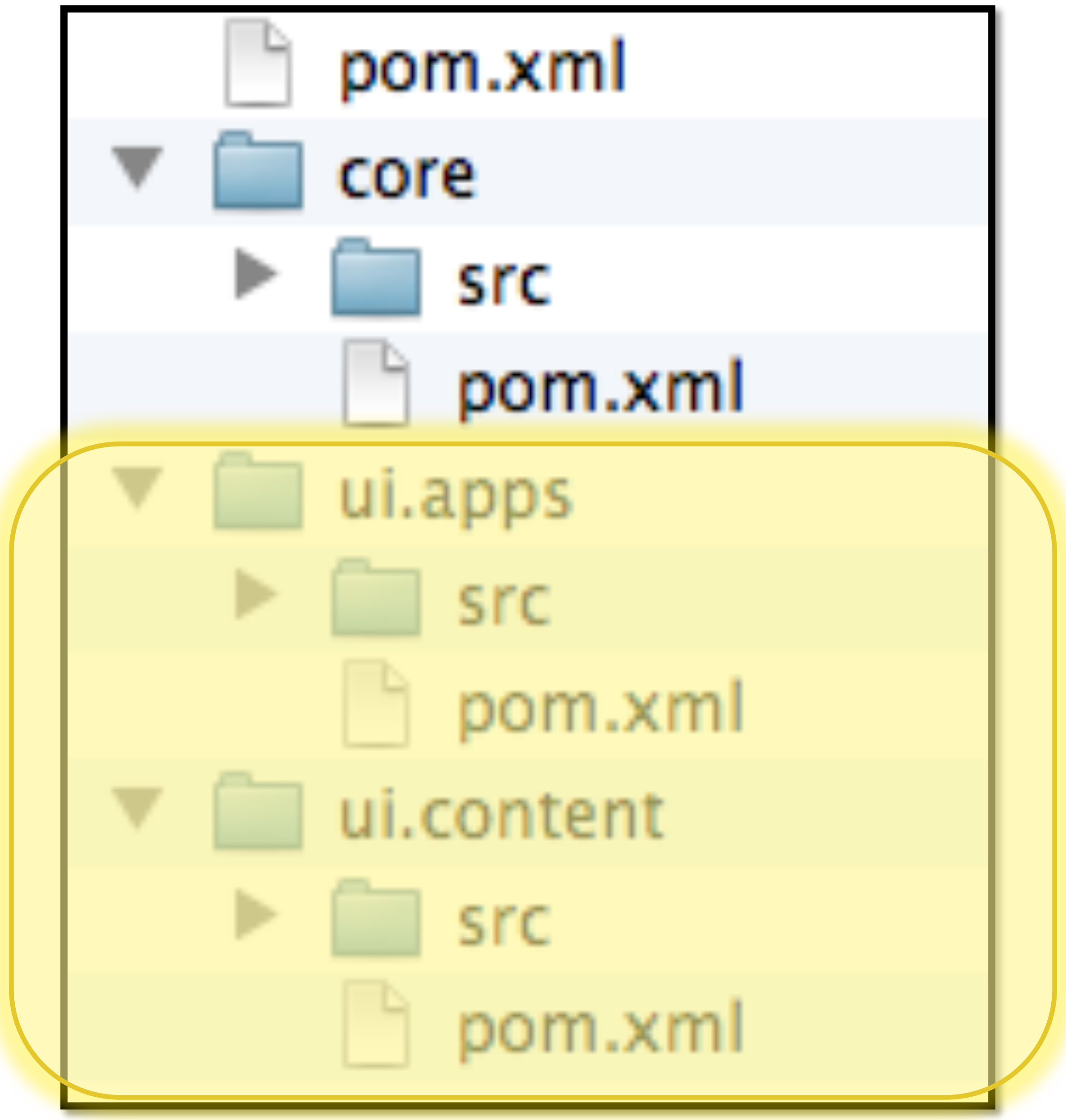
# Working with JCR content

- Install via mvn:

```
mvn clean install -PautoInstallPackage
```

new

- Install while developing in Eclipse:
  - Automatically on save, after a delay
  - Manually
  - Based on new Content module Facet
- Import content from server
- Convenient Content-Browser for viewing & editing FileVault (`.content.xml`) files



# Sling/JCR Synchronization

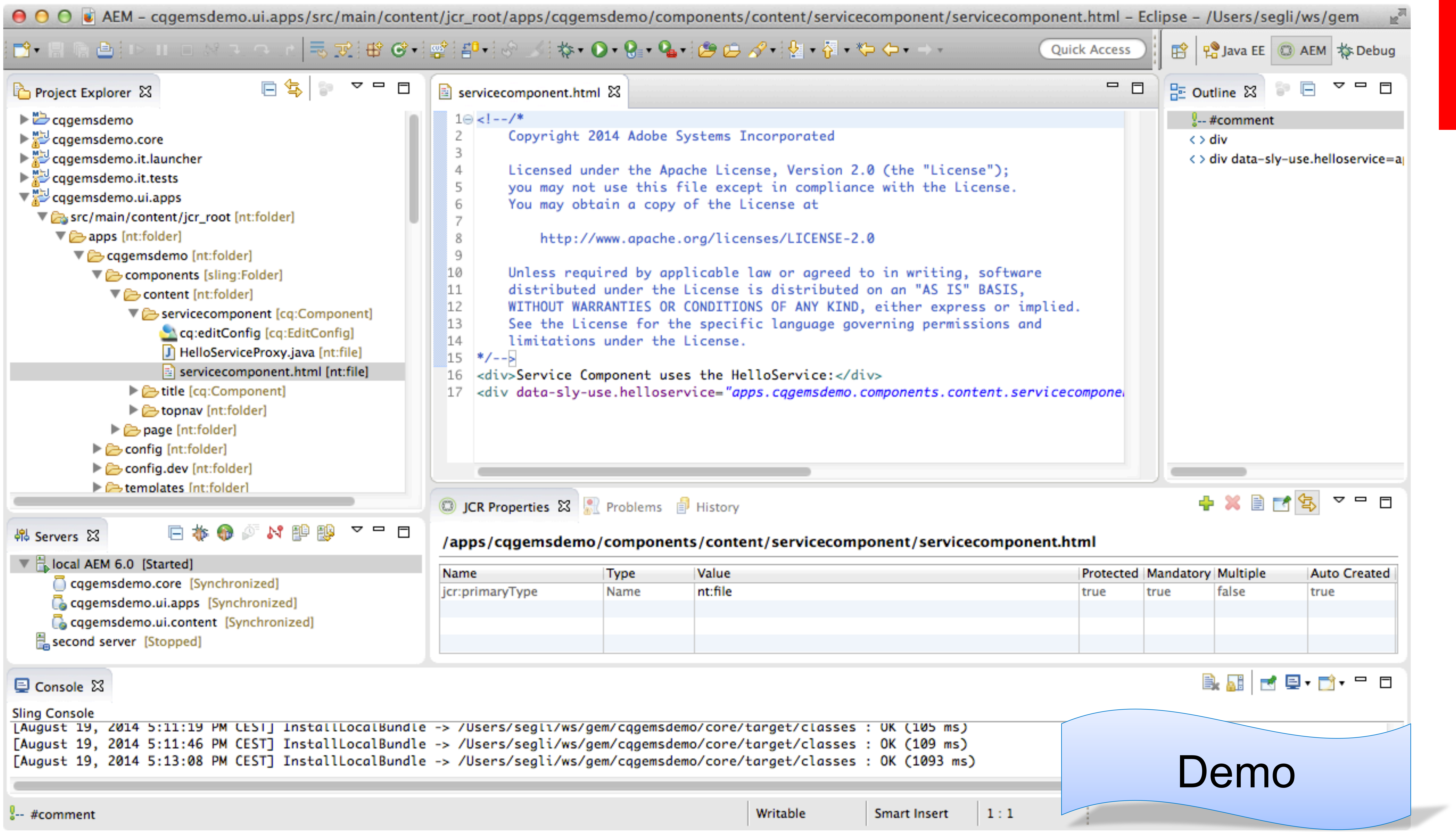
- Integrates into Eclipse’ WST framework
- New server/runtime type: AEM
- Modules with facets ‘Content’ or ‘Bundle’ module can be added
- ‘Publish’ (== export) can be customized, eg on save, delay..
- All or individual modules can be (re)published (ie exported)
- Server can be connected to in ‘run’ or ‘debug’ mode
- New Console type: ‘AEM Console’ shows log output

# Content Browser

- Added to Project Explorer for ‘Content module’ type projects
- Works directly on FileVault’s `.content.xml` etc files.
- Properties / JCR Properties view

# AEM 6.0 sample project

- Best-practice example for a multi-module project structure
- Contains four parts:
  - Parent/reactor pom.xml
  - A core (OSGi) bundle
  - Two content packages – one containing (/)apps, one containing (/)content
  - Plus three levels of testing:
    - Unit-Tests: in the core bundle
    - Integration-tests: a ‘driver’ and the actual test bundle executed on server
    - UI-Testing: tests running in the browser based on Hobbes.js


AEM – cqgemsdemo.ui.apps/src/main/content/jcr_root/apps/cqgemsdemo/components/content/servicecomponent/servicecomponent.html – Eclipse – /Users/segli/ws/gem
Quick Access
Java EE
AEM
Debug
Project Explorer
servicecomponent.html
Outline
#comment
div
div data-sly-use.helloservice=a
JCR Properties
Problems
History
Servers
Console
Writable
Smart Insert
1 : 1
#comment

```
▶ cqgemsdemo
▶ cqgemsdemo.core
▶ cqgemsdemo.it.launcher
▶ cqgemsdemo.it.tests
▼ cqgemsdemo.ui.apps
  ▼ src/main/content/jcr_root [nt:folder]
    ▼ apps [nt:folder]
      ▼ cqgemsdemo [nt:folder]
        ▼ components [sling:Folder]
          ▼ content [nt:folder]
            ▼ servicecomponent [cq:Component]
                cq:editConfig [cq:EditConfig]
                HelloServiceProxy.java [nt:file]
                servicecomponent.html [nt:file]
            ▶ title [cq:Component]
            ▶ topnav [nt:folder]
          ▶ page [nt:folder]
        ▶ config [nt:folder]
        ▶ config.dev [nt:folder]
        ▶ templates [nt:folder]
```

```
<!--/*
    Copyright 2014 Adobe Systems Incorporated

    Licensed under the Apache License, Version 2.0 (the "License");
    you may not use this file except in compliance with the License.
    You may obtain a copy of the License at

        http://www.apache.org/licenses/LICENSE-2.0

    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    See the License for the specific language governing permissions and
    limitations under the License.
*/-->
<div>Service Component uses the HelloService:</div>
<div data-sly-use.helloservice="apps.cqgemsdemo.components.content.servicecompone
```

```
▼ local AEM 6.0  [Started]
    cqgemsdemo.core  [Synchronized]
    cqgemsdemo.ui.apps  [Synchronized]
    cqgemsdemo.ui.content  [Synchronized]
  second server  [Stopped]
```

/apps/cqgemsdemo/components/content/servicecomponent/servicecomponent.html

| Name | Type | Value | Protected | Mandatory | Multiple | Auto Created |
|---|---|---|---|---|---|---|
| jcr:primaryType | Name | nt:file | true | true | false | true |

Sling Console

```
[August 19, 2014 5:11:19 PM CEST] InstallLocalBundle -> /Users/segli/ws/gem/cqgemsdemo/core/target/classes : OK (105 ms)
[August 19, 2014 5:11:46 PM CEST] InstallLocalBundle -> /Users/segli/ws/gem/cqgemsdemo/core/target/classes : OK (109 ms)
[August 19, 2014 5:13:08 PM CEST] InstallLocalBundle -> /Users/segli/ws/gem/cqgemsdemo/core/target/classes : OK (1093 ms)
```

Demo

# Outlook

- <User Feedback/>
- Sightly Support
- AEM Components Support
- Tighter integration on OSGi level
- Other IDEs (eg IntelliJ)

# Resources

AEM Developer Tools Download Site
http://eclipse.adobe.com/aem/dev-tools/

Sling IDE Tooling
https://sling.apache.org/documentation/development/ide-tooling.html

Forum for discussions & questions
http://help-forums.adobe.com/content/adobeforums/en/experience-manager-forum/adobe-experience-manager.html

Issue reporting & tracking
https://github.com/Adobe-Marketing-Cloud/aem-eclipse-developer-tools/issues

Hobbes.js: UI Testing Framework
http://docs.adobe.com/docs/en/aem/6-0/develop/components/hobbes.html

Adobe