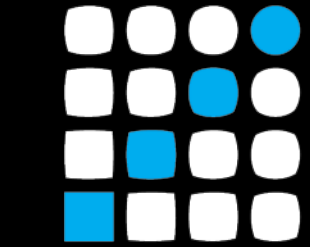


APACHE SLING & FRIENDS TECH MEETUP

10-12 SEPTEMBER 2018

Apache Sling Scripting Reloaded

Radu Cotescu, Karl Pauls – Adobe

rev. 4.20180910

adaptTo()
Who
are we?

Who are we: Mr. Scripting

- Computer Scientist @ Adobe, Basel, Switzerland
- Member of the Apache Sling PMC
- Main contributor for the HTL implementation
- Initiator of the Apache Sling Validation Framework (adaptTo 2014)

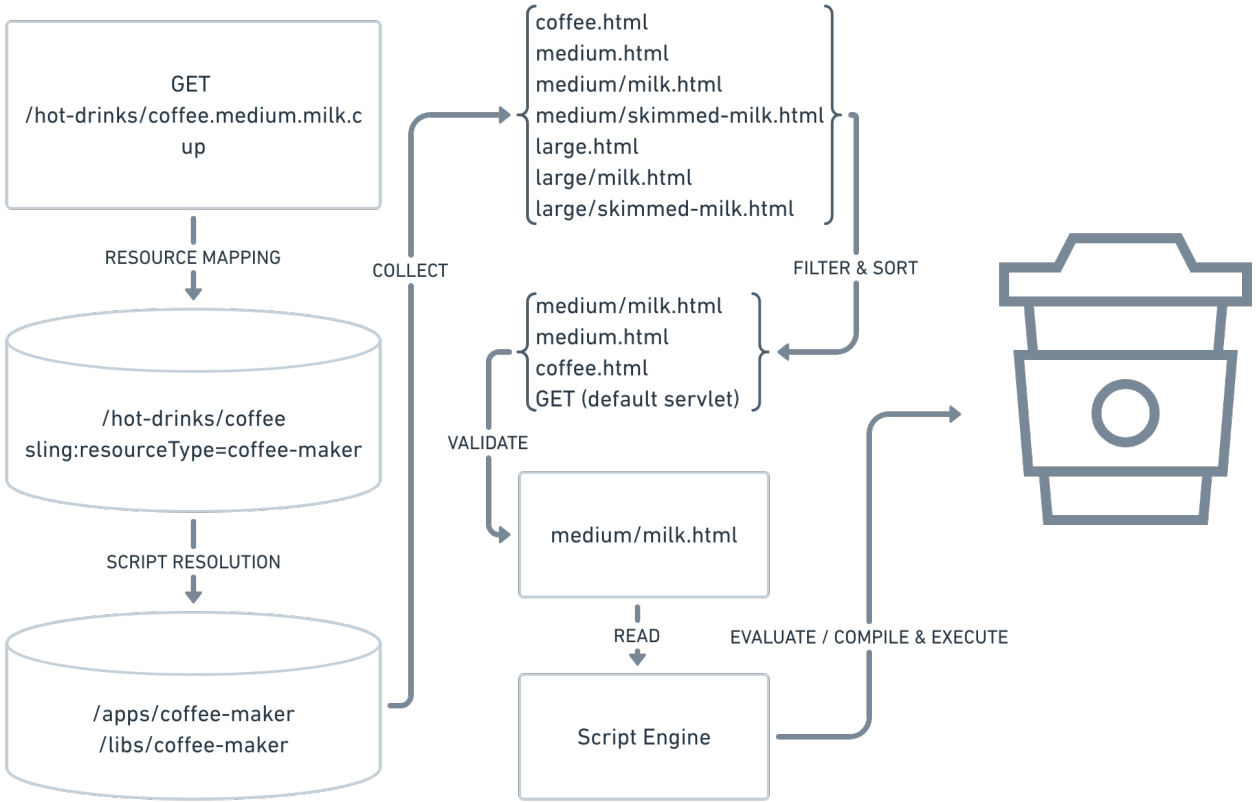
adaptTo()

Who are we: Mr. Classloader

- Computer Scientist @ Adobe, Basel, Switzerland
- Member of the Apache Software Foundation
- PMC Member Apache Felix and Sling (VP Apache Felix)
- Co-Author OSGi in Action

Scripting

A simplified view



adaptTo()

Versioning and dependencies

- There is no standard way of defining either.
- Adobe's Core Components use resource type versioning by path conventions.
- Dependencies can only be checked at runtime (but not enforced).
- What happens if your evil colleagues delete a component you were delegating to? Or worse, if they change the whole markup?

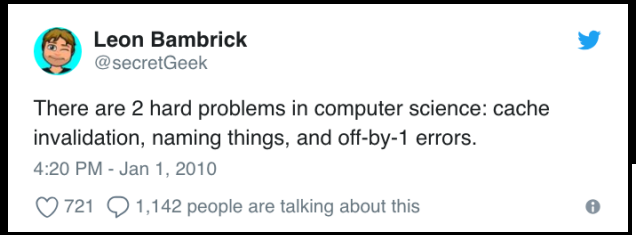
Performance

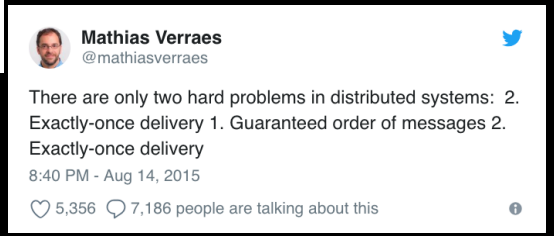
- Each script requires two trips to the persistence layer when first compiled, only to read the script.
- Sling needs to maintain some caches to keep things snappy.

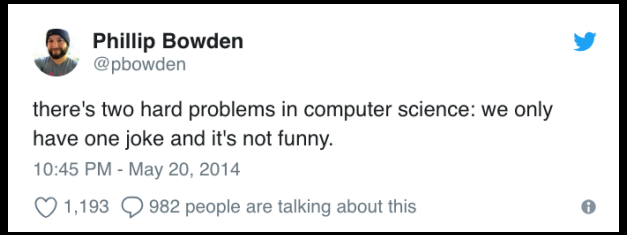
adaptTo()

Performance

“There are only two hard things in Computer Science: cache invalidation and naming things.” -- Phil Karlton







Reality check

1. What are scripts actually: content or code?
2. Are scripts authored or developed?
3. Can scripts be used freely or do they have constraints?

Reality check

If scripts are code, why do we treat them differently?

Reality check

Code:

1. provides or implements an API (HTTP in our case)
2. evolves semantically
3. is bundled into a cohesive unit, managed by one or more developers

Apache Sling Scripting Reloaded

Apache Sling Scripting Reloaded

Core principles:

1. pack scripts into OSGi bundles
2. define the resource types as versioned capabilities, with versioned requirements (Java APIs, other resource types to which scripts delegate or which scripts extend)
3. allow the platform to do what it's made to: wire things

Apache Sling Scripting Reloaded

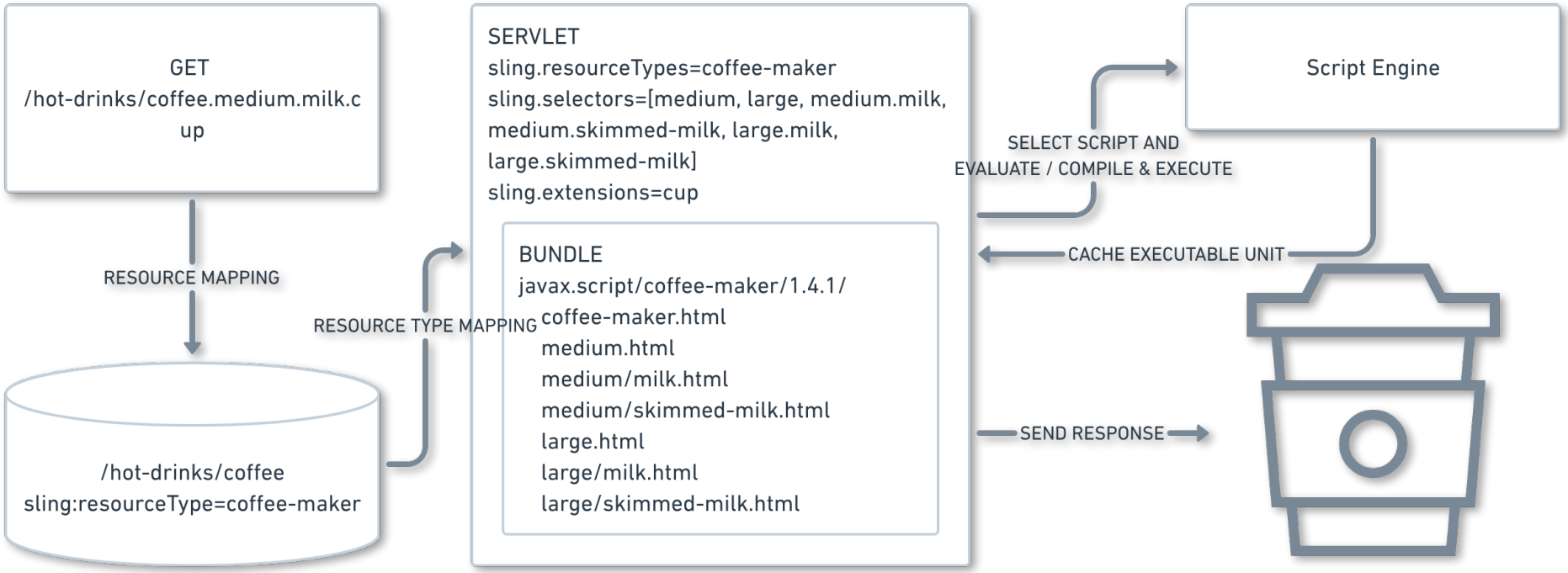
Capability - Describing a feature or function of the Resource when installed in the Environment. A capability has attributes and directives.

Requirement - An assertion on the availability of a capability in the Environment. A requirement has attributes and directives. The `filter` directive contains the filter to assert the attributes of the capability in the same Namespace.

https://osgi.org/specification/osgi.core/7.0.0/framework.module.html#framework.module.dependencies

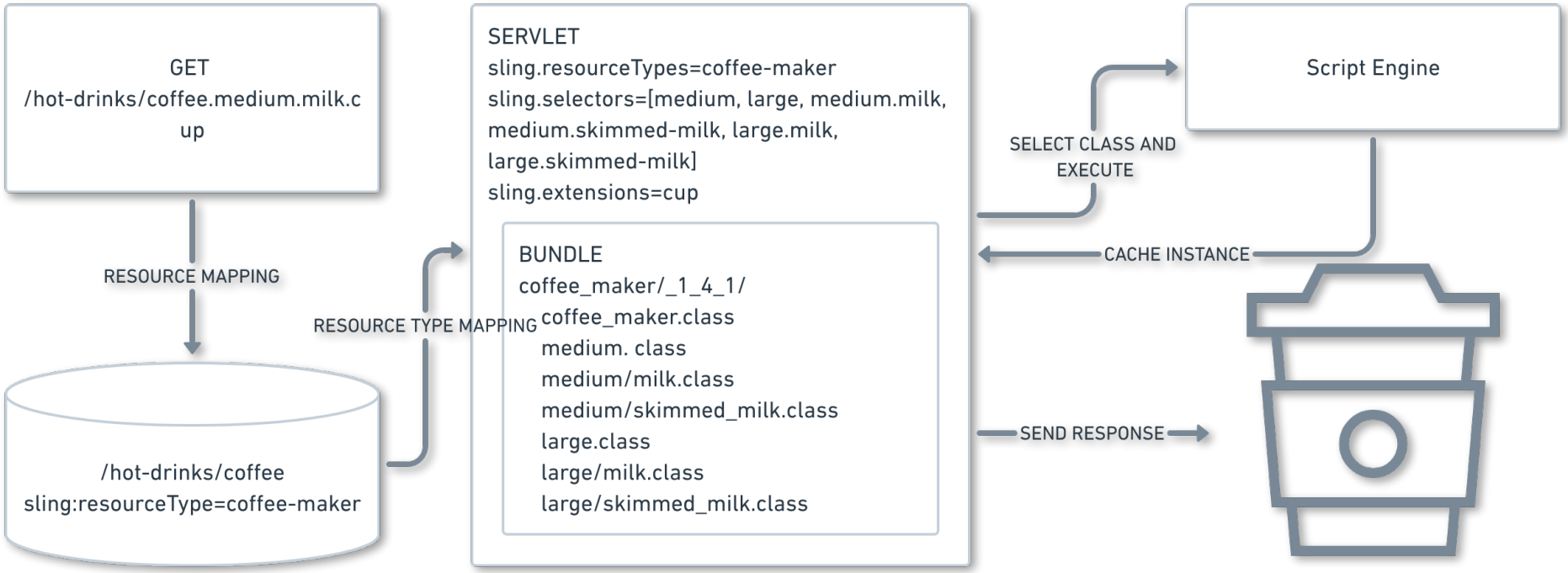
What's different?

Option 1: scripts packed as bundle entries

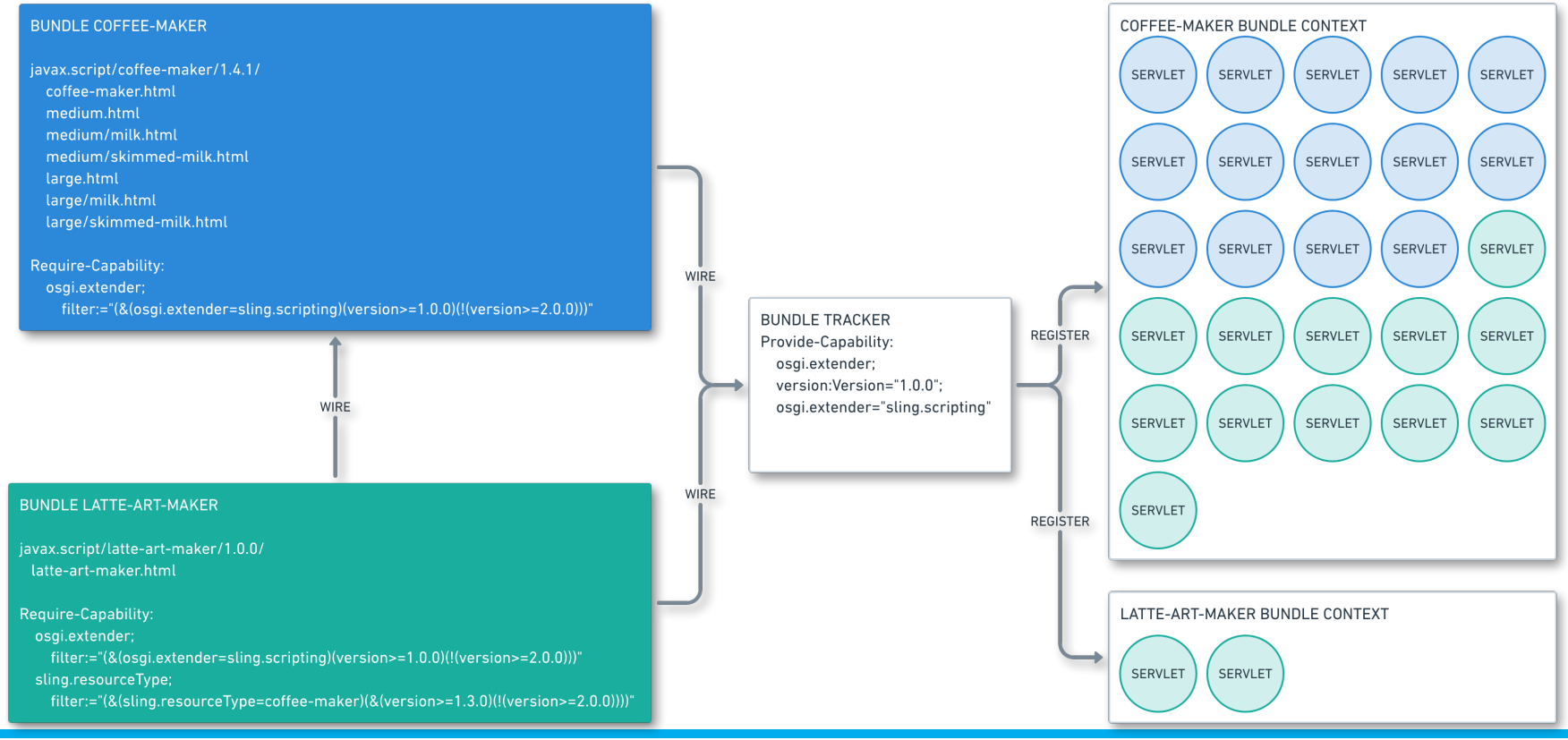


What's different?

Option 2: precompiled scripts



How?


BUNDLE COFFEE-MAKER
javax.script/coffee-maker/1.4.1/
coffee-maker.html
medium.html
medium/milk.html
medium/skimmed-milk.html
large.html
large/milk.html
large/skimmed-milk.html
Require-Capability:
osgi.extender;
filter:="(&(osgi.extender=sling.scripting)(version>=1.0.0)(!(version>=2.0.0)))"
WIRE
WIRE
BUNDLE LATTE-ART-MAKER
javax.script/latte-art-maker/1.0.0/
latte-art-maker.html
Require-Capability:
osgi.extender;
filter:="(&(osgi.extender=sling.scripting)(version>=1.0.0)(!(version>=2.0.0)))"
sling.resourceType;
filter:="(&(sling.resourceType=coffee-maker)(&(version>=1.3.0)(!(version>=2.0.0))))"
WIRE
BUNDLE TRACKER
Provide-Capability:
osgi.extender;
version:Version="1.0.0";
osgi.extender="sling.scripting"
REGISTER
REGISTER
COFFEE-MAKER BUNDLE CONTEXT
SERVLET
LATTE-ART-MAKER BUNDLE CONTEXT
SERVLET
SERVLET

adaptTo()

Sure, but how?

1 Provide-Capability / Script -> 1 Servlet / Script

Provide-Capability

```
sling.resourceType=“latte-art-maker”;
    sling.servlet.methods:List<String>=“GET”;
    version:Version=“1.0.0”
```

adaptTo()

Apache Sling Scripting Reloaded

1. New add-on module to which bundles that provide scripts have to be wired explicitly
2. Reuses the already established mechanisms for registering servlets in Apache Sling
3. Allows to build more light-weight instances that can be thrown into production with very little warm-up, when using precompiled scripts (10x faster than scripts compiled from the repository for the 1st request)

Apache Sling Scripting Reloaded

4. Provides the mechanism for deploying truly versionable scripts, with explicit dependencies, by relying on the OSGi framework
5. Removes the need of a separate `ScriptCache`
6. Removes additional pressure from the persistence layer
7. Simplifies instance and application upgrades
8. Maven plugin for generating requirements and capabilities

Demo*

*or how we can embarrass ourselves if things don't work

Apache Sling Scripting Reloaded

Future plans:

1. Enhance development workflow, similar to what script developers have now
2. Add the module to the Apache Sling Starter Application and maybe convert one of the sample apps to use bundled precompiled scripts

Q&A

Resources

- You can find our code at: https://github.com/apache/sling-whiteboard/tree/master/scripting-resolver.
- https://sling.apache.org/documentation/the-sling-engine/url-to-script-resolution.html
- https://sling.apache.org/documentation/the-sling-engine/servlets.html
- Our diagrams were designed with https://whimsical.co/flowcharts/
- Assets licensed from https://stock.adobe.com/