APACHE SLING & FRIENDS TECH MEETUP
BERLIN, 25-27 SEPTEMBER 2017

Building an Apache Sling Rendering Farm

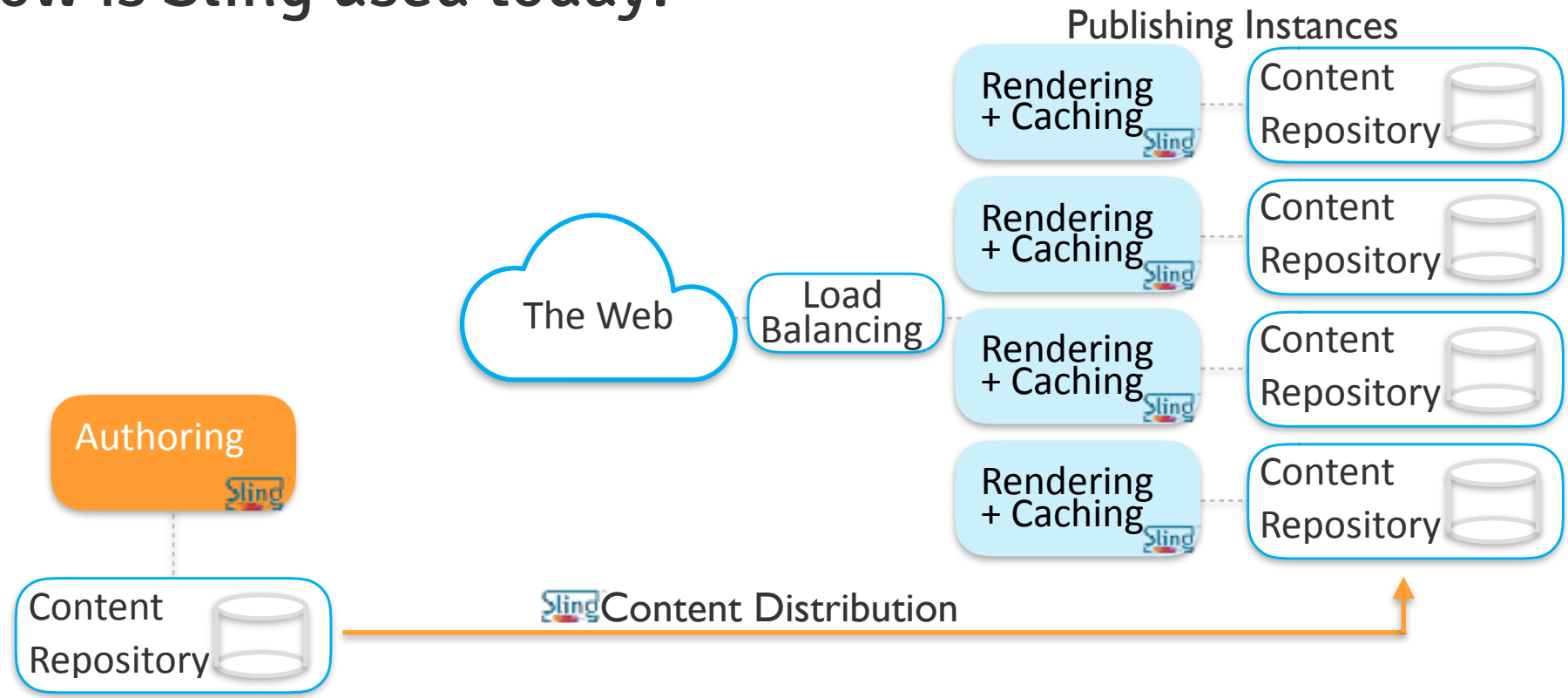
Bertrand Delacretaz
@bdelacretaz
Sling committer and PMC member
Principal Scientist, Adobe AEM team

slides revision 2017-09-25

What are we building?

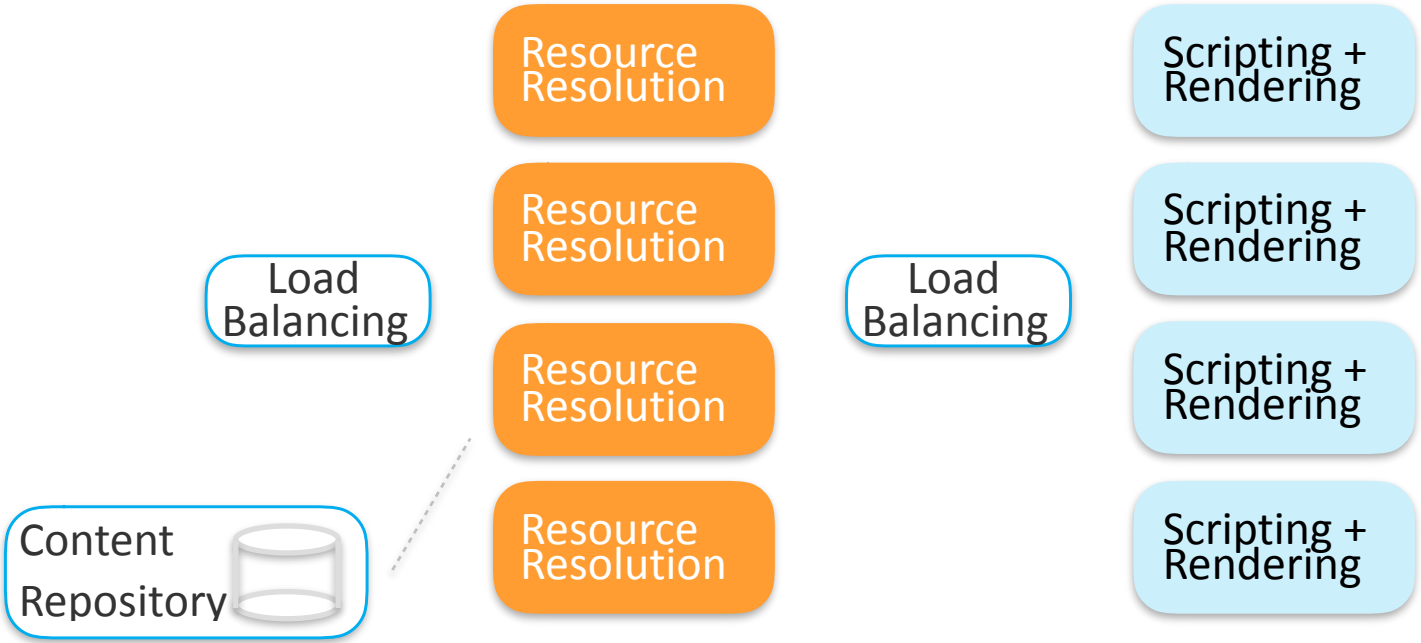
setting the stage

How is Sling used today?



Sling instances dedicated to single tenants or "friendly" tenants.

A Massive Sling Rendering/Processing Farm?

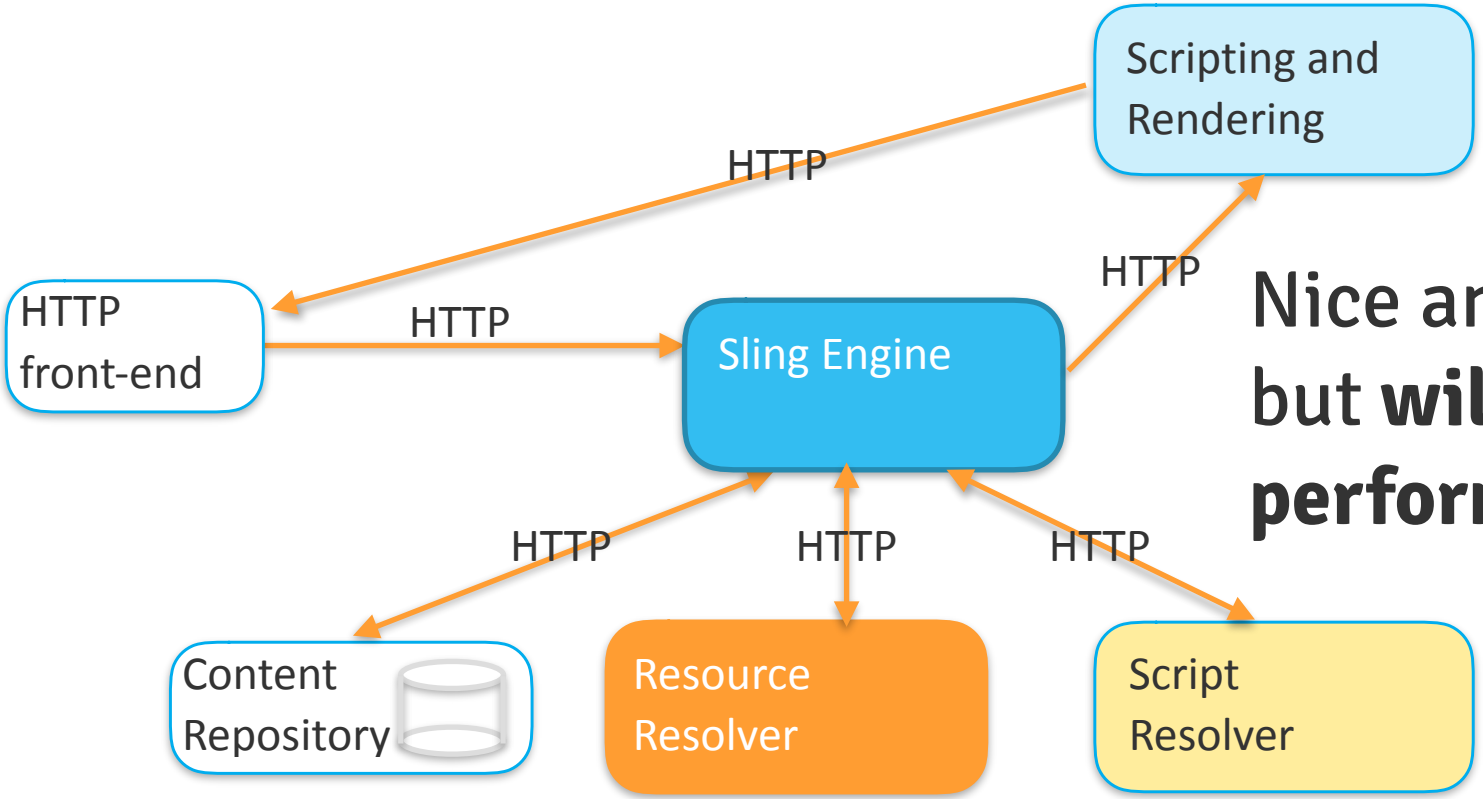


Elastic scaling at each stage
Multiple developers ("tenants") see their own world only

Federated Services

This 2017 after all

Microservices!



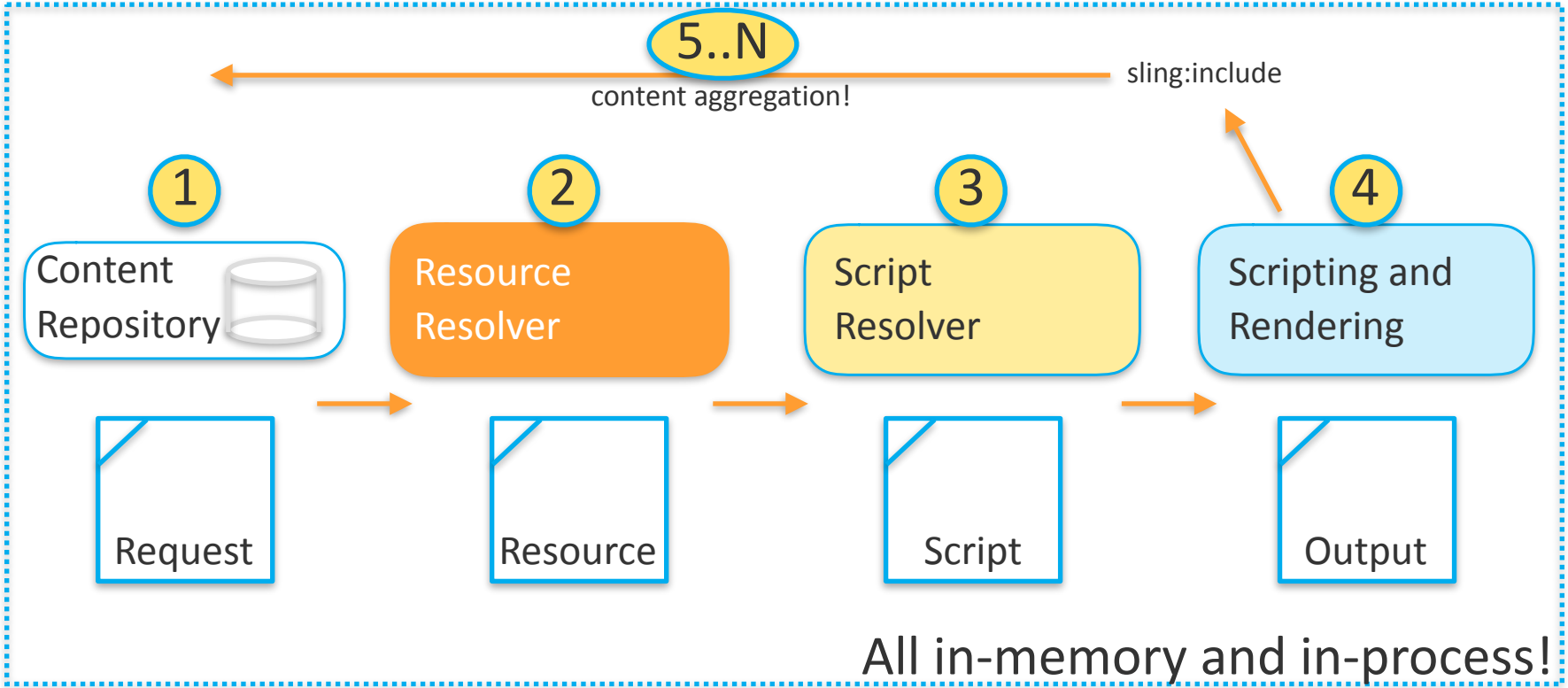
Nice and trendy, but will that perform?

Each component is an independent HTTP-based service, aka “religious microservices”

The Sling Pipeline

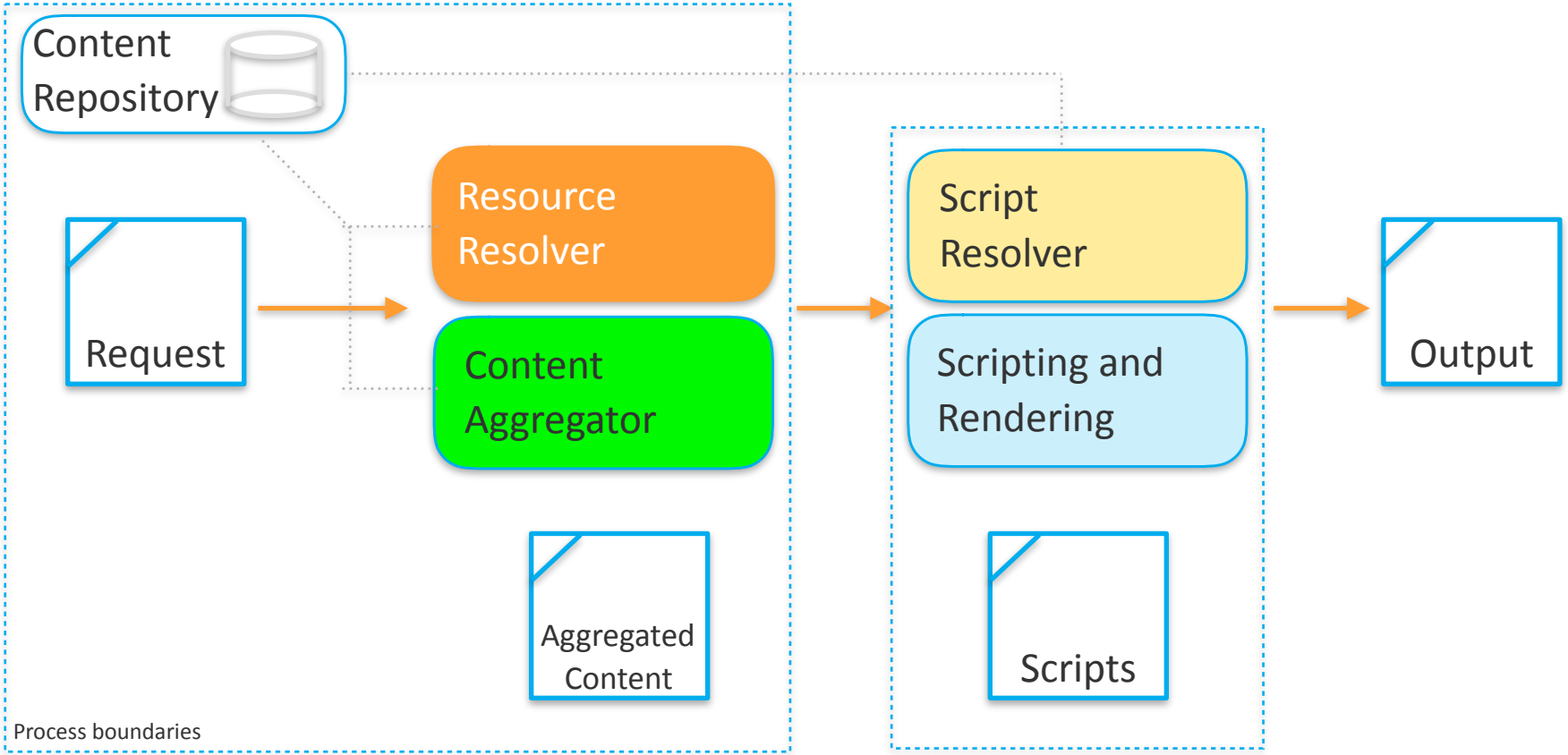
Faithfully serving requests since 2007!

Sling Request Processing Pipeline



Conceptually, the request hits the repository first, to get the Resource.
Scripts and Servlets are equivalent, considering scripts only here.

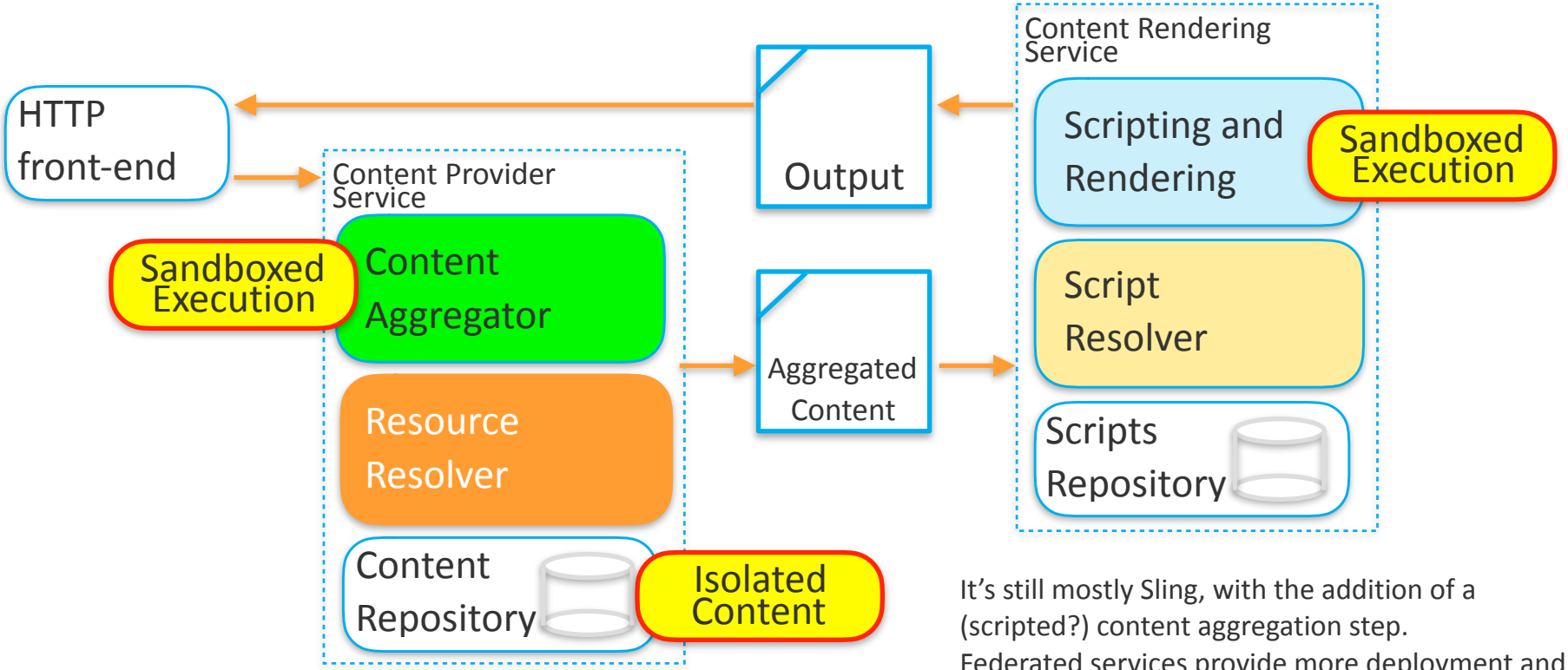
Federated Services Friendly?



Reasonably Federated?

Can we get isolation AND performance?

Reasonably Federated Sling Rendering Farm?

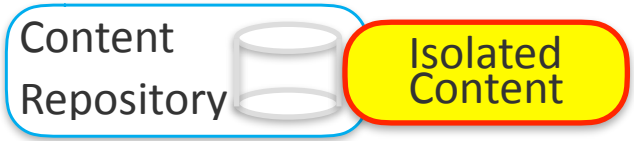


It's still mostly Sling, with the addition of a (scripted?) content aggregation step.
Federated services provide more deployment and scaling options.

Sandboxing & Isolation

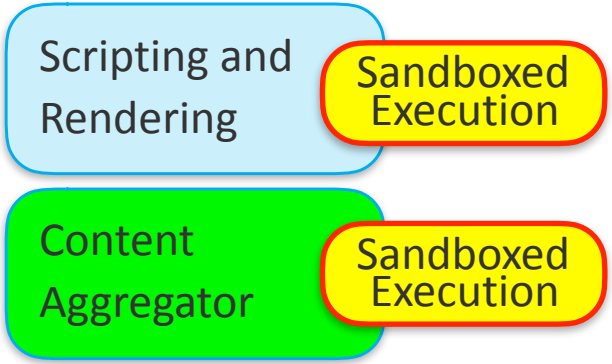
How?

Sandboxing & Isolation?



Repository Access Control can work but require a *dynamic search path* in Sling, see our experiments. Impacts caching, and mapping of incoming to resource paths is needed. Tried and tested.

Repository jails look possible with probable impact on Sling internals. Same with **multiple *SlingRepository* services**. New and more like a blacklist.

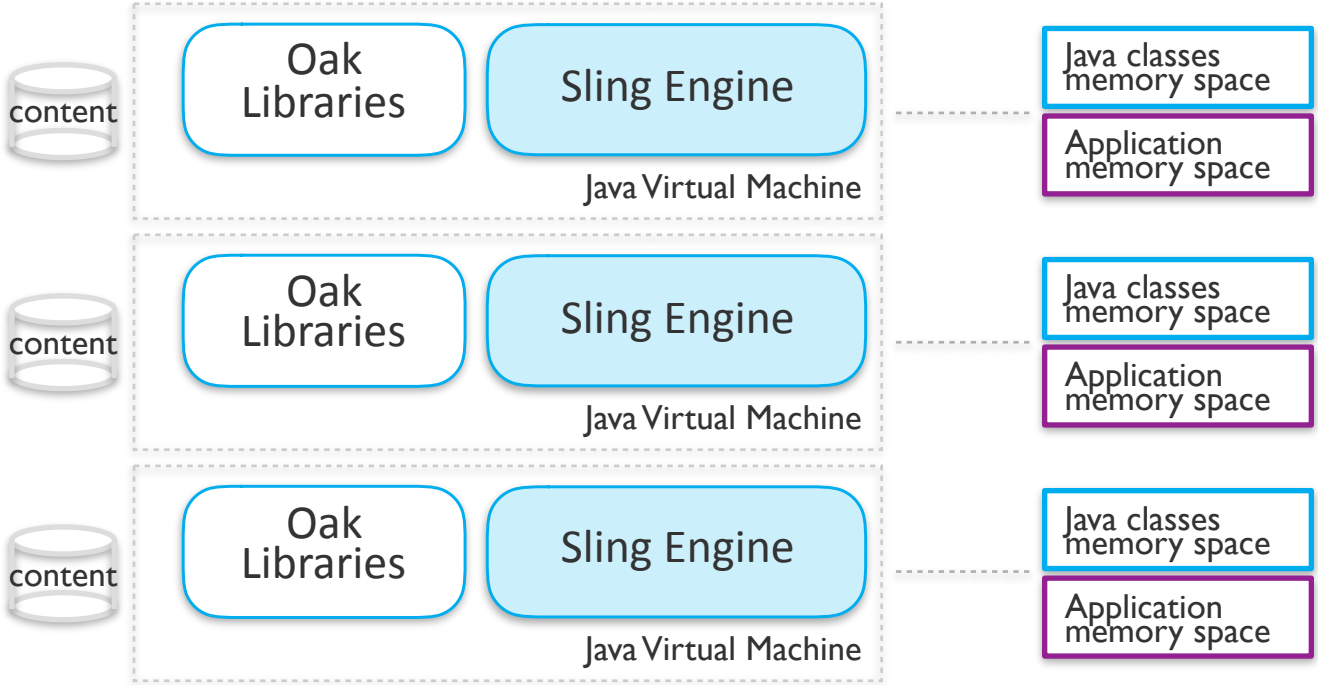


Custom, restricted languages are the safest? HTL (Use-API?), Handlebars?

Sandboxing Nashorn (JavaScript) looks possible but not ideal, see our experiments.

Sandboxing Java is *not realistic*- IBM canceled multi tenant JVM project for example.

But it's a VM, right?



Perfect isolation!

But suboptimal use of resources!
(and containers wouldn't help)

Sandboxing scripting languages?

Java classes & services

OS Resources

Infinite Loops

Memory Usage?

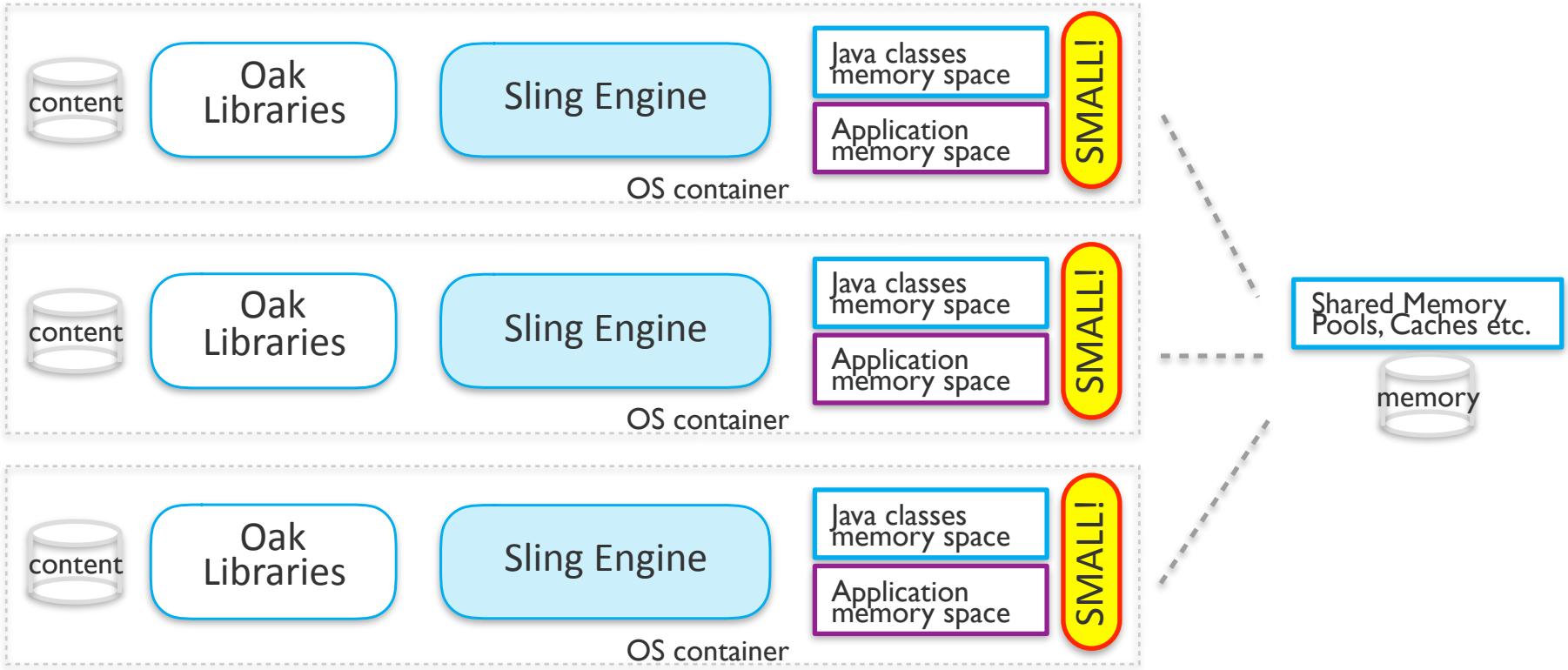
```
<%
var length = 0;
if (request.getRequestParameter("file") != null) {
    var file = null;
    // store file
    var reqPara = request.getRequestParameter("file");
    var is = reqPara.getInputStream();
    file = Packages.java.io.File.createTempFile("posttest", ".txt");
    var fout = new Packages.java.io.FileOutputStream(file);
    var c;
    while ((c = is.read()) != -1) {
        fout.write(c);
    }
    fout.close();

    // read length
    length = file.length();
}
%>
```

Many things need to be limited.

Whitelist approach is much safer -> custom languages?

HTL inherently sandboxed, except its Use-objects

Containers?

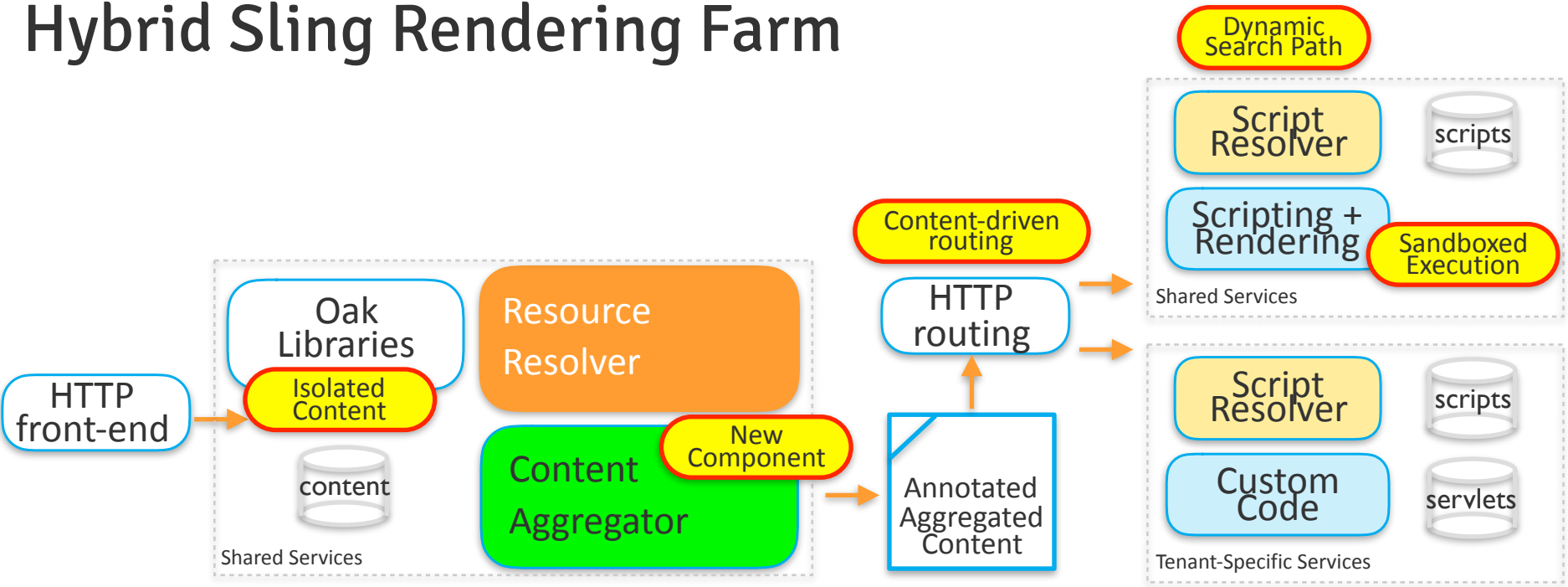


Same problem as multiple JVMs

Sharing caches, compiled scripts etc. can be a pragmatic solution.

What do we do?

Hybrid Sling Rendering Farm



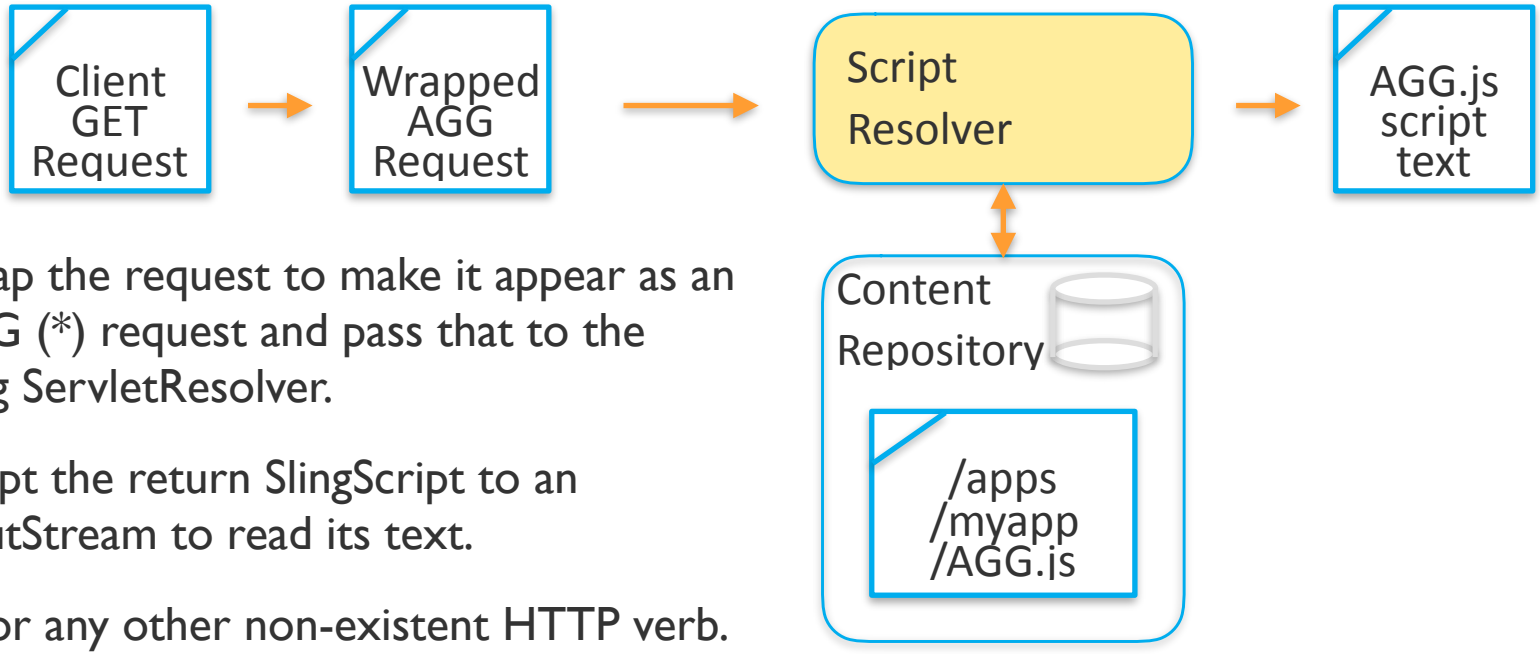
Provides the flexibility of Sling via tenant-specific services and dynamic routing.
Uses shared services for the common parts.
Allows for billable options depending on the actual routing.

Experiments

building blocks that might be reusable

Resolving new types of scripts

Experiment



Wrap the request to make it appear as an AGG (*) request and pass that to the Sling ServletResolver.

Adapt the return SlingScript to an InputStream to read its text.

(*) or any other non-existent HTTP verb.

Code at https://github.com/bdelacretaz/sling-adaptto-2017 (ContentBVP.java)

Resolving a SLING-CONTENT script

Experiment
adaptTo() Bonus Points!

```
String getAggregatorScript(SlingHttpServletRequest r) {
  String result = null;
  Servlet s =
    servletResolver.resolveServlet(
      new ChangeMethodRequestWrapper(r, "SLING-CONTENT"));
  if(s instanceof SlingScript) {
    InputStream is = ((SlingScript)s).getScriptResource()
      .adaptTo(InputStream.class);  }
    if(is != null) {
      result = IOUtils.toString(is)
    }
  }
  return result;
}
```

Code at https://github.com/bdelacretaz/sling-adaptto-2017 (ContentBVP.java)

Content Aggregation with Sling Query

Experiment

```
var $ = Packages.org.apache.sling.query.SlingQuery.$
var SearchStrategy =
Packages.org.apache.sling.query.api.SearchStrategy
var resourceResolver = resource.getResourceResolver()

var result = {
  siblings : $(resource).siblings(),

  rootChildren : $(resource).parents().last().children(),

  queryResult :
    $(resourceResolver)
    .searchStrategy(SearchStrategy.QUERY)
    .find("nt:base[title=foo]")
}
```

Used in a BindingsValuesProvider?
Or in a custom json renderer servlet which runs this script.
Inherently sandboxed due to custom language.
https://sling.apache.org/documentation/bundles/sling-query.html

Dynamic scripts/servlet search path

Experiment

```
if(dynamicServletResolver.canResolve(resource)) {
  servlet = dynamicServletResolver.resolveServlet(request);
} else {
  …existing resolver code
}
```

A fairly simple change to the SlingServletResolver - should evolve into a real extension point if desired, and probably get the request as well.

Tested in SLING-4386 - another multitenant experiment which provides tenant-specific scripts but no real isolation.

Currently requires disabling the servlet resolution cache.

Nashorn (JavaScript) sandboxing (Java Delight)

Experiment

```
NashornSandbox {
  allow(final Class<?> clazz);
  injectGlobalVariable(String variableName, Object object);
  setMaxCPUTime(long limitMsec);

  Object eval(final String javaScriptCode);

  allowPrintFunctions(boolean v);
  allowReadFunctions(boolean v);
  ...more allow functions

  // $ARG, $ENV, $EXEC...
  allowGlobalsObjects(final boolean v);
}
```

Uses Nashorn's **ClassFilter** to block Java classes

Sandboxing rewrites standard methods + user code- > **blacklisting,** not ideal

https://github.com/javadelight/delight-nashorn-sandbox (Java Delight Suite)

CODA
where to now?

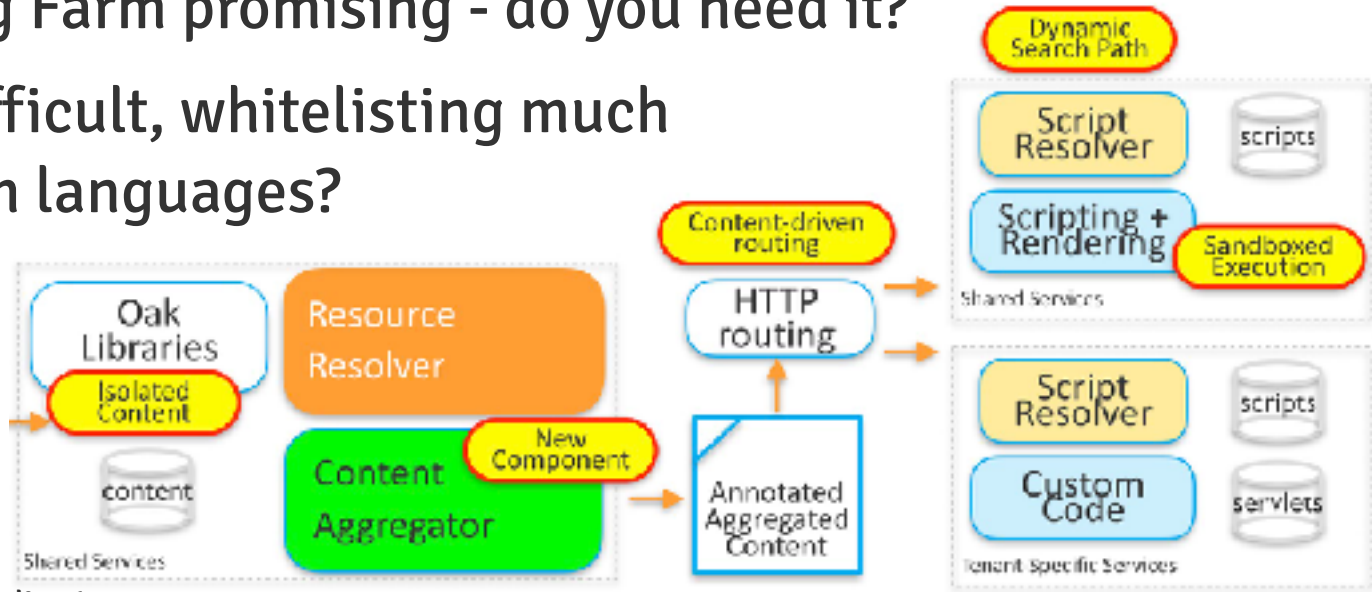
CODA

In-memory nature of Sling is an important differentiator, in good and bad ways!

Hybrid Rendering Farm promising - do you need it?

Sandboxing is difficult, whitelisting much preferred, custom languages?

Reusable experiments?



Thank you for attending!
I'm Bertrand Delacretaz (@bdelacretaz)