

adaptTo()

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

Sling Deployment Revisited

Dominik Süß, Adobe

Karl Pauls, Adobe

Where is the Problem?



SLING & AEM DEPLOYMENTS TODAY

- Patchwork of Mechanisms
- Not deterministic
- Error Prone (*human factor*)
- Inefficient

*Painful to support & debug
when failing!!!*

Patchwork of Mechanisms

NO TIME
FOR DETAILS

- OSGi Installer
 - FS Installer
 - JCR Installer
 - Launchpad / quickstart
 - PackageTransformer
 - Config Installer
- Webconsole
- *Package Manager (AEM)*



- Varying behavior
 - Esp. Package vs Bundle
- Massive interaction
- Parallel activity

Not deterministic

NO TIME
FOR DETAILS



- Stateful
- Race conditions by ambiguous dependencies
- Undefined Endstate
- Unverifiable Outcome

!!!!!!

Deployment only declares changes
not the full target state

Error Prone

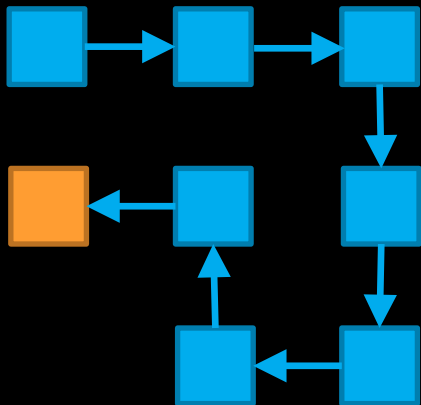
NO TIME
FOR DETAILS



- Manual dependency declaration for packages
- Filter changes can easily cause left-overs
- Instance state alters install sequence of deployment units (*pre-satisfied dependencies*)
- Altering sequence prevents reliable testing

Inefficient

NO TIME
FOR DETAILS



- Unoptimized Instruction Flow
- Retry until success
- Rewiring of OSGi Bundles (Config & DS)
- Roundtrips through nested deployment units

The Challenge

The background of the slide is a dark, grayscale image. On the left, there is a lit candle in a metal holder. To the right of the candle are three light bulbs of different types: a standard incandescent bulb, a compact fluorescent bulb (CFL), and a modern LED bulb. The bulbs are arranged in a row, and their reflections are visible on the surface they are sitting on. The text "Backward Compatible Evolution" is centered over the image, with "instead of" in a smaller font below it, and "Revolutionary Experience Change" centered below that.

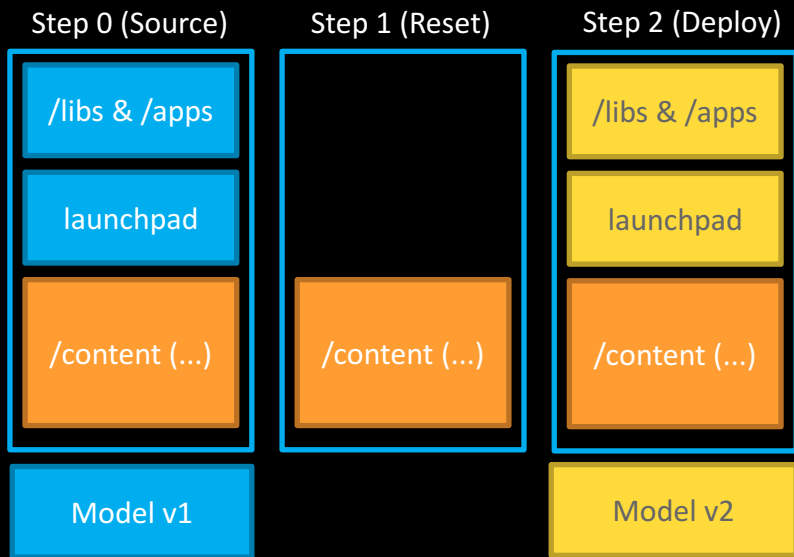
Backward Compatible Evolution
instead of
Revolutionary Experience Change

Vision

High Level Goals

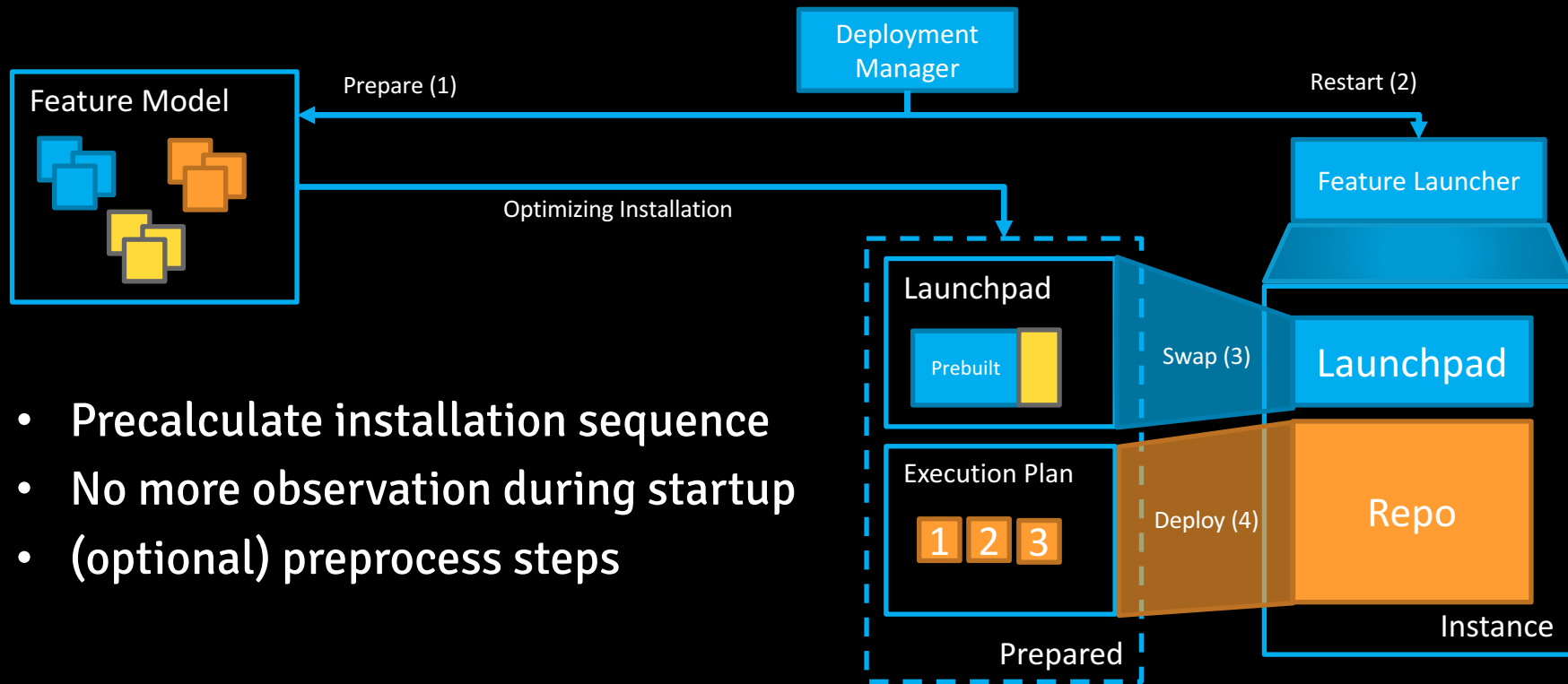
Deterministic Deployments	<ul style="list-style-type: none">• No more race conditions• Predictable• Reproducible• No more state handling
Prepared Deployments	<ul style="list-style-type: none">• Reduce install startup times• Precalculate & identify failures ahead of deployment
Conflictless Deliveries	<ul style="list-style-type: none">• Validated consistency ahead of deployment• Reduce human effort to declare dependencies
Composed Instances	<ul style="list-style-type: none">• Reduce to ONE mechanism (install, patching, upgrades, configuration)• Model reflecting layering of instance (Vendor, Integrator, Operator)

Deterministic Deployments



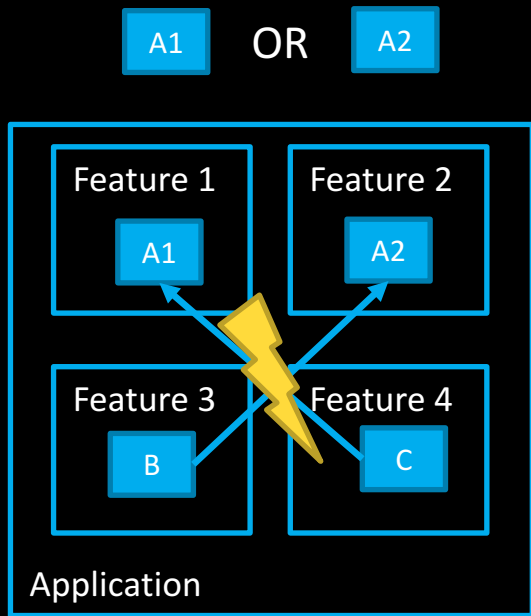
- **Complete Dependency Tree**
 - Close gap between Appcontent & Java
- **Full application state in feature model**
- **Rebuild application state each time**

Prepared Deployments

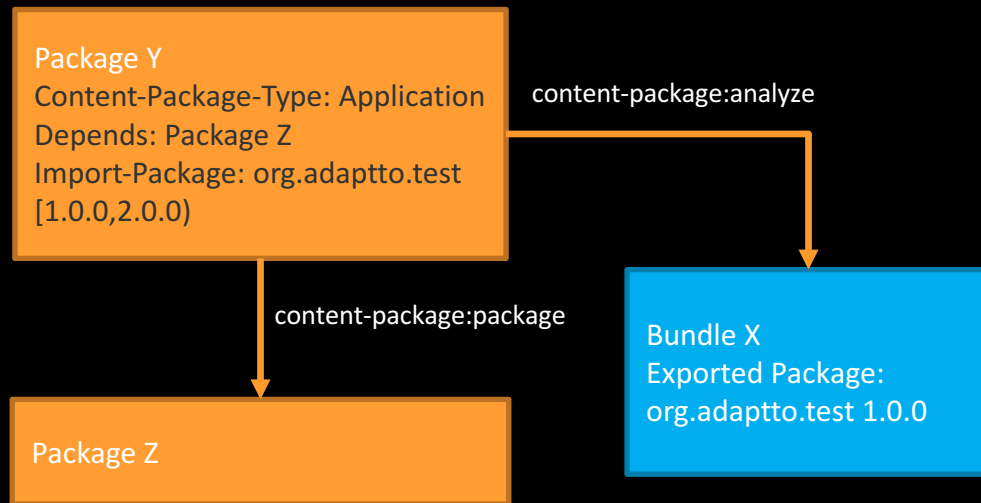


- Precalculate installation sequence
- No more observation during startup
- (optional) preprocess steps

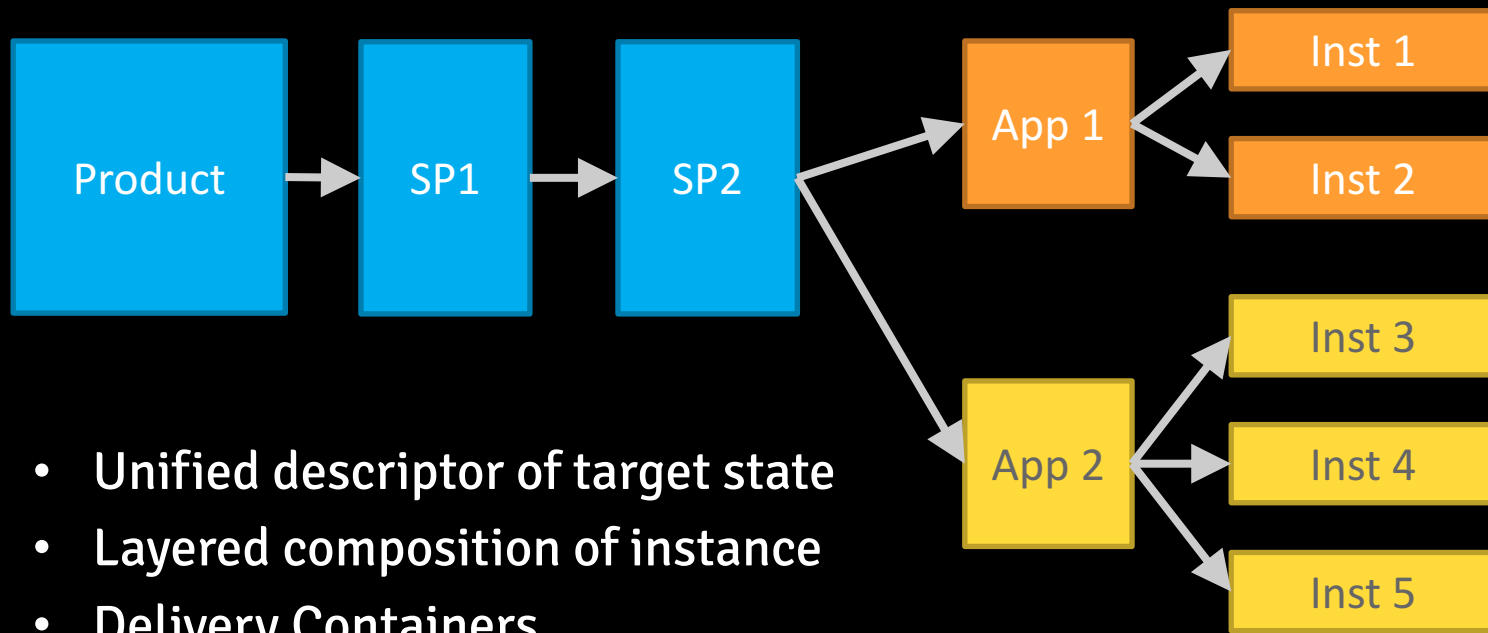
Conflictless Deliveries



- Validation of feature model
- Improved dependency metadata in vlt
- Validation of deployment units



Composed Instances

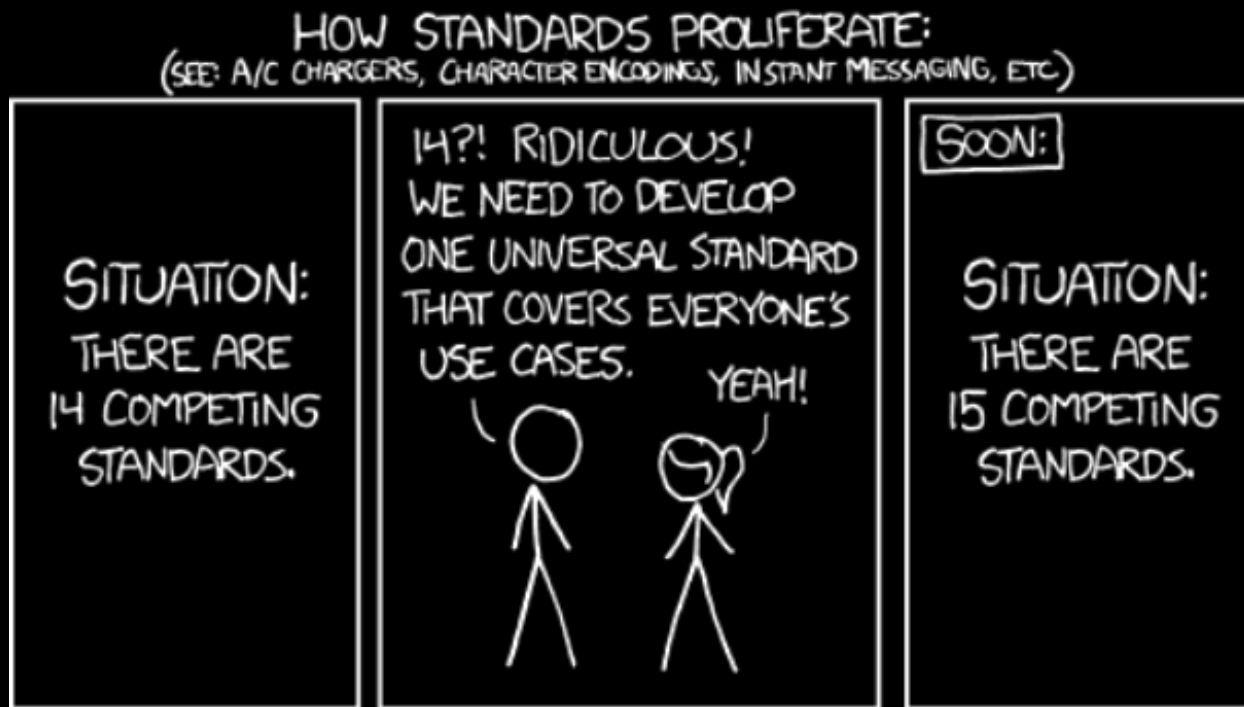


The Experience

Isolated changes in ops experience:

- Application composition (*prepare deployment*) replaces coordinated installation sequence
- All installation steps are reversible (*just remove model*)
- Unify install & configuration experiences (*eliminate variations*)

What do we do?



<https://xkcd.com/927/>

Introducing Features

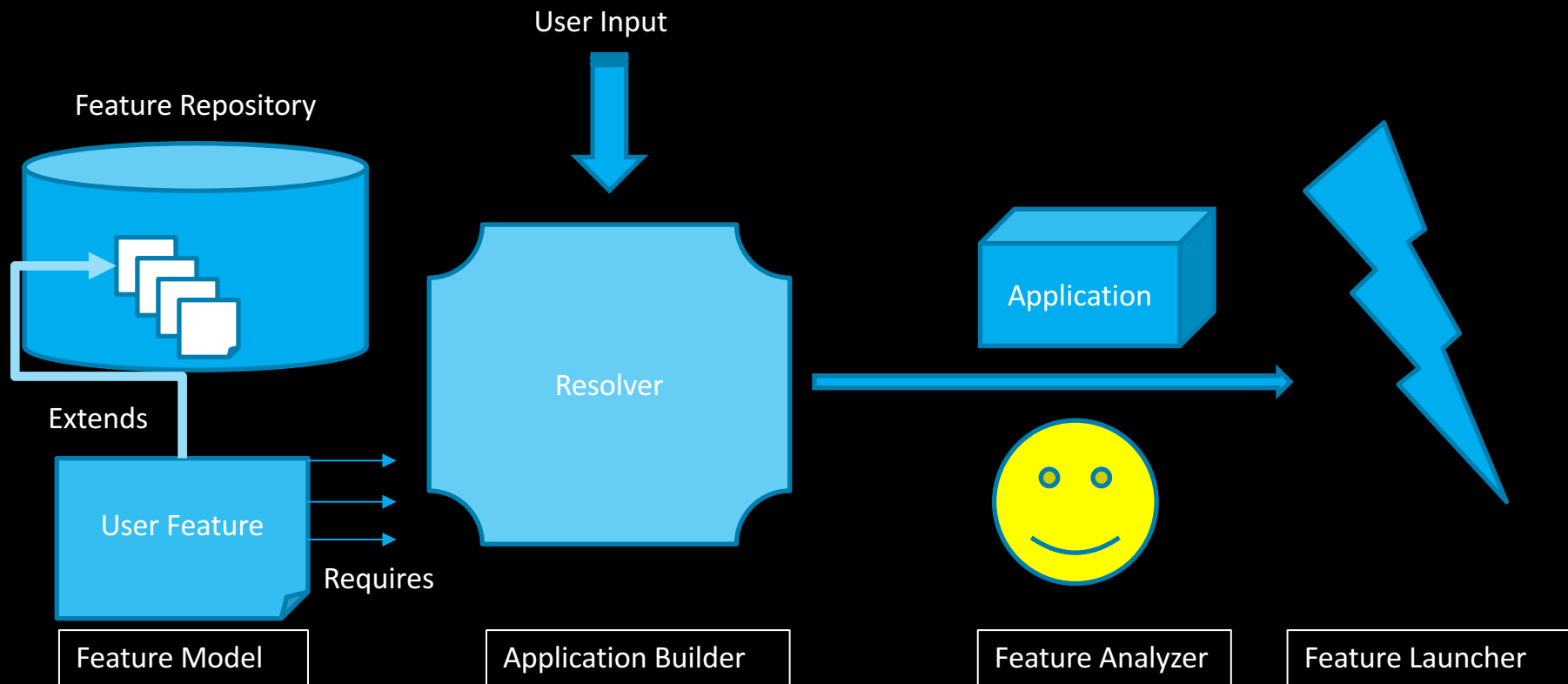


* Name:
TBD

Sling Features*

- Well defined provisioning model with named and versioned features
- Requirement/Capability based as well as include based dependencies
- Resolver based composition and extensible analysis framework
- Sling focused launcher for immutable deployments

High level flow



Feature Model

```
{
  "id" : "org.apache.sling/my.app/1.0",
  "includes" : [ {
    "id" : "org.apache.sling/sling/9",
    "removals" : {
      "configurations" : [],
      "bundles": [],
      "framework-properties" : [] }},
  "requirements" : [{
    "namespace" : "osgi.contract",
    "directives" : {"filter" :
      "(&(osgi.contract=JavaServlet(version=3.1))" }},
  "capabilities" : [{
    "namespace" : "osgi.implementation",
    "attributes" : {
      "osgi.implementation" : "osgi.http",
      "version:Version" : "1.1" }},
```

```
"bundles" : {
  "1" : [
    "org.apache.sling/security-server/2.2.0",
    "org.apache.sling/application-bundle/2.0.0",
    "org.apache.sling/another-bundle/2.1.0" ],
  "2" : [
    "org.apache.sling/foo-xyz/1.2.3" ]},
  "configurations" {
    "my.pid" {
      "foo" : 5,
      "bar" : "test",
      "number:Integer" : 7
    },
    "my.factory.pid~name" {
      "a.value" : "yeah"
    },
    "repointit:Text|true" : "...",
    "content-packages:ARTIFACTS|false" : [...]
```

A dramatic photograph of a space shuttle launching. The shuttle is silhouetted against a bright, orange and yellow sky filled with clouds. It is ascending vertically, leaving a large, billowing plume of white smoke and fire behind it. The launch pad structure is visible at the bottom, with various scaffolding and support beams. The overall scene conveys a sense of power and achievement.

DEMO TIME

What is already done?

- Feature Model Draft
 - <https://svn.apache.org/repos/asf/sling/whiteboard/cziegeler/feature/readme.md>
- Vault Metadata Extensions
- Vault Improvements
- Maven tooling for content-package (*ASF contribution*)
- Feature Launcher Prototype

What YOU can do!

- **Hackathon: Play around and provide feedback!**
 - <https://github.com/DominikSuess/adaptto2017-demo>
 - Ideas:
 - Get continuous deployments running with feature launcher
 - Reset application state using federate repo (ODT POC)
 - Work on Deployment Manager
 - Introduce precalculation (OSGi & Packages)
 - Check out content-package & tooling improvements

HAPPY HACKING