

**adaptTo()**

EUROPE'S LEADING AEM DEVELOPER CONFERENCE

27<sup>th</sup> – 29<sup>th</sup> SEPTEMBER 2021

Evolving API for AEM as a Cloudservice

*when push-release meets backward compatibility*

Dominik Süß, Adobe

# Setting the SCOPE

- JAVA API
- OSGi Configuration

*Similar challenges for Content, Markup & Clientlibraries but solution strategies presented are mostly focused on above.*

The top half of the image shows a close-up of an open old book with yellowed pages. A pair of round-rimmed glasses with thin black frames is resting on the pages. To the right, the spine and cover of another old book with a dark, worn leather binding are visible.

THE PAST – when “everything” was better..

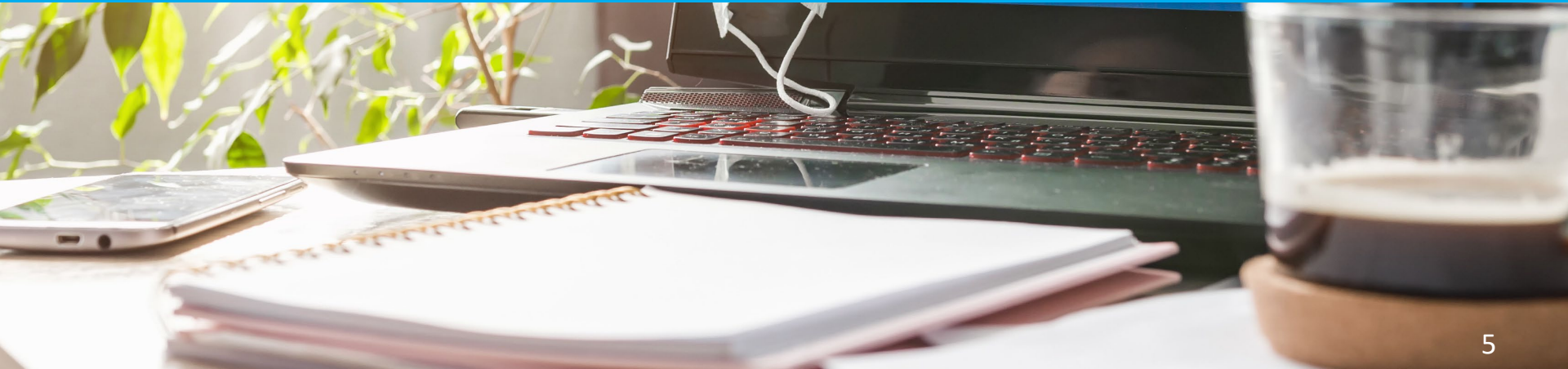


# Annual Releases & Big Projects

- Refactorings minor part of big projects
- Timing up to customer
- Migration of breaking changes in an isolated timewindow



# THE PRESENT



# Changed Landscape in the Cloud

- Continuous releases
- Push upgrades at any time
- Continuous innovation

**BUT**

- Must not break consumer on rollout
- No removal before consumption ended (*Draining*)

# Modularization & Regions

- Regions: Internal vs Global \*
- Product is Layered (Core > Modules > Customer)
- Decoupling of Modules
  - API Isolation
  - Minimize Implementation Dependencies

Design Goal: Lean Surface to keep adoption effort low

\* [Adaptto 2020 Talk about Regions](#)

# AEM Addons in the Cloud

- Push Release
- Layered on top of core product *(strict dependency layering)*
- Isolated in Feature Archives
  - Available as installable or local development
- Individual Lifecycles



- Challenge for Engineering
  - New Options require new API -> Increased surface
  - Legacy increases maintenance Overhead.
- Solutions
  - Deprecation and side-by-side Replacements  
([Stranglerfig Application Pattern](#))

- Challenge for Engineering
  - **valid** *(for implementation)* **!=** **valid** *(in integrated system)*
  - Every new option increases complexity & maintenance overhead
- Solutions
  - Shift values in Environment Variables
  - Sling API Region - Configuration Validation

## ■ API Deprecation Example

```

651     {
652         "name": "org.json",
653         "deprecated": {
654             "msg": "The org.json API is not maintained anymore. Please use an alternative. The Apache Johnzon implementation of javax.json is recommended.",
655             "since": "2021-04-30",
656             "for-removal": "2021-12-31"
657         }
658     },

```

## ■ OSGI Config Enforcement Example

```

2944 "configuration-api:JSON|false": {
2945     "region": "INTERNAL",
2946     "mode": "LENIENT",
2947     "configurations": {
2948         "org.apache.felix.eventadmin.impl.EventAdmin": {
2949             "mode": "LENIENT",
2950             "since": "2021-04-30",
2951             "enforce-on": "2021-07-31",
2952             "title": "Apache Felix Event Admin Implementation",
2953             "description": "Configuration for the Apache Felix Event Admin Implementation. This configuration overwrites configuration defined in framework properties",
2954             "properties": {
2955                 "org.apache.felix.eventadmin.ThreadPoolSize": {
2956                     "title": "Thread Pool Size",
2957                     "description": "The size of the thread pool used for event delivery. The default value is 20. Increase in case of a large amount of events. A value c",
2958                     "type": "INTEGER",

```

# Compatibility & Warnings (1/2)

Engineering  
Perspective

- Declare Deprecation & Target Removal Date
- Tracking of consumer usage
- Breaking change only without remaining consumers
  - Declarative exceptionlist to prevent accidental breaking changes

## Compatibility & Warnings (2/2)

Consumer  
Perspective

- Deprecation markers to inform about planned removals or constraints
  - [Deprecated APIs](#)
  - [Deprecated Configs](#)
- Warnings in pipelines & local analysers
  - Source of truth:
    - [AEM SDK Feature model](#) (*check .slingsgfeature artifact*)

# DEMO - API jar & local analysers



- Strict policies checked in Pipeline (*API & Content*)
- Compatibility Compliance Rating Model
- Approval & Exception only where no impact can be proven
- Explicit Exception handling in pipeline linking to BC evaluation
- Challenge: Breaking changes in 3rd party bundles.

# Automation & Release Validation of AEM

Engineering  
Perspective

- Sprouts: CI Framework for validating unreleased code
  - Initially for validating internal modules
  - Sprouts for full CI testing of open source projects
- Evergreen
  - Full IT coverage on each PR on main release branch as Quality gate (incl. BC policy validation)
- Release Validator
  - Exposing Release Candidates to adhoc clones of customer environments
  - Blackbox tests to find scenarios deviating from IT scenarios



A close-up photograph of a person's finger pressing a circular button. The button has a glowing cyan ring and the word "FUTURE" in white capital letters on a dark background. The background is a dark, textured surface.

THE FUTURE – into unknown territory...

- Innovation will **require deprecation** of legacy
  - *... or we'll all end in a horrible maintenance nightmare...*
- Deprecation will require **adoption by consumers**
- Migration must be minimal **effort** for adoption

- 100% autonomous adoption unrealistic
- Prevention of new occurrences
- Focus on ease of adoption (*low effort, clean migration paths*)
- Proactive communication where necessary

A woman with vibrant red curly hair is shown from the chest up. She is wearing a dark blue denim shirt and has a ring on her finger. She is pointing her right index finger directly at the viewer. The background is a solid, bright yellow.

CALL TO ACTION – your turn!

# Continuous adoption & contribution

- We rely on you!
- React on new deprecations
- Adopt changes
- Help us improve the flow (*Open Source Projects*)

# Q&A

- Adaptto 2020 Talk about Regions
  - <https://adapt.to/2020/en/schedule/future-proof-your-applications-with-api-regions.html>
- Sling Feature Archives
  - <https://github.com/apache/sling-org-apache-sling-feature/blob/master/docs/feature-archives.md>
- Stranglerfig Application Pattern
  - <https://martinfowler.com/bliki/StranglerFigApplication.html>
- Environment Variables in AEMaaCS
  - <https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/deploying/configuring-osgi.html?lang=en>
- Sling API Region - Configuration Validation
  - <https://github.com/apache/sling-org-apache-sling-feature-extension-apiregions/blob/master/docs/api-regions.md#configurations>
- AEM Deprecated APIs:
  - <https://experienceleague.adobe.com/docs/experience-manager-cloud-service/release-notes/deprecated-apis.html>
- AEM Deprecated OSGi Config:
  - <https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/deploying/osgi-configuration-api.html>
- AEM SDK Feature model
  - <https://mvnrepository.com/artifact/com.adobe.aem/aem-sdk-api>

# Appendix



# OSGI Config: Mode Definitive

Necessary enforcement for damaging  
Configuration w/o breaking build

Fallback on default instead of breaking

**Mode: DEFINITIVE &  
SILENT\_DEFINITIVE**

## Example

- Reconfiguring loggers messes with log monitoring integration and DevOps (blocking trackdown in case of outage)

```
"org.apache.sling.commons.log.LogManager.factory.config":{
  "properties":{
    "org.apache.sling.commons.log.level":{
      "title":"Log Level",
      "description":"Root Logger log level setting.",
      "options":[
        {
          "title":"Info",
          "value":"INFO"
        },
        {
          "title":"Debug",
          "value":"DEBUG"
        },
        {
          "title":"Trace",
          "value":"TRACE"
        },
        {
          "title":"Info",
          "value":"info"
        },
        {
          "title":"Debug",
          "value":"debug"
        },
        {
          "title":"Trace",
          "value":"trace"
        }
      ],
      "default":"INFO",
      "mode":"SILENT_DEFINITIVE"
    },
    "org.apache.sling.commons.log.names":{
      "title":"Logger",
      "description":"The logger names applicable for this logger configuration. Each logger name applies for any child unless there is a different configuration for org.apache.sling.commons.",
      "cardinality":-1
    },
    "org.apache.sling.commons.log.file":{
      "title":"Log File",
      "description":"The name and path of the log file. If this is empty, logging goes to standard output (the console)"
    },
    "org.apache.sling.commons.log.additivity":{
      "title":"Additivity",
      "description":"If set to false then logs from these loggers would not be sent to any appender attached higher in the hierarchy"
      "type":"BOOLEAN"
    }
  },
  "mode":"DEFINITIVE",
  "internal-names":[
    "access",
    "audit",
    "auditlog",
    "request",
    "history",
    "queryrecorder"
  ]
}
```

# Push upgrades & content

- Application content switched by remounting
- Mutable content changes AFTER deployment
- AEM product rollout avoiding mutable content changes for easy rollback
- Strong recommendation to follow the same pattern for customers
- Use [repointit](#) for declarative, atomic, backward compatible changes