adaptTo()

EUROPE'S LEADING AEM DEVELOPER CONFERENCE

28th – 30th SEPTEMBER 2020

AEM aa CS
^ ^
Mind the Gap

About Us

Mr Süß
Adobe, JCR Committer

A guy that loves developing AEM

Mr Reusser
headwire.com, Inc

A guy that loves developing with AEM

The Challenge

- ❑ Move into the cloud
- ❑ Scale with the needs
- ❑ Modularize and separate concerns
- ❑ Keep and embrace existing ecosystem
- ❑ Consolidate deployment models

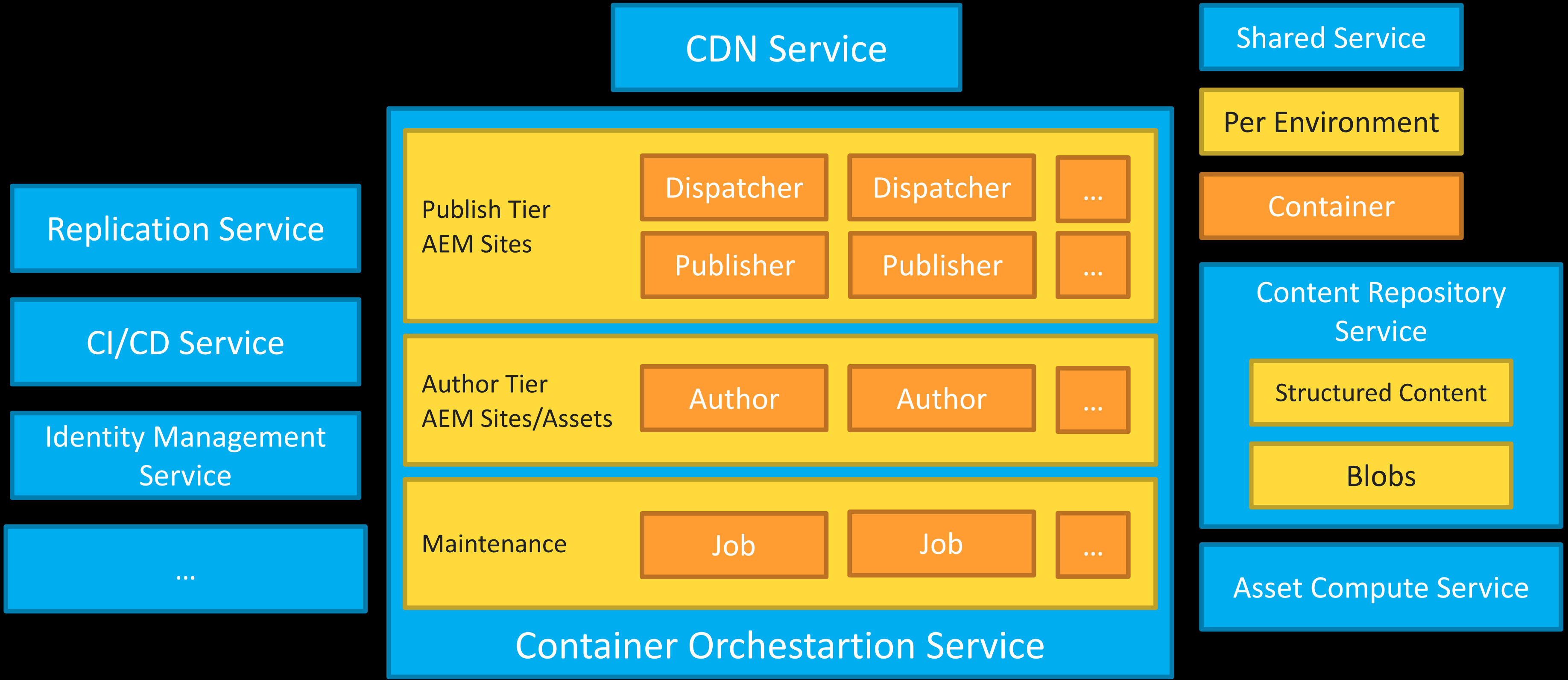
The Solution: AEM as a Cloud Service

- ✓ Moved to the cloud
- ✓ Optimized resource consumption
- ✓ AEM internals tuned for cloud
- ✓ Mostly the same APIs, AEM feels the same
- ✓ Scalable deployment model ready to serve content

Then and Now

- **Before:** You wire & control each individual instance
- **Now:** Well-orchestrated Cloud Deployment, you only define the desired state of the service via SCM

AEM aa CS Topology



CDN Service

Replication Service

CI/CD Service

Identity Management Service

…

Publish Tier
AEM Sites

Dispatcher | Dispatcher | …

Publisher | Publisher | …

Author Tier
AEM Sites/Assets

Author | Author | …

Maintenance

Job | Job | …

Container Orchestartion Service

Shared Service

Per Environment

Container

Content Repository Service

Structured Content

Blobs

Asset Compute Service

adaptTo()

How this affects YOU!

A simplistic view

- **Same:**
  - It's still AEM
  - AEM is still with Us
- **Different:**
  - Before: you chose everything
  - Now: Adobe chooses some things for you

Some Gaps

- Things you might want to know -

Developing & Testing

Development Flow

What?	Where
Development	Local
Initial Functional Testing	Local
Integrated (functional) Validation	Cloud (dev)
Nonfunctional Testing	Cloud (dev & stage)
User Acceptance Testing (functional & nonfunctional)	Cloud (stage)

adaptTo()

Debugging & Logging in the Cloud

Development Flow

- Dev Console replacing System Console in Cloud
- Stick with the error.log and split on consumer side
- Use AIO CLI Cloudmanager Plugin to tail

Shell command to tail logs: `aio cloudmanager:tail-log ENVIRONMENTID SERVICE NAME`

adaptTo()

Product Rollouts *(current status)*

Development Flow

- Stage & Prod
  - Continuous product rollouts on latest passed application
- Dev
  - Updated via CM self-service rollout* ***(trigger it!)***
  - Keep your local SDK version close to production

** May be subject to change in the future*

Mutable vs Immutable

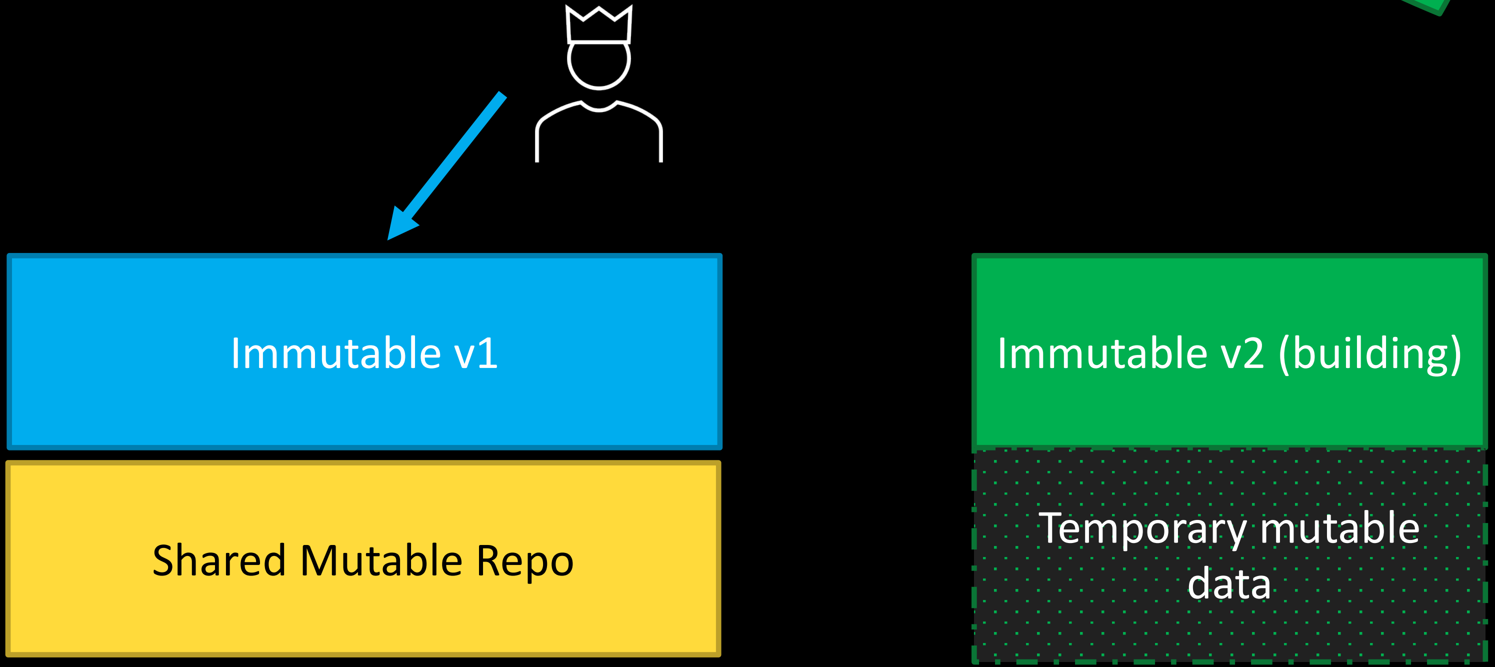
Deployment

Immutable	Mutable
• OSGI Bundles & Config • App Content /libs & /apps	• Everything else in the repository (/content, /conf, /var, /tmp,/etc ..)
Changes only at **deploy time**	• Changes only at **runtime***

* This also applies to mutable content part of the SCM
Packages not embedded in an install.author folder will be distributed to publishers via replication (autoextract)

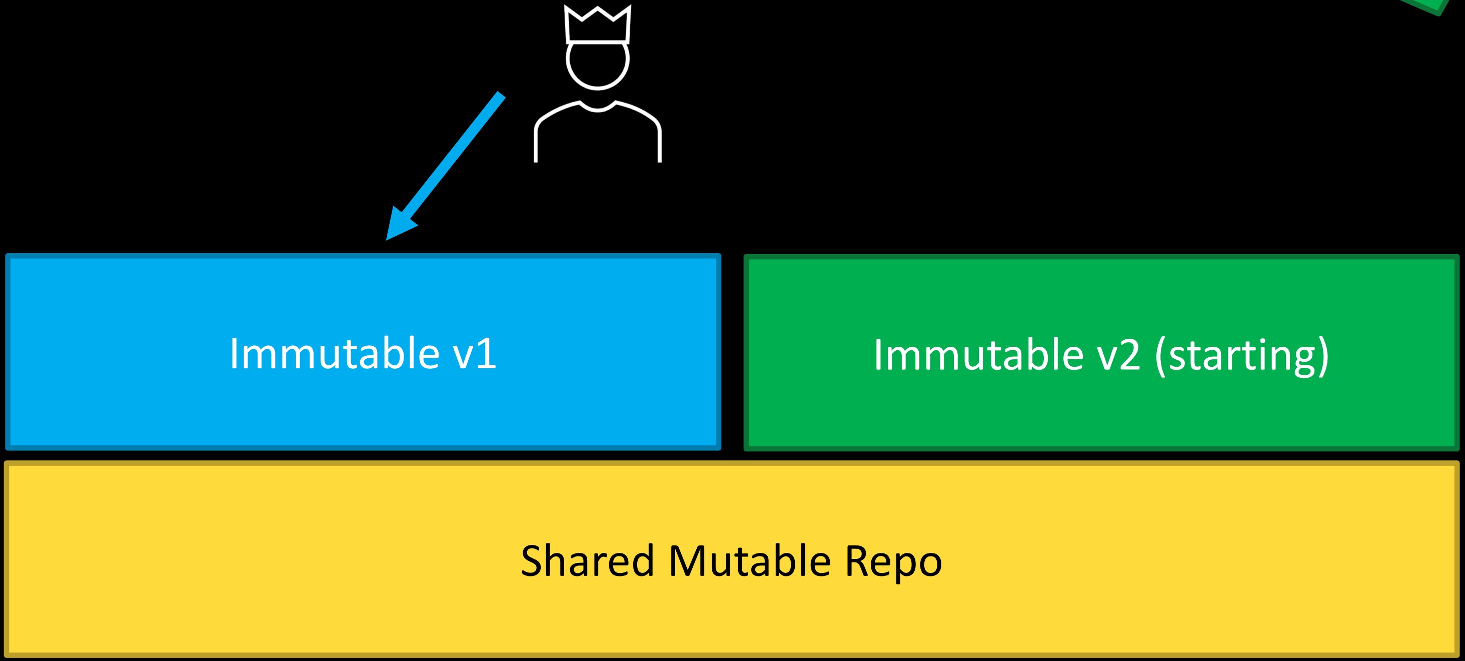
Blue-Green Author-Cluster Deployment

Deployment



Blue-Green Author-Cluster Deployment

Deployment

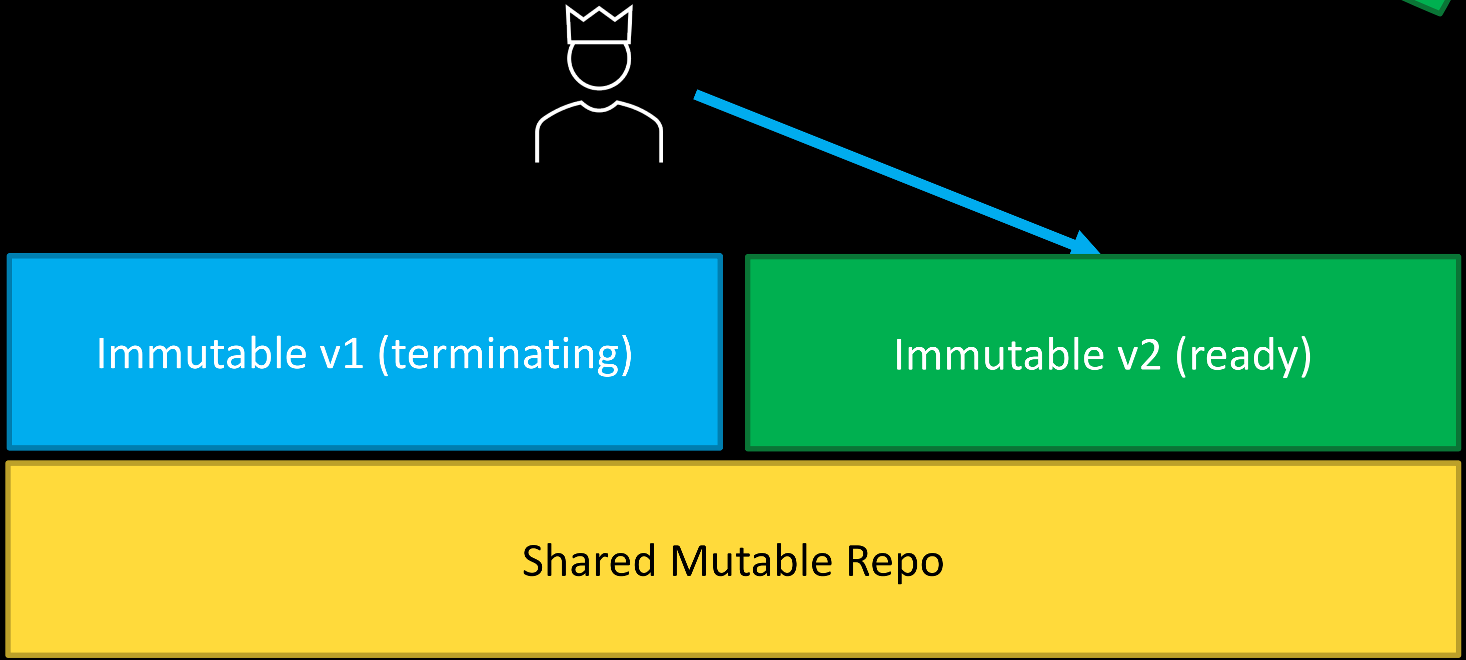

Immutable v1

Immutable v2 (starting)

Shared Mutable Repo

Blue-Green Author-Cluster Deployment

Deployment



Immutable v1 (terminating)

Immutable v2 (ready)

Shared Mutable Repo

Blue-green implications

- 1 version serving traffic
- 2 versions concurrently running
- Content **must** stay compatible until switch
- Sling Topology “*sees*” mixed version state
- No rollback of mutable content if deploy is Interrupted (blue keeps serving)

Indexes

Deployment

- Indexes are applied **before** switchover
- You got to name them right* to prevent clash with blue version
- Query Engine handles multiple versions of index definitions in parallel & binds to one serving results

* Adobe Documentation about Indexing

Mutable Content in AEM aa CS

Deployment

- Packages applied POST switchover
- Not environment (dev/stage/prod) specific
- Packages as last resort – prefer:
  - Repoinit (osgi config)
  - Lazy change after switchover (e.g. via Sling Pipes – project solution)

adaptTo()

Readiness & Liveness

Deployment

- Felix Healthchecks about to replace Sling HC & Felix SR Check
- Talk by Georg Henzler 2019
- **Readiness**: System initialized - ready to serve content
- **Liveness**: System healthy - restart may recover the system if unhealthy

AdaptTo 2019 talk: Felix Health Check

adaptTo()

Files in AEM aa CS

Runtime Behavior

- Filesystem ephemeral & limited
- Binary processing expensive **(I/O, CPU & Memory)**

Options:

- Direct Binary Access (read & write)*
- Asset processing via Asset Compute Service

Open Challenge:

- Externalization of Content Packaging

AdaptTo 2018 lightning talk: Oak Direct Binary Access

Schedulers, Eventing & Jobs

Runtime Behavior

- Instances can come & go at any time
- Oak eventing is asynchronous

No new challenge but more frequent – consider the necessary processing guarantees.

- Handling Schedulers in AEM aa CS (Kaushall Mall)
- Cluster aware coding in AEM (Jörg Hoh)

Replication Implications

Runtime Behavior

- Single replication stream (publishers are equal)
- Replication as single content entry (1 week replay)
- Reduced support of extension hooks (Content transformation etc.)
- Content installation via service-user (permission constraints)

Replication & Deployments

Deployment

- Replication on halt while rolling
- New Golden Publisher catching up
- Switchover when catchup completes

- **Open challenge**
  - **Invalid** content can block pipeline & subsequently block rollout *(deployment is dependent on active replication*)

adaptTo()

Takeaway

Mr Suess (Adobe)

Continuous deployments and a streamlined operational setup finally allow us to rapidly evaluate and roll out optimizations of AEM and validate against real customers without causing disruptions while being closer to the customer implementations.

Mr Reusser (headwire)

It is nice to see AEM evolve at a faster pace! Finally our work is shifting – away from customizing older versions of the AEM monolith – towards adding true value to AEM customers and the ecosystem by leveraging the cloud, micro services, Adobe I/O, NUI workers and Firefly against an always up to date AEM version.

The opinions expressed on this slide are the opinions of the individual person and do not reflect any opinion by Adobe or headwire. Any decisions you take based on this presentation is yours and yours only. Nor Adobe or headwire are responsible for any damages, extra work or other suffering you or your organization may experience. If you do experience any symptoms due to this presentation, please contact your Adobe Experience Manager Implementation Partner or Adobe Consulting Services.