

**adaptTo()**

APACHE SLING & FRIENDS TECH MEETUP  
BERLIN, 26-28 SEPTEMBER 2016

Managing Cloud Performance and Large Data sets  
Mike Tilburg & Tom Blackford, Adobe Systems

# Best Practices and Basic Performance Guidelines

- Performance Docs Page
  - <https://docs.adobe.com/docs/en/aem/6-2/deploy/configuring/performance.html>
  - Performance Optimization Methodology
  - Performance Guidelines – page loading times
  - Performance Monitoring
  - Configuring Performance
  - Optimizing Dispatcher Performance – cache ratios
  - Backup performance

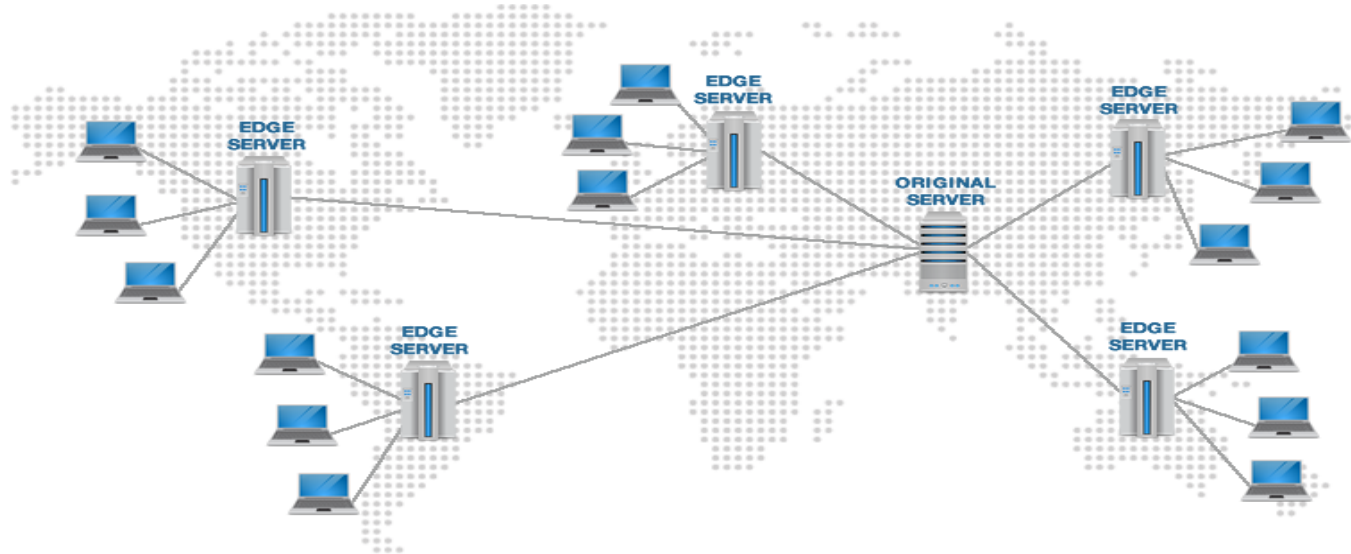
# Response Time Guidelines

Generally speaking, keep your uncached html requests to less than 100ms. More specifically, the following may serve as a guideline:

- 70% of the requests for pages should be responded to in less than 100ms.
- 25% of the requests for pages should get a response within 100ms-300ms.
- 4% of the requests for pages should get a response within 300ms-500ms.
- 1% of the requests for pages should get a response within 500ms-1000ms.
- No pages should respond slower than 1 second.

# Multi-Region: using CDN's

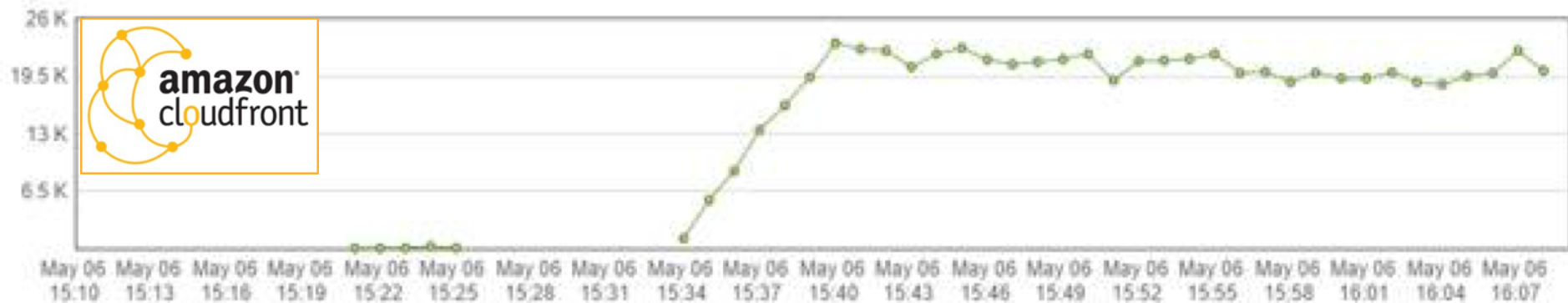
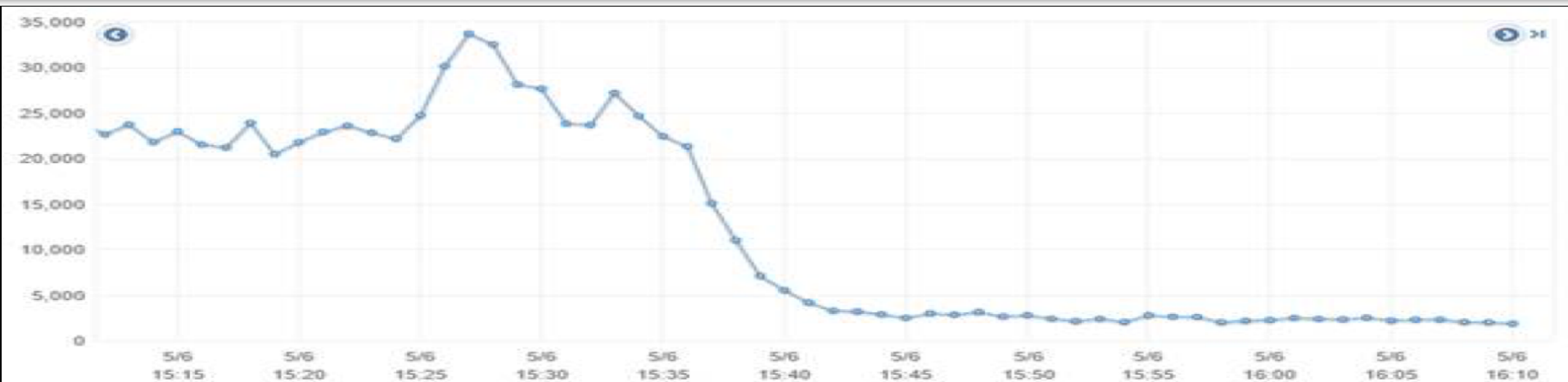
# Quick Overview of a CDN



# TTL and Invalidation Considerations

- Invalidating the entire cache can take a long time
  - 15-20 Minutes
  - TTL longer than this doesn't make sense
  - Usually very small TTL's are best
  - Use configurable TTL's for different parts of your site

# Result of CDN





# Calculating Cache Ratio

# Calculating Cache Ratio

- Cache Ratio
  - $$\frac{(\text{Total Number of Requests} - \text{Number of Requests on Publisher})}{\text{Total Number of Requests}}$$
  - Total Number of Requests – Sum of all requests from apache access\_log for example + CDN logs
  - Number of Requests on Publisher – using rlog
  - Remember if you don't have a 1:1 publisher/dispatcher pairing, you will need to add requests from all dispatchers and publishers together to get an accurate measurement

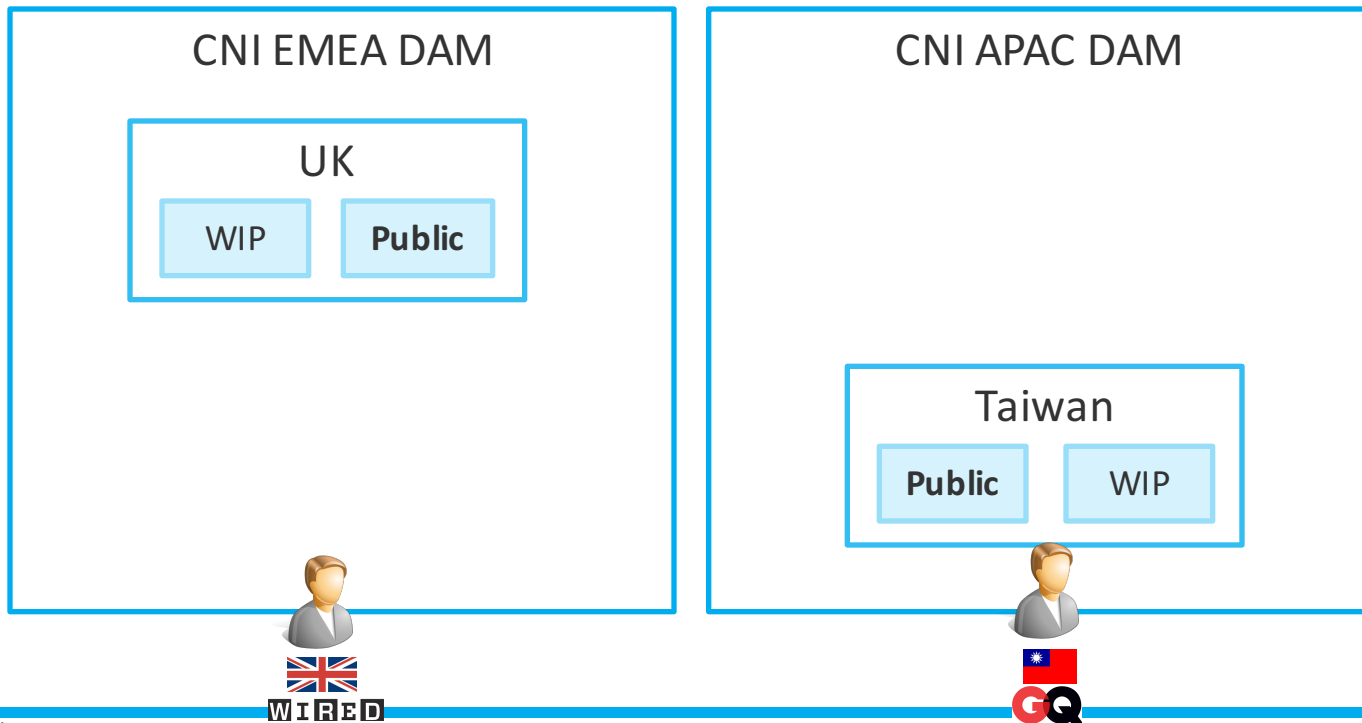
# Managing large data sets

# Background

- Condé Nast had been running AEM in Production since mid 2013
  - Hosted in Adobe Managed Services
  - Had previously upgraded from CQ 5.5 to 5.6.1 using standard ‘in place’ upgrade
- A highly complex application business logic for managing reuse of assets
- Relatively simple architecture but with some added complexity
  - Global deployment with custom ‘sharding’ of content between EMEA and APAC DAMs
  - Use of S3 Datastore
- Repository size ~400,000 assets (10TB)
  - In addition to upgrade, we were ‘merging’ in ~800,000 (15TB) of assets ingested in a separate AEM 5.6.1 instance

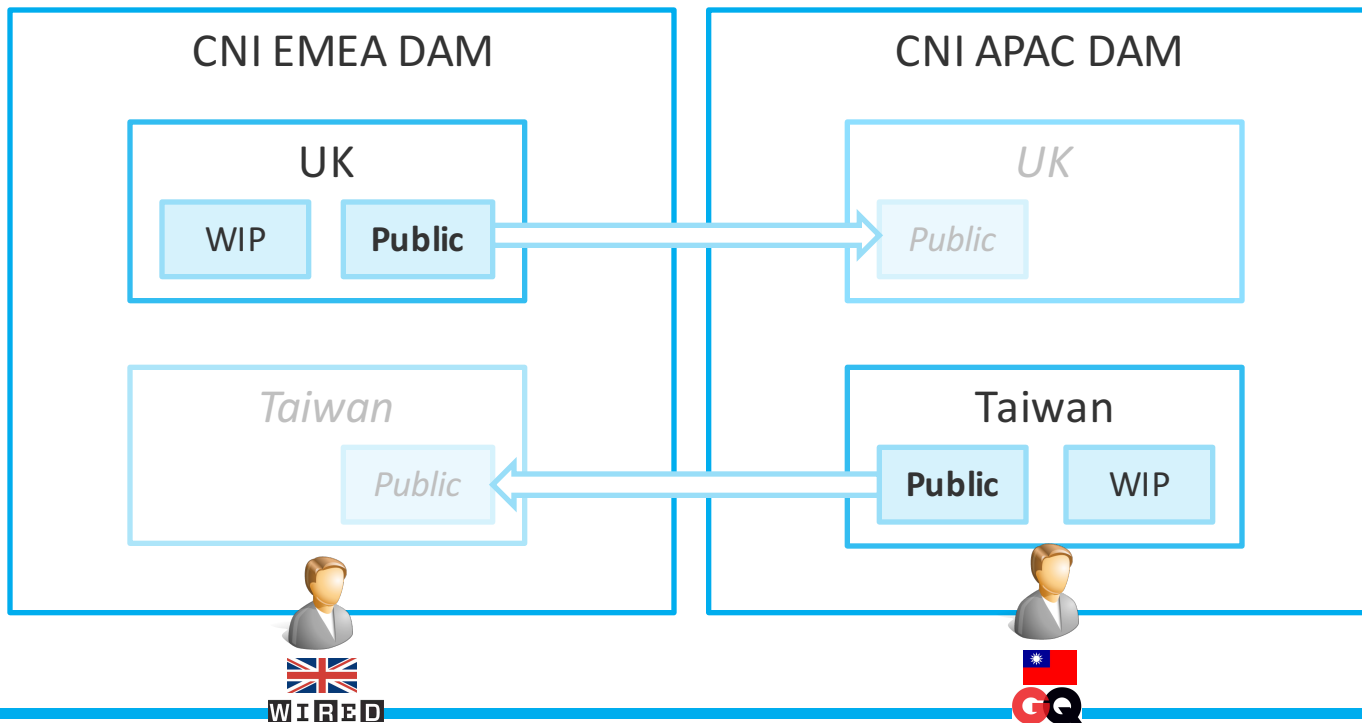
# Background

- To support users in APAC, the CNI solution has separate author instances in Singapore.



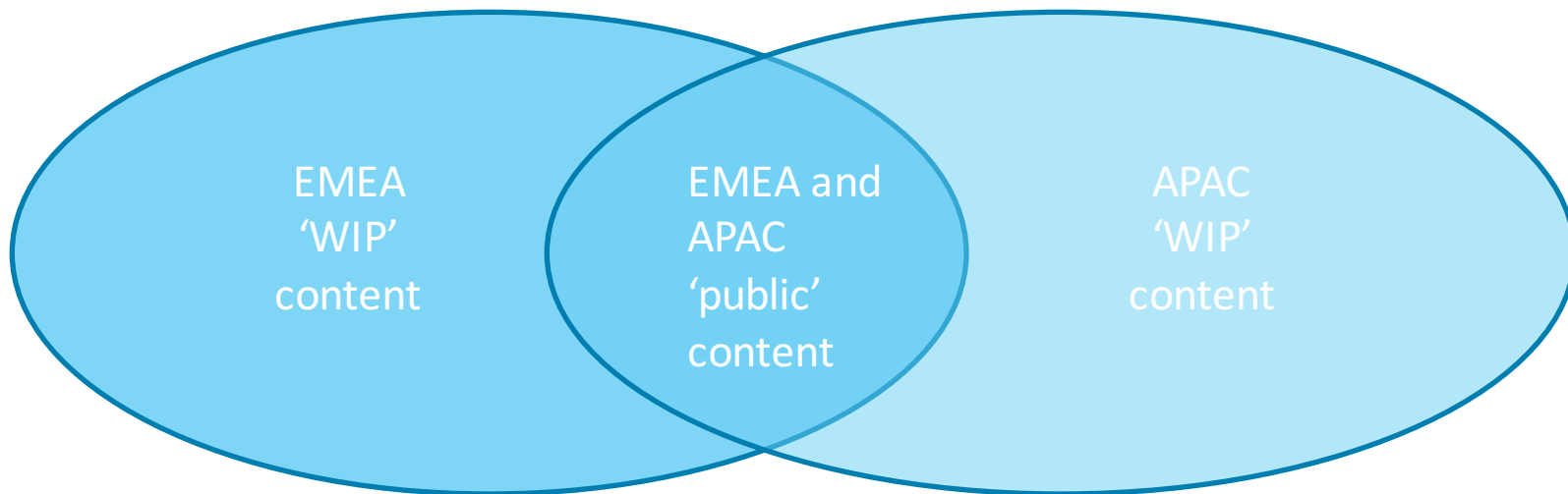
# Background

- **Public** content from each Market is replicated to the 'other' DAM.

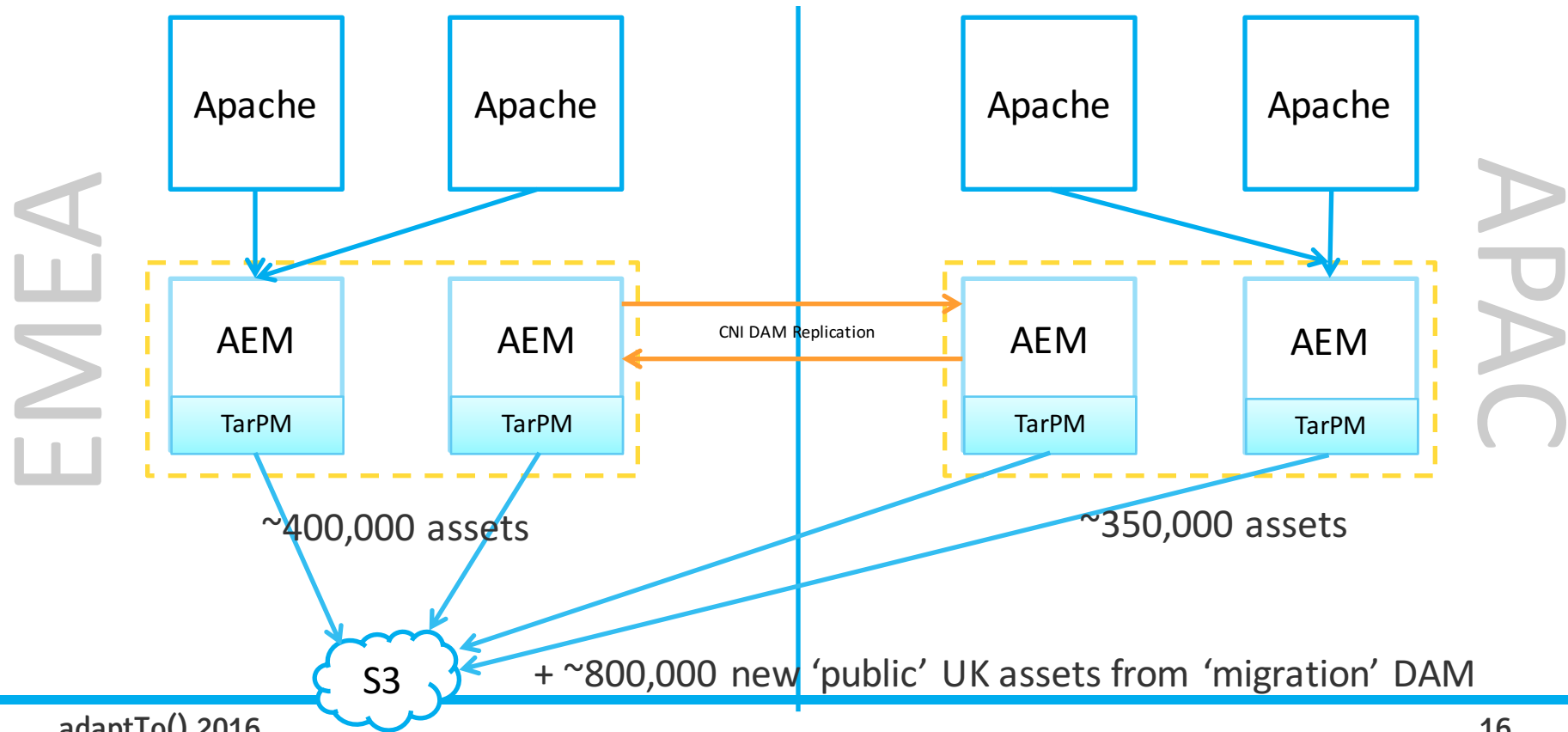


# Background

- As such, the two DAM's are not full replicas, and **would need to be upgraded separately**



# CQ 5.6.1 Architecture

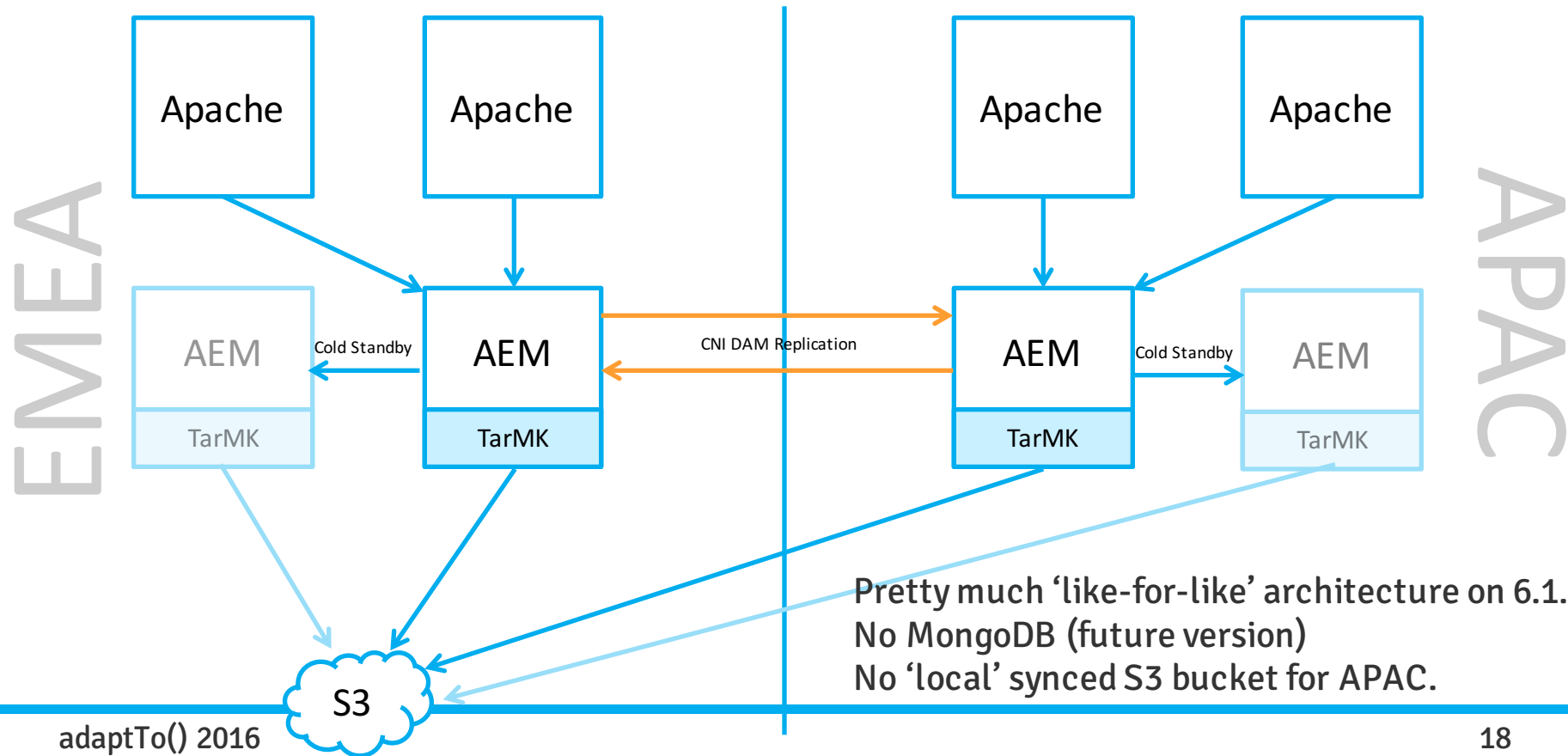




## CQ 5.6.1 Architecture

- The physical architecture, in particular, sharing the EMEA-based S3 bucket, presents some major challenges
  - Latency to the bucket is very high from APAC (>200ms compared to ~20ms in EMEA)
    - This particularly affected checking 'length' of binaries in the blobstore
    - Mitigated by RecordLengthCache in S3 Connector 1.5+
    - Less important in 6.x (see later)
  - Download speeds from the bucket are much slower in APAC (3-5MB/s compared to 40-50MB/s in EMEA)
    - Affected download of assets – mitigated (to some extent) by use of a large EBS S3 connector cache in APAC, to ensure 'in flight' assets remain in cache.

# AEM 6.1 Architecture



# Challenge 1 - Downtime

- DAM is a crucial system for CNI
  - No weekday downtime
  - All 80+ magazine titles constant 'archiving' published pages to DAM
- Upgrade had to be completed between Friday evening and Monday morning

# Challenge 1 - Downtime

## **Solution :** Side-by-Side upgrade

- Leverage crx2oak features here:
  - Subtree migration
  - Incremental migration
- Upgrade plan:
  - *Create a clean 6.1 environments in EMEA and APAC*
  - *Deploy CNI app and other settings*
  - *Migrate in 800,000 public UK 'migration' assets into each*
  - *During weeks leading up to Production golive*
    - *Take a snapshot of EMEA and APAC 5.6.1 Production*
    - *Migrate content into equivalent 6.1 environments*
  - During final golive weekend, repeat steps above for final 'delta' topups
  - Cutover to AEM 6.1

# Challenge 2 – Binary Lengths

- Oak has a useful optimisation where the length of the binary is stored within the node store as part of the blob id
- Blob ID : `98b62d90b6d032b14087286143e1aa80f80125d5#2378457644`
  - Datastore record id : `98b62d90b6d032b14087286143e1aa80f80125d5`
  - Length : `2378457644`
- Crx2oak must obtain the length of every binary property during the migration
  - The combined dataset for 6.1 has >8,000,000 binaries in the bucket
  - Assume that each of the upgraded CNI environments will contain >5,000,000 binary property references
    - EMEA –  $5,000,000 \times 0.02 = 100000 \text{ seconds (1.15 days)}$
    - APAC –  $5,000,000 \times 0.2 = 1,000,000 \text{ seconds (11 days!!!!)}$
- This is a challenge **only at upgrade time**

# Challenge 2 – Binary Lengths

## Solution : LengthCachingDataStore

- Wraps an existing CRX 2 DataStore (at upgrade time) to allow lengths of binaries to be checked from a CSV file.
- CSV file of binary lengths generated directly from S3 bucket using S3Cmd
  - As file contains all binaries in the bucket, **same CSV can be used in EMEA and APAC**
  - **Generate file on EMEA server (~4 hours) and copy to APAC server**

```
nohup s3cmd ls s3://[BUCKET_NAME] | awk '{ split($4,url,"/"); print $3|"url[4] }' > datastore-list.csv &
```

```
<DataStore class="org.apache.jackrabbit.oak.upgrade.blob.LengthCachingDataStore">  
  <param name="mappingFilePath" value="/mnt/datastoreLengthCache/datastore-list.csv" />  
  <param name="delegateClass" value="org.apache.jackrabbit.aws.ext.ds.S3DataStore" />  
  <param name="delegateConfigFilePath" value="/mnt/datastoreLengthCache/s3connector.properties"/>  
</DataStore>
```

# Challenge 3 – Indexing

- Each ‘crx2oak’ topup, required (re)index of Oak lucene indexes (damAssetLucene etc)
- Again there is a huge performance optimisation in Oak OOTB
  - The CRX 2.x NodeIndexer used to obtain streams of ALL binary properties when indexing a node, regardless of whether the mime type were supported by Tika text extraction.
    - Ie ALL asset binaries in bucket would need to be downloaded
  - Oak checks Tika support first, and obtains blobs for mime types which Tika can handle.
    - Far fewer binaries need to be downloaded from bucket (only PDF assets for CNI)
- Even so, CNI’s 6.1 Production Dataset contains ~3TB of ‘text extractable’ files
  - Indexer is single-threaded so (ignoring the latency and tika extraction time):
    - EMEA –  $3,000,000 / 30 = 100000 \text{ seconds (1.15 days)}$
    - APAC –  $3,000,000 / 3 = 1,000,000 \text{ seconds (11 days!!!!)}$
- A challenge ***every time we reindex a lucene index which includes binary props***

# Challenge 3 – Indexing

## Solution : Pre-Extraction

- Configure 'PreExtractedTextProvider' to check for extracted text in a local 'store' prior to extracting from binary property (& downloading binary from bucket)  
    Needed product change to support S3 and use during both indexing and re-indexing.  
    Generally, same store can be used in EMEA and APAC (binaries not in store get downloaded)
- Store is generated in 2 phases  
    An oak-run script to generate CSV file containing path, blob id and mime type (AEM stopped)  

```
nohup java -cp jars/*:oak-run-1.4-r1728642.jar:tika-app-1.9.jar org.apache.jackrabbit.oak.run.Main tika --nodestore /mnt/crx/author/crx-quickstart/repository/segmentstore --s3-config-path /mnt/preExtraction/s3.conf --data-file /mnt/preExtraction/dump.csv generate &
```

  
    A command to download supported binaries, extract text and add to store.  

```
nohup java -cp jars/*:oak-run-1.4-r1728642.jar:tika-app-1.9.jar org.apache.jackrabbit.oak.run.Main tika --store-path /mnt/preExtraction/store --s3-config-path /mnt/preExtraction/s3.conf --data-file /mnt/preExtraction/dump.csv extract
```
- **Re-index time ~3 hours in EMEA and APAC**



# Upgrade

- Upgrade completed between Friday evening and Saturday afternoon
  - including setup of Tar MK Cold Standby
- Resulting environment ~1,200,000 assets and 25TB in S3 (>80,000,000 nodes)
  - Current size ~1,500,000 and 30TB in S3 (~100,000,000 nodes)



#adobelife

# Thank You

Mike Tilburg  
Senior AEM Consultant  
Adobe Central EMEA  
[mtilburg@adobe.com](mailto:mtilburg@adobe.com)

Tom Blackford  
Cloud Engineer  
Adobe UK  
[blackfor@adobe.com](mailto:blackfor@adobe.com)

For more information about  
Adobe Managed Services  
Adam Pazik  
[pazik@adobe.com](mailto:pazik@adobe.com)

For more information about  
Adobe Professional Services  
Kerem Ergun  
[ergun@adobe.com](mailto:ergun@adobe.com)