

Content Relay

By: Anders Bo Rasmussen



Overview

▶ Introduction	01
▶ Solution Overview	02
▶ C4 System Context	04
▶ C4 Container	05
▶ Challenges	06
▶ Collaboration	07
▶ Conclusion	08
▶ QA	09



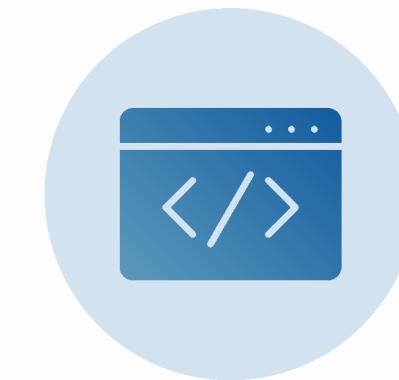


Introduction

An application needs to be in place for content distribution. So the digital platforms can consume it.

The application must expose content and assets + metadata through an interface that can be easily consumed by digital platform developers as the briefing and order process.

Solution Overview



Metadata and Asset Management Service

Acts as our ACL (Anti Corruption Layer) by aggregating data from various sources, validates, and transforms them to be in line with our domain context.
Once data is transformed and validated, notifies the Content Distribution Service of the metadata through events.

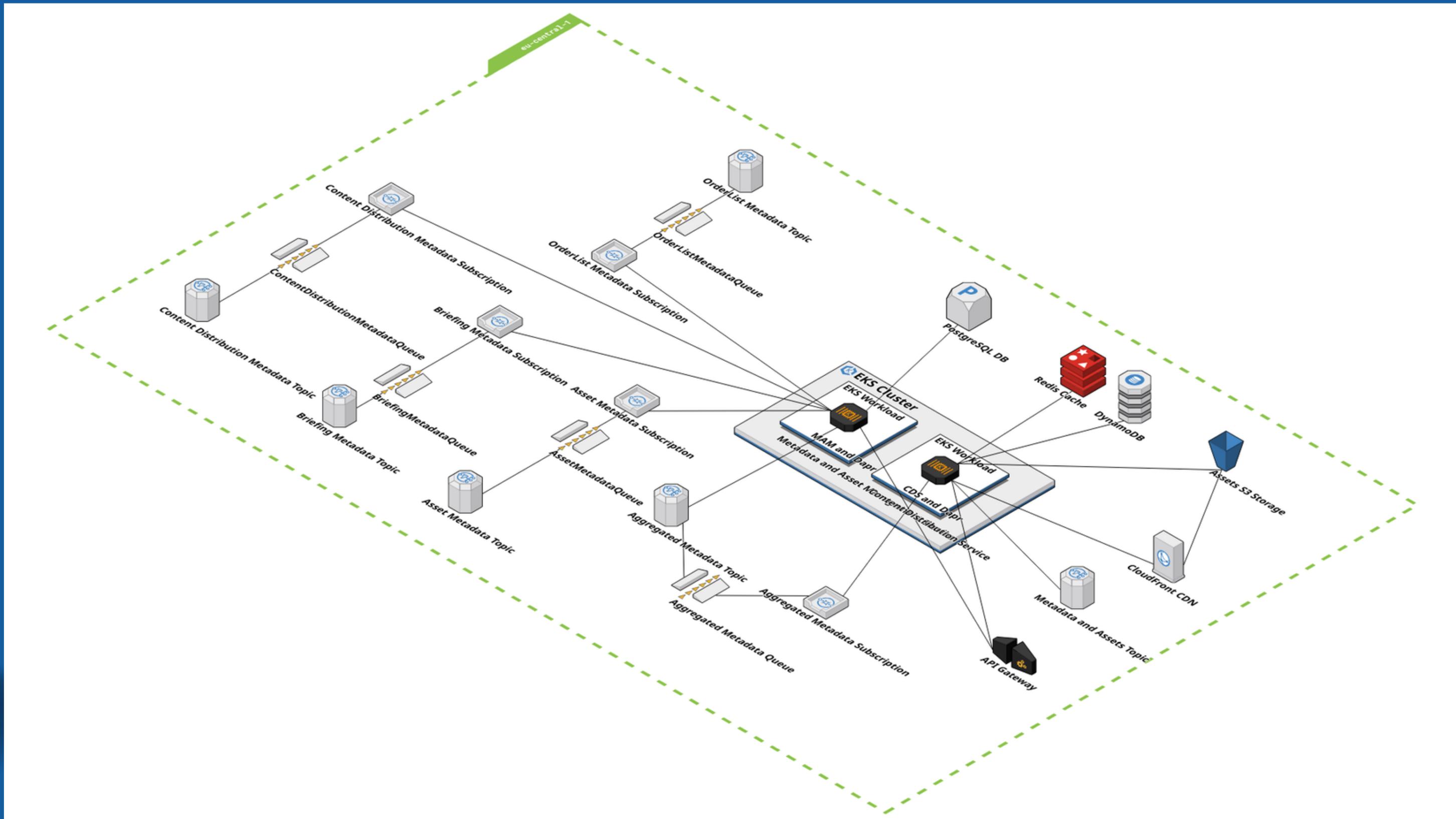


Content Distribution Service

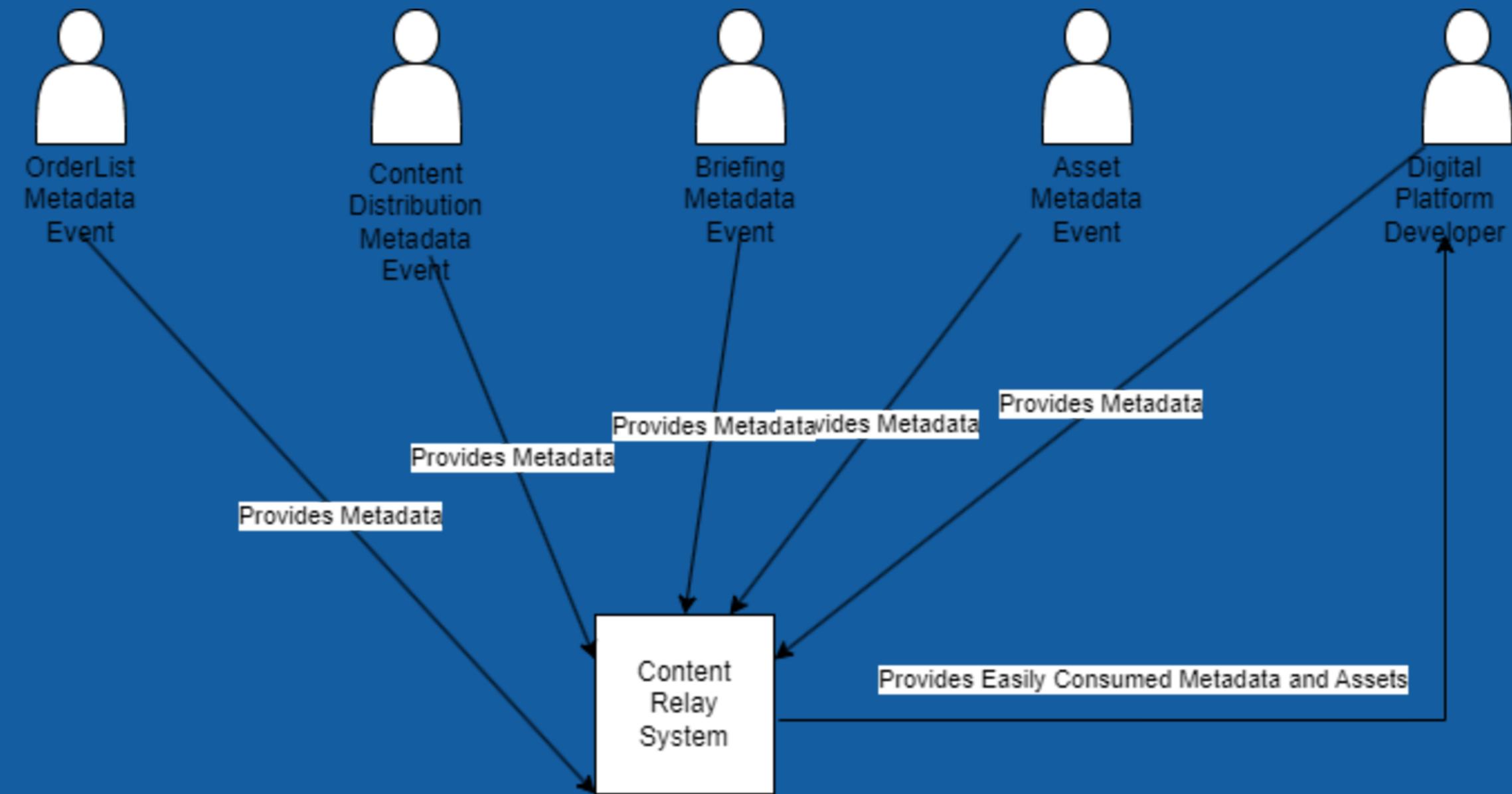
Handles uploads of assets to S3 bucket.
Notifies Cloudfront CDN of new and updated assets.
Notifies Digital Platform Developers of new and updated assets through events.
Might also support Webhooks in the future.

Stores metadata in DynamoDB for fast and easy retrieval.
Cache metadata and S3 links in Redis for fast retrieval.

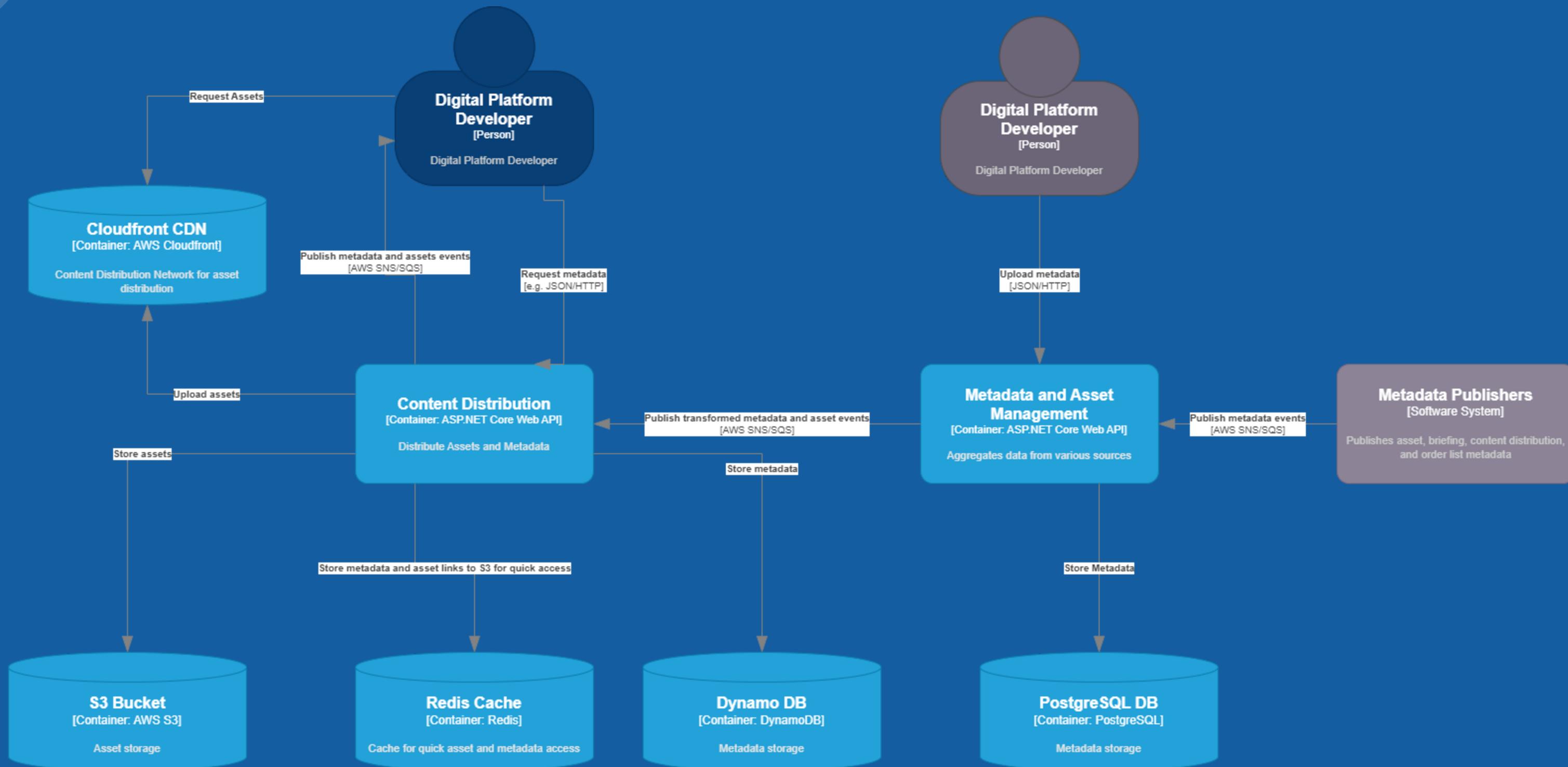
Solution Overview



C4 System Context



C4 Container



Challenges

A number of challenges quickly appeared in this case.

I wanted to have added more tests, and have scoped a MVP that was feasible in the given time frame.

Challenge 01

Way too much time was spent on the ACL

Challenge 02

"We figure it out as we go" attitude might not always work when time is short.

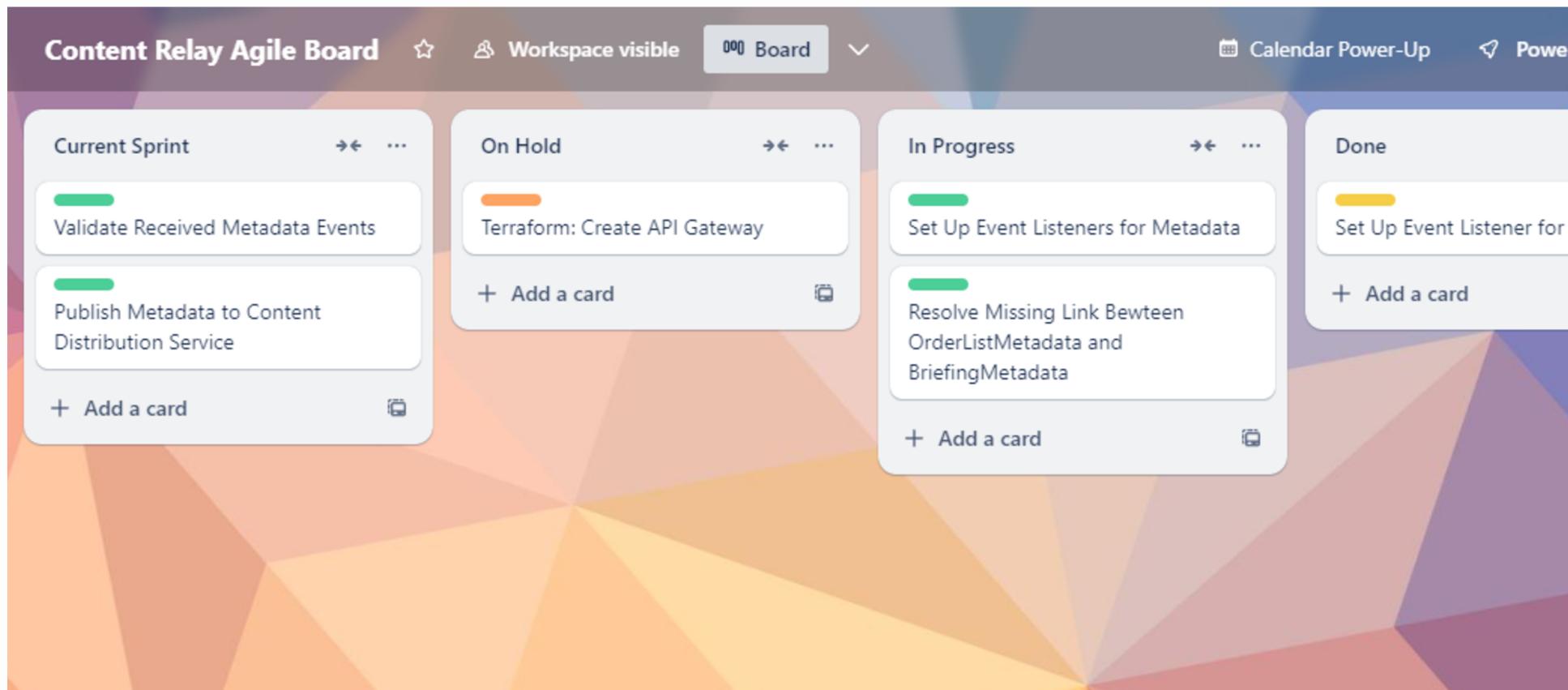
Challenge 03

No domain experts – or sparring partners.



Collaboration

Context mapping.
Agile board.
Pair programming.
MVP scoping with Product Owner.



Conclusion



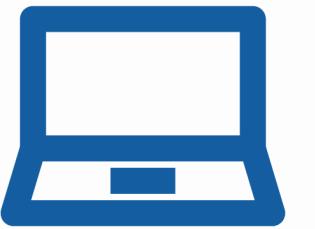
Scalability

Design was built with the idea that if we experience a spike in data coming in, we can scale the MAM service up - if lot of data was being read, we can scale the CDS instead.



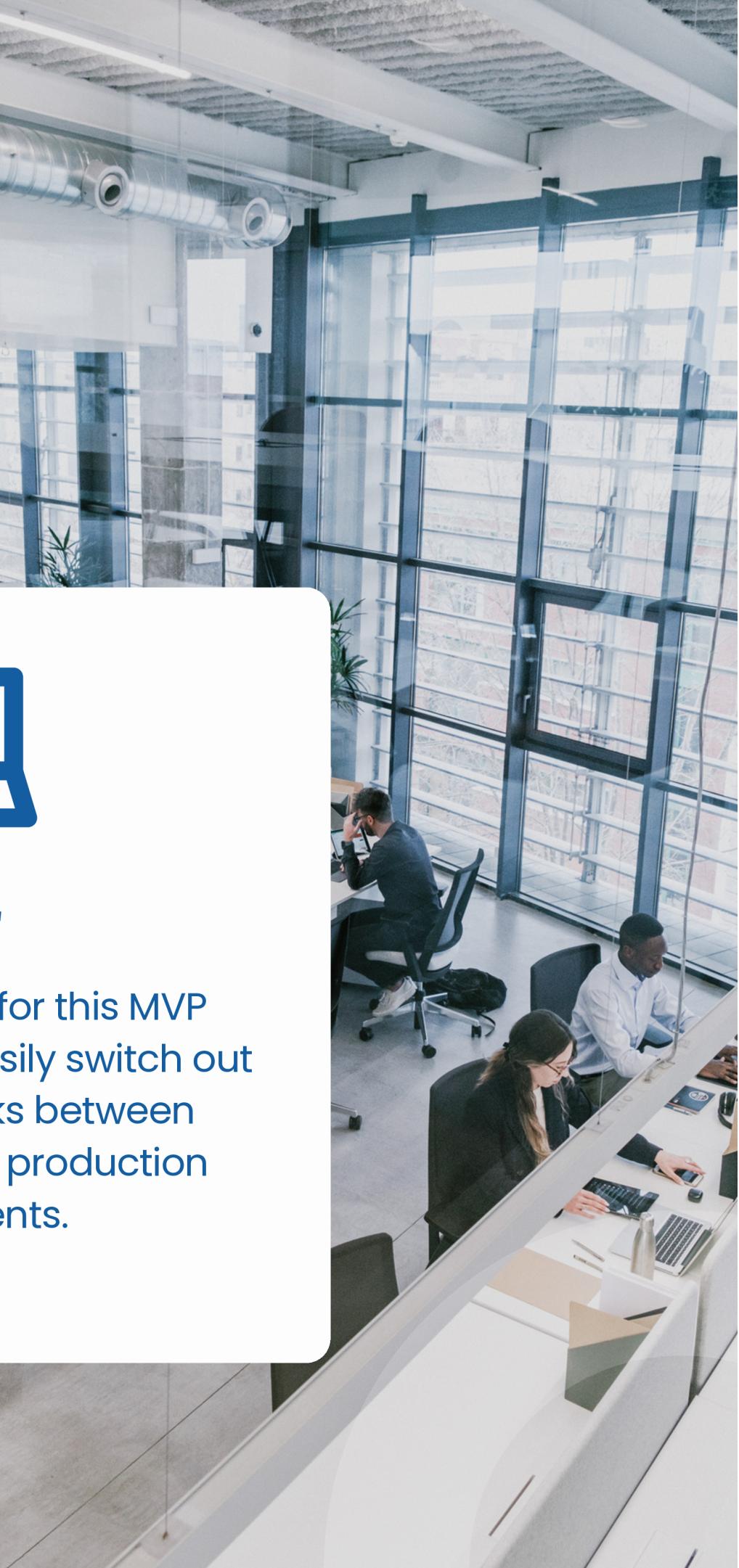
Time is short

Scope MVP according to time.



Dapr

Dapr was chosen for this MVP because you can easily switch out the building blocks between development and production environments.



QA