APACHE SLING & FRIENDS TECH MEETUP
10-12 SEPTEMBER 2018

Oak Direct Binary Access

Matt Ryan, Adobe, @mattvryan

https://bit.ly/2Mo3ngj CCO

What?

adaptTo()

Oak Direct Binary Access is…

- A new Oak feature that allows a client to
  - Download a blob stored in the repository directly from blob storage
  - Upload a blob directly to blob storage and then add it to the repository
- Makes e.g. Asset Compute Service possible (@alexkli's talk)

Advantages

- Cloud First Mindset
  - Blob uploads and downloads happen directly between clients and storage – no state is preserved in Oak
- Scalability
  - Significant reduction of load on Oak instance
- Supports independent workflow processing
- Security
  - All blob access still depends on Oak for authorization
  - URIs are signed with short (configurable) TTLs

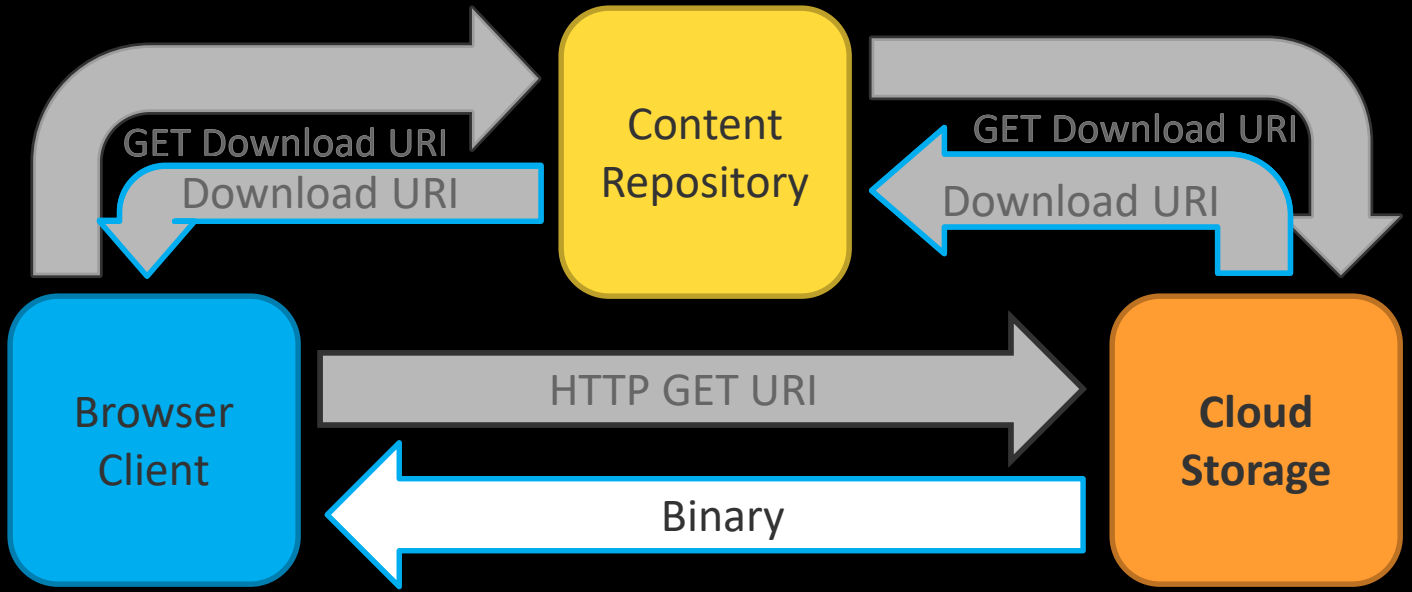
How?

Traditional Download

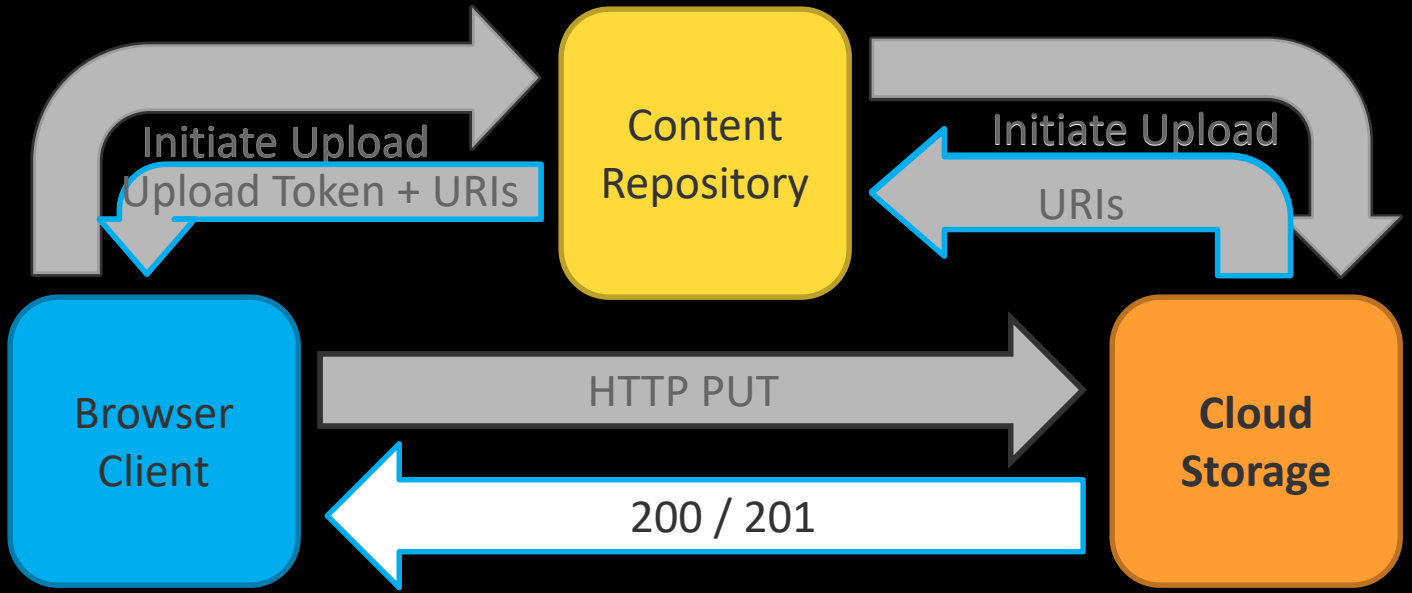
Traditional Download

Traditional Upload

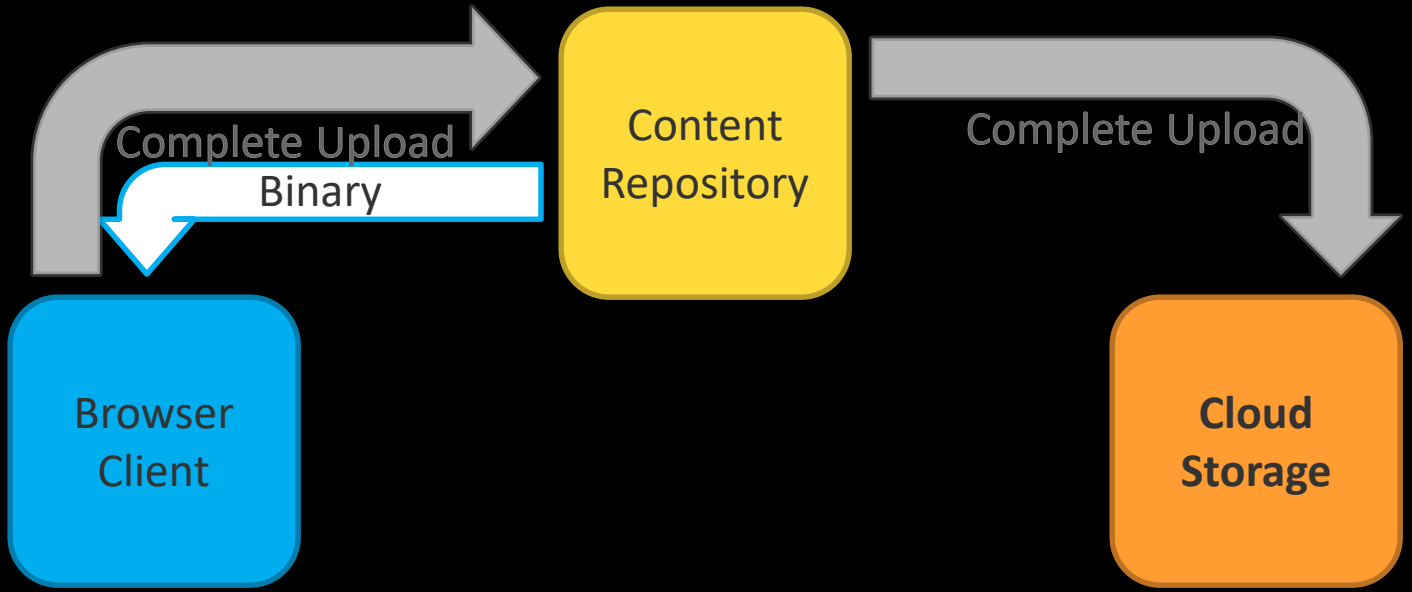
Direct Binary Access - Download



Direct Binary Access – Upload



Direct Binary Access – Upload



Download Example

```
Node node = session.getNode(”/myNode”);
Binary binary =
      node.getProperty(JcrConstants.JCR_DATA)
            .getBinary();
if (binary instanceof BinaryDownload) {
    BinaryDownloadOptions options =
      BinaryDownloadOptions.DEFAULT;
    URI uri = ((BinaryDownload)binary)
      .getURI(options);
}
```

Download Example

```
Binary binary = node.getProperty(JcrConstants.JCR_DATA)
    .getBinary();
if (binary instanceof BinaryDownload) {
    BinaryDownloadOptions options = BinaryDownloadOptions
        .builder()
        .withMediaType("application/pdf")
        .withFileName("policy.pdf")
        .withDispositionTypeAttachment()
        .build();
    URI uri = ((BinaryDownload) binary).getURI(options);
}
```

Upload Example

```
ValueFactory valueFactory = session.getValueFactory();
if (valueFactory instanceof JackrabbitValueFactory) {
    JackrabbitValueFactory vf =
        (JackrabbitValueFactory) valueFactory;
    BinaryUpload up =
        vf.initiateBinaryUpload(binarySize, numUploadParts);
    URI uploadURI = upload.getUploadURIs().iterator().next();
    // direct upload magic here
    Binary binary = vf.completeBinaryUpload(up.getUploadToken());
    Node node = session.getNode("/mynode");
    node.setProperty(JcrConstants.JCR_DATA, binary);
}
```

Multi-Part Upload Example

```
JackrabbitValueFactory vf =
    (JackrabbitValueFactory) session.getValueFactory();
BinaryUpload upload =
      vf.initiateBinaryUpload(binarySize, numUploadParts);
Iterator<URI> iter = upload.getUploadURIs().iterator();
while (iter.hasNext()) {
    URI uploadURI = iter.next();
    // Direct upload each part
}
Binary binary =
      vf.completeBinaryUpload(upload.getUploadToken());
```

When?

adaptTo()

When is it available?

- Oak Unstable – Available now (@since 1.9.8)
- Oak Stable – Will be in v.next (@since 1.10)

More Info

- Feature documentation (On GitHub):
  - https://bit.ly/2oY7dUb
- JavaDoc (On jackrabbit.apache.org):
  - https://bit.ly/2CKNbWT (JackrabbitValueFactory)
  - https://bit.ly/2CKNDV5 (BinaryUpload)
  - https://bit.ly/2MmSFqk (BinaryDownload)

Thank You!