Agenda

- Container Images
- Associating Artifacts
- Referrers Response
- Demo
- Status

Modifying the Immutable: Attaching Artifacts to OCI Images

Brandon Mitchell
Mastodon: @bmitch@fosstodon.org
Twitter: @sudo_bmitch
GitHub: sudo-bmitch

```
$ whoami
- Brandon Mitchell
- Solutions Architect @ BoxBoat, an IBM Company
- OCI Maintainer, regclient, Docker Captain
- StackOverflow, CNCF, OpenSSF
```

Container Images

- Filesystem layers
- Created with `docker build`
- Sometimes referenced with `sha256:...`
- Inspecting images to see history and config
- Multi-platform images

Content Addressable Store

```
$ ldigest=sha256:8921db27df2831fa6eaa85321205a2470c669b855f3ec95d5a3c2b46de0442c9

$ curl -s http://localhost:5000/v2/$repo/blobs/$ldigest | sha256sum
8921db27df2831fa6eaa85321205a2470c669b855f3ec95d5a3c2b46de0442c9  -
```

Layers

```
$ curl -s http://localhost:5000/v2/$repo/blobs/$ldigest | tar -tvzf - | head
drwxr-xr-x 0/0               0 2023-01-09 07:46 bin/
lrwxrwxrwx 0/0               0 2023-01-09 07:46 bin/arch -> /bin/busybox
lrwxrwxrwx 0/0               0 2023-01-09 07:46 bin/ash -> /bin/busybox
lrwxrwxrwx 0/0               0 2023-01-09 07:46 bin/base64 -> /bin/busybox
lrwxrwxrwx 0/0               0 2023-01-09 07:46 bin/bbconfig -> /bin/busybox
-rwxr-xr-x 0/0          841392 2022-11-19 05:13 bin/busybox
lrwxrwxrwx 0/0               0 2023-01-09 07:46 bin/cat -> /bin/busybox
lrwxrwxrwx 0/0               0 2023-01-09 07:46 bin/chattr -> /bin/busybox
lrwxrwxrwx 0/0               0 2023-01-09 07:46 bin/chgrp -> /bin/busybox
lrwxrwxrwx 0/0               0 2023-01-09 07:46 bin/chmod -> /bin/busybox
```

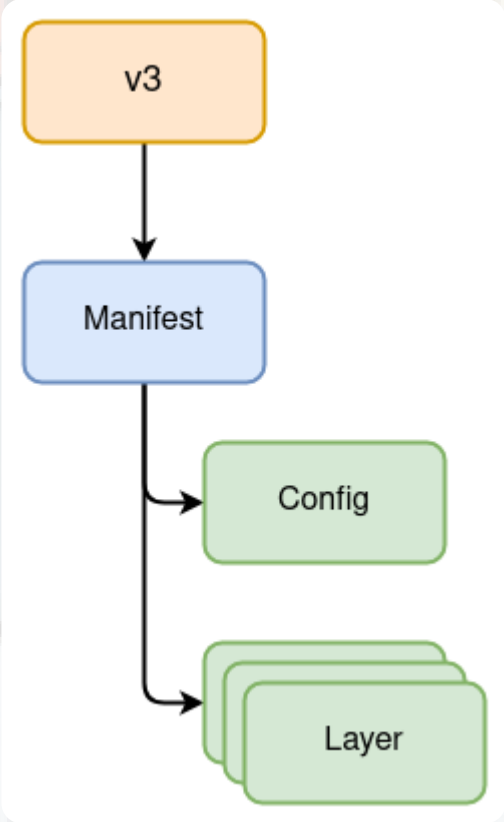
Config

```
$ curl -s http://localhost:5000/v2/$repo/blobs/$cdigest | jq .
{
  "config": {
    "Env": [
      "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
    ],
    "Cmd": [ "/bin/sh" ],
    "WorkingDir": "",
...
    "Labels": null
  },
  "history": [
    {
      "created": "2023-01-09T17:05:20.497231175Z",
      "created_by": "/bin/sh -c #(nop) ADD file:e4d600fc4c9c29... in / "
...
}
```

Image Manifest

```
$ curl -s http://localhost:5000/v2/$repo/manifests/$mdigest | jq .
{
  "schemaVersion": 2,
  "mediaType": "application/vnd.docker.distribution.manifest.v2+json",
  "config": {
    "mediaType": "application/vnd.docker.container.image.v1+json",
    "size": 1472,
    "digest": "sha256:042a816809aac8d0f7d7cacac7965782ee2ecac3f21bcf9f24b1de1a7387b769"
  },
  "layers": [
    {
      "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
      "size": 3370628,
      "digest": "sha256:8921db27df2831fa6eaa85321205a2470c669b855f3ec95d5a3c2b46de0442c9"
    }
  ]
}
```

Tags



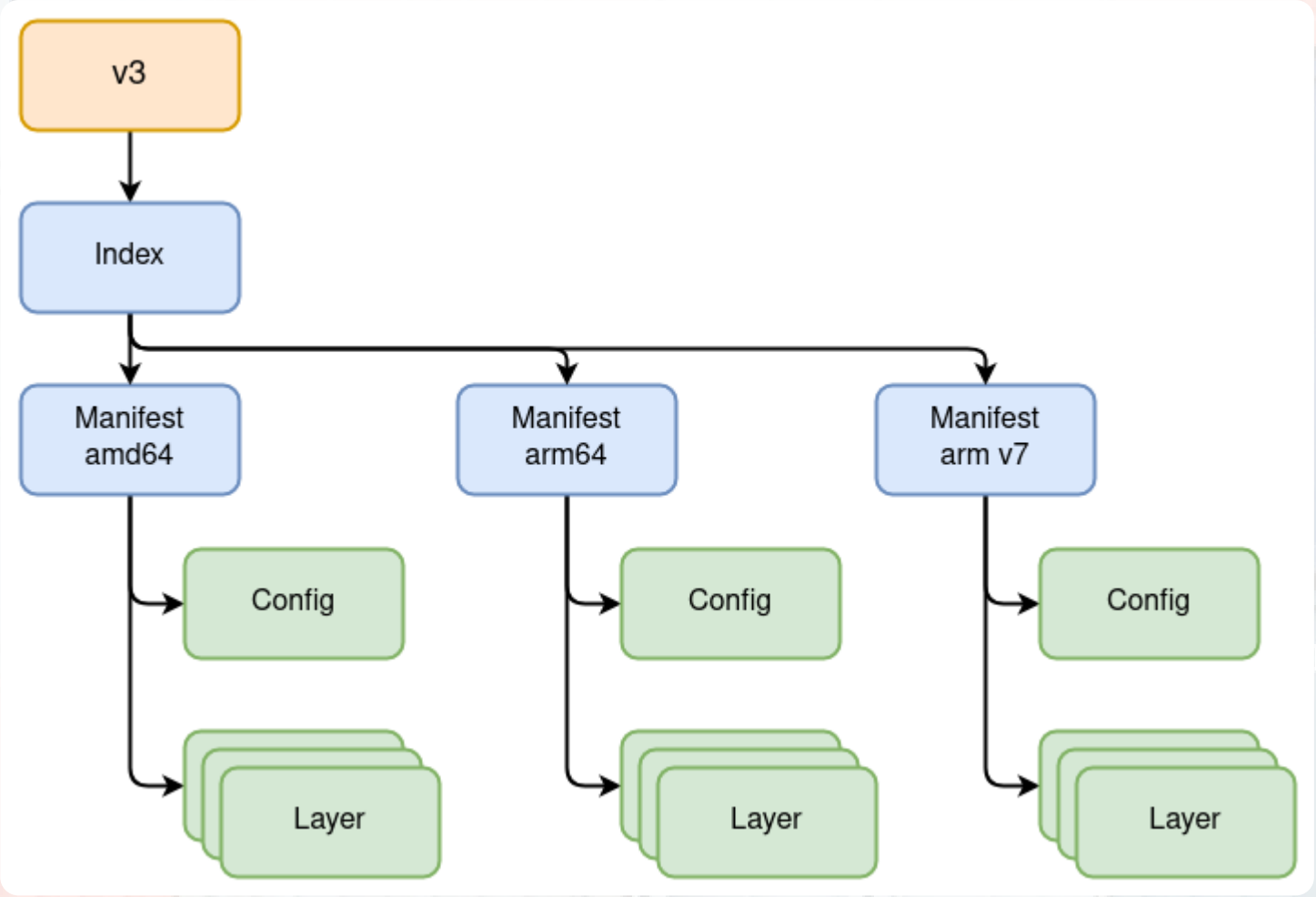
Immutability

- Content Addressable Store: content of each node is referenced by hash of itself
- DAG: Directed Acyclic Graph
- Merkle tree:
  - Manifest is the root node with a hash
  - Content of the root node is the hash of each child node

Multi-platform Images

```
$ curl -H 'Accept: application/vnd.docker.distribution.manifest.list.v2+json' -s \
  http://localhost:5000/v2/$repo/manifests/alpine | jq .
{
  "mediaType": "application/vnd.docker.distribution.manifest.list.v2+json",
  "manifests": [
    {
      "digest": "sha256:93d5a28ff72d288d69b5997b8ba47396d2cbb62a72b5d87cd3351094b5d578a0",
      "mediaType": "application/vnd.docker.distribution.manifest.v2+json",
      "platform": {
        "architecture": "amd64",
        "os": "linux"
      },
      "size": 528
    },
    {
      "digest": "sha256:01a4cdaebc9c6af607753cc538c507d0867897cdf9a1caa70bbab2eb1506c964",
      "mediaType": "application/vnd.docker.distribution.manifest.v2+json",
      "platform": {
        "architecture": "arm",
        "os": "linux",
        "variant": "v6"
...
```

Multi-platform Images



Artifacts

```
$ adigest=sha256:ea706edf61ef640bcdf3c9ac9045c28446e6b2d08541b9ad614c7267d0b87375
$ curl -s http://localhost:5000/v2/$repo/blobs/$adigest
contains electrons
```

Artifacts

```
$ adigest=sha256:ea706edf61ef640bcdf3c9ac9045c28446e6b2d08541b9ad614c7267d0b87375
$ curl -s http://localhost:5000/v2/$repo/blobs/$adigest
contains electrons

$ curl ... http://localhost:5000/v2/$repo/manifests/$amdigest | jq .
{
  "schemaVersion": 2,
  "mediaType": "application/vnd.oci.image.manifest.v1+json",
  "config": {
    "mediaType": "application/vnd.example.ebom.config",
    "size": 2,
    "digest": "sha256:44136fa355b3678a1146ad16f7e8649e94fb4fc21fe77e8310c060f61caaff8a"
  },
  "layers": [
    {
      "mediaType": "application/vnd.example.ebom.data",
      "size": 19,
      "digest": "sha256:ea706edf61ef640bcdf3c9ac9045c28446e6b2d08541b9ad614c7267d0b87375"
    }
  ]
}
```

Challenge: Associating Artifacts with Images

- SBOMs, Attestations, Vulnerability Reports, Signatures
- How do we attach to an immutable object?

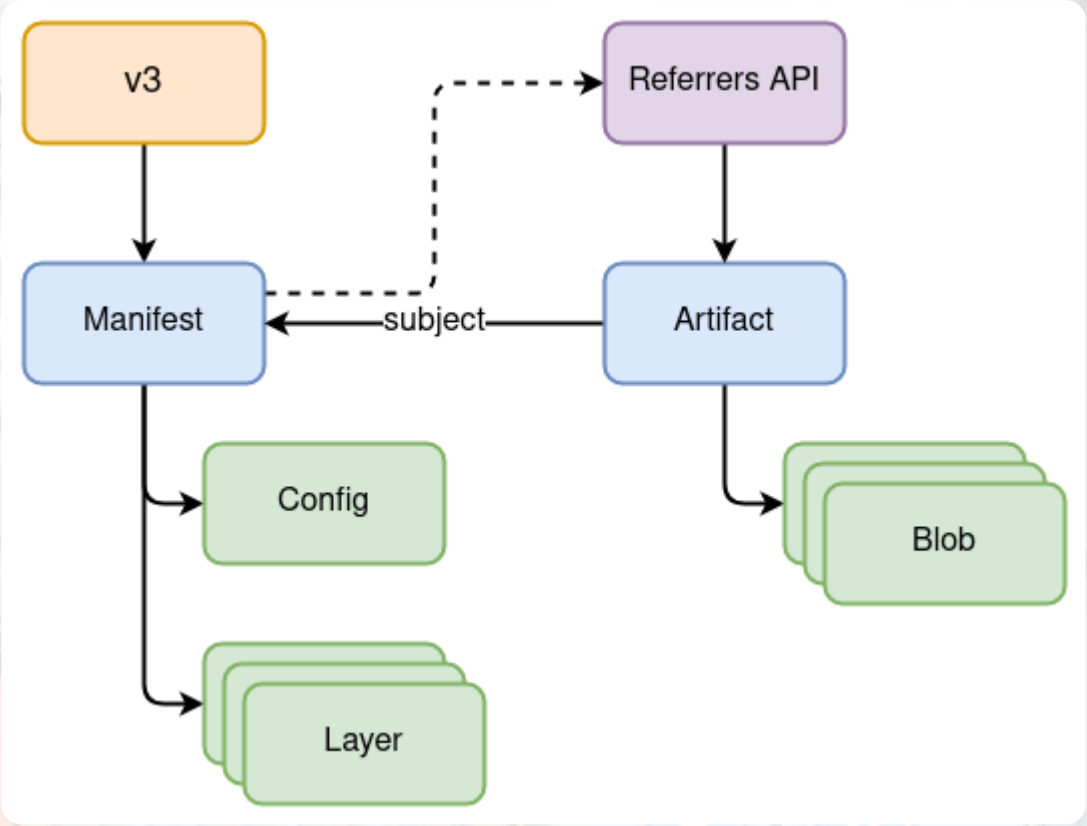
Modifying An Index?

```
{
  "mediaType": "application/vnd.oci.image.index.v1+json",
  "manifests": [
    {
      "digest": "sha256:93d5a28ff72d288d69b5997b8ba47396d2cbb62a72b5d87cd3351094b5d578a0",
      "mediaType": "application/vnd.oci.image.manifest.v1+json",
      "platform": {
        "architecture": "amd64",
        "os": "linux"
      },
      "size": 528
    },
    {
      "digest": "sha256:01a4cdaebc9c6af607753cc538c507d0867897cdf9a1caa70bbab2eb1506c964",
      "mediaType": "application/vnd.oci.image.manifest.v1+json",
      "artifactType": "application/vnd.example.ebom.config",
      "extends": "sha256:93d5a28ff72d288d69b5997b8ba47396d2cbb62a72b5d87cd3351094b5d578a0",
      "size": 1024,
...
```

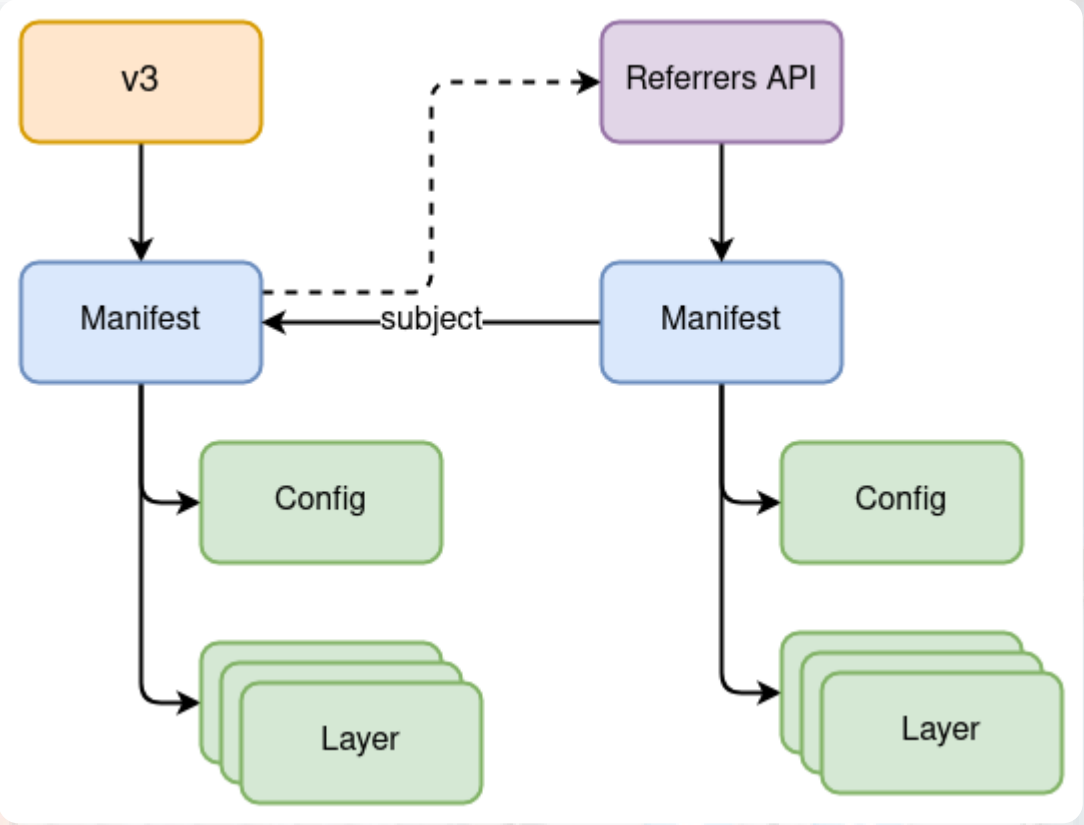
How Do We Modify the Immutable

- Adding metadata to an image would modify it
- How do we attach metadata to an existing image?
- Working Group goals:
  - Efficient on registry processing, bandwidth, and round trips
  - Attaching to existing images
  - Option to detach when copying
  - Referencing images by digest or tag
  - Multiple artifacts of the same type are possible
  - Not limited to known artifact types
- Multiple solutions

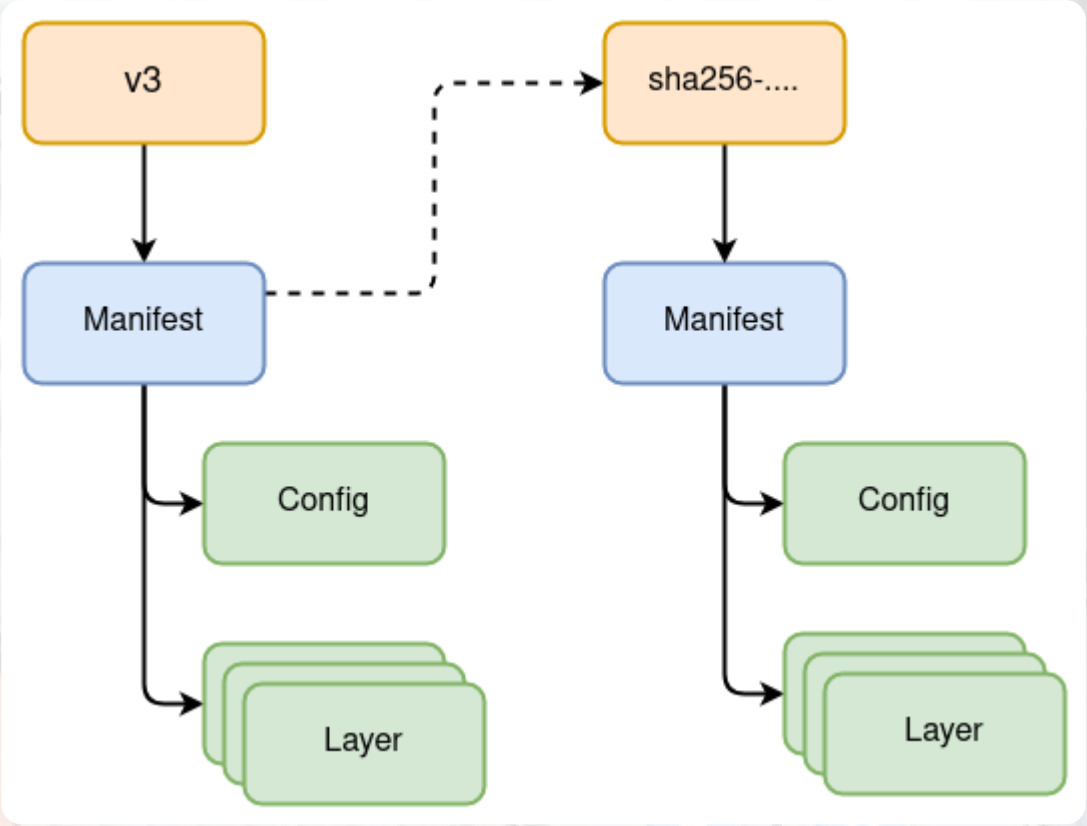
Option: New Manifest, Subject Field, and API



Option: Subject Field on Existing Manifests



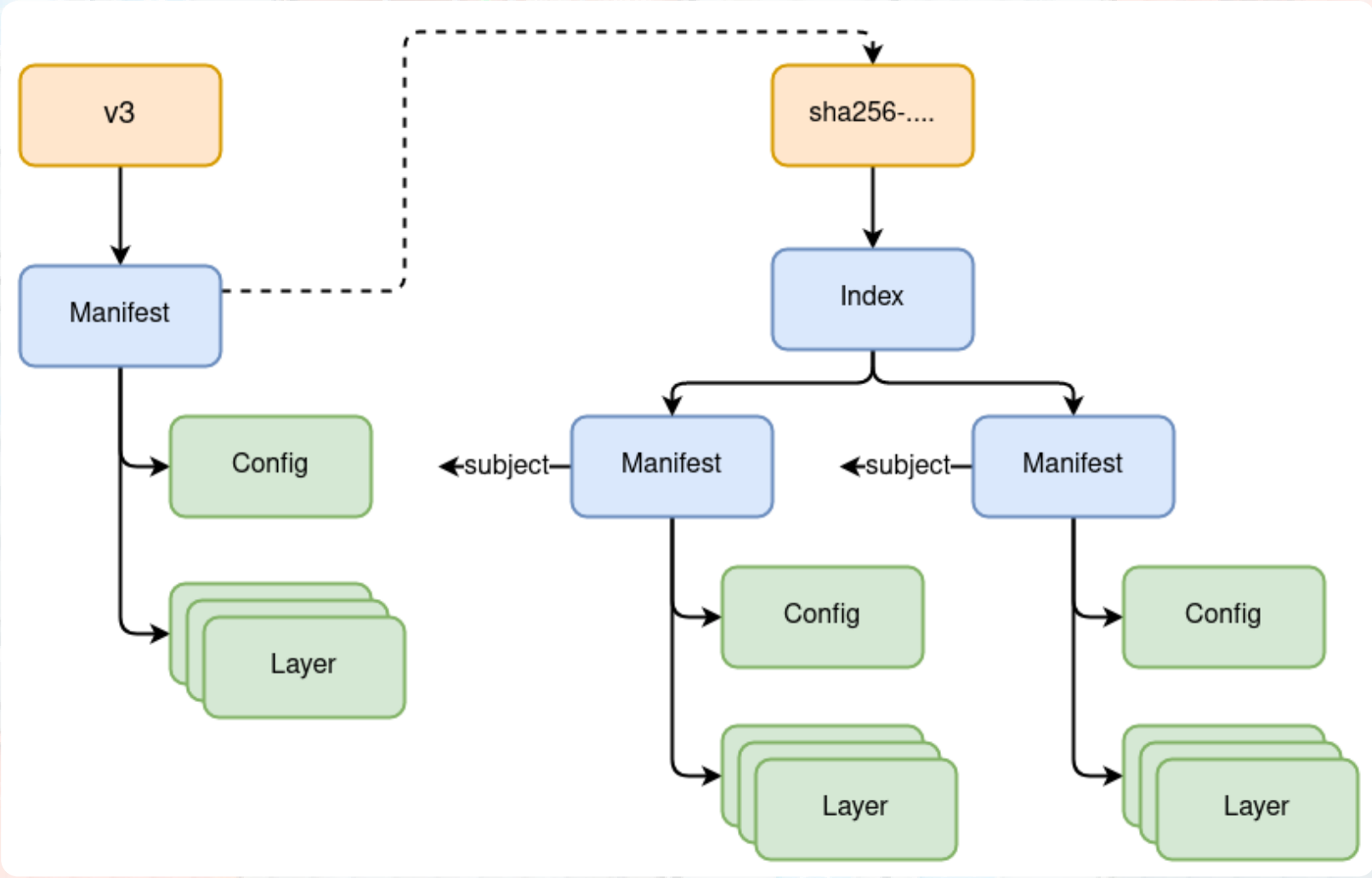
Option: Custom Tag Syntax

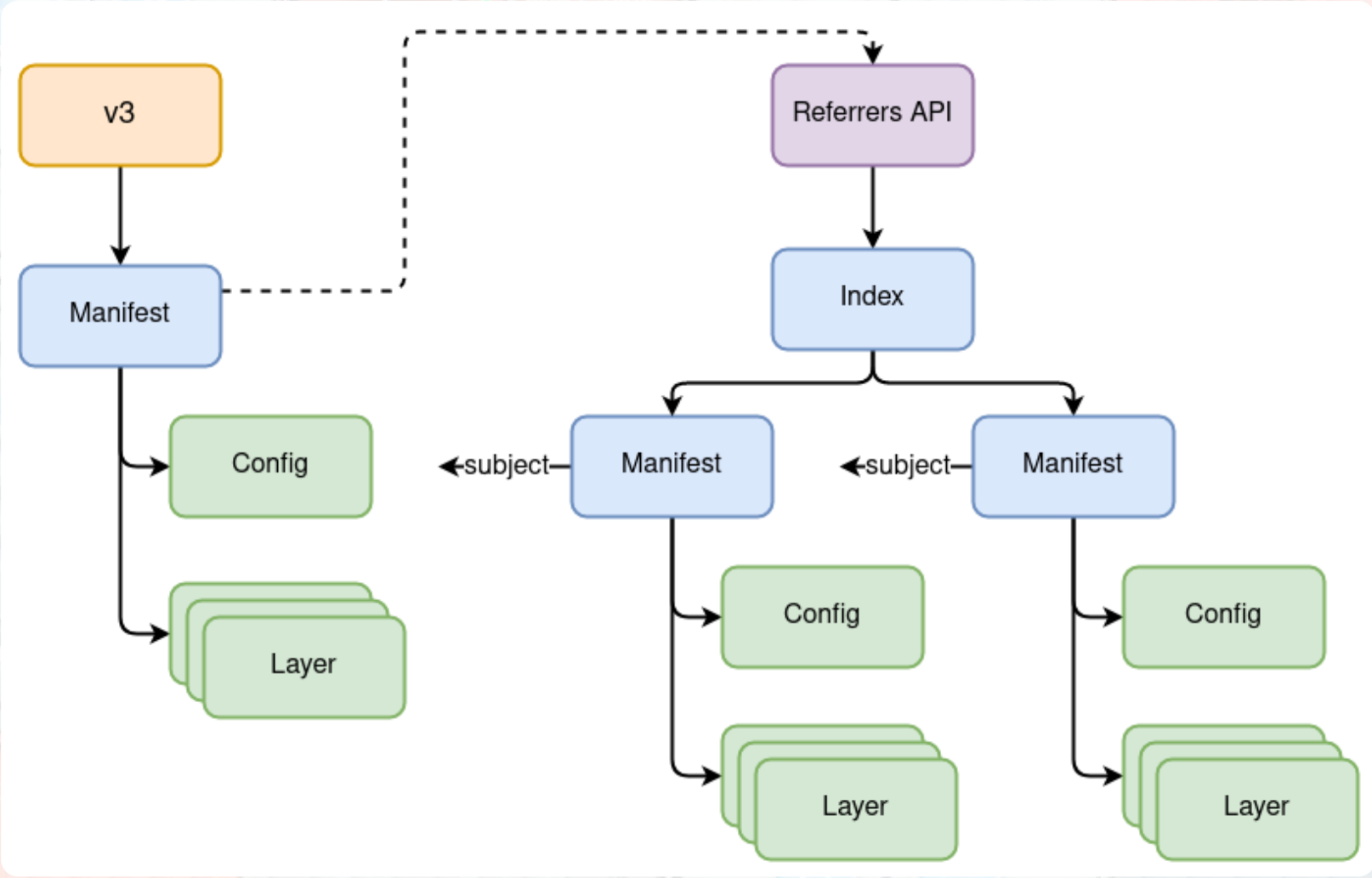


So which option did we pick?

How does this all come together?

- ~~Add an Artifact manifest~~ Retracted the Artifact manifest
- Add a `subject` and `artifactType` field to the Image manifest
- Add a referrers API to query the subject field
- Clients manage a tag if the referrers API isn't available


v3
Manifest
Config
Layer
sha256-....
Index
subject
Manifest
Config
Layer
subject
Manifest
Config
Layer


v3
Manifest
Config
Layer
Referrers API
Index
subject
Manifest
Config
Layer
subject
Manifest
Config
Layer

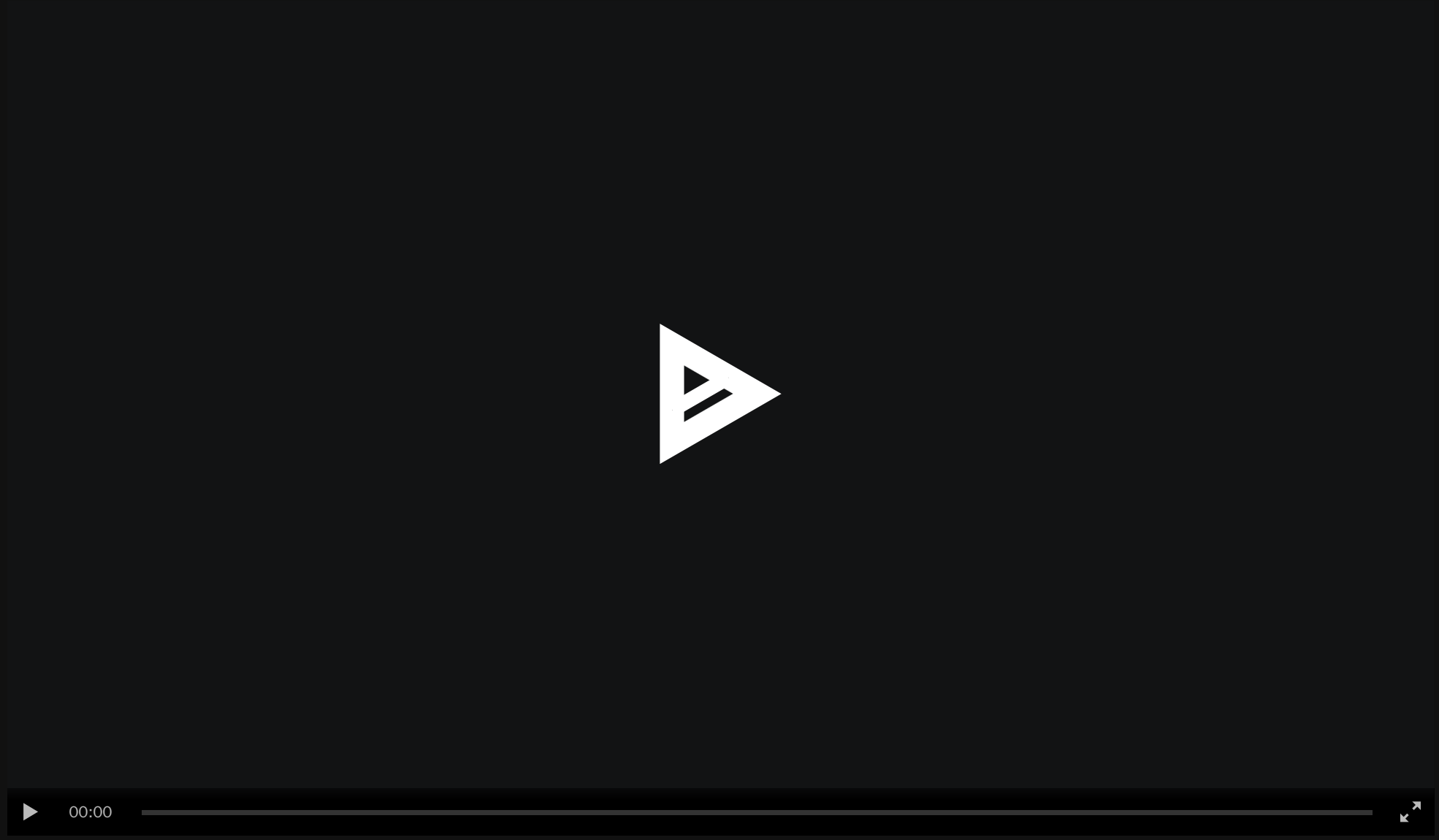
Referrers Response

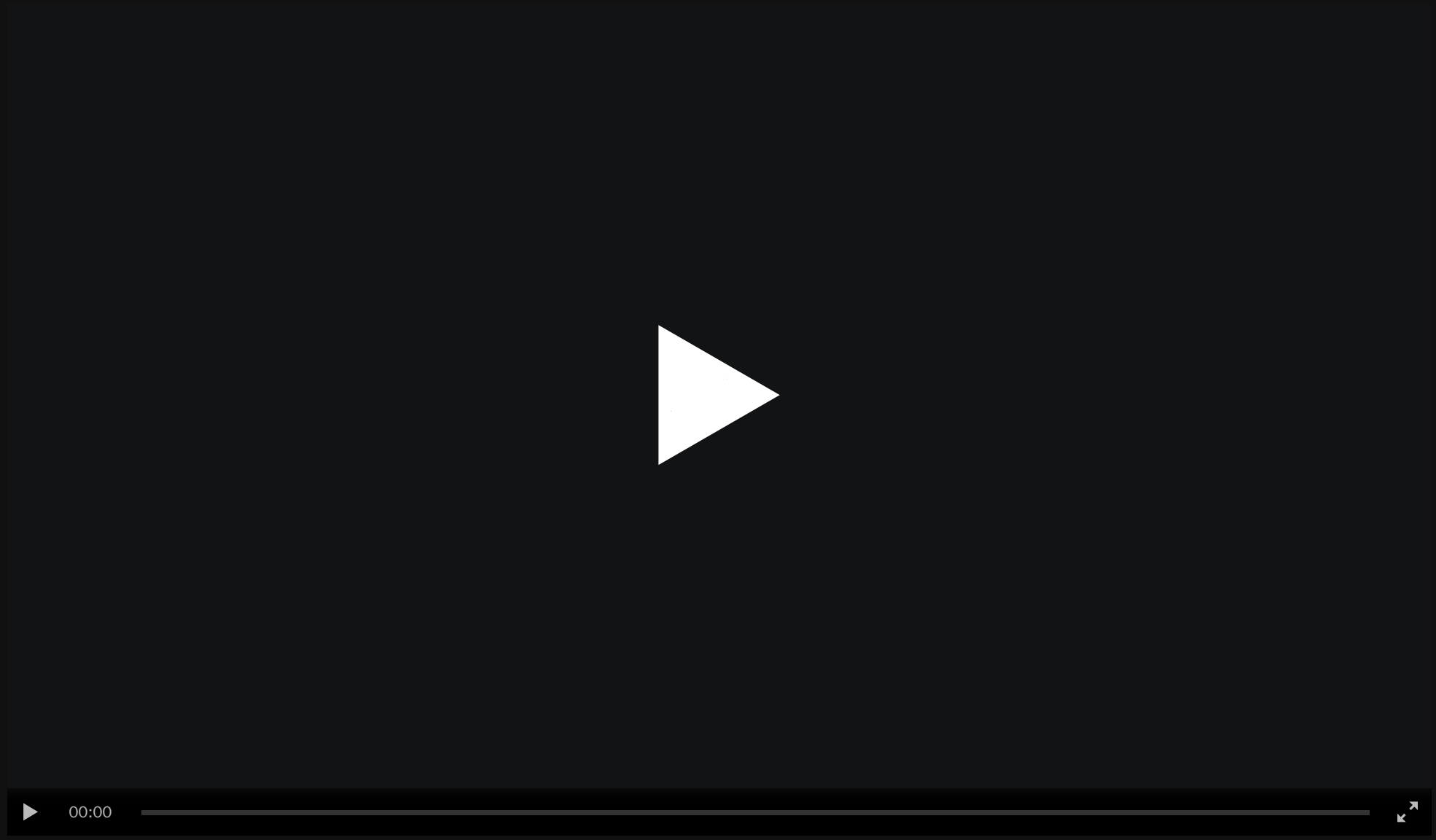
```
$ curl -I -H 'Accept: application/vnd.docker.distribution.manifest.list.v2+json' -s \
  http://localhost:5000/v2/$repo/manifests/alpine | grep Docker-Content-Digest
Docker-Content-Digest: sha256:f271e74b17ced29b915d351685fd4644785c6d1559dd1f2d4189a5e851ef753a

$ ref_tag=sha256-f271e74b17ced29b915d351685fd4644785c6d1559dd1f2d4189a5e851ef753a

$ curl -H 'Accept: application/vnd.oci.image.index.v1+json' -s \
  http://localhost:5000/v2/$repo/manifests/${ref_tag} | jq .
{
  "mediaType": "application/vnd.oci.image.index.v1+json",
  "manifests": [
    {
      "mediaType": "application/vnd.oci.image.manifest.v1+json",
      "size": 680,
      "digest": "sha256:71bc43bc5288af84620c535ec04f31080d09c4dce9da6c0011b425d13c25e4bb",
      "annotations": {
        "org.opencontainers.artifact.created": "2023-02-01T09:10:11Z"
      },
      "artifactType": "application/vnd.example.ebom"
    }
...
```

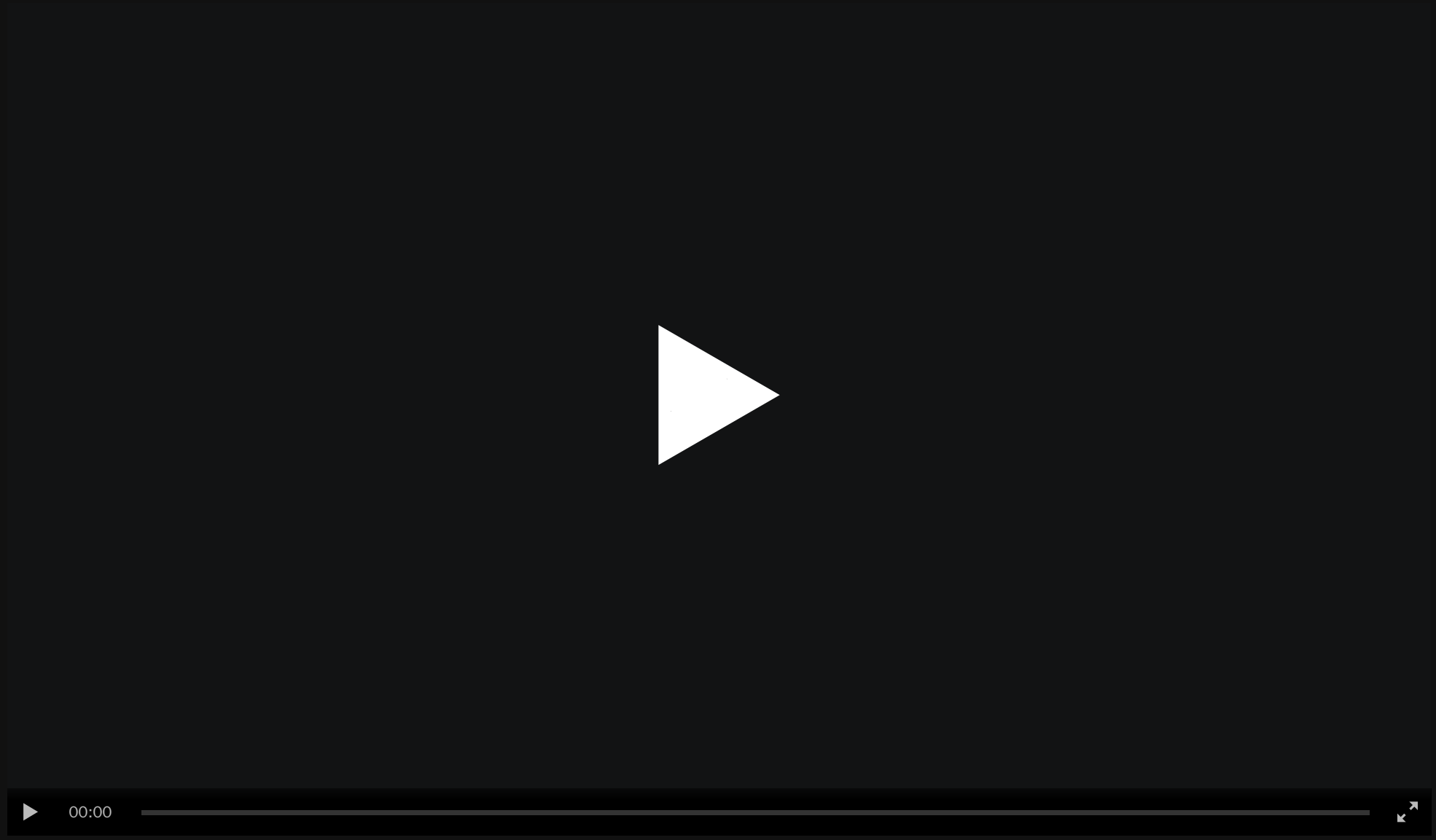
Subject and Artifact Type Fields

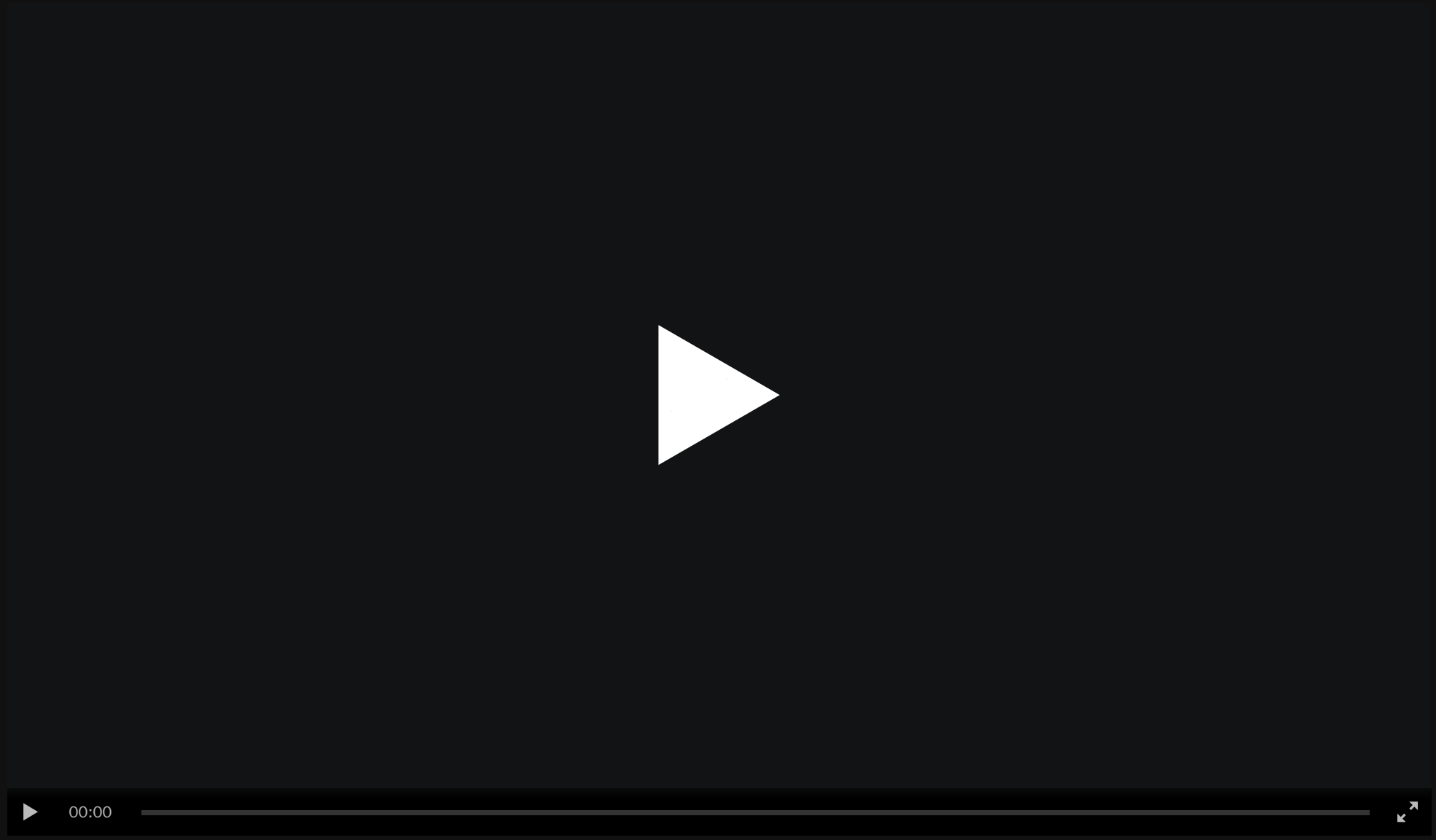
```
$ curl ... http://localhost:5000/v2/$repo/manifests/$mdigest | jq .
{
  "mediaType": "application/vnd.oci.image.manifest.v1+json",
  "artifactType": "application/vnd.example.ebom",
  "config": {
    "mediaType": "application/vnd.oci.scratch.v1+json",
    "size": 2,
    "digest": "sha256:44136fa355b3678a1146ad16f7e8649e94fb4fc21fe77e8310c060f61caaff8a"
  },
  "layers": [{...
      "mediaType": "application/vnd.example.ebom",
      "digest": "sha256:ea706edf61ef640bcdf3c9ac9045c28446e6b2d08541b9ad614c7267d0b87375"
  }],
  "annotations": {
    "org.opencontainers.artifact.created": "2023-02-01T09:10:11Z"
  },
  "subject": {
    "mediaType": "application/vnd.docker.distribution.manifest.list.v2+json",
    "size": 1638,
    "digest": "sha256:f271e74b17ced29b915d351685fd4644785c6d1559dd1f2d4189a5e851ef753a"
  }
}
```

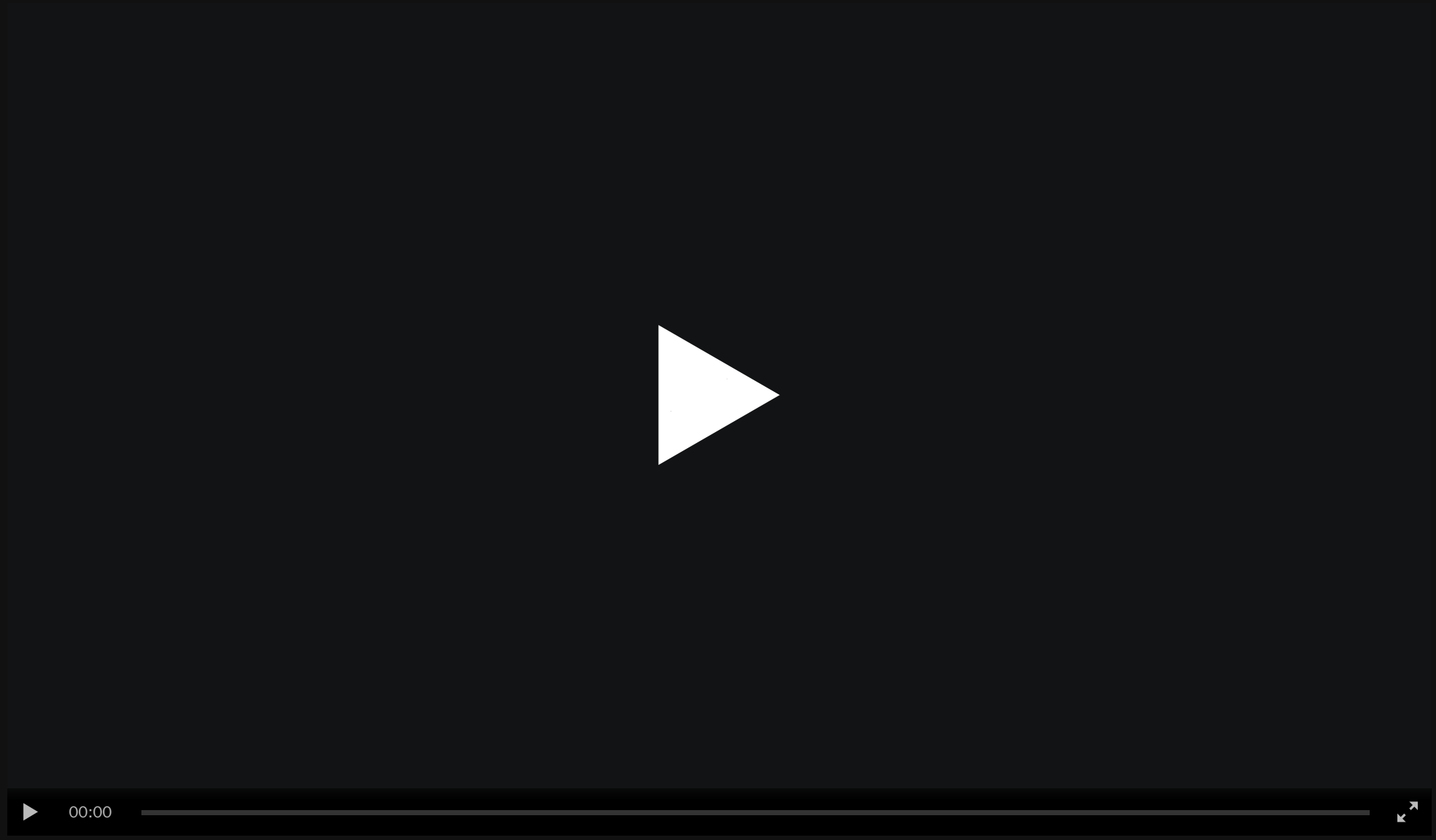
Demo


00:00


00:00


00:00


00:00


00:00

Current Status

- Release candidates:
  - image-spec 1.1.0-rc.3
  - distribution-spec 1.1.0-rc.2
- Image manifest now has an `artifactType`
- Ready for testing

Registries

- Registry support:
  - Adopted: zot, Harbor
  - Blocked: Docker Hub, ECR, GitLab
- Do not filter on unknown fields (subject and config media type)
- Enable the referrers API
  - Retroactively include manifests using the tag schema

Clients

- Client support: cosign, oras, and regclient
- Clients manage the fallback tag
- Pick appropriate artifactType values
- Use annotations responsibly

Thank You

github.com/sudo-bmitch/presentations

Brandon Mitchell
Mastodon: @bmitch@fosstodon.org
Twitter: @sudo_bmitch
GitHub: sudo-bmitch