dockercon 23

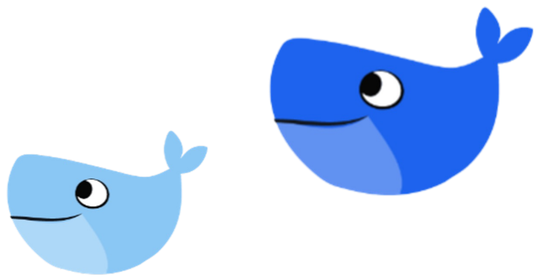
Container Images

Interactive Deep Dive

Yves Brissaud

Senior Software Engineer | Docker

𝕏 @_crev_

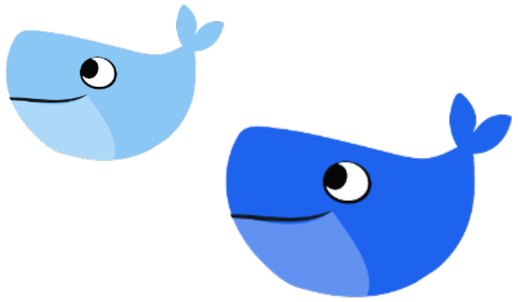


Yves Brissaud

Senior Software Engineer | Docker

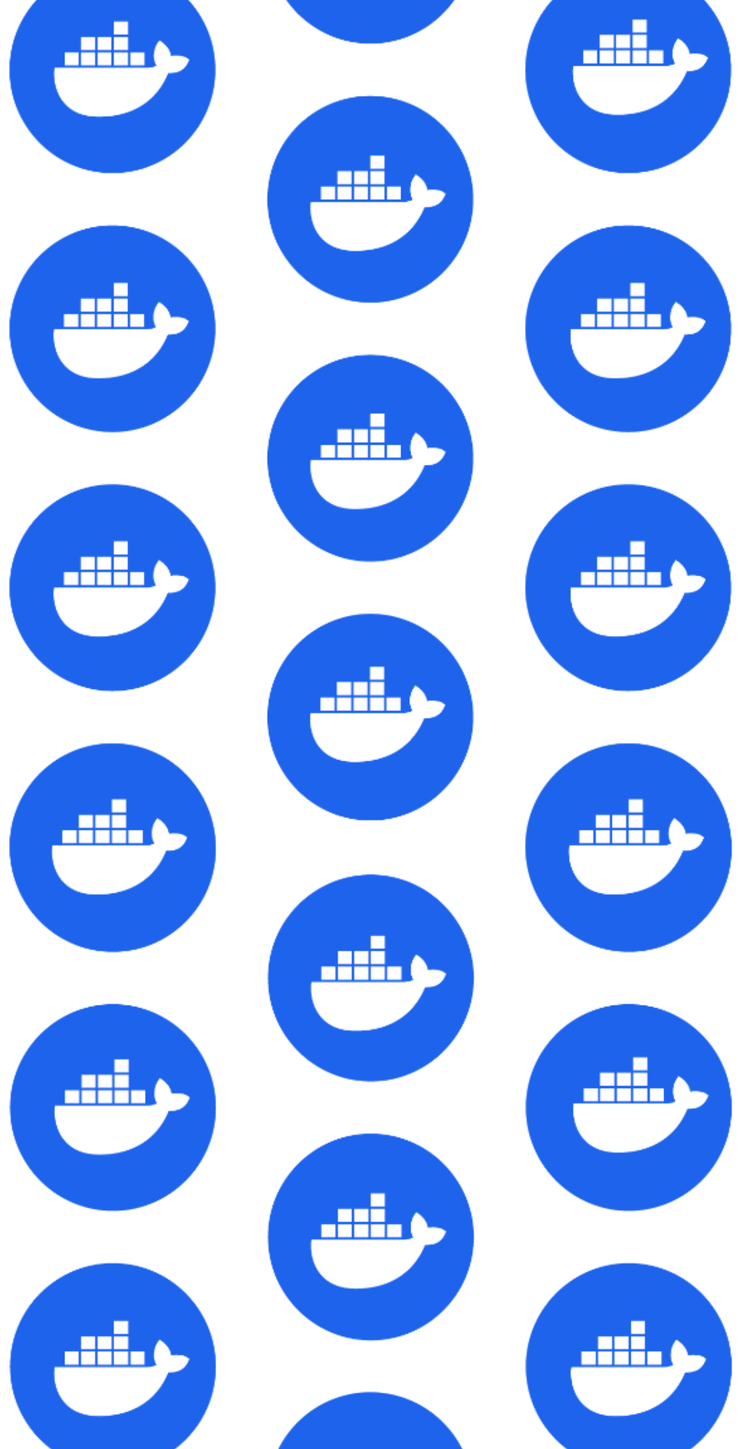
𝕏 @_crev_

docker scout



00

Intro



Why To Care About Images?

- Docker Hub Registry → Images
- Publishers (DVP, DSOS, ...) Analytics → Images, tags, pull
- Docker scout → Images, tags, images internal

What this talk is (not) about

✓ Build & Inspect multi-platform image
✓ Push & registry storage
✓ Pull & tags
✓ Update and new tags
✓ Beyond “images”

✕ Image specifications by the book
https://github.com/opencontainers/image-spec

https://www.pexels.com/photo/close-up-shot-of-keys-on-a-red-surface-2882687/

Materials

Slides available:

https://speakerdeck.com/eunomie/container-images

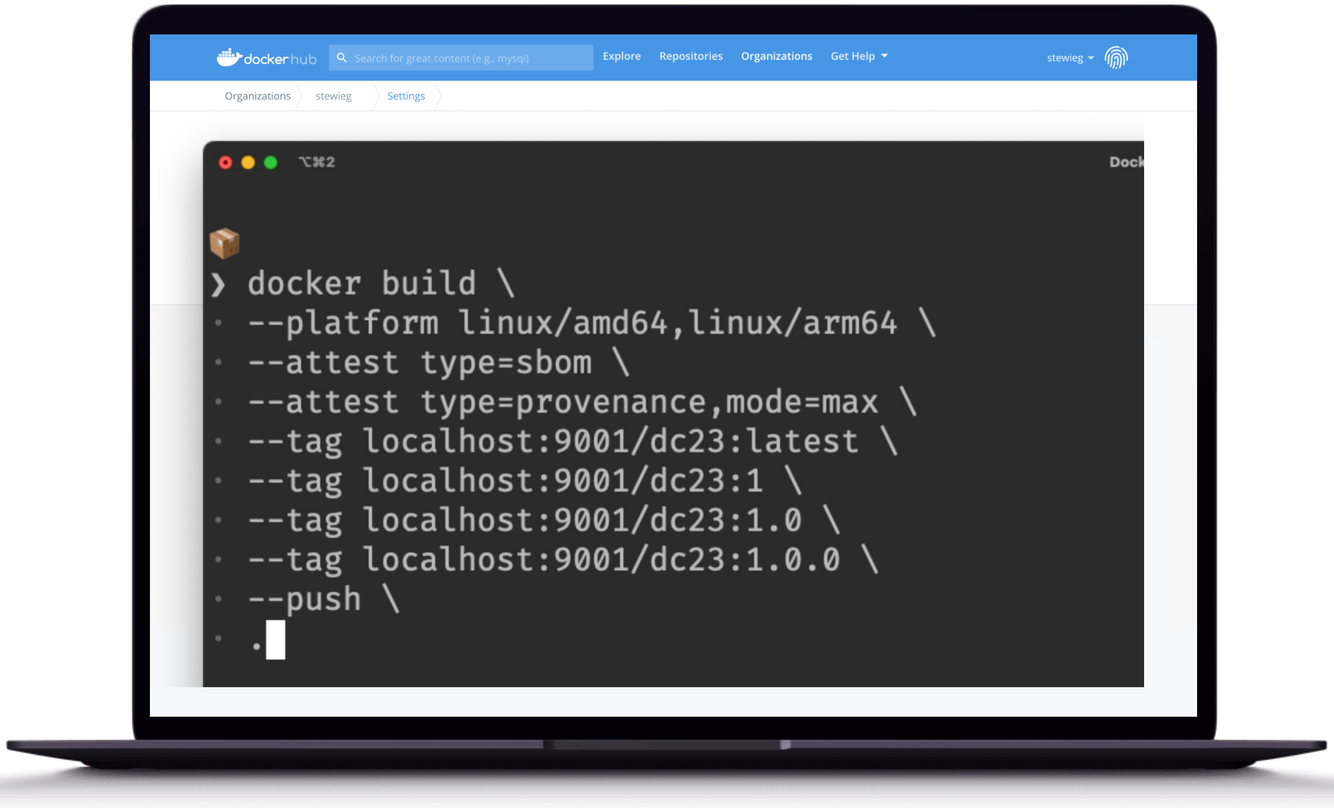
01

Build

https://www.pexels.com/photo/person-holding-yellow-and-pink-lego-blocks-298825/

Let’s build an image

- Using a base image
- For multiple architectures
- Including SSC materials
- Published on different tags



02

Inspect

https://www.pexels.com/photo/shallow-focus-photography-of-magnifying-glass-with-black-frame-924676/

Extract and Inspect

- Extract the image to a local directory
- Explore starting with *index.json*

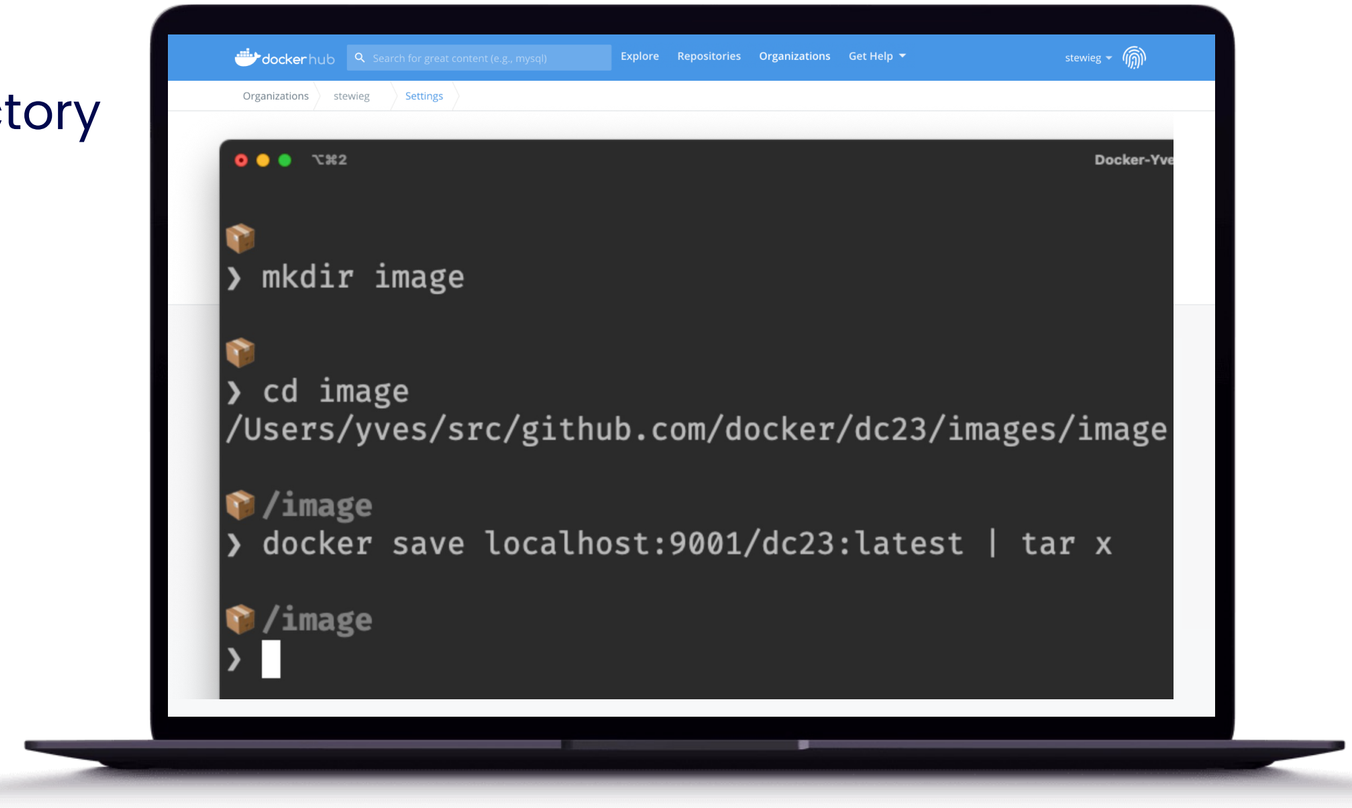


Image Index

application/vnd.oci.image.index.v1+json

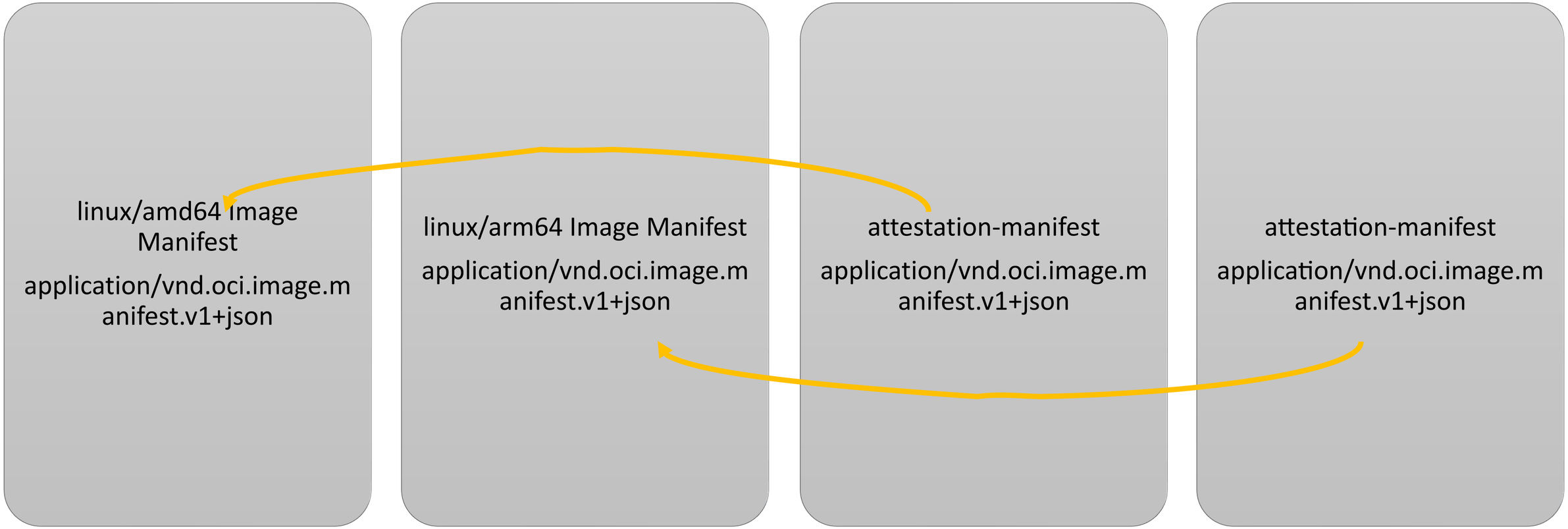



Image Index (application/vnd.oci.image.index.v1+json)
linux/amd64
linux/arm64
attestation-manifest
attestation-manifest
Config Blob
Layer
Layer
...

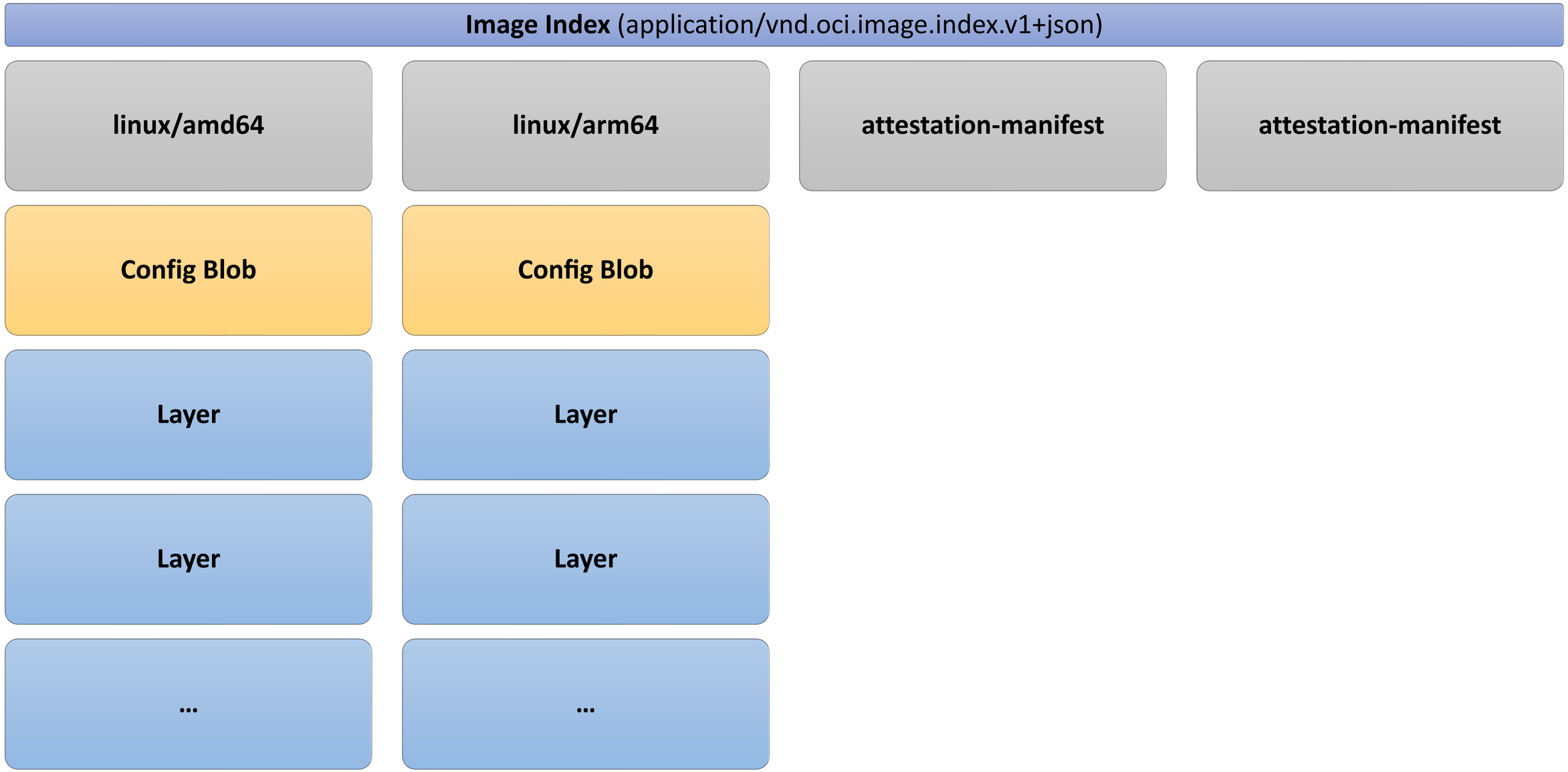


Image Index (application/vnd.oci.image.index.v1+json)
linux/amd64
linux/arm64
attestation-manifest
attestation-manifest
Config Blob
Config Blob
Layer
Layer
Layer
Layer
...
...

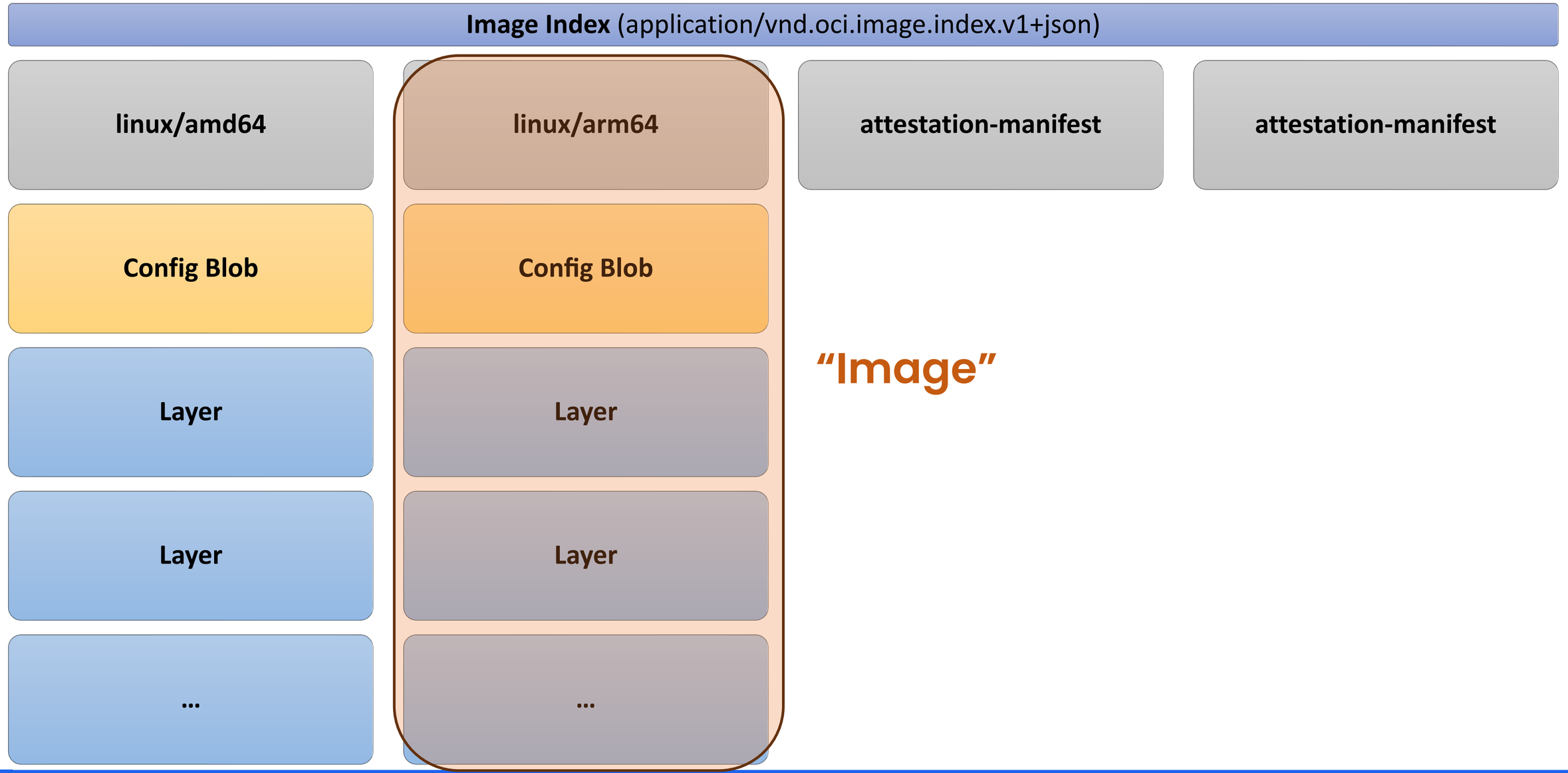
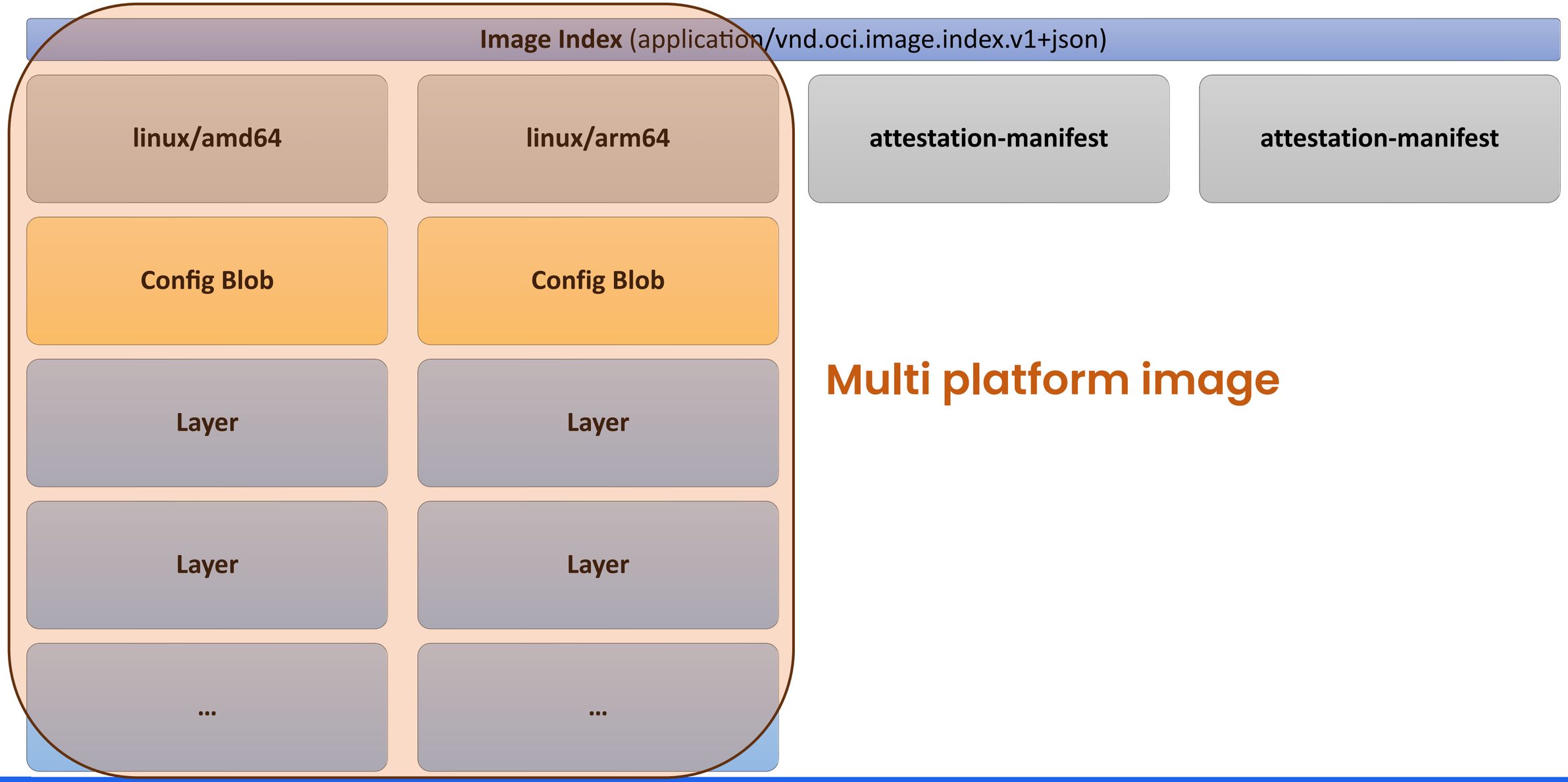


Image Index (application/vnd.oci.image.index.v1+json)
linux/amd64
linux/arm64
attestation-manifest
attestation-manifest
Config Blob
Config Blob
Layer
Layer
Layer
Layer
...
...
“Image”



Multi platform image

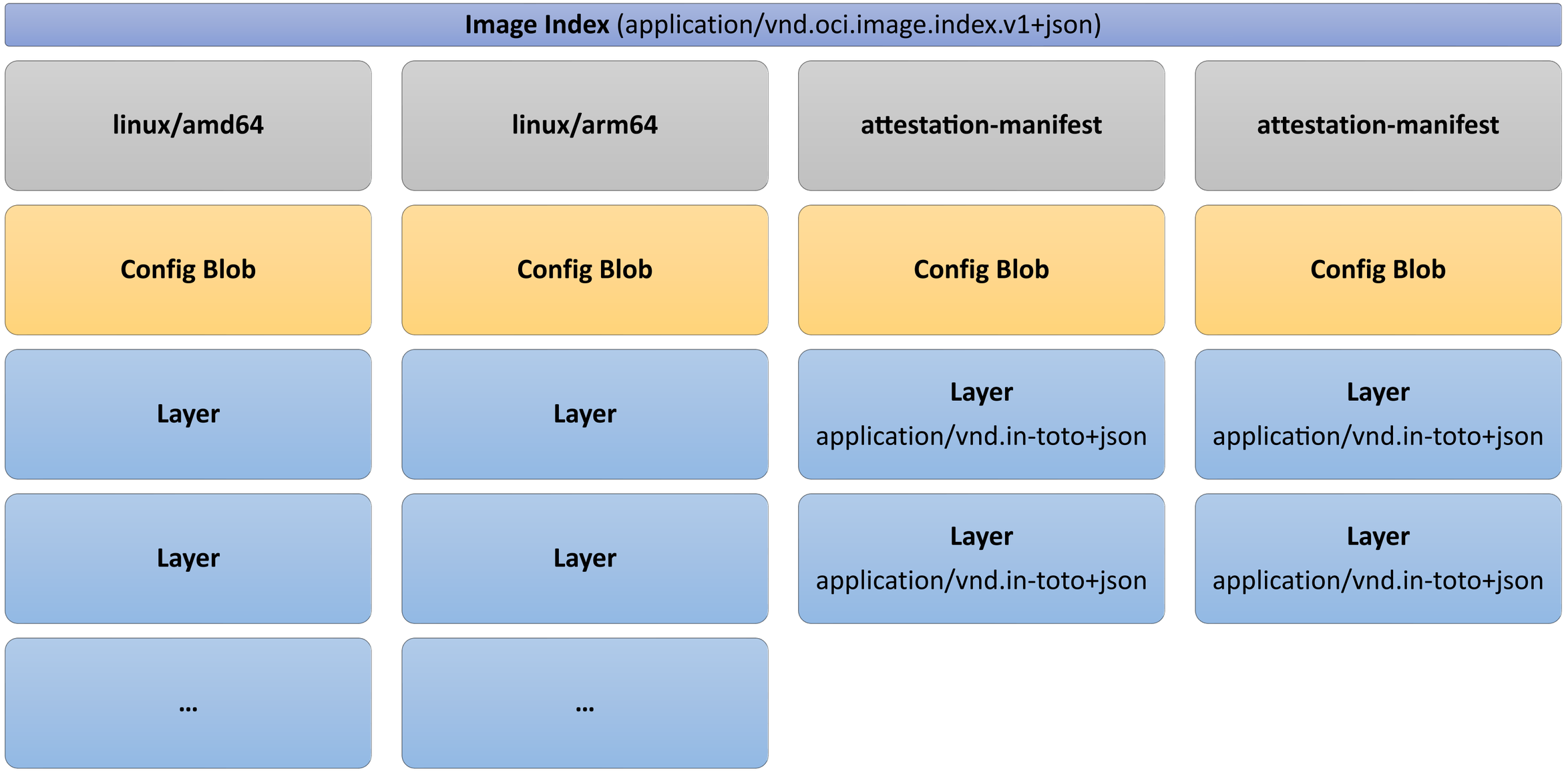


Image Index (application/vnd.oci.image.index.v1+json)
linux/amd64
linux/arm64
attestation-manifest
attestation-manifest
Config Blob
Config Blob
Config Blob
Config Blob
Layer
Layer
Layer
application/vnd.in-toto+json
Layer
application/vnd.in-toto+json
Layer
Layer
Layer
application/vnd.in-toto+json
Layer
application/vnd.in-toto+json
...
...

03

Push

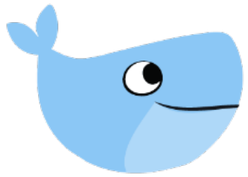

https://www.pexels.com/photo/photo-of-man-pushing-hay-bale-2600312/

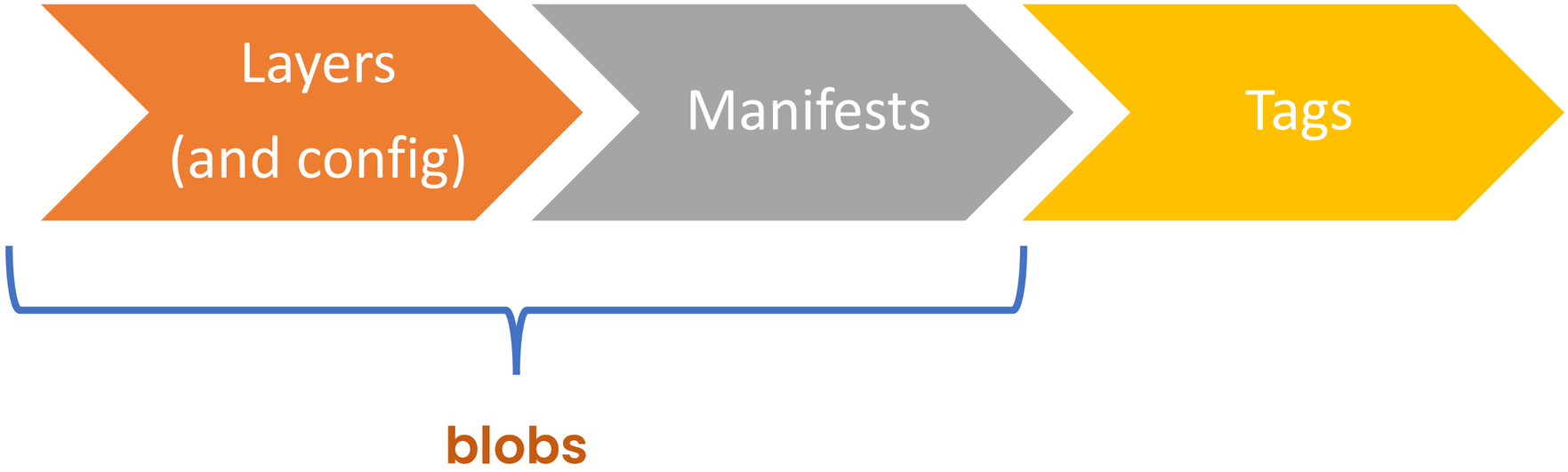
Why to push to a registry?

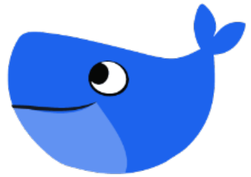
Why not just to share archives?

- ✓ Deduplication
- ✓ “Metadata” (tags)
- ✓ Versions

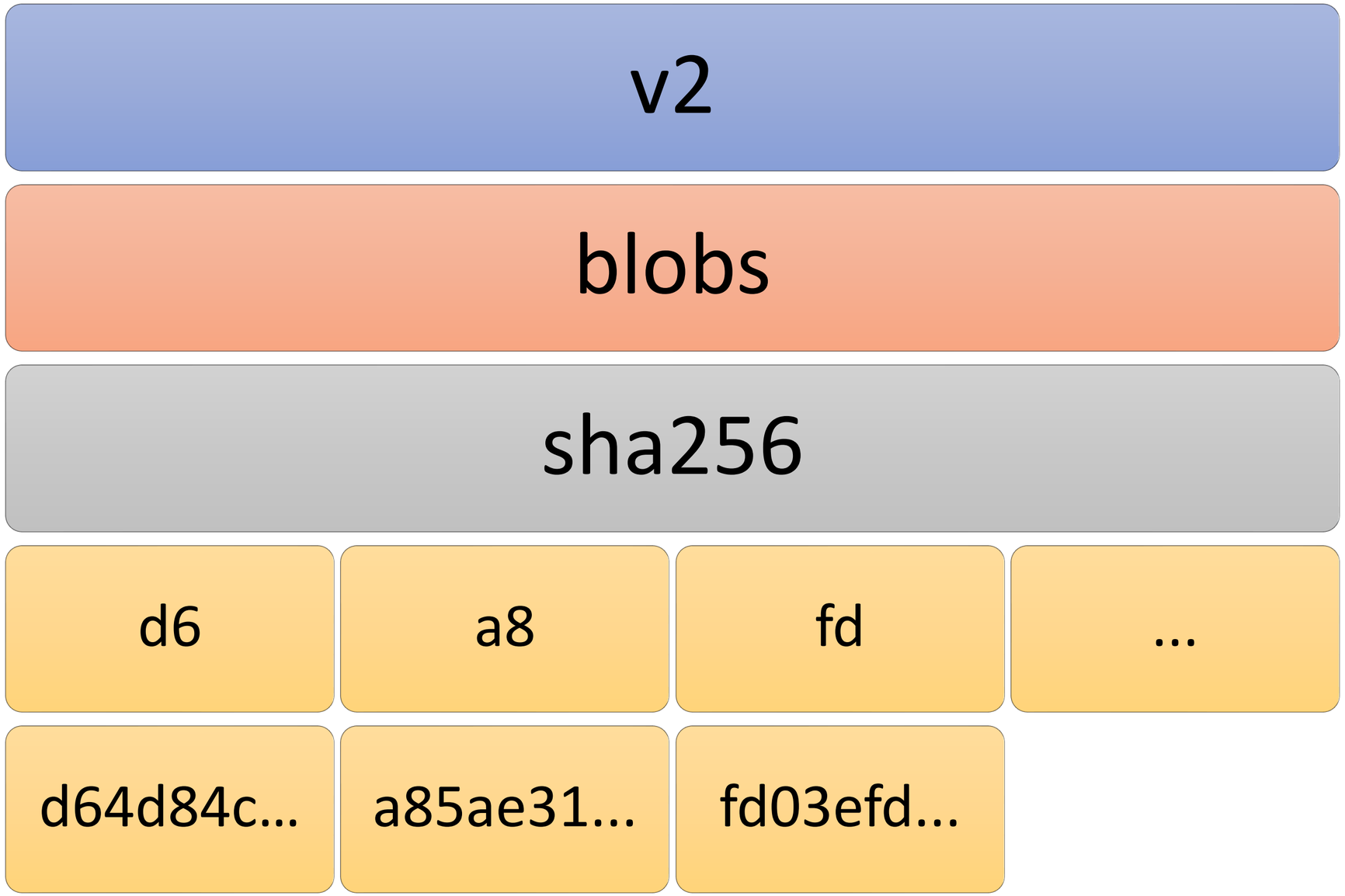
https://www.pexels.com/photo/question-mark-on-crumpled-paper-5428826/



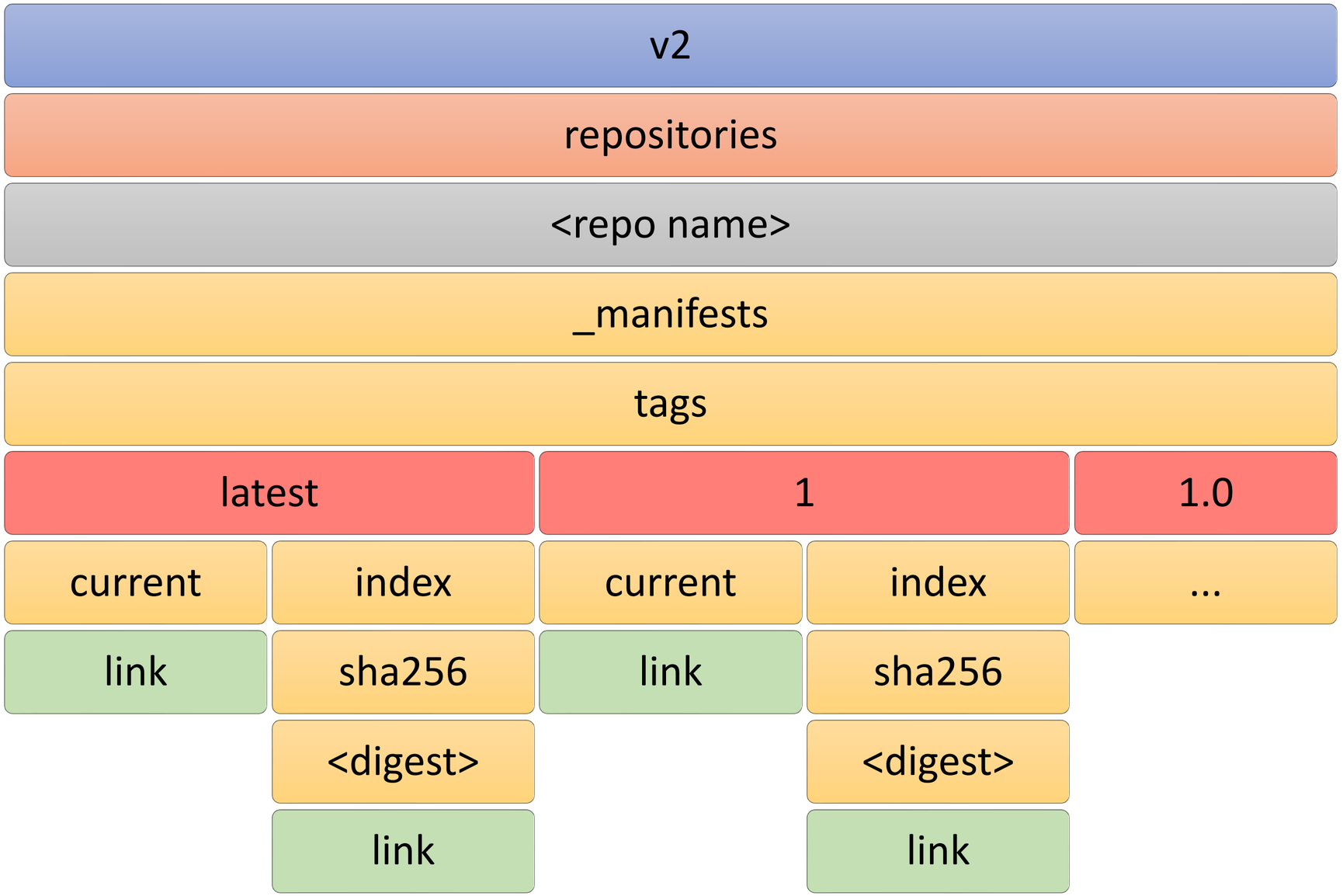

Layers
(and config)
Manifests
Tags
blobs



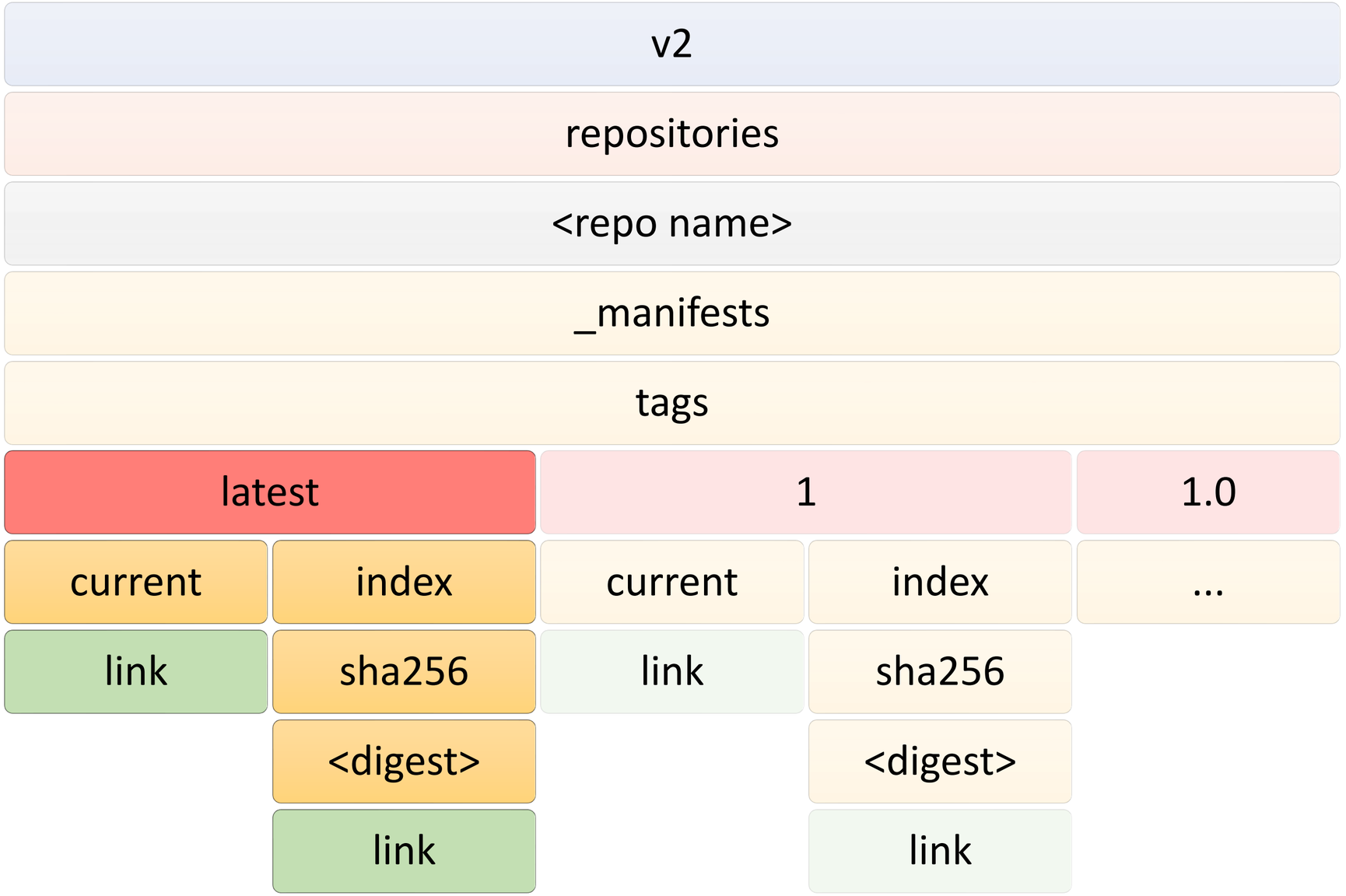
Registry View - Blobs



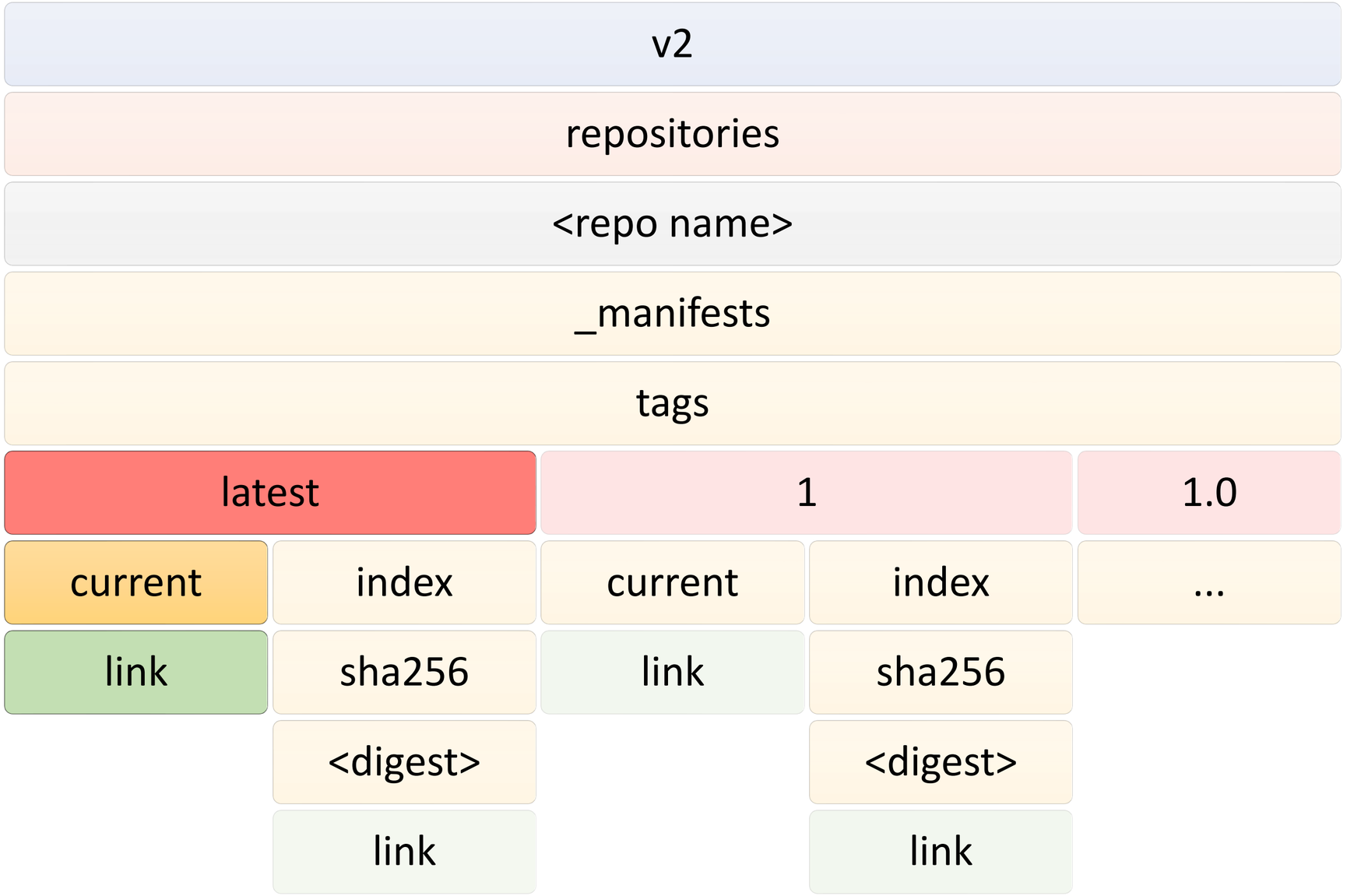
Registry View - Tags



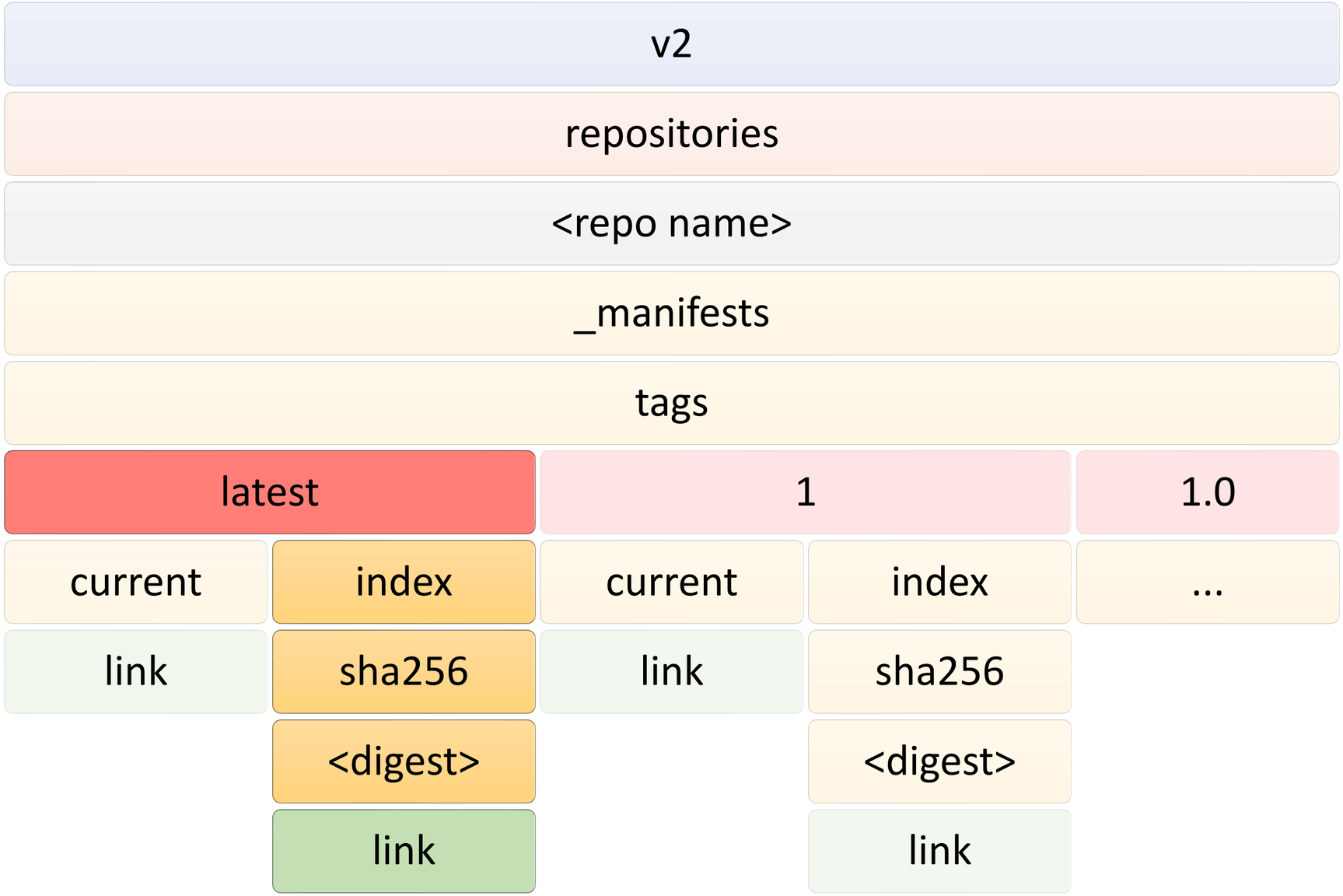
Registry View - Tags



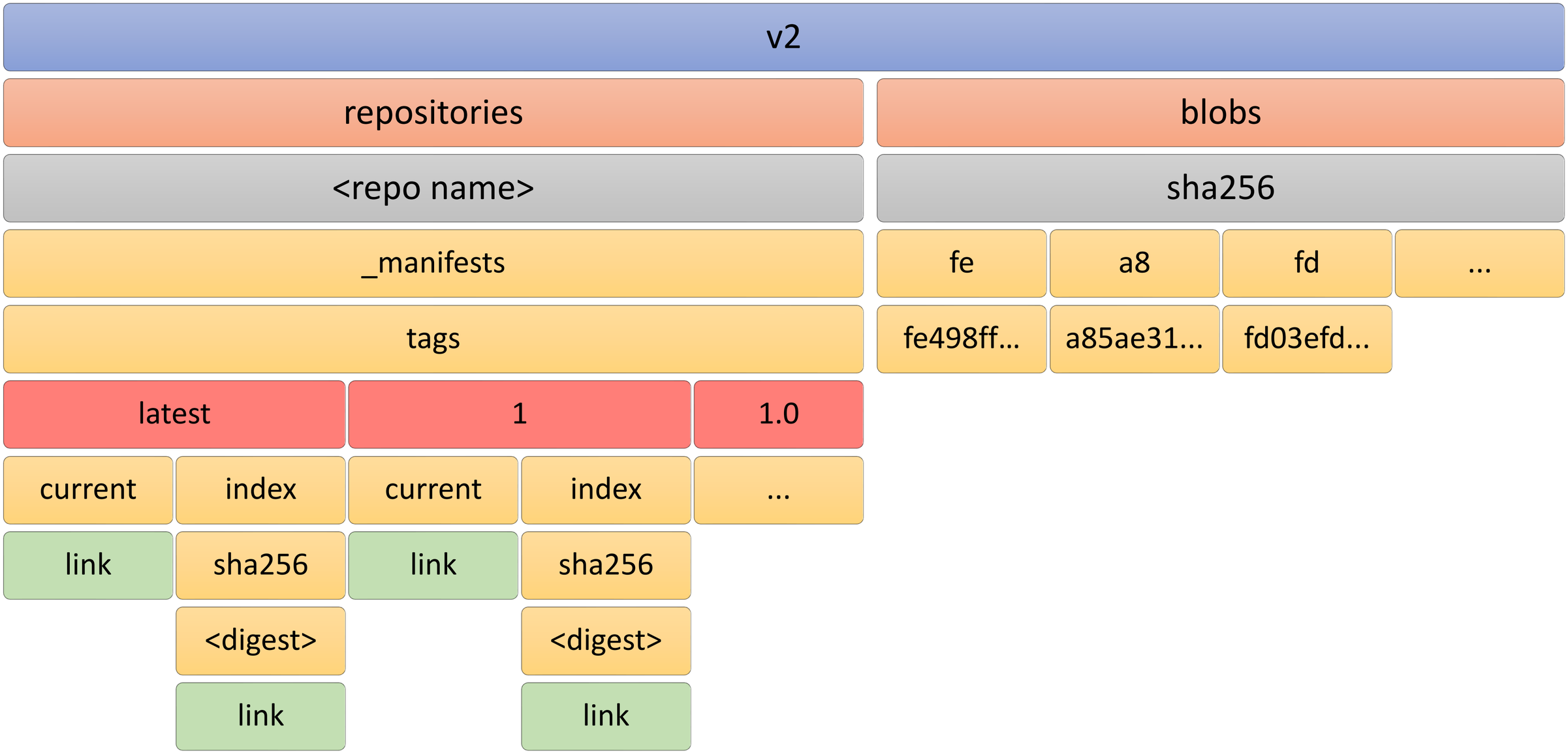
my/image:latest



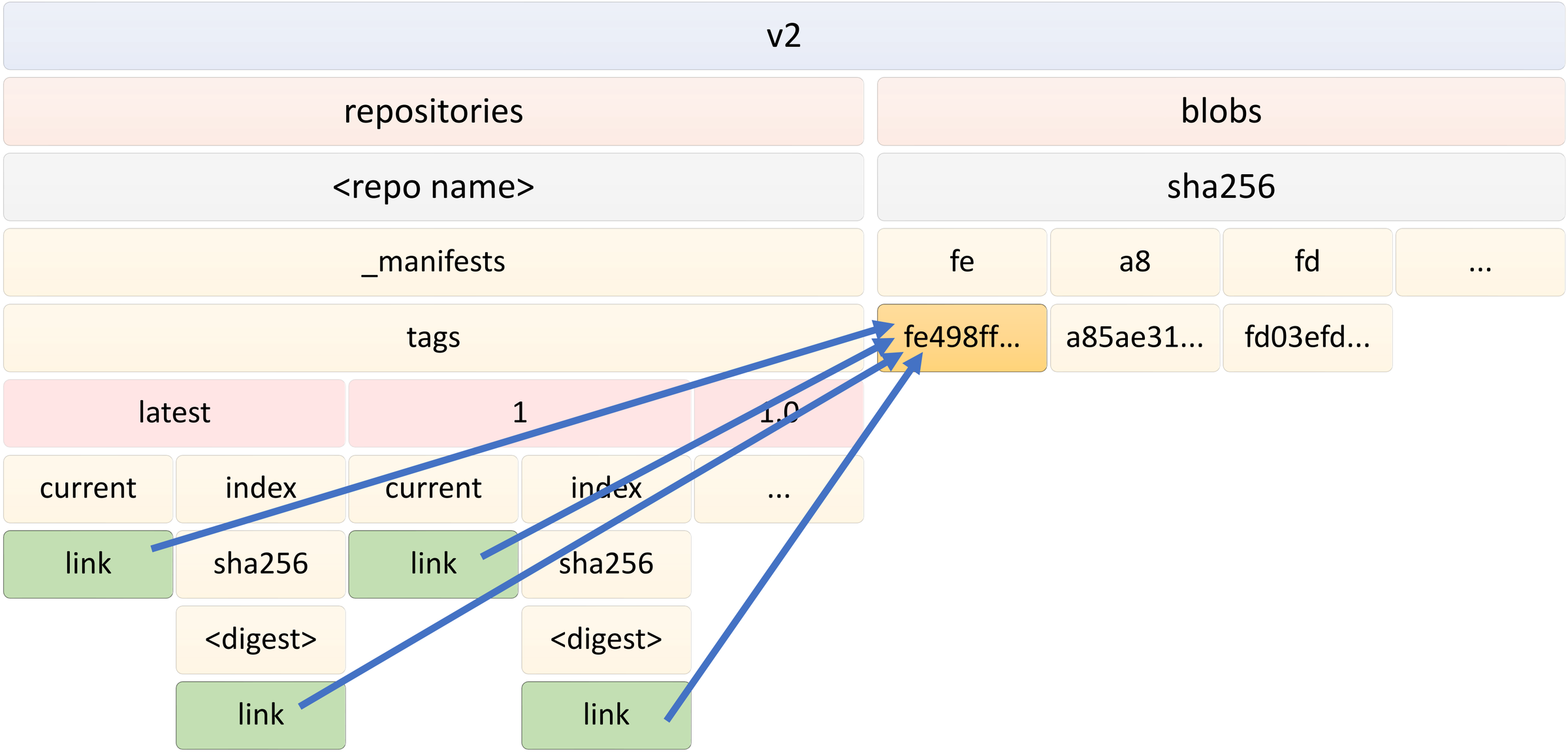
my/image:latest@sha256:...



Registry View



Registry View



04

Pull

https://www.pexels.com/photo/faceless-physician-touching-door-handle-in-building-6097735/

Pull *linux/amd64* version of *latest*

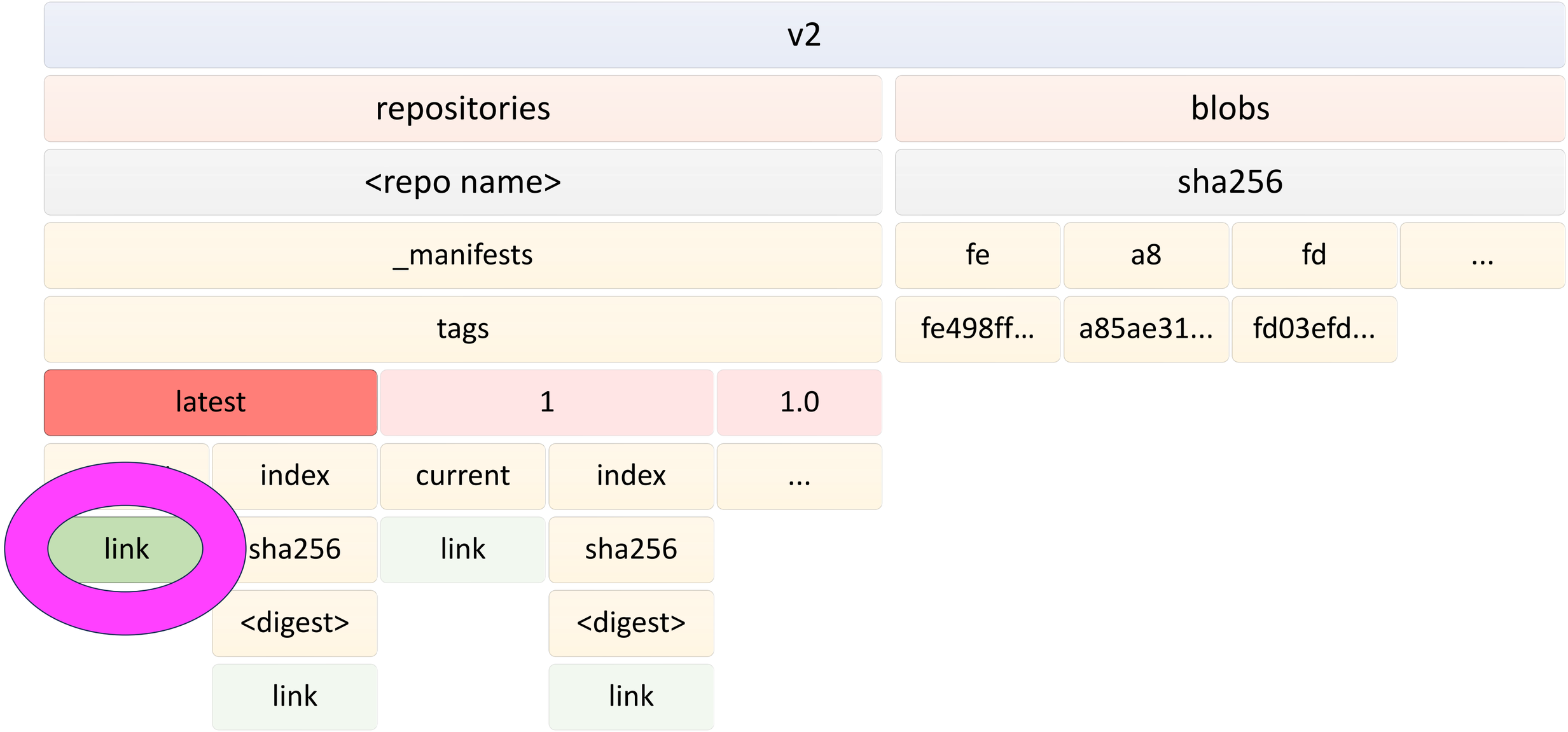
1. Convert *tag* to *digest*
2. Select the image for the right platform
3. Download *config* and *layer* blobs

Convert *tag* to *digest*

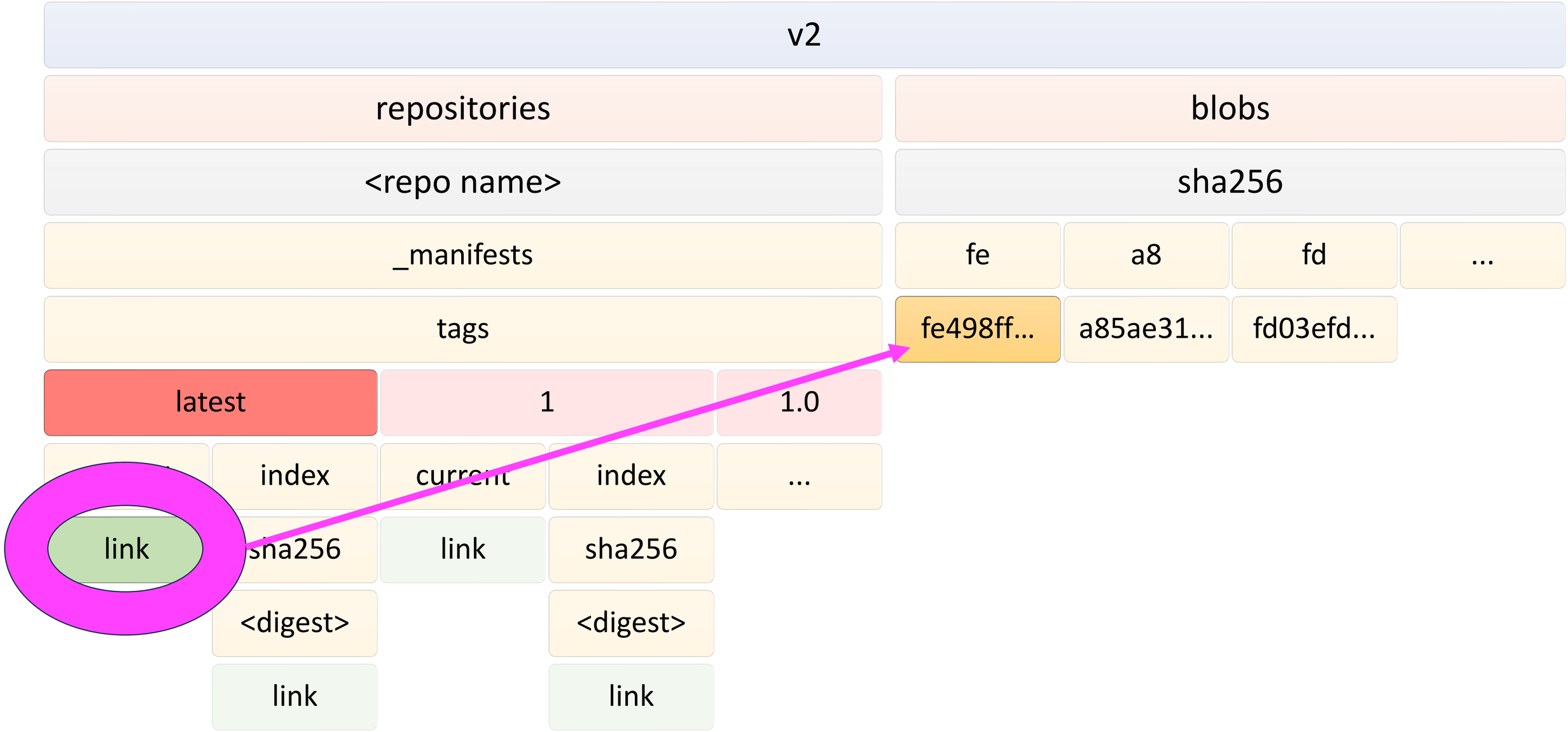
HEAD
/v2/dc23/manifests/latest

```
HTTP/1.1 200 OK
content-type:
application/vnd.oci.image.index.v1+json
docker-content-digest:
sha256:5d0cbb38e39004b97dad3beb62fdde74e51f2f
dcec80f547baa7ee5ed556cb4c
docker-distribution-api-version: registry/2.0
```

Convert *tag* to *digest*



Convert *tag* to *digest*



Find the right manifest

GET
/v2/dc23/manifests/sha256:.....


```
{
  "mediaType": "application/vnd.oci.image.index.v1+json",
  "schemaVersion": 2,
  "manifests": [
    {
    "mediaType": "application/vnd.oci.image.manifest.v1+json",
    "digest":
"sha256:d64d84c3e5d2aa34243921261687bf482631dbd1d34c4890e94a13f392d9
bfa1",
    "size": 1812,
    "platform": {
      "architecture": "amd64",
      "os": "linux"
    }
  },
```

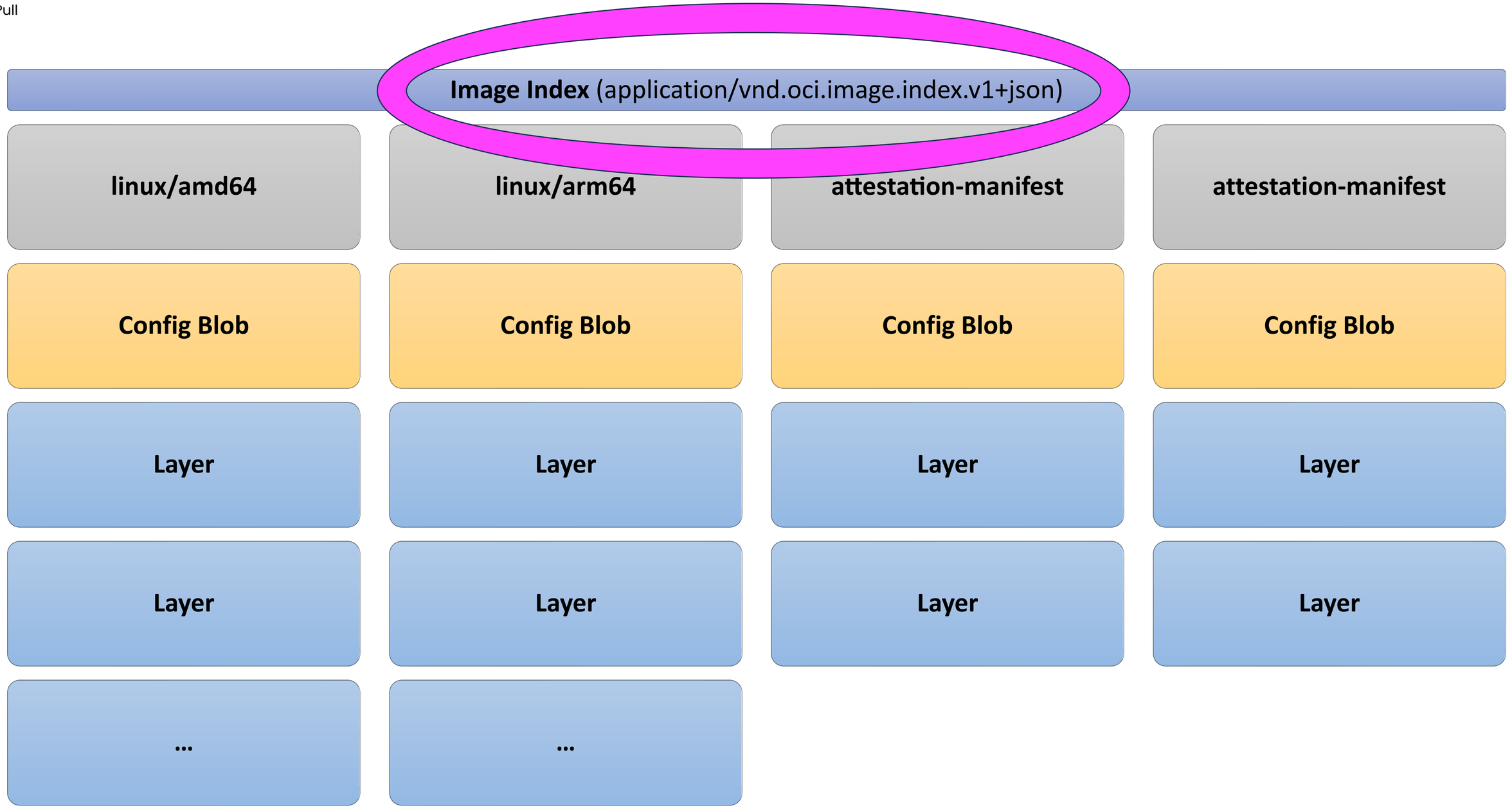


Image Index (application/vnd.oci.image.index.v1+json)
linux/amd64
linux/arm64
attestation-manifest
attestation-manifest
Config Blob
Config Blob
Config Blob
Config Blob
Layer
Layer
Layer
Layer
Layer
Layer
Layer
Layer
...
...

Find the right manifest

GET
/v2/dc23/manifests/sha256:d64...


```
{
  "mediaType": "application/vnd.oci.image.manifest.v1+json",
  "schemaVersion": 2,
  "config": {
    "mediaType": "application/vnd.oci.image.config.v1+json",
    "digest":
"sha256:e999e4251aa2c2f7c0d8846883ea6e6dace050f5c07da7103137f4972df4
e97f",
    "size": 6896
  },
  "layers": [
    {
      "mediaType": "application/vnd.oci.image.layer.v1.tar+gzip",
      "digest":
"sha256:9398808236ffac29e60c04ec906d8d409af7fa19dc57d8c65ad167e9c496
7006",
      "size": 3378609
    },
```

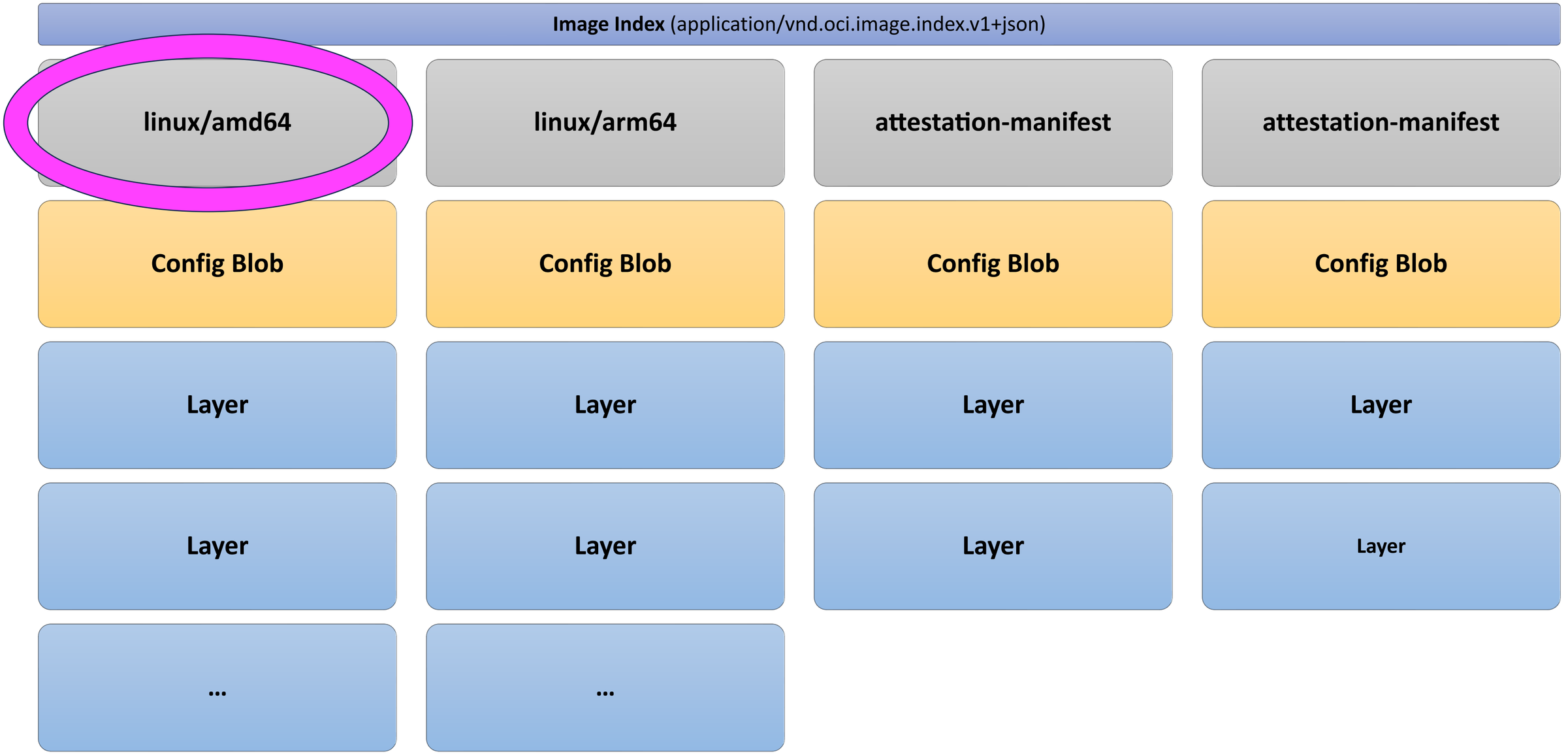


Image Index (application/vnd.oci.image.index.v1+json)
linux/amd64
linux/arm64
attestation-manifest
attestation-manifest
Config Blob
Config Blob
Config Blob
Config Blob
Layer
Layer
Layer
Layer
Layer
Layer
Layer
Layer
...
...

Download *config* and *layer* blobs

GET /v2/dc23/blobs/sha256:...

GET /v2/dc23/blobs/sha256:...

...

```
$ docker pull --platform linux/amd64 localhost:9001/dc23:latest
latest: Pulling from dc23
2651927a96a6: Download complete
83df69d10500: Download complete
10e1614aca69: Download complete
725b720f91d7: Download complete
Digest: sha256:5d0cbb3…
Status: Image is up to date for localhost:9001/dc23:latest
localhost:9001/dc23:latest
```

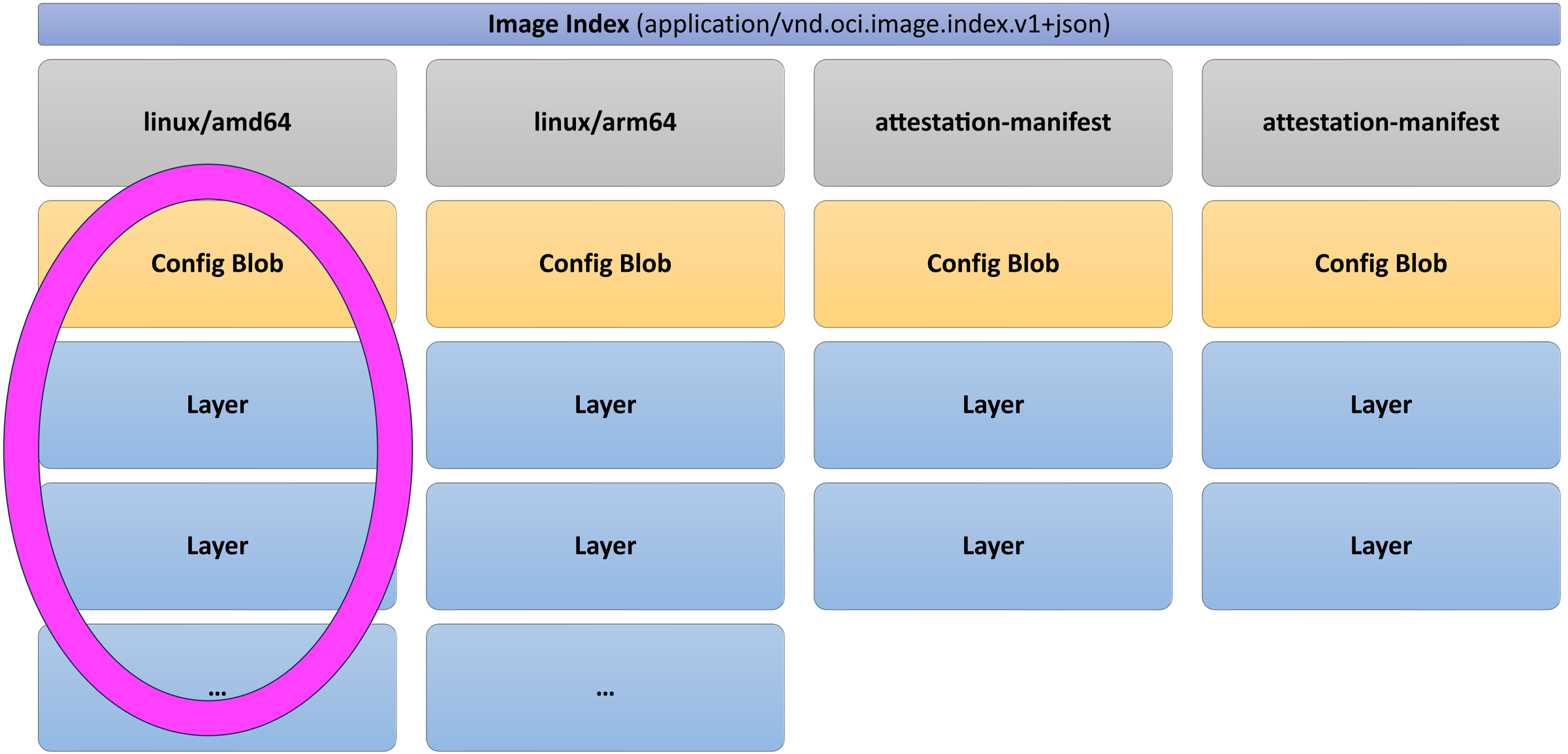


Image Index (application/vnd.oci.image.index.v1+json)
linux/amd64
linux/arm64
attestation-manifest
attestation-manifest
Config Blob
Config Blob
Config Blob
Config Blob
Layer
Layer
Layer
Layer
Layer
Layer
Layer
Layer
...
...

Requests

```
HEAD /v2/dc23/manifests/<tag name>
GET  /v2/dc23/manifests/<image index digest>
GET  /v2/dc23/manifests/<image manifest digest>
GET  /v2/dc23/manifests/blobs/<config digest>
GET  /v2/dc23/manifests/blobs/<layer digest>
GET  /v2/dc23/manifests/blobs/<layer digest>
…
```

```
→ current digest of tag
→ image index json file
→ image manifest json file for the platform
→ config blob by its digest
→ layer blob by its digest
→ layer blob by its digest
→ …
```

Pull *linux/amd64* version of ~~*latest 1*~~

1. Convert *tag* to *digest*
2. Select the image for the right platform
3. Download *config* and *layer* blobs

Pull *linux/amd64* version of ~~*latest 1*~~

1. Convert *tag* to *digest*
2. ~~Select the image for the right platform~~
3. ~~Download *config* and *layer* blobs~~

1. Same Digest!
2. Manifests already downloaded
3. Blobs already downloaded

Requests

HEAD /v2/dc23/manifests/<tag name>

~~GET /v2/dc23/manifests/<image index digest>~~

~~GET /v2/dc23/manifests/<image manifest digest>~~

~~GET /v2/dc23/manifests/blobs/<config digest>~~

~~GET /v2/dc23/manifests/blobs/<layer digest>~~

~~GET /v2/dc23/manifests/blobs/<layer digest>~~

…

→ current digest of tag

~~→ image index json file~~

~~→ image manifest json file for the platform~~

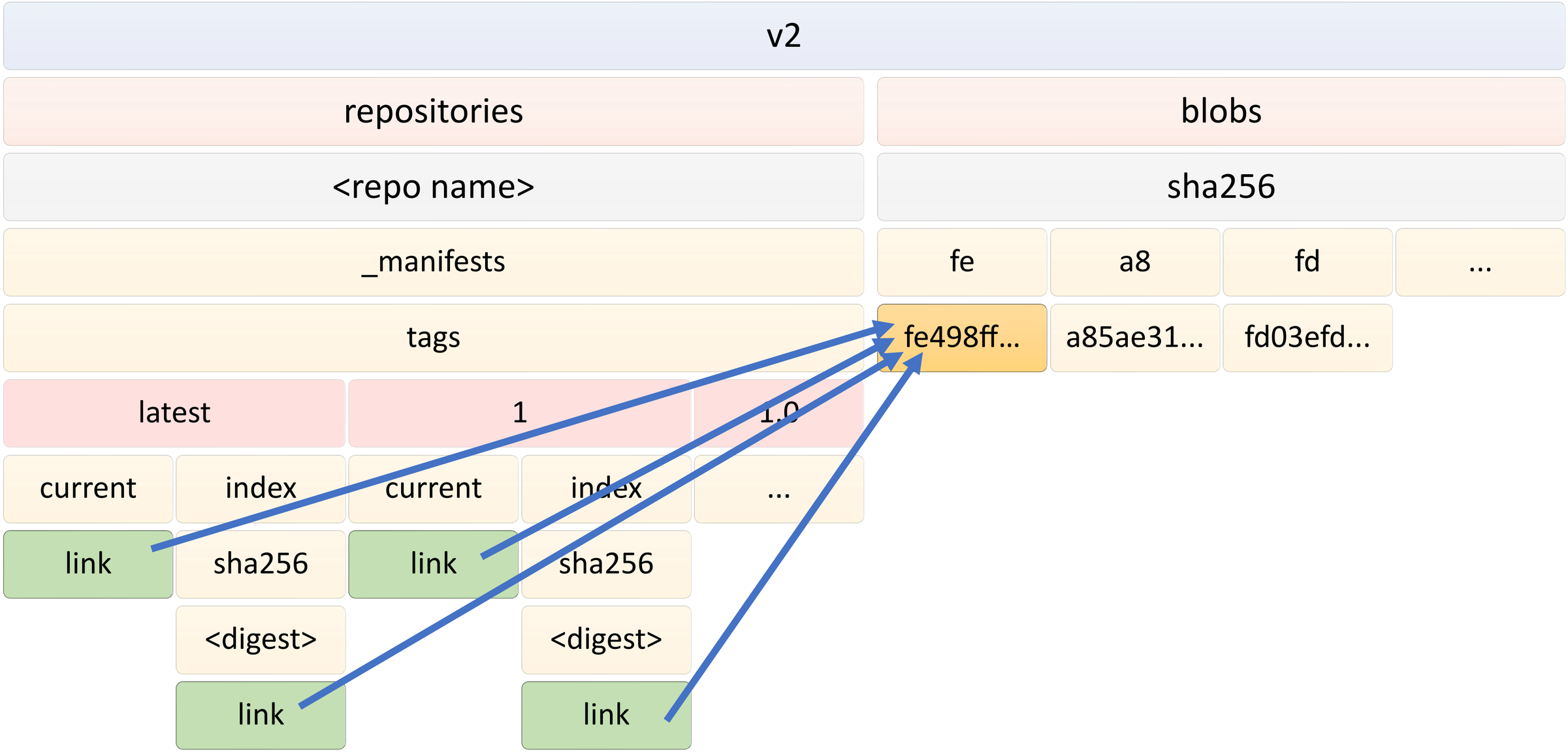
~~→ config blob by its digest~~

~~→ layer blob by its digest~~

~~→ layer blob by its digest~~

→ …

Different tags, same digest



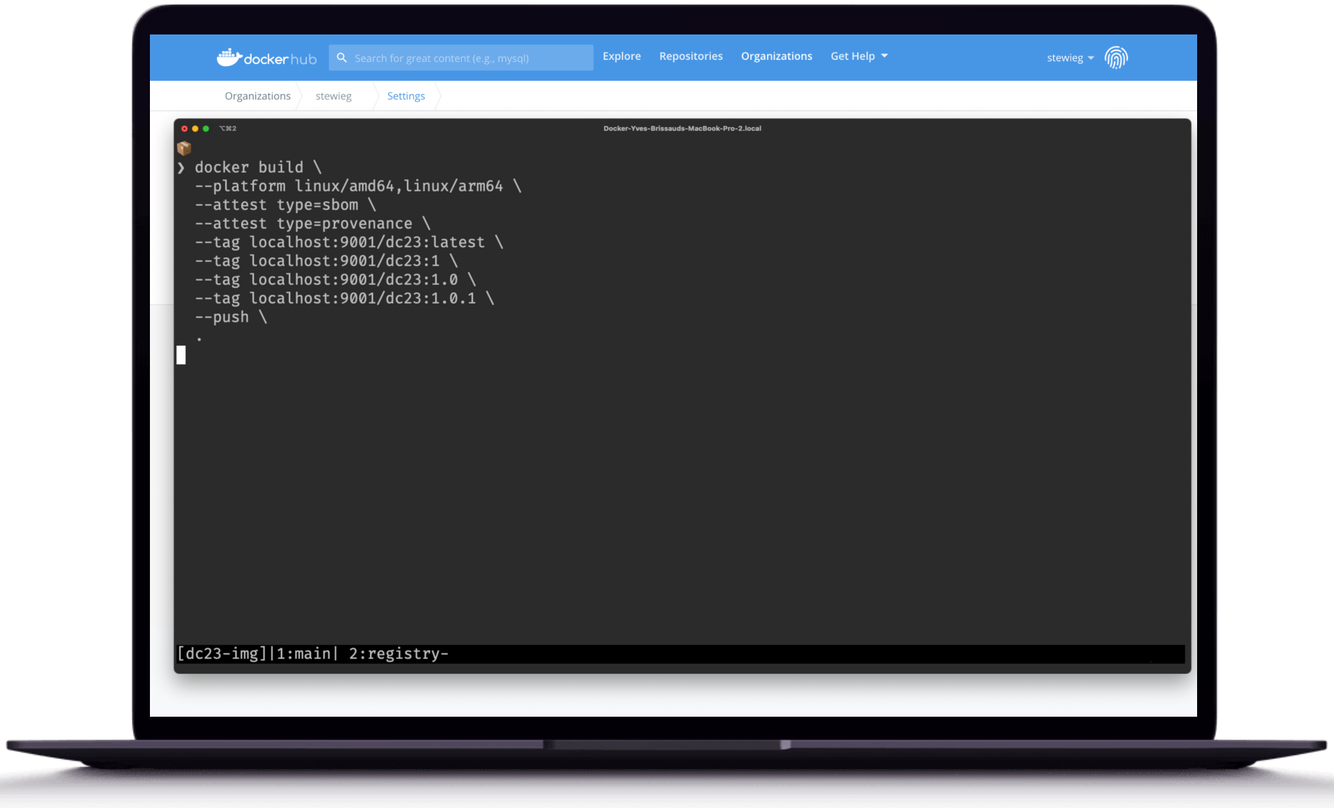
05

New Version

https://www.pexels.com/photo/rewrite-edit-text-on-a-typewriter-3631711/

Let's build a *new* image

- Edit one single layer
- Build on same *and* different tags



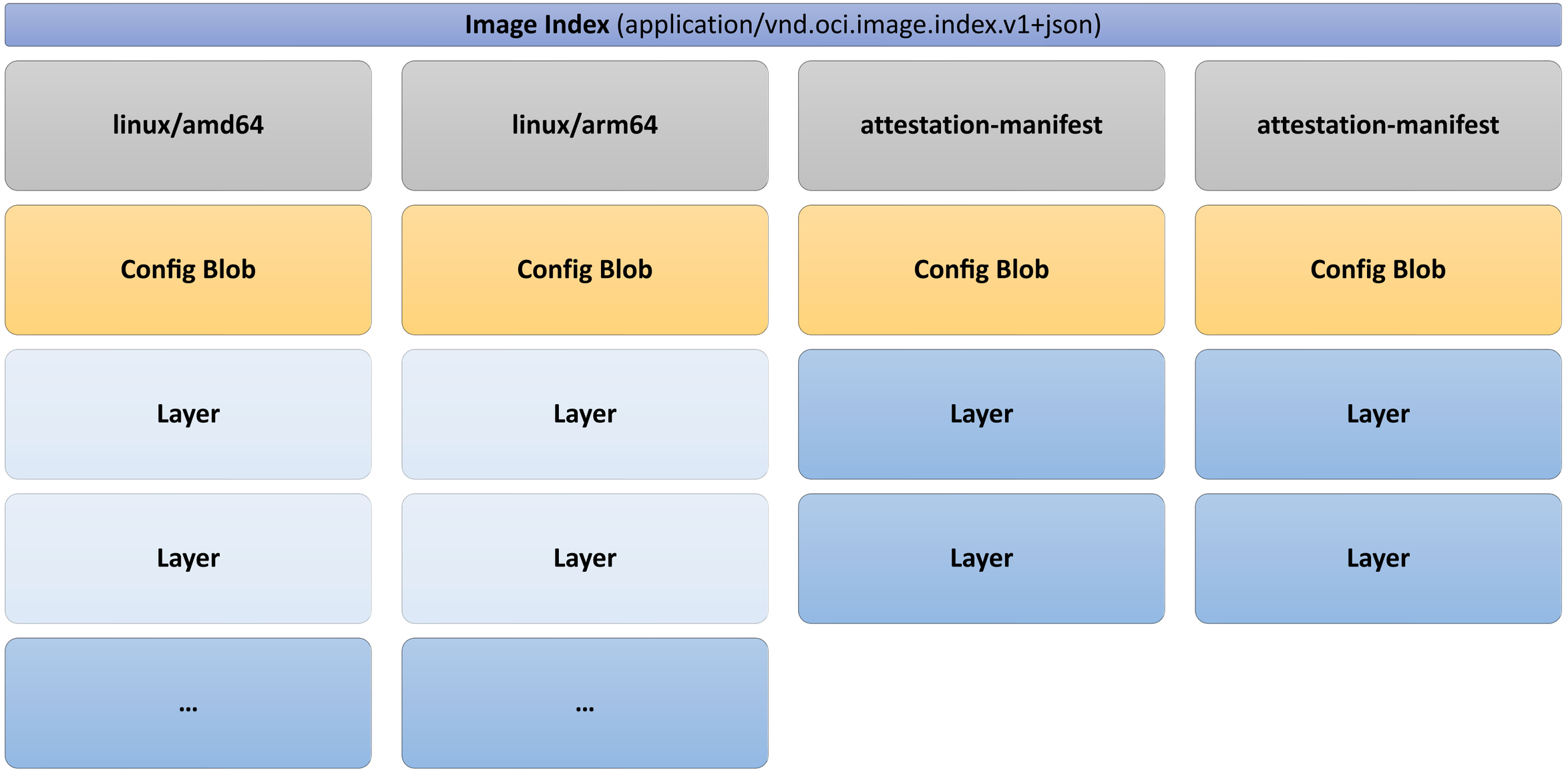
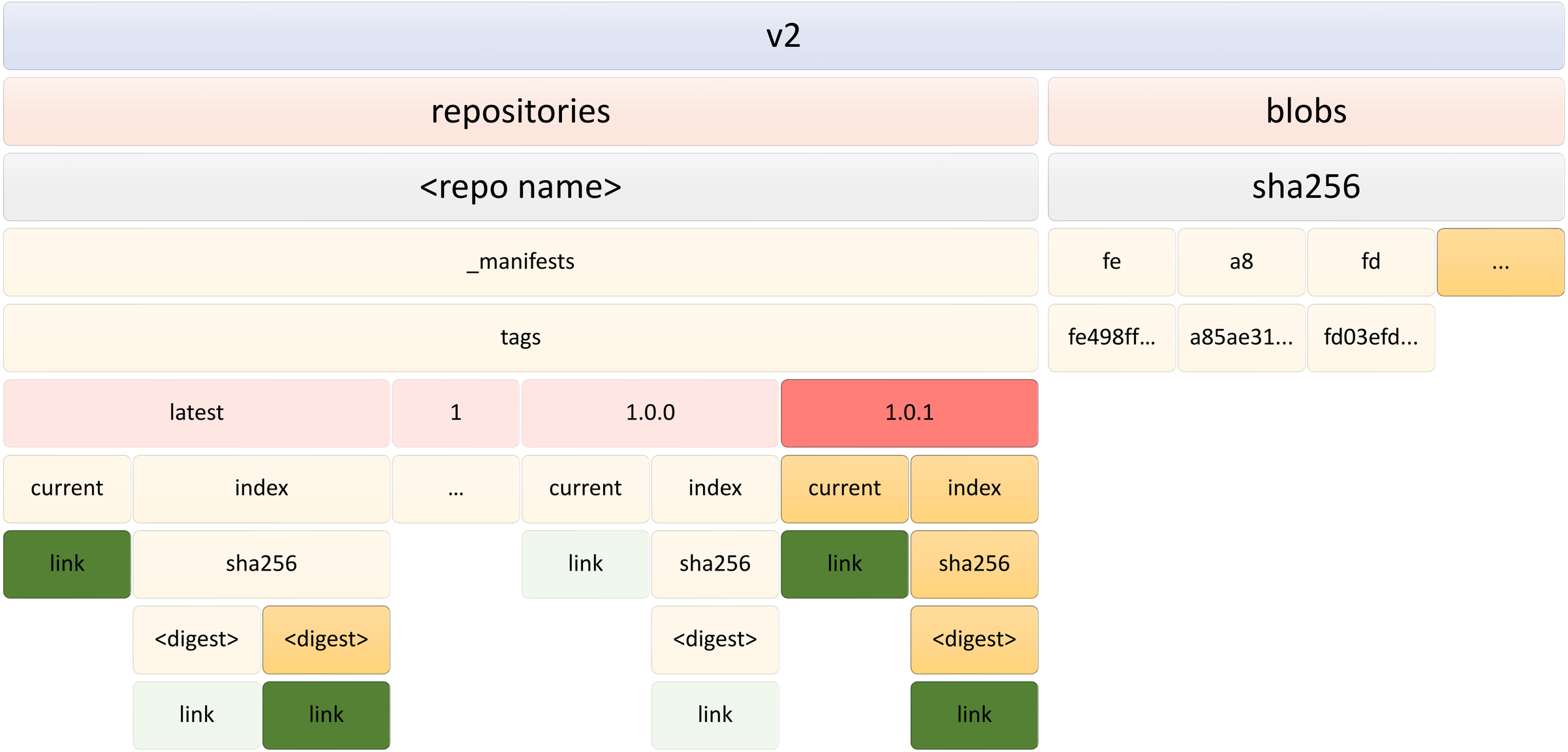
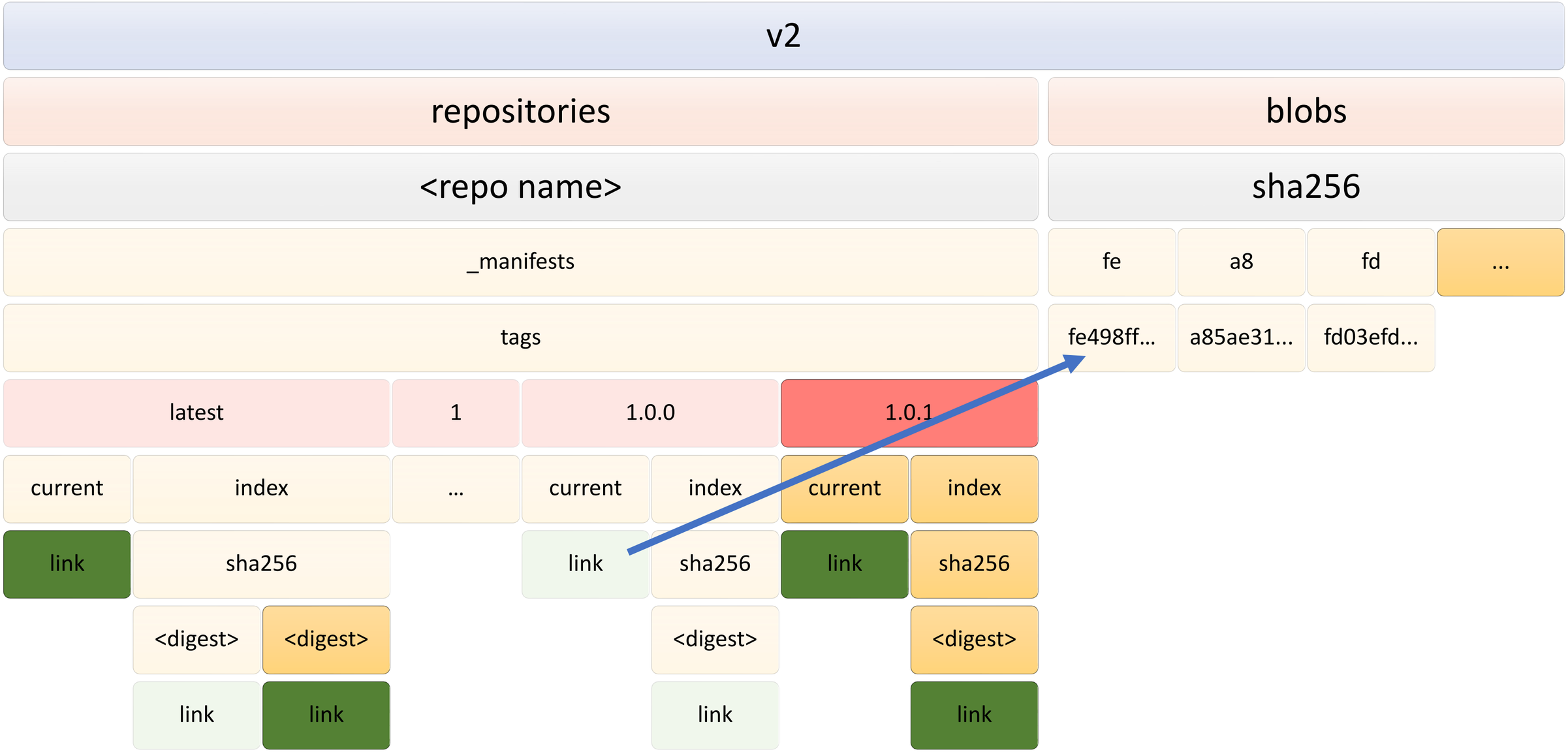


Image Index (application/vnd.oci.image.index.v1+json)
linux/amd64
linux/arm64
attestation-manifest
attestation-manifest
Config Blob
Config Blob
Config Blob
Config Blob
Layer
Layer
Layer
Layer
Layer
Layer
Layer
Layer
...
...

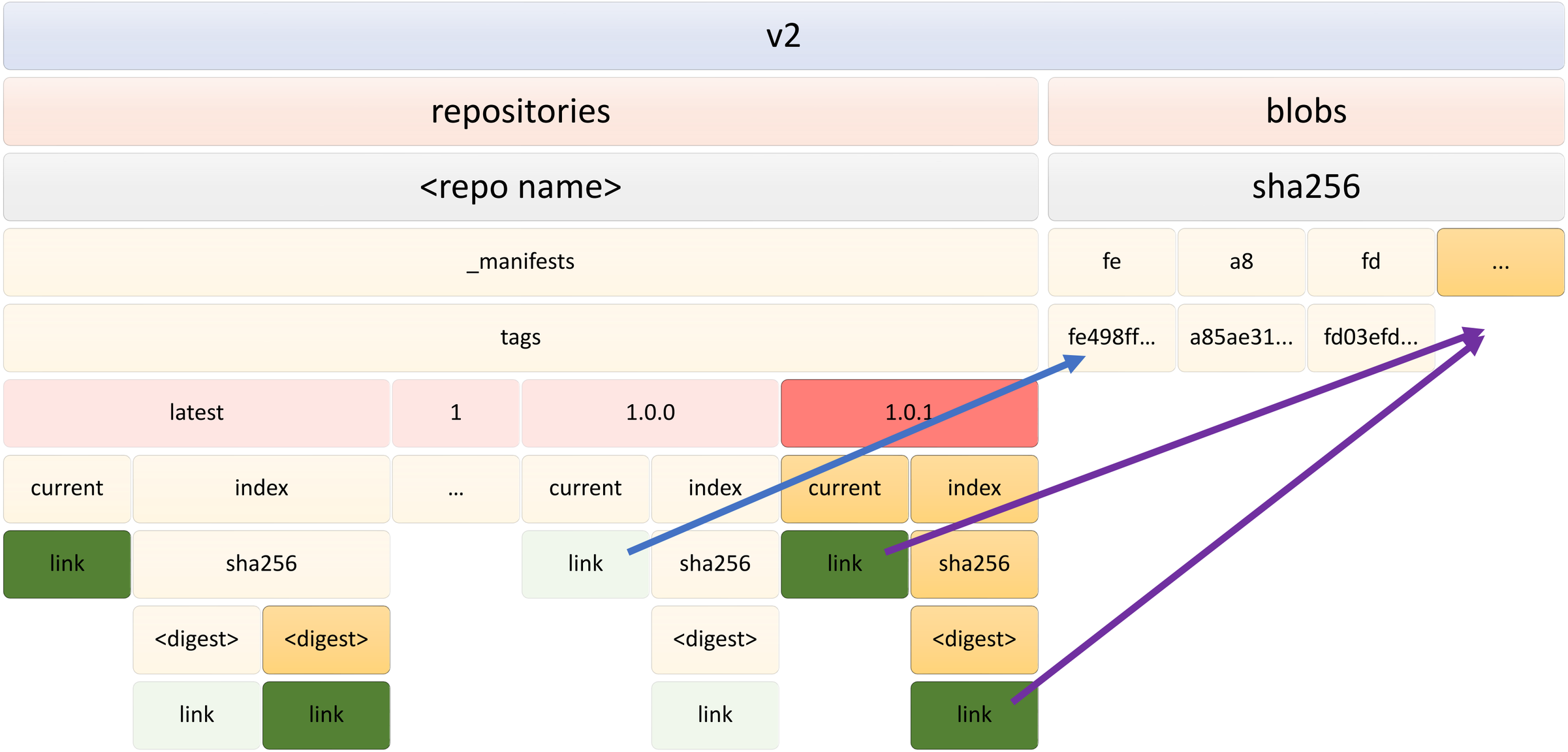
Registry View



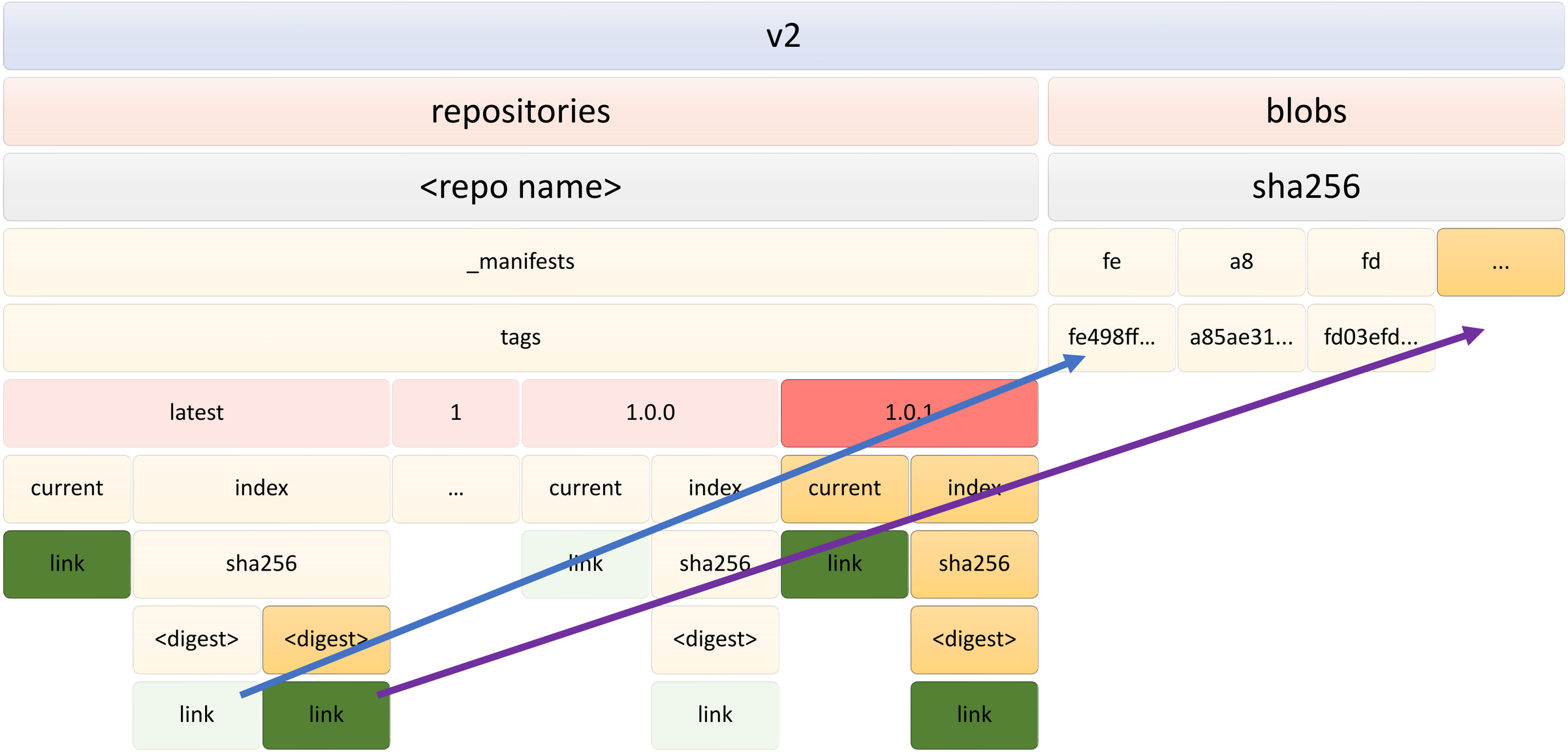
Registry View



Registry View

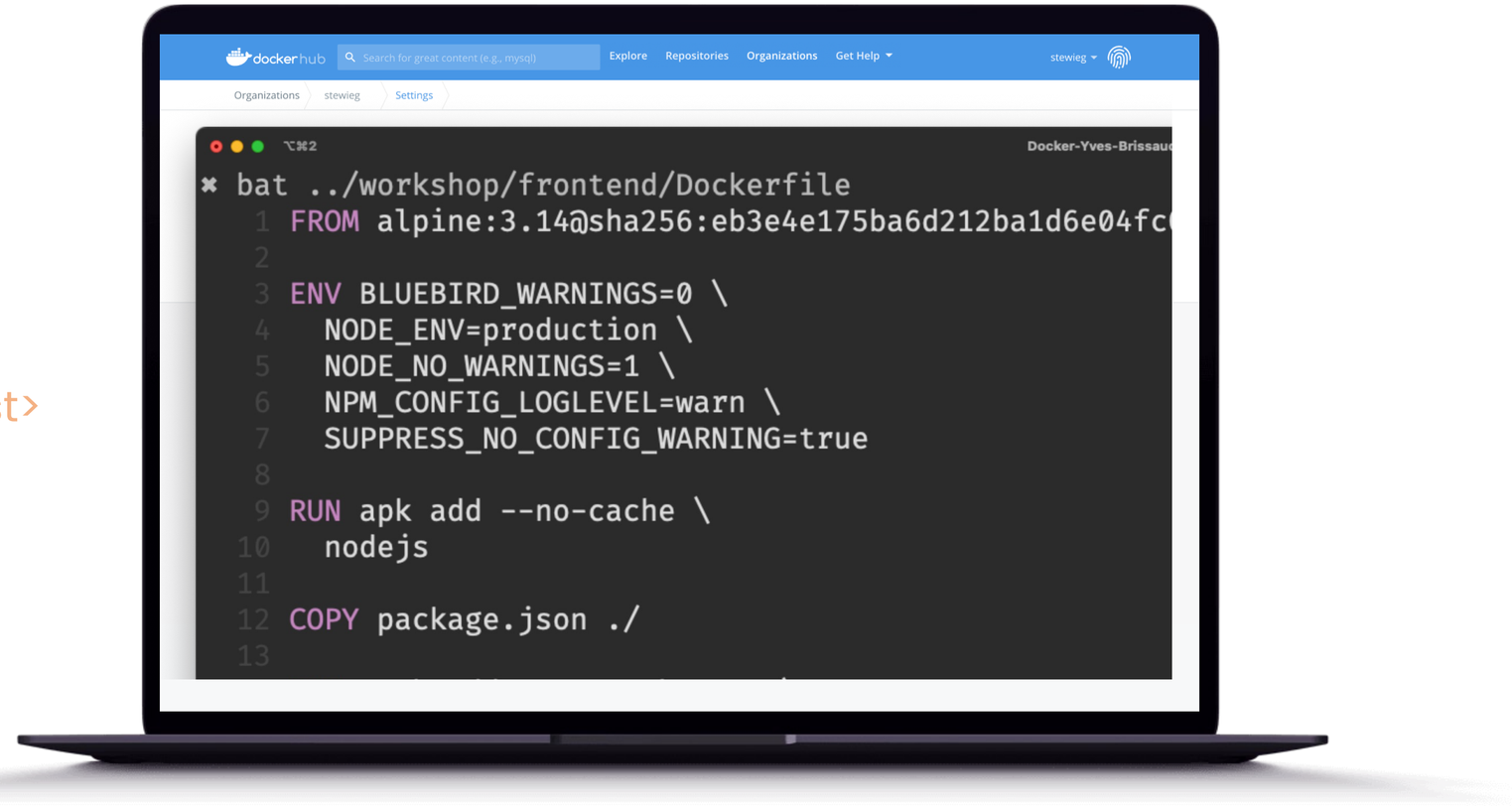


Registry View



Pin Image

FROM <repository>:tag@sha256:<digest>



06

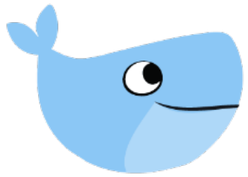
Beyond Images

https://www.pexels.com/photo/selective-focus-photography-of-assorted-color-puzzle-pieces-269399/

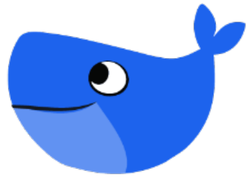
OCI Artifacts Everywhere!

Store other things than container image

Extend container images with related, non runnable, data



Store… everything



Docker Compose as OCI Image Manifest

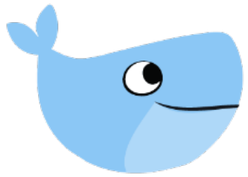

```
{
  "schemaVersion": 2,
  "mediaType": "application/vnd.oci.image.manifest.v1+json",
  "artifactType": "application/vnd.docker.compose.project",
  "config": {
    "mediaType": "application/vnd.docker.compose.project",
    "digest": "sha256:44136fa355b3678a1146ad16f7e8649e94fb4fc21fe77e8310c060f61caaff8a",
    "size": 2,
    "annotations": {
      "com.docker.compose.version": "2.22.0"
    }
  },
  "layers": [{
      "mediaType": "application/vnd.docker.compose.file+yaml",
      "digest": "sha256:839ee3e27293c4f021ad49d8e71ec85bfc69706d1f06037b848a4f13564eeba8",
      "size": 343,
      "annotations": {
        "com.docker.compose": "2.22.0"
```

Homebrew as OCI Image Manifest


```
{
  "mediaType": "application/vnd.oci.image.manifest.v1+json",
  "digest": "sha256:205f7a66495737af32db3125a63fc229622d8917b65eaf2436e4093f18948dc7",
  "size": 1911,
  "platform": {
    "architecture": "amd64",
    "os": "darwin",
    "os.version": "macOS 12"
  },
  "annotations": {
    "org.opencontainers.image.ref.name": "2.12.1.monterey",
    "sh.brew.bottle.digest": "62534bceb8f7074827fa2146dd13603018aaf07c82e22cfef96571c8133ce8a1",
    "sh.brew.tab": "{\"homebrew_version\":\"3.4.11-152-
ga3fab02\",\"changed_files\":[],\"source_modified_time\":1653865426,\"compiler\":\"clang\",\"runtime_
dependencies\":[],\"arch\":\"x86_64\",\"built_on\":{\"os\":\"Macintosh\",\"os_version\":\"macOS
12\",\"cpu_family\":\"penryn\",\"xcode\":\"13.4\",\"clt\":\"13.4.0.0.1.1651278267\",\"preferred_perl\
":\"5.30\"}}"
  }
}
```

CNAB Bundle


```
{
  "schemaVersion": 2,
  "manifests": [
    {
      "mediaType": "application/vnd.oci.image.manifest.v1+json",
      "digest": "sha256:464e2efbee1cfa84d29b3305f0901c75dc70f2fa554cbcb7a342e21cf7d7f5e1",
      "size": 188,
      "annotations": {
        "io.cnab.manifest.type": "config"
      }
    },
    {
      "mediaType": "application/vnd.docker.distribution.manifest.list.v2+json",
      "digest": "sha256:28ef97b8686a0b5399129e9b763d5b7e5ff03576aa5580d6f4182a49c5fe1913",
      "size": 2364,
      "annotations": {
        "io.cnab.manifest.type": "invocation"
      }
    }
  ],
  "annotations": {
    "io.cnab.runtime_version": "v1.0.0-WD",
    "io.docker.app.format": "cnab",
    "io.docker.type": "app",
    "org.opencontainers.artifactType": "application/vnd.cnab.manifest.v1"
```



And more...

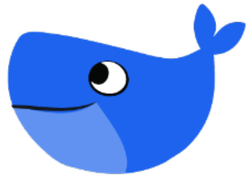
Helm Charts

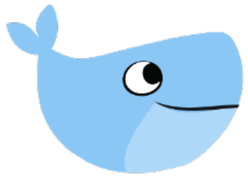
Wasm Modules

Docker Volumes

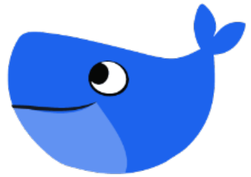
Dev Containers

...





Extend Images



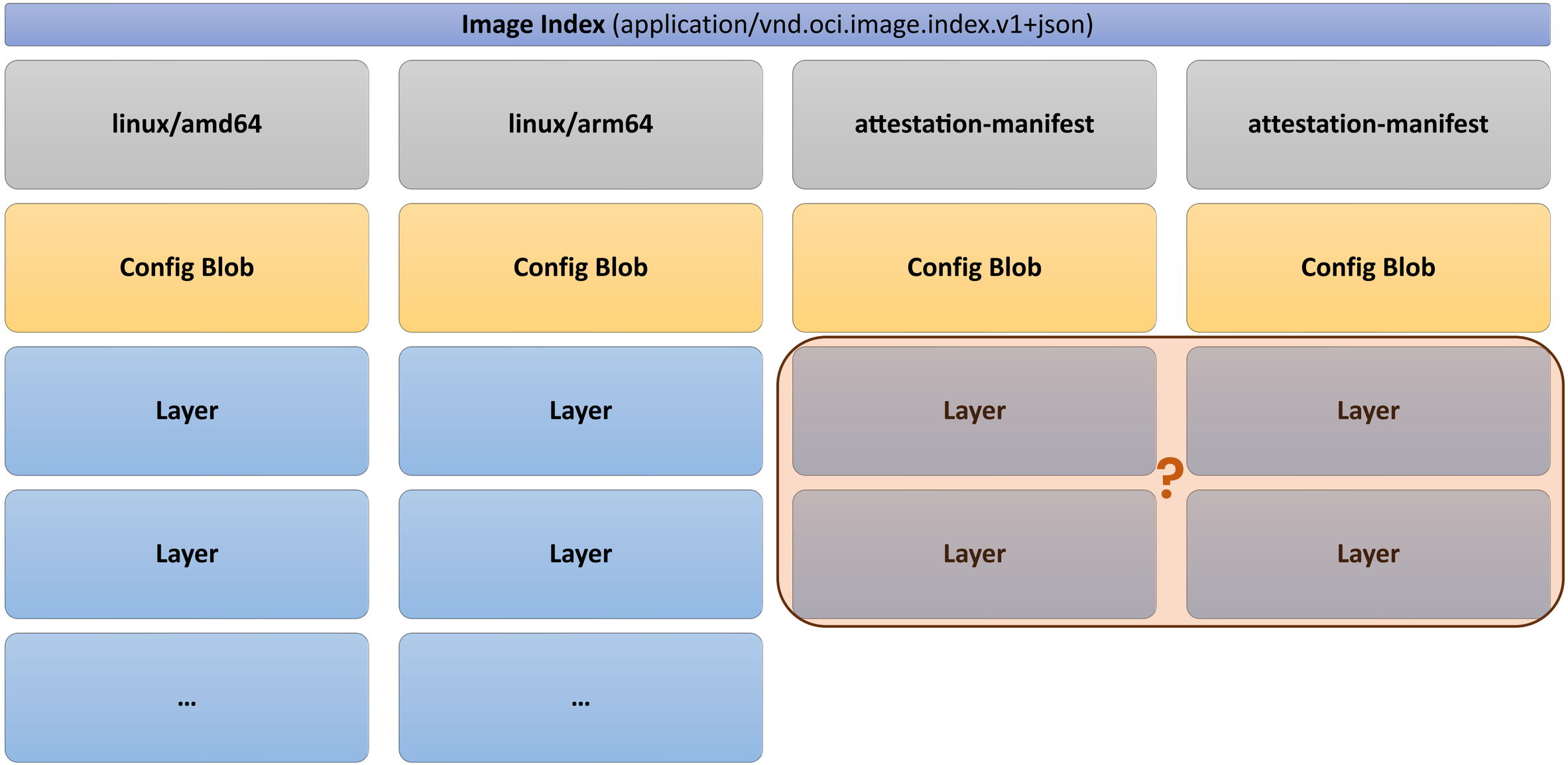
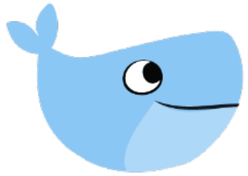


Image Index (application/vnd.oci.image.index.v1+json)
linux/amd64
linux/arm64
attestation-manifest
attestation-manifest
Config Blob
Config Blob
Config Blob
Config Blob
Layer
Layer
Layer
Layer
Layer
Layer
Layer
Layer
...
...
?

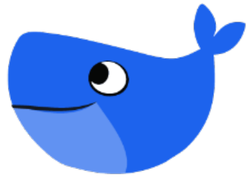


What else should we store?

Inline documentation?

Runbooks?

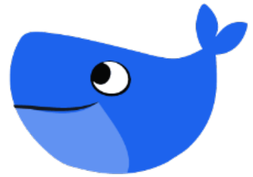
?



OCI Image and Distribution Specs v1.1

- How to create and store alternative (even non container) artifacts
- Manifest field for establishing relationships
- Query relationships

https://opencontainers.org/posts/blog/2023-07-07-summary-of-upcoming-changes-in-oci-image-and-distribution-specs-v-1-1/



THANK YOU


dockercon 23