



# From docker push To Bytes on Disk

## Inside Distribution

Wayne Warren

Adam Wolfe Gordon

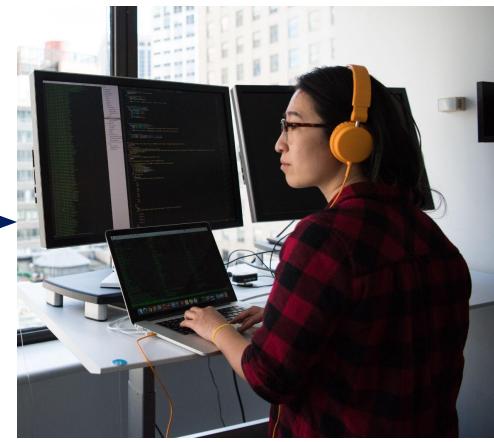
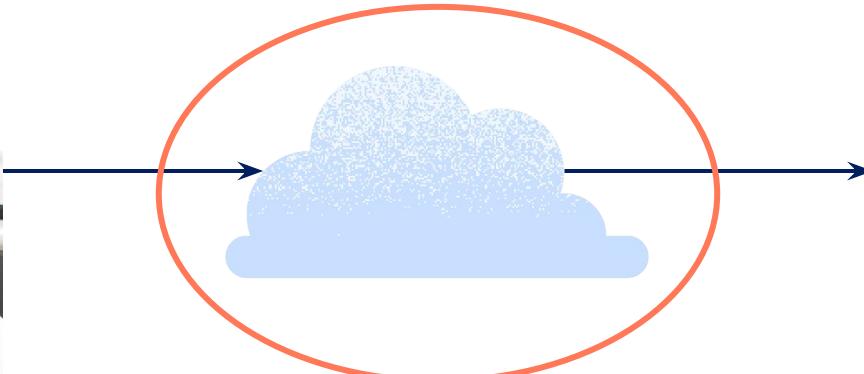
Sr. Engineer, DigitalOcean

Sr. Engineer, DigitalOcean

# What is a Container Registry?

○ ○ ○

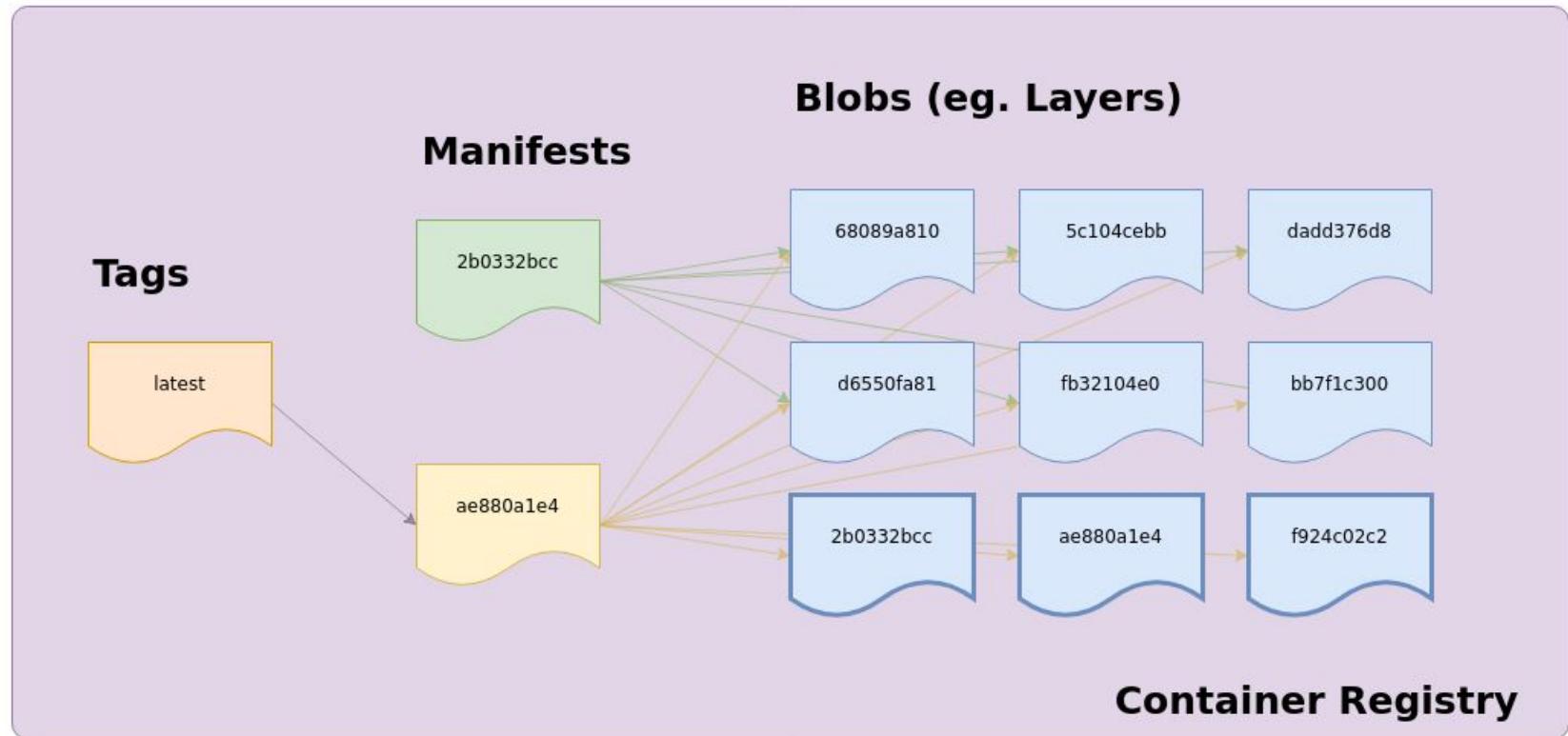
```
$ docker push registry.example.com/my-repository:latest
The push refers to repository [registry.example.com/my-repository]
86c24cefd9ff: Pushed
latest: digest: sha256:6634c44a5e60f5d3f0934922deb1cbded22a4aead6fbbaee4ea1b8c6981459233 size: 528
```



○ ○ ○

```
[~]% docker pull registry.example.com/my-repository:latest
latest: Pulling from my-repository
9c4930dab762: Pull complete
Digest: sha256:6634c44a5e60f5d3f0934922deb1cbded22a4aead6fbbaee4ea1b8c6981459233
Status: Downloaded newer image for registry.example.com/my-repository:latest
registry.example.com/my-repository:latest
```

# Content Addressable Object Store



# CNCF Distribution

[github.com/distribution/distribution](https://github.com/distribution/distribution)



Container registry  
implementation in Go



Previously called  
docker/distribution



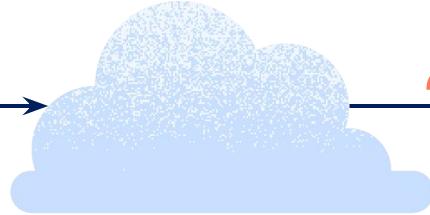
Donated to CNCF by  
Docker in 2020

○ ○ ○

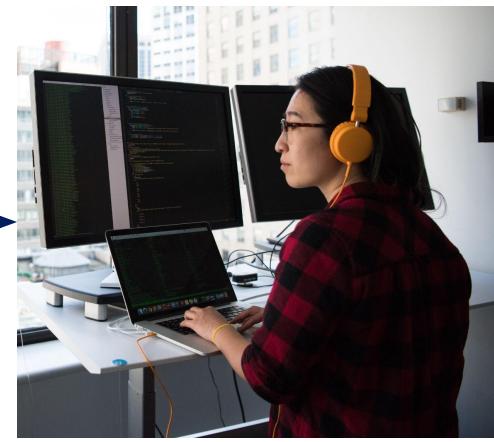
```
$ docker push registry.example.com/my-repository:latest
The push refers to repository [registry.example.com/my-repository]
86c24cefd9ff: Pushed
latest: digest: sha256:6634c44a5e60f5d3f0934922deb1cbded22a4aead6fbbaee4ea1b8c6981459233 size: 528
```



???



???



○ ○ ○

```
[~]% docker pull registry.example.com/my-repository:latest
latest: Pulling from my-repository
9c4930dab762: Pull complete
Digest: sha256:6634c44a5e60f5d3f0934922deb1cbded22a4aead6fbbaee4ea1b8c6981459233
Status: Downloaded newer image for registry.example.com/my-repository:latest
registry.example.com/my-repository:latest
```

# OCI Distribution Specification

[github.com/opencontainers/distribution-spec](https://github.com/opencontainers/distribution-spec)



Standardized HTTP API for container registries



Reached v1 in 2021



Started out as the “Docker Registry HTTP API v2”

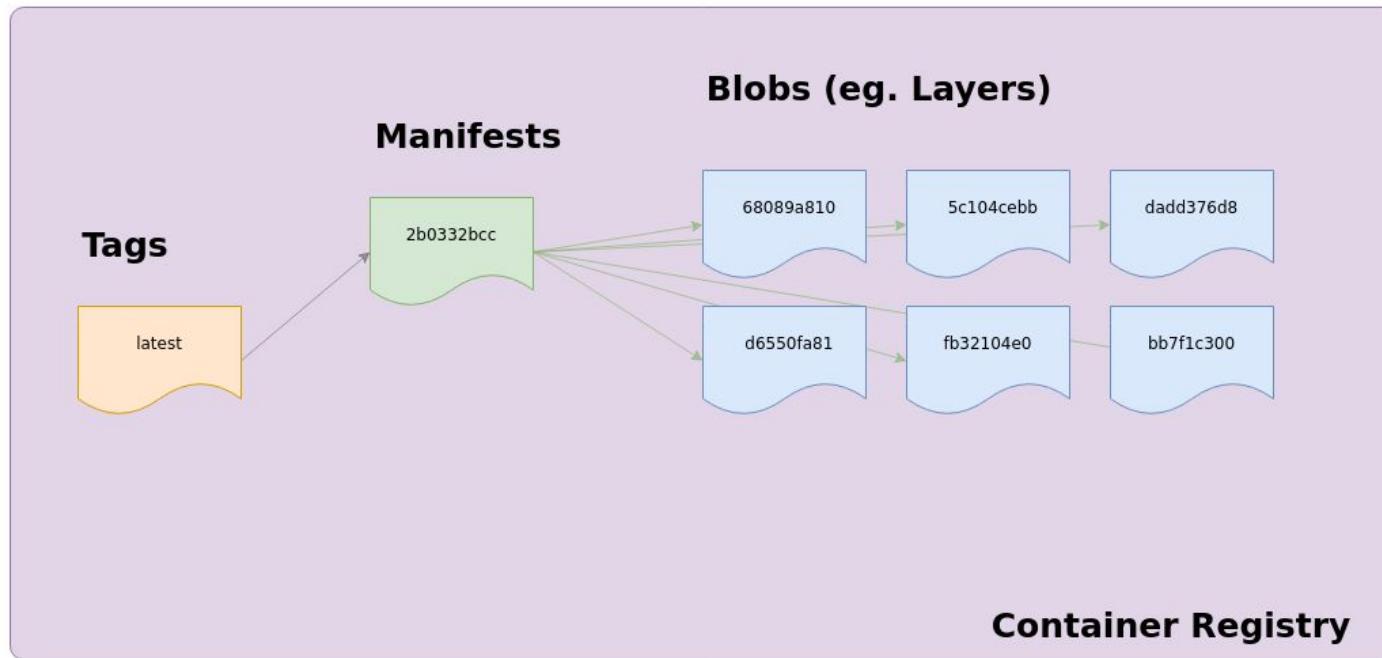
# Coming Up Next...

1. The Distribution Spec
  - a. Pushing an image
  - b. Pulling an image
2. The Distribution Codebase
  - a. Overview
  - b. Pushing an image
3. Garbage Collection

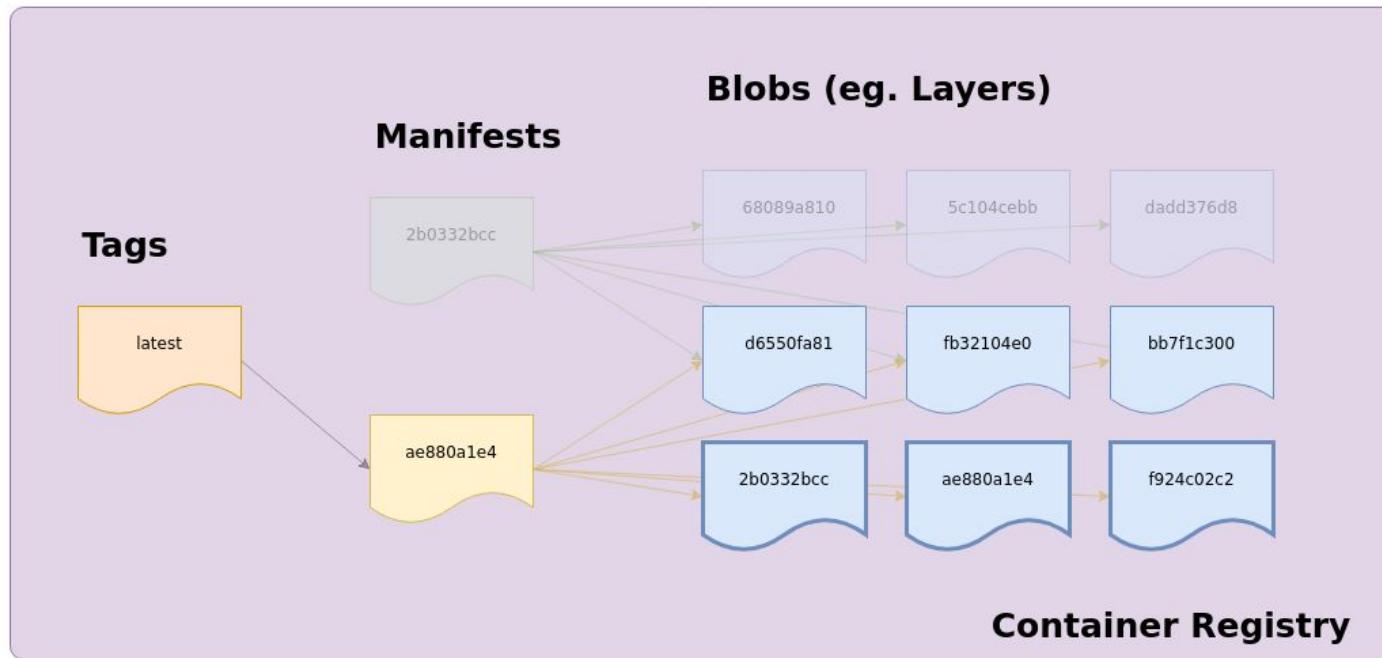


# Inside docker push

```
docker push  
registry.example.com/my-repository:latest
```



# `docker push registry.example.com/my-repository:latest`



# Finding Blobs

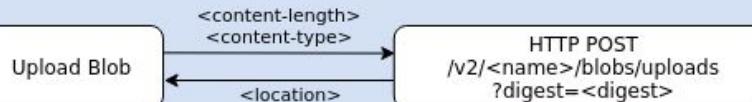
**HEAD**

/v2/<repository>/blobs/<**digest**>

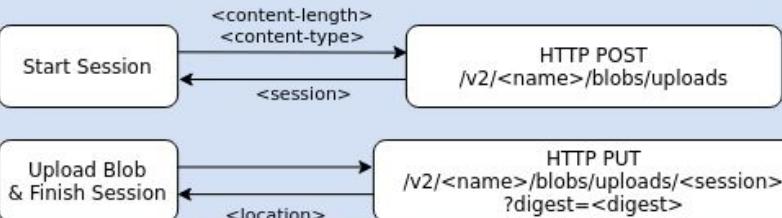


# Three Ways to Push a Blob

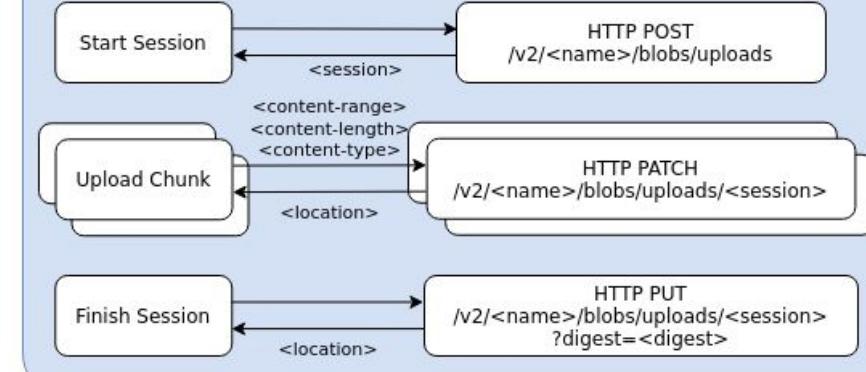
## Monolithic Blob Upload (deprecated)



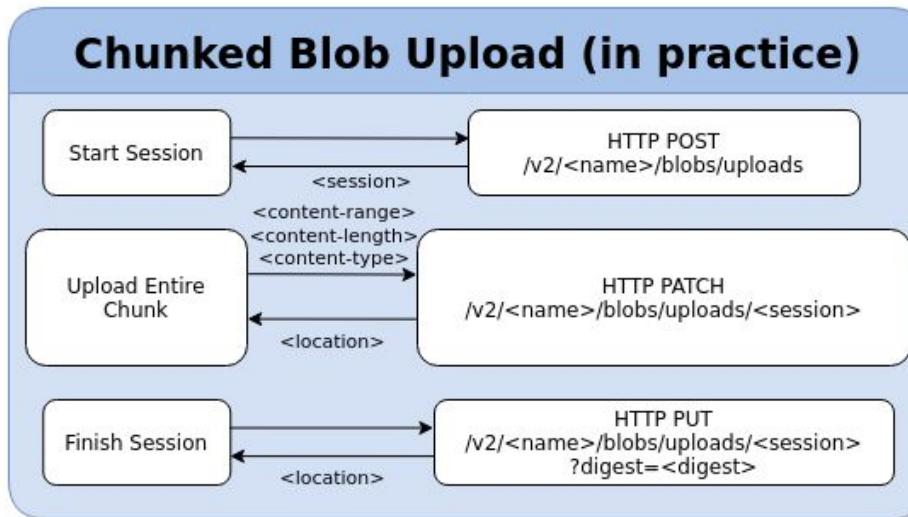
## Monolithic Blob Upload



## Chunked Blob Upload



# Pushing Blobs in Practice



# Blob “Mounting”

**POST**

```
/v2/<repository>/blobs/uploads/  
?mount=<digest>  
[&from=<other repository>]
```



# Pushing Manifests (and Tags)

**PUT**

/v2/<repository>/manifests/<**reference**>

A **reference** can be a **digest** or a **tag**

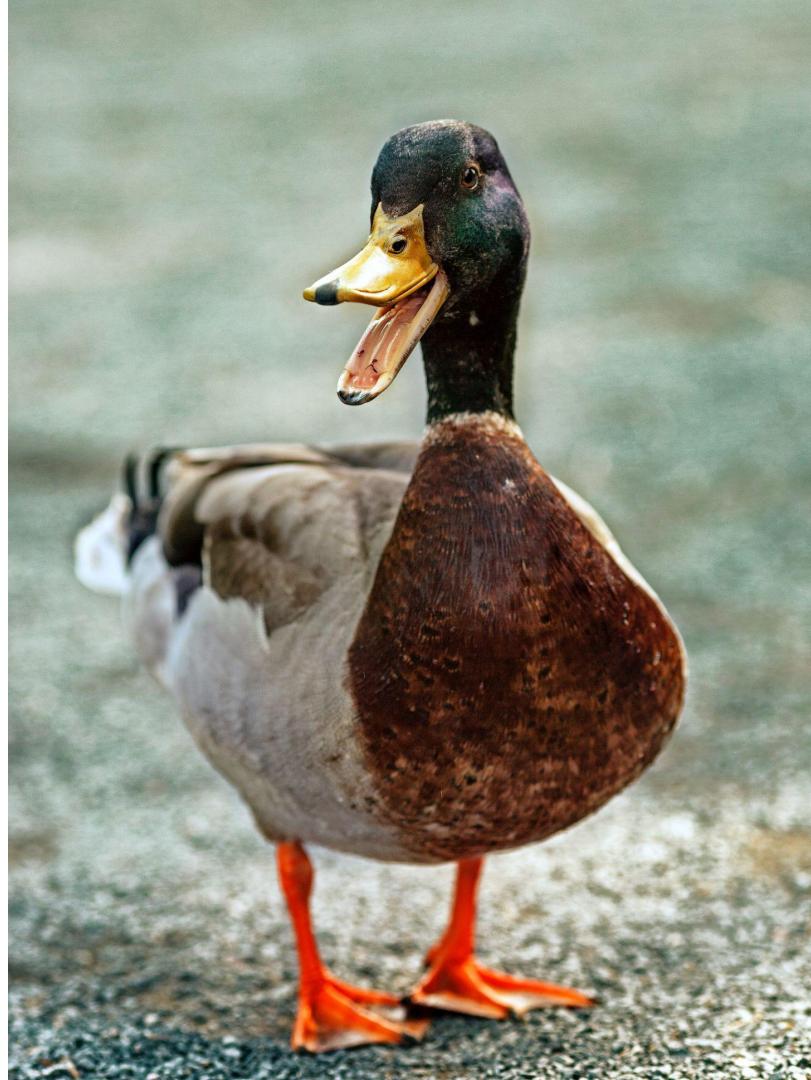
If it looks like a digest and quacks like a digest, it's a digest:

<**algorithm**>:<**encoded hash**>

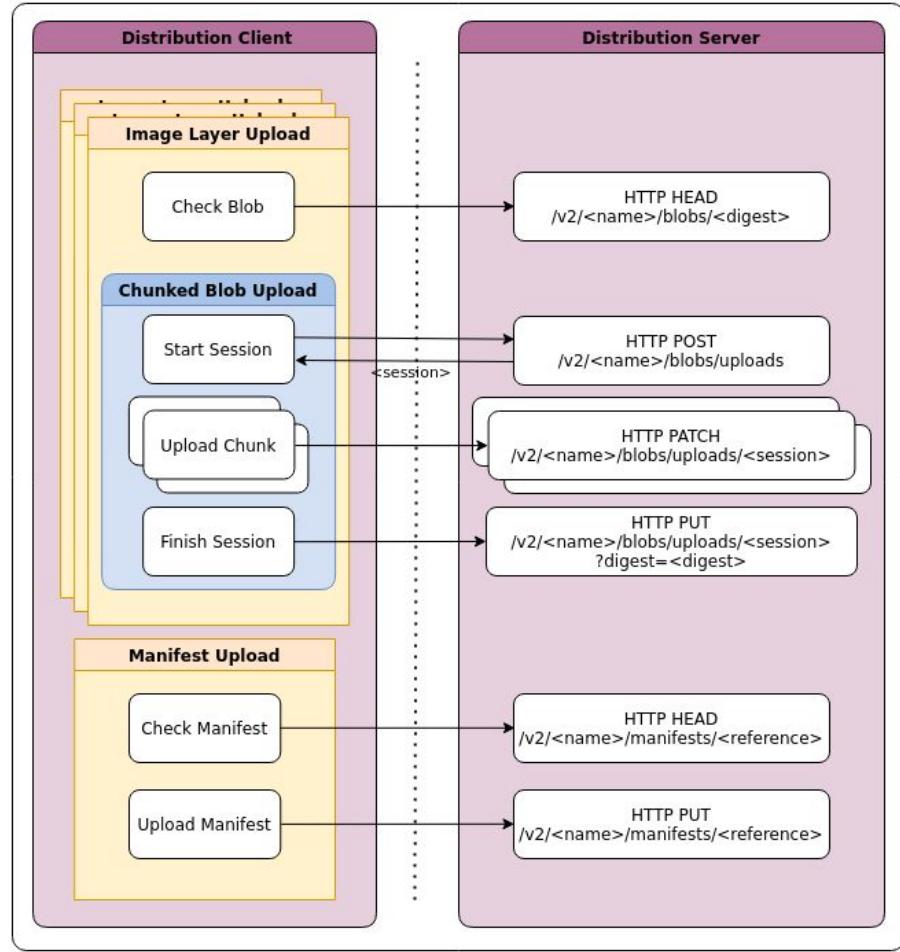
Otherwise it's validated as a tag:

[a-zA-Z0-9\_][a-zA-Z0-9.\_-]{0,127}

DigitalOcean

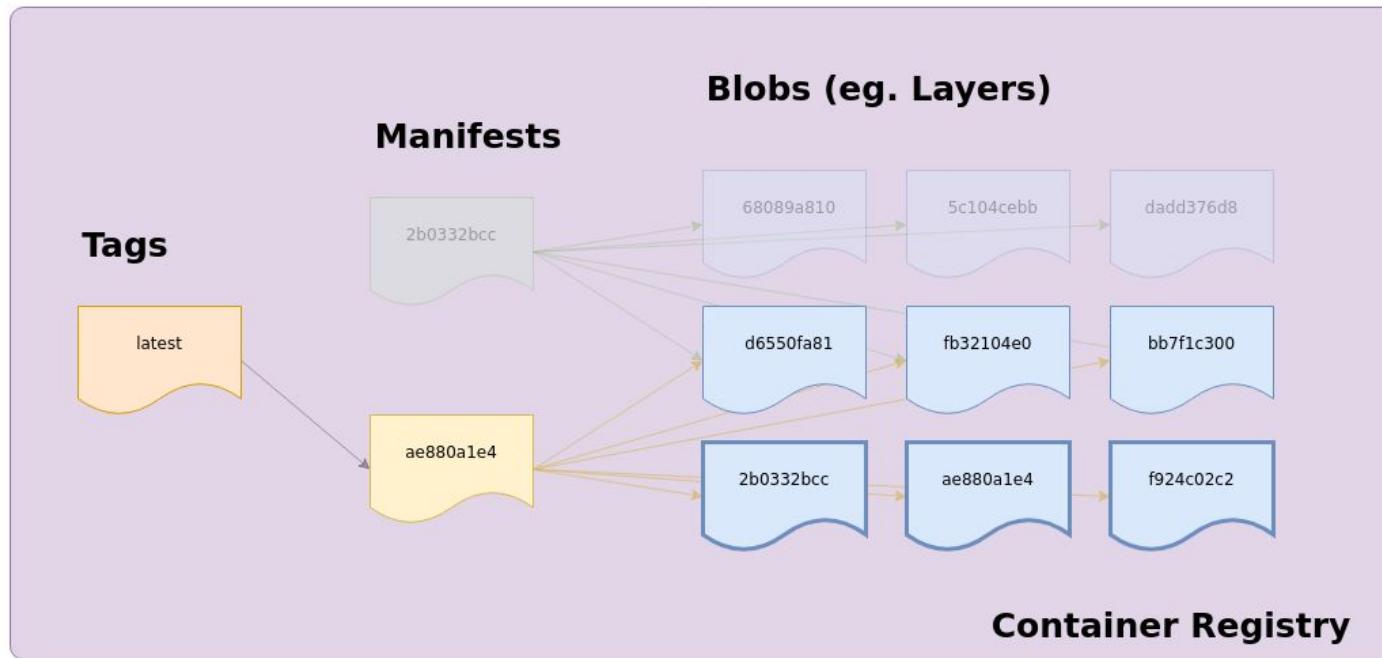


# Image Push End-to-End



# Inside docker pull

# `docker pull registry.example.com/my-repository:latest`

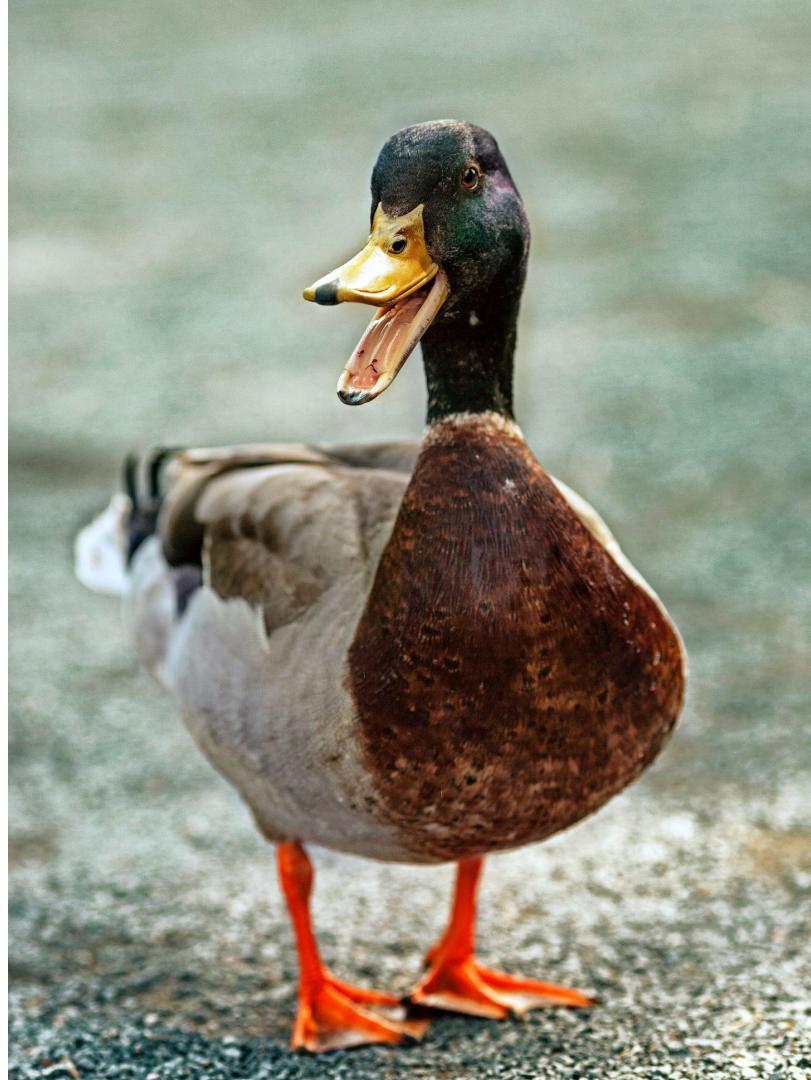


# Pulling Manifests

GET

/v2/<repository>/manifests/<reference>

A **reference** can be a **digest** or a **tag**



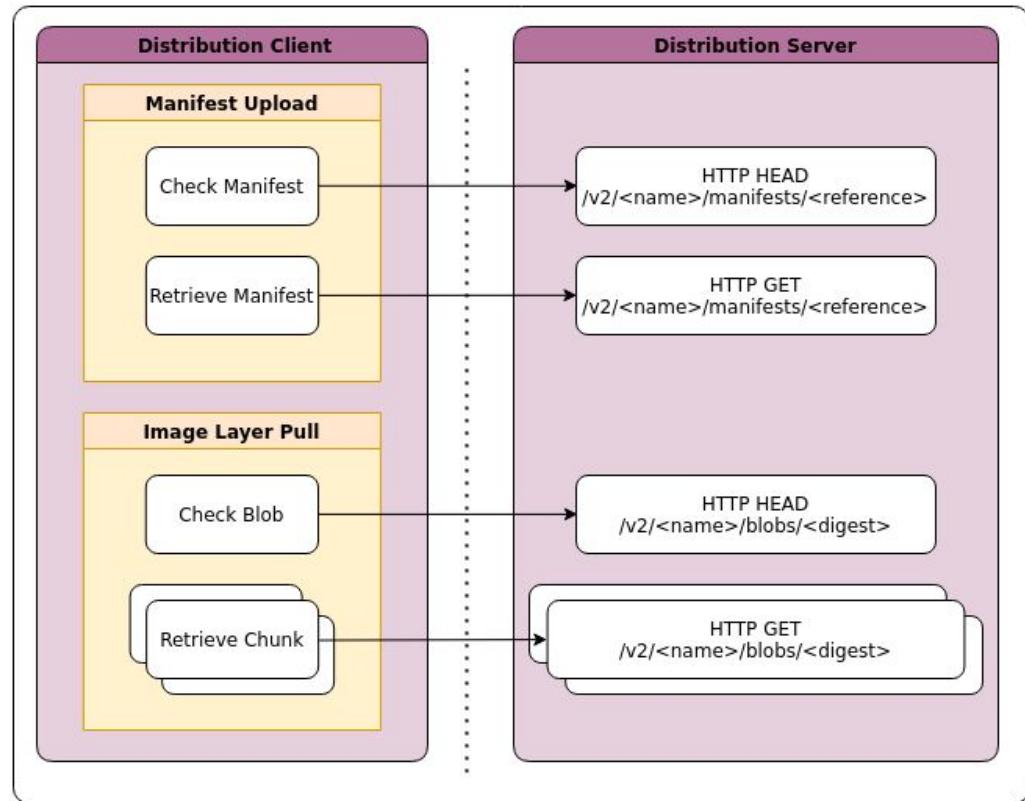
# Pulling Blobs

**GET**

/v2/<repository>/blobs/<**digest**>



# Image Pull End-to-End



# Distribution Internals

## Distribution Layers

HTTP API defined by the distribution spec.

Various OCI interfaces that abstract over operations on blobs, manifests, tags, descriptors, etc.

Filesystem-like interface that abstracts over backend storage drivers.

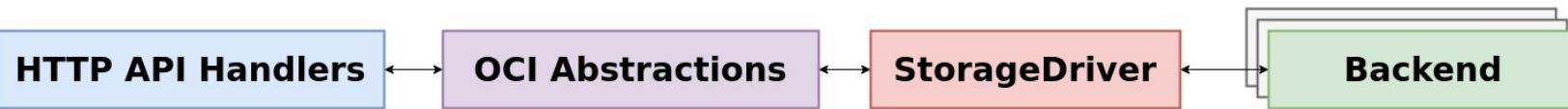
Backend storage implementations. In-memory, filesystem, S3, Azure, GCS, OSS.

**HTTP API Handlers**

**OCI Abstractions**

**StorageDriver**

**Backend**



# HTTP API Handlers

HTTP API	
GET	/v2/
GET / HEAD	/v2/<name>/blobs/<digest>
POST	/v2/<name>/blobs/uploads/
POST	/v2/<name>/blobs/uploads/?mount=<digest>&from=<other_name>
POST	/v2/<name>/blobs/uploads/?digest=<digest>
PATCH	/v2/<name>/blobs/uploads/<session-id>
PUT	/v2/<name>/blobs/uploads/<session-id>?digest=digest
DELETE	/v2/<name>/blobs/<digest>
PUT	/v2/<name>/manifests/<reference>
GET / HEAD	/v2/<name>/manifests/<reference>
DELETE	/v2/<name>/manifests/<reference>
GET	/v2/<name>/tags/list
GET	/v2/<name>/tags/list?n=<int>&last=<int>

blobHandler
GetBlob
DeleteBlob

blobUploadHandler
StartBlobUpload
GetUploadStatus
PatchBlobData
PutBlobUploadComplete
CancelBlobUpload
ResumeBlobUpload

manifestHandler
GetManifest
PutManifest
DeleteManifest

tagsHandler
GetTags

catalogHandler
GetTags

# Distribution Spec OCI Abstractions

## Types

- Manifests
- Blobs, eg:
  - Image Layers
  - Manifest Configs
- Tags
- Descriptors

## Operations

- UploadMonolithic
- UploadChunked
- Download
- CheckIfExists
- List
- Delete

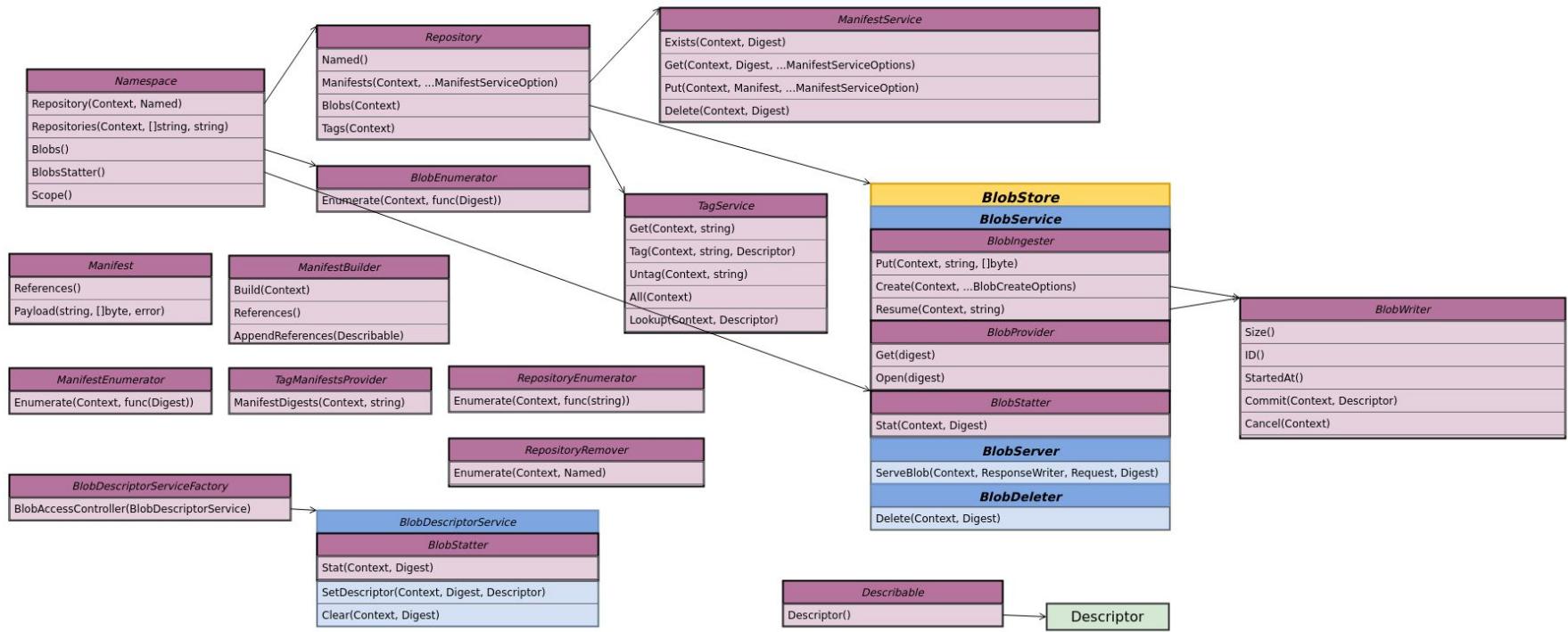
**Expectations**



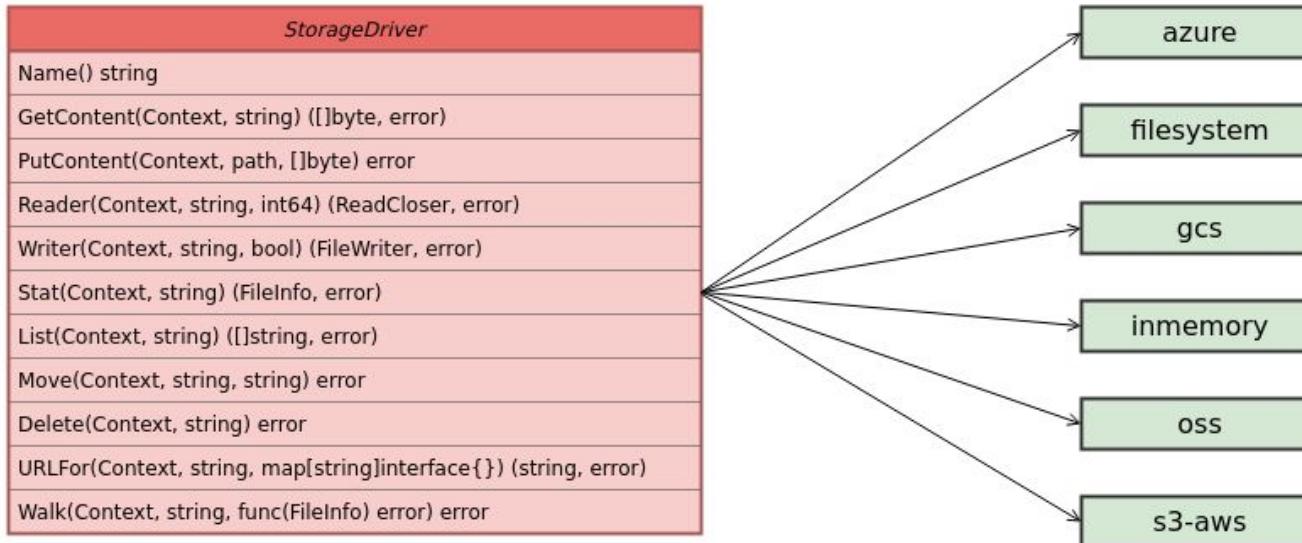
**Reality**



# OCI Abstractions

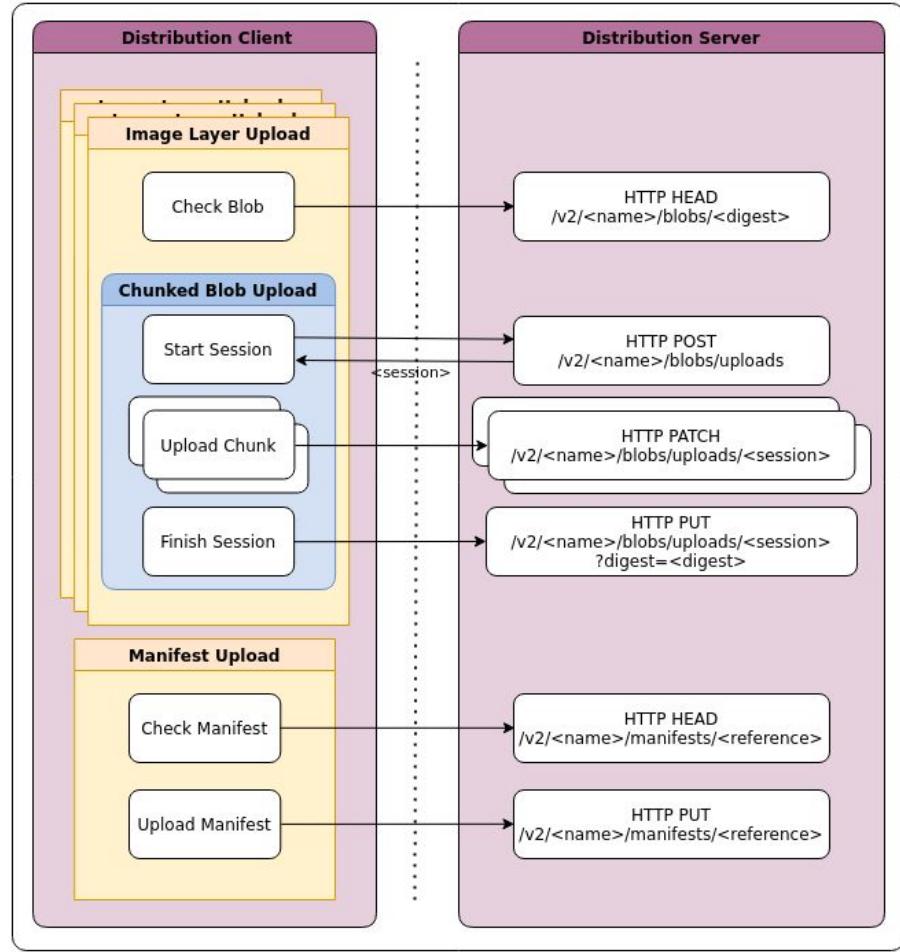


# StorageDriver



# From docker push to bytes on disk

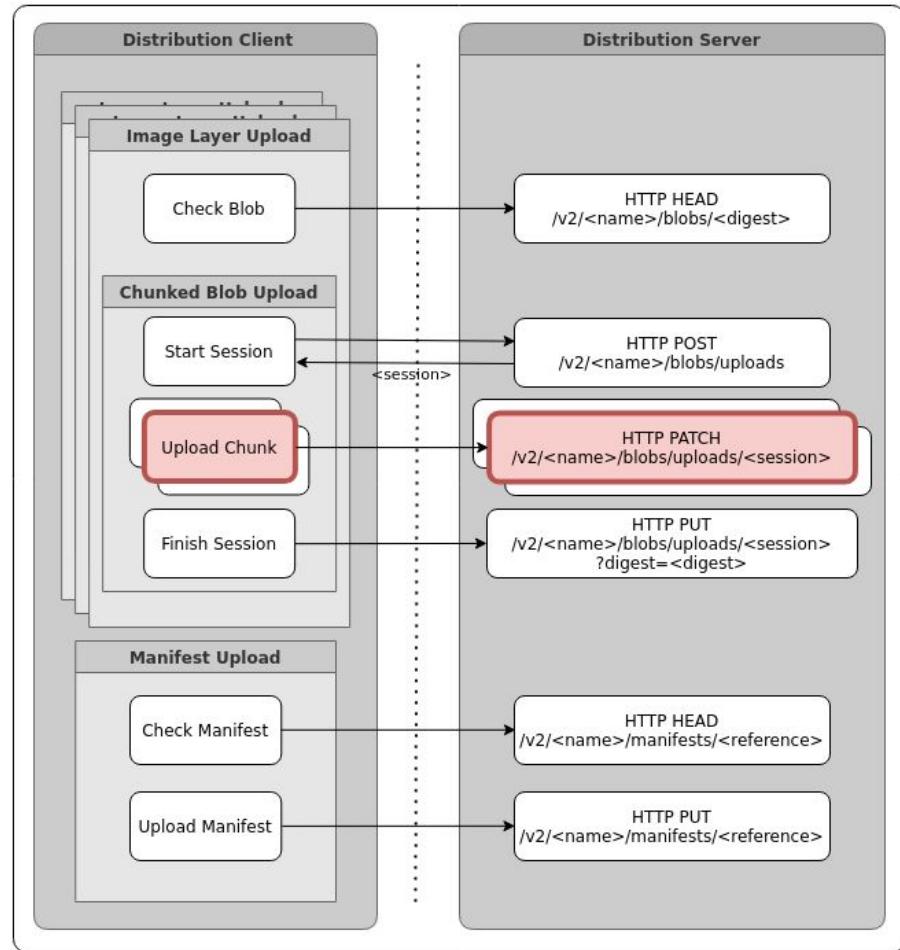
# Image Push End-to-End

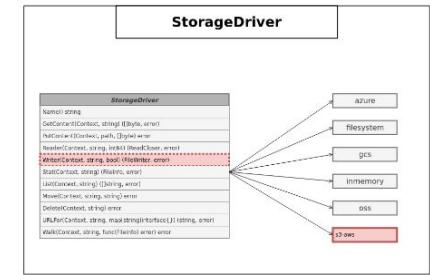
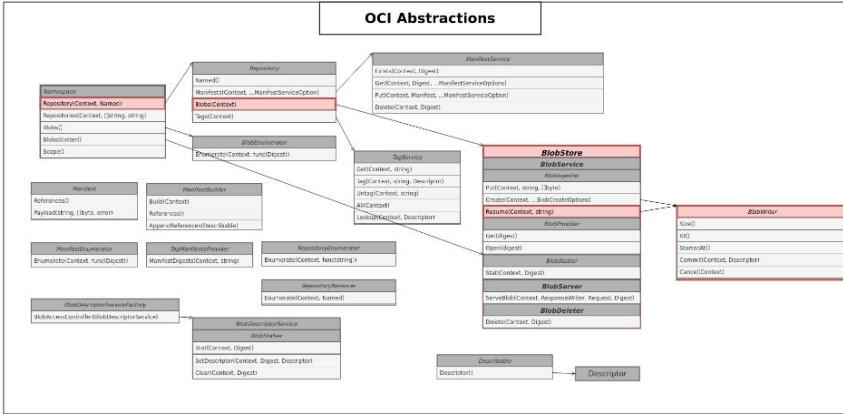
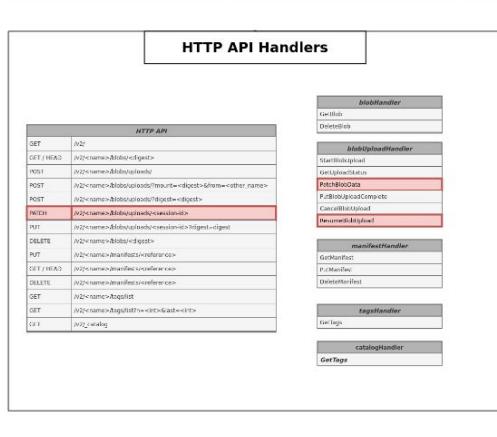


# HTTP PATCH

/v2/<n>/blobs/uploads/<s>

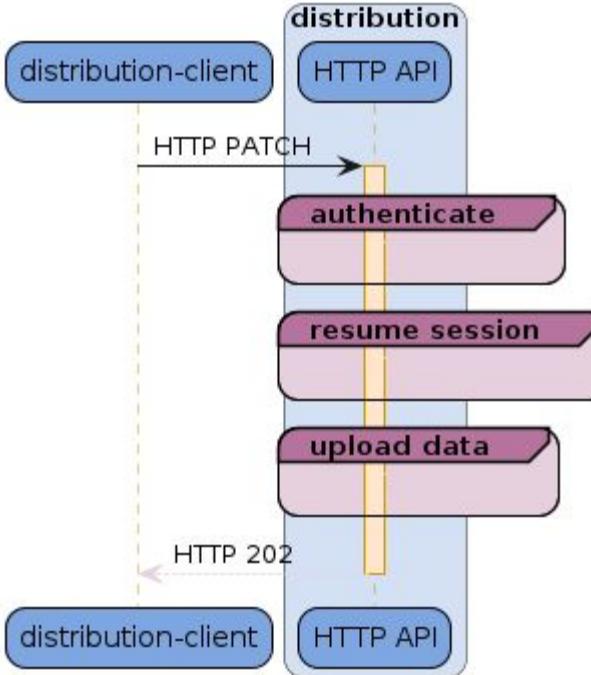
- <n> = name of image (called “repository” in distribution spec). eg:
  - “[nginx](#)” in `registry.digitalocean.com/my-registry/nginx:latest`
- <s> = UUID of blob upload session





# **blob PATCH**

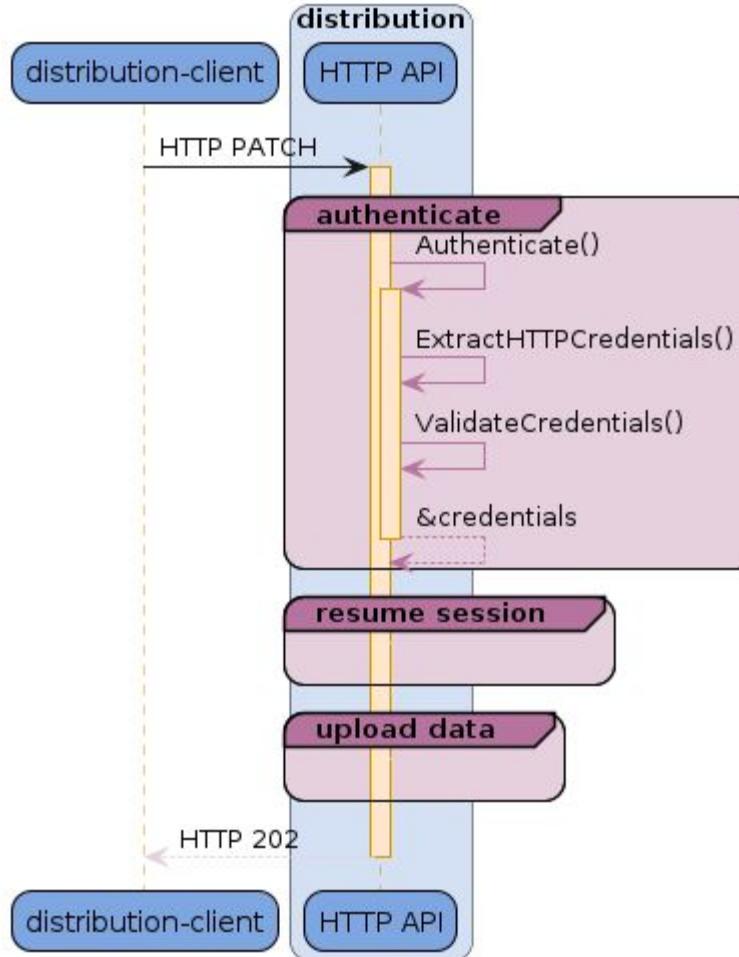
## Overview



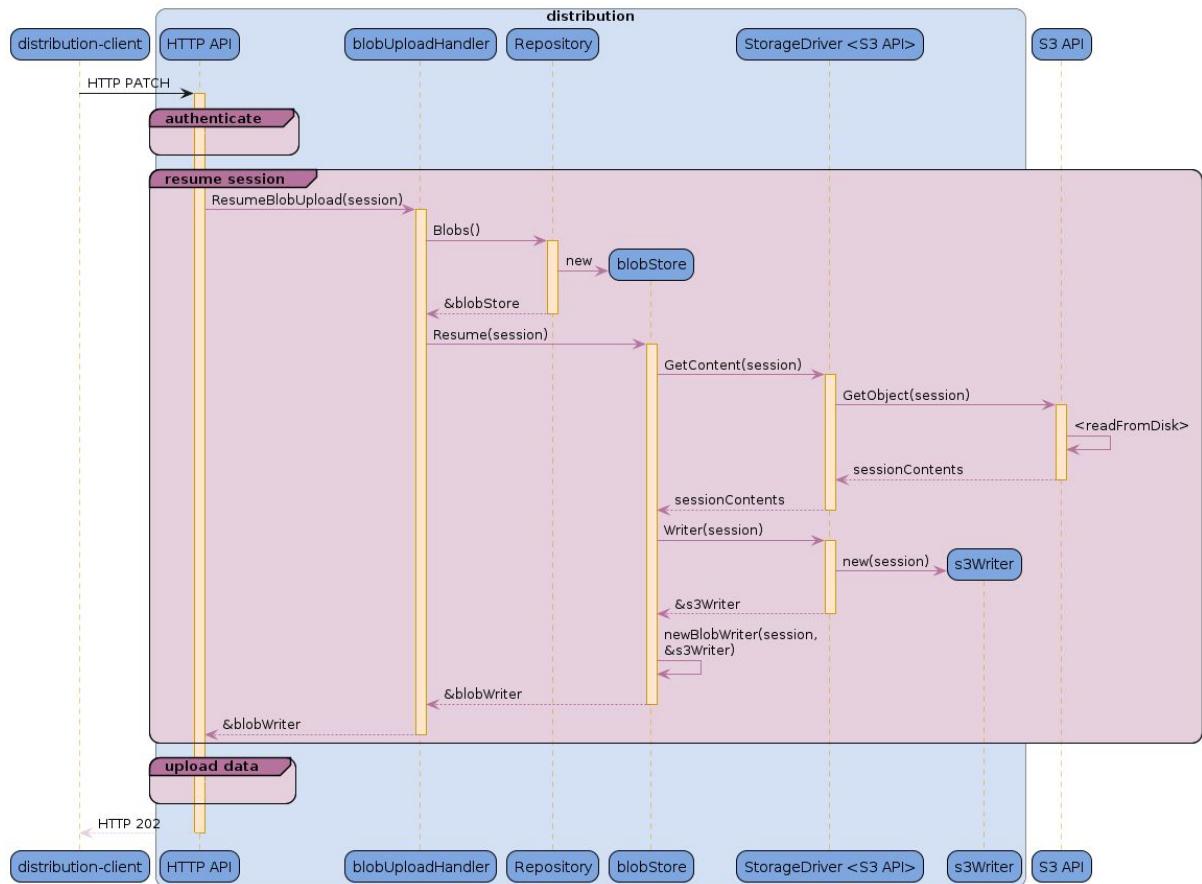
# Authentication

Supported methods:

- Basic Auth
- Bearer Token / JWT
- OAuth

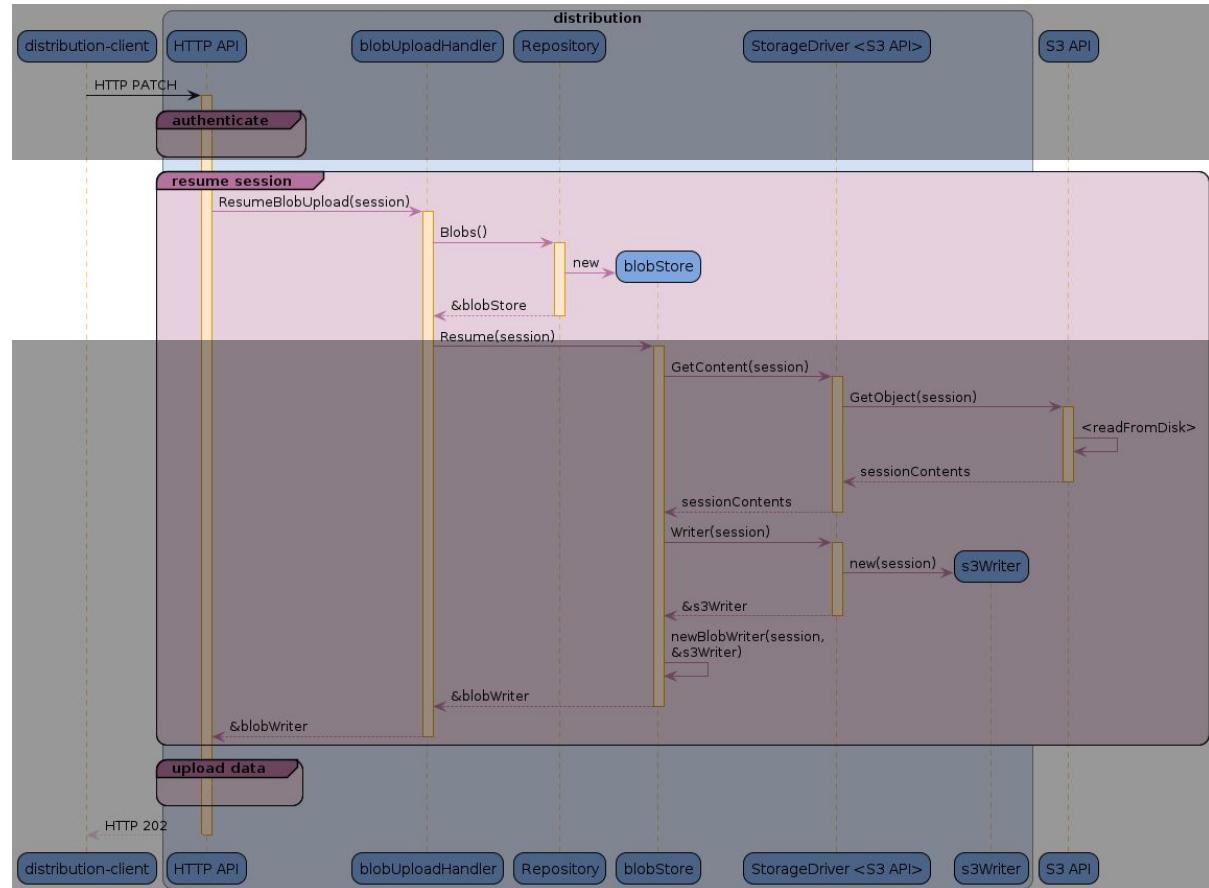


# Resume Session



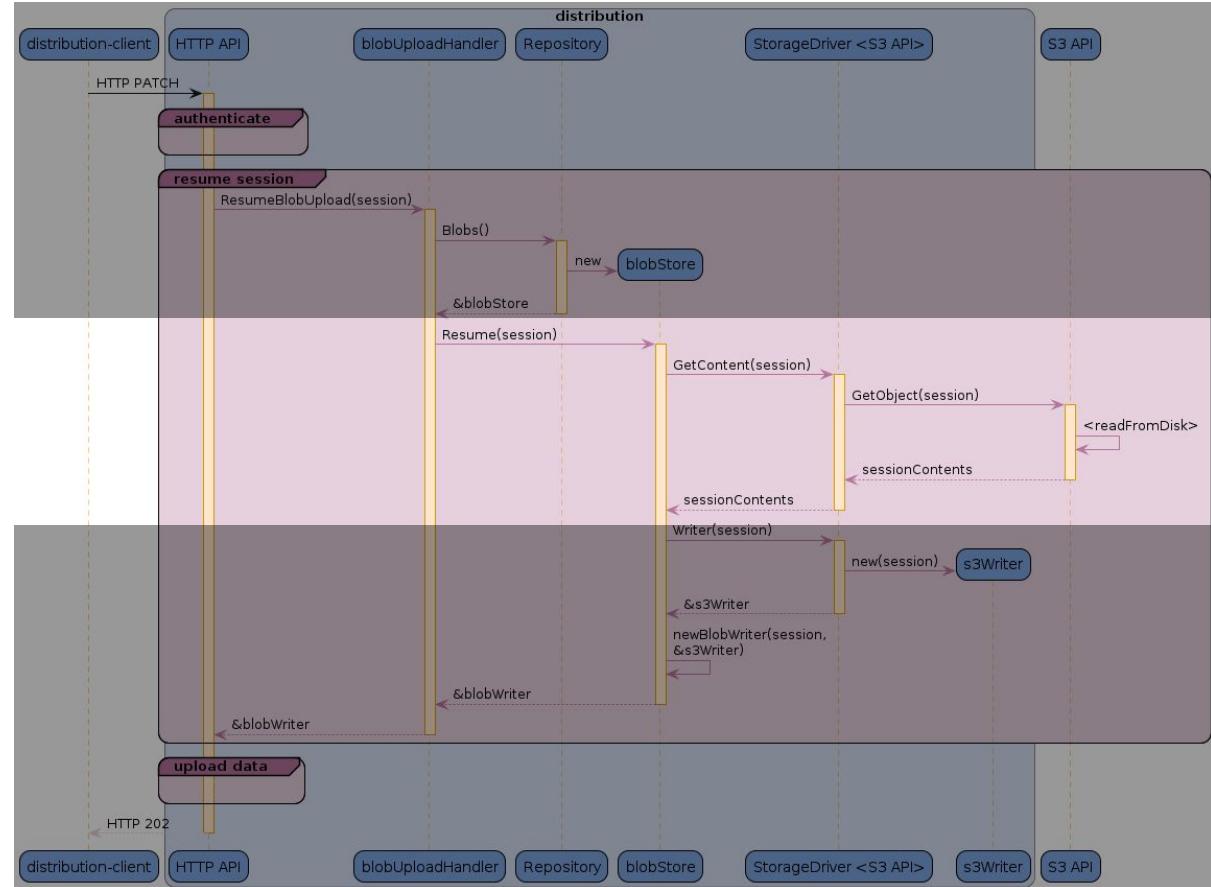
# Resume Session

- Repository (aka image name)
  - eg “**myreg**” in  
/v2/**myreg**/blobs/uploads/**SESSID**
- Distribution spec requires some kind of session identifier in URL path
  - eg “**SESSID**” in  
/v2/**myreg**/blobs/uploads/**SESSID**



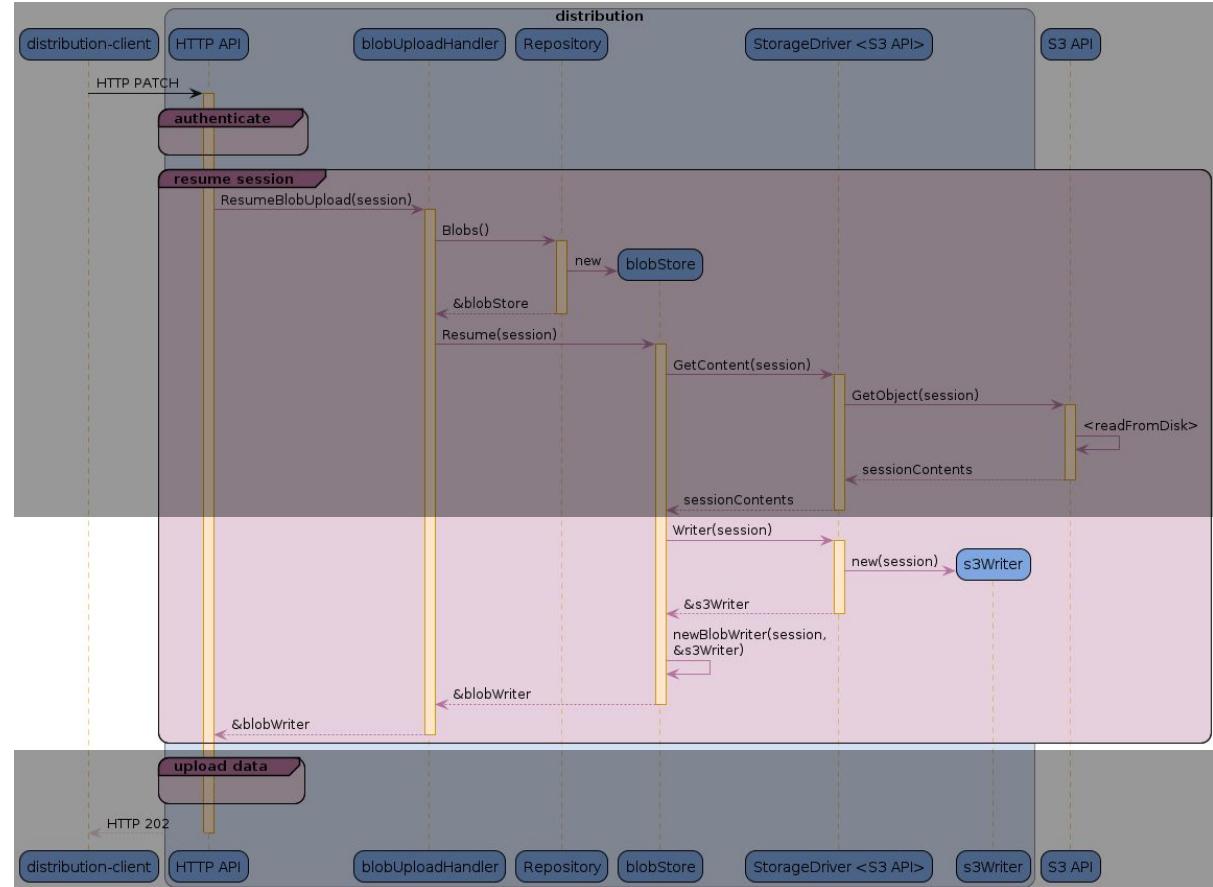
# Resume Session

- BlobStore.Resume()
  - StorageDriver.GetContent()
    - S3.GetObject()
  - Validates session info against request parameter(s)

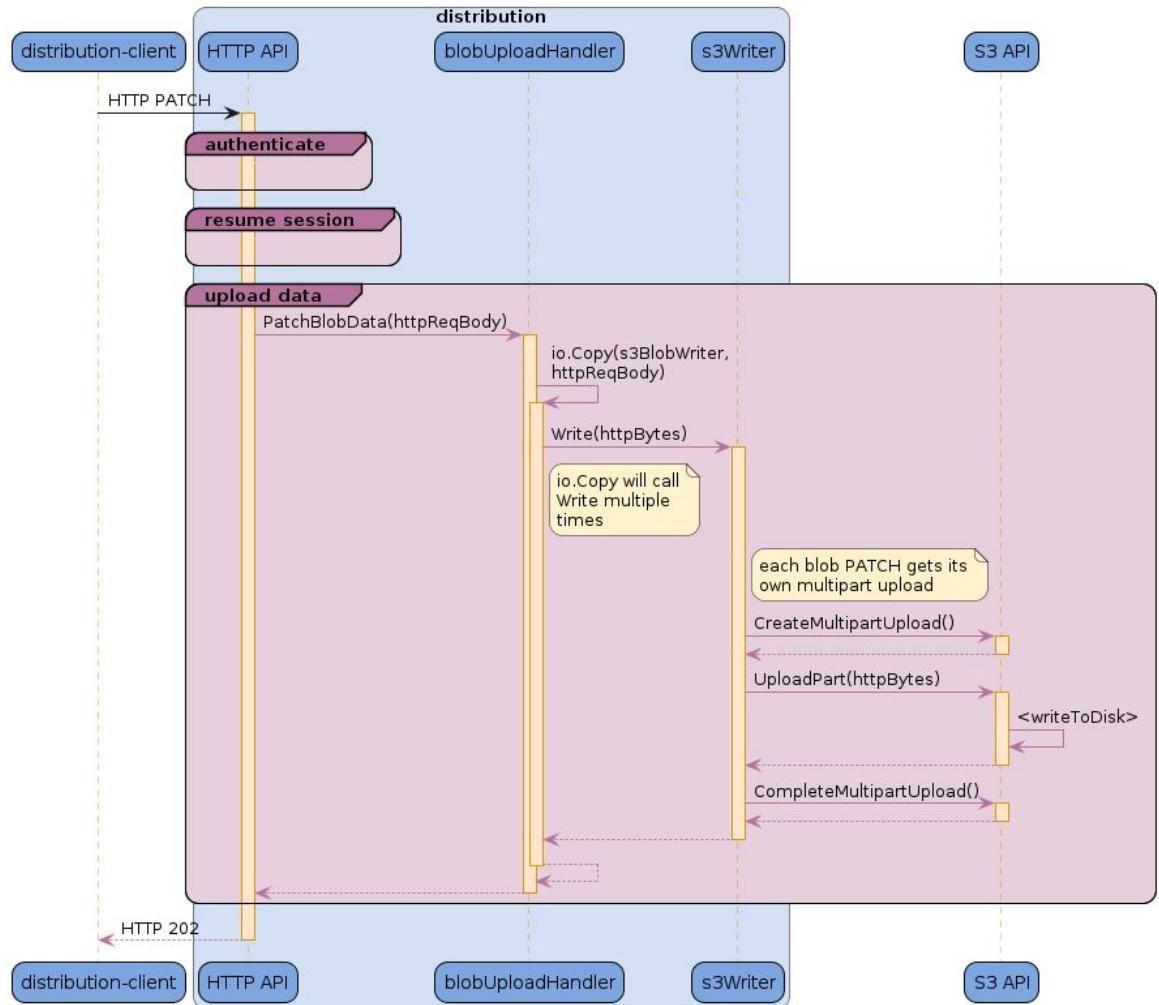


# Resume Session

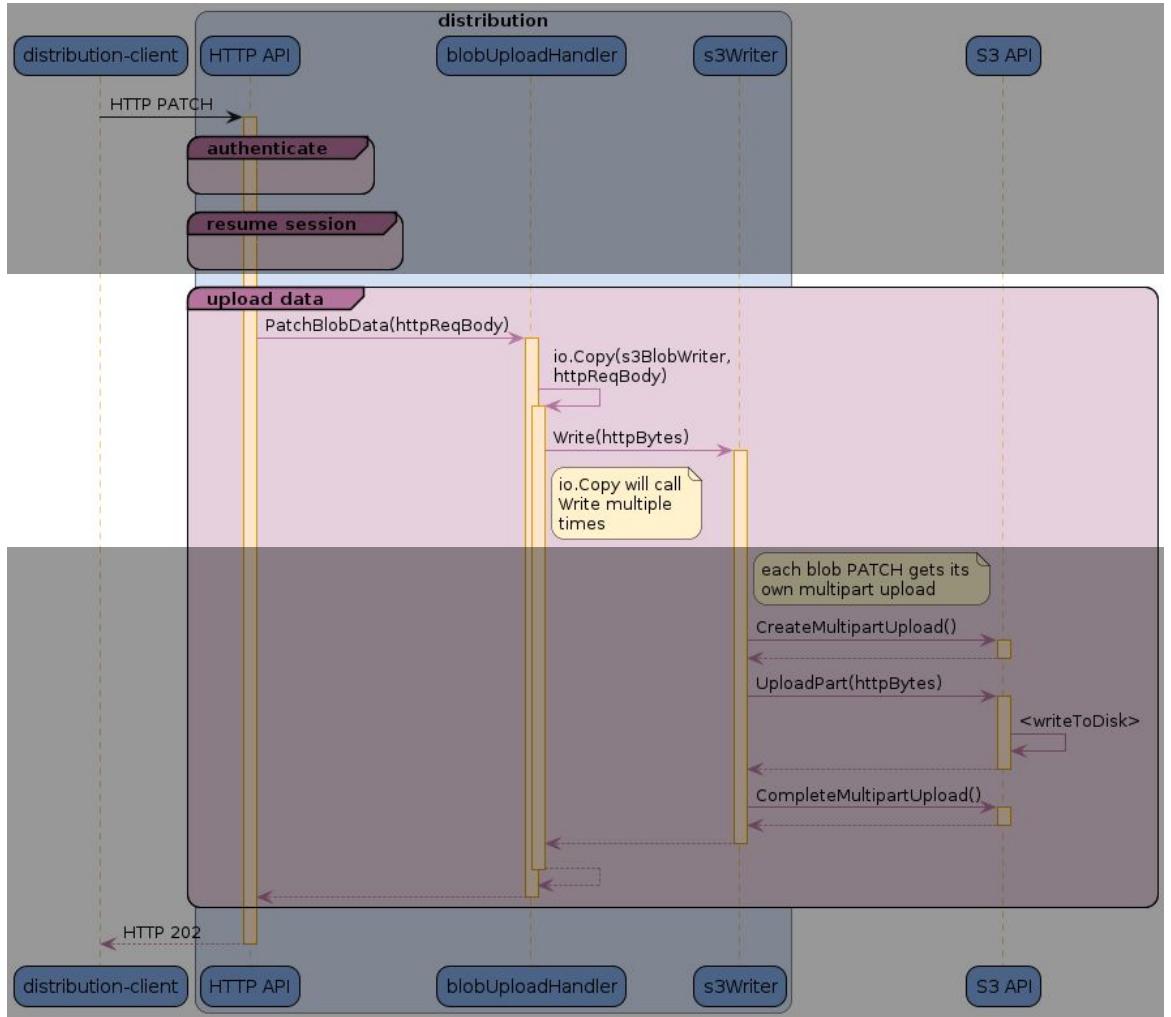
- StorageDriver.Writer()
- Returns an S3-specific blob writer to enable uploading data.



# Data Upload

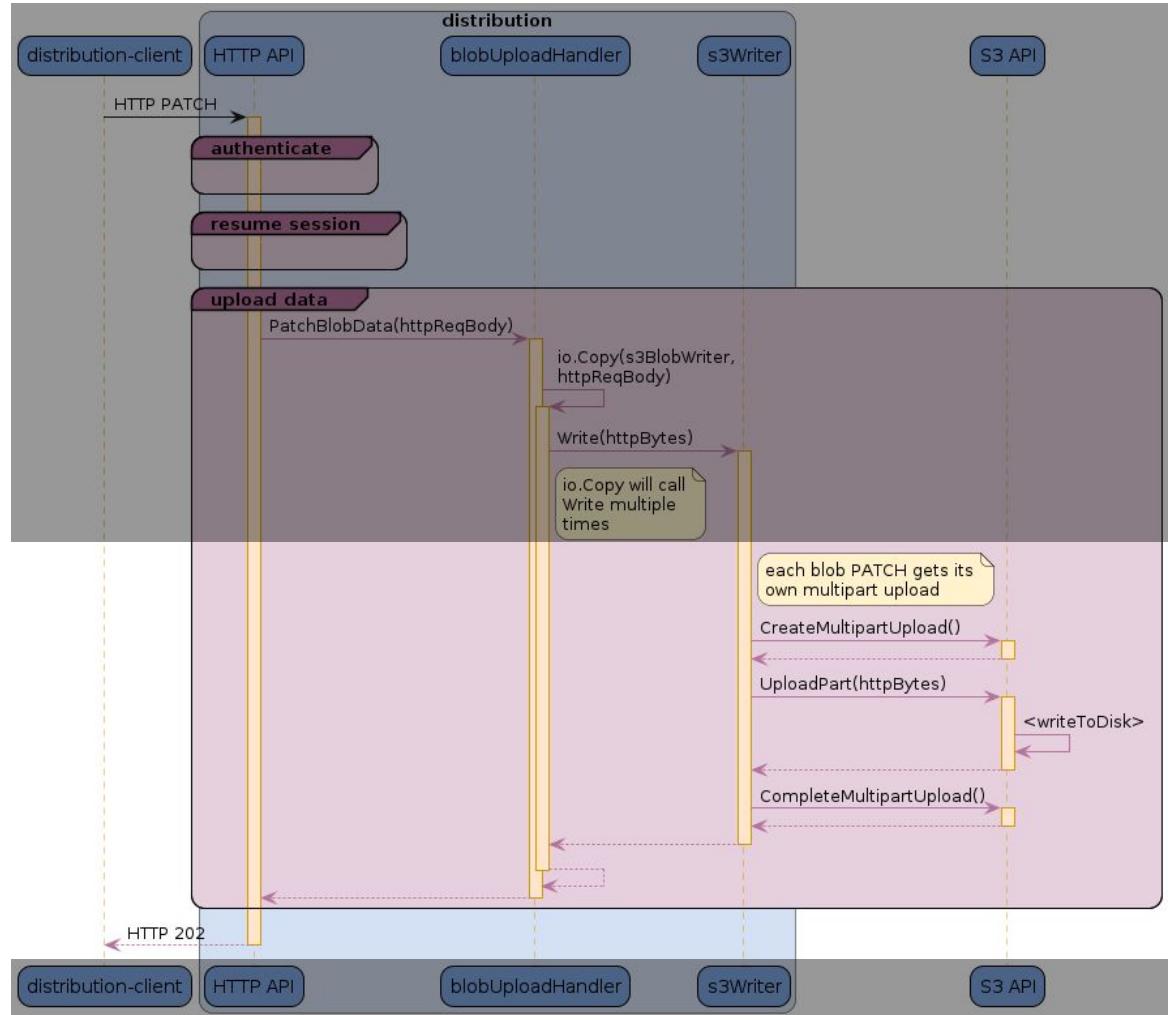


# Data Upload



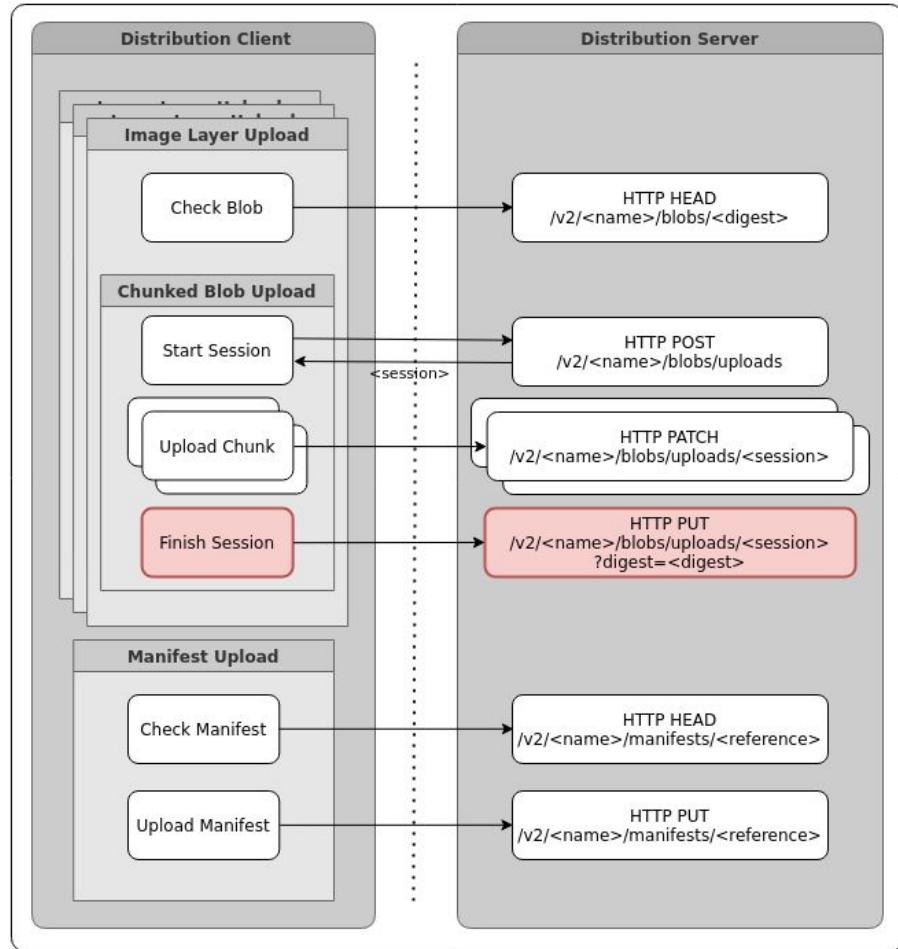
# Data Upload

- POST-PATCH-PUT chunked uploads make use of S3's multi-part upload API.
- PATCH calls may call CreateMultipartUpload if a



# HTTP PUT

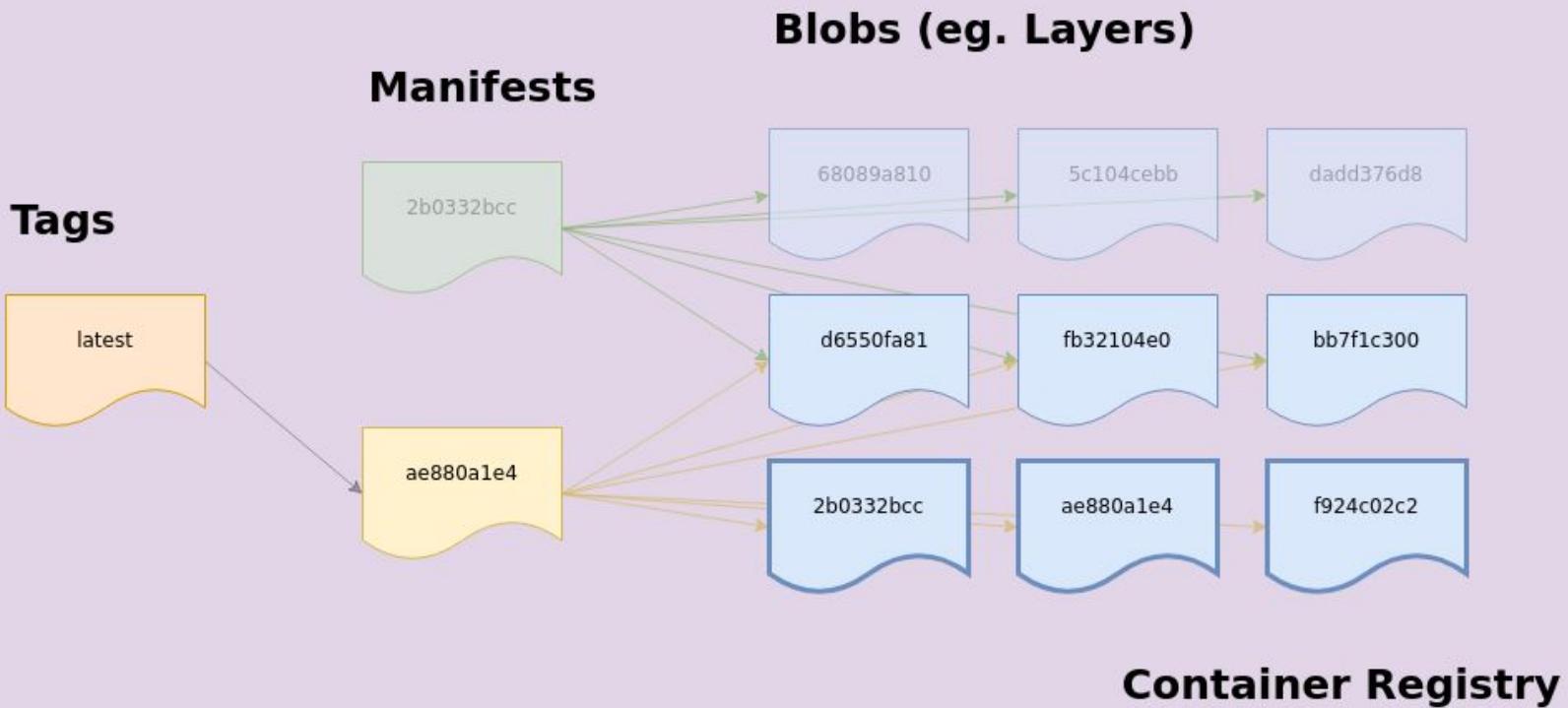
- finishes POST-PATCH-PUT session
- calls CompleteMultipartUpload



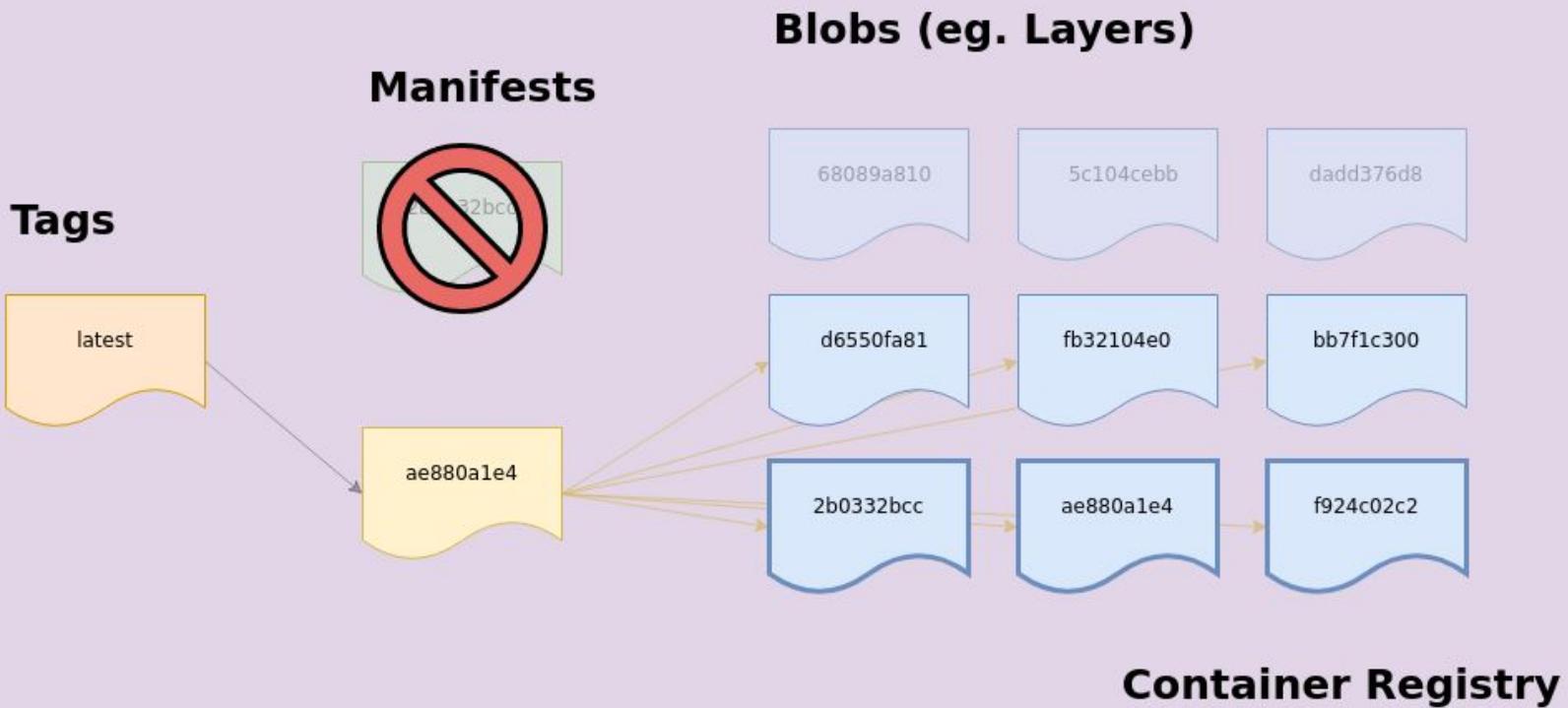
# Garbage Collection



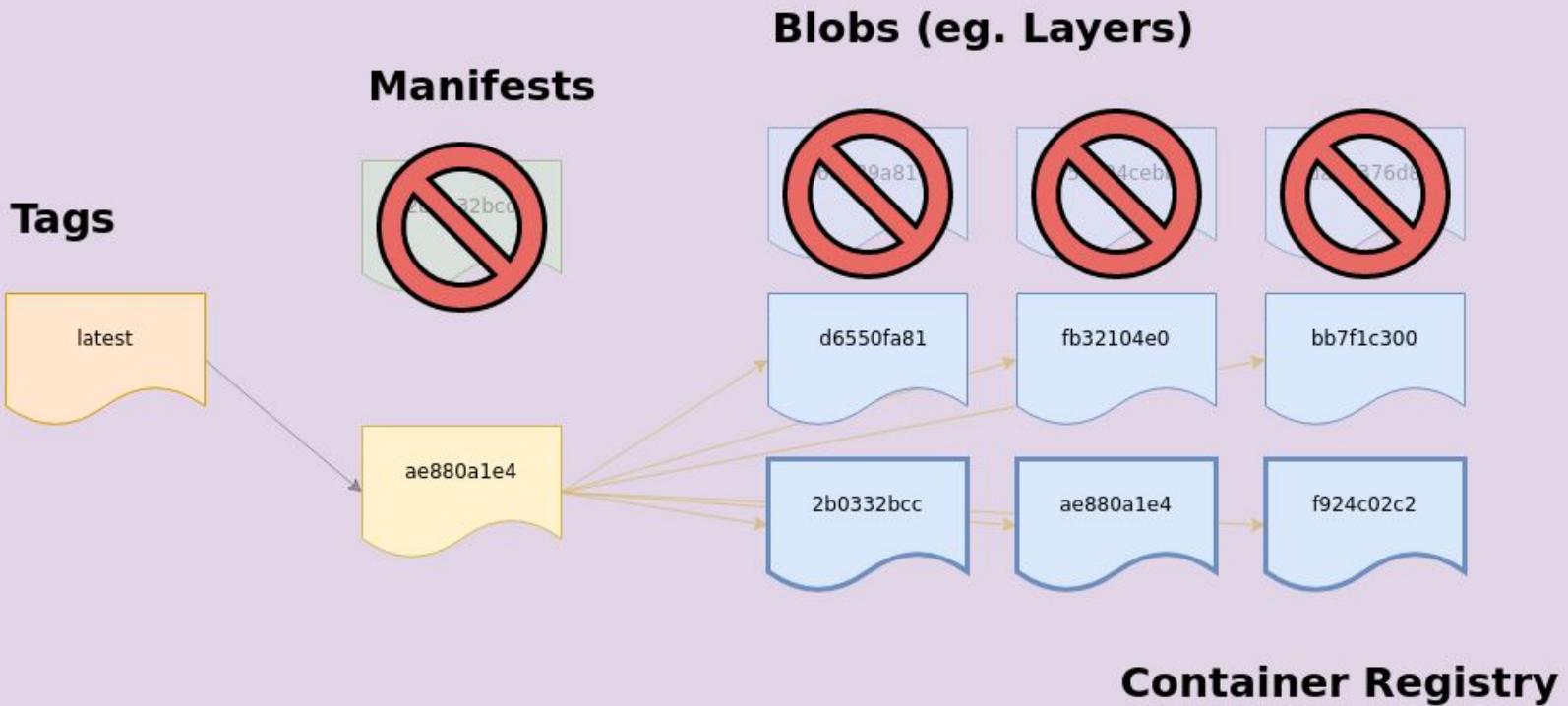
# What is Garbage Collection?



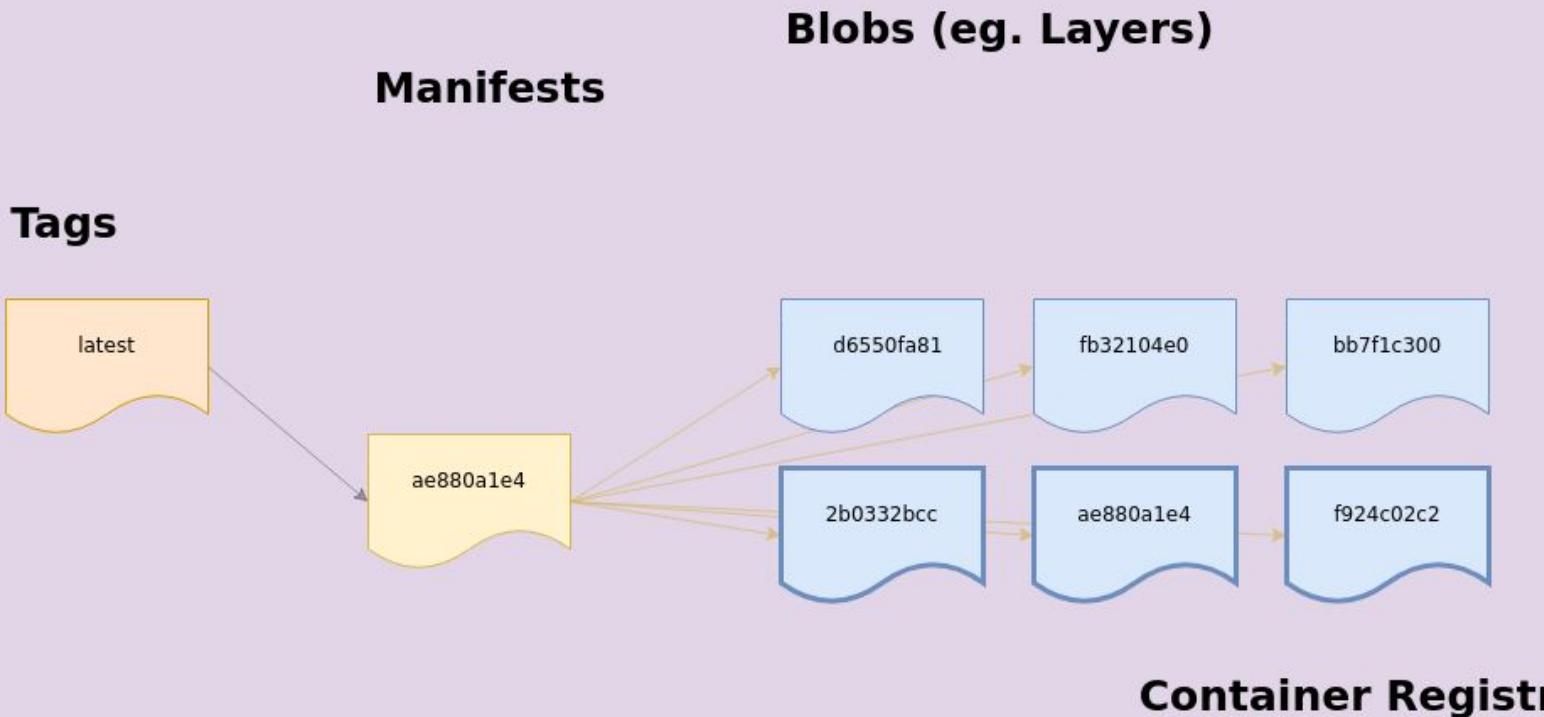
# What is Garbage Collection?



# What is Garbage Collection?

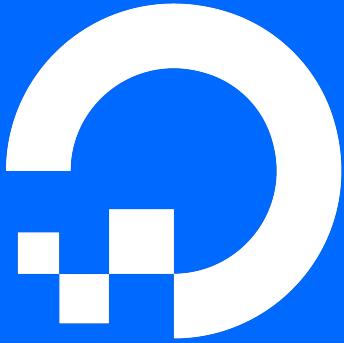


# What is Garbage Collection?



# Why Garbage Collection is Necessary

- Reads, writes, and deletes can be happening simultaneously on a given registry.
- Distribution doesn't guarantee atomic operation.
- A distribution registry must be set to read-only mode for GC to work.



# Thank You!

**Wayne Warren** [wwarren@do.co](mailto:wwarren@do.co)  
**Adam Wolfe Gordon** [awg@do.co](mailto:awg@do.co)