

Netflix Drive

Cloud file system for Studio Assets & Workflows

Tejas Chopra, Sr. Software Engineer, Netflix

Oct 12, 2022



Agenda

Netflix Drive: Introduction and motivation

Comparison with Cloud file systems

Architecture Overview

Use cases

Salient features

Performance

Introduction and motivation

What is Netflix Drive?

- Multi-interface, multi-OS cloud file system
- FUSE based
- REST API interface
- Capable of supporting disparate data, metadata, and event/alerting backends.

Motivations

- Asset metadata and data managed by disparate systems and services
- Studio applications expect POSIX file folder interface
- Studio workflow pipelines - move assets across stages of creative iterations
- Ephemeral project level access controls

Cloud File Systems

A short review and comparison

Cloud File systems

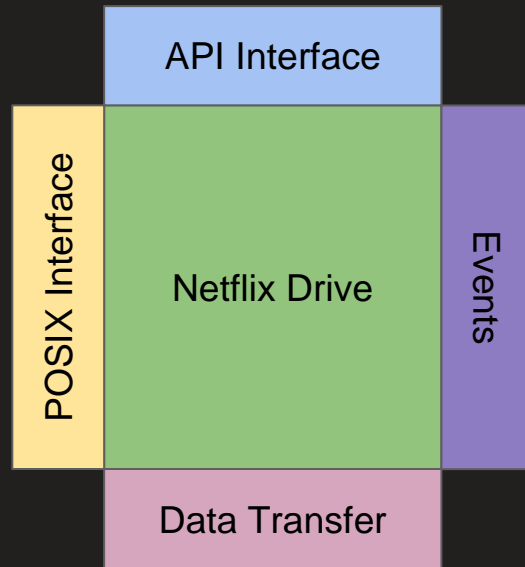
- Namespace management
- Cloud data store and metadata store
 - Mapping files to objects
 - File may be split into multiple objects
 - File attributes: size, checksum, etc.
 - Streaming bytes to data store
- Encryption
- Access control & Authentication

Netflix Drive v/s Typical Cloud File systems

	Typical general purpose cloud file systems	Netflix Drive
APIs	X	✓
Live Filesystem	✓	✓
Use case driven caching tiers	X	✓
Streaming support	✓	✓
Disparate datastores & metadata stores	X	✓
Access control & Authentication	✓	✓

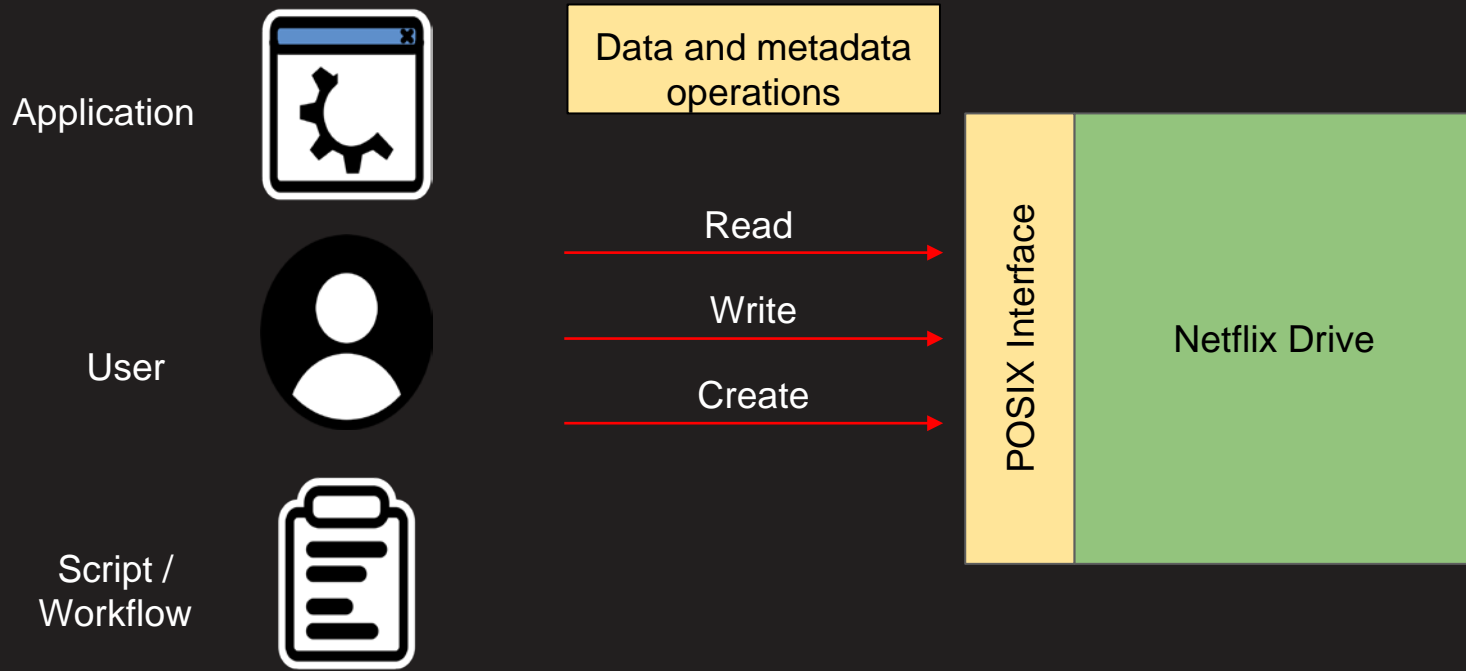
Architecture overview

Multi-interface design

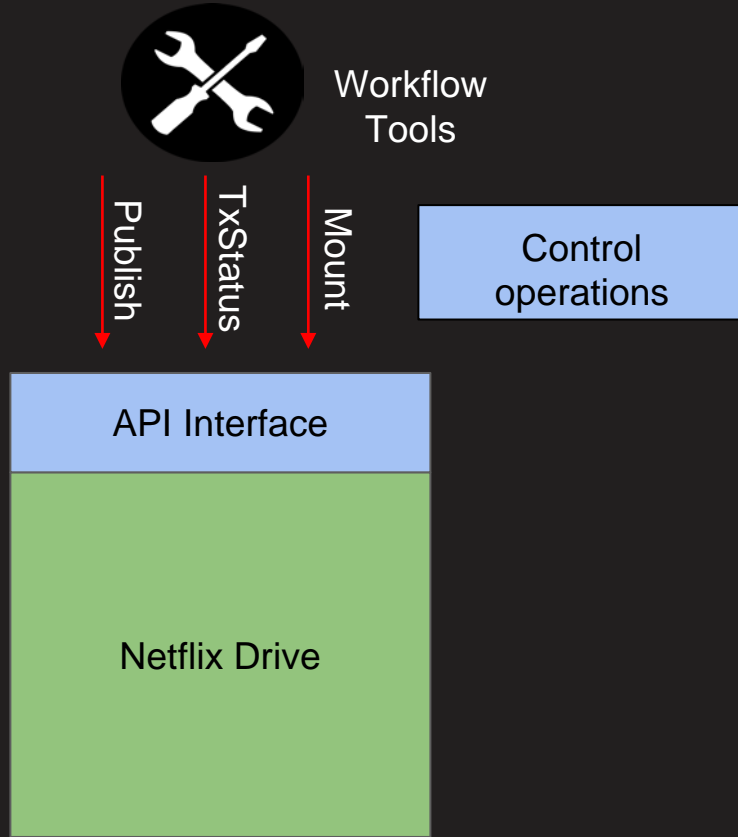


Architecture deepdive

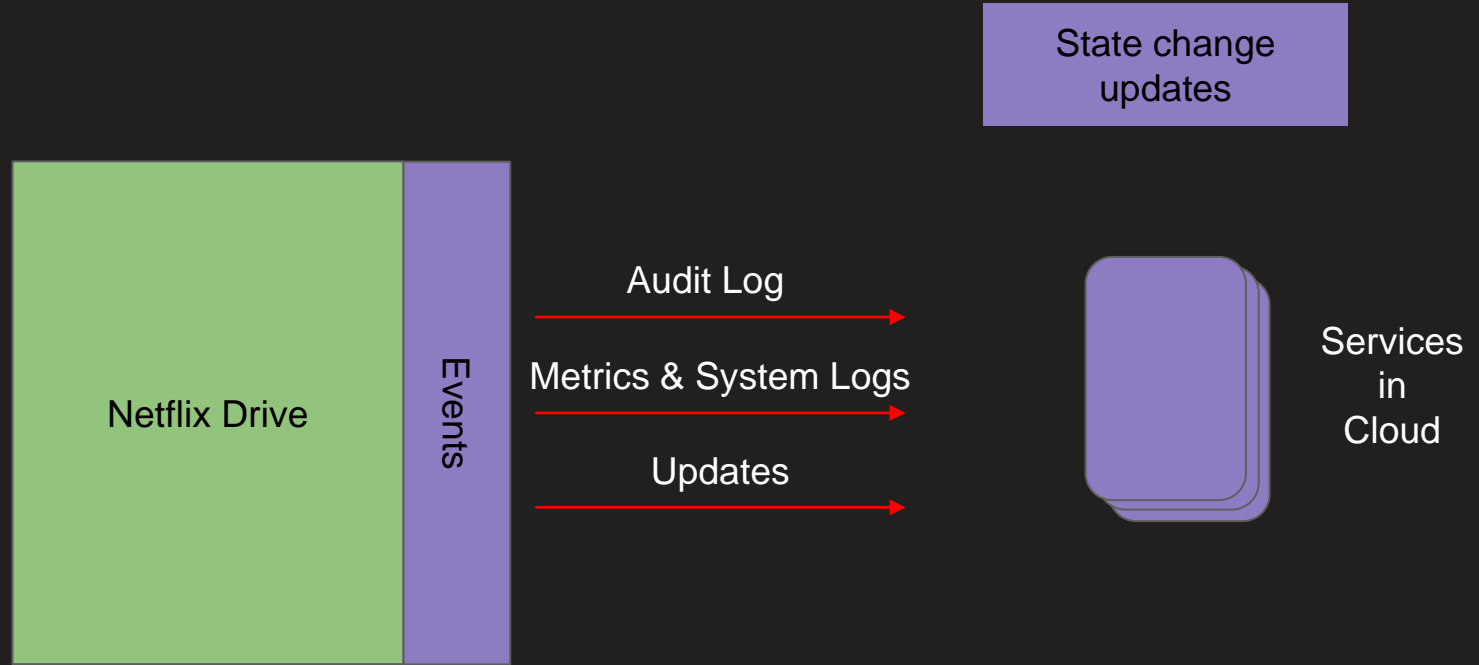
POSIX Interface



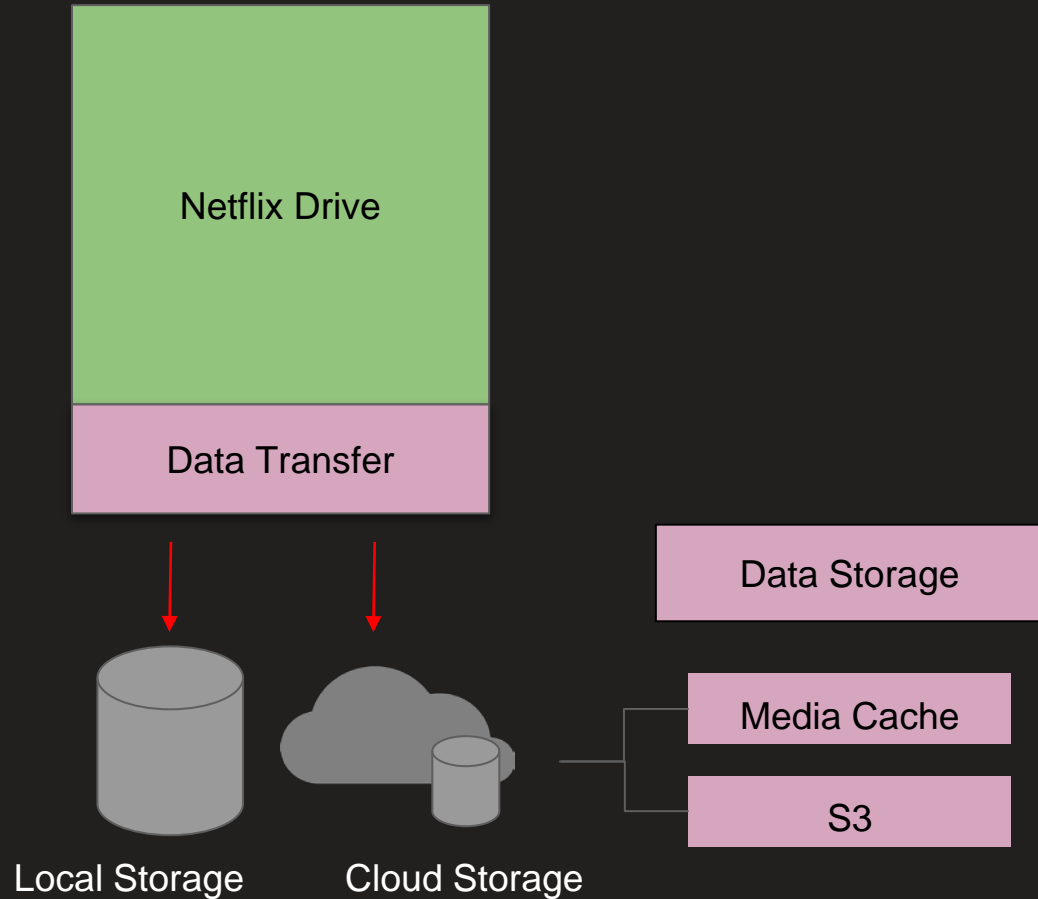
API Interface



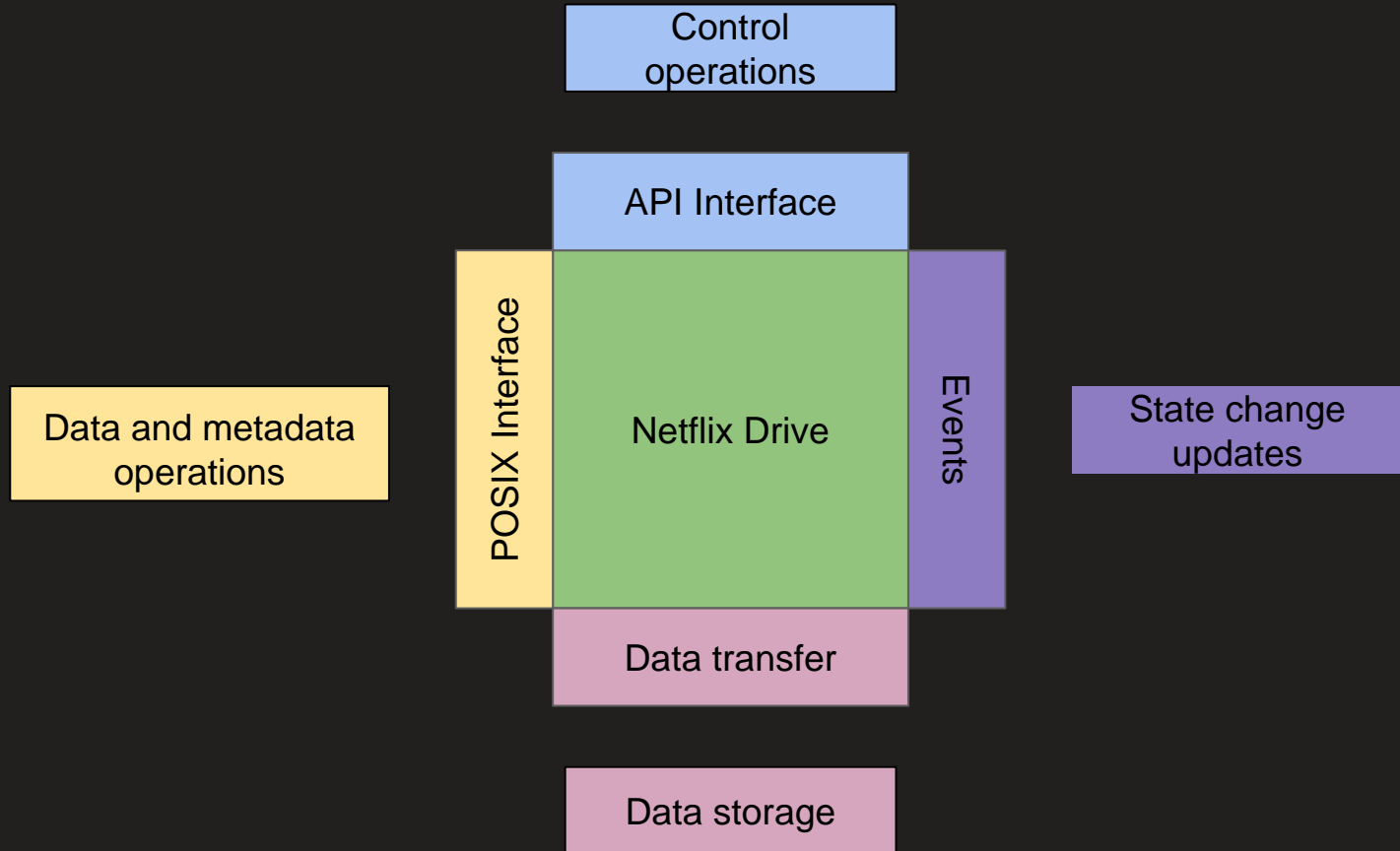
Events



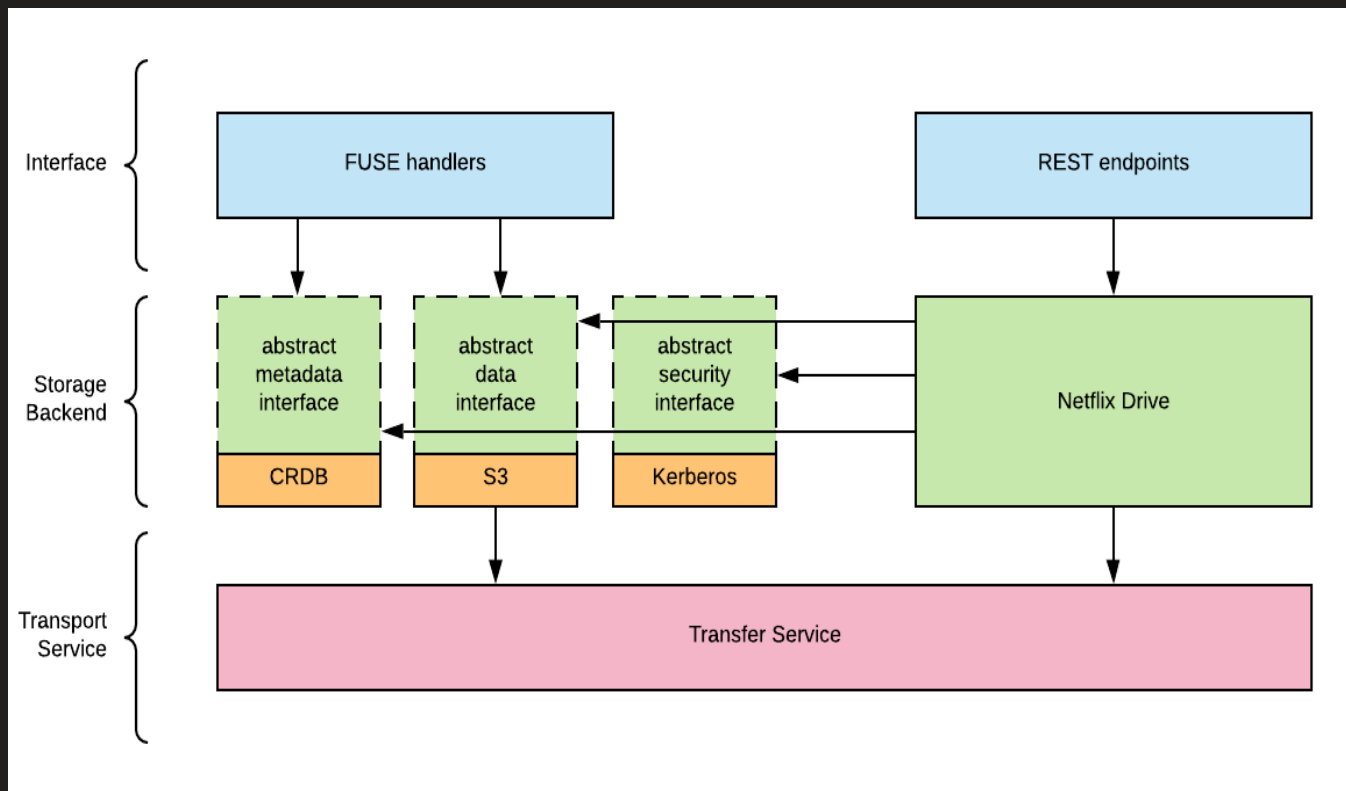
Abstracted Data Transfer Interface



Comes together as...

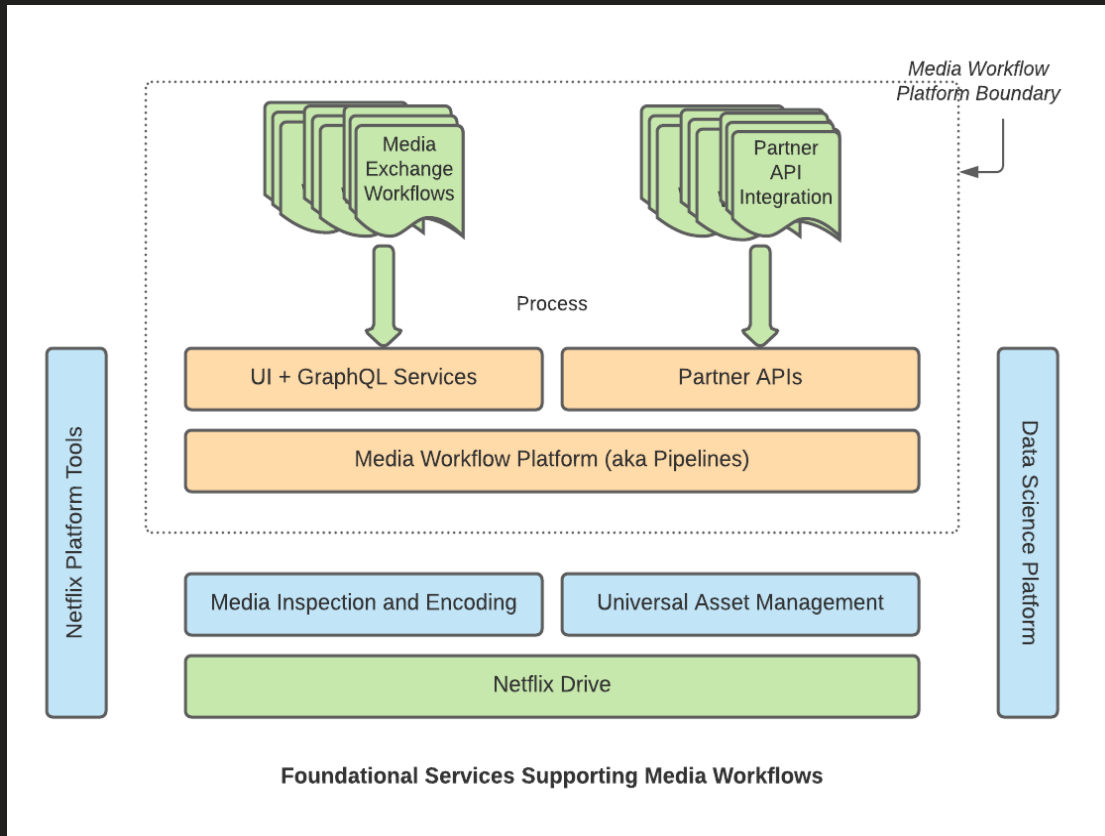


Netflix Drive Internals



Use cases

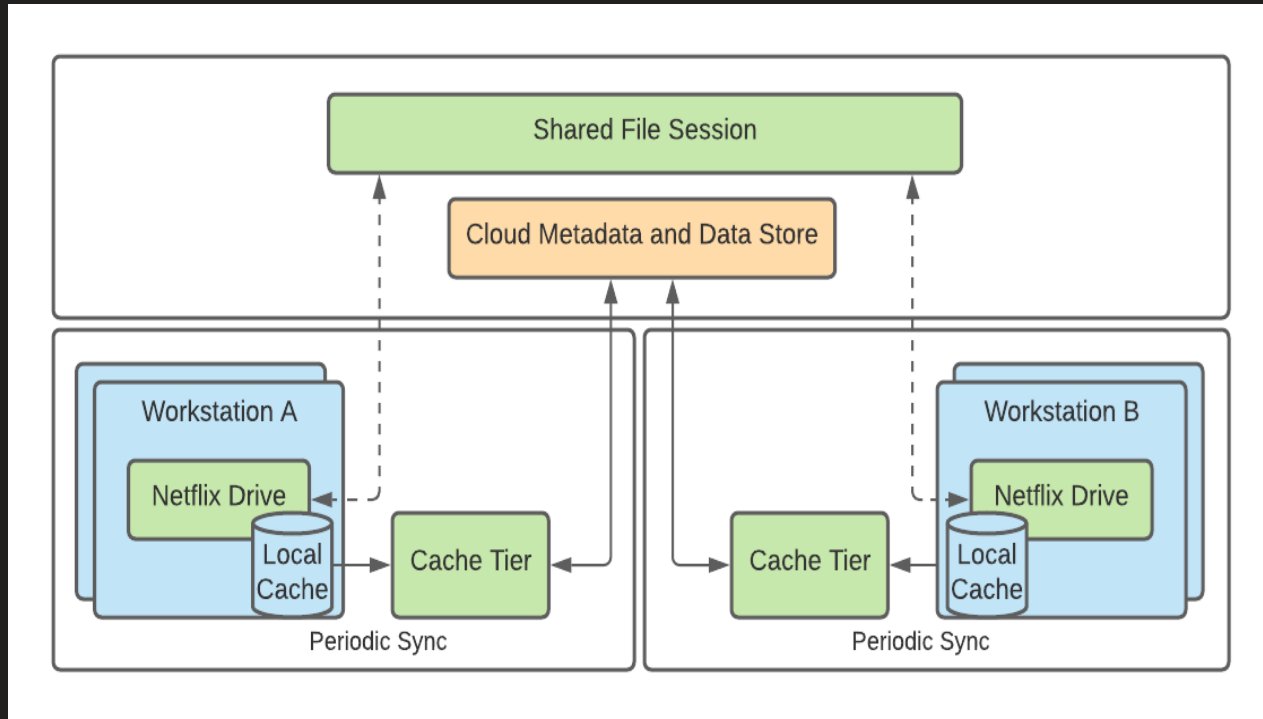
Netflix context for storage in Media Workflows



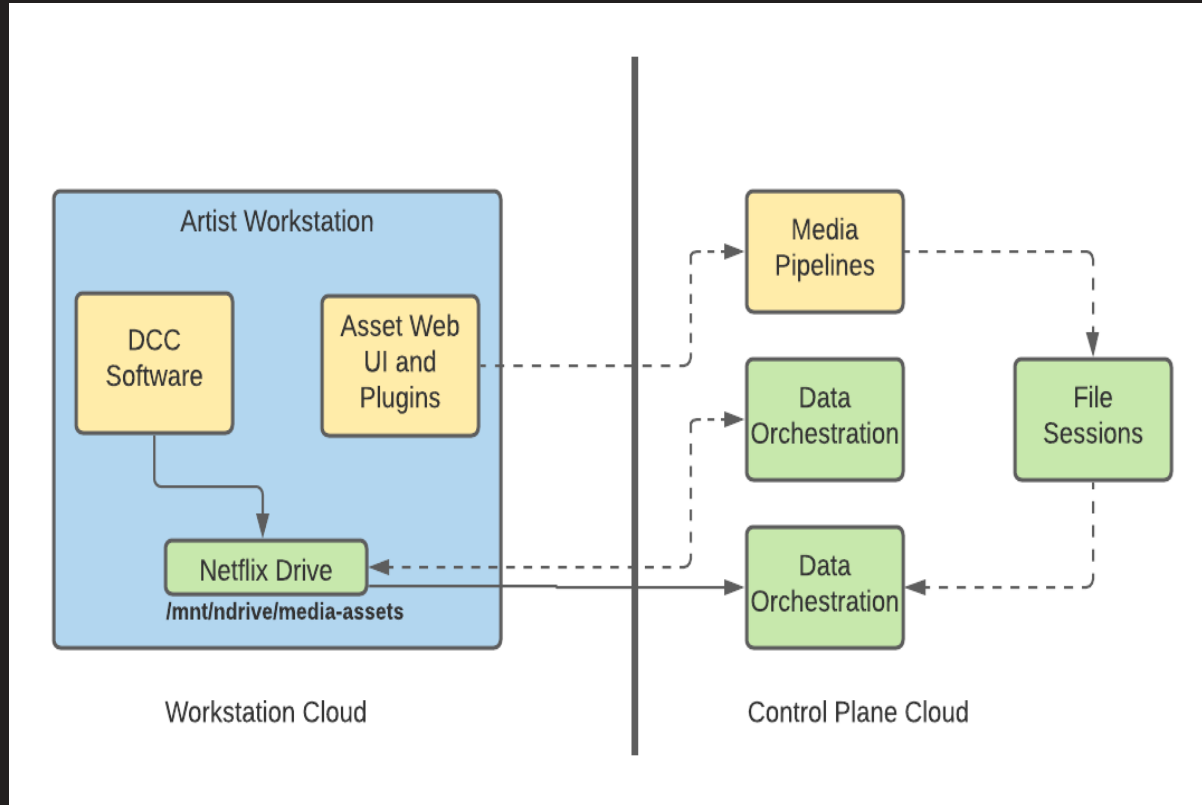
Use Case examples

- Artist workspace
- Software Preferences
- Workflow aware Asset Store
- Container Cloud Storage

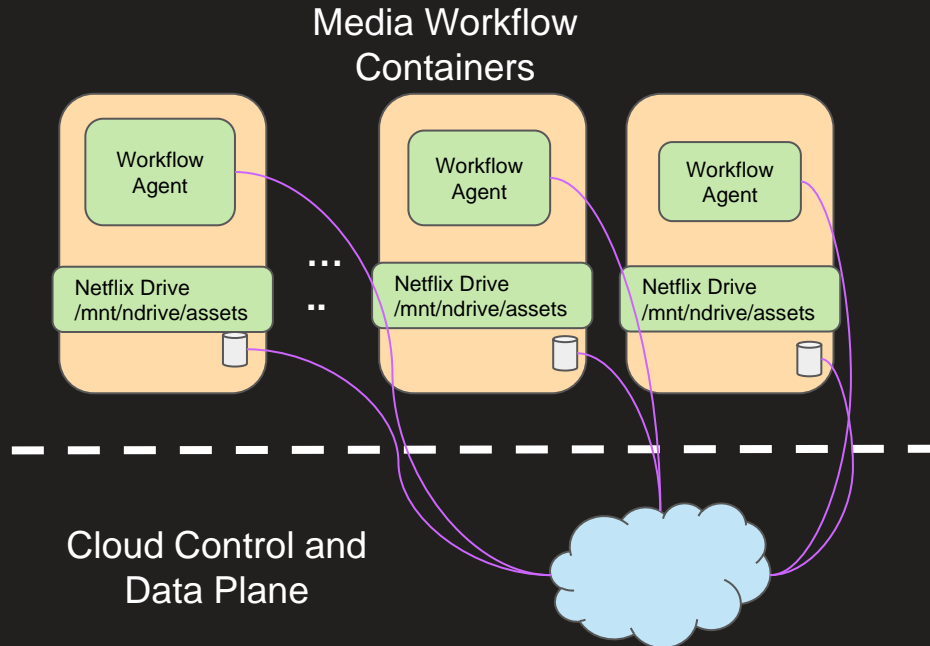
Shared workspaces



Asset Store



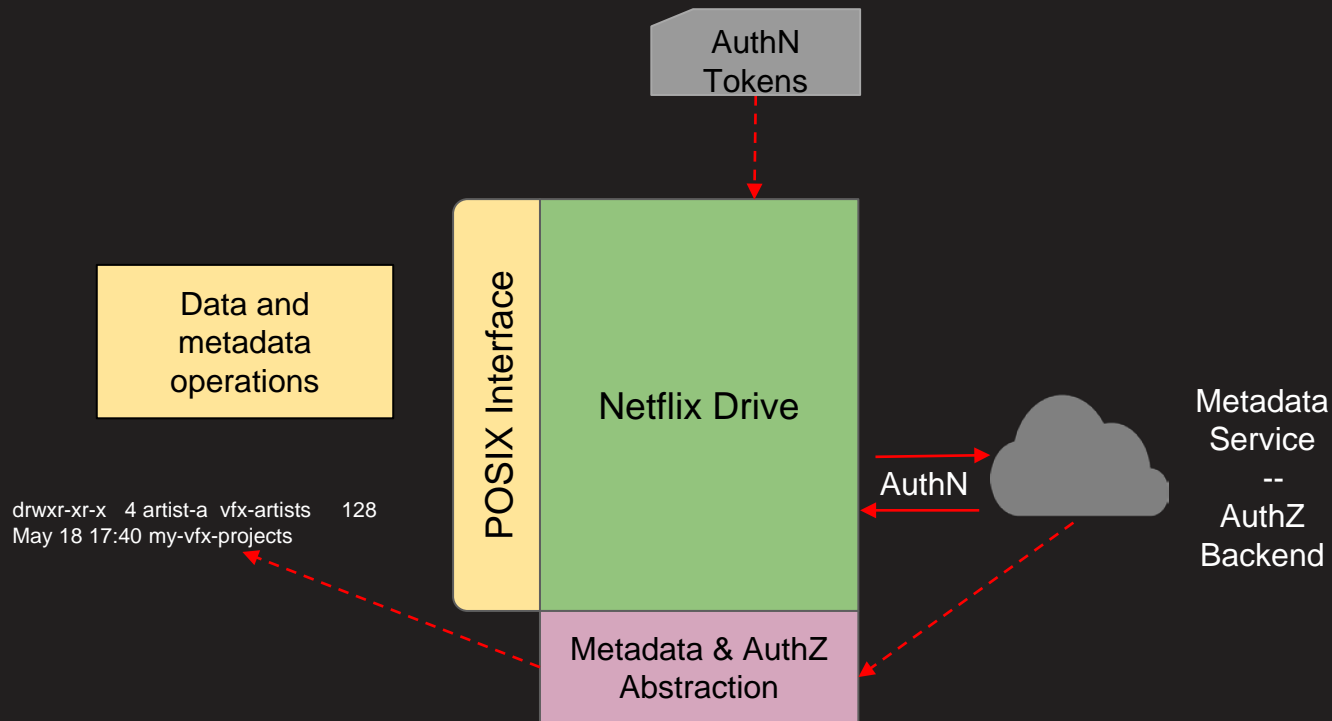
Container Cloud Storage



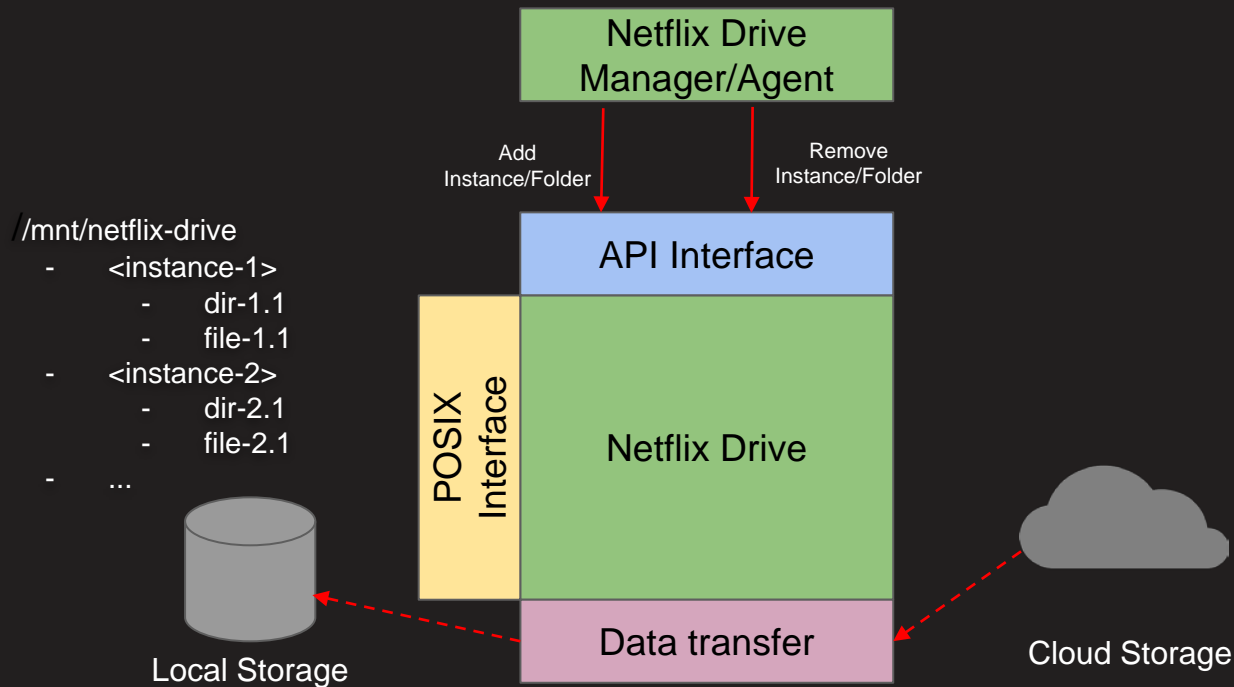
Netflix Drive Features

A peek into some distinguishing features

AuthN and AuthZ



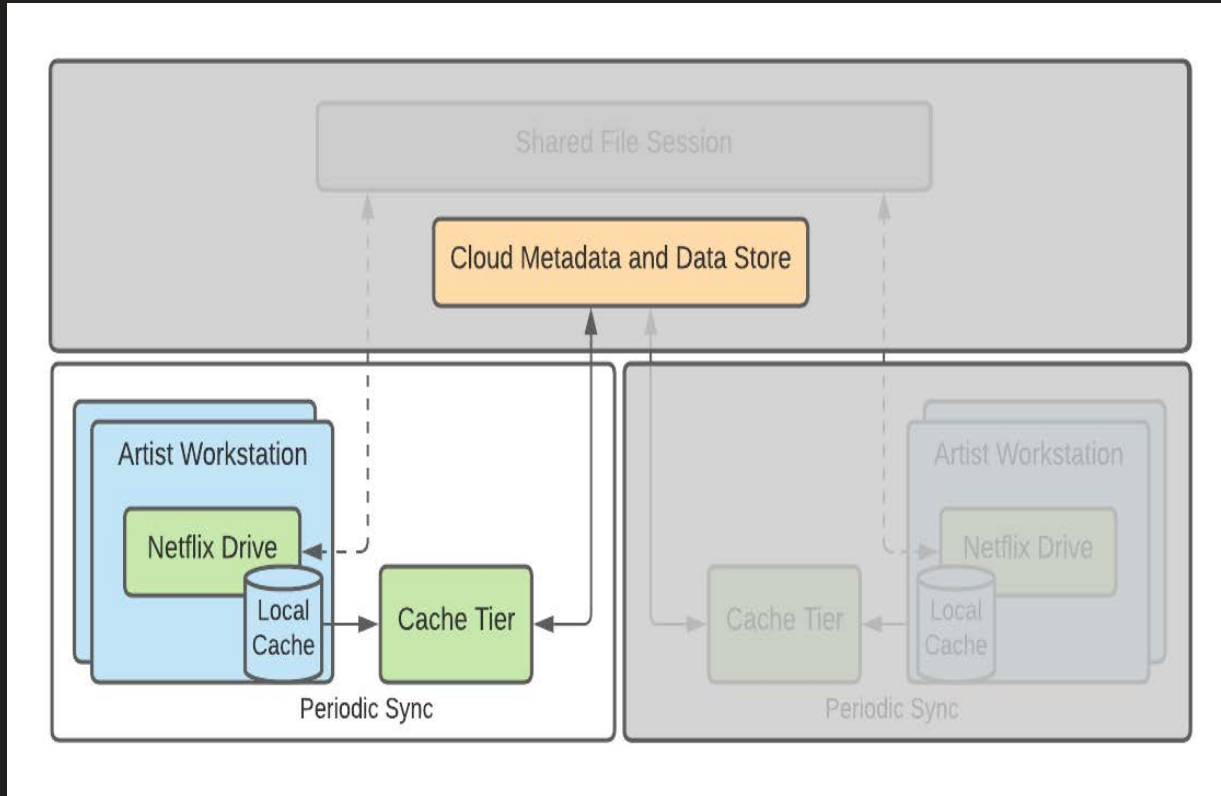
Dynamic namespaces



Multiple instances during bootstrap

```
// Local Storage used by Netflix Drive to cache files
"localFileStore":"/tmp/cache",
// Netflix Drive instances manifested under the mount.
// Here we have two instances: dynamic & user, each with a
// different backing data store & metadata store.
"instances":[
  {
    // Metadata store: Redis
    "metadataStoreUrl":"redis.netflix.test.com",
    // Data store: S3
    "dataStoreUrl":"s3.netflix.test.com",
    // Unique workspace for data persistence
    "filesystem":"redis-s3-store",
    // User security certificate & keys
    "sslCertPath":"/certs/user.crt",
    "sslKeyPath":"/certs/user.key",
    // Optional nodes that Netflix Drive may want to prefetch
    // upon startup
    "nodes":[]
  },
  // Instance type: dynamic, user, etc.
  "instanceType":"dynamic"
},
{
  // Metadata store: CockroachDB
  "metadataStoreUrl":"crdb.netflix.test.com",
  // Data store: Ceph
  "dataStoreUrl":"ceph.netflix.test.com",
  // Unique workspace for data persistence
  "filesystem":"crdb-ceph-store",
  // User security certificate & keys
  "sslCertPath":"/certs/user.crt",
  "sslKeyPath":"/certs/user.key",
  // Optional nodes that Netflix Drive may want to prefetch
  // upon startup
  "nodes":[]
},
  // Instance type: dynamic, user, etc.
  "instanceType":"user"
}
]
```

Tiered caching



APIs

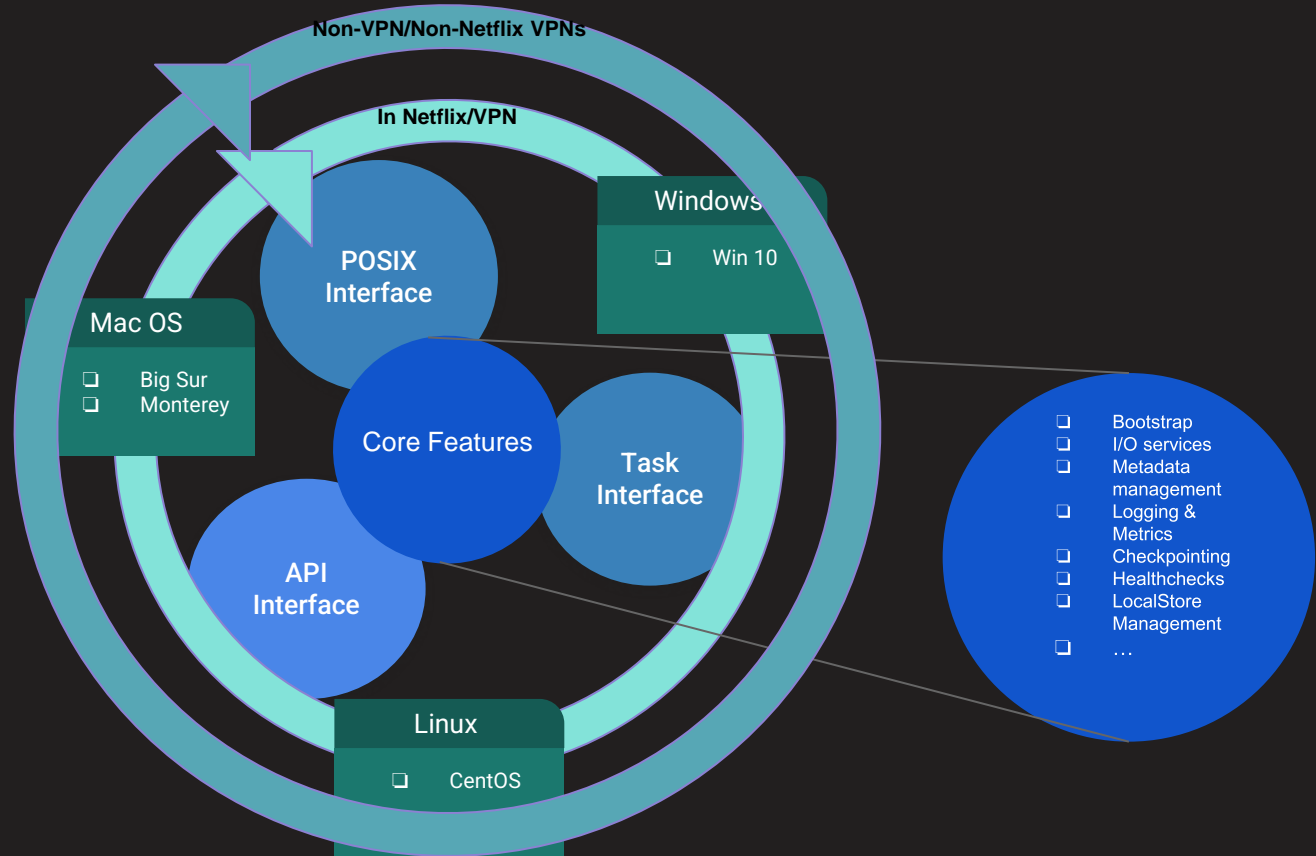
Static Namespace

```
{
  "localFileStore": "/tmp/ndrive-cache",
  "hostOs": "centos",
  "instances": [
    {
      "dataStoreUrl": "https://ds.netflix.net",
      "metadataStoreUrl": "https://mds.netflix.net",
      "taskDispatcherType": "orchestrator",
      "taskDispatcherUrl": "https://task-manager-
service.netflix.net",
      "rootUser": "studio-user-joe@netflix.com",
      "rootName": "project",
      "nodes": [
        "alb1",
        "a2b2",
        "a3b3",
      ],
      "sslCertPath": "/local-cert-store/user.crt",
      "sslKeyPath": "/local-cert-store/user.key"
    }
  ]
}
```

Dynamic Namespace

```
http://localhost:9999/api/v1/stage
{
  [
    {
      "metadataNodeId": "alb1",
      "relPath":
"project/chair.mb",
    },
    {
      "metadataNodeId": "a2b2",
      "relPath":
"project/table.mb",
    },
    {
      "metadataNodeId": "a3b3",
      "relPath":
"project/mat.mb",
    }
  ]
}
```

CI/CD and Testing



Extras

Netflix Drive Tech Blog

<https://netflixtechblog.com/netflix-drive-a607538c3055>