Netflix Drive

Speakers: Vikram Krishnamurthy, Kishore Kasi

SDC 2021

Speakers

Kishore and Vikram are part of the Global Storage Services team in Netflix' Platform Engineering organization.

The Netflix GSS team develops, builds and operates highly scalable, globally distributed cloud and edge storage solutions for Netflix Studios and other internal services.

What is Netflix Drive?

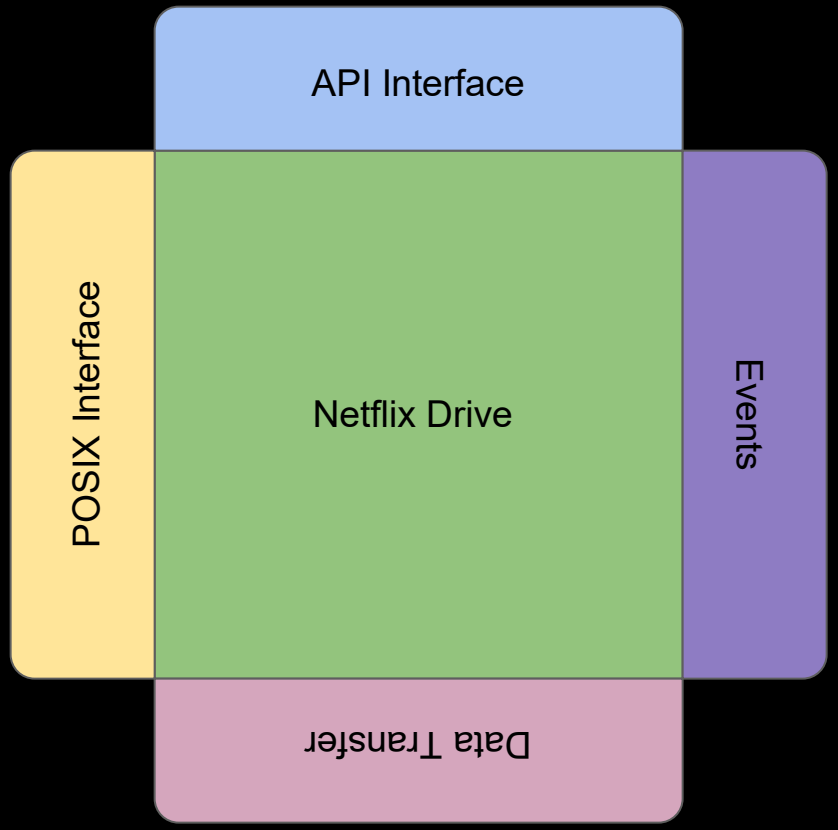
- A multi-interface, multi-OS Cloud File System with the look and feel of a local POSIX file system
- FUSE based (WinFSP based for Windows, macFUSE for macOS)
- Provides a REST API interface for seamless integration with Studio workflows
- Can be plugged into disparate data, metadata and event/alerting backends

Motivation

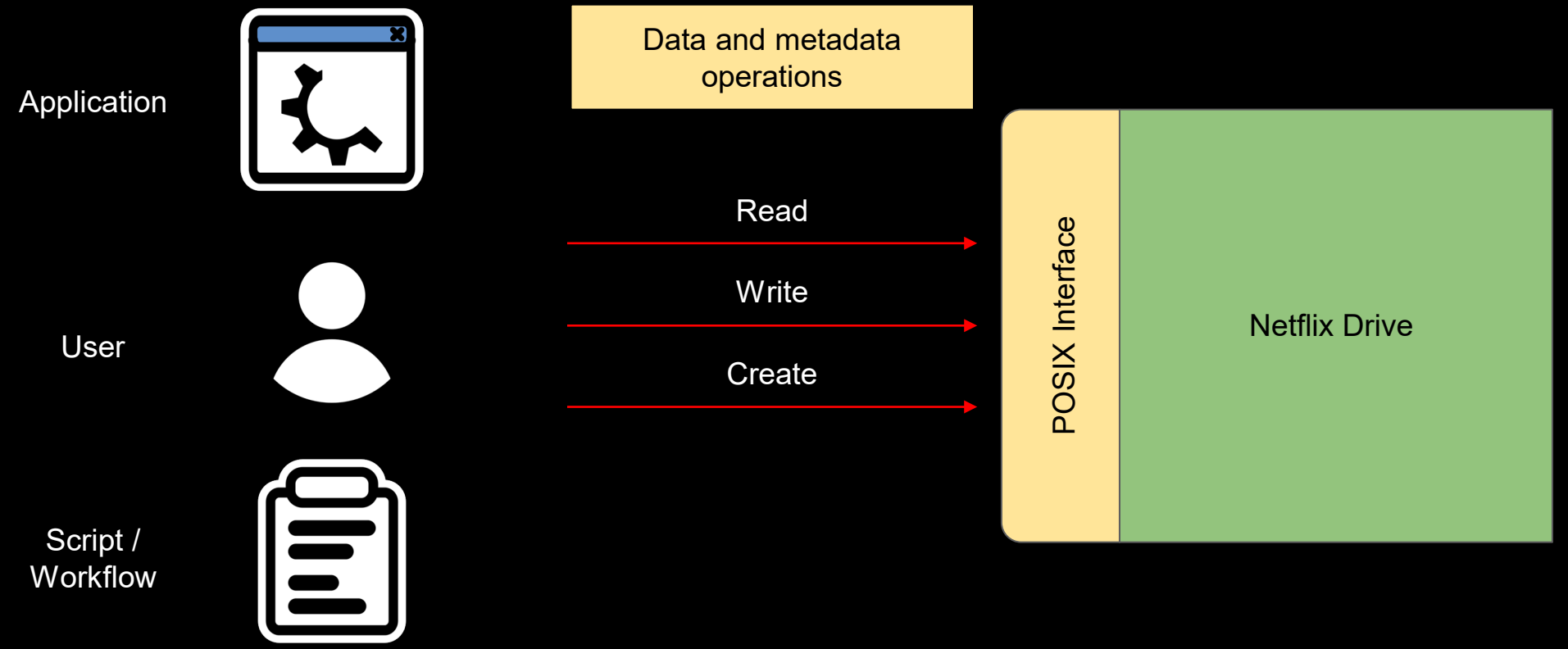
- Asset metadata and data are stored on and managed by disparate systems and services
- Studio applications and artists expect the familiar POSIX file and folder interface
- Studio workflow pipelines need to move assets across stages of creative iterations
- Ephemeral project level access controls for the duration of the creative iteration for a team of globally distributed artists

Netflix Drive

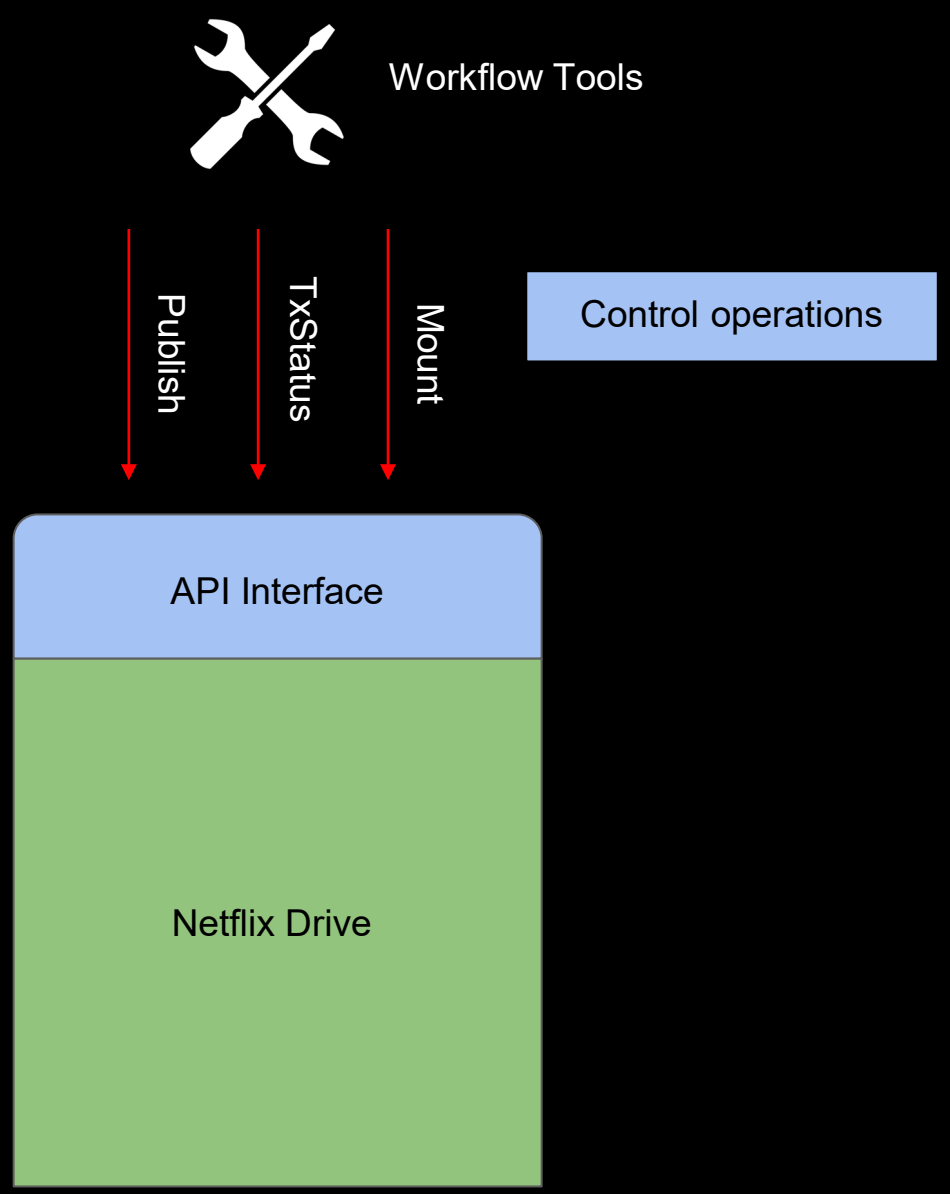
Netflix Drive



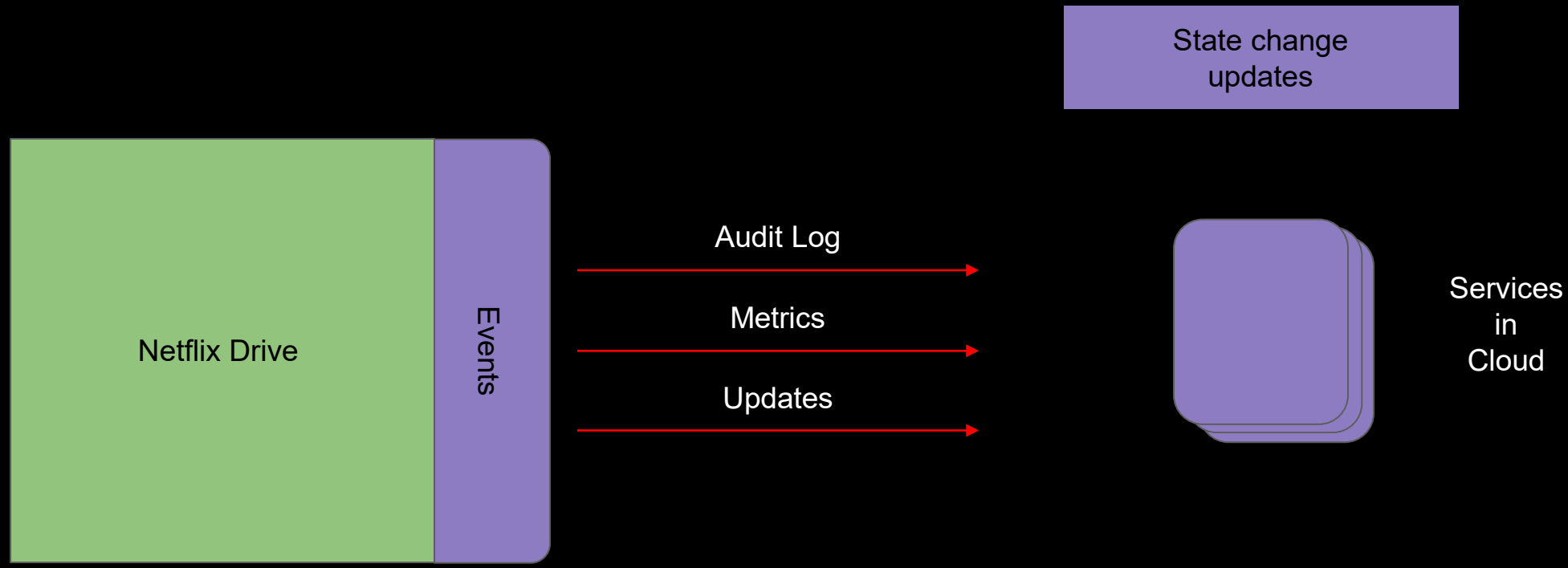
POSIX Interface



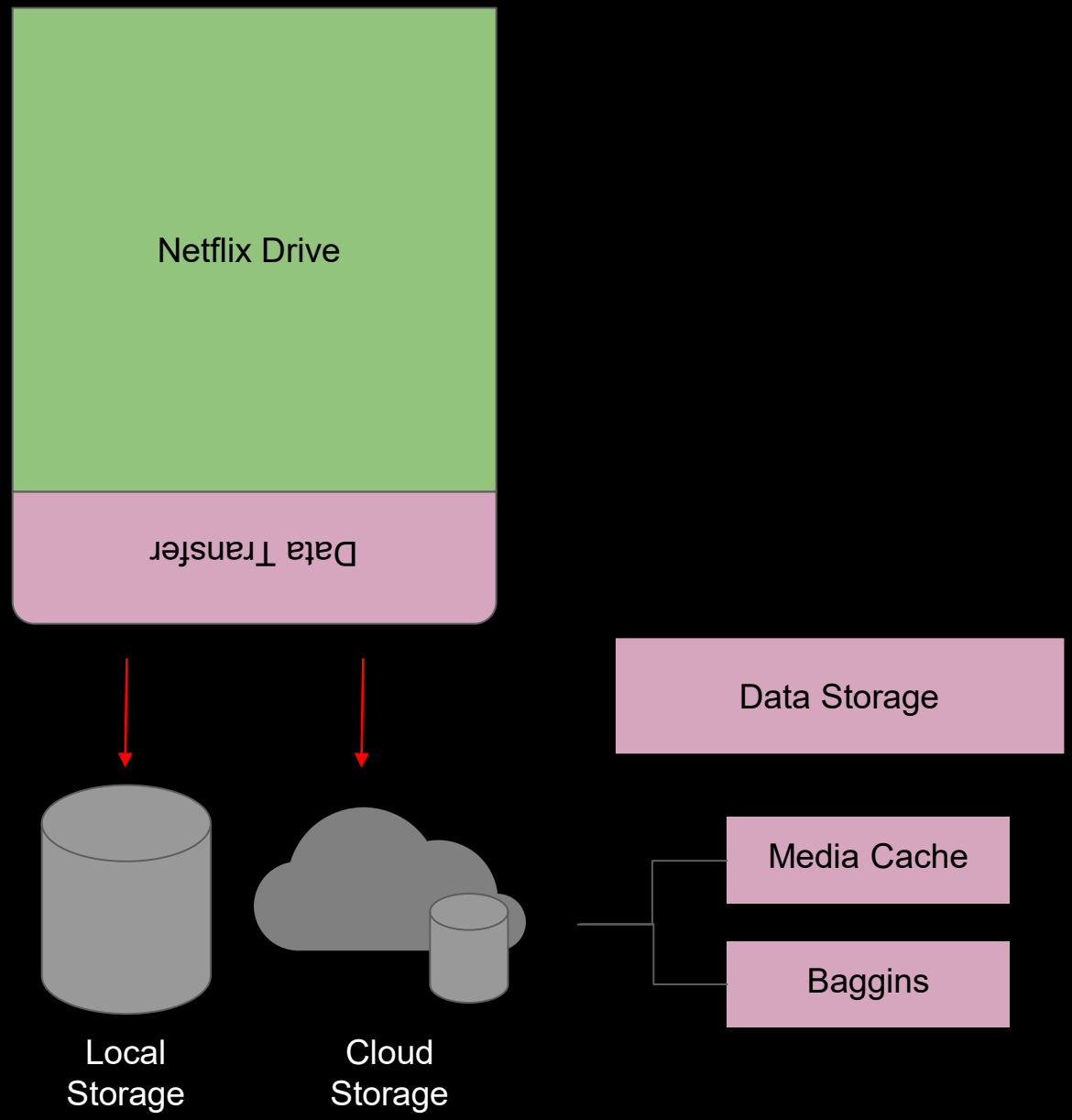
API Interface



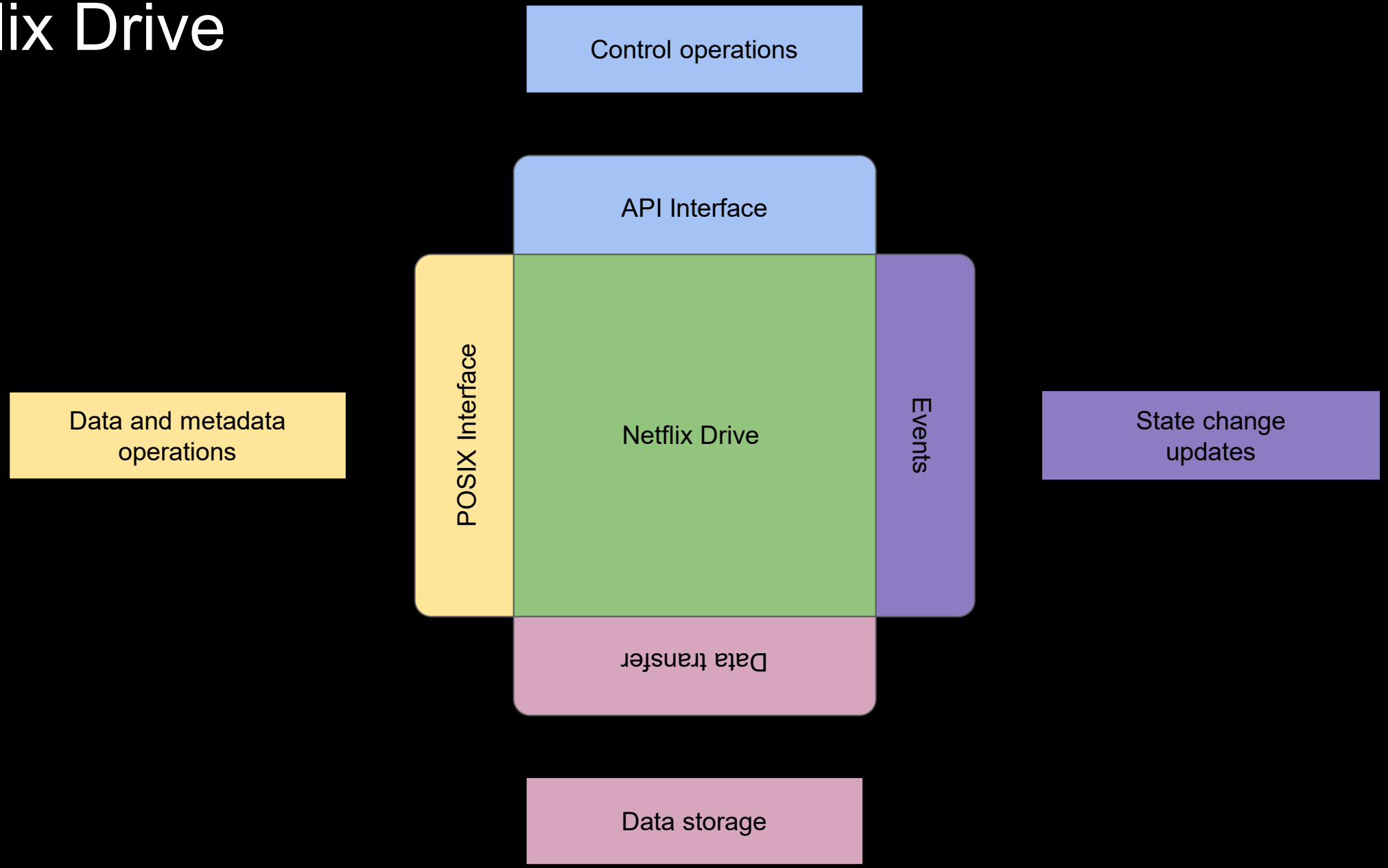
Events


State change updates
Netflix Drive
Events
Audit Log
Metrics
Updates
Services in Cloud

Data Transfer



Netflix Drive



Netflix Drive Anatomy

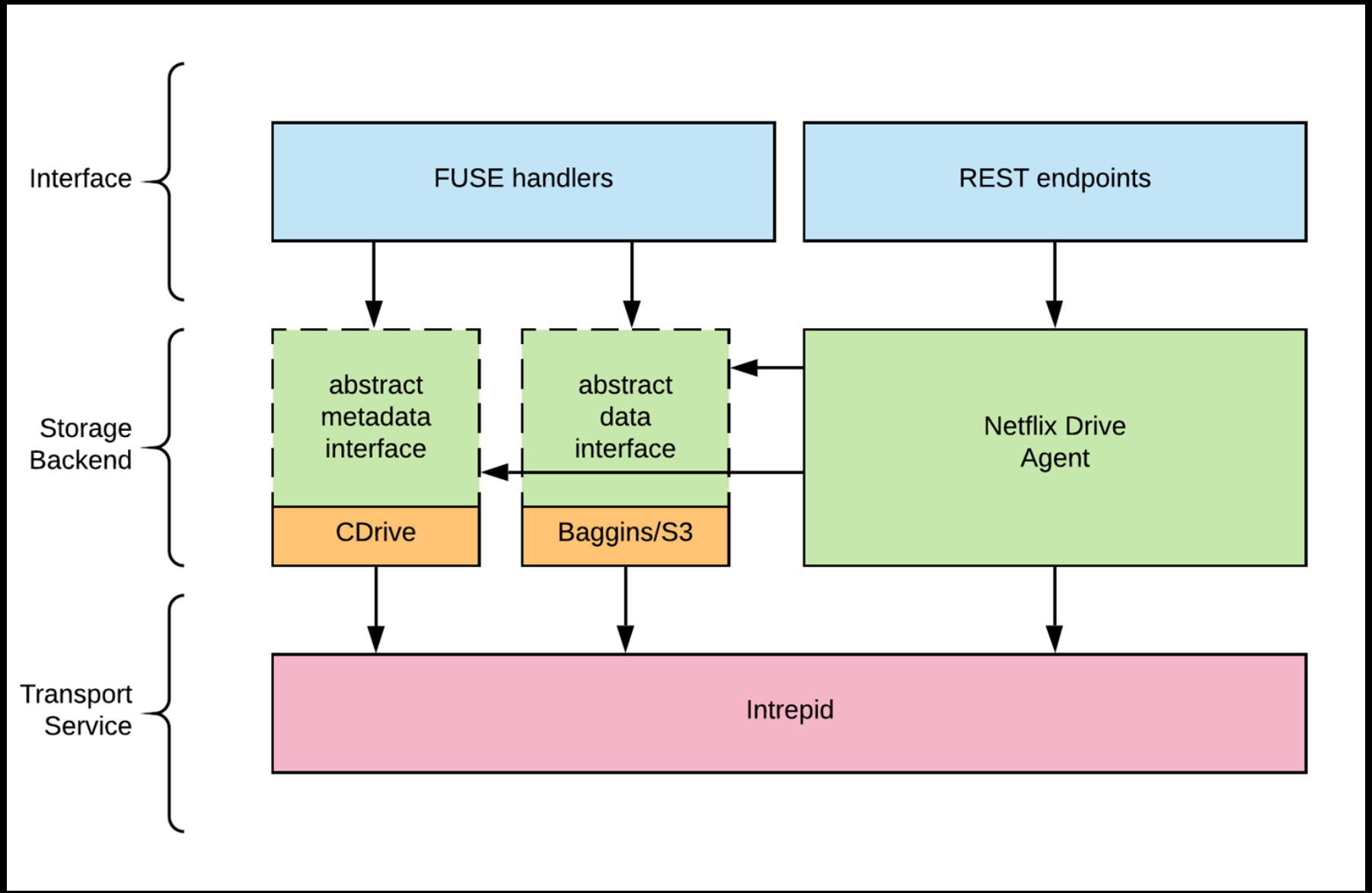
Terminology

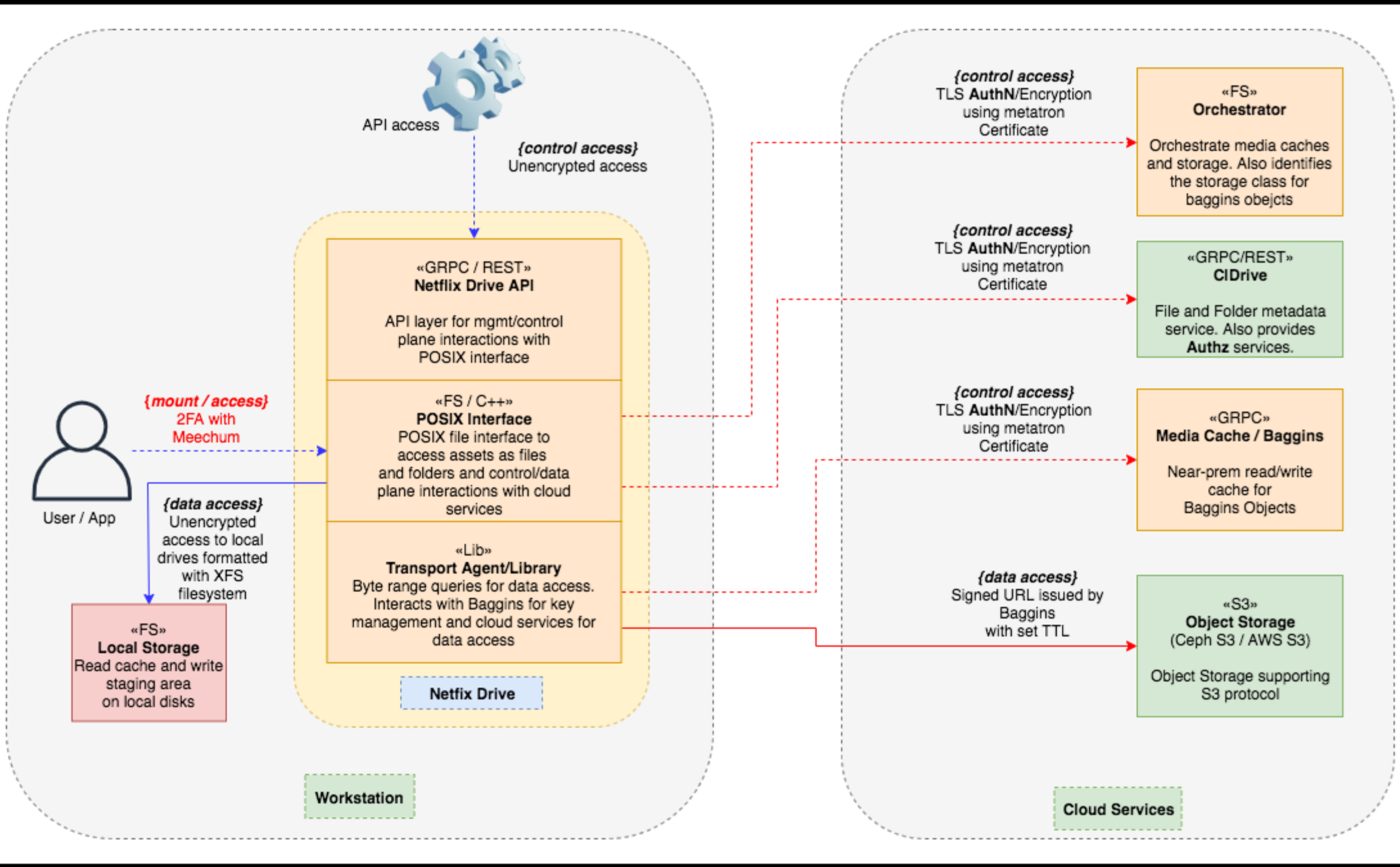
CDrive - A Studio asset aware metadata store used in Netflix

Baggins - Netflix's S3 data store

Mediacache - An S3 region aware caching tier

Intrepid - A high leverage transport protocol used by Netflix services


Interface
Storage Backend
Transport Service
FUSE handlers
REST endpoints
abstract metadata interface
abstract data interface
Netflix Drive Agent
CDrive
Baggins/S3
Intrepid


API access
{control access}
Unencrypted access
«GRPC / REST»
Netflix Drive API
API layer for mgmt/control plane interactions with POSIX interface
«FS / C++»
POSIX Interface
POSIX file interface to access assets as files and folders and control/data plane interactions with cloud services
«Lib»
Transport Agent/Library
Byte range queries for data access. Interacts with Baggins for key management and cloud services for data access
Netfix Drive
{mount / access}
2FA with Meechum
User / App
{data access}
Unencrypted access to local drives formatted with XFS filesystem
«FS»
Local Storage
Read cache and write staging area on local disks
Workstation
{control access}
TLS AuthN/Encryption using metatron Certificate
«FS»
Orchestrator
Orchestrate media caches and storage. Also identifies the storage class for baggins obejcts
{control access}
TLS AuthN/Encryption using metatron Certificate
«GRPC/REST»
CIDrive
File and Folder metadata service. Also provides Authz services.
{control access}
TLS AuthN/Encryption using metatron Certificate
«GRPC»
Media Cache / Baggins
Near-prem read/write cache for Baggins Objects
{data access}
Signed URL issued by Baggins with set TTL
«S3»
Object Storage
(Ceph S3 / AWS S3)
Object Storage supporting S3 protocol
Cloud Services

Typical Lifecycle

- Bootstrap
- Download assets (preload and/or dynamically)
- Modify assets
- Snapshot
- Upload

Initialize

- User identity
- User key
- NDrive mountpoint
- Local Storage (Cache)
- CDrive (Metadata store) Root
- Upload Dir
- CDrive endpoint
- Baggins endpoint

Namespace

From static configuration

```
{
  "localFileStore": "/tmp/ndrive   - cache",
  "hostOs": "centos",
  "instances": [
   {
     "metadataStoreType": "fast",
     "metadataStoreUrl": "https://mds    - 1.test.cloud.netflix.net:8443/api/v1",
     "taskDispatcherType": "orchestrator",
     "taskDispatcherUrl": "https://task    - manager - service.test.cloud.netflix.net:8443",
     "nDriveContainerId": "11  - 12345678",
     "fileSessionId": "11 - 12345678",
     "rootUser": "studio  - user - joe@netflix.com",
     "rootName": "fast  - projects",
     "nodes": [
     ],
     "sslCertPath": "/local  - cert - store/user.crt",
     "sslKeyPath" : "/local - cert - store/user.key"
   }
  ]
}
```

Namespace

Dynamically added via API

```
http://localhost:9999/api/v1/stage
{
        [
                {
                        "metadataNodeId":
"a1b1",
                        "relPath":
"project1/chair.mb",
                },
                {
                        "metadataNodeId":
"a2b2",
                        "relPath":
"project1/table.mb",
                },
                {
                        "metadataNodeId":
"a3b3",
                        "relPath":
"project1/mat.mb",
                }
        ]
```

Update

Filesystem (POSIX) operations

open, rename, move, write, read,
close, save, link, copy, delete, ...

REST API

- `stage`
- `save`
- `progress`
- `checkpointEnable`
- `logLevel`
- `getId`
- `more...`

Publish

Autosave

- Upload checkpointed snapshot at a configurable interval

Explicit Save

```
http://localhost:9999/api/v1/save
{
        [
                "project1/chair.mb",
                "project1/mat.mb",
        ]

}
```

Extras

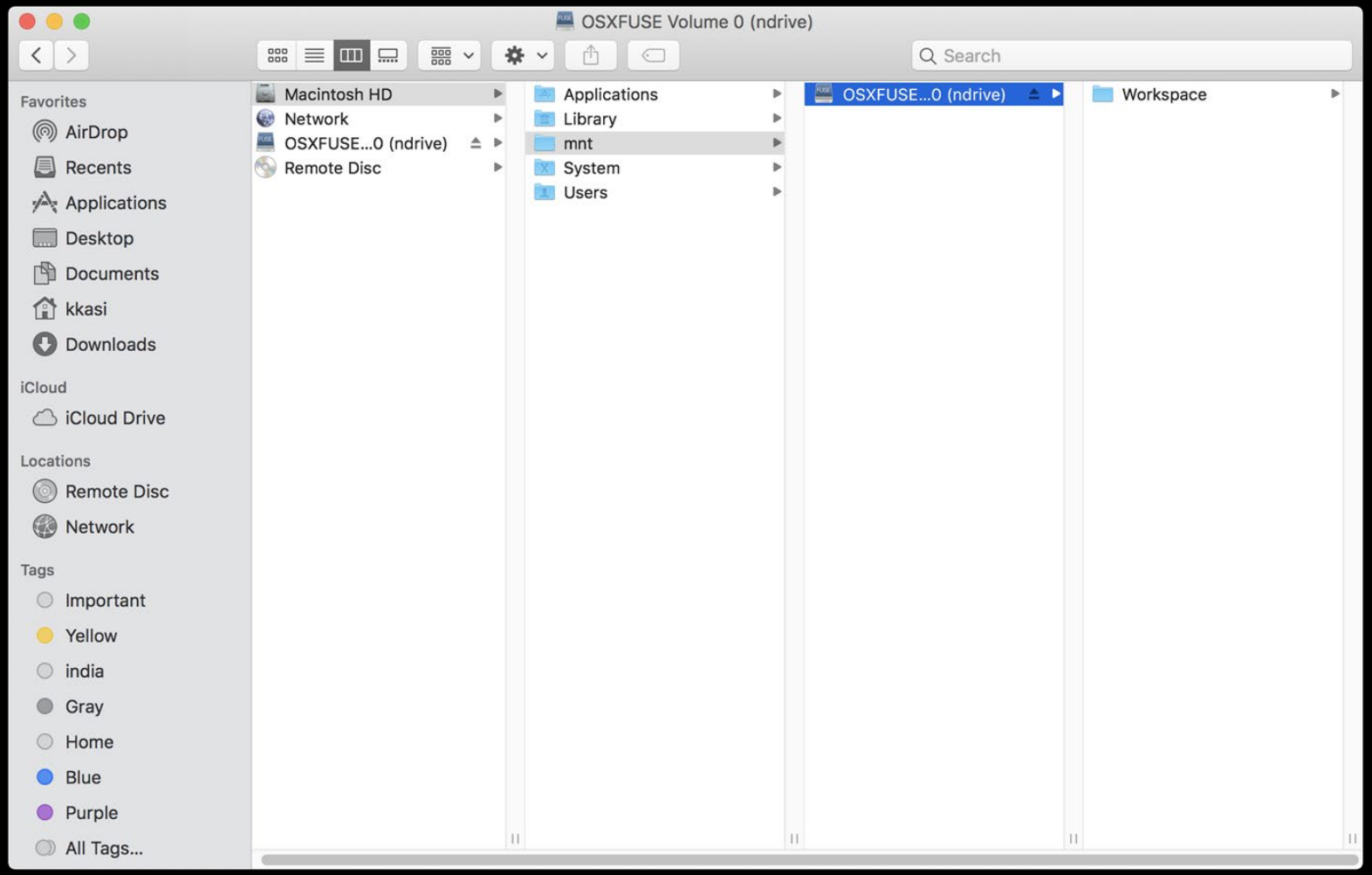
Netflix Drive Tech Blog

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

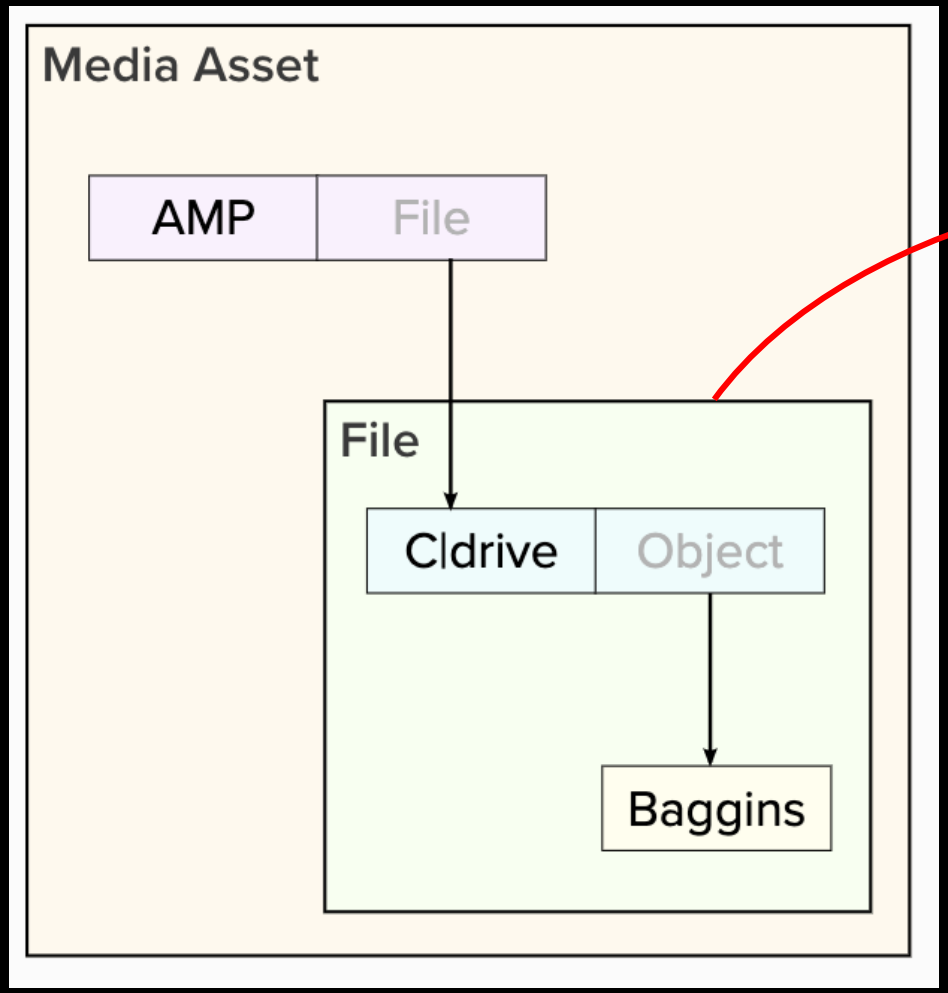
And we are hiring…

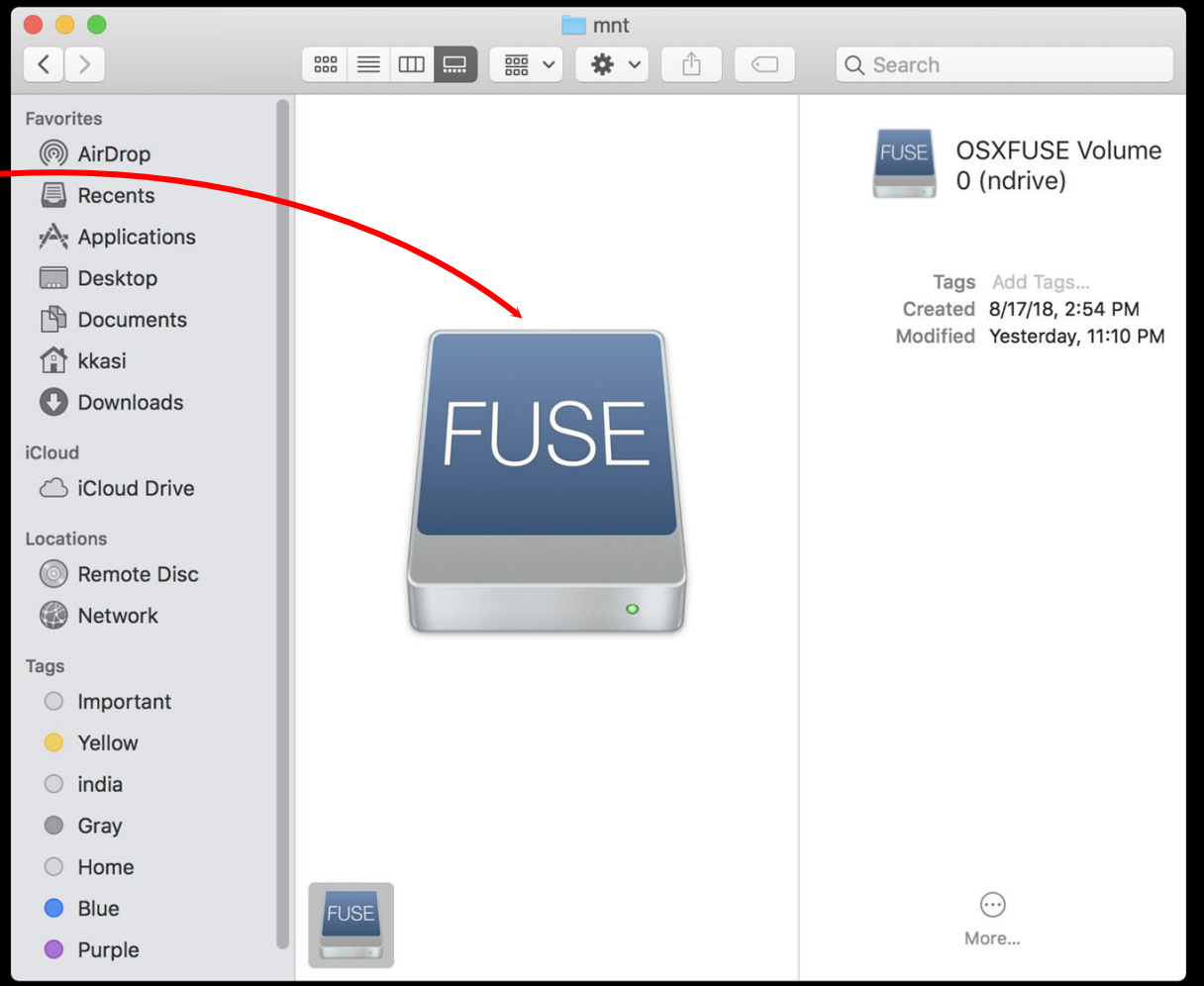
https://jobs.netflix.com/jobs/67865283

File mount

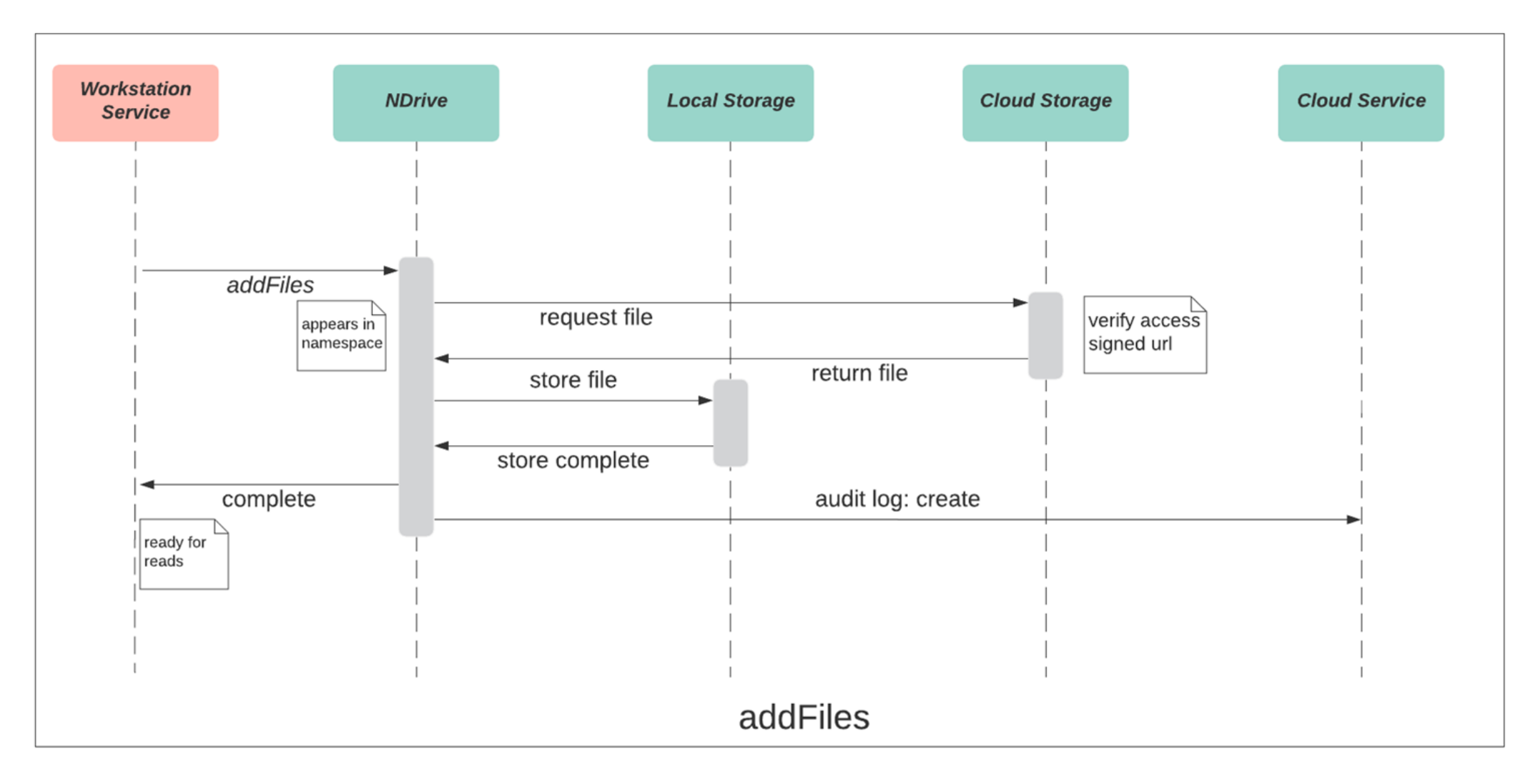

OSXFUSE Volume 0 (ndrive)
Search
Favorites
AirDrop
Recents
Applications
Desktop
Documents
kkasi
Downloads
iCloud
iCloud Drive
Locations
Remote Disc
Network
Tags
Important
Yellow
india
Gray
Home
Blue
Purple
All Tags...
Macintosh HD
Network
OSXFUSE...0 (ndrive)
Remote Disc
Applications
Library
mnt
System
Users
OSXFUSE...0 (ndrive)
Workspace

Assets and files

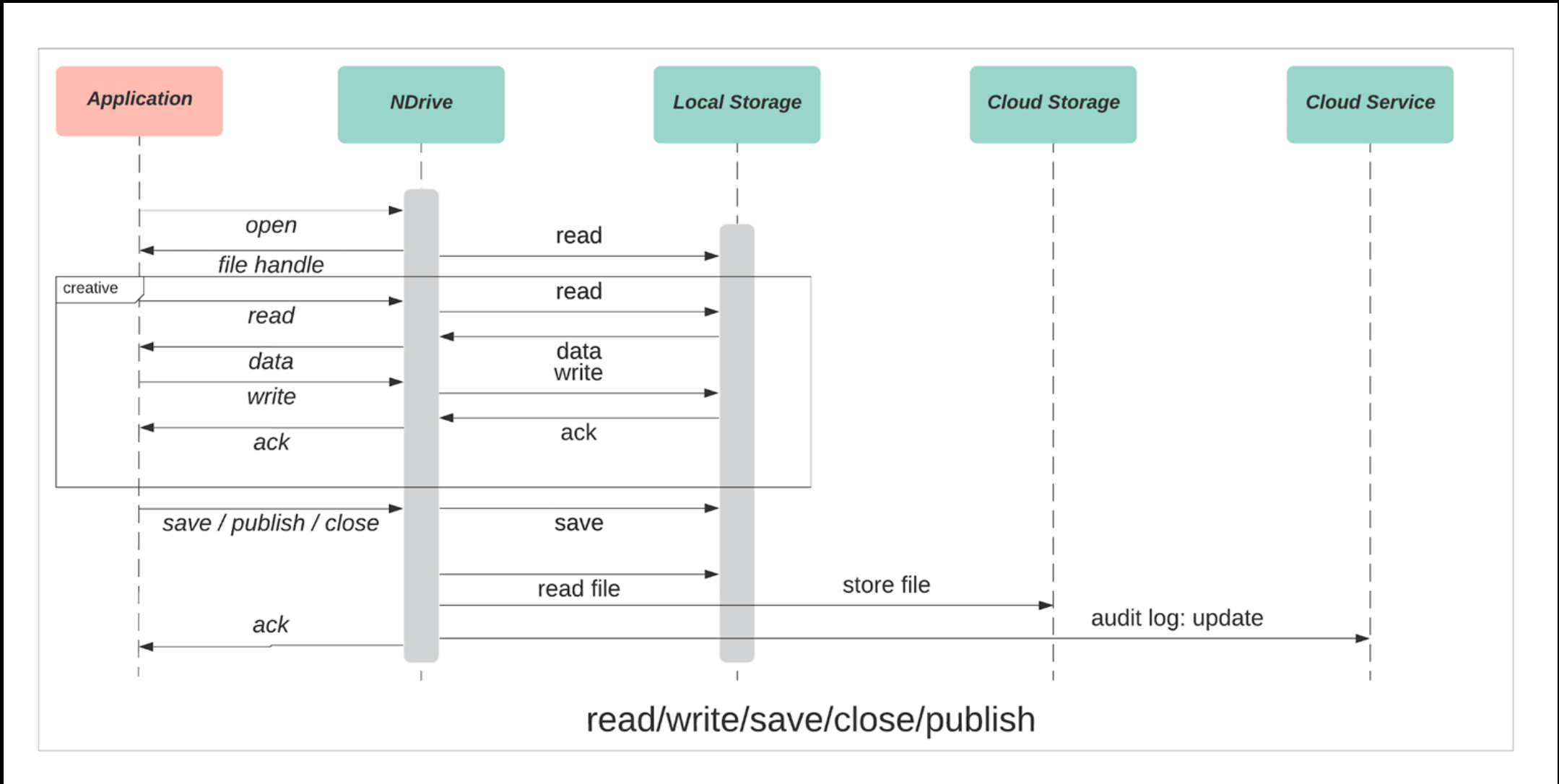




Netflix Drive Interactions



Netflix Drive Interactions



Thank You.