The Case for Dual-access File Systems over Object Storage

Kunal Lillaney, Vasily Tarasov, David Pease, Randal Burns

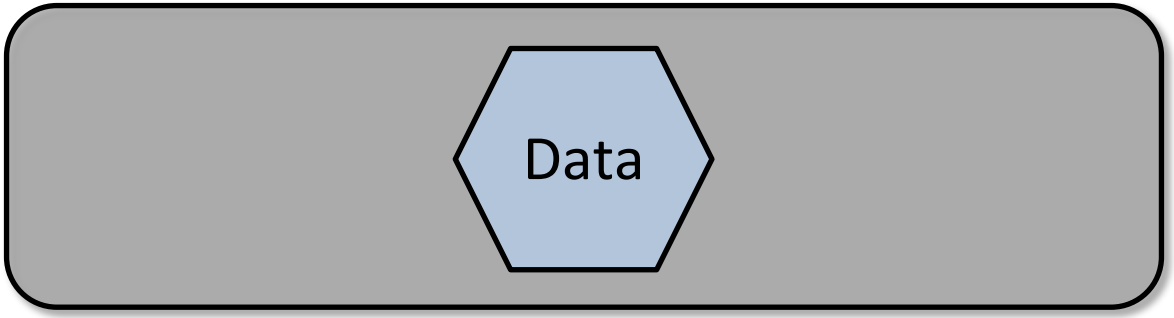
JOHNS HOPKINS UNIVERSITY

IBM Research

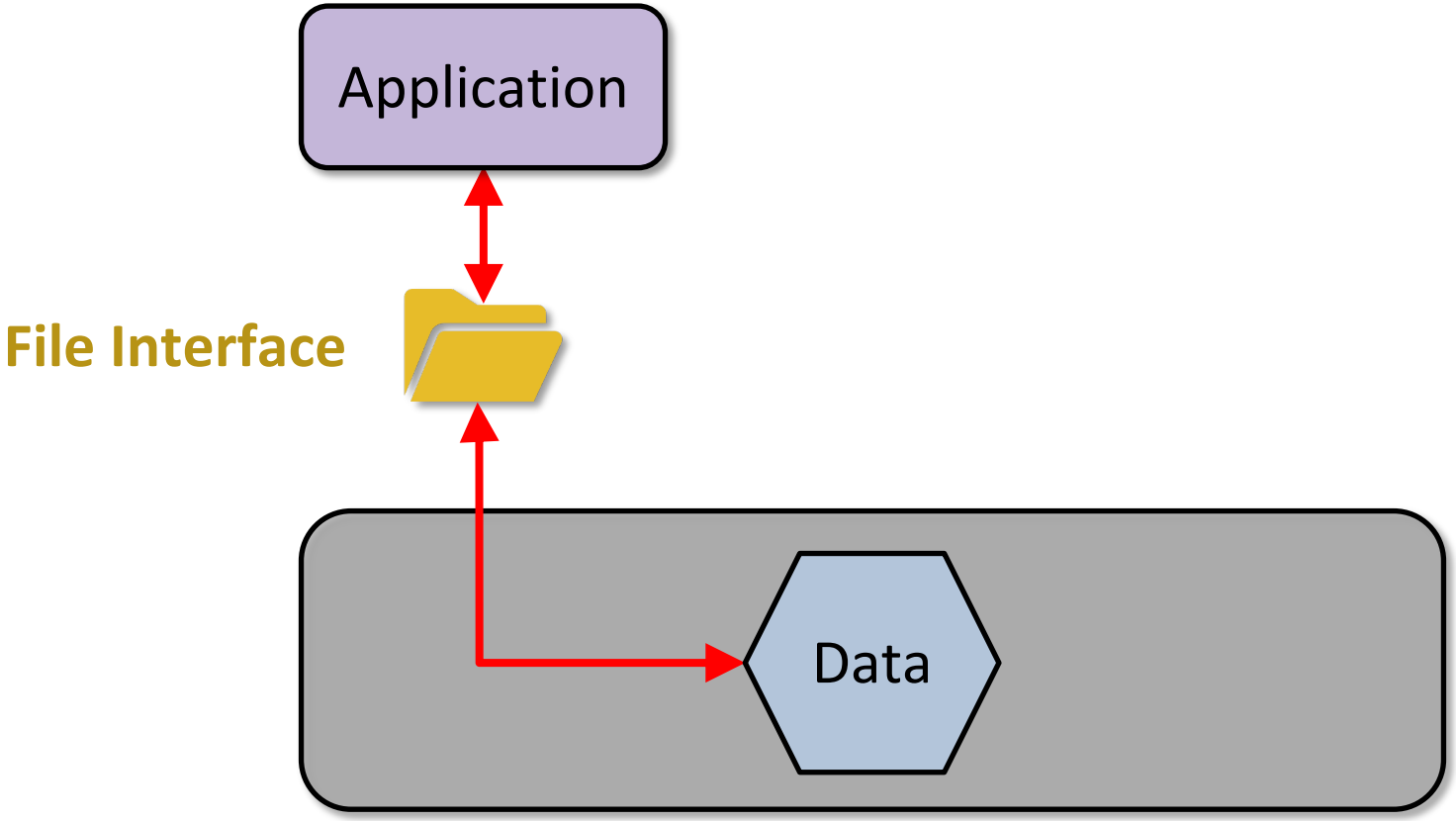
9 July, HotStorage'19 - Renton

What is Dual-Access?

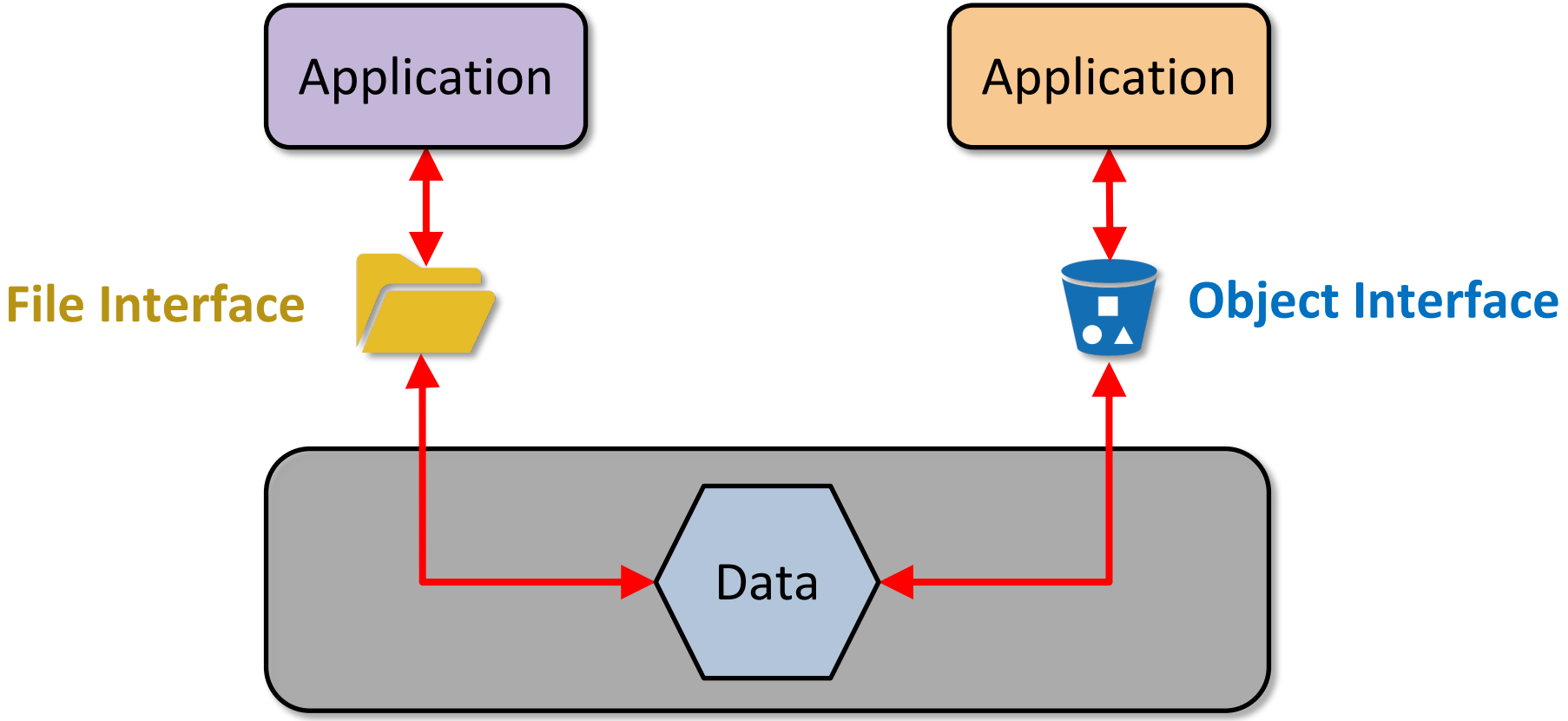
Dual Access



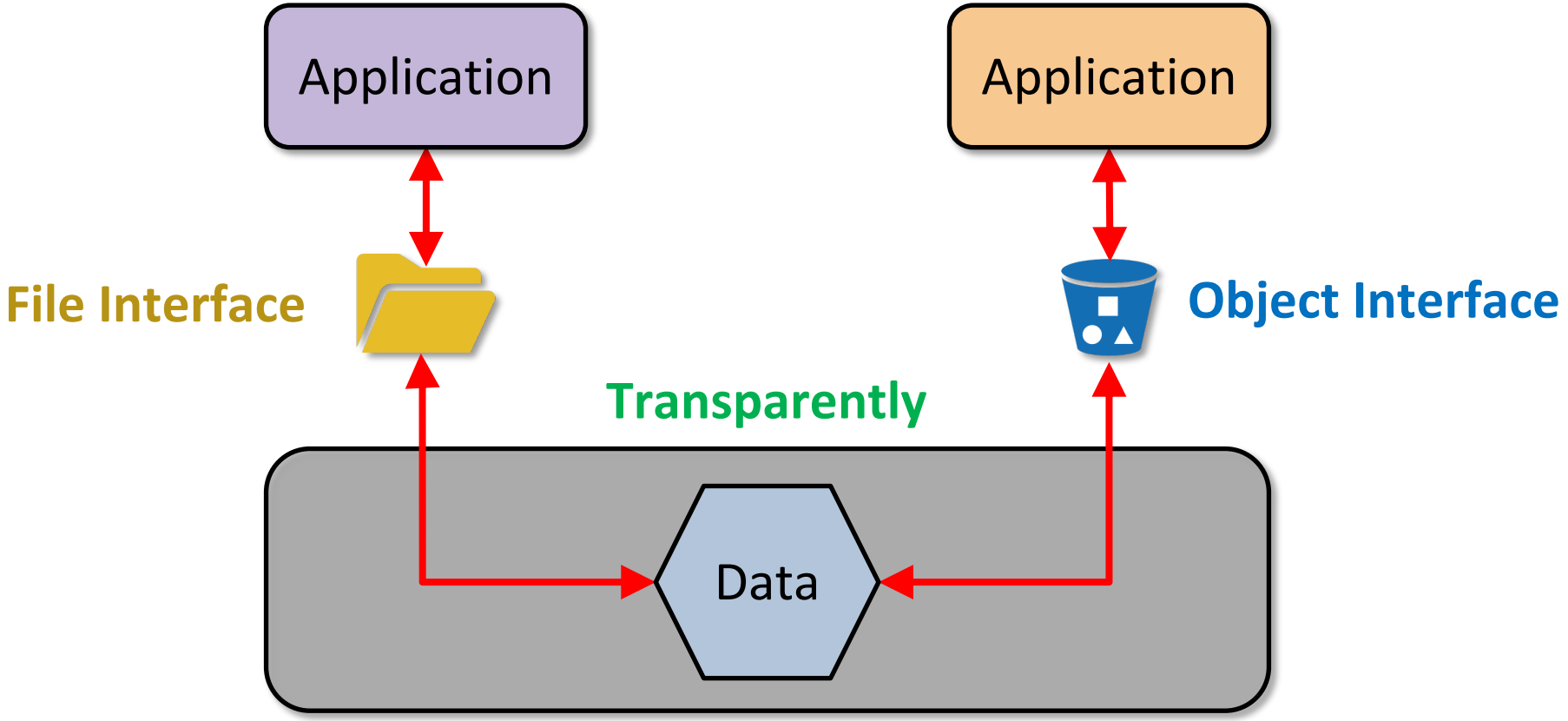
Dual Access



Dual Access


Application
Application
File Interface
Object Interface
Data

Dual Access

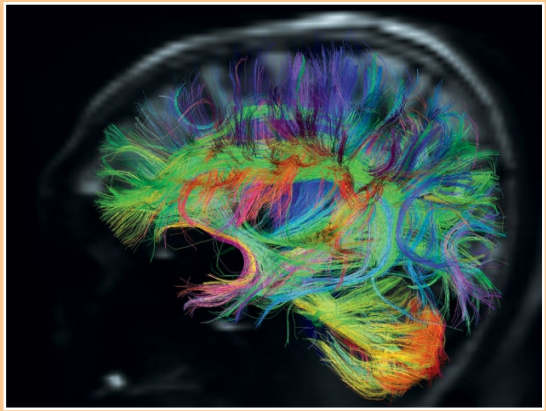

Application
Application
File Interface
Object Interface
Transparently
Data

Media

Use Cases

Life Science

Neuroscience



Geo-Informatics

Media Transcoding, Editing, Analytics

Media Transcoding, Editing, Analytics

Object Interface

Media Transcoding, Editing, Analytics

pinnacle

COREL™

FFmpeg

Ps Adobe Photoshop

File Interface

Object Interface

Media Transcoding, Editing, Analytics

pinnacle

COREL™

FFmpeg

Ps Adobe Photoshop

File Interface

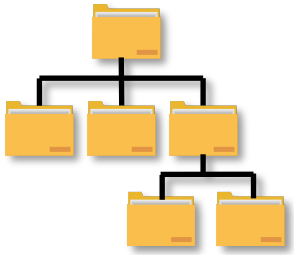
Object Interface

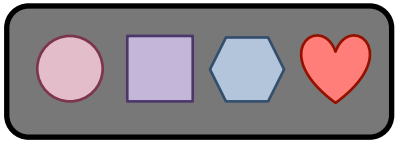
Object Interface

File Systems Vs Object Storage

Partial Writes

Namespace





Interfaces

Outline

- Design considerations
- Existing systems
- Agni
- Future work

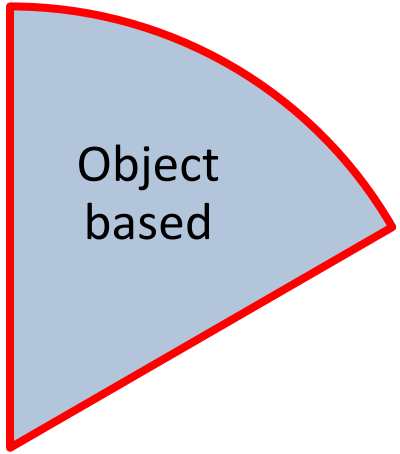
https://www.greenbiz.com/sites/default/files/styles/gbz_article_primary_breakpoints_kalapicture_screenmd_1x/public/images/articles/featured/datacenter_0.jpg?itok=iJm7ezgB&timestamp=1483504030

Outline

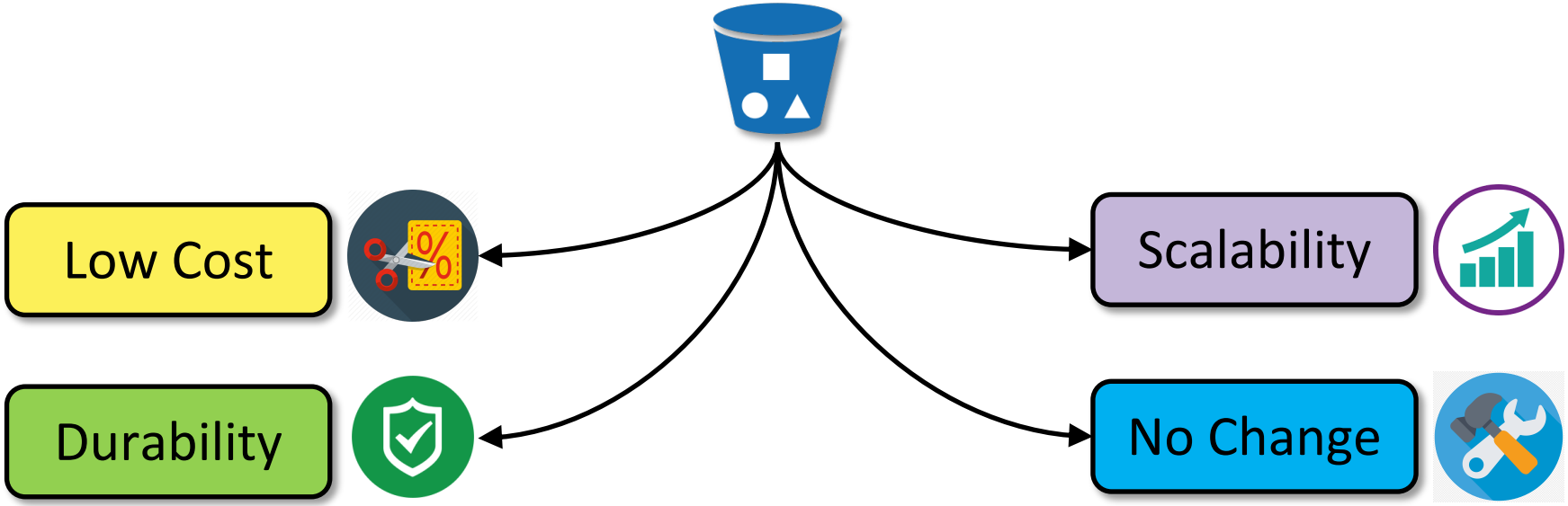
- Design considerations
- Existing systems
- Agni
- Future work

https://www.greenbiz.com/sites/default/files/styles/gbz_article_primary_breakpoints_kalapicture_screenmd_1x/public/images/articles/featured/datacenter_0.jpg?itok=iJm7ezgB&timestamp=1483504030

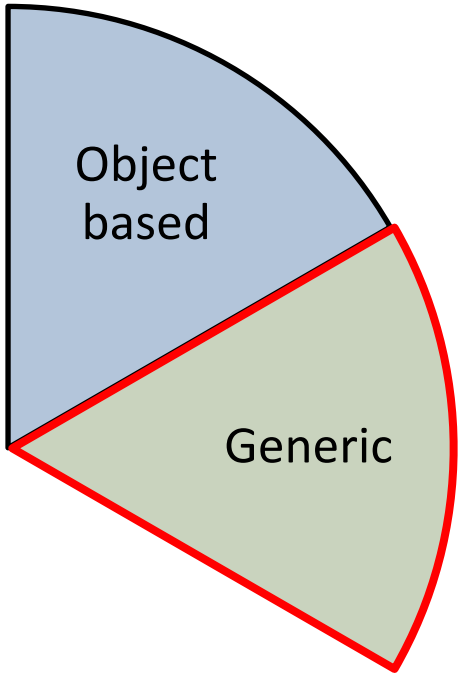
Design Considerations



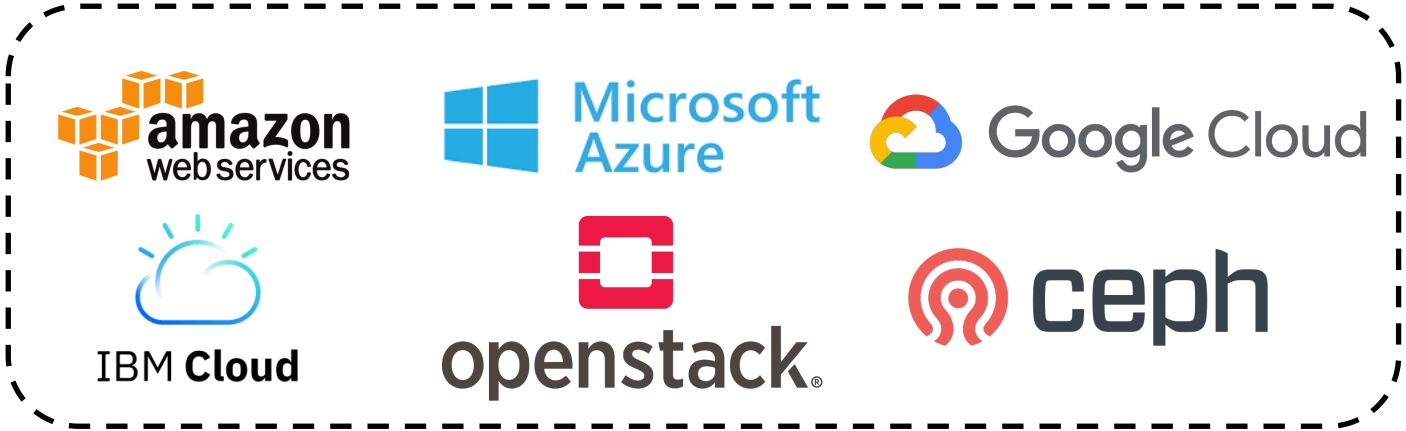
Value Proposition for Object Storage



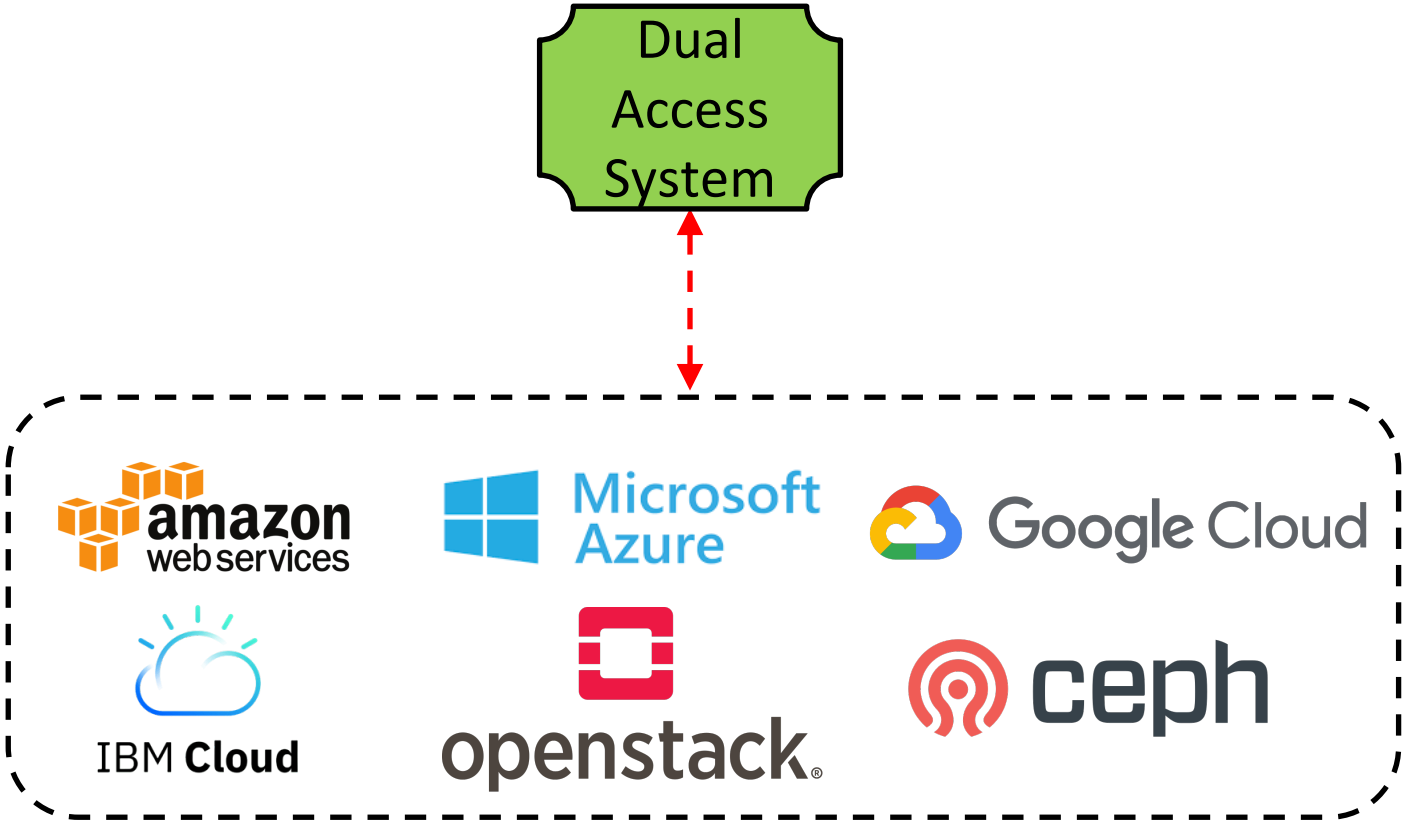
Design Considerations



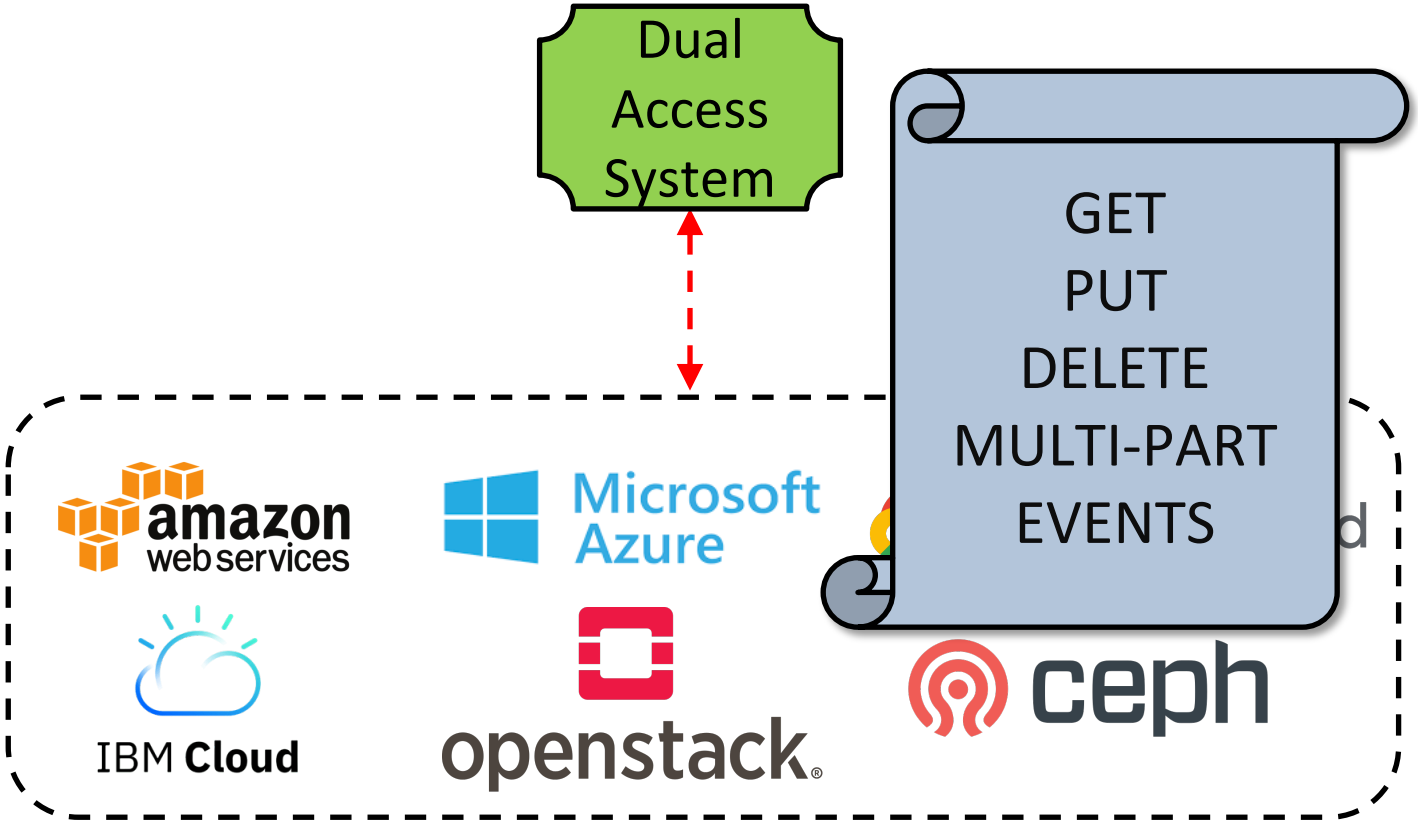
Generic → Object Store Agnostic



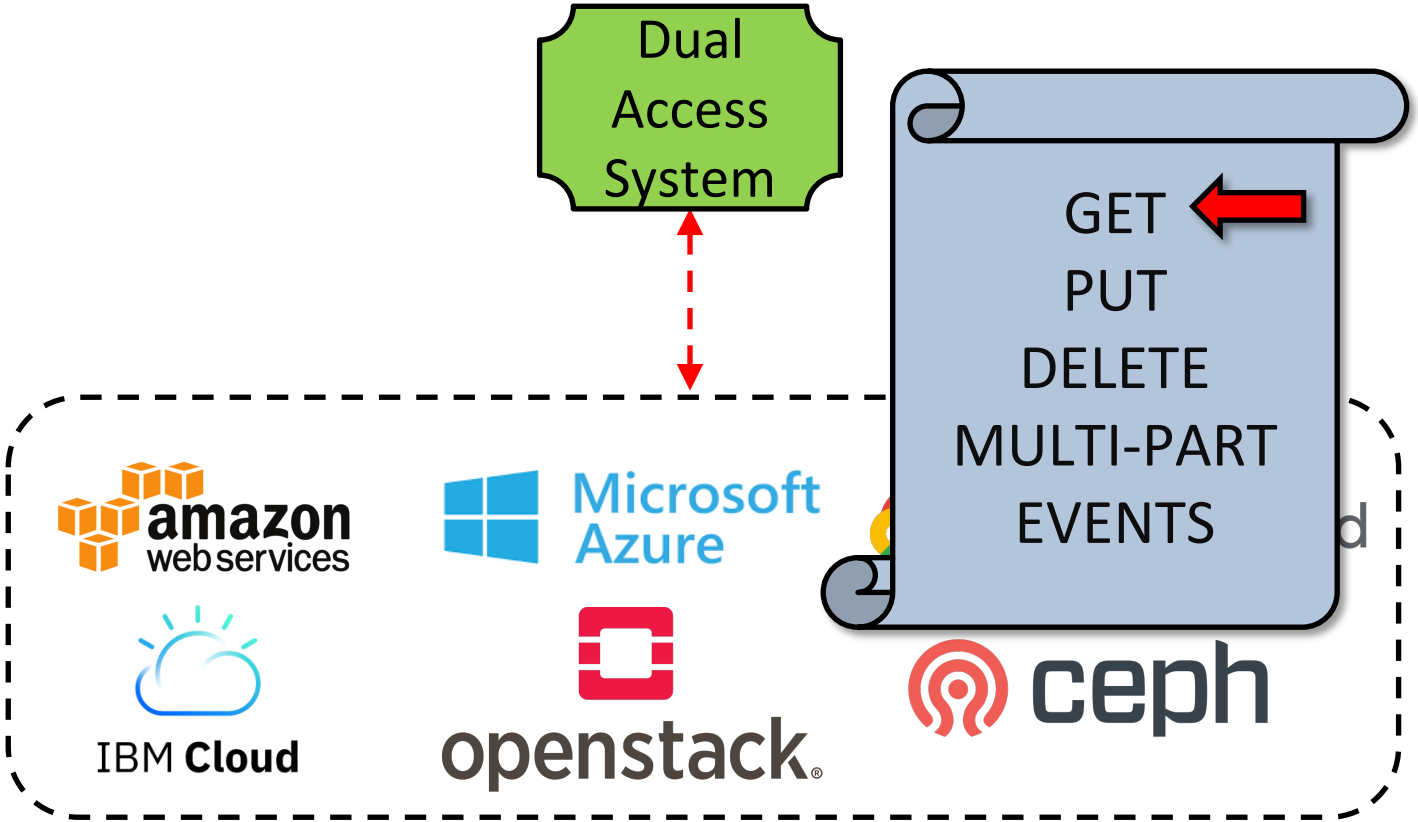
Generic → Object Store Agnostic



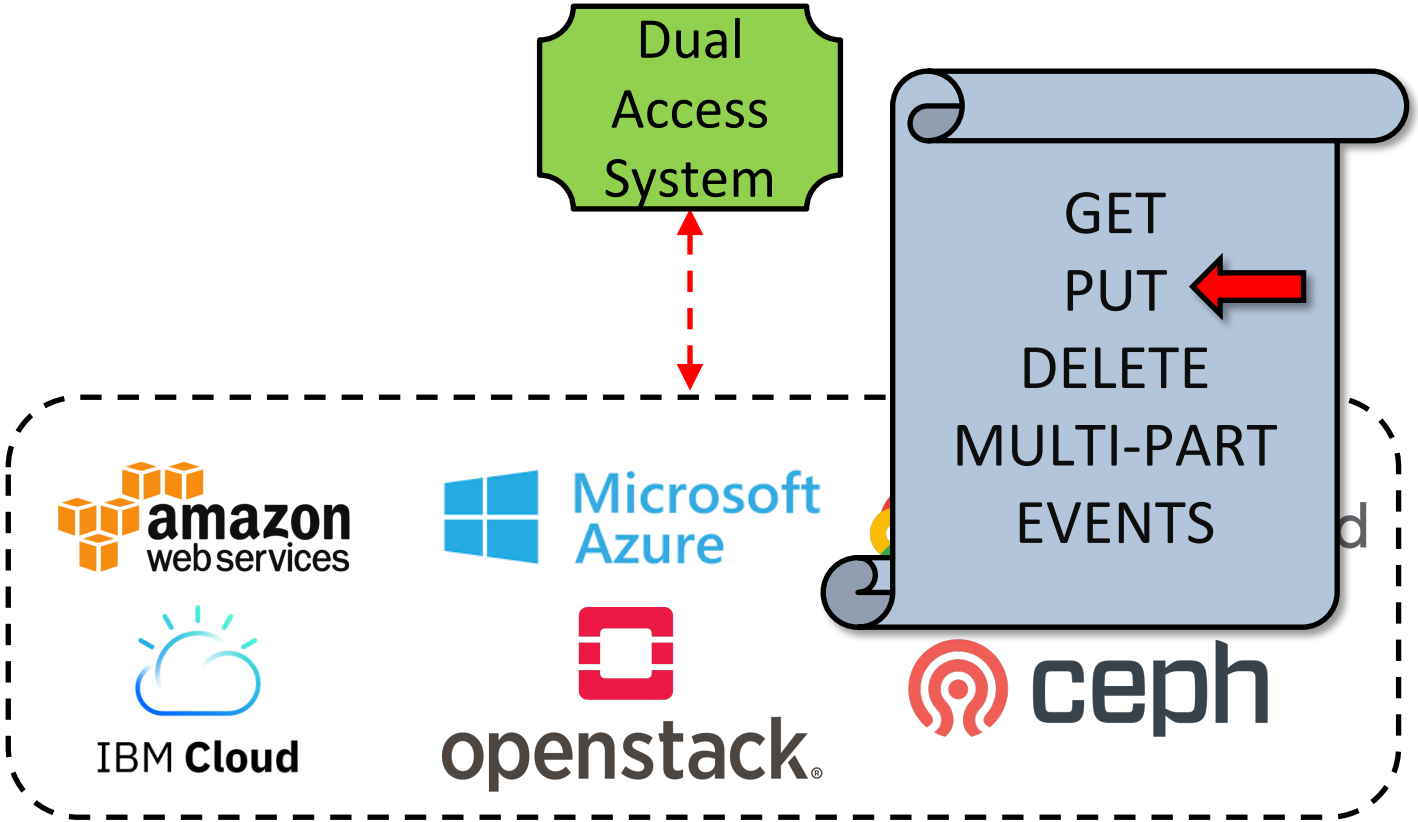
Generic → Object Store Agnostic



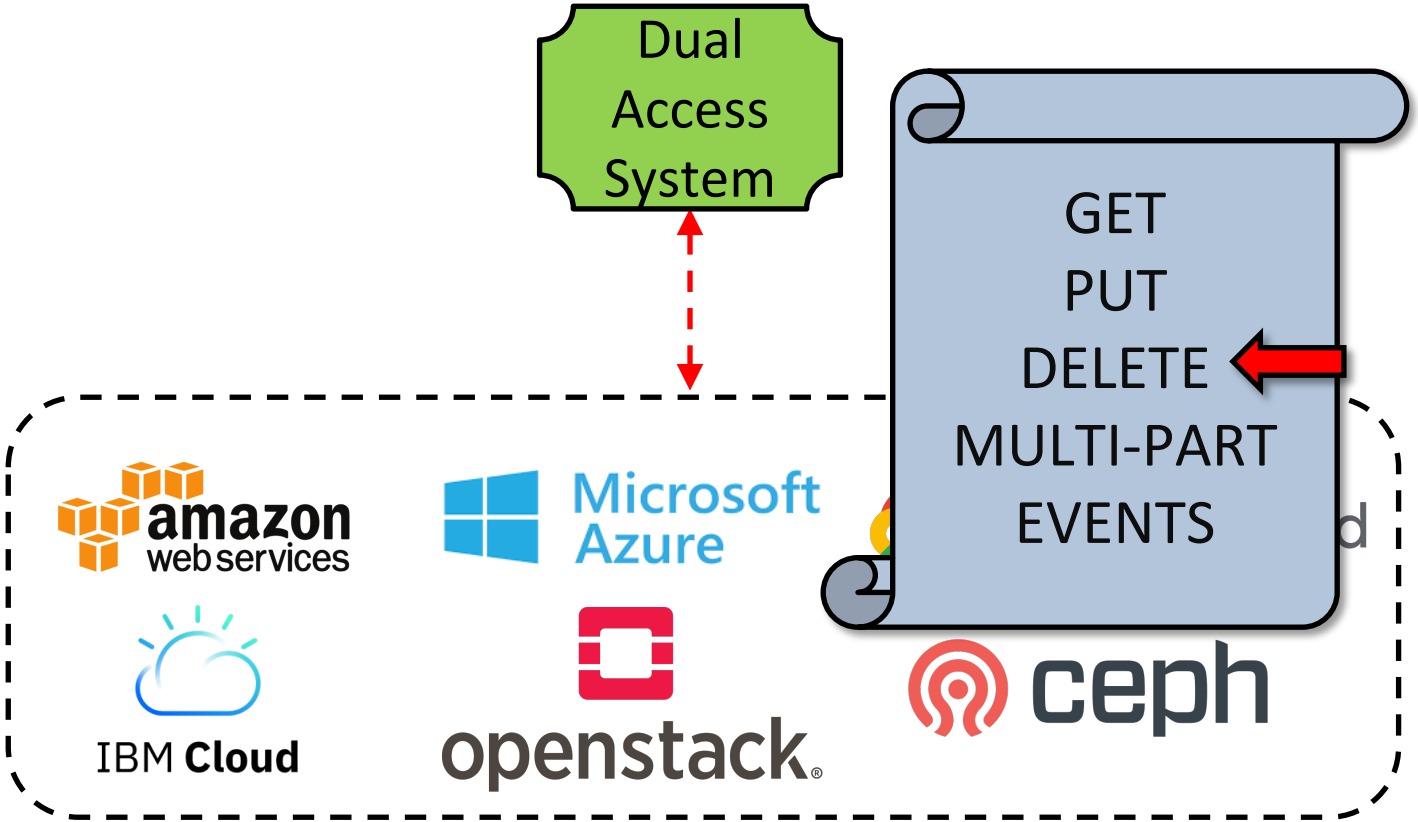
Generic → Object Store Agnostic



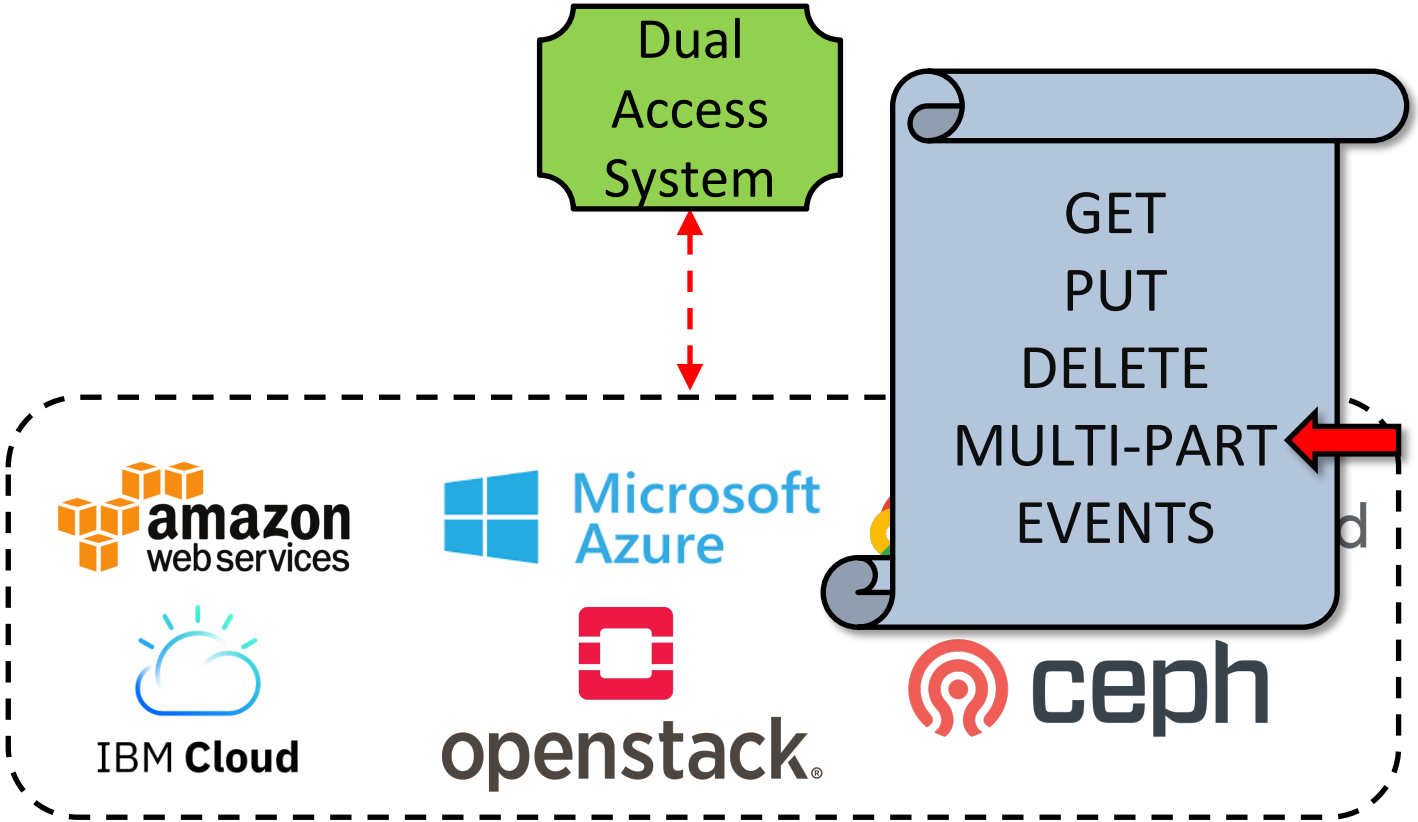
Generic → Object Store Agnostic



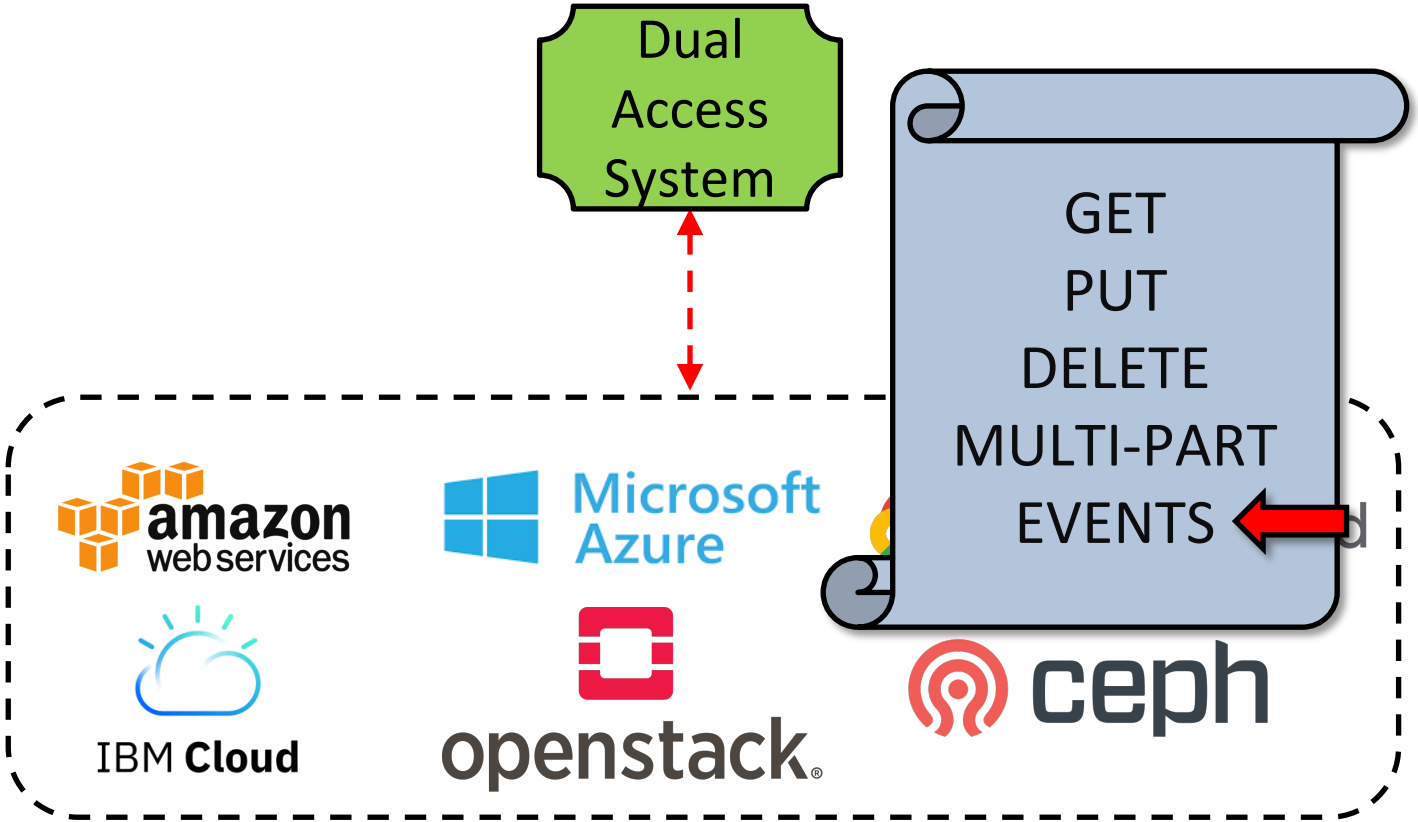
Generic → Object Store Agnostic



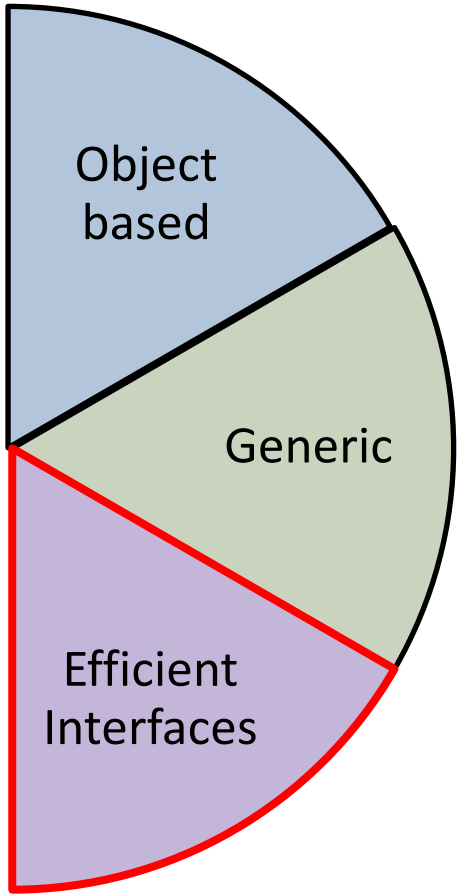
Generic → Object Store Agnostic



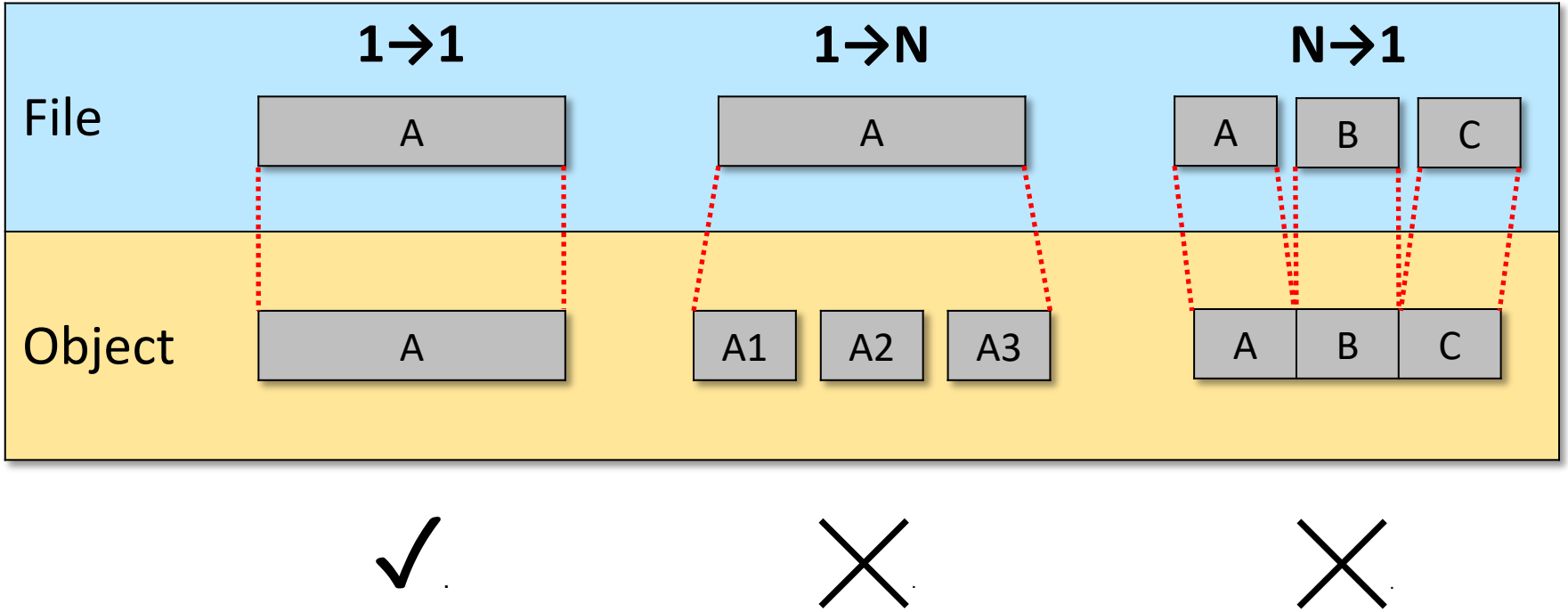
Generic → Object Store Agnostic



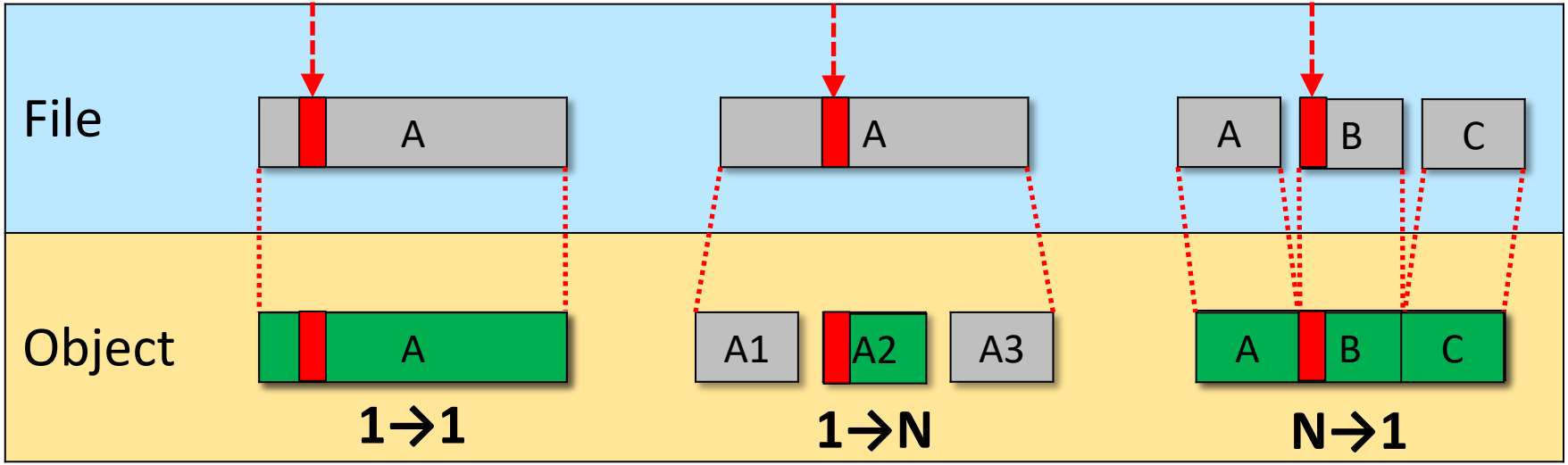
Design Considerations



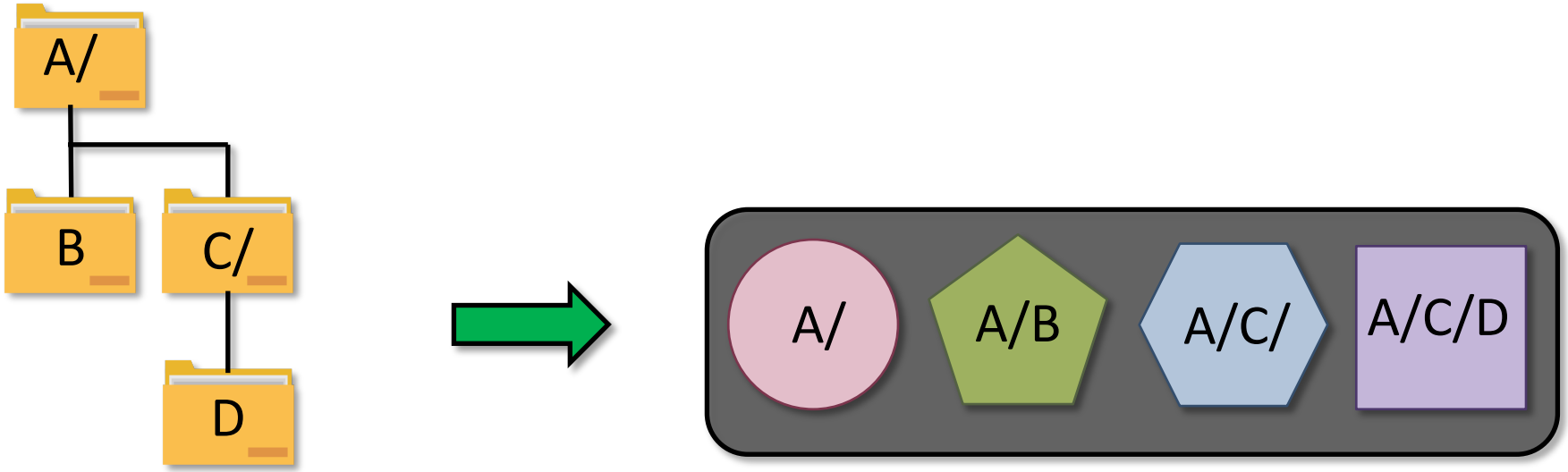
Dual Access: File to Object Mapping



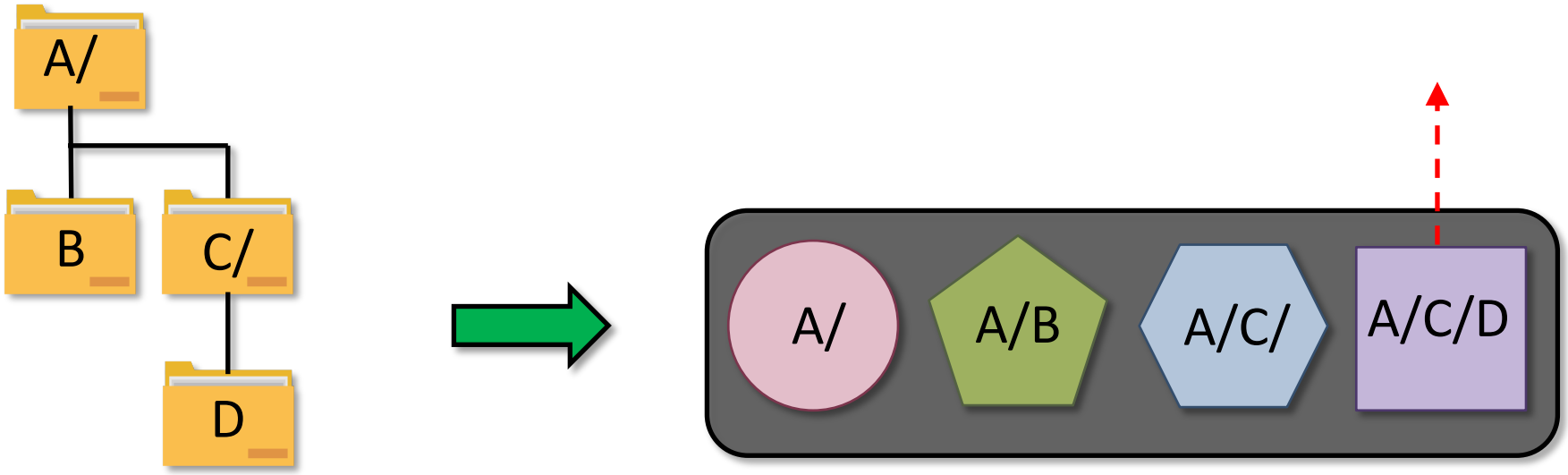
Efficiency: File to Object Mapping



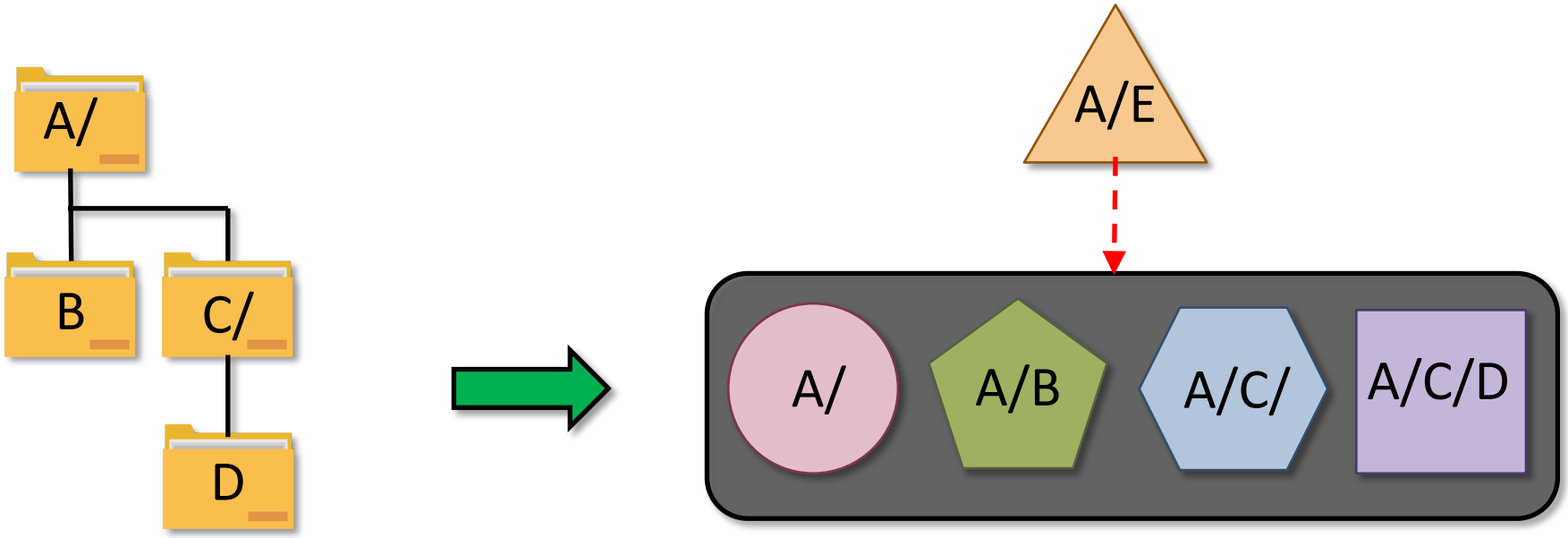
Dual Access: Object Naming



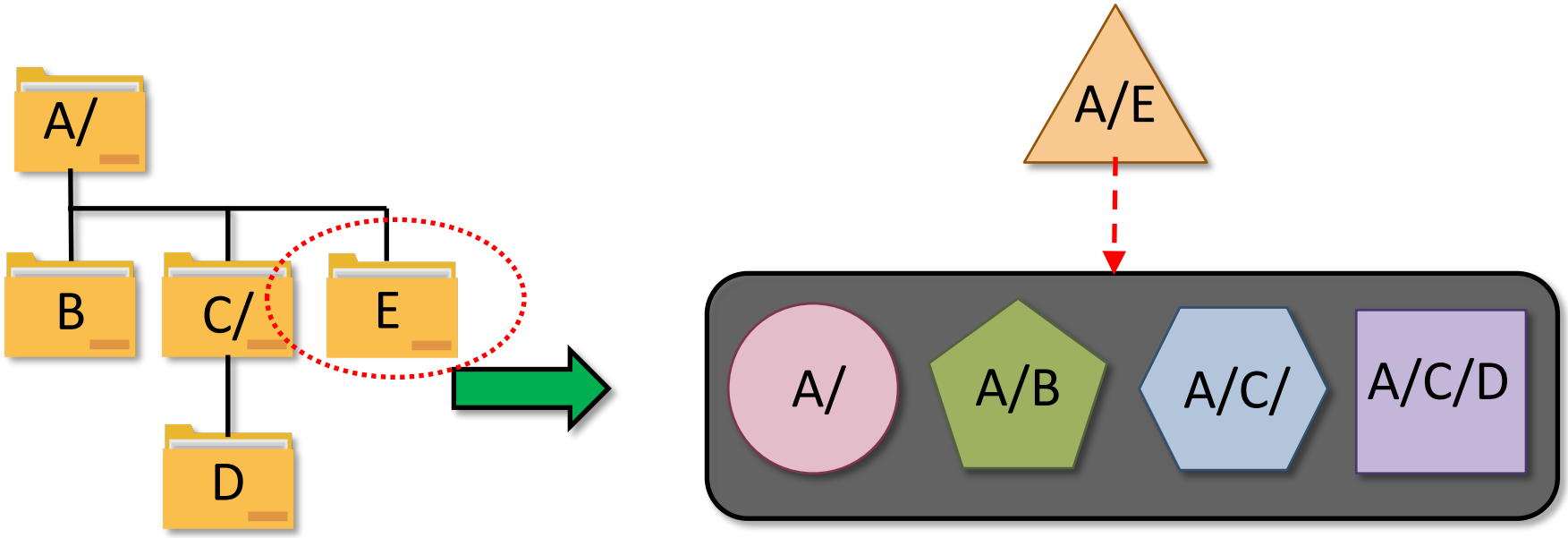
Dual Access: Object Naming



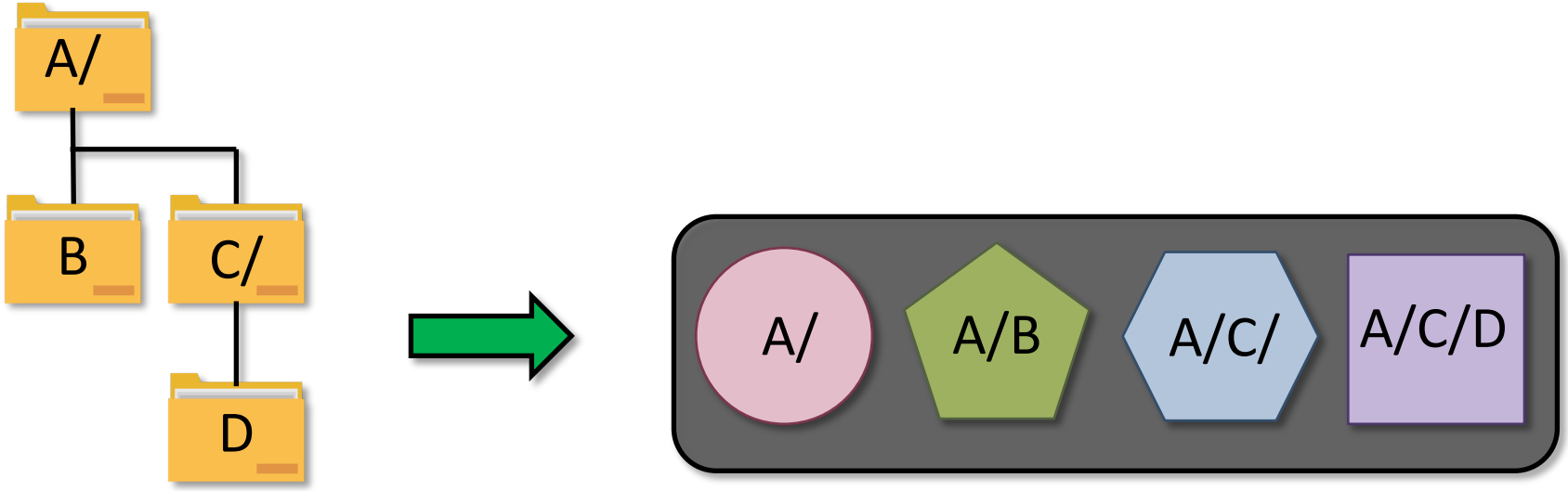
Dual Access: Object Naming



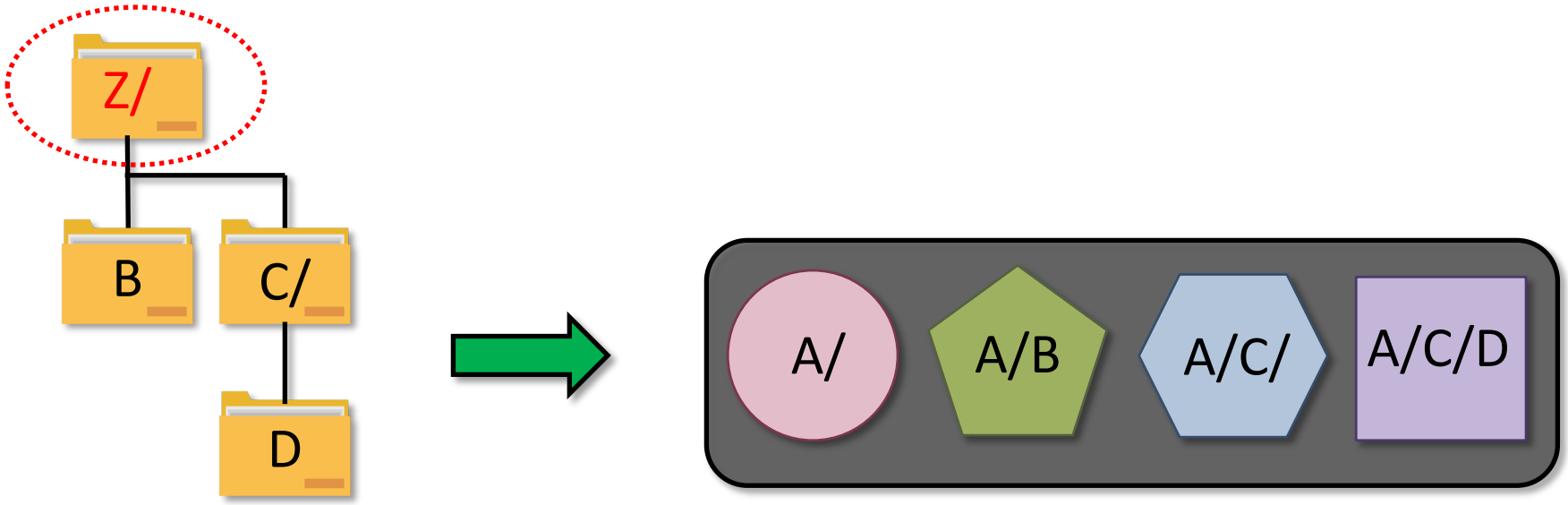
Dual Access: Object Naming



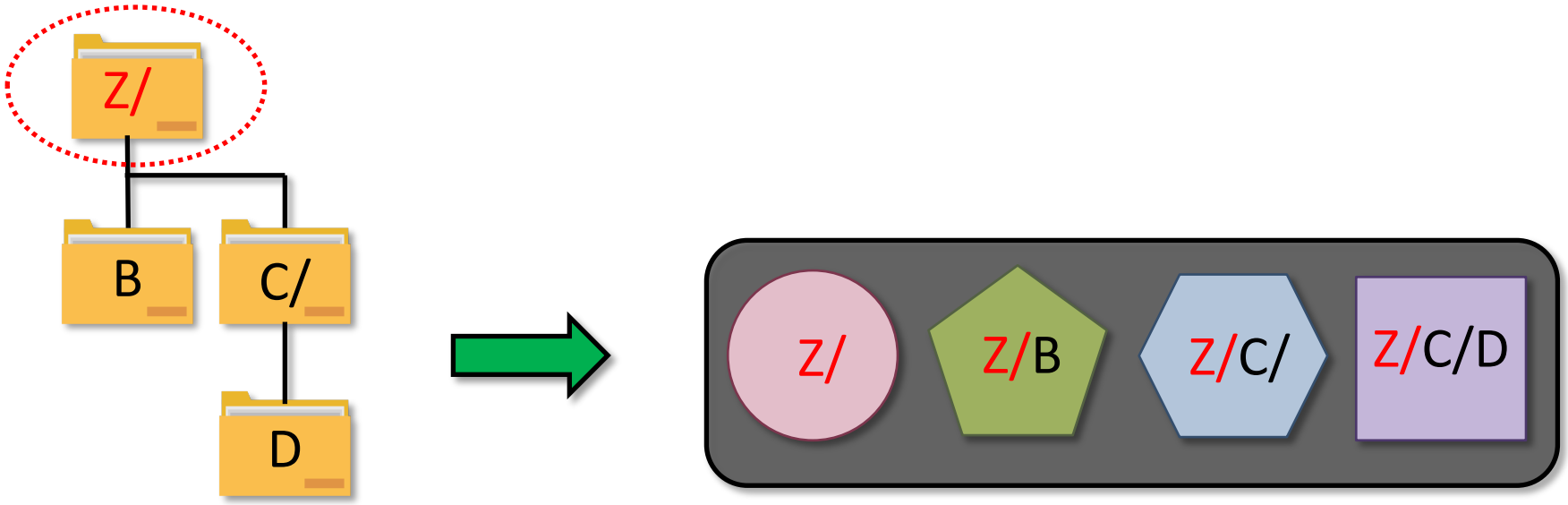
Efficiency: Object Naming



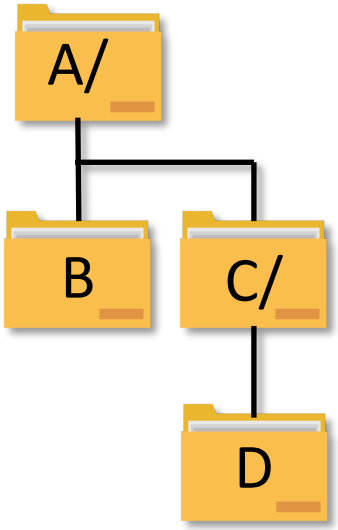
Efficiency: Object Naming



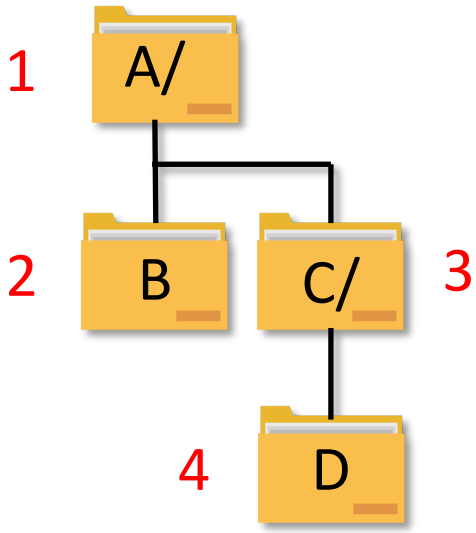
Efficiency: Object Naming



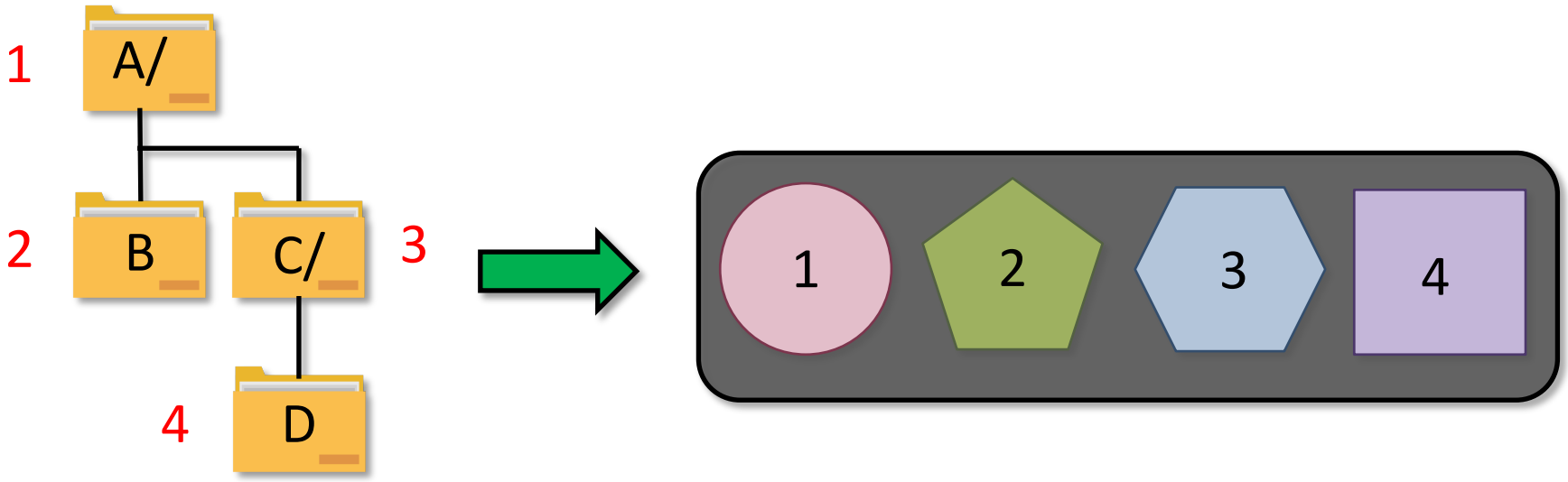
Object Naming: Inode Number



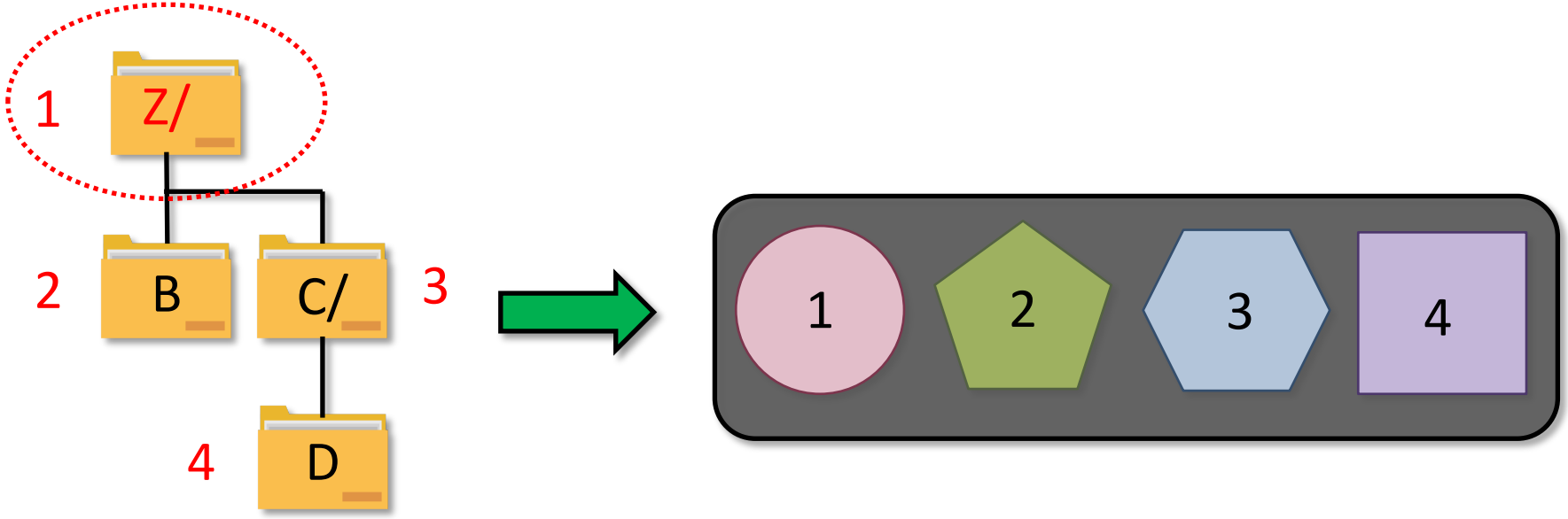
Object Naming: Inode Number



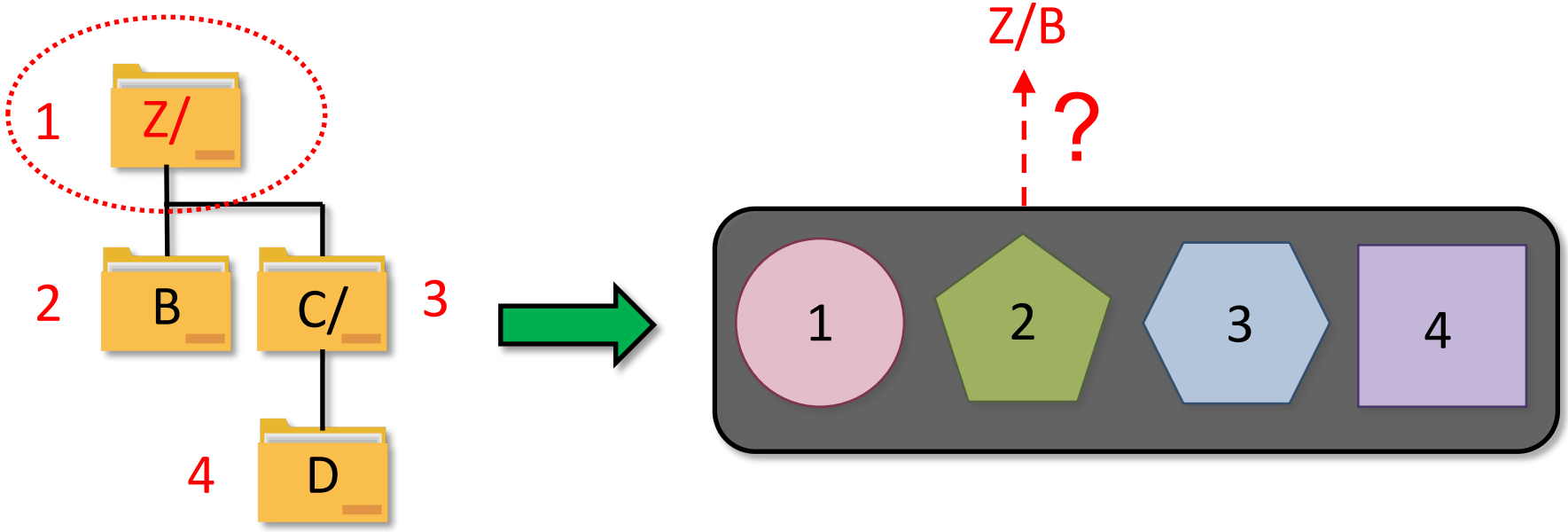
Object Naming: Inode Number



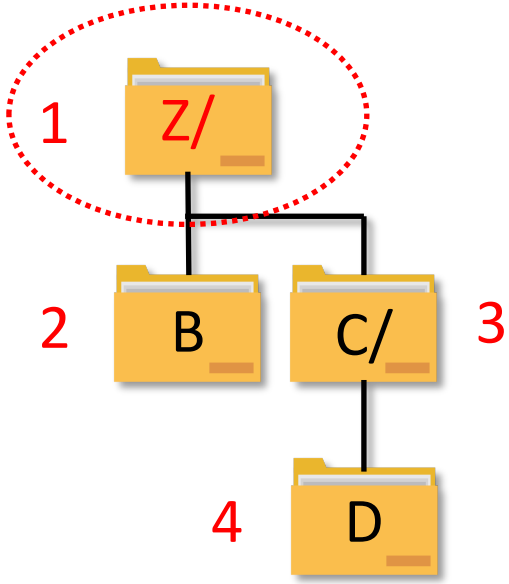
Object Naming: Inode Number



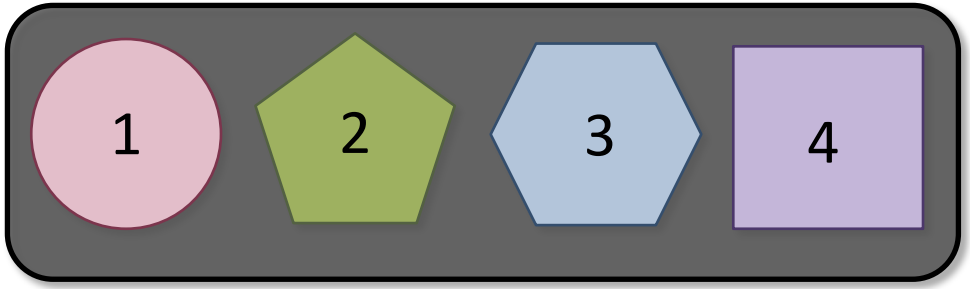
Object Naming: Inode Number



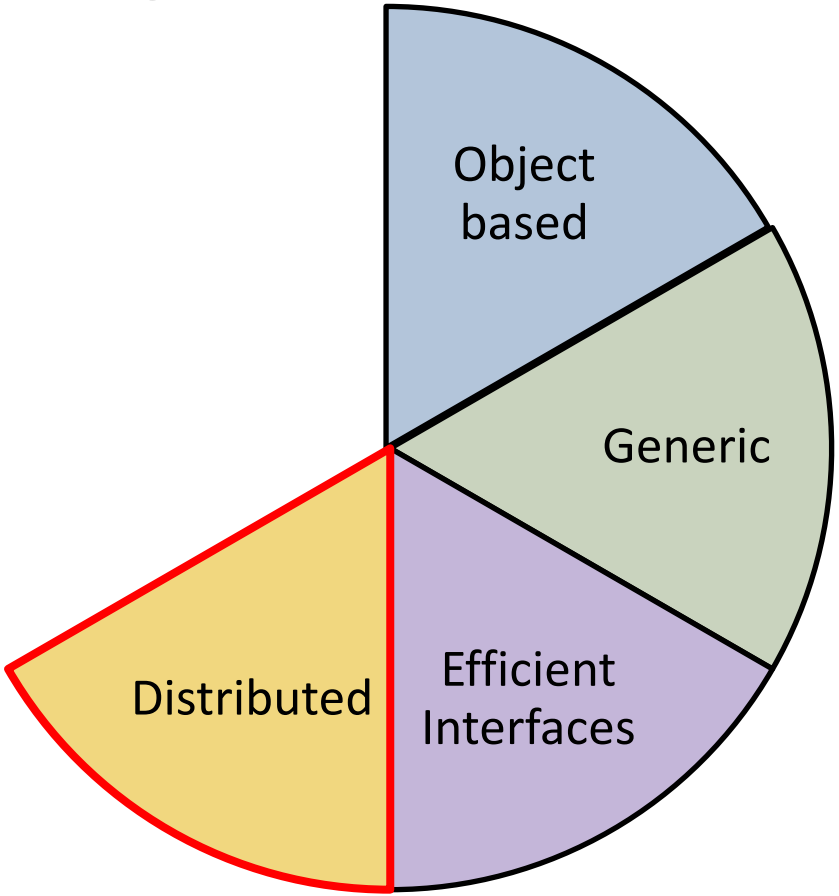
Object Naming: Inode Number



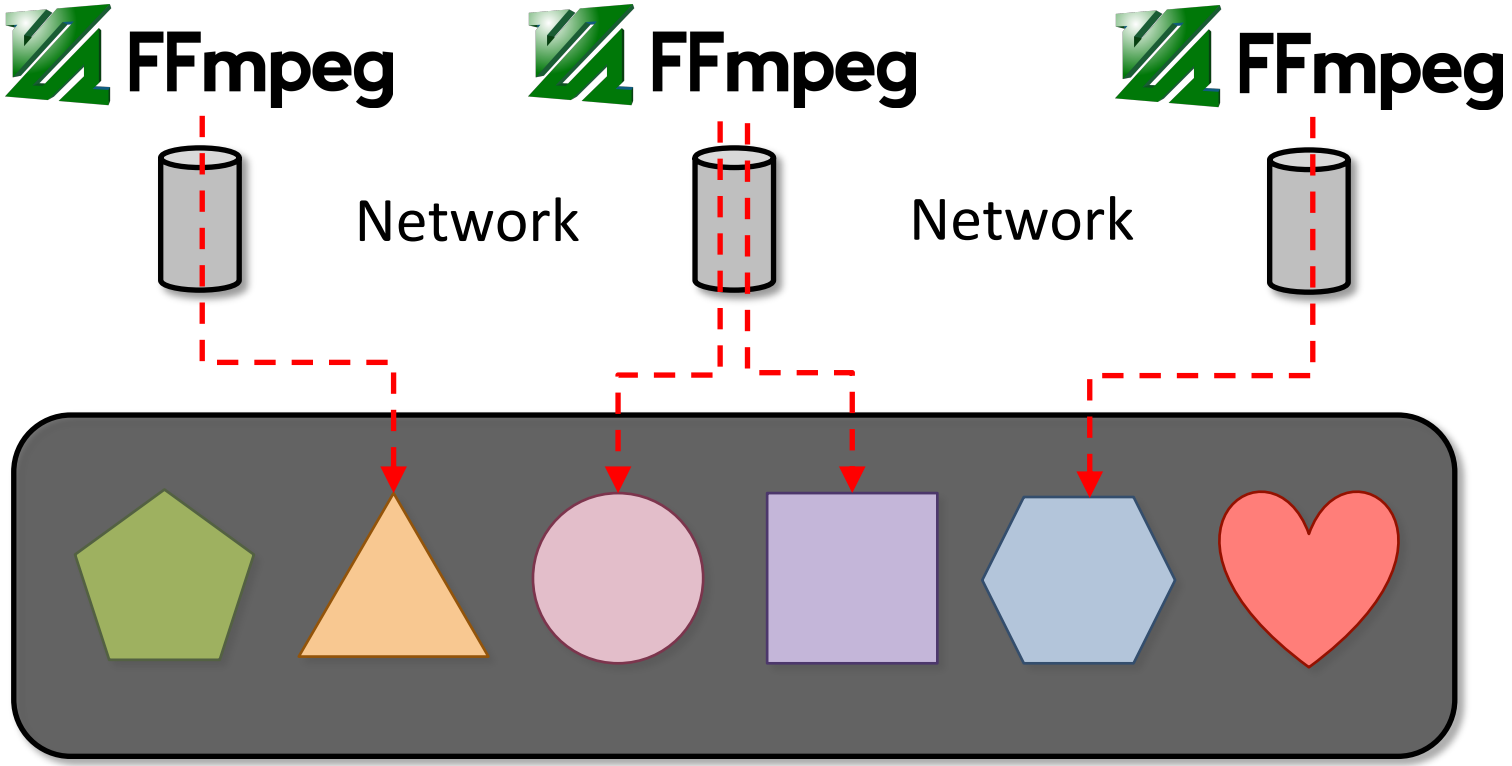
Name	Inode
Z/B	2
…	…



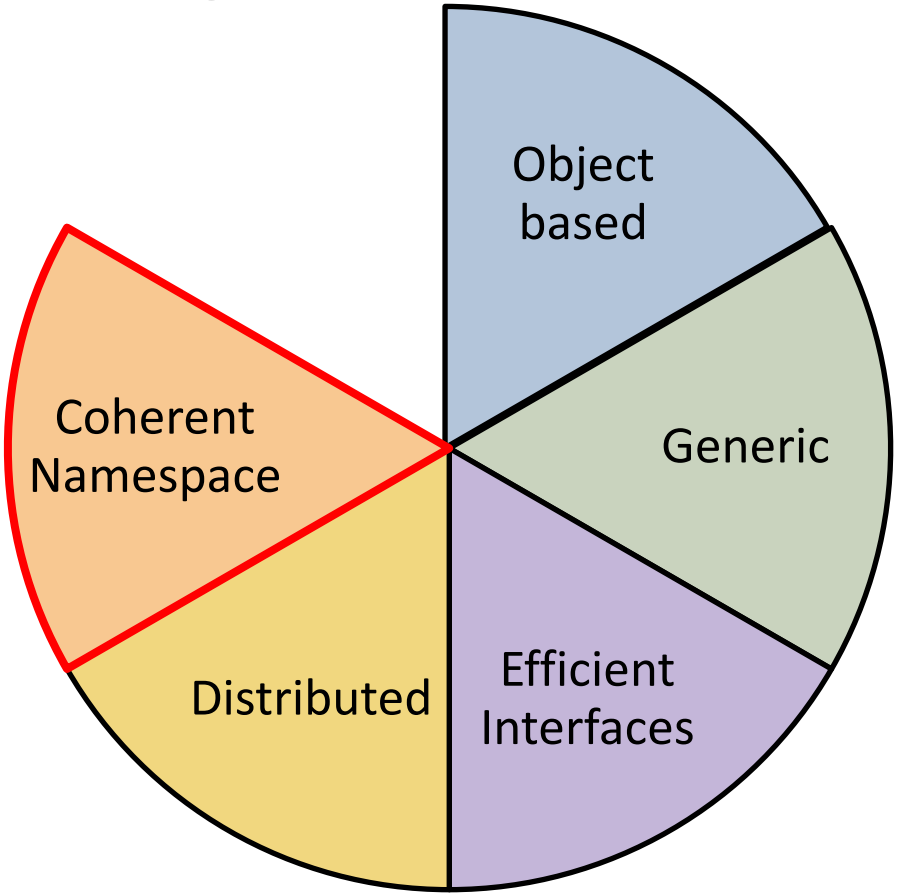
Design Considerations



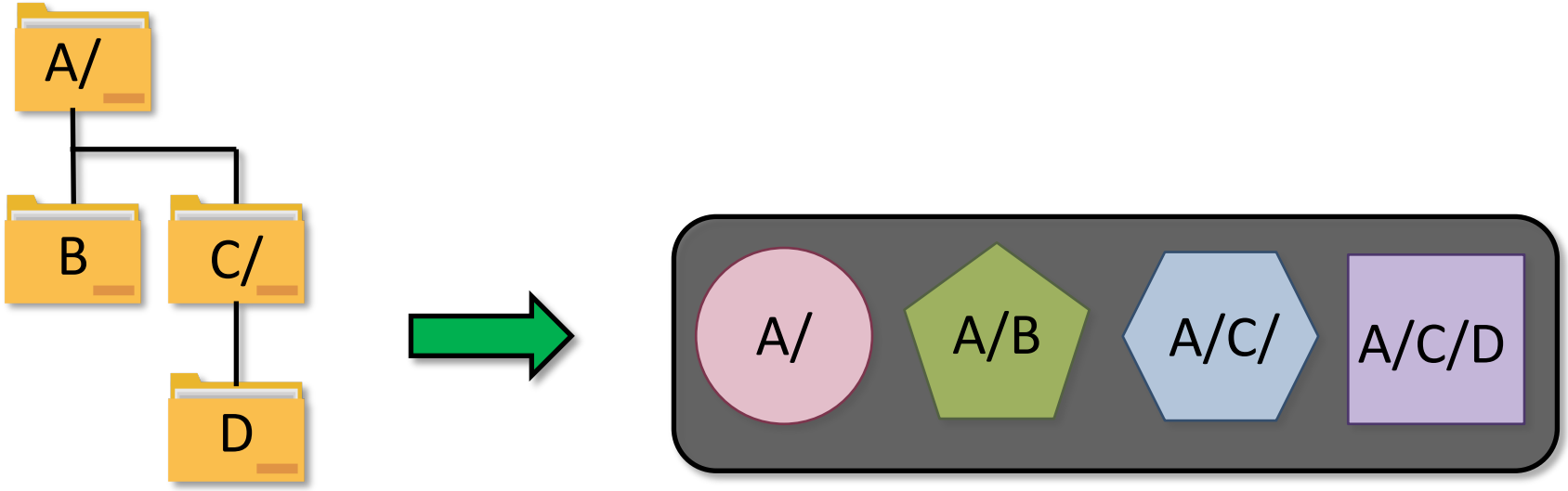
Multiple File System Mounts



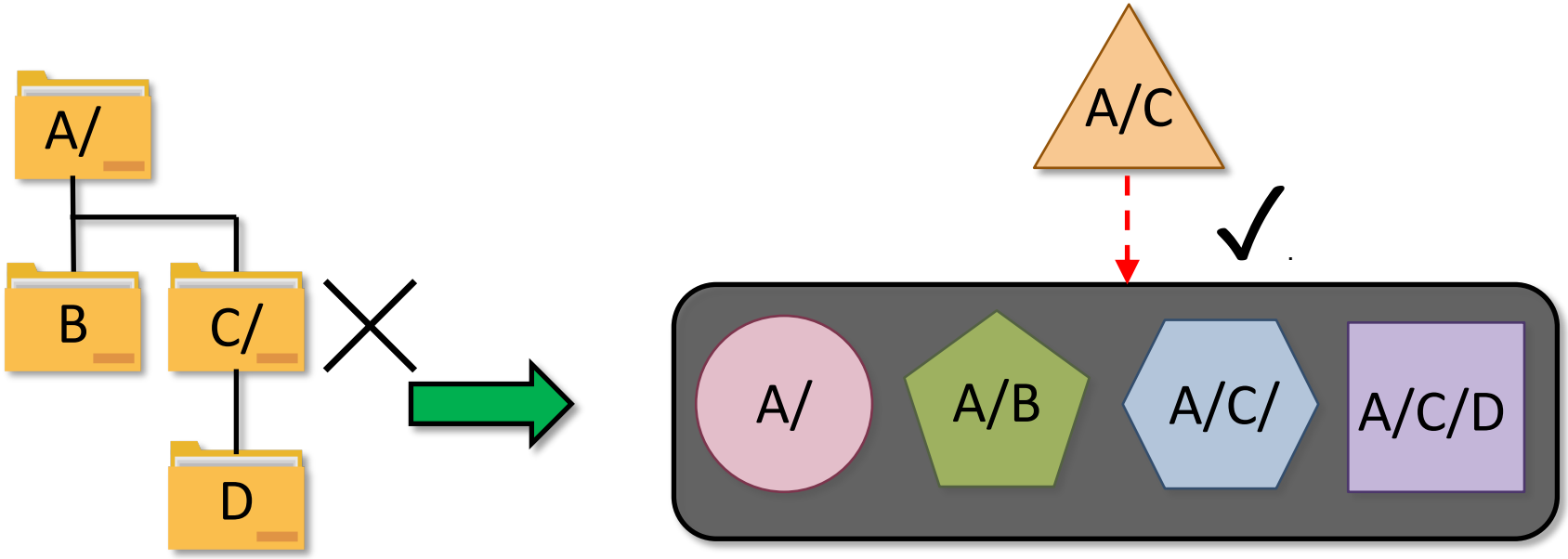
Design Considerations



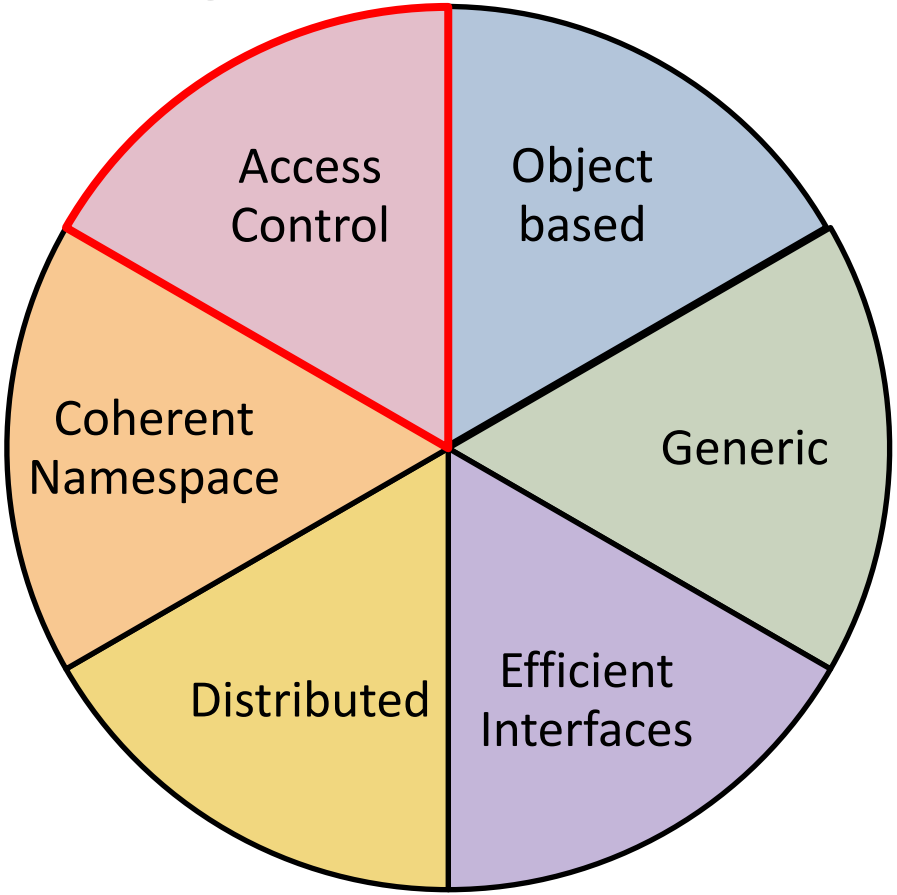
Namespace: Example of Incoherency



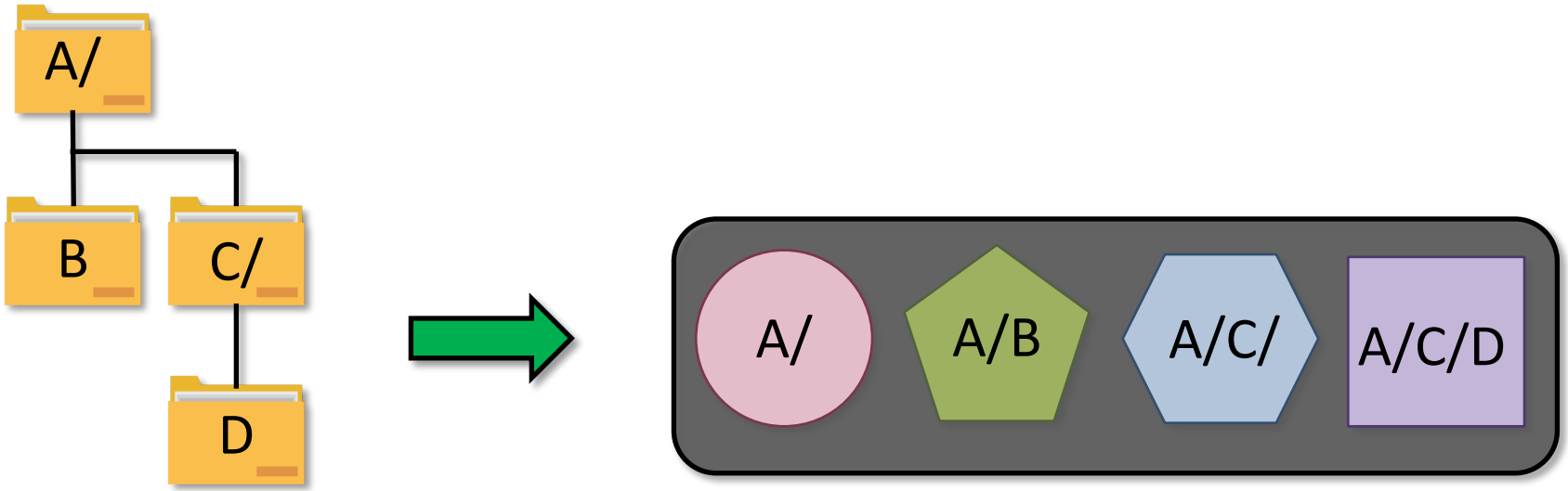
Namespace: Example of Incoherency



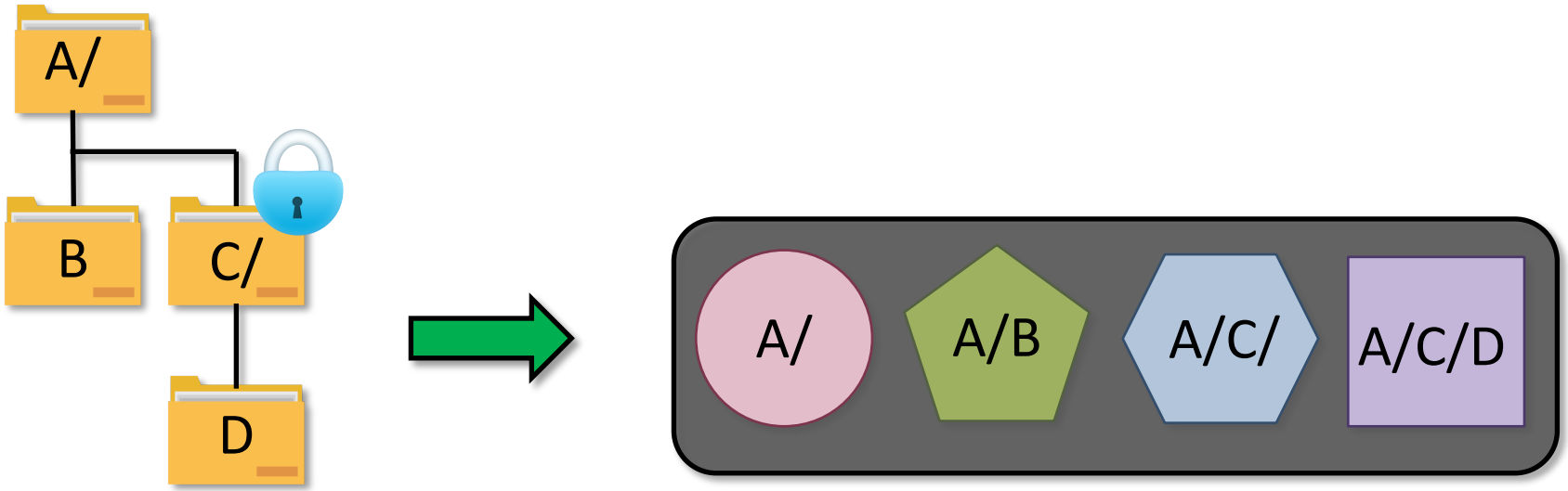
Design Considerations



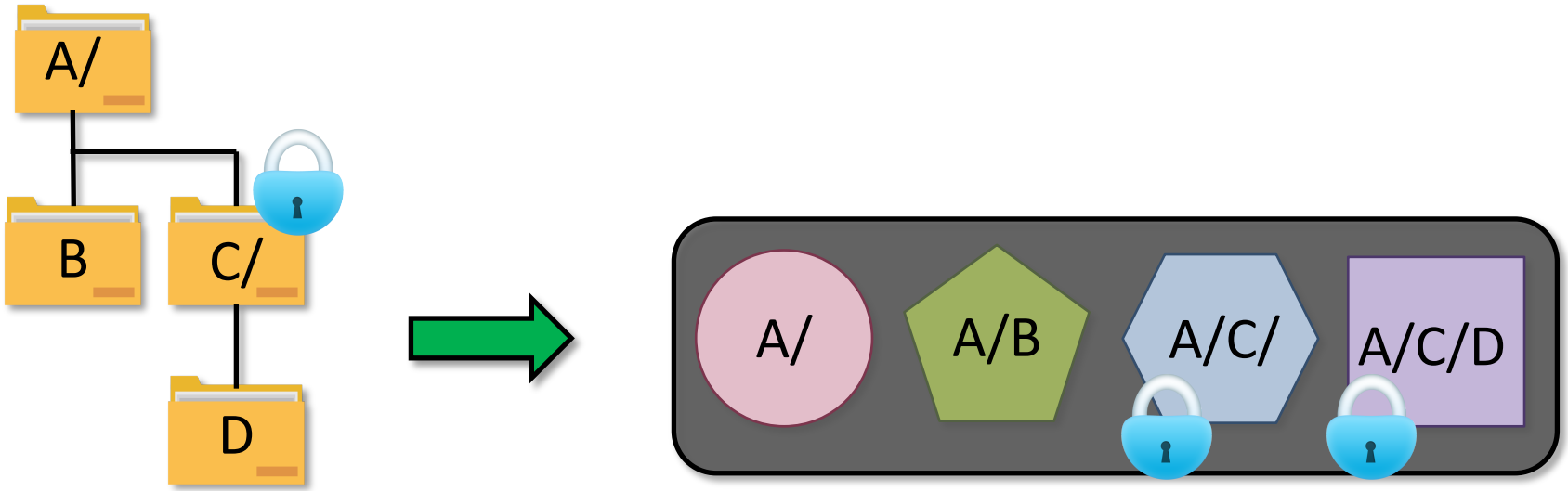
Unified Access Control



Unified Access Control



Unified Access Control



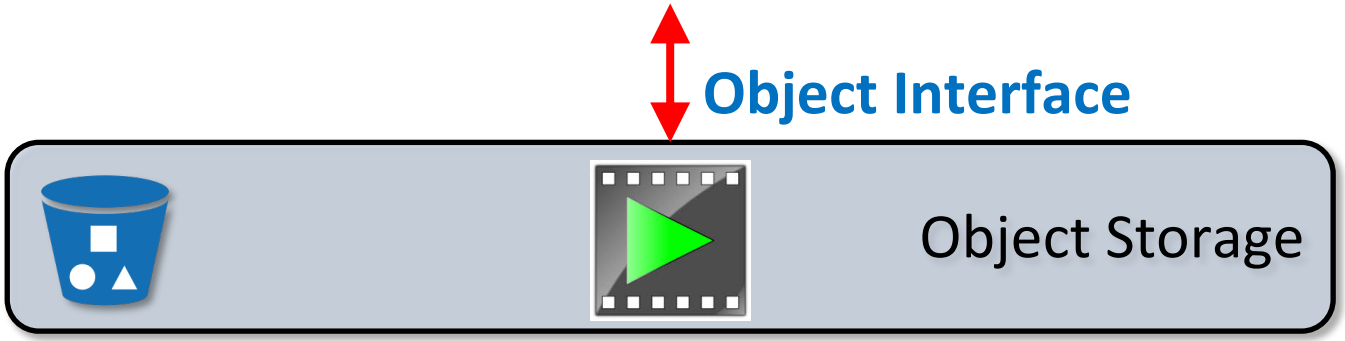
Outline

- Design considerations
- Existing systems
- Agni
- Future work

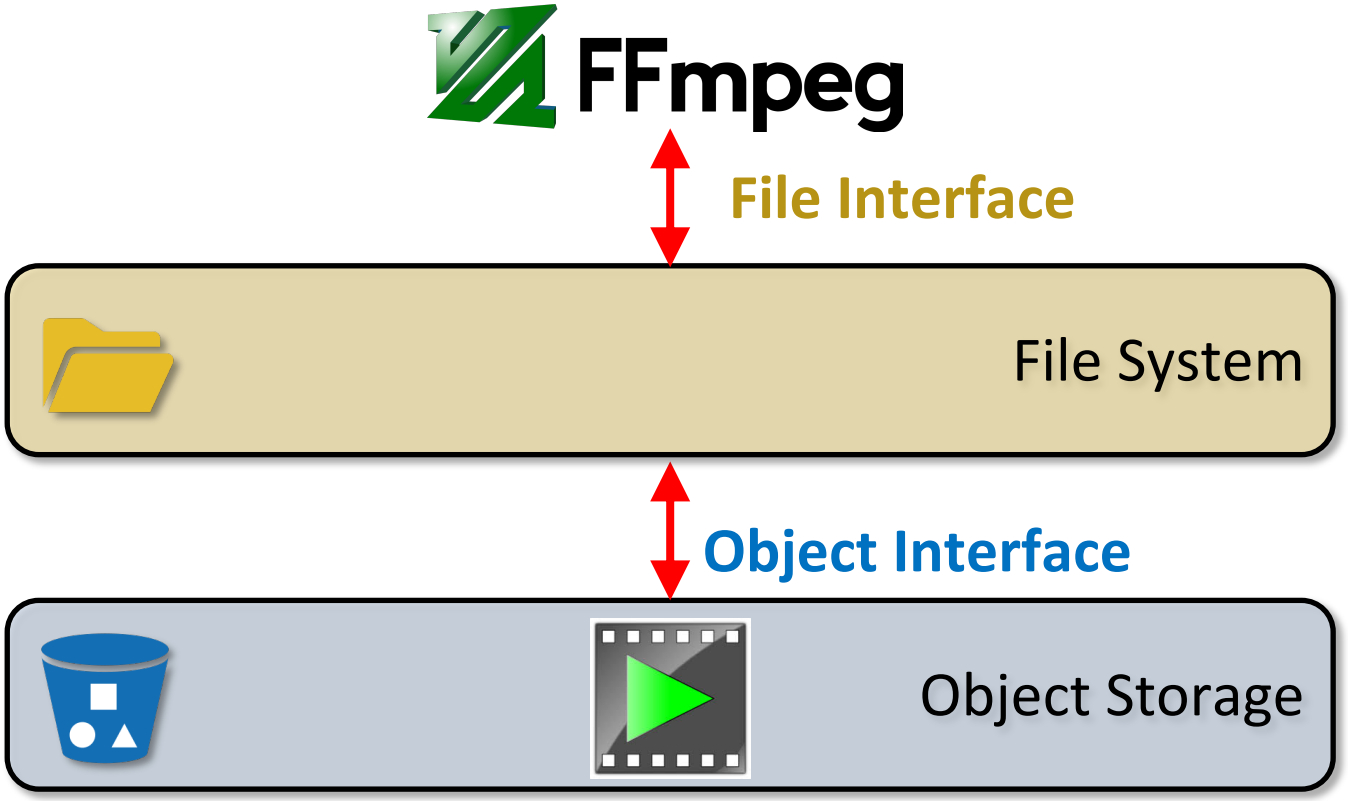
https://www.greenbiz.com/sites/default/files/styles/gbz_article_primary_breakpoints_kalapicture_screenmd_1x/public/images/articles/featured/datacenter_0.jpg?itok=iJm7ezgB&timestamp=1483504030

File Systems Paired With Object Storage

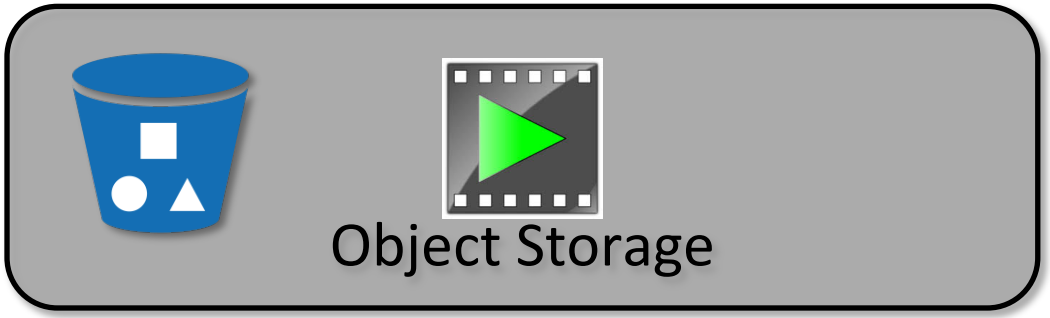
FFmpeg



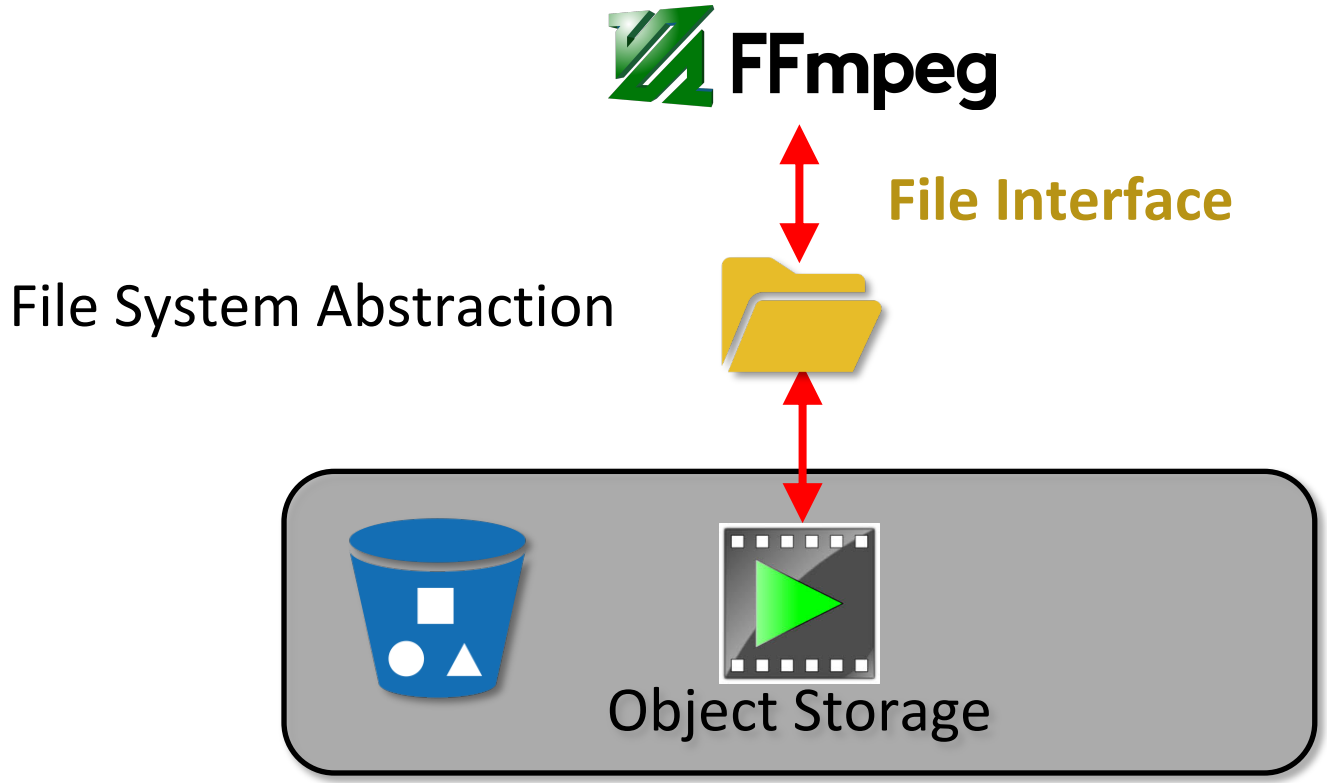
File Systems Paired With Object Storage



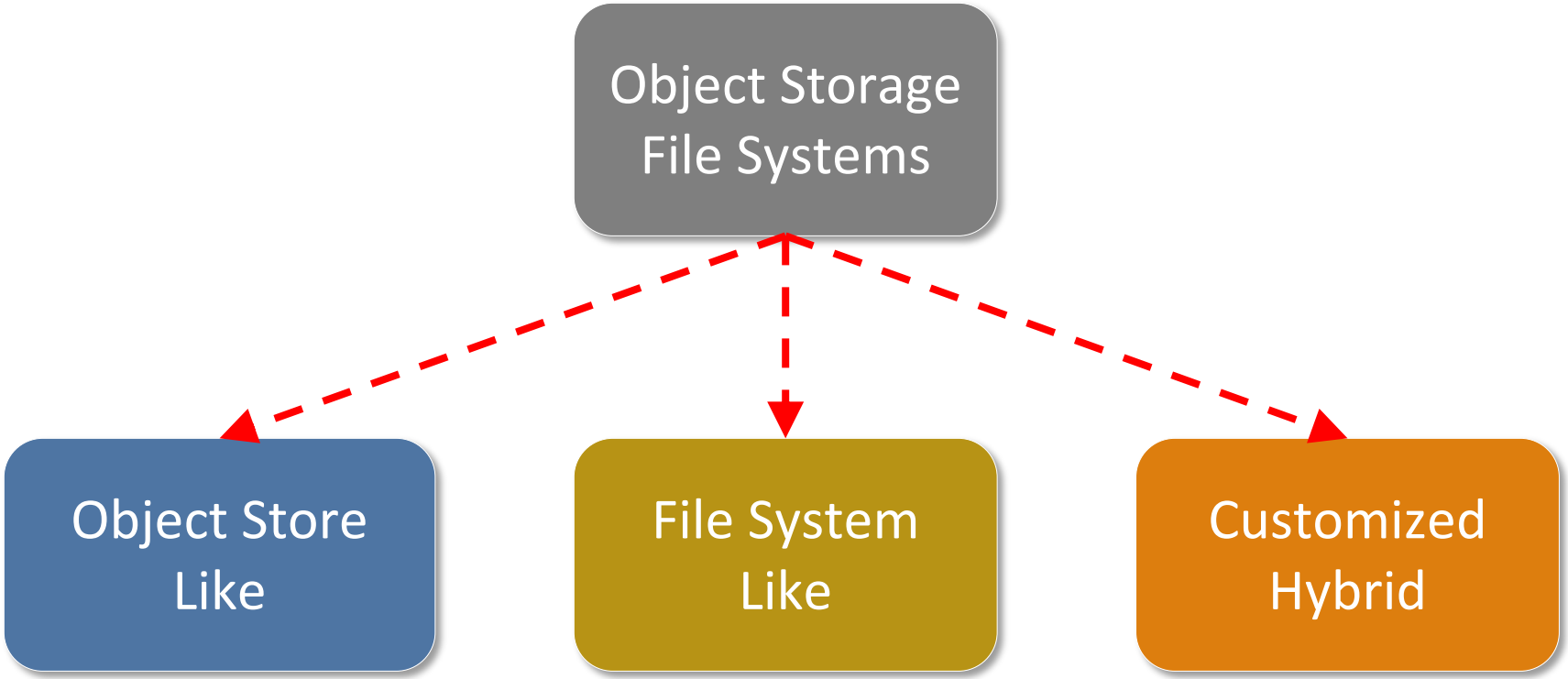
Object Storage File System



Object Storage File System



Existing Systems



Existing Systems: Object Store Like

✓

- Dual Access
- Generic

✗

- Efficient File Interfaces
- Coherent Namespace
- Distributed
- Unified Access Control

Popular file systems: S3FS, GoogleCloudFuse

Existing Systems: File System Like

- Generic
- Efficient File Interfaces
- Distributed

- Dual Access
- Coherent Namespace
- Unified Access Control

Popular file systems: CephFS, MarFS

Existing Systems: Customized Hybrid

- Dual Access
- Efficient File Interfaces
- Coherent Namespace

- Generic
- Distributed
- Unified Access Control

Popular file systems: ProxyFS, OpenIOFS

Outline

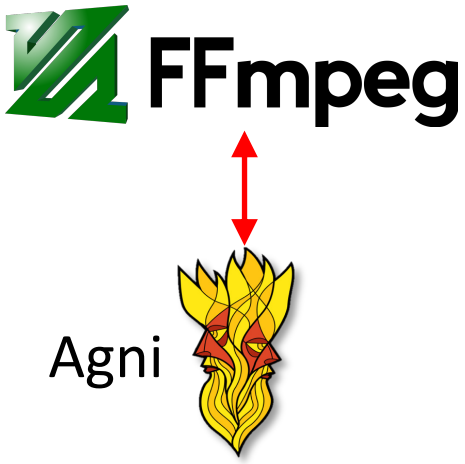
- Design considerations
- Existing systems
- Agni
- Future work

Agni

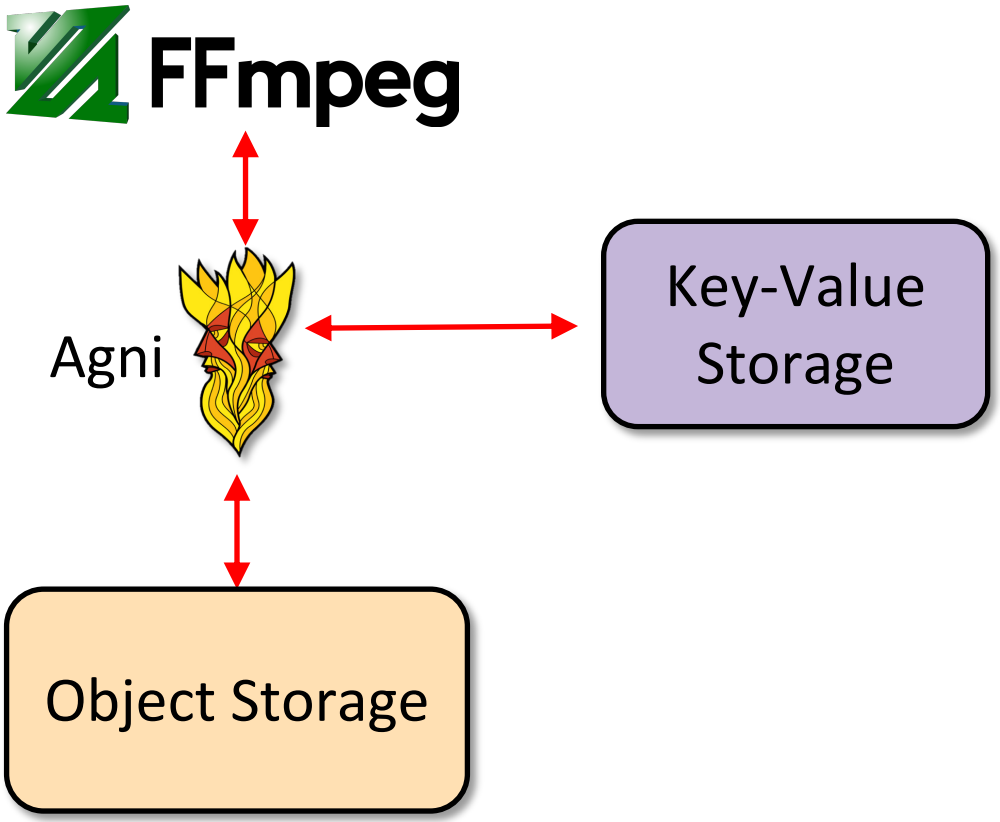
- Generic
- Dual Access
- Efficient File Interfaces
- Distributed

- Coherent Namespace
- Unified Access Control

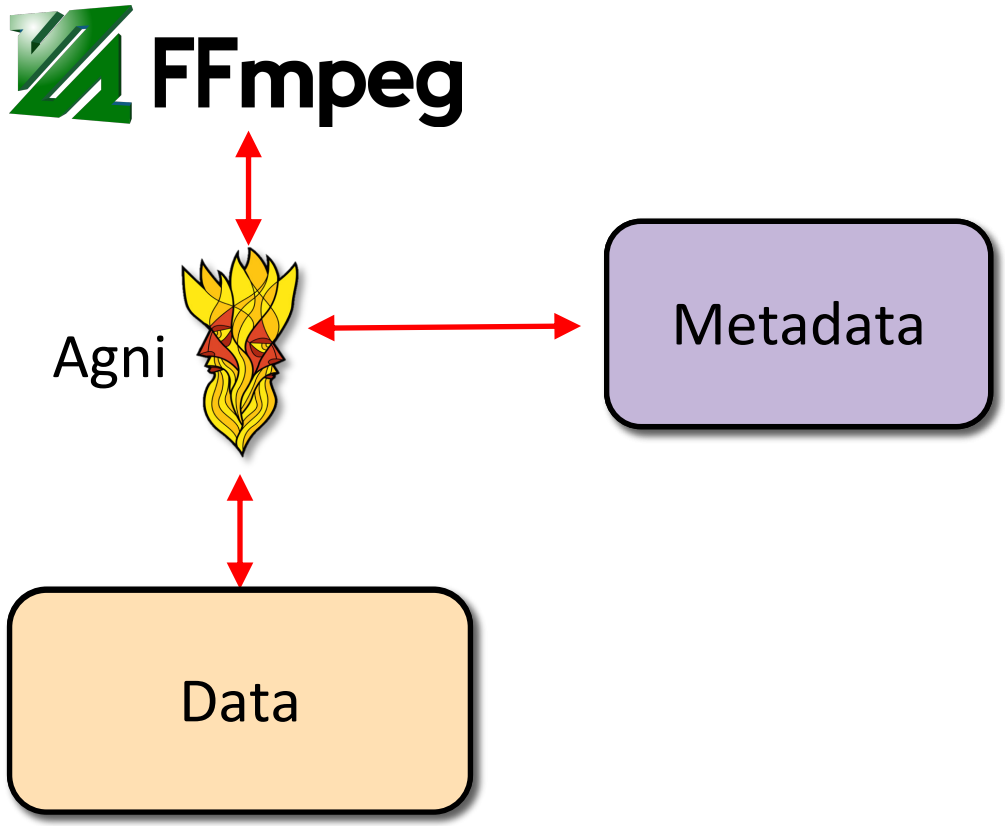
Architecture



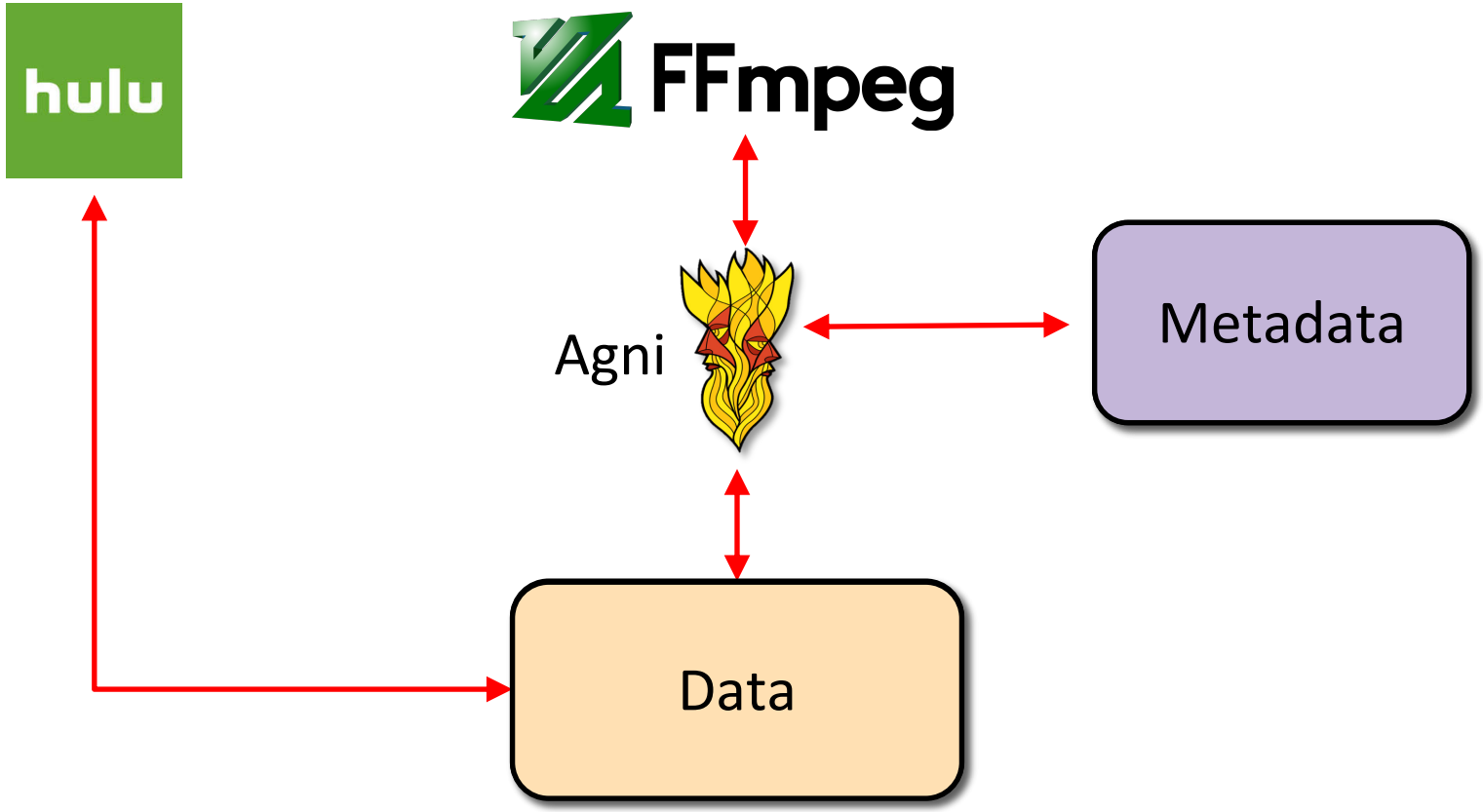
Architecture



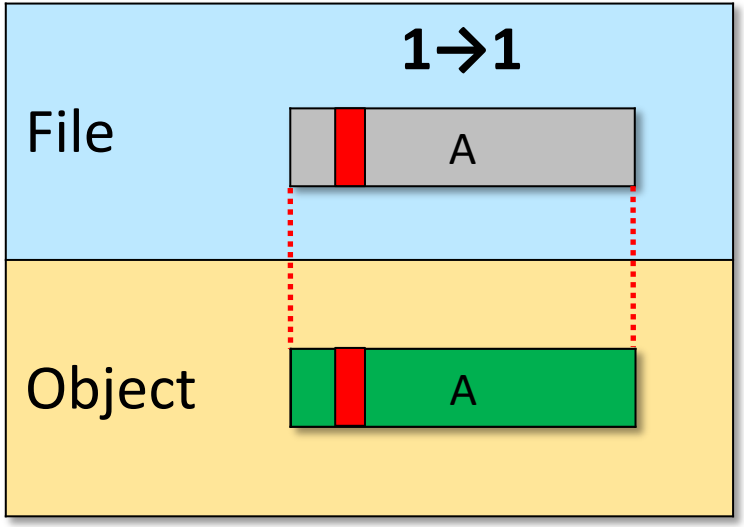
Architecture



Architecture

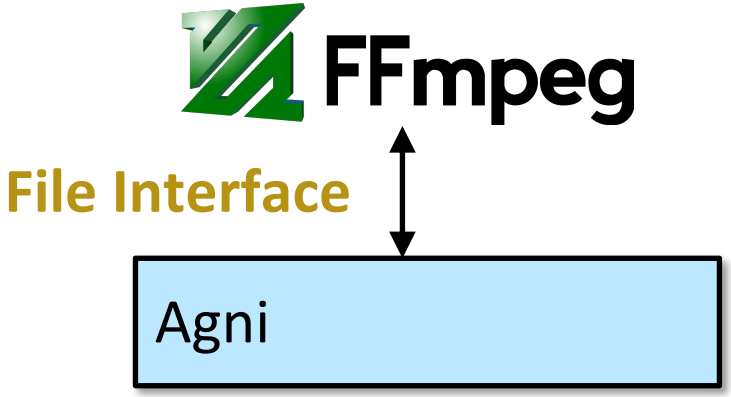


Our Design Choices

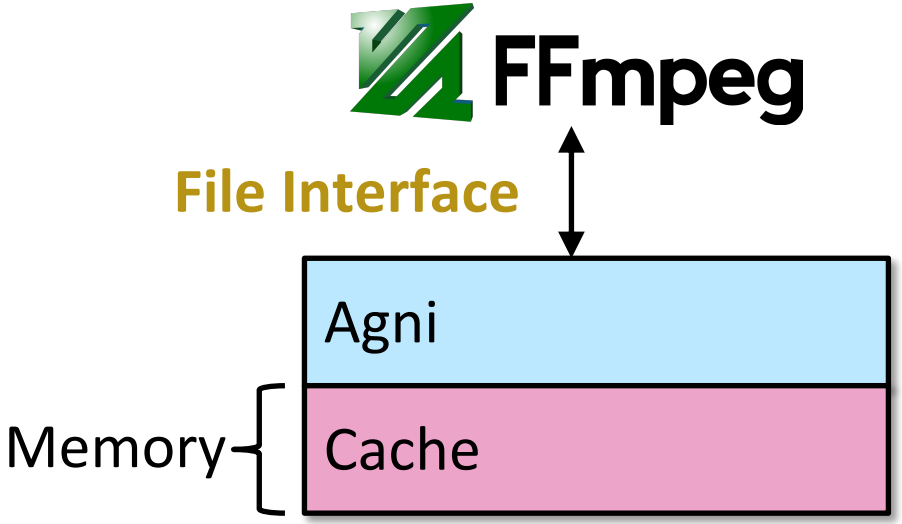


What about inefficient writes to immutable objects ?

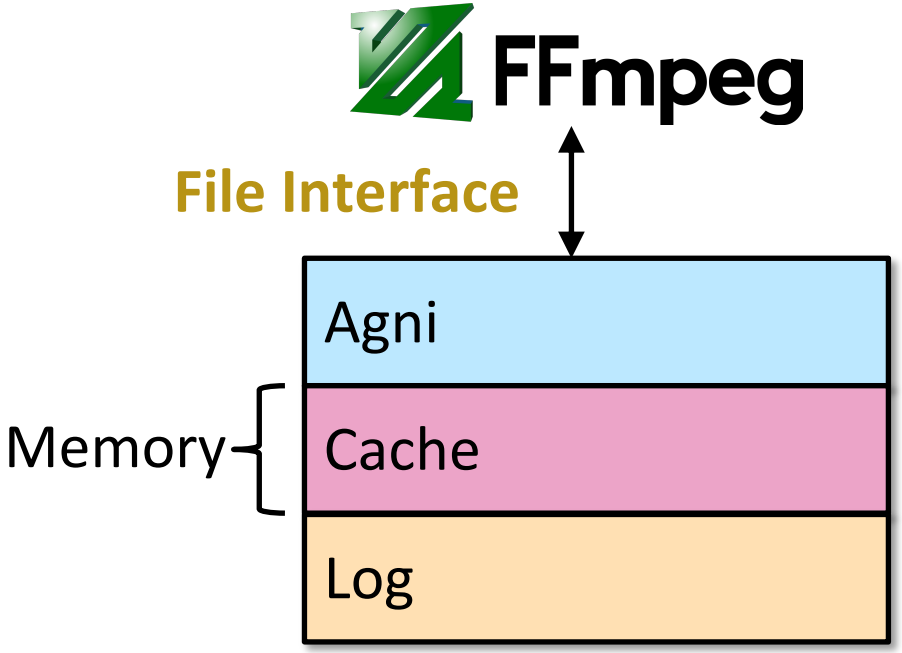
Multi-Tier Data Structure



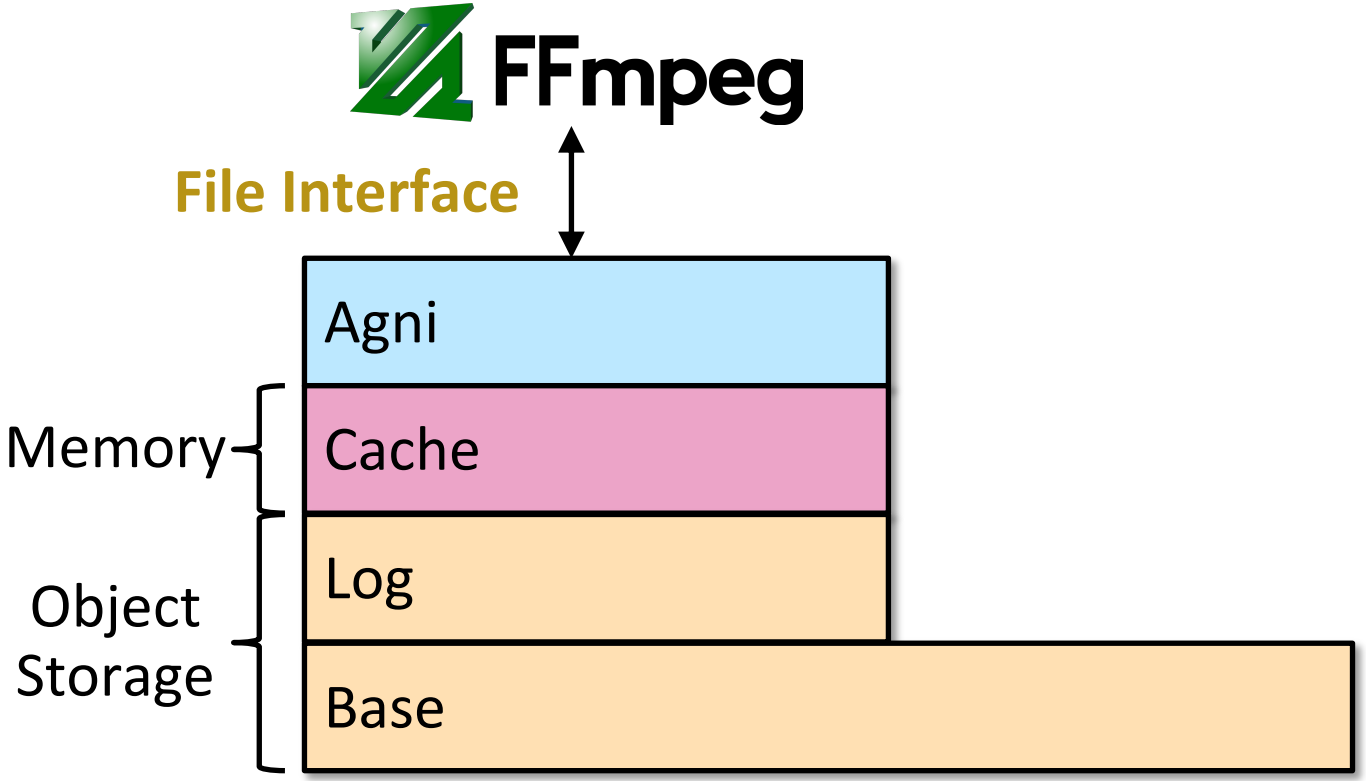
Multi-Tier Data Structure



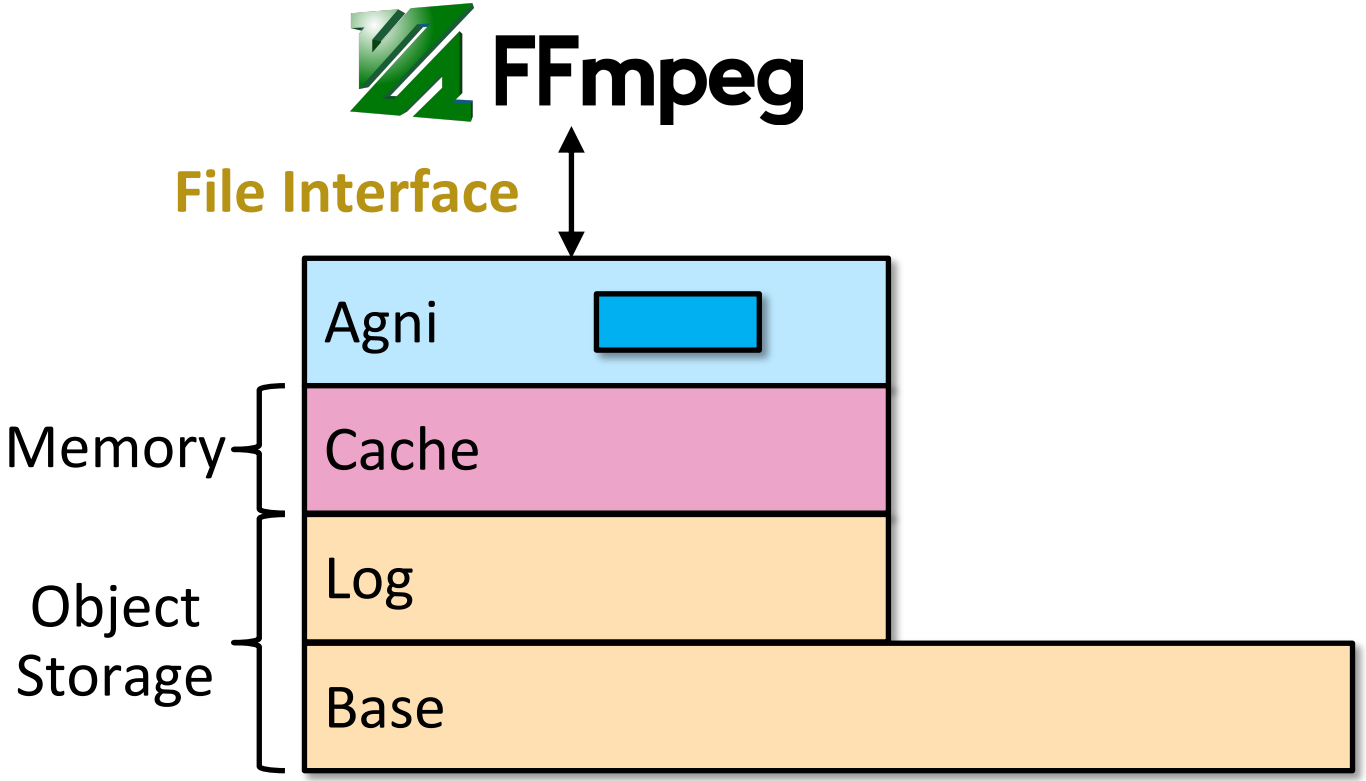
Multi-Tier Data Structure



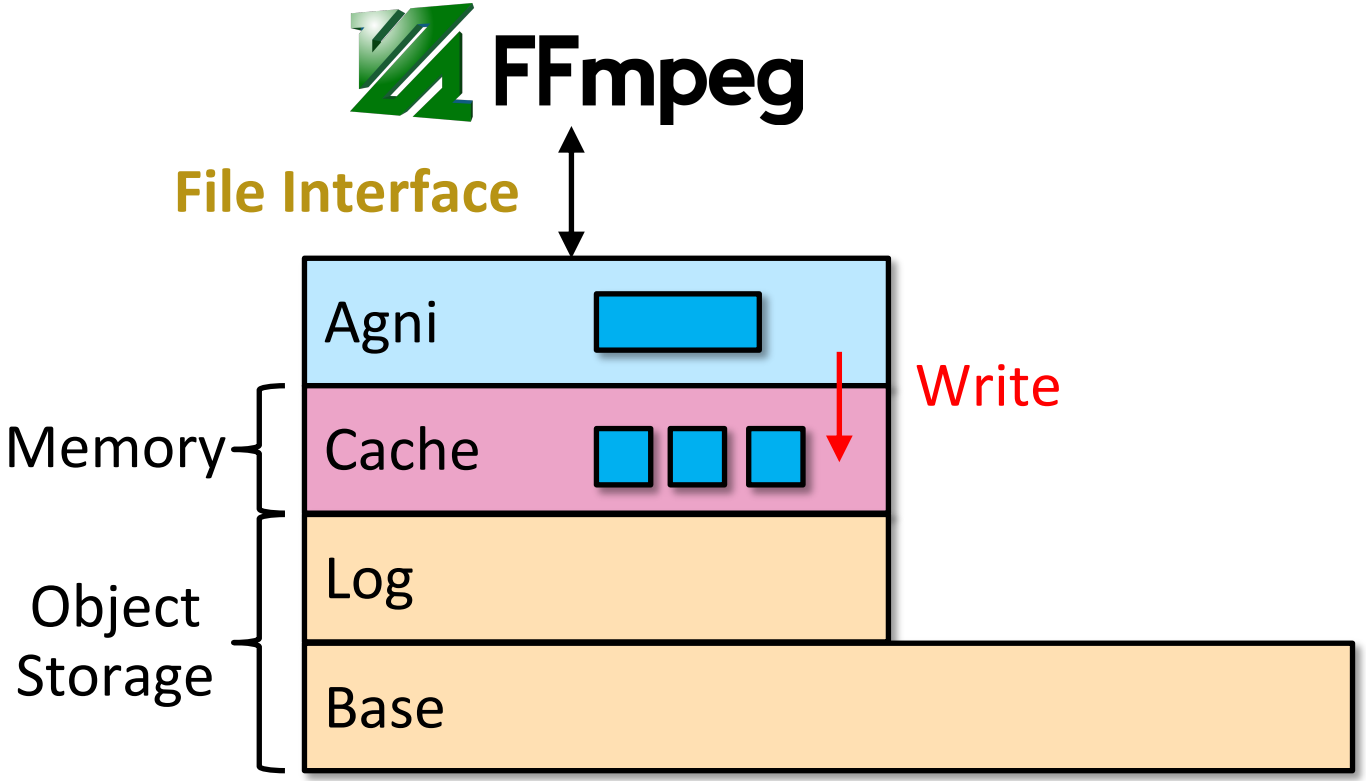
Multi-Tier Data Structure



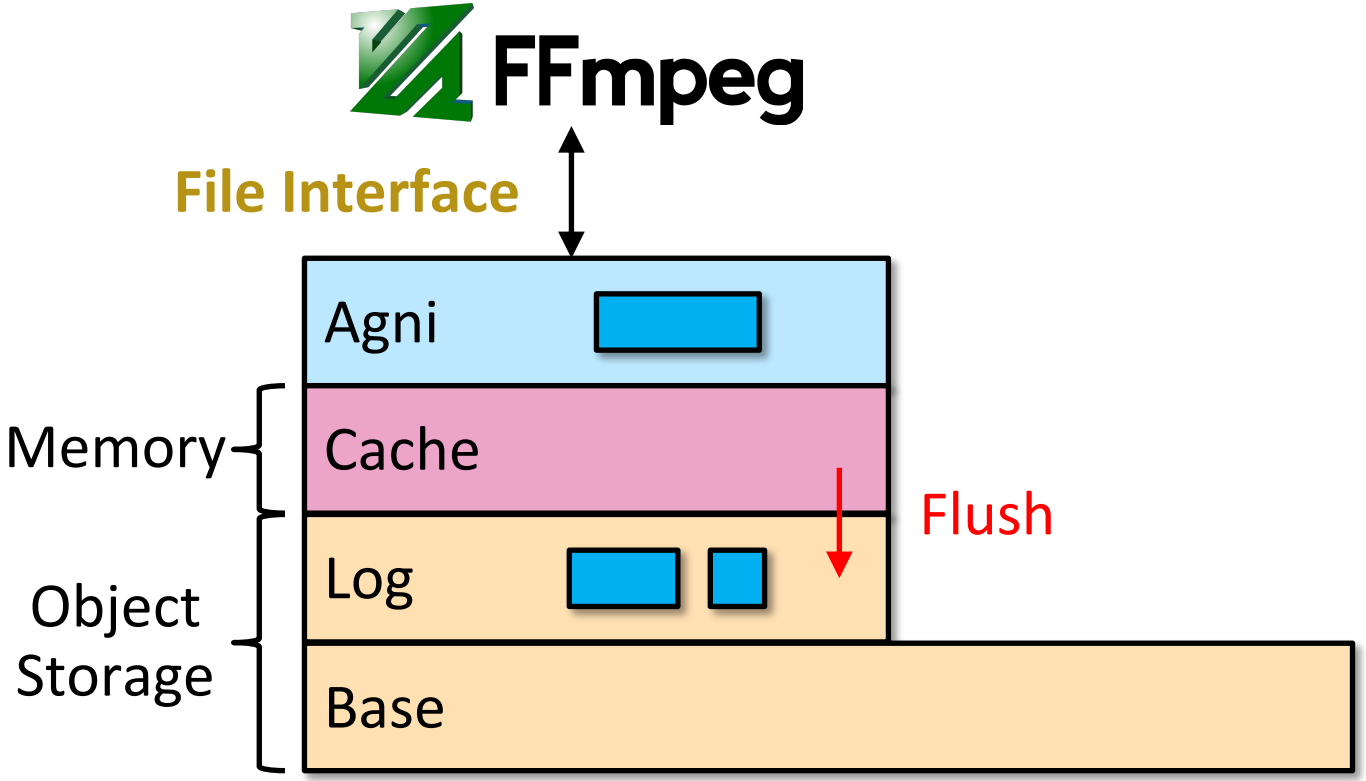
Multi-Tier Data Structure



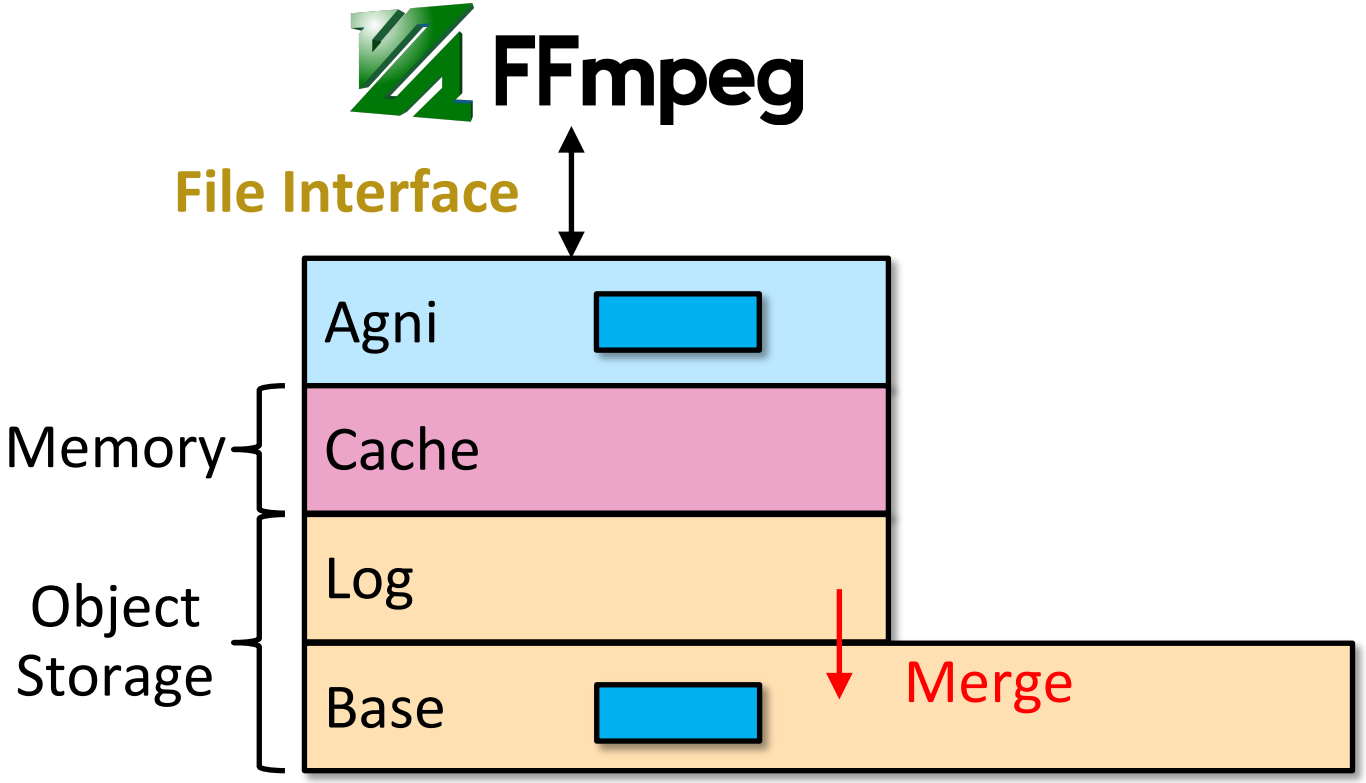
Multi-Tier Data Structure



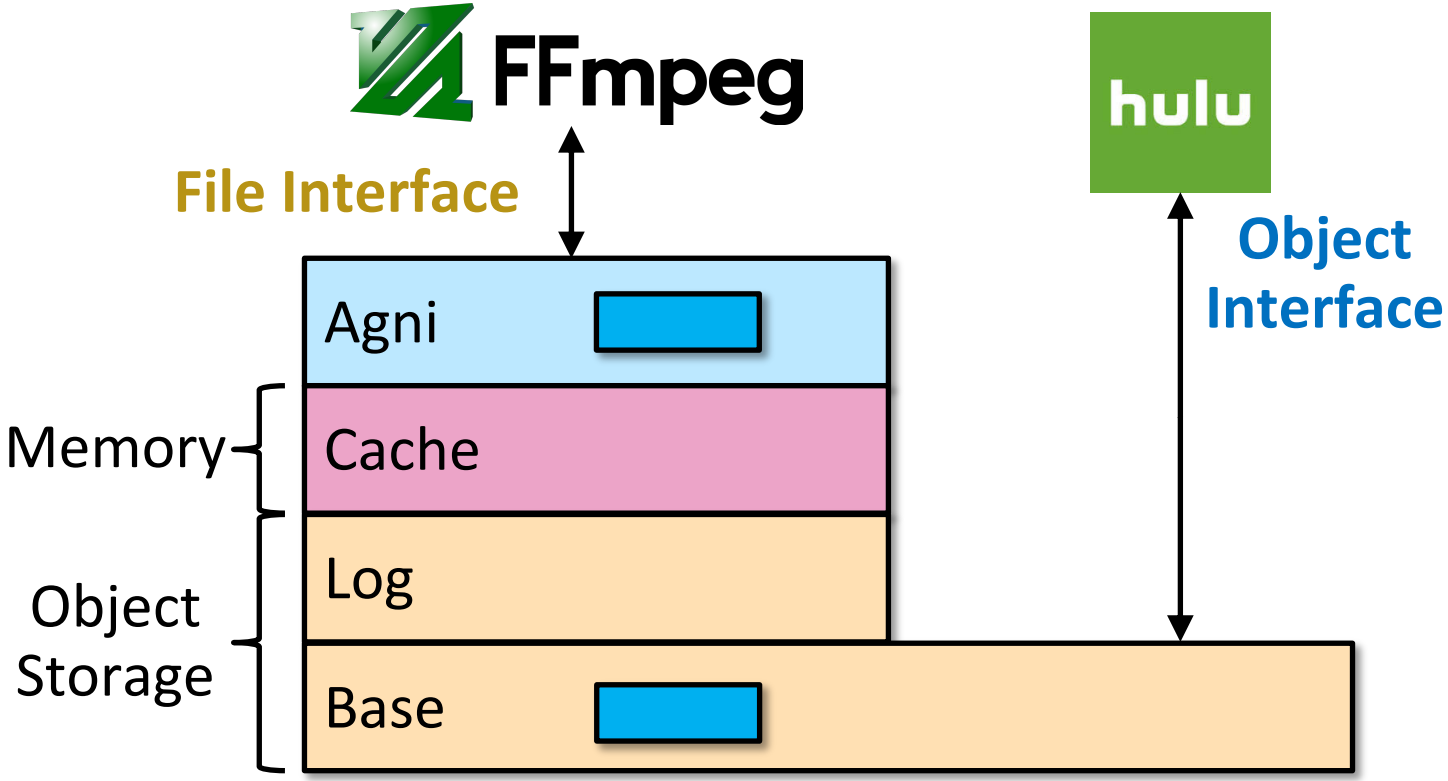
Multi-Tier Data Structure



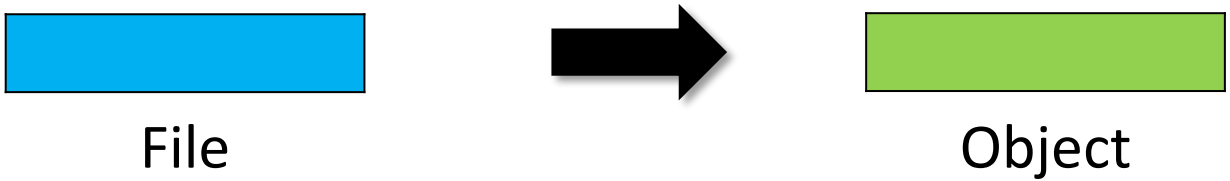
Multi-Tier Data Structure



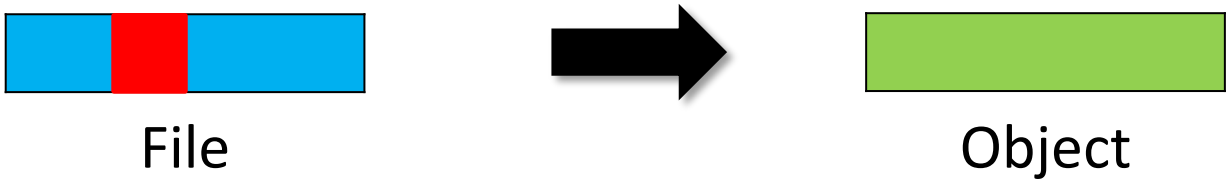
Multi-Tier Data Structure



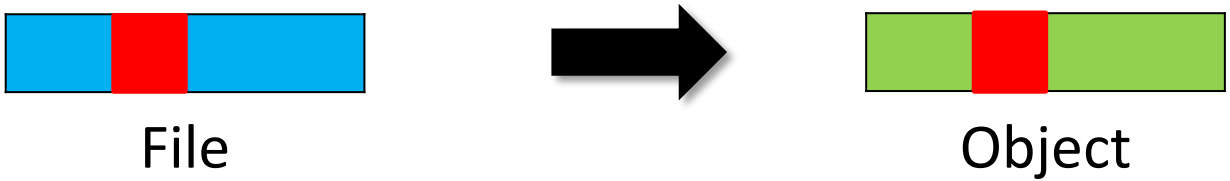
We Call This Approach Eventual 1→1 Mapping



We Call This Approach Eventual 1→1 Mapping

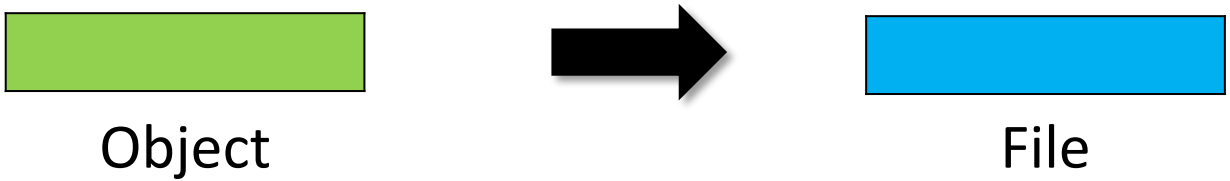


We Call This Approach Eventual 1→1 Mapping

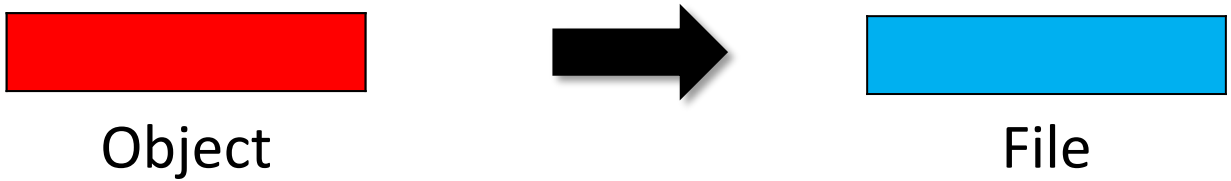


File to Object Visibility Lag

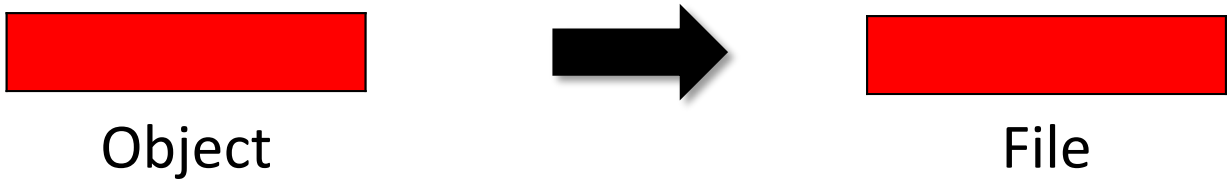
Eventual 1→1 Mapping Works Both Ways



Eventual 1→1 Mapping Works Both Ways

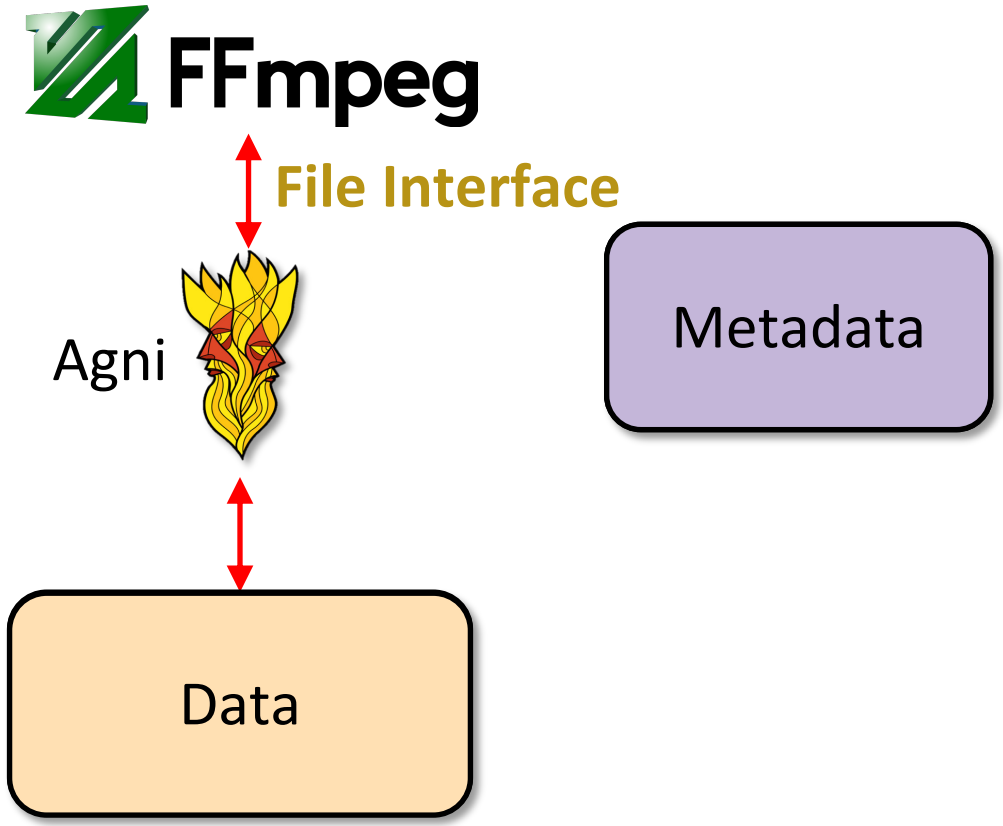


Eventual 1→1 Mapping Works Both Ways



Object to File Visibility Lag

Design: Object to File



Design: Object to File

Object Interface

FFmpeg

File Interface

Agni

Metadata

Design: Object to File

FFmpeg

File Interface

Object Interface

Agni

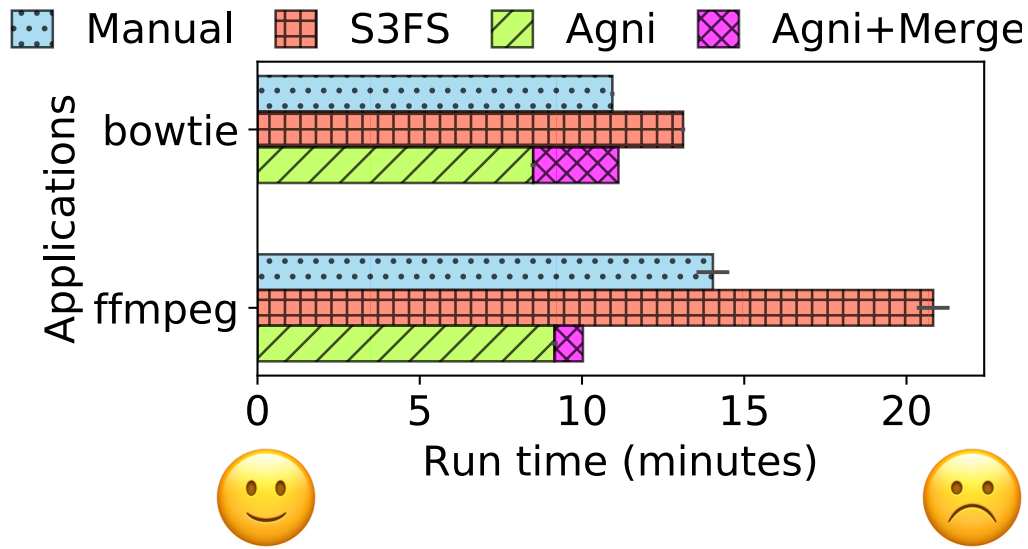
Visible

Metadata

Notification Processor

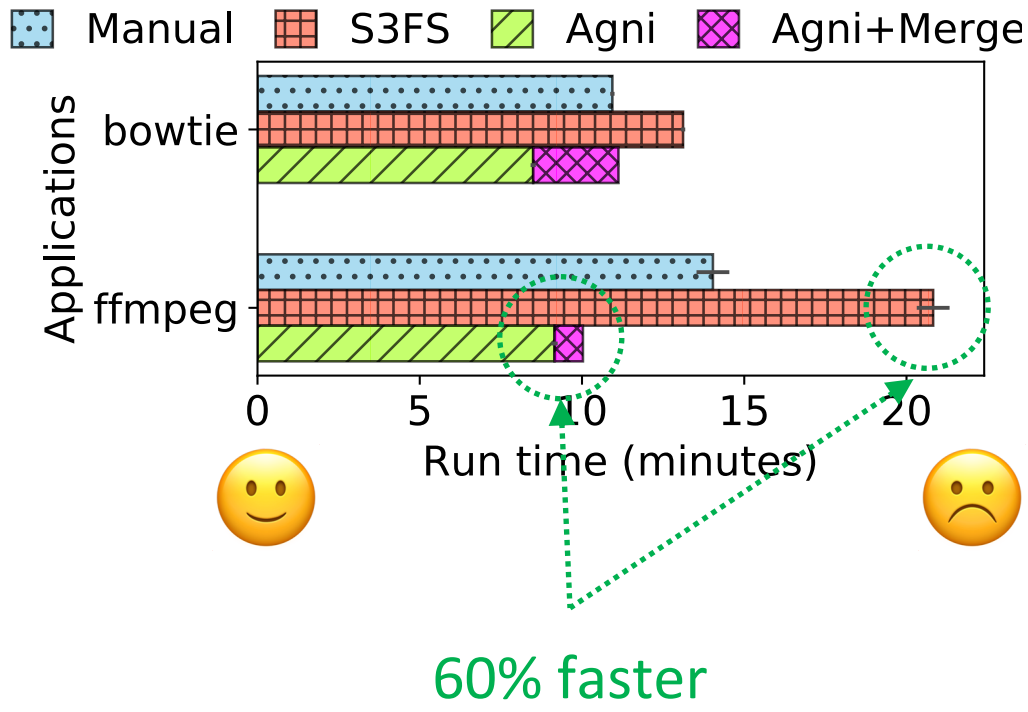
Evaluation: Applications

- ***ffmpeg***: 30 GB of MPEG to MOV files
- ***bowtie***: 8GB single genome file
- Agni+Merge denotes when dual access is enabled



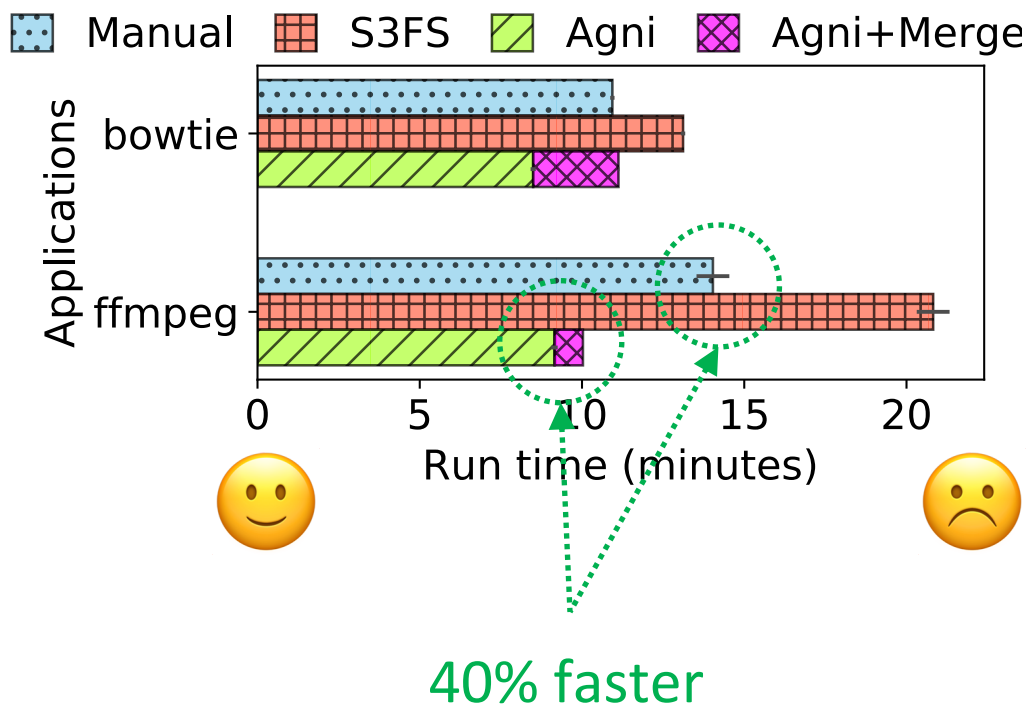
Evaluation: Applications

- ***ffmpeg***: 30 GB of MPEG to MOV files
- ***bowtie***: 8GB single genome file
- Agni+Merge denotes when dual access is enabled



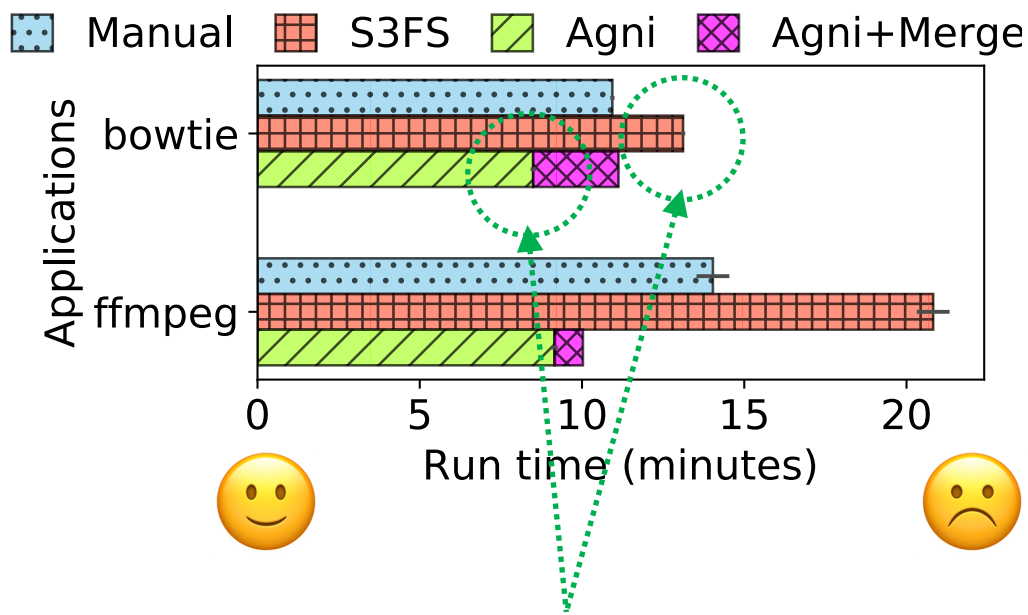
Evaluation: Applications

- ***ffmpeg***: 30 GB of MPEG to MOV files
- ***bowtie***: 8GB single genome file
- Agni+Merge denotes when dual access is enabled



Evaluation: Applications

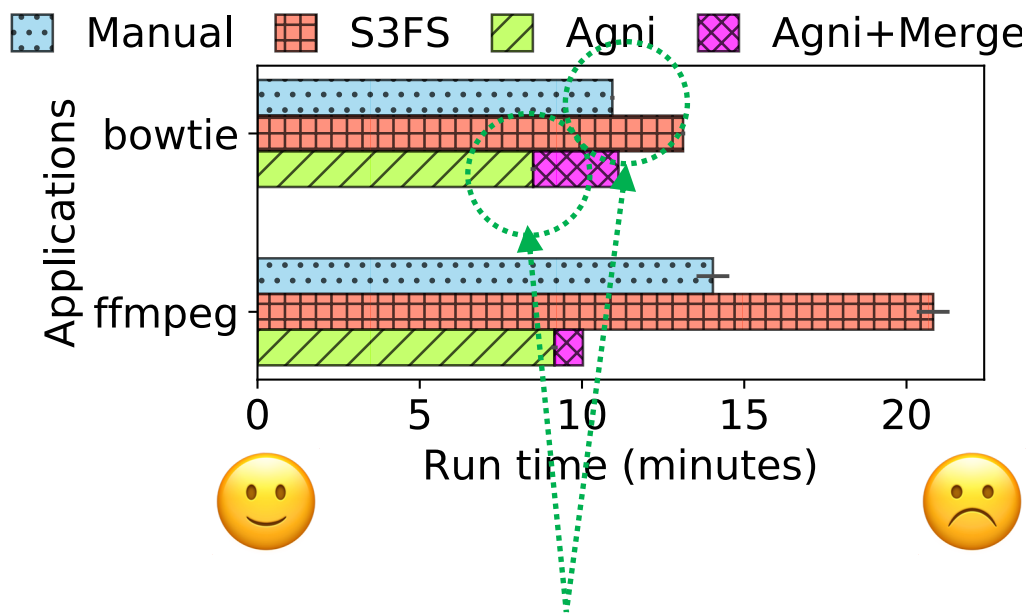
- ***ffmpeg***: 30 GB of MPEG to MOV files
- ***bowtie***: 8GB single genome file
- Agni+Merge denotes when dual access is enabled



40% faster

Evaluation: Applications

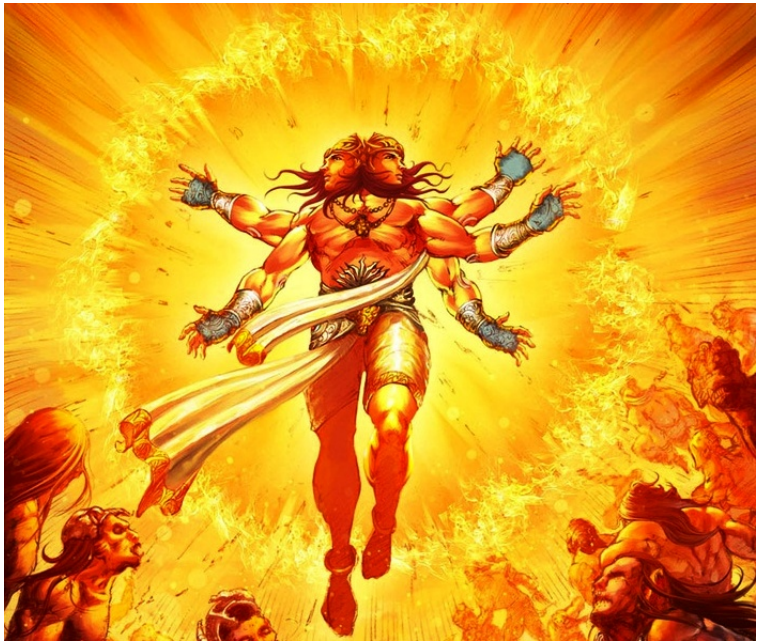
- ***ffmpeg***: 30 GB of MPEG to MOV files
- ***bowtie***: 8GB single genome file
- Agni+Merge denotes when dual access is enabled



20% faster

Outline

- Design considerations
- Existing systems
- Agni
- Future work



Future Work

- Implement coherent namespaces and unified access control
- Mode I
  - **Best** object interface performance
  - User intervention **required** for namespace coherence
  - Access control **dependent** on object store
- Mode II
  - **Additional** object interfaces
  - **Prevents** any namespace incoherence
  - Uniform access control **independent** of object storage


Thank you!
https://github.com/objectfs/objectfs
Contact: lillaney@jhu.edu

Discussion points

- Challenges in hybrid cloud deployment
- Acceptability of weak data access consistency
- Approaches to coherent namespace
- Approaches to unified access control