Introducing Apple File System

A snapshot of the next generation in storage

Session 701

Eric Tamura Manager, Local File Systems
Dominic Giampaolo Senior Software Engineer, Storage / File Systems

What is Apple File System?

What is Apple File System?

Introduction / Motivation

New Features

Demo

New APIs

What is Apple File System?

Introduction / Motivation

New Features

Demo

New APIs

Apple File System

Introducing Apple File System (APFS)

Introducing Apple File System (APFS)

Next Generation File System

Introducing Apple File System (APFS)

Next Generation File System

Designed to scale from an
Apple Watch to a Mac Pro

watchOS

iOS

tvOS

macOS

Introducing Apple File System (APFS)

Next Generation File System

Designed to scale from an Apple Watch to a Mac Pro

Designed to take advantage of flash / SSD storage

watchOS

iOS

tvOS

macOS

Introducing Apple File System (APFS)

Next Generation File System

Designed to scale from an Apple Watch to a Mac Pro

Designed to take advantage of flash / SSD storage

Engineered with encryption as a primary feature

watchOS

iOS

tvOS

macOS

Motivation

What about HFS+ ?

Motivation

What about HFS+ ?

Currently shipping HFS+ as primary file system

Motivation

What about HFS+ ?

Currently shipping HFS+ as primary file system

… but its original design is over 30 years old.

Motivation

What about HFS+ ?

Currently shipping HFS+ as primary file system

… but its original design is over 30 years old.

Designed in an era where floppies and HDDs were state of the art

Motivation

What about HFS+ ?

Currently shipping HFS+ as primary file system

… but its original design is over 30 years old.

Designed in an era where floppies and HDDs were state of the art

Single-threaded data structures

Motivation

What about HFS+ ?

Currently shipping HFS+ as primary file system

… but its original design is over 30 years old.

Designed in an era where floppies and HDDs were state of the art

Single-threaded data structures

Rigid data structures

Motivation

Why a new file system?

Motivation

Why a new file system?

Designed (and tuned) for Apple products and ecosystem

Motivation

Why a new file system?

Designed (and tuned) for Apple products and ecosystem

Scale file system footprint to support Apple Watch up to Mac Pro

Motivation

Why a new file system?

Motivation

Why a new file system?

Enhance security capabilities

Motivation

Why a new file system?

Enhance security capabilities

Add new features!

Current File System / Storage SW

Current File System / Storage SW

HFS (Standard)

Current File System / Storage SW

HFS+

HFS (Standard)

Current File System / Storage SW

HFS+

HFS+J

HFS (Standard)

HFSX (Case Sensitive)

Current File System / Storage SW

HFS+

Fusion Drive

HFS+J

CoreStorage Full Disk Crypto

HFS (Standard)

HFSX (Case Sensitive)

CoreStorage

Current File System / Storage SW

iOS/tvOS/watchOS HFS+ Per-File Crypto

HFS+

HFS+J

Fusion Drive

CoreStorage Full Disk Crypto

iOS/tvOS/watchOS HFS+

HFS (Standard)

HFSX (Case Sensitive)

CoreStorage

Current File System / Storage SW

APFS

What is Apple File System?

Introduction / Motivation

New Features

Demo

New APIs

What is Apple File System?

Introduction / Motivation

New Features

Demo

New APIs

What is Apple File System?

Improved file system fundamentals

HFS compatibility

Space sharing

Cloning files and directories

Snapshots (and reversions)

Fast directory sizing

Atomic safe-save primitives

Encryption

What is Apple File System?

Improved file system fundamentals

HFS compatibility

Space sharing

Cloning files and directories

Snapshots (and reversions)

Fast directory sizing

Atomic safe-save primitives

Encryption

Improved File System Fundamentals

Improved File System Fundamentals

Flash / SSD-optimized

Improved File System Fundamentals

Flash / SSD-optimized

Crash-protected

Improved File System Fundamentals

Flash / SSD-optimized

Crash-protected

Modern 64-bit native fields

Improved File System Fundamentals

Flash / SSD-optimized

Crash-protected

Modern 64-bit native fields

Extensible design for data structure growth

Improved File System Fundamentals

Flash / SSD-optimized

Crash-protected

Modern 64-bit native fields

Extensible design for data structure growth

Optimized for Apple software ecosystem

Improved File System Fundamentals

Flash / SSD-optimized

Crash-protected

Modern 64-bit native fields

Extensible design for data structure growth

Optimized for Apple software ecosystem

Low-latency design

Improved File System Fundamentals

Flash / SSD-optimized

Crash-protected

Modern 64-bit native fields

Extensible design for data structure growth

Optimized for Apple software ecosystem

Low-latency design

Native encryption support

What is Apple File System?

Improved file system fundamentals

HFS compatibility

Space sharing

Cloning files and directories

Snapshots (and reversions)

Fast directory sizing

Atomic safe-save primitives

Encryption

What is Apple File System?

Improved file system fundamentals

HFS compatibility

Space sharing

Cloning files and directories

Snapshots (and reversions)

Fast directory sizing

Atomic safe-save primitives

Encryption

HFS Compatibility

HFS Compatibility

Support and replace HFS+ functionality*

What is Apple File System?

Improved file system fundamentals

HFS compatibility

Space sharing

Cloning files and directories

Snapshots (and reversions)

Fast directory sizing

Atomic safe-save primitives

Encryption

What is Apple File System?

Improved file system fundamentals

HFS compatibility

Space sharing

Cloning files and directories

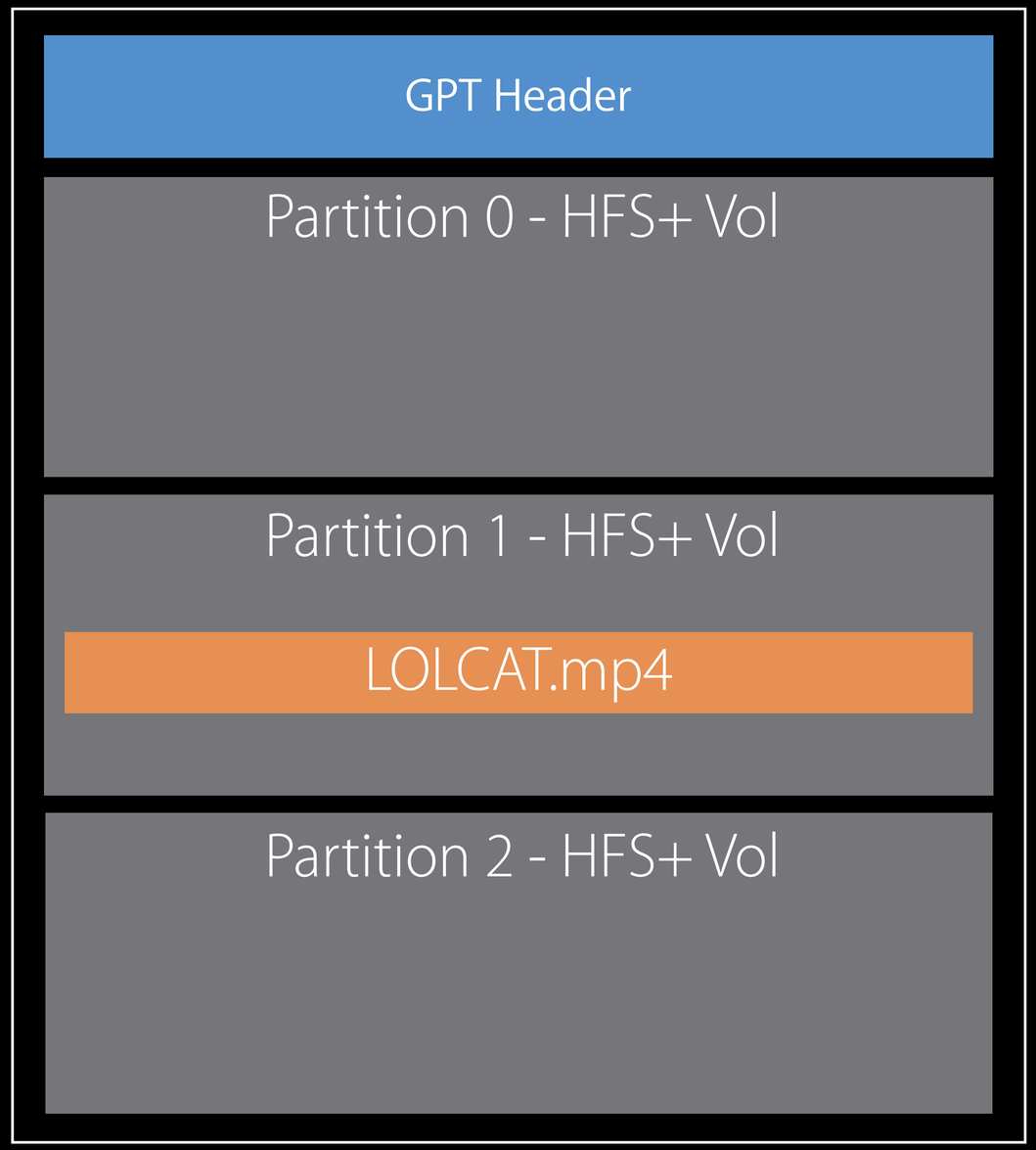
Snapshots (and reversions)

Fast directory sizing

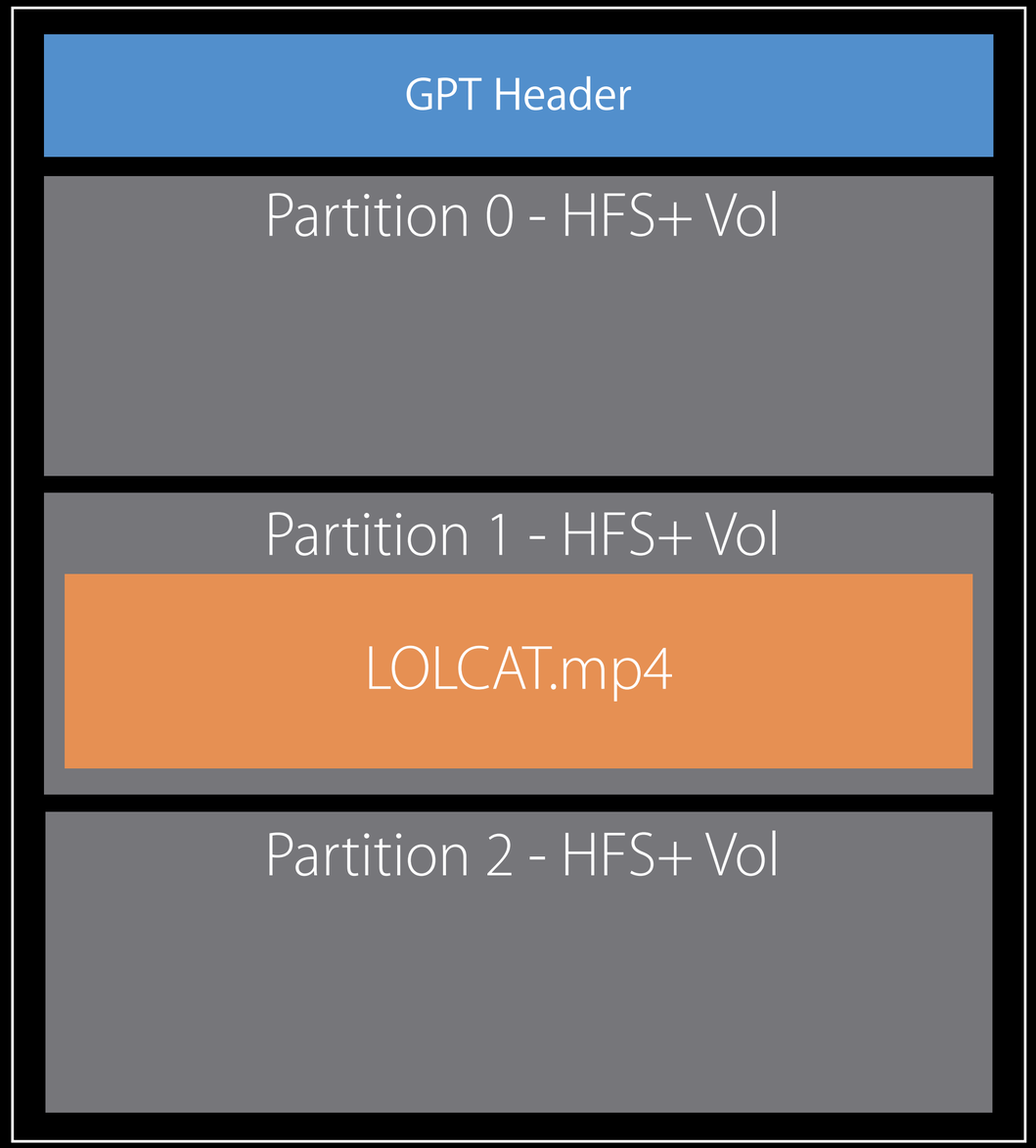
Atomic safe-save primitives

Encryption

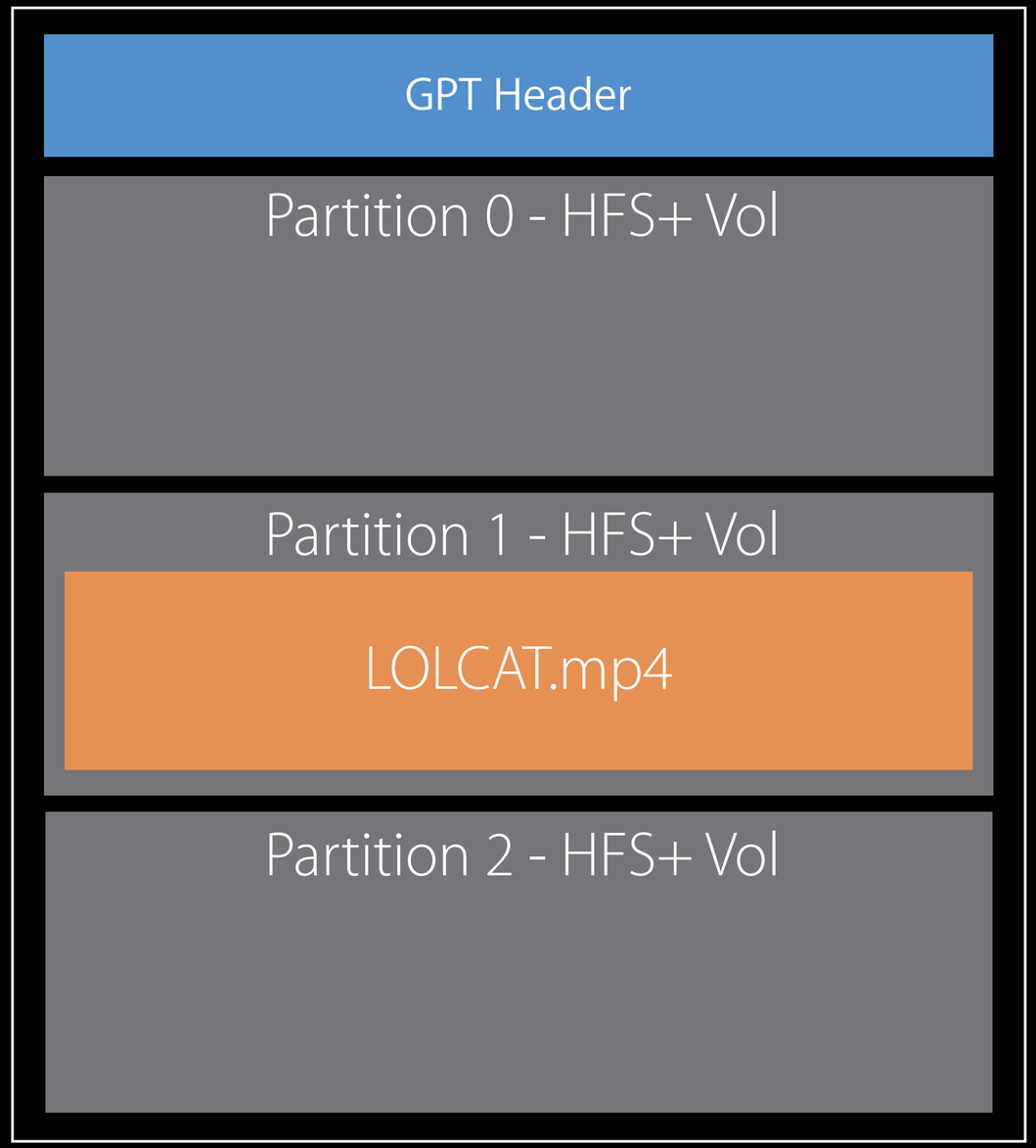
Space Sharing



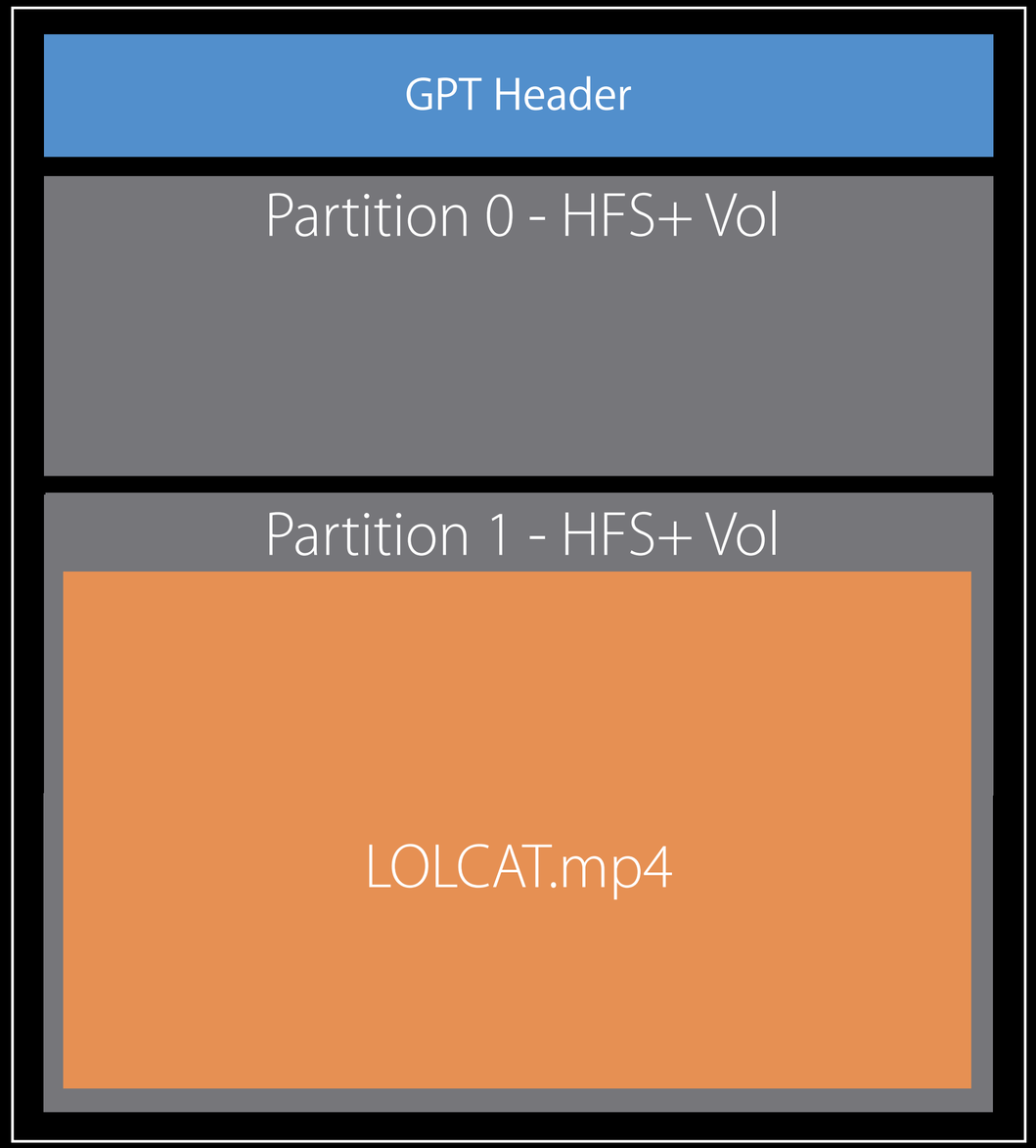
Space Sharing



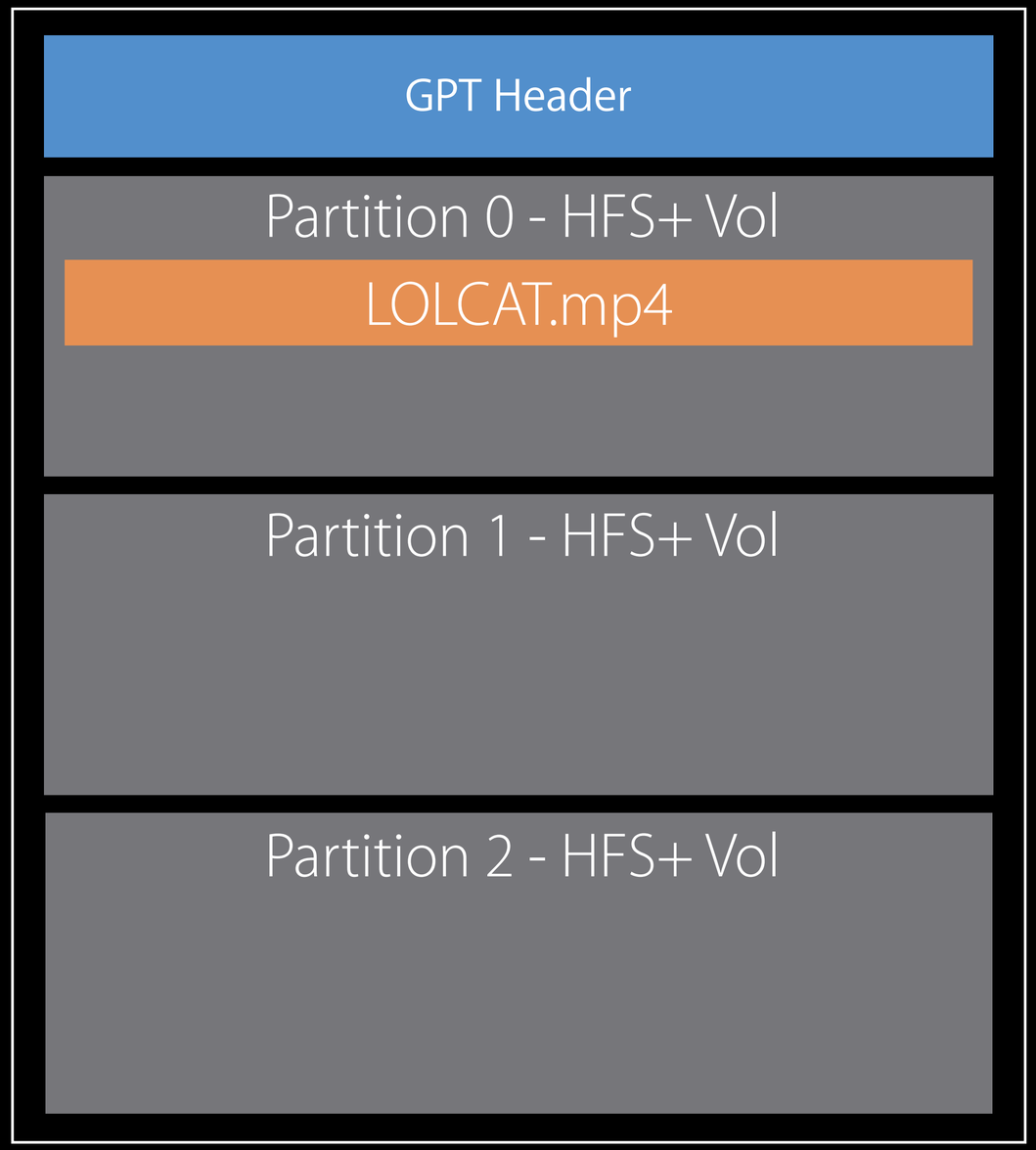
Space Sharing



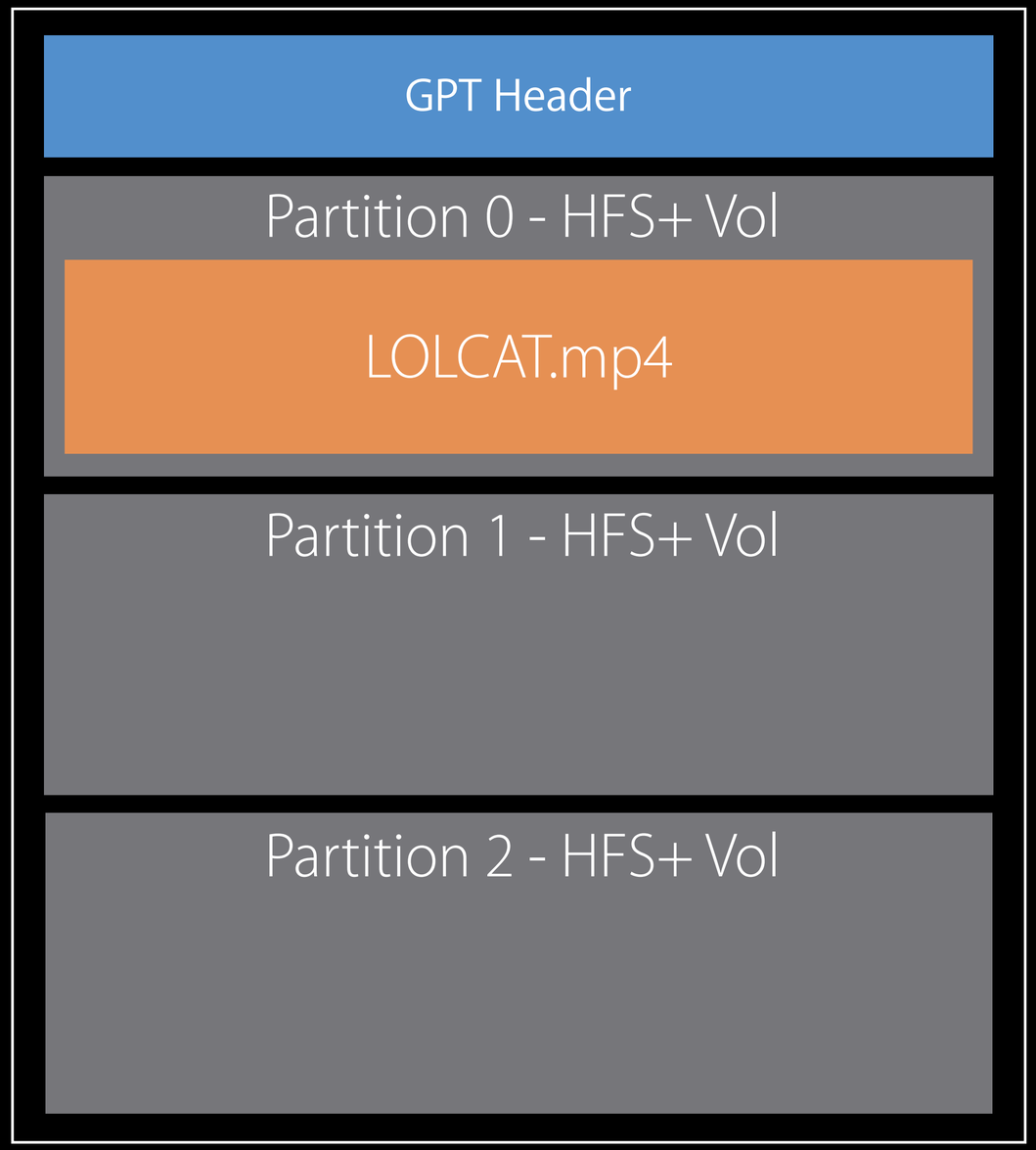
Space Sharing



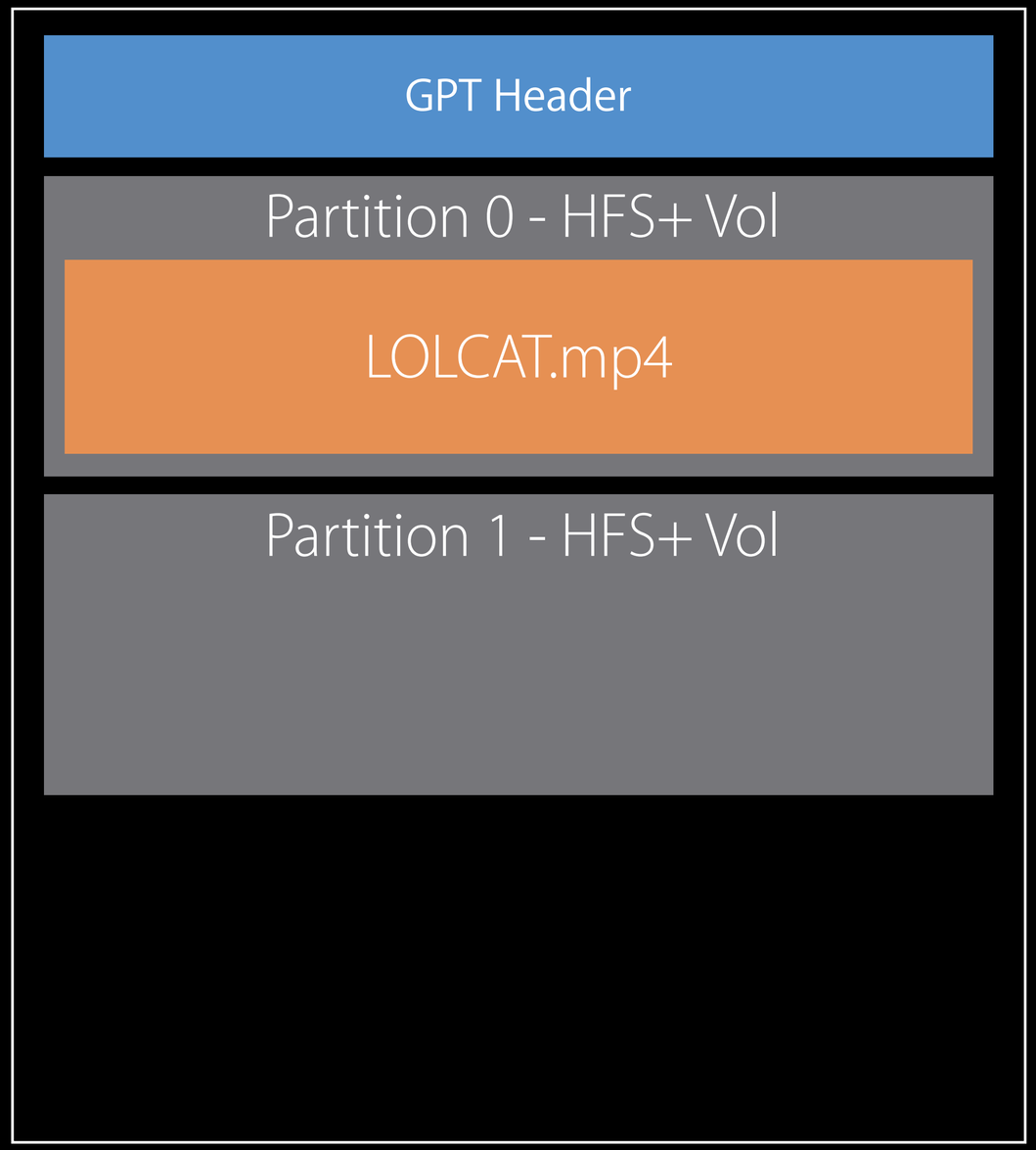
Space Sharing



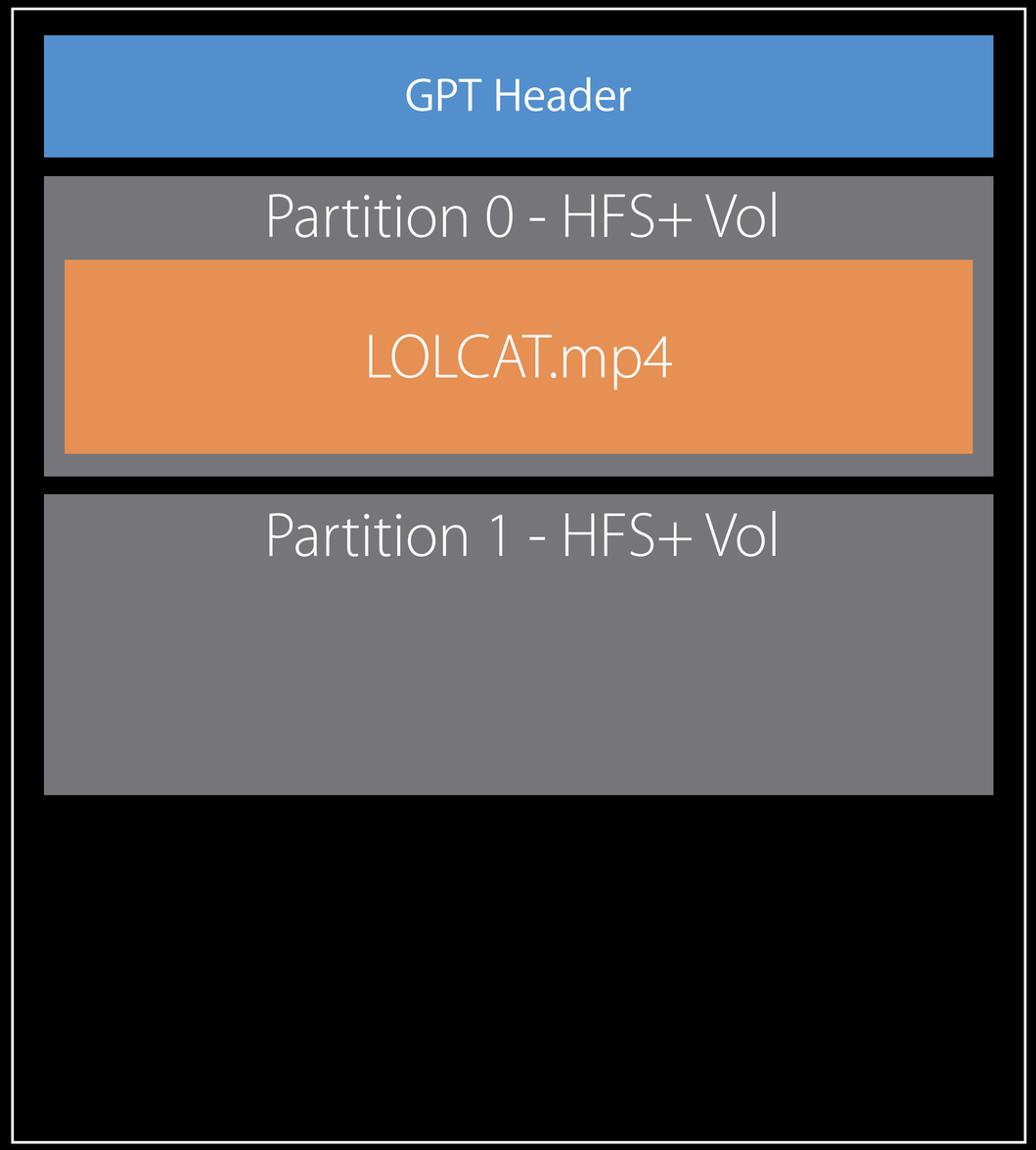
Space Sharing



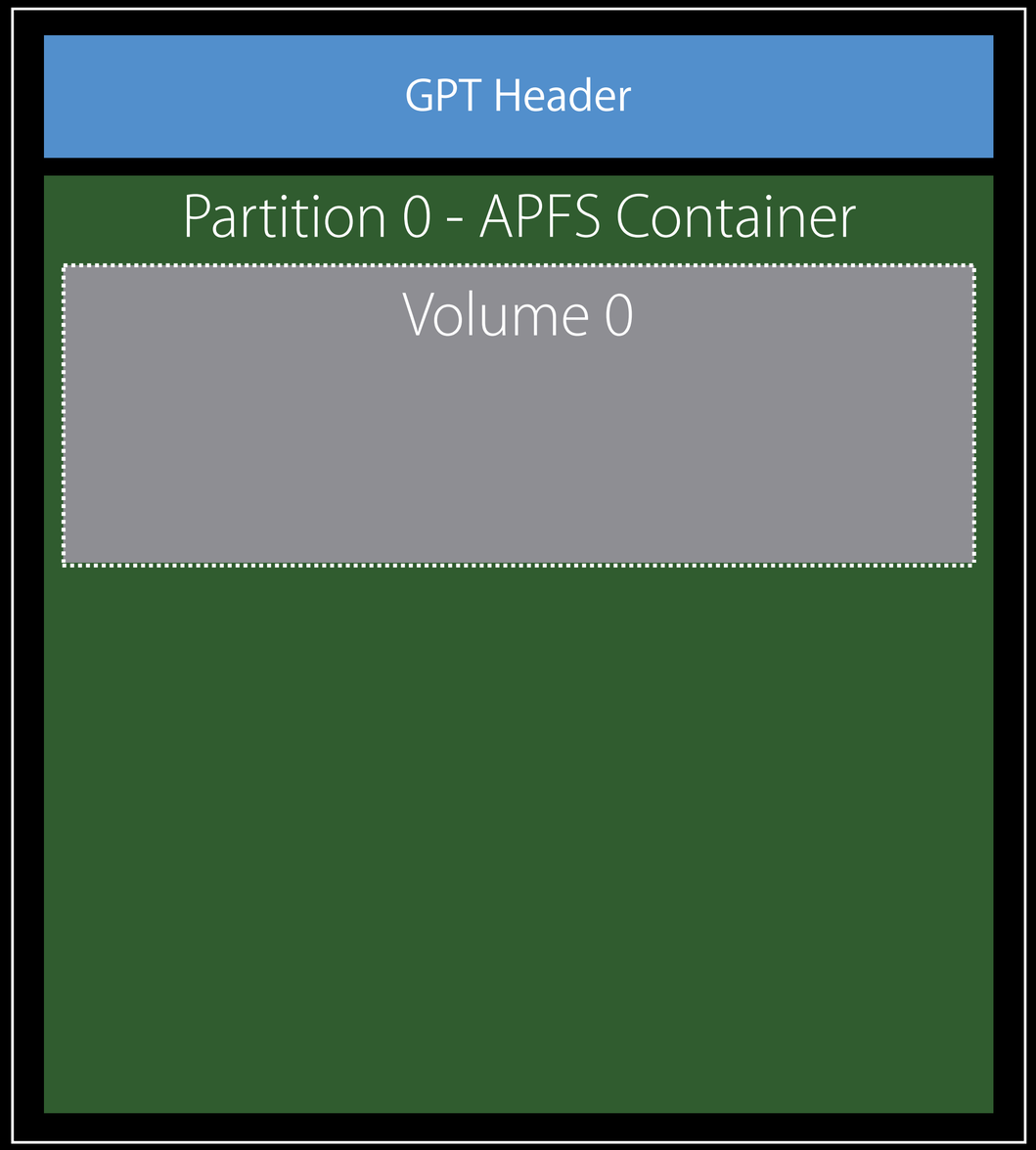
Space Sharing



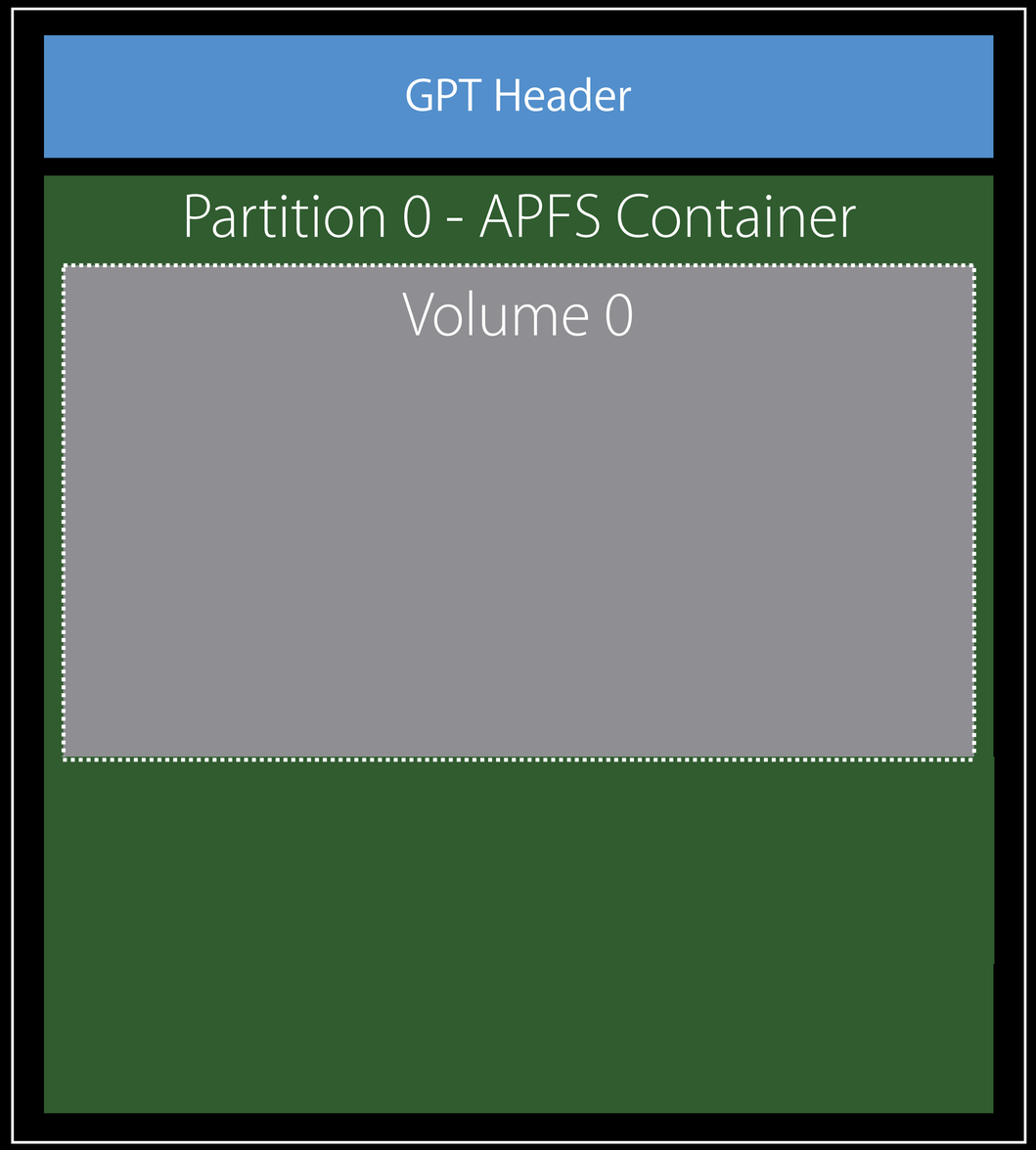
Space Sharing



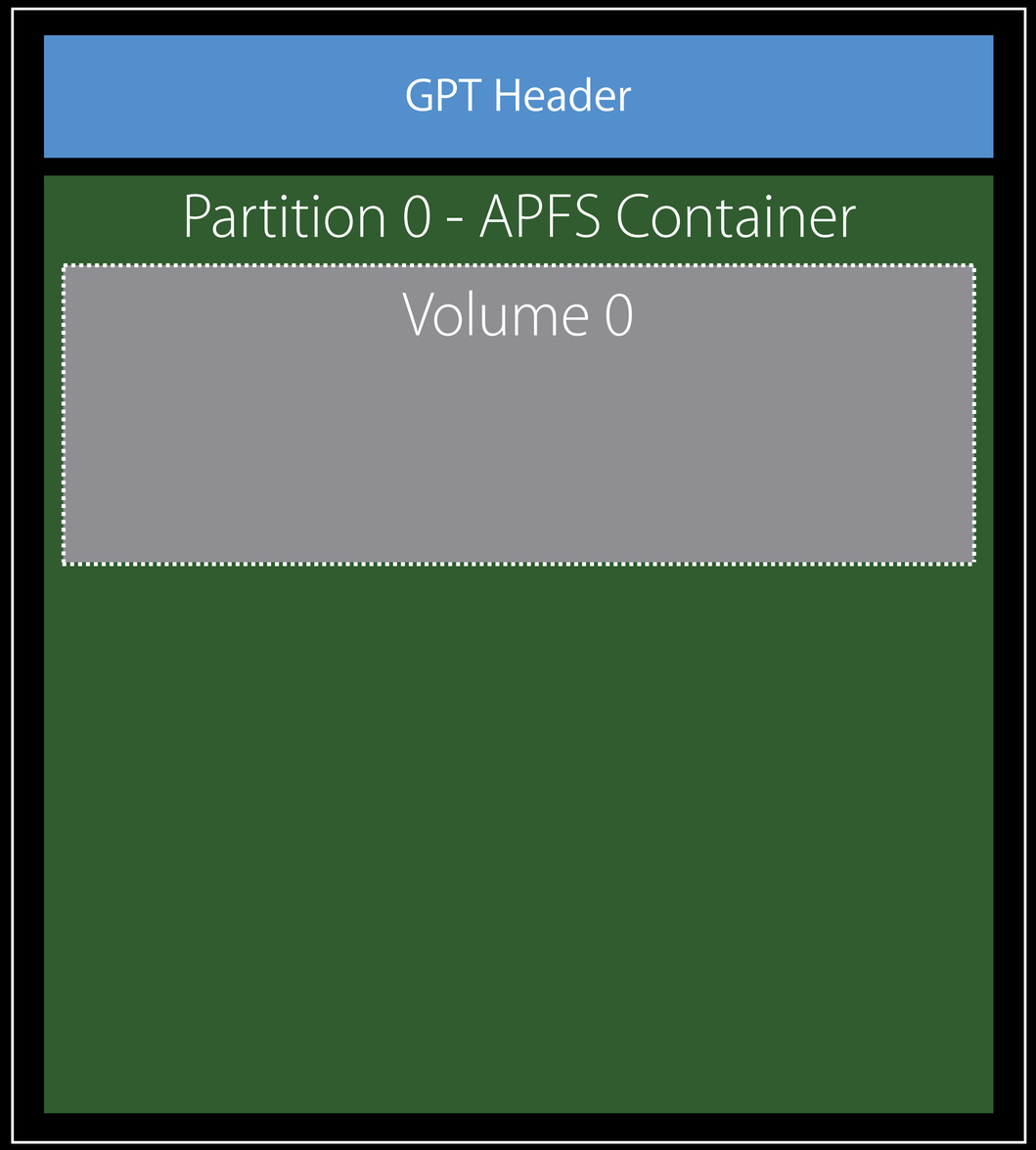
Space Sharing



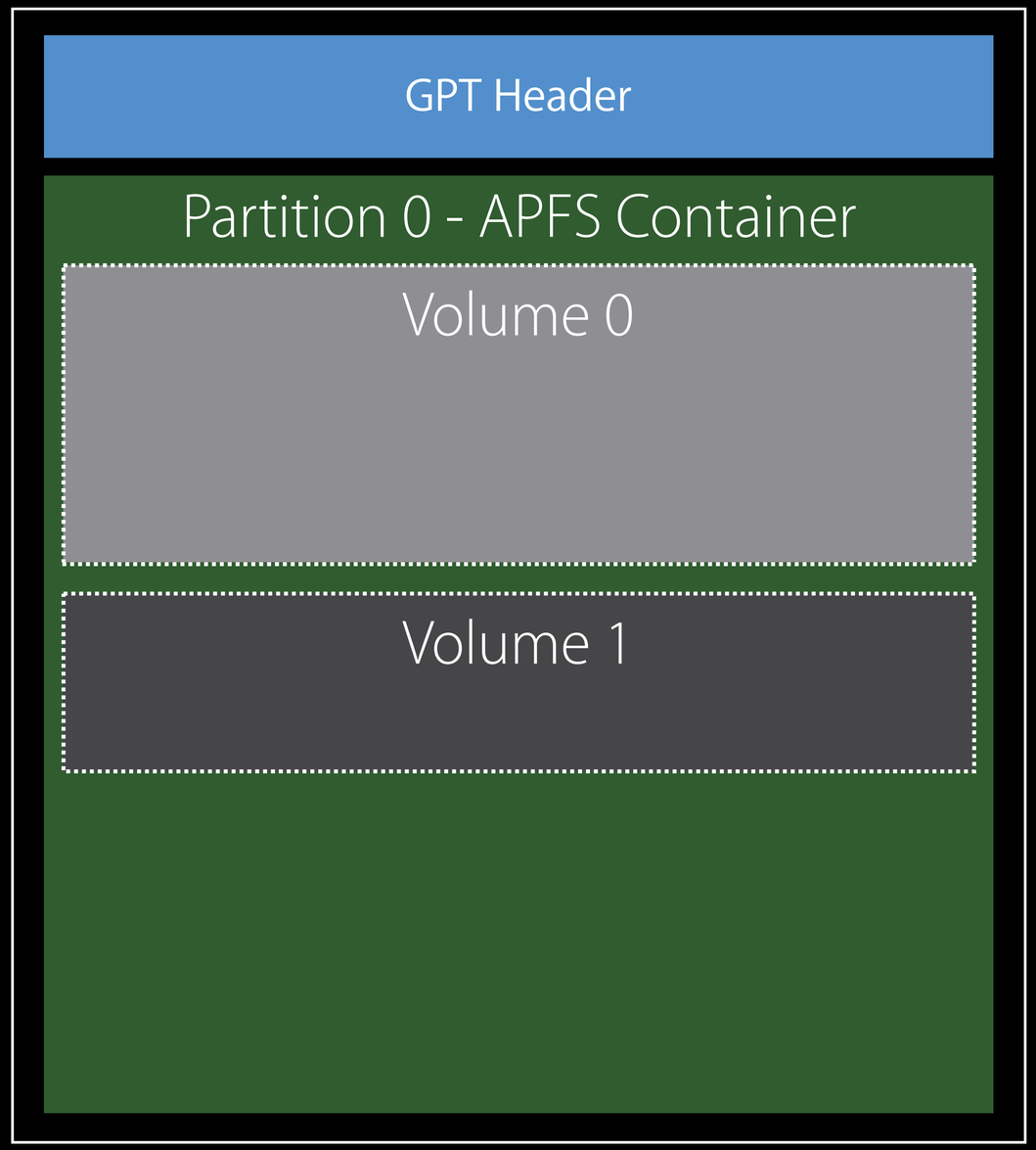
Space Sharing



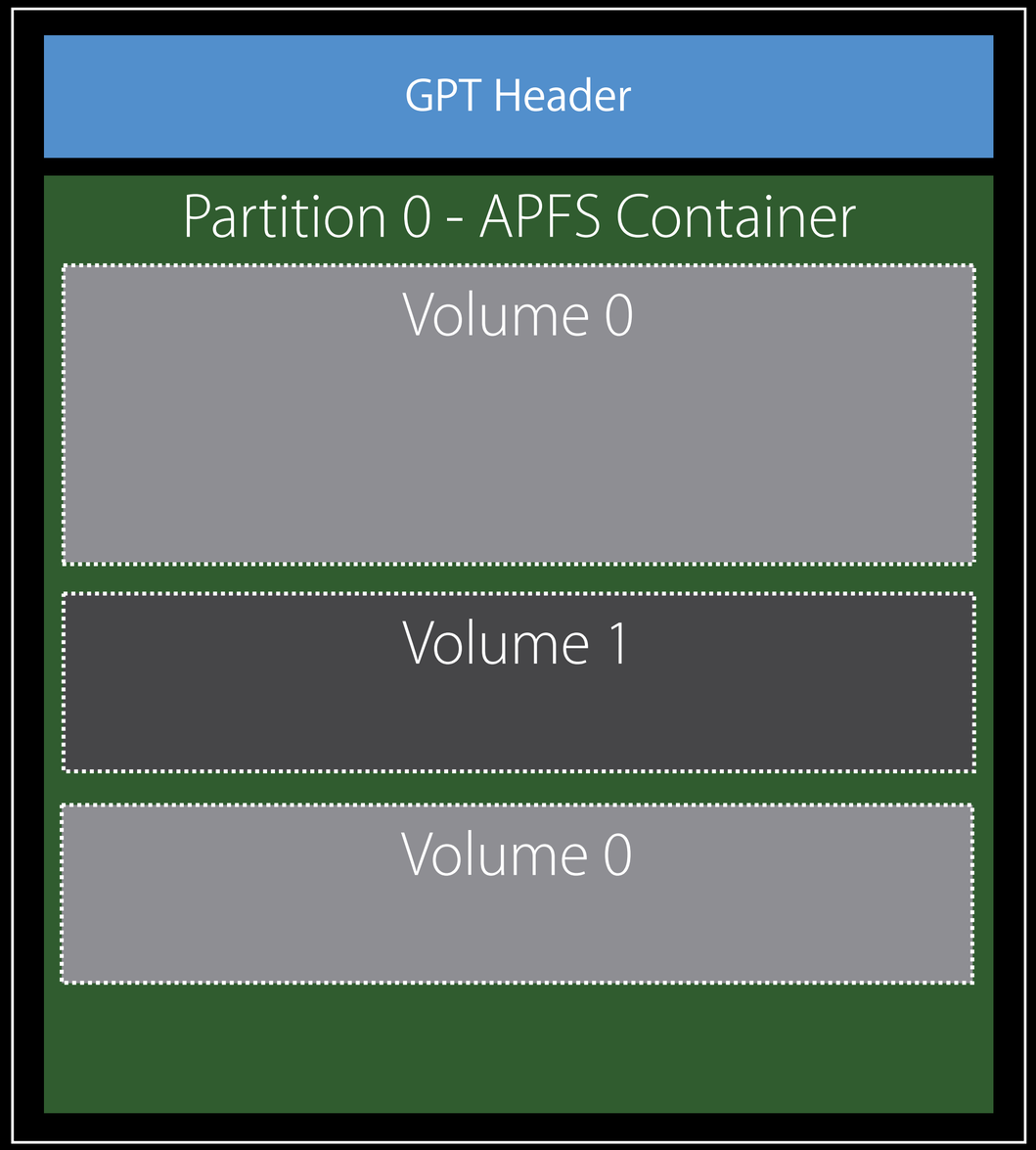
Space Sharing



Space Sharing



Space Sharing



What is Apple File System?

Improved file system fundamentals

HFS compatibility

Space sharing

Cloning files and directories

Snapshots (and reversions)

Fast directory sizing

Atomic safe-save primitives

Encryption

What is Apple File System?

Improved file system fundamentals

HFS compatibility

Space sharing

Cloning files and directories

Snapshots (and reversions)

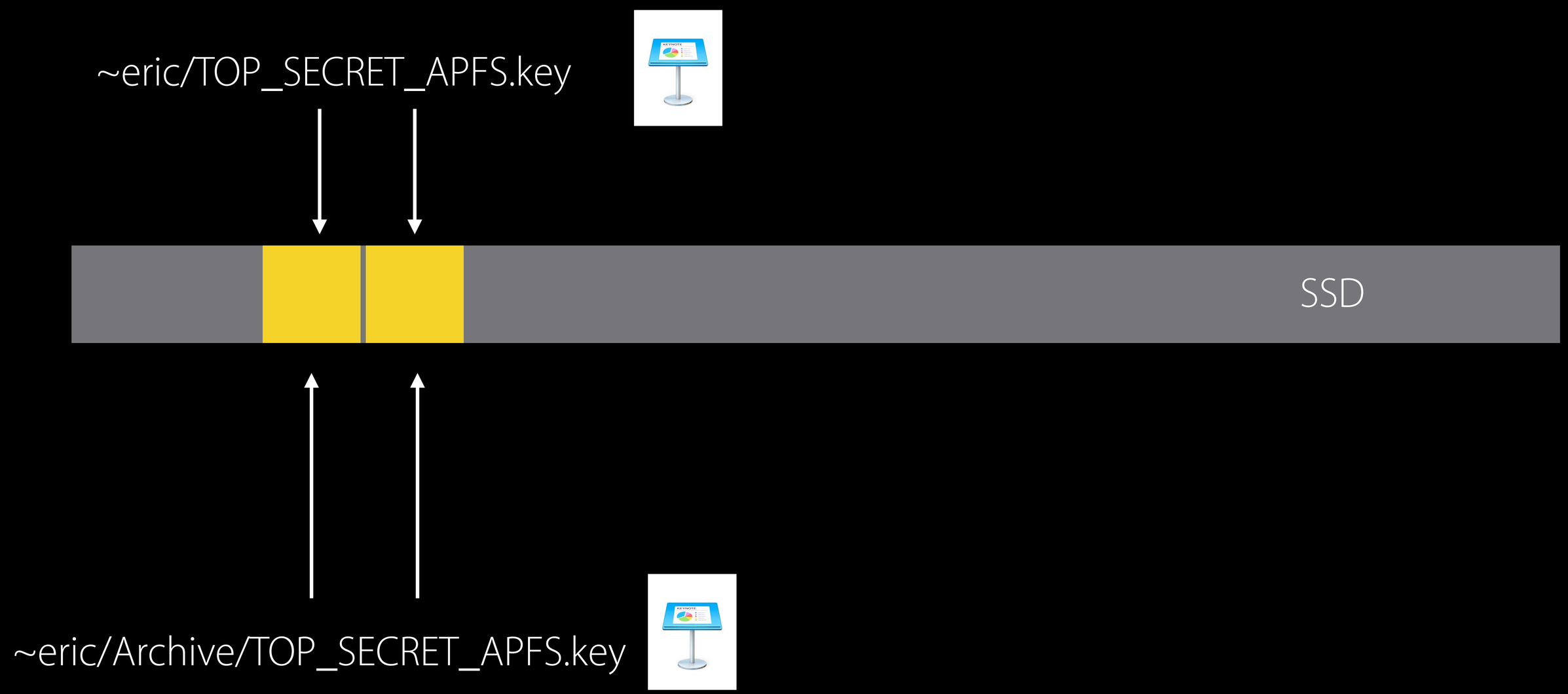
Fast directory sizing

Atomic safe-save primitives

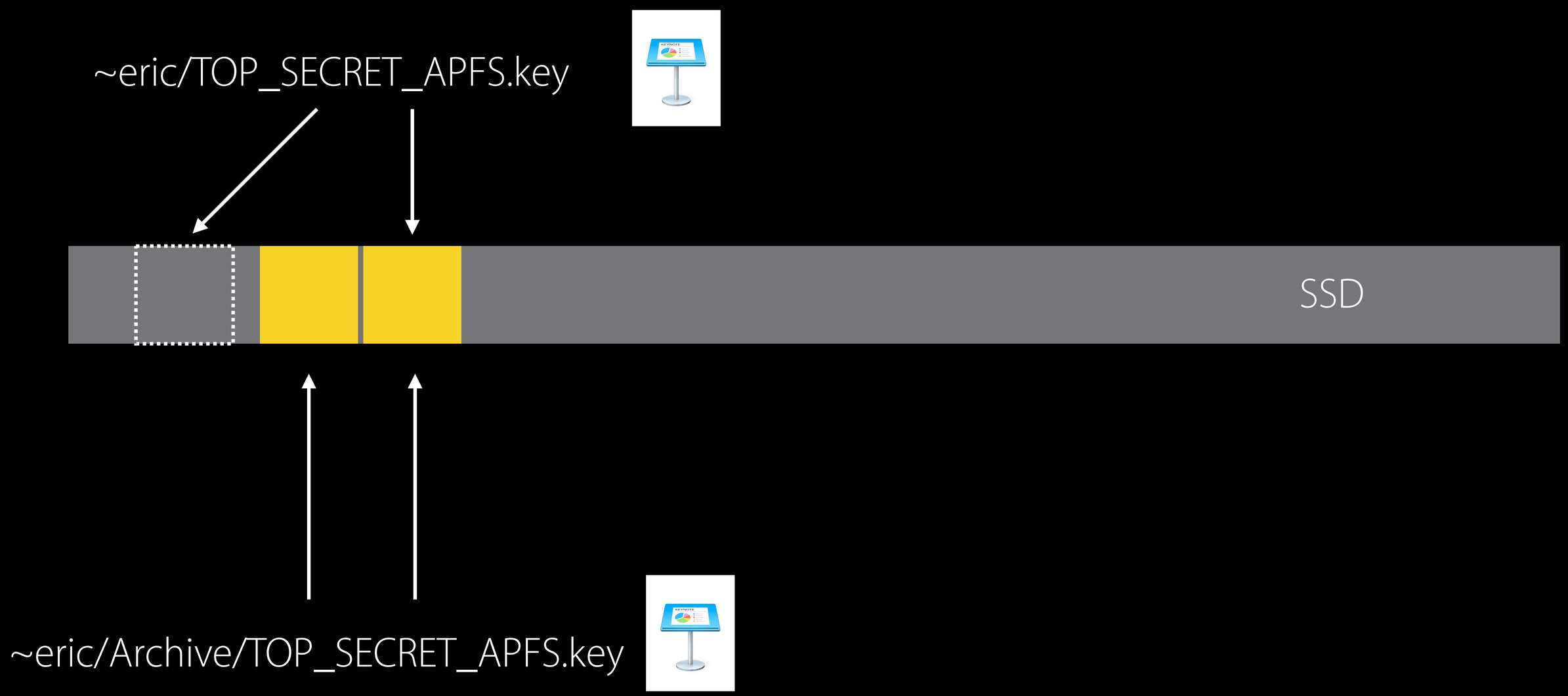
Encryption

Cloning Files and Directories

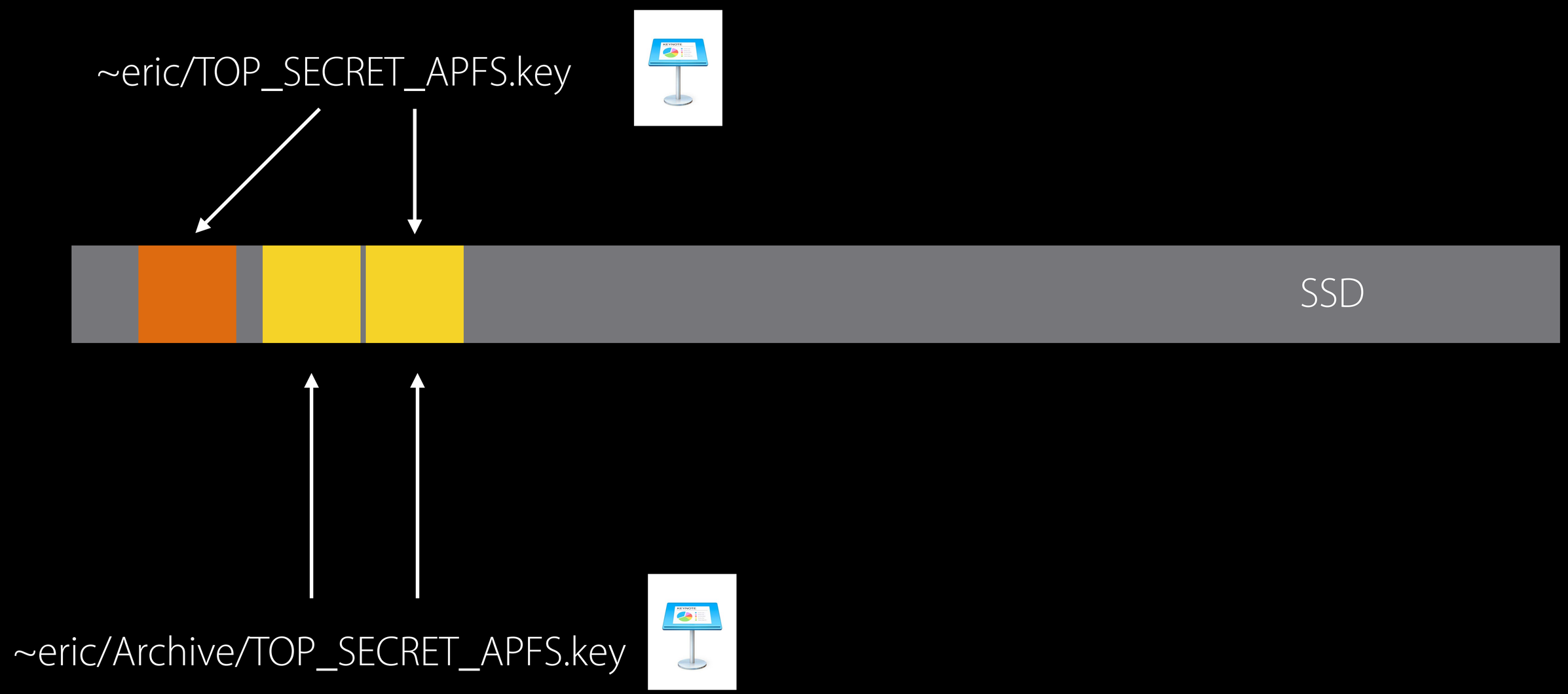
Cloning Files and Directories



Cloning Files and Directories



Cloning Files and Directories



What is Apple File System?

Improved file system fundamentals

HFS compatibility

Space sharing

Cloning files and directories

Snapshots (and reversions)

Fast directory sizing

Atomic safe-save primitives

Encryption

What is Apple File System?

Improved file system fundamentals

HFS compatibility

Space sharing

Cloning files and directories

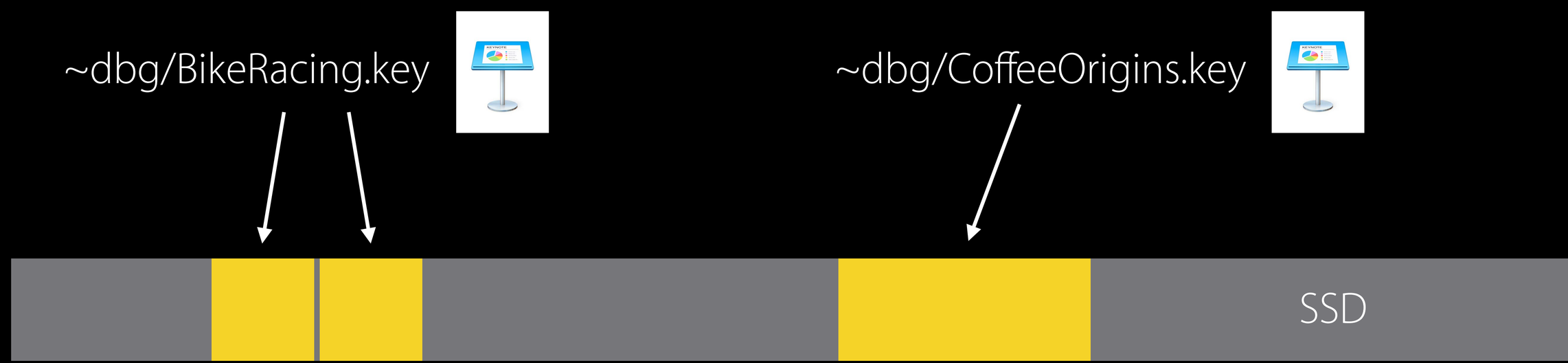
Snapshots (and reversions)

Fast directory sizing

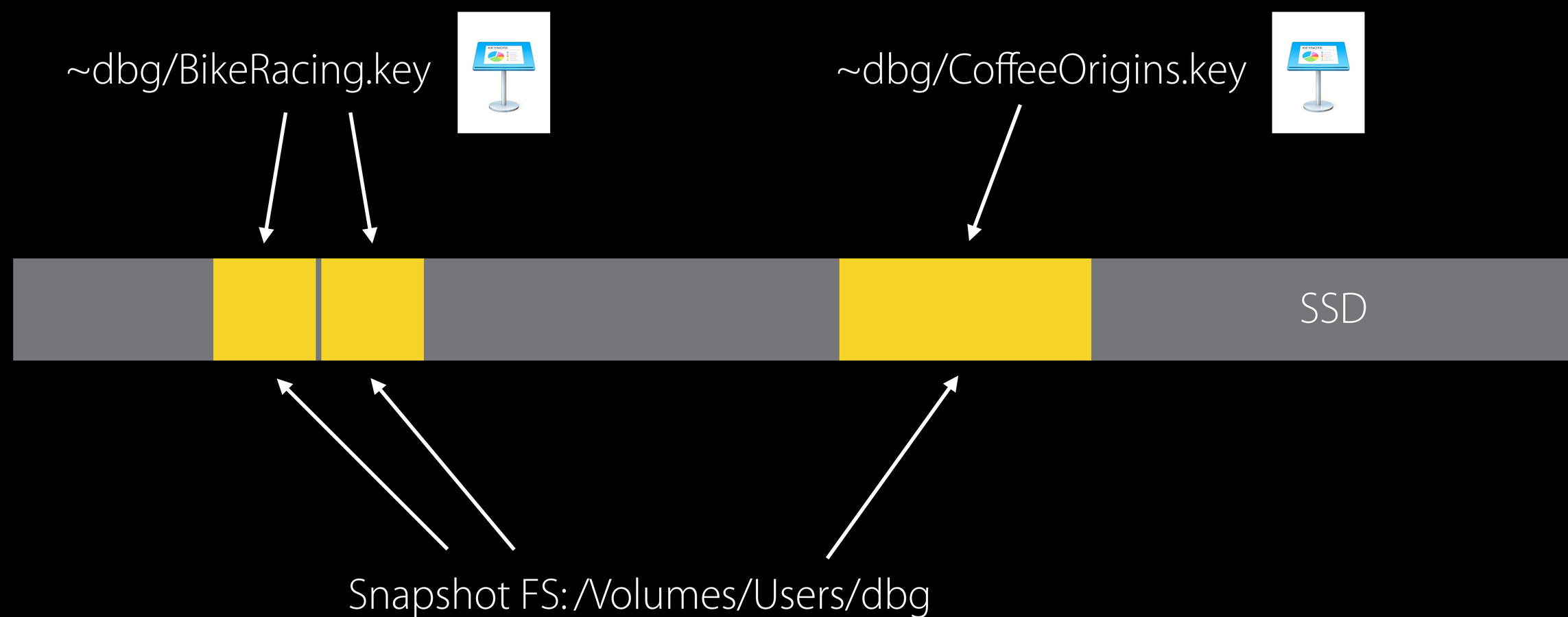
Atomic safe-save primitives

Encryption

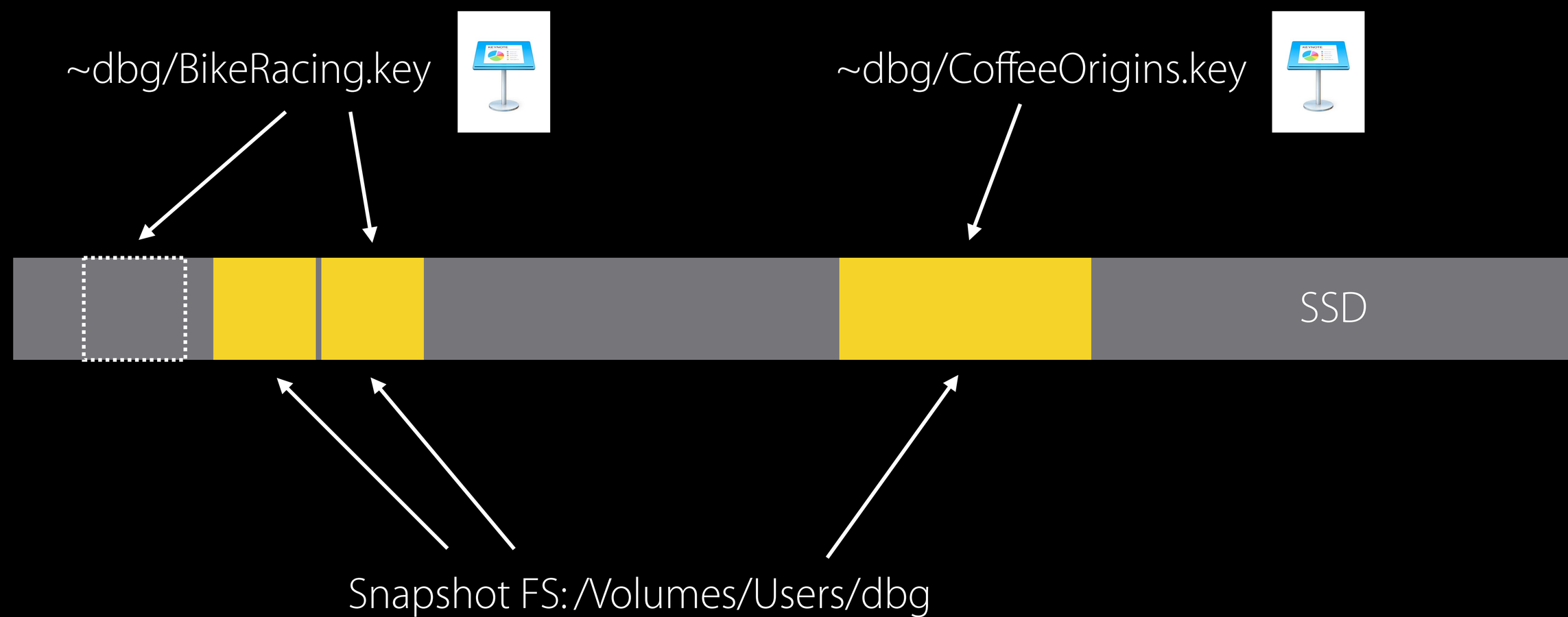
File System Snapshots



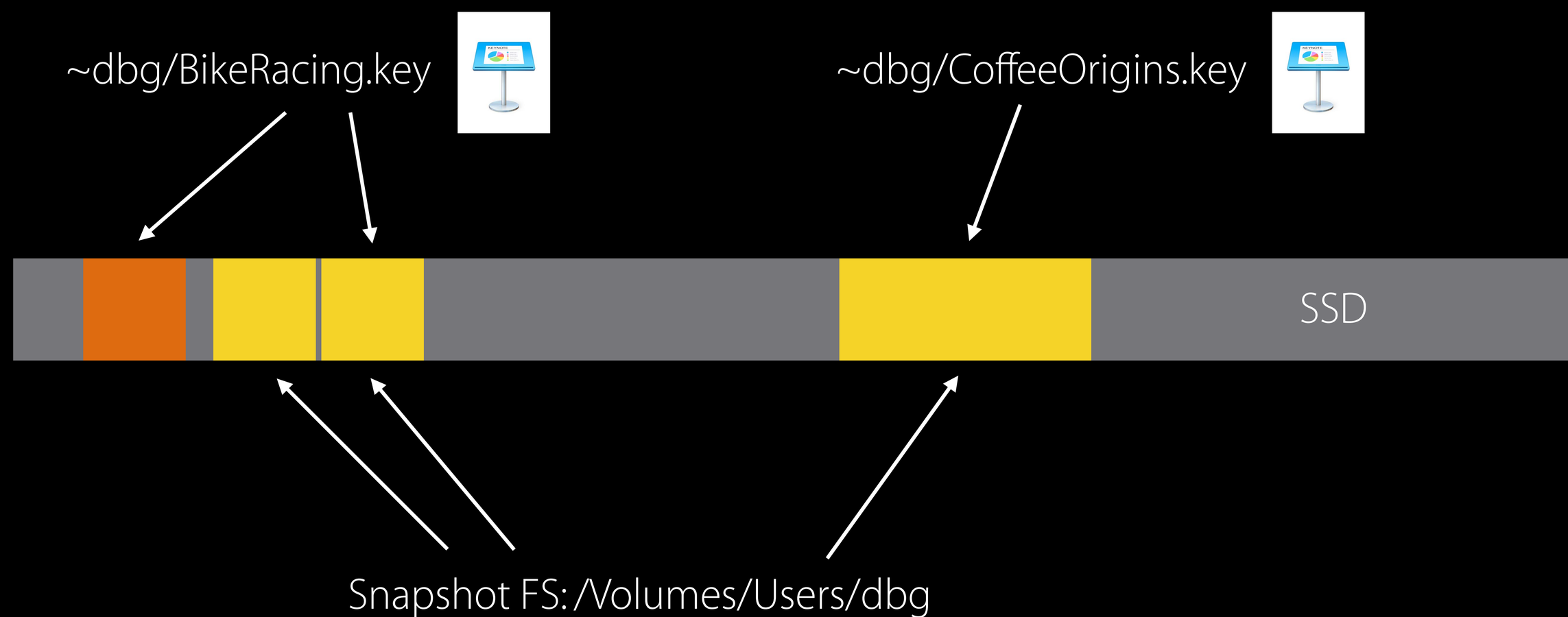
File System Snapshots



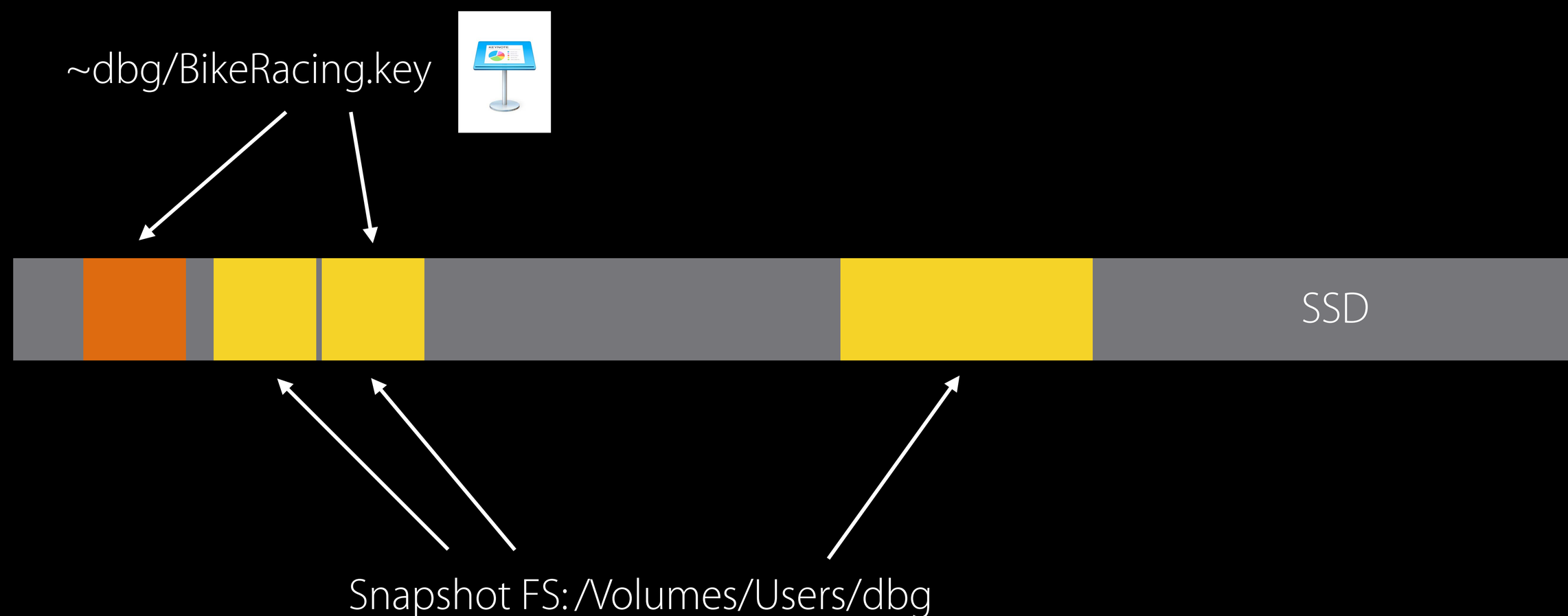
File System Snapshots



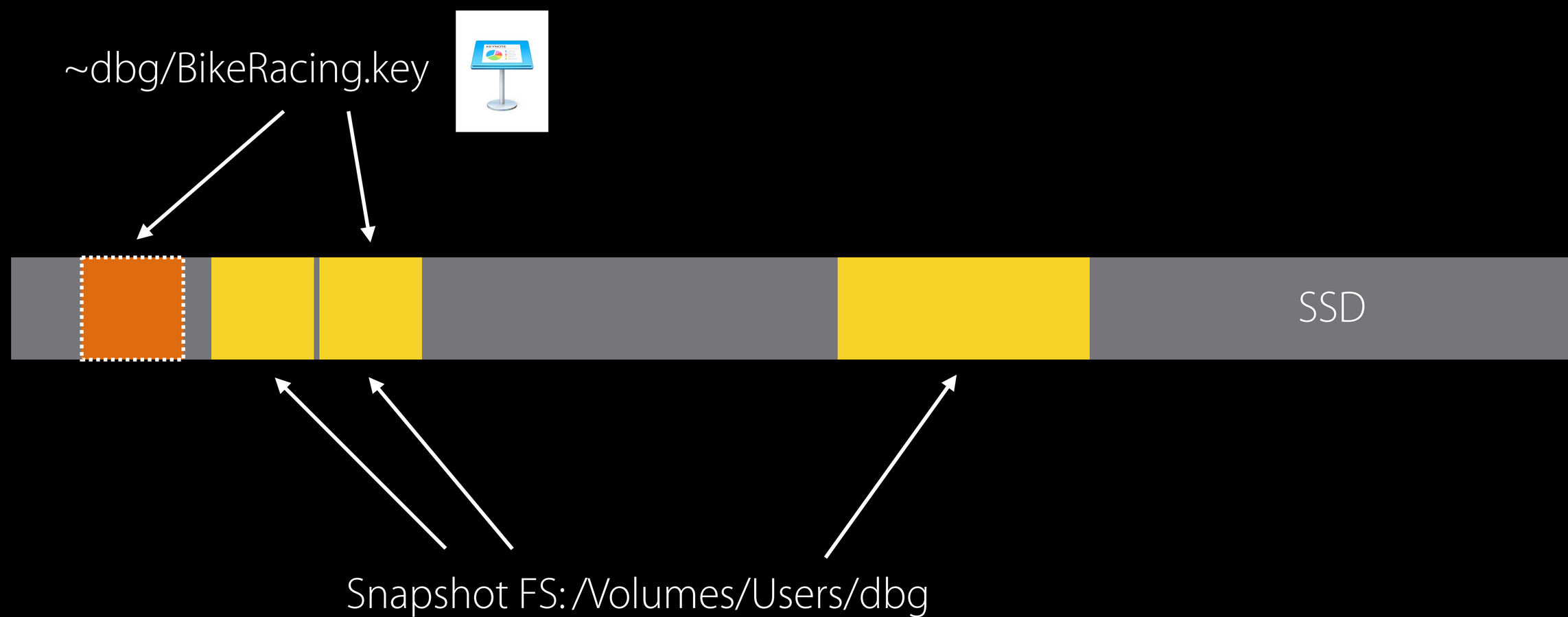
File System Snapshots



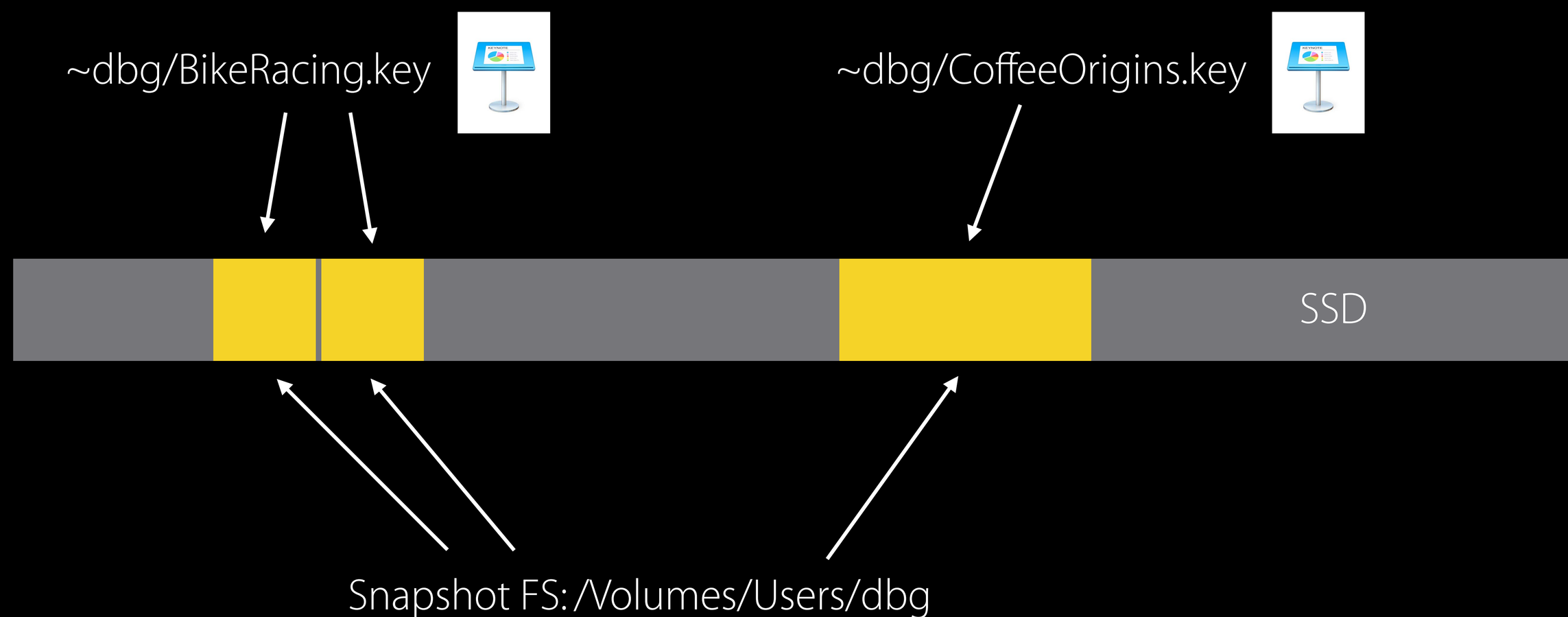
File System Snapshots



Reverting to a Snapshot



Reverting to a Snapshot



What is Apple File System?

Improved file system fundamentals

HFS compatibility

Space sharing

Cloning files and directories

Snapshots (and reversions)

Fast directory sizing

Atomic safe-save primitives

Encryption

What is Apple File System?

Improved file system fundamentals

HFS compatibility

Space sharing

Cloning files and directories

Snapshots (and reversions)

Fast directory sizing

Atomic safe-save primitives

Encryption

Fast Directory Sizing

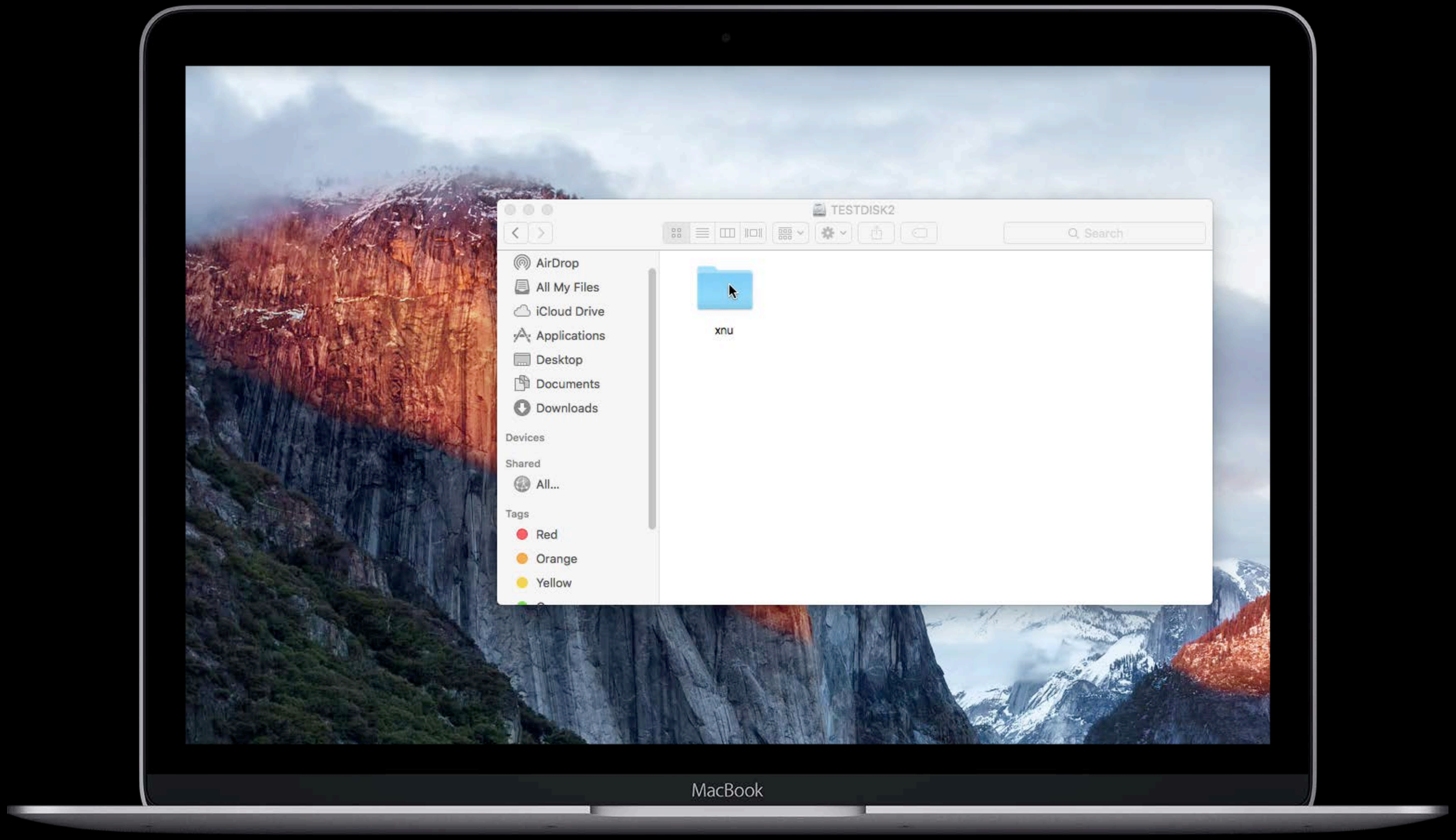
Fast Directory Sizing

How much space does a directory hierarchy use?

Fast Directory Sizing

How much space does a directory hierarchy use?

Users would like to know the answer quickly


TESTDISK2
Search
AirDrop
All My Files
iCloud Drive
Applications
Desktop
Documents
Downloads
Devices
Shared
All...
Tags
Red
Orange
Yellow
xnu
MacBook

Fast Directory Sizing

Fast Directory Sizing

The file system could keep track of this…

Fast Directory Sizing

The file system could keep track of this…

But keeping track in the file system has one main issue:

Fast Directory Sizing

The file system could keep track of this…

But keeping track in the file system has one main issue:

- How do you safely update your parent and its parent (and so on…)

Fast Directory Sizing

The file system could keep track of this…

But keeping track in the file system has one main issue:

- How do you safely update your parent and its parent (and so on…)
- Locking child -> parent is a locking order violation in file systems

Fast Directory Sizing

The file system could keep track of this…

But keeping track in the file system has one main issue:

- How do you safely update your parent and its parent (and so on…)
- Locking child -> parent is a locking order violation in file systems

APFS side-steps the problem!

Fast Directory Sizing

The file system could keep track of this…

But keeping track in the file system has one main issue:

- How do you safely update your parent and its parent (and so on…)
- Locking child -> parent is a locking order violation in file systems

APFS side-steps the problem!

- Store the size separately

Fast Directory Sizing

The file system could keep track of this…

But keeping track in the file system has one main issue:

- How do you safely update your parent and its parent (and so on…)
- Locking child -> parent is a locking order violation in file systems

APFS side-steps the problem!

- Store the size separately
- Use atomic operations to update the size

Fast Directory Sizing

The file system could keep track of this…

But keeping track in the file system has one main issue:

- How do you safely update your parent and its parent (and so on…)
- Locking child -> parent is a locking order violation in file systems

APFS side-steps the problem!

- Store the size separately
- Use atomic operations to update the size
- Small incremental cost (extra records)

What is Apple File System?

Improved file system fundamentals

HFS compatibility

Space sharing

Cloning files and directories

Snapshots (and reversions)

Fast directory sizing

Atomic safe-save primitives

Encryption

What is Apple File System?

Improved file system fundamentals

HFS compatibility

Space sharing

Cloning files and directories

Snapshots (and reversions)

Fast directory sizing

Atomic safe-save primitives

Encryption

Atomic Safe-Save (rename)

~dbg/MakeMoneyFast.key

Atomic Safe-Save (rename)

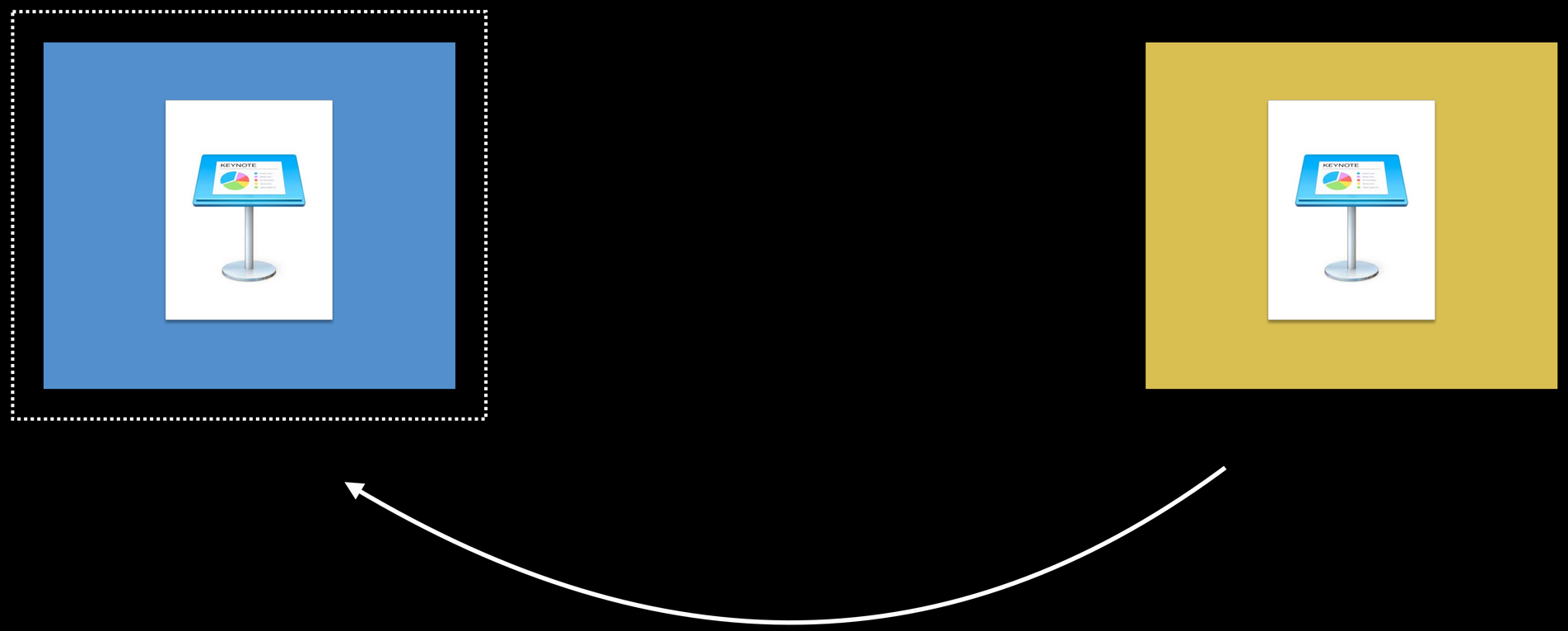
~dbg/MakeMoneyFast.key

/var/tmp/MakeMoneyFast.key

Atomic Safe-Save (rename)

~dbg/MakeMoneyFast.key

/var/tmp/MakeMoneyFast.key



Atomic Safe-Save (rename)

~dbg/MakeMoneyFast.key

Non-Atomic Safe-Save (directory)

~dbg/ClutchConcertReview.rtfd

Non-Atomic Safe-Save (directory)

~dbg/ClutchConcertReview.rtfd

/var/tmp/ClutchConcertReview.rtfd

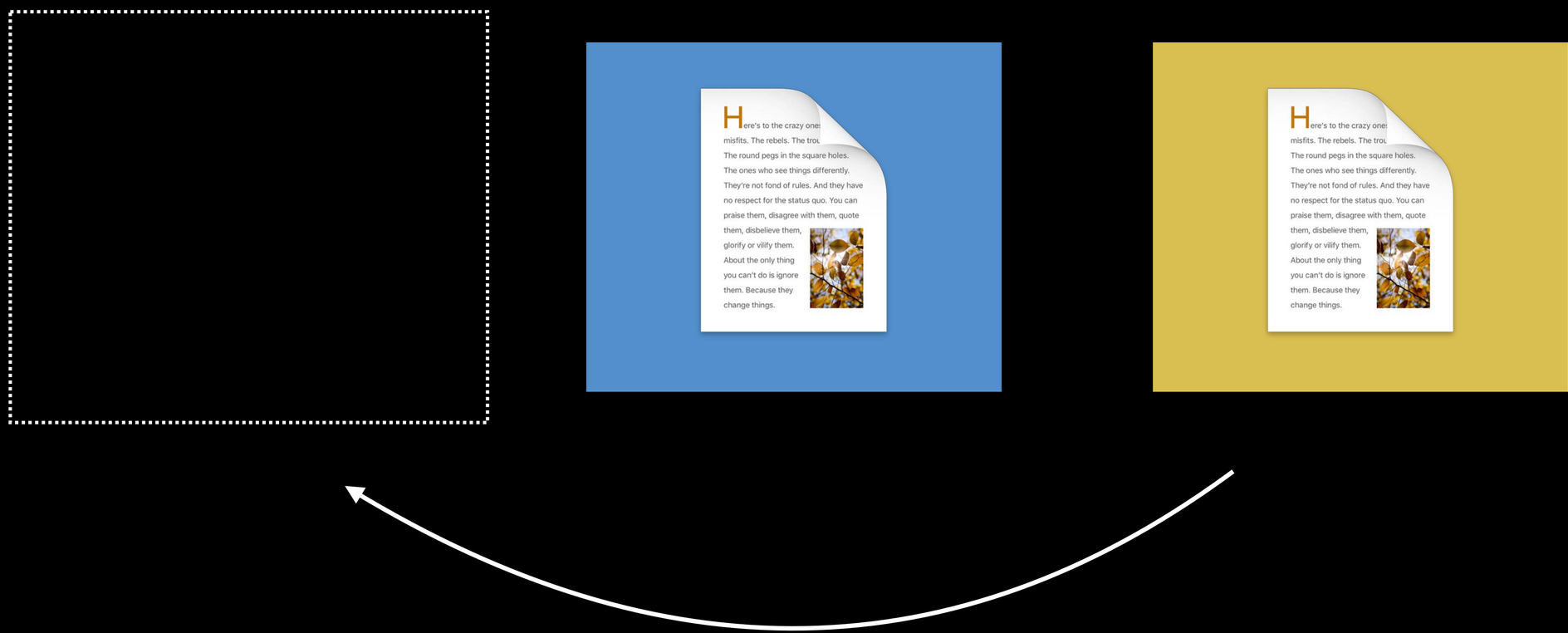
Non-Atomic Safe-Save (directory)

~dbg/ClutchConcertReview.rtfd

/var/tmp/ClutchConcertReview.rtfd

Non-Atomic Safe-Save (directory)

~dbg/ClutchConcertReview.rtfd



/var/tmp/ClutchConcertReview.rtfd

Non-Atomic Safe-Save (directory)

~dbg/ClutchConcertReview.rtfd

Non-Atomic Safe-Save (directory)

~dbg/ClutchConcertReview.rtfd

Atomic Safe-Save (renamex_np)

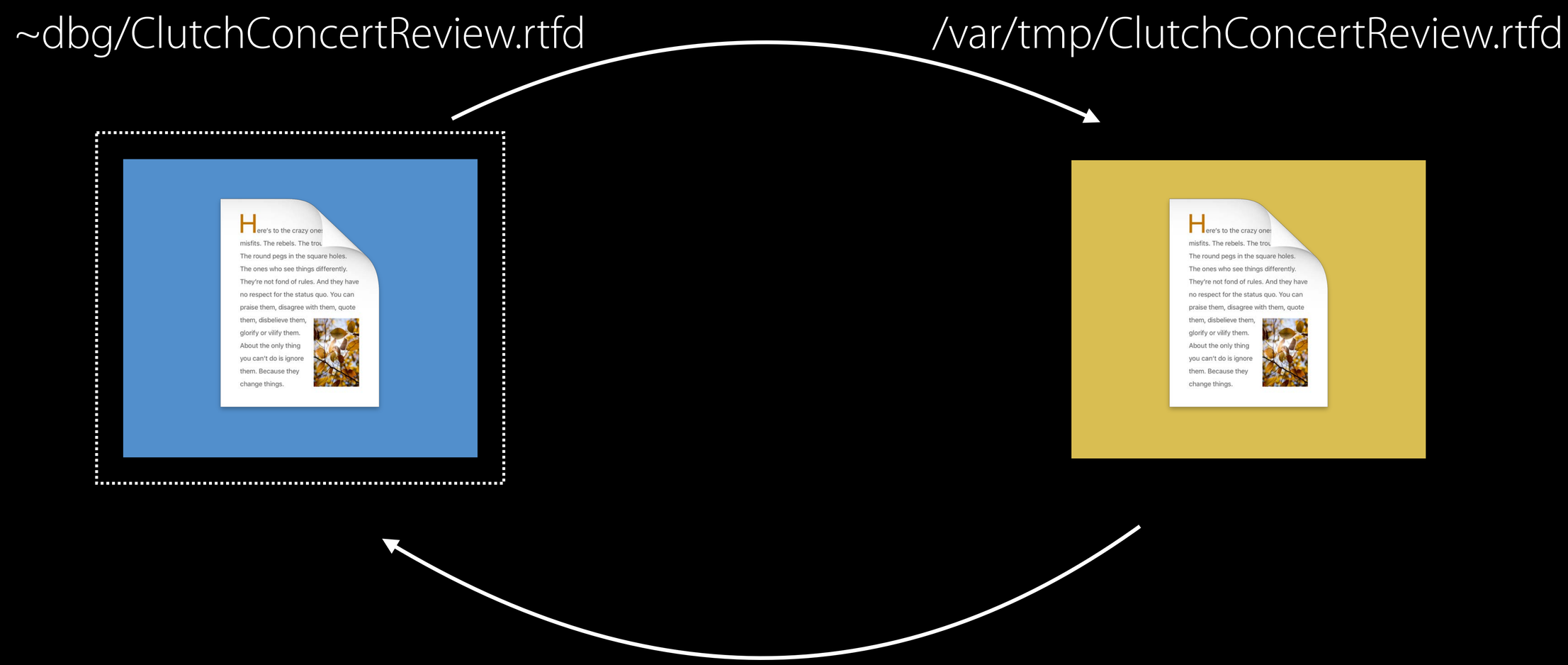
~dbg/ClutchConcertReview.rtfd

Atomic Safe-Save (renamex_np)

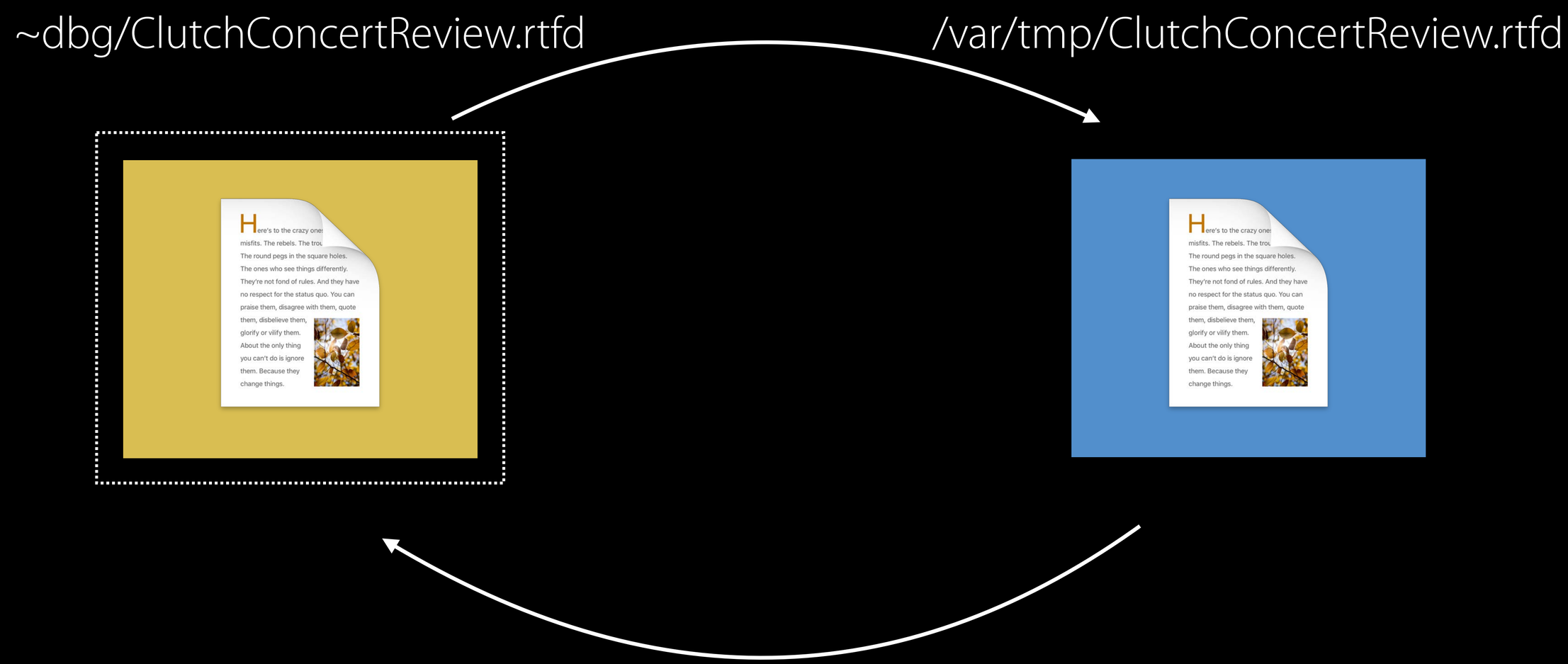
~dbg/ClutchConcertReview.rtfd

/var/tmp/ClutchConcertReview.rtfd

Atomic Safe-Save (renamex_np)



Atomic Safe-Save (renamex_np)



Atomic Safe-Save (renamex_np)

~dbg/ClutchConcertReview.rtfd

What is Apple File System?

Improved file system fundamentals

HFS compatibility

Space sharing

Cloning files and directories

Snapshots (and reversions)

Fast directory sizing

Atomic safe-save primitives

Encryption

What is Apple File System?

Improved file system fundamentals

HFS compatibility

Space sharing

Cloning files and directories

Snapshots (and reversions)

Fast directory sizing

Atomic safe-save primitives

Encryption

Encryption (HFS+)

Encryption (HFS+)

HFS+ relies on CoreStorage to provide Full Disk Encryption on Macs

Encryption (HFS+)

HFS+ relies on CoreStorage to provide Full Disk Encryption on Macs

iOS uses an HFS+ variant that supports per-file keys in conjunction with accelerated AES hardware

Encryption (APFS)

Encryption (APFS)

APFS supports multiple levels of file system encryption

Encryption (APFS)

APFS supports multiple levels of file system encryption

- No encryption

Encryption (APFS)

APFS supports multiple levels of file system encryption

- No encryption
- One key per Volume (metadata and data)

Encryption (APFS)

APFS supports multiple levels of file system encryption

- No encryption
- One key per Volume (metadata and data)
- Multi-Key Encryption

Encryption (APFS)

APFS supports multiple levels of file system encryption

- No encryption
- One key per Volume (metadata and data)
- Multi-Key Encryption
  - Metadata Encryption

Encryption (APFS)

APFS supports multiple levels of file system encryption

- No encryption
- One key per Volume (metadata and data)
- Multi-Key Encryption
  - Metadata Encryption
  - Per-File Encryption

Encryption (APFS)

APFS supports multiple levels of file system encryption

- No encryption
- One key per Volume (metadata and data)
- Multi-Key Encryption
  - Metadata Encryption
  - Per-File Encryption
  - Per-Extent Encryption

Encryption (APFS)

APFS supports multiple levels of file system encryption

- No encryption
- One key per Volume (metadata and data)
- Multi-Key Encryption
  - Metadata Encryption
  - Per-File Encryption
  - Per-Extent Encryption

APFS unifies the file system encryption software across all of our platforms

Demo

Cloning and Snapshots / APFS on macOS 10.12

What is Apple File System?

Introduction / Motivation

New Features

Demo

New APIs

What is Apple File System?

Introduction / Motivation

New Features

Demo

New APIs

Enhanced APIs

Foundation / FileManager (Swift)

Automatically adopts our new cloning and safe-save behavior

```
func copyItem(atPath srcPath: String, toPath dstPath: String) throws

func replaceItem(at originalItemURL: URL, withItemAt newItemURL: URL,
    backupItemName backupItemName: String?, options options:
    FileManager.ItemReplacementOptions = [], resultingItemURL resultingURL:
    AutoreleasingUnsafeMutablePointer<NSURL?>?) throws
```

Enhanced APIs

libcopyfile

CoreOS library for copying deep hierarchies—supports cloning!

Slightly above the POSIX layer

New flags added

```
#include <copyfile.h>


int copyfile(const char *from, const char *to,
    copyfile_state_t state, copyfile_flags_t flags);

int fcopyfile(int from_fd, int to_fd, copyfile_state_t,
    copyfile_flags_t flags);

new flag bit: COPYFILE_CLONE
Equivalent to (COPYFILE_EXCL | COPYFILE_ACL |
     COPYFILE_STAT | COPYFILE_XATTR | COPYFILE_DATA)
```

New APIs

Safe-Save APIs

New system calls

```
#include <stdio.h>

int renamex_np(const char *, const char *, unsigned int)
int renameatx_np(int, const char *, int, const char *, unsigned int)
```

New APIs

Cloning APIs

New file/directory cloning system calls

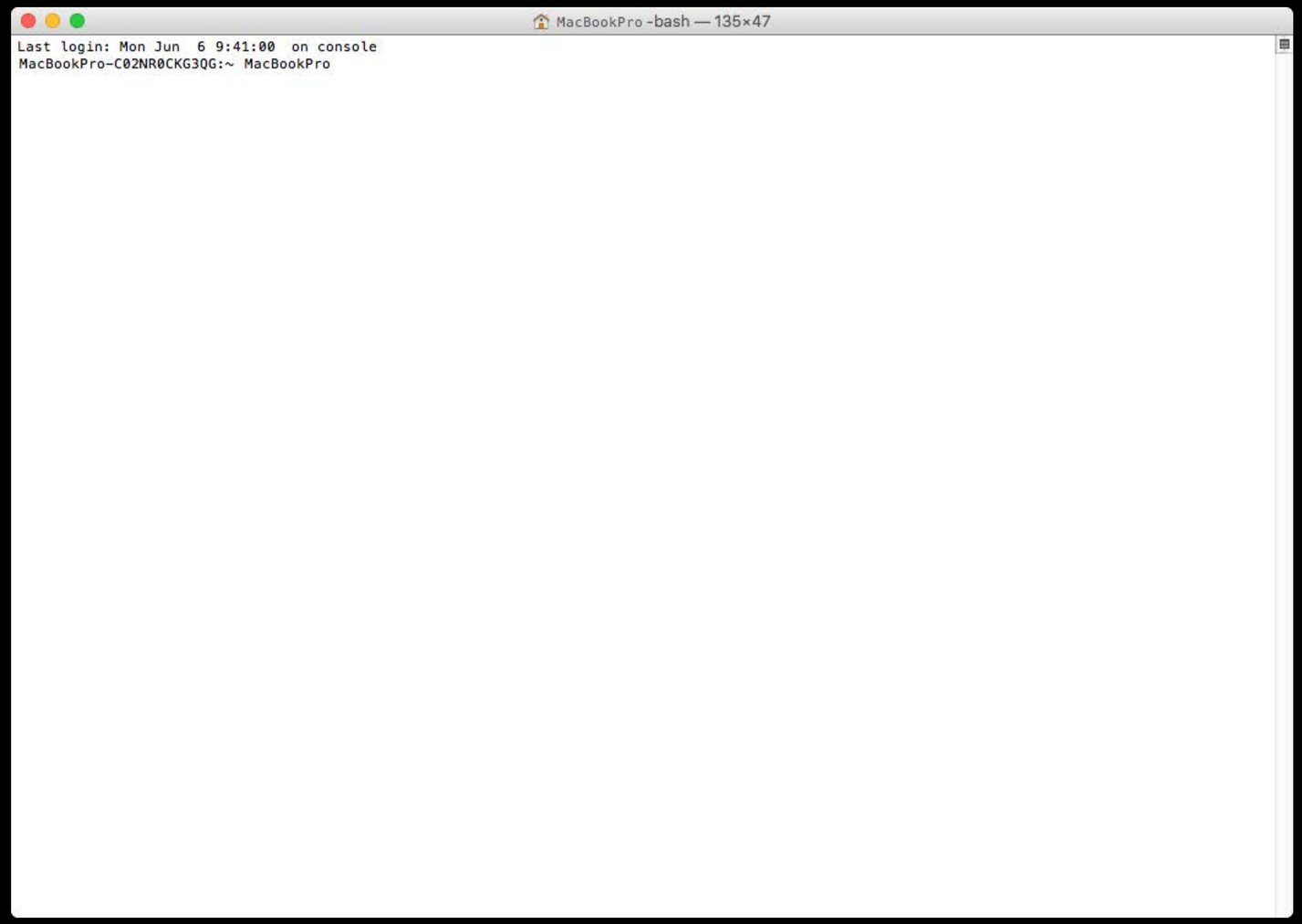
```
#include <sys/attr.h>
#include <sys/clonefile.h>

int clonefileat(int, const char *, int, const char *, uint32_t);

int fclonefileat(int, int, const char *, uint32_t);

int clonefile(const char *, const char *, uint32_t);
```

Compatibility



Compatibility

MacBookPro -bash — 135×47

```
Last login: Mon Jun  6 9:41:00  on console
MacBookPro-C02NR0CKG3QG:~ MacBookPro
```

hdiutil (disk image tool)

```
hdiutil create -fs APFS -size 1GB foo.sparseimage
```

Compatibility

MacBookPro -bash — 135×47

```
Last login: Mon Jun  6 9:41:00  on console
MacBookPro-C02NR0CKG3QG:~ MacBookPro
```

```
hdiutil  (disk image tool)
    hdiutil create -fs APFS -size 1GB foo.sparseimage
diskutil apfs …
    diskutil apfs createContainer /dev/disk1s1
    diskutil apfs addVolume disk1s1 APFS newAPFS
```

Compatibility

```
Last login: Mon Jun  6 9:41:00  on console
MacBookPro-C02NR0CKG3QG:~ MacBookPro

hdiutil  (disk image tool)
    hdiutil create -fs APFS -size 1GB foo.sparseimage
diskutil apfs …
    diskutil apfs createContainer /dev/disk1s1
    diskutil apfs addVolume disk1s1 APFS newAPFS
fsck_apfs (APFS File System Check/Repair)
```

Current Limitations of APFS in macOS Sierra

Current Limitations of APFS in macOS Sierra

Data volumes only

Current Limitations of APFS in macOS Sierra

Data volumes only

Time Machine backups with APFS

Current Limitations of APFS in macOS Sierra

Data volumes only

Time Machine backups with APFS

FileVault / Fusion Drive Support

Current Limitations of APFS in macOS Sierra

Data volumes only

Time Machine backups with APFS

FileVault / Fusion Drive Support

Case-sensitive

Compatibility

Compatibility

APFS cannot be shared over AFP (Use SMB instead)

Compatibility

APFS cannot be shared over AFP (Use SMB instead)

OS X Yosemite or earlier will not recognize Apple File System volumes

Developer Preview available in macOS Sierra 10.12

Rollout Plan

Upgrading to APFS

Upgrading to APFS

Apple will provide an in-place upgrade path for HFS+ to APFS

Upgrading to APFS

Apple will provide an in-place upgrade path for HFS+ to APFS

User data remains in place

Upgrading to APFS

Apple will provide an in-place upgrade path for HFS+ to APFS

User data remains in place

Write the new APFS metadata into HFS+'s free space

Shipping in 2017

Summary

Summary

APFS will be the default file system for all Apple products in 2017

Summary

APFS will be the default file system for all Apple products in 2017

Ultra-modern, crash-protected, space-sharing

Summary

APFS will be the default file system for all Apple products in 2017

Ultra-modern, crash-protected, space-sharing

Supports cloning, snapshots, enhanced data security features

Summary

APFS will be the default file system for all Apple products in 2017

Ultra-modern, crash-protected, space-sharing

Supports cloning, snapshots, enhanced data security features

Tuned and designed for the Apple ecosystem

More Information

https://developer.apple.com/wwdc16/701

Takeaways

Takeaways

APFS is coming soon

Takeaways

APFS is coming soon

Please test your apps against APFS with the WWDC macOS build (and run them on APFS)

Takeaways

APFS is coming soon

Please test your apps against APFS with the WWDC macOS build (and run them on APFS)

Please report any bugs you encounter via bugreporter so we can investigate

Related Sessions

How iOS Security Really Works	Nob Hill	Tuesday 4:00PM

Labs

File Systems Lab	Frameworks Lab C	Tuesday 12:30 PM

WWDC16