

What's New in the File System

Session 709

Deric Horn

Engineering Manager, File Systems

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

Agenda

Agenda

File system and frameworks update

Storage industry trends

Recent changes

Best practices

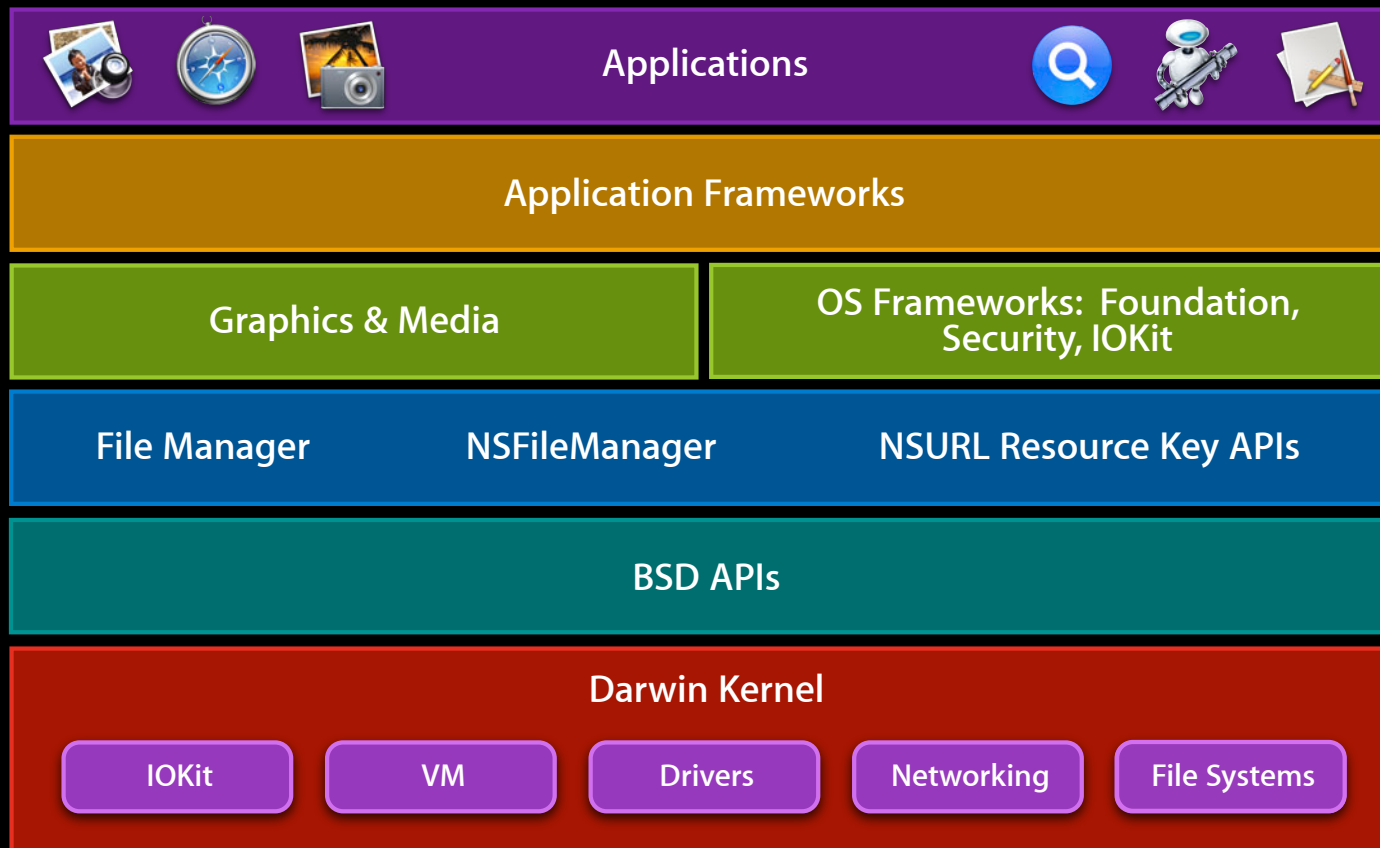
File Vault 2

What it is

How it works

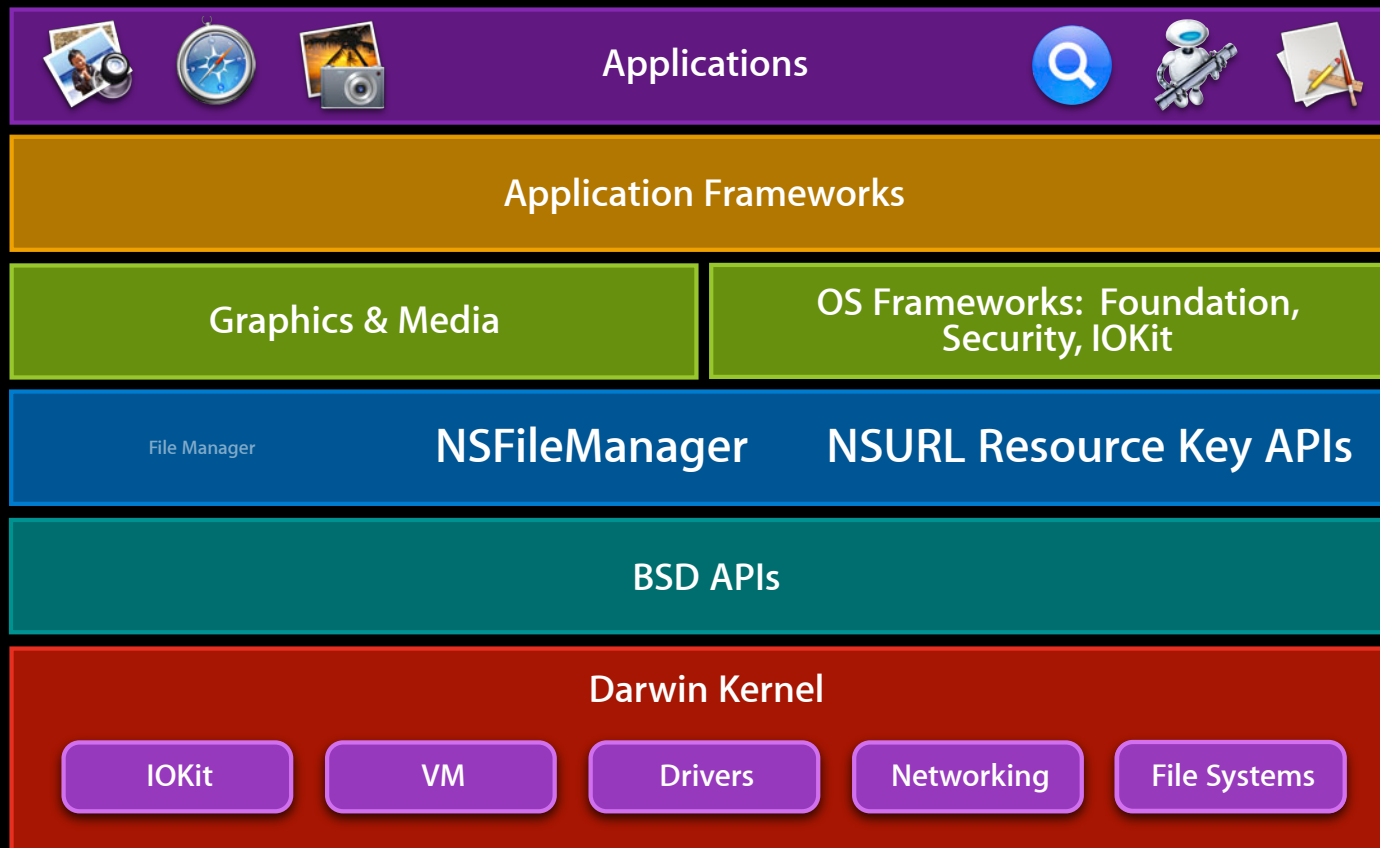
Architecture Overview

File system is ultimately responsible for your data



Architecture Overview

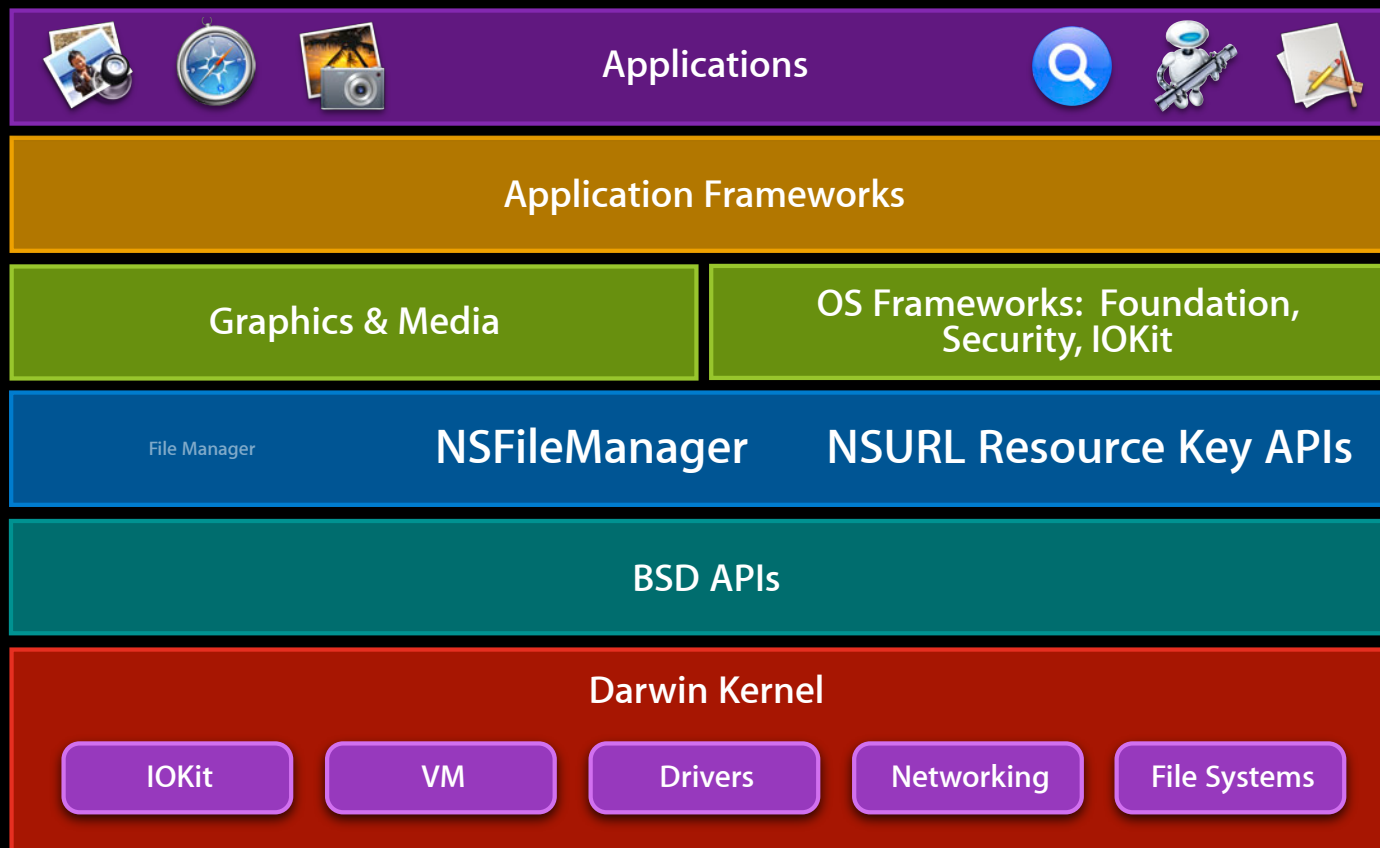
File system is ultimately responsible for your data



Architecture Overview

File Systems

File system is ultimately responsible for your data



/System/Library/Filesystems/

File Systems

Mac OS X 10.8

ExFAT	Read / Write (10.6)
FAT 16/32	Read / Write
HFS	Read Only (Deprecated)
HFS+	Journaled HFS+ everywhere
NTFS	Read Only
UDF	Read / Write

Same Code Base

iOS + OSX

- Journaled HFS+ is everywhere
- Improvements made for one platform improve the other
 - Bug fixes
 - Performance
 - TRIM
- Minor difference
 - iOS devices use case-sensitive HFS+ (HFSX)

Agenda

File system and frameworks update

Storage industry trends

Recent changes

Best practices

File Vault 2

What it is

How it works

Agenda

File system and frameworks update

Storage industry trends

Recent changes

Best practices

File Vault 2

What it is

How it works

Trends in Storage

SSDs

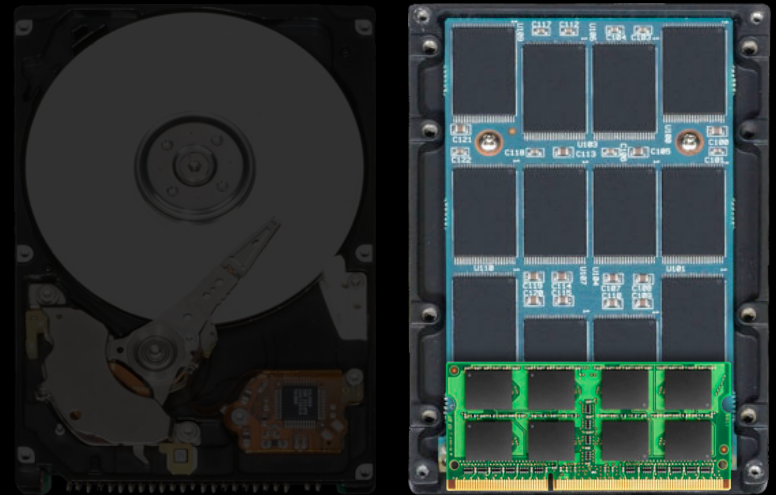
- Small, fast
 - SSD on PCIe, controller integration
 - Better random I/O: 100X faster HDD
- Trending towards fewer write cycles
 - 3000 write cycles and dropping
- Security concerns
 - Spare blocks
 - Wear leveling
- Expensive



Trends in Storage

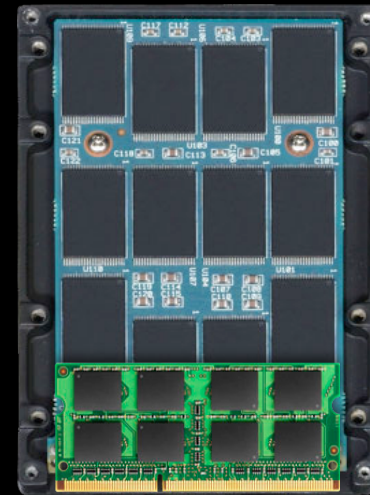
SSDs

- Small, fast
 - SSD on PCIe, controller integration
 - Better random I/O: 100X faster HDD
- Trending towards fewer write cycles
 - 3000 write cycles and dropping
- Security concerns
 - Spare blocks
 - Wear leveling
- Expensive



Trends in Storage

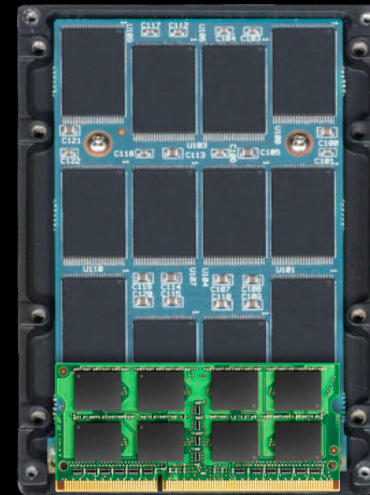
SSDs: Impact on file systems



Trends in Storage

SSDs: Impact on file systems

- Fast random I/O
 - Defragmenting less useful
 - More flexibility in future design decisions
- Trending towards fewer write cycles
 - TRIM support
 - Faster, longer lasting SSDs
 - Large sequential writes better



Trends in Storage

Distributed storage



- Devices store a subset of user's data
- Users getting new devices often (iPhone/iPad/iPod)
- Cloud
 - Secure backup
 - Efficient backup
 - Deduplication

Agenda

File system and frameworks update

Storage industry trends

Recent changes

Best practices

File Vault 2

What it is

How it works

Agenda

File system and frameworks update

Storage industry trends

Recent changes

Best practices

File Vault 2

What it is

How it works

Extended Attributes

- Introduced in OS X 10.4
 - Perfect for storing small application specific data in a file

```
getxattr(path, ptr, attrdatabuf, sizeof(attrdatabuf), 0, 0);
setxattr(path, EName, EValue, strlen(EValue)+1, 0, 0);
listxattr(path, attrnamebuf, nsize, 0);
removexattr(path, EName, 0);
```

- `ls -l@`
 - `com.apple.FinderInfo` 32
 - `com.intuit.TurboTax.2010.FilingStatus` 28
 - Maximum size: <4K
- 10.7
 - Maximum size: **128K**

Extended Attributes

Pros, cons, and guidelines

- Ideal for storing file metadata
 - Author, window position, history, etc.
- Beware of storing critical data in EAs
 - Not all volume formats support extended attributes
 - “._Files”
- Small EAs are packed in Attributes B-Tree
 - Require less than one allocation block

Mounting Network Volumes

Deprecated APIs



FSMountServerVolumeSync(...OptionBits flags)

FSMountServerVolumeAsync(...CFRunLoopRef runloop)



NetFSMountURLSync(...CFMutableDictionaryRef open_options)

NetFSMountURLAsync(...dispatch_queue_t dispatchq)

Mount options replaced by dictionaries

RunLoop and callback replaced by dispatch queue and block

Deprecated APIs

File Manager



FSGetCatalogInfo

FSGetCatalogInfoBulk



CFURLCopyResourcePropertiesForKeys

CFURLEnumerator

**File Manager replaced by NSFileManager and CFURL Resource Key APIs
Also available in iOS 5**

Agenda

File system and frameworks update

Storage industry trends

Recent changes

Best practices

File Vault 2

What it is

How it works

Agenda

File system and frameworks update

Storage industry trends

Recent changes

Best practices

File Vault 2

What it is

How it works

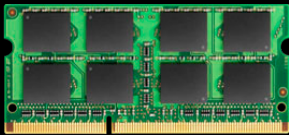
Performance Tips

Getting the most out of the file system

- Throttled I/O
 - Schedule low-priority tasks
 - Spotlight, Time Machine, background encryption...
- Higher-level APIs
 - GCD I/O background queue is throttled

Playing Nicely with Others

How the universal buffer cache works



4 GB

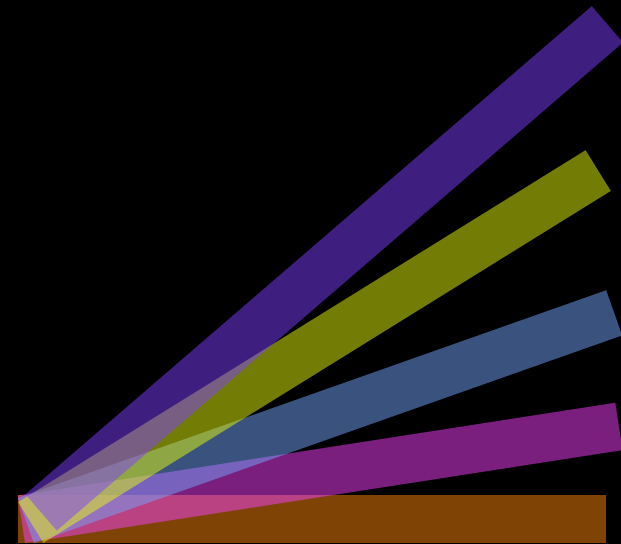
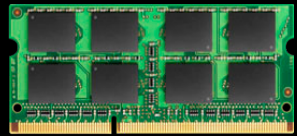


1 TB



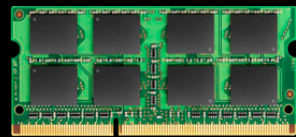
Playing Nicely with Others

How the universal buffer cache works



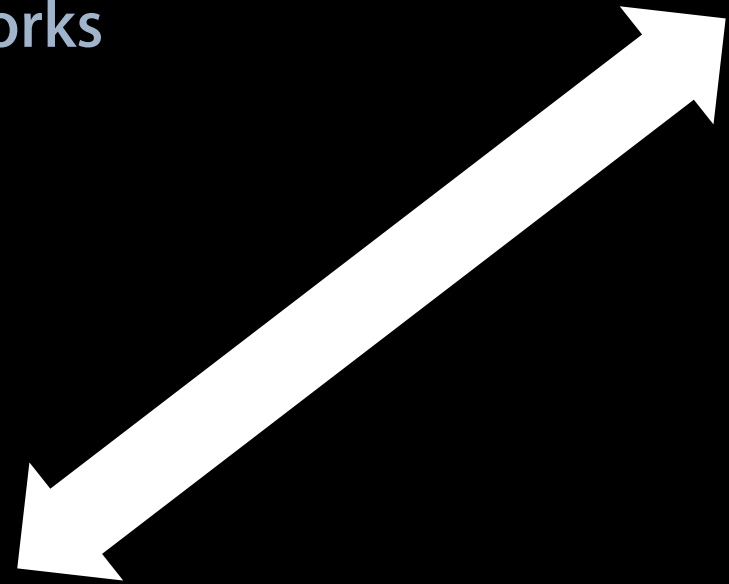
Playing Nicely with Others

How the universal buffer cache works



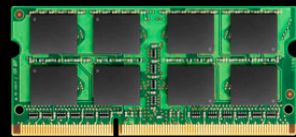
4 GB

1 TB



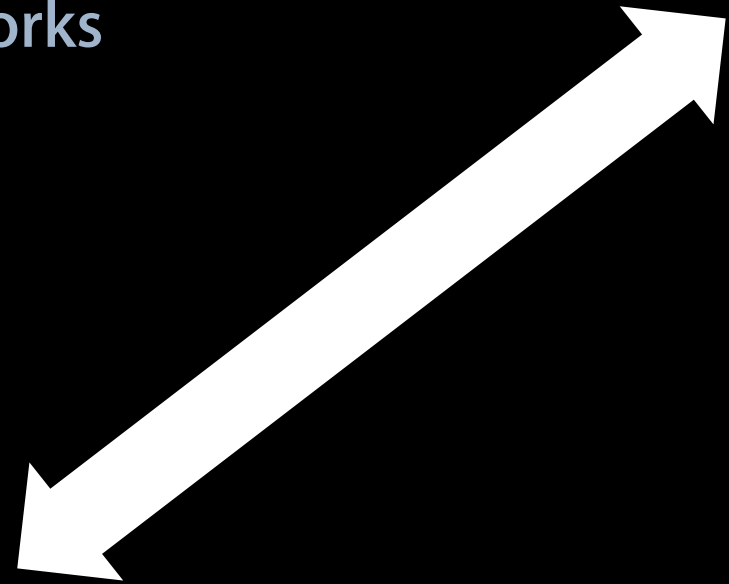
Playing Nicely with Others

How the universal buffer cache works



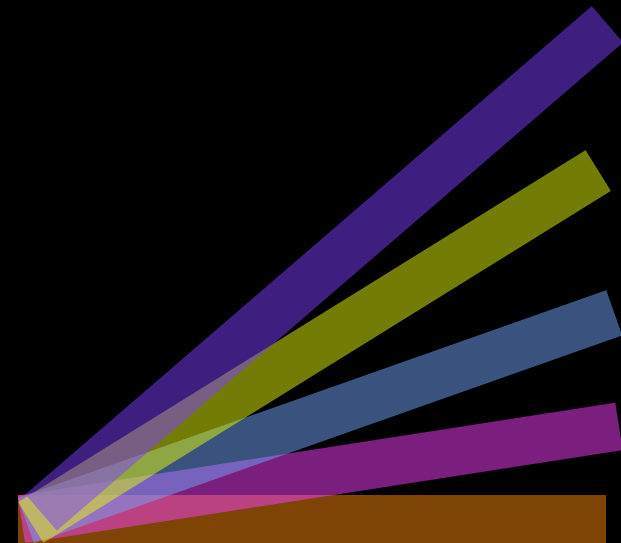
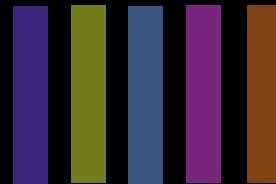
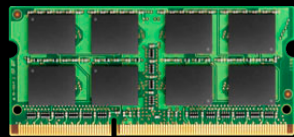
4 GB

1 TB



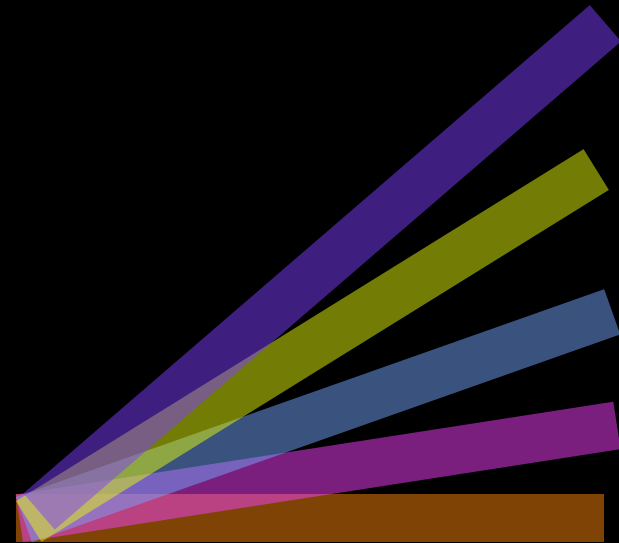
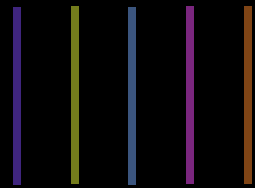
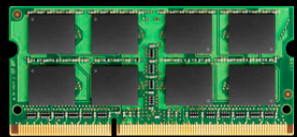
Playing Nicely with Others

How the universal buffer cache works



Performance Tips

F_NOCACHE

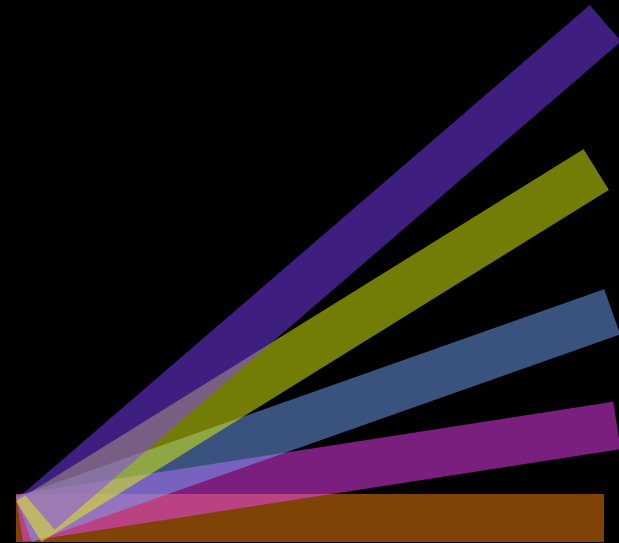
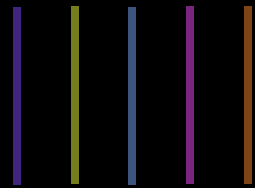
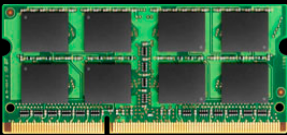


Performance Tips

F_NOCACHE

Turn caching OFF: `fcntl(fd, F_NOCACHE, 1);`

Turn caching ON: `fcntl(fd, F_NOCACHE, 0);`

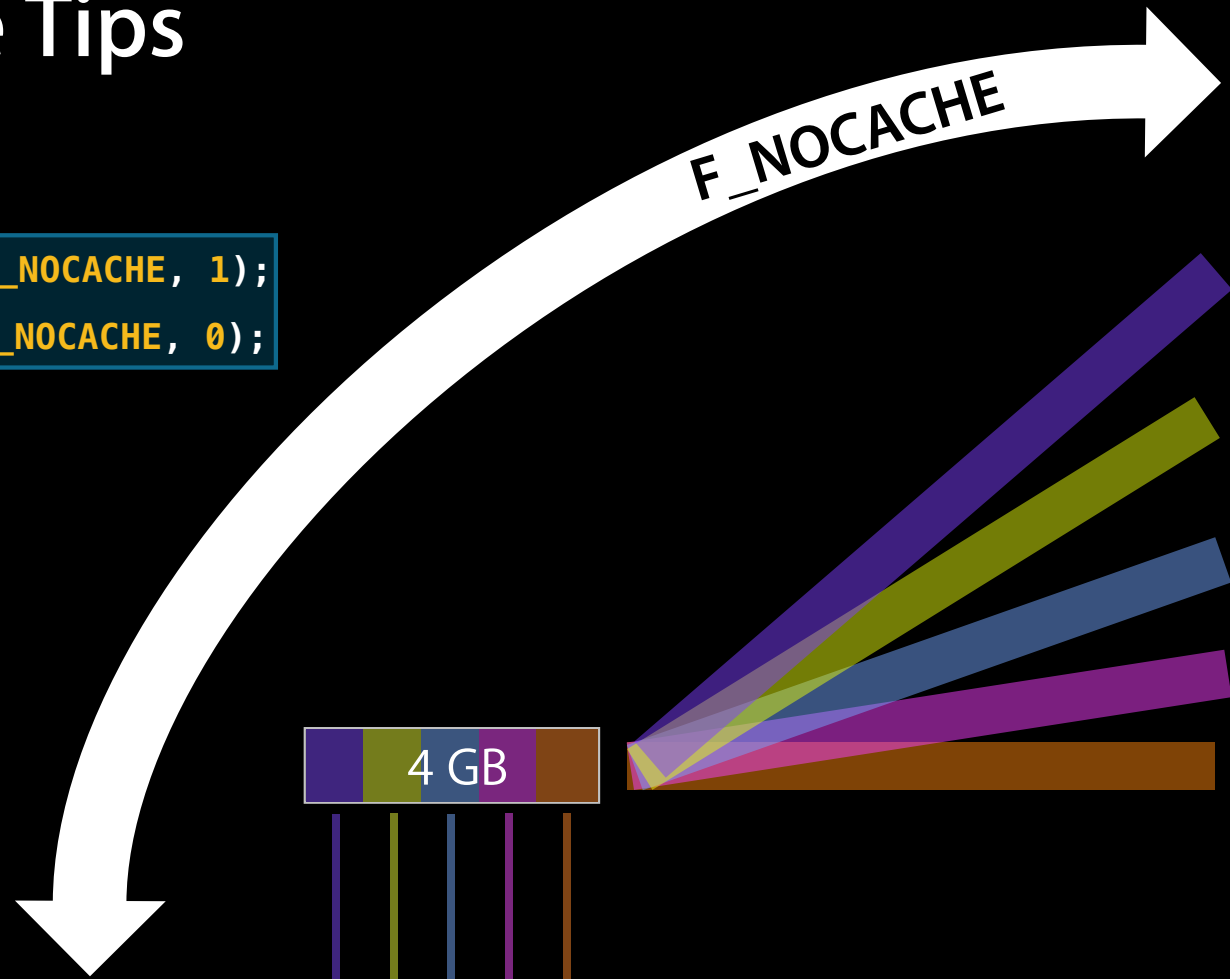
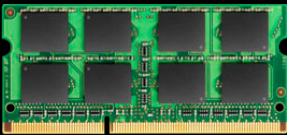


Performance Tips

F_NOCACHE

Turn caching OFF: `fcntl(fd, F_NOCACHE, 1);`

Turn caching ON: `fcntl(fd, F_NOCACHE, 0);`



4 GB

1 TB



Performance Tips

Buy in bulk: Better performance = Better battery life

- Searching - `NSSearchField`
 - `NSMetadataQuery`
 - Construct queries using `NSPredicate`
- Filesystem notification of changes to a directory hierarchy
 - `FSEvents` - Push or pull events
- Filesystem metadata
 - `CFURLCopyResourcePropertiesForKeys`
 - `CFURLEnumerator`

Performance Tips

Don't SPOD

- Keep I/O off the main UI thread
 - Applications SPOD if they don't service the event loop for two seconds
 - I/O latencies are unpredictable, especially over the network
- Ideally, UI thread should be used to handle events and update UI
- GCD and NSOperation are great ways to offload main thread



Performance Tips

Don't SPOD

- Keep I/O off the main UI thread
 - Applications SPOD if they don't service the event loop for two seconds
 - I/O latencies are unpredictable, especially over the network
- Ideally, UI thread should be used to handle events and update UI
- GCD and NSOperation are great ways to offload main thread



NSFileCoordinator

Working with iCloud

- Coordinates multiple processes reading and writing to the same file
 - Used for managing documents in iCloud
- Notifies applications of changes to update their own data structures
- Recommended usage: UIDocument and NSDocument

```
// myDocument conforms to NSFilePresenter protocol  
  
coordinator = [[NSFileCoordinator alloc] initWithFilePresenter:myDocument];  
  
[NSFileCoordinator addFilePresenter:myDocument];  
[NSFileCoordinator removeFilePresenter:myDocument];
```

Performance Tips

What's going on?

```
sudo fs_usage -w -f filesys
```


Performance Tips

What's going on?

```
sudo fs_usage -w -f filesys TextEdit > /tmp/out.txt
```

```
14:06:38.810798 mkdir /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Doc
14:06:38.811125 open /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Doc
14:06:38.811144 write F=22 B=0x12c
14:06:38.812436 fsync F=22
14:06:38.812455 close F=22
14:06:38.812541 chmod /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Doc
14:06:38.812554 getattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Doc
14:06:38.812607 utimes /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Doc
14:06:38.812636 lstat64 [ 2] /Volumes/SSD/HelloWorld.rtf.sb-1efcc842-9Y3k7V
14:06:38.812646 lstat64 /Volumes/SSD/HelloWorld.rtf
14:06:38.812689 getattrlist /Volumes/SSD/HelloWorld.rtf
14:06:38.812729 getattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Doc
14:06:38.812742 access_extended /Volumes/SSD/HelloWorld.rtf
14:06:38.812819 setattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Doc
14:06:38.813400 rename /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Doc
```

Saving a File in TextEdit

NSDocument implements "Safe Save"



```
14:06:38.810798 mkdir /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit) 0.000096 TextEdit.51451
14:06:38.811125 open F=22 (_WC_T_) /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000115 TextEdit.51451
14:06:38.811144 write F=22 B=0x12c 0.000018 TextEdit.51451
14:06:38.812328 WrData[AP] D=0x003816d8 B=0x1000 /dev/disk1s5 0.001172 W TextEdit.51451
14:06:38.812436 fsync F=22 0.001293 W TextEdit.51451
14:06:38.812455 close F=22 0.000017 TextEdit.51451
14:06:38.812541 chmod <rw-r--r--> /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000047 TextEdit.51451
14:06:38.812554 getattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000006 TextEdit.51451
14:06:38.812607 utimes /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000035 TextEdit.51451
14:06:38.812636 lstat64 [ 2] /Volumes/SSD/HelloWorld.rtf.sb-1efcc842-9Y3k7V 0.000009 TextEdit.51451
14:06:38.812646 lstat64 /Volumes/SSD/HelloWorld.rtf 0.000004 TextEdit.51451
14:06:38.812689 getattrlist /Volumes/SSD/HelloWorld.rtf 0.000007 TextEdit.51451
14:06:38.812729 getattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000009 TextEdit.51451
14:06:38.812742 access_extended /Volumes/SSD/HelloWorld.rtf 0.000004 TextEdit.51451
14:06:38.812819 setattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000047 TextEdit.51451
14:06:38.813400 rename /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000568 W TextEdit.51451
```


Saving a File in TextEdit

NSDocument implements "Safe Save"



```
14:06:38.810798 mkdir /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit) 0.000096 TextEdit.51451
14:06:38.811125 open F=22 (_WC_T_) /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000115 TextEdit.51451
14:06:38.811144 write F=22 B=0x12c 0.000018 TextEdit.51451
14:06:38.812328 WrData[AP] D=0x003816d8 B=0x1000 /dev/disk1s5 0.001172 W TextEdit.51451
14:06:38.812436 fsync F=22 0.001293 W TextEdit.51451
14:06:38.812455 close F=22 0.000017 TextEdit.51451
14:06:38.812541 chmod <r-w-r--r--> /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000047 TextEdit.51451
14:06:38.812554 getattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000006 TextEdit.51451
14:06:38.812607 utimes /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000035 TextEdit.51451
14:06:38.812636 lstat64 [ 2] /Volumes/SSD/HelloWorld.rtf.sb-1efcc842-9Y3k7V 0.000009 TextEdit.51451
14:06:38.812646 lstat64 /Volumes/SSD/HelloWorld.rtf 0.000004 TextEdit.51451
14:06:38.812689 getattrlist /Volumes/SSD/HelloWorld.rtf 0.000007 TextEdit.51451
14:06:38.812729 getattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000009 TextEdit.51451
14:06:38.812742 access_extended /Volumes/SSD/HelloWorld.rtf 0.000004 TextEdit.51451
14:06:38.812819 setattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000047 TextEdit.51451
14:06:38.813400 rename /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000568 W TextEdit.51451
```

Saving a File in TextEdit

NSDocument implements "Safe Save"



```
14:06:38.810798 mkdir /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit) 0.000096 TextEdit.51451
14:06:38.811125 open F=22 (_WC_T_) /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000115 TextEdit.51451
14:06:38.811144 write F=22 B=0x12c 0.000018 TextEdit.51451
14:06:38.812328 WrData[AP] D=0x003816d8 B=0x1000 /dev/disk1s5 0.001172 W TextEdit.51451
14:06:38.812436 fsync F=22 0.001293 W TextEdit.51451
14:06:38.812455 close F=22 0.000017 TextEdit.51451
14:06:38.812541 chmod <rw-r--r--> /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000047 TextEdit.51451
14:06:38.812554 getattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000006 TextEdit.51451
14:06:38.812607 utimes /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000035 TextEdit.51451
14:06:38.812636 lstat64 [ 2] /Volumes/SSD/HelloWorld.rtf.sb-1efcc842-9Y3k7V 0.000009 TextEdit.51451
14:06:38.812646 lstat64 /Volumes/SSD/HelloWorld.rtf 0.000004 TextEdit.51451
14:06:38.812689 getattrlist /Volumes/SSD/HelloWorld.rtf 0.000007 TextEdit.51451
14:06:38.812729 getattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000009 TextEdit.51451
14:06:38.812742 access_extended /Volumes/SSD/HelloWorld.rtf 0.000004 TextEdit.51451
14:06:38.812819 setattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000047 TextEdit.51451
14:06:38.813400 rename /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000568 W TextEdit.51451
```


Saving a File in TextEdit

NSDocument implements "Safe Save"



```
14:06:38.810798 mkdir /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit) 0.000096 TextEdit.51451
14:06:38.811125 open F=22 (_WC_T_) /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000115 TextEdit.51451
14:06:38.811144 write F=22 B=0x12c 0.000018 TextEdit.51451
14:06:38.812328 WrData[AP] D=0x003816d8 B=0x1000 /dev/disk1s5 0.001172 W TextEdit.51451
14:06:38.812436 fsync F=22 0.001293 W TextEdit.51451
14:06:38.812455 close F=22 0.000017 TextEdit.51451
14:06:38.812541 chmod <rw-r--r--> /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000047 TextEdit.51451
14:06:38.812554 getattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000006 TextEdit.51451
14:06:38.812607 utimes /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000035 TextEdit.51451
14:06:38.812636 lstat64 [ 2] /Volumes/SSD/HelloWorld.rtf.sb-1efcc842-9Y3k7V 0.000009 TextEdit.51451
14:06:38.812646 lstat64 /Volumes/SSD/HelloWorld.rtf 0.000004 TextEdit.51451
14:06:38.812689 getattrlist /Volumes/SSD/HelloWorld.rtf 0.000007 TextEdit.51451
14:06:38.812729 getattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000009 TextEdit.51451
14:06:38.812742 access_extended /Volumes/SSD/HelloWorld.rtf 0.000004 TextEdit.51451
14:06:38.812819 setattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000047 TextEdit.51451
14:06:38.813400 rename /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000568 W TextEdit.51451
```

```
14:06:38.811125 open ../HelloWorld.rtf 0.000096 TextEdit.51451
14:06:38.811144 write 0.000018 TextEdit.51451
14:06:38.812436 fsync 0.001293 W TextEdit.51451
14:06:38.813400 rename ../HelloWorld.rtf 0.000568 W TextEdit.51451
```

Saving a File in TextEdit

NSDocument implements "Safe Save"



```

14:06:38.810798 mkdir /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit) 0.000096 TextEdit.51451
14:06:38.811125 open F=22 (_WC_T_) /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000115 TextEdit.51451
14:06:38.811144 write F=22 B=0x12c 0.000018 TextEdit.51451
14:06:38.812328 WrData[AP] D=0x003816d8 B=0x1000 /dev/disk1s5 0.001172 W TextEdit.51451
14:06:38.812436 fsync F=22 0.001293 W TextEdit.51451
14:06:38.812455 close F=22 0.000017 TextEdit.51451
14:06:38.812541 chmod <rw-r--r--> /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000047 TextEdit.51451
14:06:38.812554 getattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000006 TextEdit.51451
14:06:38.812607 utimes /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000035 TextEdit.51451
14:06:38.812636 lstat64 [ 2] /Volumes/SSD/HelloWorld.rtf.sb-1efcc842-9Y3k7V 0.000009 TextEdit.51451
14:06:38.812646 lstat64 /Volumes/SSD/HelloWorld.rtf 0.000004 TextEdit.51451
14:06:38.812689 getattrlist /Volumes/SSD/HelloWorld.rtf 0.000007 TextEdit.51451
14:06:38.812729 getattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000009 TextEdit.51451
14:06:38.812742 access_extended /Volumes/SSD/HelloWorld.rtf 0.000004 TextEdit.51451
14:06:38.812819 setattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000047 TextEdit.51451
14:06:38.813400 rename /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000568 W TextEdit.51451

```

```

14:06:38.811125 open ../HelloWorld.rtf 0.000096 TextEdit.51451
14:06:38.811144 write 0.000018 TextEdit.51451
14:06:38.812436 fsync 0.001293 W TextEdit.51451
14:06:38.813400 rename ../HelloWorld.rtf 0.000568 W TextEdit.51451

```

Saving a File in TextEdit

NSDocument implements "Safe Save"



```

14:06:38.810798 mkdir /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit) 0.000096 TextEdit.51451
14:06:38.811125 open F=22 (_WC_T_) /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000115 TextEdit.51451
14:06:38.811144 write F=22 B=0x12c 0.000018 TextEdit.51451
14:06:38.812328 WrData[AP] D=0x003816d8 B=0x1000 /dev/disk1s5 0.001172 W TextEdit.51451
14:06:38.812436 fsync F=22 0.001293 W TextEdit.51451
14:06:38.812455 close F=22 0.000017 TextEdit.51451
14:06:38.812541 chmod <rw-r--r--> /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000047 TextEdit.51451
14:06:38.812554 getattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000006 TextEdit.51451
14:06:38.812607 utimes /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000035 TextEdit.51451
14:06:38.812636 lstat64 [ 2] /Volumes/SSD/HelloWorld.rtf.sb-1efcc842-9Y3k7V 0.000009 TextEdit.51451
14:06:38.812646 lstat64 /Volumes/SSD/HelloWorld.rtf 0.000004 TextEdit.51451
14:06:38.812689 getattrlist /Volumes/SSD/HelloWorld.rtf 0.000007 TextEdit.51451
14:06:38.812729 getattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000009 TextEdit.51451
14:06:38.812742 access_extended /Volumes/SSD/HelloWorld.rtf 0.000004 TextEdit.51451
14:06:38.812819 setattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000047 TextEdit.51451
14:06:38.813400 rename /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000568 W TextEdit.51451

```

```

14:06:38.811125 open ../HelloWorld.rtf 0.000096 TextEdit.51451
14:06:38.811144 write 0.000018 TextEdit.51451
14:06:38.812436 fsync 0.001293 W TextEdit.51451
14:06:38.813400 rename ../HelloWorld.rtf 0.000568 W TextEdit.51451

```

Saving a File in TextEdit

NSDocument implements "Safe Save"



```
14:06:38.810798 mkdir /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit) 0.000096 TextEdit.51451
14:06:38.811125 open F=22 (_WC_T_) /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000115 TextEdit.51451
14:06:38.811144 write F=22 B=0x12c 0.000018 TextEdit.51451
14:06:38.812328 WrData[AP] D=0x003816d8 B=0x1000 /dev/disk1s5 0.001172 W TextEdit.51451
14:06:38.812436 fsync F=22 0.001293 W TextEdit.51451
14:06:38.812455 close F=22 0.000017 TextEdit.51451
14:06:38.812541 chmod <rw-r--r--> /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000047 TextEdit.51451
14:06:38.812554 getattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000006 TextEdit.51451
14:06:38.812607 utimes /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000035 TextEdit.51451
14:06:38.812636 lstat64 [ 2] /Volumes/SSD/HelloWorld.rtf.sb-1efcc842-9Y3k7V 0.000009 TextEdit.51451
14:06:38.812646 lstat64 /Volumes/SSD/HelloWorld.rtf 0.000004 TextEdit.51451
14:06:38.812689 getattrlist /Volumes/SSD/HelloWorld.rtf 0.000007 TextEdit.51451
14:06:38.812729 getattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000009 TextEdit.51451
14:06:38.812742 access_extended /Volumes/SSD/HelloWorld.rtf 0.000004 TextEdit.51451
14:06:38.812819 setattrlist /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000047 TextEdit.51451
14:06:38.813400 rename /Volumes/SSD/.TemporaryItems/folders.501/TemporaryItems/(A Document Being Saved By TextEdit)/HelloWorld.rtf 0.000568 W TextEdit.51451
```

```
14:06:38.811125 open ../HelloWorld.rtf 0.000096 TextEdit.51451
14:06:38.811144 write 0.000018 TextEdit.51451
14:06:38.812436 fsync 0.001293 W TextEdit.51451
14:06:38.813400 rename ../HelloWorld.rtf 0.000568 W TextEdit.51451
```

Total Time: 0.003351 seconds
1/300 of a second

Agenda

File system and frameworks update

Storage industry trends

Recent changes

Best practices

File Vault 2

What it is

How it works

Agenda

File system and frameworks update

Storage industry trends

Recent changes

Best practices

File Vault 2

What it is

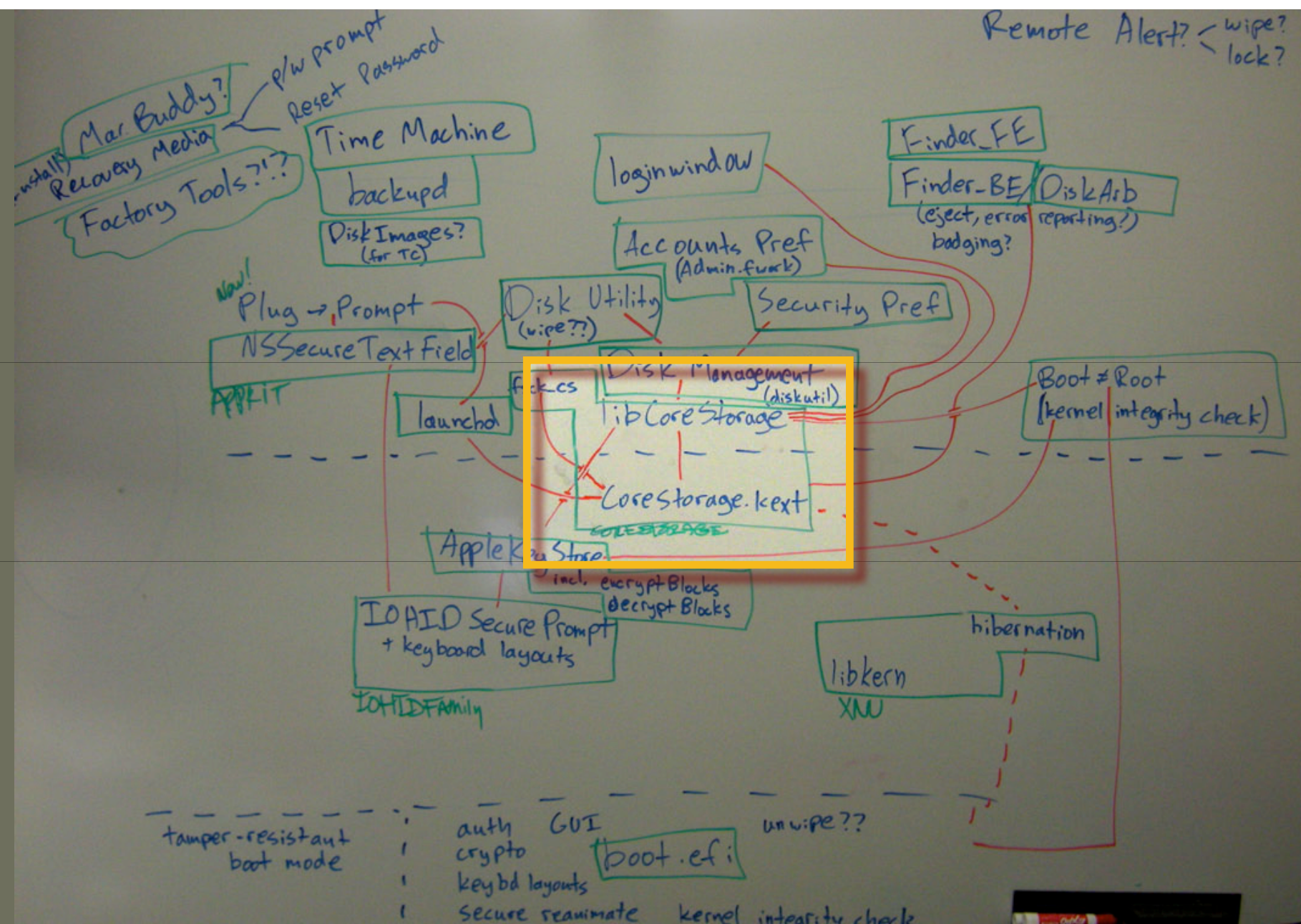
How it works

A thief recently stole a computer server belonging to a major U.S. insurance company, and company officials now fear that the personal data of nearly 1 million people could be at risk

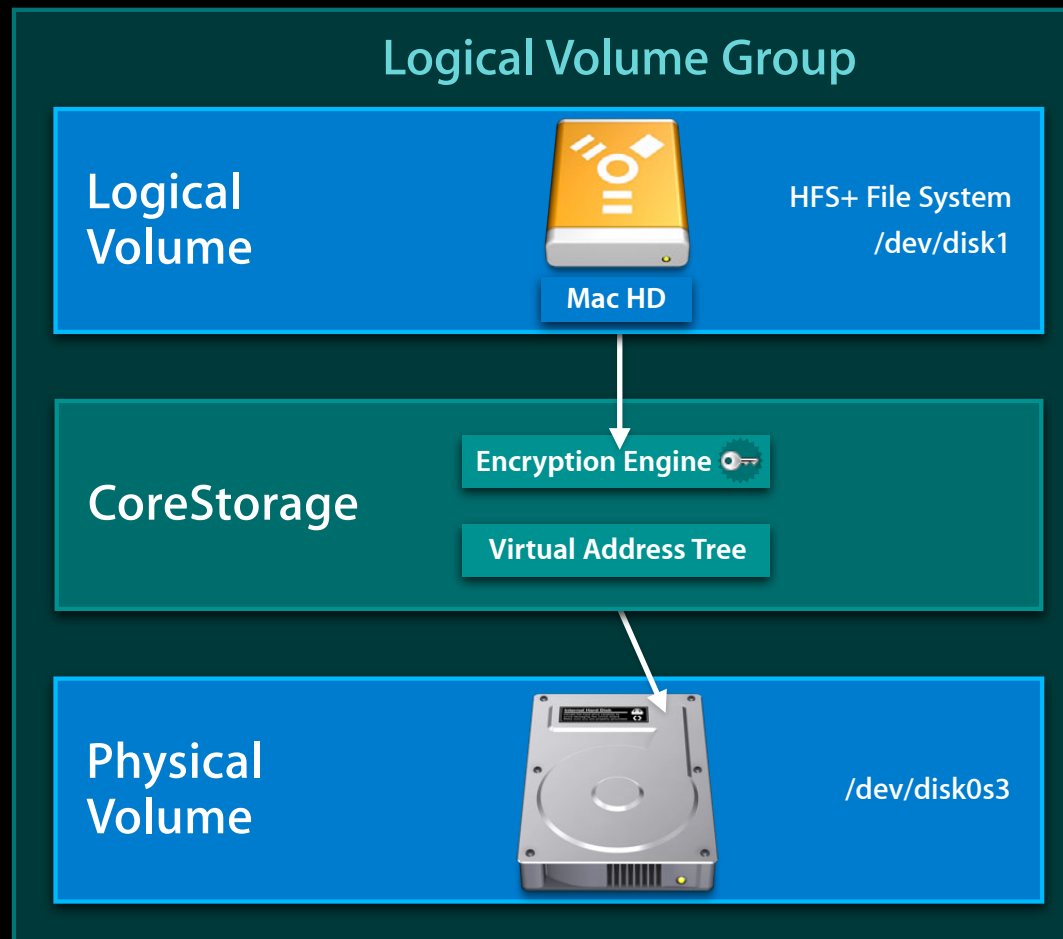
A thief recently stole a computer server belonging to a major U.S. insurance company, and company officials now fear that the **personal data of nearly 1 million people could be at risk**

California's law SB 1386, and 41 other states, requires companies to notify consumers whose personal information has been compromised

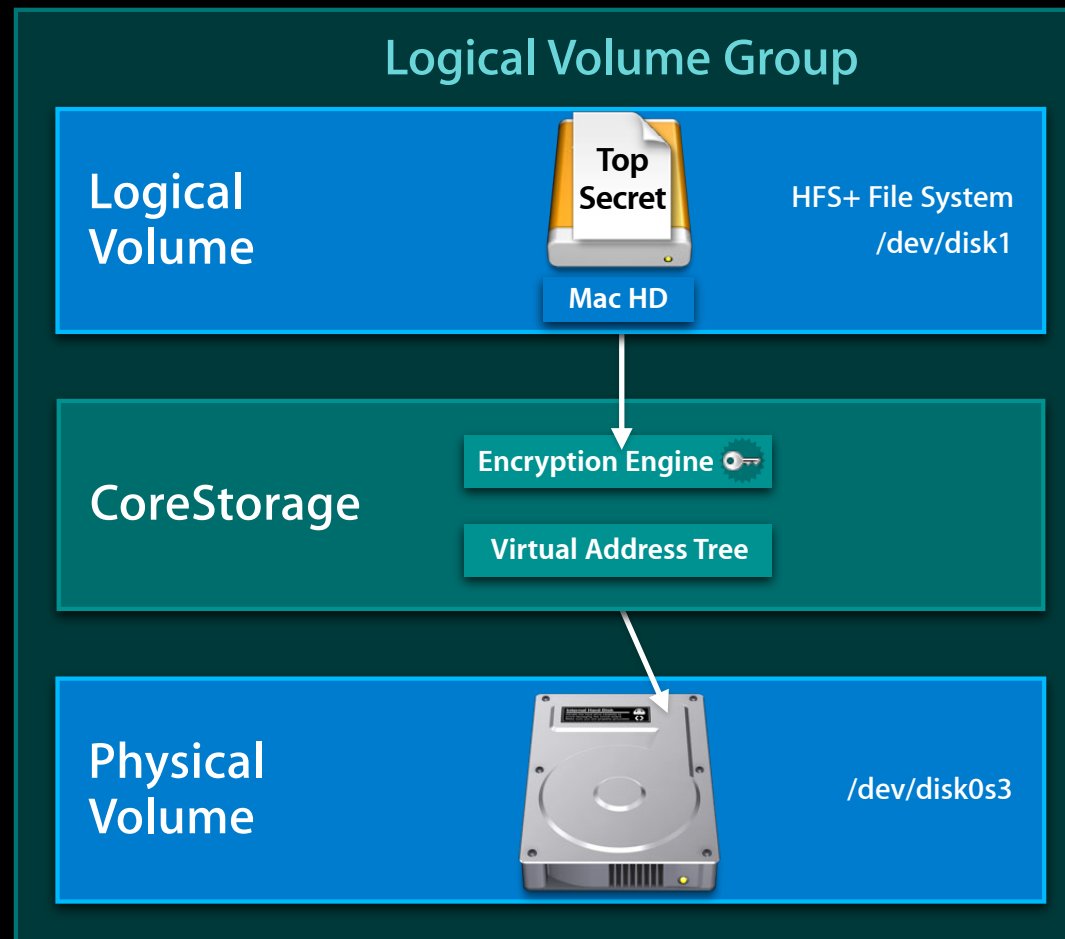
Remote Alert? < wipe?
lock?



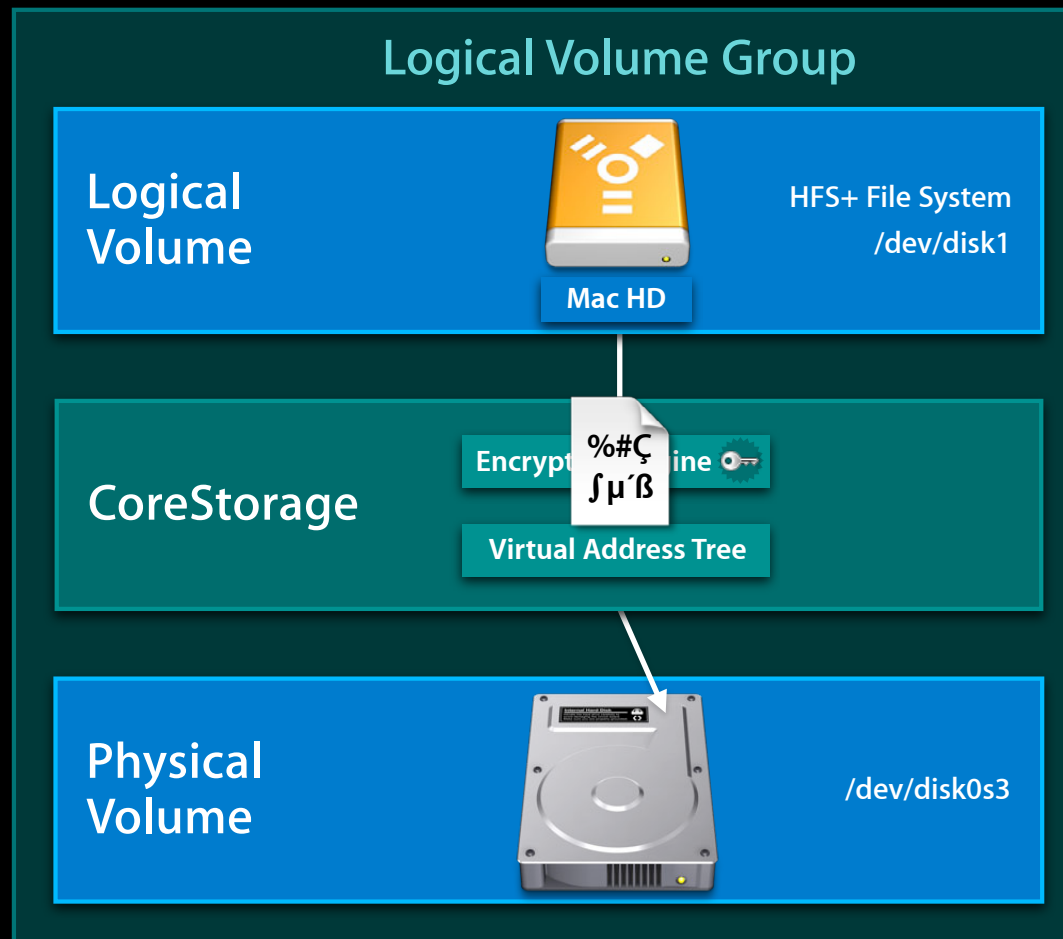
What Is CoreStorage?



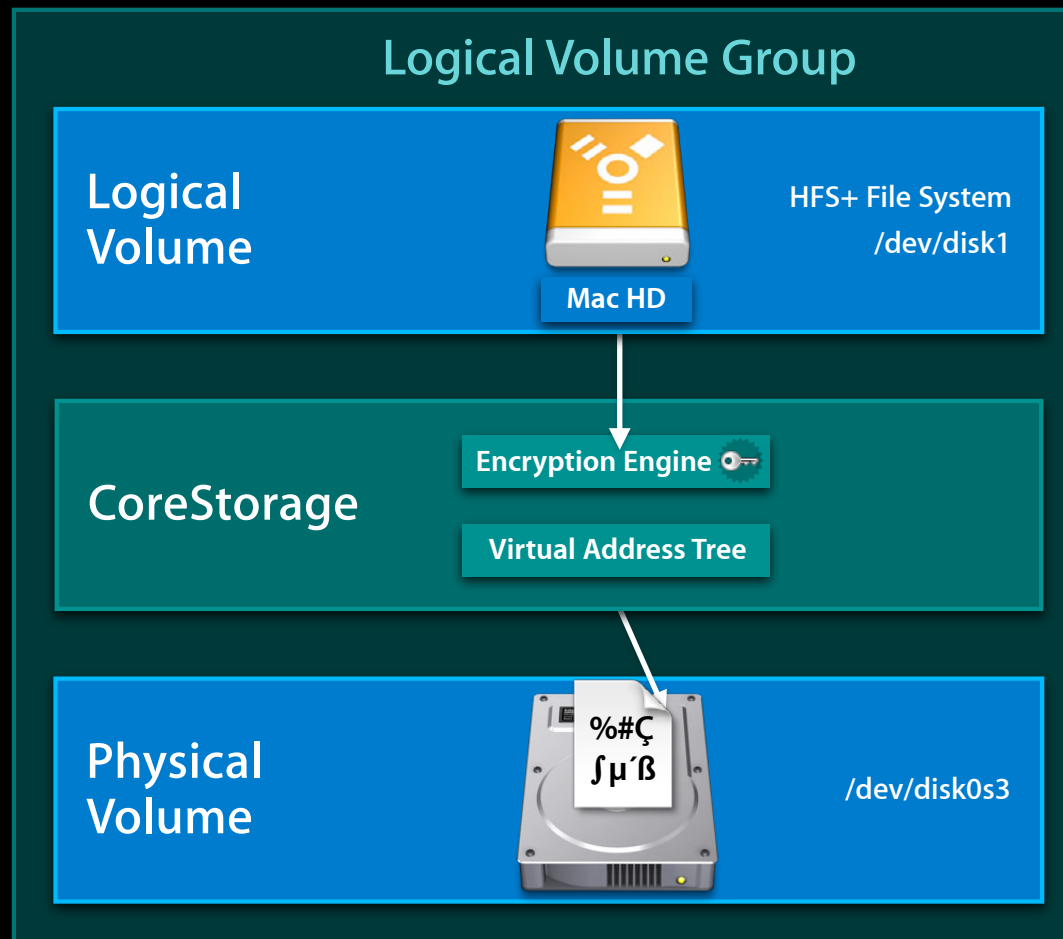
What Is CoreStorage?



What Is CoreStorage?



What Is CoreStorage?



What Is FileVault 2

Full disk encryption

- Password protects your volume
- Encrypts every block on your volume
- Includes file data and metadata
 - Names, dates, attributes, etc.
- Encrypt external drives (thumb drives)
- Keychain integration!
- Encrypted Time Machine backups



What Is FileVault 2?

AES-XTS



- AES-128 + cipher stealing
 - For each 512 byte block
 - Encrypt 16 bytes at a time with AES
 - Mix result across entire block
- FIPS compliant
- NIST standard
 - 1/27/10 - NIST recommended AES-XTS mode

What Is FileVault 2?

AES-XTS



- AES-128 + cipher stealing
 - For each 512 byte block
 - Encrypt 16 bytes at a time with AES
 - Mix result across entire block
- FIPS compliant
- NIST standard
 - 1/27/10 - NIST recommended AES-XTS mode

How big is 2^{128} ?

What Is FileVault 2?

AES-XTS



- AES-128 + cipher stealing
 - For each 512 byte block
 - Encrypt 16 bytes at a time with AES
 - Mix result across entire block
- FIPS compliant
- NIST standard

▪ 1/27/1 Imagine a computer the size of a grain of sand that could test a key in the amount of time it takes light to cross it.

If the whole planet were covered in 1 meter of these computers, it would take on average 1,000 years to crack a 128-bit key.

What Is FileVault 2?

“Treat 1st Reboot Special”

- Solves two problems
 - Keyboard no longer functions from EFI
 - User forgot their password
- Reboot into Recovery OS
 - Pair Bluetooth keyboard
 - Unencrypt volume



What Is FileVault 2?

Personal recovery

- Computer generated password to unlock FDE volume
 - EPD8-8NRF-24T3-ZDUR-J5XM-UATW
- Boots to “Password Reset” in Login Window UI
- Option of saving recovery password with Apple
 - Recovery password is encrypted with three “Bankers Questions”
 - **Apple can NOT unlock your disk**
 - **There is NO back door**

Agenda

File system and frameworks update

Storage industry trends

Recent changes

Best practices

File Vault 2

What it is

How it works

Agenda

File system and frameworks update

Storage industry trends

Recent changes

Best practices

File Vault 2

What it is

How it works

FileVault 2

How it works

Soren Spies

Storage Technologies

Understanding FileVault 2

- CoreStorage and in-place encryption
- FileVault 2 Keys
- FileVault 2 Tools
- Investigations/Requests

Understanding FileVault 2

CoreStorage -> HFS+

```
root> diskutil list
```

```
/dev/disk0
```

#:	TYPE	NAME	SIZE	IDENTIFIER
0:	GUID_partition_scheme		*121.8 GB	disk0
1:	EFI		209.7 MB	disk0s1
2:	Apple_CoreStorage		60.7 GB	disk0s2
3:	Apple_Boot	Boot OSX	650.0 MB	disk0s3
4:	Apple_HFS	Untitled	60.2 GB	disk0s4
5:	Apple_Boot	Boot OSX	134.2 MB	disk0s5

```
/dev/disk1
```

#:	TYPE	NAME	SIZE	IDENTIFIER
0:	Apple_HFS	11A395	*60.3 GB	disk1



CoreStorage

Understanding FileVault 2

CoreStorage -> HFS+

```
root> diskutil list
```

```
/dev/disk0
```

#:	TYPE	NAME	SIZE	IDENTIFIER
0:	GUID_partition_scheme		*121.8 GB	disk0
1:	EFI		209.7 MB	disk0s1
2:	Apple_CoreStorage		60.7 GB	disk0s2
3:	Apple_Boot	Boot OSX	650.0 MB	disk0s3
4:	Apple_HFS	Untitled	60.2 GB	disk0s4
5:	Apple_Boot	Boot OSX	134.2 MB	disk0s5

```
/dev/disk1
```

#:	TYPE	NAME	SIZE	IDENTIFIER
0:	Apple_HFS	11A395	*60.3 GB	disk1



CoreStorage

Understanding FileVault 2

CoreStorage -> HFS+

```
root> diskutil list
```

```
/dev/disk0
```

#:	TYPE	NAME	SIZE	IDENTIFIER
0:	GUID_partition_scheme		*121.8 GB	disk0
1:	EFI		209.7 MB	disk0s1
2:	Apple_CoreStorage		60.7 GB	disk0s2
3:	Apple_Boot	Boot OSX	650.0 MB	disk0s3
4:	Apple_HFS	Untitled	60.2 GB	disk0s4
5:	Apple_Boot	Boot OSX	134.2 MB	disk0s5

```
/dev/disk1
```

#:	TYPE	NAME	SIZE	IDENTIFIER
0:	Apple_HFS	11A395	*60.3 GB	disk1

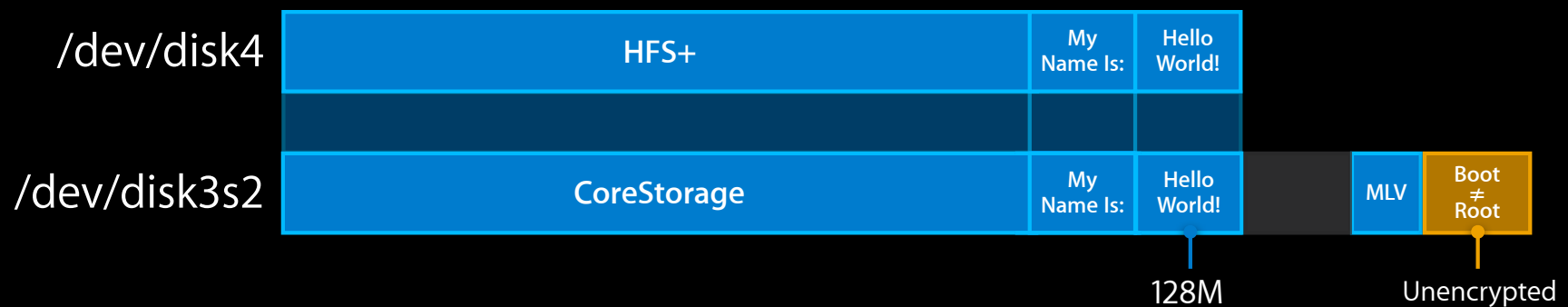
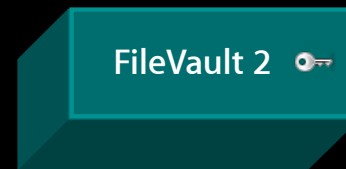


CoreStorage

Understanding FileVault 2

In-place encryption

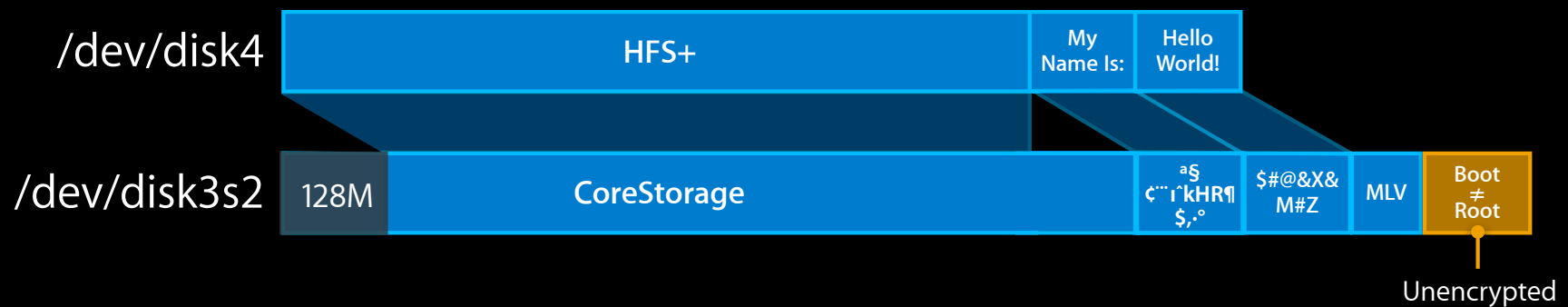
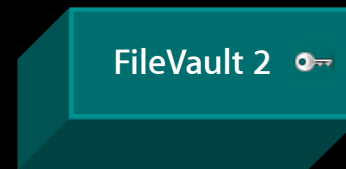
- 128 MB mapping window
- Read, modify, write
- Crash -> redo last 128 MB



Understanding FileVault 2

In-place encryption

- 128 MB mapping window
- Read, modify, write
- Crash -> redo last 128 MB



FileVault 2 Keys

FileVault 2 Keys

Storage hierarchy

- Random volume key
- Random “key encrypting key”
- Copy of “KEK” for each user



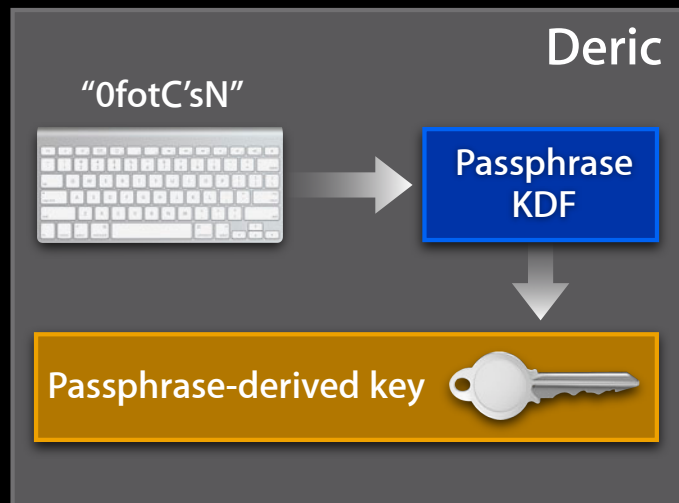
FileVault 2 Keys

Multi-user unlock



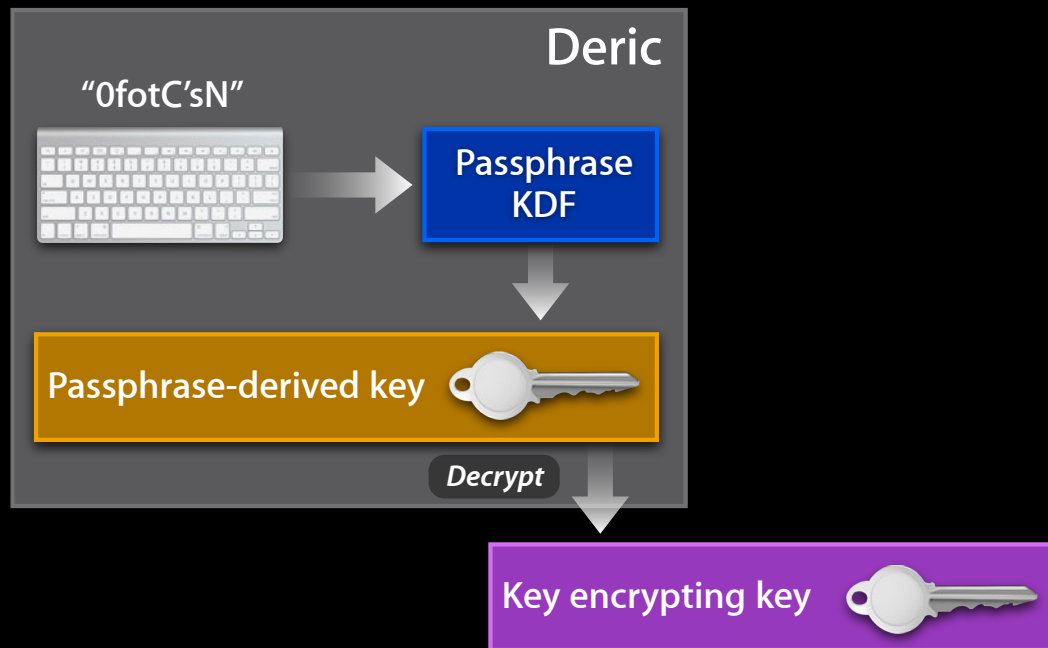
FileVault 2 Keys

Multi-user unlock



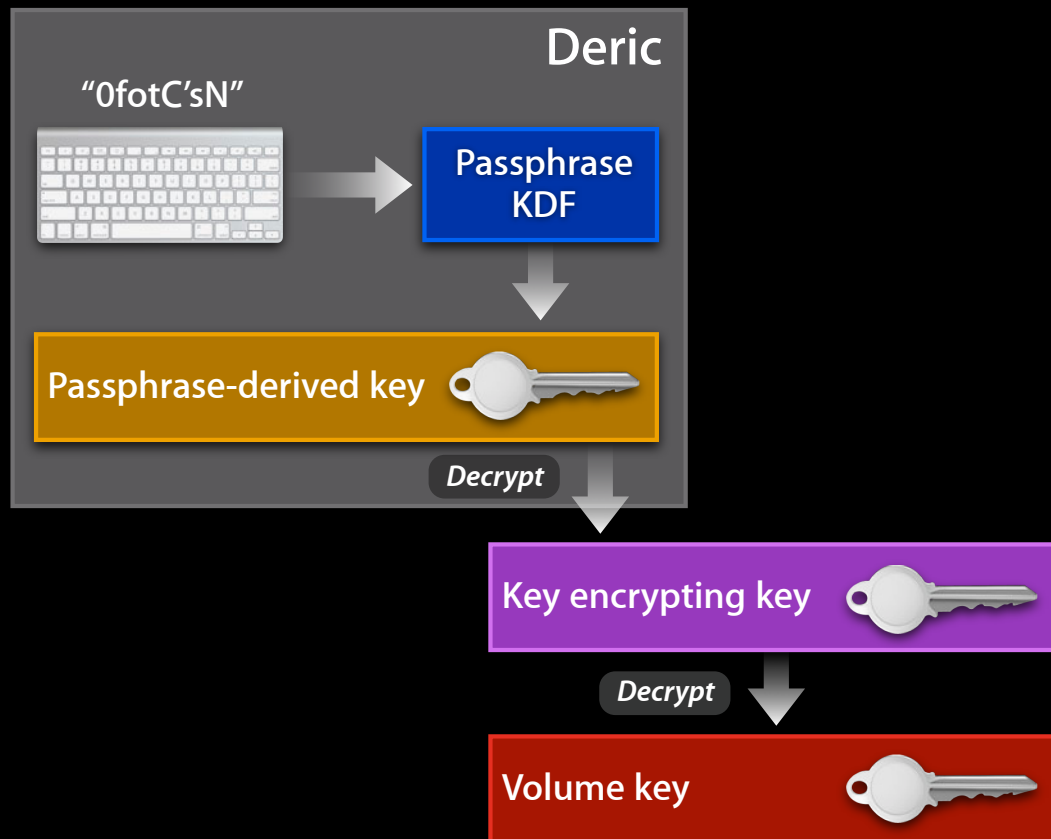
FileVault 2 Keys

Multi-user unlock



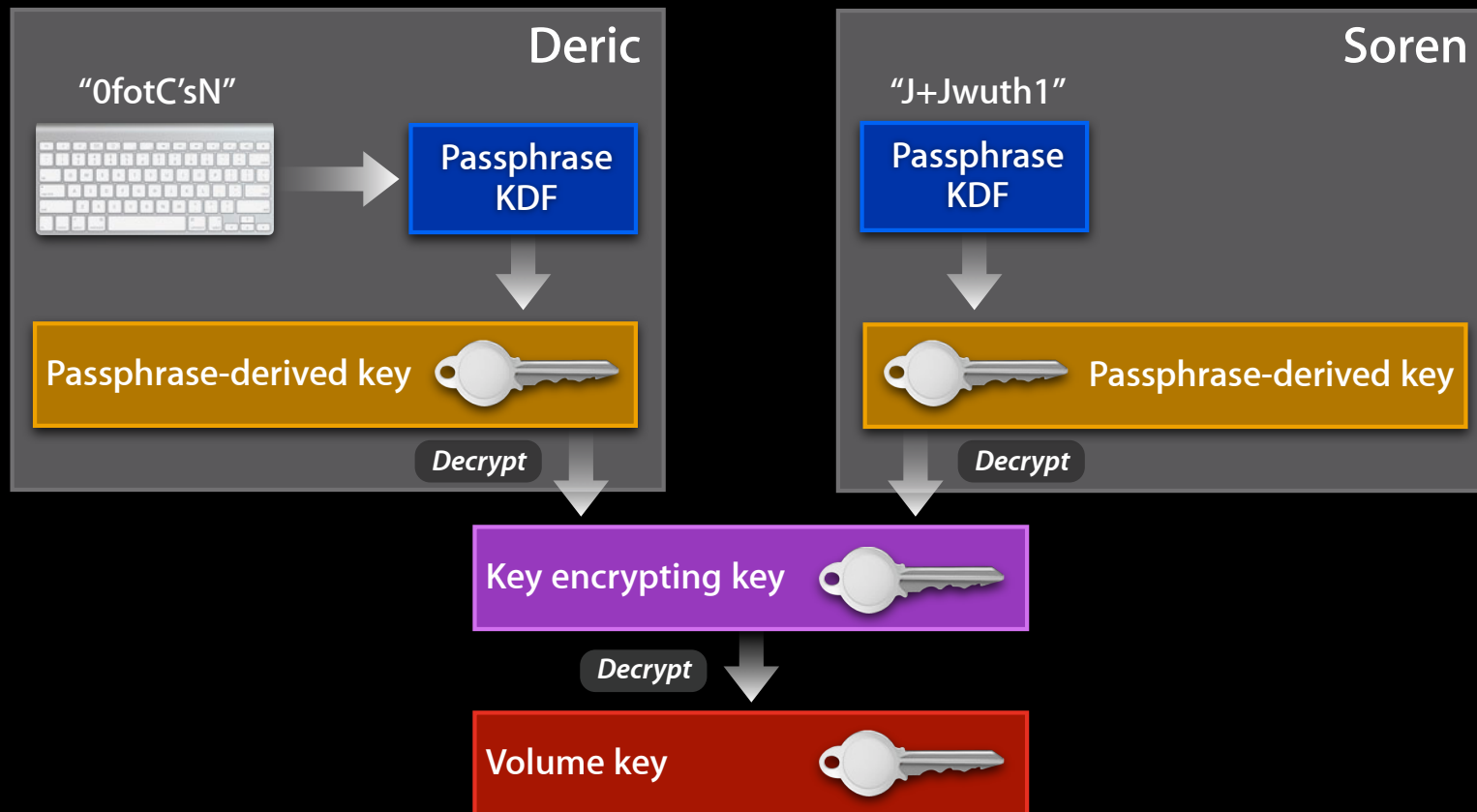
FileVault 2 Keys

Multi-user unlock



FileVault 2 Keys

Multi-user unlock



FileVault 2 Keys

User types

- OS user
 - Automatically created and synchronized
- Disk passphrase
 - Disk Utility, Time Machine, Finder
 - Optionally store in keychain
- Personal recovery
- Institutional recovery

FileVault 2 Keys

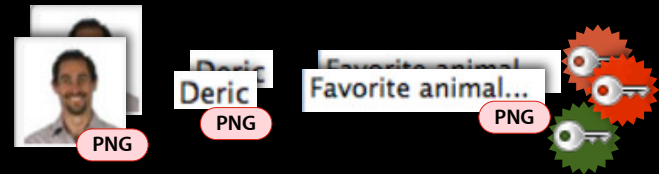
Bonus: Fast wipe

- Encrypted keys in CoreStorage metadata
 - Metadata encrypted with “wipe key”
 - Keys, users, pictures, hints, etc.
- Wipe key in the clear
- Wipe key destruction leaves data “cryptographically inaccessible”

FileVault 2 Keys

Bonus: Fast wipe

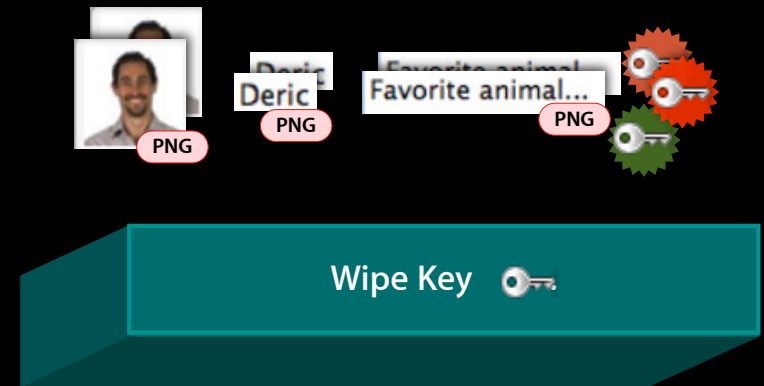
- Encrypted keys in CoreStorage metadata
 - Metadata encrypted with “wipe key”
 - Keys, users, pictures, hints, etc.
- Wipe key in the clear
- Wipe key destruction leaves data “cryptographically inaccessible”



FileVault 2 Keys

Bonus: Fast wipe

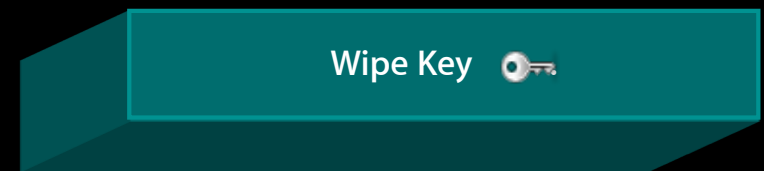
- Encrypted keys in CoreStorage metadata
 - Metadata encrypted with “wipe key”
 - Keys, users, pictures, hints, etc.
- Wipe key in the clear
- Wipe key destruction leaves data “cryptographically inaccessible”



FileVault 2 Keys

Bonus: Fast wipe

- Encrypted keys in CoreStorage metadata
 - Metadata encrypted with “wipe key”
 - Keys, users, pictures, hints, etc.
- Wipe key in the clear
- Wipe key destruction leaves data “cryptographically inaccessible”



~@`f\$∞çβ©Δ °-^~@`~ÁÍÁ%°~@-fΔÁíYRDurd©-¥©LIYGGø...•
¥PGÖÁf¥ñ`>R^58YGflR\$¥∞í`...H`~
%o@í`ΧΔ` f flØ`o`a`ΠÖ`HN`·çÓD@í`Δ†`H`ØΠ
°OUP`KΔbkh©fytstersjy f Ø`UH-îHLIGFT®`¥∞íÁ%FΔ¥

FileVault 2 Key Security

FileVault 2 Key Security

- Attacks
 - Password
 - Algorithm
- Increase attack cost

FileVault 2 Key Security

Password attacks

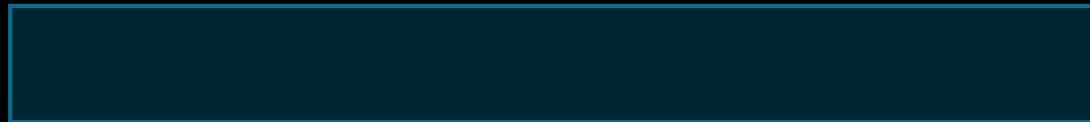
- 128-bit keys >> user passwords
 - Precomputed hashes
 - Brute force
- Attack surface
 - Password-encrypted keys
 - Encrypted volume data
- Money, time, computing resources

FileVault 2 Key Security

Mitigating password attacks

- PBKDF2(<password>, <16byteSalt>, ~100k)
 - Calibrated to 1/10 second
- Password verification
 - Store SHA256(32B_RandPrefix | KeyEncryptingKey | 32B_RandSuffix)
 - Unlock KEK, hash, compare

Puppies, 88a8 644a 9d1f 4be3 5a20 b6aa 2bc8 719d



FileVault 2 Key Security

Mitigating password attacks

- PBKDF2(<password>, <16byteSalt>, ~100k)
 - Calibrated to 1/10 second
- Password verification
 - Store SHA256(32B_RandPrefix | KeyEncryptingKey | 32B_RandSuffix)
 - Unlock KEK, hash, compare

4ffc 5343 3ce1 df7d 157a 592a 23df 4d90

~100,000 iterations
~1/10th second

FileVault 2 Key Security

Algorithm attacks

- Attacks
 - Side-channel (timing, power, etc.)
 - Reduced round variant
 - Related key
 - Known-plaintext
 - Chosen-plaintext

FileVault 2 Key Security

Mitigating algorithm attacks

- Duplicate plaintext
 - Unique ciphertexts within one volume
 - Unique ciphertexts on different volumes

FileVault 2 Key Security

Mitigating algorithm attacks: Tweaking

- Within one volume
 - Every 512 bytes tweaked with logical block address
- On different volumes
 - Random volume key
 - XTS tweaked with $\text{SHA256}(\langle \text{volKey} \rangle | \langle \text{volUUID} \rangle)$

FileVault 2 Key Security

Mitigating algorithm attacks: Tweaking

Per-block tweak

Plaintext



HFS+

Volume Key



Encrypt
.....
Decrypt

XTS
Tweak

CoreStorage

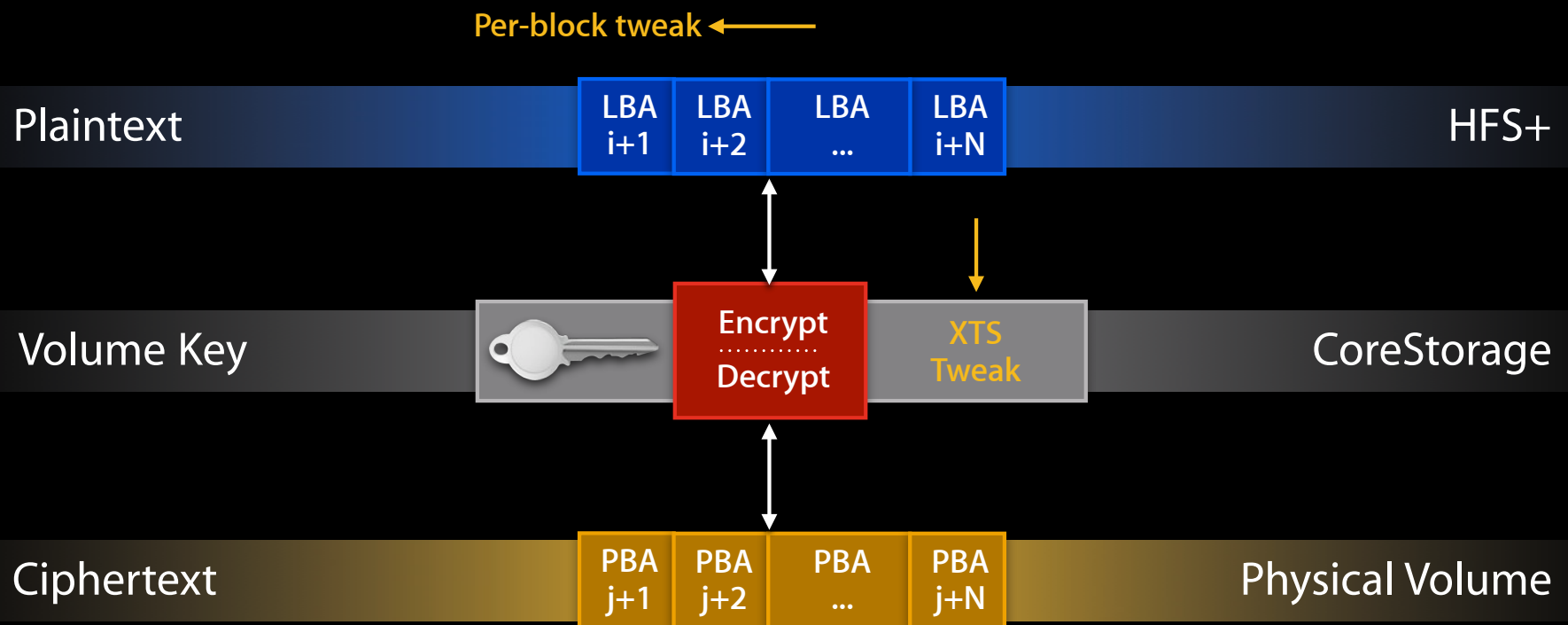
Ciphertext



Physical Volume

FileVault 2 Key Security

Mitigating algorithm attacks: Tweaking



FileVault 2 Tools

FileVault 2 Tools

diskutil(8)



- New: Modify existing CoreStorage volumes

```
diskutil cs encryptVolume  
diskutil cs decryptVolume  
diskutil cs deleteVolume
```

- Lion

```
diskutil cs create  
diskutil cs createVolume  
  
diskutil cs convert [-passphrase]
```

FileVault 2 Tools

fdsetup(8)



- Full FileVault 2 on the command line!

```
$ sudo fdsetup --enable
Enter the primary user name: soren
Enter the password for the user 'soren':
Recovery key = 'PCJ4-99VT-YGHV-G79M-RH3Z-0T55'
Please reboot to complete the process.

$ sudo reboot

...
$ sudo fdsetup --status
Encryption in progress: Percent completed = 35.30
```

FileVault 2 Tools

fdsetup(8)



- Full FileVault 2 on the command line!

```
$ sudo fdsetup --enable
Enter the primary user name: soren
Enter the password for the user 'soren':
Recovery key = 'PCJ4-99VT-YGHV-G79M-RH3Z-0T55'
Please reboot to complete the process.

$ sudo reboot
...
$ sudo fdsetup --status
Encryption in progress: Percent completed = 35.30
```

FileVault 2 Tools

fdesetup(8)



- Add, remove users

```
$ sudo fdesetup --adduser --usertoadd deric --usertoadd brad
```

```
$ sudo dscl delete /Users/brad
```

```
$ sudo fdesetup --syncusers
```

- Disable

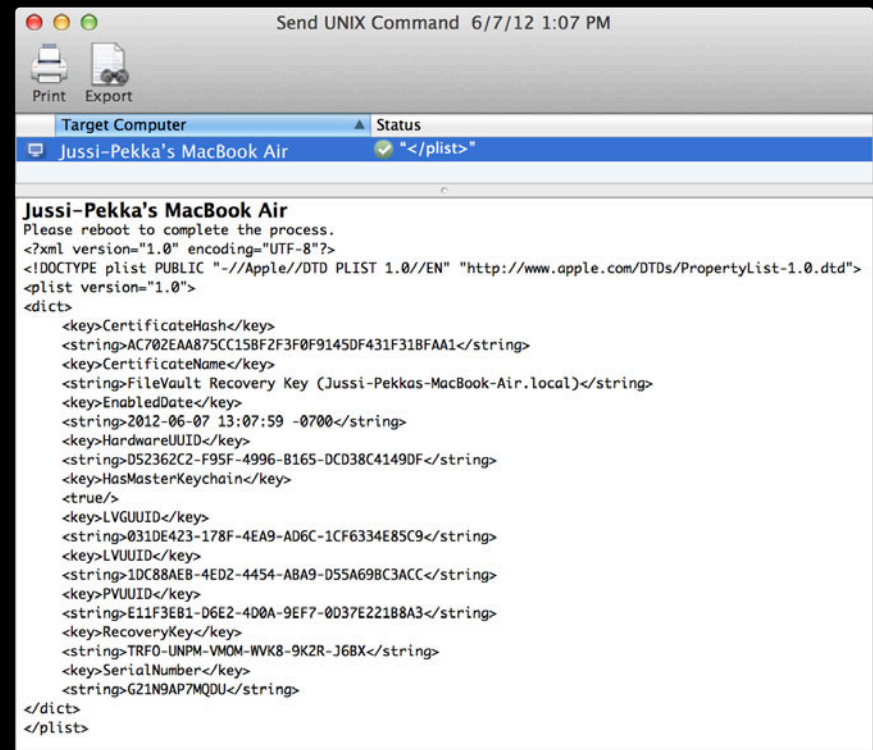
```
$ sudo fdesetup --disable
```

FileVault 2 Tools

fdsetup(8) via Remote Desktop

```
# PromptForSecrets |  
fdsetup --enable --inputplist  
--outputplist | EscrowSettings
```

- fdsetup(8) under construction



```
Send UNIX Command 6/7/12 1:07 PM  
Print Export  
Target Computer Status  
Jussi-Pekka's MacBook Air </plist>  
Jussi-Pekka's MacBook Air  
Please reboot to complete the process.  
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">  
<plist version="1.0">  
<dict>  
<key>CertificateHash</key>  
<string>AC702EAA875CC15BF2F3F0F9145DF431F318FAA1</string>  
<key>CertificateName</key>  
<string>FileVault Recovery Key (Jussi-Pekkas-MacBook-Air.local)</string>  
<key>EnabledDate</key>  
<string>2012-06-07 13:07:59 -0700</string>  
<key>HardwareUUID</key>  
<string>D52362C2-F95F-4996-B165-DCD38C4149DF</string>  
<key>HasMasterKeychain</key>  
<true/>  
<key>LVGUID</key>  
<string>031DE423-178F-4EA9-AD6C-1CF6334E85C9</string>  
<key>LVUID</key>  
<string>1DC88AEB-4ED2-4454-ABA9-D55A698C3ACC</string>  
<key>PVUID</key>  
<string>E11F3EB1-D6E2-4D0A-9EF7-0037E221B8A3</string>  
<key>RecoveryKey</key>  
<string>TRF0-UNPM-VMQM-WVK8-9K2R-J6BX</string>  
<key>SerialNumber</key>  
<string>G21N9AP7MQDU</string>  
</dict>  
</plist>
```

FileVault 2 Tools

Securing sleep

- “Data at rest”
 - Sleep \neq maximally secure
- Stand By
 - Restores keys to RAM
 - A/C power, USB devices

FileVault 2 Tools

Securing sleep

- “Data at rest”
 - Sleep \neq maximally secure
- Stand By
 - Restores keys to RAM
 - A/C power, USB devices

```
$ sudo pmset -a destroyfvkeyonstandby 1 hibernatemode 25
```

Requests and Investigations

Requests and Investigations

No promises!

- Authenticated reboot
- Recovery keys
 - Change
 - Detect use
 - Validate installation
- Low-level rekeying
- OS X Server integration
- Mandatory FileVault 2

More Information

Paul Danbold

Core OS Evangelist
danbold@apple.com

Documentation

File System Programming Guide
<http://developer.apple.com>

Apple Developer Forums

<http://devforums.apple.com>

Related Sessions

iOS App Performance: Responsiveness

Presidio
Thursday 11:30AM

Asynchronous Design Patterns with Blocks, GCD, and XPC

Pacific Heights
Friday 9:00AM

Power Management

Nob Hill
Friday 9:00AM

Labs

File System Lab

Core OS Lab B
Thursday 3:15PM

iCloud Storage Lab

Essentials Lab B
Thursday 4:30PM

OS X Performance Lab

Developer Tools Lab A
Friday 9:00AM

Power Management Lab

Core OS Lab B
Friday 10:15AM

Summary

Remember...

- Performance ~ energy consumption
- Use recommended APIs and frameworks
 - Apple always optimizing
- Beware unexpected latency
 - No I/O on main UI thread

 **WWDC2012**