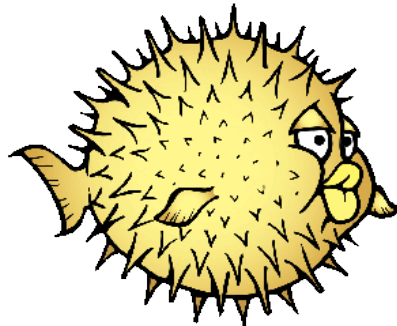


Epitome dedup for the masses

DC*BSDCon '09

Marco Peereboom



***Open*BSD**

Epitome?

- The epitome suite consists of several discrete pieces that provide storage deduplication services.
- Deduplication is defined as the elimination of redundant data.
- Epitome provides a number of services to enable three major archiving technologies:
 - CAS (Content Addressable Storage)
 - SIS (Single Instance Storage)
 - Dedup (Deduplication)

Honorable mention

- Epitome has borrowed several ideas from Plan 9's Venti. More information at:
 - <http://plan9.bell-labs.com/plan9/>
 - <http://plan9.bell-labs.com/magic/man2html/8/venti>
 - http://doc.cat-v.org/plan_9/4th_edition/papers/venti/

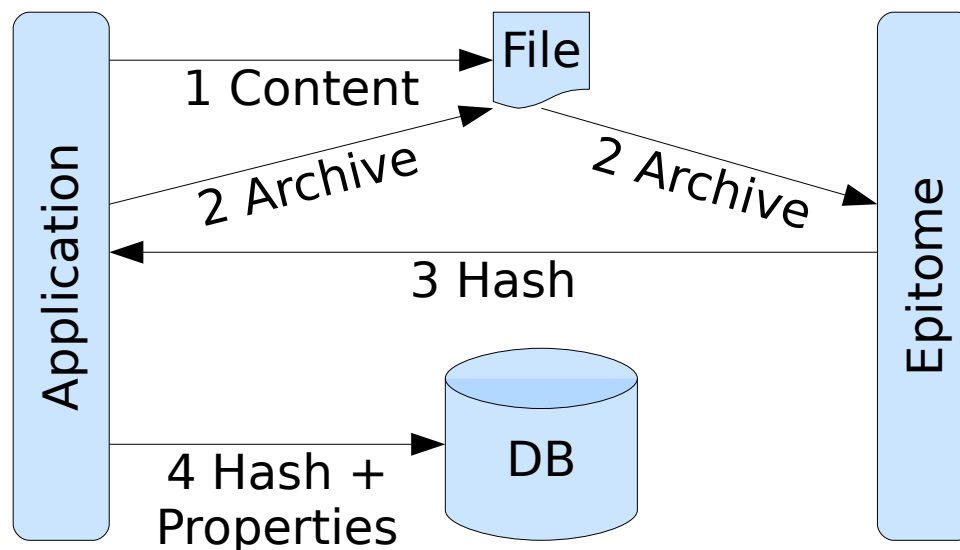


The buzzwords, CAS

- CAS, Content Addressable Storage, also referred to as associative storage, is a mechanism for storing information that can be retrieved based on its content, not its storage location.
- It is typically used for high-speed storage and retrieval of fixed content, such as documents stored for compliance with government regulations and medical content.

The buzzwords, CAS (cont.)

- CAS is a method to archive content and provide the issuer a UUID for identification at a later time.
- The user or application using this service is responsible for maintaining the UUID to content mapping.



The buzzwords, SIS

- SIS, Single Instance Storage, is essentially application deduplication and is best explained with an example:
 - Consider user A sending user B an email; minus the mail header the email is identical so a client (in this case the mail server) that uses SIS would only save the content once.
 - If this email being sent to 100 people the savings are considerable.

The buzzwords, Dedup

- Dedup is essentially the act of chunking data into arbitrary block sizes and calculating a hash over that chunk.
- If the hash does not exist save off the hash and the data.

Why are we deduping?

- Towers of Hanoi backup schemes waste valuable & scarce resources backing up the same content over and over again
- Sizes of backups disproportionate with available bandwidth
- No discernible seek time
- Disk is generally faster than tape
- Disks are cheap
- Tape is going away (really!)
- Tapes are unreliable

Who else is playing?

- Plenty of academic literature is available but no practical open source alternatives exist outside of Plan 9's Venti and Epitome.
- Vendor solutions exist however they are exorbitantly expensive
 - Vendors have gone nuts patenting everything they could come up with
 - Algorithms are very very clever that have some additions such as beacon detection

Choice of hash

- Epitome uses SHA1 (160 bit)
- The probability* of a hash collision (n is number of hashes, b is bits in the hash) is

$$p \leq \frac{n(n-1)}{2} \times \frac{1}{2^b}$$

- Example: 80M hashes @ 16KB (12.2TB of data) has a smaller than 2.18e-31 probability of a hash collision

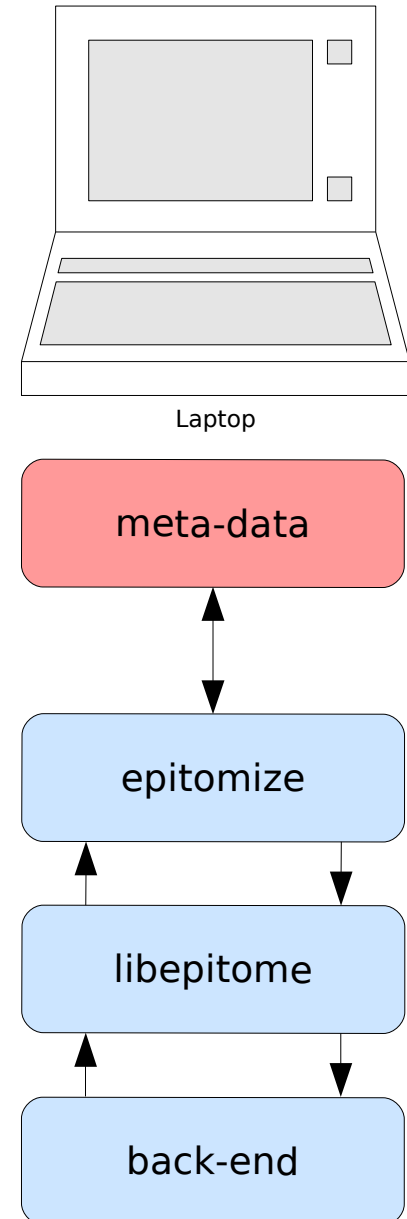
* See section 3.1 of http://doc.cat-v.org/plan_9/4th_edition/papers/venti/

Future hash considerations

- Currently Epitome does not detect collisions
- In the future Epitome will make the selection of hash function user-selectable
 - This will require some plumbing on both the client and server side
- Epitome might take the size of the chunk as a parameter to detect collisions

Epitome 1.0 Architecture

- Implemented today
 - <http://www.peereboom.us/epitome>
 - Not currently part of OpenBSD
- Has 3 working back-ends:
 - libc bdb
 - file
 - raw
- Saves meta-data locally
- Architecture & endian neutral

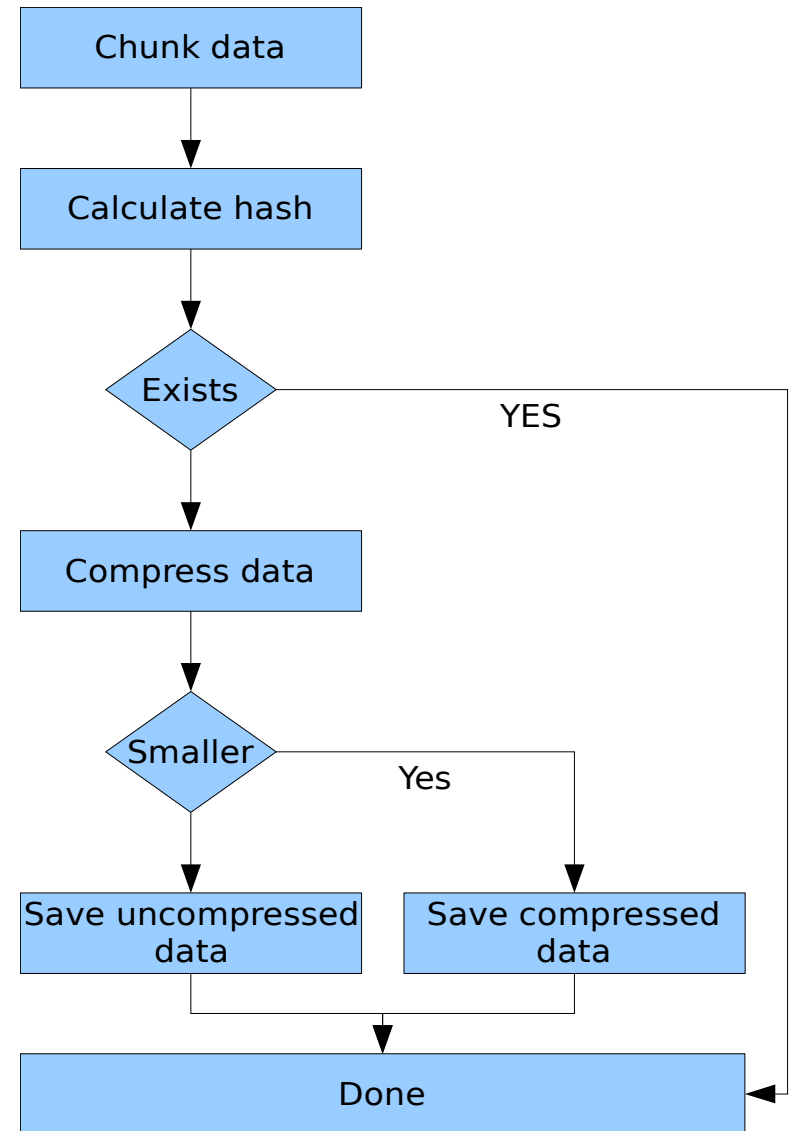


Tools

- Eprepare
 - Prepare the back-end for use with epitome
 - This tool must be run before a deduplicated backup can be made
- Epitool
 - This tool is for maintenance of the back-end and to displays statistics
- Epitomize
 - Tar like tool to create backups

Epitomize Flow

- Epitomize resembles tar as much as possible including command line options
- It will recurse over files/directories and chunk them up into user selected chunk size blocks
- Meta-data is saved locally
 - Metadata becomes essentially a snapshot of the data
- Major exceptions to tar
 - It uses regex(3) instead of glob(3)
 - It eventually will be a network protocol



Epitomize example

```
# epitomize -cvRf myarchive.md sys/
...
files & dirs:                10150
read size:                    84934719
compressed size:              26150059
shas:                          12033
unique shas:                  11468
dedup size:                   54054
total written size:           26185485
dedup reduction ratio:        1%
compression reduction ratio: 31%
overall reduction ratio:      69%
```

Epitomize example con't

```
# epitomize -cvRf myarchive2.md sys/  
...  
files & dirs:                10150  
read size:                    84934719  
compressed size:              0  
shas:                          12033  
unique shas:                   0  
dedup size:                    84934719  
total written size:            0  
dedup reduction ratio:        100%  
compression reduction ratio:  0%  
overall reduction ratio:      100%
```

Epitomize example con't

```
# vi sys/conf/GENERIC
=== enable experimental AOE support ===
# epitomize -cvRf myarchive3.md sys/
...
files & dirs:                10150
read size:                    84934718
compressed size:              1676
shas:                          12033
unique shas:                   1
dedup size:                    84931044
total written size:           1676
dedup reduction ratio:        100%
compression reduction ratio:  46%
overall reduction ratio:      99%
```

Epitomize example con't

```
# du -h myarchive*
1.2M    myarchive.md
1.2M    myarchive2.md
1.2M    myarchive3.md
# du -h sys
94.7M   sys
# mkdir /tmp/1
# mkdir /tmp/2
# epitomize -xvf myarchive2.md -C /tmp/1 sys/
  conf/GENERIC
# epitomize -xvf myarchive3.md -C /tmp/2 sys/
  conf/GENERIC
```

Epitomize example con't

```
# diff -uNp /tmp/1/sys/conf/GENERIC /tmp/2/sys/conf/GENERIC
--- /tmp/1/sys/conf/GENERIC      Sun Dec 28 16:30:27 2008
+++ /tmp/2/sys/conf/GENERIC      Sun Dec 28 16:33:12 2008
@@ -64,7 +64,7 @@ option                PPP_DEFLATE
 option                MROUTING          # Multicast router
 #option                PIM              # prot indep multicast

-#option                AOE              # softraid AOE discipline
+option                AOE              # softraid AOE discipline
  softraid0            at root           # Software RAID
  scsibus*             at softraid?
```

Epitome 1.0 Protocol

- All commands are implemented inside libepitome and currently do not travel over the network
- Primitives:
 - OPEN
 - CLOSE
 - CREATE
 - EXISTS
 - READ
 - WRITE

Epitome Protocol Open

- Open connection to the back-end
 - Open verifies that the back-end has been prepared for use
 - It will open connections to other resources (e.g. a database)
 - Prepare client & server for a session
- Open will fail:
 - if the back-end is not prepared
 - if not all resources can be obtained (database handles etc.)

Epitome Protocol Close

- Close connection to back-end
 - It will close connections to other resources (e.g. a database)
 - Notify client & server to free session resources
- Close performs a sync prior to returning
- Close can not fail

Epitome Protocol Create

- Prepare back-end for first use. This is a destructive operation that is equivalent to newfs.
 - Create new SHA database
 - Initialize tables and other resources depending on the back-end type
- Create can fail if
 - Insufficient privileges on any of the required resources
 - SHA database unavailable or already created

Epitome Protocol Exists

- Test if SHA digest exists
- Exists can fail if:
 - The SHA digest does not exist
 - A resource (e.g. database) failed

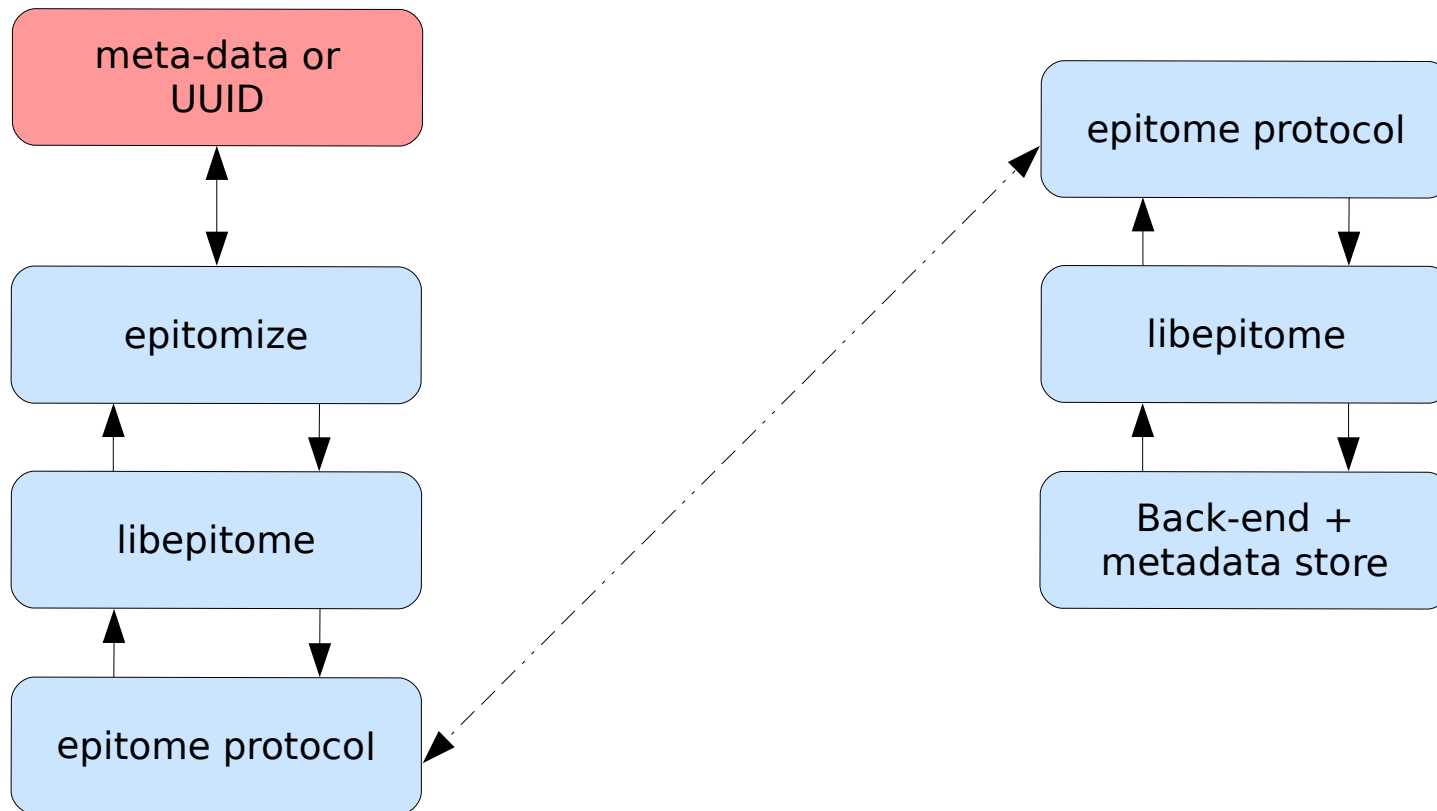
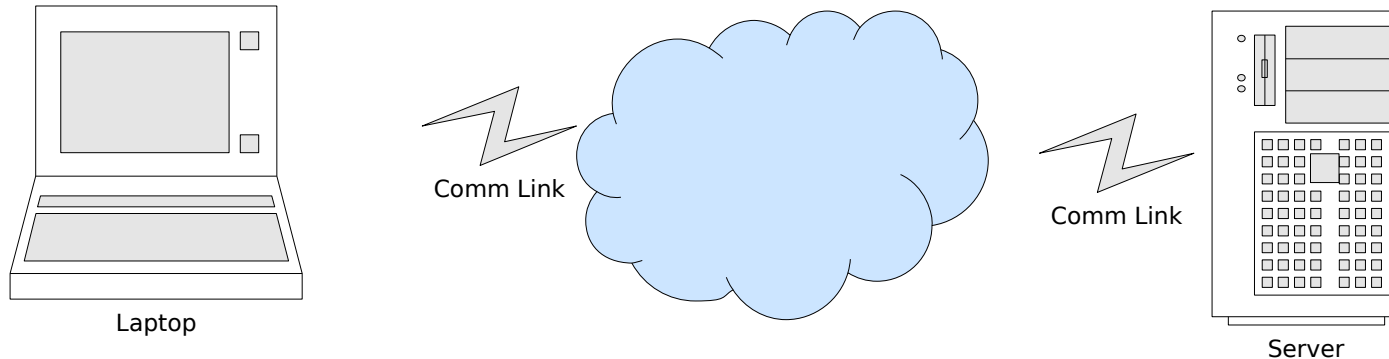
Epitome Protocol Read

- Read SHA digest data + attributes
 - SHA digest and its attributes are read into memory and decompressed if necessary
- Read can fail if:
 - SHA digest does not exist
 - Decompression fails
 - Database does not respond

Epitome Protocol Write

- Write SHA digest data + attributes
 - SHA digest and its attributes are written to and compressed if possible.
 - If compression yields no results the data is saved uncompressed
- Write can fail if:
 - SHA digest already exist
 - Database does not respond

Epitome 2.0 Architecture



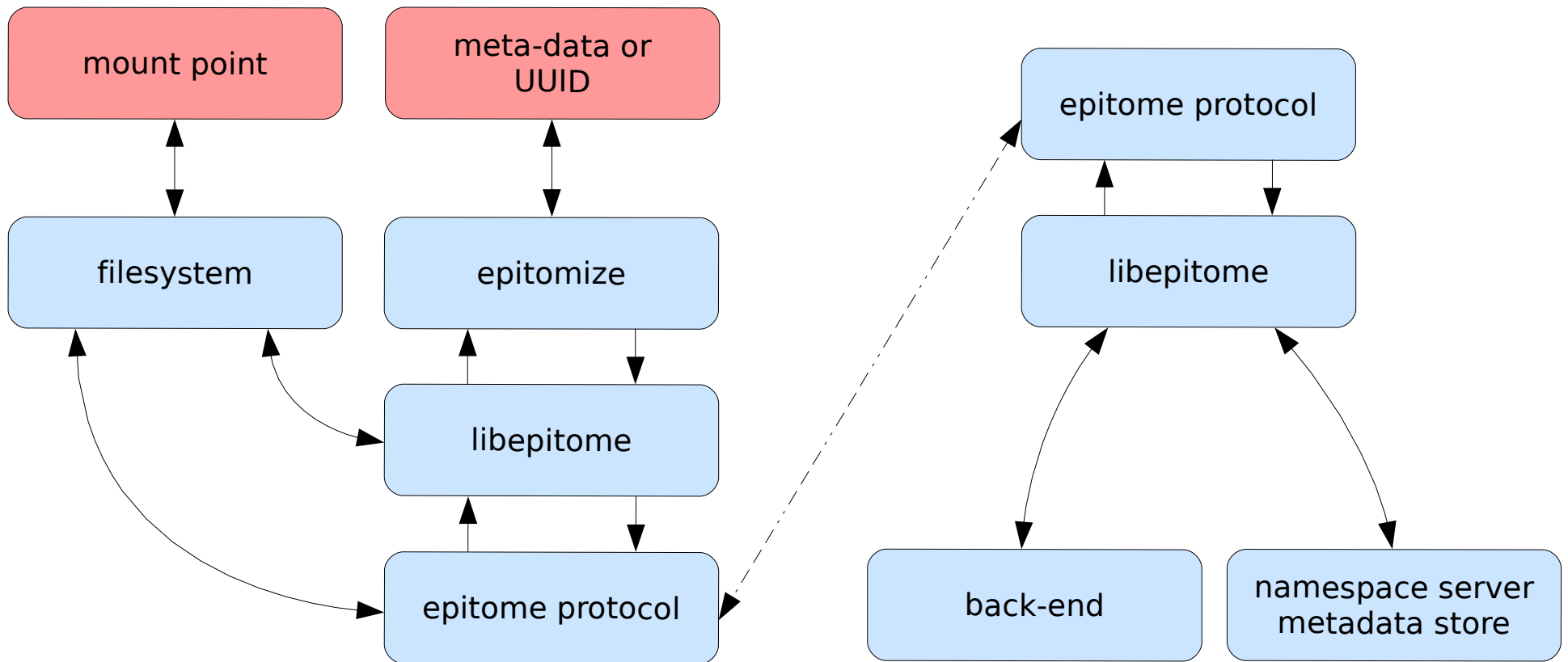
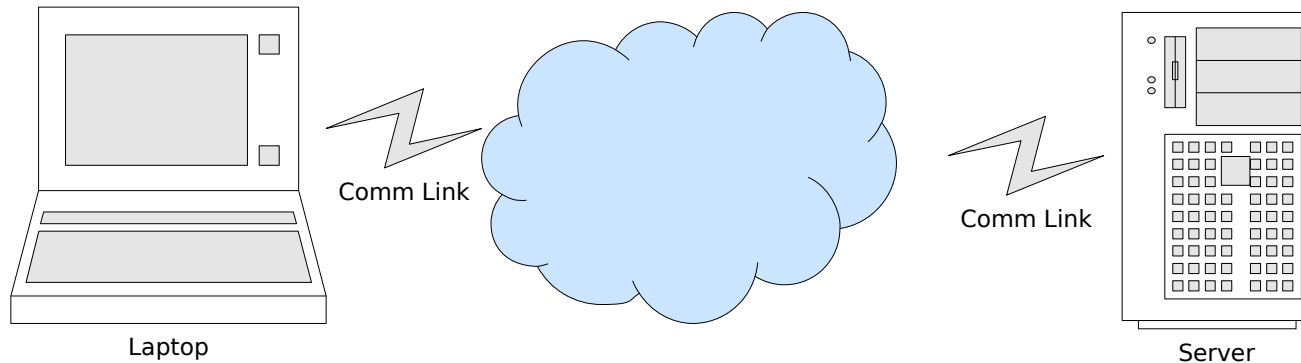
Epitome 2.0 Architecture con't

- Add network protocol
 - Code has been prototyped
- Add additional back-ends:
 - psql
 - mysql
 - Epitome optimized b+tree implementation
- Enhance the raw back-end to do read-modified-write to not waste any space

Epitome 2.0 Protocol

- All commands are usable either locally or over the network
- Metadata:
 - Write
 - Write metadata to epitome server and return a human readable SHA digest that can be used to retrieve the metadata
 - Read
 - Read metadata that contains all pieces required to extract the archive
- CAS is a side effect inherent to the metadata functionality

Epitome 3.0 Architecture



Epitome 3.0 Architecture cont'

- Add filesystem that is deduplicated
 - Inherent to the underlying methodology
all changes are preserved
 - Effectively files become “movies” that can be replayed
- Implement namespace client & server
- Add backup agents:
 - psql
 - mysql

Epitome 3 Protocol

- Namespace
 - Basically all VFS functionality required to provide classic filesystem services
- Database Agent
 - Provide translation services between epitome and databases

Conclusion

- Epitome is flexible enough to be used for a wide variety solutions
- Cross platform and architecture neutral
- It is free!

Questions?