GDC | March 20-24, 2023
San Francisco, CA

Managing Source Content For ‘Overwatch 2’

Rowan Hamilton

#GDC23

What is this talk about?

- Why homebrew source control?
- Evolution for remote work; reducing data transfer by 90%

Problem Space

- One branch is around 5Tb of data
- 4.5 million assets
- 2-week cadence for release branches
- Feature branches
- ~700Gb of data churn per month

What Do I Want

- Launch the game
  - Enumerate all assets
  - Identifier for each asset
  - Hash based on asset content
  - Reference information
- Populate Asset Browsers
  - Name/Tag/Type information about each asset
  - References again

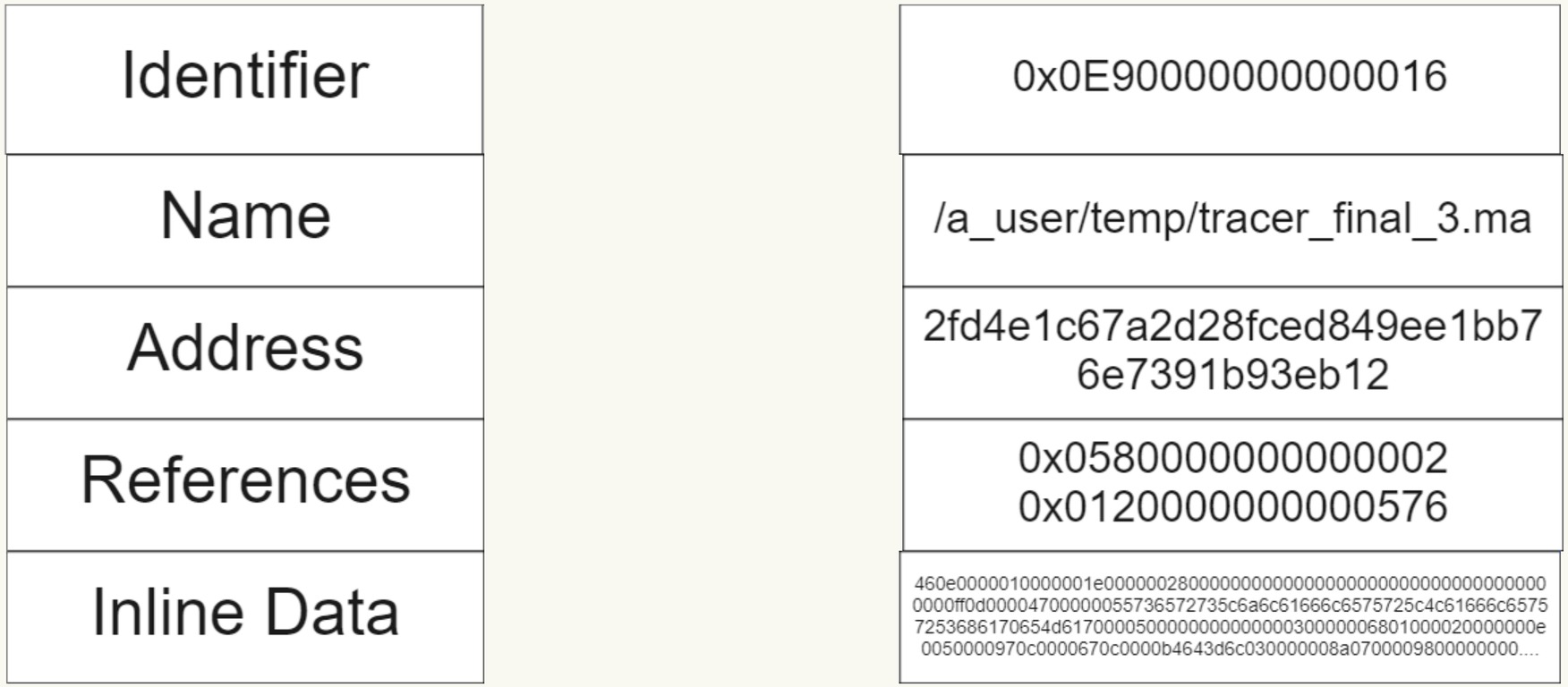
Source Control Basics

- P4 style centralized source control
  - History
  - Branches, Integration, Merging
  - Global exclusive checkouts

Source Control Basics

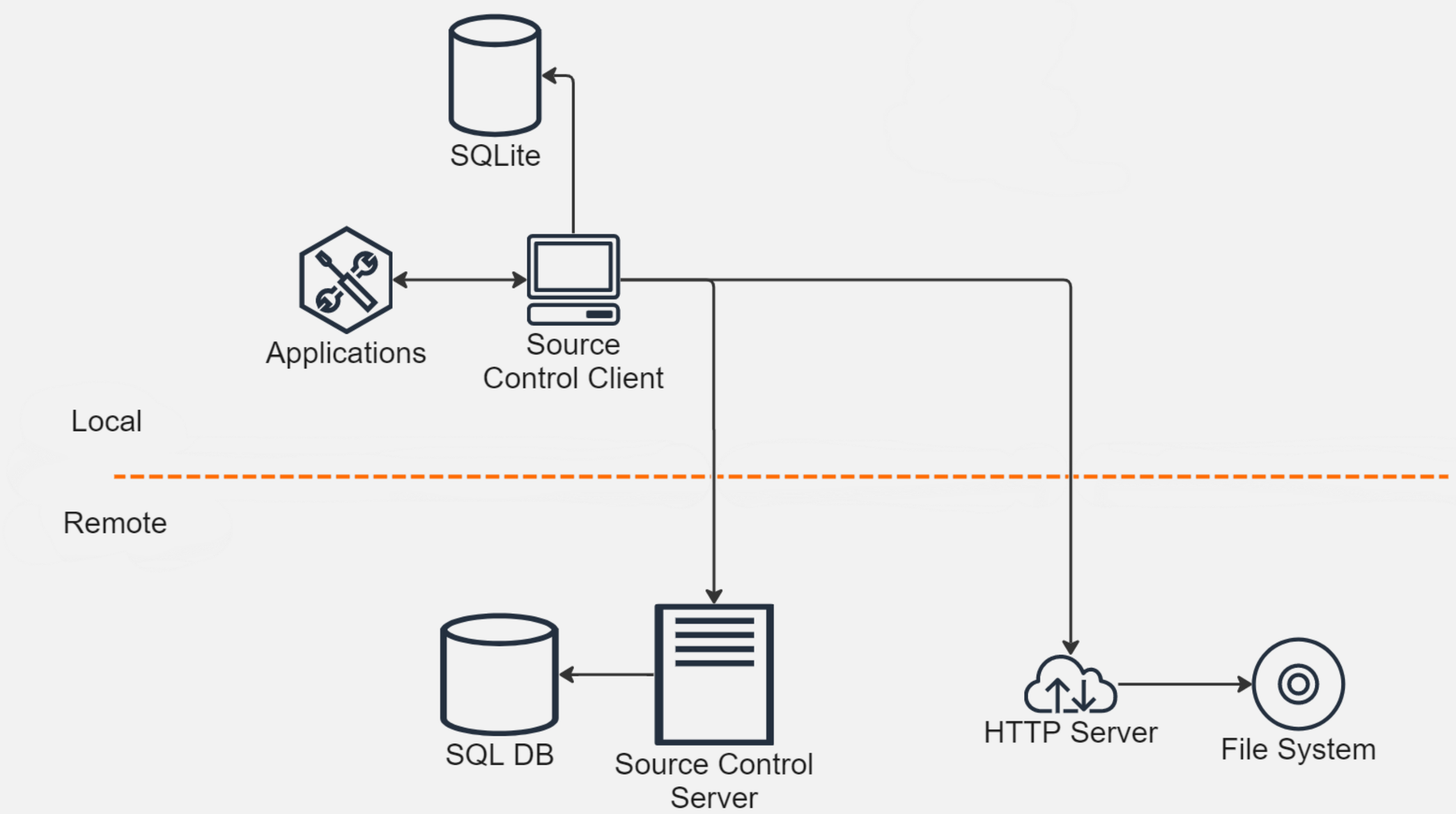
- Independent of files and the filesystem
- Assets use identifiers instead of names
- Communicates via RPC
- Data is streamed on demand

Asset Structure



Fetching Data

- Asset data is (generally) streamed on demand
- HTTP for transport
- http://{endpoint}/cas/assets/{assetAddress}


SQLite
Applications
Source
Control Client
Local
Remote
SQL DB
Source Control
Server
HTTP Server
File System


REDHILL GAMES, HELSINKI
AIRBORN STUDIOS, BERLIN
ROOM8 STUDIOS, UKRAINE
REDHILL GAMES, CYPRUS
OVERWATCH KOREA, SEOUL
BLIZZARD HQ, IRVINE
International Date Line
International Date Line
TOMORROW
0 +1 +2 +3 +4 +5 +8 +9 +10 +15 +17+18

Distributed Struggles

- Extremely high pings and low bandwidth hurt
- Better pipelining solves some of this
- More prebuilt snapshots
- Proxies solve the rest (Varnish for HTTP)

Big Files Are Problematic

- Many assets are > 2Gb
- Compression helps a little
- Delta encoding is not great for big binaries

We Need Less Data

- Time to commit a change should depend on the size of the change, not the size of the file
- Content Defined Chunking (CDC) allows you to deterministically partition a file, based on patterns in the content

CDC Basics

- Uses a rolling hash to find chunk boundaries
- Changing data in one chunk will not affect neighbors
- Tunable chunk size
- Really fast (1GB/sec+)

Example

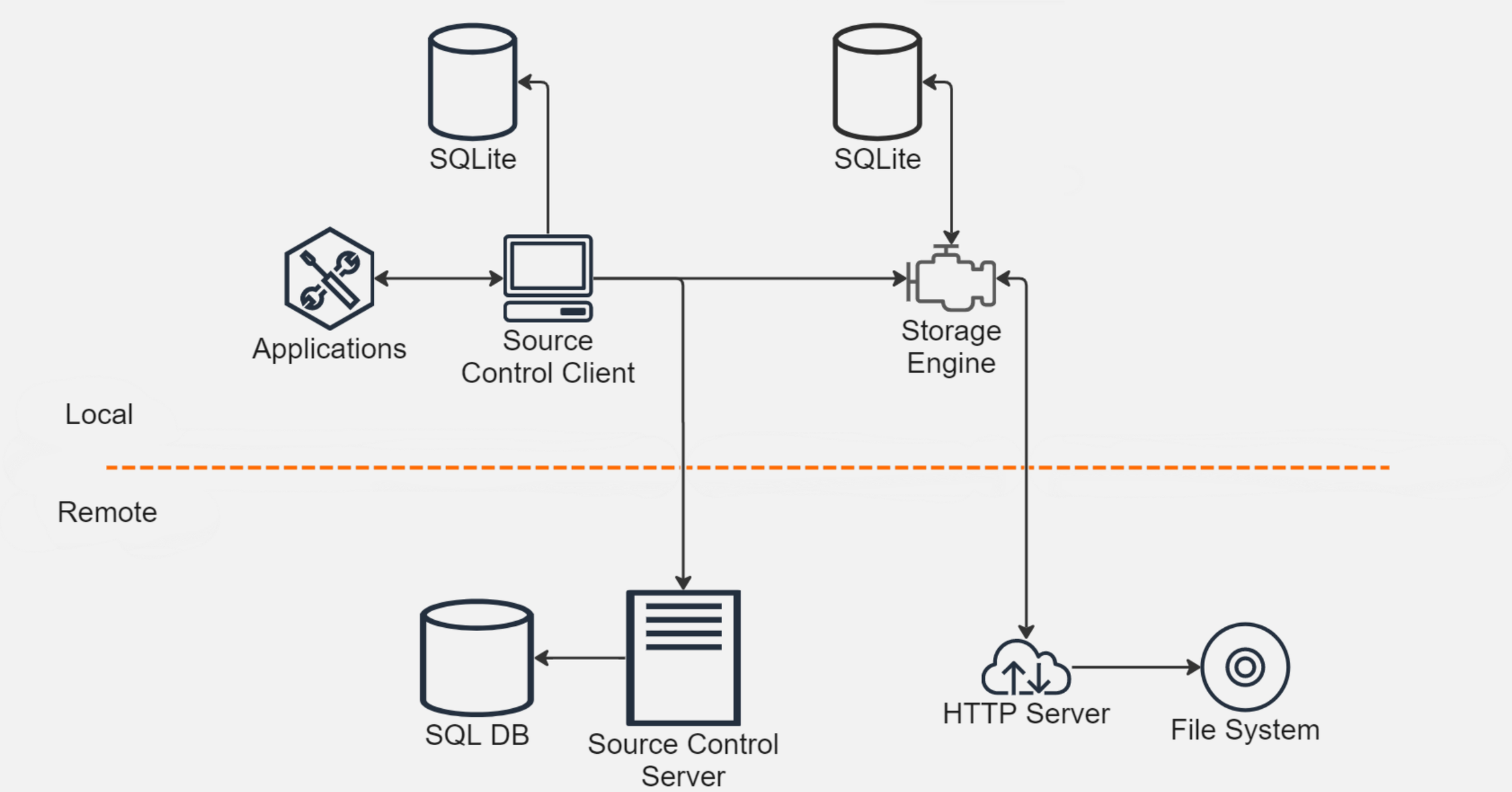

…83b3437cdcd6ec279031586d890e1
b4e1d5c2b2e02757fa0ca807e6fc3052
518c8cdcd66d6e71ae54296d80f09af3
65de5e42946071a56da93d5e89738e5b
68ac99a85b10cdcdbb9d9748d27d6…

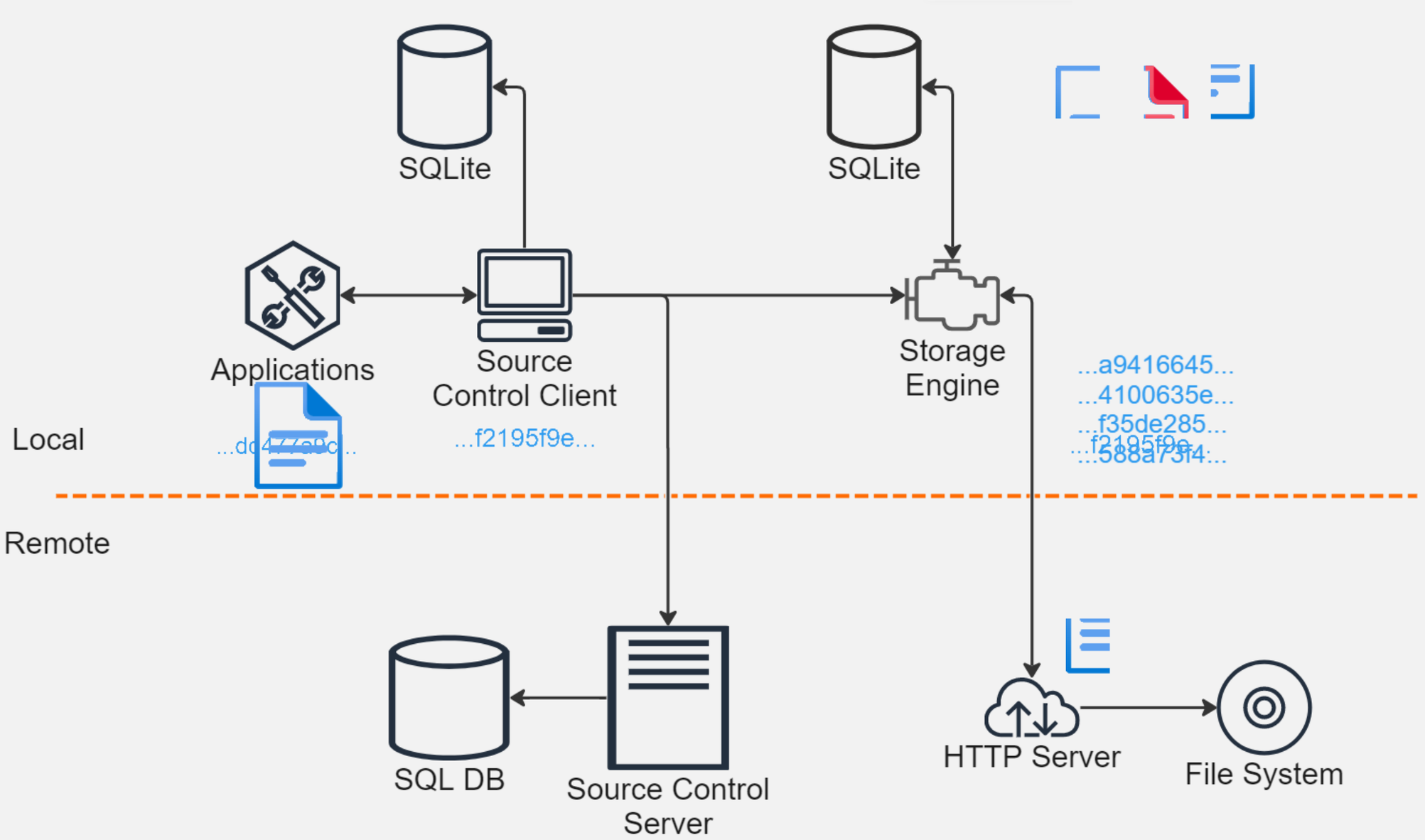
Implementation

- Represent a file as a Manifest of chunks
  - Header (4cc, version, chunk count)
  - Array of chunk addresses
- Chunk’s address is SHA1 hash after compression
- Target chunk size of 128Kb

Integration

- Source control is oblivious to file storage
  - All it has is an address
- Asset address can be either a file, or a Manifest
- CDC can be implemented with no API changes


SQLite
SQLite
Applications
Source
Control Client
Storage
Engine
Local
Remote
SQL DB
Source Control
Server
HTTP Server
File System


SQLite
SQLite
Applications
Source
Control Client
...f2195f9e...
Storage
Engine
...a9416645...
...4100635e...
...f35de285...
...f2195f9e...
...588a73f4...
Local
Remote
SQL DB
Source Control
Server
HTTP Server
File System

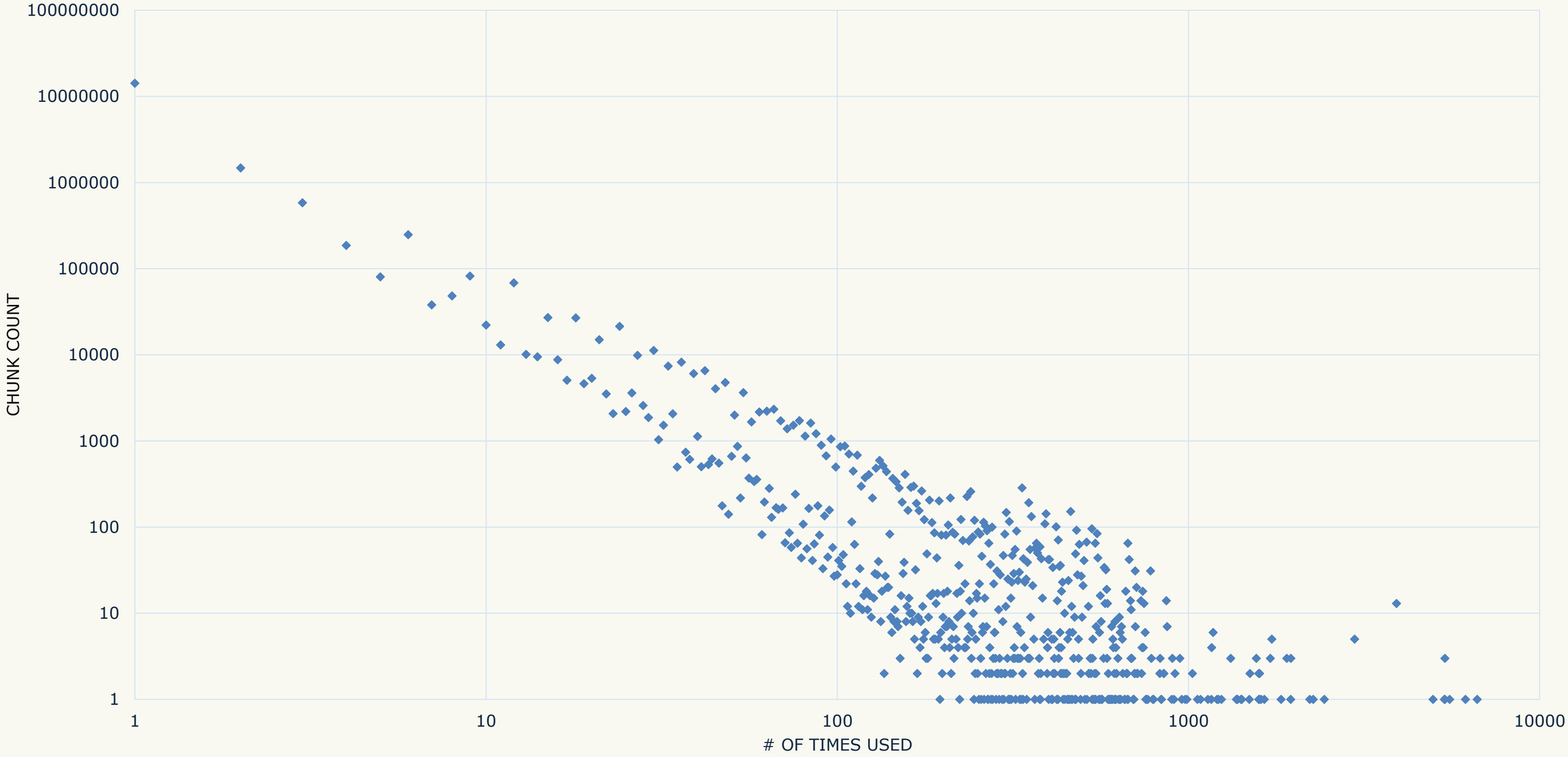
Results

- Unexpected wins through deduplication
- Branch data reduced 5Tb->1.27Tb (75%)
- Problematic files saw the biggest wins
  - PSD 2.91Tb -> 296Gb (92% saving)
  - Maya 630Gb->142Gb (78% saving)
  - TIFF 43Gb ->10Gb (76% saving)
- Some asset types saw less savings

Incremental Results

- 700Gb of data committed monthly
- Reduced to 100Gb of new chunk data (84% saving)
- Again, savings vary wildly by type
  - PSD files see average savings of 90% (So editing a 1Gb psd only requires 100Mb of changes to be uploaded)
  - Other types like audio source files only see savings in the range of ~30%

CHUNK REUSE


100000000
10000000
1000000
100000
10000
1000
100
10
1
CHUNK COUNT
1
10
100
1000
10000
OF TIMES USED

Performance

- Storage Engine backed by SQLite
- Read speed can exceed theoretical maximum of the hardware
  - Duplicated chunks only need to be read once

Conclusion

- Reduced file transfer by an order of magnitude
- More caching gives an even better UX
- Investing in tech paid off

Contributors

- Even Braudaway
- Luke Mordarski
- Jesse Blomberg
- David Clyde
- Phil Orwig

GDC | March 20-24, 2023
San Francisco, CA

Overwatch Data Pipeline
tinyurl.com/owdpl

Fast CDC
tinyurl.com/owfastcdc

Quick CDC
tinyurl.com/owquickcdc

Rapid CDC
tinyurl.com/owrapidcdc