

Device Transparency

A New Model for Mobile Storage

Jacob Strauss

Chris Lesniewski-Laas, Justin Mazzola Paluska,
Bryan Ford, Robert Morris, Frans Kaashoek

HotStorage

October 11, 2009

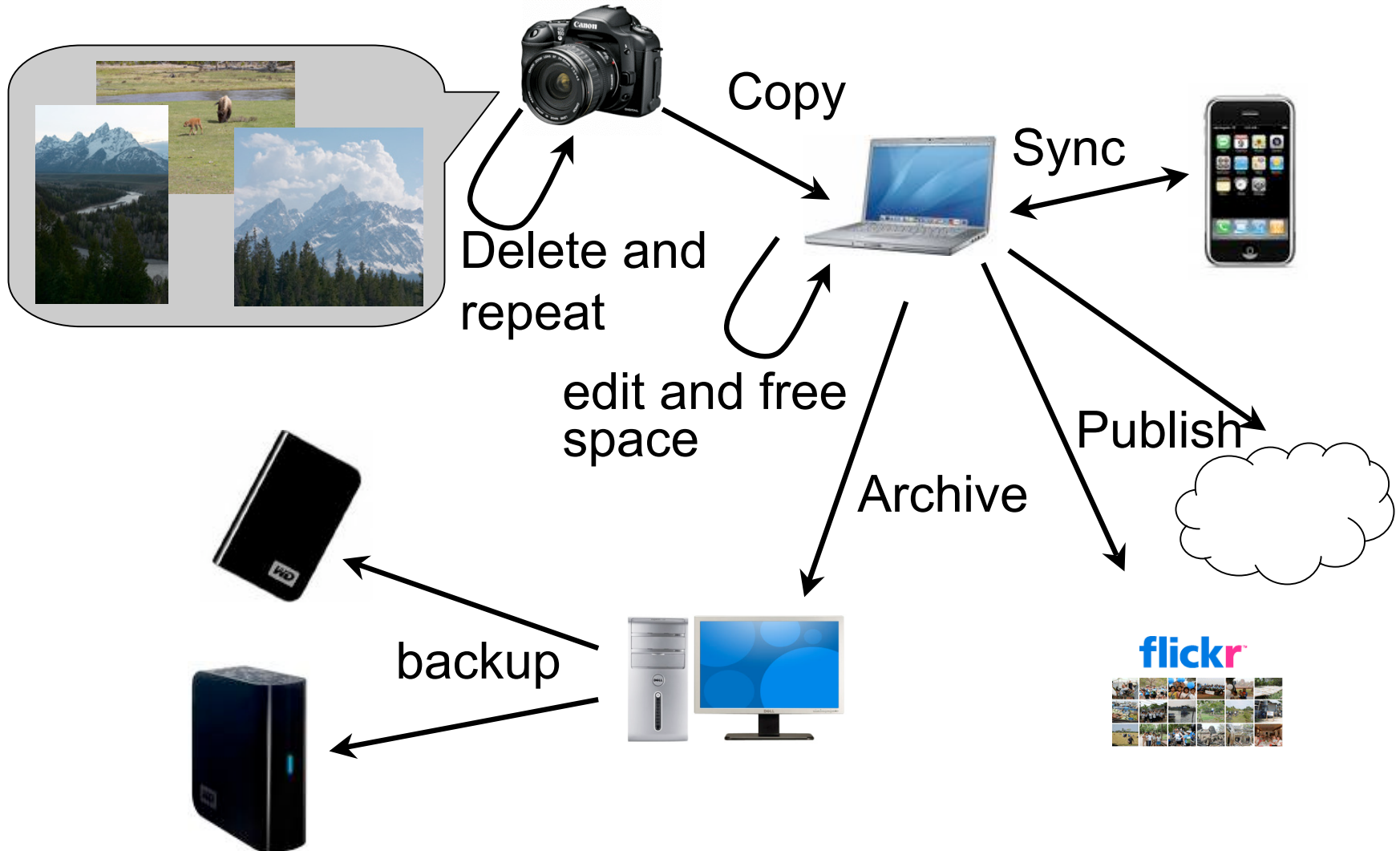
Overview

- Problem: No way to manage data across a group of mobile personal devices
 - Other than manual synchronization or central server
- Goal: Device Transparent Storage
 - The same global view from each device

Photos start on a single camera



photos end up on many devices



Where is a specific photo?

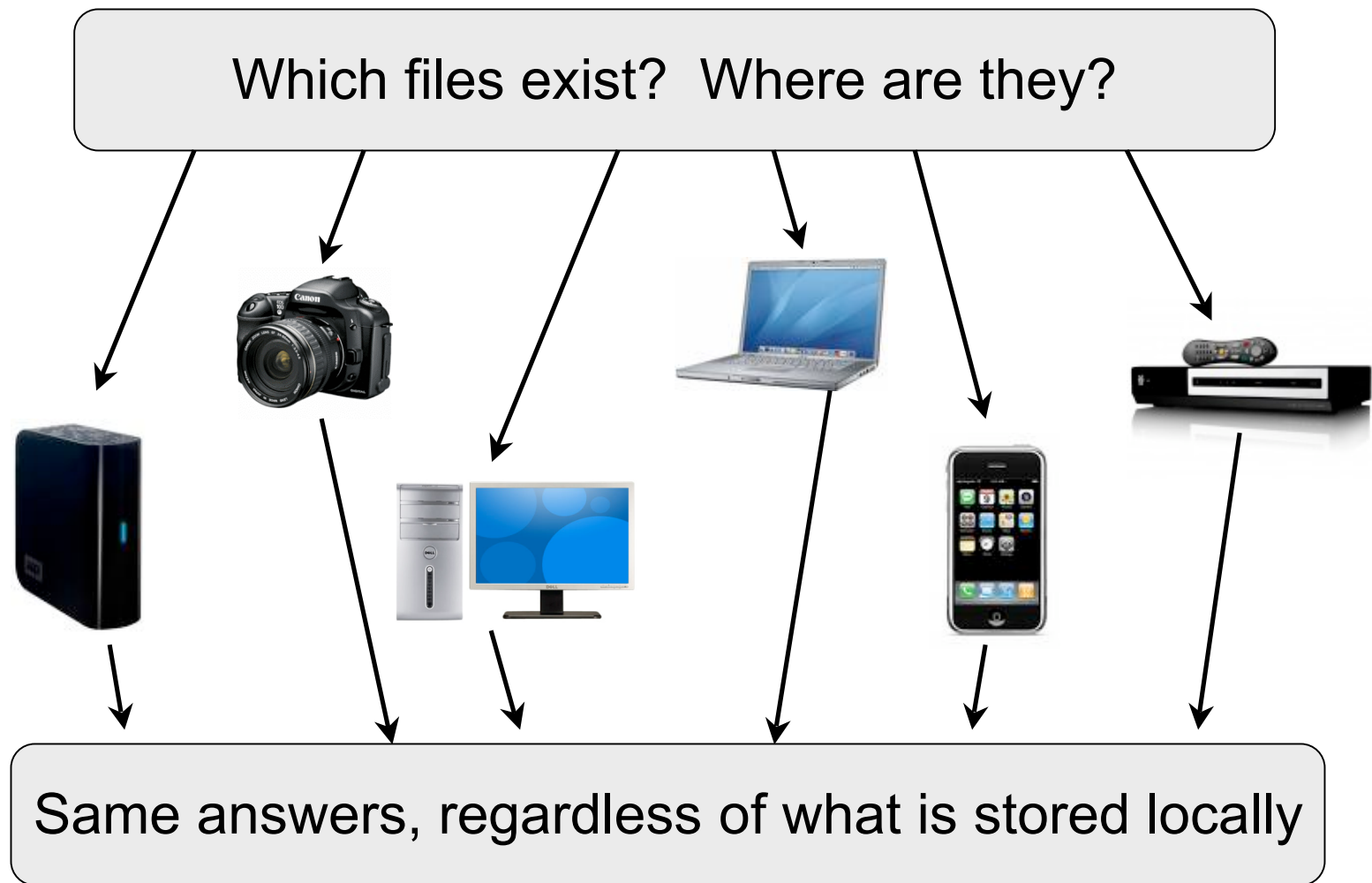
- Not all devices have the complete collection
- Archive may be incomplete or unreachable (if there is one)
- Which version of photo?



Acadia, Field of Lupines



Goal: Device Transparent Storage

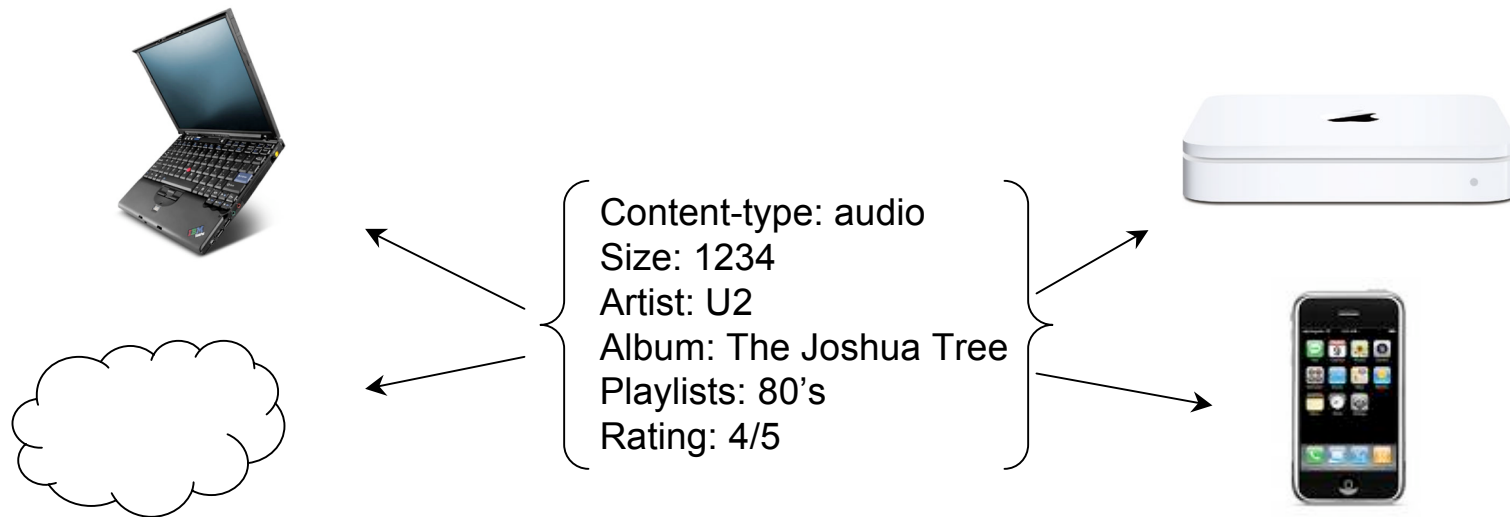


Challenges

- Disconnected Operations
 - Not always connected to network
 - Cannot rely on a central repository
 - Consistency?
- Can't fit complete collection on all devices
- Might not be able to fit complete collection on any single device

One Approach: Metadata Everywhere

- Metadata
 - everything needed to name and find objects
 - album, song name, artist, location, etc.
- Separate content from object metadata
- Replicate Metadata on all of user's devices
 - Each device knows about all objects



Good fit for personal data

- Files are Large
 - New files even larger
 - Mostly write-once
- Metadata much smaller
 - tags, folders, thumbnails
 - Sufficient to locate an object
 - Metadata does change over time
- Property of many data types
 - Music, Photos, Videos, Email



Eventual Consistency is Enough for Metadata

- Objects written-once, read-many
 - Metadata: frequent updates
 - Content: almost never updated
- No Write-Write content conflicts
 - New versions result in content copy
- Metadata updates usually combinable
 - Message status flags, song ratings, playcounts

Eyo, work in progress

- A metadata-everywhere storage system
- Design Highlights
 - Application storage API
 - Metadata synchronization
 - Content synchronization

Eyo: Storage API

- Queries over metadata to locate files

ID_list ← lookup(*query*)

- Expose object version history to applications

version_list ← get_versions(*ID*)

- Applications can add a replacement version to resolve conflicts

new_version(*ID, metadata, predecessor_versions*)

Eyo: Metadata Synchronization

- Continuous background synchronization
 - Overlay Network to locate devices
 - Exchange update log whenever possible
- Truncate history once all devices agree
- Only exchange new information

Eyo: Content Synchronization

- Send content separately after metadata synchronization
 - Bulk transfer unlike metadata
- Placement Rules
 - Map query over metadata to device(s)
 - Ex: All “photos” tagged “SOSP” → iphone
 - Allow applications to specify which device files should go to
- Status notifications to inform applications of new updates

Related Work

- Many shared mechanisms, but not shared goal
- Optimistic Replication
 - Cimbiosys & Polygraph
 - Perspective
 - TierStore
 - PRACTI
 - EnsemBlue
 - Coda
 - Bayou
- Cloud Storage Systems
 - MobileMe
 - Live Mesh

Summary

- Device Transparent Storage
 - View and Manage complete collection from any device
 - Separate metadata from content
 - Globally replicate metadata
 - Partially replicate content
 - Push updates as soon as possible
 - Change applications to hide conflicts

Backup Slides

Discussion Topics

- What happens when there's a lot of metadata?
Nothing bad
- What happens when devices run out of space?
Nothing bad
- A new storage API? Will application writers really buy into that?
They already are
- I can't use my entire photo collection on my phone -- why should it need to know about all of it?
Carrying is still useful to other devices, and "can't use" is probably wrong.
- Multiple Users?
Yes?
- What does a placement rule look like?
Users never see these
- Doesn't cloud storage obviate this whole problem?
No

Metadata is Small

Illustrative Example: One person's data collection

	# objects	total size	MD size	MD per object / overhead
Email	724,230	4.3 GB	169.3 MB	245B / 3.8%
Music	5,278	26 GB	5.1 MB	511B / 0.01%
Photos	72,380	122.8 GB	22.6 MB	328B / 0.02%

Metadata proportional to # of objects, not type

Metadata Grows (at most) Slowly

