SCFS: A Shared Cloud-backed File System

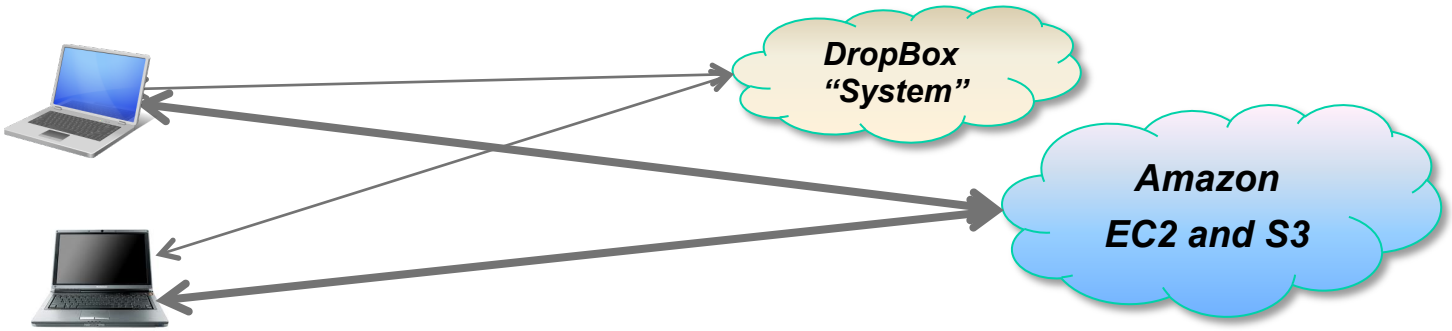
Alysson Bessani, Ricardo Mendes, Tiago Oliveira, Nuno Neves, Miguel Correia, Marcelo Pasin*, Paulo Veríssimo

University of Lisbon - Portugal

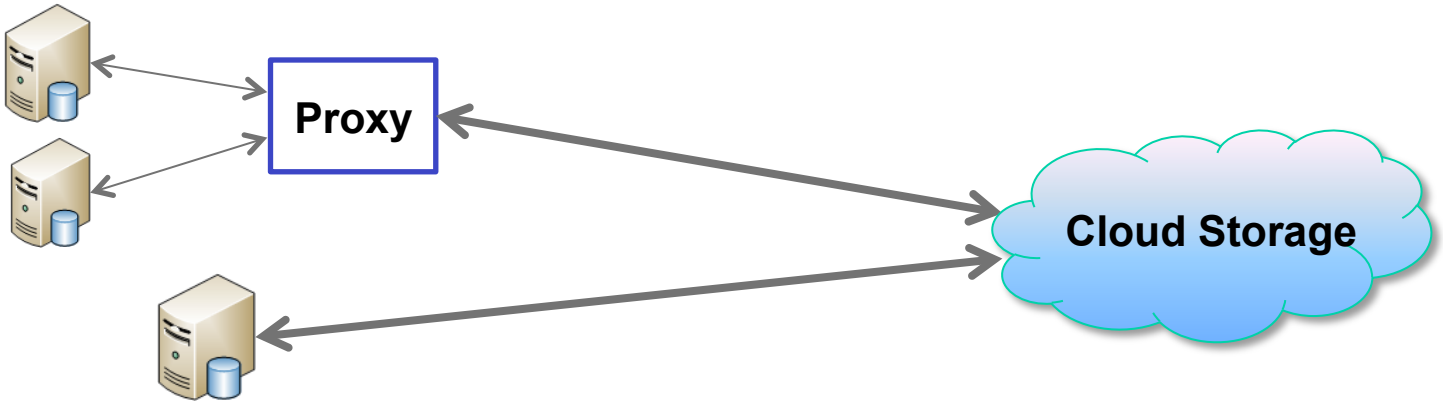
** now at University of Neuchatel - Switzerland*

Cloud-backed Storage

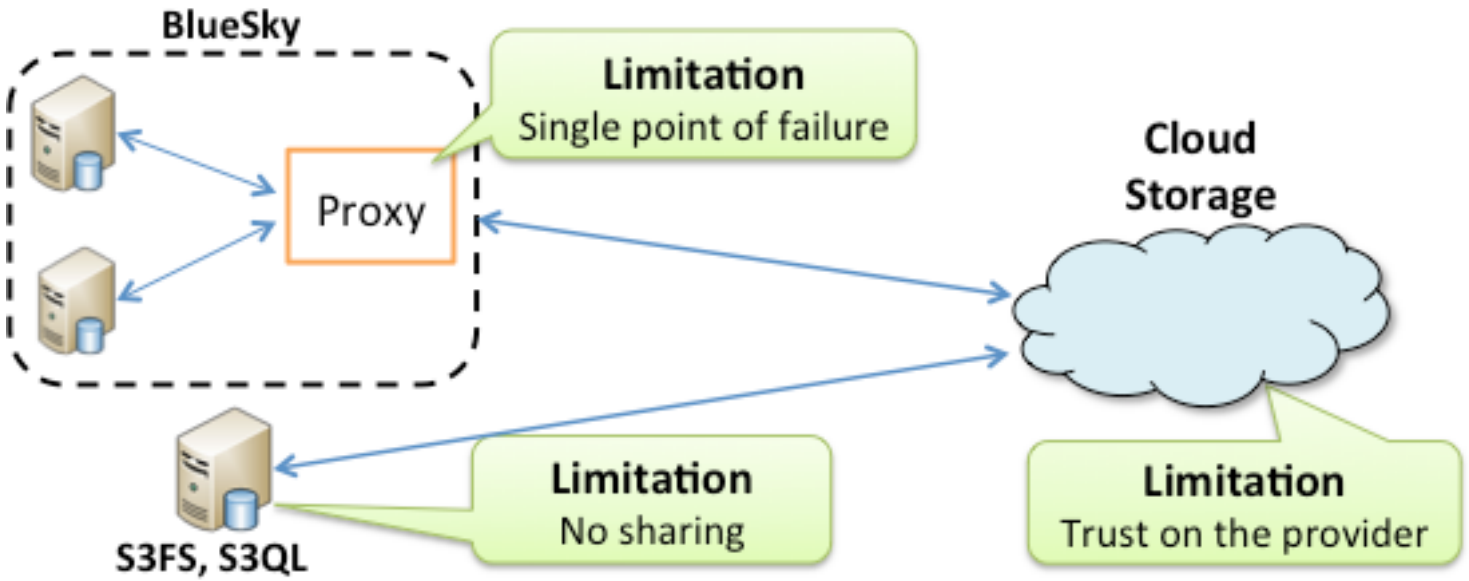
Personal Storage Services (e.g., DropBox, OneDrive)



Cloud-backed File Systems (e.g., BlueSky, S3FS)



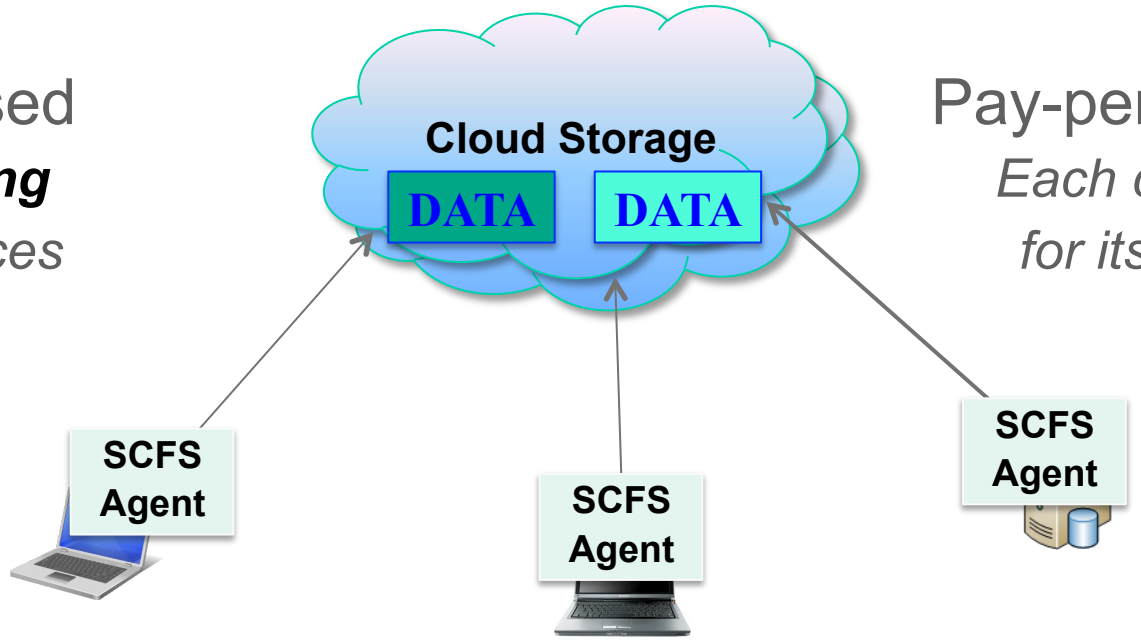
State of the Art



Shared Cloud-backed File System (SCFS)

Client-based

*Use **existing** cloud services*



Pay-per ownership

*Each client **pays** for its own files*

Strong Consistency

Controlled sharing

Access control for security and concurrency

Redundant Cloud Services

Navigators

FACULDADE DE CIÊNCIAS UNIVERSIDADE DE LISBOA

SCFS Design

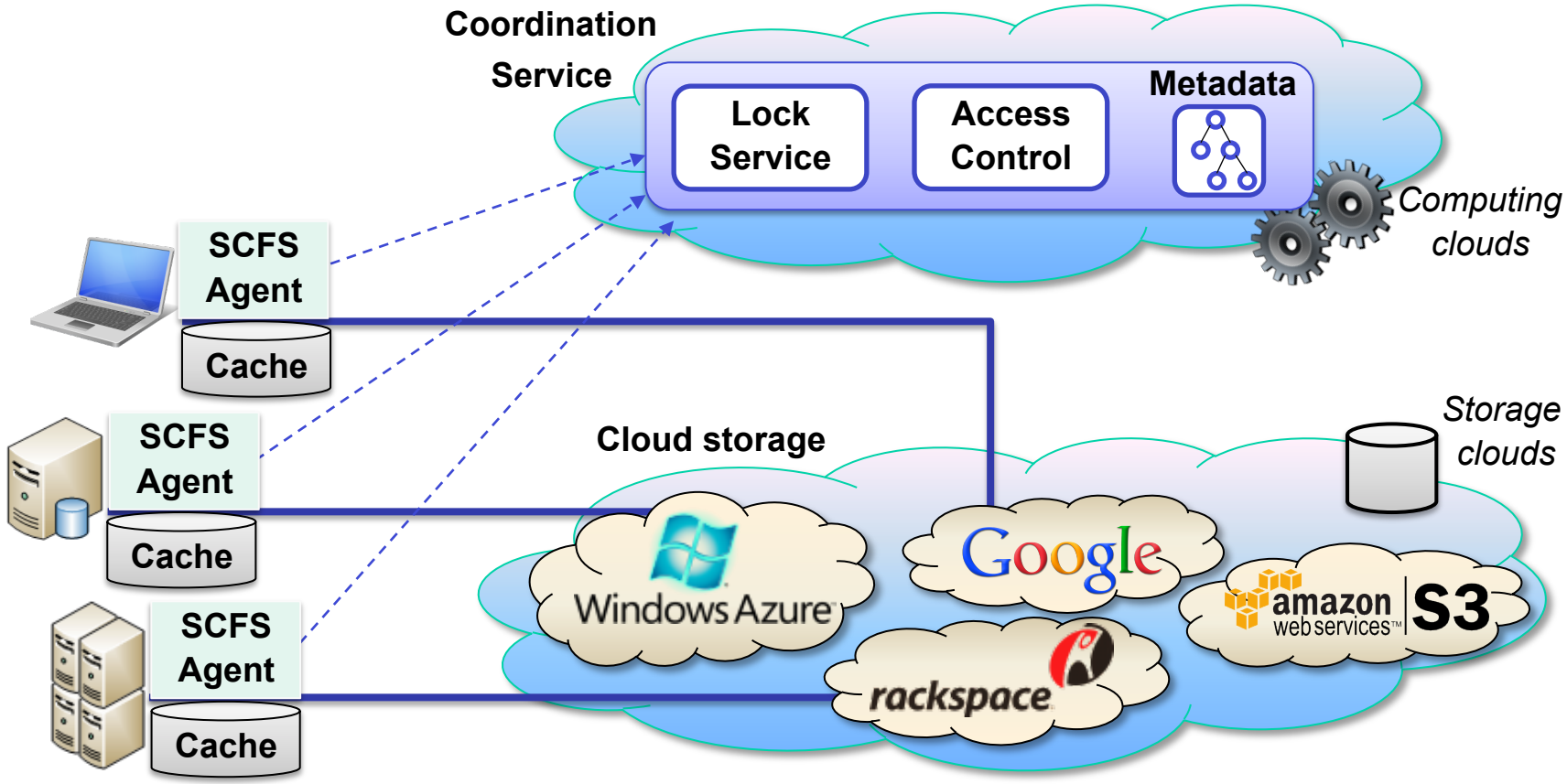
Design Choices

- **Data layout/access pattern**
  - Each file is an object (single-block file)
  - Multiple versions of the files are maintained
  - Always write, avoid reading (exploiting free writes)
- **Cache**
  - Persistent file cache
    - Local storage is used to hold copies of all/most client files
    - Opened files are also maintained in main-memory
  - Short lived main-memory metadata cache
    - To deal with bursts of *metadata* requests

Design Choices

- **Consistency**
  - Consistency-on-close semantics
  - Control of durability and consistency
  - Locks used to avoid write-write conflicts
- **Modular coordination**
  - Separate data from metadata
  - Metadata is stored in a coordination service
    - E.g., Zookeeper [ATC'10], DepSpace [EuroSys'08]
  - Also used for managing file locks

SCFS Architecture



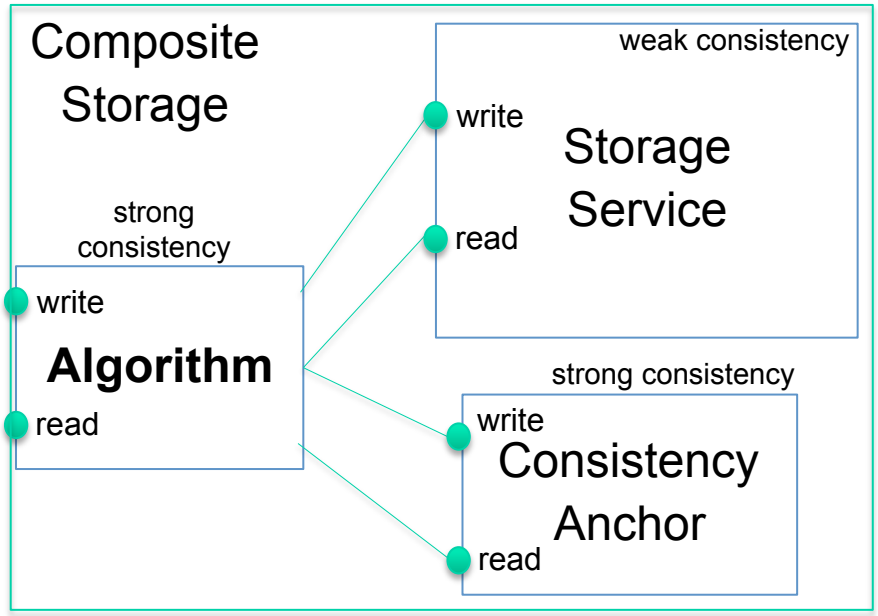
Navigators

FACULDADE DE CIÊNCIAS UNIVERSIDADE DE LISBOA

Consistency

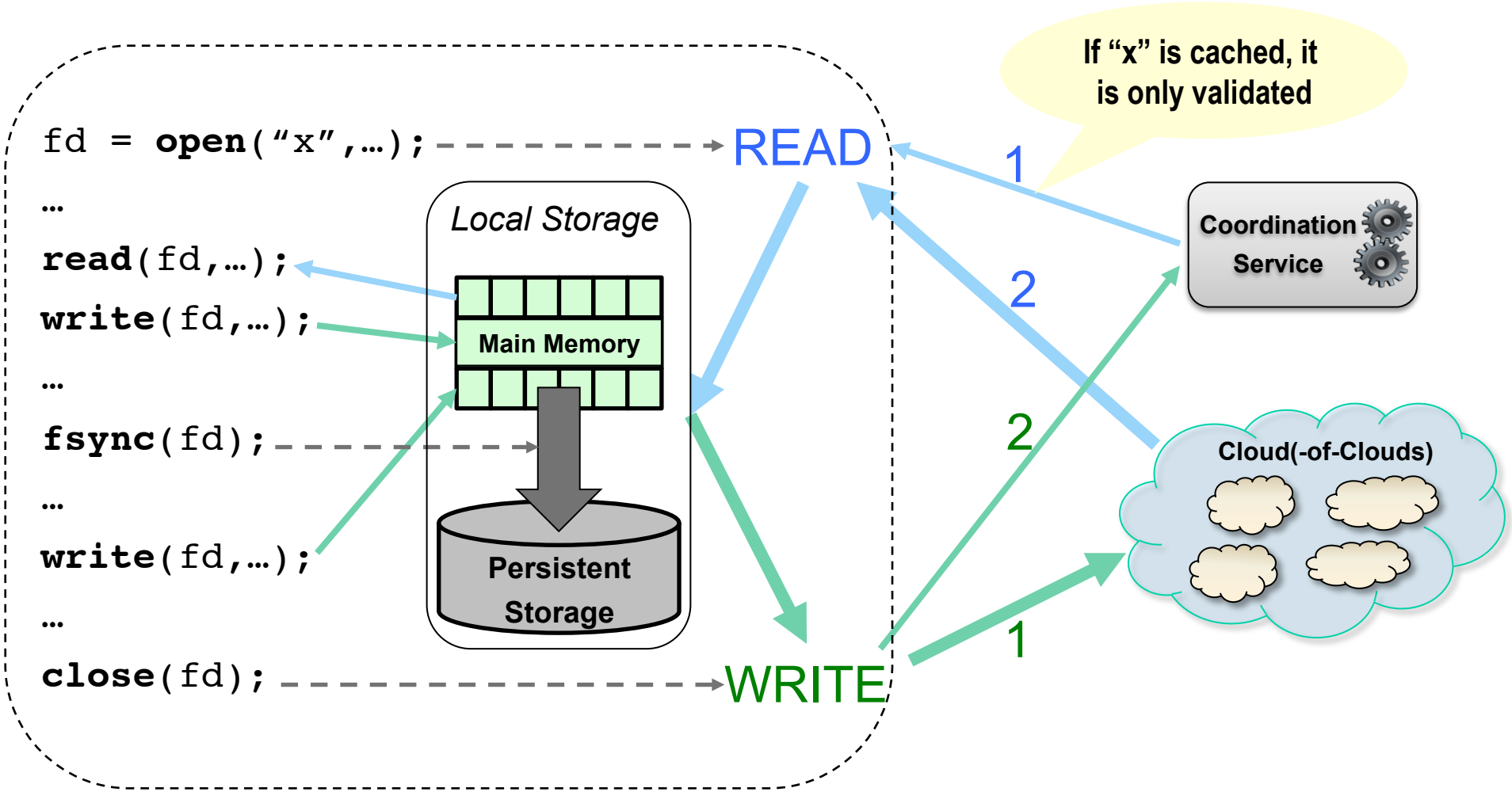
Consistency Anchor

- **Problem:** *How to provide strong consistency on top of weak consistent storage clouds?*



- **Key property:** the composite storage' consistency is the same of the consistency anchor

Consistency Anchor in SCFS



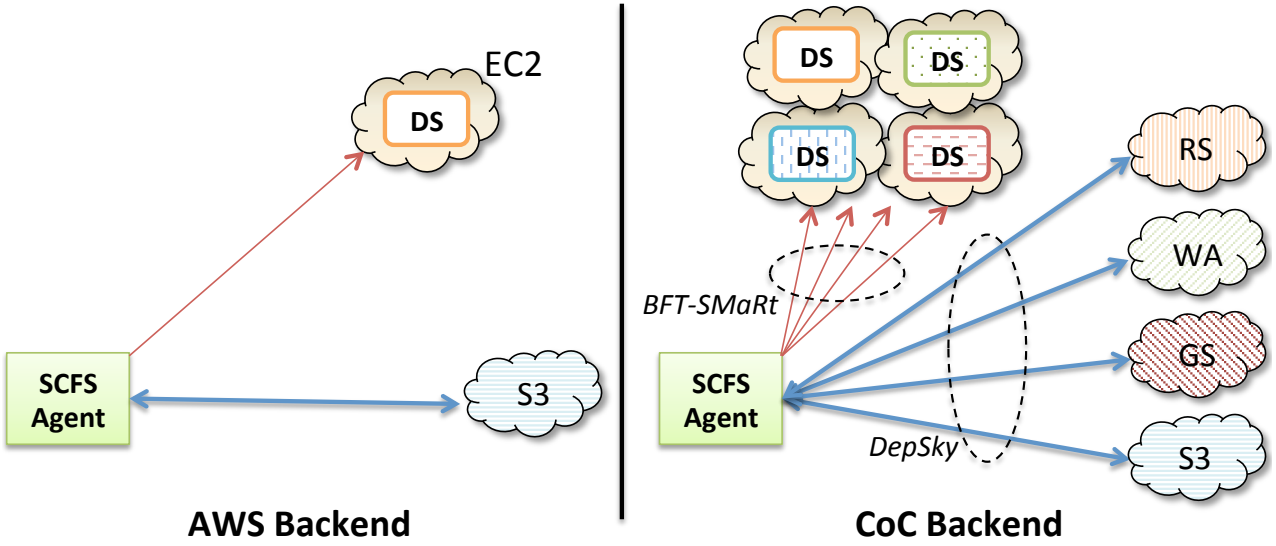
Navigators

FACULDADE DE CIÊNCIAS UNIVERSIDADE DE LISBOA

Implementation and Evaluation

SCFS Backends

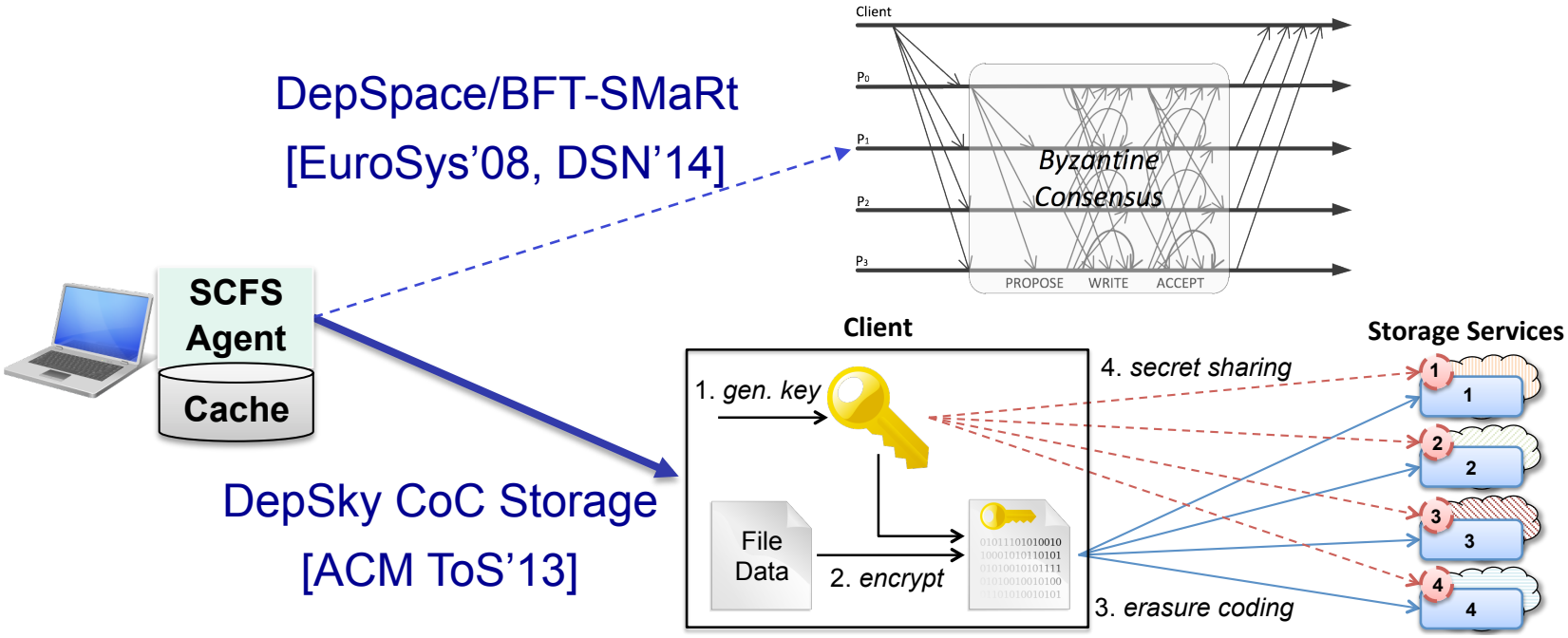
- SCFS can use different backends
  - i.e., different cloud storage and a coordination service plugin



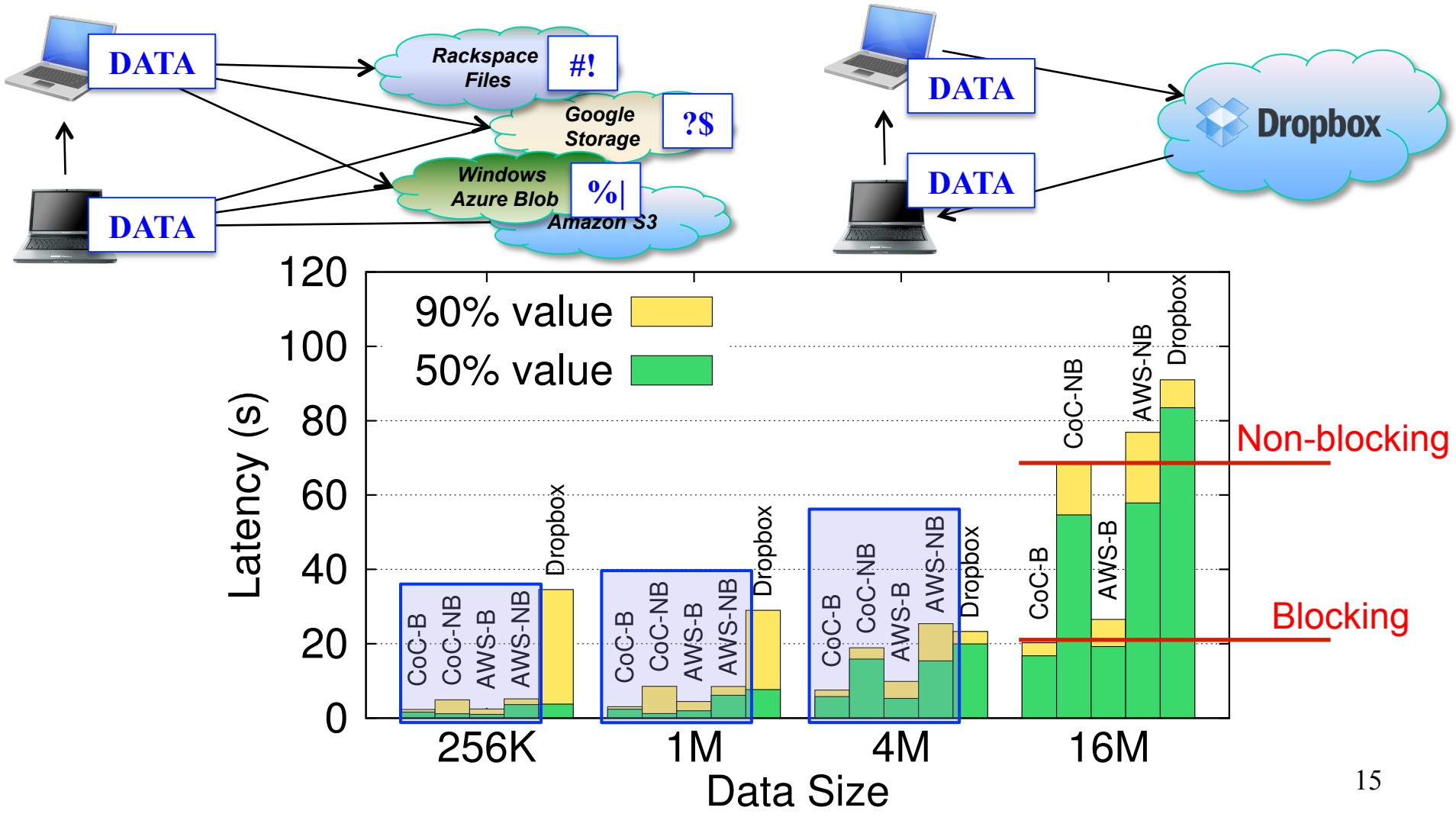
- Operation: **blocking**, non-blocking and non-sharing

The Cloud-of-Clouds Backend

- Does not require trust on any single cloud provider
  - SCFS works correctly as long as less than a third of the providers misbehave



Sharing Latency: SCFS vs DropBox



Benchmarking (Unmodified) Desktop Applications

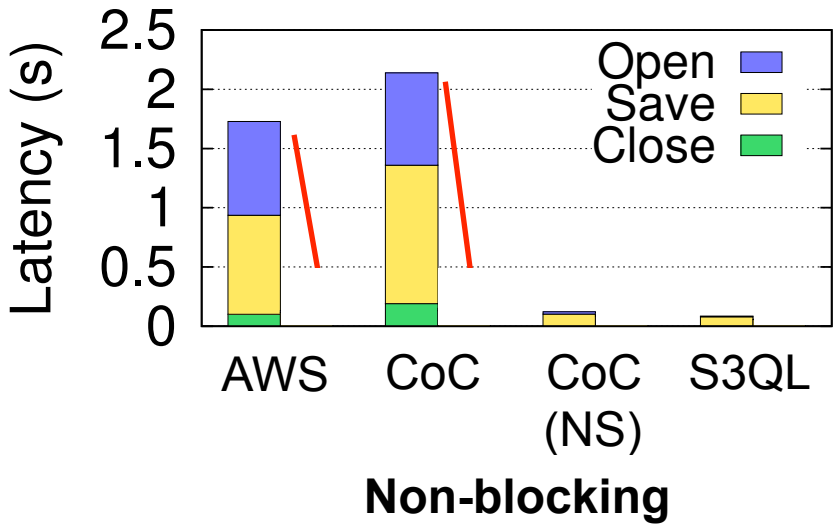
OpenOffice Writer

Open Action: 1 open(f,rw), 2 read(f), 3-5 open-write-close(lf1), 6-8 open-read-close(f), 9-11 open-read-close(lf1)

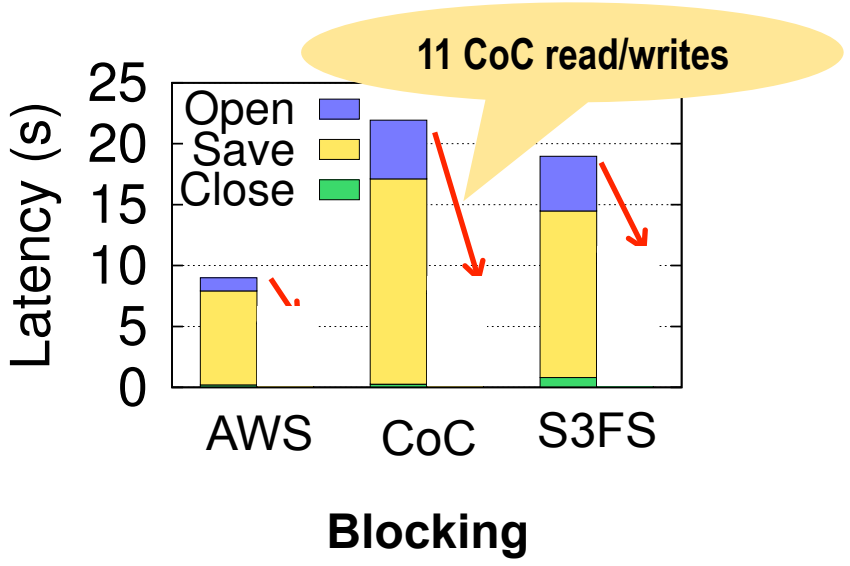
Save Action: 1-3 open-read-close(f), 4 close(f), 5-7 open-read-close(lf1), 8 delete(lf1), 9-11 open-write-close(lf2), 12-14 open-read-close(lf2), 15 truncate(f,0), 16-18 open-write-close(f), 19-21 open-fsync-close(f), 22-24 open-read-close(f), 25 open(f,rw)

Close Action: 1 close(f), 2-4 open-read-close(lf2), 5 delete(lf2)

55% 40% 80% — *lock files*



Non-blocking

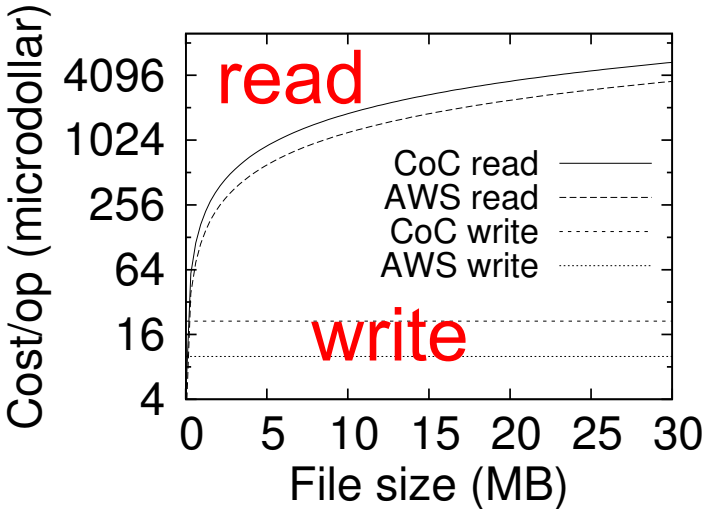


Blocking

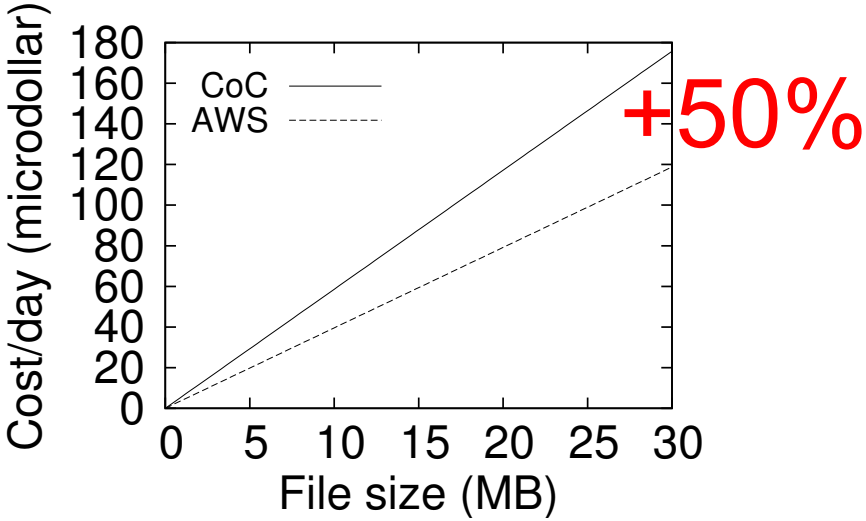
Financial Evaluation

VM Instance	EC2	EC2×4	CoC	Capacity
Large	$6.24	$24.96	$39.60	7M files
Extra Large	$12.96	$51.84	$77.04	15M files

(a) Operation costs/day and expected coordination service capacity.



(b) Cost per operation (log scale).



(c) Cost per file per day.

Wrap Up

- SCFS is a cloud-backed file systems that can be used for backup, disaster recovery and sharing data
- Key design principles:
  - Always write, avoid reading (very cheap in terms of $$$)
  - Strong consistency (despite storage cloud' weak consistency)
- Experience so far…
  - Multi-cloud replication is feasible (CoC not slower!)
    - This is a case for BFT… crash-only solution will not make things better
  - Being employed for sharing dataset metadata among Biobanks

Thanks!

- **SCFS code available at**

  http://code.google.com/p/depsky/wiki/SCFS

- DepSky and DepSpace/BFT-SMaRt also available

  http://code.google.com/p/depsky/

  http://code.google.com/p/bft-smart/