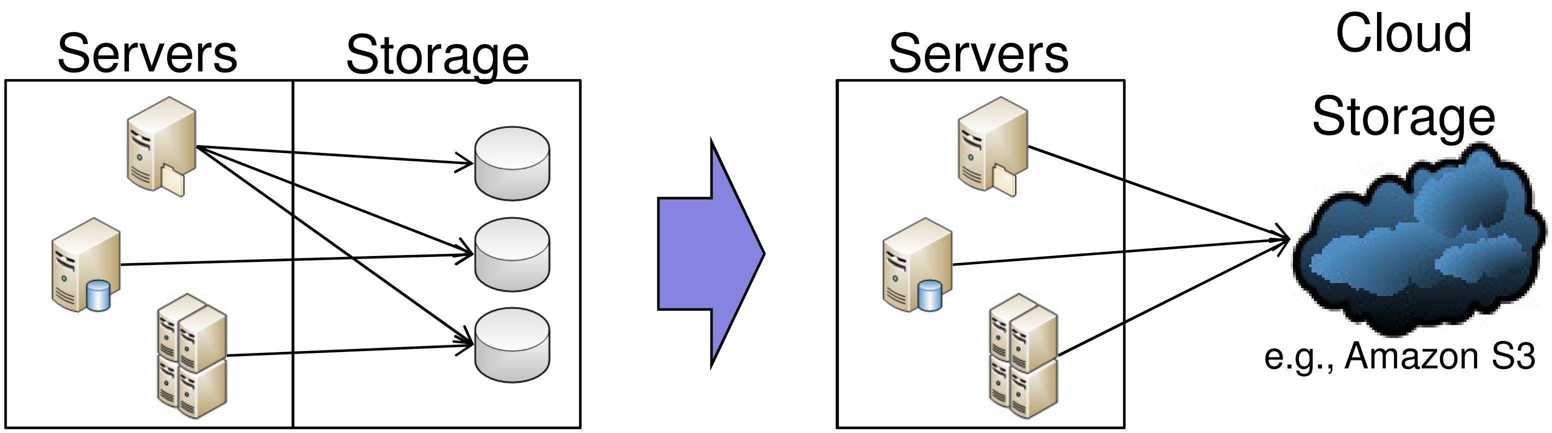
DepSky – Dependable and Secure Storage in a Cloud-of-Clouds

<u>Alysson Bessani</u>, Miguel Correia, Bruno Quaresma, Fernando André, Paulo Sousa

University of Lisbon, Faculty of Sciences

Moving to Clouds

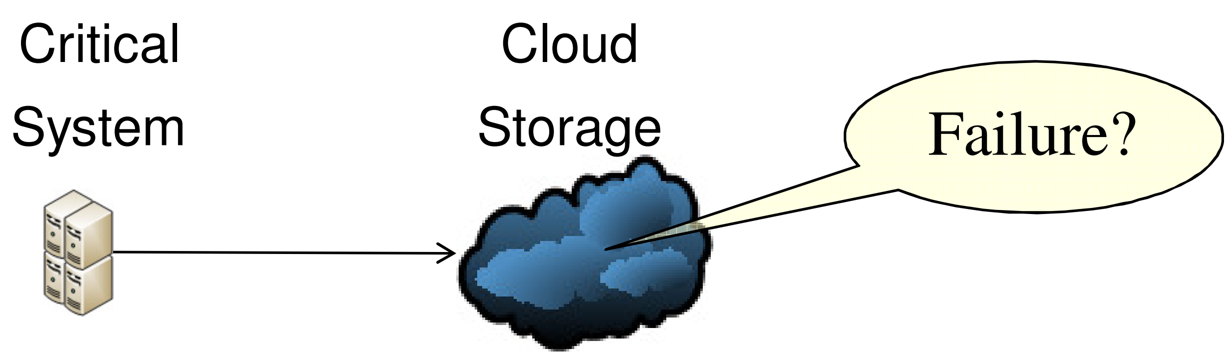
- Data is moving to the cloud



- Main reason: costs (pay-per-use model)

One Cloud is Not Enough

- Critical data storage
  - Medical records
  - Company financial data
  - Critical infrastructures data

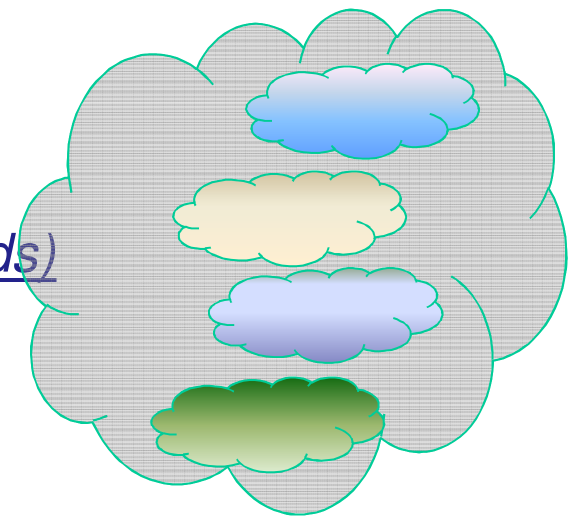


Trusted Clouds

- Two options:

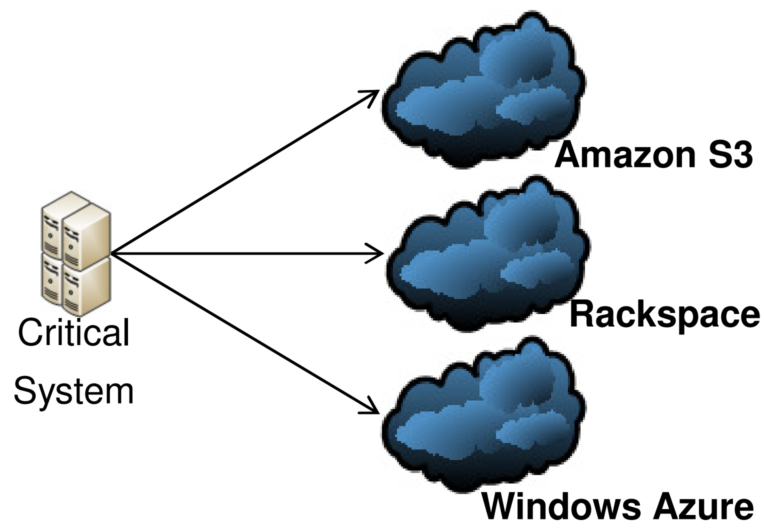
  1) Improve the cloud infrastructure

  2) Use several cloud providers (cloud-of-clouds)

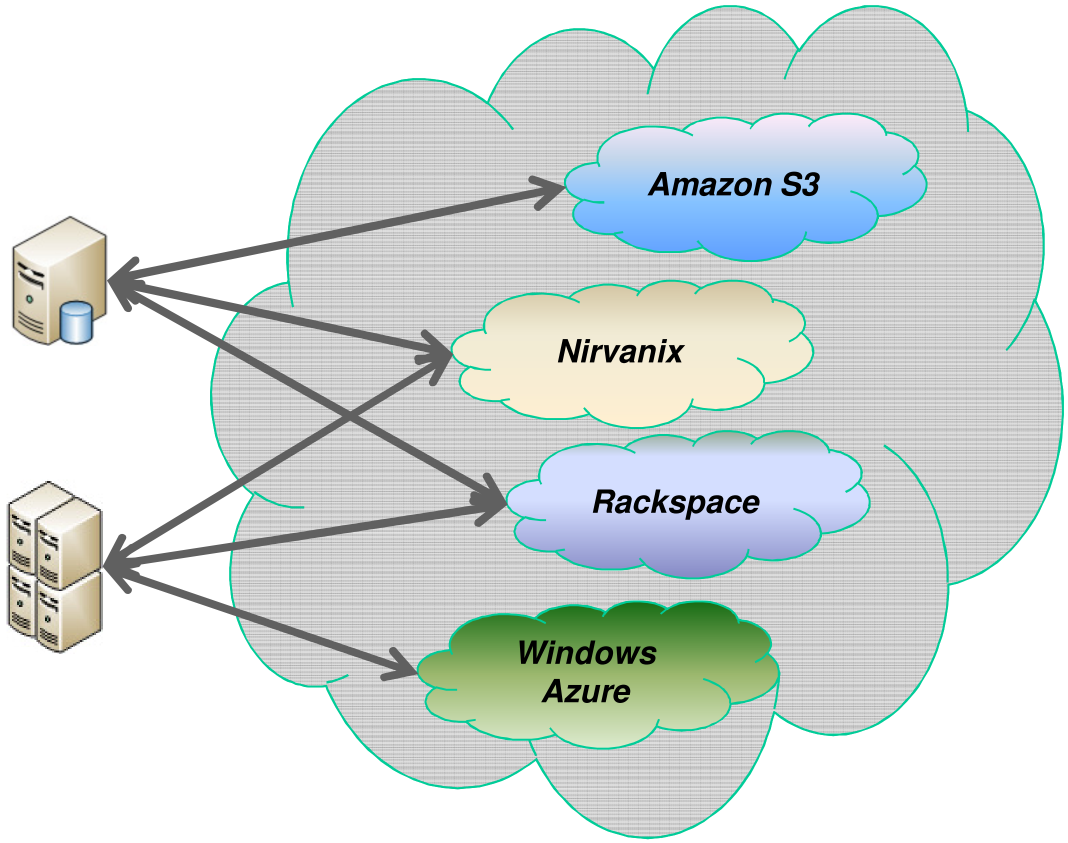


Benefits of Replication

- Datacenter and cloud outages
- Vendor lock-in
- Better read performance
- Data corruption
  - Bugs
  - Malicious insiders
  - Attacks and intrusions



Cloud-of-Clouds Object Storage



DepSky Design

DepSky Design Principles

1. No trust on individual cloud providers

Distributed trust is built by using multiple clouds

2. Use storage clouds as they are

No server-side code on the replication protocols

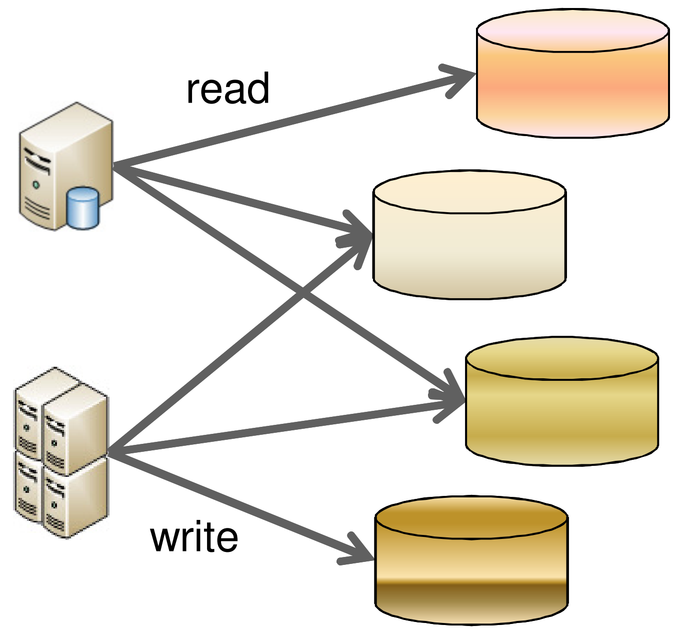
3. Data is updatable

Quorum replication protocols for consistency

Key Challenges

- How to implement an efficient replication protocol using only passive storage nodes?

- How to make it affordable?



DepSky Interface

- **write(data_unit, data)**
- **read(data_unit)**

Object Storage

- *create(data_unit)*
- *destroy(data_unit)*

- *lock(data_unit, …)*
- *unlock(data_unit)*

- *garbageCollect(data_unit, …)*
- *reconfigure(data_unit,…)*

See details on the paper!

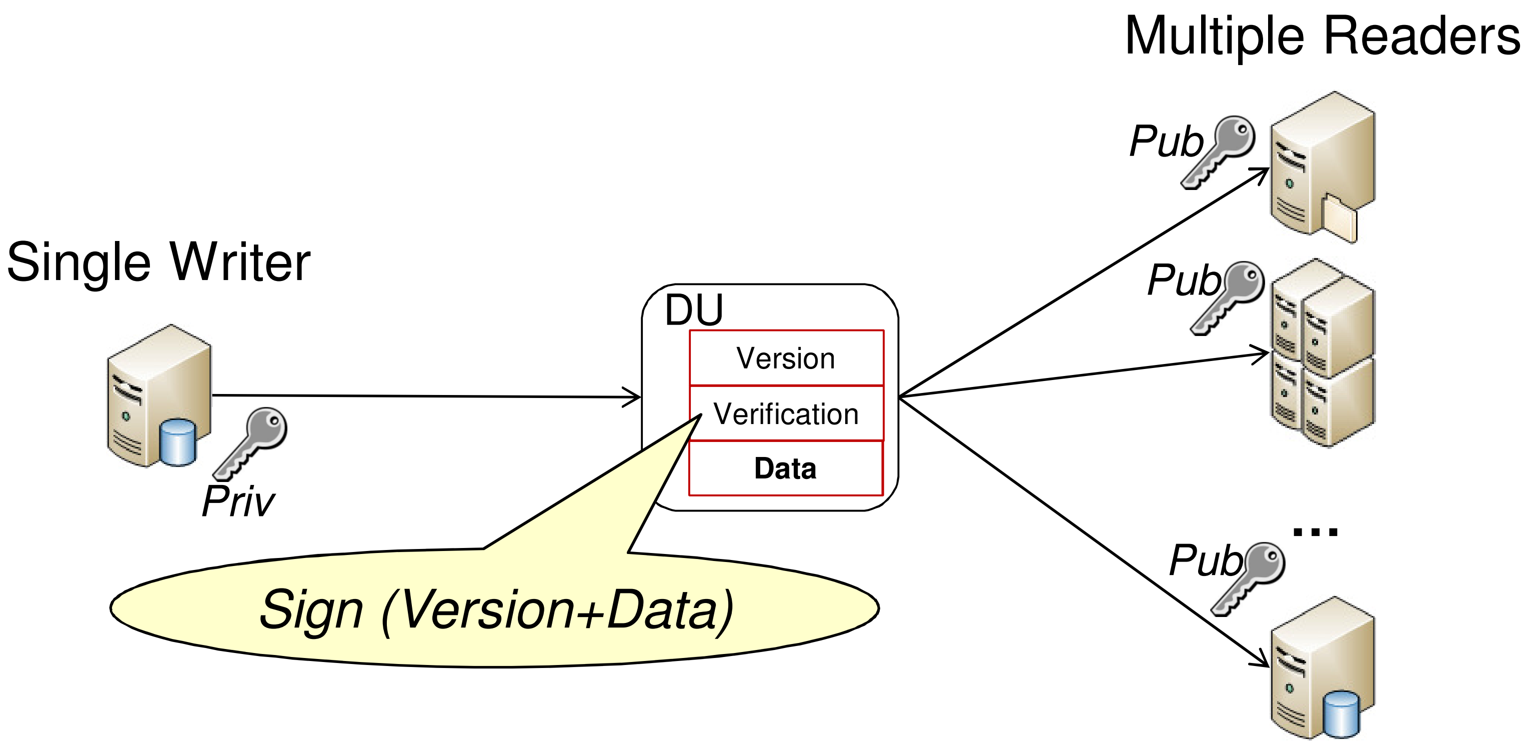
System Model

- Asynchronous distributed system
- Faults
  - Clouds can be unavail… data
  - Readers can do …
  - Writers can **crash and recover**

Byzantine faults

- *n = 3f +1* clouds to tolerate *f* faults
  - In practice: *f = 1*
- Symmetric and asymmetric cryptography

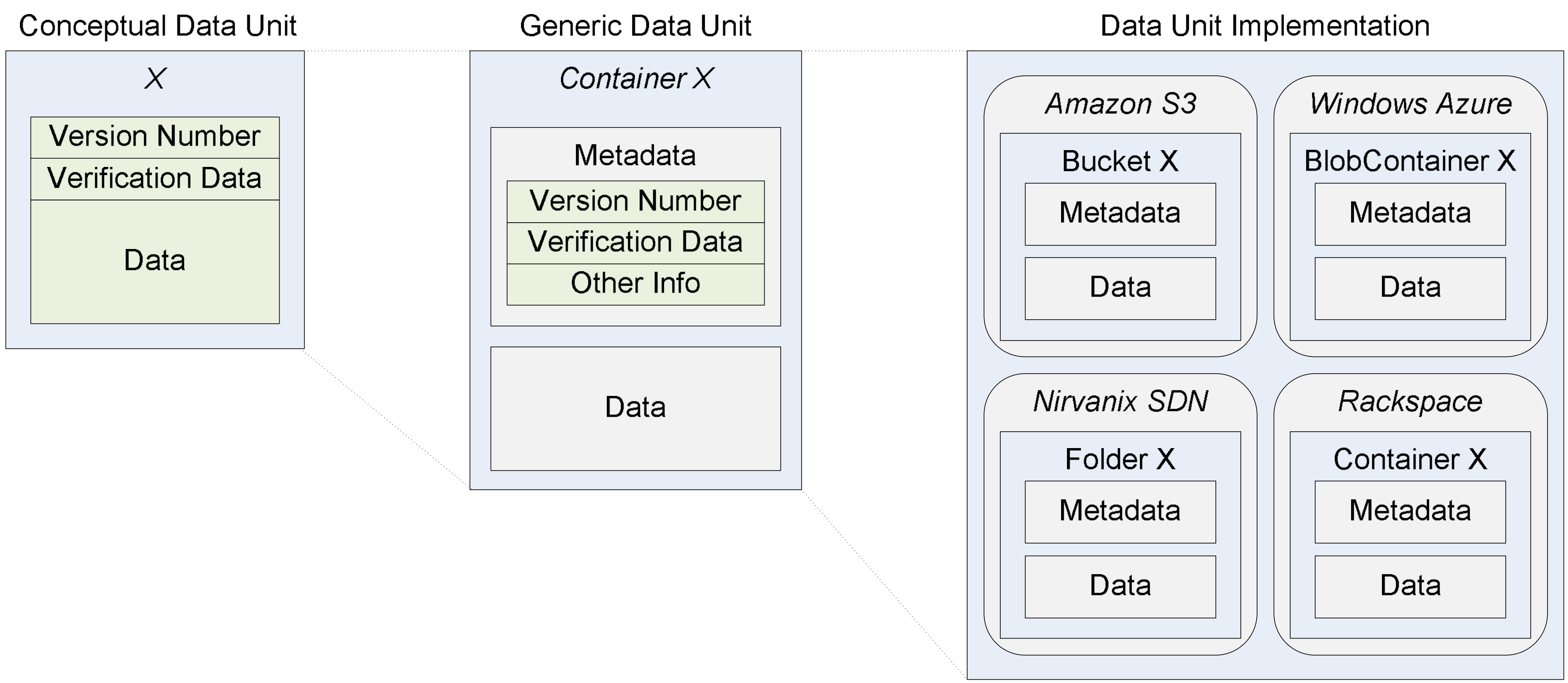
Data Model



single-writer multi-reader regular register

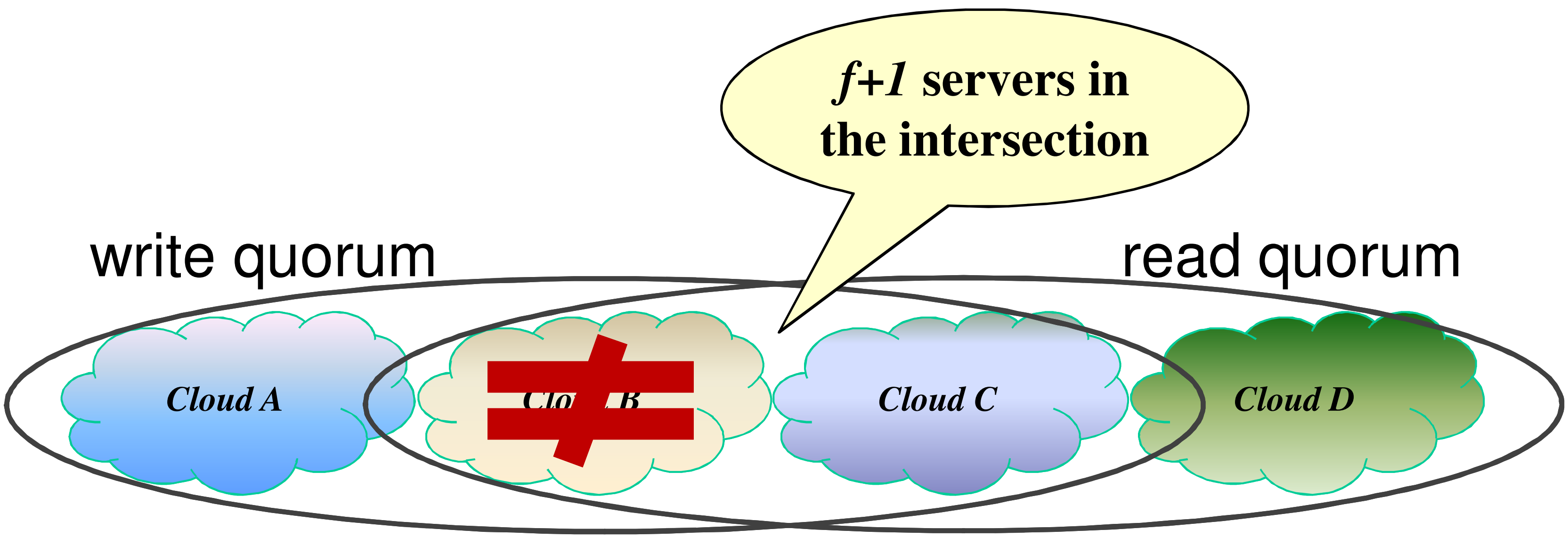
(but multiple writers are supported through a locking algorithm)

Data Model Implementation

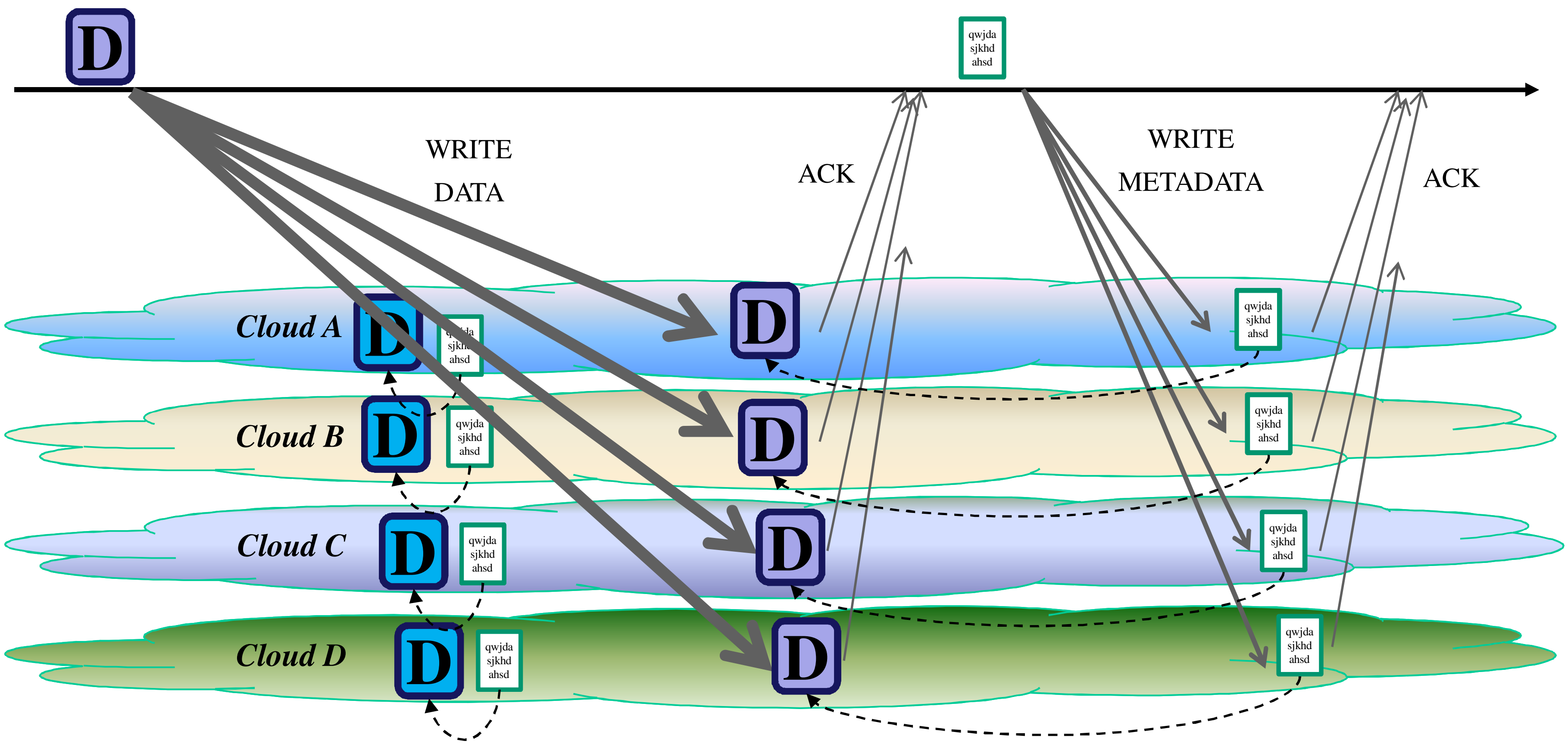


Read/Write Protocols

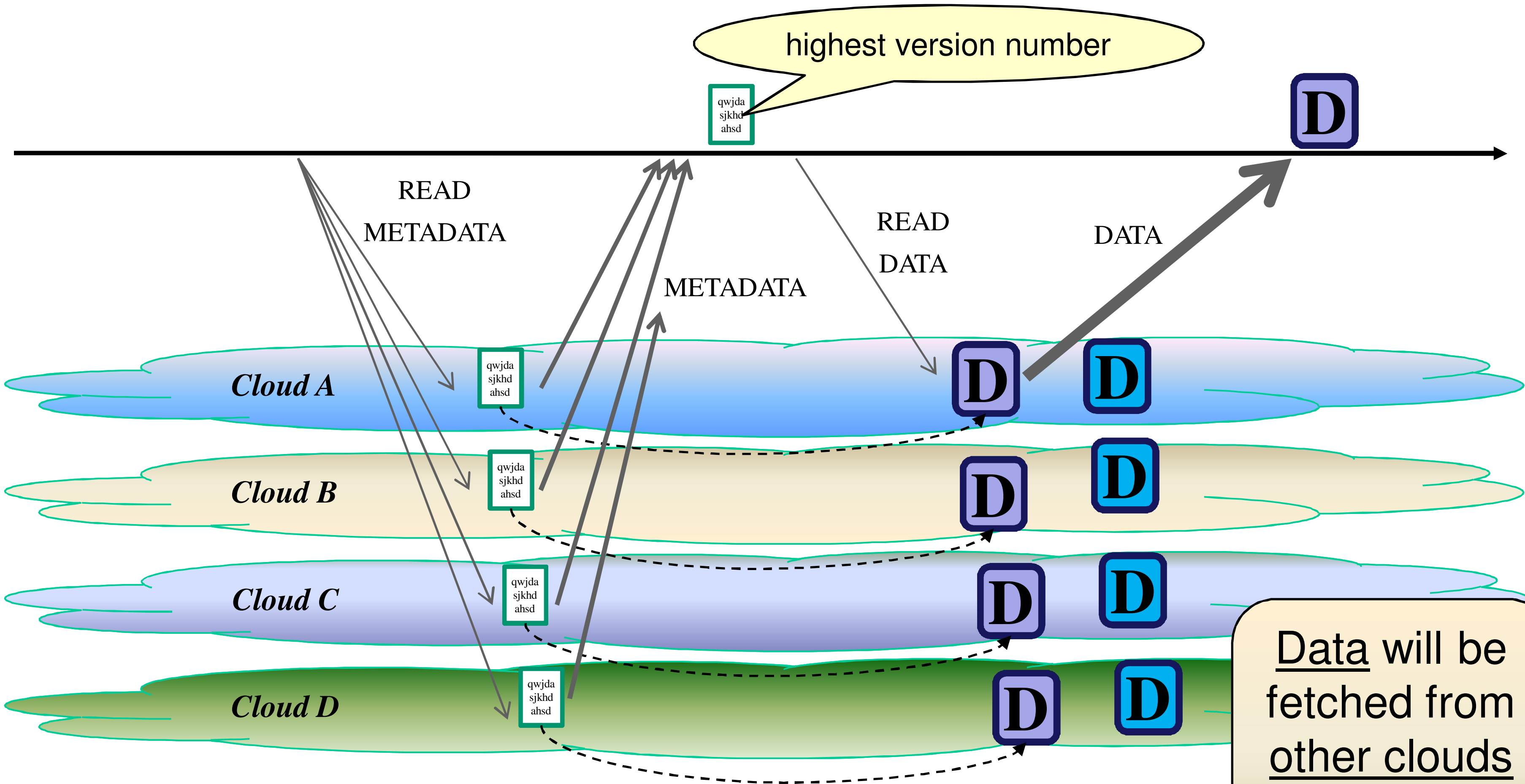
- *f*-dissemination Byzantine quorum systems [Malkhi & Reiter 1998]
  - quorums of *2f+1* servers out-of *3f+1* servers
  - data is self-verifiable (signed)



DepSky Write

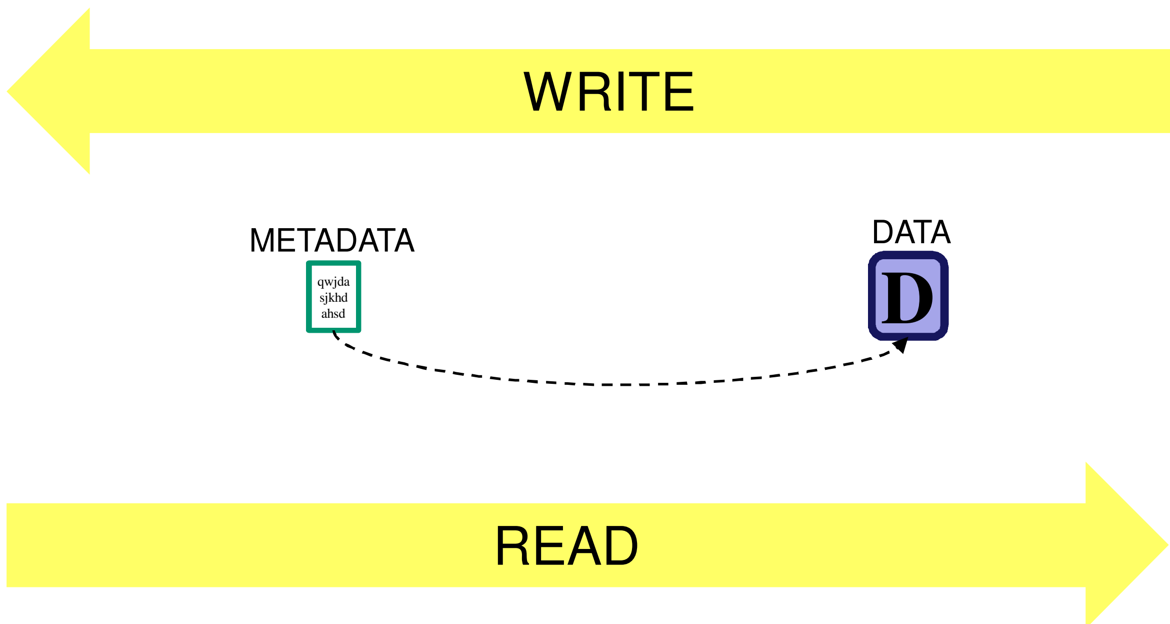


DepSky Read



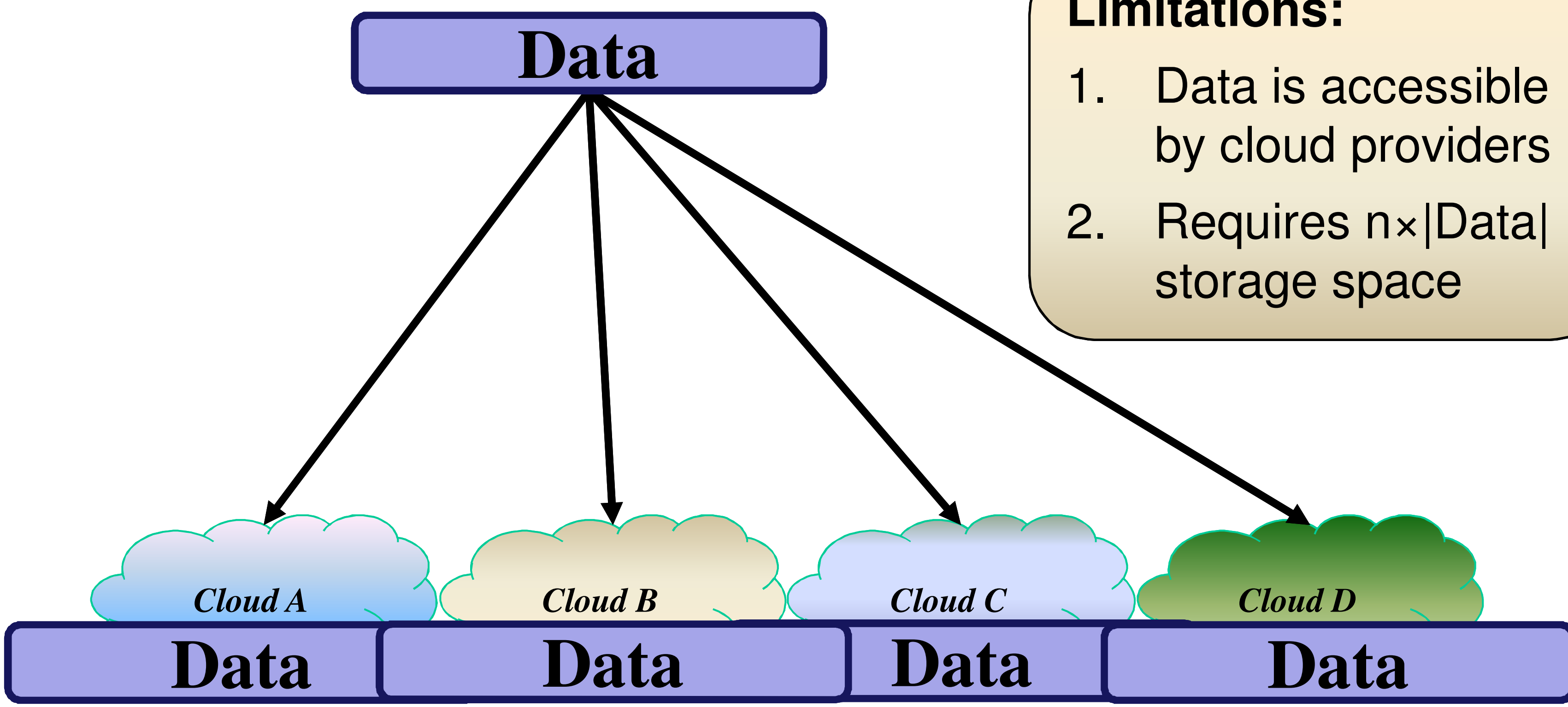
Data will be fetched from other clouds if needed.

Why does it work?



Key property: if metadata pointing to D is read, D can be read

Confidentiality

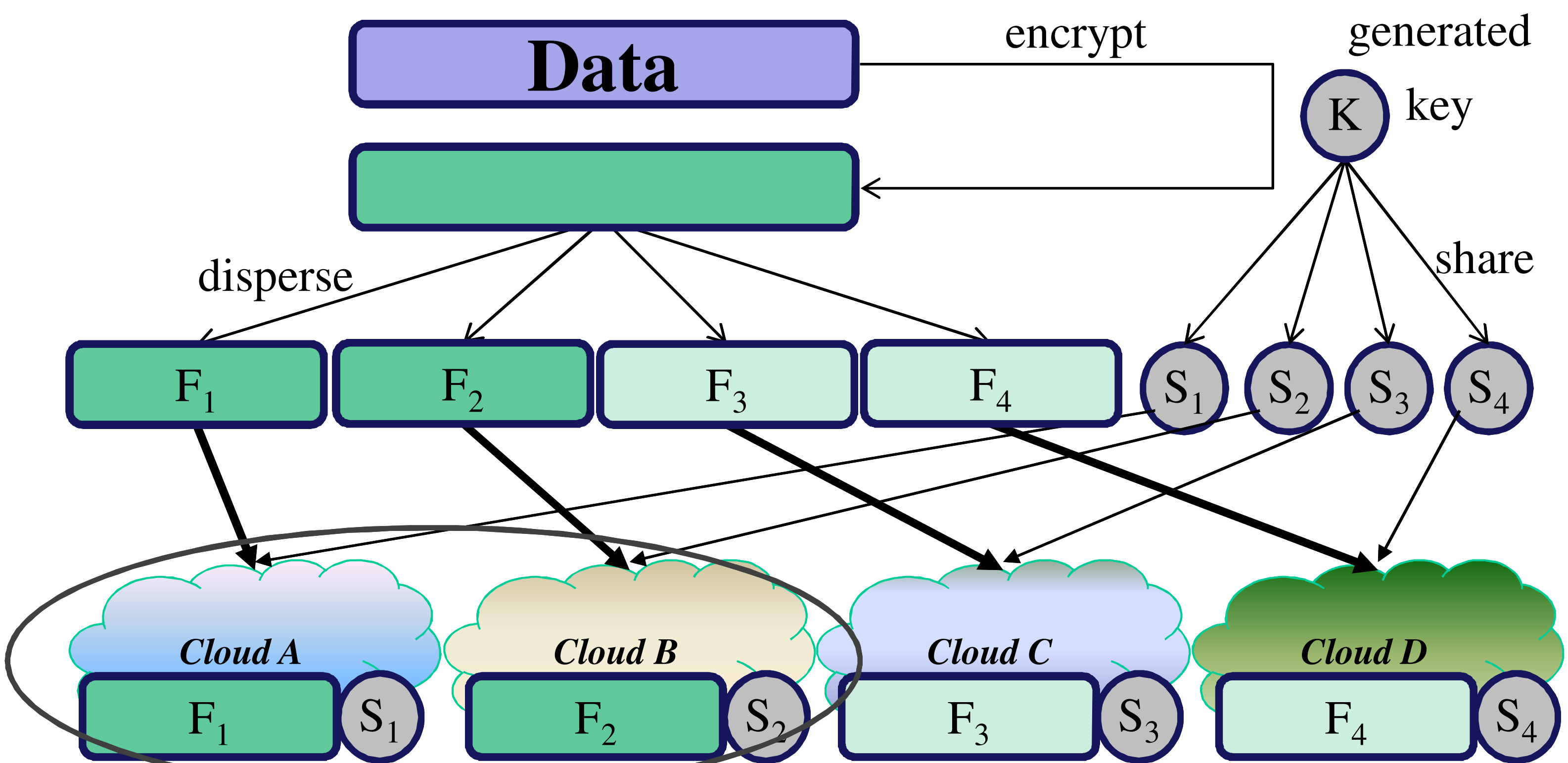


Limitations:

1. Data is accessible by cloud providers
2. Requires n×|Data| storage space

DepSky Confidentiality

Combining Erasure Codes and Secret Sharing [Krawczyk 1993]



Inverse process for reading from *f+1* shares/fragments.

Secret sharing not needed if key distribution is available.

Consistency Proportionality

- The consistency provided by DepSky is the same as the base storage clouds
  - If the weakest consistency cloud provides eventual consistency, DepSky provides eventual consistency
  - If the weakest consistency cloud provides "read your writes", DepSky provides "read your writes"
  - If the weakest consistency cloud provides regular storage, DepSky provides regular storage
- This notion may be useful for other systems

DepSky Evaluation

DepSky Performance

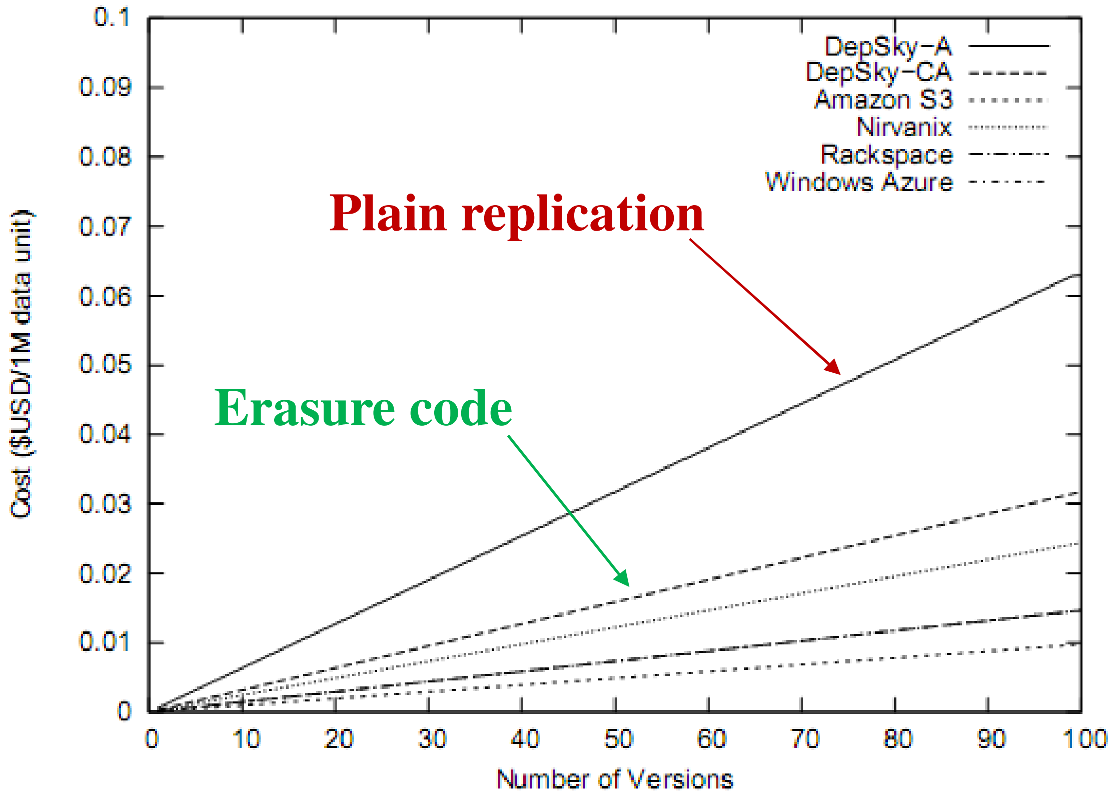
- Prototype: 3K locs (Java), REST/HTTPS
- Experimental Setup
  - Two DepSky setups: **A** (DepSky) and **CA** (DepSky with confidentiality)
  - Four commercial storage Clouds: **S3** (Amazon S3), **WA** (Windows Azure), **NX** (Nirvanix SDN) and **RS** (Rackspace)
  - Clients spread through 8 PlanetLab sites around the world
  - Three clients on each site, reading/writing data units of three sizes (100kb, 1Mb and 10Mb)
  - 437000+ reads/writes between Sep. 10th and Oct. 7th 2010
- Experiments cost: ~400€

DepSky Operations Costs ($)

Operation	DepSky-CA	Amazon S3	Rackspace	Win. Azure	Nirvanix
10K Reads	1.47	1.46	2.15	1.46	1.46
10K Writes	3.08	1.46	0.78	0.98	2.93

- Monetary costs (in USD) for 1Mb data unity and **four** clouds
  - Read cost is the same of reading from the less expensive cloud
  - Write cost is the sum of writing 50% of the DU size on each cloud
- <u>These costs don't include data storage!</u>

DepSky Storage Costs ($)

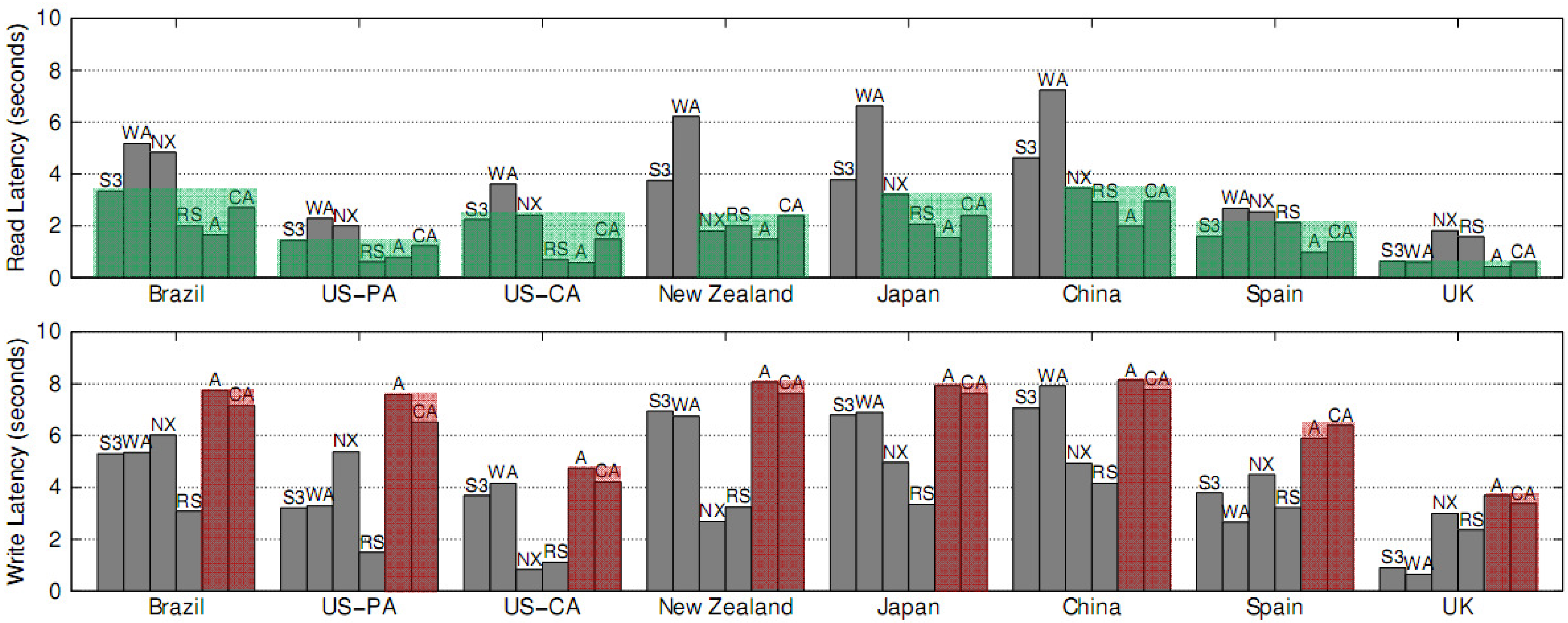


- *DepSky-CA* storage cost (1M data unit):

  2×(Avg. individual cloud cost per GB/month)

DepSky Latency (100kb DU)

DepSky **read** latency is close to the cloud with the **best** latency



DepSky **write** latency is close to the cloud with the **worst** latency

DepSky Performance

- Secret sharing latency overhead < 0.1%
- Effectiveness of read optimization
  - Fetch data first from the clouds that returned metadata faster
  - Effective in 83% (A) and 68% (CA) of reads
- Throughput **per client**:
  - 65-1480 kb/s (read) and 3-108 kb/s (write)
    - Orders of magnitude smaller than LAN BFT storage systems [Hendricks et al 2007]
  - Cloud aggregate throughput may be “infinite”

DepSky Perceived Availability

Location	Reads Tried	DepSky-A	DepSky-CA	Amazon S3	Rackspace	Azure	Nirvanix
Brazil	8428	1.0000	0.9998	1.0000	0.9997	0.9793	0.9986
US-PA	5113	1.0000	1.0000	0.9998	1.0000	1.0000	0.9880
US-CA	8084	1.0000	1.0000	0.9998	1.0000	1.0000	0.9996
New Zealand	8545	1.0000	1.0000	0.9998	1.0000	0.9542	0.9996
Japan	8392	1.0000	1.0000	0.9997	0.9998	0.9996	0.9997
China	8594	1.0000	1.0000	0.9997	1.0000	0.9994	1.0000
Spain	6550	1.0000	1.0000	1.0000	1.0000	0.9796	0.9995
UK	7069	1.0000	1.0000	0.9998	1.0000	1.0000	1.0000

- Apparently, some clouds don't provide the promised 5 or 6 9's of availability
- Internet availability plays an important role

Conclusions

- DepSky: Cloud-of-clouds storage system for untrusted clouds
  - **Techniques:** Byzantine quorum systems (integrity and availability), erasure codes (storage efficiency) and secret sharing (confidentiality)
  - Can be used on storage clouds as they are
  - Can be a foundation for more advanced storage systems
  - A use case for Byzantine fault tolerance

Conclusions

- Costs × Benefits
  - Four clouds are needed to tolerate a single “faulty cloud”
  - Reads are faster than single cloud reads
  - Writes are slower than single cloud writes
  - Monetary costs roughly twice the average costs of individual clouds
  - It can be improved: data doesn’t need to be in all *3f+1* clouds!

Questions?

http://www.tclouds-project.eu

We are hiring!

Tools for Confidentiality & Storage-Efficiency

- Information-optimal erasure codes
  - *encode(D)* generates n fragments $F_1, \ldots, F_n$
  - *decode(…)*, uses $f+1$ fragments to recover D
  - Remark: $|F_i| = ((f+1)/n) \times |D|$, *i.e.*, $\mathbf{|D|/2}$ **if** $\mathbf{f=1}$
- Secret sharing
  - *share(s)* generates n secrets $S_1, \ldots, S_n$
  - *combine(…)* uses $f+1$ secrets to recover s
  - Remarks:
    - $\mathbf{|s_i| = |s|}$
    - **no information about *s* can be obtained with *f* or less shares**

Related Work

- Data storage on diverse clouds
  - HAIL [Bowers et al 2009]: no confidentiality, no update, and requires code running on the clouds
  - RACS [Abu-Libdeh et al 2010]: no confidentiality, no integrity, no updates
- Byzantine Quorum Protocols
  - Most of them require servers running some protocol code
  - Byzantine Disk Paxos [Abraham 2006] is similar, but satisfies a weak liveness condition (finite writes)
- Untrusted Clouds
  - Depot [Mahajan et al 2010], SPORC [Feldman et al 2010], Venus [Shraer et al 2010]
  - Doesn't improve availability and require code running on the clouds
- None of these did experiments using multiple clouds

DepSky Latency (100kb DU)