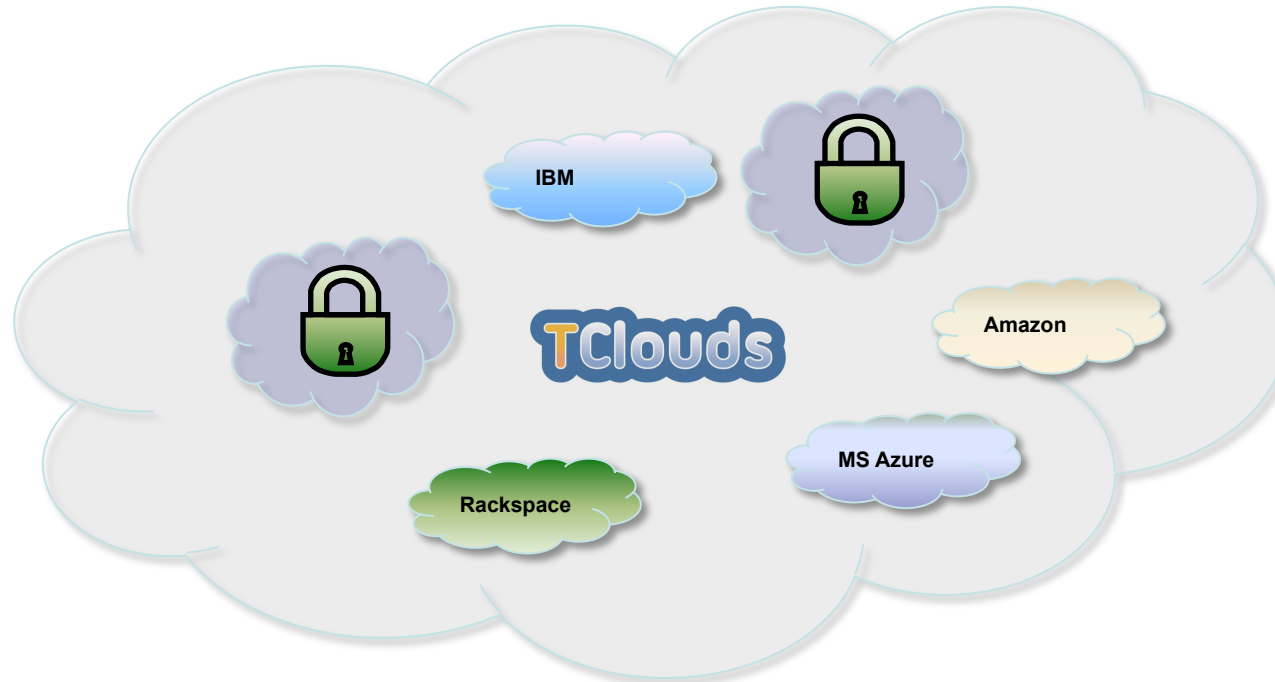# Cloud-of-Clouds Computing for Supporting Public Utilities
A Case Study

Alysson Bessani

University of Lisbon, Faculty of Sciences

Portugal

TClouds in a Nutshell

***Cloud-of-clouds**: an ecosystem composed by public and private clouds, enhanced by **TClouds** software in several ways.*

Objective: Improve security in several aspects of the cloud (storage, virtualization technology, computing, networking, secure platforms) as well as promote cloud security awareness.

TClouds Ecosystem

TClouds
TURAYA
Trusted Infr.

Cloud-of-clouds
Middleware

Trustworthy
OpenStack
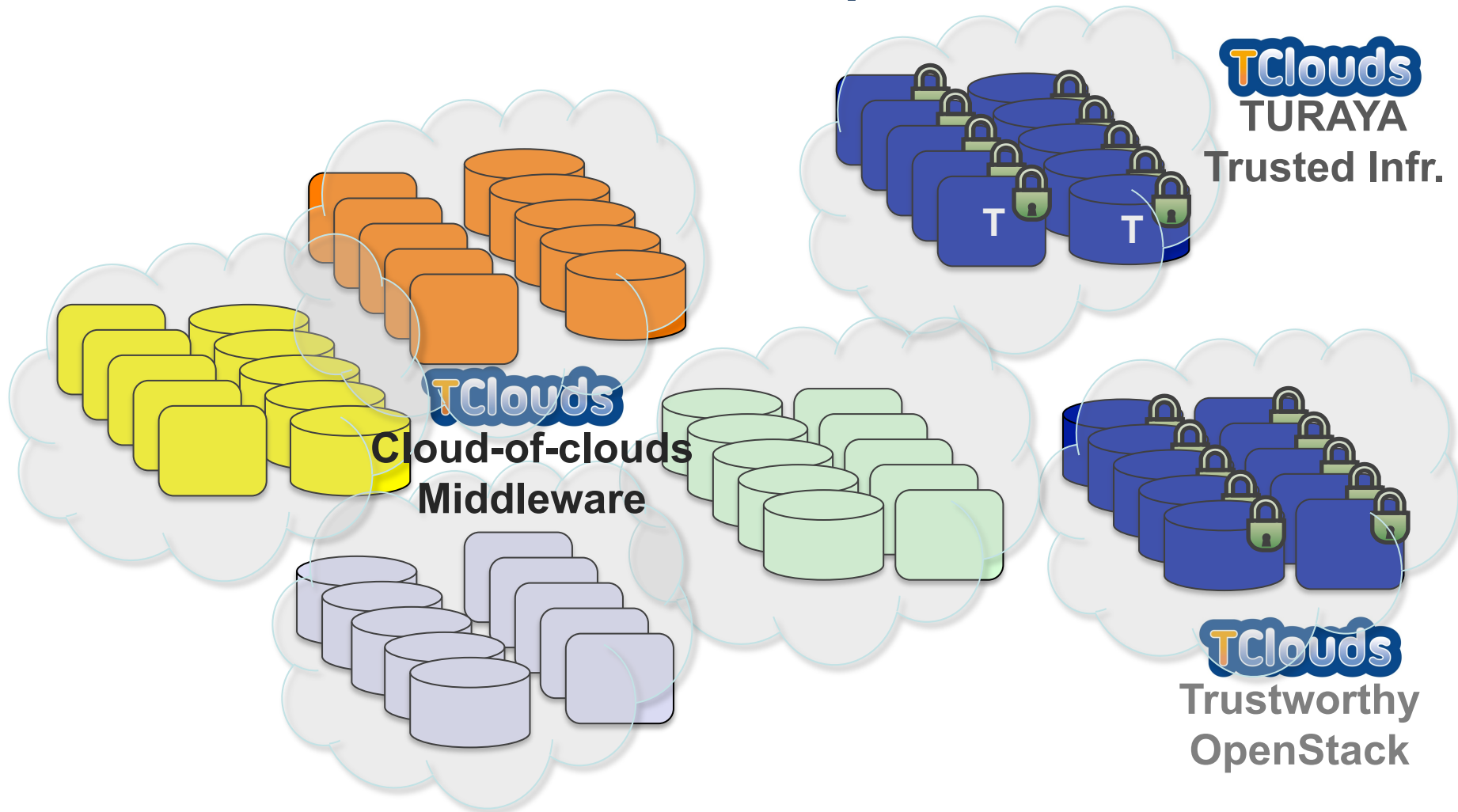
Abstractions for Cloud-of-Clouds

➢ Cloud-of-clouds services avoid <u>internet-scale single point of failures</u>, one of the main treats to cloud customers
  – Do not assume modifications or cooperation of cloud providers
➢ From cloud to cloud-of-clouds abstractions
  – Clouds abstractions
    • Storage resources (object storage)
    • Processing resources (VMs)
  – Cloud-of-clouds abstractions
    • Resilient object storage
    • State machine replication

Challenge: How to implement them efficiently?

Cloud-of-Clouds Resilient Object Storage

➢ DepSky – object read/write interface
  – Confidentiality
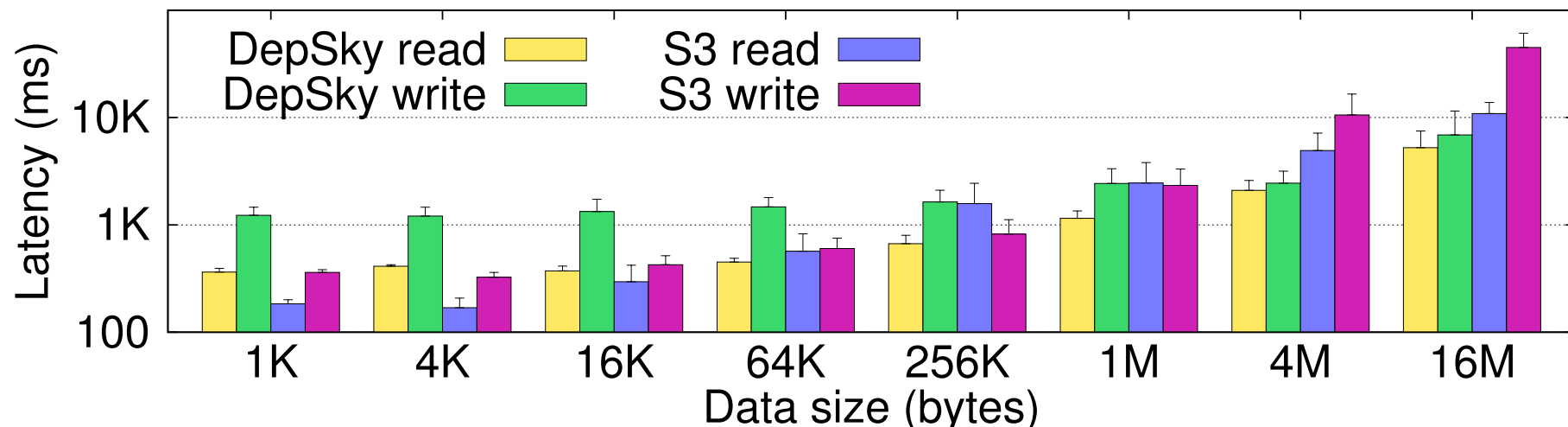 – Availability
  – Integrity
  – Cost-efficiency

Amazon
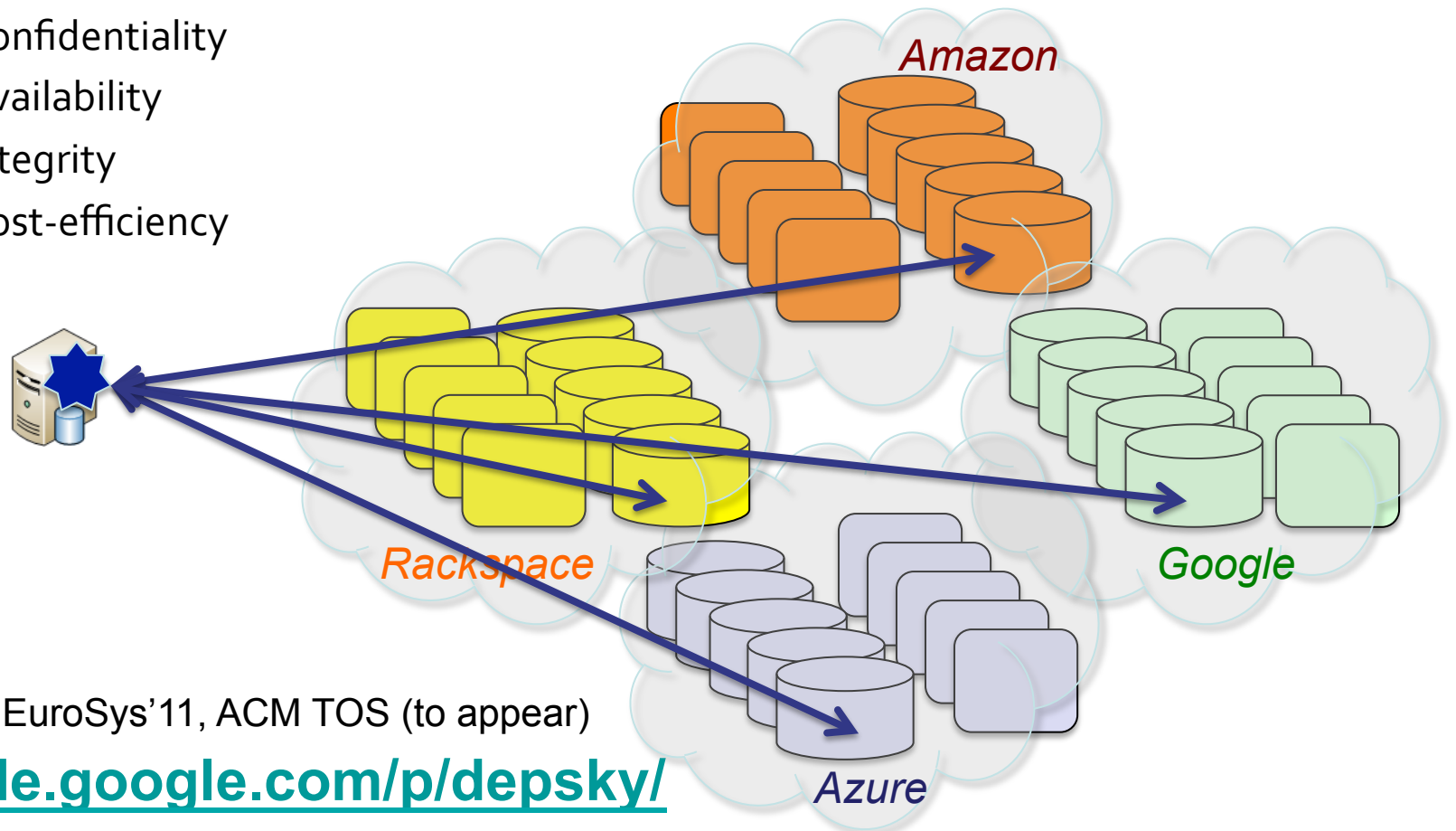
Rackspace

Google

Bessani et al. EuroSys'11, ACM TOS (to appear)

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

Azure

CoC Storage Performance

- ➤ DepSky
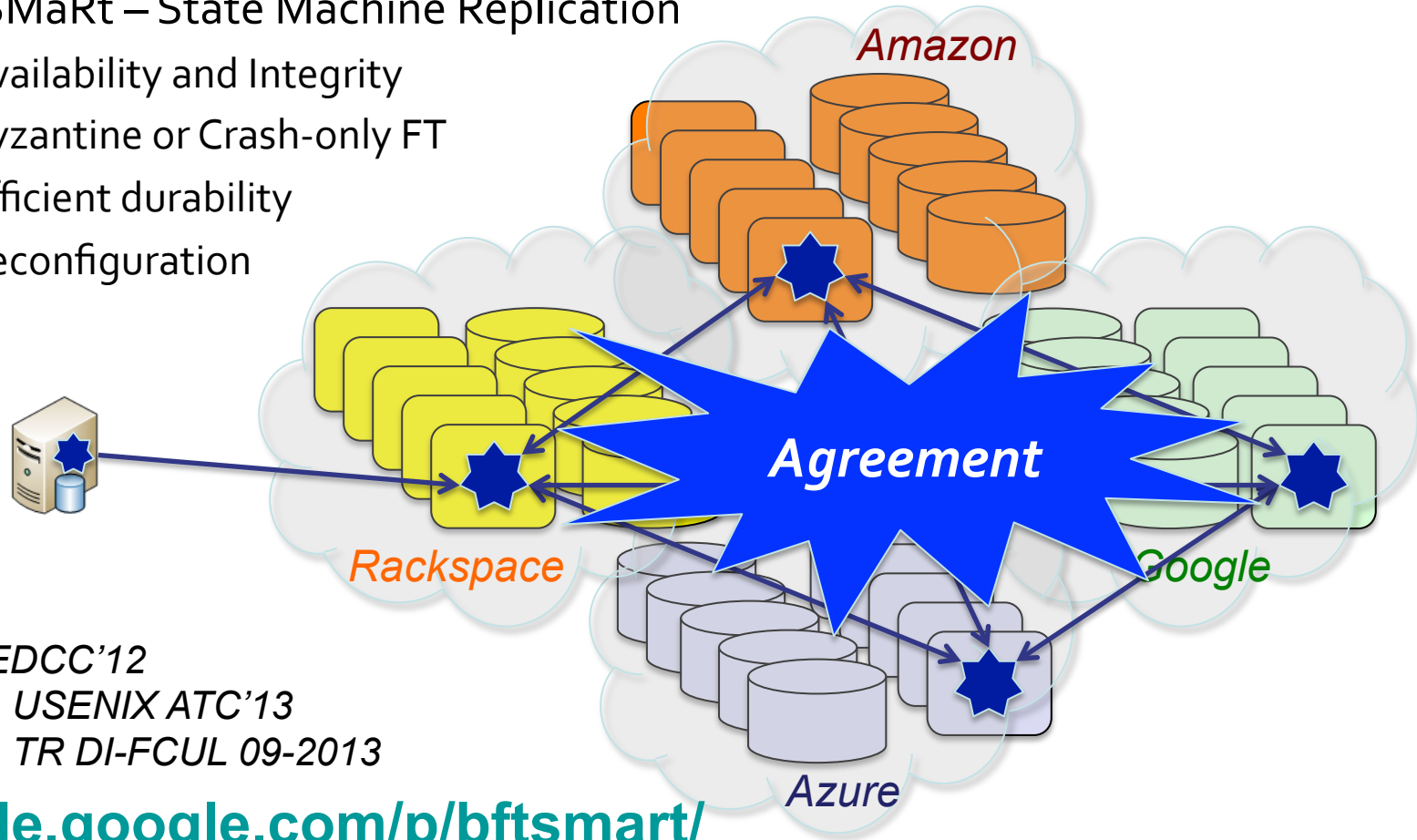 - Client in Lisbon
  - Storage services: S3, Azure B, Google Storage, Rackspace Files

Cloud-of-Clouds State Machine Replication

➢ BFT-SMaRt – State Machine Replication

   – Availability and Integrity

   – Byzantine or Crash-only FT

   – Efficient durability

   – Reconfiguration

Sousa et al. EDCC'12
Bessani et al. USENIX ATC'13
Bessani et al. TR DI-FCUL 09-2013

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

CoC SMR Performance

- ➤ BFT-SMaRt
  - Client in Lisbon
  - Replicas: EC2, Rackspace, Azure, Elastichosts (all in Europe)

	MS Operation	Local	EC2	CoC
In-memory reads	getMetadata	2.21 ± 0.85	79.85 ± 0.85	72.81 ± 1.48
	getDir	3.26 ± 0.45	84.56 ± 1.51	94.03 ± 0.81
In-memory writes	update	5.13 ± 1.42	86.32 ± 0.83	96.44 ± 1.11
	put	5.82 ± 0.88	87.15 ± 2.21	96.96 ± 1.36
	delete	3.31 ± 0.67	84.52 ± 0.91	93.95 ± 0.68

(everything in milliseconds)

Cloud-of-Clouds: Does it make Sense?

➢ From the **trust** perspective
  – Incident reports appear all the time
  – Fault independence appears to be very high

➢ From the **economical** perspective
  – Providers keep improving their "resource per money" offers
  – Secure offers are/will be very expensive and complex

➢ From the **performance** perspective
  – Better networks and ISP offers
  – New replication protocols, weak consistency
  – For hybrid clouds it can be great (<u>customer security</u>)

➢ From the **complexity** perspective
  – ???

Smart Lighting System

A PUBLIC UTILITY USE CASE

Utility Infrastructure Schedulers (UIS)

➢ User schedule actions to be executed by devices managed by a utility infrastructure (usually a SCADA system)

➢ In a smart grid, every citizen can be a user and every home can have one or more devices

Users

Utility Infrastructure

Event Scheduler

Device	Starting Date	End Date	Periodicity	Action
A	13/6/2013	13/8/2013	week	X
B	21/1/2014	21/5/2015	daily	Y
C	…	…	…	…

…
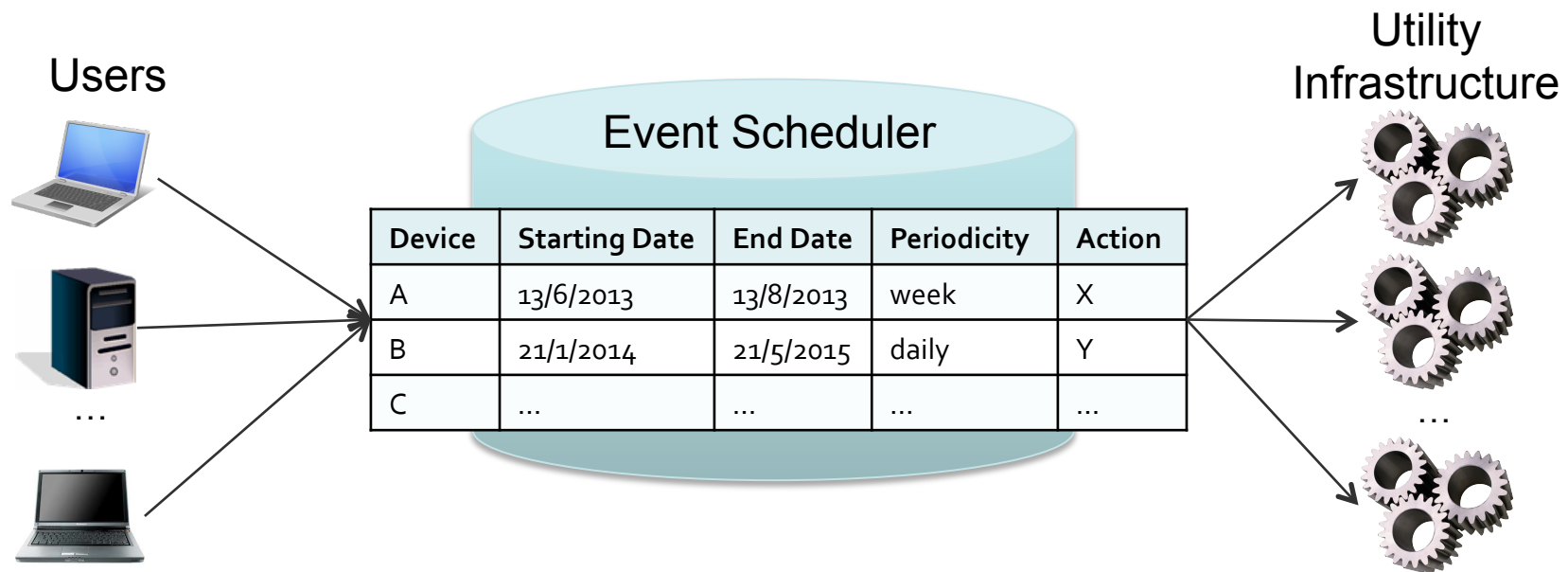
Non-functional Requirements

➢ Integrity

➢ Availability

➢ Openness   (vs. Integrity)

➢ Confidentiality   (vs. Integrity and Availability)

Conflicting requirements!!!

Are UIS Cloudy?

➢ In principle, no...

   ... but as an applied research project, we can try it

➢ However, we can use public clouds for

   – Added dependability (e.g., disaster tolerance)

   – Added scalability & elasticity

Scenario	#Users	#Devices	Transactions/day*
2012 (PT)	10s	300	1.2K
2015 (PT)	100s	12000	48K
UIS as a Service	1000s	120000	480K
Smart Grids	millions	millions	millions to billions

Only utility infrastructure-initiated

Smart Lighting System

➢ A TClouds application scenario implementing the UIS concept for managing municipalities public lighting

➢ Main partners involved: **EFACEC**, **EDP**, FCUL, SIRRIX, TUBS

# Smart Lighting System

Smart Lighting System

(A Possible Real) Architecture
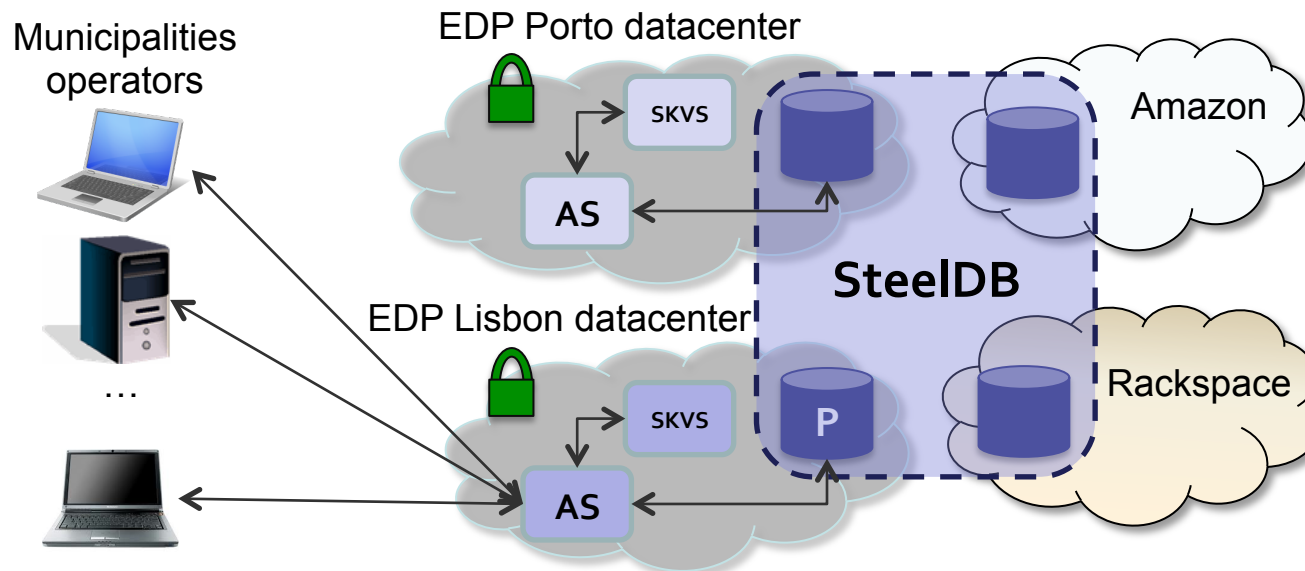
- ➢ Trusted infrastructure (by SIRRIX)
- ➢ SQL Database over BFT-SMaRt (by FCUL)
- ➢ Simple Key Value Store (by TUBS)

SteelDB: SQL Database over BFT-SMaRt

➢ Middleware-based Database replication engine

– Employs the Byzantium algorithm (Garcia et al, EuroSys'11), with some pragmatic modifications

– Satisfy Snapshot Isolation

– Tolerate crashes on clients and Byzantine faults on replicas

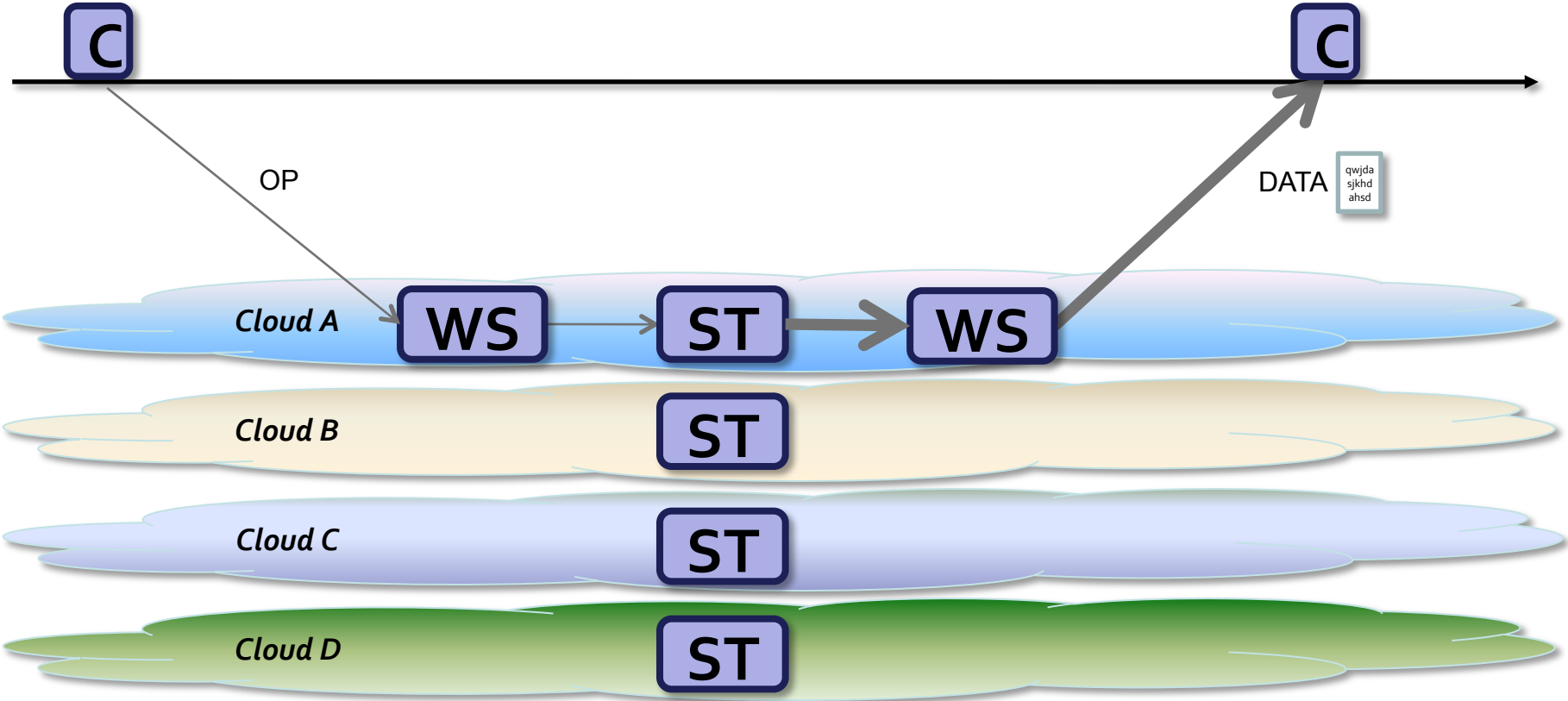– Software building blocks: BFT-SMaRt, H2 database
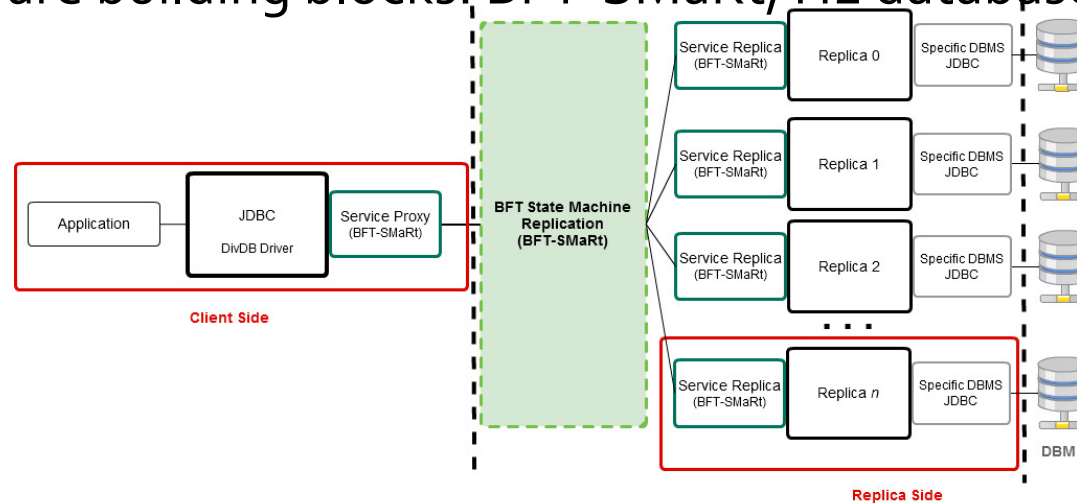
SteelDB Operation

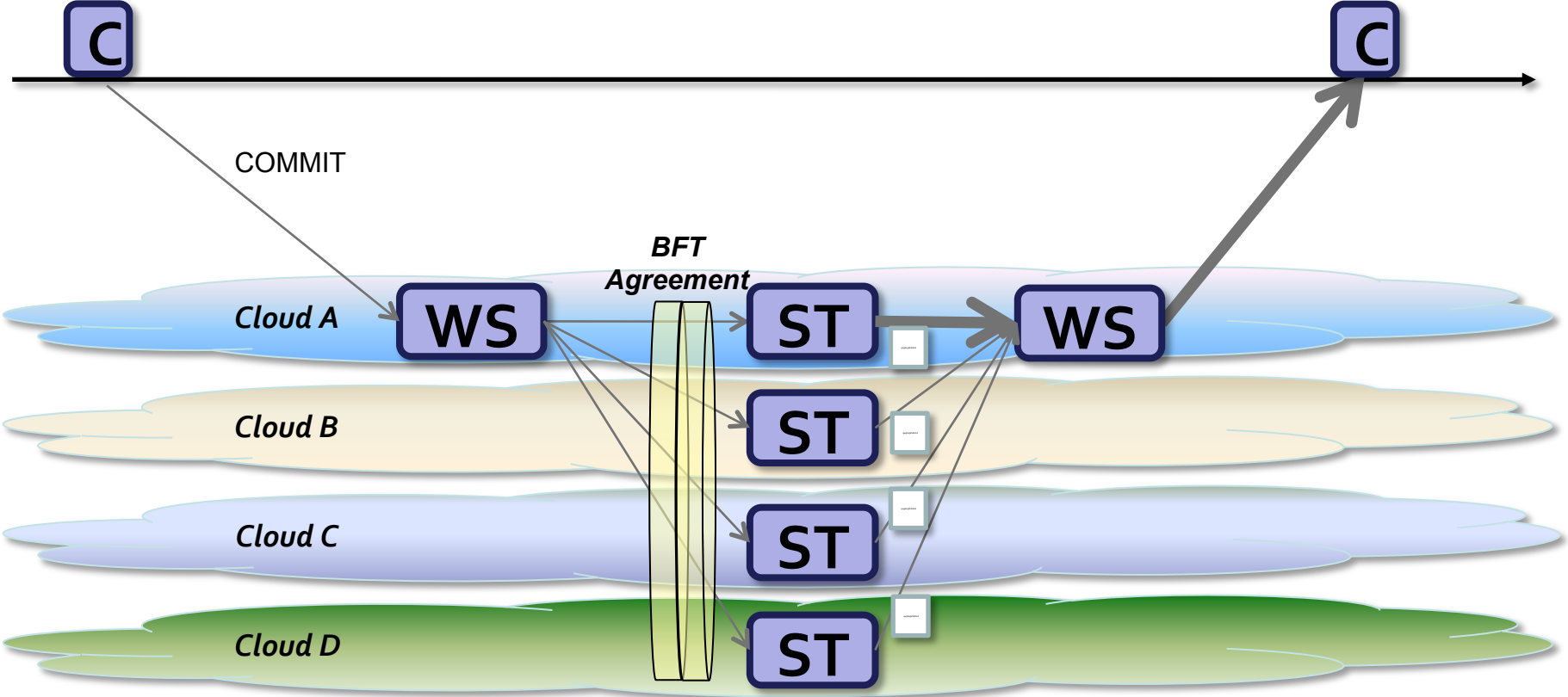## Read-only operations (reads)
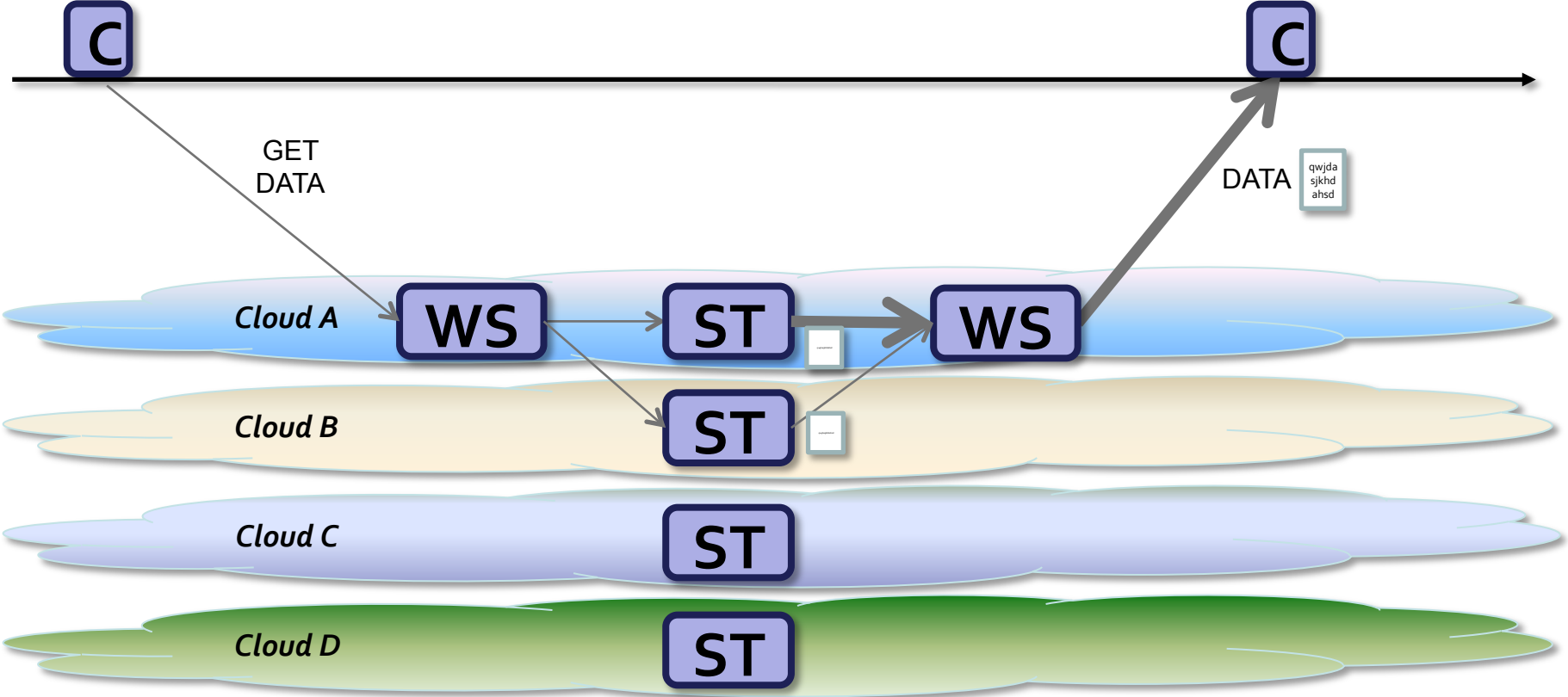
SteelDB Operation

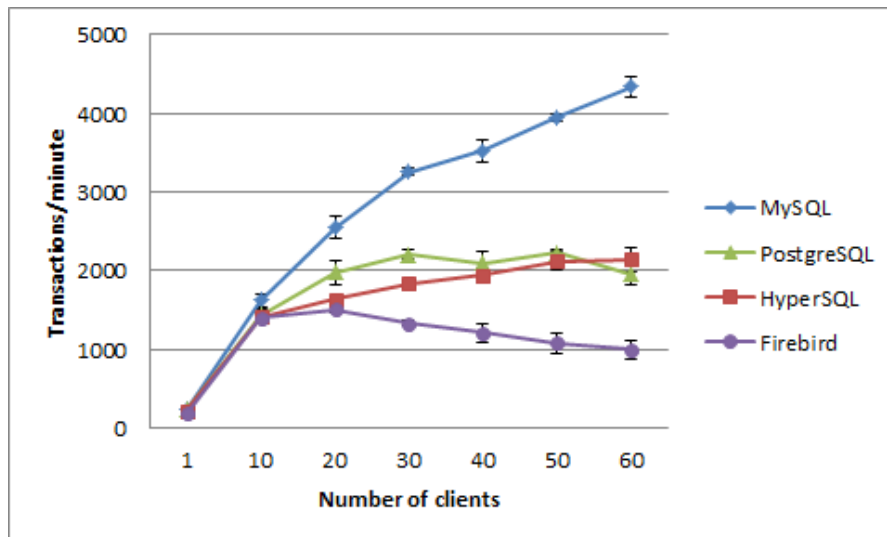Non-certified operations (non-committed reads and writes)

SteelDB Operation

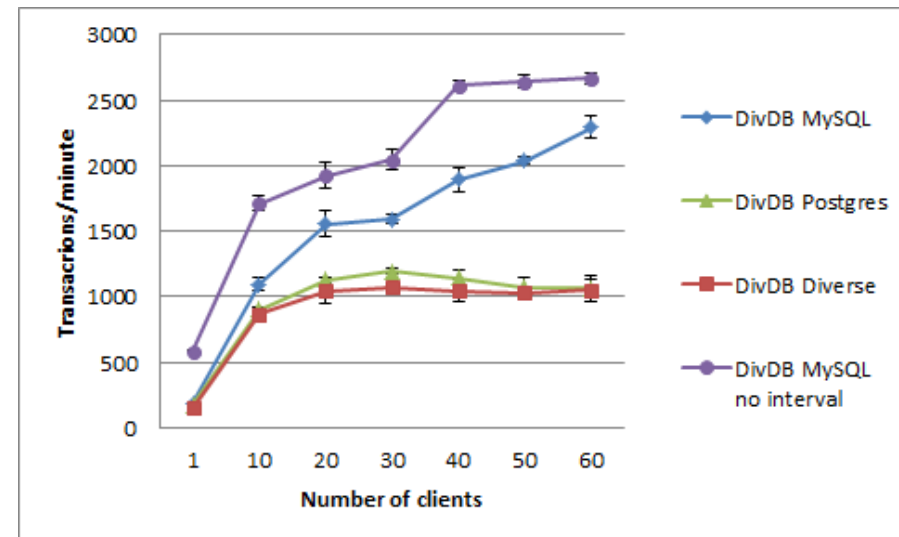Certified operations (writes and transaction commits)

What about Performance?

➢ Performance is not our first concern

➢ We don't have measurements for this scenario

➢ But we do have some encouraging numbers...

(a) Standalone DBMS.

(b) SteelDB variants.

Lessons Learned

➢ Cloud-of-clouds replication is doable

    – DepSky is a proof-of-concept, but more work is required

➢ BFT SMR is now a software artifact

    – Thanks to TClouds (and EC)

➢ Utility computing can (and want) to use the cloud

    – They don't want to be "datacenter operators"

    – SCADA vendors do use state-of-the-art software

➢ Middleware-based Database replication is too hard to impl.

    – That's why it is seldom used in practice

    – It should be better to open the database and modify it

TClouds EC CONTRACT No: 257243

"The TClouds project has received funding from the European Union's Seventh Framework Programme ([FP7/2007-2013]) under grant agreement number ICT-257243."

If you need further information, please contact the coordinator:

Technikon Forschungs- und Planungsgesellschaft mbH

Burgplatz 3a, 9500 Villach, AUSTRIA

Tel: +43 4242 233 55   Fax: +43 4242 233 55 77
E-Mail: coordination@tclouds-project.eu