Recent Advances in Cloud Computing Dependability

Paulo Veríssimo

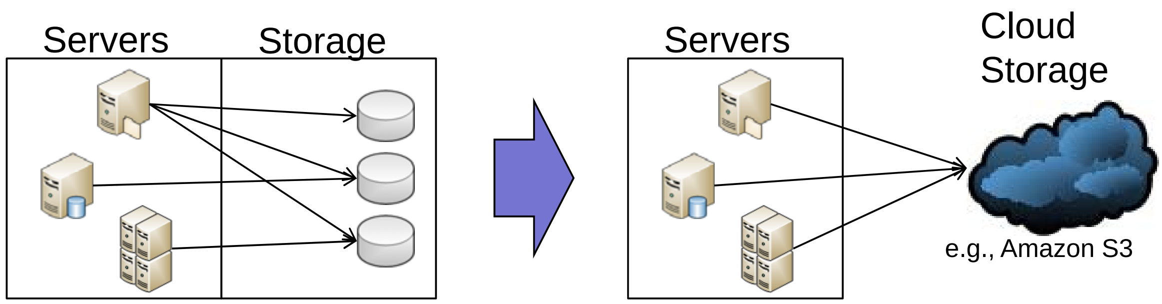
pjv@di.fc.ul.pt   http://www.di.fc.ul.pt/~pjv

joint work with: Alysson Bessani, Miguel Correia, Pedro Costa, Bernhard Kauer, Marcelo Pasin, Paulo Sousa

Univ. de Lisboa, Faculdade de Ciências (FCUL), LaSIGE, Portugal,

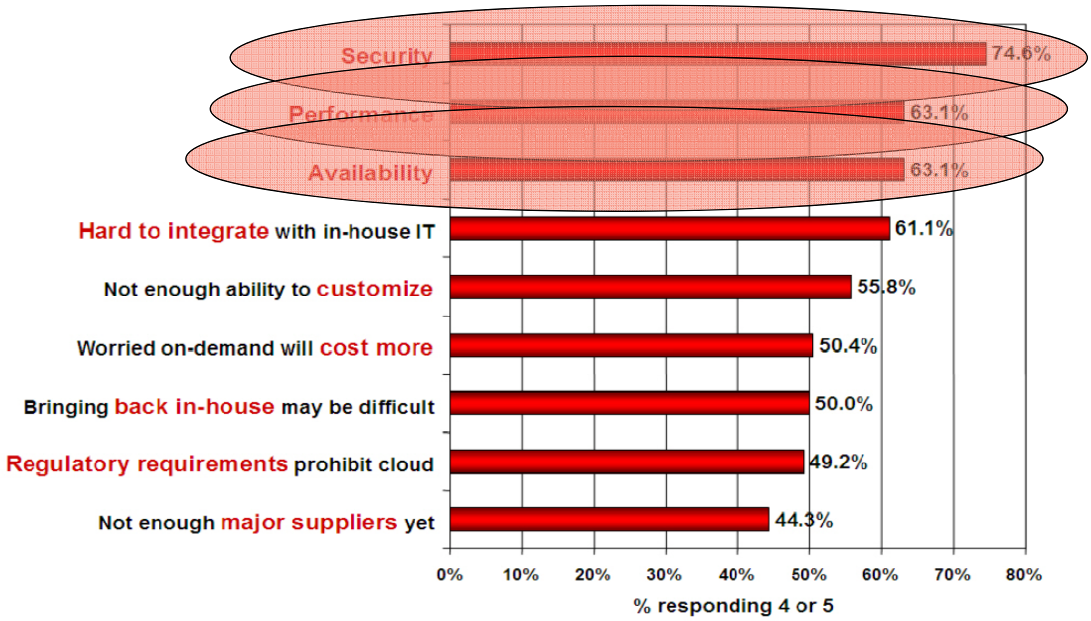
Moving to Clouds

- Data are moving to the cloud
- Main reason: costs (pay-per-use model)
- Still hesitation for critical applications (e.g., smart energy grids, health), but it's a matter of time…
- What is the risk of moving data to the cloud?



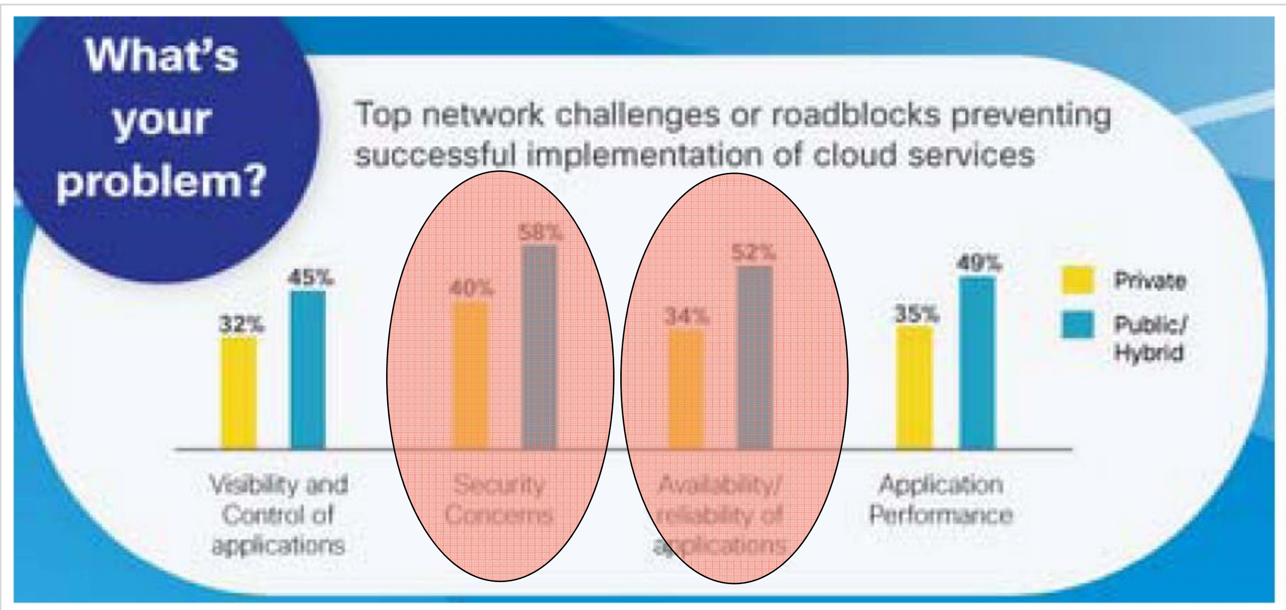
Cloudy weather ...

(many worries of cloud users)



Source: IDC Enterprise Panel, August 2008 n=244

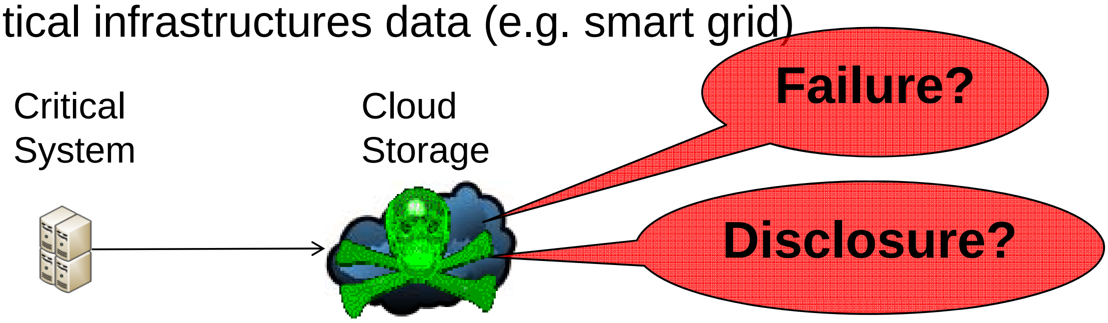
Cloudy weather ...



Source CISCO, 2012

Critical applications on the cloud?

- is depending on one cloud *(or provider thereof)* enough to build trust ?
- E.g., privacy- and security-critical data storage
  - Medical records
  - Company financial data
  - Critical infrastructures data (e.g. smart grid)



TClouds big challenge

- How to allow a swift migration path from current commodity insecure clouds to future natively resilient (secure and dependable) and cost-effective clouds?

Alternatives for cloud resilience

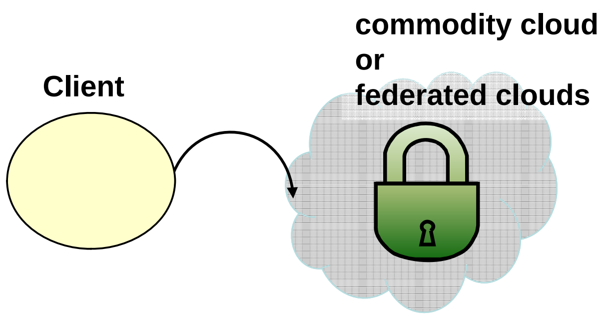
- (i) Approaches confined to single cloud provision.
- (ii) Proprietary trusted or accredited clouds may implement specific IaaS or PaaS approaches to achieving resilience.
- (iii) Federated cloud environments, which require alliance of the involved providers.
- (iv) Cloud-of-clouds environments, which take advantage of multiple independent cloud provider offers.

Trusted-Trustworthy Clouds

Options (i), (ii), (iii):

1) *Rely on improved cloud infrastructure by single or federated cloud providers*

***CON**: dependence on actual provider(s) trustworthiness (single point of failure, lock-in, collusion)*

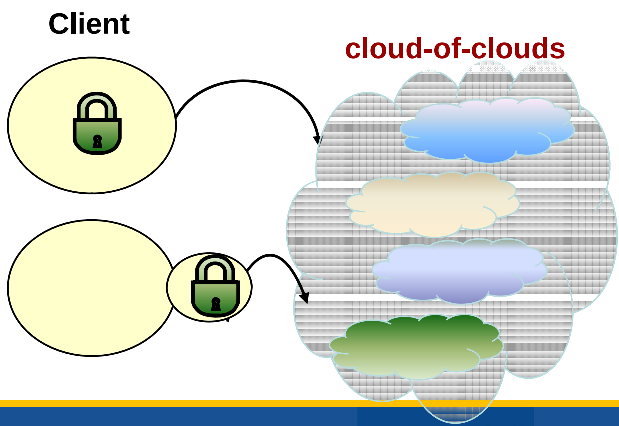


Trusted-Trustworthy Clouds

Option (iv):

2) ***cloud-of-clouds*** *– use multi-cloud environments independently*

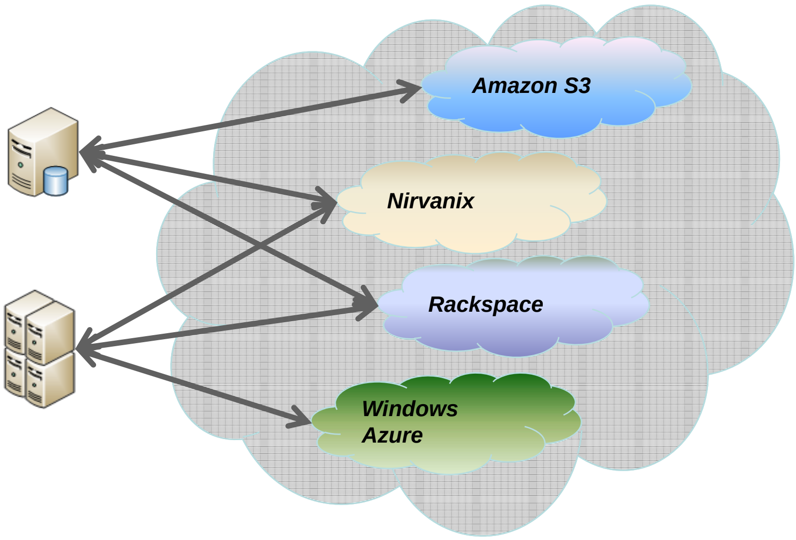
PRO*: be your own master w.r.t. trust*



Some solutions in the cloud-of-clouds world

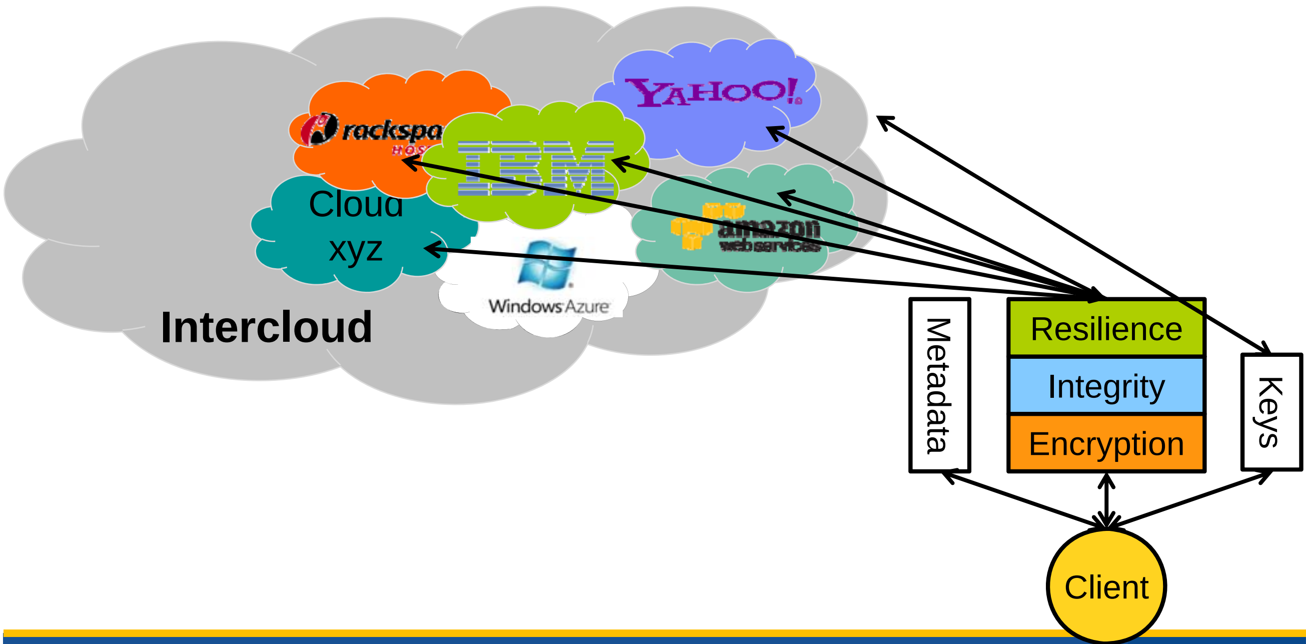
DepSky – Dependable and Secure Storage in a Cloud-of-Clouds.

A. Bessani, M. Correia, B. Quaresma, F. André, P. Sousa *[Eurosys 2011]*

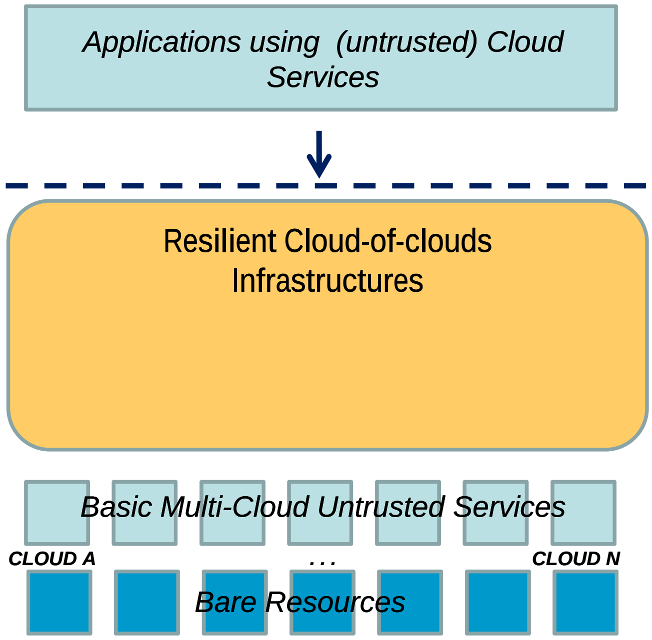


Robust data sharing with key-value stores [DSN'12]

M. Vukolić (EURECOM, France), C. Basescu (Vrije Universiteit Amsterdam), C. Cachin, R. Haas, A. Sorniotti (IBM Research Zurich) , I. Eyal, I. Zachevsky (Technion)



The TClouds Architecture: Open and Resilient Cloud-of-clouds Comput.
P. Verissimo, Alysson Bessani, Marcelo Pasin [DVDV@DSN'12]



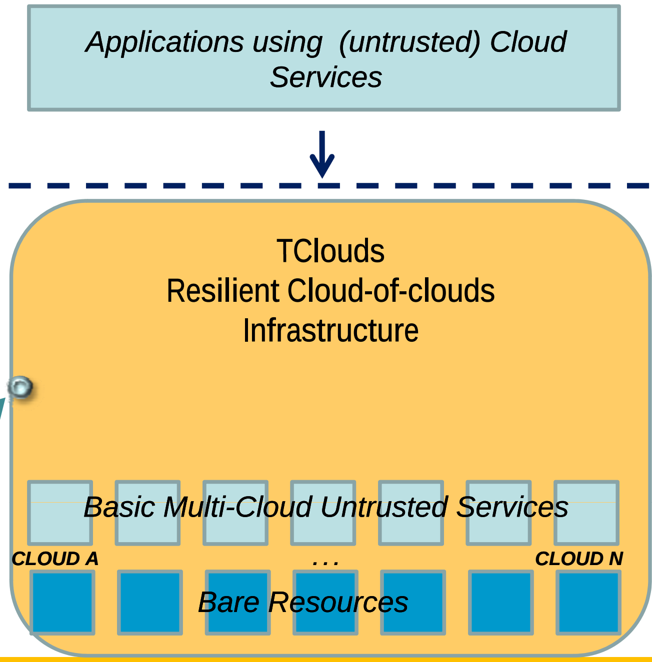
TClouds big challenge

- How to allow a swift migration path from current commodity insecure clouds to future natively resilient (secure and dependable) clouds?
- How to promote, along and at the end of this road, a diverse and open ecosystem?

- How about a coherent architecture, with modular and reusable artefacts?

Status-quo

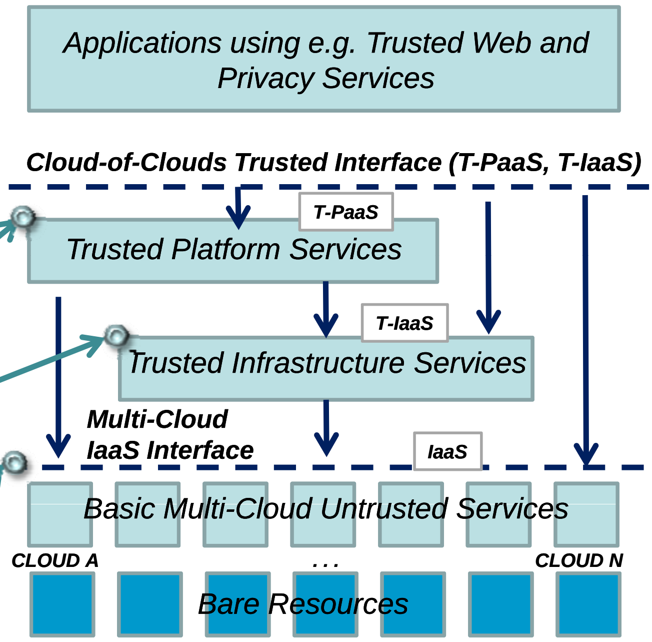
- Existing Technologies:
  - "cloudified" scenario has **availability + security needs that cannot be met by application layer alone.**
  - proprietary approaches to security **can create exclusion** and make interoperation difficult and expensive
  - **single-cloud** solutions, even if open, will not address high resilience objectives, since they are a **single point of failure**
- A solution - **resilient cloud-of-clouds infrastructure**:
  - automated computing resilience against attacks and accidents in complement or in addition to commodity clouds



Overview of the TClouds CoC architecture
(interfaces)

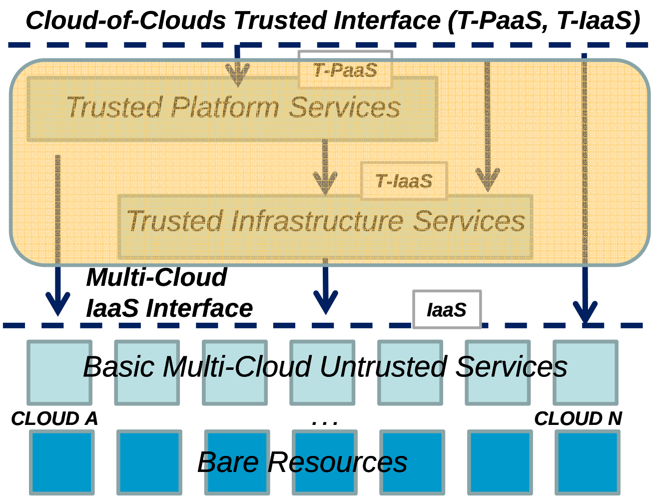
•The TClouds architecture thus provides applications with a **wealth of interfaces** to produce **incremental resilience solutions** with **single or multiple clouds** :

- TClouds Trusted Platform services (**T-PaaS)** on top of the middleware layer
- TClouds Trusted Infrastructure services (**T-IaaS)** from within the middleware layer
- Infrastructure services (**IaaS)** from available commodity untrusted clouds



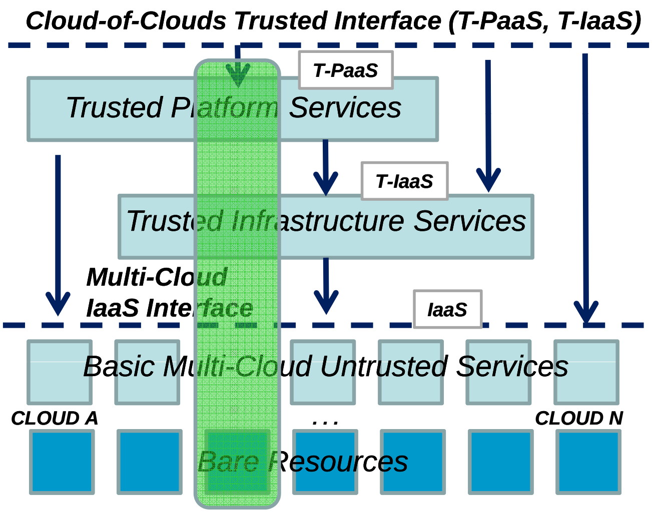
TClouds design approaches

- The TClouds architecture allows **several solutions for resilience** based on Trusted Platform or Infrastructure services (**T-PaaS, T-IaaS**) , with essentially a re-use of the same basic algorithms and mechanisms:
  - **T-PaaS, T-IaaS** implemented with a TClouds resilient middleware layer on top of commodity clouds
  - Native TClouds where resilience may also be built from scratch in the bare resources (e.g. with local low-level VM FIT mechanisms)
  - TClouds middleware is by nature cloud-of-clouds, and **T-PaaS, T-IaaS** can be implemented with any mix of native TClouds, "T-cloudified" commodity clouds with local resilience layer, and commodity clouds



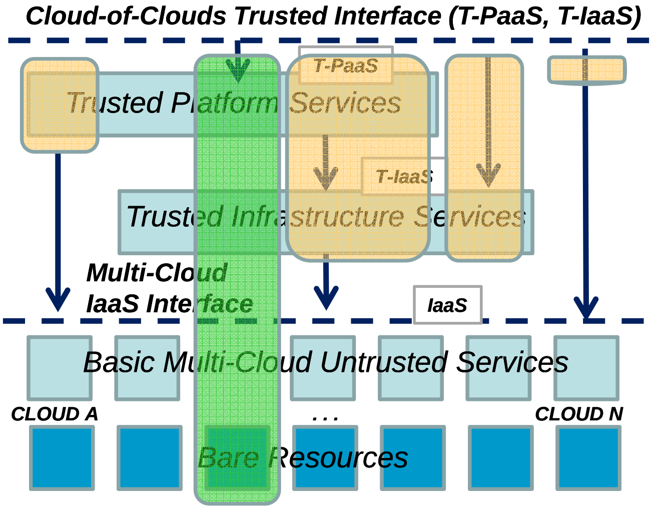
TClouds design approaches

- The TClouds architecture allows **several solutions for resilience** based on Trusted Platform or Infrastructure services (**T-PaaS, T-IaaS**) , with essentially a re-use of the same basic algorithms and mechanisms:
  - **T-PaaS, T-IaaS** implemented with a TClouds resilient middleware layer on top of commodity clouds
  - Native TClouds where resilience may also be built from scratch in the bare resources (e.g. with local low-level VM FIT mechanisms)
  - TClouds middleware is by nature cloud-of-clouds, and **T-PaaS, T-IaaS** can be implemented with any mix of native TClouds, "T-cloudified" commodity clouds with local resilience layer, and commodity clouds



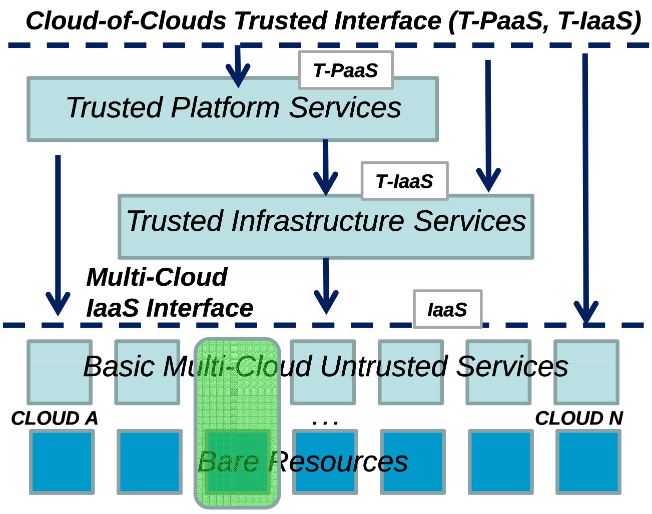
TClouds design approaches

- The TClouds architecture allows **several solutions for resilience** based on Trusted Platform or Infrastructure services (**T-PaaS, T-IaaS**) , with essentially a re-use of the same basic algorithms and mechanisms:
  - **T-PaaS, T-IaaS** implemented with a TClouds resilient middleware layer on top of commodity clouds
  - Native TClouds where resilience may also be built from scratch in the bare resources (e.g. with local low-level VM FIT mechanisms)
  - TClouds middleware is by nature cloud-of-clouds, and **T-PaaS, T-IaaS** can be implemented with any mix of native TClouds, "T-cloudified" commodity clouds with local resilience layer, and commodity clouds



TClouds design approaches

(Native TClouds with resilience built in the bare resources)



Recursive Virtual Machines for Advanced Security Mechanisms

Bernhard Kauer, Paulo Veríssimo, Alysson Bessani
University of Lisbon, Faculty of Sciences
LaSIGE

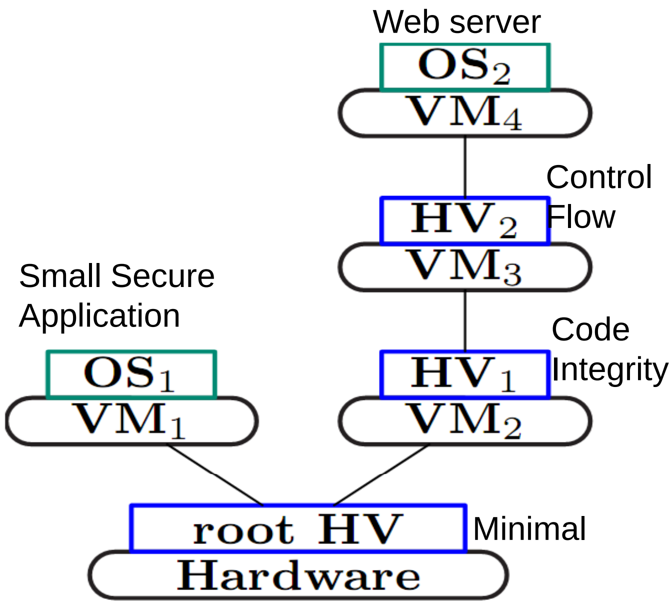
Virtual Machines & Security

- Virtualization is a key enabler of the cloud computing business model
- Leveraging virtualization for security:
  - Protect kernel code or application data
  - Intrusion detection
  - Trusted execution environments
  - Efficient SWIFIT (software implemented fault and intrusion tolerance)
  - **Providing protection and confinement for defense-in-depth architectures**

Nested Virtualization

- **Nested VM**: one virtual machine running inside of a VM (or, an hypervisor managing a VM instead of hardware directly)
- Nested VMs generalization:

  recursive virtual machines



Related Work

- **Recursive virtualization** [Popek and Goldberg 1974, Belpire and Hsu 1975]
- **Nesting two VMs**: AMD [Graf and Roedel 2009] and Intel VMX (Turtles) [Yehuda et al., 2010]
- **Exponential overhead**: more than two VMs is a killer
  - **Hardware extensions** to reduce this overhead [Poon and Moon 2010]
  - **Fluke** [Ford et al. 1996] is similar, but they provide only system call virtualization

Exponential Overhead of Nested Virtualization

- Main reason: **trap-and-emulate**
  - Parent VMs trap the virtualization instructions executed by children VMs and emulate them

Virtualization instructions usually executed by the hypervisor to handle a trap

AMD SVM	Intel VT
`clgi`	`vmread(exit-reason)`
`vmload(child-state)`	`vmread(exit-qualification)`
`vmrun(child)`	`vmread(instruction-pointer)`
`vmsave(child-state)`	`vmread(instruction-len)`
`vmload(parent-state)`	`vmwrite(instruction-pointer)`
`stgi`	`vmresume(child)`

6 instructions per event for the parent VM!
36 instructions for the grand-parent VM!

Practical Limits of Nested Virtualization (maximum allowed number of nested VMs)

Branching Factor	Interrupts per Second			
	1	10	100	1000
2	22	19	15	12
4	11	10	8	6
6	9	8	6	5
8	8	7	5	4
10	7	6	5	4

- Slow but live nested virtual machines:
  - One interrupt per second: **9 NVMs**
  - 1000 interrupts per second: **5 NVMs**

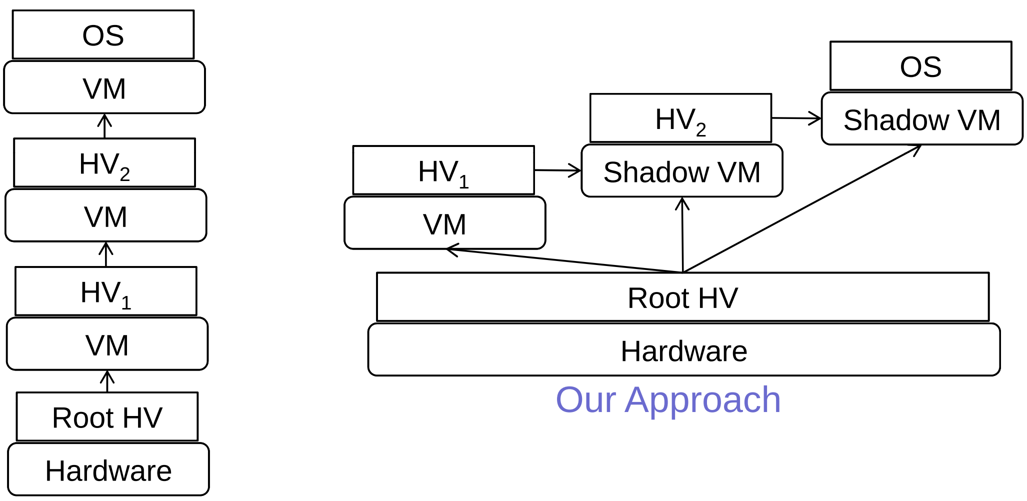
Our Contribution

- Efficient implementation of **recursive virtual machines**
- A **hypervisor architecture** that allows VMs to be stacked without the expected exponential performance overhead
- Some ideas for recursive VM use to build **advanced security mechanisms**

Core Idea

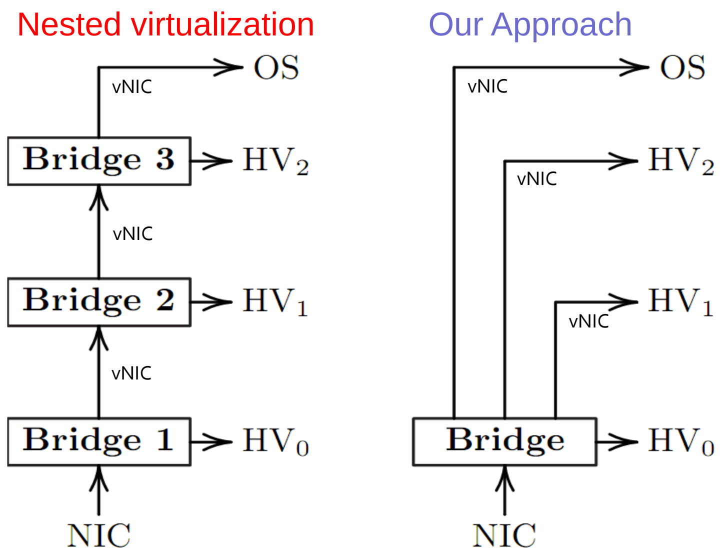
Instead of repeating the support for nested VMs in every layer, we just implement recursive VMs in the root hypervisor

Nested virtualization



I/O Virtualization

- Root hypervisor also assigns **virtual devices**
  - Every physical device is bridged only in the root VM – <u>linear overhead</u>



Network card example

Advanced Security Mechanisms

- Thin security layers

  Different hypervisors can improve different security aspects of legacy OSs

- Defense in depth

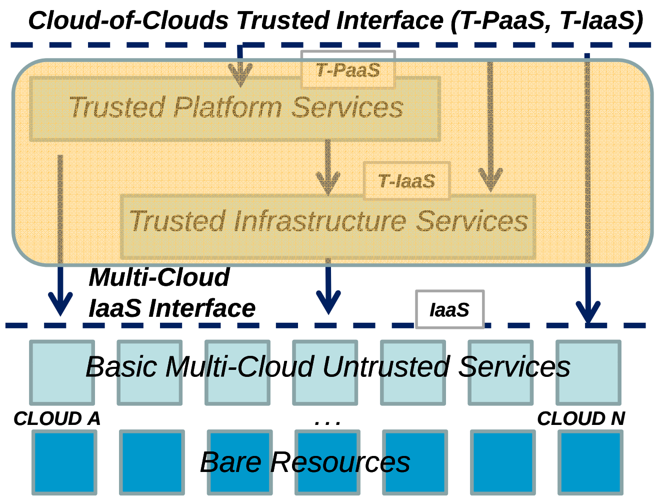
  In-depth barriers of several kinds, such as firewall-like filters, wrappers, failure and intrusion detectors, etc.

- Intrusion and fault tolerance

  Decompose trusted components (for efficient BFT) in several micro-hypervisor layers

TClouds design approaches

(resilient middleware layer on top of commodity clouds)



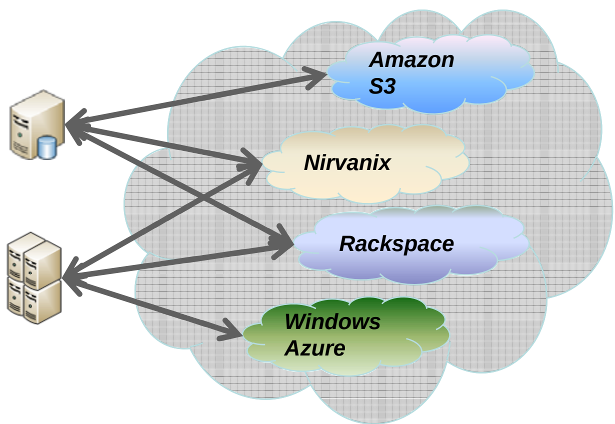
TClouds

SEVENTH FRAMEWORK PROGRAMME

A concrete proof-of-concept result with the TClouds architecture:

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

[Bessani et al., ACM Sigops Eurosys 2011]



DepSky Design Assumptions

1. No trust on individual cloud providers

Distributed trust is built by independent mechanisms over commodity multi-cloud environments

2. Use storage clouds as they are

No server-side code needed on the cloud

3. Data is updatable

Quorum replication protocols for consistency

System Model

- Asynchronous distributed system
- Faults
  - Clouds can be unavailable, corrupt
  - Readers can do whatever they wa
  - Writers can **crash and recover**

Byzantine faults

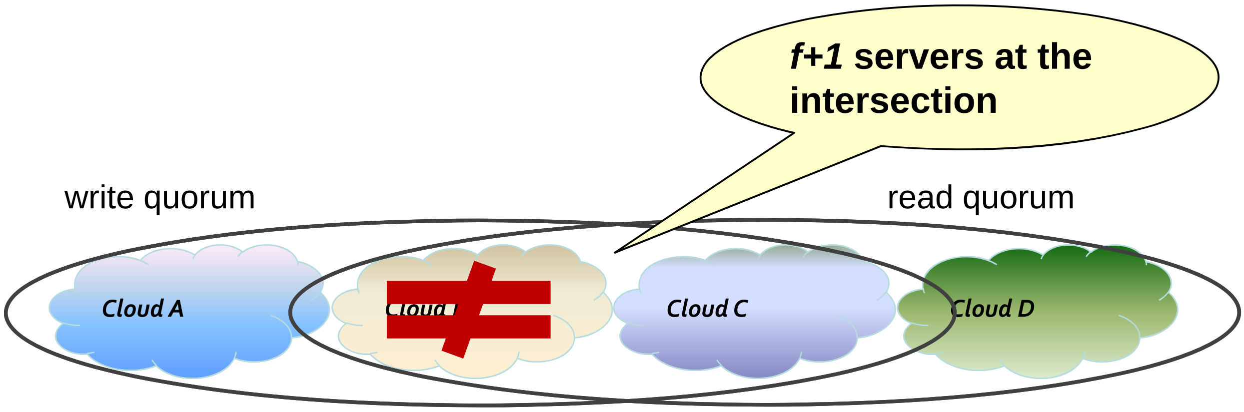
- $n = 3f + 1$ clouds to tolerate f faults
- Symmetric and asymmetric cryptography
- Data model: single-writer multiple-reader regular register

Availability and Integrity

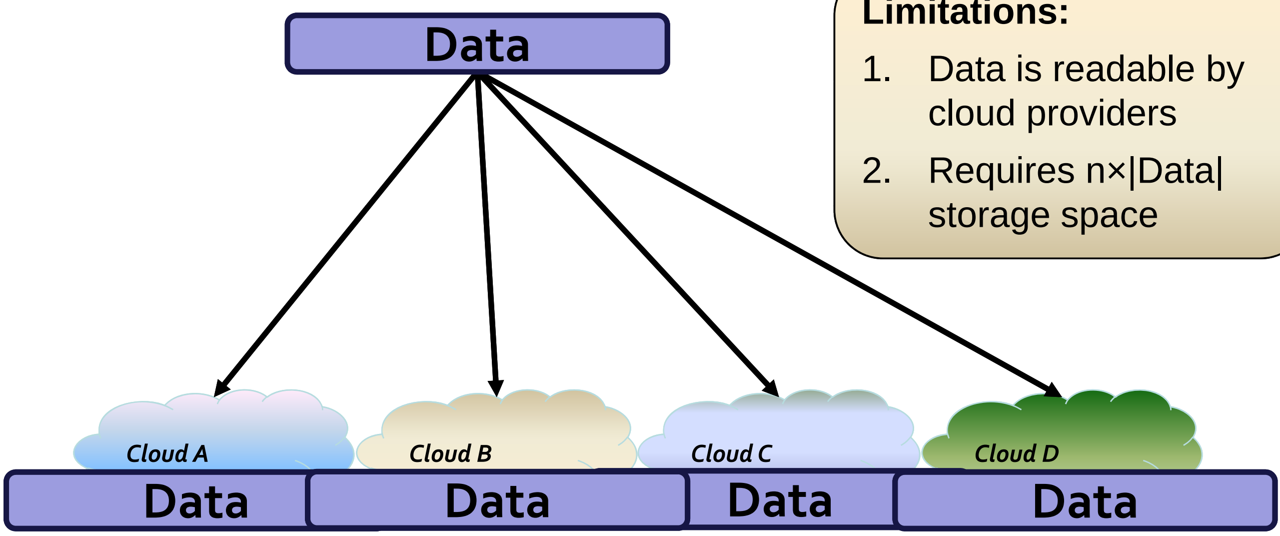
f-dissemination Byzantine quorum systems [Malkhi & Reiter 1998]

- Read/Write protocols in
  - quorums of *2f+1* servers out-of *3f+1* servers
  - data is self-verifiable (signed)

f=1



Limitations of simple replication
(in Byzantine failure scenarios)

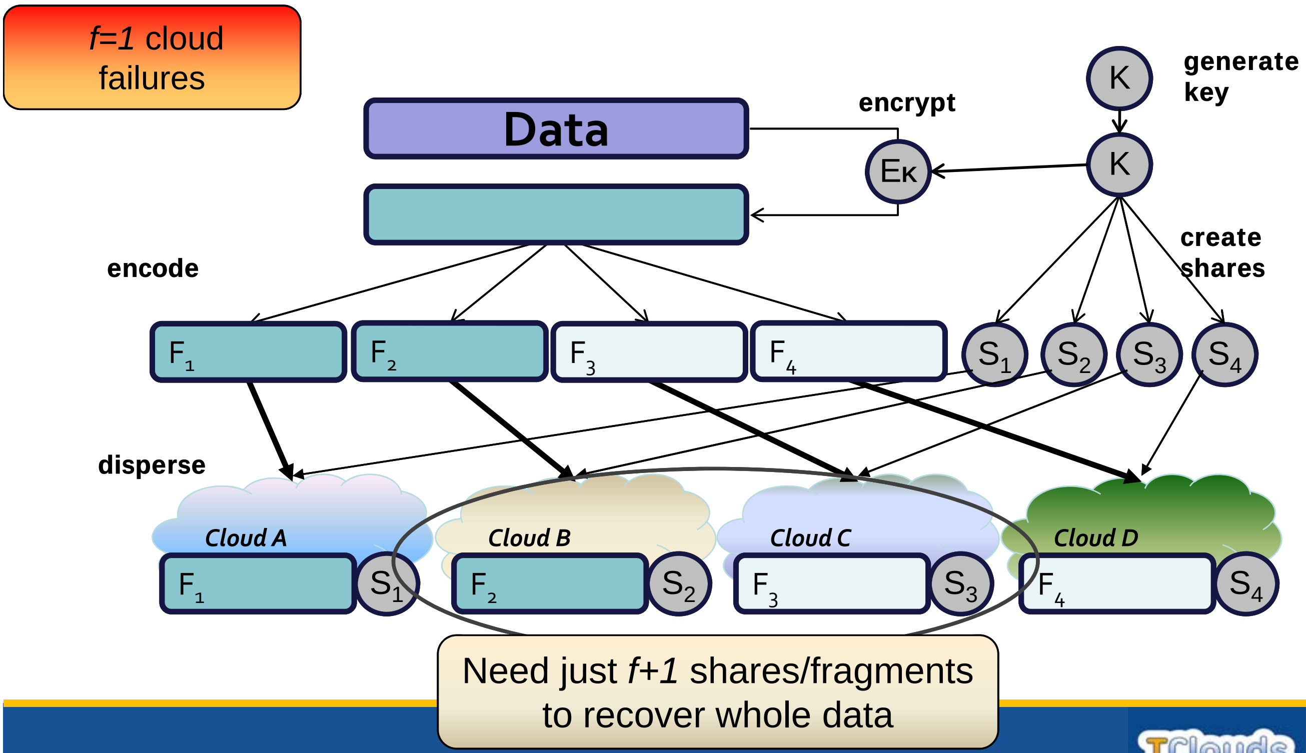


Limitations:

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

Storage Confidentiality, Availability and Efficiency

Combining Erasure Codes, Robust Secret Sharing and Quorums [Krawczyk 1993]



Consistency Proportionality

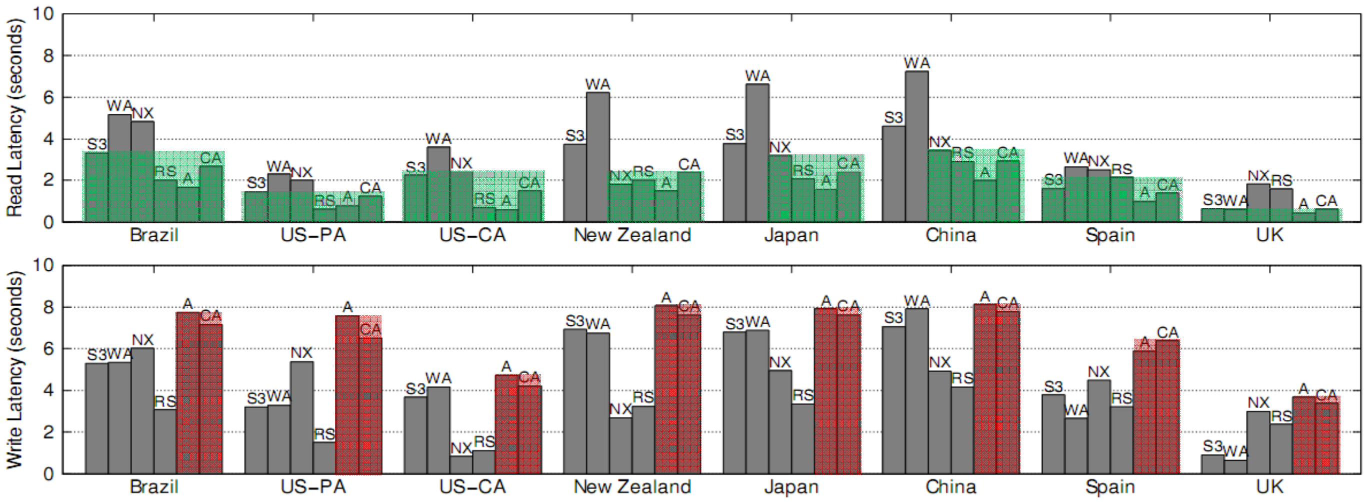
- The consistency provided by DepSky is the same as the base storage clouds
  - If the weakest consistency cloud provides eventual consistency, DepSky provides eventual consistency
  - If the weakest consistency cloud provides "read your writes", DepSky provides "read your writes"
  - If the weakest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

DepSky Evaluation

DepSky Latency (100kb DU)

A (avail.+integrity)
CA (+confidentiality)

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 Operation 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	<- /2	1.46	+	0.78	+	0.98	+	2.93

- <u>DepSky oper. Costs (USD) for 1Mb data unity and **four** clouds</u>
  - 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>DepSky storage cost (1M data unit, w/ confidentiality):</u>
  - 2×(Avg. individual cloud cost per GB/month)

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

- perceived availability of DepSky better than 0,99995
- Apparently, some clouds don't provide the promised 5 or 6 9's of availability
- Internet availability plays an important role

Paulo Verissimo
http://navigators.di.fc.ul.pt

Thank you!

TClouds