Critical applications on the cloud?

- depending on one cloud (or provider thereof) is not enough to build trust
- E.g., privacy- and security-critical data storage
  - Medical records
  - Government, Company financial data
  - Critical infrastructures data (e.g. smart grid)

![Diagram of Critical System and Cloud Storage Failure or Disclosure]
**TClouds big challenge**

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

**Trusted-Trustworthy Clouds**

Option 1:

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 1:
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)

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

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

---

**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?
TClouds Diverse ecosystem - big picture

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?
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 Architecture

TIS – key to modularity and encapsulation

- TClouds can either be **deployed by final users, third-party providers, or commodity providers** wishing to directly offer some form of cloud resilience
- To allow **seamless deployment** of these instantiations, we introduce the **TIS – TClouds Information Switch** concept, which encapsulates all or part of the services defined in the TClouds architecture
- These units, with the adequate configuration and placement, materialize the several TClouds incarnations in a modular way
Macro view: Arch. with Client-resident SW

Macro view: Arch. with OEM Appliance box
Some results:

Storage Confidentiality, Availability and Efficiency
Combining Erasure Codes, Robust Secret Sharing and Quorums

\[ f=1 \text{ cloud failures} \]

\[ f+1 \text{ shares/fragments to recover whole data} \]
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 Operation Costs ($)

<table>
<thead>
<tr>
<th>Operation</th>
<th>DepSky-CA</th>
<th>Amazon S3</th>
<th>Rackspace</th>
<th>Win. Azure</th>
<th>Nirvanix</th>
</tr>
</thead>
<tbody>
<tr>
<td>10K Reads</td>
<td>1.47</td>
<td>1.46</td>
<td>2.15</td>
<td>1.46</td>
<td>1.46</td>
</tr>
<tr>
<td>10K Writes</td>
<td>3.08</td>
<td>3.08</td>
<td>0.78</td>
<td>0.98</td>
<td>2.93</td>
</tr>
</tbody>
</table>

- **Oper. Costs** (in USD) for 1Mb data unity and **four** clouds
  - Read cost is the same as reading from the less expensive cloud
  - Write cost is the sum of writing 50% of the DU size on each cloud

- **DepSky storage cost** (1M data unit, w/ confidentiality):
  - $2 \times \text{(Avg. individual cloud cost per GB/month)}$
### DepSky Perceived Availability

<table>
<thead>
<tr>
<th>Location</th>
<th>Reads Tried</th>
<th>DEPSKY-A</th>
<th>DEPSKY-CA</th>
<th>Amazon S3</th>
<th>Rackspace</th>
<th>Azure</th>
<th>Nirvanix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brazil</td>
<td>8428</td>
<td>1.0000</td>
<td>&lt;0.9998</td>
<td>1.0000</td>
<td>&lt;0.9997</td>
<td>&lt;0.9993</td>
<td>&lt;0.9986</td>
</tr>
<tr>
<td>US-PA</td>
<td>5113</td>
<td>1.0000</td>
<td>1.0000</td>
<td>&lt;0.9997</td>
<td>1.0000</td>
<td>&lt;0.9995</td>
<td>&lt;0.9986</td>
</tr>
<tr>
<td>US-CA</td>
<td>8084</td>
<td>1.0000</td>
<td>1.0000</td>
<td>&lt;0.9997</td>
<td>1.0000</td>
<td>&lt;0.9995</td>
<td>&lt;0.9986</td>
</tr>
<tr>
<td>New Zealand</td>
<td>8545</td>
<td>1.0000</td>
<td>1.0000</td>
<td>&lt;0.9997</td>
<td>1.0000</td>
<td>&lt;0.9995</td>
<td>&lt;0.9986</td>
</tr>
<tr>
<td>Japan</td>
<td>8392</td>
<td>1.0000</td>
<td>1.0000</td>
<td>&lt;0.9997</td>
<td>1.0000</td>
<td>&lt;0.9995</td>
<td>&lt;0.9986</td>
</tr>
<tr>
<td>China</td>
<td>8594</td>
<td>1.0000</td>
<td>1.0000</td>
<td>&lt;0.9997</td>
<td>1.0000</td>
<td>&lt;0.9995</td>
<td>&lt;0.9986</td>
</tr>
<tr>
<td>Spain</td>
<td>6550</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>&lt;0.9997</td>
<td>&lt;0.9995</td>
<td>&lt;0.9986</td>
</tr>
<tr>
<td>UK</td>
<td>7069</td>
<td>1.0000</td>
<td>1.0000</td>
<td>&lt;0.9998</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

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

### Byzantine Fault-Tolerant MapReduce: Faults Are Not Just Crashes

(Map perspective)

- Byzantine Fault-Tolerant MapReduce ensures data is accurately processed even in the presence of faulty nodes.
Cloud-of-Clouds
Dependable Storage System for Smart Grid

Mnemonic for Contacts:
google “Navigators home page”

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

Thank you!