

A Resource Description Language with Vagueness Support for Multi-Provider Cloud Networks



Gregor Schaffrath, **Stefan Schmid**, Ishan Vaishnavi, Ashiq Khan, Anja Feldmann

Telekom Innovation Laboratories (Berlin) & DoCoMo Eurolabs (Munich), August, 2012

A Resource Description Language with Vagueness Support for Multi-Provider Cloud Networks



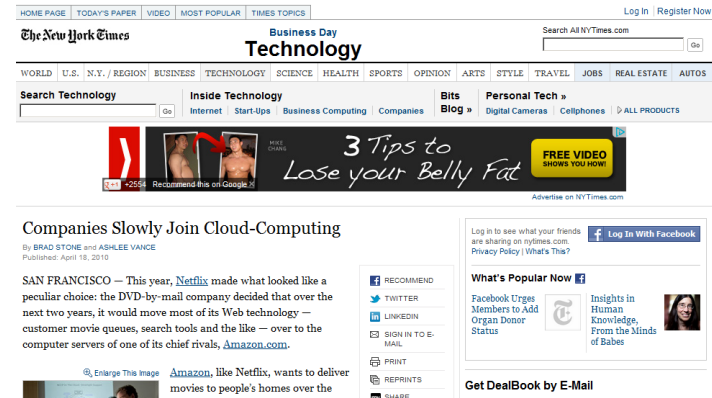
Gregor Schaffrath, **Stefan Schmid**, Ishan Vaishnavi, Ashiq Khan, Anja Feldmann

Telekom Innovation Laboratories (Berlin) & DoCoMo Eurolabs (Munich), August, 2012

Vision: Virtual Networking Cloud Resources.



Cloud computing is a big success!
But what is the point of clouds if they cannot be accessed?



CloudNets: Next Natural Step for Virtualization!

Success of Node Virtualization

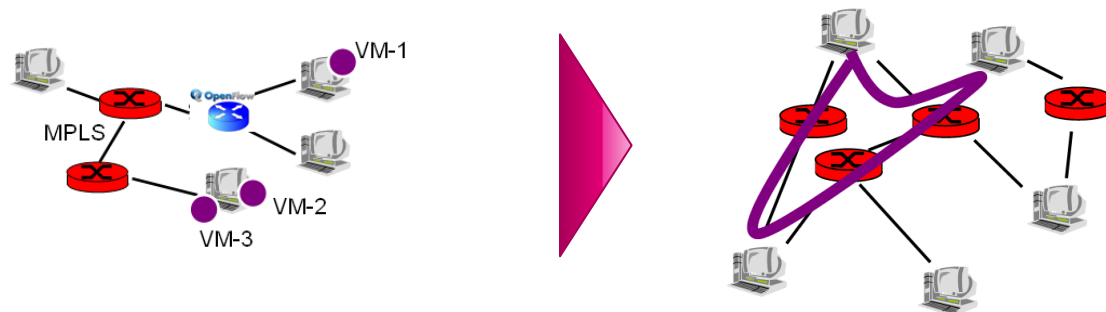
- a.k.a. end-host virtualization
- VMWare revamped server business
- VM = basic unit in datacenters...
- ... hardly any physical resources!
- VM = flexible allocation, migration...

Trend of Link Virtualization

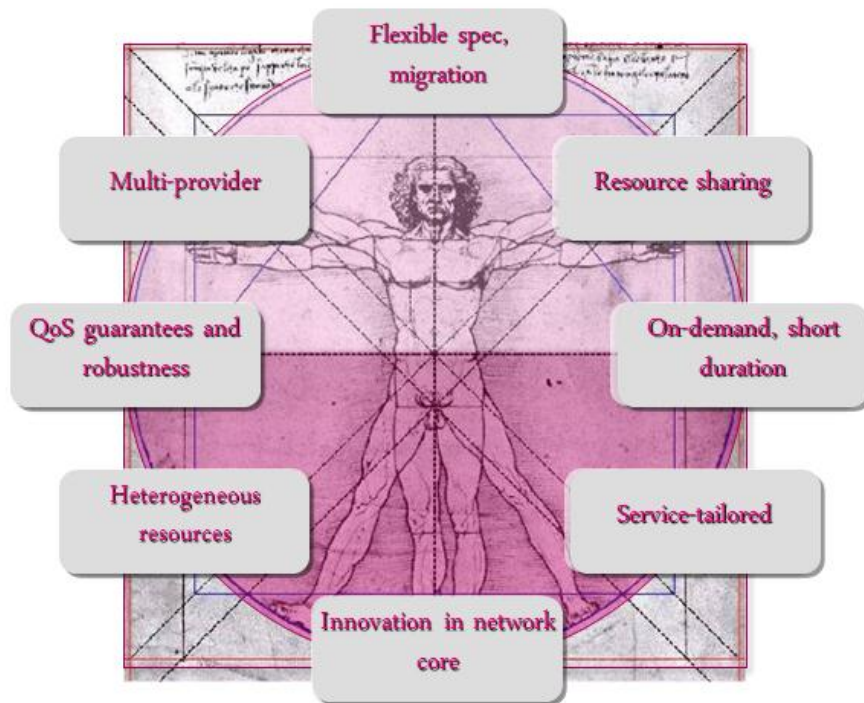
- VLANs
- Software Defined Networks (SDN), OpenFlow, ...

Unified, fully virtualized networks: **CloudNets**

„Combine **networking** with heterogeneous **cloud resources** (e.g., storage, CPU, ...)!“



Virtualization = **Decoupling** of services / CloudNets from physical infrastructure (substrate network)



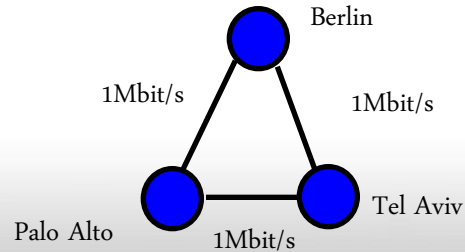
Therefore:

- Can allocate CloudNet flexibly subject to specification (flexible **service deployment**)
- CloudNets can **cohabit** same substrate network in isolation
- Abstracts **heterogeneous** resources
- CloudNets can even run different **protocol stacks**!

Some Use Cases.

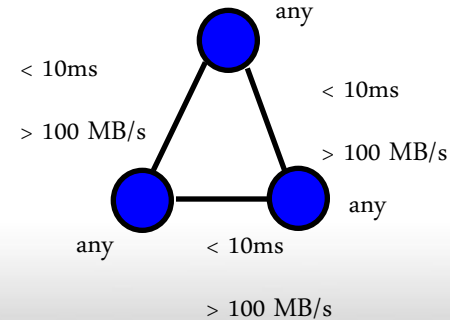
„VPN++“

Goal: Fully specified CloudNet mapping constraints (e.g., end-points for a **telco**), but with **QoS guarantees** (e.g., bandwidth) along links



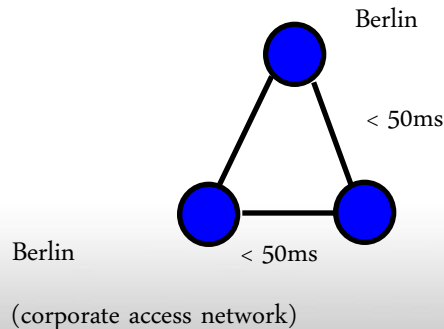
**„November 22,
1pm-2pm!“**

Datacenters



**„Guaranteed
resources, job
deadlines
met, no
overhead!“**

Spillover/Out-Sourcing

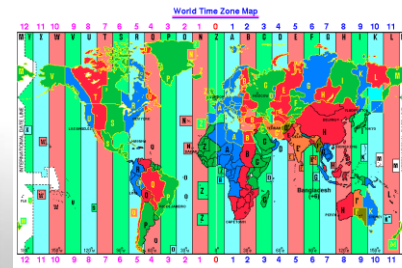


**„50 TB storage, 10
Tflops computation!“**

**„any European
cloud provider
(e.g. due to
legal issues?)“**

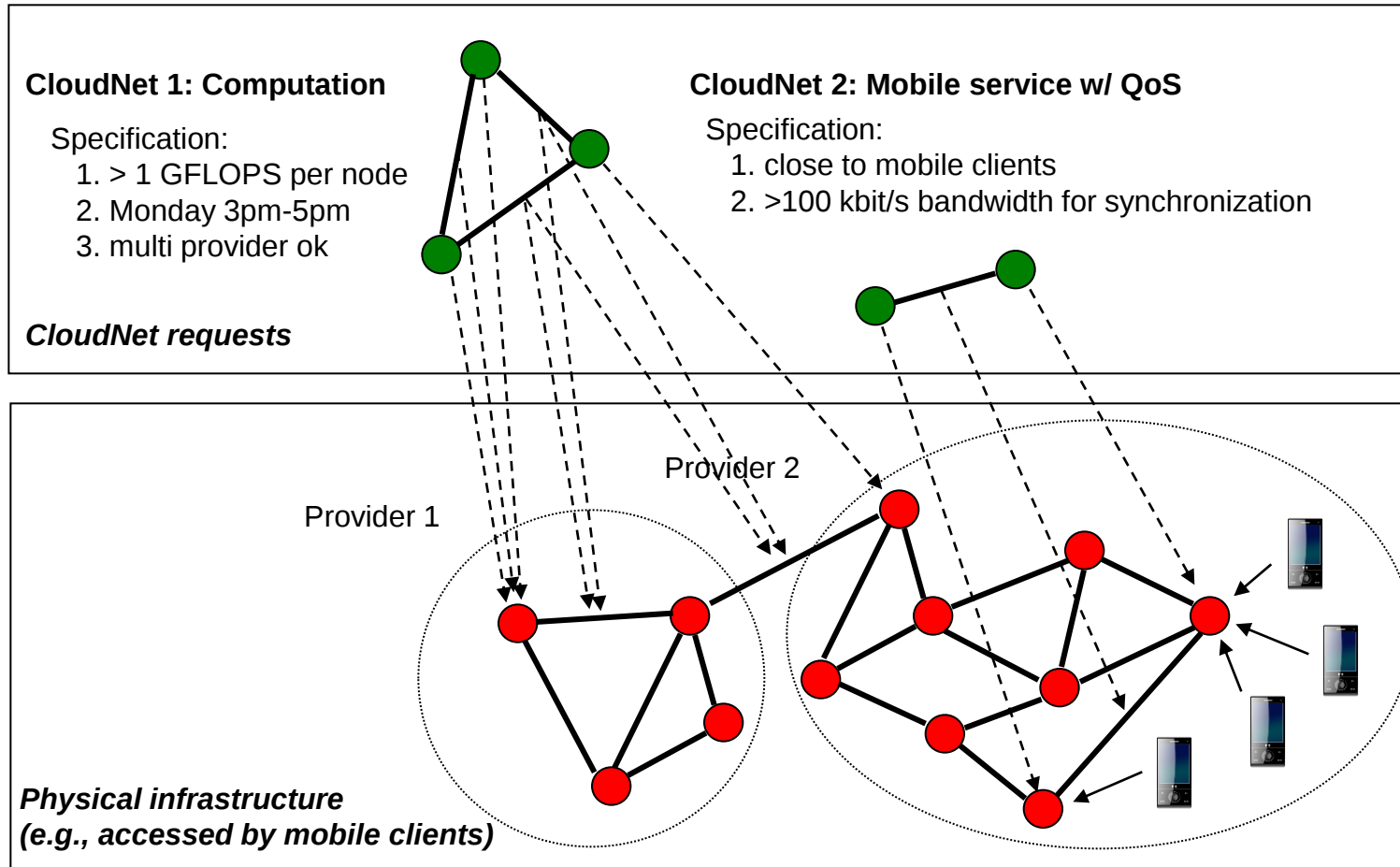
Migration

Goal: Move with the sun, with the commuters, (QoS) allow for **maintenance**, avoid roaming costs...: e.g., **SAP/game/translator server, small CDN server...**



Example of CloudNet Embedding.

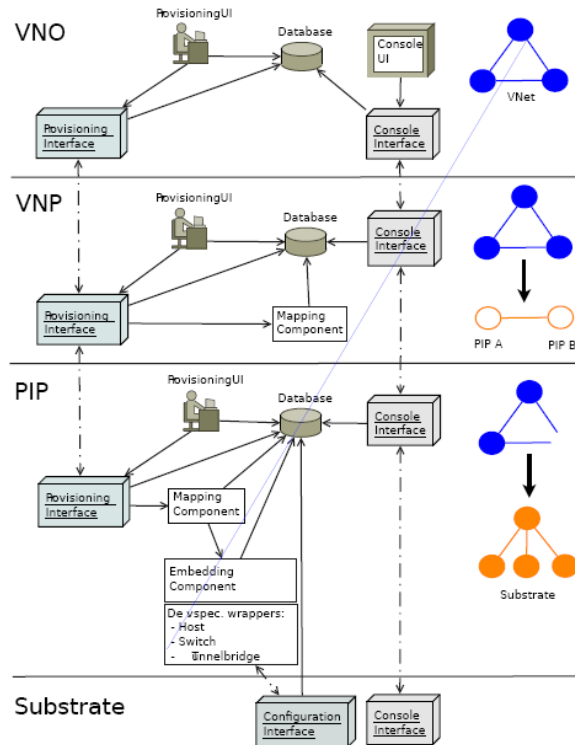
Connecting Providers (Geographic Footprint).



Business Roles.

CloudNet opens new business roles!

We propose that properties and prices are negotiated between economic roles.



Roles

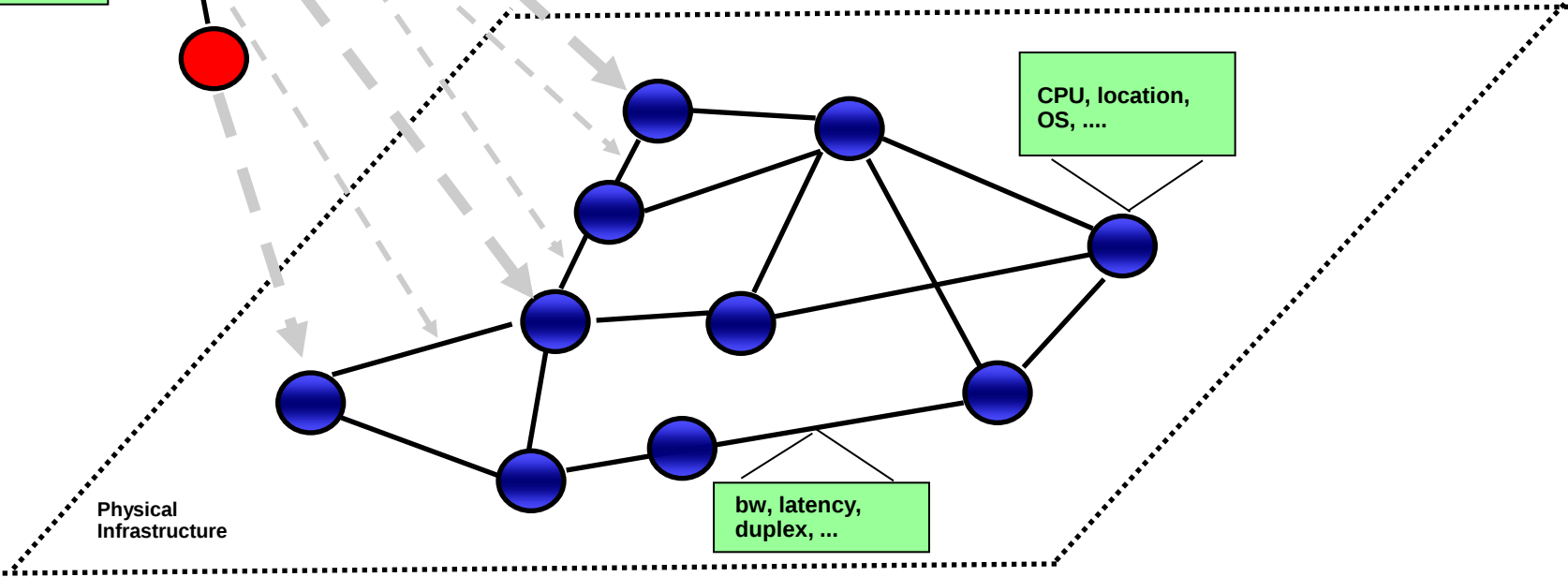
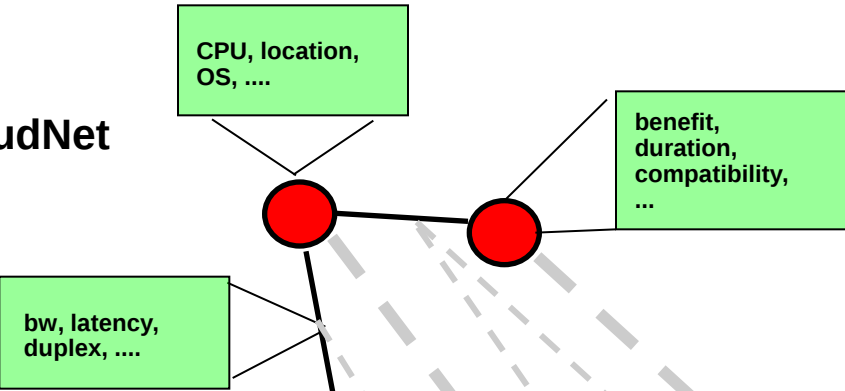
- Service Provider (SP): uses CloudNets to offer its services (e.g., **value-added application** CloudNet, or transport CloudNet)
- Virtual Network Operator (VNO): Installs and **operates** CloudNet over topology provided by VNP, offers **tailored** connectivity service, triggers cross-provider migration (by setting requirements), ...
- Virtual Network Provider (VNP): „Broker“/reseller that assembles virtual resources from different PIPs to provide **virtual topology** (no need to know PIP, can be **recursive**, ...)
- Physical Infrastructure Provider (PIP): Owns and manages physical **infrastructure**

QoS from PIP up to VNO or service provider: **accounting** via complete **set of contracts!** (unlike „sending party pays“)

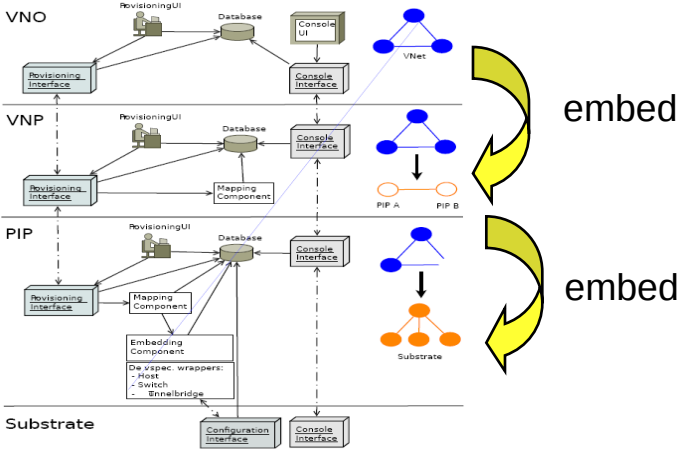


Recursive Embedding.

CloudNet



Some specifications in VNet may be missing!

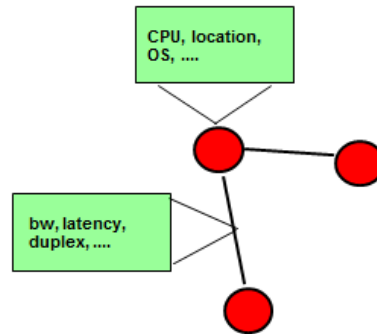


**CloudNets require a flexible
resource description
language: FleRD!**

**FleRD is used in our
CloudNet prototype...**

- **Communicate** CloudNets, substrate resources and embeddings to business partners or customers:

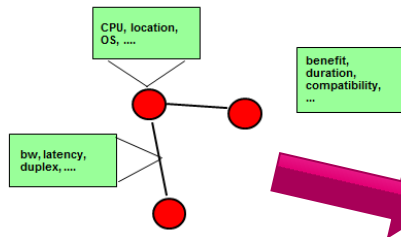
SP



benefit,
duration,
compatibility,
...



PIP

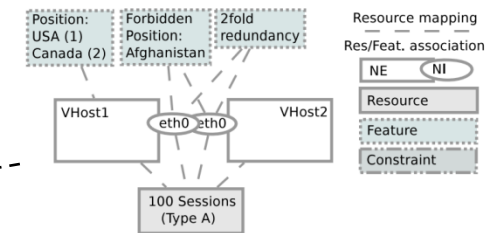
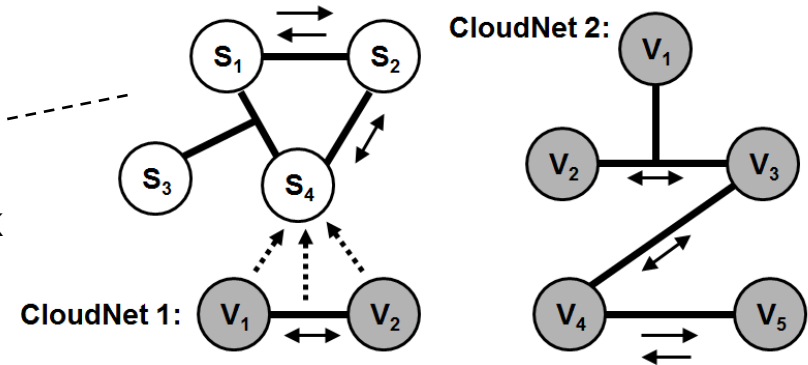


- **Store** embedding state internally:

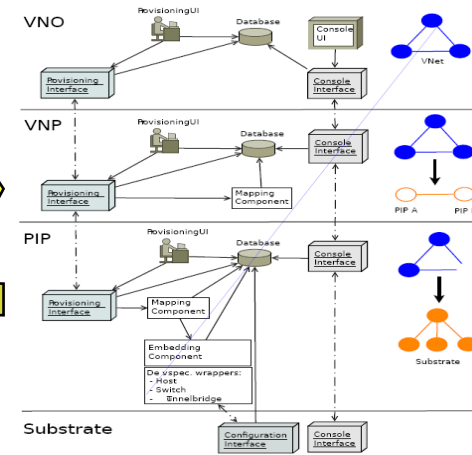


FleRD Requirements.

- Support all kinds of node (storage, computation, ...) and link (latency, bw, full-duplex/asymmetric, ...) resources (**heterogeneity**)
- Extensible, allow for **syntactic changes** over time, no need for **global agreement** on semantic values
- Facilitate resource leasing and allow PIPs to open **abstract views** on their substrate
- Allow for **vagueness and omission**: customers are unlikely to specify each CloudNet detail (e.g., KVM or Xen is fine, outsource to any European cloud provider): this opens ways for optimization (**exploiting flexibilities**)!
- Allow for **aggregation** of resources (**business secret?**)
- Non-topological requirements (e.g., wordsize compatibility)

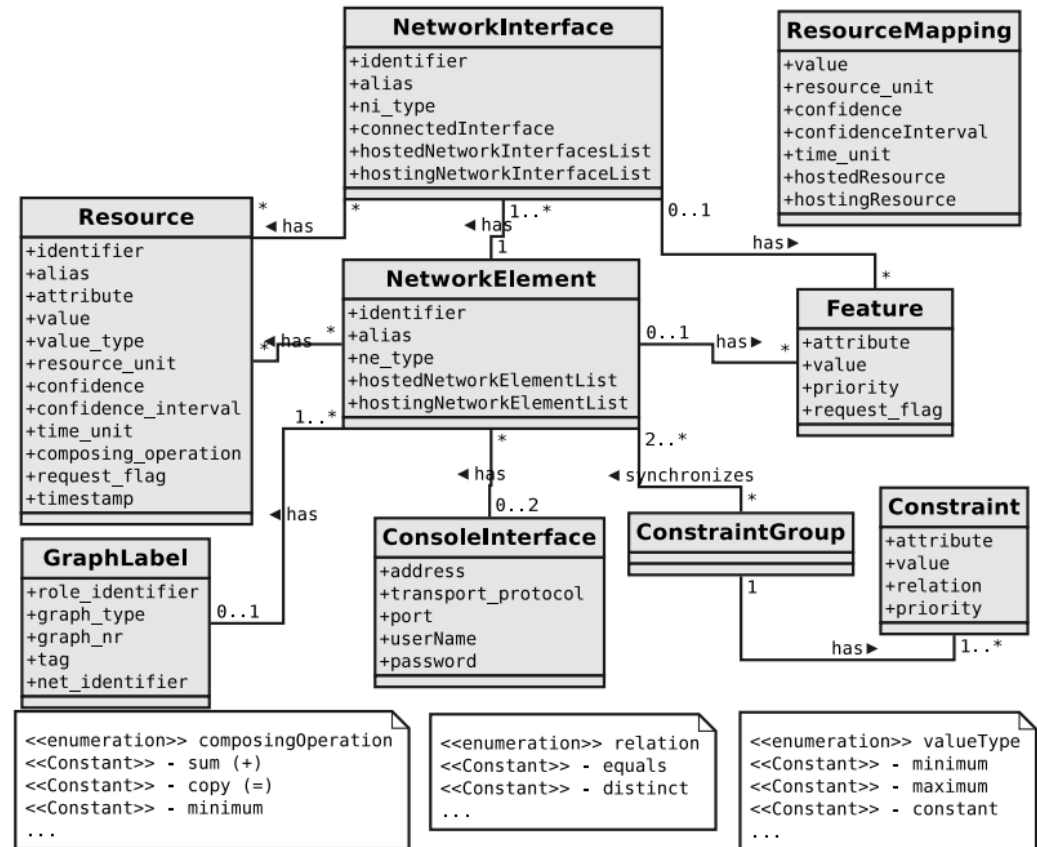


aggregate resources



FleRD Design Principles.

- Based on **generic objects** (semantic value only of **local interest**): **Network Elements (NEs)**
- Interconnected via **Network Interfaces (NI)**
- Note: NEs **can be links**, can model **multicast links** by connecting NIs accordingly!
- NE properties are described by **attribute-value-pairs**: **Resources** (e.g., amount of storage) and **Features** (e.g., CPU flags)
- Predicates** over features (e.g., for white or black listing)
- Constraint groups** allows to specify non-topological properties
- Mapping** can be recursive
- Attribute space **hierarchical** (compare e.g. to SNMP, like «link/symmetric/bw»)

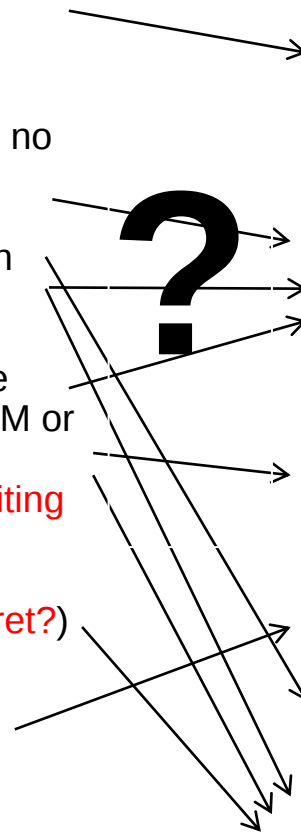


Problem:

- Support all kinds of node (storage, computation, ...) and link (latency, bw, full-duplex/asymmetric, ...) resources (**heterogeneity**)
- Extensible, allow for **syntactic changes** over time, no need for **global agreement** on semantic values
- Facilitate resource leasing and allow PIPs to open **abstract views** on their substrate
- Allow for **vagueness and omission**: customers are unlikely to specify each CloudNet detail (e.g., KVM or Xen is fine, outsource to any European cloud provider): this opens ways for optimization (**exploiting flexibilities**)!
- Allow for **aggregation** of resources (**business secret?**)
- Non-topological requirements (e.g., wordsize compatibility)

Solution:

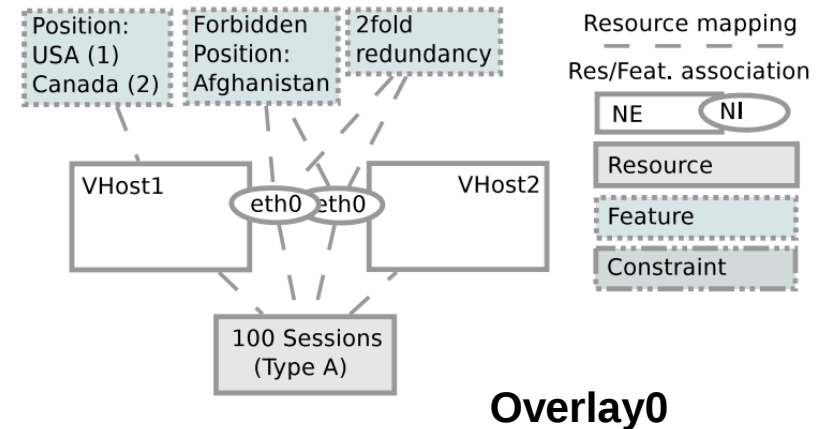
- Based on **generic objects** (semantic value only of **local interest**): **Network Elements (NEs)** interconnected via **Network Interfaces (NI)**
- NE properties are described by **attribute-value-pairs**: **Resources** (e.g., amount of storage) and **Features** (e.g., CPU flags)
- **Predicates** over features (e.g., for white or black listing)
- **Constraint groups** allows to specify non-topological properties
- **Mapping** can be recursive
- Attribute space **hierarchical** (compare e.g. to SNMP)



A Use Case: Web Service.

Service Provider asks for Web Service:

- VHost1 and VHost2 (i.e., service distributed over two processes)
- VHost1 in US or Canada (**white listing**)
- VHost2 **vague**
- Connection not via Afghanistan (**black listing**)
- **2-redundant** path

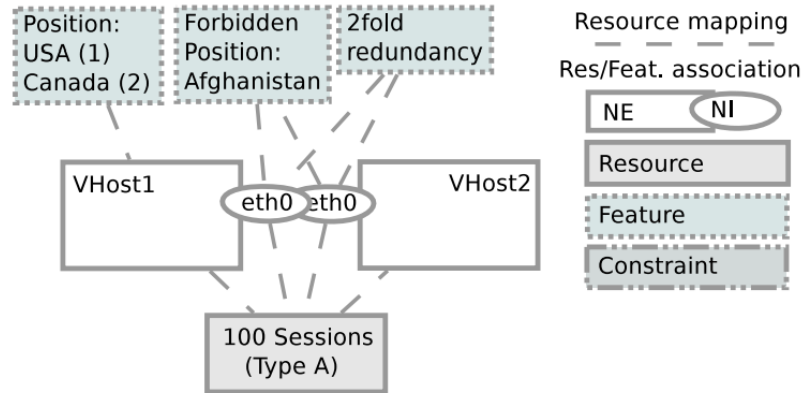


How is request embedded on physical infrastructure? Stages can all be described with FleRD!

- Different stages (could be merged if same role but just to illustrate concepts...)
- Build **Overlay0** and **Overlay1** out of request
- **Mapping layer** to annotate configuration details (e.g., VLAN tags) and to keep mapping **state**
- Substrate layer: **Underlay0** and **Underlay1**



Overlay0.



XML version (excerpt): Features of VHost 1

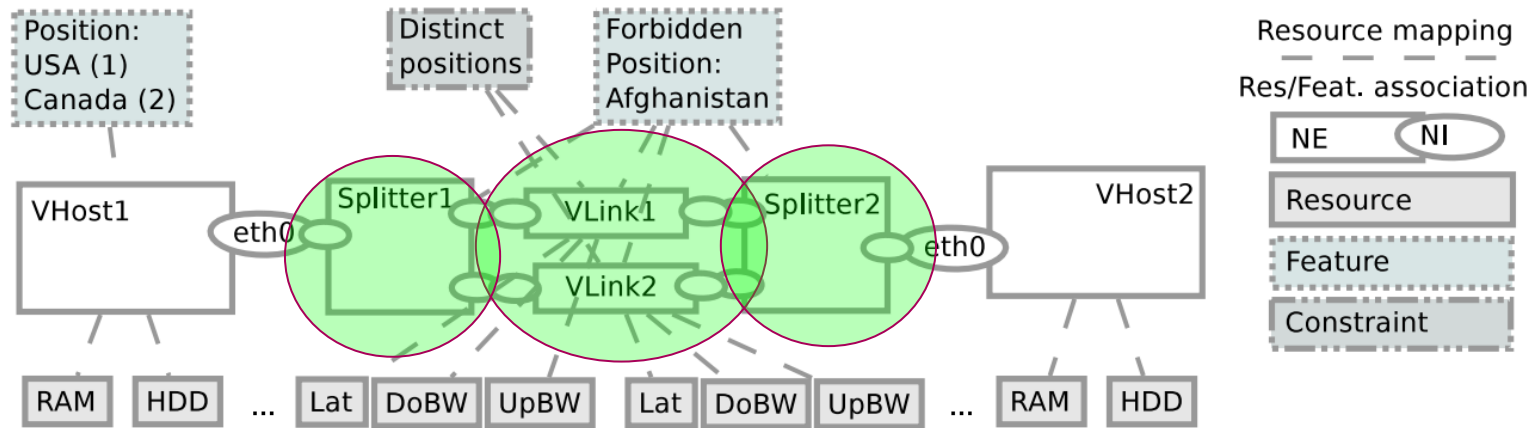
```
<features>
<! -- X: whitelist
-- this node may be hosted in the US (preferred) or Canada -- >
<Feature>
  <attribute>/position/continent/country</attribute>
  <value>/NA/USA</value>

  <priority>1</priority>
  <request_flag>true</request_flag>
</Feature>
<Feature>
  <attribute>/position/continent/country</attribute>
  <value>/NA/Canada</value>
  <priority>2</priority>
  <request_flag>true</request_flag>
</Feature>
</features>
```

two alternative
locations with
priorities



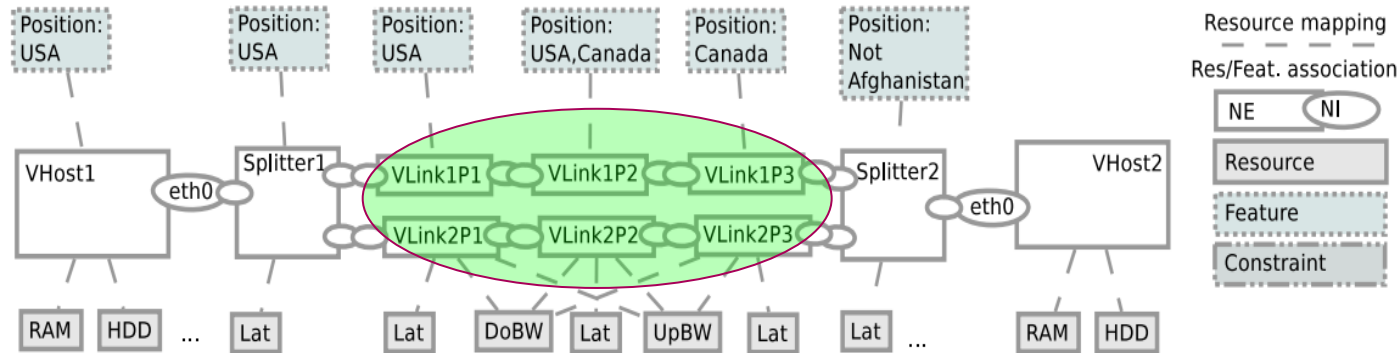
Overlay1.



- Expand request to make more concrete
- Add **two explicit links** / paths
- Virtual hosts have RAM and HDD **resources**
- Virtual links have up and down stream requirements
- **Multiplex** components (splitters) are added to split the link in disjoint paths
- **Constraints** are added to ensure disjoint paths



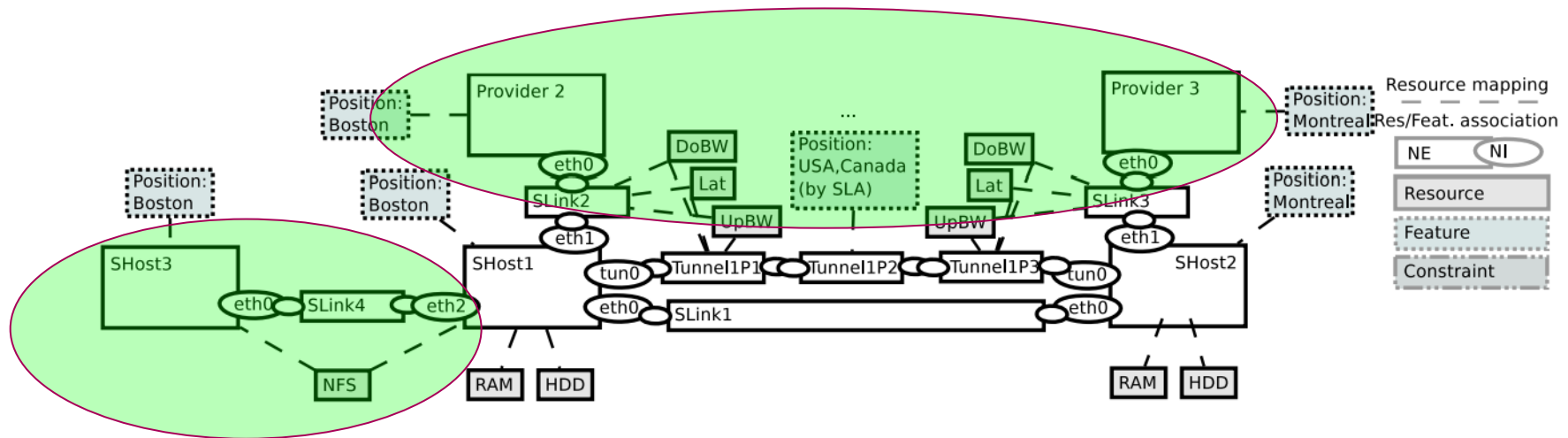
Mapping Layer.



- To annotate config details (e.g., VLAN tags) or keep mapping state
- One-to-many vs **many-to-many mappings?**
- Optional «bridge» between OL and UL
- Virtual links are expanded into **three virtual hops** (one in US, one in Canada, one between)



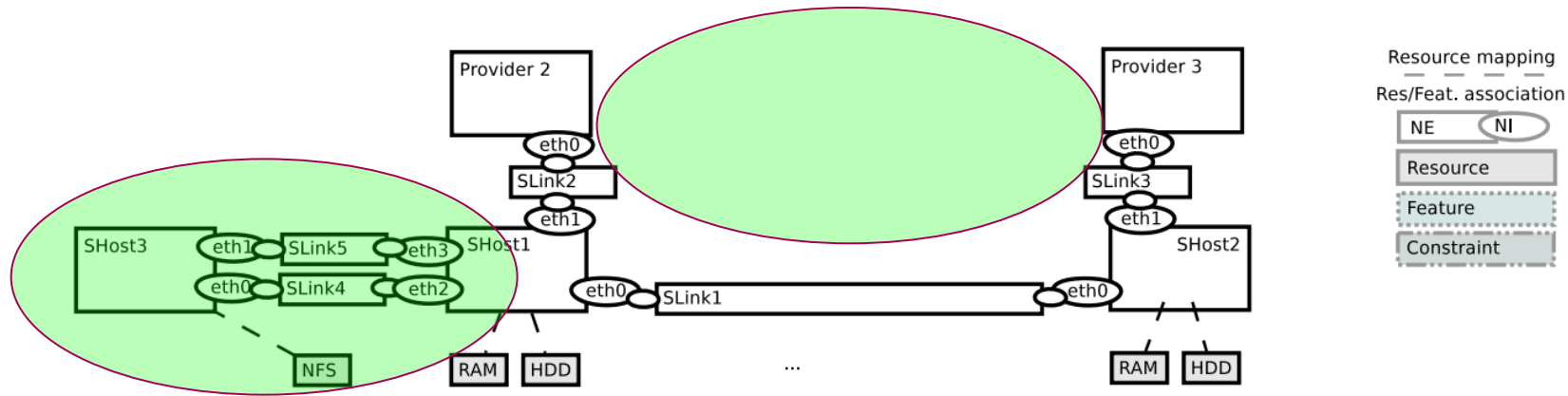
Underlay1.



- Reason for two underlays: keep both logical structure of the substrate (as used for mapping) and representation of physical setup
- Underlay1 still contains **VPN tunnel** and **shared NFS**
- VHost1 is **mapped** to SHost3 and Splitter1 is mapped to VHost1, VHost2 and Splitter2 are mapped to VHost2...
- Upper virtual link realized via **3 tunnel segments** (via external physical providers Provider2 and Provider3)
- Lower virtual link within given provider
- NFS still logically attached to substrate nodes



Underlay0.



- Physical situation of considered provider (Provider1)
- NFS hosted only by one host (SHost3), plus link to access NFS from SHost1
- Tunnel counterparts not visible



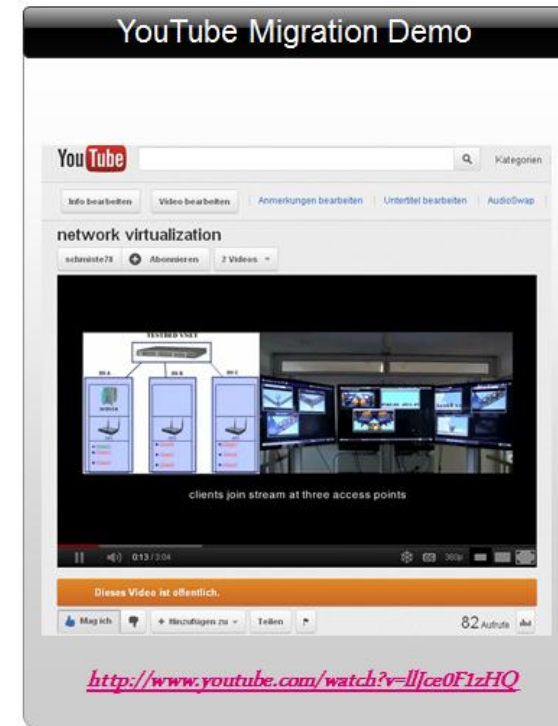
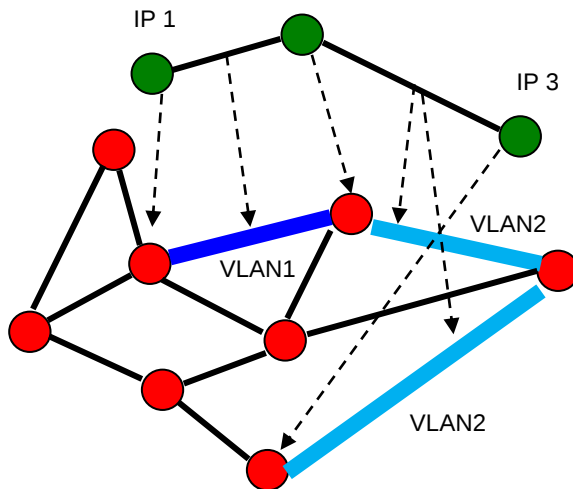
Related Work.

- **Rspec (PlanetLab, ProtoGENI)**
 - Resources not modelled as **separate referencable entities**, not extensible and unsuitable in multi-layer and multi-provider setting
- **NDL+OWL: ontology for automated reasoning**
 - not flexible and **syntactic overhead**
- **OCCI (Open Grid Forum)**
 - no multi-layers support
- **DEN-ng: business-driven network management**
 - explicit distinction between virtual and physical network, **no multi-layer support**
- **VN-SLA: service level specifications**
 - no mappings
- **VxDL: virtual execution infrastructure description**
 - no mappings
- **NOVI**
 - no resource mappings
- **NDL**
 - no support for efficient **multi-home links**, no **omission**, or black and white listing, ...



Conclusion.

- FleRD is used in our own CloudNet **prototype** implementation
- Prototype based on **VLANs**
- Flexibilities exploited via **Mixed Integer Program** embeddings



- Migration demo

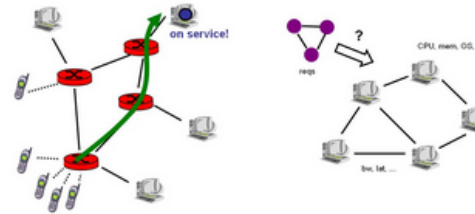


The Project Website.

Combining Clouds with Virtual Networking

The CloudNet Project

Internet Network Architectures (INET)
TU Berlin / Telekom Innovation Labs (T-Labs)
Contact: [Stefan Schmid](#)



[News](#)

[Overview](#)

[People](#)

[Magazines](#)

[Publications](#)

[Demo](#)

[Talks/Posters](#)

News

- Watch on YouTube: migration demonstrator [video!](#)
- We are looking for students and interns with good algorithmic background to contribute to Virtul [Contact us](#) for more details or have a look at some [open topics](#).

Project Overview

CloudNets are virtual networks (VNets) connecting cloud resources. The **network virtualization** paradigm allows to run multiple CloudNets on top of a shared physical infrastructure. These CloudNets can have different properties (provide different security or QoS guarantees, run different protocols, etc.) and can be managed independently of each other. Moreover, (parts of) a CloudNet can be migrated dynamically to locations where the service is most useful or most cost efficient (e.g., in terms of energy conservation). Depending on the circumstances and the technology, these migrations can be done live and without interrupting ongoing sessions. The flexibility of the paradigm and the decoupling of the services from the underlying resource networks has many advantages; for example, it facilitates a more efficient use of the given resources, it promises faster innovations by overcoming the ossification of today's Internet architecture, it simplifies the network management, and it can improve service performance.

We are currently developing a **prototype system** for this paradigm (currently based on VLANs), which raises many scientific challenges. For example, we address the problem of where to embed CloudNet requests (e.g., see [1] for online CloudNet embeddings and [2] for a general mathematical embedding program), or devise algorithms to migrate CloudNets to new locations (e.g., due to user mobility) taking into account the



Collaborators and Publications.

■ People

- **T-Labs / TU Berlin:** Anja Feldmann, Carlo Fürst, Johannes Grassler, Arne Ludwig, Matthias Rost, Gregor Schaffrath, Stefan Schmid
- **Uni Wroclaw:** Marcin Bienkowski
- **Uni Tel Aviv:** Guy Even, Moti Medina
- **NTT DoCoMo Eurolabs:** Group around Wolfgang Kellerer

■ Publications

- **Prototype:** VISA 2009, ERCIM News 2012, ICCCN 2012
- **Migration:** VISA 2010, IPTComm 2011, Hot-ICE 2011
- **Embedding:** ICDCN 2012 (*Best Paper Distributed Computing Track*)



Contact.



Dr. Stefan Schmid

Telekom Innovation Laboratories
Ernst-Reuter-Platz 7, D-10587 Berlin
E-mail: stefan@net.t-labs.tu-berlin.de

Project website:
<http://www.net.t-labs.tu-berlin.de/~stefan/virtu.shtml>

