Decentralized Control of Large-Scale Distributed System

Mario Südholt

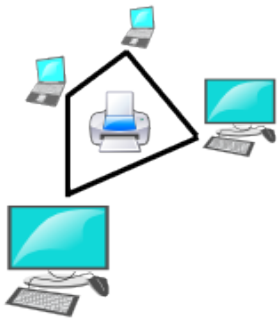
Département Informatique, Mines Nantes

IMT Colloquium; March 26, 2014

Outline

Networks and systems

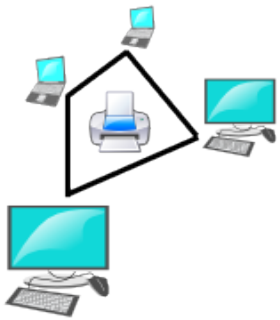
LAN



Networks and systems

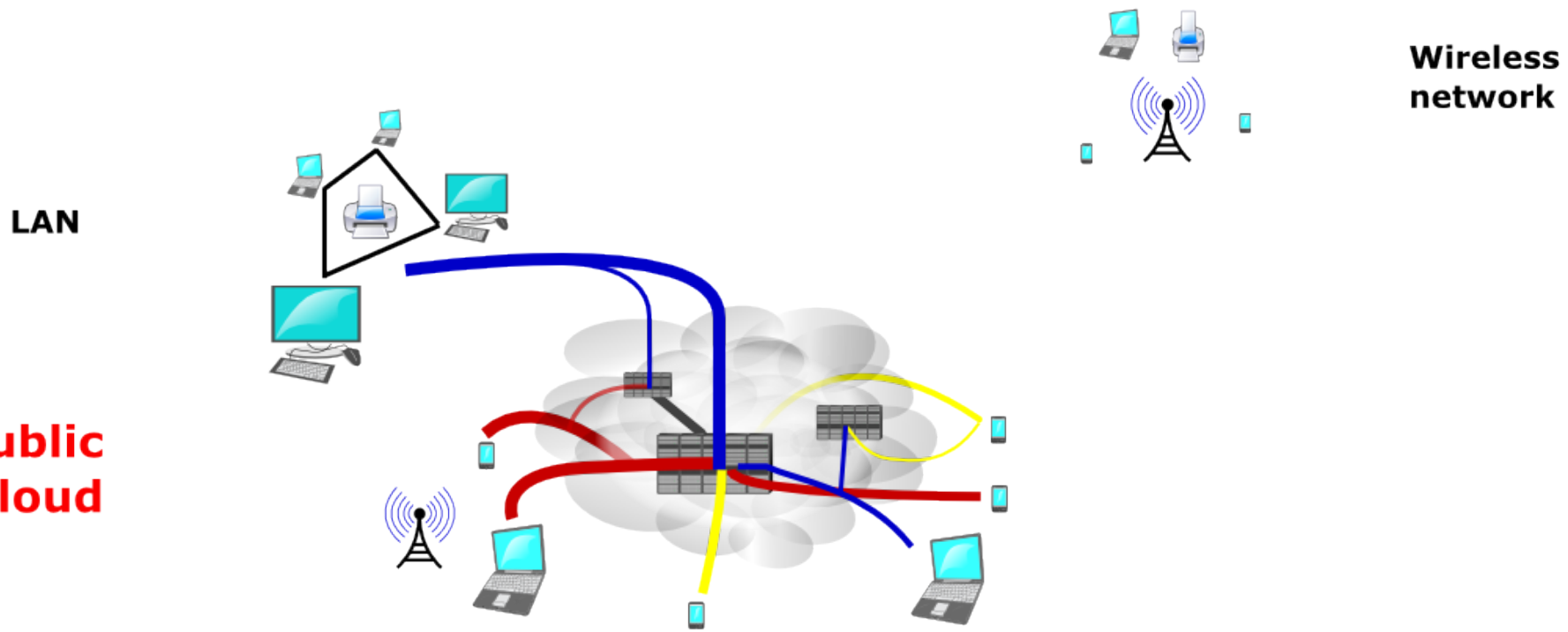
INSTITUT Mines-Télécom

LAN

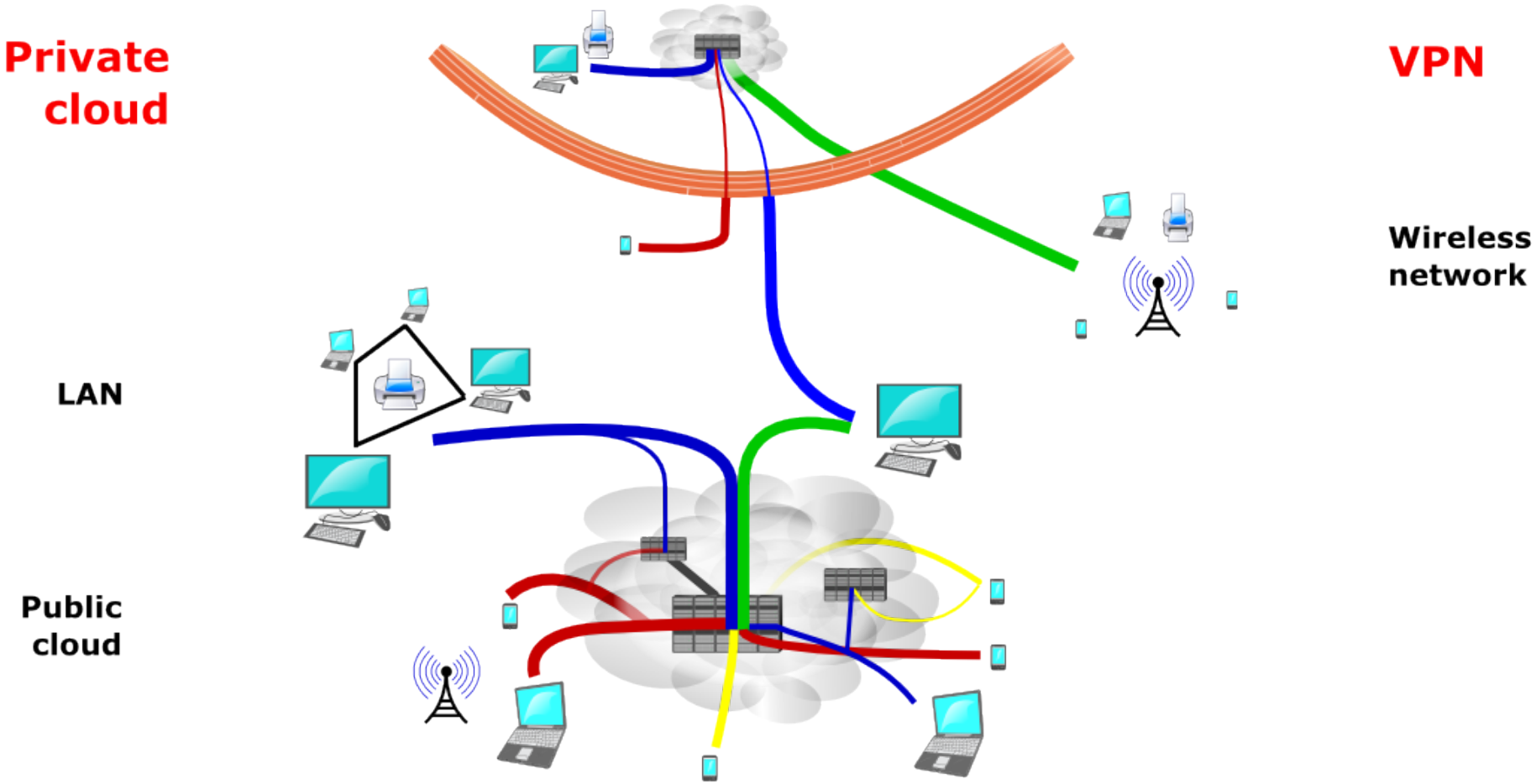


Wireless network

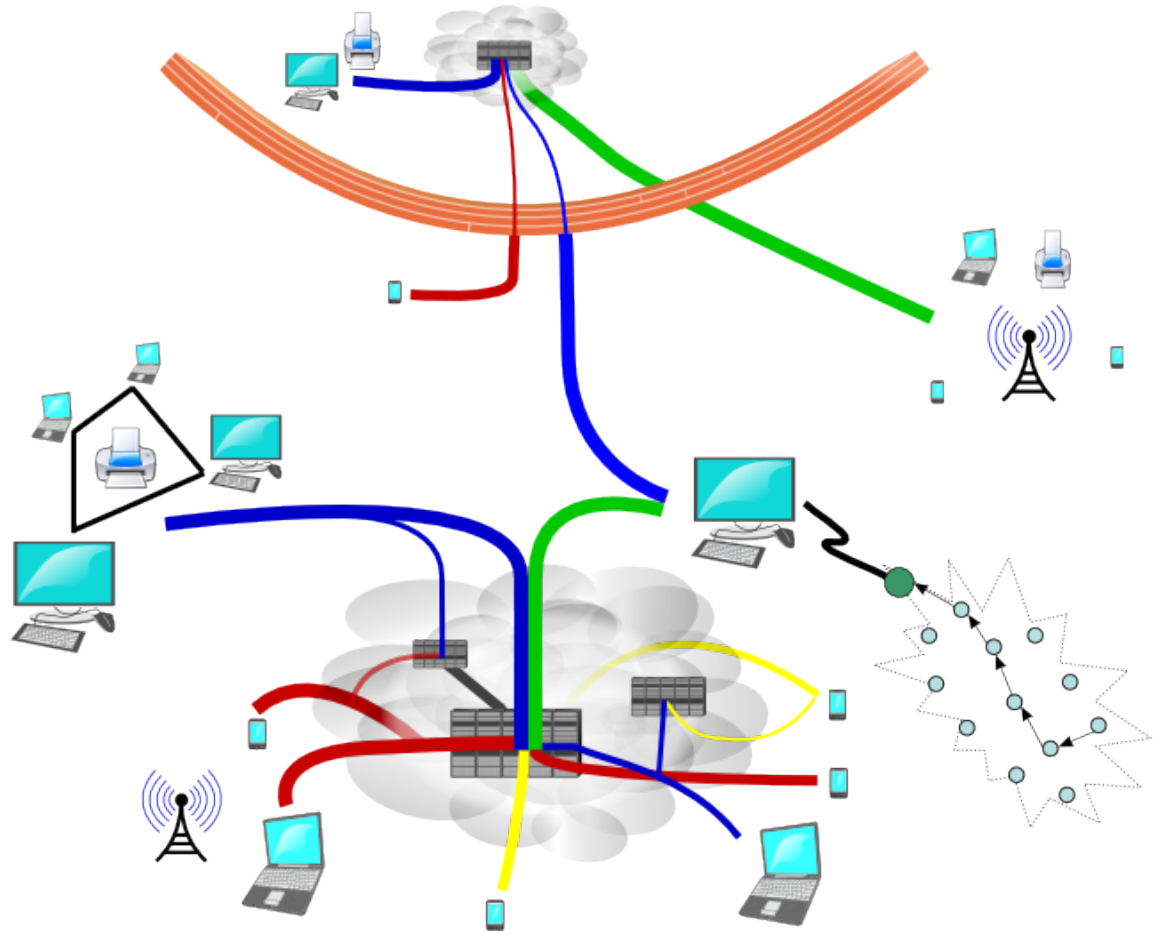
Networks and systems



Networks and systems



Networks and systems



Large-scale software systems

INSTITUT Mines-Télécom

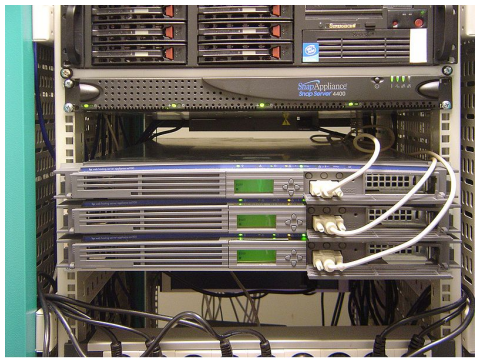
- Large-scale software systems (we are interested in)
  - The Web
  - **The Cloud**
- Major questions
  - How are they **built and coordinated**?
  - New **architectures and implementation mechanisms**?
  - How to ensure **availability, correctness and security**?
  - Handle **cross-domain functionalities** (across technical and policy domains)

The Web

- Basic model: **distributed coordination of services**
  - Loose coupling
  - (Some) Well-defined, standardized interfaces
  - REST interfaces
  - Centralization often arises: popular services, service orchestration …
- Correctness, security
  - **Standards, protocols** for low-level properties
  - But: frequent **violation of high-level properties**
    - Ex.: social cross-site forgery (S-CSRF) attacks
- Support for **distributed property enforcement**?

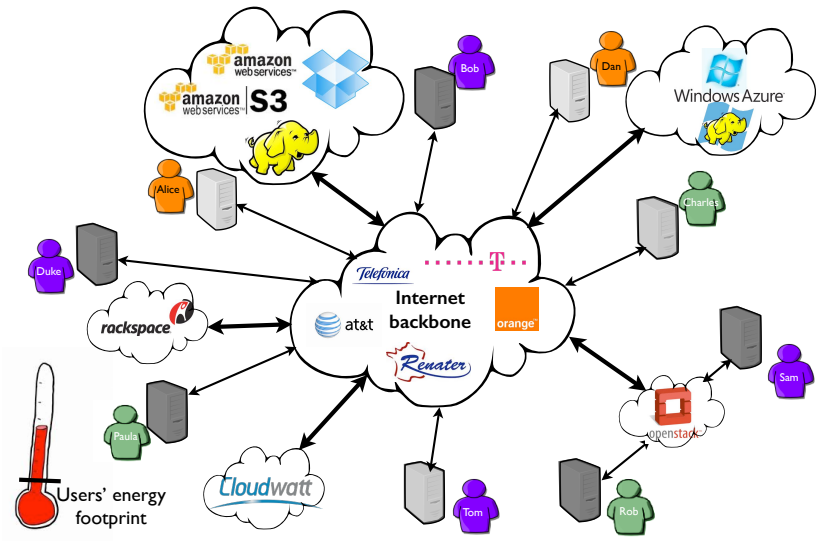
The Cloud

- **Mutualize resources** required by many users
- Many **types**
  - Public (Cloudwatt, Numergy, Amazon, Google, Microsoft, …)
  - Private, community, hybrid clouds
- Different **service levels**: IaaS, PaaS, SaaS
- Homogeneous environment
  - Hardware: datacenters (up to hundreds of thousands of servers)
  - Software: **virtual environments**



Cloud federations

- **Mutualize resources among one or several providers**
- Scale cloud services over geographic regions
  - Significant centralization
- Problems
  - Availability
  - Connectivity
  - Energy consumption ("ice clouds")
  - Legal issues (data privacy)



Issues with centralized control

INSTITUT Mines-Télécom

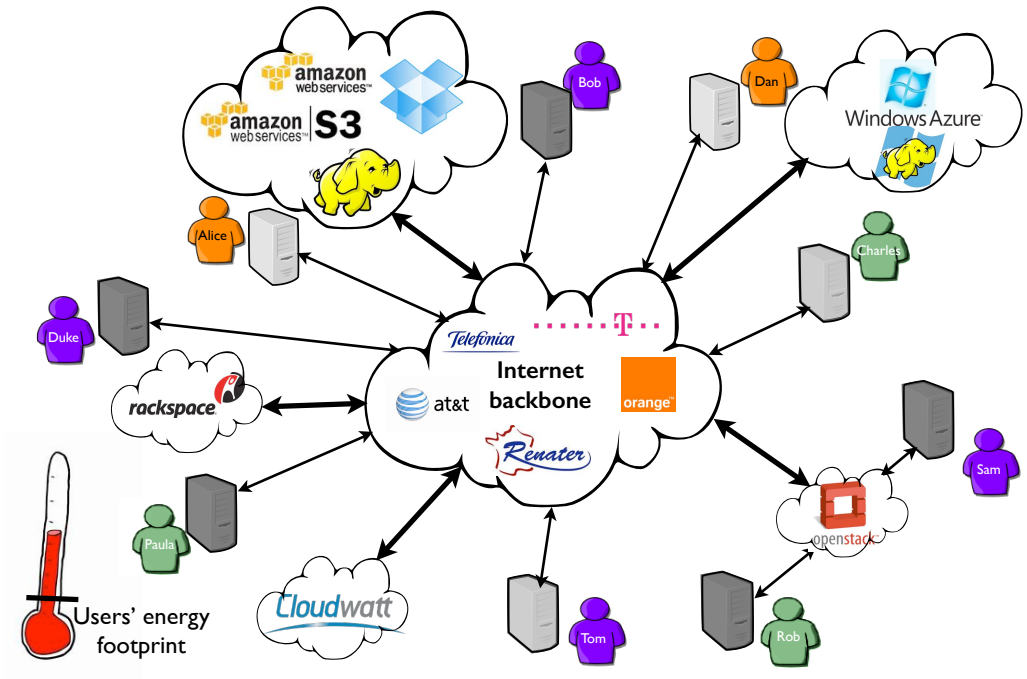
- Recap: centralized control in large-scale infrastructures
  - Cloud: significant centralization
  - Web: access to special servers, service orchestration
- Scaling issues, legal issues
- Issues on the **system architecture level**
  - Ex.: datacenters at the edge of the backbone
- **Implementation-level issues**
  - Ex.: centralized capacity planning in datacenters

Outline

1. The Discovery initiative: architecture

A new architectural principle:

from federated clouds …

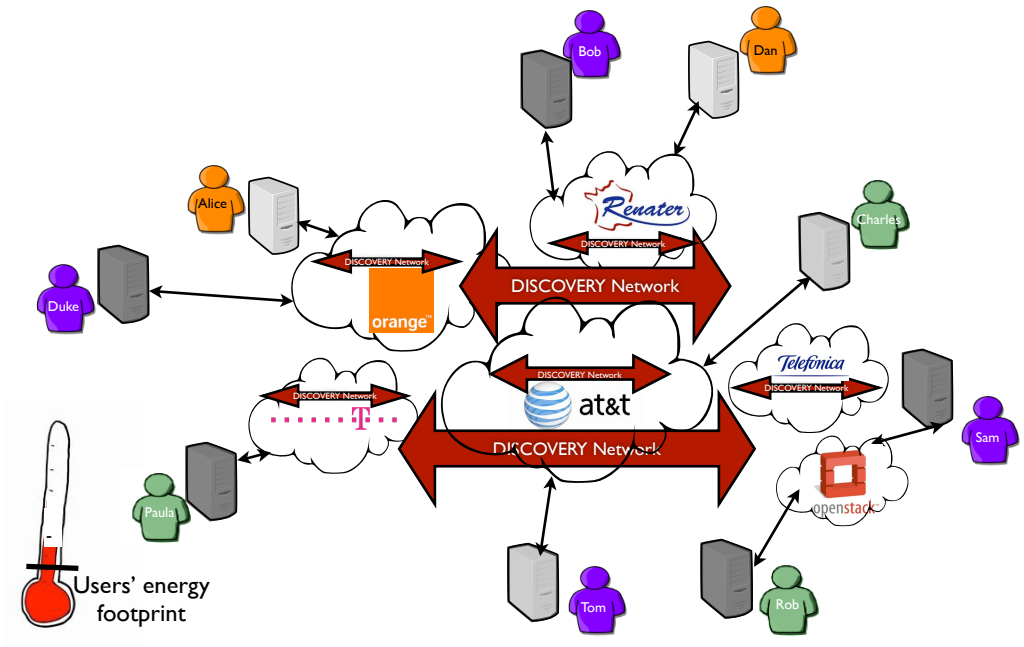


1. The Discovery initiative: architecture

INSTITUT Mines-Télécom

A new architectural principle:

. . . to cooperative clouds



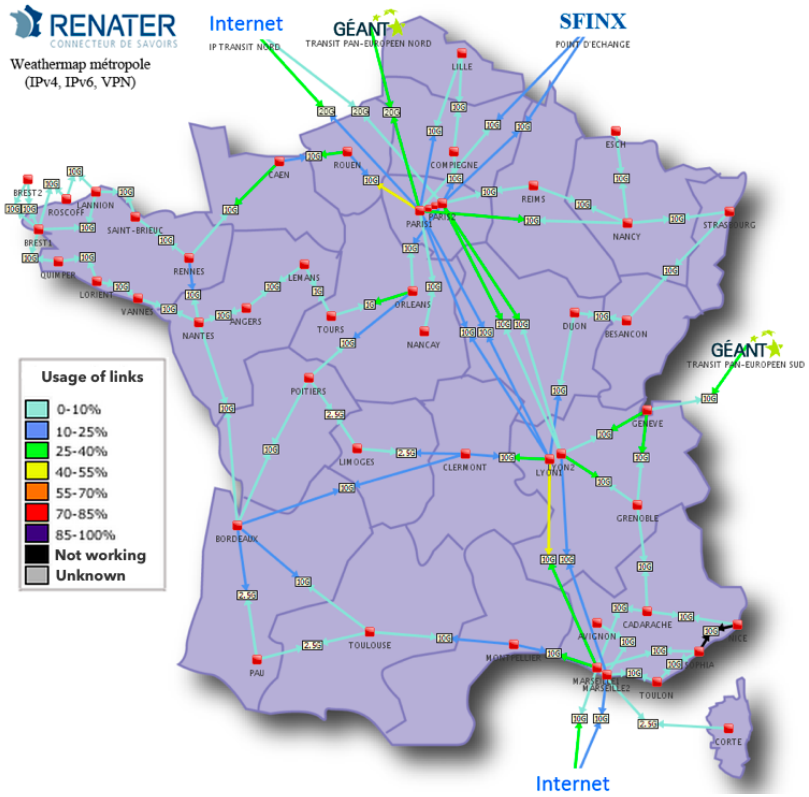
Main characteristics

- **Cooperative and autonomous management of virtual environments**
  - Manipulate virtual environments like processes in traditional OSes
- **Localization** of data and computations
  - Key to efficiency and sustainability

Locality in backbones (ex. Renater)

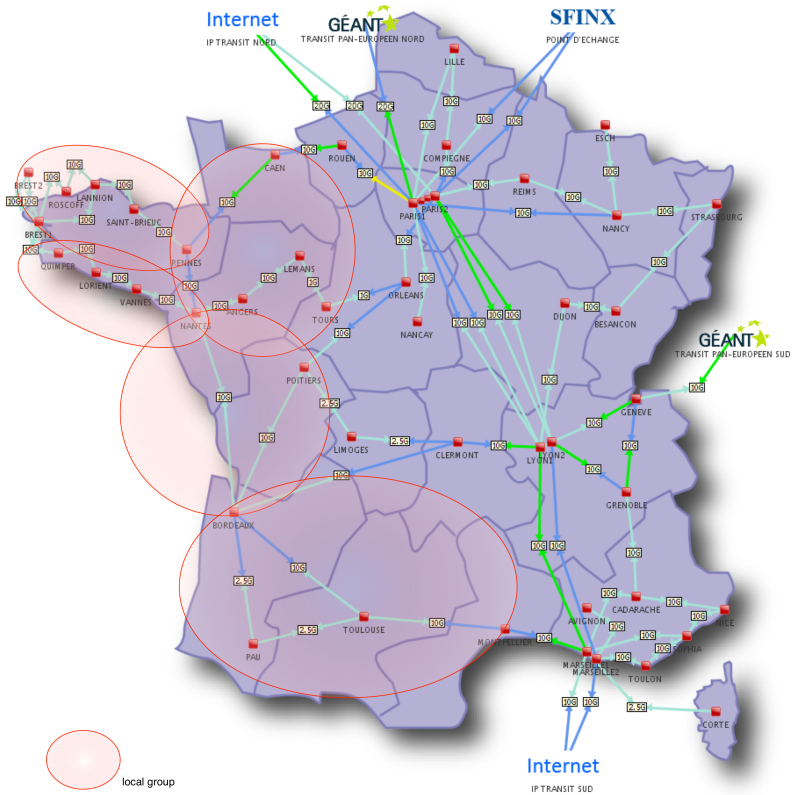
INSTITUT Mines-Télécom

- Network state on 17 May 13
  - Underutilized links
  - Redundancy
  - Evolves in terms of points-of-presence (PoP)
- **Potential for "close" Clouds**



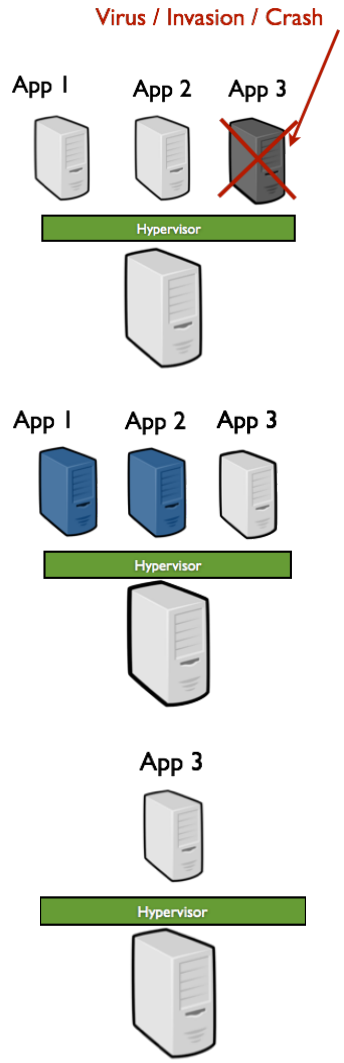
Distributed cooperative clouds (ex. Renater)

- **Close deployment** to network infrastructure
- **Extend network hubs with servers**
  - Dedicated to VM hosting
  - Proportional to PoP's size



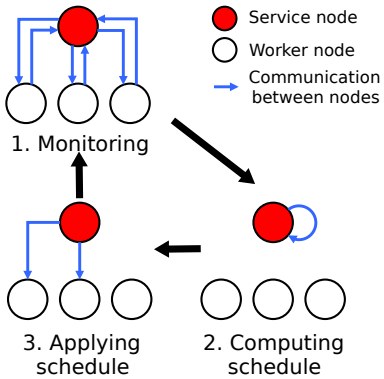
2. Capacity planning: virtual machines

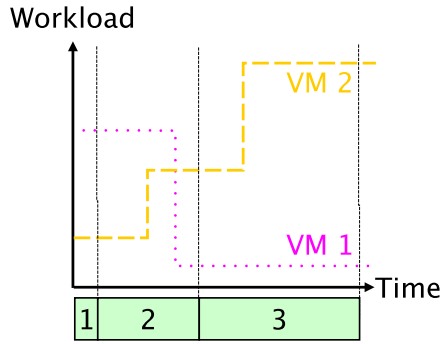
- **Virtual machines**: software emulation of a computer
- Advantages
  - Isolation
  - Snapshotting
  - Suspend/resume
  - **Fast live migration** in a datacenter
    Downtime: ca. 60ms
- But: **migration plans for large sets of VMs are costly**
  - Crucial for handling over-/underutilization
  - Migration across datacenters?



VM scheduling

- Objective: **autonomously manage millions of VMs on tens of thousands of machines**
- Limitations of current approaches because of centralization
  - Reactivity and scalability
  - Fault-tolerance (single point of failure)

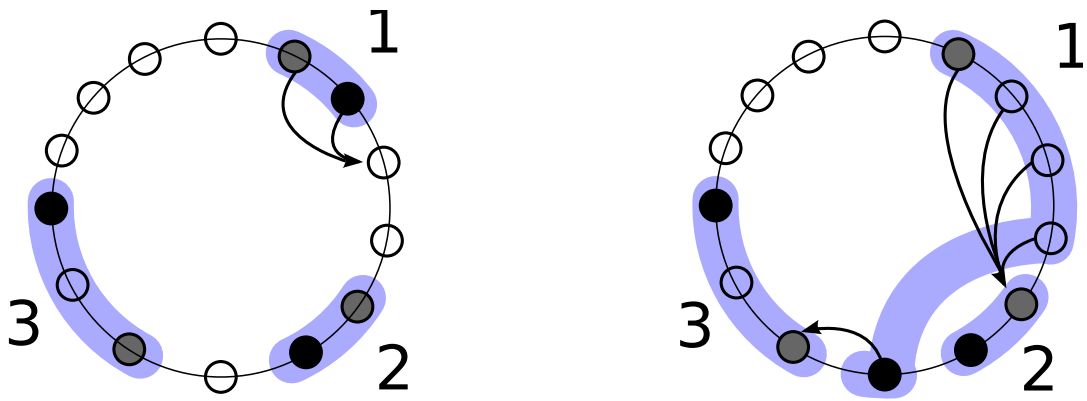




- Discovery also needs new VM scheduling strategy

Distributed VM scheduling

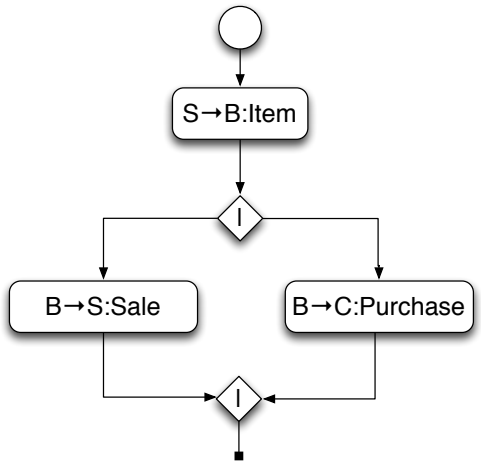
- DVMS alg.: **first fully decentralized algorithm**
  - Nodes have a local view of the system
  - Cooperation between direct neighbors to solve scheduling events



- Validation [Quesnel et al.: CCPE'12]}
  - In vivo (on Grid5000): ca. 500 physical machines, 4500 VMs
  - Simulation (using Simgrid): ca. 10K PMs, 80K VMs

3. Management of service compositions

- Service compositions (e.g., for business processes)
  - Composition programs (not manageable on large-scale)
  - Declarative definitions: orchestrations, choreographies
- **Service orchestration** (e.g., using BPEL)
  - Central chef d'orchestra
  - Subject to scalability issues (availability, lack of autonomy, …)
- **Service choreography**
  - No central orchestrator
  - Correct implementation?
  - Properties?



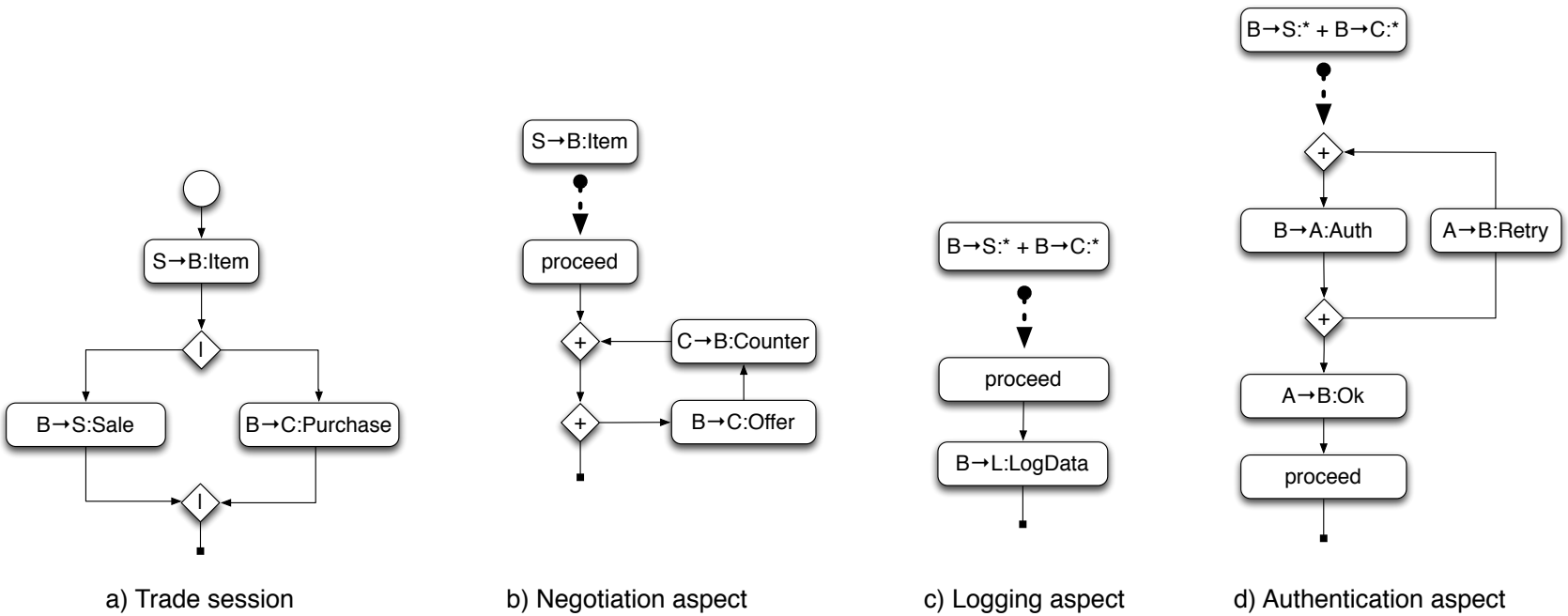
Service choreographies with session types

INSTITUT Mines-Télécom

- **Session types**: type-based fully distributed choreographies
- Global types define an interaction as a whole
  - **Projection**: compilation to correct decentralized implementation
  - Guarantee correctness properties
    - No messages send at wrong times to wrong receiver
    - No deadlocks
- From 1998 (researchers from Imperial College L., U Lisbon)
  - Multi-party session types [POPL'08]
  - Session-types with roles [POPL'12]
  - Extension by security properties [Concur'12]
- Problems
  - **Forbidden functionalities**: no race conditions
  - **Extensive rewrites for adding functionalities**

Aspectual session types

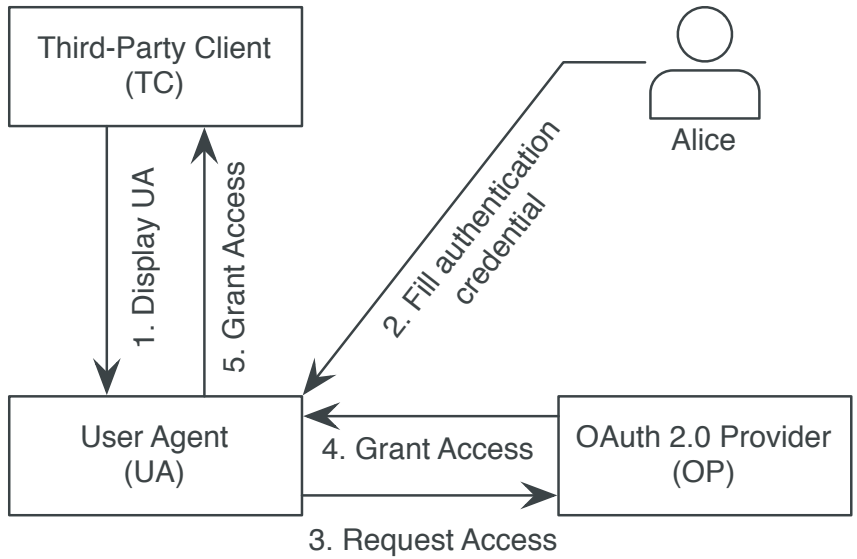
- Extension [Tabareau et al.: Modularity'14]
  - Larger set of functionalities (admit some race conditions)
  - Simple and declarative adding



a) Trade session b) Negotiation aspect c) Logging aspect d) Authentication aspect

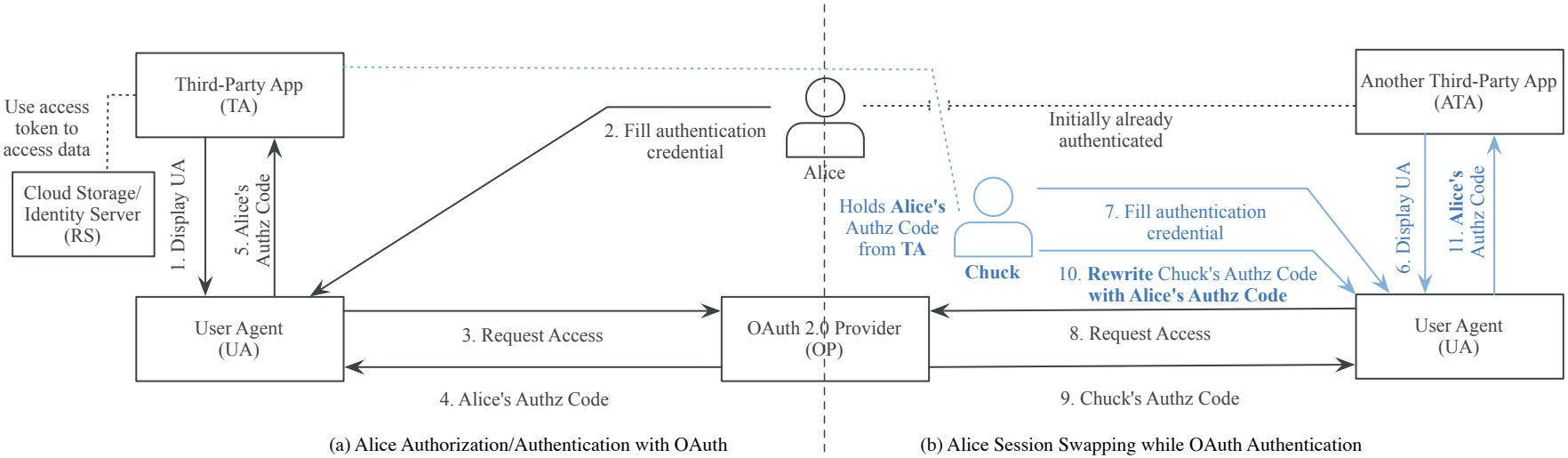
4. Protocol adaptation

- Ex. **OAuth 2.0**
  - Framework for the authorization of resource accesses
  - Access by third parties without original credentials
- Used by all major Web, Cloud and software editors companies
  - Facebook, Google, Microsoft, SAP …



Main OAuth protocol flow

OAuth single sign-on attacks



(a) Alice Authorization/Authentication with OAuth

(b) Alice Session Swapping while OAuth Authentication

- New types of distributed attacks
  - Single sign-on (SSO), **social cross-site request forgery (S-CSRF)**
  - May involve one instance of an OAuth protocol
  - May include several instances
- Problem: OAuth is a framework not a protocol
  - **Right usage has to be enforced**

Distributed transformation of protocols

INSTITUT Mines-Télécom

- **Modifications to the protocol flows needed**
  - Dynamic modifications
  - Over different steps/different instances of the protocols
  - Over different levels of the software stack
- Ex.: session identification, state introduction
- Approach [Cherrueau et al.: CoudCom'13]
  - **Domain-specific framework/protocol transformation language**
  - Invasive but controled transformation of service compositions and implementations

Conclusion

- **Centralized control (still) common and problematic**
  - Cloud architectures, capacity management, service orchestrations
- **Discovery initiative** for a cloud architecture
  - Cooperative Clouds close to users
  - Interest by large players: Renater, Orange …
- **New distributed algorithms and tools** for VM scheduling, service choreographies, protocol manipulations

Thank you for your attention!

INSTITUT Mines-Télécom

Questions?

Further information:

- Ascola research team: `http://www.emn.fr/z-info/ascola`
- `Mario.Sudholt@mines-nantes.fr`
- Discovery initiative: `Adrien.Lebre@inria.fr` (PI)