G-Lab Experimental Facility

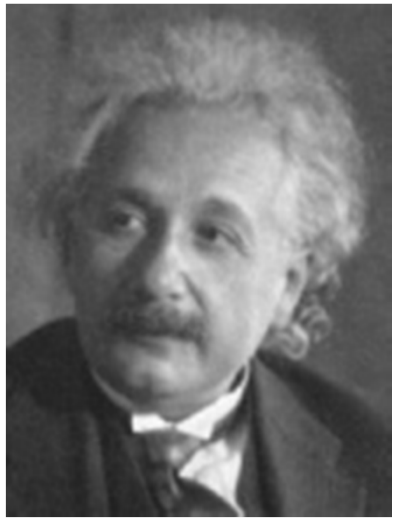
German – Korean Workshop
on Future Internet
8th – 10th March 2011
Seoul, Korea

Paul Mueller
Integrated Communication Systems Lab
Dept. of Computer Science
University of Kaiserslautern
Paul Ehrlich Bld. 34, D-67663 Kaiserslautern, Germany
Tel.+49 631 205 2263, Fax. +49 631 205 3056
www.ICSY.de

SPONSORED BY THE
Federal Ministry of Education and Research

The challenge …

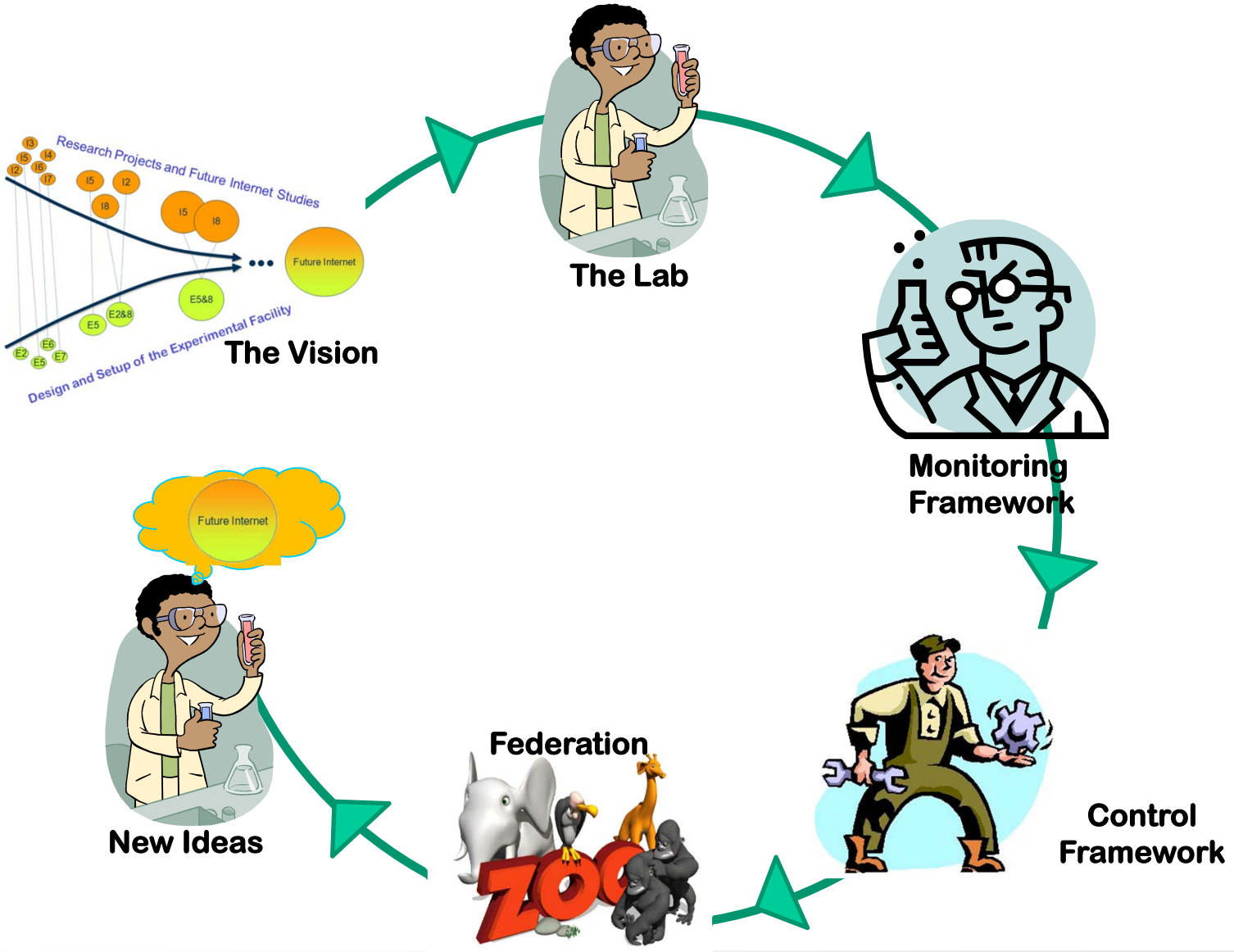
„Probleme kann man niemals mit derselben Denkweise lösen, durch die sie entstanden sind.“



“We can't solve problems by using the same kind of thinking we used when we created them.”

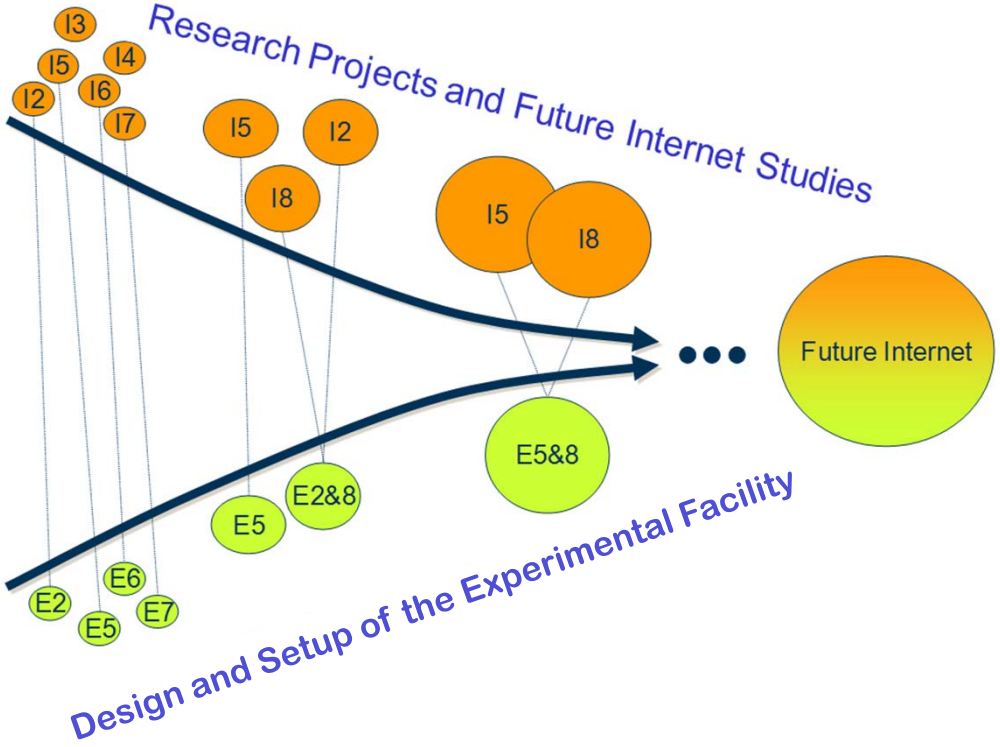
"문제를 유발한 사고방식으로는 그 문제를 절대로 해결할 수 없다."

Content


Research Projects and Future Internet Studies
I3
I5
I4
I2
I6
I7
I5
I2
I8
I5
I8
Future Internet
E5&8
E2&8
E5
E6
E2
E5
E7
Design and Setup of the Experimental Facility
The Vision
The Lab
Monitoring Framework
Control Framework
Federation
ZOO
Future Internet
New Ideas

G-Lab: Vision of the Future Internet

- Closing the loop between ***research*** and ***real-world experiments***
- Provide an **experimental facility** for studies on architectures, mechanisms, protocols and applications towards Future Internet
- Investigate interdependency of theoretical studies and **prototype development**



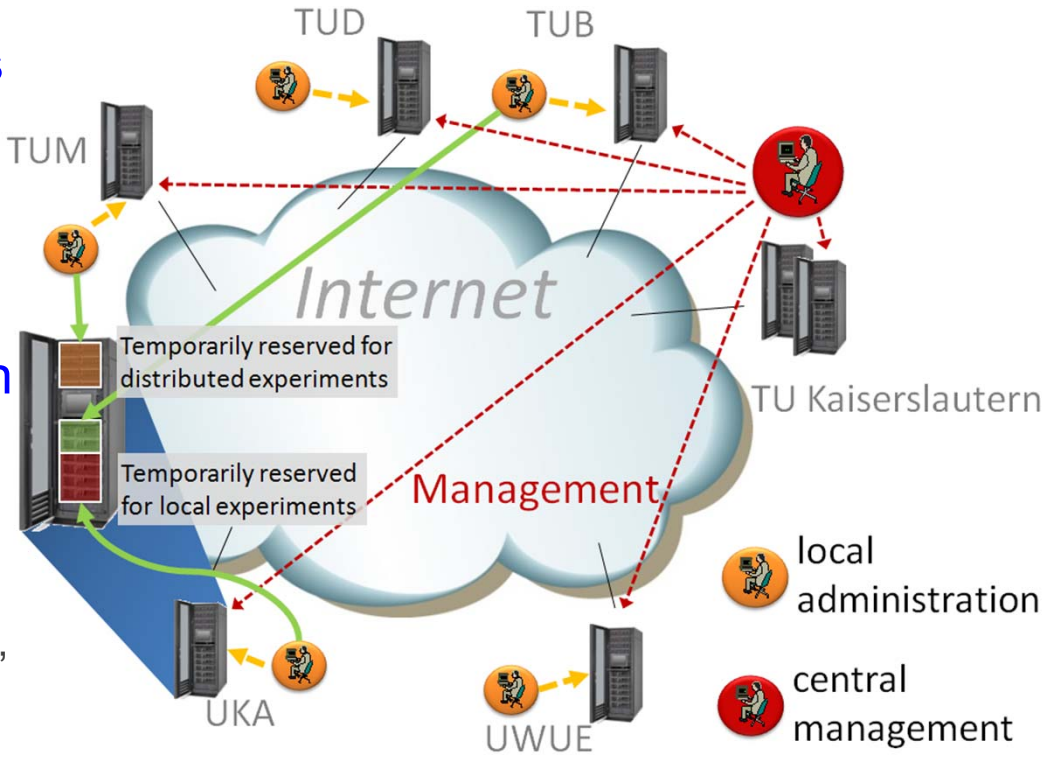
G-Lab Environment

- **Testbed:**
  - Real not simulated
  - Specific purpose
  - Focused goal
  - Known success criteria
  - Limited scale

  Not sufficient for clean slate design
- **Experimental facility:**
  - Purpose:
    - **explore yet unknown architectures**
    - **expose researchers to real thing**
    - **breakable infrastructure**
  - Larger scale (global?)
  - Success criteria: unknown

The Lab

- Full control over the resources
  - Reservation of single resource should be possible
  - Elimination of side effects
  - Testing scalability
- Exclusive resource reservation
  - Testing QoS / QoE
  - Decentralized Resources can be independently used
  - Tests on the lower layers of the network without affecting the “life” network
- Extended functionality
  - New technologies (Wireless, Sensor,…)
  - Interfaces to other testbeds (GENI, PlanetLab Japan, WinLab, …)



TUB	TU Berlin
TUD	TU Darmstadt
TUKL	TU Kaiserslautern
TUM	TU München
UKA	University Karlsruhe KIT
UWUE	University Wurzburg

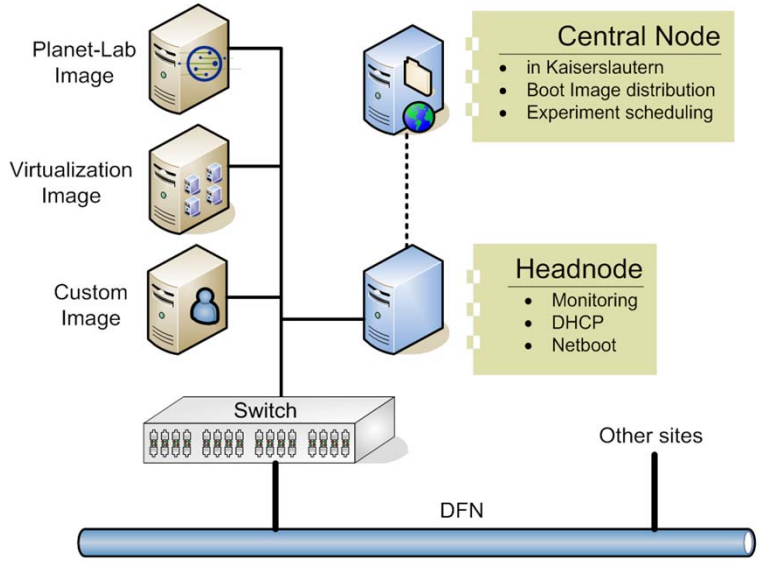
Hardware Equipment

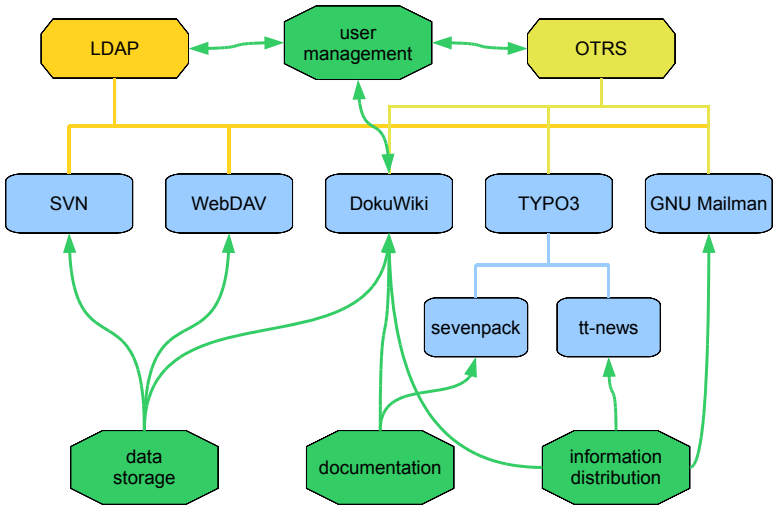
- Normal Node
  - 2x Intel L5420 Quad Core 2,5 GHz
  - 16 GB Ram, 4x 146 GB disk
  - 4x Gbit-LAN
  - ILOM Management Interface (separate LAN)
- Network Node
  - 4x extra Gbit-Lan
- Headnode
  - 2x Intel E5450 Quad Core 3,0 GHz
  - 12x 146 GB disk
- Switch Fabric CISCO 45xx
- Site requirements
  - 1 public IP address per Node
    - IPv4 and/or IPv6 addresses.
    - Virtualized nodes need additional addresses
  - Direct Internet access
    - No firewall or NAT
    - Nodes must be able to use public services (NTP, public software repositories)
  - Dedicated Links
    - dark fiber, λ wavelength, MPLS

- 174 Nodes in total (1392 cores total)

Site	Head	Network	Normal	
Kaiserslautern	1	2	47+9	Phase I
Würzburg	1	2	22	
Karlsruhe	1	2	22	
München	1	2	22	
Darmstadt	1	2	22	
Berlin	1	2	12	
Passau	1	2	2	Phase II
Hannover	1		1	
Hamburg	1		1	
Lübeck	1			
Stuttgart	1			
Total	185			

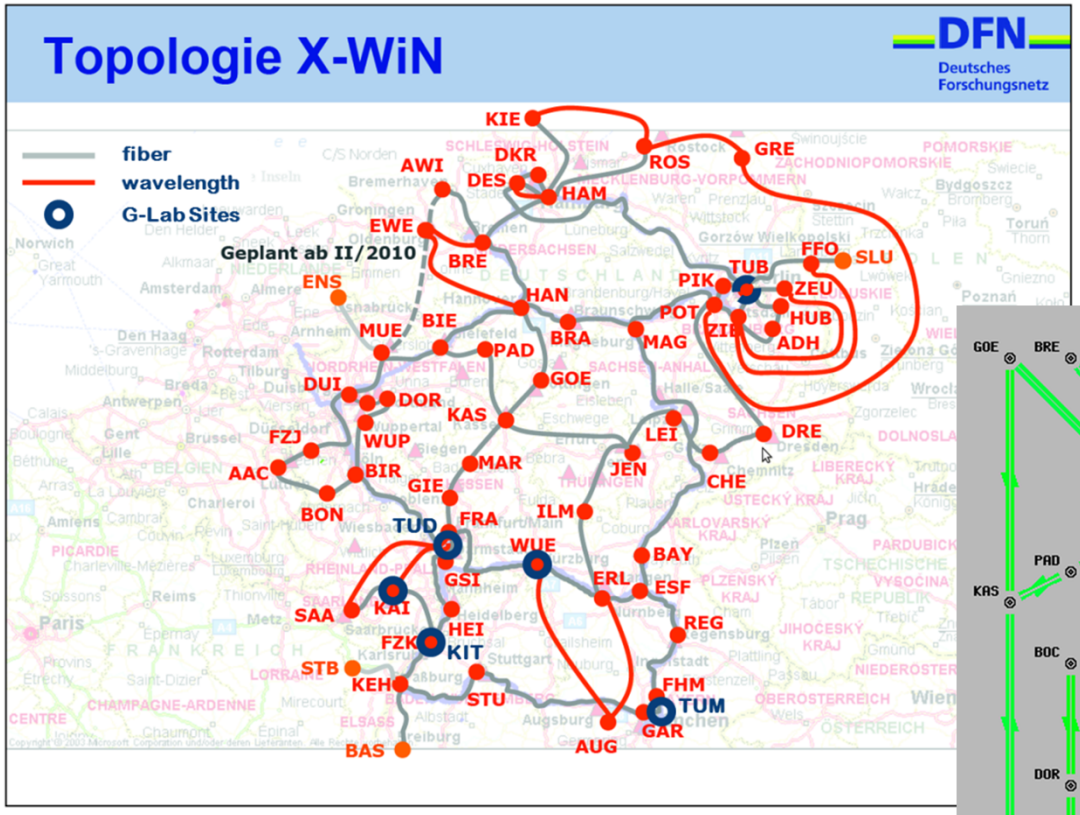
G-Lab Structure





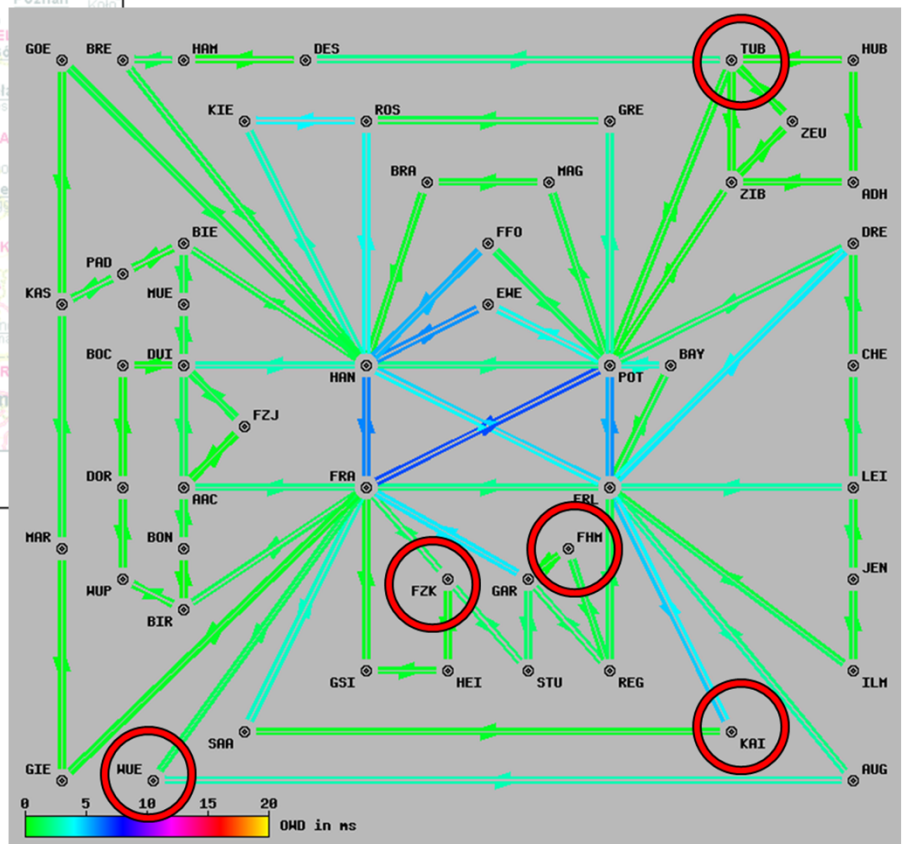
- Central Node
  - Resource management
    - Experiment scheduling
    - Resource provisioning
  - Boot Image management
    - Distributes Images
    - Assigns Images to nodes
- Each site has a Headnode
  - Manages local nodes
    - DHCP
    - Netboot
    - Monitoring
    - ILOM access
  - Executes orders from Central node
    - Local overrides possible
- G-Lab Central Services
  - Overall user management
  - Not an open platform
  - Trouble ticket system (OTRS)
  - WiKi, data storage, …
  - Based on TYPO3 (CMS)

G-Lab Network Topology



Physical Topology

IP Topology



Flexibility

- Experimental Facility is part of research experiments
  - Facility can be modified to fit the experiments needs
  - Researchers can run experiments that might break the facility
    - Experimental facility instead of a testbed
- Research is not limited by
  - Current software setup
  - Current hardware setup
  - Restrictive policies

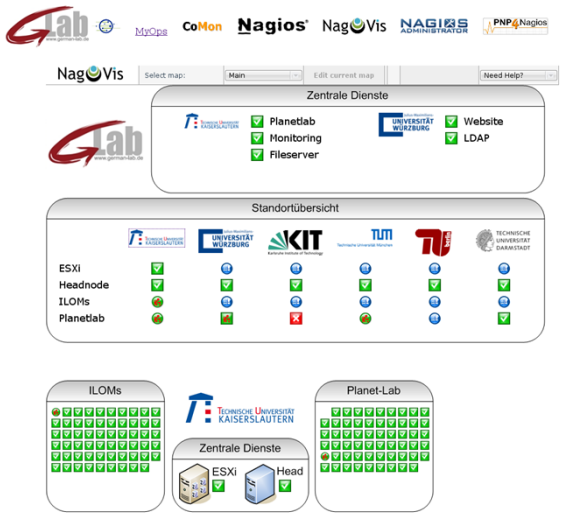
Mobility
Energy Efficiency
Sensornetworks
…

- Experimental Facility is evolving
  - Cooperative approach
    - „When you need it, build it“
    - Core team helps
  - Cooperation with other facilities (e.g. Planet-Lab, GENI, …)
  - Sustainability (as a non profit organization) / Federation

G-Lab Monitoring Framework

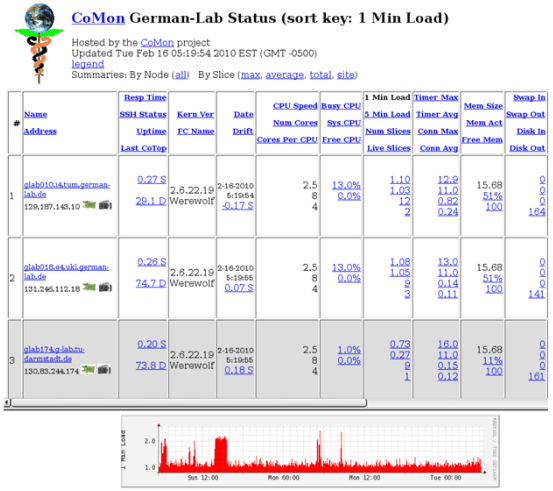
Nagios

- Central monitoring in Kaiserslautern
- Obtain information from other sites via NRPE proxy on the head-node
- Checks
  - Availability of Nodes
  - Status of special services
  - Hardware status (via ILOM)
- http://nagios.german-lab.de



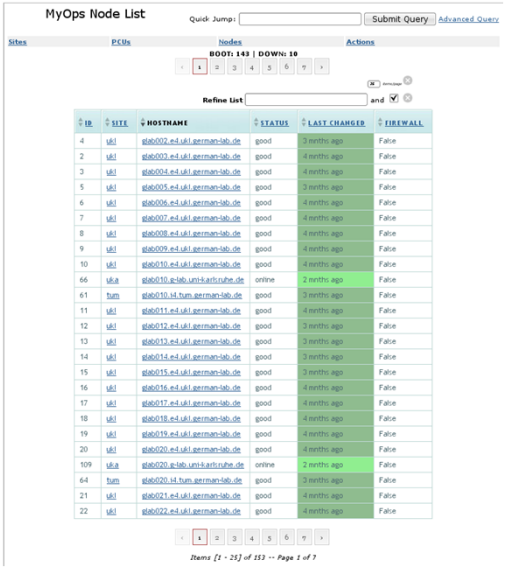
CoMon

- Planet-Lab specific monitoring
  - In cooperation with Planet-Lab, Princeton
- Monitors nodes from within
  - CPU, Memory, IO
- Slice centric view
  - Monitors experiments
- http://comon.cs.princeton.edu/status/index_glab.html

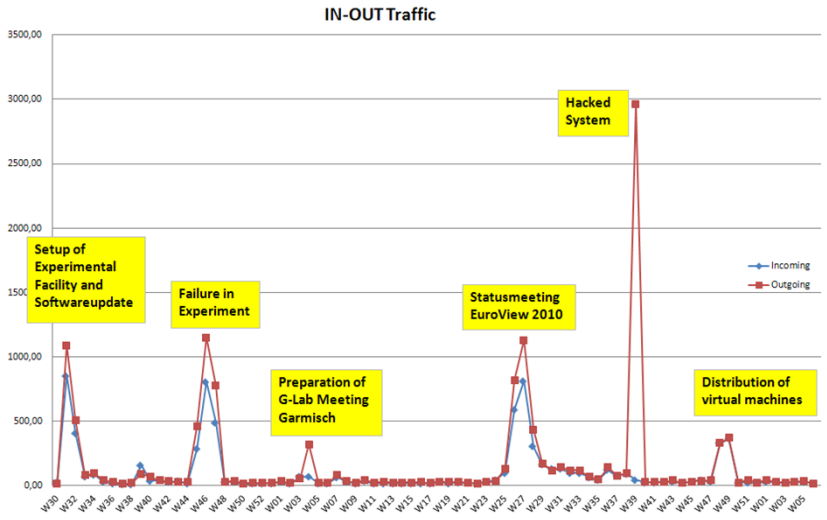


G-Lab Monitoring Framework

- MyOps
  - Planet-Lab specific tool
    - In cooperation with Planet-Lab, Princeton
  - Detects common Planet-Lab problems
  - Reacts to problems

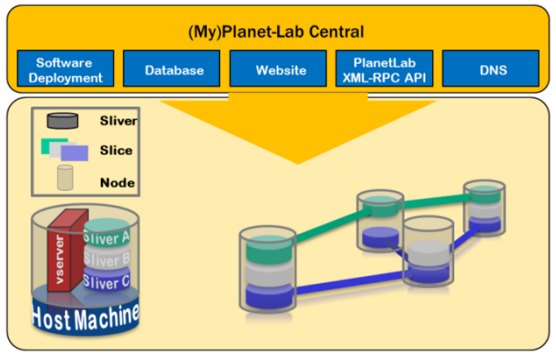


- In/Out Network traffic
  - Based on DFN connectivity
  - Important to control the lab at runtime to avoid interference with operational systems
  - Traffic patterns can be stored and related to the experiments
    - Quality assurance of the experiments
  - Further developments
    - MPLS or wavelength links



Control Framework

- **Planet-Lab**
  - Easy management of testbed-„silce“
  - Lightweight virtualization
  - Flat network
  - Rich tool support (monitoring, experiment control)



- **ToMaTo**
  - Topology-oriented
  - Multiple virtualization options
  - Virtualized and emulated networks

ToMaTo

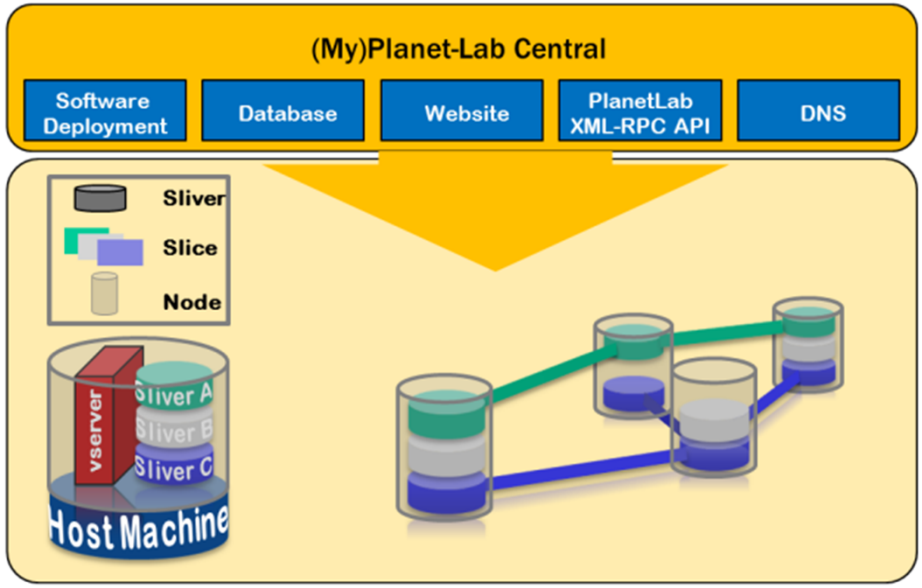
- **Seattle** (coming soon)
  - For algorithm testing
  - Executes code in custom python dialect
  - Federated with GENI Seattle

Seattle
Open peer-to-peer computing

- **Custom Boot-Images**
  - Software comes as boot image
  - Either booted directly on hardware or in virtualization

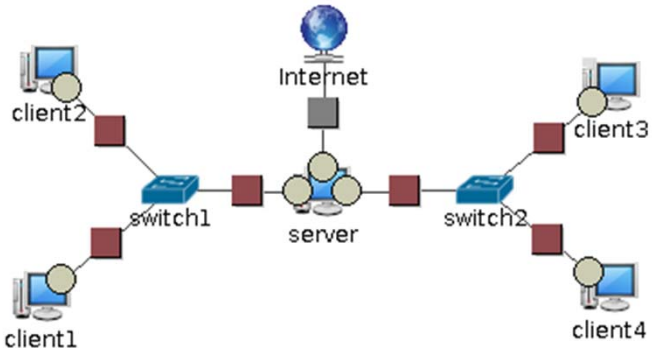
Planet-Lab Structure

- Planet-Lab
  - Testbed and software by Princeton
  - Only Software is used
  - Extended in Cooperation with Princeton
- Uses Virtualization
  - Provides virtual node access called „Sliver“
  - Slivers across several nodes form a „Slice“
- Central configuration
  - Planet-Lab Central (PLC) in Kaiserslautern
  - User management
  - Sliver management



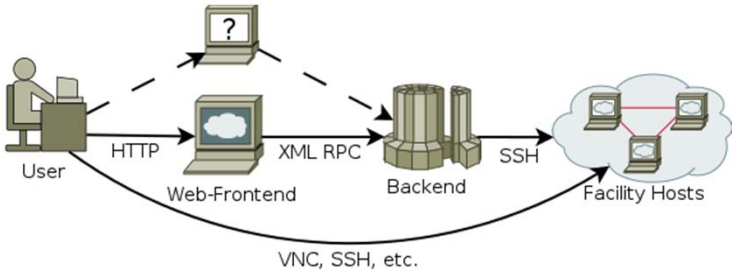
ToMaTo - A network experimentation tool

- ToMaTo „Topology Management Tool“



- Topology contains
  - Devices
    - Active components
    - E.g. computers
    - Produce/Consume data
  - Connectors
    - Networking components
    - E.g. switches, routers
    - Transport/Manipulate data

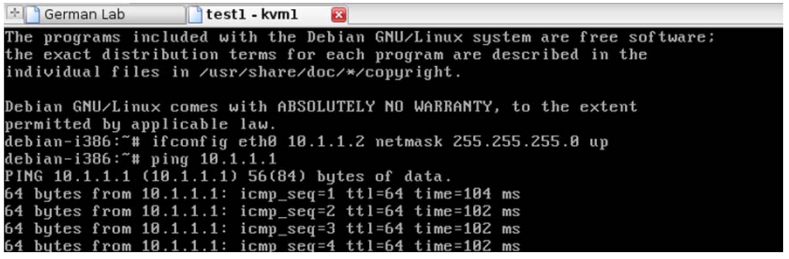
- ToMaTo contains of 3 parts

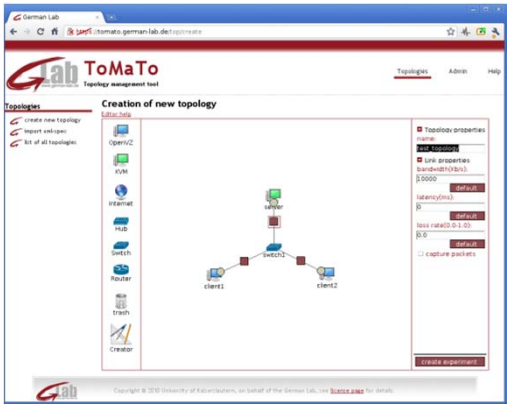


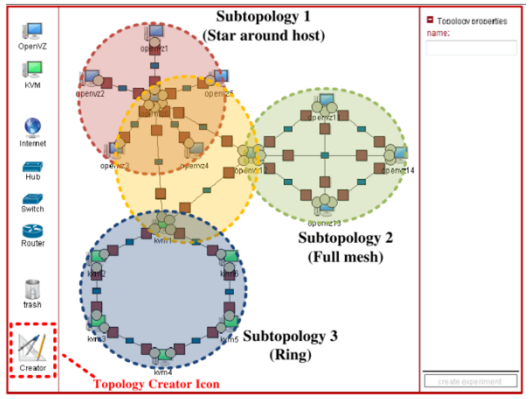
- Host part
  - Based on PROXMOX VE
  - Offers virtualization
  - Additional software available as packages
- Backend
  - Controls hosts via SSH
  - Centralized logic, resource management, user accounts
  - Offers XML RPC interface
- Frontend(s)
  - Offer a GUI to users
  - Currently only a web-based interface exists

ToMaTo – Features and editor

- Administrator/Developer features
  - Intelligent load-balancing
  - Open xml-rpc interface
  - Administrator tools
  - LDAP integration
- User features
  - Automatic network interface configuration
  - Changes to running topologies
  - Console access
  - Image up/download
  - Pcap capturing (packet capturing)
- ToMaTo graphical editor
  - Automatically creates topologies
  - Ring-, Star- and Full mesh topologies
  - Connects topologies
- Configures network interfaces
  - IP addresses
  - Netmasks
- DEMO Video gives a short introduction





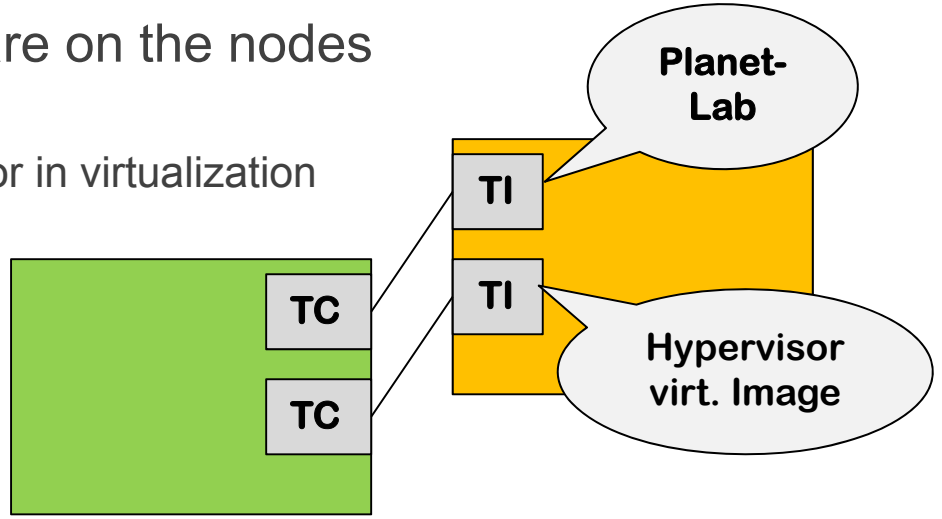


ToMaTo Application Area

- Access layer experiments
  - Consider lower layers and hardware
    - Example: Mobile handover
  - Requirements
    - Hardware access
    - Custom operating systems (Realtime)
    - Heterogeneous access technologies (3G, Wifi, etc.)
  - Needs specialized **testbeds** depending on hardware NO ToMaTo support
    - DES Testbed, Wisebed
- Network layer experiments
  - Focus on TCP/IP suite
    - Example: IPv6 extensions, TCP substitutes
  - Requirements
    - Deep OS access (modified kernels, etc.)
    - Small but complex topologies, link emulation
  - **ToMaTo offers**
    - Full kernel access via KVM
    - Complex topologies
    - Link emulation
    - Packet capturing (for analysis)
    - Easy setup of topologies
- Algorithm/Protocol experiments
  - Work on top of network layer
    - Example: P2P-Networks
  - Requirements
    - Huge but simple topologies
    - Link emulation
    - No hardware or OS access
  - **ToMaTo offers**
    - Lightweight virtualization with OpenVZ
    - Link emulation
    - Federation with other testbeds via Internet
- Legacy software experiments
  - Considers legacy software
    - „Legacy software“ refers to any widespread software with undocumented or unpublished behavior
    - Example: Skype and Windows
  - Requirements
    - Special environments, custom operating systems
    - Small but complex topologies
    - Link emulation and external packet capturing
  - **ToMaTo offers**
    - Custom operating systems with KVM (Windows
    - Access to external service via Internet connector
    - Packet capturing independent of guest OS

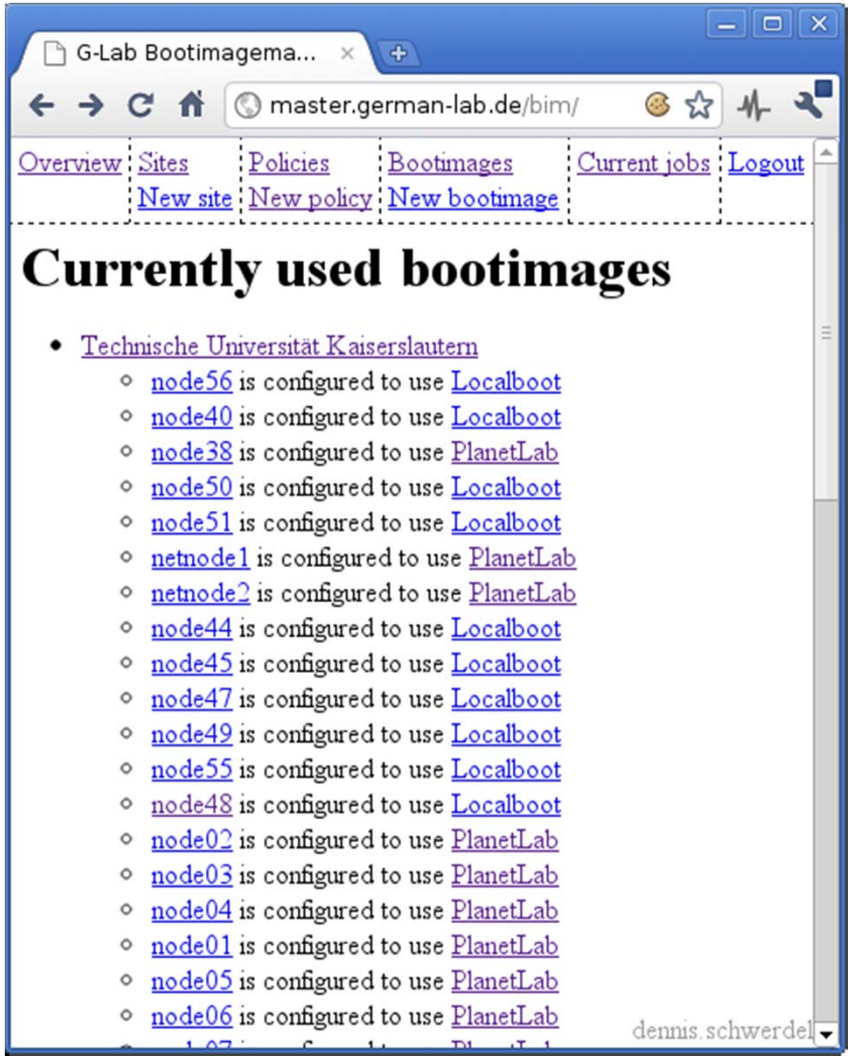
Boot Images

- Researchers can run any software on the nodes
  - Software comes as boot image
  - Either booted directly on hardware or in virtualization
- Three types of boot image
  1. Planet-Lab
     - Access for everybody
     - Easy to manage
     - Restricted hardware access
  2. Hypervisor virtualization image
     - Access for everybody
     - Unrestricted access to virtual hardware
     - Topology management via ToMaTo
  3. Custom boot image
     - Access can be restricted to specific research group
     - Unrestricted access to real hardware
- Access regulated by policy
  - Favors generic images with open access over specific images with restricted access
  - Policy does not over-regulate



Boot Image Management (BIM)

- Central component
  - Status view
  - Node and site management
  - Boot image management
    - Upload boot images to file server
  - Boot image assignment
  - Access control
  - Logging
- Headnode component
  - Fetch boot images from file server
  - Create PXE config from configuration
  - Node power control
- Central component controls headnode component

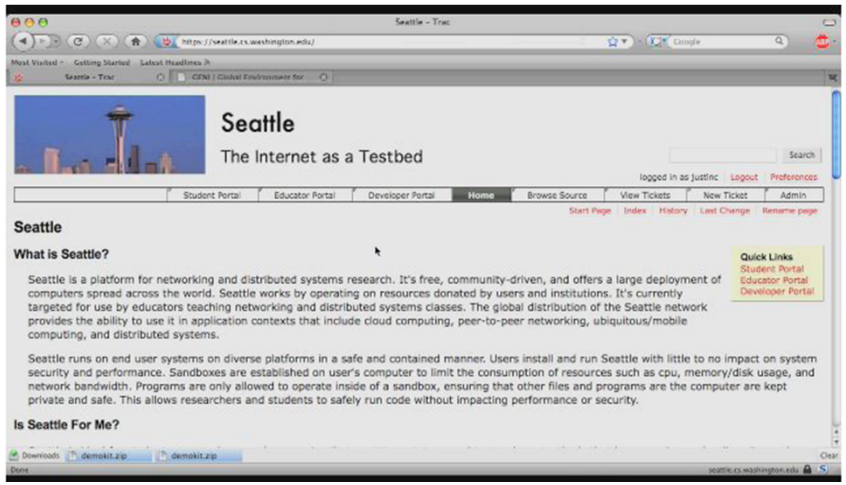


Seattle

Open peer-to-peer computing

- Testbed for python code
- Very lightweight, no virtualization, just sandbox
- Very comfortable experiment control
- Fully federated with Seattle GENI (over 1000 nodes)
- Wide variety of network types accessible
  - Sensors
  - Cell phones
  - Mobile nodes
- Coming soon in G-Lab, early tests running
- Algorithm testing
  - https://seattle.cs.washington.edu
  - Developed by Justin Cappos (University of Washington)

- Demo Video
  - https://seattle.cs.washington.edu/wiki/DemoVideo



- This five-minute demo video should help get you acquainted with the Seattle project.

Why Federation

Zoo

- Controlled environment
  - Host systems
  - Network
  - Users

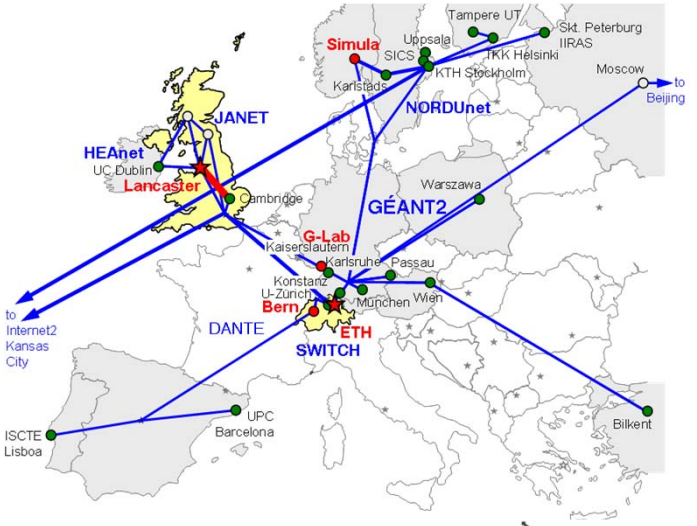
- Controlled environment for
  - development, deployment and testing of new algorithms
  - Breakable infrastructure
- Repeatable experiments
  - When dealing with new algorithms for routing, security, mobility, …
  - Improve scientific quality

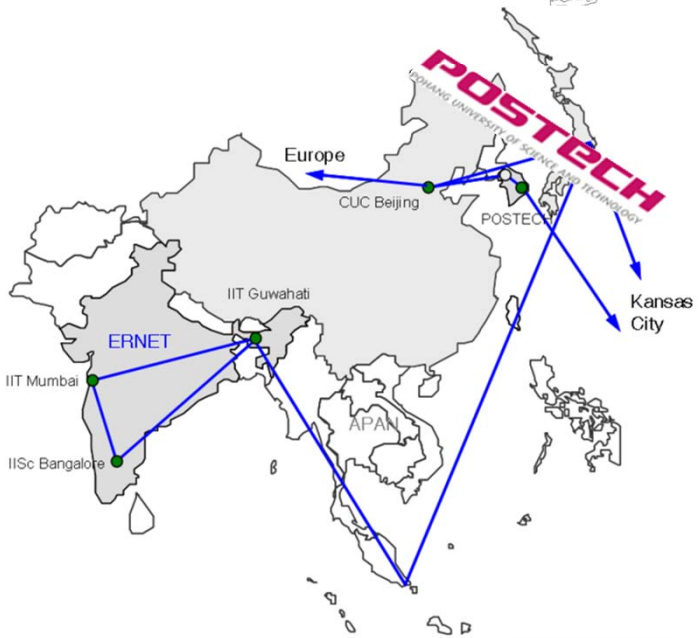
Wilderness

- Scalability
  - Hoe does new algorithms behave in the wilderness?

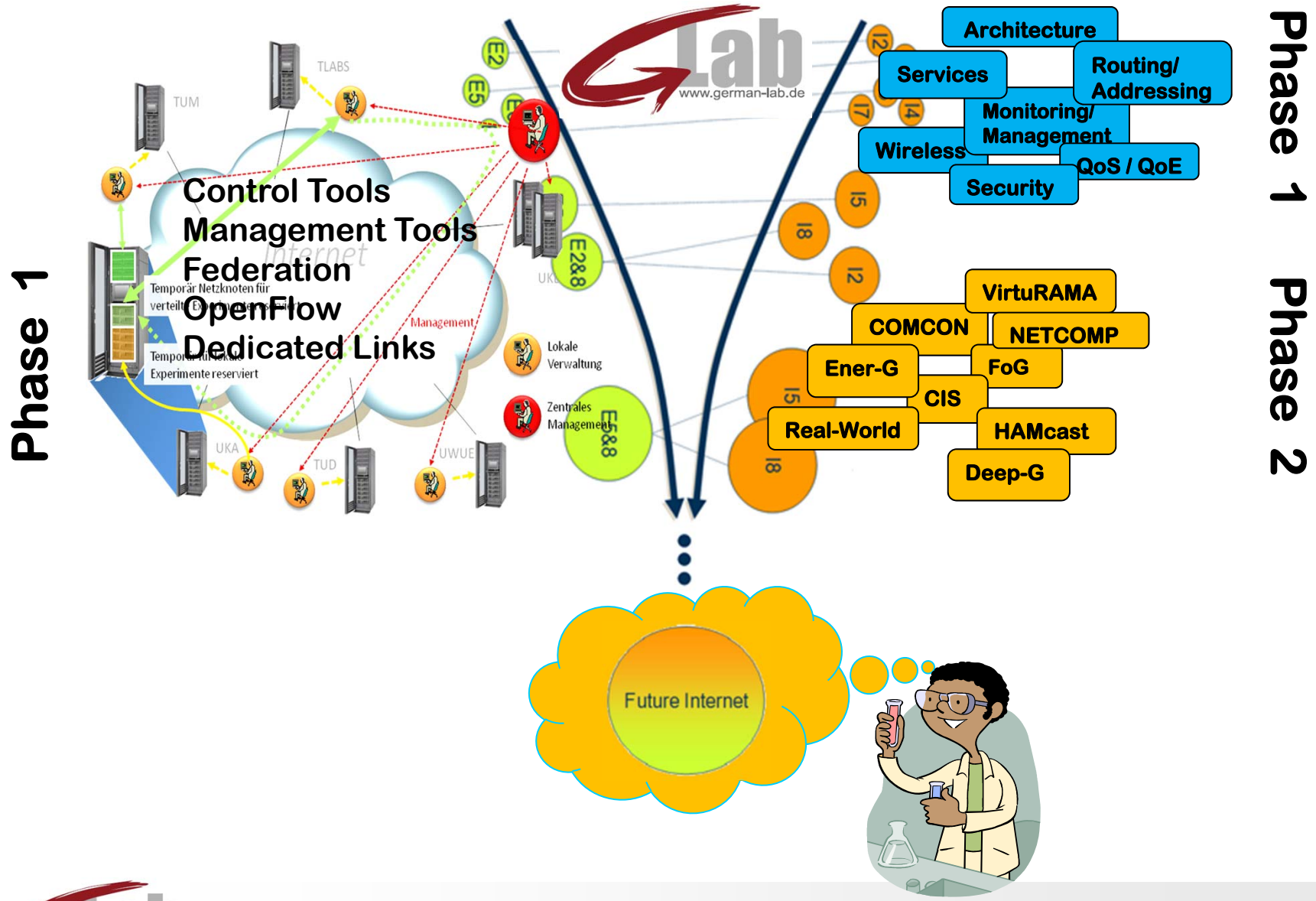
Federations

- GpENI *„Great Plains Environment for Network Innovation”*
  - US-based network testbed
  - Kaiserslautern is fan-out location for central European sites
  - Connection to G-Lab possible
- GpENI Asian flows use L2TPv3 and IP tunnels over **Internet2** to **APAN** (Asia-Pacific Advanced Network), which interconnects Asian regional and national research networks.
  - In Korea, **POSTECH** (Pohang University of Science and Technology) is connected to GpENI (**J. Won-Ki Hong)**
- GENI Federation
  - GENI connection by 1Gbit/s link from Starlink/Geant/DFN for GEC10





Conclusion


Phase 1
Control Tools
Management Tools
Federation
OpenFlow
Dedicated Links
TLABS
TUM
UKA
TUD
UWUE
Management
Lokale Verwaltung
Zentrales Management
E2
E5
E2&8
E5&8
Architecture
Services
Routing/ Addressing
Monitoring/ Management
Wireless
QoS / QoE
Security
VirtuRAMA
COMCON
NETCOMP
Ener-G
FoG
CIS
Real-World
HAMcast
Deep-G
Future Internet
Phase 1
Phase 2

- Questions ?