



Fraunhofer Institute for Open Communication Systems | Kaiserin-Augusta-Allee 31 | 10589 Berlin, Germany



Getting started with Teagle - A FIRE testbed federation tool

Sebastian Wahle, Thomas Magedanz & Konrad Campowsky
Fraunhofer Institute FOKUS | Technische Universität Berlin
sebastian.wahle@fokus.fraunhofer.de

www.fokus.fraunhofer.de/go/ngn2fi

www.av.tu-berlin.de



Agenda

- Part 01: Introduction ~ 40 min
10 Minutes Break
- Part 02: Panlab Concept & Architecture ~ 40 min
10 Minutes Break
- Part 03: Use Case ~ 30 min
- Part 04: VCT Tool videos & demo ~ 20 min
10 Minutes Break
- Part 05: How to join, how to commit resources ~ 20 min

About the Presenters



Sebastian Wahle leads the Evolving Infrastructure and Services group at NGNI within the Fraunhofer FOKUS institute in Berlin. The group is active in a number of national and international R&D projects in the Future Internet field and supports the commercial Fraunhofer NGN testbed deployments at customer's premises worldwide. The group's research activities are centered around large scale infrastructure federation, cross-layer monitoring and management, as well as cloud computing for NGNs. Sebastian received a Diploma-Engineer degree in Industrial Engineering and Management from the Technical University of Berlin. His personal research interests include Resource Federation Frameworks and Service Oriented Architectures.

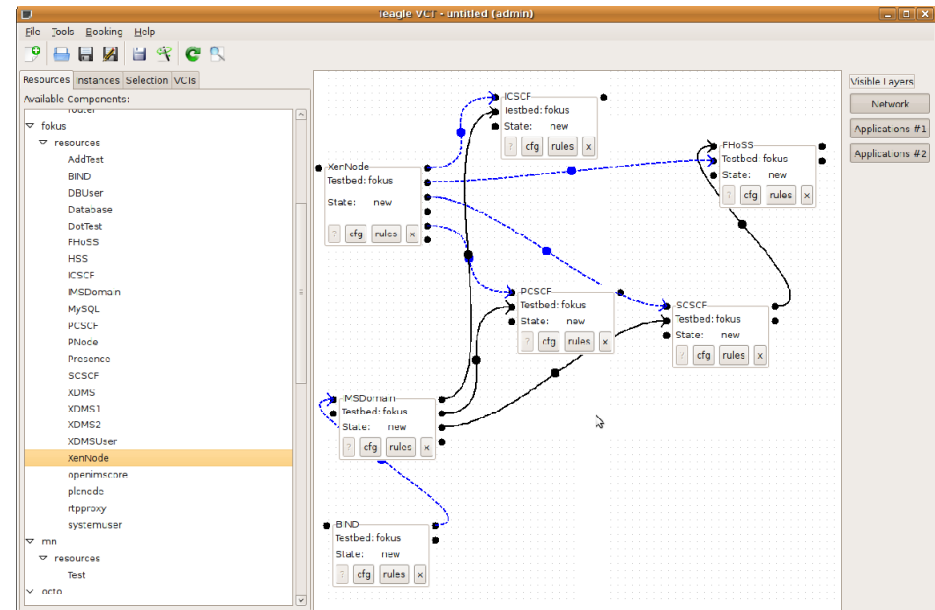
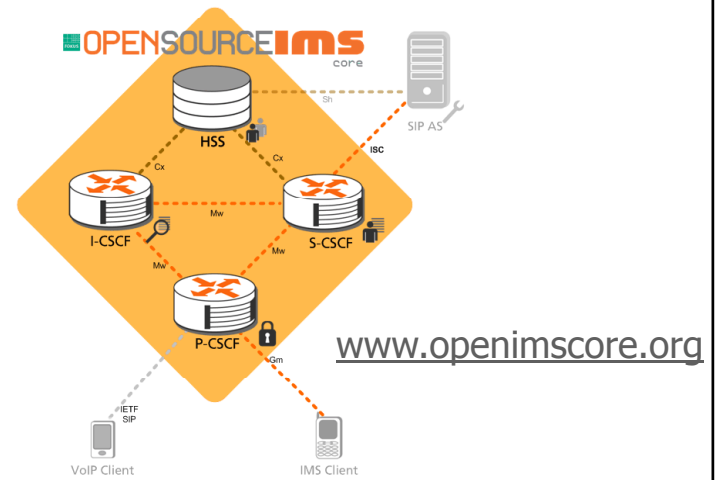
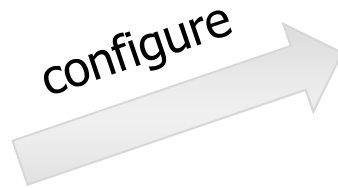


Konrad Campowsky has joined the AV (Architektur der Vermittlungsknoten) department at Technische Universität Berlin in 2010, where he is working as a researcher in the fields of testbed management and network domain federation. In collaboration with the Fraunhofer FOKUS Institute, his current work focuses on extending the testbed federation concept towards service and application levels and automating the process of composite, cross-domain service creation and management.

The Demo



www.fire-teagle.org



A custom testbed is designed with Teagle making use of infrastructure resources (hardware, virtual machines) and service layer resources (Open Source IMS core)



The Experiment

SIPNuke

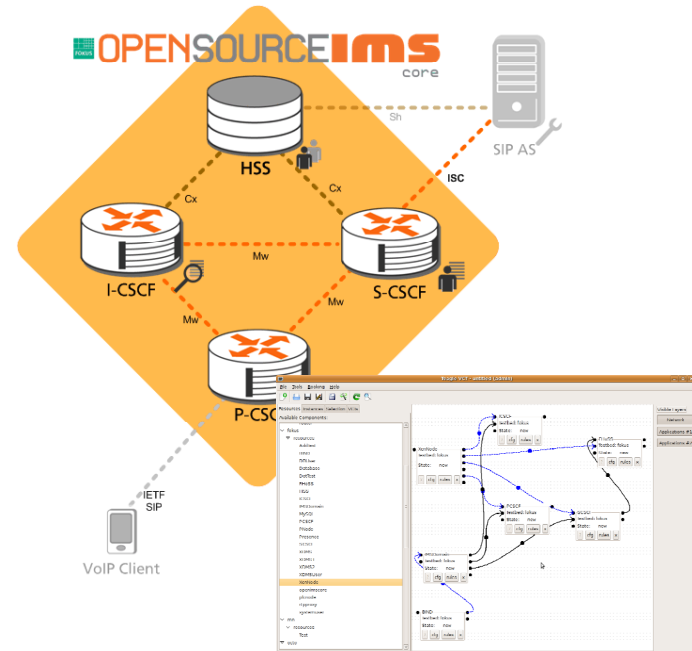
Load generator
www.sipnuke.org

stress the core ...



... and break it

www.openimscore.org



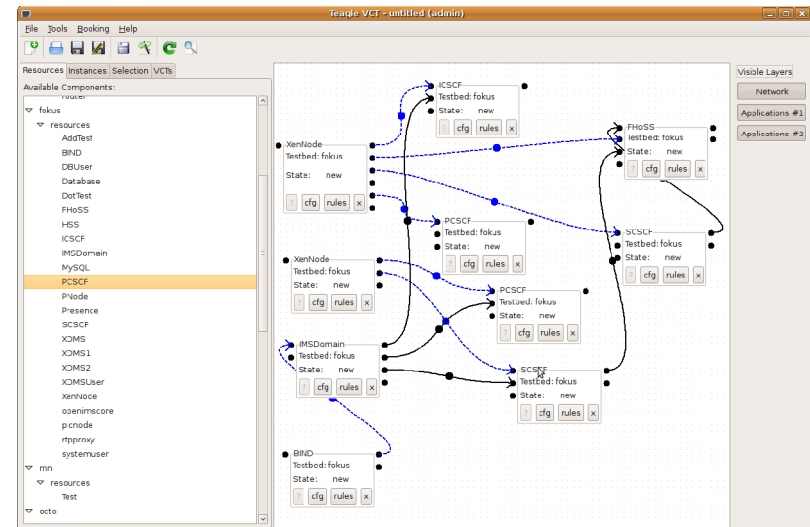
- * 50 IMS USER IDS
- * 5 REGISTER per second
- * increasing from 50 to 400 CALLS per second, looping through all users
- * UNREGISTER all users



Second experiment execution



modify testbed



... re-execute the experiment

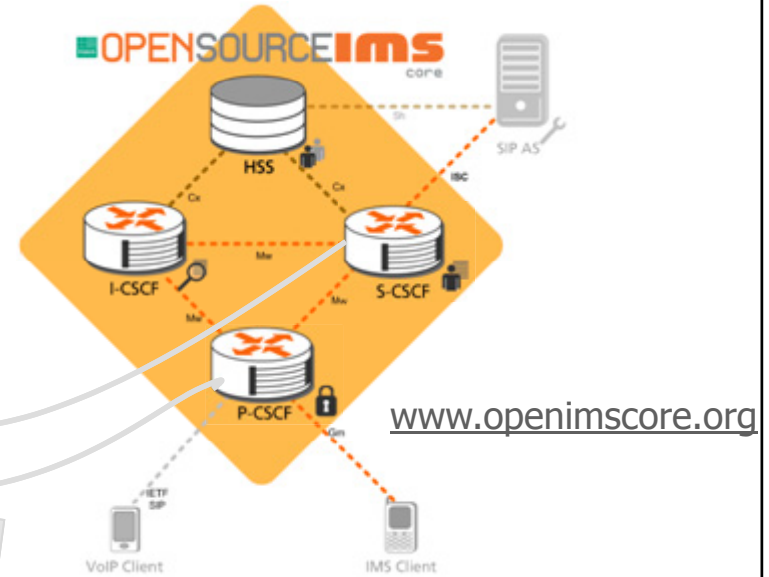


Modify the custom testbed

384 MB RAM
1 CPU



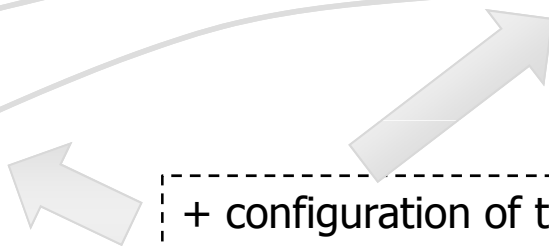
virtual machine
hosting the core



compute
resource
network
resource
storage
resource



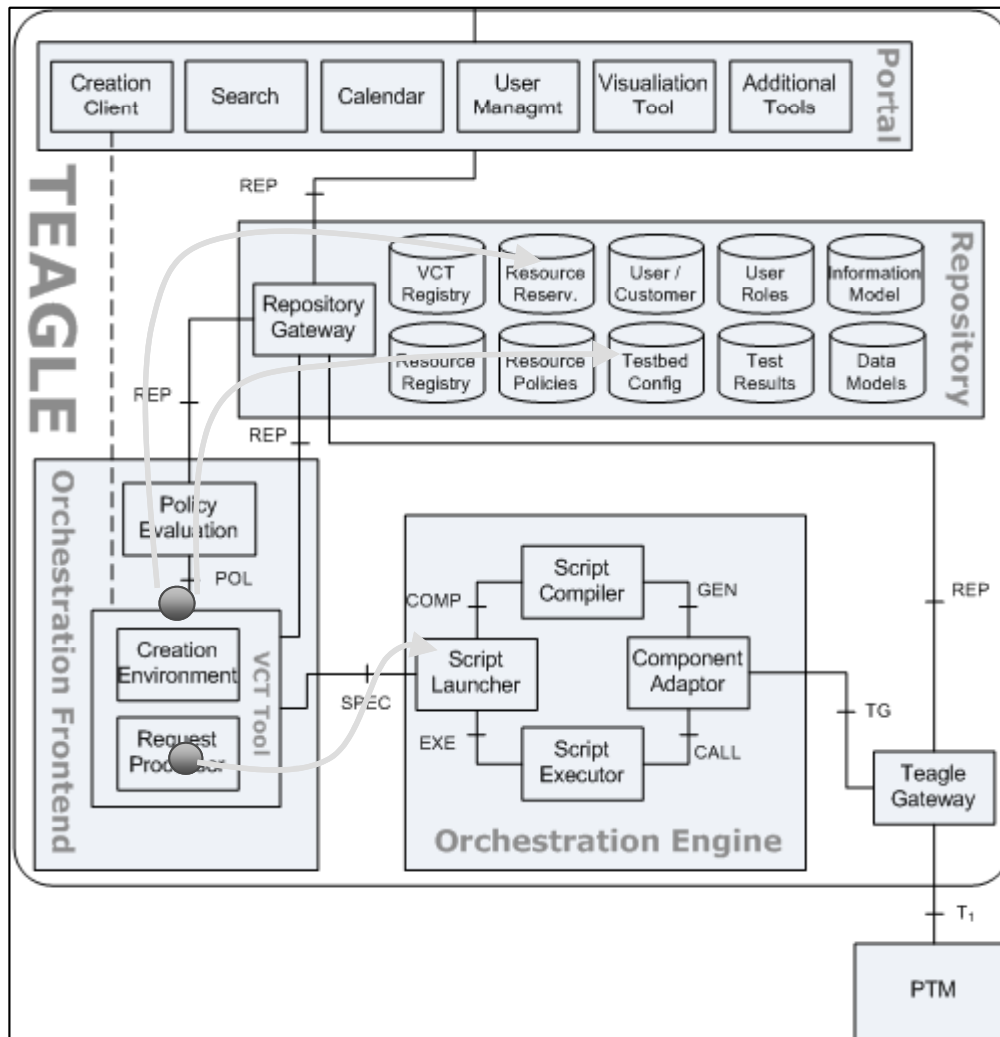
Cloud Resource



- + configuration of the new P-CSCF and S-CSCF that have been moved to more powerful cloud resources
- + re-configuration of the old core to deal with the new instances
- + re-configuration of DNS server, etc.



Behind the scenes



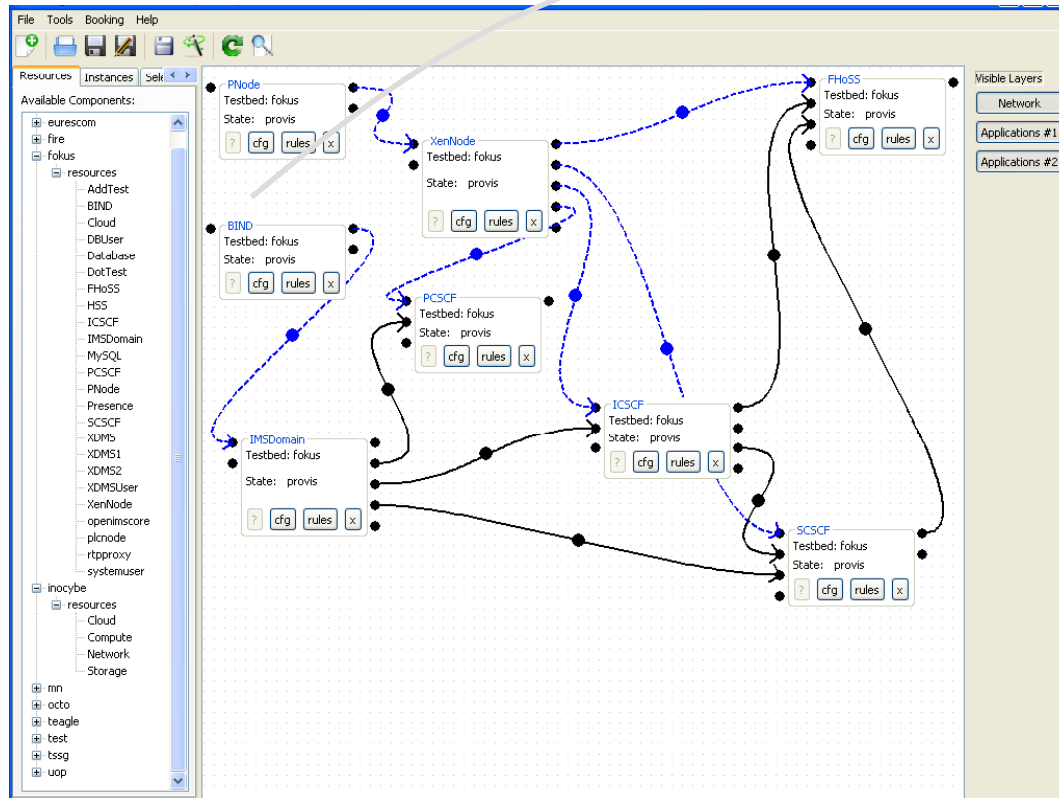
The Teagle tool helps reconfiguring the testbed and deploying new resource instances.

From the design environment requests are send to the repository to update resource configurations and reservations.

The topology-oriented testbed design is send to the Orchestration Engine by means of a XML document.

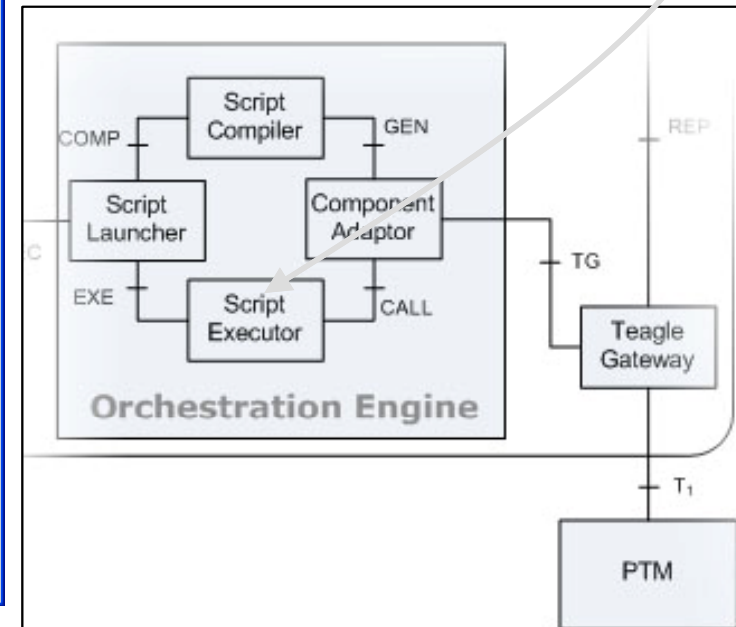


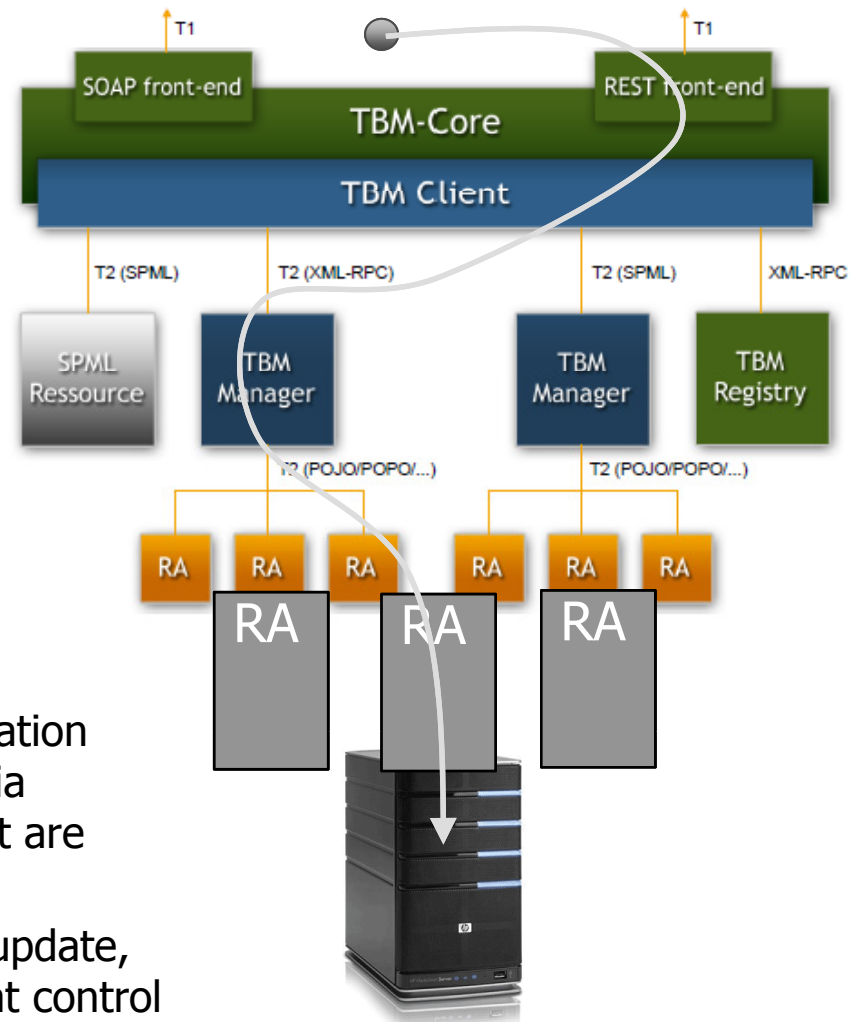
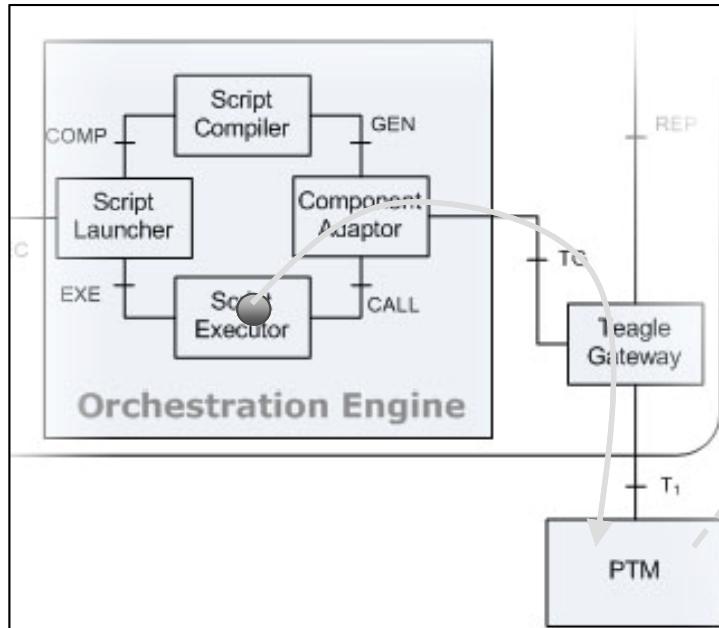
The Orchestration Engine transforms the design environment output into an executable script and executes it (ordering of provisioning requests, resolving temporary IDs, etc.)



```
class HSSProvisioning01 (VBEEntity):
  ## meta information
  META = {
    'orchestrate': {'args': ('userid',)},
  }

  def __init__(self,SESSION):
    self.SESSION = SESSION
    self.appld = PLUGINCONF.PLUGINID
    self.spatelsystem = SpatelSystem(
      SESSION,self.appld,'HSSProvisioning01')
  ## *** operation orchestrate ***
  def orchestrate(self,userid):
    ## in userid:String -> String
    from voicebench.comm.VariantManager import invokeVariant
    return invokeVariant(self,'orchestrate',userid)
  def orchestrate_v0(self,userid):
    ## in userid:String -> String
    ## use this for fake implementation
    result = "" ## default result
    return result
```



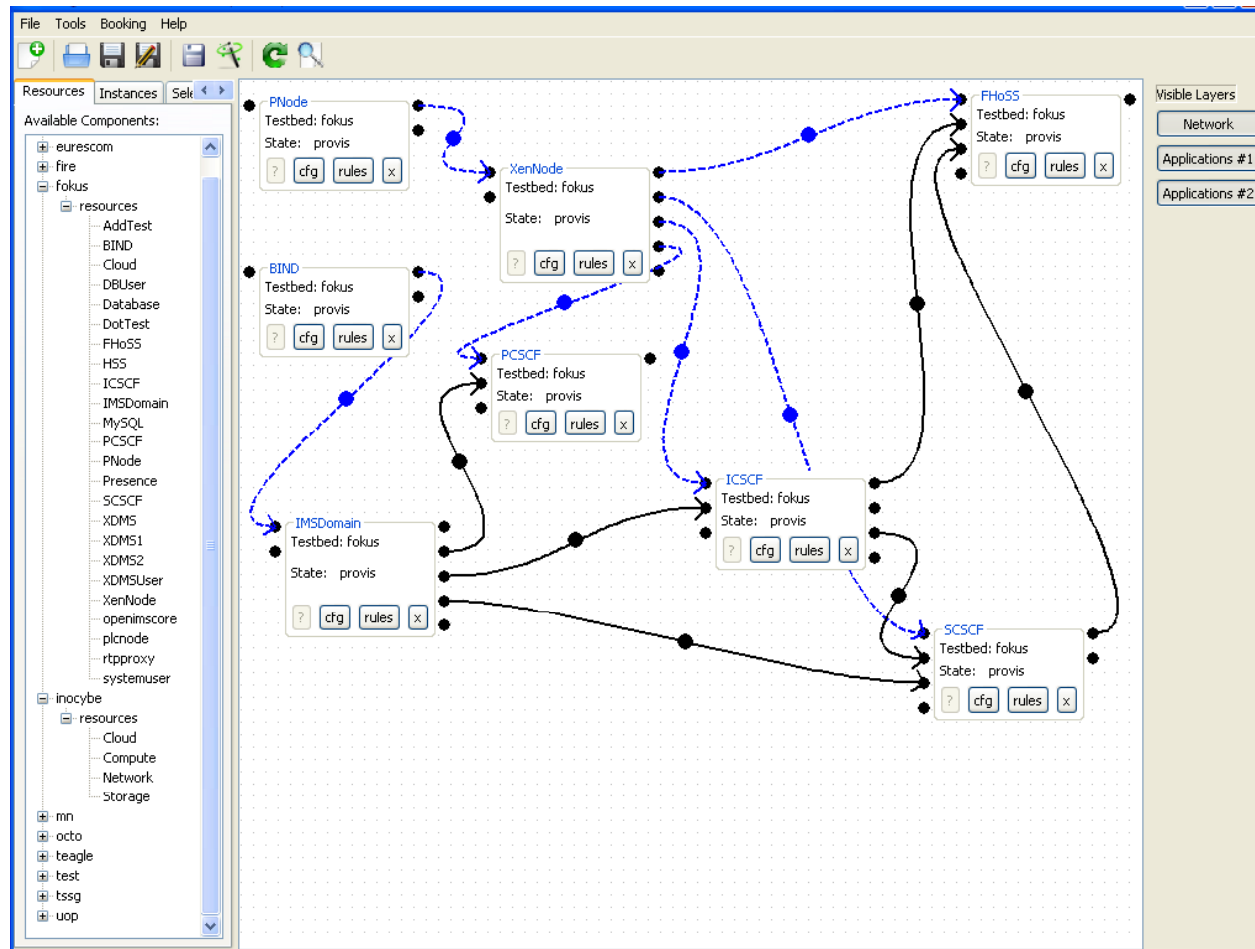


From the script executor provided by the Orchestration Engine, generic REST requests (CRUD) are sent via interface T1 to those domain managers (PTM) that are responsible for involved resources.

The PTM passes the generic CRUD (create, read, update, delete) requests to the resource adaptors (RA) that control the involved resources. Here, the generic commands are translated into resources specific requests and actions.



Some screenshots from the demo screen cast



Initial testbed design using the Teagle creation environment. The dotted lines reflect a containment relationship (e.g. the ICSCF is hosted by the XENnode). The solid lines represent a configuration reference.

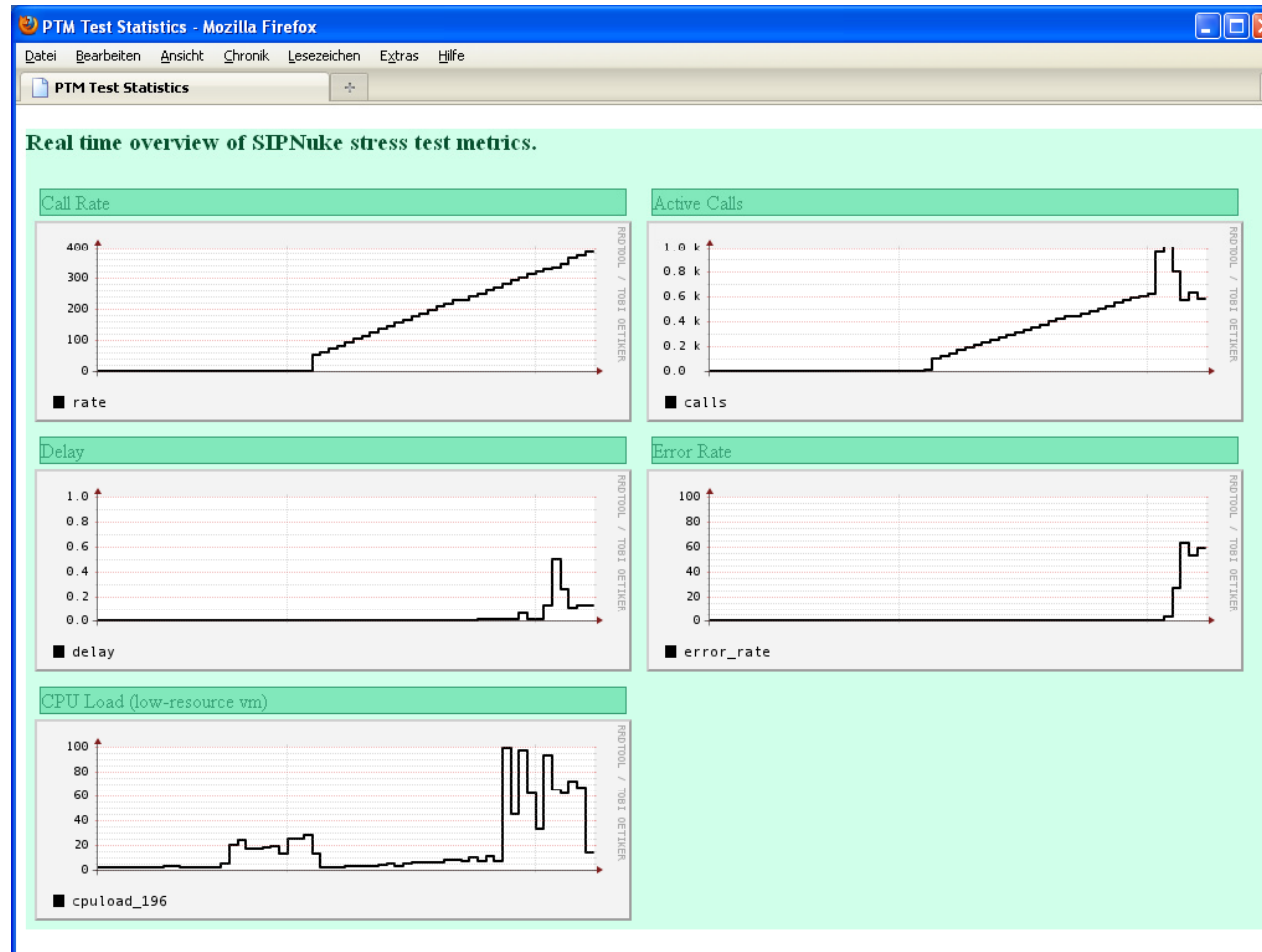


Demo Video

- Start video



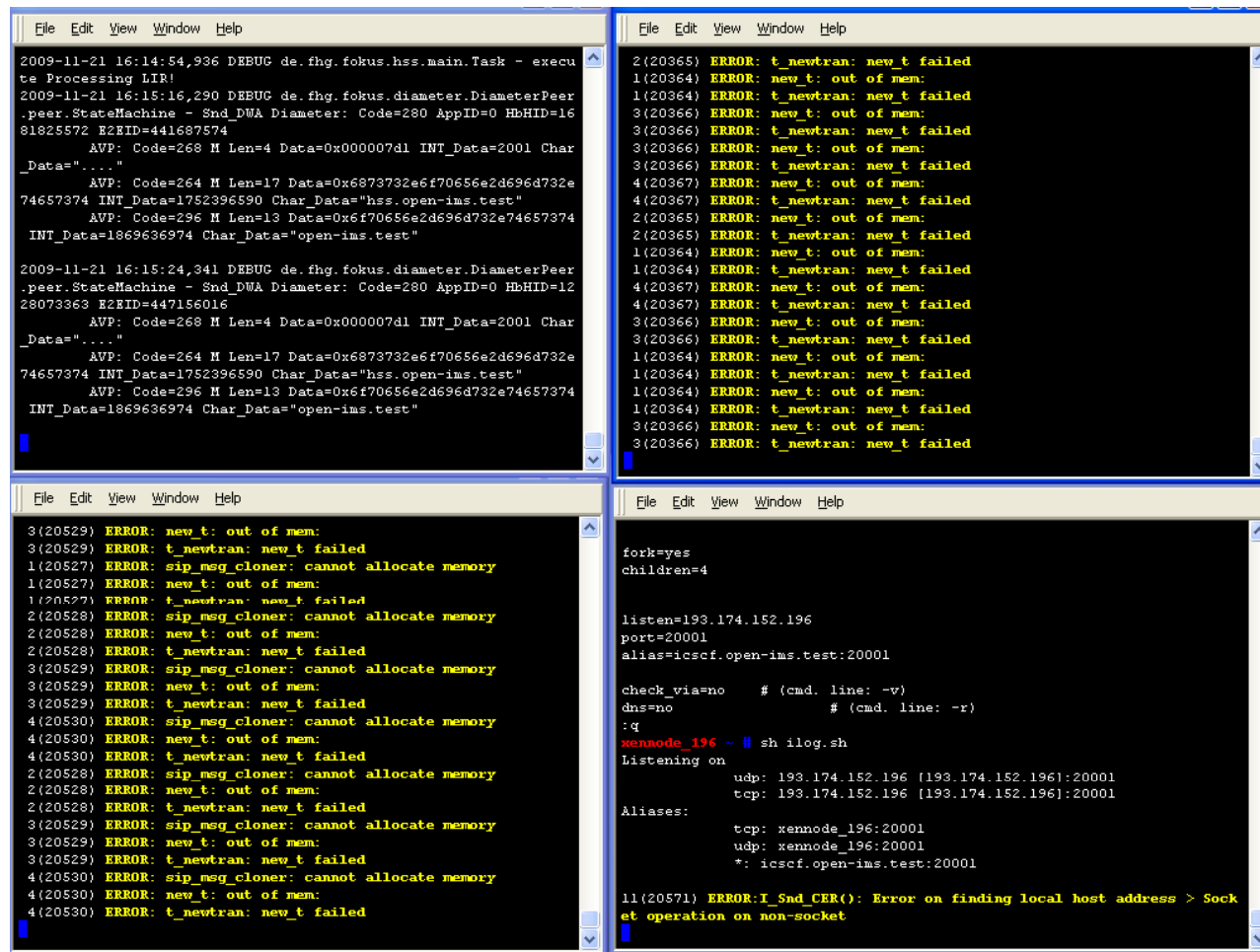
Some screenshots from the demo screen cast



Execution of the first experiment. The CPU load of the small XEN reaches 100%, additionally the machine runs out of memory. Therefore, delays and high call error rates are observed.



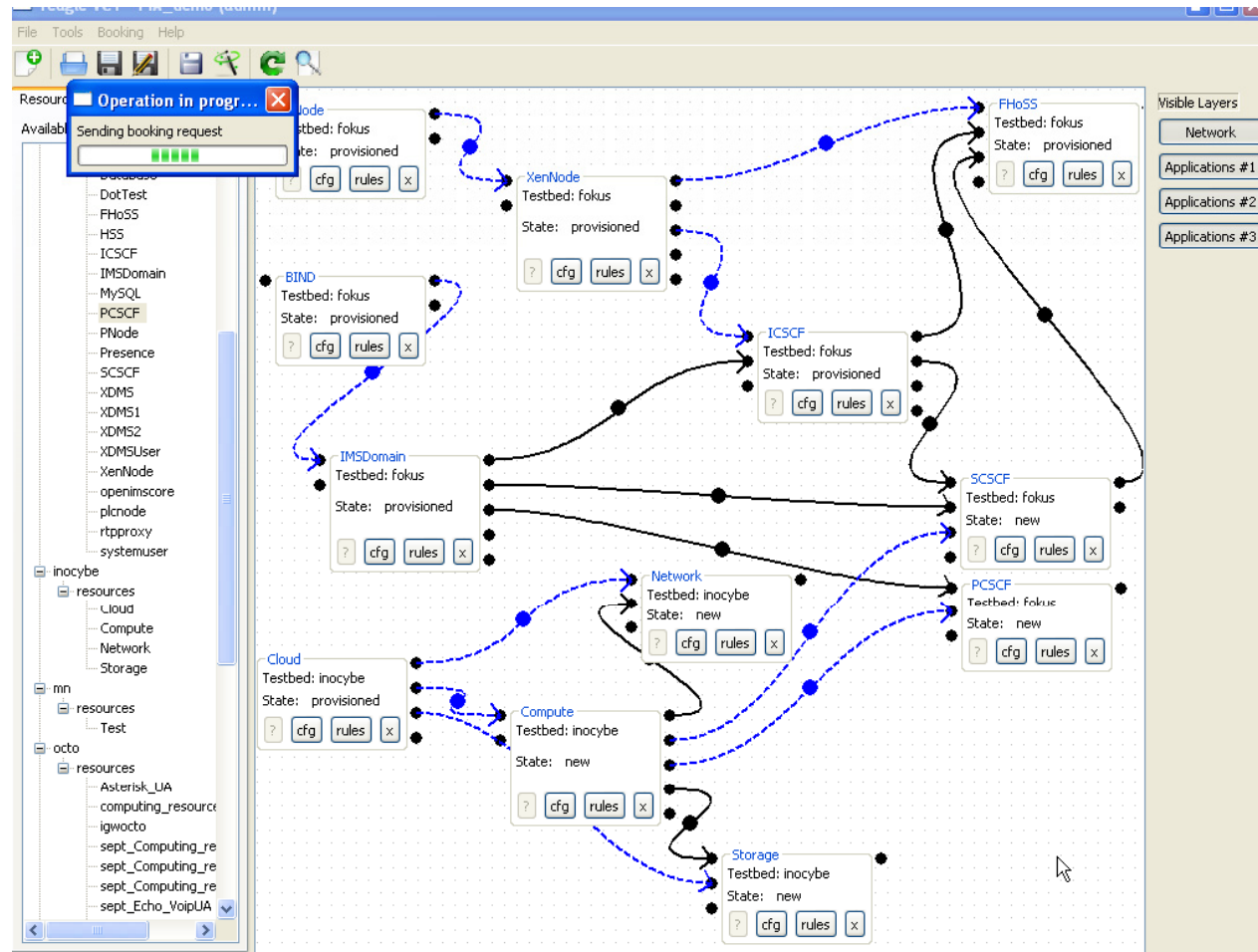
Some screenshots from the demo screen cast



The IMS Core logs show out of memory errors. At this stage the core is unstable and high delays and error rates are observed.



Some screenshots from the demo screen cast



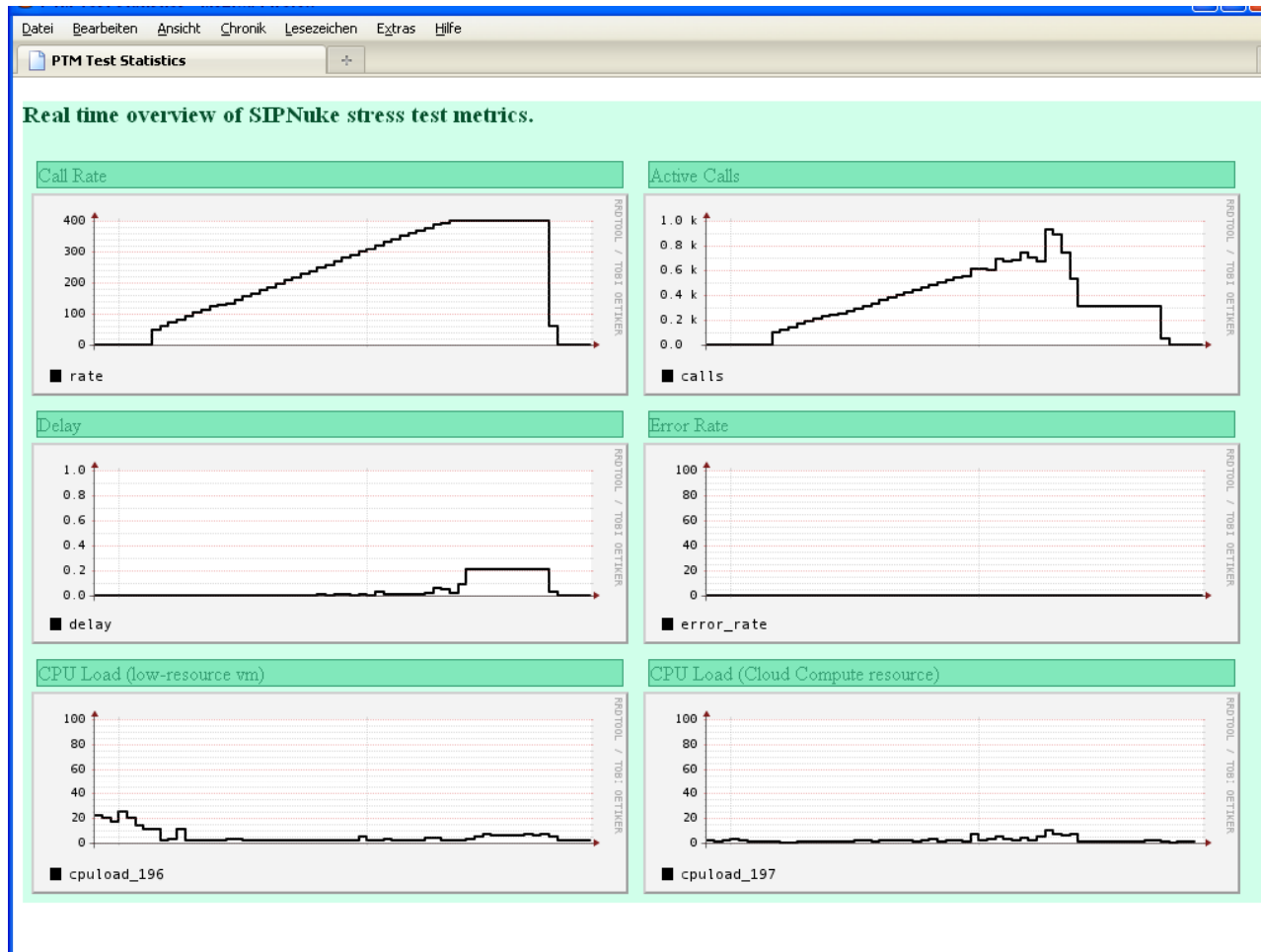
The testbed after the re-design.

The IMS Core services PCSCF and SCSCF have been moved to cloud resources using specific resource adaptors that implement the OCCI standard.

The resources that are in *state: new* are additionally booked into the testbed by sending created and update commands to the involved domains.



Some screenshots from the demo screen cast



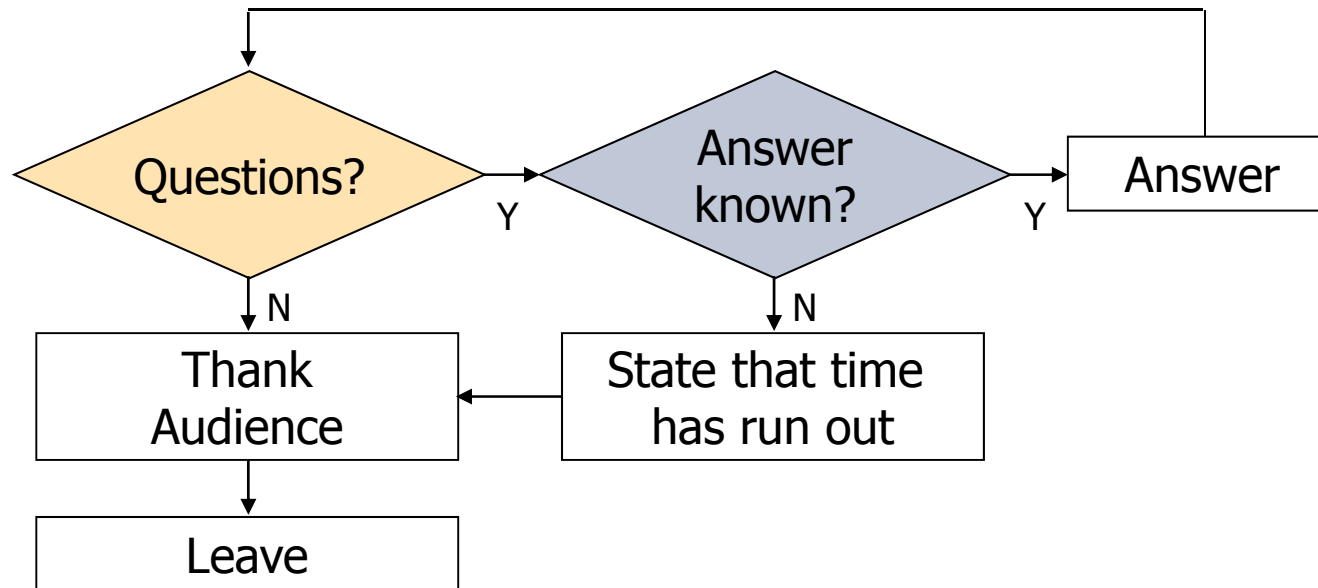
The metrics show a better performance now, using more powerful resources. 400 calls per second could be handled easily after modifying the testbed setup.

No errors and no higher delays than 0.2ms.

The CPU load on the computing resource (4 CPUs, 4GB mem) offered by the cloud stays below 20%. However, cross-site networking might introduce additional constraints. More experiments could be performed to analyze this in detail when moving services across clouds.



Thanks for your attention!



Selected References

- www.fokus.fraunhofer.de/go/ngn2fi
- www.panlab.net
- www.fire-teagle.org

- Sebastian Wahle, Thomas Magedanz, and Anastasius Gavras. Towards the Future Internet - Emerging Trends from European Research, chapter Conceptual Design and Use Cases for a FIRE Resource Federation Framework, pages 51-62. IOS Press, April 2010. ISBN: 978-1-60750-538-9 (print), 978-1-60750-539-6 (online).
<http://www.booksonline.iospress.nl/Content/View.aspx?piid=16471>

- Konrad Campowsky, Thomas Magedanz, and Sebastian Wahle. Resource Management in Large Scale Experimental Facilities: Technical Approach to Federate Panlab and PlanetLab. In 12th IEEE/IFIP Network Operations and Management Symposium (NOMS 2010). IEEE/IFIP, April 2010.

