

**“Testbeds as a Service”  
Building Future Networks  
A view into a new GN3Plus Service**

Jerry Sobieski (NORDUnet)  
TERENA Architecture Workshop  
Nov 13, 2013

# From Innovation to Infrastructure



- Network Innovation requires testing to prove out...
  - Testing in live networks can have unintended effects on non-combatants. Other users and network providers don't like being **crash test dummies**.
  - “Production” environments have the required scale but are highly risk averse.



**How do we evolve innovations from concept to production with minimal risk to infrastructure, services, and applications already in place providing on-going stable and reliable services?**



- The Network Research community needs networking research “Laboratories” in which to develop novel concepts ...
  - Constructed from stable underlying infrastructure
  - Allow high risk experiments to be carried out...
  - Yet prevent unexpected or errant behaviour from interfering with production services or other testing.
  - Provide reliable and effective work environment for the researcher
  - Enable a broad range of innovation – not focused on one particular idea – indeed architecturally technology agnostic
  - Fast: Ability to rapidly prototype new ideas
  - Flexible: The test kit/prototype and/or the test regimen can be easily modified based on analysis of the test data
  - Scalable: Ability to construct large scale test environments
  - These laboratories must be able to duplicate real world scenarios such that research results are useful and valid
- We need “network” crash test dummies...

## SA2: “Testbeds as a Service”

# GN3+SA2 “Testbeds as a Service”



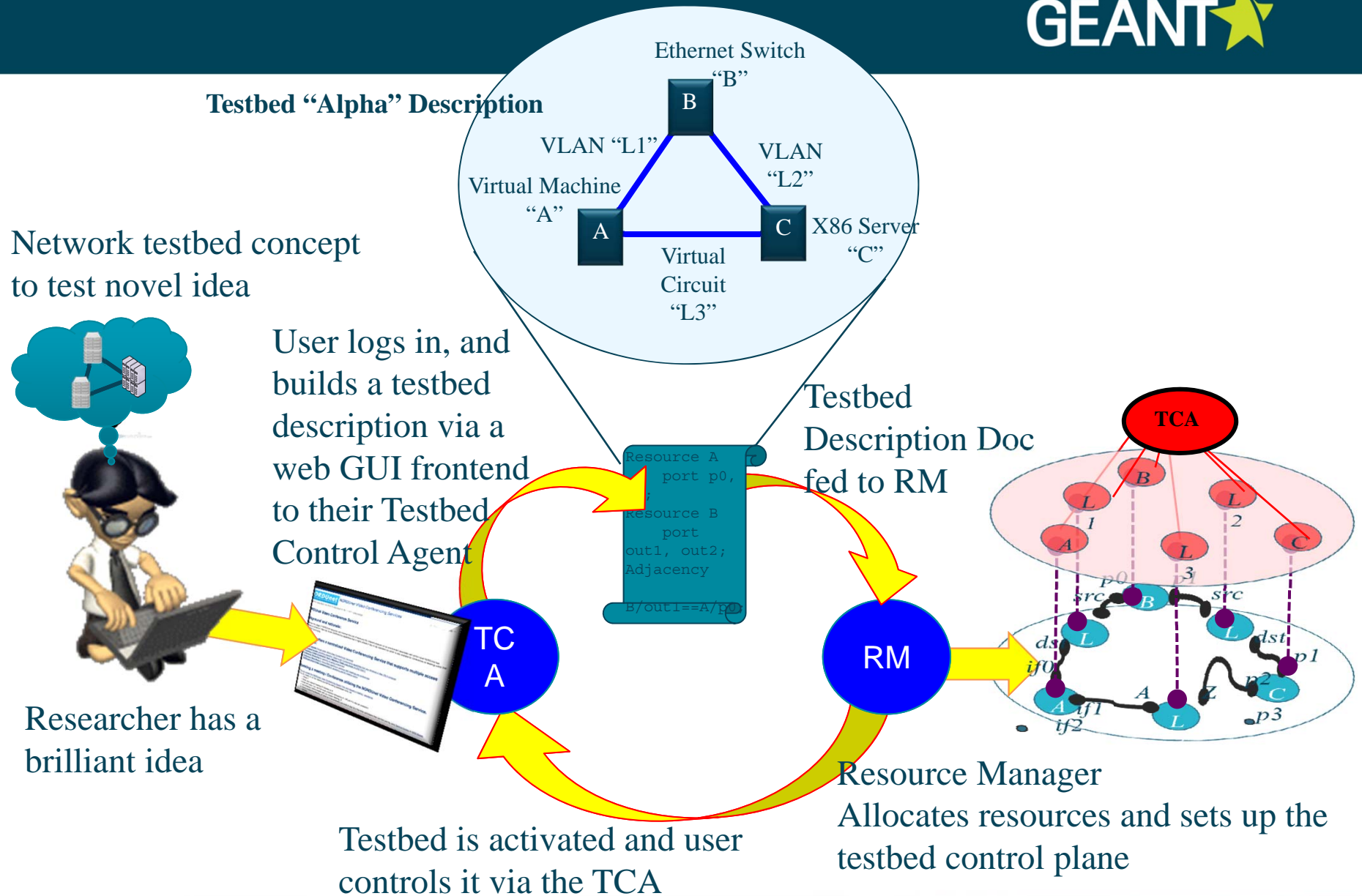
## ● SA2 will deliver two key Testbed Services:

- **Dynamic “Packet” Testbeds** – dynamically allocated, virtualized networks provisioned over packet oriented transport and switching infrastructure with a pan-European footprint.
  - *Under control of the researcher*
  - *Insulated to prevent collateral damage to other services*
  - *Flexible network topologies that can morph as necessary*
  - *Extensible to support novel hardware*
- **“Dark Fiber” Testbeds** – photonic testbeds over dark/dim fiber along long haul routes between a limited set of major EU metro areas.
  - *Virtualization of these resources is hard...but we'll see...*

## ● SA2 is a GEANT Production Service

- The test beds it creates are expected to be reliable and available.
- Which means the SA2 management and control processes/protocols must be stable and secure
- The virtualized resources must be such that they can be effectively isolated to contain rabid behaviour.
- This integrated “multi-species” virtualization represents new technology and continues to evolve in the community ... There continues to be many research efforts, and many emerging frameworks and service models... So there is no precedent for SA2 – no configuration guide, no BCPs, ...

# Dynamic Packet Testbed- How it works

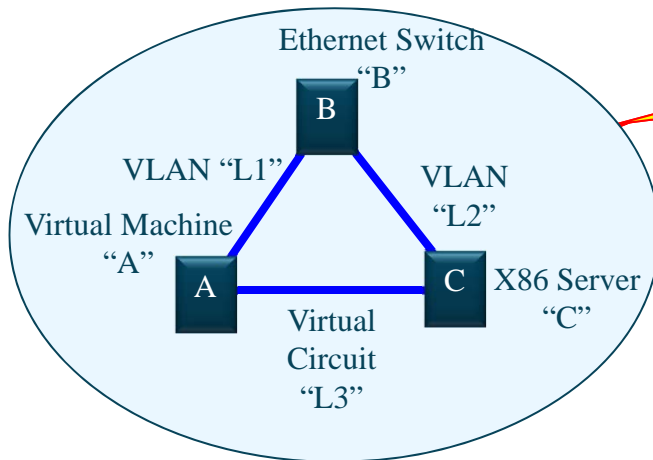


# A Brief Dive into the Internals:

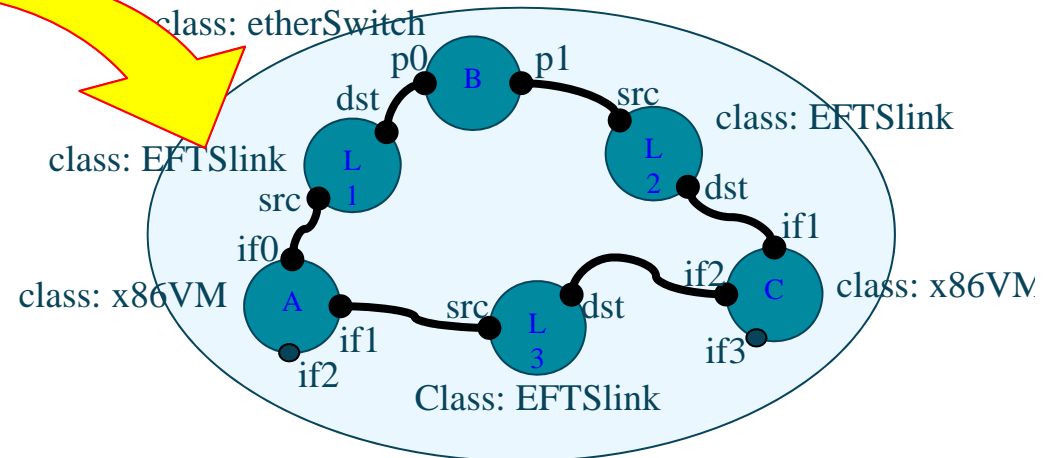


The TaaS Architecture treats all [testbed] networks as graphs

## Testbed “Alpha” Description



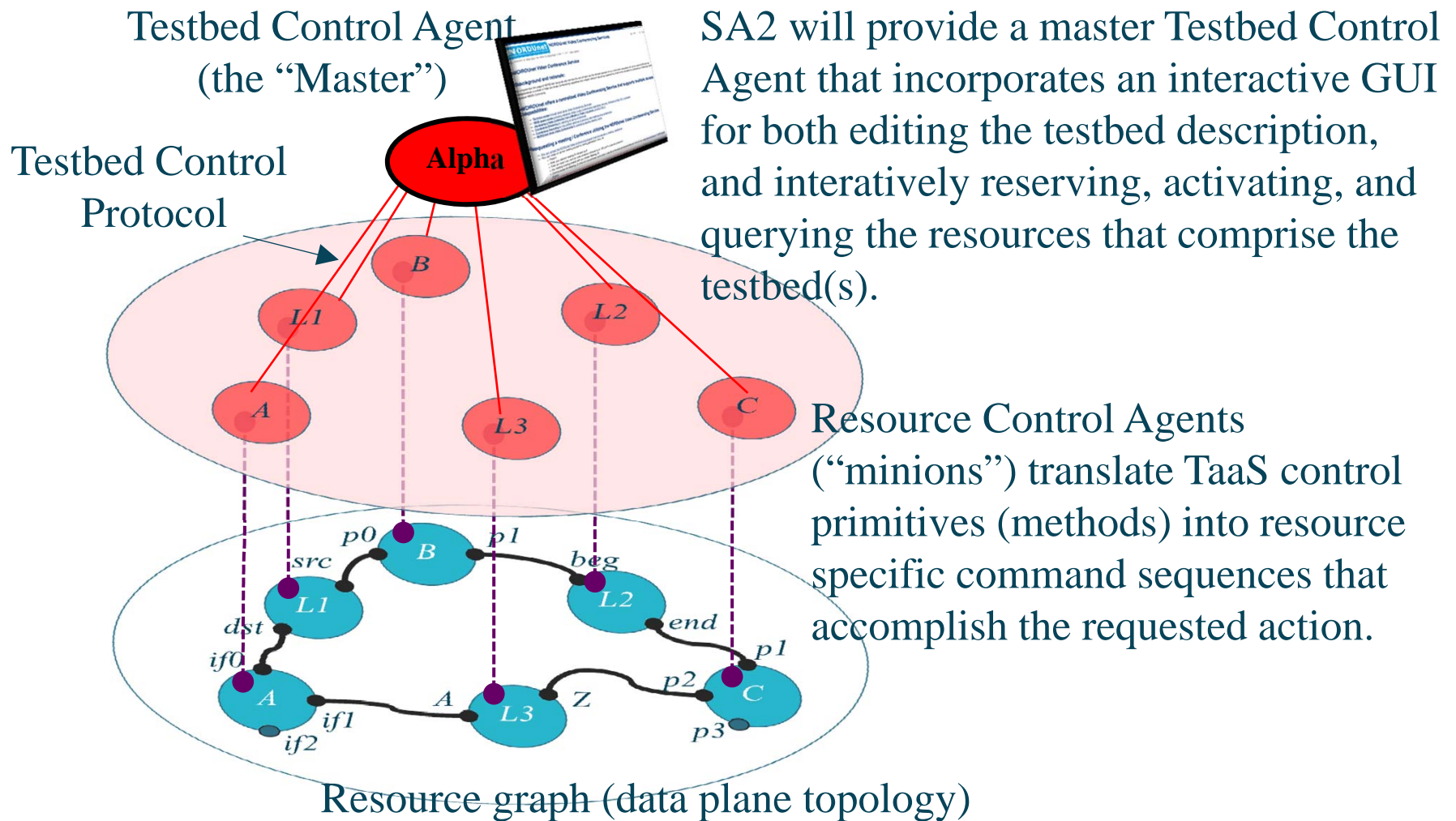
Internally, TaaS represent both nodes and links as generalized “resources” connected via virtualized data ports to create the testbed topology.



## Data plane resource graph



# Standing it up ... Testbed instantiation



# The “Gang of Five” Common Testbed Control Primitives



- **Reserve()** – A request from TCA to RM to find resources and to reserve them for this user/project. The Reserve() primitive specifies a resource class and a number of user constraints on the resource instance. The user may specify a quantity of 1 or more.
- **Activate()** – Given a reserved resource, the primitive instructs the RM to place the reserved resource into service. A resource can only be activated within its scheduled reservation window.
- **Query()** – Obtain the resource specific state information for a particular resource instance
- **Deactivate()** – Take a resource instance out of service, but retain the reservation. This drops alarms for the resource so that maintenance (for instance) might be performed, or might provide a means for re-initializing a resource by de-activating and re-activating it.
- **Release()** – deactivate a resource and release the entire reservation for that instance.



# Resource Specific Testbed Control Primitives



- Each Resource Class defines methods (control primitives) that translate TaaS TCP semantics to resource specific command sequences.
  - Each resource class must implement the gang of five..
  - Each resource class may define additional control primitives/semantics that may be specific to that class of resource only
- For instance:
  - A “LinuxVM” Class may implement a “ColdStart()” primitive that essentially re-initializes the VM via OpenStack and reboots it.
  - That same class may also implement a WarmStart() primitive that simply reboots the OS for a VM.
  - These primitives do not make sense for Virtual Circuit resource instance..

# Integrated Services Approach

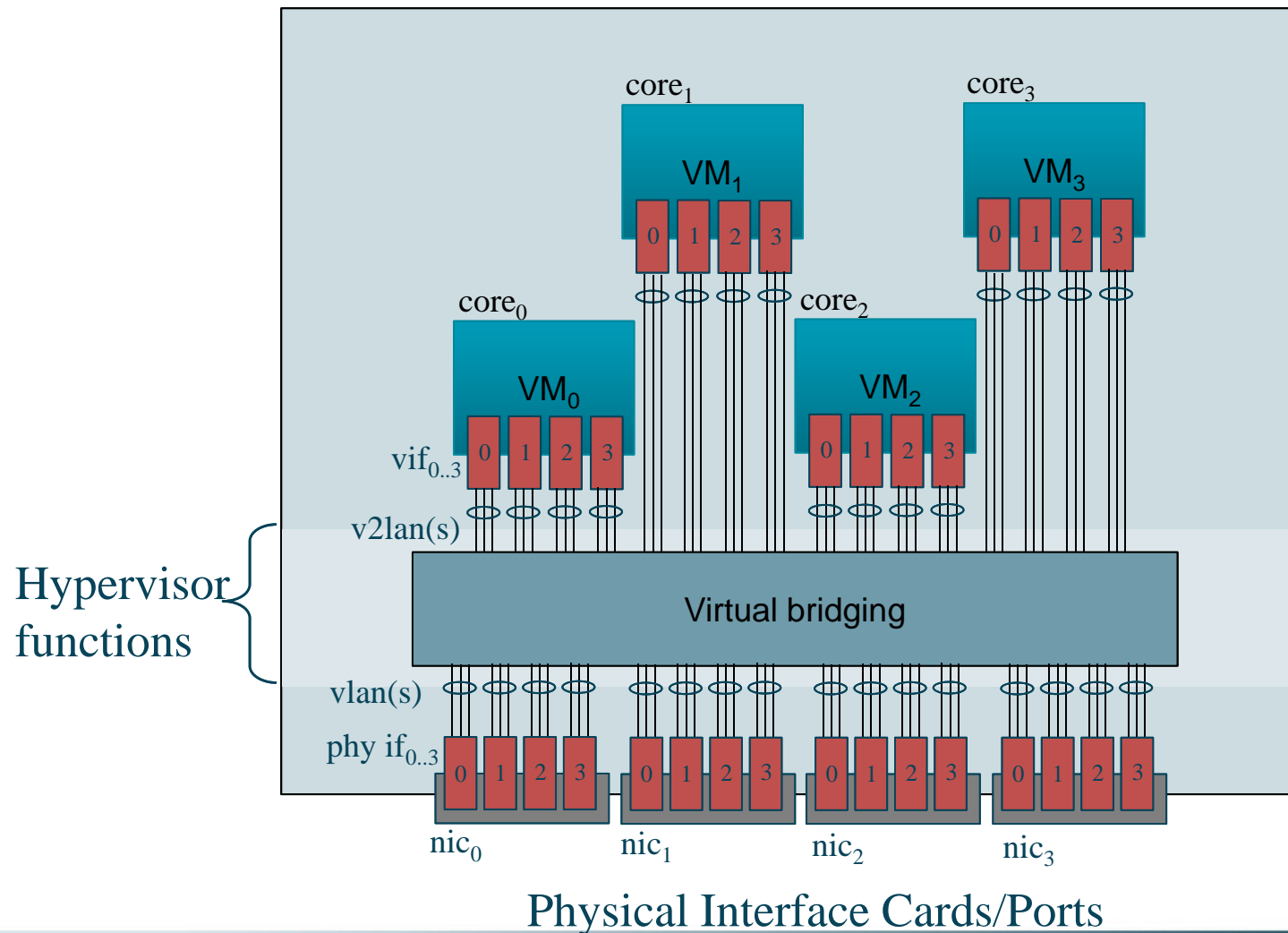


- The SA2 leverages other GN3Plus services:
  - SA1/SA6 Core Eng/Ops – A fully functional routed IP core is essential for the SA2 service access and control plane functions, and for user access to the testbed resources
  - SA3 Bandwidth on Demand will supply the multi-domain virtual circuits and inter-domain transport provisioning via NSI and AutoBAHN
  - SA7 Cloud Services will be exploring the delivery of large scale cloud resources to the GN3+ community – TaaS hopes to influence those efforts, and also to use them to scale up.
  - SA5(?) Identity Management. TaaS will rely on existing and emerging AAI conventions for users and projects and accounting.
- TaaS leverages “Virtualization” in the extreme
  - “Virtual” does not mean “imaginary” (!) - these are very real resources and real networks

# Resource Mapping: VMs to Servers

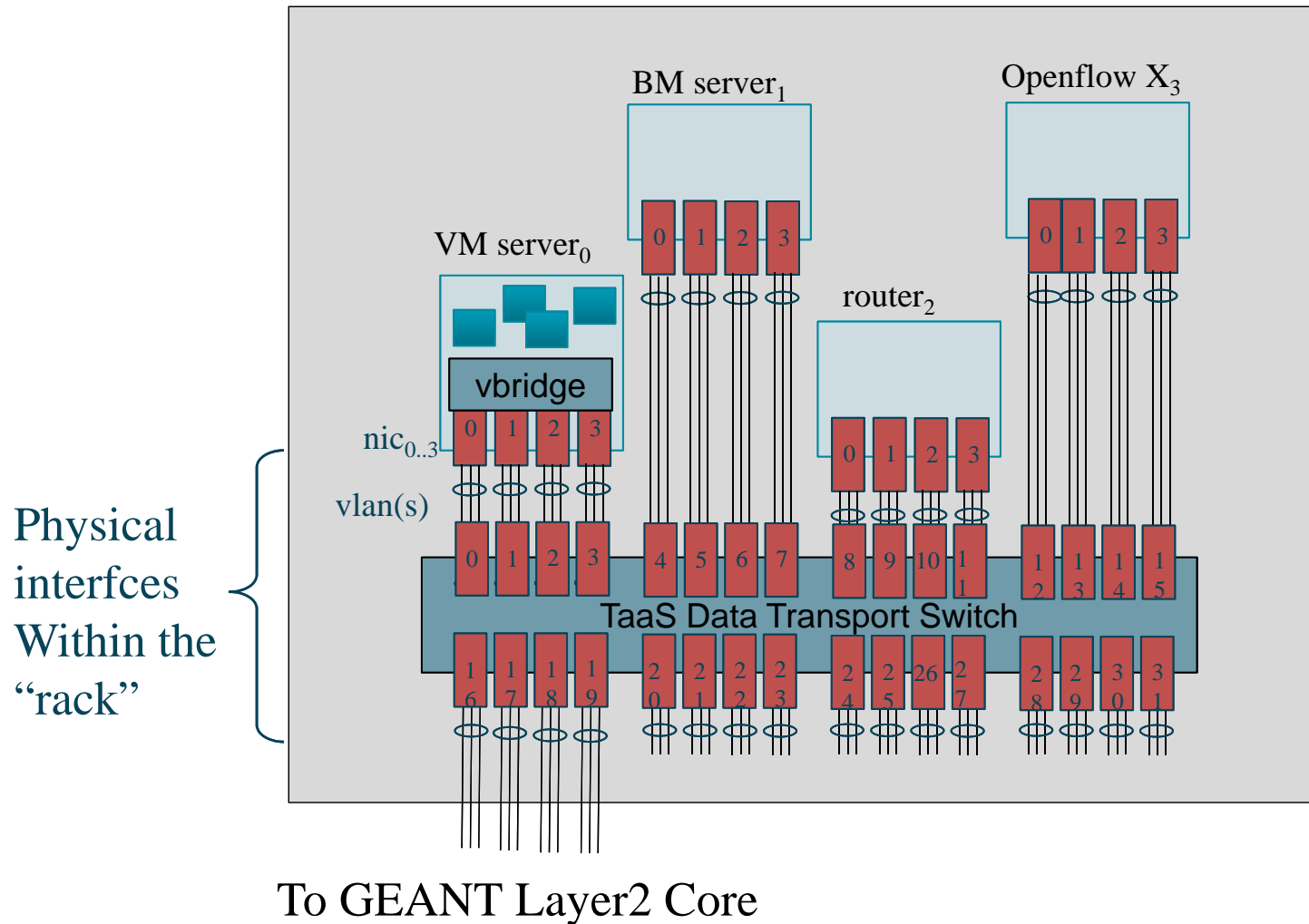


## Physical VM Server Platform

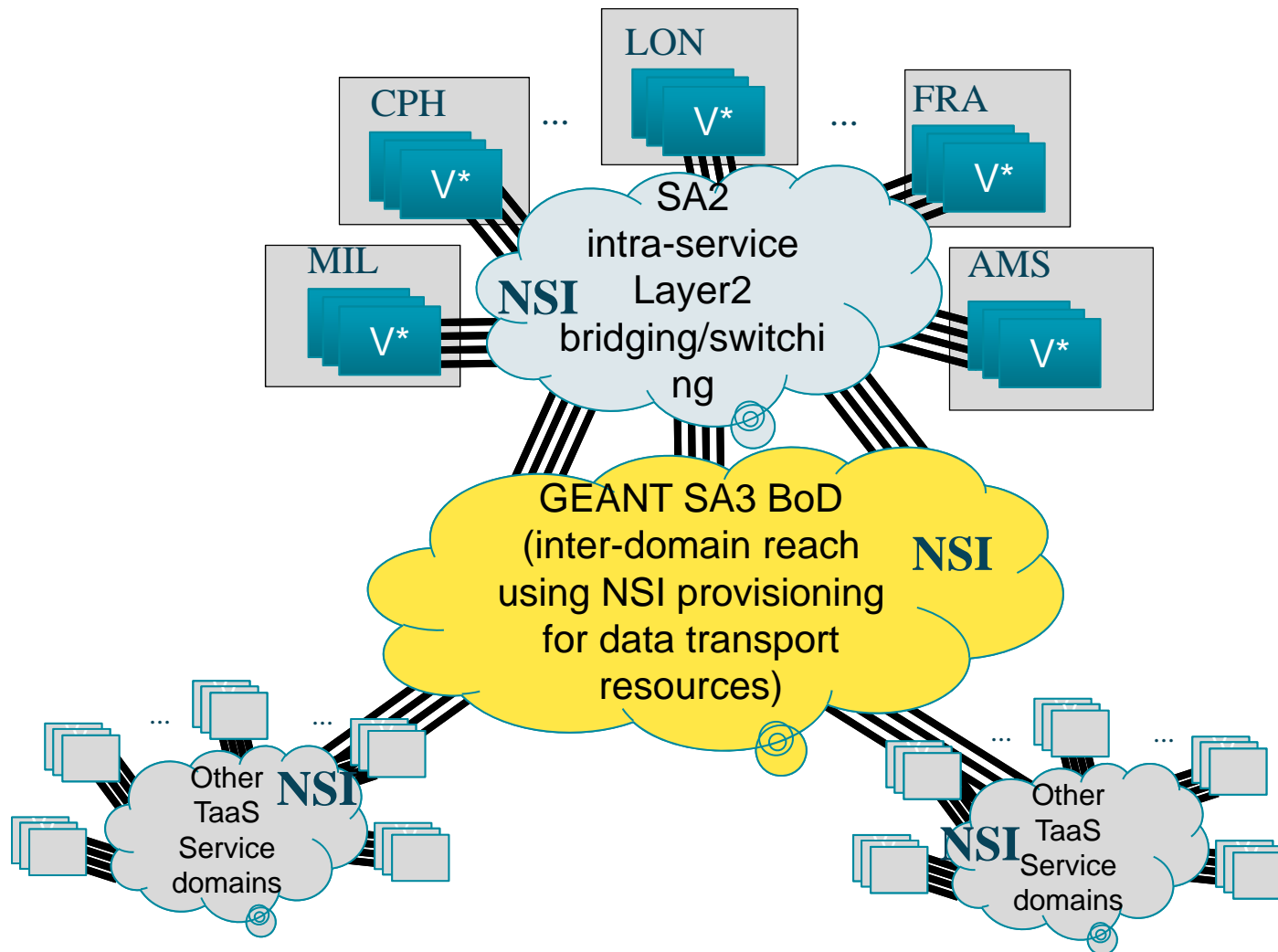


## Resource server platforms within Pods

# SA2 Pod physical connections

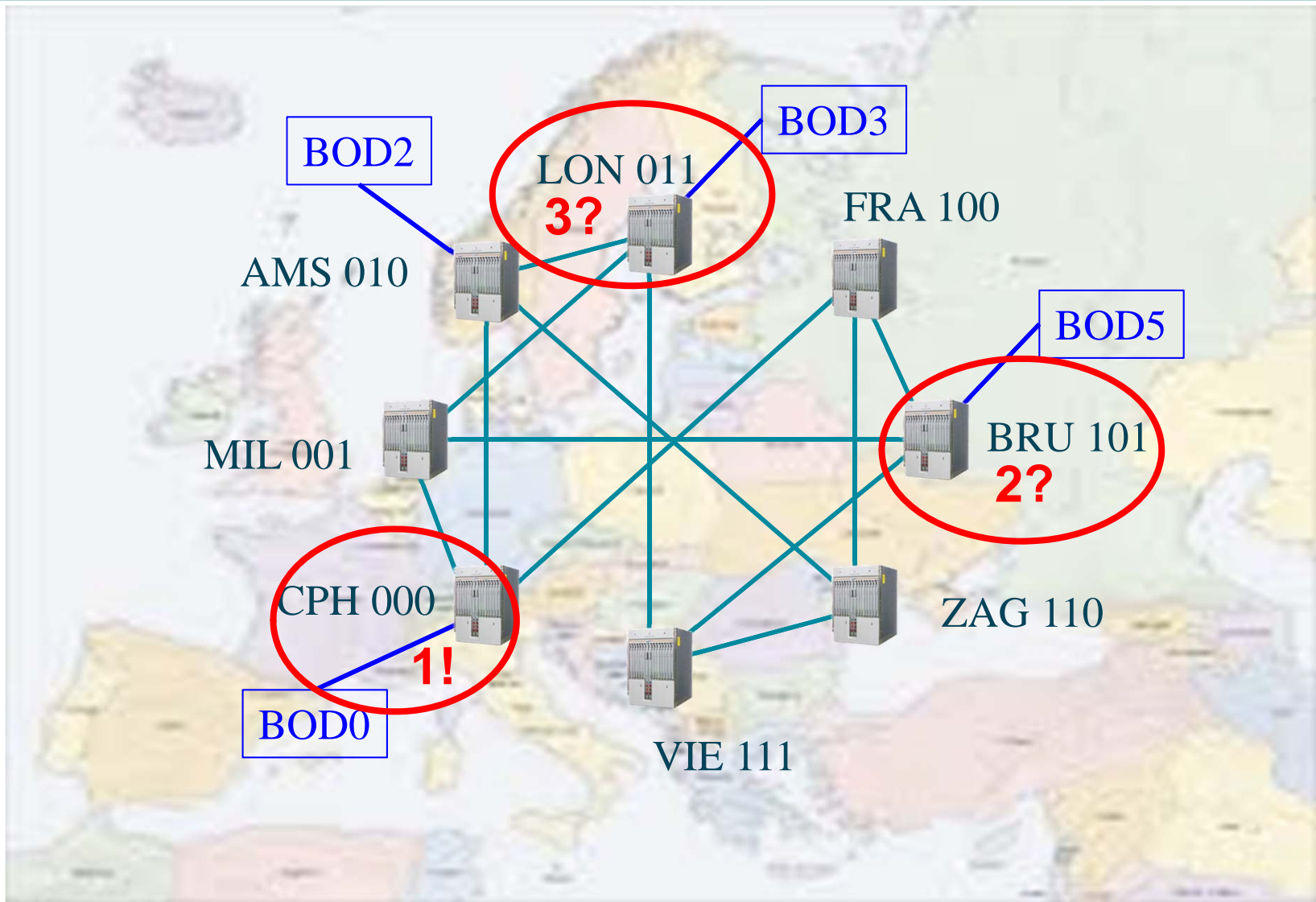


# TaaS initial multi-domain interconnection concept



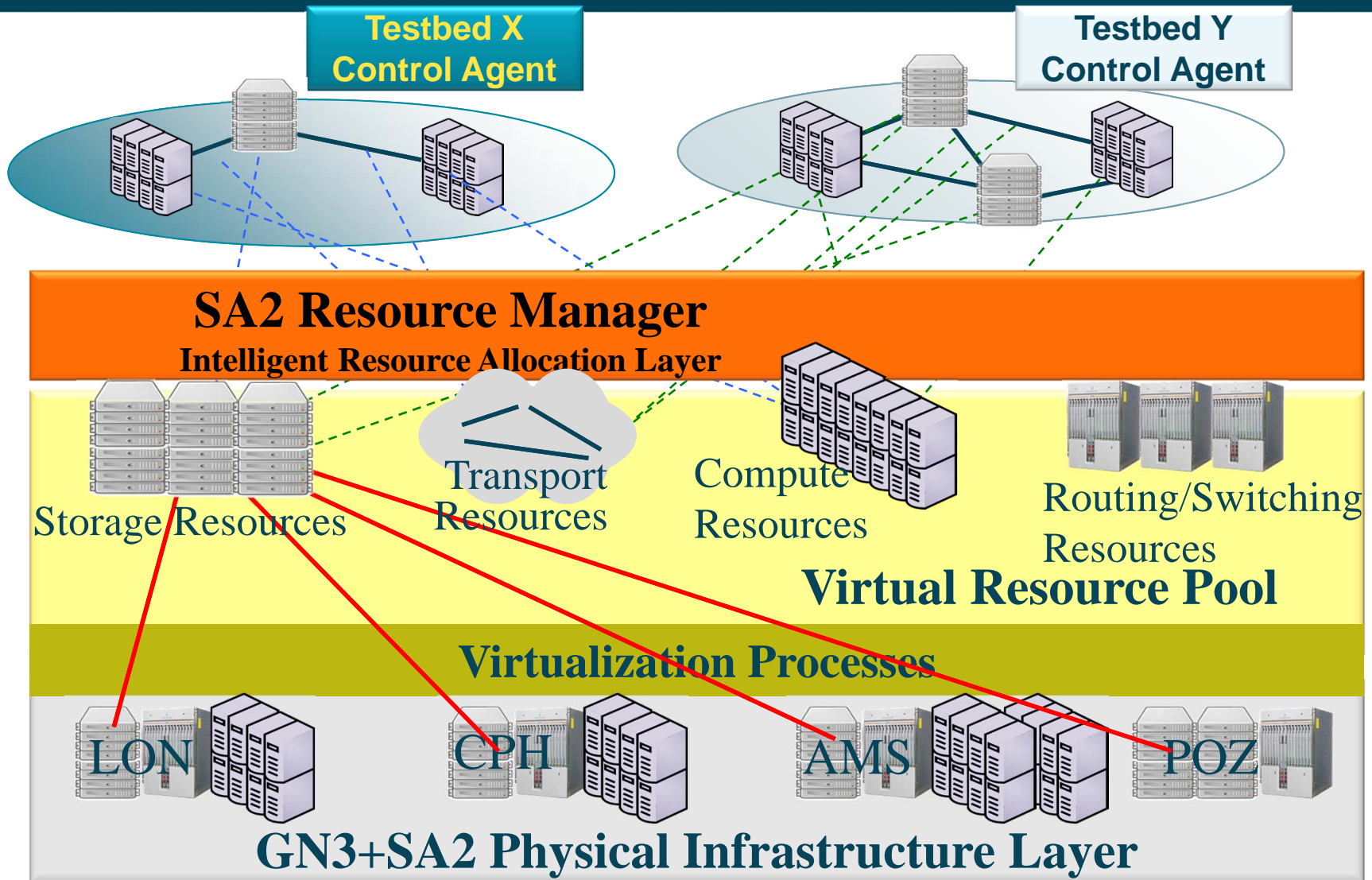
# TaaS Deployment Planning

(draft) Dec 2013

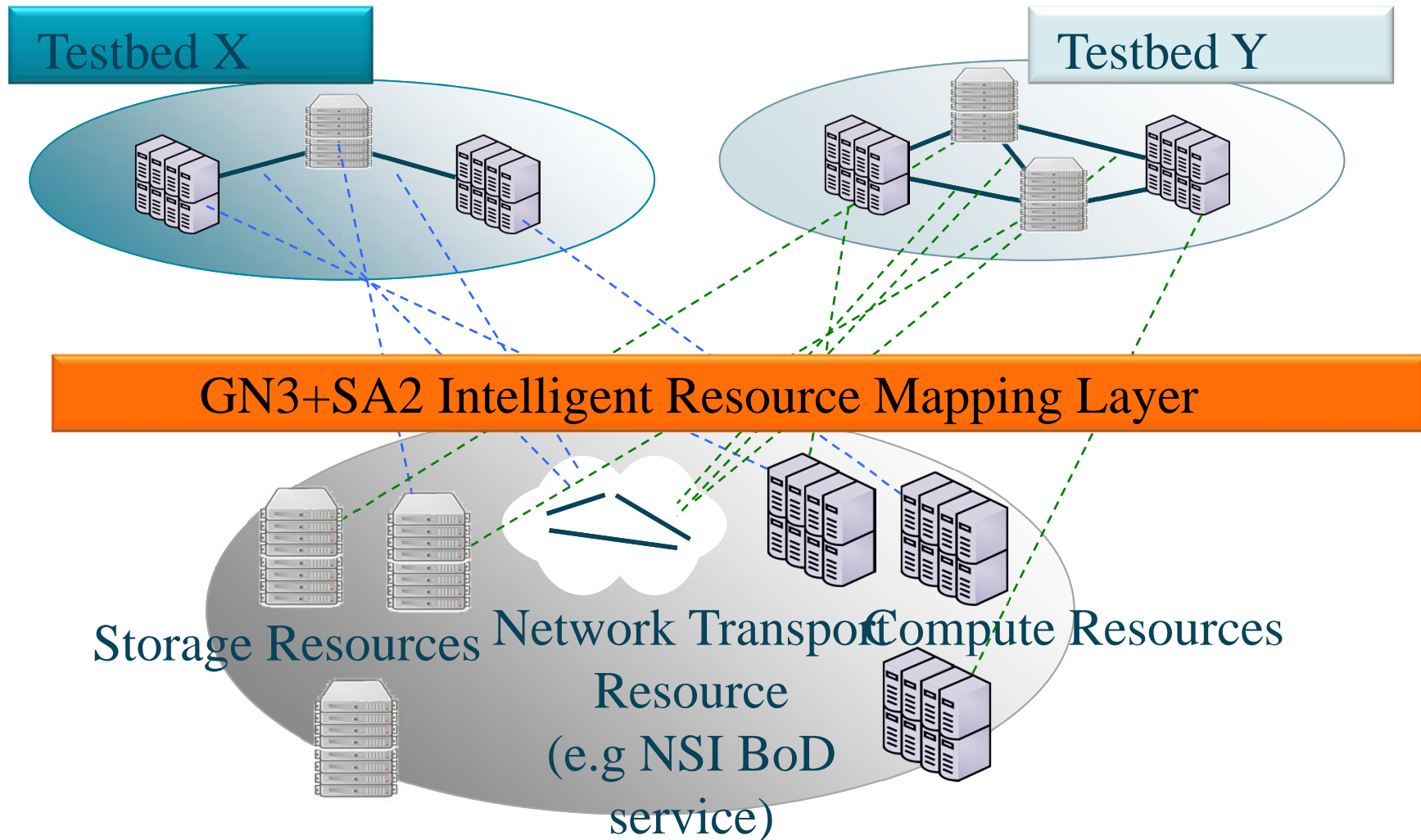




# SA2 Testbeds Phase 1

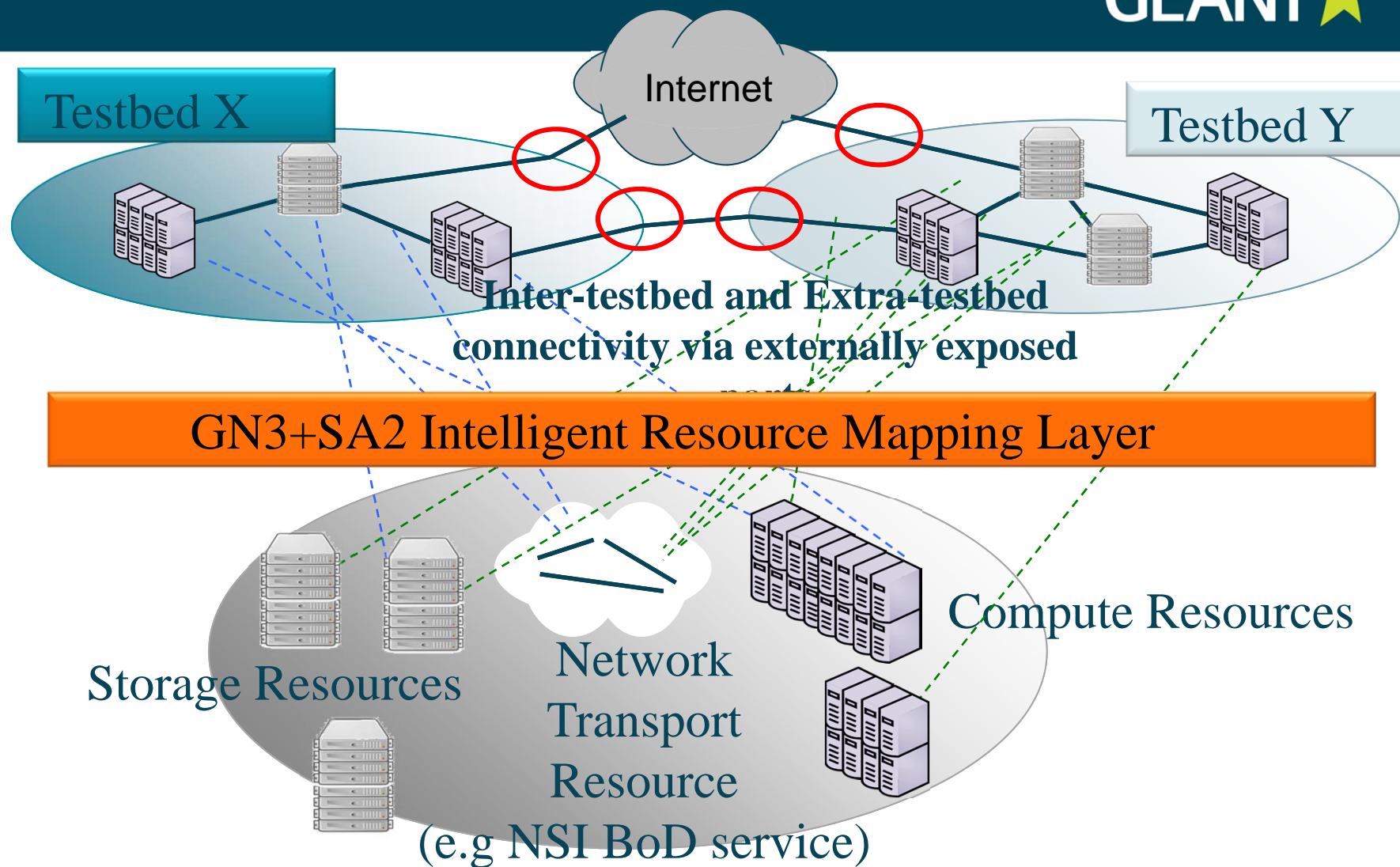


# SA2 Testbeds Phase 1

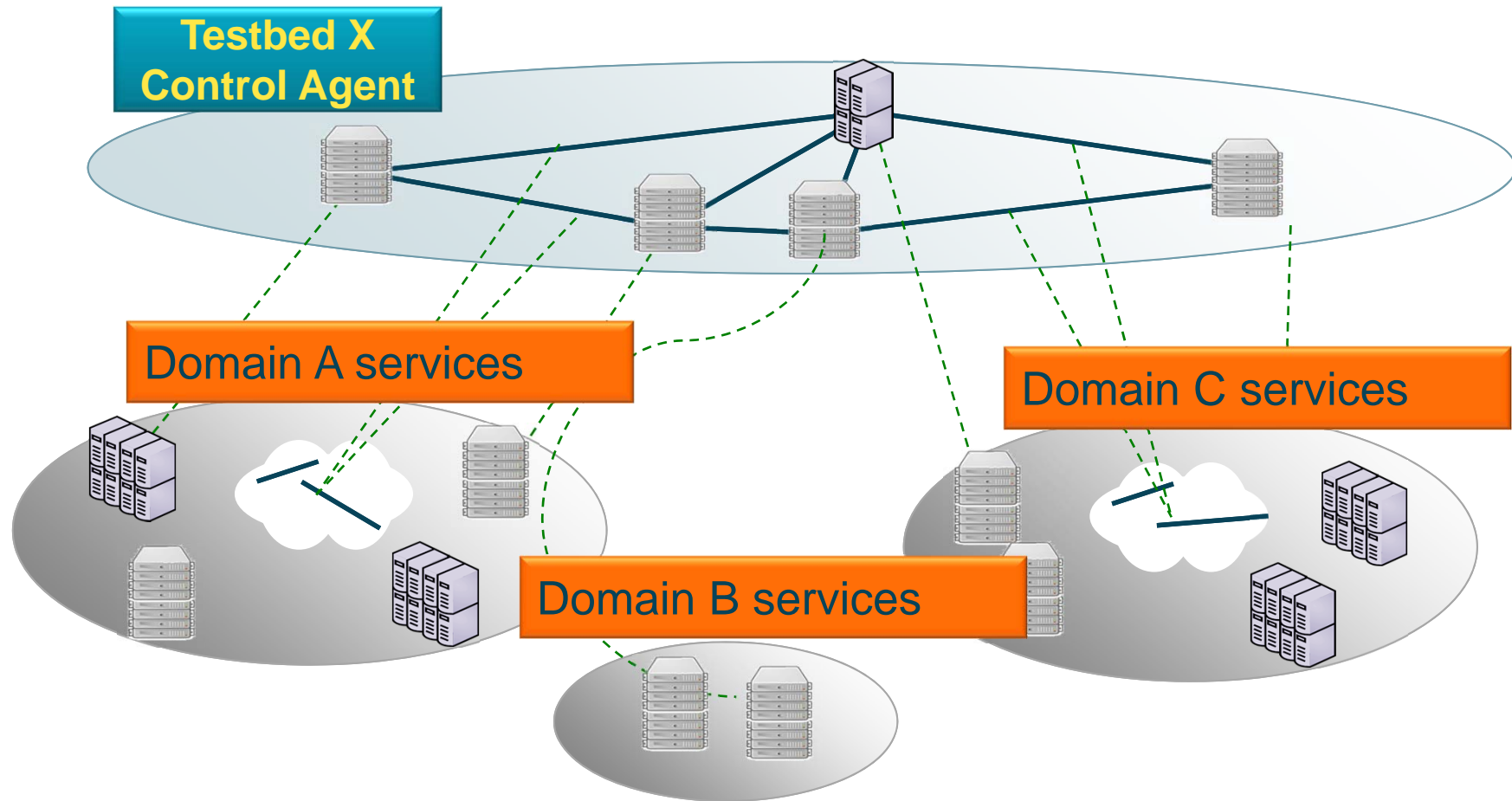


Geographically distributed physical resource pool

# SA2 Testbeds Phase 1plus



# SA2 Testbeds Phase 2



Globally integrated virtualized service domain  
(e.g. global [virtual] SDN domain..?)

# Resources that are planned for v1.0



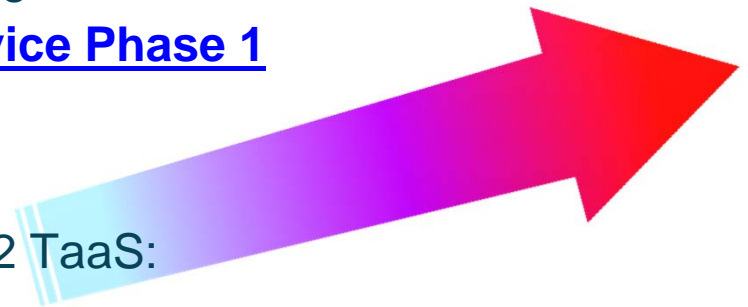
## ● Initial Resource classes:

- Computational Resources
  - *Virtual Machines (KVM)*
    - *“LinuxVM”*
      - *one VM/Core, 2GB, ~500 GB local disk*
  - *Bare Metal Server*
    - *2.3 GHz,*
  - *Switching Element*
    - *OpenFlow Switch (HP5900, OVS(?))*
    - *Virtual Router (MX routers)*
- Transport Resources
  - *Ethernet Framed Transport Service (“EFTS”)*
    - *Up to 10 Gbps*
    - *802.1Q framed*
  - *Considering a 802.1 full port service (TBD)*

# Timeline



- LOTS of SW development happening
- HW ordering, staging, and **deployment...**
- User Guides, Resource support, pperations guides
- **Dec 31, 2013 – TaaS v1.0 production service Phase 1**
- 2014 – scaling up
  - Begin migration from FED/GOFF -> SA2 TaaS:
  - Training/Seminars/Workshops
  - Richer selection of resources/attributes
  - Scaling (GN3+ target:  $10^3$  VMs,  $10^3$  VCs )
  - Reach: Inter-domain multi-domain interoperability (and scaling)





# SA2 Conspirators:



● GARR

● PSNC

● TERENA

● DANTE

● RedIRIS

● DFN

● CESnet

● RENETER

● HEAnet

● NORDUnet



# SA2 Ring Leaders



- SA2 Activity Leader: **Jerry Sobieski (NORDUnet)**



## The [actual] important people:

- T1: Hardware and Systems Eng TL: **Dom Tailor (DANTE)**
- T2: Software Development TL: **Blazej Pietrzak (PSNC)**
- T3: Service Management TL: **Peter Szegedi (TERENA)**
- T4: Multi-Domain Interoperability TL: **Fabio Farina (GARR)**



**Dom Tailor**



**Blazej Pietrzak**



**Peter Szegedi**



**Fabio Farina**

# ...Beyond the GEANT Core



- **SA2 would like the NRENs to participate in the TaaS service architecture**
  - Initially, this means offering up infrastructure and resources according to the SA2 Architecture
  - And enhancing the SA2 architecture to incorporate “brokering”
  - The NRENs can be our first experimental test of multi-domain interoperability - mid 2014 ...
- Ultimately, we want the SA2 TaaS architecture and service model to be an existence proof that large scale, dynamically allocated, multi-domain virtual networks can be deployed in a secure, stable, and globally scalable fashion.
  - And the TaaS’ architecture is able to do this effectively at scale.
- By End of GN3plus (i.e. GN4 launch) we would like there to be open consensus emerging for inter-domain virtual resource management architecture and protocol(s) so that such capabilities can be extended seamlessly and globally.

- Stay tuned... Now the real work begins...

The End

# Questions?

