Federation Interoperability – Dealing With Heterogeneity

Sebastian Wahle, Fraunhofer FOKUS, Germany, sebastian.wahle@fokus.fraunhofer.de

***Abstract*—This position paper shall contribute to the discussion around heterogeneous resource federation models, strategies, and implementation scenarios. From the author's point of view, overcoming heterogeneity is the most challenging issue where not only the resources but also the federation mechanisms and implementations themselves are heterogeneous when federating across the boundaries of federations. Federation might take place on several abstraction layers where in the end different federations might agree to federate on yet another overarching level. This will ultimately lead towards a massive federated resource pool where users can assemble desired functionality across layers and administrative boundaries upon demand in a seamless manner. This vision is driving our work.**

***Index Terms*—Heterogeneous resource federation, Teagle, federation model, interoperability**

I. Introduction

SEVERAL initiatives and projects worldwide currently investigate federation mechanisms. Among those are several well-known projects from the GENI and FIRE initiatives, a full overview has been published before [1]. All those initiatives and projects are currently designing and implementing federation mechanisms and procedures with specific use cases in mind, for example offering a large scale experimental facility to the Future Internet research community. Many of the aspects that need to be dealt with in this context are not new or scientifically interesting and have been achieved in different applications domains in the past. For example computing power federation has been tackled in the Grid domain. Another example is identity federation which has been solved for roaming in Telco networks. However, federating arbitrary resources across multiple administrative domains and on multiple federation levels, involves so many technical, operational, and legal issues that it can be considered a valid research field with many yet unsolved issues. In order to realize the vision of fully federated information and communication technology resources that can be used transparently and seamlessly, the following fields have to be addressed: resources description, resource registration, resource access control, service level agreements, resource usage policies, resource management, resource life cycle, operational procedures, legal framework, provider/user incentives, business framework, market platform, etc.

Although many of the above listed issues have been addressed and widely discussed for single domains, additional constraints arise for multi level federations where administrative domains allow resource usage beyond the first abstraction layer. For example a university might establish a resource federation where different departments adhere to a centralized resource control/management instance, resource description model, operational procedures, etc. and commit resources to a university-wide resource pool. The university might now join a nation-wide initiative (e.g. GENI) where several universities with similar resource control/management schemes agree to federate. This federation is then essentially a federation of federations. The next level is still imaginable: a federation of nation-wide federations (e.g. GENI and FIRE agree to federate). This is basically a recursive model that can be investigated at any meaningful granularity. Generally, the distribution of control power is a central characteristic of federations which is also addressed by formal definitions: *A federation is understood to be an organization within which smaller divisions have some internal autonomy* (Oxford definition). Merriam-Webster defines federal as: (1) *formed by a compact between political units that surrender their individual sovereignty to a central authority but retain limited residuary powers of government*; (2) *constituting a form of government in which power is distributed between a central authority and a number of constituent territorial units.* Although such definitions have a political background, the federation model outlined in the next section has been designed with this in mind.

II. Federation Model

This section introduces the *Base Model* and derives different levels of "surrender"; the *Central Scenario* and the *Distributed Scenario*.

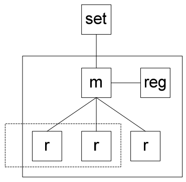

Fig 1: Federation model entities

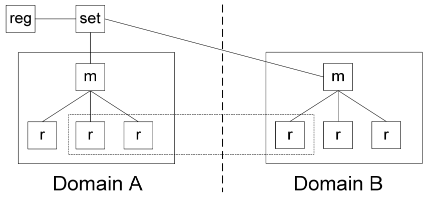

Fig 2: Full surrender scenario

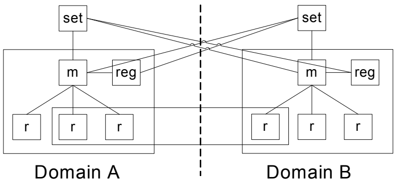

Fig 3: Federation on a par scenario

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.

The Base Model follows the definition of federation given in the previous section which uses the concept of surrendering individual sovereignty to a central authority. This understanding is extended for our field to support resource federations *on a par*. Independent of the level of surrender similar functional entities are found in most federation architectures to enable cross-domain and cross-technology federation. The entities are shown in Figure 1 and are defined below. They constitute a proposed FIRE federation model [2]. The entities are meant to be meta-entities that could be mapped to the entities defined by other approaches such as ORCA, Panlab, PlaneLab, or SFA. Such mappings shall ultimately allow for heterogeneous federations across the "silo" approaches that are currently been driven independently.

Resources (r): The model abstracts from concrete resource types. A resource can be anything that can be controlled by software. Examples are: physical and virtual machines, software packages, dedicated hardware such as sensors and routers, as well as abstract constructs such as for example domains, accounts, databases and identities. Resources may contain other (child) resources.

Domain manager (m): software that controls resources inside an administrative domain. It exposes resource management functionalities at the border of a domain and connects to a resource registry. Supported operations on resources are typically the CRUD (create, read, update, delete) commands for controlling resources via a common interface. Proper security mechanisms and policies need to be supported in order to protect administrative domains from resource misuse.

Registry (reg): holds data records for domain resources. Registries may or may not expose an interface to (external) setup utilities (set).

Creation / setup tool (set): resides within or outside of a domain and communicates with domain manager and registries. Set utilities provide a user interface for the configuration, deployment, and monitoring of virtual resource groupings.

Virtual grouping of resources (dotted rectangle): each administrative domain enables access to a number of resources. Collectively, all administrative domains provide a large pool of resources. Experiments usually require only a subset of the total resources that need to be provided in a certain configuration. This subset may or may not span the border of several domains and is here referred to as a virtual grouping.

Administrative domain (solid rectangle): is typically represented by an organization such as a research institute and provides a collection of resources.

The *Central Scenario* is what we also call the *full surrender* scenario in Figure 2 where the resources committed from domain B can be fully controlled via domain A. An example of the full surrender scenario is the Panlab federation [3],[4] where all Panlab member domains allow Teagle (section III.A), the central setup tool (set), to control resources in their domain. It relies on a central registry where resources from all member domains are registered. The advantage of this scenario is that resource representations backed by centrally administered resource models and operational procedures can be simplified. As all central solutions, this approach faces scalability, trust, and availability issues.

The *Distributed Scenario* is what we also call the *federation on a par* scenario in Figure 3 where the participating domains allow the mutual control of resources across the borders of their domains. Here, the set utilities are allowed access each other's domain managers and registries. This enables the full scale of resource sharing across organizational boundaries. However, in order to achieve this, a number of agreements need to be in place such as common resource descriptions and management interfaces. Legal and operational procedures as well as resource access and usage policies are more difficult to realize compared with the central scenario. This scenario has been implemented to federate Panlab resources and resources from a private PlanetLab installation and is described in section III.C. Other scenarios that implement something in between the two extreme scenarios explained above are possible and can be applied to meet requirements and constraints in specific federation contexts.

III. Prototypes & Use Cases

A. *Teagle & Panlab Framework*

Teagle [5] is the central federation resource search and composition engine for Panlab and can be mapped to the *set* entity of the Base Model. It provides a web-based interface that allows browsing through the federation's offerings, enables the definition of a *virtual resource grouping* (this is basically a slice in GENI terminology) and executes the provisioning thereof. A virtual resource grouping is an isolated network where the experimenter has direct access to the resources and configurations provisioned by Teagle. Each experimenter operates inside its own virtual resource grouping and has no access to other groupings. Currently, Teagle implements the following functions (see also figure 4):

- Registry (users, resources, configurations, etc.)
- Creation Environment (setup and configuration of virtual resource groupings, this is called the VCT tool)
- Request Processor (validates configurations and triggers setup execution)
- Orchestration Engine (generates an executable workflow that orchestrates services form different domains to actually provision resources for the experimenter)
- Web Portal (exposes search, configuration interfaces, and general information)

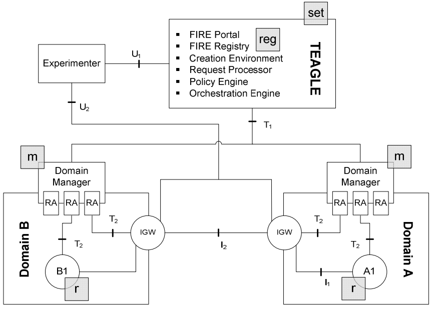

Fig 4: FIRE resource federation framework and mapping to the model entities

A resource can be anything that can be controlled by software. In our prototype implementation resources are controlled by resource adaptors (RA) that are plugged in to the Domain Manager (DM) framework. RAs are implemented following the DM framework guidelines for pluggable modules. An RA can be seen as a device driver that supports resource specific communication on interface T2. Examples for T2 communication are Service Provisioning Markup Language (SPML) based messages, the Simple Network Management

Protocol (SNMP), or command-line interface (CLI) commands. Any type of resource can be supported by the DM as long as an RA can be implemented and the configuration options can be described and modeled so that the *set* and *reg* entities (Teagle) can handle them. This approach allows us to manage heterogeneous resources that support a variety of different communication mechanisms, reside in different layers and belong to different administrative domains.

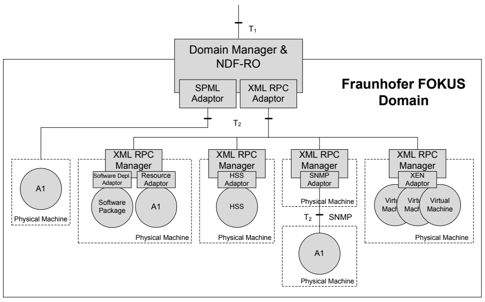

Fig 5: FOKUS administrative domain and its domain manager framework

Through its modular structure, the prototype DM supports multiple resource provisioning schemata and languages, and enables the incorporation of various resources and their native communication mechanisms. For example it is possible to instantiate several virtual machines on a physical machine and deploy software RAs on the virtual machines themselves in order to control both the container and the actual software resource residing inside the virtual node. Resources that already natively support a provisioning schema, such as SPML, can be directly controlled. We call this concept that supports both pluggable resource adaptors (SNMP, CLI, etc.) as well as pluggable provisioning schemata (SPML, XML-RPC, etc.) Network Domain Federation Remote Objects (NDF-RO). Generic management operations supported by the resources are exposed as REST (Representational State Transfer) services on interface T1. This allows for a flexible support of heterogeneous resources. Together with the management operation, an XML document is send via T1 to the DM, carrying configuration parameters to be applied to a specific resource.

B. *Use Case 1: Phosporus & HPDMnet integration*

This use case demonstrates the *Central Scenario* where a central set utility and a central resource registry are used to control the resources offered by participating domains. The involved domains are part of already existing, successful networking testbeds: Phosporus and HPDMnet.

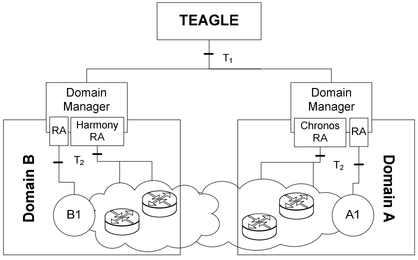

Fig 6: Phosporous (Harmony i/f) and HPDMnet (Chronos i/f) integration

Phosphorus addresses some of the key technical challenges to enable on-demand end-to-end network services across multiple domains over high speed optical networks. It uses the Harmony service interface. HPDMnet builds a high performance digital media network that provides end-to-end network services across multiple domains like Phosphorus but focuses on high bandwidth media streaming like uncompressed high-definition video streaming. It uses the Chronos interface which is a Harmony derivative. The prototype implementation allows establishing a high bandwidth network path across the Phosphorus and HPDMnet testbeds using the central *set* utility *Teagle*. Both testbeds can create dynamic network services modeled as Teagle resources. It has been demonstrated to set up a path with specific bandwidth from Canada to Spain using Teagle and its corresponding control and federation framework.

C. *Use Case 2: PlanetLab / SFA integration*

Here, the proposed scenario *on a par* is discussed. Our prototype [6] implements a full federation between a private PlanetLab (at Fraunhofer FOKUS premises, domain A) and a Panlab domain (also at Fraunhofer FOKUS, domain B). Both domains maintain their own registry services and provide a domain manager to allow the other party to provision resources in their domain. We implemented a module, called *SFAAdapter*, which acts as a bridge between the internal semantics and protocols of a Panlab domain manager and the SFA. From the viewpoint of the SFA, this module appears just as any other federation partner, exposing the interfaces of a registry and aggregate manager. Users and administrators interact with their local creation tools (Teagle on the Panlab side and slice manager on the PlanetLab side) to create and manage virtual resource groupings (VCTs in Panlab terminology, slices in PlanetLab/SFA terminology) which may span both federations. The slice managers delegate look-up requests to their local registry which will in turn query the foreign registry if necessary. Provisioning requests are issued directly to the respective domain managers which forward these requests to their components.

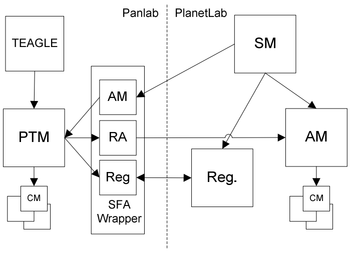

Fig 7: Panlab & PlanetLab integration

The implementation of this scenario could be achieved without major efforts because the resources federated were rather homogenous (just nodes). Greater challenges are expected when moving to arbitrary resources and services. Also, trust, policy and operational issues are expected. These are the challenges for future work.

References

[1] Thomas Magedanz and Sebastian Wahle. Control Framework Design for Future Internet Testbeds. e & i Elektrotechnik und Informationstechnik, 126(07/08):274-279, July 2009. ISSN: 0932-383X (print) ISSN: 1613-7620 (online).

[2] 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. IOS Press, 2010. To appear.

[3] Anastasius Gavras, Halid Hrasnica, Sebastian Wahle, David Lozano, Denis Mischler, and Spyros Denazis. Towards the Future Internet - A European Research Perspective, chapter Control of Resources in Pan-European Testbed Federation, pages 67 - 78. IOS Press, ISBN 978-1-60750-007-0 (2009)

[4] Website of Panlab and PII European projects, supported by the European Commission in its both framework programmes FP6 (2001-2006) and FP7 (2007-2013): http://www.panlab.net

[5] TEAGLE portal website: http://www.fire-teagle.org

[6] 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, 2010. To appear.