8

BonFIRE: A Multi-Cloud Experimentation-as-a-Service Ecosystem

Michael Boniface[1], Vegard Engen[1], Josep Matrat[2], David Garcia[2], Kostas Kavoussanakis[3], Ally Hume[3] and David Margery[4]

[1]University of Southampton IT Innovation Centre
[2]Atos
[3]University of Edinburgh
[4]INRIA

8.1 Introduction

The demand for ways to explore and understand how applications and services behave in a shared software defined infrastructures is increasing. Completely new applications are emerging, alongside "Big Data" and the convergence of services with mobile networks and the Internet of Things (IoT) all exploiting Cloud scalability and flexibility along with integration with software defined networks. These innovative technologies are creating opportunities for industry that requires a new collaborative approach to product and services that combines, commercial and funded research, early-stage and close-to-market applications, but always at the cutting edge of ideas.

The range of application sectors places significant challenges for cloud infrastructure and application providers. How to manage infrastructure resources considering the new types of demand? How will applications behave on a shared virtualised resource? This is not a new problem and some of the issues are now being addressed by Platform-as-a-Service providers, but the landscape is changing again as the convergence of cloud computing and dynamic software-defined networks picks up pace. The merging of industries and technology requires a collaborative approach to product and service innovation that allows technical and businesses exploration across the traditional boundaries of telecommunications and cloud infrastructures.

In this chapter we summarise six years of cloud and services experimentation at the BonFIRE facility which ran its last experiment on 30 May 2016. We show how BonFIRE delivered impact and broke new ground for technically advanced and sustainable Experimentation-as-a-Service (EaaS) platforms supporting cloud and service innovation with cross-cutting networking affects.

8.2 A Cloud and Services Experimentation Service

BonFIRE was a multi-site experimentation service for research and development of novel cloud and networking products and services. BonFIRE allowed customers to outsource testbed infrastructure on-demand by offering the four key capabilities necessary for experimentation: control, observability, usability and advanced cloud/network features (e.g. cross site elasticity, bandwidth on-demand). These features lead to reduced barriers to entry for providers of innovative Cloud offerings.

BonFIRE provided infrastructure capacity to support medium scale cloud experiments through a permanent infrastructure providing a hub that was used as the foundation for growth to larger scale experiments through additional on-request resources and relationships with 3rd party suppliers. BonFIRE operated a multi-cloud broker that brought together pan-European providers of cloud and network infrastructure. Uniquely, BonFIRE offered capabilities to control cloud computing and network infrastructure using a single interface, in this way experimenters could explore cross-cutting effects of applications, clouds and networks, in scenarios with increasing levels of realism. Software technologies could be deployed on demand either on a single site with highly controllable emulated networking or on multiple sites with controlled wide-area networking. No other public cloud or network provider offered this capability at the time. With a prioritisation on ensuring accuracy and confidence in results, BonFIRE allowed experimenters to control and observe the behaviour of physical and virtualised infrastructure in ways that was not offered by existing public cloud providers (e.g. Amazon, Rackspace, or Flexiant). BonFIRE achieved the differentiation by targeting Research Technology and Development (RTD) phases of the technology lifecycle rather than downstream production deployments of customer technology. BonFIRE capabilities were designed for testing and experimentation, rather than production runs where business drivers require operational decisions that prioritise service level guarantees and scale rather than controllability and observability.

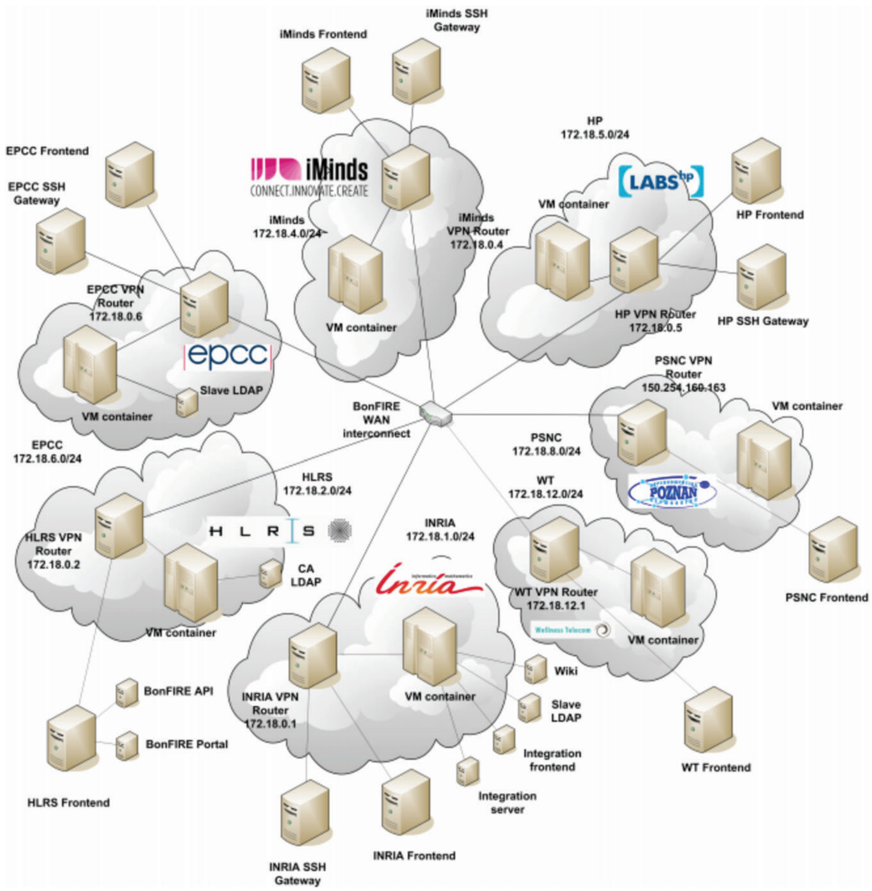

Figure 8.1 The BonFIRE infrastructure.

BonFIRE's targeted experimenters where those with insufficient capital or requirement for long-term investment in dedicated testbed facilities themselves. This includes Small and Medium Sized Enterprises (SMEs), academic researchers, and research collaborations (e.g. EC Projects). BonFIRE was not a "mass" market service, but at the same time, most users are largely self-supporting and the service was not tailored for each customer. Supporting experimenters in the development of service strategies was a key part of EaaS along with tools to transition technology from service design to service operation in production environments. BonFIRE recognised that transitioning new services from an experimental facility to production environments efficiently was essential to reduce the time to market by interoperating with production cloud providers to ensure technology could be transferred to mainstream deployment easily.

BonFIRE offered a multi-site, geographically distributed set of federated testbeds. At its peak, BonFIRE included seven sites across Europe, which offer 660 dedicated cores, with 1.5 TB of RAM and 34 TB of storage (See Figure 8.1). An additional 2,300 multi-core nodes could be added to BonFIRE on user-request using additional capacity at testbed sites, each heterogeneous in terms of Cloud managers, with OpenNebula[1], HP Cells[2] and VMWare employed; the hypervisors and the types of hardware employed are also very varied. In addition to Cloud resources, BonFIRE allowed access to the Virtual Wall emulated network facility with proxy access to Amazon EC2 resources, access to FEDERICA[3] and the AutoBAHN Bandwidth on Demand[4] service of GÉANT. More recently BonFIRE was integrated within the European Federation of future internet testbeds FED4FIRE[5] enabling many new experiments wanting to explore clouds in the context of Internet of Things and mobile networking.

8.3 Technical Approach

Design Principles and Architecture

BonFIRE offered services based on unique design principles that were not easily obtained in public clouds but are important for cloud-based testing on novel future internet applications. These principles included:

- Controllability: allow experimenters to control the infrastructure at multiple levels by specification their resourcing requirement not only on virtualisation level, but also on the underlying physical level (e.g. deploy two VMs on the same physical host).

---

[1]New applications emerge exploiting Cloud scalability and flexibility along with integration with software defined networks.

[2]HP Labs cloud-computing test bed projects –Cells as a Service, http://www.hpl.hp.com/open_innovation/cloud_collaboration/projects.html

[3]Peter Szegedi et al., "Enabling future internet research: the FEDERICA case", IEEE Communications Magazine, Vol. 49, No. 7, pp. 54–61, July 2011.

[4]GÉANT Services – AutoBAHN, http://geant3.archive.geant.net/service/autobahn/Pages/home.aspx

[5]Vandenberghe, W., Vermeulen, B., Demeester, P., Willner, A., Papavassiliou, S., Gavras, A., … & Schreiner, F. (2013, July). Architecture for the heterogeneous federation of future internet experimentation facilities. In *Future Network and Mobile Summit (FutureNetworkSummit)*, *2013* (pp. 1–11). IEEE.

- Scalability: allow experimenters to construct high-scalable infrastructure for running their experiment by adjusting the size of the infrastructure at runtime.
- Federation: provide seamless integration and unique access to cloud services under different domains of control through standard protocols.
- Heterogeneity: support provisioning of different infrastructure consisting of various VM types and networking resources from geographically distributed cloud constituents.
- Networking: provide highly networked resources allowing experimenter to emulate complex and dynamic internetworking environments for their experiments.
- Observability: allow experimenter to define and gather infrastructure-level, both virtual and physical level, and application-level metrics to evaluate and analyse experimental results.

BonFIRE was designed and operated to support testing of cloud applications based on the notion of deploying software defined infrastructure resources in ways that allows testing to monitor what's going on inside the cloud allowing understanding of the performance and behaviour of the system under test, the causes of their degradation and the opportunities to improve them. BonFIRE was not a site for production running or for routine application development. BonFIRE was for experimentation through empirical investigation, which can be in a wide variety of research areas including but not limited to elasticity, cloud reliability, networking, heterogeneous clouds and federation. Different levels of access were offered including basic cloud infrastructure, impact of cloud on an existing application, investigation of new scenarios such as next generation mobile networks.

BonFIRE provided an experimentation platform which is not only highly controllable at all levels, but also offered tools to enable experimenters to investigate in-depth. Designed for usability and versatility experimenters could quickly get down to the details of their work, often under strict time-constraints. On top of this, BonFIRE offered unique testbeds for cross-cutting research in network effects, bandwidth on demand, and heterogeneous servers, and advanced tools such as the ability to emulate contention effects. All features were offered through the BonFIRE Resource Manager (RM), facilitating access to the disparate and geographically distributed resources, and in the management plane, and perhaps above all in the choice of well-defined interfaces which enable researchers to define, control, run and re-run their experiments according to their needs.

A high level view of the BonFIRE architecture is shown in Figure 8.2. Users can interact with BonFIRE using a web Portal, an Application Programming Interface using the Experiment Manager (EM) using a declarative, multi-resource, deployment descriptors or using the BonFIRE RM that provided a RESTful, Open Cloud Computing Interface (OCCI) [REF] interface to create and manage resources one at a time. Interactions with the BonFIRE API were programmed or scripted using a variety of tools. BonFIRE used a centralized broker-wrapper architecture for federation implemented in the RM. The RM service maps user requests to the appropriate infrastructure site and used an implementation of the wrapper pattern to translate these requests to the appropriate format for each site.

Components

In this section we describe in more the components within the BonFIRE architecture.

Portal

The Portal offers the experimenter a graphical interface to the BonFIRE capabilities. It has a view of the experimenter's data, the running experiments, and the available platform capabilities. The Portal accesses the functionalities

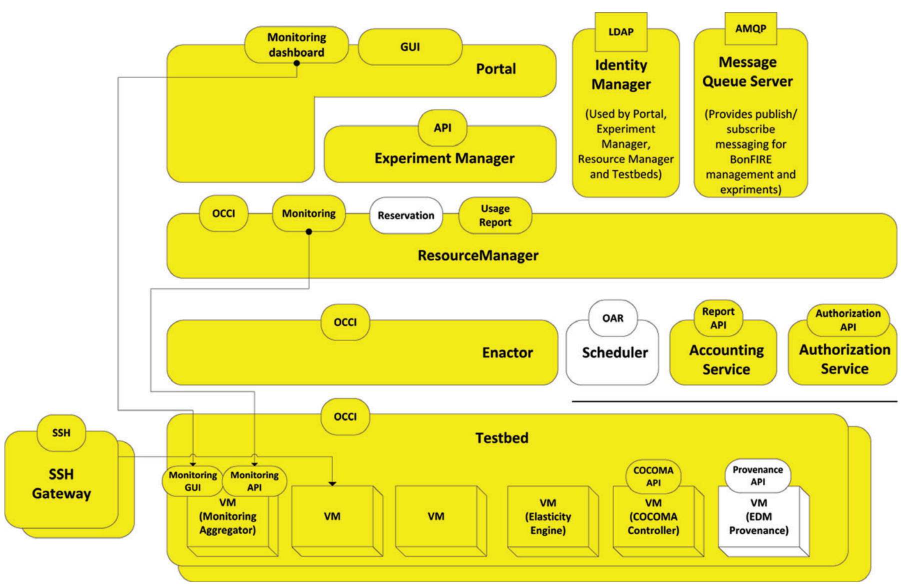

Figure 8.2 The BonFIRE architecture.

exposed by the BonFIRE Application Programming Interface (API). Every function performed through the Portal could be performed by the experimenter without using the Portal by issuing the respective HTTP requests directly to the API. The task of the Portal, however, is to make this process much more convenient and provide a concise overview of the resources and options available to the experimenter. The Portal furthermore provides additional documentation and guidance to the user. The Portal is implemented as a web application written in the python programming language and implemented as a set of plugins to the content management system and web application framework Django[6].

Experiment Manager (EM)

The Experiment Manager (EM) provides a simple RESTful HTTP interface to allow users to create a managed experiment by uploading an experiment descriptor file. The experiment description is parsed and validated immediately, and the user is notified of the success or failure of this stage. The experiment will be deployed in the background by making successive calls to the RM, and the user can check the status by doing a HTTP GET on the managed experiment resource. Through the use of GET, the user can also download the experiment log file, which lists messages on the progress of the experiment. The EM keeps track of a 'managed experiment' resource, which has a status and a link to the URL of the experiment on the RM. The managed experiment can also be deleted from the EM; this will also delete the experiment on the RM.

BonFIRE's investment to ease of use was the inception of a domain-specific, declarative experiment descriptor. The JSON-formatted BonFIRE Experiment Descriptor covers all BonFIRE features that are invoked at deployment time. Unlike the transactional OCCI interface, the user submits a single document to the EM interface. The EM identifies dependencies between resources and decides on order of execution. Consider for example an experiment that has a monitoring Aggregator using a separate storage at Cloud Site A; one compute at Cloud Site A and another one at Cloud Site A B. The EM will first create the storage; then creates the Aggregator and take its site-supplied IP; and finally create the VMs and pass that Aggregator IP to them as part of their context. The Experiment Descriptor is the cornerstone of usability for BonFIRE, the vehicle for Experimentation-as-a-Service. In the context of Cloud testing, what the users want to do is deploy large scale experiments, on

[6]https://www.djangoproject.com/

various facilities. What they then want to do is run the same experiment, under controlled conditions, to build the statistical confidence that their findings are correct and collect the data that prove it. What they may also want to do is to change the deployment to different target systems, to observe the effect.

Resource Manager (RM) and APIs

The RM is the component that provides the resource-level API through which users, and higher layers such as the Portal and EM, interact with BonFIRE. The RM is the entry point for programmatic resource level interactions with BonFIRE. The RM API is an open interface based on the Open Cloud Computing Interface (OCCI)[7] that allows experimenters to build their own clients or use direct Command Line Interface (CLI) calls to the API, which can be embedded in scripts. Through the API, BonFIRE allows experimenters to select the site on which to deploy their VM. A motivation might be a particular application topology the user is interested in studying, in which specific components of the application can be placed at specific sites. One step up from observing, the BonFIRE user can specify themselves on exactly which host to place their VM. This feature could be used to deploy their VM on the specific kind of hardware that they prefer, and BonFIRE's sites have different hardware both between them and inside them.

The Portal is an example GUI client of the RM API. Others include a client toolkit called Restfully and the BonFIRE Command Line Interface (CLI). Restfully[8] is a Ruby library that utilizes the RESTful BonFIRE API to allow deployment and control of the experiment. The experimenter can develop the logic that they need on scripts and add very complex, runtime functionality, as allowed by Ruby and its powerful libraries. The Command Line Tools are a powerful way of scripting deployment and control. They are a Python-based toolkit that encapsulates the OCCI and exposes an intuitive interface that covers all aspects of the BonFIRE functionality.

Enactor

The Enactor shields the technical details of how to communicate with each specific testbed from the higher level RM. Once the RM has decided to perform an action on a testbed, the Enactor is in charge of transforming that request onto suitable format for the appropriate testbeds through a collection of adaptors. Adaptors where classified into four different categories: OCCI adaptors (that

[7]OGF Open Cloud Computing Interface Working Group, http://www.occi-wg.org/
[8]Restfully, https://github.com/crohr/restfully/blob/master/README.md

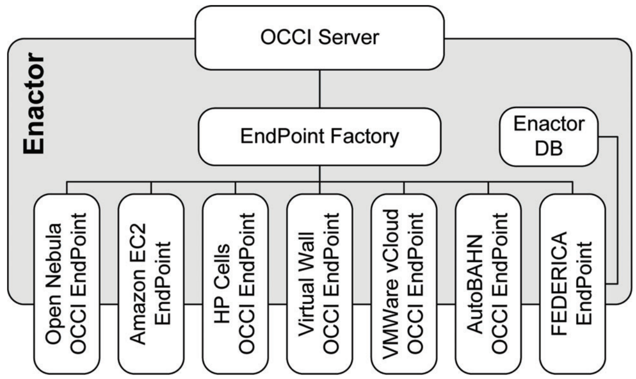

Figure 8.3 BonFIRE cloud and network infrastructure adapters.

are subdivided into five different types: OpenNebula, HP Cells, VirtualWall, and VMWare vCloud), Amazon EC2, AutoBAHN, and FEDERICA. It is possible to add other kind of adaptors outside those categories, making BonFIRE easily extendable.

The Enactor was not responsible for the security of the incoming call – but in counterpart it must enforce secure communication with the testbeds. The Enactor authenticates itself against testbed APIs (for example, by presenting a valid certificate, while user attributes are passed as HTTP headers – the testbed APIs can log/use them as they wish for auditing/accounting purposes). The Enactor supports multiple, concurrent, possibly time-consuming requests. It is a non-blocking service, capable of serving other requests while asynchronously waiting for a response from one of the testbed APIs.

Monitoring

BonFIRE provides its users with experiment monitoring facilities that support three types of metrics: VM metrics, application metrics and infrastructure metrics. BonFIRE provided this functionality through the use of the Zabbix open source monitoring software[9]. The Zabbix system adopts a client/server approach where the monitoring aggregator plays the role of the server and monitoring agents are the clients. Experimenters are free to deploy aggregators and agents in what in whatever way they wish but BonFIRE provides explicit support for the pattern where a single monitoring aggregator is deployed for each experiment. This aggregator collects data from several monitoring agents

[9] www.zabbix.com

deployed throughout the experiment and possibly also from infrastructure level aggregators deployed at each testbed.

The aggregator has been made available in the form of a dedicated virtual machine image containing an installation of the Zabbix monitoring software. This image is deployed like any other virtual machine image – no further configuration by the experimenter is required. The only requirement for the VM running the aggregator is that it must have an IP address that is reachable from the other VMs in the experiment and by the Resource Manager and Portal. This is necessary to enable the monitoring agents deployed on the individual machines to contact the aggregator and to enable the Resource Manager and Portal to expose the Zabbix API and web interface respectively.

A monitoring agent software is also included preinstalled within the images provided by BonFIRE. It needs to be configured with the IP address of the monitoring aggregator. This configuration is realized through the contextualization mechanisms of OCCI. After startup, the agent will register itself with the monitoring aggregator, from which point on the agent machine is fully integrated within the experiments monitoring system. The experimenter has the ability to further configure the agent by defining personalized metrics which should be evaluated and sent to the aggregator. This can be done through the standard mechanisms of the Zabbix software or via the contextualization section of a BonFIRE OCCI request.

The experimenter has multiple options on where to store the monitoring data of an experiment. The monitoring data can be stored either inside or outside the aggregator image. In the second option, the database of the aggregator is stored in an external, permanent storage that is mounted as an additional disk to the aggregator VM. This option enables more flexibility, the experimenter can set, on-demand, the storage size for the monitoring data, and this data is also available after the experiment's expiration or deletion. As a third option, the experimenter can use an external storage resource that was already in previously experiment. All these options are available through the BonFIRE Portal. By default the aggregator is created with an external, permanent storage with 1 GB size. As well as monitoring at the VM level, BonFIRE also supports monitoring at the infrastructure level. Those testbeds that support infrastructure monitoring have an infrastructure monitoring aggregator that gathers information regarding the whole testbed. An experiment aggregator fetches monitoring data of predefined, privilege metrics relating to those physical machines that host its virtual machines. The experiment aggregator fetches this data through the monitoring API.

Elasticity Engine

The Elasticity Engine supports three possible approaches for elasticity in BonFIRE: manual, programmed and managed. The BonFIRE components support manual elasticity by providing a Portal that allows various monitoring metric to be observed and the RM's OCCI API through which resources may be created or deleted. Additionally the architecture supports programmed elasticity via the Resource Manager's monitoring and OCCI APIs. This is done by the elasticity engine (EE) a stand-alone component able to manage the experiment based on some Key Performance Indicators (KPIs). It is basically a rules engine which can be configured via OCCI. It can be deployed inside a compute resource used by the experiment. In this way it is possible to create an elastic experiment using the portal, the experiment manager, or directly sending requests to the resource manager.

The basic functionality of the elasticity engine is to automatically increase or decrease compute resources in a running experiment. The experimenter has to pre-configure his own image using the SAVE_AS functionality. Once the image is ready he has to communicate this information to the elasticity engine which will deploy or remove compute resources automatically based on some rules expressed by the experimenter.

In order to distribute the load between different compute resources, the elasticity engine deploys a load balancer which is included in the BonFIRE standard images. The load balancer is part of the standard pool of images. It provides internally two different kinds of load balancer: HTTP and SIP. The first one is based on the open source HAProxy, with an additional HTTP interface for being managed remotely by the EE. The second one is based on Kamailio, an open source SIP proxy which offers also some functionalities of dispatching messages. Figure 16 shows an example of architecture of an elastic experiment.

CoCoMa: Controlled Contentious and Malicious Patterns

One of the main common characteristics of cloud computing is resource sharing amongst multiple users, through which providers can optimise utilization and efficiency of their system. However, at the same time this raises some concerns for performance predictability, reliability and security:

- Resource (i.e. CPU, storage and network) sharing inevitably creates contention, which affects applications' performance and reliability.
- Workloads and applications of different users residing on the same physical machine, storage and network are more vulnerable to malicious attacks.

Studying the effect of resource contention and maliciousness in a cloud environment can be of interest for different stakeholders. Experimenters may want to evaluate the performance and security mechanisms of their system under test (SuT). On the other hand cloud providers may want to assess their mechanisms to enforce performance isolation and security.

The Controlled Contentious and Malicious patterns (COCOMA) components provides experimenters the ability to create specific contentious and malicious payloads and workloads in a controlled fashion. The experimenter is able to use pre-defined common distributions or specify new payloads and workloads. VM images can be created that allow the injection of CPU, memory and disk I/O contention patterns to the physical host. COCOMA allows these types of contention to be combined and also allows variation of the intensity of contention across time. Still, all this control is not enough and affects other users on a multi-tenant physical host. To combat this, BonFIRE grant users exclusive access to physical hosts. This eliminates contention on the local disk, the memory and the CPU of the physical host, and combined with COCOMA gives BonFIRE users unique control across the whole range of zero to maximum isolation.

Networking

BonFIRE's multi-Cloud services has extensive support for controlled networking experiments. BonFIRE includes the Emulab-based [REF] Virtual Wall facility, which allows users to construct not only compute and storage resources, but also networks with user configurable bandwidth, latency and packet-loss characteristics. The user can modify these metrics at runtime, using BonFIRE's API or Portal. The Virtual Wall also allows users to inject background traffic to their networks and change the network buffering strategies. BonFIRE is also an early adopter of the GÉANT AutoBAHN pilot service of bandwidth on demand provision. AutoBAHN allows users to set up a point-to-point link with predefined bandwidth between two sites in its deployment. With the help of GÉANT, Janet and PIONIER, BonFIRE exposes this functionality to end-users that deploy their VMs on the EPCC and PSNC testbeds. Although it only allows control of bandwidth, AutoBAHN is more realistic than the Virtual Wall in that it involves real, rather than emulated network devices. In our experience, the key benefit of AutoBAHN for testers is not so much guaranteeing the quality of service, which is GÉANT's intended use, but rather policing it to within the limits of the user specification, so as to allow users to evaluate their system under known network conditions.

BonFIRE was committed to bridge the gap between advanced networking functionalities and the target cloud user community. To this end we enriched our interface to ease adoption of the network features. For example, AutoBAHN requires routing set-up on the newly created compute resources. BonFIRE exposes routing at the familiar, OCCI level, and provides simple directives as well as guidelines to declare routing on VM instantiation. This allows our users an easy, error-free way to specify routing without accessing the resource after it has been instantiated. Importantly, they get the network service without needing to go down to its level.

Experiment Data Provenance

An Experiment Data Manager (EDM) for Provenance (Prov) is used to describe the provenance of an Experiment, resources (compute, storage and network) within the experiment(s), any software/services running on the resources, any particular components as part of software/services, any users interacting with entities in an experiment. The EDM Prov will build upon the W3C PROV Data Model (PROV-DM)[10], which is a recent specification that stems from work on the Open Provenance Model (OPM)[11] with many existing vocabularies, applications and libraries/services. The PROV-DM core model allows extensions, such as subtyping (software agents running software). Other extensions for BonFIRE will be identified and made available to experimenters. PROV-DM model is very flexible, allowing experimenters to capture provenance of anything within their experiments. The model also supports bundles and collections of entities, allowing provenance of provenance. PROV-DM therefore offers a very powerful framework for experimenters to use in BonFIRE. The EDM Prov will comprise several components and will be made available in a VM image that experimenters can deploy as an optional service in BonFIRE. Other components in BonFIRE, like COCOMA, or services deployed by the experimenters on different compute resources may also generate provenance events, which need to be sent to the EDM Prov. To achieve this, the contextualisation functionality in BonFIRE can be used to provide those components with the IP of the EDM Prov, in the same way it is currently used for passing the Zabbix Aggregator IP to VMs with Zabbix Agents for monitoring.

[10]https://www.w3.org/TR/prov-dm/
[11]http://openprovenance.org/

Authentication, Authorization and Accounting

The authentication solution adopted by BonFIRE is based on existing state-of-the-art components such as Apache modules and Lightweight Directory Access Protocol (LDAP). To secure the connections between the components of the BonFIRE architecture server certificates are needed. These certificates are issued by the BonFIRE Certificate Authority (CA). The components behind the Resource Manager validate HTTP requests by using the BonFIRE Asserted ID Header field. These components trust the request from an authenticated user, because of the existing X-BonFIRE-Asserted-ID header field. The LDAP server and the BonFIRE CA are deployed on a VM with private IP address at HLRS. For security reasons access to that server is restricted. The BonFIRE CA is based on OpenSSL and the LDAP server for storing centralized information based on OpenLDAP.

The Authorization Service is used by the Resource Manager to control access to certain resource types and sites on a per-group basis. For example, the authorization service may restrict users in a group so that they can only use two named BonFIRE sites. Additionally, the Authorization Service also monitors current usage on a per-group basis and can be used to control the maximum amount of resources used by a group at any given time. The Authorization Service was added to support the degree of capability management that is required for BonFIRE open access phase.

The Accounting Service records all the usage of BonFIRE and can produce usage reports. These usage reports are essential to understand usage of BonFIRE with a view to informing sustainability decisions. The accounting reports were also envisaged as a precuser to any future billing system.

8.4 Federation of Heterogeneous Cloud and Networking Testbeds

BonFIRE offered a federated, multi-site cloud testbed to support large-scale testing of applications, services and systems. This is achieved by federating geographically distributed, heterogeneous clouds testbeds where each exposes unique configuration and/or features while giving to the experimenters (users) a homogeneous way to interact with the facility. BonFIRE supported five different types of Cloud testbed:

- OpenNebula: The currently operated OpenNebula version 3.6 includes an implementation of an OCCI server based on the OCCI draft 0.8. In order to provide valuable cloud functionality, additional fields of use were added by the BonFIRE developers in order to improve and extend the whole OCCI software stack of OpenNebula.

- HP Cells: The OCCI at HP Cells is completely stateless, so there is nothing that can get out of sync with the BonFIRE cental services or with the Cells state. BonFIRE-specific information such as groups, users, etc. are not stored, so the information retrieved on each request from the Enactor is filtered according to the permissions of the requesting user. This OCCI server was implemented specifically to support the BonFIRE project.
- Virtual Wall: The Virtual Wall emulation testbed is not a typical cloud environment, as it lacks the ability to dynamically add computes to an already running experiment. However, its functionality offers a first step to bridge the gap between network and cloud experimentation. The Virtual Wall offers the same OCCI resources as the other testbeds in BonFIRE, but their implementation is very different due to its underlying framework, Emulab. For instance, the Virtual Wall maps Compute resources to physical nodes, which prevents virtualisation, but allows the experimenter to take full control of the hardware. In response to the need of experimenters to share larger amounts of storage between different Compute resources, the Virtual Wall implements a notion of shared storage based on the Network File System (NFS).
- VMWare vCloud: vCloud does not offer by default an OCCI API. Similar to the case of HP Cells, an OCCI server was developed inside the BonFIRE project that interacts with the VMWare vCloud Director API to support VMWare Cloud facilities. The OCCI server is stateless, all the requests coming from the Enactor are translated and mapped to the proprietary API.
- Amazon EC2: The Amazon EC2 endpoint at the Enactor makes use of the API that Amazon provides to connect remotely to their Cloud services. The endpoint only allows to manage two kind of resources: storages and computes that are mapped to their Amazon equivalents, volumes or images and instances. In order to deal with the large volume of information returned, BonFIRE caches some OCCI queries in the Enactor, like listings of EC2's numerous storage resources.

BonFIRE supports experimentation and testing of new scenarios from the services research community, focused on the convergence of services and networks. In order to support network experimentation, BonFIRE is federated with the iMinds Virtual Wall testbed; and is interconnected with two network facilities: FEDERICA and AutoBAHN. The most distinctive features of the iMinds Virtual Wall are related to its networking capabilities. Whereas the other BonFIRE testbeds only provide a best-effort variant of the Network resource, the Virtual Wall implements three different types of Network resources: Default Networks that provide basic connectivity between two or more Computes; Managed Networks that provide controllable QoS (parameters that can be adjusted are bandwidth, packet loss rate and delay) over the network links; and Active Networks, that, on top of the functionality of Managed Networks, also provide the possibility to control the background traffic (UDP and TCP connections with dynamically adjustable packet size and throughput) on a network link. These networks provided by the Virtual Wall are emulated, using the Emulab software. FEDERICA is an infrastructure composed of computers, switches and routers connected by Gigabit Ethernet circuits. Through the Slide-based Federation Architecture (SFA) paradigm, FEDERICA offers to BonFIRE experimenters iso-lated network slices by means of virtualizing routers. This interconnection is aimed to help experimenters to investigate application performance through better control of the underlying network. The following changes were carried out in.

BonFIRE to incorporate these new network resources: the router resource was added to the BonFIRE OCCI and the network resource was enhanced with two new attributes: network link and vlan. Finally, since FEDERICA offers an SFA interface as federation API, it was necessary to implement an SFA endpoint at Enactor level. The FEDERICA SFA interface expects a unique XML request, where all the slice resources and their configuration are specified. This differs from the BonFIRE architecture, where each resource is requested in a single OCCI call. The main function of the BonFIRE SFA endpoint is to transform BonFIRE's OCCI information model to the SFA information model.

The federation between BonFIRE and the AutoBAHN beta-functionality offered by the GEANT facility allows the experimenters to request QoS guaranteed network connectivity services between VMs deployed on EPCC and PSNC testbeds. Overcoming the Best Effort limitation of the public Internet, dedicated network services can be established on demand for each experiment, with guarantees in terms of bandwidth, reduced jitter and service reliability. This option is fundamental to offer a controlled connectivity between VMs, so that the experimenters can evaluate the performance of their applications in environments able to emulate a variety of network conditions. In BonFIRE, a BoD service is represented by a new type of OCCI resource: the site link. Once the resource is created, it can be used to connect two networks created in the

BonFIRE sites at the edge of the site link: the traffic between the VMs attached to these networks is routed through the dedicated service. The processing of the OCCI requests for site link resources is managed at the enactor through a dedicated AutoBAHN end-point that is in charge of translating the OCCI specification into the AutoBAHN BoD service format. The Enactor endpoint acts as an AutoBAHN client.

8.5 Federation within the Broader FIRE Ecosystem

BonFIRE's infrastructure resources are only part of a highly complex and diverse Future Internet ecosystem consisting of infrastructure, services and applications. Through the EC FP7 FED4FIRE project[12], BonFIRE became part of a wider Experimentation-of-a-Service ecosystem offering access to heterogeneous Future Internet resources for experimentation such as cloud computing, wired and wireless networks, sensor networks and robotics deployed in laboratory and real world environments. The goal of FED4FIRE was to bring together European testbeds so that their resources may be used in a uniform manner by experimenters using their resources.

FED4FIRE has adopted a standardised protocol for resource reservation. The FED4FIRE federation performed a survey of its initial set of testbeds[13] and found that the most commonly used protocol for resource reservation and provisioning is the Slice-based Federation Architecture (SFA)[14]. Given that many of the federation's testbeds already supported SFA, plus the added advantage of compatibility with GENI testbeds, the SFA was adopted as the common protocol for the FED4FIRE federation, and tooling and guidance has been developed within the FED4FIRE project to support the SFA protocol, which testbeds can use to help them support the SFA protocol, thus reducing the cost of entry to the FED4FIRE federation for testbeds.

The "Slice" in the SFA is a client-side construct that is used as an identifiable container to collect resources from different provider in. The user may make a request to an SFA-compliant testbed, quoting their slice ID, and request that resources from the testbed be placed within the slice.

[12]http://www.fed4fire.eu/

[13]Vandenberghe, W., Vermeulen, B., Demeester, P., Willner, A., Papavassiliou, S., Gavras, A., Sioutis, M., Quereilhac, A., Al-Hazmi, Y., Lobillo, F. and Schreiner, F., 2013, July. Architecture for the heterogeneous federation of future internet experimentation facilities. In Future Network and Mobile Summit (FutureNetworkSummit), 2013 (pp. 1–11). IEEE.

[14]Peterson, L., Ricci, R., Falk, A. and Chase, J., 2010. Slice-based federation architecture (SFA). Working draft, version 2.

The FED4FIRE federation's choice of the SFA brings with it an access token format, the GENI Credential[15]. This enables users to use re-sources reserved in their slices, and owners of slices to grant access for other users to resources within the slice. In its basic form, the Slice Credential is a signed XML document containing the ID of the slice, certificate of the slice's owner and the ID of the slice. The Slice Credential also contains the rights the owner has on the slice, and whether the owner can delegate rights to others. There is another form of Slice Credential, the Delegated Slice Credential, and this enables the owner of a slice to grant permissions to other users on the slice.

BonFIRE had its own mechanisms for resource allocation and used different access tokens. Hence, a mapping had to be established between the BonFIRE resource allocation protocol and the FED4FIRE's chosen standard of SFA. Figure 8.4 shows the different concepts the SFA-compliant testbed and BonFIRE use. The slice is a container held by the user and is used to group resources from different testbeds together. In an SFA testbed, the user presents the slice and asks the testbed provider to allocate resources to it. In BonFIRE, the existing approach is to create an experiment at a testbed, which resources are allocated to (this is indicated by the dashed arrow in Figure 8.4). To enable holders of SFA slices to use BonFIRE a mapping between the slice identifier and a BonFIRE experiment was needed.

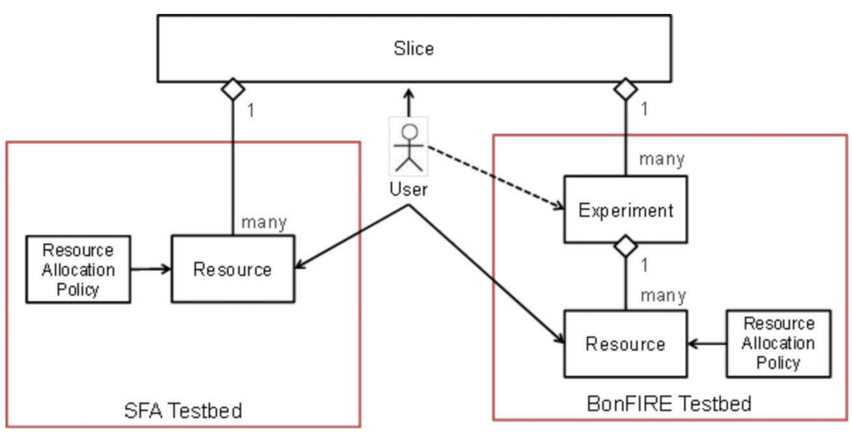

Figure 8.4 SFA-BonFIRE mapping.

[15]Available from http://groups.geni.net/geni/wiki/GeniApiCredentials

A Thin Aggregate Manager was developed that maps the BonFIRE experiment to a slice presented by the user. The existing components (specifically the Resource Manager) can continue to use the existing BonFIRE experiment ID. In use, the user also requests resources and presents their slice credential. The Thin Aggregate Manager requests an experiment be created by the Resource Manager, and the Resource Manager creates the experiment and allocates resources to it. The experiment and resource IDs are returned to the Thin Aggregate Manager.

8.6 Pioneering Open Access Experimentation and Sustainability

BonFIRE pioneered open access and sustainability of European experimentation services within the FIRE Ecosystem. In February 2013, BonFIRE launched the 1st Open Access initiative providing free access to both commercial organisations, academic institutions and other European projects outside of the BonFIRE consortium. Open access was developed as part of BonFIRE's sustainability activity as it transitioned through distinct operational phases on its route to a service offering beyond the lifetime of the project. Each phase had a distinct financial model that influences the governance and decision making of the experimentation services, and importantly the relationship with experimenters (Facility Users) as shown in Figure 8.5. The effect was that BonFIRE was no longer driven by the needs of a funded research project but by the features demanded by the experimenters external to the consortium. This was an important step towards an operational experimentation facility concerned with efficiency, accountability and customer satisfaction.

The lifecycle phases in BonFIRE's strategy are described below:

- **Pre-project conceptualisation**: concerned with defining the concept of a social and network media facility and getting buying from all stakeholders. This includes primarily supplies of services, technologies and other assets such as venue operations, technology providers and initial investors. The result of this phase is a public funded Project to implement the facility.
- **Project driving experiments**: concerned with implementing the facility in terms of technical and operational aspects. There are no Facility Users but Driving Experiments that define requirements and testcases to validate the facility offerings. The result of this phase is the 1st operational facility available for Facility users.

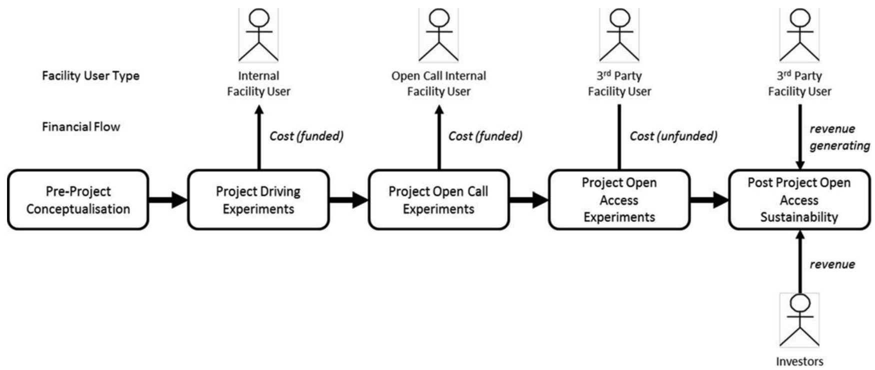

Figure 8.5 Transitions in governance and experimenter relationships.

- **Project open call experiments**: concerned with selecting and executing a set of experiments funded by the facility. The Facility Users are paid to run experiments and are acceded to the project contract with the facility providers. In return for payment Facility Users help facility providers understand how to improve the service offering by testing software and operational policies. The result of this phase is an enhanced facility that has been tailored to meet the needs of users.
- **Project open access experiments**: concerned with selecting and executing a set of experiments that are not funded by the facility. The Facility Users must pay their own costs and are not acceded to the project contract. Facility Users are therefore 3rd parties and access to IPR where needed must be governed by an appropriate license. Further legal agreements may be necessary to attribute rights, responsibilities and legal liability. Allowing 3rd party access allows the project to understand the legal and operational requirements required for post project facility use. The phase does not cover the mechanisms for revenue generation but unfunded experiments do provide test cases for simulating future business models including costs and revenues.
- **Post-project sustainability**: concerned primarily with continuing facility services. Exploitation agreements between partners were established to define how BonFIRE foreground can be used and post project governance structures implemented. Project partners must align themselves with operational roles and commit appropriate levels of resources to sustain activities.

BonFIRE successfully managed the transition between the different phases was a key factor in the success of the project. Each has placed demands on governance in terms of technical and operational requirements for the facility. For example, transitioning from open call to unfunded experiments requires the project to deal with access to the facility by third parties. During the final year the project has been concerned with the transitions from experiments funded by "Open Call" to experiments using BonFIRE through "Open Access" agreements. Finally in December 2013 the BonFIRE Foundation was established to operate the BonFIRE multi-site Cloud testing facility beyond the lifetime of the project, which continued operations until May 2016 some 18 months after the initial funded research project. The BonFIRE Foundation comprised members from world-leading industrial and academic partners, dedicated to continue to deliver services that enable developers to research new, faster, cheaper, or more flexible ways of running applications with new business models.

The BonFIRE Foundation was highly successful hosting over 50 experiments addressing a range of cloud computing challenges and through participation in the Fed4FIRE Federation BonFIRE has supported a further 11 experiments. Table 8.1 describes a few highlights from open access experiments.

The 11 Fed4FIRE experiments have used BonFIRE and finished their work successfully. Highlights included IPCS4FIRE focusing on the orchestration of cloud and user resources for efficient and scalable provisioning and operations

Table 8.1 Example open access experiments

Experiment	Description
MODA Clouds Alladin (Atos)	Atos Research and Innovation, Slovakia, are investigating a multi-Cloud application in BonFIRE that delivers telemedicine health care for patients at home. The application provides an integrated online clinical, educational and social support network for mild to moderate dementia sufferers and their caregivers. The aim of the experiment is to analyse the application behaviour in a multi-Cloud environment and improving its robustness and flexibility for peak load usage.
Sensor Cloud (Deri)	Digital Enterprise Research Institute (DERI) at the National University of Ireland, Galway, came to BonFIRE for testing scalability and stability of a stream middleware platform called Linked Stream Middleware (LSM, developed for the EC-FP7 OpenIoT and Vital projects). The experiment in BonFIRE utilises multiple sites with sensors generating up to 100,000 streaming items per second consumed by up to 100,000 clients. The data processing modules such as data acquisition and stream processing engines are run on the BonFIRE cloud infrastructure.

(*Continued*)

Table 8.1 Continued

Experiment	Description
SWAN (SCC)	This is an experiment conducted by SSC Services to analyse how one of their software solutions, SWAN, can handle large amounts of data transferred between business partners under different networking conditions. SSC Services have utilised the iMinds Virtual Wall site to achieve fine-grained control of the networking conditions in order to identify critical Quality of Service (QoS) thresholds for their application when varying latency and bandwidth. Moreover, investigating possible actions and optimisations to the SWAN components to deal with worsening conditions, to be able to deliver the expected QoS to the business partners.
ERNET	ERNET India are developing software for moving e-learning services into the Cloud and are using BonFIRE to analyse the benefits of Cloud delivery models, including multi-site deployment. In particular, they investigate fault tolerance.
JUNIPER	BonFIRE also facilitates other research projects, giving access to multiple partners to perform an experiment. One of these projects is the EC-FP7 project JUNIPER (Java Platform for High-Performance and Real-Time Large Scale Data), which deals with efficient and real-time exploitation of large streaming data from unstructured data sources. The JUNIPER platform helps Big Data analytic applications meet requirements of performance, guarantees, and scalability by enabling access to large scale computing infrastructures, such as Cloud Computing and HPC. In JUNIPER, the BonFIRE Cloud premises are used to initially port pilot applications to a production-like Cloud infrastructure. The JUNIPER experiment benefits from the availability of geographically distributed, heterogeneous, sites and the availability of fine grained monitoring information (at the infrastructure level) to test and benchmark the developed software stack. Another important advantage of BonFIRE to JUNIPER is that some of the sites owning HPC facilities, e.g., HLRS (Stuttgart), provide a transparent access (bridge) from Cloud to HPC, which is of a great importance for JUNIPER experiments.

of security services. As a result of their experiment, IPCS4FIRE were able to explore best-practices and share the optimal design with users to automatically provision and protect virtual machines without manual intervention, while minimising the time required to achieve this protection. SCS4FIRE performed experiments on the validation of Secure Cloud Storage system for multi-cloud deployments. SCS4FIRE optimized their methodology to automate the transfer of virtual machines and encrypted data volumes between multiple cloud sites, while maintaining continuous access for end users.

Finally SSC researched big data analysis components on Smart City data using cloud resources. SSC were able to validate that their Super Stream Collider middleware can achieve high scalability, continuous accessibility and high performance, for more than 100.000 clients.

8.7 Conclusions and Outlook

From Sept 2010 until May 2016, FIRE experimentation ecosystem has incorporated the BonFIRE multi-Cloud experimentation facility alongside testbeds in the networking, sensors and smart cities. The BonFIRE facility was unique in supporting services and network experimentation across multi-cloud sites focusing on a blueprint for experimentation and incorporating methodology and techniques to support repeatability and reproducibility. BonFIRE took these notions further, to deliver a facility based on four pillars: observability, control, advanced features and ease of use for experimentation. The end result was a facility that differed substantially from public Cloud offerings. Public Cloud providers will never offer the internal tracelogs and parameters of the clusters since it is highly sensitive data for their business, whereas this information is essential in research by experimentation to understand the behaviour of the Cloud applications. Also, public Clouds did not offer detailed level of control over physical and virtual resources, since their objective is to hide the complexity and operation from the users and reduce costs. Advanced features, such as user-specified bandwidth on demand and controlled networks were greatly received by the services experimenters, but are not in line with public Cloud offerings, while domain-specific tooling for experimentation is naturally not a concern. BonFIRE was funded between 2010 and 2013 and continued to be operated by the BonFIRE Foundation.

There are many emerging opportunities and requirements for Cloud-based experimentation facilities in the future driven by the needs of applications and services communities, and the ongoing convergence of software defined infrastructures. We see two major areas of expansion: embracing Big Data and enabling Mobile scenario testing. Researchers are exploring how to deal with the characteristics and demands of data within services, infrastructures, sensor networks and mobile devices, while the uptake of smartphones motivates the combination of mobile networks and Cloud computing. It is necessary to cover the full data lifecycle across multiple experimentation platforms facilities providing the necessary data interface, format, optimized transfer mechanisms, data analytics and management toolset to extract value from experimental data.

On the other hand, as the data traffic demand from mobile phones and tablet applications is exponentially growing (e.g. video, VoIP, Gaming and P2P) networks are developing to offer more capacity , higher throughput and better QoS. Future 5G networks and concepts dominate the research arena. Many telecom operators and network equipment manufacturers are embracing Network Functions Virtualization (NFV) techniques since it is envisaged that this will change the telecom industry landscape. Industry in ETSI is doing a great effort with the first sets of specifications and the "traditional" Cloud community has a lot to offer to the "virtualisation and softwarisation" of networks. Notably, this is a central research topic in the 5G PPP initiative where large-scale validation of these network virtualisation techniques are expected and experimentation platforms can play a role. A key lesson learnt from BonFIRE is that there is great value to be had from offering high-level interfaces for experimentation. Experimentation as a Service is a fact, not an endeavour, and the only way forward is to offer a truly PaaS tooling environment for experimenters on top of the IaaS layer, no matter what this infrastructure is.

Six years after the project kick-off, BonFIRE concluded its successful journey on 30 May 2016. In this period BonFIRE delivered impact consistently, breaking new ground in experimentation platforms and service delivery models across both technical and sustainability fronts. Open Access was highly successful with new and returning users, like EC FL7 RADICAL project renewing its Open Access for a third year and BonFIRE supporting the project right up until RADICAL's final review. Utilisation was high, with EPCC and Inria at times completely full and oversubscribed. The stability of the infrastructure has been remarkable, with two short, unplanned outages, both down to external factors. The services have now been decommissioned and no further access will be possible but the legacy of the BonFIRE initiative has provided a pioneering blueprint for current and future experimentation-as-a-service platforms exploring Next Generation Internet technologies.

Acknowledgements

The authors would like to thank all partners in the BonFIRE consortium.

The BonFIRE project has received funding from the European Commission as part of the FP7 work programme.