Unified Representation of Monitoring Information Across Federated Cloud Infrastructures

Yahya Al-Hazmi Jose Gonzalez Pablo Rodríguez-Archilla

Federico Alvarez T. Orphanoudakis, P. Karkazis Thomas Magedanz

Abstract—**Nowadays, one of the issues hindering the potential of federating cloud-based infrastructures to reach much larger scales is their standard management and monitoring. In particular, this is true in cases where these federated infrastructures provide emerging Future Internet and Smart Cities-oriented services, such as the Internet of Things (IoT), that benefit from cloud services. The contribution of this paper is the introduction of a unified monitoring architecture for federated cloud infrastructures accompanied by the adoption of a uniform representation of measurement data. The presented solution is capable of providing multi-domain compatibility, scalability, as well as the ability to analyze large amounts of monitoring data, collected from datacenters and offered through open and standardized APIs. The solution described herein has been deployed and is currently running on a community of 5 infrastructures within the framework of the European Project XIFI, to be extended to 12 more infrastructures.**

I. Introduction

The significance and the success of cloud computing has gained more attention by the ICT sector in recent times. With the increasing number of commercial cloud providers and their broad diversity of offerings, as well as the number of worldwide research activities working on cloud federation and interoperability (e.g. [1] [2]), cloud federation is recognized as a tremendous value for the industry.

However, cloud federation may differ in the approach and in practical terms. In this paper, we follow the approach of the European-funded project XIFI [3] that is based on the concept of establishing a federated community cloud, targeting different stakeholders, which are not falling under the conventional cloud services consumers, but also taking into account the benefits arising from the concept of emerging Future Internet and Smart Cities services (e.g. IoT) that benefits from cloud services.

Two of the most critical issues for boosting cloud federation towards much larger scales and accommodate the requirements of various communities are the management and monitoring of the federated infrastructures. Powerful and convenient tools offered through common Application Programming Interfaces (APIs) are required to support the management and monitoring of the involved infrastructures and services, as well as to provide the offerings to the different involved user communities in a common and standard manner.

Monitoring control performance is a fundamental part of every single infrastructure. The infrastructure providers require to monitor their resources (physical and virtual) to ensure their operational and availability status, while end users may be interested on monitoring their allocated resources, evaluate performance and validate Service Level Agreements (SLAs). In a federated environment, some federation services require as well monitoring information necessary for their functionalities.

The state-of-the-art solutions for cloud monitoring mainly targets homogeneous, single-domain infrastructures. Nevertheless, the trend is moving towards federated cloud architectures [4] [2]. Due to the high heterogeneity of such environment regarding resources, tools, and even legal terms, the concept of federating monitoring systems should tackle a number of challenges, including common API and uniform data model. Thus, all the requirements need to be considered to overcome procedures and develop tools to support the monitoring of several resources and services across the federated infrastructures.

The contribution of this paper is the introduction of a unified monitoring architecture for federated cloud infrastructures accompanied by the adoption of a uniform representation of measurement data. It is based on a set of adapters responsible for providing monitoring data in one single format collected from heterogeneous sources that might have different APIs and data formats. This architecture is adopted and validated within the XIFI pan-European federation [3] that aims to be the community cloud for European FI-PPP developers enabled by advanced Future Internet infrastructures in Europe.

II. Related Work

There are currently many suitable solutions for cloud monitoring. Examples include Nagios [5], Zabbix [6], GroundWork [7], CloudStatus [8], CA Nimsoft Monitor [9], EVEREST [10], MonALISA [11], and Ganglia [12], mainly targeting homogeneous, single-entity administered cloud infrastructures. Moreover, several monitoring architectures targeting cloud

management and monitoring have been proposed in the literature [13] [14] [15] [16] [17] [18]. These address the monitoring of cloud environments but cannot be applied to federated ones where a large number of resources from heterogeneous infrastructures are offered to customers.

The European project RESERVOIR has developed a monitoring solution that fits to its needs [19], partially supporting the cloud federation aspect. It provides information about services deployed in federated clouds for service management purposes. Yet, this solution does not consider providing monitoring data to cloud customers. In contrast, the Amazon monitoring system CloudWatch [20] provides monitoring data to customers regarding their running services, rather than providing data for infrastructure and service management. However, the monitoring solution presented in [4] operates across federated clouds but it is limited to datacenters monitoring. This solution requires all infrastructures to use the same monitoring tool.

Similar to our work, the Fed4FIRE federation [21] is currently working on a monitoring solution that aims at operating across federated testbeds that are not cloud specific. This work is still in its initial stage and has not been validated.

III. Problem Statement

Precise, in-depth and timely measurement and monitoring information has to be provided in a uniform representation across the federation to support multiple stakeholders. Information is useful for different reasons such as federation services to check the availability status of an infrastructure and to ensure end-to-end network and service performance, for SLA managers to validate SLAs, for datacenter managers to control the usage of resources, for developers to understand the behavior and performance of their services or applications.

To provide cross-layer monitoring in cloud infrastructures, various tools are used depending on where the measurements take place. Such tools are used to monitor infrastructure resources (physical and virtual), network connectivity performance, as well as running services and applications. However, independently administered infrastructures use diverse systems, which handle different data formats, APIs and databases. Consequently, the heterogeneity of the federated environment will result in providing unfeasible common dataset.

In order to provide the required data in a uniform manner and maintain the use of pre-existing tools that were already in place at the cloud infrastructures, there is need for a solution in charge of standardizing the format and access to the performance data of the federation. The solution shall take into consideration further technical aspects. Remarkable examples include the need for standardized interfaces federation-wide, and to deal with the rapid and dynamic changes in the federation, as the number of measurement points and diverse monitoring tools will increase. Furthermore, a well-structured data model is needed in order to provide monitoring data in a meaningful way. The collection of the data on infrastructure level must be independent from installed monitoring tools. The data needs to be provided i) on-demand (on user request) and ii) on-schedule to provide the information with the time interval required (even real-time). Monitoring components should be compatible and interoperable with the rest of the federation architecture.

IV. Own Approach

This section presents the approach we followed in the XIFI project to address the issues stated above and support the relevant functional and technical requirements. Our approach is based on the concept that each infrastructure shall adopt a specific set of adaptation mechanisms in order to become part of the federation and offer compatible services. It is therefore mandatory to define those mechanisms that allow an infrastructure to offer the full capacity of the federation services in a unified manner, achieving the necessary degree of compatibility with the rest of participating infrastructures, ensuring the highest possible quality to the end-users.

In this perspective, we define a set of adapters which are in charge of providing an abstraction layer to the monitoring system of each infrastructure. We refer to this abstraction layer as the Infrastructure Monitoring Middleware (IMM). IMM aims at providing homogeneous monitoring services by collecting and handling network and datacenter performance data from multi-domain infrastructures. In other words, considering the variety of monitoring solutions deployed by the infrastructures participating in the federation, the IMM defines the abstraction layer responsible for collecting, standardizing and publishing the multi-domain measurement results.

According to the outcomes of the study presented in [22], the most suitable approach to address a multi-domain measurement framework is to distribute the load of data among the different domains instead of processing the measurements from a central entity. This assessment must be subject to the characteristics of multi-domain networks and the impact of monitoring performance. Taking this guideline as a starting point, we proceeded to define the basis of the middleware that aims at orchestrating the monitoring systems across the federation.

Following a top-down approach, the IMM adaptation mechanism will be composed of a set of IMM instances distributed along the federation, settled in those infrastructures from where we require measurement data. An instance comprises of several adapters with different functionalities. The more instances are deployed, the more fine-grained the global monitoring will be, but with the cost of higher load and installation requirements, thus an optimization is required. IMM instances are the actual adaptation mechanism by following a distributed scheme, being the entities in charge of interacting with the different monitoring systems. However, they need an entity in a higher level of the federation in charge of aggregating and filtering the incoming data provided by each instance, querying when an on-demand request is made. This entity can be deployed in a distributed manner at the infrastructure level or as one single and centralized entity at the federation level.

V. The Architecture

A. Architectural Principles

This section presents the main architectural principles that have been considered during the design specification phase.

1) Multi-domain Compatibility: An administrator is able to monitor and control the status and performance of the infrastructure of a single domain in the manner that better fits that particular case. The real challenge comes when multiple domains get involved in a collaborative manner. By unifying the measurement systems of a set of differently administrated infrastructures, the solution proposed in this paper aims at providing a general picture of the environment, acting as if the performance data was being provided by a single domain. Considering the variety of measurement systems, the IMM will be in charge of collecting, standardizing and publishing the measurement results of this multi-domain environment.

2) Scalability: Scalability is another important architectural principle to be taken into consideration. From the federated cloud infrastructure's viewpoint, scalability issues arise from the collection of enormous amount of data from hosts (physical/virtual), network entities and services provided. Thus, the monitoring solution is able to store, retrieve and filter data to fulfil user queries, by separating recent and past data.

3) Ability to Analyze Large Amount of Data: The use of Big Data techniques in the proposed architecture let us cope with huge amounts of data generated. Indeed, as data size increases, the horizontal scaling allows storage, while the computational time needed to perform the data analysis keeps a linear growth, thus making such analysis feasible (e.g. within a tolerable processing time). By organizing such data in datasets, we can take advantage of data warehouse tools that not only facilitate querying such data, but also implement some logic to perform several aggregations and data analysis on behalf of the user.

4) Open APIs: Open APIs is the key to deal with heterogeneity and foster interaction between different stakeholders. Such APIs are the enablers for infrastructure providers and users to be part of the monitoring ecosystem and mobilize others to join, therefore contributing to build a wider community.

B. Architecture Overview

The architecture is able to cope with the control, management and request of the monitoring data. Some monitoring services can be requested by the user (on demand) or to run continuously, irrespective of the user requests (on schedule). The service access is controlled at federation level by an OAuth-based identity management (IdM) system, preventing unauthorized users to gain access to sensitive information.

The architecture is divided into different layers as illustrated in Fig. 1, where each layer is encompassing a separated functionality. The lower layer corresponds to the set of monitoring systems of the different infrastructures, which perform monitoring measurements within the same or among multiple infrastructures. The IMM has been designed with the scope of inter-connecting these single-domain systems with a common architecture. This defines the first abstraction mechanism in the proposed unified multi-domain monitoring framework which will report information to the upper Federation Monitoring Layer (FML). Sequentially, data are pushed to the FML from the IMMs where it provides a common framework for storing, aggregating and publishing the common monitoring dataset.

FML subscribes to a subset of relevant performance metrics of the federation (those which are defined in the common data model). FML elaborates the collected standard data, provides historical support and persistence, and offers some aggregation functions on the data to users via a User Interface. Potential users are: the end-users, the SLA manager, the recommendation tools that allow users to find the right services offered by the federation, and the Infographics and Status Pages that provides information on the infrastructure capacities and status services.

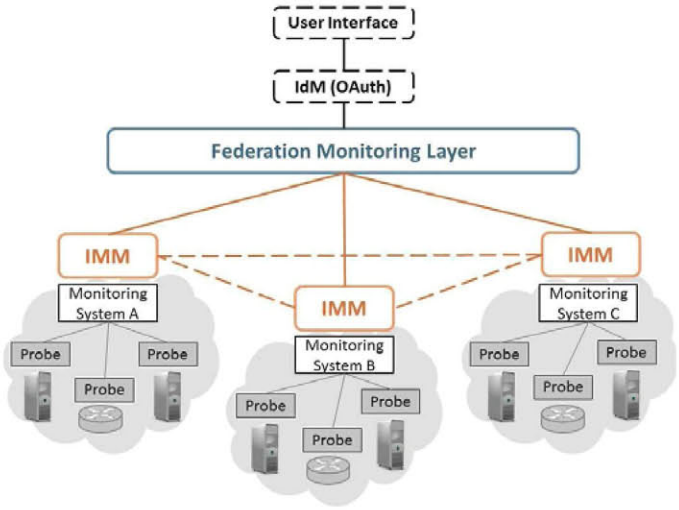

Fig. 1. High-level Deployment Overview of the IMM

C. Use of the Monitoring Services

We have grouped the concerned monitoring services into four main categories. These are presented as follows.

1) Monitoring Inter-domain Connectivity: It is mandatory to provide information about the connectivity between federated infrastructures. This assessment may be relevant either for checking the status for troubleshooting purposes or strictly from a service-oriented viewpoint. The unified monitoring framework enables to establish end-to-end performance tests along distributed infrastructures to check bandwidth and latency. To ensure inter-domain feasibility, this service follows a joint design and obeys the mentioned architectural principles.

2) QoS and SLA Assurance: Connectivity monitoring service relies on the capacity to inject test packets and follow them to measure the service provided. The volume and other parameters of the introduced traffic are fully adjustable: what implies testing, what is required, when it is needed. This emulation of scenarios will enable to check if Quality of Service (QoS) and SLAs are accomplished according to the real data obtained.

3) Distributed Datacenter Monitoring: In a federated infrastructure, a user may have access to and allocate resources from multiple cloud infrastructures. Thus, deployment of cloud resources, in particular Virtual Machines (VMs), can take place on distributed datacenters. The monitoring solution is capable of transparently deliver to the end-user monitored data, irrespectively of the underlying conditions.

4) Adaptation of Multi-Source Data: Given the heterogeneity of monitoring tools in the federated infrastructures, an adaptation service is needed in order to transform custom data into a common, flexible format. It is likely that the list of tools will extend, thus requiring the adaptation service to be generic enough to easily accommodate new sources of data.

VI. Implementation and Validation

The monitoring architecture described in this work has been adopted by the XIFI Federation [3]. By the time of writing, such community includes 5 infrastructures (called nodes in XIFI), distributed across 5 European countries, while 12 more nodes are under the integration process. The 5 core nodes (Berlin, Lannion, Waterford, Seville, and Trento) have different capacities, all together offering around 1,352 computing cores with 2,112GB of RAM and 348TB of storage capacity. The XIFI federation connectivity is based on a layer-3 Multi-Domain Virtual Private Network, provided by GÉANT [23]. This network is used by XIFI services to manage and monitor the users' resources as well as exchange information among nodes. The XIFI architecture aims to be a robust framework, providing high-availability. Based on these main requirements, the XIFI federation architecture is based on a deployment configuration that distinguishes between two types of nodes: `master` or `slave`. According to the initial set of 5 nodes, two of them hold a `master` role whereas the other 3 are stated as `slaves`. A slave node is the node where only the required XIFI software for deploying and managing user services is installed. This software comprises three main functional groups, which enable cloud computing (OpenStack-based), monitoring and security functionalities. Installation of this software is mandatory by each node willing to become part of the XIFI federation. A `master` node is the one where, in addition to the software deployed on the `slave` nodes, the centralized parts of the federation services (i.e. components needed to manage the federation) are deployed.

A. XIFI Monitoring Architecture

As aforementioned, this paper introduces a new method for uniform monitoring data representation. The Open Mobile Alliance (OMA) Next Generation Service Interface (NGSI) [24] standard plays a major role in our solution, providing the basis for an adaptation layer that implements mechanisms to normalize heterogeneous information into a common, context-based, entity-centric data format. The adaptation layer, namely the IMM, is implemented in XIFI and denoted by XIFI Infrastructure Monitoring Middleware (XIMM). It is deployed in a distributed manner along the federation, where a XIMM instance is settled in an infrastructure. Fig. 2 illustrates 3 XIMM instances that follow the same architecture and are installed in three different infrastructures. Each XIMM instance consists of four components described in the following sections.

B. NGSI Adapter

In order to fulfil the need of adaptation of multi-source data into a common format, IMM instances include the

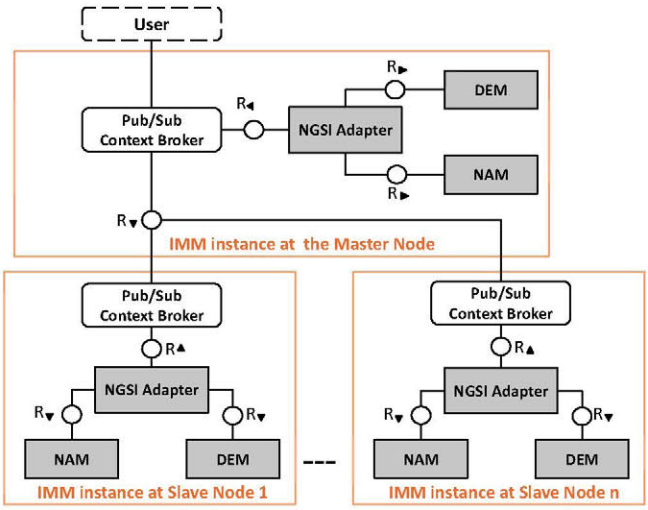

Fig. 2. IMM Architecture (3 instances deployed in a master & 2 slave nodes)

NGSI Adapter component. It is responsible for adapting raw monitoring data into NGSI context information, and updating such information into the Context Broker. NGSI Adapter itself is completely agnostic to the monitoring tools used to gather data, and the exact format of the data originated by their probes. It delegates the concrete adaptation into an extensible set of dynamically loadable `parsers`, which receive raw data from the probe (sent to NGSI Adapter as part of a HTTP request) and return the corresponding NGSI context. Thus, the adapter is as much independent from other IMM modules (NAM, DEM, etc.) as possible, only requiring specific parsers for the probes used by them. Therefore, a single NGSI Adapter is able to handle any kind of data coming from the different probes. Fig. 3 provides a graphical description of the interactions among internal and surrounding modules of NGSI Adapter.

NGSI Adapter asynchronously processes all incoming adaptation requests, so that data collectors (XIMM modules) are not blocked. Besides, possible temporary connection errors when issuing update requests to Context Broker are handled through an exponential backoff retry policy.

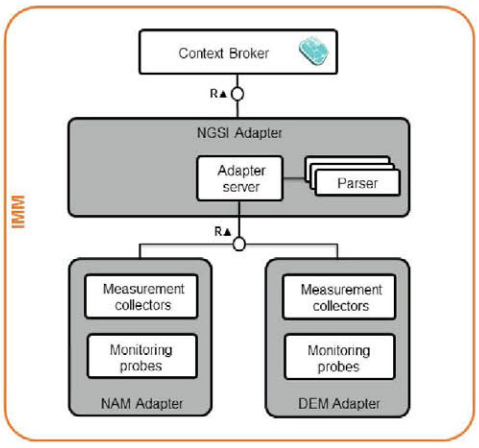

Fig. 3. Monitoring Data Adaptation

C. Network Active Monitoring Adapter

Network Active Monitoring (NAM) Adapter is the component in charge of handling cross-domain active measurements between the XIMM instances, providing a standard multi-domain monitoring mechanism able to handle latency and bandwidth-related tests. This implementation is the basis for monitoring connectivity among federated nodes. Monitoring data can be obtained by either on-demand or on-schedule requests. Historical measurements represent results of regularly scheduled tests and cover one-way delay, jitter, one-way packet loss, and achievable throughput for a path. Nevertheless, NAM Adapter also offers the possibility to request an on-demand measurement of achievable throughput or one-way latency measurement between endpoints. As depicted in Fig. 3, a NAM Adapter comprises two main modules, Monitoring Probes and Measurement Collectors, and interacts with NGSI Adapter.

Monitoring Probes: tools used to perform measurement tests between given infrastructures. Probes provide the Measurement Collectors with the raw network active monitoring data. The interface to interact with these tools is command line-based. To assure reachability, NAM implementation requires the inclusion of a pair of probes by default:

- `One-Way Delay (OWD) Monitoring Probe`: manages OWD tests. Leveraging on PerfSONAR's OWAMP [25] service, NAM's OWD Probe overcomes some existing functional requirements, which in terms of efficiency are not optimal, to enhance the operability.
- `Bandwidth (BDW) Monitoring Probe`: following the Internet 2's PerfSONAR distribution with regards to bandwidth tests (BWCTL [26]), NAM's BDW Probe is based on the network throughput tool Iperf [27].

Measurement Collectors: they are the core modules of the adapter responsible for collecting the data generated by the probes, processing and forwarding it to the upper layer via a REST-based Web Service API. Same as the probes, there are two types, OWD and BDW, according to the data they are required to handle. A collector consists of the following sub-modules. `Command Interpreter` is in charge of dealing with the probe via command line-based operations. `Format Parser` adjusts the result obtained from the command to a standard response, e.g. JSON or XML format. `Scheduler` is responsible for the timing in scheduled tests, triggering the process when the setup time is reached. `HTTP Server` handles the exchange of request/response. `Controller` is the central entity that manages the rest of components.

The NGSI Adapter is not intended to request directly the adapter (data flow is bottom-up). However, the User Interface can leverage the API to trigger an on-demand test.

It is possible to deploy this software component in a VM. Nevertheless, this configuration may carry accuracy and stability problems with the obtained values. Hence, it is strongly recommended to install the component in a physical resource. Each node in the federation shall assure the presence of at least one instance of this component to be reachable by other nodes. A node owner could deploy more instances to provide a more fine-grained status and avoid single points of failures.

D. Datacenter Monitoring Adapter

The Datacenter Monitoring (DEM) Adapter is responsible for collecting and publishing monitored data from physical and virtual servers. Thus, this adapter is of major interest for both infrastructure owners and end users since it enables one to check host resources, such as processor load, RAM utilization, disk usage and more. From an architectural viewpoint, DEM Adapter stands as the 'glue' between the source of monitoring information (monitoring probes and plugins on physical and virtual servers) and the NGSI Adapter that will post these data to the Context Broker entity, providing homogeneous representation of the data (based on a defined data model).

Given that the majority of the 5 core XIFI nodes are already using Nagios [5] as the monitoring solution and moreover Nagios is one of the de facto industry standards, widely used for monitoring infrastructures, the XIFI reference implementation of the DEM adapter is based on Nagios Core and NRPE plugin. Nevertheless, XIMM architecture is independent from specific monitoring tools, allowing for integration of monitoring data arising from other well-known monitoring tools (such as Zabbix [6], OpenNMS [28], etc.) with the minimum effort.

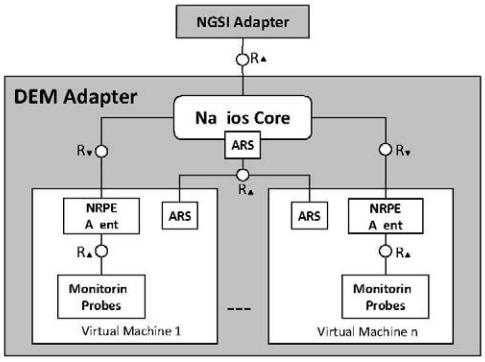

Fig. 4. DEM Adapter Architecture

As depicted in Fig. 4, a single instance of Nagios Core (server) is installed (either in a physical or a virtual server) in each infrastructure, being responsible for the registration and the collection of monitoring data from several VMs within the infrastructure. On the other hand, in each VM to be monitored, the proper tools are installed (namely NRPE, monitoring probes and Auto-Registration Service (ARS) module). Apart from the modifications and extensions required to NRPE plugin and monitoring probes (as provided by the Nagios Community [5]), there was a need for an automated registration of each VM to be monitored on Nagios Core. To fulfil this need, the ARS customized module has been developed that is capable of registering to Nagios Core a newly deployed VM in an automated way, requiring no intervention from the user. Similarly to NAM, DEM provides monitoring data that can be obtained by either on-demand or regular requests.

A DEM adapter is installed in each node, along with an instance of NGSI Adapter and Context Broker. An instance of

these is deployed on the master node for federation purposes.

E. Federation Monitoring

Federation Monitoring acts as a common framework for storing, aggregating and publishing the monitored data collected by the different adapters provided by the XIMM. It is distributed across all the nodes of the federation and elaborates monitoring data leveraging on big data analysis techniques. Besides, it provides historical support and persistence and offers some aggregation functions on the data. Fig. 5 represents an overview of the Federation Monitoring architecture. Data collection at the master node is accomplished by subscribing its Context Broker to the corresponding Context Brokers at the slave nodes, so that update notifications are received by the former on any context change at the latter. Apart from a Context Broker instance, master node comprises a BigData platform to write such data to a scalable and distributed storage (concretely, Hadoop Distributed File System (HDFS [29]) and run several MapReduce jobs for data aggregation and elaboration. Retrieval of aggregated data is done through querying tools (e.g. data warehouse solutions like Hive [30] or Spark [31]), and exposed to clients through a REST Query API. The architecture provides access to monitoring data via well-defined APIs and web-based GUI, where a user may retrieve data related to their deployed resources. To accommodate specific user's needs, monitoring dashboard includes three distinct roles, federation manager, infrastructure owner and simple user, with different privileges. For user authorization and authentication, an OAuth -based identity management system is used across the XIFI federation.

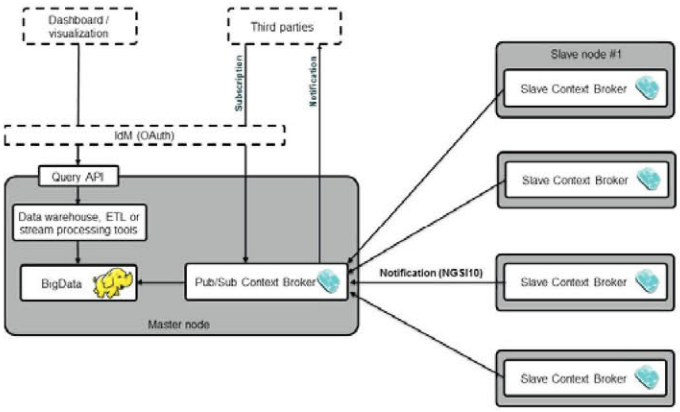

Fig. 5. Federation Monitoring Architecture

VII. Conclusion and Future Work

In the paper we presented a solution for the monitoring of federated cloud-based infrastructures with a holistic approach, addressing the different needs of both the operators of the federation and the users, concerning various required monitoring services. The proposed solution covers, in a flexible and scalable approach, both network measurements and datacenter monitoring requirements by the definition of a unified data model to gather and analyze the monitoring results. As none of the state-of-the-art solutions can provide a wide range of measurements across different domains as a stand-alone, the innovation of our solution is the composition of multiple solutions to provide a multi-domain monitoring system as well as adoption of solutions from other domains such as the NGSI from OMA. All components of the solution have been implemented and deployed in an initial federation of 5 infrastructures connected via a high-speed network within the context of the European Project XIFI. Early outcomes show the usefulness and compliance with the federation needs, as shown in the infographic and status pages of the XIFI portal. Detailed evaluation results, together with a set of respective measurements are left for future work. Furthermore, it is already planned the deployment of such architecture in 12 more infrastructures across Europe. This will then allow us to evaluate the solution performance at larger scale.

References

[1] D. Gracia-Perez *et al.*, "Cloud and network facilities federation in bonfire," in *FedICI-2013*, 2013, pp. 1–10.
[2] D. Bernstein and Y. Demchenko, "The ieee intercloud testbed–creating the global cloud of clouds," in *CloudCom 2013*. IEEE, 2013, pp. 45–50.
[3] *XIFI Projec*, Website: http://fi-xifi.eu.
[4] Y. Al-Hazmi *et al.*, "A monitoring system for federated clouds," in *CLOUDNET 2012*. IEEE, 2012, pp. 68–74.
[5] *NAGIOS Monitoring Tool*, Website, www.nagios.org.
[6] *ZABBIX Monitoring Tool*, Website, www.zabbix.com.
[7] *GroundWork*, Website, www.gwos.com.
[8] *CloudStatus*, www.hyperic.com/products/cloud-status-monitoring.
[9] *AC Nimsoft Monitor*, www.nimsoft.com/solutions/nimsoft-monitor.html.
[10] G. Spanoudakis *et al.*, "The serenity runtime monitoring framework," *Security and Dependability for Ambient Intelligence*, pp. 213–237, 2009.
[11] *MonALISA*, Website, monalisa.caltech.edu/monalisa.htm.
[12] *Ganglia Monitoring System*, Website, www.ganglia.sourceforge.net.
[13] J. Tordsson *et al.*, "Towards holistic cloud management," in *European Research Activities in Cloud Computing*, D. Petcu and J. L. Vazquez-Poletti, Eds. Cambridge Scholars Publishing, 2012, pp. 122–150.
[14] G. Katsaros, R. Kübert, and G. Gallizo, "Building a service-oriented monitoring framework with rest and nagios," in *Services Computing, 2011 IEEE International Conference on*. IEEE, 2011, pp. 426–431.
[15] S. De Chaves *et al.*, "Toward an architecture for monitoring private clouds," *IEEE Communications Magazine*, vol. 49, pp. 130–137, 2011.
[16] G. Katsaros *et al.*, "A multi-level architecture for collecting and managing monitoring information in cloud environments," in *Cloud Computing and Services Science, 2011 1st Int. Conference on*. IEEE, 2011, pp. 1–4.
[17] B. König *et al.*, "Elastic monitoring framework for cloud infrastructures," *IET Communications*, vol. 6, pp. 1306–1315, 2012.
[18] G. Katsaros *et al.*, "An integrated monitoring infrastructure for cloud environments," *CLOSER 2012*, pp. 149–164, 2012.
[19] S. Clayman *et al.*, "Monitoring services in a federated cloud - the reservoir experience," *Achieving Federated and Self-Manageable Cloud Infrastructures, IGI Global*, 2012.
[20] *Amazon CloudWatch*, Website, www.aws.amazon.com/cloudwatch.
[21] *Fed4FIRE Projec*, Website: www.fed4fire.eu.
[22] E. Salhi, S. Lahoud, and B. Cousin, "Global vs. per-domain monitoring of multi-domain networks," in *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*. IEEE, 2011, pp. 391–398.
[23] *pan-European GÉANT Network*, Website: http://www.geant.net.
[24] M. Valla *et al.*, "The context api in the oma next generation service interface," in *Intelligence in Next Generation Networks, 2010 14th International Conference on*, 2010, pp. 1–5.
[25] *OWAMP*, Website: http://software.internet2.edu/owamp/.
[26] *BWCTL*, Website: http://software.internet2.edu/bwctl/.
[27] *Iperf*, Website: http://code.google.com/p/iperf/.
[28] *OpenNMS*, Website: http://www.opennms.org/.
[29] *HDFS*, Website: http://hadoop.apache.org.
[30] *Hive*, Website: http://hive.apache.org.
[31] *Spark*, Website: http://spark.apache.org.