The Unavoidable Convergence of NFV, 5G, and Fog: A Model-Driven Approach to Bridge Cloud and Edge

Frank van Lingen, Marcelo Yannuzzi, Anuj Jain, Rik Irons-Mclean, Oriol Lluch, David Carrera, Juan Luis Pérez, Alberto Gutierrez, Diego Montero, Josep Martí, Ricard Masó, and Juan Pedro Rodríguez

The advances made in managing hyper-distributed infrastructures involving the cloud and the network edge are leading to the convergence of NFV and 5G, supported mainly by ETSI's MANO architecture. The authors propose that fog computing will become part of that convergence, and they introduce an open and converged architecture based on MANO that offers uniform management of IoT services spanning the continuum from the cloud to the edge.

Abstract

The interplay between cloud and fog computing is crucial for the evolution of IoT, but the reach and specification of such interplay is an open problem. Meanwhile, the advances made in managing hyper-distributed infrastructures involving the cloud and the network edge are leading to the convergence of NFV and 5G, supported mainly by ETSI's MANO architecture. This article argues that fog computing will become part of that convergence, and introduces an open and converged architecture based on MANO that offers uniform management of IoT services spanning the continuum from the cloud to the edge. More specifically, we created the first YANG models for fog nodes, for IoT services involving cloud, network, and/or fog, and expanded the concept of "orchestrated assurance" to provision carrier-grade service assurance in IoT. The article also discusses the application of our model in a flagship pilot in the city of Barcelona.

Introduction

Several technologies relevant to the expansion of the Internet of Things (IoT) have emerged, including network functions virtualization (NFV) [1], the fifth generation (5G) wireless systems [2], and fog computing [3, 4]. In particular, the European Telecommunications Standards Institute (ETSI) has standardized the reference architecture for NFV management and orchestration (MANO) [5], a cornerstone for building, deploying, and managing services in NFV environments. Advances in the 5G radio access network (RAN) and Multi-access Edge Computing (MEC) group at ETSI [6] are also key for the IoT evolution. MEC proposes a virtualized platform built on an NFV infrastructure, and is expected to leverage the NFV MANO architecture and application programming interfaces (APIs). The majority of service providers (SPs) will exploit NFV infrastructures not only for virtualized RANs and MEC, but also for other services, including enterprise, residential, and cloud offerings. Thus, the convergence of NFV and some of the key building blocks of future 5G architectures seems unquestionable (Fig. 1).

Fog computing addresses use cases with requirements far beyond cloud-only solution capabilities. For instance, a set of oil wells can produce petabytes of data daily, and all these data cannot be sent to the cloud due to limited or unreliable backhaul connectivity. Fog allows data to be filtered and analyzed locally, and actionable decisions to be made in real time (for safety reasons, preventive maintenance, etc.).

The complementarity between fog and cloud has traditionally been seen as a mandatory feature in any fog platform. In this article, we advocate for a different approach. Rather than specifying an architecture where fog and cloud are complementary by design, we focus on a service management architecture that literally fuses fog and cloud. We contend that the research community must start thinking about one computing fabric, managed as a single entity, in a service-centric way. With this approach, an infrastructure composed of fog nodes, network nodes, and cloud nodes is exposed to service administrators as a unified resource fabric. Administrators can then define where to instantiate resources according to the service requirements.

Compute nodes in the cloud or fog are treated architecturally the same, as the service management platform unifies the life cycle management of services that might require instances running in the fog, cloud, or a combination. Distinctive features of a fog, network, or cloud node will be captured by their corresponding YANG models [7]. A main advantage of this approach is that different IoT services can coexist and be managed in a uniform way.[1] Consider services where fog is not required (e.g., sensor communications supported through long-range radio, such as LoRA or NB-IoT [8]) vs. applications where fog is mandatory (e.g., industrial machines producing data filtered and analyzed by a fog node that is directly connected to the former through a wired interface [9]). System integrators and SPs can leverage our unified infrastructure and uniform service management to provide services to customers in both segments simultaneously.

The requirements to host and manage NFV, MEC, and fog computing services are undeniably

[1] IoT services are compositions of software services, hardware, and connected things like sensors to enable business outcomes.

Digital Object Identifier: 10.1109/MCOM.2017.1600907

Frank van Lingen, Marcelo Yannuzzi, Anuj Jain, Rik Irons-Mclean, and Oriol Lluch are with Cisco Systems; David Carrera, Juan Luis Pérez, and Alberto Gutierrez are with the Barcelona Supercomputing Center; Diego Montero is with the Technical University of Catalonia; Josep Martí, Ricard Masó, and Juan Pedro Rodríguez are with NearbySensor.

similar. SPs and enterprises embracing NFV will seek to maximize their investments, leveraging their NFV infrastructure and MANO systems to the largest extent possible. It is only a matter of time until fog computing becomes part of the convergence that we are already witnessing between NFV and 5G/MEC, driven by SPs and enterprise investments (Fig. 1).

This article describes an architectural approach that addresses some of the technical challenges behind the convergence shown in Fig. 1, with special focus on bridging the gap between cloud and fog. We introduce a model-driven and service-centric architecture that is, at the time of writing, perfectly aligned with the OpenFog Consortium (OFC) reference architecture [4].

A Model-Driven and Service-Centric Approach

Our model is based on a two-layer abstraction:

- The separation of the "service intention" (i.e., "what") from the "service instantiation" (i.e., "how")
- The decoupling of the "service instantiation" from the specifics of the devices where the instances will be ultimately deployed — independent of whether they will be instantiated in the cloud or network, or at the edge

The left side of Fig. 2 shows how this abstraction is achieved through utilization of a standardized data modeling language, YANG [7]. The right side shows a small YANG model snippet that is part of a sensor telemetry use case and multi-protocol data aggregation described later. The YANG model shows various parameters related to the tenant, the fog node, and analytics components.

YANG is used for service and device modeling. Models are machine-readable, and can be interpreted and processed by an orchestration system, which is one of the basic components of our NFV MANO implementation. A main role of the orchestration system is to translate the "what" to the "how," and enforce corresponding configurations on specific device models. The translation process is captured by mapping functions depicted on the left side of Fig. 2, which transform service definitions and input parameters to device configuration parameters. Configurations are enforced through NETCONF interfaces exposed by any device present in our infrastructure (cloud, network, or edge). NETCONF is a standard Internet Engineering Task Force (IETF) protocol used to install and update device configurations. The protocol was chosen because of its ubiquitous presence, as it has been largely adopted by SPs and enterprises as part of their service management operations.

The two-layer abstraction is not new. We believe, however, that this is a safe bet toward the convergence shown in Fig. 1, since this is precisely what many SPs and large enterprises are starting to use when adopting NFV. The novelties introduced in this article are:

- The extension of the model to cover fog and IoT. Although the utilization of NETCONF and YANG has traditionally focused on network configuration, these standards are sufficiently generic to be leveraged for any kind of device or service model. We have expanded the reach of NETCONF and YANG to fog nodes.
- The extension of orchestrated assurance to cover IoT services (i.e., service assurance is an intrinsic part of the IoT service definition in YANG).

In our model, everything is reduced to services. Infrastructure services (i.e., those dealing directly with physical resources) become an inte-

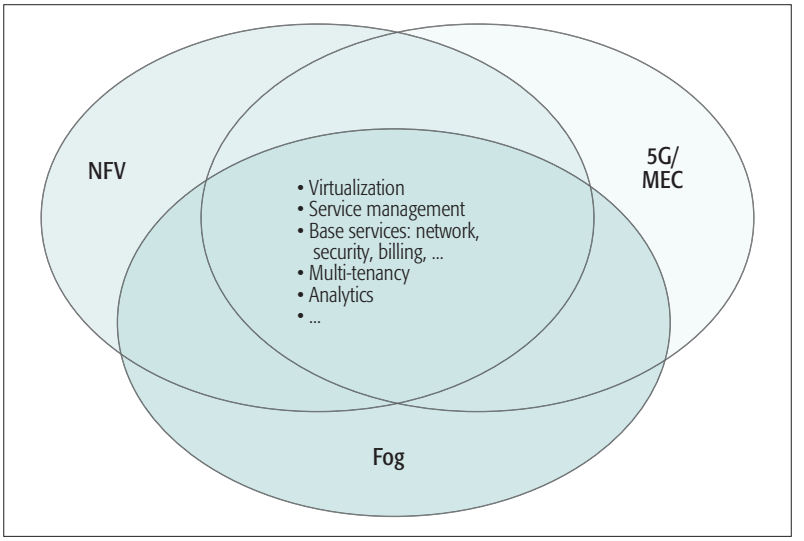

Figure 1. Different technologies with overlapping needs and challenges.

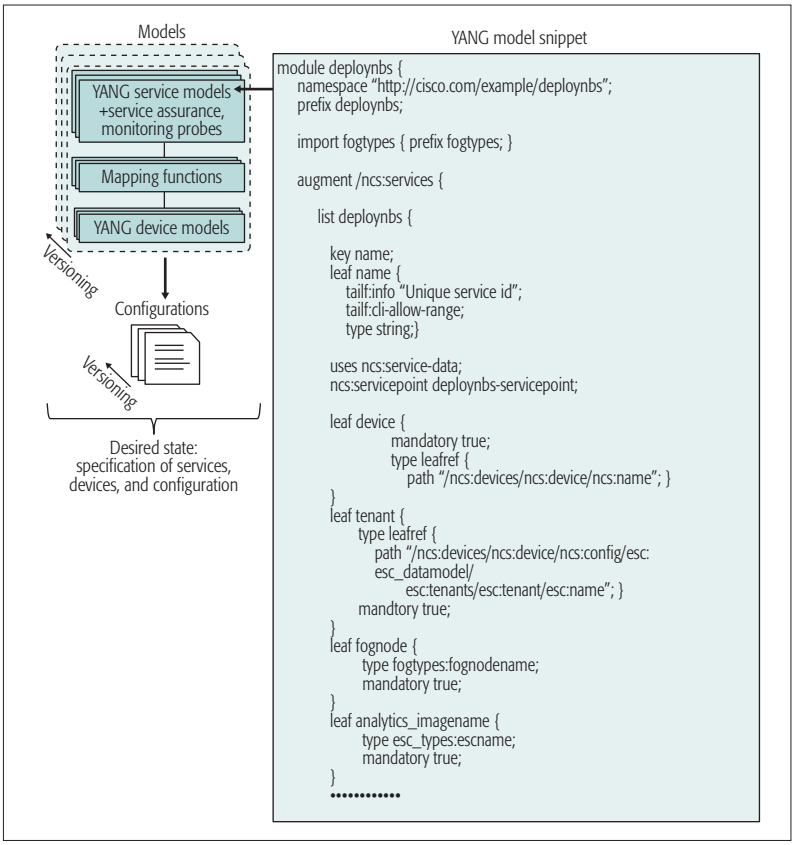

Figure 2. YANG is used for both service and device modeling, making devices transparent to service management. Service assurance is supported through distributed monitoring across the infrastructure and feeds a transactional orchestration system, which can deal with any discrepancy between the current state and the desired state of an IoT service.

This approach is proven to facilitate life cycle management of large collections of services in NFV environments, and is critical to reduce complexity when designing IoT services, with much more infrastructure heterogeneity beyond the datacenter.

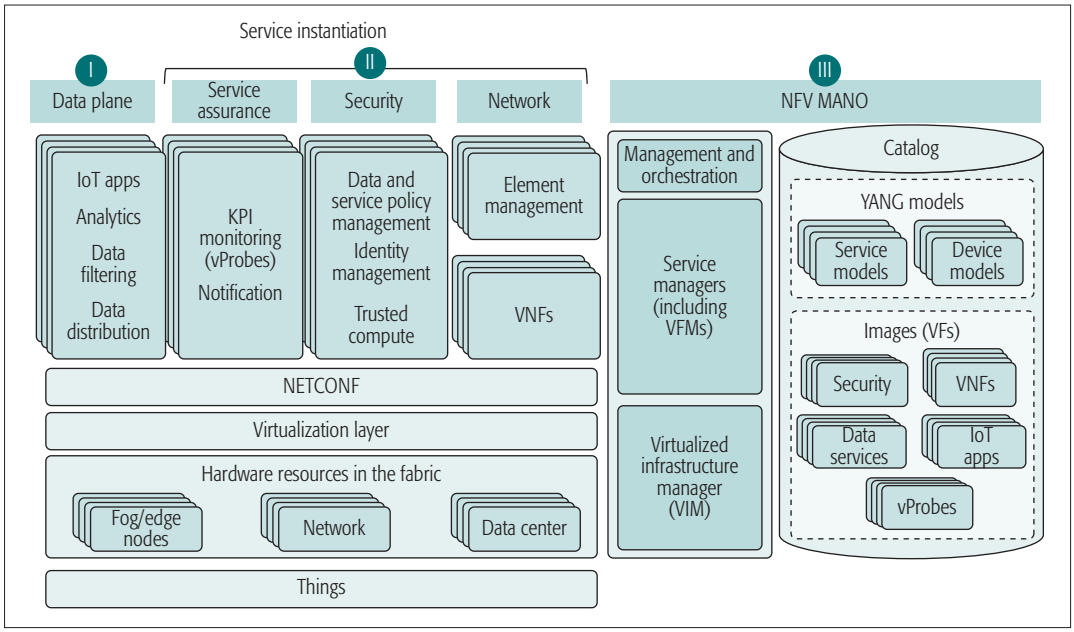

Figure 3. ETSI MANO architecture extended to cover service management beyond the traditional NFV and networking domains. The building blocks are grouped into three main categories, each of which offers a set of services that are common to the large majority of IoT deployments.

gral part of modeling and enabling higher-level services. The composition of YANG models for building services is key to breaking down the complexity of service modeling and for reusing parts of existing service catalogs. This approach is proven to facilitate life cycle management of large collections of services in NFV environments, and is critical to reduce complexity when designing IoT services, with much more infrastructure heterogeneity beyond the data center.

The workflow for deploying a service starts from a service model definition. Subsequently, the service is instantiated and the configuration is enforced on one or more devices in the infrastructure. As shown in Fig. 2, YANG models and corresponding device configurations are stored, and can be rolled back to previous models and/or configuration versions should this be necessary.

The designer of a service model can include key performance indicators (KPIs), and compare the "actual state" of the instantiated service to the "desired state" as part of the service assurance. In case of discrepancy between states, service assurance components depicted in Fig. 3 notify the service managers within the NFV MANO architecture, and orchestration services take action to align the actual state with the desired state.

Toward Converged Service Management

Our approach uses the well-known NFV MANO architecture and extends it to other service categories, beyond NFV and network devices. Figure 3 shows the main building blocks, split into three categories: the data plane; basic components to support data plane functionality (service assurance, security, and networking); and the management plane based on NFV MANO.

The first category consists of services to manipulate, share, and distribute data to other services over the cloud to edge continuum. The second category provides services to ensure secure and reliable service operation and efficient data delivery. The third category encompasses the usual MANO components extended with new models to cover fog nodes and IoT services, and IoT-specific features, especially in the areas of security and service assurance. Except for the presence of NETCONF and YANG, the blocks shown in Fig. 3 are technology-agnostic. NETCONF and YANG are present to emphasize the need for the adoption of standardized and broadly accepted interfaces and data modeling languages.

The main components and implementation (Table 1) are described in more detail below.

Data Plane: Includes data distribution and data sharing services. The data and service policy management module present in the security block (II) allows administrators to share data between tenants in a controlled and secure way. This enables sharing of data between services on multiple fog nodes, and also across fog and cloud, while adhering to policies defined in the data and service policy management module. These policies could be leveraged to build so-called data and resource pricing models [10] as an incentive to optimize resource usage in multi-tenant environments.

The platform supports the utilization of different message brokers if needed. The analytics component can be supported by databases deployed at the edge and the data center. The setup should offer multi-tenancy, and enable streaming and/or historian analytics depending on requirements.

Service Assurance: Services are linked to a certain quality of service, specified by KPIs. An effective technique starting to be used in NFV scenarios is to monitor the KPIs through a set of virtual probes (vProbes), instantiated at the points where relevant parameters need to be monitored, and usually deployed in a distributed way. A combination of passive and active vProbes can efficiently detect violations to KPIs directly at the problem source.

This technique has proven simpler and more

accurate than traditional approaches, which are often based on gathering information from multiple sources, including measurement tools, logs, monitoring systems, and so on, for root cause analysis. In our platform, KPI violation is notified to the service owner, and orchestrator or service manager, to resolve the discrepancy between observed and desired service state. Service assurance forms an integral part of the service definition and composition developed in YANG (captured through service models). The role of vProbes, their locations, and actions that need to be taken upon KPI violations are specified in the service model. Service assurance covers both the infrastructure and the services that multiple tenants will deploy on top of it.

Security: An integral part of the architecture, since the extension of NFV MANO to cover fog and IoT substantially increases the attack surface. Security elements are categorized into the following three groups.

Network-Based Security and Role-Based Access Control: Provided through specific virtual network functions (VNFs), such as firewalls, intrusion detection applications, and so on. Many of these will be instantiated in fog nodes, and service designers decide whether a security VNF is instantiated to protect an entire node (e.g., a fog node), a specific tenant execution environment (TEE) within a node, or a pool of TEEs instantiated in a single chassis or spanning across several of them. This category also covers mechanisms for controlling which services can access what data and when, as well as which users can access what services and resources and when. Mechanisms for authentication and authorization are essential to control data sharing policies and grant access to specific resources for each tenant. Different tenants usually deploy different services, and require different KPIs to be monitored. In the case of Barcelona, data associated with service assurance of different tenants was stored in a geo-distributed historian database. Enforcement of access control on the database ensured separation of information between tenants. Internal communication between components of the platform were also secured using encryption.

Host-Based Security: Includes aspects such as trusted compute based on the trusted platform module (TPM), operating system hardening, disk encryption, vulnerability management and patching policies, security information and event management (SIEM), enforcing isolation among TEEs, and so on. The Barcelona pilot covered the majority of these aspects, with strong focus on securing fog nodes.

Fog-Based Security: Security the system can provide to help protect "things" connected to fog nodes, and protect the fabric and its services from malicious things located at the network edge. The IETF has recently proposed the manufacturer usage description (MUD) specifications [11] as a first step toward a standardized and secure way of onboarding, and connecting, simple "things" to an IoT system. Fog can play an enabling role for MUD, to mediate and automate the process of device onboarding, and to enforce security policies ensuring such devices can only establish communications subject to their intended use.

Network: This covers the core networking VNFs and ancillary systems, such as virtualized switches, routers, DHCP servers, load balancers, WAN optimizers, and so on.

Component	Technology Ecosystem
Analytics	**Cisco ParStream**, Cisco Edge and Fog Fabric (EFF), Apache Storm, GE Predix, SAP, etc.
Data filtering and normalization	**Various processes for anomaly detection including Kalman filters, NearbySensor Agent for data normalization, etc.**
Data distribution	**RabbitMQ**, Cisco EFF, Apache ActiveMQ, DDS, etc.
vProbes (service assurance)	**netrounds probes, NearbySensor Assurance, etc.**
Data and service policy management	**Specifically implemented during the project.**
Identity management	**LDAP and distributed replicas**, Cisco ISE, Active Directory, etc.
Trusted compute	**TPM/TXT, OSSIM, Open Attestation, LUKS, SELinux, AppArmor, QEMU, and secured configuration files.**
VNFs	**Cisco CSR1kv, Cisco Firewalls**, Cisco ESR, Palo Alto Firewalls, load balancers, etc.
Management and orchestration	**Cisco tail-f NSO**, Puppet, Chef, etc.
Service managers (VFMs)	**Cisco Elastic Services Controller (ESC)**, Ciena Blue planet, Brocade, etc.
Virtualized infrastructure manager (VIM)	**OpenStack**, vSphere, etc.
Fog hardware	**Cisco IOx devices, Nebbiolo Technologies, NearbySensor Box, ADLink, Darveen, etc.**
YANG models	**Various models for services and devices: data sharing, analytics, NearbySensor VFs, Fog nodes, etc.**

Table 1. Potential technologies for implementing the main components depicted in Fig. 3. In bold are the ones used in Barcelona, and we also list other alternatives where applicable.

NFV MANO: Unlike traditional NFV deployments, where services are instantiated in environments with homogeneous IT and network infrastructures, for many hyper-distributed IoT environments, heterogeneity of devices and communications is more the rule than the exception. Capturing this heterogeneity in simple, standard, and machine-readable ways is essential. YANG models provide this, since not only can network elements be modeled but also fog nodes, elementary things using MUD specifications [11], as well as IoT services to be deployed. The YANG model snippet in Fig. 2 was used in Barcelona, and was part of the catalog of services and device models shown on the right of Fig. 3.

The NFV MANO block in Fig. 3 is based on ETSI's three-tier model:

- Management and orchestration
- Service managers, supporting multiple vendors
- The virtualized infrastructure manager (VIM)

Traditional VNFs in ETSI's MANO terminology correspond to a subset of virtual functions (VFs) managed by our architecture, with many of our VFs containing IoT-related functions rather than only network functions. A certain level of atom-

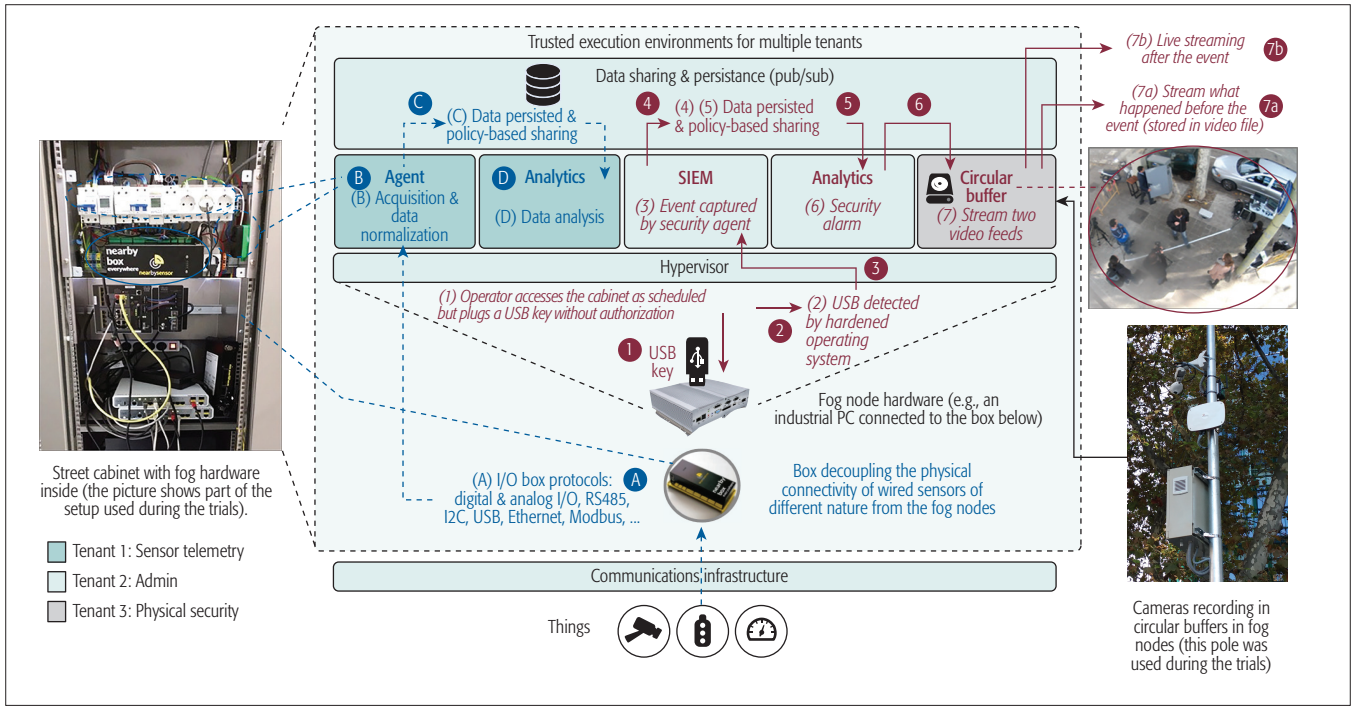

Figure 4. Two use cases deployed and managed through our converged architecture. They involve different tenants and different VFs running concurrently in a single fog node housed in a street cabinet. Sequence A–D represents the data flow for the first use case (sensor telemetry), while sequence 1–7 represents the flow for the second one (physical security of fog nodes outdoors). All the services and devices shown in the figure have their corresponding YANG models.

ic behavior when updating many services at the edge is a necessity. The MANO system achieves this through transactional operations across services involved. This is similar to a two-phase commit across multiple databases, ensuring physical devices associated to these services continue to function properly, and the system as a whole stays in a consistent state.

Services are deployed by combining the YANG models, associated images for the VFs to be instantiated and configurations of networks, message brokers and data flows, security policies, databases, and so on supporting the service. Either a user will specifically push a new service to a set of edge, network, or data center nodes, or, depending on the KPIs and overall system state, MANO will determine the best possible location for services to be deployed. Because all fabric hardware resources have NETCONF interfaces and are described through appropriate YANG device models, from a deployment perspective, there is no distinction between edge, network, or data center nodes.

Technologies for Implementing the Architecture

While previous paragraphs describe the main architectural components, Table 1 shows various technologies that can be leveraged to implement them, including ones used in the Barcelona pilot. YANG models of fog nodes or service components can be implemented by various hardware and software vendors (or developed by the open source community). These models can become part of our implementation. Indeed, several YANG models and service templates could become part of the OFC interoperability trials [4], and, together with the extended MANO architecture presented in this article, form part of a reference framework for open implementations.

Motivation and Pilot Implementation in the City of Barcelona

Barcelona realized that the more than 3000 street cabinets deployed in the city form a natural infrastructure to build out their smart city vision. Their goal is to have a single, extensible, and distributed platform from edge to cloud to address opportunities that current and future technologies for urban services will bring in an integrated way. The incentives behind this approach are described in detail in [12], but one of the aims is to reduce solution silos and the cost of operating different solutions in the city. While [12] addresses multiple use cases where fog is mandatory, this section delves into the orchestration needs, and outlines the automation and uniform life cycle management of two use cases spanning the cloud to edge continuum (Fig. 4). These use cases were recently demonstrated in Barcelona, and extend those described in [12].

Dealing with Scale and Management Complexity

Fog nodes were housed inside cabinets, such as the one shown on the left of Fig. 4. For resiliency, some services require more than one fog node per cabinet. A large fraction of cabinets will host instances of the same service, so managing the life cycle of a single service across the city may involve the configuration of thousands of fog nodes.

The development of an IoT service usually entails the integration of multiple technologies supplied by a partner ecosystem, including sensors, application-specific gateways, fog and network nodes, data brokers, security, and so on. Thus, managing the life cycle of an IoT service (i.e., onboarding devices, performing day zero

configurations, as well as managing the state and configurations after the initial deployment) can become quite complex — a challenge faced in Barcelona, and other cities and industries.

The NFV and 5G communities have addressed similar challenges, both heavily betting on the ETSI MANO architecture. We argue that this architecture will not only facilitate the convergence of NFV, 5G/MEC, and fog, but will also offer the automation means to deal with the scale and complexity posed by IoT. The goal is to hide underlying complexity from administrators, and turn IoT service management into simple and intuitive operations. The success of platforms such as Amazon's AWS Lambda, Google's search engine, or legacy technologies like TV is largely due to the way they manage scale, and the way they have abstracted the underlying complexity from end users. The pilot conducted in Barcelona followed the same principles.

Setup in Barcelona

Figure 4 offers a schematic view of a setup used during the Barcelona pilot. The left side shows a cabinet interior with several elements:

- A power distribution board with different monitoring elements and circuit breakers
- A box that enables decoupling and aggregating the physical connectivity of different families of wired sensors, simplifying the I/O requirements of the fog nodes
- The fog nodes themselves
- Others

The central part of the figure provides a logical representation of the setup, and shows data flows for two different use cases demonstrated in Barcelona. The bottom shows a set of "things" (e.g., sensors and control and actuation elements), which can be located both within and outside cabinets.

We used fog nodes supplied by different vendors (Table 1). The example in the figure shows an industrial PC, connected through Ethernet to the NearbySensor box. The top of the figure illustrates the hypervisor as well as several TEEs, which belong to three different tenants running in the fog node. The right side shows part of the external setup for one of the use cases, including a pole, a camera, and a snapshot of one of the videos captured by the latter.

We proceed to describe two use cases depicted in Fig. 4, with emphasis on automation enabling the data flows illustrated therein.

Use Case 1: Sensor Telemetry through Street Cabinets

From a data plane standpoint, this use case is depicted as sequence A–D in Fig. 4. Step A shows how specialized hardware at the network edge can help aggregate and simplify communication with different types of sensors (e.g., for monitoring temperature, power, and access), as well as a number of controllers, such as circuit breakers and uninterruptible power supplies, using various protocols and interfaces.

Data collected through the box in A is sent to an agent (B), which normalizes the data. This agent is part of a TEE that belongs to the city department in charge of monitoring the cabinets and the environment (tenant 1 in Fig. 4), and can run in a Docker container or a VM depending on security and performance requirements. Data processed by the agent can be maintained and shared in a policy-based and secure way with other processes running either on the fog node, on other fog nodes, or in the cloud (C). The data sharing and persistence block on the top belongs to the administrative tenant in charge of managing common services across tenants (tenant 2), such as data sharing policies, system-level analytics, and the security of the fog node itself. Step D shows tenant 1 subscribed to different data topics, enabling the gathering and examination of data from different sources, and triggering actionable decisions based on the result of the analysis. Applications (and their YANG service models) running in B and D can be supplied by different providers.

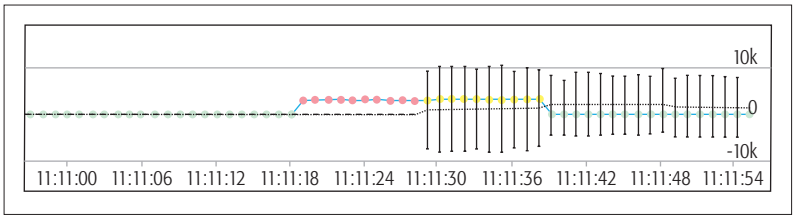

Figure 5. Power consumption data (in Watts) associated with sequence A–D depicted in Fig. 4. The dots represent the observed values B–C; the dotted curve represents estimated values and uncertainty using a Kalman filter (D).

In the example, the process in D analyzes — among other things — power consumption obtained from monitors connected to the box in A, and estimates upcoming values based on a Kalman filter (Fig. 5). The goal of the analysis is three-fold:

- Estimate and control the power consumed to prevent spikes in energy use from multiple devices.
- Dynamically manage which devices remain operatonal in case of a power outage.
- Control the service level agreement (SLA) with the energy supplier. Since the processes run locally in the fog node, this analysis and control will remain operative even if the node loses backhaul connectivity to the cloud.

The deployment of services supporting the use cases depicted at the center of Fig. 4 was entirely automated, and managed using the architecture shown in Fig. 3. Configuration of fog nodes was performed as follows:

- Zero-touch provisioning including full installation of operating system, initial configurations, security, and so on
- Deploying and configuring initial function packs, such as data sharing and persistence elements, a set of vProbes for service assurance, and more
- The tenant's TEEs, the security configurations enforcing segmentation and isolation between tenants, including their corresponding networks
- Configuration of message brokers enabling the data workflows (A–D) shown in the figure

All stages required for deploying and configuring the IoT services shown in Fig. 4 were managed with a few clicks. More importantly, instantiations

Cameras record continuously and send video to associated Fog nodes, where they are stored in circular buffers. Only when an event occurs, such as inserting a USB key, the Fog-based system triggers two video streams: what happened before the event (stored in the circular buffer); and what happens right after the event in real-time.

can be done in an individual cabinet or thousands of them across the city once the service models and the corresponding images are available from the catalog illustrated on the right side of Fig. 3. This approach reduces the operating expenditure for managing a smart city infrastructure considerably.

Use Case 2: Physical Security of Fog Nodes

Physical security of cabinets and the devices inside is of utmost importance for the city, so we implemented several security layers. This section describes how unauthorized USB access to a fog node is detected and recorded on video (cf. the right side of Fig. 4).

Cameras record continuously and send video to associated fog nodes, where they are stored in circular buffers. Only when an event occurs, such as inserting a USB key, does the fog-based system trigger two video streams: what happened before the event (stored in the circular buffer and what happens right after the event in real time. The reasons for not streaming continuously to the cloud, but deploying services at the edge using fog, are cost, privacy, and data storage overheads [12]. Actions taking place after the USB stick is plugged in (step 1 in Fig. 4) until the video is stored and made available correspond with steps 2–7 in Fig. 4: 2) the USB is detected, generating an event; 3) the event is captured by a SIEM agent; 4) and 5) reporting the event through the communication bus; 6) analysis of the event and triggering the appropriate response; and 7) streaming the videos to one or more predefined locations (e.g., in the cloud). Communications between the use case components occur transparently, and there is no semantic separation between fog and cloud deployed components.

An important aspect is the multi-tenant nature of the converged cloud/network/fog platform, as services supporting this use case were deployed on the same fog node used before. As in the previous use case, not all services and hardware needed for implementing the use case were managed by the same city department. The IT department manages the services running in the cabinets, while the cameras and their functionality are managed by another department. We leveraged common functions offered by the platform, including the data sharing service, various security and service assurance functions, and so on.

This use case demonstrates that, an IoT service is typically composed of multiple services. Services can be managed by different tenants or securely shared among multiple tenants (cf. the data sharing service in Fig. 4). Some services may be deployed and associated solely with an IoT service and tenant (e.g., the circular buffer service in Fig. 4). Besides service composition and reusability, the aim is to turn the deployment and management of IoT services into almost trivial tasks, which can be operated through a set of intuitive actions (performing a few clicks on a dashboard). All these aspects are at the heart of our converged architecture.

Related Work

The authors in [13] discuss how some of the MANO concepts can be exploited to deliver end-to-end network services using a description-based approach over a set of distributed resources. This work has similarities with ours, although our focus is mainly IoT and fog, whereas [13] is centered on the orchestration and deployment of network services. Note that the work discussed in this article sits at the intersection of NFV, 5G/MEC, and IoT and fog, and extends the concept beyond the data center and networking to fuse cloud and fog. While MEC focuses mainly on the edge of the network [6], our approach covers the continuum from edge to cloud.

Service configuration and life cycle management are important aspects of our platform. Several products such as Puppet, Chef, Ansible, and Salt provide configuration management and help automate deployment of services. Other products like Terraform, CloudFormation, and OpenStack provide infrastructure life cycle management capabilities (e.g., for day 0 configuration), which can be combined with tools like Puppet or Chef (for day 1 onward). However, all these products mainly target IT infrastructures. In IoT, the fundamental requirements in terms of connectivity (I/O interfaces), security, applications, and data management are significantly different from those that rule the life cycle management of IT servers. The compute resources available in a fog environment are typically much more heterogeneous, and the criticality of some applications requires special treatment.

This heterogeneity, combined with the criticality of some of the services connected to physical devices, creates new requirements for service life cycle management that go far beyond what state-of-the-art tools in the IT space currently offer. Our approach takes into account this heterogeneity natively.

Finally, many of the members of the OFC are already offering fog products. This is the case of Foghorn, Nebbiolo, and Cisco, just to name a few [4]. At the time of writing, none of the products available in the marketplace are focused on a converged NFV, 5G/MEC, fog, and cloud paradigm, with emphasis on exposing the infrastructure as a single and unified computing fabric.

Conclusion

This article describes an architecture that addresses some of the central challenges behind the convergence of NFV, 5G/MEC, IoT, and fog. By using a two-layer abstraction model, along with IoT-specific modules enriching the NFV MANO architecture, we introduce a promising paradigm to fuse cloud, network, and fog, and apply this to a project in the city of Barcelona. For now, we focus only on a small number of use cases. We expect that once we start expanding this model to different domains and cities, more end-user services will be developed, enabling the reutilization of service models and associated service catalogs.

References

[1] "NFV"; http://www.etsi.org/technologies-clusters/technologies/nfv, accessed May 10, 2017.

[2] I. Chih-Lin *et al.*, "5G: Rethink Mobile Communications for 2020+," *Phil. Trans. R. Soc. A*, vol. 374, 2016.

[3] F. Bonomi *et al.*, "Fog Computing: A Platform for Internet of Things and Analytics," *Big Data and Internet of Things: A Roadmap for Smart Environments in Series Studies in Computational Intelligence*, vol. 546, 2014, pp. 169–86.

[4] OpenFog Consortium; http://www.openfogconsortium.org/, accessed May 10, 2017.

[5] "Network Functions Virtualisation (NFV), Management and Orchestration"; http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf, 2014, accessed May 10, 2017.
[6] Y. C. Hu *et al.*, "Mobile Edge Computing A Key Technology towards 5G"; http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf, 2015, accessed May 10, 2017.
[7] "YANG — A Data Modeling Language for the Network Configuration Protocol (NETCONF)," IETF RFC 6020; http://www.rfc-editor.org/rfc/rfc6020.txt, accessed May 10, 2017.
[8] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low Power Wide Area Networks: An Overview," *IEEE Commun. Surveys & Tutorials*, vol 19, no. 2, 2nd qtr., 2017, pp. 855–73.
[9] "Fog Computing: Keystone of Industry 4.0, KUKA/Nebbiolo Technologies"; https://www.kuka.com/-/media/kuka-corporate/documents/press/kuka_nebbiolo_solutions.pdf, accessed May 10, 2017.
[10] M. Chiang *et al.*, *Smart Data Pricing*, Wiley, 2014.
[11] E. Lear, R. Droms, and D. Romascanu, "Manufacturer Usage Description Specification," IETF draft; https://tools.ietf.org/html/draft-ietf-opsawg-mud-05, 2017, accessed May 10, 2017.
[12] M. Yannuzzi *et al.*, "A New Era for Cities with Fog Computing," *IEEE Internet Computing Mag.*, Special Issue on Fog Computing, vol. 21, Mar./Apr. 2017, pp. 54–67.
[13] I. Cerrato *et al.*, "Toward Dynamic Virtualized Network Services in Telecom Operator Netnetworks," *J. Computer Networks*, vol. 92, 2015, pp. 380–95.

Biographies

Frank van Lingen (fvanling@cisco.com) is a technology strategist at Cisco's Corporate Strategy Office where he works on strategic innovation, emerging technologies, and customer/partner co-innovation, focusing on areas such as analytics, fog computing, AI, and the Internet of Things. He has a Ph.D. in computer science from the University of Technology Eindhoven.

Marcelo Yannuzzi (mayannuz@cisco.com) is a principal engineer at Cisco's Corporate Strategy Office, where he works on strategic innovation in the areas of fog computing, IoT, and security, and provides strategic advice on new business opportunities and technologies for Cisco. He has a Ph.D. in computer science from the Technical University of Catalonia — BarcelonaTech.

Anuj Jain (januj@cisco.com) is a director at Cisco's Corporate Strategy Office. He leads an innovation team working in the areas of fog computing, IoT, next-generation computing, artificial intelligence, and security, with a focus on disruptive technologies and strategic business opportunities for Cisco. He has an M.S. in micro-engineering from EPFL.

Rik Irons-Mclean (rironsmc@cisco.com) is a lead architect for next generation IoT platforms in Cisco's Industry Product and Technology Group. As both lead architect and technical lead, he has successfully delivered a number of emerging technology solutions for smart grid, process automation, oil and gas, building automation, and green energy. He has an M.B.A. from the Bradford University School of Management, and is currently studying for a doctorate in cyber security.

Oriol Lluch Parellada (orilluch@cisco.com) is a technology strategist at Cisco's Corporate Strategy Office where he works on strategic innovation in the field of artificial intelligence. His research interests include cloud infrastructure, fog computing, service orchestration, and artificial intelligence. He has an M.S. in telecommunications engineering from the Technical University of Catalonia — BarcelonaTech and an E.M.B.A. in management of technology from EPFL and HEC Lausanne.

David Carrera (david.carrera@bsc.es) leads the Data-Centric Computing Research Group at the Barcelona Supercomputing Center (BSC). His research interests include performance management of data-centric platforms. He has a Ph.D. in computer science from the Technical University of Catalonia — BarcelonaTech.

Juan Luis Pérez (juan.luis.perez@bsc.es) is a senior research engineer in the Data-Centric Computing Research Group at the Barcelona Supercomputing Center. His research interests include model-driven IoT. He has an M.S. in computer architecture, networks, and systems from the Technical University of Catalonia — BarcelonaTech.

Alberto Gutiérrez Torre (alberto.gutierrez@bsc.es) is a Ph.D. candidate in the Data-Centric Computing Research Group at the Barcelona Supercomputing Center. His research interests include applied machine learning, in particular, over IoT/stream data. He has an M.S. in data science from the Technical University of Catalonia — BarcelonaTech.

Diego Montero (dmontero@ac.upc.edu) is a Ph.D. candidate at the Networking and Information Technology Lab (NetITLab), where his research interests include network security, SDN, network virtualization, fog computing, and network mobility. He has an M.S. in computer architecture, networks, and systems from the Technical University of Catalonia — BarcelonaTech.

Josep Martí (jmarti@nearbysensor.com) is the managing director and R+D projects coordinator at NearbySensor, focusing on strategic cooperation with other companies and government agencies. He has an M.S. in telecom engineering from the Technical University of Catalonia — BarcelonaTech.

Ricard Masó (rmaso@nearbysensor.com) is the CTO at NearbySensor, where he is also the lead architect of its software products and solutions, devoted to the quick and simple integration of operational and information technologies. He has a degree in pedagogy from the University of Barcelona and is an associate professor for computing science at the Open University of Catalonia.

Pedro Rodriquez (jprodriguez@nearbysensor.com) is the engineering director at NearbySensors where he works on innovation in IoT, distributed systems, automation, edge computing, and embedded systems. He holds degrees in telecom engineering, electrical engineering, and economics, and an M.B.A. from the IESE Business School.

This heterogeneity, combined with the criticality of some of the services connected to physical devices, creates new requirements for service lifecycle management that go far beyond what state-of-the-art tools in the IT space currently offer. Our approach takes into account this heterogeneity natively.