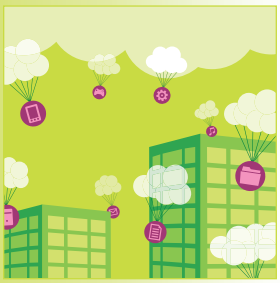

A New Era for Cities with Fog Computing

The Quadruple Silo (QS) problem refers to four silos that result from deploying commercial solutions for smart cities: physical, data, and service management silos, and their implications in administrative silos. As this analysis of a Barcelona fog computing initiative shows, a converged cloud/fog paradigm not only helps solve the QS problem, it also meets the requirements of a growing number of decentralized services where traditional cloud models fall short. In addition to exposing cases where fog computing is a must, this article shows that reasons for deploying fog are centered much more on operational requirements than on cloud-related performance issues.

Marcelo Yannuzzi, Frank van Lingen, Anuj Jain, and Oriol Lluch Parellada
Cisco Systems

Manel Mendoza Flores
Barcelona City Council

David Carrera and Juan Luis Pérez
Barcelona Supercomputing Center

Diego Montero
Technical University of Catalonia–BarcelonaTech

Pablo Chacin
Sensefields

Angelo Corsaro
PrismTech

Albert Olive
Schneider Electric

The design of platforms capable of managing urban services holistically and uniformly across different city departments is at the heart of the innovations that smart cities are expected to bring. However, the rapid evolution of technologies developed for capturing the opportunities around smart cities has led to the proliferation of proprietary and application-specific systems, or *siloed solutions*, for services such as parking, lighting, traffic, energy management, and video analysis. Each of these systems is typically deployed as a vertical solution from the "things" up to the cloud, giving rise to independent management ecosystems that show poor integration with each other.

The collateral effects of deploying dedicated solutions are starting to become apparent – especially among early adopters. For instance, cities that invested in vertical solutions for making certain urban services smarter are now shifting their focus to other more plausible models. Among these is an increasingly popular model that we recently demonstrated in Barcelona. Our model offers a common and distributed data fabric across the city for multiple departments (that is, tenants), and is based on a platform that seamlessly combines cloud computing and *fog computing* (for more details on fog, see www.openfogconsortium.org).

Here, we analyze the technical challenges that cities are facing and outline the design principles, main results, and lessons learned after embarking on an ambitious co-innovation project with the Barcelona City Council, several industrial partners, and academia. In particular, we examine the requirements of a growing number of urban services where cloud-centric approaches are insufficient and fog computing is mandatory. We describe use cases (UCs) that were implemented and demonstrated in Barcelona, providing supporting evidence for fog computing. To the best of

Published by the IEEE Computer Society
1089-7801/17/$33.00 © 2017 IEEE

Related Work in Smart City Platforms

In 2013, Nice, France launched the Connected Boulevard,[1] a smart city platform to optimize all aspects of city management, including parking, traffic, street lighting, waste disposal, and environmental quality. Although the platform included advanced data-sharing capabilities based on the Data Distribution Service (DDS) standard, it didn't address enhanced service lifecycle management or standardized models for edge nodes.

A different approach, SmartSantander,[2] focused on a European facility for research and experimentation of architectures, technologies, and applications for smart cities, while FIWARE (www.fiware.org) aimed to build an open ecosystem to ease the development of new applications for cities and other domains. However, neither SmartSantander nor FIWARE focused on fog computing.

Other smart city-related projects include those announced for Songdo (South Korea), Masdar City (Abu Dhabi, United Arab Emirates), Paredes (Portugal), Manchester (UK), Boston (US), Tianjin (China), and Singapore (for details, see www.technologyreview.com/business-report/cities-get-smarter). The approaches differ in each city, but it's clear that all will be built around secure services distributed between the edge and data centers, and that all will be managed in a coherent way.[3] For more details, see www.openfogconsortium.org.

References

1. A. Corsaro, "Connected Boulevard – It's What Makes Nice, France, a Smart City," blog, 9 Sept. 2014; http://blog.iiconsortium.org/2014/09/connected-boulevard-its-what-makes-nice-france-a-smart-city.html.
2. L. Sánchez et al., "SmartSantander: Experimentation and Service Provision in the Smart City," *Proc. 16th Int'l Symp. Wireless Personal Multimedia Communications*, 2013, pp. 1–6.
3. Y.C. Hu et al., *Mobile Edge Computing: A Key Technology towards 5G*, white paper, ETSI, 2015; www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf.

our knowledge, our project is the first to deliver a complete cloud/fog architecture that addresses the challenges currently facing many cities.

The Quadruple Silo (QS) Problem

The predominant model for installing solutions in urban spaces is based on the deployment of vendor-specific devices at the edge. The model typically includes sensors, control and actuation elements, and the corresponding gateways. In many cases, the data produced at the edge can be backhauled directly to the cloud, where different types of analytics and data management processes can be performed in a centralized way. However, other cases require more elaborated installations. A common practice is to place compute nodes closer to the data producers, in addition to the resources required in the cloud. Existing solutions in the marketplace typically come in the form of dedicated (proprietary) nodes, which are installed at the network edge for a specific application.

Figure 1 illustrates different vertical solutions combining these elements. For example, a parking system might be composed of

- self-powered sensors installed in the pavement for each parking spot, along with gateways for collecting the sensors' status within their coverage range;
- a back-end platform hosted either in a public or private cloud for managing the parking service and its data; and
- dashboards exploiting a rich set of APIs through which the solution provider exposes real-time analytics, billing information, service management functions, and so on (for more on this, see, for example, www.worldsensing.com).

In such scenarios, the in-ground sensors often rely on Low-Power Wide-Area Network technologies such as LoRa (www.lora-alliance.org) and therefore don't require the allocation of compute resources close to them. An alternative is to utilize more sophisticated sensors, such as cameras (as in Figure 1), with the advantage that they can be used for parking and video surveillance simultaneously. To detect vehicle presence, these solutions rely on video analytics, which typically run in resources embedded in the cameras themselves.

As Figure 1 shows, commercially available solutions for other services follow similar patterns, both in the type of deployment and their management structure. As we describe in our UCs later, this is true of vertical solutions for energy management (that is, systems for monitoring and controlling the power of different devices in public spaces; see www.schneider-electric.com/ww/en) and systems for traffic monitoring and regulation, such as those developed by Sensefields (www.sensefields.com).

Overall, production-ready solutions for making urban services "smarter" have traditionally focused on dedicated systems that, depending

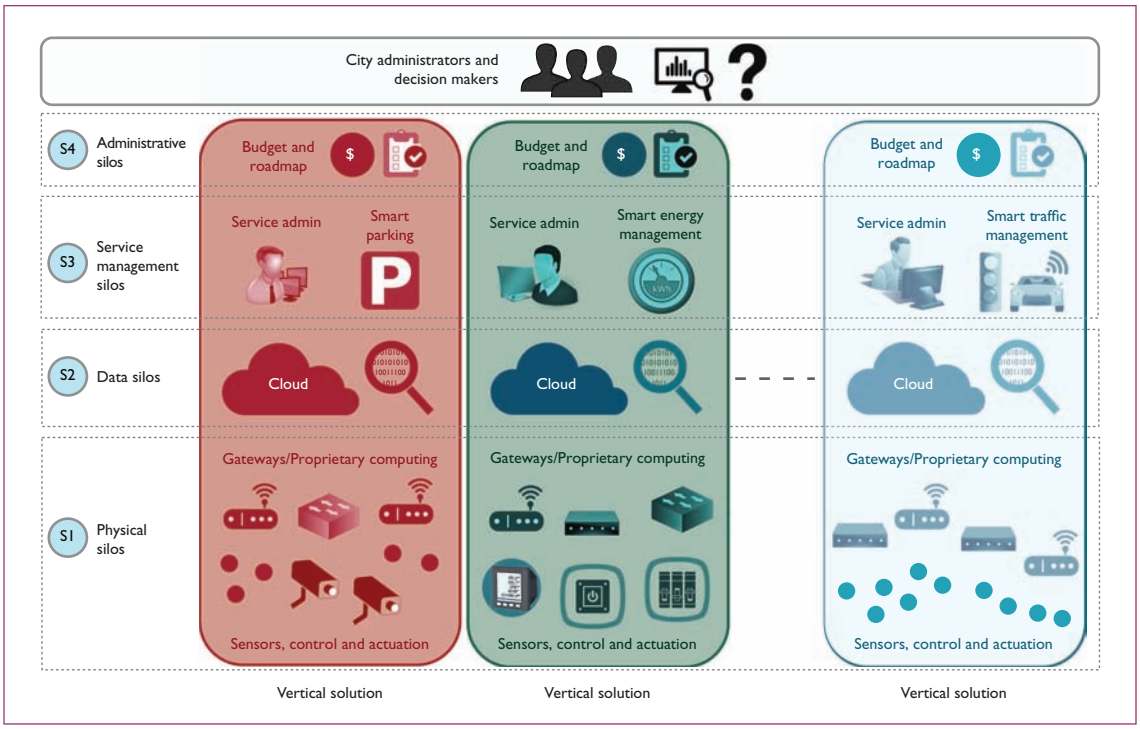

Figure 1. The quadruple silo (QS) problem. The problem refers to four silos that result from deploying commercial solutions for smart cities. Physical silos (S1) are composed of vendor-specific sensors, control, and actuation systems; proprietary gateways; and compute resources at the edge. Data silos (S2) are usually stored in the cloud and in proprietary resources at the edge. Service management silos (S3) are independent management ecosystems for managing parking, energy, traffic, and so on. Administrative silos (S4) are independent departments with different budgets, objectives, and roadmaps.

on the segments covered and their corresponding requirements, may or may not be accompanied by proprietary compute resources at the edge. This poses complex challenges for cities, because existing technologies ultimately lead to service fragmentation and the segmentation of management competencies. This is clearly visible in Figure 1, and fuels what we define as the QS problem: four categories of silos that cities face after deploying multiple vertical solutions: physical (hardware) silos (S1), data silos (S2), service management silos (S3), and administrative silos (S4).

In this scenario, basic aspects such as allocating compute resources for processing and managing data — or for managing the lifecycle of the service's hardware, software, and firmware — are vertical-specific. Moreover, siloed systems demand significant efforts to analyze relevant information across different departments, especially in real time. In fact, as the top of Figure 1 shows, city administrators rarely have access to information produced by data workflows that cross different administrative domains because vertical solutions weren't conceived for that purpose. Although some departments might manage their services using a common data center, the logical separation between vertical solutions remains unsolved. Thus, even if the corresponding back ends are hosted by a converged infrastructure, departments will suffer from data segmentation (S2).

Siloes 1–3 make it harder to align the priorities, budgets, and roadmaps across different departments (S4). Our solution addresses the challenges of the first three silos holistically, reducing barriers for this fourth silo by providing more transparency both on how resources are allocated and on the impact of uniformly deploying and managing services in various departments. We believe this will pave the way for more cohesive decisions and better alignment among city departments.

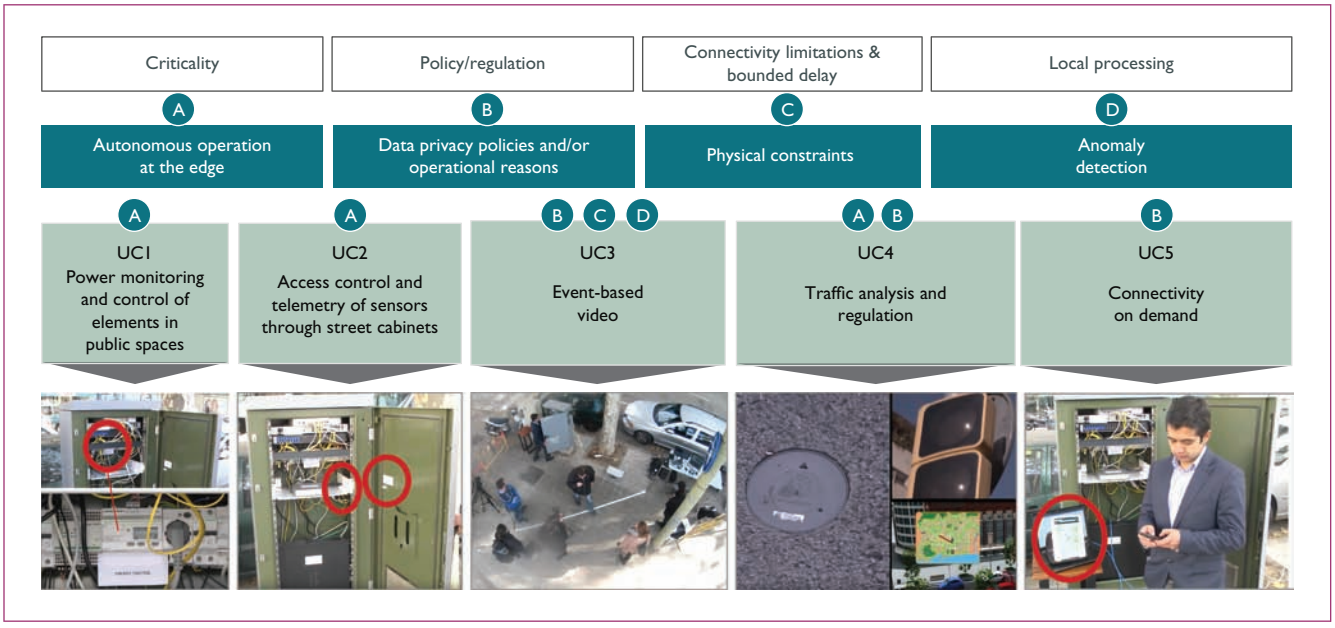

Figure 2. The five Barcelona use cases (UCs) and their fog computing requirements category. The pictures at the bottom show one of the street cabinets used during the field trials, as well as the in-ground sensors installed for traffic monitoring. As noted in the figure, (A) represents an autonomous operation at the edge; (B) represents data privacy policies and operational reasons; (C) represents physical constraints; and (D) represents anomaly detection.

Fog Computing in Barcelona

As S1 in Figure 1 illustrates, with the advent of IoT-driven solutions, the number of devices to be installed, powered, and maintained in urban spaces is increasing dramatically. This is leading to space and operational problems, as cities can't continue adding hardware for each new service deployed at the edge. Mindful of these challenges, Barcelona realized that the more than 3,000 street cabinets deployed in the city form a natural infrastructure to build out its smart city vision. Figure 2 shows one of these cabinets; the pictures were taken during our field trials in Barcelona.

A new generation of street cabinets can offer strategic control points for the city and can help accomplish goals that are essential for scaling smart city plans. From a technical perspective, the objective is to have a single, extensible, and distributed infrastructure that provides the necessary flexibility to address opportunities for current and future urban services technologies in an integrated way. This infrastructure would have nodes in three locations: the cabinets, metropolitan network, and data centers. More importantly, it should enable instantiation and management of urban services and their data in a consolidated way. From an operational standpoint, the goal is to streamline urban services and reduce hardware and service maintenance overheads. This approach is especially important in scenarios that require compute resources at the edge. In many cases, it will make more sense to aggregate multiple computing tasks in generic processing nodes hosted in the cabinets, which can also provide other functions, such as security, distributed analytics, monitoring, data normalization, and brokering.

As we describe later, we addressed five different UCs in Barcelona; to better explain their requirements, we classify them into categories (see Figure 2). A given UC requiring compute at the edge — such as fog nodes hosted in self-driving cars — might contain some or even all of the requirements from the following four categories.

Autonomous Operation at the Edge

This category encompasses urban services that must remain operative even if backhaul connectivity to the cloud is lost. This UC is typical for services where the criticality of tasks and operations demand self-sufficient means for monitoring, analyzing, and controlling different processes—such as for smart and autonomous control of energy distribution boards inside

cabinets. Here, we label this UC1; two additional UCs also required autonomous operation at the edge in Barcelona: access control to the street cabinets and telemetry of sensors within and outside them (UC2); and traffic analysis and regulation (UC4).

Data Privacy Policies/Operations

This category includes cases where: the data owner doesn't allow specific data to be moved to the cloud; the data owner wants to enforce local control and brokering for specific data exchanges; or operational or regulatory constraints preclude particular actions, such as issuing specific actuation commands from the cloud. Many IoT systems are mission critical and pose grave risks if hacked, so cyber-physical security and data privacy are among the main reasons for deploying fog computing. In the case of Barcelona, this category applies to event-based video (UC3), traffic regulation (UC4), and connectivity on demand (UC5).

Physical Constraints

This category includes cases in which data must be processed locally due to physical limitations that prevent sending and processing all of the data in the cloud. We further divide this category as follows:

- *Connectivity limitations*. In some cases, the "things" at the edge have either limited bandwidth capability to the cloud or communications are simply unreliable; they thus require localized compute resources to ensure the desired functionality. In Barcelona, for example, this applies to UC3.
- *Bounded delay*. In some cases, very low latency and/or close to real-time response is sometimes needed; this requires localized analytics, closed-loop control, and dependable actuation on physical systems (such as for self-driving cars).

Anomaly Detection

This category covers cases in which only a small fraction of the data collected is important. This is particularly relevant when UCs involve the ingestion of large data volumes at the edge, which usually requires real-time analysis and data filtering because the data scanned becomes useful only when an irregularity is detected. This category applies to the case of event-based video in Barcelona (UC3).

Use Cases

In Barcelona, we focused on five UCs, which we demonstrated using the converged architecture described later.

UC1: Power Monitoring/Element Control

The new generation of cabinets must provide a means to monitor and autonomously manage the energy distribution boards they house. These boards will plug and protect elements located both within and outside the cabinets. The systems currently offering this functionality are supplied in the form of vertical solutions (see Figure 1). These systems typically include proprietary controllers that should also be housed in the cabinets, a back end running in the cloud for gathering data from distribution boards scattered around the city, and proprietary dashboards for displaying measurements, alarms, and so on (the controllers and distribution dashboards were provided by Schneider Electric; see www.schneider-electric.com/ww/en).

Virtualizing controllers and other functions to manage energy systems offers several advantages, such as the possibility of offering catalogs of products that can be dynamically instantiated as microservices in fog nodes. These products might include functions such as

- making local decisions to keep only critical services operative during an outage (even in case of loss of cloud connectivity);
- monitoring power consumption and analyzing specific key performance indicators in real time (such as the quality of power supplied); and
- managing new resources (such as when an uninterruptible power supply is added).

As (A) in Figure 2 shows, this is considered a critical service in Barcelona and will leverage the fog nodes' horizontal functions, including service assurance for critical functions that require high availability, security, analytics, data management, and routing and switching. As we now show, the other UCs also exploit these horizontal functions.

UC2: Access Control and Cabinet Telemetry

Cabinets are exposed to the natural elements, as well as to physical abuse or even unauthorized

entry. It is therefore important that the cabinet's state be monitored, covering aspects that go beyond power control, including humidity, temperature, and the status of the door (open or closed). Entry to the cabinets can be electronically controlled, and it should be possible to access their compartments even if connectivity to the cloud is lost.

Automated entry control and environmental monitoring within the cabinets are considered critical services (A in Figure 2). Thus, the collection of data from different families of sensors (such as http://plat.one), data normalization, analysis, and access control were performed by processes running in virtualized environments in the fog nodes. In this UC, the status of certain sensors can be associated with an alarm that will go off when unauthorized access is detected. As we show in UC3, this alarm was one of the triggers used for video streaming.

UC3: Event-Based Video

The city has a large base of installed cameras. Although some are connected to a high-capacity backbone network, others are located in places where the only available option is cellular connectivity. This UC focuses on the latter (C in Figure 2). These cameras record continuously and send their video signals to the fog nodes, where they're stored in circular buffers. Only when an event occurs (D), such as when a certain noise threshold is reached or an unauthorized entry to a cabinet is detected, the fog-based system triggers two video streams that show

- what happened before the event (stored in the circular buffer in the fog node), and
- what's happening right after the event in real time.

The videos are streamed to a back end (in the cloud or other destinations).

The picture shown under UC3 in Figure 2 is a snapshot captured from one of these cameras; the video was initiated after the emulation of an unauthorized cabinet entry (in the picture's upper left corner). The reasons for not streaming continuously are typically cost, privacy, and data storage overheads (B). In this UC, a fog node can aggregate video feeds from multiple cameras nearby, and perform analysis and stream only when events are detected.

UC4: Traffic Management

Sensors in the street can register traffic and monitor aspects such as the number of vehicles, their speed, length, and other variables including estimations of queue sizes and waiting time at signalized intersections. Current solutions in this space rely on proprietary gateways to gather data from the sensors, and they're typically installed as vertical solutions. These gateways host sophisticated algorithms and analytics that are required for performing the measurements with high accuracy.

By pushing the computation to a fog node, vendors could focus on their applications and software components, and spend less effort on ancillary elements, such as maintaining their gateways, security, and so on. As Figure 2 shows, this UC is considered critical (A). In addition, due to legal regulation, it's not possible to issue commands to actuate on traffic lights directly from the cloud (B). The outsourcing of traffic regulation tasks to the fog nodes is still in its infancy and requires a comprehensive study addressing both technical and nontechnical challenges (including safety, security, resilience, and business and operational benefits). The UC demonstrated in Barcelona covered only the traffic monitoring part.

UC5: Connectivity on Demand

Given the number of activities that take place in Barcelona (concerts, sports events, conferences, and so on), the city council gets frequent connectivity requests from local TV stations, police, and so on. The connectivity is for transmitting video, covering the city's activities in real time. These requests are typically for network capacity that can't be provided through mobile networks or local WiFi, so they're currently provisioned through networking equipment hosted in the cabinets. This means that video crews, police officers, and so on must open the cabinets and get a wired connection through an internal switch/router (B).

This is a recurring need for a city that hosts many events, but provisioning the connectivity, bandwidth, and quality of service (QoS) required typically takes days. The city wanted a self-service portal that let requestors (with proper authentication) reserve bandwidth on demand and leverage the fog nodes in the cabinets to access a high-speed connection with QoS support in minutes. In Barcelona, the

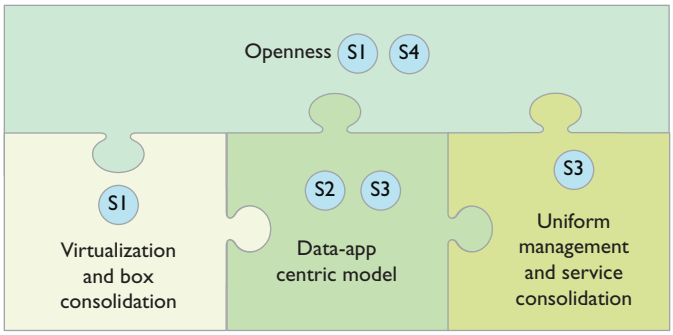

Figure 3. The fog platform's foundational principles. These basic principles address the four silos (S1–S4) in the quadruple silo problem, which results when deploying commercial solutions for smart cities.

fog nodes will replace the legacy routers and switches inside the cabinets and will host virtualized versions of them.

This is also part of the consolidation the city wants to enforce at the edge. Thus, when a request is made, the connectivity service must be automatically orchestrated and granted, which requires the configuration of fog nodes and network tunnels to reach their corresponding endpoints. And, more importantly, the cloud/fog system must correlate information from these sometimes dynamic connectivity requests with data produced by sensors that lie beneath UC2 and UC3 (such as to avoid setting off certain alarms when the door of the cabinet is opened). The capacity is reserved only for the requested period; after that, resources are released automatically. In Figure 2, the picture under UC5 shows how authorized personnel could request such connectivity through a mobile phone or a tablet, using an application that was developed during the field trials.

Design Principles

The work conducted in Barcelona focused on the design, implementation, and demonstration of a fog computing platform guided by three objectives:

1. The platform should provide a solution for problems S1, S2, and S3 as Figure 1 shows, while paving the way for more cohesive decisions and better alignment among city departments affected by problem S4.
2. The platform should manage urban services subject to the requirements and UCs described earlier — and, more importantly, demonstrate this in the streets of Barcelona.
3. The design should focus on common needs across verticals and industries. Our target was to create a platform that can serve multiple purposes and be ported and reused in other cities and other IoT domains. The street cabinets in Barcelona are a means to an end; they're simply the *point of presence* (PoP). Thus, the cabinets should not constrain the platform's reach and applicability. Other cities might rely on different PoPs for hosting fog nodes, including poles, underground installations, and so on, and should be able to leverage our designs.

Based on these objectives, the design was guided by four principles (see Figure 3). It's worth highlighting that Figure 3 shows only the basic pillars for fulfilling the three objectives mentioned earlier. Other fundamental principles for designing distributed systems of this nature — including end-to-end security, policy management, and scalability — are an integral part of the platform, and hence traverse the pillars shown in Figure 3.

The platform's openness is essential. In our model, openness has a two meanings. The platform offers not only multivendor support at the fog, network, and cloud levels, but also enables the deployment of homologated third-party applications on the city platform. Moreover, our platform should provide for uniform management of different tenants (mainly city departments, in the case of Barcelona) and their services. The goal of multitenancy is to let tenants deploy and manage their services autonomously, while benefiting from the economies of scale of integrated equipment for simultaneously providing compute resources and services to multiple departments. This includes virtualized environments that can run in the devices at the edge, network, and data center. This design principle is captured as "Virtualization and box consolidation" in Figure 3, and addresses silo S1 in the QS problem.

Another important aspect is that our design is based on the fusion of data-centric and application-centric models. This fusion is essential in IoT scenarios because, in most cases, a cloud/fog platform must efficiently manage the lifecycle of both the data and IoT applications. To this end, we have coined the term *data-app centric*, which reflects the fact that our model borrows some of the design principles and best practices that the two models offer.

Our data-app centric approach provides advanced policy management for both data and applications. This approach facilitates data sharing across user-provided applications in a controlled way. In turn, it enables new and powerful analytics, leading to actionable business intelligence across agencies and administrations. We've particularly focused on controlling and mediating data exchanges and creating sophisticated data workflows in a secure way. For instance, data produced by one department on a specific data topic can be consumed by other departments, or even third parties, based on roles and policies that control the access to the data (we delve into details of this later).

In addition to the policy framework we designed for analyzing and extracting value from the data, the platform supports the installation of applications wherever required (that is, at the network edge, the backhaul network, or the cloud). The data-app centric approach addresses silos S2 and S3 of the QS problem.

Furthermore, we focused on designing a horizontal platform that offers uniform orchestration, security, distributed analytics, and management APIs across departments; the platform particularly addresses decentralized services that require compute resources beyond the data centers. One of the design's central aspects was to make city services easy to operate and maintain through scalable orchestration and proper automation.

The design principles we followed allow fog computing to elevate IoT from point solutions to managed services at the edge. We contend that this approach will make it easier to align roadmaps and optimize budgets and purchases across departments, which will help city administrators tackle silo S4 of the QS problem.

Compared to our converged cloud/fog platform, service silos from different solution providers beyond the cloud considerably increase the operational expenses for cities. Our design offers multiple levels of consolidation, including

- box consolidation (that is, the integration of routers, switches, and compute nodes in the continuum between edge and cloud);
- the consolidation of data and its management; and
- uniform service management.

These consolidations can dramatically reduce the complexity, cost (through capital expenditures and operating expenses), maintenance, and time required to launch and deploy solutions throughout the city, while effectively addressing the challenges posed by the QS problem. In this context, *cost* refers to the comparison with the alternative siloed approach. Whether a UC is addressed with a siloed solution or through our platform, the monetary benefit of the individual UC is the same. However, once multiple UCs are deployed, our platform has two main advantages. First, it offers economies of scale, because the same edge and cloud infrastructure can be leveraged for multiple UCs. Second, our infrastructure enables sharing of data between services of different UCs, which the city can leverage to add additional value on top of the individual UCs.

Converged Architecture

Figure 4 shows a simplified version of our model. As the model shows, Barcelona initially requires a flat model composed of only two layers: fog nodes inside the cabinets and a common back end in the cloud. More complex arrangements, including a hierarchy of fog layers and hybrid clouds, might be required once the deployment of urban services begins to scale.

The architecture represents a fully distributed system, where even the individual blocks in the figure might be internally distributed. It's composed of three main groups of components: the fog nodes, the back end, and a set of cross-domain functions, including security, service assurance, and network management.

Fog Nodes

The main components inside the fog nodes include instances of virtual domains (VDs).

Instances of virtual domains (VDs). The VDs represent the execution environments for the different services running concurrently in a fog node. Each VD is application-specific and belongs to a single tenant. A tenant might have multiple instances of VDs running in the same fog node.

In our implementation, we support both Docker containers and virtual machines (VMs) on kernel-based virtual machines (KVMs). Depending on the service requirements, a VD could be instantiated as a single container or a single VM, or it could be composed of several instances

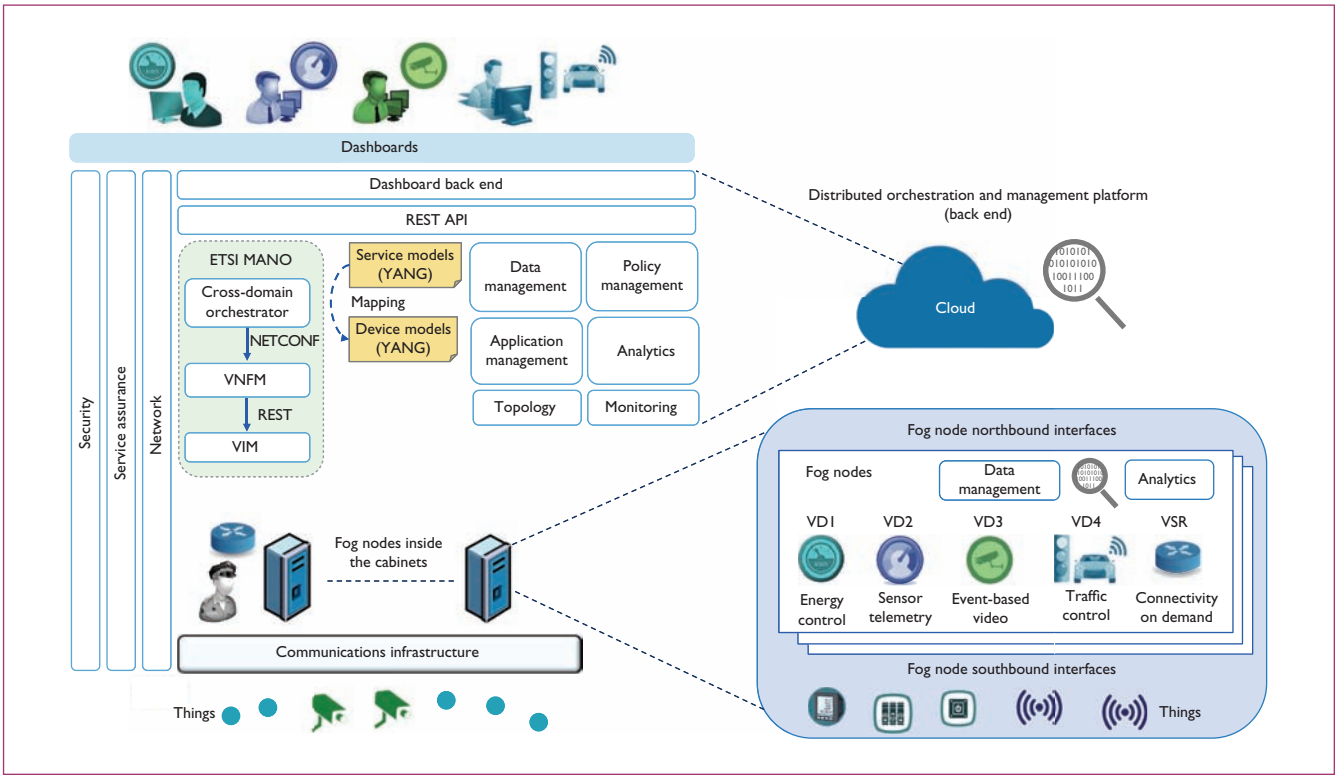

Figure 4. The architecture and its mapping to the case of Barcelona. The fog nodes on the right consolidate the five use cases covered in the project. (ETSI = European Telecommunications Standards Institute; MANO = Management and Orchestration; NETCONF = Network Configuration Protocol; VD = Virtual Domain; VIM = Virtualized Infrastructure Manager; VNFM = Virtual Network Functions Manager; and YANG = Yet Another Next Generation, which is a standardized data modeling language.)

configured and chained as a tenant-specific execution environment within a fog node.

Data management block. The data management block represents the internal buses, brokers, and data policy enforcement rules required to locally share and correlate data between specific microservices and applications. For data distribution and brokering, we deployed and tested both the Data Distribution Service (DDS) and RabbitMQ. We also used their multiprotocol capabilities to support Message Queuing Telemetry Transport (MQTT)-based applications. Regarding policy enforcement, we used Lightweight Directory Access Protocol (LDAP) in the back end and LDAP replicas at the fog node level. The replicas allow the fog nodes to continue authorizing the defined data sharing policies, even when backhaul connectivity is lost.

Analytics. Analytics can run locally, enabling autonomous operation and decision-making across VDs. The analytics also assist other key building blocks for managing urban services end to end, such as security, service assurance, and networking. In our implementation, we used Cisco ParStream (www.cisco.com/c/en/us/products/analytics-automation-software/parstream/index.html), an historian database that supports big data analytics in distributed deployments at very large scale. Different algorithms for application-specific analytics can be embedded in ParStream's core, thus enabling highly efficient ways to support analytics in multitenant environments, including service assurance, power control, and security.

Data sharing between different tenants was managed through ParStream's database, while data access control was handled through LDAP. In Barcelona, we demonstrated UCs that required data-sharing policies and access control among tenants at the fog node level. For instance, the cameras in Barcelona are managed by a specific administration (tenant *x*), while the sensors connected

through the cabinets are managed by another administration (tenant *y*).

As Figure 2 shows, we handled the data sharing workflows required to implement UC3 as follows. When processes running in VD2 (see Figure 4) detect an anomaly, an event is triggered (and stored in ParStream), which is consumed by VD3. In this example, an instance of tenant *y* (VD2) is the data writer, while an instance of tenant *x* (VD3) is one of the data readers. This is a part of a UC that requires anomaly detection, data privacy policies, local processing, and video streaming that will be sent through 3G/4G connectivity because the fog nodes are located in areas without wired connectivity.

Virtual switch/router (VSR). As previously explained, the fog nodes will consolidate and replace legacy networking gear within the cabinets. Therefore, the VSRs shown on the right in Figure 4 support the multiple networks required and act as a gateway for the applications running in the different VDs. Multiple independent instances of VSRs can run concurrently in a single fog node if required. In Barcelona, we used software routers in a virtual form factor, including Cisco's CSR1kv. For isolating the multiple networks involved, we used both virtual local area networks (VLANs) and a virtual extensible LAN (VXLAN).

Northbound interfaces. In the architecture that we conceived, all the fog nodes are provisioned with a standard Network Configuration Protocol (NETCONF) interface; the data modeling language we use in tandem with NETCONF is the IETF standard Yet Another Next Generation (YANG). NETCONF and YANG are widely adopted in industry, especially in the service provider and enterprise space. They're also an integral part of most efforts around network functions virtualization (NFV) and software-defined networking (SDN). However, the advantages of choosing NETCONF and YANG for fog computing go far beyond interoperability enablement, as they let us strategically create a common architecture for NFV, 5G, fog, and IoT. We followed this approach in Barcelona.

We created the first YANG models for a fog node, and for IoT services involving fog computing. All fog nodes used in Barcelona were endowed with Cisco ConfD, which not only supports NETCONF but also provides other key interfaces, such as command-line interface (CLI), REST, Simple Network Management Protocol (SNMP), and a Web user interface. All of these interfaces are automatically rendered thanks to the YANG models supplied with the fog nodes.

Southbound interfaces. The southbound interfaces depend on the sensor families and data collection elements to be connected. In Barcelona, the strategy we followed was to preintegrate with various vendors to cover the requirements of the different UCs implemented.

Back-End Cloud Support

The second group of components supports the back end in the cloud. Given the distributed nature of our model, some of the blocks in the back end might be hosted in the fog nodes and vice versa. The back end thus supports several components, including the following.

Orchestration system. As Figure 4 shows, the model here is the NFV Management and Orchestration (MANO) stack, which was recently standardized by the European Telecommunications Standards Institute (ETSI). ETSI MANO is based on three layers:

- an orchestrator that enables service definition and automation across three domains (fog, network, and data center),
- a Virtual Network Functions Manager (VNFM), and
- a Virtualized Infrastructure Manager (VIM).

In our architecture, we used Cisco tail-f Network Services Orchestrator (NSO) as the cross-domain orchestrator, Cisco Elastic Services Controller (ESC) as the VNFM, and OpenStack as the VIM. As Figure 4 shows, one of the cross-domain orchestrator's main roles is to map service definitions modeled in YANG to device-specific configurations based on device models that are also specified in YANG. The combination of ETSI MANO, NETCONF, and YANG is essential not only in terms of openness but also to enable a seamless convergence of cloud and fog.

Data management. This module works in concert with the ones hosted in the fog nodes and enables data workflows across city departments. Aspects such as data normalization, persistency,

Table 1. Validation of fog through use case requirements.

Use case (UC)	Aspects that require fog	The cloud alternative
UC1: Power monitoring	Autonomous operation and control of power supplies for critical services; real-time intervention	Impossible to ensure autonomous operation
UC2: Cabinet telemetry	Autonomous operation (access control must be guaranteed, even without backhaul connectivity)	Impossible to ensure autonomous operation
UC3: Event-based video	Poor connectivity, local data analysis, and anomaly detection	Unfeasible due to cost and practical reasons
UC4: Traffic management	Autonomous operation (intervene on traffic lights, even when backhaul network is unavailable); run complex algorithms to accurately monitor vehicle flows	Impossible to ensure autonomous operation; regulation prevents actuation on traffic regulators from the cloud
UC5: Connectivity on demand	Automated configuration of devices at the edge to accommodate connectivity requests	Can centrally control the configuration of standalone switches and routers, but at the cost of losing the practical advantages of box consolidation

and brokering are managed by the distributed data management blocks shown in Figure 4. As we mentioned, in our implementation, we mainly used DDS and RabbitMQ.

Application enablement framework. This framework manages the lifecycle of the applications instantiated in the VDs hosted by the fog nodes. It also manages other applications that are required for functions such as networking, security, and service assurance. Among these are the VSRs instantiated in the fog nodes, as well as other virtualized applications for malware detection, anti-virus, passive and active monitoring, and so on, each of which can be instantiated in the fog or in the cloud, depending on the service requirements.

Policy management. Together with adequate security, it glues data and application management functions. This component is key for endowing the architecture with the desired data-app centric functionality. As mentioned earlier, in our implementation, the policies for accessing resources and defining identities and roles are managed through LDAP.

Analytics. This module works in concert with the ones hosted in the fog nodes and enables historian analytics in a fully distributed way. In Barcelona, we used Cisco ParStream.

Additional functions. Other functions, including topology discovery and management, monitoring, and additional subsystems not depicted Figure 4 (such as operation and business support systems) are also an integral part of the back end. In addition, the back end offers a rich set of APIs that abstract entirely the internals and the complexity of managing the lifecycle of city services and the underlying infrastructure. The platform also lets us build dashboards for the different tenants. Through these dashboards, the appropriate authenticated user can define, deploy, update, and remove virtualized services, define access policies, create new tenants, get an overview of the deployments in the city, and so on. Our project demonstrated the different "views" of the platform from a tenant's perspective (for example, the department in charge of energy management can't access data from traffic sensors, and vice versa).

Cross-Domain Functions

In addition to the former two groups, three additional blocks are crucial to ensuring that the architecture is secure, remains connected, and offers services that are resilient to potential failures that might occur during their lifecycle (see the left side of Figure 4). For example, we configured the fog nodes using secure bootstrapping and zero touch provisioning processes, including remote attestations supported through trusted platform modules.

Validation and Lessons Learned

One of the goals of the Barcelona initiative was to validate the relevance and necessity of fog for cities. Although cloud is seen as the go-to

solution for many IoT-related challenges, we have come to understand — by talking to various cities and industries — that fog is crucial. Here, we discuss some of these findings, as well as lessons learned.

Table 1 shows the five UCs, highlighting the various requirements and the need for fog computing. The last column of the table describes, where possible, whether a viable cloud alternative exists for the requirements defined. As the table shows, fog is mandatory in four of the five UCs and is clearly desirable in the fifth for practical reasons, such as box consolidation.

Overall, the table highlights the necessity for fog to enable several smart city UCs. The project in Barcelona led to new insights into the impact of fog computing. Following are the most important lessons learned from this project.

Fog is Needed for Practical Reasons

Fog computing was introduced a few years ago in response to challenges posed by many IoT applications, including requirements such as very low latency, real-time operation, large geo-distribution, and mobility.[1] As Table 1 shows, the reality in cities is quite different at the moment. The drivers for fog computing today are mainly rooted in aspects such as autonomous operation, regulatory limitations, and box and service consolidation. Although the drivers for cities are currently different from those originally forecasted, the need for distributed computation at the edge is unquestionable.

Box Consolidation and De-silofication Are Key

Cities see the potential of connecting devices and sensors for their citizens, but they also recognize that they can't deploy new boxes (gateways) in the city each time a new solution is introduced. In the case of Barcelona, the cabinets have limited space, but even if cities mount boxes in light poles or walls, they face the aesthetics factor and a maintenance burden of managing multiple service silos, as described by the QS problem in Figure 1. In summary, box consolidation is a must to avoid the proliferation of siloed boxes from different solution providers.

Fog Offers New Business Models for Solution Providers

Fog reduces the need for these providers to build and maintain their own hardware, while extending the cloud to the edge means that fog providers can offer various "as-a-service" models (such as for analytics, security, and data sharing). These as-a-service models reduce the software and maintenance complexity for solution providers. Indeed, we've already confirmed that several third-party solution providers are willing to revise their strategy and possibly change their business models, as their (siloed) focus on hardware, security, and analytics has now migrated to the platform.

Platform Flexibility Leads to Innovation

Because of fog, cities can consolidate services on a single platform, thereby reducing the infrastructure and operational costs. One of the findings of our project was that, by lowering the complexity and incremental cost to deploy new services at the edge, cities are becoming more agile in deploying new services and are more inclined to innovate.

Pre-integration Is Essential

Service providers and system integrators play an important role in the digitization of cities, and they will be important partners when launching platforms like the one we describe here. However, no two cities are the same. While the premise of uniform service management, automation, and de-silofication is valid for all cities, the types of sensors and third-party applications used will differ from city to city. Pre-integration — that is, providing connectors to sensors and their protocols, and deployment of third-party services — is therefore key, and would enable service providers and system integrators to rapidly roll out such a platform in multiple cities and countries.

Fog Computing Isn't Limited to Constrained Devices

Fog computing, by virtue of the network's edge, is sometimes associated with constrained devices (limited CPU, memory, disk space, and so on). Our project showed that the cabinets in Barcelona can house multicore industrial PCs with multiple gigabytes of memory and up to terabytes of disk space, not only to enable the deployment of various virtualized services but also to plan for new ones. A spectrum of edge computing devices will emerge, ranging from constrained devices to full-blown data center-

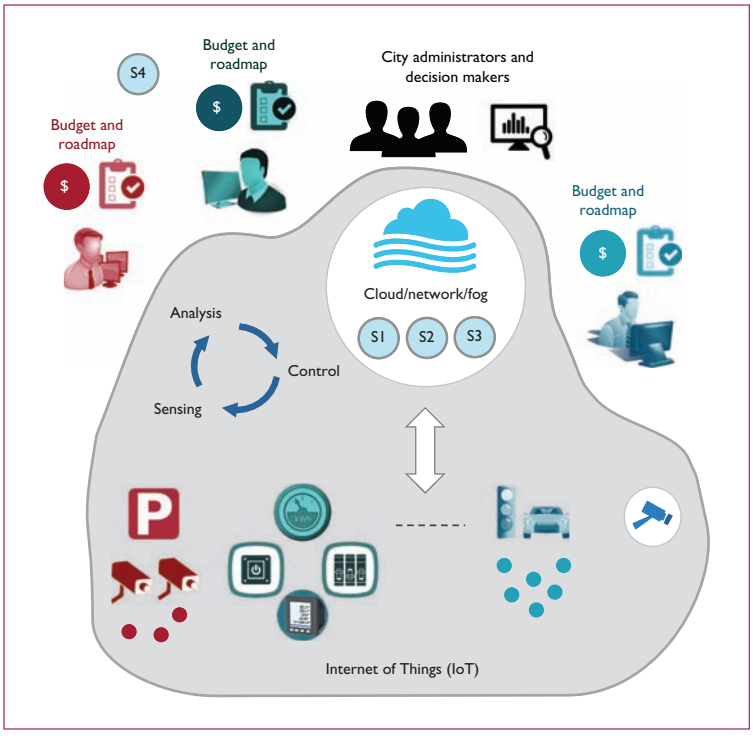

Figure 5. The proposed platform. By fusing cloud, network, and fog, this common platform breaks down silos S1, S2, and S3 in the QS problem.

like servers. One of the challenges going forward is to ensure that the platform we developed is capable of supporting this wide variety of edge devices.

The project carried out in Barcelona demonstrates that a platform like the one shown in Figure 5 can efficiently address the QS problem. We believe this approach can be applied to various other domains, including oil and gas, manufacturing, and utilities, and that it can also help address some of the main challenges that edge computing will face.[2] One of these challenges is to determine how data might be shared between many different processes running at the edge and the cloud, particularly to support functions such as smart offloading.[3]

Acknowledgments

We thank the city of Barcelona for its support during the project, as well as many other partners involved, including H. Antunes, R. Milito, I. Errando, J. Balagué, J. L. Ferrer, S. Figuerola, O. Trullols, K. Macpherson, J. Bienert, D. Eckstein, H. Werner, C. Byers, R. Zheng, R. Irons-Mclean, F. Osman, J. Greengrass, H. van't Hag, and K. Steele.

References

1. 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*, vol. 546, 2014, pp. 169–186.
2. W. Shi et al., "Edge Computing: Vision and Challenges," *IEEE Internet of Things J.*, vol. 3, no. 5, 2016, pp. 637–646.
3. M. Hassan et al., "Help Your Mobile Applications with Fog Computing," *Proc. Fog Networking for 5G and IoT Workshop*, 2015; doi:10.1109/seconw.2015.7328146.

Marcelo Yannuzzi is a principal engineer at Cisco's Chief 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. Yannuzi has a PhD in computer science from the Technical University of Catalonia–BarcelonaTech. Contact him at mayannuz@cisco.com.

Frank van Lingen is a technology strategist at Cisco Systems. His research interests include distributed systems, analytics, and fog computing within the context of the Internet of things. Van Lingen has a PhD in computer science from the University of Technology Eindhoven. Contact him at fvanling@cisco.com.

Anuj Jain is a director at Cisco's Chief 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. Jain has an MS in micro-engineering from EPFL. Contact him at januj@cisco.com.

Oriol Lluch Parellada is a technology strategist at Cisco Systems. His research interests include cloud infrastructure, fog computing, service orchestration, and artificial intelligence. Lluch Parellada has an MSc in telecommunications engineering from the Technical University of Catalonia–BarcelonaTech and an EMBA in management of technology from the EPFL and HEC Lausanne. Contact him at orilluch@cisco.com.

Manel Mendoza Flores is a network and security architect at the Barcelona City Council, where he's involved in multiple projects aimed at improving the quality of life of Barcelona's citizens using new technologies. Mendoza Flores has a BE in telecommunications from the Technical University of Catalonia–BarcelonaTech. Contact him at mmendozaf@bcn.cat.

David Carrera leads the Datacentric Computer Research Group at the Barcelona Supercomputing Center (BSC). His research interests include performance management of data-centric platforms. Carrera has a PhD in computer science from the Technical University of Catalonia–BarcelonaTech. Contact him at david.carrera@bsc.es.

Juan Luis Pérez is a senior research engineer in the Datacentric Computer Research Group at Barcelona Supercomputing Center. His research interests include the model-driven IoT. Luis Pérez has an MS in computer architecture, networks, and systems from the Technical University of Catalonia–BarcelonaTech. Contact him at juan.luis.perez@bsc.es.

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

Pablo Chacin is a researcher and practitioner with more than 20 years of experience in industry and academia. His areas of interest include distributed systems, software architecture, and middleware services. Chacin has a PhD in computer science from Technical University of Catalonia–BarcelonaTech. Contact him at pchacin@sensefields.com.

Angelo Corsaro is the chief technology officer at ADLINK Technology, where he looks after technology strategy and innovation for the company's Industrial Internet of Things (IIoT) platform. His research interests include large-scale mission/business critical distributed systems, real-time systems, functional programming, IoT, and fog and cloud computing. Corsaro has a PhD in computer science from Washington University in St. Louis. Contact him at angelo.corsaro@prismtech.com.

Albert Olive is a systems architecture expert at Schneider Electric. He has focused on several national and international projects to design control and power distribution architectures within Schneider. Contact him at albert.olive@schneider-electric.com.