



Edge and Cloud Pricing for the Sharing Economy

José María García, Pablo Fernández, and Antonio Ruiz-Cortés • University of Sevilla

Schahram Dustdar • TU Wien

Miguel Toro • University of Sevilla

As technology resonates in all layers of society, the impulse of shifting toward new spaces for a cooperative economy can be envisioned. Smart cities represent an ideal laboratory to design and explore new opportunities, offering a significant impact to citizens' lives.

Within the smart city scenario, researchers and practitioners have a concrete context where underlying challenges are well-established.¹ In this area, there's a growing technology stack that's aligned with Internet of Things (IoT) principles and benefits from big data techniques. Specifically, this clear vision² has paved the way to incorporate *edge devices*³ and complex real-time analytics. In such a context, we've seen a growing number of projects (<https://eu-smart-cities.eu>) embrace these challenges, addressing problems such as the management of multiple IoT providers by city authorities (the VITAL Project; <http://vital-iot.eu>) or procuring sustainable growth in the city (the GrowSmarter Project; www.grow-smarter.eu), with very promising outcomes in scenarios such as traffic control, environmental quality, and sustainable energy, among others.

Moreover, in the next several years, the likely advent of subtle economy shifts toward a growing "servitization" of society opens new opportunities to advance into evolved smart cities scenarios where public organisms, private corporations, and citizens join forces to create cooperative ecosystems that work together to improve society in novel ways. Examples of this potential can be outlined in different problem dimensions:

- To improve efficiency with actions involving new ways of dynamic traffic management

based on real-time citizens' data, predictive forecast of traffic flows, or optimization of public transportation routing.

- To improve well-being by devising new ways of boosting collaboration among citizens by efficiently sharing their resources and time, or citizen awareness to enforce a collaborative city services quality control.
- To improve environmental quality with measures such as pollution prediction and minimization based on dynamic actions by citizens, or improving energy redistribution by means of a coordinated Net Energy Metering (www.seia.org/policy/distributed-solar/net-metering).

Despite the technology's maturity, with different solid IoT platforms and frameworks currently available, these opportunities are tightly coupled with two open challenges: On the one hand, the lack of a generalized business model for public-private collaboration involving citizens that boosts the creation of cooperative ecosystems; and on the other hand, the lack of tools to design and operate solutions taking into account aspects such as risk, quality of service (QoS), or costs, so that all involved parties are aware of their benefits and responsibilities in the collaboration.

Here, we embrace the principles of the sharing economy to outline a framework to design and operate cooperative ecosystems for both public/private

The Role of Operational Agreements

Agreements play a crucial role in society, where they act as the trustworthy connectors among peers in business transactions regulating stakeholders' rights and responsibilities.

During the last decade, academia has boosted a shift toward formalizing the electronic contracting process, where operational agreements are designed, analyzed, and enforced in an automated way. As part of this shift, different works have spotlighted key elements such as the agreement structure^{1,2} or how they're managed.^{3,4}

To develop a realistic, fully-fledged agreement, various dimensions should be tackled:

- The metrics dimension, where different service properties are identified and described jointly with a measurements framework that includes its granularity, temporal intervals, and units. Specific metrics calculators could be highly coupled with domain-specific elements or be agnostic to general service properties.
- The agreement's temporality dimension, which can include elements such as the general availability of the whole agreement or the time zones involved in the measurement of the different terms.
- The pricing dimension, where cost-aware elements are incorporated, including different purchasing options (such as pay per use or recurring plans), the billing process of the service, or a rich cost model with discounts; in this context, it's worth noting that during recent years, blockchain technologies have emerged as an interesting promising framework to boost trustworthy transactions involving cost.
- Limits dimensions that define the operational boundaries allowed for a specific user. These limits could include quotas (attached to fixed static temporal windows) or rates (attached to dynamic windows) over specific metrics.
- A guarantees dimension, where service-level objectives are defined. These objectives should ideally be bounded to spe-

cific penalties and rewards in case the objectives are either under- or over-fulfilled.

- A legal dimension, where the different lawful aspects are included, such as the regulation framework, the disclosure and liability clauses, or the claiming procedure in case parties are subject to compensations.

To address the agreement's different dimensions, we foresee the need for an expressive coherent model to incorporate the different elements in a common processable artifact. With such a model, researchers could develop a catalog of analysis operations that could process the information of different agreement offerings. In turn, these offerings would support the creation of automated management tools that handled risk and operations to improve and boost the contracting process.

These open challenges represent an interesting horizon where explicit operational agreements are leveraged as first-class citizens that contain the knowledge to regulate an automated governance of IT infrastructures. In the context of a sharing-economy scenario, embracing such an agreement model would represent a valuable impact, because the richer the operational agreement models (and tools), the more dynamic markets could become.

References

1. A. Andrieux et al., *Web Services Agreement Specification Draft, Version 09/2005*, 2005; www.ogf.org/documents/GFD.107.pdf.
2. J.M. Garcia et al., "Modeling Service Level Agreements with Linked USDL Agreement," *IEEE Trans. Services Computing*, vol. 10, no. 1, 2017, pp. 52–65; doi:10.1109/TSC.2016.2593925.
3. S. Hudert et al., "Negotiating SLAs — An Approach for a Generic Negotiation Framework for WS-Agreement," *J. Grid Computing*, 2009, vol. 7, no. 2, 2009, pp. 225–246; doi:10.1007/s10723-009-9118-3.
4. C. Muller, M. Resinas, and A. Ruiz-Cortes, "Automated Analysis of Conflicts in WS-Agreement," *IEEE Trans. Services Computing*, vol. 7, no. 4, 2014, pp. 530–544; doi: 10.1109/TSC.2013.9.

organizations and citizens, taking into account the different stakeholders' business model. We exemplify this framework in a particular scenario of environmental quality, where a local public administration is interested in analyzing environmental quality conditions throughout different neighborhoods of a city. In this use case, the public administration needs a computing infrastructure to analyze data obtained from IoT platforms and edge devices belonging to individual neighbors, in order to develop policies and concrete actions to improve the

city's environmental quality and prevent risky situations for citizens when high pollution episodes occur, for instance.

In the following, we first discuss a framework that enables this sharing economy-based scenario, outlining the models and processes taking place within the proposed cooperative ecosystem. Then we present a realization of this framework through its application to the introduced environmental quality improvement use case. Finally, we conclude the article by discussing the foresight of these cooperative

ecosystems that facilitates the sharing economy in the smart cities domain.

Enabling the Sharing Economy in Smart Cities

To realize this motivating scenario, we propose a framework that allows interested parties to both design their solutions and provision the needed components to actually cooperate and obtain their corresponding pursued benefits. As Figure 1 shows, the proposed ecosystem integrates diverse platforms and devices, ranging from

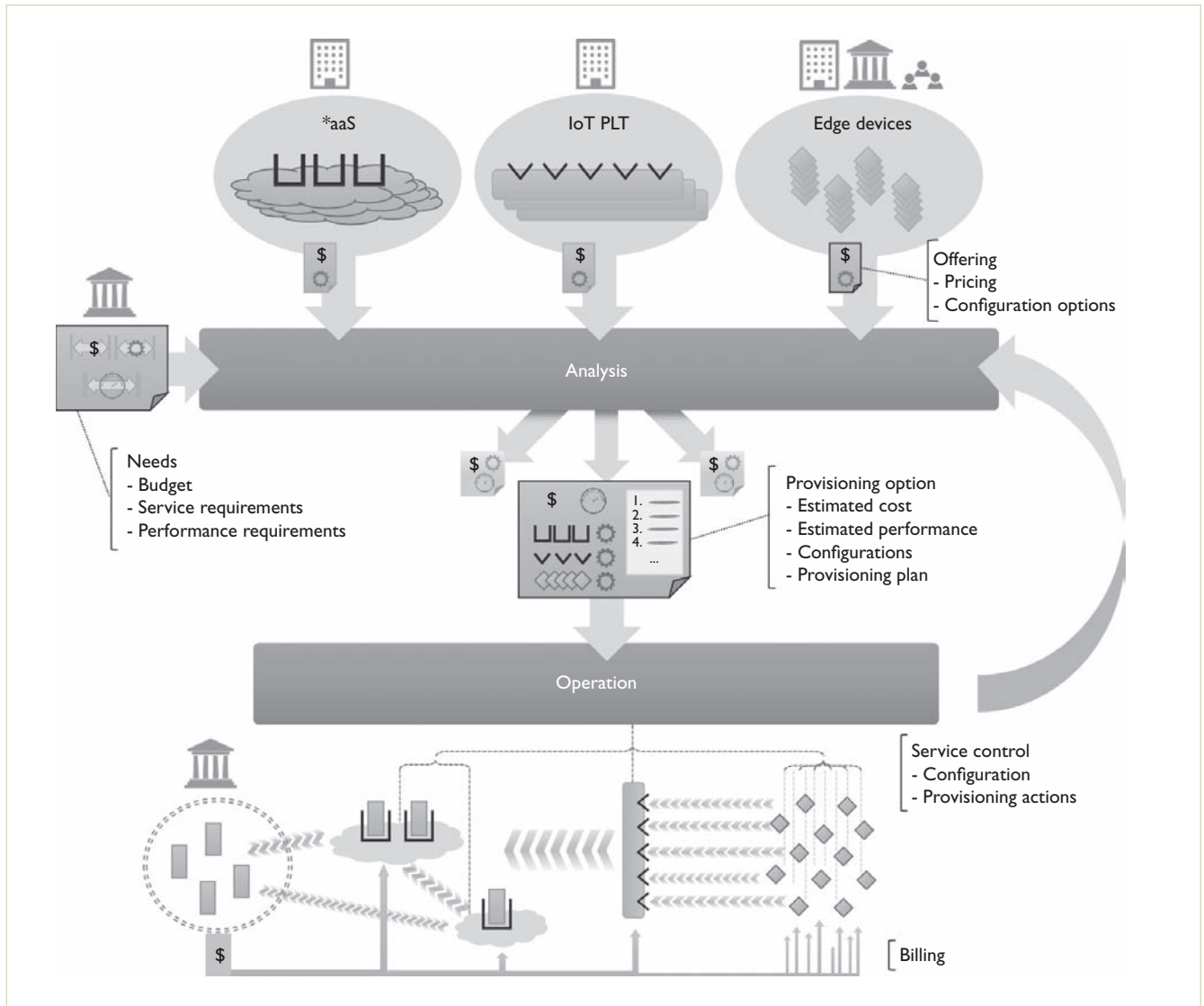


Figure 1. Ecosystem for public and private collaboration. IoT PLT stands for Internet of Things platforms.

third-party IoT platforms and cloud infrastructures for complex data analytics to edge devices owned and operated by different stakeholders.

Each party cooperating in this ecosystem has to provide information concerning their needs and offerings, in order to analyze them to obtain the different provisioning options for achieving their goals. The framework we describe focuses only on pricing and budget analysis, considering the various pricing plans offered by the needed services that best suit the project budget. Nevertheless, additional analysis operations involving risks

or legal aspects can be implemented, considering more generic information models such as agreements (see the related sidebar for further details). In the following, we discuss in detail the ecosystem shown in Figure 1, focusing on the analysis and operation phases.

Analyzing Offerings and Project Needs

The main entry point to our framework consists of the information about each component that every partner has to feed into it. On the one hand, providers must publish their service offerings so that interested parties can

discover them. The framework supports the registration of services provided by public administrations and private parties, including companies and individual citizens. As Figure 1 shows, offerings not only include the service capabilities in terms of their different configuration options, such as computational power or available sensors, but also their pricing details, which might include various plans so that a potential user (such as a city council) can choose a suitable pricing plan depending on their expected usage or performance needs.

For instance, a citizen willing to share data collected from their owned edge device, which comprises sensors such as a thermometer and a hygrometer, might register it as a service providing temperature and humidity measurements of its location. In addition to providing the service configuration information, the citizen might describe different pricing plans, each defining the following options:

- measuring frequency (for example, by the hour, minute, and second);
- billing frequency (monthly, weekly, and hourly);
- billing scheme (pay-per-use or reserved instances); and
- prices (such as unit costs and reservation fees).

On the other hand, project needs specify the public administration's requirements on particular services, which in our use case range from cloud services (*aaS in Figure 1) to specific edge device services, that are needed to implement a city improvement project such as the environmental quality use case introduced before. These requirements mainly state first, the particular *service requirements* (for example, computational power, memory, or sensor features) that are needed to achieve the client's goals; second, the *performance requirements* for each service, including the expected usage schedule, frequency of operations, and density of sensors and spatial distribution; and possibly third, the available *budget*.

Figure 2 shows an example of the project needs using a YAML serialization of the underlying model of our use case scenario. Essentially, *needs* are composed of a series of services that represent the different components that a user (for example, a city council) needs to put in place to achieve their goals. The left side of Figure 2 describes requirements for the infrastructure as a service (IaaS) platform needed to analyze the sensor data,

id: SevilleCityCouncil	- id: TemperatureSensors
services:	configuration:
- id: CityTemperatureAnalyticsIaaS	category: sensor
configuration:	sensorFeatures:
category: IaaS	- thermometer
memory: 4	schedules:
cpu: 2	- id: HourlySchedule
schedules:	window:
- id: AnalysisSchedule	frequency: hourly
window:	instances:
from: '2016-10-11T08:30:00.000Z'	density: 3
duration:	distribution: uniform
value: 12	location:
unit: hours	latitude: 37.3577205
frequency: daily	longitude: -5.9856252
instances: 1	

Figure 2. User needs excerpt in YAML. The left side describes requirements for the infrastructure as a service (IaaS) platform needed to analyze the sensor data, including the usage schedule. The right side specifies sensor services, where the user can also state the usage schedule and the required distribution of edge devices.

including the usage schedule. Sensor services are specified on the right side, where the user can also state the usage schedule and the required distribution of edge devices. Each type of service is associated with its required configuration, which describes the hardware requirements for each requested instance. Regarding the expected usage schedule, each service enumerated in the user needs is associated with one or more scheduling items, which are *temporal composites* that detail the validity periods and number of instances of the same configuration that are needed. The validity period of a scheduling item is a time interval that might be periodic, and possibly disjointed or overlapped with others. Figure 2 only shows a simplified version of these scheduling items, but they can be more expressive, allowing clients to specify complex scheduling constraints and global validity periods.

By analyzing the specification of the project needs, the framework searches for suitable cloud services,

IoT platforms, and edge devices to obtain different provisioning options. These options are computed by considering the user's budget limitations, as well as the various pricing plans from *aaS, IoT platforms, and edge devices registered in the framework. Performance and scheduling restrictions are also taken into account, resulting in specific provisioning plans, along with their corresponding service configurations. After the user chooses the most suitable provisioning option, the different components can be deployed and operated according to the derived plan.

Operational Support

The framework we describe in this article also supports the automatic operation of the different cloud services and edge devices needed to fulfill the project needs. As described before, a service *provisioning plan* is derived from the described needs in the analysis phase, taking into account scheduling and budgetary restrictions. This

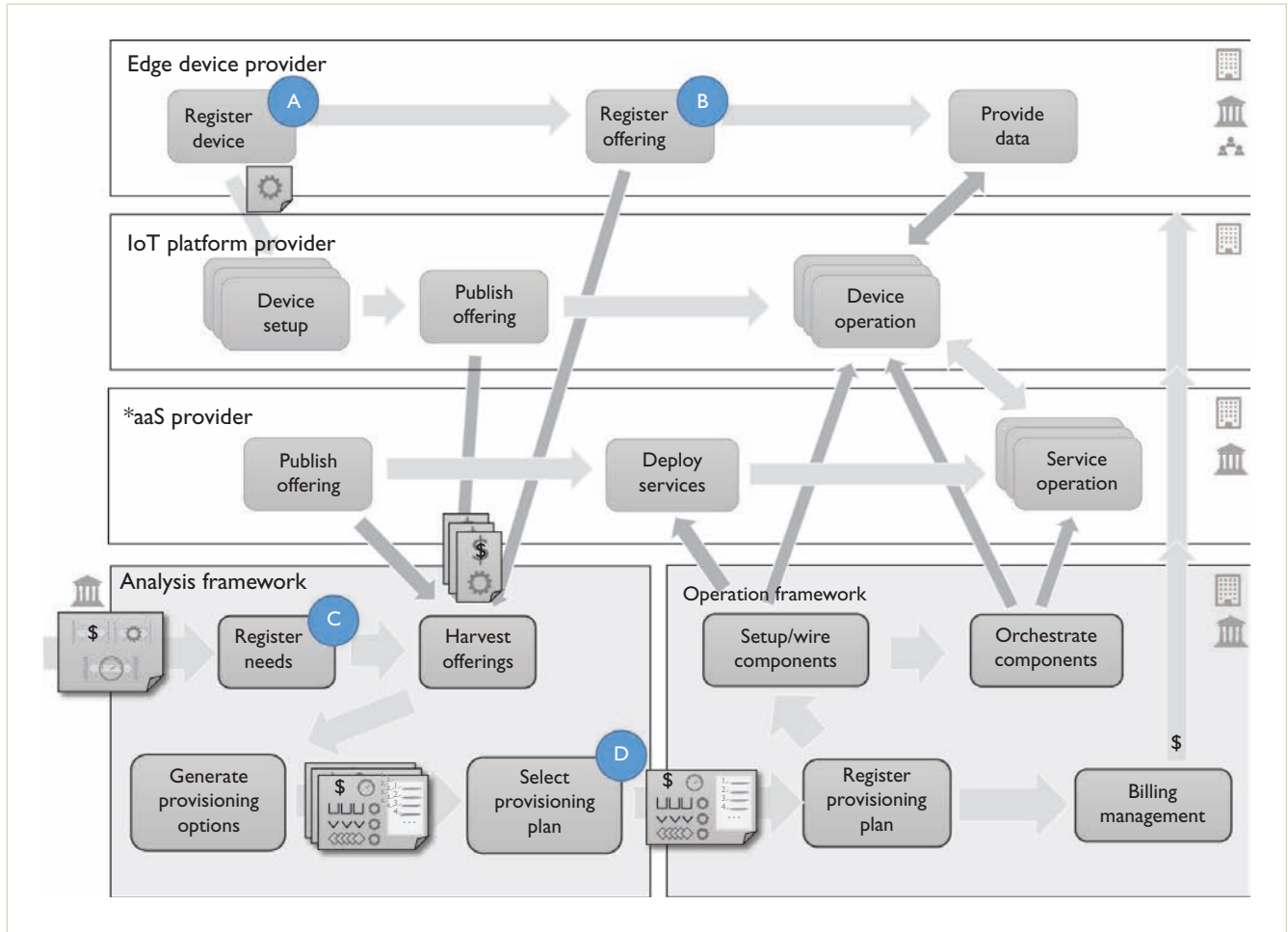


Figure 3. Dynamics of the providers and framework. The items labeled A, B, C, and D are discussed further in Figure 4.

Table 1. Provisioning actions catalog.

*aaS	Internet of Things (IoT) platform	Edge device
Purchase/reserve instance	Purchase/reserve stream	Purchase/reserve instance
Deploy service	Start stream	Start data reception
Undeploy service	Stop stream	Stop data reception

plan can be used both to deploy and to orchestrate the different services involved in the solution.

The particular provisioning actions depend on the nature of the service to be deployed, as outlined in Table 1. We consider that any component of the ecosystem might be purchased or reserved according to their pricing plans. This purchase action is usually needed when the analysis detects that

for deploying a particular service, it's better to reserve some instances with the required configuration rather than using an on-demand or pay-as-you-go purchasing option. In turn, when the expected usage schedule isn't long enough, the instances can directly be started when needed, without including a purchase action in the provisioning plan. Because purchasing options (pricing plans) depend on each service

provider, the framework optimizes provisioning plans depending on the available choices for each provider, including additional information when needed, such as the type of reservation and the purchase term, among others.

Each provisioning action refers to the concrete instance type (that is, configuration) that fulfills the service requirements, while the performance requirements dictate the necessary series of provisioning events that specify the time instant when the actions will be performed. An orchestrator or service controller (as showcased in Figure 1) is responsible for the correct execution of the provisioning plan. Furthermore, this execution entails a series of billing events, according to the pricing plans of the corresponding cloud services, IoT platforms, or edge devices that conform to the solution.

Finally, the framework also supports a dynamic adaptation to changes in the ecosystem. Thus, when a particular service stops working or its features or pricing plans change, the analysis phase must be executed again, to find alternative provisioning options to continue satisfying the user's needs. Nevertheless, user needs might also change in time; hence, the need to start the cycle, analyze them, and obtain a suitable provisioning plan to operate the solution.

Realization and Application of the Framework

Figure 3 highlights the different dynamics of the scenario depicting the activities and information flow among the main participants. Specifically, as introduced previously, three different kinds of providers are identified: edge device providers, IoT platforms, and *aaS providers.

Edge device providers are the actual owners of sensors that could provide valuable data in a certain location of the city. These providers have a three-step lifecycle, starting with the registration of their devices in a particular IoT platform. Then, they register their offering in the analysis framework, describing their different usage and pricing plans. Finally, they start providing the data to the platform. From an organizational perspective, edge device providers can correspond with either public entities (for example, the traffic or air sensors in public spaces), private corporations (such as closed-circuit television from banks), or citizens offering their own devices (such as weather sensors).

IoT platforms typically correspond with infrastructures that give support for receiving and managing a continuous flow of data from a variable set of sensors. These platforms provide functionality for defining reliable and consistent streams of information jointly with domain-agnostic analytical capabilities for events processing and aggregation. From an organizational perspective, IoT platforms are normally

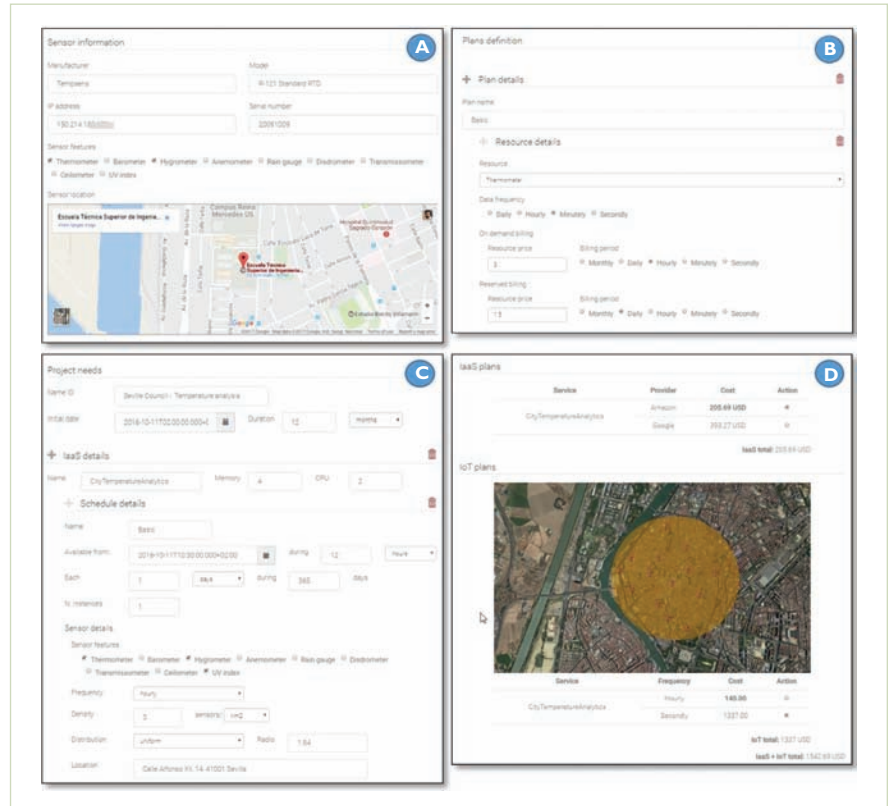


Figure 4. Tooling screenshots. These screenshots can be grouped according to different roles in the scenario: A/B for the edge device provider perspective and C/D for the public institution using the analysis and operation framework.

provided by private corporations, such as Amazon Web Services' (AWS) IoT or IBM Watson IoT.

The *aaS providers represent an abstraction to group different kinds of services: from a low-level IaaS to a higher-level of software as a service. As an example, the traditional IaaS providers publish specific offerings with different instance configurations and pricing models (such as on-demand or reserved usage). Next, in order to use the instances, a deployment process is performed, where the software assets are bound to the different instances. Finally, the provider typically offers a control API to launch the service operation (for example, to start up or shut down the instance). From an organizational perspective, the providers can be private institutions (such as AWS, Google Cloud, or IBM Bluemix) or public institutions with their own cloud infrastructure.

The analysis and operation framework proposed (see Figure 3) interacts with the different providers to give support for a specific city improvement project need. Specifically, the needs are registered (typically by the city council) into the analysis framework containing the budget, performance, and scheduling constraints. Next, based on the offerings harvested from the providers, the provisioning options are generated; each of these options contain an estimated cost and performance, along with the configuration and provisioning plan needed to operate the ecosystem to support the project. Once the appropriate option is selected by the council, the operation framework comes into play and the provisioning plan selected is enforced in two different stages: first, the setup phase, where the configuration is leveraged to the providers so each component is wired to create the ecosystem; and second, the orchestration

phase, which takes advantage of the control APIs of the IoT platform and the services to coordinate the ecosystem's behavior. Furthermore, the operation framework is in charge for the billing management to register different transactions based on the pricing terms defined in the provider offerings.

To exemplify the scenario in a particular domain, Figure 4 showcases four different fragments of screenshots from the tooling that relate to the activities highlighted in Figure 3. These screenshots can be grouped according to different roles in the scenario: A/B for the edge device provider perspective and C/D for the public institution using the analysis and operation framework. On the one hand, screenshot A shows a particular case of an edge device registration that includes two sensors (a thermometer and a hygrometer), which are located in some specific geocoordinates. Then, screenshot B shows how the different pricing plans are defined for the device; for example, two billing models (on-demand and reserved) with alternative pricing depending on the data frequency. On the other hand, screenshot C shows a particular project's needs specifying some IaaS and sensor requirements. Finally, in screenshot D, a set of provisioning options (along with their cost) is presented to the user outlining the IaaS configurations from the different providers as well as the alternatives for the IoT platform and the suitable edge devices located.

We envision a collaborative ecosystem where edge devices can be shared by their owners, obtaining a remuneration after their use by interested parties. The definition of their pricing plans and their publication on integrated solutions such as the framework we proposed in this article, will open many opportunities to leverage the sharing economy. We outline the application of these principles to smart cities scenarios, in particular to the environmental quality use case, where we already have implemented a

preliminary prototype that showcase the analytics and operational capabilities of these scenarios.

Nevertheless, there are still some open challenges that remain to be addressed in order to fully implement the sharing economy vision within this domain. First, possible integrations with existing IoT platforms, such as Thingspeak, Xively, or FIWARE IoT enablers must be considered, because extensions to support pricing plans for edge devices are needed to enable a marketplace for sharing data. Second, autonomous operational support is a promising research area, where automatic solutions to deploy and orchestrate the complete ecosystem can be developed. There are standard models, such as the Topology and Orchestration Specification for Cloud Applications (TOSCA), that can be of use in this area,⁴ though dynamics and self-adaptation remains to be tackled. Finally, there are other future directions that pose additional open challenges, such as the integration of blockchain technologies, or the applicability of the proposed framework to similar domains such as disaster prevention. □

Acknowledgments

This work has been partially supported by the European Commission (FEDER), and the Spanish and Andalusian R&D&I programs under grants TIN2015-70560-R and P12-TIC-1867.

References

1. J.M. Schleicher et al., "Towards the Internet of Cities: A Research Roadmap for Next-Generation Smart Cities," *Proc. ACM 1st Int'l Workshop on Understanding the City with Urban Informatics*, 2015; doi: 10.1145/2811271.2811274.
2. A. Zanella et al., "Internet of Things for Smart Cities," *IEEE Internet of Things J.*, vol. 1, no. 1, 2014, pp. 22–32; doi:10.1109/JIOT.2014.2306328.
3. J. Biron and J. Follett, *Foundational Elements of an IoT Solution: The Edge, the Cloud, and Application Development*, O'Reilly, 2016.
4. T. Binz et al., "Portable Cloud Services Using TOSCA," *IEEE Internet Computing*, vol. 16, no. 3, 2012, pp. 80–85, 2012; doi:10.1109/MIC.2012.43.

José María García is an accredited associate professor in the Applied Software Engineering Group at the University of Sevilla. His work focuses on service-oriented computing, semantic technologies, and agreements. García has a PhD in computer science from the University of Sevilla. Contact him at josemgarcia@us.es; www.isa.us.es/members/josemaria.garcia.

Pablo Fernández is an accredited associate professor in the Applied Software Engineering Group at the University of Sevilla. His work focuses on service agreements, automated trading, and business processes. Fernández has a PhD in computer science from the University of Sevilla. Contact him at pablofm@us.es; www.isa.us.es/members/pablo.fernandez.

Antonio Ruiz-Cortés is a full professor of software and services engineering and he heads the Applied Software Engineering Group at the University of Sevilla. His current research focuses on service-oriented computing, software product lines, and business process management. He's an associate editor of *Springer Computing*. Contact him at aruiz@us.es; www.isa.us.es/members/antonio.ruiz.

Schahram Dustdar is a full professor of computer science, and he heads the Distributed Systems Group at TU Wien. He's currently a visiting professor at the University of California, Berkeley, and was a visiting professor at University of Sevilla, Spain. His work focuses on distributed systems. Dustdar is an IEEE Fellow, a member of the Academia Europaea, an ACM Distinguished Scientist, and recipient of the IBM Faculty Award. He is on the editorial boards of *IEEE Internet Computing* and *Computer*. He's an associate editor of *IEEE Transactions on Services Computing*, *ACM Transactions on the Web*, and *ACM Transactions on Internet Technology*. He's the editor-in-chief of *Springer Computing*. Contact him at dustdar@dsg.tuwien.ac.at; <http://dsg.tuwien.ac.at/staff/sd/>.

Miguel Toro is a full professor in the Department of Computer Languages and Systems at the University of Sevilla. His current research focuses on service-oriented computing, software engineering, and simulation. He has received numerous awards, including the Fama Award from the University of Sevilla in 2012. Contact him at migueltoro@us.es.