Framework Blueprint for Building Energy Management Systems

João Pinho
Instituto Superior Técnico, Universidade de Lisboa, Portugal
joao.pinho@tecnico.ulisboa.pt

ABSTRACT

Buildings are major energy consumers of today. A solution for optimization of building energy consumption implies coordinating devices in order to maximize energy efficiency. However, the control logic underlying such techniques is currently implemented on hardware controllers commissioned by proprietary solutions, thus limited and difficult to integrate with other systems. Therefore, integrating multiple energy efficient control techniques is usually more complex and expensive. Moreover, while it is technically possible to connect devices directly to TCP/IP networks, most Home and Building Automation Systems (HBASs) still use non-IP field level communication protocols. In this context, no proper framework for centralized energy control systems has been documented in literature. In this work we present a platform over which applications featuring energy-efficiency techniques can be built on. The proposed solution has been validated on a real-world automated scenario to attest its practical applicability.

1. INTRODUCTION

One of the greatest challenges faced by software developers when creating energy-efficient and control software is to create an appropriate abstraction on software [1, 2]. In the particular case of energy control software, different devices might be working on different protocols within the same building. Therefore, installing an energy control software in a building can be hard and sometimes expensive, since it has to be compatible with multiple communication protocols [3]. Moreover, energy-efficient and control techniques are mainly implemented in hardware controllers, therefore, an abstraction of the energy-efficient and control firmware is not an easy task.

Techniques such as like day-lighting—achieved by dimming luminaries that are not needed during daylight [4]—reduce up to 40% of buildings energy consumption [5]. Coarsely speaking Heating, Ventilation and Air Conditioning (HVAC) systems are responsible for around 70% of the energy requirements of a building, while the remaining are mostly spent on lighting systems [6, 7, 8].

Building Automation Systems (BASs) aim at improving user comfort while using resources more efficiently, in particular the consumption of energy. A BAS consists of a distributed control system working over a digital network of devices designed to monitor, control and enable a more adequate building automation according to user requirements. Buildings that use BAS are commonly known as Smart Buildings (SBs).

Building Energy Management Systems (BEMSs) combine both intelligent and *"green"* buildings technology. To achieve being intelligent, energy-saving, *"green"* and other environmental objectives, an unified management of energy consumption is designed to improve equipment operation and reduce energy consumption [9, 10].

A centralized energy control and monitoring software will give us the ability to apply energy-efficient and control techniques even if the back-end building automation systems do not support it. This way it becomes possible to seamlessly develop and deploy applications committed to maximize energy efficiency across different buildings, despite their technology heterogeneity.

This framework will consist of a software layer residing above the individual systems and their specific protocols. Its role is to provide an uniform method for accessing data from, and issuing commands to, the various building devices. Therefore, the framework must be protocol agnostic and vendor neutral, supporting all devices equally. Web services which are emerging Information Technologies (IT) give us an elegant way to solve some of the above problems [11].

This work approaches the problem from the pragmatic perspective that, monitoring our buildings through the usage of sensing and actuation devices should be as natural and easy as humanly possible. If we think of a building where each room is equipped with energy metering, sensing and actuator devices, such framework should be able to collect all metering data, provide a simple way to query this data and give actionable and straightforward methods to process it.

Such capabilities would enable us to draw conclusions in real-time for later analysis, and also would give us the ability to deploy energy efficiency techniques that could actually be measurable, since the operational baselines of our buildings would be stored outside the deployed BAS solutions.

According to our knowledge no framework that abstracts energy-efficient control techniques through software applications has been proposed so far. To address our research, this work was conducted and validated in a real world environment by deploying it on the facilities of the University of Lisbon (Taguspark campus).

In the following section we analyze the requirements of building automation and energy management systems. Existing middlewares for BAS, commercial tools and also BEMS standards from the literature are discussed in Section 5. Section 6, describes the solution for our framework and its architecture. Our framework is then evaluated throughout three practical case studies (Section 7). Section 8, presents the conclusions and implications of our work.

2. BUILDING AUTOMATION SYSTEMS

A BAS provides means to control and monitor indoor environment conditions, either manually or automatically, through a grid of smart devices ranging from sensors, actuators and controllers. The historical root of BAS was the automation of HVACs, providing means to control and adjust their parameters in order to maximize user comfort. Although, the reach of BAS has been further extended to include all kinds of BASs such as lighting, security, chiller and steam systems, among other [12].

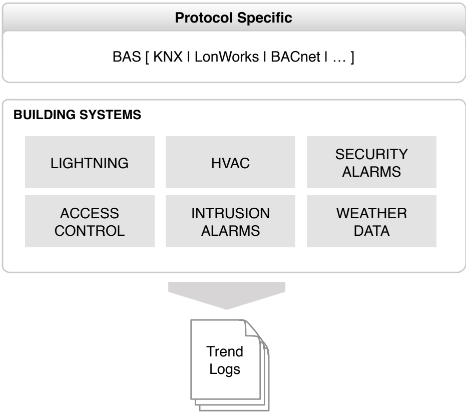

Figure 1: Overview of building automation systems and their subsystems - On the top, the protocol specific layer presents heterogeneous BAS protocols. Bellow we have BAS subsystems, programmed for a specific field level protocol. At the bottom, trend logs record data produced by these subsystems.

Besides enabling automation of electric components, BAS can provide data such as electric consumption, temperature and heat radiation through monitoring on precise readings from available equipment, generating trend logs (Figure 1) and information charts.

A common issue regarding BAS consists on the difficulty of deploying several distinct solutions inside the same facility [13]. For instance, in the University of Lisbon (Taguspark campus), there are at least three commissioned BASs running in parallel for different purposes. This type of scenario poses a problem related with BASs interoperability, since BASs are by nature heterogeneous.

The heterogeneity between different BAS technologies can be troublesome, because they hinder facility managers work, by forcing them to collect data from different systems into a consolidated data warehouse [13].

Also, collecting data only from monitoring points, typically is not enough to determine what to improve or the sources of a problem, such as over consumption of some electric equipment [14]. Furthermore, facility managers are left with inaccurate and small amounts of relevant data, when metering systems are installed instead of a BEMS. [14].

According to *Interval Data Systems*——a well known software house of BEMS solutions—it is not unusual for large facilities to have 30,000 points or more (a large college campus can easily have well over 100,000 points) [15]. With such a large number of data collection points (Figure 2), each generating one record every 15 minutes, the total data set comprises over a billion records per year [14, p. 70]. Such amount of data can be unfeasible to manage by facility managers, thus leading to huge amount of data being discarded.

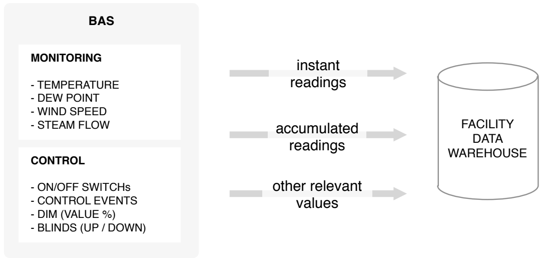

Figure 2: An example of monitoring and control points being collected from a BAS (left figure). Those points represent instant or accumulated readings processed and stored into a data warehouse (right figure).

Another topic of concern is trending data, for every monitoring and control point collected there is associated data related with it, data such as weather, maintenance tasks and other facility operations [14].

3. ENERGY MANAGEMENT SYSTEMS

An Energy Management System (EMS) is a “set of interrelated or interacting elements, to establish an energy policy and objectives, together with processes and procedures to achieve those objectives” (ISO 50001:2011) [16]. Energy Management Systems provide a set of tools that enables the monitoring of facility operations through the gathering of all building data related to environment and equipment operations. Moreover, they help increase user awareness on how building is performing [17].

The BEMS makes data available so that end users are able to perform in-depth diagnostics, analysis, and monitoring in a fraction of the time they would took with earlier methods. The resulting information enables building owners to make radical improvements in their systems operations, staff deployment, and information dissemination inside and outside of their buildings [14].

Being made of a consistent set of monitoring and analytical tools, an EMS covers the following main aspects of a building [9, 14]:

Used utilities data: All operational-related (electricity, natural gas, chilled water and other applicable) data must be gathered and consolidated into a data warehouse [18].

Data normalization: The acquired data must be normalized and restructured in a consistent data model in order to be usable.

Interactive and actionable data: All the collected data must be presented in an intelligible and actionable format to the user. By actionable we mean that, the facility manager, must be able to act based on the data gathered from the BASs.

Measurement and verification: Performance measurement and verification provides the ability to compare energy use, before and after, some steps have been taken to address potential energy savings.

3.1 Data Acquisition

Being data a core component of any analytical and monitoring tool, an EMS needs to know how to collect information from all the sources of data available in a facility [19]. Data can come from metering systems, BAS, utility reports (requiring manual data insertion) or from space management systems [14, 20]. Another aspect that these systems must take into account are billing rates, besides collecting energy related operational-data an EMS must be able to translate such data into meaningful monetary values.

It is very common for a medium-large facility to have more than one billing-rate for different buildings. Also in order for an EMS be able to correlate consumptions with such billings rates, a data acquisition process must be studied first [14, 21]. Most utility vendors have their reports available through electronic formats, for instance EDP—a Portuguese electrical company—makes those billings available on-line through a web-based system, other like gas or water companies make those reports available through paper-based support, sent monthly or bi-monthly.

Thus the best way to gather consumptions is to install smart meters for the various utilities directly on buildings [14, 21].

3.2 Utilities & BAS data

In large corporations, featuring more than one building, is very common to have buildings generating utilities, such as, chilled water for air conditioning and steam. Those utilities either purchased or generated, produce data that must be acquired, processed and normalized into a data warehouse. An EMS must be able to relate this data with billing rates and other factors, such as, weather data and calculated data. For instance, for steam, is very common to compute flow and condensation return values [14].

Nowadays, it is common for a facility to feature one or more BAS to control HVACs, lighting, weather stations, along with other building automated systems. It is important to notice the distinction between BAS and EMS systems — BAS are responsible for controlling HVACs not EMSs, as EMS are not control systems. EMSs provide intelligent and actionable data to their facility managers while BAS provide them ways to control and monitor them.

3.3 Metering Systems & Weather Data

Metering Systems are used to measure utilities consumption. Usually they store the collected data on their own dedicated database systems [14, 21]. An EMS should be able to read data from those database systems or from meters directly when appropriated. When an EMS is installed those metering systems—if not already—should be configured to provide not only instantaneous readings, but also totalized values, enabling the EMS to fill the gaps, between failed readings, with the totalized values [22].

Weather data is also an important variable to take into account, as it improves the accuracy of energy forecasting models. Weather data also enables an EMS to understand the context around a period of consumption. For instance, during cold or hot days HVAC systems are expected to consume more or lesser energy [23].

4. MIDDLEWARE FOR BAS

Middleware has been defined as a distributed system service that includes standard programming interfaces and protocols [24]. These services are called middleware because they act as a layer above the device-specific and networking software and below business applications. Umar [25] presents an extensive treatment of the subject.

Architecture research regarding middleware focuses on the problems and effects of integrating components with off-the-shelf middleware. Di Nitto and Rosenblum [26] describe how the usage of middleware and predefined components can influence the architecture of a system being developed and, conversely, how specific architectural choices can constrain the selection of middleware [27].

5. RELATED WORK

We have explored fundamental variables to the solution proposed on this document. These variables were *(i)* BEMS standards, where we try to determine what are the features involved on these systems, *(ii)* middleware solutions for BAS and their requirements are also carefully analyzed, and finally *(iii)* a survey of commercial tools regarding BEMS is performed, where we try to determine what are their features and how they address BASs heterogeneity issues.

During this research, mostly due to the amount of information involved, we also found convenient to detail BEMS standards, BASs middlewares, and BEMS commercial solutions along with their features.

Being the context our work sensitive around the concept of real-time data, we have also studied tools and applications that could help us address issues related with data collection, storage, processing, and visualization of data providing from various sensing and actuator devices.

5.1 Commercial Tools

In commercial buildings the set of energy-efficient control techniques that can be implemented is vast, in order to avoid wastes and reduce energy consumption [28]. Based on the on-line documentation some of the most used energy control systems were analyzed, in order to better understand how scalability affects the number of energy-efficient control techniques a software can actually implement.

5.2 Real-time Data Interfacing

When dealing with sensing and actuator devices continuously we face the problem of real-time data [29]. Due to the various interpretations one may have about what is in fact real-time data, it is important to give the definition of what we mean by real-time data.

Real-time data, comprises the flow of information typically segregated and transmitted into small units of data representing some sort of state, instantaneous reading or aggregated computation of some kind, all associated with a given time-stamp and providing from multiple specific sources, either, by means of a polling or publish/subscription strategy. The set of sources being queried, not necessarily in a synchronous way, generates a flow of near real-time data whose state is a function of the variable time.

During the analysis of our solution, interfacing with real-time data became a problem worth studying. According to our definition above, the continuous data flow generated at a given time, must be stored to avoid data loss. During this phase of persisting data, an adequate way to process it and extract actionable information must be possible.

6. FRAMEWORK FOR BEMS SOLUTION

An energy management system can be seen as a composition of different services, each aiming at exploring or exposing data under a certain way. These services are combined to create a platform, upon which new applications can be built on [30]. In this thesis, we present a platform to enable the deployment of energy efficiency techniques, by means of software modules, using sensing and actuator devices.

This platform is able to interface with BAS devices and other sensing or actuator devices. By controlling devices through software, it is possible to program logic, regarding energy efficiency techniques, directly from software modules and without requiring reprogramming micro-controllers with new firmware.

After our extensive analysis on BEMS Standards, Middlewares for BASs (Section 4), BEMS Commercial Tools (Section 5.1) and Real-time data interfacing techniques (Section 5.2), we believe to have come up with an elegant and simple solution.

We will now describe in detail each component involved in our solution. Illustrative pictures will partially describe each section to give more context and by the end of this section a diagram of the complete system is presented.

6.1 Data Integration Layer

The data integration layer provides integration capabilities between the platform and the devices, such as sensing, actuator, or BAS devices. Its goal is to provide seamless integration between hardware devices, ranging from BAS devices to simple HTTP IP-based sensors.

6.1.1 Device Drivers

Device drivers are responsible for the communication with field level systems such as BASs. A device driver is a software module specific for a BASs protocol.

6.1.2 Device Gateway Service

This service provides a way to inter-communicate with BASs by means of gateway drivers. Being a stateless service, it provides a simple interface through an Application Programming Interface (API) containing two endpoints, one for reading and other for writing to and from a device.

6.1.3 Telegraf Data Input Plugins

The service responsible for data collection, uses *Telegraf* to fetch data within regular intervals. To be able to fetch data, our data collection service requires a plugin that has knowledge about how to query a data source. In our solution, we have written a plugin, named *HTTP Meter Plugin*, to interface with IP-based meters using the HTTP protocol. *Telegraf* already comes with an HTTP plugin that is able to fetch data and parse JSON, but it does not support Basic Auth protocol or forwarding of requests via an authenticated proxy. These two features are essential for interfacing with IP-based meters on the University of Lisbon - Taguspark facilities. Internally, our plugin supports *BasicAuth* authentication protocol and *RequestForwarding* using the Linux command *wget*, a non-interactive network downloader.

6.2 Data Access Layer

The Data Access Layer consists of three components (i) Data Querying, this component ensures real-time data is being continuously aggregated before being stored into a persistent storage, (ii) Data Collection, this component is a service that knows how to fetch data from various sources of information and then dispatches it to a persistence service, and (iii) Data History, a set of persistence services that store real-time data into a metrics database and system-wide data on a NoSQL database.

6.2.1 Data History

The Data History component consists of two persistence services (i) a Metrics Database, to store real-time data, and (ii) a NoSQL Database, to store more or less structured sets of information regarding devices, configurations and notification subscriptions.

To store real-time data we use a metrics database called InfluxDB. This database is ideal for dealing with data originated from sensors. It is able to organize readings into measurement tables and is prepared to deal with flexible data structures, enabling device readings data to mutate without compromising the database schema.

6.2.2 Data Querying

Continuously collecting data from various devices across entire building infrastructures can be seen as the process of taking a snapshot of the current state of our facilities at a given time T.

This flow of snapshot data is captured multiple times per hour and can result in enormous quantities of data [31]. Most of the time the exact measure of a given meter at a given time T is not the most relevant information to work with.

Therefore, a way to continuously evaluate and aggregate data as it arrives is a requirement, in order to enable fast queries over real-time data. If every graphical visualization, report or API that needs to access our data has to aggregate it in memory every single time it is needed, that solution can turn out to be hard to scale in the long run, as more data continuously arrives at our database.

Data querying component achieves continuous data aggregation using InfluxDB Continuous Queries. Continuous queries enable down-sampling of raw data to convert high frequency data into low frequency data.

6.3 Core Services Layer

The Core Services Layer consists of two main components (i) the *Energy Manager Service*, a front-facing back-end service, that exposes APIs for channel management, channel categories, channel notifications, and real-time data feed, and (ii) the *Data Processing Service*, a stream processing service that processes streams of data, raising alerts when certain conditions are met, and reporting alerts to the *Energy Manager Service*. Figure 3 presents an overview of the relations between these components.

Data Processing Service, is a stream processing service

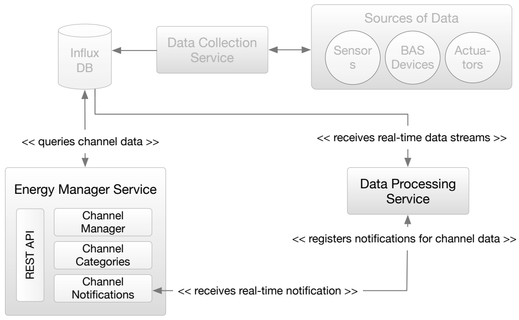

Figure 3: Core Services Layer Overview, presenting two components, Energy Manager Service and a Data Processing Service.

that processes streams of data. It raises alerts when certain conditions are met and reports those alerts to the *Energy Manager Service*.

6.3.1 Energy Manager Service

Energy Manager Service is super set of micro-services that aggregates the micro-services: *Real-time Channel Data Feed*, *Channel Manager*, *Channel Categories* and *Channel Notifications*.

The main idea around this micro-service architecture is that new functionalities can be developed and deployed in the platform by exposing a new micro-service with its own API. Each of these micro-services exposes a version of their APIs through HTTP Headers, e.g, *Content-Type: application/vnd.micro-service.endpoint.v1+json*.

Through this API Versioning Scheme applications can decide whether to support certain functionalities of the platform or to gracefully ignore them.

Channel Manager, exposes a Create, Retrieve, Update and Delete (CRUD) API over the entity Channel. In the context of our platform, a *Channel* is an abstract concept, that defines a communication channel from where we can read and write into. An equipment with various sensors and actuators would have N channels associated with it. This way, to avoid creating strong bindings with the concept of equipment, device or other less generic. We introduced the concept of *Channel Categories* to group one or more channels.

Channel Categories, enable the grouping of one or more channels into a category.

Channel Notifications, exposes an API for channel notifications. Notifications are triggered every time an alert is reported from the *Data Processing Service*. Notifications Manager exposes an API to receive alert events. Alert events are stored associated with their respective channel. When a new alert is received, a broadcast message is sent to all clients subscribing real-time data feed—via web sockets—letting front-facing User Interfaces (UIs) to update accordingly to a new alert, showing a visual notification or engaging the end-user by other means.

6.3.2 Data Processing Service

Data Processing Service works by subscribing to a stream of data over a measurements table from InfluxDB. Every time a new point is registered on the measurements table a User Defined Function (UDF) is executed with the new point. The User Defined Function (UDF) can keep internal state to aggregate data if needed.

An UDF observes points in a data stream and when a specific criteria is met, an alert is sent to the *Channel Notifications Manager* on the *Energy Manager Service*.

This data processing service uses a technology called *Kapacitor*. *Kapacitor* is a engine that performs data processing over sets of time series data. This engine has a larger set of applications, than just *Alerts*, such as monitoring through event processing, Extract, Transform and Load (ETL) tasks over large sets of data, and other.

For working over a set of data using *Kapacitor* we use Stream Tasks. These tasks work over streams of data. Tasks for processing the stream are define in a DSL called TICKscript. These tasks define a pipeline that control which data should be processed and how.

6.4 Application Layer

Applications at this layer have access to a concise collection of services that together solve the heterogeneity problem faced by BASs [32]. Furthermore, these applications have access to a centralized data history service, containing data that spans devices and equipments of an entire building.

The following applications provide simple functionality through web user interfaces. These applications consume the set of services made available on our platform to provide administration, monitoring and energy efficiency capabilities to facility managers.

Central Administration, provides management features to create channels, channel categories and preview pop-up notifications about alerts.

Energy Dashboard, displays energy consumption data read from smart meters installed on the IST Taguspark facilities. The installed meters measure voltage and current and through data aggregation via continuous queries we are able to plot the kWh consumed.

Auto Dimming Luminaries, is a simple application that turns on a light bulb when the quantity of light in a room is scarce (lux <350), meaning its getting dark inside the room [4]. And turns the light bulb off when a good quantity (lux >500) of light is detected again.

7. VALIDATION

Our solution have been tested in a real world environment, by deploying it in the University of Lisbon, Taguspark campus. Taguspark facilities provide a sensor network of nearly fifty energy sensors, scattered within and outside the building. These sensors monitor building energy consumption. Despite energy sensors, Taguspark facilities are also equipped with BASs featuring KNX technology and OPC[1] servers that enable actuation over many devices and equipments.

[1]OPC is a software interface standard that allows Windows programs to communicate with industrial hardware devices. OPC is implemented in server/client pairs. The OPC server is a software program that converts the hardware communication protocol used by a PLC into the OPC protocol.

To assess that our solution enables (i) the collection of energy data on the facilities, (ii) the development of applications that consume such data in a standard format and are able to easily expose it to end-users, and (iii) that the system architecture is modular enough to integrate with multiple BASs, we performed three case studies on the Taguspark facilities.

During this chapter, we will refer to our platform as named during its development, the *Ensight Platform*, meaning *Energy Insight Platform*. Our approach results in three distinct case studies over our platform. The case studies related with data collection, aggregation and visualization involve gathering data from multiple energy sensors. These two first cases are very important to demonstrate a complete workflow of data collection, persistence, aggregation and visualization.

The simple existence of a centralized store of energy sensing data gives room for a whole eco-system of applications for analytics and data mining. In the first two case studies we have setup a dashboard application for data visualization.

7.1 Case Study: IST TagusPark - KNX Energy Meters

In this case study we collected data from nearly 90 KNX sensors on the IST Campus of Taguspark, including MIT Department (Nucleus 14), Nucleus 16, Amphitheater A4 and A5 and also a Weather Station located outside the facilities.

From the available sensors, four of them are Energy Meters as detailed in Figure 4. We have configured our platform to collect their data every 10 seconds for a period of 2 days. During this period we have collected 69,120 data points.

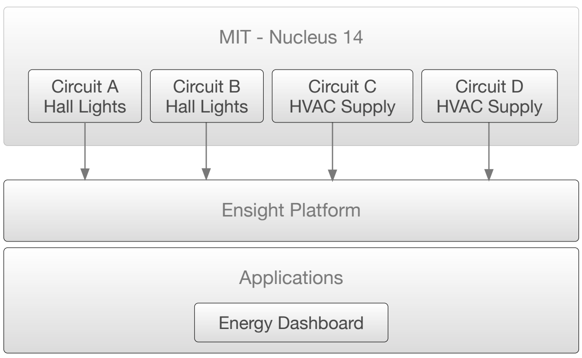

Figure 4: MIT Department - Nucleus 14, featuring four KNX energy meters, being queried by Ensight Platform. Energy Dashboard application sits on top of our platform to render collected data.

To manage this incoming stream of data we configured two data retention policies. Data retention policy "default" holds data forever in our metrics database, while data retention policy "last-day" holds data received on the last 24 hours.

A *Continuous Query* was created to continuously aggregate data from our energy meters (Listing 1). This continuous query is evaluated every 15 seconds to grab data from the past hour, and aggregate the results into intervals of 15 minutes (Figure 1). The aggregated data was stored on the "default" retention policy to make data permanently available.

Listing 1: Continuous Query to aggregate KNX Energy Meters data

```
CREATE CONTINUOUS QUERY
 "knx-energymeters.15m.intervals"
ON "ensight-device-metrics"
RESAMPLE EVERY 15s
BEGIN
  SELECT mean(value) * 230 AS watts
  INTO "default".
      "knx-energymeters.15m.interval"
  FROM "last-day"."taguspark-knx-meters"
  WHERE time > now()-1h AND
      "name" =~ /.*Energy\ Meter.*/
  GROUP BY time(15m),
      id, "name", deviceAddress
END
```

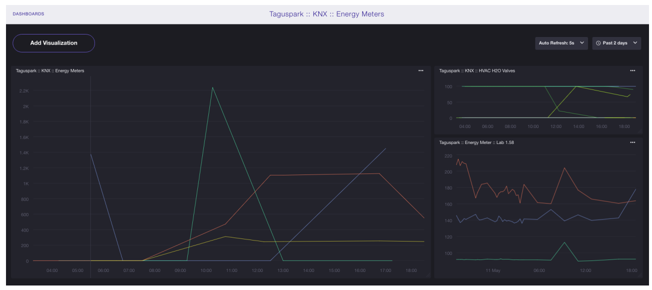

Figure 5: KNX Energy Meters Dashboard displaying a data window of 2 days.

When dealing with KNX sensors, all communication is carried out through a special device called Ethernet/KNX Gateway. During this case study, we discovered that KNX gateways tend to be very limited in the number of simultaneous connections, in our particular case, the models installed at ours campus only allow 5 simultaneous connections. Because of this, our Device Gateway Service had to be fine tunned to cache open connections established between the service and the KNX gateway, so that, it was possible to have some kind of guarantee, in terms of device readings.

Also, during our study data marshaling revealed to be problematic, since readings over KNX devices return data in a binary format. Such binary format is specific to the KNX protocol, and currently the best framework to handle message data marshaling with these devices is a framework called *Calimero*. Calimero only enables us to read simple integers, doubles and text.

It is hard do understand, why a KNX Meter is configured to return a decimal value without any semantic meaning around it. We have discovered that, a typical KNX Meter is configured to be an isolated module that measures current and outputs that measurement.

Changing a KNX Meter to instead send us the pair current and voltage, and let us calculate watts/hour if we want to, revealed to be unfeasible. Although, KNX protocol is Open Source, KNX devices are not free or open to be modified freely. Also, before being installed, KNX sensors need to

Listing 2: Continuous Query that aggregates KNX Weather Station sensors data

```
CREATE CONTINUOUS QUERY
  "knx-sensors.15m.intervals"
ON "ensight-device-metrics"
RESAMPLE EVERY 15s
BEGIN
 SELECT mean(value) AS reading_value
 INTO "default"."knx-sensors.15m.interval"
 FROM "last-day"."knx-meters"
 WHERE time > now() - 1h AND
   "name" !~ /.*Energy\ Meter.*/
 GROUP BY time(15m),
   id, "name", deviceAddress
END
```

be configured by means of a process called *Commissioning*. Such operation is locked-in by a *Commissioning Software* called *ETS*.

Despite the technical challenges of dealing with KNX devices, we also managed to collected data from the Weather Station located outside Taguspark facilities. Currently, through KNX sensors we have access to sensors that measure humidity, temperature, global solar radiation, luminosity, enthalpy, dew point, wind speed and CO_2.

We have configured our system to collect data from all those sensors and aggregate it by means of a continuous query (Listing 2).

We also built a dashboard to aggregate all the data from all the meters (Figure 5). This dashboard give us the ability to, visualize and analyze the data being collected by the KNX sensors in near real-time.

7.2 Case Study: IST TagusPark - HTTP Energy Meters

Taguspark facilities are equipped with sensors relying on different technologies despite KNX. In this case study we focus our attention on the IP-based Energy Meters. These meters are reachable within the Local Area Network (LAN) of the campus and we can perform readings by issuing simple HTTP requests.

Currently we have 7 energy meters installed in our test site (University of Lisbon - Taguspark campus). To request data from one of these Energy Meters we must perform an HTTP request to a reading endpoint, similar to https://<meter-ip-address>/reading. The response is a JavaScript Object Notation (JSON) payload with the measured *current* and *voltage*, per phase. Since these meters are attached to triphasic energy boards, we receive a measurement of current and voltage per each phase.

We have configured our platform to collect data from each of these sensors. Since all the sensors are IP-based, our data collection service can interface with them directly without passing by our Device Gateway Service.

When looking at the collected data, we observe immediately a problem. The collected data shows us the *current* measured in amperes and the *voltage* measured in volts per phase. But such data can hardly be seen as actionable data, since it is hard to perform an analysis over data that, does not map directly to the same units we are used to deal on a daily basis, such as kW/h, present on every energy bill.

To make some sense out of this data, we have configured our platform to compute the energy measurements into watts per phase, and also the aggregated total of all phases per meter. We started by creating a continuous query for near real-time data aggregation (Listing 3).

With this continuous query we can think about the data being collected as the energy consumed by 15 minute periods in watts per each phase. It is also easy to compute the aggregated total per meter by summing the watts of each phase. To better visualize this data, we have built a dashboard to aggregate all the information into a centralized view (Figure 6).

Listing 3: Continuous Query that aggregates Energy Meters data

```
CREATE CONTINUOUS QUERY
  "taguspark-ip-energymeters.15m.interval"
ON "ensight-device-metrics"
RESAMPLE EVERY 15s
BEGIN
  SELECT
    mean(phases_1_current)*
    mean(phases_1_powerfactor)*
    mean(phases_1_voltage)
    AS phases_1_watts,

    mean(phases_2_current)*
    mean(phases_2_powerfactor)*
    mean(phases_2_voltage)
    AS phases_2_watts,

        mean(phases_3_current)*
    mean(phases_3_powerfactor)*
    mean(phases_3_voltage)
    AS phases_3_watts
  INTO "default".
    "ip-energymeters.15m.interval"
  FROM "last-day"."ip-energymeters"
  WHERE time > now()-1h
  GROUP BY time(15m), id, "name"
END
```

7.3 Case Study: Energy Insight - Sensors and Actuators

During the development of this work, a major concern was—possible shared in common with the readers of this document—how can we truly defend such work and criticize the existing BASs and their devices, without deeply understanding how they are built and how they work from the inside out.

Some of the objectives of our platform are to ease the difficulties that arise after turning these devices *ON*, such as where and how does data get stored, how do we aggregate data, how do we abstract the devices and BASs to the end user, how can we act and throw notifications in reaction to a given device reading.

Figure 6: Energy Meters dashboard, with consumptions per meter and the aggregated total.

In this case study we implemented a simple energy efficiency technique over our platform called daylight harvesting [33]. This technique consists in adjusting the amount of electric light need to light a space, according to the amount of natural light measured in the outside.

Therefore, for implementing this technique not only we have used our platform, but also, we decided to make it over a custom built micro-controller. We used a very common Microcontroller Unit (MCU)[2] called ESP8266 (Figure 7). We choose this unit because it is very efficient in terms of energy consumption and also very cheap—built with a 80 MHz processor, a 2.4GHz Wi-Fi module and 4MB of EEPROM memory. Its selling price rounds the 3$ USD and can be found in the majority of online stores like Amazon.

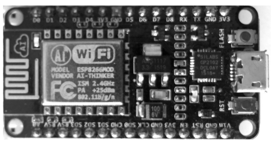

Figure 7: NodeMCU ESP8266 Module attached to an Amica R2 development board.

We acquired also a light sensor to measure luminosity in lux, and a relay module (Figure 8) to control electric current through our MCU module. The relay module enables us to decide whether we should supply or not energy current to our power plug.

During this study we built 3 devices equipped with light sensors. The devices were installed into different points of the room, with the main goal of trying to capture the light variances within different parts of the room. One of the devices, besides incorporating a light sensor, had attached to it an additional relay module to control the ON and OFF functionality of a table lamp.

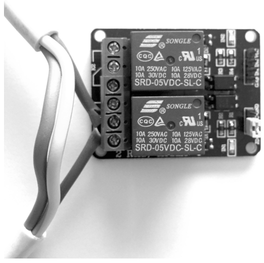

Figure 8: Relay Module actuator connected to a power plug electric cable.

We start by collecting data from the 3 sensors. A continuous query computes the 5% percentile for each sensor, for the last 30 seconds and computes the mean value of the 3 percentiles. The result of this query is then stored on a measurements table that persists its data during one month.

The interpretation of our query is the following: given a 5% percentile with value X, 5% of the values measured are bellow X, therefore, 95% of the remaining values are above X. We compute the average of the 5% percentile of all sensors, and this gives us the average value above of 95% of all luminance readings are.

So according to this assumption, if we grab the last minute 5% percentile mean value of luminance for our room, we can make some assumptions about the light conditions inside the room.

According to a study[3] published by the *National Optical Astronomy Observatory*[4] a luminance value of 500 lux means a normal office is well supplied in terms of light. Bellow that value, electrical light must be turned on in order to compensate the loss of natural light within the room.

The assumption part on our platform is made by a data processing service. In this service we have configured a script to process a data stream over the 5% percentile mean value of the room luminance.

7.4 Evaluation Results

The analysis of our cases studies makes possible to draw a set of conclusions regarding the validity of the *Ensight Platform*. Our platform was able to apportion multiple systems, a KNX-based BAS, IP-based meters and sets of custom built sensors also IP-based. We were able to transform all the gathered data into actionable information and visualize it by means of custom tailored dashboards. The usage of continuous queries to aggregate our data, enabled us to shrink the enormous amounts of data collected from multiple sensors into a more precise set of information, upon which conclusions could be taken. We discovered that data retention policies for sensor data is a good technique to manage large data sets of readings. And that by means of aggregations of sensors raw data we can afford to lose some data, while keeping an overall idea of what was measured. We also learned that a data processing service is of the utmost importance, in order for our platform to be reactive. Reacting to events

[2]A microcontroller is a small computer (System on a Chip - SoC) on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals.

[3]http://www.noao.edu/education/QLTkit/ACTIVITY_Documents/Safety/LightLevels_outdoor+indoor.pdf

[4]http://www.noao.edu/education

lets our platform trigger alerts and expose them to its applications. Due to this functionality, our second case study was able to listen for events of luminance variances, and act based upon what it was programmed to do, i.e, to ensure good light conditions within a room. In our case studies, we were able to attest: historic, monitoring and alarmistic capabilities in our platform. Such capabilities are crucial to support applications for buildings and their management.

8. CONCLUSIONS

From our analysis of the literature regarding the topic of Building Automation, the problem related with heterogeneity of multiple technologies and protocols is well known. Studying the existing proposals, there are a couple of commercial and open source solutions that try to address the problem of BAS heterogeneity. Some try also to solve the problems that a BAS solution itself does not provide typically, such as data history collection, management and alarmistic by means of software modules. Furthermore, we were not able to find any literature documenting or defining the proper architecture for a framework for building energy management.

The assessment conducted in this work concludes that it is not possible to create a monolithic application able to apportion every possible scenario for energy management in buildings. A platform built on the concept of micro services, where each component is configurable, revealed to be—according to our cases studies—a more flexible approach. Through multiple iterations over our solution we have discovered that if all the complexity were left to applications, it could be in fact possible to develop a platform capable of providing actionable data for applications to work with. On the other hand, having a model where every feature of our platform is introduced as an application, granular applications for controlling daylight harvesting, HVAC smart control or simple monitoring will lead to less complex and modular units of software. Therefore, more manageable, maintainable and scalable.

Nevertheless, three case studies were performed to attest the practical applicability of our platform. The first two case studies, tried to test specifically if the platform would be able to support multiple heterogeneous systems—using different protocols. On the last case study the approach was different, and our objective was to attest that if we discard a BAS and just insert simple sensors into the platform, we were able by means of software to achieve the same functionality that a typical BAS would provide. But this time without the complexity that a BAS brings with it. Software can evolve easily, be versioned and updated without requiring the replacement of a physical device, or human interaction.

Choosing a modular architecture for structuring our platform proved to be the most correct way to address the flexibility requirements proposed by this work, easing the addition of new services and the modification of the current ones.

9. FUTURE WORK

There is an endless number of directions worth pursuing regarding future work. By enabling developers to program against a common object model, applications for energy efficiency developed in one building are no longer attached to the BAS configuration of that same building. Furthermore enhancements to the framework itself will enable existing installations to upgrade their systems without having to modify running applications code, which is a huge breakthrough.

Alerting Message Broker, in our current implementation, alert messages are received from a data processing service and published to the applications. Those applications are then free to behave as they wish according to their programming. We think a more sophisticated method could be developed using a solution such as RedisDB or Apache Kafka—both message broker systems designed with high availability and throughput in mind.

Unified Device Model, applying energy-efficiency techniques by using simple and cheap device sensors is one a major advantage. However, it would be nice to have a convention to define the standard set of APIs every device— communicating with our platform—should have.

Case Study for Data Retention Policies by Appliances, the objective of such case study would be to understand for how long, typically, data by appliance should be persisted on a metrics database. It is also important to understand, by appliance, how data should be aggregated as it ages.

Graphical User Interface (GUI), an Energy Management GUI is composed by a set of different components, such as Dashboard Building and Visualization UI, Channel Configuration (the abstraction of devices), Alarmistic Message Viewer, Notifications UI, Building Schematics with live data [34]. An in-depth research to define all these components, their requirements and minimal-set of functionalities is an important study to be addressed.

References

[1] S. Szucsich, "Web Services in Building Automation with focus on BACnet / WS," Automation Systems Group, TU Vienna, Tech. Rep. 183/1-151, 2010.

[2] A. Malatras, A. H. Asgari, and T. Baugé, "A service-oriented architecture for building services integration," *J. Facil. Manag.*, vol. 6, no. 2, pp. 132–151, 2008. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true{&}db=bth{&}AN=32596733{&}site=ehost-live{&}scope=cite

[3] S. Katipamula, E. E. Services, P. Armstrong, P. Northwest, M. R. Brambley, R. G. Pratt, and D. P. Chassin, "Building Controls and Facilities Management in the 21 st Century Facilities Management for the 21 st Century," 2000, pp. 27–38.

[4] O. M. Rudolph Verderber, James Jewell, "Buiding Design: Impact of the lighting control system for a daylight strategy," in *EEE-IAS Annu. Meet.*, 1987, pp. 1–8.

[5] R. P. Leslie, "Capturing the daylight dividend in buildings: Why and how?" *Build. Environ.*, vol. 38, no. 2, pp. 381–385, 2003.

[6] P. Littlefair, J. Ortiz, and C. D. Bhaumik, "A simulation of solar shading control on UK office energy use," *Build. Res. Inf.*, vol. 38, no. 6, pp. 638–646, dec 2010. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/09613218.2010.496556

[7] S. Kiliccote and M. A. Piette, "Advanced Control Technologies And Strategies Linking Demand Response And Energy Efficiency," pp. 1–9, 2005.

[8] E. Tetri, W. Pohl, and B. Lichtlabor, "Concepts and Techniques for Energy Efficient Lighting Solutions," 2004.

[9] X. MA, R. Cui, Y. Sun, C. Peng, and Z. Wu, "Supervisory and Energy Management System of large public buildings," *Mechatronics Autom. (ICMA), 2010 Int. Conf.*, pp. 928–933, 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5589969

[10] T. Derek and J. Clements-Croome, "What do we mean by intelligent buildings?" *Autom. Constr.*, vol. 6, no. 5-6, pp. 395–400, sep 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0926580597000186

[11] J. Bai, Y. Hao, and G. Miao, "Integrating Building Automation Systems based on Web Services," *J. Softw.*, vol. 6, no. 11, pp. 2209–2216, nov 2011.

[12] W. Kastner, G. Neugschwandtner, S. Soucek, and Others, "Communication Systems for Building Automation and Control," *Proc. IEEE*, vol. 93, no. 6, pp. 1178–1203, 2005.

[13] C. Bogen, M. Rashid, and E. W. East, "A Framework For Building Information Fusion," in *Proc. CIB W78 Conf.*, Sophia-Antipolis, France, 2011, pp. 26–28.

[14] G. Cmar, "Reinventing Facilities Operations (and Culture) Through Advanced Use of Data," *Energy Eng.*, vol. 103, no. November 2011, pp. 37–41, 2006. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01998590609509461

[15] A. Khan and K. Hornbæk, "Big Data from the Built Environment," in *Proc. 2nd Int. Work. Res. Large*. New York, NY, USA: ACM, 2011, pp. 29–32. [Online]. Available: http://doi.acm.org/10.1145/2025528.2025537

[16] "Energy management systems - Requirements with guidance for use," 2011. [Online]. Available: http://www.iso.org/iso/home/standards/management-standards/iso50001.htm

[17] A. Lewis, D. Riley, and A. Elmualim, "Defining High Performance Buildings for Operations and Maintenance," *Int. J. Facil. Manag.*, vol. 1, no. 2, p. 16, 2010.

[18] P. Antunes, P. Carreira, and M. Mira da Silva, "Towards an energy management maturity model," *Energy Policy*, vol. 73, no. i, pp. 803–814, oct 2014. [Online]. Available: http://dx.doi.org/10.1016/j.enpol.2014.06.011

[19] R. André, "SmartOffice Information Systems and Computer Engineering," Tech. Rep. October, 2014. [Online]. Available: https://fenix.tecnico.ulisboa.pt/downloadFile/844820067123731/MEIC-76590-Rodolfo-Santos-Article.pdf

[20] N. Motegi, M. A. Piette, S. Kinney, and K. Herter, "Web-based Energy Information Systems for Energy Management and Demand Response in Commercil Buildings," in *Inf. Technol. Energy Manag.* Fairmont Press, 2003, pp. 55–74. [Online]. Available: http://eis.lbl.gov/pubs/lbnl-52510.pdf

[21] B. L. Capehart and T. Middelkoop, *Handbook of Web Based Energy Information and Control Systems.* Fairmont Press, 2011. [Online]. Available: https://books.google.com/books?id=D{_}t3AQoT40kC{&}pgis=1

[22] B. L. Capehart, W. C. Turner, and W. J. Kennedy, *Guide to Energy Management (7th Edition).* Fairmont Press, 2012. [Online]. Available: ISBN:978-0-88173-671-7

[23] L. Wang, S. Greenberg, J. Fiegel, A. Rubalcava, S. Earni, X. Pang, R. Yin, S. Woodworth, and J. Hernandez-Maldonado, "Monitoring-based HVAC commissioning of an existing office building for energy efficiency," *Appl. Energy*, vol. 102, no. 0, pp. 1382–1390, 2013.

[24] P. a. Bernstein, "Middleware: a model for distributed system services," *Commun. ACM*, vol. 39, no. 2, pp. 86–98, 1996.

[25] A. Umar, *Object-oriented client/server Internet environments.* Prentice Hall, 1997.

[26] E. Di Nitto and D. Rosenblum, "Exploiting ADLs to Specify Architectural Styles Induced by Middleware Infrastructures," in *Proc. 21st Int. Conf. Softw. Eng.*, 1999, pp. 13–22. [Online]. Available: http://doi.acm.org/10.1145/302405.302406

[27] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Ph.D. dissertation, University of California, Irvine, 2000. [Online]. Available: http://www.ics.uci.edu/{~}fielding/pubs/dissertation/top.htm

[28] S. Selkowitz, "Daylighting and Energy Efficient Buildings: Challenges and Opportunities!" 2011. [Online]. Available: http://www.radiance-online.org/community/workshops/2011-berkeley-ca/presentations/day2/SS{_}KeynoteAddress.pdf

[29] M. Blackstock and R. Lea, "Towards a distributed data flow platform for the web of things," in *Proc. 5th Int. Work. Web Things*, ser. WoT '14. New York, NY, USA: ACM, 2014, pp. 34–39.

[30] C. Busemann, V. Gazis, R. Gold, P. Kikiras, a. Kovacevic, a. Leonardi, J. Mirkovic, M. Walther, and H. Ziekow, "Enabling the usage of sensor networks with service-oriented architectures," in *Proc. 7th Int. Work. Middlew. Tools, Serv. Run-Time Support Sens. Networks - MidSens '12*, ser. MidSens '12. New York, NY, USA: ACM, 2012, pp. 1–6. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2405167.2405168

[31] A. Margara, *Processing Flows of Information : From Data Stream to Complex Event Processing*, 2011, vol. 44, no. June. [Online]. Available: ISBN:9781450304238

[32] K. De Craemer, G. Deconinck, K. D. Craemer, and G. Deconinck, "Analysis of State-of-the-art Smart Metering Communication Standards," in *5th Young Res. Symp.* IEEE, 2010, pp. 1–6. [Online]. Available: https://lirias.kuleuven.be/handle/123456789/265822

[33] B. Roisin, M. Bodart, A. Deneyer, and P. D'Herdt, "Lighting energy savings in offices using different control systems and their real consumption," *Energy Build.*, vol. 40, no. 4, pp. 514–523, 2008.

[34] J. LaMarche, K. Cheney, K. Roth, O. Sachs, and M. Pritoni, "Home Energy Management: Products & Trends," ACEEE Summer Study on Energy Efficiency in Buildings, pp. 165–175, 2012.