

Review

The Internet of Lights: An Open Reference Architecture and Implementation for Intelligent Solid State Lighting Systems

Emi Mathews ^{1,*} , Salih Serdar Guclu ², Qingzhi Liu ², Tanir Ozcelebi ² and Johan J. Lukkien ²

¹ Embedded Systems Innovation by TNO, 5612 AP Eindhoven, The Netherlands

² Department of Mathematics and Computer Science, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands; s.s.guclu@tue.nl (S.S.G.); q.liu.1@tue.nl (Q.L.); t.ozcelebi@tue.nl (T.O.); j.j.lukkien@tue.nl (J.J.L.)

* Correspondence: emimathews@gmail.com; Tel.: +31-685000161

Received: 30 June 2017; Accepted: 8 August 2017; Published: 11 August 2017

Abstract: The Internet of Things (IoT) is opening up new services and is stimulating changes in industries. The lighting industry is also embracing this change by establishing an Internet of Lights (IoL). This article highlights the main benefits and the challenges to face while going towards IoL. To address these challenges and cater to the specific requirements of lighting networks, an IoL reference architecture, Open Architecture for Intelligent Solid State Lighting Systems (OpenAIS), has been proposed. This article provides an overview of the OpenAIS architecture and explains how one can design specific systems based on this architecture. It also zooms into the configurations and design choices made in a pilot system in a real office building showing the validity of the architecture. A comparison of the OpenAIS system with a state-of-the-art commercial solution shows that IoL systems can exceed proprietary systems in several key performance indicators, such as security, interoperability, extensibility and openness.

Keywords: Internet of Things; Internet of Lights; reference architecture; lighting systems; building automation

1. Introduction

The Internet of Things (IoT) stands for an updated vision of the Internet. It extends the Internet Protocol (IP) communication to billions of resource-constrained endpoints ('things') [1], such as intelligent luminaires and sensors, reaching into the physical world. Things are typically connected into resource-constrained access networks, with low power, lossy, low bitrate asymmetric links and limited group communication primitives. As a network IoT connects uniquely identifiable things to 'regular' Internet services and fast networks. Information about things can be collected and their states can be changed from anywhere, anytime and by anything [1]. IoT enables seamless communication, contextual services and data sharing between things and is bringing radical changes in several industries by converging multitudes of vertical markets [2].

The lighting industry is currently going through a transformation to Solid State Lighting (SSL) such as LED-based systems to enable increased control capabilities (e.g., switching and dimming) and reduced operational costs and energy consumption. However, to stimulate the transition, added value propositions are needed, which are often difficult to achieve with the existing closed and proprietary lighting standards. The fragmented standards and their restrictive Application Program Interfaces (API) often lead to incompatibilities between vendors and limit interoperability with other building services. Embracing IoT in lighting systems creates new opportunities and value propositions. IoT is now maturing, and it is economically feasible to connect each luminaire to the Internet. The

transformation from traditional lighting to SSL makes it way easier to convert light points to IP end-nodes. Hence, it is an excellent opportunity to establish the Internet of Lights (IoL), i.e., an advanced lighting system with IoT at its core.

A transition towards IoT has several benefits: It enables using the network infrastructure in the building for controlling and powering the lighting systems rather than using a dedicated lighting network. Having IP connectivity to all light points enables flexibility and interoperability with other systems such as Building Automation Systems (BAS), smart grids and cloud services. It enables the transition from command-oriented lighting control to service-oriented lighting control and, as a result, can bring in a large variety of new services, create new ecosystems, stimulate investments and innovations and benefits from the worldwide developments in protocols and tools. For example, sharing occupancy data collected by presence detectors used for lighting controls with BAS for air conditioning or with cloud for data analytics opens up new possibilities and services. Overall, it can increase the comfort and well-being of the people in a building, lead to more efficient use of the building and even help to achieve certifications such as BREEAM or LEED [3] by increasing the building performance rating and reducing the carbon footprint.

In this article, we present how to design and realize an IoT-based lighting system for indoor office buildings using an IoT-centric intelligent lighting architecture developed by the European Union (EU) project OpenAIS [4]. The paper is organized as follows: The motivation of going towards an open architecture, as well as the goals to achieve while embracing IoT are discussed in Section 2. The proposed OpenAIS IoL architecture is presented in Section 3. A pilot system is being built using the OpenAIS reference architecture. Section 4 provides a deeper insight into its system design. An analysis of the system is presented in Section 5, where we compare the OpenAIS system with a state-of-the-art system and explain how the architecture provisions the Key Performance Indicators (KPIs). Finally, Section 7 concludes the work and summarizes the challenges yet to solve.

2. Motivation

2.1. Existing Lighting Standards

Over the years, numerous standards have been developed for lighting systems and building automation such as DALI [5], BACnet [6], KNX [7], LonWorks [8] and Modbus [9]. Some are them more specialized, e.g., DALI is used for lighting controls, and Modbus is designed primarily for industrial control; whereas others, such as BACnet, KNX and LonWorks, are more general and can be used for the control of the whole building. Building automation could use DALI for lighting and BACnet, KNX or LonWorks for controlling the rest. Such a combination is possible by using gateways that translate the communication protocols, the data formats and the semantics. However, BACnet, KNX and LonWorks do not easily work with each other, and this often leads to incompatibilities/interoperability issues. Moreover, none of these standards dominate the market in any particular way, which leads to highly fragmented markets.

A closer look at the most prevailing standards in the domain of lighting control and building automation is given below:

- BACnet is a communication protocol for building automation and control networks. The BACnet protocol defines a number of data link/physical layers, including ARCNET, point-to-point, master-slave/token-passing, Ethernet, BACnet/IP, LonTalk and ZigBee [10]. It is widely used in today's heating, cooling and ventilation market, but not for lighting controls because of the complexity and the relatively high cost per light point. BACnet is designed for use in closed networks, and to the best of our knowledge, no commercial product has implemented BACnet security, even though it is in the standard [6].
- KNX is also standard for home and building control and more prevalent in Europe. The main physical communication medium is Twisted Pair (TP) wires. Other media, such as Powerline (PL), Radio Frequency (RF), infrared and Ethernet (also known as KNXnet/IP), are also used [7].

Security was always a minor concern, as any breach of security requires local access to the network. However, this leads to many security vulnerabilities in KNX.

- LonWorks was a family of products originally developed by the Echelon Corporation with a proprietary communications protocol called LonTalk, but is now accepted as an open international ISO/IEC 14908 family of standards with the proprietary hooks removed [8]. The LonTalk communication protocol is useful for building automation applications designed on a low bandwidth, for networking devices over media such as twisted pair, powerlines, fiber optics and RF [8]. LonWorks defines the content and structure of the information that is exchanged. The proprietary nature and limited extensibility diminished its market. The move towards an open standard could not help in withstanding the competition of other open standards and made it almost outdated.
- DALI (Digital Addressable Lighting Interface) is a data protocol and transport mechanism for lighting control. A DALI system can be made up of control gear, control devices and bus power supplies [5]. However, there is no security defined for DALI.

Given the benefits of going towards IoT, extending these lighting standards to bring in such benefits is an option. Although they are standardized, when it comes to the details, e.g., application interoperability, membership is needed. This restricted nature of the standards, together with their limited APIs and lack of a security mechanism make them infeasible for IoL. Hence, a new standard for lighting designed to natively support constrained devices and networks and secure wireless communication is needed.

2.2. Goals of the IoL Standard

The main goals of creating an IoL standard are:

- IP-based: Using a general communication protocol such as IP facilitates a flexible service-oriented approach. Many types of services can share the same network at the same time. Heterogeneous devices and protocols are allowed in IP. The end-to-end connectivity property of IP provides basic interoperability where devices can communicate without protocol translation, and data can be shared across many systems. A wide variety of software and tools, e.g., diagnostics and management tools, are now available, and the IP system can benefit from the developments and innovations of a worldwide community. Moreover, IP offers much better security.
- Open and reusable: For the wider acceptance of lighting and building control communities, creating an open standard is better, as closed ones will yield doubts by potential adopters on the availability, cost (e.g., unfair IPR license terms) and freedom of use. Openness stimulates investments and third party development, leading to an ecosystem of components and services, as well as vendors of those. Moreover, instead of building each block from scratch, reusing the existing ones as much as possible is preferred. This will reduce the time and cost of development, as well as the efforts for standardization.
- Extensible: The standard should be designed in a way that it can rapidly absorb new developments and changes in the market and evolve accordingly. Provisions for changing or updating to the latest communication technologies, easily adding new devices and applications to an existing system, updating the software and upgrading the resources, etc., should be supported.
- Interoperable: An interoperability standard allows different systems to work collaboratively. It can be collaborative decision-making, sharing some data and reports between the systems or creating logical groups between systems. Therefore, the new IoL standard should make the lighting systems interoperable with other systems, especially with BAS and mutually benefit from the systems' capabilities.
- Secure: IoL systems can easily benefit from the state-of-the-art IT security techniques and their advancements. The support of a worldwide community to constantly improve them is an added benefit. Ease of use for legitimate users, protection of data and the integrity of the system need to be supported.

2.3. New Challenges in IoL

Although there are several benefits IoL can bring, it will also introduce new challenges.

- **Performance:** Moving away from today's dedicated lighting network to an IT network with cloud-based communication of IoT raises several questions. Ensuring reliability and guaranteed performance of dedicated lighting systems in an Internet-connected luminaires' world is the key issue to solve.
- **Security:** With IoL, the system becomes more vulnerable to attackers and attack vectors. Making the system secure while opening it to the Internet is a central issue in all IoT systems. Careful monitoring of security vulnerabilities and updating to the latest security provisions are needed. Methods to detect security violations, to prevent leaking of sensitive information and to recover from an attack without huge overhead are core concerns in the security design.
- **Privacy:** Data collection and analytics enabled by IoT can be beneficial. However, revealing information about individual users such as occupancy patterns, motion tracks and usage profile can lead to privacy issues. The system must support privacy requirements such as right to delete data or to be forgotten. Measures need to be taken to prevent privacy violation while enabling data sharing.
- **Energy:** The transition to SSL provides huge savings when compared to the conventional fluorescent or incandescent lighting. In a modern lighting system, the control logic, power distribution logic and interface logic consume additional power. The increased standby power consumption of IP devices should not jeopardize the overall energy efficiency brought by the SSL. Furthermore, intelligent control algorithms should be employed to reduce the energy usage

3. OpenAIS IoL Architecture

As a first step towards creating an IoL standard that is open, IP-based, extensible, interoperable and secure, the EU Horizon 2020 project OpenAIS has been set up with key players from the lighting industry and IoT. One of the key outcomes of the OpenAIS project is to develop an IoL architecture with novel solutions for network connectivity and security that can later be standardized. In this section, an overview of the proposed IoL architecture of OpenAIS is described.

The IoL architecture is developed as a reference architecture, i.e., a template for specifying concrete system architectures. The architecture is designed to support a wide range of deployment scenarios and use cases, which includes retrofitting and refurbishment (backward compatibility to legacy), as well as future office buildings [11]. It envisions creating an open ecosystem to enable a wider community to deliver the smartness of light and allow easy adaptability to cater to the diversity of people and demands. It foresees that the lighting systems, as well as the building management systems will converge to an all-IP-based configuration, with Internet of Things concepts at the heart of new lighting system architectures.

The key objectives of the OpenAIS reference architecture are:

- Define an open architecture for lighting systems with standardized open APIs;
- Make the system interoperable with BAS, cloud services and other systems;
- Increase the building value and reduce the carbon foot print by combining IoT, LED technology and smart grids;
- Easy to specify, buy, install, maintain and use IoL systems for all stakeholders in the value chain.

The OpenAIS reference architecture is described from the viewpoint of different stakeholders using five concurrent views, namely logical, physical, deployment, networking and security views.

3.1. Logical View

The logical view presents the functional decomposition of the system into various functions as experienced by stakeholders that interact with the system. There are two main types, an application

layer containing functions that implement domain-specific functionality of the lighting system and an infrastructural layer containing functions that take care of the system infrastructure.

The application layer functions are:

- Sense: Detects events or changes in the environment such as presence, light-level and user inputs;
- Actuate: Generates light;
- Control: Implements lighting behavior/algorithm;
- DataCollect: Collect, process and store data;
- Group: Helps in administering entities that belong together in an application logic such as actuators, sensors and control;
- Scene: Helps in creating specific effects or scenarios such as a presentation scene in a meeting room using a set of actuator settings;
- Gateway: Interfacing with legacy or other non-OpenAIS systems.

The infrastructural layer functions are:

- Discovery: Helps in detecting the available application layer functions in the system;
- Communication: Supports an infrastructure for communication between the various functions;
- Update: Enables software updates;
- Security: Supports authorization and authentication and protects the confidentiality and integrity of the system against attacks;
- Configuration: Supports updating the static parameters in the system
- DeviceContainer: A container for the properties of a physical device and implements the functions and parameters that relate to a single device such as its IP-address, MAC address, reset, power states and health status.

OpenAIS adopts Sense-Control-Actuate (SCA) models where each sensor updates the actual value/state to the controllers, which then sets the actuators based on this information from sensors and from other controllers. The information flow from function *A* to *B* is shown with a directed edge from *A* to *B* in Figure 1. The controllers support stacking of Control functions in several layers and overriding features which will be further explained in Section 3.3. The DataCollect function collects data from sensors, actuators, controllers or other data-collectors who expose data for storage and analysis. The Group function helps in retrieving group details or administering group members such as a set of controllers, data-collectors, sensors and actuators.

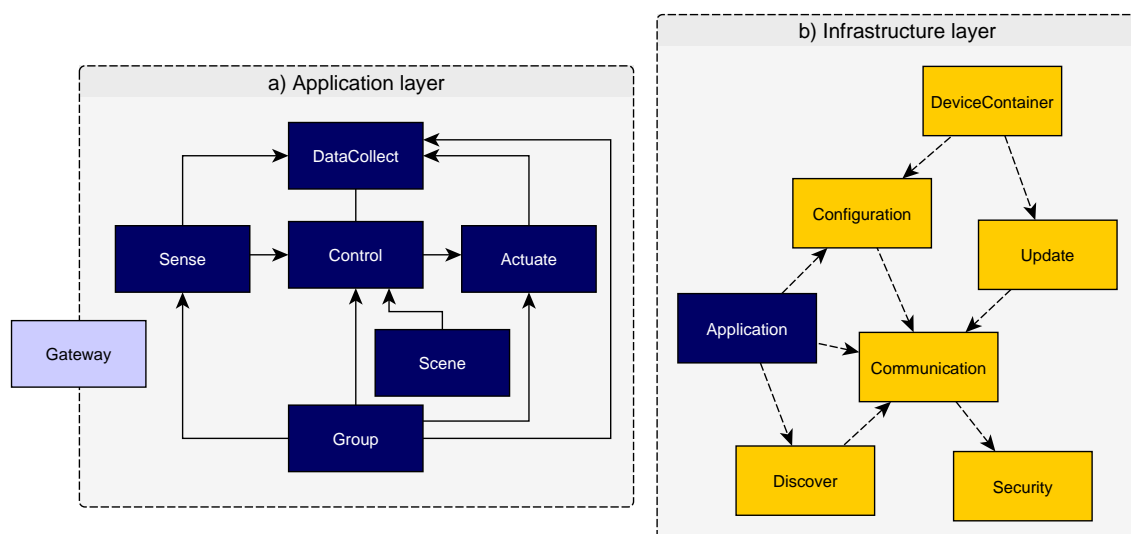


Figure 1. Functional decomposition of the OpenAIS lighting system.

The relations between the infrastructure layer functions shows that the Application function (could be any one of the application layer functions like Sense, Actuate or Control) is configured by the Configuration function, while it makes use of the Communication function for application-level communications and the Discovery function for detecting available functions. The dependency relationship, function *A* uses *B*, is shown with a directed dashed edge from *A* to *B* in Figure 1. The DeviceContainer function uses the Configuration function for parameter settings and the Update function for software update. Discovery, Update and Configuration functions make use of the Communication function for communication, which in turn uses the Security function to make the communication secure.

These functions expose their functionality through certain interfaces. A generic categorization of interfaces defined in OpenAIS is:

- IControl: Interface through which a caller can execute a certain method in a function, like setting a light level or a color.
- IData: Interface through which a function communicates data or changes in its data to the outside world. Data producers send the data out to all interested entities, and the receivers determine how to handle them.
- IConfig: Interface through which static parameters can be set, e.g., addresses, commissioning information, algorithmic parameters, scene values, regulation curves, thresholds.
- IDiscover: Interface through which the element can be discovered on the network.
- IDebug: Interface to configure debugging functionality and trigger testing/debugging operations.

3.1.1. Object Data Model

The OpenAIS Object Data Model (ODM) illustrates the resources of the lighting system in a structured fashion. Rather than introducing new protocols to access the resources, the OpenAIS ODM relies on RESTful standard protocols such as HTTP and CoAP. The representation format of the resources are also intended to be any of the widely-accepted industrial standards like XML, JSON or CBOR.

The OpenAIS ODM defines OpenAIS Objects, which are collections of Resources. Objects must be instantiated to make use of the Resources defined for an Object and can be instantiated multiple times. Within the context of the OpenAIS ODM, resources of the RESTful protocols can be Objects, Object instances and Resources. Therefore, the OpenAIS ODM is agnostic to different techniques of accessing the Resources. This is because different infrastructures may enforce restrictions on resource paths, such as limited paths or restricted names. For the sake of interoperability, vendors are expected to clearly explain the method of accessing their Resources.

Figure 2a shows the hierarchy between the Device, the Object, the Object instance and the entailed Resources. OpenAIS Objects could be of the type Physical or Logical. A Physical Object represents one hardware instance (e.g., a light-point, a sensor) and controls the associated physical effects. A Logical Object represents one controlled aspect of a hardware instance (e.g., intensity, color, sensor). A hardware instance can have only one Physical Object instance, but may have multiple logical instances as shown in Figure 2b. The Device Object shown in Figure 2b features the DeviceContainer functionality of OpenAIS and represents the whole device. Therefore, there is a one-to-one relationship between the device and the Device Object. In addition to the illustrated Objects in Figure 2b, OpenAIS also defines Organizational Objects (e.g., Group, Security) and Functional Objects (e.g., BasicControl, Scene, DataCollect).

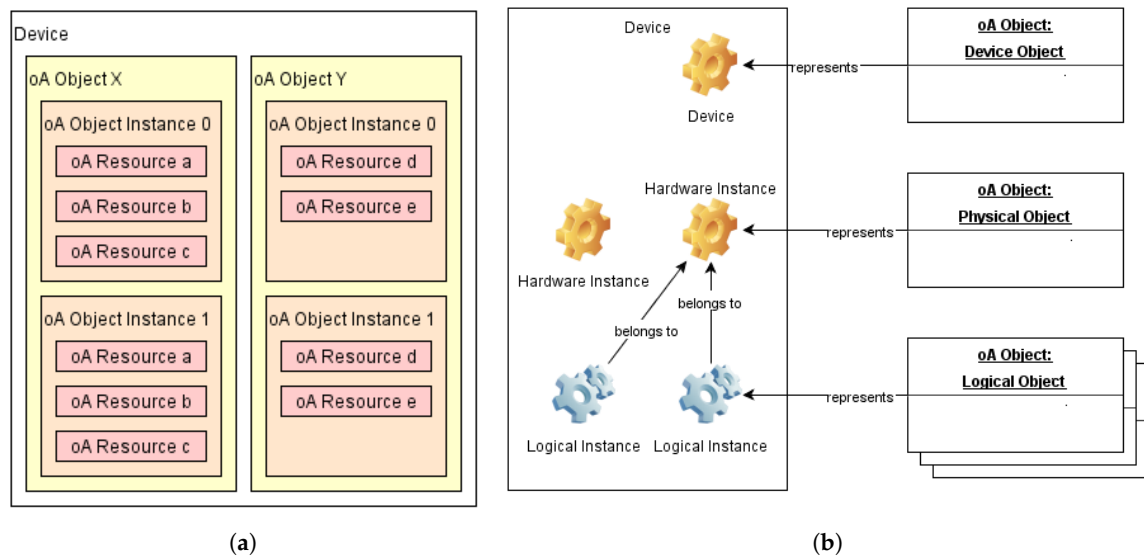


Figure 2. OpenAIS Object Data Model. OpenAIS is abbreviated as oA. (a) Object Data Model hierarchy; (b) An example of a device with various Object types.

3.1.2. Groups

We have seen that the Group function is one of the application layer functions. In lighting, many tasks are often related to a set of sensors, controllers, actuators or data-collectors, and hence, Group is an important concept. In this section, we see how a Group can be realized in OpenAIS.

An OpenAIS group is formed by sharing a group vector between the grouped entities. The group vector provides the information needed to execute the grouping to all members of a group. It provides an Application Group ID, which is a unique identifier of the group. It also provides a Security Group ID that points to the Security Object that controls the encryption/decryption of the group communication. Additionally, the group vector provides a Multicast Group ID, which is the IPv6 multicast address used for the group communication. Ideally the members of Application, Multicast and Security Groups are all the same. However, implementation restrictions may lead to sharing Multicast and Security Group IDs (Object instances) with more than one Application Group.

3.2. Physical View

The physical view describes the physical components or devices in lighting systems, which include luminaires, sensors, area (floor, building) controllers, IT-infrastructure components, cloud computing and management and security systems.

As the reference architecture can be used for several systems with varying technology and design choices for components, the physical view gives only a representation of example system realizations that can be made out of the reference architecture. OpenAIS supports an open structure for lighting systems with respect to scale, topology and hardware.

Figure 3 shows an example physical view of an OpenAIS system with luminaires and sensors (standalone and user interface switches) that are connected together to a local field network using wired and wireless networks. Within local networks, all devices use the same network technology, and they need not be fully separated geographically. The network access layer may contain standard IT components such as switches, access points, OpenAIS devices such as Gateways to translate legacy/non-OpenAIS devices to OpenAIS devices and border routers to connect wireless network, e.g., 6LoWPAN [12], to the wired backbone.

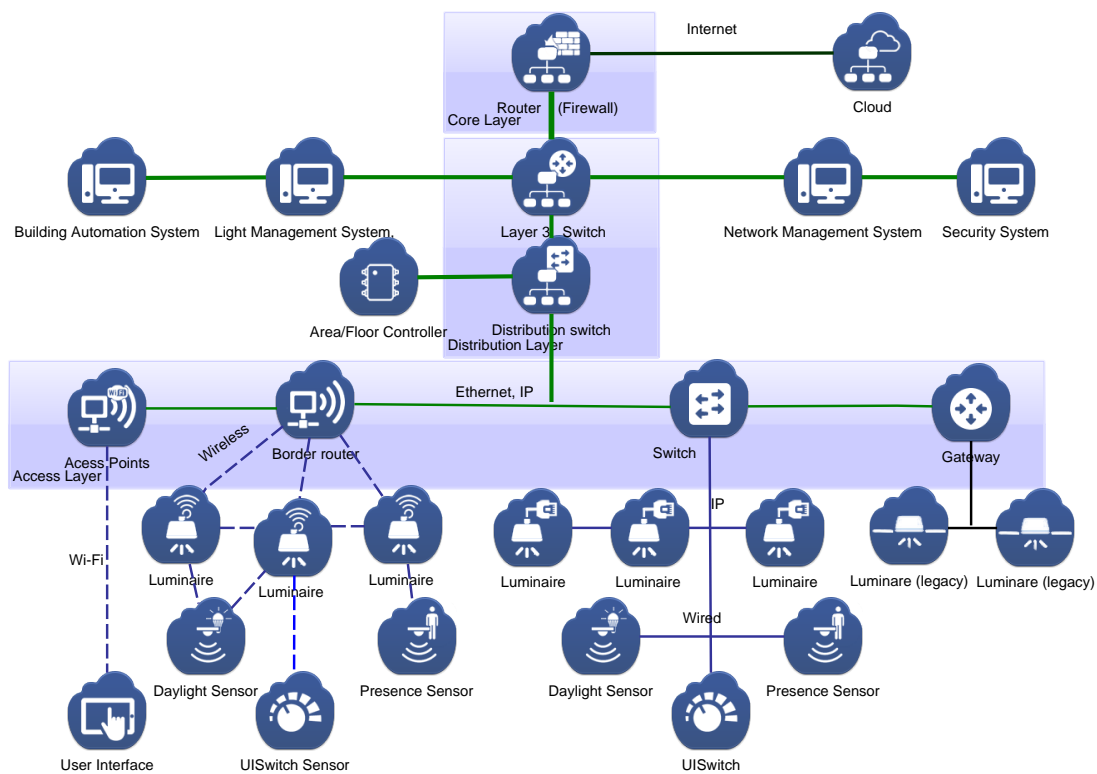


Figure 3. A system realization example.

The backbone network contains core routers and switches providing high-speed access for the various management systems such as the network, lighting system, building automation and security system. The OpenAIS cloud also uses the backbone network protected by a firewall to get connected to the field devices.

OpenAIS does not impose any restrictions on the supported field devices. A direct integration of other field devices, e.g., shading devices, is also feasible. However, the corresponding data models need to be developed. The legacy/non-OpenAIS devices need to be integrated through the Gateway.

An OpenAIS system can extend network size and coverage to any type of network topology. System designers can decide on the appropriate network size and topology based on their requirements and budget. The choice of wireless, wired or a mix of both depends on deployment scenarios. For example, in the case of retrofitting or refurbishing, wireless devices may be preferred to avoid rewiring. One of the main restrictions in the networks, especially wireless networks, is to limit the network diameter to a few hops. Increasing the hop count degrades the performance. System designers can decide on the appropriate network diameter based on the trade-off between performance and cost. For better performance, they can also choose a wired network, but lose the benefits of the wireless networks, such as flexibility and ease of installation.

3.3. Deployment View

The deployment view shows the mapping of the abstract logical functions upon real software and hardware components. Mapping of logical functions to physical devices is easy in some cases, e.g., a Sense function to a standalone sensor or an Actuate function to a Light Point; but it gets complex with other application functions. For example, the Control function can be deployed to a dedicated device like an area controller or it could be deployed in a luminaire. A luminaire with integrated sensors has thus Sense, Control and Actuate functionality in the same physical device.

The versatility of the control deployment allows OpenAIS systems to be designed as centralized, fully-distributed or intermediary control modeled systems. In centralized models, the Control functions

are allocated to one central controller that handles all decisions, whereas in fully-distributed models, they are allocated in all luminaires. In larger configurations, dedicated lighting controllers per group, area or floor are possible. Such stand-alone controllers are optional elements and can be even deployed on other IT-devices like local servers or even on the cloud. This flexibility in Control function deployment is supported by the provision for stacked control; a feature that allows for different levels of Control functions in the system with overriding capabilities, i.e., simple ones can be superseded by more versatile ones. It also allows extending the control behavior, i.e., a new Control function with a higher functionality can be added to extend existing functionality without replacing the existing one. This helps the architecture to be future-proof. Although the laws governing stacking can be made complex, the majority of the Control functionality needs for common lighting systems can be achieved in practice with a small number of Control layers with a straight forward relationship.

3.4. Network View

The typical IoT architectural models try to connect clients to a server in the cloud, which would limit their usage for real-time applications. OpenAIS solves this problem by allowing device-to-device(s) communication and thereby extending IoT to real-time applications like lighting systems.

OpenAIS IoL supports both wired and wireless networks and mandates IPv6 communication for all end nodes. IPv6 is the main decoupling point in the architecture, as the underlying physical, data-link layer stack choices are not mandated; instead, default choices have been proposed in the reference implementation, which are Thread [13] for wireless and Ethernet for wired networks. The envisaged network stack is depicted in Figure 4.

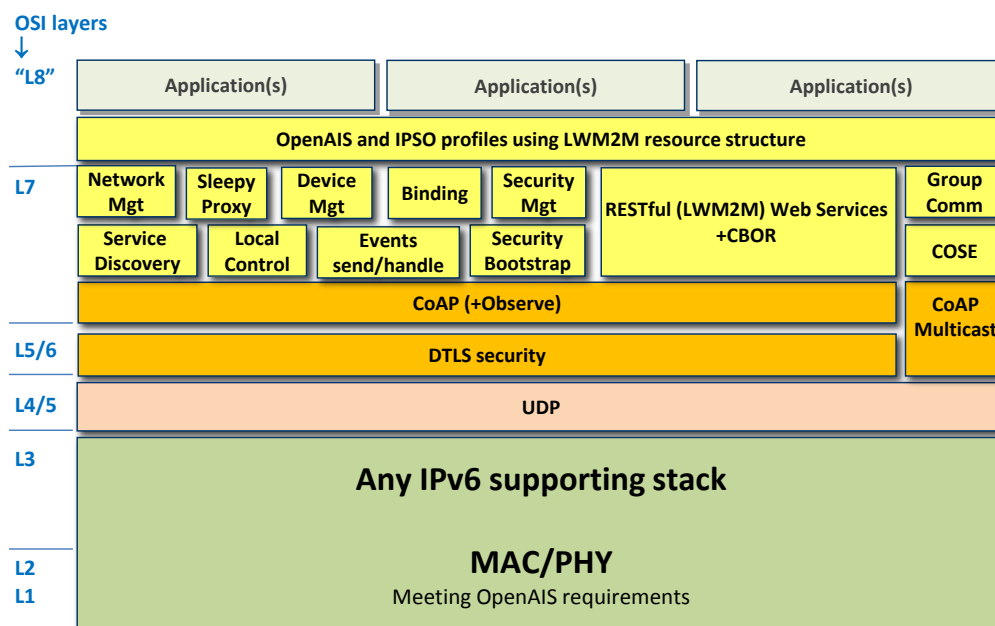


Figure 4. OpenAIS network stack.

UDP is used for transport in conjunction with CoAP (including CoAP observe and CoAP multicast) [14] to support constrained devices. For transport layer security, DTLS is used. RESTful web service interfaces are used for interfacing between the applications. To build the RESTful interfaces, the Open Mobile Alliance (OMA) Lightweight M2M (LWM2M), an efficient and emerging IoT framework [15], is adopted. It supports bootstrapping, (DTLS) security, registration and Object/Resource access.

Although LWM2M enables both device management and application data communication, not all functions required for the lighting and building control market are supported. The LWM2M standard is focused on the device-to-cloud (back-end server) communication pattern and expects that field devices have a reliable connection to a server in the cloud, which cannot be guaranteed in a lighting system deployment. The application data communication in lighting system is mostly device-to-device(s), i.e., the communication is mostly between sensor, controller and actuator as detailed in the SCA model.

To support such communication, OpenAIS Group Communication (OGC) is defined. It enables secured and unsecured group communication using the CoAP multicast [14,16] protocol over IPv6 multicast, serial IPv6 unicast or a combination of these. In special cases, unicast can be used in group communication, to either address a single group member individually or for situations where the reachability of (some) devices is hampered by router settings.

The URI below shows how a CoAP request is made to address an Application Group using OGC:

$$/g/\langle object - ID \rangle/\langle group - ID \rangle/\langle resource - ID \rangle$$

where $\langle object - ID \rangle$ is the Object ID to which the group request is targeted, $\langle group - ID \rangle$ is the system-wide unique Application Group ID and $\langle resource - ID \rangle$ is the resource identifier that determines on which resource this request will act.

To deal with the unreliability of IPv6 UDP multicast communication, at the network level, a (one time) repetition policy of all operational multicast messages is adopted. At the application level, errors such as missing communications or an absence of a device can be detected by a liveness check that analyses the periodic reporting of the current states/commands sent. When an absence of an Object instance is detected, the provision for graceful degradation allows the device to revert to default safe behavior and resume its normal operation once the Object instance returns. For serious errors, a reset must be executed to bring a device back to a known and stable state.

OpenAIS Group Communication operations are secured at the application layer using OSCOAP/COSE [17].

3.5. Security View

As OpenAIS systems are fully networked and connected to the Internet, they are prone to attacks and need to be protected against threats like unauthorized access, control, use of data and modification of the configuration of the lighting system. Security is provided as an internal feature of the system and works independently from site-protecting firewalls. The OpenAIS security architecture supports authorization, authentication, confidentiality and security of the communication, data privacy and integrity of the system against attacks. It reuses the LWM2M specification as much as possible. However, additional changes required for the lighting-specific applications such as support for role-based access control, OGC and the bootstrap process (not depending on Internet connectivity to a central server) are needed. For wireless links, link layer security is mandatory, while for wired links, it is optional.

There are essentially two types of Client/Server interactions occurring in the system, OGC and device-to-cloud communication, where the former is secured using Object Security at the application layer and the latter using DTLS sessions at the transport layer. The OpenAIS authorization policy for client-server interactions, applicable to both OGC and device-to-cloud communication, demands that only authorized clients with an authorization level greater than or equal to the category (security level) of the server resource are allowed to access the resource. For this, the authorized clients are categorized into one of multiple roles (e.g., lighting operational, commissioning, maintenance, etc.), and six access levels (0 to 5) are provided to support role-based access. To implement the authorization policy, OpenAIS uses the Access Control Lists and the Security Object from the LWM2M specification. For the OpenAIS Group Communication, all group members have Level 2 (lighting operational) access to the resources that are defined as accessible for group communication by the data model.

For unicast CoAP requests and responses with required access level greater than 2 (e.g., commissioning), the communication must be secured using DTLS. If the Security Object's security mode is passwords, the J-PAKE cipher suite must be used, otherwise the cipher suites and security procedures specified in the LWM2M specification must be used. Unicast CoAP requests and responses for resources at access Level 2 may use either the Object Security (if OpenAIS Group Communication is used) or DTLS. For all multicast communication, COSE-based Object Security as defined by [17] must be used.

The Object Security format used in both multicast and unicast communication within a group (OGC) for Access Level 1 and 2 resources is currently being standardized within the IETF ACE working group [18]. It refers to OSCOAP as the method to protect CoAP messages using COSE-secured Objects.

4. System Solution

To validate the OpenAIS IoL architecture, the OpenAIS project is progressing with a pilot system installation in a real office at a premier building. The lifecycle of the OpenAIS system starting from the component and system design to use and maintenance is shown in Figure 5. It illustrates the five phases of the lifecycle and related requirements. In this section, a deeper insight into certain aspects of the system solution within the scope of the article is presented.

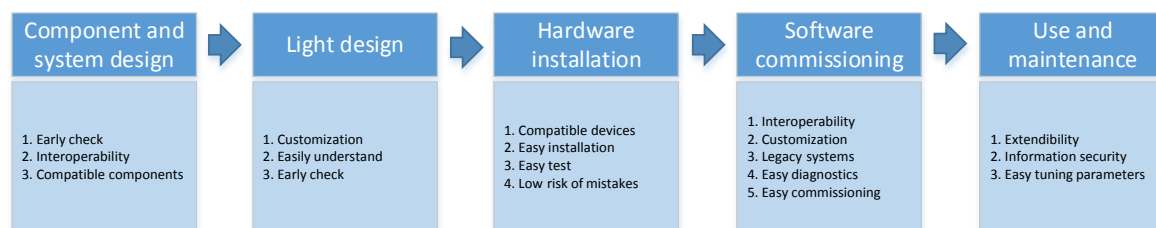


Figure 5. Lifecycle of an OpenAIS lighting system and related requirements in each phase.

4.1. System Requirements

The components of an integrated system have the following requirements:

- The electronic components as part of the OpenAIS system for SSL lighting systems need to be interchangeable and allow replacement without changing the complete system.
- Interoperability of the components is guaranteed on the physical level, the level of network connections and IP-end points.
- The network stack must support the IEEE 802.15.4, Ethernet, Thread/6LoWPAN, IPv6, UDP, DTLS, CoAP (including multicast) and LWM2M protocols.
- The OpenAIS lighting shall comply with lighting safety standards in each country.

Some key functions and performance metrics that need to be achieved in the integrated implementation are given in Table 1.

Table 1. Key parameters and performance metrics.

Operation	Maximum Acceptable Delay
Manual switch	300 ms
Synchronization	100 ms
Non-sync-responding check	200 ms
Startup Time	5 min

4.1.1. Virtual Prototype

To validate system configuration and behavior in the early design phases and avoid errors and issues in the actual system development, we have created a model-based lighting system virtual

prototype. A lighting system is specified using a Domain-Specific Language (DSL). DSLs are formal languages with formal syntax that generate executable models (simulators) for a particular application domain and thereby allowing static and dynamic validation of system instances.

We have created the following DSLs:

- Building DSL, describing a building and its physical components;
- Control template DSL, describing reusable Control functionality (e.g., cell office control behavior);
- Control DSL, deploying the Control template DSL's templates onto the Building DSL's physical components;
- Network DSL, describing the network topology;
- Environment DSL, describing environmental triggers (e.g., occupancy and daylight patterns);
- Visualization DSL, providing 2D visual models (e.g., buildings, luminaires and light outputs).

These DSLs are coupled and executed in a co-simulation framework, which addresses the timeliness of execution, synchronicity, data exchange and coherency of simulators.

Typical issues that affect the robustness of the system like power failure, start-up delays and memory errors are currently being simulated by using a fault model. Additionally, connectivity and bandwidth issues causing packet loss, delays, out-of-sync and variability in the delivery rate are also studied. Optimization strategies on adapting network topologies, retransmitting messages, fine-tuning of protocol parameters and adaptation of the control algorithms are being carried out to make sure that these inconsistencies do not affect much the performance and that the system is still responsive and behaves correctly.

4.1.2. Deployment

To validate the IoL architecture, a pilot installation in a real office setting with a paying customer had been envisioned. One of the premium buildings in Eindhoven, The Netherlands, The White Lady, a former Philips factory and now a national industrial monument, is the pilot site for validation [4]. The system design and specification has been completed, and the installation will take place in the last quarter of 2017. Approximately 400 luminaires, with a mix of manufacturers, wired PoE and wireless controls will be installed in one of the floors of the building for a 3–6-month trial period. In addition to evaluating energy savings, enhanced lighting comfort and personal controls using apps, the complete 'stakeholder journey' from specifying, installing, using and maintaining the system will be evaluated. The total cost of ownership and the return on investment will be assessed and qualified accordingly. The pilot system design follows the regulations and relevant EU standards such as EN12464-1 [19] for lighting requirements for indoor work places and EN-15193 [20] for energy measurements.

The system configuration deployed in one of the wings of the pilot is given in Table 2.

Table 2. System configuration of the pilot.

Luminaire	Area	Number	Communication	Power
Slimblend suspended	Open office, corridor, meeting rooms.	165	Wired UPoE	Wired UPoE
PowerBalance recessed	Toilets, printer rooms, utility rooms.	10	Wireless	Mains power
Slimblend downlight recessed	Cockpit areas.	19	Wireless	Mains power
Taskflex desk light	Open office.	47	Wireless	Mains power

Here, we list the basic deployment of the functionality in the system.

- For each restricted area like conference rooms, the coffee room and phone booths, one or more control groups are created.

- Each group has a Control function deployed on one of its luminaires, which is configured to listen to specific sensors.
- In the open areas, logical areas are formed (e.g., a set of four desks), and the corresponding Control function is deployed on one of its luminaires. The corridor's controller is also configured and deployed in this way.
- There is one floor controller that monitors all controllers in adjacent rooms and directly controls the Control functions in the corridor. Control functions like the Automatic Demand Response (ADR) are also included in the floor controller.
- Data are gathered by DataCollect functions on the floor controllers, which have interfaces to the central controller in the building.

The power supply for luminaires has two forms. The first form is a parallel power supply. Each luminaire has its own power cable. A group of power cables is connected to a PoE switch. The second form is a series supply. The power cable of each luminaire is connected to a neighboring luminaire. Only the power cable of the first luminaire is connected to the PoE switch. In the implementation, both combinations are used depending on how the luminaires are grouped for power supply and how the existing power cables are deployed.

The requirements on Lux level change much in various conditions, which further affects the deployment of luminaires. Table 3 demonstrates the Lux requirements in various conditions. The layout of the luminaires deployed depends on these requirements, and a suitable lighting design for the pilot has been created by professional lighting designers. Similarly, a network layout has also been designed. To meet the performance requirements given in Table 1, in the wireless networks deployed, we limit the hop-count to a maximum of two between the wireless devices and the border routers.

Table 3. Lux requirements in various conditions.

Lux Level	Area or Activity
20–30	Car parks
<100	Corridors, rest areas
150	Stairs and escalators
200	Lounges and dining rooms
300	Background lighting (e.g., IT office, classroom)
500	General lighting (e.g., office, meeting room, kitchens)
1000	Precision lighting (e.g., quality control)

To achieve a stable background light level for users, OpenAIS automatically balances the light level to surrounding light conditions, including daylight and other indoor lighting. Light sensors are deployed to provide input for the dimming values for luminaires to compensate for daylight. Control Objects in luminaires use information from all of the commissioned sensors (on/off, dimmer, local presence, daylight) to switch the light on when users are present and tune to the right dimming level. OpenAIS IoL supports building such application logics.

We will zoom into the following aspects that are relevant to the scope of the article.

4.2. Commissioning Plan

OpenAIS commissioning is adopting a pre-programmed workflow where devices are pre-programmed for their targeted operations prior to their installation. Site documentation on grouping and binding, parametrization and location identification information and system credentials are made available before commissioning. During the commissioning phase, a localization step is carried out where the relationship between the actual location of the device and the device ID is established. Afterwards, the devices are configured, functional verification is performed and a handover to the off-site commissioner (who refines the commissioning based on the customer request) is made.

To simplify the commissioning process and to reduce the commissioning time, OpenAIS is building a smart commissioning tool that can store the pre-programming workflow and assist the commissioner to easily localize, do grouping, binding and parametrization, set the system credentials and rectify on-site errors. The commissioning tool is used to setup and maintain various lighting configurations of devices. It links to the LWM2M server to aggregate device information and records the localization data during the commissioning operation. This information is stored in an external database for further processing. The commissioning tool mainly targets the themes shown in Table 4. Based on these commissioning themes, the OpenAIS system is tested for whether it can provide the required services.

Table 4. The testing themes of the commissioning tool.

Theme	Testing Requirements
Device and object management	Manage and display the devices in the lighting system; configure and reconfigure the Object instances in devices
Configuration management	Re-configure the lighting system, by modifying configuration settings, for changing or repairing
	Re-deploy a commissioned system to operate as soon as possible
	Import a pre-configured configuration; verify whether the current installation is complete and complies with it
Auto discovery	Display the list of discovered devices in the network
Assignment of controls	Create any group of Objects and set up the expected behavior into devices
Location information	Set up new services by using of the location information of devices provided by the lighting system
Security deployment	Guarantee a secure lighting system; prevent misuse of normal users and the intrusion of malicious users

4.3. Out-Of-The-Box Operation

To verify the OpenAIS system after connection and installation, the devices are programmed with some specific behaviors called the out-of-the-box operations. These operations can be used for a first-step testing. This illustrates the basic operation of a system without commissioning. All actuators and sensors operate independently during out-of-the-box testing. For simplicity, the out-of-the-box operation excludes the Internet connection and security control.

There are mainly three testing categories in out-of-the-box operations, including physical devices, operational objects and networking. The out-of-the-box operations of physical devices demonstrate whether the devices are able to achieve correct states as in Table 5. The operational objects are used for testing a simple control relation between devices as shown in Table 6. Networking operations are to set up the basic connection operation of devices.

Table 5. The out-of-the-box operations of physical devices.

Devices	Testing Operation	Result Indicating Success
Luminaires	Power up	100% light output
	Establish network connection	50% light output
Switch sensors	Switch connected lights ON and OFF	Change the ON/OFF state of lights
Presence sensors	Change presence state between no presence and presence	<ol style="list-style-type: none"> 1. The last sensor switches lights off after 5 s if its state is set to no presence 2. The first sensor switches lights on if its state is set to presence

Table 5. Cont.

Devices	Testing Operation	Result Indicating Success
Light sensors	Change light values of sensors	Dim lights by 50% of actual value, if the sensor value suddenly changes
Communication	Discover devices and operational status	<ol style="list-style-type: none"> 1. Return network address, device ID and product ID 2. Return actual settings 3. Accept switch and dim commands

Table 6. The out-of-the-box operations of operational objects.

Objects	Testing Operation
Sensor Objects	<ol style="list-style-type: none"> 1. Sensors send events to the group of Control Objects 2. Sensors multicast events and status to Control Objects using preconfigured Group Objects
Actuator Objects	The Control Object in the operational devices handles them
Control Objects	The Control Object listens to sensor events and controls local actuator directly
Group Objects	The Group Object is pre-shared. Group-ID = 0xFFFF; Security ID = nil; Multicast ID = FF05::222

The basic settings for out-of-the-box networking operation are:

- The devices can setup connections using IPv6.
- Devices can setup a network without additional configuration once switched-on.
- The network allows dynamic connectivity, which means that devices can be switched-on and can join the network at any moment.
- The IPv6 multicast addresses are pre-programmed in the devices.

4.4. User Interaction with OpenAIS-Based Systems

The scope of intelligence and expectations from a smart building are subject to rapid change. Social demands from lighting may push unexpected requirements yet to be known. In order to cope with possible changes of expectations, OpenAIS chooses to be agnostic for the scope of intelligence. Instead of drawing boundaries of intelligence, OpenAIS leaves the largest possible room for different approaches towards the definition of intelligence. Hence, OpenAIS provides fundamental sensing and lighting control functionalities in the most performance effective way that is achievable by 2020's technology.

Applications of OpenAIS are expected to exhibit smartness by utilizing the functionalities of OpenAIS. Developing applications for an existing lighting system used to require deep knowledge of the system (software architecture and network protocols) and the deployed instance of the system (network topology and interface of each device). However, OpenAIS drastically decreases the need for such knowledge through two innovations: IPv6-based lighting and the generic lighting API.

IPv6-based lighting and the lighting API enable the application development ability for almost any software developer rather than a small subset of specialists who possess up to date knowledge about a complex system. The OpenAIS ODM provides interfaces for the minimal set of operations sufficient for any lighting system deployment. Therefore, the ODM is the key building block of the API. However, the ODM itself is unable to address the expected functionality of an easily usable API.

The most tackling challenge of the ODM-based API is creating an infrastructure for allowing the user application to access functionalities of the ODM. Even though an API may allow accessing interfaces of all of the Objects, OpenAIS API enables access only for the interface of Control Objects. This is because Control Objects already utilize all functionalities of other Objects, providing a natural abstraction mechanism. Since access is restricted to Control Objects, the API becomes agnostic to the vendor differentiations of other Objects and their interaction with the Control Objects, i.e., future ODMs for different devices like luminaires may still utilize the existing API. This powerful flexibility

is key to maintaining the API with minimal effort over the course of years. Moreover, the data model of the controller is expected to be similar to the Control Objects of different vendors. Hence, user applications will have backward compatibility and portability. However, the user application is not a part of the lighting system, despite utilizing the system's functionalities. Therefore, user applications require a way to access the lighting system, which is provided by the LWM2M server. The LWM2M server can also translate HTTP requests to CoAP, allowing a simple web browser of a smartphone to access the system. The OpenAIS API is built using the HTTP protocol from the user application to the LWM2M server, which are then translated to OpenAIS CoAP messages that are sent to the Control Object.

Like other lighting systems, OpenAIS is designed to be agnostic to building plans. Because a lighting system has no knowledge of the location of Objects when it is installed, locations of rooms, hallways, sensors, switches, luminaires, and other entities, such location information is stored in the commissioning database during the deployment. Subsequently, the commissioning database enables the lighting system to locate specific Objects belonging to devices. CoAP messages of OpenAIS require the logical address of the Object in order to utilize functionalities of the corresponding Object. However, user applications usually want to act on the location of the Object, rather than its logical address. The OpenAIS API allows user applications to access the commissioning database in order to query the logical address of an Object based on its location. User applications can create CoAP messages for the intended Objects based on their location and utilize the functionality of the data model. Any user application that can access the LWM2M server can control all of the functionalities of OpenAIS. Therefore, access to the LWM2M server is protected with verified usernames and passwords. Users are asked to provide their credentials in order to access the LWM2M server.

4.5. Time Synchronization

An OpenAIS network is an IoT mesh network that needs time synchronization for several purposes. First of all, intelligent lighting requires an ordering of sensing and control events in time. Secondly, network-wide asynchronous coordination strategies for distributed applications are required. Finally, logging and debugging are only possible when there is time synchronization between the entities that log event-time pairs.

The Network Time Protocol (NTP), which is widely in use on the Internet, cannot be directly employed here due to its resource requirements from the network and the devices. In the OpenAIS project, we developed the Mesh Time Protocol (MTP) for time synchronization of nodes in a mesh network to one resource-rich node on the same mesh, such as a Gateway. The Gateway itself is synchronized with the Coordinated Universal Time (UTC) using NTP. The OpenAIS MTP is used to disseminate the UTC time in the mesh using radio broadcasts.

MTP is a modification of the 6LNTP protocol [21]. Nodes synchronize their time using broadcast messages, through their neighbours hop-by-hop. In order to synchronize its time, a node sends a unicast request (REQ) message to the Gateway. This triggers the Gateway to send a synchronization (SYNC) message to its one-hop neighbours, and the Gateway captures timestamp t_0 of transmitting this message. After transmitting the SYNC message, the Gateway sends a correction (CORR) message that contains t_0 to its one-hop neighbours. Having received both messages, a node in the one-hop neighbourhood of the Gateway can calculate the sender's time offset with respect to itself and uses this information to correct its local clock (to sync with the Gateway). The nodes that have already synced with the Gateway then send SYNC messages to the nodes in their one-hop neighbourhood, and the time propagates to the entire network in this way.

5. System Analysis

To analyze the capabilities of the OpenAIS prototype and evaluate its performance, we identified a set of critical aspects of the system from the user requirements. These aspects are then processed into well-defined technical criteria. To make the criteria measurable and comparable, a team of 10 experts

from the OpenAIS architectural team jointly defined the criteria and then broke them down into three well-defined sub-criteria per criterion, which are in turn split into five independent levels for scoring. Table 7 shows two criteria, namely performance and use of open standards and their respective sub-criteria. In some cases, the sub-criteria might have different priority levels, which are assigned based on the expert opinions. For example, the sub-criteria time-to-light, synchronicity and start-up time of the criterion performance are assigned weights 0.5, 0.35 and 0.15, respectively. The total score of a criterion is then the weighted sum of the score of its sub-criteria. Although, this process relies on expert opinions for scoring, the results are often useful and reproducible.

Table 7. An excerpt from the scoring of OpenAIS and LITECOM.

Criteria	Sub-Criteria	Weight	oA Score	oA Total	LITECOM Score	LITECOM Total
Performance	Time To Light (TTL)	50%	4	3.85	4	4.35
	Synchronicity when switching/ dimming a group of devices	35%	4		5	
	Startup time until the system is in regular state after power loss	15%	3		4	
...
Use of open standards	Use of open standards to realize IP connectivity (inclL1, L2, L3)	35%	5	4.35	2	1.35
	Use of open standard application- level IoT framework (L4–L7)	35%	4		1	
	Use of open data models	30%	4		1	

To avoid different interpretations or confusions during scoring, per sub-criterion, five independent levels for scoring are defined; two examples are given in Table 8. Although, this process relies on expert opinions for scoring, the well-defined scoring levels reduce the differences in scoring among the experts. Additionally, ambiguities were resolved in the team discussions. For the clearly quantifiable sub-criteria, the results from empirical analysis or simulation/mathematical models were used to back the scoring. This makes the results of the otherwise fairly subjective process useful and reproducible.

Table 8. Details of the scoring metric of the sub-criteria.

Score	Use of Open Standards to Realize IP Connectivity	Use of Open Standard IoT Framework	Use of Open Data Models
5	All IP connectivity options are open standards	An open standard framework is used without any oA-specific extensions	Use of the fully-standardized oA ODM; based on open standard existing data model(s); no vendor-specific interfaces
4	All IP connectivity options are open standards with the exception of one option not being fully open standard	An open standard framework is used with some oA-specific extensions	Use of standardized oA ODM; based on open standard existing data model(s); 10% vendor-specific interfaces
3	All IP connectivity options are open standards with the exception of one being proprietary or closed	A standard framework is used; the standard is partially open; may have oA-specific extensions	Use of the standardized oA ODM; 20% of interfaces are vendor-specific, i.e., closed
2	Some of the IP connectivity options are open standards	A closed standard framework is used	Use of the small standardized oA ODM; <50% of interfaces are vendor-specific, i.e., closed
1	None of the IP connectivity options are open standards	No standard framework used at all	No open data models used; or 50% of interfaces are vendor-specific, i.e., closed

Table 8. Cont.

Score	Time To Light (TTL)	Synchronicity when switching/dimming	Startup time until the system is in regular state after power loss
5	TTL <= 150 ms	All luminaires in a group start/stop at the same time; stop on the same value; change of values are uniform	<=1 min
4	TTL <= 200 ms	All luminaires in a group start/stop at the same time; stop on the same value	<=2 min
3	TTL <= 250 ms	All luminaires in a group start/stop at the same time	<=5 min
2	TTL <= 500 ms	All luminaires in a group start at the same time	<=10 min
1	TTL > 500 ms	No synchronicity	>10 min

For comparing the performance of the OpenAIS system against a state-of-the-art system, we chose the most recent heritage system called LITECOM that was introduced in 2014 [22]. LITECOM is a DALI standard-based product with intelligent lighting controls. A DALI-based system is selected for comparison because many buildings that deploy other automation systems like KNX often use DALI subsystems interfaced via gateways for lighting controls. To score LITECOM system, two architects who developed LITECOM and that were also involved in the development of the OpenAIS architecture were approached.

5.1. Comparison of OpenAIS Pilot Implementation and LITECOM

A comparison of OpenAIS pilot and LITECOM based on eight criteria is shown Figure 6. The evaluation is undertaken by the designers of both architectures, according to a list of decision-making criteria, which is generated by intelligence lighting domain experts. The spider diagram shows that the OpenAIS system exceeds LITECOM in all KPIs evaluated except for performance and power efficiency. OpenAIS is really strong in:

- Use of open standards;
- Security;
- Business control points, i.e., vendor differentiation.

As discussed in Section 2.2, one of the key goals of going towards IoL is to use open standards. Hence, the OpenAIS system is using open standards to realize IP connectivity (including L1, L2, L3), a standard application-level IoT framework (L4–L7) and open data/object models. This also allows reusing existing standards and software from the wider IT domain. LITECOM is based on the DALI standard and uses proprietary data models.

We have seen that existing lighting standards are weak when it comes to security. Being an IoT system, security is core to the OpenAIS architecture, and it makes use of the state-of-the-art security mechanisms in IT systems and adds on top of it.

Business control points (vendor differentiation) allow vendors to deploy their own differentiating offer without conflicting with the standard. This allows them to provide additional functionalities above the standard features. OpenAIS systems easily support multiple vendors and allow such functionality, whereas LITECOM systems are designed as single vendor systems.

In other aspects, such as interoperability, extensibility and scalability, OpenAIS is rated better than LITECOM. The power efficiency is slightly less than that of LITECOM, but comparable. The performance of general-purpose IP-based communication is less than the fully-optimized task-specific communication in the heritage systems. However, this lower rating is still in an acceptable range, as stated in Table 1.

A SWOT analysis of the OpenAIS system highlighting strengths, weaknesses, opportunities and threats is given in Table 9.

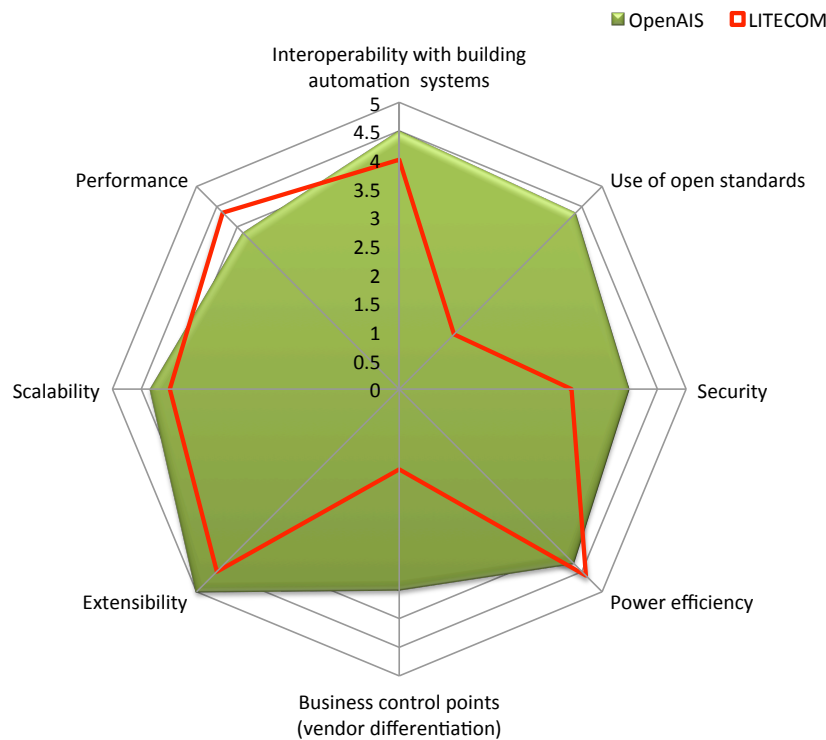


Figure 6. OpenAIS pilot implementation vs. LITECOM scored based on the opinion of a team of experts in both systems.

Table 9. SWOT analysis of the OpenAIS System.

SWOT	Impact	Evaluation
Strengths	Positive, internal	<ol style="list-style-type: none"> 1. Extensibility for functionality, network size and coverage, scalability for offices of any size (Section 5.2.1). 2. Interoperability of BAS and lighting infrastructure (Section 5.2.2). 3. State-of-the-art security for authorization, authentication, confidentiality, privacy and malicious attack (Sections 3.5 and 5.2.3). 4. Business control points for differentiated vendor products (Section 5.2.5). 5. Open standards for communications and data model (Section 5.2.6).
Opportunities	Positive, external	<ol style="list-style-type: none"> 1. Standardization of OpenAIS architecture, data models and security through Fairhair, Internet Protocol for Smart Objects (IPSO) and Internet Engineering Task Force (IETF) (Section 5.3). 2. Extension to various application domains, such as hospitals and shopping malls. 3. Use of artificial intelligence (e.g., face recognition) and user preference elicitation to personalize and automate lighting control. 4. Definition and implementation of Software Defined Lighting (SDL) by virtualization of the communication system for lower operating costs, faster application deployment, more granular security, etc. 5. Development of energy management components for lighting systems and plugging these into city-scale smart-grid. 6. Saving more energy by using energy harvesting, regulating the effects of external light sources (e.g., daylight, moonlight, city lights) to light distribution inside buildings using energy optimized algorithms, for example, for smart windows and shades.

Table 9. Cont.

SWOT	Impact	Evaluation
Weaknesses	Negative, internal	<ol style="list-style-type: none"> 1. Relatively lower performance, in switching synchronicity and start-up time (Section 5.2.4). 2. Deployment must be pre-customized based on floor map. The position selection of routers and controllers depends on the applications and environment. 3. The OpenAIS architecture leaves (cloud) data storage and analysis to vendors, which may be considered as both an advantage and a threat. Policies for data storage and handling are not yet well established, which could lead to privacy issues. Strong policies and enforcement of these policies are needed.
Threats	Negative, external	<ol style="list-style-type: none"> 1. Security of the IoT systems is not fully resolved. There is a need to prove that the extended security solutions for IoT are virtually fail-proof with respect to the relevant threat models. 2. The lighting industry may not support the standardization of the OpenAIS solution sufficiently. 3. Emergence of a new widely-accepted IoT platform may force OpenAIS to change its protocols. 4. Existing lighting system solutions such as LITECOM and EnLight with considerable customer reach or a new solution may evolve into a competing IoT standard that has a large impact (Section 6.1). 5. The cost of hardware deployment may be a concern. The virtualization of hardware components for utilization by multiple applications could be a solution that yields lower cost. 6. The life cycle carbon emission of OpenAIS lighting system compared with existing lighting systems is yet unknown.

5.2. Provisions to Support KPIs

The details of the KPIs and architectural provisions to support them are discussed below.

5.2.1. Extensibility

OpenAIS systems can easily extend their functionality, network size and coverage. We have seen in Section 3.3 that the stacking of Control functions allows extending functionality easily by adding new Control functions (even from the cloud) that can override or extend existing ones. The support for adding identical Object (instances) to one physical device also helps with extending the system behavior without the need to update its software. At the time of configuration/commissioning, the appropriate behavior can be enabled. The provision for adding new or renewed Objects also helps with supporting future communication protocols and additional protocol integration without conflicting with the functionality of the already commissioned system. The modular structure allows trusted (third) parties to add plugins or change single modules of device software and thereby update the software. Hardware drivers can also be updated without jeopardizing the already commissioned working system by relying on the original API.

5.2.2. Interoperability with BAS

Many times, IT infrastructures, BAS and (connected) lighting infrastructures are seen as separate investments where the combination does not have added value for building owners, due to the typical lack of interoperability between these systems. OpenAIS makes interoperability with BAS possible thanks to the following aspects. First of all, it allows transparency of input/output, trends, operations and maintenance data of the lighting subsystem. Furthermore, in general, the amount of effort needed by experts (from both domains) for configuration and maintenance of connections between BAS and lighting systems is a concern. Therefore, the fact that the OpenAIS solution is based purely on open standards instead of proprietary solutions is a big advantage. In the OpenAIS project, an extensive

study of potential gateway architectures for connecting legacy networks to an OpenAIS system was performed. As a proof of concept, a ZigBee gateway that can mediate discovery, switching (on/off) and multicast interactions between the ZigBee network and the OpenAIS IoL network was implemented and tested successfully. Thirdly, building-wide energy optimizations are possible if data from both systems can be taken into account.

5.2.3. Security and Privacy

Security is the utmost requirement for a large networked system to be deployed in an office environment. Modern lighting systems offer many features requiring the use of personal information, most importantly the presence sensor information. Use of such sensitive information mandates not only authenticated access, but also authorization. Privacy can be ensured by setting up policies on the access of data based on proper authentication and authorization. The OpenAIS ODM flexibly supports different user roles with the corresponding authorization levels. Moreover, OpenAIS relies on state-of-the-art security techniques for not hampering the overall performance of the system. Instead of the traditional network security mechanism of the public private key pair, OpenAIS utilizes a novel multicast security protocol by employing symmetric keys for secure transmission of IP multicast packets. Thus, security does not add a significant load for time-to-light, the most important performance criterion.

The security of IoT devices is not a fully-resolved issue. A diverse threat model where the protection of devices, network, data and applications is needed. The security solutions of OpenAIS that extend the state-of-the-art security techniques need to be fail-proof. Furthermore, strong policies and enforcement of these policies for data storage and handling needs to be established.

5.2.4. Performance

As pointed out in Table 1, many lighting operations have to be completed within certain deadlines. The most performance-sensitive operations are time to light, switching synchronicity of a group of luminaires (variance of time to light within the group) and start-up time of the system after a power loss. OpenAIS systems can be designed to get the same time to light performance as LITECOM, but synchronicity or start-up time may not be on a par with it. The cost of using open standards and keeping the architecture secure makes the performance of OpenAIS slightly less when compared to LITECOM. This is because closed systems can usually take advantage of cross-layer network protocol stack design, and they tend to impose tailored solutions rather than highly automated solutions.

Even though the performance of OpenAIS systems heavily depends on the performance of open Internet standards, architectural decisions of OpenAIS led to achieving the boundaries stated in Table 1. As the performance of the standards usually improves with the technological advancements, OpenAIS is expected to stand the test of time for many years to come, in terms of performance.

5.2.5. Business Control Points (Vendor Differentiation)

OpenAIS is not a product by itself. It is a base for lighting vendors to manufacture interoperable products that comply with common principles of performance, security, scalability and extensibility while agreeing upon the same interfaces for semantic communication. OpenAIS leaves great room for each vendor to differentiate their product from the other vendors. For example, if a vendor innovates for more advanced products, OpenAIS actually makes it easier to add the advanced capabilities of devices or services into the connected luminaire network.

5.2.6. Use of Open Standards

OpenAIS strongly utilizes open standards and does not bring forward proprietary solutions. This design decision contributes to elevating other aspects of the architecture such as interoperability, extensibility, security and business control points.

OpenAIS solely relies on existing standards for IP connectivity of the devices. However, IoT is more than the IP connectivity. There are also IoT standards for upper layers of the protocol stack (i.e., L4–L7). For that, OpenAIS utilizes LWM2M, which is published by OMA. Moreover, data models of the OpenAIS also follow open standards in order to keep the possible information change simple.

5.2.7. Scalability

OpenAIS is intended for offices with any size. Typically, it is more difficult to add new devices to large setups since this increases maintenance costs and degrades the performance. However, OpenAIS scales quite well thanks to the flexibly stacking Control Objects. For large setups, such flexible hierarchical stacking of Control Objects ensures scalability without performance degradation or loss of functionality.

5.2.8. Power Efficiency

Power efficiency is the most important reason to switch over to SSL. The power consumption of the whole lighting system also includes consumption of devices such as control, interface and network/infrastructure, other on-premises devices, such as servers and databases, as well as various power losses. The standby power consumption of LED luminaire and sensors in OpenAIS is mostly in line with any modern lighting system. The infrastructural devices' power efficiency is slightly lower than dedicated lighting devices. Having a smart lighting control, for example occupancy and daylight-based control, helps OpenAIS achieve additional energy savings. The interoperability with BAS allows building-wide energy optimization. Intelligent data analytics provides accurate energy consumption data and reporting capabilities that can further optimize energy usage.

5.3. Interoperability Specification and Standardization

OpenAIS regards IoT standards (protocols and frameworks) as carriers and uses them according to their original specification. The OpenAIS Interoperability Specification (OpenAIS-IS) enables third parties to develop components independently and to integrate them into a working system consisting of components that satisfy this specification. It specifies a minimal core (OpenAIS-MC) that is essential for the OpenAIS innovation and provides explicit means to extend a system beyond OpenAIS-MC. Third parties are free to extend functionality beyond OpenAIS-MC provided that it does not interfere with OpenAIS-MC.

The three pillars of the OpenAIS-IS are: (i) a reference specification; (ii) a reference implementation; and (iii) a method for interoperability validation. The reference specification gives a high-level specification of concepts, with limited relationships to particular technologies. It gives what is common in different realizations and defines the OpenAIS reference architecture (main scenarios of use, commissioning, management and update), as well as the OpenAIS protocols and interfaces. Furthermore, it presents a data model that considers devices as containers of Actuator, Sensor and Control Objects. The reference implementation, on the other hand, is a detailed instantiation of the reference specification. The reference specification and the reference implementation together constitute the OpenAIS-IS. Finally, interoperability validation is used to test the implemented rules and processes for interoperation. OpenAIS defines a set of interoperable interfaces of the system, as well as interoperability test cases (scenarios) and their required outcomes.

The vision of the recently-formed Fairhair alliance (partner program of IEEE-ISTO) [23] supports the OpenAIS IoT approach and gives an opportunity to standardize (parts of the) OpenAIS specification for a wider scope. As the OpenAIS ODMs are built on top of LWM2M/IPSO models, standardization through the IPSO alliance is also possible. The OpenAIS security for Group Communication is currently being standardized in Internet Engineering Task Force (IETF) [18].

6. Related Work

6.1. Lighting Systems Products and Projects

There is a wide range of products coming into the lighting market. Daintree Networks based on ZigBee PRO [24], Enlighted Inc. wireless network based on IEEE 802.15.4 [25], Gooee (a full-stack IoT solution), the LITECOM lighting management system from Zumtobel [22], Philips Connected Office Lighting [26], etc., are examples of proprietary IP-based lighting systems. There are products that provide wireless extension to DALI [27].

There were also a number of projects related to building automation systems and lighting. EnLight [28] was an EU project that developed an architecture and a decentralized lighting control by applying the publish-subscribe design pattern, which gives scalability and a network-stack-independent eventing system. GreenerBuildings [29] was an EU Seventh Framework Programme (FP7) project to develop an energy-aware adaptation of public buildings using smart objects and cloud systems for increased robustness and failure resilience. Self-organising, Cooperative, and robUst Building Automation (SCUBA) [30] was an EU FP7 project to address the challenges of the fragmented BAS market by creating a novel systematic engineering approach via an integrated design tool chain and an online integration and control framework.

6.2. IoT Architecture and Framework

There are several competing alliances led by the world's prominent semiconductor, electronic and telecom industries resulting in various IoT platforms and frameworks. The AllSeen Alliance led by Qualcomm with more than 180 members is a popular one [31]. The AllJoyn [31] is an open source framework from AllSeen with a set of system services that enables interoperability among products and applications across manufacturers using a D-Bus message bus. There is also an AllJoyn-based Lighting Service Framework (LSF) to provide an open and common way of communicating among connected lighting products. The Open Connectivity Foundation (OCF), formerly Open Interconnect Consortium (OIC), with more than 300 members, is another prominent one [32] that provides a competing framework called IoTivity [32] hosted by the Linux Foundation. It aims at defining a common communication framework based on industry standard technologies for IoT and provides the certification and branding for reliable interoperability in IoT. OCF enables RESTful manipulation of resources across devices. OIC has acquired a major player, the Universal Plug-n-Play (UPnP) Forum, that pioneered the networking software protocols of today's smart home. UPnP is deployed in billions of home entertainment devices and Internet gateway devices. The acquisition helps to boost their efforts for standardization in IoT. OneM2M [33] is another standard driven by telecom companies based on the design of the European Telecommunications Standards Institute (ETSI) M2M. Machine-to-Machine (M2M) communication is migrating towards IP-based technology, and oneM2M aims at developing technical specifications for a common M2M service layer that can be readily embedded within various hardware and software to connect the wide range of devices worldwide with M2M application servers. The Open Mobile Alliance (OMA) [15] has come with standards for managing lightweight and low capability devices on a variety of networks. The OMA Lightweight M2M (LWM2M) [15] includes device management and service enablement for LWM2M devices and defines the application layer communication protocol between a LWM2M server and a LWM2M client. It specifies a simple RESTful Object Model and API for reading, setting and executing resources on any device. The Internet Protocol for Smart Objects (IPSO) [34] published their Smart Objects, which are built on top of the LWM2M. It defines a number of standard device functions 'Objects' that are useful for lighting systems.

There are many standardization organizations working on the standardization of IoT providing IoT definitions, reference architectures and models. The interesting ones are European Telecommunications Standards Institute (ETSI), International Telecommunication Union (ITU), Institute of Electrical and Electronics Engineers (IEEE), Internet Engineering Task Force (IETF), National

Institute of Standards and Technology (NIST), Organization for the Advancement of Structured Information Standards (OASIS) and World Wide Web Consortium (W3C). ETSI is a member of oneM2M [33] involved in the standardization of M2M interface. The ITU-T Y.2060 [35] provides an IoT reference model with four layers, namely the application layer, service support and application support layer, network layer and device layer. The IoT Architecture working group of IEEE is standardizing an architectural framework for the Internet of Things [36]. The IETF working group standardized IoT communication protocols such as 6LoWPAN and CoAP [12,14]. W3C is working on the Web of Things to reduce IoT fragmentation [37]. The IoT World Forum [38] has published an IoT reference model with seven levels, which are physical devices and controllers (the things), connectivity, edge (Fog) computing, data accumulation, data abstraction, application and collaboration and processes. With numerous players, alliances and standards, there are also initiatives to support the convergence and interoperability of IoT standards, such as the Alliance for Internet of Things Innovation (AIOTI) initiated by the European Commission [39].

There are also various projects that address IoT architecture and frameworks. The EU FP7 project Internet of Things Architecture (IoT-A) has come with an Architectural Reference Model (ARM), for creating open interoperable systems and integrated environments and platforms [40]. The IoT ARM consists of an IoT reference model providing the highest abstraction level for the definition of model and an IoT reference architecture for building compliant IoT architectures. The IoT reference model includes the domain model, information model and functional model together with the communication model and trust, security and privacy Model as the sub-models of the functional model. The EU FP7 IoT@Work project [41] focuses on industrial and automation environments to create self-managing resilient networks employing middleware and service-oriented application architecture.

7. Conclusions and Outlook

In this article, we explained how the lighting industry can benefit from IoT by moving from the traditional closed and proprietary systems to secure, extensible, interoperable and service-oriented systems. We presented an Internet of Light architecture, OpenAIS, designed to address the challenges while making this transition. An overview of the OpenAIS IoL architecture with a deeper look from different architectural perspectives has been provided. Additionally, a system solution explaining the design of a pilot system, with its configurations and design choices, has been provided. An analysis of the system by comparing it with a state-of-the-art commercial solution shows how IoL systems can exceed proprietary systems in several KPIs such as security, interoperability, extensibility and openness. A SWOT analysis of the OpenAIS system highlighting strengths, weaknesses, opportunities and threats is also provided.

The system analysis shows that despite careful design choices, the performance of the IoL is slightly lower than proprietary solutions. The transition towards IoT enables using/sharing the network infrastructure in the building instead of employing a dedicated network for each building services. Ensuring reliability and guaranteed performance of dedicated lighting networks in shared networks will be a challenge. The security and privacy of IoL is an issue not fully resolved. Careful monitoring of security vulnerabilities and updating to the latest security provisions are needed. To ensure privacy, strong policies and their enforcement for data storage and handling are needed. A careful study on the impact of IoL on various stakeholders and the changes it brings in to the lighting value chain and building sector need to be carefully analyzed.

Acknowledgments: This article is enabled by the OpenAIS project, co-funded by the Horizon 2020 Framework Programme of the European Union under Grant Agreement Number 644332. Special thanks to all OpenAIS partners and architects who contributed to the OpenAIS reference architecture, especially Walter Werner, the architecture leader, Ben Pronk, Esko Dijk and Stephanie Schwendinger, whose work has been referenced in the article. We also thank the three peer reviewers for their constructive comments and suggestions that contributed to improving the article.

Author Contributions: J.J.L. conceived of the article and, together with E.M., T.O. and S.S.G., outlined it. E.M. wrote the Introduction, the IoL Architecture, (a part of) the System Analysis, Related Work and Conclusion sections. Q.L. contributed to the system design, SWOT analysis and wrote the corresponding sections. S.S.G. contributed to the analysis of the system, the user-interface to the system and their writing. T.O. contributed to time synchronization and writing a part of the Introduction, System Solution and System Analysis. J.J.L. contributed to the interoperability analysis. All authors contributed to the review of the document.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

6LoWPAN	IPv6 over Low-power lossy Wireless Personal Area Networks
ACE	Authentication and Authorization for Constrained Environments
ADR	Automatic Demand Response
API	Application programming interface
ARCNET	Attached Resource Computer NETwork
BACnet	Building Automation and Control Networks
CBOR	Concise Binary Object Representation
CoAP	Constrained Application Protocol
COSE	CBOR Object Signing and Encryption
DALI	Digital Addressable Lighting Interface
DSL	Domain Specific Language
DTLS	Datagram Transport Layer Security
EU	European Union
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
IoL	Internet of Lights
IP	Internet Protocol
IPR	Intellectual Property Rights
IPSO	Internet Protocol for Smart Objects
IPv6	Internet Protocol Version 6
ISTO	Industry Standards and Technology Organization
J-PAKE	Password Authenticated Key Exchange by Juggling
JSON	JavaScript Object Notation
KPIs	Key Performance Indicators
LED	Light-Emitting Diodes
LWM2M	Lightweight M2M
MTP	Mesh Time Protocol
Mgt	Management
NTP	Network Time Protocol
oA	OpenAIS
ODM	Object Data Model
OpenAIS	Open Architectures for Intelligent Solid State Lighting Systems
OSCOAP	Object Security of CoAP
OSI	Open Systems Interconnection
PoE	Power over Ethernet
REST	Representational state transfer
SDL	Software-Defined Lighting
SSL	Solid-state Lighting
SWOT	Strengths, Weaknesses, Opportunities and Threats
UDP	User Datagram Protocol
UPoE	Universal Power Over Ethernet
UTC	Coordinated Universal Time
XML	Extensible Markup Language

References

1. Minerva, R.; Biru, A.; Rotondi, D. Towards a Definition of the Internet of Things (IoT). *IEEE Internet Things* **2015**. Available online: http://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf (accessed on 9 August 2017)
2. Baumgartner, T.; Wunderlich, F.; Jaunich, A.; Sato, T.; Bundy, G.; Griefsmann, N.; Hanebrink, J. *Lighting the Way: Perspectives on the Global Lighting Market*; Technical Report; McKinsey & Company: New York, NY, USA, 2012.
3. Saunders, T. *A Discussion Document Comparing International Environmental Assessment Methods for Buildings*; BRE: Watford, UK, 2008.
4. OpenAIS—Open Architectures for Intelligent Solid State Lighting Systems. Available online: <http://openais.eu/> (accessed on 30 June 2017).
5. International Electrotechnical Commission. *Digital Addressable Lighting Interface*; International Electrotechnical Commission: Geneva, Switzerland, 2014.
6. BACnet. *A Data Communication Protocol for Building Automation and Control Networks (ANSI Approved)*; ASHRAE: New York, NY, USA, 2012.
7. KNX System Specification—Architecture. Available online: www.knx.org (accessed on 30 June 2017).
8. LONWORKS. ISO/IEC 14908 LONWORKS Open Systems and Related Standards. Available online: <http://www.lonmark.org/> (accessed on 30 June 2017).
9. Modbus Organization Inc. Modbus. Available online: <http://www.modbus.org/> (accessed on 30 June 2017).
10. Newman, M. *BACnet: The Global Standard for Building Automation and Control Networks*; Momentum Press: New York, NY, USA, 2013.
11. Mathews, E. Deliverable: 2.3-Final Architecture of OpenAIS System. Available online: <http://openais.eu/en/results/> (accessed on 30 June 2017).
12. Montenegro, G.; Kushalnagar, N.; Hui, J.; Culler, D. *RFC 4944—Transmission of IPv6 Packets over IEEE*; Internet Engineering Task Force (IETF) Network Working Group: Fremont, CA, USA, 2007; Volume 802.
13. Thread Stack Fundamentals. Available online: https://threadgroup.org/Portals/0/documents/whitepapers/Thread%20Stack%20Fundamentals_v2_public.pdf (accessed on 30 June 2017).
14. Shelby, Z.; Hartke, K.; Bormann, C. *RFC 7252—The Constrained Application Protocol (CoAP)*; Internet Engineering Task Force (IETF): Fremont, CA, USA, 2014.
15. Open Mobile Alliance. *Lightweight Machine to Machine Architecture*; Version 1.0; Open Mobile Alliance: San Diego, CA, USA, 2013.
16. Rahman, A.; Dijk, E. *RFC 7390—Group Communication for CoAP*; Internet Engineering Task Force (IETF): Fremont, CA, USA, 2014.
17. Selander, G.; Mattsson, J.; Palombini, F.; Seitz, L. *Object Security of CoAP (OSCOAP)*; Internet-Draft; Internet Engineering Task Force (IETF): Fremont, CA, USA, 2015.
18. Somaraju, A.; Kumar, S.; Tschofenig, H.; Werner, W. *Security for Low-Latency Group Communication*; Internet-Draft; Internet Engineering Task Force (IETF): Fremont, CA, USA, 2016.
19. European Committee for Standardization (CEN). *12464-1: Light and Lighting—Lighting of Work Places—Part 1: Indoor Work Places*; European Committee for Standardization: Brussels, Belgium, 2002.
20. *EN 15193: Energy Performance of Buildings—Energy Requirements for Lighting*; European Committee for Standardization: Brussels, Belgium, 2007.
21. Hong, S.; Kim, D.; Ha, M.; Bae, S.; Park, S.J.; Jung, W.; Kim, J.E. SNAIL: An IP-based wireless sensor network approach to the internet of things. *IEEE Wirel. Commun.* **2010**, *17*, 34–42.
22. LITECOM Next-Generation Lighting Management. Available online: <http://www.zumtobel.com/com-en/products/litecom.html> (accessed on 30 June 2017).
23. The Fairhair Alliance, Enabling Lighting and Building Automation via the Internet of Things. Available online: <http://www.fairhair-alliance.org/> (accessed on 30 June 2017).
24. Wireless 101: Mesh Networking. Available online: www.daintree.net/products/why-daintree-networks/wireless-101-mesh-network (accessed on 30 June 2017).
25. Enlighted Inc. Wireless Network. Available online: www.enlightedinc.com/resources (accessed on 30 June 2017).
26. Philips Lighting—Connected Office Lighting. Available online: <http://www.lighting.philips.nl/systemen/connected-lighting/connected-office-lighting.html> (accessed on 30 June 2017).

27. Lunatone wDALI. Available online: <http://lunatone.at/en/dali-systems/wdali/> (accessed on 30 June 2017).
28. EnLight—Energy Efficient and Intelligent Lighting Systems. Available online: www.enlight-project.eu (accessed on 30 June 2017).
29. GreenerBuildings. Available online: www.greenerbuildings.eu (accessed on 30 June 2017).
30. SCUBA—Self-Organising, Cooperative, and RobUst Building Automation. Available online: www.aws.cit.ie/scuba (accessed on 30 June 2017).
31. AllSeen Alliance, AllJoyn Framework. Available online: <https://allseenalliance.org/framework> (accessed on 30 June 2017).
32. Open Connectivity Foundation and IoTivity. Available online: <https://openconnectivity.org/developer/reference-implementation/iotivity> (accessed on 30 June 2017).
33. OneM2M Standards for M2M and the Internet of Things. Available online: www.onem2m.org (accessed on 30 June 2017).
34. IPSO Alliance: Defining Smart Objects. Available online: www.ipso-alliance.org/ (accessed on 30 June 2017).
35. ITU-T Y.2060. Available online: <https://www.itu.int/rec/T-REC-Y.2060-201206-I> (accessed on 30 June 2017).
36. EEE PROJECT P2413—Standard for an Architectural Framework for the Internet of Things (IoT). Available online: <https://standards.ieee.org/develop/project/2413.html> (accessed on 30 June 2017).
37. WEB OF THINGS AT W3C. Available online: <https://www.w3.org/WoT/> (accessed on 30 June 2017).
38. IoT World Forum Reference Model. Available online: <https://www.iotwf.com/resources> (accessed on 30 June 2017).
39. Alliance for Internet of Things Innovation (AIOTI). Available online: <https://aioti-space.org/> (accessed on 30 June 2017).
40. Bassi, A.; Bauer, M.; Fiedler, M.; Kramp, T.; van Kranenburg, R.; Lange, S.; Meissner, S. (Eds.) *Enabling Things to Talk. Designing IoT Solutions with the IoT Architectural Reference Model*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 163–211.
41. Final Framework Architecture Specification. Available online: <https://www.iot-at-work.eu/downloads.html> (accessed on 31 January 2016).



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).