LED WATCH

James Brodrick

CAN WE TALK? YES AND NO

Interoperability among connected lighting systems? It depends

Lighting systems are increasingly incorporating network interfaces and sensors, morphing into data-collection platforms that are capable of implementing advanced adaptive lighting strategies. Picture data-driven lighting-energy management in buildings and cities, combined with value-added services like space utilization, office scheduling and inventory management. Interoperability is key to unlocking the full potential of all this new data, but at this early stage, that potential is limited by significant fragmentation of the underlying technologies and interfaces. Today's connected lighting systems (CLS) are generally not natively interoperable; they can't be assumed to be able to communicate with, and more importantly, make use of, data produced by each other or other systems.

In 2017, the U.S. Department of Energy (DOE) initiated the first of a series of studies to characterize and comment on the current state of CLS interoperability. At present, interoperability between connected lighting systems offered by different vendors (or in some cases, even between different solutions provided by the same vendor) is facilitated primarily through application programming interfaces (APIs)—which define protocols for how outside systems make specific requests of, and receive responses from, specific CLS resources.

An API might, for example, define how an outside system would request the energy consumption over the last 24 hours from a specific luminaire, or would request that a set of luminaires reduce their light level from 100 to 50 percent. So the first study set out to examine the diversity of APIs in several connected lighting systems, characterizing the extent of interoperability they provide and illustrating challenges, limitations and trade-offs encountered over the course of the exploration.

API CHARACTERISTICS

Not all APIs were readily available (e.g., described on an unrestricted web page or in a downloadable document). Some required contacting the CLS developer—who, in some instances, provided the API documentation readily, and in others required an explanation of why the API documentation was being requested and how it would be used. A few developers required the signing of a non-disclosure agreement in order to obtain the API.

The APIs encountered used a wide variety of authentication schemes, which determine whether the API request is valid. System resources described in APIs were named and organized in myriad and, in some cases, inconsistent ways. While some APIs had a logically consistent resource-tree structure, others were apparently developed incrementally over multiple version releases as more features were added over time, and incremental extensions were not always implemented consistently or logically.

APIs encountered also used a wide variety of data models, each apparently custom-developed. The models were not always well designed or documented, or even self-consistent. They might exist in very different locations within API resource trees, and different naming conventions or labels, units and data types were used for seemingly similar, or at least related, parameters. In some cases, models from different systems were functionally incompatible.

Some API developers supported cross-origin resource sharing (CORS), which allows requests to be made to or across multiple domains (e.g., different CLS); while others either didn't, or didn't do so readily. Some connected lighting systems didn't provide API access to historic data outside a certain time window, or only provided real-time data. In some cases, the real-time update rate was unspecified, seemingly inconsistent, or asynchronous, and was dependent on

DOE RECOMMENDATIONS

DOE intends to continue to explore viable interoperability approaches and invites collaboration with system integrators, industry consortia and CLS developers. DOE will also continue to identify potentially high-value use cases and invites lighting industry stakeholders to suggest additional use cases to explore, and their relative value. Interested parties should email *doe.ssl.updates@ee.doe.gov*.

Among DOE's recommendations:

- CLS developers should make their APIs readily available and ensure that documentation is synchronized with software updates. They should also facilitate easy, efficient bug reporting.
- API developers should name and organize resources in readable, logical and consistent ways and should consider adopting a well-thought-out, modern and evolving common-resource organization.
- CLS developers should fully document API data models, including reporting unit (when applicable), data type, resolution and accuracy. They should also consider adopting well-thought-out and well-maintained common-data models—ideally created by entities with application expertise specific to the data model.
- API developers should explore approaches to reducing system integrator effort—such as providing support for multiple measurement units and time aggregations for reported data, and providing support for control schemes beyond switching between or pushing out of new device profiles.
- API developers should consider the implementation of publish-subscribe models for reported data.
- API developers should consider the implementation of override or prioritization schemes that support adaptive control of configurable system devices.
- CLS developers should support user exploration of new and previously unproven use cases and facilitate easy, efficient bug reporting.

how frequently the CLS could be queried through the API.

INTEGRATION VIA API

Integrating heterogeneous and asynchronous resources residing in different connected lighting systems into a single interoperable platform via their APIs required some degree of software coding. While the type and amount of coding required can vary by the approach pursued, integration via APIs requires more than configuration of a set of user-selectable options and subsequent validation. In lieu of sufficient API documentation, system integrators may have to ascertain, sometimes through crude trial-and-error experimentation, what an available API resource represents (e.g., a controllable light, a light controller, an energy reporter) and what information and data model it uses. Converters or translators may need to be developed to handle inconsistent data models. Further, an information model that suitably encompasses the myriad data models to be integrated may need to be inferred in order to effectively aggregate data from or broadcast commands to all systems.

Integration via an existing "platform" might require less development time, in particular if the platform was designed for or had been previously used to implement the desired functionality. But the use of such "platforms" typically comes with trade-offs between user-friendliness, capability and flexibility. Further, they must be maintained to address hardware, firmware and software updates, and ideally should support new systems as they enter the market.

Integrating multiple connected lighting systems through APIs doesn't result in a homogeneous system; at best, it yields a common user interface and experience. However, effectively hiding the underlying complexity can require a significant amount of back-end integration work. Managing what functionally remains distributed at one or more system levels can be challenging. Differences between network protocols, device representations and access policies that affect performance must be understood, normalized if possible, and managed—not only initially, but over the course of hardware, firmware and software upgrades.

INTEROPERABILITY USE CASES

The implementation of real-life use cases can expose previously unseen or unanticipated issues. In this first study, two use cases were simulated in the DOE Connected Lighting Test Bed to illustrate the relative effort required to use APIs to enable features and capabilities that are facilitated by interoperability. The results were enlightening and offer very useful insights to CLS tech developers, system integrators, manufacturers and other stakeholders.

In the first use case, researchers tried to aggregate energy-use data from six connected lighting systems. This effort exposed issues beyond the impact of working with different data models. Energy measurement accuracy and resolution were generally unspecified in marketing literature and API documentation, raising concern as to how to statistically treat data from different sources. Further, the lack, in some cases, of API access to historic energy data required the system integrator to query the CLS in real time and log energy or power data, thereby requiring dedicated storage and computing resources.

Some connected lighting systems didn't provide API access to historic data outside a certain time window, or only provided real-time data

The researchers explored the ability to broadcast lighting commands across all integrated systems as a second-use case. They instructed specific luminaires in each system to turn off, turn on and dim. This experiment exposed latency differences—time delay between sending the signal and the system responding—among systems, which might arguably compromise both the functionality and performance needs of the use case associated with the lighting command.

The complete study report is available online at *https://energy.gov/eere/ssl/connected-lighting-interoperability*.

James Brodrick is the lighting program manager for the U.S. Department of Energy, Building Technologies Office.