2021 International Young Engineers Forum (YEF-ECE) | 978-1-6654-1264-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/YEF-ECE52297.2021.9505145

Synoptics of Things (SoT): An Open Framework for the Supervision of IoT Devices

Bruno Serras
ISEL - Instituto Superior de Engenharia de Lisboa
IPL – Instituto Politécnico de Lisboa
POLITEC&ID
Lisbon, Portugal
A43538@alunos.isel.pt

Carlos Gonçalves
ISEL - Instituto Superior de Engenharia de Lisboa
IPL – Instituto Politécnico de Lisboa
POLITEC&ID
Lisbon, Portugal
carlos.goncalves@isel.pt

Tiago Dias
ISEL - Instituto Superior de Engenharia de Lisboa
IPL – Instituto Politécnico de Lisboa
POLITEC&ID
Lisbon, Portugal
tiago.dias@isel.pt

A. Luís Osório
ISEL - Instituto Superior de Engenharia de Lisboa
IPL – Instituto Politécnico de Lisboa
POLITEC&ID
Lisbon, Portugal
lo@isel.ipl.pt

***Abstract*—In the industry, there is a need to manage and monitor systems and their associated devices. This process can be quite complex when the technological landscape is composed of several heterogeneous systems. In such scenarios, both the systems and their devices can be from different suppliers, each one with its own interfaces, often proprietary and non standardized. Moreover, several different administrators can be responsible for the supervision of a system, which greatly complicates the process of supervising multiple systems. Yet, overviewing a technological landscape can be facilitated by the concept of Synoptics, as a fundamental part of a supervisory control and data acquisition system. As modern applications become more and more web based, it makes sense to use web browsers to implement the concept of Synoptics. In this paper, we present the Synoptics of Things (SoT) framework to extend the Synoptics concept to the web domain. The proposed SoT framework explores the concept of Web Components as a standard to enable the development of custom and reusable components for modern web browsers. Such components, or Widgets-IoT, as we call them, can function as an abstraction of both IoT devices and informatics systems, facilitating the interaction with the supervising operators. Accordingly, we present and discuss the use of the SoT framework in an Informatics System of Systems reference implementation to monitor and control the involved systems and the various executive services that compose them.**

***Index Terms*—Internet of Things, SCADA, Synoptic, Web Component, Widget**

I. INTRODUCTION

Typically, the technological landscape of an industrial system is composed of several heterogeneous systems, where each system is composed of numerous devices from different suppliers. The process of controlling and monitoring such devices can be quite challenging because suppliers provide different protocols and Application Programming Interfaces (APIs) to access them, often nonstandard proprietary solutions. To make matters worse, such technological environments are usually regarded as a system of systems, where each system has an administrator implementing its own monitoring strategy of the devices. As a result, aggregating the information made available by all the different devices and systems is a challenging endeavor that is normally accomplished using Supervisory Control and Data Acquisition (SCADA) systems.

SCADAs are custom systems, usually implementing proprietary solutions, that are used to manage and monitor industrial processes [1]. They are comprised of computers and peripheral devices like Programmable Logic Controllers (PLC), as well as Graphical User Interfaces (GUI), such as Synoptics. A Synoptic is a Human Machine Interface (HMI) and serves as the operator window of the supervisory system. It presents an overview of the processes, with diagrams that can show the state of the system and alarm displays.

In this paper, we present the Synoptics of Things (SoT) framework to facilitate the supervision of the devices that comprise industrial systems and promote an open market competitive technology landscape for organizations. The SoT framework provides an open platform for developing and presenting on a web browser widgets representing abstractions of all the system elements, ranging from the cyber-physical systems to the software services of the involved informatics systems. As modern applications become more web-based, the SoT framework adopts the Web Components [2] specification to develop the widgets as custom and reusable web components, which are called `Widgets-IoT`.

The remaining of this paper is organized as follows. Section II briefly reviews SCADA supervision interfaces. Section III presents the Synoptics of Things concept and Section IV discusses the usage of the Synoptics of Things framework in the SITL-IoT project [3]. Finally, Section V draws the major conclusions and discusses some directions for future work.

II. State of SCADA Supervision Interfaces

A SCADA system is used to manage and monitor industrial processes, which that can be located either within premises or geographically dispersed around thousands of square kilometers, such as water pipes, power lines or railways.

SCADA systems share some common elements like PLCs and Remote Terminal Units (RTUs) [4]. PLCs have the capability to control complex processes are used substantially in SCADA systems. RTUs are designed to support SCADA remote stations, as they are equipped with wireless radio interfaces in order to support situations where wire-based communication is not viable.

An essential element of a SCADA system is also its interaction with the supervising operator, known as HMI. Synoptics are part of HMI because they are displayed on a monitor screen and can be interacted by a mouse click or, more recently, by touch. Synoptics display an overview of an industrial process, providing global visibility of the managed process. They should be intuitive so that the operator can quickly act in case of an abnormality. Fig. 1 shows a Synoptic that monitors a water treatment process. The interface to show the Synoptic was built using the multimedia viewer Adobe Flash Player, which was discontinued in December 31, 2020 due to security vulnerabilities [5].

Nowadays, most SCADA interfaces are native applications that are proprietary to a particular system of an organization. The main objective of the Synoptics of Things (SoT) framework is to provide an open platform for an agile and efficient development of Synoptics that can be used in various contexts to manage heterogeneous software and hardware artifacts from multiple suppliers. The adoption of modern web standards in the SoT framework is also an important step towards this goal.

III. The Synoptic of Things Concept

The SoT is a framework that enables the use of web browsers to create and configure Synoptics using `Widgets-IoT`. A `Widget-IoT` is an abstraction of a "Thing", may it be an hardware or a software element of a system, and contains properties related to the widget customization, such as color and size, and properties of the represented "Thing", e.g. user credentials for accessing a video camera or the unity of measurement for a temperature sensor.

A system's operator can interact with a `Widget-IoT` in a Synoptic to receive visual messages based on the state of the monitored things. For example, a widget glowing green means the "Thing" is operating correctly, while glowing yellow means that it needs attention, and glowing red means that there is a critical error. In addition, the operator can interact with the `Widgets-IoT` to control the "Things", similarly to what happens in modern SCADA systems.

Fig. 2 depicts the primary elements of the SoT framework using a SysML Block Definition Diagram. As it can be seen, a `Synoptic of Things` is composed of elements of type `Widget-IoT`. A `Widget-IoT` is a `Web Component` that can also aggregate other Web Components. For example, a video camera `Widget-IoT` has a lens, a body, and a microphone, each of these elements can be an individual `Web Component`. This composition is represented by the cardinality of the attribute `components` of the element `Widget-IoT` being $0..*$.

This aspect of every element being a `Web Component` provides reusability in the sense that these components can be used to develop other `Widget-IoT` (e.g., a temperature sensor can be associated to multiple devices). In addition, a `Widget-IoT` must have a context menu (`ContextMenu`) that is accessible when double-clicking the widget. This menu

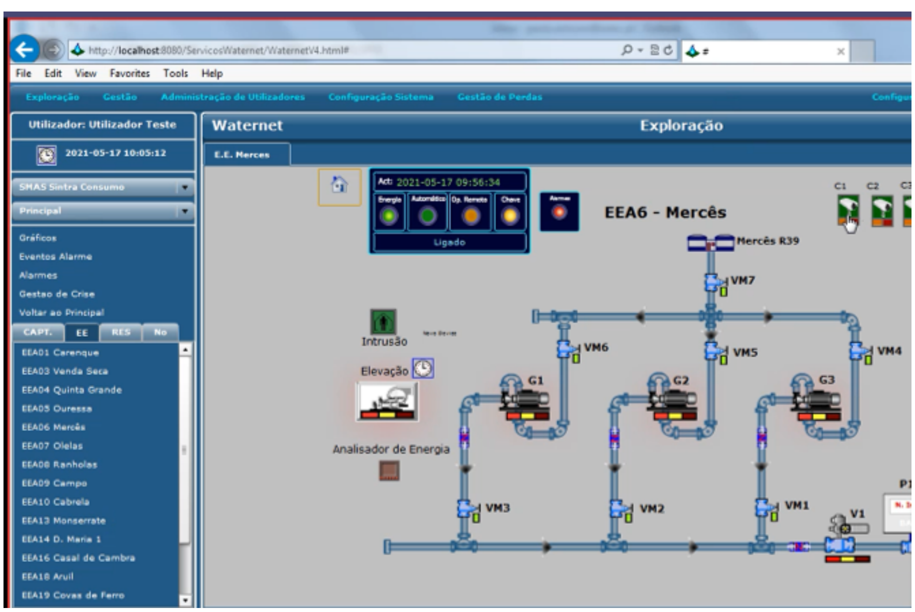

Fig. 1. Synoptic of a water treatment process.

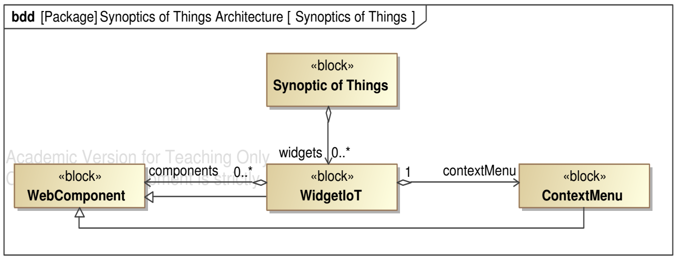

Fig. 2. Simplified SysML block definition diagram of the SoT model.

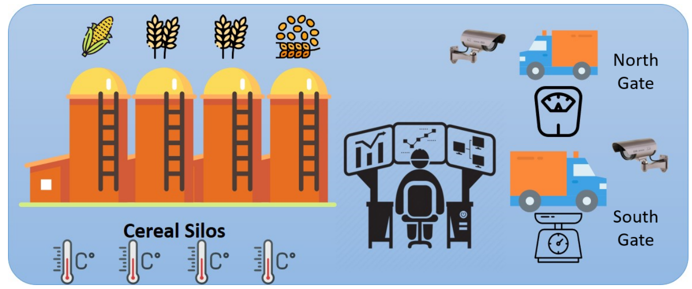

Fig. 3. The SITL-IoT Project case study scenario.

offers a set of actions related to the widget to interact with the associated "Thing", e.g., a video camera `Widget-IoT` could have a set of actions to modify properties of the video such as resolution and bit rate.

To develop a `Widget-IoT` several different implementations of the Web Components specification are available, like the Polymer Project [6] or LitElement [7], both from Google. The graphical representation of a `Widget-IoT`, i.e. the widget itself, can be implemented using the Scalable Vector Graphics (SVG) open standard [8], which works well with other web standards like Cascading Style Sheets (CSS), Document Object Model (DOM) and JavaScript.

IV. Using the SoT Framework in the Context of the SITL-IoT Project

Sistema de Inteligência nos Terminais Logísticos (SITL-IoT) is a research project that addresses the challenges faced by a maritime terminal silos operator when evolving its industrial agri-food environment towards an open technology infrastructure using the Informatics System of Systems (ISoS) framework [9]. Fig. 3 presents a simplified view of such technological landscape depicting only a subset of the elements necessary for loading and unloading cereals to/from trucks.

As it can be seen, access to the industrial facilities is done using two gates for truck control: North for inbound and South for outbound. The trucks are weighted both inbound and outbound using industrial scales from different suppliers with its own specific weigh controller technology and interfaces. Also, the truck movements between the seaport and the agri-food are recorded using a proprietary Closed-Circuit Television (CCTV). The silos include several temperature sensors that are used to monitor the temperature at regularly spaced levels of its structure. This weighing bridge infrastructure, the temperature sensor elements, and other cyber-physical systems are modeled as IoT devices, which are encapsulated as `Services` in the ISoS framework [6]. All the ISoS elements, i.e. the Informatic System (`ISystem`), the Cooperation Enabled Services (`CES`) and the `Services`, can have an associated `Widget-IoT` to monitor their state. SoT panels can be created only for the `ISystems` and `CES` elements, as they are a composition of of one or more `Service` elements.

The proposed SoT framework provides a web interface for creating and configuring `Widgets-IoT`, named Widgets-IoT Configurator, as depicted in Fig. 4. This web interface is divided in two panels - a left one and a right one. The left panel lets the user select a type of `Widget-IoT`, e.g., a video camera, a silo, or a weigh bridge. After selecting the desired type of `Widget-IoT`, the user is prompted to configure the widget, such as typing the desired name, choosing the size of the widget and selecting the background color. For a specific type of `Widget-IoT`, such as a video camera, other information may be required, like credentials for accessing the device.

Created `Widgets-IoT` appear in the right panel sorted by their type. By right-clicking the name of a widget the user has access to a context menu, where it is possible to view the created widget, edit its configurations or simply deleting it. This web interface also has the option to import previously created widgets, as well as exporting them, by using the Extensible Markup Language (XML) standard to represent the widgets information (**Import** and **Export** buttons in the upper right corner). Fig. 4 shows the Widgets-IoT Configurator web interface with some widgets already created that represent the scenario previous described in Fig. 3, with a context menu associated to the "Camera North" widget.

The SoT Configurator web interface can also be used to create and configure Synoptic panels. Using this interface, the user can add previously created `Widgets-IoT` to a Synoptic and drag and drop them in a way that represents a monitoring scenario. This web interface also allows the user to import and export Synoptic panels with XML files. Fig. 6 shows a SoT panel for the scenario depicted in Fig. 3, where in the left side we can see buttons to: i) import the previously created `Widgets-IoT`; ii) import a previously created synoptic panel; and iii) export the current synoptic panel.

Fig. 5 presents the SysML block definition diagram of the `Widget-IoT` of a silos. A `SiloWIoT` is a `Widget-IoT` that represents a silo and has a set of components such as a thermometer (`Thermometer`), a humidity sensor (`HumidiySensor`) and a fan (`Fan`). Each one of these elements is a Web Component as described previously in Section III, and have a particular set of actions associated, e.g., viewing the value reported by the thermometer or changing the speed of the fan. These actions are accessible using the `ContextMenu` of each `Widget-IoT` elements (Fig. 2).

The `SiloWIoT` was used to develop the SoT panel shown

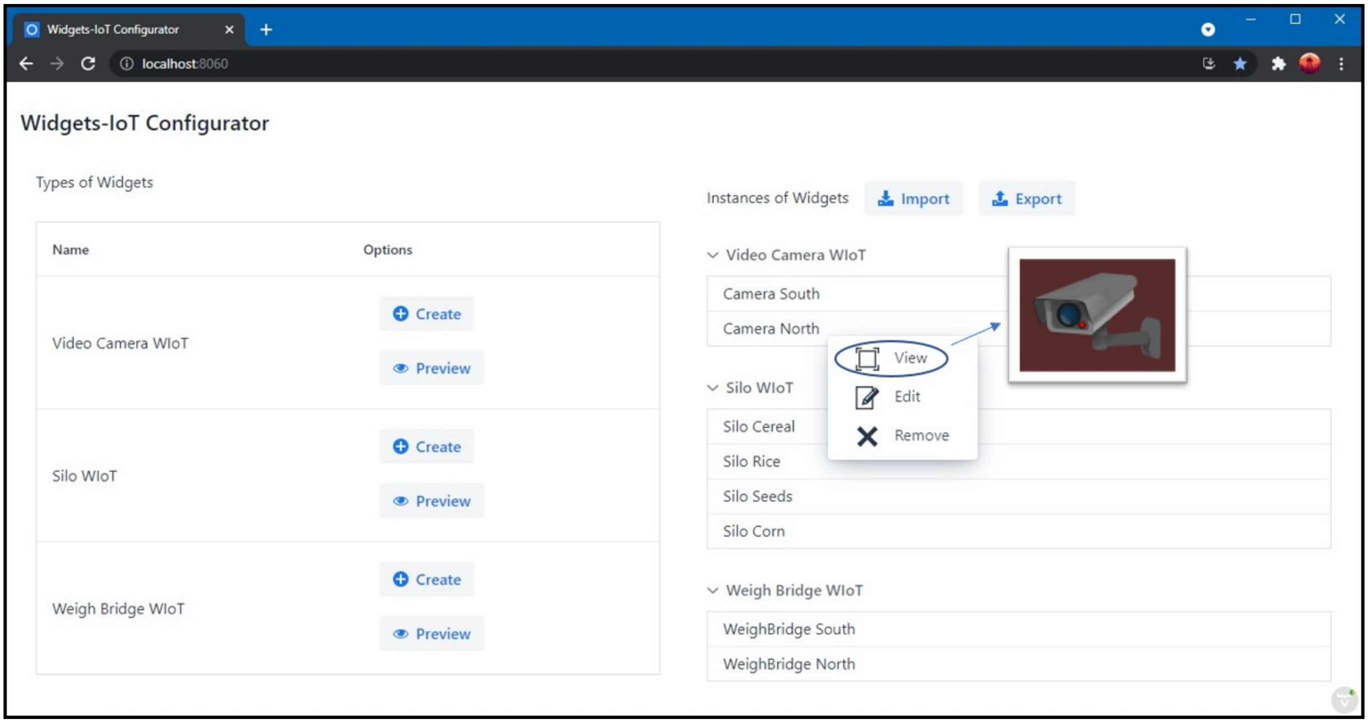

Fig. 4. Widgets-IoT Configurator web interface.

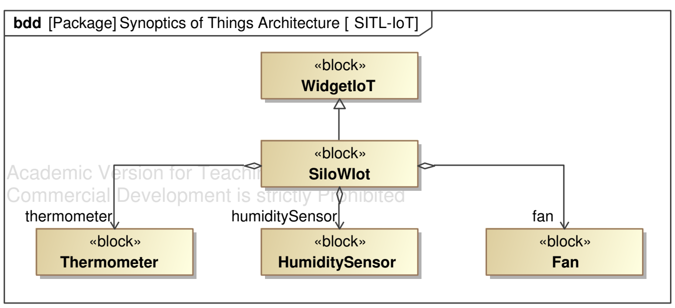

Fig. 5. SysML block definition diagram for a silo Widget-IoT.

in Fig. 6. As it can be seen this `Widget-IoT` can represent distinct operation scenarios. The green color represents a silo operating under normal conditions. Conversely, the red color represents an abnormal working condition requiring the operator action, e.g., a specific zone of the silo being at a high temperature due to low humidity. In such scenario, the operator would take the necessary actions to avoid combustion and damage of the silo, such as activating a fan to lower the temperature (this fan could be a Web Component associated to this `Widget-IoT`). As it can be seen in Fig. 6 it is also possible for the operator to visualize the filled capacity of the silos.

V. Conclusions and Future Work

This paper presents and discusses the Synoptics of Things framework that aims to facilitate the supervision of industrial processes with web based Synoptics. These Synoptics are implemented using `Widgets-IoT`, which are web components representing abstractions of the IoT devices and the software services of the informatics systems involved in such industrial processes. A `Widget-IoT` provides mechanisms not only to monitor the corresponding ”Thing” but also to control its state. Therefore, by using the proposed SoT framework it becomes possible to visualize all the elements of a technological landscape as fundamental ”Things” under a modern supervisory control and data acquisition system (SCADA) based on open web standards. The paper also discusses the application of the SoT framework in the SITL-IoT project, showing that it was successfully used to monitor and control different types of ”Things” modeled as `Widgets-IoT`, from IoT devices to software services.

For future work, we envision the concept of Synoptics of Synoptics, where a synoptics could represent a higher level view of a process and allow an operator to select and operate other synoticts corresponding to subparts of the process. This higher level of abstraction could be useful if an operator wants to get a view of the state of the whole process but doesn’t want to get into the level of detail of an individual device. Besides, to make a Synoptic more interesting, `Widgets-IoT` could establish connections between each other to represent its associations with other devices.

The information coming from the ”Things” could be showed by the `Widget-IoT` in one of two ways: i) the `Widget-IoT` asks the device for its information from time to time (or when the operator purposefully asks for new information) or, ii) the `Widget-IoT` acts as a subscriber of the information from the device. The second method is much

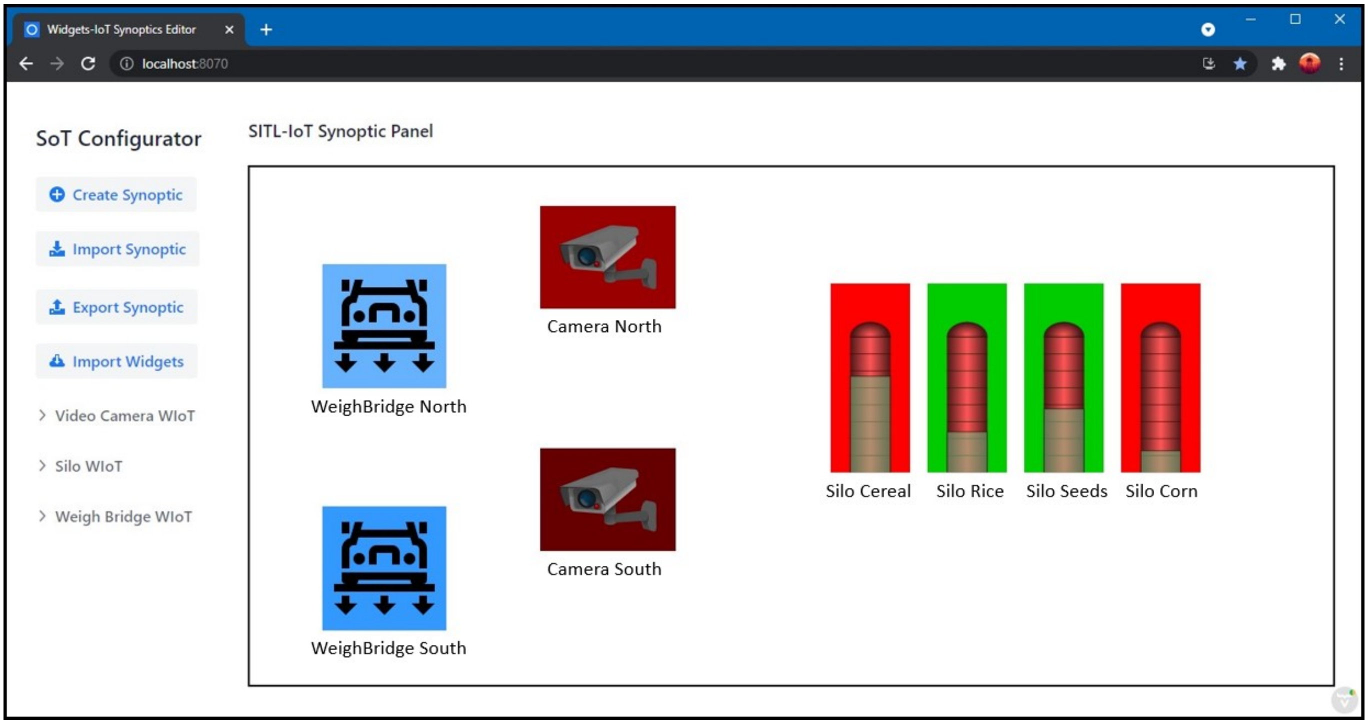

Fig. 6. SoT panel of the study case scenario.

more interesting than the first one, because there is no need to constantly query the device for new information if there were no changes in the state. By using the Publish/Subscribe messaging pattern, the `Widget-IoT` is only notified when the state of the device changes. The Orion Context Broker [10], developed as part of the FIWARE initiative, allows managing the subscriptions and therefore notify clients (in this case, our `Widgets-IoT`) when it is relevant to do it.

We also plan to delve into Augmented Reality (AR), with the help of the WebAssembly runtime [11], in order to allow the operators to navigate through the Synoptic of the process, as if they were in the real world. This way, the operator can have a The WebAssembly runtime (abbreviated Wasm) provides a way to run code written in multiple languages on the web at near-native speed, enabling deployment on the web for client and server applications. Currently, WebAssembly has support for all major browser engines.

Acknowledgment

This work had partial support from the SINCRO/ANSR research project, from the PT2020 SITL-IoT project promoted by FORDESI company, and from BP Portugal through the research project HORUS.

References

[1] R. P. S. Lishev and A. Georgiev, “Laboratory SCADA systems – the state of art and the challenges,” 2015.
[2] B. Farrell, *Web Components in Action*. Manning Publications, 2019.
[3] A. L. Osório, L. M. Camarinha-Matos, T. Dias, and J. Tavares, “Adaptive Integration of IoT with Informatics Systems for Collaborative Industry: the SITL-IoT Case,” in *Collaborative Networks and Digital Transformation*, pp. 43–54, Springer International Publishing, 08 2019.
[4] J. F. Keith Stouffer and K. Kent, “Guide to supervisory control and data acquisition (SCADA) and industrial control systems security,” 2006.
[5] “Adobe Flash Player EOL,” 2021. Available at https://www.adobe.com/pt/products/flashplayer/end-of-life.html.
[6] “Polymer library,” 2021. Available at https://polymer-library.polymer-project.org/.
[7] “Litelement,” 2021. Available at https://lit-element.polymer-project.org/guide.
[8] “SVG: Scalable vector graphics,” 2021. Available at https://developer.mozilla.org/en-US/docs/Web/SVG.
[9] A. L. Osório, A. Belloum, H. Afsarmanesh, and L. M. Camarinha-Matos, “Agnostic Informatics System of Systems: The Open ISoS Services Framework,” in *Collaboration in a Data-Rich World* (L. M. Camarinha-Matos, H. Afsarmanesh, and R. Fornasiero, eds.), pp. 407–420, Springer International Publishing, 2017.
[10] “FIWARE Orion,” 2021. Available at https://fiware-orion.readthedocs.io/en/master/.
[11] “Webassembly,” 2021. Available at https://webassembly.org/.