Journal of Industrial Information Integration 26 (2022) 100276

Contents lists available at ScienceDirect

Journal of Industrial Information Integration

journal homepage: www.sciencedirect.com/journal/journal-of-industrial-information-integration

Full length article

High level control of chemical plant by industry 4.0 solutions

Pierfrancesco Bellini, Daniele Cenni, Nicola Mitolo, Paolo Nesi *, Gianni Pantaleo, Mirco Soderi

Department of Information Engineering, requirements analysis, dashboard design, technical infrastructure, security, process development, data ingestion, process modeling and development, DISIT Lab, University of Florence, Florence, Italy

ARTICLE INFO

Keywords:
IoT
Industry 4.0
Data driven
Synoptics

ABSTRACT

The push towards Industry 4.0 is constraining the industries to work in integrated supply chains. This implies to be open to the integration their production plants with other plants, and to provide access to their data and processes. In most cases, this also means to give access at data and flows to their customers to perform some synchronizations, permit supervision, and to create integrated control rooms, with synoptics and dashboards. This activity is also facilitated by the introduction of IoT solutions with IoT Devices, IoT Brokers, etc., which have a completely different approach with respect to DCS and SCADA solutions usually adopted in the industry for controlling their local productions. In this paper, Snap4Industry with its IoT development environment and framework for implementing the Control and Supervision of complex plant with multiple DCS and/or chains in the view of Industry 4.0 is presented. In particular, the paper describes the motivations/requirements and the actions performed to extend IoT Snap4City 100% open-source platform to comply with Industry 4.0 requirements. The main additions for creating Snap4Industry solution have been on: (i) industry protocols, (ii) custom widgets and synoptics Dashboards, (iii) new MicroServices for Node-RED for enabling the usage of synoptics as event driven devices, (iv) usage of WebSocket secure for the communication with custom Synoptics and Widgets for dashboards, automatized process for producing synoptics templates according to GDPR, and (v) the possibility of creating integrated workflows from data ingestion to synoptics. The paper also presents the assessment of the solution in terms of volume data and messages exchanged. The research has been founded into SODA R&D project founded by Regional Tuscany Gov.

1. Introduction

Most of the Industry 4.0 solutions present control systems as DCS (Distributed Control Systems), SCADA (Supervisory control and data acquisition), PLC (programmable Logic controller), etc. PLC are typically directly connected to the production machines and work with a high-rate regular periodic loop by: reading the status of all variables/inputs, computing control outputs, and finally producing outputs to the actuators (the new setting) [10]. The loop is controlled by precise period since in most cases the computation of the control function depends on the value of the sampling period (T2-T1), such as in the case of PID control (proportional, integrative, and derivative control models), etc., where in most cases the sampling period correspond to the acting period. Above the PLC level, DCS/SCADA solutions are typically applied that may put together more than one PLC solution with supervision activities. In most cases, the DCS/SCADA solutions provide some HMI (human machine interface) for the control room operation. In some cases, the PLCs are replaced by RTU (remote terminal units), which are more general-purpose computer-based solutions for low-level control resulting more programmable, while PLC are typically programmable by using the so-called Ladder Diagrams. PLC programming models are based on simple logic equations (substantially a set of clauses in the form: *If* < *Condition* > *then* < *Action* >) visually formalized with the above semantic for sampling and acting: reading all data for computing Conditions and finally applying Actions by writing outputs. There are streams of discussions in which DCS is considered more process oriented while SCADA more event driven; while in reality, the providers of those solutions are improving them thus making top level DCS and SCADA products very similar. In fact, often, they are used as interchangeable terms when a network of control systems is coordinated to work together, and data collected are stored into some database for historical data observation mainly. In addition, DCS/SCADA also provide suitable

* Corresponding author.
E-mail address: paolo.nesi@unifi.it (P. Nesi).

https://doi.org/10.1016/j.jii.2021.100276
Received 21 January 2021; Received in revised form 28 July 2021; Accepted 28 August 2021
Available online 2 September 2021

control graphic interfaces which may represent the plant as synoptics. The synoptics may represent the status of the plant in real-time with icons, animated icons, numbers, graphs, flows, etc., and in some cases, they are also interactive. This means that the same synoptic elements can be interacted on the screen to send commands to the control network for: changing parameters, turning on/off some switch, controlling valves, etc.

In large industry plants, multiple DCS/SCADA controllers are present to control different production lines and/or machines. Thus the large industries need/use hierarchical solutions for: (i) collecting data from several sources and standards with different rates (from inside the plant an may be outside), both inputs and values of the control parameters also acted by the users in the central control room, (ii) creating the history of the high level data values and decisions taken, (iii) monitoring the status of the system and eventually generating some higher level alarms and actions, (iv) representing the higher level status of the system with some graphical representation and synoptics, (v) connecting the production with maintenance activities. At low level, the different DCS/SCADA typically communicate with PLC and devices with MODBUS, Profibus, CanBUS, FieldBus, HART, AS-I, etc., and others low-level protocols, while at higher level, they may communicate by using protocols such as OPC/OPC-UA, DDS of OMG, MTConnect, etc.

In the context of Industry 4.0 applications, the concept of manufacturing as a service is also addressed. Thus, IoT solutions are going to be integrated in the factory, and among different plants and factories, playing a sort of glue among the different processes, by offering at value chain partners the process data/status and possible actions (see Fig. 1), and may be Manufacturing as a Service, MaaS. To this end, devices such as IoT Edge and IoT Gateways allow to collect sensor data from the element plant which are not directly connected to the control of the single production line (for example, the temperature in the storage, the status of traffic outside the industry, the condition of the trucks), and close the loop with the high-level control via some actuator and/or alarm. In addition, similarly to RTU they can send data towards the DCS/SCADA or on cloud for telemonitoring/control [8,11]. This approach sees the usage of Node-RED / node.js solutions applied in many case [1,14], to implement more flexible solutions and to take into account in the plant also of other external parameters in the connection / integration of the different plant machines and equipment. In fact, Node-RED can support protocols such as ModBus, OPC/OPC-UA, MQTT, AMQP, NGSI, I2C, TCP IP, RS232, etc., and can be used on hardware based on ARM, i86, etc., and on many operating systems (Windows, Linux, Raspberry Pi, etc.). A mixt of IoT and Industry 4.0 and also many home-automation protocols (ALEXA, Philips HUE, TP Link, KNX, ENOCEAN, ZigBee, …) are supported on Node-RED. Thus, Node-RED being based on node.js engine in JavaScript, is probably one of the best solutions for system integration. On one hand, Node-RED may be not very effective in terms of performance since the round trip in: reading, computing and acting presents some limitations in terms of performance with respect to RTU developed in native code as C++/C, Assembly, and Python, and even wrt PLC. On the other hand, Node-RED approach can be a suitable solution for implementing IoT Gateways mixing up different protocols and data sources and thus for rapid prototyping and when high performance is not requested. In fact, there are a number of industrial solutions that use Node-RED as RTU gateway with control logic such as SmartTech [https://netsmarttech.com/page/st-one], for running H24 processes in production lines of relevant customers. [https://iotobox.com/]

In this paper, **Snap4Industry** with its IoT development environment and framework for implementing the Control and Supervision of multiple production chains in the view of Industry 4.0 as a service, is presented. In particular, the paper describes the motivations/requirements and the actions performed to extend IoT Snap4City platform to comply with Industry 4.0 requirements. Snap4City is 100% open source and it is available at [Https://www.snap4city.org], [4], any interested can download and install on local and on cloud without fee. The main additions for creating Snap4Industry solution have been the: (i) addition of a number of industry protocols, (ii) possibility of producing custom widgets and synoptics connecting them respecting security into a Dashboard system, (iii) addition of a number of new MicroServices for Node-RED for enabling the usage of synoptics as event driven devices in/out, (iv) usage of WebSocket secure for the communication with custom Synoptics and Widgets for dashboards, automatized process for producing synoptics templates and instances according to GDPR (General Data Protection Regulation European guidelines), (v) the possibility of creating integrated workflows from data ingestion to synoptics.

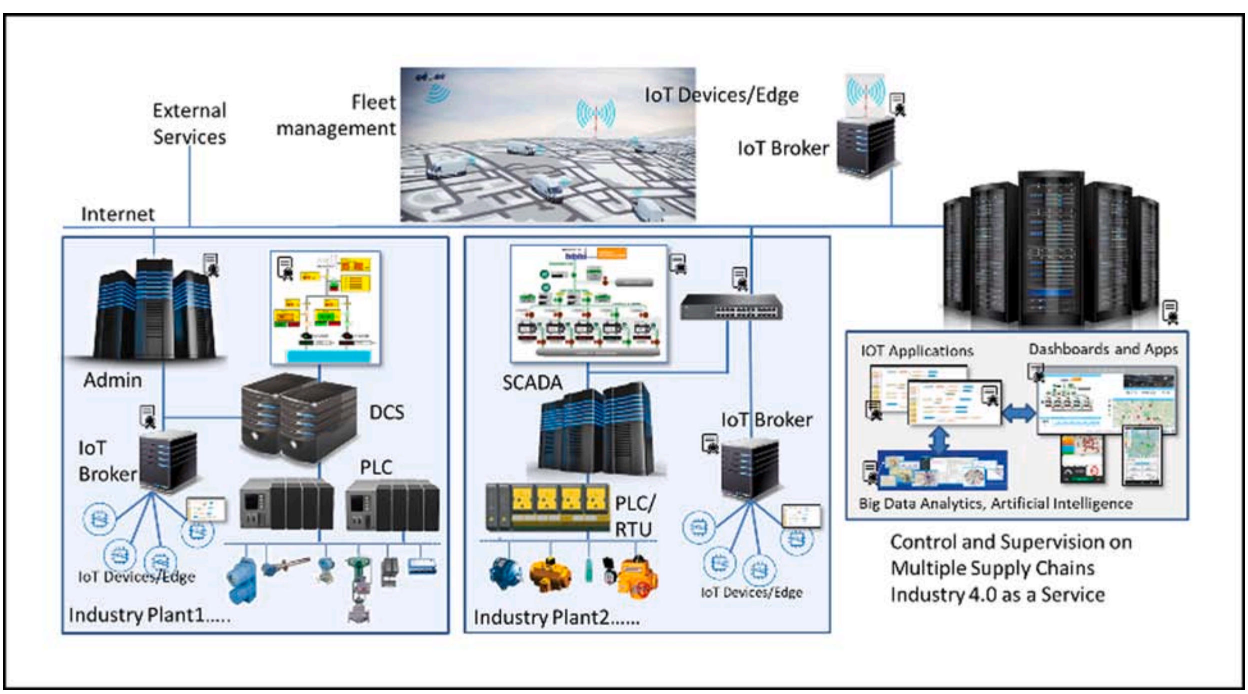


Fig. 1. The integration of Industry 4.0 with the support of IoT, Big Data and cloud technologies .

Please note that the additions have been incorporated in the Snap4City platform to enlarge the coverage of the solution. The new capabilities have been exploited to implement the higher-level control in a large chemical plant in Italy which is ALTAIR. The Altair Chemical has been the first European KOH producer mercury-free chlor-potash plant. ALTAIR SODA-4.0 project aimed to develop and implement an integrated management system and optimization of the production and consumption processes of hydrogen and chlorine. It has been based on a new production structure based on a new NaOH (caustic soda) production plant. Thus Snap4Industry has been fundamental for the realization of an automated system capable of interfacing all the information present in the DCS and in the company management system to make the best decisions in terms of productivity, energy consumption and to carry out operations with the highest possible degree of safety, always going to carry out perfectly coded operations, without committing human errors and / or various oversights. https://www.altairchimica.com/blog/progetti-di-ricerca-e-sviluppo-green-field-peas-e-soda/

This paper is structured as follows. In Section II, the requirements of a solution for control and supervision of multiple chains from cloud, in the view of Industry 4.0 as a service as presented. Section III describes the Snap4Industry architecture. In Section IV, the proposed Production Plan Management System and its implementation for the ALTAIR chemical plant are discussed. The section includes description of continuous production and plan, data ingestion, workflow management, and decision support. In Section V, the development environment for creating synoptics based on SVG graphics is presented. Section VI presents the integration of custom synoptics with the visual programming in IoT Applications/Node-RED and thus the MicroServices developed based on Web Sockets and their performance. In Section VII some performance indicators regarding the volumes on data and messages of the Production Plan Management System are reported. Conclusions are drawn in Section VIII.

2. Requirements for industry 4.0 as a service

As described above, the research reported in this paper consisted on the design and implementation of a solution for covering the business logic and the human machine interaction of Industry 4.0 applications in the context of high-level controls of a large factory among different systems or among different factories working on the same or in any way connected supply chains. Therefore, the main needs of the solution that can cover Fig. 1 can be formalized as follow. *Please note that the context is the usage of the solution for enabling the high-level control and supervision of complex production lines as well as multiple supply chains based on Industry 4.0, by using IoT, cloud, machine learning and big data technologies*. Thus, implementing in some sense Industry 4.0 as a Service, **I4aaS** solution. The following requirements have been identified and elicited on the basis of a set of interviews with a large number of industries in the Tuscany and north of Italy area since the start of the Industry 4.0 wave. In that period the Tuscany Region has started the process for identifying the so called Smart Specialization Strategies, S3, and thus a large number of meeting and visits in the industries facilitated the identification of the requirements. The identification and then the satisfactory of those requirements by the Snap4Industry solutions allowed us to create a solution covering a large range of functionalities demanded by the high level integration, supervision and hyper-automation, which were the main issues demanded.

The following requirements have been divided in functional and non functional. Functional requirements are those that define a function (operative feature) of the system allowing to perform a specific behavior in output on the basis of some input. The non-functional requirement are substantially the criteria which can be used to assess the performance of the system, for instance, security, scalability, modularity, portability, etc.

The solution has to provide support for the following functional requirements:

1 remoting functionalities which are present at level of RTU, SCADA, DCS on cloud for monitoring data and actuations from/to general Control Room and remote devices (e.g., for telecontrol, and to place the basis of as a service solutions). The support consists in the integration of the functionalities of these elements and/or the interoperability with already present solutions, reporting them on high-level dashboards, control panels, mobile devices and synoptics as event driven connections.

2 programming the business and control logics with a uniform visual language, which could be used at low level as well as in the integration on cloud and mobile. DCS and SCADA present their own programming languages for implementing business processes, while typically the remote control or the supply chain integration have to be performed with web-oriented programming. Most IoT development environments are mainly based on developing applications for IoT Devices, or for IoT Edges with native programming languages as happen in the RTU. The adoption of a uniform programming model (at least on Edge and Cloud) provides more flexibility in moving the processes and reusing the code, thus reducing the maintenance costs and increasing reliability.

3 workflow management of integrated activities among the different processes, devices and users. For example, in the management of maintenance ticketing, in the making decision process among the different levels of control, in the dialog with the operators on the field and control room operators, etc.

4 a large number of different protocols, such as those adopted by classic DCS/SCADA but also IoT, Industry 4.0, home automation and smart city protocols. This means that the integration tool and language have to be capable of performing requests by using many different available models and protocols [7], such as pull protocols to obtain data (e.g., Rest Call, Web services, FTP, HTTP/HTTPS, etc.), and push protocols to receive data via data-driven subscriptions (WS, MQTT, NGSI, COAP, AMQP, etc.), with and without the TLS support, using proxy or not, etc.

5 flexible execution and direct communications among business logic processes which can be executed at level of IoT Edge, and on Cloud. The possibility of (i) having processes on premise, on the field or on cloud and the possibility of exchanging data/commands each other would guarantee the possibility of exploiting high performance computational capabilities; (ii) accessing to the IoT Edge processes to program/update them from internet, for example for maintenance [12].

6 exploitation of IoT Devices which are accessible/registered via some IoT Broker, regardless of their protocols. This feature is strongly relevant, since the exploitation of IoT Devices with data driven processes can be used for integrating data sources and actuators in the higher-level control of the industrial plants not addressed by the DCS/SCADA. For example, for monitoring and controlling aspects such as: movements of internal carts, flows of pieces, status of temporary storages, movements of trucks, contextual external data as traffic flow, environmental variables, etc. In fact, the data coming from the former generation of production machines are typically insufficient for controlling the production process and distributed supply chains among different plants.

7 management of IoT data driven privacy according to data ownership and rule and thus of GDPR guidelines [5, 16]. This would allow partners of the supply chain to provide access at their customers to specific data, events and dashboards for direct monitor of the production phases or lots in which they are involved and in real time. This approach is presently very hard to be implemented with SCADA/DCS solutions, that in most cases are not web based and do not provide sophisticated access control.

8 executing data analytics as well as business intelligence and machine learning solutions on powerful processors exploiting big data store on cloud and on premise. For example, for computing: machine learning training, prediction models, optimized production plans, anomaly detection, predictive maintenance taking into account tickets and production, and integrations along the supply chains, etc.
9 exploiting big data storage to save and retrieve historical data related to the whole supply chain but managing the access to them according to the GDPR [3]. This implies that each factory has to be capable to autonomously control the access rights to its data and provides the access to the other industrial partners of the supply chain in a controlled manner. The whole data can be used for creating a knowledge base expert system for computing KPIs (Key Performance Indicator) and deductions. The KPI usually put in relationships the production with the sales factors.
10 exploiting External Services from local and cloud processes to take them into account on the Data Analytic, planning, optimization. For example, for collecting: costs of energy/gas, environmental data, contextual marketing data, fluctuations of contextual data, integrations with other factories and services (for example, delivery and transport, traffic), etc. This feature is viable only if it is possible to connected processes on premise and on cloud with secure communications.
11 creation powerful dashboards on IoT Edge and on cloud (e.g., user interface) for business intelligence and remote control (telecontrol, telemonitoring, supply chain control, etc.) with respect to those that can be produced on IoT Edge computers. Dashboards for remote control have to be capable to: (i) access and drill down on historical data, (ii) provide interactive elements such as Virtual Sensors and Actuators (buttons, knobs, switches, lamps, motors, etc.), to collect actions from the users, as new events for settings/controls the plant, (iii) to interact with on cloud IoT Applications/Processes and which are the business logic of the user interface. Communication with Dashboard can be event driven or in pull, secure and according to GDPR. Data visualization should be expressive enough, including values in real time data, time trends, histograms, pie charts, Kiviat, barlines, multi-trends, etc., but also actuations as custom widgets, buttons, switch, dimmers, selections, etc.
12 producing and exploiting synoptics as graphic rendering of data and plant status and for collecting actions according to the event driven paradigm. They have to be integrated and controllable directly from the IoT Applications.

In addition, the solution has to satisfy a number of **non-functional requirements,** such as: *robustness* (in terms of availability and fault tolerance), *scalability* (to be capable of serving from small to very large businesses with corresponding volumes of processes per second), *security* (authentication, authorization, secure connection, etc.), *full privacy respect* (compliance with data privacy, according to the GDPR), and *openness* in terms of providing the possibility of adding new modules and functionalities, of communicating with any protocol and services and finally to open source.

In this paper, we are focusing the attention on requirements regarding the production of an integrated Production Plan Management System and thus aspects related to: remote control, unified programming model, workflow, integrated synoptics with Node-RED/IoT App, data analytics and planning. Please note that the possibility of addressing Synoptics has led to create a number of new MicroServices that where not present in the Node-RED library, and thus a huge improvement for the IoT and Node-RED communities. As described in the next section, a number of the above-mentioned requirements are verified by the former Snap4City solution, and in [6,3,2].

3. Snap4Industry architecture

In this section, the general architecture of **Snap4Industry** is presented as reported in Fig. 2. Snap4Industry solution has been developed by extending the Snap4City architecture from which it has inherited and maintained: (i) the conformance with GDPR, (ii) the usage of IoT Apps in Node-RED which can be executed on IoT Edge, Cloud and mobiles with the capability of interacting with one another and with users via messages, (iii) the production of Dashboards exploiting data from storage and Km4City Knowledge Base, KB [4], (iv) the production of IoT Apps supporting dashboards, data analytics and resource sharing, minimizing required technical expertise from programmers. IoT Apps are Node.JS processes developed in Node-RED using a large set of Snap4City MicroServices for exploiting the architecture services on: storage, search and retrieval, dashboards, synoptics, user interface controls, data analytics, transcoding, semantic search, etc. [2,3]. The Snap4City MicroServices are distributed into two official libraries of Node-RED nodes by the JS Foundation portal. The two libraries are dedicated to final users (basic) and to developers (advanced). The version for Developers (to be installed on top of the basic version for final users) presents several nodes/blocks that can accept complex JSON message inputs to create strongly dynamic IoT Applications. Both Libraries of Snap4City Nodes can be installed in any Node-RED tool on any operating system: Linux, Raspberry pi, Windows, etc. On such grounds, also in Snap4Industry, the IoT Applications can be obtained as follows:

$$\textbf{IoTApp} = \textbf{Node} - \textbf{RED} + \textbf{Snap4CityMicroServices}$$

The Big Data cluster storage includes an RDF (Resource Description Framework) store for the KB and semantic reasoning, and an ElasticSearch Index (OpenDistro for elastic Search). The whole solution presents textual indexes on all fields, semantic indexes, and elastic aggregations to perform advanced “faceted” filtering queries [4]. For this reason, we can state that Snap4City provided support to satisfy requirements 4, 6, 7, 8, 9, and 10, while the others have been fully satisfied by the extensions implemented for the Snap4Industry. In the former architecture, some relevant limitations were detected in producing real time even driven solutions with end-to-end connections from IoT Devices/Control and Dashboards, which are mandatory features in the Industry 4.0 cases; see for examples requirements: 1, 2, 3, 5, 11, and 12. For these reasons a number of improvements have been designed and implemented to successfully realized **Snap4Industry** and validating in into the high-level control of Altair Chemical plant.

Snap4Industry architecture (see Fig. 2) is capable to address the above-mentioned industry 4.0 aspects.

From left to right, the **Data Ingestion** processes can be in push and pull. **Push data driven ingestion processes** of ingestion are typically performed by:

(a) **IoT Brokers** that: (i) need to be registered on the IoT Directory of the platform to abstract from their protocols and to automatically get registered all their IoT Devices and data models on the KB, (ii) send any message in real time to Apache NIFI and from it into the Elastic Search storage (which plays the role of the so-called data shadow in other platforms). To this end, NIFI is capable to connect any new message as an instance of its specific IoT Device and Model referring to the KB by using the so called ServiceURI. This allows to avoid store static information of the single devices, and to save only dynamic data, optimizing the storage consumption.
(b) **IoT Apps** which are Node-RED processes which can be ready to receive messages in Push according to protocols such as: Web Socket, http/https listening, etc. In this case, the IoT App may adapt/convert the even driven messages to some IoT Device and Broker to reconduct the solution at the (a) case or can send the message to NIFI if the data model is registered. In some cases, the ingested data may need to be saved only into the KB, for example, when real time data change the relationships among city element,

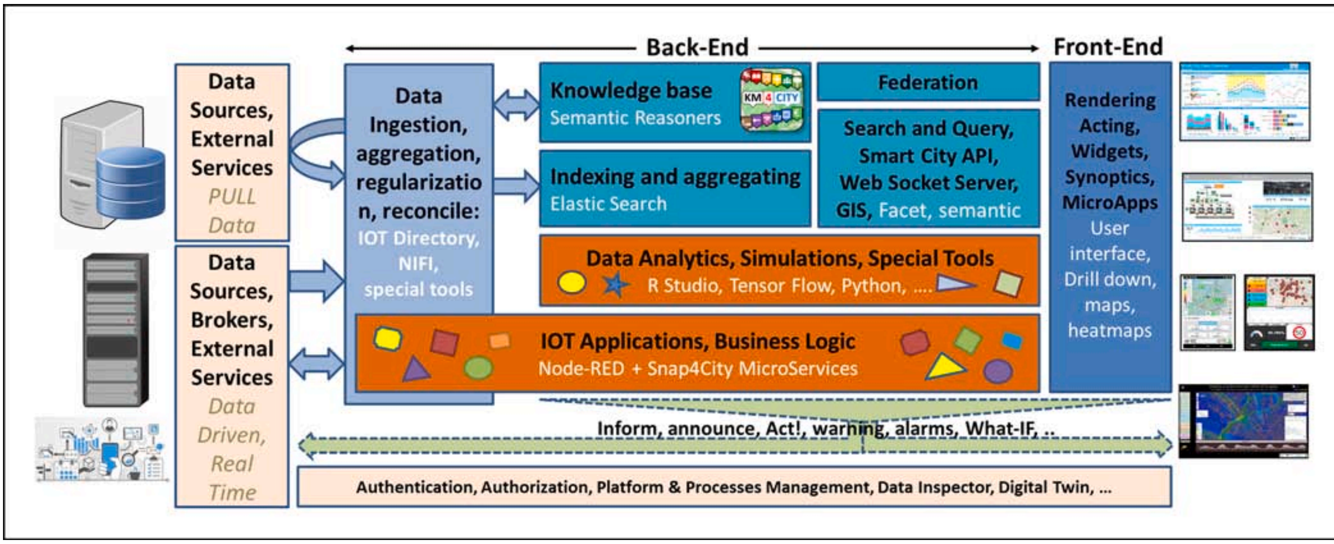


Fig. 2. Architecture of Snap4Industry Solution .

this is the case of GTFS data corrections. The collected data may need to be regularized/reconciled, according to the KB and then pushed into the big data cluster storage, in Elastic Search. The reconciliation process allows for the reconnection of data to entities already in place, for example, by connecting new instances with those already in place elements in KB.

Pull data ingestion processes can be implemented in several manners to collect data from several different sources, protocol and formats. For example, by activating: web crawling, IoT Apps, Python and Java processes, etc. They can exploit pull protocols such as get/post in Http, ftp, Rest Call, WS, etc. All these processes can also send the data as messages into IoT Brokers to reconduct the process to the above case (a), or may send data directly into the KB as in (b). Pull processes may be sporadic or periodic for planned data gathering. In IoT Apps, the pull processes may exploit External Services as Rest Call, which are automatically converted in Nodes for Node-RED. Most of the Industry 4.0 processes provide periodic updates of data such as ModBus, OPC-UA, etc., from SCADA/DCS. They can be regarded as Push/Pull cases.

In the **Snap4Industry architecture**, the data entering in the system can be (i) stored for further processing (data analytics, planning, prediction, anomaly detection, etc.), as well as (ii) consumed in real time for remote dashboards, control and synoptics. The IoT Apps executed on IoT Edge need to directly communicate with the control systems (DCS/SCADA) in the Industry and with Dashboards which in turn may have control elements, synoptics and graphic widgets. The IoT App exploits the basic nodes of Node-RED Node.JS plus Snap4City MicroServices, which have been extended from the former version, as described in the following. Once the data are in the storage, they can be accessed by using APIs which are exploited by the: Data Analytics processes, Business Intelligence layer and Dashboards.

In the following subsections and sections, the main improvements applied to the former Snap4City solution to realize Snap4Industry are described. The changes performed allowed to cover full set of requirements presented in Section II. The main improvements have been the performed to add the possibility of:

a **setting integrated workflows** for defining the business logic of applications involving multiple actors: decision, makers, operators, etc., on their respective devices.
b **producing custom widgets** panel, and connecting them by respecting security and privacy, GDPR [3].
c **event driven Virtual IoT Devices in/out, sensors/actuators as graphic elements (**synoptics and custom widgets**),** thus via new MicroServices, and **WebSocket.**
d **using WebSocket secure** for the communication with Synoptics and custom widgets.
e **producing synoptics templates and instances supporting GDPR**.
f accessing to IoT App (Node-RED) programming interface on IoT Edge from Internet, thus providing support for remote maintenance and development [12].
g the extension of the **KB for modeling Industry 4.0** concepts. This aspect is not reported in this paper, and details can be recovered from [9].

Please note that the additions have been incorporated in the Snap4City platform to enlarge the coverage of the former solution.

4. Production plan management system

In this section, the description of the presented Production Plan Management System and its implementation for the ALTAIR chemical plant are reported.

A typical chemical production plant presents a continuous production H24/7, 365 days/year, taking into account prime matter arriving at the plant and the delivering of the resulting products outside (the delivering is performed by big trucks/tanks 5 days a week, at the working hours). This means that the production plan has to be progressively computed in advance and activated at run time. A **Production Plan** consists of a set of: (i) settings for the production machines for each hour for the next 24/48 h, and (ii) values of simulated/predicted production on the basis of the parameters and comparable with the actual values after production. The production data allows to perform the plan for delivering and transportation of resulting products to be sure that all the production is removed from the local storage in time. Moreover, the plan has to take into account eventual planned maintenance, and the inefficiency corrections due to the degradation of performances of the different machines.

According to the above-reported analysis, the goal has been the design and implementation of a general approach for implementing high-level ***Production Plan Management Systems*** focussed on realizing functionalities not available from standard DCS/SCADA solutions; in particular to: (i) access at the consumptive monitoring of the productions process with progressive daily view and weekly view (the global production of the day and of the week) from any locations and devices, (ii) compute and view KPI of the production efficiency from any devices and locations, (iii) view the real time and history of data and KPI regarding resulting production wrt planned, (iv) perform multiple plans using different parameters and exploit support to make decision about the activation of preferred plan, (v) provide access to re-examine past process productions and plans to learn to improve the performance.

In ALTAIR, the **Production Plan** is decided every day taking the best among Possible Plans computed by mean of an *Optimized Production Planner* which implements the mathematic/chemical formula describing the chemical production process, and by taking in input a number of parameters, and corrective aspects. The complexity of the process consists in the: (a) management of the parameters and of their variations to take the best compromised solution, (b) actualization of the production plant status/prediction over time for computing the *Optimized Production Planner* several times per day (it takes about 15 min for the computation of the plan for the next two days), (c) selection and adoption (put in production) of the Production Plan among the *Possible Plans* computed by exploiting a *Decision Support* (which computes and allows decision makers to take into account of KPIs computed on the basis of the *Possible Plans*, production parameters, status of the plant, orders, etc.). Please note that, each possible plan and thus also the Production Plan has to be feasible according to the plant capabilities. The planned production matter has to be delivered away from the storage plant otherwise the called transports may remain empty, or the opposite, an excess of produced matter, may block the production of the successive day(s) for the lack of safe storage. The general objective is the satisfaction of the orders, may be anticipating some of them when the forecast and the storage allow, but: minimizing the energy costs by taking into account that a relevant amount of energy is also internally produced by the plant as hydrogen and cogeneration; reducing the stock; producing the right products since the input matter may be used for different products, etc.

The above approach has been exploited to implement the Industry 4.0 Production Plant Management System in ALTAIR chemical plant (see Fig. 3). To this end, the solution has been structured in 4 main modules: *Data Ingestion, Production Plan Management, Decision Support, Optimized Production Planner*. Data Ingestion is described in subSection III.A.

4.1. Data ingestion and flows vs IoT applications

This subsection is devoted to the analysis of the Data Ingestion processes and flows based on Snap4Industry and realized for implementing the *Production Plan Management System* for Altair *Control Room*. According to Fig. 3, the *management, predictions* and *optimized production plans* are computed on the basis of the data coming from the:

- *Plant Status* includes about 400 data sources/streams collected/produced by the DCS/SCADA control systems changing every second: (i) managing the different production lines of the industrial plant, (ii) reporting the eventual off-line status of some any element of the production plant, (iii) reporting the status of physical storages and stocks silos for primary matter, intermediate results, and resulting final products.
- *Production Parameters* which are used for setting the plant activity, target efficiency, quality level of productions, preferences in reusing energy and consuming derived intermediate products instead of selling them, target KPIs values, etc.
- *Orders* which are collected from administrative and marketing databases, may be changing every few minutes. From this data source, it is possible to compute the production targets of the next days for each single product on the basis of order, recurrent orders, and preventive stocks.
- *Other Costs* which are collected from external providers via External Services which may change every day such as: (i) Energy Costs, since the plant is also capable to produce energy at the expense of some prime matter, the cost of internally produced energy may be higher or lower wrt those of local market/grid (the energy costs are provided as array of cost for the next 24 h, with a resolution of 1 hour), (ii) delivery and transportation costs/conditions, planning, etc.

Please note that each time one of these parameters change, they impact on results of the *predictions* and *optimized production plan.* This means that: (i) each Possible Plan may differ with respect to the others

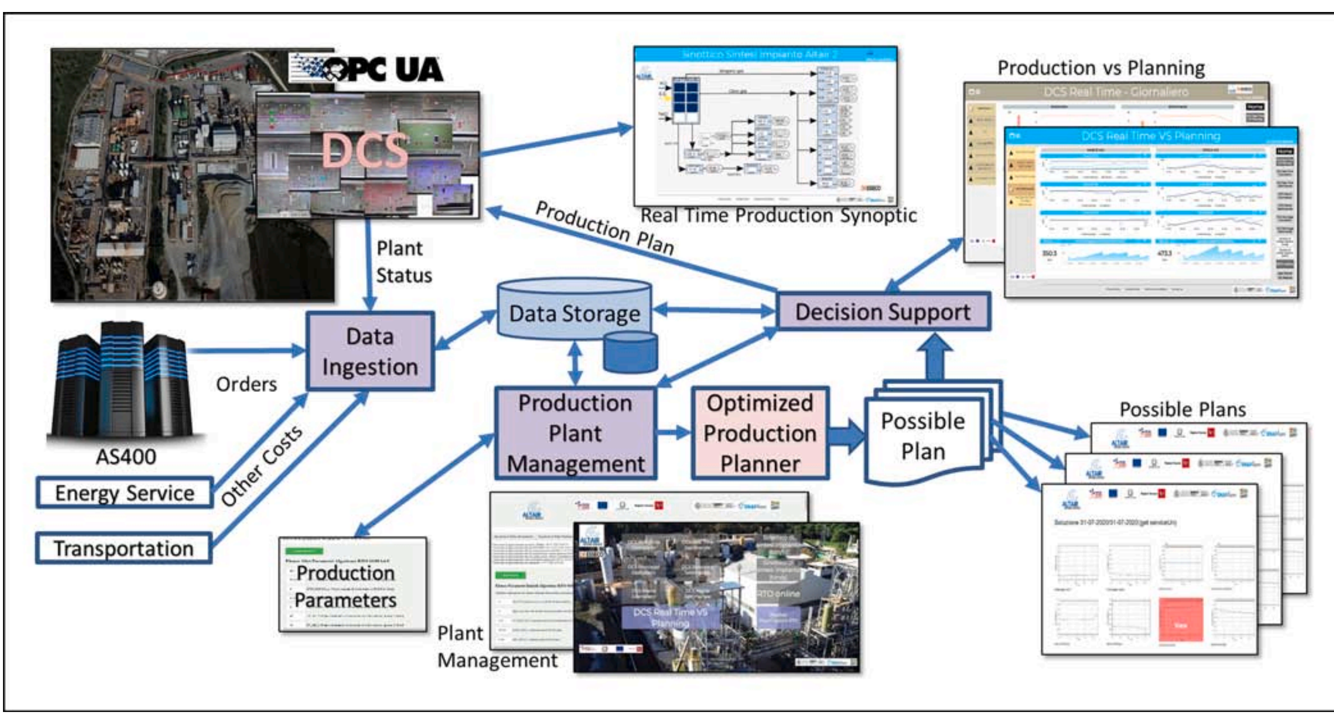


Fig. 3. Altair High-Level flow: from data ingestion to *Production Plant Management System* .

computed on the same day since a large number of parameters, and thus, the decision can be only taken on the basis of selected KPIs, (ii) the computing conditions have to be saved for further analysis of the decision taken wrt the conditions and parameters.

In Fig. 4, the technical implementation regarding data flow of the functional architecture of Fig. 3 is presented. In the solution, IoT Apps are Node-RED processes on Docker containers deployed on private cloud as IoT Edge. Data is received in push and/or pull, and almost every data message with several attributes is considered an IoT Device instance. To this end, the IoT Devices have been registered, and their data structure/model formalized. The formal registration of the *IoT Device* data model is performed into the *IoT Directory* which saves the model into the KB [16, 3]. Once registered, the received IoT messages by an IoT App can be sent into the system via the *IoT Broker* which save them automatically into the *Data Storage*. The IoT Apps can collect data in Push/Pull and in some cases some preliminary computation is performed. IoT Devices from the field (sensors and actuators) can directly send/receive data to/from the IoT Brokers, and the messages are directly saved into the Data Storage at each change, via NIFI Apache. Processes of IoT App, IoT Brokers, Web Sockets, etc., are data driven and thus are activated/fired by the arrival of new messages/events or, in the case of IoT App working in pull, by an internal scheduler.

The arrival of a new message in the Broker may provoke the sending of a new message into the data storage as well as a set of messages towards the user interface clients (dashboards and mobile Apps). In fact, each client browser showing one or more Dashboards and each mobile App connected to the Dashboard Manager need to have established a number of permanent WebSocket connections to receive in real time any change on data. This implies that a large number of Web Sockets may be needed to connect all clients. To this end, Kafka has been adopted in Snap4Industry for distributing messages via those Web Sockets. Please note that the WebSocket connection with user interface clients (dashboards and mobile Apps) is bidirectional since they may provide graphic elements to send command to the system.

In any industrial integration, the above-mentioned data messages in input are differently temporized. The following Descriptions refer to Fig. 4.

DCS/SCADA data messages are received in push from OPC-UA and converted into an IoT App which is posting them as an IoT Device instance according to a defined Data Model via on IoT Orion Broker FiWare. The IoT App play the role of adapter/converter and brings the minimal data into the Data Storage Elastic Search via NIFI Apache. The reception of a new data message on broker leads to send the message to NIFI. DCS/SCADA may produce a new data set every second. With the aim to perform high-level control, to save all messages may be not necessary; thus, the DCS status is saved every 30–15 s. This may also depend on the velocity of the production process phase. In the specific Altair case, some data conversion is also performed to make them ready for the production plan computation, and for reducing the number of variables to those strictly necessary for the high-level control. The DCS data includes measured values from productions flows, the settings reached by the plant, status of the material storages, etc., such as for AcidoEsausto, FeCl2pot, FeCl3pot, FeCl3std, HCl32, HCl35, HCl36, KOH, HCl, K2CO3aq, NaOH50, that are the most important to represent the plant status and these are the values used by the planner algorithm. The most relevant production results are in ton(t), that is the weights of material over time, while the setting parameters estimated to activate the plan are for example: current in A(t), Temperature in Celsius T(t), primary matter NaCLton(t), etc. The IoT Device Model used to save the real time data of DCS has been also used to save the planned values. This permits to perform 1:1 comparison of planned wrt resulted production. On the other hand, the DCS provides the status of the plant as measured variables every second, while each Possible Plan provides the new values of storage, consumption and settings for 24 h per day, two days in the future starting from the 01:00 of tomorrow, so that 24/48 samples.

Administrative data are orders of the products. They may be short terms one shot order as well as recurrent orders and/or long terms orders one shot or recurrent. The administrative data has to be collected by accessing to the databases – e.g., Oracle, DB2 on AS400 via ODBC interface, and the flow of new order stop with the business hours. Potentially, the AS400 performs a check every 10 min. Every time a new update is provided the IoT App is capable to detect to start the process of update to save the new status. The arrival of any new order may motivate the recomputing of the plan and prediction. In particular, the following parameters are considered for each order: article code, progressive number, article prefix, document number, shipping date, etc. And, this happens a numbers of times per day without any precise rate. Then the information regarding the active orders is processed to compute a target production amounts (in ton) for each possible product code which has to be produced into the next 1–2 days, in agreement with the committed deadlines of the marketing and taking into account the delivering time. Please note that, even if the production plan is typically performed once a day, the planning is always performed for 2 days to be sure to have a recovery plan in the case in which the next day play would

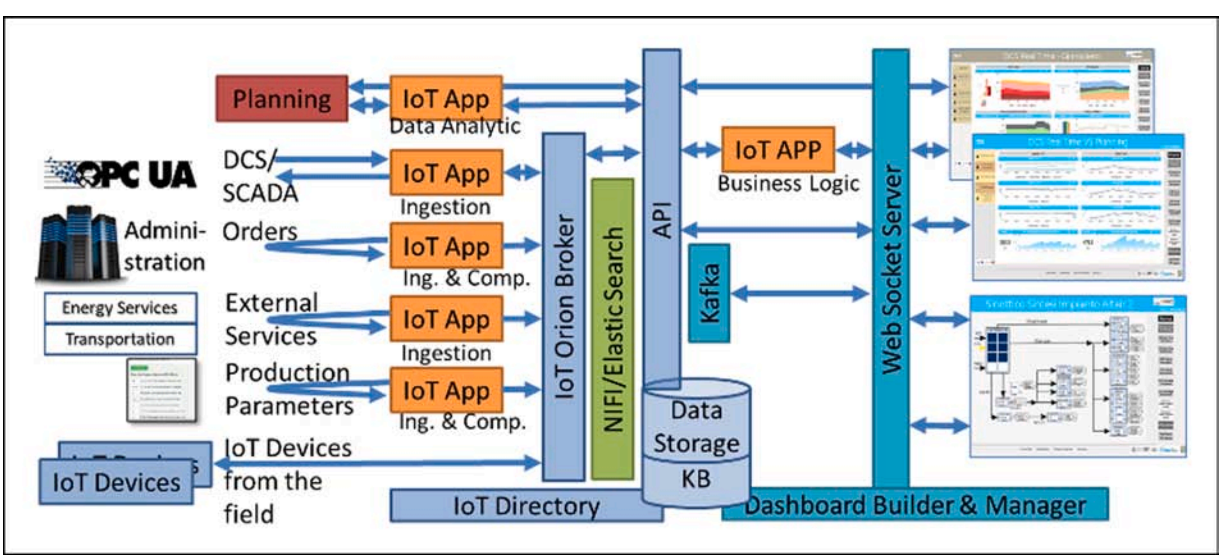


Fig. 4. Snap4Industry: technical architecture of the Data Ingestion and data flows: from data sources, to IoT App processes, storage and Dashboards with custom widgets and Synoptics. .

fail for some reason. It would not be perfect since not actualized, but better than nothing.

External Services are acquired by means of an IoT App in pull and converted in IoT Devices instance to be saved in the storage. They are typically sampled according to their rate of change, for example a single time per day. For example, in the case of Energy the operator provides every day at 15:30 CET from an FTP site the price in euro/MWh for each hour of the next 24 h (mercateoelettrico.org). So that, the IoT Device Model includes a vector of prices. Every day an instance of every external service data is saved into the storage in their corresponding IoT Device to save the initial planning conditions.

Production Parameters are default values in a web page/form proposed to the plant manager by the Plant Management System and are generated by the *IoT App Business Logic* (that is discussed in the following). They may be accepted or modified to be saved and used in the computing of the next production plan. Each modification is written on an instance of its corresponding IoT Device into the storage to save the initial planning conditions.

IoT Devices from the field may be sensors and actuators providing data and accepting to perform actions completely asynchronously with respect to the DCS/SCADA processes. They are typically event driven and may adopt IoT protocols towards IoT Brokers registered on IoT Directory.

4.2. Production plan management vs IoT applications

The **Production Plan Management System** has been realized by implementing the logic and the production of web pages into IoT App by using the same programming paradigm and constructs adopted for the Data Ingestion and Business Logic of the user interface. The most relevant parts of the Production Plan Management logic are reported in Fig. 5, which is substantially a workflow diagram, controlling and synchronizing man, machine e processes. The data ingested are saved into the Data Storage as described in the previous section. Then the Decision Makers (represented in Fig. 5) need web pages forms to see/modify the data, finalize them, and making the final decision selecting the Production Plan. This implies that, they have to be entitled to perform a number of corrections at the inputs parameters before putting in execution the Optimized Production Planner.

The IoT Apps implementing the Production Plant Management System include flows for: (i) producing the web pages in which the data forms are presented, (ii) providing web pages and selection menu to move towards the logic of the application (for example, accepting the provided list of orders or making some changes to obtain the Consolidated Orders, from the List of Active Orders on the Data Storage), (iii) put in execution the *Optimized Production Planner* (execute the optimization) [15] passing to the process the last values of the input data which have been saved into the Data Storage, (iv) provide access to the history of the executions, including the evidence of what has been selected, decisions taken.

The IoT Apps exploit a database in which the status of the user interaction and the history of executions of the Optimized Production Planner are saved together with the information about which plan has been activated and put in production. The workflow formalized into IoT Apps produces the web pages of the user interface and is capable to react at the changes performed by Optimized Production Planner process which is written in Python, and that works asynchronously with the IoT App, producing the results on database.

Please note that, the processes for computing a Possible Plan by using the Optimized Production Planner has to be based on a coherent set of input data sets of the above-mentioned data. To this end, before to activate the computation, the set of inputs are saved for their further reuse to compute alternative Possible Plans by changing selected parameters, for example from Production Parameters, Orders, other costs.

4.3. Decision process vs dashboards and synoptics

The set of computed Possible Plans for next day is compared to identify the preferred Production Plan. In fact, the optimization may need to be executed by using small differences on parameters, with the aim of improving the planning efficiency and to avoid leading the plant over its secure limits. To this end, the Decision Makers get access to the Possible Plan and compare the plans by using actual/planned data and KPIs (see Fig. 6 with the table of executions). Relevant KPIs are the: (i) exploitation efficiency of the plant that is estimated as the ration from the production (planned, performed) with respect the maximum which can be obtained from the plant, (ii) amount of energy needed to produce a ton of Cl, (iii) reduction of errors performed by the decision makers

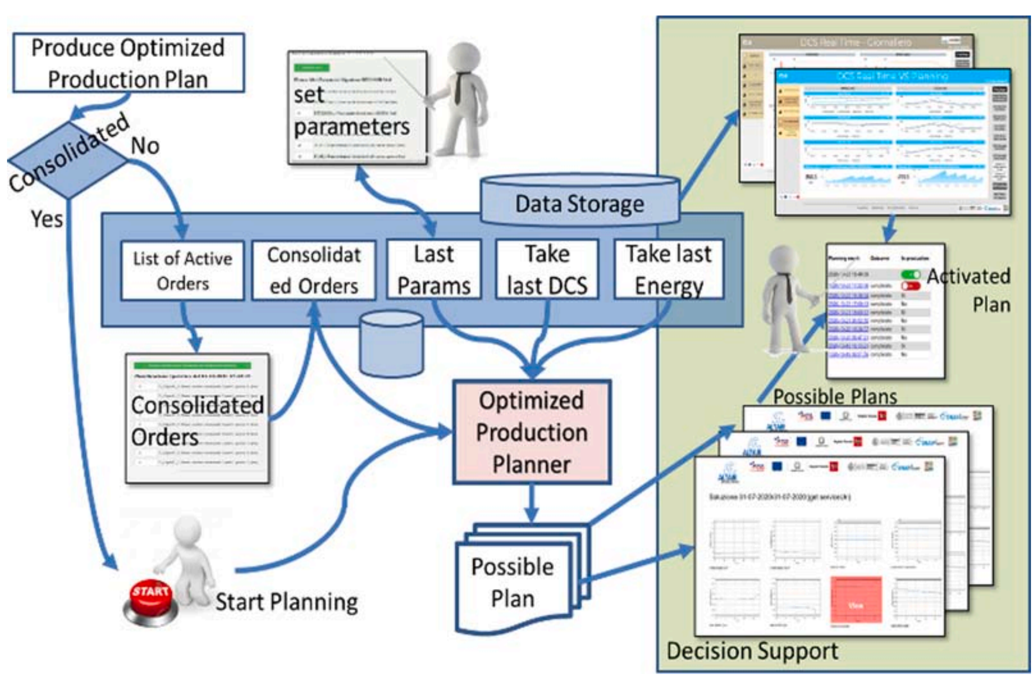


Fig. 5. Production Plan Management flows and logic into IoT Applications.

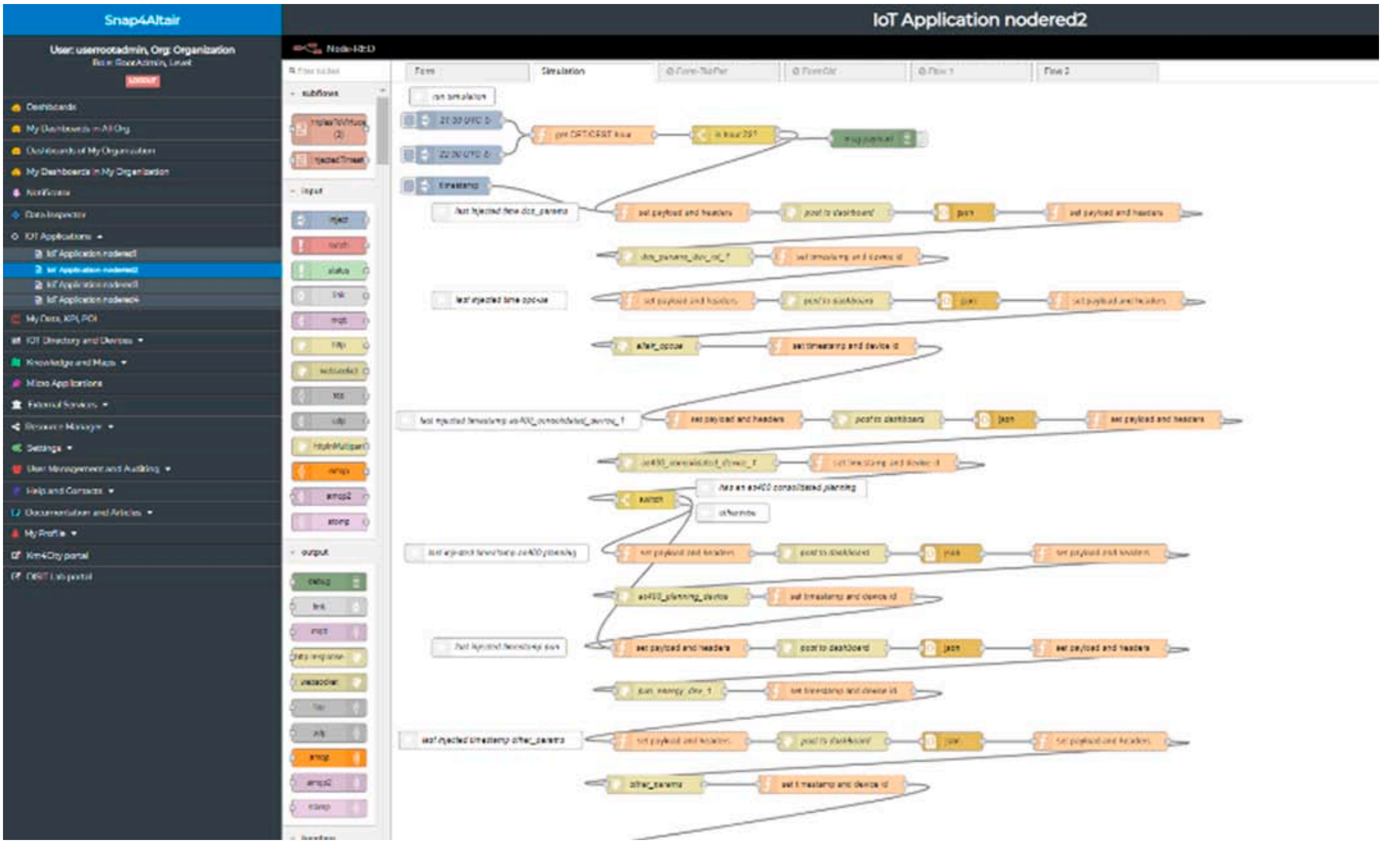
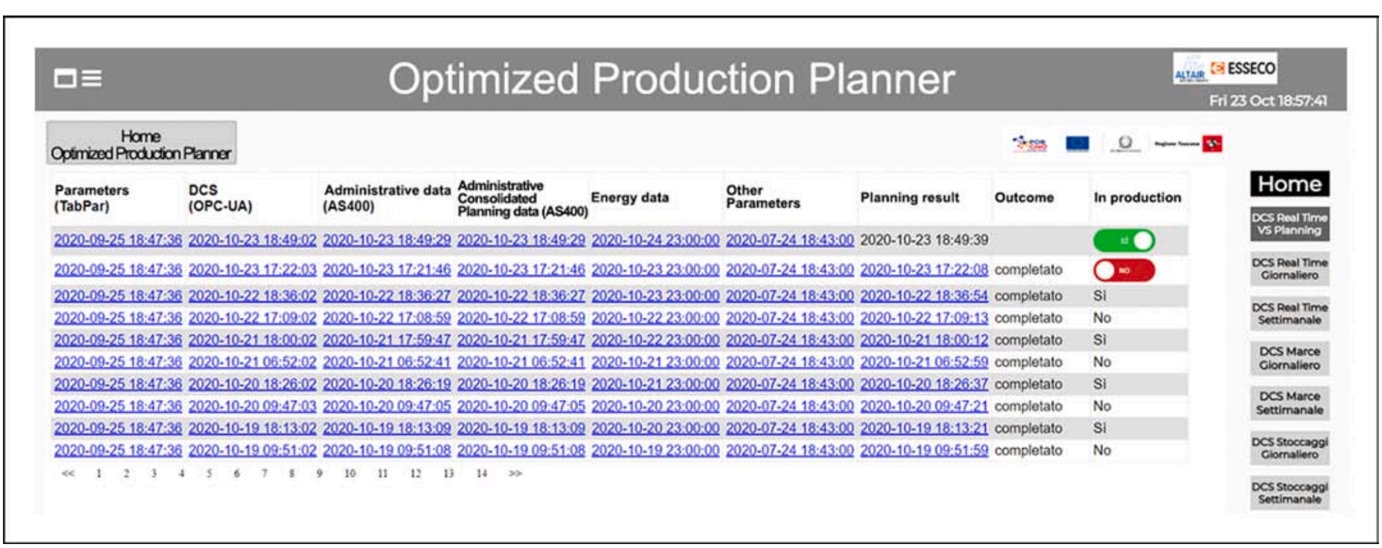


Fig. 6. Example of workflow into IoT App, Node-RED.

with respect to the maximum efficiency which could be obtained, and that has been obtained before the introduction of the supervisor.

In Fig. 7, the web interface, dynamically produced by the IoT App, reporting the executions of the Optimized Production Planner and the selected Planned Plan is presented. Each row represents an execution and allows, through a button, to select the Possible Plan to become the Production Plan "in production". Table columns include: DCS parameters, OPC-UA (DCS), Administrative data (AS400), Administrative Consolidated Planning data (AS400), Energy data, Other parameters, Planning result, In production. Every parameter column reports a link to the respective device data used to run the planning algorithm. The user is able to select an execution for production up to 11pm of every day so that from the 00:00 the control room and plant can work with the new plan. For all the weekdays the actualization is performed for 24 h. On Friday, the actualization has to plan for 3 days since no new orders arrive on Saturday and Sunday.

The whole workflow and thus the Business Logic and web pages of the *Production Plant Management System* has been implemented by using the IoT Apps and the Dashboard Builder. This also means that the IoT Apps are capable to activate other IoT Apps performing and managing algorithms of Prediction and Planning which are implemented in Python. Thus, resulting in a uniform visual language to formalize data



Parameters (TabPar)	DCS (OPC-UA)	Administrative data (AS400)	Administrative Consolidated Planning data (AS400)	Energy data	Other Parameters	Planning result	Outcome	In production
2020-09-25 18:47:36	2020-10-23 18:49:02	2020-10-23 18:49:29	2020-10-23 18:49:29	2020-10-24 23:00:00	2020-07-24 18:43:00	2020-10-23 18:49:39		sì
2020-09-25 18:47:36	2020-10-23 17:22:03	2020-10-23 17:21:46	2020-10-23 17:21:46	2020-10-23 23:00:00	2020-07-24 18:43:00	2020-10-23 17:22:08	completato	no
2020-09-25 18:47:36	2020-10-22 18:36:02	2020-10-22 18:36:27	2020-10-22 18:36:27	2020-10-23 23:00:00	2020-07-24 18:43:00	2020-10-22 18:36:54	completato	Sì
2020-09-25 18:47:36	2020-10-22 17:09:02	2020-10-22 17:08:59	2020-10-22 17:08:59	2020-10-22 23:00:00	2020-07-24 18:43:00	2020-10-22 17:09:13	completato	No
2020-09-25 18:47:36	2020-10-21 18:00:02	2020-10-21 17:59:47	2020-10-21 17:59:47	2020-10-22 23:00:00	2020-07-24 18:43:00	2020-10-21 18:00:12	completato	Sì
2020-09-25 18:47:36	2020-10-21 06:52:02	2020-10-21 06:52:41	2020-10-21 06:52:41	2020-10-21 23:00:00	2020-07-24 18:43:00	2020-10-21 06:52:59	completato	No
2020-09-25 18:47:36	2020-10-20 18:26:02	2020-10-20 18:26:19	2020-10-20 18:26:19	2020-10-21 23:00:00	2020-07-24 18:43:00	2020-10-20 18:26:37	completato	Sì
2020-09-25 18:47:36	2020-10-20 09:47:03	2020-10-20 09:47:05	2020-10-20 09:47:05	2020-10-20 23:00:00	2020-07-24 18:43:00	2020-10-20 09:47:21	completato	No
2020-09-25 18:47:36	2020-10-19 18:13:02	2020-10-19 18:13:09	2020-10-19 18:13:09	2020-10-20 23:00:00	2020-07-24 18:43:00	2020-10-19 18:13:21	completato	Sì
2020-09-25 18:47:36	2020-10-19 09:51:02	2020-10-19 09:51:08	2020-10-19 09:51:08	2020-10-19 23:00:00	2020-07-24 18:43:00	2020-10-19 09:51:59	completato	No

Fig. 7. Controlling and selecting Production Plan “in Production” from Possible Plans “planning results”. .

processes of: data ingestion, data conversion/adaptation, workflow, analytic and business logic.

Therefore, the Decision Maker can access to a number of Dashboards and synoptics to select the preferred plan (see Fig. 8). **Dashboards** have to provide: (i) real time monitoring of the production plant status, and can be accessible to supply chain partners to different kinds of users (set of dashboards allow to control the production of 15 different products in real time with their several parameters), (ii) views the real time and history of production values wrt planned and wrt the settings imposed and KPI, which allow to verify if the production is aligned to the plan, (iii) access and view multiple possible plans predicted values and controls, and eventual errors and critical warning and limits identified in the computation, (iv) produce events/settings for the general integrated system (e.g., the pressing of a button, the setting of a value), and in this case the access has to be strongly controlled to according to the user profile.

These aspects are discussed in the following section on Web Sockets. The new value is produced as a Web Socket message [13] for a corresponding device that reaches the broker, which in turn, it is sending the message to all subscribed processes and IoT Apps. On the other hand, the actions on the dashboards can produce some action on DCS/SCADA as well.

5. Synoptics development environment

In this section, the introduction of synoptics and their integration into the workflow and IoT solution are presented. This new functionality has constrained to change the architecture in agreement to Fig. 4. A synoptic is a graphic panel with graphic elements that may represent plan elements, data, animations and can be interactive. For example, to turn on/off a switch, change a valve status, move slider, insert a value, etc. In Fig. 8, an example is reported, other can be recovered from https://www.snap4city.org . Typically, these graphic layouts are designed in SVG (Scalable Vector Graphics) or other graphics approaches. The advantage of SVG is its integration on Web browsers. The resulting synoptic is typically addressed by proprietary solutions which allow to connect them to DCS and SCADA with proprietary tools. In Snap4Industry all the solution is based on Open-Source components and the new developments have been also released on Affero GPL on GITHUB/DISIT, Snap4City, providing this new feature to all the Node-RED and IoT communities.

In the context of IoT solutions, the possibility of connecting the Synoptics with IoT App has been identified very interesting to cover some of the above-mentioned requirements. In the DCS/SCADA, the usage of the SVG is mainly confined to create synoptics. On the other hand, the power of SVG could be also exploited for creating single interactive/animated graphics Widgets in Dashboards: a single switch, a lamp, an alarm blinking alarm, and a dynamic pin on maps, etc. Thus, our approach started with the possibility of using the design of SVG graphics as a generic tool for creating Custom Widget Templates that can be instantiated to create both single widget and complex Synoptics for Dashboards. At the instantiation time the Template is used to create an actual Widget defining the connections with specific WebSocket streams according to Fig. 4, to play the role of event driven graphic interface.

In Fig. 9, the procedure to create an SVG based Custom Widget into a Dashboard for IoT Control, is reported. The production starts with the creation of the SVG graphic design by using the Inkscape open source SVG Editor (https://inkscape.org/news/2020/05/04/introducing-inkscape-10/). The production of SVG with the editor has to follow a few of guidelines to be sure that the modality in which variables are defined into the SVG are compatible with the successive steps and Snap4Industry tools. The SVG may: contain animations, visualize effects according to the value of some messages received, collect data from the users, be freely zoomed, provide transparent background, etc. In addition, they can be created by exploiting a large part of the market and open libraries of SVG symbols. Once the SVG of the Template is created, it can be uploaded to platform. Thus, a **Custom Widget Template** is created as private of the user, according to GDPR, and can be shared with other users or fully published for everybody by the owner. A Custom Widget Template can be instantiated for creating specific **Custom Widgets/Synoptics,** the identified variables in Reading and Writing have to be linked to WS and associated with some real variable in the system or connected with an IoT App.

5.1. GDPR functionalities and aspects

Please note that: an SVG Template, the corresponding instances as Custom Widget/Synoptic, the dashboards which exploit them, and the related variables connected to the Synoptic: once created are private of the user which has created them. According to GDPR, the user can make them public or can give them in access to other users. In Snap4Industry the grant authorization can be provided to Organizations (all users belonging to), Group of Users, and to single users. All the connections are performed via a WebSocket Secure, into HTTPS, thus implementing end-2-end secure communications even on Synoptics.

6. Performance assessment of synoptics vs WS

In the connection and interaction from Synoptic (read as SVG Custom Widgets) and the rest of the infrastructure, including IoT Apps and storage, there are 3 main Cases as depicted in Fig. 10.

In **Case 1**, the data can be collected from the Synoptic via WS and saved (connected in writing) or read (connected in reading) directly into/from the storage. This is the case in which the data arriving from the DCS and saved into the Data storage are directly show on the Synoptic. This means that for changing the light intensity the IoT App writes into the storage and it is the storage that provokes an event in the WS for changing the lamp status. It implies that the velocity for closing the loop, *round trip (moving the slider and observing the change on the lamp)* depends on the velocity of writing/reading on/from the storage. Please note that, when the slider is moved a new value is saved into the database of MyKPI and an event is created into the IoT App and to the WS of lamp connected. Thus, it has been possible to sustain without loss of messages up to 5 changes per second. Which means 10 WS messages per second (change the lamp and change the slider).

In **Case 2**, the Synoptic is connected with the WS Server which can be read/written directly from specific Nodes of IoT App for communicating with Synoptics. This is the case in which the IoT App directly communicate the KPI values into the Synoptic and may be saving them on Data Storage. This modality has been added to speed up the process and the limit is imposed by the capabilities of WS server and Node-RED to manage event driven processes. In this case, it is possible to save the value in the storage but is not needed to read it again to close the loop that is closed into the Node-RED IoT App on cloud. With the aim of assessing the performance, the IoT App has been subscribed to the WS of the Slider and once the value it sends the value to Lamp and again on the slider with some random changes, so that to create an infinite loop. From the measures performed the round trip is performed at maximum of 20 changes per second without missing data. Which means 40 WS messages per second (change the lamp and change the slider).

In **Case 3**, the Synoptics are connected with the WS Server that directly shortcut the reading with the writing and thus the loop is close in nearly real time. Please note that the round loop is closed passing from the web page to the WS Server via internet without the IoT App. The maximum number of messages per second without loss is about 2500 messages per second (3 sender and 1 receiver). On the other hand, this modality is only used for user interaction and may be used for closing the loop from two WS belonging to different parts of the supply chain, for example, an action in the plant and a visualization toward a different partner remotely connected.

Therefore, for connection synoptics with the IoT App, the Micro-Services (of the Snap4City library on official Node-RED Palette) reported

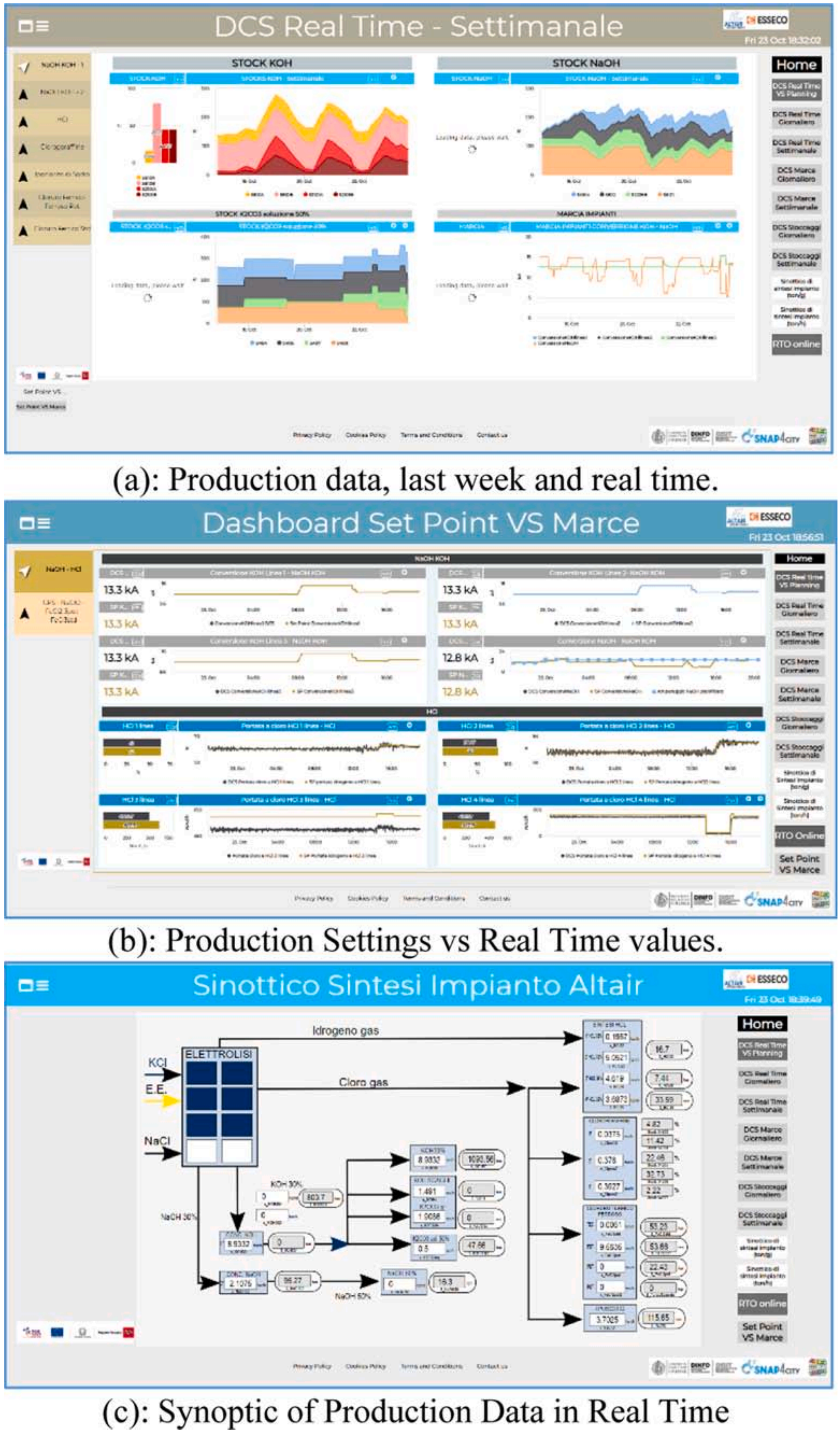


Fig. 8. Dashboards of the Decision Support system.

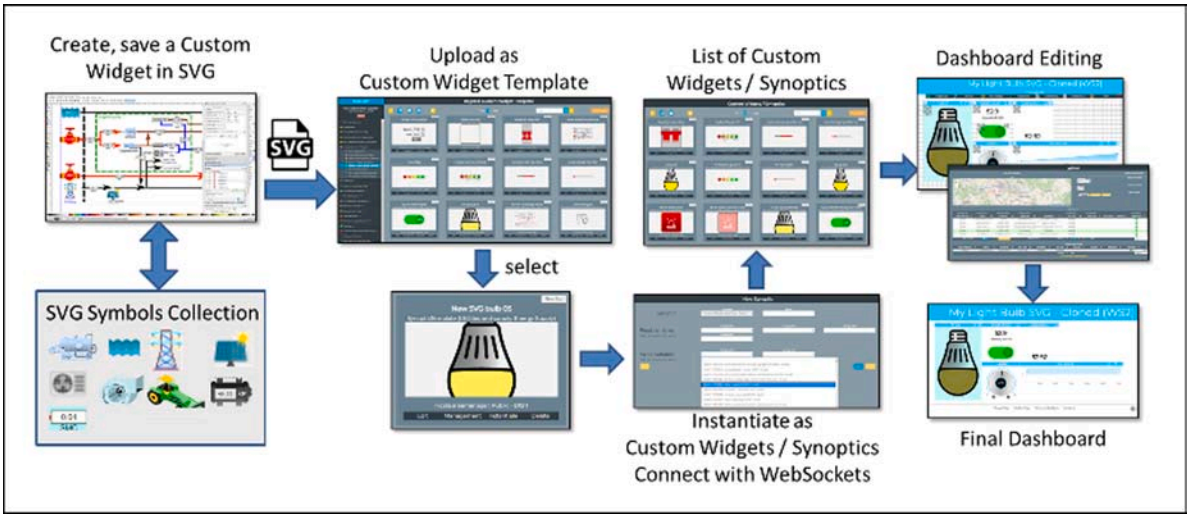


Fig. 9. The process for producing Custom Widgets into Dashboards for graphic description of IoT Control. .

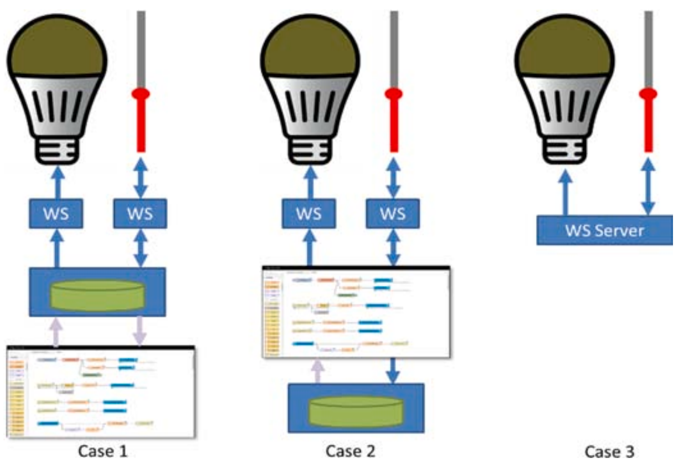


Fig. 10. Connections with SVG Custom widget cases.

in Table 1 have been implemented. In addition, the Synoptics variables can be changed by connecting their corresponding WS to MyKPI or Sensors values of the Snap4Industry platform directly in configuration/instantiation phase.

In the development of solutions, the synoptics may use a mixt of the above-mentioned cases according to the needs in terms of velocity and of saving data changes on storage.

On the other hand, in most of the cases for supply chain integration, it may need to have together: fast events, machines and productions (for example those for manufacturing), and IoT devices just aside to glued. Slow processes for example may be those for: moving material among plants, heating processes, chemical production, electrolysis, milling, etc. In classical DCS/SCADA slower signals are saved in the storage more often than what is really needed, thus limiting the collection of historical data which is mandatory for big data algorithms and predictions. On the other hand, the presented event driven approaches can allow to perform the integration of the different rates with the aim of high-level supervision and control feasible. When the needed rate is not very high as in the DCS solutions closing the loop aside the numerical control.

Table 1
MicroService of IoT app for synoptics.

Send a message into the IoT App once a MyKPI is written.
Send a message into the IoT App once a Synoptic variable is written.
Send a message from the IoT App to a Synoptic variable.
Read the last message of a Synoptic variable.

7. Some performance indicators

In Section IV, some KPI performance indicators regarding the production efficiency have been mentioned. They are mainly related to the: (i) capability of the Optimized Production Planner to perform the correct computation of the 15 different chemical products from the lines, (ii) capability of the decision makers to make the correct decisions in selecting the best planning for the next day(s), and to the (iii) entire solution to avoid critical conditions, recover from micro faults, etc. This paper is mainly focussed on the IoT Infrastructure and not on the first two aspects (i) and (ii), but more at system level. To this end, in this section, some numbers regarding the volume of the activity are reported.

The plant is production 15 different products: Chloroparaffins 45%, 47% and 52%; Ferrous Chloride ($FeCl2$); Potable and Standard Ferric Chloride ($FeCl3$); Hydrochloric acid 32%, 35% and 36%; Potassium carbonate in solution and granulate ($K2CO3$); potassium hydroxide (KOH); Sodium hypochlorite ($NaClO$); Sodium hydroxide 30% and 50% ($NaOH$). The DCS data includes 133 measured variables plus date and time, of which:

- 39 refer to the storages status for the different products;
- 18 refer to the production flows of the products;
- 6 refer to the production flows internally consumed products;
- 18 refer to the amount of product loaded on trucks;
- 35 refer to internal data monitoring the production lines;
- 17 refer to the set points, i.e., the production flows values set by the DCS operators to regulate the production produced on the basis of the Selected Production Plan by the Decision Makers.

In Table 2, the metrics regarding the usage of the system of dashboards are reported. They refer to the most used dashboards in a period of 80 days among the 37 that have been produced for the Production Plan Management system. From the data it is possible to see that almost all dashboards have been accessed every day, only for 3 days we had maintenance in the period. Most of the dashboards have been accessed multiple times and by multiple users. The most requested dashboards are those on control of production vs settings and vs plan (which is called “Set Point VS Settings 1/2″) and on the plan (e.g., “Optimizer production planner”). The Optimized Production Planner has been put in execution with an average of 2.3 times per day. Moreover, the dashboards related to real time and daily activities have been accessed for more than

Table 2
Results of the dashboard system usage.

	N. of Days	Mean number of accesses per day
Snap4Altair Main	75	28,13
Optimizer production planner	77	9,58
Set Point VS Settings 1/2	17	6,76
NaOH KOH - DCS daily	32	5,19
NaOH KOH - DCS weekly	28	5,00
DCS Real Time - daily	36	4,75
DCS settings Real Time - daily	28	3,96
KOH - DCS VC Planning - weekly	41	3,95
DCS Storage - daily	22	3,64
Synoptic Altair daily	42	3,57
DCS Real Time - weekly	39	3,51
Dashboard Set Point VS setting 2/2	17	3,47
DCS Real Time VS Planning	44	3,30
DCS Storage - weekly	33	3,06
DCS setting Real Time - weekly	35	3,03
Optimizer production planner results	15	2,73
Synoptic Altair daily	37	2,54
Cloroparaffine - DCS daily	12	2,50
Cl Ferrico/Ferroso Potable - DCS daily	10	2,50

24,000 min, and those for weekly information for about 8000 min in the period. The

The solution has reduced the time spent in recovering data to control the plant of 68% with respect to typical time spent on monitoring the activities via excel files and reports. It has also reduced the number of errors in the control of the production process.

The volume of data collected and managed are reported in Table 3. From that table it is possible to assess the size of the data needed to be stored for each month which is about 15 M In addition, each data ("doc" in Elastic Search term) is based on 2000 bytes of associated metadata, links, relationships, date and timestamp, etc. So that, the global volume for each year of work is about 350 Gbyte. In this computation we have taken into account only one computation of the plan. On the other hand, even performing 10 computations, the global volume is only marginally changing. The total amount of messages acquired per day is of 1276. The messages exchanged in the platform have been about 20 times bigger since the messages are acquired, saved, visualized, used in the data analytics several times, etc. So that, the usage of multiple Node.JS (IoT Applications have been used).

7.1. Notes on security

The above presented solutions has been developed exploiting the Snap4City architectural solution and tools which have been assessed in terms of security by means of a PENetration test, details are reported on [3]. It should be also noticed that, smart city solutions are much more complex to be defended from cyberattacks since the number of services exposed outside the solutions is typically higher than those exposed in industrial applications. In the cases of Snap4Industry most of the dashboards are only visible from inside the plant and the others via a VPN, and firewalls. So that the risk are strongly reduced, on the other hand, the risk of penetration is the most critical.

Table 3
Data volumes produced by the plant.

	Rate	#var	#Doc 30 days	#msg 30 days	mean doc per day
DCS from OPC_UA	1m	134	5,787,594	43,191	192,919.80
Used DCS on Plan	1h	47	16,967	361	565.57
Orders AS400	10m	12	9,122,784	760,232	304,092.80
Proposed Orders AS400	10 m	106	333,900	3150	11,130.00
Used Proposed Orders AS400	Nx1day	107	3210	30	107.00
DCS Plan Planned	1h	36	25,920	720	864.00
Energy Data	1h	3	2163	721	72.10
Para-meters	10 x day	45	110,404	110,404	3680.13
			15,402,942	918,809	513,431.40

The activities to be performed for a pentest are those reported in the following. During the tests an activity of a stress test was also carried out. The penetration tests were performed following the following steps:

1 Intelligence gathering activities against a target: in this step, information about the target has been acquired using the OSINT (Open Source Intelligence) public accessible sources, and the information was collected using different tools such as: SpiderFoot (https://www.spiderfoot.net/), Maltego (https://www.maltego.com/), Shodan (https://www.shodan.io/).
- Service detection and identification: in this step, the target has been analyzed to find the services exposed and the versions of the used tools. This activity has been performed using tools such as: Necfraft (https://www.netcraft.com) on the main domain, ZenMap to identify its open target ports (https://nmap.org/zenmap such as 80/tcp 443/tcp 5060/tcp 8080/tcp), (https://www.owasp.org/index.php/Category:OWASP_DirBuster_Project). The "theHarvester" script has been used (https://github.com/laramies/theHarvester) to retrieve any user's profiles leaked by the platform; no profiles were found.

2 **Vulnerabilities detection, verification and analysis**: in this step, the target has been analyzed and deeply scanned for the different vulnerabilities using tools such as OWASP ZAP (https://www.owasp.org), Pentest-Tools and Burpsuite (https://portswigger.net/burp) and verified using sqlmap (http://www.sqlmap.org) and xenotix6 (https://www.owasp.org/index.php/OWASP_Xenotix_XSS_Exploit_Framework).

A detailed set of attacks has been performed on domain and subdomains using the OSWAP ZAP and Arachni (https://www.arachni-scanner.com) tools. Even proceeding separately by subdomains, any complete analysis required several hours of computation. An analysis of the false positive alerts has been carried out to highlight just the true risks. For the main domain, a number of URLs have been identified and a complete attack required several hundred thousand requests for about 39 h of computation. The pentests found some vulnerabilities that have been immediately solved.

8. Conclusions

The strong push towards Industry 4.0 has constrained many industries to work at developing integrated and interoperable chains, internal and external with the supply chains. In most cases, it implied to open at the integration of their production processes with those of other industries in the chain, and thus with their customers and suppliers. This fact led to be capable to give access at data and flows and to create integrated dashboards for their customers and to perform some synchronization and supervision, and may create integrated control rooms, synoptics and dashboards, and allow to collect data for big data analytic, business intelligence, machine learning on production processes. This activity has been in most cases facilitated by the introduction of IoT solutions with IoT Devices, IoT Brokers, etc., which provide a completely different approach with respect to DCS and SCADA solutions usually adopted in the industry for controlling their local productions. In this paper, **Snap4Industry** architecture with its IoT support and development environment and framework allowed the implementation of Production Plan Management Systems to control and supervising multiple chains (and provide integration with supply chains) in the aim of

Industry 4.0. In particular, the paper described the motivations, requirements and the actions performed to extend IoT Snap4City 100% open-source platform to comply with Industry 4.0 requirements. The main additions for creating Snap4Industry solution have been on: (i) a number of industry protocols, (ii) tools for producing custom widgets and synoptics and connecting them respecting security into a Dashboard system, (iii) a number of new MicroServices for Node-RED to enable the usage of synoptics as event driven devices in/out, (iv) the usage of WebSocket secure for the communication with custom Synoptics and Widget for dashboards automating the process for producing synoptics which are GDPR compliant, (v) the possibility of creating integrated workflows from data ingestion to synoptics, also addressing data analytics for production simulation and prediction. The resulting benefits for the production plant of Altair have been: (a) the reduction of time needed to understand the status of the production with respect to the planned, (b) the reduction of time needed to make the decision about the best plan to put in production, (c) the reduction of the number of errors in controlling the whole production plan also in introducing the settings into the plant from the selected Production Plan. The research has been founded into SODA R&D RT project and tested on chemical plant of ALTAIR, one of the main chemical industry in Italy.

CRediT authorship contribution statement

Pierfrancesco Bellini: Conceptualization. **Daniele Cenni:** Conceptualization. **Nicola Mitolo:** Visualization. **Paolo Nesi:** Conceptualization, Methodology, Investigation, Supervision, Writing – review & editing. **Gianni Pantaleo:** Writing – review & editing, Visualization. **Mirco Soderi:** Writing – review & editing, Visualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors would like to thank ALTAIR for which we have developed the solutions described in this paper, SODA project partners and Regione Toscana for founding of ALTAIR. Former technologies as Snap4City and Km4City are 100% open source of DISIT Lab and can be accessed at https://www.snap4city.org.

References

[1] Andrei Nicolae, Adrian Korodi, Node-red and OPC UA based lightweight and low-cost historian with application in the water industry, in: 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), IEEE, 2018.

[2] C. Badii, P. Bellini, A. Difino, P. Nesi, G. Pantaleo, M. Paolucci, MicroServices suite for smart city applications, sensors, MDPI (2019), https://doi.org/10.3390/s19214798.

[3] C. Badii, P. Bellini, A. Difino, P. Nesi, Smart city IoT platform respecting GDPR privacy and security aspects, IEEE Access (2020), https://doi.org/10.1109/ACCESS.2020.2968741Badii.

[4] C. Badii, E.G. Belay, P. Bellini, D. Cenni, M. Marazzini, M. Mesiti, P. Nesi, G. Pantaleo, M. Paolucci, S. Valtolina, et al., Snap4City: a scalable IoT/IoE platform for developing smart city applications, in: Proceedings of the 2018 IEEE Smart World, Guangzhou,China, 2018, pp. 8–12.

[5] C. Badii, P. Bellini, D. Cenni, A. DiFino, P. Nesi, M. Paolucci, Analysis and assessment of a knowledge based smart city architecture providing service APIs, Future Gener. Comput. Syst. 75 (2017) 14–29, https://doi.org/10.1016/j.future.2017.05.001.

[6] P. Bellini, et al., Smart City architecture for data ingestion and analytics: processes and solutions, in: 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService), IEEE, 2018.

[7] J. Dizdarević, F. Carpio, A. Jukan, X. Masip-Bruin, A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration, ACM Comput. Surv. 51 (2019) 116.

[8] S. Karnouskos, A.W. Colombo, Architecting the next generation of service-based SCADA/DCS system of systems, in: IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society, Melbourne, VIC, 2011, pp. 359–364, https://doi.org/10.1109/IECON.2011.6119279.

[9] Km4City ontology (2020) https://www.snap4city.org/19, https://www.snap4city.org/download/video/DISIT-km4city-City-Ontology-eng-v5-1.pdf.

[10] Reinhard Langmann, Leandro F. Rojas-Peña, A PLC as an Industry 4.0 component, in: 2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV), IEEE, 2016.

[11] Yang. Lu, Cyber physical system (CPS)-based industry 4.0: a survey, J. Ind. Integr. Manage. 2 (2017), 1750014, 03.

[12] G. Merlino, D. Bruneo, S. Distefano, F. Longo, A. Puliafito, Enabling mechanisms for Cloud-based network virtualization in IoT, in: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, 2015, pp. 268–273, https://doi.org/10.1109/WF-IoT.2015.7389064.

[13] Diego RC Silva, et al., Latency evaluation for MQTT and WebSocket Protocols: an Industry 4.0 perspective, in: 2018 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2018.

[14] Mohamed Tabaa, et al., Industrial communication based on modbus and node-RED, Procedia Comput. Sci. 130 (2018) 583–588.

[15] M. Vaccari, R. Bacci di Capaci, E. Brunazzi, L. Tognotti, P. Pierno, R. Vagheggi, G. Pannocchia, Implementation of an Industry 4.0 system to optimally manage chemical plant operation, in: Proceedings of 21st IFAC World Congress, Berlin (Germany), 2020.

[16] S. Valtolina, F. Hachem, B.R. Barricelli, E.G. Belay, S. Bonfitto, M. Mesiti, Facilitating the development of IoT applications in smart city platforms, in: Int. Symposium on EndUser Development, Cham, Switzerland, Springer, 2019, pp. 83–99.