FIWARE
Open APIs for Open Minds

FIWARE IMPACT STORIES

SMART MANUFACTURING

CAPRI Platform & the PySpark Connector: a bridge between FIWARE and Apache PySpark

With the contribution of

ENGINEERING
THE DIGITAL TRANSFORMATION COMPANY

FIWARE - OPEN APIs FOR OPEN MINDS
October 18, 2023 @ FIWARE Foundation, e.V. – www.fiware.org

Challenge & Context

European process industries stand on the brink of a great transformation to become circular and climate-neutral by 2050[1]. They face energy and environmental efficiency issues due to shortages in raw materials supply, energy prices, and anti-pollution policies, which represent important constraints that must be dealt with, forcing them to adapt and adjust their manufacturing processes, regardless of their complexity[2].
Digitalization enables faster development of new solutions and speeds up engineering. It also allows for more intelligent operations and facilitates cross-sectoral collaboration by tracking data and sharing information in an effective and secure manner.

The European co-funded **CAPRI**[3] (Cognitive Automation Platform for European PRocess Industry digital transformation) project brings **cognitive solutions** to the **process industry** by developing and testing an innovative Cognitive Automation Platform (CAP). It has been **validated** in three different **use cases** - namely within the **asphalt, steel and pharmaceutical industries**, leveraging the combination of sensors, information technologies, and new AI algorithms called **"Cognitive Solutions"**. It provides **cognitive tools** that enable more **flexibility** in operation, **improve performances** across different indicators (KPIs) and **ensure state-of-the-art quality control** of products and intermediate flows.

[1] European Commission, Internal Market, Industry, Entrepreneurship and SMEs - Transition pathways for European industrial ecosystems, News Release, 2021-2023.
[2] European Commission, Internal Market, Industry, Entrepreneurship and SMEs - Transition pathways for European industrial ecosystems, News Release, 2021-2023.
[3] Digitalisation represents a new challenge for the European process industries, which need to handle an increasingly wide range of actions. Cognition capabilities will permit the sector to improve its flexibility and performance. The EU-funded CAPRI project will establish, test and demonstrate an advanced cognitive automation platform (CAP) for process industry digital transformation. The platform will help the sector increase its flexibility of operations and improve performance through different indicators and cutting-edge quality control of products and intermediate flows. The CAP will be modular and scalable, allowing the development and integration of advanced applications that address manufacturing challenges in significant process sectors such as asphalt, steel-making and pharma.

Three main technical objectives have been pursued:

- digital transformation and automation of the process industry through technologies like data collection, storage, and knowledge extraction, to provide detailed insights into process control and resource availability;
- improved performance and flexibility in the process industry via digitalization to dramatically accelerate changes in resource management and in the design and the deployment of disruptive new business models;
- next-generation process industry plans for autonomous operation of plants based on embedded cognitive reasoning, while relying on high-level supervisory control, and providing support for optimised human-driven decision-making.

The **Cognitive Automation Platform** aims to support the entire data flow, from data collection to data utilisation, including:

- **Facilitating data acquisition** from heterogeneous sources (IIoT and custom systems).
- **Allowing the historicization of data** generated in the IoT or industrial field on ad-hoc storage.
- **Enabling the usage of data** supporting applications.
- **Providing a business analytics suite** based on machine learning and cognitive algorithms.
- **Ensuring the persistence of the output** derived from the analytics performed (both in terms of models and predictions).
- **Managing business analytics** based on batch data and streaming data.
- **Allowing the management of edü ge** or wide scenarios.
- **Integrating security modules** for user management.
- **Upholding data sovereignty** principles.

From this scenario, the PySpark Connector[4] was born, **with the aim of enriching the FIWARE solution for Python[5]**, a commonly used programming language. This need arises from the necessity to exchange data bidirectionally between the previously mentioned Cognitive Solutions and the Cognitive Automation Platform.

Solution

The CAP is based on a layered reference architecture, as depicted in Figure 1, which can be summarized into several functional macro-components:

- Smart Field contains the Industrial IoT (IIoT) physical layer, composed of machines, sensors, devices, actuators and adapters.
- External Systems defines enterprise systems (ERPs, PLMs, customized, etc.) for supporting processes and adapters.
- Smart Data Management and Integration contains the Data Management and the Data Integration sub-modules. Regarding Data Management, it defines information and semantic models for data representation of Data in Motion (DiM), Data at Rest (DaR) and Situational Data. Furthermore, this component is responsible for data storage, data processing, and the integration of data analytics and cognitive services.
- Smart Data Spaces and Applications represents the data application services for representing and consuming historical, streaming, and processed data.

[4] FIWARE PySpark Connector is a FIWARE Generic Enabler (GE) made of a receiver and a replier subcomponents allowing a bidirectional communication between the FIWARE Context Brokers (CB) and PySpark. The component works on a low-level socket communication implementing a message passing interface between the two aforementioned counterparts. This interface is equipped with a parser function, hence permitting the creation of both NGSIv2 and NGSI-LD entities ready to use in a custom PySpark algorithm. Once data are preprocessed inside the PySpark environment, the component also provide a write-back interface (via REST API) to the CBs.

[5] Nowadays, Python is one of the most widely used programming languages. It leads significantly in AI and neural networks due to the fact that it comes with prebuilt libraries such as Numpy for scientific calculations, Scipy for advanced computing, and Pybrain for machine learning. For these reasons, the Cognitive Solution developers in the CAPRI Project selected this language to implement their algorithms.

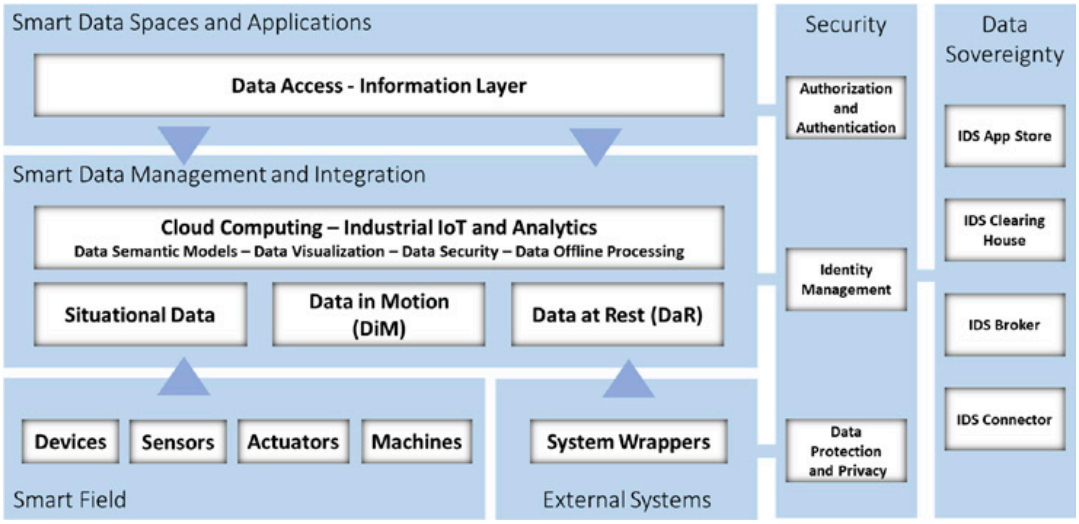

Figure 1 - CAP Reference Architecture

- Security defines components for the authorization and authentication of users and systems. It also integrates modules for data protection and privacy.

Components in every layer can be combined following a Lego-bricks-like approach, adhering to the exposed data schema, **making the architecture flexible and adaptable to the specific needs of the various application domains in the process industry**. Simultaneously, the modularity enables the adoption of a microservice design for the application, resulting in smaller software code that can be organised as docker containers. This allows them to run on smaller processing elements and with restricted resources.

In the pharmaceutical use case, the CAP has been developed using Open-Source technologies from the FIWARE catalogue and is based on Apache[6] technologies, as shown in Figure 2.

[6] Apache HTTP Server is an open source cross-platform web server. It is fast and secure enough to be used to run major websites. There is no licensing cost, making it cheap for small projects. It can also be extended with modules and add-ons to meet almost any website need.

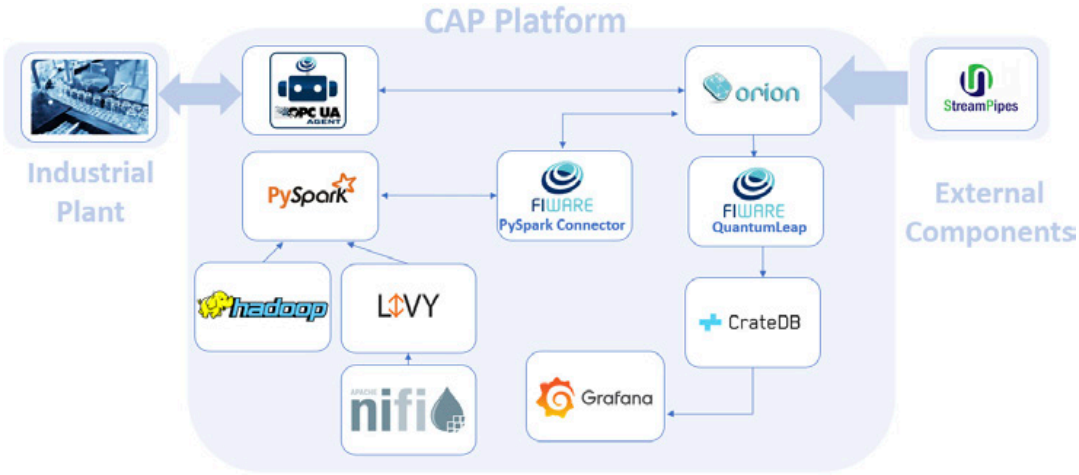

Figure 2 - Pharma CAP Blueprint

The industrial plant allows the CAP to operate in streaming mode (Data in Motion), but through Apache Nifi it is possible to feed the CAP in batch mode (Data at Rest). With Apache Livy, the execution of PySpark commands becomes feasible.

Once the data from the industrial plant is collected and published in the Orion Context Broker, the bidirectional exchange of data from Orion to the cognitive solutions integrated in Apache PySpark is facilitated by the **FIWARE PySpark Connector**, a component acting as a bridge between FIWARE and Apache components.

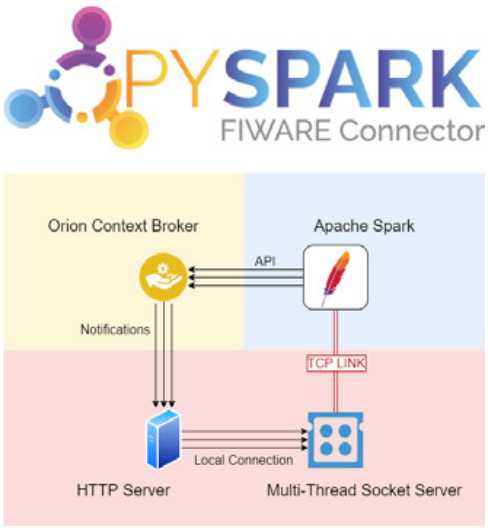

Figure 3 - PySpark Connector Architecture

PySpark Connector

The FIWARE PySpark Connector is an incubated FIWARE Generic Enabler composed of a receiver and a replier, allowing bidirectional communication between the FIWARE Context Broker[7] and Apache PySpark. The component operates through low-level socket communication, implementing a message-passing interface between the two counterparts. This interface is equipped with a parser function, enabling the creation of NGSIv2 or NGSI-LD[8] entities for use in a custom PySpark algorithm. Once data has been pre-processed within the PySpark environment, the component also provides a write-back interface (via REST API) to the Context Broker.

[7] FIWARE NGSI is the API exported by a FIWARE Context Broker, used for the integration of platform components within a "Powered by FIWARE" platform and by applications to update or consume context information. FIWARE NGSI API specifications have evolved over time, initially matching NGSI-v2 specifications, now aligning with the ETSI NGSI-LD standard. The FIWARE Community plays an active role in the evolution of ETSI NGSI-LD specifications which were based on NGSIv2 and commits to deliver compatible open source implementations of the specs.

[8] A Context Broker component is the core and mandatory component of any "Powered by FIWARE" platform or solution. It enables to manage context information in a highly decentralised and large-scale manner. A wide variety of Data Connectors are also available.

As Python is currently one of the most widely used programming languages for data analysis, boasting an extensive array of scientific libraries for data processing and visualisation, the FIWARE PySpark Connector serves as a natural extension to the AI domain, further expanding the FIWARE environment. **The FIWARE Orion Context Broker is positioned at the heart of the infrastructure, facilitating the exchange of context information in a powerful manner**, particularly in this case through a new communication channel with Apache PySpark.

How it works

The mechanism behind the PySpark connector is quite straightforward: it sets up a basic HTTP server to receive notifications from the FIWARE Context Broker and then transfers this data into Apache PySpark using the SocketTextStream function, which generates an input TCP[9] source for building Resilient Distributed Datasets (RDDs), the streaming data unit of Apache PySpark. Figure 4 illustrates the detailed process of connector setup, followed by data reception, management, processing, and sinking.

The first phase involves setting up the **FIWAR E PySpark connector**, assuming that the PySpark session has already started. The Prime function of the FIWARE PySpark connector is executed. Once the connector's multi-thread socket server (MTSS) starts, it remains in a listening phase. When the PySpark streaming context is initialised the PySpark's SocketTextStream function creates a TCP input using any available local IP and port, connecting to MTSS. Subsequently, the MTSS preserves PySpark's TCP socket, and the end streaming context and RDD channel are returned.

[9] Transmission Control Protocol (TCP) is a communications standard that enables application programs and computing devices to exchange messages over a network. It is designed to send packets across the internet and ensure the successful delivery of data and messages over networks.

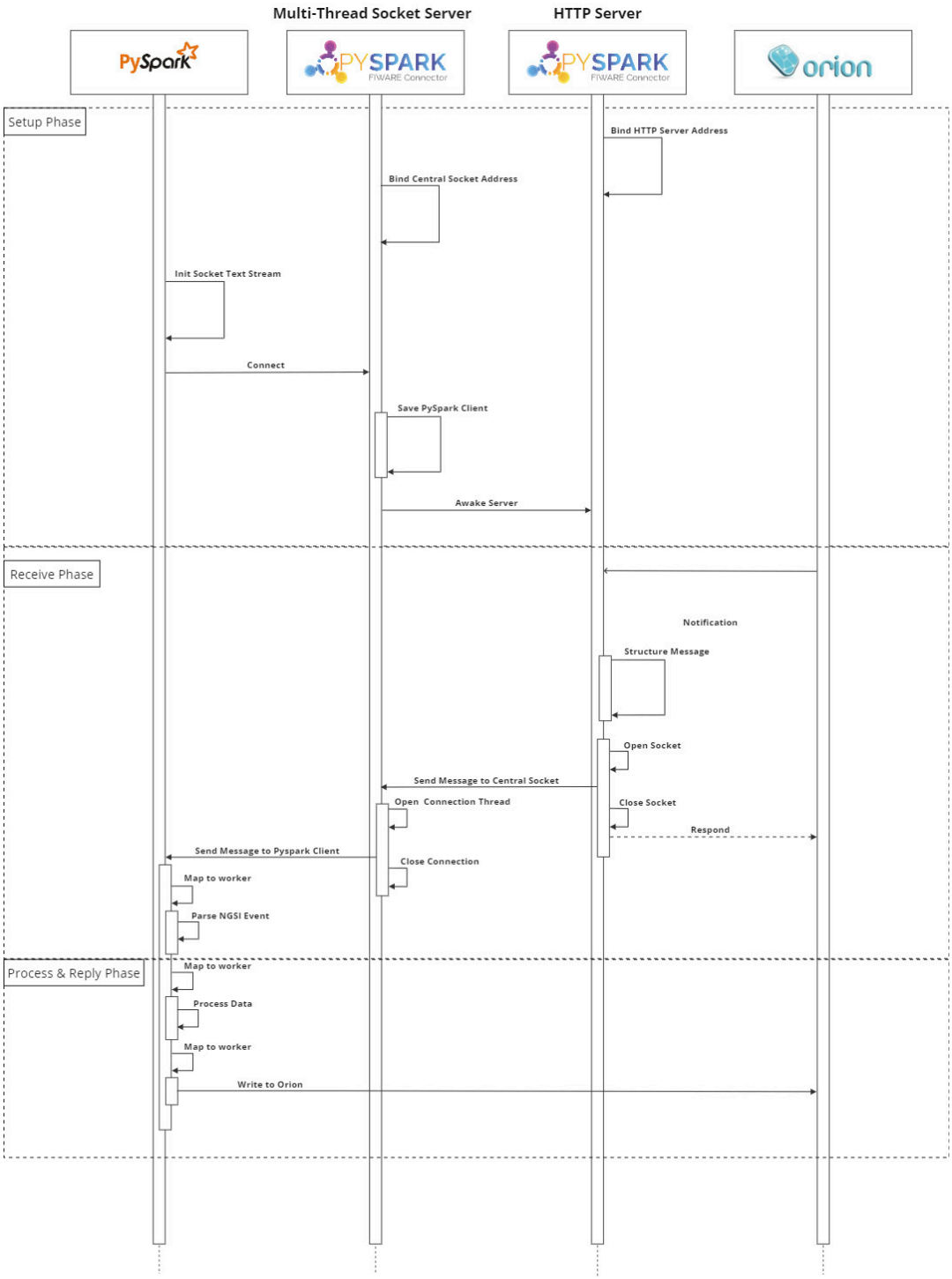

Figure 4 - PySpark Connector information flow

The second phase is for **data reception**. Assuming that the connector's HTTP Server is subscribed to the FIWARE Context Broker for a particular entity, when an attribute of the subscribed entity changes, FIWARE Context Broker sends a notification to the connector's HTTP Server. During this phase, the HTTP server organizes the incoming HTTP packet into a string message and opens a random socket, sending the incoming message to the central socket of MTSS. This socket receives the message and opens a new thread to manage this connection, passing the message to PySpark's SocketTextStream. Subsequently, PySpark maps the incoming message to a worker machine where the string message is parsed into an NGSIEvent object. The NGSIEvent is now available as RDD.

The third phase is dedicated to **data processing**. In this phase, the Spark driver maps the RDD containing NGSIEvent objects to a worker. After this, a map function utilises a custom processing function, taking an RDD as input and returning the custom function's output as "mapped" RDD. The result is then returned as an RDD.

The fourth phase oversees **data write-back**. The Spark driver executes the forEachRDD function, which then passes each RDD to the forEachPartition function, thereby mapping the final result to a worker. The forEachRDD function is invoked on RDD synchronisation, while forEachPartition allows the setup of connector parameters only once, after which it iterates on incoming RDDs. The forEachPartition requires a callback function that uses an iterator as an argument. In the end, the Spark Driver sinks the data flux, mapping an output function to the worker, and the connector sends a POST/PATCH/PUT request to the Orion Context Broker, displaying Orion's response.

In summary, **the FIWARE PySpark connector runs a multi-thread socket server that enables the listening phase, facilitated by its subscription to the FIWARE Context Broker**. It receives data and forwards it to Apache PySpark via a SocketTextStream. Once Apache PySpark completes the processing phase and returns the result to the FIWARE PySpark Connector, it can structure the incoming information in NGSIv2/LD format and invoke the Orion Context Broker API to update the context information.

Benefits & Impact

Although the PySpark Connector may be seen as a single component facilitating bi-directional interaction between Apache PySpark and the FIWARE Context Broker, it also ensures real-time monitoring of the Cognitive Solutions integrated into the Cognitive Automation Platform.

In a broader perspective, the PySpark Connector enhances the robustness of the tablet production process through dedicated control loops integrated into cognitive algorithms that rely on data-driven and hybrid parametrized process models. Furthermore, it maintains consistent product quality by continually checking and monitoring results obtained during the various stages of the production process.

Finally, the deployment of the PySpark connector within the Cognitive Automation Platform has a significant impact on reducing labour efforts, as it now supports or replace offline product testing (analytics) with automated processes in the Cognitive Automation Platform.

Added value through FIWARE

The Cognitive Automation Platform, with its FIWARE incubated PySpark Connector, is a modular and Open-Source framework capable of deploying cognitive functions from the edge to the cloud.

It is built on FIWARE's best-in-class European Open Source Community and complemented with APACHE's worldwide OSS market leadership.

Data Interoperability is achieved by adopting the NGSIv2/LD Data Model as a common data format in the industrial sector, ensuring compatibility with several technological components and facilitating easy data exchange within the CAP platform.

This interoperability, made possible through the use of FIWARE Open Source components, represents an element of CAPRI's replicability in the process industry. It enables the management of cognitive tasks, as well as data collection, storage, processing, and presentation directly from the plant.

Next steps

The next steps in evolving the FIWARE PySpark Connector represent a significant expansion of its capabilities, transforming it from a Single Input, Single Output (SISO) component into a Multi-Input, Multi-Output (MIMO) connector. This change will enhance its scalability and versatility across a wide range of use cases.

The connector will be upgraded to support multiple PySpark jobs simultaneously. This means it can handle multiple data streams and processing tasks concurrently, increasing its efficiency.

Additionally, the connector will incorporate support for the NGSI-LD temporal API, enabling it to work with time-series data and be compliant with components such as FIWARE Mintaka[10]. This capability is crucial for applications that require historical data analysis and trend identification.

Finally, it will be enabled for integration with popular messaging brokers like Apache Kafka[11] and/or ActiveMQ[12]. This will allow seamless data exchange with a

[10] Mintaka is an implementation of the NGSI-LD temporal retrieval api. It relies on the Orion-LD Context Broker to provide the underlying database.
[11] Apache Kafka is a distributed event store and stream-processing platform. It is an open-source system developed by the Apache Software Foundation written in Java and Scala. The project aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds.
[12] Apache ActiveMQ is an open source message broker written in Java together with a full Java Message Service client. It provides "Enterprise Features" which in this case means fostering the communication from more than one client or server.

variety of systems, making it an integral part of larger, distributed data processing architectures.

Security will also be a top priority, with the integration of components like FIWARE Keyrock[13] and/or FIWARE Wilma[14] for secure connections. This will ensure that data exchanges are protected, and only authorised entities can access the connector.

To conclude, the evolution of the FIWARE PySpark Connector includes further stress testing and extensive integration tests across several production systems. These activities are intended to demonstrate the connector's efficiency in real-world IoT applications, leading to a substantial enhancement of its capabilities.

This, in turn, will contribute significantly to the end of the incubation and its full inclusion in the FIWARE Catalogue.

References

- This work has been supported by the CAPRI Project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 870062.
- More information on the FIWARE PySpark Connector
- More information about the Cognitive Automation Platform on GitHub

[13] Keyrock is the FIWARE component responsible for Identity Management. Using Keyrock (in conjunction with other security components such as PEP Proxy and Authzforce) enables you to add OAuth2-based authentication and authorization security to your services and applications.This project is part of FIWARE. For more information check the FIWARE Catalogue entry for Security.

[14] Wilma is a PEP Proxy - it can be combined with other security components such as Keyrock and Authzforce to enforce access control to your backend applications. This means that only permitted users will be able to access your Generic Enablers or REST services. Identity Management allows you to manage specific permissions and policies to resources allowing different access levels for your users.This project is part of FIWARE. For more information check the FIWARE Catalogue entry for Security.

Author & Contributors

Mattia Giuseppe Marzano, *Software Development Specialist,* mattiagiuseppe.marzano@eng.it

Gabriele De Luca, *Technical Manager,* gabriele.deluca@eng.it

Marta Calderaro, *Project Manager,* marta.calderaro@eng.it

Angelo Marguglio, *Head of Digital Industry R&I Unit,* angelo.marguglio@eng.it

Engineering – www.eng.it

Categories

Domains (s)	Smart Industry, Smart Manufacturing, Process Industry
User (s)	Business
Key words	Automation Platform, Cognitive Solutions, Process Industry

Contact us

Having any questions? Want to contribute with another Impact Story?
Please contact **Tonia Sapia** @ tonia.sapia@fiware.org

Want to see more Impact Stories?
Please visit www.fiware.org/about-us/impact-stories/

FIWARE
Open APIs for Open Minds

FIWARE IMPACT STORIES

SMART MANUFACTURING

CAPRI Platform & the PySpark Connector: a bridge between FIWARE and Apache PySpark

FIWAREMarketplace

Be certified and featured in the FIWARE Marketplace.

GO TO THE MARKETPLACE

Never miss an update or a new Impact Story. Join our Newsletter!

SUBSCRIBE

Find Us On

Twitter Facebook LinkedIn YouTube Github

October 18, 2023 @ FIWARE Foundation, e.V. – www.fiware.org