1

vf-OS IO Toolkit

Victor Anaya [1], Nejib Moalla [2], Ludo Stellingwerff [3], José Luis Flores [4] and Francisco Fraile [1]

[1] Universitat Politecnica de Valencia, 46980 Valencia, Spain
[2] Universite Lumiere Lyon 2, DISP Laboratory, Bron, France
[3] Almende BV, Stationsplein 45 unit D1. 116, 3013AK Rotterdam
[4] IK4-Ikerlan Technology Research Centre, 20500 Gipuzkoa, Spain
vicanfon@upvnet.upv.es

Abstract. The growing variety of programming languages in software engineering feeds the permanent need of interoperability enablers between IT solutions. In agile and integrated software development, it is vital to ensure the communication between exiting IT system and new ones. In addition, the connected smart factories paradigm extends the perimeter of enterprise information system to cover the direct integration of sensors and machines. vf-OS is an H2020 European project aimed at providing a development environment with an integrated, virtual view on manufacturing processes. Within vf-OS, the IO Toolkit provides tools for on-premises facilitating the access to manufacturing devices and existing business software, while also facilitating the interaction with the rest of the vf-OS infrastructure. As such it allows developers to build secure and controlled gateways into the factories, as part of the capability set for their applications.

1.1. Introduction

The World is facing the fourth industrial revolution based on ICT, specifically architectures and services, as key innovation drivers for manufacturing companies. Factories are complex systems of systems and there is a need to develop a platform on which future manufacturing applications can be built.

vf-OS, virtual factory Open Operating System, is an Open Operating System for Virtual Factories composed of a kernel, application programming interface, and middleware specifically designed for the factory of the future. The purpose is to attract talent from solution developers and to provide accessible manufacturing smart applications to European SMEs.

Enterprise interoperability [JAK 12] remains the common research umbrella for software-oriented initiatives aiming to provide development frameworks and toolkits. To ensure engineering efficiency, structured interoperability methodologies are requested to improve the sustainability of software-oriented initiatives and solutions.

1.2. vf-OS IO Toolkit

The IO Toolkit is the development framework provided by vf-OS to Manufacturing and Service Providers wanting to generate new device drivers and API Connectors with the purpose to widen the range of shop floor devices and line of business systems that can be connected to applications developed for the vf-OS platform (called vApps).

Any new developed device driver or API connector (generically called IO Component) can be published to the Marketplace (see top-right part of Fig 1).

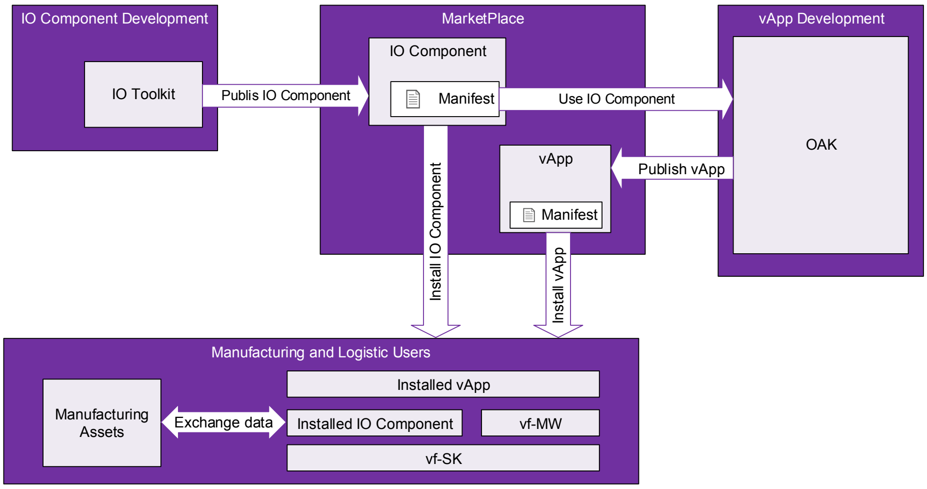

Fig 1 *IO Toolkit Positioning*

If a vApp uses a specific IO Component, then the Manifest file of the vApp describes this dependency, so that users can be informed before installing the vApp.

Software Developers can thus develop functionalities that interact with any manufacturing asset that is connected to vf-OS via an IO Component.

1.3. IO Components

An IO component is a generic term to refer to Device Drivers or API connectors. A Device Drivers a kind of IO component that interacts with physical industrial devices to integrate sensors' data and send commands to devices. An example is a driver for consuming data from devices supporting the OPC UA protocol. In the other hand, API Connectors are IO components that interact with legacy software in factories (such as ERPs, databases or tailored ICT solutions).

The common structure followed by every IO component is shown in the next layered schema:

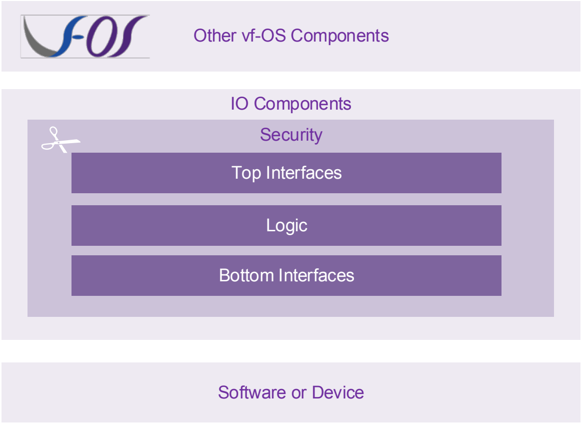

Fig 2. *I/O components functionalities*

1.3.1. Top Interfaces

Top interfaces are the interfaces that any IO component must implement to be vf-OS compliant. Those interfaces, bootstrapped by the IO toolkit provide mechanism to invoke and consume data and functionality provided by drivers and APIS. The different interfaces are:

– REST API Server Composition: providing specific REST APIs that new drivers and API connectors must follow to be called by the vf-OS platform components.

– Messaging endpoints: providing drivers and connectors ways to interact with the vf-OS Messaging component when sending and receiving new messages.

– Pub/Sub endpoints: providing ways to interact with the vf-OS Pub/Sub component, providing triggers, events and subscription functionalities.

– Metadata template composition: to generate a metadata template file used by vApp developers.

– Registration endpoints: providing functionality to drivers and API connectors to be registered themselves in the Platform and the Marketplace.

– Logging endpoints: to register logs that will be consume by the System Dashboard and the Marketplace.

1.3.2. Logic

The IO Toolkit will provide a set of functionalities available when developing the internal logic of a driver or API connector. Every new driver and component will have available a set of libraries when developing its internal logic. Those libraries are functionally aggregated as:

– Internal Storage: to connect and query the internal databases used to store configuration and operational data, both relational and non-relational databases.

– Key-value store: to access key-value stores to manage associative key-value pairs.

– Edge computing: to process sensor data as close to the data source as possible, instead of processing raw sensor data in the cloud.

– API Access Control: for authentication and security, which can be used alone or in combination to issue credentials and control access.

– API Lifecycle management: The IO Toolkit must provide API lifecycle management tools (prototype, publish, deprecate, and block).

1.3.3. Bottom Interfaces

Bottom interfaces are the proprietary interfaces part of well-known communication standards used by devices and business applications in manufacturing companies.

The IO Toolkit will provide out of the box the bottom interfaces (Table 1).

Table 1. Drivers and API connectors to be supported by vf-OS platform (extract)

Protocol	Description
OPC UA	OPC UA is a machine to machine communication protocol for industrial automation developed by the OPC Foundation.
MQTT	ISO standard (ISO/IEC PRF 20922) publish-subscribe-based "lightweight" messaging protocol for use on top of the TCP/IP protocol.
Modbus	Ethernet serial communication protocol extensively used in SCADA .
GPIO	Generic hardware interface integrated in single board computers (eg Raspberry Pi) widely used in fast prototyping.
MTConnect	Protocol to exchange data between industrial network components.
MS Excel	Spreadsheets are widely used in SMEs to build simple data analysis and reporting applications.
STEP	STEP a file format used in CAD to work with 3D models. The STEP file format is widely used in the design of mechanical parts and is supported by the most popular CAD Software tools.
ODBC	Standard API to access database management systems.
OData	REST protocol to build API Connectors with leading business software applications and systems (eg IBM WebSphere, Microsoft Dynamics CRM, Acumatica ERP, SAP ERP).
SAP ERP	OData-based API Connector
NAV ERP	Microsoft Dynamics NAV is an ERP system developed by Microsoft

1.3.4. IO Toolkit Security

The IO Toolkit Security incorporates a mapped set of security requirements and a documented unit testing plan in security for the development of new vf-OS IO Components. This set of security requirements are based on the recommendations of the most relevant and authoritative organizations in the field of security such as SysAdmin Audit, Networking and Security Institute (SANS), OpenWeb Application Security Project, or National Institute of Standards and Technology.

Based on these recommendations, the requirements are created and provided as a set of checklists. These will require concrete specifications of aspects such as the use of the corresponding libraries for implementing OpenID [OPE 17] for fulfilling the holistic authentication and authorization model of vf-OS (XACML [OAS 17]) or the specification of valid time periods of sessions, error management, or memory management policies.

Finally, the IO Toolkit Security will provide endpoints for the vf-OS Security Command Centre REST APIs (Identity Service, Policy Administration, and Continuous Security Monitoring).

1.4. Containerization

The vf-OS platform technical infrastructure [VFO 17] is based on microservices, and Docker [DOC 17] is the core technology used with that purpose.

Developed drivers and API connectors will be deployed by the vf-OS marketplace as Docker containers. The benefits that Docker provides the IO Toolkit Execution Services are a Standardised API for connecting various services, the possibility to port applications to different environments and adapt to several deployment patterns, and several security features such as signed containers, through the Docker Trust feature.

1.5. Conclusions

The vf-OS IO Toolkit is the core component in the vf-OS platform providing the functionality to consume and interact with physical device drivers and existing business software solution. The IO Toolkit will provide a containerisation solution to run drivers and API connects as microservices in the context of the vf-OS technical architecture. The driver could be run in a specific webserver on a docker machine providing very flexible deployment alternatives. The IO toolkit will provide a catalog of driver and connector templates that will empower developers on the development of new Device drivers and API connectors on the vf-OS Platform. The IO Toolkit will provide a convenient development documentation, along with a standard versioning scheme for the Toolkit and the produced components. Finally, vf-OS requires new drivers and connectors to be properly validated. With that purpose, the IO Toolkit will provide a testing framework to check the compliance of the IO component development.

1.6. References

[JAK 12] P. JAKOLE, P., *An integrated approach to software engineering*, Springer Science & Business Media, 2012.

[OPE 17] OPENID., *OpenID Specifications* http://openid.net/developers/specs/. [Accessed 30 10 2017].

[OAS 17] OASIS., *eXtensible Access Control Markup Language (XACML) Version 3.0* http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html. [Accessed 30 10 2017].

[VFO 17] vf-OS consortium, *D11.5: Technical Set Up and Quality Toolset v1.2*, 2017.

[DOC 17] Docker Inc., *Docker* https://www.docker.com/. [Accessed 30 10 2017].