Henri Pettinen, Marko Elo

Utilizing multifunctional display computer as a local gateway in industrial IoT use cases

Abstract: The Industry 4.0 paradigm emphasizes the increased level of automation and data exchange in manufacturing technologies. Hence, industrial companies have to deal with vast amounts of data gathered from the production. Usually it is not practical to send all the measured data outside the local premises for further processing. Onboard processing, temporary storing, and appropriate forwarding of the data are key operations to be handled by an industrial gateway.

CrossControl produces computers for industrial use, with integrated displays. Most common use case for such display computers are in heavy industrial vehicles, with connectivity to the onboard vehicle control bus. By utilizing a dedicated connectivity module, the onboard display can have a wireless connection as well, for on-site device and backoffice connectivity.

This conference paper for Automaatiopäivät23 introduces key parts of the Productive4.0 project which addresses the general theme of Industry 4.0. The paper addresses the use cases of machine and fleet management offered as an industrial Software as a Service (SaaS). The pilot implementation of CrossControl's gateway device is presented with the focus being on the software stack, information flow architecture and the graphical user interface. Moreover, applying Service Oriented Architecture (SOA) on gateway and edge level is also discussed.

Keywords: Industry 4.0, Internet of Things, IoT, Productive4.0, Multifunctional display, Arrowhead framework, Graphical user interface, Gateway, SaaS, SOA, CPS.

***Corresponding Author: Henri Pettinen:** CrossControl Oy, E-mail: henri.pettinen@crosscontrol.com

Second Author: Marko Elo: CrossControl Oy, E-mail: marko.elo@crosscontrol.com

1 Introduction

In a continuously digitizing world, industrial machines have reached the point where they are capable of harvesting data and efficiently exchanging it with other systems over wireless media. This brings a vast number of benefits, one being, for example the improvements in the overall situational awareness at the work site. The very same trend is prevailing in every business sector and everyday life. Internet of Things (IoT) is the umbrella concept for the physical objects being able to sense their surroundings and then sharing that information with peer systems and other surrounding infrastructure.

In industrial domain, the collected data provides valuable assets for multiple stakeholders, including but not limiting to the end users, system integrators, maintenance service providers, OEMs, and even insurance brokers. However, the data is usually valuable for hostile parties as well, whose interest is to attack system vulnerabilities and engage industrial espionage or cause disruption in general. The characteristics of Internet-based connections force companies to apply hardening approaches to their digitized production environments, setting tougher security requirements also to gateway and edge devices in IoT concepts.

Productive4.0 is a European-wide innovation project involving more than 100 partners from 19 European countries. As an ARTEMIS Innovation Pilot Program (AIPP), it aims for providing industry-driven digitalization and internet connectivity use cases for multiple industrial domains, with proof of concept implementations conducted by project partners organized as co-creation teams.

CrossControl Oy is specialized in control and information system solutions for industrial vehicles operating in demanding environments. The company is involved in Productive4.0 project, with one of the main contributions in a use case focusing on offering machine and fleet management as an industrial service.

This paper is partitioned as follows: Chapter 2 introduces CrossControl product platform supporting the scope of IoT concepts. Chapter 3 focuses on describing the key principles behind the Arrowhead framework. Chapter 4 is dedicated for enlightening two Productive4.0 use cases, and Chapter 5 describes downstream details by CrossControl's implementation

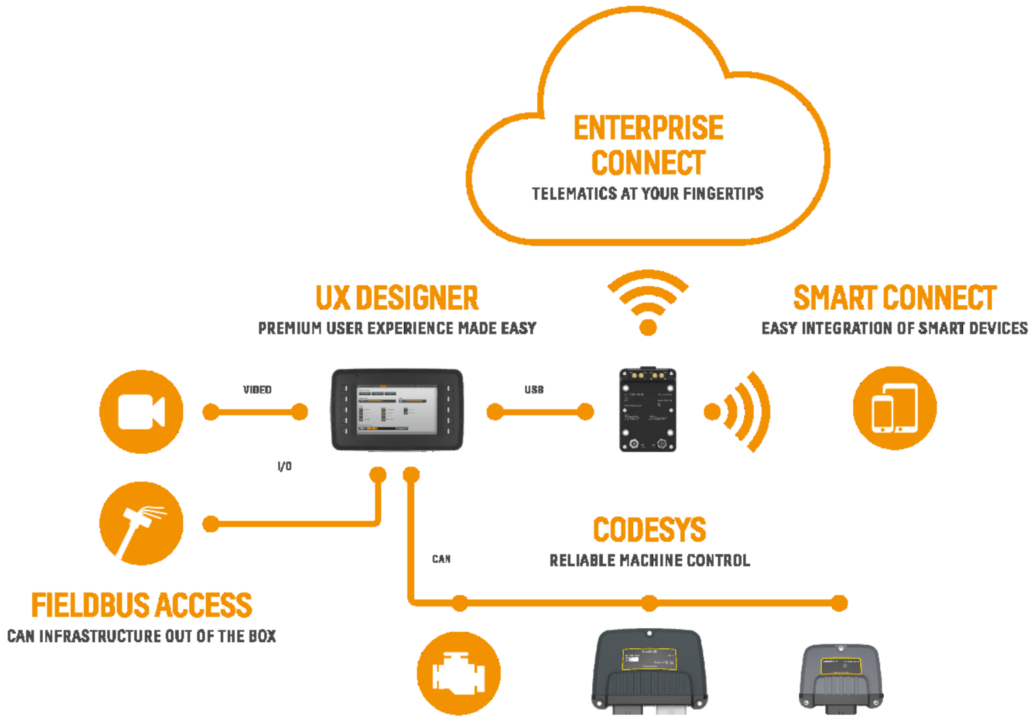
ENTERPRISE CONNECT
TELEMATICS AT YOUR FINGERTIPS
UX DESIGNER
PREMIUM USER EXPERIENCE MADE EASY
SMART CONNECT
EASY INTEGRATION OF SMART DEVICES
VIDEO
USB
I/O
CAN
CODESYS
RELIABLE MACHINE CONTROL
FIELDBUS ACCESS
CAN INFRASTRUCTURE OUT OF THE BOX

Fig. 1. LinX application platform comprises various premade software and hardware components.

in the use case context. Chapter 6 outlines the upcoming steps of the use case, along with a few development ideas. Chapter 7 concludes the paper.

2 CrossControl platform

This chapter introduces the relevant parts of CrossControl product platform in an IoT-featured application context, with specific match to requirements driven by Productive4.0.

2.1 Overview

CrossControl's CCpilot display product line is capable of fulfilling divergent functions needed in the industrial field. They can be used as an onboard instrumentation display, human machine interface (HMI), video monitor, electronic manual, or as an operator's connected task manager, for instance.

The software development kit (SDK) and runtime library for CrossControl displays, called as LinX, provides the tools and protocols for applications operating in onboard data collection, monitoring, visualization, logging, and exchange with the cloud. Functionality provided by LinX is illustrated in figure 1. The SDK comes with virtual machines for each different target in the CCpilot product line, facilitating and accelerating application development and deployment. The SDK contains all the display specific environment variables and configurations, enabling developers to cross-compile software applications for the display.

2.2 CCpilot VS

CCpilot display, model name VS, is involved in the targeted Productive4.0 use case. The VS is based on an ARM Quad Cortex-A9 CPU and 12" TFT multi-touch display, running a custom-made Linux distribution as its operating system. It has four CAN ports supporting ISO 11898 CAN 2.0B protocol, an Ethernet interface

Fig. 2 CCpilot VS display computer.

supporting 10BASE-T/100-BASE-TX/1000-BASE-T connection and Auto-MDIX, total of four USB ports, and a light sensor.

2.3 CrossLink AI

CrossLink AI is a communication module, connecting to CCpilot displays via USB. It supports a selection of wireless technologies enabling connectivity with on-site peers and enterprise systems. The add-on module provides WLAN access point, Bluetooth, 3G and GPS functionality for the displays.

3 Arrowhead architecture

Arrowhead framework is an academic approach for a generic and distributed framework for IoT automation applications. The first version was developed in ARTEMIS project Arrowhead, with further evolution taking place in projects EMC2 and FAR-EDGE. The Arrowhead framework is being utilized e.g. in projects Opti, MANTIS, and Productive4.0.

The framework aims to increase the interoperability of various systems and enhance the use of Service Oriented Architecture (SOA) in industrial domain, with Industry4.0-based thinking in mind. In its terminology every internet connected object is abstracted to a **service**. Typical for SOA is that different single functionality providing services can be utilized together in order to achieve functionality of a large software application [1]. Arrowhead framework follows this approach in several ways, e.g. by providing service orchestrator component. The orchestrator dispatches service requests to find the best suitable services to be invoked for a specific query or function call. The service orchestrator and other core components of Arrowhead framework are discussed in more detail later in this paper.

The way Arrowhead increases the level of automation is based on the idea of local clouds, that is, a whole cloud infrastructure on enterprise’s own local network. All physical components must be inside a closed industrial network in order to achieve all benefits of the Arrowhead framework. The framework also strives to improve real-time data handling, with data and system security, and with built-in methods for efficient scalability. The local clouds can communicate through defined end points, forming larger clouds, and networks of clouds. These end points create interfaces for every local cloud, establishing a controlled access point for external actors.

Arrowhead core services are the mandatory components in every local cloud infrastructure that must be up and available. This comprises of three components in total: Service registry, Orchestrator and Authorization system. The Service registry stores all new and available services inside the local cloud and information about them. The Orchestrator utilizes this registry and its information to inform the services in local cloud that are willing to consume other services. In other words, the Orchestrator knows all available services and where they reside. By requesting a specific service from the Orchestrator, it returns the direct address to the exploitable service. The Authorization system verifies that a given consumer is allowed to access the addressed service. [2] These core services are required in order to deploy the minimum Arrowhead compliance in the local cloud. More information of further services, including Plant description, Configuration, System registry, Device registry, Event handler, QoS manager, Historian and Gatekeeper, can be found from the Arrowhead documentation.

Arrowhead provides flexibility to add new machines on the work site as they can be automatically registered as Arrowhead services when they first time connect to the local cloud’s Service registry. After that, the services provided by the new machine can be discovered by other registered services inside the local cloud or even outside it, if made accessible.

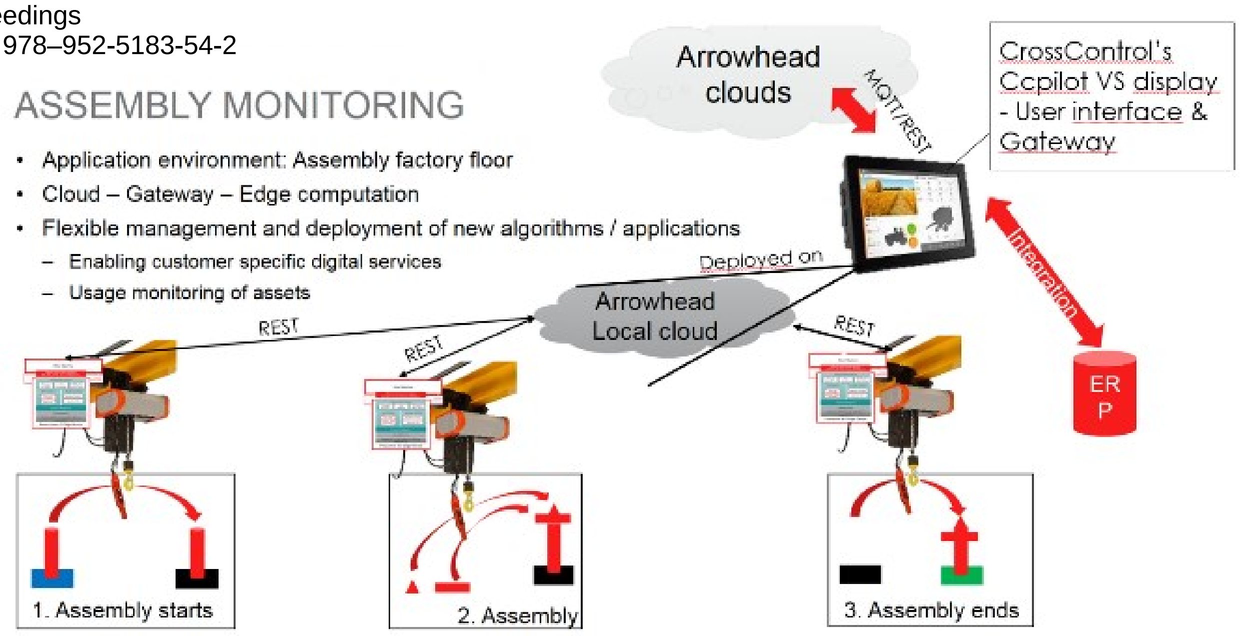
ASSEMBLY MONITORING
Application environment: Assembly factory floor
Cloud – Gateway – Edge computation
Flexible management and deployment of new algorithms / applications
Enabling customer specific digital services
Usage monitoring of assets
Arrowhead clouds
MQTT/REST
CrossControl's Ccpilot VS display - User interface & Gateway
Deployed on
Arrowhead Local cloud
Integration
REST
REST
REST
ER P
1. Assembly starts
2. Assembly
3. Assembly ends

Fig. 3. Demonstration of the lifting business use case

4 Use case descriptions

This chapter introduces two Productive4.0 use cases, involving piloting of the Arrowhead framework in a fleet management context, deployed on CrossControl system platform.

4.1 Lifting business use case

In various assembly tasks there are lifting machines involved to help the process. Assembly, like any other part of manufacturing, requires control and monitoring to improve the process. Further added value can be created by harnessing the collected data to offer customer-specific, differentiated digital services. Such special requirements may cover e.g. field data forwarding to ERP or other backoffice systems, or specific functional safety triggers for onboard systems to react to predefined irregular data values.

This use case centralizes to a chain hoist, and the hoist OEM manufacturer's interest is to develop a method to track the movement of the chain hoist and then recognize the different work phases of the ongoing assembly process. Assembly process is divided into three phases which are the start of the assembly, the assembly itself, and the end of the assembly. These phases can be identified when the load and the vertical movement of the chain hoist are monitored. Figure 3 depicts this assembly monitoring process and clarifies the display computer's role as a gateway in this use case.

The chain hoist has an IoT edge device attached to it. The edge device retrieves sensor data from the hoist and forwards the data to the gateway, within the local network. It takes its power supply straight from the hoist and it can send the measured values to multiple subscribers. The data is conveyed in the JSON message format. Local Arrowhead cloud can be established on the work site and its core services can be ran on the CCpilot VS or some other computer on site dedicated for the gateway role.

The gateway dispatches a defined set of the collected data to the manufacturing company's backoffice systems and provides selected information also for other Arrowhead clouds. The communication is done using HTTP and MQTT protocols, usually over wireless connections.

4.2 Rock crushing use case

Another use case is associated with vibrating screens which are machines primarily utilized in the mineral processing industry for separating material according to size. Because of their high frequency vibration, the screens have to withstand high accelerations. These machines are relatively prone to mechanical part wear and tear, which requires well-planned maintenance activities to avoid downtime and loss of efficiency in operation.

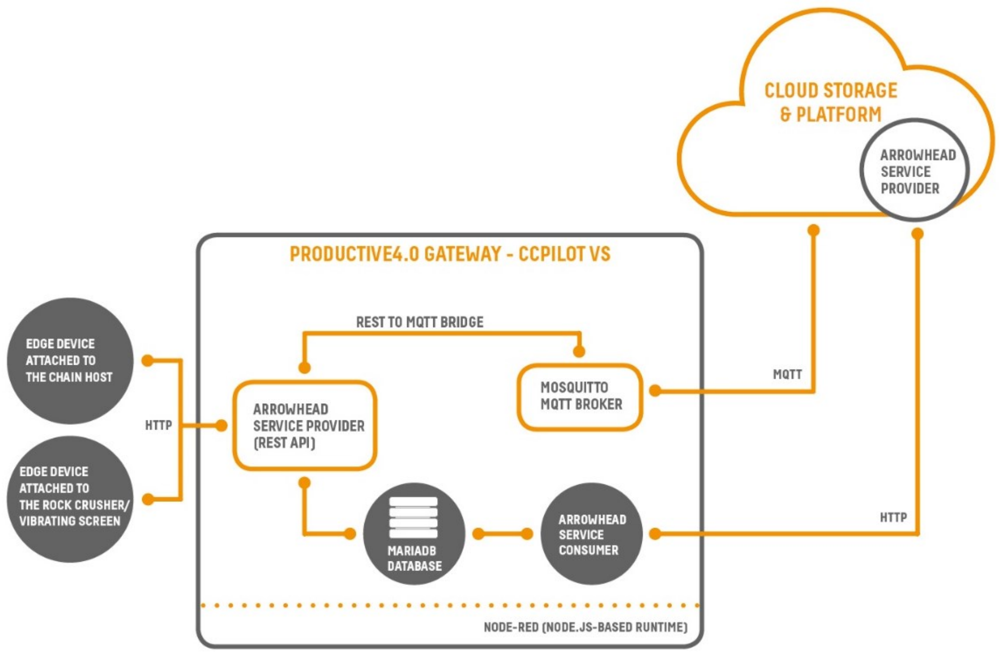
CLOUD STORAGE & PLATFORM
ARROWHEAD SERVICE PROVIDER
PRODUCTIVE4.0 GATEWAY - CCPILOT VS
REST TO MQTT BRIDGE
EDGE DEVICE ATTACHED TO THE CHAIN HOST
EDGE DEVICE ATTACHED TO THE ROCK CRUSHER/ VIBRATING SCREEN
HTTP
ARROWHEAD SERVICE PROVIDER (REST API)
MOSQUITTO MQTT BROKER
MQTT
MARIADB DATABASE
ARROWHEAD SERVICE CONSUMER
HTTP
NODE-RED (NODE.JS-BASED RUNTIME)

Fig. 4. Data flow in Productive4.0 use cases and gateway device's software and technology stack.

Predictive maintenance practices require data from condition monitoring measurements, including screen movement and acceleration levels. Further, the bearings can also be monitored for excessive noise and heat, therefore requiring multiple types of sensors. Analysis services for the CBM (Condition Based Maintenance) logic can be run at the gateway and accommodated into the Arrowhead framework, enabling also the distribution of warning and alarm messages in the regular communication infrastructure of the pilot.

The data gathered from the condition monitoring shall be analyzed further so that accurate maintenance plans can be generated and the lifetime of the mechanical parts in different environments can be predicted in advance. The optimal solution for data analysis is not always achieved by putting all the computing and data storing to the cloud level. The balance between local processing and cloud computing shall be optimized, leaving the possibility to make adjustments later on when more knowledge of the process is gained.

Further requirements of this use case are being specified, but the general architecture follows the guidelines of the chain hoist use case. The gateway computer, instantiated by CCpilot VS, has a similar role for local data processing, HMI and network connectivity. The display would be probably mounted to the vibrating screen itself, hardening the requirements for its environmental ruggedness.

5 Implementation

This chapter introduces the current state of the implementation for the Productive4.0 lifting business use case.

4.1 Software stack

The outline of the data flow and the software components deployed for CCpilot VS in both Productive4.0 use cases is simplified in figure 4, enabling the gateway functionality of the targeted IoT concept.

The data flow logic is made utilizing a programming tool called Node-RED. It uses a Node.js-based runtime and is run on a web browser. With its interactive user interface consisting of node blocks and wires between them, it enables the design and deployment of the logical data flow subsystem. [3] This solution also supports further customization of the data flow by embedding custom JavaScript code blocks in the process. Node-RED applications running on the VS can exploit Python code blocks as well, through the Node-RED libraries and Python3 interpreter installed on the VS. The core purpose for Node-RED is to convert the messages coming from the edge devices from REST to MQTT. This conversion and transfer component is called as a bridge in figure 4. REST (Representational State Transfer) is a software architectural model for standardizing web services. MQTT (Message Queuing

Telemetry Transport) is a lightweight application layer publish/subscribe protocol laying on top of the TCP/IP protocol stack, with built-in username and password-based authentication.

Mosquitto MQTT broker serves the MQTT clients and conveys their MQTT messages [4]. The MQTT messages can be seen and subscribed to by any actor with access to the MQTT broker. The gateway device is usually required to store data to a local database. MariaDB is a light, open-source SQL-based database, well suited for piloting and testing in an environment of embedded devices typically having limited memory storages [5].

Java runtime environment is also installed on the display in order to develop and deploy the Arrowhead core services and custom Arrowhead-compliant services inside the Java virtual machine. Those services are exploitable also by other local cloud services, or conversely VS can exploit other available Arrowhead services.

4.2 Graphical user interface

The gateway computer is also required to operate as an onboard HMI, with a graphical user interface. The display illustrates select raw measurement signals coming from the sensors, and it can be enabled to acts as a HMI control board. For example, the direction of the movement of the chain hoist can be visualized, including the current load it is lifting. The current version of the UI made for this use case is depicted in figure 5.

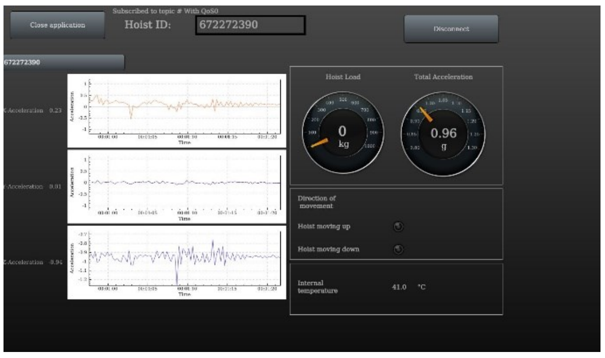

Fig. 5. Simple UI for visualizing the measurement data.

CrossControl SDK builds on Qt, with industrially pre-designed graphical widgets for rapid application development. As an open standard, Qt supports cross-platform application development and overall reusability of code.

6 Discussion

The finalization of Arrowhead services at the gateway is in progress, for full implementation of the functionality depicted in figure 4. Data storing as a service will enable the field machines to get a flexible, temporary data storage for the retrieved sensor data before it is conveyed to the cloud. The service consumers will find the storing service from within the local cloud, and in case there would be multiple displays offering the same service, service discovery and load balancing can be applied across multiple gateways. A specific service will be created also for implementing computing for local analytics.

The chain hoist use case enables further data analysis performed at the gateway. For example, specific tasks performed by the chain hoist operator can be recognized and analyzed, and integrated with other available context data, increasing the situational awareness. General task progress can be shown on the display and synchronized with workflow management or ERP in the cloud.

Finally, the security measures need to be considered. As for the time being in this pilot, the data flowing through the gateway is not subject to strong encryption. Industrialization of the concept potentially requires a X.509 certificate based encryption solution, or equivalent.

7 Conclusion

This paper presents an IoT concept architecture, as piloted in Productive4.0 use cases in the scope of fleet management offered as a service. As one of the key components in the use cases, the role of the gateway computer has been discussed in more detail, with requirements for gathering data from various edge devices, data processing and analysis, and dispatching of the data to the cloud and potentially other connected systems.

The use cases also demonstrate CrossControl's industrial display computer running as a gateway and as an onboard HMI in the pilot environment, visualizing select telemetry and dashboard state values of the monitored machine.

8 References

[1] Velte A. T., Cloud Computing: A Practical Approach, New York: McGraw Hill, 2010.

[2] Delsing J., IoT Automation: Arrowhead Framework, Boca Raton: CRC Press, 2017.

[3] Node-RED, March 2019, Available: https://nodered.org/.

[4] Light R. A., Mosquitto: server and client implementation of the MQTT protocol," The Journal of Open Source Software, vol. 2, 2017.

[5] MariaDB, March 2019, Available: https://mariadb.org/.