Report on Emerging IoT Platforms for the Industrial Internet

Project Acronym	IoT4Industry
Document-Id	R.1
File name	
Version	FINAL document
Date	Start: 01 October 2014 End: 31 November 2014
Author(s)	Oliver Jung
QA Process	Violeta Damjanovic-Behrendt

Table of contents

1. Motivation

Note that this document is an addition to the project Deliverables. Here we summarize IoT platforms with the potential to support the Industrial Internet applications (Industry 4.0 vision). Within the past years, new platforms, tools and operating systems addressing the IoT have constantly emerged. Some of these approaches are rather generic and can be adjusted to different domains; others are quite specific and aim at various applications.

2. „All-in-one“ IoT cloud service providers

One major business case seems to be an „all-in-one“ IoT cloud service provider that takes care of collecting data, (cloud) storage, analysis, visualization, creating additional value and triggering events. The major drawbacks of these approaches are that they disable various security mechanisms and the degree of adaption to specific use cases is rather low. Furthermore, the users are losing the connection with their data, and do not know much about the underlying technology. Consequently, such approaches are not suitable for sensitive data and critical devices. Nonetheless, these service providers can provide a quick and easy access to IoT technology for experimental settings. Some major providers will be named in the course of this chapter.

2.1. GroveStreams [1]

GroveStreams offers APIs for collecting data and takes care of (cloud) storage, analysis, visualization, creating additional value and triggering events.

2.2. Exosite [2]

Pursues a similar approach to GroveStream.

2.3. SensorCloud [3]

Pursues a similar approach to GroveStream and Exosite.

2.4. Xively [4]

Pursues a similar approach to GroveStream, Exosite and SensorCloud.

2.5. SeeControl [5]

Pursues a similar approach to GroveStream, Exosite, SensorCloud and Xively. Furthermore, they offer a “no coding” solution with predefined drag-and-drop analysis elements.

2.6. ThingSpeak [6]

ThingSpeak pursues a similar approach to GroveSteam, Exosite, SensorCloud and Xively by offering APIs for collecting data and performing actions on it. Furthermore, the user can set up various data channels and complex analysis tasks due to MATLAB online support.

2.7. Wolfram Data Drop [7]

Wolfram offers an open source service for accumulating data in the Wolfram Cloud. The Wolfram Data Framework adds semantics to the data and makes it computable. Data can be added via APIs or directly from hardware ports. There are libraries for the Raspberry Pi. Arduino, mobile devices and embedded systems will be supported in the future, too. As an integrated part of the Wolfram Language, the Data Drop is available in all Wolfram platforms, such as the Device Analytics Platform, the Programming Cloud and Mathematica.

3. Other “all-in-one” solutions

Apart from the cloud service providers, there are other providers and platforms aiming for an “all-in-one” IoT solution. In contrast to the cloud services, these platforms consist of various modules for each functionality and can be deployed to different local instances in most cases. Nonetheless, the users are still dependent on their selection of tools and changes are hard to implement. Concerning the Industrial Internet vision, these solutions are more secure than the cloud services and provide easy to use interfaces.

This chapter deals with various “all-in-one” solutions for different application types.

3.1. Oracle Internet of Things [8]

Oracle offers software platforms for each relevant level of IoT technology. For example, “Oracle Java Embedded” delivers Java-enabled intelligent systems for any connected device of any size and in any market; the “Oracle Business Process Management Suite” helps to integrate IoT with enterprise IT; “Oracle Event Processing” enables real-time big data analysis.

3.2. ThingWorx IoT Platform [9]

The ThingWorx IoT platforms offers a lot of predefined functionality and interfaces that can be composed in a “drag and drop” manner with their “Mashup Builder” in order to create IoT applications quickly. It supports a variety of devices and platforms and is scalable.

4. Middleware solutions

Middleware solutions allow for the integration of IoT functionality into “traditional” back- and front-end platforms. These solutions serve various programming languages and are open source or at least highly configurable. This allows for adapting to existing environments and keeping data and data transmission secure and transparent. We have taken a look at some of the best known platforms for various cases of application and some that might become relevant in the future.

4.1. Node-RED [10]

Node-RED is a simple open source visual tool, developed under the Apache 2.0 license for wiring the IoT. Hardware devices, APIs and online services can be connected by selecting predefined components and setting their interaction in a browser-based flow editor. The provided functionality includes those supported by the most major protocols and platforms; for example, HTTP, MQTT, Email, Raspberry, MongoDB, MySQL, PostgreSQL, Sentiment Analysis, etc. The functionality can be extended by defining custom components or using the JavaScript editor.

4.2. Macchina.io [11]

macchina.io is an open source software toolkit under the Apache 2.0 license for building embedded IoT applications on Linux-based devices, such as the Raspberry Pi. It implements a web-enabled, modular and extensible JavaScript and C++ runtime environment and provides building blocks that enable easy communication to sensors, devices and (cloud) services.

Furthermore, it is based on industry proven components, such as the POCO C++ libraries and the V8 JavaScript engine. Due to its C++ environment it runs on embedded Linux devices with as little as 32MB RAM, as well as desktop Linux. Relevant protocols such as CoAP, HTTP(S), MQTT, Bluetooth, Modbus, RS-232, etc. are supported by default.

4.3. Kaa [12]

Kaa provides server and endpoint SDK components. The SDKs get embedded into the connected device and implement real-time bi-directional data exchange with the server. SDKs are available for C, C++, and Java. The C SDK requires 10kB RAM and 40kB ROM, the C++ and Java SDKs require more RAM and ROM since they target at more powerful platforms. Kaa is pre-integrated with major platforms, such as Linux, Android, Raspberry, iOS and Windows.

Kaa handles all communication including data consistency and security, device interoperability, and failure-proof connectivity. Furthermore, it supports virtually any hardware and communication layers, promotes structured data and is scalable without downtime by adding more nodes to the cluster. The back-end provides REST interfaces for server integration and is pre-integrated with major processing systems, such as mongoDB, Hadoop, Oracle DB and Spark.

4.4. IoTivity [13]

The IoTivity framework APIs are based on a resource-based, RESTful architecture model. The framework operates as middleware across several languages and operating systems and has four essential building blocks:

- **Discovery:** IoTivity discovery supports multiple discovery mechanisms for devices and resources in proximity and remotely.
- **Data transmission:** IoTivity data transmission supports information exchange and control based on a messaging and streaming model.
- **Data Management:** IoTivity data management supports the collection, storage and analysis of data from various resources.
- **Device management:** IoTivity device management supports configuration, provisioning and diagnostics of devices.

4.5. Ethereum [14]

Ethereum is a decentralized software platform for the IoT working on blockchains over multiple nodes. The development is at an early stage.

5. Hardware and operating systems

IoT specific operating systems and hardware enable Industrial Internet applications without the need to take care about each aspect of communication and security, since they are supported by default. Of course these hardware and operating system solutions are not suitable for any kind of device but they simplify the handling of suitable devices enormously. This chapter brings in several development hardware platforms and operating systems aiming at energy efficient IoT devices.

5.1. Tessel [15]

Tessel provides cheap hardware modules for IoT development with standardized Tessel ports in order to change applications quickly. Each Tessel board offers 2 Tessel module ports, 2 USB ports, WiFi, Ethernet and microUSB and delivers a 580MHz processor, 64MB DDR2 RAM and 32MB Flash. There are modules for Bluetooth low energy, GPS, RFID, servo control, and more.

Tessel runs JavaScript and supports NPM. Furthermore, there is language support for Python and Rust. The Tessel libraries hide away much of the low level programming in order to create short and functional code.

5.2. ARM mbed IoT Device Platform [16]

The mbed IoT Device Platform by ARM consists of tools and services for software, cloud development and communication, the “mbed OS” aiming at energy-efficiency, productivity, connectivity and security, and the “mbed Device Server” created for scalable solutions.

The “mbed OS” provides a C++ application framework and a component architecture that eliminates much of the low-level work normally associated with MCU code development. Key benefits are:

- Support for Bluetooth low energy, Zigbee, 6LoWPAN, etc.
- Automation of power management

- Support of a wide range of ARM Cortex-M based hardware
- Support for OMA Lightweight M2M protocol
- Support for all key open standards for connectivity and device management

5.3. Contiki [17]

Contiki is an open source operating system for the Internet of Things. Contiki connects tiny low-cost, low-power microcontrollers to the Internet by supporting relevant protocols, such as HTTP, 6LoWPAN, RPL and CoAP. Contiki applications are written in standard C. Furthermore, Contiki provides mechanisms for estimating the system power consumption and serves systems that may need to run for years on a pair of AA batteries.

5.4. TinyOS [18]

TinyOS is an open source operating system that aims at devices with extremely low capabilities requiring no more than 1kB RAM and 4kB ROM. It provides similar functionality like Contiki but uses its trimmed version of C, called nesC, that requires additional programming knowledge.

5.5. RIOT-OS [19]

RIOT-OS provides similar functionality like Contiki and TinyOS, and is now actively developed to support more and more platforms in order to gain more flexibility. Additional features in comparison to Contiki and TinyOS are C++ language support, multithreading, modularity and real-time capability.

6. Messaging systems and semantics

When talking about IoT, we usually deal with devices transferring a huge amount of data. Regardless of the middleware and other platforms that are part of the environment, this data has to be exchanged in a structured and efficient way in order to utilize all opportunities and process each data stream. In this chapter, we point out some suitable semantics and messaging systems.

6.1. IOTDB [20]

The Internet of Things Database founded by David Janes offers simple semantics for the IoT. The associated project “HomeStar” is based on “Node.JS” and provides an easy to start with management interface.

6.2. Apache Kafka [21]

Apache Kafka implements publish-subscribe messaging by setting up a single cluster as a central data backbone, which can be extended without downtime. Data streams are partitioned and spread over a cluster of machines to allow data streams larger than the capability of any single machine. Messages are replicated within the cluster to prevent data loss. Each broker can handle terabytes of messages without performance impact.

6.3. RTI Connext DDS [22]

The RTI Connext DDS software provides a messaging solution for intelligent machines and devices. The decentralized architecture provides low latency, high throughput and scalability, non-stop availability, plug and play support and easy embedding. There are four solutions available, targeting at different devices:

- **Professional**: Offers a rich and comprehensive platform
- **Secure**: Offers robust security capabilities
- **Micro**: Aims at resource-constrained devices that do not require an operating system
- **Cert**: Aims at systems subject to safety certifications.

7. Development frameworks

When developing applications for the IoT, any “traditional” development framework could be helpful. Nonetheless, many protocols and approaches are pretty hard to implement correctly. Therefore, we point out some development frameworks with support for IoT relevant standards or aiming at specific approaches, like agents, that are useful for IoT scenarios.

7.1. Eclipse IoT [23]

Eclipse has been one of the most used IDEs ever since and now offers an IoT version including support for standards such as MQTT, OMA, etc. It has already been used for various open-source implementations, such as Mosquitto [24] (a lightweight MQTT broker) or Californium [25] (a CoAP framework).

7.2. JADE [26]

JADE provides a Java-based agent development framework to build multi-agent systems using an agent communication protocol standardised by the "Foundation for Intelligent Physical Agents (FIPA)". The agents exchange their messages by means of the agent communication language ACL, using one of the commonly used communicative acts like inform, request, agree, not understood, and refuse. It furthermore enables remote control of agents via a remote GUI.

7.3. Jadex [27]

The Jadex reasoning engine follows the „Belief Desire Intention" (BDI) model and allows for programming intelligent agents in XML and Java, that can be deployed on middleware like JADE.

8. IoT communities

In order to allow for efficiently using and exchanging each of IoT technologies, there have to be clear definitions and requirements for ensuring interoperability. Therefore, we name a few communities and consortiums taking care of this challenge.

8.1. Allseen Alliance [28]

Community for collaboration to help enable the IoT / Internet of Everything.

8.2. Open Interconnect [29]

Consortium with the goal of defining connectivity requirements and ensuring interoperability of the emerging IoT.

References

[1] GroveStreams. Online: https://grovestreams.com/
[2] Exosite. Online: https://exosite.com/
[3] SensorCloud. Online: http://www.sensorcloud.com/
[4] Xively. Online: https://xively.com/
[5] SeeControl. Online: http://www.seecontrol.com/
[6] ThingSpeak. Online: https://thingspeak.com/
[7] Wolfram Data Drop. Online: https://datadrop.wolframcloud.com/
[8] Oracle Internet of Things. Online: http://www.oracle.com/us/solutions/internetofthings/overview/index.html
[9] ThingWorx IoT Platform. Online: http://www.thingworx.com/IoT-Platform
[10] Node-RED. Online: http://nodered.org/
[11] Macchina.io. Online: http://macchina.io/
[12] Kaa. Online: http://www.kaaproject.org/
[13] IoTivity. Online: https://www.iotivity.org/
[14] Ethereum. Online: https://www.ethereum.org/
[15] Tessel. Online: https://tessel.io/
[16] ARM mbed IoT Device Platform. Online: http://www.arm.com/products/internet-of-things-solutions/mbed-IoT-device-platform.php
[17] Contiki. Online: http://www.contiki-os.org/
[18] TinyOS. Online: http://www.tinyos.net/
[19] RIOT-OS. Online: http://www.riot-os.org/
[20] IOTDB. Online: https://iotdb.org/
[21] Apache Kafka. Online: http://kafka.apache.org/
[22] RTI Connext DDS. Online: https://www.rti.com/products/index.html
[23] Eclipse IoT. Online: http://iot.eclipse.org/
[24] Mosquitto. Online: http://www.eclipse.org/mosquitto/
[25] Californium. Online: https://www.eclipse.org/californium/
[26] JADE. Online: http://jade.tilab.com/
[27] Jadex. Online: https://vsis-www.informatik.uni-hamburg.de/vsis/research/lookproject/27
[28] Allseen Alliance. Online: https://allseenalliance.org/
[29] Open Interconnect. Online: http://openinterconnect.org/