



HAL
open science

Industrial Internet of Things: Security of Interoperability

Fergal Martin-Tricot, Cédric Eichler, Pascal Berthomé

► **To cite this version:**

Fergal Martin-Tricot, Cédric Eichler, Pascal Berthomé. Industrial Internet of Things: Security of Interoperability. RESSI, May 2019, Erquy, France. hal-03021574

HAL Id: hal-03021574

<https://hal.science/hal-03021574v1>

Submitted on 24 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Abstract

Industry uses more and more *IoT* components to have a better control on production and logistic processes. This growing openness induces some security risks. Moreover, the multiplicity of open and proprietary protocols used in *IoT* makes the interoperability management paramount, especially in the industrial world. We therefore propose to study data security in heterogeneous industrial *IoT* environments.

Industry and *IoT*

Machine to Machine (*M2M*) is more and more deployed in industry

- Allows real-time and direct control on processes
- Raises two main questions:
 - ⇒ Proprietary systems → closed communication protocols
 - ⇒ New communication and collaboration approach → attack surface enlarged

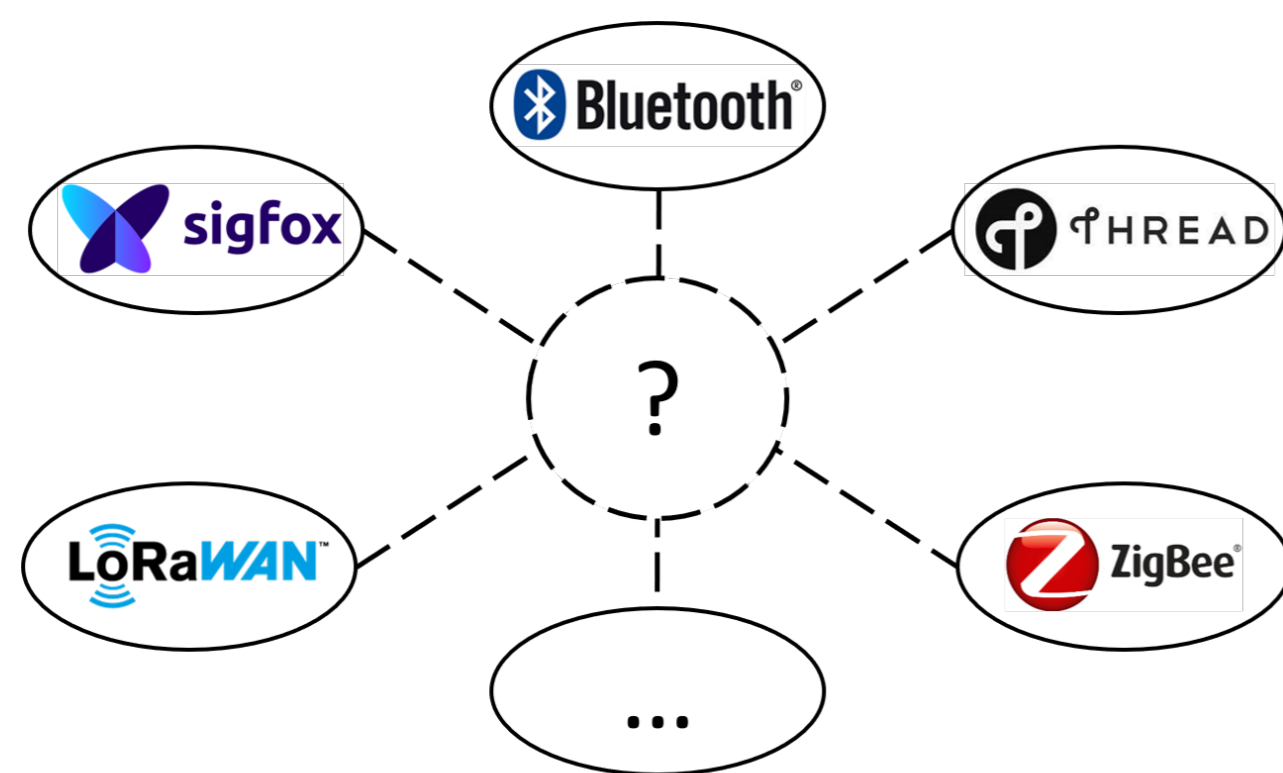
The next step: Internet of Things (*IoT*)

- Increasingly investigated for industry needs
- Rationalisation of deployments with cost-less and energy-efficient devices
- Usually with wireless protocols:
 - ⇒ New security issues (no physical access needed...)
 - ⇒ A multitude of used protocols → interoperability difficulties

Interoperability

Interoperability, a central point of industrial *IoT*

- Being locked into one manufacturer environment can be problematic
- Multiplicity of used protocols
- Required for control centralization



Standardization, the solution?

- Best way to solve interoperability in IT world
- Some initiatives exist (INTER-IoT, Intel IoT...)
- One stands out by its international support: **oneM2M**

oneM2M

The standard to rule them all:

- Based on smartM2M, an ETSI workgroup
- Supported by almost a dozen national and international standard institutes
- Facilitates management and interoperability in *M2M* and *IoT* networks



This standard is based on a modular approach where each entity has a specific role. The most relevant for us: Interworking Proxy Entity (*IPE*), the heart of interoperability [1]:

- An interface between *oneM2M* and a third party protocol
- Translates data and instructions

⇒ Important security role between *oneM2M* and third party protocols' security measures

Test Scenario

Objectives:

- ⇒ Experiment *oneM2M* with a third party protocol: *ZigBee*
- ⇒ Study the data security: interfaces, exchanges, translations...

The scenario:

- ⇒ A temperature sensor node and a display node (communicating with *ZigBee*)
- ⇒ We want to get those data in a *oneM2M* architecture

Implementation of oneM2M Architecture

Chosen *oneM2M*'s implementation: *IoT OCEAN* [2]

- Open-source
- One of the most actively developed

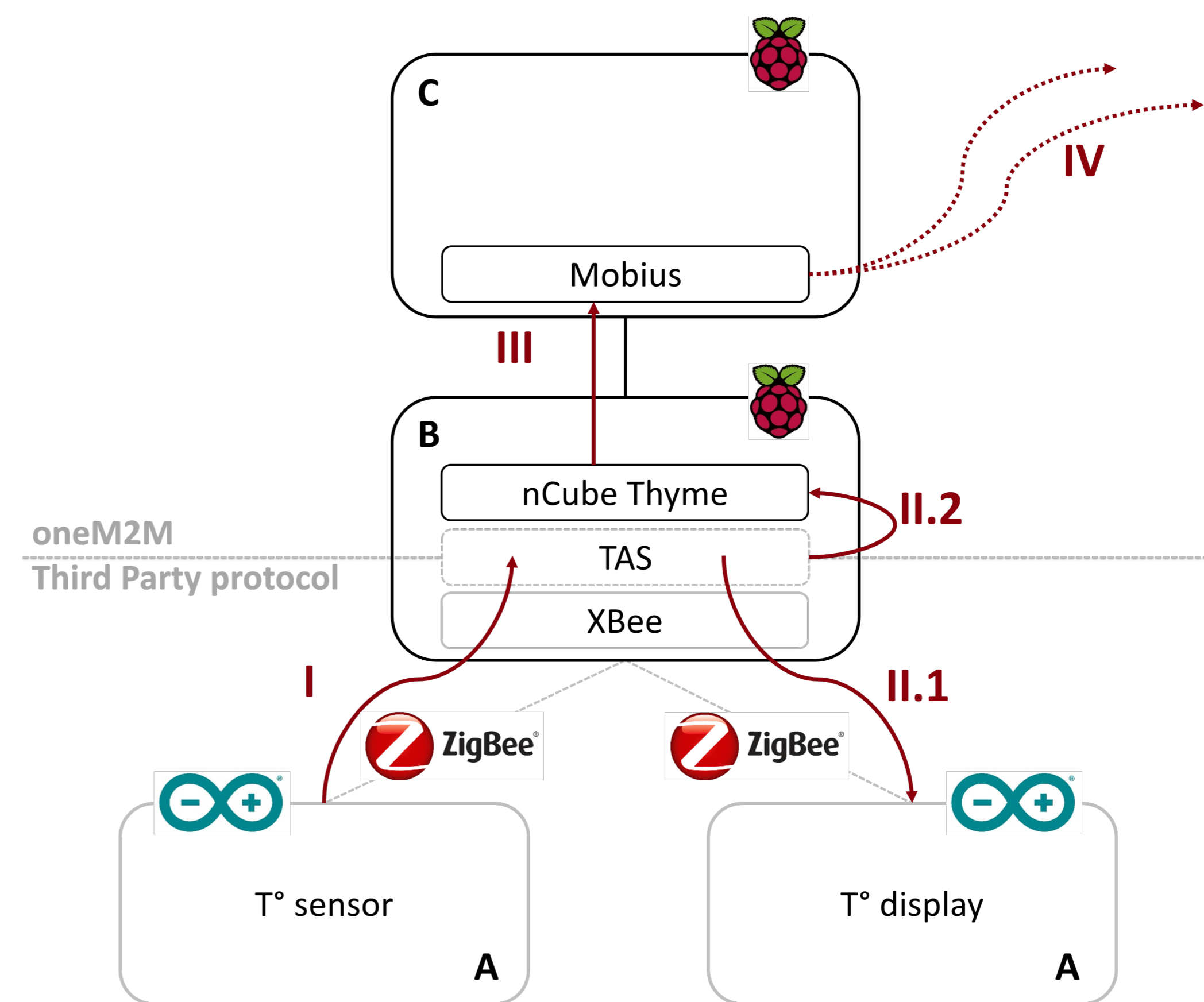


Figure 1: Implemented architecture

Implementation (cf. Figure 1):

- 2 Arduino - Temperature sensor/display - *non-oneM2M Node (NoDN)*
- Raspberry Pi - *Interworking Proxy Entity (IPE ⇒ TAS in IoT OCEAN)*, *Application Dedicated Node (ADN-AE ⇒ nCube in IoT OCEAN)*,
- Raspberry Pi - *Infrastructure Node (IN-CSE ⇒ Mobius in IoT OCEAN)*

Thing Adaptation Software (TAS):

- *IoT OCEAN*'s approach of the *IPE*
- Software component that communicates with third-party protocol and *nCube*
- Based on an implementation sample, homemade to work with a specific protocol
- Our *TAS* was done in Javascript and communicates with *nCube* through its REST API
- Integration of a new protocol → a new *TAS*

Data path in this system (cf. Figure 1):

- The sensor node sends its data, in *ZigBee*, to the *TAS*
- The *TAS* processes data, then re-sends them to the display node (II.1) in *ZigBee*, and to *nCube* in HTTP (II.2)
- nCube* immediately shares the data it receives with *Mobius*
- If needed, *Mobius* can share these data with others actors

Security concerns

Inherent to *IoT* protocols:

- Security in most popular ones has already been largely studied
- On mature protocols, the majority of security issues are caused by improper implementation [3]

Inherent to interoperability:

- Unsolved open challenge: how to ensure end-to-end security in heterogeneous deployments?
- The study of security mechanisms in *oneM2M* can solely be theoretical as no open-source solution implements all of them (for now)
- The security of the interface (the *IPE*) is a critical yet midly studied point

Conclusion

- *oneM2M* is thus a promising standard to solve interoperability issues in the industrial world.
- There are some strong security functionalities in *oneM2M* (up to the *IPE*) and in the classical *IoT* protocols (up to the edge).
- Unfortunately, when using *oneM2M* as an interoperability tool, it is necessary to have a blind trust in the *IPE* and the third-party protocol edge.
- To ensure end-to-end data security, one must find solutions to mitigate this issue.

[1] J. Yun, R. C. Teja, N. Chen, N.-M. Sung, and J. Kim, "Interworking of *oneM2M*-based *IoT* systems and legacy systems for consumer products," in *2016 IEEE International Conference on Information and Communication Technology Convergence (ICTC)*, oct 2016.

[2] M. Ryu, J. Yun, T. Miao, I.-Y. Ahn, S.-C. Choi, and J. Kim, "Design and implementation of a connected farm for smart farming system," in *2015 IEEE SENSORS*, nov 2015.

[3] G. Tuna, D. G. Kogias, V. C. Gungor, C. Gezer, E. Taşkın, and E. Ayday, "A survey on information security threats and solutions for machine to machine (*M2M*) communications," *Journal of Parallel and Distributed Computing*, vol. 109, pp. 142–154, nov 2017.