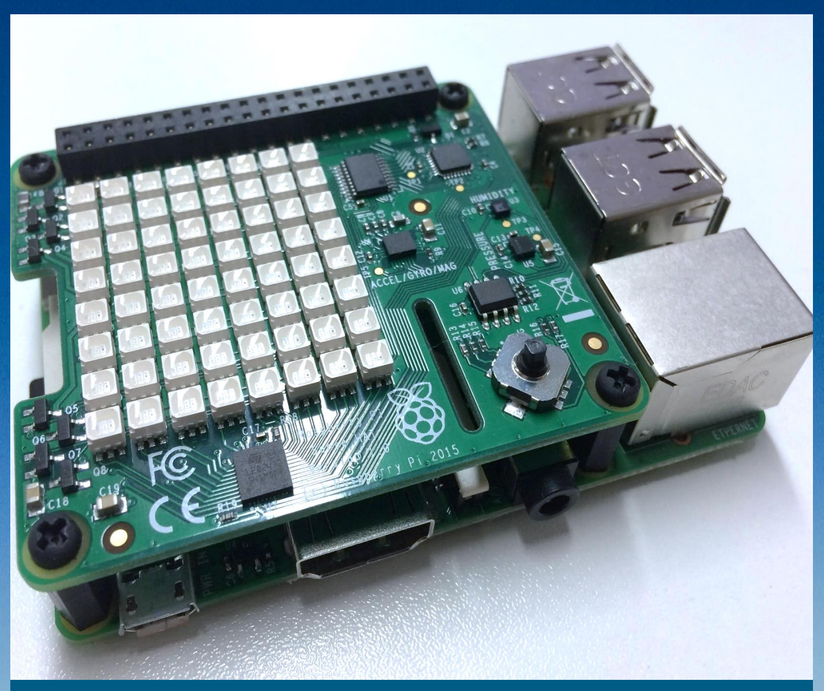
Federation: A Multi-Vendor Internet of Things

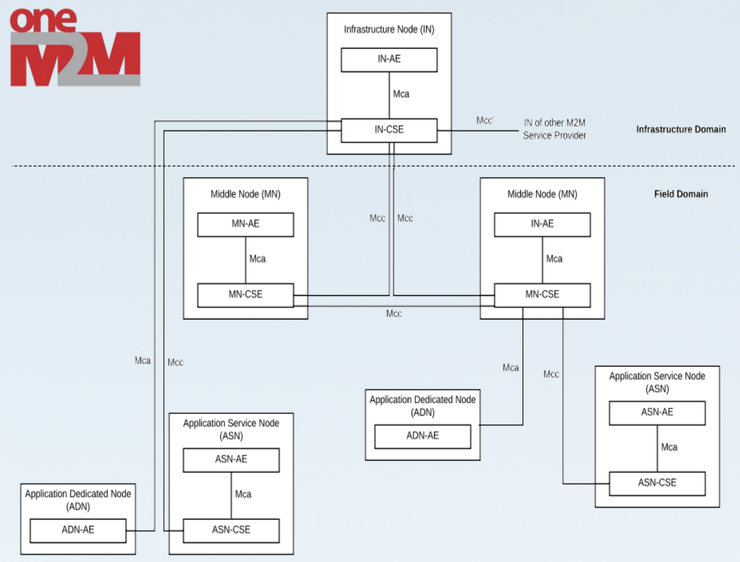
Team 30: Adib Pournazari, Altay Adademir, Dennis Parchkov, Edmond Ipindamitan, and Matthew Consterdine.
Supervised by Mohammed El-Hajjar, examined by Mike Wald.

Abstract
For the mass deployment of the Internet of Things to be a success, a global standard for machine to machine communication needs to become established. In this report, the open oneM2M standard for Machine to Machine communication will be explored. This project will research its capabilities, how to make use of the standard, and ultimately build systems with IoT sensors upon it. Using these systems data streaming, live video and federation will be investigated.
This project is research orientated, investigating federation with InterDigital as the client. They created the oneTRANSPORT data marketplace using the oneM2M standard that exposes proprietary data to users, analysts and developers.



Background

oneM2M[1] is a global standards organisation focusing on standardising Machine to Machine communication. They have created many specifications which supply a framework helping Internet of Things vendors to develop secure and interoperable products. oneM2M is comprised of three distinct layers: the application layer which is where the team will work, the common services layer which provides functionality, and the network services layer which abstracts away the underlying protocol. Armed with this knowledge, the team created a plan: develop a oneM2M application that would read and federate data between gateway and server, and between server and oneTRANSPORT.



Hardware

The team evaluated different hardware platforms, ultimately choosing a Raspberry Pi with a Sense HAT. Unlike a laptop it is cheap with ample GPIO, unlike an Arduino it is powerful enough to run Linux, and unlike Intel Edison it enjoys widespread support from manufacturer and from within the open-source community. To push the specification to the limits, a camera was purchased and attached to the Raspberry Pi via a serial header. Then stills were captured, and transmitted via oneM2M federation.

Infrastructure

After discounting the client's Azure instance due to configuration issues, the team switched to a Digital Ocean droplet. For security, a LetsEncrypt HTTPS certificate was acquired and a VPN tunnel was created to provide a tunnel to bypass NAT.

OM2M Connecting things

After evaluating different oneM2M platforms (KETI OCEAN, Eclipse OM2M, OpenDayLight IoTDM, ATIS oneM2M, and IoTOASIS SI) Eclipse OM2M[2] was chosen because of it's many features, open source codebase, and active development community. Python scripts were written to interface with the Sense HAT, returning a data point or stream. This approach avoided raw i2c communications. Using the oneM2M standard and example code, a plugin was developed to call the scripts, process the output, and store in the integrated H2 database. Finally, OM2M automatically synchronises the data between gateway and server. To aid development, unit and integration tests were written and run.
Video streaming was tested, but ran poorly.

open mtc

OpenMTC[3], a Python oneM2M implementation, was released a few months into the project. The team decided to investigate, and ultimately adopt OpenMTC. Python allowed for quicker iteration cycles with improved performance - the Python scripts could be directly integrated into the plugin. After the disappointing results of OM2M video streaming (about 5 frames per second at a low resolution), the team reimplemented it using the 'python-picamera' library. Results were impressive, and after tweaking certain parameter 20 frames per second was achieved at a high (720p) resolution. This proves that the oneM2M standard is capable of high resolution video streaming, even if not all implementations are equally capable.

oneTRANSPORT Open marketplace for data

InterDigital is a research organisation specialising in the Internet of Things, earning tens of thousands of patents. They embraced oneM2M when building oneTRANSPORT[4], a smart transport data platform. In this project, the team federated firstly between OM2M gateway and server, OM2M server to OpenMTC, and finally OpenMTC to oneTRANSPORT. Internal OM2M federation was unsurprisingly a success, but due to resource type 28 (Flexible Containers) being unimplemented by OpenMTC, federation was unfortunately only achieved in one direction. Finally, federation between OpenMTC and oneTRANSPORT was successful, after patching the Python code to add a custom authentication header required by the platform.

Future Work

- Investigate additional federation mechanisms.
- Increase complexity and scale, to test the design.
- Explore new environments using CoAP or MQTT.
- Improve video streaming with a MJPEG proxy.
- Visualise data with an interactive dashboard.

Conclusion

The project successfully demonstrated federation between gateway and client, different implementations of the oneM2M standard, and specifically with InterDigital's oneTRANSPORT platform. Video streaming was proven feasible with OpenMTC providing a stable 20 frames per second. A Raspberry Pi powered platform was developed. And finally, if more companies adopt the standard, interoperability will create a network of interconnected devices. Communication will enrich Internet of Things data and help change the world.

Footnotes

1. oneM2M Technical Specifications: https://onem2m.org/technical/published-documents
2. Eclipse OM2M Project Homepage: https://www.eclipse.org/om2m
3. OpenMTC Project Homepage: http://www.openmtc.org
4. oneTRANSPORT Data Marketplace: https://onetransport.io

Image Credits

- City Render by Regiars, CC BYSA 3.0: https://commons.wikimedia.org/wiki/File:Территория "СМАРТ Сити Казань".jpg
- Raspberry Pi by Lucasbosch, CC BYSA 3.0: https://commons.wikimedia.org/wiki/File:Raspberry_Pi_B+_illustration.svg
- Cloudy Sky by Matteo Fusco, Unsplash: https://unsplash.com/photos/m94kn8Rp61Q
- Icons from Microsoft Segoe UI Emoji