25th ITS World Congress, Copenhagen, Denmark, 17-21 September 2018

EU-TP1590

Living Lab Bus platform for IoT service development in public transport context

Immo Heino[1], Juho Kostiainen[1], Janne Lahti[1], Juhani Linna[2], Olli Pihlajamaa[1]

1. VTT Technical Research Centre of Finland Ltd., Finland
2. University of Tampere, Finland

Abstract

The living lab paradigm involves people in the innovation process and is used for studying user read needs in an everyday use environment. Living Lab Bus (LLB) is a cooperative approach bringing together different partners involved in providing transport and mobility services with an aim to contribute towards ease, attractiveness and efficiency of public transportation. Ten electric buses in normal operation in the Helsinki region serve as the LLB platform to spark and test new ideas and solutions as well as for collecting user feedback on and ideas for new services. The LLB collaboration provides a concrete platform enabling easier development, testing and demonstration of new solutions in a real public transport context. The implemented platform is a mixture of commercially available off-the-shelf (COTS) and prototype components and software provided by VTT, universities and private companies. This paper describes the system architecture and technical details of the LLB platform and design principles behind the applied technical solutions. The system has to ensure enough functional flexibility to respond to evolving needs and to assure overall security – despite the system's openness.

Keywords: living lab, public transport, IoT

Introduction

Living lab is a paradigm referring to a community driven involvement in innovation processes in real-life context. It has been suggested as an efficient way to engage all stakeholders (end-users, companies, researchers, public authorities and policy makers) in the early stages of open innovation process in order to experiment and develop new concepts, products and services (Pallot 2006, Higgins 2011, Van der Walt 2011). In living lab set-ups, new technologies and concepts can be examined in an everyday use environment gaining better understanding of real user needs and how the solutions are used and adopted. The aim of living labs is to increase product and service quality, reduce market-based risks and increase the rate of market diffusion. Tang et al. (2012) also see living labs as a link for establishing public-private partnerships.

Living Lab Bus (LLB, www.livinglabbus.fi) is a research and development project focused around multi-stakeholder collaboration to facilitate innovation and faster validation and deployment of new solutions. It comprises a concrete development platform and a cooperative approach bringing together different transport and mobility service ecosystem partners such as users, service and technology providers, transport operators, cities, research institutes etc. (Kostiainen et al. 2016). Its aim is to contribute towards sustainable mobility and increasing the attractiveness of public transport as well as developing new commercial opportunities. With the LLB platform, where a fleet of electric buses in normal operation is used as a concrete real world laboratory environment, organizations and individuals following the "commonly agreed LLB ground rules" are able to innovate, test and develop solutions to improve public transport experience and operation. In addition to the buses and other public transport vehicles, the broader LLB context extends to parts of the related infrastructure, such as bus stops, depots and terminals.

The LLB environment (Figure 1) allows the development, testing and demonstration of information and technology solutions in a real public transport context. The fleet of 10 electric buses developed by Linkker, owned by the Helsinki Region Transport authority (HSL), operated by various bus operators and equipped with the LLB onboard setup:

- acts as a mobile Internet of Things (IoT) sensor platform providing an extensive selection of real-time and accumulated history data,
- provides edge computing capabilities (with an onboard computer) for controlling collection of data and enabling its pre-processing as well as hosting 3rd party software, and
- allows means for provision of services and content to passengers and bus drivers utilizing onboard computing power, public displays and a mobile service provision channels.

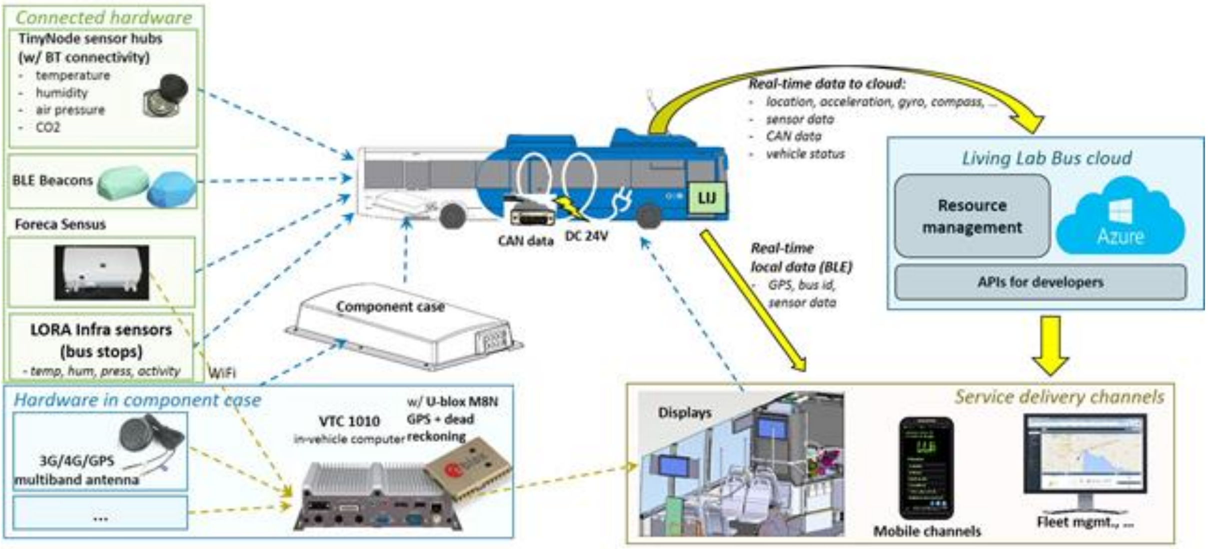

Figure 1 - The LLB service development platform.

Furthermore, two test buses that are not operating on any line can be used for pre-testing solutions before exposing them in public.

The LLB project enables several ways to exploit its offering. Developer Portal provides a complete set of information and tools to build (mobile) web applications in the LLB Landing Page Framework (LLPF) and utilize the open LLB APIs. Application development in LLPF enables the provision of the apps through the LLB Landing Page that forms a single source for all mobility services offered in the LLB context. Furthermore, applications can be coupled with tailorable feedback elements that allow developers to request feedback from application users.

The set of open LLB APIs consists of interfaces to access real-time data as well as for accumulated history data from all the buses in the LLB fleet. The history data can be fetched as data chunks consisting of data for a specified day and bus. The LLB APIs can be used both in the development of applications using the LLB Landing Page Framework and for other kinds of applications and services.

In addition to software development, the LLB buses can be used for testing hardware and complete third party systems in a real bus environment. The solutions to test can be standalone systems but they can also utilize the basic onboard LLB setup, for example to get power and internet connectivity.

The main emphasis of this paper is to describe the system architecture and the technical implementation of this LLB platform as an Internet of Things (IoT) system. The following chapters will give some theoretical background to the generic IoT architectures and explain in more details the requirements, design goals and technical implementation of the LLB environment. The service development tools offered by the LLB Development Portal (e.g. tools provided by LLPF) are excluded from this paper.

Internet of Things

Internet of Things (IoT) can be defined as a near future concept, where objects like home appliances, vehicles and other everyday life things are equipped with sensors, actuators and computing units. These things are able to harvest information from the environment (sensing) and interacting with the physical world (actuation/command/control) by using existing internet infrastructure and standards to provide services for information transfer, analytics, applications, and communications (Atzori 2010, Gubbi, 2013, Zanella 2014). There is no generally accepted definition for the Internet of Things. Despite the fuzziness around the IoT definition, a rough layered architecture (Figure 2) for IoT systems has been identified (Wu 2012, Khan 2012):

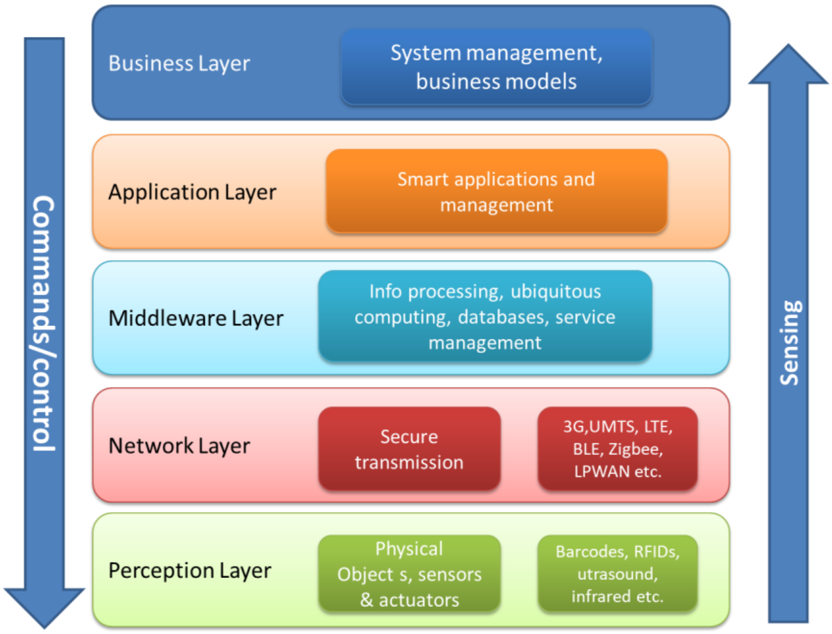

Figure 2 - Layered IoT architecture (adapted from Khan 2012).

1) Perception or Device Layer consisting of the physical objects, related sensors and actuators. This layer collects object specific information such as location, temperature, orientation, motion, vibration, acceleration, humidity, chemical composition of the air etc. from sensor devices. Actuators affect the physical world by changing the state of objects. The assembled information and actuator commands are passed through the Network Layer.

2) Network (or Transmission) Layer securely transfers the information from sensor devices to the information processing system and conveys commands from the upper layers to actuators. The transmission medium can be wired or wireless (3G, 4G or 5G, Wifi, Bluetooth, ZigBee etc.) The network layer passes the information from Perception Layer to Middleware Layer and vice versa.

3) Middleware Layer performs information processing based on collected information and makes automatic decisions based on the computed results. Since an IoT system could contain hundreds or thousands of sensors producing huge amounts of data for further processing, IoT can benefit greatly from the virtually unlimited capabilities and resources of cloud computing to compensate the technological limitations of the lower layers (e.g., storage, processing and energy).

4) Application Layer: This layer provides management for the applications utilizing the object information processed in the Middleware Layer. These applications can be related to, for example, smart health, smart farming, smart home, smart city and, as in our specific case, smart mobility.

5) Business Layer is responsible for the management of the overall IoT system including the applications and services. The real success of IoT technology depends on good business models, and

based on analysing collected data, this layer will enable taking future actions and the development of business strategies.

Low power requirements for system on chip (SoC) circuits as sensors and the increased usage of cloud computing as part of IoT systems have brought up some fundamental issues related to this approach. The main challenges are very modest computing power at the sensor/actuator level and unreliable latency times for cloud based responses. In vehicle environments especially, where some application needs either real-time or predictable response latencies, this kind uncertainty is untenable. To alleviate the observed problems, fog or edge computing (Bonomi 2012) – a highly virtualized platform offering computing, storage and networking services between the sensors and the cloud at the bottom of middleware layer or upper network layer – has been proposed as a solution. The fog concept, where computing is partially brought to the edge of networks (Yi 2015), provides needed elasticity of resources and decreases unnecessary data traffic and thus latency, increasing reliability and quality of applications and services.

The fog computing architecture of the Living Lab Bus platform

The LLB system architecture (Figure 3) closely follows the generic IoT layered architecture described earlier.

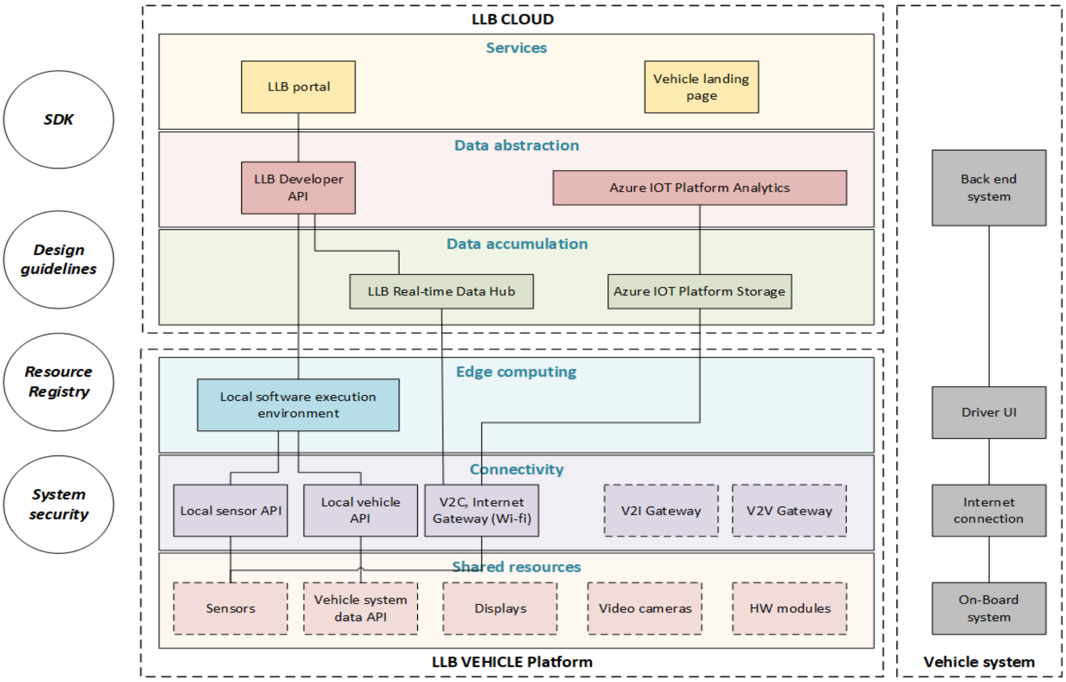

Figure 3 - The Living Lab Bus IoT architecture.

LLB's device (perception) layer contains various types of sensors such as location, acceleration, temperature, nitrogen and carbon dioxide etc for monitoring traffic environment. Additionally, the ability to read the vehicle Controller Area Network (CAN bus) provides over 100 different measurements from a vehicle's internal data bus to use for various purposes such as battery monitoring etc.

Based on the sensor information and related data analysis algorithms applied either edge or cloud computing layers, different types of actuators can be activated accordingly. Currently LLB IoT system deploys only a limited set of all possible actuator types – namely public displays, showing route related contents, and programmable feedback collection devices installed on exit railings. Auxiliary actuators like a remote stop push button is planned to be included later.

Connectivity to sensors and actuator devices at the network layer is mostly enabled by wireless links to ease the sensor installation process (e.g. tedious wiring is not needed). Some active devices such as GPS receivers and displays need constant power and data cabling from gateway units. At the LLB's middleware layer, extendable open mobile edge computing units (so called OMEC gateways) gather sensor data from various sources and send it to the LLB cloud service in real time utilizing 4G mobile network or pre-process and use the data locally.

The LLB cloud service at the application layer offers data collected from the vehicles via several publicly accessible APIs for enabling new service innovations opportunities both for commercial companies as well as developer communities such as the Helsinki Region Transport developer community and universities. Developed services for the LLB environment can be hosted by the cloud (web applications), by mobile devices or by OMEC gateway units.

Business layer functionality is responsible for analysis and visualization of the collected information and handling the user feedback. Based on this information, service providers can identify more closely user needs and adjust service offerings accordingly.

Requirements for the LLB platform and the practical implementation

The real world working environment sets some specific requirements to system management, safety and security. Maintaining and altering the dynamic and constantly evolving LLB environment requires high flexibility both in hardware and software configurations. Buses are in daily operation as much as possible and minimal interruptions for daily schedules are desirable, so especially the required time for physical (hardware) modifications should be minimized. This goal could be reached by using standardized and interoperable systems (e.g. "plug & play" components) enabling easy installations of new components and replacement of failed units.

Albeit the ongoing standardization efforts such as the Information Technology for Public Transportation initiative (ITxPT) targeted to ease some interoperability problems related to building up of public transportation vehicles ICT infrastructure, there is still plenty of development work ahead to enable easy power plugins and wirings for installing auxiliary ICT devices into vehicles like buses or trams. In addition, space for computing units' device racks or cabinets is seldom taken into account in vehicle chassis or interior designs (Figure 4). These facts set some demanding preconditions to physical device designs.

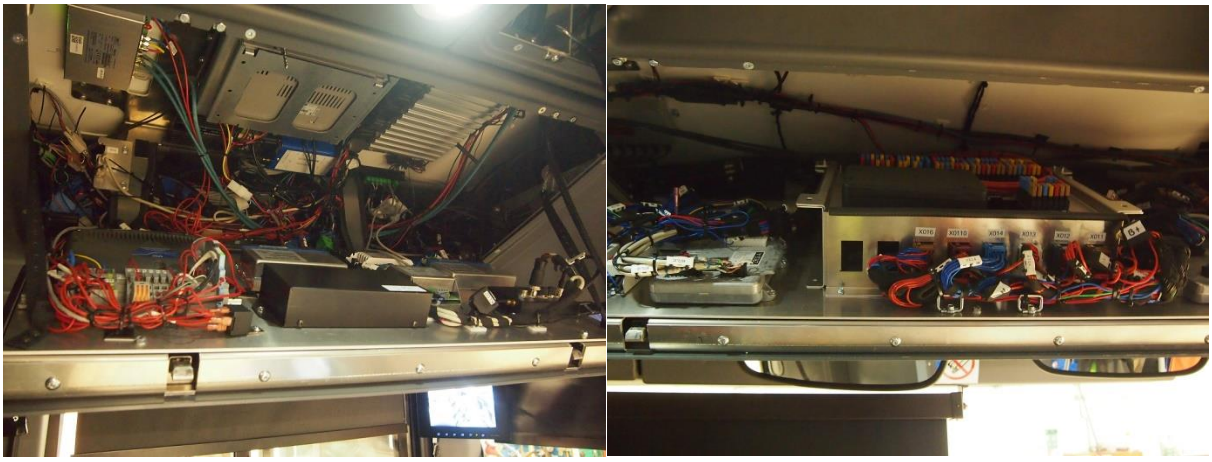

Figure 4 - Very limited cabinet space for additional equipment.

The maintenance of installed components requires constant condition/state tracking at hardware and software level, so monitoring is needed to alert of any unusual incidents. Support for the system resource and configuration management is also necessity since not all the vehicle setups in the LLB fleet are identical. Some LLB vehicles might contain different sensors and actuator configurations and there might also be differences among the buses' standard wiring and interiors that LLB system maintenance personnel should be aware of when planning and implementing modifications to the LLB system.

Traffic safety is an essential requirement for public transport and supplemental systems – such as the LLB platform – should not compromise it. All hardware additions into the buses' interior (e.g. sensors, extra displays and feedback units) need to be placed in a way that does not hamper driver's attentiveness and awareness or passengers comfort and safety. Since public property can often be vulnerable for mischief, placement and secure attachment of devices that are not under a driver's or public's constant supervision (e.g. sensors and feedback devices) requires some careful planning.

One of the most crucial aspects of the LLB platform is the security of the vehicle ICT systems. ICT systems have an extremely important role in the Linkker electric buses – including battery charging control and guiding energy efficient driving as well as the information systems. In the LLB environment, two separated computing systems co-exist in a bus: the mission critical driving system (i.e. normal systems) and the open mobile edge computing (OMEC) system (i.e. the test platform equipment). These systems must be isolated from each other since the mission-critical vehicle part has to be protected from possible malfunctions or malicious components installed into the OMEC environment.

OMEC units are also responsible for context (location) sensitive contents on the public displays used as part of the LLB environment. Since the OMECs are connected to the outside world via internet, the OMEC units might be attractive targets for cracking attempts with the aim of feeding inappropriate content into the public displays. Also, the OMECs' display software could stick on showing only "frozen/black screen" content in a case of malfunction. Due these reasons, a display monitoring system is highly desirable.

Design and practical implementation aspects of the LLB system

In the LLB project these described challenges are addressed by using a careful system design and applying by some ICT security countermeasures. For system security reasons the open and shared OMEC execution environment enabling the addition of hardware and supporting 3rd party software hosting in the buses is physically isolated from the rest of the vehicle's ICT systems. The only connection between these systems is a one way (read-only) connection to an auxiliary CAN interface and to the main power switch that sends the activation signal to the local OMEC gateway in a bus when a driver turns the main switch off. This activation signalling is needed to shut down the gateway in such a way that the integrity of the OMEC unit's software is guaranteed.

Some compromises for the system security have been forced to make for non-critical data. Sensors attached to various places of the bus's chassis are connected to an OMEC gateway by using unencrypted wireless connections. Simple sensors often cannot provide computing features to assure data integrity and security due to complex encryption algorithms increasing energy usage of these battery operated sensors. Since the information collected by these sensors is distributed via open access APIs anyway and is not mission critical in any sense, data transmission between sensors and a gateways has not been encrypted.

Software maintenance, such as operating system and application updates for OMEC gateways, needs Over The Air (OTA) functionality e.g. connection via the internet. This access is provided by using 4G wireless network and utilizing secured protocols such as Secure Shell (SSH). Security could be further improved by using virtual private networks (VPN) such as IPSec. Inside an OMEC gateway, "lightweight virtualization" provided by Docker containers protects 3rd party software providers' applications from disturbing each other.

LLB platform's monitoring system includes both overseeing OMEC units and display contents. Each OMEC gateway in the system periodically sends its status information in addition to the normal sensor content messages. A watchdog software in the cloud service alerting of any abnormal measurements supervises this status information. Monitoring the display content is implemented as a stream of screen captures sent periodically from the buses to the "control centre". Captured individual screenshots are tiled up as an overview screen from where an operator can easily glance if the content is appropriate or if the display is dark or frozen. Unfortunately, human intervention is still needed (for more advanced content recognition than detecting a blank display) since the semantic content interpretation of content pictures is beyond computers' (artificial intelligence) current capabilities.

The practical implementation of the LLB platform is a mixture of standard commercially available off-the-shelf (COTS) systems and prototype components and software provided by VTT, universities and private companies

Adding new hardware and getting a power supply for devices might be tedious even without considering required wirings. To ease these tasks, a standardized water and dustproof (IP65) cabinet case is used as a main building block for the on-board LLB hardware. The housing is roomy enough to accommodate other computing components in addition to the LLB's OMEC unit (Figure 5, left). This modular and extendable industrial proven box has a DIN-rail enabling CAN bus interfacing and time relay controlled power supply. Unfortunately, no feasible easy to use wiring alternative has been found for this purpose. Since the requirements for placement of auxiliary devices seem to be very heterogeneous, there may not be a universally acceptable solution for power and data cabling.

LLB's wireless sensor units are based on VTT's battery powered TinyNode sensor hubs (Figure 5, right). A TinyNode hub contains a set of integrated measurement devices such as a 3D accelerometer, magnetometer and inclinometer, air pressure, humidity and temperature sensors. Additional attachment options include CO_2 and pressure difference sensors that are not directly integrated into a hub's circuit board due to power consumption issues. TinyNode's expected battery lifetime is 3-6 months depending on a measurement interval (adjustable), a sensor type and communication needs. TinyNodes are able to communicate with the OMEC gateways by using either BLE or proprietary LoRa Low-Power Wide-Area Network (LPWAN) communications technology. BLE based unit are attached inside the bus chassis and LoRaWAN enabled devices are utilized at bus stops and other traffic hubs where the BLE's range does not extend.

In addition to the basic TinyNode sensors populated by every LLB busses, some vehicles are equipped with additional measuring devices. These more complex sensors that require a constant power source include Foreca's (a Finnish weather service company) measuring box attached to tailgate of a bus, which is able to measure and send CO_2, NO_2, temperature, humidity and infrared video information to the OMEC unit via BLE or WiFi. Another similar type of complex sensor device is Vaisala's air measuring station attached to a bus roof providing CO, CO_2, NO_2, temperature, humidity, ozone (O_3) and micro particle counts measurements. Several video cameras are also planned to be mounted to vehicles ceilings enabling to passenger real time counting and detecting if baby carriages or wheelchairs occupy the middle part of a bus.

Figure 5 - The OMEC gateway unit box ready for installation (left) and LoRaWAN connected TinyNode measurement hub (right).

The onboard OMEC gateway unit (Figure 5, left) is based on industrial strength PC running Ubuntu Linux 16.04 LTS operating system. The computer has a built in sensor set including an accelerometer, a gyroscope and a compass, a CAN bus interface and good connectivity to external GPS antennas and displays. Communication support includes 4G, WiFi and BLE and has been extended by an additional LoRaWAN access point connected via a USB interface. OMEC gateway hosts the software that will provide context sensitive information on the buses' 21" or 27" high definition (HD) displays.

Every gateway in the LLB system sends collected information from wireless sensor to the back end cloud service once every second by using the Message Queuing Telemetry Transport (MQTT) protocol (Figure 6). The utilized cloud service, Microsoft Azure, enables virtualizing Linux operating system servers processing incoming data from sensors and supports storing the output information as binary large objects, blobs. Since every electric bus sends approximately 200 bytes long measurement information bursts once every second, the cumulative daily raw data is in the gigabytes range (approximately 50-100 megabytes per a bus depending on daily schedule). The collected and processed information in blobs is publicly accessible via APIs offering third parties (e.g. companies and developer communities) an opportunity to utilize that information as the basis for or addition to new value adding services.

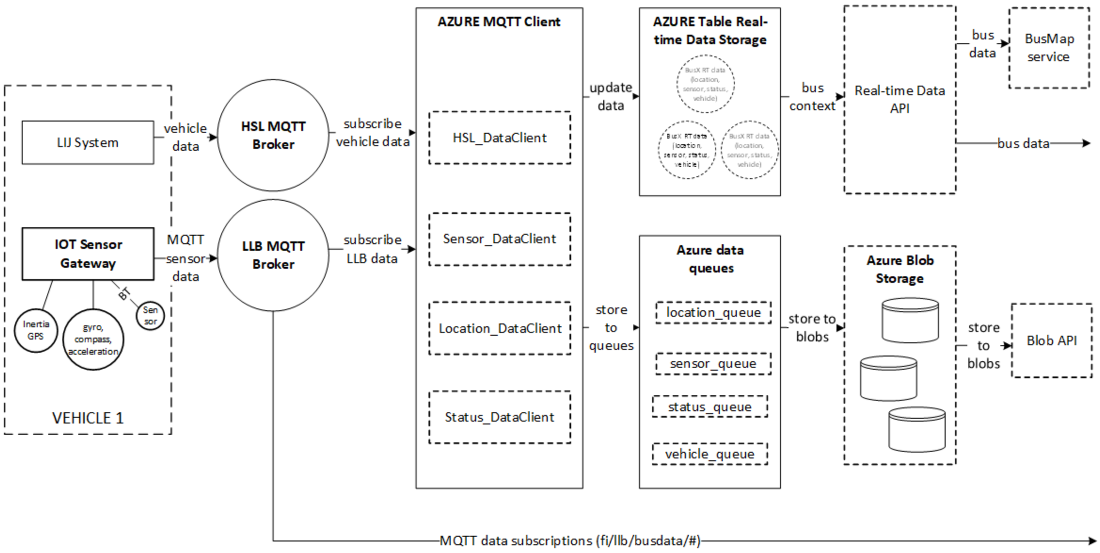

Figure 6 - Schematic diagram of the LLB components and communications.

LLB environment's actuator devices, namely the planned remote stop-unit enabling pressing the "stop button" from distance outside the bus for registered users (to avoid misuse or harassment) and feedback collection units are connected wirelessly using either proprietary, Bluetooth Low Energy or LoRaWAN protocols. Feedback devices targeted to all travellers are attached to middle door exit rails enabling easy to use public access. Travellers' mobile phones as an alternate feedback mechanism are connected to the system via an app utilizing 4G network and Azure cloud service.

Current status and future work

The LLB platform is operational in buses on three different lines in the Helsinki region. Each line has two or three electric buses equipped with the LLB IoT system. Some bus stops are also equipped with TinyNode LoRaWAN modules for enabling monitoring environmental conditions such as temperature and humidity.

At the time of writing this paper information collected by the LLB system has been used as a basis for participants' internal product developments and utilized for new service innovation courses in Aalto University, University of Tampere and hackathon events. Some new service ideas, like "remote stop button" i.e. requesting stopping the bus from outside the vehicle (by informing the bus driver about waiting passengers at upcoming bus stops) have already emerged and are planned to be implemented and tested in the LLB context.

Since the realization of the first LLB platform buses in summer 2017, there has been an increasing interest to utilize the environment as a test bed for new product and service development. Innovative companies have inquired about the opportunity for bringing new devices like displays, display systems, feedback units, cameras, sensors and service offerings to be tested in the LLB environment. Possible application areas beyond transportation have also been identified such as utilizing the LLB fleet as an environmental monitoring platform. By attaching normal and hyperspectral cameras on dashboards and using experimental software in OMEC units it might be possible to monitor for example snow conditions during winters and need for plant irrigation during summer times or using sensors outside the bus to monitor air quality in a city.

As the amount of new sensors and actuator combinations in LLB environment is grows and the heterogeneity of installed devices diversifies, the need for resource configuration management system also increases. The next logical step is to implement a resource and configuration management support system for keeping in the pace with constantly evolving assets.

Conclusions

This paper has described the available living lab platform for developing and testing services and technologies in a public transport context. It is an ongoing work of the Living Lab Bus project to provide an open innovation platform to speed up the development, demonstration and deployment of value adding products and services that improve public transportation – whether it is vehicle and operation efficiency, technology innovations, attractive design or infotainment services.. The focus of this paper was on the LLB platform architecture and technical details that support changing hardware and software modifications based on the ecosystem parties' new and diverse requirements.

The LLB system has to ensure sufficient functional flexibility to respond to evolving needs and to assure overall security despite the system's openness. A fog computing architecture approach was selected to achieve these aims. It provides needed elasticity of resources and decreases unnecessary data traffic and thus latency between OMEC units and the back end cloud service, increasing reliability and quality of applications and services. The fog architecture approach also helps to enhance the modularity and extendibility of the system to support bringing in new sensor devices, data sources and actuators to the system in a flexible way.

The implemented LLB platform is a mixture of commercially available off-the-shelf (COTS) and prototype components and software developed by VTT, Finnish universities and private companies. Being an evolving environment with new parties and ideas coming in gradually, it poses challenges for designing and implementing suitable and fit-for-purpose hardware and solutions for hosting 3rd party software reliably and cost efficiently. The most time consuming efforts related to physical changes in the LLB environment is adjusting wirings (cabling). Therefore, the ease and rate of implementing new ideas varies significantly based on the type of solution, ranging from devices requiring a power source to simply accessing data through an API. While the platform at this point is mainly used for developing and testing individual solutions, it also provides an opportunity for considering more comprehensive, modular vehicle information systems. For this, future research could include for example public transport on-board IT system standardization efforts such as the Information Technology for Public Transportation (ITxPT) initiative to ease and ensure practical device connectivity.

References

Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer networks*, *54*(15), 2787-2805.

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing* (pp. 13-16). ACM.

Bonomi, F., Milito, R., Natarajan, P., & Zhu, J. (2014). Fog computing: A platform for internet of things and analytics. In *Big Data and Internet of Things: A Roadmap for Smart Environments* (pp. 169-186). Springer International Publishing.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, *29*(7), 1645-1660.

ITxPT. Information Technology for Public Transportation. http://itxpt.org/

Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012). Future internet: the internet of things architecture, possible applications and key challenges. In *Frontiers of Information Technology (FIT), 2012 10th International Conference on* (pp. 257-260). IEEE.

Kostiainen, J., Hautala, R., Hinkka, V. & Aapaoja, A. (2016). Setting up an ITS innovation environment with many stakeholders. 23rd World Congress on Intelligent Transport Systems, At Melbourne, Australia.

Schaffers, H., Komninos, N., Pallot, M., Trousse, B., Nilsson, M., & Oliveira, A. (2011). Smart cities and the future internet: Towards cooperation frameworks for open innovation. *The future internet*, 431-446.

Yi, S., Li, C., & Li, Q. (2015). A survey of fog computing: concepts, applications and issues. In *Proceedings of the 2015 Workshop on Mobile Big Data* (pp. 37-42). ACM.

Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things journal*, *1*(1), 22-32.