Challenges of DevOps ready IoT Testbed

1st Janis Judvaitis
Institute of Electronics and Computer Science
Riga, Latvia
janis.judvaitis@edi.lv

2nd Krisjanis Nesenbergs
Institute of Electronics and Computer Science
Riga, Latvia
krisjanis.nesenbergs@edi.lv

3rd Rihards Balass
Institute of Electronics and Computer Science
Riga, Latvia
rihards.balass@edi.lv

4th Modris Greitans
Institute of Electronics and Computer Science
Riga, Latvia
modris.greitans@edi.lv

***Abstract*—Developing, deploying, testing and validating complex software systems is complicated and as a result security, privacy and quality in general often suffer due to limited resources and rapid development cycles. In the case of IoT system development there are additional levels of complexity and risks. In this paper we present our ongoing effort towards providing and validating a solution for these problems as a toolset/TestBed implementing previously established ENACT IoT DevOps concepts and Framework, aimed at ensuring continued Quality of Service and application of best practices during development cycle of IoT systems.**

***Index Terms*—IoT TestBed; DevOps; Secure IoT;**

I. Introduction

Forces behind the rapid growth of the Internet-of-Things (IoT) market both in number of active devices[1] and market value[2] are at direct odds with security and privacy of said devices and their users. Companies rushing to carve out their own IoT market share are faced with the complex, dynamic and challenging realities of developing, testing and debugging these devices [1].

In order to counteract potential lack of such expensive development aspects as security, privacy and quality in general driven by these market forces an appropriate toolset is required for implementing best practices in development and related operations (DevOps). Such software engineering best practices and tools to ensure Quality of Service and ease of use while allowing continuous evolution of complex systems in an agile fashion with rapid innovation cycles are at the heart of the DevOps movement [2], and have a stable place in classic software development.

As such a solution for IoT was previously unavailable, ENACT DevOps (EDO) concept and Framework was born to facilitate creation and operation of trustworthy Smart IoT Systems [3]. This framework defines related research challenges, and proposes solutions in form of novel IoT platform enablers bundled in several toolkits for the whole Life-Cycle of Smart IoT System development (see Section II).

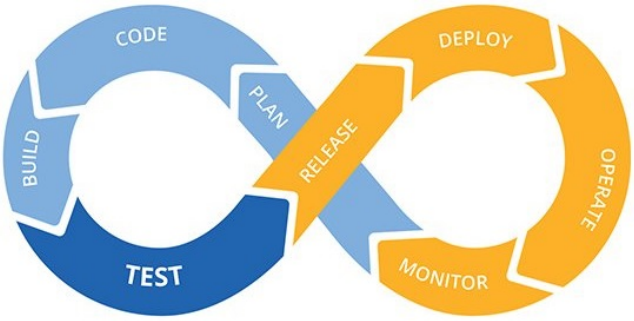

Fig. 1. Product life-cycle stages[3]

In this paper we describe our work towards one part of this framework - an IoT development TestBed implementing the EDO approach - from the scope and requirements to practical concerns and challenges.

The rest of the paper is structured as follows: Section II describes the Preliminary work and State of the Art, including general EDO Framework concepts and previous attempts at TestBed development by our team, Section III reveals specific DevOps TestBed related requirements, and Section IV concludes with future challenges and conclusions of the ongoing work to implement this DevOps enabling IoT TestBed.

II. Preliminary work and background

An in-depth description of the research challenges in IoT and how a specifically designed IoT DevOps framework might be beneficial has been previously published by Ferry et al. [3] presenting EDO Framework for development, operation and quality assurance of trustworthy Smart IoT Systems. A brief overview of the main points in this work follows, laying

The research leading to these results has received funding from the European Commissions H2020 Programme, grant agreement no. 780351 (ENACT)

[1]7B devices mid 2018, 10B projected by 2020 - https://iot-analytics.com/product/state-of-the-iot-2018/

[2]235B USD in 2017, 520B USD projected by 2021 - https://www.bain.com/insights/unlocking-opportunities-in-the-internet-of-things/

[3]http://www.bestdevops.com/has-devops-changed-the-role-of-a-tester/

ground for current ENACT project, followed by our previous IoT TestBed development work for context.

A. ENACT DevOps (EDO) framework

DevOps as a concept is is well established as an approach to ensure a rapid and efficient value delivery to market through tight collaboration between the developers (Dev) and the teams that deploy and operate the software systems (Ops), in essence to decrease the gap between product design and its operation by automation and continuous processes, supported by different tools at various stages of the product life-cycle.

In the EDO Framework a next step is taken to formalize this process for application in trustworthy Smart IoT Systems in spite of such IoT research challenges identified by Ferry et al. as: (i) Support for Continuous Delivery of trustworthy smart IoT systems; (ii) support for Agile Operation of trustworthy smart IoT systems; (iii) support for Continuous Quality Assurance strengthening trustworthiness of Smart IoT Systems; and (iv) leveraging capabilities of existing IoT platforms and fully exploiting legacy, proprietary and off-the-shelf software components and devices. Afterwards they present the EDO enablers relating to 8 stages of IoT development life-cycle as shown in Figure 1 and separated into three toolkits as follows:

ENACT Continuous Delivery Toolkit for agile and continuous evolution of IoT systems with early detection of issues in development process, consisting of (i) Orchestration and Continuous Deployment Enabler and (ii) Test, Emulation and Simulation Enabler.

ENACT Agile Operation Toolkit for ensuring automated operation of the developed systems and ensuring their trustworthiness during operation, consisting of (i) Context-Aware Self-adaptation Enabler, (ii) Root Cause Analysis Enabler and (iii) Context Monitoring and Actuation Conflict Management Enabler.

ENACT Trustworthiness Toolkit for crosscutting trustworthiness concerns of IoT systems (e.g. robustness, security and privacy), consisting of (i) Robustness and Resilience Enabler, (ii) Risk Management Enabler, and (iii) Security and Privacy Monitoring and Control Enabler.

These enablers are designed to be loosely coupled and in this paper we describe ongoing ENACT project work on providing a subset of these enablers to be integrated with existing IoT platforms in the form of a DevOps ready IoT TestBed. Specifically, existing TestBed is enriched and applied in use case for Intelligent Transport Systems, to assess the feasibility of IoT services in the domain of train integrity control for the logistics and maintenance of the rolling stock and on-track equipment.[4] The first iteration of on-board IoT system for train integrity control [4] was developed and demonstrated using EDI TestBed in DEWI project [5]. Afterwards, in ENACT project, a rework of DEWI ITS on-board IoT system was done introducing on-track IoT system satisfying EDO Framework needs.

[4]https://www.enact-project.eu/ucs.php

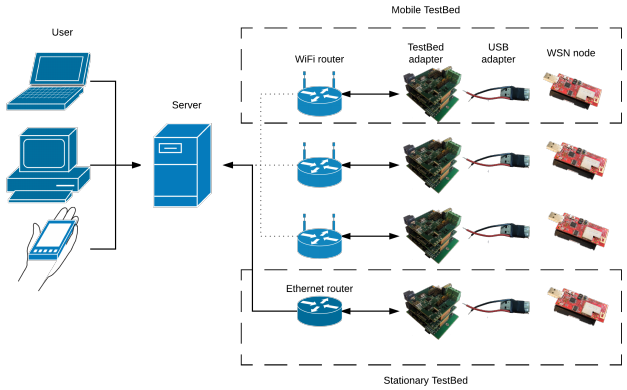

Fig. 2. EDI TestBed hardware architecture

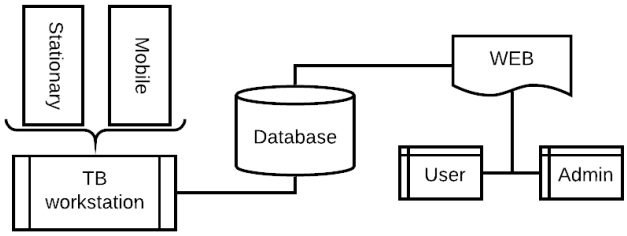

Fig. 3. EDI TestBed software architecture

B. EDI TestBed

EDI TestBed [6] is a set of debugging tools that help increase the development speed of IoT and WSN systems. The TestBed is comprised of 100+ nodes that are distributed around a 7 floor building for validation and research in sensor network and wireless network protocols. The current hardware and software architectures of EDI TestBed can be seen in Figures 2 and 3.

Currently the EDI TestBed has no tools assisting with DevOps integration and there is only very basic support functionality required from a DevOps ready TestBed: (i) support for heterogeneous IoT network, (ii) basic logging functionality, (iii) remote device access for reprogramming and bi-directional serial communication(UART), (iv) remote power supply control for controlled power consumption measurements and emulation of supply voltage changes, (v) analog interface simulation and emulation and (vi) mobile TestBed workstations.

To develop a DevOps ready IoT testbed on this basis, a set of requirements are defined in the next section.

III. DevOps TestBed requirements

A. Testbed role in EDO framework

Although an IoT testbed as such could potentially provide a wide range of services and features, the emphasis must be on the tools provided to support efficient development

and operations of trustworthy smart IoT systems. Thus, the envisioned DevOps ready IoT development TestBed is aimed to implement the first of three EDO toolkits described in Section II-A - the ENACT Continuous Delivery Toolkit. This means, that it must be an enabling technology in orchestration and continuous deployment as well as testing, emulation and simulation of smart IoT systems during their development.

B. Continuous delivery toolkit related requirements

Even though there are requirements related to specific enablers within the toolkit described in the next subsections, there are some non-enabler-specific requirements as well, specifically:

(i) Toolkit modularity, supported by a robust messaging backbone and API - as all of the ENACT toolkits must contain loosely coupled enablers, the related DevOps services provided by the TestBed must also be modular, interchangeable and easy to extend in the future. Specifically:

Message brokering system - the current TestBed infrastructure is a monolith system with multiple task specific communication channels, leading to DevOps problems with the Testbed infrastructure itself and limiting the rate at which new needed modules can be developed. A messaging system and broker can take care of this decoupling, and provide an easy plug-and-play ability to add such critical modules as complete system state logging, target IoT device output logging as well as promote easy scaling of the TestBed system for use in development and testing of larger IoT networks;

Standardized low level API - the messaging system on its own can manage passing of required messages between the modular components of the TestBed, such as hardware testing endpoints, logging servers or user stations, but the types of messages must still be declared in a strong API, so that introduction of new modules does not require a complete rework of the existing communications protocols.

(ii) Hardware invariant abstraction, supporting heterogeneity - the final toolkit needs to support development and testing of complex networks consisting of different types of node hardware, as well as different types of TestBed endpoint hardware (e.g. static endpoint, mobile endpoint, endpoint with software defined radio, endpoint with extremely precise energy measurement capabilities etc.) and for the DevOps processes to be usable, the end user must not be burdened with the technical differences in this hardware, but instead should work with standardized abstractions. Specifically:

TestBed architecture with heterogeneity built in from the start - all other requirements must be considered in the lens of heterogeneity with reliable and expandable detection of devices, API with gracious fall-backs in case of unavailable specific hardware features etc., leading to capability to encourage development of more complex heterogeneous trustworthy IoT networks;

Seamless bi-directional serial communications (UART) support - UART protocol is ubiquitous in IoT devices and should be a first-class citizen in the TestBed hardware environment and UART messaging to and form all TestBed hardware components must be transparently integrated in the system allowing like-a-local work flow with hardware;

Available remote micro controller debugging capabilities - development and testing requires low level debugging facilities, which are often unavailable in remote abstracted systems. A quality DevOps enabled Testbed needs such capabilities to support agile and continuous evolution and to make it easy to identify the source of many problems.

(iii) Easy integration with existing tool chains - in the most basic abstraction TestBed should be one of many tools used in trustworthy smart IoT system DevOps life cycle, so accordingly it should provide the users with interface of a basic tool set, which can be easily integrated in any existing work flow as necessary, including:

Well rounded command line interface (CLI) - interactive web based interface of current TestBed version is not easily integrated into existing development pipelines, thus users cannot deploy their code to IoT nodes, debug, test, validate and script them in continuous manner using classic DevOps tools and their robust CLI interfaces minimizing the gap between trustworthy IoT and traditional system development life cycles. Thus high level integrated development environments (IDE) could interact with planned TestBed CLI tools via plugins for complete IoT smart system life cycle integration;

Seamless remote back-end tools - to make the EDI TestBed more DevOps friendly a background daemon must be introduced, which provides full EDI TestBed functionality to local machine or network for use on actual work space, supporting not only CLI, but also local-like interaction with the remote hardware, e.g. reverse control through GPIO for event detection and precise timing, ground truth data such as precise time synchronization, possible radio channel routes, and logic analyzer functionality for advanced IoT device behaviour analysis and debugging, protocol decoders for most popular protocols used in IoT (SPI, I2C UART).

C. Orchestration and Continuous Deployment enabler related requirements

TestBed should conform to several requirements related to facilitating engineering and continuous development in a decentralized way:

Remote simultaneous reprogramming of target IoT devices - the most basic requirement of a TestBed enabling continuous delivery is to allow centralized mass control and reprogramming of the target IoT network. This must be supported by a robust API as mentioned above.

Automated hardware unit testing support - for early detection of issues in software development process, easy testing is required - ability to run tests on end nodes each time they are reprogrammed, comparing the results to some baseline and reporting errors/inconsistencies together with relevant data about this specific node.

Precise power measurements - as IoT devices use battery power it is critical to develop them with optimal power consumption in mind. To develop and continually improve the power consumption of the system under development,

precise power consumption data must be available in real-time resulting in ability to differentiate between the costs of ”smart” and ”trustworthy” in terms of power consumption to make better trade-off decisions.

D. Test, Emulation and Simulation enabler related requirements

To assess the behaviour and trustworthiness of the system during its life-cycle TestBed must provide:

System testing support - like with hardware unit testing in the case of continuous deployment, automated system tests should also be supported with plug-in type smart test result analysis highlighting the differences between different runs of tests or a specified baseline.

Testing and verification continuum - to ease the gradual migration from test to operational environment the TestBed should provide the functionality to seamlessly transfer IoT devices from local tests/validation to target location using mobile TestBed workstations using identical work flow on local nodes and remote nodes.

Real-Virtual radio interface - for performance assessment while the system is (a) not complete or (b) completely located in TestBed, special endpoints (workstations) with software defined radio (SDR) [7] should be added to TestBed, providing radio connectivity (a) between real and simulated nodes and (b) between nodes located in different physical locations (e.g. local TestBed and forest etc.). This enables decentralized processing through SDR channel allowing part of unoptimized algorithms to be run on servers while developing IoT devices.

Precise power control - to test IoT systems in close to real physical environment TestBed should provide functionality of power supply control simulating battery discharge or power drop due to ambient temperature changes etc.

Automated recorded, simulated and physical security tests - IoT system trustworthiness testing through automated basic security attacks, simulated threat actors and physical access attacks (both analog and digital) should be provided for advanced security analysis of IoT device as a separate entity.

IV. Conclusion

In this paper we have provided a road map and related practical concerns and challenges in development of a DevOps ready IoT testbed, that conforms to EDO approach using the existing EDI TestBed as baseline.

Summary of features to be added include:

- Robust messaging backbone (using messaging broker like MQTT);
- Standardized TestBed low-level API providing seamless access to back-end tools;
- Well rounded CLI for integration in existing tool-chains;
- Complete system state and output logging through plug-ins;
- Remote micro controller debugging (GDB like) functionality;
- Hardware update with SDR nodes and related software for real to virtual communication;

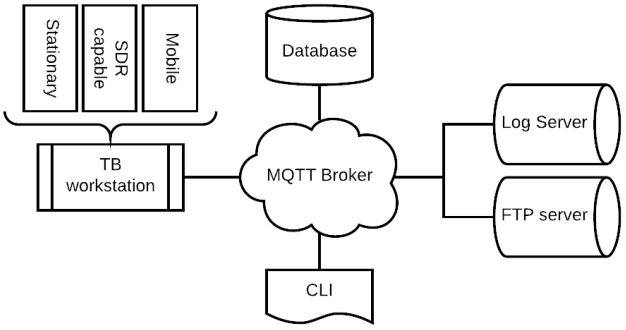

Fig. 4. EDI TestBed next version architecture blocks

- Automated hardware unit testing, including (i) automated system tests on specific events like reprogramming and (ii) smart test result analysis.
- Analog and digital interface simulation, including (i) automated basic security attacks and (ii) simulated possible security threat actors.
- Easy local interaction with remote TestBed hardware, including (i) reverse control through GPIO, (ii) ground truth data, (iii) direct UART link and (iv) logic analyzer, with these functions: (i) automated protocol decoding and (ii) precise event timing between different workstations.

Even though much of the TestBed hardware can be reused, the software part must be completely rebuilt - the planned system schematic can be seen in figure 4. The work of implementing this TestBed as an ENACT Continuous Delivery Toolkit is currently underway and the applicability of the result to enabling DevOps for trustworthy smart IoT system development will be evaluated using ENACT ITS use case and expanded as necessary based on these results.

References

[1] A. Taivalsaari and T. Mikkonen, “A roadmap to the programmable world: software challenges in the iot era,” *IEEE Software*, vol. 34, no. 1, pp. 72–80, 2017.

[2] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, “Devops,” *IEEE Software*, vol. 33, no. 3, pp. 94–100, 2016.

[3] N. Ferry, A. Solberg, H. Song, S. Lavirotte, J.-Y. Tigli, T. Winter, V. Muntés-Mulero, A. Metzger, E. R. Velasco, and A. C. Aguirre, “Enact: Development, operation, and quality assurance of trustworthy smart iot systems,” in *International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*. Springer, 2018, pp. 112–127.

[4] N. Barkovskis, A. Salmins, K. Ozols, M. A. M. García, and F. P. Ayuso, “Wsn based on accelerometer, gps and rssi measurements for train integrity monitoring,” in *2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2017, pp. 0662–0667.

[5] W. Rom, P. Priller, J. Koivusaari, M. Komi, R. Robles, L. Dominguez, J. Rivilla, and W. Van Driel, “Dewi–wirelessly into the future,” in *2015 Euromicro Conference on Digital System Design*. IEEE, 2015, pp. 730–739.

[6] R. Ruskuls, D. Lapsa, and L. Selavo, “Edi wsn testbed: Multifunctional, 3d wireless sensor network testbed,” in *Advances in Wireless and Optical Communications (RTUWO), 2015*. IEEE, 2015, pp. 50–53.

[7] F. K. Jondral, “Software-defined radio: basics and evolution to cognitive radio,” *EURASIP journal on wireless communications and networking*, vol. 2005, no. 3, pp. 275–283, 2005.