View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Electronic Communications of the EASST (European Association of Software Science and...

Electronic Communications of the EASST
Volume 080 (2021)

Conference on Networked Systems 2021 (NetSys 2021)

Demonstration: A cloud-native digital twin with adaptive cloud-based control and intrusion detection

William Tärneberg, Martin Gunnarsson, Maria Kihl, Christian Gehrmann

4 pages

Guest Editors: Andreas Blenk, Mathias Fischer, Stefan Fischer, Horst Hellbrueck, Oliver Hohlfeld, Andreas Kassler, Koojana Kuladinithi, Winfried Lamersdorf, Olaf Landsiedel, Andreas Timm-Giel, Alexey Vinel

ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Demonstration: A cloud-native digital twin with adaptive cloud-based control and intrusion detection

William Tärneberg[1], **Martin Gunnarsson**[2 1], **Maria Kihl**[1], **Christian Gehrmann**[1]

[1] Dept. of Electrical and Information Technology at Lund University, Lund, Sweden
[2] RISE Cybersecurity, RISE Research Institutes of Sweden, Lund, Sweden

Abstract: Digital twins are taking a central role in the industry 4.0 narrative. However, they are still illusive. Many aspects of the digital-twins have yet to materialize. For example, to what degree will they be integrated into cloud and industry 4.0 systems as well as how and if they should augment their physical counterpart. Those choices are accompanied by challenging security aspects, many of which have to be studied partially. In this paper, we present a novel digital-twin demonstrator that enables experimentation and advanced research on such systems. The demonstrator is cloud-native, has a distributed adaptive control system, incorporates edge and public clouds, a PLC, intrusion detection, a wireless network emulator, and an attacker.

Keywords: Digital-Twin, Cloud-native, Cyber security, Intrusion Detection, Test-bed, Wireless emulation, Feedback control, Distributed computing

1 Introduction

Digital Twins (DTs) have become integral to the Industry 4.0 narrative and have come to incorporate more than telemetry of sensor values. The relationship between the Physcial Twin (PT) and the DT has crystallized, and the cloud has become the de-facto deployment environment for DTs. In recent development [TSGK20], DTs have also been proposed to host augmenting cloud-based controllers [STÅK20]. The opportunity is such that a controller in the DT has access to abundant computational capacity, has deep access to the state of the PT, and can be manipulated and adapted in run time.

Exposing telemetry and models through a DT carries its risks, which has been documented [GG20]. Further, exposing controllers through the digital twin is a novel idea that introduces new challenges and opportunities, particularly the detection of an intrusion by malicious actors.

This incorporation of cloud-based controllers and intrusion detection in digital twins is a novel topic in the research community. Challenges remain in formalizing as well as in realizing such systems in a robust manner. Such systems-of-systems have many unforeseen dynamics and can not be effectively studied in simulators or as a hypothetical.

In this paper, we present a fully functioning cloud-native test-bed for studying such DT systems. The test-bed includes: a Cyber-Physical System (CPS), a wireless network emulator, a Programmable Logic Controller (PLC), an intruder agent, a Kubernetes (K8S)-cluster that hosts a DT, intrusion detection, and a distributed cloud-based controller that spans the CPS/PT, the DT, and a set of remote cloud resources. The resulting test-bed can demonstrate the end-to-end realization of a cloud-native DT system and allows, for example, the experimentation of security aspects and stability issues arising from network and cloud variability.

2 Test-bed implementation

In this section, we present a Commercial off-the-shelf (COTS)-based DT test-bed, which is a significant extension to the Industry-4.0 test-bed presented in [STÅK18]. A high-level overview of the system is illustrated in Figure 1. The test-bed incorporates: a physical CPS with an uncomplicated CPS, an edge K8S cluster, COTS-based PT and DT, an intruder, a network emulator, a PLC, and a control system that spans the production process and the edge cluster. The system's components are detailed below.

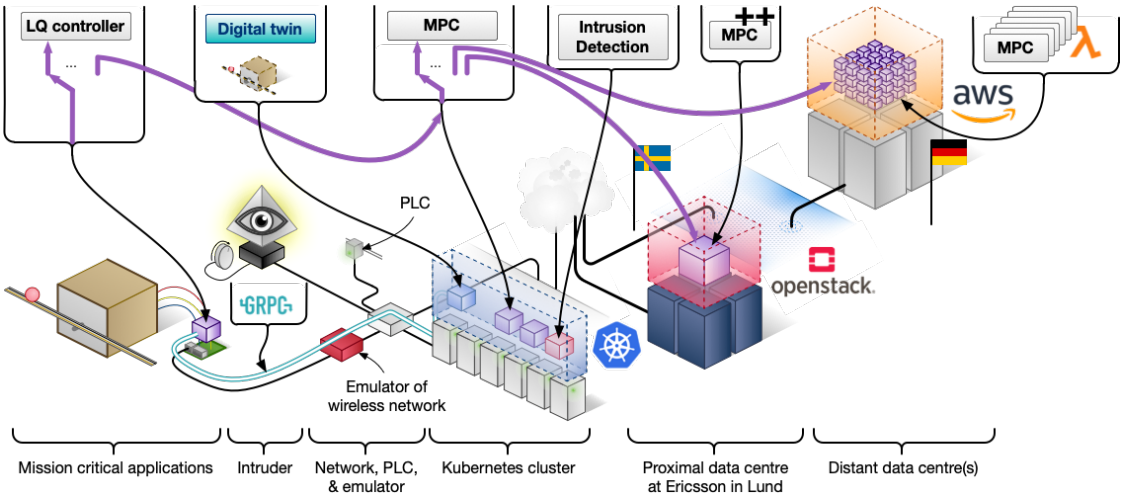

Figure 1: A high level overview of the proposed test-bed, composed of a physical CPS and an edge DC that hosts a DT, which hosts a feedback control system as well as an intrusion detector and mitigation strategy.

The test-bed represents a slice of a vertically integrated CPS, with a critical application connected to local and remote cloud services, as pictured in the Industry-4.0 reference architecture [HR15].

2.1 Cloud infrastructure

The edge cloud is six-node K8S cluster proximal to the PT. K8S is an open-source container orchestration platform, often viewed as the de-facto industry standard. Additionally, K8S is extensible though a large number of open-source projects ranging from security to tracing. The six nodes of the cluster are identical, unimposing, but sufficient COTS desktop PCs, similar to what one might find in a corporate Information and Communications Technology (ICT) infrastructure. The nodes solely host the cluster and they are connected over a solitary COTS 1000BASE-T network. Further, the cluster is shared with other tenants and their processes. The cluster has been equipped with an nginx ingress and prometheus operator. The nginx ingress is exposed using the K8S NodePort paradigm. Storage is realized with Rook. In addition to the edge cloud, the test-bed also includes a set of public cloud resources.

2.2 Physcial Twin/Digital Twin pair

The Digital Twin (DT) and the PT are implemented in Python. The DT is hosted on the K8S cluster, and is thus packaged as a docker container. The PT is deployed to a CPS-side Raspberry Pi that can interface with the physical CPS. The DT implementation has five primary functions: 1. communication gateway to the CPS 2. CPS state-store 3. feedback controller augmentation 4. input augmentation 5. intrusion detection and mitigation

All communication between the PT and the DT is over a Local Area Network (LAN) and uses a protocol defined in Protocol Buffers (Protobuf) [1] which is realized in gRPC [2]. As specified in [GG20] all communication with the CPS from the outside world goes through the DT. This communications includes, for example, new configurations and health checks. The state of the CPS is synced periodically to the DT, using the protocol specified in [GG20].

The DT controller augments the local controller at the CPS. The choice of controller can be made dynamically at the CPS in a manner presented in [SWÅK20] or be set externally through the communication gateway. The controllers are also implemented in Python.

2.3 CPS and controllers

The CPS is a simple mechanical contraption, the *ball and beam* process. The ball and beam is a classic feedback control system that has a simple and visible objective. The process has a well-described model [SEKÅ19] and it requires timely feedback to remain stable, must be operated at reasonably high frequency, and it is a naturally constrained problem. These properties are well suited for our targeted evaluation. The CPS is connected to a Raspberry Pi which is connected to the internet and the LAN.

The test-bed incorporates two controllers as enabled by the system model presented in Section 2. The PT controller is implemented as a Linear–quadratic regulator (LQR) while the DT is implemented as a Model Predictive Control (MPC). An MPC is a more capable controller of the two. We refer to [SWÅK20] for a full detail of the controller implementations. The CPS can be successfully controlled in the range 5 Hz to 40 Hz. The PT dynamically adapts the DT controller to match the current state of the infrastructure. In fact, the PT can deploy and use a controller in the public cloud resources. In addition to the PT and DT controllers, there is a PLC able to control the plant. The PLC is connected to the plant over the LAN.

2.4 Attacker and intrusion detection

Connected to the network there is a Raspberry Pi that acts as an attacker. The attack is targeted at the control signal to and from the plant. The attack is such that it adds a variable noise to one or more signals. The design of an attack can be specified by the user of the system.

The DT is equipped with an intrusion detector. It is able to expediently detect attacks and mitigate them, by reverting to the ancillary controller in the PT. Once the attack subsides, the PT switches back to the more capable controller in the DT.

[1] https://developers.google.com/protocol-buffers/
[2] https://grpc.io/

2.5 Network emulator

To allow for experimentation of the inclusion of wireless connectivity, a network emulator sits between the PT and the rest of the infrastructure. The emulator, emulates bit errors, given a user-specified channel model, and signal strength.

2.6 User interface

The inner workings of DT are visualized through Grafana dashboard. A light on the PT indicates when it is being intruded upon. Additionally, the PT has a screen that shows the state of the PT and which controller it is currently employing. Furthermore, the network emulator has a web interface, that allows for the input of user-specified settings. The nature and intensity of an attack is set using a rotary dial on the attack node.

Acknowledgements: This work has been partially funded by the Wallenberg AI, Autonomous Systems and Software Program (WASP), the ELLIIT strategic research area on IT and mobile communications, Sweden's Innovation Agency (VINNOVA) under the 5G-PERFECTA Celtic Next project, the Swedish Foundation for Strategic Research under the SEC4FACTORY project.

Bibliography

[GG20] C. Gehrmann, M. Gunnarsson. A Digital Twin Based Industrial Automation and Control System Security Architecture. *IEEE Transactions on Industrial Informatics* 16(1):669–680, 2020.

[HR15] M. Hankel, B. Rexroth. The reference architectural model industrie 4.0 (rami 4.0). *ZVEI* 2(2):4–9, 2015.

[SEKÅ19] P. Skarin, J. Eker, M. Kihl, K.-E. Årzén. Cloud-Assisted Model Predictive Control. In *International Conference on Edge Computing*. 2019.

[STÅK18] P. Skarin, W. Tärneberg, K.-E. Årzén, M. Kihl. Towards Mission-Critical Control at the Edge and Over 5G. In *International Conference on Edge Computing*. IEEE, 2018.

[STÅK20] P. Skarin, W. Tärneberg, K.-E. Årzén, M. Kihl. Control-over-the-cloud: A performance study for cloud-native, critical control systems. In *International Conference on Utility and Cloud Computing (UCC)*. 2020.

[SWÅK20] P. Skarin, T. William, K.-E. Årzén, M. Kihl. Control-over-the-cloud: A performance study for cloud-native, critical control systems. In *Accepted to UCC*. 2020.

[TSGK20] W. Tärneberg, P. Skarin, C. Gehrmann, M. Kihl. Prototyping intrusion detection in an industrial cloud-native digital twin. In *Submitted to International Conference on Industrial Technology*. 2020.