Project Title	AN OPEN, TRUSTED FOG COMPUTING PLATFORM FACILITATING THE DEPLOYMENT, ORCHESTRATION AND MANAGEMENT OF SCALABLE, HETEROGENEOUS AND SECURE IOT SERVICES AND CROSS-CLOUD APPS
Project Acronym	RAINBOW
Grant Agreement No	871403
Instrument	Research and Innovation action
Call / Topic	H2020-ICT-2019-2020 / Cloud Computing
Start Date of Project	01/01/2020
Duration of Project	36 months

D7.5 – Standardization Activities Report – Final

Work Package	WP7 – Dissemination, Exploitation and Communication
Lead Author (Org)	Thomas Pusztai (TUW)
Contributing Author(s) (Org)	Frank Scherber (IFAT), Panagiotis Gouvas (UBI), Heini Bergsson Debes (DTU), Marco Rapelli (POLITO)
Due Date	31.12.2022
Actual Date of Submission	30.12.2022
Version	1.0

Dissemination Level

X	PU: Public (*on-line platform)
	PP: Restricted to other programme participants (including the Commission)
	RE: Restricted to a group specified by the consortium (including the Commission)
	CO: Confidential, only for members of the consortium (including the Commission)

The work described in this document has been conducted within the project RAINBOW. This project has received funding from the European Union's Horizon 2020 (H2020) research and innovation programme under the Grant Agreement no 871403. This document does not represent the opinion of the European Union, and the European Union is not responsible for any use that might be made of such content.

Versioning and contribution history

Version	Date	Author	Notes
0.1	11.11.2022	Thomas Pusztai (TUW)	Table of Contents and initial content
0.2	09.12.2022	Frank Scherber (IFAT)	Write Section 5
0.3	14.12.2022	Thomas Pusztai (TUW)	Write Section 3
0.4	15.12.2022	Thomas Pusztai (TUW)	Write Sections 1, 2, 4, and 7. Document ready for internal review.
0.5	19.12.2022	Heini Bergsson Debes (DTU), Marco Rapelli (POLITO)	Review document and make minor changes
0.6	21.12.2022	Thomas Pusztai (TUW)	Incorporate reviewers' suggestions
0.7	22.12.2022	Panagiotis Gouvas (UBI), Thomas Pusztai (TUW)	Write Section 6
0.9	23.12.2022	Frank Scherber (IFAT), Marco Rapelli (POLITO), Thomas Pusztai (TUW)	Final review and minor changes
1.0	30.12.2022	Christina Stratigaki (UBI)	QA Review and Submission

Disclaimer

This document contains material and information that is proprietary and confidential to the RAINBOW Consortium and may not be copied, reproduced or modified in whole or in part for any purpose without the prior written consent of the RAINBOW Consortium

Despite the material and information contained in this document is considered to be precise and accurate, neither the Project Coordinator, nor any partner of the RAINBOW Consortium nor any individual acting on behalf of any of the partners of the RAINBOW Consortium make any warranty or representation whatsoever, express or implied, with respect to the use of the material, information, method or process disclosed in this document, including merchantability and fitness for a particular purpose or that such use does not infringe or interfere with privately owned rights.

In addition, neither the Project Coordinator, nor any partner of the RAINBOW Consortium nor any individual acting on behalf of any of the partners of the RAINBOW Consortium shall be liable for any direct, indirect or consequential loss, damage, claim or expense arising out of or in connection with any information, material, advice, inaccuracy or omission contained in this document.

Table of Contents

List of Abbreviations

Abbreviation	**Definition**
BoD	Board of Directors
CCGRID	IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing
CPC	Certification Program Committee
DAA	Direct Anonymous Attestation
IoT	Internet of Things
K8S	Kubernetes
LF Edge	Linux Foundation Edge
OASIS	Organization for the Advancement of Structured Information Standards
RIM	Reference Integrity Manifest
TC	Trusted Computing
TCG	Trusted Computing Group
TNC	Trusted Network Communications
TOSCA	Topology and Orchestration Specification for Cloud Applications
TPM	Trusted Platform Module
TSC	Technical Steering Committee
TSS	TPM Software Stack

Executive Summary

This document contains a comprehensive report of RAINBOW's standardization activities conducted throughout the duration of the project. It first describes the goals of the standardization activities, as well as the working groups targeted, i.e., the Open Horizon and Centaurus projects, the Trusted Computing Group, and the Eclipse IoT community. Then, for each working group, an overview of its purpose and goals is provided, followed by RAINBOW's contributions, and the activities undertaken.

This document extends and replaces D7.4 "Standardization Activities Report - Version 1".

1 Introduction

The present deliverable is prepared in the context of Work Package 7 "Dissemination, Exploitation and Communication" and it is associated with the work being done within task T7.2 "Standardization Activities".

This deliverable extends and replaces D7.4 "Standardization Activities Report - Version 1" and complements D7.3 "Open-Source Contributions, Dissemination, Clustering and Workshop Activities Report - Final" led by AUTH, by providing a comprehensive overview of the standardization activities that have been undertaken as part of RAINBOW.

These activities include contributions to the large-scale and industry oriented Open Horizon and Centaurus projects by TUW, the Trusted Computing Group by IFAT, and the submission of an Eclipse IoT project proposal by UBI.

The remainder of this report is structured as follows: Section 2 outlines the goal of the standardization activities and lists the target standardization groups. Sections 3, 4, and 5 discuss the target groups, our contributions, and the activities that have been undertaken. Section 6 describes the Eclipse IoT project proposal initiated by RAINBOW and Section 7 concludes this report.

2 Standardization Strategy

2.1 Goal

The goal of the RAINBOW standardization activities is to set a standard in the development of fog computing architectures of future applications and allow them to benefit from the outcomes of RAINBOW. There are multiple ways to achieve this goal. One option is the direct contribution to relevant standards, such as OASIS' Topology and Orchestration Specification for Cloud Applications (TOSCA) or the Trusted Computing Group's Trusted Platform Module (TPM) standard, while another option is to contribute to software platforms that will likely be used as a base for creating a large number of fog applications in the future, making them similar to a de-facto industry standard. For RAINBOW, we have pursued both options.

2.2 Target Standardization Groups

The RAINBOW DoA states the goal of contributing to three standardization groups. These were originally defined as OASIS TOSCA, TCG, and a third one to be chosen after starting the project.

We have chosen to target the following groups as part of the RAINBOW standardization activities:

- Open Horizon[1] software platform for managing containerized workloads on the network edge (see Section 3). Originally started by IBM, Open Horizon is now part of the Linux Foundation Edge and has the potential for widespread adoption in edge/fog applications.
- Centaurus[2] is an open-source platform for building private and public Cloud and Edge infrastructures (see Section 4). Originally started by Futurewei Technologies, Centaurus is now part of the Linux Foundation and has the potential for adoption by large Cloud/Edge providers.
- Trusted Computing Group (TCG)[3], which is the consortium responsible for the Trusted Platform Module (TPM) standard (see Section 5).

Originally, OASIS TOSCA[4] was supposed to be targeted. It is an open standard for modelling the topology of cloud applications, including the relationships between their components and the management processes involved. The original plan for RAINBOW was to extend TOSCA for the fog, thus making an official contribution to this standard a logical step. However, we decided to take a Kubernetes-native approach for the design and development of RAINBOW. Kubernetes is one of the most popular container orchestration systems in production use, which means that support for this platform

[1] https://www.lfedge.org/projects/openhorizon/
[2] https://www.centauruscloud.io
[3] https://trustedcomputinggroup.org
[4] https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca

leads to higher chances of adoption of RAINBOW by the open source and industrial community.

Since RAINBOW does not make use of TOSCA, it no longer makes sense to contribute to this standard, leading to the exclusion of TOSCA from our list of target standardization groups. Instead, we have decided to focus more on the Open Horizon and Centaurus platforms, because of their potential to make a large impact in the near future.

An additional goal of RAINBOW, as demanded by the DoA, is the submission of a project proposal to the Eclipse IoT community. The efforts undertaken in this respect are described in Section 6.

3 Open Horizon

3.1 Project Overview

Open Horizon[5] is an open-source platform for managing containerized application deployments and machine learning on edge nodes. The main goal is managing the lifecycle of applications without administrator intervention by finding the most appropriate devices for their execution and managing updates of these applications. Furthermore, Open Horizon enables an application on one device to use capabilities of other devices.

The project was initiated by IBM and contributed to Linux Foundation Edge (LF Edge) in April 2020. Within LF Edge, Open Horizon already has many synergies with other projects. LF Edge classifies it as a Stage 2 "Growth" project, i.e., a project that is likely to have an impact, while offering favourable possibilities for RAINBOW to contribute [1] [2].

3.2 RAINBOW Contribution

Our intent was to donate RAINBOW components, after suitable modification, to the Open Horizon platform, with the goal that these components should outlive the RAINBOW project and be used in production applications.

Since Open Horizon is part of LF Edge and supported by IBM, it has the potential for becoming very influential in the edge/fog community. Thus, a contribution to this project, would influence a large number of applications, which could be created based on Open Horizon, leading to a direct impact of RAINBOW in production-level applications.

At the initial presentation of the RAINBOW platform to the Open Horizon Technical Steering Committee on March 29, 2021, Open Horizon expressed interest to evaluate the inclusion of RAINBOW components in Open Horizon if the RAINBOW consortium could guarantee long term maintenance of these components.

After the second release of the RAINBOW platform, the RAINBOW consortium decided that the following two components are the most suitable for contribution to Open Horizon:

1. The Fogify simulator for facilitating testing of some Open Horizon scenarios.
2. The Polaris SLO Framework for implementing complex SLOs within Open Horizon.

[5] https://www.lfedge.org/projects/openhorizon/

3.3 Activity Report

TUW has established contact to Open Horizon's Technical Steering Committee (TSC) in February 2021.

On 29 March 2021 TUW and UCY presented[6] the RAINBOW platform and the contribution candidate components to the Open Horizon TSC [3]. The presentation and discussion have sparked interest in the RAINBOW platform and the suggested contributions. Specifically, they expressed the following interests:

- The RAINBOW Sidecar Proxy for bootstrapping Open Horizon services on edge nodes.
- The RAINBOW Monitoring service for gathering and analysing metrics about running applications on an Open Horizon cluster. This is an aspect that is currently missing from the Open Horizon platform.
- The Fogify fog simulator was identified as a possible additional contribution, if the other contributions are successful.

The Open Horizon TSC has remarked that a condition for contributions to their project, is the availability of support for the donated components not only for the duration of RAINBOW, but also after the RAINBOW project ends. For this reason, we decided after the second release of RAINBOW, to offer the following two components, which are going to remain in the research focus of their respective authors:

1. Fogify simulator by UCY
2. Polaris SLO Framework by TUW

These components were officially offered[7] to Open Horizon on November 18, 2022. Their final decision is pending as of the writing of this deliverable.

[6] The presentation can be found in Annex I – Communication with Open Horizon.

[7] See Annex I – Communication with Open Horizon

4 Centaurus

4.1 Project Overview

Centaurus[8] is a novel open-source platform for building unified and highly scalable public or private distributed Cloud and Edge systems. It was started by Futurewei Technologies and donated to the Linux Foundation. Futurewei remains the largest contributor and driver behind the project. The project consists of multiple SIGs and subprojects, whose overall aim is to allow setup and management of large-scale Cloud-Edge infrastructures. One of its main components is Arktos [4], which extends Kubernetes to provide multitenancy, unified management of containers and VMs, and support for up to 50,000 nodes. Other subprojects include Mizar, a high-performance Cloud network interface and Fornax, a fault tolerant Edge computing framework.

Contributing to this project offers a valuable opportunity for RAINBOW to include some of its technology in a Cloud-Edge platform with a very high potential of making an impact on industry in the near to mid future.

4.2 RAINBOW Contribution

The original scheduler of Centaurus is a modified version of the Kubernetes default scheduler tuned for scalability, however, without any Edge- or SLO-awareness. This is a gap that the RAINBOW Scheduler [5] [6] is capable of filling. Thus, it was proposed for forking into the Centaurus project.

4.3 Activity Report

TUW has been cooperating with Futurewei since early 2020 and it became an official member of the Centaurus project. At the end of 2021 we proposed our scheduler contribution to Centaurus and it was quickly accepted. With the latest release of the RAINBOW scheduler, the project was ready for being contributed. The scheduler was forked[9] into the Centaurus GitHub organization on December 14, 2022.

TUW actively continues scheduler development under and has evolved it into an orchestrator-independent, distributed scheduler with support for multiple clusters. A scientific publication on this work has been submitted to the IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID) 2023 in December 2022.

[8] https://www.centauruscloud.io
[9] https://github.com/CentaurusInfra/polaris-scheduler

5 Trusted Computing Group

5.1 Group Overview

The Trusted Computing Group (TCG)[10] is a not-for-profit organization formed to develop, define and promote open, vendor-neutral, global industry specifications and standards, supportive of a hardware-based root of trust, for interoperable trusted computing platforms. TCG's core technologies include specifications and standards for the Trusted Platform Module (TPM), Trusted Network Communications (TNC) network security and self-encrypting drives. TCG also has working groups to extend core concepts of

- trust into fog and cloud security,
- virtualization and other platforms,
- computing services from the enterprise to the Internet of Things.

5.2 RAINBOW Contribution

The TCG is the internationally accepted standardization group, which sets all relevant technologies and standards for Trusted Computing (TC).

RAINBOW is working on and using these basic TC standards for building trusted systems. Therefore, both the RAINBOW project and TCG benefit from the continuous exchange of standardization and background information.

Infineon's Security business unit is market leader for semiconductor-based security and system solutions with smart card security controllers. For TC solutions, Infineon is worldwide market leader for TPM with the tamper-resistant hardware and the standard security firmware according to the TCG specification and applications. The TPM chip is compatible and interoperable with the standardized host software regarding device driver, board integration and TPM Software Stack (TSS). Due to this customer-oriented background, Infineon has a broad experience for developing, testing and integrating hardware parts (chips) as well as the necessary software and know-how to combine them to system solutions.

The RAINBOW project partner Infineon Technologies AT (IFAT) is an active member of the TCG and contributes intensively to the TCG standards, using experience from its wide security background, especially to the TPM, the TSS specification and the platform system solution standards. Additionally, the RAINBOW project partner DTU has also established an academic liaison with TCG.

Infineon participates in TCG as follows:

- TCG Promoter level membership

[10] http://www.trustedcomputinggroup.org

- TCG President and Chair of the TCGs Board of Directors (BoD)
- Chair of the Certification Program Committee (CPC), which targets especially the alignment of all TCG Compliance and Conformance evaluation for product certification,
- Chair of the TSS working group,
- Co-chair of the Security Evaluation working group,
- Contributes actively in Working Groups, e.g., the Cyber Resilience, TPM, PC Client, and Technical Committee

Main tasks attributed to the RAINBOW project:

- Infineon along with DTU and UBITECH contributed with related organizational and technical support, technology and know-how:
  - Intensive technical and organizational links and capabilities to the TCG in BoD, TPM, TSS and the CPC (Infineon is present and engaged in a number of relevant groups and activities in the TCG).
  - Specific TC support and management implementations, which could be used as a source of experience and as test objects.
  - Existing and novel concepts and their realization for using TC in datacentres (Cloud, Fog, edge) and in embedded controllers, like embedded systems or automotive (non-PC), market demands and evaluation requirements.

- DTU along with Infineon and UBITECH contributed to the TCG standardization and discussed the latest advances in DAA and attestation/revocation that have been performed by RAINBOW:
  - Evaluation of the concept for Reference Integrity Manifest (RIM) and the Attestation Framework Requirements of enhanced remote attestation variants in the Attestation WG focusing on the verification of properties for containers towards the creation of trust aware service graph chains. Validation of the current concept with TCG members and the impact on the current specification when it comes to the zero-touch attestation aspect offered by RAINBOW.
  - Evaluation of the DAA mechanism for integration in the CJDNS network. DAA is a privacy-preserving platform authentication mechanism which is supported by TCG and they are interested in promoting the use of DAA in various application domains. The fact that RAINBOW introduces the DAA as an integral part of CJDNS is an interesting aspect for the TNC working group.

- Infineon's previous experiences which are relevant to the tasks attributed to the RAINBOW project:
  - TC testing experience from already developed commercial modules, APIs and management software, together with the development results from the FutureTPM, E2SG or Cumulus project for the Linux world and already

  experience for TPM functionality implementation with embedded processors like embedded systems and other communication parts.
- Participation in several trust and security related public projects at EU and national level also in leading positions:
  - FP2020 and Horizon Europe, Eniac, ITEA
  - Germany: BMBF, BMWi

5.3 Activity Report

Infineon participated in a variety of working groups with leading positions as chair and active contributor to specifications. Infineon attended also the members meeting as chair of the TCGs Board of Directors. Since the beginning of the RAINBOW project, several regular phone conferences and the following members meetings with about 70-100 participants each were used for dissemination activities:

- TCG Members Meeting February 2020
- TCG Members Meeting June 2020
- TCG Members Meeting October 2020
- TCG Members Meeting February 2021
- TCG Members Meeting June 2021
- TCG Members Meeting October 2021
- TCG Members Meeting February 2022
- TCG Members Meeting July 2022
- TCG Members Meeting October 2022

As part of the DAA mechanism enhanced in the context of RAINBOW, DTU worked together with the TCG responsible for the TSS implementation towards the definition of a new privacy-preserving revocation model presented in the ACM WiSec 2021 conference [7].

6 Eclipse IoT Project Proposal

6.1 Eclipse IoT Overview

Eclipse IoT[11] is an open-source community that aims to provide projects that allow developers to build their own Internet of Things (IoT) solutions using interoperable open-source software. The Eclipse IoT project was started in 2012 – today there are more than 45 projects hosted within this community. As part of the Eclipse Foundation, the Eclipse IoT community has high visibility both in industry and academic circles. Getting a project accepted for inclusion is, thus, a big step towards extending the project's reach.

6.2 RAINBOW Contribution

RAINBOW's intent was to submit a proposal for a new project to the Eclipse IoT Community. After the second release of the RAINBOW platform, we decided to propose a Secure Admission Control Protocol for Kubernetes Control Plane targeting IoT devices over Mobile Ad-Hoc Networks[12]. This addresses a shortcoming in the Kubernetes (K8S) landscape, because such networks are currently largely neglected by the K8S community.

6.3 Activity Report

After deciding on the project's aim, the proposal for inclusion in the Eclipse IoT community was finalized and submitted in December 2022.

[11] https://iot.eclipse.org

[12] The text of the proposal can be found in Annex II – Eclipse IoT Proposal.

7 Conclusion

In this Deliverable, a report of the RAINBOW Standardization Activities was presented. The project's strategy is to influence fog computing applications of the future and let them benefit from the outcomes of RAINBOW. To this end, we have chosen three target groups. Open Horizon is an open-source platform developed by the Linux Foundation Edge with the aim of managing the lifecycle of production edge applications, which makes it a good target for influencing future fog applications. Centaurus is a Linux Foundation open-source platform for creating and managing large scale private or public Cloud-Edge infrastructures and presents a good opportunity for contributing to a software product that may be used by Cloud/Edge vendors in the future. The Trusted Computing Group is responsible for the TPM and other security-related standards and is, thus, a good place for influencing the security aspects of future applications. In addition to these contributions, RAINBOW has submitted a project proposal to the Eclipse IoT community, which is very active in fog-related projects. We reported on our activities regarding all target groups and presented the final outcomes.

8 References

[1] LF Edge, "Open Horizon," [Online]. Available: https://www.lfedge.org/projects/openhorizon/. [Accessed 09 06 2021].

[2] LF Edge, "Project Stages: Definitions and Expectations," [Online]. Available: https://wiki.lfedge.org/display/LE/Project+Stages%3A+Definitions+and+Expectations. [Accessed 09 06 2021].

[3] LF Edge, "TSC 2021-03-29," [Online]. Available: https://wiki.lfedge.org/display/OH/TSC+2021-03-29. [Accessed 10 06 2021].

[4] P. Huang, Y. Bai, F. Li, X. Ding, Q. Chen, D. Vij, Du Peng and Y. Xiong, "Arktos: A Hyperscale Cloud Infrastructure for Building Distributed Cloud," in *2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC)*, 2022.

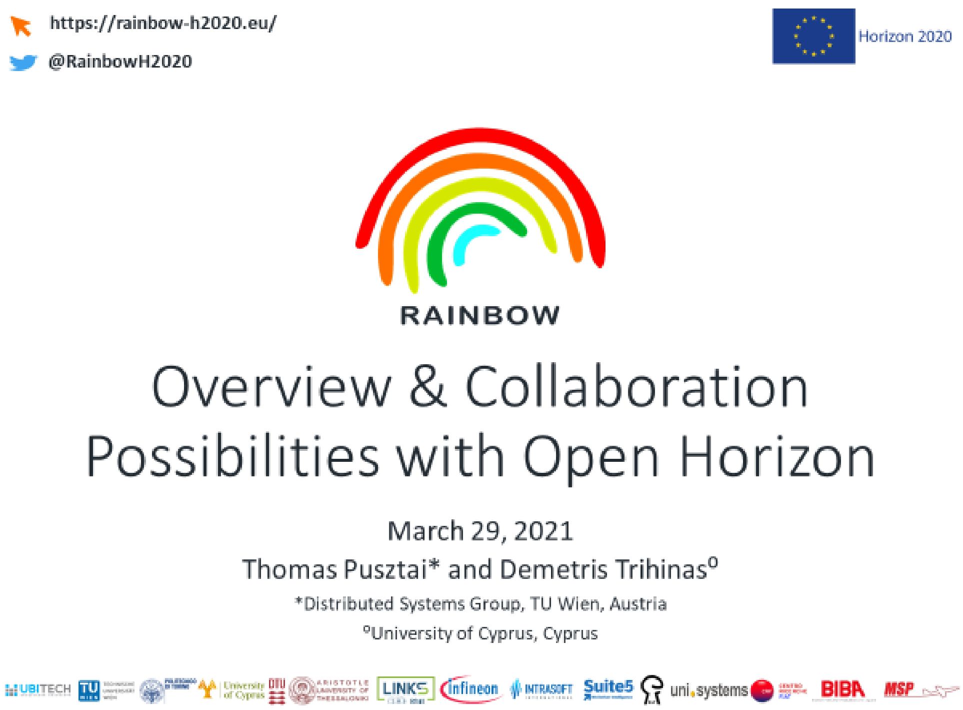
[5] T. Pusztai, F. Rossi and S. Dustdar, "Pogonip: Scheduling Asynchronous Applications on the Edge," in *2021 IEEE 14th International Conference on Cloud Computing (CLOUD)*, 2021.

[6] T. Pusztai, S. Nastic, A. Morichetta, V. Casamayor Pujol, P. Raith, S. Dustdar, D. Vij, Y. Xiong and Z. Zhang, "Polaris Scheduler: SLO- and Topology-aware Microservices Scheduling at the Edge," in *2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC)*, 2022.

[7] B. Larsen, T. Giannetsos, I. Krontiris and K. Goldman, "Direct anonymous attestation on the road: efficient and privacy-preserving revocation in C-ITS," in *WiSec '21: Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2021.

Annex I – Communication with Open Horizon

RAINBOW Presentation at Open Horizon TSC Meeting on March 29, 2021



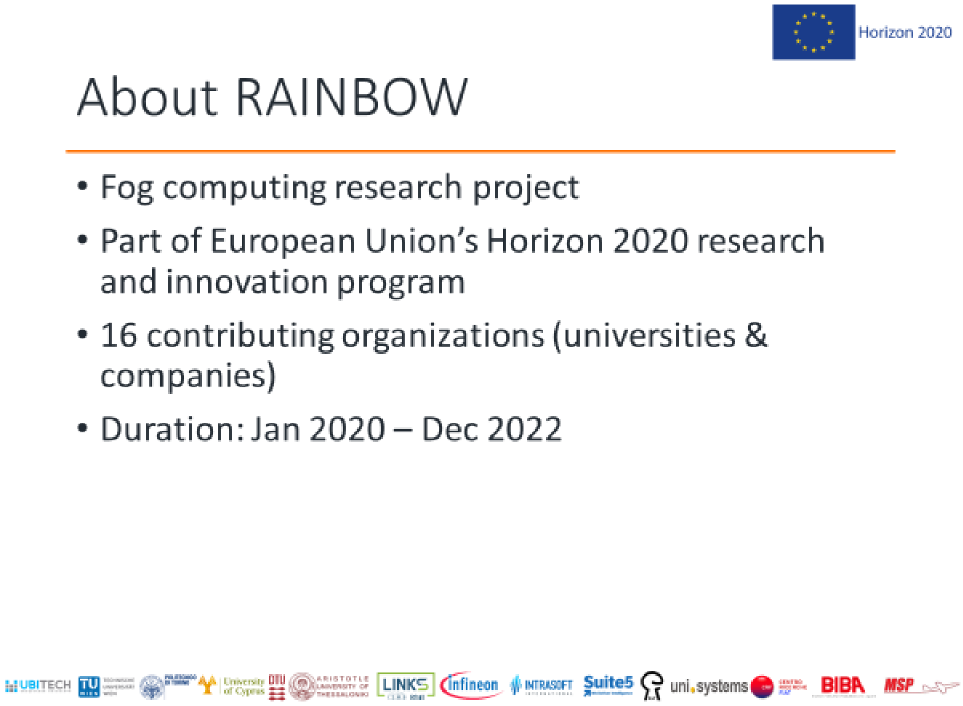
https://rainbow-h2020.eu/
@RainbowH2020

Horizon 2020

RAINBOW

Overview & Collaboration Possibilities with Open Horizon

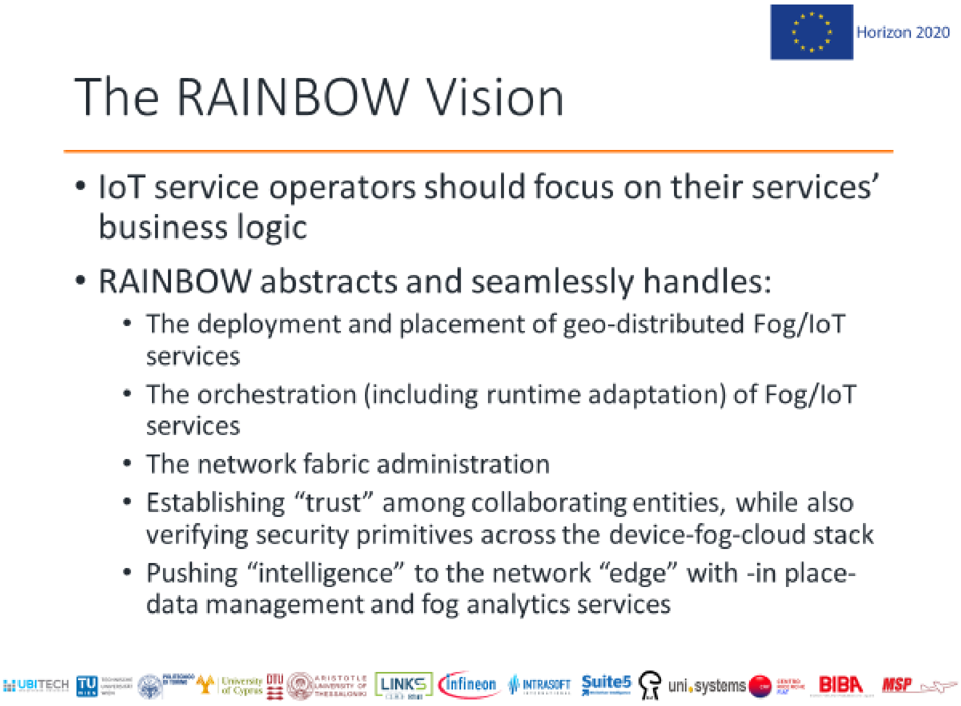
March 29, 2021
Thomas Pusztai* and Demetris Trihinas°
*Distributed Systems Group, TU Wien, Austria
°University of Cyprus, Cyprus



Horizon 2020

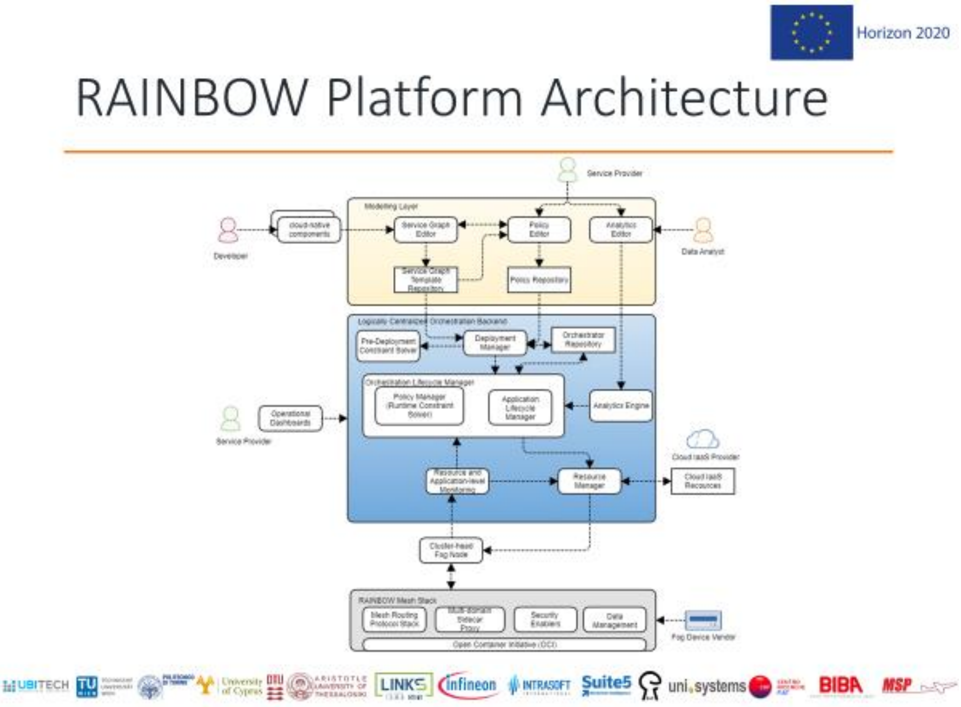
About RAINBOW

- Fog computing research project
- Part of European Union’s Horizon 2020 research and innovation program
- 16 contributing organizations (universities & companies)
- Duration: Jan 2020 – Dec 2022

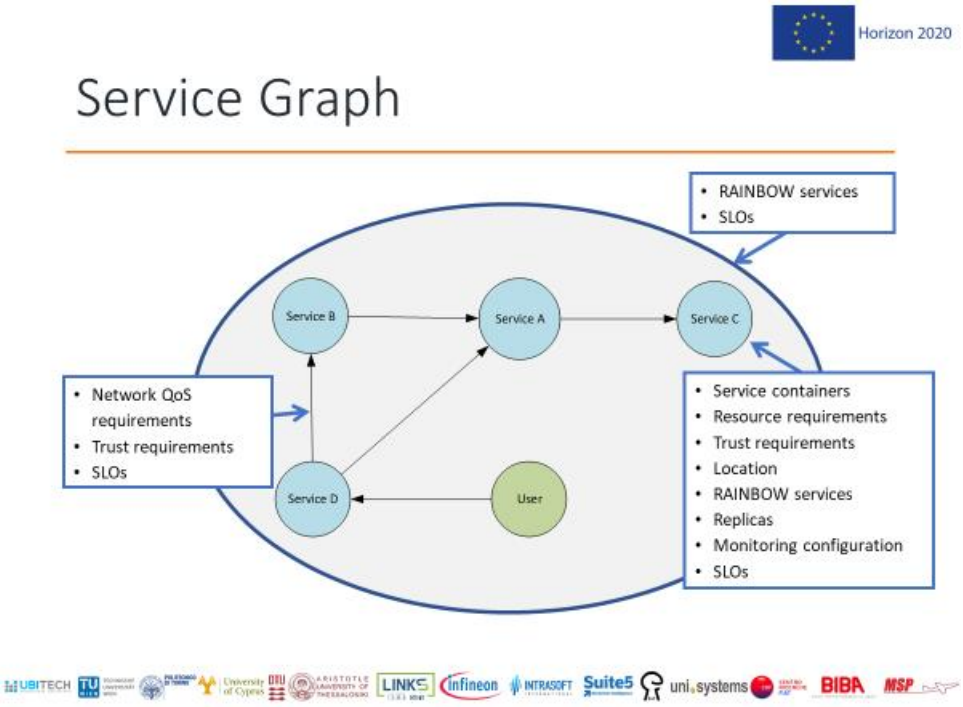


The RAINBOW Vision

- IoT service operators should focus on their services' business logic
- RAINBOW abstracts and seamlessly handles:
  - The deployment and placement of geo-distributed Fog/IoT services
  - The orchestration (including runtime adaptation) of Fog/IoT services
  - The network fabric administration
  - Establishing "trust" among collaborating entities, while also verifying security primitives across the device-fog-cloud stack
  - Pushing "intelligence" to the network "edge" with -in place- data management and fog analytics services



RAINBOW Platform Architecture



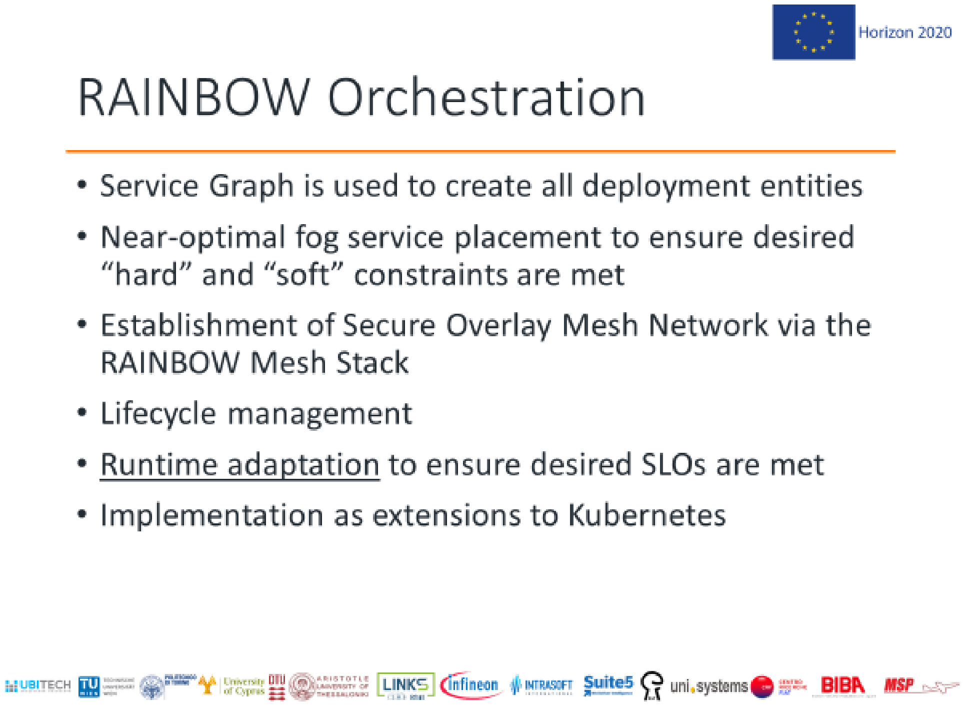
Horizon 2020

Service Graph

- RAINBOW services
- SLOs

- Network QoS requirements
- Trust requirements
- SLOs

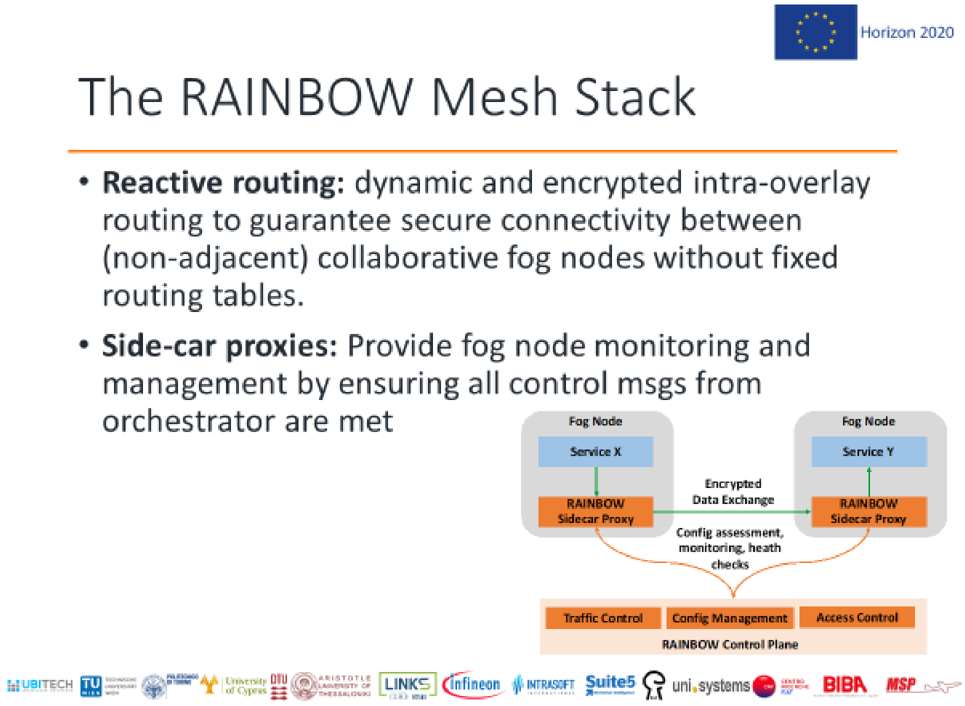
- Service containers
- Resource requirements
- Trust requirements
- Location
- RAINBOW services
- Replicas
- Monitoring configuration
- SLOs



Horizon 2020

RAINBOW Orchestration

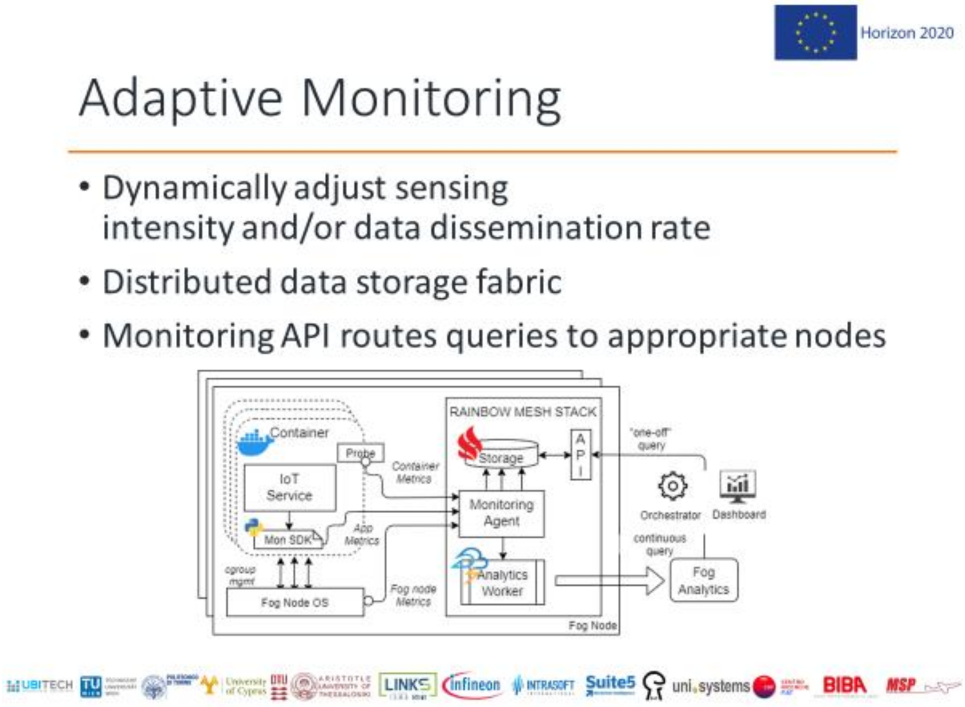
- Service Graph is used to create all deployment entities
- Near-optimal fog service placement to ensure desired “hard” and “soft” constraints are met
- Establishment of Secure Overlay Mesh Network via the RAINBOW Mesh Stack
- Lifecycle management
- Runtime adaptation to ensure desired SLOs are met
- Implementation as extensions to Kubernetes



Horizon 2020

The RAINBOW Mesh Stack

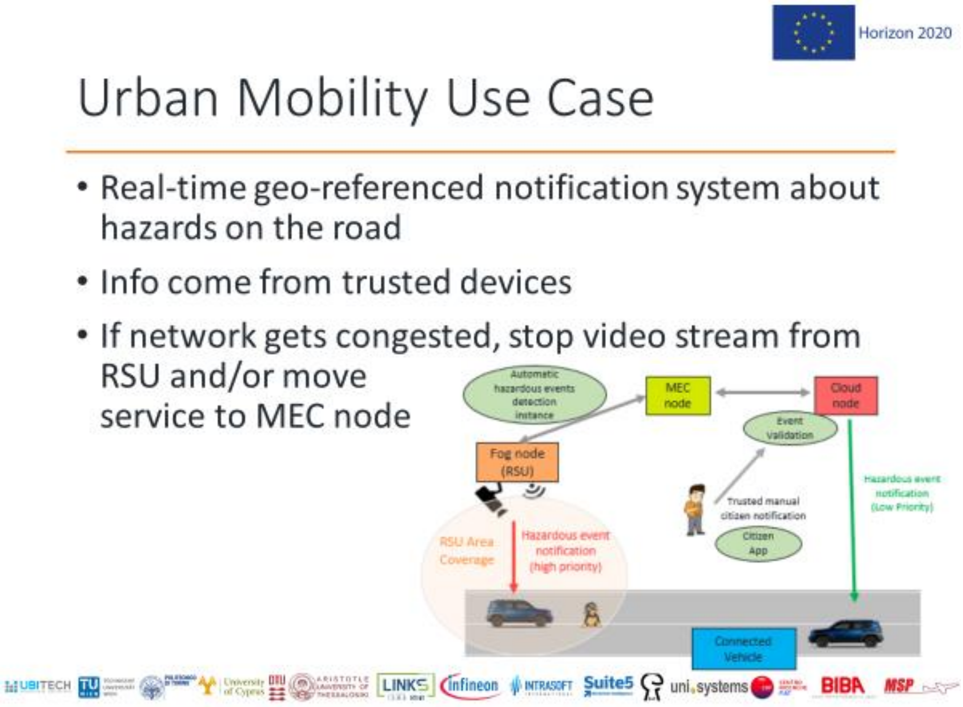
- **Reactive routing:** dynamic and encrypted intra-overlay routing to guarantee secure connectivity between (non-adjacent) collaborative fog nodes without fixed routing tables.
- **Side-car proxies:** Provide fog node monitoring and management by ensuring all control msgs from orchestrator are met



Horizon 2020

Adaptive Monitoring

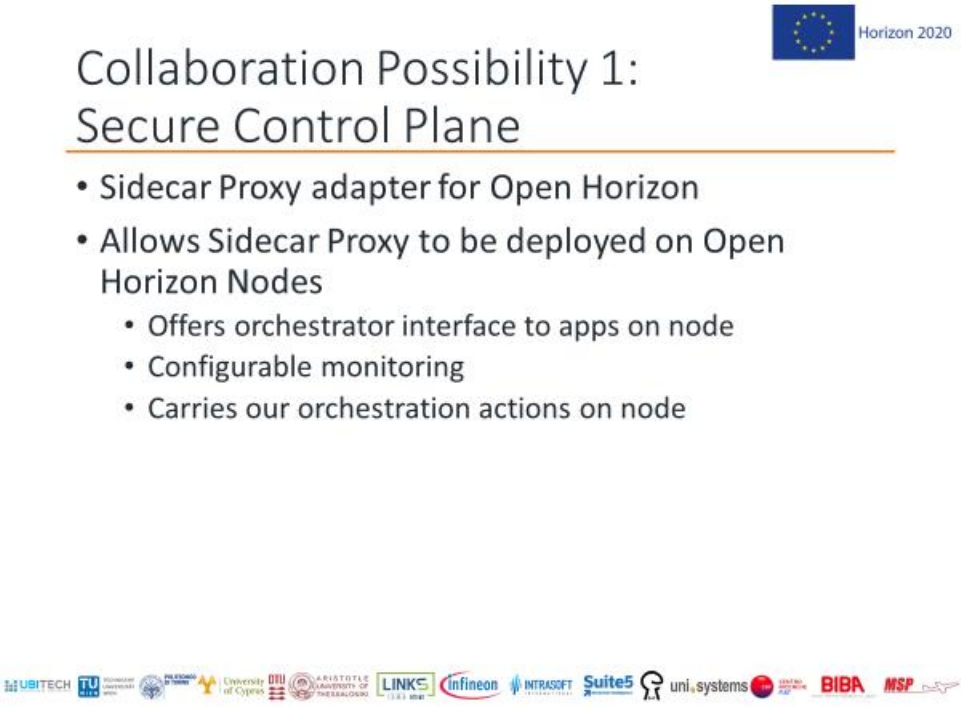
- Dynamically adjust sensing intensity and/or data dissemination rate
- Distributed data storage fabric
- Monitoring API routes queries to appropriate nodes



Urban Mobility Use Case

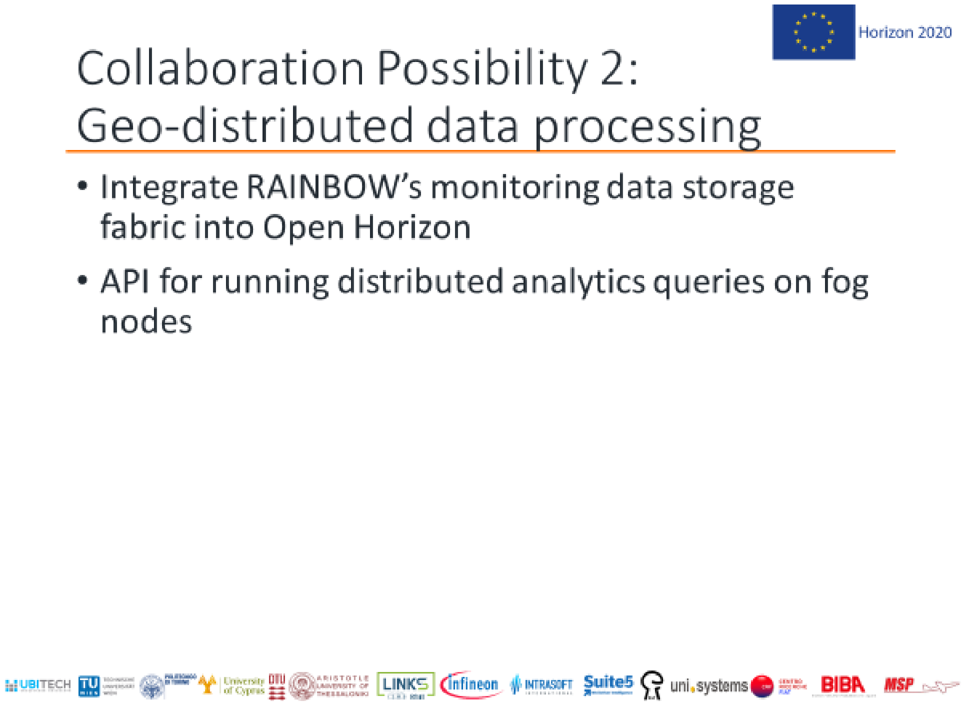
- Real-time geo-referenced notification system about hazards on the road
- Info come from trusted devices
- If network gets congested, stop video stream from RSU and/or move service to MEC node

Automatic hazardous events detection instance
MEC node
Cloud node
Event Validation
Fog node (RSU)
Hazardous event notification (Low Priority)
Trusted manual citizen notification
Citizen App
RSU Area Coverage
Hazardous event notification (high priority)
Connected Vehicle



Collaboration Possibility 1: Secure Control Plane

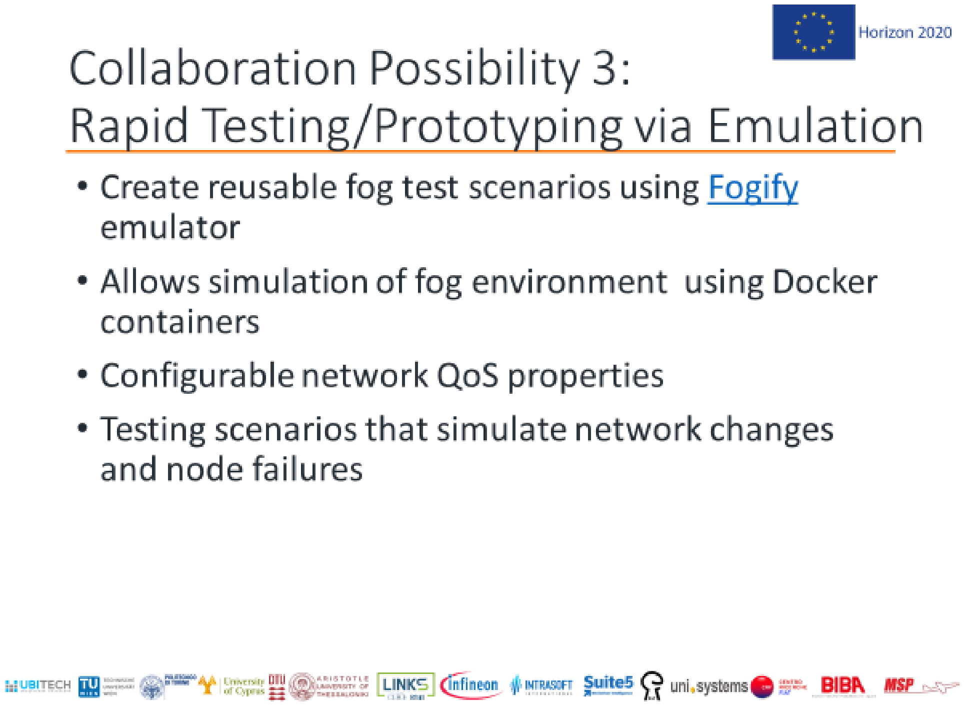
- Sidecar Proxy adapter for Open Horizon
- Allows Sidecar Proxy to be deployed on Open Horizon Nodes
  - Offers orchestrator interface to apps on node
  - Configurable monitoring
  - Carries our orchestration actions on node



Horizon 2020

Collaboration Possibility 2: Geo-distributed data processing

- Integrate RAINBOW's monitoring data storage fabric into Open Horizon
- API for running distributed analytics queries on fog nodes



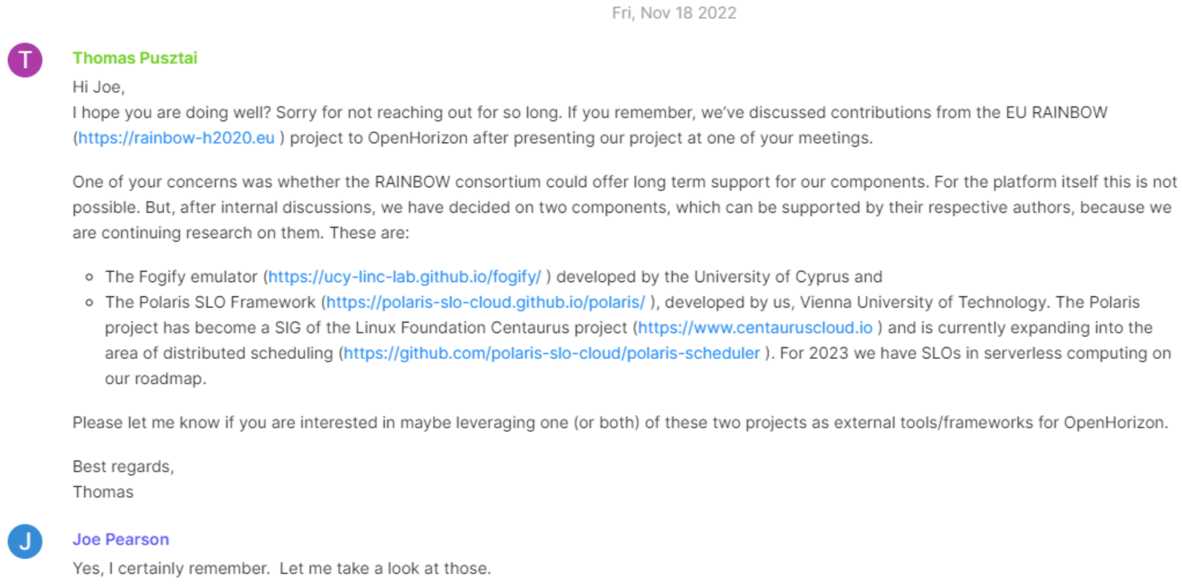
Horizon 2020

Collaboration Possibility 3: Rapid Testing/Prototyping via Emulation

- Create reusable fog test scenarios using Fogify emulator
- Allows simulation of fog environment using Docker containers
- Configurable network QoS properties
- Testing scenarios that simulate network changes and node failures

Thank you!

Final Offer of RAINBOW Components to Open Horizon on Nov 18, 2022



Fri, Nov 18 2022

Thomas Pusztai

Hi Joe,
I hope you are doing well? Sorry for not reaching out for so long. If you remember, we've discussed contributions from the EU RAINBOW (https://rainbow-h2020.eu) project to OpenHorizon after presenting our project at one of your meetings.

One of your concerns was whether the RAINBOW consortium could offer long term support for our components. For the platform itself this is not possible. But, after internal discussions, we have decided on two components, which can be supported by their respective authors, because we are continuing research on them. These are:

- The Fogify emulator (https://ucy-linc-lab.github.io/fogify/) developed by the University of Cyprus and
- The Polaris SLO Framework (https://polaris-slo-cloud.github.io/polaris/), developed by us, Vienna University of Technology. The Polaris project has become a SIG of the Linux Foundation Centaurus project (https://www.centauruscloud.io) and is currently expanding into the area of distributed scheduling (https://github.com/polaris-slo-cloud/polaris-scheduler). For 2023 we have SLOs in serverless computing on our roadmap.

Please let me know if you are interested in maybe leveraging one (or both) of these two projects as external tools/frameworks for OpenHorizon.

Best regards,
Thomas

Joe Pearson

Yes, I certainly remember. Let me take a look at those.

Annex II – Eclipse IoT Proposal

Eclipse IoT proposal: **Secure Admission Control for Kubernetes Control Plane targeting IoT devices over MANETs**

The adoption of Kubernetes (K8S) in the IoT ecosystem is growing at a fast pace since 'containerized orchestration system is necessary to deliver the right information to the right place in as close to real-time as possible'[13]. On the other hand, several edge devices rely on Mobile Ad-Hoc Networks (hereinafter MANETs) especially in swarm applications (e.g., drones, factory robots). Thus, there is a need to enhance the existing Kubernetes control plane business logic to support **secure autonomous admission control** of K8S cluster member nodes.

With the term autonomous we imply that all existing manual tasks that relate to a static pre-configuration of a joining node must be substituted by **an admission control protocol** that is tailored to MANETs. In the frame of this Eclipse IoT proposal such an admission protocol is proposed.

The dynamicity and the uncertainty that is introduced in the operational environment of MANETs raises significant complexity to the control plane protocol. More specifically, in an IoT device that is part of a mesh cluster the following tangible problems have to be addressed:

- Static assignment of **IPs and node identifiers** cannot be performed. In fact, nodes should be able join and leave freely; completely unsupervised i.e., without the existence of a coordination entity like K8S Controller.
- **Node integrity** has to be proven in the sense that rogue nodes shall not join a cluster
- Links among the nodes are **established and torn down in an opportunistic manager**. To this end, the topology may vary significantly between two consecutive timestamps. That is to say that no centralized routing protocol can be used in its vanilla version to accommodate the admission signalling.

The proposed protocol will attempt to tackle this complexity through the introduction of **specific autonomic behaviours**; thus, adhering to self-* principles such as:

- **Self-Configuration**: Automatic configuration of IoT nodes that wish to join a K8S cluster
- **Self-Healing**: Automatic discovery, and correction of faults
- **Self-Optimization**: Automatic monitoring and control of resources to ensure the optimal functioning with respect to the defined requirements
- **Self-Protection**: Proactive identification and protection from arbitrary attacks

[13] https://www.cncf.io/blog/2020/09/25/kubernetes-could-be-the-one-to-make-the-internet-of-things-iot-reach-its-potential/

Figure 1 provides a state transition diagram of the proposed admission protocol from the initial state of a node to the level of final onboarding to a Kubernetes cluster.

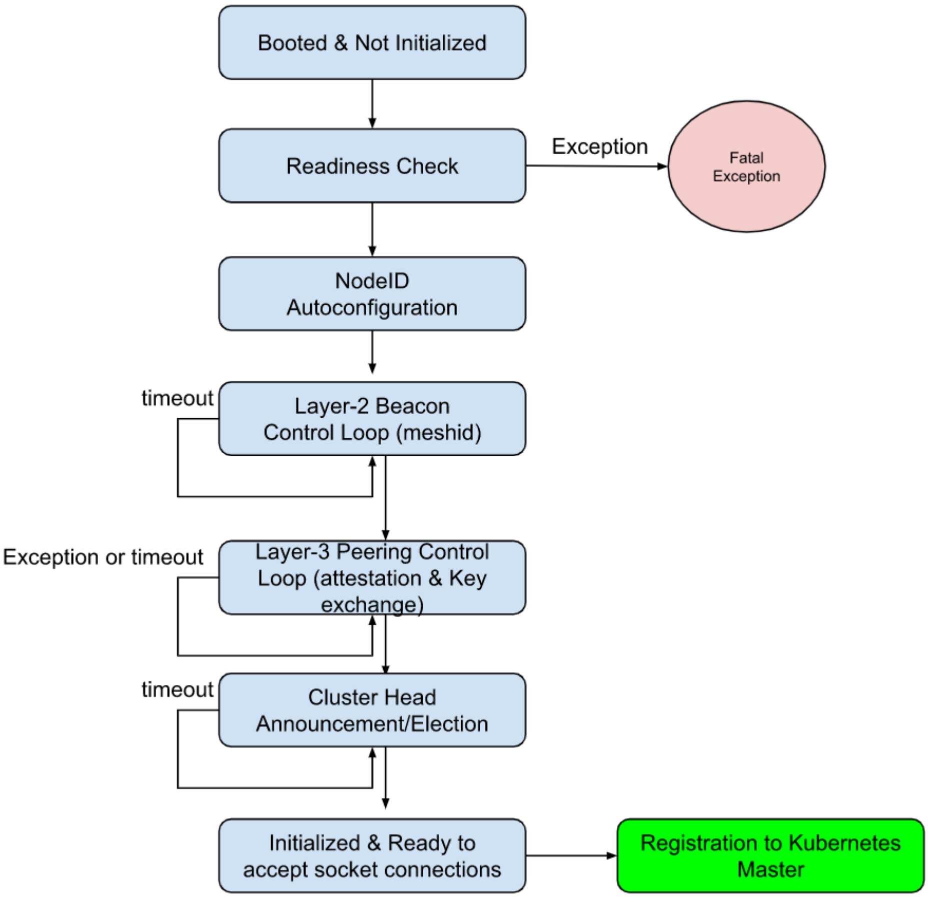

Booted & Not Initialized
Readiness Check
Exception
Fatal Exception
NodeID Autoconfiguration
timeout
Layer-2 Beacon Control Loop (meshid)
Exception or timeout
Layer-3 Peering Control Loop (attestation & Key exchange)
timeout
Cluster Head Announcement/Election
Initialized & Ready to accept socket connections
Registration to Kubernetes Master

Figure 1: Admission Control Protocol

In the initial state, the IoT device has powered up and the operating system has booted. At this point, it should be clarified that issues such as secure-boot are outside the scope of the admission protocol since the boot-time attestation process has to be provided by the IoT board (using hardware-assisted crypto-primitives).

After successful boot, a readiness-check process will be performed in order to verify that the specific IoT device is able to participate in a cluster. This process shall evaluate: a) the validity of the architecture along the Linux kernel features; b) the existence of Open Container Interface (OCI); c) the existence of hardware or software-based Trusted Platform Module (which is necessary since it exposes crypto primitives that are appropriate for autoconfiguration, attestation, signatures validity, encryption/decryption); d) the existence of a Keystore and a Truststore on the /root filesystem and e) the existence of a mesh-enabled (e.g. 802.11s) physical network card.

The next steps of the protocol aim to materialize bridging i.e., to tackle all signalling for joining a cluster. This assumes that the IoT node is in physical proximity with an existing established cluster member or to state it in a more formal way that **has a wireless**

network transceiver that is able to perform a link with an adjacent node. However, joining a cluster requires the node to have **a layer-3 identity** i.e., IP address.

According to the operational assumption raised above, there is **no DHCP server and hence no IP distribution authority**. Therefore, the node must be assigned an IP address which **does not collide with any IP of nodes that exist in the cluster**. In addition to that, the IP must be bound to the identity of the node and **has to be used for connecting to the "public" internet** which refers to the logical centralized Kubernetes cluster.

To cope with this complexity, the proposed protocol relies on:

a) **IPv6 address spaces** which are vastly enlarged compared to IPv4
b) **Randomly generated public keys** that are used widely in asymmetric cryptography
c) **Collision avoidance hashing algorithms** that are fed/**initialized by the randomly generated public key** per node
d) a **Distributed Hash Table-based routing scheme**

In a DHT-based network, each node is assigned a unique identifier, or "key," and is responsible for storing and managing data associated with that key. When a node wants to send data to another node, it looks up the destination node's key in the DHT and routes the data directly to it. This allows nodes to communicate with each other without the need for a central server or authority. It also enables the network to scale horizontally, as new nodes can join and leave the network without disrupting the flow of data.

The proposed protocol leverages an existing encrypted DHT-based routing scheme called Yggdrasil[14] in order to achieve **IPv6 autoconfiguration**. Prior to the autoconfiguration step each node should perform an attestation protocol in order to achieve **configuration-integrity verification**. When integrity is verified IoT nodes can join a K8s cluster by interacting with an announced Cluster Head that coincides with the Kubernetes master node.

[14] https://yggdrasil-network.github.io