

DeviceQoSEvaluator Support System

System Description

Abstract

This document provides system description for the **DeviceQoSEvaluator Support System**.

Contents

1 Overview	3
1.1 Significant Prior Art	3
1.2 How This System Is Meant to Be Used	3
1.3 System functionalities and properties	4
1.4 Important Delimitations	5
2 Services produced	6
2.1 service qualityEvaluation	6
2.2 service deviceQualityDataManagement	6
2.3 service monitor	6
3 Security	7
4 References	8
5 Revision History	9
5.1 Amendments	9
5.2 Quality Assurance	9

1 Overview

This document describes the DeviceQoSEvaluator Support System, which is responsible for collecting and calculating Key Performance Indicators (KPIs) from devices operating within a Local Cloud. Based on the calculated KPIs this support system assists in finding the best service provider system to fulfill specific needs. Using these KPIs, the system assists in selecting the most suitable service providers to meet specific requirements.

The rest of this document is organized as follows. In Section 1.1, we reference major prior art capabilities of the system. In Section 1.2, we describe the intended usage of the system. In Section 1.3, we describe fundamental properties provided by the system. In Section 1.4, we describe delimitations of capabilities of the system. In Section 2, we describe the abstract services produced by the system. In Section 3, we describe the security capabilities of the system.

1.1 Significant Prior Art

The strong development on cloud technology and various requirements for digitisation and automation has led to the concept of Local Clouds (LC).

"The concept takes the view that specific geographically local automation tasks should be encapsulated and protected." [1]

One of the key building blocks in realising such a Local Cloud is the ability to ensure quality services across various aspects. The DeviceQoSEvaluator Support System is intended to provide this functionality at the device performance level.

The previous versions of the Arrowhead Framework (4.6.x) have a system (qos monitor) that offers similar functionality, however there are some key differences, even on conceptual level:

- The 5th generation of the Arrowhead Framework enables the integration of multiple service-quality evaluator systems through a common evaluation service, which this system also implements.
- In addition to network communication latency, this system enables the measurement of CPU and RAM usage, as well as network interface throughput.

1.2 How This System Is Meant to Be Used

The DeviceQoSEvaluator provides device-level performance measurement capabilities, as well as mechanisms for evaluating (filtering or ranking) systems based on their associated device KPIs. When deployed within a Local Cloud, this support system periodically queries the ServiceRegistry Core System to synchronize its internal data model of registered systems and their devices. This internal model serves as the basis for executing measurement routines and computing the corresponding KPIs. Device entities are derived from the set of registered system and device addresses. If multiple systems are registered with an identical network address, these systems are interpreted as being hosted on the same underlying device.

Basic performance metrics are available by default; however, augmented measurement capabilities can be enabled by deploying a lightweight device-side agent that executes additional measurement routines.

Based on the collected device-level performance metrics, the corresponding KPIs are derived, and systems can be filtered or ranked according to configurable KPI sets and weighting factors.

1.3 System functionalities and properties

1.3.1 Functional properties of the system

Based on the periodic measurements, the following KPIs are continuously calculated:

- **Basic Measurement**

Measurement: Round-Trip Time

Metrics: MIN, MAX, MEAN, MEDIAN, CURRENT

The measurement is performed by initiating TCP connection attempts to a potentially closed port on the target device. The elapsed time between the connection request and the resulting “connection refused” response is recorded (in milliseconds). If the randomly selected port happens to be open, a new port is selected until a closed one is found. A single measurement consists of multiple connection attempts to produce stable metric values.

- **Augmented Measurements**

Measurements: CPU Total Load, Memory Used, Network Egress Load, Network Ingress Load

Metrics: MIN, MAX, MEAN, MEDIAN, CURRENT

The measurements are performed by the device-side agent, which continuously monitors the current load percentages and stores the sampled values within a configured time window. The DeviceQoSEvaluator system periodically retrieves these samples and computes the corresponding metric values, which are then stored for a configurable retention period.

A device's QoS is calculated on demand and based on a given KPI set with associated weighting factors for a given time window. Example:

$$Q_i(T) = w_1 \cdot RoundTripTime(T)_{current} + w_2 \cdot NetworkEgressLoad(T)_{max} + w_3 \cdot CpuTotalLoad(T)_{mean}$$

1.3.2 Non functional properties of the system

- If an Authentication system is present in the Local Cloud, the DeviceQoSEvaluator will use its service(s) to verify a requester system before responding to its request.
- If X.509 authentication policy is used instead of a dedicated Authentication system, the DeviceQoSEvaluator will verify the provided certificate before any response.
- If the Blacklist system is present in the Local Cloud, the DeviceQoSEvaluator may use its service(s) to check the requesters.

1.3.3 Data stored by the system

In order to achieve the mentioned functionalities, DeviceQoSEvaluator is capable to store the following information set:

- **Device entities:** The derived device entities and the associated network addresses, ports and other relevant data that are necessary for performing the measurements.
- **System names:** The unique system names associated with a device entity.
- **Measurement metrics:** Time series measurement data associated with a device entity.

1.4 Important Delimitations

- If the Local Cloud does not contain an Authentication system or is not using X.509 certificates, there is no way for the DeviceQoSEvaluator to verify the requester system. In that case, the DeviceQoSEvaluator will consider the authentication data that comes from the requester as valid.
- DeviceQoSEvaluator is not able to perform measurements without cooperating with the ServiceRegistry Core System.
- DeviceQoSEvaluator instantiates internal routines for all derived device entities. As Local Cloud size increases, the system may require additional computational resources to execute measurements without incurring significant processing delays.

2 Services produced

2.1 service **qualityEvaluation**

The purpose of this service is to filter or sort systems based on a specified KPI set with associated weighting factors over a defined time window. The service is offered for both application and Core/Support systems.

2.2 service **deviceQualityDataManagement**

The purpose of this service is to query the measurement metric values and also to trigger the synchronization with ServiceRegistry Core System manually. This service is offered for Core/Support systems.

2.3 service **monitor**

Recommended service. Its purpose is to give information about the provider system. The service is offered for both application and Core/Support systems.

3 Security

For authentication, the DeviceQoSEvaluator can utilize an other core system, the Authentication system's service to verify the identities of the requester systems. If no Authentication system is deployed into the Local Cloud, the DeviceQoSEvaluator can accept X.509 certificates for identification. If none of these are available, the DeviceQoSEvaluator trusts the requester system's self-provided identity.

For authorization, qualityEvaluation is available for everyone who can get access information via service orchestration (or directly from the ServiceRegistry, if direct lookup is allowed in the Local Cloud). The device-QualityDataManagement service is available for those who has the appropriate permissions.

The implementation of the DeviceQoSEvaluator can decide about the encryption of the connection between the DeviceQoSEvaluator and other systems.

4 References

- [1] J. Delsing and P. Varga, *Local automation clouds*. Boca Raton: Taylor & Francis Group, 2017, p. 28.
[Online]. Available: <https://doi.org/10.1201/9781315367897>



ARROWHEAD

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	YYYY-MM-DD	5.2.0		Xxx Yyy

5.2 Quality Assurance

No.	Date	Version	Approved by
1	YYYY-MM-DD	5.2.0	