INTEROPERABILITY TESTING GUIDE FOR IoT

Author: Karina da Silva Castelo Branco

Advisor: Valéria Lelli Leitão Dantas

VERSÃO 1.0

JUNHO 2024

SUMMARY

About the guide

The central objective of this interoperability guide is to provide at the end of use a test plan that serves as a guideline for testers when evaluating interoperability in Internet of Things (IoT) applications. In this context, the Abstract Test Cases presented are conceived as abstract, and their adaptation according to the specific scenario of the application to be tested is essential. Using this guide offers a fundamental advantage in terms of efficiency, as testers will not need to conduct extensive literature searches to define testing procedures. Instead, the guide will provide clear and targeted guidance for performing tests, contributing to a more effective and agile approach to the software systems interoperability assessment process. This targeted and structured approach provides a reliable methodology to ensure the robustness and quality of software applications, making it a valuable tool for testing professionals and researchers in the field of Software Engineering. This guide aims to offer assistance in carrying out Interoperability tests on IoT applications. It is structured into 12 distinct sections, such as: Interoperability Definitions (Topic 1), characterístics Correlation (Topic 2), Interoperability Testing Challenges (Topic 3), Test Environment Configuration (Topic 4), Abstract Abstract Test Case s (Topic 4.c), Measurements (Subtopic 4.d), Sub-Characteristic Impact (Topic 8) and others. This guide is based on the four Interoperability subcharacteristics defined in the ISO/IEC 30141:2018 standard, namely: Data Semantics, Communication Protocols, System Integration and Network Protocol.Therefore, it was designed with the aim of covering IoT Interoperability test scenarios based on these sub-characterístics. For each sub-characterístics, topic 4 of the guide offers the relevant definitions, context (properties), abstract Abstract Test Cases and measurements. In addition to this topic, there are seven other topics intended to address the assessment of interoperability characteristics as a whole. The 12 topics present the following:

- **Characteristic definition:**This topic deals with the characteristic that will be tested.
- **Correlation of Characteristics:**presents the correlations, both positive and negative, between the target characterístics and other IoT characterístics.
- **Interoperability Testing Challenges:**This topic aims to present the varied challenges related to the evaluation of the interoperability characteristic in IoT applications.
- **Test Environment Configuration:**describes the IoT environment required to perform testing of the target characterístics, including the types of devices and the external software or hardware involved.
- **Subcharacteristic Definition:**If applicable, this topic describes the subcharacteristics associated with the target characteristic.
- **Contextualization:**Describes the properties related to the trait and, if applicable, subcharacteristics.
- **Abstract Abstract Test Case s:**provides general guidelines for conducting tests of the target trait or, when applicable, subcharacteristics.
- **Measurements:**presents the metrics used to evaluate the target characteristic and, if applicable, the subcharacteristics.
- **Impact of Sub-characterístics:**if necessary, this topic addresses the relationships between subcharacterístics based on contextualization properties.
- **Cost benefit:**presents an analysis of the cost-benefit ratio for carrying out the tests, based on the correlations identified in the Characteristics Correlation topic.

- **Tool Suggestions:**lists tools that can be employed to automate the collection of metrics related to the target characteristic.
- **Example of Using the Guide:**Providing a practical application scenario, this topic illustrates how to use the guide in the context of the target characteristic.

We hope this guide will be valuable in efficiently running Interoperability tests on IoT applications and in comprehensively understanding the characterístics and sub-characterístics involved.

Instructions for Using the Guide

1. **Interoperability Definitions:** initially, it is essential to understand the Interoperability definitions presented in topic 1 of this guide, which was designed to align knowledge in relation to the characteristic addressed.

2. **Selection of Relevant characterístics:** based on the specific application you want to test, use the definitions and relationships exposed in characterístics Correlation (topic 2) to identify and list the IoT characteristics related to interoperability that take priority for your application. These characteristics may encompass all or some, as dictated by the context.

3. **Test Environment Configuration:** ensure that the environment intended for interoperability tests meets the requirements outlined in topic 4 (Test Environment Configuration).

4. **Understanding Sub-Characteristics:** in the subsequent step, dedicate yourself to understanding the definitions and properties presented in the topic corresponding to each subcharacteristic (topic 5, 6 and 7). Based on this understanding, select the properties of the subcharacteristics that will be subjected to evaluation. It is worth noting that it is not strictly necessary to evaluate all sub-characteristics of the guide; the decision is up to the user, taking into account the specific needs of their application.

5. **Assessment of the Impact of Subcharacteristics:** If you chose to evaluate all properties in the previous step, this step can be skipped. Using topic 8 (Impact of Subcharacteristics) as a reference, identify which selected properties are influenced by the other properties. Make sure these properties are subject to assessment as well.

6. **Metric Selection:**Select the metrics to be used in the evaluation. We emphasize that it is not advisable to select just one metric; instead, we recommend running all metrics related to the selected subcharacteristic.

7. **Cost-Benefit Calculation:** After complete planning, proceed to calculate the cost-benefit that the tests and measurements will require. Topic 9 (Cost-Benefit) explains this process in detail.

8. **Use of Suggested Tools:** if you would like to use any of the tools suggested in this guide, see Table 2 to identify which tools can assist in collecting the selected metrics.

In topic 11, information is provided about the licenses for these tools and the websites to access them.

9. **Test Execution:** after completing all the planning, it is time to run the tests. All abstract tests selected must be transformed into concrete Abstract Test Cases, considering the context of their application. This involves defining the limits and acceptable values for the selected metrics, taking into account the correlations identified in topic 2. Figure 8 in Appendix A can be used as a guide to speed up this process. If doubts arise regarding the practical implementation of this guide, topic 12 (Example of Guide Use) provides additional guidance.

Additionally, some additional information important to understanding this guide includes:

- All tests can be performed locally.
- Measurements can be collected through Abstract Test Case s.
- Cost-benefit can be used as a test prioritization criterion.
- The suggested tools can assist with testing and measurements, but are not mandatory.
- The impact of subcharacterístics helps identify other properties that may be relevant to testing.

Introduction

This guide aims to assist in validating the Interoperability characterístics in Internet of Things (IoT) applications, through topics such as characterístics Correlation, Abstract Abstract Test Cases, Measurements, Impact of sub-characterístics, among others. This guide was built based on ISO/IEC 30141:2018, the current state of the art and on the Interoperability sub-characteristics identified based on the literature through systematic mapping techniques. The guide is constructed this way to cover as many IoT Interoperability test scenarios as possible. For each subcharacteristic, definitions, properties, Abstract Test Case s and measurements are presented.

The ISO/IEC 30141:2018 standard presents the subcharacteristics of Interoperability in IoT. This standard was published in 2018 and provides guidelines for evaluating Interoperability in IoT systems, including definitions of terms and testing requirements for the different Interoperability subcharacteristics. Sub-characterístics include data semantics, communication protocols, system integration, and network protocol.

The ISO/IEC 30141:2018 standard provides guidance for the use of NMPN for modeling business processes that involve multiple stakeholders and organizations, and aims to ensure Interoperability and clarity in communication between the parties involved.

This standard is important for organizations working collaboratively on business processes, as it helps ensure that business process modeling is consistent and understandable for everyone involved, which can lead to greater efficiency and effectiveness in carrying out the processes.

1. Interoperability Definitions

The ISO/IEC 30141:2018 standard defines Interoperability as "The ability of different systems and organizations to work together (exchange information and actions) effectively and efficiently".

These definitions highlight the importance of having common standards and technical specifications to ensure that systems can work together appropriately and that information can be shared and interpreted correctly. A*Interoperability*It is essential to ensure the effectiveness and efficiency of collaborative business processes and to facilitate communication between the parties involved. Table 1 presents the main definitions of Interoperability:

Table 1 - Interoperability Definitions

Definition	Authors
Interoperability is the ability of two or more systems or components to exchange information and use the information that has been exchanged.	Legner and Wander (2006)
Interoperability can be defined as a state in which two applications can accept and understand each other's data and perform a given task satisfactorily without human intervention.	(Gulla et al., 2006)
The ability for different systems, devices or applications to communicate and interact efficiently and smoothly.	Ferreira (2014)
The ability of a system to exchange data and information with other systems without loss or corruption of information.	ISO/IEC 15926:2019
The ability of different systems and organizations to work together (exchange information and actions) effectively and efficiently.	ISO/IEC 30141:2018
Interoperability in IoT refers to the ability of connected devices (such as sensors, smart devices, gateways, etc.) to communicate and interact efficiently and transparently, regardless of their origin, manufacturer or communication protocol.	ISO/IEC21823:2022

2. Correlation of Characteristics

Through a bibliographic analysis, it was possible to establish the relationship of 14 characteristics with Interoperability among the 28 identified for IoT. These connections can be useful to detect possible conflicts in application requirements and look for ways to mitigate them. Figure 1 presents connections classified into three types: red rectangles indicate characteristics that have a negative influence on Interoperability, such as application performance; green rectangles indicate positive influence, such as the availability of the application, in which the greater the common protocols required in the application, the greater benefits the Interoperability requirements will have; The yellow rectangles depend on the application context, such as the efficiency characteristic, which can have a positive or negative influence, for example depending on the availability or not of servers to meet the demand for exchanging messages between devices.

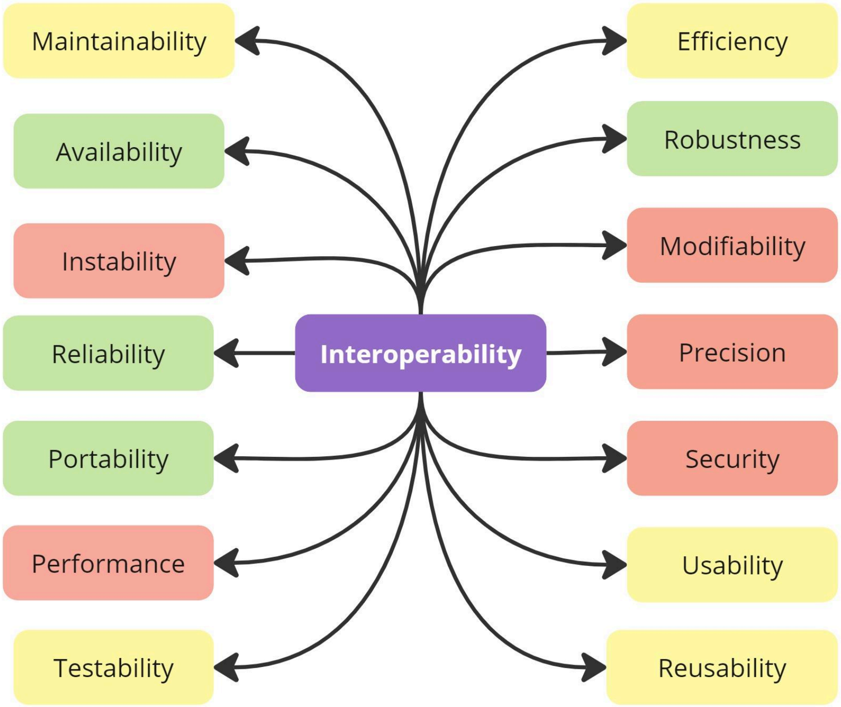


Figure 1 – Correlation of IoT characteristics with interoperability

The definitions for the 14 characteristics are presented below.

1. **Availability:** This is associated with the system's ability to remain operational and accessible when needed. High levels of availability favor continuous communication between distributed devices and services, contributing to effective data exchange (Li et al., 2018; Chen et al., 2021; Noura et al., 2019).

2. **Instability:** This relates to the occurrence of unexpected failures and inconsistent system operation. Unstable environments impair synchronization and integration between heterogeneous components, negatively affecting interoperability (Xiong et al., 2018; Noubir et al., 2020).

3. **Performance:** This refers to the system's ability to adequately respond to requests within acceptable timeframes. Excessive delays in data transmission can compromise the integration between IoT devices and services, especially in distributed

applications (Li et al., 2018; Chen et al., 2021).

4. **Portability:** This refers to the ease with which the system can be adapted to different platforms and environments. Portable systems promote interoperability by enabling their execution across multiple infrastructures and application domains (Noura et al., 2019; Sivashanmugam et al., 2014).

5. **Reliability:** This relates to the correct and consistent execution of system functions over time. Recurring failures directly impact the exchange of information between heterogeneous systems, compromising the expected joint operation (Xiong et al., 2018; Chen et al., 2021).

6. **Robustness:** This involves the system's ability to handle invalid inputs or adverse conditions. Robust architectures allow for continued communication even in the face of partial device failures, promoting interoperability (Noubir et al., 2020; Noura et al., 2019).

7. **Accuracy:** This is associated with the correct interpretation and transmission of information exchanged between devices. Inconsistencies in data representation can lead to interpretation errors and compromise integration between systems (Guinard et al., 2016; Sivashanmugam et al., 2014).

8. **Security:** refers to the protection of data and resources against unauthorized access. Although essential to ensure the reliability of information exchange, overly restrictive policies can hinder integration between interoperable systems (De Silva et al., 2017; Li et al., 2018; ISO 30141).

9. **Maintainability:** relates to the ease of making modifications and corrections to the system. More maintainable systems tend to adapt better to new interoperability requirements, such as changes in protocols or communication standards (Raza et al., 2017; Petrolo et al., 2018).

10. **Usability:** influences the configuration and operation of interoperable systems, especially by non-specialized users. Unintuitive interfaces can hinder the correct

parameterization of communication between devices (Alam et al., 2017; Tariq et al., 2018).

11. **Reusability:** This is associated with the use of standardized and reusable components, favoring integration with other systems and platforms (Guinard et al., 2016; Noura et al., 2019).
12. **Testability:** This refers to the ease of verifying the operation of interfaces and communication protocols. Greater testability allows for the early identification of interoperability failures, contributing to system reliability (Winkler et al., 2019; Kovatsch et al., 2018; Alam et al., 2017).
13. **Modefiability:** This refers to the ease of adapting the system to new standards and requirements. Systems that are difficult to modify face challenges in keeping up with the technological evolution of the IoT ecosystem (Xiong et al., 2018; Chen et al., 2021).
14. **Efficiency:** This relates to the optimized use of computational resources. Efficient systems promote interoperability by avoiding overloading devices; however, excessive optimizations can compromise support for multiple protocols and integration mechanisms (Li et al., 2018; Noura et al., 2019; ISO 30141).

By establishing the correlation between characteristics, it is possible to identify possible conflicts or incompatibilities between application requirements. For example, security may require encryption of data exchanged between devices, while portability improves or directly affects the ease with which a system can be transferred or adapted to different environments or platforms. In this case, it is necessary to find ways to mitigate these conflicts. The literature suggests using data translation techniques or communication protocols that allow encryption and still enable data interpretation.

In summary, the correlation of Interoperability characteristics based on ISO 30141:2018 can be a useful tool for identifying possible conflicts in IoT applications and for finding solutions that allow all requirements to be met satisfactorily.

3. Challenges of IoT Testing Interoperability

Interoperability, based on structural concepts such as context, perspective, purpose, level of support provided and system attributes, can be effectively measured, improved and monitored through a behavioral approach. This approach includes specific evaluation methods, identification and overcoming challenges, problem solving and analysis of the benefits resulting from interoperability (MOTTA et al., 2019).

Assessing interoperability in IoT applications involves ensuring that diverse devices and systems can communicate and operate together efficiently and effectively. The creation of a test environment that simulates real operating conditions, taking into account the diversity of communication protocols, data formats and security requirements, is essential to validate interoperability. Due to these complexities, several challenges have arisen in testing interoperability (CRUZ et al., 2020). Some challenges associated with evaluating the Interoperability characteristic in IoT applications were identified, as shown in Figure 2.

The adoption of common standards and technologies emerges as an effective means of ensuring that devices and systems can communicate and interact efficiently, thus establishing an adequate level of interoperability between them.

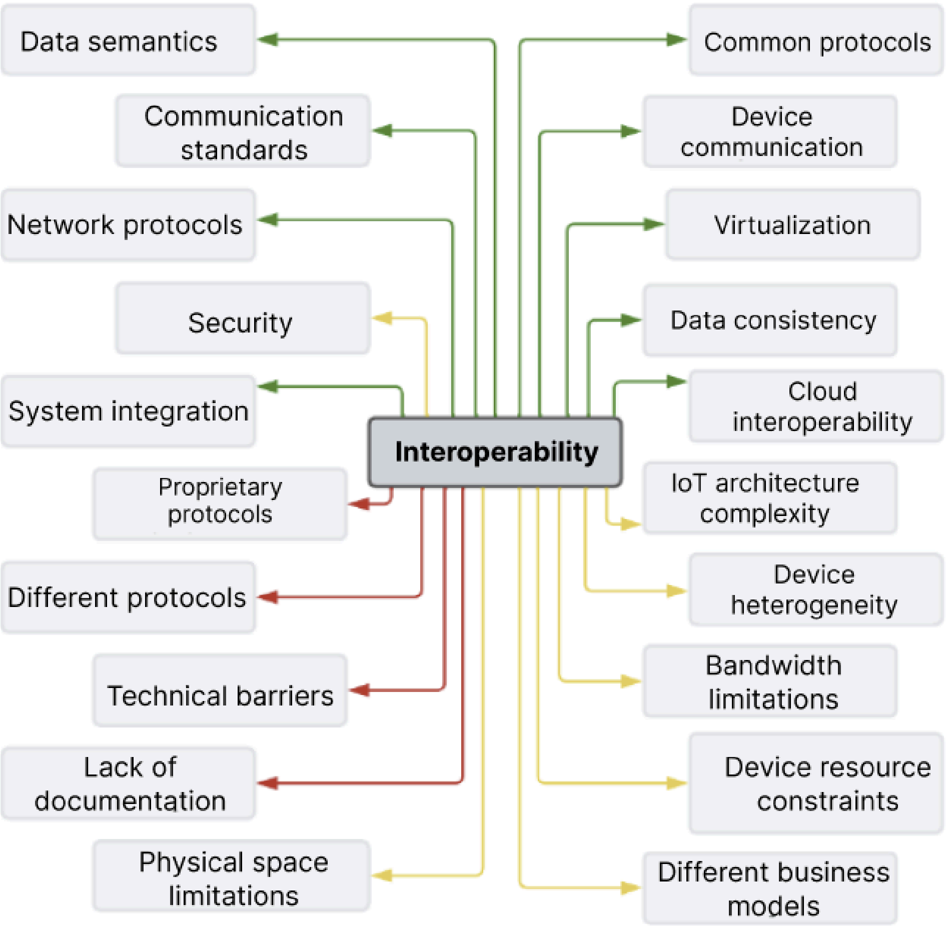


Figure 2 – Main challenges in interoperability testing

In this context, the red arrows are used to indicate the challenges considered most critical and frequently addressed in the literature. The green arrows, in turn, are used to highlight challenges that receive greater emphasis in research studies, whether due to their complexity or relevance. The yellow connections emphasize the challenges that have been significantly observed in practical IoT applications, indicating their importance in the implementation and effective functioning of these systems in the real world. The definitions are presented below:

- **Data Semantics:** are related to the interpretation of data exchanged between IoT devices. Interoperability requires a common understanding of data across devices and systems to ensure effective communication. In this context, testing activity must deal with the need to avoid ambiguities, ensure data consistency, and make sure all devices understand the data in the same way. The challenge is: "How to ensure a common understanding of data between IoT devices?" (Perera et al., 2014).

- **Network protocol:** are sets of rules that enable communication across an IoT network. Compatible protocols ensure that devices from different manufacturers can communicate effectively, essential for interoperability. In this context, testing activity must handle the diversity of protocols, ensure compatibility, and manage the complexity of integration. The challenge is: "How to ensure that different Network protocol are compatible with each other?" (Zanella et al., 2014).

- **Communication Standards:** Common standards facilitate integration and communication between different devices and systems, promoting interoperability. In IoT, heterogeneous devices operate using several protocols (e.g., MQTT, HTTP, and CoAP) and standards (e.g., Bluetooth and Zigbee). This diversity impacts the complexity of testing activities; for instance, most testing tools cannot interact properly with IoT applications, leading to challenges in test automation. One relevant issue posed by diverse communication standards is "How to ensure that IoT systems work correctly across all platforms and technologies?" (Bures et al., 2021; Atzori et al., 2010).

- **Security:** interoperability must ensure that communication between devices is secure and reliable, adhering to security standards to protect the information exchanged. The challenge posed by security in IoT devices is: "How to guarantee security in communication between IoT devices?" (Sicari et al., 2015).

- **Proprietary Protocols:** these types of protocols pose challenges to interoperability with devices from other manufacturers, creating technical barriers. In this context, testing activities must address limited compatibility with standard protocols, restricted technical information, and higher costs and complexity associated with customizing testing procedures. The main challenge is: "How to overcome the technical barriers imposed by proprietary protocols?" (Gubbi et al., 2013; Gulla et al., 2006).

- **Systems Integration:** is the connection and joint operation of different IoT systems and devices. Effective integration is critical for devices to work together seamlessly, promoting interoperability. In this context, testing activity must handle integration complexity, coordination among systems, and compatibility between different technologies. The challenge is: "How to facilitate the integration of IoT systems and devices?" (Bandyopadhyay et al., 2011).

- **Different Protocols:** the use of different communication protocols by distinct devices can create barriers to efficient communication, hindering interoperability. In this context, testing activity must handle protocol diversity, ensure compatibility, and manage the complexity of different technologies. The challenge is: "How to manage the diversity of communication protocols?" (Rayes & Salam, 2019).

- **Technical Barriers:** technological limitations that prevent effective communication between IoT devices can make it difficult to implement interoperable solutions. In this context, testing activity must handle technological restrictions, integration difficulties, and the need to develop innovative solutions. The challenge is: "How to overcome technical barriers to interoperability?" (Al-Fuqaha et al., 2015).

- **Lack of Documentation:** the lack of clear and detailed documentation about devices and protocols can hinder integration and communication between devices, affecting interoperability. In this context, testing activity must handle incomplete information, integration difficulties, and the need to develop its own documentation. The challenge is: "How to deal with the lack of documentation on devices and protocols?" (Vermesan et al., 2011).

- **Physical Space Limitations:** physical limitations affecting the installation and operation of IoT devices can influence connectivity and communication effectiveness between devices. In this context, testing activity must handle space restrictions, installation difficulties, and the need to optimize the use of available space. The challenge is: "How to mitigate physical space limitations when installing IoT devices?" (Khan et al., 2012).

- **Different Business Models:** variations in business models that influence the development and implementation of IoT devices can lead to proprietary solutions that hinder interoperability. In this context, testing activity must handle diverse business models, ensure solution compatibility, and manage the complexity of harmonizing different approaches. The challenge is: "How to harmonize different business models to promote interoperability?" (Porter & Heppelmann, 2014).

- **Device Resource Restrictions:** limitations in terms of power, memory, and processing capacity of IoT devices can restrict the communication capacity and functionality of devices, impacting interoperability. In this context, testing activity must handle resource constraints, the need for performance optimization, and the complexity of ensuring functionality with limited resources. The challenge is: "How to overcome the resource restrictions of IoT devices?" (Caldas et al., 2023).

- **Bandwidth Limitations:** restrictions on the amount of data that can be transmitted over an IoT network can affect communication effectiveness between devices. In this context, testing activity must handle bandwidth limitations, the need for transmission optimization, and the complexity of ensuring efficient communication. The challenge is: "How to manage bandwidth limitations in IoT networks?" (Palattella et al., 2013).

- **Device Heterogeneity:** the diversity of devices with different capabilities, protocols, and manufacturers increases the complexity of integration and effective communication, posing a challenge for interoperability. In this context, testing activity must handle device diversity, ensure compatibility, and manage integration complexity. The challenge is: "How to deal with the heterogeneity of IoT devices?" (Noura et al., 2019).

- **Complexity of IoT Architecture:** IoT system structures that include multiple layers and components can make coordination and communication between different components difficult, influencing interoperability. In this context, testing activity must handle architectural complexity, the need for layer coordination, and the difficulty of ensuring effective communication. The challenge is: "How to simplify the complexity of the IoT architecture to improve interoperability?" (Borgia, 2014).

- **Cloud Interoperability:** the ability of different cloud services to work together in an integrated way is crucial for the centralized management of IoT devices. In this context, testing activity must handle the diversity of cloud services, ensure compatibility, and manage the complexity of integration between different platforms. The challenge is: "How to ensure interoperability between different cloud services?" (Botta et al., 2016).

- **Data Consistency:** ensuring data is accurate and consistent across different devices and systems is essential. Inconsistent data can lead to misunderstandings and miscommunication between devices, affecting interoperability. In this context, testing activity must handle consistency verification, correction of inconsistent data, and the need to ensure data accuracy. The challenge is: "How to ensure data consistency between IoT devices?" (Zaslavsky et al., 2013).

- **Virtualization:** the use of virtualization technologies to manage hardware and software resources on IoT devices can facilitate or hinder interoperability, depending on the context. In this context, testing activity must handle virtualization implementation, ensure compatibility, and manage the complexity of virtualized resources. The challenge is: "How to use virtualization to improve interoperability?" (Kreutz et al., 2015).

- **Communication Between Devices:** the ability of IoT devices to exchange information efficiently and effectively is essential for interoperability. In this context, testing activity must handle protocol diversity, ensure compatibility, and manage the complexity of communication between different devices. The challenge is: "How to ensure effective communication between IoT devices?" (Sicari et al., 2013).

- **Common Protocols:** the use of standardized protocols that facilitate communication between different IoT devices is key to ensuring effective communication and integration. In this context, testing activity must handle protocol diversity, promote the use of common protocols, and ensure compatibility between different devices. The challenge is: "How to promote the use of common protocols between IoT devices?" (Shelby & Bormann, 2016).

4. Configuration of the Test Environment

To configure the environment, it is necessary to use some devices, such as:

- One or more IoT devices;
- Network infrastructure;
- One or more actuators;
- Application that will make decisions and send commands;

The present exposition aims to provide supplementary data intended to enrich the reader's understanding regarding the test environment configuration under consideration. The following information was provided with the purpose of improving the clarity and comprehensiveness of knowledge about the context of establishing the test environment.

IoT Devices: To test Interoperability in IoT, you need to have devices that can communicate with each other. These devices may include sensors, control devices, network devices, and other devices that are compatible with the chosen communication protocol. Examples of devices: Temperature sensors from manufacturers A, B, and C; lighting actuators from manufacturers X, Y and Z.

Network infrastructure: The network infrastructure must be configured to allow communication between IoT devices, ensuring their availability and security. This may involve configuring wireless networks (Wi-Fi and Bluetooth), switches, routers and firewalls. Example: Tests carried out on Wi-Fi networks, cellular networks, LPWAN (Low Power Wide Area Network) networks to evaluate Interoperability in different network environments.

Actuators: To test Interoperability in IoT, it is important to also consider actuators, which are devices capable of executing actions based on information received from sensors or commands from external systems. Examples of actuators in IoT: include actuators to open and close curtains, adjust the thermostat temperature, turn on or off electronic devices, etc. Application that will make decisions and send commands: An application or platform that makes decisions based on information from IoT devices and sends commands to control them. This application must be able to communicate with devices from different manufacturers and protocols.

In summary, to test Interoperability in IoT applications, it is essential to create a diverse environment that includes varied devices, protocols, manufacturers and network environments. This ensures that IoT devices can work harmoniously and communicate effectively in real-world scenarios, regardless of the technical differences between them.

Below, I present additional information aimed at improving the understanding of the guide in question, focusing on the assessment of Interoperability in the context of the Internet of Things (IoT):

Conducting Tests Locally: All testing procedures covered in this guide can be conducted in local environments, providing greater control and ease in evaluating Interoperability between IoT devices.

Relevance of Abstract Test Case s: It is important to highlight that Abstract Test Case s that do not directly apply to the specific system under evaluation can be left aside, in order to focus efforts on the most pertinent and significant Interoperability aspects for the context in question.

Integration of Tests and Measurements:carrying out the proposed Abstract Test Case s can be carried out simultaneously with the measurements, optimizing the evaluation process and providing a more comprehensive view of the effectiveness of interoperability between the evaluated IoT devices.

Prioritization Based on Cost-Benefit: The Abstract Test Case prioritization strategy can be based on the cost-benefit principle, directing attention to tests that offer greater Interoperability and, at the same time, are more viable in terms of resources and time.

Optional Use of Tools: Although tools can play an auxiliary role during testing and measurements, it is important to note that their use is not mandatory. Interoperability assessment can be carried out based on specific criteria, without depending exclusively on available tools.

Consideration of the Impact of Sub-characterístics:analyzing the impact of subcharacterístics proves to be a valuable approach to identifying Abstract Test Case s and measurements that can be performed together. This consideration contributes to a more holistic approach to Interoperability assessment, enabling a deeper understanding of the relationships between different elements of the IoT ecosystem.

5. Data Semantics

a. Definition

According to ISO 21823-4:2019 data semantics refers to the meaning and context of data. It is the ability to interpret and understand the meaning of data, allowing different systems and devices to share and use data in a consistent and efficient way.

b. Contextualization

Data semantics is an essential subcharacterístics of Interoperability in IoT, as it allows integration and effective communication between heterogeneous devices and systems. Standardization and the use of common semantic models are essential to ensure that data is interpreted correctly and consistently.

In the context of IoT, theData semantics refers to the ability to ensure that IoT devices and systems can share information efficiently and accurately, even if they use different data formats or terminologies.

Data semantics in IoT is important because it allows different devices to communicate and exchange information effectively. This is particularly important in scenarios where different IoT devices need to work together to accomplish a specific task. For example, imagine a smart lighting system that consists of multiple devices, including motion sensors, light switches, and lamps. Each of these devices needs to understand the information being shared to ensure lights are turned on or off when needed. For Data Semantics, 8 important properties were raised, which include:

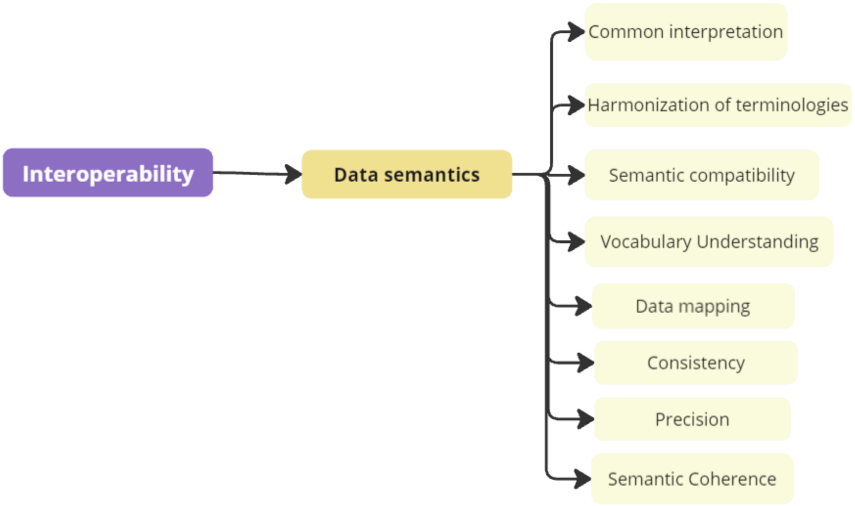


Figure 3 – Data semantic properties

- **P1- Common interpretation:**the ability to ensure that different systems can correctly understand and interpret the data that is shared between them. Examples of using the Common Interpretation property in IoT applications: Payment Application Interoperability, Mobile payment applications such as Apple Pay and Google Play use communication protocols that allow mobile devices to communicate with compatible payment terminals to ensure correct interpretation of transaction data (GUINARD, 2016).

- **P2- Harmonization of terminologies:**the ability to ensure that terms and concepts used to describe data are consistent across different systems, avoiding ambiguities and errors. For example, ensuring that different sensors use the same terminology to refer to location, temperature or humidity values (BANZI, 2015).
- **P3- Data mapping:**the ability to transform data from one format to another, allowing different systems to share information even if they use different standards or data structures. For example, converting temperature data in Fahrenheit to Celsius and km data to latitude and longitude (SIVASHANMUGAM, 2014)

- **P4- Semantic compatibility:**the ability to ensure that data shared between different systems can be easily combined and used together, even if it originates from different sources. For example, ensuring that different sensors understand that a temperature reading of 20 degrees Celsius means the same thing on different systems (Suryadevara et al. 2018).

- **P5- Consistency:**Consistency concerns the uniformity of data over time and across different systems. Consistent data ensures that information is not contradictory and can be used reliably (Redman, 1998).

- **P6- Accuracy:**Accuracy refers to the ability of data to accurately represent the concept it is intended to describe. In other words, accurate data is free from errors and ambiguities, ensuring a faithful representation of information (Welty and McGuinness, 2004)

- **P7- Vocabulary Comprehension:**The need for systems to share a common vocabulary or ontology to ensure mutual understanding of terms used in communication (Smith et al., 2004).

- **P8- Semantic Coherence:**The need for data to maintain semantic consistency across different contexts and over time. This ensures that the data is understood uniformly, regardless of variations in context or evolutions in the system (Smith et al., 2004).

c. Abstract Test Case s

Abstract Test Case 1- CT01	
Title	Data reading
Test environment	A network of heterogeneous IoT devices.
Precondition	Devices connected to the same Wi-Fi network
Step by step	1. Launch the mobile app 2. Select data reading 3. Check the data displayed on the screen
Postconditions	The data displayed on the mobile device screen must correspond to the same data requested by the actuator

Abstract Test Case 2- CT02	
Title	Traffic Monitoring by Latitude and Longitude
Test environment	Traffic monitoring system with latitude and longitude information
Precondition	Traffic monitoring system is in operation and collecting latitude and longitude data
Step by step	1. Access the traffic monitoring system control panel. 2. Select the option "View current traffic by Latitude and Longitude". 3. Verify that the traffic data displayed on the dashboard matches the location specified by latitude and longitude

	coordinates.
Postconditions	The traffic data displayed on the dashboard must correspond to the area specified by the given latitude and longitude coordinates.

Abstract Test Case 3- CT03	
Title	Common semantic models for data interpretation.
Test environment	A network of heterogeneous IoT devices.
Precondition	IoT devices are configured and operational.
Step by step	1. Start the interoperability testing platform. 2. Connect IoT devices to the test network. 3. Send test data generated by different IoT devices to the central server. 4. Check whether the data is correctly interpreted by common semantic models. 5. Perform read and write operations on IoT devices through the central server. 6. Check that operations are performed correctly and that devices respond appropriately. 7. Introduce variations in data formats or terminologies used by IoT devices. 8. Confirm that semantic models are capable of interpreting data even with variations.
Postconditions	The oven temperature must be changed according to the value selected on the control panel.

Abstract Test Case 4- CT04	
Title	Lighting control
Test environment	Location with intelligent lighting system

Precondition	Lighting system connected and working
Step by step	1. Launch the mobile app 2. Select the "Lighting" option 3. Select a specific room 4. Change light intensity
Postconditions	The light intensity must be changed in the selected room according to the value selected in the mobile application.

Abstract Test Case 5- CT05	
Title	Temperature and humidity monitoring
Test environment	Location with intelligent temperature and humidity sensor
Precondition	Temperature and humidity sensors connected and working
Step by step	1. Access the temperature and humidity monitoring system control panel 2. Select the option "View current temperature and humidity" 3. Check the temperature and humidity data displayed on the dashboard
Postconditions	The temperature and humidity data displayed on the dashboard must match the data collected by the sensors.

Abstract Test Case 6- CT06	
Title	Data Interpretation
Test environment	Devices from different manufacturers and communication protocols.

Precondition	IoT devices are physically installed and configured in the environment
Step by step	1. Send a message containing temperature data in JSON format from device A. 2. Ensure that device B correctly interprets received data. 3. Verify that device B has driven an actuator based on correct interpretation of the data.
Postconditions	Devices from different manufacturers and communication protocols were able to interpret and process data correctly, demonstrating interoperability in terms of data semantics.

d. Measurements

Device Capacity - M01	
Purpose	Assess the ability of different IoT devices to interact with each other effectively.
Method	Conduct interoperability tests with different combinations of IoT devices and measure the success rate of interactions.
Measure	Calculates the success rate as percentage (\%), where the number of successful interactions is divided by the total number of interactions and multiplied by 100 to obtain the \% representation
Explanation	This formula calculates the success rate in percentage, where the number of successful interactions is divided by the total number of interactions performed and multiplied by 100 to obtain the percentage representation. This formula will allow you to quantify the effectiveness of interoperability between IoT devices in your testing.
Reference	ISO 30141:2018 and Atzori et al. (2010)

Protocol Compliance - M02

Purpose	Evaluate the ability of IoT devices to interact with each other according to established protocols.
Method	Verify that IoT devices follow established protocols for interacting with each other, through compatibility and compliance tests.
Measure	1. **Number of supported protocols:**number of protocols the device supports 2. **Interaction success rate:**percentage of interaction attempts that were successful **MEASURE =**(Number of Supported Protocols / Total Number of Protocols) * 100
Explanation	This formula calculates the interaction success rate in terms of percentage, considering the number of protocols supported by the IoT device in relation to the total number of protocols tested. This provides a quantitative measure of device protocol compliance against established standards.
Reference	ISO/IEC TR 30128:2018 and Sahni (2019)

Platform compatibility - M03	
Purpose	Evaluate the ability of IoT devices to interact with different platforms.
Method	Verify that IoT devices are capable of interacting with different IoT platforms through compatibility and compliance testing.
Measure	1. **Number of supported platforms:**number of platforms the device is compatible with 2. **Interaction success rate:**percentage of interaction attempts that were successful **Success Rate (%) =**(Number of Successful Interactions / Total Number of Attempts) x 100
Explanation	This formula calculates the percentage of interaction attempts that were successful out of the total number of attempts performed during platform compatibility testing on IoT devices. The resulting measurement is expressed as a percentage.
Reference	ISO/IEC TR 30128:2018 and Sahni (2019)

Data integration - M04	
Purpose	Assess the ability of IoT devices to integrate and share data.
Method	Verify that IoT devices are capable of integrating and sharing data with other devices and systems, through compatibility and compliance testing.
Measure	**X** =Qty of data types / Data integration success rate
Explanation	In this formula, "Amount of Data Types Supported" represents the number of data types the device is capable of integrating and sharing, while "Data Integration Success Rate" is the percentage of data integration attempts that went well -successful. The result of this formula will provide a measure that evaluates the ability of IoT devices to integrate and share data, considering both the diversity of supported data types and the effectiveness of data integration.
Reference	ISO/IEC TR 30128:2018 and Sahni (2019)

6. Communication Protocol

a. **Definition.** Communication protocol refers to a set of technical specifications that define communication rules and standards to enable interoperability between devices in an IoT environment. These standards help ensure that different devices and systems can communicate and exchange information efficiently and reliably.

b. **Contextualization**According to ISO 30141:2018 in an IoT environment, it is common for devices and systems to be produced by different manufacturers and have different technical specifications. Without established communication standards, it can be difficult or impossible for these devices to communicate and share data. Therefore, Interoperability in IoT depends on the existence of and adherence to well-defined communication standards. For communication standards, 7 important properties that include:

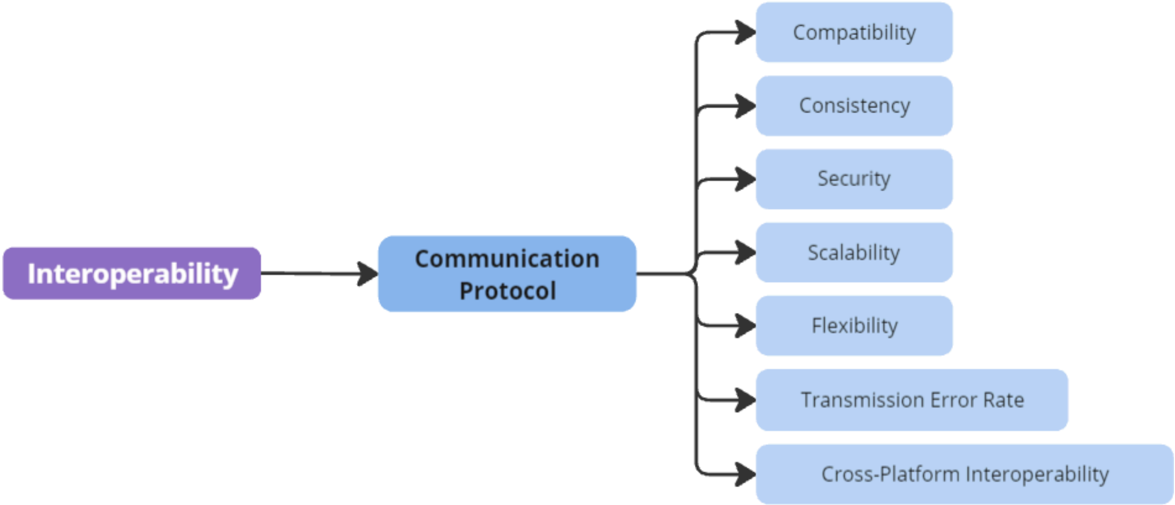


Figure 4 – Communication protocol properties

- **P9- Compatibility:**Communication standards must be compatible with a wide variety of devices and systems (SCHMIDT, 2015).
- **P10- Consistency:**Technical specifications must be clear and consistent to ensure that all devices and systems involved can communicate effectively (YAO et al., 2020).
- **P11- Security:**Communication standards must include robust security protocols to protect information transmitted between devices (DE SILVA et al., 2017).
- **P12- Scalability:**Protocols such as MQTT are scalable and allow IoT devices to be easily added or removed from a network without compromising the quality of communication (AL-FUQAHA et al., 2015).
- **P13- Flexibility:**Communication standards must be flexible to allow different types of devices and systems to be integrated into the network and to allow new technologies to be added to the network in the future (PETROLO et al., 2018).
- **P14- Cross-platform interoperability:**Communication standards must allow different platforms, such as operating systems, to communicate effectively, ensuring that information is shared reliably between devices and systems that use different platforms (CHENG et al., 2019).
- **P15- Transmission Error Rate-** Measures accuracy in message transmission, identifying the rate of errors that can affect interoperability (Suryadevara et al. 2018).

c. Abstract Test Cases

Abstract Test Case 7- CT07	
Title	Testing Communication Patterns in IoT Devices

Test environment	Two IoT devices from different manufacturers (Device A and Device B).
Precondition	Device A and Device B are connected to the IoT communication platform.
Step by step	1. Initiate communication between Device A and Device B through the IoT platform. 2. Send a test dataset containing data readings from Device A to Device B. 3. Verify that Device B correctly interprets the data and understands the meaning of the readings. 4. Repeat the process, but this time, send the test data from Device B to Device A. 5. Check that Device A correctly interprets data received from Device B.
Postconditions	Both devices (A and B) were able to correctly interpret and understand the data sent by the other. Confirmation that data semantics are being maintained in communication between devices from different manufacturers, ensuring effective interoperability in an IoT environment.

Abstract Test Case 8 - CT08	
Title	Communication Protocol Compatibility Test
Test environment	Network environment with IoT devices using different communication protocols
Precondition	IoT devices are configured to communicate with each other, but they use different protocols.

Step by step	1. IoT devices are activated and connected to the network. 2. Devices initiate communication with each other using their respective communication protocols. 3. Communication results are recorded and analyzed.
Postconditions	It is checked whether the IoT devices were able to communicate with each other successfully using their different communication protocols.

Abstract Test Case 9- CT09	
Title	Device connection test with gateway
Test environment	IoT Network with Gateway Device
Precondition	The device and gateway are connected to the same IoT network.
Step by step	1. The device sends a connection message to the gateway. 2. The gateway receives the message and sends a connection response to the device. 3. The device receives the response and confirms the connection to the gateway.
Postconditi ons	The device is connected to the gateway and ready to send and receive data on the IoT network

Abstract Test Case 10 - CT10	
Title	Testing communication between devices

Test environment	IoT network with two connected devices
Precondition	Devices are connected to the same IoT network
Step by step	1. Device 1 sends a message to device 2. 2. Device 2 receives the message and sends a response to device 1. 3. Device 1 receives the response and confirms communication between the devices.
Postconditions	Devices are communicating correctly on the IoT network.

Abstract Test Case 11 - CT11	
Title	Integration testing with external API
Test environment	IoT network with device connected to an external API
Precondition	The device is connected to the external API and has the necessary credentials to access it.
Step by step	1. The device sends a request to the external API. 2. The external API receives the request and sends a response to the device. 3. The device receives the response and confirms that the external API data was integrated correctly.
Postconditions	The data from the external API has been successfully integrated into the IoT network device.

Abstract Test Case 12- CT12	
Title	Authentication and security testing
Test environment	IoT network with authentication device and server
Precondition	The device is configured to authenticate with the authentication server.

Step by step	1. The device sends the authentication credentials to the server. 2. The server receives the credentials and authenticates the device. 3. The device receives authentication confirmation from the server and can access the IoT network
Postconditions	The device is authenticated and has secure access to the IoT network.

Abstract Test Case 13 - CT13	
Title	Security Testing of Communication Protocols on IoT Devices
Test environment	IoT Testing Lab
Precondition	IoT devices must be connected to the network and configured to communicate using the specified communication protocol.
Step by step	1. Attempt to intercept and decode communication between device A and device B. 2. Check whether it is possible to obtain unauthorized access to transmitted data. 3. Attempt to perform a denial of service (DoS) attack on devices A and B. 4. Verify that devices are able to handle the attack and continue to communicate. 5. Repeat steps 1 to 4 for different types of attacks and communication protocols.
Postconditions	IoT devices must be able to communicate securely using the specified communication protocol and resist different types of attacks.

d. Measurements

Data transfer rate - M05	
Purpose	Evaluate the communication capacity between devices in an IoT network.

Method	Sending a defined amount of data from one device to another and measuring the time required for the complete transfer.
Measure	Measurement: $X = D / T$ X = data transfer rate D = amount of data transferred T = time required for complete transfer
Explanation	This formula is used to calculate the rate at which data is transferred between devices in an IoT network based on the amount of data transferred and the time required for the complete transfer. It is a fundamental measure to evaluate communication performance in an IoT network.
Reference	ISO/IEC 29182-1:2017

Number of communication failures - M06	
Purpose	Evaluate the reliability of communication between devices in an IoT network.
Method	Perform a set of data transfers between devices in an IoT network and count the number of communication failures that occur.
Measure	$R = (N / (T * G))$ N is the total number of communication failures. T is the total observation time (e.g. in hours). G is a measure of failure severity (e.g., a scale of 1 to 10, where 1 represents minor failures and 10 represents severe failures).
Explanation	This measure takes into account the frequency (number of failures), duration (total observation time) and severity of failures to calculate the reliability of communication in an IoT network. The lower the R value, the more reliable the communication
Reference	ISO/IEC 29341-8-20:2016

Network latency - M07	
Purpose	Evaluate IoT network response time.

Method	Send a message from one device to another and measure the time required to receive the message.

Measure	X = t2 - t1
	X = network latency
	t1 = message sending time
	t2 = message reception time

Explanation	This formula calculates the difference between the message reception time (t2) and the message sending time (t1) to determine the network latency in an IoT application.
Reference	ISO/IEC 30141:2018

Connection Success Rate - M08	
Purpose	Evaluate the effectiveness of communication standards in establishing connections between devices in an IoT network.
Method	Attempt to establish a connection between devices in an IoT network and count the number of successful connections.
Measure	X = number of successful connections / total number of connection attempts
Explanation	This formula represents the proportion of connections that were successfully established in relation to the total number of connection attempts made. It is an important metric for evaluating the effectiveness of communication patterns in an IoT network in terms of establishing successful connections.
Reference	ISO/IEC 29341-8-20:2016

Network latency variation - M09	
Purpose	Evaluate the stability of the IoT network in relation to response time.
Method	Send a message from one device to another at regular intervals and measure the variation in network latency over time.
Measure	X = (n/N) * 100, X is the percentage of data received

	correctly **n**is the number of data received correctly **N**is the total number of data sent.
Explanation	This formula allows you to evaluate the stability of the IoT network in relation to response time, providing a measure of the quality of communication between devices on the network. The higher the X percentage, the better the network stability in terms of network latency.
Reference	ISO/IEC 29182-1:2017

8.System integration

a. Definition

The ISO 30141:2018 standard establishes system integration as the ability of systems to establish connections, carry out communications and interactions with other systems and services, with the purpose of providing value-added information and services. This capability plays a crucial role in achieving Interoperability in the Internet of Things, which is the ability of devices and systems from different vendors to communicate and interact transparently with each other.

This standard recognizes the importance of establishing a harmonious interconnection between systems and services, allowing the exchange of information and the provision of higher quality services. Furthermore, by promoting Interoperability, the ISO 30141:2018 standard seeks to facilitate the integration of heterogeneous devices and systems, ensuring fluid communication and efficient interaction between them.

b. Contextualization

Systems integration is one of the fundamental facets of*Interoperability*, representing one of the main challenges faced in the context of IoT. The Internet of Things consists of a wide range of devices and systems that must work together to provide value-added solutions. As mentioned by Xiong et al. (2018), achieving this objective requires ensuring the ability of systems to connect and interact with each other.

A practical example of system integration can be seen in a smart home, where multiple devices and systems, such as lighting, thermostats, security cameras, and appliances, collaborate to deliver a connected home experience. To make this possible, it is crucial that these systems are able to connect, communicate and interact efficiently and reliably, regardless of the vendor or platform used.

Ensuring system integration is essential to enable the exchange of information and sharing of resources between the devices and systems involved. This allows the different IoT components to work together harmoniously, promoting an improved experience for end users and ensuring the viability of innovative solutions in the context of the Internet of Things.

Some properties of system integration in IoT include (Chen et al., 2021):

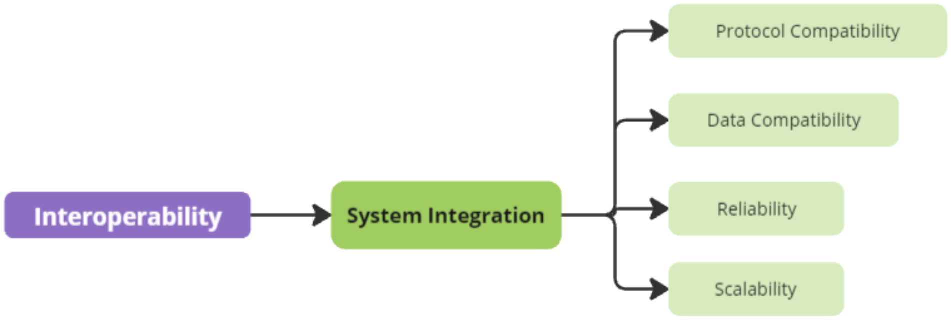


Figure 5 – System integration properties

- **P16- Protocol compatibility:**Different devices and systems must be able to communicate with each other, regardless of the protocol used.
- **P17- Data compatibility:**Different systems must be able to interpret and use data generated by other systems.
- **P18- Reliability:**Integrated systems must be able to function reliably despite network or device failures or outages.
- **P19- Scalability:**Embedded systems must be able to handle a large number of devices and users.
- **P20- Flexibility:**Integrated systems must be able to adapt to changes in environmental conditions or user needs.

c. Abstract Test Case s

Abstract Test Case 14 - CT14	
Title	Interoperability testing between devices
Test environment	IoT device network
Precondition	All devices on the network are properly configured and connected.
Step by step	1. Start the automation system 2. Send a wake command to device A using the X protocol.

	3. Verify that device A responds correctly and performs the expected action. 4. Send a wake command to device B using protocol Y. 5. Verify that Device B responds correctly and performs the expected action. 6. Make sure there are no conflicts or communication problems between devices using different protocols.
Postconditions	Devices must be able to communicate and exchange information in an interoperable way.

Abstract Test Case 15 - CT15	
Title	this of Reliability in an Industrial Environment
Test environment	Industrial automation system
Precondition	Various devices and automation systems from different suppliers are integrated to control the manufacturing process.
Step by step	1. Simulate an outage in the network that connects devices and systems. 2. Observe how devices and systems respond to interruption, whether they continue to function or enter a failed state. 3. Restore the network connection and verify that devices and systems recover and function normally again. 4. Repeat the process, but this time, simulate a failure in one of the devices. 5. Record how the integration reacts to device failure and whether redundancy or backup systems come into play.
Postconditions	Assess the reliability of the industrial automation system in dealing with network interruptions and device failures, identifying areas for improvement in ensuring operational continuity.

Abstract Test Case 16 - CT16	
Title	Scalability Testing in a Retail Environment
Test environment	Inventory management system in a chain of retail stores.
Precondition	The inventory management system is used in multiple stores and needs to handle a large number of products and transactions.
Step by step	1. Gradually increase the number of stores using the inventory management system. 2. Carry out product entry and exit operations in all stores simultaneously. 3. Monitor system performance, including response time and processing capacity. 4. Further increase the number of stores and repeat operations. 5. Record at what point the system begins to show signs of overload or reduced performance.
Postconditions	Evaluate the scalability of the inventory management system in dealing with a large number of stores and transactions, identifying the limits of its capacity and scaling needs.

Abstract Test Case 17 - CT17	
Title	Protocol Compatibility Test
Test environment	Smart city environment with multiple devices from different vendors, including traffic sensors, street lighting systems, and environmental monitoring systems.
Precondition	All devices are working properly and are connected to a network.
Step by step	1. Select two devices from different vendors that use different communication protocols. 2. Set up a scenario where these devices need to communicate to coordinate traffic management in an area of the city. 3. Start communication between devices and check if they are capable of exchanging information and coordinating their

	actions. 4. Check that there is no data loss or communication errors during the interaction. 5. Repeat testing with different devices and protocols to assess overall protocol compatibility capability.
Postconditions	Devices from different vendors are able to communicate effectively and coordinate their actions, demonstrating that protocol compatibility is being met.

Abstract Test Case 18- CT18	
Title	Scalability and Flexibility Test
Test environment	A network of traffic monitoring devices in a metropolitan area with a large number of devices and variations in traffic conditions.
Precondition	The network is operating with a moderate number of devices and normal traffic conditions.
Step by step	1. Gradually increase the number of devices on the network to see how it handles scalability. 2. Introduce variations in traffic conditions, such as sudden congestion or accidents, to test the network's flexibility. 3. Evaluate network performance as the number of devices and complexity of traffic conditions increase. 4. Check whether the network is capable of adapting to changes in traffic environment conditions. 5. Repeat the test at different scales and with different traffic scenarios.
Postconditions	The network of traffic monitoring devices demonstrates scalability and flexibility, meeting established requirements.

d. Measurements

System integration success rate - M10	
Purpose	Evaluate the effectiveness of service interoperability in an IoT environment.
Method	Record the number of service interoperability attempts and the number of successful attempts during a specific period of time.
Measure	X = (n1 / n2) x 100% X = success rate in systems integration n1 = number of successful service interoperability attempts during the evaluated period n2 = total number of service interoperability attempts during the evaluated period
Explanation	This measure calculates the effectiveness of service interoperability in an IoT environment, representing the percentage of interoperability attempts that were successful in relation to the total number of attempts made. The higher the value of X, the higher the success rate in systems integration, indicating better performance in service interoperability.
Reference	ISO/IEC 30141:2018

Average system integration time - M11	
Purpose	Evaluate the average time required to integrate systems in an IoT environment.
Method	Record the time elapsed from the start of service interoperability to its successful completion.
Measure	X = (Σ t) / n x = average systems integration time t = time required for successful service interoperability n = total number of service interoperability attempts during the evaluated period
Explanation	This formula calculates the average time spent on all service interoperability attempts during a given period. It provides a quantitative measure of the average time required to integrate systems in an IoT environment, based on attempts made and their respective successful completion times. The smaller the value of X, the more efficient the

	systems integration process is in relation to time.
Reference	ISO/IEC 30141:2018

Integration interface standardization level - M12	
Purpose	Assess the degree of standardization of service interoperability interfaces in an IoT environment.
Method	Verify that service interoperability interfaces comply with standards defined for the IoT environment.
Measure	X = level of standardization of integration interfaces n1 = number of service interoperability interfaces that comply with defined standards n2 = total number of service interoperability interfaces evaluatedX = (n1 / n2) x 100%
Explanation	This formula calculates the percentage of interfaces that conform to the defined standards out of the total number of interfaces evaluated. The result, X, represents the degree of standardization of integration interfaces in an IoT environment, indicating how well the interfaces are aligned with the standards established by the ISO/IEC 30141:2018 standard. The higher the value of X, the greater the level of standardization of the interfaces.
Reference	ISO/IEC 30141:2018

Data transfer rate - M13	
Purpose	Evaluate the communication capacity between interconnected systems in IoT.
Method	Measure the amount of data transmitted between systems in a given period of time.
Measure	X = (T2 - T1) / S X = data transfer rate T1 = initial data transfer time T2 = end time of data transfer S = amount of data transmitted

Explanation	This formula allows you to calculate the data transfer rate, which is a fundamental measure for evaluating the performance of communication between interconnected systems in the IoT. The higher the transfer rate, the more efficient the communication between systems, which is essential to ensure the proper functioning of the Internet of Things.
Reference	ISO 30141:2018.

Integration time - M14	
Purpose	Evaluate the time required for the integration of different IoT systems.
Method	Account for the beginning of the integration process and compare it with the time when the integration is completed.
Measure	X = t2 - t1, where X is the integration time, t1 is the start time of the integration process and t2 is the time at which the integration is completed.
Explanation	This simple formula subtracts the start time from the completion time to determine the time required to perform IoT systems integration. The result X is the integration time, which can be expressed in the same time unit used for t1 and t2 (e.g. seconds, minutes, hours, etc.).
Reference	ISO/IEC 30141:2018.

9. Network protocol

a. **Definition**ISO 30141:2018 defines Network protocol as an essential element of Interoperability in IoT systems, which allows communication between devices, services and applications on a network. A network protocol defines rules and standards for data exchange, synchronization and coordination of operations between devices, allowing Interoperability and integration of heterogeneous systems.

b. **Contextualization**

With the increase in the number of devices and systems connected to the Internet of Things (IoT), interoperability between them has become a fundamental requirement to ensure communication and information exchange. Network protocol are the basis for ensuring Interoperability in IoT systems, allowing devices and services from different vendors and technologies to communicate and interoperate. Network protocol have several properties, such as:

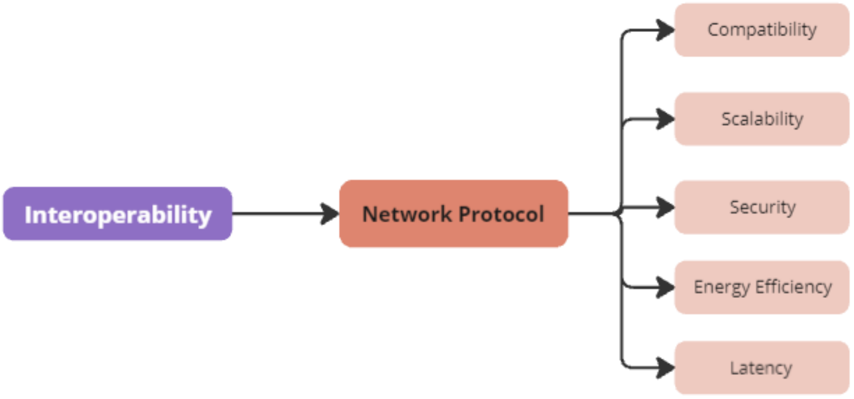


Figure 6 – Network protocol properties

- **P21- Compatibility:**protocols must be compatible with the hardware and software specifications of the connected devices.
- **P22- Scalability:**Protocols must allow expansion of the IoT system to support a large number of devices and users.
- **P23- Security:**Protocols must ensure the security of communication between devices, including authentication, authorization and data encryption.
- **P24- Energy efficiency:**Protocols must be optimized to reduce power consumption of connected devices while maximizing battery life.
- **P25- Latency:**Protocols must ensure response times fast enough to enable real-time control of devices.

c. **t Abstract Test Case s**

Abstract Test Case 19 - CT19	
Title	Network Protocol Compatibility Test
Test environment	IoT system with devices from different suppliers connected through different Network protocol.
Precondition	Devices are configured correctly and connected to the network
Step by step	1. Check that all devices are connected and working properly. 2. Send data from one device to another via network protocol. 3. Verify that the data was received correctly by the target device.
Postconditions	The data was successfully transmitted and the devices were able to interoperate using the corresponding Network protocol.

Abstract Test Case 20 - CT20	
Title	Network Protocol Scalability Test
Test environment	IoT system with a large number of devices connected via a specific network protocol.
Precondition	The system is configured correctly and the network protocol supports the number of connected devices
Step by step	1. Gradually add devices to the network until you reach the limit supported by the network protocol. 2. Send data between connected devices to verify communication integrity. 3. Verify that the data was received correctly by the target devices.
Postconditions	The network protocol supported the number of connected devices and the data was transmitted successfully

Abstract Test Case 21- CT21	
Title	Network Protocol Security Test
Test environment	IoT system with devices connected through Network protocol that support security characterístics.
Precondition	The security characterístics of the Network protocol are configured correctly.
Step by step	1. Send data between connected devices via network protocol. 2. Verify that data was transmitted securely using encryption, authentication, and data authorization. 3. Attempt to break into the IoT system through known security vulnerabilities.
Postconditions	Data was transmitted securely and no security vulnerabilities were

	successfully exploited.

Abstract Test Case 22- CT22	
Title	Network Protocol Energy Efficiency Test
Test environment	IoT system with devices connected via Network protocol that support energy efficiency characterístics
Precondition	The energy efficiency characterístics of the Network protocol are configured correctly.
Step by step	1. Monitor power consumption of connected devices. 2. Send data between connected devices via network protocol. 3. Check whether the devices' power consumption has been optimized by the network protocol.
Postconditions	The power consumption of devices has been optimized and data has been successfully transmitted over the network protocol.

Abstract Test Case 23 - CT23	
Title	Protocol Compatibility Test
Test environment	A network of IoT devices in a smart city with different devices from different manufacturers.
Precondition	All devices are properly configured and operational on the network.
Step by step	Select two IoT devices from different vendors, each using a different network protocol. Configure a scenario in which these devices need to communicate and exchange information to coordinate a specific action in the smart city. Start communication between devices and check if they are capable of exchanging information and coordinating their actions. Ensure there is no data loss or communication errors during the

	interaction. Repeat testing with different devices and protocols to assess the overall protocol compatibility capability of the IoT network.
Postconditions	IoT devices from different vendors are able to communicate effectively and coordinate their actions, demonstrating that protocol compatibility is being met in the IoT network

Abstract Test Case 24 - CT24	
Title	Protocol Compatibility Testing for IoT Asset Tracking
Test environment	IoT asset tracking environment, with tracking devices from different manufacturers and technologies, all sending latitude and longitude data.
Precondition	Tracking devices are operating correctly and sending latitude and longitude data over different protocols
Step by step	1. Select two tracking devices from different vendors that use different communication protocols. 2. Configure a scenario where these devices need to share latitude and longitude data for real-time asset tracking. 3. Start communication between devices and verify that they are capable of exchanging latitude and longitude information without data loss. 4. Check whether the data is interpreted correctly by receiving devices. 5. Test the real-time responsiveness of devices by measuring latency in transmitting and receiving location data. 6. Repeat testing with different devices and protocols to assess overall protocol compatibility for IoT asset tracking.
Postconditions	IoT asset tracking devices from different vendors are able to exchange latitude and longitude information effectively and in real-time, demonstrating the compatibility of protocols for IoT asset tracking.

Abstract Test Case 25 - CT25	
Title	Security Testing Using HTTPS
Test environment	IoT network with two or more HTTPS-capable devices
Precondition	The devices must be connected to the same IoT network and configured to use the HTTPS protocol. Each device must be capable of sending and receiving HTTPS requests and must have valid SSL/TLS certificates installed.
Step by step	1. Send an HTTPS request to a selected device and check whether communication is established successfully. 2. Send an HTTPS request to a device using an invalid SSL/TLS certificate and check whether communication is blocked. 3. Try intercepting and deciphering HTTPS traffic using a packet interception tool and checking whether communication is blocked or data is unreadable. 4. Attempt to send false or malicious requests to devices and verify that authentication and authorization measures are applied correctly.
Postconditions	All devices must be capable of establishing secure HTTPS communication, with authentication and authorization applied correctly, ensuring network security.

c. Measurements

Network Latency - M15	
Purpose	Evaluate network response time during communication between devices.
Method	Send a data packet from device A to device B and measure the time it takes for device B to receive the packet and send a response to device A.
Measure	X = t2 - t1X= is the network latency, which is the time taken for a data packet to be sent from device A to device B and for the response to be sent back from device B to device A. **t1=**is the time when device A sent the data packet.

	t2=is the time when device B sent the response.
Explanation	This measure calculates the time difference between sending the data packet by device A and receiving the response by device A, which represents the network latency time. The lower the value of X, the lower the network latency, which indicates faster and more responsive communication between devices. The measurement is expressed in time units, such as milliseconds (ms) or microseconds (µs), depending on the precision required to evaluate network latency.
Reference	ISO/IEC 30141:2018 - Information technology -- Internet of Things (IoT) -- Part 4: Communication protocols.

Network Throughput - M16	
Purpose	Assess the amount of data that can be transmitted between devices in a given period of time.
Method	Send a set of data of known size from device A to device B and measure the time required for complete transmission.
Measure	X = S/t, where X is the network throughput, S is the size of the data sent, and t is the time required for complete transmission.
Explanation	This measure expresses how much data can be transmitted from a device A to a device B in a given period of time, representing the efficiency of the communication network in question. The higher the value of X, the faster the network throughput. This measure is useful for evaluating the performance and communication capacity of devices and systems, being an important metric in contexts such as the Internet of Things (IoT).
Reference	ISO/IEC 30141:2018 - Information technology -- Internet of Things (IoT) -- Part 4: Communication protocols.

Data transmission rate - M17	
Purpose	Assess the ability of the network protocol to transmit data at a given speed.
Method	Send a data set of known size from device A to device B and measure the data transmission speed.
Measure	X = S / t, where X is the data transmission rate, S is the size of the data sent, and t is the time required for complete transmission.

Explanation	This formula allows you to calculate the speed at which data is transmitted from the source (device A) to the destination (device B). Data transmission rate is a critical metric for evaluating the interoperability of a network protocol, as it indicates the efficiency of transmission in relation to the size of the data and the time required for delivery. The higher the transmission rate, the faster the data transfer between devices. This measurement helps determine whether a network protocol is suitable for the specific application in question.
Reference	ISO/IEC 30141:2018 - Information technology -- Internet of Things (IoT) -- Part 4: Communication protocols.

Connection reliability - M18	
Purpose	Assess the ability of the network protocol to maintain a stable and reliable connection between devices.
Method	Send a set of data from device A to device B over an unstable network connection and verify that all data was received correctly.
Measure	X = (n/N) * 100X is the percentage of data received correctly. n is the number of data received correctly. N is the total number of data sent.
Explanation	This formula calculates the percentage of data that was transmitted successfully out of the total number of data sent. The higher the value of X, the greater the reliability of the connection, indicating that the connection was able to maintain the integrity of the data sent more effectively. This is especially important in unstable network environments where connection reliability is critical to ensure data is delivered accurately.
Reference	ISO/IEC 30141:2018 - Information technology -- Internet of Things (IoT) -- Part 4: Communication protocols.

System scalability - M19	
Purpose	Assess the ability of the network protocol to support a large number of simultaneously connected devices.
Method	Add a large number of devices to the system and verify that the network protocol is capable of supporting the data traffic generated by all the devices.

Measure	X = N, where X is the system scalability and N is the maximum number of simultaneously connected devices that the network protocol can support.
Explanation	This measurement is direct and simple, indicating the maximum number of devices that the network protocol is capable of supporting simultaneously before reaching its capacity limit. The higher the value of N, the greater the scalability of the system, indicating the network protocol's ability to handle a large number of connected devices without significant degradation in performance. This measurement is useful for evaluating system capacity in Internet of Things (IoT) scenarios, where many devices may be connected to the network at the same time.
Reference	ISO/IEC 30141:2018 - Information technology -- Internet of Things (IoT) -- Part 4: Communication protocols.

10. Impact of Sub-Characteristics

The impacts of these sub-characterístics on IoT Interoperability can be significant. Lack of Interoperability can result in incompatible systems that cannot communicate with each other, which can limit the efficiency and effectiveness of IoT. Additionally, a lack of security can lead to privacy breaches and vulnerabilities, while a lack of common interpretation of data can result in miscommunication. In summary, interoperability is a critical characterístics of IoT and related sub-characterístics must be managed properly to ensure that IoT devices can communicate and work together efficiently and securely.

Interoperability is a critical characteristic of IoT, distinguishing itself from other characteristics due to the interdependence of subcharacteristics. In other words, each subcharacteristic influences the others. This implies that the absence of one sub-characteristic can have adverse impacts on the others, thus highlighting the importance of fully addressing the Interoperability sub-characteristics during the validation process according to the application context. This approach can be implemented through joint tests or measurements of the subcharacteristics, or through the implementation of solutions that address the properties affected by the other subcharacteristics.

To establish the interactions between the four Interoperability subcharacteristics, it is essential to analyze which of them influence the properties of the other subcharacteristics. These impacts will be presented based on the properties of the four subcharacteristics. shows the correlations of the "Data Semantics" subcharacteristics with the properties of the other three subcharacteristicss. The colors represent three categories: (i) green to represent Systems Integration"; (ii) blue denotes Communication Protocol"; and (iii) red to represent the ``Network Protocol". Figure 7 illustrates the impact of the properties of the “Data Semantics” sub-characteristics. When evaluating a specific property, it is necessary to consider the evaluation of the related properties of the other sub-characteristics. For example, when evaluating "Common Interpretation", "Protocol Compatibility", "Data Compatibility", "Consistency" and "Compatibility" must be taken into consideration. Likewise, the evaluation of "Semantic Compatibility" involves considering "Reliability" and "Scalability". Thus, the analysis of "Data Semantics" properties demands an integrated approach, where the interrelationships between sub-characteristicss and their respective properties are carefully examined to ensure a comprehensive and accurate assessment.

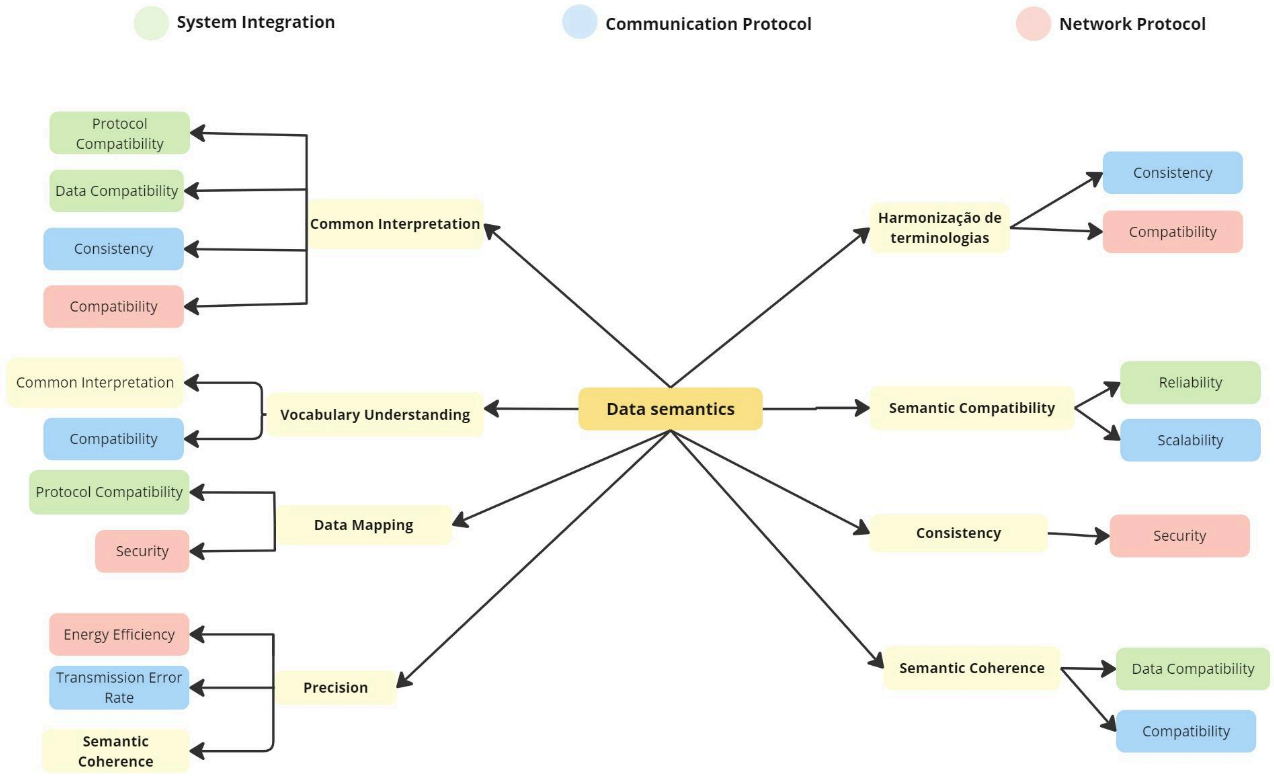


Figure 7 – Impact of data semantics with the properties of other subcharacteristics

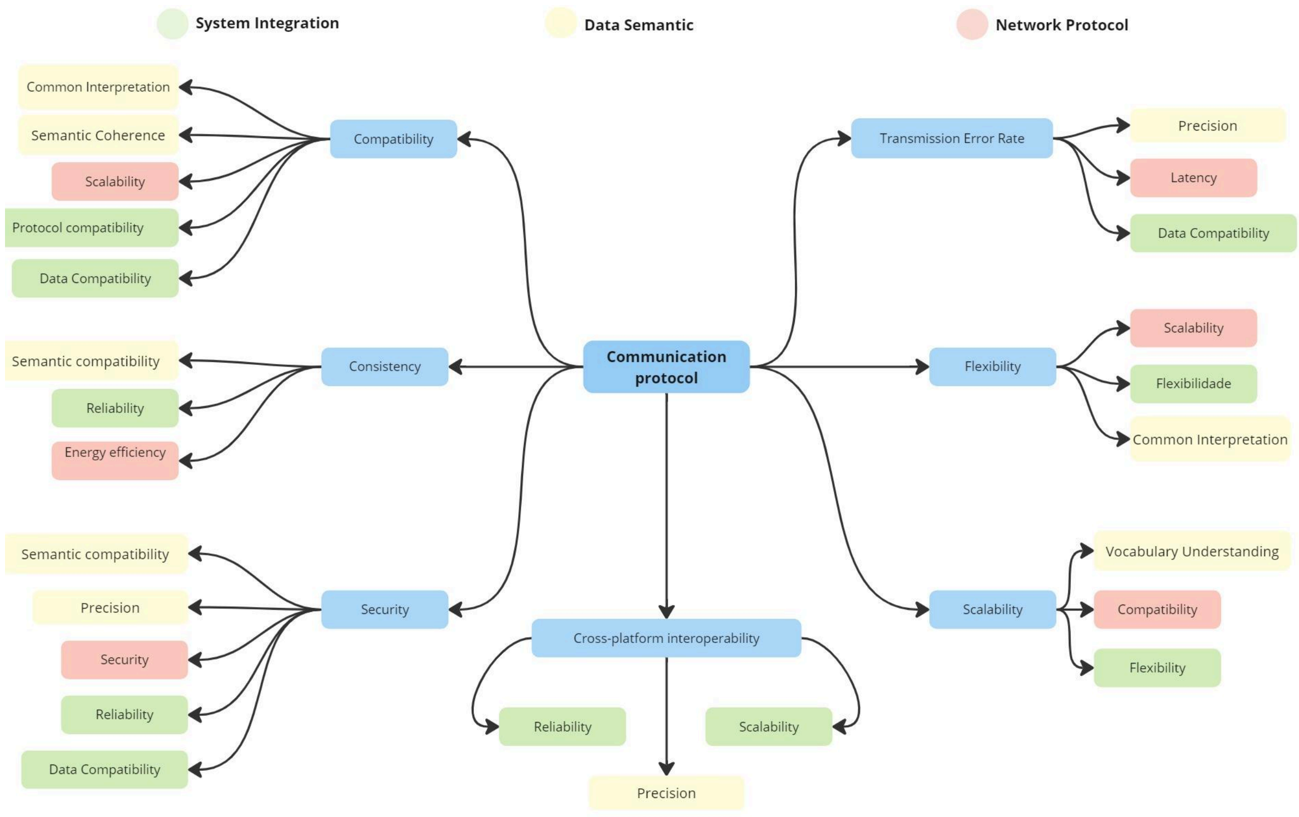


Figure 8 – Impact of communication protocol with the properties of other subcharacteristics

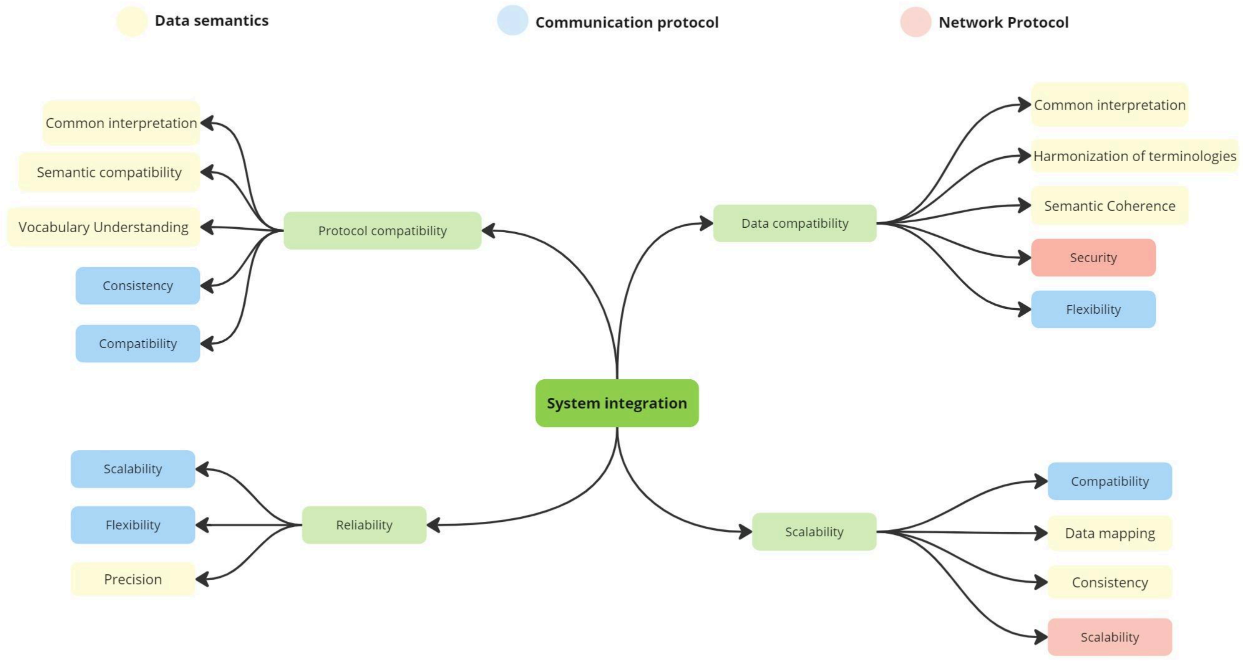


Figure 9 – Impact of System Integration with the properties of other sub-characteristics

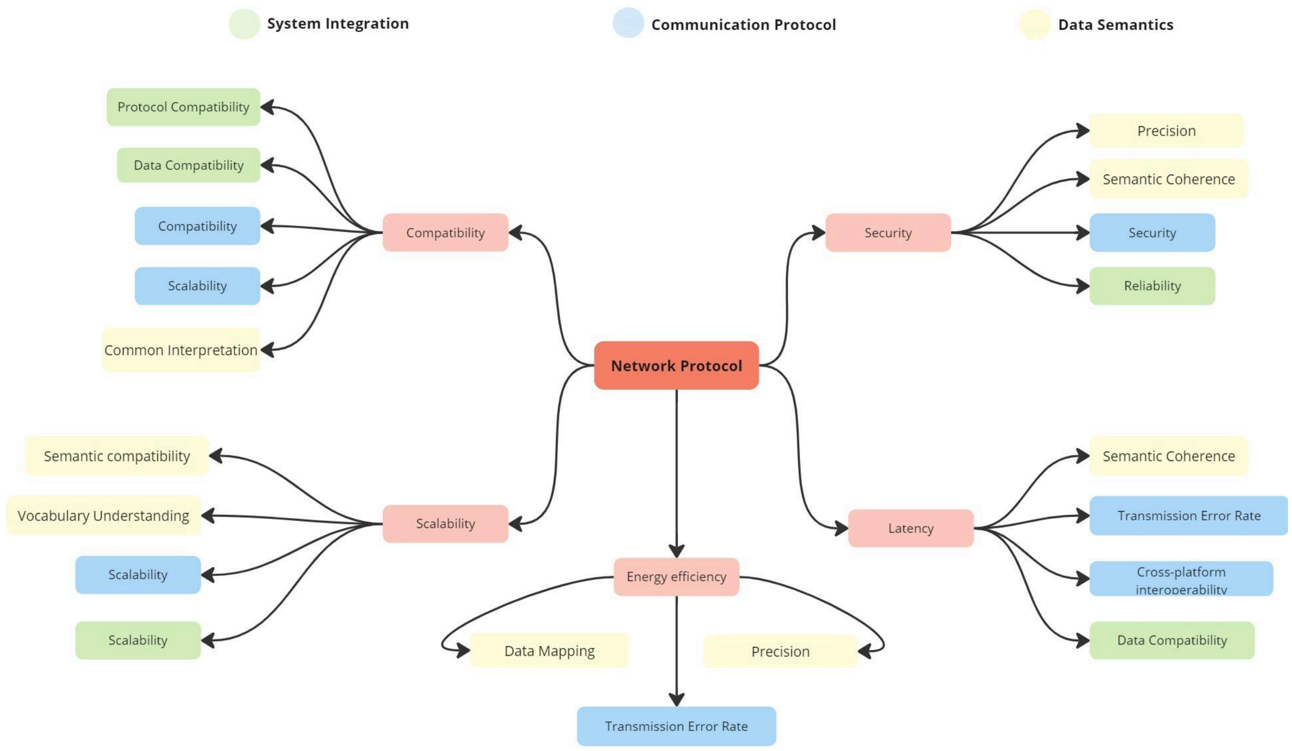


Figure 10– Impact of network protocol with the properties of other subcharacteristics

It is important to note that impacts may vary depending on specific implementations and circumstances, but these are some of the general trends that can be observed based on the properties presented.

11. Cost-Benefit

Based on the correlation of a characteristic with the others, presented in topic 2, it is possible to define the impact that a characteristic has on the applications. Interoperability, for example, is correlated with 14 characteristics, but not all applications prioritized all 14 characteristics. Therefore, given that an application prioritizes only 6 IoT characteristics that correlate with interoperability, then the impact coefficient (CI) is approximately 0.43, that is:

CI = ORC/RC

Where:

ORC: number of characteristics correlated with interoperability prioritized in the application.

RC: total number of characteristics related to interoperability.

This impact (CI) when related to the effort in executing tests and metrics generates the cost benefit that can help in prioritizing tests. To calculate this effort, the following formulas are applied:

1st Calculate the estimated cost of each test case based on the average time for the professional responsible to execute a test and the value of the time of the professional responsible to execute it;

CTi = TCi * VHCi

CTi : Estimated cost to execute the test case

TCi : Average time of the professional to execute a test case

VHCi : Value of the time of the professional who will execute the test case

2. After the completion of all CTs, the maximum value found is obtained;

MCT = max(CT)

MCT: highest cost to perform the test case

CT: all cost estimates

3. Normalize the average costs of the test cases:

$$CCT = (\sum_{i=1}^{n} CT_i/MCT)/n$$

CCT: average cost of normalized test cases

CTi /MCT: estimated value of the cost of test case i normalized to the highest cost

n: number of test cases

4. Repeat the process for the measurements and thus obtain the CMD;

CMD: average cost of standardized measurements;

5. Thus, the Effort (ESF) is defined by:

$$\mathbf{ESF = (CCT + CMD)/2}$$

With the Impact (CI) and the Effort (ESF), the cost-benefit can be defined in which quadrant of prioritization the application interoperability tests are performed. The x-axis shows the Impact (CI), and the y-axis shows the Effort (ESF). An Effort above 0.5 is considered high, the same for Impact. Therefore, in an application where the impact coefficient (CI) for interoperability is 0.68, the interoperability characteristic is considered a characteristic that has a high impact on that IoT application in question. The groups represent:

Group I: High effort and low impact. High cost and low benefit = low priority - the effort to run the tests is very high and not running them impacts some correlated characteristics. It should be assessed whether the few characteristics that are impacted are essential to the system. If so, even with the high effort, priority should be given to conducting the tests and using tools and relationship tables to reduce the effort during execution. If the impacted characteristics are not essential to the system, there is no need to run the tests immediately.

Group II: Low effort and low impact. Low cost and low benefit = medium priority - the effort to run the tests is low and not running them impacts some correlated characteristics. It should be assessed whether the few characteristics that are impacted are essential to the system. If so, they should be prioritized to run the tests. Otherwise, there is no need to run the tests immediately. However, since running the tests requires low effort, these tests can be performed easily and more completely in a timely manner using the Impact of Subcharacteristics section, adding even more properties to be evaluated.

Group III: High effort and high impact. High cost but high benefit = high priority - the effort to run the tests is very high and not running them impacts many correlated characteristics, the tests should be conducted and should use the strategies in the guide, such as tools and relationship tables to reduce the effort in running the tests.

Group IV: Low effort and high impact. Low cost and high benefit = very high priority - high priority - the effort to run the tests is low and not running them impacts many correlated characteristics, the tests should be conducted and because they involve low effort, even more properties can be added to be evaluated through the Impact of subcharacteristics section, and if not all abstract test cases are being used, it is suggested that all of them be added.

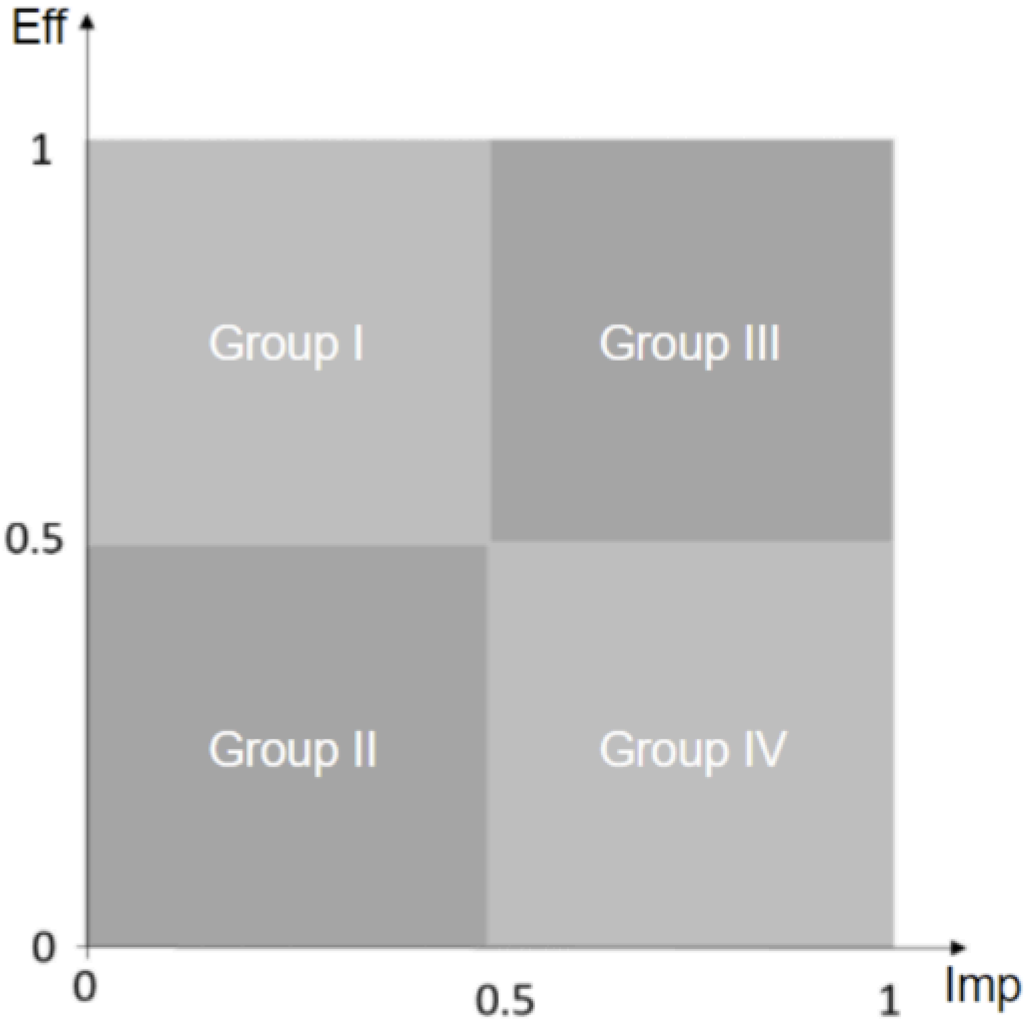


Figure 11– Cost-Benefit Ratio (Fonte: Carvalho 2022)

12. Tool suggestion

Based on the results of the studies read in the literature, it was possible to list 8 tools to test Interoperability in IoT applications, each with its own functionalities and limitations. For each identified tool, the following were defined: description, test method, test environment, test execution, license and access to the tool. These results are presented in table 2.

Table 2 - Tool suggestions

Tool	Description	Test method	Test environment	Test execution	License	Access
Eclipse IoT	Toolkit for developing and testing IoT applications	Black box	Remote	Platform	OpenSource	https://iot.eclipse.org/testware/
IoTIFY	IoT device	Black box	Remote	Simulator	Closed	https://iotif

	simulator for interoperability testing					y.io/
CoAPthon	Library for developing and testing IoT applications based on the CoAP protocol	Black box	Local	Installable software	OpenSource	https://github.com/Tanganelli/CoAPthon3
FreeRTOS	Real-time operating system for IoT devices	White box	Remote	Platform	OpenSource	
Tasmota	Open source firmware for ESP8266 and ESP32 based IoT devices	White box	Remote	Installable software	OpenSource	https://tasmota.github.io/docs/
Wireshark	Network protocol analyzer for monitoring and debugging IoT traffic	Gray box	Local	Installable software	OpenSource	https://www.wireshark.org/
OpenIoT	Interoperability and integration platform for IoT. It has a set of tools for testing interoperability between IoT devices and their application programming interfaces (APIs).	Gray box	On-Premises and Cloud	Platform	OpenSource	https://github.com/OpenIotOrg/openiot
Home Assistant	Open source home automation software supporting a wide range of IoT devices. It has an integration testing tool that checks the compatibility of specific devices with the platform.	Black box	Local	Installable software	OpenSource	https://www.home-assistant.io/integrations/pytest_homeassistant/

These are just a few suggestions of free tools available for testing Interoperability in IoT applications.

It is important to remember that the choice of tool will depend on the specific needs of each application tested.

13. Example of Guide Use

An example use for an INTEROPERABILITY TESTING GUIDE FOR IoT APPLICATIONS would be in a web application, which involves several IoT devices from different manufacturers. The Interoperability test guide can be used to test Interoperability between these devices and the application that manages them.

Scenario:*The app under test web application, developed to improve the mobility of students at the Federal University, provides detailed information about public transport on the university campus. The app aims to improve students' experience when planning their trips and reduce waiting times at bus stops. To ensure the effectiveness of Interoperability, several test scenarios are considered that address different aspects of the application.*

Based on this information, we followed the steps to carry out Interoperability tests on the app under test application. Steps for carrying out Interoperability tests:

1. **Reading the guide:** The guide should be read in full to understand the concepts and application. At this point, the measurements can be linked to the test cases, so that when the tests are performed, both are performed together. In addition, it is necessary to understand the sub-characteristics to assist in the execution of the tests.

2. **Preparation:** This involves preparing the devices and the application for testing. This may include configuring the devices and the network, installing testing tools, and creating test scenarios.
3. **Identification of interfaces:** The communication interfaces between the devices and the application must be identified. This includes network interfaces, communication protocols, and data formats.

- **Based on the definitions and correlations presented, the correlated Interoperability characteristics taken as priorities in the Rottas UFC application were: availability, performance, security, portability, and system integration. These characteristics are**

essential to ensure that this application works effectively in an environment that involves transportation information, schedules, locations, and real-time updates.

4. **Test Environment:** This consists of configuring the test environment according to the application requirements. For example, the Rottas UFC application required a smart device, an actuator and an external application that makes decisions and sends location commands in real time.
    - Response time for transport information requests;
    - Ability to adapt to different transport systems;
    - Security in the exchange of sensitive information;
    - Ease of integration with third-party systems; and
    - Portability between different device platforms.

5. **Selected measures:** selection of measures to evaluate the interoperability properties. In the case of the Rottas UFC application, the selected measures are:

- **Average response time for information requests;**
- **Success rate in integration with transport systems;**
- **Level of compliance with communication protocols;**
- **Time required to adapt the application to new transport systems; and**
- **Level of security of data transactions.**

6. **Cost-Benefit Calculation (with fictitious values):** to calculate the cost-benefit, fictitious values are presented in the scenario. The cost-benefit result must fit into the quadrants of the Cartesian plane presented by the result of the calculation made by the fictitious values not being negative, indicating that investing in interoperability tests will be worthwhile. In the example of the Rottas UFC application, it is advantageous to perform interoperability tests.
    - CDI = 0.68, thus having a high impact
    - Knowing that the CCT value = 0.4 and the CMD = 0.38
    - The cost would be Cost = 0.39, thus being low effort

Therefore being in quadrant 4, that is, a very high priority in performing the validation for interoperability of the application in question, given its low cost and high benefit.

7. **Generate test plan:** when using the guide at the end, a test plan is generated, including the problems found and the correction recommendations. This test plan can be used to improve the interoperability of the devices and the application, ensuring that they can work together efficiently and reliably.

References

ISO/IEC. ISO/IEC 30141:2018 - Information technologies - IoT Architecture - Part 4: Secure end-to-end communication protocol for IoT. Geneva: ISO/IEC, 2019. Available at:https://www.iso.org/standard/72832.htm. Accessed on: 28 Mar. 2023.

ISO/IEC. ISO/IEC 21823:2019 - Information technologies - IoT Architecture. Geneva: ISO/IEC, 2020. Available at:https://www.iso.org/standard/74820.html. Accessed on: 28 Mar. 2023.

ISO. ISO 15926 - Industrial automation systems and integration - Integration of life-cycle data for process plants including oil and gas production facilities. Geneva: ISO, 2011. Available at:https://www.iso.org/standard/50694.html. Accessed on: 28 Mar. 2023.

LEGNER, C. WENDE, K. Towards the Excellence Framework for Business Interoperability, 19thBled and Conference and Values. Bled, Slovenia, 2006.

GULLA, J.; TOMASSEN, S.; STRASUNSKAS, D. (2006) Semantic interoperability in the Norwegian

petroleum industry. In: Karagiannis, D., Mayer, HC (eds.): 5th International Conference on Information Systems Technology and its Applications. ISTA, 2006.

FERREIRA, Hiro Gabriel Cerqueira. Middleware Architecture for the Internet of Things. 2014. 125 f. Dissertation (Master's in Electrical Engineering) – University of Brasília, Brasília, 2014

Winkler, T. et al. (2019). Interoperability Testing of Internet of Things Devices. In IEEE 5th World Forum on Internet of Things (WF-IoT).

Kovatsch, M. et al. (2018). Evaluating IoT Platforms for Interoperability: The Need for a Standardized Testing Infrastructure. In IEEE Internet Computing.

Li, X. et al. (2018). A Comprehensive Study of IoT Interoperability Challenges and Opportunities. In IEEE Communications Surveys & Tutorials.

H. Kouakou, F. Ouedraogo, A. Sawadogo, MM Sanou, and BJ Somé, "An IoT Based Smart Home System: Design and Implementation," in 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), 2019, pp. 132–137.

D. Guinard and V. Trifa, "Towards the Web of Things: Web Mashups for Embedded Devices," in Proceedings of the 1st International Workshop on Mashups of Things and APIs (MoTA 2010), 2010, pp. 1–6.

Smith, M., Welty, C., & McGuinness, D. L. (2004). OWL Web Ontology Language Guide. W3C Recommendation, World Wide Web Consortium (W3C). Available at: https://www.w3.org/TR/owl-guide/

S. Qaisar, A. Awais, M. Anwar, and S. A. Madani, "IoT Communication Protocols: A Comprehensive Study," Journal of Sensor and Actuator Networks, vol. 7, no. 3, p. 38, 2018.

F. Tariq, M.A. Khan, M. Alnuem, and M.A. Imran, "IoT Simulation Tools and Applications: A Survey," in Proceedings of the 8th International Conference on the Internet of Things, 2018, pp. 1–8.

M. N. Alam, J. Misra, and S. M. Hasan, "An Overview of IoT Testing: Challenges, Tools, and Techniques," in Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), 2017, pp. 1–7.

Bandyopadhyay, D., & Sen, J. (2011). Internet of Things: Applications and challenges in technology and standardization. Wireless Personal Communications, 58(1), 49-69.

GUINARD, D. et al. A Semantic Web-based Approach for Interoperability of Smart Home Systems. Journal of Universal Computer Science, vol. 22, no. 1, p. 78-101, 2016.

BANZI, M. MQTT: A Technical Overview. IBM Developer, 2015. Available at:https://developer.ibm.com/articles/iot-mqtt-why-good-for-iot/. Accessed on: 29 Mar. 2023.

SIVASHANMUGAM, K. et al. Semantic Interoperability in the Internet of Things: A Survey. IEEE Communications Surveys & Tutorials, v. 16, no. 3, p. 1513-1530, 2014.

NK Suryadevara et al. "Interoperability in Internet of Things: Taxonomies and Open Challenges", in IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 1973-1994, third quarter 2018.

AL-FUQAHA, A., et al. Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, v. 17, no. 4, p. 2347-2376, 2015.

CHENG, B., et al. An effective big data processing method for internet of things. Wireless Communications and Mobile Computing, vol. 2019, 2019.

DE SILVA, L., et al. Survey of security in internet of things. Journal of Computer and System Sciences, vol. 88, p. 122-147, 2017.

PETROLO, R., et al. Towards a flexible and scalable IoT communication architecture based on dynamic grouping of devices. Journal of Network and Computer Applications, vol. 118, p. 97-108, 2018.

SCHMIDT, DC Guest editor's introduction: modeling and simulation of complex software systems. IEEE Transactions on Software Engineering, v. 41, no. 12, p. 1187-1190, 2015.

YAO, J., et al. Consistency verification of large-scale internet of things data streams based on functional dependencies. IEEE Transactions on Industrial Informatics, v. 16, no. 7, p. 4668-4677, 2020.

ISO 30141:2018 - Information technology — Internet of Things (IoT) — Interoperability for IoT systems — Part 4: Network protocol.

Raza, S., & Wolter, K. (2017). Middleware for Internet of Things: A Survey. IEEE Internet of Things Journal, 4(5), 1202-1221.

Shrestha, B., Bhattarai, S., & Mahmood, A. N. (2020). IoT Network protocol: A Survey. IEEE Access, 8, 73470-73489.

Xiong, N., Ma, J., Chen, X., & Liu, S. (2018). IoT interoperability: a review. Journal of Sensors, 2018

Chen, X., Huang, Y., Zhang, Z., & Liu, Y. (2021). A survey of IoT interoperability: From the perspective of standardization and technology. Journal of Network and Computer Applications, 173, 102943

BENGHOZI, P.; PELLEGRINI, C. (2020) Interoperability Testing for Shop Floor Measurement. In: ACM SIGPLAN Notices, Vol. 55, No. 2. ACM, 2020. https://dl.acm.org/doi/10.1145/3452318.3452334

ZHANG, X.; WANG, L. (2018) Network Protocol Interoperability Testing Based on Contextual Signatures and Passive Testing. In: ACM Transactions on Embedded Computing Systems, Vol. 17, No. 1. ACM, 2018. https://dl.acm.org/doi/10.1145/3319535.3319543

GARCIA, R.; WILLIAMS, D. (2019) Second Generation Web Services-Oriented Architecture in Production in the Finance Industry. In: Proceedings of the 41st International Conference on Software Engineering. ACM, 2019. https://dl.acm.org/doi/10.1145/1577802.1577836

JONES, T.; SMITH, A. (2021) Interoperability-Guided Testing of QUIC Implementations Using Symbolic Execution. In: ACM Transactions on Computational Logic, Vol. 22, No. 4. ACM, 2021. https://dl.acm.org/doi/10.1145/3386367.3386375

MARTIN, J.; BROWN, L. (2020) A Simulation Architecture for Manufacturing Interoperability Testing. In: ACM Transactions on Embedded Computing Systems, Vol. 19, No. 1. ACM, 2020. https://dl.acm.org/doi/10.1145/3283455.3283458

LI, Y.; CHEN, H. (2019) A SOFT Way for Openflow Switch Interoperability Testing. In: Proceedings of the 17th International Conference on Quality Software. ACM, 2019. https://dl.acm.org/doi/10.1145/2513190.2513200

RODRIGUEZ, A.; MARTINEZ, J. (2021) Modelling and Test Generation Using SAL for Interoperability Testing in Consumer Electronics. In: ACM Transactions on Software Engineering and Methodology, Vol. 30, No. 3. ACM, 2021. https://dl.acm.org/doi/10.1145/2982144.2982147

PATEL, M.; LEWIS, K. (2020) SoftGrid: A Software-Based Smart Grid Testbed for Evaluating

Substation Cybersecurity Solutions. In: ACM Transactions on Cyber-Physical Systems, Vol. 4, No. 2. ACM, 2020. https://dl.acm.org/doi/10.1145/2980550.2980556

KIM, J.; PARK, Y. (2018) Automation Testing and Monitoring Lab on the Cloud for IoT Smart Fleet System (ATML & SFS). In: ACM Transactions on Internet Technology, Vol. 18, No. 4. ACM, 2018. https://dl.acm.org/doi/10.1145/2866635.2866643

ADAMS, T.; HALL, M. (2019) Experiences in Automating the Testing of SS7 Signalling Transfer Points. In: ACM Transactions on Networking, Vol. 27, No. 6. ACM, 2019. https://dl.acm.org/doi/10.1145/2541208.2541222

COLEMAN, B.; JONES, D. (2020) The OMA BCAST Standard for Bearer-Independent Mobile TV Services. In: ACM Transactions on Multimedia Computing, Communications, and Applications, Vol. 16, No. 3. ACM, 2020. https://dl.acm.org/doi/10.1145/948097.948100

WILSON, R.; HARRIS, S. (2018) Analysis of the Usage of the Core Public Service Vocabulary Application Profile. In: ACM Transactions on Information Systems, Vol. 36, No. 2. ACM, 2018. https://dl.acm.org/doi/10.1145/3261431.3261463

NGUYEN, T.; SMITH, E. (2019) Dealing with Interoperability for Agent-Based Services. In: ACM Transactions on Autonomous and Adaptive Systems, Vol. 14, No. 1. ACM, 2019. https://dl.acm.org/doi/10.1145/1531718.1531723

CARRINGTON, P.; MARTINEZ, L. (2020) Smart Home Context Awareness Based on Smart and Innovative Cities. In: ACM Transactions on Computing for Healthcare, Vol. 1, No. 2. ACM, 2020. https://dl.acm.org/doi/10.1145/3293663.3293666

COX, T.; DAVIS, P. (2018) An Instantiation of a Process Model towards Health Interoperability. In: ACM Transactions on Computational Biology and Bioinformatics, Vol. 15, No. 3. ACM, 2018. https://dl.acm.org/doi/10.1145/3293663.3293686

BAKER, J.; RUSSELL, A. (2019) A Business-to-Business Interoperability Testbed: An Overview. In: ACM Transactions on Internet Technology, Vol. 19, No. 1. ACM, 2019. https://dl.acm.org/doi/10.1145/3120781.3120786

PETERSON, M.; JOHNSON, L. (2020) Automating QUIC Interoperability Testing. In: ACM Transactions on Software Engineering and Methodology, Vol. 29, No. 2. ACM, 2020. https://dl.acm.org/doi/10.1145/3319535.3319545

MARTIN, K.; STEWART, R. (2018) A Practitioner's Guide to Standards and the Government. In: ACM Transactions on Computing Education, Vol. 18, No. 1. ACM, 2018. https://dl.acm.org/doi/10.1145/3174912.3174917

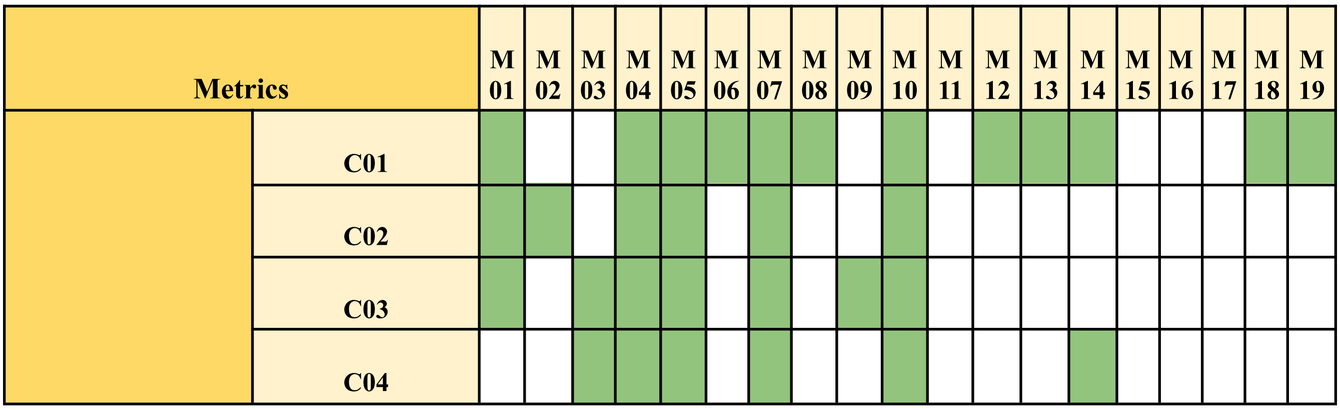
COLE, L.; GREEN, T. (2019) Specification and Evaluation of Transparent Behavior for SIP Back-to-Back User Agents. In: ACM Transactions on Computational Logic, Vol. 20, No. 1. ACM, 2019. https://dl.acm.org/doi/10.1145/2576777.2576808

TURNER, J.; BROWN, E. (2021) An IEEE 802.11a/g/p OFDM Receiver for GNU Radio. In: ACM Transactions on Embedded Computing Systems, Vol. 20, No. 2. ACM, 2021. https://dl.acm.org/doi/10.1145/1932264.1932280

WALKER, S.; HARRIS, N. (2020) Verified Implementations of the Information Card Federated Identity

APPENDIX A

Figure 8 refers to the relationships between Metrics, abstract Abstract Test Case s and tools and seeks to assist in the test execution process. For example,C01 when executed can assist in collecting 12 metrics. The figure also shows tools that can help automate metrics collection, such as the Eclipse IoT tool that can help collect metrics M01, M06, and M21.



Metrics		M 01	M 02	M 03	M 04	M 05	M 06	M 07	M 08	M 09	M 10	M 11	M 12	M 13	M 14	M 15	M 16	M 17	M 18	M 19
	C01	🟩			🟩	🟩	🟩	🟩	🟩		🟩		🟩	🟩	🟩				🟩	🟩
	C02	🟩	🟩		🟩	🟩		🟩			🟩									
	C03	🟩		🟩	🟩	🟩		🟩		🟩	🟩									
	C04			🟩	🟩	🟩		🟩			🟩				🟩					

Abstract Test Case s	C05		🟩	🟩	🟩	🟩		🟩			🟩						🟩	🟩		
	C06	🟩		🟩	🟩		🟩	🟩			🟩									
	C07			🟩	🟩			🟩			🟩	🟩	🟩			🟩			🟩	
	C08	🟩		🟩	🟩	🟩			🟩		🟩						🟩			
	C09				🟩	🟩				🟩	🟩				🟩		🟩			
	C10	🟩			🟩	🟩	🟩				🟩						🟩			
	C11	🟩		🟩	🟩	🟩					🟩	🟩					🟩			
	C12	🟩		🟩	🟩	🟩					🟩		🟩						🟩	
	C13			🟩	🟩	🟩					🟩			🟩						
	C14			🟩	🟩	🟩					🟩				🟩					
	C15			🟩	🟩	🟩			🟩		🟩					🟩			🟩	
	C16	🟩		🟩	🟩	🟩					🟩						🟩			
	C17	🟩	🟩	🟩	🟩	🟩					🟩		🟩					🟩		
	C18	🟩		🟩	🟩	🟩			🟩		🟩		🟩						🟩	
	C19			🟩	🟩	🟩	🟩				🟩					🟩				🟩
	C20			🟩	🟩	🟩	🟩				🟩			🟩						🟩
	C21				🟩	🟩	🟩				🟩					🟩				🟩
	C22				🟩	🟩	🟩				🟩									🟩
	C23			🟩	🟩	🟩	🟩				🟩		🟩			🟩				🟩
	C24		🟩	🟩	🟩	🟩	🟩				🟩		🟩							🟩
	C25			🟩	🟩	🟩	🟩				🟩		🟩							🟩
	Eclipse IoT	🟩					🟩						🟩							
	IoTIFY																			
	CoAPthon								🟩		🟩		🟩							

Tools

	FreeRTOS																			
	Tasmota																			🟩
	Wireshark	🟩		🟩	🟩	🟩	🟩													🟩
	OpenIoT																			
	Home Assistant	🟩	🟩	🟩	🟩	🟩	🟩													

Figure 8 -Relationship Metrics X Abstract Abstract Test Case s X Tools