SECURE AND AGILE CONNECTED THNGS

**SCRATCh**

****

Copyright © 2018-2022, SCRATCh

Deliverable <XX>



Deliverable <XX>: User Requirements

**Work package:** **WP2**

**Affected milestone:** **M?**

**Partners involved:** **ANYWI**

**Date:** **12/02/2020**

**Deliverable version:**  **v 0.1**

**Editor:**  **Franklin Selgert, AnyWi**

**Author(s):** **Fanklin Selgert, AnyWi**

<Author2>, <Organization2>

<…>

<AuthorN>, <OrganizationN>

**Responsible Contact:** <Name>

<Organization>

<Address>

<Mail>

<Business phone>

Approval

|  |  |  |  |
| --- | --- | --- | --- |
| Role Approver | Name Approver | Approval date | Signature |
| Process Owner |  |  |  |
| Quality Assurance |  |  |  |

Inhoud

[Introduction 3](#_Toc32572869)

[Purpose 3](#_Toc32572870)

[Scope 3](#_Toc32572871)

[Stakeholders 3](#_Toc32572872)

[Overview 3](#_Toc32572873)

[Product description 4](#_Toc32572874)

[Use Case and Workflow 4](#_Toc32572875)

[Device Users, Use Environments and User Interface 4](#_Toc32572876)

[Device User Description 5](#_Toc32572877)

[Device Use Environments 5](#_Toc32572878)

[Device User Interface 5](#_Toc32572879)

[User Requirements 5](#_Toc32572880)

[Intended Use 5](#_Toc32572881)

[Primary Operating Function Requirements 5](#_Toc32572882)

[Functional Requirements From End-User Perspective 6](#_Toc32572883)

[Security & Privacy 8](#_Toc32572884)

[Quality 9](#_Toc32572885)

[Product Claims? 9](#_Toc32572886)

[Applicable Documents 10](#_Toc32572887)

[Referenced Documents 10](#_Toc32572888)

Introduction

Purpose

This document describes the User requirements for the SCRATCh Generic Demonstrator (SGD)

Scope

This document focuses on business requirements of the SCRATCh Generic Demonstrator system realized by the integration of the sub-systems Gateway and End node

Stakeholders

The main (typical) stakeholders and their interests for this document are:

* Product Management, input in this document.
* Software Developer,
* Service and Operations, input for service and operations.
* System Architect, input for product requirement, design, safety and security
* Quality Assurance, check compliance with QMS and applicable quality system requirements
* Requirements Manager, for consistency of system requirements
* Validation Manager, input for verifications activities

Overview

The sections of this document contain informative and/or normative texts.

* All *normative requirement texts* are tagged with a requirement identifier consisting of a tag ("BR") and an ID number like in:

**<Headline>**

|  |
| --- |
| **<[Applicable Release]> -UR.SGD.<ID number>** |

<Normative text describing the requirement.>

 The requirement **<Headline>** is formulated in a tagged and structured manner, such that tracing can easily recognize the origin of the requirement:

**<ReqType>.<(Sub)System>.<High level Feature/Requirement>**

 where:

* + **<ReqType>** indicates the type of requirement (e.g. Business Requirement (BR), System Requirement (SR), etc)
  + **<(Sub)-System>** indicates the (sub)-system (e.g. ????)
  + **<High level Feature/Requirement>** provides a headline text indicating the feature / requirement

*Examples:*

      E.g.: *UR.SGD.intended-use*

The unique identifier of each element consists of the following elements:

* **<[Applicable Release]>** indicates the specific release that the requirement is applicable for.
* **ID number>** a numeric identifier, unique within the content of this requirements document.

Product description

SCRATCh Generic demonstrator (SDG) is a development and test environment where tooling developed in SCRATCh can be tested, demonstrated and evaluated. The generic demonstrator is using a commonly available development environment e.g. GITHUB and Docker for distribution of released software. Partners can extend functionality outside of the generic demonstrator in order to obtain a better fit for demonstrating a specific use case. The SDG has two goals:

1. Test and improve the use of (SCRATCh) tools for the development of secure IoT systems.
2. Test an IoT product/ system and simulate its lifecycle by testing secure update features and monitoring, in other words keeping the system safe during it lifecycle.

Use Case and Workflow

The main use case relates to the SDG, is described in the table below.

|  |  |  |
| --- | --- | --- |
| **#** | **Step description** | **In/Out Scope** |
| 1 | A software developer creates a software update in (GitHub) | In scope |
| 2 | The software update is built and checked with available SCRATCh tooling | In scope |
| 3 | The built is packaged for distribution, of grid in house test on staging may be done. | In scope |
| 4 | The package is released for deployment in Github | In scope |
| 5 | The package is deployed using a push or pull mechanism | In scope |
| 6 | Alternate step, the package is received by a gateway and set for deployment on a end node as defined in the released package. | In scope |
| 7 | After successful deployment a message is send to the xxx | In scope |
| 8 | After a failed deployment a message is send to the xxx | In scope |
| 9 | Monitor the system and simulate security issues, to enhance system response to security breaches. | ? |

Device Users, Use Environments and User Interface

Device User Description

The generic demonstrator is used by experienced developers and Generic IT specialists, belonging to the companies defined in the SCRATCh project.

As it a development testing tools there is no specific support, only support by the SCRATCh community exist.

Device Use Environments

The SDG is used in a testbed or lab environment.

Device User Interface

No limitations for development

User Requirements

This chapter lists User requirements including acceptance criteria relating to:

* Security
* performance
* Functional requirements from end user perspective,
* Primary Operating Functions. A Primary Operating Function is a function that involves user interaction that is related to the security of the IoT environment

Intended Use

UR.SGD.IntendedUse

|  |
| --- |
| 1.0-UR.SGD..SGD.10x |

The SDG shall allow to initiate and test secure configuration and firmware updates of Gateway devices and IoT end nodes. The IoT end nodes are categorized as

1. Non-OS. No capability for encryption.
   * AVR 2k RAM etc.
2. Non-OS. Capability for encryption.
   * ESP32
   * ARM cortex M33
3. OS
   * ARM cortex A
   * Intel

Primary Operating Function Requirements

This chapter lists product requirements including acceptance criteria relating to:

* Safety/security
* performance

Functional requirements from end user perspective, and indicating Primary Operating Functions. A Primary Operating Function is a function that involves user interaction that is related to the security safety of the IoT environment. These requirements wil get special attention and tracking.

UR.SGD..primary

|  |
| --- |
| 1.0-UR.SGD.20x |

As a **stakeholder**, I want to have the option to intervene with the system with the intention of limiting damage in case of a serious hack event. (out of scope for generic demonstrator)

Functional Requirements From End-User Perspective

This chapter lists other functional and non-functional requirements from the perspective of various types of users of the product.

For each requirement it is indicated whether it is **new**, or if it is a **legacy** requirement, also available in older versions of the product.

UR.SGD.Development.build

|  |
| --- |
| 1.0-UR.SGD.101 |

As **software developer**, I want to be able to use different build environments, e.g. OpenWRT, Yocto Project (OpenEmbedded) (openembedded), NanoBSD (FreeBSD)

UR.SGD.Development.versioning

|  |
| --- |
| 1.0-UR.SGD.102 |

As **software developer** I want to have multiple tool for versioning source code and documents tailored towards the type of documents, e.g. subversion, git, cvs, mercurial for source code and Nextcloud, Sharepoint for binary (documents)UR.SGD.Development.test

|  |
| --- |
| 1.0-UR.SGD.103 |

As **software developer**, I want to be able to quickly test code on targeted devices.

UR.SGD.Development.config

|  |
| --- |
| 1.0-UR.SGD.104 |

As **software developer,** I want a modular layered configuration system, allowing me to create 'base' image configuration and 'application' specific configuration.

UR.SGD..Development.config

|  |
| --- |
| 1.0-UR.SGD.105 |

As **software developer**, I want that development changes in an image are trackable, allowing them to be feed back into the modular configuration system.

UR.SGD.Development.opensource

|  |
| --- |
| 1.0-UR.SGD.106 |

As **software developer**, I want to use Free and open-source software (FOSS), with a specific license requirements (e.g. only BSD licensed tools).

UR.SGD.Development.bugs,pub

|  |
| --- |
| 1.0-UR.SGD.107 |

As **software developer**, I want to be able to publish bugs found in the Free and open-source software (FOSS) back to the community by using having bridges between my bug-tracking and the tooling used by the maintainer of the Free and open-source software (FOSS).

UR.SGD.Development.requirements

|  |
| --- |
| 1.0-UR.SGD.108 |

As **software developer**, As software developer, I want easy access to high level requirements ,and visa versa, to convert them to technical specifications and unit test.

UR.SGD.Development.documentation

|  |
| --- |
| 1.0-UR.SGD.109 |

As **software developer**, I want to document technical specifications and (unit) test in the software code.

UR.SGD.product.certification

|  |
| --- |
| 1.0-UR.SGD.110 |

As **Product Manager**, I want a mechanism to set certification criteria of the development at hand at very start of development, clear for all developers.

UR.SGD.Development.security.notification

|  |
| --- |
| 1.0-UR.SGD.111 |

As **software developer**, I want to receive automatic notification of know bugs/security issues found in used libraries.

UR.SGD..Development.security.reporting

|  |
| --- |
| 1.0-UR.SGD.112 |

As **software developer**, I want to be able to report security issues to the required public security exposure channels.

UR.SGD..Development.security.reporting

|  |
| --- |
| 1.0-UR.SGD.113 |

As **software developer**, I want to be able to report security issues to the security officer of the upstream software maintainer, using the appropriate means

UR.SGD..Development.security.reporting

|  |
| --- |
| 1.0-UR.SGD.114 |

As **software developer** I want to be able to report security issues, without private leaking information of my systems.

UR.SGD..Development.image.update

|  |
| --- |
| 1.0-UR.SGD.115 |

As **software developer**, I want to be able to update certain components within an image without having to create a completely new image.

UR.SGD..Development.licensing

|  |
| --- |
| 1.0-UR.SGD.116 |

As **software developer**, I want to have a tool for license management for used software sources/libraries.

UR.SGD..Development.documentation

|  |
| --- |
| 1.0-UR.SGD.117 |

As **software developer**, I want to have a tool for abstracting and listing all used tools, libraries etc.

UR.SGD.remote.support

|  |
| --- |
| 1.0-UR.SGD.118 |

As **a Product manager**, I want remote support for all devices installed at customer premises.

UR.SGD.maintenance.tracking

|  |
| --- |
| 1.0-UR.SGD.119 |

As **Service manager**, I want an easy system to track licensing, versions in relation to installed assets.

UR.SGD.maintenance.patch

|  |
| --- |
| 1.0-UR.SGD.120 |

As **Service manager**, I want to be able to patch compromised devices.

UR.SGD.maintenance.control

|  |
| --- |
| 1.0-UR.SGD.121 |

As **End User**, I want to have control over updates executed in mine installed base

UR.SGD.maintenance.policy

|  |
| --- |
| 1.0-UR.SGD.122 |

As **End User**, I want to be able to define a policy that regulates the actions that are allowed remotely on the installed base. Witch can include an energy shutdown of equipment to contain damage.

UR.SGD.maintenance.security.automation

|  |
| --- |
| 1.0-UR.SGD.123 |

As **End User**, I want to have an automated security control mechanism that detects hacking attempts and act in them within a set policy.

Security & Privacy

UR.SGD.Security.Privacy

|  |
| --- |
| 1.0-UR.SGD.200 |

As **Product** **Manager**, I want to be compliant with security and privacy standards.

UR.SGD.Security.Privacy

|  |
| --- |
| 1.0-UR.SGD.201 |

As **Product** **Manager**, I want all communication with the cloud to be encrypted (OWASP IOT nr 7)

UR.SGD..Security.Privacy

|  |
| --- |
| 1.0-UR.SGD.202 |

As **Product** **Manager** I want no privacy sensitive data communicated to the cloud, when no such agreement exists to store this data-remotely and/or the cloud is not compliant with applicable regulations e.g. GDPR (OWASP IOT nr 6)

UR.SGD..Security.Privacy

|  |
| --- |
| 1.0-UR.SGD.203 |

As **Product** **Manager**, I want a secure authentication and authorization mechanism between cloud and Gateway. (OWASP IOT nr 3)

UR.SGD.Security.Privacy

|  |
| --- |
| 1.0-UR.SGD.204 |

As **Product** **Manager**, I want an easy and secure mechanism to authenticate and authorize end nodes to the gateway. (OWASP IOT nr 1)

UR.SGD..Security.Privacy

|  |
| --- |
| 1.0-UR.SGD.205 |

As **Product** **Manager**, I want maintenance updates to use a secure mechanism to authenticate, authorize and encrypt the updates. (OWASP IOT nr 4)

Quality

UR.SGD.Regulatory.Standards

|  |
| --- |
| 1.0-UR.SGD.301 |

As **Quality Manager,** I want the product to be compliant with existing regulatory standards and directives for the applicable country.

Product Claims?

The SGD shows a part of the holistic approach for IoT security as set out in the SCRATCh FPP.

# Applicable Documents

## Referenced Documents

|  |
| --- |
| documents |

|  |  |  |
| --- | --- | --- |
| *Reference* | *Reference Id* | *Document title* |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |