SONY

White paper

Root of trust
The security model of Sony's mSafety

Published: March 2021
Document no: v1 r1

mSafety

This white paper is published by:
Sony Network Communications Europe B.V

Mobilvägen 4
223 62 Lund
Sweden

sonynetworkcom.com

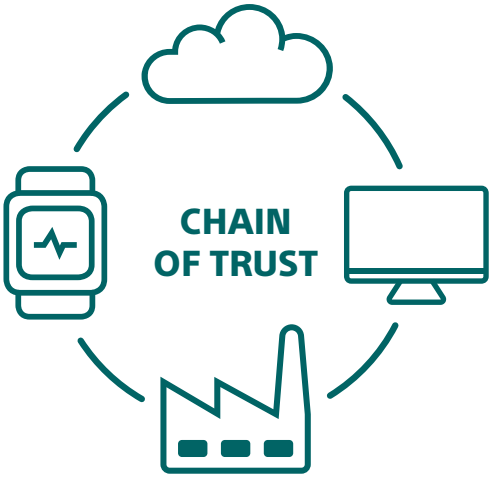


Table of contents

Root of trust - The security model of Sony's mSafety

Executive summary

The mSafety security model provides a chain of trust that guarantees the security of the mSafety system. This model ensures that the customer can implement and maintain a secure product that consists of a tamper proof application, running on mSafety wearables that are bound to the customer's back-end. The chain of trust is a combination of:

- The mSafety wearable secure boot
- The production of mSafety wearables
- The mSafety cloud back-end
- The customer's back-end
- The safekeeping of private customer keys by the customer

In addition, mSafety provides support for customers to perform end-to-end encryption of application data from the wearable to the customer's back-end.

Introduction and scope

The technological capabilities for health monitoring have taken huge leaps in performance and reliability as improvements in connectivity and sensor technology have been made more readily available. The new technologies support the development of wearables for business applications, enabling better solutions for remote health and safety monitoring.

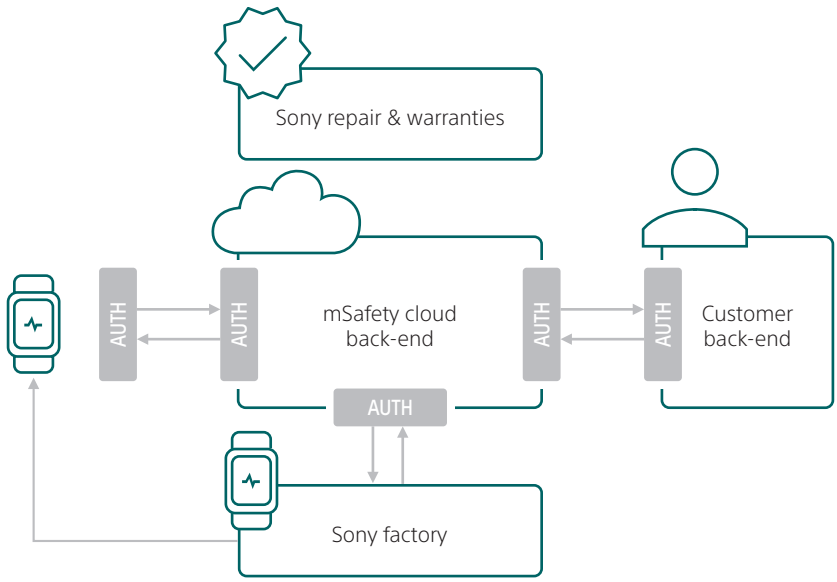
However, use cases in industries like healthcare, medicine and security put complex requirements on the confidentiality, integrity and availability (CIA) of wearable platforms. These requirements have so far not been fully met by existing consumer wearables. The answer to this problem is a complete wearable solution that handles customer data with high levels of care and integrity and provides a reliable connection between the wearable and back-end.

The scope of this white paper is to describe how Sony's mSafety solution is designed to provide the security needed for wearable applications that manage chronic conditions remotely, enable active assisted living for elderly and track or aide people engaged in potentially hazardous work or leisure activities.

Terms, definitions and abbreviations

mSafety wearable	The wearable part of the mSafety system. Wearables come in two varieties, development and production.
mSafety OS	The complete set of mSafety binaries running on the wearables. It also includes all APIs accessible to the wearable application.
Wearable application	An application, usually developed by an mSafety customer, running on the mSafety wearable.
mSafety wearable SDK	The development environment for mSafety wearables. It includes tools, a reference application and documentation. It also includes the mSafety OS so that a complete update package consisting of the OS and the application can be built using the tools in the SDK.
mSafety cloud back-end	The mSafety cloud back-end handles communication between the wearable and customer back-end. It also contains the mSafety Portal.
mSafety Portal	The part of the mSafety cloud back-end that is used to manage users and wearables and to access downloads and documentation.
Customer back-end	The customer's back-end application.
Platform core	Software that sets up basic functions on the wearable and loads various modules.
Keystore	A binary object in the mSafety OS containing customer public keys signed by mSafety.
TLS	Transport Layer Security. A protocol used for secure network communication.

mSafety security overview



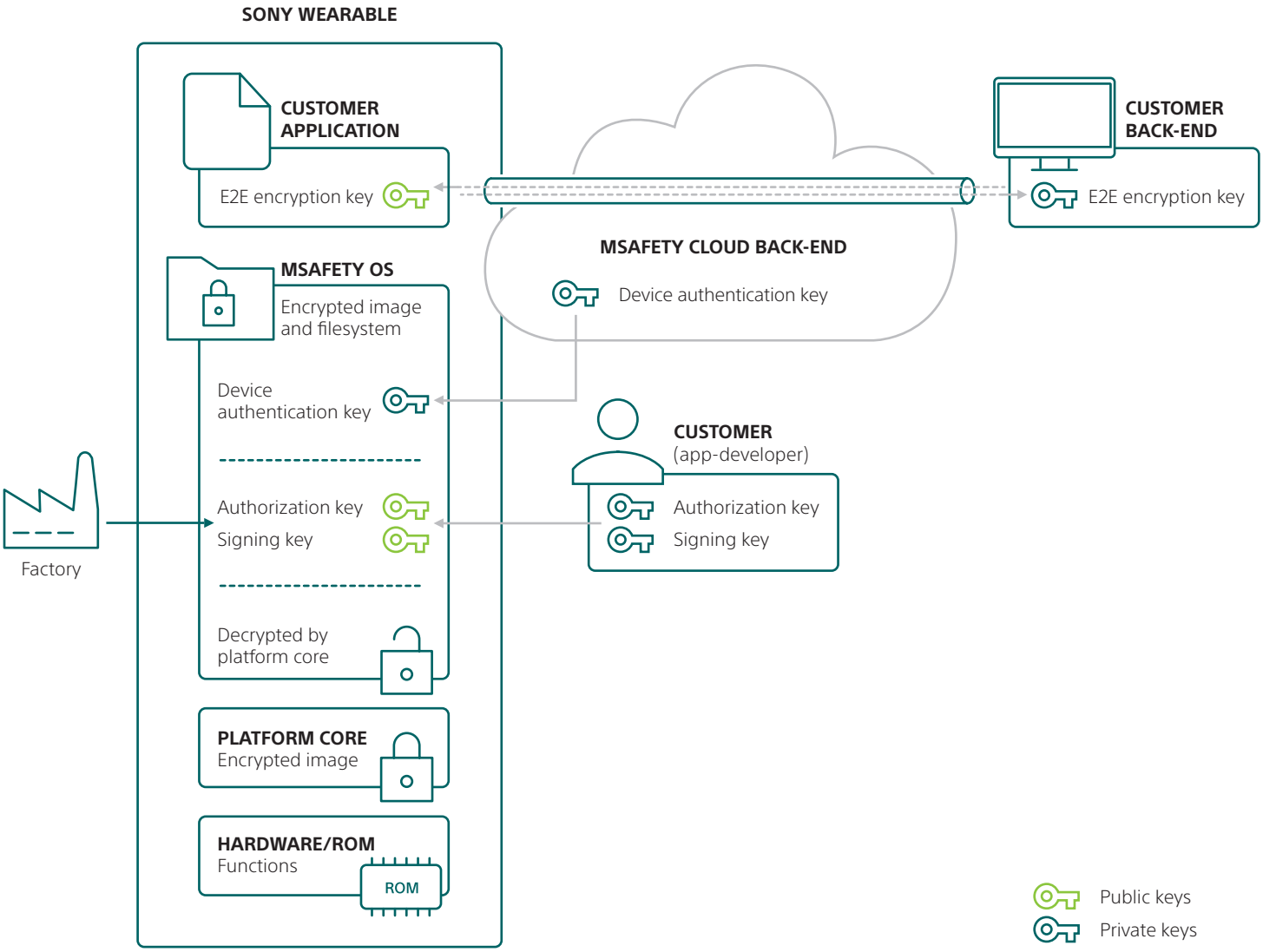
The mSafety security stack is a combination of:

- The mSafety wearable secure boot
- The production of mSafety wearables
- The mSafety cloud back-end
- The customer's back-end
- The safekeeping of private customer keys by the customer

All communication between a customer's wearables and back-end uses Transport Layer Security (TLS 1.2) across the different network segments.

Apart from TLS, in a production environment, customers are also responsible for implementing end-to-end encryption between their wearables and their backend so that end-user data is only visible to the customer. While mSafety provides a way to do this, customers are free to choose other implementations.

mSafety wearable security



Wearable manufacturing

As the mSafety wearable manufacturer, Sony controls the device manufacturing process and provides warranties for produced wearables. In the factory, the wearables are provisioned with certificates and customer specific keys that bind them to the customer's back-end.

During the manufacturing process, there are a few steps that are of particular interest from a security perspective as they set up the Secure Boot Chain:

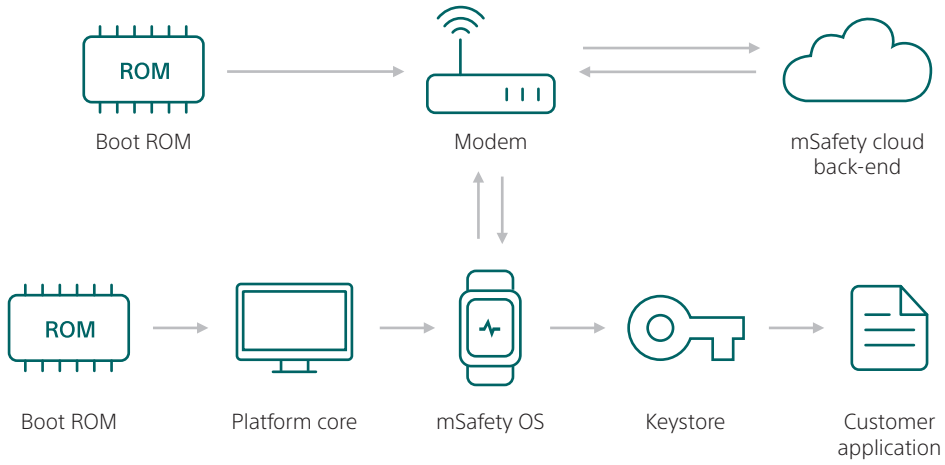
- An image decryption key is written into read-only memory, thus making sure that only the platform core and the mSafety OS can run on the device.
- A signed keystore containing the customer's public keys is provisioned to the wearable
- A unique device authentication key is generated and bound to the mSafety cloud back-end during the provisioning of the wearable. This is the private part of the TLS 1.2 client authentication key pair.
- The wearable is flashed with a package that contains the mSafety OS and an application. What OS version and application the package contains is specified by the customer in the mSafety Portal.

Secure boot

The secure boot is the root of trust that makes it possible to keep data on the wearable secure while also allowing the secure installation of software updates.

When a wearable is booted up, the decryption key loads and runs the platform core. The platform core then decrypts, loads and executes the mSafety OS containing the customer keystore.

Finally, the mSafety OS validates the integrity of the customer keystore, extracts the customer's public signing key and uses it to validate the customer's wearable application before it is executed. In parallel with starting the customer's application the mSafety OS activates the modem and starts to authenticate the wearable with the mSafety cloud back-end.



Updates over the air

Downloaded packages pushed to the wearable are always verified with the customer signing key before they are installed. This makes it impossible for malicious packages to be installed.

Wearable on-boarding to the mSafety cloud back-end

After a wearable has been bound to a specific customer's back-end during the production process, it will keep a "created" state in the mSafety Cloud back-end until it is started and bootstrapped. The first time a wearable connects to the cloud back-end, it is bootstrapped with a configuration (unique to the wearable) that is used in subsequent communication. The process also authenticates the device certificate and verifies authorization. When the wearable has been bootstrapped it can send data, through the mSafety back-end, to the customer's back-end.

mSafety cloud back-end security

The mSafety cloud back-end provides the following functions:

- Device authentication and device management.
- Message routing between a customer's wearable application and back-end.
- Wearable firmware management.
- User management.
- mSafety wearable SDK and OS distribution.

The mSafety Cloud back-end governs all communication between a customer's wearables and back-end so that it fulfils the security requirements and acts as the foundation of secure zero touch on-boarding. It also restricts traffic from mSafety wearables so that they can only communicate with a specific customer's endpoints configured in the mSafety Cloud back-end.

The mSafety cloud back-end uses a Store and Forward architecture to make sure that data is received by the customer's back-end. If the data is consumed by the customer's back-end, the retention time is 24h. In an error scenario, where the data is not consumed, the retention time is 96 h.

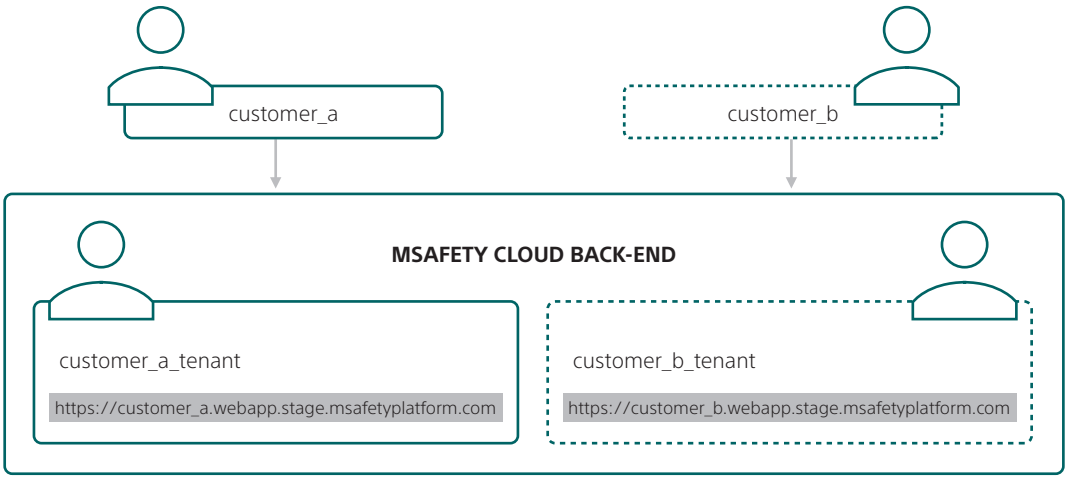
All communication between the wearable, the mSafety Cloud back-end and the customer's back-end uses TLS 1.2 across the different network segments.

Customer on-boarding

On-boarding of a customer in the mSafety Cloud back-end means that:

- A customer binds its back-end with the mSafety Cloud back-end and gets an account in the mSafety Portal.
- mSafety allows the segregation of development and production environments. The two environments are identical from a functional point of view, the difference lies in the provisioning of secrets.
- When the binding between a customer's back-end and the mSafety Cloud back-end is complete, it ensures that only correctly manufactured wearables can contact the customer's back-end.

Multi-tenant system



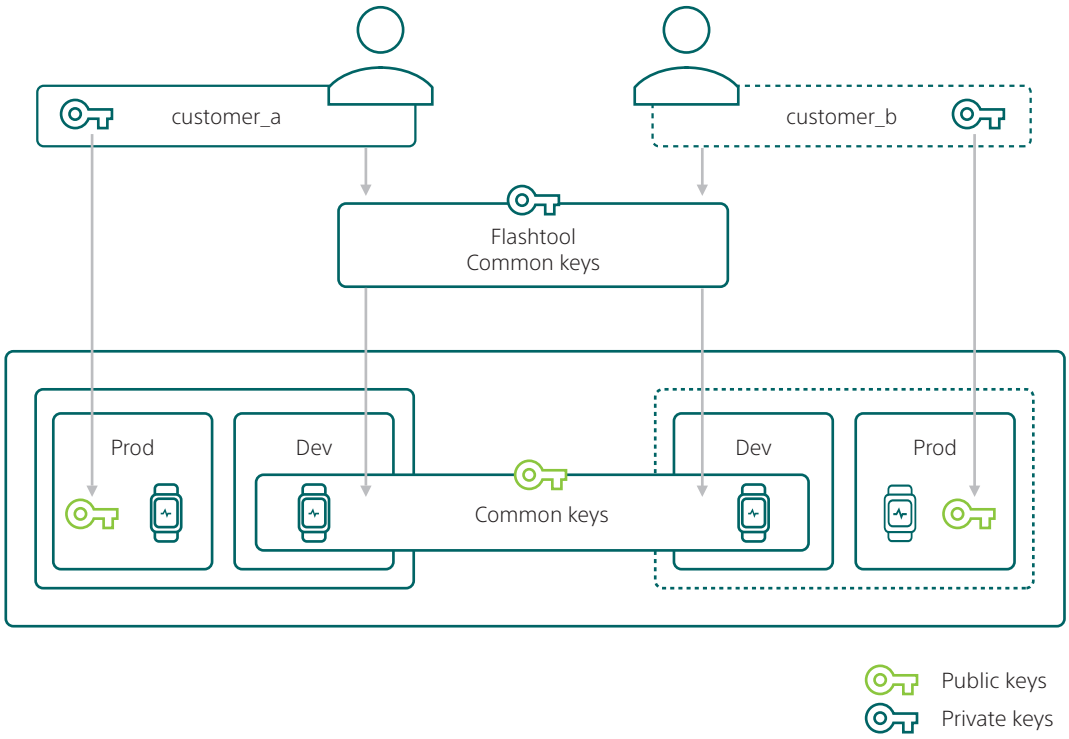
The mSafety cloud back-end is a multi-tenant system in which each customer is identified by a customer ID. The customer ID can be chosen by the customer, but it cannot be changed once set.

It's possible to configure an alias to be used instead of the actual customer ID if anonymity is required.

Unless an alias is configured, the customer ID is visible in the mSafety Portal URL. The customer ID is also the first part of the customer's MQTT topics for publishing and subscribing to wearable application data.

Wearables in the development and production environment

Each customer has access to one development and one production environment in



the mSafety Cloud back-end. Both environments are identical except for the data in the systems.

Wearables in the development environment are provisioned with the same public keys while wearables in the production environment are provisioned with customer specific public keys.

All wearables, regardless of environment, are bound to a customer using the device-specific Device Authentication Key.

mSafety wearable SDK

The mSafety wearable SDK contains, among other things, a tool called Flashtool.

Out of the box, Flashtool contains private signing and authorization keys that are common for all mSafety development wearables. These keys are used for flashing and signing packages used for updating development wearables.

To configure Flashtool for use in a production environment, the customer must generate customer specific private signing and authorization keys that replace the common keys. Those keys must then be stored safely by the customer.

Subscribe to wearable data

There are different ways for customers to subscribe to the data published by their wearables.

The basic method is to set up a Web Hook to which data from the customer's wearables is posted. Sony recommends the use of AWS security features for customers who intend to use AWS for their back-end application. Reasons:

- Allow users and services in the customer's AWS account to assume mSafety roles and call mSafety back-end APIs.
- Configure services, e.g. SQS or Kinesis, to receive data from the customer's mSafety wearables, via the mSafety Cloud back-end.

Adding users in the mSafety Portal

When a customer is onboarded to the mSafety Cloud back-end, a master administrator at the customer is configured. The administrator then adds further users through the mSafety Portal and assigns the roles:

- Admin
- Developer
- Power user
- User

These roles have different levels of access to the APIs of the mSafety Cloud back-end.

End-to-end (E2E) encryption

Before production of live wearables begins, it is the customers responsibility to implement end-to-end encryption of application data between the wearables and their back-end.

The mSafety OS provides libraries to help customers with this implementation process. In addition, a reference implementation for the customer's back-end is provided.

The solution offered by mSafety is based on the Noise Protocol Framework with exchange/agreement according to X25519 and AES256-GCM for encryption.

To implement end-to-end encryption the wearable application needs a customer specific static public key that is embedded into the customer's application and is protected by the secure boot chain. The corresponding static private key is part of the customer's back-end application. This key pair is generated by the customer.

The key pair is used as the starting point for deriving the symmetric keys necessary to implement bidirectional end-to-end encryption.

Order wearables

Production of wearables can be ordered from mSafety once:

- On-boarding is complete
- The customer's team has developed a secure wearable application
- The customer can prove that they have set up an E2E encrypted communication between their wearables and their back-end.

Customer considerations

Consider the following in order to adhere to industry best practice and align with proposed IoT Cyber Security Legislation like ETSI TS 103 645 and ETSI EN 303 645.

The mSafety customer should:

- Keep their private keys secure. mSafety does not keep copies of private customer keys. If a customer should lose a private key, it is irretrievably lost.
- Define a process for keeping the application software up to date, including but not limited to working with the latest version of the mSafety wearable SDK and the components it contains.
- Have a designated person in charge for incidents, including but not limited to security, as an interface to mSafety.
- Understand and make use of the development and production environments provided by mSafety.
- The safekeeping of usernames and passwords to the mSafety Portal.
- Only assign admin rights to the mSafety Portal when needed.
- Before production of wearables can begin, customers must implement end-to-end encryption of application data between wearables and their back-end.

Trademark and acknowlegement

Visit sonynetworkcom.com for more information.

Contact

Sony Network Communications Europe BV
Mobilvägen 4
223 62 Lund
Sweden

E-mail: msafety@sony.com

Website: sonynetworkcom.com/msafety