UBIRCH SIM Overview

Delivery

The UBIRCH client is provided as a SIM application (SIGNiT) and additional library code that handles the communication between customer code and the SIM card application. The library code is provided as open source. A test kit based on Pycom modules is available.

Library Source Repository: github.com/ubirch/ubirch-protocol-sim
Test Kit Source Repository: github.com/ubirch/ubirch-testkit

Requirements

- a system with access to a modem that supports `AT+CSIM` commands
- alternatively, a UBIRCH test kit

Security Considerations

The SIM application is protected by a unique PIN. The example test kit code handles retrieving the PIN from the UBIRCH backend. Developers should consider storing this PIN securely on the device, as it is the key to cryptographic functionality provided by the SIM application.

Functional Description

The SIM application client provides signature and chaining services to seal original data, generated on embedded devices through the SIM card. It takes care of packaging the hashed data and signing the package into the **UBIRCH PROTOCOL PACKET (UPP)**. Sending the UPP to the UBIRCH backend must be handled by the customer application. At the backend, the anchoring in the blockchain is performed. The backend can also be used to verify already anchored UPPs.

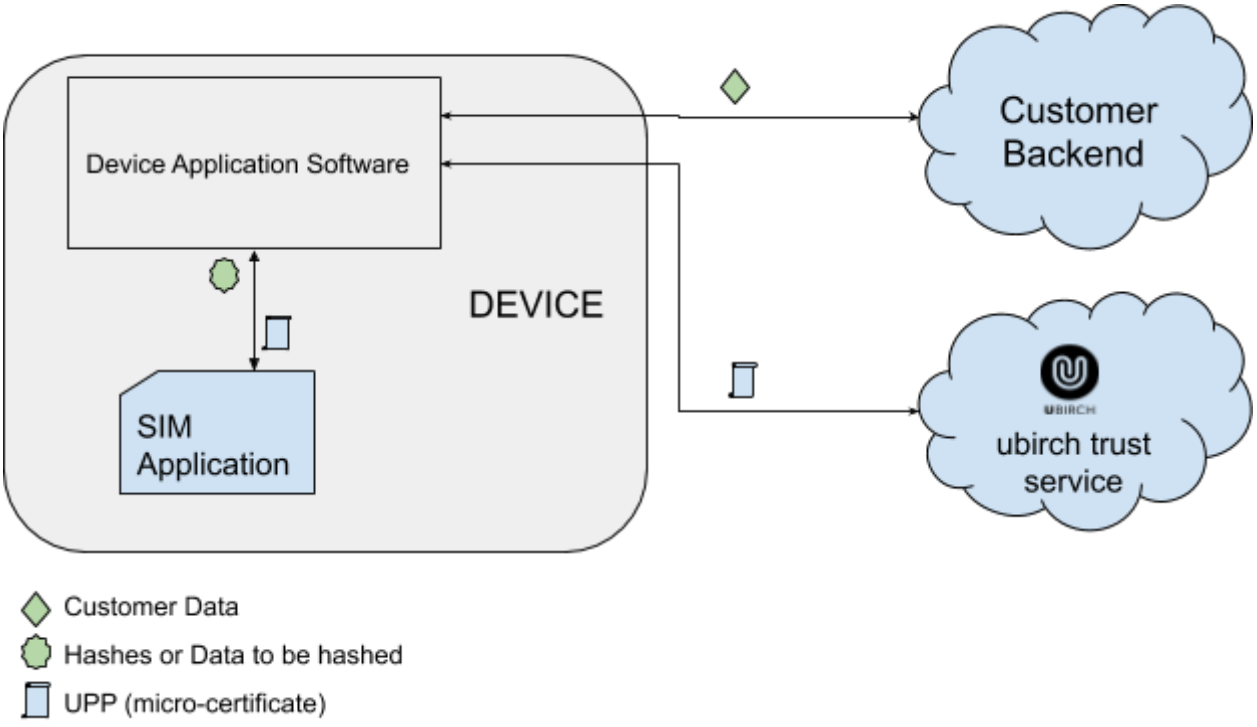
The original data must be stored in a customer database to be able to execute verification requests at a later stage. UBIRCH does not store any original sensitive data!

> ⚠ If the SIM Card is used together with the test kit, the sensor data is sent to the UBIRCH Simple Dataservice, which stores the data. It is an example for a data service to be implemented by the customer.

UPP data is sent to the SIM application via SIM APDU commands. The data encoding and handling of AT commands is done by the library code. Each SIM card comes pre-provisioned with a Universally Unique Identifier (UUID) and a cryptographic key pair that is registered with

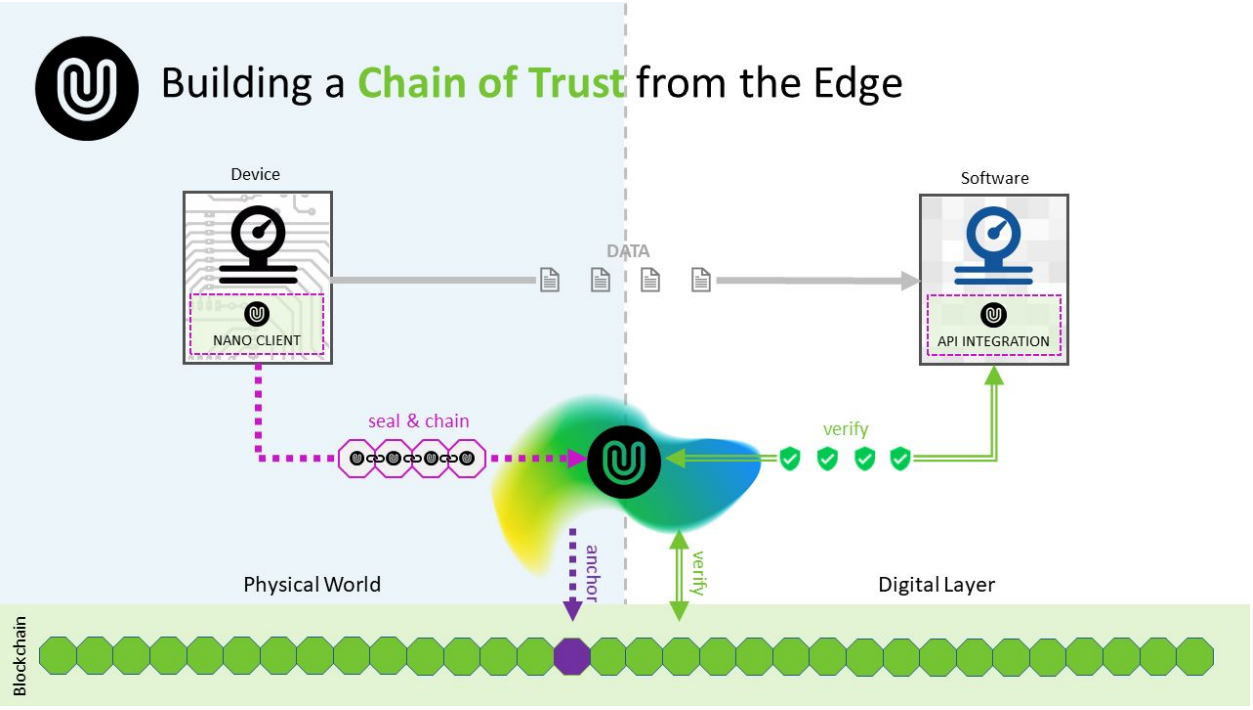
the UBIRCH backend system and just needs to be claimed using the IMSI of the SIM at the UBIRCH console.

To claim the SIM and start working with the SIM Application, simply follow the steps in Setup SIM card and device. If you still cannot manage to setup your device, please also check the FAQ and if this does not help, contact us



Overview: UBIRCH TRUST SERVICE

The UBIRCH Trust Service is a fast cloud-based backend responsible for identity management, blockchain anchoring, device and account management. It offers simple to use REST API endpoints to anchor incoming UPPs and to verify received data.



To improve performance, scalability and to keep transaction cost manageable, the UBIRCH Trust Service creates its own merkle-tree structure, aggregating incoming UPPs into larger root-hashes, which get anchored into a blockchain every minute.

The Trust Service is built as a Kubernetes cluster being hosted on Microsoft AZURE. All performance-critical components can equally be deployed on-premise, should the necessity arise. It is optimized for very high throughput.

Process

The following two sections will shortly describe the two most important processes of sealing and verifying data.

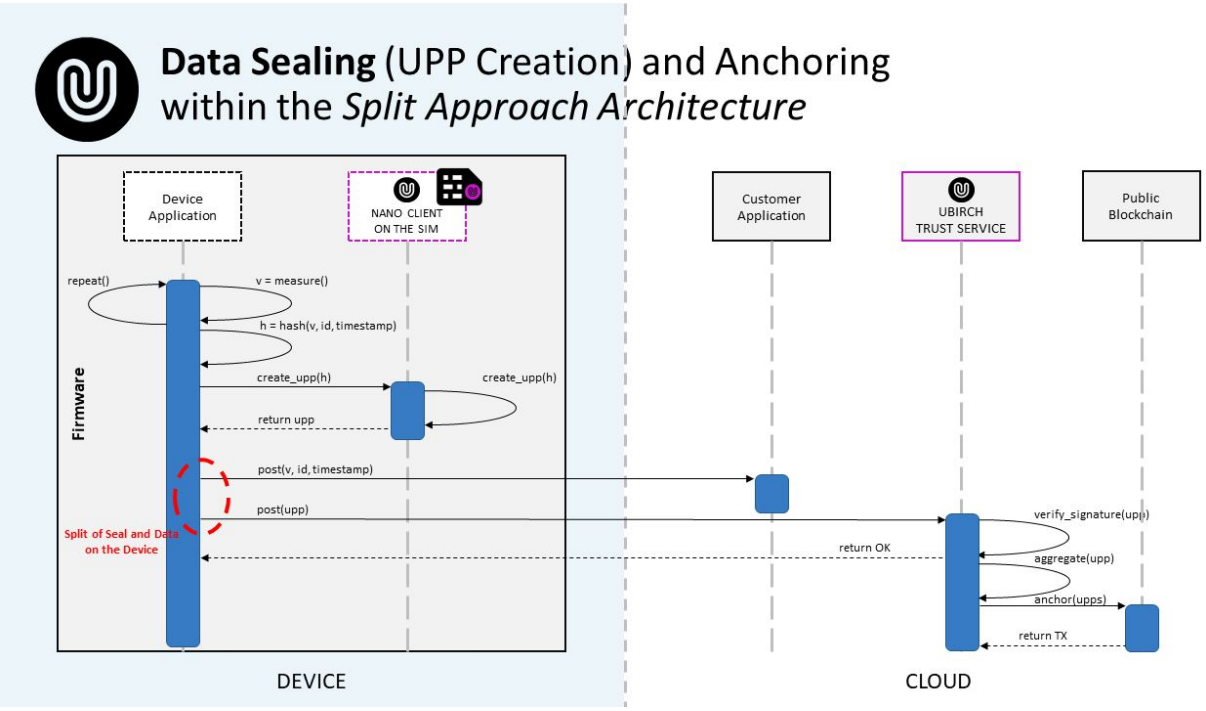
In general, the process always consists of:

1. Seal data at the point of its 'birth' with the UPP
2. Anchor the UPP into the blockchain

3. Verify received data against its UPP in the blockchain

Seal and Anchor Data

The following simplified sequence diagram uses pseudo code to show the process of sealing and anchoring data. This example shows the usage of an application which is sharing data /measurements, test results) with any kind of data receiver. This is just an example to show the process and not necessarily the exact final architecture to use UBIRCH.

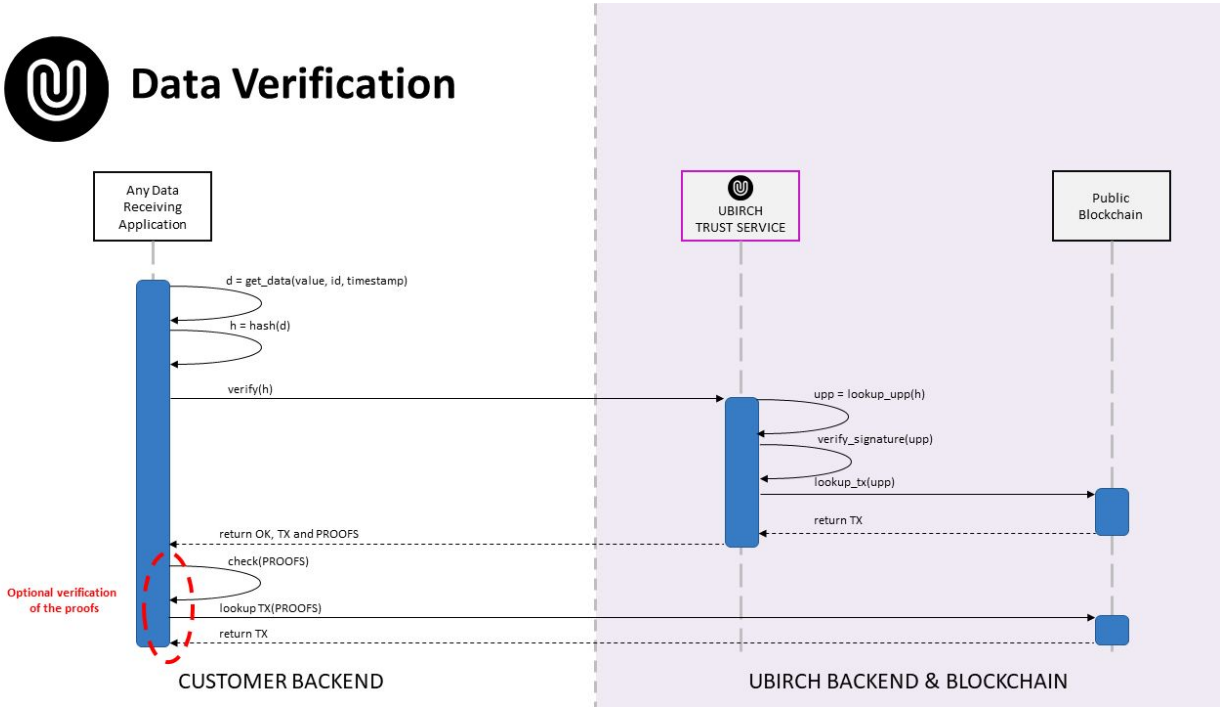


1. The customer device creates data
2. The customer device hashes the data, as a unique digital fingerprint
3. The customer device sends the hash to the UBIRCH NANO client on a SIM, where a signed UPP is created.
4. The customer device sends the data to the customer application for storage and further processing
5. The customer device sends the UPP to the UBIRCH TRUST SERVICE.
6. The UBIRCH TRUST SERVICE verifies the origin of the UPP by checking the signature.
7. The UBIRCH TRUST SERVICE aggregates the UPP
8. The UBIRCH TRUST SERVICE anchors the UPP in the public blockchain

Verify Data

Each received data packet which has been sealed with the UBIRCH CLIENT (has been *UBIRCHed*) at the place of its 'birth', can easily be verified by the receiver, regarding its authenticity, integrity and chain validity. Since the seal is not directly attached to the data and anchored to the blockchain, the verification can be done by anyone, who has (access to) the data. This process is completely independent from the channel of transmission, which has been used to share the data and is also beyond any system boundaries.

The following simplified sequence diagram uses pseudo code to show the process of verifying *UBIRCHed* data (HASH-method).



1. The customer application acquires the data to verify
2. The customer application recreates the hash of the data, like it was created on the device (before creating the original UPP)
3. The customer application sends this hash to the verification endpoint of the UBIRCH TRUST SERVICE
4. The UBIRCH TRUST SERVICE looks up the original UPP, based on the incoming hash
5. The UBIRCH TRUST SERVICE checks the signature of the found UPP
6. The UBIRCH TRUST SERVICE looks up the blockchain transaction, which contained the according UPP

7. The UBIRCH TRUST SERVICE returns OK (or not-OK) and all the proofs needed to cryptographically reproduce the validation result

Contact

UBIRCH GmbH
Im Mediapark 5
50670 Köln
https://ubirch.com

Email: info@ubirch.com
Helpdesk: https://ubir.ch/helpdesk
Developer Page: https://developer.ubirch.com/