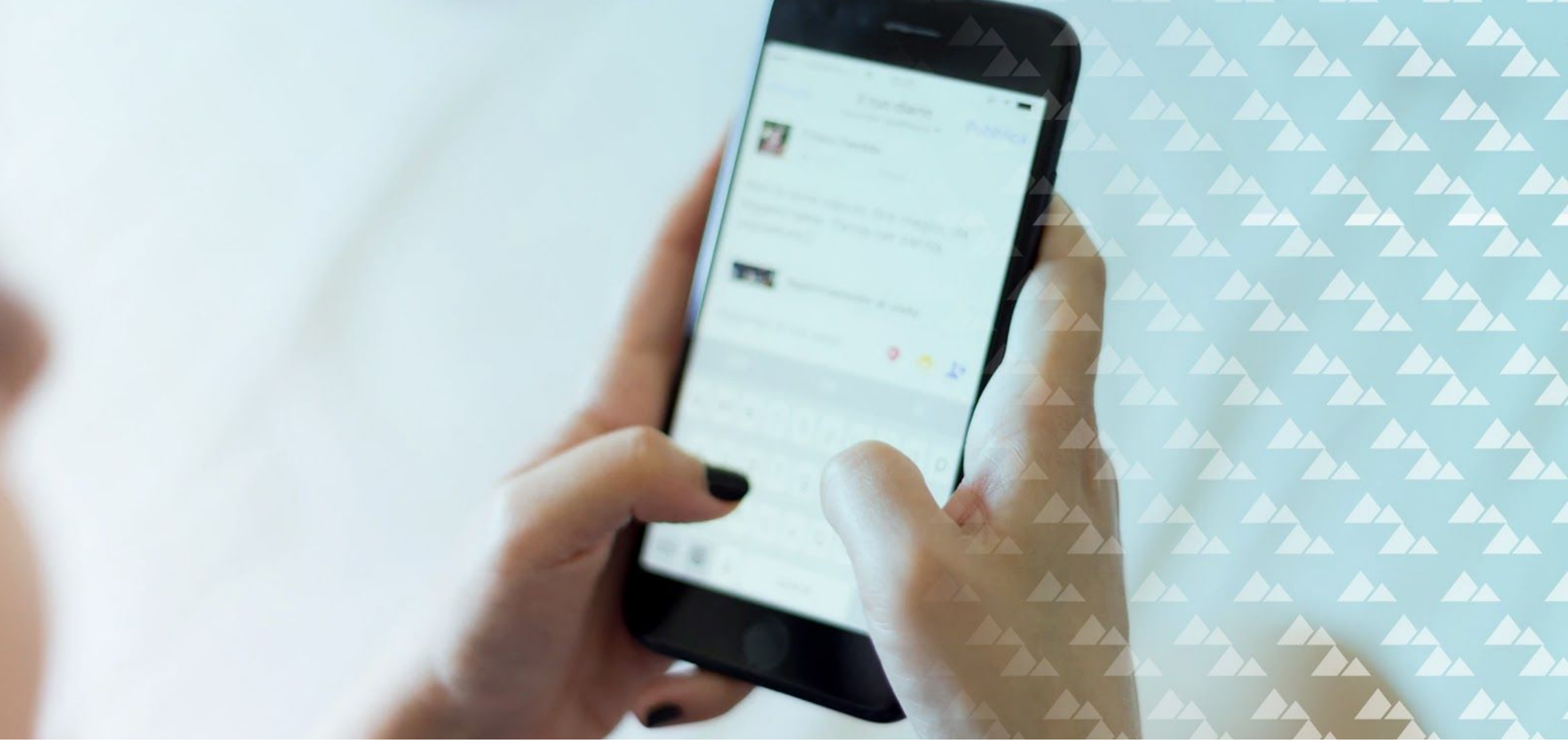


SECURITY IS OUR PRIORITY

IN OUR SMART SPACE SOLUTION, MANY DIFFERENT COMPONENTS COMMUNICATE CONTINUOUSLY WITH EACH OTHER. OUR PRIMARY GOAL IS ENSURING THIS COMMUNICATION IS RELIABLE AND CRYPTOGRAPHICALLY SECURE.

Digital security is a constantly changing field and something that needs to be tracked, evaluated and responded to regularly. With this in mind, we designed and built our ecosystem to rapidly deliver cryptographically-signed updates - including security fixes and improvements - transparently, without any impact on our customers or people interacting with our Smart Spaces. As well as being able to respond to security issues rapidly, we use state of the art, open-source, peer-reviewed and proven security standards and, in keeping with security best-practise, do not develop our own security components.

This document gives an overview of the current state and future development of each component related to security.

MOBILE PHONE

On both iOS and Android, wherever possible, we rely on the underlying Operating System's security frameworks. This means that any security updates issued by Google or Apple respectively is automatically applicable to the security of our application.

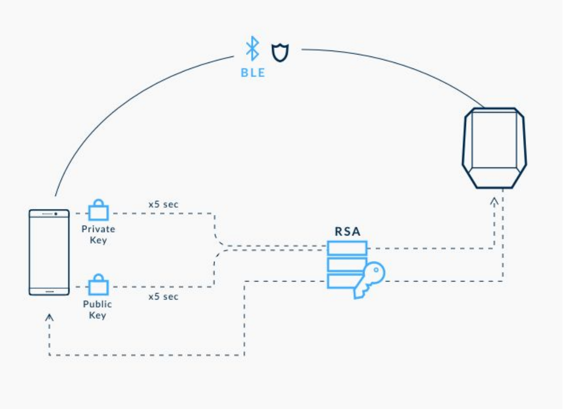
As well as this focus on cryptographic security, we also ensure that the user of the app is at authenticated at least once before it's possible to interact with the app, again through mechanisms powered by recommend best practise for the operating system. These are typically fingerprint or face recognition, or a passphrase.

CREDENTIAL STORAGE

iOS offers a central system to store secure credentials called Keychain[1]. Like all of Apple's own applications, we rely on this to store the credentials for accessing our backend infrastructure.

Android offers a similar central system for storing cryptographic keys and process ciphers called the AndroidKeyStore[2]. Again, we use it, as Google's own applications do, to encrypt and decrypt users' credentials.

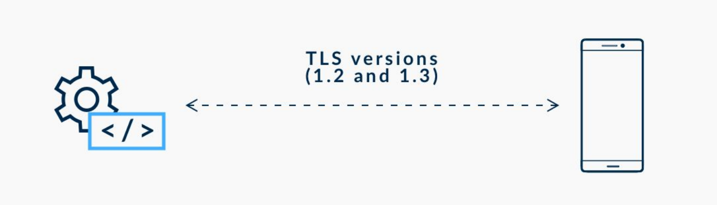
MOBILE PHONE AND ACCESS HUB



Communication between a user's mobile phone and our access control hardware (such as the Access Hub) occurs using Bluetooth Low Energy [3] (BLE). This technology is used in healthcare devices, smart tags, sports trackers, and many other fields, and is an advancement of existing Bluetooth technology.

On top of BLE's own authentication layer, we also encrypt all confidential data transmitted between mobile phone and our access control hardware with RSA[4] private-public-key encryption.

MOBILE PHONE AND BACKEND



For communication between users' mobile devices and our backend infrastructure, the most recent TLS[5] versions (1.2 and 1.3) are being used. To prevent man-in-the-middle attacks, our systems use HTTP

[1] https://developer.apple.com/documentation/security/keychain_services
[2] https://developer.android.com/training/articles/keystore
[3] https://en.wikipedia.org/wiki/Bluetooth_Low_Energy
[4] https://en.wikipedia.org/wiki/RSA_(cryptosystem)
[5] https://en.wikipedia.org/wiki/Transport_Layer_Security

public key pinning[6]. This ensures that any communication between a user's phone and our APIs will be denied if it has not been signed with our certificate.

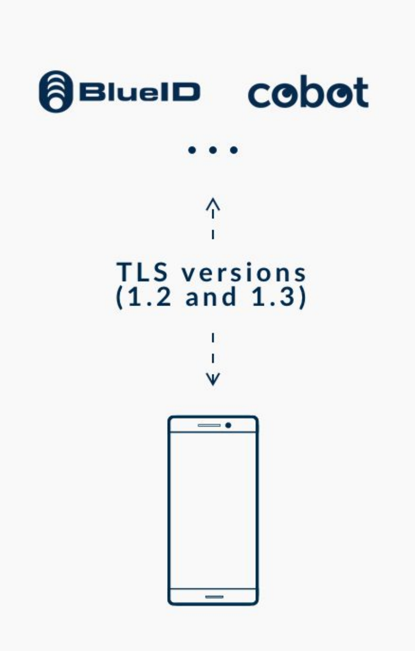
We use OAuth2[7], the industry standard, as our user authentication methodology. This ensures that each of a user's mobile devices has a unique token to communicate with our API, meaning we have fine-grained control of access and permissions. In case of an emergency, any of these tokens can be revoked or deleted directly from our administration systems. This would immediately revoke the device's access.

Additionally any communication with building infrastructure happens using the RSA private-public-key encryption in combination with AES encryption (128-GCM, 192-GCM or 256-GCM, depending on the support of the mobile phone). This means that any and all communication can not be decrypted by any other system communicating on the same network infrastructure.

MOBILE PHONE AND 3RD PARTY SYSTEMS

We believe our responsibility for security extends to our integrations with third-party suppliers such as BlueID[8] and MyRenz. With this in mind, we work with these partners to ensure they too are using the industry-standard tools, such as strong TLS encryption and public key pinning.

Similarly, when working with third-party identiy providers (such as OpenID, Cobot, or Google's Identity platform) we ensure that those services are also enforcing strong TLS (1.2, 1.3) encryption.



BACKEND

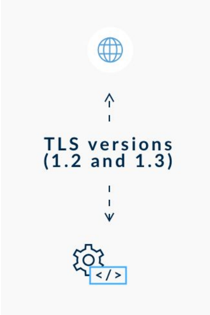
The backend is used to set up and manage the Sensorberg platform. It provides an overview and management of devices and sensors.

Our hosted service offering is situated in a German datacenter and consists of multiple hardware servers connected via an internal network. All internal traffic is shielded from network equipment not under Sensorberg's control. You can read more specifics about the on-site security directly at Hetzner.

[6] https://en.wikipedia.org/wiki/HTTP_Public_Key_Pinning
[7] https://oauth.net/2/
[8] https://www.blueid.net/technology/security/

Connections to the servers are only possible by the infrastructure team and only via a Virtual Private Network[9]. Accessing this VPN requires a short-lived, one-time password token and Secure Shell[10] with key exchange. All access attempts to the servers are logged in an audit log and every command and action performed on the server is streamed to a append-only central logging instance.

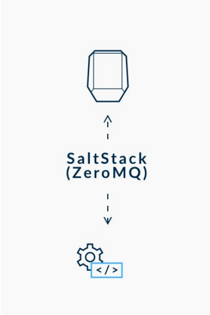
Our infrastructure is managed by various configuration management systems, all of which are under version control and automatically tested. We make hourly backups of all databases.



BACKEND AND BROWSER

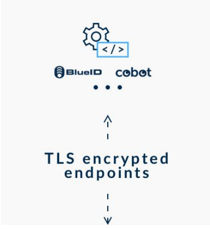
Communication between a user's browser and our management backend is only through strong TLS (1.2, 1.3) encryption. In addition, we use web-design best practises to avoid all common browser attack vectors, such as cross-site scripting and man-in-the-middle attacks.

All cookies related to our platformed are mandated as secure cookies[11]. This means that the content of the cookies can only be read only by our management platform's domain and only by scripts which were loaded over a TLS secure connection.



BACKEND AND ACCESS HUB

For performing an action via remote interaction (e.g.. closing shutters in the office from home), there is a need to tell the access hub to perform a specific action on an actuator. For this communication, we utilise SaltStack[12], which itself uses ZeroMQ. All communication within ZeroMQ is fully encrypted[13] using elliptic curves[14].



BACKEND AND 3RD PARTY BACKEND

For all third-party integrations (e.g. integration with a resource-management platform such as Cobot), we ensure that all communication again takes place only over secure

[9] https://en.wikipedia.org/wiki/Virtual_private_network
[10] https://en.wikipedia.org/wiki/Secure_Shell
[11] https://en.wikipedia.org/wiki/Secure_cookie
[12] https://www.saltstack.com/
[13] http://zeromq.org/topics:encryption
[14] https://en.wikipedia.org/wiki/Curve25519

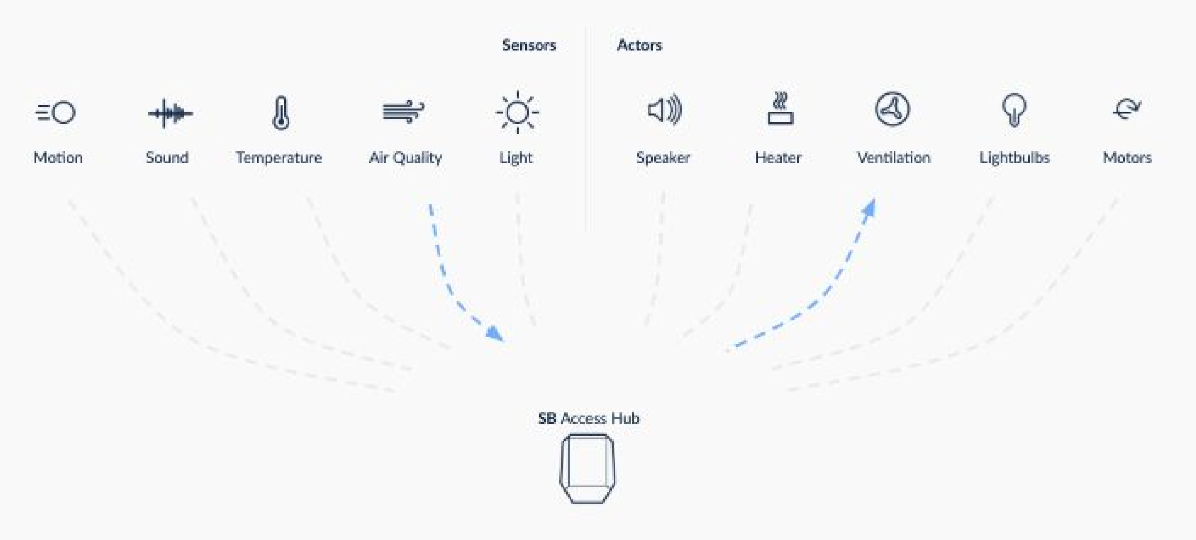
TLS-encrypted endpoints. At point is any information sent unencrypted over public networks.

ACCESS HUB

The Access Hub is the only element in our infrastructure which is directly connected to actuators and sensors.

These devices which should be part of a separate virtual local area network (VLAN). Since the access hub needs an internet connection, it is highly recommended to shield it from rest of the devices inside the local network. This prevents attack vectors, like brute-forcing or just overloading the device with too many requests.

ACCESS HUB AND SENSORS OR ACTUATORS



Security regarding communication between third-party sensors/actuators and Sensorberg's Access Hub depends on their individual support of communication and encryption protocols. While Sensorberg maintains a list of supported protocols, we cannot enforce strong security patterns on 3rd-party devices. With this in mind, we only ever send simple device commands and information to these devices, with no sensitive information such as user details ever being communicated to them. We are happy to discuss the security of each device type and protocol (e.g. ZWave[15]) in detail upon request.

DEVICE BRIDGE

The Bridge is used to marshall and monitor all Access Hubs within a locationIt is used for monitoring, maintenance purposes, access cache, and localised device control.

[15] https://en.wikipedia.org/wiki/Z-Wave#Security

It is a key component inside the building to ensure high-availability of the system, even when temporarily disconnected to the internet.

BUILDING PROCESSING UNIT AND BACKEND

All communication between our backend and local Device Bridges take place over a TLS-encrypted and certificate-based secure channel.

DEVICE BRIDGE AND ACCESS HUBS

All communication between Access Hubs and our Device Bridges take place within a dedicated VLAN and therefor separated on the network layer, making them opaque to all other devices on the network. In addition, the connection itself is again TLS- encrypted.

ADDITIONAL INFORMATION

For any additional information, please check our Documentation Portal[16], Developer Portal[17], or contact us at security@sensorberg.com.

[16] https://documents.sensorberg.com
[17] http://developer.sensorberg.com