4 Gegensprechanlage

Im Rahmen dieses Projekts wurde eine Gegensprechanalage zwischen einer Doorbird Türklingel und der ioBroker Visualisierung entwickelt.

4.1 Asterisk

Asterisk ist eine Open-Source Software zur Realisierung von Telefonanlagen. Es dient zum Aufbau und zur Steuerung von Telefongesprächen zwischen verschieden Telekommunikations-Endpunkten wie z.B. herkömmlichen Telefonen und VoIP-Geräten.

In diesem Projekt wird Asterisk dazu eingesetzt die Kommunikation zwischen der Doorbird Türklingel und der ioBroker Visualisierung im Webbrowser zu realisieren. Dabei kommt zum Aufbau und zur Steuerung der Kommunikationssitzung das Session Initiation Protocol (SIP) zum Einsatz.

4.1.1 Installation

Bevor Asterisk installiert werden kann müssen zuerst die notwendigen Compiler und Buildtools installiert werden. Diese kann z.B. über die Installation der folgenden Pakte mit apt (Advanced Packaging Tool) erreicht werden: gcc, g++, make, patch

Anschließend wird Asterisk heruntergeladen und entpackt. Für diese Projekt wurde die Version 17.1.0 verwendet. Nach dem Asterisk entpackt ist, muss zuerst das beiliegende Skript zum Installieren der benötigten Abhängigkeiten (contrib/scripts/install_prereq install) ausgeführt werden.

Als nächstes kann das Konfigurationskript (configure) ausgeführt werden. Dieses bereit die Kompilierung von Asterisk für das Zielsystem vor. Als nächstes sollte zusätzlich die Unterstützung für den Opus-Codec und einige Demo-Sounds aktiviert werden. Dazu wird zuerst „make menuselect.makeopts" ausgeführt. Dadurch kann das Programm „menuselect" dazu verwendet werden die Optionen „codec_opus", „CORE-SOUNDS-EN-ULAW" und „MOH-OPSOUND-ULAW" zu aktivieren.

Mit dem Befehl „make" wird Asterisk anschließend kompiliert. War die Kompilierung erfolgreich kann Asterisk mit dem Befehl „make install" installiert werden. Mit dem Befehl „make config" kann Asterisk als Systemdienst registriert werden, wodurch es beim Systemstart automatisch mitgestartet wird.

4.1.2 Konfiguration

Nachdem Asterisk installiert wurde muss es konfiguriert werden. Dazu werden die folgenden Konfigurationsdateien unter „/etc/asterisk" erstellt/angepasst:

- modules.conf
- http.conf
- rtp.conf
- pjsip.conf
- extensions.conf

Im Rahmen des Projekts wurde passende Konfigurationsdateien erstellt, die einfach in das Verzeichnis kopiert werden können. Neben den Konfigurationsdateien ist auch noch ein TLS-Zertifikat erforderlich. Im Folgenden werden urz der Zweck der Konfigurationsdateien erläutert und anschließend gezeigt wie das TLS-Zertifikat erstellt wird.

modules.conf

Asterisk bietet zwei Treiber zur Verwendung von SIP: „chan_sip" und „chan_pjsip". Da im Folgenden „chan_pjsip" verwendet wird, wird zur Vermeidung von Störungen „chan_sip" deaktiviert. Dazu wird der „modules.conf" die Zeile „noload => chan_sip.so" hinzugefügt.

http.conf

In der „http.conf" wird der HTTP- und HTTPS-Server konfiguriert. HTTP wird auf Port 8088 festgelegt und HTTPS auf Port 8089. Der Eintrag „tlscertfile" muss für die Verwendung von HTTPS auf ein gültiges TLS-Zertifikat verweisen.

rtp.conf

Die „rtp.conf" enthält die Konfigurierung für das Real-Time Transport Protocol (RTP), das nach Aufbau der Kommunikationssitzung via SIP zur Übermittlung der Daten verwendet wird.

pjsip.conf

In der „pjsip.conf" werden die Kommunikationsendpunkte definiert. Zuerst werden die über WebRTC angebunden Endpunkte definiert, danach die klassischen über UDP verbunden Clients.

extensions.conf

Die „extensions.conf" dient zur Steuerung der Anrufe. Hier wird festgelegt welche Telefonnummer zu welchem Client gehört. Darüber hinaus wird unter der Telefonnummer 200 eine Testnachricht abgespielt.

TLS Zertifikat

Als letztes muss ein gültiges TLS-Zertifikat für die verschlüsselten Kommunikationsverbindungen erstellt werden. Dazu wird zuerst ein Ordner „cert" unter „/etc/asterisk" erstellt. In diesem wird zuerst für das CA-Zertifikat ein RSA-Schlüsselpaar erstellt. Anschließend kann damit das CA-Zertifikat (ca.crt) erstellt werden.

```
openssl genrsa -out ca.key 4096
openssl req -x509 -new -nodes -key ca.key -sha256 -subj '/CN=Asterisk Root CA/O=HHN/C=DE' -days 3650 -out ca.crt
```

Im nächsten Schritt wird wieder RSA-Schlüsselpaar erstellt. Dieses Mal jedoch für das TLS-Zertifikat für Asterisk. Danach wird eine temporäre Konfigurationsdatei angelegt, die festlegt, dass das Zertifikat kein CA-Zertifikat ist, es für Signierung und Verschlüsselung verwendet werden kann und für die angegebene IP-Adresse gilt.

```
openssl genrsa -out asterisk.key 2048
cat > csr.cnf <<-EOF
[req]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment
subjectAltName = IP:$ipAddress
EOF
```

Nun kann ein CSR (Certificate Signing Request) für das TLS-Zertifikat erstellt werden. Mithilfe des CSRs, der Konfigurationsdatei, dem CA-Zertifikat und des CA-Schlüssels kann anschließend das TLS-Zertifikat erstellt und signiert werden.

```
openssl req -new -sha256 -key asterisk.key -subj '/CN='$ipAddress -out asterisk.csr
openssl x509 -req -in asterisk.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out asterisk.crt -days 3650 -sha256 -extensions req -extfile csr.cnf
```

Als letztes muss das Zertifikat noch in das passende Format gebracht werden. Mit den folgenden Befehlen wir eine PEM-Datei erstellt die den Schlüssel im PCKS#8 Format und das Zertifikat enthält:

```
openssl pkcs8 -topk8 -inform PEM -outform PEM -nocrypt -in asterisk.key -out asterisk.pem
cat asterisk.crt >> asterisk.pem
```

Damit dem Zertifikat vertraut wird muss das CA-Zertifikat im Webbrowser, aus dem später mit Asterisk kommuniziert wird, importiert werden. In Chrome wird dazu beispielsweise in den erweiterten Einstellungen der Punkt „Zertifikate verwalten" unter „Datenschutz & Sicherheit" aufgerufen. In dem nun geöffneten Dialog (Abbildung 61) wird der Tab „Vertrauneswürdige Stammzertifizierungstellen" ausgewählt. Dort kann über den Button „Importiern" das CA-Zertifikat (ca.crt) importiert werden.

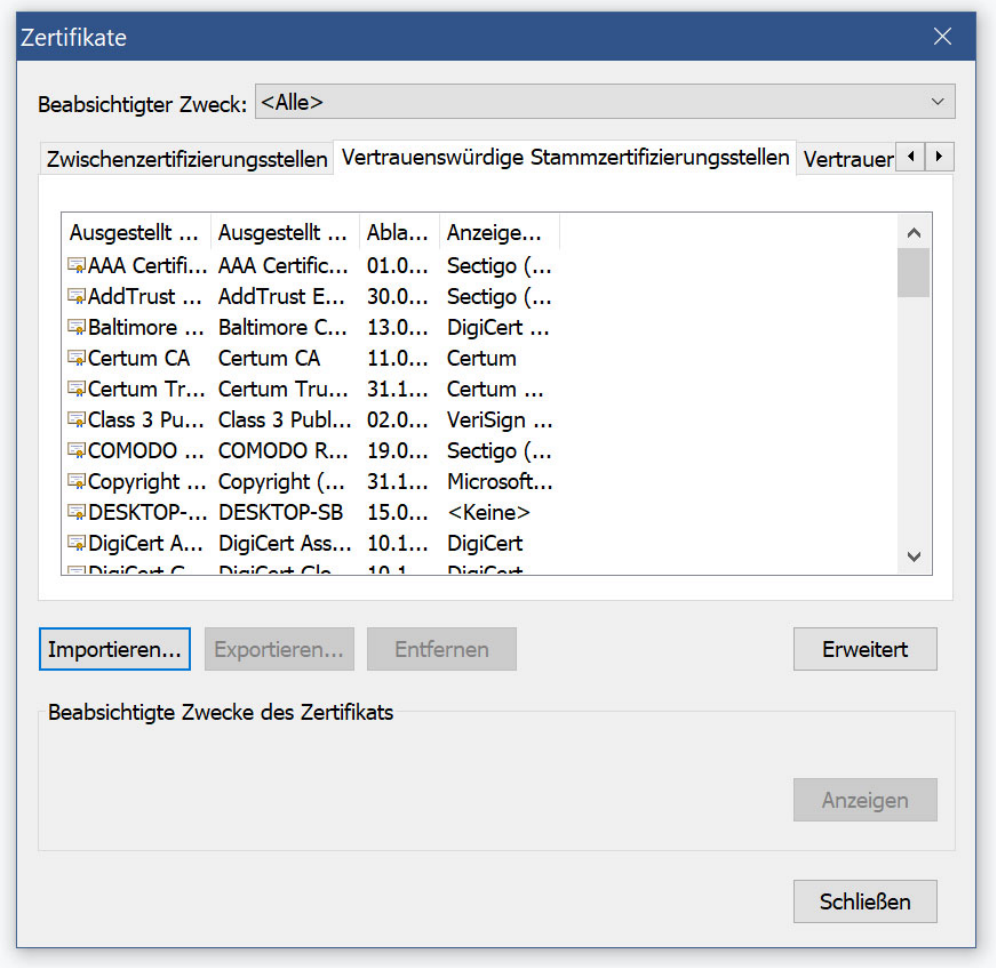

Abbildung 61 - Zertifikatsverwaltung Chrome

Die gleiche Vorgehensweise lässt sich leicht auf andere Browser übertragen. Bei Firefox ist es zu beachten, dass im Dialog, der nach Auswahl der Zertifikatsdatei erscheint, wie in Abbildung 62, der Haken bei „Dieser CA vertrauen, um Webseiten zu identifizieren" gesetzt sein muss.

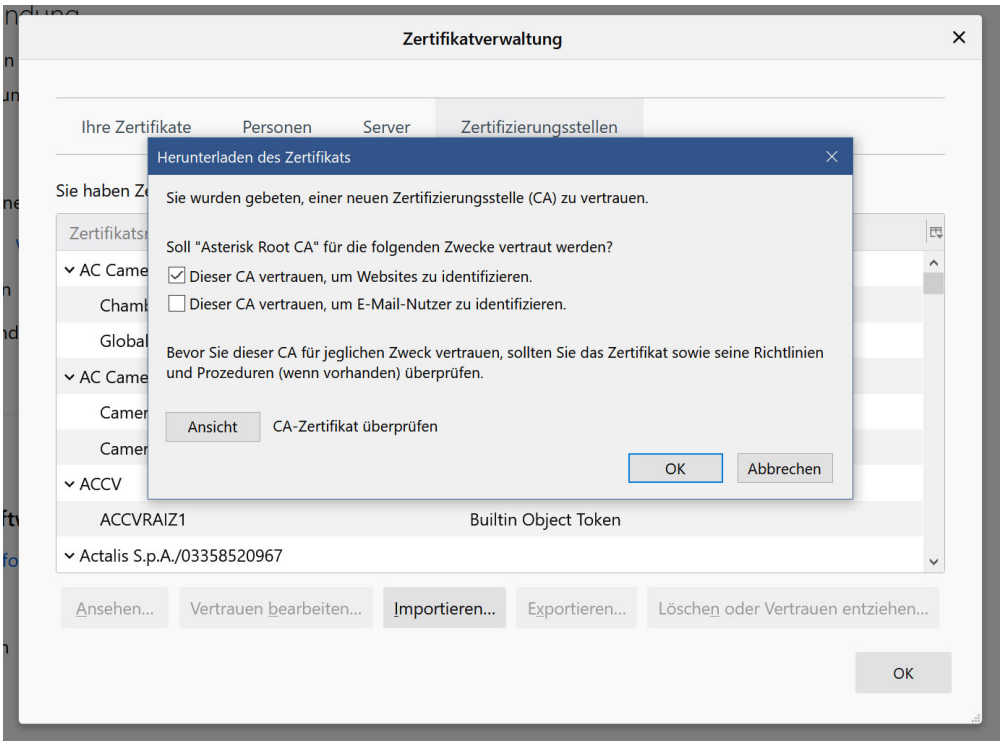

Abbildung 62 - Zertifikatsimport Firefox

4.1.3 Installationsskript

Um die Installation und Konfiguration von Asterisk zu automatisieren wurde im Rahmen dieses Projekts ein Skript erstellt, das diese Aufgabe übernimmt. Das Skript lädt alle benötigten Programme herunter, installiert Asterisk, kopiert die beiliegenden Konfigurationsdateien auf das System und generiert das benötigte TLS-Zertifikat. Dabei muss die IP-Adresse, unter der das Gerät später erreichbar ist, eingegeben werden. Wurde das Skript erfolgreich ausgeführt muss nur noch das CA-Zertifikat in die entsprechenden Webbrowser importiert werden.

4.2 HTTPS für die ioBroker-Visualisierung

Die Kommunikation mit Asterisk aus der ioBroker-Visualisierung heraus ist nur dann möglich, wenn die Visualisierung via HTTPS aufgerufen wird. Damit dies möglich ist muss zuerst HTTPS aktiviert werden.

Zuerst muss dafür ein Zertifikat erstellt werden. Dazu kann das Skript „init-ioBoker-cert.sh" verwendet werden. Dieses erzeugt einen Ordner mit allen notwendigen Dateien. Das CA-Zertifikat (ca.crt) muss in allen Webbrowsern/Systemen, in denen die Visualisierung verwendet wird, importiert werden.

In ioBroker muss nun der als Schraubenschlüssel dargestellte „System"-Button betätigt werden. In der „System"-Ansicht wird dann der Reiter „Zertifikate" ausgewählt. In diesem Reiter werden, wie in **Error! Reference source not found.** gezeigt, zwei Einträge erstellt: Einer für das Zertifikat (ioBroker.crt) und ein weiterer für den privaten Schlüssel (ioBroker.key).

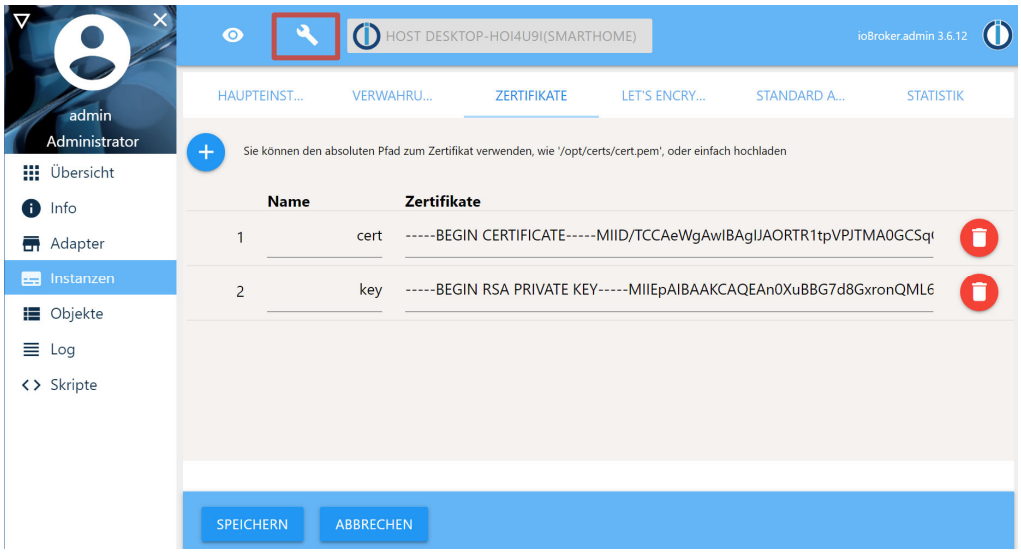

Abbildung 63 - ioBroker System-Ansicht

Jetzt kann die Konfiguration des „Web"-Adapters geöffnet werden. Dazu wird das Schraubenschlüssel-Symbol der Adapterinstanz betätigt (siehe Abbildung 64). In der in **Error! Reference source not found.** gezeigten Konfigurationsansicht muss anschließend die Checkbox „Verschlüsselung (HTTPS)" aktiviert werden. Als öffentliches Zertifikat muss der zuvor angelegte Eintrag „cert" und als privates Zertifikat der Eintrag „key" gewählt werden. Sollte im Feld „Socket.IO Instance" etwa anders wie „integriert" eingetragen sein, ist für den dazu verwendeten Adapter ebenfalls die Aktivierung von HTTPS erforderlich.

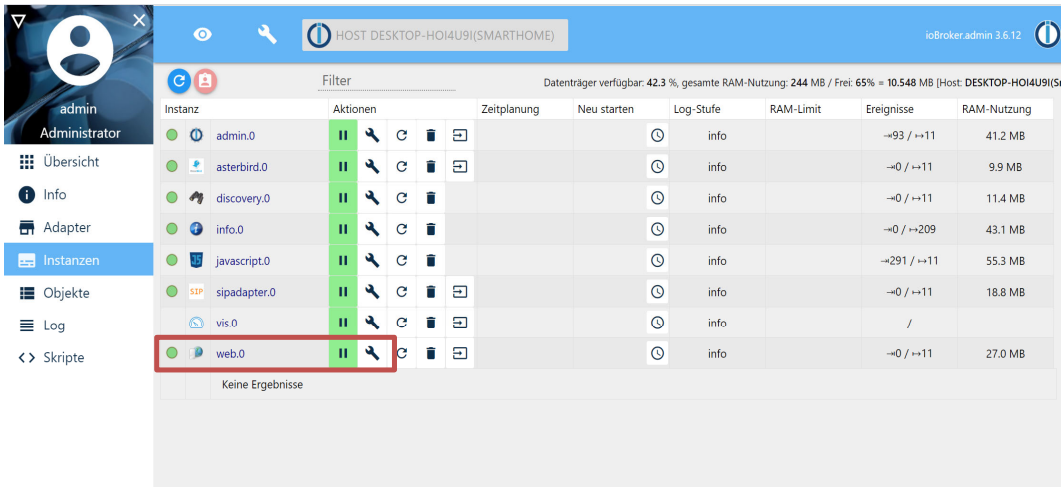

Abbildung 64 - Web-Adapter Instanz

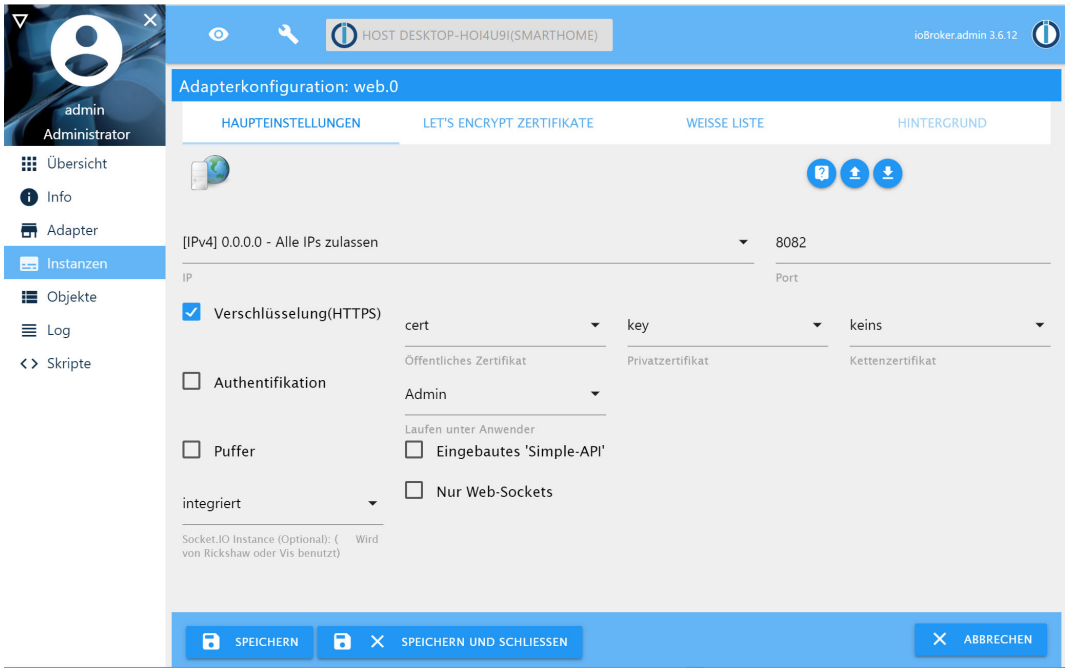

Abbildung 65 - Web-Adapter Konfiguration

4.3 Adapter

Adapter stellen in ioBroker Schnittstellen zu anderen Systemen dar. Durch Adapter lässt sich ioBroker um weitere Funktionalitäten erweitern. Adapter können auch selbst implementiert und aufgespielt werden.

Hierzu kann auf eine Adaptervorlage zurück gegriffen werden welcher schon die wichtigsten Komponenten besitzt. Dies kann über das folgende Repository bezogen werden: https://github.com/ioBroker/ioBroker.template.

In diesem Fall haben wurde ein Adapter mit VIS für die Kommunikation über SIP mit einer Türsprechanlage entwickelt.

4.3.1 Installation

Um den Adapter zu installieren öffnet man zunächst im Hauptmenü das Adaptermenü unter dem Reiter „Adapter". Danach öffnet man das „Adapter aus eigner URL installieren" Menü (Abbildung 66Abbildung 66).

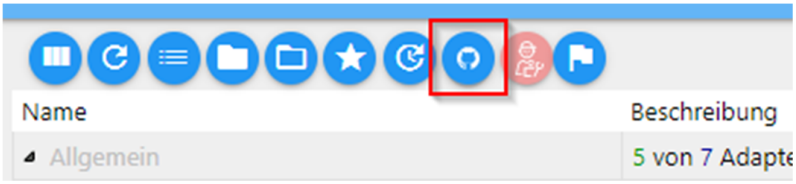

Abbildung 66 - Adapter aus eigner URL installieren

Anschließen muss der Reiter „BELIEBIG" ausgewählt werden (Abbildung 67). Im Eingabefeld „URL oder Dateipfad" kann nun der Pfad des Adapters angegeben werden. In diesem Fall liegt der Adapter in einem Git Repository. Abschließend kann der Adapter durch einen bestätigen des „INSTALLIEREN" Buttons installiert werden.

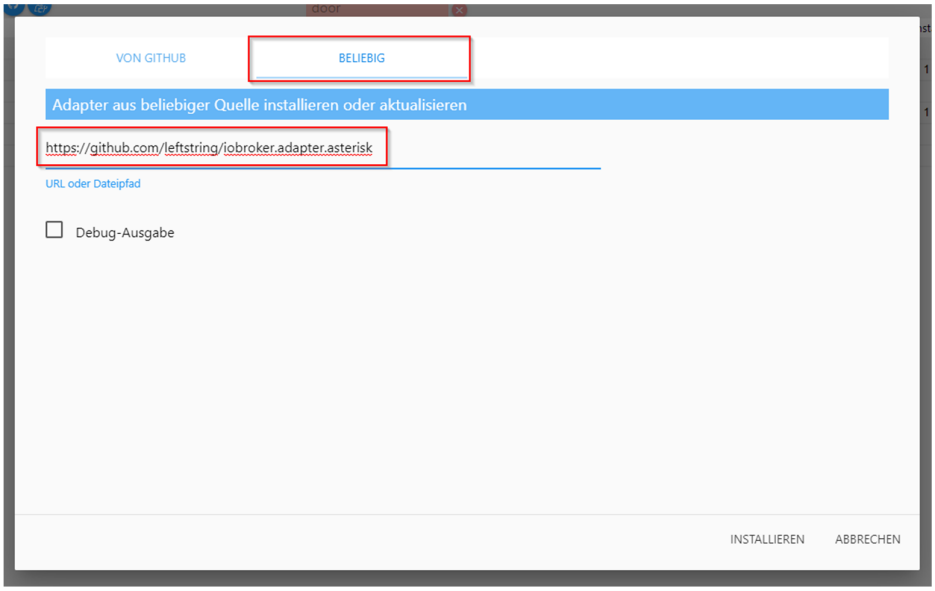

Abbildung 67 – Installationsmenü

4.3.2 Konfiguration

Nach dem die der Adapter installiert wurde und eine Instanz angelegt wurde kann der Adapter konfiguriert werden. Die Konfiguration kann man erreichen in dem man zunächst im Hauptmenü den Reiter „INSANZEN" auswählt und dann bei der entsprechenden Instanz auf den Schraubenschlüssel klickt. Danach sollte die Adapterkonfiguration wie in Abbildung 68 sichtbar sein.

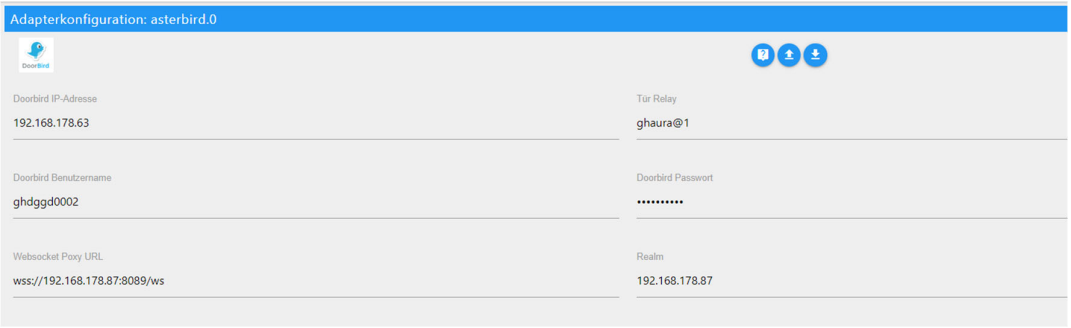

Abbildung 68 - Adapterkonfiguration

Hier muss nun die IP-Adresse des Doorbird Controllers eingetragen werden. Zu dem Werden die Zugangsdaten für einen Doorbird-Account benötigt. Wenn man die Tür öffnen Funktion nutzen möchte muss der demensprechende Türkontakt eingetragen werden.

Um eine Verbindung zu Asterisk aufbauen zu können muss noch der entsprechende Websocket Proxy und die Realm IP eingetragen werden.

Wichtig ist, dass sowohl Doorbird als auch Asterisk davor demensprechend konfiguriert werden mussten.

Um die Konfiguration abzuschließen, muss das dazugehörige Widget in der VIS platziert werden. Anschließen wird beim ersten Öffnen der VIS ein Dialog eingeblendet, hier müssen noch die SIP-Zugangsdaten eingetragen werden (Abbildung 70 und Abbildung 70).

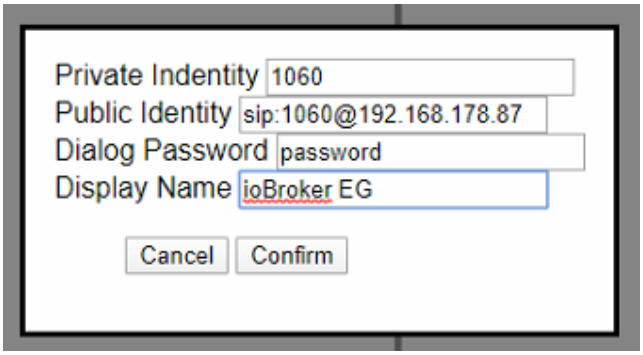

Abbildung 69 - Widget Konfiguration 1

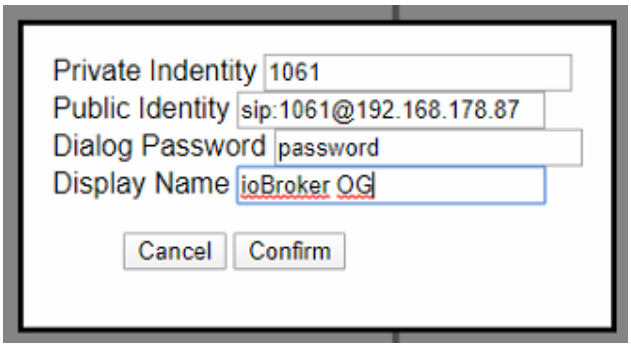

Abbildung 70 - Widget Konfiguration 2

4.3.3 Vorschaubild und Videostream

Der entwickelte Adapter zeigt sobald es an der Türe klingelt ein Vorschaubild. Dazu wird vom Doorbird-Controller jede Sekunde über dessen HTTP-API ein Bild angefordert und angezeigt. Sobald der Benutzer sich entscheidet gegenzusprechen wird auf einen Videostream umgeschaltet. Der Grund weshalb der Videostream erst beim Gegensprechen angezeigt wird ist, dass der Doorbird immer nur einen Videostream gleichzeitig erlaubt.

4.3.4 Tür öffnen

Über das Widget lässt sich auch die Türe öffnen. Da der dafür notwendige Aufruf aufgrund der Same-Origin-Policy nicht direkt aus dem Widget (Webbrwoser) möglich ist, wird die Adapterinstanz (Backend) über ein Flag benachrichtigt. Dort wird dann im Node.js-Code über die HTTP-API des Doorbird das entsprechende Relay angesprochen. Dieses lässt sich in der Konfiguration der Adapterinstanz festlegen.

4.3.5 SIP-Anbindung

Die Kommunikation aus dem Widget heraus mit Asterisk wurde mit der sipML5 API realisiert. Diese ist komplett in JavaScript geschrieben und erlaubt es via SIP und WebRTC Audio und Video Anrufe durchzuführen.

Da die Verwendung von WebRTC von den Webbrowsern nur von sicheren Webseiten (HTTPS) aus erlaubt ist, ist es nötig ioBroker, wie in Kapitel 4.2 beschrieben, für die Verwendung von HTTPS zu konfigurieren. Ebenfalls notwendig ist die Konfiguration von TLS für Asterisk, da auch nur sichere WebRTC Verbindungen erlaubt sind.

Für Lautstärkeregelung wurde zwei Buttons und ein Schieberegler eingebaut, die es erlauben die Lautstärke schnell und einfach zu regeln. Der Standardlautstärke ist dabei genau auf die Mitte eingestellt.

4.4 Konfiguration Doorbird

Um die Doorbird Türsprechanlage für die Kommunikation mit dem Adapter zu konfigurieren. Muss zunächst die Doorbird App aus dem Google Playstore oder App Store heruntergeladen werden. Um die Türsprechanlage zu konfigurieren muss man sich mit dem Admin Account auf dem Doorbird Gateway anmelden. Anschließen können in den Einstellungen unter dem Punkt „SIP Settings" die Zugangsdaten des SIP Servers eingetragen werden, wie auf Abbildung 71 zu sehen.

Um die Daten zu übernehmen muss der „Save"-Button oben rechts gedrückt werden. Wenn der Verbindungsaufbau funktioniert hat wird der Code 200 im Feld „Last error code" zurückgegeben.

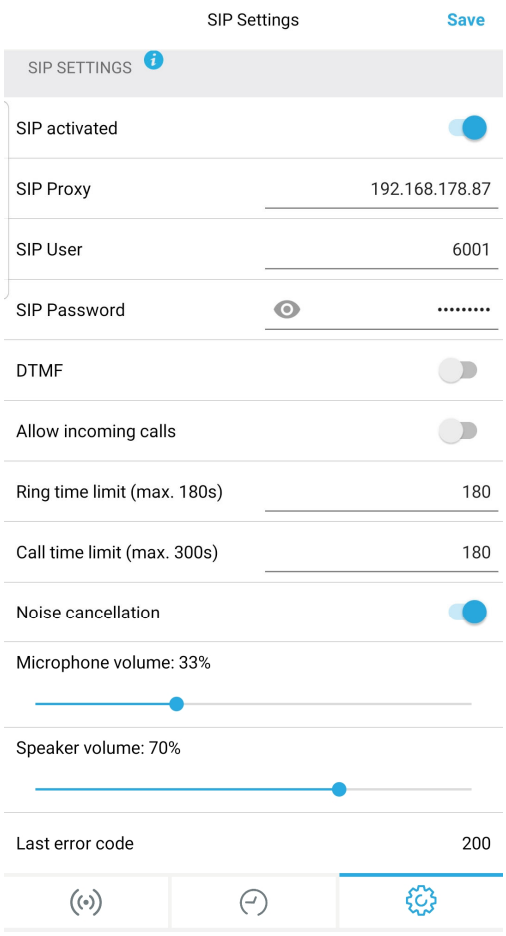

Abbildung 71 - SIP Settings

Anschließend müssen die SIP Anrufe konfiguriert werden. Also welche SIP Clients angerufen werden sollen. Diese können in den Einstellungen unter dem Punkt „SIP Calls" (Abbildung 72) gefunden werden. Mit dem Button „Add" können lassen sich SIP Clients hinzufügen, die mit der entsprechenden Konfiguration (Abbildung 73) angerufen werden.

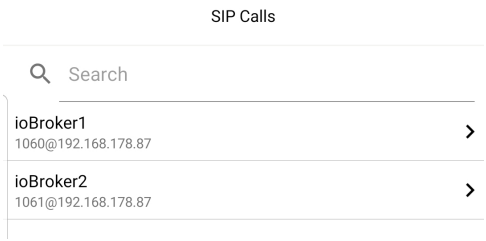

Abbildung 72 - SIP Calls

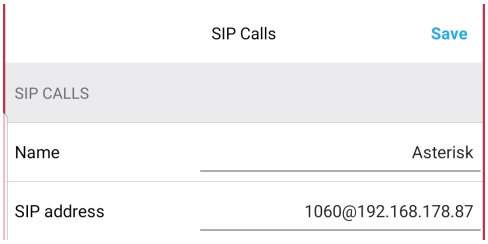

Abbildung 73 - SIP Calls Settings

Damit die SIP Clients bei Betätigung der Klingel angerufen werden, muss unter dem Einstellungspunkt „Schedule for actions" noch etwas konfiguriert werden (Abbildung 74). In diesem Menüpunkt kann man Einstellen was passieren soll, wenn geklingelt wird, dies kann auch abhängig von der Uhrzeit konfiguriert werden. Zunächst muss bei der Selektion oben links „SIP Calls" ausgewählt werden. Anschließend kann man über den kleinen Pfeil oben in der Mitte die verschiedenen Clients auswählen und entsprechend konfigurieren. Über den Button oben rechts kann man die Zeittabelle komplett aktivieren bzw. deaktivieren.

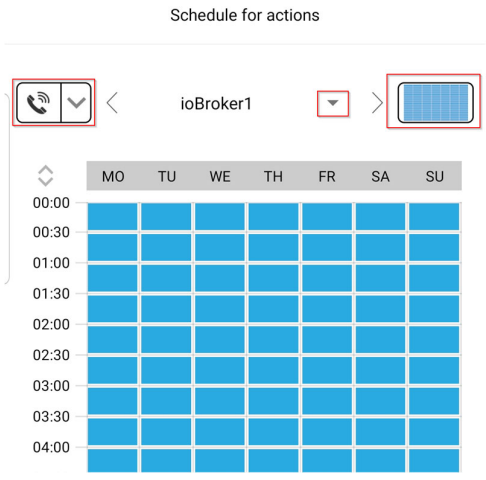

Abbildung 74 - Doorbird Schedule for actions

Wichtig ist es, dem User der später in den Adapter eingetragen wird, entsprechend die API zu aktivieren.

4.5 Zusätzlicher SIP-Adapter

Neben dem Adapter für die Doorbird Türklingel wurde im Rahmen des Projekts auch noch ein generisch verwendbarer reiner SIP-Adapter mit Widget entwickelt. Dieser enthält nur die SIP-Funktionalität. Das heißt es können, wie in Abbildung 75Abbildung 75 gezeigt, Anrufe getätigt und Empfangen werden.

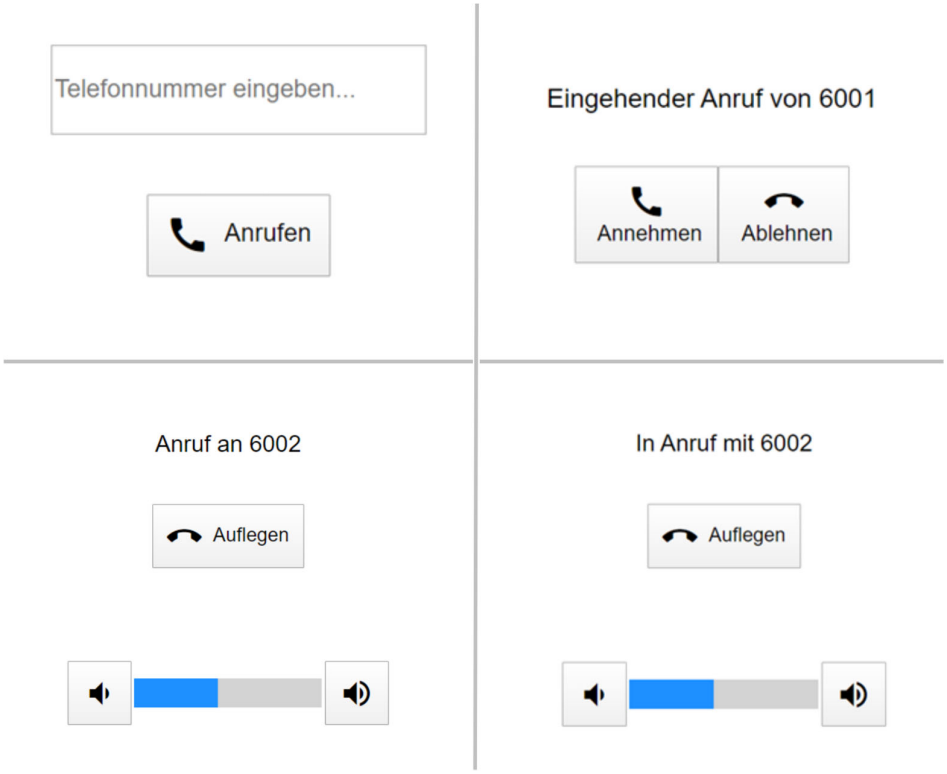

Abbildung 75 - SIP-Adapter

5 SIP-Adapter Step-by-Step Anleitung

Schritt 1. Asterisk installieren (auf Linux)

1. Installationsskript ausführen

   ```
   ./install-asterisk.sh
   ```

   ➔ IP-Adresse des Geräts eingeben
   ➔ Weitere Nachfragen bestätigen

2. CA-Zertifikat in benötigte Webbrowser importieren

Schritt 2. Zertifikat für ioBroker erstellen

1. Skript für Erstellung des Zertifikats ausführen

   ```
   ./init-ioBroker-cert.sh
   ```

   ➔ IP-Adresse des Geräts auf den ioBroker läuft eingeben

2. CA-Zertifikat in benötigte Webbrowser importieren

Schritt 3. HTTPS für ioBroker aktivieren

Siehe Kapitel 4.2

(Verwenden Sie die Dateien aus dem vom Skript erzeugten „cert"-Ordner)

Schritt 4. Adapter installieren

Siehe Kapitel 4.3.1

Schritt 5. Clients konfigurieren

Siehe Kapitel 4.3.2

Hinweise:

- Die IP ist durch die IP der Maschine, auf der Asterisk ausgeführt wird, zu ersetzen
- Identity und Passwort sind vom Benutzer abhängig
- Beim SIP-Adapter werden alle Einstellungen im Widget vorgenommen