univention
be open.

UCS Intercom Service app

Release 1.2

Univention GmbH

Oct 25, 2022

CONTENTS

Welcome to the documentation about the Univention app **UCS Intercom Service**. The app installs the ICS (UCS Intercom Service), an intermediary for communication between applications like *Nextcloud*, *OX App Suite* and *Matrix*. The functionalities *File-picker*, *Video conference*, create and accessing the *Univention-Portal navigation* endpoint from other apps require the app **UCS Intercom Service**.

This documentation is for system administrators who operate the app **UCS Intercom Service** from Univention App Center connected to the LDAP directory in UCS (Univention Corporate Server). It covers the following topics:

1. *Architecture* (page 9)
2. *Installation* (page 3)
3. *Configuration* (page 5)
4. *Requirements and limitations* (page 15)
5. *Troubleshooting* (page 17)

This documentation doesn't cover the following topics:

- Usage of UCS, see *UCS 5.0 Manual* [1].

To understand this documentation, you need to know the following concepts and tasks:

- Use and navigate in a remote shell on Debian GNU/Linux derivative Linux distributions like UCS. For more information, see Shell and Basic Commands[1] from *The Debian Administrator's Handbook*, Hertzog and Mas [2].
- Manage an app through Univention App Center[2] in *UCS 5.0 Manual* [1].

The app **UCS Intercom Service** supports *Keycloak* as *IdP*. In the context of this document, the term *IdP* is synonymous for *Keycloak* in the context of an *IdP* in OIDC (OpenID Connect).

Your feedback is welcome and highly appreciated. If you have comments, suggestions, or criticism, please send your feedback[3] for document improvement.

[1] https://www.debian.org/doc/manuals/debian-handbook/short-remedial-course.en.html#sect.shell-and-basic-commands
[2] https://docs.software-univention.de/manual/5.0/en/software/further-software.html#computers-softwareselection
[3] https://www.univention.com/feedback/?intercom-service-app=generic

CHAPTER

ONE

INSTALLATION

You can install the app **UCS Intercom Service** like any other app with Univention App Center.

UCS offers two different ways for app installation:

- With the web browser in the UCS management system
- With the command-line

For general information about Univention App Center and how to use it for software installation, see Univention App Center[4] in *UCS 5.0 Manual* [1].

1.1 Prerequisites

Installing this app has various prerequisites:

1. ICS supports OIDC. *Nextcloud (>=23.0)* and *OX App Suite (>=7.10.6)* must authenticate through OIDC, as well.
2. For working CSRF (Cross-Site-Request-Forgery) protection all other apps need to be up to date.
3. ICS requires the *Nordeck* bot up and running for Matrix.
4. ICS requires its three secrets before installation, see *Secrets* (page 5) for details.
5. ICS requires a configured *Keycloak* (>=12.0), including a valid Intercom OIDC client before installation.

1.2 Add ICS client to IdP

To prepare the existing *IdP* for the installation of the app **UCS Intercom Service**, use the following steps:

1. Enter the *Keycloak Admin Console*.
2. Create a OIDC Client. Recommendation is to use the default value `intercom` for the *Client ID* and leave the *Root URL* empty and save it.

   During app installation, **UCS Intercom Service** asks for the *Client ID*.
3. Set *Access Type* to `confidential`.
4. Set *Service Accounts Enabled* and *Authorization Enabled* to `On`.
5. Set *Backchannel Logout URL* to your intended domain for ICS with protocol and append the `backchannel-logout` path, for example:

   ```
   https://ics.example-domain.example-tld/backchannel-logout
   ```

   This step requires *Keycloak* >=12.0.0.

[4] https://docs.software-univention.de/manual/5.0/en/software/app-center.html#software-appcenter

6. Set the valid redirect URL to your intended ICS domain with protocol and append the `/callback` path, for example:

```
https://ics.example-domain.example-tld/callback
```

7. Go to the tab *Credentials*, copy the secret and save it to `/etc/intercom-client.secret`.
8. Go to the tab *Client Scopes* and add `offline_access` to *Assigned Default Client Scopes*.

1.3 Installation with the web browser

To install ICS from the UCS management system, use the following steps:

1. Use a web browser and sign in to the UCS management system.
2. Open the *App Center*.
3. Select or search for *Intercom Service* and open the app with a click.
4. To install *Intercom Service*, click *Install*.
5. Leave the *App settings* in their defaults or adjust them to your preferences. For a reference, see *Settings* (page 6).
6. To start the installation, click *Start Installation*.

Note: To install apps, the user account you choose for login to the UCS management system must have domain administration rights, for example the username `Administrator`. User accounts with domain administration rights belong to the user group `Domain Admins`.

For more information, see Delegated administration for UMC modules[5] in *UCS 5.0 Manual* [1].

1.4 Installation with command-line

To install the app **`UCS Intercom Service`** from the command-line, use the following steps:

1. Sign in to a terminal or remote shell with a username with administration rights, for example `root`.
2. Choose between default and custom settings and run the appropriate installation command.

   For installation with default settings, run:

```
$ univention-app install intercom-service
```

   To pass customized settings to the app during installation, run the following command:

```
$ univention-app install --set $SETTING_KEY=$SETTING_VALUE intercom-service
```

> **Caution:** Some settings don't allow changes after installation. To overwrite their default values, set them before the installation. For a reference, see *Settings* (page 6).

   Example: To define a different Keycloak-realm in ICS, run:

```
$ univention-app install intercom-service \
  --set intercom-service/keycloak/realm-name=master
```

[5] https://docs.software-univention.de/manual/5.0/en/central-management-umc/delegated-administration.html#delegated-administration

CHAPTER

TWO

CONFIGURATION

The app **UCS Intercom Service** offers various configuration options. Some settings don't allow changes after installation. Therefore, you must set them **before** installation. You find those settings marked with *Only before installation* in *Settings* (page 6). You can change all other settings at any time after the installation.

To change settings after installation, sign in to the UCS management system with a username with administration rights and go to *App Center* ‣ *UCS Intercom Service* ‣ *Manage Installation* ‣ *App Settings*. On the appearing *Configure UCS Intercom Service* page, you can change the settings and apply them to the app by clicking *Apply Changes*.

The App Center then *re-initializes* the Docker container for the app **UCS Intercom Service**. *Reinitialize* means the App Center throws away the running ICS Docker container and creates a fresh ICS Docker container with the just changed settings.

2.1 Intercom Service

The app **UCS Intercom Service** provides the back end for inter-app communication of *Nextcloud*, the *UCS Portal*, *UMC*, *Matrix* through the *Nordeck* bot and *OX App Suite*.

> **Warning:** This app doesn't configure any *Keycloak* settings. It requires an existing client and realm setup in *Keycloak*.

2.2 Secrets

The app **UCS Intercom Service** requires secrets, that aren't automatically generated. Those secrets are:

/etc/intercom-client.secret The client secret for authenticating with the *IdP*. You can retrieve the client secret from the *Keycloak Admin Console* in the *Authorization* tab of the *intercom-client*.

/etc/matrix.secret The secret for back end communication with the *Matrix* server. You can retrieve it from the automatic join app service on the system running *Matrix*.

The following command shows how to retrieve the secret for the back end communication with the *Matrix* server:

```
$ kubectl exec --stdin --tty synapse-0 -n matrix-000-prod -- \
    /bin/bash -c "cat /data/autojoin-appservice.yaml \
    | grep as_token \
    | sed -e 's/as_token. \(.\+\)/\1/'"
```

/etc/portal.secret The secret to communicate with the UCS Portal navigation service. You can retrieve the secret from `/etc/portal-navigation-service.secret`.

2.3 Settings

The following references show the available settings within the app **UCS Intercom Service**. Univention recommends to keep the default values.

intercom/settings/client-id

Defines the OIDC client name of ICS in *Keycloak*. The file `/etc/ics_client.secret` stores the secret of this client.

Required	Default value	Set
Yes	`intercom`	Only before installation

intercom/settings/intercom-url

Defines the URL where you can reach ICS. This needs to be a externally reachable address as it's used by the browser to connect to ICS.

Required	Default value	Set
Yes	`https://ics.@%@domainname@%@`	Only before installation

intercom/settings/base-url

Defines the base URL used to identify with the *IdP*. This URL must match the base URL defined in the OIDC client used on the *IdP*. The value should be the same as in *intercom/settings/intercom-url* (page 6).

Required	Default value	Set
Yes	`https://ics.@%@domainname@%@`	Only before installation

intercom/keycloak/url

URL of the *Keycloak* instance that ICS uses as *IdP*. ICS ignores this value, if *intercom/settings/issuer-base-url* (page 6) is defined.

Required	Default value	Set
Yes	`https://id.@%@domainname@%@`	Only before installation

intercom/keycloak/realm-name

Name of the realm containing the configured OIDC ICS client. ICS ignore this value, if *intercom/settings/issuer-base-url* (page 6) is defined.

Required	Default value	Set
Yes	`UCS`	Only before installation

intercom/settings/issuer-base-url

Defines a full base URL for the OIDC token issuer. Usually, the *IdP Keycloak* issues OIDC tokens.

This variable overwrites `intercom-service/keycloak/url` and `intercom-service/keycloak/realm-name`.

Only set this variable, if you really need to change the default URL generated from the before mentioned variables.

Required	Default value	Set
No	`None`	Only before installation

intercom/settings/origin-regex

Defines the origin *CORS* regular expression. Normally this will be the shared domain name. Changing this value may have security implications.

Required	Default value	Set
Yes	`@%@domainname@%@`	Only before installation

intercom-service/settings/proxy

This setting is passed to *node-axios* within the container. It allows or disallows connections through a proxy server between ICS and apps like *Matrix*, *Nextcloud*, or *OX App Suite*, instead of a direct connection to the back ends.

Required	Default value	Set
Yes	`False`	Only before installation

intercom/matrix/url

Defines the URL, where you can reach the *Matrix* server. The file `/etc/ics_matrix_as.secret` stores the Matrix secret.

Required	Default value	Set
Yes	`https://matrix.` `@%@domainname@%@`	Only before installation

intercom/matrix/server-name

Defines the server name of the *Matrix* server, that is a unique identifier configured in *Matrix*. The server name must match the configured server name in *Matrix*.

It isn't necessarily the server name defined in *intercom/matrix/url* (page 7).

Required	Default value	Set
Yes	`matrix.@%@domainname@%@`	Only before installation

intercom/matrix/login-type

Defines the login type that ICS uses for the *Matrix* server.

Refer to the Matrix[6] documentation for more information about login types.

Required	Default value	Set
Yes	`uk.half-shot.msc2778.login.` `application_service`	Only before installation

intercom/matrix/nordeck-mode

Defines the connection mode of the *Nordeck* bot.

Possible values: `test`, `live`, `test proxies`.

Required	Default value	Set
Yes	`test`	Only before installation

intercom/matrix/nordeck-url

Defines the URL, where you can reach the *Nordeck* bot.

[6] https://matrix.org/docs/

Required	Default value	Set
Yes	`https://meetings-widget-bot.@%@domainname@%@`	Only before installation

intercom/portal/portal-url

Defines the URL for the UCS portal. The file `/etc/ics_portal.secret` stores the Portal API key.

Required	Default value	Set
Yes	`@%@ucs/server/sso/fqdn@%@`	Only before installation

intercom/ox/ox-origin

Defines the *OX App Suite CORS* setting. Usually, this value is will be the same as the *OX App Suite* external address.

Required	Default value	Set
Yes	`https://webmail.@%@domainname@%@`	Only before installation

intercom/ox/audience

Defines the *OIDC audience* setting for *OX App Suite* that *OX App Suite* uses in the *IdP Keycloak*.

Required	Default value	Set
Yes	`oxoidc`	Only before installation

intercom/nextcloud/audience

Defines the *OIDC audience* setting for *Nextcloud* that *Nextcloud* uses in the *IdP Keycloak*.

Required	Default value	Set
Yes	`ncoidc`	Only before installation

intercom-service/nextcloud/url

Defines the URL where you can reach *Nextcloud*.

Required	Default value	Set
Yes	`https://fs.@%@domainname@%@`	Only before installation

intercom-service/nextcloud/origin

Defines the *Nextcloud CORS* setting. Usually this value is the same as *intercom-service/nextcloud/url* (page 8).

Required	Default value	Set
Yes	`https://fs.@%@domainname@%@`	Only before installation

CHAPTER

THREE

ARCHITECTURE

The **UCS Intercom Service** app architecture consists of the following elements:

- The operating environment UCS with the App Center and the Docker engine running the ICS container.
- The ICS software based on *Node.js®* running in the ICS container.
- The *Keycloak Identity Access Management*, used by ICS to authenticate sessions and retrieve login tokens for applications.
- A *Redis* container to store OIDC sessions.

Note: The following sections may refer to the *browser* rather than to the *client* to avoid confusion with OIDC clients configured in *Keycloak*.

Back end communication as referred to in the following section isn't related to the *Backchannel Logout* path, which is a specific part of the OIDC protocol.

3.1 Design decisions

The app **UCS Intercom Service** aims to provide a way to facilitate *CORS* conform communication to different back ends directly from the browser. It can proxy, modify, and authenticate requests and use *Keycloak* as *IdP* and its own session storage to hold OIDC sessions. It acquires those sessions through a silent background login, provided a valid OIDC cookie is already available in the browser.

3.2 Overview

Starting from the basics, ignoring everything related to login, authentication, and sessions for now, this is how ICS works on a basic level.

1. The browser opens the intended app.
2. The app contains ICS related JavaScript code as part of its responses.
3. This code instructs the browser to send a requests to ICS, after the browser needs communication to a separate app.
4. ICS acts as a middleware to modify and forward those requests appropriately to the relevant, second app using a back end communication channel.
5. ICS receives the response and finally sends an appropriately modified response to the browser.

For a visual representation, refer to Figure 3.1.

The following list describes how this fits into the wider OIDC authentication scheme. See also Figure 3.2.

1. The browser starts unauthenticated at the login endpoint of an app that supports ICS, for example *Matrix*, *Nextcloud*, or *OX App Suite*.

2. The browser follows the OIDC login procedure. The app redirects the browser to the *IdP Keycloak* and upon successful login assigns an OIDC session for the app to the browser.
3. The browser requests an action, for example to create a video conference, as part of a calendar entry. The browser requests an interaction from *OX App Suite* to *Matrix*. In detail, the browser requests the *Nordeck* bot that runs in the Matrix user front end *Element*.
4. A silent login happens in the background, that uses the information stored in the browser to authenticate ICS with the *IdP Keycloak* through a hidden IFrame[7].
5. The functional interaction begins as displayed in Figure 3.1.
6. ICS sends a requests to the back end, usually another app on UCS. ICS acts as a middleware between the browser and back ends, for example apps.

Note: ICS may use shared secrets rather than relying on OIDC authentication when communicating with back ends.

Warning: Back end communication is only safe, if done through HTTPS or a secure network. Otherwise, attackers may eavesdrop secrets on application layer.

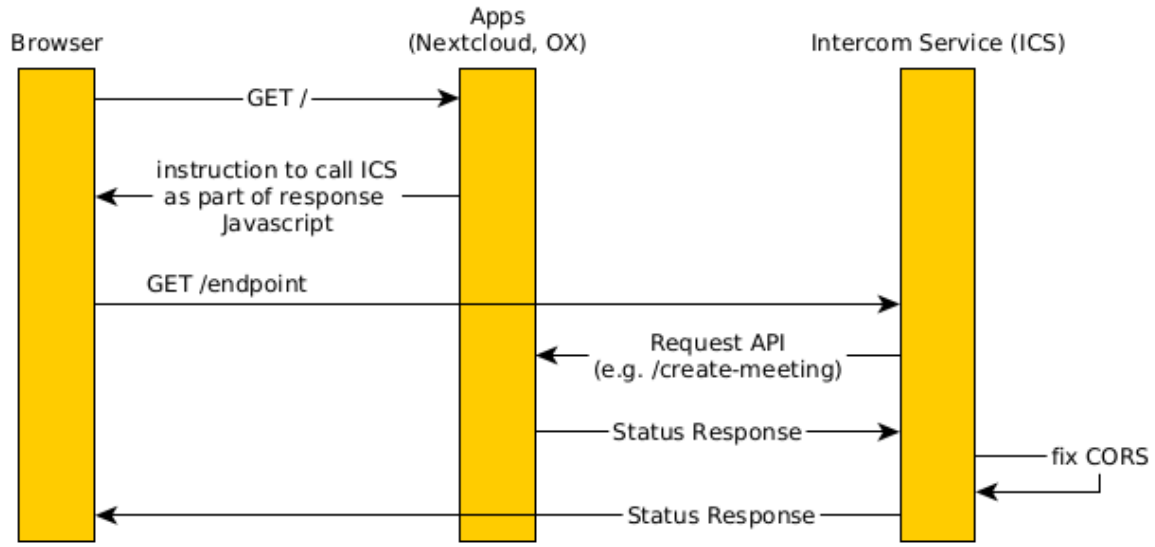

Figure 3.1: Interactions of ICS without OIDC

[7] https://en.wikipedia.org/wiki/HTML_element#Frames

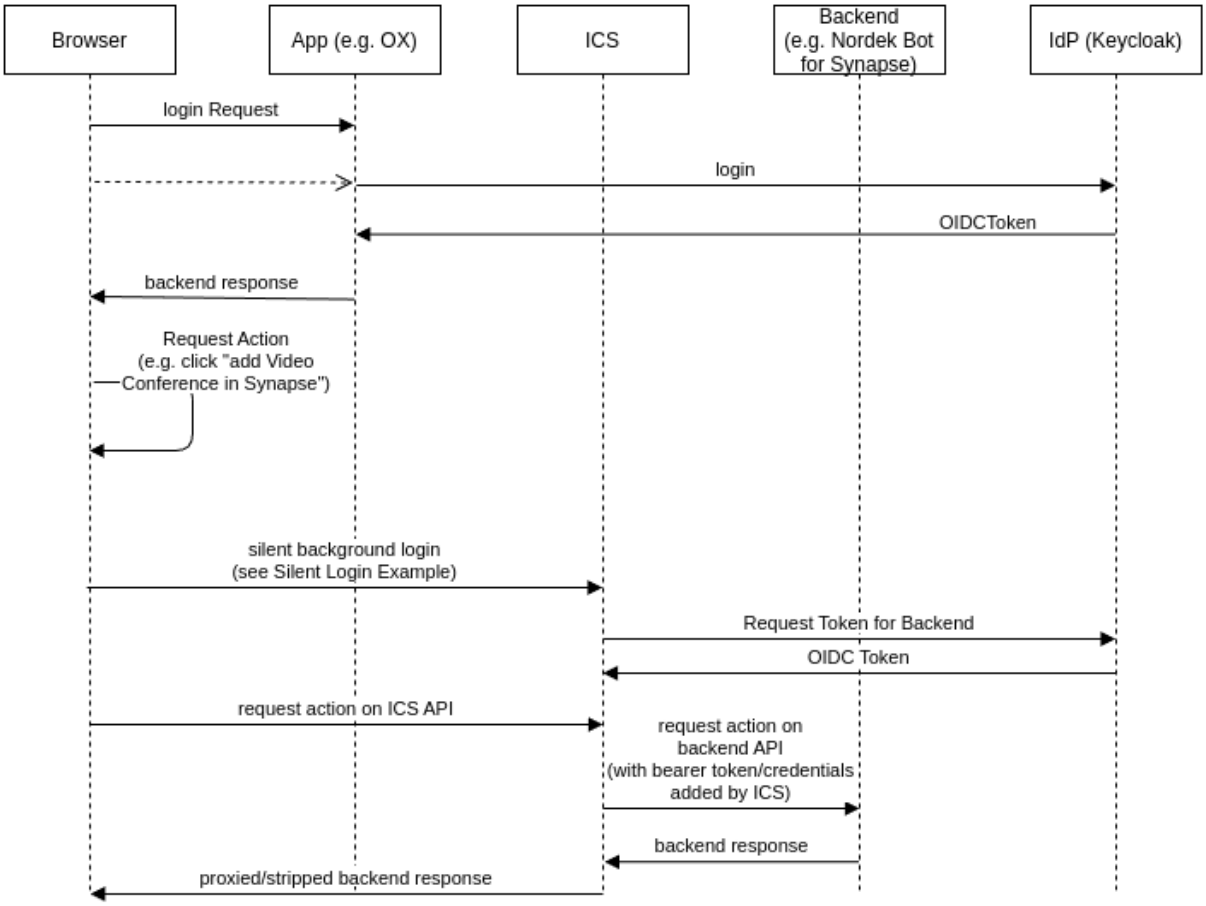

Figure 3.2: Interactions of ICS with OIDC, *OX App Suite* and *Nordeck*

3.3 Portal navigation

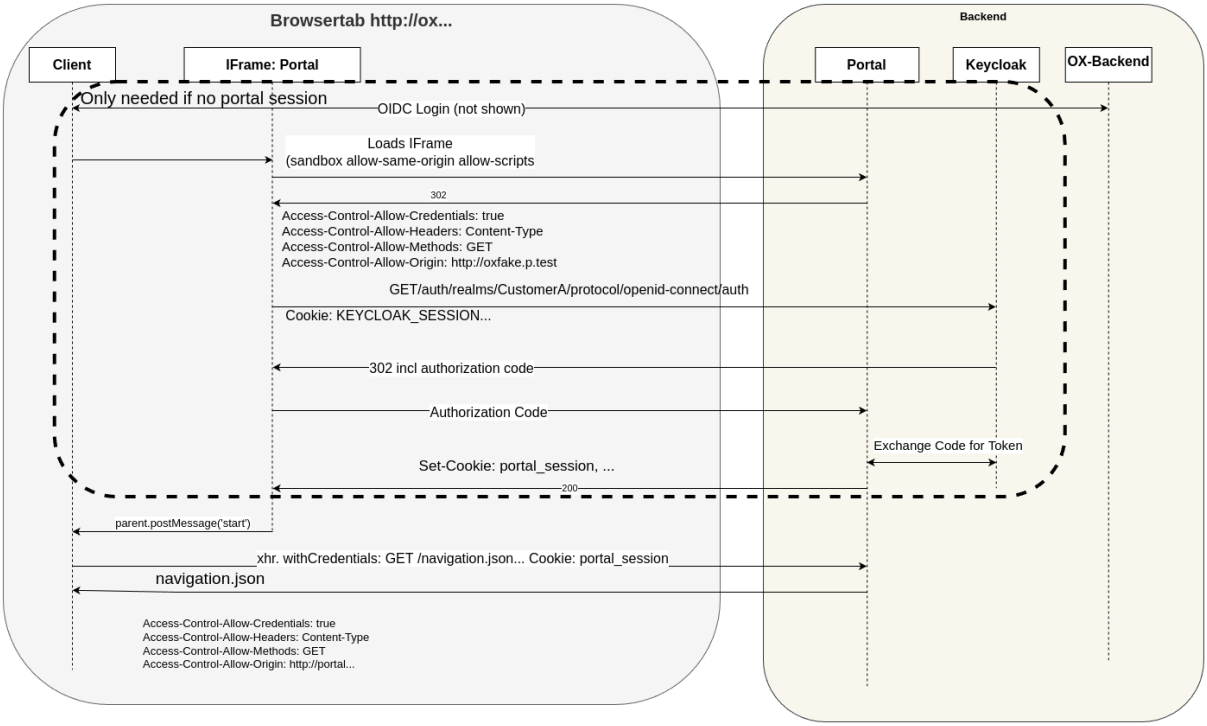

Figure 3.3: Communication overview for the `Central Navigation` capability, which requires cross app communication between *OX App Suite* and the *UCS Portal*.

3.4 Filepicker

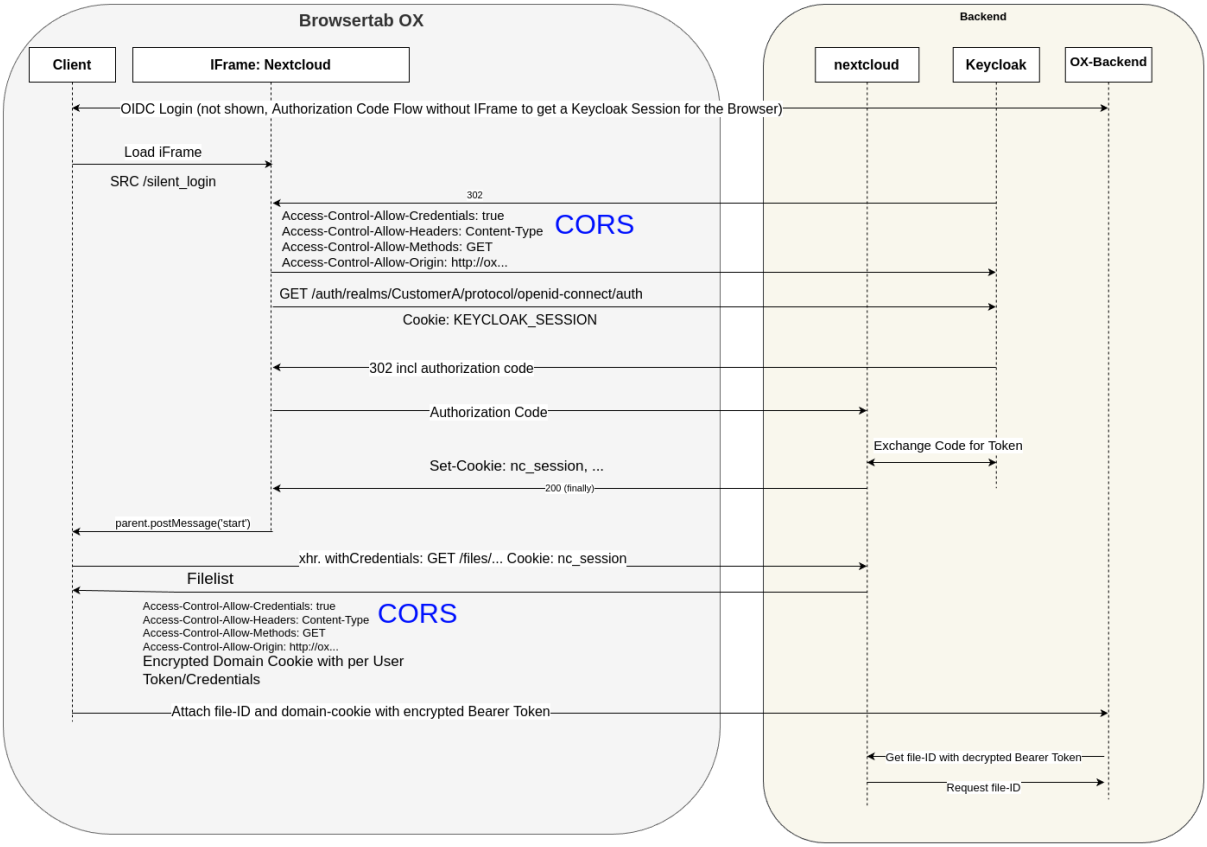

Figure 3.4: Communication overview for the `Filepicker` capability, which requires cross-app communication between *OX App Suite* and *Nextcloud*.

3.5 Endpoints

The app **UCS Intercom Service** offers the API endpoints listed below.

3.5.1 General

/ Alive test only

/silent Silent OIDC login endpoint

/backchannel-logout Endpoint for OIDC backchannel logout requests

3.5.2 App specific

/fs Proxy for Nextcloud

/navigation.json Proxy to Univention-portal for central navigation data

/nob Proxy for the *Nordeck* bot. In a testing environment, developers can use this endpoint to requests to the plain *Matrix* `UserInfo` service.

3.6 Terms

The document uses the terms that may not be clear to the reader. The following list provides context and explanation.

CORS CORS stands for *Cross-Origin Resource Sharing* and is a mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served.

For more information about CORS (Cross-Origin Resource Sharing), refer to Wikipedia: Cross-origin resource sharing[8].

IdP stands for *Identity Provider*. An IdP offers user authentication as service. In the context of the app **UCS Intercom Service** the software *Keycloak* offers the IdP service to ICS and its app back ends.

OIDC audience The OIDC audience is a required claim within the ID Token for all OAuth 2.0 flows used by OIDC. According to the specification, it must contain the OAuth 2.0 `client_id` of the relying party as audience value.

For more information, see section ID Token[9] in *OpenID Connect Core 1.0 incorporating errata set 1* [3].

[8] https://en.wikipedia.org/wiki/Cross-origin_resource_sharing
[9] https://openid.net/specs/openid-connect-core-1_0.html#IDToken

CHAPTER

FOUR

REQUIREMENTS AND LIMITATIONS

To ensure a smooth operation of the app **UCS Intercom Service** on UCS, administrators need to know the following requirements and limitations.

4.1 Cross-Site-Request-Forgery protection

Warning: CSRF protection wasn't extensively tested and may break at any time.

CSRF protection may not be working for *Matrix*, *Nextcloud*, and *OX App Suite* versions released before September 2022.

CHAPTER

FIVE

TROUBLESHOOTING

When you encounter problems with the operation of the app **UCS Intercom Service**, this section provides information where you can look closer into and to get an impression about what's going wrong.

5.1 Log files

The app **UCS Intercom Service** produces different logging information in different places.

/var/log/univention/appcenter.log Contains log information around activities in the App Center.

The App Center writes ICS relevant information to this file, when you run app lifecycle tasks like install, update and uninstall or when you change the app settings.

/var/log/univention/join.log Contains log information from join processes. When the App Center installs ICS, the app also joins the domain.

ICS Docker container The app uses a custom built node image. The App Center runs the container. You can view log information from the ICS Docker container with the following command:

```
$ univention-app logs intercom-service
```

/var/log/apache2/*.log Reverse proxy logs may contain relevant information for queried URLs by **UCS Intercom Service**, for example the status of middleware queries to other components.

Note: For externalized setups, like for example the *BMI-UX* setup, ICS proxies the queries through the external *HA-Proxy*. Therefore, you can find the proxy log files in `/var/log/haproxy.log` on the *HA-Proxy* server.

5.2 Common problems

Failing to provide the protocol, for example HTTP or HTTPS, for middleware relevant URLs like *intercom-service/nextcloud/url* (page 8), *intercom/portal/portal-url* (page 8), and *intercom/matrix/nordeck-url* (page 7) leads to an error during the request in the form of:

```
TypeError: Cannot read properties of null (reading 'split')
  at required (/app/node_modules/requires-port/index.js:13:23)
```

CHAPTER

SIX

CHANGELOG

This changelog documents all notable changes to the ICS app. Keep a Changelog[10] is the format and this project adheres to Semantic Versioning[11].

6.1 1.2

Released: 29. September 2022

6.1.1 Added

- Various debug logs

6.1.2 Changed

- Apply firewall rules during installation to make ICS accessible from outside of UCS.
- Set Docker DNS based on the *UCR* variables *nameserver1*, *nameserver2* and *nameserver3*.

6.1.3 Security

- The *Filepicker* functionality of ICS now fetches a separate token for authenticating with the file hosting application *Nextcloud*. The *OX* OIDC-client in the IdP must be allowed, to fetch a token for the *Nextcloud* OIDC-client. This was always intended, but not correctly enforced in earlier versions.

6.1.4 Fixed

- Update deprecated usage of *express.urlencoded*.
- ICS health check failed because of *Nordeck* URL returning *404*.
- Video conferences created as the wrong user.
- Central navigation returning *navigation.json* for the wrong user under certain circumstances.

[10] https://keepachangelog.com/en/1.0.0/
[11] https://semver.org/spec/v2.0.0.html

6.2 1.1

Released: 16. September 2022

6.2.1 Added

Stability

- ICS split the cookie headers by a logic that didn't consider certain cases. Now, ICS uses a standard cookie library for the handling cookie headers.
- During app installation, ICS tests the URLs of the required services **`Keycloak`**, *Nextcloud*, *Nordeck*, and *UCS Portal*, if it can reach them. The installation shows a warning, if the test can't reach the services. Additionally, ICS runs a health check within the Docker container every 60 seconds to test, if it can reach the services.

Refreshing Access Tokens A middleware that automatically refreshes access tokens when they expire.

6.2.2 Changed

- Improve the readability of user documentation.

6.2.3 Security

- The *Redis* database provides persistence for app sessions. The update applies the following security fixes to *Redis*:
  - Password protection provided in `/etc/intercom-redis.secret`.
  - The Redis container is only accessible from the **`docker-compose`** internal network (`external: false`).
- Verify the JWT (JSON Web Token) access or ID token with the public key of the *Keycloak* issuer.
- Enable `backchannel-logout` and remove the appropriated app-session from ICS.

6.2.4 Fixed

- Convert the uppercase value for the environment variable `PROXY` to lowercase. Using the variable in JavaScript requires the value in a lowercase string.

6.3 1.0

Released: 22. August 2022

6.3.1 Added

- Endpoint for OIDC silent login against **Keycloak** on `/silent`.
- Endpoint to securely proxy requests from *Open-Xchange* to *Nordeck* on `/nob`, allowing the creation of Element videoconferences from *Open-Xchange*.
- Endpoint to securely proxy requests from *Open-Xchange* to *Nextcloud* on `/fs`, allowing to use the email *Filepicker* with *Nextcloud*.
- Endpoint to securely proxy requests from *Open-Xchange* to *UCS Portal* `/navigation.json`, allowing for use of *UCS Portal* central navigation from *Open-Xchange*.
- Session storage with *Redis*.

BIBLIOGRAPHY

[1] *UCS 5.0 Manual*. Univention GmbH, 2021. URL: https://docs.software-univention.de/manual/5.0/en/.

[2] Raphaël Hertzog and Roland Mas. *The Debian Administrator's Handbook*, chapter Shell and Basic Commands. Freexian SARL, First edition, 2020. URL: https://www.debian.org/doc/manuals/debian-handbook/short-remedial-course.en.html#sect.shell-and-basic-commands.

[3] *OpenID Connect Core 1.0 incorporating errata set 1*. OpenID Foundation, 2014. URL: https://openid.net/specs/openid-connect-core-1_0.html.

INDEX