1. Unisync – Einleitung

Unisync ermöglicht die automatisierte Erstellung von Messenger-Chatgruppen, die den Kursräumen in ILIAS entsprechen. Die Leichtgewichtige, grundsätzlich auch an andere LMS anpassbare Lösung, entstand im Rahmen eines Teilprojekts des von der Stiftung Innovation in der Hochschullehre geförderten InduKo Projekts.

Eine Randbedingung die bei der Entwicklung beachtet werden musste, war die Tatsache, dass ein Systemzugriff auf Ilias nicht vorhanden war, warum es nicht möglich war, Daten über Systemschnittstellen zu übertragen. Ohne direkte Anbindung an ILIAS war der Einsatz eines Web-Scrapers die einzige Möglichkeit, um Kursdaten aus dem System auszulesen, um diese dann an das Matrix-System zu übertragen. Der große Vorteil ist, dass diese leichtgewichtige Lösung ohne tiefgreifende Systemintegration und Anpassungen an Ilias oder anderen Lernmanagementsystemen auskommt. Das erleichtert nicht nur die Adaption in andere Umgebungen und Lernmanagementsysteme, dadurch wird auch das Sicherheitsrisiko erheblich reduziert, da keine Eingriffe in bestehende Systeme erforderlich sind und deren Integrität sowie bewährte Sicherheitsmechanismen vollständig erhalten bleiben.

Unisync kommt völlig ohne zusätzliche App-Installation zur Kurs-Synchronisation aus. Ein Link, der beispielsweise in Ilias hinterlegt werden kann oder als Lesezeichen im Browser, ermöglicht es Lehrenden, ihre Ilias-Kurse automatisiert in Messenger-Chaträume zu überführen. Nach der Anmeldung am LMS werden die Kursdaten dann an das Matrix-System übertragen, wo automatisch Chatgruppen erstellt werden. Die Teilnehmerinnen und Teilnehmer der Kurse werden den Gruppen zugeordnet und Einladungslinks werden versendet. Die Chatgruppen sind anschließend in Matrix-basierten Messenger wie Element oder FluffyChat sichtbar, abhängig von den Präferenzen der Hochschule.

Diese Lösung ist flexibel, unabhängig von spezifischen Frontends oder Systemstrukturen, reduziert den Wartungsaufwand und erhöht die Sicherheit. Nutzerinnen und Nutzer müssen keine zusätzliche Software installieren und die Lösung kann an die Bedürfnisse unterschiedlicher Hochschulen angepasst werden. Unisync entstand im Rahmen eines Teilprojektes des von der Stiftung für Innovation in der Hochschullehre geförderten InduKo-Projekts (Innovation durch Kollaboration).

2. Unisync – Kernaspekte und Architektur

Unisync ist eine effiziente und flexible Synchronisationslösung zur automatisierten Erstellung von Matrix-Chat-Räumen für ILIAS-Kurse. Um diese Ziele zu erreichen, wurde eine Architektur entwickelt, bei der ein Matrix-Server als API-Endpunkt fungiert, der Raum-Erstellung, Synchronisation und Einladung von Nutzenden ermöglicht. Dieser Prozess eliminiert manuelle Eingriffe und stellt sicher, dass alle in Ilias-Kurse eingeschriebenen Studierenden in die entsprechenden Kommunikationsräume aufgenommen werden. OpenID Connect wird für die Authentifizierung eingesetzt, ergänzt durch eine Zwei-Faktor-Authentifizierung (2FA) für zusätzliche Sicherheit. Alle Kommunikationsinhalte werden mit der Ende-zu-Ende-Verschlüsselung von Matrix geschützt. Um die Nutzung so intuitiv wie möglich zu gestalten, wurde besonderer Wert auf eine reibungslose Benutzererfahrung gelegt. Eine

minimalistische, intuitive Oberfläche erleichtert den Anmeldeprozess und die Synchronisation der Kurse.

Die Architektur basiert auf drei wesentlichen Komponenten: dem Frontend, dem Backend und dem Matrix-Server. Diese arbeiten eng zusammen, um die automatische Erstellung und Verwaltung von Kommunikationsräumen zu ermöglichen, die den Kursräumen in ILIAS entsprechen. Der folgende Abschnitt beschreibt die Funktionsweise des Systems im Detail.

Das Frontend stellt die Schnittstelle für die Benutzer dar, um Anmeldedaten einzugeben und den Synchronisationsprozess zu überwachen. Es wurde mit Technologien wie HTML, CSS, JavaScript und Bootstrap entwickelt, um eine benutzerfreundliche und responsive Oberfläche zu gewährleisten.

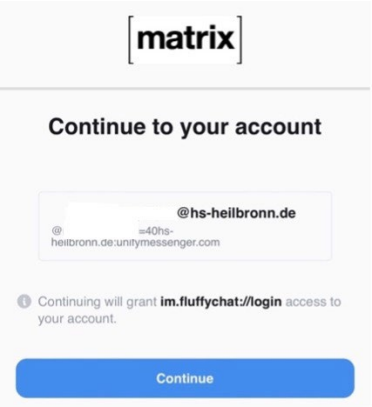
Die Nutzerinnen und Nutzer beginnen den Prozess, indem sie ihre ILIAS-Zugangsdaten in die Eingabemaske des Unisync-Systems eingeben. Falls erforderlich, unterstützt das System zusätzlich die Zwei-Faktor-Authentifizierung (2FA), um die Sicherheit zu erhöhen. Über die Benutzeroberfläche können sie außerdem den Fortschritt der Synchronisation in Echtzeit verfolgen und Details zu den Matrix-Räumen, wie Raum-ID und Teilnehmerlisten, einsehen.

Die Kommunikation zwischen dem Frontend und dem Backend erfolgt über AJAX (Asynchronous JavaScript and XML). Diese Technologie ermöglicht es, Daten in Echtzeit zu aktualisieren, ohne dass eine vollständige Seitenaktualisierung erforderlich ist. Alle Daten, die zwischen dem Frontend und dem Backend ausgetauscht werden, sind zur Sicherung der Vertraulichkeit der Datenübertragung HTTPS verschlüsselt.

Das Backend bildet die zentrale Komponente für die Datenverarbeitung und Synchronisation. Es wurde mit FastAPI entwickelt, einem leistungsstarken und modernen Webframework für Python, das eine schnelle und effiziente API-Kommunikation ermöglicht. Zu den Hauptaufgaben des Backends gehört zunächst die Authentifizierung der Nutzer am ILIAS-System. Dieser Prozess basiert auf dem OpenID-Connect-Protokoll (OIDC).

Nach der erfolgreichen Authentifizierung wird ein automatisierter Web-Scraper gestartet, der mithilfe der Technologien Playwright und BeautifulSoup4 arbeitet. Dieser Scraper navigiert eigenständig durch das ILIAS-System und ruft dabei die Liste der zugeordneten Kurse und Teilnehmer ab. Alle relevanten Daten, einschließlich der Kurstitel, Beschreibungen und Teilnehmerlisten, werden extrahiert. Diese Informationen sind entscheidend für die spätere Synchronisation mit dem Matrix-System.

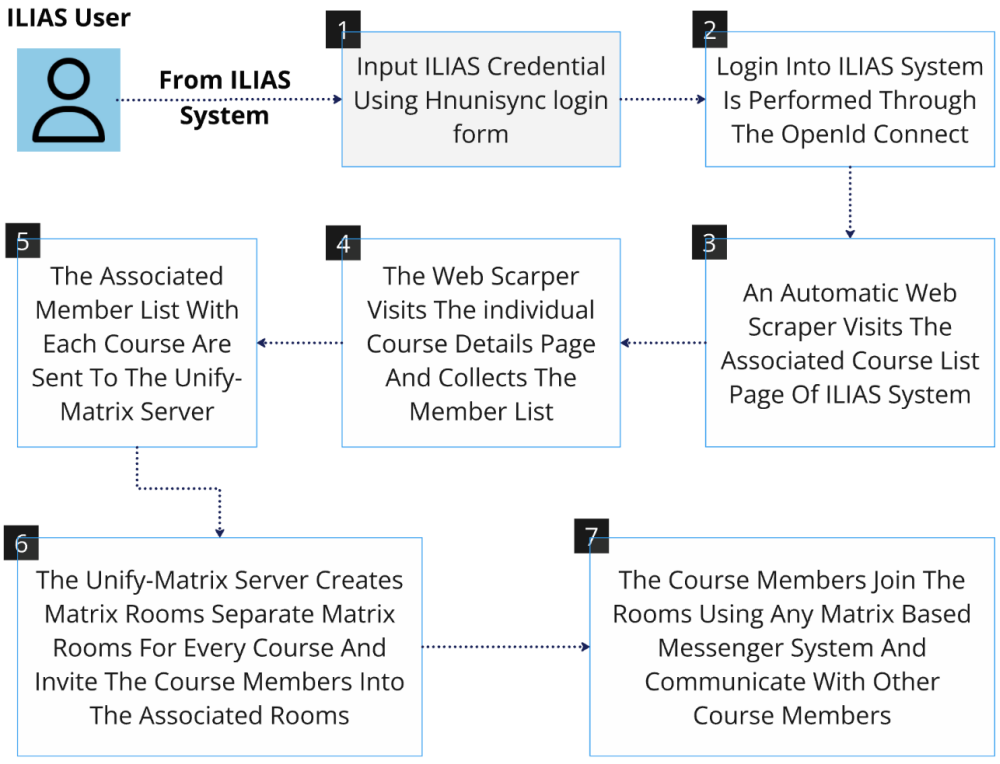
Die extrahierten Daten werden im Backend analysiert und aufbereitet. Damit die Nutzerzuordnung reibungslos funktioniert, ist es notwendig, dass die Nutzeridentitäten in ILIAS und im Matrix-Server eindeutig zueinander passen, was entweder dadurch gewährleistet werden kann, dass die Nutzer im Matrix-Server so angelegt sind, dass sie mit den ILIAS-Nutzern übereinstimmen oder durch eine Übersetzung im Backend: D.h. sollte es Unterschiede in den Konventionen der beiden Systeme geben, kann das Backend die ILIAS-Nutzerinformationen in die entsprechenden Matrix-Nutzer übersetzen, da die Namensgebungskonvention bekannt ist, nach denen die Nutzer im Matrix-Server über OpenID angelegt wurden.



Dieser Screen zeigt den Matrix User, wie er durch die Verbindung zu OpenID angelegt wurde. Die User müssten entweder identisch sein oder es müsste, wie in diesem Fall, ein entsprechendes Mapping vorgenommen werden.

Dieses Mapping ist erforderlich, wenn eine direkte Anbindung bzw. Integration mit Systemzugriff an der Hochschule aus Sicherheitsgründen nicht möglich ist. Das Backend übernimmt im Anschluss die Kommunikation mit dem Matrix-Synapse-Server. Hierbei wird die Matrix-Nio-Bibliothek genutzt, die es ermöglicht, Anweisungen zur Erstellung neuer Chat-Räume zu übermitteln und die Teilnehmer den entsprechenden Räumen zuzuweisen. Dabei wird sichergestellt, dass nur korrekt zugeordnete Nutzer in die Matrix-Räume aufgenommen werden können. Diese Vorgehensweise gewährleistet eine funktionierende Lösung innerhalb der gegebenen Restriktionen.

Hnunisync and Unify-Matrix Process flow



Der Matrix-Synapse-Server fungiert als zentrale Plattform für die Verwaltung der Kommunikationsräume; zu seinen zentralen Aufgaben gehört die Erstellung von Chat-Räumen für jeden Kurs in ILIAS. Dabei spiegelt die Struktur der erstellten Räume die Kursorganisation in ILIAS wider, sodass eine klare Zuordnung zwischen Kursen und Kommunikationsräumen gewährleistet ist.

Ein weiterer wichtiger Aspekt ist die automatische Verwaltung der Teilnehmer. Basierend auf den vom Backend bereitgestellten Daten lädt der Synapse-Server die entsprechenden Teilnehmer in die zugehörigen Chat-Räume ein. Dieser Prozess wird durch die API-gestützten Funktionen von Matrix ermöglicht und vollständig vom Backend initiiert. Der Synapse-Server stellt eine sichere Kommunikation durch die Verwendung von Ende-zu-Ende-Verschlüsselung sicher. Diese Technologie schützt alle Nachrichten und Daten vor unbefugtem Zugriff und gewährleistet die Vertraulichkeit und Integrität der Kommunikation innerhalb der Räume. Somit bleibt der Austausch zwischen Lehrenden und Studierenden vollständig privat und sicher.

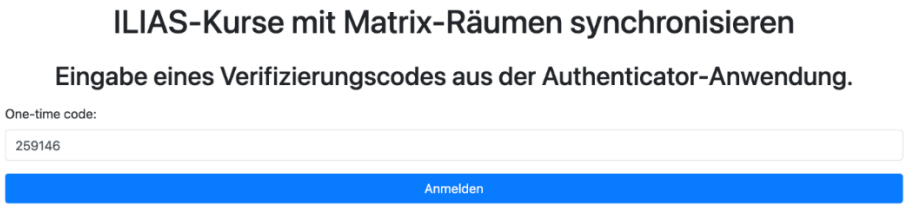
Nach der Erstellung der Räume erhalten die Nutzerinnen und Nutzer Einladungen, die es ihnen ermöglichen, den Kommunikationsräumen beizutreten. Diese Räume können mit jeder beliebigen Matrix-basierten Messenger-Anwendung genutzt werden, darunter Element, FluffyChat und andere.

Unisync – User – Flow: Vom Login zur Synchronisation und Kommunikation

Unisync bietet einen sicheren Login-Prozess, der auf den Hochschul-Zugangsdaten basiert. Die Benutzer, die diese Synchronisation durchführen, sind Lehrpersonen. Der Authentifizierungsprozess beginnt damit, dass die Nutzerinnen und Nutzer ihre Hochschul-Zugangsdaten eingeben. Diese Eingabe erfolgt in der Benutzeroberfläche von Unisync, die in der folgenden Abbildung zu sehen ist.

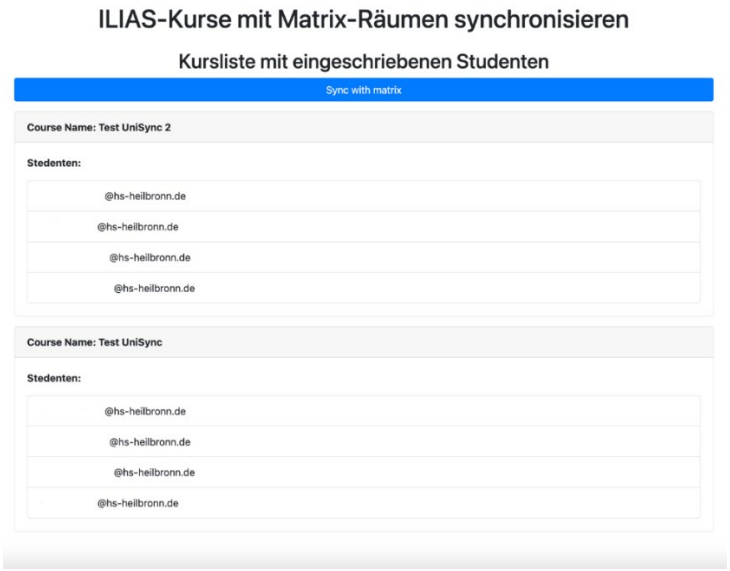
Nach Eingabe der Zugangsdaten wird die Identität der Nutzerinnen und Nutzer über den OpenID Connect-Authentifizierungsablauf verifiziert, wodurch sichergestellt wird, dass nur autorisierte Personen Zugriff auf die Plattform erhalten. Nachdem die Nutzerinnen und Nutzer ihre HHN-Zugangsdaten eingegeben haben, wird ein zusätzlicher Schritt erforderlich: die Eingabe eines einmalig gültigen Verifizierungscodes (OTP). Dieser Code wird von einer Authentifizierungs-App wie Google Authenticator generiert.

Die folgende Abbildung zeigt die Benutzeroberfläche, in der der einmalige Verifizierungscode eingegeben wird.

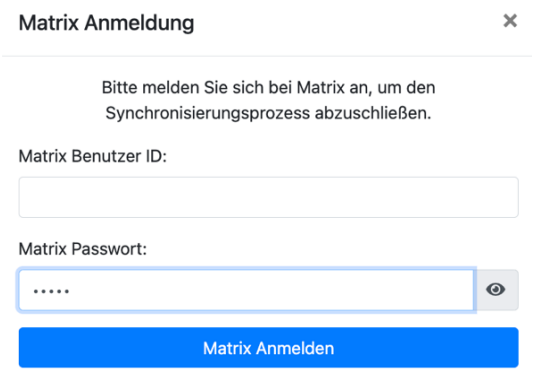


Nach erfolgreicher Authentifizierung verwendet Unisync den Web-Scraping-Mechanismus, um automatisch Informationen über die im ILIAS-System eingeschriebenen Kurse und die zugehörigen Teilnehmer zu sammeln. Dieser automatisierte Prozess extrahiert Details wie die Kursnamen sowie die vollständigen Teilnehmerlisten.

Die gesammelten Informationen werden in einer übersichtlichen Benutzeroberfläche präsentiert, die es den Nutzern ermöglicht, ihre Kurse mit den entsprechenden Matrix-Räumen zu synchronisieren. Dies geschieht durch einen einfachen Klick auf die Schaltfläche „Sync with Matrix“.



Um den Synchronisationsprozess abzuschließen, melden sich die Nutzerinnen und Nutzer mit ihrer Matrix-Benutzer-ID und ihrem Passwort am Unify-Matrix-Server an. Dieser sichere Login gewährleistet, dass Unisync eine Verbindung zu Matrix herstellen und die Studierenden korrekt in die entsprechenden Chat-Räume hinzufügen kann.



Unisync erstellt nun automatisch Matrix-Räume, die nach den jeweiligen Kursen benannt sind und lädt alle eingeschriebenen Studierenden ein, den entsprechenden Räumen beizutreten. Eine Bestätigungsseite gibt den Lehrpersonen eine Übersicht über die durchgeführte Synchronisation. Diese umfasst unter anderem die Matrix-Raum-ID sowie eine Liste der erfolgreich hinzugefügten Studierenden.

Die Ansicht für den Nutzer bzw. die Nutzerinnen wird doch die Auswahl des jeweils bevorzugten Matrix-Messengers bestimmt. Hier bspw. ein Screenshot des open Source Messengers FluffyChat mit dem entsprechenden transparenten Einladungshinweis; nach Bestätigung durch den User erfolgt der automatische Kursbeitritt

Kurzbeschreibung des verwendeten Technologie-Stacks

Der genutzte Stack gliedert sich in zwei Hauptkomponenten: die Webanwendung und den Matrix-Server.

Die Unisync-Webanwendung verwendet mehrere Frameworks und Bibliotheken, um Benutzer-Authentifizierung, Kurs-Synchronisation, Web-Scraping und die Kommunikation mit dem Matrix-Server umzusetzen. Zu den zentralen Technologien gehören:

- Python FastAPI wird für die Erstellung von APIs eingesetzt.
- Uvicorn Dient als ASGI-Server für die Ausführung von FastAPI-Anwendungen.
- Pydantic übernimmt die Validierung und Verarbeitung von Daten.
- Jinja2 wird für die dynamische Erstellung von HTML-Inhalten mittels Templates genutzt.
- Gunicorn ermöglicht die effiziente Ausführung von Python-Webanwendungen.

Für die Automatisierung von Web-Scraping-Prozessen kommen die folgenden Werkzeuge zum Einsatz:

- Playwright automatisiert Browser-Interaktionen, um Kursdaten aus dem ILIAS-System abzurufen.

- BeautifulSoup4 analysiert HTML-Dokumente und extrahiert strukturierte Informationen aus dem ILIAS-System.

Die Interaktion mit APIs und die Unterstützung asynchroner Abläufe wird durch folgende Technologien ermöglicht:

- Matrix-Nio, diese Client-Bibliothek dient der Interaktion mit dem Matrix-Protokoll und unterstützt die Verwaltung von Kommunikationsräumen und Teilnehmern.
- Quart, ein Framework, das asynchrone Prozesse ermöglicht und mit Python asyncio kompatibel ist.
- Requests und Httpx dienen der Durchführung von HTTP-Anfragen und der Kommunikation mit APIs.

Für die Interaktion mit dem Matrix-Protokoll kommen zum Einsatz:

- Matrix-Nio, diese Client-Bibliothek ermöglicht die Interaktion mit Matrix-Servern und Funktionen wie die Erstellung von Räumen und die Verwaltung von Teilnehmern.
- Matrix-Synapse ist der verwendete Matrix-Homeserver, der die Verwaltung und das Hosting von Matrix-Räumen übernimmt.

Auf der Backend-Seite werden folgende Technologien verwendet:

- Quart ermöglicht die Verarbeitung von asynchronen Backend-Prozessen.
- Request dient der Durchführung von HTTP-Anfragen an externe Dienste.

Login-Prozess: Aktueller Workaround und Lösungsmöglichkeit

Nach einem Serverwechsel hatte unser neu installierter Matrix-Server, incl. neuer Domain, gegenüber der vorhergehenden Version, zuletzt keinen Zugang zu den OpenID Connect-APIs unserer Hochschule mehr. Dadurch ist es nicht möglich den standardmäßigen, tokenbasierten Authentifizierungsablauf zu nutzen.

Um dieses Problem zu umgehen, verwendet das Backend das Automatisierungstool Playwright. Dieses simuliert Benutzerinteraktionen und übernimmt die Eingabe der Zugangsdaten und des einmaligen Codes (OTP) im Namen der Nutzer. Dabei navigiert es durch den vom Identity Provider bereitgestellten Login-Prozess. Obwohl diese Lösung funktional ist, weicht sie von den Standards des OIDC-Protokolls ab und bringt darum Sicherheits- und Compliance-Risiken mit sich. Der fehlende direkte Zugriff auf die Hochschul-OpenID Connect API erhöht außerdem die Komplexität des Systems. Die Wartung der Automatisierungsskripte erweist sich als aufwendig und fehleranfällig, da Änderungen am Login-Workflow des Identity Providers die Automatisierung unterbrechen können.

Dies erfordert eine kontinuierliche Überwachung und regelmäßige Updates, um die Funktionalität aufrechtzuerhalten.

Es handelt sich hier um einen Workaround, da der direkte Zugriff auf die von der Hochschule bereitgestellte OpenID Connect API nach dem Serverwechsel nicht mehr verfügbar gemacht wurde. Dieser Ansatz wird jedoch nicht als Best-Practice-Lösung empfohlen. Stattdessen sollte die standardisierte Nutzung von OpenID Connect gewählt werden. Der Vollständigkeit halber werden dennoch im Folgenden die Schwachstellen des gewählten Workarounds für den Login-Prozess kurz beschrieben.

Die direkte Eingabe von Benutzername, Passwort und OTP in die Anmeldemaske von Unisync führt zu potenziellen Sicherheitsrisiken wie unbefugtem Zugriff, Man-in-the-Middle-Angriffen und einer potenziellen Exposition sensibler Daten im Speicher. Zudem fehlt die durch OIDC bereitgestellte Ende-zu-Ende-Verschlüsselung, wodurch Benutzerdaten bei einem Sicherheitsvorfall im Backend gefährdet sein könnten. Die Umgehung der Einwilligungsseite des IdP beeinträchtigt Transparenz und Zustimmung. Ohne ID- oder Zugriffstoken fehlen wichtige Sicherheitsfunktionen wie Token-Überprüfung, Ablauf sowie Widerruf. Nutzer werden nicht ausreichend informiert, dass ihre Anmeldedaten direkt von Unisync verarbeitet werden. Die Abhängigkeit von Playwright erhöht den Wartungsaufwand, da Änderungen an der Login-Seite des IdP häufige Updates der Automatisierungsskripte erfordern. Ressourcenintensive Browserautomatisierung belastet die Systemleistung und kann bei Netzwerkverzögerungen oder durch Sicherheitsmaßnahmen wie Captchas scheitern, was den Workflow unterbricht.

Um die beschriebenen Schwachstellen des Workarounds zu vermeiden wird empfohlen, Unisync direkt mit dem OIDC-Provider der jeweiligen Hochschule zu integrieren. Das umfasst die Konfiguration einer Client-ID, eines Client-Secrets sowie sicherer Redirect-URIs. Diese direkte Integration ermöglicht die Nutzung der OIDC-Standards und vereinfacht die Authentifizierungsabläufe und die Sicherheit wird erhöht. Nutzer werden künftig auf die Login-Seite des IdP umgeleitet und nach der erfolgreichen Authentifizierung stellt der IdP ein ID-Token für die Identitätsprüfung und ein Access-Token zur Autorisierung von Datenzugriffen bereit. Die Authentifizierung wird durch die Umleitung zur IdP-Login-Seite und die Validierung der bereitgestellten Token in Unisync umgesetzt. Die Benutzeroberfläche bleibt dabei unverändert, während tokenbasierte Workflows die Sicherheit und Zuverlässigkeit steigern.

Für die OIDC-Integration sind Änderungen an der Serverkonfiguration erforderlich. Die CORS-Einstellungen müssen angepasst werden, um Anfragen des IdP zuzulassen, Firewall-Regeln müssen so aktualisiert werden, dass eine sichere Kommunikation ermöglicht wird. Zudem muss die Unisync-Domain in den Einstellungen des IdP als vertrauenswürdig registriert werden.

Durch die Integration eines standardisierten OIDC-Workflows werden Sicherheitsrisiken beseitigt, die Einhaltung von Compliance-Standards sichergestellt, die Systemkomplexität reduziert und Stabilität sowie Skalierbarkeit des Systems werden verbessert.

Umfassende Tests über das Entwicklungsteam hinaus sowie ein Feldtest wurden während des Projekts nicht durchgeführt.