Lufthansa Systems

DHBW
Duale Hochschule
Baden-Württemberg
Mannheim

Duale Hochschule Baden-Württemberg
Mannheim

Bachelorarbeit

Analyse und Integration einer passwortlosen Authentifizierung im Unternehmenskontext

Studiengang Cyber Security

Verfasser:	Luka Tsipitsoudis
Matrikelnummer:	4110112
Kurs:	TINF20CS1
Bearbeitungszeitraum:	06.06.2023 – 29.08.2023
Abgabedatum:	29.08.2023
Betreuer:	Stefan Köster

Ehrenwörtliche Erklärung

Ich versichere hiermit, dass ich meine Projektarbeit mit dem Thema: „*Analyse und Integration einer passwortlosen Authentifizierung im Unternehmenskontext*“ selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort, Datum

Abstract

Deutsch

Inhaltsverzeichnis

Abbildungsverzeichnis **v**

Tabellenverzeichnis **vi**

Abkürzungsverzeichnis **vii**

1 Einführung **10**
1.1 Problemstellung & Ziel der Arbeit 10
1.2 Aufbau der Arbeit 10
1.3 Referenzierte Arbeiten 10

2 Grundlagen **11**
2.1 Einführung in cFront 11
2.2 CIA-Triade 11
2.3 Arten der Authentifizierung 13
2.4 Passwortbasierte Authentifizierung 14
2.4.1 Speicherung 16
2.4.2 Faktor Mensch 17
2.5 Passwortlose Authentifizierung 18
2.5.1 Magic Link 22
2.5.2 One Time Password (OTP) 22
2.6 YubiKey 22
2.6.1 Usability 22
2.7 Fido2 24
2.7.1 Webauthn 27
2.7.2 CTAP2 29
2.7.3 Sicherheit 31

3 Umsetzung **33**
3.1 Aktueller Stand der LSY 33
3.2 Integration eines Yubikeys in die LSY 33
3.3 User Feedback 37
3.3.1 Rahmen des Feedbacks 37
3.3.2 Auswahl der Teilnehmer 37
3.3.3 Inhalt der Demonstration 38

3.4 Wirtschaftlichkeit 40
3.5 Nutzung des passwortlosen Verfahrens im privaten Kontext 40

Literaturverzeichnis **41**

Abbildungsverzeichnis

Abbildung 2.1 Umsetzungsmöglichkeit mit Keycloak 11
Abbildung 2.2 Umsetzungsmöglichkeit mit Keycloak 12
Abbildung 2.3 Umsetzungsmöglichkeit mit Keycloak 12
Abbildung 2.4 Umsetzungsmöglichkeit mit Keycloak 13
Abbildung 2.5 Umsetzungsmöglichkeit mit Keycloak 14
Abbildung 2.6 Umsetzungsmöglichkeit mit Keycloak 15
Abbildung 2.7 Umsetzungsmöglichkeit mit Keycloak 19
Abbildung 2.8 Umsetzungsmöglichkeit mit Keycloak 22
Abbildung 2.9 Umsetzungsmöglichkeit mit Keycloak 25
Abbildung 2.10 Umsetzungsmöglichkeit mit Keycloak 26
Abbildung 2.11 Umsetzungsmöglichkeit mit Keycloak 27

Abbildung 3.1 Aktuelle Umsetzung der Abteilung 33
Abbildung 3.2 Umsetzungsmöglichkeit mit Azure Active Directory (AD) 33
Abbildung 3.3 Umsetzungsmöglichkeit mit Keycloak 34
Abbildung 3.4 Veränderter Keycloak-Login 34
Abbildung 3.5 Authentication Flow 35
Abbildung 3.6 Registrierung (vereinfacht) 36
Abbildung 3.7 Anmeldung (vereinfacht) 36
Abbildung 3.8 Veränderter Keycloak-Login 38
Abbildung 3.9 Veränderter Keycloak-Login 38
Abbildung 3.10 Veränderter Keycloak-Login 39
Abbildung 3.11 Veränderter Keycloak-Login 39

Tabellenverzeichnis

Abkürzungsverzeichnis

LSY	Lufthansa Systems GmbH & Co. KG
FIDO	Fast Identity Online
W3C	World Wide Web Consortium
SFA	Single-Factor Authentication
MFA	Multi-Factor Authentication
CTAP2	Client-to-Authenticator Protocol 2
ITU-T	International Telecommunication Union Telecommunication Standardization Sector
MITM	Man In The Middle
PAKE	Password Authenticated Key Exchange
EUF-CMA	Existential Unforgeability under a Chosen Message Attack
PQ	Post-Quantum
KEM	Key Encapsulation Mechanism
TLS	Transport Layer Security
puvProtocol	PIN/UV Auth Protocol
SUF	Strongly Unforgeable
UF	Unforgeable
OTP	One-Time Password
HOTP	HMAC-based One-Time Password
TOTP	Time-based One-Time Password
HMAC	Hash-based Message Authentication Code
CDS	???
AD	Active Directory

1 Einführung

Diese Arbeit beschäftigt sich mit passwortlosen Authentifizierungsverfahren. Im Folgenden werden zunächst die Problemstellung und das Ziel der Arbeit erläutert. Anschließend wird der Aufbau der Arbeit beschrieben und auf verwandte Arbeiten eingegangen:

1.1 Problemstellung & Ziel der Arbeit

Die Problemstellung dieser Arbeit bezieht sich auf den aktuellen, passwortlosen Ansatz der Authentifizierung im Unternehmenskontext der Lufthansa Systems GmbH & Co. KG (LSY). Trotz ihrer hohen Etablierung und Verbreitung bieten passwortlose Authentifizierungsverfahren nicht nur Vorteile, sondern auch eine hohe Anzahl an Angriffsvektoren.

Ziel dieser Arbeit ist es daher passwortlose Authentifizierungsverfahren genauer zu betrachten. Verschiedene passwortlose Verfahren werden vorgestellt und ihre Vor- und Nachteile aufgezeigt. Dabei soll ein besonderes Augenmerk auf den Vergleich der Angriffsvektoren von passwortlosen und passwortbasierten Verfahren gelegt werden. Einer der passwortlosen Verfahren wird begründet ausgewählt und detailiierter betrachtet. Dabei wird analysiert, ob das Verafahren für die LSY geeignet ist und welche Anpassungen vorgenommen werden müssen. Betrachtet werden insbesondere die Aspekte der Sicherheit und der Benutzerfreundlichkeit. Der Fokus liegt auf der Frage, ob passwortlose Verfahren eine Alternative darstellen, welche Passwörter gänzlich ersetzen.

1.2 Aufbau der Arbeit

1.3 Referenzierte Arbeiten

2 Grundlagen

2.1 Einführung in cFront

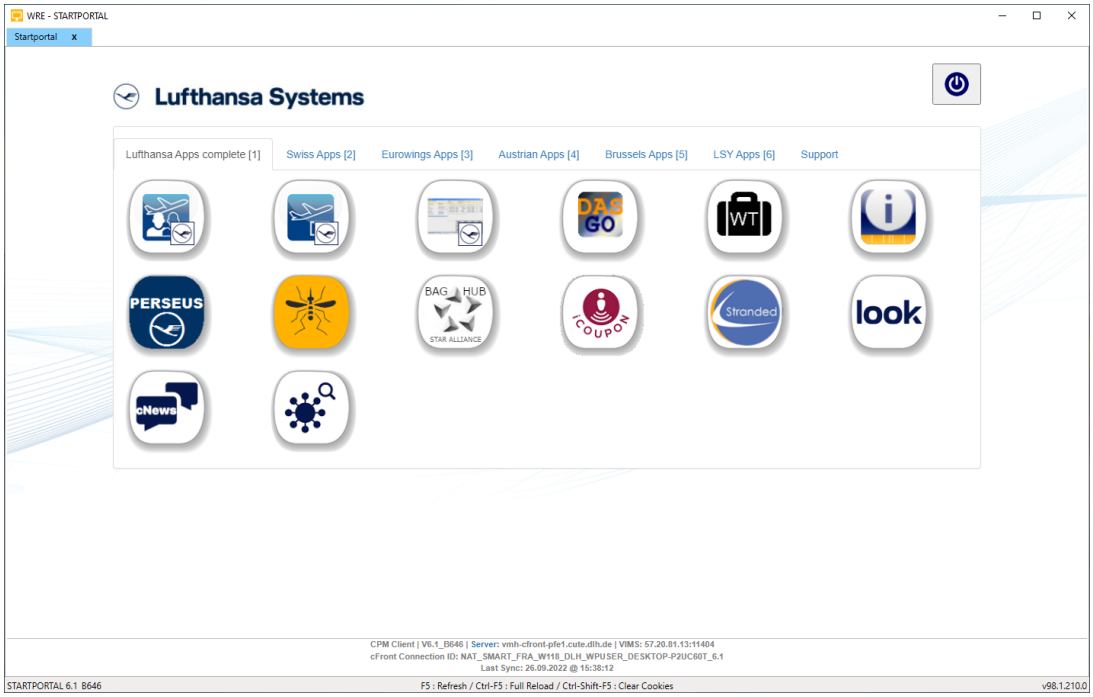

Abbildung 2.1: Umsetzungsmöglichkeit mit Keycloak

2.2 CIA-Triade

- Vertraulichkeit gehört zu den wichtigsten Schutzzielen in der Informationssicherheit [1].
- Kommt vom lateinischen Wort *confidere* [1].
- besagt, dass Informationen und Daten geschützt werden müssen, sodass diese nur von autorisierten Personen und für autorisierte Zwecke genutzt werden können [1].

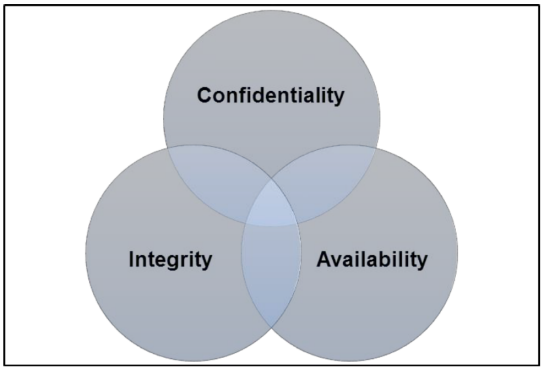

Abbildung 2.2: Umsetzungsmöglichkeit mit Keycloak

Additional Tenets	Relation to CIA triad
Authenticity	Integrity
Non-repudiation	Integrity
Correctness in specification	Integrity and Availability
Responsibility	Integrity
Integrity of people	Integrity
Trust	Confidentiality and Integrity
Ethicality	Integrity
Identity management	Confidentiality, Integrity and Availability

Abbildung 2.3: Umsetzungsmöglichkeit mit Keycloak

- Einschränkungen des Zugriffs auf Informationen und Daten, um die Privatssphäre und persönliches Eigentum zu schützen [1].
- Da der Fokus immer mehr auf wirtschaftliche Aspekte liegt, hat die Vertraulichkeit im Vergleich zu früher an Bedeutung verloren [1].
- Wird beispielsweise geschützt durch Verschlüsselung, Authentifizierung oder Sicherheitsprotokolle [2].
- kommt vom lateinischen *tangere*, was so viel wie berühren bedeutet. Mit der Vorsilbe *In-* wird daraus etwas, was so viel bedeutet wie *unberührbar* [1].
- Beinhaltet die Garantie, dass Daten nicht verändert werden können, ohne dass dies bemerkt wird [2].

- Beispielsweis, dass eine empfangene Nachricht genau so ankommt, wie sie gesendet wurde [2].
- Wird beispielsweise geschützt durch Firewall, Intrusion Detection Systeme oder digitale Signaturen [2].
- kommt vom lateinischen *valere* was so viel bedeutet wie *stark sein* [1].
- In der Informationssicherheit bezieht sich die Zuverlässigkeit auf einen zeitnahen und zuverlässigen Zugriff auf Informationen und Daten [1].
- Das bedeutet dass der Zugriff möglichst ohne unterbrechungen und unabhängig vom Standort erreichbar werden kann [2].
- Verfügbarkeit kann beispielsweise durch Netzwerksicherheit oder Fehlertoleranz (beispielsweise bein Authentifizierungsversuche) gewährleistet werden [2].

2.3 Arten der Authentifizierung

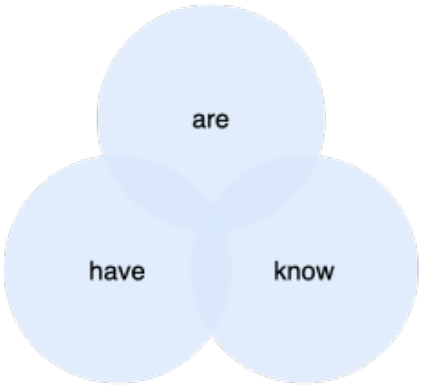

Abbildung 2.4: Umsetzungsmöglichkeit mit Keycloak

- Die Authentifizierung dient häufig als erste Verteidigungslinie von Systemen. [3]
- Faktor Something you know. Diese Methode nutzt Informationen, welche nur dem Nutzer bekannt sind, um seine Identität zu bestätigen [3].

- Faktor Something you have. Diese Methode nutzt physische Objekte, welche sich im Besitz des Nutzers befinden, um seine Identität zu bestätigen. Dazu gehören u.a. Smartcards und Hardware-Token [3].
- Faktor Something you are. Diese Methode nutzt biometrische Daten des Nutzers, um seine Identität zu bestätigen. Dazu gehören u.a. Fingerabdrücke, Iris-Scans und Gesichtserkennung [3].
- Ein Problem dieser Methode ist, dass sich menschliche Eigenschaften im Laufe der Zeit verändern können. Auch Verletzungen oder Krankheiten können die biometrischen Daten verändern [3].
- Nicht standardmäßig, aber weiterer Faktor ist something you perform or produce. Diese Methode nutzt beispielsweise die Stimme oder die (digitale) Unterschrift des Nutzers, um seine Identität zu bestätigen [3].

2.4 Passwortbasierte Authentifizierung

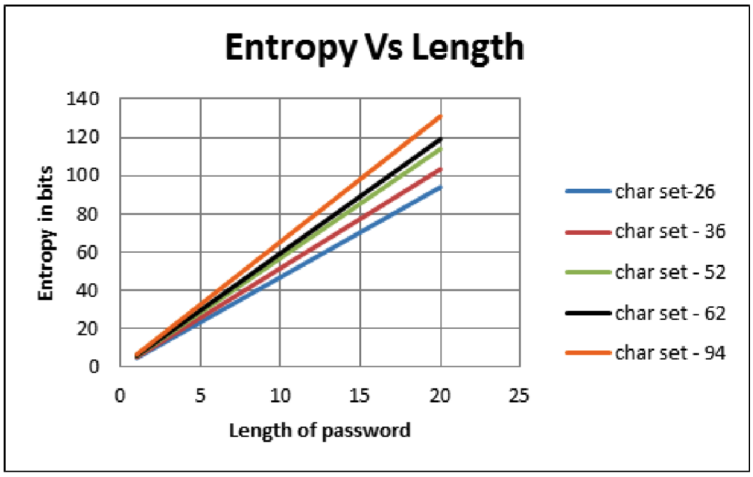

Abbildung 2.5: Umsetzungsmöglichkeit mit Keycloak

- Die heutzutage am häufigsten genutzte Methode zur Authentifizierung ist die passwortbasierte Authentifizierung [4] [3] [5].
- Die Sicherheit von Systemen basiert somit auf der Sicherheit der Passwörter [3].

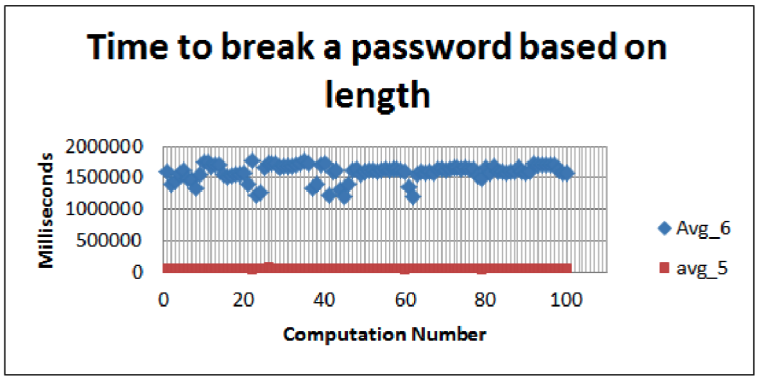

Abbildung 2.6: Umsetzungsmöglichkeit mit Keycloak

- Passwörter gelten als eins der größten Risiken für Systeme, da sie viele Angriffsvektoren bieten [5] [6].
- 81% der Hackerangriffe basierten auf der Kompromittierung von Passwörtern [7].
- 2017 waren Phishing E-Mails die häufigste Angriffsmethode [7].
- Obwohl es bereits alternative Ansätze gibt, werden Passwörter weiterhin genutzt. Das liegt an der Einfachheit und dem geringen Aufwand, welche die Nutzung von Passwörtern mit sich bringt [5].
- Eine Vielzahl an großen Unternehmen wurden bereits Opfer von der Veröffentlichung von Passwörtern, obwohl ein hoher Aufwand betrieben wird diese zu schützen. Da sich die Enthüllung der Passwörter allerdings als Angriffsziel bei Angreifern etabliert hat, ist selbst ein hoher Aufwand nicht ausreichend [3].
- Dabei handelt es sich am häufigsten um alphanumerische Passwörter, welche aus einer Kombination von Groß- und Kleinbuchstaben, Zahlen und Sonderzeichen bestehen [4].
- Passwörter können durch verschiedene Angriffe kompromittiert werden. Angreifer können Zugriff auf die Datenbank erhalten, in welcher die Passwörter gespeichert sind. Aber auch auf persönlicher Ebene können Passwörter erlangt werden. Aufgeschriebene Passwörter können in fremde Hände geraten. Auch Social Engineering kann genutzt werden, um Passörter mit

Hilfe von Phishing oder Keyloggern zu erlangen. Häufig lassen sich Passwörter allerdings auch mit Hilfe von Brute-Force- oder Dictionary-Attacken kompromittieren [4] [8].

- Brute-Force-Attacken versuchen alle möglichen Kombinationen von Zeichen, welche ein Passwort enthalten kann, auszuprobieren. Je höher dabei die Anzahl an möglichen Kombinationen ist, desto aufwändiger wird es ein Passwort zu erraten.
- Je länger ein Passwort, desto schwieriger zu knacken. Länge auch wichtiger als Zeichenraum [4].
- Hier auch kurz auf die Mathematik dahinter eingehen.
- Studien zeigen, dass Nutzer dazu neigen gleiche oder ähnliche Passwörter für verschiedene Zugänge zu nutzen [4] [9].
- Verfügen Angreifer über ein Passwort eines Nutzers, können häufig auch andere Zugänge übernommen werden [4] [8].
- Obwohl die Angriffsvektoren und Schwachstellen von Passwörtern schon lange bekannt sind, bleiben diese unverändert bestehen. [9].

2.4.1 Speicherung

- Viele Angreifer versuchen Passwörter zu kompromittieren, indem sie Zugriff auf die Datenbank erhalten, in welcher die Passwörter gespeichert sind. Mit Hilfe von Passwörtern erhoffen sich die Angreifer Zugriff auf Systeme und Netzwerke [3].
- Passwörter können auf verschiedene Arten gespeichert werden. Dadurch können verschiedene Angriffsvektoren entstehen [4].
- Plaintext am schlechtesten. Werden die Passwörter in lesbarer Form gespeichert, können Angreifer alle Passwörter auslesen, sobald sie Zugriff auf die Datenbank haben. Dabei muss kein weiterer Aufwand betrieben werden [4].
- Verschlüssellung besser, aber nicht optimal. Verschlüssellung ist zurückführbar. Gelangen Angreifer an den benötigten Schlüssel, können sie alle Passwörter entschlüsseln und auslesen [4].

- AM besten Hashing mit Salt. Sobald ein Passwort gehasht wurde, kann es nicht mehr zurückgerechnet werden. Durch einen individuellen Salt kann ebenfalls verhindert werden, dass Angreifer die Passwörter mit Hilfe von Rainbow-Tables entschlüsseln können [4].
- auch noch zwei salts möglich - einer public einer private. schützt vor offline angriffen [4].
- Vielleicht hier noch ganz kurz auf Hash Funktionen eingehen?

2.4.2 Faktor Mensch

- Die Sicherheit ist nicht nur von den technischen Aspekten abhängig [9].
- Ein Großteil der Angriffsfläche von Passwörtern entsteht durch den Faktor Mensch [5].
- Von Menschen erstellte Passwörter sind keine echten Zufallswerte. Das liegt insbesondere daran, dass Nutzer sich Passwörter merken können müssen. Daher beinhalten Passwörter häufig Informationen, welche einen Bezug zum Nutzer haben. Dazu gehören beispielsweise Namen, Geburtsdaten, Adressen oder andere persönliche Informationen. Auch Passwörter, welche einfache Tastaturmuster beinhalten sind sehr beliebt. Dazu zählen beispielsweise „qwertz“ oder „123456“ [4] [3] [5].
- Das Hauptproblem entsteht dabei durch die benötigte Einprägsamkeit der Passwörter [5].
- Es ist sehr schwierig für Nutzer sich verschiedene komplexe Passwörter zu merken. Daher neigen Nutzer dazu, einfache Passwörter zu nutzen oder Passwörter für verschiedene Zugänge zu wiederholen [4].
- Das ist der Hauptgrund dafür, dass Nutzer dazu neigen, einfache Passwörter zu nutzen oder Passwörter für verschiedene Zugänge zu wiederholen [5].
- Diese Faktoren führen dazu dass die Anzahl an genutzten Passwörtern deutlich geringer ist als die gesamte Menge an möglichen Passwörtern [3].

- Ebenfalls ist häufig die Motivation der Nutzer gering komplexe Passwörter zu erstellen. Dies liegt häufig daran, dass die Nutzer nicht die Gefahr erkennen und nicht überzeugt von Guidelines und Richtlinien zur Erstellung von Passwörtern sind [5].
- Nutzer tendieren dazu bewusst schwache Passwörter zu erstellen, die den Anforderungen der Richtlinien entsprechen. Das führt zu einem kontraproduktiven Effekt, da die Sicherheit geringer wird [5].
- Sehr komplexe Richtlinien führen demnach nicht zwangsmäßig zu einer höheren Sicherheit. Vielmehr kann das Gegenteil erreicht werden [5] [8].
- Aktive Internet-Nutzer verwalten durchschnittlich 15 Passwörter pro Tag [9].
- Eine der größten Schwachstellen ist also die Wahl des Passwortes durch den Nutzer [3].
- Ein Domino Effekt kann entstehen, wenn mit Hilfe eines Passwortes weitere Passwörter kompromittiert werden. So können mehrere Systeme indirekt davon betroffen sein, sobald ein Passwort kompromittiert wurde [9].
- Das macht von Menschen erstellte Passwörter anfälliger für Angriffe, da diese einfacher zu erraten sind [4].
-

2.5 Passwortlose Authentifizierung

- Unter der passwortlosen Authentifizierung werden verschiedene Verfahren zusammengefasst, welche die Nutzung von Passwörtern ersetzen.
- Während bei passwortbasierten Verfahren also der Faktor *something you know* genutzt wird, wird bei passwortlosen Verfahren auf andere Faktoren zurückgegriffen.
- Die Fast Identity Online (FIDO) Allianz nutzt die Bezeichnung passwortlos, um eine Single-Factor Authentication (SFA) oder Multi-Factor Authentication (MFA) mit Hilfe eines Authentifizierungsgerätes zu beschreiben [6].

- Passwortlose Verfahren werden als sicherer angesehen, da viele der in **Passwortbasierte Authentifizierung** aufgeführten Angriffsvektoren nicht auf passwortlose Ansätze anwendbar sind [10] [11].
- Auch die Benutzerfreundlichkeit soll durch passwortlose Verfahren verbessert werden, da diese sich keine Passwörter mehr merken müssen [10].
- Allerdings haben sich passwortlose Verfahren noch nicht durchgesetzt und sind nicht weit verbreitet.
- Das liegt auch daran, dass für passwortlose Verfahren häufig zusätzliche Hardware benötigt wird, welche mit zusätzlichen Kosten verbunden ist [10].
- Auch die Umgewöhnung an eine neue Art der Authentifizierung wird als eine Hürde für die Etabliserung von passwortlosen Verfahren angesehen [10].
- Viele verschieden Möglichkeiten Passswortlose Authentifizierung zu implementieren:
- lediglich ein Überblick, da eine detailiertere BEschreibung den Scope dieser Arbeit überschreiten würde.

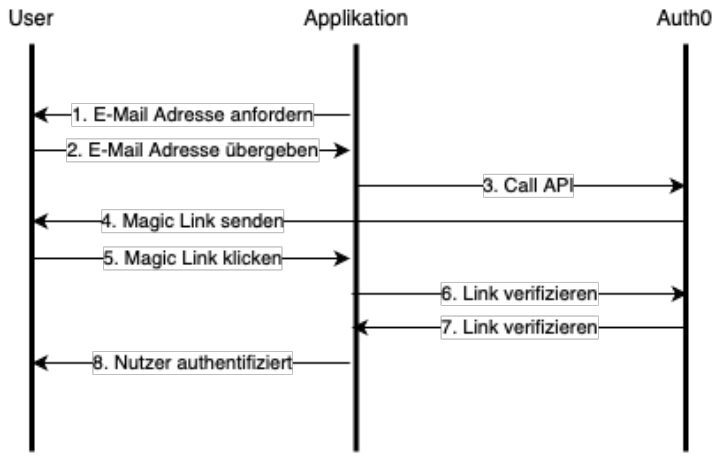

Abbildung 2.7: Umsetzungsmöglichkeit mit Keycloak

- Magic Link: Bei einem Magic Link handelt es sich um eine Authentifizierungsmöglichkeit, bei welcher Nutzer lediglich ihren Benutzernamen oder ihre E-Mail-Adresse zur Anmeldung angeben müssen. Anschließend erhält der Nutzer eine E-Mail mit einem dazugehörigen Link, welcher genutzt

wird, um seine Identität zu bestätigen. [10] [11] Dieser Link beinhaltet einen Authentication Code, welcher im Hintergrund abgeglichen und validiert wird. Ist die Validierung erfolgreich wird der Nutzer authentifiziert und angemeldet. Nach der Anmeldung verliert der AUthentication Code seine Gültigkeit und somit auch der Link selbst. [10] Die Sicherheit dieses Verfahrens basiert dabei auf der Annahme, dass der Mail-Server bzw. der Zugang zum Account des Nutzers ausreichend geschützt ist. Ist diese Annahme nicht gegeben können sich auch andere Personen mit dem Link des eigentlichen Nutzers authentifizieren ohne autorisiert zu sein. [10] Passwort bleibt für Zugriff auf E-Mail Zugang notwendig, allerdings würde so die Anzahl an Passwörtern für den Nutzer reduziert werden. Vorteile: Sehr benutzerfreundlich und einfach [11]. Implementierung und Kosten zur Instandshaltung verhältnißmäßig gering [11]. Nachteile: Nutzung von Spam-Filtern, kann die Benutzerfreundlichkeit stark beeinträchtigen. So könnten die zugehörigen Mails fälschlicherweise als Spam klassifiziert werden oder eine erhöhte Wartezeit auf die Mail entstehen [11]. Die Sicherheit des Verfahrens hängt von der Sicherheit des Mail-Servers ab. Ist dieser nicht ausreichend geschützt, können Angreifer Zugriff auf die Mails erhalten und sich somit auch mit dem Link authentifizieren [10]. Dies kann geschehen, ohne dass der Nutzer davon etwas mitbekommt [10].

- One-Time Password (OTP): Das Konzept hinter OTPs ähnelt dem des Magic Links. Nutzer geben ihre E-Mail-Adresse oder ihre Handynummer an (diese können ebenfalls einem Benutzernamen zugewiesen sein) und erhalten eine E-Mail/SMS, welche ein OTP beinhaltet [10] [11]. Dieses Wird vom System abgeglichen und validiert. Ist die Validierung erfolgreich wird der Nutzer authentifiziert und angemeldet. Nach der Anmeldung verliert das OTP seine Gültigkeit. [10] Häufig werden OTPs allerdings nicht für eine oben beschrieben SFA genutzt, sondern dienen als zusätzlicher Faktor für eine MFA. [10] So können beispielsweise Authenticator Apps zur Bereitstellung con OTPs genutzt werden, um die etabliertere passwortbasierte Authentifizierung sicherer zu gestalten. Im Gegensatz zu statischen, von Anwendern gewählten Passwörtern sind OTPs dynamisch erzeugt und haben nur eine geringe Lebensdauer. So wird eine höhere Sicherheit gewährleistet, da OTPs nur schwierig durch stupides Erraten oder Brute Force Attacken erbeutet werden können. [10] Für die Umsetzung von OTPs gibt es mehrere Möglichkeiten. Zwei häufig verwendete Optionen sind HMAC-based One-Time Password (HOTP) und Time-based One-Time Password (TOTP). [10] HOTPs basieren auf der technischen Spezifikation RFC 4226. Sie wer-

den mit Hilfe von Hash-based Message Authentication Code (HMAC) und unabhängig von der Zeit generiert. Neue HOTPs können Event-basiert von dem Nutzer angefordert werden. [10] TOTPs basieren auf der technischen Spezifikation RFC 6238 und werden in Abähngigkeit zu der Zeit erstellt. Sie ändern sich nach einem vordefinierten Zeitintervall und sind somit sehr kurzlebig. [10]

- Vorteile: sehr effektiv für MFA [11]. sehr benutzerfr4eundlich und einfach [11]. bieten eine erhöhte Sicherheit und verringern die Angriffsvektoren von statischen Passwörtern. [10] haben sich für MFA bereits etabliert [11]. Vielzahl an Möglichkeiten, per mail/sms/app oder security keys [10] [11].
- Nachteile: Häufig nur für MFA genutzt also nicht passwortlose SFA je nach implementierung ähnliche nachteile wie Magic Links OTPs als SFA werden nicht überall unterstützt.
- Biometrische Daten: Eine häufig genutzte Methode zur Authentifizierung auf mobilen Endgeräten ist die Nutzung von biometrischen Daten [11]. Hierbei werden einzigartige biometrische Merkmale des Nutzers genutzt um seine Identität zu verifizieren. Dazu gehören beispielsweise Fingerabdrücke oder eine Gesichtserkennung [11]. Diese Variante kann auch im Unternehmenskontext unter anderem mit Windows Hello for Business genutzt werden. Vorteile: Viele mobile Endgeräte arbeiten bereits mit biometrischen Daten [11]. Keine große Umgewöhnung für Nutzer, da diese oftmals bereits mit biometrischen Daten arbeiten [11]. nahezu einzigartig und somit deutlich schwieriger anzugreifen als passwörter [11]. Bereits für Unternehmenskontext verfügbar, beispielsweise mit Windows Hello for Business. Nachteile: Äußere Bedingungen können die Erkennung von biometrischen Daten beeinträchtigen. beispielsweise schlechtes licht bei Gesichtserkennung oder staubige Umgebungen bei Fingerabdruckscanner [11]. Biometrische Daten können sich im Laufe der Zeit verändern. Auch Verletzungen oder Krankheiten können die biometrischen Daten verändern [3]. Im Unternehmenskontext häufig verschiedene Hersteller und Geräte, welche nicht alle biometrischen Daten unterstützen oder nicht untereinander kompatibel sind [11].
- FIDO2: WIRD IN KAPITEL GENAUWER BESCHRVIEBEN

2.5.1 Magic Link

2.5.2 One Time Password (OTP)

- Passwörter die sich mit jedem Login ändern. Dadurch wird das Risiko verringert, dass das Passwort erraten werden kann [3].

2.6 YubiKey

Abbildung 2.8: Umsetzungsmöglichkeit mit Keycloak

- Ein Security Key ist eine Hardware, welche es ermöglicht einen Nutzer zu authentifizieren, indem dieser mit dem Security Key interagiert (beispielsweise durch einen Knopfdruck) [12].
- Häufig werden Security Keys so designed, dass sie per USB an einen Computer angeschlossen werden können [12].
- Die YubiKeys 5 ermöglichen drei Arten der Authentifizierung: 1. SFA Ersetzt Passwörter durch ein passwortloses *tap-n-go* Verfahren. 2. **2FA!** (**2FA!**) Sichert ein Passwort zusätzlich mit einem *tap-n-go* Faktor ab. Der Security ist somit der zweite Faktor (*something you have*). 3. MFA Verbindet die passwortlose *tap-n-go* Authentifizierung mit einer PIN. (EIG AUCH SFA ODER NICHT)

2.6.1 Usability

- Vorteile:

- Ergebnisse zeigen, dass Nutzer grundsätzlich bereit sind, Passwörter durch passwortlose Verfahren zu ersetzen [13].
- Passwortlose Verfahren mit Yubikey wurden mehr akzeptiert als tradionelle passwortbasierte Verfahren [13].
- Implizite Garantie, dass sich lediglich Nutzer authentifizieren können, welche auch im Besitz des Authentifizierungsgerätes sind. [13].
- Durch die Nutzung von FIDO2 kann die Usability verbessert werden, da Nutzer sich keine Passwörter mehr merken müssen. Häufig wird das Verwalten der immer höher werdenden Anzahl an Passwörtern als Problem angesehen [13] [6].
- Es wird ein deutlich geringerer kognitiver Aufwand benötigt, da Nutzer keine neuen Passwörter mehr erstellen und merken müssen [13].
- Zum aktuellen Zeitpunkt wird FIDO2 bereits von einer Vielzahl an Browsern unterstützt. Zusätzlich bieten immer mehr Online-Dienste die Möglichkeit an sich mit Hilfe von FIDO2 zu authentifizieren [13] [6].
- Es handelt sich um offene und standardisierte Protokolle [6].
- Nachteile:
- Im Falle einer SFA wird der Verlust des Authentifizierungsgerätes als größtes Problem angesehen. Bei Verlust hat auch der Nutzer keinen Zugriff mehr und aktuell gibt es noch keine sichere und effiziente Möglichkeiten, um den Zugriff wiederherzustellen (vor allem ohne Pause) [13].
- Da es sich um zusätzliche Hardware handelt kann diese ebenfalls kaputt gehen [6].
- Im Unternehmenskontext, kann die Verwaltung und Verteilung der Authentifizierungsgeräte zu einem Problem werden [6].
- Da es sich um Hardware handelt, können Zugänge nicht an vertraute Personen weitergegeben werden, da der Zugang nur mit dem Authentifizierungsgerät möglich ist [13].
- Ohne das Authentifizierungsgerät sind keine spontanen Logins möglich [13].
- Es wird ein physischer Aufwand benötigt, da das Authentifizierungsgerät mitgeführt werden muss [13].

- Bereits das aus der Tasche holen des Authentifizierungsgerätes ist für manche Nutzer bereits eine Hürde [6].
- Authentifizierungsgeräte sind häufig mit Kosten verbunden, welche vom Nutzer getragen werden müssen [13].
- Nutzer haben Probleme ein neues Verfahren für die Authentifizierung zu nutzen, da sie sich an das alte Verfahren gewöhnt haben. Das führt dazu, dass Nutzer das neue Verfahren als kompliziert und ungewohnt empfinden. Sie verfügen häufig nicht über das nötige Wissen, um die Funktion und Sicherheit des Verafahrens zu verstehen [13].
- Selbst Nutzern, welchen das Konzept der passwortlosen Authentifizierung gefällt, nutzen häufig weiterhin Passwörter [6].
- Nutzer wollen keine Angewohnheiten verändern, wenn die nicht dazu gezwungen sind [6].
- Nutzer verwenden lieber Passwörter, da sie das Konzept und die Technologie besser verstehen [13].
- Nicht zwangsweise schneller als die Nutzung von Passwortmanagern [6].
- Allgemein fällt das Feeback von Nutzern weniger positiv aus, wenn diese vorher bereits Passwortmanager genutzt haben [6].
- Fazit:
- Insgesamt lassen sich noch nicht alle Szenarien mit FIDO2 abdecken. Es gibt noch spezielle Fälle, in welchen die Nutzung von Passwörtern weiterhin notwendig ist [13].
-

2.7 Fido2

- FIDO2 wird von der FIDO und dem World Wide Web Consortium (W3C) entwickelt und bereitgestellt [13] [6].
- Die FIDO Allianz ist eine Organisation mit weltweit über 250 Mitgliedern. Darunter befinden sich Unternehmen wie Google, Microsoft, Apple Amazon, Facebook, Visa und viele mehr [13] [6].

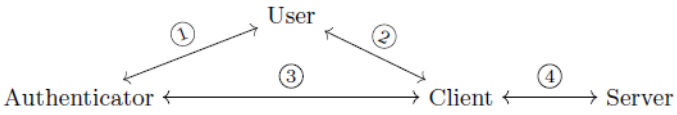

Figure 1: Communication channels

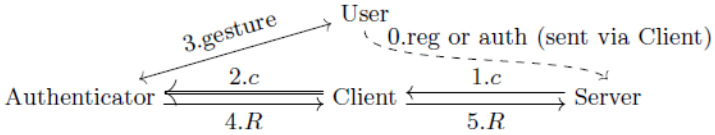

Figure 2: FIDO2 flow (simplified): double arrow = CTAP2 authorized command.

Abbildung 2.9: Umsetzungsmöglichkeit mit Keycloak

- Ziel ist es Nutzer zu authentifizieren, ohne, dass diese eine Passwort nutzen müssen [8] [7].
- Basiert auf der Nutzung eines internen oder externen Authentifizierungsgerätes [8] [7].
- Dabei können Authentifizierungsgeräte, ebenfalls mit einer PIN oder einem biometrischen Merkmal, geschützt werden [6].
- Hierbei ist ein PIN allerdings nicht gleichzusetzen mit einem Passwort. Der PIN wird lediglich für das Authentifizierungsgerät genutzt und wird auch nur auf diesem gespeichert [6] [7].
- Es handelt sich dabei also auch nicht um eine MFA, sondern, um einen einzelnen Faktor, welcher lediglich den Zugriff das Gerät selbst authentifiziert [7].
- FIDO2 unterstützt sowohl MFA als auch SFA [13] [6].
- Viele Alternativen zur passwortbasierten AUthentifizierung existieren bereits. Diese werden allerdings nur in einem sehr geringen Ausmaß genutzt [6].
- Stellt Zugangsdaten bereit, welche nicht gephisht oder von Datenlecks betroffen sein können [13].

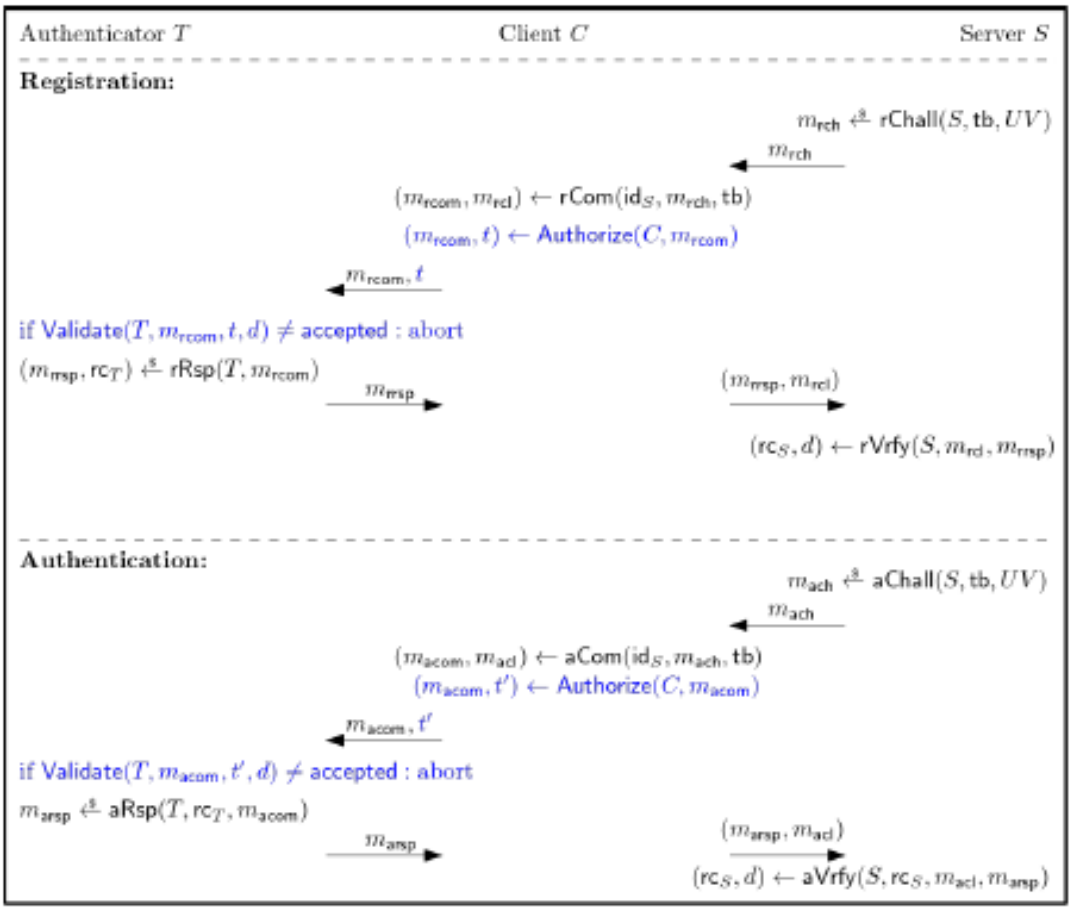

Abbildung 2.10: Umsetzungsmöglichkeit mit Keycloak

- Das liegt daran, dass keine geteilten Geheimnisse zwischen Nutzer und Dienst existieren, welche auf einem Server gespeichert werden [8].
- Wird von fast allen Browsern standardmäßig unterstützt [13].
- Viele verfügbare Authentifizierungsgeräte. Z.B. Security Keys oder auch Smartphones. Beispielsweise Apples Touch ID oder Face ID [13].
- Besteht aus zwei Komponenten: CTAP2 für die Kommunikation zwischen Client und Authentifizierungsgerät und WebAuthn für die Kommunikation zwischen Client und Server [6].
- Dabei wird WebAuthn von der W3C spezifiziert und CTAP2 von der FIDO

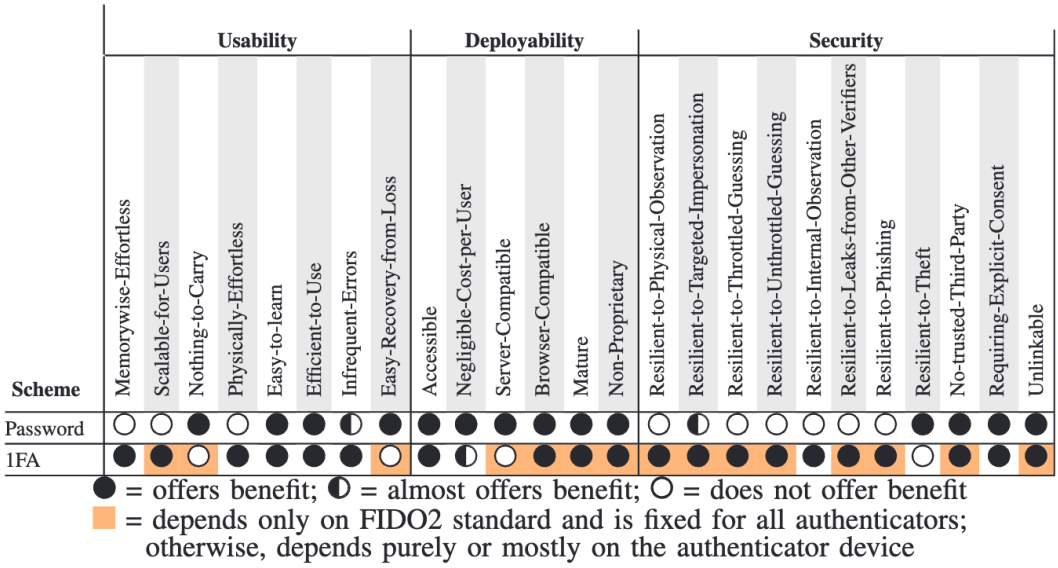

Abbildung 2.11: Umsetzungsmöglichkeit mit Keycloak

Allianz [6].

2.7.1 Webauthn

- WebAuthn ist ein Standard, welcher von dem W3C entwickelt wird. Das Protokoll erlaubt es Webanwendungen Nutzer zu authentifizieren. Dies kann dabei auch über Client-to-Authenticator Protocol 2 (CTAP2) erfolgen [13]. ?
- Wurde 2019 ein offizieller Webstandard [6].
- Spezifiziert eine standardiserte, vom Browser unabhängige JavaScript API zur Authentifizierung von Nutzern für Webanwendungen. So können Webanwendungen eine Authentifizierung integrieren, welche resistent gegenüber Phishing, Datenlecks und Passwortdiebstahl ist. Anstelle von geteilten Geheimnissen nutzt WebAuthn public-key Kryptographie, um einzigartige Zugangsdaten für jede Webanwendung zu erstellen, welche nur auf dem Gerät des Nutzers gespeichert werden [6].
- Passwortloses Challenge-Response-Verfahren zwischen Client und Server [7].

- WebAuthn unterstützt zwei Opertationen: Registrierung und Anmeldung [7].
- In der Registrierungsphase sendet der Server dem Authentifizierungsgerät über den CLient eine zufällige Challenge. In dieser Phase signiert das Authentifizierungsgerät mit Hilfe seines privaten Schlüssels die CHallenge und sendet zusätzlich öffentliche Anmeldedaten für zukünftige Anmeldungen an den Server. Meldet sich ein bereits registrierter Nutzer an, wird die Challenge des Servers erneut von dem Authentifizierungsgerät signiert zurück an den Server gesendet. Der Server kann die Signatur mit Hilfe des öffentlichen Schlüssels verifizieren und den Nutzer authentifizieren [7].
- Registrierungsphase: Der Server S sendet eine challenge message m_{rch} über den Client C an den Security Key. Diese Challenge beinhaltet eine randomisierte Nonce, Parameter (beispielsweise, ob eine Nutzerverifizierung notwendig ist) und optional einen wert *tb*, welcher den zugrunde liegenden Kanal eindeutig identifiziert (typischerweise eine Transport Layer Security (TLS) Verbindung). Der Client C erhält die challenge message m_{rch} und wandelt diese in eine command message m_{rcom} und eine client mesasage m_{rcl} um. die command message m_{rcom} wird an den Security Key T übermittelt. Der Security Key T erzeugt öffentlich-privates Schlüsselpaar, welches an den Server S gebunden ist und diesem ermöglicht eine Verifizierung, während der folgenden AUthentifizierungsphase durchzuführen. Zudem gibt der Security Key T eine response message m_{rrsp} aus. Der Client übergibt diese und die client message m_{rcl} an den Server S. Die response message m_{rrsp} beinhaltet einen *attestation type*, welcher es dem Server S ermöglicht eine Verifizierung während der Registrierungsphase durchzuführen und beinhaltet den öffentlichen Schlüssel. WebAuthN 2 unterstützt fünf *attestation types*. Häufig werden die types *None* und *Basic* verwendet. Die restlichen types sind *Self*, *AttCA* und *AnonCA*. [14]
- Authentifizierungsphase: Der Client empfängt die challenge message m_{ach} von Server S und wandelt diese in eine command message m_{acom} und eine client message m_{acl} um. Die command message m_{acom} wird an den Security Key T übermittelt. Der Security Key T erzeugt eine response message m_{arsp}, welche mit dem privaten Schlüssel signiert wird und sendet diese an den Server S (über den Client C). Der Server S akzeptiert die response message m_{arsp} und die client message m_{acl} nur, wenn sie sich mit dem dazugehörigen öffentlichen Schlüssel verifizieren lassen. [14]

- Die Sicherheit von WebAuthn basiert auf dem Beweis, dass RSASSA PKCS1-v1_5 und RSASSA-PSS als Existential Unforgeability under a Chosen Message Attack (EUF-CMA) gelten und der Annahme, dass SHA-256 kollisionsresistent ist [7].

2.7.2 CTAP2

- 2018 wurde CTAP2 als internationaler Standard der International Telecommunication Union Telecommunication Standardization Sector (ITU-T) anerkannt [7].
- CTAP2 ist ein Protokoll auf der Anwendungsebene, welches für die Kommunikation zwischen eines WebAuthn Clients und eines konformen Authentifizierungsgerätes genutzt wird. Das Authentifizierungsgerät kann dabei ein externes Gerät sein wie beispielsweise ein Security Key, welches über USB, Bluetooth oder NFC eine Verbindung mit dem Client aufbaut. Aber auch ein internes Gerät wie beispielsweise ein Fingerabdruckscanner oder ein Trusted Platform Module können als Authentifizierungsgerät genutzt werden [13].
- CTAP2 spezifiziert, die Kommunikation zwischen einem Authentifizierungsgerät und einem Client. Der Client ist dabei üblicherweise ein Webbrowser. Das Ziel ist es zu garantieren, dass der CLient das AUthentifizierungsgerät nur nutzen darf, wenn der Nutzer dies erlaubt. Dafür muss der Nutzer beispielsweise einen Knopf am Authentifizierungsgerät drücken und/oder sich mit Hilfe eines PINs oder eines biometrischen Merkmals beim AUthentifizierungsgerät authentifizieren [7].
- Das Ziel ist es somit einen Client an das AUthentifizierungsgerät zu binden. Ist ein CLient nicht an das authentifizerungsgerät gebunden, kann dieser sich nicht authentifizieren [7].
- besteht aus mehreren Phasen. 1. In der Setup Phase initialisiert ein Client C' einen PIN, welcher vom User übergeben wird an den Security Key T. 2. In der Binding Phase tauschen ein Client C (nicht zwangsweise C') und der Security Key T einen gemeinsamen Verbindungsstatus aus, wenn der Client C in der Lage ist, Informationen über die auf dem Security Key T gespeicherte PIN zu liefern. So soll eine einzigartige Verbindung zwischen dem Client C und dem Security Key T hergestellt werden. Schlägt der Client C drei mal in Folge fehl die PIN zu liefern, wird der Security Key T

neu gestartet und der Verbindungsstatus wird zurückgesetzt. Schlägt der Client C insgesamt acht mal fehl, wird der Security Key T gesperrt. 3. Ist diese Phase erfolgreich, autorisiert der Client C jeden Befehl, indem eer einen Tag t ausgibt, welcher mit der command message an den Security Key T übermittelt wird. Der Security Key T fährt lediglich fort, wenn eine *positive decision* d des Users vorliegt (beispielsweise einem Knopfdruck) und validiert darauf hin die command message und den Tag t. [14]

- CTAP2 nutzt unauthentifizierten Diffie-Hellman Schlüsselaustausch [7].
- Dieser kann von Man In The Middle (MITM) Angriffen betroffen sein [7].
- In der Binding Phase sendet das Authentifizierungsgerät dem CLient ein pinToken, welcher beim hochfahren des Authentifizierungsgerätes generiert wird. Dieser pinToken wird lokal auf dem AUthentifizierungsgerät gespeichert und wird von dem verbundenen Client in der Access Channel Phase genutzt, um die nachfolgenden Nachrichten des Clients zu autorisieren [7].
- Jedem AUthentifizierungsgerät wird ein pinToken pro hochfahren zugeordnet. Das bedeutet mehrere CLients erezugen mehrere Access Channels mit dem selbem Authentifizierungsgerät und dem selben pinToken [7].
- Dadurch wird die Sicherheit von CTAP2 limitiert ??

2.7.2.1 CTAP2.1

- Gilt in Verbindung mit WebAuthn 2 als Post-Quantum (PQ) bereit, da ein Operationsmodus ermöglicht wird, der nur kryptographische Primitive, digitale Signaturen und Key Encapsulation Mechanism (KEM) verwendet [14].
- Im Gegensatz zu CTAP2 basiert CTAP2.1 nicht auf unauthentifizierten Diffie-Hellman Schlüsselaustausch, sondern auf einem sogeannten PIN/UV Auth Protocol (puvProtocol), wodurch die PQ-Sicherheit ermöglicht wird [14].
- In CTAP2 wird der Verbindungszustand als *pinToken* definiert, welcher aus mehreren 128 Bit-Blöcken besteht und keine maximale Begrenzung der Länge beseitzt. In CTAP2.1 wird der Verbindungszustand als *pinUvAuthToken* definiert welcher eine feste Länge von 128 oder 256 Bit besitzt [14].

- Der pinToken von CTAP2 wird bis zum nächsten Neustart wiederverwendet. Der pinUvAuthToken von CTAP2.1 wird nach jeder erfolgreichen Authentifizierung neu generiert. Das führt dazu, dass CTAP2.1 eine Strongly Unforgeable (SUF)-t' Sicherheit aufweist und CTAP2 lediglich eine Unforgeable (UF)-t' Sicherheit [14].
- CTAP2 erlaubt es Security Keys und Clients nur den pinUvAuthToken zu teilen, wenn der Nutzer den korrekten PIN eingegeben hat. CTAP2.1 ermöglicht zusätzlich, dass der Nutzer sich mit Hilfe eines biometrischen Merkmals authentifiziert [14].

2.7.3 Sicherheit

- FIDO2 ist eine Erweiterung des FIDO U2F Protokolls und bietet die selbe Sicherheit wie public key Kryptographie [13].
- Es handelt sich um geprüfte assymetrische Kryptographie [6].
- Es handelt sich dabei um ein Challenge-Response-Verfahren mittels Hardware basierten Authentifizierungsgeräten. Dies bietet einige Vorteile gegenüber passwortbasierten Verfahren. Es gibt keine geteilten Geheimnisse zwischen Usern und Diensten, welche durch Phishing oder Datenlecks kompromittiert werden können. Dabei ist das selbe Authentifizierungsgerät für mehrere Dienste nutzbar, ohne, dass sich dabei eine Verknüpfung zurückführen lässt [13] [6].
- lediglich die Session kann kompromittiert werden [8].
- Authentifizierungsgeräte lassen sich mit zusätzlichen PINs oder biometrischen Merkmalen absichern, um sich ebenfalls vor Diebstahl schützen [7].
- Unauthentifizierter Diffie-Hellman Schlüsselaustausch könnte durch ein Password Authenticated Key Exchange (PAKE) Verfahren ersetzt werden [7].
- Das Paper gibt an, dass dieses sicherer und effizienter sein soll [7].
- In folgendem Szenario: 1. Der Nutzer besitzt einen Security Key, welcher mit einem drückbaren Knopf oder ähnlichen ausgestattet ist. 2. Der Security Key ist mit einem geheimen PIN geschützt. 3. Der Nutzer autorisiert vertrauten Clients auf den Security Key zuzugreifen. 4. Der Nutzer verbindet seinen Security Key mit mehreren Clients und nutzt diese um sich bei mehreren Webdiensten zu registrieren/anzumelden. Dann ist versichert,

dass: 1. Die Authentifizierung von dem Security Key durchgeführt wurde, welcher die genutzen Zugangsdaten bei dem Webdienst registriert hat. 2. ein autorisierter Befehl auf den Security Key zugegriffen hat. 3. und dieser autorisierte Befehl von einem autorisierten Client beauftragt wurde (sollte der Nutzer den korrekten PIN eingegeben haben). Dies setzt voraus, dass: 1. Der Security Key nicht gestohlen wurde. 2. Der PIN des Security Keys nicht kompromittiert wurde. 3. Der autorisierte Client nicht kompromittiert wurde (korrekte Ausführung von CTAP2 und CLient ist nicht von böswilliger Software betroffen). [7].

- Wird ein Security Key gestohlen, kann dieser nur genutzt werden, wenn ebenfalls der PIN bekannt ist [7].

3 Umsetzung

3.1 Aktueller Stand der LSY

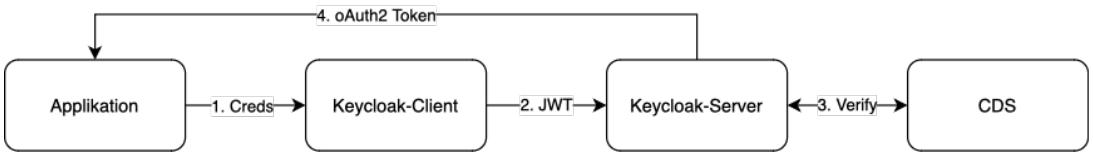

Abbildung 3.1: Aktuelle Umsetzung der Abteilung

- Innerhalb der Abteilung wird eine passwortbasierte Authentifizierung durchgeführt, welche zusätzlich durch MFA geschützt wird.
- Webanwendung nutzt einen eigene Anwendung, welche sich als Keycloak-Client aussgibt. Der Nutzer gibt seine Zugangsdaten an den Keycloak-Client weiter, welche diese verarbeitet. Dieser wandelt die Zugangsdaten in einen validen JWT-Token um und übergibt diesen an den Keycloak-Server. Dieser validiert die Zugangsdaten gegen die ??? (CDS). Ist die Validierung erfolgreich, wird vom Keycloak-Server ein oAuth2-Token erstellt und zurück an die Applikation übergeben.
- Innerhalb der LSY können sich Applikationen allerdings auch gegen das Azure AD authentifizieren lassen.

3.2 Integration eines Yubikeys in die LSY

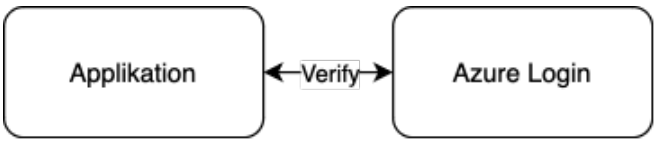

Abbildung 3.2: Umsetzungsmöglichkeit mit Azure AD

- Grundsätzlich gibt es zwei Möglichkeiten in die aktuelle Applikation eine passwortlose Authentifizierung mit Hilfe eines Yubikeys zu integrieren: Die Nutzung der Authentifizierung gegen das Azure AD oder die eine veränderte Nutzung der aktuellen Keycloak-Lösung.

Eine Lufthansa-weite Policy für die Nutzung der Azure AD verbietet allerdings die Nutzung eines Security Keys für eine SFA. Hier kann der Security Key lediglich als zweiter Faktor genutzt werden. Dies kann jeder Nutzer selber verwalten. Registriert ein Nutzer seinen Security Key in seinem Profil, erscheint bei der Anmeldung (nach der Eingabe des Passwortes) ein zusätzliches Feld:

Abbildung 3.3: Umsetzungsmöglichkeit mit Keycloak

Verwendet der Nutzer einen Security Key wird dieser im Folenden aufgefordert die zugehörige PIN einzugeben und den Knopf des Security Keys zu drücken. Grundsätzlich ist eine passwortlose Authentifizierung mit Hilfe eines Security Keys innerhalb der Azure AD möglich. Da eine Änderung dieser Lufthansa Policy notwendig wäre, übersteigt dies allerdings den Rahmen dieser Arbeit.

Da allerdings aktuell eine beschriebene Nutzung von Keycloak stattfindet und Keycloak eine passwortlose Authentifizierung mit Hilfe eines Security Keys unterstützt, wäre eine Umsetzung mit Hilfe von Keycloak möglich. Hierbei wird die aktuelle Lösung verändert und entsprechend angepasst:

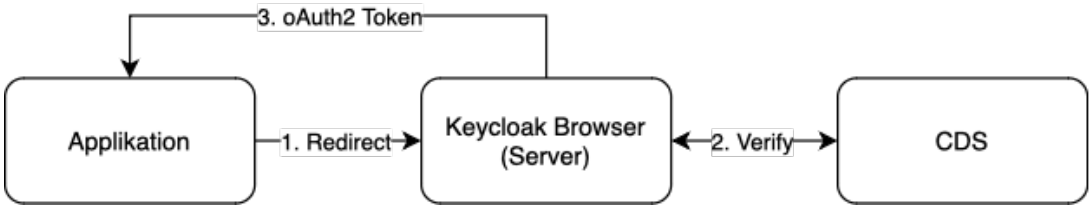

Abbildung 3.4: Veränderter Keycloak-Login

Statt bei einer Anmeldung einen Client zu simulieren bietet Keycloak die Möglichkeit eine Anmeldung über eine Nutzeroberfläche zu realisieren. Dafür wird ein

redirect auf die Keycloak-Login-Seite durchgeführt. Bei einer erfolgreichen Verifizierung wird der Nutzer zurück auf die Applikation geleitet und vom Keycloak-Server mit Hilfe eines oAuth2-Tokens authentifiziert.

Um eine Passwortlose Authentifizierung in Keycloak zu ermöglichen, muss der Authentication Flow für eine Browser-Anmeldung modifiziert werden. Der angepasste Authentication Flow besteht aus folgenden Schritten:

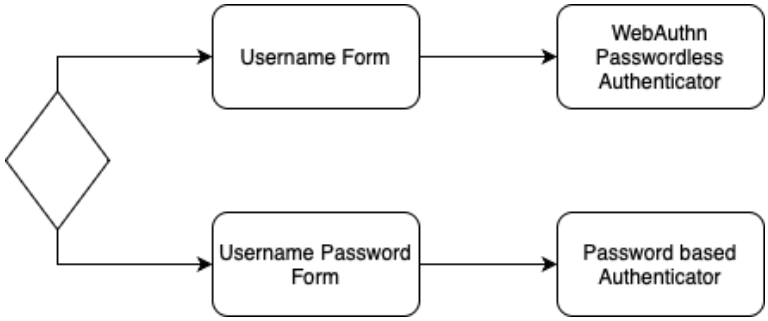

Abbildung 3.5: Authentication Flow

Hierbei wird die untere Hälfte der Grafik weiterhin ermöglicht, da es sich lediglich um einen Test handelt. Grundsätzlich wird diese nicht ermöglicht, da Keycloak eine reine passwortlose SFA unterstützt.

Die obere Hälfte der Grafik entspricht dem für diese Arbeit relevanten Authentication Flow. Dabei wird der User zunächst aufgefordert seinen Nutzernamen einzugeben und anschließend seinen Security Key zu verwenden. Dies ist notwendig, um die Nutzung eines Security Keys für mehrere Zugänge zu ermöglichen. Ermöglicht man lediglich die Nutzung eines Zugangs pro Security Key, so wird die Eingabe des Nutzernamens nicht benötigt. Zusätzlich erfolgt eine Konfiguration des Keycloak-Servers, welche die Registrierung eines Security Keys bei der Registrierung eines neuen Nutzers ermöglicht.

Mit Hilfe dieser Konfiguration werden zwei Abläufe ermöglicht. Eine neue Registrierung:

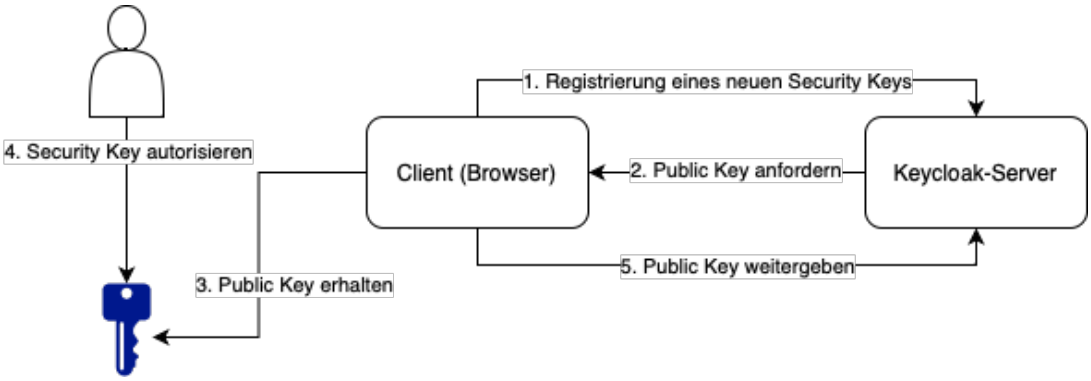

Abbildung 3.6: Registrierung (vereinfacht)

Sowie eine neue Anmeldung:

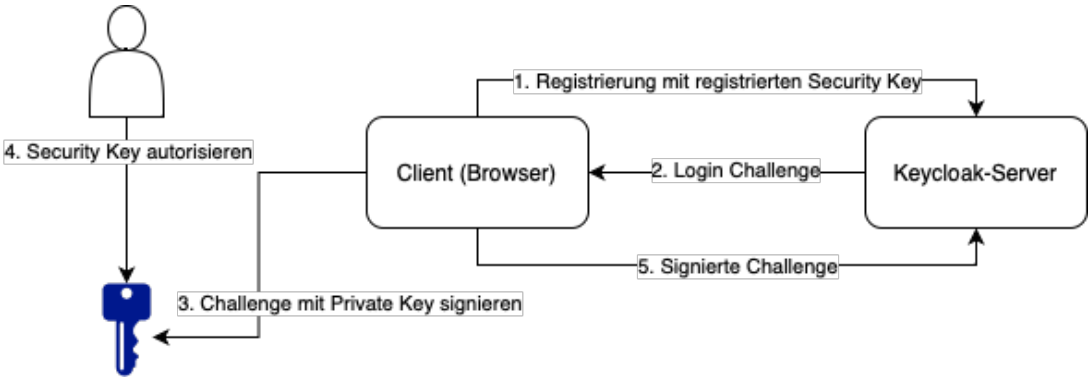

Abbildung 3.7: Anmeldung (vereinfacht)

Die Grafiken stellen den vereinfachten Ablauf der Registrierung und Anmeldung mit Hilfe eines Security Keys dar. Die detailierte Darstellung der Funktionsweise ist in Kapitel 2.7 zu finden. Der entscheidende Unterschied der beiden Prozesse ist allerdings, dass bei der Registrierung lediglich der öffentliche Schlüssel übergeben wird, während bei der Anmeldung der private Schlüssel benötigt wird. Dieser wird allerdings nicht übergeben, sondern signiert eine Login Challenge, welche vom Keycloak-Server generiert wird. Kann der Keycloak-Server die Signatur mit Hilfe des gespeicherten öffentlichen Schlüssels verifizieren, wird der Nutzer authentifiziert. Sowohl die Registrierung als auch die Anmeldung erfolgen hierbei also nicht über die Anwendung selbst, sondern über den Keycloak-Server und dessen Nutzeroberfläche.

3.3 User Feedback

3.3.1 Rahmen des Feedbacks

Um eine Aussage über die Akzeptanz eines passwortlosen Verfahrens innerhalb der LSY zu treffen wird ein interaktiver Fragebogen erstellt. Dieser wird in der Abteilung cGroup Solutions, welche zuständig für das in Kapitel beschriebene Produkt cFront ist, verteilt. Es handelt sich dabei um eine Abteilung mit 15 Personen.

Für die Arbeit wurde der Anmeldevorgang für das Produkt cFront wie in Kapitel beschrieben angepasst. Die Teilnehmer des Fragebogens wurden an einem Tag befragt. Alle Teilnehmer bekamen vor der Befragung eine Demonstration der Registrierung und Anmeldung mit Hilfe eines Security Keys, sowie eine Demonstration einer möglichen Anmeldung mit Hilfe eines Passkeys. Für die Demonstration wurde ein *Yubikey Series 5 NFC* als Security Key verwendet. Während der Befragung wurden den Teilnehmern keine Informationen über die Funktionsweise oder die technischen Details des Fido2 Protokolls gegeben. Während der Demonstration und der Durchführung des Fragebogens erhielten alle Teilnehmer ebenfalls die Möglichkeit Kommentare zu hinterlassen, welche ebenfalls auf dem Fragebögen festgehalten wurden. Die Ergebnisse des Fragebogens sind im Anhang zu finden.

3.3.2 Auswahl der Teilnehmer

Zur Durchführung des Fragebogens wurden alle Mitglieder des Teams eingeladen, eine Teilnahme war jedoch freiwillig. Zwei der Mitglieder der Abteilung konnten auf Grund eines Urlaubs nicht an der Befragung teilnehmen. Vor der Durchführung wurden alle Teilnehmer darüber informiert zu welchem Zweck die Daten für diese Arbeit erhoben werden. Die Befragung stand dabei nicht anonym statt, um einen Austausch zwischen dem Autor und den Teilnehmern zu ermöglichen. Da die Befragung die Abteilung der Teilnehmer betrifft sollten diese somit eine Möglichkeit bekommen, ihre Gedanken zu dem modifiziertem Anmeldevorgang zu teilen.

3.3.3 Inhalt der Demonstration

Allen Teilnehmern wurde vor der Befragung eine Live-Demonstration der Registrierung und Anmeldung mit Hilfe eines Security Keys gezeigt. Der Security Key wurde zu Beginn der Demonstration in einen üblichen USB-Slot eines Firmenlaptops eingesteckt und nach der Demonstration an die Teilnehmer übergeben. Diese sind in mehrere Schritte unterteilt. Zunächst bestätigt der Nutzer, dass er sich mit Hilfe eines Security Keys anmelden/registrieren möchte:

Abbildung 3.8: Veränderter Keycloak-Login

Darauf folgt ein Dialogfeld des Browsers, welcher den Nutzer dazu auffordert zu bestätigen, dass der Security Key registriert wird. Dieser Schritt ist einmalig und findet nur bei der Registrierung statt. Ist der Security Key bereits registriert, wird dieser Schritt übersprungen:

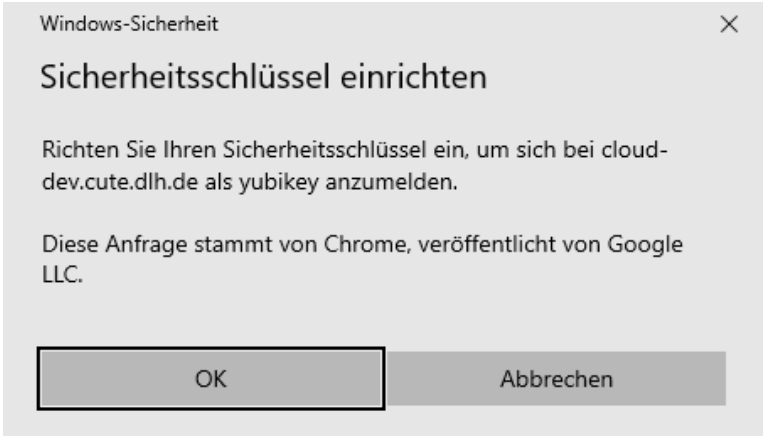

Abbildung 3.9: Veränderter Keycloak-Login

Nach der Bestätigung des Dialogs muss der Nutzer den PIN des Security Keys eingeben:

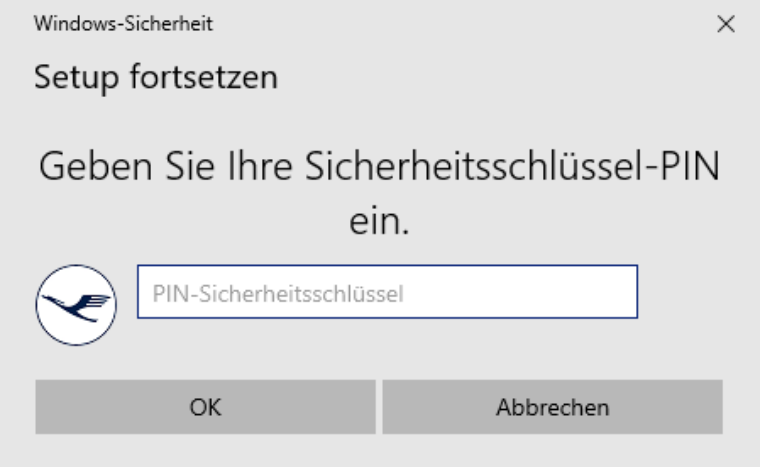

Abbildung 3.10: Veränderter Keycloak-Login

Ist die richtige PIN eingegeben wurden, erscheint ein letztes Fenster, welches den Nutzer dazu auffordert den Knopf des Security Keys zu drücken. Erst danach ist der Browser dazu autorisert sich mit Hilfe des Security Keys gegen den Keycloak-Server zu registrieren oder anzumelden:

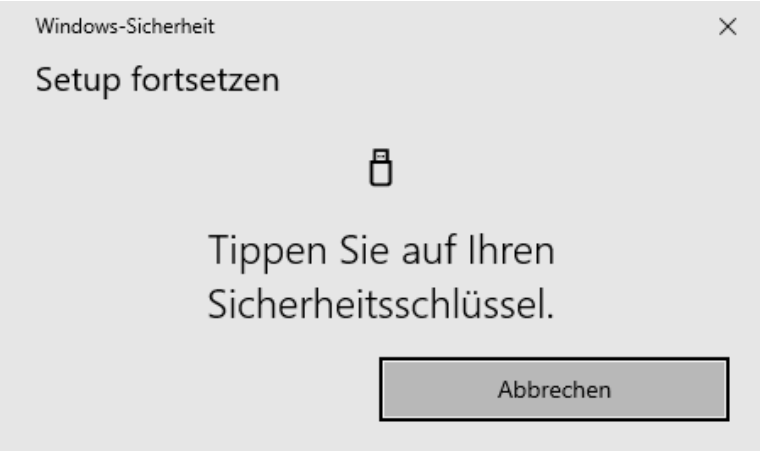

Abbildung 3.11: Veränderter Keycloak-Login

Sobald der Knopfdruck erfolgt, wird der Nutzer erfolgreich eingeloggt. Diese Informationen wurden den Teilnehmern ebenfalls während der Durchführung des Fragebogens mitgeteilt.

- Fragebogen:

- How old are you?
- What is your role within the team?
- Wie zufrieden bist du mit der registrierung? (1-5)
- Wie zufrieden bist du mit der Anmeldung? (1-5)
- Findest du die passkey variante besser als einen yubikey?
- Have you ever used a security key before? Yes, and I still do - Yes, but I stopped using it - No - I don't know
- If Yes in which context? (private - work - both)
- Do you currently use a password manager at work? (Yes, I use it all/most of the time - Yes, but I only use it sometimes - No)
- Do you know how the Fido2 protocol works? (Yes - No - I don't know)
- Would you pay for a security key (about 50€)? (Yes - No - I don't know)
- Do you think security keys are more secure than passwords? (Yes - No - I don't know)
- Comments:

3.4 Wirtschaftlichkeit

3.5 Nutzung des passwortlosen Verfahrens im privaten Kontext

Literaturverzeichnis

[1] S. Samonas und D. Coss, „The CIA strikes back: Redefining confidentiality, integrity and availability in security.," *Journal of Information System Security*, Jg. 10, Nr. 3, 2014.

[2] A. Agarwal und A. Agarwal, „The security risks associated with cloud computing," *International Journal of Computer Applications in Engineering Sciences*, Jg. 1, Nr. Special Issue on, S. 257–259, 2011.

[3] S. Boonkrong, „Security of passwords," *Information Technology Journal*, Jg. 8, Nr. 2, S. 112–117, 2012.

[4] K. Chanda, „Password security: an analysis of password strengths and vulnerabilities," *International Journal of Computer Network and Information Security*, Jg. 8, Nr. 7, S. 23, 2016.

[5] M. Yıldırım und I. Mackie, „Encouraging users to improve password security and memorability," *International Journal of Information Security*, Jg. 18, S. 741–759, 2019.

[6] F. M. Farke, L. Lorenz, T. Schnitzler, P. Markert und M. Dürmuth, „{"You} still use the password after {all"}–Exploring {FIDO2} Security Keys in a Small Company," in *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*, 2020, S. 19–35.

[7] M. Barbosa, A. Boldyreva, S. Chen und B. Warinschi, „Provable security analysis of FIDO2," in *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part III 41*, Springer, 2021, S. 125–156.

[8] M. Morii, H. Tanioka, K. Ohira et al., „Research on integrated authentication using passwordless authentication method," in *2017 IEEE 41st annual computer software and applications conference (COMPSAC)*, IEEE, Bd. 1, 2017, S. 682–685.

[9] B. Ives, K. R. Walsh und H. Schneider, „The domino effect of password reuse," *Communications of the ACM*, Jg. 47, Nr. 4, S. 75–78, 2004.

[10] R. S. Chowhan und R. Tanwar, „Password-less authentication: methods for user verification and identification to login securely over remote sites," in *Machine Learning and Cognitive Science Applications in Cyber Security*, IGI global, 2019, S. 190–212.

[11] V. Parmar, H. A. Sanghvi, R. H. Patel und A. S. Pandya, „A comprehensive study on passwordless authentication," in *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, IEEE, 2022, S. 1266–1275.

[12] J. Reynolds, T. Smith, K. Reese, L. Dickinson, S. Ruoti und K. Seamons, „A tale of two studies: The best and worst of yubikey usability," in *2018 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2018, S. 872–888.

[13] S. G. Lyastani, M. Schilling, M. Neumayr, M. Backes und S. Bugiel, „Is FIDO2 the kingslayer of user authentication? A comparative usability study of FIDO2 passwordless authentication," in *2020 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2020, S. 268–285.

[14] N. Bindel, C. Cremers und M. Zhao, „FIDO2, CTAP 2.1, and WebAuthn 2: Provable security and post-quantum instantiation," *Cryptology ePrint Archive*, 2022.