JUSTUS-LIEBIG-UNIVERSITÄT-GIESSEN
ALLG. BWL UND WIRTSCHAFTSINFORMATIK
UNIV.-PROF. DR. AXEL SCHWICKERT

Schwickert, Axel; Schramm, Laura; Schick, Lukas;
Dörr, Lea

Authentifizierungsmethoden – Reader zur WBT-Serie

ARBEITSPAPIERE WIRTSCHAFTSINFORMATIK

Nr. 11 / 2021
ISSN 1613-6667

Arbeitspapiere WI Nr. 11 / 2021

Autoren: Schwickert, Axel; Schramm, Laura; Schick, Lukas; Dörr, Lea

Titel: Authentifizierungsmethoden – Reader zur WBT-Serie

Zitation: Schwickert, Axel; Schramm, Laura; Schick, Lukas; Dörr, Lea: Authentifizierungsmethoden – Reader zur WBT-Serie, in: Arbeitspapiere WI, Nr. 11/2021, Hrsg.: Professur BWL – Wirtschaftsinformatik, Justus-Liebig-Universität Gießen 2021, 34 Seiten, ISSN 1613-6667.

Kurzfassung: Die vorliegende WBT-Serie erläutert die Authentifizierungsmethoden OpenID Connect und FIDO2.

Zunächst werden die Grundlagen von Authentifizierungsmethoden und deren Funktionsweisen beschrieben. Anschließend werden die gängigen Authentifizierungsmethoden für Web-Anwendungen vorgestellt und deren Vorteile und Risiken besprochen.

Die Erläuterung der Authentifizierungsmethoden OpenID Connect und FIDO2 erfolgt nachfolgend typgleich. Zunächst werden technische Grundlagen beschrieben. Anschließend werden Akteure, Schnittstellen und Abläufe der Registrierung sowie der Authentifizierung besprochen. Abschließend erflogt eine Darstellung der jeweiligen Vor- und Nachteile.

Schlüsselwörter: Authentifizierung, OpenID Connect, FIDO2, Verschlüsselung,

A Zur Einordnung der WBT-Serie

Die WBT-Serie richtet sich an Interessenten des Themenbereiches „Authentifizierungsmethoden“.

Für Ihr Selbststudium per WBT müssen Sie einen Internet-Zugang haben – entweder auf Ihren eigenen PCs, auf den PCs im JLU-Hochschulrechenzentrum, in den JLU-Bibliotheken oder dem PC-Pool des Fachbereichs.

B Die Web-Based Trainings

Der Stoff zu diesem Thema ist in Lerneinheiten zerlegt worden und wird durch eine Serie von Web-Based-Trainings (WBT) vermittelt. Mit Hilfe der WBT kann der Stoff im Eigenstudium erarbeitet werden. Die WBT bauen inhaltlich aufeinander auf und sollten in der angegebenen Reihenfolge absolviert werden.

WBT-Nr.	WBT-Bezeichnung	Bearbeitungs-dauer
1	Grundlagen zu Authentifizierungsmethoden	90 Min.
2	Authentifizierungsmethode OpenID Connect	90 Min.
3	Authentifizierungsmethode FIDO2	90 Min.

Tab. 1: Übersicht WBT-Serie

Die Inhalte der einzelnen WBT werden nachfolgend in diesem Dokument gezeigt. Alle WBT stehen Ihnen rund um die Uhr online zur Verfügung. Sie können jedes WBT beliebig oft durcharbeiten. In jedem WBT sind enthalten:

- Vermittlung des Lernstoffes,
- interaktive Übungen zum Lernstoff,
- abschließende Tests zum Lernstoff

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Seite

1 Grundlagen zu Authentifizierungsmethoden

1.1 Grundlagen der Datenübermittlung im Client-Server-Konzept

1.1.1 Einleitung

Für Unternehmen ist es zunehmend relevant, dass ihre Mitarbeiter und Kunden ortsungebunden über das Internet auf bestimmte Software-Systeme (sog. „Web-Anwendungen“) des Unternehmens zugreifen können.

Web-Anwendungen sind Software-Systeme, die auf einem Web-Server betrieben werden und per Web-Browser bedienbar sind.

Endkunden können über die Web-Anwendung „Web-Shop“ in einem Unternehmen einkaufen. Eine typische Web-Anwendung für einen Mitarbeiter eines Unternehmens kann der Web-Zugriff auf das CRM-System des Unternehmens sein.

Eine speziellere Web-Anwendung aus der produzierenden Industrie ist, dass Mitarbeiter bestimmte Maschinen aus der Ferne über das Internet steuern können. Schauen wir uns das am Besten am Beispiel der Siemex AG an.

1.1.2 Die Siemex AG

Die Siemex AG ist als produzierendes und verarbeitendes Unternehmen tätig und stellt unter anderem Fertigungsmaschinen für Industrie-Kunden her.

Der Hauptsitz der Siemex AG liegt zwar in Deutschland, jedoch agiert das Unternehmen international und hat daher Kunden und Mitarbeiter auf der ganzen Welt.

Die IT-Abteilung der Siemex AG entwickelt unter anderem Web-Applikationen für Industrie-Kunden. Mithilfe dieser Web-Applikationen (synonym „Web-Anwendungen“) lassen sich beispielsweise die Fertigungsanlagen bei Kunden bis ins kleinste Detail aus der Ferne kontrollieren, steuern und warten.

Dabei muss die Siemex AG sicherstellen, dass nur berechtigte Mitarbeiter und Kunden auf bestimmte Web-Anwendung zugreifen können.

Dazu erfordern Web-Anwendungen die Authentifizierung des Benutzers (User).,

1.1.3 Web-Applikationen der Siemex AG

Web-Applikationen, wie auch die Siemex AG sie entwickelt, sind Software-Systeme, die auf einem Web-Server betrieben werden und über das Internet per Web-Browser vom Mitarbeiter (Client) bedient werden.

Web-Applikationen basieren auf der Funktionalität des Client-Server-Konzepts.

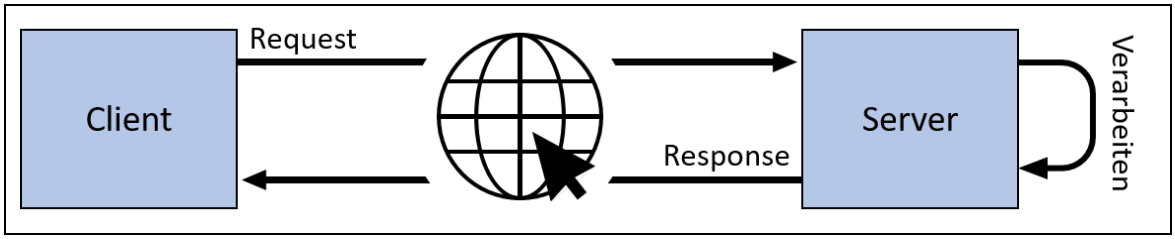

Abb. 1: Das Client-Server-Konzept

1.1.4 Komponenten im Client-Server-Konzept

Das Client-Server-Konzept setzt voraus, dass es mindestens zwei verteilte Komponenten in einem System gibt:

- Mindestens eine Komponente („Server“) bietet über Schnittstellen Dienste an und
- mindestens eine Komponente („Client“) fragt Dienste von Servern ab.

1.1.5 Kommunikation im Client-Server-Konzept

Die Kommunikation im Cient-Server-Konzept basiert auf Nachrichten.

Der Client schickt einen „Request“ („Anfrage-Nachricht“) an den Server, der die geforderte Ressource oder den geforderten Dienst eindeutig beschreibt.

Der Server verarbeitet den Request.

Anschließend antwortet der Server mit einer passenden „Response“ („Antwort-Nachricht“), die die geforderte Ressource oder den geforderten Dienst liefert.

1.1.6 Aufruf der Web-Applikation der Siemex AG

Ein typisches Beispiel für den Einsatz des Client-Server-Konzepts ist das Aufrufen einer Web-Seite im Web-Browser. Am besten schauen wir uns die Funktionsweise Schritt für Schritt an:

1. Eingabe der URI

Der User/Mitarbeiter gibt zuerst eine URI (Uniform Resource Identifier) in einem Web-Browser (Client) ein. Eine URI ist eine eindeutige Adresse für eine Ressource im Internet, die einen Server und den Endpunkt auf diesem Server spezifiziert.

Der Endpunkt kann dabei als Server-interne Adresse verstanden werden. Über eine URI lässt sich ein bestimmter Dienst oder eine bestimmte Ressource abfragen. Eine Untergruppe von URI sind die URLs (Uniform Resource Locator).

Eine URL spezifiziert neben der Position der Ressource auch die Art, wie die Ressource zu erreichen ist (z. B. das Protokoll).

2. Web-Browser sendet Request an Web-Server

Wenn der User einen URI auf der Tastatur eingibt und „Enter“ drückt, schickt der Web-Browser (Client) einen Request an die angegebene URI über das offene Internet.

3. Web-Server verarbeitet Anfrage

Der Web-Server (Server), dem die URI zugeordnet werden kann, verarbeitet nun den Request und sucht in seinem Dateisystem nach der Web-Seite (Ressource), die über den angegebenen Endpunkt erreichbar ist.

4. Web-Server schickt Response an Web-Browser

Der Web-Server schickt die Web-Seite als Response über das offene Internet an den Web-Browser des Users (Client). Dort wird die Benutzeroberfläche der Web Site angezeigt.

1.1.7 Anforderungen an die Web-Applikation der Siemex AG

Okay, jetzt ist mir im Groben klar, wie die Kommunikation mit Web-Applikationen auf Basis des Client-Server-Konzeptes funktioniert.

Die IT-Abteilung der Siemex AG hat die entsprechenden Web-Applikationen in Eigenregie entwickelt. So ist genau der Funktionsumfang abgebildet, der zur Fernsteuerung der Fertigungsmaschinen benötigt wird.

Damit die Maschinen jedoch nicht von unbefugten Dritten gesteuert werden können, enthalten die Web-Applikationen umfassende Schutz- und Sicherheitsmechanismen.

1.1.8 Authentifizierung der Mitarbeiter an der Web-Applikation

Genau! Eine zentrale Sicherheitsmaßnahme innerhalb der Web-Applikation ist, dass nur berechtigte Mitarbeiter auf diese zugreifen können.

Dazu bedarf es der eindeutigen Authentifizierung des Mitarbeiters.

Die Authentisierung stellt den Nachweis einer Person dar, dass sie tatsächlich diese Person ist.

Als Authentifizierung wird der Vorgang bezeichnet, bei dem die behauptete Authentisierung überprüft wird, also ob es sich bei einer Person oder einer Repräsentation einer Person (wie z. B. Benutzer-Daten) um eine bekannte, echte Identität handelt.

Weibliche Person:

> „Ich bin eine berechtigte Mitarbeiterin. Um mich in der Web-Applikation zu authentisieren, gebe ich meine Benutzer-Daten (z. B. Benutzername und Passwort) ein.
>
> Die gängigen Authentifizierungsmethoden gucken wir uns im nächsten Kapitel genauer an!“

1.2 Faktoren der Authentifizierung

1.2.1 Einleitung

Männliche Person:

> „Okay, wir haben gelernt, dass als Authentifizierung der Vorgang bezeichnet wird, bei dem nachgewiesen werden soll, ob es sich bei einer Person um eine bekannte, echte Identität handelt.
>
> So soll sichergestellt werden, dass nur berechtigte Mitarbeiter auf eine Web-Applikation der Siemex AG zugreifen können.“

Weibliche Person:

> „Ok, aber wie kann ein System überprüfen und verifizieren, ob jemand berechtigt ist?
>
> Das schauen wir uns besser nochmal genauer an.“

1.2.2 Authentifizierung durch Benutzer-Daten

Damit ein System verifizieren kann, ob eine Person zugangsberechtigt ist, müssen dem System Informationen mitgeteilt werden, über die nur eine berechtigte Person verfügt.

Meistens werden dazu die Benutzer-Daten abgefragt.

Die Benutzer-Daten einer Person bestehen i. d. R. aus einem Benutzer-Namen (User Name) und einem zugehörigen Benutzer-Passwort (User Password).

Diese Benutzer-Daten repräsentieren ein Benutzer-Konto (User Account).

1.2.3 Faktoren der Authentifizierung

Die Verwendung einer Kombination aus einem Benutzernamen und dem Faktor „Passwort“ ist jedoch nur eine Möglichkeit zur Authentifizierung.

Um die Identität gegenüber der Web-Applikation zu bestätigen, können folgende Faktoren mit einem Benutzer-Namen kombiniert werden:

- **Information**

  Berechtige Nutzer haben spezifisches Wissen. Zum Beispiel kennen sie ein geheimes Passwort zu einem Nutzernamen.

- **Besitz**

  Berechtige Nutzer sind im Besitz von spezifischer Hardware. Zum Beispiel besitzen sie einen Security-Token mit USB-Anschluss.

- **Biometrie**

  Berechtigte Nutzer erhalten Zugriff basierend auf ihren persönlichen und einzigartigen Körpermerkmalen. Dies kann z. B. mithilfe eines Fingerabdruck- oder Iris-Scans erfolgen.

1.2.4 1-Faktor-Authentifizierung

Zahlreiche Web-Applikationen nutzen derzeit die sogenannte 1-Faktor-Authentifizierung, bei der nur der Faktor „Information“ zur Authentifizierung eines Benutzers eingesetzt wird.

Dafür gibt der Benutzer seine Benutzer-Daten (Name und Passwort) in einen Login-Bildschirm im Web-Browser ein und der Server der Web-Applikation prüft, ob für diese Benutzer-Daten der Zugriff auf die Web-Applikation erlaubt ist oder nicht.

Im Falle dieser 1-Faktor-Authentifizierung ist der einzig eingesetzte Faktor die Information, bestehend aus Benutzername (z. B. der E-Mail-Adresse) und Passwort.

1.2.5 Risiken der 1-Faktor-Authentifizierung

Die Authentifizierung mithilfe des einzigen Faktors Information (Benutzername und Passwort) gilt als unsicher und ungeeignet für eine Authentifizierung an sicherheitsrelevanten Systemen.

Werden Passwörter oder Passwortrichtlinien nicht vorgegeben, setzen Nutzer häufig unsichere Passwörter oder verwenden die gleiche Kombination aus Benutzername und Passwort für unterschiedliche Dienste.

Wenn diese Kombination aus Benutzername (oftmals die E-Mail-Adresse) und Passwort einem Dritten bekannt wird, kann sich dieser ohne weitere Hürde authentifizieren.

Phising, Malware und Social Engineering haben bei der 1-Faktor-Authentifizierung große Erfolgschancen.

1.2.6 2-Faktor-Authentifizierung

Um die Zuverlässigkeit einer Benutzer-Authentifizierung zu erhöhen, wird im Rahmen von Authentifizierungsmethoden immer öfter ein zweiter Faktor eingesetzt.

Im Falle der 2-Faktor-Authentifizierung findet eine Kombination aus mehreren Faktoren statt. Meist wird der Faktor Information mit einer weiteren Information, wie einer TAN oder SMS, kombiniert.

Nachdem ein Benutzer beim Login seine Benutzer-Daten erfasst hat, verlangt der Web-Server vom Benutzer die Eingabe weiterer Identifikationsangaben.

Dazu schickt der Web-Server z. B. eine TAN (Transaktionsnummer) per SMS-Nachricht (Short Message Service) an das Smartphone des Benutzers. Der Benutzer muss diese TAN dann ebenfalls im Login-Bildschirm eingeben, um Zugang zur Web-Applikation zu erhalten.

Immer öfter kommen auch die Faktoren Besitz und Biometrie zum Einsatz.

1.2.7 Vorteile der 2-Faktor-Authentifizierung

Eine Authentifizierung über einen zweiten Faktor ist somit um einiges sicherer als die 1-Faktor-Authentifizierung.

1. Selbst bei einem Passwortverlust (1. Faktor) erhalten Dritte nur dann Zugriff, wenn sie ebenfalls im Besitz des 2. Faktors sind.
2. Der Einsatz von einfachen Passwörtern ist weniger riskant.
3. Benutzerdaten (Benutzername und Passwort) stellen alleine kein lohnendes Ziel für Angriffe dar.

1.2.8 Datenübermittlung bei der Authentifizierung

Die Authentifizierungsmethoden haben zwei wesentliche Eigenschaften gemein:

- Die Benutzer-Daten werden zwischen dem Web-Browser des Users und dem Web-Server über Netzwerkleitungen des offenen Internet übertragen.
- Die Benutzer-Daten müssen auf Seiten des Benutzers und des Servers gespeichert werden.

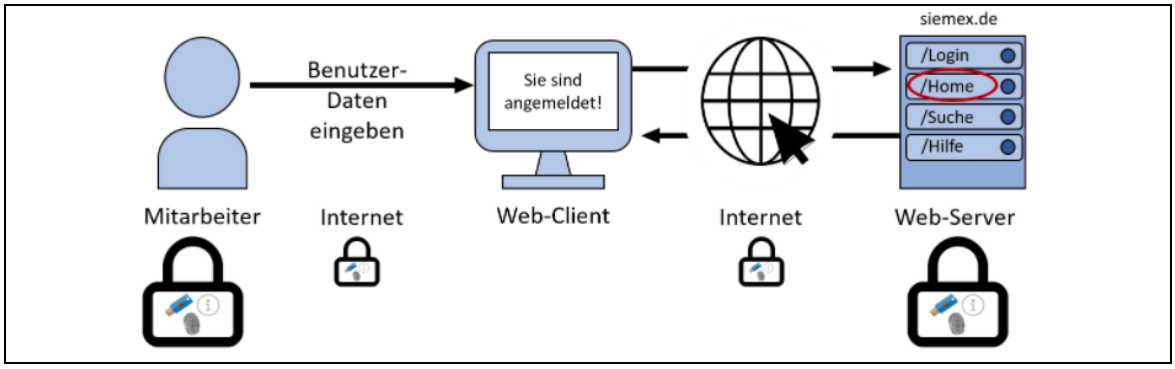

Abb. 2: Datenübermittlung bei der Authentifizierung

Die Benutzer-Daten müssen also beim User, beim Server und auf dem Weg dazwischen geschützt werden.

1.2.9 Sicherheitsprobleme bei der Authentifizierung

Genau diese beiden Eigenschaften (1. Speicherung und die 2. Übermittlung von Daten während der Authentifizierung) sind die Ursache für die allseits bekannten Sicherheitsprobleme von geschützten Web-Anwendungen im offenen Internet.

Während der Authentifizierung sind die Daten beim User, beim Server und auf dem Weg dazwischen zu schützen. Die Übermittlung der Daten wird über das offene Internet durchgeführt.

1. **Datenspeicher bei Client und Server**

   Per Phishing und Malware werden Benutzer-Daten von User-Rechnern abgegriffen. Durch gehackte oder kompromittierte Server können ganze Benutzer-Datenbanken in falsche Hände geraten.

2. **Datenübermittlung über das Internet**

   Das Abhören von Telekommunikation durch Unbefugte ist heute gang und gäbe. Auch eine Verschlüsselung aller kritischen Daten kann hier nur bedingt Abhilfe schaffen.

1.3 Einsatz-Szenarien

1.3.1 Einleitung

Die Fertigungsmaschinen beim Kunden der Siemex AG werden mit Hilfe von Web-Anwendungen (auf dem Web-Server vorgehalten) überwacht. Den Zugriff auf die Web-Anwendung durch Nutzer (User) muss Auth-Server („Authentifizierung“) erlauben.

Es gibt drei Szenarien zur Lokalisierung der beiden Server. Unabhängig vom Szenario befinden sich die Maschinen einer Fertigungsanlage immer beim Kunden. Die drei Szenarien schauen wir uns nachfolgend detailliert an.

1.3.2 Szenario 1: „On-Premises“

Bei Szenario 1 befinden sich der Web-Server zum Betreiben der Web-Anwendung und der Auth-Server beide vor Ort beim Kunden. Dieses Szenario wird „On-Premises“ („vor Ort“) genannt.

On-Premises-Web-Anwendungen haben den Vorteil, dass vertrauliche Daten nie den vom Kunden kontrollierten Bereich verlassen. Diesem Vorteil stehen u. a. die Nachteile der Kosten von On-Premises-Web-Anwendungen gegenüber. Die Anschaffungskosten für Hardware und Software sowie die Beschäftigung von Administrations-Personal schränken die Skalierbarkeit von On-Premises-Web-Anwendungen stark ein.

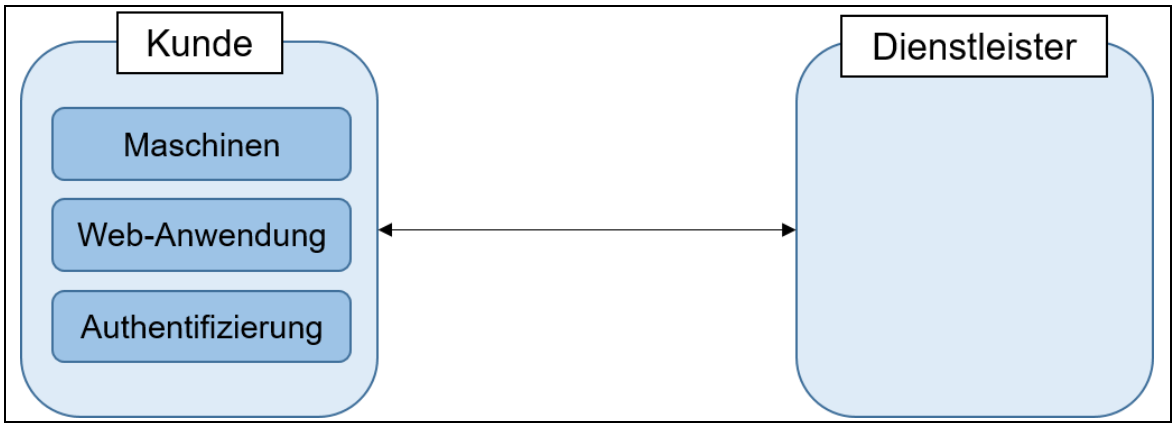

Abb. 3: Szenario 1: „On-Premises“

1.3.3 Szenario 2: „Cloud-Based“

Bei Szenario 2 befinden sich der Web-Server zum Hosten der Web-Anwendung und der Auth-Server beide nicht beim Kunden, sondern bei Dienstleistern. Die Dienstleister bieten dem Kunden die Web-Anwendung zur Nutzung über das offene Internet als Service an. Dieses Szenario wird „Cloud-Based“ genannt.

Cloud-Based-Web-Anwendungen sind skalierbar, da der Kunde selbst nicht in Infrastruktur investieren muss. Stattdessen werden die Dienste von Anbietern wie z. B. Amazon

Web Services oder Microsoft Azure nach Bedarf in Anspruch genommen. Allerdings laufen die Web-Anwendung und die Authentifizierung auf Kunden-externen Servern und somit liegen vertrauliche Daten auch auf Kunden-externen Servern.

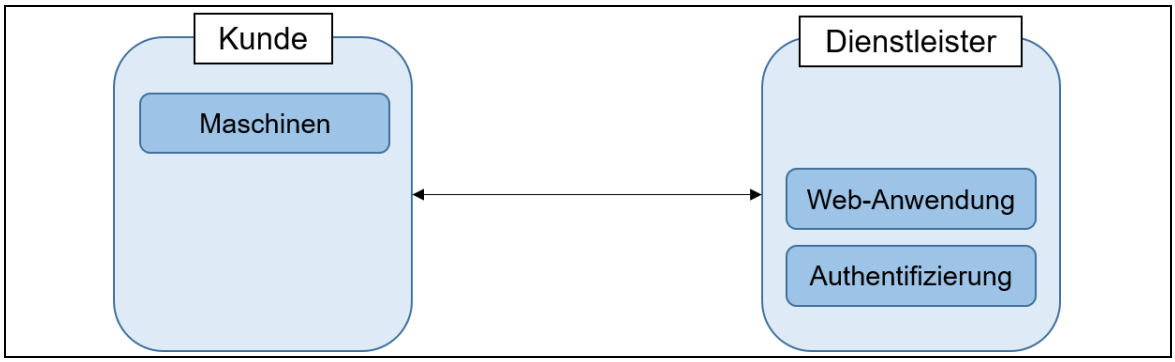

Abb. 4: Szenario 2: „Cloud-Based“

1.3.4 Szenario 3

Szenario 3 ist eine Mischung aus Szenario 1 und Szenario 2. Dabei befindet sich der Web-Server zum Betreiben der Web-Anwendung („Web-Anwendung“) beim Kunden und der Auth-Server („Authentifizierung“) bei einem Dienstleister.

In diesem Szenario legen Web-Anwendungen keine Maschinendaten der kundenseitigen Fertigungsanlage auf externen Servern ab. Nur die Authentifizierungs-Informationen müssen auf Kunden-externen Servern vorgehalten werden. Die Auslagerung der Authentifizierung reduziert den Administrationsaufwand auf Seiten des Kunden

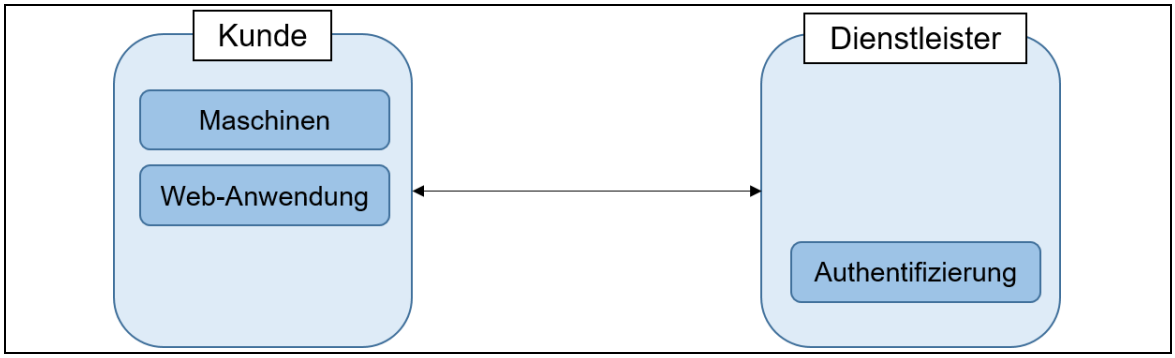

Abb. 5: Szenario 3

1.3.5 Szenario 2: „Cloud-Based“ in der Siemex AG

Von der Siemex AG entwickelte Web-Anwendungen werden auch von der Siemex AG betreut. So betreibt und administriert die IT-Abteilung der Siemex AG sowohl den Web-Server als auch den Auth-Server. Die Siemex AG nutzt also das „Cloud-Based“-Szenario.

Die Siemex AG bietet ihren Kunden die Nutzung der Web-Anwendung über das Internet als Service an.

1.3.6 Spezielle Authentifizierungsmethoden

Männliche Person:

„Okay, jetzt ist mir klar, wie die Kommunikation mit Web-Applikationen auf Basis des Client-Server-Konzeptes funktioniert.

Auch haben wir gelernt, welche unterschiedlichen Faktoren zur Authentifizierung eingesetzt werden können.

Abschließend wurde gezeigt, dass es drei Szenarien gibt, wo Web-Server und Auth-Server positioniert werden können. Damit sind die Grundlagen der Authentifizierungsmethoden nun bekannt. Darauf aufbauend schauen wir uns in den nächsten beiden WBT die Authentifizierungsmethoden „OpenID Connect“ und „FIDO2“ an.

1.4 Typische Aufgabenstellungen

1.4.1 Typische Aufgabenstellungen – Grundlagen zu Authentifizierungsmethoden

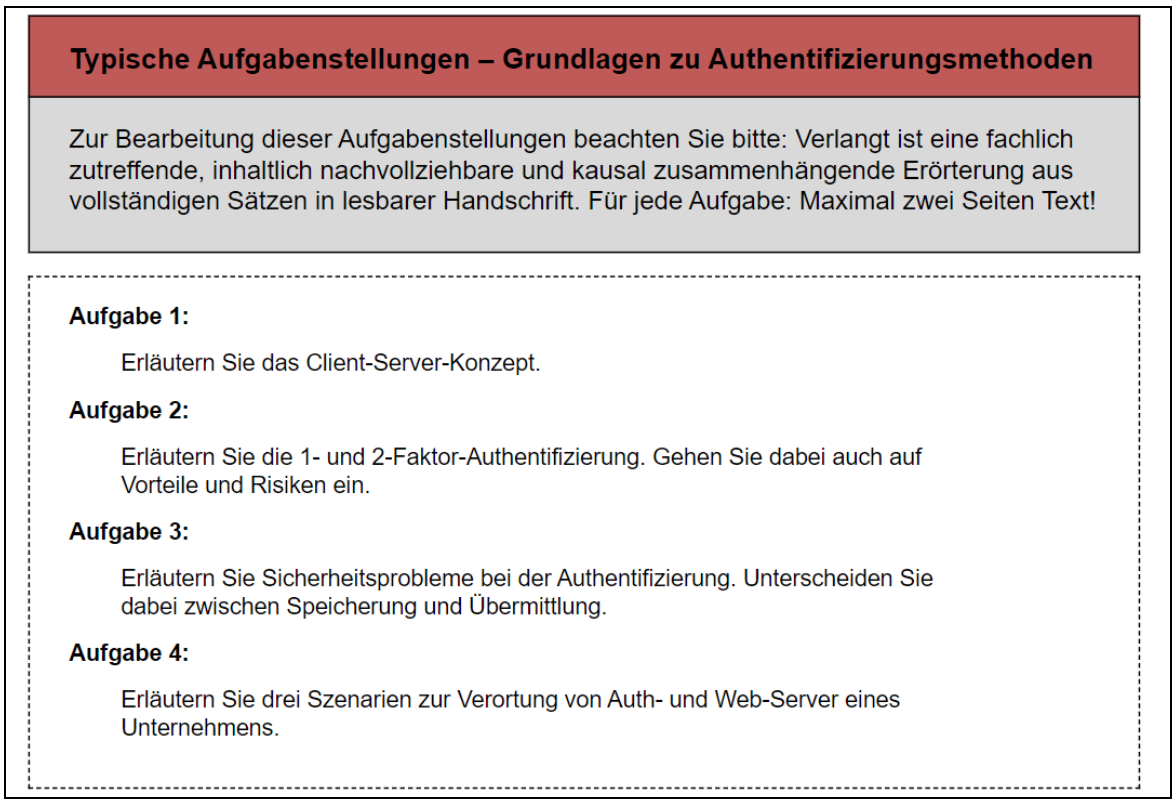

Abb. 6: Typische Aufgabenstellungen – Grundlagen zu Authentifizierungsmethoden

2 Authentifizierungsmethode OpenID Connect

2.1 Grundlagen von OpenID Connect

2.1.1 Zurück in der Siemex AG

Zuletzt haben wir über die Grundlagen von Authentifizierungsmethoden gesprochen.

Dabei haben wir gelernt, dass es bei der Speicherung und der Übermittlung von Nutzerdaten über das Internet zu Sicherheitsproblemen kommen kann.

Um diese Probleme zu lösen und Authentifizierungsvorgänge sicher zu gestalten, schauen wir uns die Authentifizierungsmethode „OpenID Connect“ an. Im anschließenden WBT widmen wir uns der Authentifizierungsmethode „FIDO2“.

Beides sind Lösungen zur konkreten Abwicklung eines Authentifizierungsvorgangs.

2.1.2 OpenID Connect

Wollen Sie einen Web-Dienst nutzen, müssen Sie sich bei diesem Web-Dienst mit Ihren Account-Informationen Kennung (Username) und Passwort anmelden (authentifizieren). Für jeden Web-Dienst benötigen Sie eine eigene Kennung mit dem zugehörigen Passwort.

OpenID Connect ist eine Authentifizierungslösung mit einem Intermediär, der für Sie die Anmeldungen an vielen verschiedenen Web-Diensten übernimmt. Derzeit bieten sich immer häufiger Apple, Google oder Facebook als solche Intermediäre an. Sie benötigen dazu einen Account bei diesem Intermediär: eine Apple-ID, einen Google- oder Facebook-Account. Und Sie müssen dem Intermediär mitteilen, dass Sie ihn als Intermediär für Anmeldevorgänge auf Web-Diensten nutzen wollen.

Zum Beispiel bietet Ihnen Spotify („Web-Dienst“) auf dem Anmeldebildschirm an, sich mit Apple, Google, Facebook oder Ihrem Spotify-Account anzumelden. Wenn Sie auf „Weiter mit Apple“ klicken, nutzt Spotify Ihre Apple-ID zur Anmeldung bei Spotify. Sie müssen sich somit nur die Account-Informationen Ihres Intermediärs merken, nicht unbedingt aber die einzelnen Account-Informationen von allen Ihren Web-Diensten.

Die meisten der Intermediäre arbeiten nach dem Funktionsprinzip von OpenID Connect (OIDC). Auf den folgenden Seiten wird Ihnen erklärt, wie OpenID Connect funktioniert.

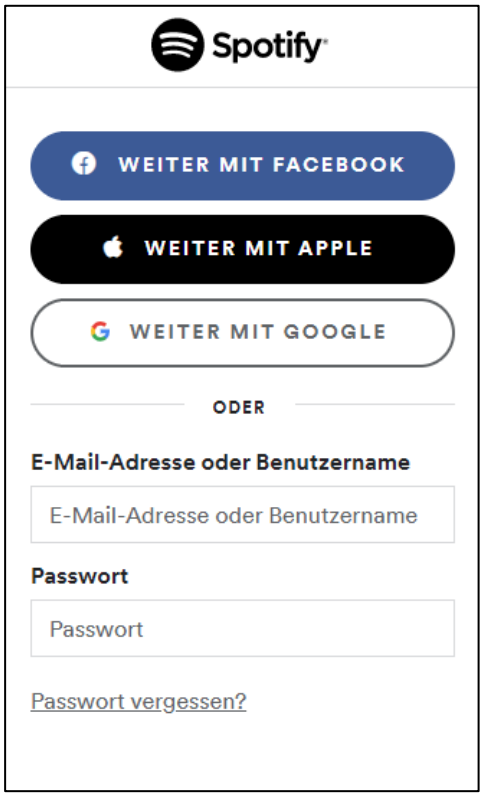

Abb. 7: Anmeldung bei Spotify

2.1.3 Die technische Grundlage: OAuth 2.0

Zur Realisierung von OpenID Connect wurde auf ein Protokoll mit dem Namen „OAuth 2.0“ (Open Authorization) zurückgegriffen.

OAuth wurde entwickelt, um eine sichere Kommunikation zwischen Anwendungen (App-to-App) auf Basis von APIs (Application Programming Interfaces) zu ermöglichen. APIs stellen die Schnittstellen vonAnwendungen dar.

Dieses Funktionsprinzip wurde in OpenID Connect übernommen. Open-ID Connect erweitert OAuth jedoch um eine Mensch-zu-Maschine-Schnittstelle. Dadurch wird eine Kommunikation zwischen Mensch und Anwendung ermöglicht.

OpenID Connect erweitert somit das Protokoll OAuth 2.0 um eine Kommunikation zwischen Client (Mensch) und Server (Web-Applikation). Es gibt des Weiteren vor, wann wer welche Daten übermittelt oder verarbeitet.

2.1.4 Die Erweiterung: OpenID Connect

OpenID Connect stellt jedoch noch viel mehr dar: OpenID Connect stellt eine Schnittstelle zwischen Mensch und Maschine bereit, die auch bei komplexeren Anwendungsfällen (bspw. E-Government, E-Health) eingesetzt werden kann. OpenID Connect unterstützt Web-Dienste bei der Implementierung des Authentifizierungssystems und stellt

eine Login-Schnittstelle für Menschen bereit. Es besteht keine Notwendigkeit zur Entwicklung proprietärer Authentifizierungsverfahren für Web-Dienste. Web-Dienste können stattdessen einsetzen, was OpenID Connect ihnen bietet.

Um genau verstehen zu können, wie ein solcher Authentifizierungsvorgang mithilfe einer vertrauenswürdigen Instanz („dem Intermediär“) vonstattengeht, werden im Folgenden Szenarien, Akteure, Abläufe und Einsatzbereiche von OpenID Connect im Rahmen der Siemex AG erläutert.

2.1.5 Szenarien zu OpenID Connect

Es existieren unterschiedliche Szenarien, wie OpenID Connect eingesetzt werden kann. Diese können nach der Lokalisierung von Authentifizierungs- und Web-Server unterschieden werden.

Im zweiten Szenario „Cloud-Based“ befinden sich der Web-Server zum Hosten der Web-Anwendung und der Auth-Server beide nicht beim Kunden, sondern bei Dienstleistern. Die Dienstleister bieten dem Kunden die Web-Anwendung zur Nutzung über das offene Internet als Service an.

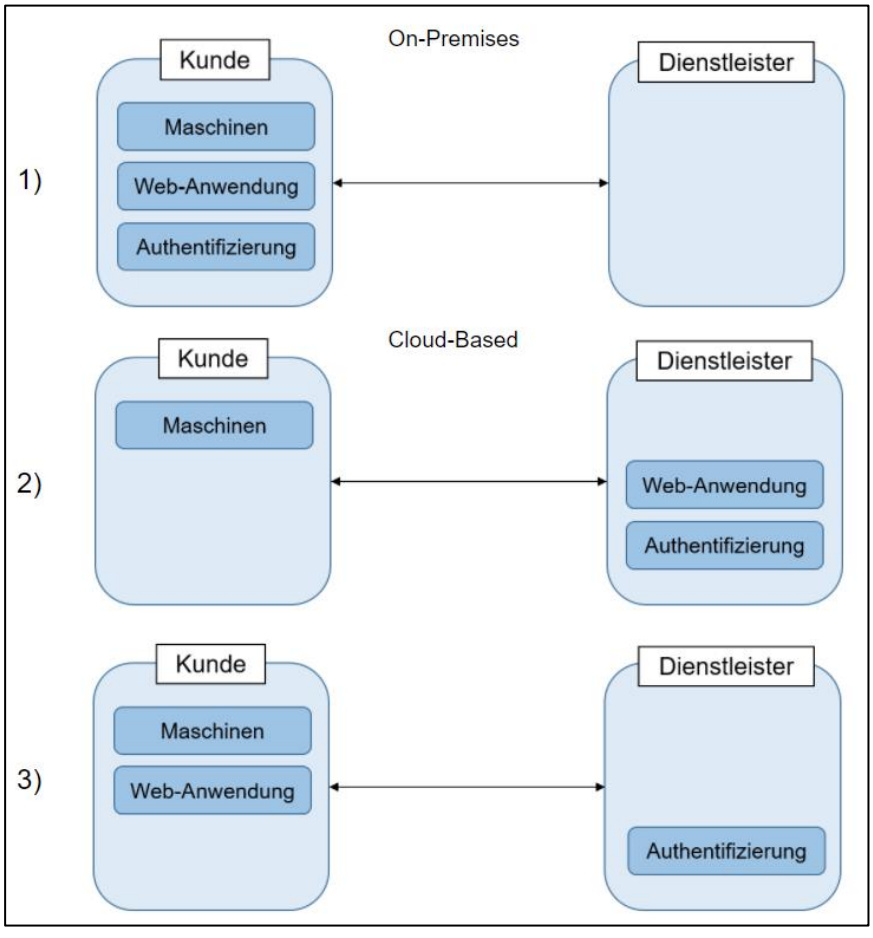

Abb. 8: Szenarien zu OpenID Connect

2.1.6 Szenario 2: „Cloud-Based“ in der Siemex AG

Wie bereits in WBT 1 aufgezeigt, entscheidet sich die IT-Abteilung der Siemex AG für das „Cloud-Based“-Szenario zum Einsatz von OpenID Connect.

Aber wie funktioniert OpenID Connect denn jetzt genau?

Um das zu verstehen, müssen wir uns zunächst die Akteure ansehen. Anschließend betrachten wir die Abläufe eines OpenID-Connect-Verfahrens.

Puh, jetzt wird es anspruchsvoll!

2.2 Ablauf der Authentifizierung in OpenID Connect

2.2.1 Akteure in OpenID Connect

Bevor wir uns den Ablauf der Authentifizierung hinter OpenID Connect anschauen können, müssen wir uns zuerst einen Überblick über die Akteure in OpenID Connect verschaffen.

- **User mit Web-Browser**

  Der User ist der Eigentümer seines Accounts und der damit verbundenen Daten.

  Er ist in der Lage, sich zu authentifizieren und ist autorisiert, auf die Maschinendaten zuzugreifen, die die Web-Anwendung liefert.

- **Web-Anwendung**

  Die Web-Anwendung besteht bei OIDC aus einem „Front-End“ und einem „Back-End“.

  Das Front-End ist die Benutzeroberfläche, die über einen Web-Browser abrufbar ist. Das Front-End wird nachfolgend „Client-Anwendung“ genannt.

  Das Back-End ist die Komponente der Web-Anwendung, die das Front-End ausliefert und auf die Daten von den angebundenen Fertigungsmaschinen zugreift.

- **Auth-Server**

  Der Auth-Server führt die Anmeldung der User und alle damit verbundenen Aktionen für die Web-Anwendung durch.

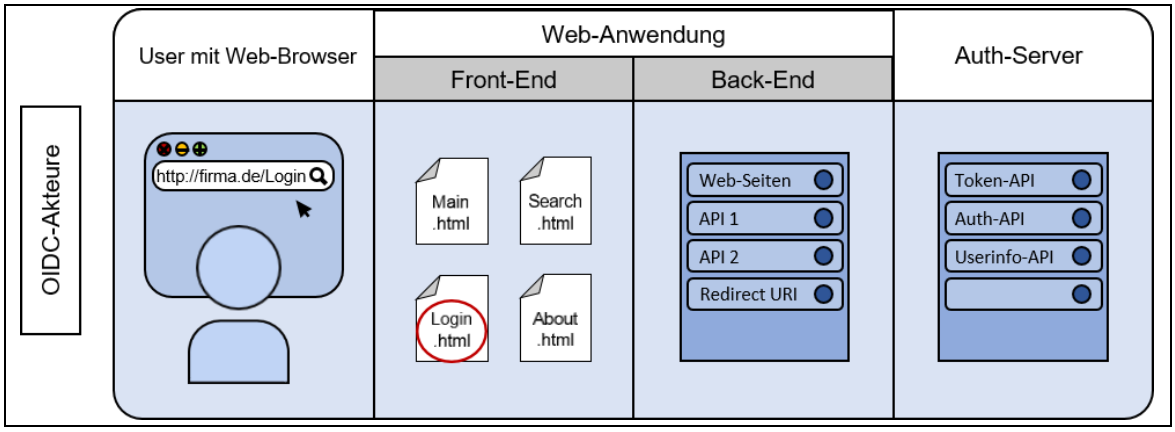

Abb. 9: Akteure in OpenID Connect

2.2.2 Abläufe in OpenID Connect

Super, jetzt kennen wir schon mal die beteiligten Akteure im Zusammenhang mit OpenID Connect. Jetzt bringen wir diese in Verbindung mit dem Prozess, den jeder Nutzer durchlaufen muss, um sich mit OpenID Connect bei einem Dienst erstmalig zu registrieren.

Dieser Prozess besteht aus sieben Schritten, die wir uns auf der nachfolgenden Seite im Detail ansehen.

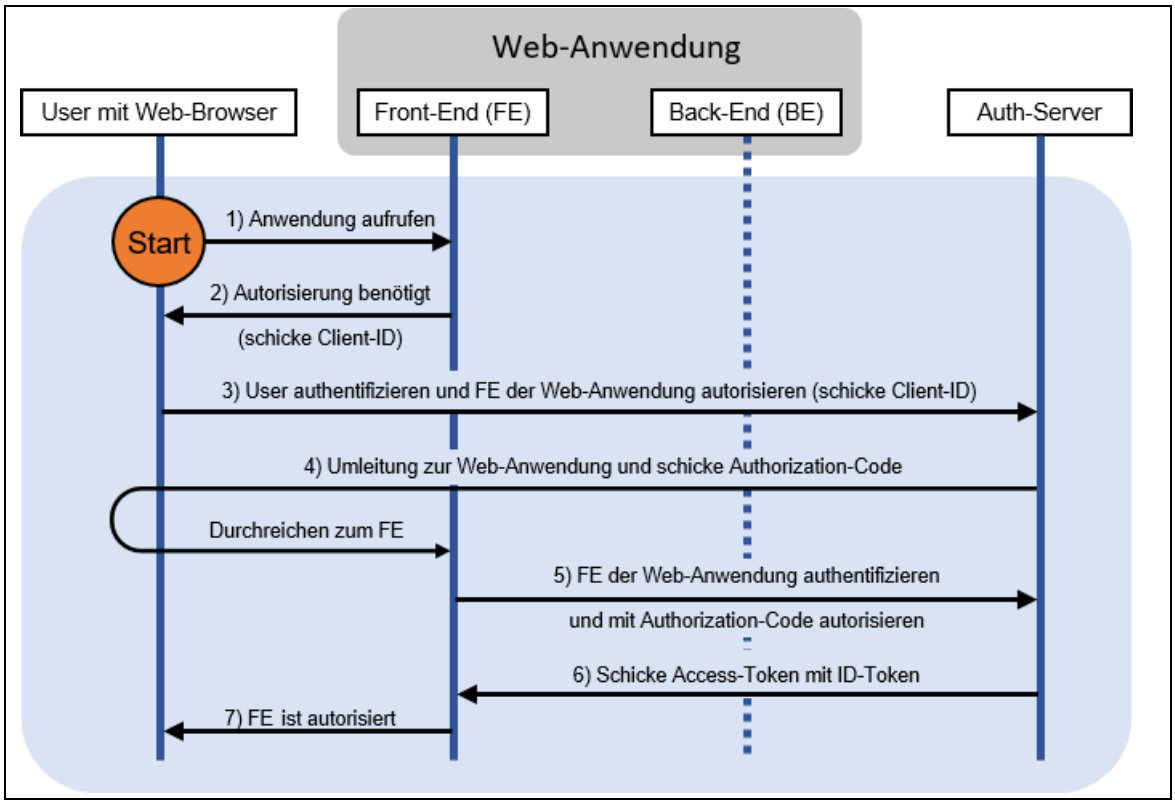

Abb. 10: Registrierungsprozess in OpenID Connect

2.2.3 Registrierungsprozess in OpenID Connect

- **Schritt 1: Anwendung aufrufen**

  Der User öffnet in seinem Web-Browser die Client-Anwendung (das Front-End).

- **Schritt 2: Autorisierung benötigt (schicke Client-ID)**

  Die Client-Anwendung soll auf das Back-End der Web-Anwendung zugreifen. Dafür muss der User die Client-Anwendung autorisieren. Damit diese Autorisierung stattfinden kann, muss der User sich bei der Web-Anwendung authentifizieren. Der Web-Browser des Users erhält dazu eine Client-ID (User-Name) und ein Client-Secret (Passwort) von der Client-Anwendung (Front-End).

- **Schritt 3: User authentifizieren und Client autorisieren (schicke Client-ID)**

  Der Web-Browser wird von der Client-Anwendung an den Auth-Server der Web-Anwendung weitergeleitet.

  Die Client-ID der Client-Anwendung wird an den Auth-Server geschickt, um mitzuteilen, welche Client-Anwendung auf die Web-Anwendung zugreifen will.

  Der User muss sich nun beim Auth-Server der Web-Anwendung mit Username und User-Passwort authentifizieren und den Zugriff der Client-Anwendung auf die Web-Anwendung autorisieren.

- **Schritt 4: Umleitung zum Client mit Authorization-Code**

  Der Auth-Server erstellt für die übergebene Client-ID einen Authorization-Code und der User wird zurück zur Client-Anwendung umgeleitet. Der Authorization-Code wird durch den Web-Browser des Users an die Client-Anwendung weitergeleitet.

- **Schritt 5: Client-Anmeldung mit Authorization-Code**

  Die Client-Anwendung sendet den Authorization-Code zusammen mit ihrer Client-ID und ihrem Client-Secret an den Auth-Server. Client-ID und Client-Secret authentifizieren die Client-Anwendung beim Auth--Server. Der mitgelieferte Authorization-Code bescheinigt dem Auth-Server die Autorisierung der Client-Anwendung durch den User zum Zugriff auf die Web-Anwendung.

- **Schritt 6: Access-Token**

  Wenn die Client-ID des Authorization-Codes mit der Client-ID der authentifizierten Client-Anwendung übereinstimmt und der Authorization-Code valide ist, stellt der Auth-Server der authentifizierten Client-Anwendung ein Access-Token aus und schickt diesen zusammen mit einem ID-Token an die Client-Anwendung.

- **Schritt 7: Client ist autorisiert**

  Die erstmalige Autorisierung des Users für die Web-Anwendung ist nun abgeschlossen. Der User ist registriert.

2.2.4 Authentifizierungsprozess mit OpenID Connect

Okay, den Ablauf zur initialen Registrierung eines Nutzers mit Hilfe von OpenID Connect habe ich verstanden.

Jeder registrierte Nutzer kann sich anschließend bei der Web-Anwendung anmelden und Sie nutzen, um die angeschlossenen Maschinen zu steuern. Bei jeder Anmeldung erfolgt eine Authentifizierung des Nutzers.

2.2.5 Ablauf der Authentifizierung mit Hilfe von OpenID Connect

Da der Nutzer im initialen Registrierungs-Prozess nach Schritt 6 erstmalig gegenüber dem Dienst authentifiziert wurde, fahren wir nun in

Schritt 7 mit der Anmeldung des Users fort.

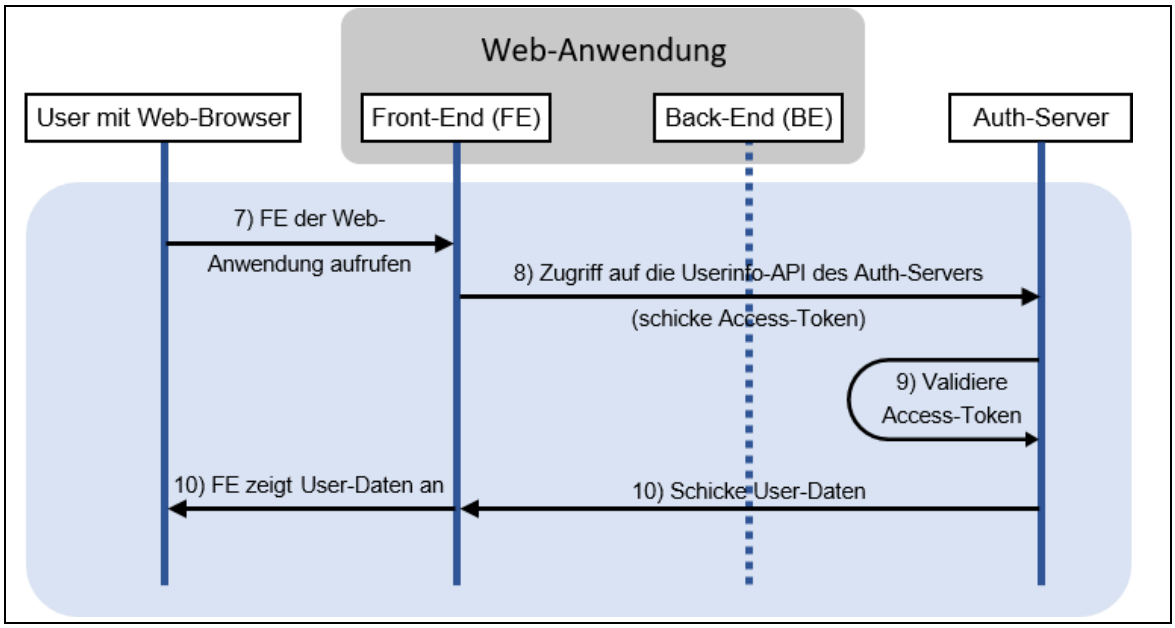

Abb. 11: Authentifizierungsprozess mit OpenID Connect

- **Schritt 7: Front-End der Web-Anwendung aufrufen**

  Der User wird nach seiner erfolgreichen erstmaligen Registrierung auf das Front-End der Web-Anwendung umgeleitet.

- **Schritt 8: Zugriff auf die Userinfo-API des Auth-Servers mit Access-Token**

  Das Front-End der Web-Anwendung fügt den Access-Token in den Authorization-Header der HTTP-Nachricht am Auth-Server ein.

- **Schritt 9: Validiere Access-Token**

  Der Auth-Server prüft in seiner Datenbank, ob es den angegebenen Access-Token gibt und die mitgeschickten Parameter valide Werte enthalten.

- **Schritt 10: Schicke User-Daten und zeige sie im Front-End der Web-Anwendung an**

  Der Auth-Server schickt der Web-Anwendung die geforderten User-Daten zu. Über das Front-End kann der User nun seine User-Daten nutzen, um sich an der Web-Anwendung anzumelden.

2.3 Einsatzgebiete und Probleme von OpenID Connect

2.3.1 Einsatz von OpenID Connect

Für die IT-Abteilung der Siemex AG ist klar: Die Zuverlässigkeit einer Benutzer-Authentifizierung für eine Web-Anwendung wird wesentlich erhöht, indem ein dritter Akteur (der Auth-Server) die Authentifizierung eines Benutzers (1. Akteur) bei der Web-Anwendung (2. Akteur) bestätigt.

Dazu überprüft der dritte Akteur als vertrauenswürdige Instanz die Gültigkeit der Benutzer-Daten (1. und/oder 2. Faktor) für den Zugriff auf eine Web-Anwendung.

Aktuell bieten sich z. B. Apple, Google oder Facebook als vertrauenswürdige Instanzen an. Diese Instanzen müssen dazu vorab natürlich die gültigen Benutzer-Daten kennen, um ihre Authentifizierungsaufgabe erfüllen zu können.

Nicht nur Apple, Google und Facebook bieten ihre Authentifizierungsdienste basierend auf OpenID Connect als Intermediär an. OpenID Connect ist ein offener Standard und kann von jedem verwendet werden, um einen eigenen Autorisierungsserver bereitzustellen.

2.3.2 Probleme von OpenID Connect

Wir können festhalten, dass OpenID Connect die Anmeldung eines Benutzers an einer Web-Anwendung vereinfacht. OpenID Connect kann dabei von unterschiedlichen Dienstleistern oder auch vom Betreiber der Web-Anwendung selbst bereitgestellt werden.

Dennoch müssen bei einer OpenID-Lösung die persönlichen Benutzer-Daten beim User und beim Auth-Server gespeichert und geschützt werden.

Nicht nur die Daten auf Unternehmensseite sind ein Problem.

Die Benutzer sind selbst verantwortlich, sichere Passwörter zu verwenden und diese sicher und für unbefugte Dritte unzugänglich aufzubewahren.

Auch die autorisierende Instanz (z. B. Apple, Google) muss die Benutzer-Daten sicher speichern und schützen.

Die Benutzerdaten müssen zudem zwischen Benutzer und autorisierender Instanz sicher, unverändert und vor unbefugten Dritten geschützt übertragen werden.

2.3.3 Alternative zu OpenID Connect

Wir müssen feststellen, dass die Authentifizierungsmethode OpenID Connect nicht den Sicherheitsansprüchen der Siemex AG genügt.

Die Probleme und Verantwortung im Umgang mit der Speicherung von geheimen Benutzerdaten möchte weder die Geschäftsleitung noch die IT-Abteilung weiter tragen.

Eine Kollegin aus der Abteilung IT-Sicherheit war während der Auswahl von OpenID Connect leider im Urlaub. Sie ist nun zurück und hat der IT-Abteilung eine weitere Authentifizierungslösung vorgeschlagen. Diese soll sogar vollständig ohne Passwörter auskommen.

Die Rede ist von „FIDO2“.

Diese Authentifizierungsmethode schauen wir uns im nächsten WBT genauer an.

2.4 Typische Aufgabenstellungen

2.4.1 Typische Aufgaben – Authentifizierungsmethode OpenID Connect

Typische Aufgabenstellungen – Authentifizierungsmethode OpenID Connect

Zur Bearbeitung dieser Aufgabenstellungen beachten Sie bitte: Verlangt ist eine fachlich zutreffende, inhaltlich nachvollziehbare und kausal zusammenhängende Erörterung aus vollständigen Sätzen in lesbarer Handschrift. Für jede Aufgabe: Maximal zwei Seiten Text!

Aufgabe 1:

Erläutern Sie kurz was OpenID Connect ist und leistet. Gehen Sie anschließend auf die Vorteile bei der Authetifizierung im Internet mit OpenID Connect ein.

Aufgabe 2:

Nennen Sie die OIDC-Akteure und erläutern Sie, welche Rolle sie im Authentifizierungsprozess einnehmen.

Aufgabe 3:

Erläutern Sie den Registrierungsprozess von OpenID Connect.

Aufgabe 4:

Erläutern Sie den Authentifizierungsprozess von OpenID Connect.

Abb. 12: Typische Aufgabenstellungen – Authentifizierungsmethode OpenID Connect

3 Authentifizierungsmethode FIDO2

3.1 Grundlagen von FIDO2

3.1.1 Zurück in der Siemex AG

Die bisherigen Anstrengungen zur Einführung einer geeigneten sicheren Authentifizierungslösung in der Siemex AG waren noch nicht abschließend erfolgreich.

- **1-Faktor- und 2-Faktor-Authentifizierungslösungen (WBT 1)** bieten keinen ausreichenden Schutz zum Authentifizieren an den Web-Applikationen, da Benutzer auf Benutzer-Daten (Benutzer-Name und Passwort) angewiesen sind. Die Probleme im Umgang mit Benutzer-Daten können mit Hilfe eines zweiten Faktors zwar gemildert, aber nicht gelöst werden.
- **OpenID Connect (WBT 2)** zieht einen Intermediär zur zusätzlichen Bestätigung einer Identität hinzu. Aber auch bei dieser Lösung müssen Benutzer-Daten, wie Benutzer-Namen und Passwörter, eingesetzt werden.

Die IT-Abteilung der Siemex AG trifft sich daher heute erneut, um eine weitere Authentifizierungslösung zu diskutieren.

3.1.2 FIDO2 als Alternative

Um die genannten Probleme im Umgang mit Benutzer-Daten beheben zu können, gibt es inzwischen weitere Formen der Authentifizierung.

Diese ermöglichen bspw. ein passwortloses Registrieren und Anmelden und umgehen damit die Probleme der Speicherung von Benutzer-Daten. Auch können mehr als zwei Faktoren miteinander verknüpft werden.

Dabei setzen diese neuen Formen auf bereits bekannte Verschlüsselungsmethoden (wie die Public-Key-Kryptographie) und kombinieren diese mit biometrischen Merkmalen, Hardware-Keys, Smart-Cards oder TPM-Modulen zur Anmeldung an einer Web-Applikation.

Eine Lösung für diese starke und passwortlose Authentifizierung ist **FIDO2 („Fast IDentity Online“)**

3.1.3 FIDO2 – Historische Entwicklung 2013-2015

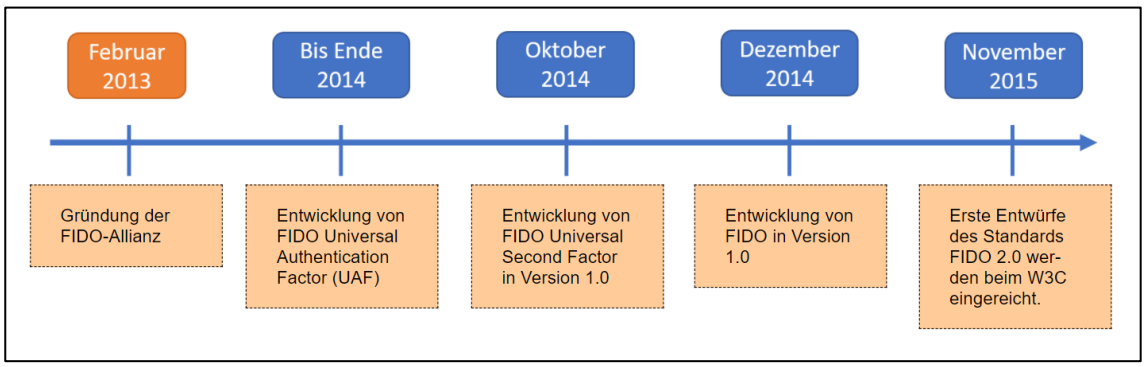

Abb. 13: FIDO2 – Historische Entwicklung 2013-2015

3.1.4 FIDO2 – Historische Entwicklung 2016-2019

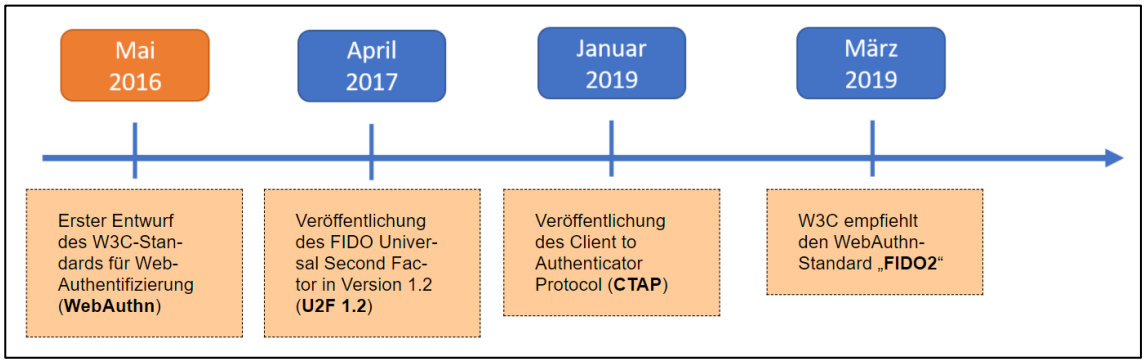

Abb. 14: FIDO2 – Historische Entwicklung 2016-2019

3.1.5 FIDO2

FIDO2 (Fast Identity Online) ist eine Authentifizierungsmethode, die das Verwenden von Passwörtern beim Registrieren und Anmelden an Web-Applikationen überflüssig machen will.

FIDOs Mission: „FIDO Authentication is the Answer to the World's Password Problem“.

Grundlage der Authentifizierung ist das Konzept der Public-Key-Kryptographie, bestehend aus einem Schlüsselpaar eines öffentlichen (public keys) und eines privaten (private keys) Schlüssels.

Aber warum ist FIDO2 eine neue Entwicklung, wenn es die Public-Key-Kryptographie bereits seit langer Zeit gibt?

3.1.6 Die Innovation FIDO2

FIDO2 ist eine Innovation, da es das Konzept der Public-Key-Kryptographie auf die Nutzer-Authentifizierung (Registrierung und Anmeldung) anwendet und die Nutzer mit Hilfe von Hard- und Software von technischen Details fernhält.

Die FIDO Alliance wirbt mit folgenden Vorteilen:

- Vermeidung von Datendiebstählen
- Wachstum für Unternehmen, aufgrund der Aufnahme in den Kreis „FIDO-enabled services“
- Verbesserte Nutzererfahrung und dadurch höhere Besucherzahlen, Steigerung des Markenimages und der Mitarbeiterproduktivität
- Zertifizierungen, Zukunftssicherheit und Kosteneinsparungen

Wie ermöglicht FIDO2 dies?

FIDO2 ermöglicht in Kombination mit geeigneter Hardware und biometrischen Merkmalen des Nutzers (Fingerabdruck, Gesichtserkennung, Spracherkennung) ein passwortloses Authentifizieren mit zwei oder mehr Faktoren.

3.1.7 FIDO2 – Kryptographische Grundlagen

Damit FIDO2 ein passwortloses Authentifizieren ermöglichen kann, wird also ein Verfahren aus der Public-Key-Kryptographie eingesetzt. Dieses Verfahren beschreibt die Vorgänge der Ver- und Entschlüsselung von Informationen mithilfe von Schlüsselpaaren und wird asymmetrisches Verschlüsselungsverfahren genannt.

Asymmetrische Verschlüsselungsverfahren setzen ein Schlüsselpaar, bestehend aus einem öffentlichen (public key) und einem privaten (private key) Schlüssel, ein.

Beide Schlüssel eines Schlüsselpaares sind einzigartig und mathematisch eindeutig voneinander abhängig.

3.1.8 Ein Beispiel zur asymmetrischen Verschlüsselung

Der wesentliche Unterschied der asymmetrischen Verschlüsselung zur symmetrischen Verschlüsselung ist, dass die asym. Verschlüsselung mit zwei unterschiedlichen Schlüsseln arbeitet.

Die asymmetrische Verschlüsselung wird üblicherweise zur Verschlüsselung von E-Mails oder Dateien eingesetzt.

Wir schauen uns am besten ein Beispiel an.

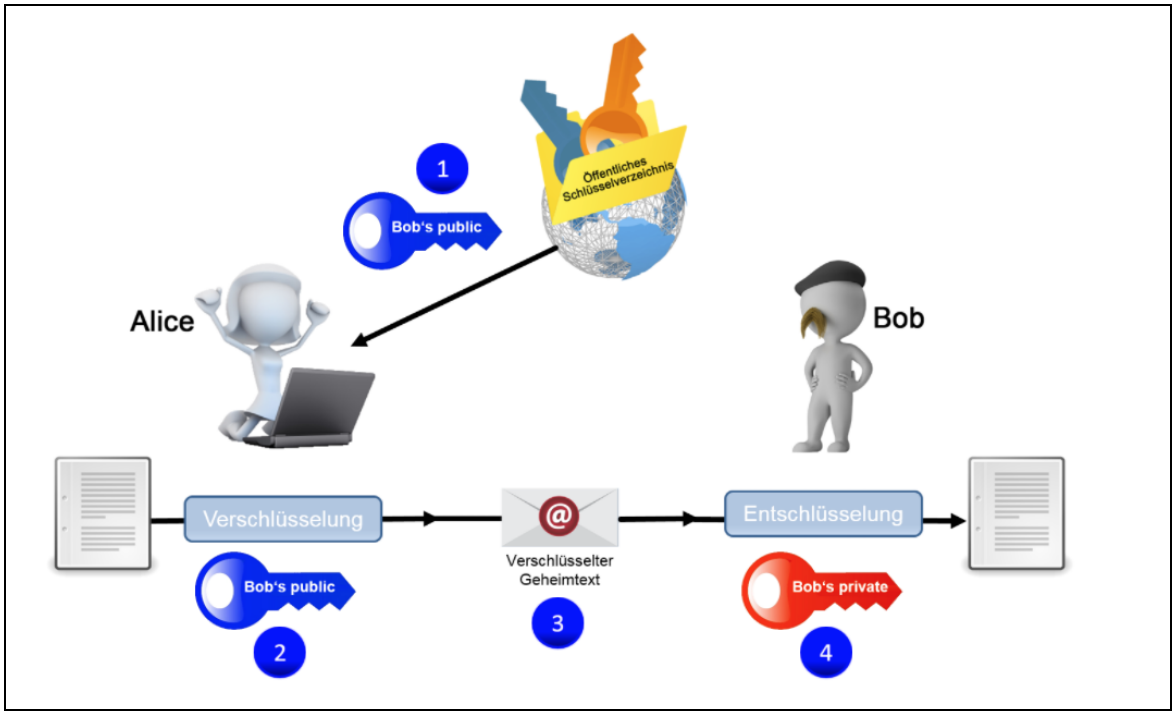

Abb. 15: Beispiel zur asymmetrischen Verschlüsselung

- **Schritt 1:**

  Alice holt sich den öffentlichen Schlüssel von Bob aus der öffentlichen Schlüssel-Liste.

- **Schritt 2:**

  Alice schreibt den Klartext ihrer Nachricht „Klartext“ und verschlüsselt ihn mit dem öffentlichen Schlüssel von Bob. Es entsteht eine Nachricht mit dem Geheimtext.

- **Schritt 3:**

  Alice schickt die Datei mit dem Geheimtext per E-Mail an Bob. Wenn jemand unterwegs die Datei abgreift und öffnet, findet er nur den unverständlichen Geheimtext.

- **Schritt 4:**

  Nur Bob kann die Geheimtext-Datei mit seinem privaten Schlüssel in Klartext umwandeln.

Info: Weitere Informationen zum Thema „Verschlüsselung“ sind der WBT-Serie „Verschlüsseln, Entschlüsseln und Signieren von Dateien und E-Mails“ zu entnehmen.

3.1.9 Asymmetrische Verschlüsselung bei FIDO2

Der **öffentliche Schlüssel** wird genutzt, um Informationen zu verschlüsseln.

Der **private Schlüssel** wird nur zum Entschlüsseln und Signieren verwendet. Der private Schlüssel kann nur die Informationen entschlüsseln, die mit dem dazugehörigen öffentlichen Schlüssel verschlüsselt wurden.

Der **öffentliche Schlüssel** des Schlüsselpaars wird mit Hilfe von FIDO2 an die jeweiligen Dienstanbieter, bei welchen man sich anmelden möchte, übermittelt.

Der **private Schlüssel** des Schlüsselpaars verbleibt auf den Endgeräten des Benutzers und dient als Identitätsbeweis.

FIDO2 übernimmt die Verwaltung und Nutzung dieser Schlüsselpaare. So erstellt FIDO2 im Namen des Benutzers für jeden Dienst ein eigenes Schlüsselpaar, legt es ab und stellt es wenn notwendig, zur Verfügung. Diese Aufgaben waren zuvor den Benutzern selbst überlassen und mit einem großen Aufwand und vielen Fallstricken verbunden. Viele Benutzer schreckten daher vor der Nutzung asymmetrischer Verschlüsselungsverfahren zurück.

3.1.10 Das Challenge-Response-Verfahren

Die Authentifizierung bei der Anmeldung und bei der Registrierung eines Benutzers mit FIDO2 läuft in einem asymmetrischen „Challenge-Response-Verfahren“ ab.

Challenge-Response-Verfahren weisen die Kenntnis von einem Geheimnis nach, ohne dieses Geheimnis preiszugeben. Asymmetrische Challenge-Response-Verfahren basieren auf asymmetrischen Schlüsselpaaren. Die Geheimnisse sind daher die privaten Schlüssel der betreffenden asymmetrischen Schlüssel-Paare.

FIDO2 nutzt das Challenge-Response-Verfahren zur Authentifizierung von Usern einer Web-Anwendung.

Wie genau das Challenge-Response-Verfahren funktioniert, sehen wir uns auf der nächsten Seite an.

3.1.11 Authentifizierung mit Hilfe des Challenge-Response-Verfahrens

FIDO2 nutzt das Challenge-Response-Verfahren zur Authentifizierung von Usern einer Web-Anwendung.

Wie das Verfahren funktioniert, schauen wir uns einmal genauer an.

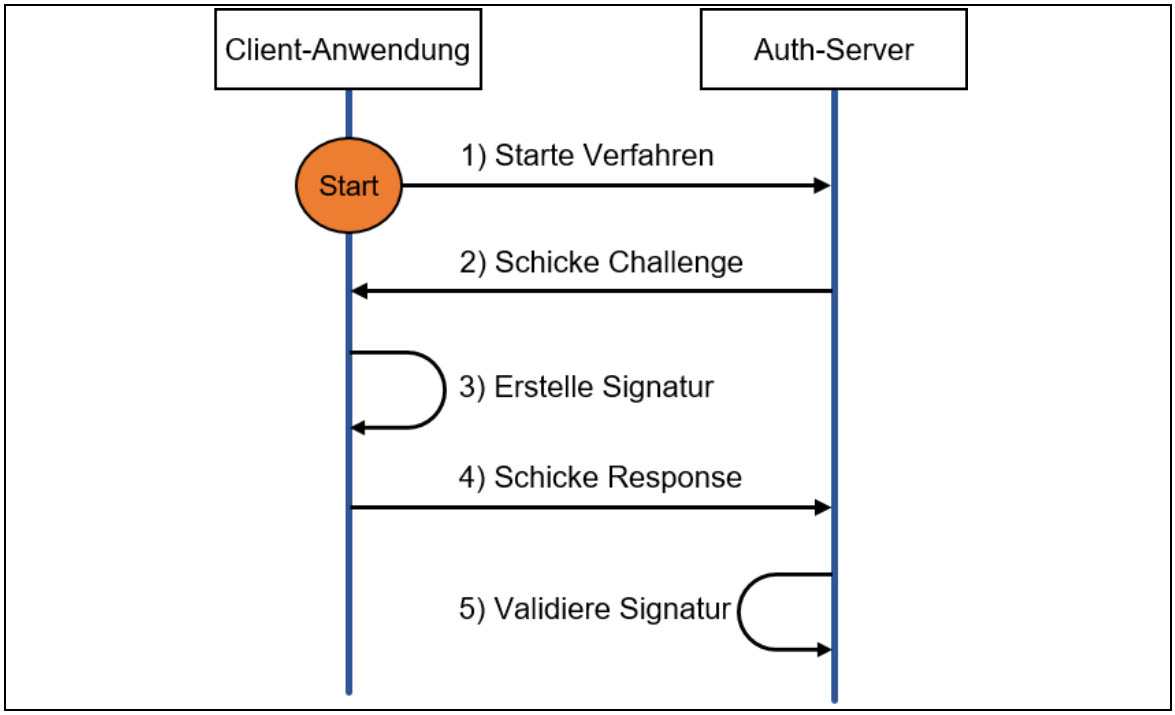

Abb. 16: Das Challenge-Response-Verfahren

- **Schritt 1: Starte Verfahren**

  Um ein asymmetrisches Challenge-Response-Verfahren mit Signaturen zu starten, schickt die Client-Anwendung dem Auth-Server eine Nachricht und signalisiert damit, dass die Client-Anwendung sich authentifizieren will.

- **Schritt 2: Schicke Challenge**

  Der Auth-Server generiert eine Zufallszahl als Challenge. Diese Challenge schickt der Auth-Server an die Client-Anwendung.

- **Schritt 3: Erstelle Signatur**

  Die Client-Anwendung wendet ihren privaten Schlüssel auf die Challenge an. Die Client-Anwendung „signiert“ damit die Challenge.

- **Schritt 4: Schicke Response**

  Die Client-Anwendung schickt die signierte Challenge (die „Signatur“) als Response an den Auth-Server.

- **Schritt 5: Validiere Response**

  Der Auth-Server wendet den öffentlichen Schlüssel der Client-Anwendung auf die signierte Challenge (die „Signatur“) an. Auf Grund des eindeutigen mathematischen Zusammenhangs zwischen öffentlichem und privatem Schlüssel eines Schlüsselpaars kann der Auth-Server erkennen, dass die bei ihm eingegangene Signatur mit dem privaten Schlüssel der Client-Anwendung erstellt wurde.

3.2 Ablauf der Authentifizierung in FIDO2

3.2.1 Akteure in FIDO2

Bevor wir uns den Ablauf der Authentifizierung hinter FIDO2 anschauen können, müssen wir uns zuerst einen Überblick über die Akteure in FIDO2 verschaffen.

- **User**

  Der Benutzer (User) einer Web-Anwendung muss bei FIDO2 den Authentifikator „freischalten“, damit über diesen Authentifikator die Authentifizierung des Users vorgenommen werden kann.

- **Authentifikator**

  Ein Authentifikator ist ein IT-System beim User, das die asym. Schlüsselpaare für FIDO2 generiert, aufbewahrt und auf Nachrichten anwendet. Authentifikatoren können in Computern oder mobilen Endgeräten integriert sein oder als externe Sicherheitsschlüssel vorliegen.

- **Web-Anwendung**

  Bei FIDO2 sind die Web-Anwendung und der Web-Browser ein Akteur. Dieser Akteur ist für die Kommunikation mit dem Auth-Server zuständig. Der Web-Browser muss die WebAuthn-API implementieren, um mit dem Auth-Server per FIDO2 zu kommunizieren.

- **Auth-Server**

  Der Auth-Server muss die Web-Authn-API implementieren, um mit dem Browser des Users per FIDO2 zu kommunizieren.

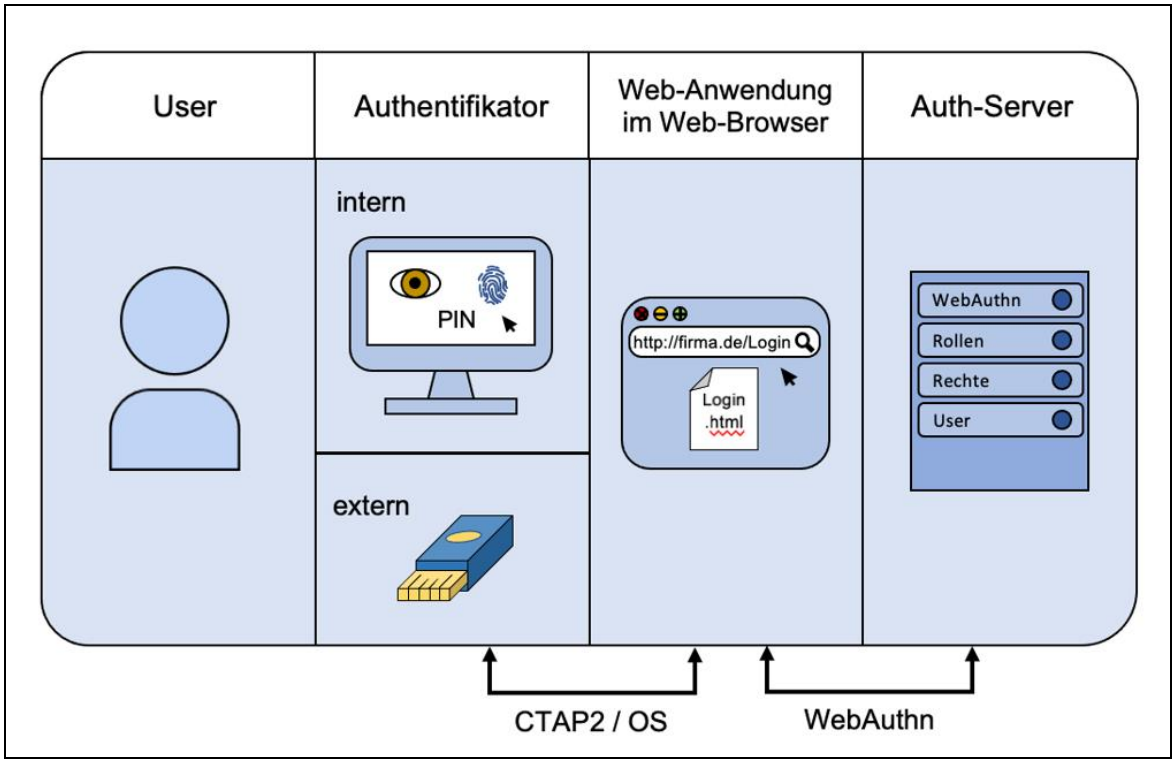

Abb. 17: Akteure in FIDO2

3.2.2 Schnittstellen in FIDO2

Damit die verschiedenen Akteure in FIDO2 miteinander kommunizieren können, bedarf es geeigneter Schnittstellen. Im Falle von FIDO2 sind dies CTAP2 („Client to Authenticator Protocol") und WebAuthn. CTAP2: Das CTAP2-Protokoll vermittelt zwischen externen Authentifikatoren und der FIDO2-eigenen Schnittstelle (WebAuthn-API) im Web-Browser.

WebAuthn: WebAuthn ist eine vom W3C-Konsortium zum Internet-Standard erklärte Web-API (Web-Schnittstelle). Diese ermöglicht die Kommunikation zwischen dem Client und dem Server.

3.2.3 FIDO2 – Ablauf der Registrierung

Super, jetzt kennen wir die beteiligten Akteure und Schnittstellen im Zusammenhang mit FIDO2. Jetzt bringen wir diese in Verbindung mit dem Prozess, den jeder Nutzer durchlaufen muss, um sich mit FIDO2 bei einem Dienst erstmalig zu registrieren.

Dieser Prozess besteht aus sieben Schritten, die wir uns auf der nächsten Seite im Detail ansehen.

3.2.4 Registrierungsprozess in FIDO2

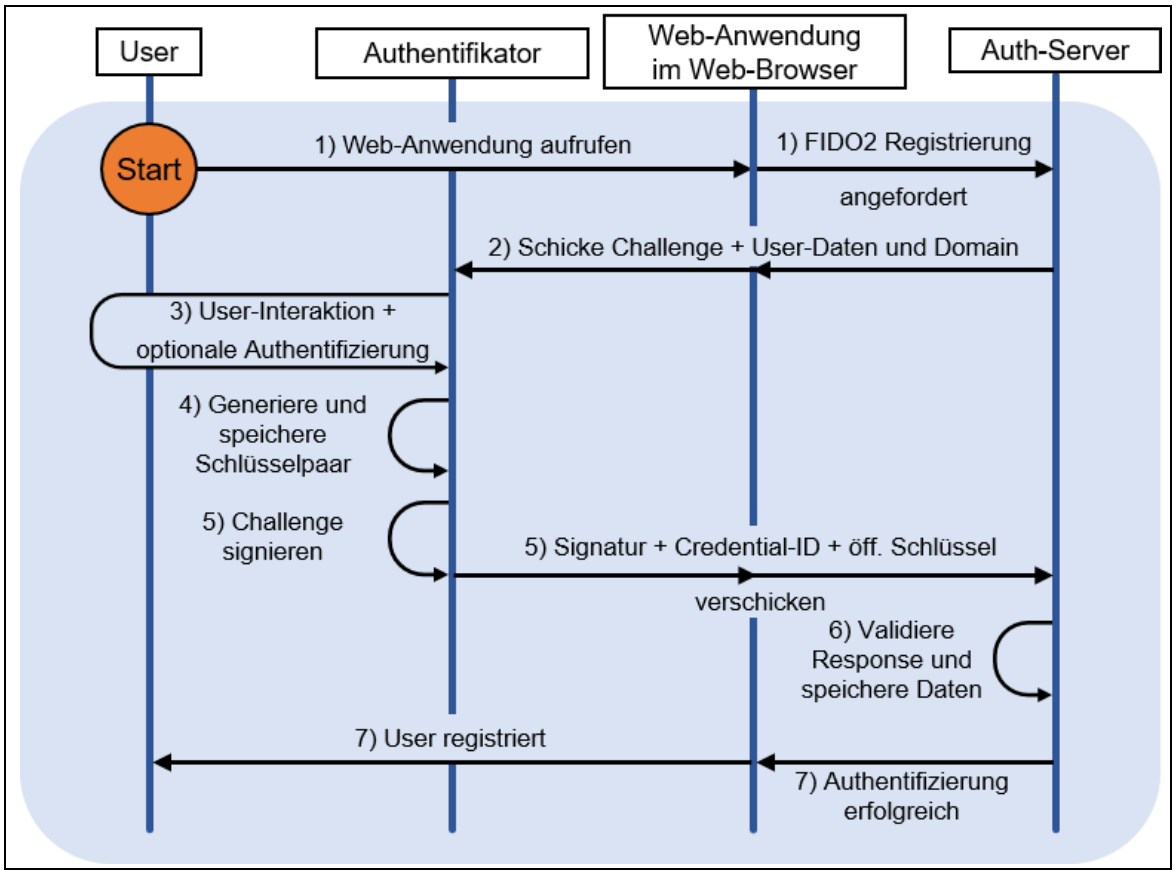

Abb. 18: Registrierungsprozess in FIDO2

- **Schritt 1: Web-Anwendung aufrufen, Authentifizierung anfordern**

  Der User ruft die Web-Anwendung auf. Die Web-Anwendung merkt, dass der User nicht angemeldet ist und leitet den User auf den Auth-Server. Da der User noch keinen Account bei der Web-Anwendung besitzt, muss er einen neuen Account registrieren. Dazu wählt der User zuerst einen User-Namen und gibt weitere User-Daten wie z. B. die E-Mail-Adresse ein.

- **Schritt 2: Schicke Challenge, User-Daten und Domain zum Authentifikator**

  Der Auth-Server generiert eine Challenge und schickt diese mit User-Daten (User-ID, User-Name und Anzeigename) und der Domain der Web-Anwendung über die Web-Authn-API an die Web-Anwendung im Web-Browser.

  Von da wird alles über CTAP2 an den Authentifikator weitergeleitet.

- **Schritt 3: User-Interaktion und optionale Authentifizierung**

  Der User muss bei FIDO2 sein Einverständnis mit einem Authentifizierungsvorgang physisch mitteilen. Wenn das Licht am Authentifikator blinkt, muss der User den Knopf auf seinem Authentifikator drücken. Bei Authentifikatoren mit UV-

Funktion (User Verification) muss der User anschließend die lokale Authentifizierung mit PIN oder Biometrie durchführen.

- **Schritt 4: Generiere und speichere Schlüsselpaar**

  Auf dem Authentifikator ist ein eindeutiges Geheimnis gespeichert. Ein Authentifikator generiert ein asymmetrisches Schlüsselpaar des Users aus diesem Geheimnis und der Domain der Web-Anwendung. Der private Schlüssel des Users wird mit einer „Credential-ID“, der Domain der Web-Anwendung, der User-ID und weiteren Daten auf dem Authentifikator abgelegt.

  Die Credential-ID identifiziert das Schlüsselpaar eines bestimmten Accounts des Users bei einer bestimmten Web-Anwendung auf dem Authentifikator und beim Auth-Server dieser Web-Anwendung (zur Erinnerung: Ein User benötigt bei FIDO2 für jede Web-Anwendung ein eigenes Schlüsselpaar).

  Die Domain der Web-Anwendung gibt den Gültigkeitsbereich des Schlüsselpaars an und die User-ID kann bei Bedarf zur Identifizierung des Users vom Authentifikator eingesetzt werden.

- **Schritt 5: Challenge signieren und Response verschicken**

  Die Challenge (wurde in Schritt 2 vom Auth-Server an die Web-Anwendung geschickt und per CTAP2 an den Authentifikator weitergeleitet) wird nun vom Authentifikator mit dem privaten Schlüssel des Users signiert. Diese Signatur wird mit dem öffentlichen Schlüssel des Users inklusive zugehörigem Zertifikat und der Credential-ID als Response über CTAP2 an den Web-Browser übermittelt. Die gesamte Response wird von der Web-Anwendung über die WebAuthn-API an den Auth-Server geschickt.

- **Schritt 6: Validiere Response und speichere Daten**

  Der Auth-Server validiert die Response, indem er alle übergebenen Parameter und die Signatur überprüft. Um die Signatur zu prüfen, wendet der Auth-Server den öffentlichen Schlüssel des Users auf die Signatur an und überprüft die Zertifizierung. Der Auth-Server speichert den öffentlichen Schlüssel mit der Credential-ID ab und ordnet den öffentlichen Schlüssel einem User-Account zu.

- **Schritt 7: Erfolgreiche Authentifizierung und Registrierung**

  Die Registrierung wurde erfolgreich abgeschlossen und der User wird auf die-Web-Anwendung geleitet, wo er sich mit seinem User-Account anmelden kann.

3.2.5 Authentifizierungsprozess mit FIDO2

Okay, den Ablauf zur initialen Registrierung eines Nutzers mit Hilfe von FIDO2 habe ich verstanden. Nun sollten wir uns noch anschauen, wie FIDO2 beim erneuten Authentifizieren unterstützt.

Der Prozess der Registrierung muss für jeden Dienst einmalig erfolgen. Der Prozess der Authentifizierung muss hingegen jedes Mal vollzogen werden, wenn ein Nutzer sich zuvor entweder abgemeldet hat oder einige Zeit seit der letzten Anmeldung vergangen ist.

3.2.6 Ablauf der Authentifizierung mit Hilfe von FIDO2

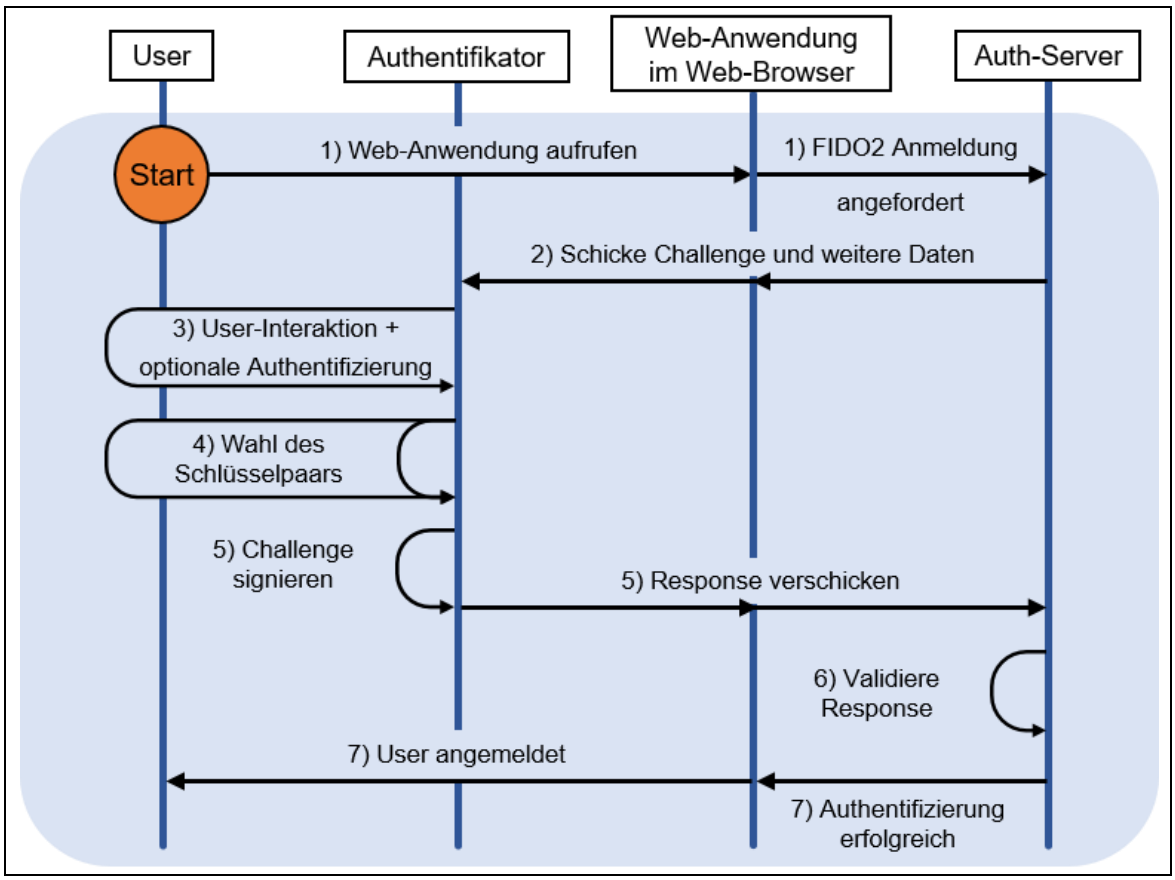

Abb. 19: Authentifizierungsprozess in FIDO2

- **Schritt 1: Web-Anwendung aufrufen, Anmeldung anfordern**

  Der (registrierte) Benutzer (User) ruft die Web-Anwendung auf und gibt seinen Benutzer-Namen ein. Die Web-Anwendung leitet den (registrierten) Benutzer zum Auth-Server.

- **Schritt 2: Schicke Challenge, User-Daten zum Authentifikator**

  Der Auth-Server generiert eine Challenge. Die Challenge wird zusammen mit der Domain der Web-Anwendung, einer Liste erlaubter Credential-IDs für den Benutzer und der Benutzer-Verifikations-Anforderung (UV oder UP) über WebAuthn an die Web-Anwendung im Web-Browser geschickt.

  Über die CTAP2-Schnittstelle werden die Challenge und die Benutzer-Daten an den Authentifikator weitergeleitet.

- **Schritt 3: User-Interaktion und optionale lokale Authentifizierung**

  Wie bei der Registrierung muss der Benutzer (User) bei der Anmeldung mit FIDO2 sein Einverständnis mit einem Authentifizierungsvorgang physisch mitteilen. Wenn das Licht am Authentifikator blinkt, muss der Benutzer den Knopf auf seinem Authentifikator drücken. Die übergebene Benutzer-Verifikations-Anforderung des Auth-Servers muss erfüllt werden. Bei geforderter „User Verification“ (UV) muss der Benutzer sich lokal per PIN oder Biometrie authentifizieren.

- **Schritt 4: Benutzer wählt Schlüsselpaar**

  Der Authentifikator zeigt dem Benutzer nun alle seine registrierten User-Accounts für die betroffene Web-Anwendung (Domain). Der Benutzer wählt einen Account aus. Der Authentifikator kennt damit die Credential-ID des Benutzer-Accounts, welchen der Benutzer nutzen will.

- **Schritt 5: Challenge signieren und Response verschicken**

  Der Authentifikator signiert die Challenge mit dem privaten Schlüssel des Benutzers. Diese Signatur wird zusammen mit der Domain der Web-Anwendung und der Information zur Benutzer-Verifikations-Anforderung (UV oder UP) als Response über CTAP2 an die Web-Anwendung im Web-Browser übergeben. Über Web-Authn wird die Response an den Auth-Server weitergeleitet.

- **Schritt 6: Validiere Response**

  Der Auth-Server validiert die Response, indem er alle übergebenen Parameter und die Signatur überprüft. Um die Signatur zu prüfen, wendet der Auth-Server den öffentlichen Schlüssel des Users auf die Signatur an.

- **Schritt 7: Authentifizierung und Anmeldung erfolgreich**

  Der Benutzer wurde erfolgreich authentifiziert und angemeldet. Jetzt kann der Benutzer auf die Web-Anwendung zugreifen.

3.3 FIDO2 als Lösung vieler Sicherheitsprobleme

3.3.1 FIDO2 – Lösung statt Milderung

Der IT-Abteilung der Siemex AG gefällt die neue Authentifizierungslösung „FIDO2“.

FIDO2 kann die genannten Sicherheitsprobleme wesentlich verringern, da mit Hilfe von FIDO2 keine kritischen Benutzer-Daten auf der Server-Seite gespeichert oder über Netzwerkleitungen des offenen Internet übertragen werden müssen.

Die Sicherheit der Benutzer-Daten auf der Benutzer-Seite wird durch spezielle technische Vorrichtungen somit kategorial erhöht.

3.3.2 FIDO2 – Weitere Vorteile

Neben der gesteigerten Sicherheit durch passwortloses Anmelden entstehen der Siemex AG weitere Vorteile durch den Einsatz von FIDO2.

Usability:

Mitarbeiter können sich überall mit ihrem Hardware Key einloggen ohne Passwörter parat haben zu müssen. Sie müssen nur noch für einen sicheren Ablageort des FIDO2-Geräts sorgen und keine Verwaltung von komplexen Passwörtern bewerkstelligen.

Kosten:

Die IT-Abteilung muss keine Verwaltung und Ausgabe von Benutzerkennungen führen. Mitarbeiter können sich eigenständig registrieren und authentifizieren. Lediglich eine Rechtezuweisung muss durch die IT-Abteilung erfolgen.

Sicherheit:

Die Gefahren durch Social Engineering, Datenlecks, Phishing und Malware werden stark reduziert.

3.3.3 FIDO2 – Authentifizierung ohne Autorisierung

Aber Achtung: FIDO2 ist zwar eine Lösung zur Authentifizierung, beinhaltet jedoch keine Konzepte und Prozesse für eine Autorisierung.

Dies bedeutet: Ein Mitarbeiter kann sich mit Hilfe von FIDO2 zwar sicher und erfolgreich an einer der Web-Applikationen der Siemex AG authentifizieren. Eine anschließende Rechtezuweisung innerhalb der Systeme kann mit Hilfe von FIDO2 jedoch nicht erfolgen.

FIDO2 kann somit einen Zugang zu einer Web-Applikation absichern, jedoch den Mitarbeitern keine entsprechenden Rechte innerhalb des Systems zuweisen.

Aus diesem Grund hat sich die IT-Abteilung dazu entschieden, zusätzlich zu FIDO2 als Authentifizierungslösung, OpenID Connect als Autorisierungslösung zu integrieren.

3.4 Typische Aufgabenstellungen

3.4.1 Typische Aufgaben – Authentifizierungsmethode OpenID Connect

Typische Aufgabenstellungen – Authentifizierungsmethode FIDO2

Zur Bearbeitung dieser Aufgabenstellungen beachten Sie bitte: Verlangt ist eine fachlich zutreffende, inhaltlich nachvollziehbare und kausal zusammenhängende Erörterung aus vollständigen Sätzen in lesbarer Handschrift. Für jede Aufgabe: Maximal zwei Seiten Text!

Aufgabe 1:

Erläutern Sie, warum FIDO2 eine Innovation ist.

Aufgabe 2:

Erläutern Sie den Einsatz der asymmetrischen Verschlüsselung innerhalb von FIDO2.

Aufgabe 3:

Erläutern Sie den Registrierungsprozess von FIDO2.

Aufgabe 4:

Erläutern Sie die wesentlichen Vorteile von FIDO2.

Abb. 20: Typische Aufgabenstellungen – Authentifizierungsmethode FIDO2

Impressum

Reihe: **Arbeitspapiere Wirtschaftsinformatik** (ISSN 1613-6667)

Bezug: http://wi.uni-giessen.de

Herausgeber: Prof. Dr. Axel Schwickert
Prof. Dr. Bernhard Ostheimer

c/o Professur BWL – Wirtschaftsinformatik
Justus-Liebig-Universität Gießen
Fachbereich Wirtschaftswissenschaften
Licher Straße 70
D – 35394 Gießen
Telefon (0 64 1) 99-22611
Telefax (0 64 1) 99-22619
eMail: Axel.Schwickert@wirtschaft.uni-giessen.de
http://wi.uni-giessen.de

Ziele: Die Arbeitspapiere dieser Reihe sollen konsistente Überblicke zu den Grundlagen der Wirtschaftsinformatik geben und sich mit speziellen Themenbereichen tiefergehend befassen. Ziel ist die verständliche Vermittlung theoretischer Grundlagen und deren Transfer in praxisorientiertes Wissen.

Zielgruppen: Als Zielgruppen sehen wir Forschende, Lehrende und Lernende in der Disziplin Wirtschaftsinformatik sowie das IT-Management und Praktiker in Unternehmen.

Quellen: Die Arbeitspapiere entstehen aus Forschungs-, Abschluss-, Studien- und Projektarbeiten sowie Begleitmaterialien zu Lehr-, Vortrags- und Kolloquiumsveranstaltungen der Professur BWL – Wirtschaftsinformatik, Prof. Dr. Axel Schwickert, Justus-Liebig-Universität Gießen sowie der Professur für Wirtschaftsinformatik, insbes. medienorientierte Wirtschaftsinformatik, Prof. Dr. Bernhard Ostheimer, Fachbereich Wirtschaft, Hochschule Mainz.

Hinweise: Wir nehmen Ihre Anregungen zu den Arbeitspapieren aufmerksam zur Kenntnis und werden uns auf Wunsch mit Ihnen in Verbindung setzen.

Falls Sie selbst ein Arbeitspapier in der Reihe veröffentlichen möchten, nehmen Sie bitte mit einem der Herausgeber unter obiger Adresse Kontakt auf.

Informationen über die bisher erschienenen Arbeitspapiere dieser Reihe erhalten Sie unter der Web-Adresse http://wi.uni-giessen.de/