JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN
PROFESSUR BWL – WIRTSCHAFTSINFORMATIK
UNIV.-PROF. DR. AXEL SCHWICKERT

Fabian, S.; Schwickert, Axel

OpenID Connect und FIDO2 – Die Kombination in Keycloak

ARBEITSPAPIERE WIRTSCHAFTSINFORMATIK

Nr. 9 / 2021
ISSN 1613-6667

Arbeitspapiere WI Nr. 9 / 2021

Autoren: Fabian, S.; Schwickert, Axel

Titel: OpenID Connect und FIDO2 – Die Kombination in Keycloak

Zitation: Fabian, S.; Schwickert, Axel: OpenID Connect und FIDO2 – Die Kombination in Keycloak, in: Arbeitspapiere WI, Nr. 9/2021, Hrsg.: Professur BWL – Wirtschaftsinformatik, Justus-Liebig-Universität Gießen 2021, 21 Seiten, ISSN 1613-6667.

Kurzfassung: Das vorliegende Arbeitspapier WI Nr. 9/2021 befasst sich mit der Kombination von OpenID Connect (OIDC) und FIDO2. Die Grundlagen zu OIDC und FIDO2 finden sich im Arbeitspapier WI Nr. 8/2021 „OpenID Connect und FIDO2 – Ein Vergleich“.

OpenID Connect (OIDC) ist eine Authentifizierungsmethode für Web-Anwendungen, die das Protokoll OAuth 2.0 erweitert. OIDC wurde im Jahr 2014 als Standard der OpenID Foundation (eine Non-Profit-Standardisierungsorganisation, gegründet 2007) verabschiedet. OIDC liegt (Stand März 2021) in Version 1.0 vor. OIDC enthält zwar den Vorgang der Authentifizierung eines Users, spezifiziert aber nicht die dafür anzuwendende Authentifizierungsmethode. OIDC spezifiziert jedoch die Autorisierung eines authentifizierten Users zum Zugriff auf eine Web-Anwendung. FIDO2 ist eine passwortlose Authentifizierungsmethode für Web-Anwendungen der FIDO Alliance (ein offener Industrieverband, gegründet 2012). Im Februar 2016 wird FIDO2 erstmals vom W3C erwähnt. Im April 2018 wird FIDO2 dann als Standard veröffentlicht. Da FIDO2 eine reine Authentifizierungsmethode für User einer Web-Anwendung ist und nicht die Autorisierung von Usern zum Zugriff auf die Web-Anwendung regelt, muss ein zusätzliches Verfahren diese Autorisierung umsetzen. Dazu bietet sich OIDC an. OIDC und FIDO2 können für eine sichere Authentifizierung und Autorisierung kombiniert werden. Im vorliegenden Arbeitspapier wird die praktische Umsetzung dieser Kombination gezeigt. Dazu wird die IAM-Software (Identity- and Accessmanagement) „Keycloak“ eingesetzt. Keycloak umfasst eine Vielzahl von Funktionen, um Authentifizierungs- und Autorisierungsvorgänge feingranular zu konfigurieren.

Schlüsselwörter: OpenID Connect, OIDC, FIDO2, OAuth, Keycloak, Authentifizierungsmethode, Auth-Server, Access-Token, WebAuthn, Authentifikator, Trusted Platform Module, Secure Element, CATP2, Challenge Response, Signatur, Kryptographie, asymmetrische Verschlüsselung

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

Auth-Server ..... Authentifizierungs-Server

FIDO ..... Fast Identity Online

IAM ..... Identity- & Access-Management

ID ..... Identifikationsnummer

Inc ..... Incorporated

IT-System ..... Informationstechnologie-System

OIDC ..... OpenID Connect

PIN ..... Personal Identification Number

SAML ..... Security Assertion Markup Language

USB ..... Universal Serial Bus

1 Ausgangssituation

In der vorliegenden Arbeit wird eine Implementierung der Authentifizierungsmethode FIDO2 in Kombination mit OIDC beschrieben. Für diese Implementierung wird die IAM-Software „Keycloak“ eingesetzt. IAM steht für „Identity- and Accessmanagement“. Mit IAM-Software wird sichergestellt, dass nur befugte Personen (Authentifizierung) einen definierten Zugang zu einer Web-Anwendung (Autorisierung) haben. Keycloak ist ein Software-System, mit dem die Authentifizierung und Autorisierung von Usern durchgeführt werden kann. Zur Autorisierung setzt Keycloak OIDC oder SAML[1] ein. Zur Authentifizierung bietet Keycloak viele verschiedene und konfigurierbare Methoden an.

Keycloak ist ein Open-Source-Software-System, das für den Betreiber auch im kommerziellen Einsatz kostenfrei ist.[2] Keycloak wurde im Jahr 2014 als JBOSS-Community-Projekt erstmals veröffentlicht.[3] Im März 2018 wurde die Entwicklung von Red Hat übernommen. Seitdem wird Keycloak durch Red Hat veröffentlicht.[4] Mittlerweile (Stand März 2021) befindet sich Keycloak in der Version 12.0.4.[5] Keycloak umfasst eine Vielzahl von Funktionen, um Authentifizierungs- und Autorisierungsvorgänge feingranular zu konfigurieren.[6] Konkurrenzprodukte von Keycloak sind z. B. Amazon Cognito[7] und Auth0 Services[8].

1 SAML (Security Assertion Markup Language) ist ein von dem Unternehmen OASIS standardisiertes Framework für die sichere Übertragung der Daten von Anwendungen über das offene Internet. Siehe Campbell, Brian et al.: Security Assertion Markup Language (SAML) 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants, Internet Engineerig Task Force (Hrsg.), Mai 2015, https://tools.ietf.org/pdf/rfc7522.pdf, abgerufen am 09.03.2021, S. 2.

2 Vgl. Koserwal, Abhishek: Keycloak: Core concepts of open source identity and access management, Red Hat Inc. (Hrsg.), 11. Dezember 2019, https://developers.redhat.com/blog/2019/12/11/keycloak-core-concepts-of-open-source-identity-and-access-management/, abgerufen am 01.03.2021. Vgl. auch Apache Software Foundation (Hrsg.): Apache License Version 2.0, Januar 2004, https://www.apache.org/licenses/LICENSE-2.0.txt, abgerufen am 01.03.2021, S. 2.

3 Vgl. Burke, Bill: Keycloak 1.0 Final Released, lists.jboss.org (Hrsg.), 11. September 2014, https://lists.jboss.org/archives/list/keycloak-dev@lists.jboss.org/thread/U3KXSAUJSN4TONLRLFCKA3EZWWEDW7NZ/, abgerufen am 13.03.2021.

4 Vgl. Reinhardt, Martin: IT-Systemzugriffe verwalten: Identity und Access Management mit Keycloak, in: iX 12/20, Heise Medien GmbH & Co. KG (Hrsg.), S. 108f.

5 Vgl. Keycloak.org (Hrsg.): Release Notes, https://www.keycloak.org/docs/latest/release_notes/index.html, abgerufen am 11.03.2021.

6 Vgl. Reinhardt, Martin: IT-Systemzugriffe verwalten: Identity und Access Management mit Keycloak, a. a. O., S. 110.

7 Vgl. Amazon Web Services Inc. (Hrsg.): Amazon Cognito Entwicklerhandbuch, 2021, https://docs.aws.amazon.com/de_de/cognito/latest/developerguide/cognito-dg.pdf#what-is-amazon-cognito, abgerufen am 11.03.2021, S. 1f.

8 Vgl. Auth0 Inc. (Hrsg.): Get Started, https://auth0.com/docs/get-started, abgerufen am 11.03.2021.

In der vorliegenden Arbeit wird eine Instanz von Keycloak lokal in einem Docker-Container[9] installiert. „Lokal“ bedeutet hier, dass die Keycloak-Instanz auf dem Endgerät der Verfasser der vorliegenden Arbeit installiert ist. Diese Keycloak-Instanz soll die Authentifizierung und Autorisierung des Users einer Web-Anwendung (in Angular[10] erstellt) übernehmen, die ebenfalls lokal auf dem Endgerät der Verfasser der vorliegenden Arbeit betrieben wird. Die Keycloak-Instanz und die Web-Anwendung können alternativ auf entfernten Server-Rechnern installiert und über Netzwerkverbindungen angesprochen werden.[11]

In der vorliegenden Arbeit wird nicht darauf eingegangen, wie ein Server oder Endgerät aufgesetzt und konfiguriert wird, um eine Keycloak-Instanz darauf zu betreiben.[12] Es wird nicht beschrieben, wie eine Instanz von Docker auf einem Server-Rechner oder Endgerät installiert und in Betrieb genommen wird. Ebenso wird die Installation einer Keycloak-Instanz in einem Docker-Container nicht dargelegt. Für die Implementierung von FIDO2 wird eine Keycloak-Instanz in einem Docker-Container auf einem Endgerät der Verfasser der vorliegenden Arbeit betrieben. Über den „localhost“[13] auf dem Port 8080 ist die Keycloak-Instanz erreichbar.

Damit die Keycloak-Instanz konfiguriert werden kann, wird die Administrator-Konsole von Keycloak (siehe Kapitel 2) eingesetzt. Die Administrator-Konsole ist eine Web-Applikation, die eine Keycloak-Instanz zu ihrer eigenen Steuerung mitbringt. Um auf die Admi-

9 Das Unternehmen Docker Inc. vertreibt Software-Produkte, mit denen „Container“ erstellt und ausgeführt werden können. Ein Container ist die Instanz einer Software, die so verpackt ist, dass sie schnell und zuverlässig Umgebungs-unabhängig ausgeführt werden kann. Siehe Docker Inc. (Hrsg.): What is a Container? – A standardized unit of software, https://www.docker.com/resources/what-container, abgerufen am 09.03.2021.

10 Angular ist eine TypeScript-basierte Entwicklungs-Plattform, mit der modulare Web-Anwendungen entwickelt werden. Siehe Angular.io (Hrsg.): What is Angular?, https://angular.io/guide/what-is-angular, abgerufen am 09.03.2021.

11 Docker-Container können auf Docker-Anwendungen unabhängig von der Umgebung ausgeführt werden. Deshalb kann die Keycloak-Instanz vom lokalen Endgerät der Verfasser der vorliegenden Arbeit auch auf entfernten Servern installiert werden. Um die Web-Anwendung auf einem entfernten Server-Rechner zu betreiben, muss nur der Quellcode der Web-Anwendung auf diesem Server-Rechner gespeichert werden.

12 Für nähere Informationen zur Konfiguration und Installation von Keycloak siehe Keycloak.org (Hrsg.): Server Administration Guide, https://www.keycloak.org/ docs/latest/server_admin/, abgerufen am 30.12.2020. Siehe auch Saeed, Hasnat: Setup Keycloak Server on Ubuntu 18.04, Medium.com (Hrsg.), 31. Juli 2019, https://medium.com/@hasnat.saeed/setup-keycloak-server-on-ubuntu-18-04-ed8c7c79a2d9, abgerufen am 08.03.2021.

13 Ein Computer wird als „localhost“ bezeichnet, wenn dieser sich bei HTTP-Kommunikation selbst addressiert, also die Client- und die Server-Rolle zugleich innehat. Siehe Eastlake, Donald; Panitz, Aliza: Reserved Top Level DNS Names, Juni 1999, https://tools.ietf.org/html/rfc2606, abgerufen am 10.03.2021, S. 2.

nistrator-Konsole zugreifen zu können, muss ein Entwickler sich dort mit einem Administrator-Account anmelden. Dieser Administrator-Account wird bei der Installation der Keycloak-Instanz registriert. In der vorliegenden Arbeit wird der Vorgang zur Installation der Keycloak-Instanz nicht betrachtet. Es wird davon ausgegangen, dass ein Administrator-Account nutzungsfertig eingerichtet wurde.

Die Web-Anwendung, die mit der Keycloak-Instanz verbunden werden soll, wurde mit der Entwicklungs-Plattform Angular erstellt. Die Web-Anwendung besteht nur aus den für die Verbindung zur Keycloak-Instanz benötigten Elementen. Die Web-Anwendung wird über den „localhost“ auf dem Port 4200 betrieben.

In der vorliegenden Arbeit wird nicht betrachtet wie ein Web-Server auf entfernten Server-Rechnern oder einem lokalen Endgerät aufgesetzt und konfiguriert werden muss, um Web-Anwendungen zu hosten. In der vorliegenden Arbeit wird nicht erläutert, wie Web-Anwendungen mit Angular entwickelt werden. Das umfasst auch die Implementierung von Modifikationen zur Verbindung einer Angular-Web-Anwendung mit der Keycloak-Instanz, da diese Modifikationen nicht direkt für die Implementierung von FIDO2 relevant sind. Diese Modifikationen werden nur benötigt, damit die Web-Anwendung mit der Keycloak-Instanz kommunizieren kann.[14]

2 Konfiguration einer Keycloak-Instanz für FIDO2

Nachfolgend wird textlich und mit den Abbildungen 1 bis 17 schrittweise erläutert, wie eine Keycloak-Instanz konfiguriert werden muss, um FIDO2 als Authentifizierungsmethode in Kombination mit OIDC als Autorisierungsmethode einzusetzen. Es wird hier nur auf die FIDO2-spezifischen Konfigurationen eingegangen.

Schritt 1: In der Administrator-Konsole der Keycloak-Instanz anmelden

Im ersten Schritt muss ein Entwickler die Administrator-Konsole der Keycloak-Instanz in seinem Web-Browser öffnen. Um auf die Administrator-Konsole zuzugreifen, muss der Entwickler die URL „http://localhost:8080“ aufrufen. Die in Abbildung 1 dargestellte

14 Für nähere Informationen dazu, wie eine Angular-Web-Anwendung modifiziert werden muss, um eine Keycloak-Instanz als IAM-Dienstleister zu nutzen, siehe Kumar, Anjan: Keycloak Integration in Angular Application, dev.to (Hrsg.), 05. Februar 2021, https://dev.to/anjnkmr/keycloak-integration-in-angular-application-5a43, abgerufen am 09.03.2021.

Web-Seite wird angezeigt. Der Entwickler gelangt über den hervorgehobenen Bereich mit Bezeichnung „Administration Console“ auf die Administrator-Konsole.

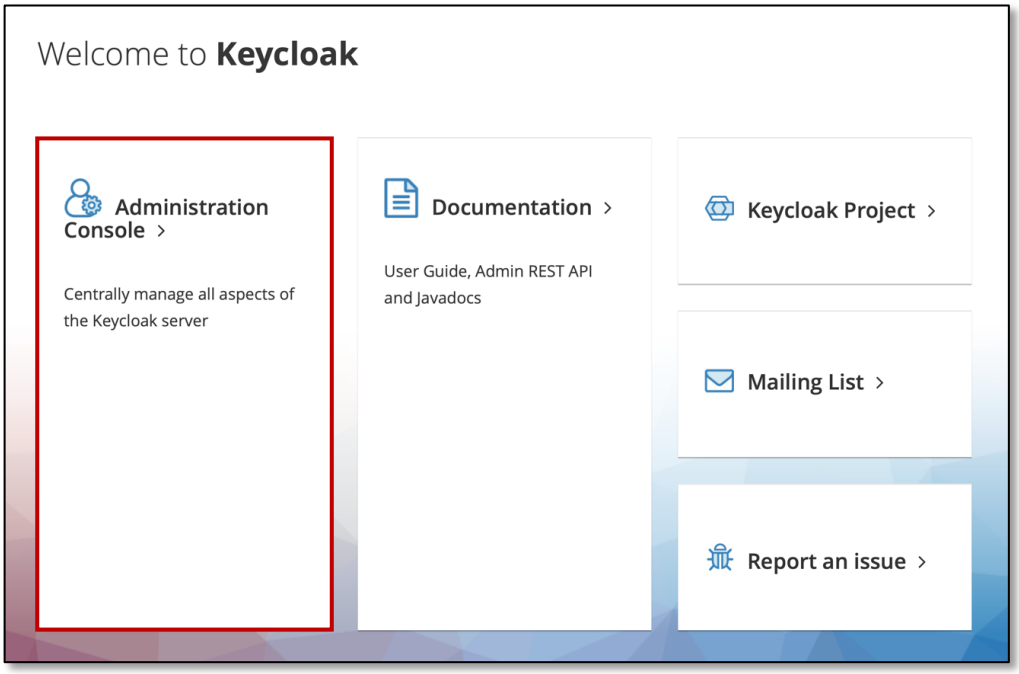

Abb. 1: Die Keycloak-Start-Seite

Der Entwickler wird auf die in Abbildung 2 dargestellte Web-Seite weitergeleitet. Hier muss der Entwickler Username und Passwort von seinem Administrator-Account angeben, um Zugriff zur Administrator-Konsole zu bekommen.

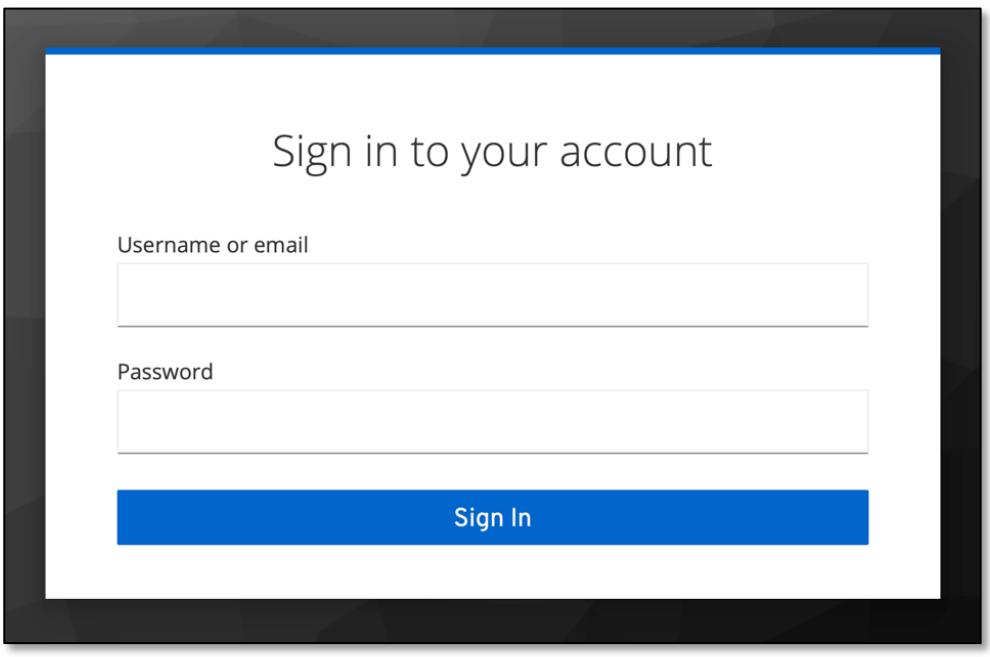

Abb. 2: Die Anmeldemaske zur Administrator-Konsole

Schritt 2: Realm anlegen

Dem Entwickler wird der Zugang zu der Administrator-Konsole gewährt (siehe Abbildung 3). Auf der linken Seite der Administrator-Konsole befindet sich eine Menü-Leiste für die Navigation zwischen den Web-Seiten der Administrator-Konsole für die Konfiguration der Keycloak-Instanz. Die in Abbildung 3 dargestellte Menü-Leiste am linken Rand der Administrator-Konsole wird zur besseren Übersichtlichkeit in den nachfolgenden Abbildungen nicht mehr dargestellt. Es wird textlich beschrieben, wenn eine andere Web-Seite über die Menü-Leiste ausgewählt werden muss.

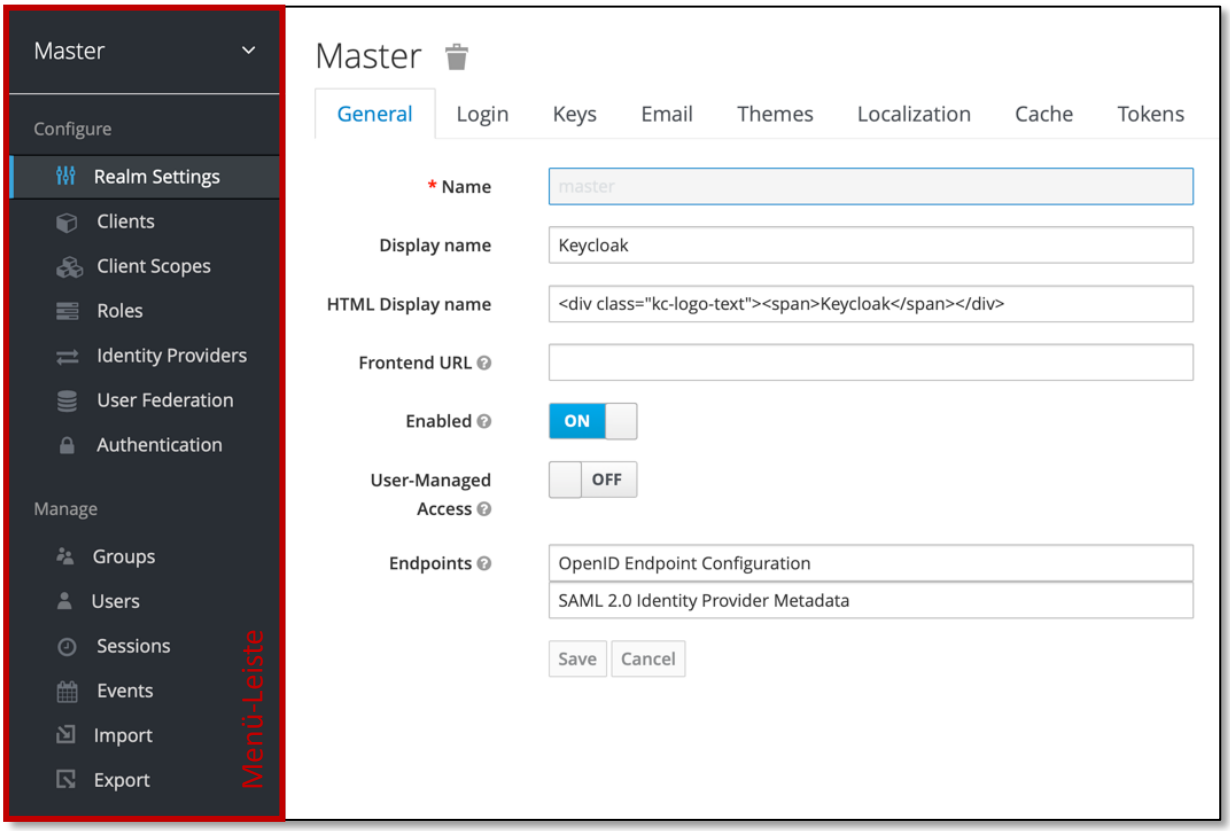

Abb. 3: Die Administrator-Konsole von Keycloak

Ganz oben in der Menü-Leiste befindet sich ein Drop-Down-Menü mit der Bezeichnung „Master“. Dieses Drop-Down-Menü dient zum Wechsel zwischen „Realms“. Ein Realm ist bei Keycloak ein begrenzter Bereich mit eigenen Konfigurationen, Nutzern, Rollen u. v. m. Das Master-Realm ist bei einer Keycloak-Instanz immer vorinstalliert und vorausgewählt. Das Master-Realm sollte nicht für den Produktiv-Einsatz verwendet werden. Stattdessen

können Entwickler eigene Realms erstellen.[15] Um ein bestehendes Realm auszuwählen, muss die Bezeichnung des Realms im Drop-Down-Menü ausgewählt werden. Das Drop-Down-Menü hat aber auch eine Option mit der Bezeichnung „add Realm“, über die ein neues Realm erstellt werden kann. Nachfolgend wird ein neues Realm erstellt.

Nach der Auswahl von „add Realm“ wird der Entwickler auf die in Abbildung 4 dargestellte Web-Seite geleitet. Hier muss eine Bezeichnung für das neue Realm festgelegt werden. In der vorliegenden Arbeit ist die Bezeichnung des neuen Realms „FIDO2_Realm“. Das Realm wird durch den Button „Create“ erzeugt.

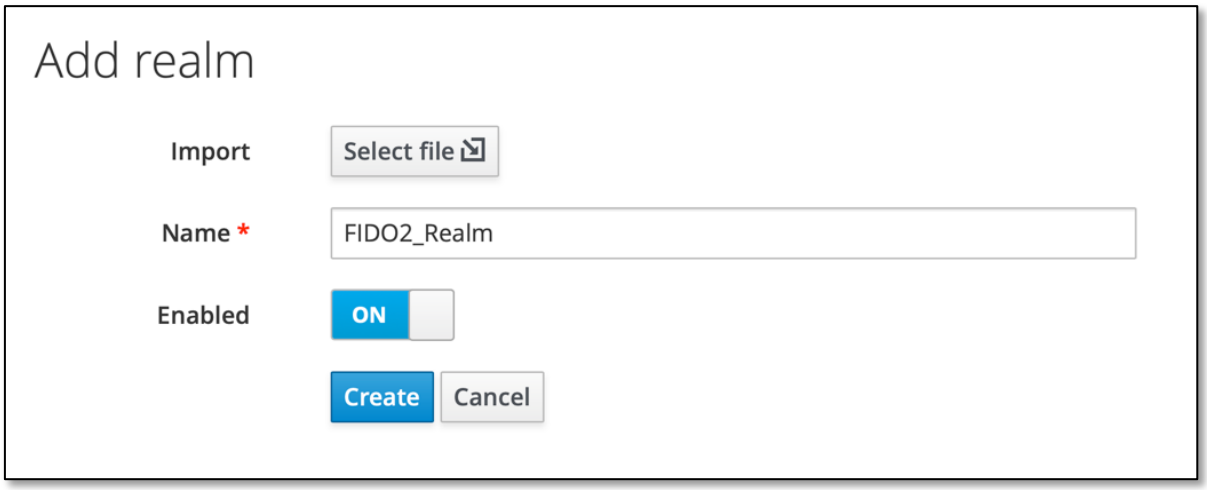

Abb. 4: Der Keycloak-Instanz ein neues Realm hinzufügen

Schritt 3: Realm konfigurieren

Der Entwickler wird automatisch auf die Realm-Einstellungen (Menü-Leiste „Realm Settings“) im Reiter „General“ (siehe Abbildung 5) weitergeleitet. Für die FIDO2-Implementierung in der vorliegenden Arbeit sind ausschließlich Konfigurationen im Reiter „Login“ relevant.

In Abbildung 5 werden die Login-Konfigurationen dargestellt. Für die vorliegende Arbeit sollen die Werte der in Abbildung 5 hervorgehobenen Schiebeschalter geändert werden. Damit Entwickler die User-Accounts für die Web-Anwendung nicht manuell in der Administrator-Konsole anlegen müssen, wird die Registrierung durch den User („User registration“) aktiviert. Die Anmeldung mit der E-Mail-Adresse des Users („Login with

15 Vgl. Reinhardt, Martin: IT-Systemzugriffe verwalten: Identity und Access Management mit Keycloak, a. a. O., S. 110.

email“) wird deaktiviert, damit einfache und kurze Usernamen (z. B. „User“ statt „max.mustermann@web.de“) vergeben werden können.[16]

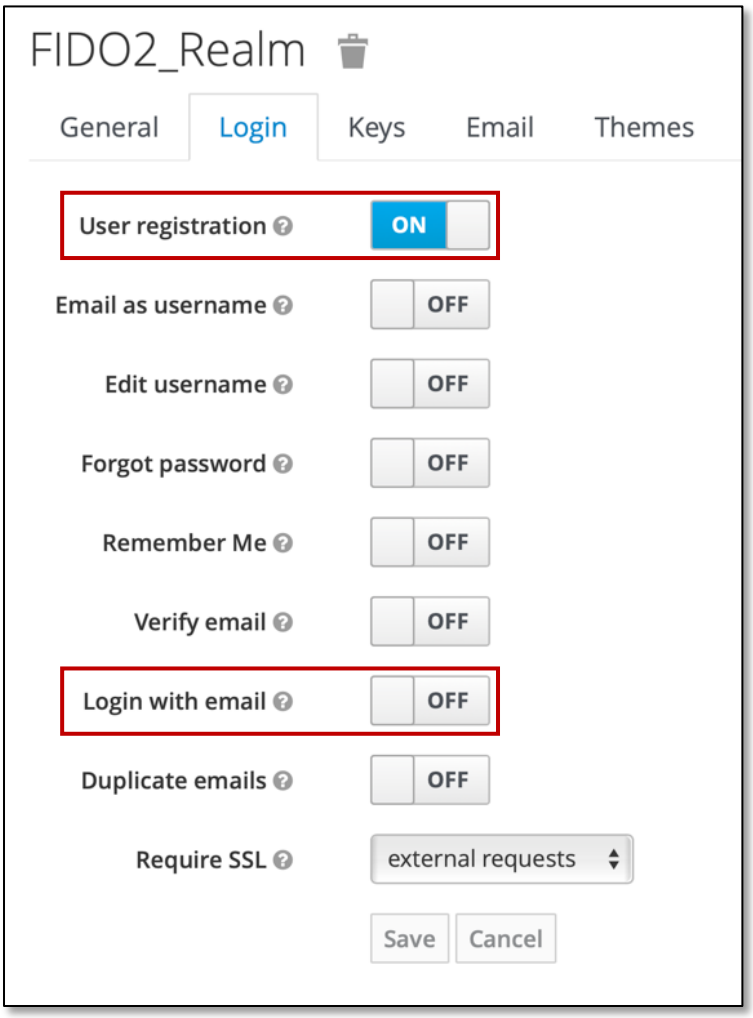

Abb. 5: Realm-Einstellungen – „Login“

Schritt 4: Client anlegen

Damit eine Anwendung auf die Keycloak-Instanz zugreifen kann, muss zuerst ein „Client“ in der Keycloak-Instanz konfiguriert werden. Ein Client repräsentiert eine Anwendung, die eine User-Authentifizierung von der Keycloak-Instanz anfordert.[17] Die Verwaltungsoberfläche für die Clients eines Realms ist über die Menü-Leiste unter „Clients“ erreichbar. In den Client-Einstellungen werden dem Entwickler alle bestehenden Clients in einer Tabelle angezeigt. Die in Abbildung 6 dargestellten Clients sind alle vorinstalliert.

16 Die Deaktivierung von „Login with email“ ist nicht relevant für die Implementierung von FIDO2. Sie wird nur vorgenommen, um die Implementierung in Kapitel 3 schneller und einfacher zu testen.

17 Vgl. Keycloak.org (Hrsg.): Server Administration Guide, a. a. O., abgerufen am 30.12.2020.

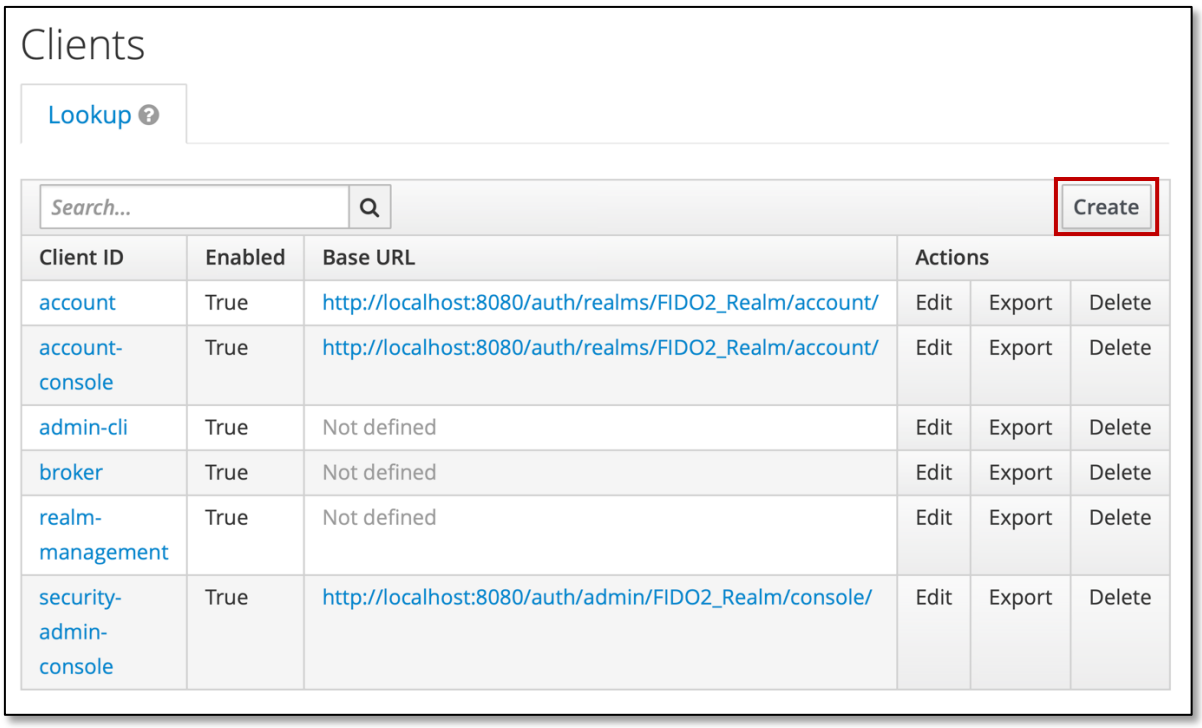

Abb. 6: Die Übersicht aller Clients der Keycloak-Instanz

Die vorinstallierten Clients sollten nicht für den Einsatz bei einer Web-Anwendung modifiziert werden. Über den Button „Create“ am rechten oberen Rand von Abbildung 6 können neue Clients angelegt werden. Es öffnet sich die in Abbildung 7 dargestellte Web-Seite zur Erstellung eines Clients. Der Entwickler muss nun eine eindeutige Client-ID (für OIDC; siehe Kapitel 3 in Arbeitspapier WI Nr. 8/2021) und ein Client-Protokoll wählen. In der vorliegenden Arbeit ist die Client-ID des neuen Clients „FIDO2_Client“. Bei dem Client-Protokoll kann zwischen OIDC und SAML gewählt werden. Da in der vorliegenden Arbeit eine Kombination aus OIDC und FIDO2 betrachtet wird, wählt der Entwickler die Option „openid-connect“ für OIDC. Optional kann der Entwickler eine „Root-URL“ angeben. Diese Root-URL wird für die automatische Konfiguration von Umleitungs-URLs eingesetzt. Ohne Konfiguration einer Root-URL müssten mehr manuelle Konfigurationen vorgenommen werden. In der vorliegenden Arbeit ist die Root-URL „http://localhost:4200“. Das ist die URL der Web-Anwendung, auf die die Keycloak-Instanz nach erfolgreicher Anmeldung weiterleiten soll.

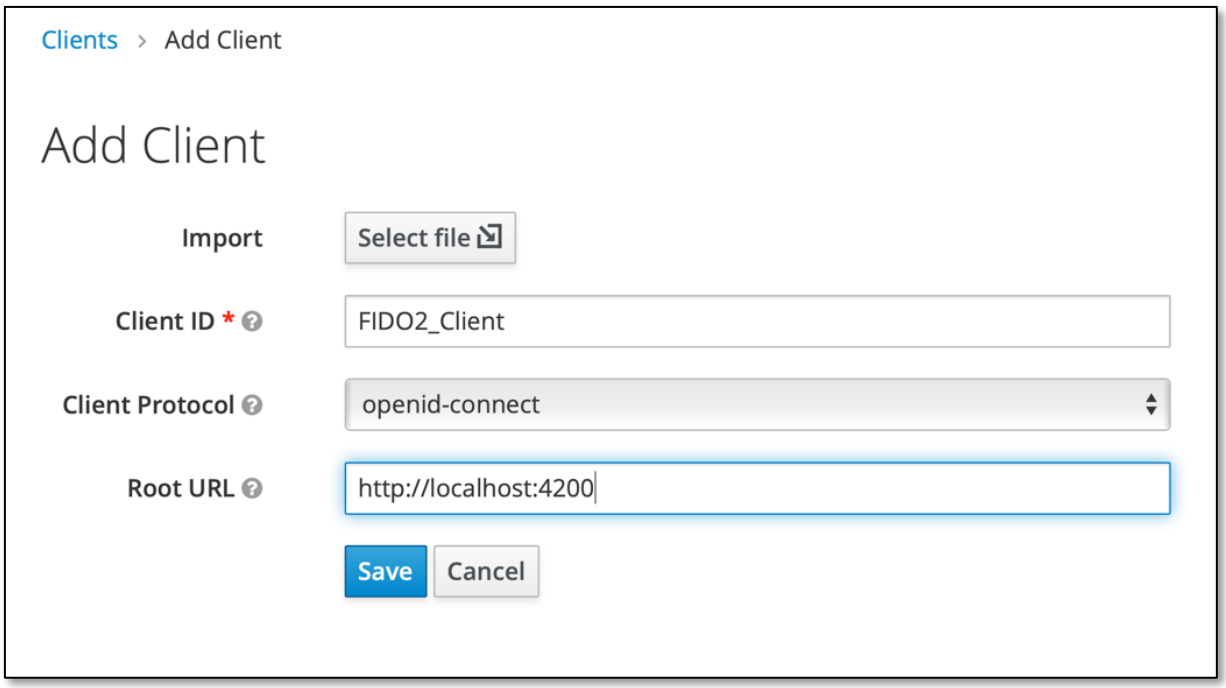

Abb. 7: Der Keycloak-Instanz einen neuen Client hinzufügen

Schritt 5: Authentifizierungs-Flow anlegen

Nach Schritt 4 kann die Keycloak-Instanz bereits für die Authentifizierung eines Users mit Username und Passwort eingesetzt werden. Damit FIDO2 für die Authentifizierung des Users eingesetzt wird, muss ein neuer „Authentifizierungs-Flow“ angelegt werden. Unter „Authentication“ in der Menü-Leiste sind die Konfigurationen der Authentifizierung zu finden; dazu gehören auch Authentifizierungs-Flows. Im Reiter „Flows“ der in Abbildung 8 dargestellten Authentifizierungs-Einstellungen (Menü-Leiste „Authentication“) kann der User bereits vorhandene Authentifizierungs-Flows[18] einsehen und modifizieren, aber auch neue Authentifizierungs-Flows anlegen. Das in Abbildung 8 hervorgehobene Drop-Down-Menü (links oben in Abbildung 8 ist „Browser“ ausgewählt) dient zur Auswahl eines vorhandenen Authentifizierungs-Flows. Der ausgewählte Authentifizierungs-Flow wird in der Tabelle unterhalb des Drop-Down-Menüs angezeigt (siehe Abbildung 8). Der gewählte Authentifizierungs-Flow kann damit konfiguriert werden.

18 Bei einer Keycloak-Instanz sind die folgenden acht Authentifizierungs-Flows vorinstalliert: „Browser“, „Direct Grant“, „Registration“, „Reset Credential“, „Clients“, „First Broker Login“, „Docker Auth“ und „HTTP Challenge“.

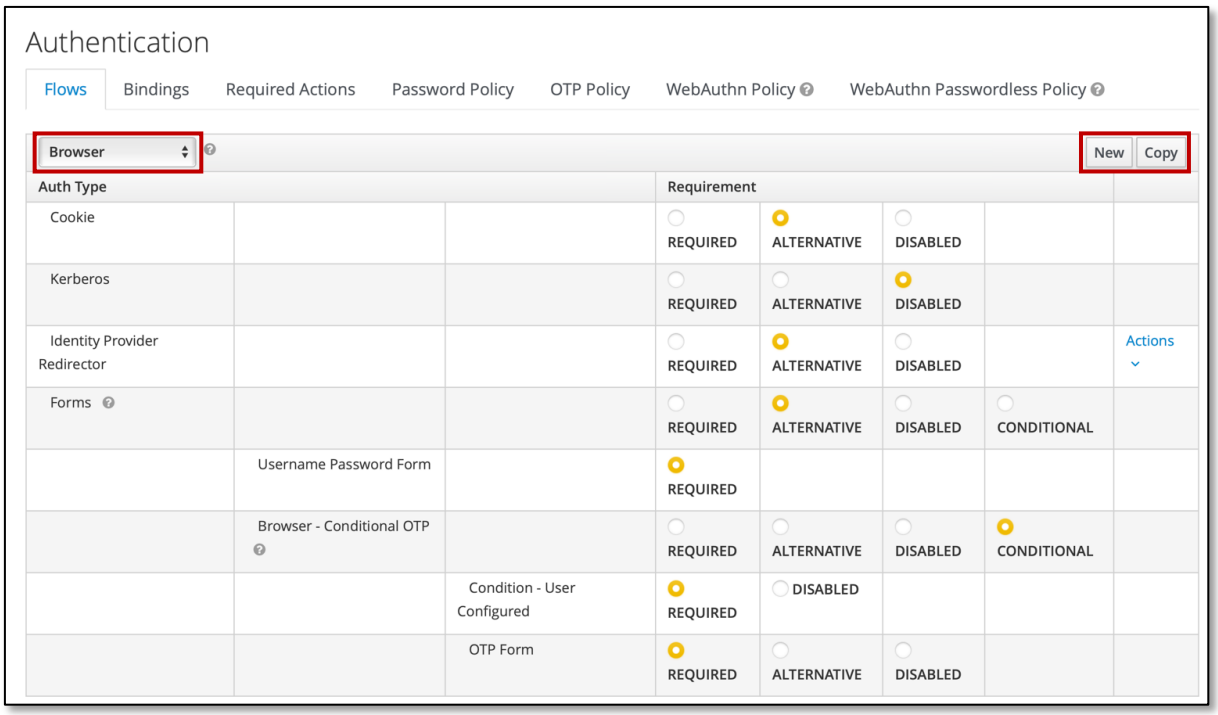

Abb. 8: Konfiguration von Authentifizierungs-Flows

Um einen neuen Authentifizierungs-Flow anzulegen, nutzt der Entwickler den Button „New“ in Abbildung 8. Es ist außerdem möglich, einen bereits vorhandenen Authentifizierungs-Flow über den Button „Copy“ zu duplizieren. In der vorliegenden Arbeit wird der Authentifizierungs-Flow „Browser“ ausgewählt und dupliziert. Der neue Authentifizierungs-Flow muss benannt werden. In der vorliegenden Arbeit wird der neue Authentifizierungs-Flow mit „FIDO2_Flow“ bezeichnet.

Schritt 6: Authentifizierungs-Flow neu konfigurieren

Um mit dem kopierten Authentifizierungs-Flow FIDO2 umzusetzen, werden die überflüssigen Komponenten des Authentifizierungs-Flows „Browser“ aus dem Authentifizierungs-Flow „FIDO2_Flow“ entfernt. Die Komponenten „Kerberos“, „Username Password Form“, „FIDO2_Flow Browser – Conditional OTP“, „Condition – User Configured“ und „OTP Form“ (siehe Abbildung 9) werden gelöscht. Um eine Komponente zu löschen, wird für die betreffende Zeile im Drop-Down-Menü mit Bezeichnung „Actions“ in der Spalte ganz rechts die Option „Delete“ gewählt. Es verbleiben die Komponenten „Cookie“, „Identity Provider Redirector“ und „FIDO2_Flow Forms“.

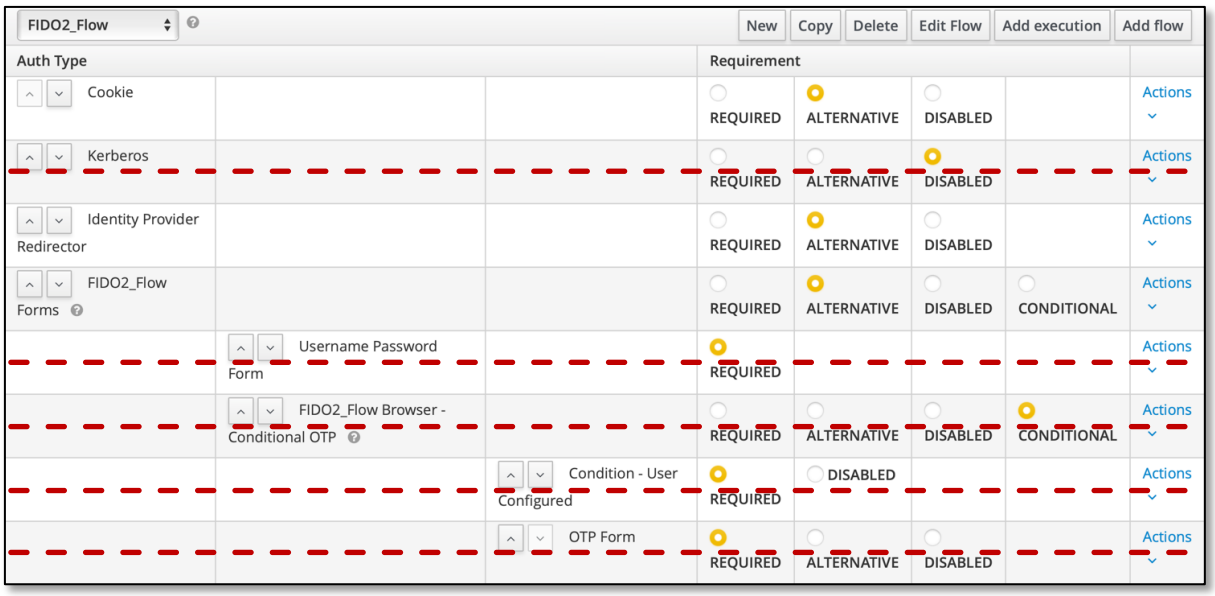

Abb. 9: Browser-Flow-Komponenten aus FIDO2_Flow löschen

Schritt 7: Execution „Username Form“ hinzufügen

Der in Abbildung 10 dargestellte Authentifizierungs-Flow „FIDO2_Flow“ muss um bestimmte Komponenten erweitert werden, damit dieser Flow FIDO2 umsetzt.

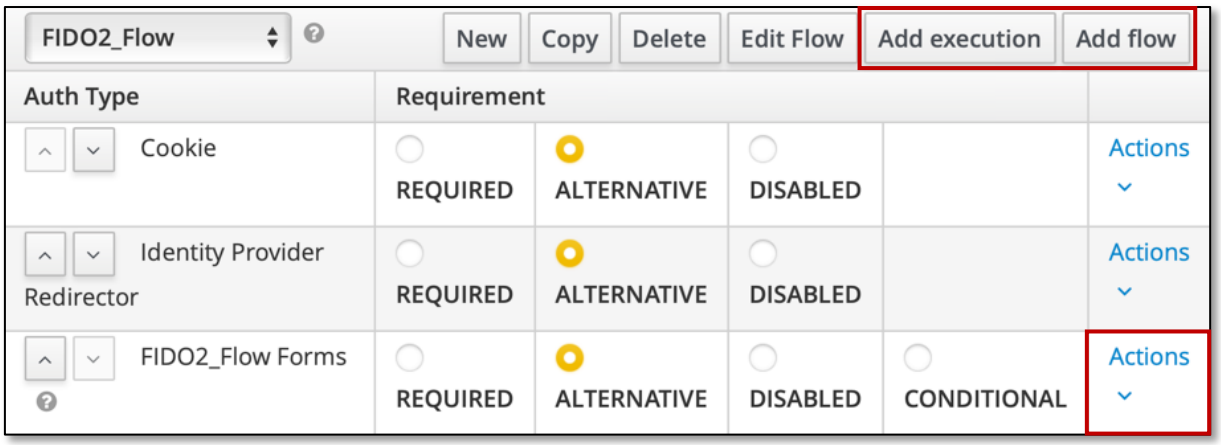

Abb. 10: Ausgangspunkt für den Aufbau des FIDO2_Flows

Komponenten lassen sich in die zwei Kategorien „Sub-Flows“ und „Executions“ einordnen. Sub-Flows (z. B. „FIDO2_Flow Forms“ in Abbildung 10) gruppieren Komponenten, die definierte abgeschlossene Teile des Haupt-Flows (z. B. „FIDO2_Flow“, „Browser“ usw.) umsetzen. Executions sind durch Keycloak vorgefertigte Funktionalitäten, die über Flows gruppiert werden, um eine komplexere Funktionalität umzusetzen. Komponenten können über die Buttons „Add execution“ (für eine Execution) und „Add flow“ (für einen Sub-

Flow) in der rechten oberen Ecke der Abbildung 10 dem Haupt-Flow (FIDO2_Flow) zugefügt werden. Um einem Sub-Flow eine Komponente hinzuzufügen, muss der Entwickler in dem in Abbildung 10 hervorgehobenen Drop-Down-Menü „Actions“ des betreffenden Sub-Flows die Option „Add execution“ (für eine Execution) oder „Add flow“ (für einen Sub-Flow) auswählen.

Um FIDO2 umzusetzen, muss dem Haupt-Flow („FIDO2_Flow“) keine Komponente hinzugefügt werden. Der Sub-Flow „FIDO2_Flow Forms“ wird jedoch nachfolgend mit Executions befüllt, die zusammengenommen FIDO2 umsetzen. Mit der Auswahl von „Add execution“ im „Actions“-Drop-Down-Menü des Sub-Flows „FIDO2_ Flow Forms“ wird die in Abbildung 11 dargestellte Web-Seite aufgerufen. Hier kann aus 33 Executions eine Execution ausgewählt werden. Zuerst muss die Execution „Username Form“ ausgewählt werden (siehe Abbildung 11).

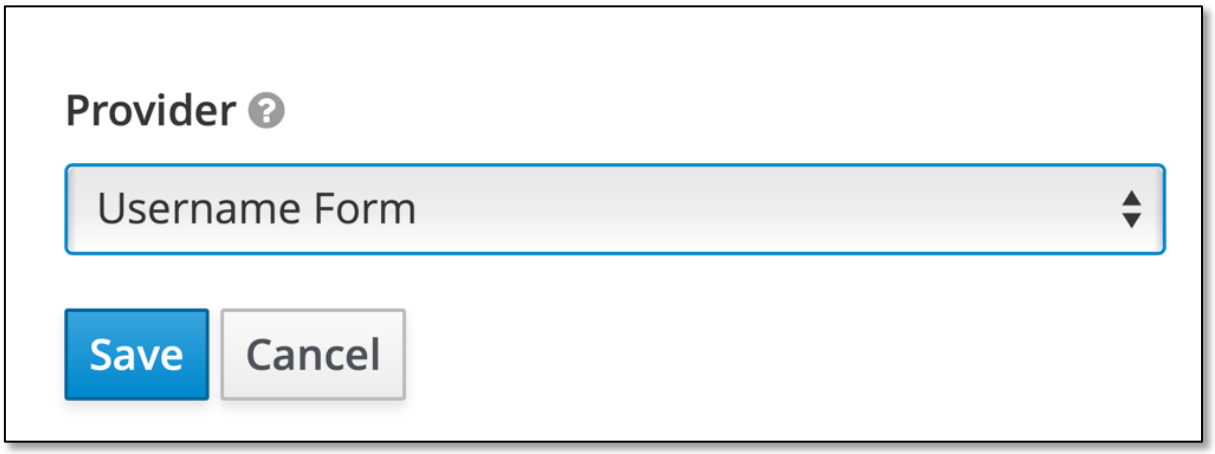

Abb. 11: Execution „Username Form“ hinzufügen

Schritt 8: Execution „WebAuthn Passwordless Authenticator“ hinzufügen

Nach der Execution „Username Form“ muss für FIDO2 noch die Execution „WebAuthn Passwordless Authenticator“ dem Sub-Flow „FIDO2_Flow Forms“ hinzugefügt werden. Dazu wird Schritt 7 mit der Execution „WebAuthn Passwordless Authenticator“ wiederholt. Die Execution „WebAuthn Passwordless Authenticator“ muss noch aktiviert werden. Dazu muss in der Spalte „Requirement“ für diese Execution der Wert „Required“ und nicht „Disabled“ aktiviert sein (siehe Abbildung 12). In Abbildung 12 ist der fertige FIDO2_Flow dargestellt.

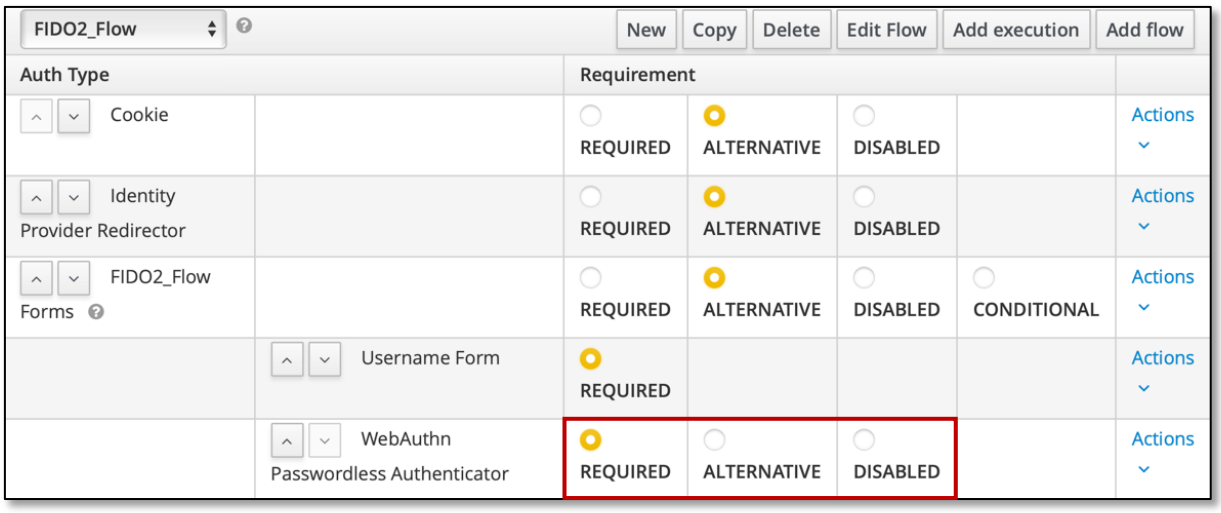

Abb. 12: Der fertige FIDO2_Flow

Schritt 9: Flow-Bindings festlegen

Ein Authentifizierungs-Flow wird erst dann von einer Keycloak-Instanz eingesetzt, wenn der Authentifizierungs-Flow in den „Flow-Bindings“ eingetragen ist (siehe Abbildung 13). Unter dem Reiter „Bindings“ in den Authentifizierungs-Einstellungen (Menü-Leiste „Authentication“) können die Flow-Bindings eingesehen und verändert werden (siehe Abbildung 13). Um die Anmeldung des Users bei der Web-Anwendung mit FIDO2 umzusetzen, muss lediglich der Wert des Flow-Bindings „Browser Flow“ (nicht zu verwechseln mit dem Authentifizierungs-Flow „Browser“) auf „FIDO2_Flow“ gesetzt sein.

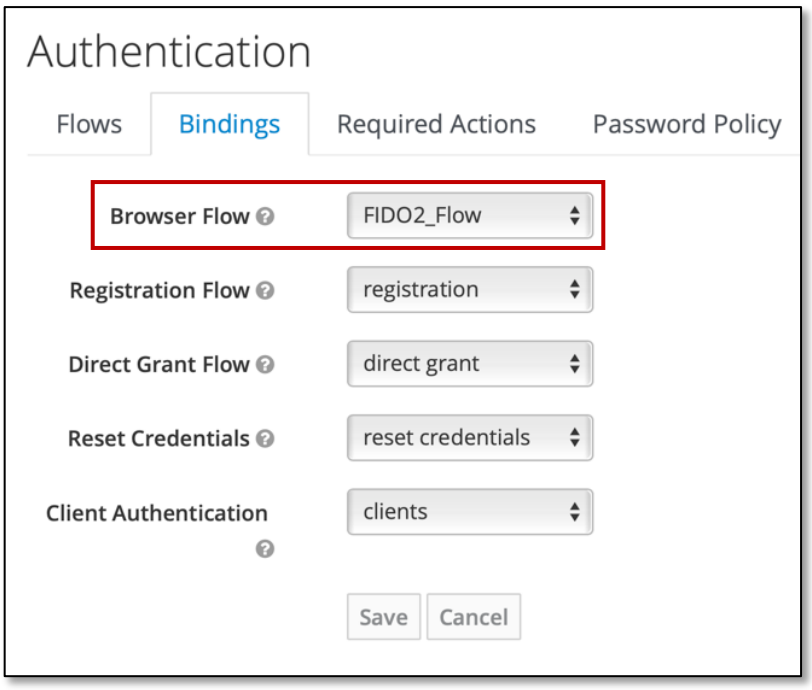

Abb. 13: Die Flow-Bindings

Schritt 10: Einrichten von FIDO2 bei Registrierung erforderlich

Damit ein User FIDO2 zur Anmeldung bei einer Web-Anwendung einsetzen kann, muss der User sich bei der Web-Anwendung auch mit FIDO2 registrieren. Um die Registrierung des Users mit FIDO2 umzusetzen, muss kein zusätzlicher Authentifizierungs-Flow angelegt werden. Lediglich im Reiter „Required Actions“ bei den Authentifizierungs-Einstellungen (Menü-Leiste „Authentication“) müssen einige Änderungen vorgenommen werden. Die Required Actions sind Aktionen, die ein User in der Web-Anwendung durchführen kann („Enabled“), um seinen User-Account zu modifizieren (z. B. Passwort ändern, E-Mail-Adresse verifizieren usw.). Außerdem können Required Actions als verbindlich („Default Action“) festgelegt werden. In Abbildung 14 werden die voreingestellten Konfigurationen gezeigt.

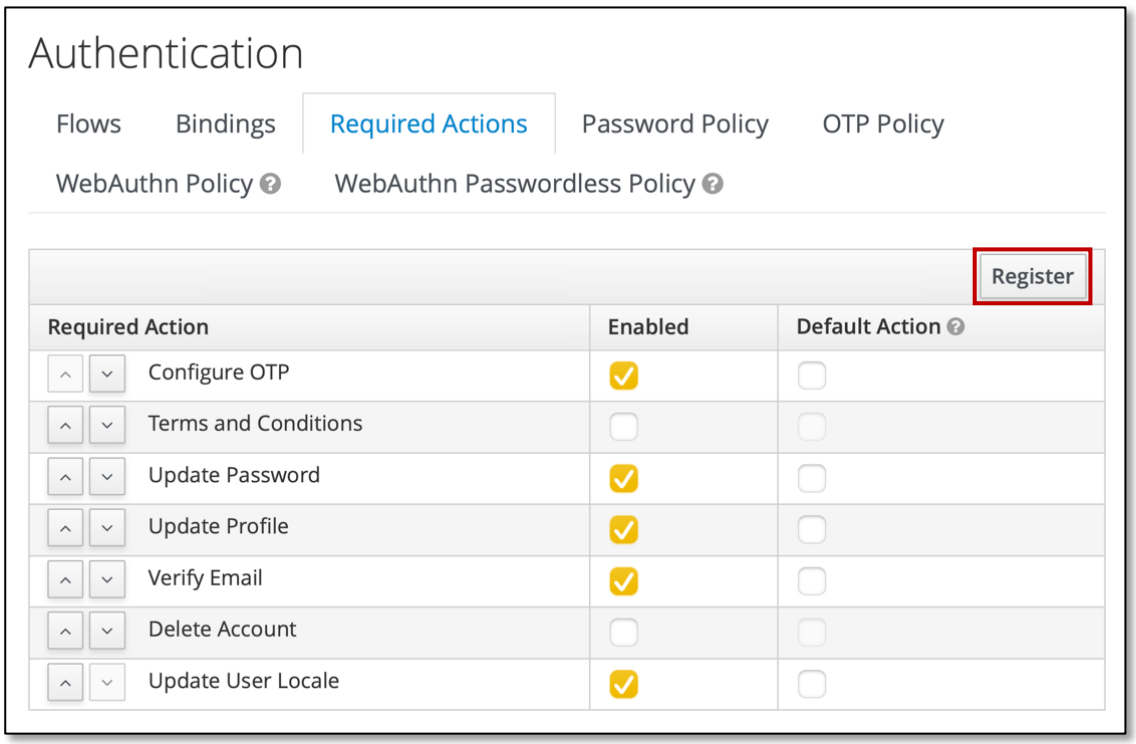

Abb. 14: Authentifizierungs-Einstellungen – „Required Actions“

Für die Implementierung von FIDO2 muss eine neue Required Action hinzugefügt werden. Über den Button „Register“ am oberen rechten Rand von Abbildung 14 wird ein Pop-Up-Fenster getriggert, in dem eine neue Required Action ausgewählt werden kann (siehe Abbildung 15). Für die Implementierung von FIDO2 muss die Required Action „Webauthn Register Passwordless“ gewählt werden (siehe Abbildung 15).

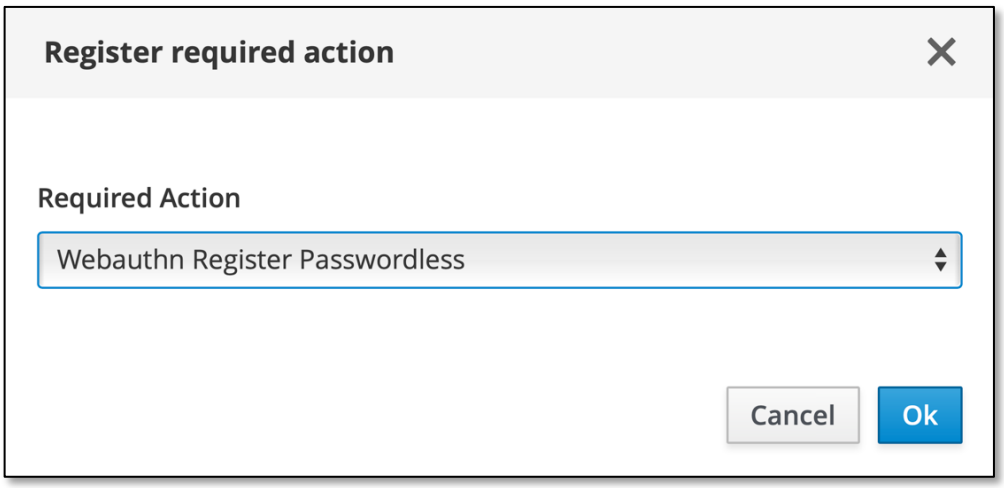

Abb. 15: Hinzufügen einer neuen Required Action

Die Required Action „Webauthn Register Passwordless“ muss als verbindlich festgelegt werden. Dazu müssen die in Abbildung 14 dargestellten Voreinstellungen lediglich an der in Abbildung 16 hervorgehobenen Stelle per Mausklick geändert werden.

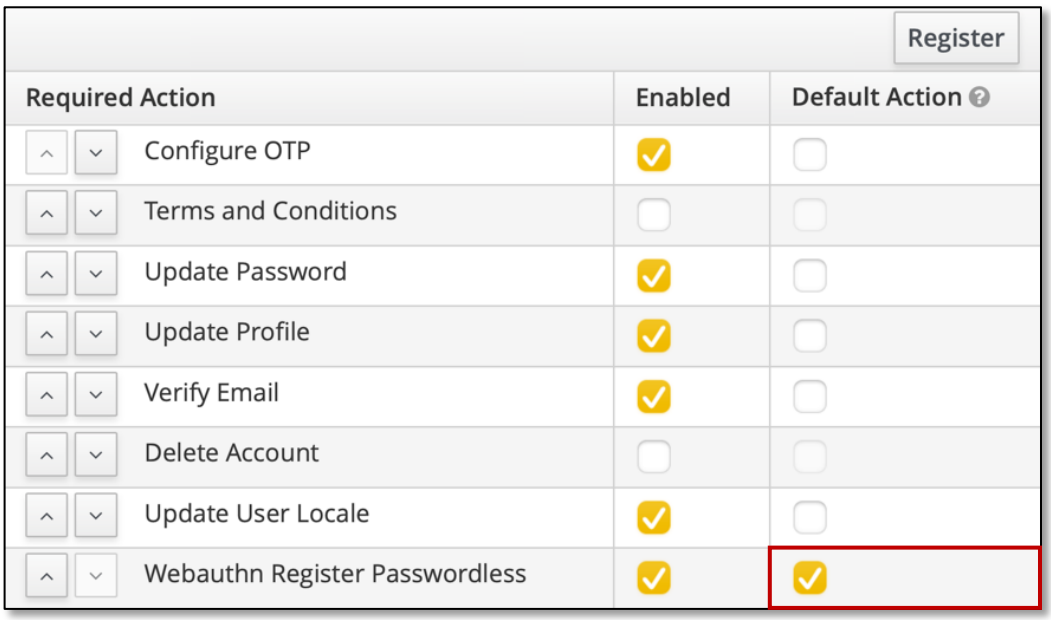

Abb. 16: FIDO2 ist verbindlich für die Registrierung

Schritt 11: Password bei Registrierung entfernen

Damit ein User bei der Registrierung für die Web-Anwendung kein Password festlegen muss, wird ein weiterer Konfigurationsschritt benötigt. Dafür muss im Reiter „Flows“ der Authentifizierungs-Einstellungen (Menü-Leiste „Authentication“) der Authentifizierungs-Flow „Registration“ ausgewählt werden. Um Passwort-Eingaben aus dem Registrierungs-

vorgang zu entfernen, muss in der Spalte „Requirement“ für diese Execution der Wert „Disabled“ und nicht „Required“ aktiviert sein (siehe Abbildung 17).

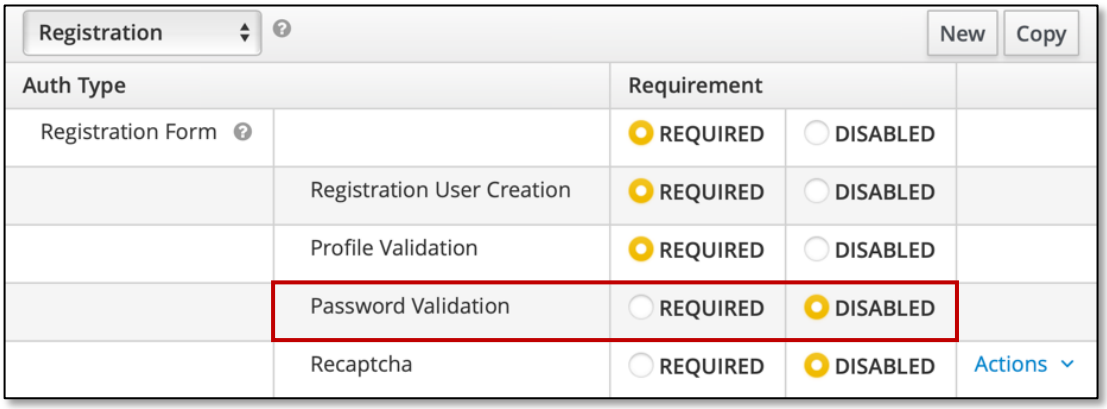

Abb. 17: Der Authentifizierungs-Flow „Registration“

Eine Keycloak-Instanz gibt Entwicklern die Möglichkeit, die Authentifizierung mit FIDO2 zusätzlich zu modifizieren. Im Reiter „WebAuthn Passwordless Policy“ der Authentifizierungs-Einstellungen (Menü-Leiste „Authentication“) können weitere Konfigurationen (z. B. UV-Anforderung; siehe Kapitel 4 in Arbeitspapier WI Nr. 8/2021) an FIDO2 vorgenommen werden. Diese Konfigurationen sind jedoch nur optional, um FIDO2 mit einer Keycloak-Instanz umzusetzen. Deshalb werden diese weiteren Konfigurationen in der vorliegenden Arbeit nicht betrachtet. Die Konfiguration der Keycloak-Instanz ist damit abgeschlossen

3 Test der prototypischen Implementierung von FIDO2

Nachfolgend soll in diesem Kapitel die prototypische Implementierung von FIDO2 getestet werden (ein Funktionalitäts-Test, kein Sicherheits-Test o. ä.). Dazu wird eine einfache Web-Anwendung (entwickelt mit Angular) über den localhost auf Port 4200 gehostet, während die Keycloak-Instanz aus Kapitel 2 über den localhost auf Port 8080 erreichbar ist. Für den Test nutzen die Verfasser der vorliegenden Arbeit einen Google Titan Security-Key (siehe Abbildung 18) als Authentifikator. Dieser Titan Security-Key wird über USB mit dem Endgerät der Verfasser der vorliegenden Arbeit verbunden.

Abb. 18: Ein Google Titan Security-Key (Authentifikator in Originalgröße)19

Beim Aufruf der URL der Web-Anwendung („http://localhost:4200“) wird der User direkt auf die Keycloak-Instanz („http://localhost:8080“) weitergeleitet. Dem User wird eine von Keycloak mitgelieferte Anmeldemaske gezeigt, über die der User sich beim „FIDO2_Realm“ anmelden und Registrieren kann (siehe Abbildung 19). Der User hat keinen Account bei der Web-Anwendung. Deshalb muss er den Button „Register“ betätigen und einen User-Account registrieren.

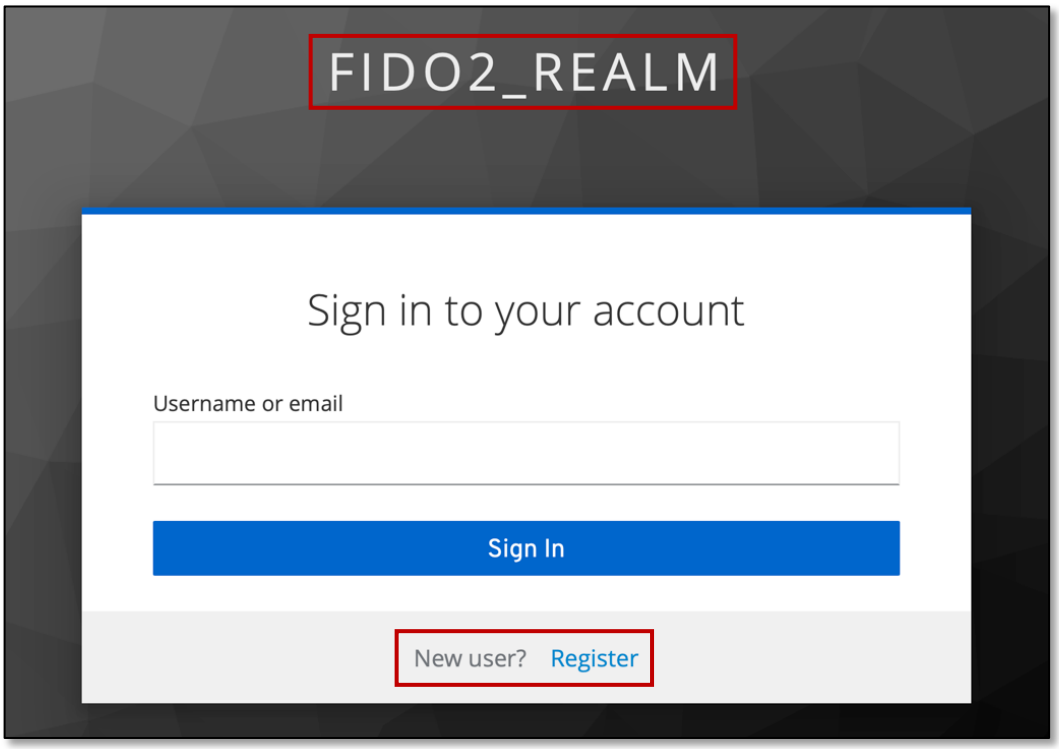

Abb. 19: Anmeldemaske der Web-Anwendung

19 Google Ireland Limited (Hrsg.): Titan Security Key, https://store.google.com/product/titan_security_key, abgerufen am12.03.2021.

Der User wird auf die in Abbildung 20 dargestellte Web-Seite geleitet, wo er die Daten für seinen Account eingeben muss.[20] Mit dem Button „Register“ wird der OIDC-Registrierungsvorgang (siehe Kapitel 3 in Arbeitspapier WI Nr. 8/2021) durchgeführt.

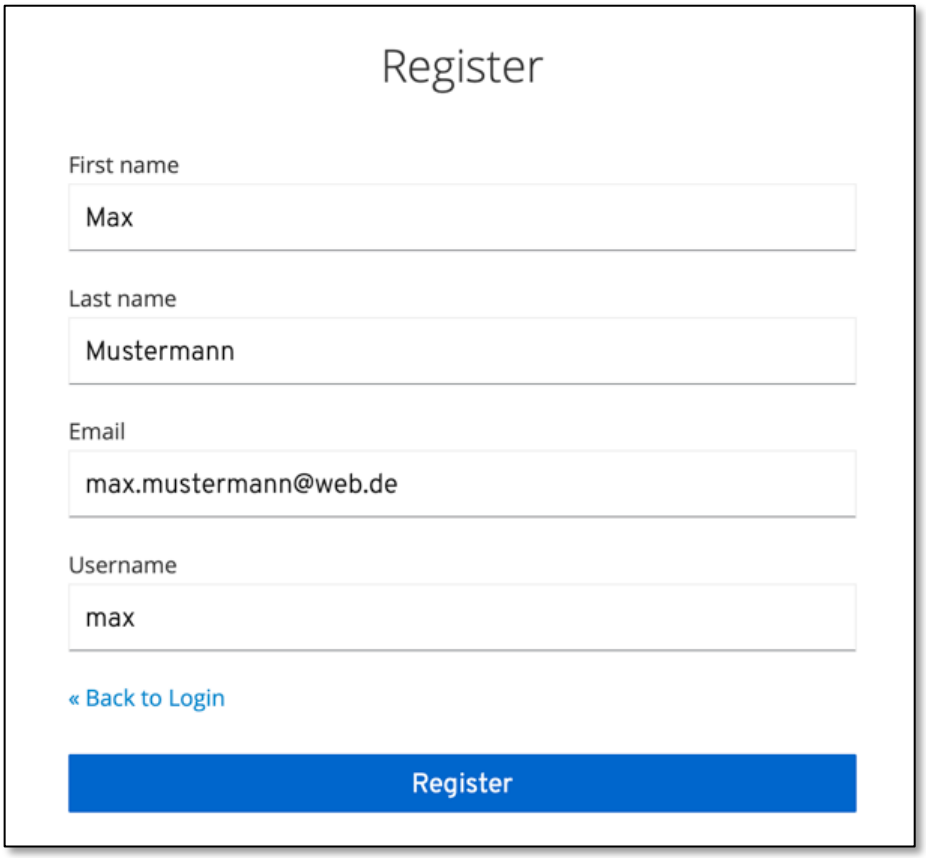

Abb. 20: Registrierung – Account-Daten angeben

Auf Grund von Konfigurationsschritt 10 in Kapitel 2 startet nach dem OIDC-Registrierungsvorgang automatisch der Registrierungsvorgang von FIDO2 (siehe Kapitel 5 in Arbeitspapier WI Nr. 8/2021). Zuerst muss der User den FIDO2-Authentifikator mit seinem Endgerät verbinden und freischalten (siehe Abbildung 21). Der User verbindet seinen Authentifikator mit seinem Endgerät (über USB, Bluetooth oder NFC) und betätigt den Knopf auf dem Authentifikator. Wenn der Authentifikator User Verification (siehe Kapitel 5 in Arbeitspapier WI Nr. 8/2021) ermöglicht, muss der User sich per PIN oder Biometrie lokal authentifizieren, um den Authentifikator freizuschalten.

20 Durch den Konfigurationsschritt 11 in Kapitel 2, muss der User hier kein Password mehr festlegen. Wenn Konfigurationsschritt 11 nicht durchgeführt wird, muss der User hier ein Password erfassen. Dieses Passwort kann jedoch nicht für Anmeldevorgänge eingesetzt werden, da die Authentifizierung des Users der Web-Anwendung hier ausschließlich mit FIDO2 abläuft.

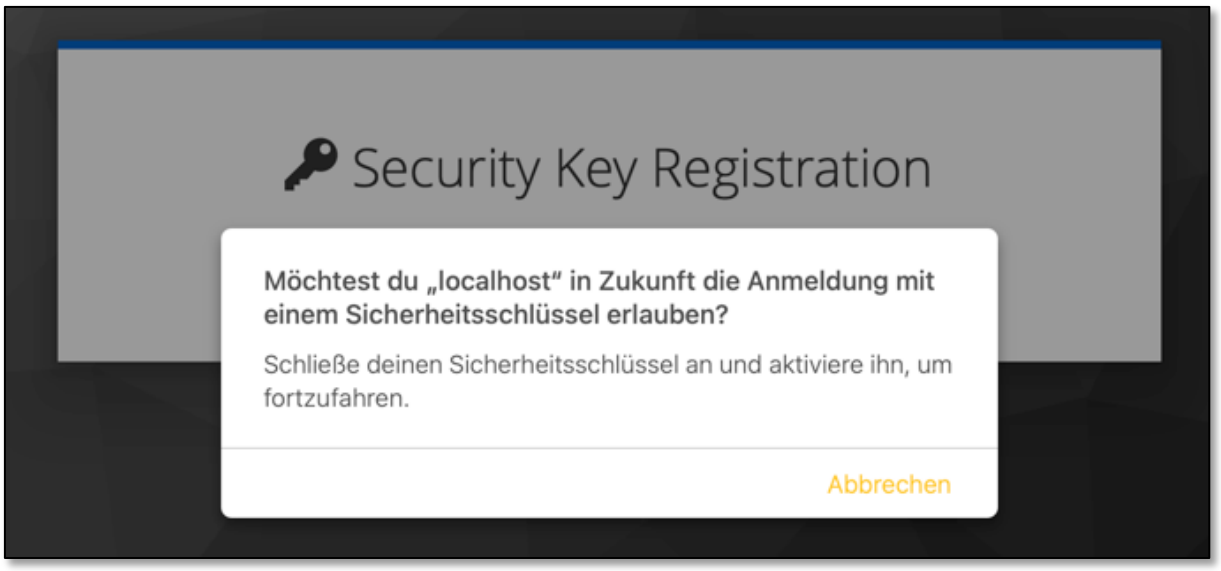

Abb. 21: Registrierung – Authentifikator verbinden und freischalten

Nachdem der Authentifikator freigeschaltet wurde, kann der User den Account auf dem Authentifikator optional noch benennen, um Accounts auf dem Authentifikator durch Bezeichnungen unterscheiden zu können (siehe Abbildung 22).

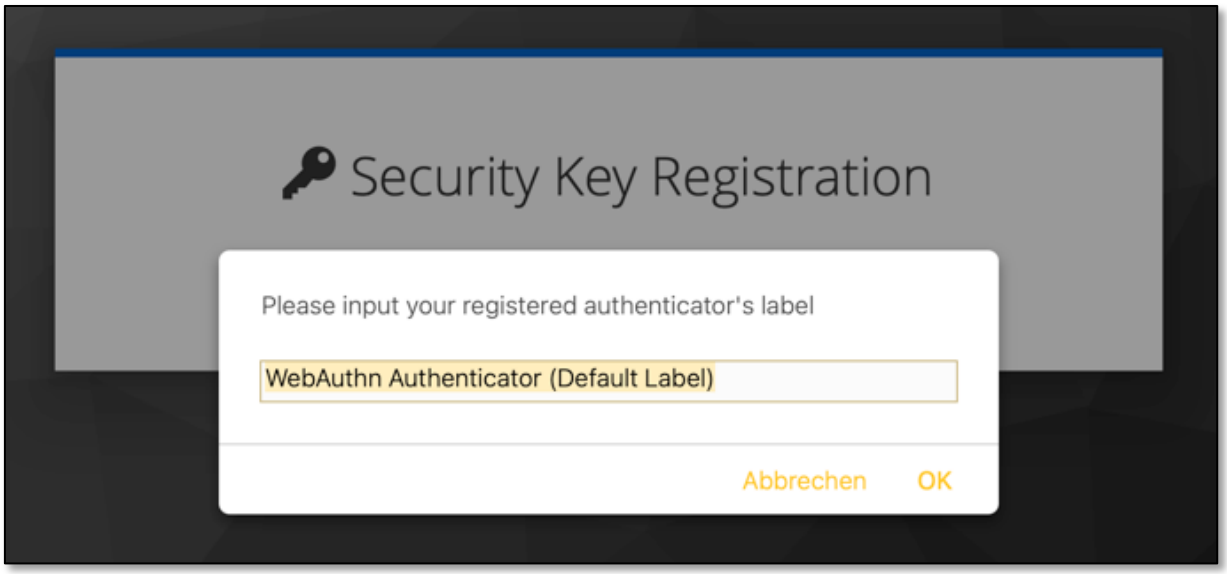

Abb. 22: Registrierung – Account-Bezeichnung festlegen

Damit ist der Registrierungsvorgang von FIDO2 abgeschlossen. Nach dem erfolgreichen Registrierungsvorgang wird der User von der Keycloak-Instanz automatisch bei der Web-Anwendung angemeldet. Abbildung 23 zeigt die Startseite der Web-Anwendung, bei der

sich der User mit Username „max“ registriert hat.[21] Die Web-Anwendung fragt die User-Daten und einen OIDC ID-Token (siehe Kapitel 3 und Anhang B.2 in Arbeitspapier WI Nr. 8/2021) von der Keycloak-Instanz (Auth-Server) ab und zeigt die User-Daten und den ID-Token an.

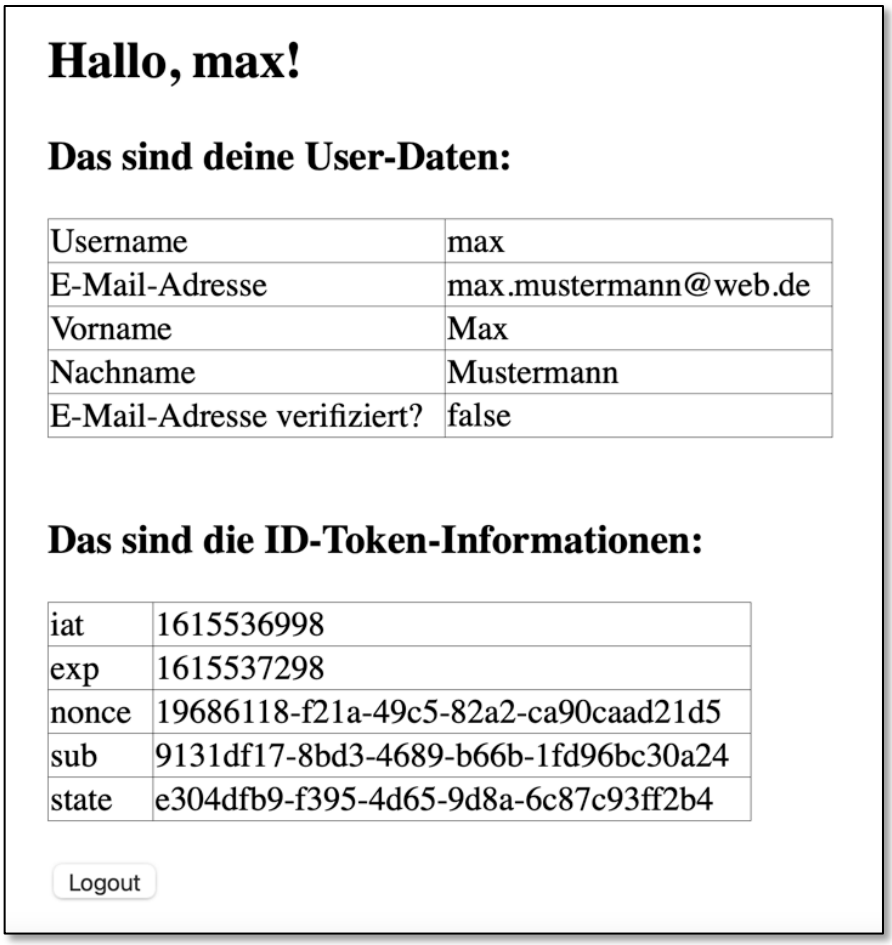

Abb. 23: Die Startseite der Web-Anwendung

Um nach dem geschilderten Registrierungsvorgang den Anmeldevorgang separat zu testen, muss der User sich zuerst bei der Web-Anwendung abmelden. Durch Betätigen des Buttons „Logout“ in Abbildung 23 wird der User abgemeldet und automatisch zur Anmeldemaske (siehe Abbildung 19) geleitet. Der User gibt seinen Usernamen ein und startet über den Button „Sign In“ den OIDC-Anmeldevorgang (siehe Abbildung 8 in Kapitel 3 des Arbeitspapiers WI Nr. 8/2021). Sobald der User sich im OIDC-Anmeldevorgang authentifizieren muss, wird der User auf die in Abbildung 24 dargestellte Web-Seite geleitet. Der

21 Diese Web-Anwendung wurde von den Verfassern der vorliegenden Arbeit mit Angular gebaut.

User aktiviert seinen Authentifikator und muss diesen per Knopfdruck und optional per PIN oder Biometrie freischalten.

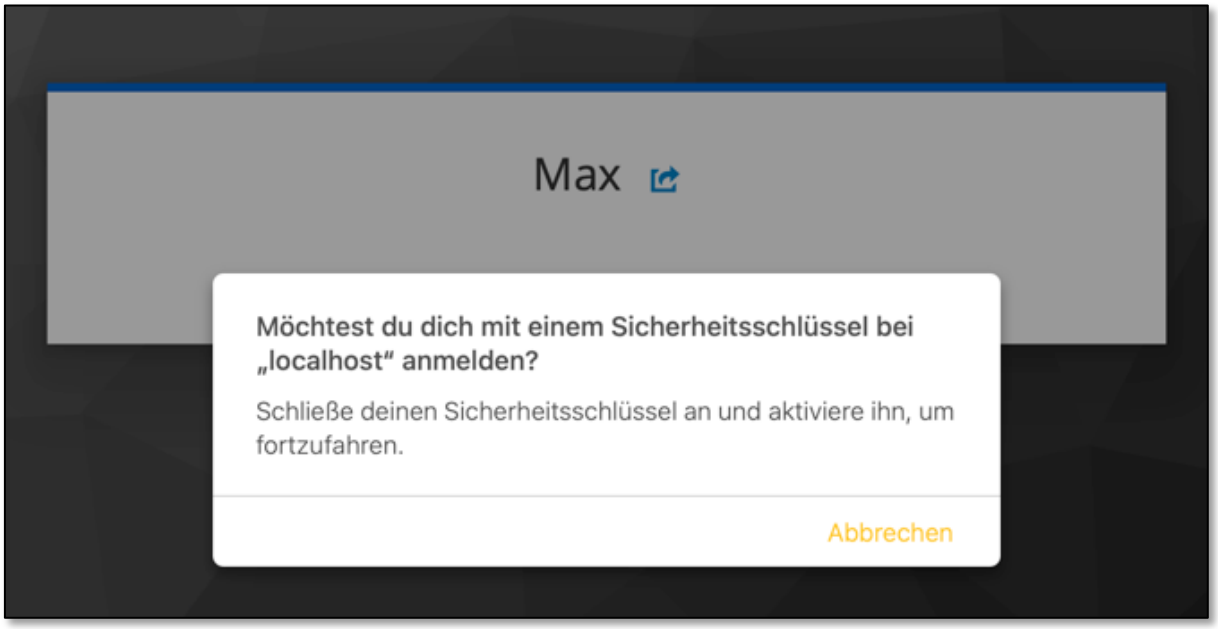

Abb. 24: Anmeldung mit einem FIDO2-Authentifikator

Da der User bei dem OIDC-Anmeldevorgang in der vorliegenden Arbeit vor der Authentifizierung mit FIDO2 seinen Usernamen angegeben hat, kann der Authentifikator den passenden User-Account für die Web-Anwendung aus seinem Account-Bestand ohne Zutun des Users auswählen.[22] Die Authentifizierung mit FIDO2 ist damit abgeschlossen und der OIDC-Anmeldevorgang wird gemäß Abbildung 8 in Kapitel 3 des Arbeitspapiers WI Nr. 8/2021 mit der Autorisierung durch das Ausstellen eines Access-Tokens fortgesetzt. Sobald der OIDC-Anmeldevorgang abgeschlossen ist, wird dem User Zugang zu der in Abbildung 23 dargestellten Web-Anwendung gewährt.

22 Es gibt nur ein Schlüsselpaar für einen Account bei einer bestimmten Web-Anwendung auf einem Authentifikator (siehe Kapitel 5 in Arbeitspapier Nr. 8/21).

Literaturverzeichnis

1. Amazon Web Services Inc. (Hrsg.): Amazon Cognito Entwicklerhandbuch, 2021, https://docs.aws.amazon.com/de_de/cognito/latest/developerguide/cognito-dg.pdf#what-is-amazon-cognito, abgerufen am 11.03.2021.
2. Angular.io (Hrsg.): What is Angular?, https://angular.io/guide/what-is-angular, abgerufen am 09.03.2021.
3. Apache Software Foundation (Hrsg.): Apache License Version 2.0, Januar 2004, https://www.apache.org/licenses/LICENSE-2.0.txt, abgerufen am 01.03.2021.
4. Auth0 Inc. (Hrsg.): Get Started, https://auth0.com/docs/get-started, abgerufen am 11.03.2021.
5. Burke, Bill: Keycloak 1.0 Final Released, lists.jboss.org (Hrsg.), 11. September 2014, https://lists.jboss.org/archives/list/keycloak-dev@lists.jboss.org/thread/U3KXSAUJSN4TONLRLFCKA3EZWWEDW7NZ/, abgerufen am 13.03.2021.
6. Campbell, Brian et al.: Security Assertion Markup Language (SAML) 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants, Internet Engineerig Task Force (Hrsg.), Mai 2015, https://tools.ietf.org/pdf/rfc7522.pdf, abgerufen am 09.03.2021.
7. Docker Inc. (Hrsg.): What is a Container? – A standardized unit of software, https://www.docker.com/resources/what-container, abgerufen am 09.03.2021.
8. Eastlake, Donald; Panitz, Aliza: Reserved Top Level DNS Names, Juni 1999, https://tools.ietf.org/html/rfc2606, abgerufen am 10.03.2021.
9. Google Ireland Limited (Hrsg.): Titan Security Key, https://store.google.com/product/titan_security_key, abgerufen am12.03.2021.
10. Keycloak.org (Hrsg.): Release Notes, https://www.keycloak.org/docs/latest/release_notes/index.html, abgerufen am 11.03.2021.
11. Keycloak.org (Hrsg.): Server Administration Guide, https://www.keycloak.org/docs/latest/server_admin/, abgerufen am 30.12.2020.

12. Koserwal, Abhishek: Keycloak: Core concepts of open source identity and access management, Red Hat Inc. (Hrsg.), 11. Dezember 2019, https://developers.redhat.com/blog/2019/12/11/keycloak-core-concepts-of-open-source-identity-and-access-management/, abgerufen am 01.03.2021.

13. Kumar, Anjan: Keycloak Integration in Angular Application, dev.to (Hrsg.), 05. Februar 2021, https://dev.to/anjnkmr/keycloak-integration-in-angular-application-5a43, abgerufen am 09.03.2021.

14. Reinhardt, Martin: IT-Systemzugriffe verwalten: Identity und Access Management mit Keycloak, in: iX 12/20, Heise Medien GmbH & Co. KG (Hrsg.), S. 108-114.

15. Saeed, Hasnat: Setup Keycloak Server on Ubuntu 18.04, Medium.com (Hrsg.), 31. Juli 2019, https://medium.com/@hasnat.saeed/setup-keycloak-server-on-ubuntu-18-04-ed8c7c79a2d9, abgerufen am 08.03.2021.

Impressum

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

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