Anwendungen und Scopes im DHÖS

IAM - Identity and Access Management

Exported on 10/30/2024

Table of Contents

Inhalt

1 Architektur

Das DHÖS ist in mehreren Ebenen unterteilt, wobei von jeder Ebene nur auf die direkt darunterliegende zugegriffen wird – teilweise mit einem API-Gateway dazwischen:

1	Benutzerinterfaces (z.B. Mobile-Apps, Web-UIs, Rich-Client, ...)		
	API-Gateway		
2	Fachservices		
3	Fachobjektservices	Technische Suchservices	
	API-Gateway		
4	DHÖ-Primary	DHÖ-Secondary	ZIMP

2 Hintergrund

AW-Rechte im F-IAM hängen immer an einer Anwendung.

Pro Anwendung können mehrere Scopes definiert werden. Besitzt eine Anwendung nur einen Scope, wird i.d.r. "XXX.Main" verwendet, wobei "XXX" der Name der Anwendung ist.

Jeder Zugriff auf eine andere Anwendung/Webservice muss durch ein Access-Token abgesichert werden, wobei der zugehörige Scope im Token enthalten sein muss.

Jedes Token darf nur die Scopes **einer** Anwendung beinhalten. (Grund: IT-Sicherheit sowie Größenbeschränkung der Header und damit der Tokens)

Für jedes Access-Token ist ein Token-Exchange erforderlich.

I.d.R. werden Anwendungen und Services auf mehrere Services der darunterliegenden Ebene zugreifen.

Um einen Token-Exchange durchführen zu können, muss die Anwendung bzw. der Service als OIDC-Client am F-IAM angebunden sein.

Für jeden OIDC-Client wird im F-IAM konfiguriert, für welche Scopes er Tokens anfragen kann: per Auth-Code-Flow (i.d.R. nur genau den eigenen) oder per Token-Exchange (für Webservice-Requests).

Um die Anzahl der notwendigen Token-Exchanges (und damit Latenzen) zu reduzieren, soll nicht jeder Service als eigene Anwendung konfiguriert werden. Stattdessen werden Services möglichst zusammengefasst, sodass mit einem einzelnen Token auf mehrere Services zugegriffen werden kann.

TODO: Im Rahmen des Auth-Code-Flow muss immer dieselbe Instanz die Tokenanfrage vornehmen, die auch die Authentifizierungsanfrage gestartet hat, denn zum einen muss das Secret des OIDC-Client bekannt sein, zum anderen muss die Code-Challenge des PKCE bekannt sein, die dynamisch für jede Anfrage generiert wird. Der Ansatz, dass das API-Gateway vor den Fachservices die Tokenanfragen für alle Fachservices vornimmt, wird also höchstwahrscheinlich nicht funktionieren.

3 Zuschnitt der Anwendungen und Scopes

3.1 Ebene 1: Benutzerinterface

Auf Ebene 1 ist jede Entität eine eigene Anwendung mit eigenem OIDC-Client, eigenem Scope (i.d.R. XXX.Main) und eigenen Rechten.

Die Anwendungen können jeweils auf verschiedene Fachservices zugreifen, für die sie im F-IAM freigeschaltet werden müssen.

Pro Fachservice-Gruppe (= Domäne = Anwendung) und pro Session benötigen sie einen Access-Token, den sie per Token-Exchange vom F-IAM abfragen müssen. Ein Caching der Access-Tokens kann sinnvoll sein.

3.2 Ebene 2: Fachservices

Die Fachservices werden in Domänen gruppiert, was dann jeweils einer Anwendung entspricht. Folglich sind auch die Rechte immer jeweils einer Domäne zugeordnet, so dass auch verschiedenen Servcies derselben Domäne dieselben Rechte auswerten könnten.

Jeder Service besitzt einen eigenen Scope, so dass der Zugriff darauf individuell konfiguriert werden kann: <Domäne>.<Service>

Jeder Service wird als eigener OIDC-Client angebunden, so dass sein Zugriff auf andere Services individuell konfiguriert werden kann.

3.3 Ebene 3: Fachobjektservices

Pro Fachobjekt soll es vier Fachobjektservices geben – jeweils einen für die CRUD-Operationen.

Die Fachobjektservices werden zusammen als einzelne Anwendung betrachtet, wobei jeder Fachobjektservice einen eigenen Scope besitzt.

Da alle Fachobjektservices in der darunterliegenden Ebene lediglich auf DHÖ-Primary zugreifen, ist eine individuelle Konfiguration der Scopes nicht erforderlich, so dass die Gesamtheit der Fachobjektservices als einzelner OIDC-Client konfiguriert werden, die per Token-Exchange auf DHÖ-Primary zugreifen können.

TODO: Greifen alle Fachobjektservices tatsächlich lediglich auf DHÖ-Primary zu? Anderenfalls wäre zu prüfen, ob eine Aufteilung in verschiedene OIDC-Clients sinnvoll wäre.

TODO: Die Anzahl der möglichen Rechte kann die Tokengröße sprengen, so dass die Rechte per userprofile dynamisch abgefragt werden müssen. Das könnte den Vorteil des eingesparten Token-Exchange durch den aufrufenden Services zunichte machen. Ggf. kann aber ein Caching innerhalb der Gesamtheit der Fachobjektservices (nicht pro Fachobjektservice) implementiert werden. Dies wäre zu prüfen.

3.4 Ebene 3: Technische Suchservices

TODO: Der Zuschnitt der TSS ist noch nicht weiter diskutiert worden. Wie viele wird es geben? Wer greift auf sie zu? Worauf greifen sie zu?

3.5 Ebene 4: DHÖ-*

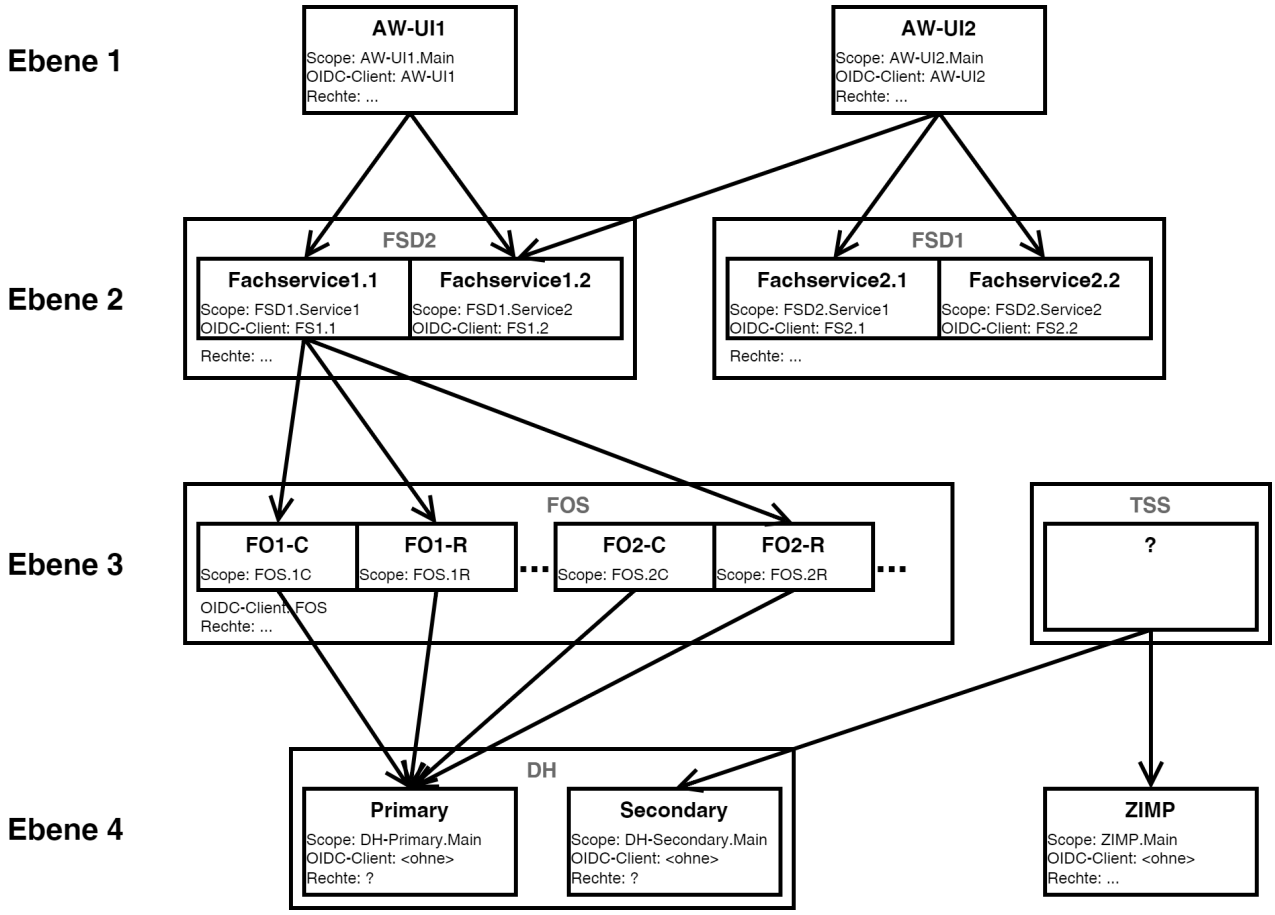
DH-Primary und Secondary sind jeweils eigenständige Anwendungen mit eigenem Main-Scope und eigener Rechteliste. Da die Service keine weiteren P2ß-Serviceaufrufe vornehmen, benötigen sie keinen Token-Exchange und müssen somit somit auch nicht als OIDC-Clients angebunden werden.

TODO: Wurde im Workshop nicht explizit gesprochen. Passt das so?

3.6 Ebene 4: ZIMP

ZIMP ist schon jetzt als eigenständige Anwendung mit einem einzelnen Scope (ZIMP.Main) und eigenen Rechten am F-IAM angebunden. Es können also mit einem einzelnen Token auf sämtliche dahinterliegenden Register Abfragen vorgenommen werden.

4 Visualisierung



4.1 Ebene 1

Es gibt zwei AWs auf UI-Ebene: AW-UI1 und AW-UI2, jeweils mit eigenem Scope, eigenen Rechten und eigenständig als OIDC-Client am F-IAM angebunden.

4.2 Ebene 2

Es gibt zwei Fachservicedomänen mit jeweils zwei Services. Jeder Fachservice hat einen eigenen Scope und ist eigenständig als OIDC-Client am F-IAM angebunden. Die Rechte hängen an der Fachservicedomäne.

4.3 Zugriffe Ebene 1 auf Ebene 2

Da AW-UI1 nur auf Services der FSD1 zugreift, muss er nur einen einzelnen Token-Exchange durchführen und kann mit dem dem FSD1-Token beide Fachservices aufrufen.

AW-UI2 benötigt zwei separate Access-Tokens für die beiden Fachservicedomänen.

4.4 Ebene 3

Für jede CRUD-Operation auf jedem Fachservice gibt es einen eigenen Service mit eigenen Scope, Sie alle teilen sich sich die Anbindung am F-IAM als einzelner OIDC-Client sowie eine gemeinsame Liste von Recten.

4.5 Zugriff Ebene 2 auf Ebene 3

Jeder Fachservice muss lediglich nur einen einzelnen Token-Exchange durchführen, da alle FOS-Scopes zu derselben Anwendung gehören und somit in einem einzelnen Token enthalten sein dürfen. Mit diesem Token können dann sämtliche Fachobjektservices aufgerufen werden, für die der jeweils Fachservice freigeschaltet ist. (Tatsächlich darf er auch nur für genau diese Scopes das Token anfordern.)

4.6 Ebene 4

Alle Services sind jeweils eigenständige Anwendungen mit eigenen Main-Scope und eigenen Rechten. Da die Service keine weiteren P2ß-Serviceaufrufe vornehmen, benötigen sie keinen Token-Exchange und müssen somit somit auch nicht als OIDC-Clients angebunden werden.

4.7 Zugriff Ebene 3 auf Ebene 4

Alle Fachobjektservices müssen einen einzelnen Token-Exchange vornehmen, um ein Token für DH-Primary zu erhalten.