Eine Architektur für mobile Anwendungen bei der Bundesagentur für Arbeit

Dr. Martin Merck
ORACLE Deutschland B.V. Co. KG
München

Daniel Wieser
IT-Systemhaus der Bundesagentur für Arbeit
Nürnberg

Schlüsselworte

Mobile, REST, OAuth 2.0, OAG, FMW, SOA Suite, SOA.

Einleitung

Mobile Anwendungen (Apps) sind in aller Munde und sind der aktuelle Trend im digitalen Kundenkontakt. Auch die Bundesagentur für Arbeit will künftig verstärkt Ihre Dienstleistungen auch auf mobilen Endgeräten zur Verfügung stellen und so als moderner Dienstleister auf das Kundenverhalten, besonders der jüngeren Generation, reagieren.

Im Umfeld der App-Entwicklung hat sich eine Architektur etabliert welche über REST-Services mittels JSON aus dem Internet auf Daten und Dienste in den Backendsystemen zugreifen. Diese Architektur stellt Unternehmen und Behörden jedoch vor eine Vielzahl neuer Probleme. Zum Beispiel wird bei der BA eine konsequente SOA Strategie bei der Implementierung von Geschäftsprozessen verfolgt und diese Dienste im Intranet deployed. Ein direkter Zugriff auf diese Dienste aus dem Internet ist bisher nicht möglich. Ein exponieren der Dienste im Internet ist jedoch mit hohen Sicherheitsrisiken verbunden und nicht erwünscht. Im Folgenden werden zunächst die Sicherheitsrisiken an verschiedenen Nutzungsszenarien diskutiert. Aus diesen Risiken werden Anforderungen an die Absicherung der exponierten Dienste hergeleitet. Anhand dieser Anforderungen wird eine Referenzarchitektur entwickelt. Diese basiert auf dem offenen OAuth Protokoll zur Kommunikation zwischen der App und den Backend-Diensten. Ferner wird eine Fassade im Extranet eingeführt welche den Zugriff auf die gesicherten SOA-Services regelt.

Benutzergruppen und Schutzbedarf

Zur Untersuchung der Sicherheitsanforderungen mobiler Anwendungen ist es wichtig, die Einsatzgebiete und Nutzerkreise zu unterscheiden. Prinzipiell müssen zwei Typen von Nutzerkreisen betrachtet werden:

- **Mitarbeiter**: Mitarbeiter eines Unternehmens, welche mit mobilen Endgeräten auf Informationen und Anwendungen des Unternehmens zugreifen. Dabei entsteht dem Unternehmen jedoch eine Reihe von Risiken, die hauptsächlich mit der Offenlegung von vertraulichen Informationen und Daten zu tun haben. Da die mobilen Endgeräte nicht innerhalb des abgesicherten Netzwerks des Unternehmens betrieben werden, können viele der typischen Schutzmechanismen eines Unternehmens in diesem Fall nicht greifen. Die wichtigsten Szenarien sind:

- **Datenversickerung „data leakage“**: Daten auf dem Endgerät können vom Endbenutzer über eingebaute Mechanismen des Geräts weitergeleitet werden z.B. durch Drucken, E-Mail Weiterleitung oder Speicherung in Clouddiensten. Dabei können sowohl durch eine unsichere Übertragung als auch durch nicht vertrauenswürdige Partner die Daten zweckentfremdet werden. Sowohl unbewusste als auch wissentlich verursachte Datenlecks des Mitarbeiters können nicht durch Sicherungsmaßnahmen innerhalb der Unternehmens-IT abgefangen werden.
- **Datendiebstahl**: Vertrauliche oder sensitive Daten des Unternehmens welche nicht genügend geschützt auf dem Gerät gespeichert werden, können auf vielfältige Weise von dem Gerät gestohlen werden. Dies kann durch eingeschleuste Vieren oder sonstige Schadprogramme oder durch „jailbreaking“ und direkten Zugriff auf die Dateiablage des Geräts erfolgen. Besonders nach einem Diebstahl oder Verlust des Geräts sind Daten wie Passwörter oder Zugangsdaten, personenbezogene Informationen oder geistiges Eigentum des Unternehmens gefährdet.
- **Phishing**: Zusätzlich zu den traditionellen Web-Phishing Angriffen können durch entsprechende, nicht vertrauenswürdige Apps oder SMS-Phishing Zugangsdaten zu den Unternehmenszugängen entwendet werden.

Typischerweise wird diesen Bedrohungen durch den Einsatz firmeneigener Geräte und durch ein stark kontrolliertes Gerätemanagement MDM (Mobile Device Management) begegnet. Modernere Techniken wie MAM (Mobile Application Management) erlauben die Absicherung der Unternehmensanwendungen und Daten in speziellen, sicheren Containern, und so eine Koexistenz der Nutzung des Geräts für Firmenanwendungen mit persönlichen und privaten Produktivitätswerkzeugen.
Da aktuell bei der Bundesagentur für Arbeit, dieser Anwendungsfall nicht im Fokus steht, wird nicht weiter auf diese Szenarien und die zugehörigen Sicherungsmaßnahmen eingegangen.

- **Kunden**: Endbenutzer, welche mit Hilfe ihrer mobilen Endgeräte Dienstleistungen des Unternehmens nutzen. Die Informationen, die im Normalfall hier abgerufen werden sind für den Kunden bestimmt. Sie sind in diesem Sinn, für diesen Kunden öffentlich. Das Unternehmen muss trotzdem den Kunden unterstützen und ihn bei Bedarf auf mögliche Risiken hinweisen. Für das Unternehmen selbst besteht das Hauptaugenmerk auf der Absicherung der exponierten Dienste. Dabei sind folgende Angriffsszenarien möglich:
  - **DoS Attacken**: Die exponierten Dienste müssen gegen unberechtigten und übermäßigen Gebrauch geschützt werden. Ferner soll unberechtigter Gebrauch erkannt und soweit möglich geblockt werden.
  - **Datenweitergabe**: Besonders sensitiv ist die Datenweitergabe von persönlichen Daten des Endbenutzers an Dritte. Zugriffe auf Dienste und APIs, welche persönliche Daten des Benutzers exponieren, dürfen nur nach strenger Authentisierung des Endbenutzers Daten liefern. Ferner sollten so wenig personenbezogene Daten wie möglich auf dem Endgerät gespeichert werden. Zur Unterstützung des Endbenutzers ist es vorteilhaft, sein Gerät für diesen Fall zu registrieren und ihm die Möglichkeit zu bieten, das Gerät bei Außerdienstsetzung, Verlust oder Diebstahl zu de-autorisieren.
  - **Datenveränderung**: Da alle Daten über öffentliche Netze übertragen werden, sollten diese vor Weitergabe und Veränderung geschützt werden. Dazu ist auf jeden Fall eine Transportverschlüsselung zu verwendet. Daten, die auf dem Endgerät abgelegt werden, sollten nach Möglichkeit verschlüsselt werden.
    Ferner muss bei allen APIs, welche zu Veränderung von Daten in den Backendsystemen führen, sichergestellt werden, dass diese nur mit einer

entsprechenden Autorisierung verwendet werden. Diese Zugriffe sollten auch gegen Attacken wie SQL-Injektion abgesichert werden.

Problematik des Zugriffs von Mobilen Applikationen auf die Geschäftslogik

Zur Illustration der Problematik beim Einsatz von mobilen Applikationen wird in Abb. 1**Error! Reference source not found.** der Zugriff auf eine Geschäftsanwendung mittels eines Browsers mit dem entsprechenden Zugriff aus einer mobilen Applikation verglichen.

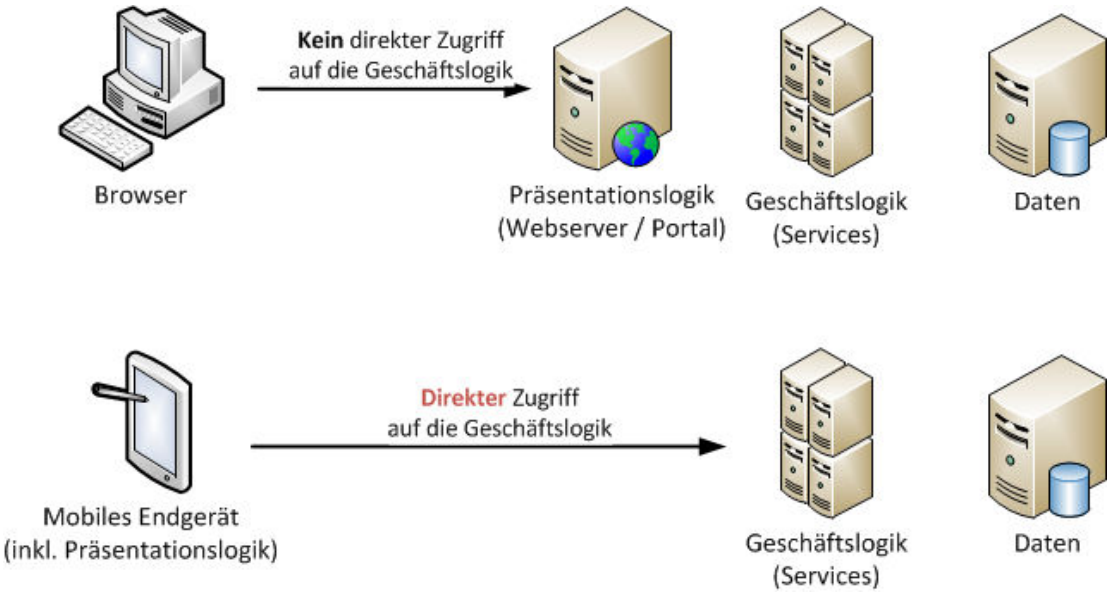

Abb. 1: Unterschiede beim Zugriff auf Unternehemensdienste mit mobilen Endgeräten.

Bei einer klassischen Webanwendung (3-Schicht Architektur) wird die Präsentationsschicht von der Anwendungslogik und der Datenschicht getrennt. Dem Endnutzer (Browser) werden nur die schon vordefinierten Darstellungen (Seiten, Formulare, etc.) geliefert und er erhält keinen direkten Zugriff auf die Geschäftslogik.

Bei mobilen Applikationen wird in Gegensatz dazu die Präsentationslogik auf dem Endgerät realisiert. Dadurch erfolgt der Zugriff auf die Geschäftslogik direkt vom Endgerät. Dies hat einige wichtige Konsequenzen für die Architektur der Anwendungen. Die Architektur entspricht dem Veröffentlichen von Web-Services mit ähnlichen Sicherheitsanforderungen. Jedoch haben sich im Bereich mobiler Anwendungen neue Standards für den Zugriff auf diese Web-Dienste herauskristallisiert. Im Gegensatz zu den HTTP/SOAP und XML Technologien traditioneller Web-Services werden REST und JSON verwenden. Dies erfolgt hauptsächlich zur Verringerung des Ressourcenbedarfs (CPU Ressourcen, Netzwerk Bandbreite) auf den mobilen Endgeräten. Ferner vereinfacht die Verwendung von JSON und Standard HTTP Mechanismen die Entwicklung der mobilen Anwendungen. Diese Technologiewahl hat jedoch zur Folge, dass Sicherheitsstandards für SOAP Web-Services wie WS-Security, SAML und WS-Trust nicht verwendet werden können. Ferner wurden Mechanismen wie WS-Trust zur Absicherung von Web-Service Kommunikation zwischen einer begrenzten Anzahl von vertrauenswürdigen Partnern konzipiert. Im Bereich mobiler Applikationen kann die Anzahl der

Partner jedoch nicht kontrolliert werden, und es besteht in vielen Fällen keine Vertrauensbeziehung zu diesen Partnern..

Anforderungen der BA an REST-Schnittstellen

Bei der BA wurden für REST-Dienste folgende Anforderungen identifiziert:

- REST APIs sollen soweit abgesichert werden, das ein „throtteling“ bzw. Abschalten von einzelnen Client-Typen (API Konsumenten) möglich ist um die im Intranet liegenden Dienste vor Angriffen auf die REST API und unerwartet hohem Verkehrsaufkommen zu isolieren.
- Gegebenenfalls möchte die BA die Nutzung von APIs nur nach Abschluss einer Nutzungsvereinbarung mit einem Partner erlauben. Es sollten Mechanismen vorgesehen werden, der einen entsprechenden Prozess unterstütz.
- Zugriffe auf persönliche Informationen sollen über die in der DMZ-Web bereitgestellte Sicherheitarchitektur authentisiert werden.

Referenzarchitektur für mobile Applikationen bei der BA

Die Referenzarchitektur ist in Abb. 2 dargestellt. Die normale Verfahrenslogik der BA ist im Intranet deployed und steht als SOA Services zur Verfügung. Ein Enterprise Service Bus (ESB) ist in dieser Darstellung aus Gründen der Übersichtlichkeit nicht mit dargestellt. Alle Zugriffe auf Fachdaten sollen über diese Services abstrahiert werden, sodass nur diese einen Zugriff auf die fachlichen Datenbanken in der Datenzone haben.

In der Informations- und Dienstezone wird ein API-Gateway als sicherer Eingangspunkt für Serviceaufrufe aufgebaut. Sie soll als Sicherheitskomponente dienen und mögliche Angriffe wie DoS oder SQL Injection frühzeitig abfangen.

In dieser Referenzarchitektur wird nun als zusätzliche Komponente eine „Mobile App Fassade“ in der Abb. 1:

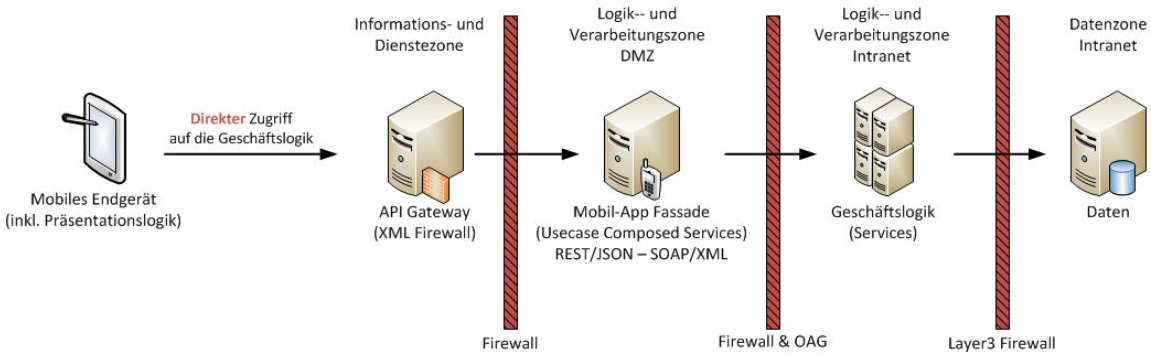

Abb. 2: Referenzarchitektur zur Bereitstellung mobiler Applikationen.

Logik- und Verarbeitungszone der DMZ eingeführt, die als Bindeglied zwischen der DMZ und dem Intranet dient. Die Mobile App Fassade erfüllt in dieser Architektur 3 wesentliche Funktionen:

1. Sie dient als Protokollwandler zwischen den REST/JSON Aufrufen der Mobilen Applikation und den SOAP Web-Services der Verfahren im Intranet.

2. Typischerweise wird die Geschäftslogik der Fachverfahren der BA als SOA Dienste im Intranet bereitgestellt. Durch den Einsatz der Mobile-App-Fassade kann der Lebenszyklus der Web-Services der Verfahren von dem Lebenszyklus der mobilen Applikation entkoppelt. Da nur der Lebenszyklus der Fassade und der Geschäftslogik direkt beeinflusst werden kann, ist dies besonders wichtig um eine flexible Entwicklung der Verfahren und mobiler Applikationen zu ermöglichen. Änderungen an den Verfahren können durch die Fassade transparent für die mobile Applikation gehalten werden.
3. Die Fassade dient als Integrationsplattform für die mobilen Applikationen. Es wird angestrebt, eine 1:1 Beziehung zwischen einer Fassade und einer mobilen Applikation (bzw. einer Hauptfunktionalität einer mobilen Applikation) zu etablieren. Die Fassade übernimmt hierbei die Kommunikation und Orchestrierung aller benötigten Fachdienste um die Funktionalitäten der App zu realisieren. Ein wichtiger Teil der Fassade ist hierbei eine Beschränkung der exponierten Schnittstellen der Web-Dienste auf die rein durch die App benötigten Funktionalitäten. Dadurch kann die exponierte API für die mobile Applikation auf das Minimum beschränkt werden und mögliche Angriffsvektoren reduziert werden.

Ein wichtiger Aspekt der Fassade ist auch der unterschiedliche Benutzerkontext in dem sie ausgeführt wird. Durch die Verortung der mobilen App Fassade in der DMZ kann diese im Kontext des Internet-Nutzers ausgeführt werden. Dadurch kann die Benutzerauthentisierung für mobile Anwendungen mit dem WebSSO der BA verknüpft werden. Beim Übergang von der DMZ ins Intranet übernimmt eine zusätzliche Oracle API Gateway Instanz den Identitätswechsel (Identity switching) zu einem technischen Benutzer, welcher zur Ausführung der fachlichen Web-Dienste berechtigt.

Zur Realisierung der App-Fassade bietet sich die Oracle SOA Suite 12c an. Mit ihr können SOAP-Services einfach als REST-Services exponiert werden. Ferner können mithilfe von BPEL die Backend-Dienste so orchestriert werden, dass eine optimale Schnittstelle für die mobile App bereitgestellt werden kann.

Da REST Services auf Protokollebene reine HTTP Zugriffe darstellen ist es wichtig, bei der Absicherung von REST APIs darauf zu achten, dass die Fassaden auf dedizierten Servern bereitgestellt werden, welche nur von der API Gateway erreichbar sind. Besonders im Falle von Anwendungen, welche als klassische Webanwendungen bereitgestellt werden und für welche auch eine REST-API zur Verfügung gestellt wird um eine native mobile App zu unterstützen, ist darauf zu achten, dass diese auf verschiedenen Servern bereitgestellt werden. Anderenfalls ist die REST-API auch über einen normalen HTTP-Aufruf möglich, welcher nicht über das API-Gateway sondern über die normalen Reverse Proxies und Webgates des Internetzugangs geroutet wird.

Verhalten von mobilen Applikationen (Apps)

Für die nachfolgenden Sicherheitsbetrachtungen ist es wichtig, das Verhalten von typischen Apps auf mobilen Endgeräten zu verstehen. Bei der Nutzung klassischer Webanwendungen wird der Zugriff von einem PC über einen „User-Agent“ (Browser) gestartet. Falls nötig wird eine Sitzung gestartet, welche für die Zeit der Nutzung einen Kontext bietet und auch SSO und ähnliche Funktionalitäten ermöglicht. Nach der Nutzung der Dienste schließt der Benutzer seinen „Browser“ und beendet so alle noch aktive Sitzungen.

Für den Benutzer eines mobilen Endgerätes verhalten sich Apps in ähnlicher Weise. Sie werden durch den Benutzer gestartet und nach Nutzung des Dienstes scheinbar beendet, indem eine neue App gestartet wird oder auf das mobile Betriebssystem gewechselt wird. Dies ist jedoch nicht der Fall, da die App nach dem Wechsel zu einer anderen App oder auf einen anderen Bildschirm im Hintergrund geöffnet bleibt. Sie kann auch weiterhin mit externen Diensten kommunizieren (z.B. zur Verfolgung

von Änderungen an Aktienkurse) oder Dienste des Gerätes benutzen (wie z.B. die Geodienste wie GPS um den aktuellen Ort zu bestimmen und relevante Informationen zu besorgen). Dadurch kann eine Sitzung der mobilen App über Monate bestehen bleiben, solange das Gerät kontinuierlich mit Energie versorgt wird. Erst durch vollständiges Herunterfahren des Geräts werden alle Apps beendet. Die App kann dem Benutzer auch Informationen in Form von Push-Mitteilungen anzeigen, ohne dazu im Vordergrund angezeigt werden zu müssen..

Dieses Verhalten hat vielfältige Auswirkungen auf die Backendsysteme, welche mit den Apps die Dienste für den Benutzer bereitstellen. Zum einen können Apps eine deutlich höhere Last erzeugen, da sie regelmäßig Aktionen ausführen können, ohne dass diese vom Benutzer ausgelöst werden. Andererseits verliert das Sitzungskonzept klassischer Webanwendungen seine Bedeutung. Zwar wäre es möglich nach einer Anmeldung durch die App eine Sitzung zu starten, diese könnte durch die App jedoch automatisch beliebig erneuert werden (auch mit zwischengespeicherten Passwörtern des Nutzers). Es ist in diesem Zusammenhang vorteilhaft Mechanismen zu entwickeln, welche eine bessere und gezielte Kontrolle der Zugriffe auf die Ressourcen in den Backend-Systemen erlauben

Absicherung des Zugriffs auf APIs mittels Token

Als neues Konzept zur Absicherung von APIs, besonders im Umfeld mobiler Applikationen, wird das Token eingeführt. Es symbolisiert eine Wertmarke, mit der einmalig ein Dienst (z.B. eine Waschmaschine in einem Waschsalon) benutzt werden kann. Soll der Dienst ein zweites Mal benutzt werden, muss eine neue Wertmarke erworben werden. Durch diesen Prozess kann die Nutzung des Dienstes an einer zentralen Stelle (der Wertmarken-Ausgabestelle) kontrolliert werden. Ferner können unterschiedliche Wertmarken für verschiedene Dienste verwendet werden. Wertmarken können direkt an den Nutzer des Dienstes vergeben werden oder von diesem über Dritte bezogen werden, welche so die Nutzung eines Dienstes kontrollieren können.

Dieses Konzept wurde in die digitale Welt übertragen um ein Verwaltungsinstrument zur Absicherung von Zugriffen auf Dienste zu ermöglichen. Die Token können einfache zufällige Zeichenketten sein, welche eine einmalige Nutzung eines Dienstes erlauben, oder kryptographisch abgesicherte Nachrichten, welche zusätzliche Meta-Informationen zum erlaubten Umfang der zu nutzenden Service beinhalten.

Eine App bezieht also über eine kontrollierte Stelle Token, welche sie berechtigt einen Zugriff auf einen Dienst durchzuführen. Zum Erwerb des Token muss sich die App authentisieren und unter Umständen eine Berechtigung vom Benutzer, für den Zugriff auf den Dienst, einholen. Mit dem Token kann sie nun den Dienst einmalig nutzen. Damit Apps Token nicht sammeln können, haben diese nur eine beschränkte Gültigkeitsdauer (z.B. 5-30 Minuten). Falls mehrere Dienste verwendet werden sollen, bzw. falls Dienste mehrmals aufgerufen werden, müssen neue Token erworben werden. Die Vergabe der Token erlaubt nun eine Steuerung der Nutzung von Diensten an zentraler Stelle.

Einsatz von OAuth 2.0 mit mobilen Endgeräten

OAuth 2.0 ist ein Autorisierungsprotokoll welches eine webseitenübergreifende Autorisierunge erlaubt. Dazu wird der Zugriff auf „Ressourcen“ über Token gesteuert, die von einem „Authorization Server“ ausgestellt werden. Neben dem klassischen „dreibeinigem“ OAuth, welches für den Zugriff auf Web-Ressourcen durch andere Web-Dienste verwendet wird, werden in der Spezifikation auch einfachere „Grants“ spezifiziert, welche zur Absicherung von REST-Schnittstellen verwendet werden können.

Eine wichtige Eigenschaft des OAuth Protokolls besteht darin, dass alle Interaktionen über ein einfache REST-API definiert wurden, welches auf reinen HTTP-Mechanismen basiert. Diese Eigenschaft von OAuth ermöglicht auch einen einfachen Einsatz von OAuth im mobilen Umfeld.

Alle Zugriffe auf die REST-Dienste im Backend-System werden mit einem „Access Token“ abgesichert. Um die nötigen Token zu erhalten muss sich der Client (App oder anderer REST Consumer) beim „Authentication Server“ identifizieren. Dies erfolgt über eine „Client ID“ und ein „Client Secret“. Da das Token nur einmal oder für eine kurze Zeit gültig ist, kann der „Authentication Server“ nun dazu benutzt werden, Zugriffsregeln (sogenannte Policies) zu definieren und durchzusetzen.

Das OAuth Verfahren kann zum Beispiel mit Hilfe des Oracle API Gateway (OAG) realisiert werden. Dabei dient der OAG sowohl als „Authentication Server“ als auch als Proxy für den „Resource Server“. Dies vereinfacht die Implementierung erheblich, da die REST-Services („Resources“) nicht besonders angepasst werden müssen. Für komplexere Sicherheitsszenarien kann auch der Oracle Access Manager Mobile & Social (OAMMS) verwendet werden. Mit ihm ist eine volle OAuth 2.0 Implementierung möglich, es müssen dann jedoch die REST Fassaden auch mit dem OAMMS Server kommunizieren.

In Abb. 3 ist eine entsprechende Policy für einen REST-Service einer App dargestellt. Der OAG überprüft in einem ersten Schritt das „Access Token“. In einem zweiten Schritt wird ein „Throtteling“ implementierte, welches die Anzahl der API-Aufrufe pro Zeiteinheit limitiert. Dies dient zum Schutz der Backend-Dienste und der Inhouse Applikationen die auf diese Zugreifen. Danach wird der REST-Request an die App-Fassade weitergeleitet. Bei der BA werden noch weitere Prüfungen im OAG realisiert um auch einen Schutz gegen fehlerhafte oder manipulierte Daten sicherzustellen.

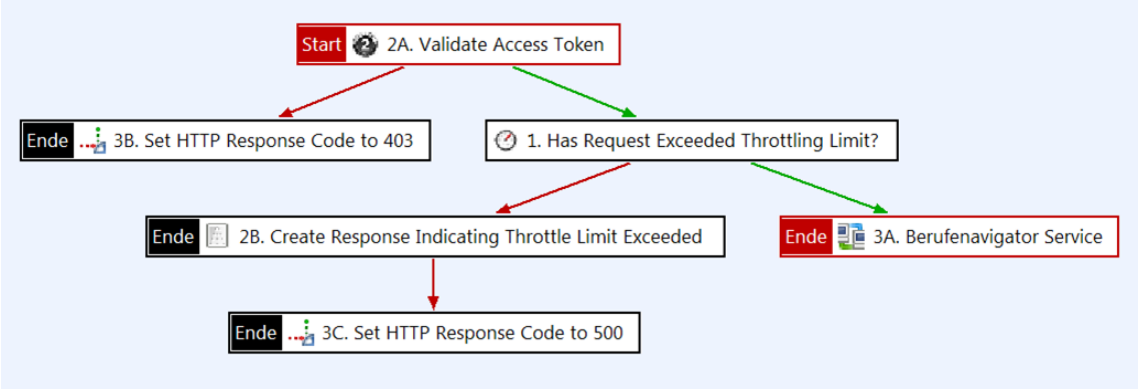

Abb. 3: OAG Policy zur Absicherung der Zugriffe auf REST-Dienste.

Zusammenfassung

In diesem Beitrag wurden die Herausforderungen bei der Bereitstellung von REST-Diensten für mobile Anwendungen besprochen. Mithilfe einer Architektur mit einer mobilen App-Fassade als Proxy zwischen App und Geschäftslogik, kann auch für diesen Anwendungsfall eine drei-Schicht Architektur realisiert werden. Zusammen mit dem OAuth 2.0 Protokoll können die REST-Schnittstellen der App-Fassade in einer REST-Welt abgesichert werden (Äquivalent zur Absicherung von SOAP-Schnittstellen mittels SAML-Assertions).

Kontaktadresse:
Dr. Martin Merck
ORACLE Deutschland B.V. Co. KG
Riesstr., 25
D-80992 München

Telefon: +49 (0) 89-1430 1271
Fax: +49 (0) 89-1430 2107
E-Mail martin.merck@oracle.com
Internet: www.oracle.com