# Sicherheitsaspekte und Best Practices in Geodateninfrastrukturen für den Einsatz in Smart Cities

**ANDREAS MATHEUS**[1]

*Zusammenfassung: Heutige Geodateninfrastrukturen basieren zumeist auf offenen Standards des Open Geospatial Consortiums, die eine Interoperabilität für Daten- und Dienste sicherstellen. Bedenkt man, dass diese Standards oft 10 Jahre oder älter sind und keine Sicherheitsaspekte berücksichtigen, so muss sich die Frage auftun ob diese Standards noch zeitgemäß sind, wenn es darum geht, geschützte oder schützenswerte Daten- und Dienste in Geodateninfrastrukturen mit modernen Web-Applikationen abzurufen.*
*In diesem Beitrag werden einige relevante Sicherheitsaspekte und Best Practices mit dem Ziel beschrieben, existierende Geodateninfrastrukturen „aufzurüsten" damit branchenübergreifende Nutzung abgesicherter Dienste möglich wird. Finden Sie am Beispiel Smart Cities heraus wie sich moderne grundlegende IT-Security und altgediente OGC Web Services komplementieren.*

## 1 Einleitung und Motivation

INSPIRE war der große Motor in Europa, der die Entwicklung von Geodateninfrastrukturen in den letzten 10 Jahren maßgeblich geprägt hat. Diese Entwicklung ist allerdings zum Stehen gekommen, denn der originäre INSPIRE Use Case war und ist der freie Austausch von offenen Geodaten spezieller Themen um eine gemeinsame Umweltpolitik in Europe zu ermöglichen. Und laut Gesetzgebung soll genau dies Ende 2018 möglich sein. Obwohl es durchaus wünschenswert wäre diese Infrastruktur auch für kommerzielle Nutzung zu verwenden, ist dies so ohne weiteres nicht möglich, denn Sicherheitsaspekte und deren interoperable Umsetzung fehlen seit der ersten Stunde. Überprüft man die aktuell gültigen Implementation Guidelines für z.B. den INSPIRE Download Service (V3.1 von 09.08.2013 oder INSPIRE View Service (V3.11 vom 04.04.2013) so stellt man fest, dass es keine Implementation Guidance für Security gibt[2]. Wenn man zusätzlich die normativen Referenzen untersucht, so muss man zu dem Schluss kommen, dass INSPIRE nur auf HTTP und nicht einmal auf HTTP**S** betrieben werden darf ohne eine Interoperabilitätsgarantie zu verlieren. Denn weder die Implementation Guidelines noch die OGC Web Service Standards erfordern HTTP über TLS (HTTPS). Mehr Informationen zu diesem Thema sind in OGC Testbed Engineering Reports verfügbar (OGC #15-022, OGC #16-048r1 und OGC #17-021r2).
Sind nun Geodateninfrastrukturen wie INSPIRE unbrauchbar und müssen ersetzt werden, wenn es darum geht das Konzept von Smart Cities umzusetzen? Das Konzept von Smart Cities ist eine Evolution in die Vernetzung verschiedenartiger Daten für Städte, um effizienter, produktiver und gleichzeitig attraktiver und grüner zu werden. Die intelligente Vernetzung von Geodaten mit

[1] Secure Dimensions GmbH, Waxensteinstr. 28, D-81377 München, E-Mail: am@secure-dimensions.de
[2] Der View Service hat beispielsweise 92 Implementation Requirements, jedoch kein einziges adressiert Security!

Daten anderer Branchen ist dabei eine der wesentlichen Herausforderungen. So ist eins der Technologieziele die Vernetzung von möglichst vielen Sensoren und anderen Messplattformen, um relevante Umweltdaten zu gewinnen. Damit man nicht bei Null anfangen muss, sollte es ebenfalls möglich sein, existierende Geodateninfrastrukturen so zu erweitern, dass über diese Geodaten, Stadtmodelle, Sensordaten und Daten von kommerziellen Anbietern wie z.B. Energiekonsum und Umweltdaten abgerufen werden können. Dies erfordert allerdings die Integration einer interoperablen und modularen sowie flexiblen Sicherheitslösung.
In diesem Beitrag soll für den speziellen Use Case „Smart Cities" aus dem Projekt Smart Sustainable Districts[3] exemplarisch gezeigt werden, wie eine beispielhafte Geodateninfrastruktur mit OGC Web Services elegant „aufgerüstet" werden kann um die Vernetzung von zugriffsgeschützten Sensorinformationen zu ermöglichen, ohne die eigentliche Basis stark verändern zu müssen.

## 2 Sicherheitsaspekte und Standards

Das Schlagwort „Sicherheit" im Zusammenhang mit Geodateninfrastrukturen wird oft missverstanden, wenn eine genaue Definition fehlt. Eine umfassende Definition erforderlicher Sicherheitsaspekte, die für Geodateninfrastrukturen relevant sind, findet man in ISO 7498-2 und ISO Multipart Standard 10181. In ISO 10181 werden diese Aspekte dann mittels verschiedener Frameworks standardisiert um eine Umsetzung in verteilten, offenen Architekturen zu ermöglichen: „*The series of Recommendations | International Standards on Security Frameworks for Open Systems addresses the application of security services in an Open Systems environment, where the term "Open Systems" is taken to include areas such as Database, Distributed Applications, Open Distributed Processing and OSI. The Security Frameworks are concerned with defining the means of providing protection for systems and objects within systems, and with the interactions between systems. The Security Frameworks are not concerned with the methodology for constructing systems or mechanisms*." [ISO 10181-1, p.7]
In ISO 10181 sind folgende Frameworks definiert: *Authentication [ISO 10181-2]*, *Access Control [ISO 10181-3]*, *Non-Repudiation [ISO 10181-4]*, *Confidentiality [ISO 10181-5]*, *Integrity [ISO 10181-6]* und *Security Audit and Alarms [ISO 10181-7]*. ISO 10181-1 bietet einen Überblick.
Ohne die verschiedenen Frameworks im Detail vorzustellen kann man festhalten, dass *Non-Repudiation* und *Security Audits and Alarms* In Geodateninfrastrukturen üblicherweise nicht relevant sind.
Relevant sind hingegen *Confidentiality* und *Integrity,* die eine wichtige Rolle bei der Umsetzung von Informationsflusskontrollen haben, aber auch vor unerlaubtem Mithören schützen (Confidentiality) sowie das unbemerkte Verändern von Information verhindern (Integrity). Bei der Umsetzung muss man grundsätzlich unterscheiden, ob ein Schutz nur während der Übertragung garantiert werden soll oder ob ein persistenter Schutz, unabhängig vom Übertragungsmedium etabliert werden soll. Zum Schutz bei der Übertragung kann eine Implementierung basierend auf

[3] https://www.gis.bgu.tum.de/projekte/smart-sustainable-districts-ssd/

dem OSI-Referenzmodell auf der Anwendungsschicht (Layer 7) oder Transportschicht (Layer 4) erfolgen. Im ersten Fall ist SOAP plus WS-Security eine standardisierte Möglichkeit, die allerdings zumeist nur für die Absicherung von Kommunikation zwischen Anwendungen im Back-Office eingesetzt wird. Client Anwendungen werden üblicherweise mittels eines verschlüsselten Kanals angebunden; also HTTP über TLS (HTTPS). Weil diese standardbasierte Lösung (IETF RFC 7246) im OSI Referenzmodell zwischen Layer 7 und 4 angesiedelt ist hat sie eine grundlegende Abhängigkeit: sie erlaubt nur die Verschlüsselung eines TCP Kanals zwischen Netzwerkendpunkten; nicht Anwendungen!

Die wichtigsten Frameworks, die eigentlich zur Absicherung einer Geodateninfrastruktur immer genannt werden, sind Authentication und Access Control. Zumeist existiert eine Abhängigkeit zwischen Access Control (also der Zugriffskontrolle) und der Authentication (also der Nachprüfbarkeit von behaupteten Identitäten), denn Zugriffsrechte werden oft mit identifizierten Benutzern verknüpft.

Für die Umsetzung der Authentication gibt es scheinbar endlos viele standardbasierte Möglichkeiten und proprietäre Verfahren, die immer für einen ganz bestimmten Zweck entwickelt wurden. Zwei wichtige Standards im Zusammenhang mit diesem Beitrag sind SAML2 und OpenID Connect.

Die Security Assertion Markup Language (SAML2) wurde 2005 von OASIS standardisiert und definiert verschiedene Protokolle sowie XML Strukturen zum Austausch von Zusicherungen. Hierbei können Zusicherungen die Identität eines Benutzers garantieren oder dessen Zugriffsrechte beschreiben. Es wird SOAP mit WS-Security sowie XML Digital Signature und XML Encryption eingesetzt um den Austausch von Zusicherungen bzgl. Vertraulichkeit und Integrität abzusichern. Zusätzlich werden Schlüsselpaare verwendet um ebenfalls die Authentizität von Zusicherungen überprüfen zu können. SAML2 wird heutzutage beispielsweise in allen produktiven akademischen Föderationen verwendet. Eine Auflistung der weltweit produktiven Föderationen (aktuell 69) findet man unter https://refeds.org/federations.

OpenID Connect ist eine Spezifikation, die nicht von einem bekannten Standardisierungsgremium verabschiedet wurde. Es wird als Community- oder quasi-Standard bezeichnet der eine Abfrage von benutzerbezogenen Informationen als Erweiterung von OAuth2 ermöglicht. OAuth2 ist als RFC 6749 (und RFC 6750) vom IETF 2012 verbschiedet worden. Allerdings sind bei der Nutzung von OpenID Connect die Vorgaben von OAuth2 zu berücksichtigen: Im Wesentlichen geht es darum, dass ein OAuth2 Access Token nur zur Rechtedelegation verwendet wird, aber damit keine personenbezogen Daten abrufbar sind. Bei der Erweiterung von OpenID Connect kann zusätzlich ein sog. ID-Token übertragen werden, das dann der Applikation ermöglicht, bestimmte personalisierte Seiten aufzubauen. Ebenso ist es möglich, dass personenbezogene Informationen mit dem Access Token abgefragt werden können. Im Vergleich zu SAML2 sind die OpenID Connect Zusicherungen als JSON strukturiert, die zur Umsetzung von Integrity oder Confidentiality als JWT oder JWE kodiert werden können. JWT ist in RFC 7519 definiert und erlaubt die Anwendung einer digitalen Signatur auf die JSON Struktur. JWE ist in RFC 7516 definiert und erlaubt die Verschlüsselung von JSON und damit die Umsetzung von Confidentiality. Zur Umsetzung von Confidentiality und Integrity können beide Verfahren nacheinander angewendet werden. In einer Geodateninfrastruktur ist es wichtig, standardisierte Verfahren für die Authentifizierung zu verwenden, um eine grundlegende Interoperabilität zwischen den Ser-

viceanbietern auch auf der Ebene der Sicherheit zu gewährleisten. Dies ist wichtig, damit u.a. ein geeignetes Rechtemanagement durchgeführt werden kann.
Für die Umsetzung von Access Control (also der eigentlichen Zugriffskontrolle) muss nicht notwendigerweise ein Standard eingesetzt werden, wenn man eigenverantwortlich in der eigenen Domäne agiert, denn dieses Framework hat dann keinen Einfluss auf die Interoperabilität der Geodateninfrastruktur. Allerdings sollte berücksichtigt werden, dass eine Abstimmung von Zugriffsrechten über die einzelnen Anbieter hinweg dann zu Schwierigkeiten und Missverständnissen kommen kann, wenn jede Organisation eine eigene Beschreibung der Zugriffsrechte nutzt. Eine „general purpose“ Sprache wurde von OASIS entwickelt: die eXtensible Access Control Markup Language (XACML) ist heute in verteilten Main-Stream IT Lösungen im Einsatz. Damit die Sprache auch zur Definition und Durchsetzung von geographischen Zugriffsbeschränkungen verwendet werden kann wurde vom OGC GeoXACML entwickelt und 2008 standardisiert. Im Wesentlichen definiert GeoXACML den Datentyp Simple Geometry aus ISO 19107 und 19125 und unterstützt das Geometrie-Encoding WKT sowie GML 2 und 3. Durch Anwendung von GeoXACML können interoperable, raum-zeit-basierte Zugriffskontrollregeln definiert und durchgesetzt werden. Als Beispiel kann der Ortsbezug von Benutzern und Ressourcen (Features oder Karten) oder dessen Historie berücksichtigt werden.

# 3 OGC Web Services in modernen Geodateninfrastrukturen

Der Einsatz von OGC Web Services ermöglicht das Abrufen von Geodaten in verschiedenen Formaten über standardisierte Schnittstellen. Allerdings sind OGC Web Service Spezifikationen autark bzgl. Security was die Frage aufwirft, ob mit diesen Diensten eine moderne GDI überhaupt aufgebaut werden kann. Der Begriff „modern“ zielt darauf ab, dass nicht nur Dienste abgesichert sind, sondern deren Zugriff über moderne Web-Applikationen erfolgt; also nicht mehr vorwiegend durch Desktop Applikationen.

## 3.1 OGC Web Services und HTTPS

OGC Web Service Spezifikationen sind ca. 10 Jahre oder älter und mit dem Ziel entwickelt, offenen Zugang zu Geodaten interoperabel zu ermöglichen. Heutzutage wird aber durch Policies vorgeschrieben, dass HTTPS verwendet werden muss. Ebenso implementieren Web-Browser, dass zu einer eingegebenen Web-Adresse standardmäßig ein HTTPS Verbindungsaufbau erfolgt. Wie in OGC Testbed Engineering Reports veröffentlicht, ist eine Service-Instanz die auf HTTPS betrieben wird nicht mehr standardkonform, weil keine normativen Anforderungen für dieses Hosting in den OGC Standards existieren. Somit ist man als Service-Anbieter darauf angewiesen zu hoffen, dass Client Anwendungen auch HTTPS unterstützen. Dies ist kein Thema für Webbasierte Anwendungen, weil sie ja im Web-Browser ausgeführt werden. Allerdings wurde in OGC Testbeds immer wieder festgestellt, dass Desktop-Applikationen entweder über keinen HTTPS Support verfügen, oder aber z.B. die Zertifikatprüfung unsachgemäß implementiert wurde.
Damit das Hosting von OGC Web Services auf HTTPS wieder auf interoperablen Grundlagen verankert ist hat das OGC in der Standards-Arbeitsgrupp OWS Common – Security SWG bereits einen Draft Standard entwickelt, der vermutlich im Juni 2018 verabschiedet wird. Der OGC Web

Services Security Standard (OGC #07-007) definiert ein Standardvorgehen um Sicherheits-Meta-Information in die Service-Capabilities zu integrieren. Das vorgestellte Verfahren ist backwards-compatible und ermöglicht somit eine Nutzung von kompatiblen Diensten mit existierenden Client-Anwendungen. Optional kann eine Service-Instanz pro Netzwerkendpunkt ebenso den Support anderer Security Features beschreiben. Es können ebenso Sicherheitsanforderungen wie Authentication oder Access Control, sowie Integrity und Confidentiality für SOAP basierte Service Instanzen beschrieben werden. Ziel ist es hierbei der Client-Anwendung Metainformationen mitzuteilen, damit z.B. die Konfiguration einer SAML2 oder OpenID Connect Authentication automatisch erfolgen kann.

### 3.2 Authentifizierungs-Föderation

Um eine Geodateninfrastruktur mit geschützten Diensten nutzbar zu machen ist ein GDI übergreifendes Rechtemanagement erforderlich. Damit dies geschehen kann muss es möglich sein, dass ein Benutzer sich nur einmal Anmelden muss und danach seine Identität bei allen Diensteanbietern akzeptiert wird. Dies kann durch die Abtrennung von Nutzermanagement und Service-Provisioning und dem Einsatz von SAML2 geschehen. Es gibt dann die sog. Identity Provider Entitäten die Benutzerkonten verwalten, eine Loginfunktionalität bereitstellen und anfragenden Service Provider Entitäten Zusicherungen über die Identität bereitstellen. Ein wesentlicher Vorteil von SAML2 gegenüber OpenID Connect ist das Trustmanagement: Es ist zwischen Entitäten und nicht anwendungsabhängig. Ein weiterer Vorteil von SAML2 ist die Unterstützung von Single-Sign-On was eine wesentliche Benutzerfreundlichkeit sicherstellt, denn ein Benutzer kann nach einmaligem Login viele geschützte Dienste unterschiedlicher Anbieter in einer Anwendung nutzen, ohne sich bei jedem Anbieter erneut anmelden zu müssen.
Die Trennung von Service und Nutzerverwaltung in verschiedene Entitäten hat unter der neuen Datenschutzrichtlinie in Europa (GDPR) die im 25. Mai 2018 in Kraft treten wird einen weiteren wesentlichen Vorteil. Weil ein Service Provider dann keine personenbezogenen Daten mehr erheben und verwalten muss, entfallen bestimmte Anforderungen die die GDRP verlangt. Speziell zertifizierte Unternehmen könnten nun die Benutzerverwaltung übernehmen und die Funktionalitäten für die Ausweispflicht implementieren. Allerdings muss der Service Provider ein (vereinfachtes) Privacy Statement bereitstellen, in dem vor allem das Gebot der Datensparsamkeit für personenbezogene Daten garantiert wird. Dies macht erforderlich, dass für jeden Zugriff nur so viele personenbezogene Daten genutzt werden wie zur Erbringung des Auftrags erforderlich ist.

### 3.3 Web-basierte Anwendungen und CORS

Web-basierte Applikationen nutzen JavaScript zur Implementierung und werden im Web-Browser ausgeführt. Bibliotheken wie OpenLayers, AngularJS oder Leaflet ermöglichen die schnelle und funktionsstarke Umsetzung von Geo-Web-Apps. Allerdings unterliegen diese JavaScript-Anwendungen der sog. Same Origin Policy, die 2010 vom W3C veröffentlicht wurde und in jedem Web-Browser durchgesetzt wird. Diese Policy verhindert, dass JavaScript Code, der von einem Web-Server geladen wurde, ohne weiteres auf andere Web-Server zugreifen kann. Das W3C hat in 2014 die sog. Cross-Origin Resource Sharing (CORS) Recommendation herausgegeben, in der genau definiert wird wie dieser Zugriffsschutz funktioniert. Bestimmte Zugriffe auf andere Web-Server die durch JavaScript (z.B. AJAX) ausgeführt werden, werden vom

Web-Browser dahingehend untersucht, ob bestimmte HTTP Antwort Header vorhanden sind. Welche dies sind, hängt vom jeweiligen HTTP Request ab, der durch JavaScript ausgeführt wurde.
Diese Regelung des Zugriffs auf andere Web-Server als jene, von der die JavaScript Anwendung geladen wurde, hat für die Nutzung in einer Geodateninfrastruktur erhebliche Auswirkungen. Wird eine Web-Anwendung beispielsweise von GeoApps.de geladen und sollen dann OGC Web Services von Service-A.net und Service-B.de aufgerufen werden, so fordert der Web-Browser die Existenz von CORS HTTP Antwort Headern. Im einfachsten Fall macht die Web-Anwendung eine HTTP GET Anfrage auf z.B. Service-A.net. Für diese Anfrage setzt der Web-Browser das HTTP Header „Origin: GeoApps.de“. Bevor der Web-Browser die Antwort von Service-A.net an die Web-Applikation weitergibt wird geprüft ob der geforderte HTTP Header „Access-Control-Allow-Origin“ vorhanden ist. Ist der Wert entweder „*“ oder „GeoApps.de“ so wird die Antwort an die Web-Applikation weitergegeben. Es ist also erforderlich, dass jeder Service-Betreiber seinen Web-Server so konfiguriert, dass entsprechende CORS Header gesetzt werden. Dies ist relativ kompliziert, da CORS verschiedene Use Cases vorsieht.
Die Existenz von Same Origin Policy und CORS hat allerdings erhebliche Auswirkungen auf die Performance von Web-Applikationen, denn beispielsweise wird für alle Anfragen die nicht Standard HTTP Header beinhalten, ein sog. Pre-Flight-Request vom Web-Browser verschickt. Nicht Standard HTTP Header sind z.B. der Header Authorization der zur Übertragung von User Informationen verwendet wird. Beispielsweise in HTTP Basic Authentication wird hier der UserName:Password String als Base64 kodiert übertragen. Ebenso kann ein OAuth2 Access Token als HTTP Header Authorization Bearer verschickt werden. Durch diese Anfragen erhöht sich also die Netzwerkauslastung, denn einem HTTP GET oder POST Request wird nun ein OPTIONS Request vorangestellt.
Letztendlich muss jeder Service-Anbieter CORS ordnungsgemäß implementieren, was üblicherweise weit darüber hinaus geht ein „*“ zurückzuliefern. Neben dem obligatorischen Withelisting der vertrauenswürdigen Domains sind auch Überlegungen für HTTP POST, PUT und DELETE erforderlich. Jedenfalls muss die HTTP Methode OPTIONS ohne Zugriffsschutz bereitgestellt werden, damit die Pre-Flight-Request ordnungsgemäß beantwortet werden.

### 3.4 Session Management und CORS

Sollen Services abgesichert werden, so muss ein Session Management etabliert werden mit dem üblicherweise ein Security-Context verknüpft ist. Grundsätzlich gibt es zwei Modelle: Serverseitiges und Client-seitiges Session Management. Beim Server-seitigem Modell nutzt man üblicherweise HTTP Cookies, wie sie in RFC 6265 definiert sind, damit die Client-Anwendung in einer Anfrage auf eine Session verweisen kann. Dies ist eine elegante Methode für Web-Applikationen und Web-Browser basierten Zugriffen, denn die relevanten HTTP Cookies werden automatisch im Web-Browser verwaltet. Allerdings wird bei Web-Anwendungen die einen Cross-Site-Request machen das Cookie nur mitgeschickt, wenn der Web-Server das entsprechende HTTP CORS Header „Access-Control-Allow-Credentials“ auf „true“ gesetzt hat. Eine weitere negative Eigenschaft von Server-Side Sessions ist deren Laufzeit: Die Session kann jederzeit ungültig werden und somit kann der Request der Client-Anwendung ungültig sein. Es ist

somit nicht ratsam ein Server-seitiges Session Management zu nutzen, wenn Web-Anwendungen Zugriff haben sollen.
Für Web-Anwendungen sollte Client-seitiges Session Management mit zustandslosen Services genutzt werden, wie es z.B. von OAuth2 unterstützt wird. Es ist die Aufgabe der Web-Anwendung immer ein gültiges Access Token zu haben. Somit kann sichergestellt werden, dass keine komplizierten Callback-Funktionen erforderlich sind, um die Session zu erneuern. Allerdings löst die Übertragung von Access Tokens im HTTP Header (von RFC 6750 empfohlene Art und Weise) bei Requests, die nicht an denselben Web-Server gerichtet sind, von dem die Applikation geladen wurde, einen CORS Pre-Flight-Request aus. Wegen dem damit einhergehenden Performanceverlust muss abgewogen werden, ob die Übertragung des Access Token als Teil der URL tragbar ist. Die Übertragung vom Access Token in der URL ist nach RFC 6750 ebenso zulässig, wird jedoch nach RFC 2396 als „unwise“ eingestuft: „*It is clearly unwise to use a URL that contains a password which is intended to be secret. In particular, the use of a password within the 'userinfo' component of a URL is strongly disrecommended except in those rare cases where the 'password' parameter is intended to be public*.” [RFC 2396, p24]. Die größte Gefahr, dass die Access Token im Klartext ausgegeben werden, ergibt sich durch Access Logs auf dem Web-Server, aber auch durch Redirect Angriffe in denen die ursprüngliche URL - die das Access Token enthält - als HTTP Referer Header mitschickt wird.

### 3.5 SAML2 vs. Oauth2 Session Management und Cloud Deployment

Vergleicht man die Protokolle von SAML2 und OAuth2 mit denen eine Session aufgebaut wird, so stellt man fest dass SAML2 auf einem 2-Way Handshake basiert: Die Client-Anwendung bekommt beim ersten Zugriff ein temporäres Session Cookie ausgestellt, das dann beim zweiten Zugriff (nach erfolgreichem Login) durch ein echtes Session Cookie ausgetauscht wird. Beim Deployment hinter einem Load-Balancer, wie es typischerweise in Cloud-Umgebungen der Fall ist, muss nun darauf geachtet werden, dass die Session-Information bei allen Instanzen hinter dem Load-Balancer verfügbar ist. Dies ist nicht erforderlich, wenn OAuth2 Access Tokens verwendet werden, denn das Session Management obliegt ja dem Client.
Ob nun allerdings SAML2 Sessions oder OAuth2 Access Tokens verwendet werden hängt nicht nur vom Deployment sondern im Wesentlichen davon ab, wodurch auf die geschützte Resource zugegriffen werden soll: Hat man eine Web-Applikation, so können Access Tokens genutzt werden. Steht allerdings nur der Web Browser zur Verfügung, weil beispielsweise HTML href Links gefolgt wird, ist OAuth2 unbrauchbar weil keine Logik vorhanden ist, die das Access Token einschleust. Dies ist der SAML2 Use Case: Öffnen von geschützten Resourcen direkt durch den Web Browser.
Es kommt somit auf den Use Case an, ob SAML2 Sessions, OAuth2 Access Tokens oder etwa beide Mechanismen genutzt werden müssen um den Zugriff auf eine Web-Resource oder einen Web Service abzusichern.

## 4 Security Best Practice am Beispiel Smart Cities

An der Technischen Universität München wird am Lehrstuhl für Geoinformatik derzeit an dem Projekt Smart Sustainable Districts gearbeitet. Im Rahmen des Projekts wurde die Smart District

Data Infrastructure (SDDI) (MOSHREFZADEH et al. 2017) entwickelt, die zeigt wie eine Realisierung des Smart Cities Konzepts durch intelligente Vernetzung von CityGML-basierten Stadtmodellen mit Sensorinformationen möglich ist. Hierbei werden überwiegend Standards des OGC verwendet.

Im Rahmen dieses Beitrags soll der folgende Use Case betrachtet werden: Benutzer der SDDI können eine Web-Anwendung laden, mit der die Anzeige eines 3D-Stadtmodells ermöglicht wird. Hierbei wird das 3D.Stadtmodell von einem OGC Web Feature Service abgerufen und mittels CityGML kodiert. Jedes Gebäude ist nun mit einem weiteren OGC Web Feature Service verbunden, über den detaillierte Informationen zum jeweiligen Gebäude durch einen Mausklick abgerufen werden können. Um das Smart Cities Konzept zu realisieren enthalten nun die empfangenen detaillierten Gebäudedaten über das Dynamizer-Konzept (CHATURVEDI & KOLBE 2016) weitere Verlinkung mit zusätzlichen Sensordaten, die von anderen Anbietern mittels OGC Sensor Observation Services (SOS) bereitgestellt werden. In diesem Use Case sind dies Umweltdaten und Energiekonsum der Gebäude. Der Benutzer hat nun die Möglichkeit, durch Klick auf einen entsprechenden Link, diese vernetzten Zusatzinformationen abzurufen. Diese Informationen können entweder im Web-Browser oder durch zusätzliche Web-Applikationen dargestellt werden.

Vom 1. September bis 15. Dezember 2017 wurde in Zusammenarbeit mit Secure Dimensions GmbH ein Security-Konzept umgesetzt, das es ermöglicht den Abruf der detaillierten Gebäudeinformation sowie der verlinkten Sensor-Zusatzinformationen zu schützen. Es soll also jeder Benutzer in der Lage sein das 3D-Stadtmodell zu sehen, aber die verlinkte Sensor-Information kann nur durch autorisierte Benutzer abgerufen werden. Insbesondere sollte es möglich sein, dass sich Benutzer mittels TUM oder Google Login anmelden können und sich nur einmal anmelden müssen, um auf die Ressourcen unterschiedlicher Anbieter zugreifen zu können (Single-Sign-On). Die Umweltdaten können alle Benutzer nach erfolgreichem Login sehen; die Smart-Meter Informationen eines Gebäudes – also z.B. der Stromverbrauch – soll nur durch TUM Benutzer möglich sein. Aus technischer Sicht war eine der wesentlichen Herausforderung eine Absicherung der Verlinkung von Zusatzinformationen zu garantieren, obwohl die eigentlichen Links im CityGML Modell oder in der Web Feature Service Antwort nur **ohne** Sicherheitskontext verfügbar sind. Und dies benutzerfreundlich durch Single-Sign-On das anbieterübergreifend akzeptiert wird.

Um eine realitätsnahe Umgebung für die Umsetzung des Security-Demonstrators zu haben, wurden Anwendungen und Services verteilt: die 3D CityGML Web-Anwendung wird von einem Anbieter bereitgestellt (www.3dcitydb.org); der WFS mit den detaillierten Gebäudeinformationen wird von Secure Dimensions bereitgestellt (wfs.sddi.secure-dimensions.de); der SOS für die Umweltdaten sowie der SOS für die Smart-Meter-Daten werden von der TUM bereitgestellt (ssdsos1.gis.bgu.tum.de) und (ssdsos2.gis.bgu.tum.de).

Die folgende Abbildung zeigt die Komponenten des Security-Demonstrators im Überblick.

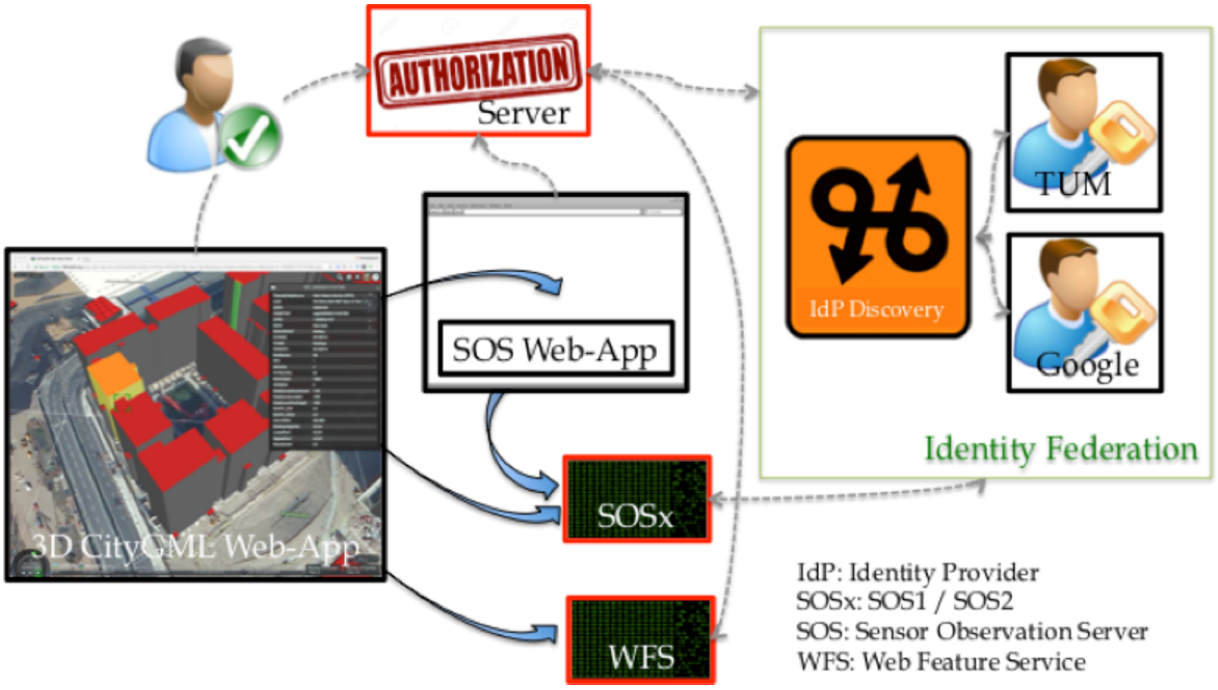


Abb. 1: Architektur Sketch des Security Demonstrator Projektes

Aus diesem Use Case ergeben sich zusammenfassend die folgenden Anforderungen, die bei der Integration einer Security-Lösung zu beachten sind:

1. Single-Sign-On: Wegen Benutzerfreundlichkeit muss ein einmaliges Login ausreichen
2. Föderation: Login via Google oder Technische Universität München
3. Unbeschränkter (anonymer) Zugriff auf die 3D CityGML Web-Applikation und das 3D Stadtmodell
4. Access Control: Benutzer mit Login Google können verlinkte Umwelt-Daten abrufen (im Web-Browser und in spezieller Web-Applikation)
5. Access Control: Benutzer mit Login TUM können verlinkte Umwelt-Daten und Smart-Meter-Daten abrufen (im Web-Browser und in spezieller Web-Applikation)
6. Web-Browser Same Origin Policy und W3C CORS

### 4.1 Umsetzung Föderation und Single-Sign-On

Die Umsetzung des Logins via Google oder TUM erfolgte durch eine SAML2 Föderation. In dieser Föderation sind die Identity Provider TUM und Google die beiden Entitäten, die das User Login bereitstellen. Beide unterstützen Single-Sign-On durch die Verwendung des SAML2 Session Initiators „PreviousSession“. Die Auswahl der Login-Entität wurde durch einen SAML2 kompatiblen Identity Provider Discovery Service ermöglicht. Dieser IdP DS oder auch WAYF (Where Are You From) bietet dem Benutzer neben der Auswahl des Login-Anbieters auch das persistente Speichern der Auswahl an. Dies erhöht die Benutzerfreundlichkeit, denn der Benutzer sieht die Login-Auswahl erst wieder, wenn die Vorauswahl aufgehoben wurde.

Dass Single-Sign-On via Google unterstützt wird, merkt der Benutzer dann, wenn er bereits bei Google angemeldet ist und die 3D-CityGML-Anwendung ebenfalls im Web-Browser ausgeführt wird. Klickt der Benutzer nun bei der 3D-CityGML-Anwendung auf Login und hat Google voreingestellt, so wird einfach dieses Login verwendet.

### 4.2 Umsetzung Zugriff auf Gebäude-Zusatzinformationen

Die Gebäude-Zusatzinformationen können per Klick auf das Gebäudeklötzchen von einem WFS abgerufen werden. Hierzu muss der Benutzer angemeldet sein. Die eigentliche WFS Abfrage wird von der 3D CityGML Applikation (einer Web-Applikation) beim Klick durchgeführt. Obwohl der Benutzer-Login durch SAML2 Identity Provider erfolgt, ist – wie zuvor erörtert - für diese Art des Zugriffs durch die Web-Applikation die Nutzung von SAML2 Sessions ungeeignet. Es ist somit erforderlich den Zugriff durch OAuth2 Access Tokens abzusichern.
Zur Nutzung von OAuth2 Access Tokens waren drei Dinge umzusetzen:

1) Zuerst wurde ein OAuth2 kompatibler Authorization Server installiert und die 3D- CityGML-Anwendung registriert. Das Login erfolgt über Google oder den TUM Identity Provider.
2) Der WFS mit den Gebäudezusatzinformationen wird als OAuth2 Resource Server betrieben.
3) Die 3D-City-GML-Anwendung wurde erweitert, um Access Tokens vom Authorization Server abzurufen und bei Anfragen an den WFS mitzuschicken.

### 4.3 Umsetzung Zugriff auf verlinkte Sensor-Daten

Der Zugriff auf die verlinkten Sensor-Daten zu den Gebäuden kann ebenfalls via Klick auf einen Link erfolgen. Hierbei werden die Ergebnisse der SOS Abfrage entweder direkt im Web-Browser angezeigt oder es wird eine geschützte Applikation geladen, die auf die SOS API zugreift. Hierbei ist die Verwendung von SAML2 Sessions bestens geeignet, denn damit kann der direkte Zugriff durch den Web-Browser erfolgen. Somit wurden bei SOS – für die Umwelt- sowie Smart-Meter Informationen – durch SAML2 Service Provider abgesichert.

### 4.4 Umsetzung von CORS

Damit die Auswirkung der Same Origin Policy des Web Browsers und dessen Lösung umgesetzt werden konnte, wird ein WFS-Proxy für die Gebäudezusatzinformation von Secure Dimensions betrieben. Somit ist jeder Zugriff von der 3D-CityGML-Anwendung auf den WFS eine Cross-Site Anfrage. Um die Funktionalität bzgl. CORS zu gewährleisten wurde der WFS entsprechend für Simple, wie auch Pre-Flight-Requests konfiguriert.

### 4.5 Demonstration

Das Ergebnis des Security Demonstrators ist online verfügbar. Als Einstieg muss die 3D-CityGML-Anwendung im Web-Browser geladen werden. Dazu kann die folgende URL genutzt werden: https://3dcitydb.org/qeop-web-map-security/3dwebclient/index.html

## 5 Fazit & Ausblick

Es ist möglich, eine moderne Geodateninfrastruktur mit interoperabler Sicherheit auf der Basis von existierenden OGC Web Service und Encoding Standards aufzubauen. Es wurde gezeigt wie durch die Kombination entsprechender Sicherheitsstandards – hauptsächlich SAML2 und O-Auth2 – eine modulare und interoperable Sicherheitslösung für Authentication in eine existierende Geodateninfrastruktur integriert werden kann. Die Realisierung des Security-Demonstrators basiert im Wesentlichen auf Open Source Mainstream IT Softwarekomponenten mit denen jederzeit ein Produktivsystem aufgebaut werden könnte. Die Implementierung der SAML2 Komponenten basiert auf der Internet2 Software Shibboleth[4]; die Implementierung des OAuth2 Authorization Servers basiert auf der Implementierung von Brent Shaffer[5];
Obwohl die Implementierung auf HTTPS ein Erfolg war muss berücksichtigt werden, dass der Betrieb von OGC Web Services auf HTTPS streng genommen nach den existierenden OGC Standards nicht vorgesehen ist. Aber hierzu wird das OGC in Kürze einen Standard verabschieden, der diese Lücke schließt.
Die Ergebnisse dieses Beitrages haben aber ebenso gezeigt, dass eine moderne GDI neben der Umsetzung von üblichen Sicherheitsaspekten wie Authentication und Access Control vor allem darauf angewiesen ist, dass ebenso Mainstream IT Unterstützung sichergestellt wird. Insbesondere sei hier auf die Anforderungen durch Web-Anwendungen und die sich daraus ergebende Unterstützung wie CORS hingewiesen. Eine einfache Konfiguration, die „*“ für die erlaubte Origin zurückliefert, sollte nie verwendet werden, es sei denn man versteht genau welche Auswirkungen dies hat.
Im Hinblick auf die kommende GDPR gibt sicherlich noch viel zu tun; auch für Geodateninfrastrukturen mit Benutzerverwaltung. Wichtig erscheint jedoch das beschriebene Konzept, die Verwaltung von Benutzern und das Service-Angebot zu trennen, damit Service Anbieter nicht zu sehr mit der GDRP belastet werden. Stattdessen sollten qualifizierte Unternehmen die Benutzerkonten verwalten. Ein föderatives Identity Management System kann auch in GDIs durch die Verwendung von SAML2 erfolgen, wie es in akademischen Föderationen bereits tagtäglich genutzt wird.
Um den Zugriff auf geschützte Dienste durch Web-Applikationen und Desktop-Applikationen zu ermöglichen, kann OAuth2 mit Access Tokens verwendet werden. Beispielsweise hat Secure Dimensions in Kooperation mit der TUM im Rahmen des OGC Testbed 13 SAML2 und OAuth2 Plugins[6] für QGIS 2.18.x als Open Source entwickelt und bietet eine fertig erstellte DLL für Windows 10 an, die von der Testbed 13 Projektseite heruntergeladen[7] werden kann.
Die Umsetzung des Security Demonstrators im Smart Sustainable District Projekt hatte die Integration von Authentication mit einfacher Zugriffskontrolle zum Ziel. Somit ergibt sich als Ausblick die Integration einer Zugriffskontrolle, die nicht nur rollenbasierte sondern auch orts-

[4] https://www.internet2.edu/products-services/trust-identity/shibboleth/
[5] https://github.com/bshaffer/oauth2-server-php
[6] https://github.com/securedimensions
[7] https://www.tb13.secure-dimensions.de/

abhängige Rechte durchsetzt. So könnte beispielsweise der Ort des Benutzers zusätzliche Zugriffsrechte auf Gebäude-Zusatzinformationen ermöglichen.
Zum Schluss stellt sich die Frage in wie weit die Best-Practices-Ergebnisse dieses Beitrags auch auf andere Geodateninfrastrukturen wie z.B. INSPIRE übertagen werden können. Diese Frage kann nicht pauschal beantwortet werden, aber es steht zu hoffen, dass dieser Beitrag eine entsprechende Diskussion in Gang setzt. Dies erscheint insbesondere wünschenswert, da das Ergebnis des JRC Projektes „Authentication, Authorization and Accounting (AAA) for Data and Services in EU Public Administrations“ [JRC, Action Item 1.17, ISA Programme] bereits eine Access Management Federation auf der Basis von SAML2 empfiehlt.

## 6 Literaturverzeichnis

BRÖRING, A., STASCH, C. & ECHTERHOFF, J., 2012: OGC Sensor Observation Service Interface Standard, OGC #12-006.

CANTOR, S., KEM, J., PHILPOTT, R. & MALER, E., 2005: Assertion and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0.

CHATURVEDI, K. & KOLBE, T.H., 2016: Integrating Dynamic Data and Sensors with Semantic 3D City Models in the context of Smart Cities. Proceedings of the 11th International 3D Geoinfo Conference (ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences), ISPRS.

GRÖGER, G., KOLBE, T.H., NAGEL, C. & HÄFELE, K.-H., 2012: OGC City Geography Markup Language (CityGML) Encoding Standard, OGC #12-019

HARDT, D., 2012: The OAuth 2.0 Authorization Framework, IETF RFC 6749.

INITIAL OPERATING CAPABILITY TASKFORCE, 2013: Technical Guidance for the implementation of INSPIRE Download Services, Version 3.1.

INITIAL OPERATING CAPABILITY TASKFORCE, 2013: Technical Guidance for the implementation of INSPIRE View Services, Version 3.11.

ISO 7498-2, 1989: Information processing systems – Open Systems Interconnectin – Basic Reference Model – Part 2: Security.

ISO 10181-1, 1996: Information Technology – Open Systems Interconnectin – Security frameworks for open systems: Overview.

ISO 10181-2, 1996: Information Technology – Open Systems Interconnectin – Security frameworks for open systems: Authentication Framework.

ISO 10181-3, 1996: Information Technology – Open Systems Interconnectin – Security frameworks for open systems: Access Control Framework.

ISO 10181-4, 1997: Information Technology – Open Systems Interconnectin – Security frameworks for open systems: Non-Repudiation Framework.

ISO 10181-5, 1996: Information Technology – Open Systems Interconnectin – Security frameworks for open systems: Confidentiality Framework.

ISO 10181-6, 1996: Information Technology – Open Systems Interconnectin – Security frameworks for open systems: Integrity Framework.

ISO 10181-7, 1996: Information Technology – Open Systems Interconnectin – Security frameworks for open systems: Security audits and alarms.

JONES, M. & HARDT, D., 2012: The OAuth 2.0 Authorization Framework: Bearer Token Usage, IETF RFC 6750.

JONES, M. & HILDEBRAND, J., 2015: JSON Web Encryption (JWE), IETF RFC 7516.

JONES, M., BRADLEY, J. & SAKIMURA, N., 2015: JSON Web Token (JWT), IETF RFC 7519.

MATHEUS, A., 2008: Geospatial eXtensible Access Control Markup Language (GeoXACML) Version 1.0, OGC #11-017.

MATHEUS, A., 2015: OGC Testbed-11 Implementing Common Security Across the OGC Suite of Service Standards, OGC #15-022

MATHEUS, A., 2016: OGC Testbed-12 OWS Common Security Extension ER, OGC #16-048r1

MATHEUS, A., 2017: OGC Testbed-13 Security ER, OGC #17-021r2

MATHEUS, A., 2018: OGC Web Services Security Standard, OGC #17-007 (verfügbar voraussichtlich Juni 2018)

MOSHREFZADEH, M., CHATURVEDI, K., HIJAZI, I., DONAUBAUER, A. & KOLBE, T.H., 2017: Integrating and Managing the Information for Smart Sustainable Districts - The Smart District Data Infrastructure (SDDI). In: Kolbe, T.H., Bill, R. & Donaubauer, A. (Hrsg.): Geoinformationssysteme 2017 – Beiträge zur 4. Münchner GI-Runde. Wichmann Verlag.

MOSES, T., 2005: eXtensible Access Control Markup Language (XACML) Version 2.0.

RISSANEN, R., 2013: eXtensible Access Control Markup Language (XACML) Version 3.0.

RESCORLA, E., 2000: HTTP Over TLS, IETF RFC 2818.

SAKIMURA N., BRADLEY, J., JONES, M., DE MEDEIROS, B. & MORTIMORE, C., 2014: OpenID Connect Core 1.0 incorporating errata set 1

VANDENBROUCKE, D., MATHEUS, A., FRIGNE, D., DE GRAEF, P., COPIER, R. & SMITH, R., 2014: Authentication, Authorization & Accounting for Data and Services in EU Public Administrations, D4.1.5 – Final technical report.

VRETANOS, P., 2010: OpenGIS Web Feature Service 2.0 Interface Standard, OGC #09-025r1.

W3C RECOMMENDATION, 2014: Cross-Origin Resource Sharing.