Klaus Stranacher, Bernd Zwattendorfer, Sandra Fruhmann, Patrick Koch - "Umsetzung eines vertrauenswürdigen Open Government Data" - D-A-CH Security 2014, pp. 236-247

Umsetzung eines vertrauenswürdigen Open Government Data

Klaus Stranacher[1], Bernd Zwattendorfer[1], Sandra Fruhmann[2], Patrick Koch[2]

[1] E-Government Innovationszentrum (EGIZ)
{Klaus.Stranacher, Bernd.Zwattendorfer}@egiz.gv.at

[2]Technische Universität Graz
{Sandra.Fruhmann, pkoch}@student.tugraz.at

Zusammenfassung

In den letzten Jahren hat sich Open Government Data (OGD) zur Veröffentlichung von Daten der öffentlichen Verwaltung etabliert. Das wurde auch auf europäischen Level erkannt und die PSI Direktive wurde novelliert. Aufgrund dieser gestiegenen Popularität ist es verwunderlich, dass Sicherheitsaspekte, wie Authentizität und Integrität, bisher wenig bis keine Beachtung fanden. Weder die OGD Prinzipien noch die PSI Direktive erwähnen detaillierte Sicherheitsaspekte, abseits des Schutzes privater oder persönlicher Daten. Im e-Government Forschungsbereich wurde hierzu ein Konzept für ein vertrauenswürdiges OGD, basierend auf elektronischen Signaturen, entwickelt. Dieses sehr rudimentäre Konzept erwähnt jedoch keine Integrationsstrategie in bestehende Infrastrukturen noch werden konkrete Signaturformate festgelegt. In dem vorliegenden Beitrag erweitern wir dieses Konzept und identifizieren anfangs Anforderungen für einen konkrete Umsetzung eines vertrauenswürdigen OGD. Auch werden OGD Formate hinsichtlich ihrer Signaturfähigkeit analysiert. Darauf aufbauend wurde eine modulare und adaptierbare Architektur entwickelt, die eine einfache Integration in bestehende Infrastrukturen erlaubt. Um unsere Architektur zu evaluieren wurde ein server-seitiges Web-Service entwickelt um vertrauenswürdiges OGD zu veröffentlichen. Schließlich wurde auch eine exemplarische client-seitige Smartphone Applikation umgesetzt, die die Verwendung zeigt und Möglichkeiten zur Verifikation des vertrauenswürdigen OGD zur Verfügung stellt.

1 Einleitung

Open Government Data (OGD) ist derzeit eines der meisten diskutierten Themen im e-Government Bereich. Auf europäischen Level wurde diese Wichtigkeit durch die Digitale Agenda für Europa[1] und die novellierte PSI Richtlinie [Euro13] manifestiert. Diese Erweiterung der PSI Direktive nähert sich dabei stark an die Prinzipien der Open Government Data Bewegung an. Nichtsdestotrotz erwähnen weder die OGD Prinzipien noch die aktualisierte PSI Richtlinie irgendwelche Sicherheitsaspekte, abseits des Datenschutzes. Insbesondere Fragen hinsichtlich der Authentizität und Integrität der Daten bleiben unbeantwortet. Aus diesem Grund schlagen die Autoren von [StKZ12] ein Konzept für ein

[1] http://ec.europa.eu/digital-agenda/

vertrauenswürdiges OGD vor, das auf der Nutzung von elektronischen Signaturen – zur Wahrung der Authentizität und Integrität der Daten – beruht. Dieses Konzept ist jedoch sehr rudimentär und beinhaltet keinerlei Umsetzung oder Integrationsstrategie in bestehende Infrastrukturen.

Aus diesem Grund präsentieren wir in dem vorliegenden Beitrag eine modulare und adaptierbare Architektur für ein vertrauenswürdiges OGD und dessen Implementierung. Der Rest dieses Beitrags ist daher wie folgt aufgebaut. Abschnitt 2 präsentiert kurz das vorgeschlagene Konzept für ein vertrauenswürdiges OGD. In Abschnitt 3 werden konkrete Anforderungen für eine Umsetzung des Konzepts identifiziert. Im folgenden Abschnitt 4 werden Signatur- und OGD-Formate analysiert. Insbesondere wird die Signaturfähigkeit von OGD-Formaten evaluiert. Abschnitt 5 enthält schließlich unsere entwickelte Architektur. Zur Evaluierung der Architektur wird in Abschnitt 6 die darauf aufbauende Implementierung vorgestellt. Diese Implementierung enthält eine server-seitige Komponente zur vertrauenswürdigen Veröffentlichung und einer client-seitigen Smartphone Applikation, die zeigt wie diese Daten konsumiert werden können. Abschließend fassen wir unsere Ergebnisse zusammen und ziehen ein Fazit.

2 Vertrauenswürdiges Open Government Data

2.1 Einleitung

Für die Bereitstellung von OGD besteht eine Reihe von Anforderungen, wie beispielsweise Vollständigkeit, leichter Zugang oder Primärquelle. Weitere Details zu diesen Anforderungen finden sich in den Open Data Prinzipien der Open Government Working Group [OGW07] bzw. in den Rahmenbedingungen für Open Government Data Plattformen der Projektgruppe Cooperation Open Government Data Österreich [COGD12].

In sämtlichen Anforderungen für OGD befinden sich jedoch keinerlei Informationen hinsichtlich der Sicherheit und Vertrauenswürdigkeit der bereitgestellten Daten. Auch die PSI-Richtlinie [Eur03, Euro13] befasst sich mit dieser Thematik nicht. Diese Problematik wurde von den Autoren von [StKZ12] aufgegriffen und sie haben hierzu ein Konzept für ein vertrauenswürdiges OGD entwickelt, dass im Folgenden vorgestellt wird.

2.2 Konzept

Das Grundprinzip für ein vertrauenswürdiges OGD beruht auf der Verwendung von elektronischen Signaturen. Abbildung 1 veranschaulicht dieses Grundprinzip. Dabei befinden sich die zu veröffentlichenden Daten in der Domäne der OGD Bereitstellerin bzw. des OGD Bereitstellers. Diesen Daten werden nun, vor der Veröffentlichung, von der Bereitstellerin bzw. dem Bereitsteller mit dem privaten Signaturschlüssel der Bereitstellerin bzw. des Bereitstellers signiert. Anschließend werden die signierten Daten an geeigneter Stelle publiziert und stehen als vertrauenswürdiges OGD zum Download zur Verfügung.

Die OGD Bezieherin bzw. der OGD Bezieher kann nun die Daten inkl. der Signatur downloaden. Die Signatur der Daten kann nun von der Bezieherin bzw. vom Bezieher überprüft werden. Dies kann mittels einer Signaturprüfung in der eigenen Domäne oder auch über ein externes Signaturprüfservice erfolgen. Nach einer erfolgreichen Signaturprüfung kann die Bezieherin bzw. der Bezieher auf die Vertrauenswürdigkeit der Daten vertrauen.

Durch die Verwendung einer elektronischen Signatur durch die Bereitstellerin bzw. den Bereitsteller bieten sich folgenden Vorteile:

- *Integrität der Daten*: Durch die Sicherstellung der Integrität der Daten ist gewährleistet, dass nachträgliche Änderungen der signierten Daten erkannt werden können. Davon profitieren sowohl Bezieherinnen und Bezieher als auch Bereitstellerinnen und Bereitsteller. Einerseits können sich Bezieherinnen und Bezieher darauf verlassen richtige Daten erhalten zu haben, andererseits können Bezieherinnen und Bezieher nicht behaupten falsche Daten erhalten zu haben (Vorteil für Bereitstellerinnen und Bereitsteller).
- *Authentizität der OGD Bereitstellerin bzw. OGD Bereitstellers*: Die Bezieherin bzw. der Bezieher kann die Identität der Bereitstellerin bzw. des Bereitstellers feststellen.

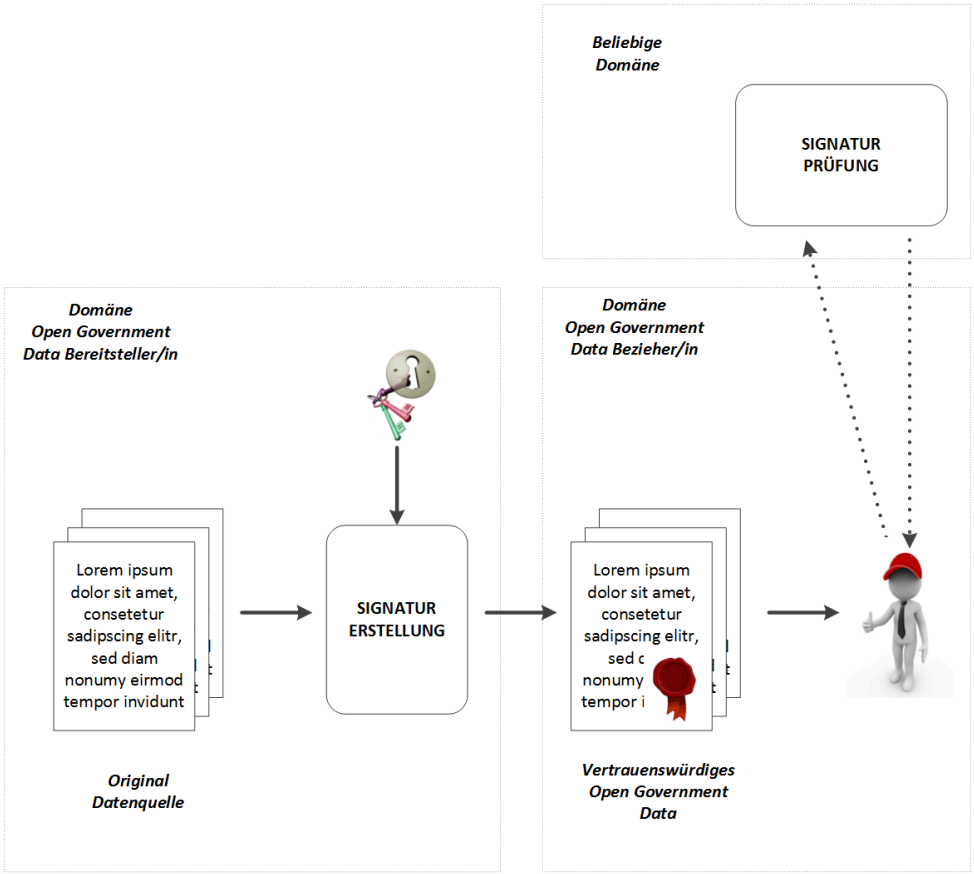

Abb. 1: Grundprinzip vertrauenswürdiges OGD [StKZ12]

3 Anforderungen

Die Autoren von [StKZ12] haben keine konkreten Anforderungen für eine reale Umsetzung eines vertrauenswürdigen OGD definiert. Dies möchten wir nun nachholen und haben daher folgenden Anforderungen identifiziert, die von der Architektur und Implementierung erfüllt werden müssen:

- *Modularität und Anpassungsfähigkeit*: Die IT-Infrastruktur ist ständigen Veränderungen unterworfen. Diese müssen sowohl von der Architektur als auch der Implementierung so weit wie möglich berücksichtigt werden. Aus diesem Grund

müssen Architektur und Implementierung einem modularen und anpassungsfähigen Ansatz verfolgen. Zusätzliche Module, im Speziellen Module zur Unterstützung weiterer Signatur- und Datenformate, sollen einfach eingebunden werden können.

- *Einfache Integration*: In den meisten Fällen existiert bei den zuständigen Stellen eine entsprechende Infrastruktur zur Veröffentlichung von OGD. Um weitere Kosten zu minimieren und auch bereits investierte Kosten bestmöglich auszunutzen müssen die Architektur und die Implementierung einfach in die bestehende Infrastruktur eingebunden werden können. Eine einfache Integration, kombiniert mit einer ebenfalls einfachen und intuitiven Konfiguration, ermöglicht ein schnelles Aufgreifen der Lösung seitens der Stellen, die OGD veröffentlichen.
- *Interoperabilität*: Interoperable Dienste sind einer der Hauptziele der EU Digitalen Agenda für Europa – speziell im grenzüberschreitenden Kontext. Für unseren Anwendungsfall betrifft dies vor allem die Signatur- und OGD Datenformate. Aus diesem Grund müssen die Architektur und die Implementierung den Vorgaben des Europäischen Interoperabilitäts-Framework (EIF)[2] folgen. Insbesondere müssen aktuelle Entscheidungen im Bereich Signatur- und OGD Datenformate berücksichtigt werden.
- *Automatische Verarbeitung*: Architektur und Umsetzungen müssen so designt und entwickelt werden, dass sie einen automatische Weiterverarbeitung der Daten ermöglichen. Jede manuelle Interaktion soll eliminiert werden.

4 Daten- und Signaturformate

4.1 Analyse Signaturformate

Die rechtliche und organisatorische Basis für elektronische Signaturen im E-Government Kontext ist durch die EU Signaturrichtlinie [Euro00] und ihre nationale Umsetzungen gegeben. Das Signaturgesetz legt dabei unterschiedliche Ausprägungen von Signaturen mit unterschiedlichen Rechtswirkungen fest. So ist eine fortgeschrittene elektronische Signatur, die mit einer sicheren Signaturerststellungseinheit erstellt wurde und auf einem qualifizierten Zertifikate beruht, der handschriftlichen Unterschrift gleichgestellt.

Durch die große Anzahl an unterschiedlichen Signaturformaten – sowohl standardisierte als auch proprietäre – entstanden Interoperabilitätsprobleme, speziell im grenzüberschreitenden Kontext. Um hier zu einer Harmonisierung beizutragen und folglich die Interoperabilität zu steigern, wurde von der Europäischen Kommission Referenzformate für fortgeschrittene elektronische Signaturen definiert. Diese Referenzformate sind in der EU Kommissionsentscheidung 2011/130/EU [Euro11] festgelegt.

Für die Analyse der Signaturformate sind daher auch nur diese Referenzformate von besonderem Interesse. Die Referenzformate sind:

1. CAdES-BES/EPES Signaturen
2. XAdES-BES/EPES Signaturen
3. PAdES-BES/EPES (Teil 3) Signaturen

[2] http://ec.europa.eu/isa/documents/eif_brochure_2011.pdf

Im Folgenden werden diese drei Formate detaillierter betrachtet und untersucht.

CAdES: CAdES steht für CMS Advanced Electronic Signature und ist ein ETSI-Standard, der auf CMS (Cryptographic Message Syntax) basiert und als ETSI TS 101 733 veröffentlicht ist. Ziel dieses Standards war es CMS dahingehende zu erweitern, um den Anforderungen für fortgeschrittene elektronischen Signaturen aus der EU Signaturrichtlinie gerecht zu werden. Diese Anforderungen werden von den CAdES Ausprägungen Basic Electronic Signature (BES) und Explicit Policy Electronic Signature (EPES) erfüllt. Das Basisformat CMS basiert dabei auf PKCS#7. CMS und somit CAdES erlauben sowohl das Verschlüsseln als auch das Signieren von Daten. Die CMS Spezifikation sieht vor, dass sämtliche Daten in so genannten ContentInfo-Containern abgelegt werden. Die Daten selbst werden dabei ASN.1[3] codiert. Mittels CAdES-Signaturen können prinzipiell beliebige Daten signiert werden, da die zu signierenden Daten als binäre Strings behandelt werden und das Format der Daten keinen Einfluss hat.

XAdES: Analog zu CAdES wurde der Standard XAdES ins Leben gerufen um – basierend auf XML-Signaturen (XMLDSIG) fortgeschrittene elektronische XML-basierte Signaturen zu ermöglichen. XAdES steht hierbei für XML Advanced Electronic Signature und ist, ebenso wie CAdES, ein ETSI Standard, veröffentlich unter ETSI TS 101 903. Analog zu CAdES erfüllen die Ausprägungen XAdES-BES und EPES die Anforderungen einer fortgeschrittenen Signatur. XMLDSIG und somit XAdES nutzen einen Referenzierungsmechanismus um Daten zu signieren und auch so genannte XAdES-Properties in die Signatur miteinzubetten. Durch das Einbetten dieser Properties erhält man fortgeschrittene Signatur. XAdES-Signaturen eignen sich offensichtlich speziell für XML-basierte bzw. text-basierte Daten. Andere Daten lassen sich prinzipiell auch über eine Base64-Codierung signieren. Dies hat jedoch eindeutige Performance-Nachteile.

PAdES:PAdES steht für PDF Advanced Electronic Signatures und ist ebenso ein ETSI-Standard, der als ETSI TS 102 778 veröffentlicht ist. Der Standard setzt dabei auf dem PDF-Standard auf und umfasst Einschränkungen und Erweiterungen um den Anforderungen für fortgeschrittene elektronische Signaturen zu genügen. Dafür wurden in PAdES Teil 3 Profile für PAdES-BES und PAdES-EPES Signaturen spezifiziert. Der PDF Standard spezifiziert nun, dass für eine PDF-Signatur eine CMS-Signatur in das Dokument eingebettet wird. Analog gilt das auch für PAdES. D.h. für eine PAdES-Signatur wird entsprechende CAdES-Signatur eingebettet. PAdES-Signaturen eignen sich prinzipiell nur für PDF-Dateien. Mittels XFA[4] (XML Forms Architecture) würden sich prinzipiell auch XML-Daten signieren lassen, hat jedoch in der Praxis keine Relevanz.

4.2 Analyse OGD Formate

Im Bereich von Open Government Data wird eine Reihe von unterschiedlichen Datenformaten eingesetzt. Die Palette reicht hier von simplen textbasierten Formaten, über strukturierte Daten zu Containerformaten. Teils sind diese Datenformate für den allgemeinen Einsatz gedacht (z.B. CSV) oder sie sind auf bestimmte Einsatzgebiete beschränkt (z.B. geographische Datenformate, wie GML).

[3] Abstract Syntax Notation One.

[4] Hiermit lassen sich XML-kodierte Daten in PDF-Dokumenten übertragen.

Prinzipiell kann man bei Datenformaten zwischen strukturierten und nicht strukturierten Daten unterscheiden. Strukturierte Daten erfüllen eines der wichtigsten Grundprinzipien von Open Government Data und sind maschinenlesbar und somit geeignet zur maschinellen Weiterverarbeitung. Im Gegensatz dazu bieten nicht strukturierte Datenformate diese Möglichkeit nicht oder nur in einem beschränktem Ausmaß. Sie werden größtenteils dazu benutzt die Daten visuell ansprechend darzustellen. Sowohl bei strukturierten als auch nicht strukturierten Format kommen teils auch Containerformate zum Einsatz. Hier besteht das Datenformat nicht nur aus einer Datei, sondern aus mehreren. Diese Dateien werden dann in einem eigenen Container zusammengefasst und sind danach somit in sich abgeschlossen.

Im Folgenden werden die wichtigsten und gängigsten[5] Datenformate, die innerhalb von Open Government Data eingesetzt werden, analysiert.

Anmerkung: Schnittstellen, die teils fälschlicherweise auch als Datenformat bezeichnet werden, werden nicht behandelt. Das trifft insbesondere auf folgende Schnittstellen zu: WMS (Web Map Service), WFS (Web Feature Service), RSS (Really Simple Syndication) und WMTS (Web Map Tile Service).

CSV: CSV steht für Comma-separated values und ist eine Textdatei, die es ermöglicht einfach strukturierte Daten auszutauschen. Das Format ist allgemein gültig, d.h. es können prinzipiell beliebige Daten damit übertragen werden, insofern sie einer einfachen Struktur folgen. Die Strukturierung erfolgt dabei über eigene Trennzeichen für Datensätze und Datenfelder. Da CSV Daten textbasiert sind, lassen sie sich sowohl mit CAdES und XAdES signieren. Welcher Signaturvariante der Vorzug gegeben wird hängt dabei vom konkreten Anwendungsfall ab.

XML: XML steht für eXentsible Markup Language und ist eine Auszeichnungssprache, die von W3C spezifiziert ist. Die Spezifikation hat formell den Status einer Empfehlung, ist aber seit vielen Jahren ein de-facto Standard. XML ist ein reines Textformat, bietet jedoch die Möglichkeit der strukturierten Beschreibung unterschiedlichster Daten. Um eine konkrete Struktur zu definieren bedient man sich im Allgemeinen eines XML Schemas, das sehr komplexe Strukturierungen zulässt. Für XML-Daten eignen sich offensichtlich XAdES-Signaturen perfekt. Prinzipiell können XML-Daten auch mit CAdES signiert werden. Dies erscheint aber wenig sinnvoll und wird in der Praxis auch nicht angewendet.

KML: KML[6] ist die Abkürzung für Keyhole Markup Language und ist ein, auf XML basierende, Auszeichnungssprache für Geodaten basierend auf Google Earth und Google Maps. Diese Sprache ist dabei sehr mächtig, da sie neben geographischen Informationen (wie Punkte, Linien, Bilder, etc.) auch bestimmte Sachverhalte auf Geodaten abbilden kann. Es ist beispielsweise möglich die Informationen für die Verkehrsauslastung in bestimmten Regionen in einer KML-Datei abzuspeichern. KML basiert auf XML. Dadurch eignet sich XAdES sehr gut als Signaturformat. Für die komprimierte Variante als KMZ-Datei würde aber prinzipiell auch das CAdES-Format geeignet sein.

GML: GML ist ebenso wie KML eine Auszeichnungssprache für Geodaten. GML steht dabei für Geography Markup Language. Sie basiert, ebenso wie KML, auf XML. GML dient der standardisierten Kodierung von geographischen Informationen. D.h. GML wird dazu genutzt

[5] Basis für diese Entscheidung war eine Analyse der Datenformate verschiedener OGD Plattformen wie data.gv.at, offenedaten.de und data.gov.uk.

[6] Auch als KMZ bekannt, wobei KMZ einen ZIP-komprimierte KML-Datei darstellt.

um Objekte (wie Straßen, Häuser, Brücken, etc.) inkl. ihrer Eigenschaften zu beschreiben. In Kontrast dazu dient KML dazu der Visualisierung geographischer Informationen (über Ortsmarken beispielsweise). So kann zum Beispiel das GML-Objekt „Grazer Hauptplatz" über eine KML Ortsmarke visualisiert werden. GML basiert auf XML. Dadurch eignet sich XAdES sehr gut als Signaturformat. Prinzipiell kann man auch die GML-Datei Base64-kodieren und dann mittels CAdES signieren. Dieser Zugang ist aber praktisch nicht sinnvoll.

SHP: SHP bedeutet Shapefile und wurde vom Environmental Systems Research Institute spezifiziert. Es ist ein Vektor-Datenformat für Geodaten. Durch seinen langen Bestand und die weite Verbreitung im GIS[7]-Umfeld hat SHP sich als de-facto Standard in diesem Bereich herauskristallisiert. SHP repräsentiert dabei beispielsweise Flüsse, Seen oder Brücken über Vektor-Funktionen, wie Punkt, Linien oder Polygone. Durch das zur Verfügung stellen in einem Containerformat eignen sich CAdES-Signatur am besten zur Sicherung der Integrität und Authentizität. Auch an dieser Stelle könnte man prinzipiell wieder die Base64-Kodierung mittels XAdES signieren. Das ist aber wiederum weder sinnvoll noch praktikabel.

SVG: SVG ist die Abkürzung für Scalable Vector Graphics und ist eine W3C Empfehlung. Dieses Format dient der Beschreibung (zweidimensionaler) Vektorgrafien. SVG ist dabei XML basiert, kann jedoch auch komprimiert gespeichert werden. SVG basiert auf XML. Dadurch eignet sich XAdES sehr gut als Signaturformat. Für die komprimierte Variante würde aber prinzipiell auch das CAdES-Format geeignet sein.

PDF: PDF steht für Portable Document Format und ist ein seit vielen Jahren etablierter Standard. Seit der Version 1.7 ist PDF ein ISO-Standard. Der PDF-Standard ist sehr umfangreich und basiert auf PostScript. Der Hauptzweck von PDF ist stark auf die Repräsentation und Visualisierung von elektronischen Inhalten fokussiert. Auch erlaubt PDF die Einbindung von Formularen, Kommentaren und dynamischen Inhalten. Für PDF-Daten eignen sich offensichtlich PAdES-Signaturen perfekt. Prinzipiell könnte man auch die PDF-Daten Base64 codieren und mittels XAdES signieren. Dieser Variante erscheint aber weder sinnvoll noch praktikabel.

ZIP: Das ZIP-Dateiformat dient zweierlei Zwecken. Zum einen wird es als Containerformat genutzt um mehrere Dateien und Verzeichnis zusammenzufassen. Andererseits dient es auch der Komprimierung der enthaltenen Dateien. Das Format gibt es seit 1989 und hat sich schnell zu einem weltweiten Standard herauskristallisiert. ZIP-Daten eignen können prinzipiell wieder Base64-kodiert mittels XAdES signiert werden. Für den praktischen Einsatz eigen sich jedoch CAdES-Signaturen weit besser.

4.3 Zusammenfassung Signaturfähigkeit

Im den vorhergehenden Abschnitten wurde die gängigsten OGD-Datenformate analysiert – speziell hinsichtlich ihrer Signaturfähigkeit. Tabelle 1 fasst die Ergebnisse zusammen. Es konnte für jedes OGD-Datenformat zumindest ein sinnvolles und praktikables Signaturformat gefunden werden. Wobei speziell XAdES und CAdES – je nach OGD Format – die beste Wahl sind. PAdES-Signaturen sind hierbei nur für PDF-Daten geeignet.

Tab. 1: Zusammenfassung Signaturfähigkeit OGD Formate.

Format	Kurzbeschreibung	CAdES	XAdES	PAdES

[7] Geographische Informationssysteme.

CSV	Text basiertes Format mit der Möglichkeit einer einfach strukturierten Angabe der Daten über Trennzeichen.	✓	✓	o
XML	Text basiertes Format mit der Möglichkeit sehr komplexe Strukturen über XML-Elemente abzubilden.	o	✓	o
KML/ KMZ	XML basiertes Geodatenformat speziell für Google Earth und Google-Maps	✓[8]	✓	o
GML	XML basiertes Geodatenformat zur Beschreibung von Objekten und deren Eigenschaften	o	✓	o
SHP	Geodatenformat speziellen für den Einsatz in GIS-Anwendungen und aus mehreren Dateien bestehend	✓	o	o
SVG	XML basiertes Format für skalierbare Vektorgrafiken	✓[9]	✓	o
PDF	Datei im weltweit etablierten PDF-Format	x	o	✓
ZIP	Containerformat in dem beliebige Dateien komprimiert gespeichert werden können	✓	o	o

5 Architektur

5.1 Einleitung

Die Architektur für ein vertrauenswürdiges OGD umfasst sowohl eine Server- als auch einen Clientseite. Die Serverseite befindet sich in der Domäne der OGD Bereitstellerin bzw. der OGD Bereitstellers und ist verantwortlich für das Aufbringen der elektronischen Signatur auf die zu veröffentlichenden Daten. Clientseitig, d.h. von der OGD Bezieherin bzw. dem OGD Bezieher, werden diese signierten Daten konsumiert. Die folgenden Unterabschnitte befassen sich nun detaillierte mit der server- und clientseitigen Architektur.

5.2 Serverseitige Architektur

Abbildung 2 veranschaulicht die serverseitige Architektur und zeigt wie diese in eine existierende Infrastruktur integriert werden kann. Die existierende Infrastruktur wurde hierzu generalisiert und besteht aus folgenden Komponenten (inklusive dem ursprünglich Prozessfluss):

- *Datenquelle*: Repräsentiert die originale Datenquelle, die als Basis für die Erzeugung von OGD herangezogen wird.
- *OGD Vorbereitung*: Diese Komponente generiert die OGD Daten aus der originalen Datenquelle und übermittelt diese an die Komponente OGD Veröffentlichung.
- *OGD Veröffentlichung*: Die generierten OGD Daten werden von dieser Komponenten in geeigneter Weise veröffentlicht.

[8] Prinzipiell eignet sich XAdES für KML besser. Für die ZIP-komprimierte Variante (KMZ) kommt aber auch CAdES in Frage.

[9] Prinzipiell eignet sich XAdES für SVG besser. Für die ZIP-komprimierte Variante kommt aber auch CAdES in Frage.

Diese existierende Architektur wurde nun erweitert um ein vertrauenswürdiges OGD zu ermöglichen. Diese erweiterte Architektur wurde modulmäßig aufgebaut und besteht aus folgenden Komponenten (inklusive deren Interaktion und dem Prozessfluss):

- *Konfiguration*: Diese Komponente beinhaltet die gesamte Konfiguration und stellt entsprechende Schnittstellen zur Abfrage der Konfiguration zur Verfügung. Dabei enthält die Konfiguration zumindest folgende Daten: (1) die Mapping der OGD Datenformat auf das entsprechende Signaturformat (siehe Abschnitt 4.2) und (2) die privaten Schlüssel, die zur Erzeugung der Signatur benötigt werden. Diese Schlüssel können dabei Softwareschlüssel (als Datei hinterlegt) oder auch Hardwareschlüssel (über ein Hardware Security Modul eingebunden) sein
- *Datenformat Erkennung*: Aufgrund der individuelle Charakteristik der verschiedenen OGD Datenformate, versucht diese Komponente das Datenformat der einlangenden OGD Daten herauszufinden. Die OGD Daten selbst und das erkannte Datenformat werden in weiterer Folge an den Datenformat Broker übermittelt.
- *Datenformat Broker*: Basierend auf dem konfigurierten Format-Mapping wählt der Broker das entsprechende Signaturformat für die eingelangten OGD Daten aus und leitet die OGD Daten an den Connector weiter.
- *Connector*: Der Connector stellt eine gemeinsame Schnittstelle zwischen Datenformat Broker und den Signatoren für die verschiedenen Signaturformate zur Verfügung.
- *Signatoren*: Basierend auf der EU Kommissionsentscheidung 2011/130EU und dem Ergebnis präsentiert in Abschnitt X sind derzeit drei verschiedene Signatoren definiert (PAdES-, CAdES- und XAdES-Signator). Jeder Signator bekommt über die Konfiguration den entsprechenden privaten Schlüssel und signiert damit die erhaltenen Daten entsprechend der jeweiligen Spezifikation. Durch die abstrahierte gemeinsame Schnittstelle können an dieser Stelle auch weitere zusätzliche Signatoren prinzipiellen hinzugefügt werden.

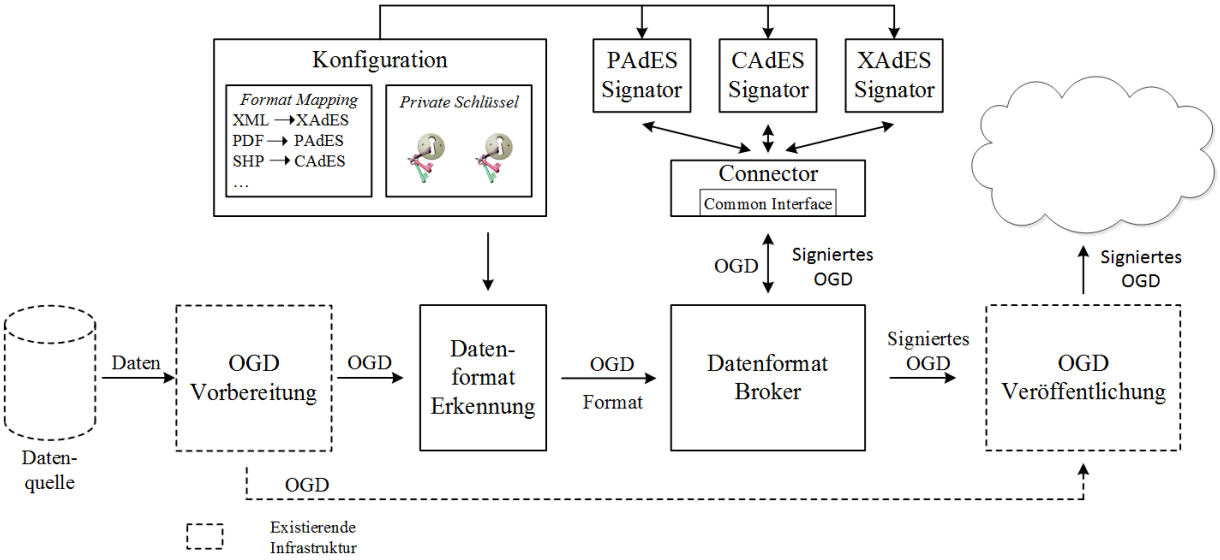

Abb. 2: Serverseitige Architektur

5.3 Clientseitige Architektur

Abbildung 3 illustriert die clientseitige Architektur. Die Architektur ist allgemein gehalten, das sie als Smartphone-App, als Webapplikation oder jeden anderen Applikationstyp angewendet werden kann. Die involvierten Komponenten und der Prozessfluss sind:

- *Daten Retrieval*: Diese Komponente bezieht die signierten OGD Daten von der veröffentlichten Quelle und leitet diese an den Signaturprüfungs-Broker weiter.
- *Signaturprüfungs-Broker*: Dieser Broker startet den Signaturprüfprozess und evaluiert dann das erhaltene Prüfergebnis. Abhängig von der konkreten Implementierung kann die Signaturprüfung über eine interne (selbst implementierte) Signaturprüfung erfolgen oder es wird ein externe Signaturprüfdienst genutzt. Nach der Evaluierung des Prüfergebnisses erfolgt jedenfalls die Weiterleitung an die App Engine.
- *Interne Signaturprüfung und externer Signaturprüfdienst*: Diese Komponenten übernehmen die eigentliche Signaturprüfung. Entweder kann eine eigene – interne – Signaturprüfkomponente implementiert werden oder man nutzt einen externen Signaturprüfdienst. Externe Prüfdienste sind im Allgemeinen vorteilhaft da sie mehrere Signaturformate unterstützen und zumeist auch eine Webschnittstelle zur Verfügung stellen, die weit leichter zu nutzen ist als selbst eine komplexe Signaturprüfung zu implementieren. Speziell Applikationen, die wenig Ressourcen zur Verfügung haben, wie Smartphone-Apps, können davon profitieren.
- *App Engine*: Die App Engine repräsentiert die eigentliche Applikation und dient der Visualisierung der erhaltenen Daten, wenn die Signatur erfolgreich geprüft werden konnte.

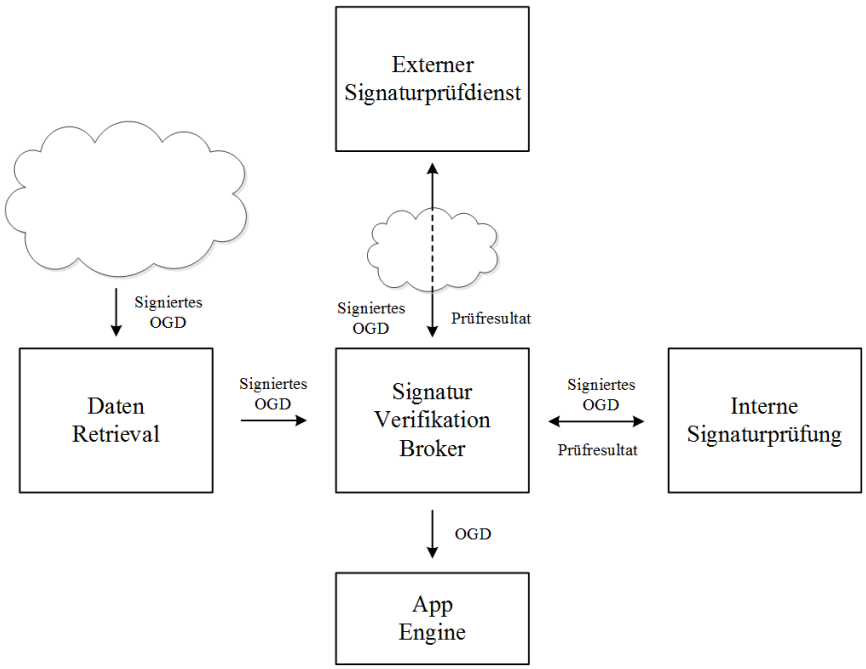

Abb. 3: Clientseitige Architektur

6 Implementierung

6.1 Einleitung

Zur Evaluierung der vorgeschlagenen Architekturen wurde eine prototypische Implementierung der Server- und Client-Seite vorgenommen. Die folgenden Unterabschnitte befassen sich mit diesen beiden Implementierungen.

6.2 Serverseitige Implementierung

Für die serverseitige Komponente wurde ein prototypischer Dienst implementiert, dem OGD Daten übergeben werden und die signierten OGD Daten zurückliefert. Abbildung 4 zeigt diese konkrete serverseitige Umsetzung. Die Datenformat Erkennung beruht dabei auf der Format-Erkennung von [Zeff11]. Für die Erstellung der Signatur dient das Open-Source Modul MOA-SS[10] zur Erstellung von serverseitigen CAdES bzw. XAdES Signaturen. Es werden daher sämtliche OGD Datenformate unterstützt, die zumindest auf eines der beiden Formate mappen. Nach Erhalt der signierten OGD Daten können diese entsprechende veröffentlicht werden.

Diese prototypische Umsetzung steht dabei als Webapplikation als auch als Command-Line Version als Anhang zu diesem Dokument zur Verfügung.

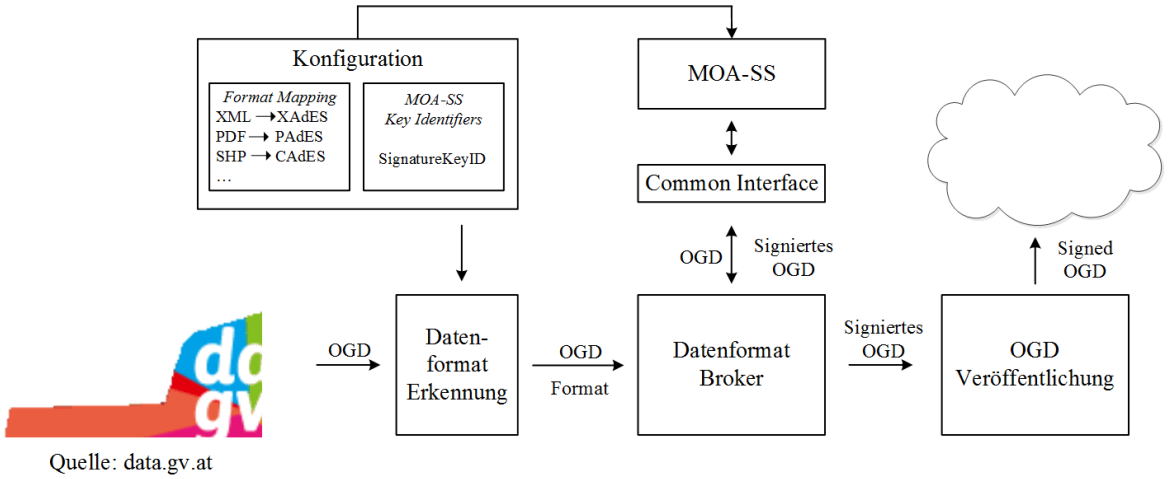

Abb. 4: Serverseitige Implementierung

6.3 Clientseitige Implementierung

Auf Client-Seite wurde eine prototypische Android App implementiert, die einen konkreten signierten OGD Datensatz einliest, die Signaturprüfung vornimmt und die Daten anschließend darstellt. Als OGD Datensatz wurde dabei "NEXTBIKE NÖ Fahrradverleihsystem"[11] von der Plattform data.gv.at herangezogen. Diese XML-basierten Daten repräsentierten die Verfügbarkeit von Fahrrädern bei verschiedenen Verleihstationen im Bundesland Niederösterreich. Der gesamte Datensatz wurde mittels der serverseitigen Implementierung signiert und auf einem Webserver veröffentlicht.

[10] https://joinup.ec.europa.eu/software/moa-idspss

[11] http://www.data.gv.at/datensatz/?id=96d176fb-dfd4-49de-91fc-b4997ab353ba

Diese signierten OGD Daten werden von der Android App eingelesen. Anschließend erfolgt die Signaturprüfung, wobei diese – aufgrund der limitierten Ressourcen – über einen externen Signaturprüfdienst erfolgt. Da die entsprechende Web-Service Schnittstelle vom Signaturprüfservice der RTR[12] unter http://signaturpruefung.gv.at noch nicht zur Verfügung steht, wurde stattdessen das entsprechende Service von http://www.buergerkarte.at/signature-verification/ genutzt. Nach einer erfolgreichen Signaturprüfung (siehe Meldung „Valid“ in Abbildung 5 - links) werden die Daten im Listenformat angezeigt. Bei Auswahl einer Verleihstation (siehe Abbildung 5 - rechts) gibt es noch die Möglichkeit sich diese Station auf Google Maps anzeigen zu lassen.

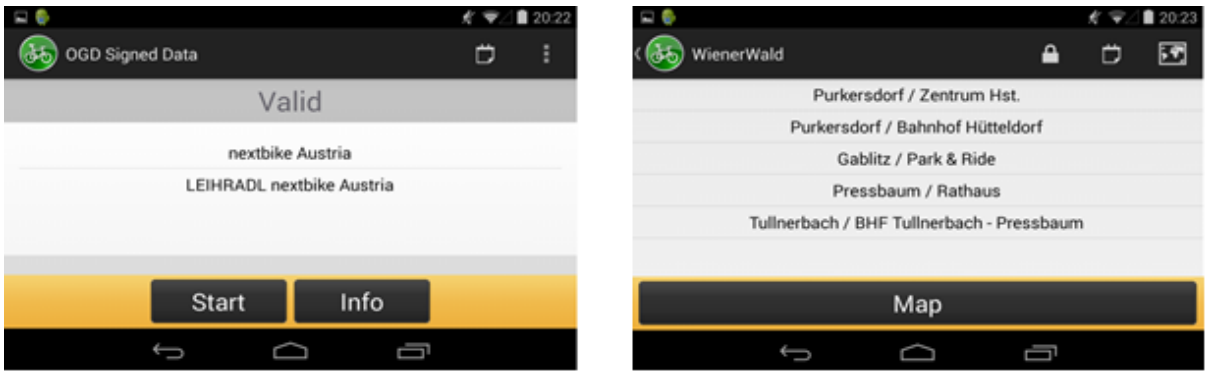

Abb. 5: Screenshots Android App

7 Zusammenfassung

Unsere Implementierung hat gezeigt, dass ein vertrauenswürdiges OGD möglich und realisierbar ist. Die Implementierung hat ebenso gezeigt, dass sie die gestellten Anforderungen aus Abschnitt 3 erfüllen kann. Ein vertrauenswürdiges OGD kann dabei immer eingesetzt werden, wenn die Authentizität und Integrität der Daten von Wichtigkeit ist und hat sowohl für OGD Bereitstellerinnen bzw. Bereitsteller als auch OGD Bezieherinnen bzw. Bezieher klare Vorteile.

Literatur

[COGD12] Projektgruppe Cooperation Open Government Data Österreich, Rahmenbedingungen für Open Government Data Plattformen, White Paper, Version 1.1.0, 30.07.2012

[Euro00] European Union, Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures, Official Journal L 13, pp.12-20 19.01.2000.

[Euro03] European Union (2003) Directive 2003/98/EC of the European Parliament and the Council of 17 November 2003 on the re-use of public sector information.

[Euro11] European Commission Decision, Establishing minimum requirements for the cross-border processing of documents signed electronically by competent authorities under Directive 2006/123/EC of the European Parliament and of the Council on services in the internal market, notified under document C(2011) 1081, 2011/130/EU, 25.02.2011.

[12] Österreichische Rundfunk und Telekom Regulierungs GmbH, https://www.rtr.at/

[Euro13] European Union (2013), Directive 2013/37/EU of the European Parliament and of the Council of 26 June 2013 amending Directive 2003/98/EC on the re-use of public sector information.

[OGW07] Open Government Working Group, 8 Principles of Open Government Data, http://www.opengovdata.org/home/8principles, 2007

[StKZ12] Stranacher Klaus, Krnjic Vesna und Zefferer Thomas: Vertrauenswürdiges Open Government Data. 1.OGD D-A-CH-LI Konferenz. Seiten 27-39

[Zeff11] Zefferer et al., Secure and Reliable Online-Verification of Electronic Signatures in the Digital Age, in Proceedings of the IADIS International Conference WWW/INTERNET, 2011, pp. 269-276