Eine verteilte Autorisierungsinfrastruktur unter Berücksichtigung von Datenschutzaspekten

Stefan Dürbeck · Jan Kolter
Günther Pernul · Rolf Schillinger

Zugriffskontrolle in IT-Systemen stellt seit jeher eine besondere Herausforderung an Systemarchitekten dar. Nur wenige Zugriffskontrollverfahren sind für den Einsatz in verteilten Systemen geeignet, um den Anforderungen der Dienstanbieter sowie der Benutzer gerecht zu werden.

Einleitung

Etliche Forschungsprojekte aus dem Bereich der Informationssicherheit befassen sich seit einigen Jahren mit Architekturen, die versuchen, den Schutz persönlicher Nutzerdaten mit dem für die effektive Zugriffskontrolle von behördlichen und geschäftlichen Transaktionen nötigen Informationsbedürfnis seitens der Dienstanbieter in Einklang zu bringen. Besonders die Bereiche der eCommerce- und eGovernment-Anwendungen stellen hier einen Brennpunkt der beiden Interessenssphären von Zugriffskontrolle und Datenschutz dar, da im kommerziellen und behördlichen Umfeld in großem Maße sensible Personendaten in die Bearbeitung durch (elektronische) Dienste von hoher Outputqualität einfließen. Die sich wandelnde IT-Landschaft im behördlichen Umfeld einerseits und einschlägige politisch-rechtliche Vorgaben (wie die EU-Richtlinien zum Datenschutz und zur Dienstleistungserbringung [11, 12]) andererseits schaffen ein Umfeld, in dem neue Antworten auf die widerstrebenden Interessen von Datenschutz und datenintensiven Geschäftsprozessen gefunden werden müssen. Das europäische Forschungsprojekt Access-eGov (http://www.accessegov.org) hat es sich zum Ziel gesetzt, in diesem Umfeld eine verteilte serviceorientierte AAI zu entwickeln, in der den Anforderungen dynamischer Autorisierung und datenschutzgerechter Zugriffskontrolle eine besondere Bedeutung beigemessen wird [9].

Grundlagen

Im Umfeld verteilter Systeme existieren schon seit geraumer Zeit verschiedene grundlegende Zugriffskontroll- und Datenschutzkonzepte. Sie bilden die Ausgangslage für das Verständnis und zur Einordnung und sollen im Folgenden überblicksweise vorgestellt werden.

Zugriffskontrollverfahren

Unter Zugriffskontrolle (Access Control) versteht man allgemein das automatisierte, gegebenenfalls auf Regeln gegründete Verhindern unerlaubter oder missbräuchlicher Ressourcenzugriffe beziehungsweise das Ermöglichen regelkonformer Nutzung derselben. Da Vertraulichkeit eine Grundvoraussetzung jeglicher Art der Interaktion darstellt, wird die Autorisierung (Festlegung der Nutzerprivilegien) und damit einhergehend die Zugriffskontrolle als ein Basisdienst verlässlicher IT-Systeme angesehen. Über die Zeit haben sich verschiedene Autorisierungsmodelle entwickelt, von denen die benutzerbestimmbare Zugriffskontrolle (discretionary access control, DAC), die regelbasierte (mandatory access control, MAC) und die rollenbasierte Zugriffskontrolle (role-based access control,

DOI 10.1007/s00287-009-0411-0

Stefan Dürbeck · Jan Kolter · Günther Pernul · Rolf Schillinger
Lehrstuhl für Wirtschaftsinformatik I – Informationssysteme,
Universität Regensburg,
Universitätsstraße 31, 93053 Regensburg
E-Mail: {stefan.duerbeck, jan.kolter, guenther.pernul,
rolf.schillinger}@wiwi.uni-regensburg.de

Published online: 14 January 2010

Zusammenfassung

Traditionelle Verfahren der Rechtezuweisung (Autorisierung) und Zugriffskontrolle sind nur eingeschränkt geeignet, die Anforderungen an das Management der Nutzerprivilegien und an die Durchsetzung einer Sicherheitsstrategie in skalierbaren und hoch flexiblen verteilten Systemen umzusetzen. Dafür besser geeignet sind Sicherheitsinfrastrukturen, genauer AAIs – authentication and authorization infrastructures – und PMIs – privilege management infrastructures – die in der Lage sind, umfassende Sicherheitsdienstleistungen in einer Föderation von Systemen aus unterschiedlichen Domänen anzubieten. Dieser Beitrag enthält die Darstellung einer datenschutzorientierten AAI im Umfeld von eGovernment, die attributbasierte Zugriffskontrolle, eine XACML-Sicherheitsarchitektur zur Umsetzung und eine besondere Berücksichtigung der Datenschutzaspekte bei der Weitergabe der Nutzerattribute beinhaltet.

RBAC) die wichtigsten Vertreter darstellen [18]. Diese Ansätze stammen allerdings mehrheitlich aus der Zeit geschlossener Einzelanwendungen oder spezifischer Anwendungsgebiete und genügen daher nur eingeschränkt den Anforderungen an weitläufige, skalierbare und verteilte Umgebungen, wie bspw. den eingangs geschilderten.

Flexibler und dafür besser geeignet zeigen sich attributbasierte Zugriffskontrollrichtlinien (attribute-based access control, ABAC), in denen allgemeine Nutzerattribute zur Laufzeit ausgewertet, mit Objekt- und Umfeldbedingungen abgeglichen und dadurch die Zugriffskontrollentscheidung dynamisch (ereignisgesteuert und zur Laufzeit) gestaltet werden kann. Aufgrund ihrer flexiblen Architektur und der Fähigkeit, auch vielschichtige Zugriffskontrollsemantiken auszudrücken [23], eignen sich ABAC-Modelle besonders für den Einsatz in SOAs [27]. Das ABAC-Prinzip unterscheidet sich von bisherigen Ansätzen durch den Verzicht auf die statische Festlegung von Zugriffsregelungen zugunsten von dynamischen Verfahren zur Ausübung von Subjektrechten auf Objekte [23].

Die XACML-Spezifikation [20] ist als OASIS-Standard verfügbar und unterstützt das Hauptmerkmal des ABAC-Ansatzes, die Integration von Subjekt- und Objektattributen in Zugriffsrichtlinien (access policies). Der Standard definiert eine mächtige Policy-Sprache [20] zur Zugriffskontrolle, auf die an dieser Stelle aber nicht näher eingegangen wird. Zusammen mit der Policy-Sprache legt der XACML-Standard eine Autorisierungsinfrastruktur fest, die generisch genug ist, um ein Autorisierungsmodell nach ABAC zu implementieren [22]. Um dem Einsatzgebiet der verteilten Systemlandschaften gerecht zu werden, unterscheidet die XACML-Architektur auch auf logischer Ebene nach Komponenten zur Policy Administration (Verwaltung), zum Policy Enforcement (Durchsetzung) und zur Policy Evaluation (Auswertung oder Abgleich). Hierfür definiert XACML die Funktionsbausteine Policy Enforcement Point (PEP), Policy Administration Point (PAP), Policy Decision Point (PDP) und Policy Information Point (PIP). Jede der genannten Komponenten hat jeweils nur eine spezifische Aufgabe im Zugriffskontrollprozess (Abb. 1): Der PEP nimmt Zugriffsanfragen entgegen und leitet diese weiter an den PDP, der für den regelbasierten Vergleich der überreichten Attribute und die Entscheidung über die Zugriffsgewährung zuständig ist. Die Access Policy, anhand derer der PDP seine Entscheidung trifft, wird vom PAP erfragt, der die Speicherung und Verwaltung von Zugriffsrichtlinien verantwortet. Der PIP stellt sog. Attribute-Authority-Informationen über Attribute und Zugang zu diesen zur Verfügung (z. B. zu lokal an den PIP angebundenen Verzeichnisdiensten mit entsprechender lokal bekannter Autorisierung). Die XACML-Architektur kennt auch einen Context Handler, der Datenflüsse zwischen den einzelnen Komponenten leitet und kontrolliert. Die Web-Service-Profile-Spezifikation von XACML (WS-XACML) [2] definiert die Anwendung von XACML in einem SOA-Umfeld. Während WS-Security [21] sogenannte Security Tokens für Webservices beschreibt, spezifiziert WS-XACML multifunktionale Authorization Tokens, um den Transfer der Zugriffsentscheidung zu einem PDP zu ermöglichen. Somit wird der PDP auch in die Lage versetzt, Zugriffsentscheidungen im Auftrag des eigentlichen Dienstanbieters zu treffen, dessen Ressource geschützt werden soll. Darüber hinaus beschreibt WS-XACML das Format der Autorisierung, der Zugriffskontrolle und der Datenschutzrichtlinien für Webservices, die nicht im

Abstract
Common best-of-breed practices of authorisation and access control are not capable of meeting today's requirements for the management of user privileges and enforcing a security strategy in scalable and highly flexible distributed systems. Some security infrastructures, so called AAIs – authentication and authorization infrastructures – and PMIs – privilege management infrastructures – are better suited and capable of offering fully-fledged security services in a federation of systems out of various domains. This article comprises a privacy-oriented AAI for eGovernment that incorporates attribute-based access control and a XACML-based security architecture to enforce privacy when it comes to propagating user attributes across system boundaries.

allgemeinen WS-Policy-Standard der W3C enthalten sind [26].

Datenschutz und Nutzerverhalten

Das Benutzerverhalten und die Art und Weise, wie Internetnutzer personenbezogene Daten preisgeben, haben sich in den vergangenen Jahren bedeutend gewandelt. Ein Hauptgrund für die größere Freizügigkeit im Umgang mit den eigenen Daten liegt in der immer größeren Beliebtheit und Nutzung von sozialen Netzwerken im Internet, die mittlerweile zu Multi-Millionen-Nutzerplattformen geworden sind. Besonders der Siegeszug des eCommerce und eben jener sozialer Netzwerke sowie personalisierter Foren ist ohne die Preisgabe personenbezogener Daten nicht möglich. Eine steigende Anzahl an Nutzern nimmt jedoch gerade diesen Trend zur Weitergabe von Daten an immer neue Dienstanbieter als Bedrohung persönlicher Bewegungsfreiheit im Internet wahr [10]. Als ein Beispiel einer datenschutzfördernden Technologie sei die „Platform for Privacy Preferences" (P3P) [7] genannt, die diese Bedenken wahrnimmt und eine Datenschutzrichtliniensprache (privacy policy language) bereitstellt, die es den Dienstbetreibern erlaubt, ihre jeweiligen Datenschutzrichtlinien in standardisierter Form zu veröffentlichen. Eine P3P-Policy beschreibt, wie personenbezogene Nutzerdaten behandelt werden, einschließlich des Zwecks der Datensammlung und der weiteren Verwendung. Auf der Nutzerseite werden dabei die Datenschutzpräferenzen zusammengestellt und beispielsweise in die „A P3P Preference Exchange Language" (APPEL) genannte Sprache übertragen [8]. Ein sogenannter Privacy Agent nutzt daraufhin ebenjene Informationen, um zu überprüfen, ob die Datenschutzrichtlinie einer zu besuchenden Website mit den Datenschutzinteressen des Besuchers kompatibel ist. Diese praktische Anwendung von P3P-Richtlinien ist mittlerweile seit geraumer Zeit dokumentiert [1, 3].

AAI mit ABAC und Datenschutz

In diesem Abschnitt beschreiben wir den konzeptuellen Aufbau einer ABAC-basierten Zugriffskontrollinfrastruktur und spiegeln den beispielhaften Ablauf einer Zugriffsanfrage unter Berücksichtigung von Datenschutzaspekten bei der Attributweitergabe wider. Zunächst wird jedoch näher auf die Grundidee und Zielsetzung der Architektur eingegangen.

Grundidee und Zielsetzung

Wie bereits eingangs beschrieben, besteht der grundlegende Gedanke eines ABAC-Systems darin, Zugriff auf Ressourcen aufgrund von statischen oder dynamischen Attributen eines Dienstnutzers (Subjekt) zu gewähren bzw. zu verweigern. Beim Einsatz für den Zugriffskontrollvorgang in SOAs und anderen flexibel interagierenden Systemarchitekturen bietet ABAC den Vorteil der Unterstützung von dynamischen Abläufen. Allgemein setzt ABAC allerdings die Offenlegung von personengebundenen und datenschutzbedürftigen Subjektattributen voraus, die Dienstanfrager oftmals nicht ohne Bedenken bereit sind zu übermitteln. Aufgrund der komplexen Beziehungen zwischen Akteuren in SOAs lässt sich die Vielfältigkeit dieser Subjektattribute grob erahnen [19]. Besonders Benutzer von weitläufigen und räumlich verteilten Mehrbenutzersystemlandschaften zeigen ein erhöhtes Maß an Zurückhaltung bei der unkontrollierten Weitergabe dieser Arten von Attributinformationen. Um diesen Bedenken in Form eines standardisierten Protokollablaufes Rechnung zu tragen, ist eine Sicherheitsarchitektur notwendig, die den üblichen ABAC-Systemansatz, wie er in der XACML-Spezifikation [20] festgelegt wurde, um Datenschutz gewährleistende Systemkomponenten erweitert. Dabei liegt der Schwerpunkt darauf,

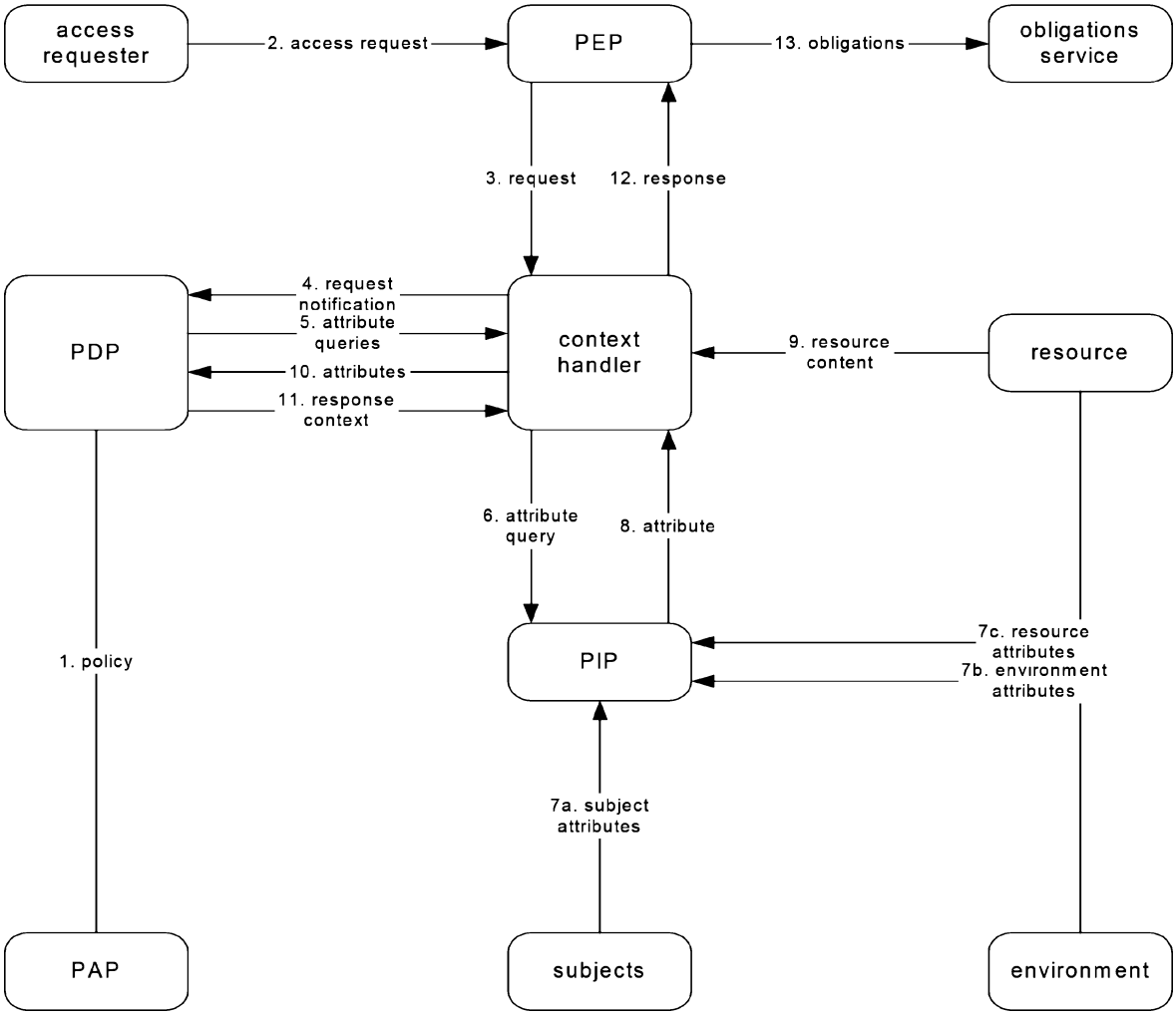

Abb. 1 Vollständige XACML-Architekturübersicht [20]

es dem Benutzer zu ermöglichen, den Datenfluss (Datenspeicherung und -weiterverarbeitung ausgenommen) seiner schutzbedürftigen Attribute zu kontrollieren. Da die manuelle, von Fall zu Fall zu entscheidende Zustimmung eines Nutzers bei jedem Dienstzugriff oftmals als unverhältnismäßig und unkomfortabel ausscheidet, erscheint die automatisierte Entscheidung mittels individueller, selbstkonfigurierter Datenschutzpräferenzen (d. h. Regeln zur Attributoffenlegung) als sinnvoll [7, 8]. Beim Zugriff auf eine Ressource, die durch ein ABAC-Verfahren geschützt wird, sollten diese Präferenzen zur Laufzeit dynamisch ausgewertet werden, wenn der Diensterbringer eine Sammlung an Attributen liefert, die der Dienstanfrager vorzuweisen hat. Wenn Nutzer ihre Datenschutzpräferenzen individuell festlegen, ist es nicht unwahrscheinlich, dass einige Attribute aufgrund von vorher bestimmten Regeln nicht zur Weitergabe an einen Dienstanbieter freigegeben werden. Aus diesem Grund machen wir uns den flexiblen, generischen Charakter des XACML-Ansatzes zu Nutze (s. Abschn. Zugriffskontrollverfahren), und trennen die PDPs räumlich und organisatorisch von den funktionalen Dienstanbietern [16]. Unter diesen Voraussetzungen ist eine direkte Übertragung von Nutzerattributen zum Dienstanbieter nicht nötig, da der ausgelagerte PDP als einziger Systemakteur die Attribute und deren jeweilige Werte kennen und bewerten muss. Die Trennung der PDPs vom Dienstanbieter ermöglicht sogar den Einsatz mehrerer redundanter PDPs in einem Rechnernetzwerk,

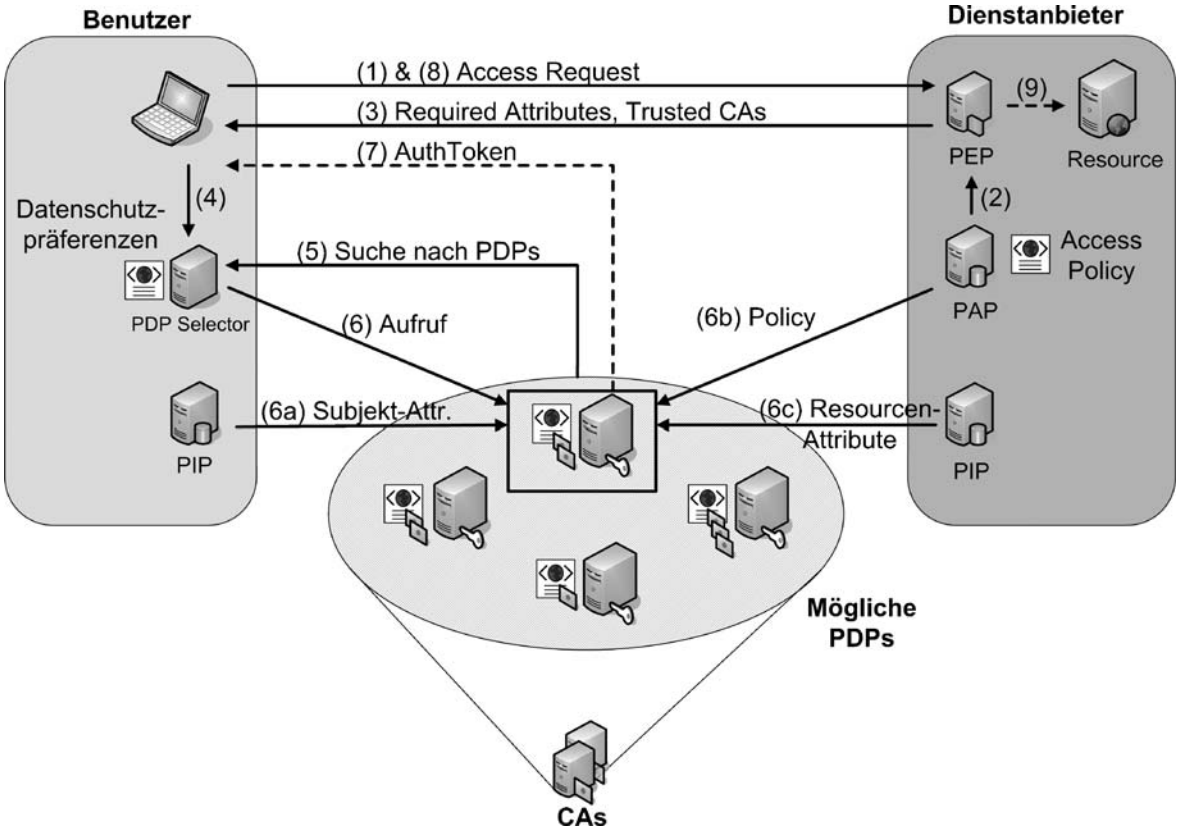

Abb. 2 Ein ABAC-System mit aus Datenschutzgründen verteilten PDPs

aus denen der Nutzer nach seinen eigenen Kriterien unter Datenschutzaspekten auswählen kann. Jeder PDP kann dabei eine Vielzahl an unterschiedlichen Funktionalitäten anbieten. In einem dynamischen Auswahlvorgang kann dann derjenige PDP ausgewählt werden, dessen funktionale Fähigkeiten am ehesten den Datenschutzpräferenzen des Benutzers entsprechen. Folgerichtig wird die Zugriffsentscheidung durch einen PDP vollzogen, dem der Benutzer ohne (oder mit weniger) Bedenken die Behandlung bestimmter Attribute anvertraut. An diesem Punkt sei anzumerken, dass die räumliche und organisatorische Trennung des PDP vom direkten Einfluss des Dienstanbieters nur dann möglich ist, wenn der Dienstanbieter den Bewertungs- und Entscheidungsalgorithmen des PDP vollständig vertraut und zusätzlich auch den Betreiber des PDP für absolut vertrauenswürdig hält. Die im Folgenden vorgestellte Architektur berücksichtigt diesen Aspekt und nutzt zur Sicherstellung dieses Vertrauensverhältnisses eine PKI. Im nächsten Abschnitt beschreiben wir die vorgeschlagene Infrastruktur aus konzeptueller Architektursicht und stellen den prototypischen Ablauf eines Zugriffskontrollvorganges dar.

Architektur und beispielhafter Protokollablauf

Das im vorigen Abschnitt dargestellte ABAC-System basiert auf der XACML-Architektur (s. Abschn. Zugriffskontrollverfahren). Darin wird der eigentliche Zugriffskontrollvorgang auf mehrere logische Systemakteure wie PEP, PDP, PAP und PIP aufgeteilt. Um nun dynamisch zur Laufzeit des Kontrollvorganges einen PDP auszuwählen, der den individuellen Datenschutzpräferenzen entspricht, wird die herkömmliche XACML-Architektur mit Elementen und Akteuren dahingehend erweitert, dass dem Datenschutz bzw. dem gegenseitigen Vertrauen während des Datenflusses ein besonderes Augenmerk zukommt.

Abbildung 2 zeigt die vorgeschlagene Architektur. Abgesehen von den zu erwartenden Zugriffskontrollakteuren zeigt die Übersicht eine

Anzahl an alternativen, redundanten PDPs. Der „PDP Selector“ auf Benutzerseite ist zuständig für die Suche, den Vergleich und die Auswahl eines geeigneten PDP. Die Möglichkeit, dass ausgelagerte PDPs Zugriffsentscheidungen außerhalb des Einflussbereichs des betreffenden Dienstanbieters ausführen können, setzt ein hohes Maß an gegenseitigem Vertrauen voraus, da allein aufgrund der Entscheidung des PDP der Zugriff auf eine Ressource gewährt wird. Zur Sicherstellung dieses Vertrauensverhältnisses schlagen wir den Einsatz einer PKI vor, die jeden PDP mit Vertrauenszertifikaten ausstattet.

Die Vertrauenszertifikate stellen sicher, dass ein bestimmter PDP dazu autorisiert wurde, eine bestimmte Art von Attributen zu evaluieren. Da nicht unbedingt jeder Dienstanbieter jeder Certificate Authority (CA) vertraut, kann prinzipiell jeder PDP beliebig viele Zertifikate unterschiedlicher Aussteller für unterschiedliche Attribute vorhalten.

Ein beispielhafter Zugriffskontrollvorgang beginnt wie in Abb. 2 damit, dass ein Benutzer (Subjekt) eine geschützte Ressource eines Dienstanbieters anfrägt (1). Der zuständige PEP erhält die Zugriffsanfrage (Access Request) und kontaktiert denjenigen PAP, der die Access Policy (Zugriffsrichtlinie) der betreffenden Ressource vorhält (2). Der PAP verarbeitet dann die Zugriffsrichtlinie und leitet daraus die für die Zugriffsentscheidung nötige Menge an Attributtypen (Required Attributes) ab, die dann an den Benutzer zurückgesendet werden (3), welche dieser wiederum im Rahmen der Zugriffskontrolle preiszugeben hätte, wie z. B. das Geburtsdatum oder die Kreditkartennummer. Hierbei wird dem anfragenden Subjekt auch eine Liste an CAs genannt, denen der Dienstanbieter vertraut. Die benötigten Subjektattribute können auch alternative bzw. sich gegenseitig ausschließende Attribute enthalten. Um die Chancen auf die Zugriffserlaubnis zu erhöhen, kann sich ein Benutzer über mehrere alternative Attributtypen informieren lassen, die ein PDP akzeptieren würde. Nachdem die Attributinformationen übermittelt wurden, ruft der Benutzer den PDP Selector (4) auf, um nach möglichen PDPs zu suchen (5). In einem ersten Schritt filtert der PDP Selector alle diejenigen PDPs aus, die kein Zertifikat zur Bewertung der infrage kommenden Attribute besitzen. Der PDP Selector wird zusätzlich mit der Information aus den Schritten (2) und (3) versorgt. Wie zuvor erwähnt, muss ein passender PDP auch die Datenschutzpräferenzen des Dienstanfragers erfüllen. Anforderungen dieser Art können beispielswiese die Offenlegung der Kreditkartennummer ausschließlich auf Finanzinstitute beschränken. Ein Nutzer könnte auch festlegen, dass das Geburtsdatum nur dann übermittelt wird, falls eine physikalisch sichere Verbindung besteht. Aus diesem Grund publiziert ein möglicher PDP immer auch eine Anzahl an technischen Informationen und funktionalen Eigenschaften über geforderte Dienstattribute. Nachdem der PDP Selector die verbliebenen PDPs aussortiert hat, werden die funktionalen Fähigkeiten der PDPs mit den persönlichen Datenschutzpräferenzen des Benutzers verglichen. Falls nun die Kreditkartennummer und das Geburtsdatum vom Dienstanbieter gefordert werden, können nur PDPs unter der Kontrolle von Finanzinstituten bzw. solche mit sicherer Verbindung zur weiteren Bearbeitung der Zugriffskontrolle infrage kommen. Sollte die PDP-Auswahl keine Ergebnisse liefern, wird der Benutzer entsprechend benachrichtigt und der weitere Zugriffskontrollvorgang abgebrochen; sollten mehrere gleichwertige PDPs ausfindig gemacht werden, wird bisher automatisch der zuerst genannte gewählt. An dieser Stelle im Zugriffskontrollprozess wären auch weitere Optionen denkbar. Sobald nun ein passender PDP ausgewählt wurde, wird die ursprüngliche Dienstanfrage (6) zusammen mit den zur Überprüfung geforderten Subjektattributen vom nutzerseitigen PIP aus versendet (6a). Der PDP erhält vom PIP bzw. PAP des Dienstanbieters sowohl die Zugriffsrichtlinie (6b) als auch die Beschreibungen der geforderten Attribute (6c). In der Folge bewertet der PDP alle geforderten Attribute unter Berücksichtigung der Zugriffsrichtlinie und fällt eine Entscheidung. Im Falle einer positiven Zugriffsentscheidung („Zugriff wird gewährt“), gibt der PDP ein „Authorization Token“ an den Benutzer aus (7). Dieser verwendet das Token dazu, um auf die geschützte Ressource zuzugreifen (8). Das Token wird vom PEP geprüft, und nachdem es anerkannt wird, kann der Ressourcenzugriff gewährt werden (9).

Abbildung 3 zeigt ein Sequenzdiagramm des eben beschriebenen Zugriffskontrollvorganges, das die Rolle der jeweiligen Systemakteure innerhalb der Zugriffskontrollarchitektur verdeutlicht. Die technischen Details der Umsetzung mittels XACML werden in [17] näher geschildert. Der nächste Abschnitt gibt einen genaueren Überblick in die konkrete Integration der vorgeschlagenen Sicherheitsarchitektur und deren Umsetzung im EU-Projekt Access-eGov.

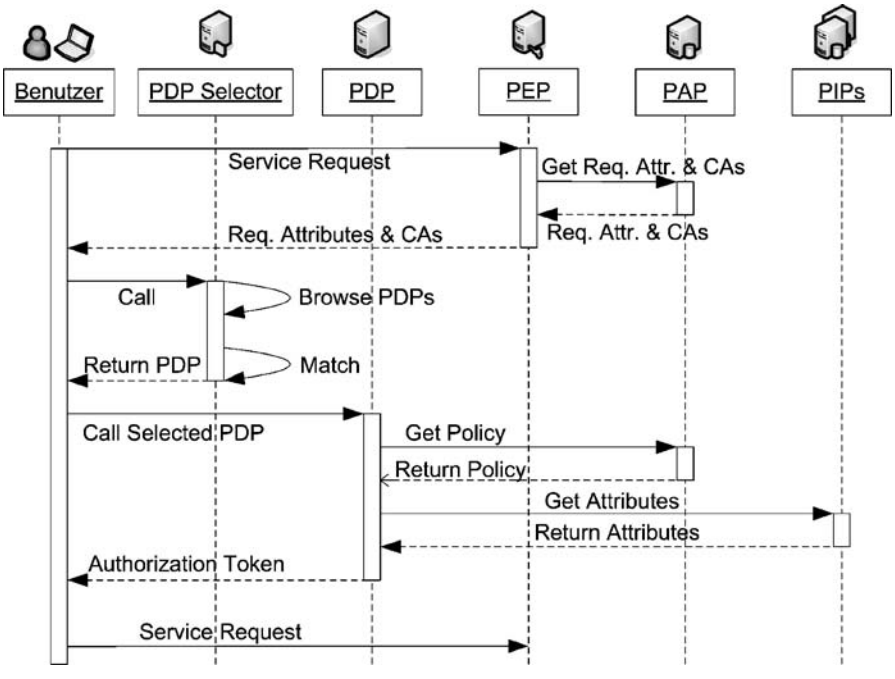

Abb. 3 Sequenzdiagramm des ABAC-Kontrollflusses

Einbindung in Access-eGov

Das europäische Forschungsprojekt Access-eGov hat sich die Interoperabilität von im Internet verteilten eGovernment-Diensten zum Ziel gesetzt. Dies soll technisch durch die Zusammenstellung (Composition) von semantisch annotierten, einzelnen Diensten zu komplexen Dienstketten erreicht werden. Besondere Aufmerksamkeit wird hierbei der Tatsache beigemessen, dass nur eine begrenzte Anzahl an automatisierbar ausführbaren eGovernment-Diensten überhaupt online und öffentlich zur Verfügung steht. Selbst die wenigen vorhandenen eGovernment-Dienste sind nur in den seltensten Fällen semantisch annotiert. Aus diesem Grund ist die gesamte Access-eGov-Plattform auch dazu erdacht worden, zusätzlich zu bald hinzukommenden elektronischen Diensten auch traditionelle (offline) und daher meist formulargebundene Dienste in den Behörden aufzunehmen und in den Lebenssituationen („Life Events“) genannten Dienstgruppen anzubieten und ebenso in Prozessabläufe einplanen zu können.

Eine sorgfältige Erhebung der Anforderungen von Behördenseite hat die Grenzen für die funktionale Spezifikation der für Access-eGov nötigen Komponenten ergeben [15, 16]. Abbildung 4 zeigt einen Überblick über die daraus resultierende serviceorientierte Architektur.

Benutzer, Dienstanbieter der öffentlichen Hand und die Access-eGov-Plattform selbst stellen die Hauptakteure innerhalb der Access-eGov-Architektur dar und werden durch Managementwerkzeuge, generische Prozessmodelle mit den ihnen zugrundliegenden Ontologien sowie durch die semantisch annotierten Dienstbeschreibungen unterstützt. Die Benutzer sind indirekt über ihre webbasierte „Personal Assistant Client“-Software (PAC oder „Persönlicher Assistent“) als Akteure dargestellt, anhand derer annotierte Dienste aufgerufen und der aktuelle Ausführungszustand gespeichert werden können (siehe Abb. 4).

Der Persönliche Assistent ist als eine Art virtueller Softwareagent konzipiert und stellt für den Benutzer die Schnittstelle hin zu den serverseitigen Access-eGov-Komponenten dar. Um Dienstanfragen schneller ausführen zu können, hält der PAC auch die für die unterschiedlichen Dienste benötigten Benutzerattribute in Form von Nutzerprofilen vor (Abb. 5), die um Datenschutzpräferenzen erweiterbar und räumlich über das gesamte Access-eGov-Netzwerk verteilt speicherbar sind. Die dynamischen Komponenten interagieren auf einem Peer-to-Peer-Netzwerk, über das jede Komponente Zugriff auf verteilt gespeicherte Ontologien, Dienstbeschreibungen, Dienstziele (sog. „Goals“) und generische Prozessmodelle hat. Sogenannte „Com-

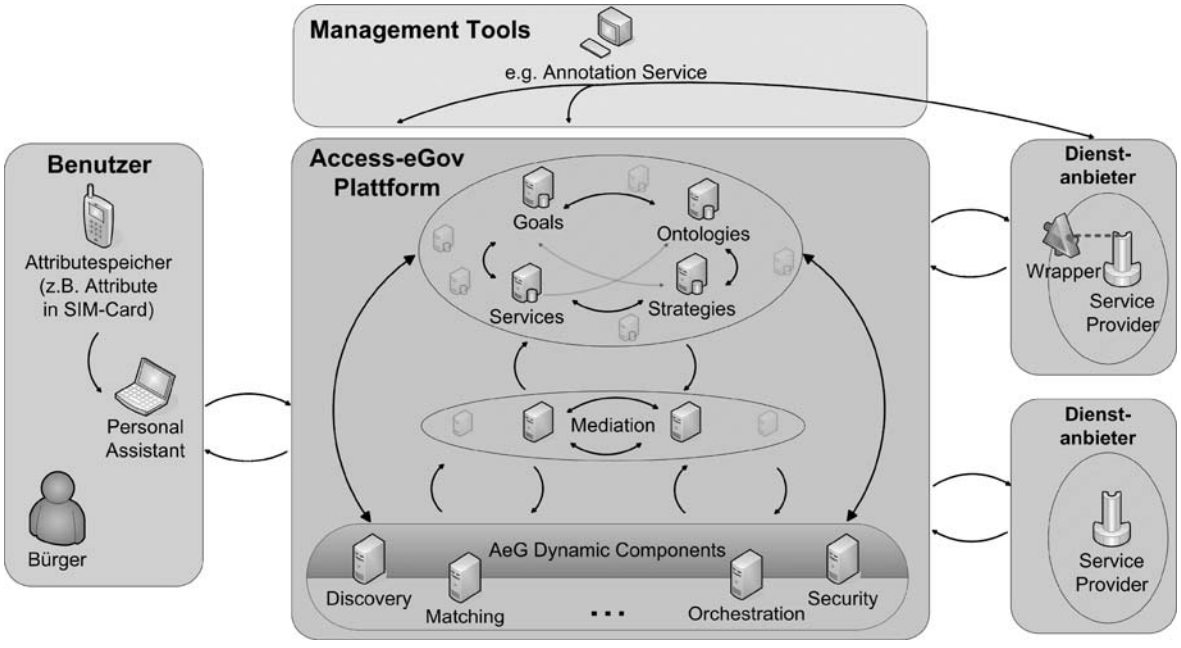

Abb. 4 Systemarchitektur von Access-eGov

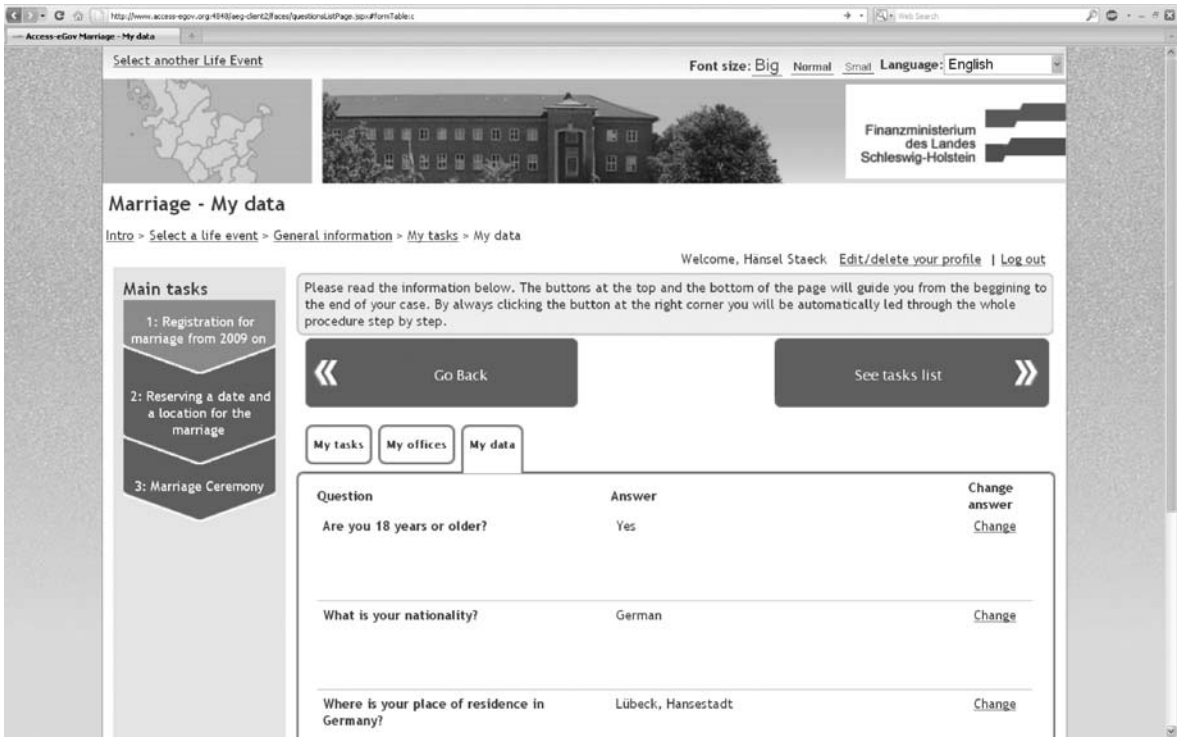

Abb. 5 Attribute im Persönlichen Assistenten von Access-eGov

plex Goals“ stellen den übergeordneten Zweck einer Dienstanfrage dar (z. B. „Heiraten“), die der Nutzer mit Access-eGov ausführen möchte. Die Such- und Abgleichkomponenten (Discovery und Matching) unterteilen dann dieses Goal anhand eines anpassbaren generischen Prozessmodells in einzelne Unteraufgaben, die ihrerseits jeweils einzelnen Behördendiensten nach räumlicher Zuständigkeit

zugeordnet werden. Der PAC erstellt daraufhin eine für den Benutzer speziell angepasste, nach Diensten geordnete Liste, welche die jeweils nötigen Einzelschritte umfasst und bietet die Möglichkeit, elektronisch verfügbare eServices auch direkt aus Access-eGov heraus auszuführen (Orchestration). Diese Dienste werden über Wrapper-Komponenten angesprochen (Abb. 4) und in die für den Benutzer individuell ermittelte Abarbeitungsreihenfolge dynamisch eingebunden.

Sobald die Abfolge von Discovery, Matching und Orchestration beendet wurde und ein einzelner Dienst angefordert wird, tritt der Zugriffskontrollprozess wie im vorigen Abschnitt beschrieben in Kraft. Der PAC kann allgemeine sowie speziell attributbezogene Datenschutzpräferenzen als XACML Assertions speichern. Die serverseitigen Security-Komponenten der Access-eGov-Infrastruktur bieten den Dienstbereitstellern dieselbe Funktionalität an, um die Datenschutzpraktiken und die für den Dienstzugriff geforderten Attribute festzulegen. Im Falle einer erfolgreichen Prüfung wird dem PAC ein Authorization Token (im Sinne eines Tickets) ausgestellt, das dann vom Benutzer genutzt werden kann, um auf den eigentlich gewünschten Dienst (z. B. „Ansuchen um einen Behördentermin zur Eheschließung in Kropp-Stapelholm“) zugreifen zu können. Aus Sicht der Systemintegration besteht der optimale Ansatz darin, das Konzept der Ausstellung der Authorization-Tokens an die Systemarchitektur anzupassen. So kann die Zusammenlegung von Behördendiensten und Access-eGov-Plattform zu einer Single-Sign-On-Domäne den administrativen Aufwand und redundante Datenspeicherung begrenzen helfen. Im Laufe des Projekts hat sich herausgestellt, dass Single-Sign-On-Lösungen aufgrund der rechtlichen und verwaltungstechnischen Gegebenheiten nur durch großen strukturellen Mehraufwand umzusetzen sind. So ist es für öffentliche Verwaltungen rechtlich und administrativ problematisch, die räumliche Aufsicht über den Autorisierungsvorgang zu ihren Bürgerdiensten an Dritte zu delegieren.

Die in dem Projekt entwickelte Sicherheitsinfrastruktur ist aufgrund ihrer Schnittstellen so flexibel aufgebaut, dass die Komponenten auch als netzwerkweit verfügbare Dienste angeboten werden können. Diese werden semantisch mithilfe einer speziell angepassten, sicherheitsbeschreibenden Ontologie annotiert, sodass die daraus resultierenden Dienstprofile zusammen mit den herkömmlichen, funktionalen Behördendiensten in einem Service Repository hinterlegt werden können. Somit können diese Dienste ebenso zur Bildung komplexer Autorisierungsprozesse konkateniert werden, die dann bei der Orchestrierung von organisationsübergreifenden Zugriffskontrollen zur Laufzeit von Dienstanfragen Verwendung finden.

Die physische Umsetzung dieser Sicherheitsinfrastruktur an der Schnittstelle von Access-eGov-Plattform, Benutzern und Dienstanbietern (Abb. 4) baut auf einer Vielzahl von bereits bestehenden Technologien auf. Somit kamen verschiedene Implementierungen von XACML, PKI-Konzepten und semantischen Technologien als Kandidaten für die Realisierung in Betracht. Während bestehende PKI- und zu einem gewissen Grade XACML-Implementierungen durchaus als praxistauglich bezeichnet werden können, zeigten sich vor allem bei den semantischen Technologien noch Defizite in der praktischen Umsetzung. Hierbei ist die rechentechnische Komplexität zur Laufzeit und damit einhergehend der Ressourcenverbrauch zu nennen, den das Arbeiten mit Ontologien mit sich bringt. In verschiedenen Implementierungsansätzen wurde deutlich, dass die sogenannten Repositories, die die semantischen Daten speichern und wieder abfragen sowie die „Reasoner“-Komponenten, die das oben genannte Matching leisten, am ehesten von Laufzeitoptimierungen profitieren können. Ein weiteres Problem in der praktischen Anwendbarkeit der bereits existierenden Implementierungen ist trotz des stabilen Betriebs deren Unausgereiftheit: so sind neuere Versionen in den seltensten Fällen semantisch abwärtskompatibel zur Vorgängerversion und ziehen somit regelmäßige Veränderungen an der Wissensrepräsentation nach sich.

Verwandte Lösungen

Attributbasierte Zugriffskontrollsysteme werden in einer Vielzahl verteilter Anwendungen eingesetzt. So beziehen sich beispielsweise in [24] die Autoren beim Aufbau eines attributzertifikatbasierten ABAC-Systems auf die AKENTI-Engine (http://acs.lbl.gov/Akenti/) bei der Umsetzung in einer Grid-Computing-Umgebung. Der dort vorgestellte Ansatz integriert jedoch keine datenschutzrelevanten Techniken. Auf eher theoretischer Ebene beschreibt [5] ein einheitliches Framework zur Formulierung und Evaluierung von aussagen-

logischen Regeln zur Kontrolle von Dienstzugriffen und zur Offenlegung von Informationen. Obwohl dort eine mächtige Sprache zur Definition von Policies erdacht wird, beschränkt sich das Framework auf theoretische Festlegungen der Zugriffskontrolle und auf Policies zur Offenlegung von Attributinformationen bzw. deren logischen Abgleich. In [22] ergänzen die Autoren ein ABAC-System mit einer Inferenz-Maschine, um die semantische Interoperabilität von Zugriffskontrollrichtlinien sicherzustellen.

Die Definition von Datenschutzpräferenzen war und ist Gegenstand von verschiedenen PET-Initiativen [7, 13]. Darauf aufbauend wurde beispielsweise im Rahmen des PRIME-Projektes (https://www.prime-project.eu/) die Festlegung von sog. Data Handling Policies vorgeschlagen [4]. Diese Richtlinien legen die Art und Weise fest, in welcher der Empfänger personenbezogener Informationen diese zu behandeln und weiter zu verarbeiten hat. Für diese Art von Informationsverarbeitung wurde auch ein Managementmodell zur Verpflichtung informationsverarbeitender Systeme auf datenschutzgerechte Verarbeitung (privacy obligation management model) vorgeschlagen [6], das entsprechende Funktionen bereitstellt, um die Durchsetzung von Datenschutzregeln seitens der Benutzer zu überwachen und Regelverstöße entdecken zu können. In [14] werden XACML-basierte Richtlinien zur Weitergabe von Attributen seitens der Identity Provider benutzt. Ähnlich unserem Ansatz unterstreicht der Autor die Eignung von XACML, um die Richtlinien zur Weitergabe von Attributen zu modellieren. Dort wird jedoch zuallererst auf die kontrollierte Attributausstellung durch einen Identity Provider eingegangen, nicht jedoch auf das ganzheitliche Zusammenspiel der Komponenten in einer Zugriffskontrollarchitektur.

Fazit

Hochgradig verteilte Systeme, wie beispielsweise Systeme, die auf SOA beruhen, müssen in zunehmendem Maße auf Sicherheitsfunktionalitäten vertrauen, welche speziell auf ihr Einsatzumfeld angepasst wurden, um gleichzeitig die Daten der Nutzer sowie die Dienste der Anbieter vor unvorhergesehenem Zugriff schützen zu können. Ein ABAC-basiertes Autorisierungssystem ist in dieser Hinsicht flexibel genug, um auch im Umfeld von SOAs eingesetzt werden zu können. Jedoch setzt der Einsatz voraus, dass die Benutzer (Personen sowie Prozesse) eine Vielzahl an qualifizierbaren Eigenschaften in Form von Attributen offenlegen müssen. Dieser Umstand kann durchaus mit den Datenschutzanforderungen seitens menschlicher Nutzer in Konflikt geraten. Im Rahmen anwendungsbezogener eGovernment-Forschung wurde ein attributbasiertes Zugriffskontrollsystem erarbeitet, das auf physikalisch über ein Netzwerk verteilte redundante PDPs aufbaut. Ausgehend von der Definition individueller Datenschutzpräferenzen in Form von Regeln zur Offenlegung von Attributen erlaubt der vorgestellte Ansatz die dynamische Auswahl derjenigen PDPs, die diesen Präferenzen genügen. Der eigentliche Zugriffskontrollvorgang wird dabei von einer PKI unterstützt, um den Nachweis von Ursprung und Integrität der Vertrauenszertifikate zu gewährleisten. Zur Festlegung, zur Verteilung und zum Abgleich von Datenschutzpräferenzen gegen PDP-Eigenschaften hat sich die OASIS WS-XACML-Spezifikation als tauglich erwiesen. Der beschriebene Ansatz ermöglicht es zusätzlich, den vollständigen Zugriffskontrollprozess dynamisch zur Laufzeit auszuhandeln, ohne tief greifende Änderungen an den zu schützenden Diensten vornehmen zu müssen. Darüber hinaus sind keine statischen Verknüpfungen einzelner PDPs zu einem Dienstanbieter im Vorfeld nötig.

Die in diesem Beitrag vorgestellte verteilte Autorisierungsinfrastruktur konnte im Rahmen des Forschungsprojekts Access-eGov umgesetzt werden. Sie bildet derzeit die technologische Basis für darauf aufbauende Forschungsvorhaben im Umfeld verteilter, dynamischer Zugriffskontrollsysteme.

Danksagung

Die Arbeit, die dieser Veröffentlichung zugrunde liegt, wurde durch Zuwendungen seitens der Europäischen Union ermöglicht. Wir danken für diese Unterstützung sowie unseren Projektpartnern für ihre hilfreichen Kommentare und Diskussionsbeiträge im Rahmen des Projektes Access-eGov (Contract No. FP6-2004-27020).

Literatur

1. Anderson A (2004) The Relationship Between XACML and P3P Privacy Policies. http://research.sun.com/projects/xacml/XACML_P3P_Relationship.html (Abruf August 2009)
2. Anderson A (2006) Web Services Profile of XACML (WS-XACML) Version 1.0. OASIS Working Draft 8
3. Anderson A (2006) Sun Position Paper. W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement

4. Ardagna CA, De Capitani di Vimercati S, Samarati P (2006) Enhancing User Privacy Through Data Handling Policies. Proc. of the 20th Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSec 2006), Sophia Antipolis, Frankreich
5. Bonatti PA, Samarati P (2002) A Uniform Framework for Regulating Service Access and Information Release on the Web. J Comput Secur 10(3):241–271
6. Casassa Mont M (2006) Towards Scalable Management of Privacy Obligations in Enterprises. Proc. of the Third International Conference on Trust, Privacy, and Security in Digital Business (TrustBus '06), Krakau, Polen, pp 1–10
7. Cranor L et al. (2006) The Platform for Privacy Preferences 1.1 (P3P1.1) Specification. W3C Working Group Note
8. Cranor L, Langheinrich M, Marchiori M (2002) A P3P Preference Exchange Language 1.0 (APPEL 1.0). World Wide Web Consortium Working Draft
9. Dürbeck S, Schillinger R, Kolter J (2007) Security Requirements for a Semantic Service-oriented Architecture. Proc. of the 2nd International Conference on Availability, Reliability and Security (ARES '07), Wien, Österreich
10. Earp JB, Baumer D (2003) Innovative Web Use to Learn About Consumer Behavior and Online Privacy. Commun ACM 46(4):81–83
11. European Union (1995) Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Official Journal of the European Communities No L 281/31, October 1995
12. European Union (2006) Directive 2006/123/EC of the European Parliament and of the Council of 12 December 2006 on services in the internal market. Official Journal of the European Union No L 376/36, December 2006
13. Hansen M, Krasemann H (2005) Privacy and Identity Management for Europe – PRIME White Paper. PRIME deliverable D15.1.d, http://www.prime-project.eu.org/whitepaper/ (Abruf September 2005)
14. Hommel W (2005) Using XACML for Privacy Control in SAML-Based Identity Federations. IFIP International Federation for Information Processing CMS 2005 LNCS 3677, pp 160–169
15. Klischewski R, Ukena S, Wozniak D (2006) User Requirements Analysis & Development/Test Recommendation. Access-eGov deliverable D2.2, http://www.accessegov.org/ (Abruf September 2009)
16. Kolter J, Schillinger R, Pernul G (2007) Building a Distributed Semantic-aware Security Architecture. Proc. 22nd Int. Information Security Conference (SEC2007), Sandton, South Africa
17. Kolter J, Schillinger R, Pernul G (2007) A Privacy-enhanced Attribute-based Access Control System. 2007. Proc. of the 21st Annual IFIP WG 11.3 Working Conference on Data and Application Security (DBSec 2007), Redondo Beach, CA, USA
18. Lopez J, Oppliger R, Pernul G (2004) Authentication and Authorization Infrastructures (AAIs): A Comparative Survey. Comput Secur 23(7):578–590
19. MacKenzie CM, Laskey K, McCabe F, Brown PF, Metz R (2006) Reference Model for Service Oriented Architecture 1.0. OASIS Standard
20. Moses T (2005) eXtensible Access Control Markup Language (XACML) Version 2.0. OASIS Standard
21. Nadalin A et al. (2006) Web Services Security: SOAP Message Security 1.1, http://docs.oasis-open.org/wss/v1.1/ (Abruf Januar 2010)
22. Priebe T, Dobmeier W, Kamprath N (2006) Supporting Attribute-based Access Control with Ontologies. Proc. of the 1st International Conference on Availability, Reliability and Security (ARES '06), pp 465–472. Los Alamitos, CA, USA, IEEE Computer Society
23. Priebe T, Dobmeier W, Muschall B, Pernul G (2005) ABAC – Ein Referenzmodell für attributbasierte Zugriffskontrolle. Jahrestagung Fachbereich Sicherheit der Gesellschaft für Informatik (Sicherheit '05), S 285–296, Universität Regensburg, Deutschland
24. Thompson M, Johnston W, Mudumbai S, Hoo G, Jackson K, Essiari A (1999) Certificate-based Access Control for Widely Distributed Resources. Proc. of the 8th USENIX Security Symposium, Washington, DC, USA
25. Tomasek M, Paralic M et al. (2006) Access-eGov Components Functional Descriptions. Access-eGov deliverable D3.2, http://www.accessegov.org/ (Abruf Dezember 2009)
26. World Wide Web Consortium (2006) Web Services Policy 1.2 – Framework (WS-Policy), http://www.w3.org/Submission/WS-Policy/ (Abruf Januar 2010)
27. Yuan E, Tong J (2005) Attributed Based Access Control (ABAC) for Web Services. Proc. of the IEEE International Conference on Web Services (ICWS'05), pp 561–569, Washington, DC, USA, IEEE Computer Society