Die elektronische Steuererklärung (ELSTER) - Status und Ausblick - OpenElster

Christine Randlkofer

ELSTER
Bayerisches Landesamt für Steuern
Meiserstr. 8 80333 Muenchen
Christine.Randlkofer@lfst.bayern.de
emailadresse@autor2

Abstract: ELSTER ist als „Elektronische Steuererklärung“ bereits bundesweit bekannt. Nahezu alle steuerlich relevanten Daten können von Bürgern und Unternehmen elektronisch abgegeben werden. Mehrere Millionen Datensätze werden inzwischen jährlich über ELSTER übermittelt.

Um hohe Datensicherheit zu gewährleisten, hat ELSTER bereits Anfang 2006 eine PKI-basierte elektronische Authentifizierung und Verschlüsselung eingeführt. Jeder ELSTER-Anwender kann über vom ELSTER-Trustcenter ausgegebene elektronische Zertifikate authentisiert werden, sensible Daten können auf diese Weise verschlüsselt werden. Besitzt der Anwender bereits eine Signaturkarte, so kann auch diese verwendet werden.

Derzeit prüft die Steuerverwaltung, die von ELSTER ausgegebenen Zertifikate und die ELSTER-Sicherheitsinfrastruktur auch anderen Behörden zur Nutzung in ihren eGovernment-Anwendungen zur Verfügung zu stellen. Diesen soll es damit ermöglicht werden ebenfalls starke Anwender-Authentisierung und elektronische Signatur zu integrieren, ohne eigene Zertifikate auszugeben und eigene Verfahren zur Anwendung dieser Zertifikate entwickeln zu müssen. Dieses Verfahren trägt den Arbeitstitel „OpenElster“.

1 Mit ELSTER gewinnen alle

Die Verwendung von Zertifikaten und Chipkarte als Mittel zur Authentisierung von Nutzern gegenüber Dienstanbietern ist gemeinhin als sicheres Verfahren anerkannt.

Viele eGovernment-Projekte leiden unter der mangelnden Verbreitung qualifizierter elektronischer Signaturen. Mögliche Nutzer besitzen keine Signaturkarte und sind wegen der hohen Kosten, dem schwierigen Prozedere bei der Beschaffung und den oftmals nicht ersichtlichen Vorteilen nicht zur Anschaffung entsprechender Karten bereit.

Unabhängig davon wird von den Behörden oftmals eine qualifizierte elektronische Signatur gefordert, obwohl lediglich eine gesicherte Authentifizierung des Anwenders notwendig wäre. Zudem ist auch bei Verwendung einer Signaturkarte immer noch eine weitere Registrierung (Mapping der Zertifikatsdaten auf die Ordnungsbegriffe der Behörde) notwendig.

ELSTER bietet derzeit die Möglichkeit, dass sich Bürger und Unternehmen über ein Web-Portal bei der Steuerverwaltung direkt registrieren. Je nach gewünschtem Funktionsumfang können die Anwender hierzu eine Signaturkarte, einen USB-Stick mit Signaturfunktion oder einen von der Steuerverwaltung kostenlos zur Verfügung gestellten Soft-Token (ein asymmetrisches Schlüsselpaar auf Softwarebasis mit zugehörigem Zertifikat) benutzen, um den kompletten Datenverkehr mit dem Finanzamt über eine eindeutige Authentifizierung abzuwickeln. Die Steuerverwaltung hat hierzu eine eigene PKI-Infrastruktur aufgebaut, die kostenlos für jeden Bürger Zertifikate herausgibt.

Diese Zertifikate und die Technologie zur Anwendung dieser Zertifikate für Authentisierung und elektronischer Signatur könnten auch für andere von Behörden betriebene eGovernment-Anwendungen eingesetzt werden.

Für eine eGovernment-Anwendung würden sich damit folgende Vorteile ergeben:

- Die Anwendung könnte auf eine hohe Anzahl von bereits ausgegebenen Zertifikaten zurückgreifen (ca. 350.000 Zertifikate bei einem Zuwachs von ca. 30.000 Zertifikaten/Monat, Stand März 2007)
- Anwender die noch nicht über ein ELSTER-Zertifikat verfügen, könnten sich über das ELSTER-Web-Portal kostenlos, einfach und schnell ein Zertifikat auf Softwarebasis besorgen. Für höhere Sicherheit könnte über einen Online-Shop ein USB-Stick mit Signaturfunktion bezogen werden, der die gleiche Sicherheit wie eine

Signaturkarte bietet. Das zur Anwendung nötige Zertifikat wird kostenlos durch ELSTER ausgegeben.

- Zur Anwendung von zertifikatsbasierten Sicherheitsverfahren ist auf Anwender- und Serverseite Software nötig, welche ein Authentisierungsprotokoll (Challenge / Response) durchführt und elektronische (nicht qualifizierte) Signaturen erzeugt bzw. prüft. Die auf Clientseite hierzu nötige Software (Java-Klassen) würde von der Steuerverwaltung zur Verfügung gestellt werden. Auf Serverseite könnte die eGovernment-Anwendung über Web-Services die Authentisierung prüfen und die Verifikation von signierten Daten vornehmen. Eigene IT-Infrastruktur (Hardware, Software, Verzeichnisdienst) und Betriebspersonal wäre damit nicht nötig.
- Es wäre auch möglich, dass die eGovernment-Anwendung auf bereits bei ELSTER verwendete Signaturkarten zurückgreifen könnte. Die von der Steuerverwaltung zur Authentisierung/Signatur-Prüfung bereitgestellten Web-Services würden in diesem Fall automatisch das Trustcenter des Signaturkartenherausgebers abfragen. Die eGovernment-Anwendung müsste sich nicht um die komplexe Einbindung unterschiedlichster Signaturkarten kümmern. Die Signaturkarten aller namhaften deutschen Trustcenter werden bereits jetzt unterstützt.

2 Modular und offen – ein attraktiver Weg

Die ELSTER-Technologie ist modular aufgebaut und setzt ausschließlich auf Standardtechnologien. Komplexe Kommunikationsprotokolle und aufwendige Applikationspakete sind nicht notwendig.

Der mögliche OpenElster-Authentifizierungsclient könnte auf allen Java-fähigen Clientsystemen arbeiten. Er würde alle Details des Umgangs mit Schlüsseln und Zertifikaten kapseln.

Die zum Betrieb einer PKI nötigen Verfahren (Registrierung von Anwendern, Ausgabe/Sperrung von Zertifikaten, Zertifikatserneuerung und Betreiben von Verzeichnisdiensten) würden von der Steuerverwaltung vorgenommen werden. Für die serverseitig nötige Prüfung der Authentisierung/Signatur würden Web-Services zur Verfügung stehen, die auf Servern der Steuerverwaltung gehostet würden.

Die OpenElster-Software selbst kodiert keinerlei Wissen oder auch nur Annahmen über konkrete eGovernment-Prozesse und Anwendungen, sondern liefert „nur“ Security-Kernfunktionalität.

Hierdurch werden Freiräume für die Implementierung der eigentlichen Anwendung geschaffen, z.B. eines Web-Portals. Dies macht in der Regel ein (internes) Systemhaus. Als Integrator kennt dieser die eGovernment-Prozesse im Detail, ist aber nicht unbedingt

Experte für automatisierte IT-Sicherheit auf Basis von PKI. Mittels der OpenElster-Software kann der Integrator aber die notwendigen Sicherheitsfunktionen und -prozesse in die jeweilige eGovernment-Anwendung flexibel und transparent und damit für den Endanwender weitgehend unsichtbar integrieren.

Im folgendem werden ein Anwendungsbeispiel und die Funktionalität von OpenElster näher beschrieben.

3 Anwendungsbeispiel - Web-Portal

Der Bürger erwartet von einer dienstleistungsorientierten Behörde einen einfachen und plattformunabhängigen Zugang zu dem Leistungsangebot – rund um die Uhr, personalisiert und ohne aufwendige Softwareinstallation beziehungsweise Updates. Dies kann über ein Web-Portal realisiert werden.

Dieses Web-Portal könnte von der Behörde bzw. von einem von ihr beauftragten Dienstleister unter Einbindung des OpenElster-Authentifizierungsclients (ESiCl1) bzw. durch Aufruf von Web-Services des OpenElster Authentifizierungsdienstes (WIESEL) realisiert werden (siehe Bild).

[1] ESiCl: ELSTER Signature Client Library

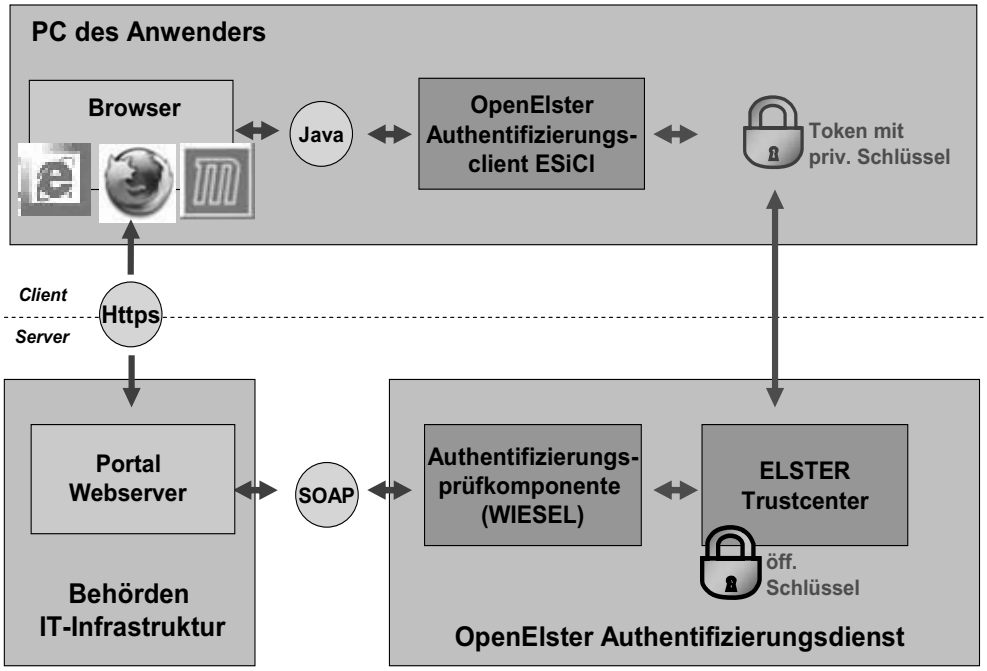

Bild: Beispielrealisierung eines Portals mit OpenElster

Die OpenElster-Sicherheitsplattform besteht aus drei Komponenten, die in ihrem Zusammenspiel höchste Sicherheit und Integrationsfähigkeit gewährleisten:

- Der Authentifizierungsclient bietet elektronische Signatur und Authentisierung über ein Java-Applet, welches den plattform-unabhängigen Einsatz über jeden Web-Browser ermöglicht. Er übernimmt die Erzeugung der Schlüssel auf Anwenderseite (wird über das ElsterOnline-Portal vorgenommen), die Authentifizierung für die Systemanmeldung durch Abfrage des serverseitigen Authentifizierungsdienstes und die Signaturerzeugung. Aktuell werden softwarebasierte Schlüssel und alle gängigen Smartcards namhafter Trustcenter unterstützt. Die Einbindung weiterer Karten ist möglich.
- Die serverseitig eingesetzte Authentifizierungskomponente (WIESEL) stellt zentrale Sicherheitsdienste zur Verfügung, um die Identität und elektronische Signaturen der Anwender überprüfen zu können. Diese Komponente ist ebenfalls in Java realisiert

und über Web Services / SOAP flexibel ansprechbar. Im Rahmen von ELSTER übernimmt sie die Steuerung von Registrierung und Anmeldevorgang, das Verwalten (Anlegen, Freischalten, Sperren) eines anwenderspezifischen Accounts, die automatische Erneuerung abgelaufener ELSTER-Zertifikate vor Ablauf der Gültigkeitsdauer und das Logging aller sicherheitsrelevanten Vorgänge.

Für die Behörden-Anwendung würde sie die Authentisierung und Signaturprüfung übernehmen.
Hohe Sicherheit und Performance wird durch eine integrierte HSM-Box gewährleistet.

Das ELSTER-Trustcenter ermöglicht im Rahmen von ELSTER die Schlüssel- und Zertifikatsvergabe auf Basis von Software-Schlüsseln oder USB-Signatursticks. Diese Funktionalität wird über das ElsterOnline-Portal zur Verfügung gestellt.

PKI-basierte individuelle Verschlüsselung z.B für elektronische Formulare kommt zwar bei ELSTER zum Einsatz, kann aber anderen Behörden aus rechtlichen Gründen nicht als Web-Service zur Verfügung gestellt werden. Bei Erzeugung von Schlüsseln und Zertifikaten durch ELSTER werden jedoch auch ein Verschlüsselungsschlüssel und ein zugehöriges Zertifikat erzeugt, welches von der Behörden-Anwendung selbst zur Nutzer-individuellen Verschlüsselung verwendet werden kann. Der OpenElster-Authentifizierungsclient enthält die nötigen Funktionen zur Ver- bzw. Entschlüsselung.

4 Standards

OpenElster unterstützt soweit relevant alle international gängigen Standards:

- **Java**
  Sowohl die clientseitige als auch die serverseitigen Komponenten sind in Java programmiert. Dies ermöglicht betriebssystemübergreifenden Einsatz und Ablauffähigkeit in allen gängigen Browsern. Die Client-Komponente ist zudem bzgl. minimalen Codeumfang optimiert, um kurze Ladezeiten im Web-Browser gewährleisten zu können.

- **Web Services/SOAP**
  Die serverseitigen Komponenten sind über Web Services ansprechbar. Die Entwickler der eGovernment-Anwendung, die diese Web-Services verwenden würden daher kein Spezialwissen benötigen.

- **X.509**
  Die von ELSTER erstellten Zertifikate und Sperrlisten entsprechen X.509 v3. Damit ist Interoperabilität mit nahezu allen Trustcentern und den meisten Signaturanwendungen sichergestellt.

- **pkcs#11**
  Der Zugriff zu hardwarebasierten Kryptografie-Token (z.B. Smartcards) erfolgt über pkcs#11. Da nahezu jeder Kartenherausgeber entsprechende Treiber mitliefert, kann eine Vielzahl von Karten unterstützt werden. Die Integration neuer Karten ist mit überschaubarem Aufwand durchführbar.

- **pkcs#12**
  Dieser „Personal Information Exchange Syntax Standard" definiert ein Dateiformat, das dazu benutzt wird, private Schlüssel mit dem zugehörigen ELSTER-Zertifikat passwortgeschützt zu speichern. Da viele Signatur- und Verschlüsselungsanwendungen (z.B. Web-Browser, Mail-Clients) dieses Format verwenden, ist es möglich durch ELSTER erzeugte Software-Schlüssel auch in diesen Anwendungen zu importieren und zu verwenden.

- **XML Dsig**
  Die Signatur von XML-Daten erfolgt nach diesem Standard. Es werden XML-Signaturen vom Typ Enveloping Signatur unterstützt, d.h. der Signatur Tag befindet sich innerhalb des signierten Dokuments. Dabei ist nicht das gesamte Dokument signiert, sondern lediglich die „Nutzdaten". Das Signaturzertifikat, ggf. auch weitere Attributzertifikate und Ausstellerzertifikate, werden dabei im Abschnitt KeyInfo abgelegt.

- **eCard- API**
  ELSTER plant die Unterstützung der eCard-API. Diese soll auf lange Sicht bei der Unterstützung von Karten statt pkcs#11 zum Einsatz kommen.

5 Stabilität / Performance

Alle beschriebenen Komponenten sind seit Ende 2005 im Echteinsatz mit hohen Anforderungen an Performance und Stabilität. Sie sind ausgelegt und getestet auf bis zu 30 Mio. Zertifikate bei 120.000 Signaturprüfungen / Stunde.
Die Prüfung einer 4 kB großen Datei dauert durchschnittlich 1,1 sec bei ca. 30

Anfragen/sec. In der Prüfung enthalten sind: XML-Datei einlesen, parsen, formal prüfen (gültiges Dokument, Signatur an richtiger Stelle,...), Zertifikat aus LDAP holen, kompletten Zertifizierungspfad (2-stufig) prüfen, inkl. Prüfung gegenüber Sperrliste, Prüfung der Signatur der Sperrliste und XML-Signatur prüfen.