

Polizon – WS-Policy Editor

IT Security / Betreuer: Prof. Dr. Eric Dubuis
 Experte: Andreas Dürsteler

Web Services wie SOAP verwenden Beschreibungssprachen um ihre Dienste und Schnittstellen zu spezifizieren. Ein mögliches Element dieser Beschreibung ist eine WS-Policy (Web Services Policy).

Eine Policy enthält diverse Regeln, wie Client und Server sich zu verhalten haben. Eine Policy kann ein sehr komplexes Regelwerk darstellen: Ein logisches Konstrukt von Mindestanforderungen, optionalen und alternativen Regeln.

Ziel ist ein grafischer Editor, der den Anwender bei der Erstellung einer Policy aktiv unterstützt.

Die Policy

Policies werden in XML modelliert. Die einzelnen Regeln, Assertions genannt, werden dabei über logische Operatoren verknüpft. So lässt sich ein komplexes Regelwerk erstellen: «Ich will eine synchrone Verschlüsselung oder eine asynchrone, aber dann mit einem Token. In beiden Fällen will ich aber eine Signatur.»

Policies lassen sich in Policies verschachteln, lassen sich referenzieren und eine Assertion kann selbst eine Policy enthalten um die eigenen Parameter als Regelwerk abbilden zu können. Client und Server handeln dann eine definitive Konfiguration aus, die keine der Regeln verletzt.

Der Editor

Das Ziel ist ein grafischer Editor, der eine einfache Bearbeitung einer Policy zulässt. Der Editor selbst benötigt eine gewisse Intelligenz:

- Er muss die Policy und ihre Assertions validieren können. Jede Assertion hat ihre eigene Spezifikation.
- Er muss dem Benutzer eine aktive Hilfestellung bieten. Dazu gehören Vorschläge, was an einer bestimmten Stelle alles eingefügt werden darf, Beschreibungen zu den Assertions und Web-Links zu deren Spezifikationen.

Die Spezifikationen der Assertions sind sehr offen. Neue Standards

können nicht nur neue Assertions definieren, sondern auch bestehende erweitern. Durch ein Plug-in System soll sich der Editor erweitern lassen, um neue Assertions «kennen zu lernen».

Das Plug-in System

Ein Plug-in kann dem Editor beliebig viele Assertions bekannt machen. Der Editor kann gleichzeitig mehrere Plug-ins geladen haben. Um den offenen Standards gerecht zu werden, kann ein Plug-in Teile eines anderen Plug-ins übersteuern. Ein Plug-in kann auch einfach nur neue Assertions definieren.

Das Plug-in im Plug-in

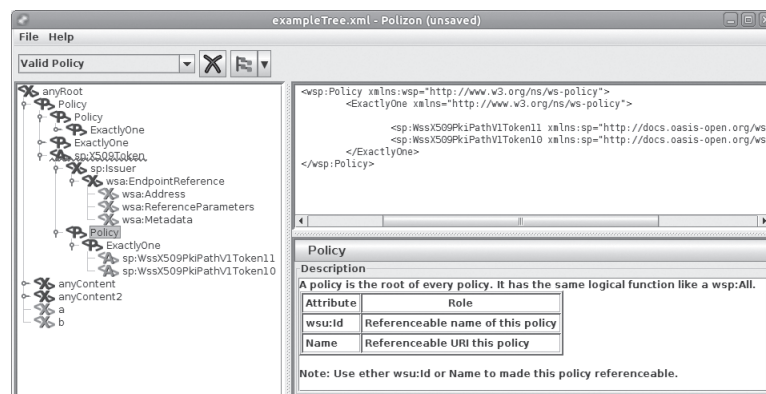
Eigentlich müsste für jedes neue Set an Assertions ein neues Plug-in in Java, der Entwicklungssprache des Editors, programmiert werden. Diesem Umstand wird entgegen gewirkt, indem der Editor bereits mit einem flexiblen Plug-in ausge-

liefert wird. In diesem sind die unterstützten Assertions nicht hart kodiert; das Plug-in liest die Assertions aus XML-Dateien ein und erzeugt anhand derer die Assertions für den Editor.

Das Plug-in hat seine eigene Spezifikation, wie eine Assertion beschrieben werden muss, um dann geparsed und als Assertion in den Editor geladen zu werden. Diese ist so gehalten, dass es sehr einfach ist eine neue Assertion zu definieren. So können neue Assertions als XML Plug-in erstellt werden. Die Java Plug-in Schnittstelle ist somit nur noch nötig, wenn ein Standard eine Assertion definieren würde, die zu komplex wäre, um dass sie sich im XML Plug-in eigenen Dialekt beschreiben lassen würde.



Adrian Fahrer



Polizon während der Bearbeitung einer Policy