

Web Services

Bartłomiej Świercz

Katedra Mikroelektroniki i Technik Informatycznych

Łódź, 2 grudnia 2005 roku

Oprogramowanie napisane w różnych językach i uruchomione na różnych platformach może wykorzystać Web Services do wymiany danych za pomocą sieci komputerowej w sposób przypominający standardową komunikację pomiędzy procesami działającymi na jednym komputerze. W przypadku użycia otwartych standardów możliwa jest współpraca programów napisanych w wielu językach (np. Java, Python) i różnych platform systemowych (np. MS Windows i Linux).

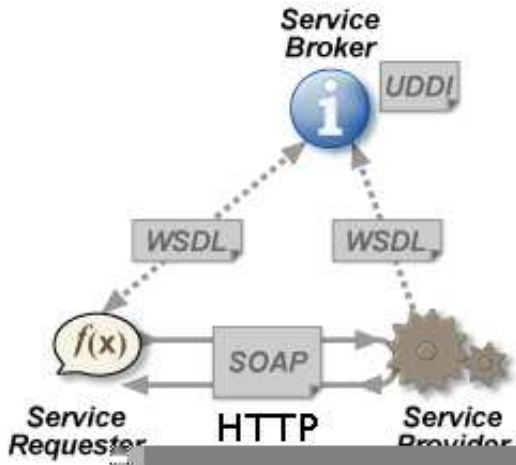
Istnieją dwie organizacje (W3C i OASIS) odpowiedzialne za standaryzację i rozwój architektury systemów nazywanych Web Services.

Powołana została również specjalna organizacja WS-I zajmująca się promocją technologii Web Services.

Definicja W3C (<http://www.w3.org/TR/ws-arch/>):

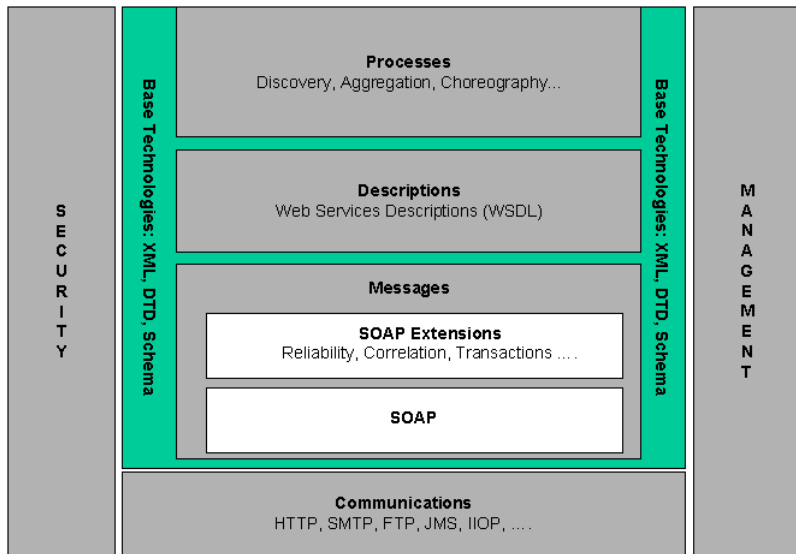
*A Web service is a **software system** designed to support interoperable **machine-to-machine interaction over a network**. It has an **interface** described in a machine-processable format (specifically **WSDL**). Other systems interact with the Web service in a manner prescribed by its description using **SOAP messages**, typically conveyed using **HTTP** with an **XML serialization** in conjunction with other Web-related standards.*

Architektura Web Services



Rysunek: Autor: H. Voormann

Architektura stosowa (Web Services Protocol Stack)



Rysunek: Źródło: W3C

Języki znaczników używane w systemach Web Services:

- BEEP - Blocks Extensible Exchange Protocol
- BPEL - Business Process Execution Language
- E-Business XML
- SOAP - Simple Object Access Protocol
- UDDI - Universal Description, Discovery, and Integration
- WSDL - Web Services Description Language
- WSFL - Web Services Flow Language
- WSCL - Web Services Conversation Language
- XML-RPC - XML Remote Procedure Call

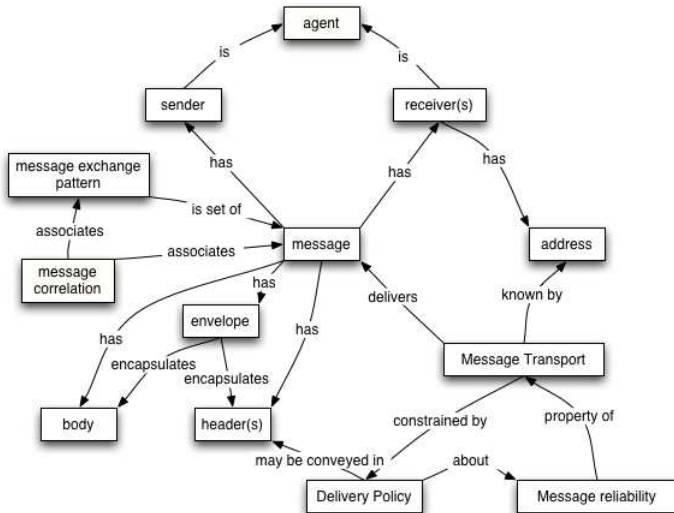
Na obecny kształt systemów Web Services mają wpływ poniższe standardy:

- SOAP 1.1
- WSDL 1.1
- UDDI 2.0
- XML 1.0 (Second Edition)
- XML Schema Part 1: Structures
- XML Schema Part 2: Datatypes

Na obecny kształt systemów Web Services mają wpływ poniższe standardy:

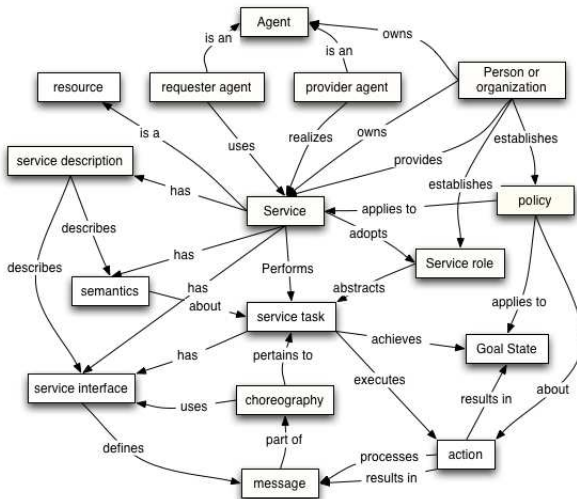
- RFC2246: The Transport Layer Security Protocol version 1.0
- RFC2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile
- RFC2616: HyperText Transfer Protocol 1.1
- RFC2818: HTTP over TLS
- RFC2965: HTTP State Management Mechanism
- The Secure Sockets Layer Protocol version 3.0

Komunikacja zorientowana na wiadomości



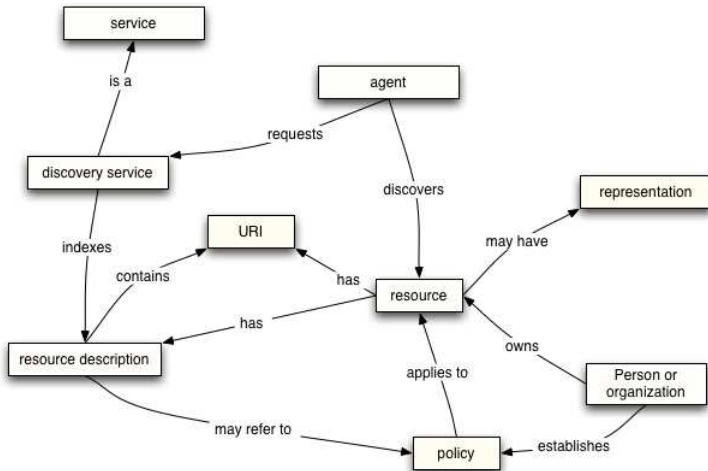
Rysunek: Źródło: W3C

Model aplikacji zorientowanej na serwisy



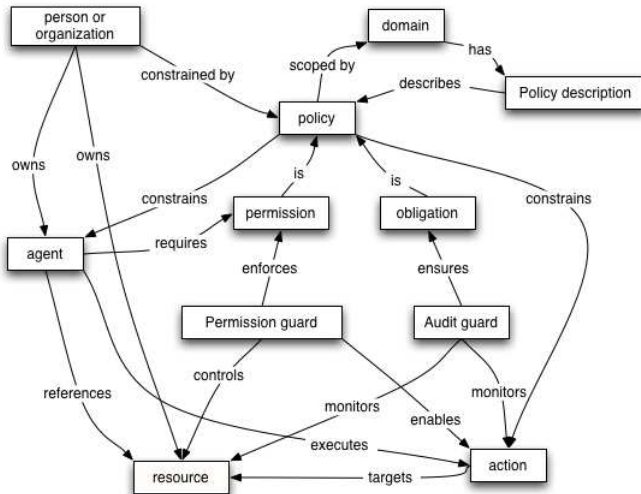
Rysunek: Źródło: W3C

Model aplikacji udostępniającej zasoby



Rysunek: Źródło: W3C

Model polityki bezpieczeństwa



Rysunek: Źródło: W3C

- Web Services pozwalają na współdzielenie programów napisanych w różnych językach i działających na różnych platformach.
- Web Services używają otwartych standardów i protokołów bazujących na tekście czytelnym dla człowieka.
- Poprzez wykorzystanie protokołu HTTP jako kanału transportowego Web Services pozbywają się problemu obecności firewall w sieci.
- Web Services pozwalają na łączenie zasobów wielu instytucji i firm.

- Ciągle brakuje standardów dla bardziej zaawansowanych technik (np. transakcje) realizowanych w oparciu o Web Services.
- Aplikacje oparte o Web Services mają słabą wydajność w porównaniu do aplikacji zbudowanych w oparciu o inne technologie rozproszone (DCOM, CORBA, RMI).
- Poprzez wykorzystanie protokołu HTTP i obejściu większości zapór sieciowych pojawiają się problemy z właściwym zabezpieczeniem aplikacji.

Powody dla których warto użyć Web Services

Podstawowym powodem dla którego warto użyć technologii Web Services jest naturalny podział projektu na moduły, co ułatwia późniejszy rozwój aplikacji.