

# Podstawy implementacji usług sieciowych OGC

Tomasz Kubik

Instytut Informatyki, Automatyki i Robotyki  
Politechnika Wrocławska



URZĘDOWA GEOINFORMACJA W INTERNECIE – MOŻLIWOŚCI I OCZEKIWANIA  
Wisła, 5-7.09.2007

# Usługi sieciowe

- Usługa sieciowa to kawałek logiki biznesowej, umieszczony gdzieś w Internecie, do którego istnieje dostęp poprzez standardowe protokoły internetowe jak **HTTP** lub **SMTP**, [1].
- Usługi sieciowe to po prostu aplikacje posiadające API dostępny z sieci.
- Interfejs usługi z założenia ukrywa szczegóły jej implementacji. Stąd usługi można tworzyć w dowolnym języku programowania i dla dowolnej platformy systemowej pod warunkiem, że widoczność i dostęp do tych usług będzie spełniać przyjęte dla usług sieciowych standardy. W ten sposób zapewnia się interoperacyjność.
- Cechy usługi sieciowej
  - luźno powiązane
  - gruboziarniste
  - zdolne do działania w trybie synchronicznym (zdalne wywoływanie procedur) i asynchronicznym (wymiana dokumentów)

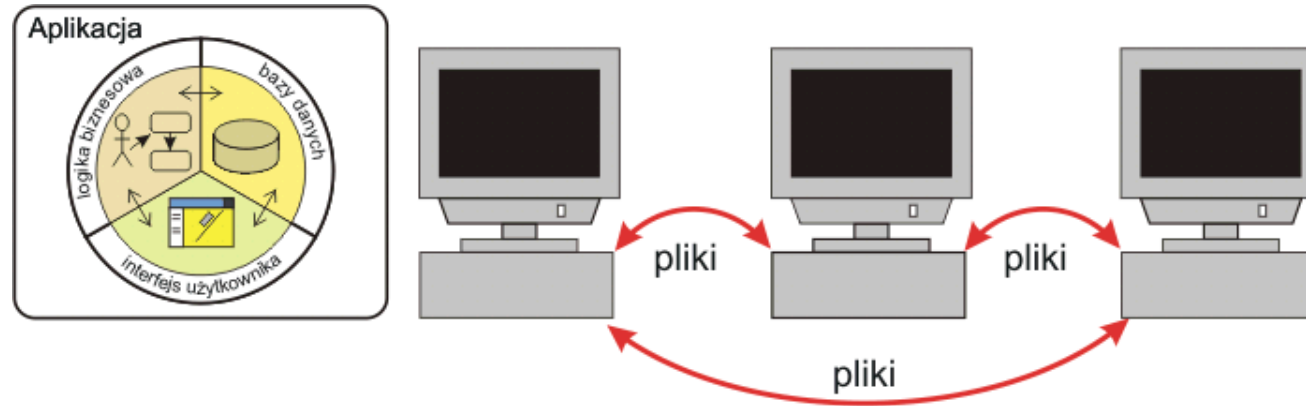
1. David Chappell, Tyler Jewell: “Java Web Services”, O'Reilly, First Edition, March 2002

## Usługi sieciowe bazujące na XML

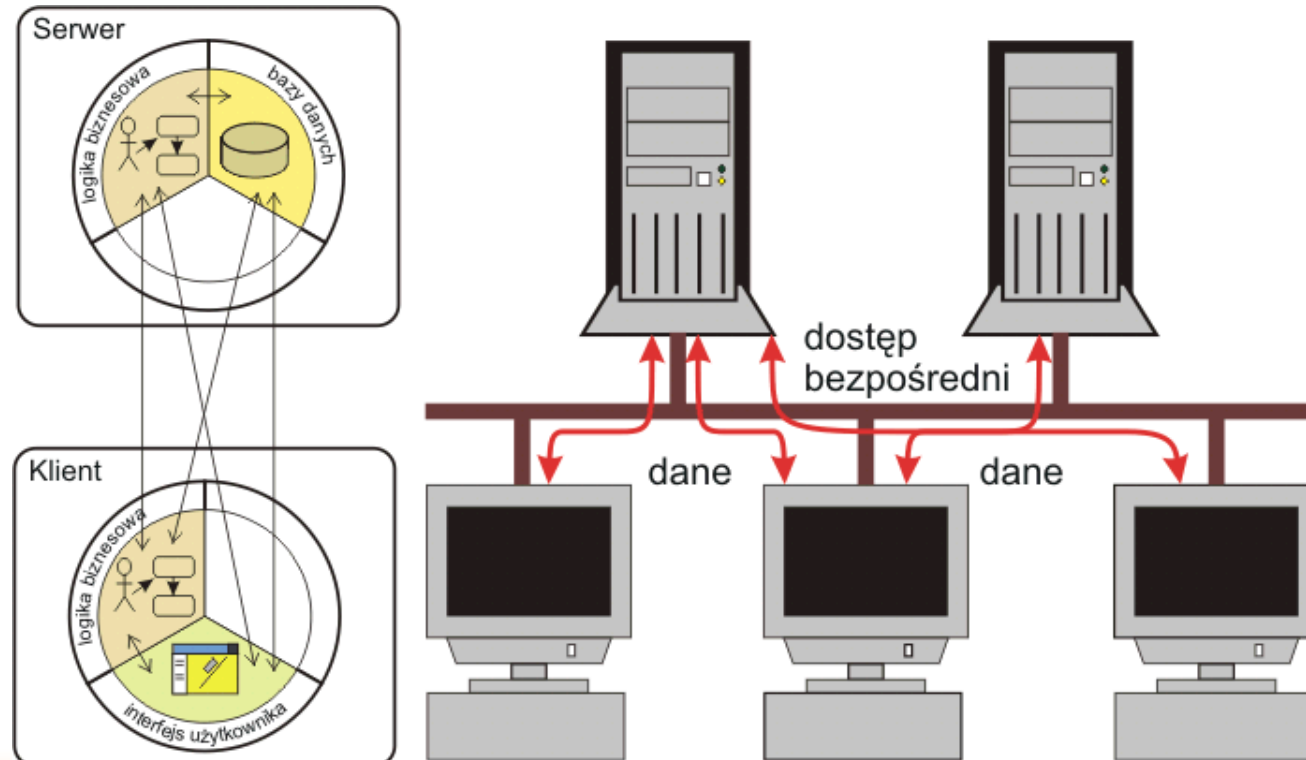
- Jeśli standardem opisu usługi sieciowej jest XML, to taka usługa nazywana jest **usługą sieciąową bazującą na XML**. Opis usługi w takim przypadku jest dokumentem XML, zawierającym wszystkie potrzebne do interakcji z usługą informacje, włącznie z opisem formatu komunikatów (dzięki którym są wykonywane operacje), protokołów transportowych i lokalizacji.

# Architektury rozwiązań

desktop

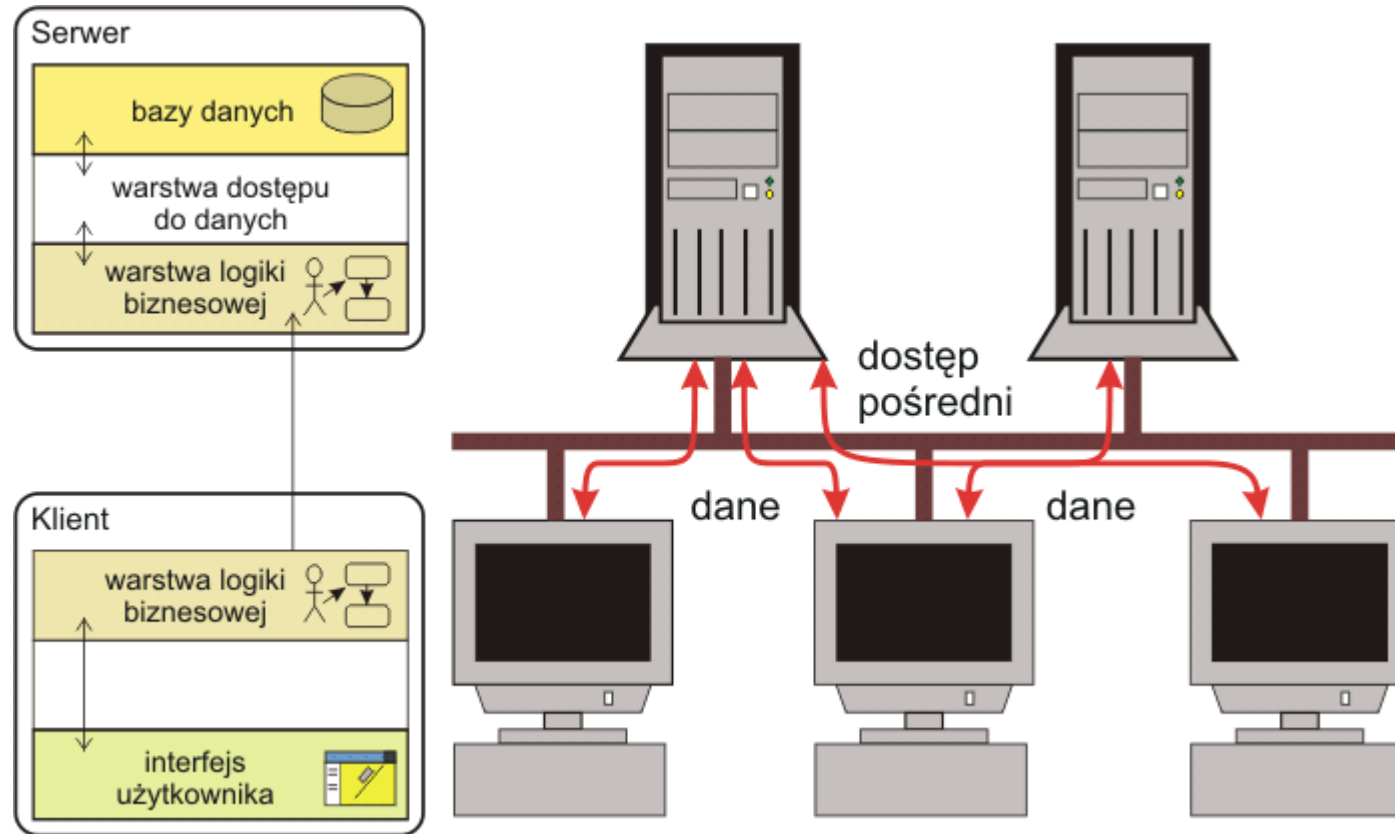


klient-serwer



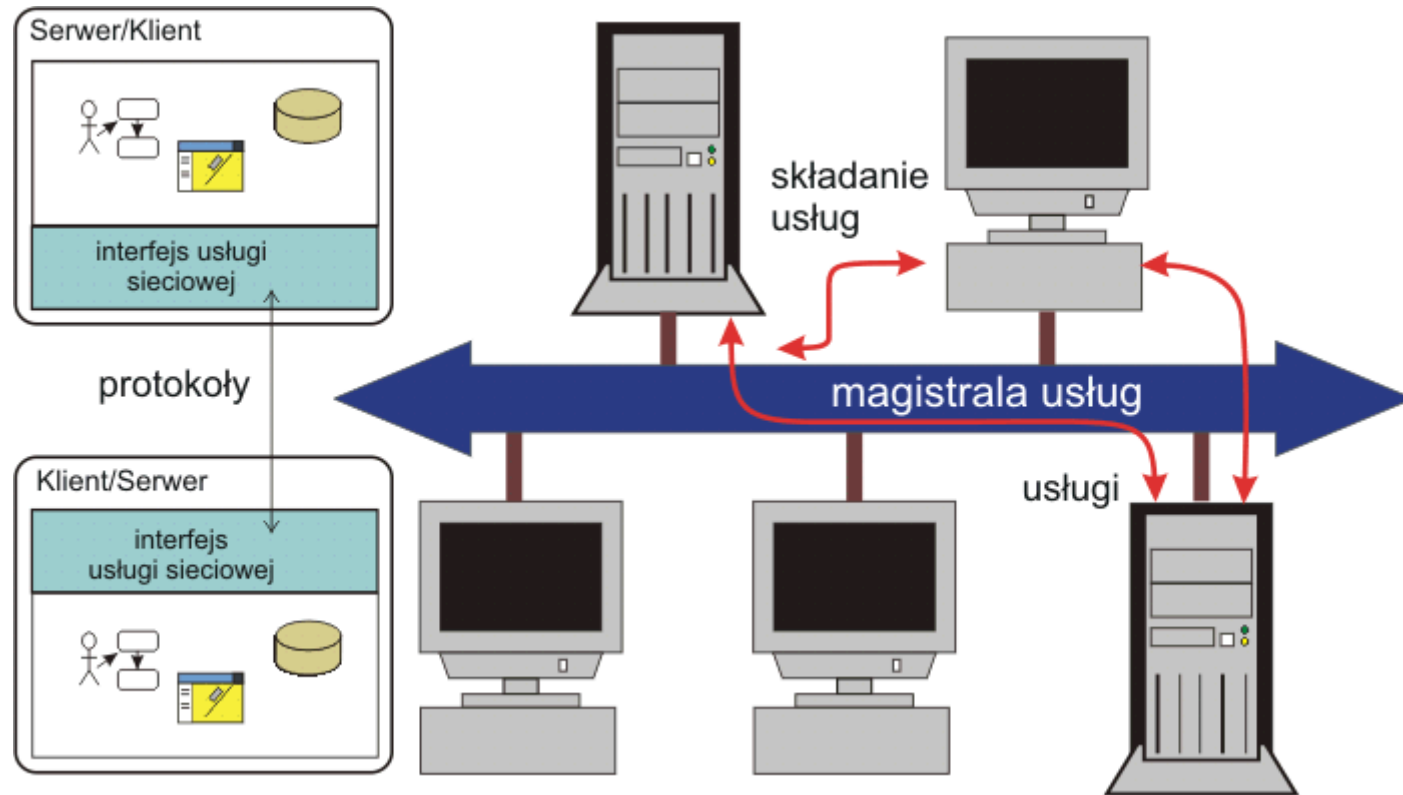
# Architektura rozwiązań

## n-warstwowa



# Architektura rozwiązań

## zorientowana na usługi



# Organizacje dostarczające specyfikacje dla usług sieciowych

<p>W3C World Wide Web Consortium <a href="http://www.w3c.org">www.w3c.org</a> Infrastructure (HTTP, XML, SOAP, SSL, WSDL)</p>	<p>WS-I Web Services Inteoperability Organization <a href="http://www.ws-i.org">www.ws-i.org</a> Web Services (WS-SP, WS-BSP)</p>
<p>OASIS Organization for the Advancement of Structured Information Standards <a href="http://www.oasis-open.org">www.oasis-open.org</a> Integration &amp; Interoperability (ebXML, UDDI)</p>	<p>OGC Open Geospatial Consortium <a href="http://www.opengeospatial.org">www.opengeospatial.org</a> Geospatial Standards (WMS, WFS, WCS, CS-W, ...)</p>

WSDL ↔ GetCapabilities  
UDDI ↔ CS-W

# Obszary zastosowań technologii informatycznych i ich funkcje

Komponenty	Technologie	Funkcje
Portale	HTML, XML, XSL, JSP, ASP, PHP, Java, .NET	Wyszukanie, wizualizacja, zamawianie, publikowanie, administracja
Usługi	XML, SOAP, WSDL, OWS, GML	Udostępnianie danych, tworzenie wartości dodanej, wykonywanie transakcji
Bazy danych	SQL, XQuery	Zarządzanie danymi w formacie rastrowym, wektorowym, tablicowym



# Metody zapytań HTTP

- HTTP/1.1

- **GET**, **POST**, **PUT**, **DELETE**, **HEAD**, **TRACE**,  
**OPTIONS**, **CONNECT**

- HTTP/1.0

- **GET**, **HEAD**, **POST**,
- **PUT**, **DELETE**, **LINK**, **UNLINK**  
(zaimplementowane przez niektóre serwery i klientów, ale niespójnie lub w ogóle niezdefiniowane)

# Najczęściej wykorzystywane metody zapytań HTTP

## GET

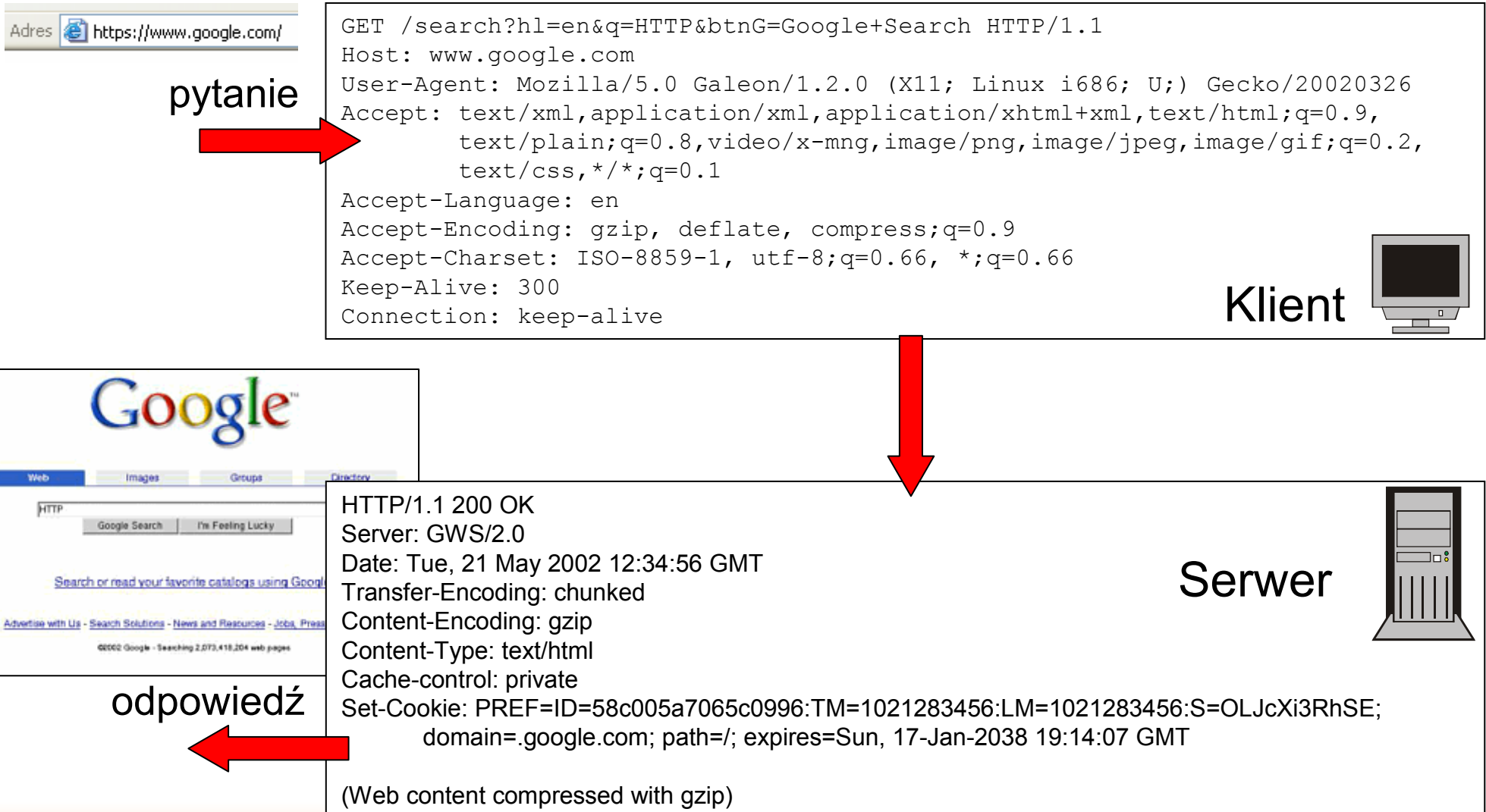
- zazwyczaj wywoływane z paska adresu przeglądarki
- można użyć jako zakładkę
- parametry przekazywane za pomocą par klucz-wartość

## POST

- wywoływane z wnętrza aplikacji
- nie można użyć jako zakładkę
- stosowane do przekazywania w zapytaniu obszernych danych

```
<a href="http://www.google.com/search?hl=en&ie=ISO-8859-1&q=HTTP&btnG=Google+Search">HTTP</a>
```

# Przykład komunikacji z metodą HTTP GET



# Elementy zapytania HTTP

- linia żądania
- nagłówki HTTP
  - nagłówki ogólne (pojawiają się w zapytaniu i odpowiedzi HTTP)
  - nagłówki żądania (są specyficzne dla konkretnego zapytania HTTP)
  - nagłówki encji (pojawiają się, jeśli pojawia się zawartość)
- zawartość



```
GET /request=WMS HTTP/1.1
```

```
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shock  
wave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application  
/mword, */*
```

```
Accept-Language: pl
```

```
Accept-Encoding: gzip, deflate
```

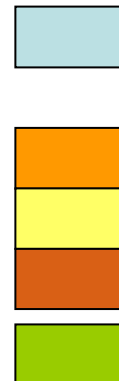
```
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1  
.4322)
```

```
Host: localhost:3541
```

```
Connection: Keep-Alive
```

# Elementy zapytania HTTP

- linia żądania
- nagłówki HTTP
  - nagłówki ogólne (pojawiają się w zapytaniu i odpowiedzi HTTP)
  - nagłówki żądania (są specyficzne dla konkretnego zapytania HTTP)
  - nagłówki encji (pojawiają się, jeśli pojawia się zawartość)
- zawartość



```
POST /search HTTP/1.1
```

```
Host: www.google.com
```

```
User-Agent: Mozilla/5.0 Galeon/1.2.5 (X11; Linux i686; U;) Gecko/20020606
```

```
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,  
text/plain;q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2,  
text/css,*/*;q=0.1
```

```
Accept-Language: en
```

```
Accept-Encoding: gzip, deflate, compress;q=0.9
```

```
Accept-Charset: ISO-8859-1, utf-8;q=0.66, */*;q=0.66
```

```
Keep-Alive: 300
```

```
Connection: keep-alive
```

```
Content-Type: application/x-www-form-urlencoded
```

```
Content-Length: 31
```

```
hl=en&q=HTTP&btnG=Google+Search
```

# Elementy odpowiedzi HTTP

- linia statusu (wersja HTTP, kod statusu, krótki opis statusu)
- nagłówki HTTP
  - nagłówki ogólne (pojawiają się w zapytaniu i odpowiedzi HTTP)
  - nagłówki odpowiedzi (są specyficzne dla konkretnej odpowiedzi HTTP)
  - nagłówki encji (pojawiają się, jeśli pojawia się zawartość)
- zawartość



```
HTTP/1.1 302 Object moved
```

```
Server: Microsoft-IIS/5.1
```

```
Date: Thu, 23 Aug 2007 17:48:48 GMT
```

```
X-Powered-By: ASP.NET
```

```
Location: localstart.asp
```

```
Content-Length: 121
```

```
Content-Type: text/html
```

```
Set-Cookie: ASPSESSIONIDACQBQQS=CEGLAHJCPAEBAIINILNPHKAF; path=/
```

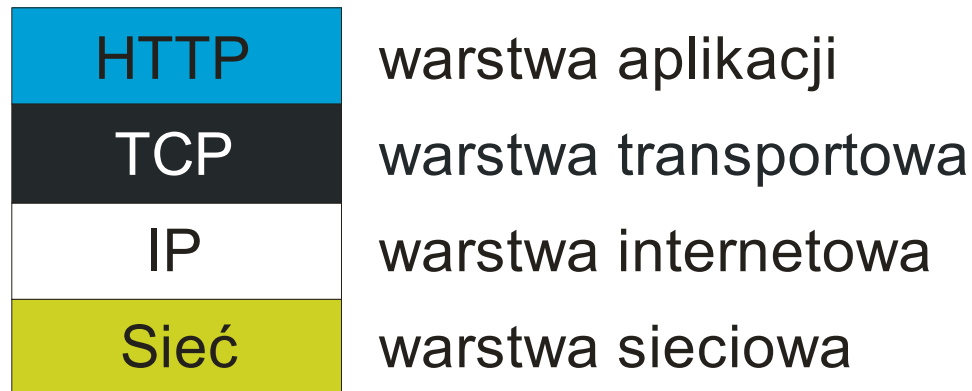
```
Cache-control: private
```

```
<head><title>Object moved</title></head>
```

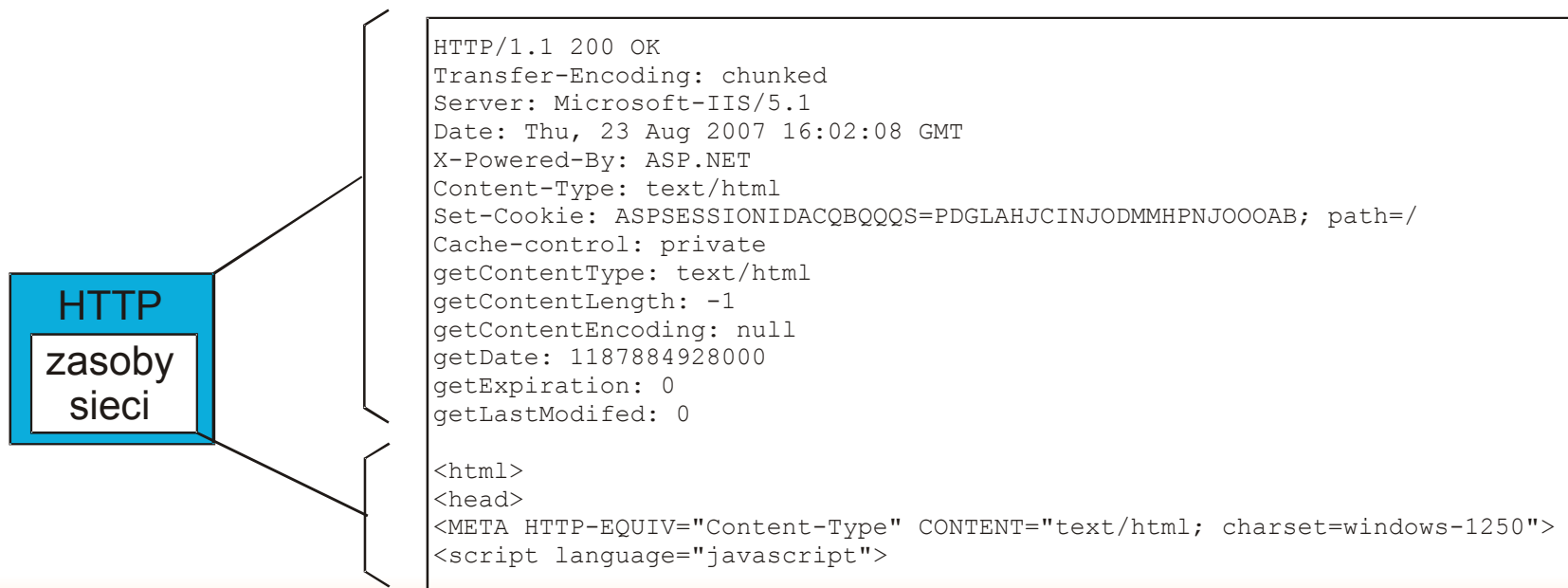
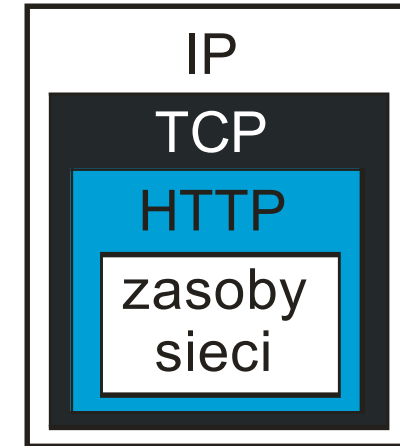
```
<body><h1>Object Moved</h1>This object may be found <a  
HREF="">here</a>.</body>
```

# Czterowarstwowy model protokołów sieciowych

- Model „klasyczny”



- Model zagnieżdżony

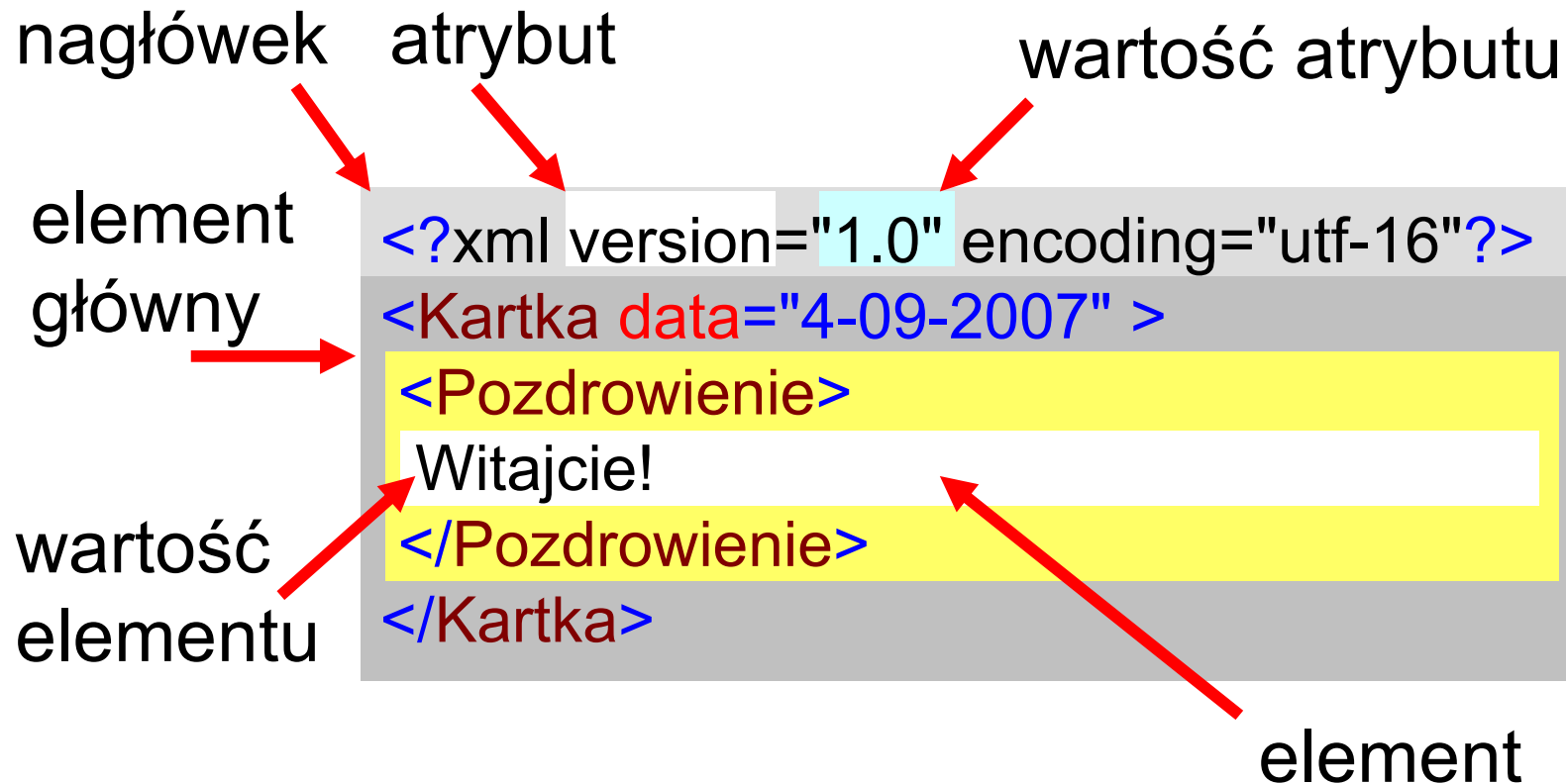


# Czym jest XML?

- Skrót od *eXtensible Markup Language*
- Język znaczników dla dokumentów o strukturze hierarchicznej
- Pochodzi od SGML
- Standard przekazywania informacji wraz z ich opisem
- Plik tekstowy



# Podstawowe elementy dokumentu XML



# Programowe przetwarzanie XML

- parsowanie i przeglądanie pliku XML
- selekcja podzbiorów, elementów, atrybutów
- język zapytań do danych w formacie XML
- transformacja dokumentów XML
- powiązanie z bazami danych
- wizualizacja i formatowanie
- implementacja usług sieciowych

# Cechy poprawnego dokumentu

- Na początku deklaracja:

```
<?xml version="1.0"?>
```

- Dokument zawiera jeden *element* nazywany *korzeniem* (*element główny*)

- *Element* może zawierać inne elementy oraz *atrybuty*

- *Element* otoczony jest znacznikami początku <...> oraz końca </...>

- Element pusty (nie zawierający innych elementów) można opisać jednym *znacznikiem nieparzystym* <.../>

- Atrybuty elementu wymienia się w jego znaczniku, podając od razu wartość w cudzysłowie:

```
<element attr1="val1" attr2="val2" />
```

- Komentarze jak w HTML <!-- ... -->

- rozróżnianie wielkości liter

- Sekcje CDATA

```
<![CDATA[
```

Każdy <znacznik /> tutaj zostanie zignorowany.

```
]]>
```

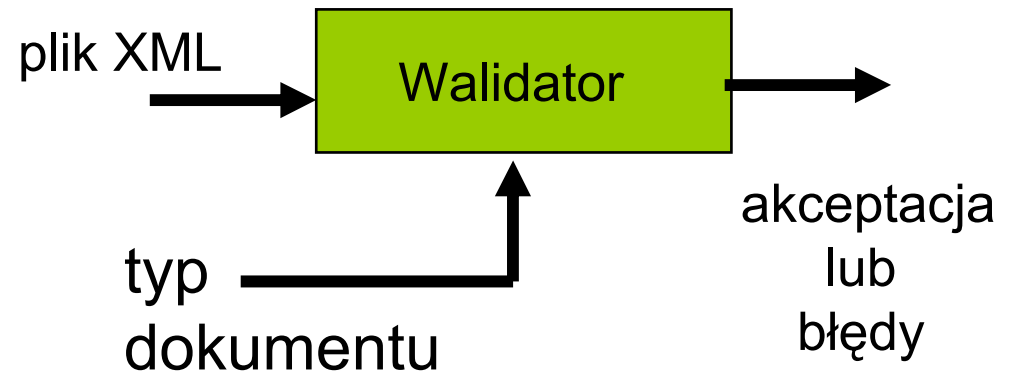
# Walidacja z DTD

## dokument kartka.xml

```
<?xml version="1.0" encoding="utf-16"?>
<!DOCTYPE GREETING SYSTEM "kartka.dtd">
<Kartka data="4-09-2007" >
  <Pozdrowienie>
    Witajcie!
  </Pozdrowienie>
</Kartka>
```

## schemat kartka.dtd

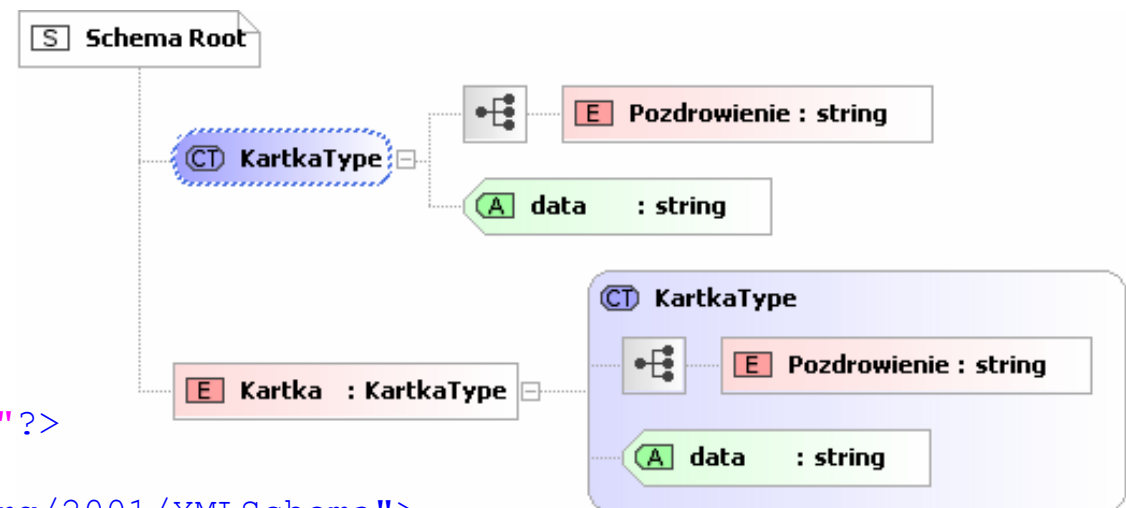
```
<?xml version="1.0" encoding="utf-8"?>
<!ELEMENT Kartka (Pozdrowienie)>
<!ELEMENT Pozdrowienie (#PCDATA)>
<!ATTLIST Kartka data CDATA #REQUIRED>
```



# Walidacja z XML Schema

dokument kartka.xml

```
<?xml version="1.0"?>
<Kartka xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="kartka.xsd" data="4-09-2007" >
  <Pozdrowienie>
    Witajcie!
  </Pozdrowienie>
</Kartka>
```



schemat kartka.xsd

```
<?xml version="1.0" encoding="utf-16"?>
<!-- Plik przykładowy-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="KartkaType">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="Pozdrowienie" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="data" type="xs:string" />
  </xs:complexType>
  <xs:element name="Kartka" type="KartkaType" />
</xs:schema>
```

# Porównanie sposobów walidacji oraz cechy XPath

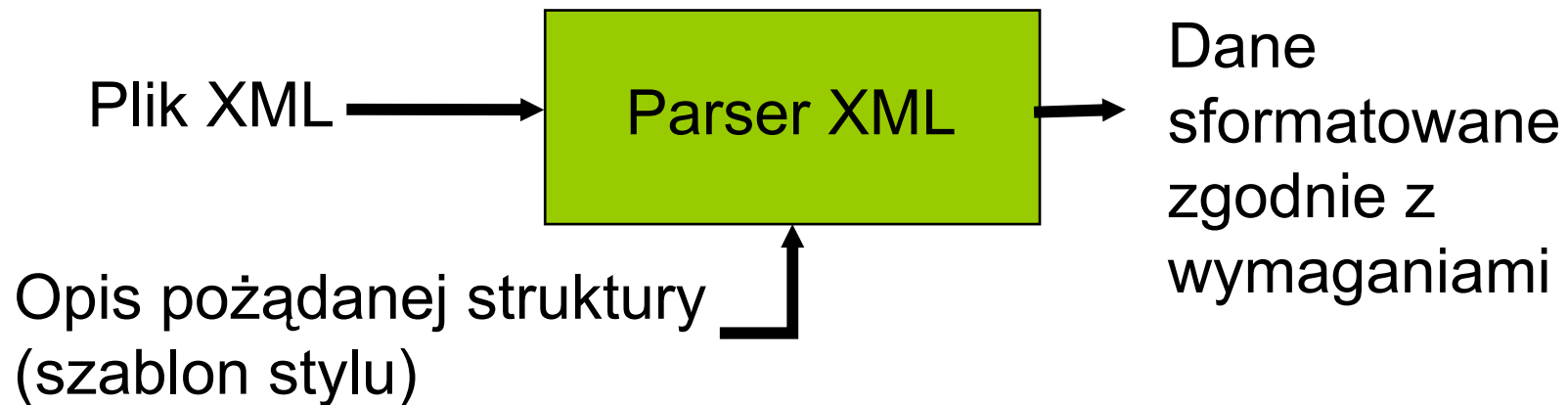
- Różnice między DTD a Schema
  - plik Schema jest plikiem XML
  - można nakładać silne restrykcje na wartości (własne typy, wyrażenia regularne, typy abstrakcyjne)
  - można elastycznie budować strukturę dokumentu
- XPath
  - Pozwala poruszać się po dokumencie
  - Główny element w XSLT
  - Ogromna biblioteka funkcji
  - Bardzo podobne do ścieżek w drzewie katalogów w UNIX
  - Ponadto dodatkowy język „poruszania się” po węzłach

# Technologie XML

- Walidacja i opis danych
  - DTD, RDF, XSD
- Selekcja i wskazywanie
  - XPath, XPointer
- Przekształcanie
  - XSLT
- Łączenie i wiązanie
  - XLink
- Reprezentacja graficzna
  - SVG, VML, X3D, GML
- Protokoły i zastosowania sieciowe
  - SOAP, WSDL, UDDI

# Cechy XSLT (Extensible Stylesheet Language Transformation)

- Określa jak interpretować dokument XML jako dokument innego formatu
- Umożliwia przetworzenie na HTML, PDF oraz inny rodzaj XML, np. SVG
- Umożliwia tworzenie informacji o danych już istniejących (zliczanie, sortowanie)

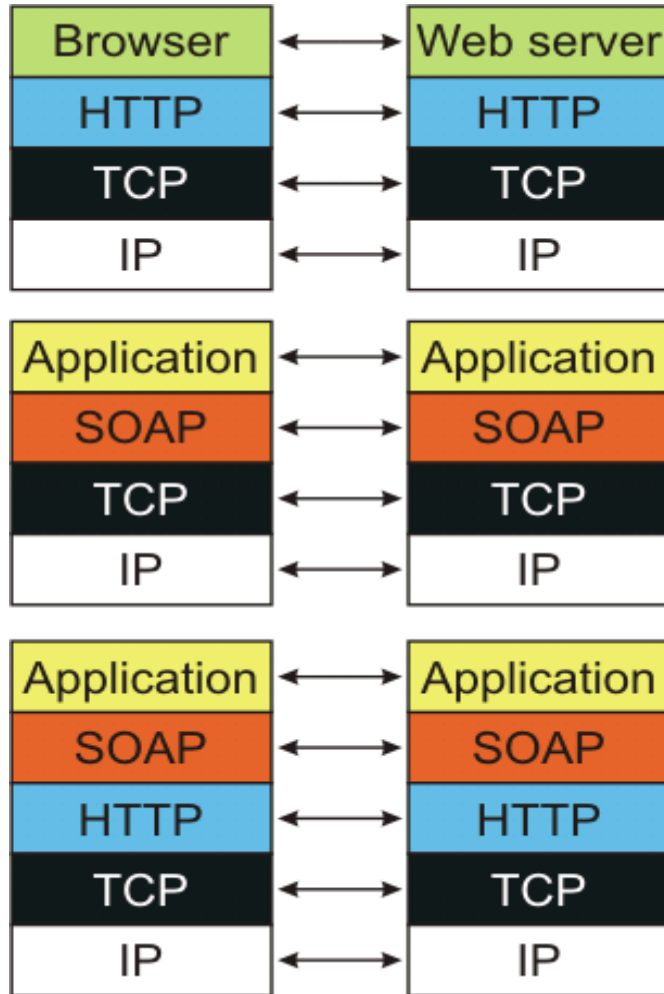




# SOAP (*Simple Object Access Protocol*)

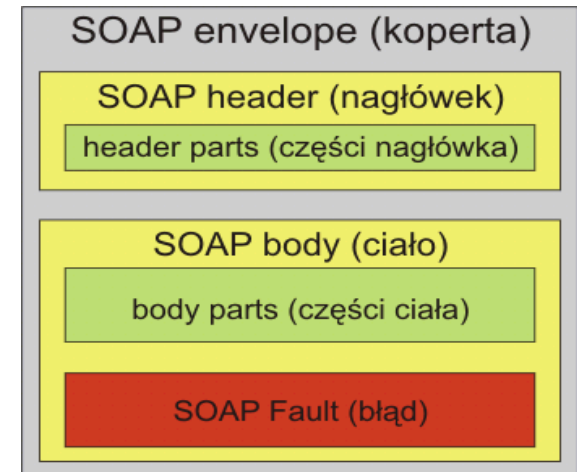
- Protokół bazującym na XML i XSD, służącym do wymiany informacji pomiędzy aplikacjami w środowisku rozproszonym
- Jest wykorzystywany w systemach rozproszonych zorientowanych na przesyłanie wiadomości i zdalne wywoływanie procedur (RPC - *Remote Procedure Call*)
- Może być wykorzystywany jako kolejna warstwa nad warstwą transportową (TCP, HTTP)
- Użyty nad HTTP daje możliwość implementacji usług w środowisku sieciowym z zaporami ogniowymi i serwerami proxy
- Specyfikacja SOAP określa standard formatu wiadomości SOAP oraz mówi jak używać SOAP nad warstwą transportową

# SOAP nad warstwą transportową

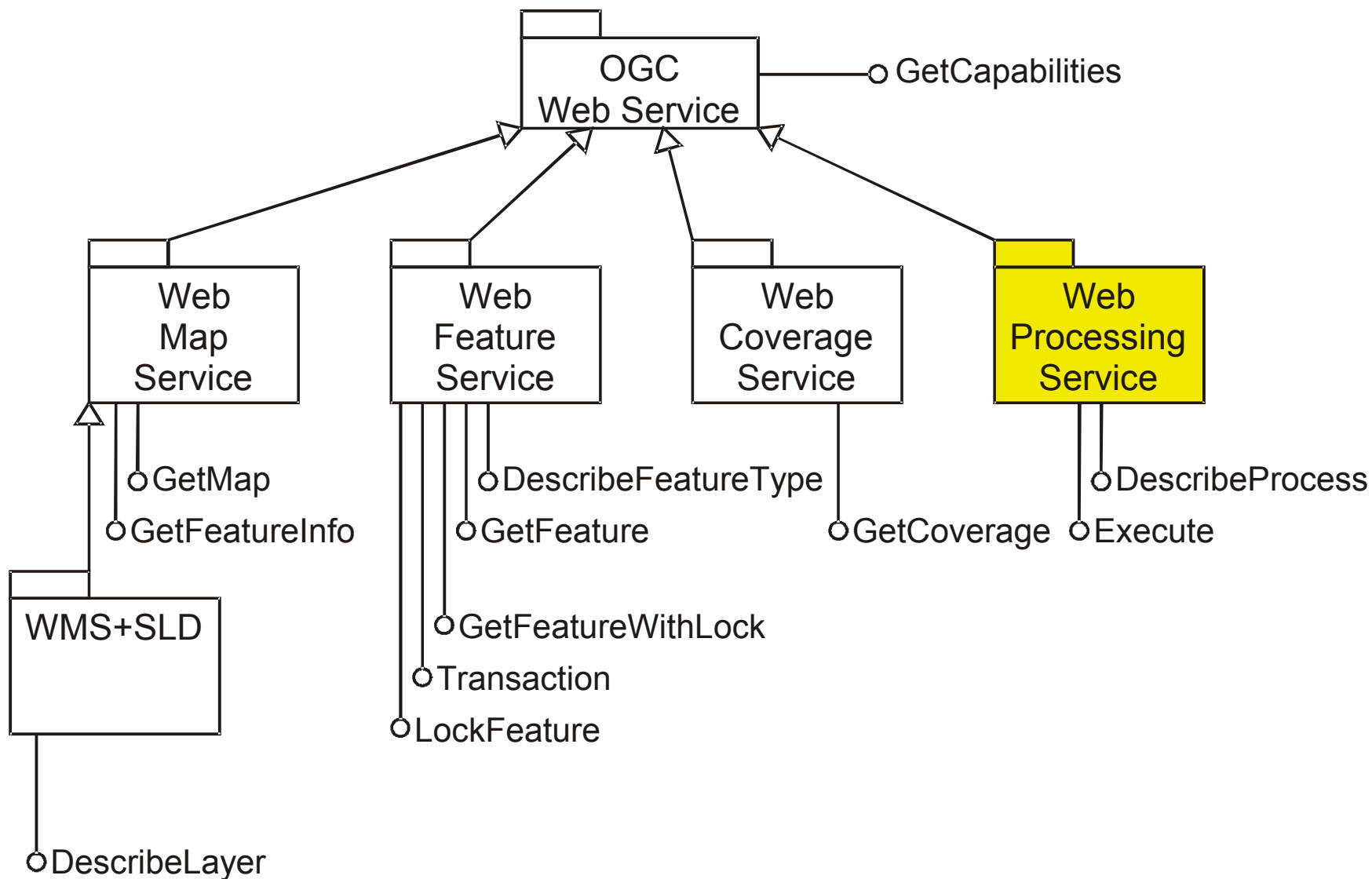


```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <authHeader
xmlns="http://mojadres.com/ppp">
      <authToken>
        skdfs213jj4jkaerJKDSN334021SWDM
      </authToken>
    </authHeader>
  </soapenv:Header>
  <soapenv:Body>
    <SubmitInvoice
xmlns="http://mojadres.com/ppp">
      <invoiceDoc></invoiceDoc>
    </SubmitInvoice>
  </soapenv:Body>
</soapenv:Envelope>
  
```

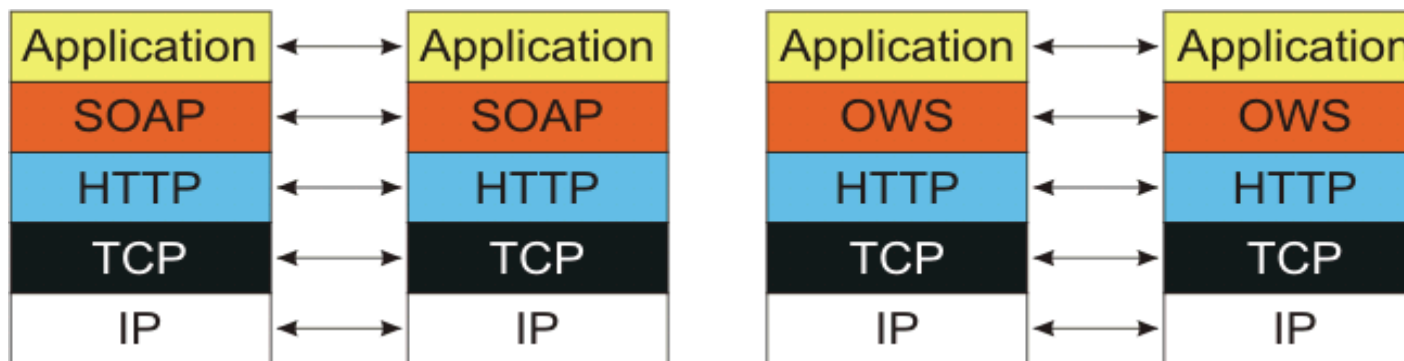


# Interfejsy wybranych usług sieciowych OGC



# Opis interfejsów OWS (*OGC Web Service*)

- OGC definiuje usługi i dostarcza specyfikacji implementacyjnych ich interfejsów (protokołów)
- Podobnie jak SOAP opis protokołów bazuje na XML i XSD (dla starszych wersji usług jest DTD), zaś komunikacja na nich oparta realizowana jest jako kolejna warstwa nad protokołem HTTP
- Zamiast WSDL do uzyskania opisu interfejsu w sposób automatyczny wykorzystywane są zapytania zdefiniowanego protokołu (GetCapabilities)
- Rolę UDDI pełnią usługi CSW (Catalogue Service-Web)



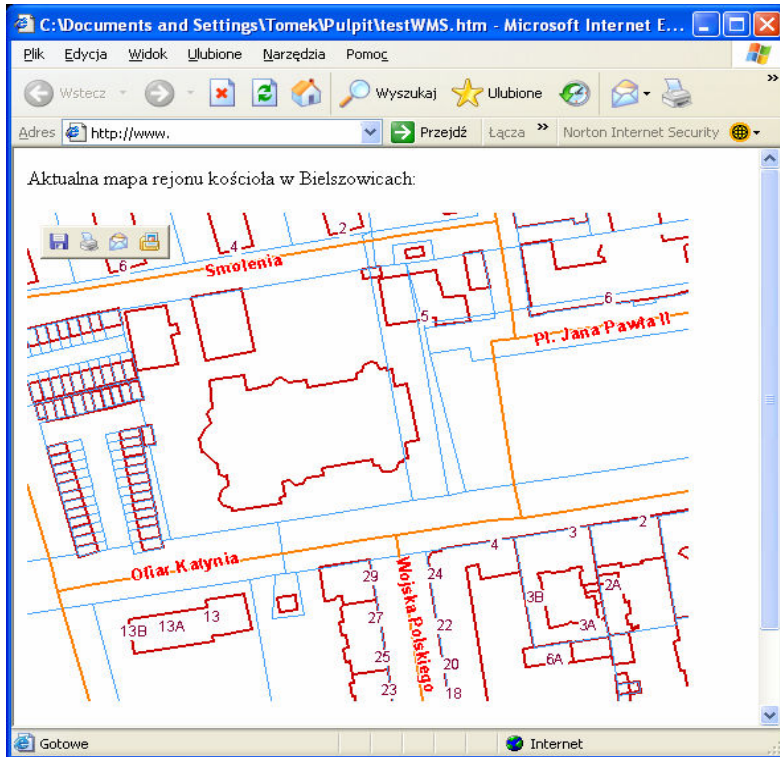
# Protokół WMS (*Web Map Server*)

- Jest to bezstanowy protokół służący do udostępniania map poprzez HTTP, który musi implementować GET (klucze w parametrach nie są czułe na wielkość liter, wartości są czułe na wielkość liter)
- Operacje
  - GetCapabilities
    - VERSION, SERVICE (gdyż jeden serwis może dostarczać wielu usług jak WMS, WFS, WCS), REQUEST
    - <http://serwer.wms/mapa?version=1.1.1&service=WMS&request=GetCapabilities>
  - DescribeLayer
    - VERSION, SERVICE, REQUEST, LAYERS
    - <http://serwer.wms/mapa?version=1.1.1&service=WMS&request=DescribeLayer&layers=rivers>
  - GetMap
    - FORMAT, BBOX (minx,miny,maxx,mxy), SRS, WIDTH, HEIGHT, LAYERS, STYLES, TIME (choć nie wszystkie) i inne

# Dodatkowe operacje dla WMS + SLD

- GetStyles
  - zwraca opis stylu SLD dla wybranej warstwy
  - Parametry
    - VERSION, SERVICE, REQUEST, LAYERS
- GetLegendGraphic
  - dynamiczna ikona legendy dla danej warstwy
  - Parametry
    - VERSION, SERVICE, REQUEST, FORMAT, LAYER, SLD
- PutStyles
  - zapamiętuje dokument SLD na serwerze WMS
- Przykłady użycia parametru SLD
  - &sld=[http://localhost/ms\\_ogc\\_workshop/sld/rivers.sld](http://localhost/ms_ogc_workshop/sld/rivers.sld)
  - &sld\_body=<entire\_sld\_document>
  - można przesłać SLD\_BODY w metodzie HTTP POST

# Przykład użycia zapytania WMS jako zakładki



```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=Windows-1250">
</head>
<body>
Aktualna mapa rejonu kościoła w Bielszowicach:
<br><br>

<br><br>
</body>
</html>
```

W przypadku błędów serwer będzie zwracał informacje o nich w dokumencie XML

# Przykład odpowiedzi serwera WMS

<http://ogc.intergraph.pl/map/Request.aspx?request=GetCapabilities&service=WMS>

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE WMT_MS_Capabilities SYSTEM "http://ogc.intergraph.pl/map/schemas/capabilities_1_1_1.dtd"
[ <!ELEMENT VendorSpecificCapabilities EMPTY]>
<WMT_MS_Capabilities updateSequence="53" version="1.1.1">
  <Service>
    <Name>OGC:WMS</Name>
    <!--Human-readable title for pick lists-->
    <Title>Map service</Title>
    <!--Narrative description providing additional information-->
    <Abstract>
    <KeywordList />
    <!--Top-level address of service or service provider-->
    <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="http://www.intergraph.com" xlink:type="simple" />
    <ContactInformation>
    <Fees>none</Fees>
    <AccessConstraints>none</AccessConstraints>
  </Service>
  <Capability>
    <Request>
      <GetCapabilities>
      <GetMap>
      <GetFeatureInfo>
      <DescribeLayer>
      <GetLegendGraphic>
    </Request>
    <Exception>
      <UserDefinedSymbolization SupportSLD="1" UserLayer="0" UserStyle="1" RemoteWFS="0" />
    <Layer>
  </Capability>
</WMT_MS_Capabilities>
```



# Usługa WMS

## ● Cechy

- prostota
- szybkość (rasteryzacja po stronie serwera, przez sieć przesyłanie obrazu)
- możliwość przesyłania wektorów (text/xml)
- prezentacja danych, a nie udostępnianie danych
- możliwość zadawania pytań o punkt na mapie
- usługi WMS można integrować (kolejne warstwy mapy)

# Operacje WCS

- Rastrowy równoważnik WFS, dostarczający rzeczywiste dane rastrowe w rastrowym formacie (a nie samą wizualizację danych)
- Operacje
  - GetCapabilities
    - VERSION, SERVICE, REQUEST
    - <http://serwer.map?version=1.0.0&service=WCS&request=GetCapabilities>
  - DescribeCoverage
    - VERSION, SERVICE, REQUEST, COVERAGE
    - <http://serwer.map&version=1.0.0&service=WCS&request=DescribeCoverage&coverage=toronto>
  - GetCoverage
    - VERSION, SERVICE, REQUEST, COVERAGE, CRS, BBOX

# Operacje WFS

- GetCapabilities

- VERSION, SERVICE, REQUEST
- <http://serwer.wfs/wfs?version=1.0.0&service=WFS&request=GetCapabilities>

- DescribeFeatureType

- VERSION, SERVICE, REQUEST, TYPENAME
- <http://serwer.wfs/wfs?version=1.0.0&service=WFS&request=DescribeFeatureType&typename=rivers>

- GetFeature

- VERSION, SERVICE, REQUEST, TYPENAME, FILTER (opcjonalne), BBOX (może być zrobione przez FILTER)

# Przykład filtru OGC

- Zapytanie SQL

- select \* from roads where roadtype = 1

- Postać filtru

```
<Filter>  
  <PropertyIsEqualTo>  
    <PropertyName>roadtype</PropertyName>  
    <Literal>1</Literal>  
  </PropertyIsEqualTo>  
</Filter>
```

**KONIEC**