

# SOAP

na przykładzie Allegro WebAPI

19.10.2009

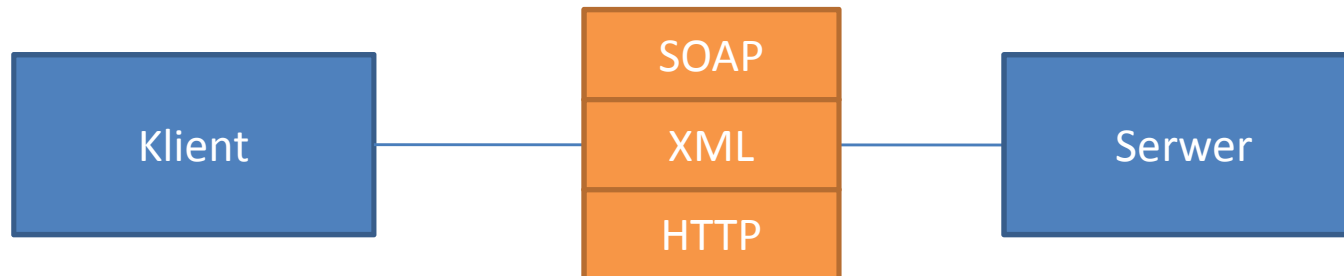
KN Atena

# Web Service

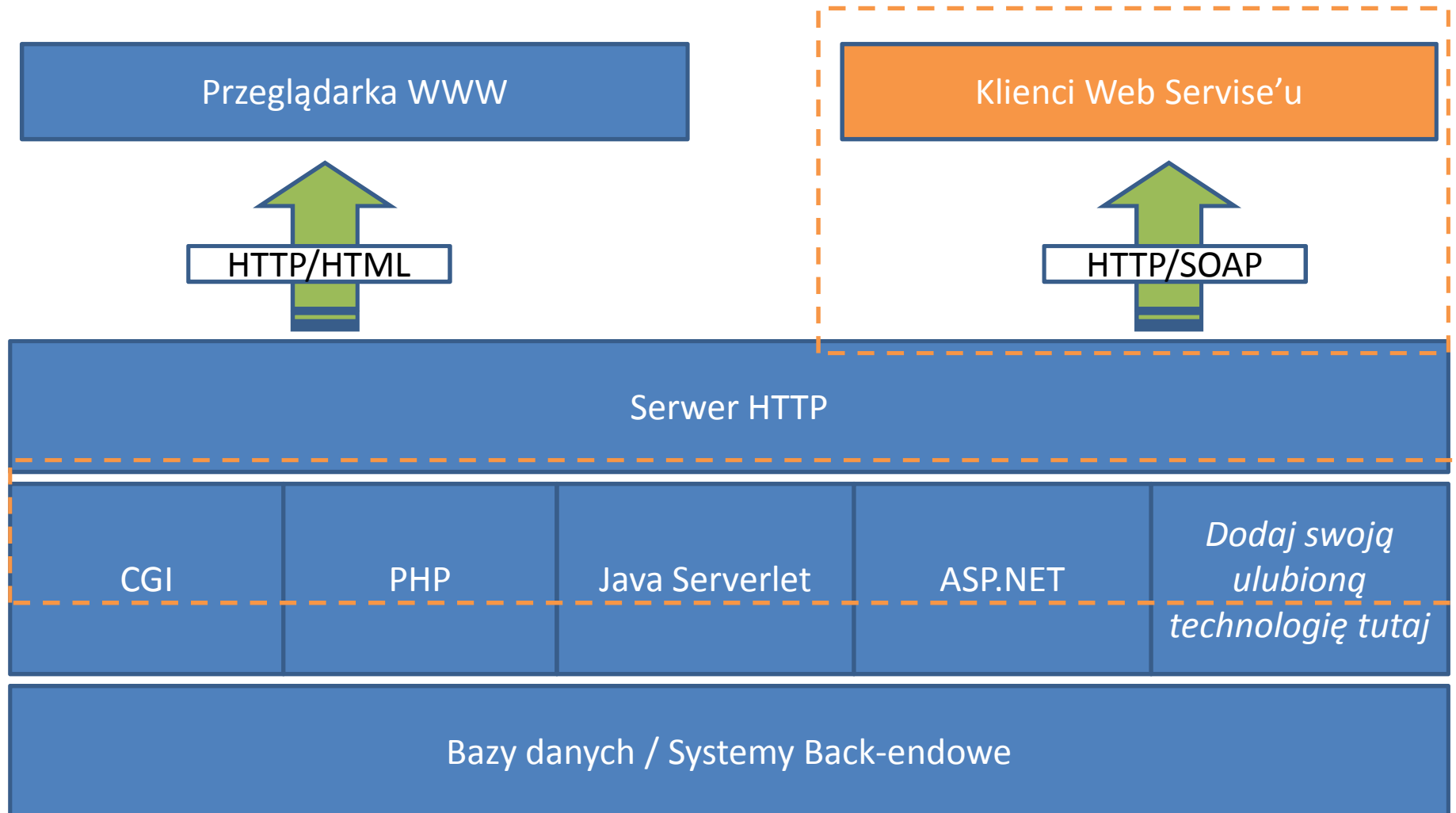
- **Web-based Service** - Usługa oferowana przez stronę WWW



- **Web Service** – Usługa oferowana przez ustandaryzowane protokoły



# Web Service



# SOAP

- Protokół wymiany ustrukturyzowanych informacji przede wszystkim przez wywoływanie zdalnych metod (RPC - *Remote Procedure Calls*)
- Napisany w XML
- Nie jest protokołem transportującym lecz opakowującym dane
- HTTP jest najpopularniejszym protokołem transportującym SOAP
- Rozwijany przez W3C
- Podobny do DCOM, CORBA, Java RMI

# Zalety

- Niezależny od platformy i języka (implementacje Apache, Microsoft, Java, PHP i inne)
- Wspólny język do wymiany danych między aplikacjami wykonanymi w różnych technologiach (nie ma problemu kompatybilności implementacji)
- Interfejs dostępowy bez wglądu do wewnętrznej struktury źródła danych
- Dzięki działaniu na HTTP nie ma problemu z firewallami

# Budowa wiadomości SOAP

SOAP Envelope (ang. koperta)

SOAP Header (opcjonalnie)

Header Block

...

Header Block

SOAP Body

Body Block

...

Body Block

```
<?xml version="1.0" ?>
<env:Envelope
  xmlns:env="http://www.w3.org/soap-envelope">

  <env:Header>
    <data:headerBlock
      xmlns:data="http://example.com/header"
      env:actor="http://example.com/actor"
      env:mustUnderstand="true">
      ...
    </data:headerBlock>
    ...
  </env:Header>

  <env:Body>
    <data:bodyBlock
      xmlns:data="http://example.com/header">
      ...
    </data:bodyBlock>
    ...
  </env:Body>

</env:Envelope>
```

# Przykład wiadomości

```
public int RectField(int x, int y) { return x * y; }
```

## Żądanie

```
POST /ExampleAPP HTTP/1.1
Host: example.com
Content-Type: text/xml;charset="utf8"
Content-Length: nnn

<?xml version="1"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org
  /soap/envelope/">
  <soap:Body>
    <RectField>
      <x>3</x>
      <y>5</y>
    </RectField>
  </soap:Body>
</soap:Envelope>
```

## Odpowiedź

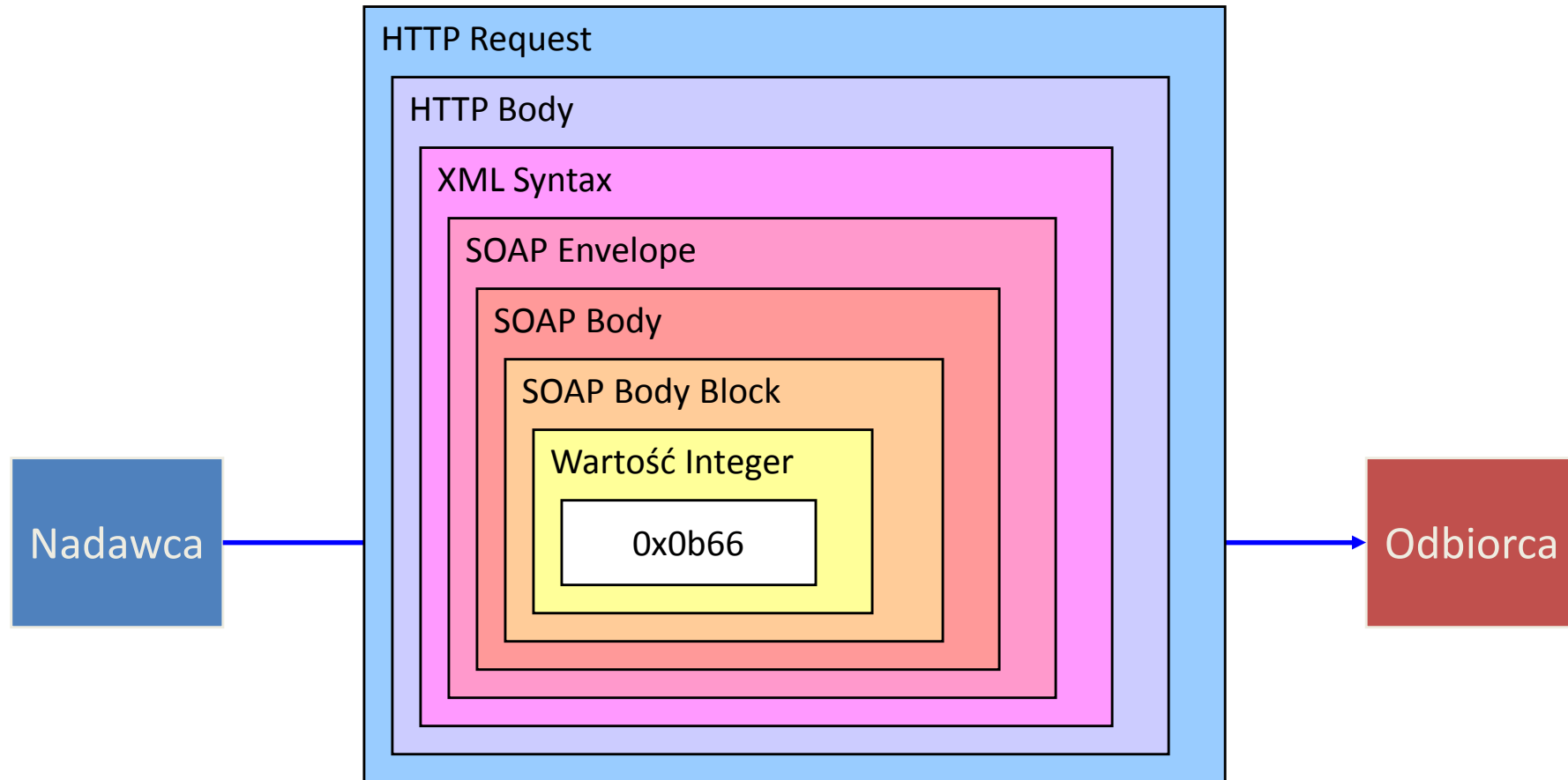
```
HTTP/1/1 200 OK.
Content-Type: text/xml;charset="utf8"
Content-Length: nnn

<?xml version="1"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org
  /soap/envelope/">
  <soap:Body>
    <RectFieldResponse>
      <RectFieldResult>15</RectFieldResult>
    </RectFieldResponse>
  </soap:Body>
</soap:Envelope>
```

# Kodowanie

- Typy skalarne
  - Kodowane przy użyciu XML Schema *simple types*
- Typy złożone
  - Tablice: wpisy identyfikowane przez pozycję
    - Tablice wielowymiarowe nie są wspierane przez wszystkie implementacje
  - Struktury: wpisy identyfikowane przez nazwę
- Akceptowane są tylko dane tekstowe ale istnieją metody obejścia tego ograniczenia np.:
  - MIME(base64)/MTOM(XOP - XML-binary Optimized Packaging)
  - Serializacja

# Wydajność



# Inne

- W nagłówku wiadomości możemy przekazywać dodatkowe informacje np. o uwierzytelnieniu czy odpłatności
  - np. `<soap:Header><ID>B93FF457D6458A396E3C4</ID></soap:Header>`
  - `<ID soap:mustUnderstand="1">B93FF457D6458A396E3C4</ID>`
- Mechanizm pośredników
- Faults – dostarczają informacji o błędach występujących przy przetwarzaniu wiadomości
- WSDL - Opisuje zbiór operacji i formaty żądań potrzebnych do ich użycia

# Gdy nie mamy Web Service'u...

- Pobieranie
  - DOM+XPath
- Wysyłanie
  - CURL
  - Ręczne wysyłanie żądań HTTP

# Przykład komunikacji z Allegro

- C#

- Dodajemy plik WSDL do źródeł danych (Data Sources) i generujemy referencje webowe
- Tworzymy obiekt web service'u

```
pl.allegro.webapi.AllegroWebApiService myallegro = new pl.allegro.webapi.AllegroWebApiService();
```

- I już możemy używać metod jakiś potrzebujemy np.

```
myallegro.doLogin(login, password, country_code, webapi_key, version_code, out hash, out time);
```

- PHP

- Tworzymy obiekt klasy

```
$soapclient = new SoapClient('http://webapi.allegro.pl/uploader.php?wsdl');
```

- I korzystamy z metod

```
$params = array("user-login"=>„login", "user-password"=>„password",  
"country-code"=>„country_code", "webapi-key"=>„webapi_key", "local-version"=>„local_version");
```

```
$response = $soapclient->__soapCall("doLogin", $params);
```

*Lub bezpośrednio*

```
$response = $soapclient->doLogin($params);
```