

# Web Services / Gridy





Autor: Dariusz Dwornikowski  
[tdi@vercom.pl](mailto:tdi@vercom.pl)  
[tdi@kill-9.pl](mailto:tdi@kill-9.pl)

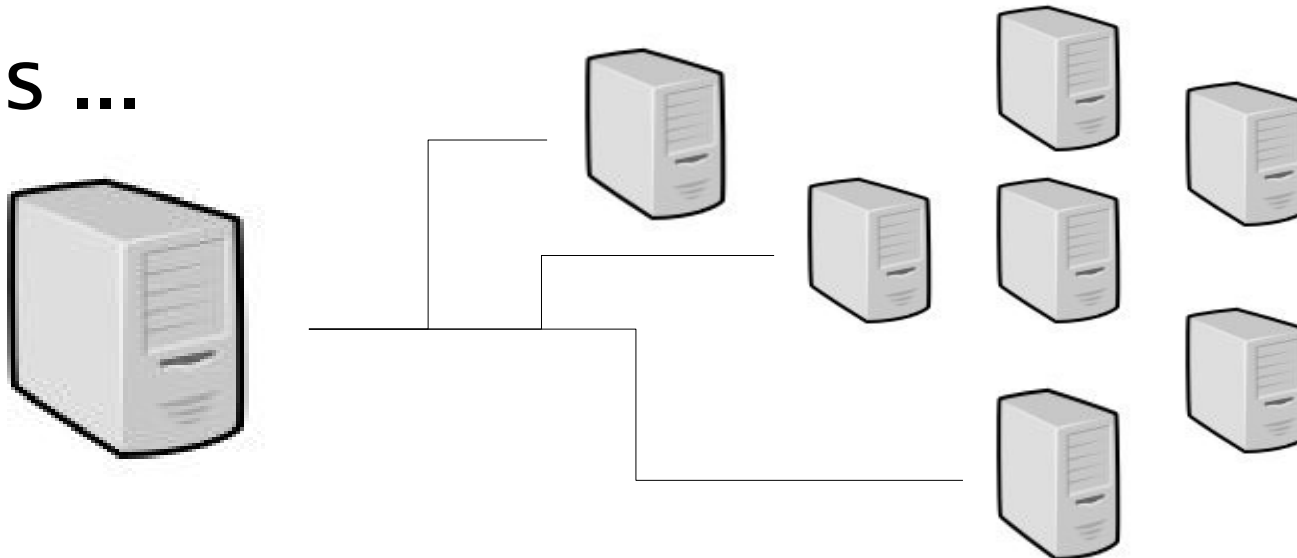
# Web Services - wstęp

SOA/Web Services – odpowiedź na potrzeby  
komercyjnego internetu

pryzmat biznesowy

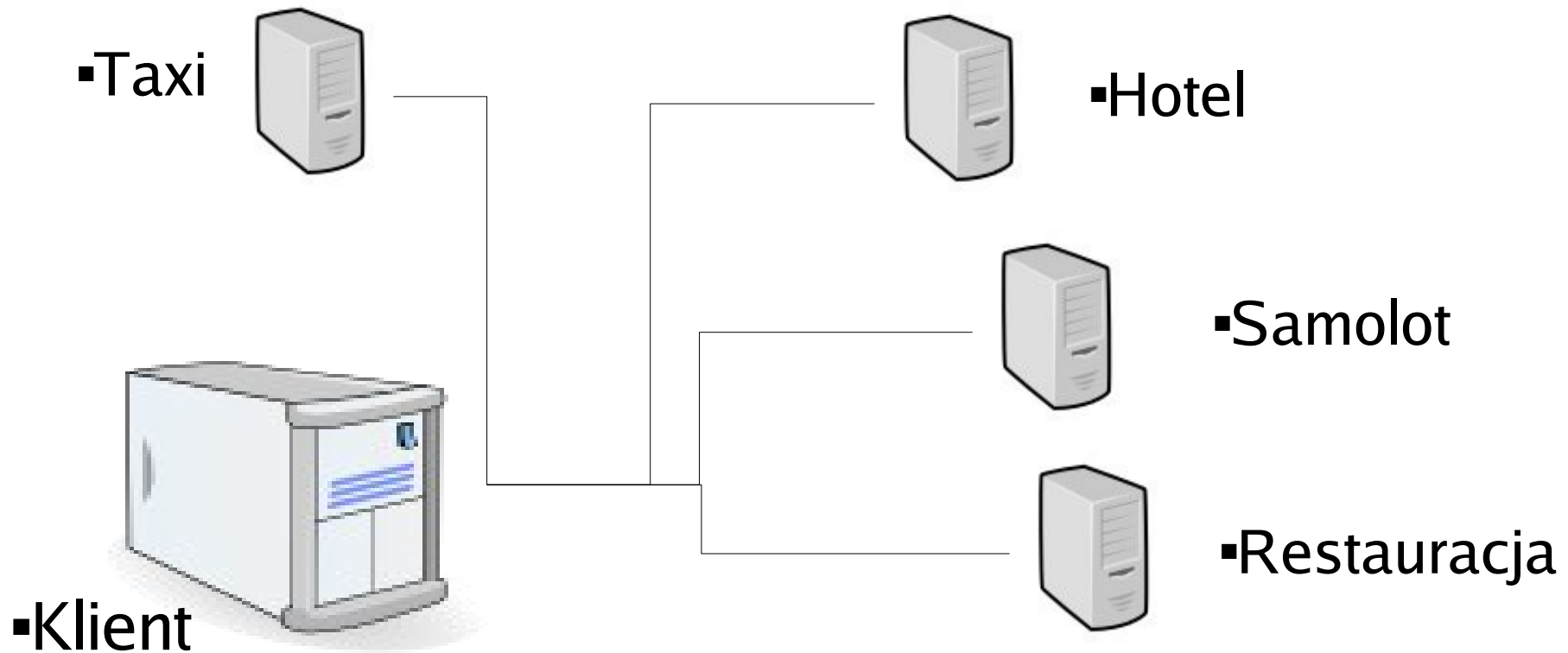
# Sieć WWW dziś

- Miliony osobnych usług
- Miliony standardów i protokołów
- Brak integracji pomiędzy serwisami
- W konsekwencji - niedostępność
- Chaos ...



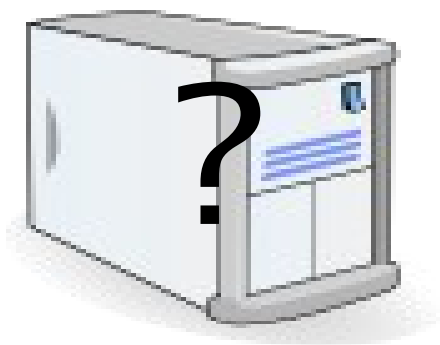
# Problemy - Wiele usług

- Rozproszenie usług, związanych ze sobą



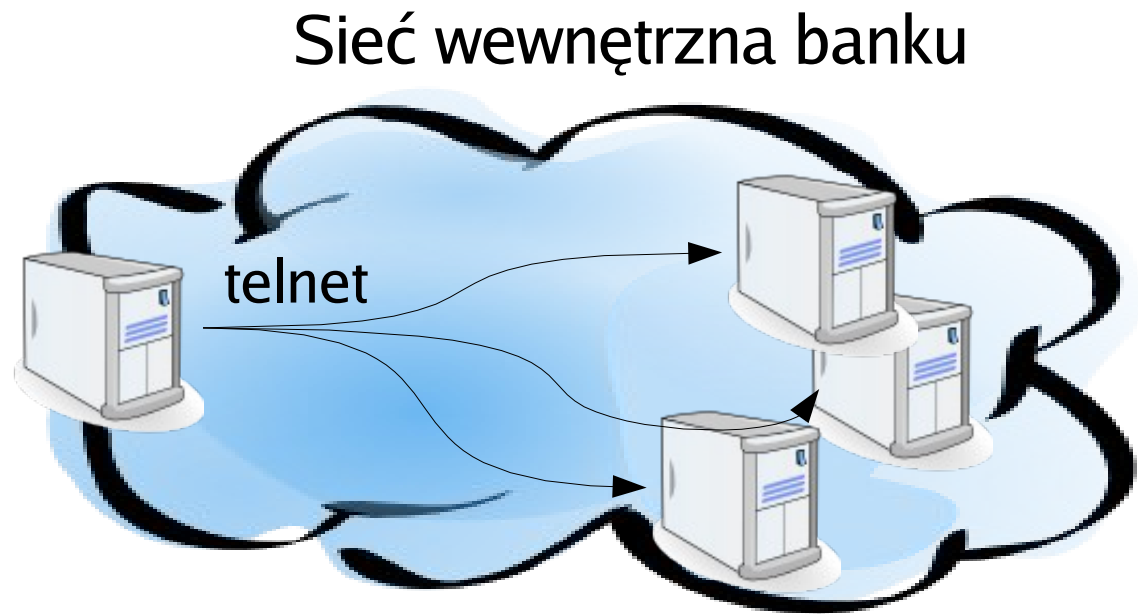
# Problemy - Wiele usług

- Przymus zapamiętywania, dokonywania wyborów, znania procedur – to nie jest zajęcie dla klientów
- Mała konkurencja – wysokie ceny – często trudno wyszukać tańsze usługi



# Problemy - Protokoły

- Różnorodność protokołów, zarówno znanych jak i własnych
  - TCP/IP ( własne rozwiązania )
  - telnet/ssh
  - Http, smtp
  - CORBA
  - DCOM
  - Czysty XML
  - Itd ...



# Problemy - Protokoły

- Własne rozwiązania, tzw. “kombinowane” wpływają na powstawanie luk w bezpieczeństwie
- Nie są skalowalne
- Każda zmiana rodzi problem (o ile jest możliwa)
- Z punktu widzenia rozwoju - Ślepy zaułek ...

# Problemy – Brak integracji między usługami

- Czy nie byłoby by prościej gdyby usługi www komunikowały się między sobą, wyręczając klienta ?

Dziś to niemożliwe :

- Obawa przed utratą rynku
- Obawa o bezpieczeństwo firmy
- Trudność wprowadzania dużych zmian
- Strach przed zmianami

# Web Services

Web Services, **częściowo** ,są odpowiedzią na problemy dzisiejszej sieci web.

Definiują **standardy** udostępniania interfejsów aplikacji w sieci web.

# Web Services

Web Services korzystają z już **znanych technologii**.

Standardy starają się **wtopić** w już istniejące technologie, niż je rewolucjonizować.

**Łatwe użytkowanie** (od strony klienta) to jeden z głównych celów.

**Przezroczystość** dla klienta.

**Multiplatformowość.** ( standardy łatwo implementować )

# Web Services - dziś

Można powiedzieć że dzisiejszy sposób przeglądania stron to także web service :

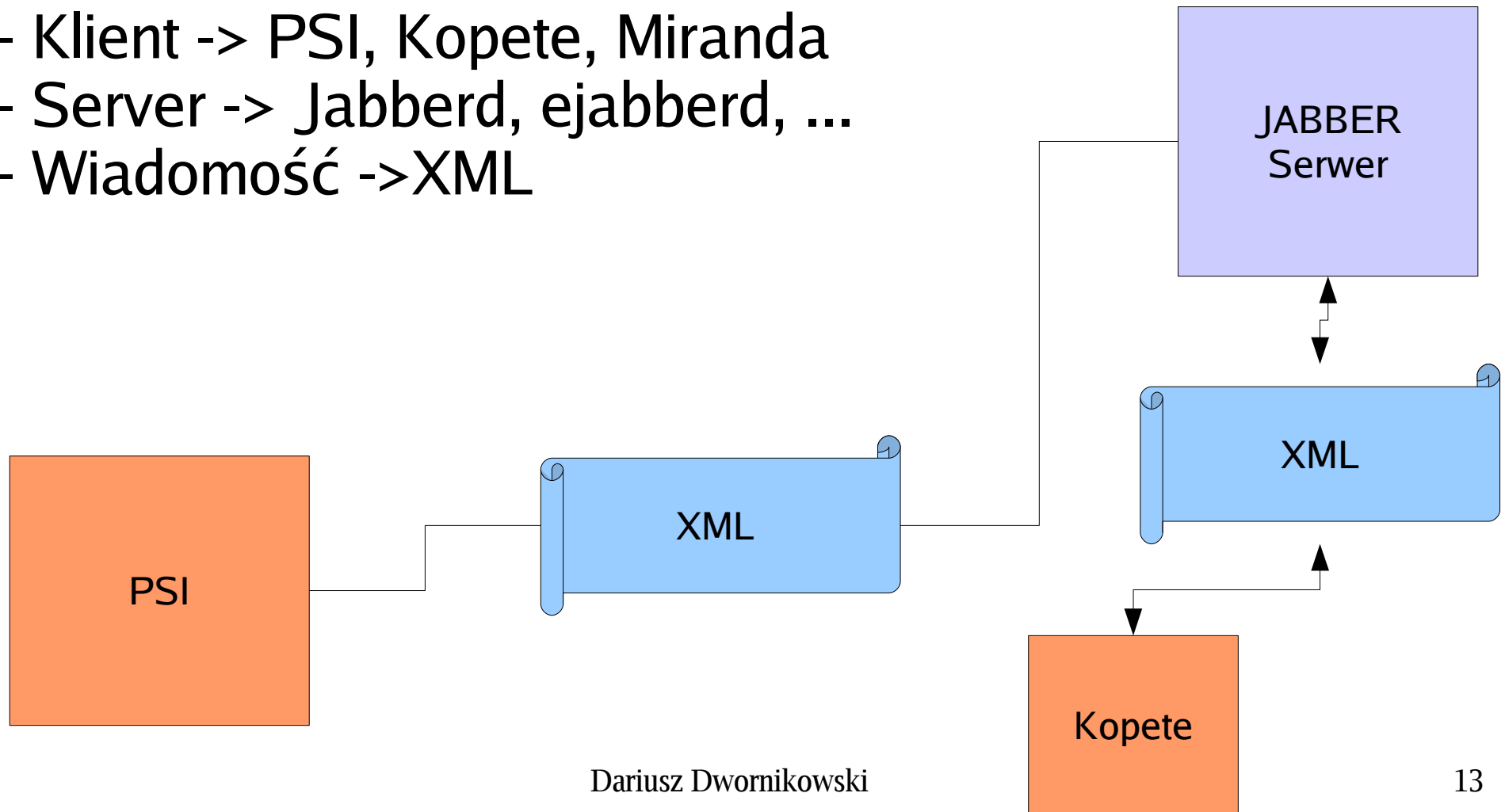
- Klient -> Przeglądarka WWW
- Server -> Apache
- Wiadomość -> HTML/js



# Web Services - dziś

Znany wszystkim Jabber, także działa jak Web Service :

- Klient -> PSI, Kopete, Miranda
- Server -> Jabberd, ejabberd, ...
- Wiadomość -> XML

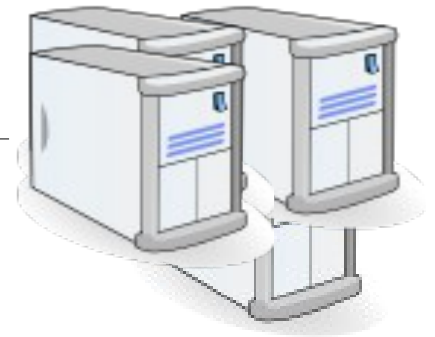


# Przykład – Serwis zakupowy

Portal e-MEGAMARKET

Katalog sklepów

Klient – zakup laptopa



Sklep z najniższą ceną

Hurtownia/Producent



BANK



# Web Services – pojęcia kluczowe

- **WSDL** ( Web Services Description Language) – abstrakcyjny język oparty na XML opisujący interfejs Web Service'u
- **UDDI** ( Universal description, discovery and integration ) - standard katalogowania web services, sposób na “wystawianie” ich na świat
- **SOAP** – protokół wymiany danych pomiędzy web service a klientem

# WSDL

Dokument WSDL składa się z trzech głównych komponentów:

- **definicje typów danych** ( Data type definitions )
- **definicji abstrakcyjnych metod** (Abstract operations)
- **dowiązań do usług** ( Service bindings)

Metody abstrakcyjne mapowane są na specyficzne protokoły transportu:

- HTTP
- SMTP
- SOAP RPC
- ...

# WSDL

...a teraz prościej...

**WSDL** można porównać do instrukcji obsługi, która daje programiście pojęcie o tym co oferuje dana **usługa**.

# WSS – Web Service Security

- Rozszerzenie Web Services o mechanizmy bezpieczeństwa
- Integracja z już istniejącymi rozwiązaniami (Kerberos)
- Rozszerzenia wprowadzane w WSDL
- Bezpieczeństwo w fazie projektowania

# SOAP

**SOAP** – to standard (protokół) przesyłania danych pomiędzy klientem, a web servicem.

POST /InStock HTTP/1.1

Host: www.example.org

Content-Type: application/soap+xml; charset=utf-8

Content-Length: nnn

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

```
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
```

```
  <soap:Body xmlns:m="http://www.example.org/stock">
```

```
    <m:GetStockPrice>
```

```
      <m:StockName>IBM</m:StockName>
```

```
    </m:GetStockPrice>
```

```
  </soap:Body>
```

```
</soap:Envelope>
```

# SOAP

Odpowiedź serwera WS:

HTTP/1.1 200 OK

Content-Type: application/soap+xml; charset=utf-8

Content-Length: nnn

<?xml version="1.0"?>

<soap:Envelope

xmlns:soap="http://www.w3.org/2001/12/soap-envelope"

soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/stock">

<m:GetStockPriceResponse>

<m:Price>34.5</m:Price>

</m:GetStockPriceResponse>

</soap:Body>

</soap:Envelope>

# UDDI

- Katalogowanie ( directory )
- Upublicznianie
- Efektywne wyszukiwanie (discovery)
- Charakter rozproszony ( jak LDAP )
- Zorientowane na integrację biznesu
- Analogia : Yellow Pages, LDAP

# UDDI

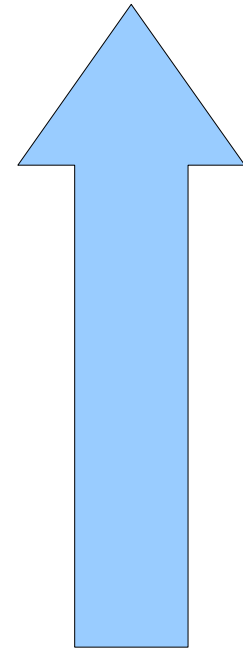
Sam **Web Service** jest niczym, kiedy nie można go udostępnić innym.

**UDDI** będzie ewaluowało.

Połączenie z katalogami ebXML.

# Wpływ WS na biznes

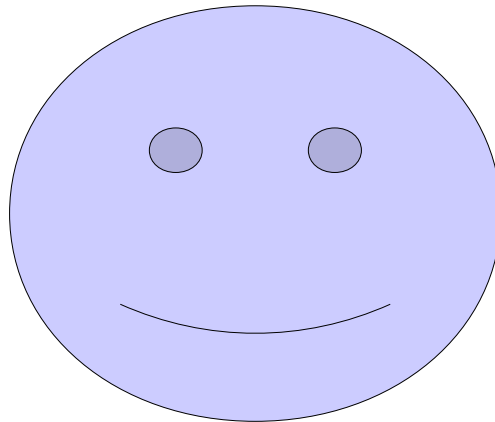
- Rozwój usług pośredniczących
- Portale “full service”
- Portale doradcze / pomocnicze
- Portale wyszukiwania
- Giełdy



# Wpływ WS na biznes

- Oszczędność czasu klienta = zadowolenie

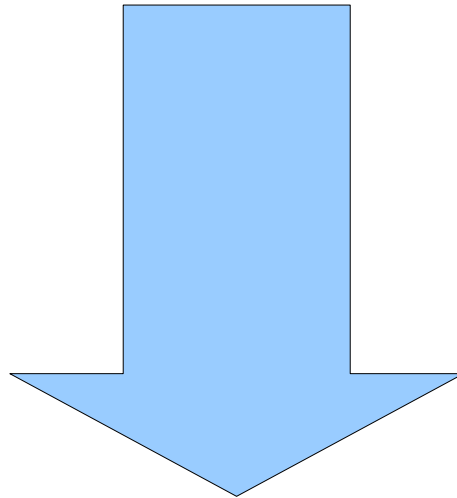
Klient to z natury “leń”, woli pełną obsługę, wtedy jest zadowolony.



# Wpływ WS na biznes

- Niższe koszty korzystania z serwisów

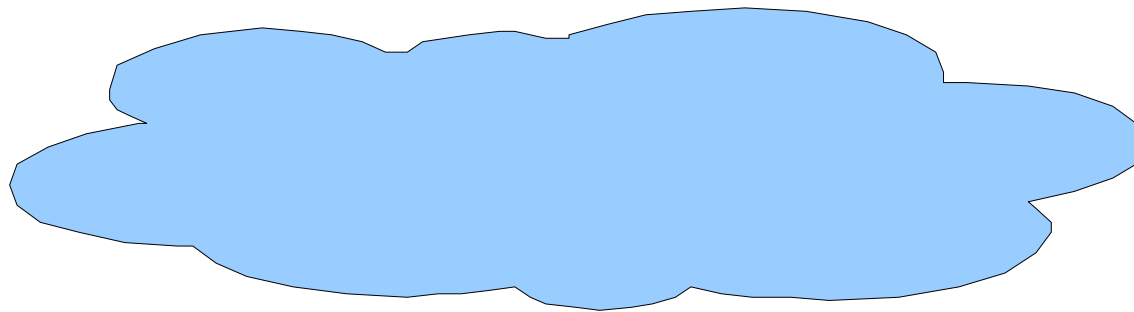
Integracja obniży koszty, co zapewne istotnie wpłynie na obniżenie cen produktów oraz usług w internecie oraz w świecie rzeczywistym.



# Wpływ WS na biznes

- Korporacje i rozwój B2B

Firmy o wspólnych celach połączą się, by zwiększyć dochody.



# Potrzeba mocy i łączenia

- Rozwój biznesu i nauki potrzebuje mocy obliczeniowej
- Nowy rodzaj usług – “service outsourcing”
- Zaawansowane usługi budowane z puzzli

Wystarczy połączyć wiele pomniejszych usług w jedną całość, by uzyskać nową funkcjonalność.

# Przykład Service Outsourcing

- Portal gazetowy

Klient wybiera sobie profile informacji.

Portal płaci za możliwość korzystania z web serviców danej gazety jej wydawcom.

Odsprzedaje to w nowej formie – multi gazety.

# Web Services – furтка dla GRIDów

Drastyczny boom usługowy może nie mieć wystarczającej ilości mocy by sprostać wymogom internetu.

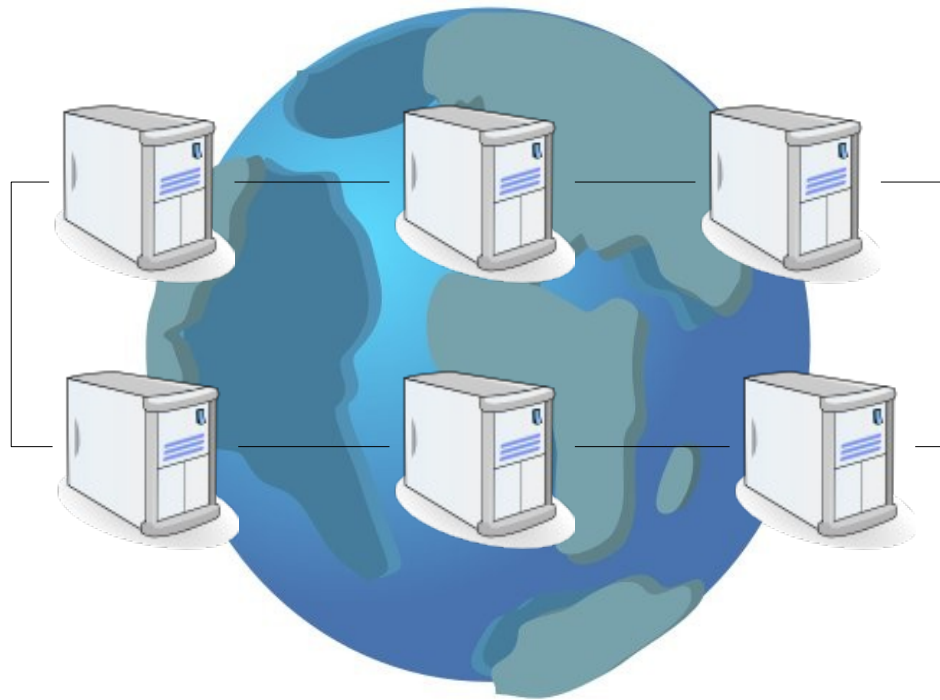
Potrzeba taniego, wydajnego rozwiązania.

GRID

# GRID

**System rozproszony**, składający się z wielu heterogenicznych węzłów, udostępniających zasoby.

Moc w gniazdku.



# GRID

- Grid Services opierają się na idei Web Services
- Zupełna przezroczystość
- Zasoby za pieniądze
- Gridy i WS są naturalną odpowiedzią na potrzeby ludzkości
- GRID ma być dostępny jak prąd w gniazdku

# GRID – wszechobecna intersieć

- Natychmiastowa dostępność
- Wysoka wydajność - nieosiągalna w inny sposób
- Wiele usług w jednym miejscu
- Ponad granicami
- Z GRID skorzystają wszyscy

# GRID – pola użyteczności

- Nauka ( wspólna praca )
- Edukacja ( wirtualne szkolenia )
- Wojsko/Rządy ( problemy różnego rodzaju, prognozy )
- Korporacje ( moc obliczeniowa, modelowanie )
- Klienci końcowi (business everywhere, virtual reality )
- Ludzkość jako ogół (globalna wioska)

# GRID – to nie fikcja



- OGSA ( Open Grid Services Architecture )
- Architektura oparta na Web Services
- Implementacja wyznacza standardy
- Dostępna za darmo
- Globus Toolkit

# Platformy – Web Services

- JAVA ( j2ee )
- .NET
- APACHE AXIS (Tomcat, Jboss, IBM WScE)
- Perl, Python, C++, Ruby, PHP
- Mono (??)
- Wiele innych .. ( łatwość implementacji )

# Podsumowanie – Web Services

- Wysoki poziom abstrakcji
- Technologie ukierunkowane na użyteczność
- Łatwość w implementacji usług
- Rozwój biznesu
- Standardy
- Integracja

# Podsumowanie - wady

- Asynchroniczny charakter
- Rozwiązania Open Source są w fazie niemowlęcej
- Dość duża moc obliczeniowa
- Koszt profesjonalnych rozwiązań

# Linki

- <http://www.oasis-open.org/>
- <http://www.w3.org/2002/ws/>
- <http://msdn.microsoft.com/webservices/>
- [www-128.ibm.com/developerworks/webservices/](http://www-128.ibm.com/developerworks/webservices/)
- <http://www.globus.org/ogsa/>

Dziękuję za uwagę !!