Inżynieria Oprogramowania

Architektura systemów informatycznych

Wydział Matematyki, Informatyki i Mechaniki
Uniwersytet Warszawski
www.mimuw.edu.pl/~dabrowski

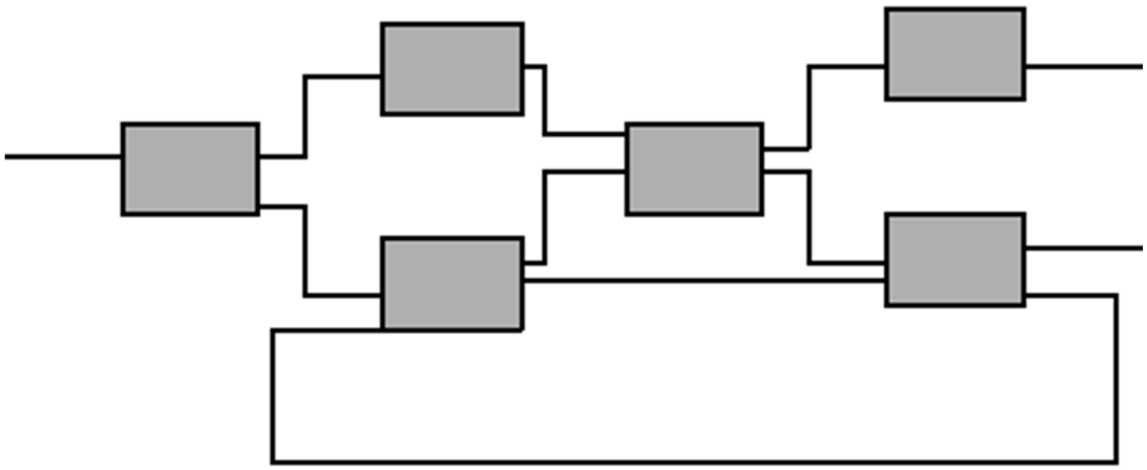
Błędne interpretacje

- *Architecture is just paper*
- *Architecture and design are the same things*
- *Architecture and infrastructure are the same things*
- *<my favorite technology> is the architecture*
- *A good architecture is the work of a single architect*
- *Architecture is simply structure*
- *Architecture can be represented in a single blueprint*
- *System architecture precedes software architecture*
- *Architecture cannot be measured or validated*
- *Architecture is a science*
- *Architecture is an art*

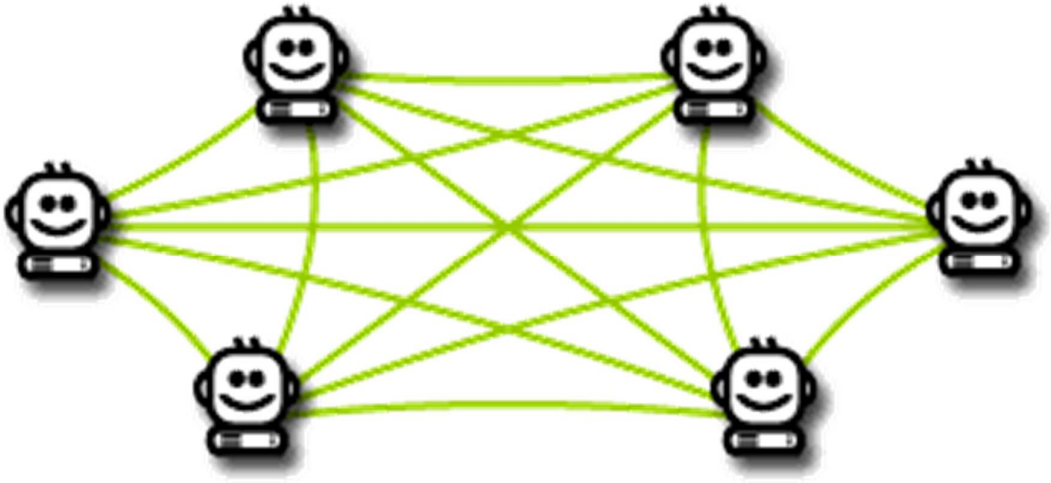
Przykłady architektur

- Przykłady

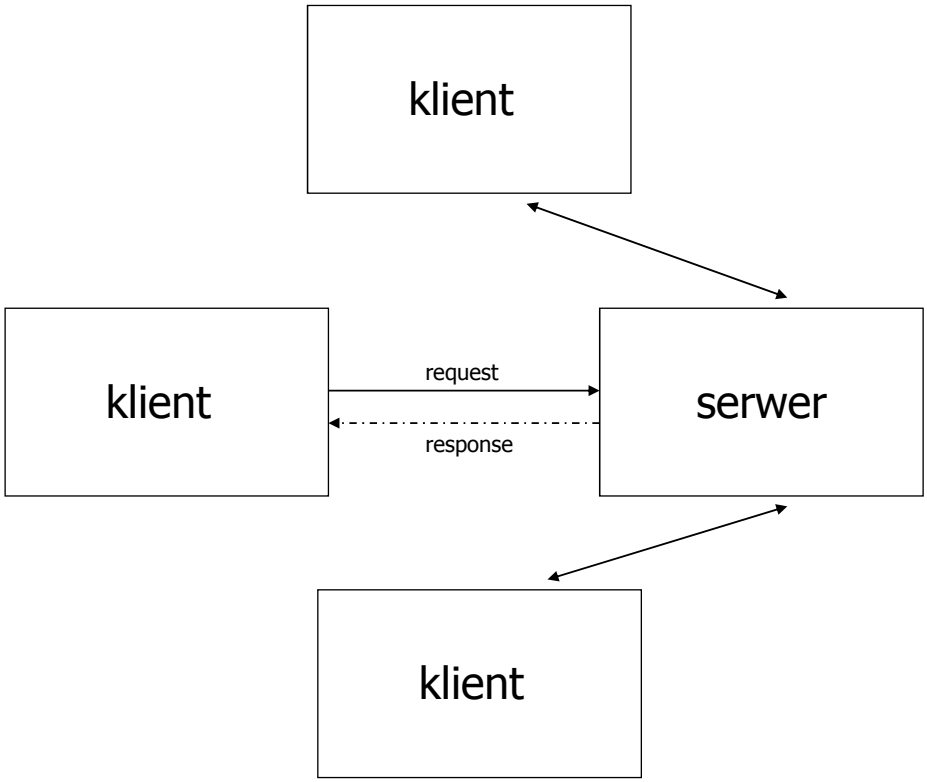
Pipe–and–filter



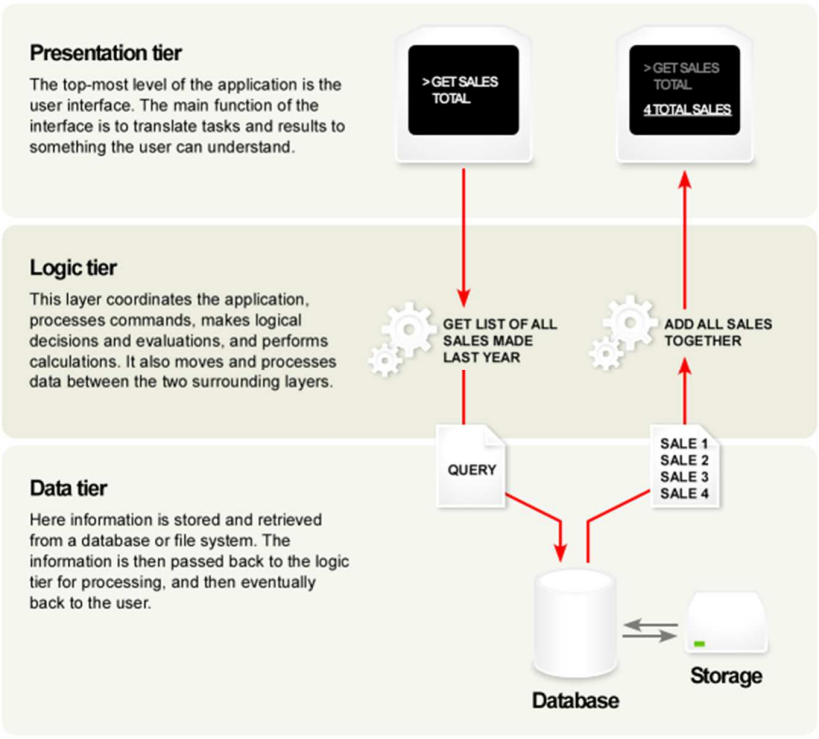
Peer-to-peer



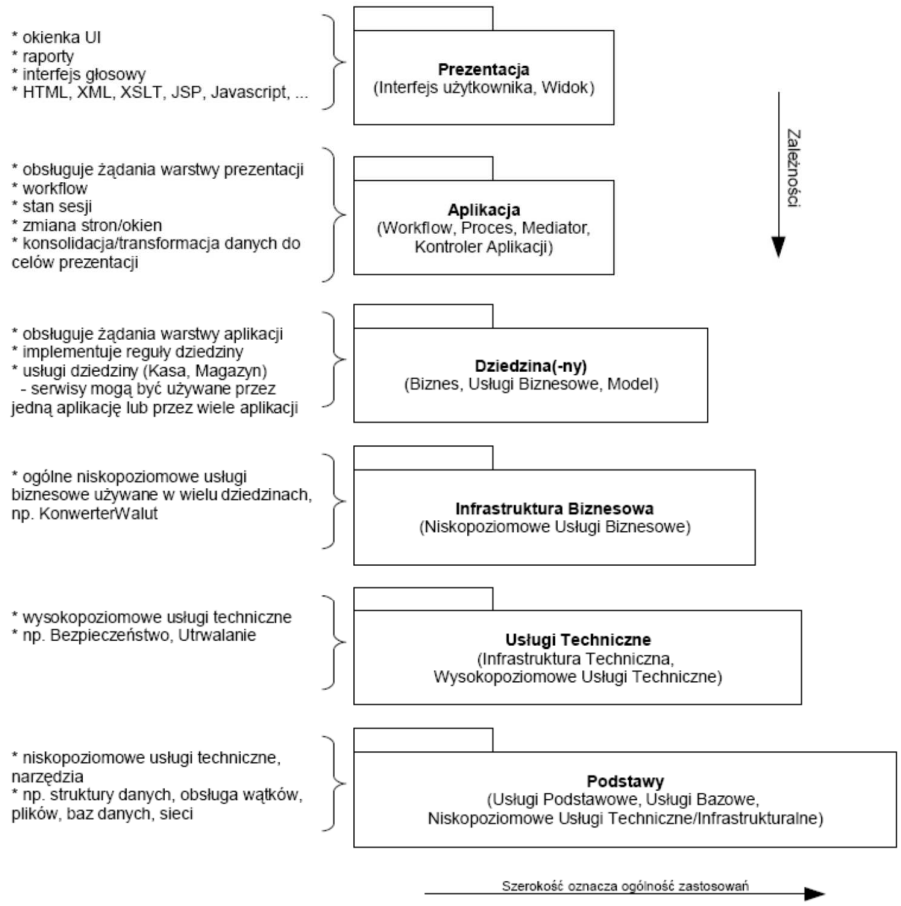
Client–server



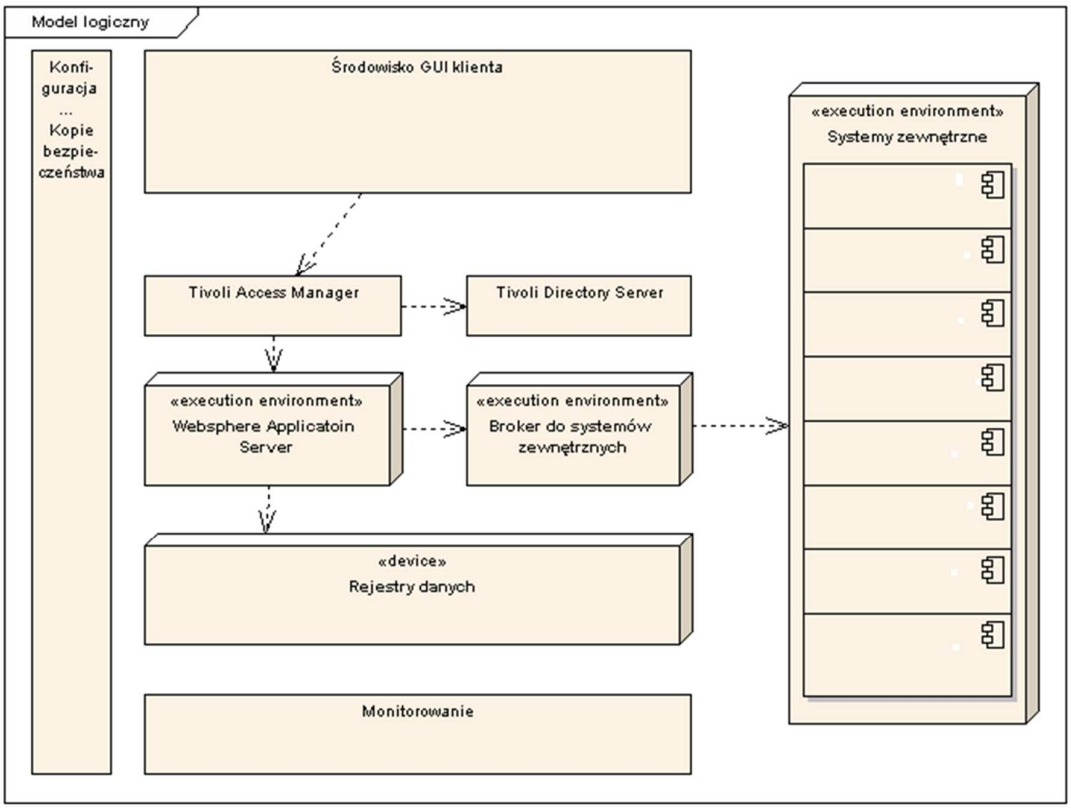
Three–tier

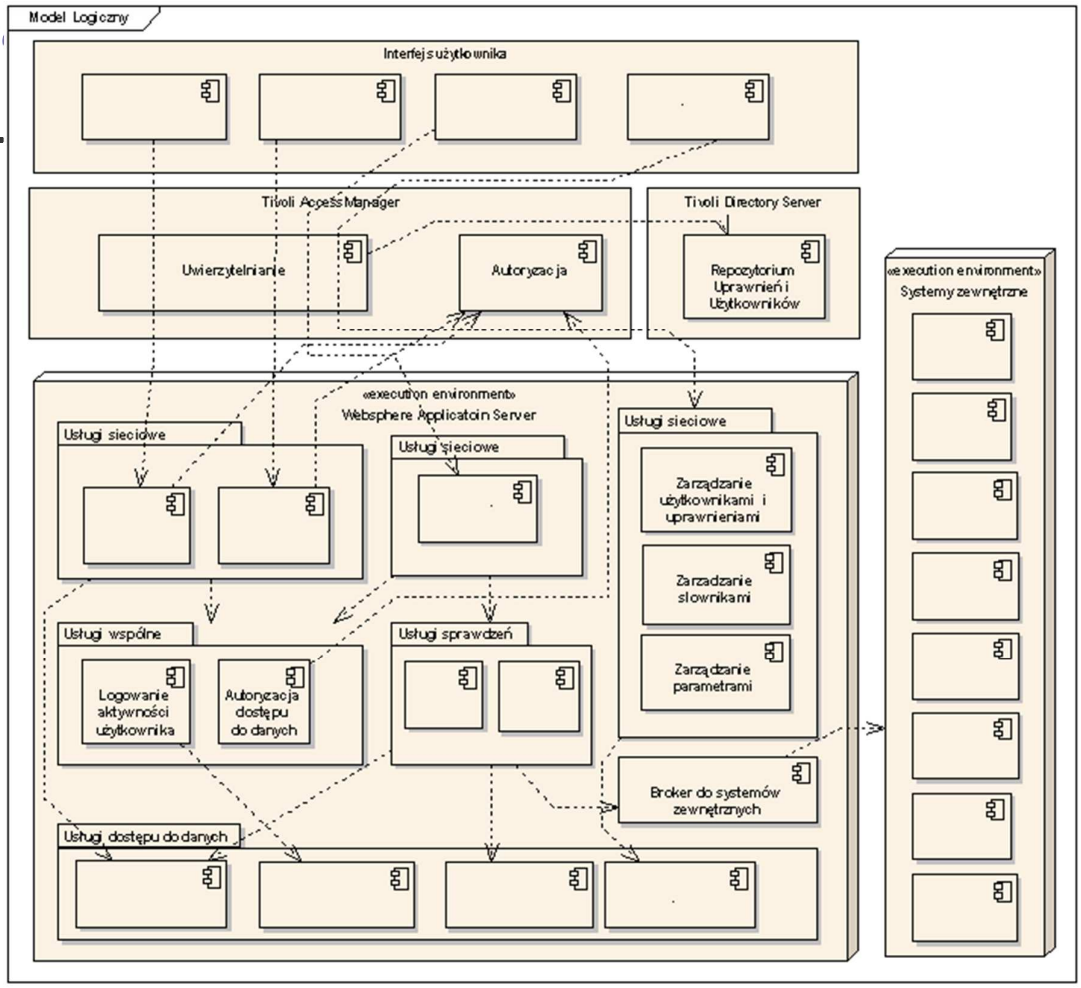


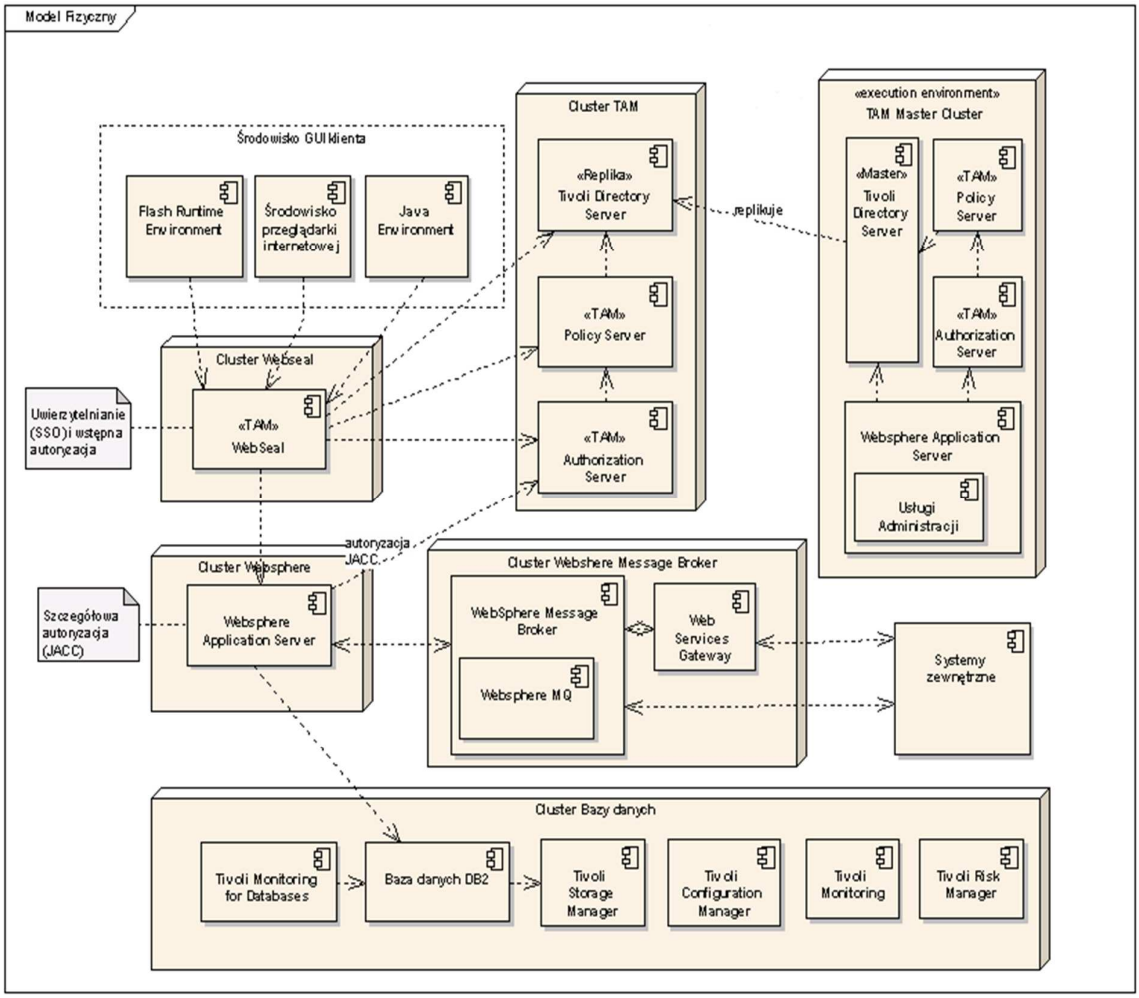
N–tier



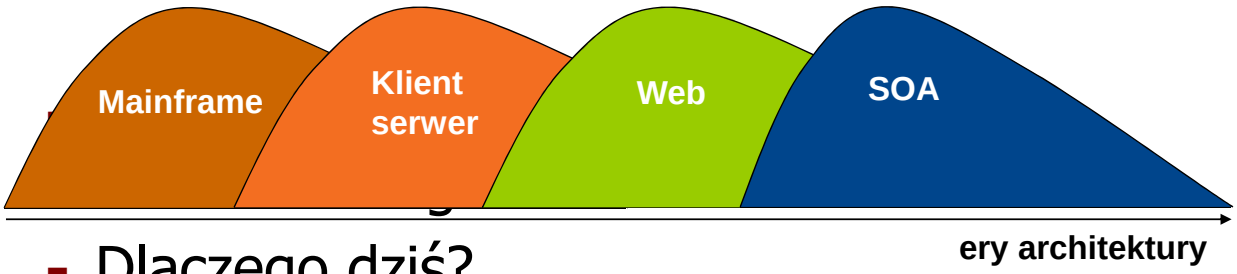
Przykład


Model logiczny
Konfi-
guracja
...
Kopie
bezpie-
czeństwa
Środowisko GUI klienta
Tivoli Access Manager
Tivoli Directory Server
«execution environment»
Websphere Applicatoin
Server
«execution environment»
Broker do systemów
zewnętrznych
«device»
Rejestry danych
Monitorowanie
«execution environment»
Systemy zewnętrzne


Model Logiczny
Interfejs użytkownika
Tivoli Access Manager
Uwierzytelnianie
Autoryzacja
Tivoli Directory Server
Repozytorium
Uprawnień i
Użytkowników
«execution environment»
Systemy zewnętrzne
«execution environment»
Websphere Applicatoin Server
Usługi sieciowe
Usługi sieciowe
Usługi sieciowe
Zarządzanie
użytkownikami i
uprawnieniami
Zarządzanie
słownikami
Zarządzanie
parametrami
Usługi wspólne
Logowanie
aktywności
użytkownika
Autoryzacja
dostępu
do danych
Usługi sprawdzeń
Broker do systemów
zewnętrznych
Usługi dostępu do danych

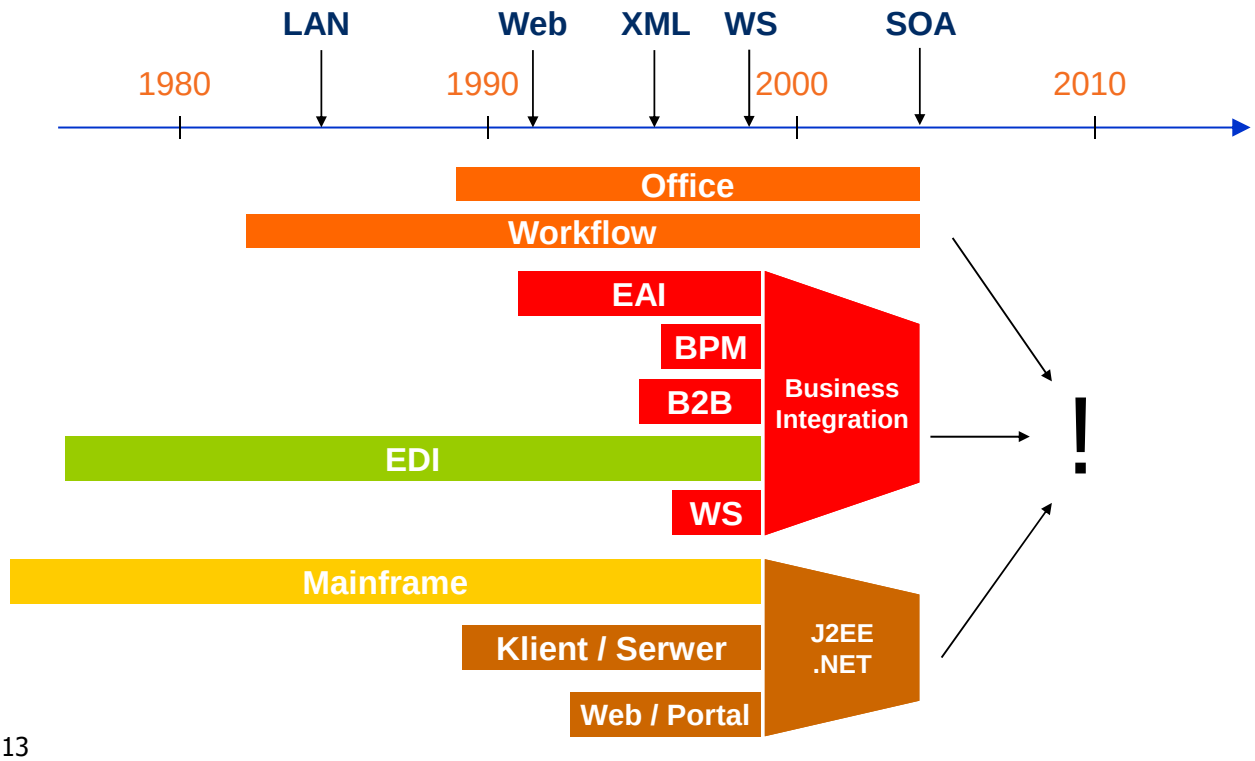


Dziś od „functionality” ważniejsze „connectivity”



- Dlaczego dziś?
  - Osiągamy przełom technologiczny pomiędzy
    - Mocą obliczeniową
  - Oraz
    - Możliwościami łączności

Jak dotarliśmy do architektury zor. na usługi?



Service-oriented architecture

- Od rzemieślników do dostawców usług
  - Czyli jak dotarliśmy do architektury zorientowanej na usługi?
- Świat połączony
  - Czyli co dokładnie kryje się pod pojęciem SOA?
- Droga ku SOA
  - Czyli czy wdrożenie SOA to tylko zakup systemów IT?

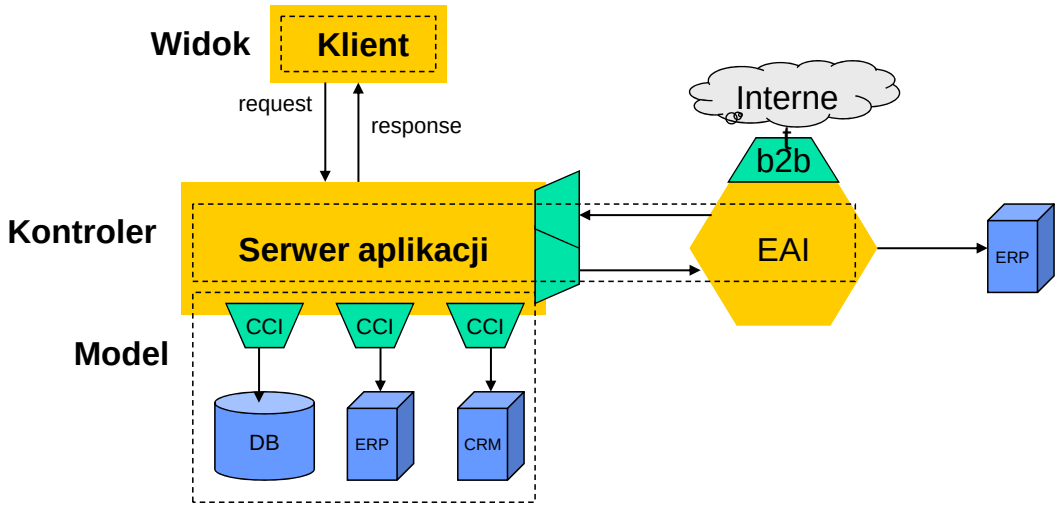
Usługi w świecie rzeczywistym

- Dobrze zdefiniowane
  - Łatwe w użyciu, ustandaryzowane interfejsy
- Samowystarczalne
  - Brak widocznych zależności od innych usług
- Zawsze i łatwo dostępne
  - Ale brak aktywnego oczekiwania
- Niezależne od kontekstu konsumenta
  - Uruchamiane w kontekstach nie przewidzianych
- Bardzo konkurencyjne
  - Konkurują zakresem, nie sposobem implementacji
- Można łączyć
  - Istniejące usługi w nową usługę

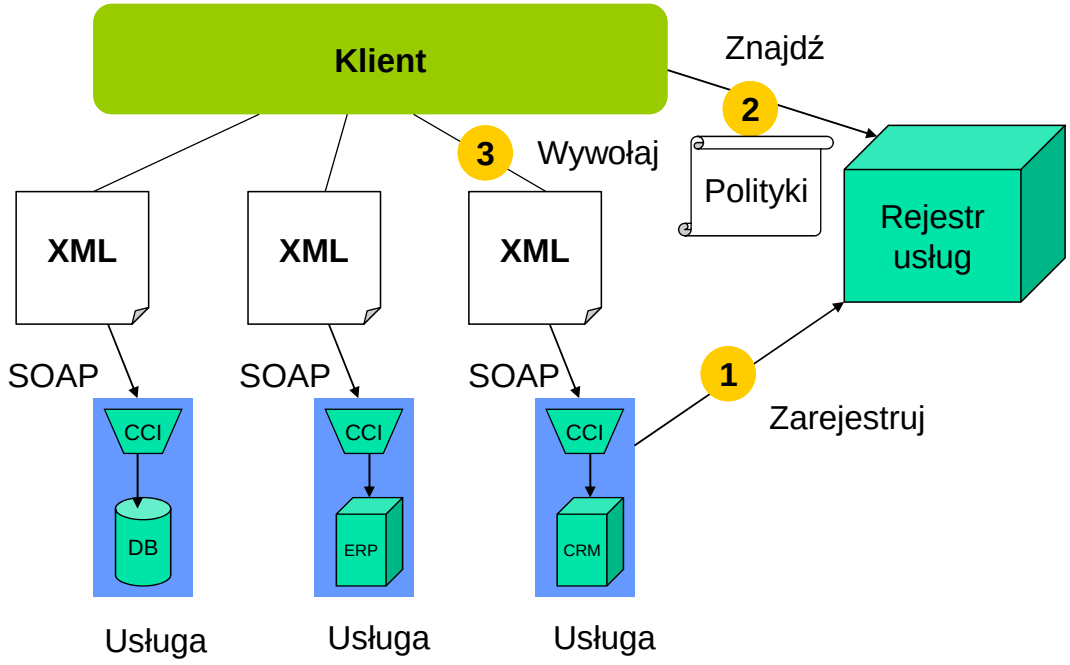
Konsumenci, dostawcy, brokerzy

- Konsument
  - Wyraża zamiar
- Dostawca
  - Udostępnia ofertę
- Broker
  - Znajduje najlepszą odpowiadającą ofertę
  - Reklamuje oferty na różne sposoby
    - Odpowiadające różnym zamiarom
  - Zapewnia „usługę katalogową” („yellow pages”)
- A gdyby korzystanie z usług wymagało wskazywania
  - Dokładnych współrzędnych geograficznych usługi?
  - Dokładnego adresu, numer telefonu lub faksu usługi?

Web: Model, widok, kontroler (J2EE, .Net, ...)



SOA: komponent usługą działającą poza klientem



SOA vs Komunikaty

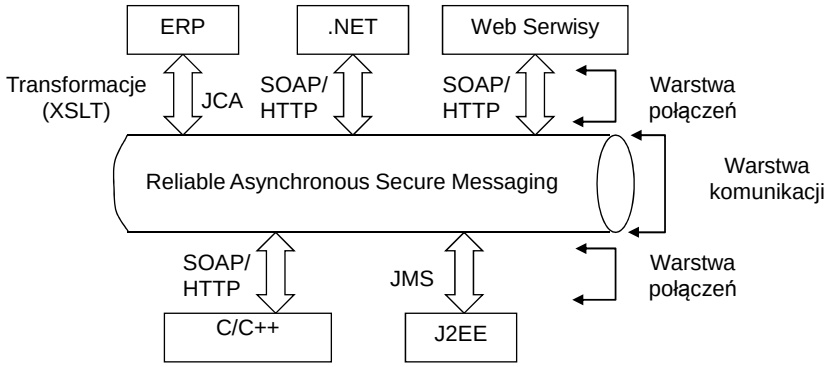
- Usługami sterują komunikaty
  - Usługi jako „czarne skrzynki”
  - Komunikaty wchodzące i wychodzące
  - Pozostałe elementy to (nieistotne) szczegóły implementacyjne
- Różne warstwy transportowe komunikatów
  - Email, IP, TCP/IP, HTTP, Web-service, MSMQ, MQ-Series, …
- Różne struktury komunikatów
  - XML, Binary, …
- Czy dopuszczamy utraty komunikatów
  - Komunikaty idempotentne; lub
  - Komunikacja niezawodna

SOA vs Web Serwisy

- Otwarty standard oferowania usług
- Możliwość wymiany dokumentów strukturalnych
  - Różny zestaw informacji, metadane (informacje o informacji)
- Luźne powiązanie
  - Niewielkie nakłady na konsumpcję usługi
  - Przyjmowanie komunikatów nie w pełni rozumianych
    - XML, WSDL
  - Późne wiązanie
    - Lokalizacja niezależna od mechanizmu wołania + katalogi usług
- Możliwe interakcje peer-to-peer
  - Request / response
    - Niewydajny i ograniczający model interakcji
  - Wzrost poziomu granularności
    - Zacierają się granice między klientem a serwerem

SOA vs ESB

- Kręgosłup SOA
  - Działa jako broker komunikatów
  - Zapewnia system kolejkowania komunikatów
  - Standard przemysłowy wymiany komunikatów
    - Np. SOAP lub JMS
- Zapewnia współpracę aplikacji wysokopoziomowych z niskopoziomowymi poprzez standardowe adaptery i interfejsy



SOA vs BPM

- SOA
  - Konstruowanie komponentów oprogramowania które mogą być re-używane w kontekście nieznanym w momencie ich projektowania
  - Kompozycja vs Rozszerzanie (OO)
- BPM
  - Zdolność do
    - Precyzyjnego modelowania
  - Oraz
    - Zmiany
  - Kontekstu w którym komponenty (przedsiębiorstwa) są wykorzystywane

Przejście na architekturę zorientowaną usługowo

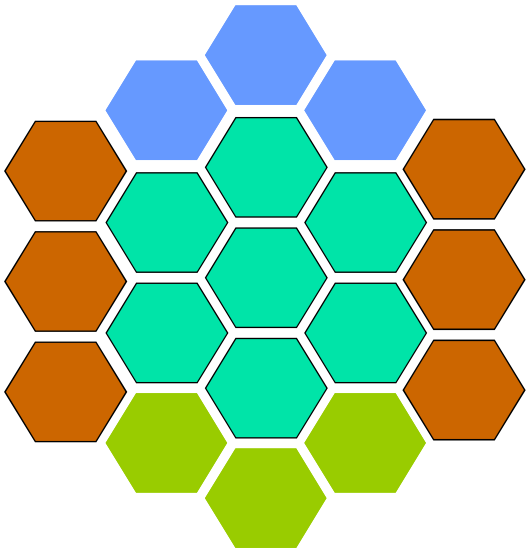
- **Zorientowane na funkcje**
- **Zbudowane by trwać**
- **Wydłużony cykl wytwórczy**

- **Zorientowane na koordynację**
- **Zbudowane by się zmieniać**
- **Budowane i wdrażane przyrostowo**

- **Silosy aplikacji**
- **Ściśle powiązane**
- **Zorientowane obiektowo**
- **Predefiniowana technologia**

- **Rozwiązania dla przedsiębiorstw**
- **Luźno powiązane**
- **Zorientowane na komunikację**
- **Abstrakcyjne**

Budowanie rozwiązań SOA



Logika biznesowa przedsiębiorstwa
Model biznesowy

Globalna logika biznesowa
Podatki, handel, …

Koordynacja biznesu (metadane)
Transakcje, procesy, …

Interfejs użytkownika

Dotarcie do SOA

- Osiągnięcie przez organizację kontroli
  - Nad zasobami IT
- Zorganizowanie logiki biznesowej
  - W sposób niezależny od kontekstu jej wykorzystania
- Ponowna implementacja warstwy kontrolera
  - Wykorzystanie technologii „koordynacji”
    - Orkiestracja
    - Choreografia
- Stworzenie nowej warstwy widoku

- Ale
  - Wymagane dobre rozumienie własnego biznesu
  - Jest to inwestycja (usług >= procesów)
  - Może być dużym narzutem dla małych przedsiębiorstw

Kiedy nie warto wdrażać SOA?

- Stabilne, homogeniczne środowiska IT
  - SOA może nie być istotne
  - SOA może być nieefektywne kosztowo
- Organizacja nie oferuje i nie wykorzystuje usług opartych na IT
  - Nie ma potrzeby zapewnienia elastyczności
- Systemy czasu rzeczywistego
  - SOA bazuje na luźno powiązanej komunikacji asynchronicznej

Podsumowanie

- Zorientowanie na usługi to nowy paradygmat obliczeniowy
- To nie jest nowa nazwa
  - API
  - Komponentu
- To jest nowa koncepcja konstrukcji oprogramowania
  - SO to zmiana podejścia porównywalna (jeśli nie większa) do OO
  - SO wymusza postrzeganie oprogramowania przez pryzmat
    - Większej ilości konsumentów
    - Nieznanego kontekstu

Podsumowanie

- SOA jest ciągle przed nami: ewolucja, nie rewolucja
- Podnosi poziom abstrakcji i reużywalność systemów IT
- Możemy oczekiwać trendu do posiadania i publikowania usług
  - Już się zaznacza (np. Google)
  - Dotknie szczególnie aplikacji Mainframe i Klient/Serwer (ERP, CRM, ...)
- Oprogramowanie będzie bogatsze i sprytniejsze
  - Może stać się koszmarem, jeśli uwzględnimy ryzyka bezpieczeństwa
    - Których obecny poziom jest już wysoki
  - Nie wszystkie technologie są dojrzałe
    - Potrzeba zapewnienia zbieżności standardów