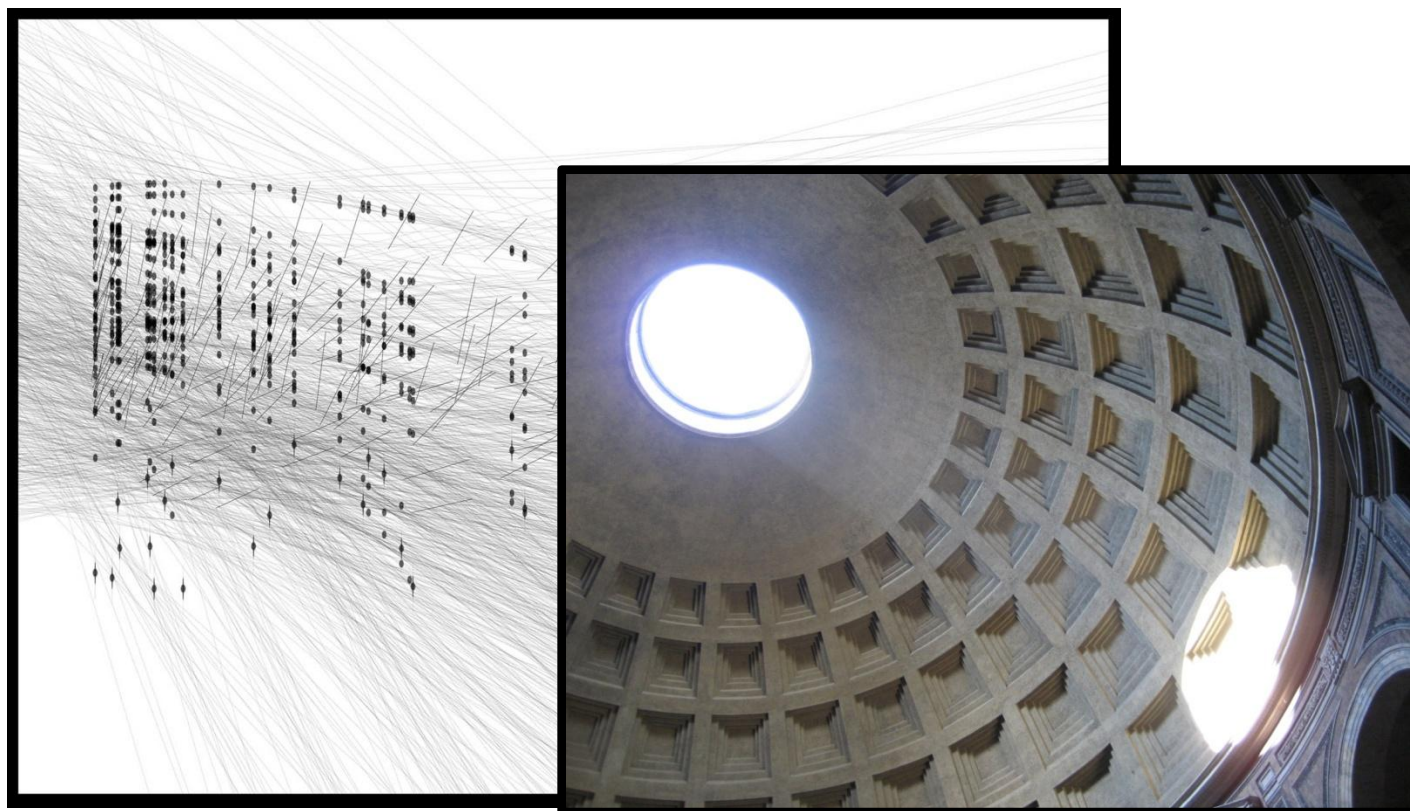


Architektura i integracja systemów.

Andrzej Ratkowski

Wykład 3. Metody wytwarzania architektury i wzorce architektoniczne.



Informacje organizacyjne

Andrzej Ratkowski, a.ratkowski@elka.

pok. 555, konsultacje: środy 17:00-18:00

strona: andrzejratkowski.blogspot.com

Zaliczanie: 2 kolokwia x 15 punktów +

Projekt: 30 punktów = 60 punktów

Kolokwia: **11 kwietnia, 6 czerwca**

Projekt podzielony jest na 2 etapy po 20 + 10 punktów

Ogłoszenie tematów: **14 marca**

Oddawanie: **E1 17 kwietnia, E2 6 czerwca**

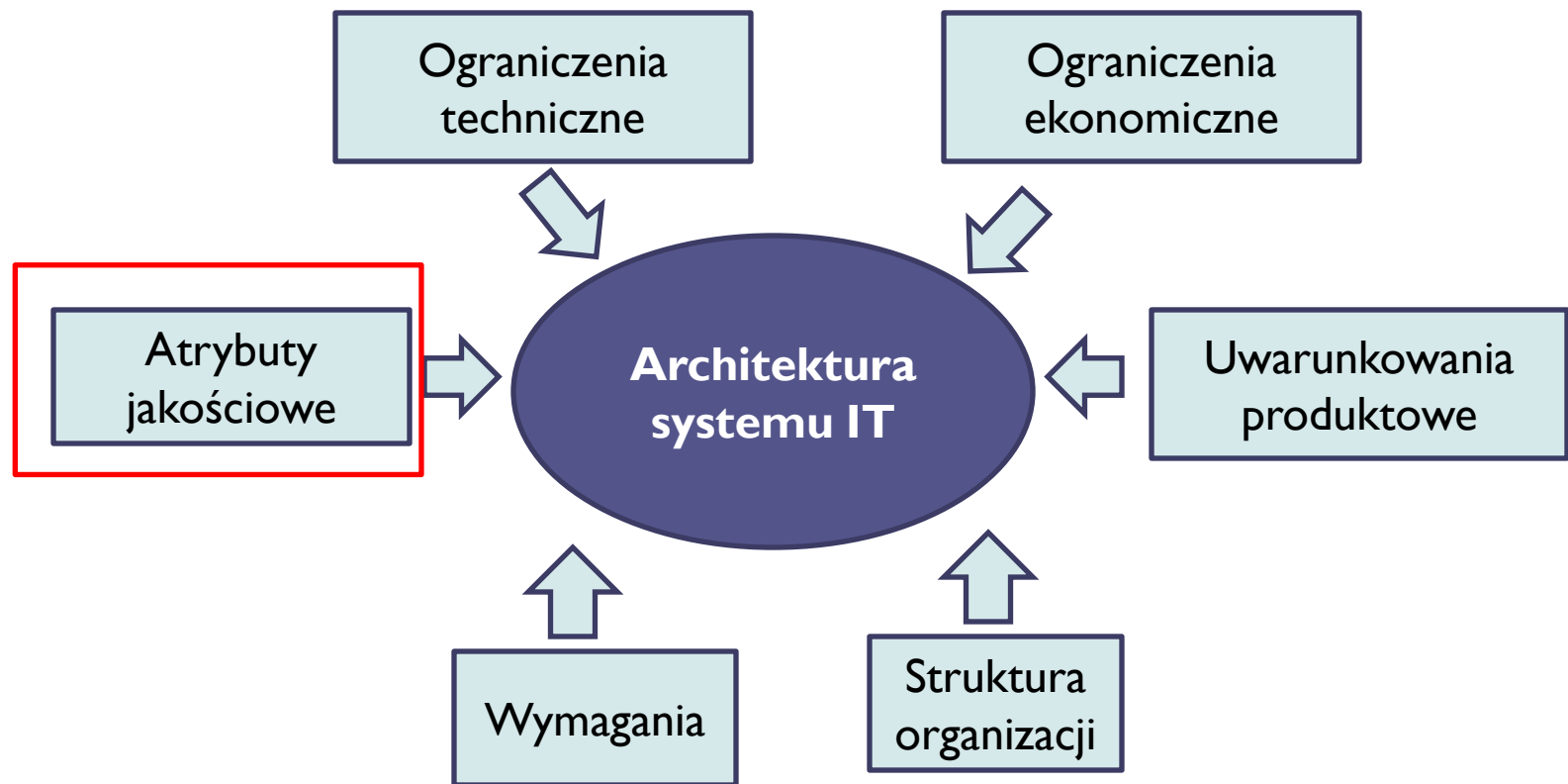


Architektura

Architektura programu lub systemu informatycznego to struktura lub struktury systemu na które składają się elementy oprogramowania, zewnętrzne właściwości tych elementów i relacje między nimi.

[Bass, Clements, Kazman]

Czynniki wpływające na architekturę



Atrybuty systemu

- Atrybuty jakościowe
 - Modyfikowalność / konserwowalność
 - Wydajność
 - Niezawodność
 - Bezpieczeństwo
 - Ergonomia
 - Interoperacyjność (ang. interoperability)
- Ponowne użycie komponentów
- „Time-to-market” / konstruowalność

Metody projektowania architektury

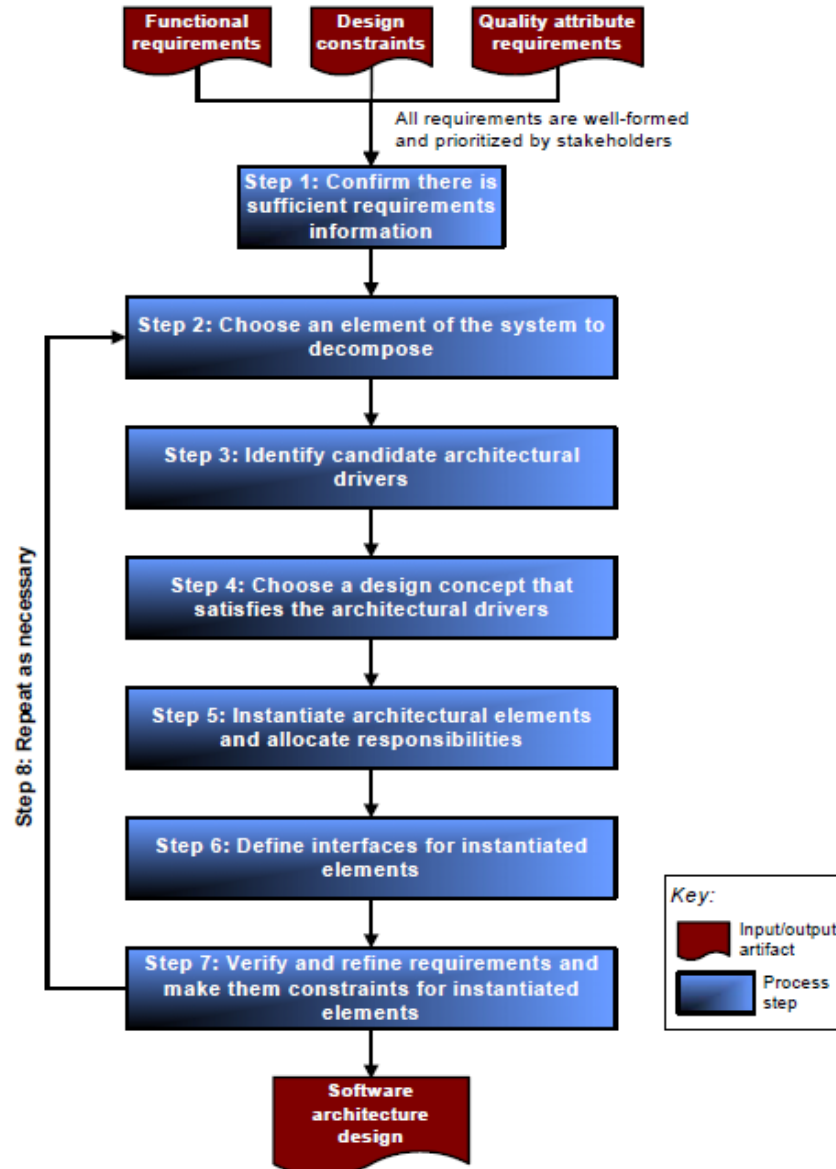
Przykłady

- M1 Attribute-Driven Design (ADD)
- M2 RUP 4+1 Views
- M3 Business Architecture Processes Organization (BAPO)

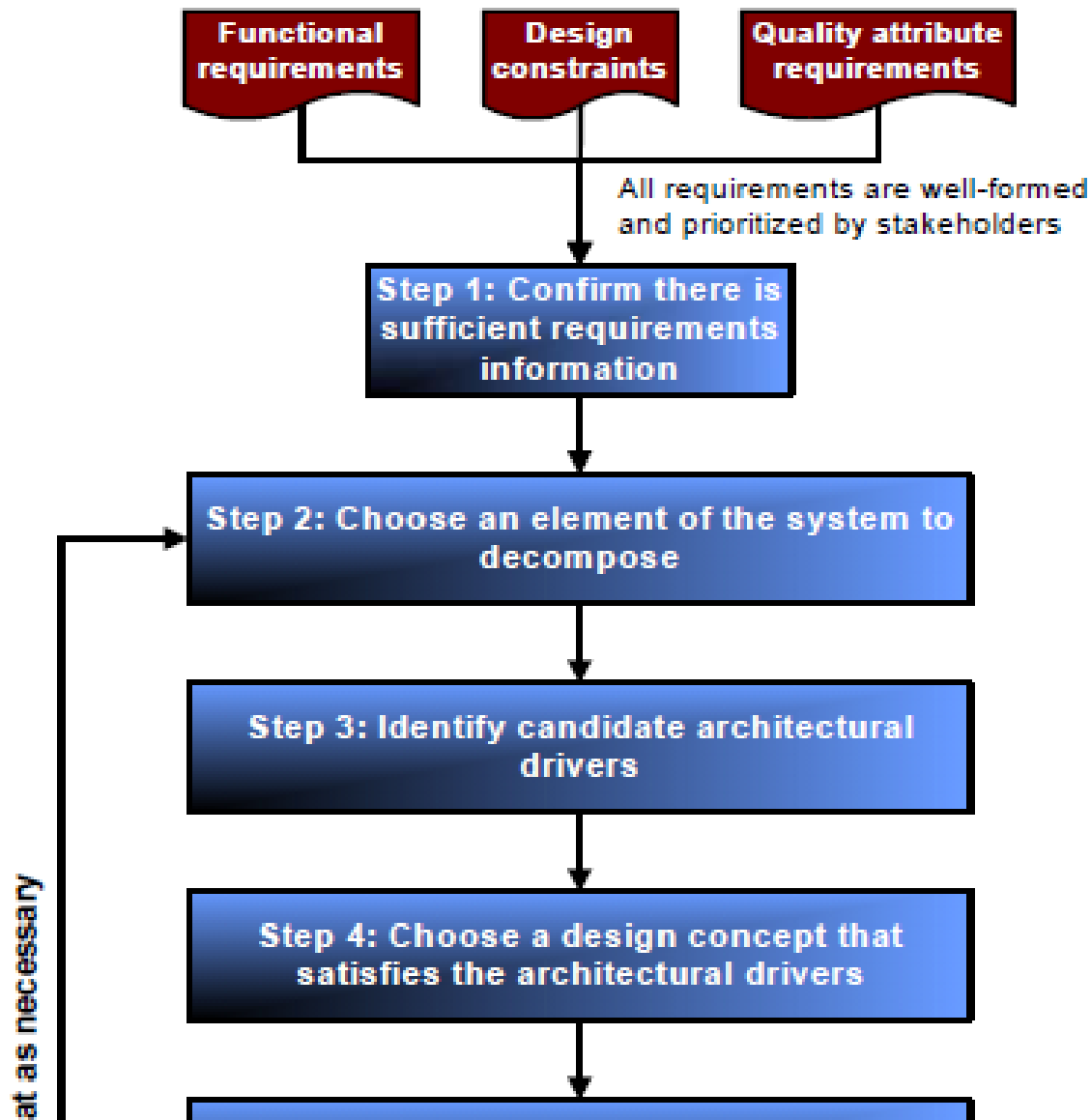
MI *Attribute-Driven Design – ADD*

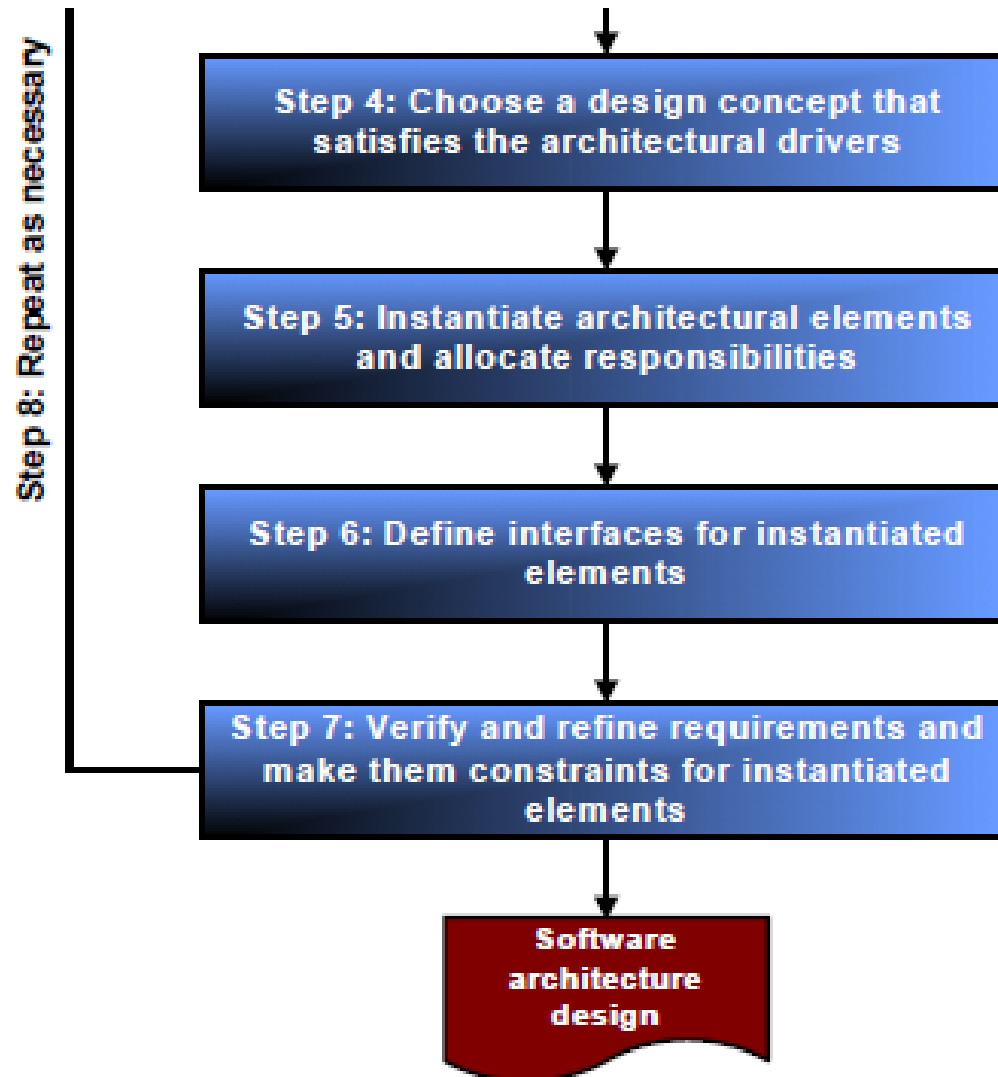
- Autor: Rob Wojcik SEI (Carnegie Mellon)
- Projektowanie architektury rozpoczyna się po wstępnej analizie wymagań – a więc na początku tworzenia systemu
- Należy zidentyfikować **czynniki kształtujące architekturę (architectural drivers)** – kluczowe wymagania funkcjonalne, jakościowe i biznesowe
- Rekurencyjne projektowanie architektury
- www.sei.cmu.edu/reports/06tr023.pdf

MI *Attribute-Driven Design – ADD*

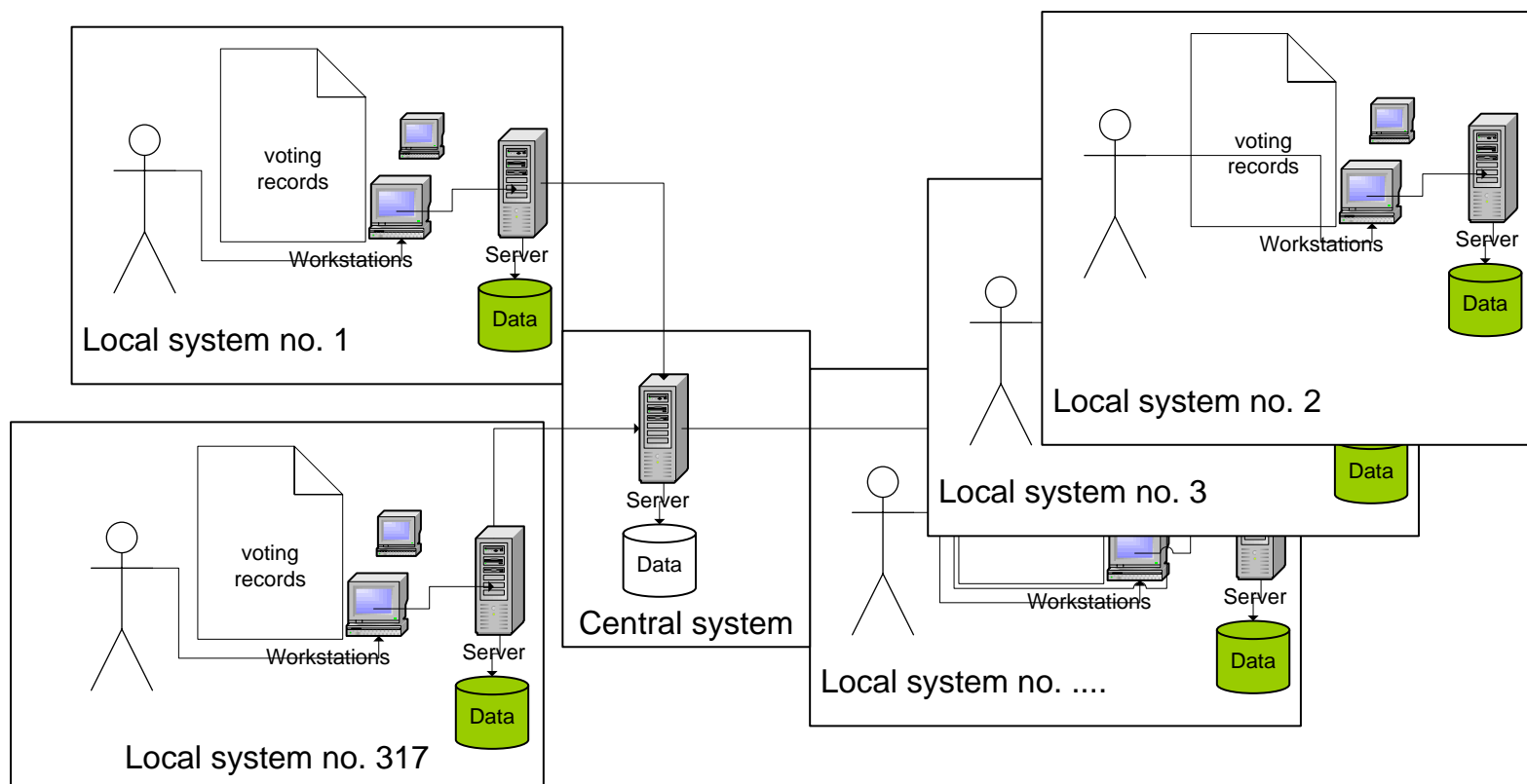


MI *Attribute-Driven Design – ADD*



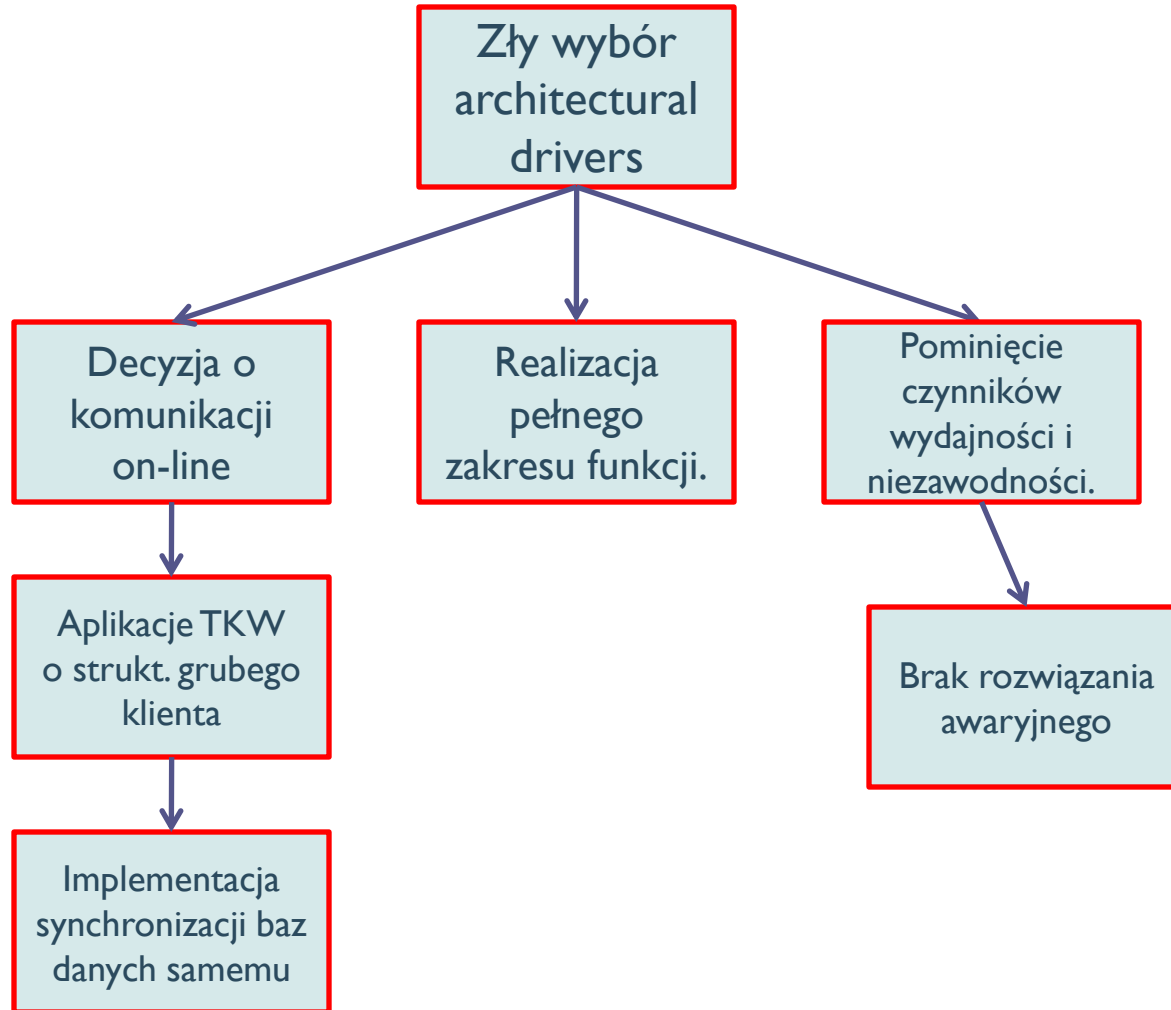


System obsługi wyborów, 2002.10.27 Architektura_(I)



Co się stało ?

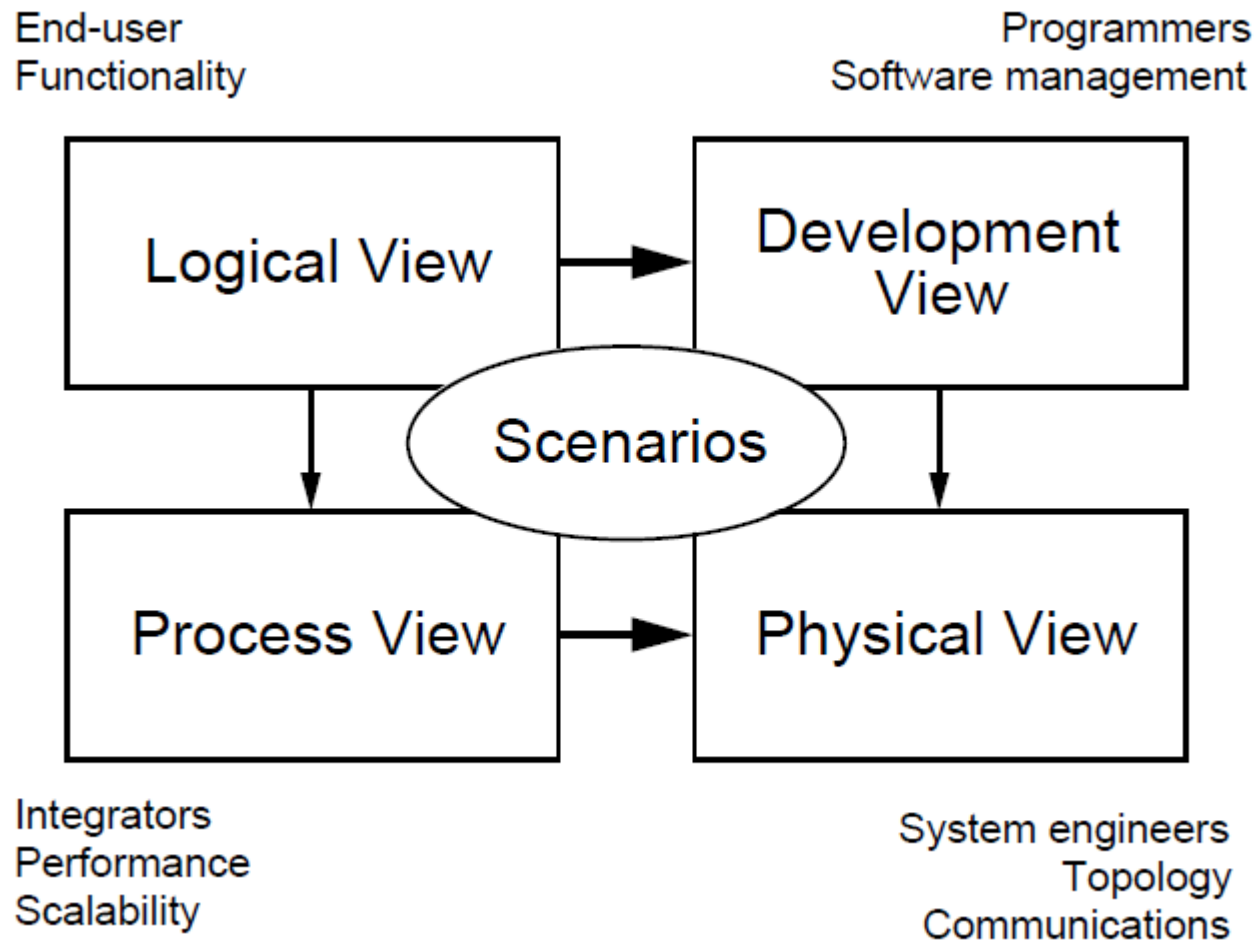
Jakie błędy popełniono?



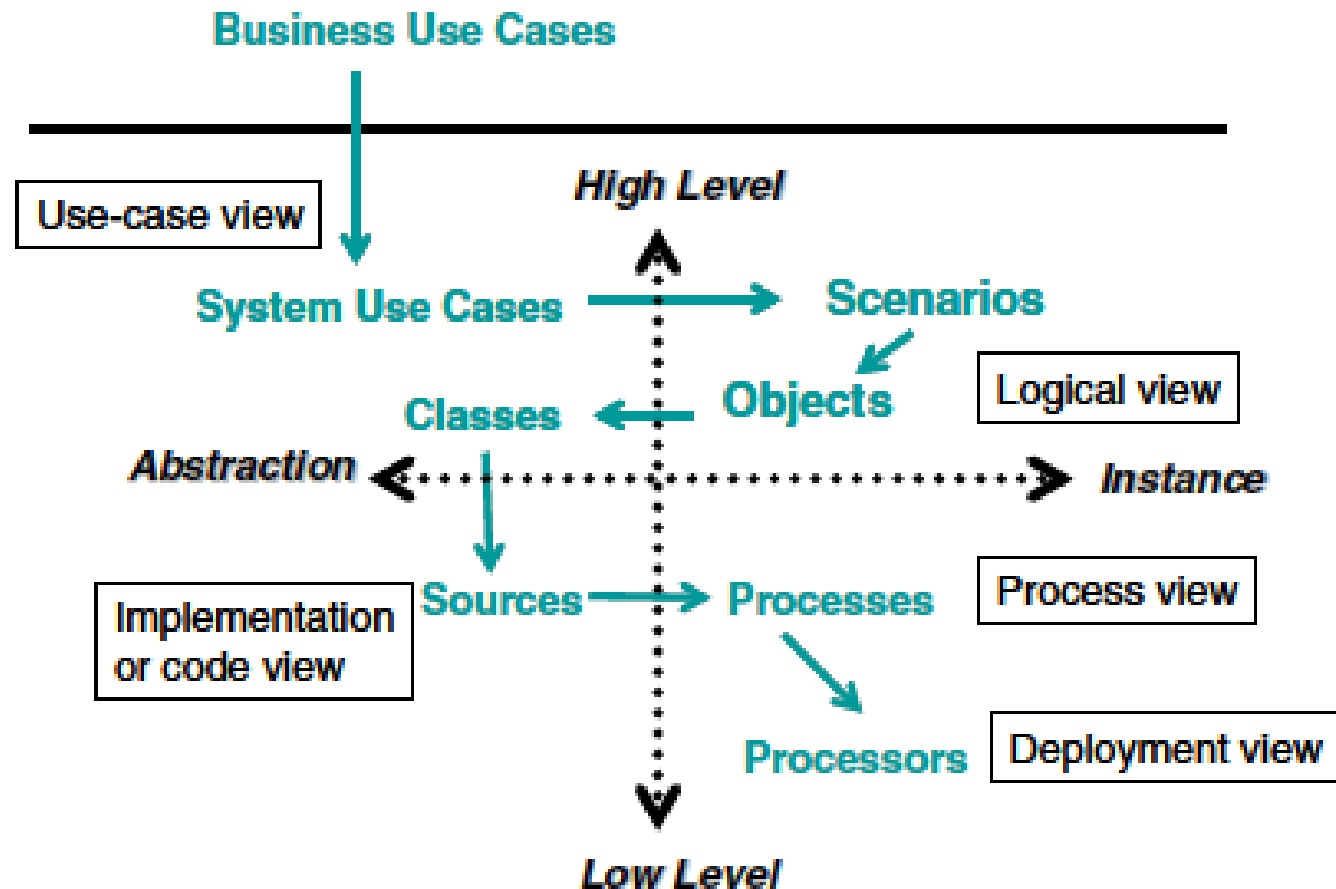
M2 RUP 4+1 Views

- Autor: Philippe Kruchten (Rational - IBM)
- Opiera się na wykorzystaniu „4+1” Views
- Wspiera iteracyjny model wytwarzania oprogramowania

M2 RUP 4+1 Views



M2 RUP 4+1 Views



M2 RUP 4+1 Views

- Architektura powstaje w fazie opracowania (*elaboration phase*).
- Powstaje wykonywalny prototyp architektury (*executable architectural prototype*).

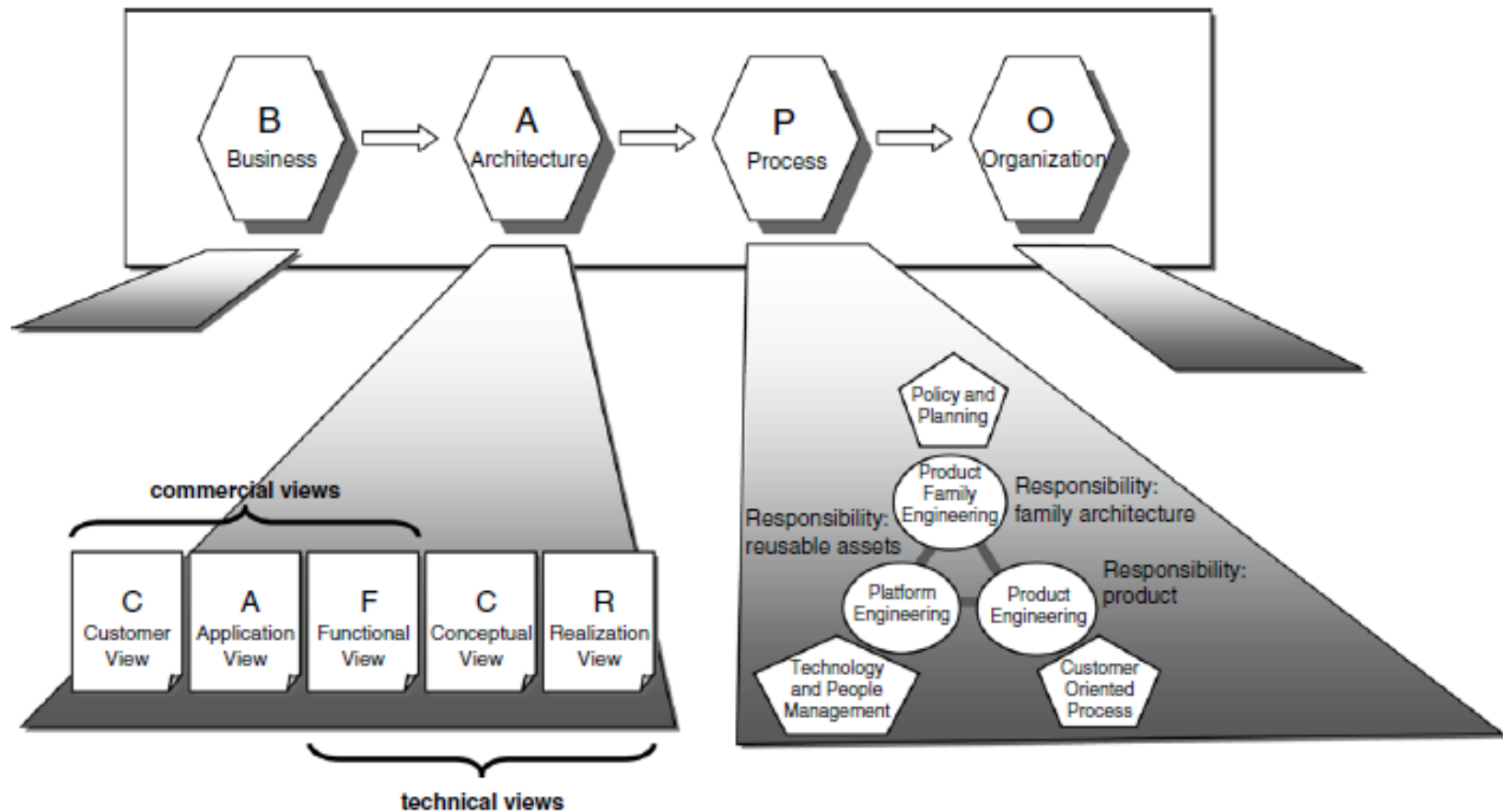
M3 Business Architecture

Processes Organization BAPO

- Kilku autorów (Phillips Research)
- Zapewnia (stara się zapewnić) tworzenie architektury, która optymalnie odpowiada potrzebom biznesowym, procesom wytwarzania oprogramowania i organizacji.
- Powstał w kontekście linii produktowych oprogramowania.

M3 Business Architecture

Processes Organization BAPO



Metody projektowania architektury

Przykłady

- M1 Attribute-Driven Design ADD
- M2 RUP 4+1 Views
- M3 Business Architecture Processes Organization (BAPO)

Metody rzadko używane przemysłowo w pełnej wersji. Ale można spotkać ich elementy w praktyce.

Wzorce architektoniczne

- **Wzorzec architektoniczny** – konfigurowalny wzorzec struktur systemów IT na różnych poziomach abstrakcji
- W literaturze rozpatruje się je w kontekście tzw. widoków
 - Widok reprezentuje architekturę z punktu widzenia pewnego wybranego aspektu strukturalnego
 - Komunikacja/interakcja między komponentami/systemami
 - Poziomy abstrakcji
 - Przepływ i przetwarzanie danych
 - Itd... itp.
 - Jest tego przynajmniej kilkanaście (nie licząc szczegółowych wzorców projektowych w programowaniu).

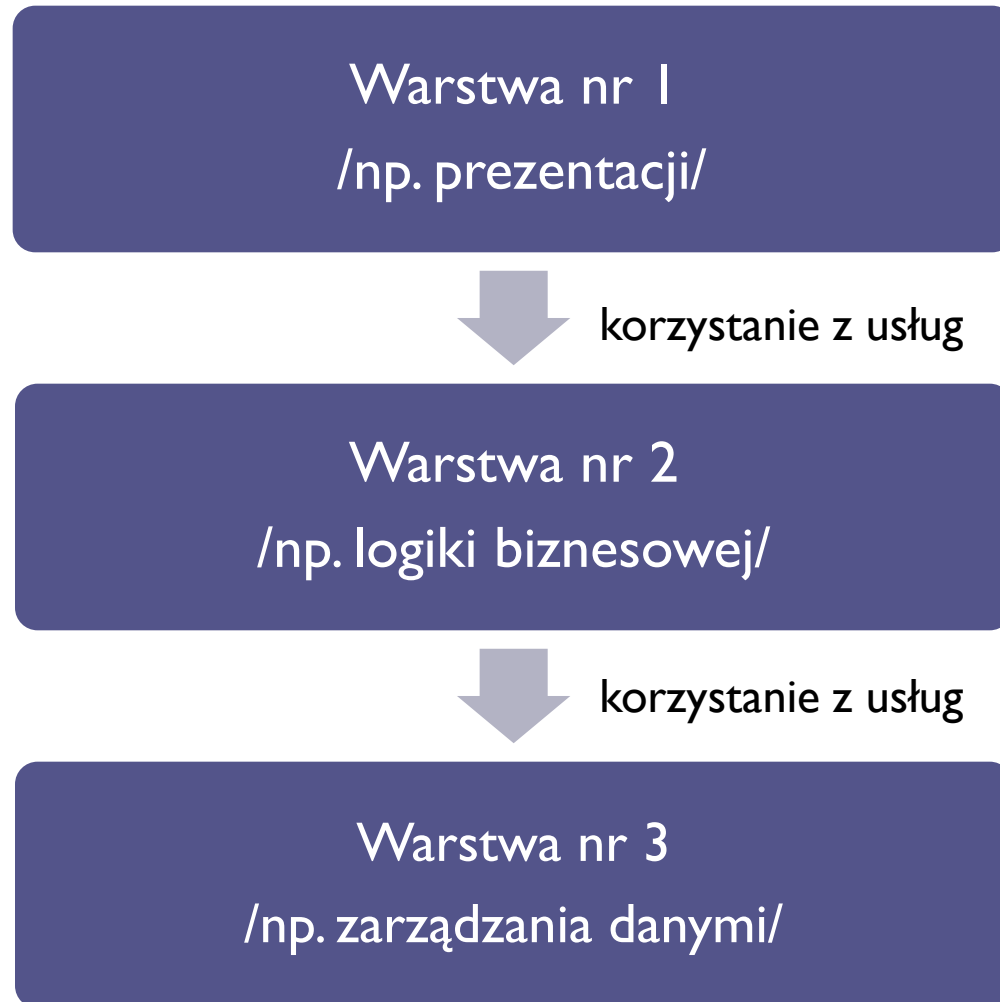
Opis wzorca

- Opis wzorca
 - Kontekst – typowa sytuacja, w której pojawia się problem
 - Problem
 - Rozwiązanie
- Ale historycznie rzecz biorąc...
 - Najpierw powstały rozwiązania, a potem były wzorce

Wzorce architektoniczne – przegląd (I)

- Widok poziomów abstrakcji
 - Warstwy
 - Warstwa pośrednicząca (np. maszyna wirtualna)
 - Np. middleware

Warstwy

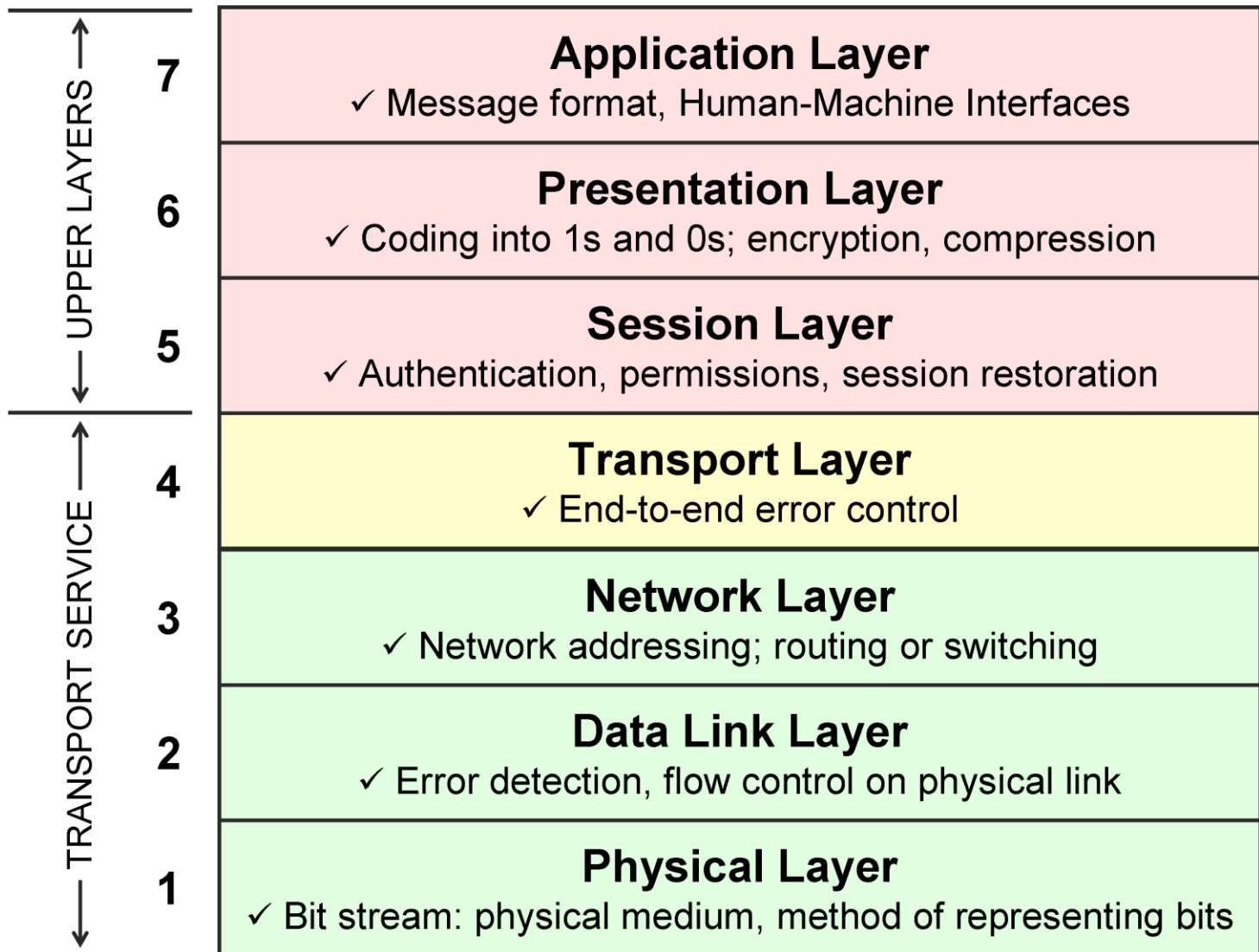


Architektura 3-warstwowa
(ang. three tier)

Warstwy

- **Problem:** podział systemu na niezależne fragmenty rozwijane niezależnie (lub potencjalnie dostarczane) przez różne podmioty. Zwiększenie przenośności, modyfikowalności i ponownego użycia
- **Wady:** jak każdego pośrednika – wnosi swój narzut, budowa warstwowa „od zera” jest bardziej kosztowna

Warstwy - przykład

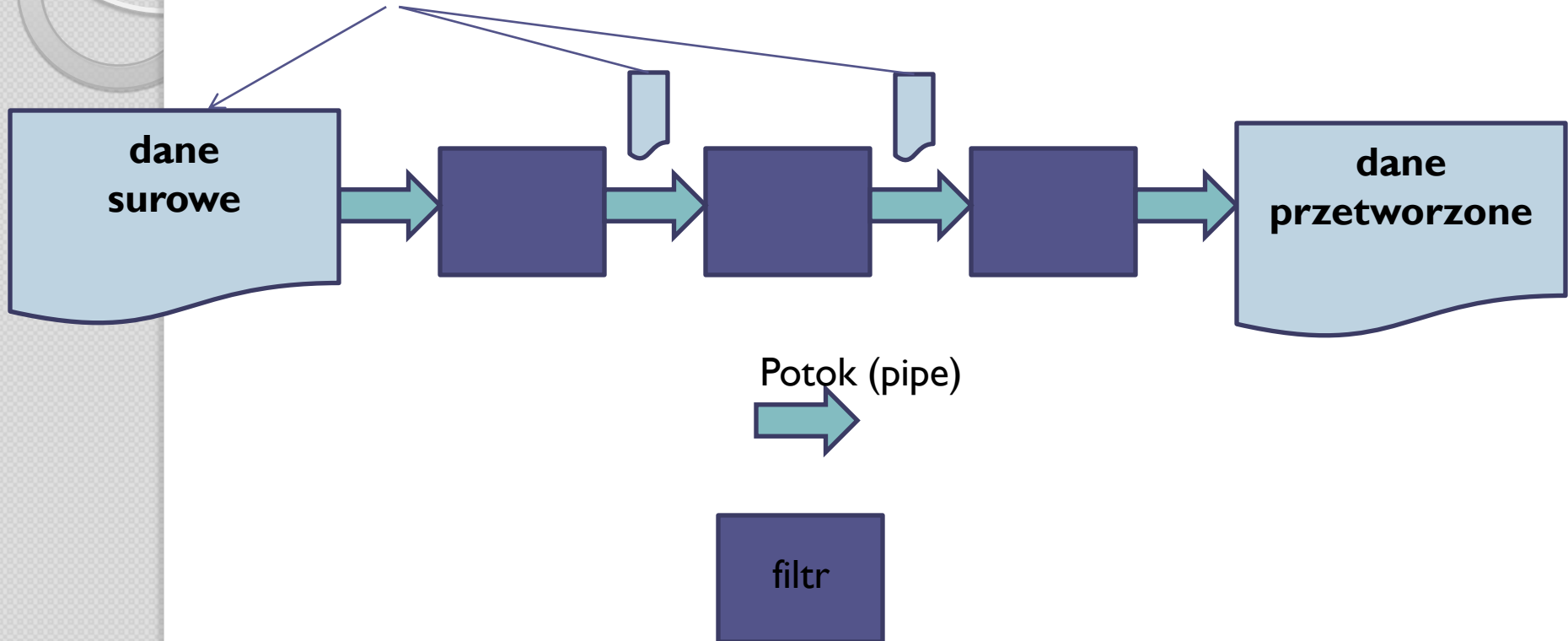


Wzorce architektoniczne – przegląd (2)

- Widok przepływu danych
 - Potoki (pipes) i filtry
 - Sekwencyjne przetwarzanie wsadowe
 - Procesy biznesowe (vide zajęcia z SOA i procesów)
 - Przepływ danych bez narzuconej struktury

Potoki i filtry

Całość lub fragment danych



Problem: realizacja sekwencyjnego przetwarzania przez niezależne, luźno-powiązane, uniwersalne narzędzia

Sekwencyjne przetwarzanie wsadowe



- Analogia do taśmy produkcyjnej
- Można skomponować w styl potoki i filtry

Wzorce architektoniczne – przegląd (3)

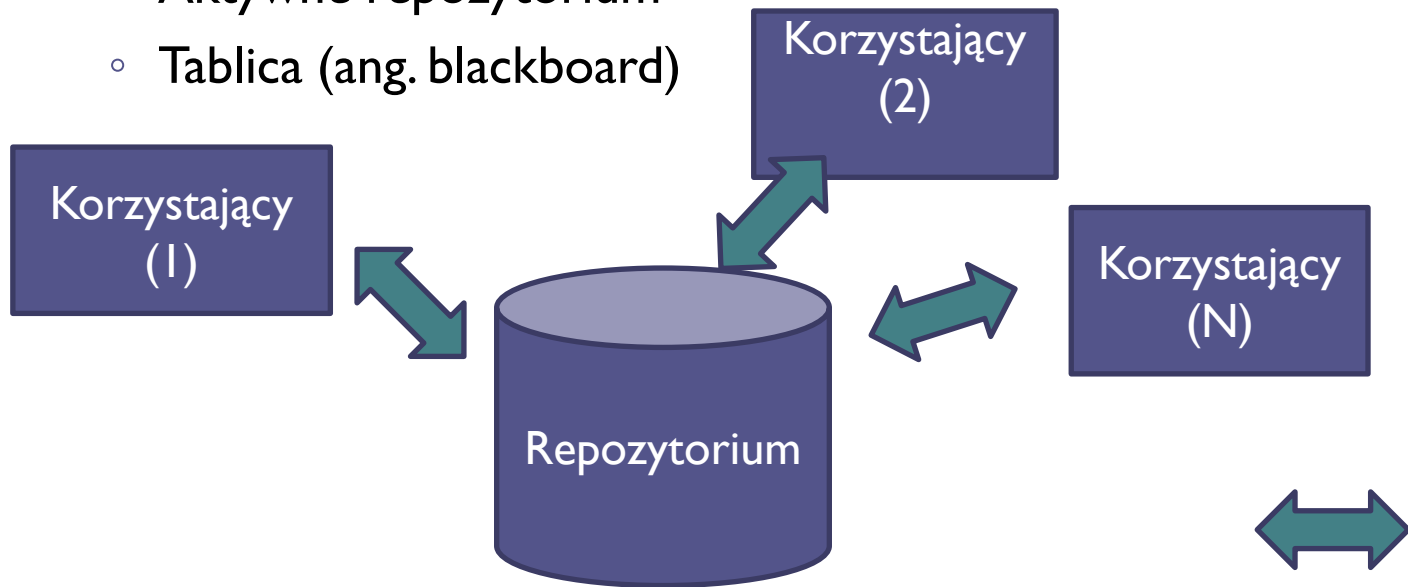
- Widok przechowywania danych

- Współdzielone repozytorium

- Typowy przykład: baza danych, zbiór plików z kodem źródłowym w systemie zarządzania wersjami
 - Często realizowany jako klient-serwer

- Aktywne repozytorium

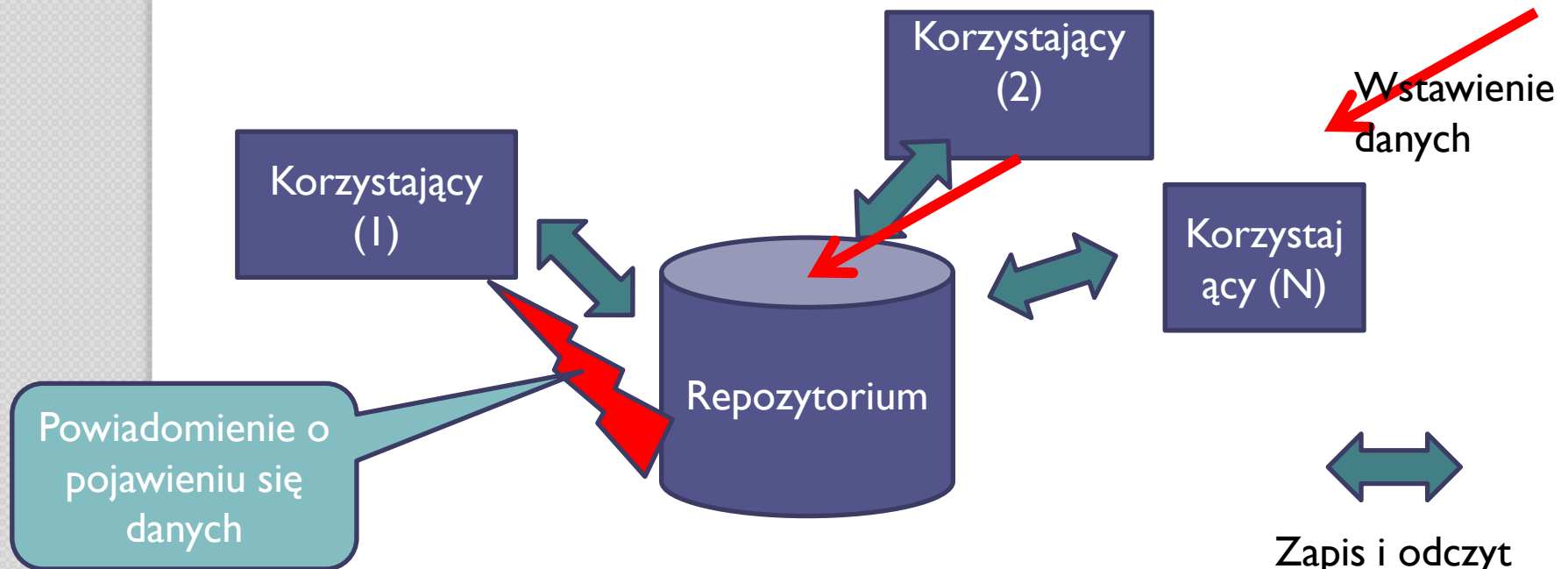
- Tablica (ang. blackboard)



Zapis i odczyt

Aktywne repozytorium

- Repozytorium informuje o pojawieniu się danych lub ich zmianie
- Blackboard – odmiana a.r. – korzystający zapisują się na dane i są o nich informowani



Wzorce architektoniczne – przegląd (4)

- Widok abstrakcji od sprzętu (widok środowiska wykonawczego)
 - Interpreter
 - Maszyna wirtualna
 - Systemy regułowe

Wzorce architektoniczne – przegląd (5)

- Widok interakcji z użytkownikiem
 - Model-view-controler
 - Prezentacja-abstrakcja-sterownik

Model-widok-kontroler

- **Model** – dane i logika biznesowa
- **View** – prezentuje dane użytkownikowi
- **Kontroler** – dostaje dane o interakcji z widoku



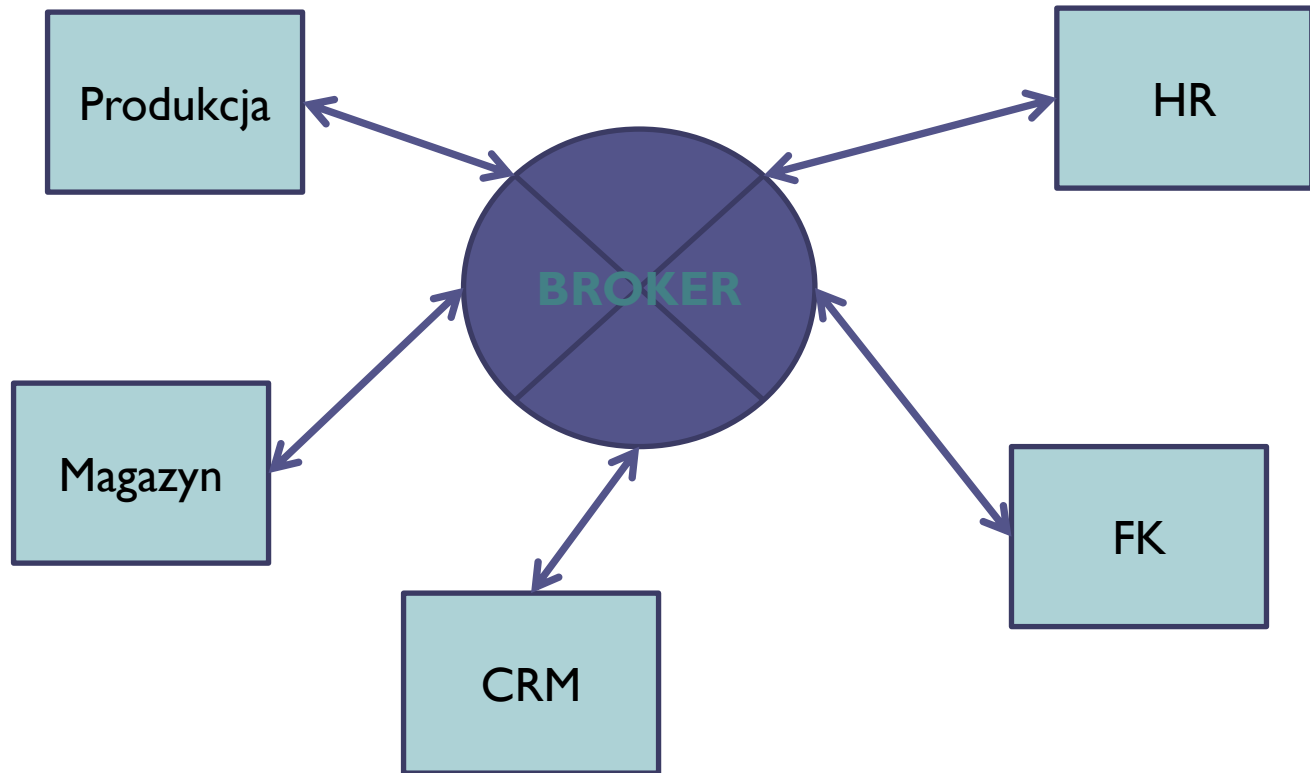
Model-widok-kontroler - problem

- **Problem:** jak odseparować interfejs użytkownika od logiki biznesowej zachowując możliwość reagowania na działania użytkownika i/lub zmiany danych, na których oprogramowanie działa

Wzorce architektoniczne – przegląd (6)

- Widok interakcji / komunikacji między komponentami
 - Wywołanie bezpośrednie (w programie)
 - Wywołanie zdalnej procedury (umiejscowionej gdzieś w sieci)
 - Wywołanie pośrednie (przez zdarzenie)
 - Broker
 - Publikuj – zarejestruj się (zgłoś się)
 - Kolejowanie wiadomości
 - Magistrala
 - Klient-serwer
 - Tu: chodzi o podział funkcji – na zdefiniowane usługi serwera
 - I korzystanie z nich przez klienta
 - Peer-to-peer – brak jednoznacznie wskazanych usługodawców i usługobiorców

Broker



Broker

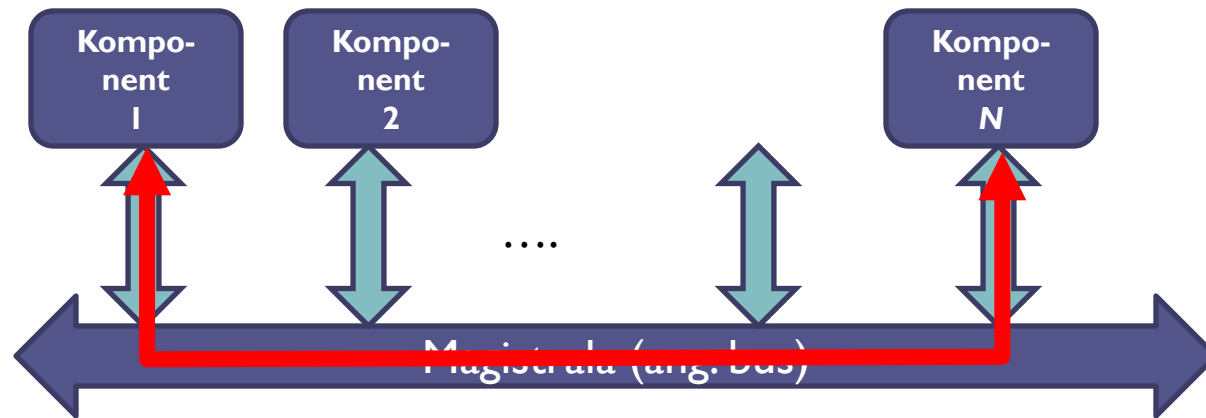
- **Broker** – jest *de facto* odmianą warstwy pośredniczącej
 - Nie komunikujemy się bezpośrednio z adresatem komunikatu
 - Posługujemy się komponentem pośredniczącym
 - Ujednolica sposób wymiany informacji narzucając własny standard interfejsu i formaty przekazywanych danych – wystarczy znać format brokera, nie trzeba znać każdego systemu z osobna
 - Informatyczny odpowiednik poczty, czy firmy kurierskiej
- Odmianą brokera jest **magistrala oraz publikuj-i-zarejestruj (się)**
- **Publikuj i zarejestruj (się)** – przypomina prenumeratę czasopisma (wymaga brokera)

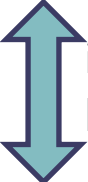
Broker

- **Problemy:**

- Jak ułatwić wymianę informacji między rozproszonymi systemami?
 - Jak wyeliminować konieczność ich lokalizacji, szczegółów umiejscowienia i dostępu do systemów usługodawców?
 - Jak zminimalizować liczbę protokołów komunikacyjnych, które musi znać usługobiorca korzystający z usług zewnętrznych?

Magistrala



 interfejs
komunikacyjny

Wzorce architektoniczne – przegląd (7)

- Widok rozmieszczenia komponentów
 - Architektura monolityczna
 - Architektura warstwowa
 - Architektura klient-serwer

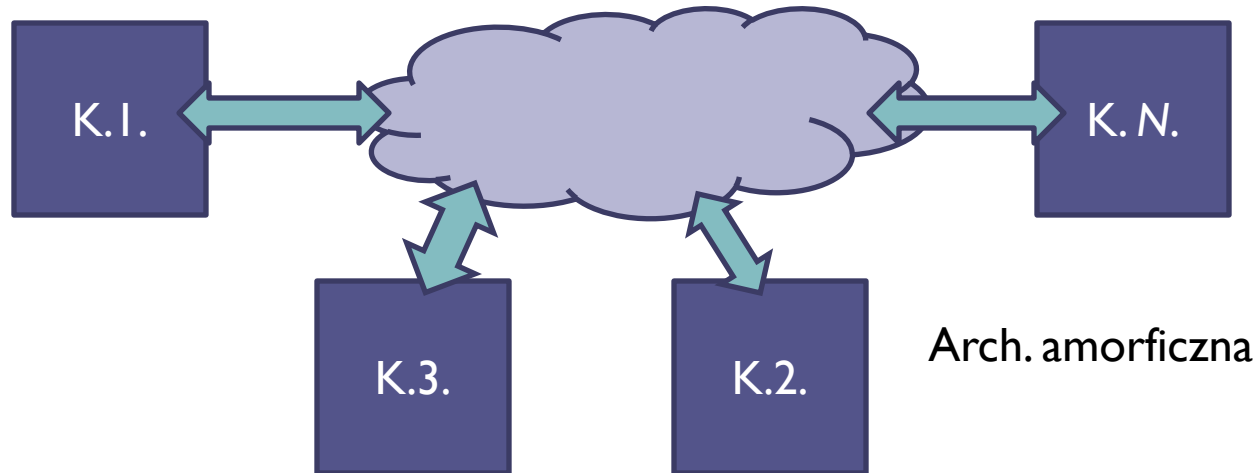
Klient-serwer



Problem:

- uwspólnienie zasobów (najczęściej danych)
- zapewnienia dostępności i wydajności kluczowych operacji
- ograniczenie obciążenia operacjami prezentacji wyników

Architektura amorficzna (peer-to-peer)



- **Problem:**

- Jak nie narzucając struktury powiązań między bliżej nie określonymi jednostkami systemu, zapewnić wymianę informacji między nimi, umożliwiając im współpracę i współkorzystanie z zasobów.

Literatura

Attribute Driven Design

<http://www.sei.cmu.edu/reports/06tr023.pdf>

Wzorce architektoniczne

<http://vyuka.pecinovsky.cz/vse/422/ArchitecturalPatternsRevisited.pdf>