

Paweł Rajba

pawel@ii.uni.wroc.pl

<http://www.itcourses.eu/>

Integracja aplikacji

Agenda

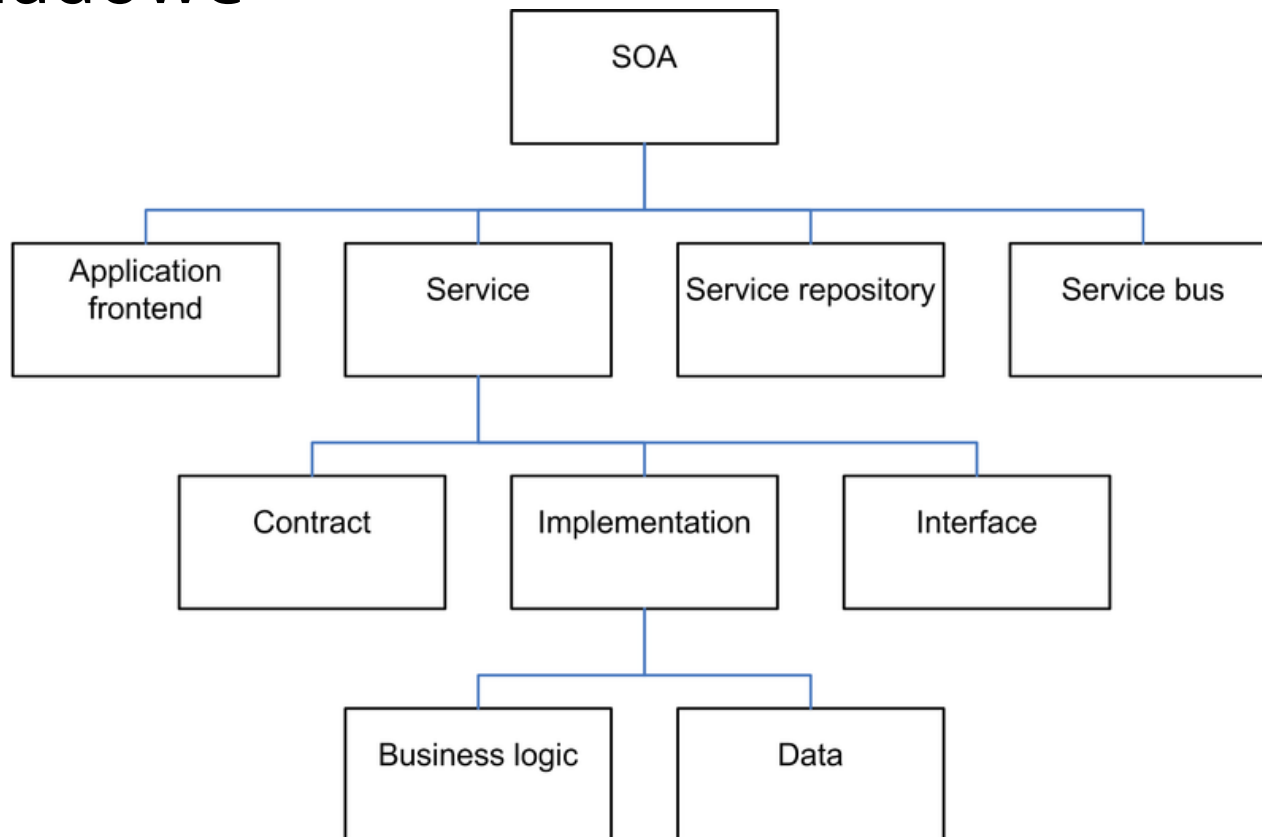
- SOA
- DDD & Context Maps
- Messaging Patterns
- Enterprise Integration Patterns

SOA

- SOA czyli Service Oriented Architecture
 - Podział systemu na komponenty, które udostępniają swoją funkcjonalności poprzez określony interfejs
 - oraz z wykorzystaniem standardów dot. komunikacji
 - Kilka własności
 - Stanowi fasadę dla funkcjonalności, które udostępnia
 - Jest taką „czarną skrzynką” o określonym API
 - Po wykonaniu każdej operacji powinien być spójny (transakcje)
 - Powinien być zabezpieczony i odporny na błędne użycie
- Uwaga: WebServices i SOA to nie to samo
 - Chociaż można zaimplementować SOA za pomocą WS

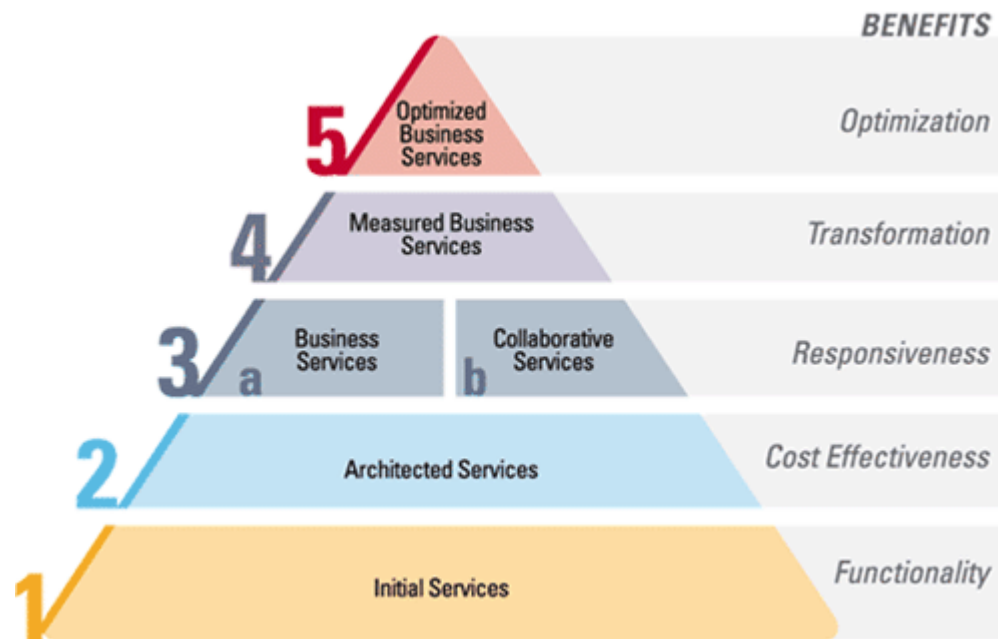
SOA

■ Składowe



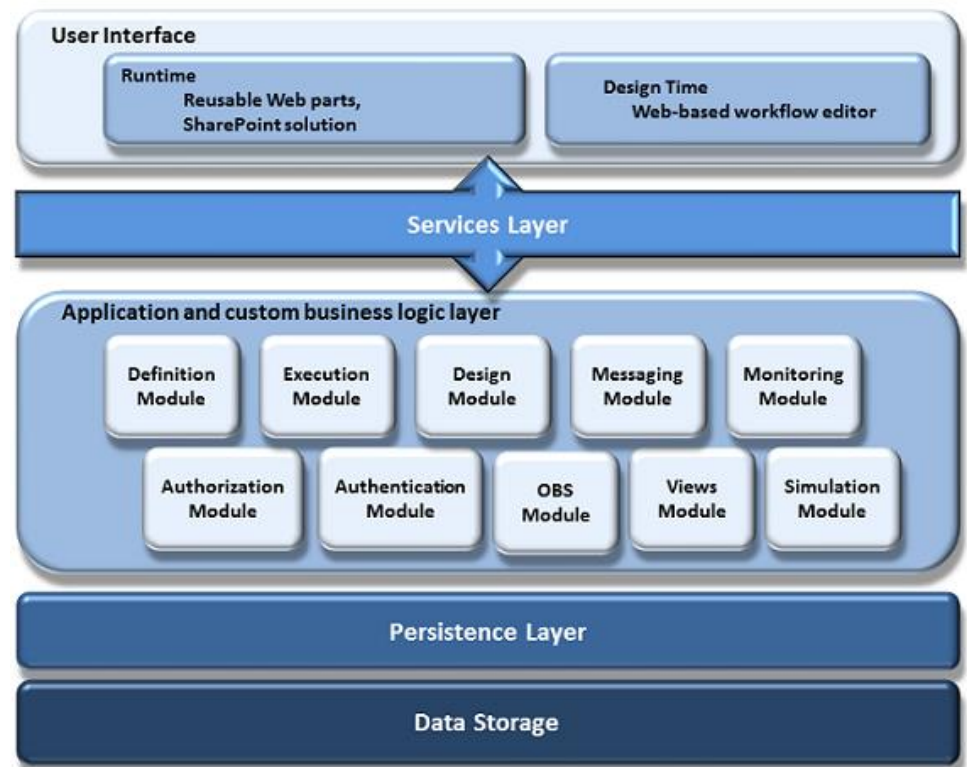
SOA

- Dojrzałość zastosowania usług



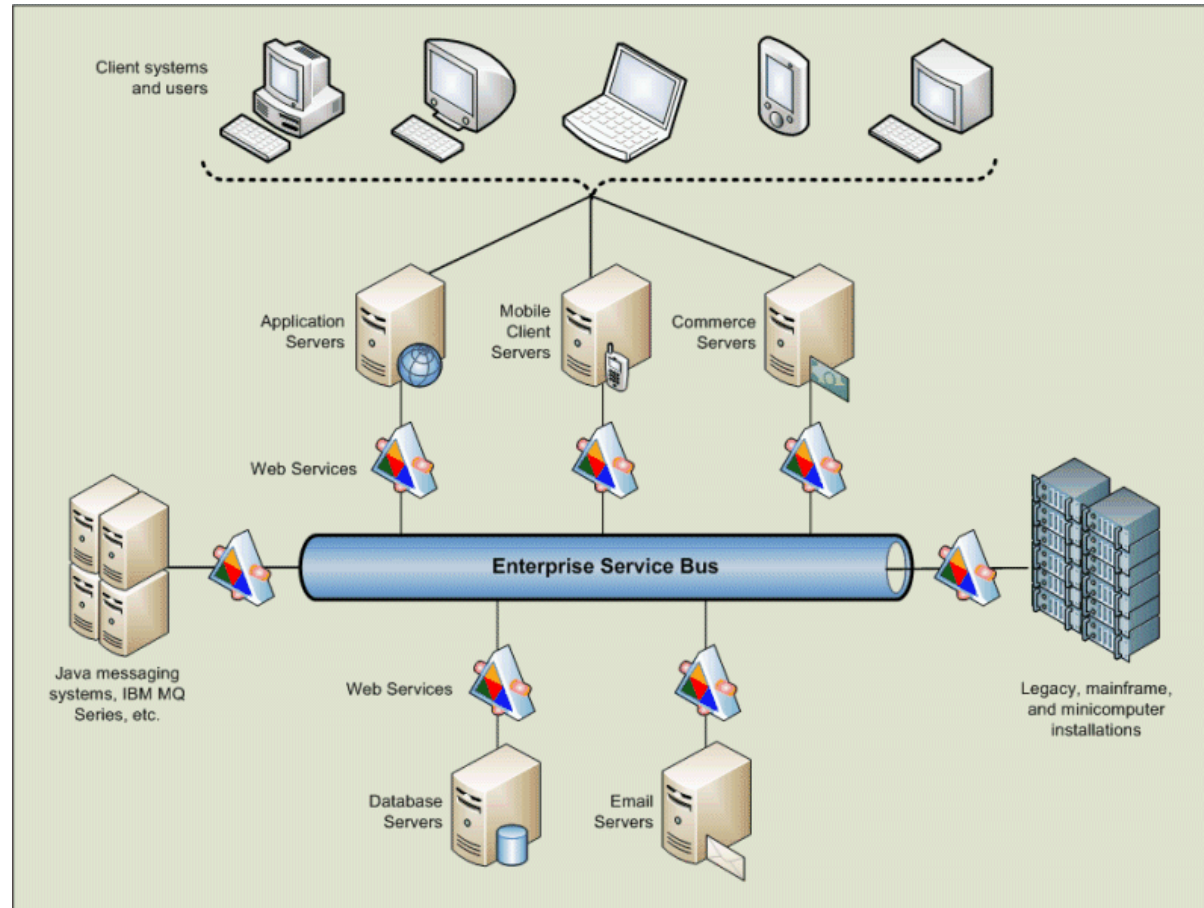
SOA

- Główne zyski z zastosowania SOA
 - Abstraction
 - Autonomy
 - Formal contract
 - Loose coupling
 - Reusability
 - Statelessness



Implementacja SOA

- WebServices
- ESB



DDD & Context Maps

- W integracji aplikacji istotna jest relacja pomiędzy aplikacjami i tej relacji konsekwencje
- Problem modeli i ich translacji
- Przyjrzymy się, jak wygląda to z perspektywy DDD
 - I zamiast aplikacji będziemy rozpatrywać Bounded Context

Wzorce relacji

- Partnership
 - Zespoły obu systemów razem pracują nad integracją i wypracowują kompromisy
 - Jest wspólny cel i albo oba zespoły będą miały sukces, albo porażkę
- Shared kernel
 - Wyróżniona wspólna część modelu
 - Powinna być jak najmniejsza i zgodna z Ubiquitous Language obu systemów
 - Ewentualne zmiany zawsze powinny być uzgodnione z drugą stroną

Wzorce relacji

- Customer-Supplier
 - Relacja oparta na upstream-downstream
 - Relacja partnerska: customer może określić potrzeby, które supplier stara się dostarczyć
- Conformist
 - Relacja oparta na upstream-downstream
 - Dostawca ma interesu w realizacji potrzeb odbiorcy i to odbiorca musi się dostawać

Wzorce relacji

- Separate Ways
 - Bounded contexty, które nie mają z sobą relacji
- Big Ball of Mud
 - Jeden wielki monolit

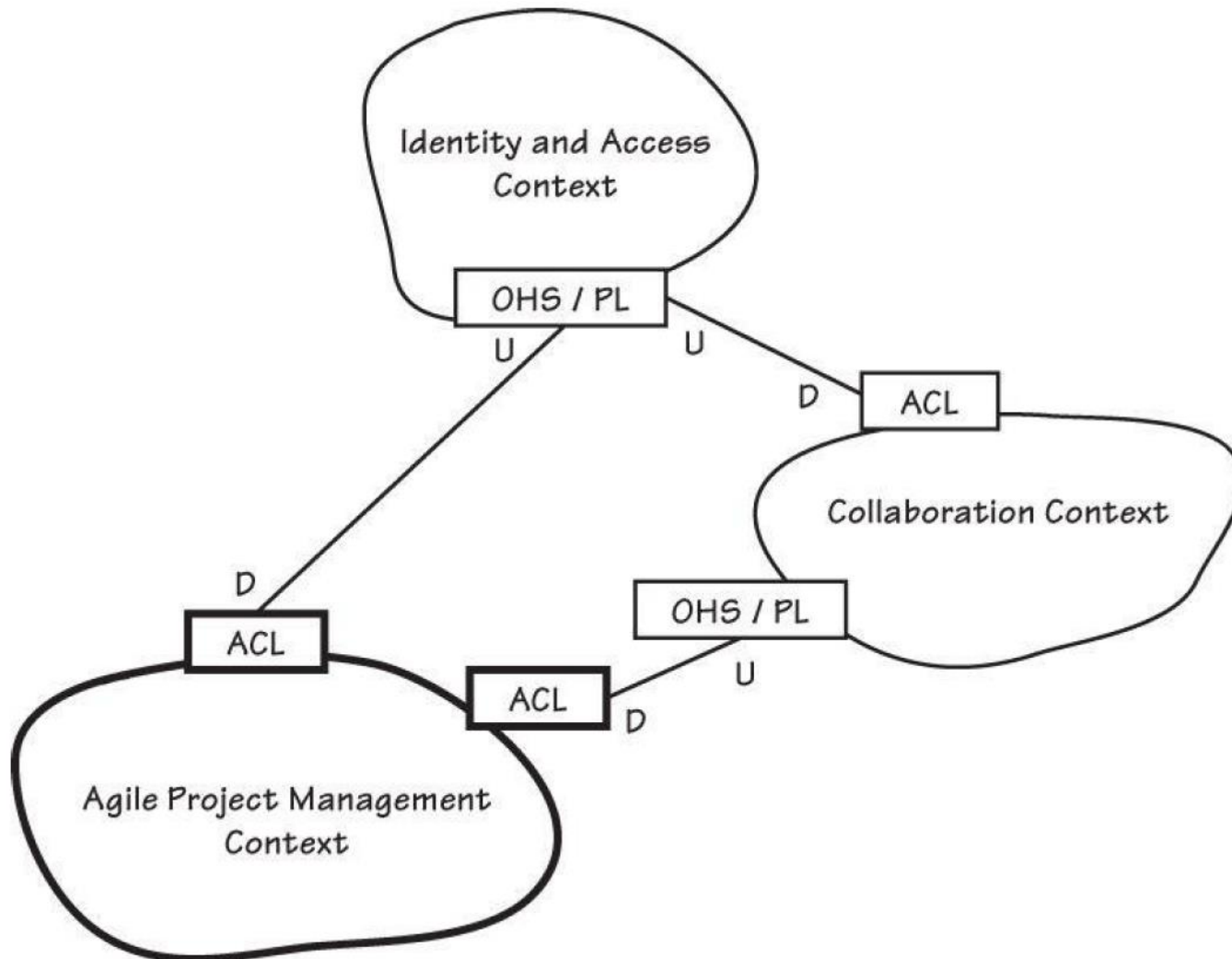
Dodatkowe elementy

- Anticorruption Layer (ACL)
 - Warstwa zapewniająca translacje zewnętrznego modelu do wewnętrznego
 - Zwykle po stronie downstream
- Open Host Service (OHS)
 - Definicja protokołu, dzięki któremu można zbudować integrację
- Published Language (PL)
 - Definicja języka (formatu) udostępnionych danych

Context Map

- Diagram przedstawiający
 - Bounded contexty
 - Translacje
 - Relacje między nimi
 - Elementy dodatkowe jak OHS czy PL

Context Map



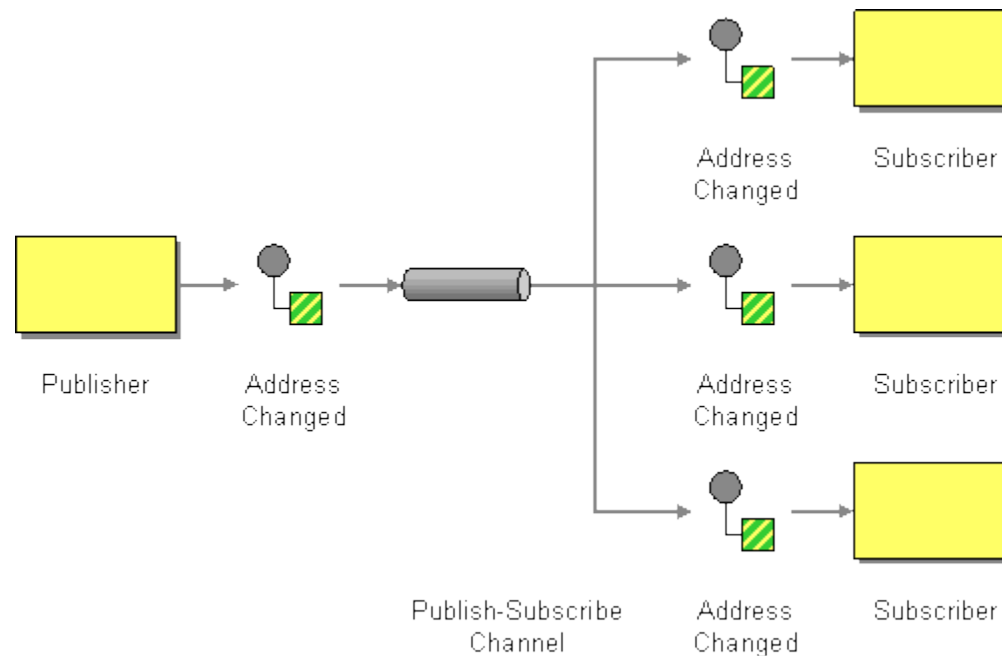
Messaging Patterns

- Publish/Subscribe
- Request/Reply
- Fire/Forget

- Realizacja
 - Synchroniczna
 - Asynchroniczna

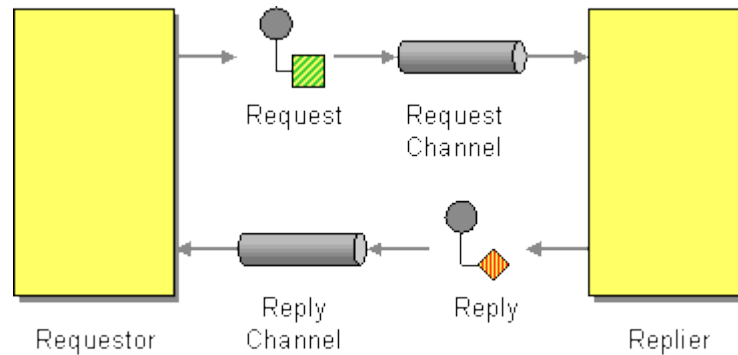
Messaging Patterns

- Publish/Subscribe



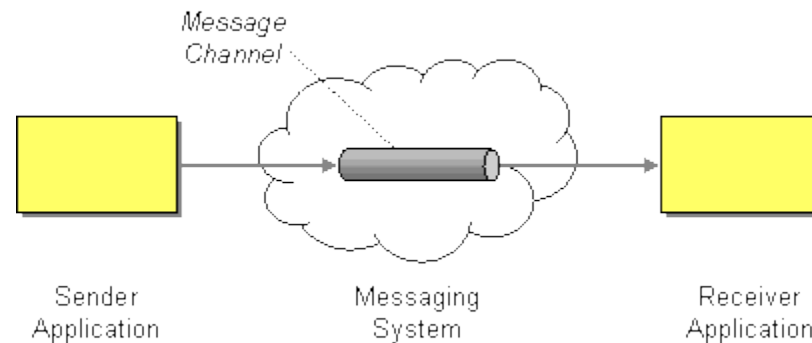
Messaging Patterns

- Request/Reply



Messaging Patterns

- Fire/forget (request only)



Enterprise Integration Patterns

- Oglądamy katalog
 - <http://www.eaipatterns.com/toc.html>

Literatura

■ SOA

- http://en.wikipedia.org/wiki/Service-oriented_architecture
- <http://www.pnmsoft.com/resources/bpm-tutorial/soa-tutorial/>
- http://www.enterprise-architecture.info/EA_Services-Oriented-Enterprise.htm
- <http://www.soapatterns.org/>
- <http://www.slideshare.net/SubhaPrasad/soa-14718016>

■ ESB

- <http://msdn.microsoft.com/en-us/library/ff648282.aspx>
- http://en.wikipedia.org/wiki/Enterprise_service_bus

■ EIP

- <http://www.eaipatterns.com/>
- https://access.redhat.com/site/documentation/en-US/Fuse_ESB/4.3.1/html-single/Implementing_Enterprise_Integration_Patterns/index.html

■ Messaging Patterns

- <http://msdn.microsoft.com/en-us/library/ff649664.aspx>
- <http://msdn.microsoft.com/en-us/library/aa480027.aspx>
- <http://camel.apache.org/async.html>