

Modele [Software | Platform] as a Service

... czyli od SaaS'a do $\frac{P}{I}$ -aaS'a

Radosław Adamus
IIS, PŁ



Politechnika
Łódzka



- Software as a Service
- Modele chmury obliczeniowej a architektura warstwowa
- Platform as a Service
 - Przykładowe platformy
- Usługi aplikacyjne
 - Kolejki komunikatów
- Inne rodzaje platform
 - DBaaS, iPaaS, mBaaS

- Aplikacja nie jest własnością klienta, udostępniana jest poprzez Internet.
- Obowiązki zarządzania, aktualizacji, pomocy technicznej po stronie dostawcy.
- Użytkownik kupuje działające rozwiązanie o określonej funkcjonalności bez radzenia sobie z infrastrukturą informatyczną oraz zapleczem technicznym.
 - dostęp do najnowszych technologii informatycznych bez długotrwałych wdrożeń i dużych inwestycji.
- Opłaty
 - najczęściej cykliczne (abonament zamiast jednorazowego wydatku w momencie zakupu).

- Webowy dostęp do komercyjnego oprogramowania
- Scentralizowane zarządzanie oprogramowaniem
- Model dostarczania – jeden – do –wielu
- Użytkownik nie przyjmuje się aktualizacjami ani poprawkami
- Możliwa integracja pomiędzy elementami oprogramowania (API)

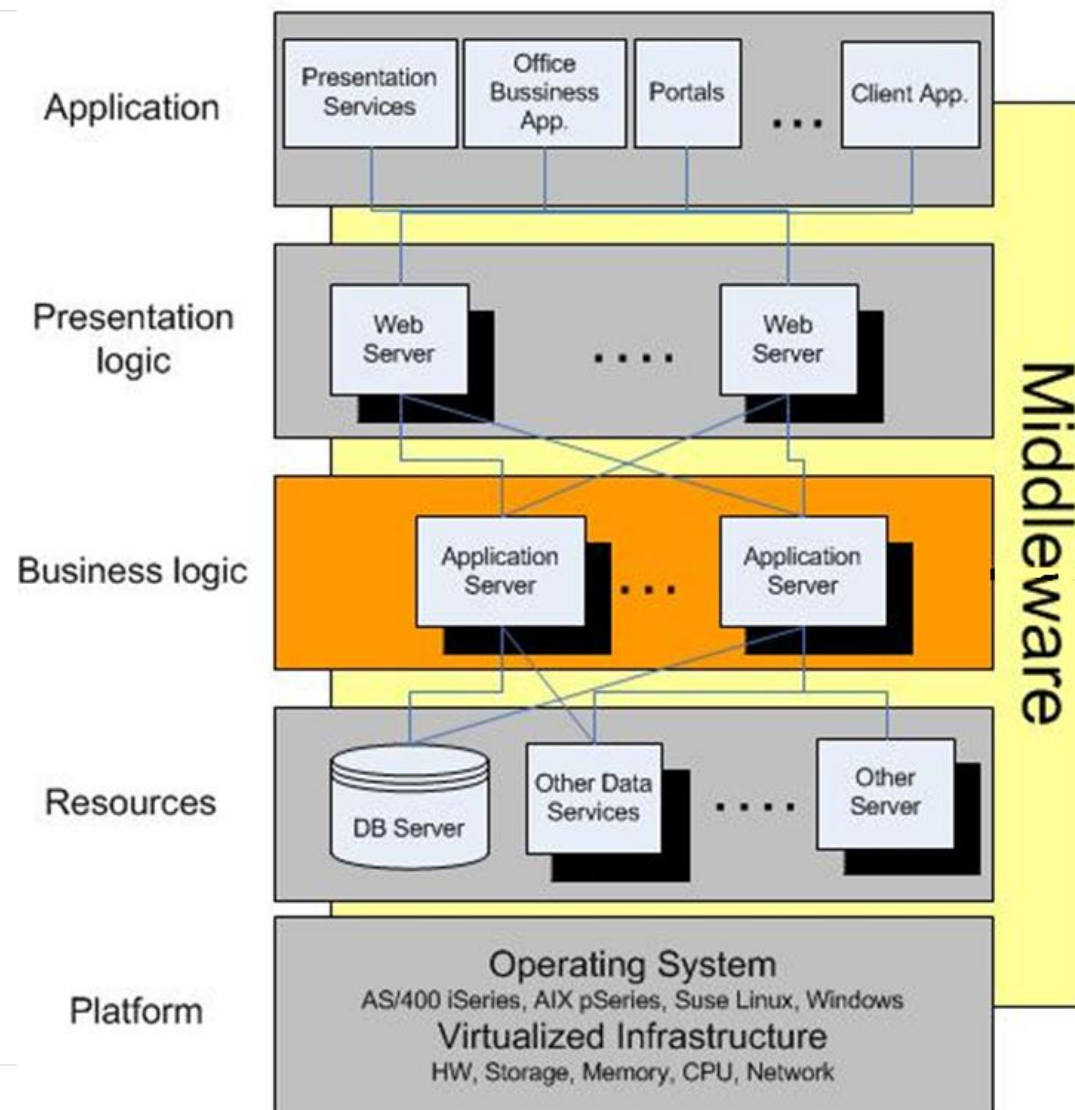
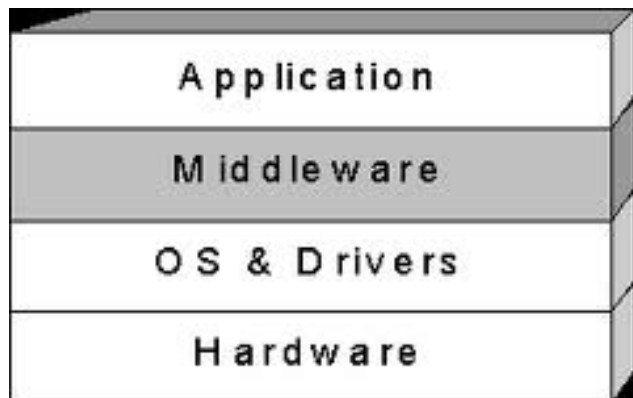


IaaS to infrastruktura

- Jednostki obliczeniowe
- Load-balancers, auto-scaling groups, VPC,
- Ok., ale aplikacje potrzebują:
 - Środowiska uruchomieniowego, kontenerów aplikacji, usług aplikacyjnych, systemów baz danych...
- Jednym słowem potrzebują platform
 - Produkcyjnej, testowej, developerskiej ...
- W IaaS wszystko trzeba „postawić” samodzielnie na dostępnej infrastrukturze
- Niech ktoś spróbuje postawić np. system bazy danych,
 - żeby się skalował (klaster?), jak zapewnić wymaganą przepustowość operacji zapisu, odczytu?
 - a co z aktualizacjami?



Architektura warstwowa

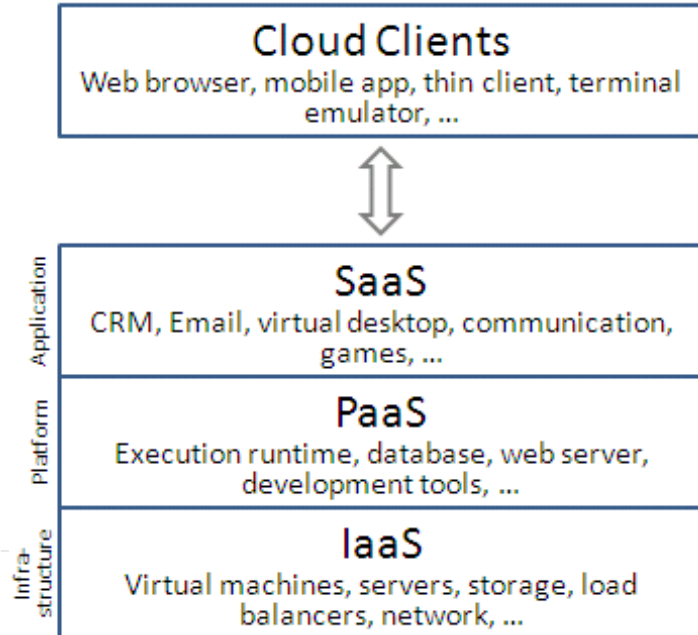


Everything-as-a-Service (XaaS)

- IaaS – administrator, devops
 - Host provisioning, load balancing, public and private network connectivity, firewalls, storage
- PaaS – developer
 - Solution stack as a service
 - Application services
 - Care free environment for developers
 - SaaS for developers
- SaaS – klient
 - Dropbox, Netflix, Spotify, Facebook,...

- A **platform as a service (PaaS)** offering, usually depicted in all-cloud diagrams between the SaaS layer above it and the IaaS layer below, is a broad collection of application infrastructure (middleware) services (including application platform, integration, business process management and database services). However, the hype surrounding the PaaS concept is focused mainly on application PaaS (aPaaS) as the representative of the whole category.

<http://www.gartner.com/it-glossary/platform-as-a-service-paas>



http://en.wikipedia.org/wiki/File:Cloud_computing_layers.png

1. Usługi dedykowane rozwojowi, testowaniu, wdrażaniu, hostowaniu i wsparciu oprogramowania.
 - Połączone w jedno zintegrowane środowisko
2. Narzędzia (webowe) pomagające w tworzeniu, modyfikacji, testowaniu i wdrażaniu różnych scenariuszy interfejsów użytkownika.
3. Dzierżawiona (multitenant) architektura, w której wielu użytkowników wykorzystuje tę samą aplikację do rozwijania oprogramowania
4. Wbudowana, skalowalność (w aplikacje)



- Application aPaaS
- Integration iPaaS
- Database as a Service - DaaS
- mBaaS
- ...

Platformy PaaS

AppScale

GridGain
CLOUD COMPUTING

flexiscale™

amazon
web services™

af appfog

Engine Yard™

Visual WebGui

Google App Engine

ca technologies



redhat
OPENSIFT

Appistry

salesforce.com

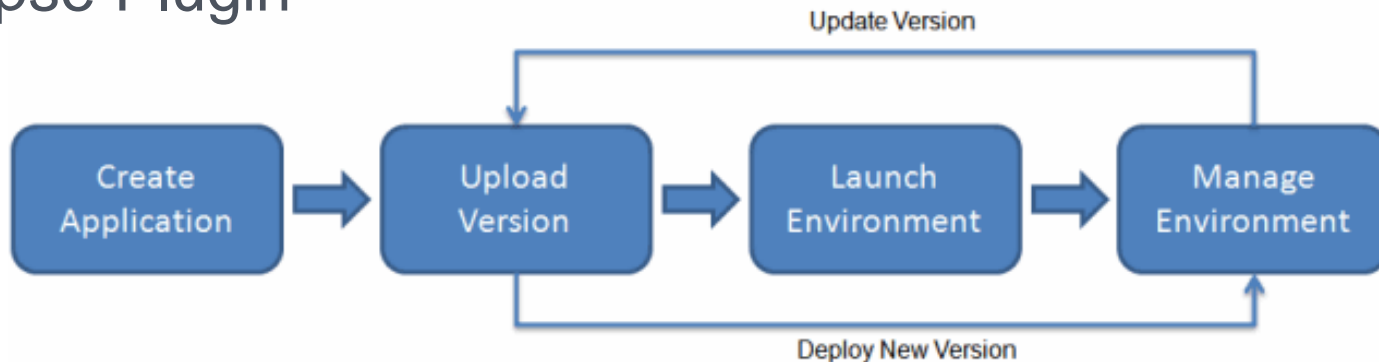
Windows Azure™

caspio™

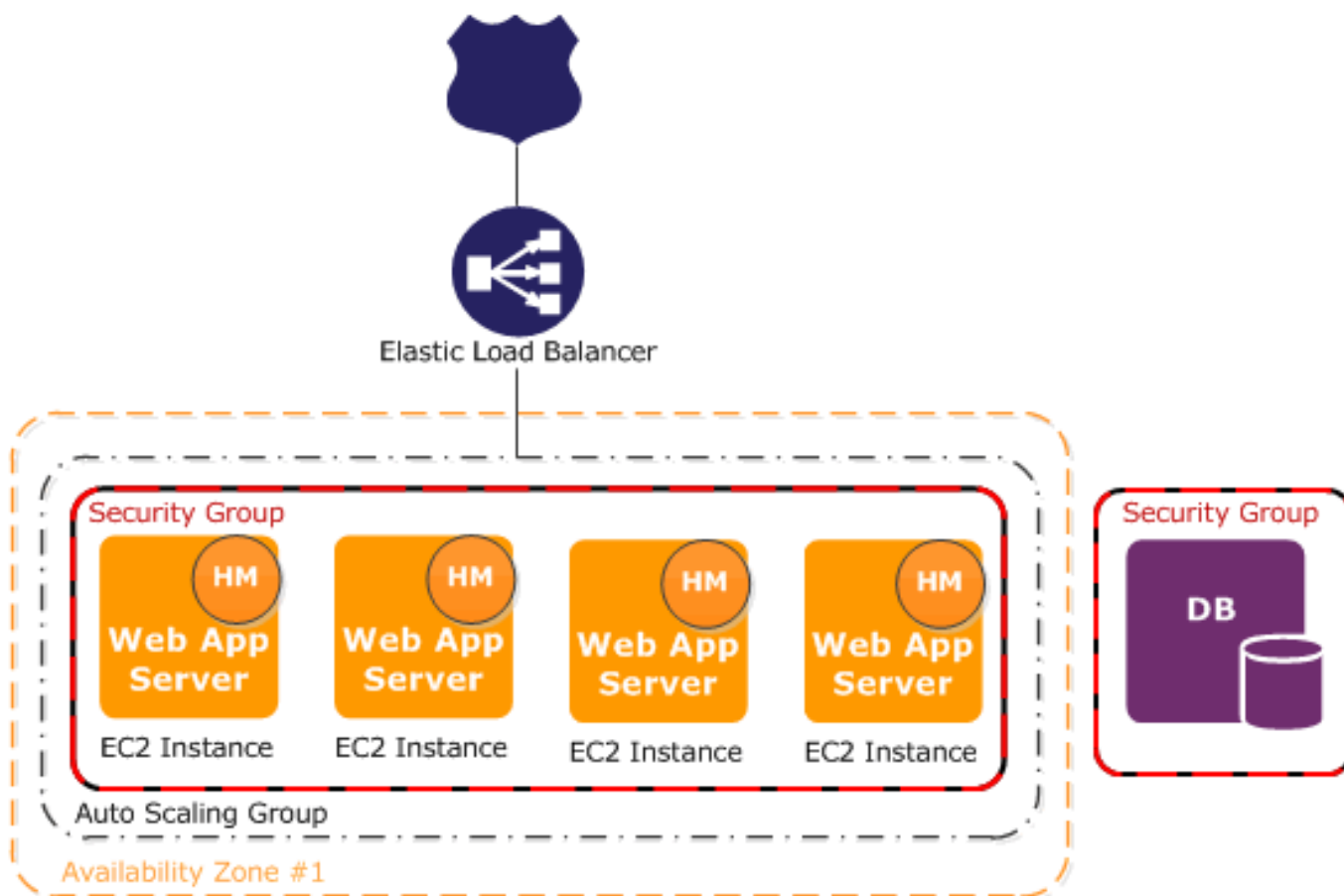
heroku



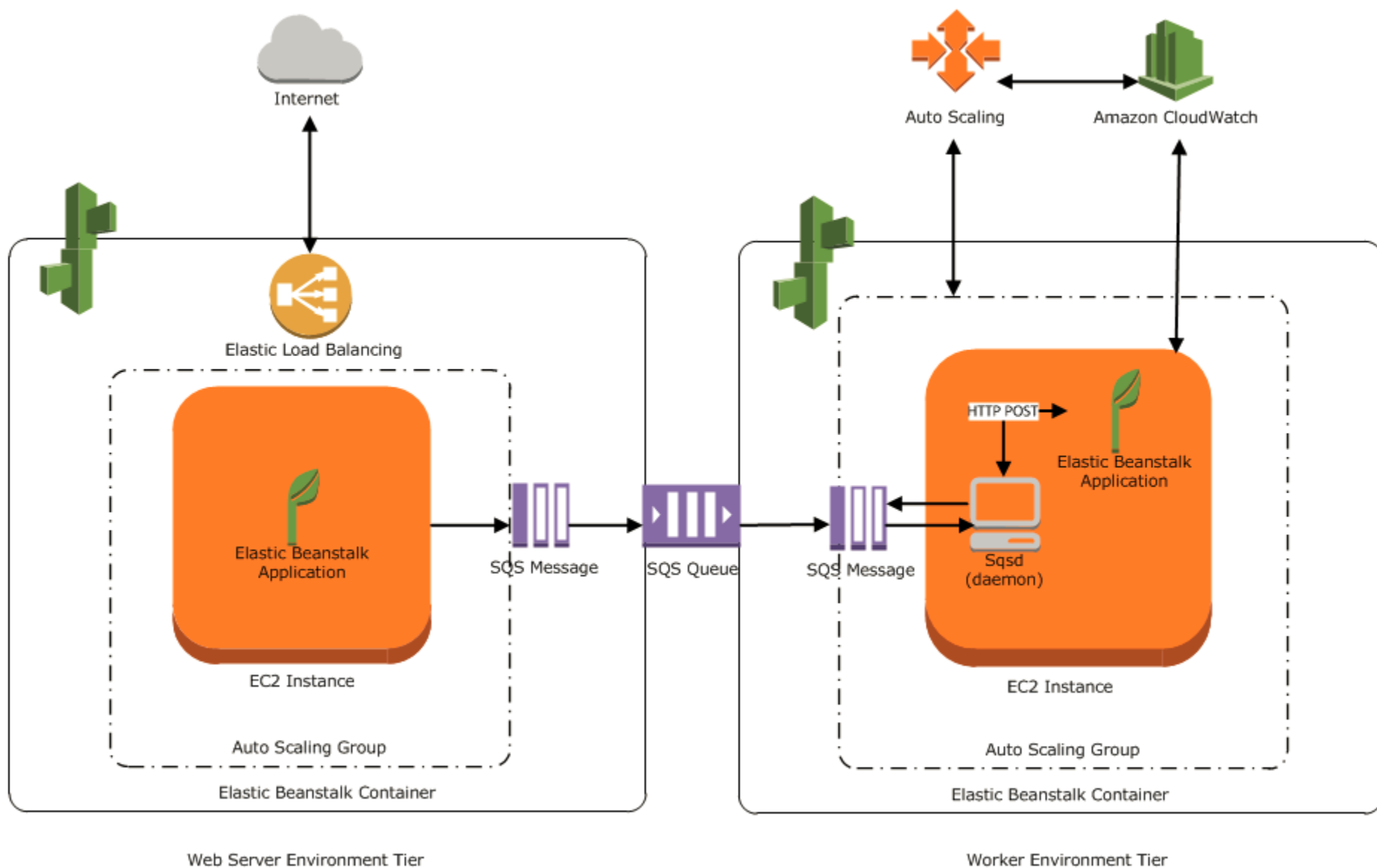
- Automatyzuje proces wdrażania i zarządzania aplikacją w chmurze AWS
 - Środowisko uruchomieniowe (web server, worker)
 - Predefiniowane konfiguracje (np. nodeJS, PHP, Tomcat, Python, Ruby)
 - Rodzaj środowiska (instancja EC2, ASG + ELB)
 - Dostępność bazy danych RDS, kolejek komunikatów SQS
- Automatyczny deploy,
- Eclipse Plugin



- Web Server



Architektura Elastic Beanstalk (2 z 2)



- Ruby, Java, Node.js, Python, Clojure, Scala
- Dyno – abstrakcyjna jednostka obliczeniowa (izolowane środowisko uruchamiające procesy)
- Dodatki (addons)
 - Data stores, mobiles, search, logging, email, queueing, analytics, caching, monitoring
- Integracja z systemem Git

- **The Twelve Factors App**

- I. Codebase
- II. Dependencies
- III. Config
- IV. Backing Services
- V. Build, release, run
- VI. Processes
- VII. Port binding
- VIII. Concurrency
- IX. Disposability
- X. Dev/prod parity
- XI. Logs
- XII. Admin processes



- Budowanie i uruchamianie aplikacji na infrastrukturze Google
- Środowiska
 - Java, Python, PHP, Go
- Cechy
 - Trwały skład – Datastore (zapytania, transakcje, sortowanie)
 - Skalowanie i load balancing
 - Izolowane procesy




- Środowiska
 - Java, Ruby, PHP, Python, Node, Scala, .NET
- Bazy danych
 - MySQL, Postgres, Redis,
- Usług aplikacyjne
 - RabbitMQ
- Zarządzanie kodem
 - Git, Svn, Mercurial
- Inne
 - DaaS – ClearDB, MongoHQ, MongoLab, Redis Cloud,

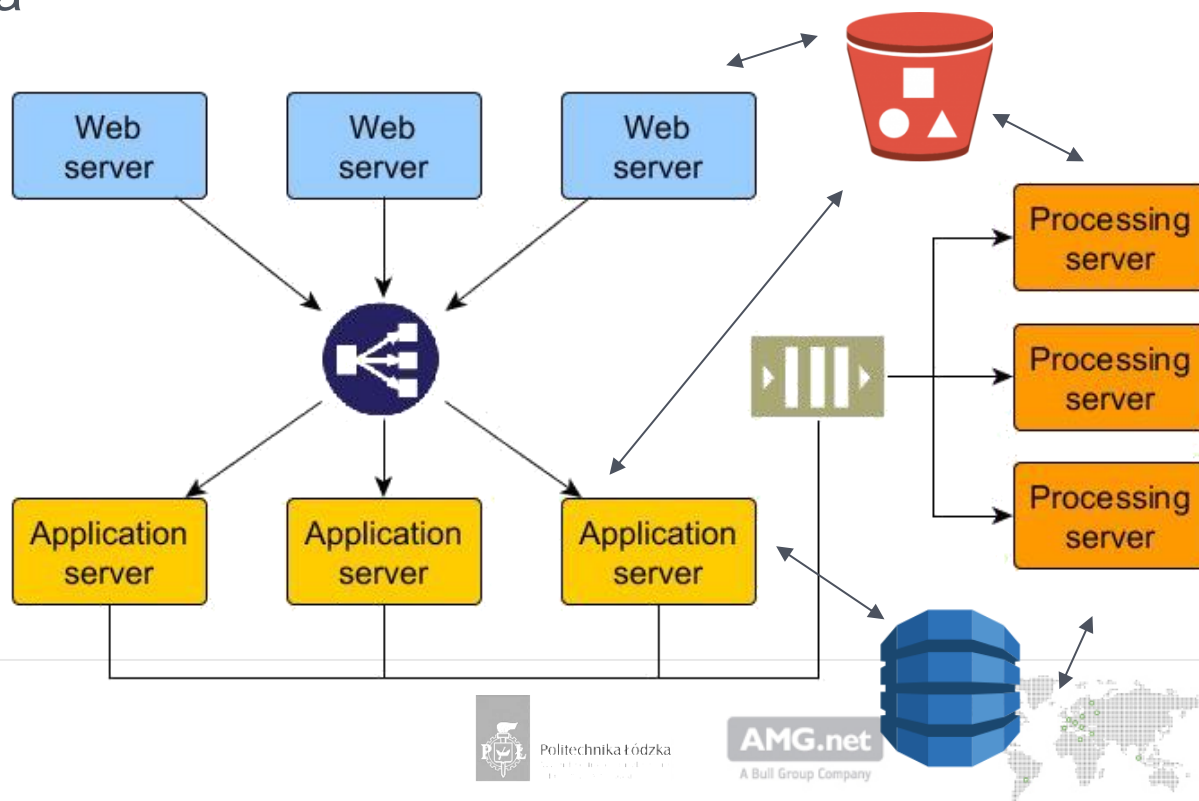
- Poziomie, np.:
 - Usługi kolejek komunikatów
 - Usługi powiadomień (notyfikacje)
 - Usługi bazodanowe
- Pionowe, np.:
 - Transkodowanie plików filmowych



- Kolejka komunikatów
 - Komponent reprezentujący asynchroniczny protokół komunikacji
 - Model publisher/subscriber (producent/consument) – nadawca i odbiorca nie muszą być dostępni w tym samym czasie
 - Cechy, parametry:
 - Trwałość, bezpieczeństwo, gwarancja dostarczenia, reguły usuwania, rozmiar wiadomości,
 - Kolejki rozproszone
- Zastosowania:
 - Kolejki w systemach operacyjnych
 - Element łączący niezależne aplikacje (komponenty) – message-oriented middleware
 - JBoss Messaging, JORAM, Apache ActiveMQ, Sun Open Message Queue, Apache Qpid, RabbitMQ
- Standard JMS API



- Komponent implementujący kolejkę komunikatów dostępny jako jedna z usług PaaS.
- Amazon SQS 
 - Luźne powiązania
 - buforowanie



- Konfiguracja

Czas, jaki wiadomość będzie niewidoczna dla odbiorców po pobraniu jej z kolejki

Default Visibility Timeout:	<input type="text" value="30"/>	seconds ▾	Value must be between 0 seconds and 12 hours.
Message Retention Period:	<input type="text" value="4"/>	days ▾	Value must be between 1 minute and 14 days.
Maximum Message Size:	<input type="text" value="256"/>	KB	Value must be between 1 and 256 KB.
Delivery Delay:	<input type="text" value="0"/>	seconds ▾	Value must be between 0 seconds and 15 minutes.
Receive Message Wait Time:	<input type="text" value="0"/>	seconds	Value must be between 0 and 20 seconds.

- Konfiguracja

Maksymalny czas, jaki wiadomość będzie czekała w kolejce

Default Visibility Timeout:	<input type="text" value="30"/>	<input type="text" value="seconds"/>	Value must be between 0 seconds and 12 hours.
Message Retention Period:	<input type="text" value="4"/>	<input type="text" value="days"/>	Value must be between 1 minute and 14 days.
Maximum Message Size:	<input type="text" value="256"/>	<input type="text" value="KB"/>	Value must be between 1 and 256 KB.
Delivery Delay:	<input type="text" value="0"/>	<input type="text" value="seconds"/>	Value must be between 0 seconds and 15 minutes.
Receive Message Wait Time:	<input type="text" value="0"/>	<input type="text" value="seconds"/>	Value must be between 0 and 20 seconds.

- Konfiguracja

Maksymalny rozmiar wiadomości

Default Visibility Timeout:	<input type="text" value="30"/>	<input type="text" value="seconds"/>	Value must be between 0 seconds and 12 hours.
Message Retention Period:	<input type="text" value="4"/>	<input type="text" value="days"/>	Value must be between 1 minute and 14 days.
Maximum Message Size:	<input type="text" value="256"/>	<input type="text" value="KB"/>	Value must be between 1 and 256 KB.
Delivery Delay:	<input type="text" value="0"/>	<input type="text" value="seconds"/>	Value must be between 0 seconds and 15 minutes.
Receive Message Wait Time:	<input type="text" value="0"/>	<input type="text" value="seconds"/>	Value must be between 0 and 20 seconds.

- Konfiguracja

Opóźnienie pierwszego dostarczenia wiadomości w kolejce

Default Visibility Timeout:	<input type="text" value="30"/>	seconds ▾	Value must be between 0 seconds and 12 hours.
Message Retention Period:	<input type="text" value="4"/>	days ▾	Value must be between 1 minute and 14 days.
Maximum Message Size:	<input type="text" value="256"/>	KB	Value must be between 1 and 256 KB.
Delivery Delay:	<input type="text" value="0"/>	seconds ▾	Value must be between 0 seconds and 15 minutes.
Receive Message Wait Time:	<input type="text" value="0"/>	seconds	Value must be between 0 and 20 seconds.

- Konfiguracja

Maksymalny czas oczekiwania na wiadomość w kolejce

Default Visibility Timeout:	<input type="text" value="30"/>	seconds ▾	Value must be between 0 seconds and 12 hours.
Message Retention Period:	<input type="text" value="4"/>	days ▾	Value must be between 1 minute and 14 days.
Maximum Message Size:	<input type="text" value="256"/>	KB	Value must be between 1 and 256 KB.
Delivery Delay:	<input type="text" value="0"/>	seconds ▾	Value must be between 0 seconds and 15 minutes.
Receive Message Wait Time:	<input type="text" value="0"/>	seconds	Value must be between 0 and 20 seconds.

- Relacyjne bazy danych jako usługa
 - Amazon Relational Database Service (MySQL), Microsoft SQL Azure (MS SQL), GenieDB (MySQL), Heroku PostgreSQL, ...
- Nierelacyjne bazy danych jako usługa
 - MongoHQ, MongoLab, Firebase, <https://redislabs.com>, ...
 - SimpleDB, DynamoDB



- Narzędzia integrujące usługi bazujące na chmurze obliczeniowej
 - Rozszerzenie platform integracyjnych (ESB) na poziom chmur
 - Silnik integracyjny jako usługa w chmurze
- Cechy
 - Łączniki (Connectors) – łączenie usług SaaS (łączenie chmur)
 - Integracja usług w chmurze z klasyczną architekturą
 - Prostota integracji (drag and drop)
 - Wysoka dostępność i niezawodność silnika integracyjnego
 - Pomiary i monitoring
- Platformy
 - CloudHub (MuleSoft)
 - Ponad 120 łączników
 - Mapowanie danych
 - Integracja wsadowa i czasu rzeczywistego
 - Narzędzia (Mule Studio, dashboard)



- Mobile/cloud Backend as a Service
 - Model pozwalający twórcom aplikacji mobilnych łączyć aplikacje z „zapleczem” w chmurze obliczeniowej
 - Trwały skład danych
 - Zarządzanie użytkownikami
 - Powiadomienia
 - Integracja z usługami systemów „social networking”
 - Dostępność „zaplecza” zapewniana jest:
 - dla wielu platform mobilnych (iOS, Windows Phone, Android, HTML5, Blackberry)
 - w wielu chmurach (AWS, Microsoft Azure, Rackspace)
- Stosunkowo młody segment
 - Pierwsze startupy w 2011 roku (np. Kumulos)
- Duże wymagania dotyczące liczby wywołań API



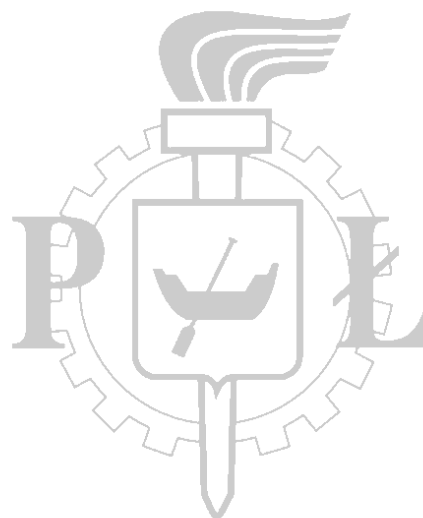
- Michael P. McGrath: Understanding PaaS, O'Reilly, 2012
- http://www.rackspace.com/knowledge_center/whitepaper/understanding-the-cloud-computing-stack-saas-paas-iaas
- <http://java.dzone.com/articles/what-platform-service-paas>
- <http://www.wired.com/insights/2014/01/talkin-bout-app-store-revolution-enabling-monetizing-api-economy/>
- <http://www.tomsitpro.com/articles/paas-providers,1-1517.html>
- Mobile Needs A Four-Tier Engagement Platform
http://d1cqpkecg5rwnz.cloudfront.net/docs/Mobile_Needs_A_Four_Tier_.pdf



wydział
elektrotechniki
elektroniki
informatyki
i automatyki

AMG.net

A Bull Group Company



Dziękujemy za uwagę