AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE   
WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I ELEKTRONIKI

KATEDRA INFORMATYKI



**Serwisy w architekturze chmury obliczeniowej**

**PRACA MAGISTERSKA**

**Michał Soboń**

mikesobon@gmail.com

**Promotor: dr inż. Piotr Nawrocki**

Kraków 2012

Oświadczam, świadomy odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszą pracę dyplomową wykonałem osobiście i samodzielnie (w zakresie wyszczególnionym we wstępie) i że nie korzystałem ze źródeł innych niż wymienione w pracy.

Spis treści

[Wstęp 5](#_Toc319956807)

[Struktura pracy 5](#_Toc319956808)

[1. Koncepcja usługi 6](#_Toc319956809)

[1.1 Serwisy w nowoczesnych systemach informatycznych 6](#_Toc319956810)

[1.2 Definicja Serwisu 6](#_Toc319956811)

[1.3 Architektura zorientowana na usługi (service oriented architecture – SOA) 6](#_Toc319956812)

[2. Architektura chmury obliczeniowej 7](#_Toc319956813)

[2.1 Koncepcja chmury obliczeniowej 7](#_Toc319956814)

[2.2 Modele chmury obliczeniowej 7](#_Toc319956815)

[Kolokacja 7](#_Toc319956816)

[Oprogramowanie jako usługa 7](#_Toc319956817)

[Platforma jako usługa 7](#_Toc319956818)

[Infrastruktura jako usługa 8](#_Toc319956819)

[Komunikacja jako usługa 8](#_Toc319956820)

[2.3 Charakterystyka wybranych implementacji chmur obliczeniowych 8](#_Toc319956821)

[Amazon EC2 8](#_Toc319956822)

[Google AppEngine 8](#_Toc319956823)

[Microsoft Azure 8](#_Toc319956824)

[Heroku 8](#_Toc319956825)

[2.4 Aplikacja jako serwis 8](#_Toc319956826)

[3. Architektura systemu udostępniającego oprogramowanie w chmurze obliczeniowej 9](#_Toc319956827)

[3.1 Analiza wymagań 9](#_Toc319956828)

[3.1.1 Wymagania funkcjonalne 9](#_Toc319956829)

[3.1.2 Wymagania niefunkcjonalne 9](#_Toc319956830)

[3.2 Opis architektury systemu 9](#_Toc319956831)

[3.2.1 Architektura portalu udostępniającego aplikacje jako usługi 9](#_Toc319956832)

[Single Sign – On 9](#_Toc319956833)

[3.2.2 Architektura aplikacji - usług 9](#_Toc319956834)

[4. Implementacja systemu 9](#_Toc319956835)

[4.1 Wykorzystane technologie 9](#_Toc319956836)

[4.1.1 Google AppEngine 9](#_Toc319956837)

[4.1.2 Microsoft Azure 9](#_Toc319956838)

[4.1.3 Amazon EC2 9](#_Toc319956839)

[4.1.4 Heroku 9](#_Toc319956840)

[4.1.5 Play! 9](#_Toc319956841)

[4.1.6 Spring 9](#_Toc319956842)

[4.2 Model architektury rozproszonej 9](#_Toc319956843)

[4.3 Implementacja portalu udostępniającego aplikacje 10](#_Toc319956844)

[4.4 Implementacja przykładowych aplikacji - usług 10](#_Toc319956845)

[5. Testy systemu 10](#_Toc319956846)

[Podsumowanie 10](#_Toc319956847)

[Wykaz skrótów 11](#_Toc319956848)

[Spis ilustracji 11](#_Toc319956849)

[Spis tabel 11](#_Toc319956850)

[Spis listingów 11](#_Toc319956851)

[Bibliografia 12](#_Toc319956852)

[Spis ilustracji i tabel 13](#_Toc319956853)

[Aneksy 13](#_Toc319956854)

# Wstęp

W obecnych czasach Internet staje się nieodłącznym atrybutem komputera. Wraz z jego rozwojem ludzie zyskali dostęp do coraz większej wiedzy o technologii przez znacząco wzrosły ich wymagania dotyczące kompleksowości posiadanych przez nich rozwiązań informatycznych. Główną przeszkodą dla użytkowników były wysokie ceny złożonego oprogramowania oraz sprzętu niezbędnego do jego poprawnego funkcjonowania – istniała spora grupa użytkowników dla których zakup był nieopłacalny.

Jednym z rozwiązań tego problemu jest dostarczenie usługi polegającej na dostępie do aplikacji (bądź ich zestawu) wraz z możliwością bezpiecznego przechowywania niezbędnych danych. Jej koszt jest zależny od intensywności użytkowania co znacząco poszerza rynek potencjalnych nabywców. Takie podejście jest korzystne zarówno dla odbiorców, jak i twórców aplikacji. Klient nie musi martwić się o zapewnienie stosownego sprzętu wraz z jego obsługą przez co może całkowicie skupić się na swoim biznesie. Dostawca oprogramowania zyskuje unifikacje środowiska, w którym działa aplikacja, co powoduje zmniejszenie ilości wersji oprogramowania wykorzystywanych przez klientów. Dodatkowo istnieje możliwość zmiany sposobu finansowania twórców oprogramowania - zamiast jednorazowej opłaty licencyjnej mogą pobierać abonament, którego wysokość zależy od stopnia wykorzystania usług. Pozwala to na zwiększenie stabilności dochodów, co skutkuje zwiększeniem możliwości rozwoju.

Niniejsza praca ma za zadanie przedstawić możliwości wykorzystania architektury opartej o serwisy działające w chmurach obliczeniowych realizującej model Software as a Service. Opisuje także architekturę, implementację oraz testy przykładowej platformy. Przykładowa aplikacja składa się z kilku podstawowych komponentów:

* Portalu zapewniającego użytkownikom dostęp do aplikacji jako usług, jest on odpowiedzialny za przechowywanie danych o użytkowniku, rozliczenia finansowe oraz zapewnie Single Sign-On
* zestawu aplikacji klienckich udostępnianych przez Portal jako usługi

Portal jest odpowiedzialny za zapewnienie użytkownikom dostępu do aplikacji klienckich w nim zarejestrowanych. Aplikacje dzięki implementacji odpowiedniego interfejsu rejestrują się w Portalu i mogą być udostępniane klientom.

## Struktura pracy

Celem niniejszej pracy jest prezentacja możliwości wykorzystania chmur obliczeniowych jako środowiska działania usług oraz stworzenie przykładowej implementacji.

Pierwszy rozdział pracy prezentuje koncepcję usług (serwisów) oraz możliwości ich zastosowania we współczesnych systemach informatycznych.

Kolejny rozdział poświęcony jest chmurom obliczeniowych, zawiera on także oprócz wyjaśnienia pojęcia chmury omówienie różnych jej modeli wraz z prezentacją istniejących implementacji.

Trzeci rozdział poświęcony jest architekturze przykładowej aplikacji. Zawiera opis wymagań funkcjonalnych i niefunkcjonalnych. Przedstawia schemat najważniejszych jej elementów oraz prezentuje zastosowane technologie.

Czwarty rozdział prezentuje testy zbudowanego sytemu (jeśli ich nie będzie zostanie złączony z rozdziałem 3.)

Podsumowanie opisuje dalsze możliwości rozwoju oraz potencjalne problemy, zawiera także wnioski.

1. Koncepcja usługi

* 1. Serwisy w nowoczesnych systemach informatycznych

Nowoczesne systemy informatyczne nie są jednolitym blokiem realizującym wymagania – są konglomeratem serwisów (usług) wykorzystywanych w miarę potrzeb. Umożliwia to łatwiejsze utrzymanie systemu oraz upraszcza jego rozwój czy rekonfigurację.

* 1. Definicja Serwisu

Serwis(usługa) jest to element oprogramowania mogący działać niezależnie, posiadający zdefiniowany interfejs, który definiuje i udostępnia realizowane funkcje. Szczegóły implementacyjne są nieistotne i niewidoczne dla klienta wykorzystującego dany serwis. Zastosowanie interfejsów umożliwia wymianę poszczególnych serwisów bez wpływania na całość systemu. Ważnym elementem każdego serwisu jest implementacja protokołu komunikacyjnego, przy użyciu którego możliwa jest komunikacja z innymi serwisami.

* 1. Architektura zorientowana na usługi (service oriented architecture – SOA)

Koncepcja tworzenia oprogramowania jako grupy zorkiestrowanych, współpracujących ze sobą usług, które spełniają wymagania użytkownika. Istotne jest słabe powiązanie pomiędzy poszczególnymi usługami – umożliwia to wielokrotne użycie serwisów. Wzajemna komunikacja pomiędzy współpracującymi serwisami jest prowadzona przy użyciu zdefiniowanego protokołu komunikacyjnego. Komunikaty przesyłane są w formacie sprecyzowanym w protokole (najczęściej oparty o XML, JSON). Aby uniknąć nadmiernych narzutów na komunikację usługi powinny być gruboziarniste.

1. Architektura chmury obliczeniowej

Chmura obliczeniowa to system komputerowy dostarczający obliczenia, składowanie danych czy infrastrukturę jako usługi. Historycznie wywodzi się z systemów gridowych. Jest zbudowany w oparciu o współdzielone, reużywalne zasoby z wykorzystaniem technik wirtualizacji umożliwiających w izolowanie danych poszczególnych użytkowników. Dostęp do chmury obliczeniowej odbywa się poprzez sieć komputerową.

* 1. Koncepcja chmury obliczeniowej
  2. Modele chmury obliczeniowej

Możemy wyróżnić kilka podstawowych modeli chmur obliczeniowych

### Kolokacja

Kolokacja – usługa polegająca na udostępnieniu miejsca w data center przeznaczonego do umieszczenia serwera stanowiącego własność klienta

### Oprogramowanie jako usługa

Oprogramowanie jako usługa (software as a service – SaaS) – najstarszy i najbardziej dojrzały model chmury obliczeniowej. Polega na udostępnieniu użytkownikowi dostępu do aplikacji znajdującej się gdzieś w sieci komputerowej. Pozwala mu to na korzystanie z produktu kiedy chce i gdzie chce, dodatkowo odbiorca nie musi troszczyć się o sprzęt i oprogramowanie niezbędne do działania aplikacji - należy to do odpowiedzialności dostawcy. Ewentualna opłata za użytkowanie może być naliczana w zależności od dokonanej ilości operacji, bądź abonamentowo za określony czas. Model SaaS posiada pewne niedogodności jak obawy o bezpieczeństwo danych czy możliwość nadmiernej ingerencji dostawcy. Przykładową komercyjną platformą SaaS jest Salesforce.com udostępniająca aplikacje do zarządzania.

### Platforma jako usługa

Platforma jako usługa (platform as a service - PaaS) – jest to kombinacja platformy deweloperskiej oraz istniejących rozwiązań z zakresu chmur obliczeniowych. Zapewnia infrastrukturę niezbędną do wdrożenia aplikacji stworzonych w oparciu o dostarczone API, jak również umożliwia rozwijanie istniejących aplikacji. Istotną cechą jest dynamiczna skalowalność – pozwala to skupić się twórcom aplikacji na funkcjonalności produktu, za dostarczenie niezbędnych zasobów odpowiedzialny jest dostawca. Przykładowe implementacje to Google AppEngine czy Microsoft Azure.

### Infrastruktura jako usługa

Infrastruktura jako usługa (infrastructure as a service - IaaS) – dostarcza infrastrukturę, jest platformą opartą o wirtualizację. Przeznaczona jest dla klientów posiadających w swoich szeregach specjalistów odpowiedzialnych za tworzenie i zarządzanie systemami informatycznymi, jednakże nieposiadających (bądź nie chcących posiadać) infrastruktury. Stworzenie własnego data center jest drogą inwestycją, obniżającą zdolność firmy do dynamicznego zarządzania kosztami. IaaS pozwala elastycznie dostosowywać infrastrukturę IT do bieżących potrzeb przedsiębiorstwa co powoduje zmniejszenie strat wynikających z istnienia niewykorzystywanych zasobów. Za przykład może posłużyć Amazon EC2 czy GoGrid.

### Komunikacja jako usługa

* Komunikacja jako usługa (Communication as a service – CaaS) – platforma zapewniająca środowisko telekomunikacyjne niezbędne dla funkcjonowania organizacji bez konieczności inwestowania w drogi sprzęt.
  1. Charakterystyka wybranych implementacji chmur obliczeniowych

Charakterystyka najpopularniejszych rozwiązań

### Amazon EC2

### Google AppEngine

### Microsoft Azure

### Heroku

* 1. Aplikacja jako serwis

Integracja z platforma udostepniajaca,

protokól komunikacyjny

Zasady uzyskiwania dostępu do aplikacji,

koszty uzytkowania,

1. Architektura systemu udostępniającego oprogramowanie w chmurze obliczeniowej
   1. Analiza wymagań

Głównym elementem systemu jest Portal udostępniający aplikacje jako usługi. Jego celem jest przechowywanie danych o użytkownikach, zbieranie informacji o wykorzystaniu usług. Udostępnia on interfejs umożliwiający aplikacjom klienckim dostęp do posiadanych danych oraz stanowi punkt integracji dla usługi single sign -on . Do Portalu (dzięki udostępnionemu interfejsowi) podłączają się aplikacje klienckie, które stanowią udostępniane na żądanie użytkownika oprogramowanie biznesowe.

* + 1. Wymagania funkcjonalne

### Portal udostępniający aplikacje

Podstawowe wymagania funkcjonalne:

* Tworzenie, edycja i usuwanie użytkowników i administratorów
* Dodawanie i usuwanie aplikacji udostępnianych w Portalu
* Wyświetlanie listy dostępnych aplikacji wraz z opisem
* Zgłaszanie żądań o dostęp do aplikacji
* Monitorowanie wykorzystania aplikacji przez użytkowników w systemie punktowym
* Wyświetlanie listy aplikacji użytkownika
* Udostępnienie interfejsu umożliwiającego rejestrację aplikacji klienckich w Portalu
* Udostępnienie interfejsu umożliwiającego aplikacjom naliczanie kosztów użytkowania aplikacji,

### Przykładowa aplikacja kliencka 1 – Staff Manager

Aplikacja stanowi wirtualną kartotekę pracowników firmy, która przechowuje podstawowe dane pracownika.

Podstawowe wymagania funkcjonalne:

* Mianowanie użytkownika administratorem aplikacji
* Single sign-on z Portalem
* Dodawanie, edycja oraz usuwanie informacji personalnych o pracownikach – imię, nazwisko, adres zamieszkania, PESEL, NIP, data zatrudnienia, uwagi

### Przykładowa aplikacja kliencka 2 – Warehouse Manager

Aplikacja ma za zadanie uproszczenie zarządzania magazynem – przechowuje informacje o produktach znajdujących się na stanie. Dodatkowo przechowywana jest informacja o ilości i lokalizacji towarów.

Podstawowe wymagania funkcjonalne:

* Mianowanie użytkownika administratorem aplikacji
* Single-sin-on z Portalem
* Dodawanie, edycja oraz usuwanie informacji o produkcie. Są to nazwa, waga, rozmiar opakowania, uwagi oraz jego ilość i aktualne położenie w magazynie
  + 1. Wymagania niefunkcjonalne

### Portal udostępniający aplikacje

Podstawowe wymagania niefunkcjonalne:

* Implementacja z wykorzystaniem Play! Framework
* Wdrożenie na platformę Heroku
* Wykorzystanie bazy danych oferowanej przez platformę

### Aplikacje klienckie (wersja ogólna – podzielić na Staf i Warehouse Manager)

* Implementacja z wykorzystaniem Django/AppEngine
* Implementacja z wykorzystaniem innej chmury obliczeniowej
  1. Opis architektury systemu
     1. Architektura portalu udostępniającego aplikacje jako usługi

### Single Sign – On

* + 1. Architektura aplikacji - usług

1. Implementacja systemu
   1. Wykorzystane technologie
      1. Google AppEngine
      2. Microsoft Azure
      3. Amazon EC2
      4. Heroku
      5. Play!
      6. Spring
   2. Model architektury rozproszonej

Diagram przedsawiajacy portal oraz aplikacje hostowane w rożnych cloudach

* 1. Implementacja portalu udostępniającego aplikacje
  2. Implementacja przykładowych aplikacji - usług

1. Testy systemu

Podsumowanie

# Wykaz skrótów

GAE – Google App Engine

IaaS – Infrastructure as a Service

PaaS – Platform as a Service

SaaS - Software as a Service

SOA – Service Oriented Architecture

SSO – Single Sign-On

# Spis ilustracji

# Spis tabel

# Spis listingów

# Bibliografia

* Service-Oriented Cloud Computing Architecture <http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F5498003%2F5501429%2F05501650.pdf%3Farnumber%3D5501650&authDecision=-203>
* Cloud computing — Issues, research and implementations <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4588381>
* Automatyczna skalowalność aplikacji na platformach Amazon WS i Google App Engine <http://www.ki.agh.edu.pl/sites/default/files/publikacje/164/amazon.pdf>
* SaaAS - The mobile agent based service for cloud computing in internet environment <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5582438>
* A Model of Scheduling Optimizing for Cloud Computing Resource Sevices Based on Buffer-pool Agent <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5576181>
* Service Replication Strategies with MapReduce in Clouds <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5741385>
* Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing <http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F71%2F5756300%2F05719609.pdf%3Farnumber%3D5719609&authDecision=-203>
* Ubiquitous cloud: Managing service resources for adaptive ubiquitous computing <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5766853>
* From outsourcing to Cloud computing: Evolution of IT services <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5996009>
* Service-Oriented Computing and Cloud Computing: Challenges and Opportunities <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5617062>
* <http://www.cloudtweaks.com/2011/02/cloud-computing-for-dummies-saas-paas-iaas-and-all-that-was/>
* Google App Engine Campfire One Transcript <http://code.google.com/intl/pl-PL/appengine/articles/cf1-text.html>
* Amazon EC2 Overview <http://aws.amazon.com/ec2/>
* Windows Azure <http://www.microsoft.com/windowsazure/features/>
* iCloud <http://en.wikipedia.org/wiki/ICloud>
* Nimbus <http://www.nimbusproject.org/>
* Eucalyptus <http://www.eucalyptus.com/>
* Eucalyptus System for Clouds <http://fatcat.ftj.agh.edu.pl/~i7szczep/koala.pdf>
* OpenNebula <http://www.opennebula.org/>
* OpenStack [http://openstack.org](http://openstack.org/)
* VM WebSpere<http://www.ki.agh.edu.pl/technologie/vmware-vsphere>

# Spis ilustracji i tabel

# Aneksy