

# Metodyka tworzenia portali biznesowych

## WYKŁAD 7

## **Chmura obliczeniowa**

- Chmura obliczeniowa (ang. *Cloud Computing* - dosłownie obliczenia / przetwarzanie w chmurze) jest bardzo szybko rozwijającym się modelem polegającym na użytkowaniu usług dostarczanych przez sieć internetową.



# Chmura obliczeniowa w sieci Internet

- połączone ze sobą komputery, serwery, bazy danych, centra składowania danych, itp.

# Klient

- **Cienki klient** – typ klienta, w przypadku którego zadania realizowane są głównie na serwerze, a rola użytkownika polega przede wszystkim na przekazywaniu żądań do serwera za pomocą dedykowanej aplikacji, np. przeglądarki internetowej.
- **Gruby klient** – typ klienta, w przypadku którego realizacja zadań odbywa się częściowo na komputerze użytkownika (klienta), a częściowo na serwerze.

## **Chmura obliczeniowa**

- Chmura obliczeniowa - usługi (serwisy) obliczeniowe oferowane przez zewnętrzne podmioty i dostępne na życzenie w dowolnym momencie, skalujące się dynamicznie w odpowiedzi na zmieniające się zapotrzebowanie.

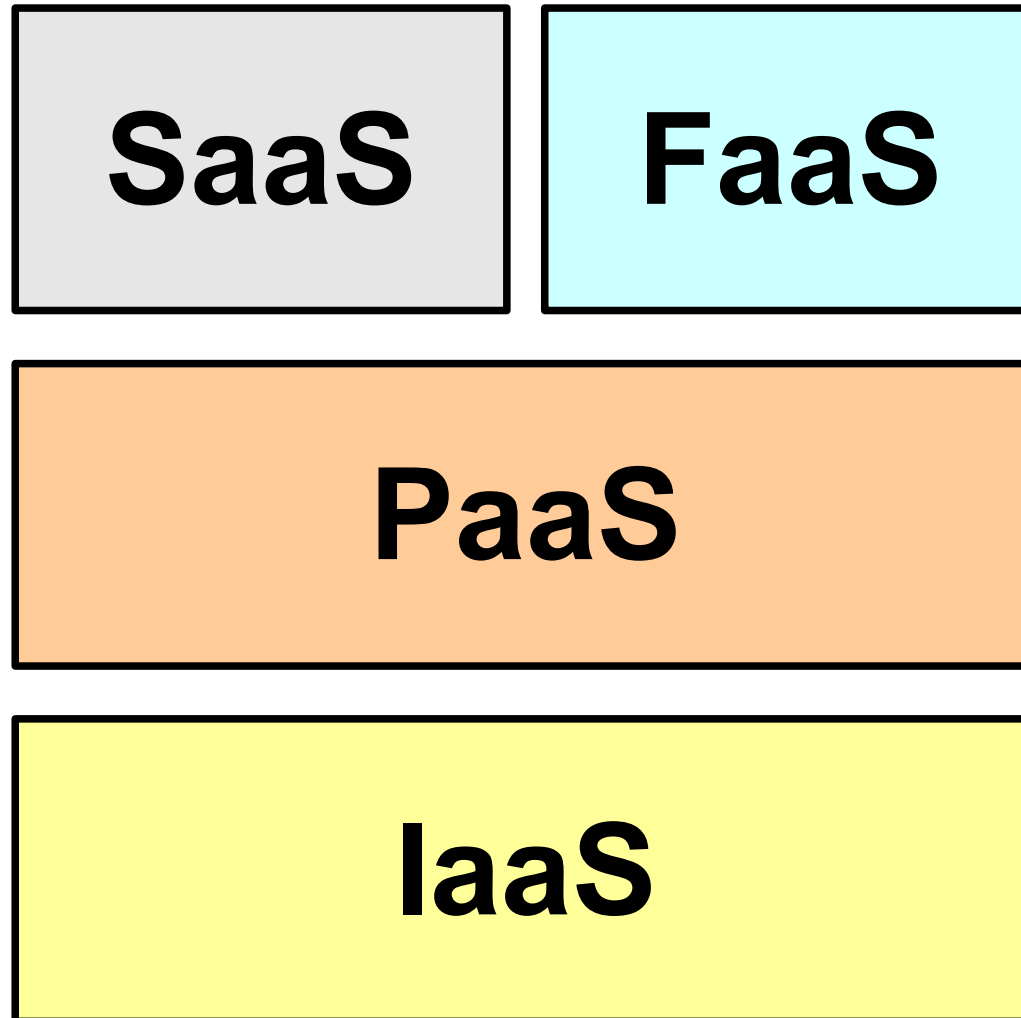
## Klasyfikacja warstw chmury

- IaaS (*Infrastructure as a Service*) - Infrastruktura jako usługa - zapewnione są: sprzęt oraz obrazy maszyny wirtualnej dla różnych systemów operacyjnych.
- PaaS (*Platform as a Service*) - Platforma jako usługa - udostępniana jest platforma do pisania aplikacji.
- SaaS (*Software as a Service*) - Oprogramowanie jako usługa - aplikacje dostępne na życzenie.

## **Klasyfikacja warstw chmury**

- FaaS (*Framework as a Service*) - Szkielet jako usługa – środowisko przyległe do SaaS pozwalające programistom na rozszerzanie gotowych funkcjonalności aplikacji SaaS.
- CaaS (*Communications as a Service*) – Komunikacja jako usługa – usługi komunikacyjne.

# Klasyfikacja warstw chmury





## **Zasady definiujące przetwarzanie w chmurze**

- Pula zasobów obliczeniowych dostępnych dla każdego zarejestrowanego użytkownika.
- Wirtualizacja zasobów obliczeniowych w celu maksymalizacji wykorzystania sprzętu.
- Elastyczne skalowanie, w górę lub w dół, w zależności od potrzeb.
- Automatyczne tworzenie nowych wirtualnych maszyn oraz usuwanie maszyn istniejących.
- Naliczanie opłat jedynie za wykorzystane zasoby.

## **Rodzaje chmur**

- Chmury publiczne - usługi IT dostawcy oferowane są wielu odbiorcom.
- Chmury prywatne - wewnętrzne zasoby firmy (centrum danych) nie są udostępniane publicznie.
- Chmury hybrydowe.

# Chmura prywatna

- ZALETY: *Na podstawie: Wdrożenia w chmurze. Computerworld 33/2012*
  - Określona lokalizacja danych
  - Infrastruktura w obrębie sieci firmowej
  - Niezależność od łączy internetowych
  - Zarządzanie chmurą przez Dział IT w firmie
  - Określona odpowiedzialność za realizację usług i bezpieczeństwo danych
  - Możliwość dostosowania zasobów do potrzeb firmy ze wcześniejszym planowaniem rozwoju
  - Realizacja usług nie oferowanych przez zewnętrznych dostawców

# Chmura prywatna

- ZALETY: *Na podstawie: Wdrożenia w chmurze. Computerworld 33/2012*
  - Rozliczanie kosztów i zużycia w obrębie firmy
  - Wiarygodny backup
  - Dostosowanie do potrzeb firmy

# Chmura prywatna

- WADY: *Na podstawie: Wdrożenia w chmurze. Computerworld 33/2012*
  - Duże koszty inwestycyjne związane z budową chmury
  - Wyższe koszty utrzymania niż przy chmurze publicznej
  - Usługi dostosowane przede wszystkim do dostępu z sieci firmowej
  - Czasami brak standardowych usług oferowanych przez podmioty zewnętrzne ze względu na nieopłacalność ich budowy w chmurze prywatnej

# Chmura publiczna

- ZALETY: *Na podstawie: Wdrożenia w chmurze. Computerworld 33/2012*
  - Praktycznie zerowa inwestycja w sprzęt
  - Łatwość wdrażania niewielkich projektów
  - Proste rozliczanie kosztów
  - Dostęp do usług z różnych miejsc
  - Możliwość migracji obciążenia między dostawcami w przypadku standardowych usług
  - Możliwość realizacji kontraktów krótkoterminowych
  - Możliwość obsługi planowanych szczytów obciążenia

# Chmura publiczna

- WADY: *Na podstawie: Wdrożenia w chmurze. Computerworld 33/2012*
  - Przeciwwskazania związane z przechowywaniem danych na zewnątrz
  - Brak odpowiednich umów SLA (umów o gwarantowanym poziomie świadczenia usług)
  - Uzależnienie od łączy internetowych
  - Wpływ obciążeń innych klientów na pracę własnych aplikacji
  - Szybki wzrost kosztów operacyjnych przy wdrożeniach o dużej skali
  - Możliwość silnego uzależnienia od dostawcy usług

# Chmura publiczna

- WADY: *Na podstawie: Wdrożenia w chmurze. Computerworld 33/2012*
  - Czasami wątpliwy backup



# Chmura hybrydowa

- ZALETY: *Na podstawie: Wdrożenia w chmurze. Computerworld 33/2012*
  - Dynamiczne dostosowanie do potrzeb firmy
  - Bardzo dobre podejście zarówno do małych jak i dużych projektów
  - Możliwość pozostawienia krytycznych danych i usług w firmowym IT
  - Bardzo dobra (często automatyczna) obsługa szczytów obciążenia poza chmurą prywatną
  - Możliwość hostowania aplikacji tam, gdzie jest to uzasadnione przy uwzględnieniu kosztów i bezpieczeństwa

# Chmura hybrydowa

- WADY: *Na podstawie: Wdrożenia w chmurze. Computerworld 33/2012*
  - Niezbędne inwestycje (mniejsze niż dla chmury prywatnej)
  - Niezbędna integracja z zasobami zewnętrznego dostawcy
  - Utrudnione rozliczanie kosztów
  - Podatność na awarie u zewnętrznych dostawców lub łączy internetowych (mniejsza niż przy chmurze publicznej)

## **Chmura Amazon EC2**

- Chmura Amazon EC2 należy do chmur klasy IaaS (Infrastruktura jako usługa).
- Została udostępniona w 2006 roku.
- Jest największą chmurą tej klasy.
- Posiada niewielkie wsparcie automatyczne dla skalowania i korekcji błędów (nakłada to większe wymagania na aplikacje - elastyczność musi zostać oprogramowana w oparciu o API chmury).

## **Chmura Amazon EC2**

- Programista może wybrać dowolny język programowania.
- Programista posiada całkowitą kontrolę nad aplikacją.
- W chmurze konieczna jest ręczna obsługa, ale w zamian otrzymuje się odpowiednik maszyny fizycznej, na której można zainstalować system i wszystkie wymagane komponenty.

## Chmura Amazon EC2

- Najpopularniejszą konfiguracją chmury jest LAMP:

**L** - System operacyjny: **Linux**

**A** - Serwer WWW: **Apache**

**M** - System zarządzania relacyjnymi bazami danych: **MySQL**

**P** - Język skryptów po stronie serwera: **PHP**

## **Chmura Amazon EC2**

- API udostępniane jest w dwóch wersjach:
  - wersji opartej o protokół SOAP,
  - wersji opartej o protokół HTTP.

## **Chmura Microsoft Azure**

- Chmura Microsoft Azure należy do chmur klasy IaaS (Infrastruktura jako usługa).
- Oferuje także dodatkowe usługi charakterystyczne dla chmur klasy PaaS (Platforma jako usługa).
- Wiele aplikacji Microsoft przenoszonych jest do chmury, co czyni chmurę również chmurą klasy SaaS (Oprogramowanie jako usługa).

## **Chmura Microsoft Azure**

- System operacyjny: Windows Server 2008 zmodyfikowany pod kątem pracy w chmurze.
- Środowisko wirtualne działa pod kontrolą narzędzia monitorującego Microsoft Hypervisor uruchamianego na sprzęcie w centrach danych Microsoft.



# Chmura Microsoft Azure

- Warstwy systemowe:
  - składowanie danych
  - kontroler odpowiadający za zarządzanie modelowaniem, wdrażaniem i realizacją zapotrzebowania na zasoby
  - maszyna wirtualna
  - środowisko programistyczne

## **Chmura Microsoft Azure**

- Aplikacje tworzone mogą być w środowisku programistycznym Visual Studio z wykorzystaniem bibliotek .NET.
- Aplikacje kompilowane są do środowiska Common Language Runtime niezależnego od języka.

# Chmura Microsoft Azure

- Podstawowe zestawy usług:
  - Live Services
  - SQL Services
  - .NET Services
  - Share Point Services
  - CRM Services

## **Chmura Google App Engine**

- Chmura Google App Engine należy do chmur klasy PaaS (Platforma jako usługa).
- Przeznaczona jest do uruchamiania tradycyjnych aplikacji internetowych.
- Chmura najlepiej sprawdza się w przypadku aplikacji typu „żądanie-odpowiedź”.
- Google App Engine wykorzystuje model CGI.

# Chmura Google App Engine

Podstawowe języki programowania używane w chmurze Google App Engine



**Java**



**Python**



**PHP**

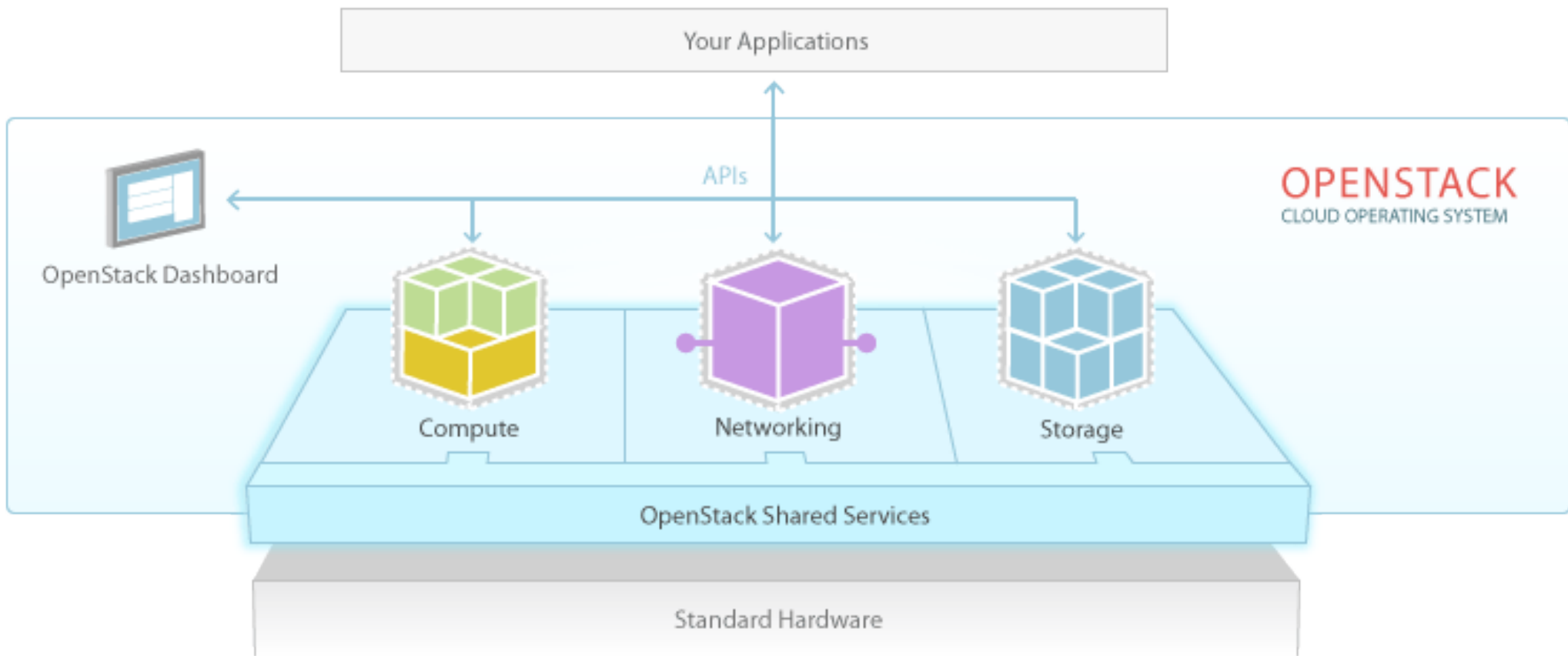


**Go**

# Chmura Google App Engine

- Środowisko programistyczne:
  - sandbox (piaskownica) - izoluje aplikację we własnym, bezpiecznym środowisku niezależnym od sprzętu, systemu operacyjnego i fizycznej lokalizacji na serwerze
  - środowisko uruchomieniowe języka Java (obsługiwane przy użyciu standardu JavaServlet)
  - środowisko uruchomieniowe języka Python
  - magazyn danych - rozproszona usługa składowania danych obsługująca język zapytań i transakcje

# OpenStack - "chmurowy" system operacyjny



Źródło: <http://www.openstack.org/software/>

# OpenStack - wprowadzenie

<http://www.openstack.org/software/>



# OpenStack

- OpenStack - otwarte i wolne oprogramowanie (rozprowadzane na licencji Apache Licence 2.0) służące do tworzenia publicznych i prywatnych chmur obliczeniowych w modelu IaaS (*Infrastructure as a Service*).
- OpenStack jest alternatywą dla innych rozwiązań IaaS.
- Projekt OpenStack wspierany jest przez wiele firm (ponad 200), m.in.: IBM, HP, SUSE Linux, Cisco, Dell.

# OpenStack

- Początki projektu OpenStack sięgają roku 2010 (rozpoczęcie prac przez NASA i Rackspace Inc. nad nową technologią dla publicznych i prywatnych chmur obliczeniowych).
- OpenStack napisany został przede wszystkim w języku Python.
- OpenStack posiada API zgodne z Amazon EC2.

# OpenStack

- Oficjalny serwis WWW projektu OpenStack:

<http://www.openstack.org>

# OpenStack

- Zalety OpenStack:
  - pełna kontrola wszystkich elementów związanych z zarządzaniem infrastrukturą (procesory, pamięć, przestrzeń dyskowa),
  - elastyczne zarządzanie siecią (możliwość dostosowania infrastruktury do konkretnych potrzeb),
  - 3-krotna redundancja danych,
  - natychmiastowe tworzenie nowych instancji,
  - rozbudowane API,

# OpenStack

- Zalety OpenStack (cd.):
  - zarządzanie bazą obrazów systemowych,
  - rozbudowana możliwość zarządzania instancjami,
  - skalowalność instancji i systemu magazynowania (storage),
  - możliwość definiowania dostępu do instancji,
  - zarządzanie instancjami z poziomu WWW,
  - możliwość tworzenia kopii migawkowych instancji,
  - zaawansowane statystyki użycia.

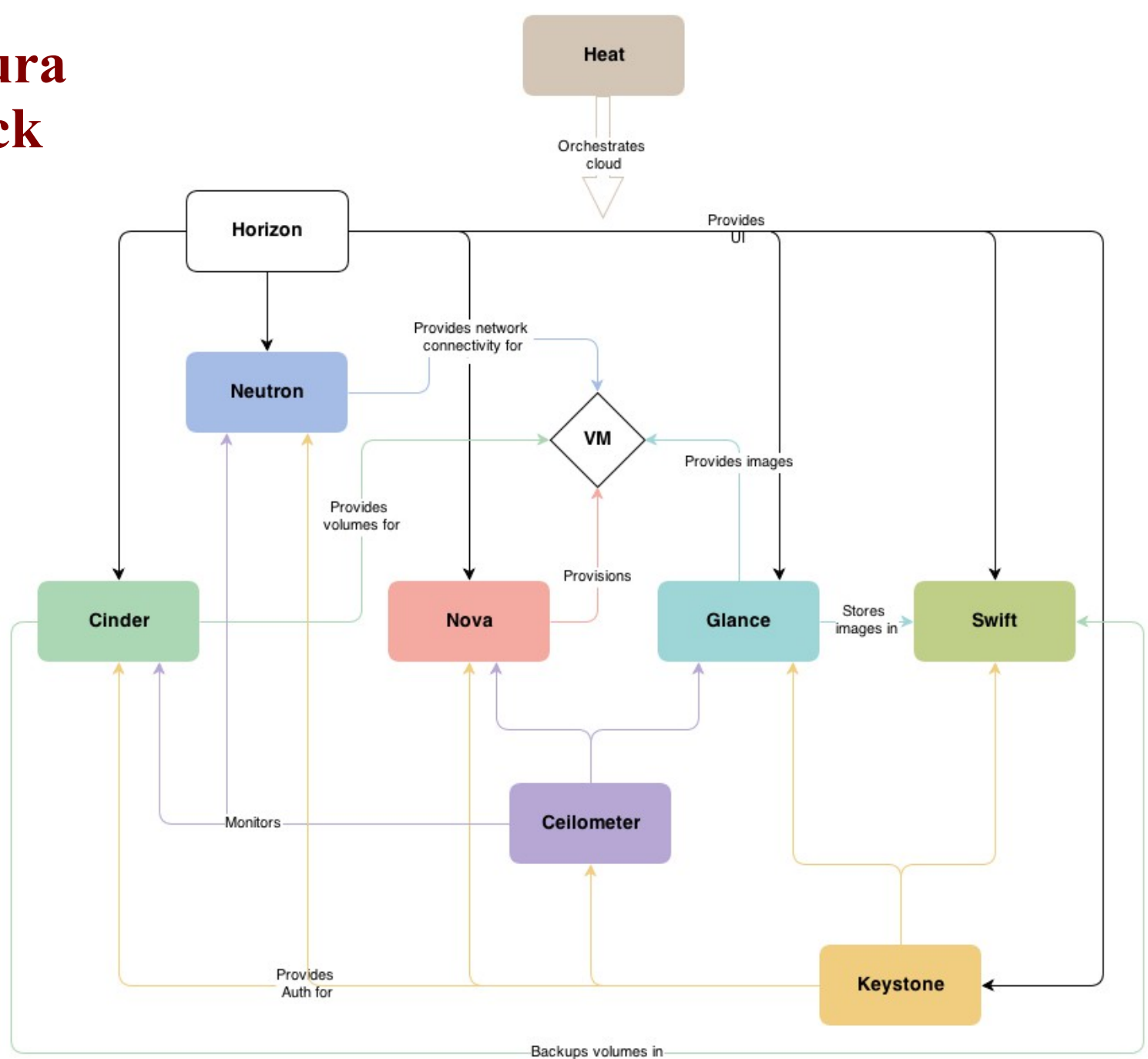
# OpenStack

- Podstawowe komponenty OpenStack:
  - **Compute (Nova)** - komponent odpowiedzialny za realizację obliczeń (m.in. przydzielanie zasobów).
  - **Storage (Swift)** - skalowalny, redundantny system magazynowania (storage).
  - **Block Storage (Cinder)** - blokowy system magazynowania.
  - **Networking (Neutron)** - system zarządzania siecią.
  - **Orchestration (Heat)** - system zarządzania aplikacjami.

# OpenStack

- Podstawowe komponenty OpenStack (cd.):
  - **Dashboard (Horizon)** - graficzny interfejs administratorów i użytkowników.
  - **Identity Service (Keystone)** - system kontroli dostępu do usług.
  - **Image Service (Glance)** - system zarządzania obrazami dysków i serwerów.
  - **Telemetry (Ceilometer)** - system rozliczania.
  - **Database (Trove)** - system relacyjnych i nierelacyjnych baz danych jako usługi.

# Architektura OpenStack



Źródło: <http://docs.openstack.org/admin-guide-cloud/content/conceptual-architecture.html>