Programowanie i wdrażanie aplikacji sieciowych w języku Python

Marcin Rataj

Politechnika Lubelska
Wydział Elektrotechniki i Informatyki

czerwiec 2010

Plan prezentacji

Cel pracy

1. Przedstawienie języka Python oraz wybranych, pokrewnych technologii jako wydajnego, darmowego i otwartego środowiska do tworzenia aplikacji sieciowych.
2. Zapoznanie czytelnika ze środowiskiem programistycznym przy pomocy konkretnych przykładów.
3. Utworzenie i wdrożenie aplikacji demonstrującej omawiane zagadnienia.

Plan pracy

Język Python

Wieloparadygmatowy język ogólnego użytku stawiający na oszczędną składnię i czytelny kod.

Cechy

- dojrzały i sprawdzony,
- prosta, przejrzysta i czytelna składnia,
- wspiera programowanie *obiektowe*, *strukturalne* i *funkcyjne*,
- interpretowany (+tryb interaktywny),
- dynamiczny system typów,
- automatyczne zarządzanie pamięcią,
- bogaty wybór otwartych bibliotek.

Framework Django

Napisany w Pythonie szkielet wysokiego poziomu, służący do tworzenia aplikacji internetowych.

Cechy

- projekt jest zwykłym *pakietem* języka Python,
- budowa aplikacji w oparciu o zasoby URI,
- własny język szablonów,
- elastyczny system cache,
- modularność,
- *batteries included*.

Baza CouchDB

Baza danych zorientowana na dokumenty.
Alternatywa dla baz relacyjnych.

Cechy

- wykorzystanie istniejących standardów (HTTP, JSON),
- wysoka współbieżność (brak blokad),
- dynamiczna struktura dokumentów (brak schematu),
- odporność na awarie,
- *B-drzewa* jako wewnętrzna struktura danych,
- dwukierunkowa replikacja.

Przejrzyste środowisko programistyczne

Praca na dokumentach bazy danych w sposób naturalny dla podstawowych struktur języka.

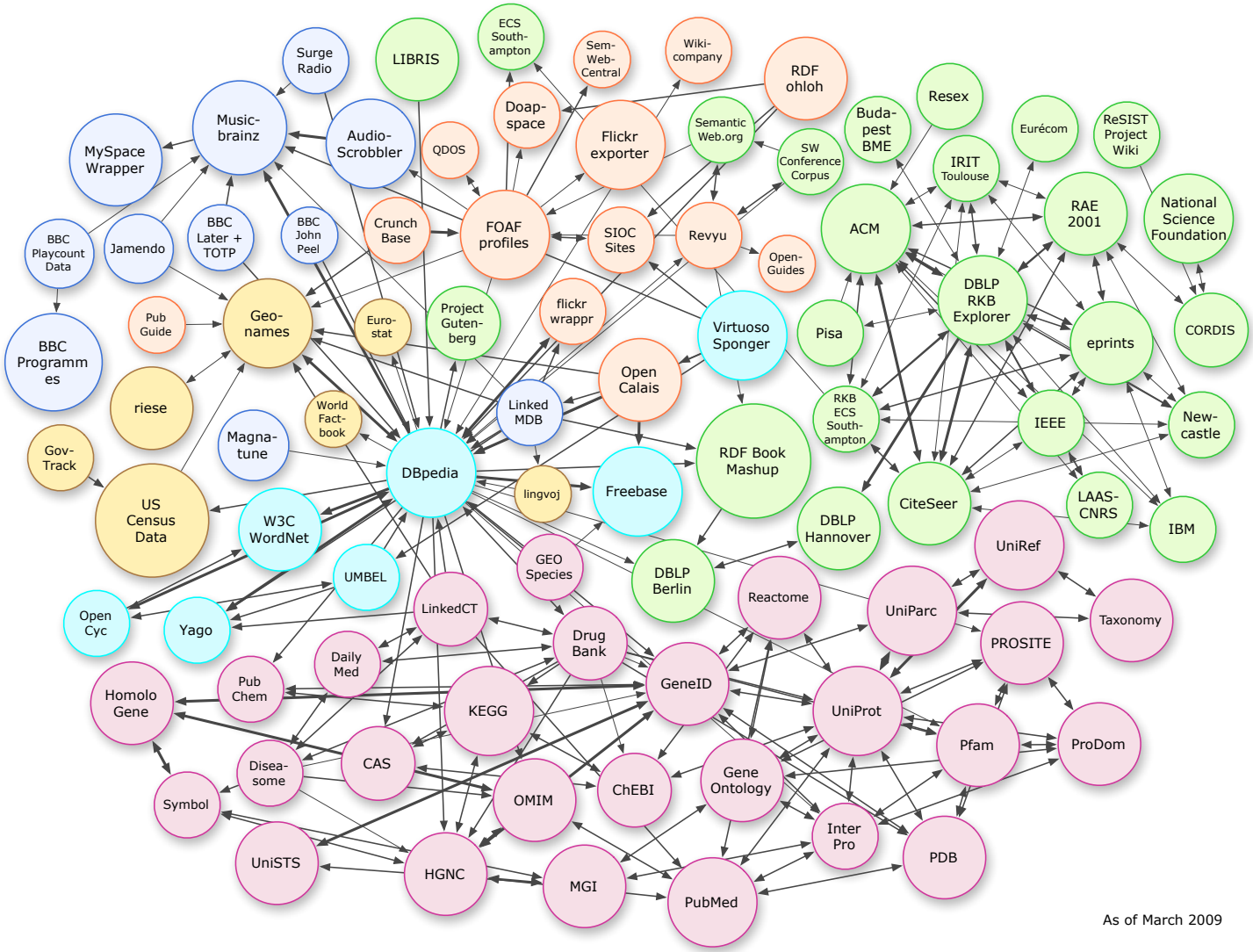
Czytelny kod, elastyczne struktury danych

```
artist = CachedArtist.get_or_create(id)
if 'name' not in artist:
    artist.name     = 'Tool'
    artist['type'] = 'group'
    artist.dates    = {'from': '1999'}
    artist.save()
```

Dwie ważne zmiany ostatniej dekady

1. treść statyczna $\longrightarrow$ *dynamiczna*
2. aplikacja natywna $\longrightarrow$ *internetowa*

Internet jako otwarta baza danych



Źródło: `http://richard.cyganiak.de/2007/10/lod/`

Czym jest Web API?

Interfejs komunikacji gwarantujący:

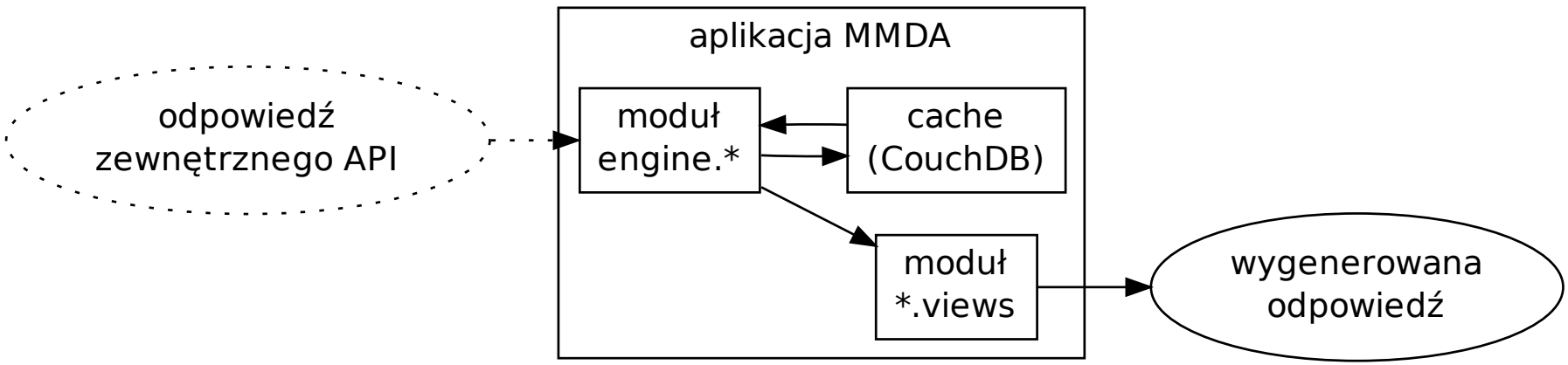
- stabilność oraz niezmienność,
- semantyczną reprezentację danych,
- przejrzystość *(dokumentacja)*,
- interoperacyjność.

Najczęściej spotykane metodyki

SOAP komunikacja przy użyciu zapytań XML-SOAP

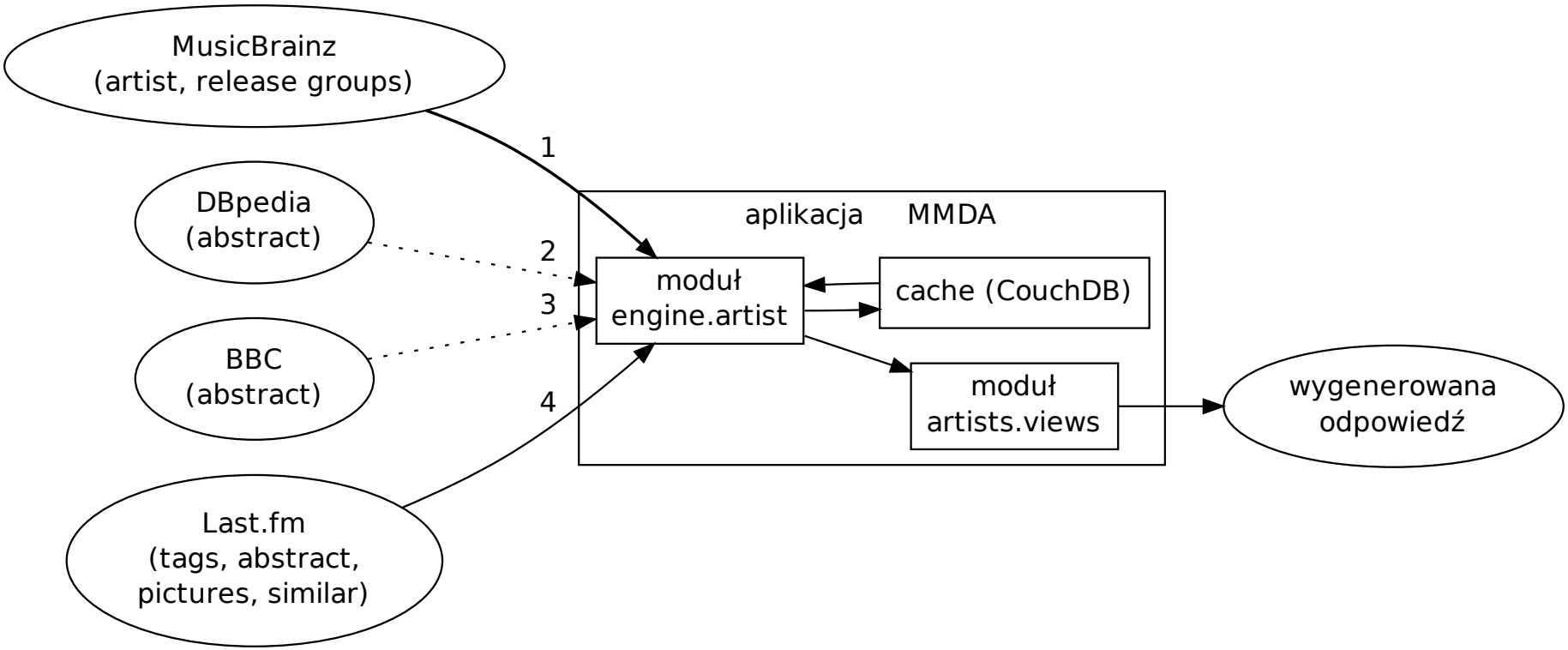
REST wywoływanie zasobów identyfikowanych przez URI

MMDA: agregacja i buforowanie treści



Źródło: opracowanie własne

Przykład: profil artysty

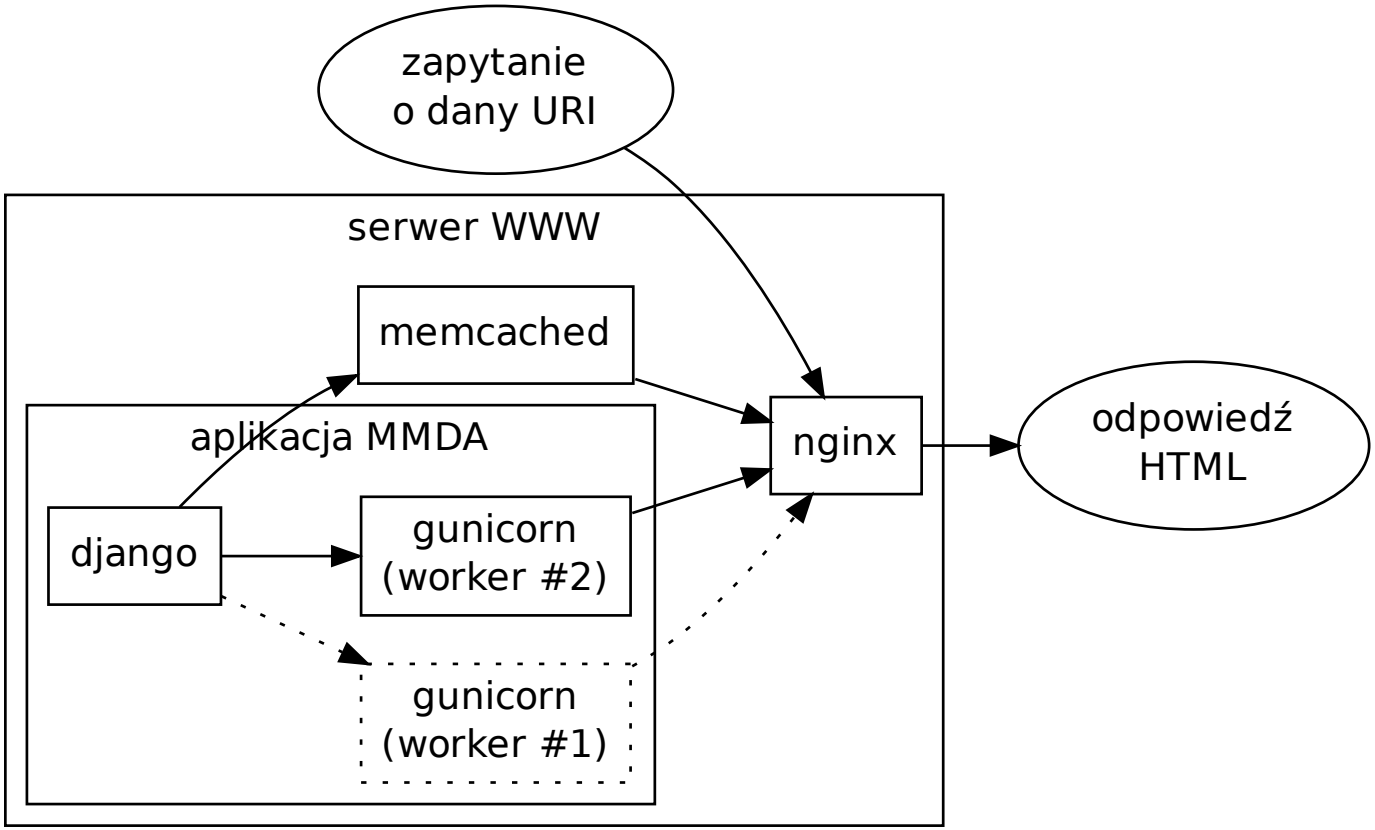


Źródło: opracowanie własne

Prezentacja treści: profil artysty

Źródło: `http://music.local/artist/chuan-jing-xian-ci/c50e52e1-01f4-494d-b1ef-376fe3581d90/`

Wdrożenie: Nginx + Gunicorn + Memcached



Źródło: opracowanie własne

Wdrożenie: wzrost wydajności

Test: obciążenie danego zasobu

Pula 1000 żądań przy zachowaniu 100 równoczesnych połączeń.

Średnia ilość obsłużonych żądań na sekundę:

3x Gunicorn: 2,55 $\longrightarrow$ *3,33*

3x Gunicorn + Memcached: 3,33 $\longrightarrow$ *29,17*

Memcached: 3,33 $\longrightarrow$ *709,11*

Podsumowanie i wnioski

1. Praca realizuje postawione cele.
2. Umożliwia zapoznanie się z wiedzą teoretyczną popartą praktycznymi przykładami.
3. Opisane rozwiązania i narzędzia stanowią łatwe, szybkie, spójne i czytelne środowisko programistyczne.
4. Architektura wdrożenia ma istotny wpływ na wybrane aspekty wydajności aplikacji.

Koniec

Dziękuję za uwagę.