



Cross application notification system

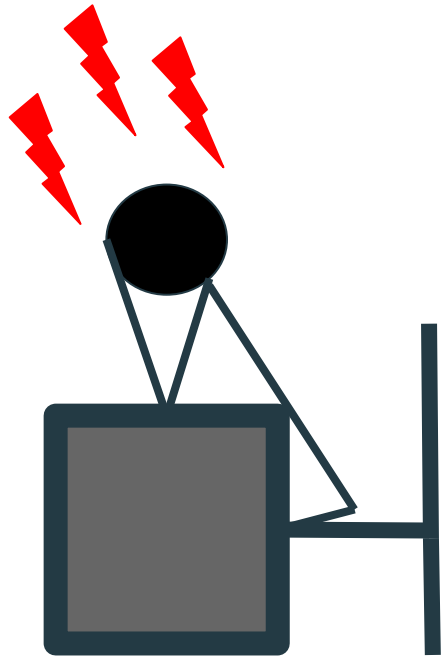
Wieloplatformowy system przesyłania i agregacji
powiadomień



Część 1

Zakres problemów jakie porusza
system

DLACZEGO?



Cross application
notification system



Duża ilość serwisów do których jesteśmy podłączeni lub które chcemy monitorować.
Czasem są to serwisy które nie udostępniają aplikacji do przeglądania powiadomień jakie generują (np. logi z własnych aplikacji webowych itp.).



Każdy z nich przesyła wiele informacji, które są dla nas istotne lub których nie chcemy przegapić.



Dostęp do powiadomień uzyskujemy tylko po zalogowaniu do konkretnych systemów czy odczytów konkretnych logów.



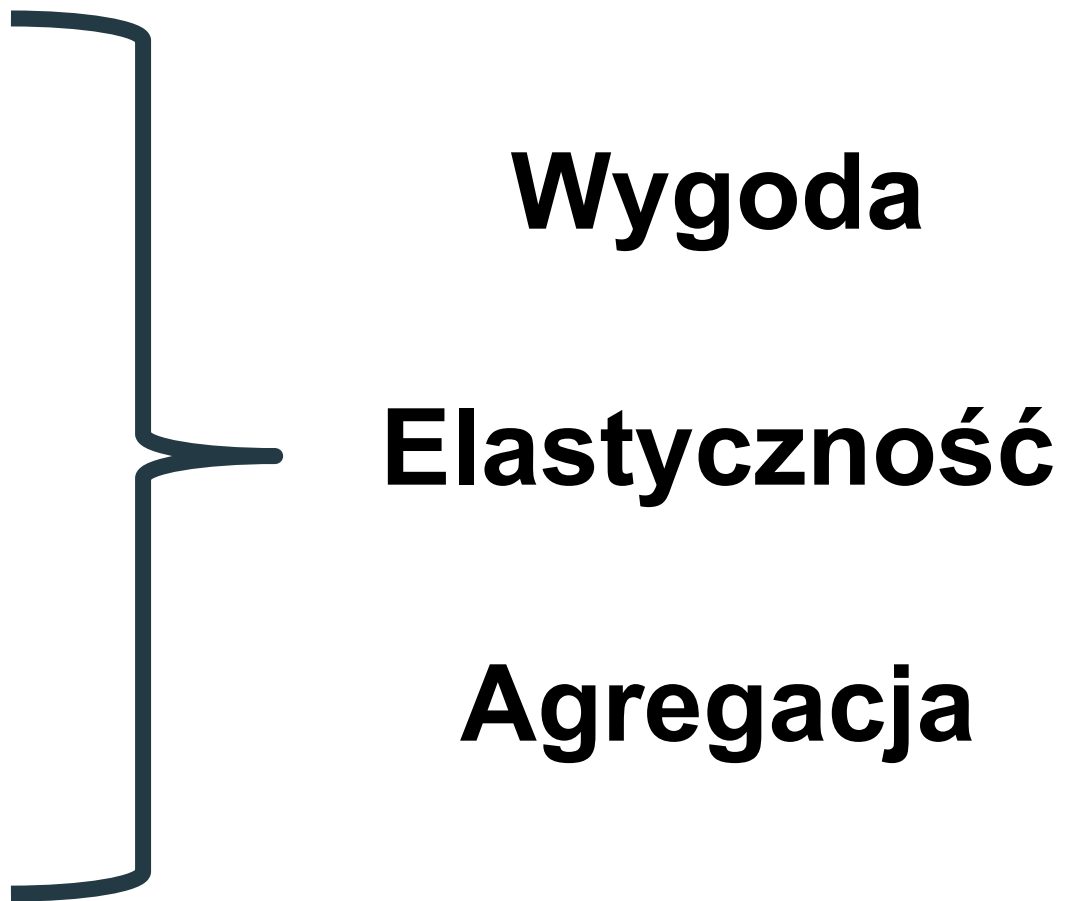
Informacje dla nas ważne często przeplatają się z tymi które dużego znaczenia dla nas nie mają.



WYGODA | ELASTYCZNOŚĆ | AGREGACJA

Cross application notification system umożliwia:

- Dostęp do twoich wszystkich notyfikacji w jednym miejscu!
- Brak straty czasu na przełączanie się między serwisami w celu odczytu informacji!
- Segregację notyfikacji według źródeł!
- Agregację powiadomień w taki sposób abyś nie stracił żadnych cennych informacji!



Cross application notification system

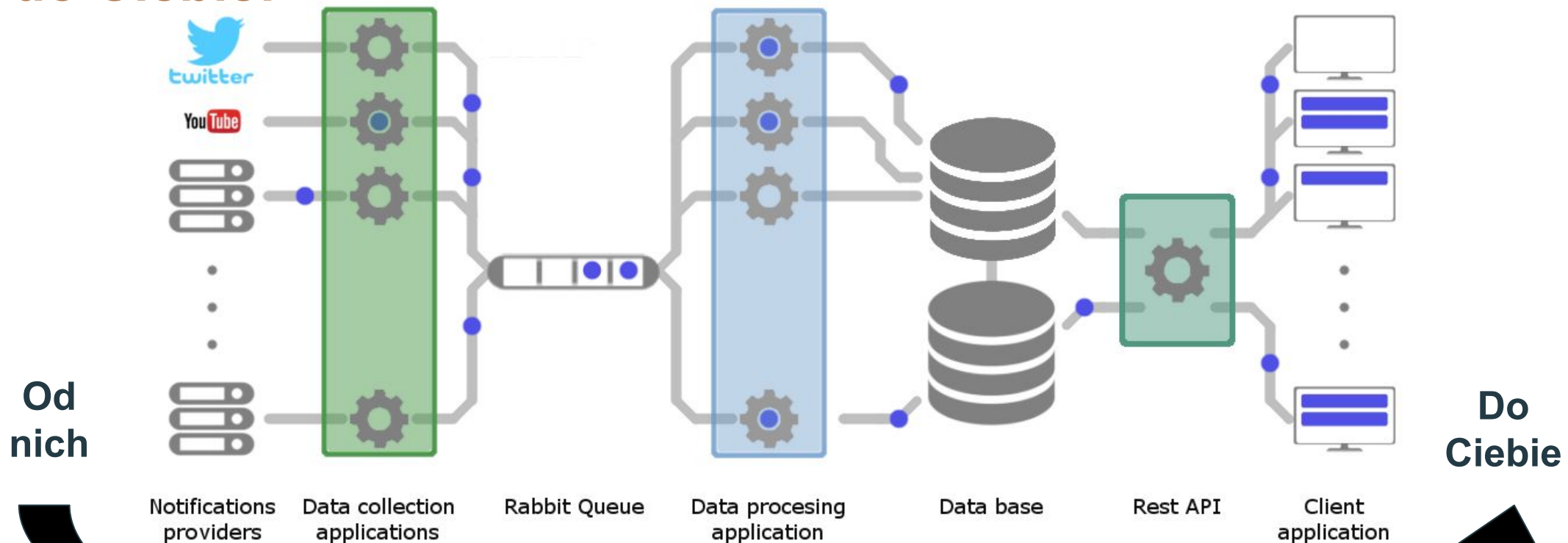
Kto może na tym skorzystać:

- Specjaliści z branży finansowej
- Osoby pracujące z Social Media
- Administratorzy
- Analitycy
- Osoby chcące mieć jedno miejsce do podglądu ze swoich serwisów - krótko mówiąc dla osób ceniących swój czas

Część 2

Krótkie przedstawienie systemu

System gwarantuje, że powiadomienie zawsze dojdzie do Ciebie!

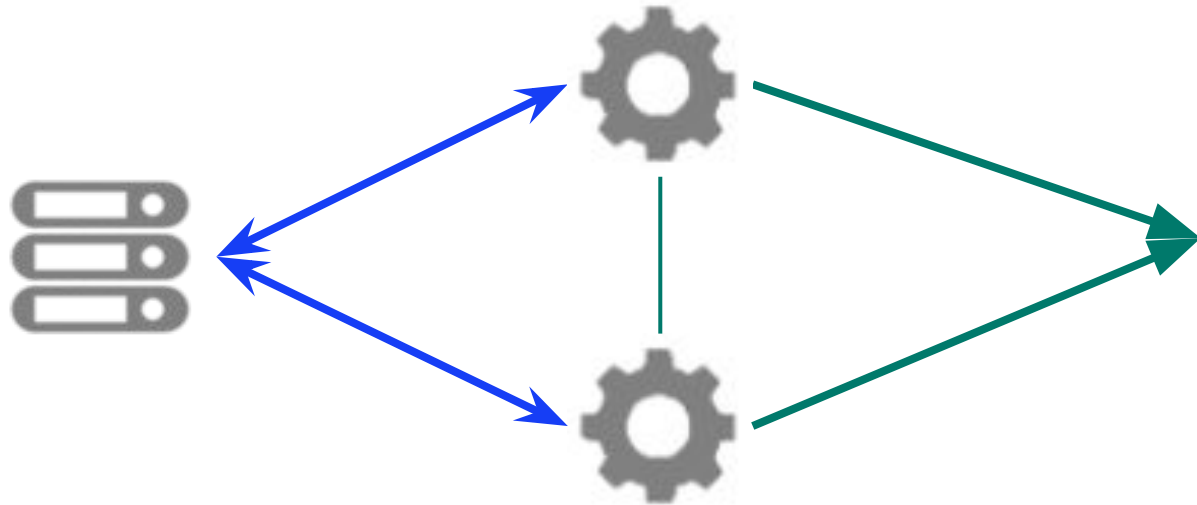


Cross application notification system

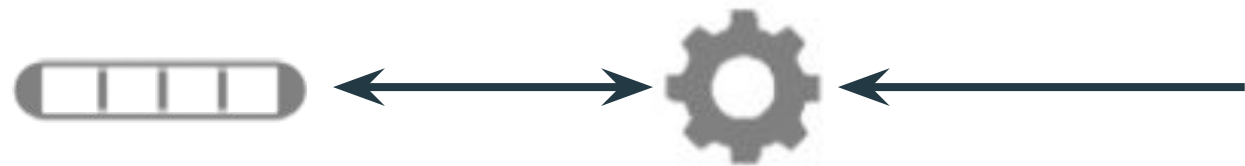
WYGODA | ELASTYCZNOŚĆ | AGREGACJA 7

Krótką wycieczka po systemie cz. 1

Starannie zaprogramowane mikroserwisy zbierające powiadomienia dbają o to, aby każda notyfikacja została przechwycona i przesłana do kolejki.



Mikroserwis łączący kolejkę i bazę danych dokładnie agreguje powiadomienia w taki sposób w jaki zażyczył sobie użytkownik i zapisuje je w bazie danych.



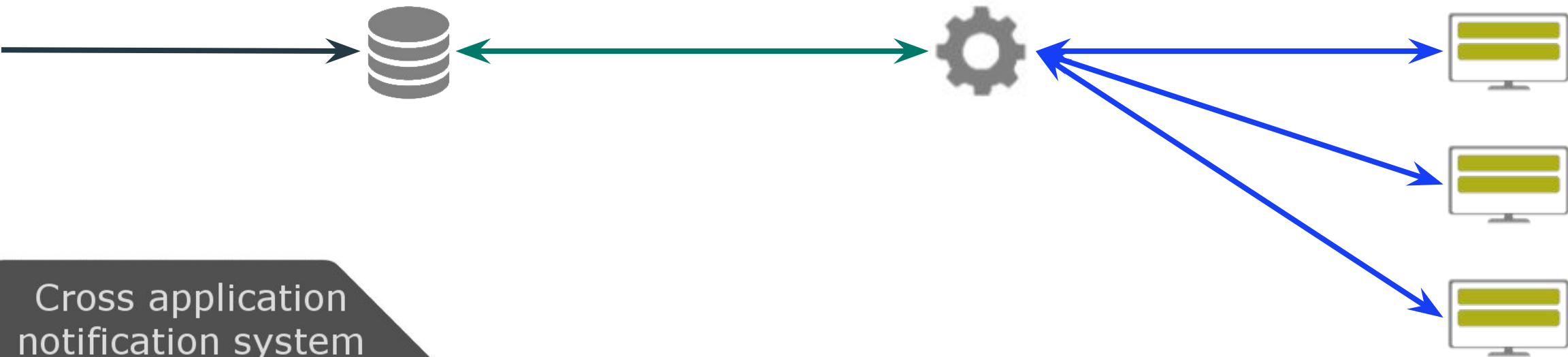
Cross application notification system

WYGODA | ELASTYCZNOŚĆ | AGREGACJA

Krótką wycieczka po systemie cz. 2

REST Api obsługuje połączenie pomiędzy aplikacjami klienckimi a bazą danych, dbając aby do docelowych użytkowników doszły tylko takie informacje jakich oczekują.

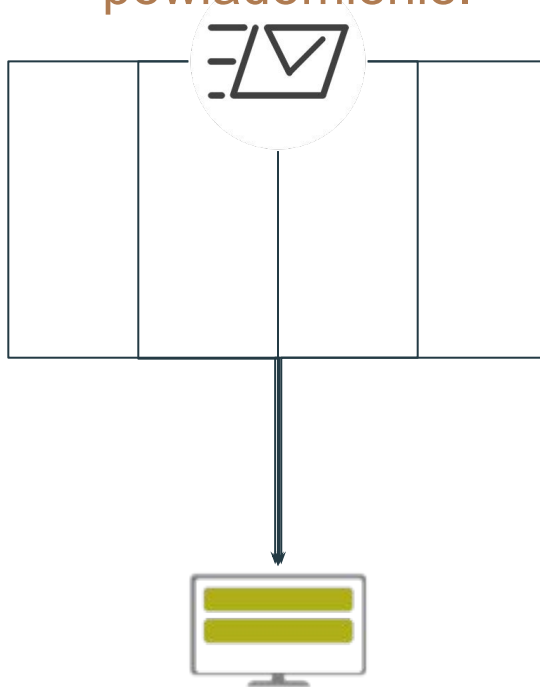
Użytkownicy wygodnie i spokojnie mogą przeglądać informacje, którymi na co dzień są zasypywani.



WYGODA | ELASTYCZNOŚĆ | AGREGACJA

Agregacja

To użytkownik definiuje jak ma trafić do niego powiadomienie.



Nasz system pozwala wybrać użytkownikowi sposób agregacji powiadomień na wiele sposobów:



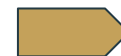
Sposób agregacji:

- zliczanie powiadomień
- dostarczanie ostatniego powiadomienia
- dostarczenie pierwszego powiadomienia



Po czym następuje agregacja:

- po tytule
- po treści
- po własno zdefiniowanym fragmencie

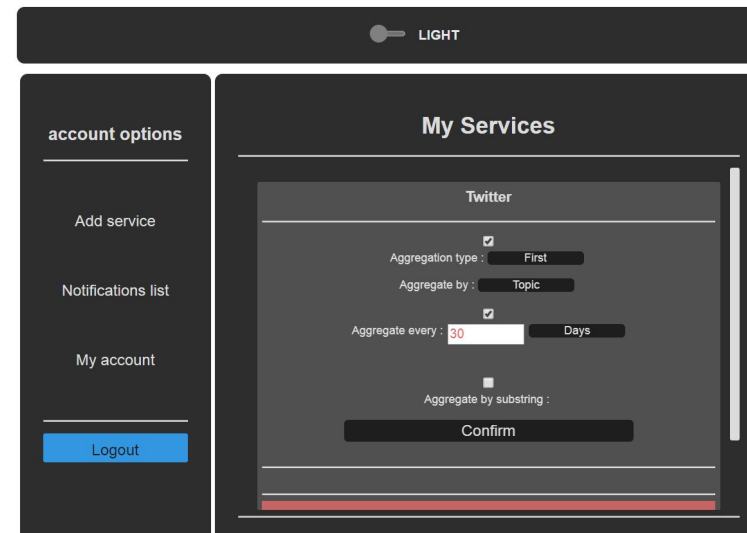
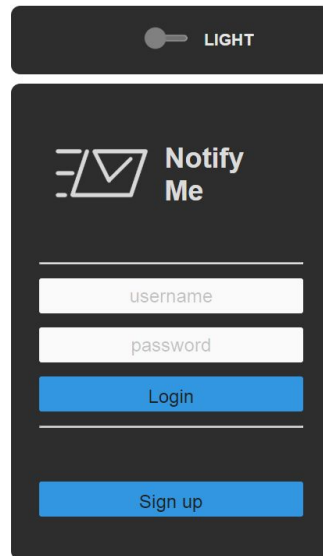
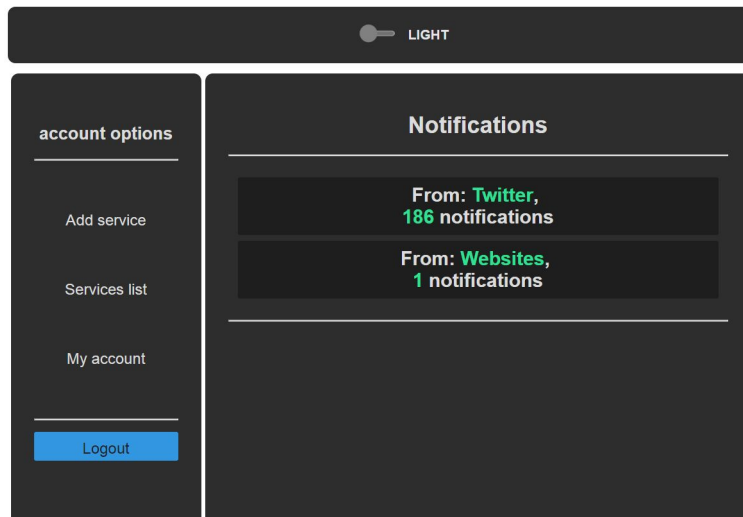


Agregacja po czasie

Część 3

Przedstawienie projektu

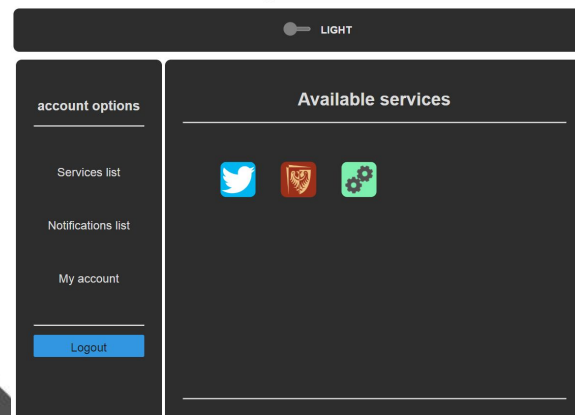
Wiele słów, ale jak to wygląda w praktyce?



Ale jeśli zdjęcia i prezentacja to za mało.....

Sprawdź sam jak to działa!
Link do pobrania:

https://35.204.202.104/ChromePlugin_v1.0.zip



Cross application
notification system

WYGODA | ELASTYCZNOŚĆ | AGREGACJA 12

Standardy sprawdzania kodu

Każdy fragment systemu który wdrażamy musi być na początku sprawdzony pod względem wyszukania błędów oraz zaakceptowany. Wygląda to następująco:

a) gdy kod jest pisany przez osobę odpowiedzialną za ściąganie powiadomień z innych serwisów:

- **sprawdza**: osoba odpowiedzialna za kolejkę i jej obsługę
- **akceptuje**: osoba odpowiedzialna za komunikację z bazą danych

b) gdy kod jest pisany przez osobę odpowiedzialną za kolejkę i jej obsługę:

- **sprawdza**: osoba odpowiedzialna za komunikację z bazą danych
- **akceptuje**: osoba odpowiedzialna za utrzymanie bazy danych

c) gdy kod jest pisany przez osobę odpowiedzialną za komunikację z bazą danych:

- **sprawdza**: osoba odpowiedzialna za utrzymanie bazy danych
- **akceptuje**: osoba odpowiedzialna za aplikację kliencką

d) gdy kod jest pisany przez osobę odpowiedzialną za utrzymanie bazy danych:

- **sprawdza**: osoba odpowiedzialna za aplikację kliencką
- **akceptuje**: osoba odpowiedzialna za konteneryzację na serwerze

e) gdy kod jest pisany przez osobę odpowiedzialną za aplikację kliencką:

- **sprawdza**: osoba odpowiedzialna za konteneryzację na serwerze
- **akceptuje**: osoba odpowiedzialna za ściąganie powiadomień

f) gdy kod jest pisany przez osobę odpowiedzialną za agregację na serwerze:

- **sprawdza**: osoba odpowiedzialna za ściąganie powiadomień
- **akceptuje**: osoba odpowiedzialna za kolejkę i jej obsługę

g) gdy kod jest pisany przez parę osób naraz:

- **sprawdza**: osoba oddelegowana przez lidera projektu
- **akceptuje**: lider projektu

Rzeczy na które nie starczyło czasu

Niestety nie wszystkie zaplanowane rzeczy udało się nam zrealizować na dzień dzisiejszy.

Części które chcemy dopracować lub stworzyć do Konferencji Projektów Zespołowych to:



Możliwość podłączenia własnego mikro serwisu który generuje powiadomienia.



Podłączenie do systemu konta mailowego, które będzie służyło za komunikację z użytkownikami.



Odwzorowanie funkcjonalności plugina w aplikacji na Androida.



Zaimplementowanie trzeciego wspieranego przez nas źródła powiadomień.

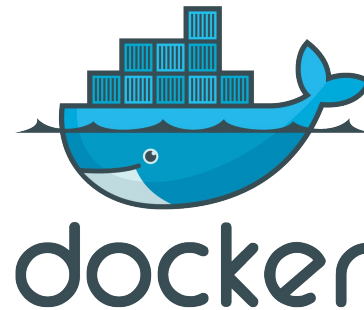


Powiadomienia web push.

Użyte technologie:



Google Cloud Platform



DATADOG



Cross application notification system

WYGODA | ELASTYCZNOŚĆ | AGREGACJA

Prezentacja systemu

Projekt realizowany był przez studentów Politechniki Wrocławskiej, Wydziału Elektroniki, Kierunku Informatyka:

 **Filip Baszak**

 **Mateusz Wójcik**

 **Arek Sokołowski**

 **Bartosz Gardziejewski**

 **Adam Maciak**

Z pomocą opiekunów z firmy NOKIA:

 **Dominik Markiewicz**

 **Mateusz Sołtysik**

Cross application
notification system



Dziękujemy za uwagę

Filip Baszak, Mateusz Wójcik, Adam Maciak, Bartosz Gardziejewski, Arkadiusz Sokołowski
Wroclaw University of Technology, Department of Electronics, IT Faculty



Czas na pytania

Baza danych



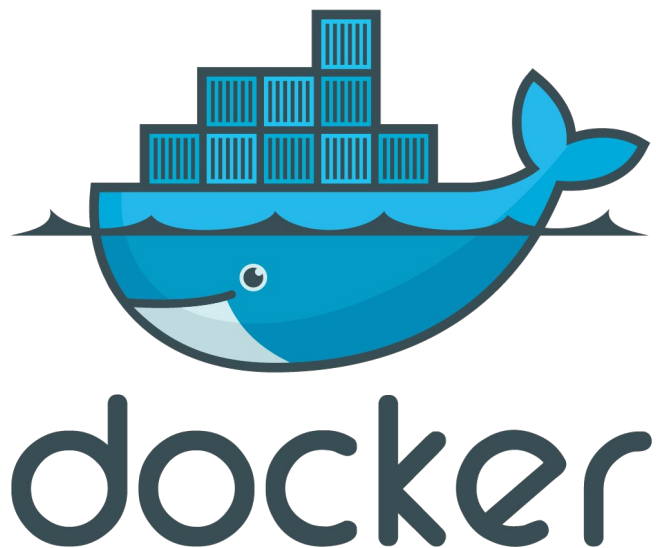
Baza danych - Widoki

View	Table	Column
User_subscriptions	User	Login
	Subscription	Name
Notification_name	Subscription	Name
	Notification	Topic
	Notification	Message
	Notification	Time
	Notification	Flag

Cross application
notification system

WYGODA | ELASTYCZNOŚĆ | AGREGACJA

Dlaczego ten docker?



Do wdrożenia i uruchomienia naszej aplikacji rozproszonej wybraliśmy Dockera. Pozwoli on umieścić nasz program oraz jego zależności w lekkich, przenośnych, wirtualnych kontenerach które można uruchomić na prawie każdym serwerze z systemem Linux.

Cross application
notification system

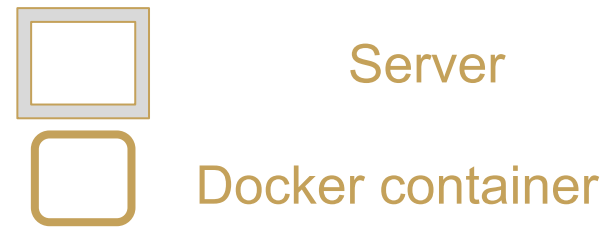
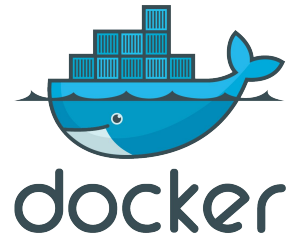
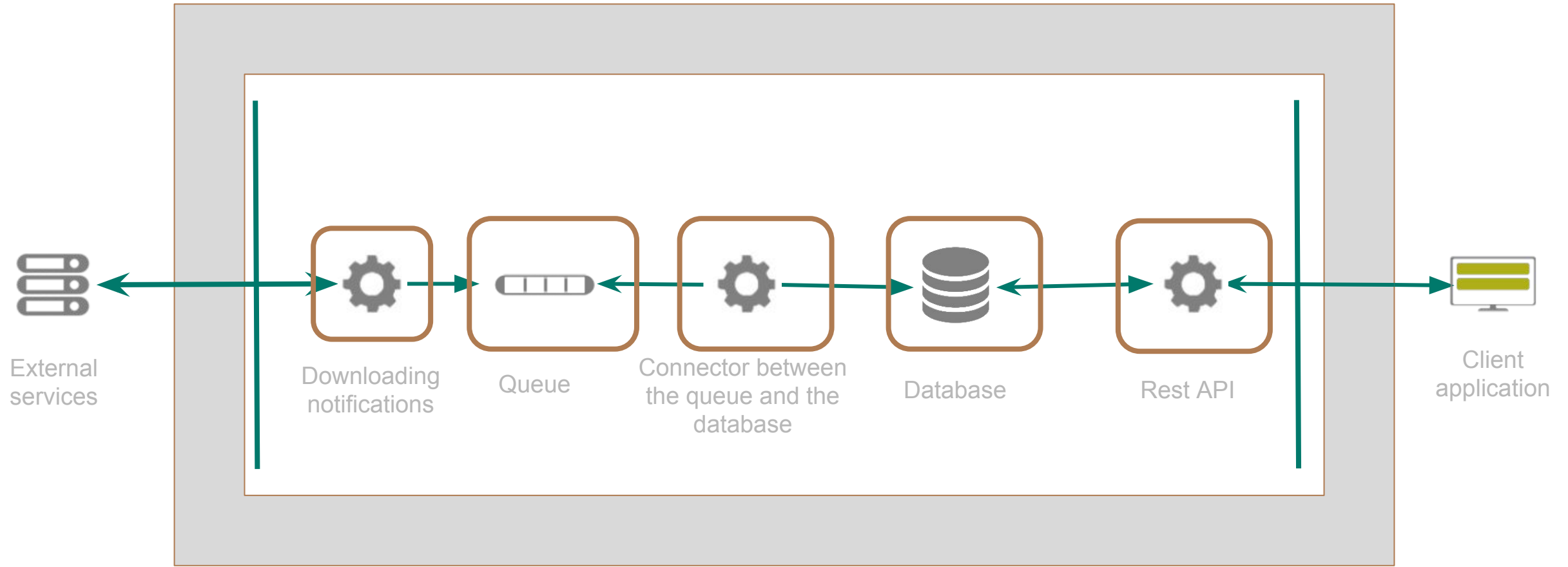
WYGODA | ELASTYCZNOŚĆ | AGREGACJA

Format notyfikacji w systemie

Notification JSON format:

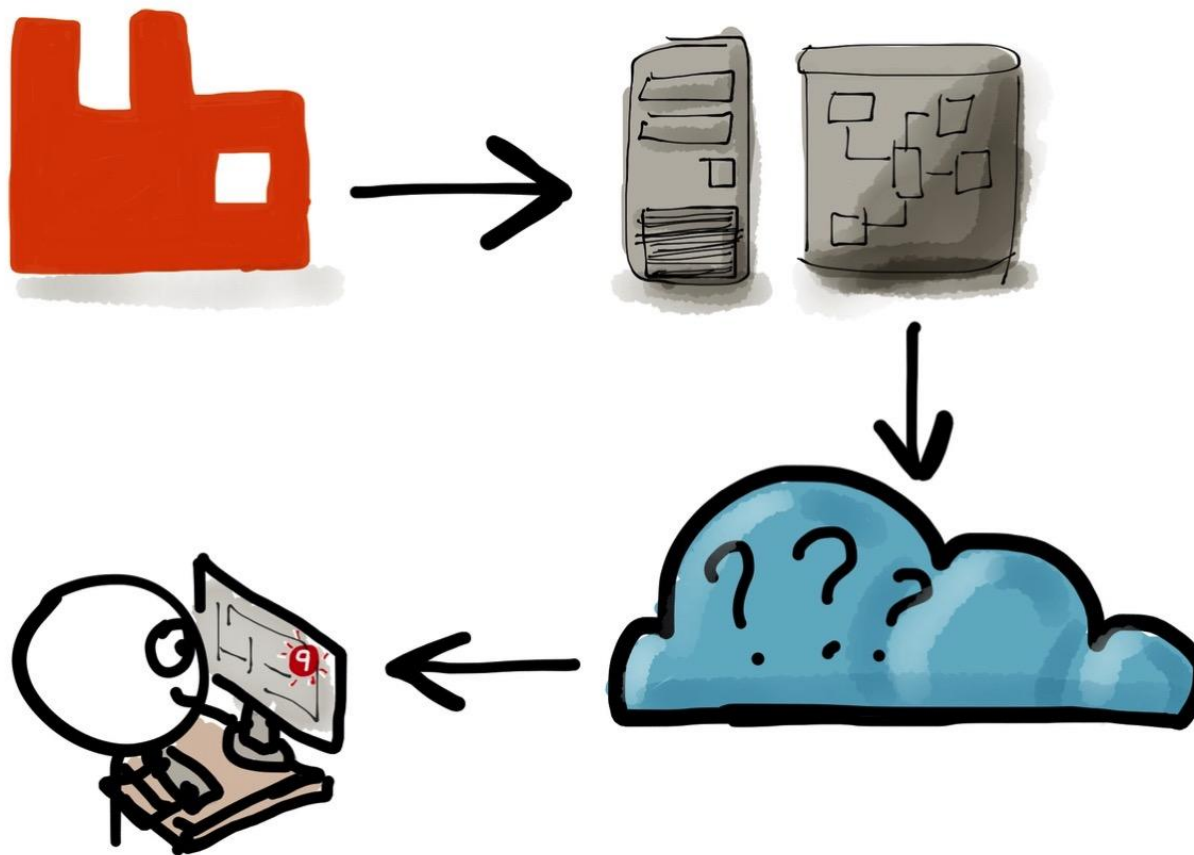
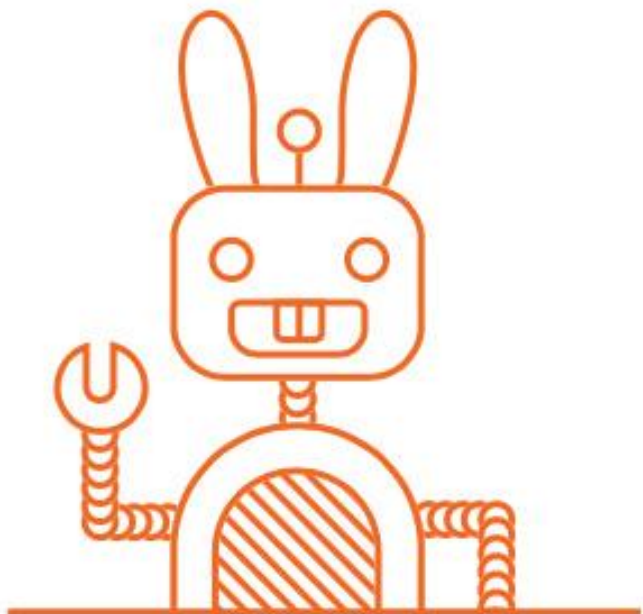
```
{  
  notificationID : ,  
  userID : ,  
  sourceID : ,  
  flag : ,  
  topic : ,  
  message : ,  
  timestamp : ,  
  priority :  
}
```

Struktura systemu



Dlaczego RabbitMQ?

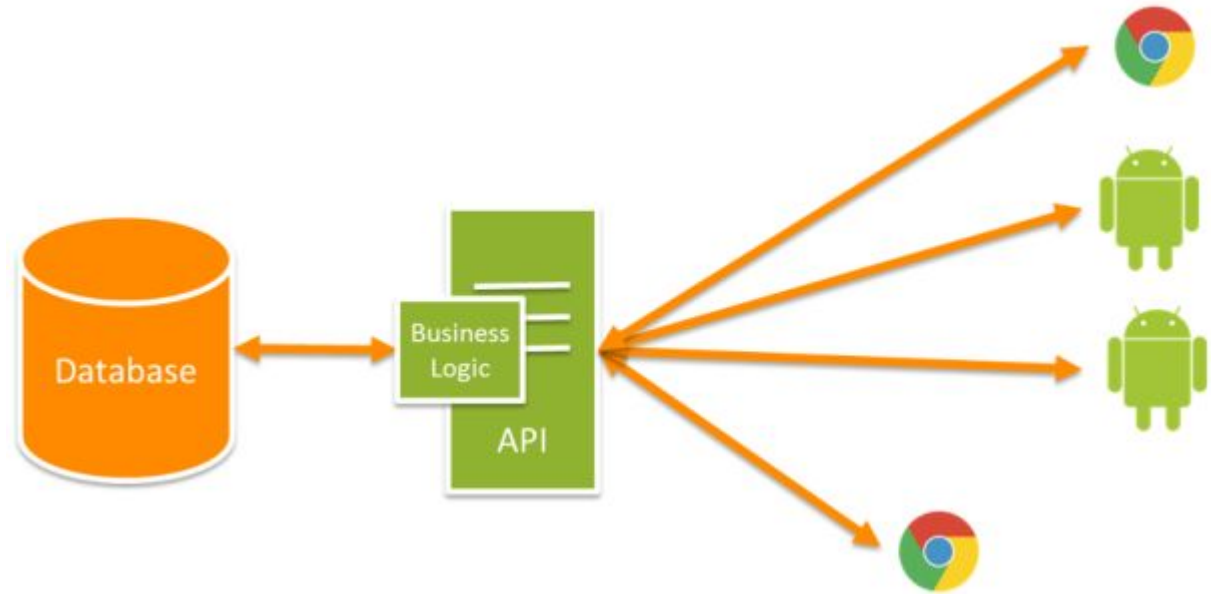
 RabbitMQ by Pivotal.



Cross application
notification system

WYGODA | ELASTYCZNOŚĆ | AGREGACJA

Rest API



Cross application
notification system

WYGODA | ELASTYCZNOŚĆ | AGREGACJA