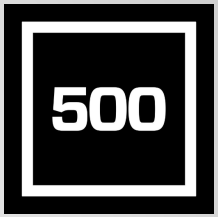
Sprytne projektowanie aplikacji rozproszonych

@ArturZylinski (azim)

@ArturZylinski



- Tech Leader w **Growbots** - A.I. for Sales
  - 500 Startups
  - 50+ dockerowych instancji
  - #python #docker #elasticsearch #ansible #rabbitmq #reactjs
- uwielbiam algorytmikę i ...
- … zwiedzać świat, grać w piłkę

O czym opowiem?

1. Projektowanie *jakich* aplikacji?
2. Czy warto? Trochę teorii
3. Krok po kroku
    a. Komunikacja
    b. Kompozycja
4. Demo

Service-Oriented Architecture (SOA)

Def.: Aplikacje są NIE - ZA - LEŻ - NE

Czy warto?

Lowering maintenance costs

KISS principle

Scalability

Parallel Development

Easier testing

Czy warto?

Loose coupling

Single responsibility principle

More Agile

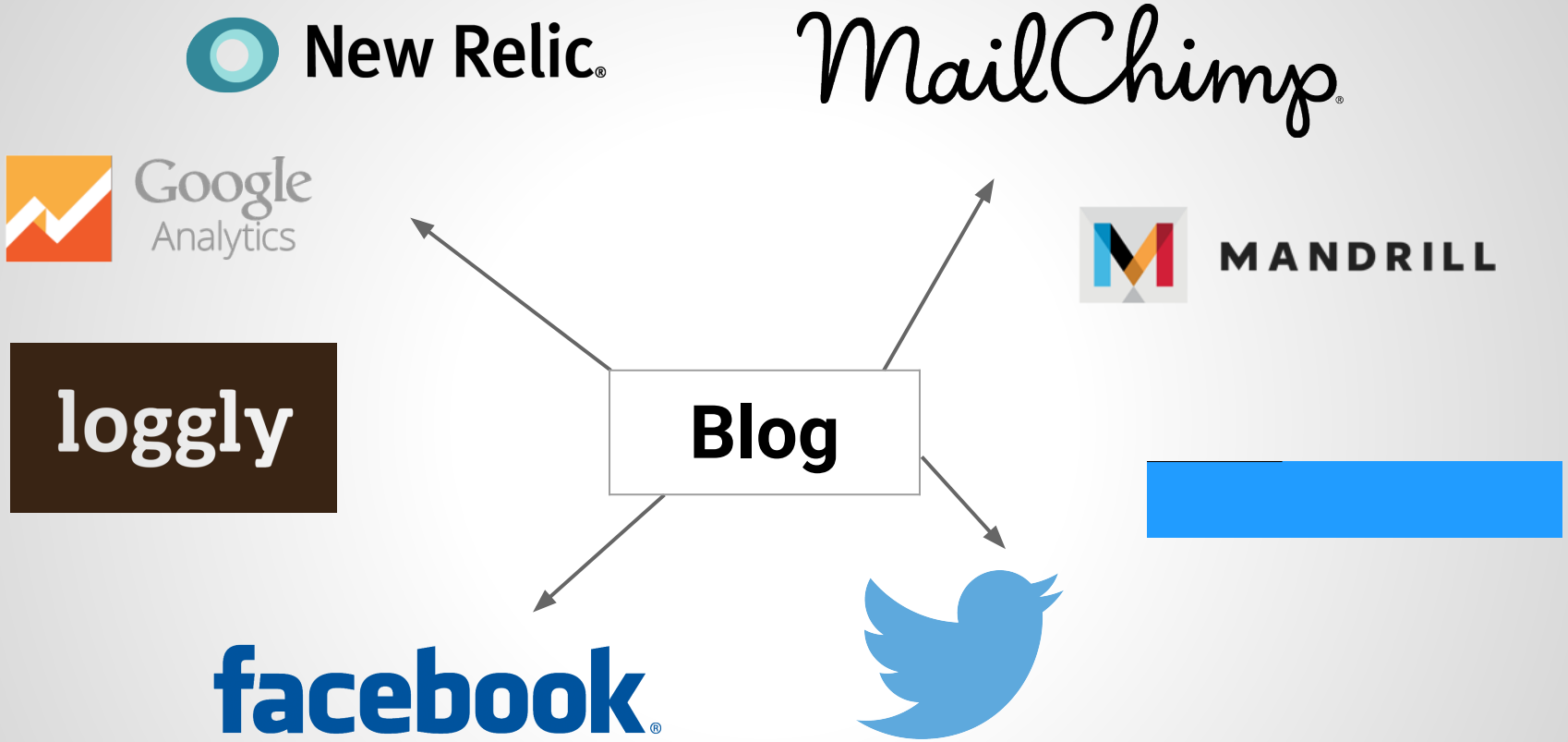
Enhancing architectural flexibility

Krok #0: Jak zacząć?

Tak naprawdę już
(nieświadomie?) zaczęliśmy

Blog

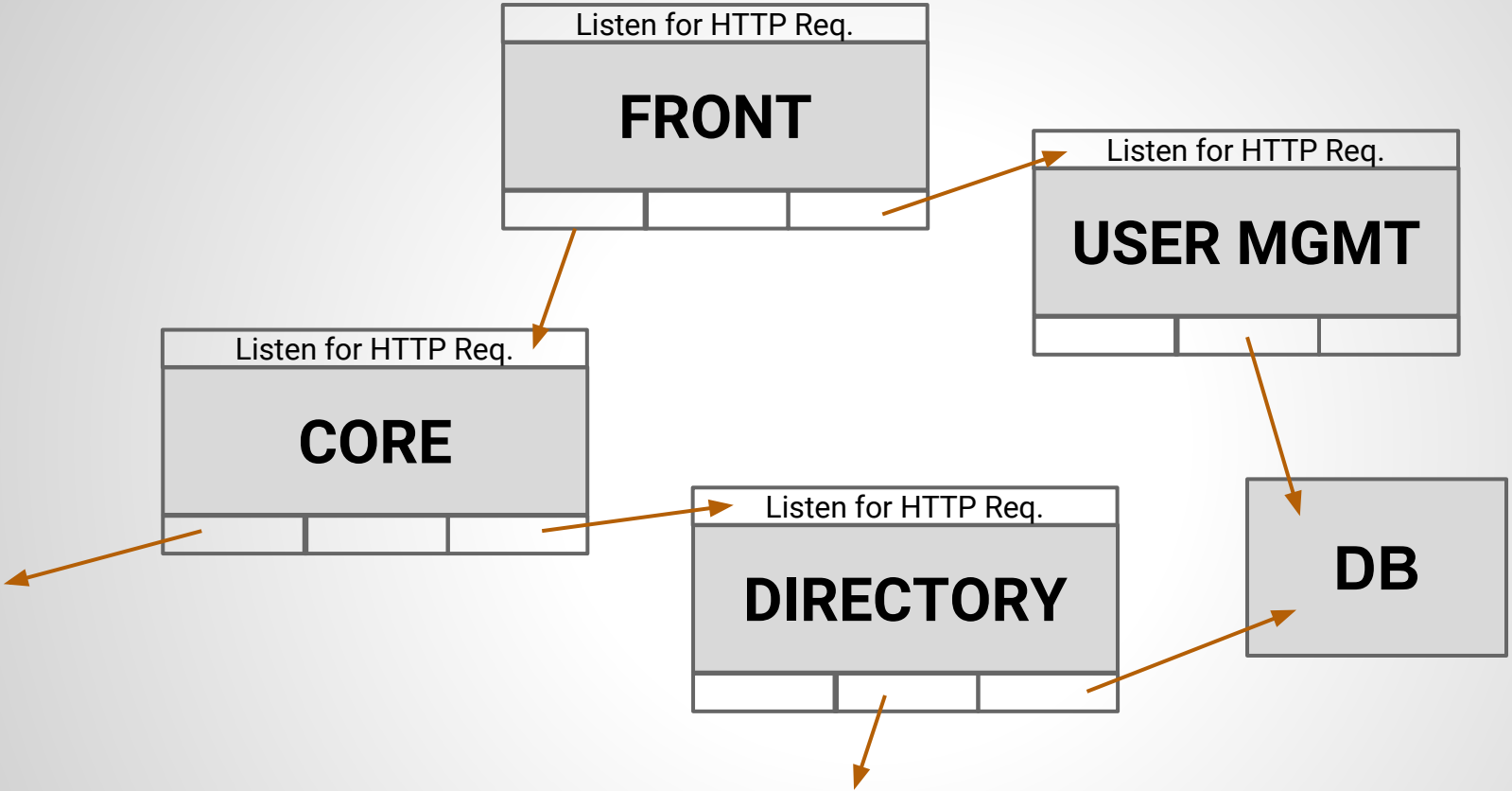
Nawet najprostszy blog...



Nawet najprostszy blog będzie wymagał integracji z zewnętrznymi serwisami

Krok #1: Komunikacja

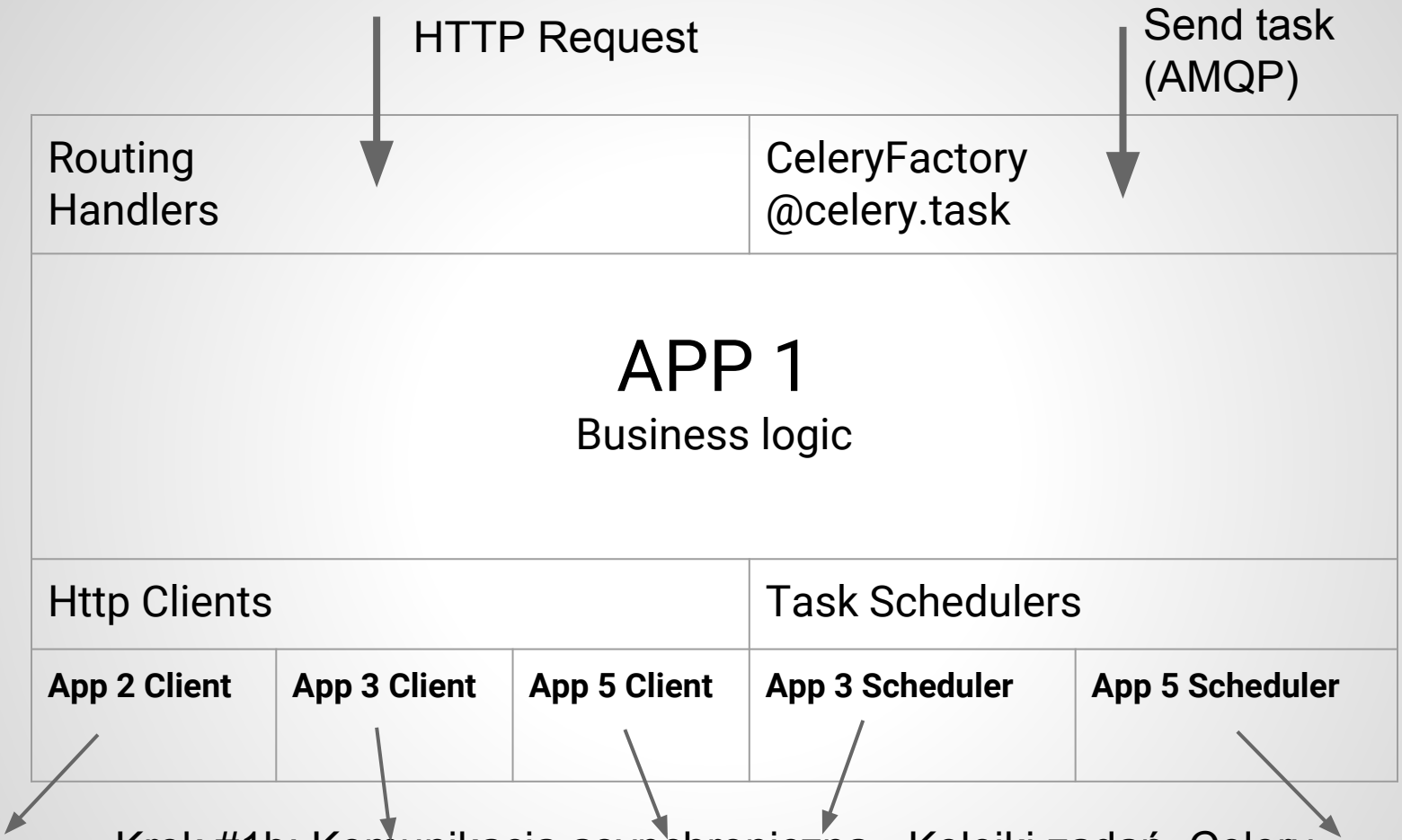
HTTP, API Clients, RESTful



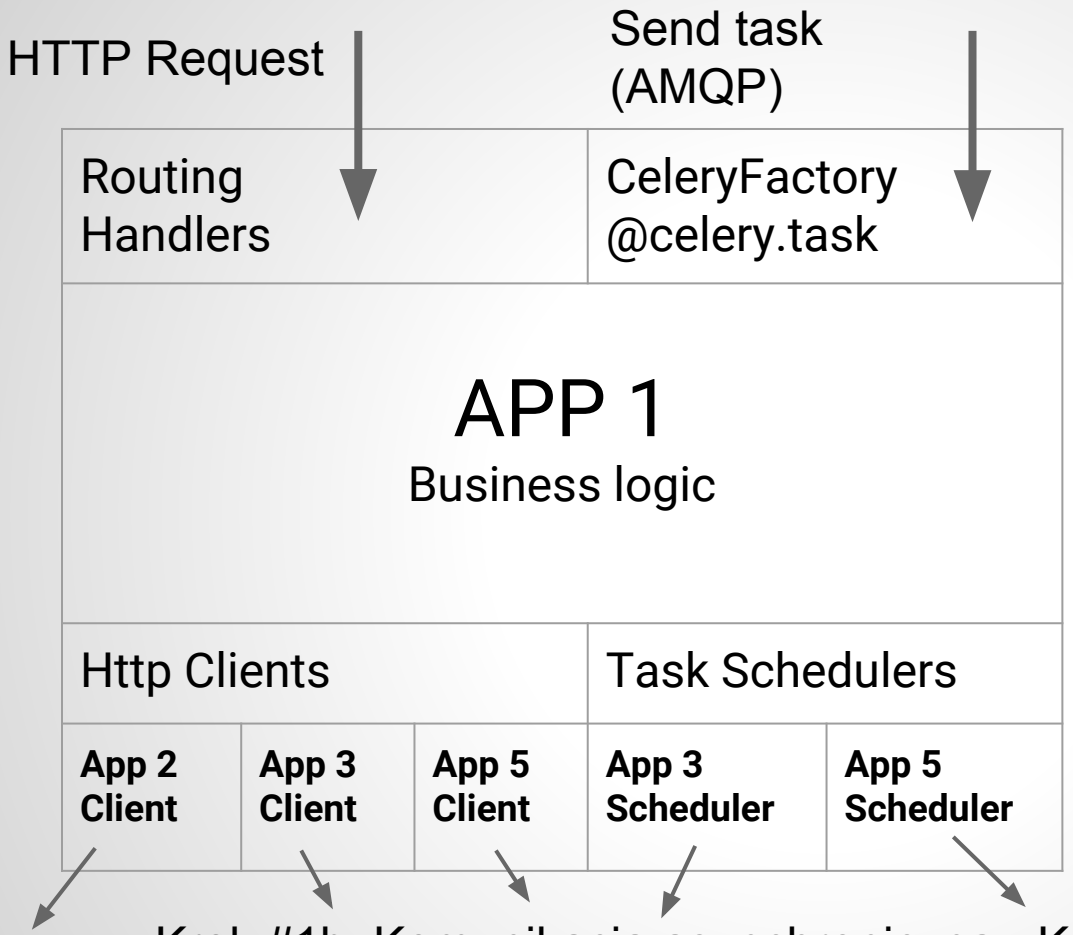
Krok #1: Komunikacja - HTTP, API Clients, RESTful

Krok #1b: Komunikacja asynchroniczna

Kolejki zadań, Celery (RabbitMQ, Redis), Apache Kafka***



Krok #1b: Komunikacja asynchroniczna - Kolejki zadań, Celery



Endpoints

Czarna skrzynka

Clients / Schedulers

(wspólne dla wszystkich aplikacji)

Krok #1b: Komunikacja asynchroniczna - Kolejki zadań, Celery

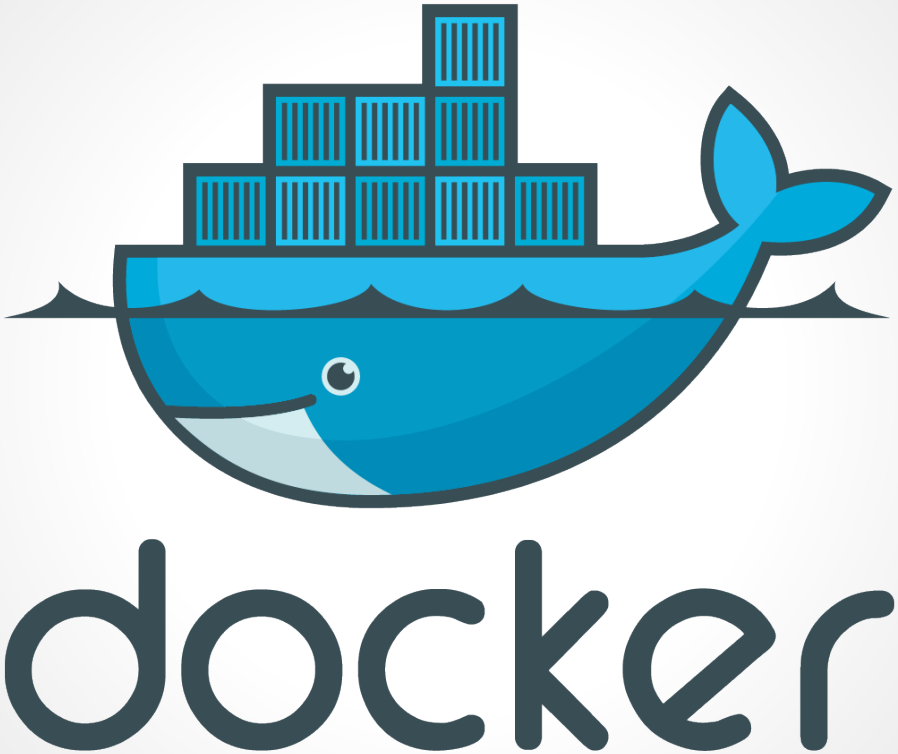
Tips & Tricks

Jak współdzielić kod między serwisami?

- paczki Pythona
- osobne repozytorium (*shared/common*)
- wersjonowanie
- łatwe i szybkie rozwiązanie?
  - dev. - *symbolic link*
  - stg./prod. - osobna kopia

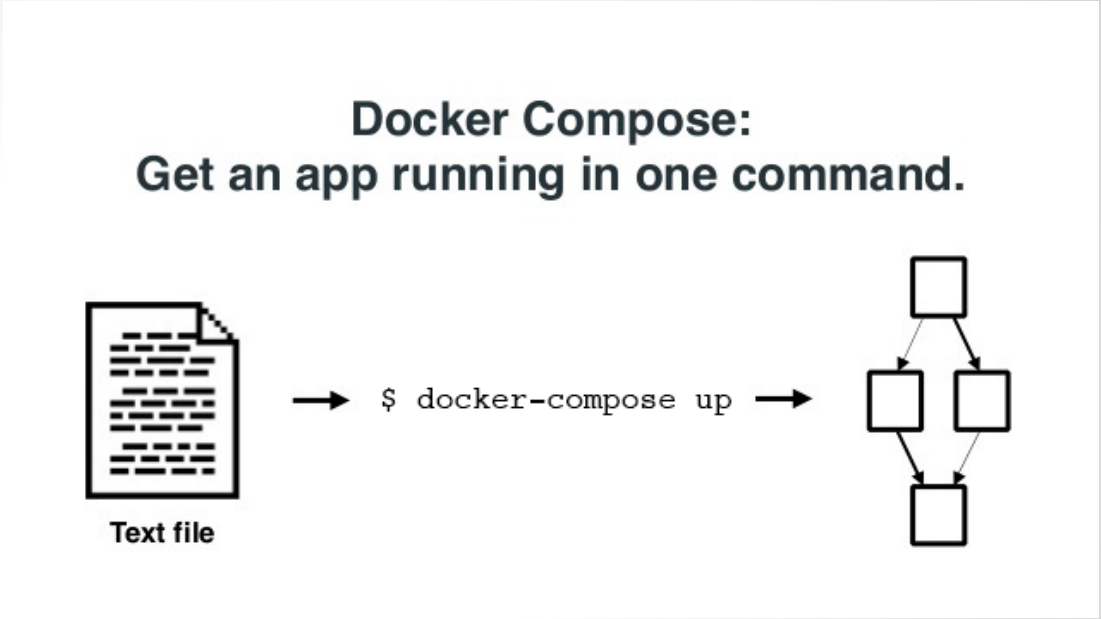
Krok #2: Kompozycja

Kontenerowa rewolucja


docker

docker-compose.yml

```
web:
    build: .
    links:
        - db
    ports:
        - "8000:8000"
db:
    image: postgres
```



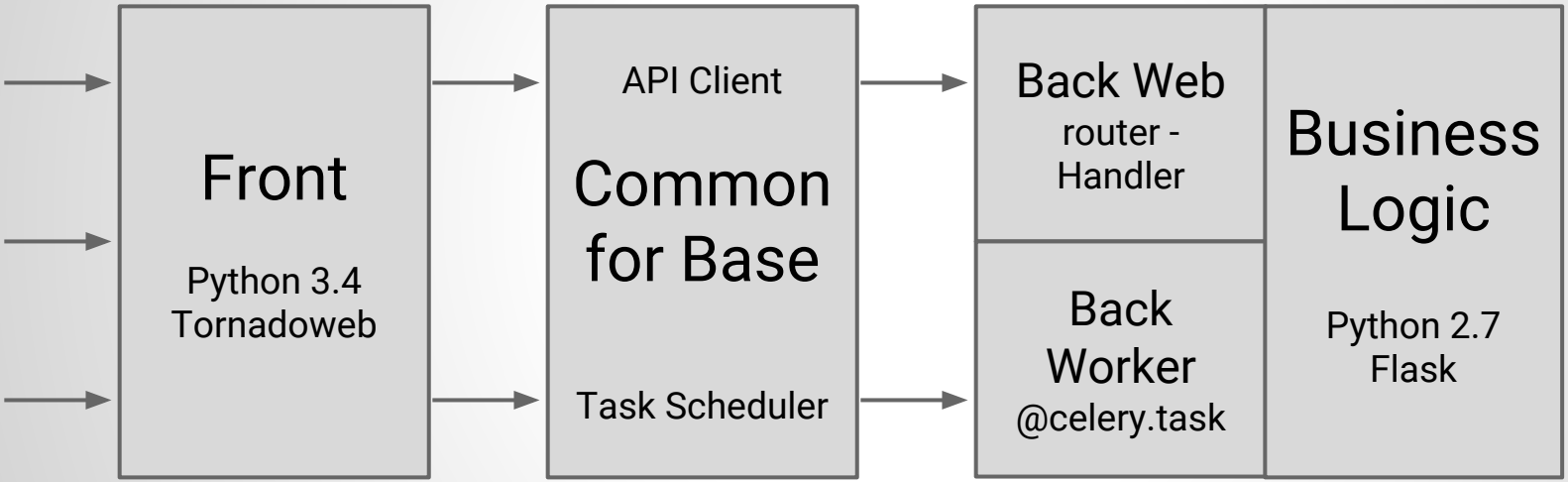
blog.docker.com

Krok #2: Kompozycja - Kontenerowa rewolucja

DEMO time

https://github.com/azylinski/pywaw48-counter-soa-demo

Co widzieliśmy?



+ RabbitMQ

+ Redis

Hmm… żadnych problemów? Naprawdę???

Potencjalne trudności

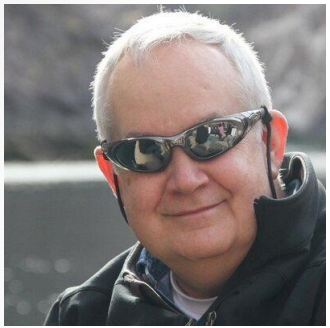
- autoryzacja
- monitorowanie, system logów
- bardziej złożona architektura
- ...

Czy NAPRAWDĘ warto?

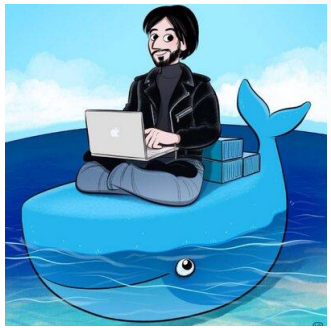
Nie wierzcie na słowo

Nie wierzcie na słowo

Adrian Cockcroft
@adrianco



Fred George
@fgeorge52



Solomon Hykes
@solomonstre

Martin Fowler
@martinfowler

“Mówią jak jest”

Nie wierzcie na słowo

- The State of the Art in Microservices by Adrian Cockcroft
- Implementing Service Oriented Architecture by Amazon Web Services
- Micro-Service Architecture by Fred George
- Keynote: Microservices by Martin Fowler

- Docker Channel
- Docker London Meetup, January 2015:
  Andrew Martin – Building and Testing Docker Containers
- DockerCon EU: Keynote on Orchestration
  (Docker Machine, Swarm, Compose)

Co dalej?

- Micro-services
- OS: CoreOS
- Service discovery: etcd
- Messaging system: Apache Kafka
- Docker - Machine, Swarm and Compose
- I dużo, dużo więcej ...

Pięknie dziękuję

Pytania?

@ArturZylinski
artur@growbots.com
www.growbots.com