

SIMPLE WEB APPLICATION ARCHITECTURE

MACIEJ ŁOTYSZ

MACIEJ.LOTYSZ@BLSTREAM.COM

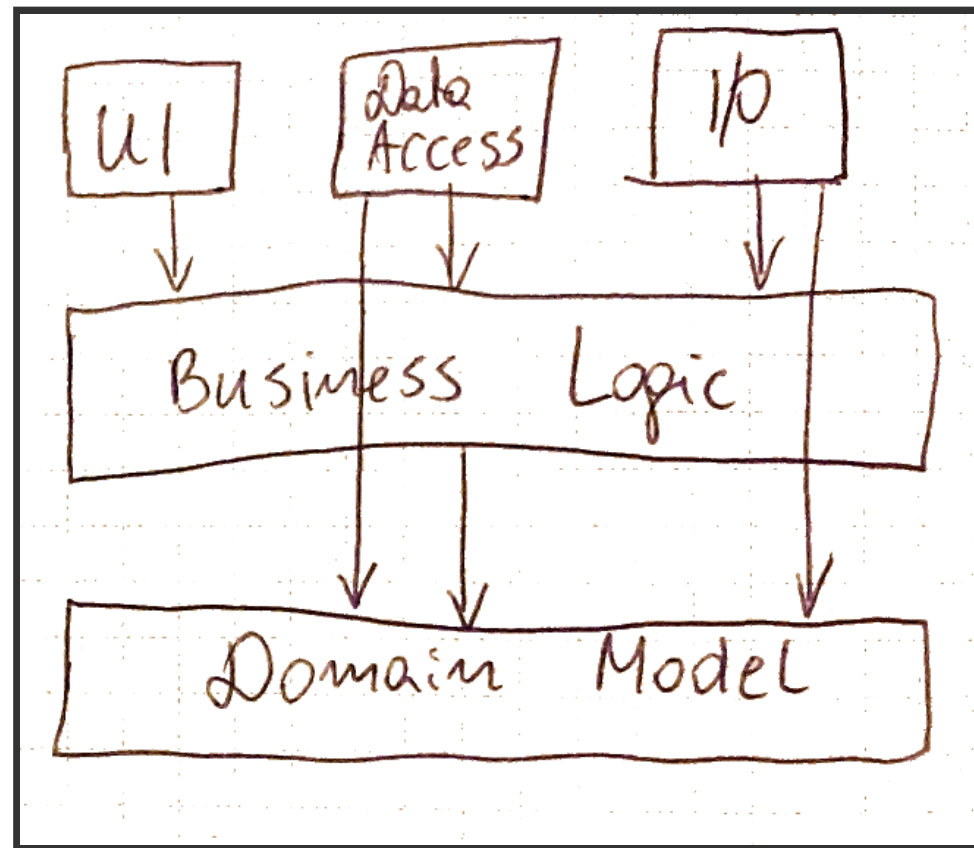
POCZĄTEK

- rozwijany przez 7 lat system sprzedażowy, SLOC = **250k**
- struts + jsp + spring + hibernate + jboss 4
- czas kompilacji: 1m 30s + deploy/restart (30s)

CEL ZMIANY

- nowy projekt UX -> zwiększenie ergonomii systemu
- testowalny, skalowalny, niezawodny
- czas realizacji: 3 miesiące...

ZAŁOŻENIA DLA NOWEJ WERSJI (IDEAŁ)



Autor: Andrzej Bednarz

ARCHITEKTURA

- oddzielenie warstw: widoku (frontend) od logiki (backend)
- backend: modularny, bezstanowy, testowalny
- łatwy w utrzymaniu: monitorowanie wydajności, jakości i stanu
- maksymalna prostota!

KOMPONENTY FRONTENDU

BIBLIOTEKI



- angularjs M-odel C-ontroller V-iew
- restangular - REST API + promises
- lodash - map, reduce, filter, ...

UI: TWITTER BOOTSTRAP 2.X

Example

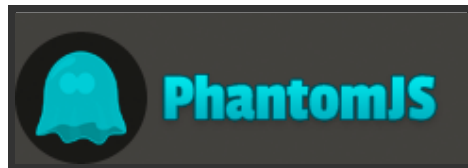
Action ▾ Action ▾ Danger ▾ Warning ▾ Success ▾ Info ▾ Inverse ▾

```
1. <div class="btn-group">
2.   <a href="#" data-toggle="dropdown">Action ▾</a>
3.   <a href="#" data-toggle="dropdown">Another action</a>
4.   <a href="#" data-toggle="dropdown">Something else here</a>
5.   <span data-toggle="dropdown" class="dropdown-toggle">
6.     <ul class="dropdown-menu">
7.       <!-- dropdown menu links -->
8.     </ul>
9. </div>
```

Action
Another action
Something else here
Separated link

TESTING

- karma - test runner
- jasmine - BDD for js



- testy js bez przeglądarki

BACKEND COMPONENTS

- **dropwizard-core** - lekki framework ułatwiający tworzenie RESTowego API
- **dropwizard-guice** - dependency injection
- **dropwizard-jaxws** - wsparcie dla usług JAX-WS
- **jasper reports** - wydruki: PDF/RTF

JETTY - KONTENER SERWLETÓW

używany przez: Apache ActiveMQ, Alfresco, Apache Geronimo, Apache Maven, Apache Spark, **Google App Engine**, **Eclipse**, FUSE, Twitter's Streaming API and Zimbra, Lift, Eucalyptus, Red5, Hadoop

JERSEY

- referencyjna implementacja JAX-RS 1.1/2.0
- Reactive Jersey...nadchodzi

JACKSON

- mapowanie encji z/do JSON
- mapowanie konfiguracji YAML na model w java

RDBMS

- **jdbi** - "tak powinno wyglądać JDBC API"
- **hibernate** - ORM

METRYKI

- metrics (autor: Coda Hale) to biblioteka pozwalająca mierzyć stan krytycznych komponentów systemu w czasie rzeczywistym
- Jetty, Logback, Log4j, Apache HttpClient, Ehcache, JDBC, Jersey

POZOSTAŁE

- **liquibase** - zarządzanie zmianami w RDBMS
- **hibernate validator** - API dla walidacji
- **Apache HTTP client** - wygodna konfiguracja, pooling
- **Jersey client** - klient REST, pooling
- **freemaker/*mustache*/jsp** - szablony
- **joda time** - wygodne API do czasu
- **logback** - wydajne logowanie
- **guava** - biblioteki użytkowe dla java używane w projektach google

EFEKTY

- po 3 miesiącach pierwszy proces biznesowy
- po 1,5 roku rozwoju nie mamy dość wybranych technologii
- licznik wdrożeń: 220
- klient zadowolony

PROGRAMISTA

- SLOC = 70k
- czas kompilacji:
 - frontend: **13 sekund**
 - backend: **6 sekund**
- czas uruchomienia backendu: **30 sekund** (brak deploy)
- nie ma przeszkód technicznych dla używania TDD/BDD

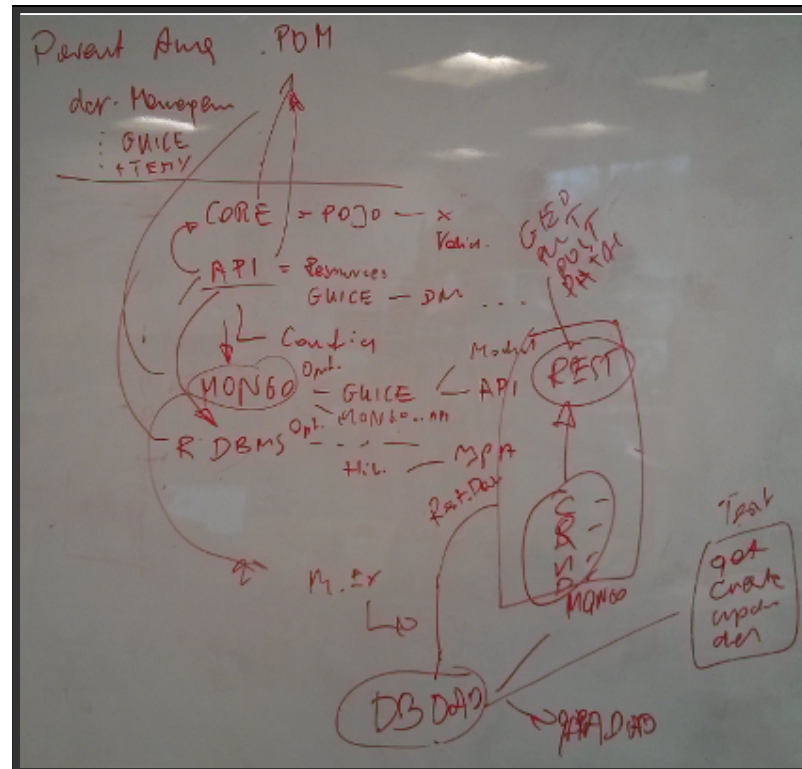
WYGODA DLA SUPPORTU

- jedna instancja backendu = jeden proces w Linuxie
- konfiguracja w YAML
- metryki dostępne w czasie rzeczywistym
- jeden plik logów, wygodny do parsowania format
- /healthcheck (stan systemu: OK/ERROR)
- scheduler: cron + curl <http://server/task/doSomeBatchJob>

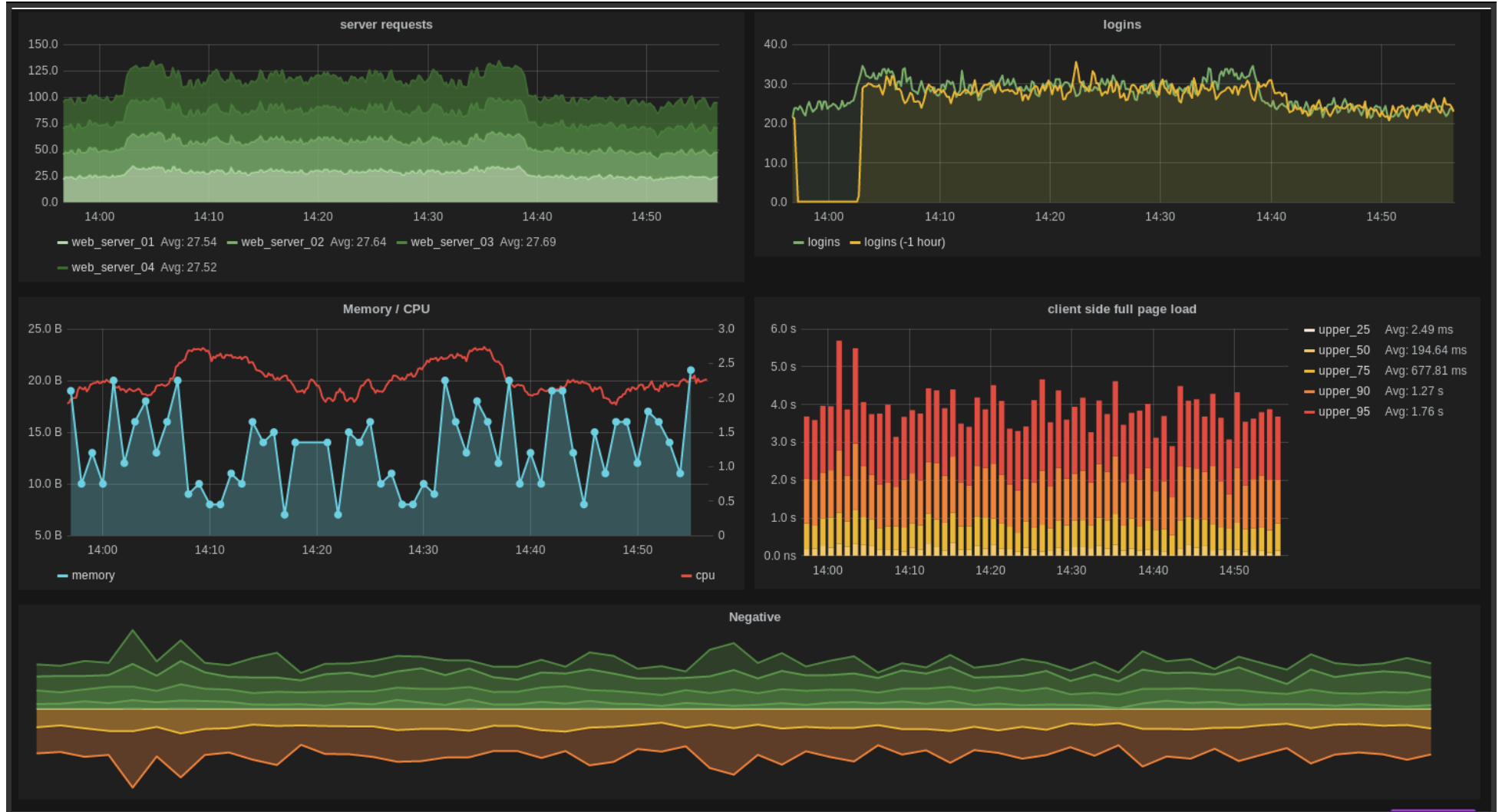
PLANY

- wdrożenie TDD/BDD
- reactive programming?
- zwiększenie wydajności: cache po stronie klienta
- java 1.8 + dropwizard 0.8

REALIZACJA W PROGRAMIE PATRONAŻOWYM



METRYKI (GRAPHITE + GRAFANA)



LOGI I INCYDENTY (KIBANA)



DZIĘKUJĘ ZA UWAGĘ

- zapraszamy do patronatu w przyszłym roku
- jesteśmy na <https://www.facebook.com/blstream>