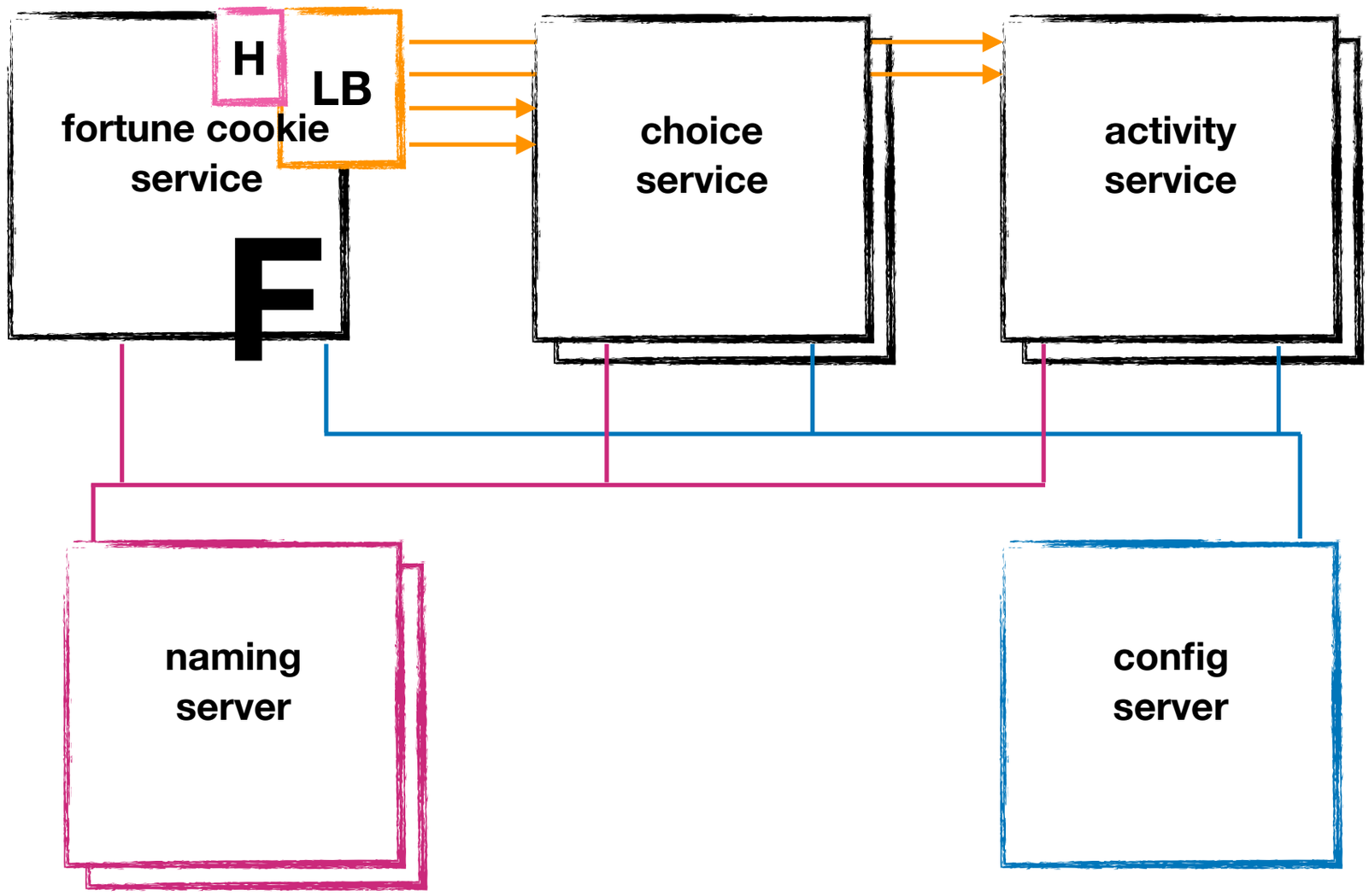
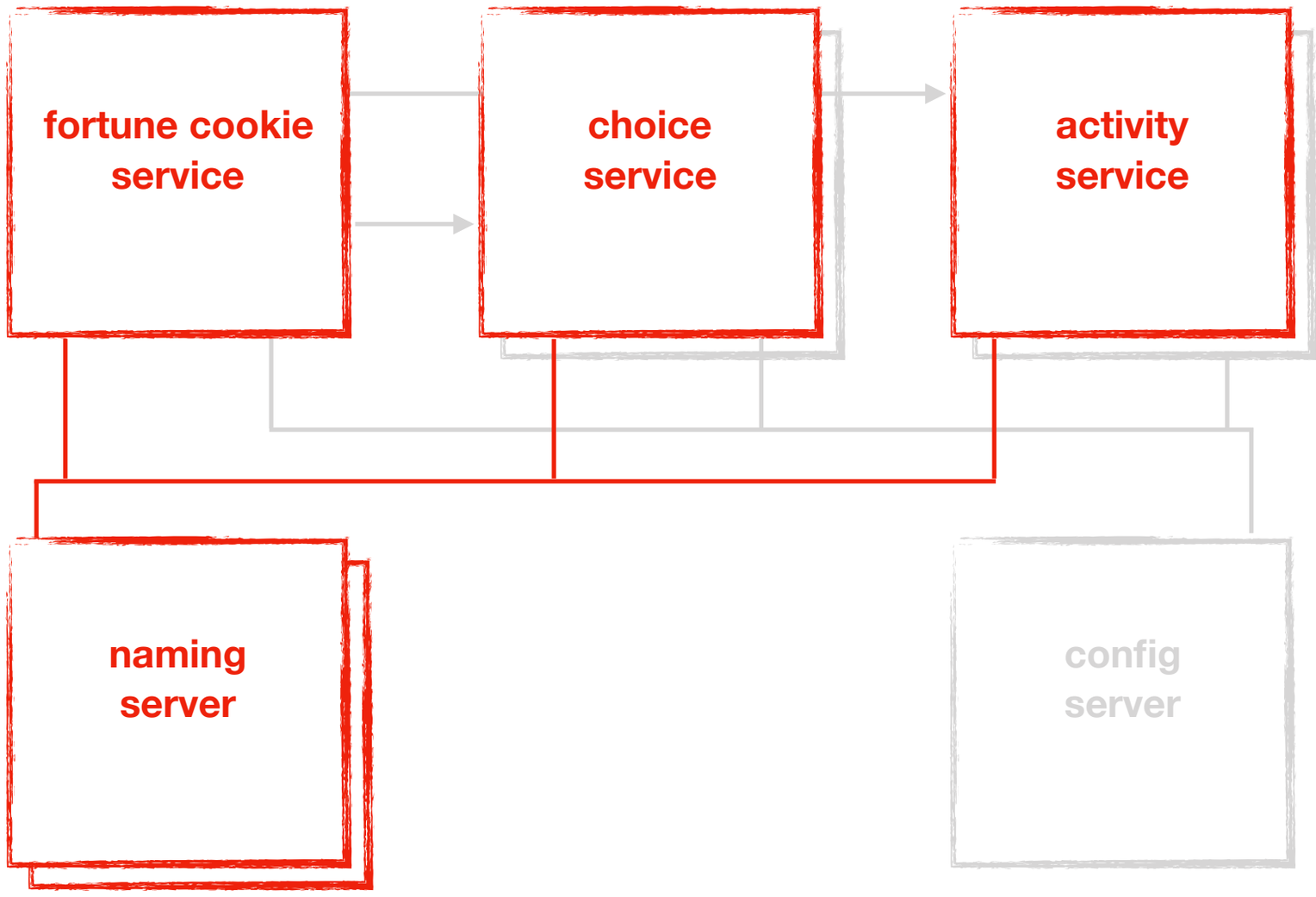
Spring Cloud

Fortune



Spring
Cloud

I Serwisy i Eureka



Spring Cloud

I Serwisy i Eureka

- serwer Eureka
- serwis fortune-cookie-service
  - jako klient Eureki
- zadanie - utworzenie serwisów, klientów Eureki
  - decision-service
  - activity-service
- refaktor fortune-cookie-service
  - korzysta z decision-service i activity-service
- refaktor Eureka - 2 instancje

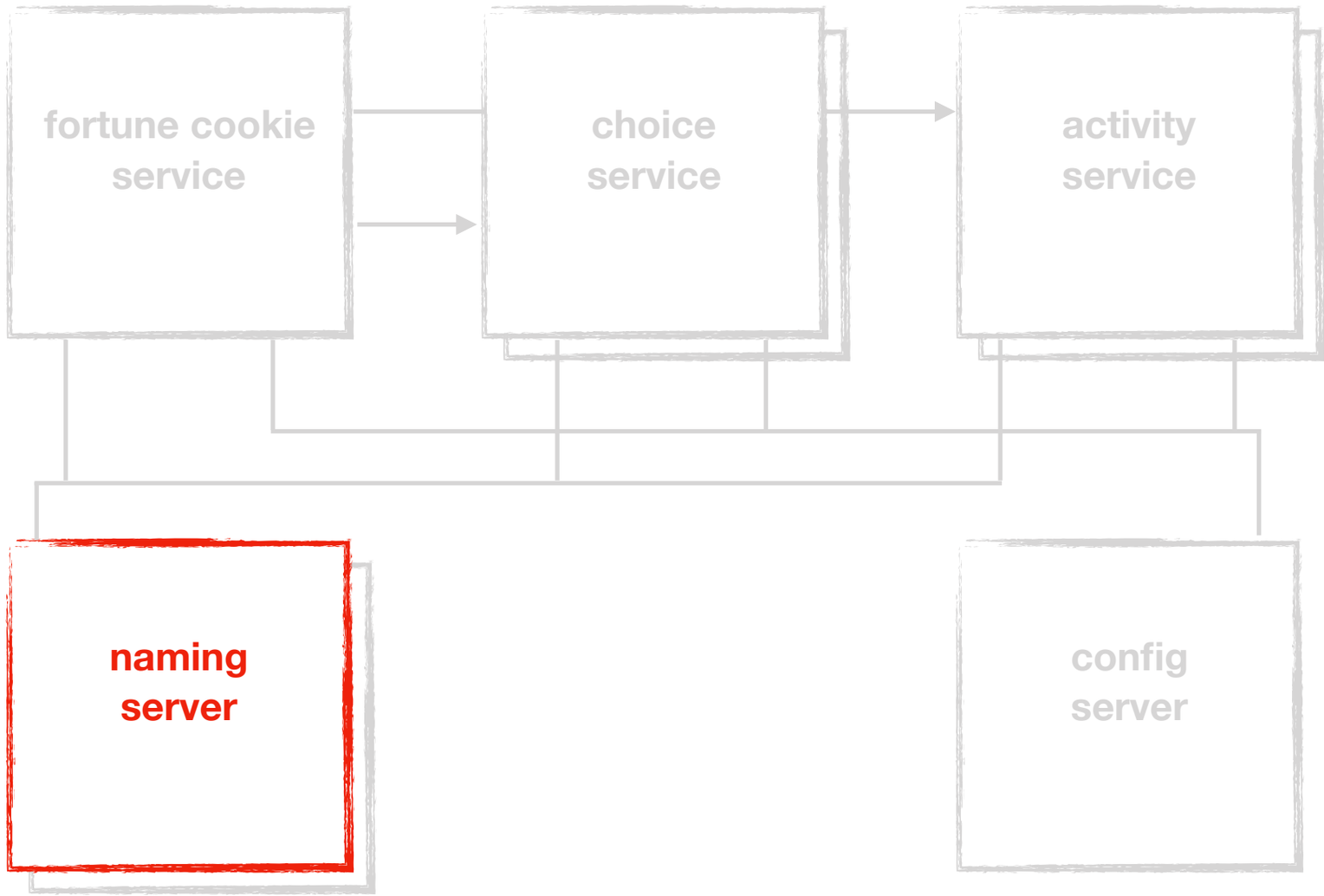
Spring Cloud

I Serwisy i Eureka

- **serwer Eureka**
- serwis fortune-cookie-service
  - jako klient Eureki
- zadanie - utworzenie serwisów, klientów Eureki
  - decision-service
  - activity-service
- refaktor - fortune-cookie-service
  - korzysta z decision-service i activity-service
- refaktor Eureka - 2 instancje

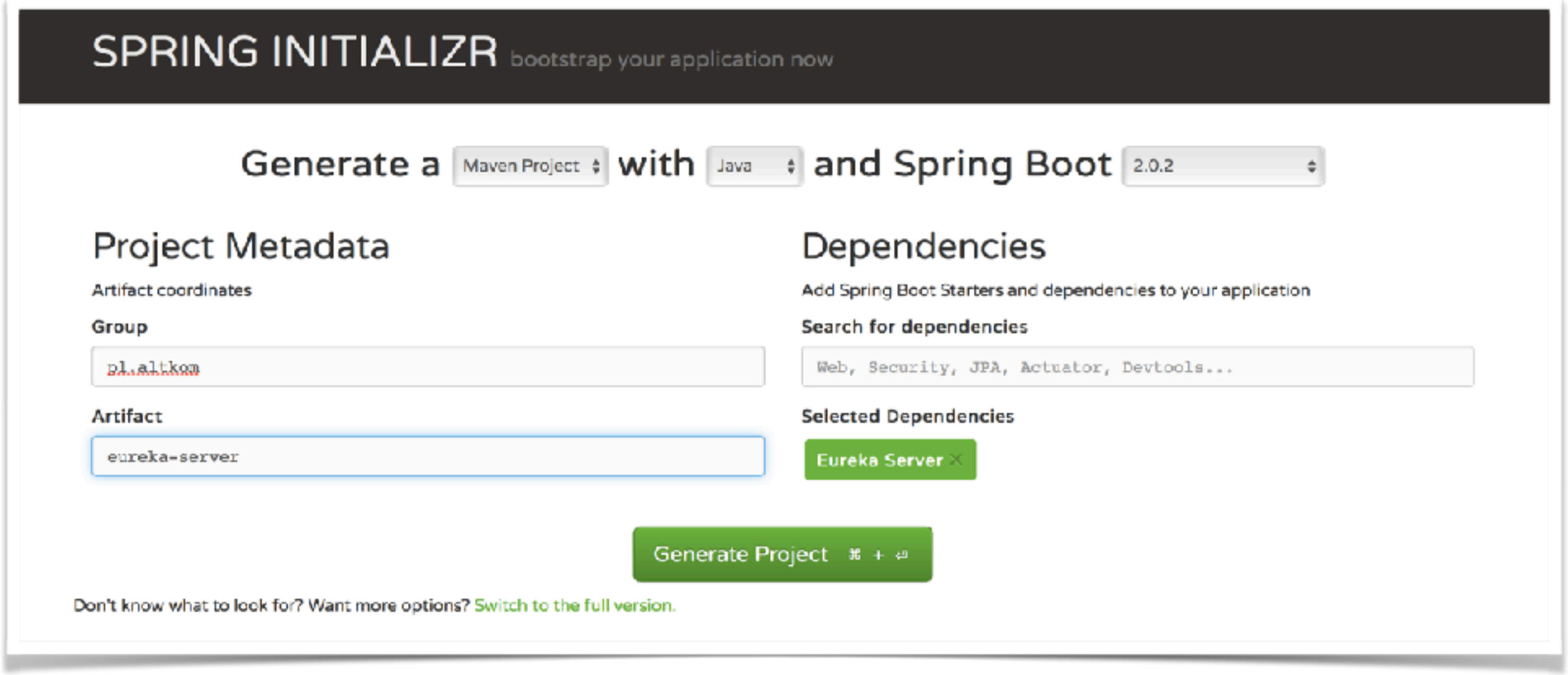
Spring Cloud

Serwer Eureka



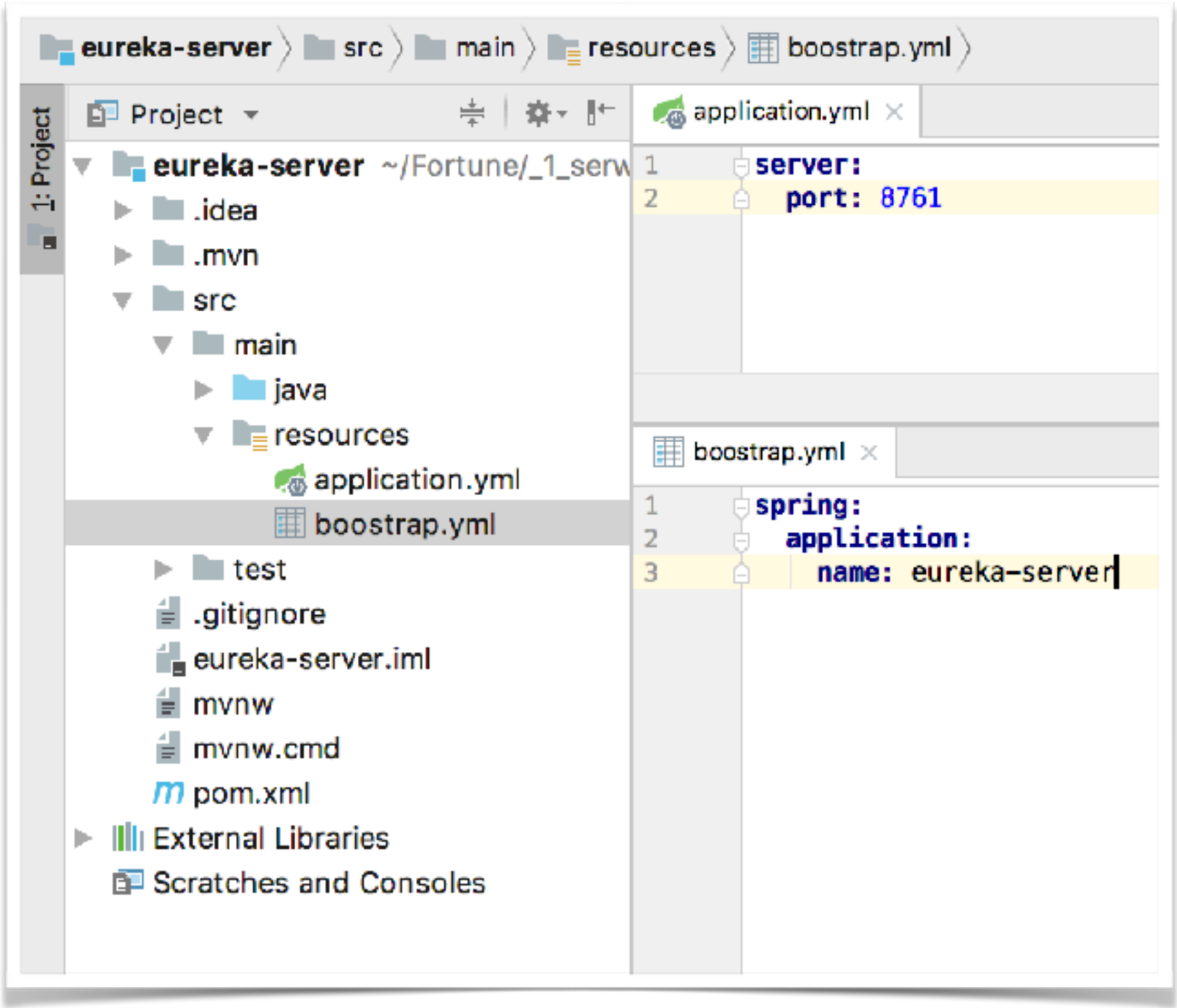
Spring Cloud

szkielet projektu

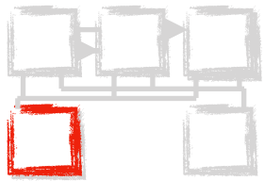


Spring Cloud

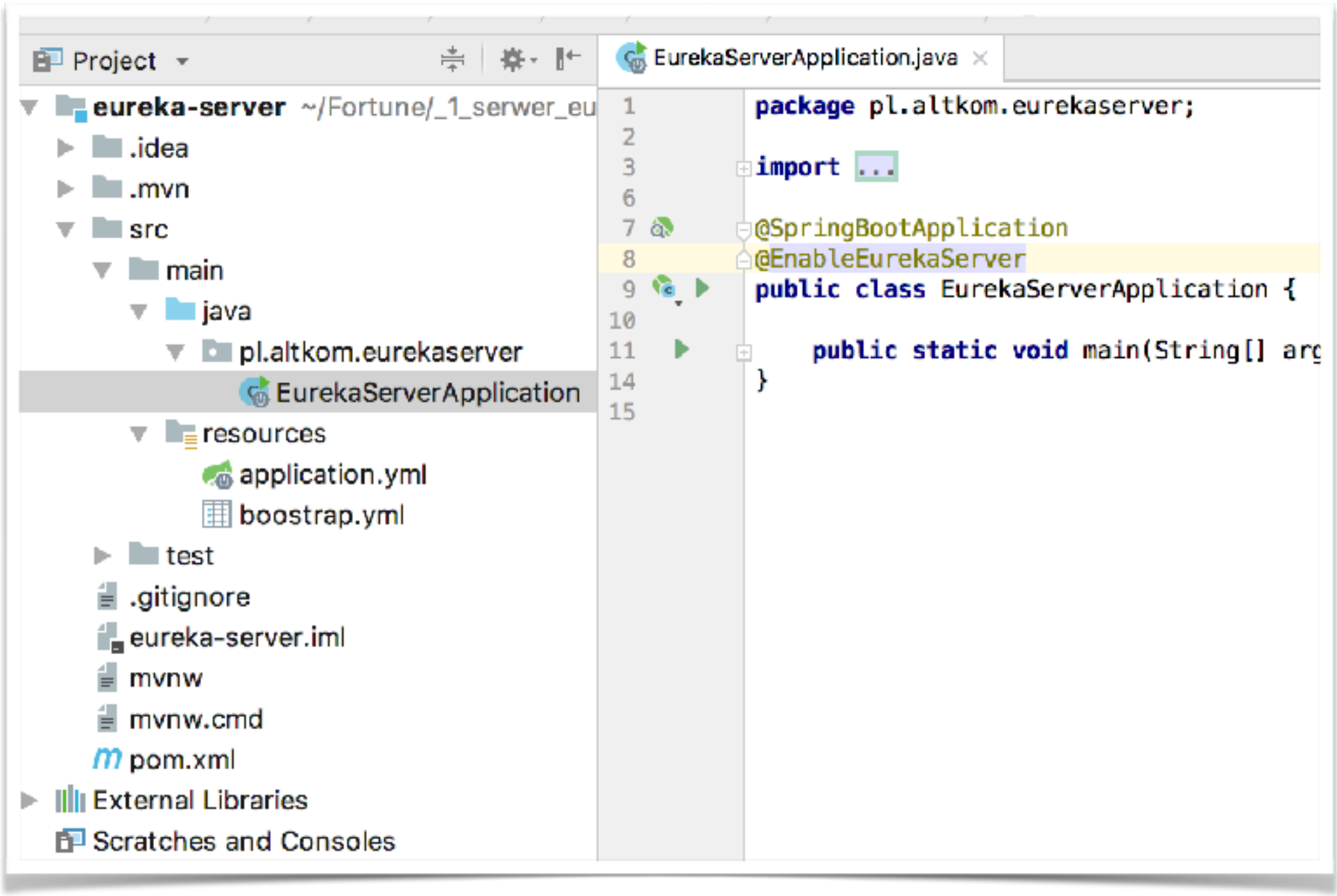
pliki konfiguracyjne



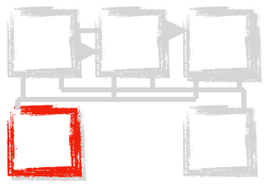
Spring Cloud



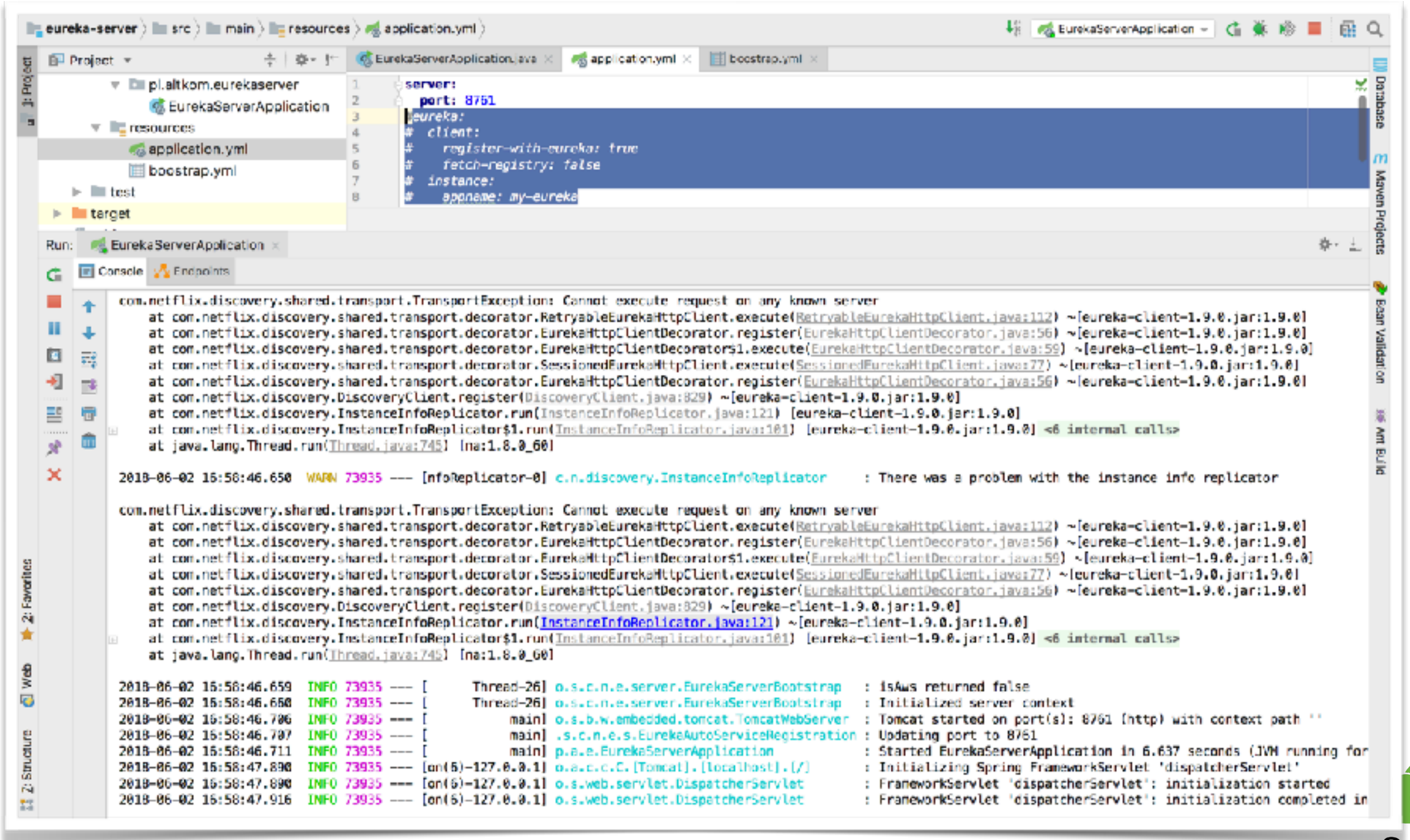
dodanie @



Spring Cloud

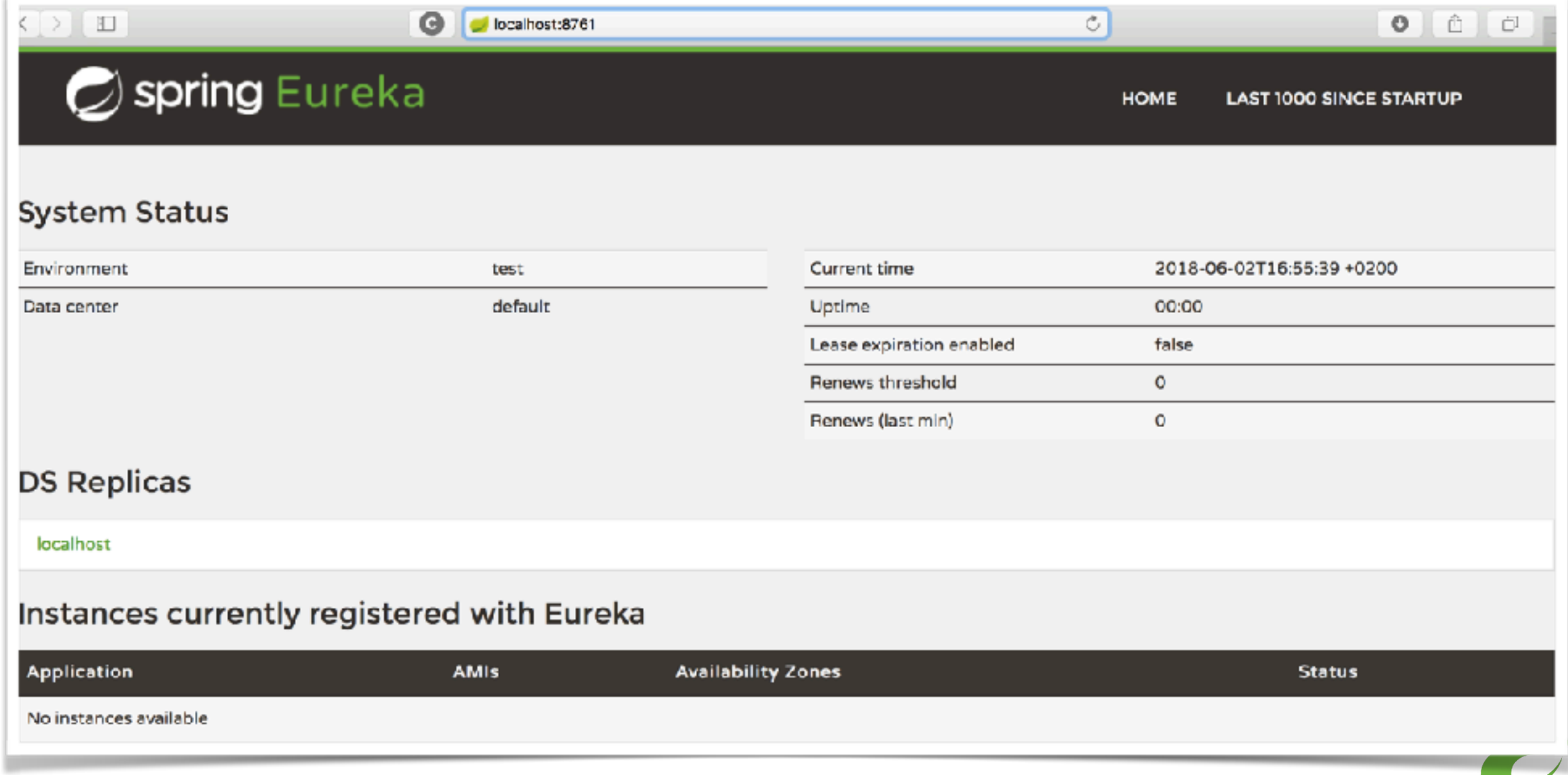


niechęć do samotności ;)



Spring
Cloud

UI Eureki



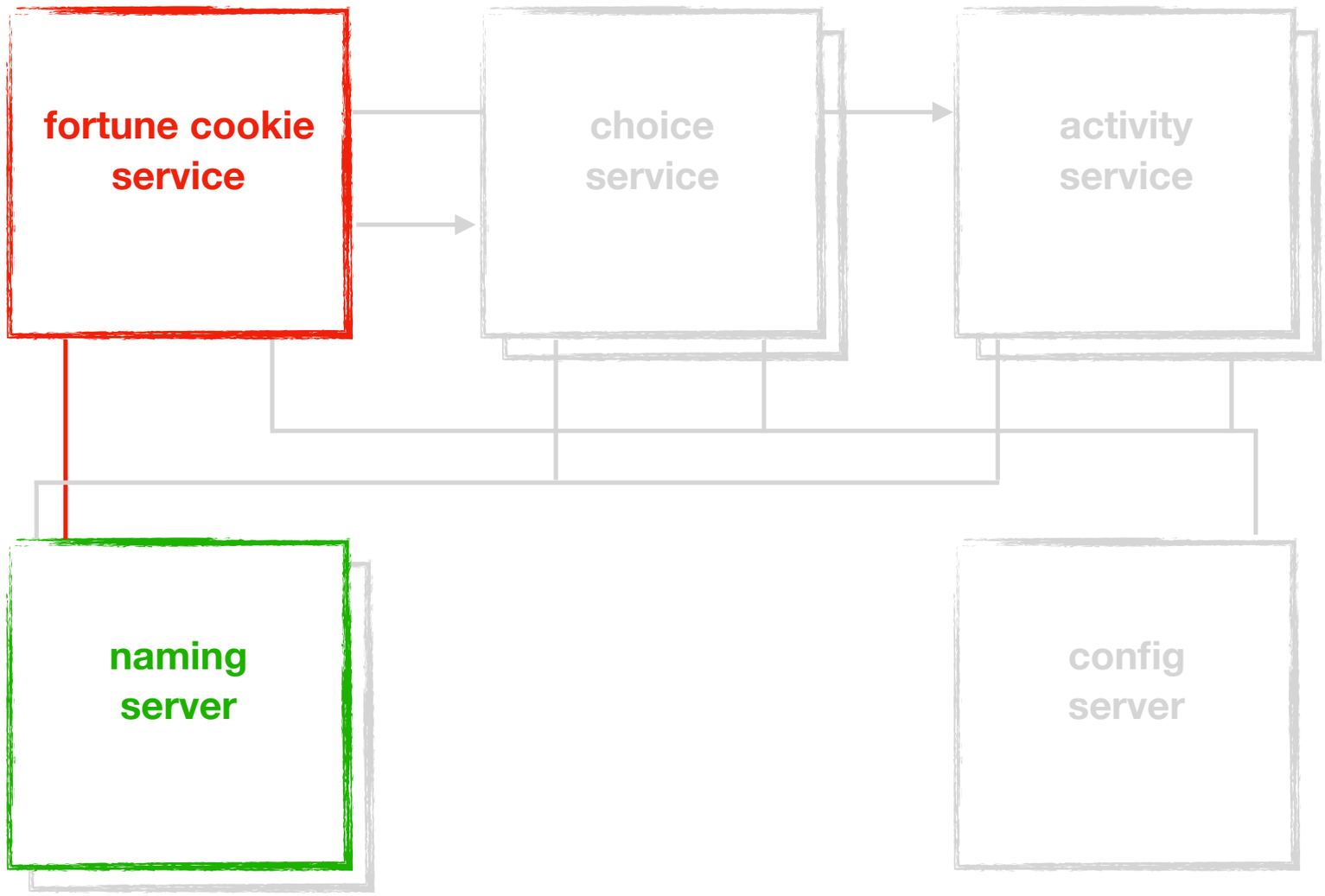
Spring Cloud

I Serwisy i Eureka

- serwer Eureka
- **serwis fortune-cookie-service**
  - **jako klient Eureki**
- zadanie - utworzenie serwisów, klientów Eureki
  - decision-service
  - activity-service
- refaktor - fortune-cookie-service
  - korzysta z decision-service i activity-service
- refaktor Eureka - 2 instancje

Spring Cloud

fortune-cookie-service



Spring Cloud

szkielet projektu

SPRING INITIALIZR bootstrap your application now

Generate a Maven Project with Java and Spring Boot 2.0.2

Project Metadata

Artifact coordinates

Group

pl.altkom

Artifact

fortune-cookie-service

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

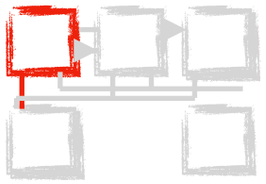
Web, Security, JPA, Actuator, Devtools...

Selected Dependencies

Web × Eureka Discovery × Actuator ×

Generate Project ⌘ + ⏎

Spring Cloud



pliki konfiguracyjne

fortune-cookie-service [~/Fortune/_2_fortune_cookie_service/fortune-cookie-service] - .../src/main/resources/application.yml [fortune-cookie-service]

fortune-cookie-service › src › main › resources › application.yml

FortuneCookieServiceApplication

Project

bootstrap.yml

```
1  spring:
2    application:
3      name: fortune-cookie-service
```

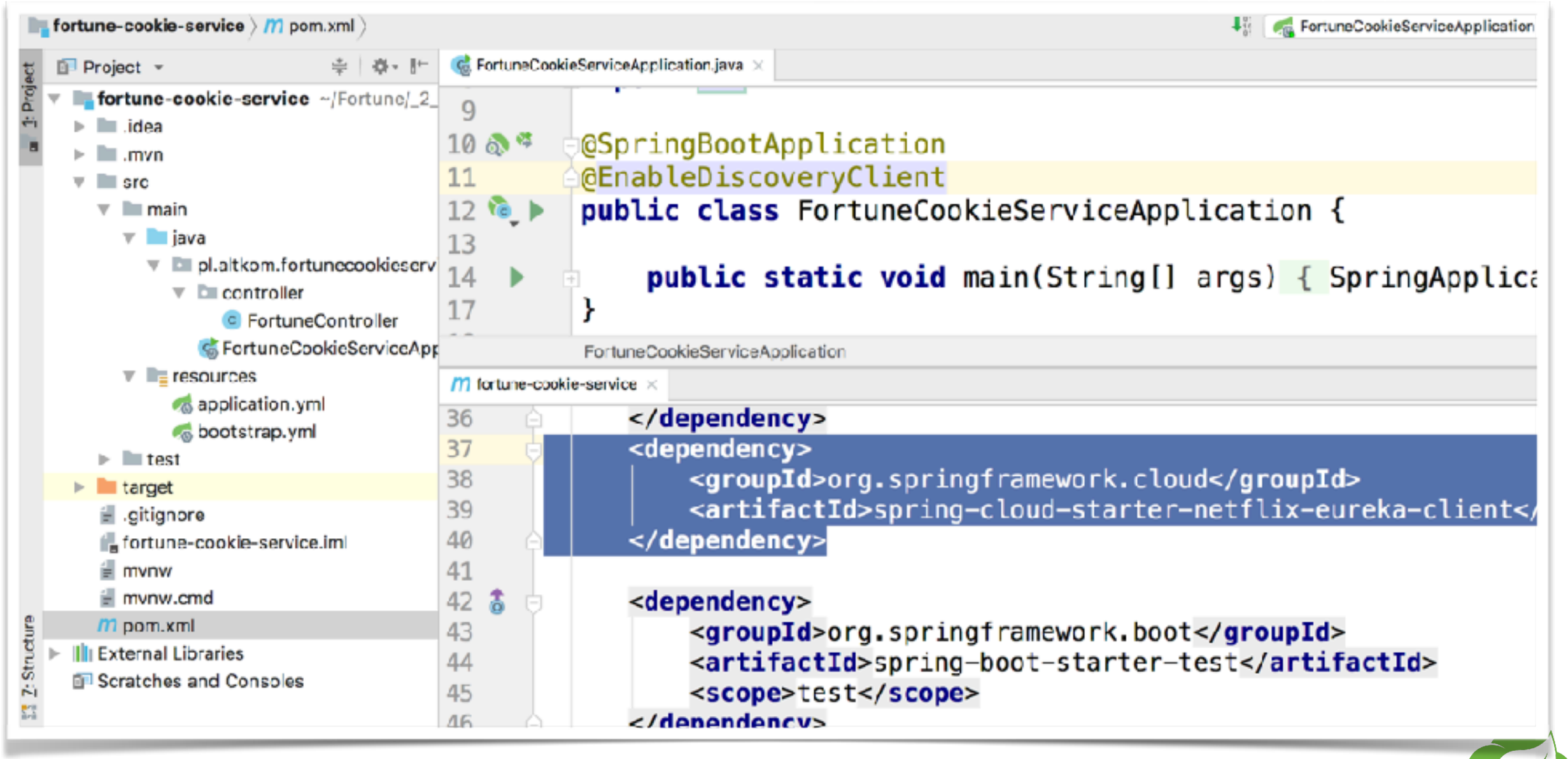
Document 1/1 › spring: › application: › name: › fortune-cookie-servi...

application.yml

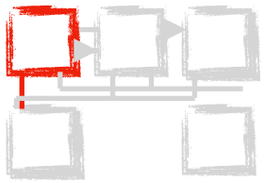
```
1  eureka:
2    client:
3      serviceUrl:
4        defaultZone: http://localhost:8761/eureka/
5  server:
6    port: 8080
7  fortunes: It's a good day for a cookie!, Take a nap!
```


Spring Cloud

zależność i @



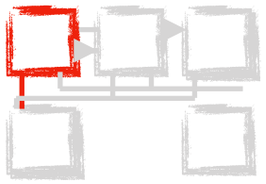
Spring Cloud



kontroler

```
@RestController
public class FortuneController {

    @Value("${fortunes}")
    private String fortunes;

    @GetMapping("/fortune")
    @ResponseBody // automatic conversion to JSON
    public Fortune fortune() { return getFortune(); }

    private Fortune getFortune() {
        String[] sentences = fortunes.split( regex: ",");
        int i = (int)Math.round(Math.random() * (sentences.length - 1));
        return new Fortune(sentences[i]);
    }

    class Fortune {
        private String fortune;

        public Fortune(String fortune) { this.fortune = fortune; }

        public String getFortune() { return fortune; }
    }
}
```


Spring Cloud



test serwisu

Normal | Basic Auth | Digest Auth | OAuth 1.0 | No environment

http://localhost:8080/fortune GET

Header Value Manage presets

Send Preview Add to collection

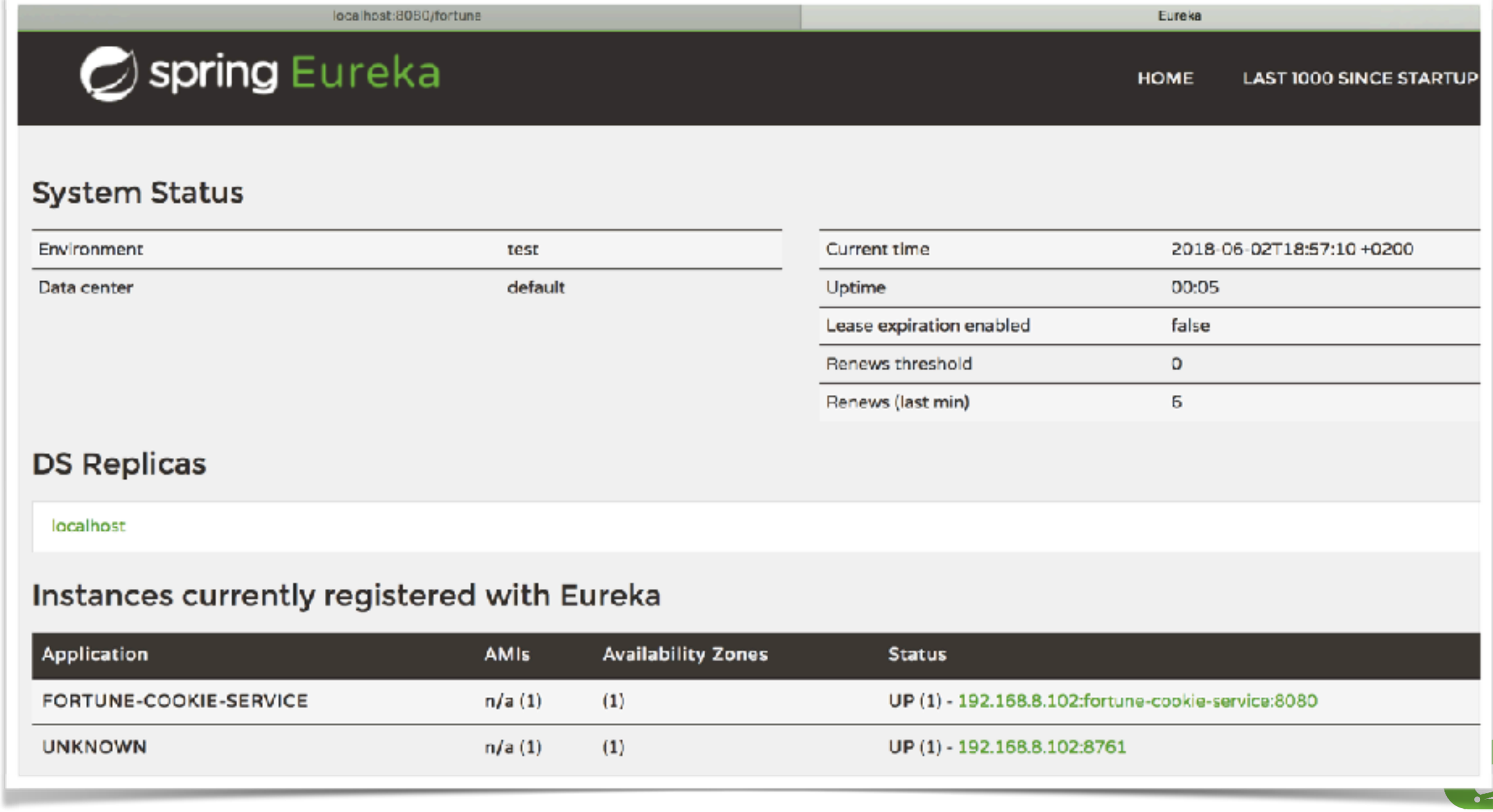
Body | Headers (3) STATUS 200 TIME 154 ms

Pretty Raw Preview JSON XML

```
1 {
2     "fortune": "It's a good day for a cookie!"
3 }
```

Spring Cloud

serwis w Eurece



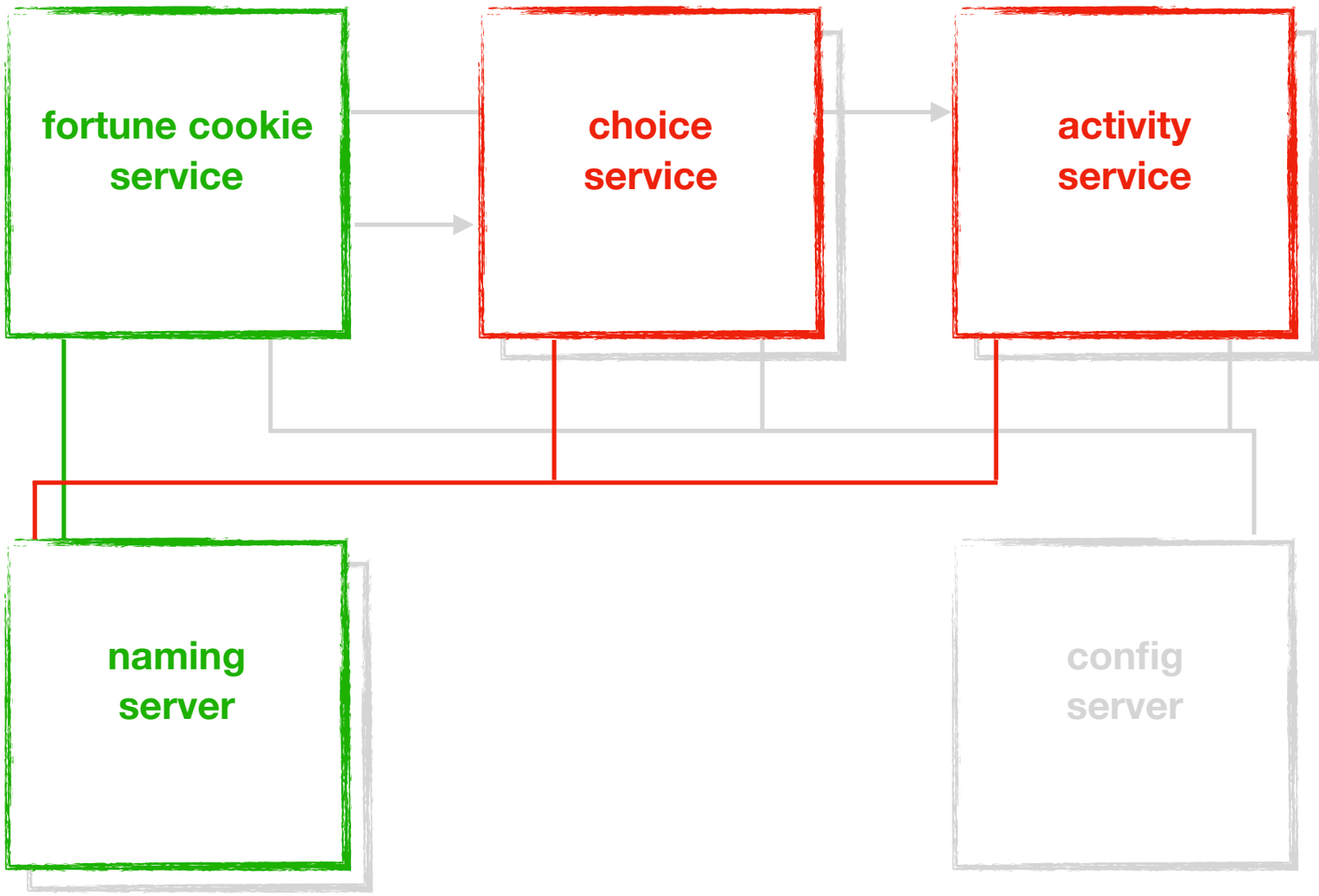
Spring Cloud

I Serwisy i Eureka

- serwer Eureka
- **serwis fortune-cookie-service**
  - **jako klient Eureki**
- **zadanie - utworzenie serwisów, klientów Eureki**
  - **decision-service**
  - **activity-service**
- refaktor - fortune-cookie-service
  - korzysta z decision-service i activity-service
- refaktor Eureka - 2 instancje

Spring Cloud

zadanie - implementacja

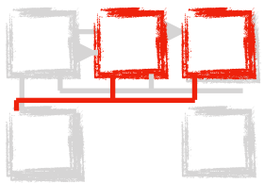


Spring Cloud

zadanie - implementacja

- **analogicznie** do **fortune-cookie-service**
- decision-service
  - **8000**
  - **Get**Mapping(“/decision")
  - It’s a good idea, It’s a bad day etc.
- activity-service
  - **8010**
  - **Get**Mapping(“/activity”)
  - to run a marathon, for cookies etc.

Spring Cloud



test decision-service

Normal | Basic Auth | Digest Auth | OAuth 1.0 | No environment

http://localhost:8000/decision

GET

Header | Value

Manage presets

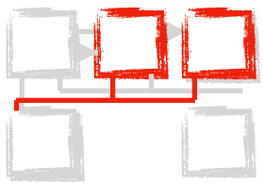
Send | Preview | Add to collection

Body | Headers (3) | STATUS 200 | TIME 9 ms

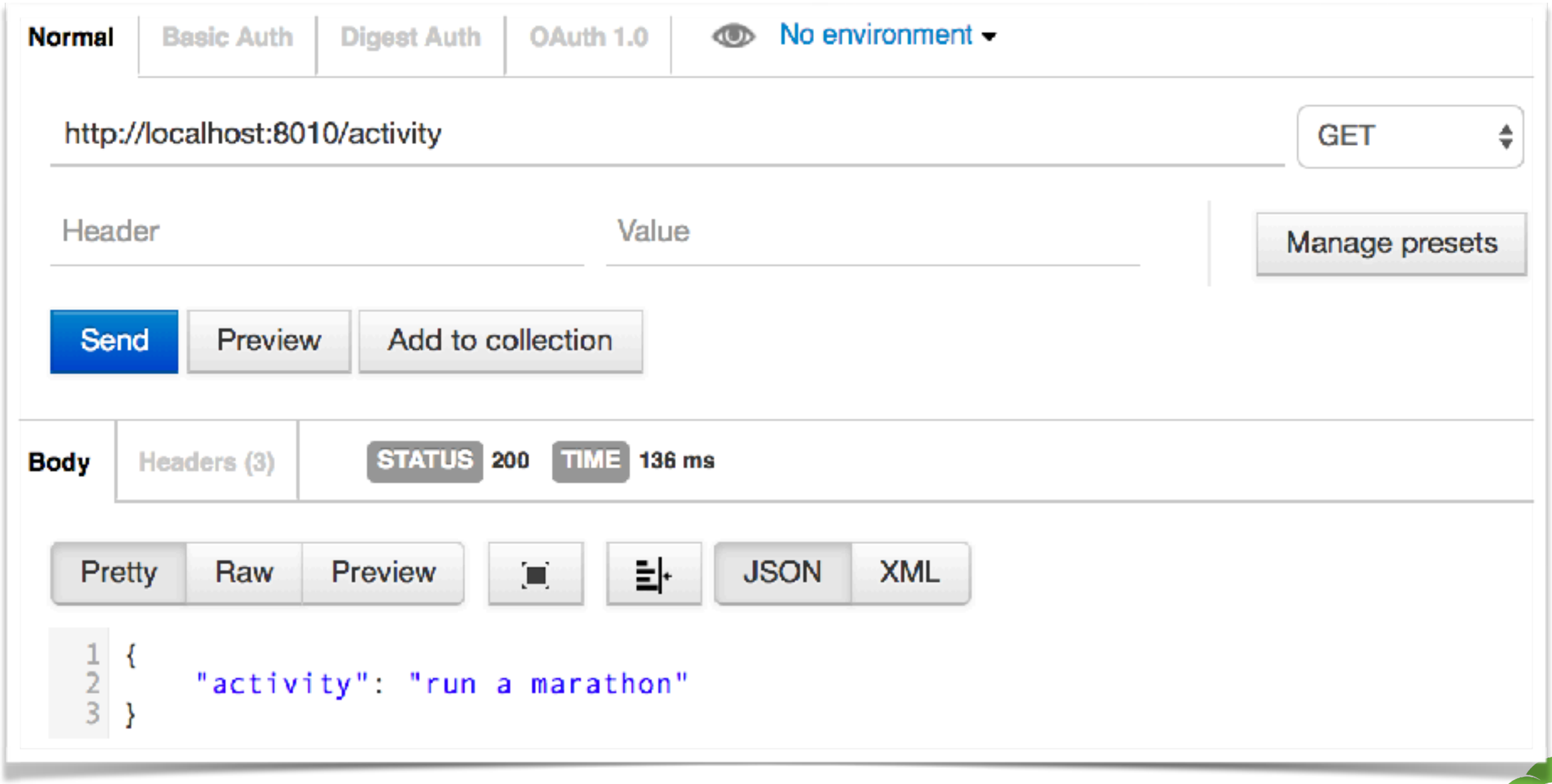
Pretty | Raw | Preview | JSON | XML

```
1 {
2     "decision": "It's time"
3 }
```

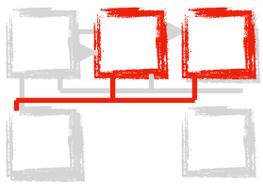

Spring Cloud



test activity-service



Spring Cloud



serwisy w Eurece

spring Eureka

HOME LAST 1000 SINCE STARTUP

System Status

Environment	test	Current time	2018-06-02T19:45:12 +0200
Data center	default	Uptime	00:53
		Lease expiration enabled	true
		Renews threshold	6
		Renews (last min)	16

DS Replicas

localhost

Instances currently registered with Eureka

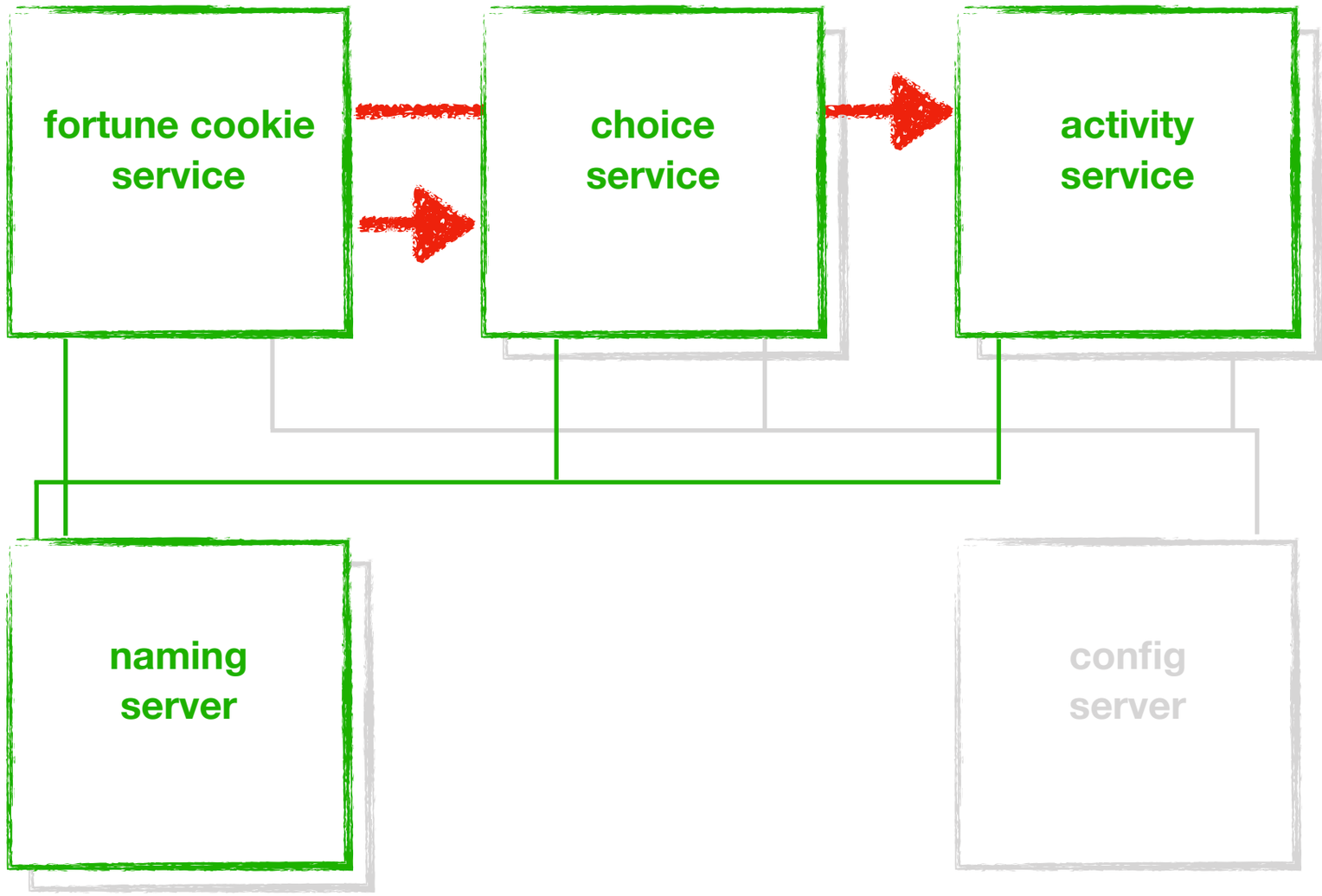
Application	AMIs	Availability Zones	Status
ACTIVITY-SERVICE	n/a (1)	(1)	UP (1) - 192.168.8.102:activity-service:8010
DECISION-SERVICE	n/a (1)	(1)	UP (1) - 192.168.8.102:decision-service:8000
FORTUNE-COOKIE-SERVICE	n/a (1)	(1)	UP (1) - 192.168.8.102:fortune-cookie-service:8080
UNKNOWN	n/a (1)	(1)	UP (1) - 192.168.8.102:8761

I Serwisy i Eureka

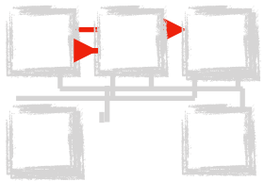
- serwer Eureka
- **serwis fortune-cookie-service**
  - **jako klient Eureki**
- zadanie - utworzenie serwisów, klientów Eureki
  - decision-service
  - activity-service
- **refaktor - fortune-cookie-service**
  - **korzysta z decision-service i activity-service**
- refaktor Eureka - 2 instancje

Spring Cloud

refaktor fortune-cookie-service



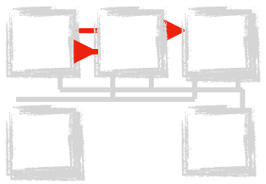
Spring Cloud



konroller

```
@Autowired
private org.springframework.cloud.client.discovery.DiscoveryClient discoveryClient;

@GetMapping("/fortune")
@ResponseBody // automatic conversion to JSON
public Fortune fortune() {
    String fortune =
            getDataFromService("DECISION-SERVICE") + " " + getDataFromService("ACTIVITY-SERVICE");
    return new Fortune(fortune);
}

public String getDataFromService(String service) {
    List<ServiceInstance> list = discoveryClient.getInstances(service);
    if (list != null && list.size() > 0) {
        URI uri = list.get(0).getUri();
        if (uri != null) {
            String responseInJSON = (new RestTemplate()).getForObject(uri, String.class);
            return convertJSONResponseToSentence(responseInJSON);
        }
    }
    return null;
}
```


Spring
Cloud



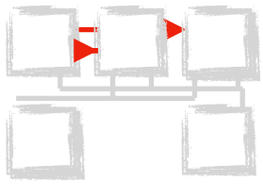
konwersja JSON’a

```
private String convertJSONResponseToSentence(String responseInJSON) {
    JSONArray dataFromService = JsonPath.read(responseInJSON, jsonPath: "$..*");
    return (String)dataFromService.get(0);
}
```

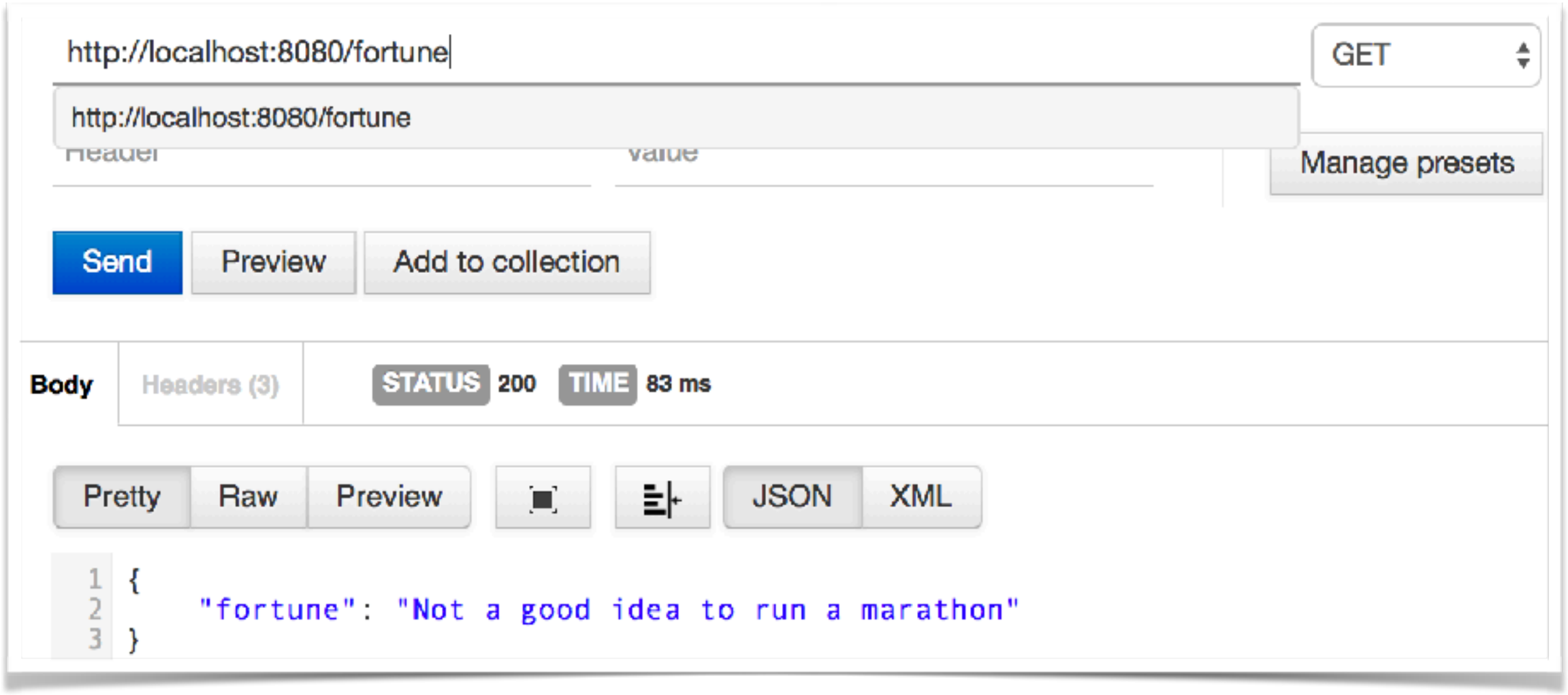
ontroller › Fortune

```
<dependency>
    <groupId>com.jayway.jsonpath</groupId>
    <artifactId>json-path</artifactId>
    <version>2.4.0</version>
</dependency>
```


Spring Cloud



test



*** uwaga, będzie bug - co trzeba poprawić? :)**

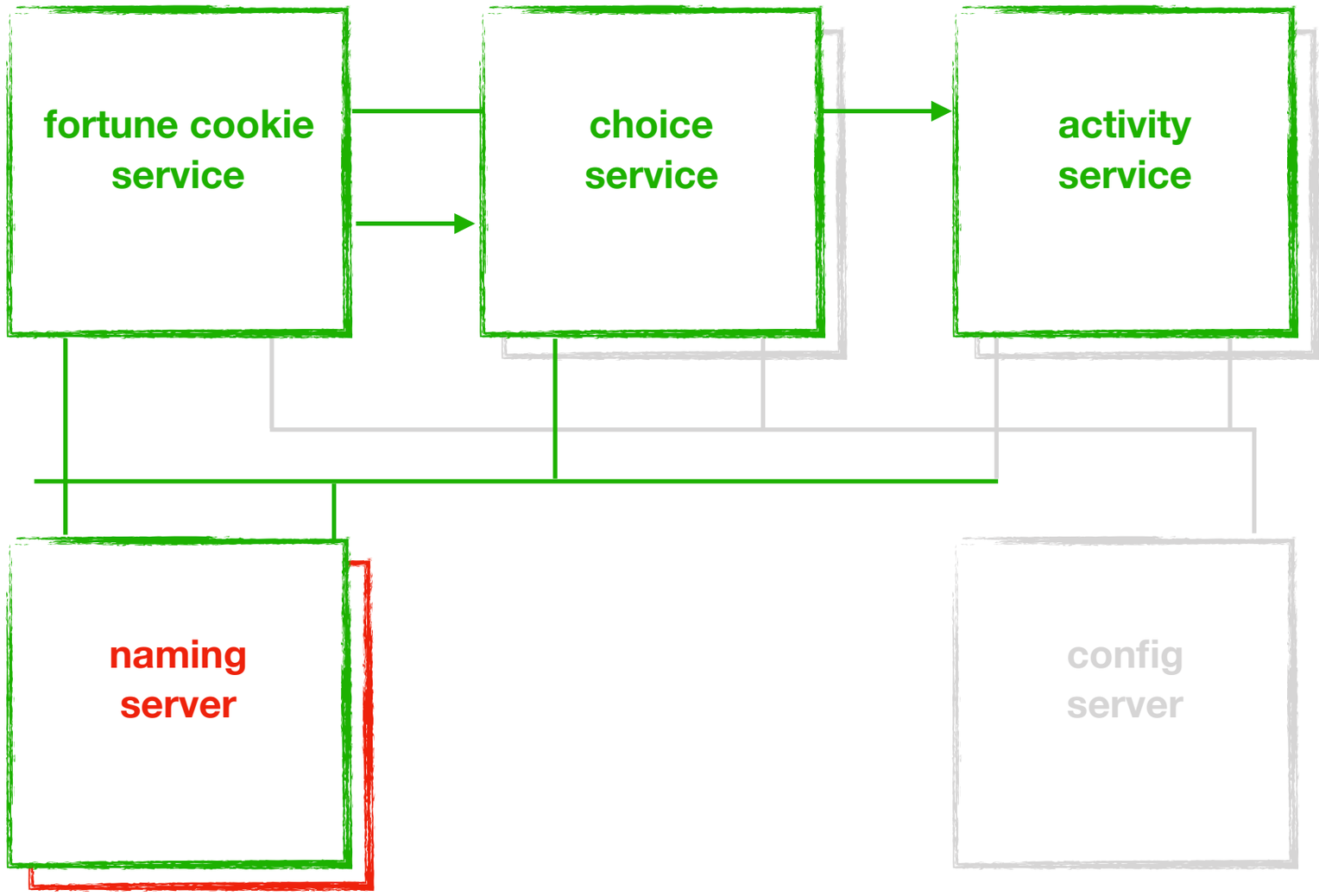
Spring Cloud

I Serwisy i Eureka

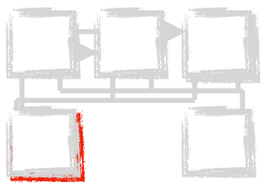
- serwer Eureka
- **serwis fortune-cookie-service**
  - **jako klient Eureki**
- zadanie - utworzenie serwisów, klientów Eureki
  - decision-service
  - activity-service
- refaktor - fortune-cookie-service
  - korzysta z decision-service i activity-service
- **refaktor Eureka - 2 instancje**

Spring Cloud

refaktor Eureka



Spring Cloud

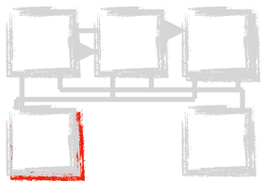


dodanie wpisu

c:\WINDOWS\system32\drivers\etc\hosts

```
# START Eureka
127.0.0.1    eureka-primary
127.0.0.1    eureka-secondary
# END Eureka
```


Spring Cloud



plik konfiguracyjny - eureka

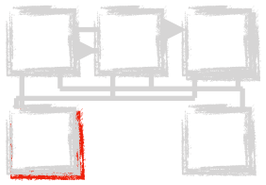
eureka-server › src › main › resources › application.yml

Project

```
eureka-server ~/Fortune/_5_refactor_eure
  .idea
  .mvn
  src
    main
      java
      resources
        application.yml
    test
  target
  .gitignore
  eureka-server.iml
  mvnw
  mvnw.cmd
  pom.xml
External Libraries
Scratches and Consoles
```

application.yml

```yaml
---
spring:
  profiles: eureka-primary
server:
  port: 8761
eureka:
  instance:
    hostname: eureka-primary
  client:
    registerWithEureka: true
    fetchRegistry: true
    serviceUrl:
      defaultZone: http://eureka-secondary:8762/eureka
---
spring:
  profiles: eureka-secondary
server:
  port: 8762
eureka:
  instance:
    hostname: eureka-secondary
  client:
    registerWithEureka: true
    fetchRegistry: true
    serviceUrl:
      defaultZone: http://eureka-primary:8761/eureka
```

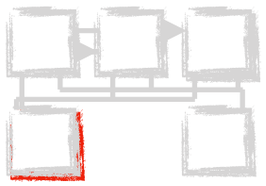


plik konfiguracyjny - serwisy

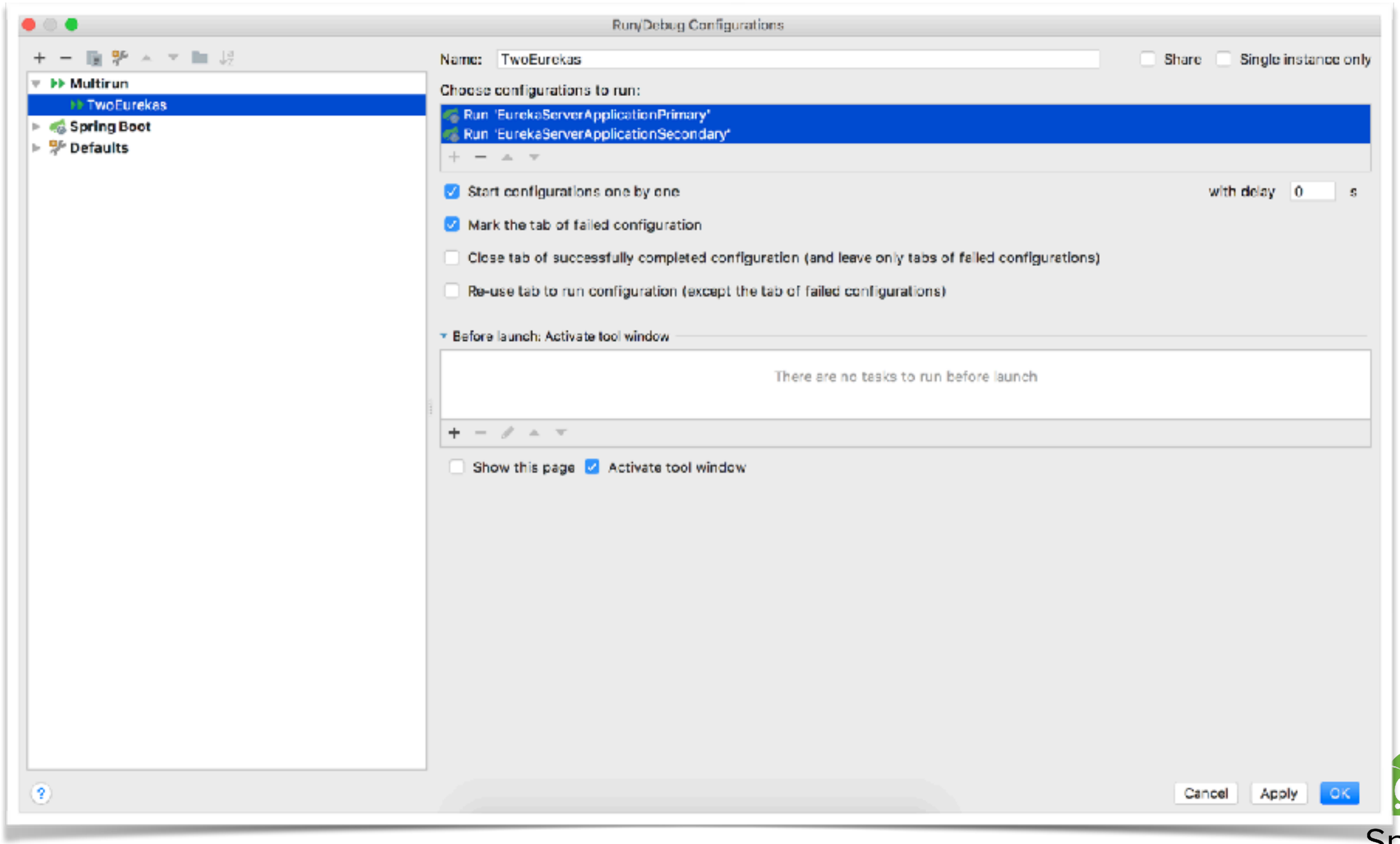
we wszystkich serwisach:

```yaml
eureka:
  client:
    serviceUrl:
      defaultZone: http://eureka-primary:8761/eureka,http://eureka-secondary:8762/eureka
```

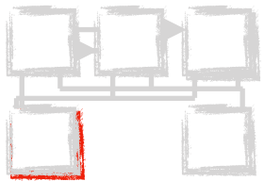

Spring Cloud



użycie Multirun - 2 instancje



Spring Cloud



UI Eureki

spring Eureka

HOME LAST 1000 SINCE STARTUP

System Status

Environment	test
Data center	default

Current time	2018-06-03T09:33:08 +0200
Uptime	00:01
Lease expiration enabled	false
Renews threshold	5
Renews (last min)	0

DS Replicas

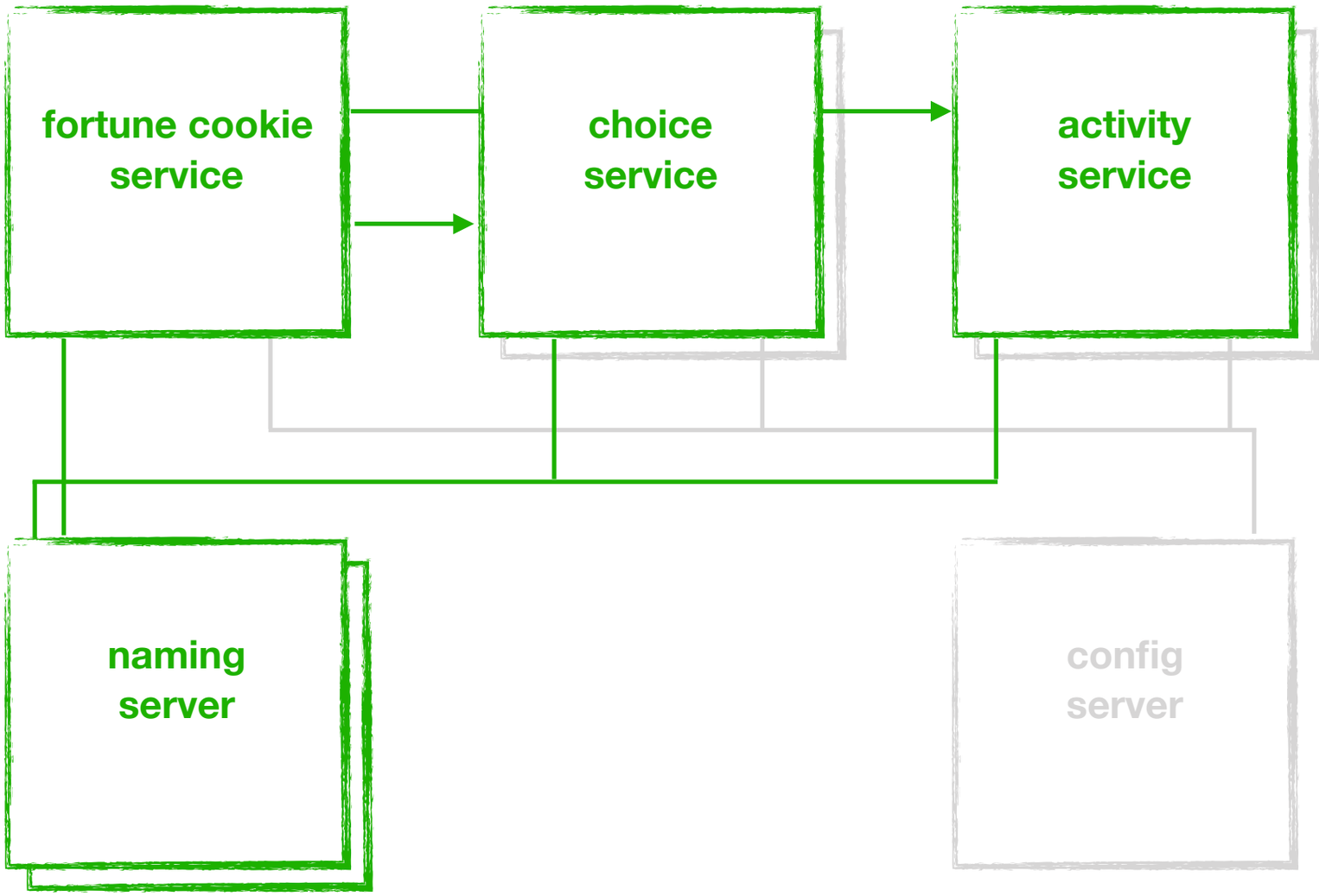
eureka-secondary

Instances currently registered with Eureka

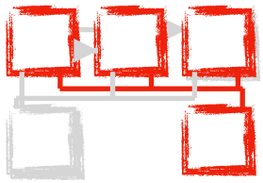
Application	AMIs	Availability Zones	Status
ACTIVITY-SERVICE	n/a (1)	(1)	UP (1) - 192.168.8.102:activity-service:8010
DECISION-SERVICE	n/a (1)	(1)	UP (1) - 192.168.8.102:decision-service:8000
FORTUNE-COOKIE-SERVICE	n/a (1)	(1)	UP (1) - 192.168.8.102:fortune-cookie-service:8080
UNKNOWN	n/a (2)	(2)	UP (2) - 192.168.8.102:8762 , 192.168.8.102:8761

General Info

Serwisy i Eureka - podsumowanie



Spring Cloud

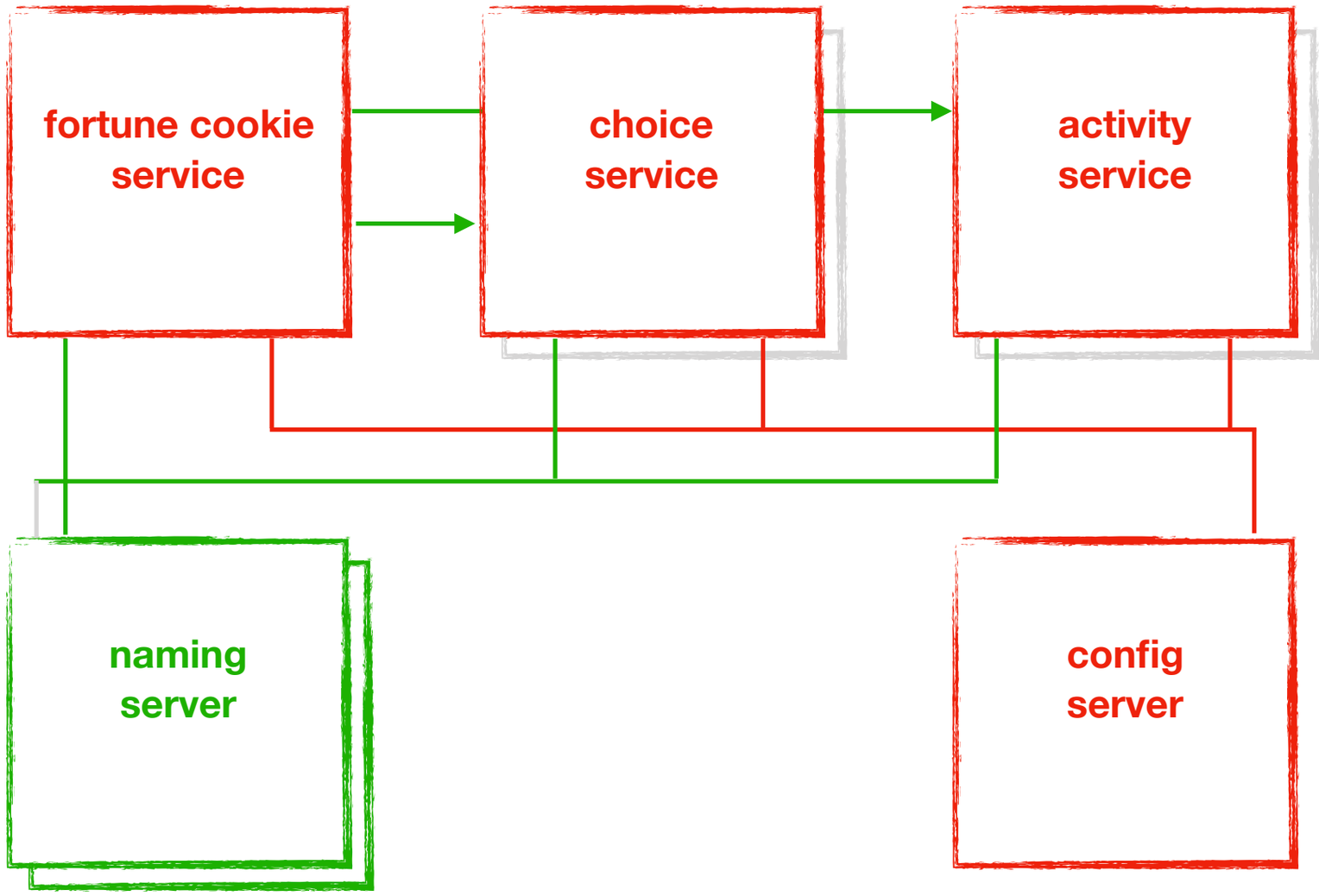


II Serwisy i Config Server

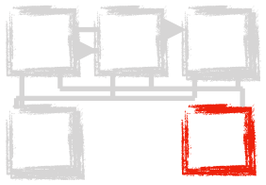
- utworzenie repo Git, pierwszy commit
- komponent Config Server
- refactor / podłączenie activity-service do Config Server
    - profil default
    - profil dev
- zadanie - podłączenie pozostałych serwisów
    - analogicznie jak activity-service
        - decision-service
        - fortune-cookie-service

Spring Cloud

II Serwisy i Config Server



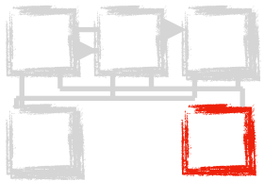
Spring Cloud



II Serwisy i Config Server

- **utworzenie repo Git, pierwszy commit**
- komponent Config Server
- refactor / podłączenie activity-service do Config Server
  - profil default
  - profil dev
- zadanie - podłączenie pozostałych serwisów
  - analogicznie jak activity-service
    - decision-service
    - fortune-cookie-service

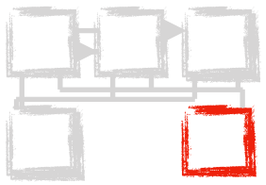
Spring Cloud



GIT i pierwszy commit

- utworzenie repo
  - git init
- utworzenie pliku konfiguracyjnego (vi lub edytor)
  - vi activity-service.yml
    - **zawartość pliku na kolejnym slajdzie**
- przygotowanie i commit pliku
  - git add -A && git commit -m 'add activity-service config, profiles: default & dev'

Spring Cloud



GIT i pierwszy commit

```
eureka:
  client:
    serviceUrl:
      defaultZone: http://eureka-primary:8761/eureka,http://eureka-secondary:8762/eureka
---
spring:
  profiles: lazy
activities: to grab a bear,to eat a cookie,to take a nap
---
spring:
  profiles: crazy
activities: to run a marathon, to go hiking
```

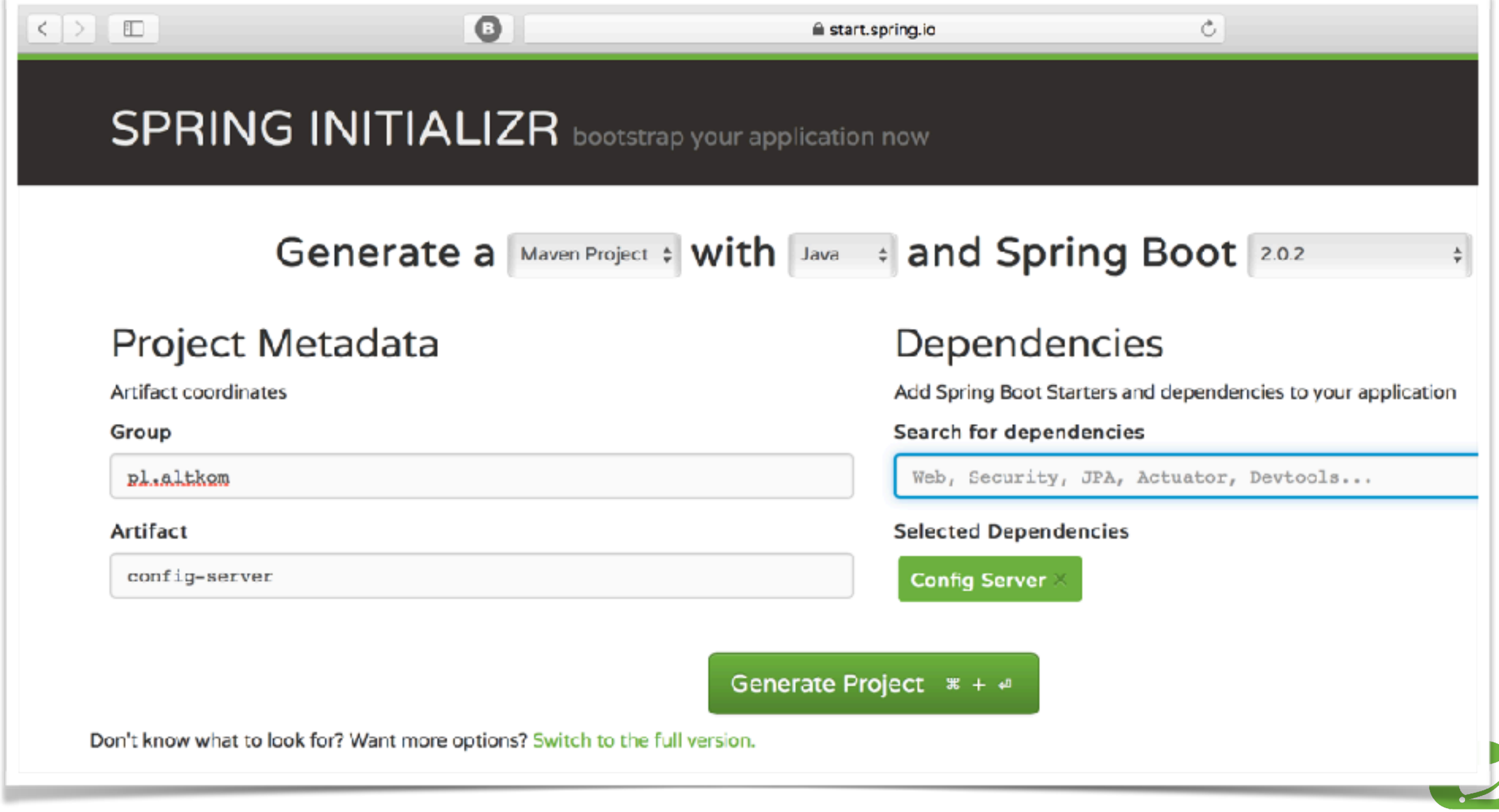

Spring Cloud

II Serwisy i Config Server

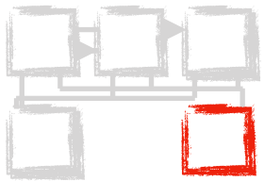
- utworzenie repo Git, pierwszy commit
- **komponent Config Server**
- refactor / podłączenie activity-service do Config Server
  - profil default
  - profil dev
- zadanie - podłączenie pozostałych serwisów
  - analogicznie jak activity-service
    - decision-service
    - fortune-cookie-service

Spring Cloud

szkielet projektu



Spring Cloud



konfig i @

config-server [~/Fortune/_6_config_server_activity/config-server] - .../src/main/resources/application.yml [config-ser

config-server › src › main › resources › application.yml

Project

1: Project

config-server ~/Fortune/_6_config_server_

- .idea
- .mvn
- src
  - main
    - java
      - pl.altkom.configserver
        - ConfigServerApplication
    - resources
      - application.yml
  - test
- target
- .gitignore
- config-server.iml
- mvnw
- mvnw.cmd
- pom.xml

External Libraries

Scratches and Consoles

application.yml

```yaml
spring:
  cloud:
    config:
      server:
        git:
          uri: file:///Users/lukasz/Fortune/_6_config_server_activity/localrepo
  application:
    name: config-server
server:
  port: 8888
```

Document 1/1 › spring: › cloud: › config: › server: › git: › uri: › file:///Users/lukasz...

ConfigServerApplication.java

```java
package pl.altkom.configserver;

import ...

@SpringBootApplication
@EnableConfigServer
public class ConfigServerApplication {

    public static void main(String[] args) { SpringApplication.run(ConfigServerAppl
}
```

Spring Cloud

test profilu default

Normal | Basic Auth | Digest Auth | OAuth 1.0 | No environment

http://localhost:8888/activity-service/default

GET

http://localhost:8888/activity-service/default

Manage presets

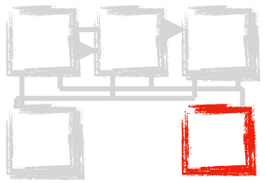
Send | Preview | Add to collection

Body | Headers (3) | STATUS 200 | TIME 85 ms

Pretty | Raw | Preview | JSON | XML


```
{
    "name": "activity-service",
    "profiles": [
        "default"
    ],
    "label": null,
    "version": "2aba6013727daeabf290a871f710521efb846b93",
    "state": null,
    "propertySources": [
        {
            "name": "file:///Users/lukasz/Fortune/_6_config_server_activity/localrepo/activity-service.yml (document #0)",
            "source": {
                "eureka.client.serviceUrl.defaultZone": "http://eureka-primary:8761/eureka,http://eureka-secondary:8762/eureka"
            }
        }
    ]
}
```


Spring Cloud



test profilu crazy

Normal | Basic Auth | Digest Auth | OAuth 1.0 | No environment

http://localhost:8888/activity-service/crazy GET

Header Value Manage presets

Send Preview Add to collection

Body | Headers (3) STATUS 200 TIME 60 ms

Pretty Raw Preview JSON XML


```
{
    "name": "activity-service",
    "profiles": [
        "crazy"
    ],
    "label": null,
    "version": "2aba6013727daeabf290a871f710521efb846b93",
    "state": null,
    "propertySources": [
        {
            "name": "file:///Users/lukasz/Fortune/_6_config_server_activity/localrepo/activity-service.yml (document #2)",
            "source": {
                "spring.profiles": "crazy",
                "activities": "to run a marathon, to go hiking"
            }
        },
        {
            "name": "file:///Users/lukasz/Fortune/_6_config_server_activity/localrepo/activity-service.yml (document #0)",
            "source": {
                "eureka.client.serviceUrl.defaultZone": "http://eureka-primary:8761/eureka,http://eureka-secondary:8762/eureka"
            }
        }
    ]
}
```

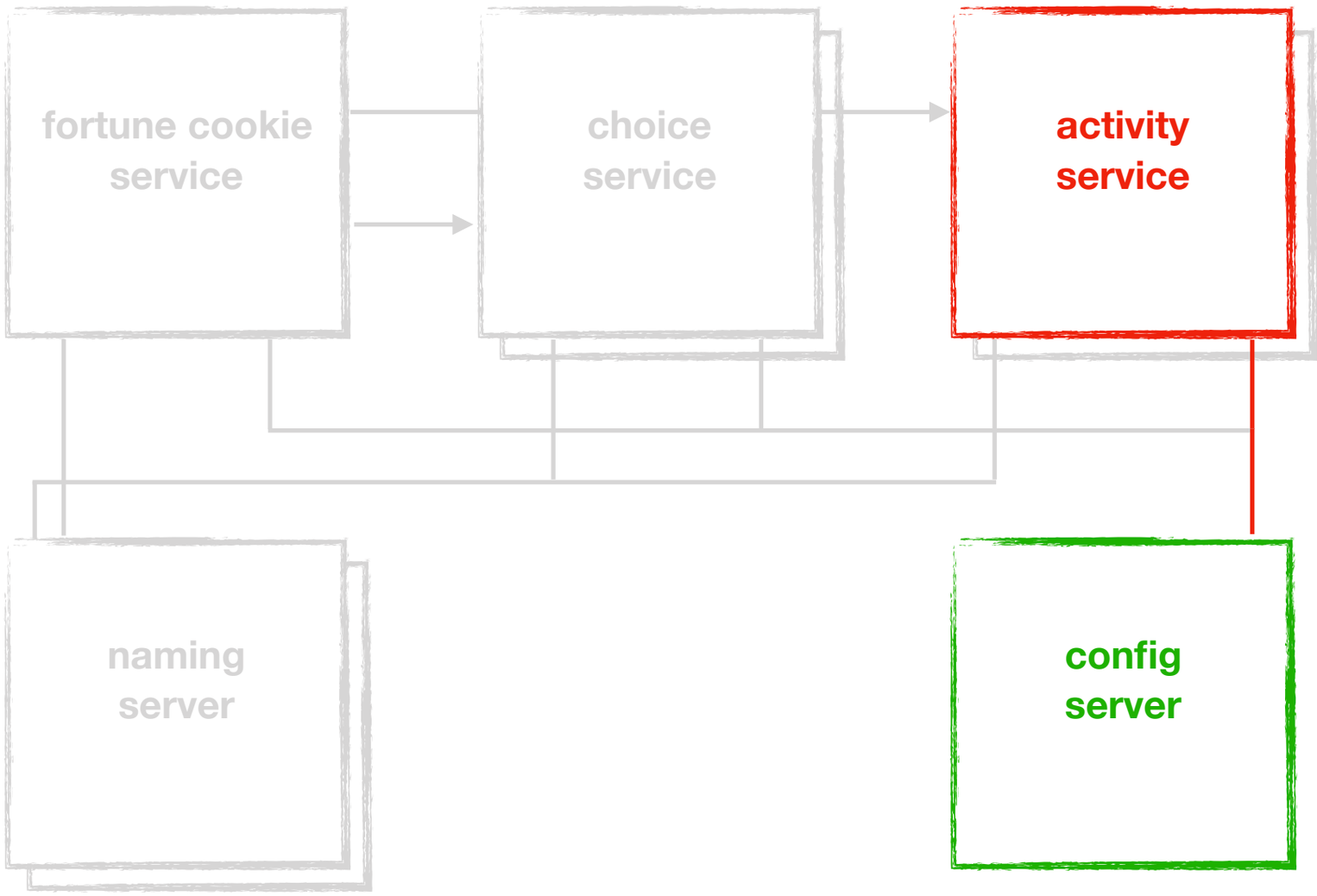

Spring Cloud

II Serwisy i Config Server

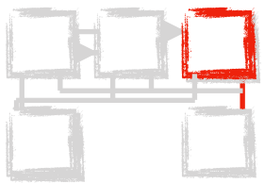
- utworzenie repo Git, pierwszy commit
- komponent Config Server
- **refactor / podłączenie activity-service do Config Server**
  - **profil lazy**
  - **profil crazy**
- zadanie - podłączenie pozostałych serwisów
  - analogicznie jak activity-service
    - decision-service
    - fortune-cookie-service

Spring Cloud

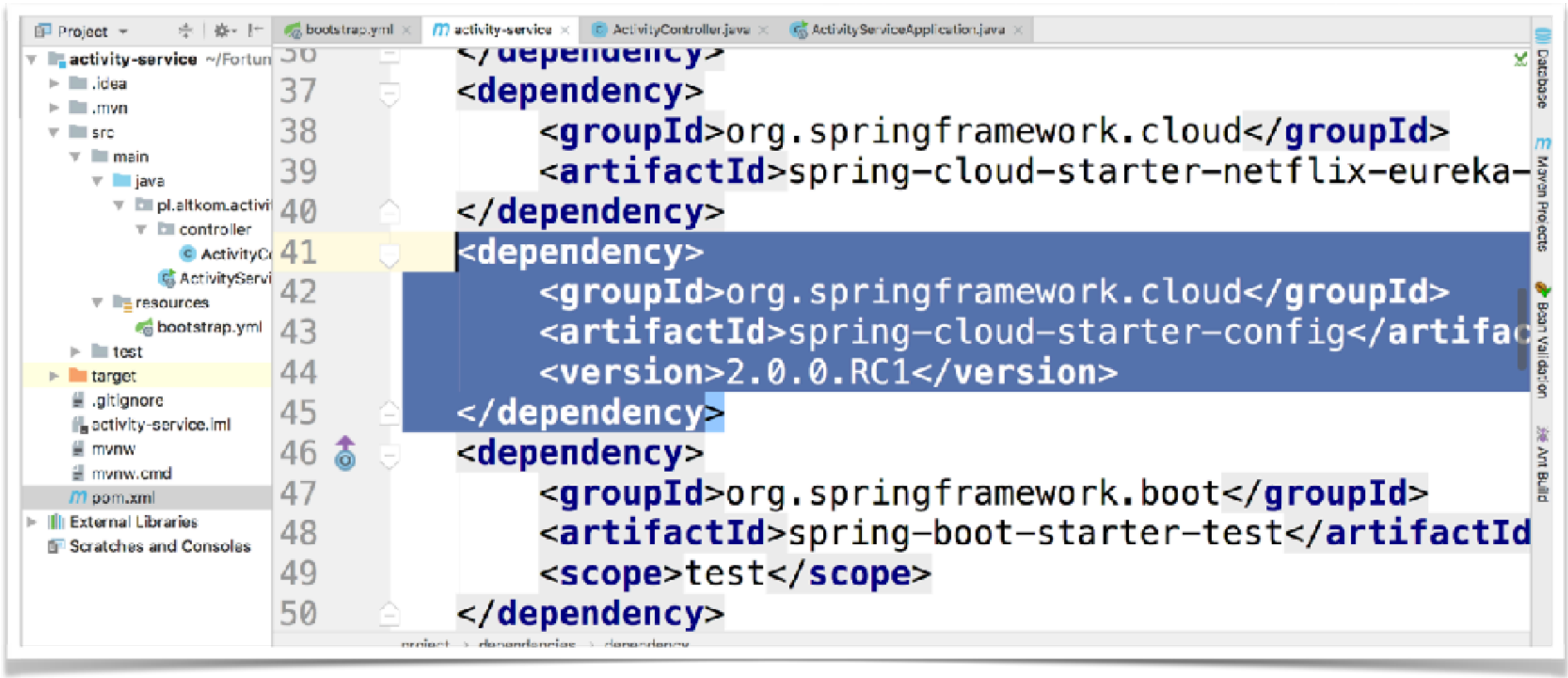
refaktor activity-service



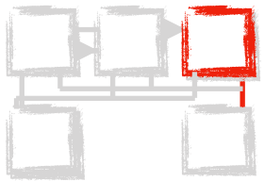
Spring Cloud



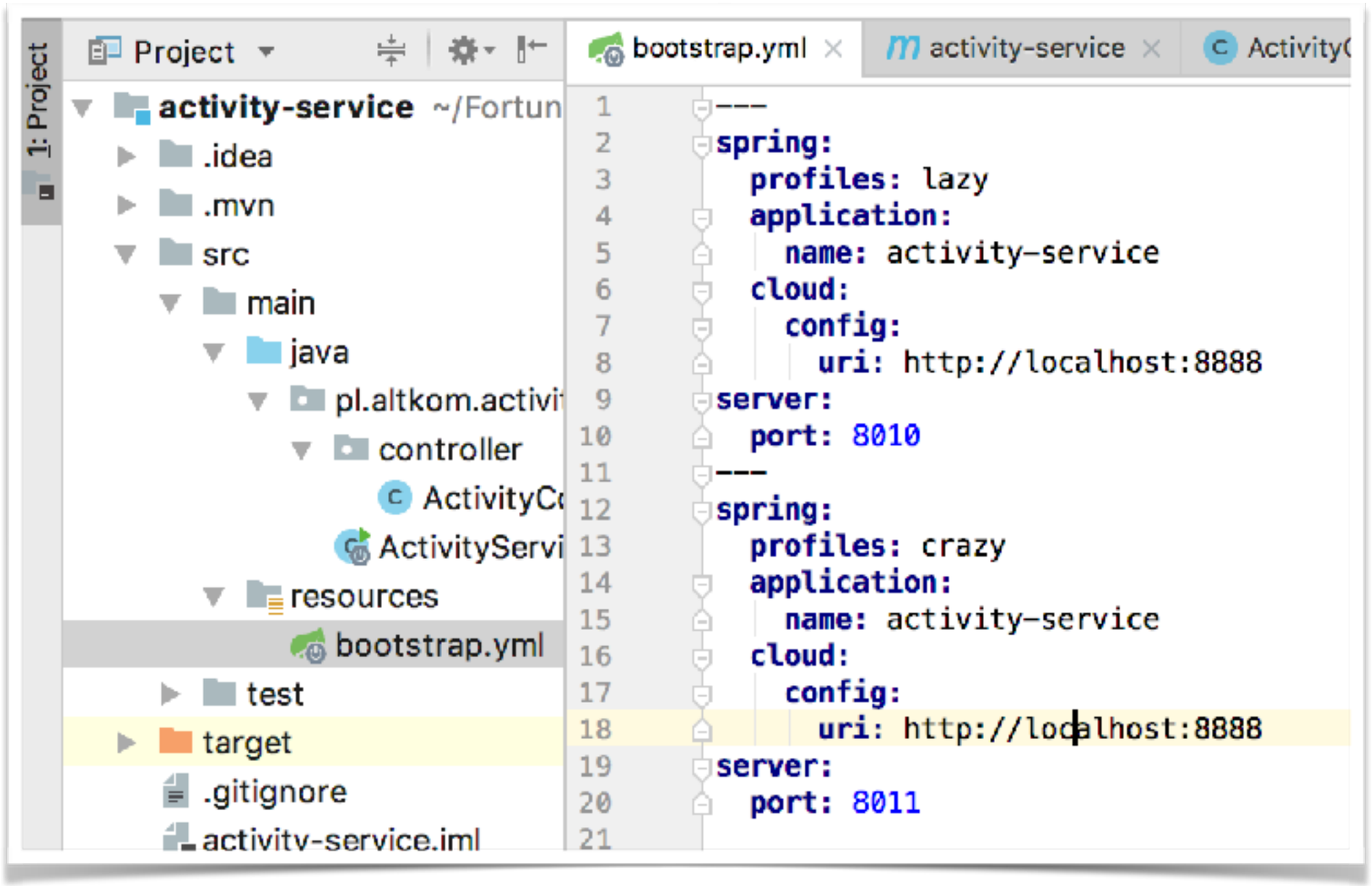
dodanie zależności



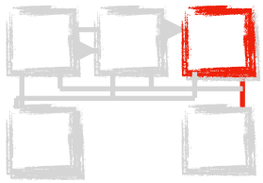
Spring Cloud



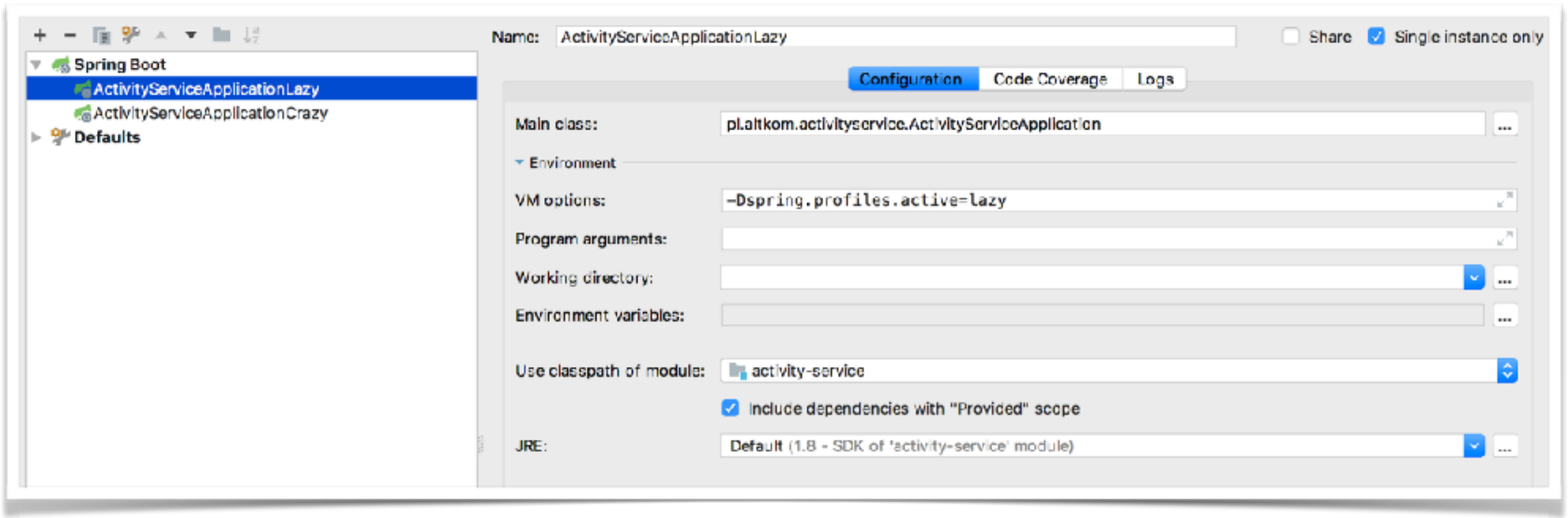
plik konfiguracyjny



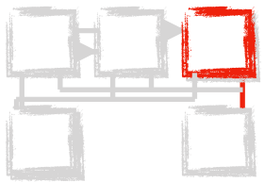
Spring Cloud



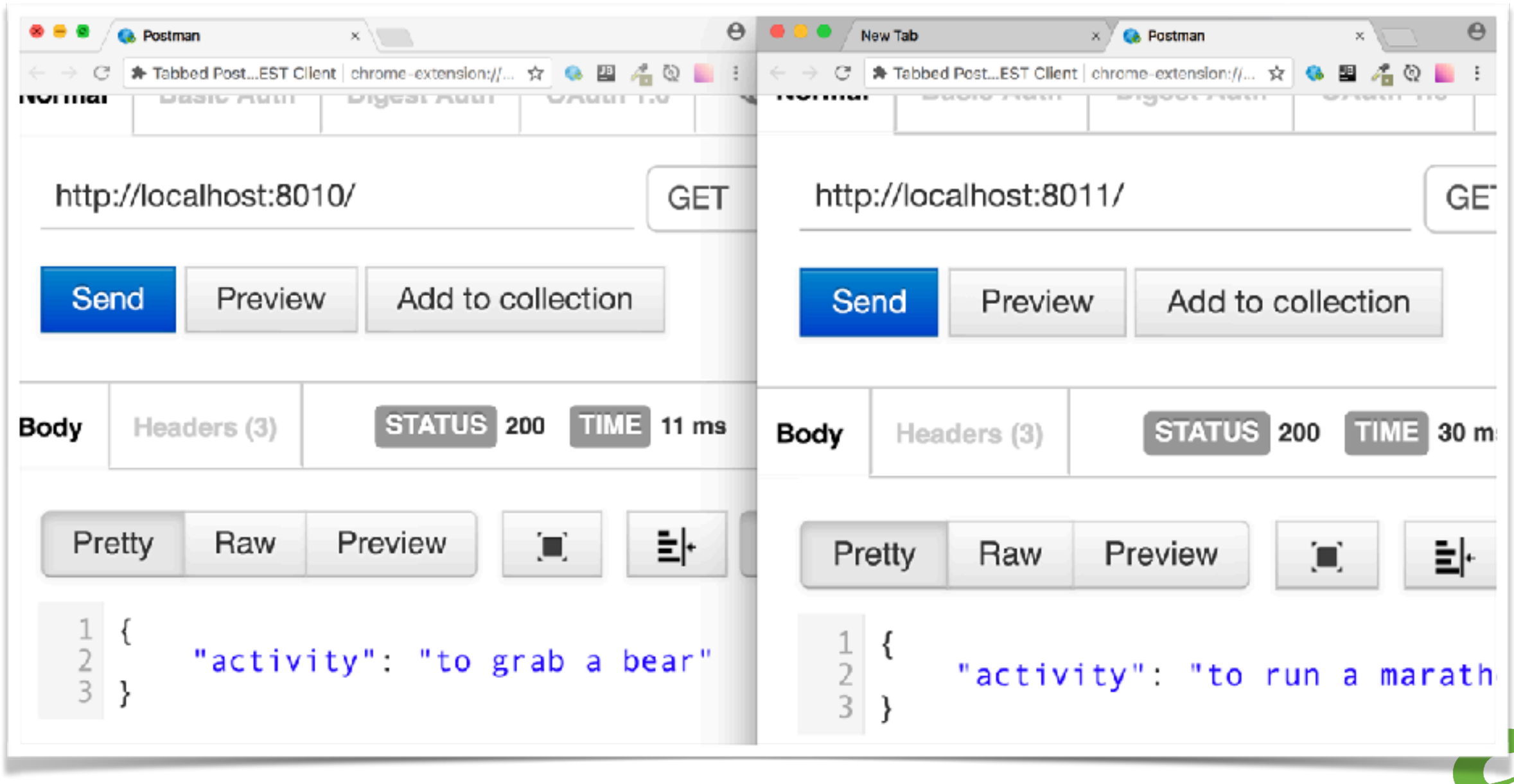
uruchamianie z profilem



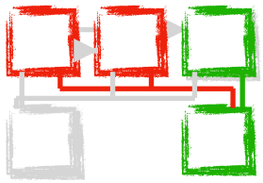
Spring Cloud



test obu instancji



Spring Cloud

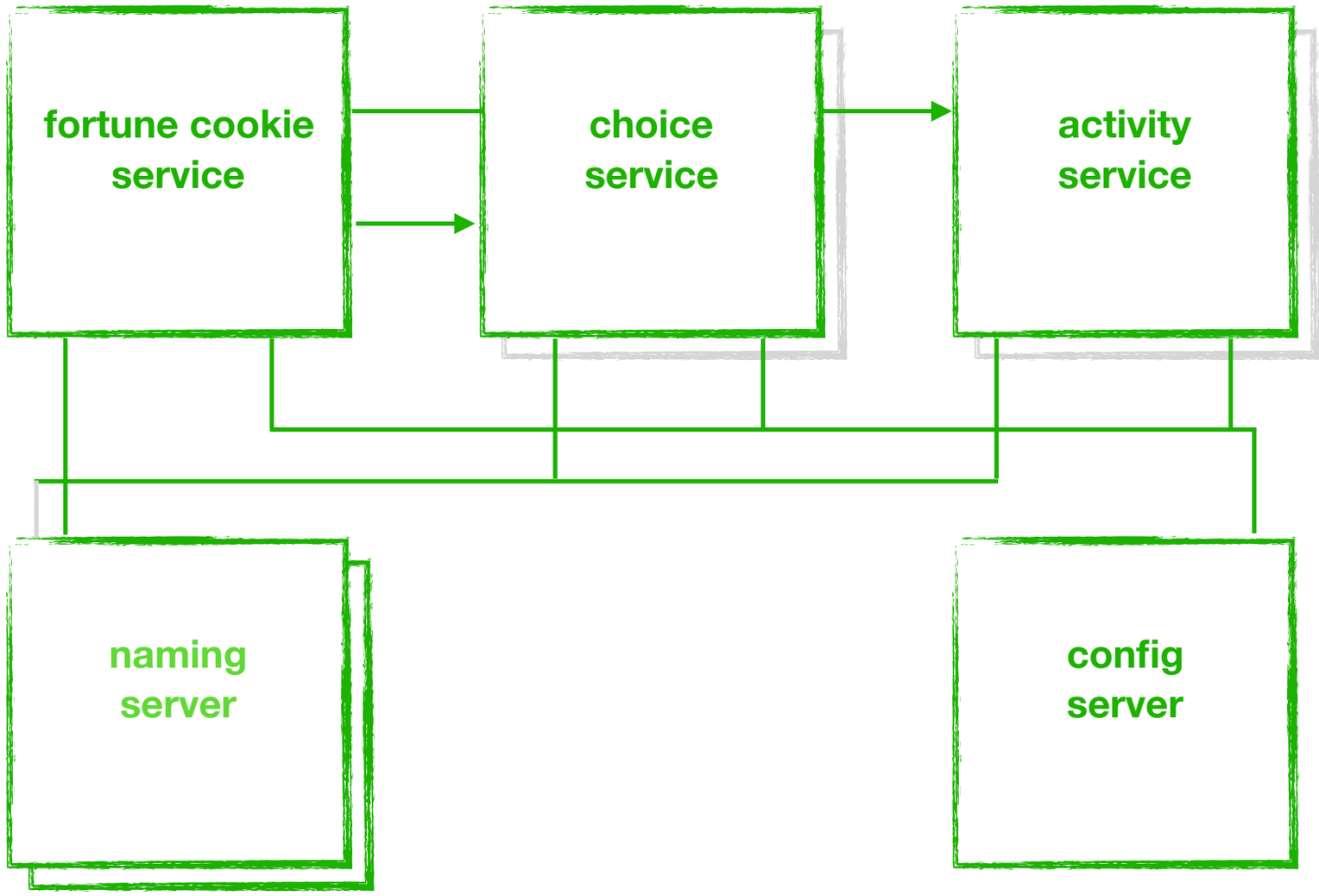


zadanie

- utworzenie repo Git, pierwszy commit
- komponent Config Server
- refactor / podłączenie activity-service do Config Server
  - profil lazy
  - profil crazy
- **zadanie - podłączenie pozostałych serwisów**
  - **analogicznie jak activity-service**
    - **decision-service (profil polite, rough)**
    - **fortune-cookie-service (bez profilu)**

Spring Cloud

Serwisy i Config Server - podsumowanie

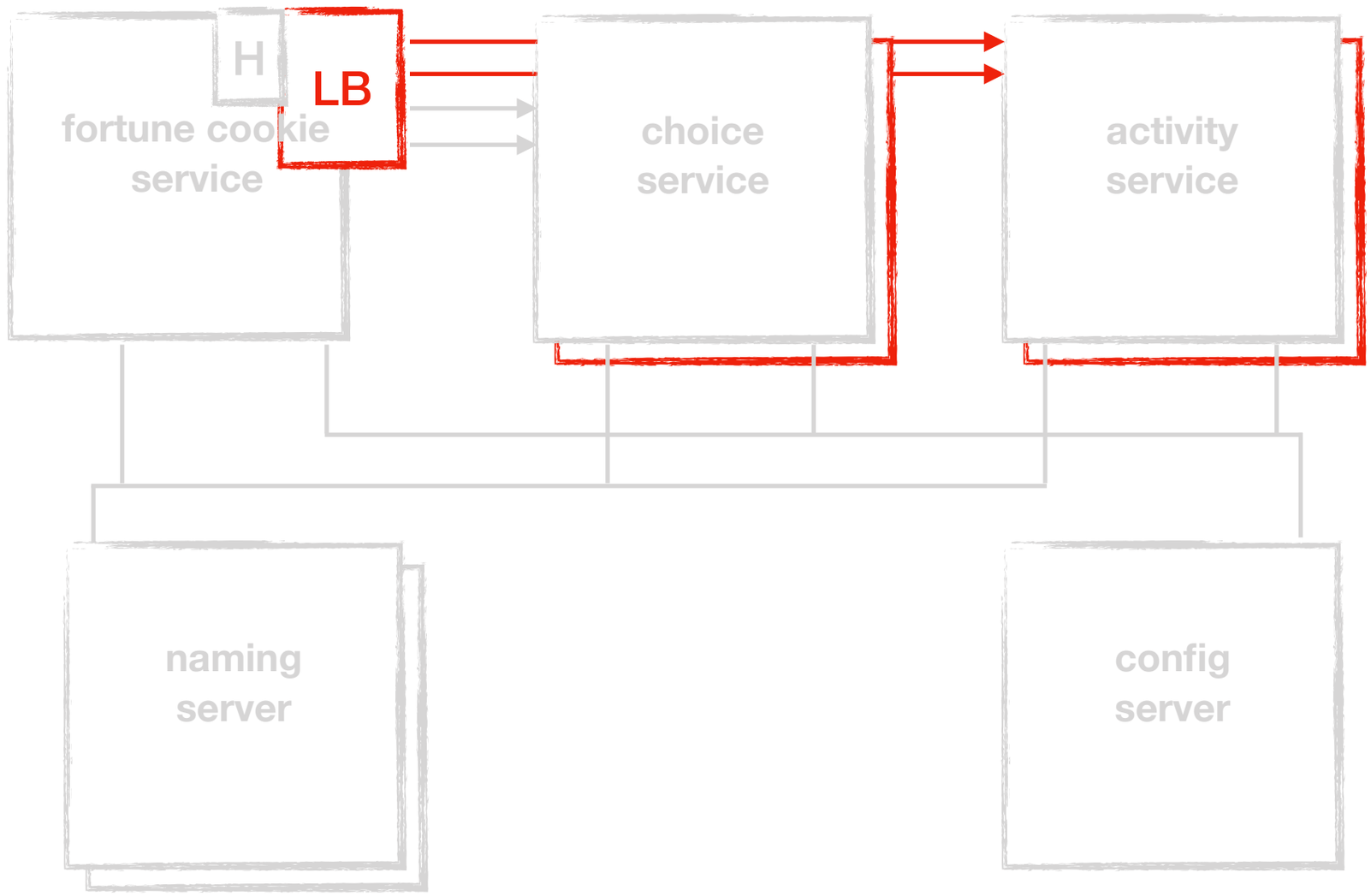


Spring Cloud

III Ribbon

- refaktor fortune-cookie-service
  - użycie Ribbon
- test
  - uruchomienie podwójnych instancji
    - decision-service
      - profile polite i rough
    - activity-service
      - profile lazy i crazy
- w logach fortune-cookie-service
  - powinno być widać instancje w current list of Servers
- requesty do fortune-cookie-service
  - powinny zwracać mix 4 instancji (w configu różne dane dla każdej)

III Ribbon

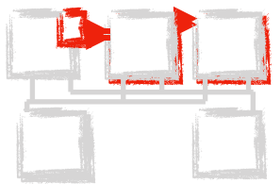


Spring Cloud

III Ribbon

- **refaktor fortune-cookie-service**
  - **użycie Ribbon**
- test
  - uruchomienie podwójnych instancji
    - decision-service
      - profile polite i rough
    - activity-service
      - profile lazy i crazy
- requesty do fortune-cookie-service
- powinny zwracaź mix 4 instancji (w configu różne dane dla każdej)
- w logach fortune-cookie-service c
- powinno być widać instancje w current list of Servers

Spring Cloud



refaktor - kontroler i główna

```
    @Autowired
    @Qualifier("ribbonRestTemplate")
    private RestTemplate ribbonRestTemplate;

    @GetMapping("/fortune")
    @ResponseBody // automatic conversion to JSON
    public Fortune fortune() {
        String fortune =
                getDataFromService("DECISION-SERVICE") + " " + getDataFromService("ACTIVITY-SERVICE");
        return new Fortune(fortune);
    }

    public String getDataFromService(String service) {
        String responseInJSON = (ribbonRestTemplate).getForObject( url: "http://" + service, String.class);
        return convertJSONResponseToSentence(responseInJSON);
    }
```

FortuneController › ribbonRestTemplate

CookieServiceApplication.java

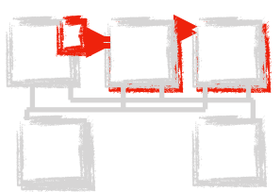
```
import ...

@SpringBootApplication
@EnableDiscoveryClient
public class FortuneCookieServiceApplication {

    public static void main(String[] args) { SpringApplication.run(FortuneCookieServiceApplication.class, args); }

    @Bean
    @LoadBalanced
    RestTemplate ribbonRestTemplate() {
        return new RestTemplate();
    }
}
```


Spring Cloud

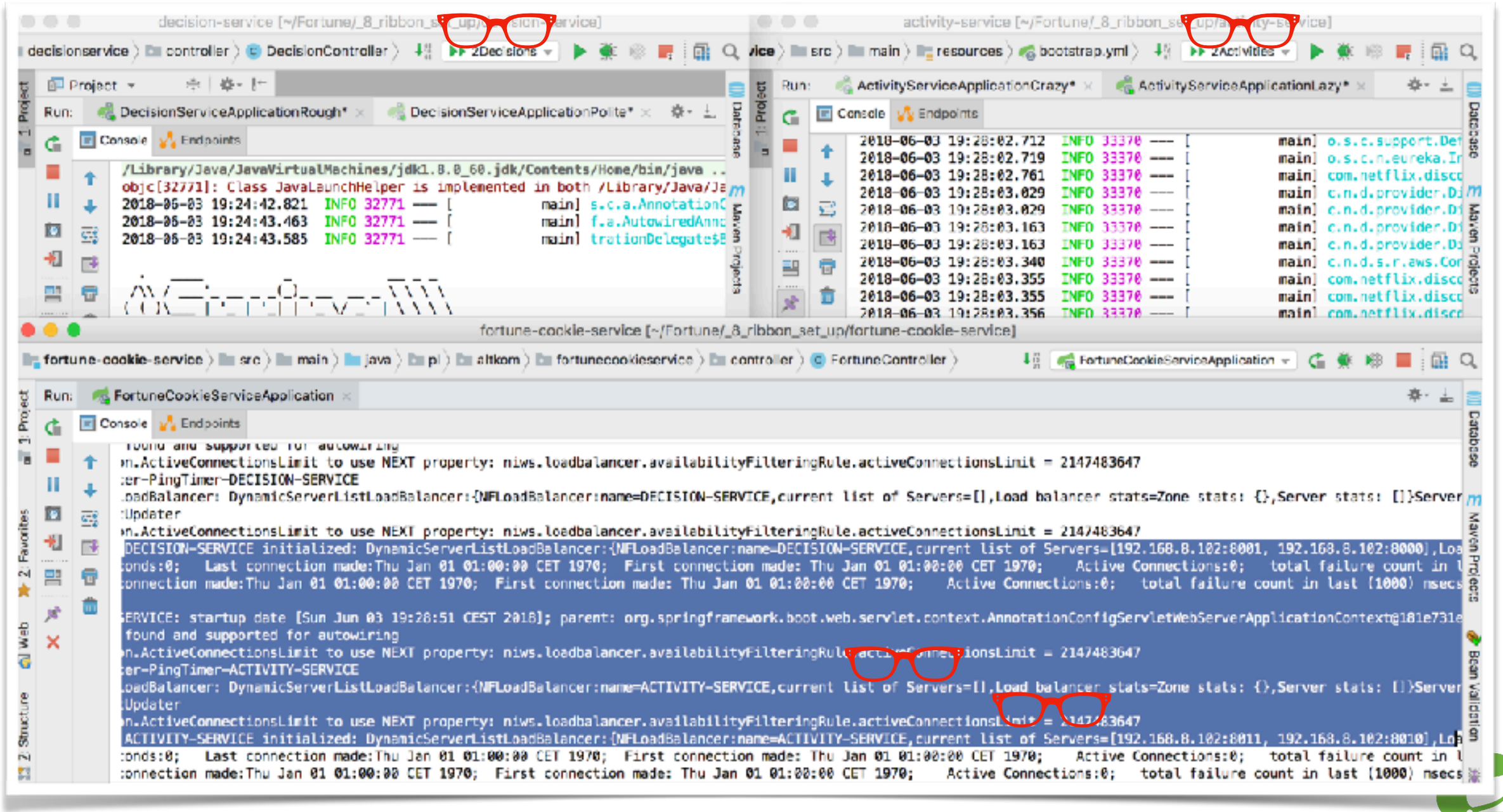
III Ribbon

- refaktor fortune-cookie-service
  - użycie Ribbon
- **test**
  - **uruchomienie podwójnych instancji**
    - decision-service
      - profile polite i rough
    - activity-service
      - profile lazy i crazy
  - **w logach** fortune-cookie-service
    - powinno być widać instancje w **current list of Servers**
- **requesty** do fortune-cookie-service
  - **powinny zwracać mix 4** instancji (w configu różne dane dla każdej)

Spring Cloud

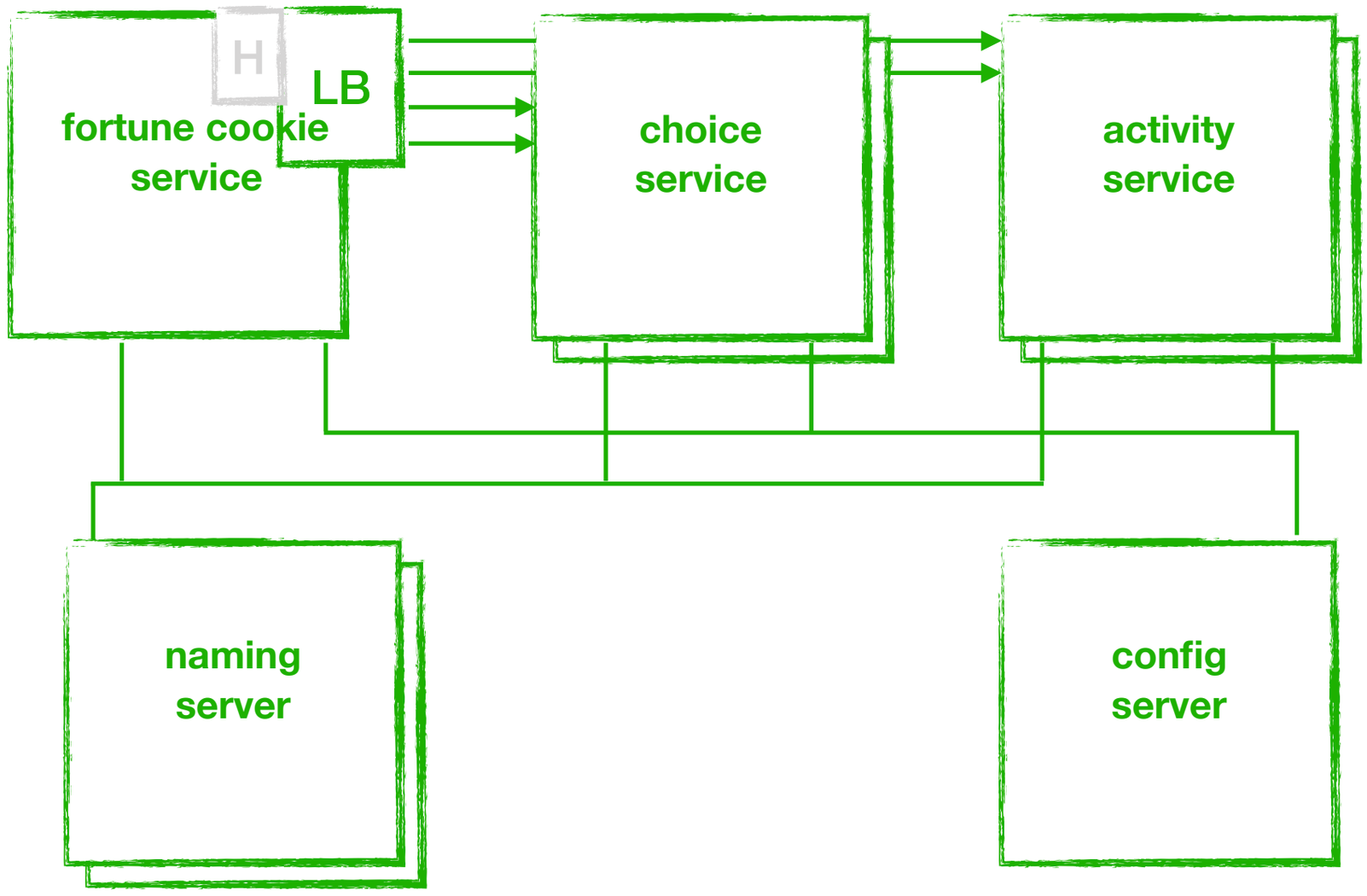


test



Spring
Cloud

Ribbon - podsumowanie



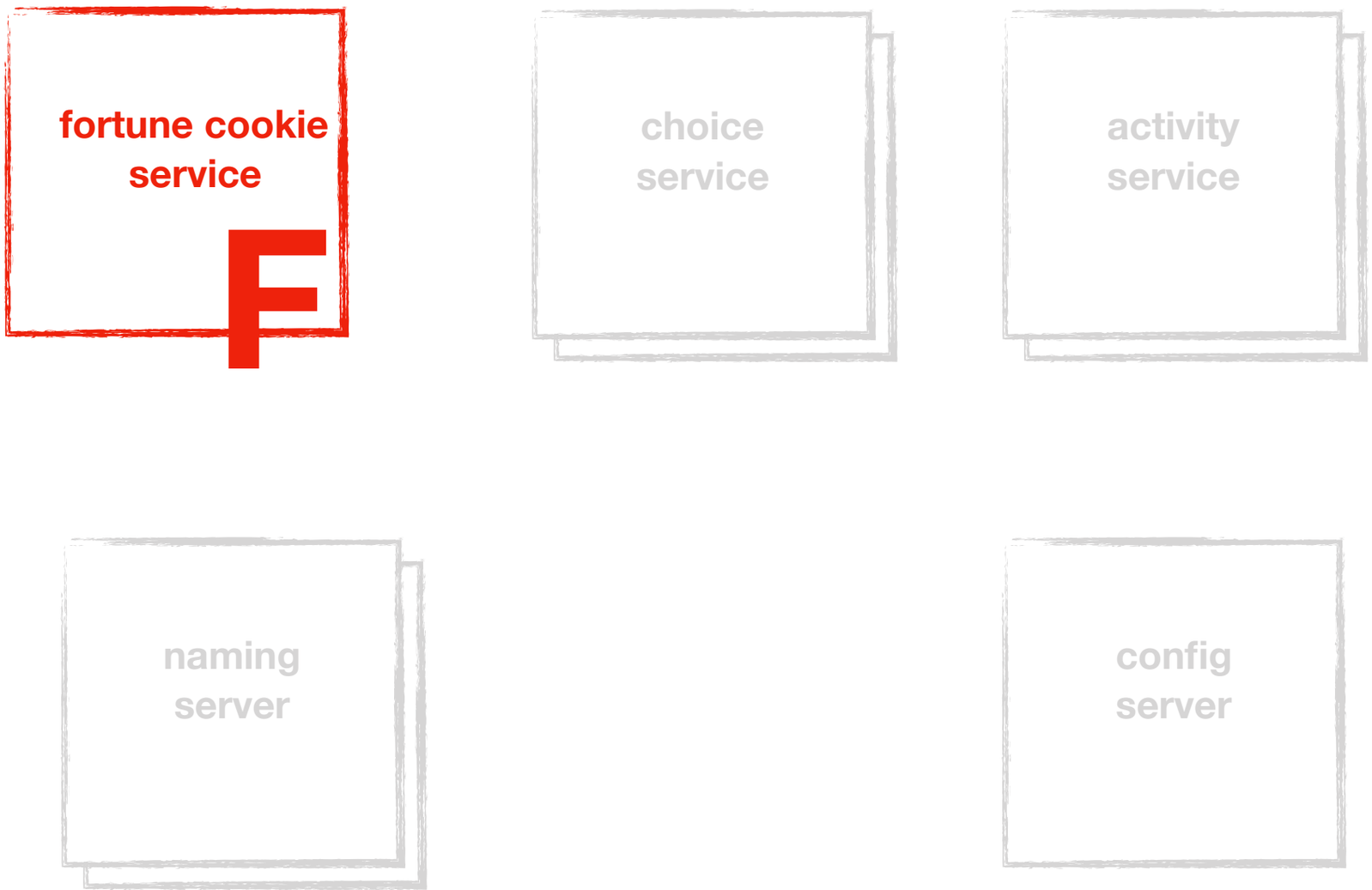
Spring
Cloud

IV Feign

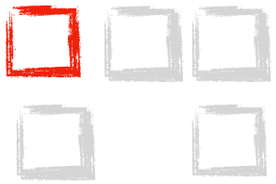
- refaktor fortune-cookie-service
  - zależności maven
  - dodanie interfejsów
  - @EnableFeignClients
  - użycie w kontrolerze

Spring Cloud

IV Feign



Spring Cloud

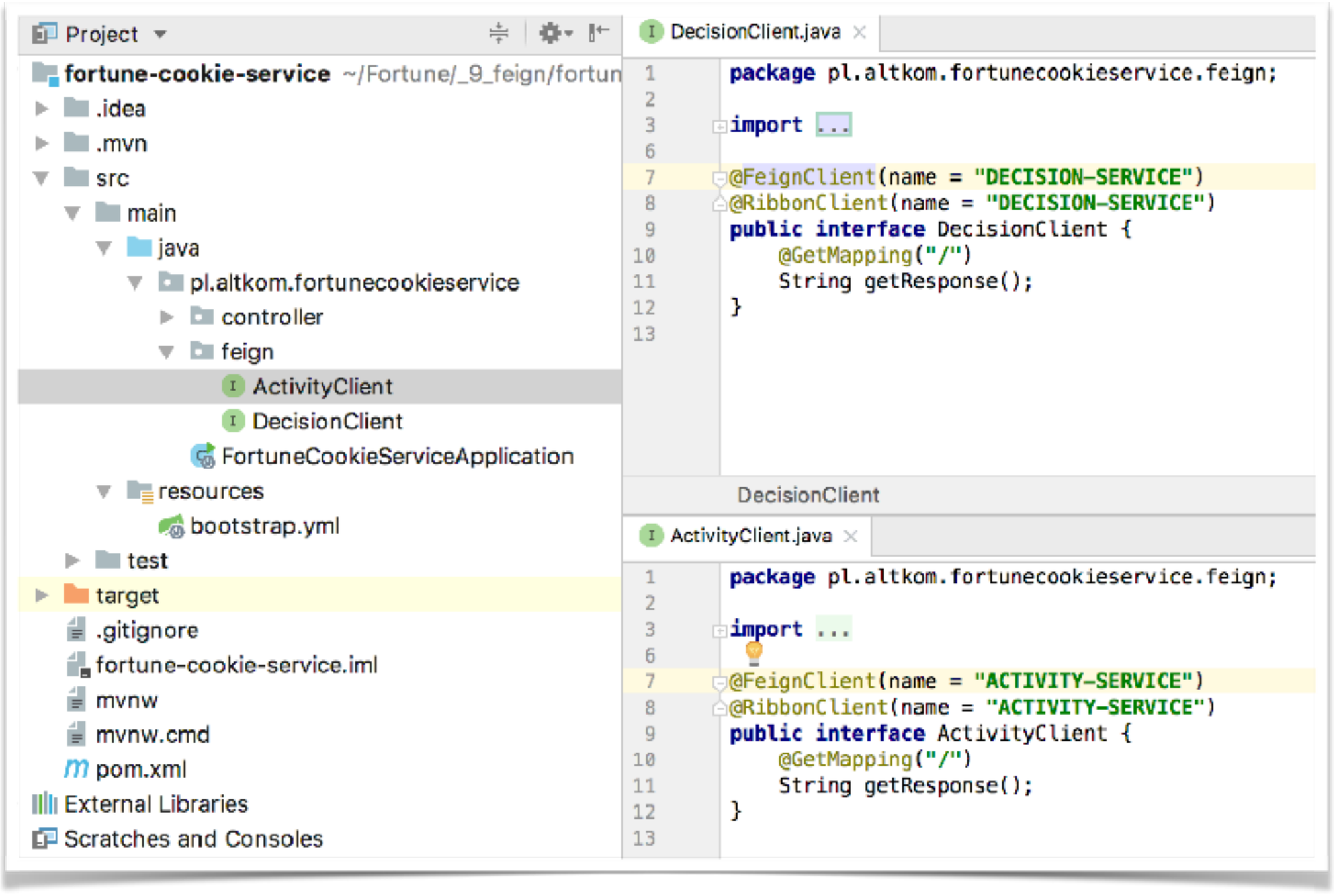


zależności maven

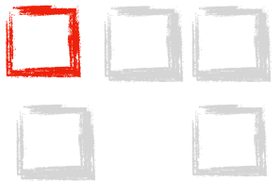
```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-openfeign</artifactId>
</dependency>
```


Spring Cloud

dodanie interfejsów



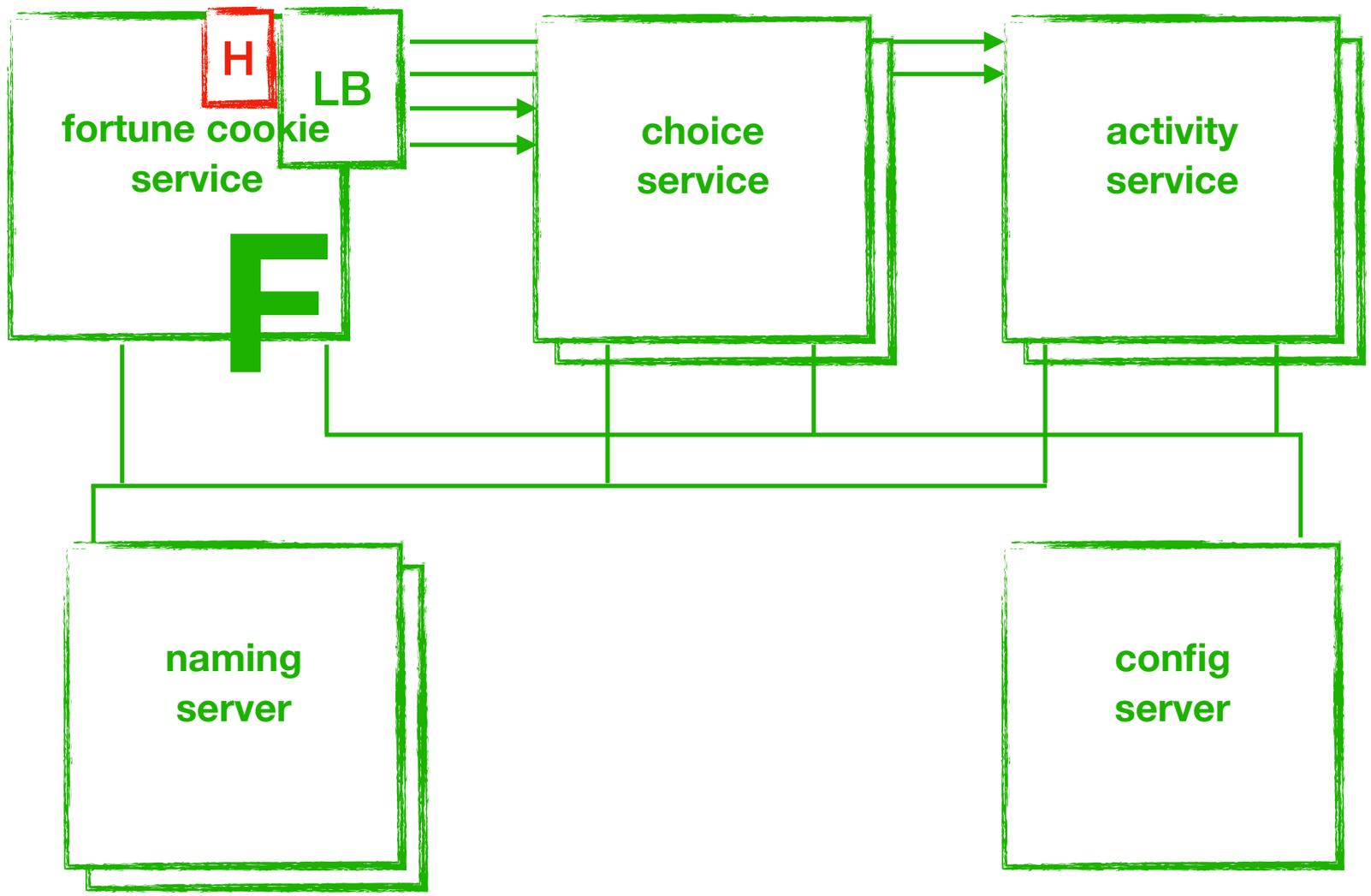
Spring Cloud



użycie w kontrolerze

```
@RestController
public class FortuneController {

    @Autowired
    private ActivityClient activityClient;
    @Autowired
    private DecisionClient decisionClient;

    @GetMapping("/fortune")
    @ResponseBody
    public Fortune fortune() {
        String fortune =
                convertJSONResponseToSentence(decisionClient.getResponse()) + " " +
                        convertJSONResponseToSentence(activityClient.getResponse());
        return new Fortune(fortune);
    }

    private String convertJSONResponseToSentence(String responseInJSON) {
        JSONArray dataFromService = JsonPath.read(responseInJSON, jsonPath: "$..*");
        return (String) dataFromService.get(0);
    }
```

V Hystrix



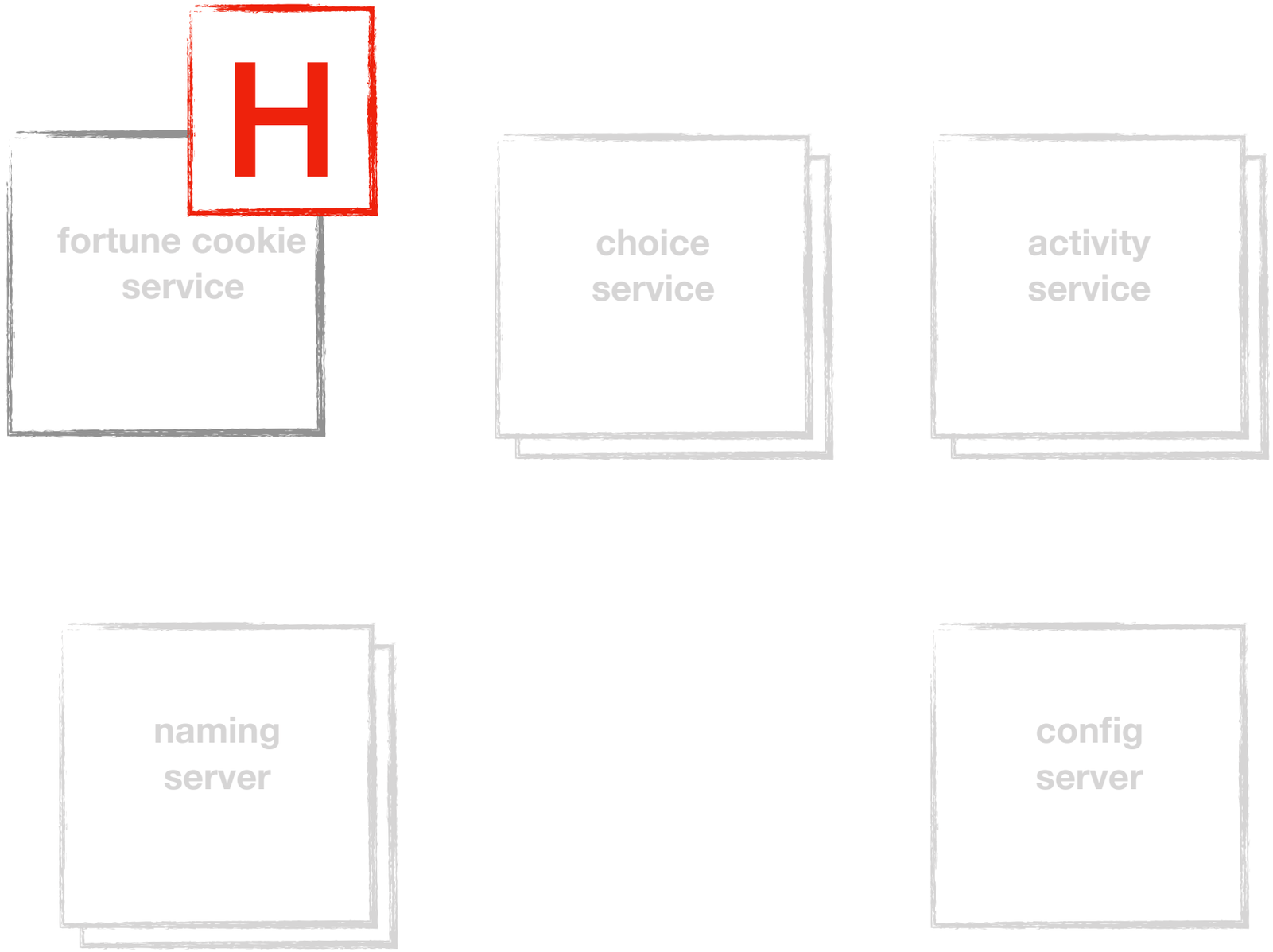
Spring
Cloud

V Hystrix

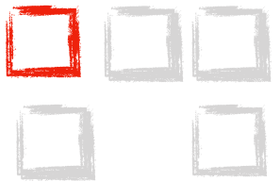
- refaktor fortune-cookie-service
  - dodaj zależność w pom
  - dodaj adnotacjw w głównej
  - w adnotacjach @FeignClient ustaw fallback’i
  - zaimplementuj fallback’i

Spring Cloud

V Hystrix



Spring Cloud

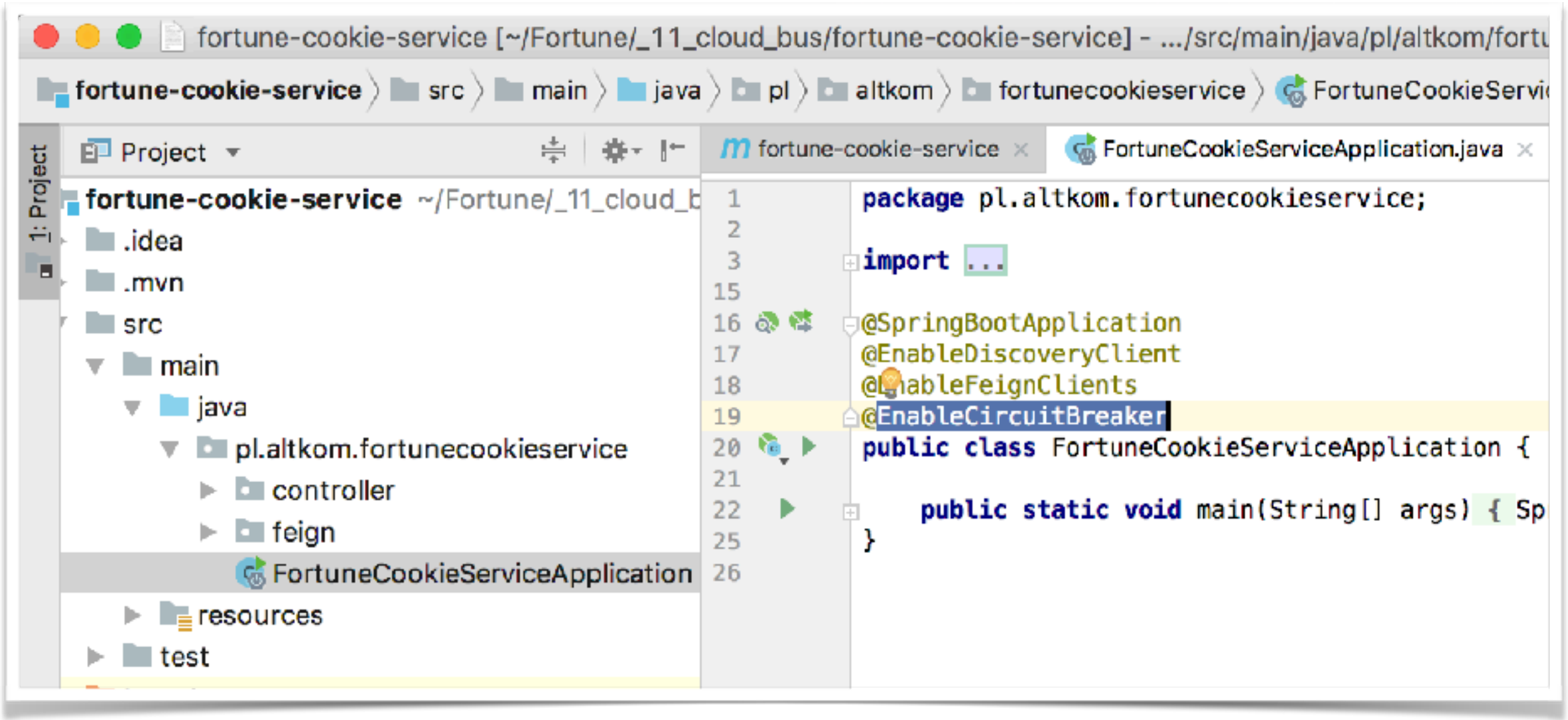


dodaj zależność w pom

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-hystrix</artifactId>
    <version>1.4.4.RELEASE</version>
</dependency>
```

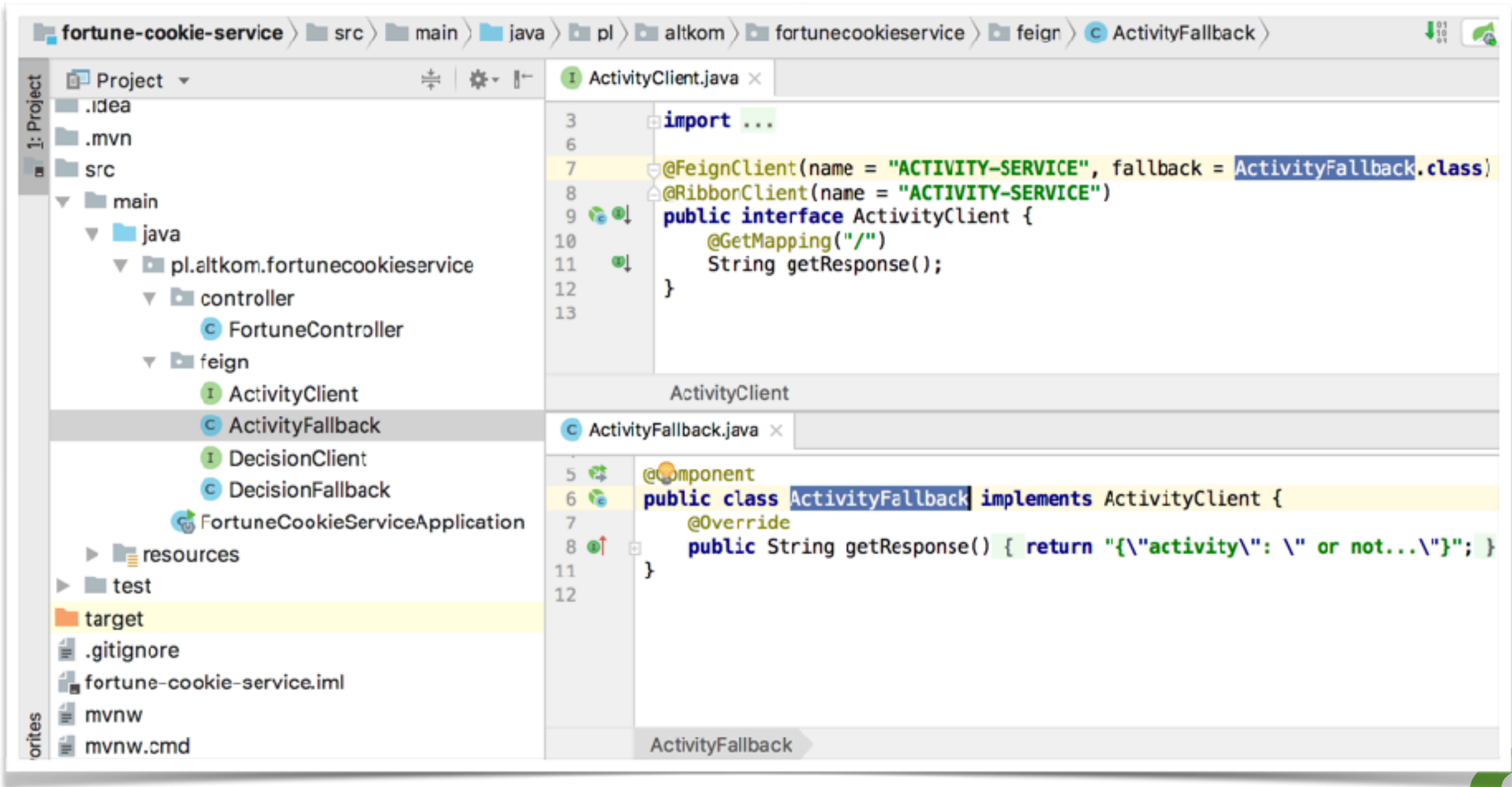

Spring Cloud

dodaj adnotację w głównej

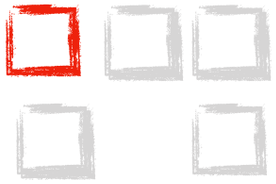


Spring Cloud

fallback: activity-service



Spring Cloud

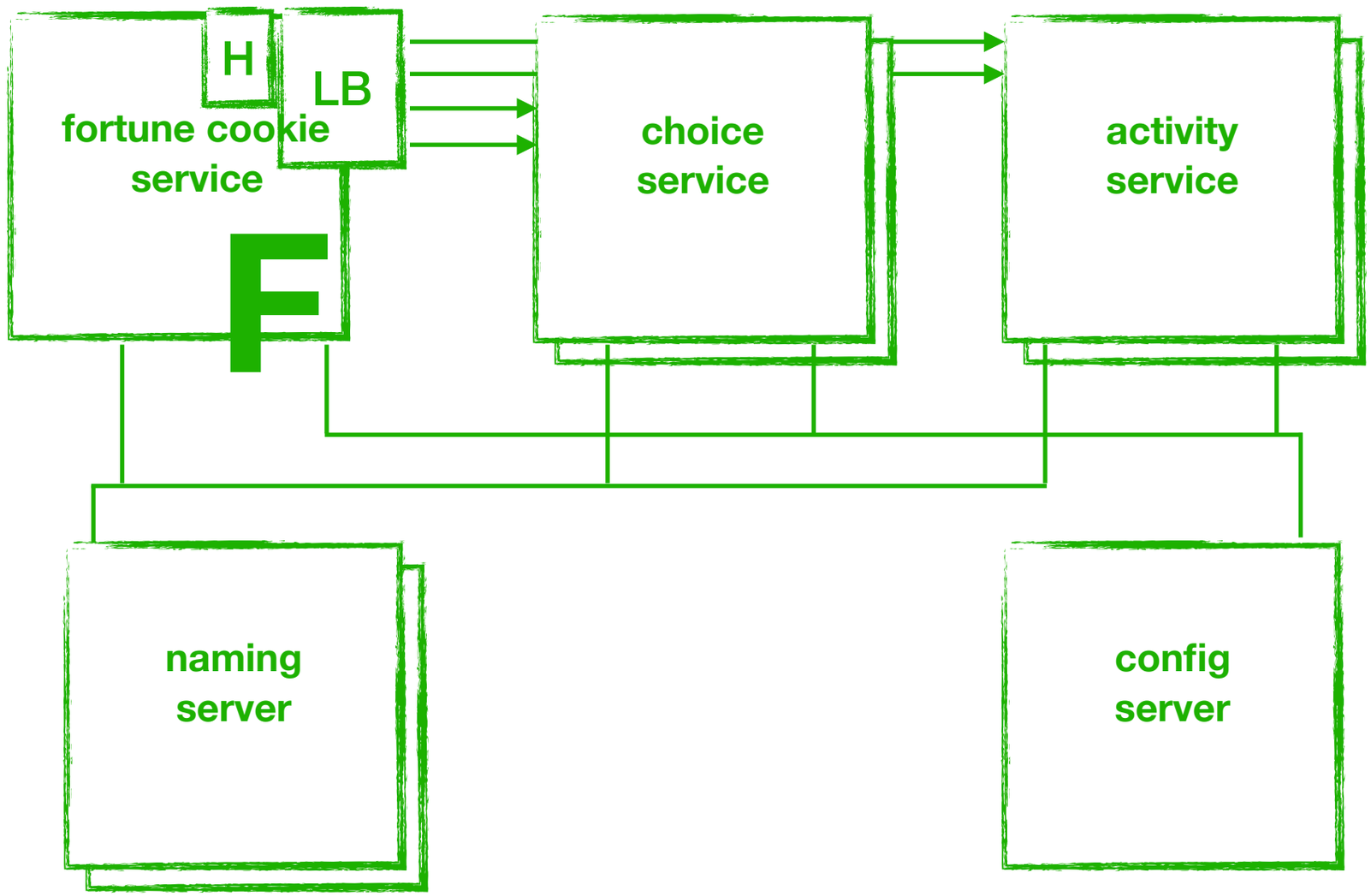


zadania

- implementacja fallback dla DecisionClient
- test
  - po wyłączeniu activity-service
  - po wyłączeniu decision-service

Spring Cloud

Hystrix - podsumowanie



Spring Cloud