



THE
DEVELOPER'S
CONFERENCE

PREPARANDO NOSSAS APLICAÇÕES PARA FALHAREM COM FEATURE TOGGLE E CONFIGURAÇÕES DISTRIBUÍDAS

Trilha Java Enterprise

Emmanuel Neri

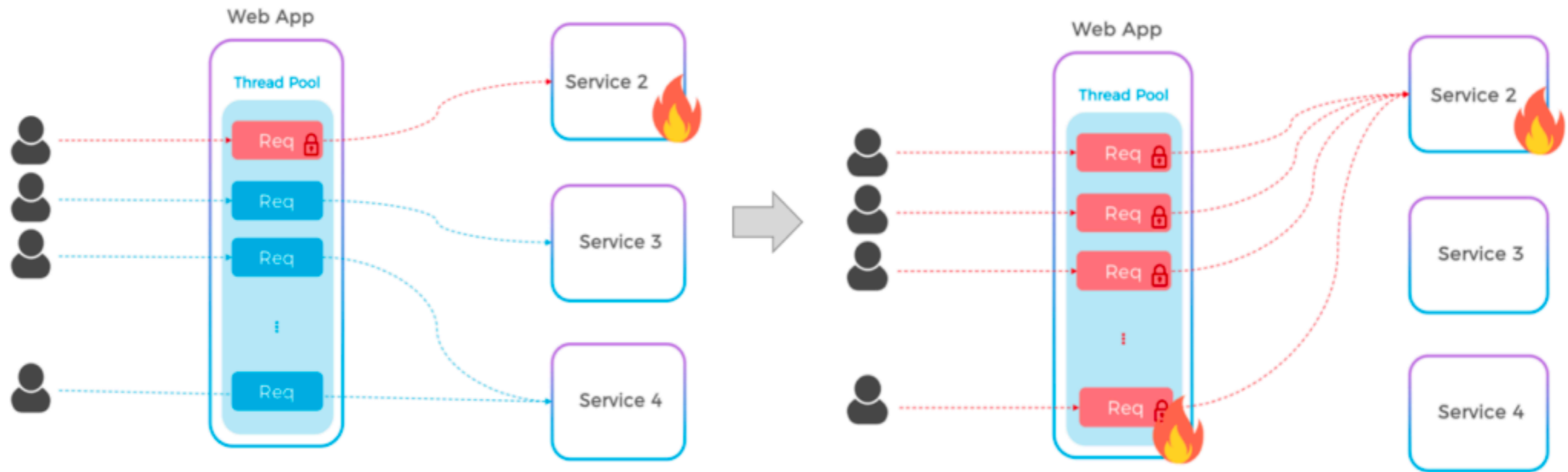
@emmanuelnerii



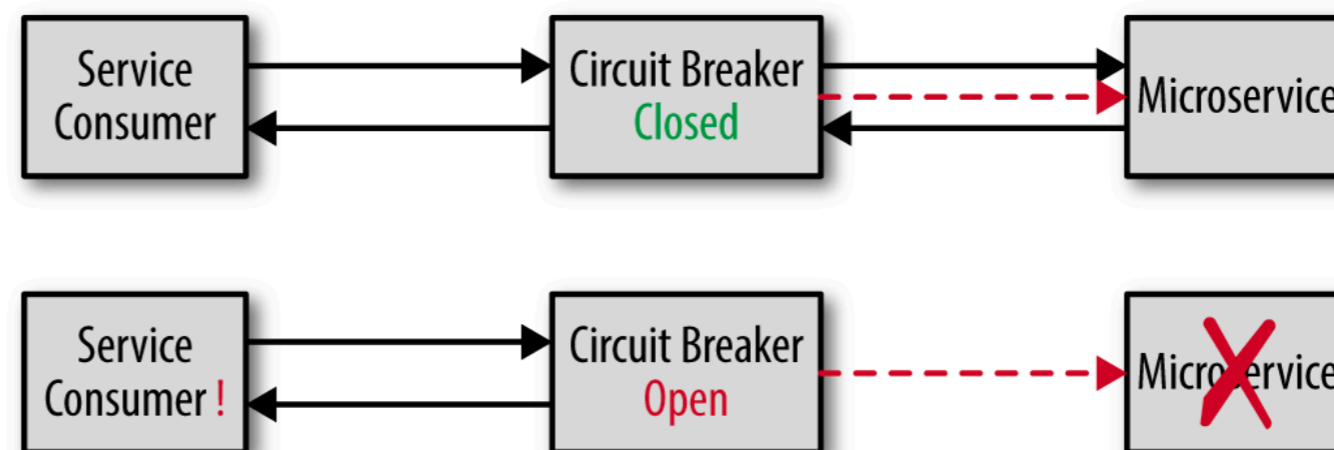
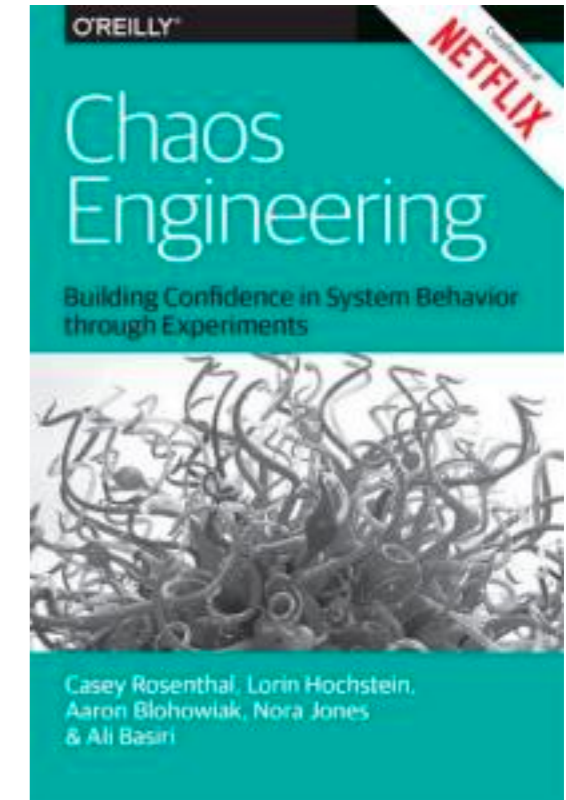
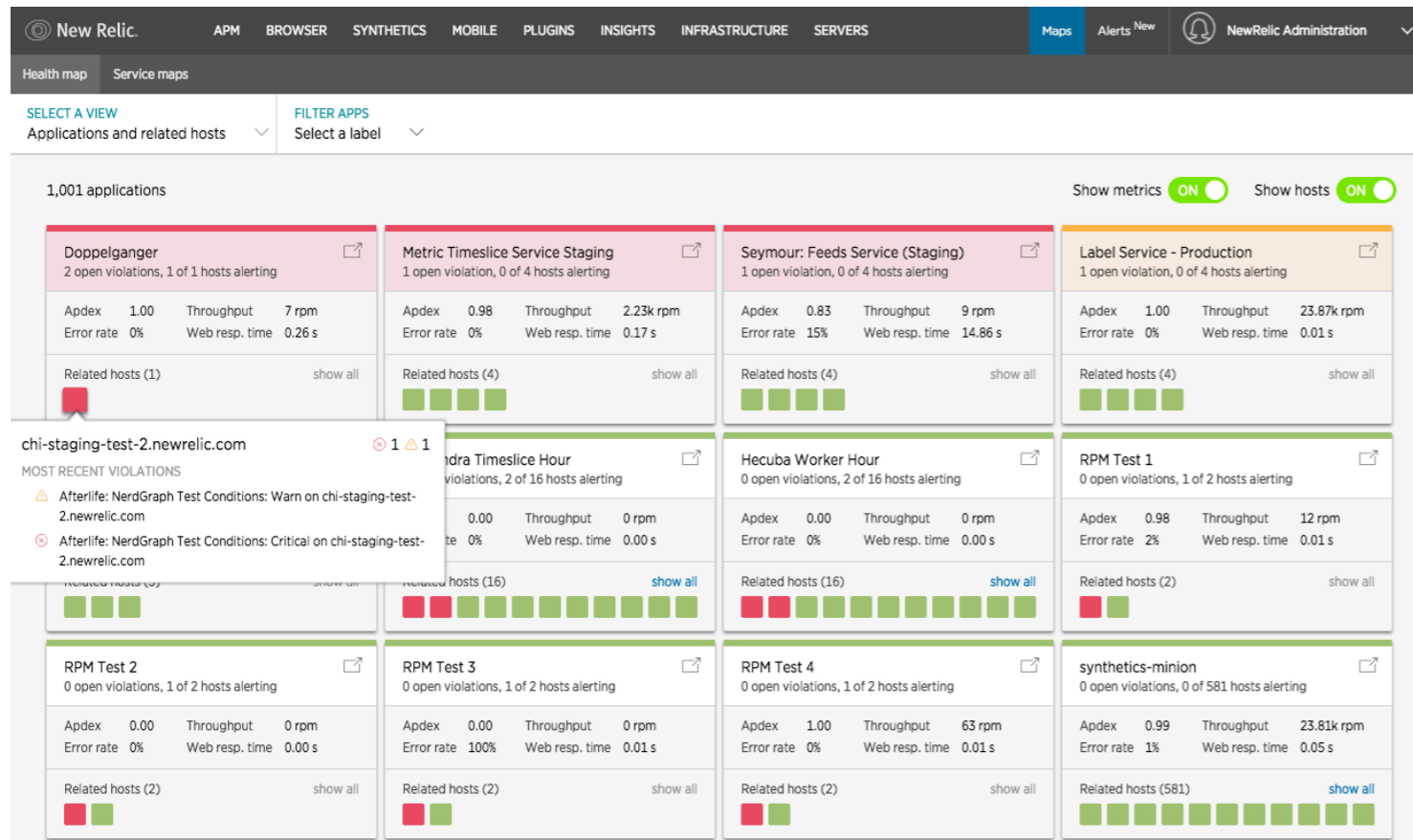


- ▶ Mestre em Desenvolvimento de Tecnologia
- ▶ Desenvolvedor (Java) desde 2010
- ▶ Software Developer na Jaya





POSSIBILIDADES





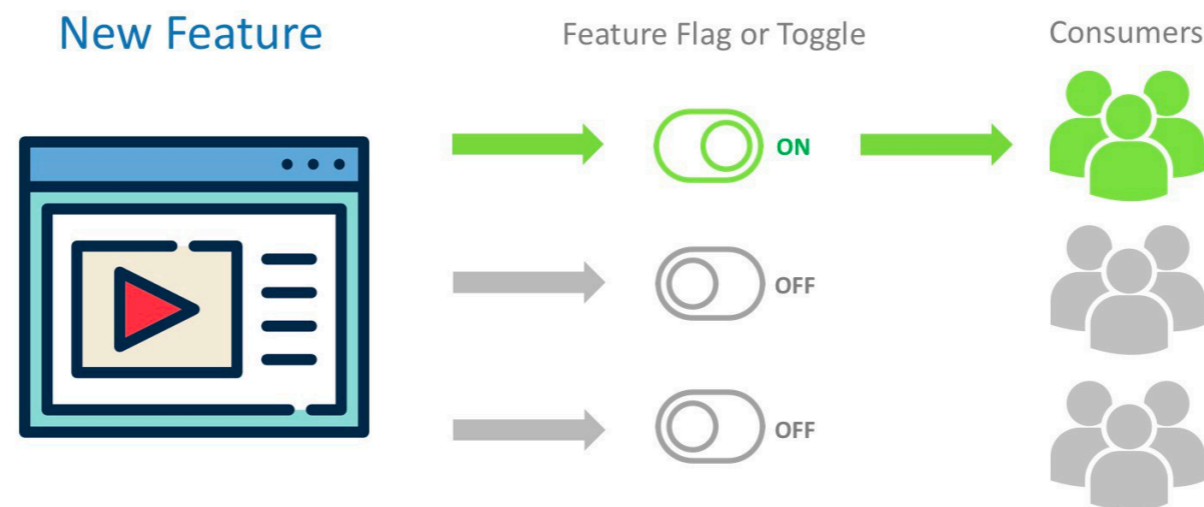
update Order set value = 0 where status in (:status)

valor.divide(quantidade, 2)




Feature Toggles (often also referred to as Feature Flags) are a powerful technique, allowing teams to modify system behavior without changing code. They fall into various usage categories, and it's important to take that categorization into account when implementing and managing toggles.

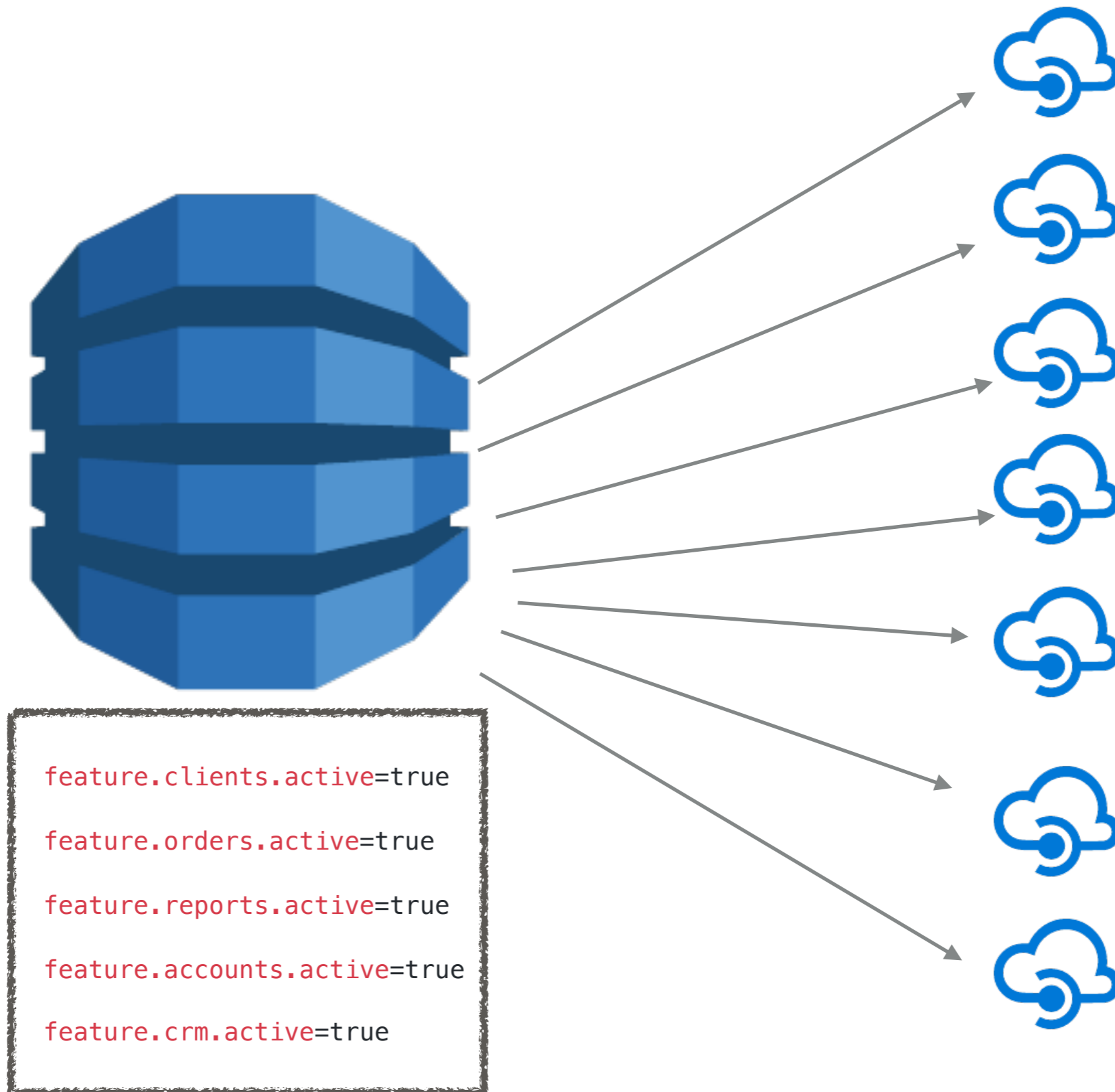
<https://martinfowler.com/articles/feature-toggles.html>

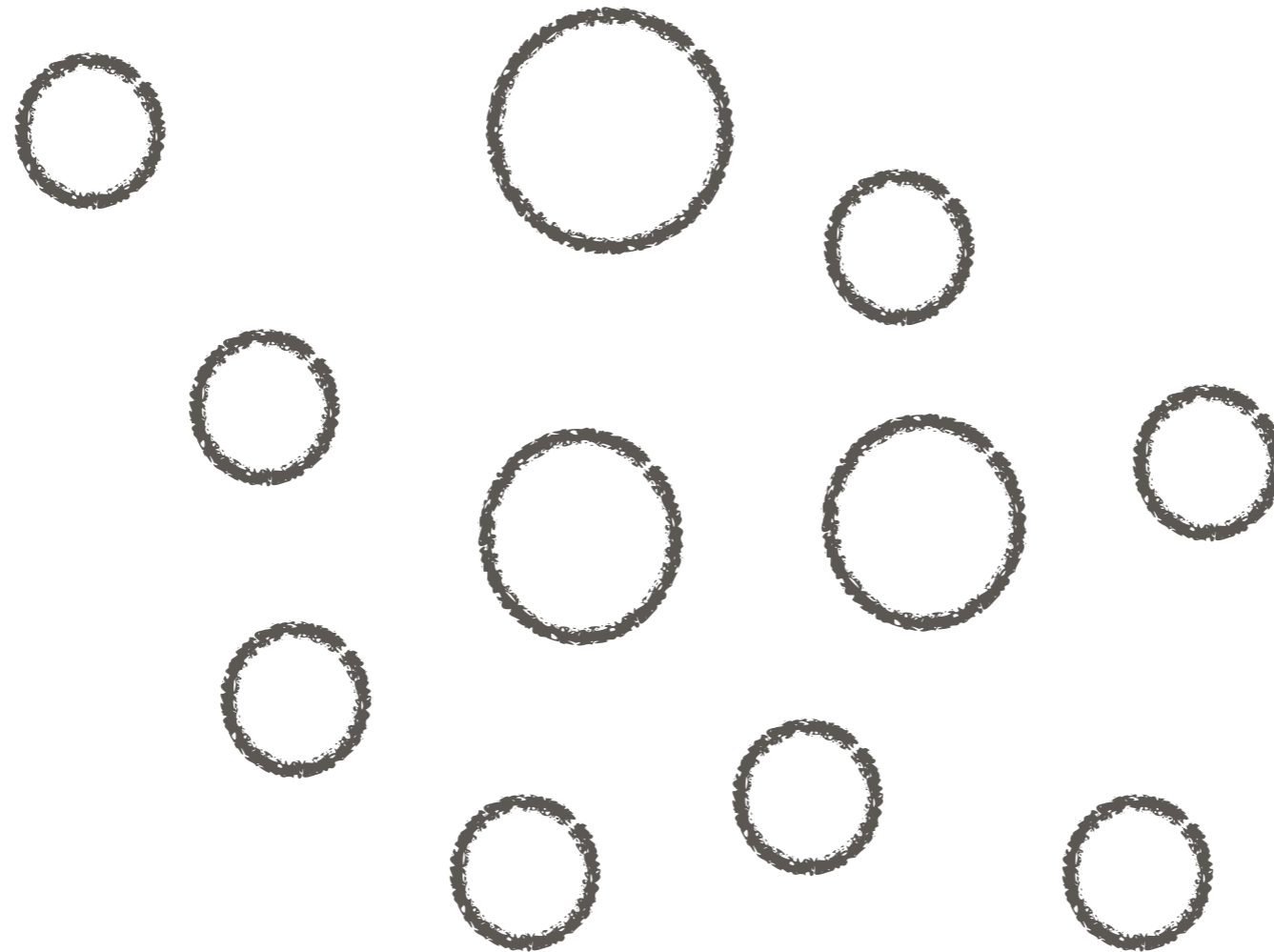
Feature Flags or Toggles



- ▶ Business Toggle
- ▶ Toggle Configuration
- ▶ Canary release
- ▶ A/B Testing
- ▶ Circuit Breaker
- ▶ Trunk based development
 - ▶ Continuous Delivery

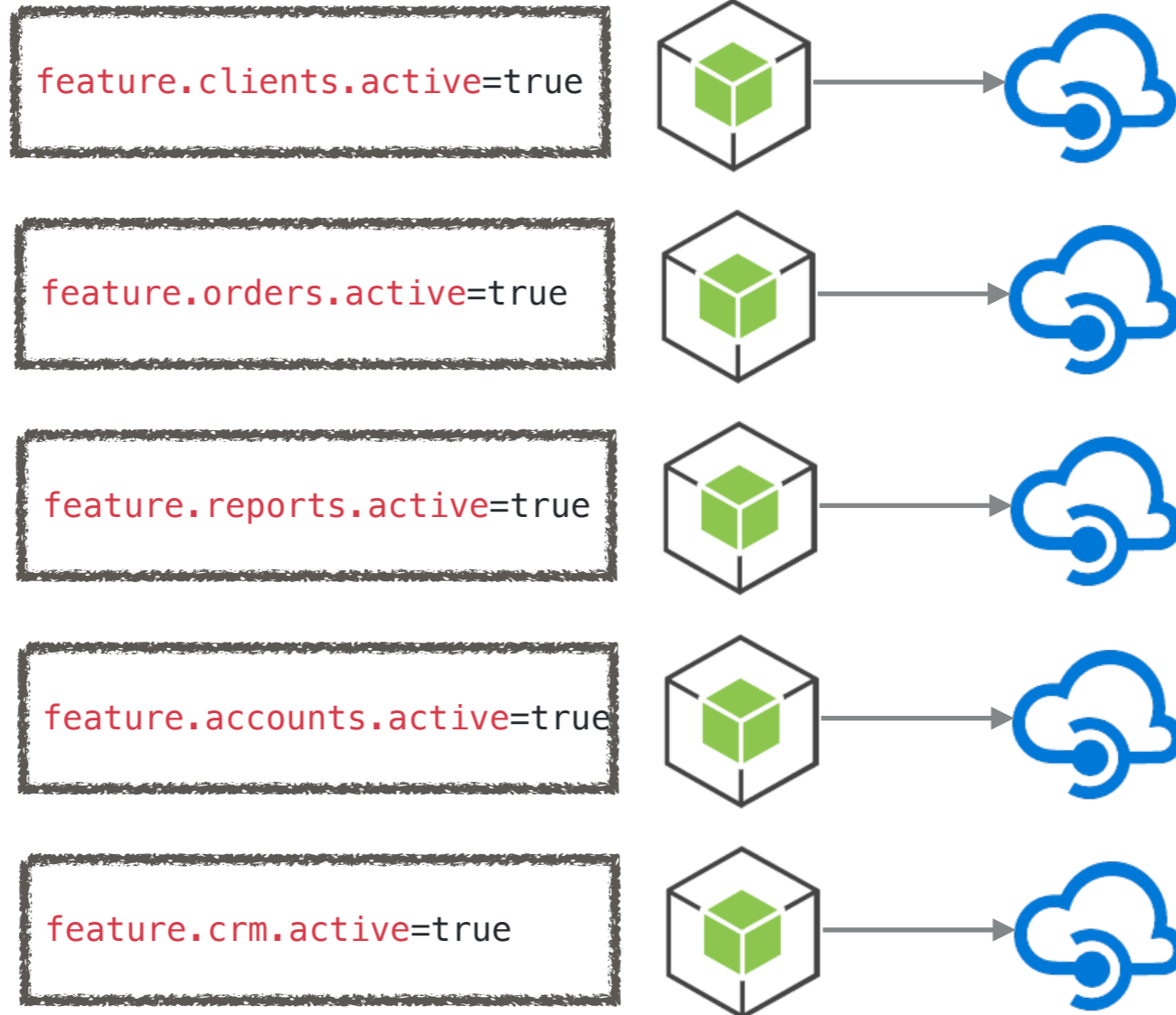
Feature	Status
X	
Y	
Z	



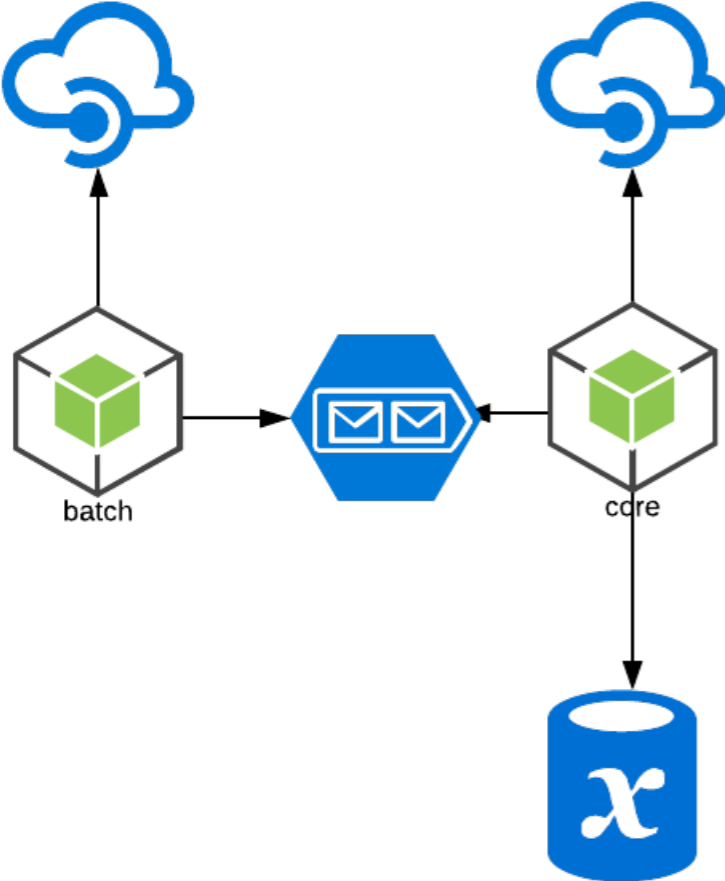


MICROSERVICES

MICROSERVICES



MICROSERVICES



instance 1



instance 2



instance 3

CONFIGURAÇÕES EM MICROSERVICES



```
spring.rabbitmq.host=localhost  
spring.rabbitmq.port=5672
```

```
spring.data.mongodb.host=localhost  
spring.data.mongodb.port=27017
```



```
spring.rabbitmq.host=localhost  
spring.rabbitmq.port=5672
```

```
spring.data.mongodb.host=localhost  
spring.data.mongodb.port=27017
```



```
spring.rabbitmq.host=localhost  
spring.rabbitmq.port=5672
```

```
spring.data.mongodb.host=localhost  
spring.data.mongodb.port=27017
```

CONFIGURAÇÕES EM MICROSERVICES



dev



homolog



prod



processor



customers



dev



homolog



prod



dev



homolog



prod



files



orders



dev

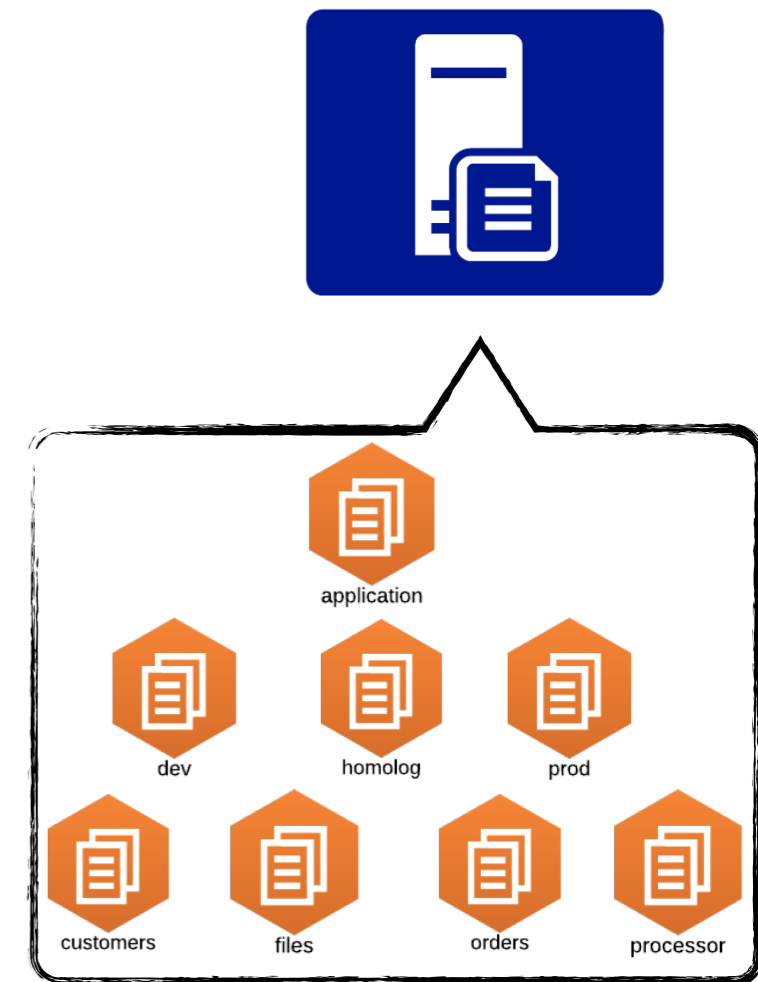
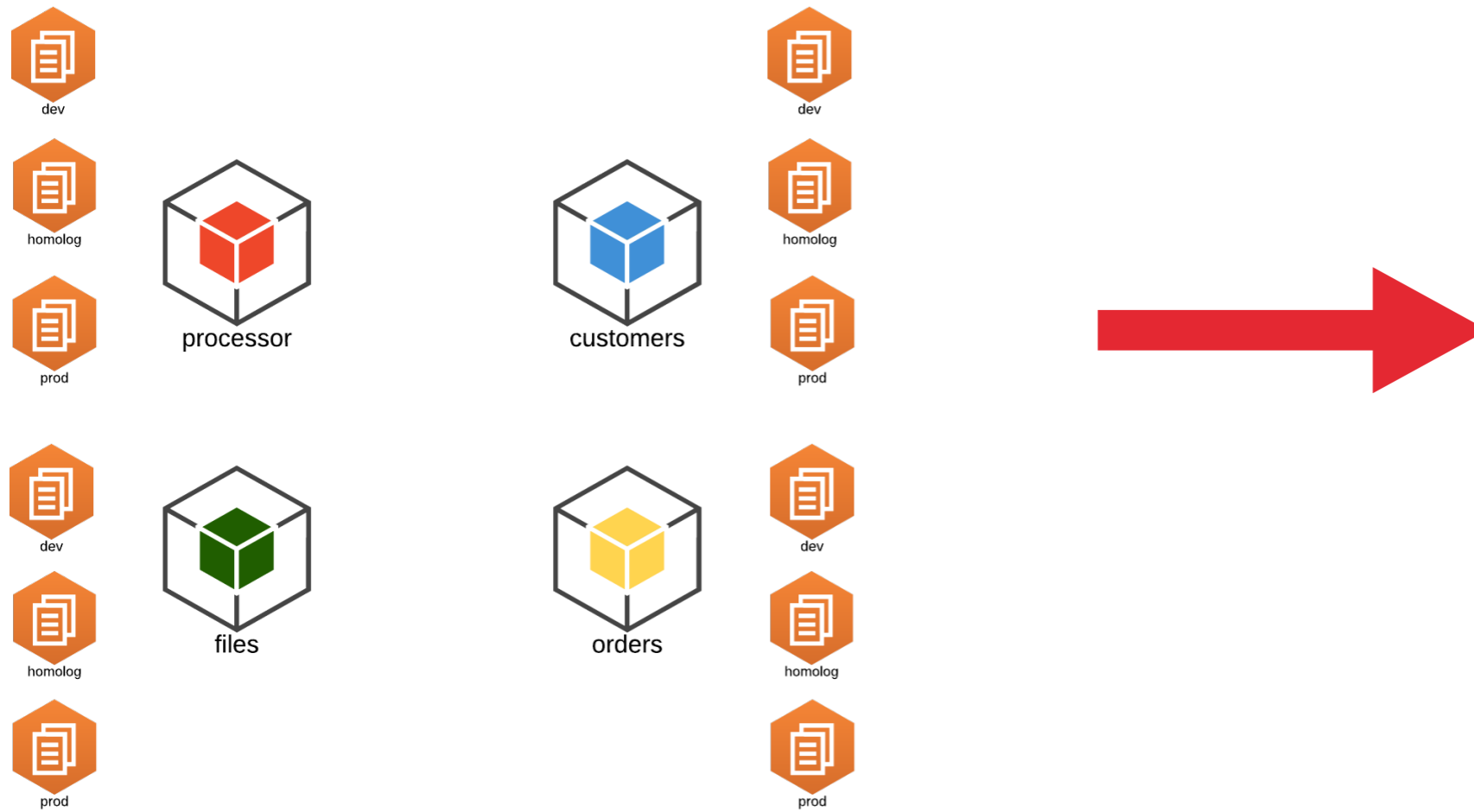


homolog



prod

CONFIGURAÇÕES DISTRIBUÍDAS



- ▶ Netflix Archaius
- ▶ Consul
- ▶ Spring Cloud Config
- ▶ Configmap

SPRING-CLOUD-CONFIG-SERVER



REPOSITÓRIO
CÓDIGO FONTE

FILE STORE

BANCO DE
DADOS



JDBC

SPRING-CLOUD-CONFIG-CLIENT

.JAR



SPRING CLOUD CONFIG



processor



orders



files



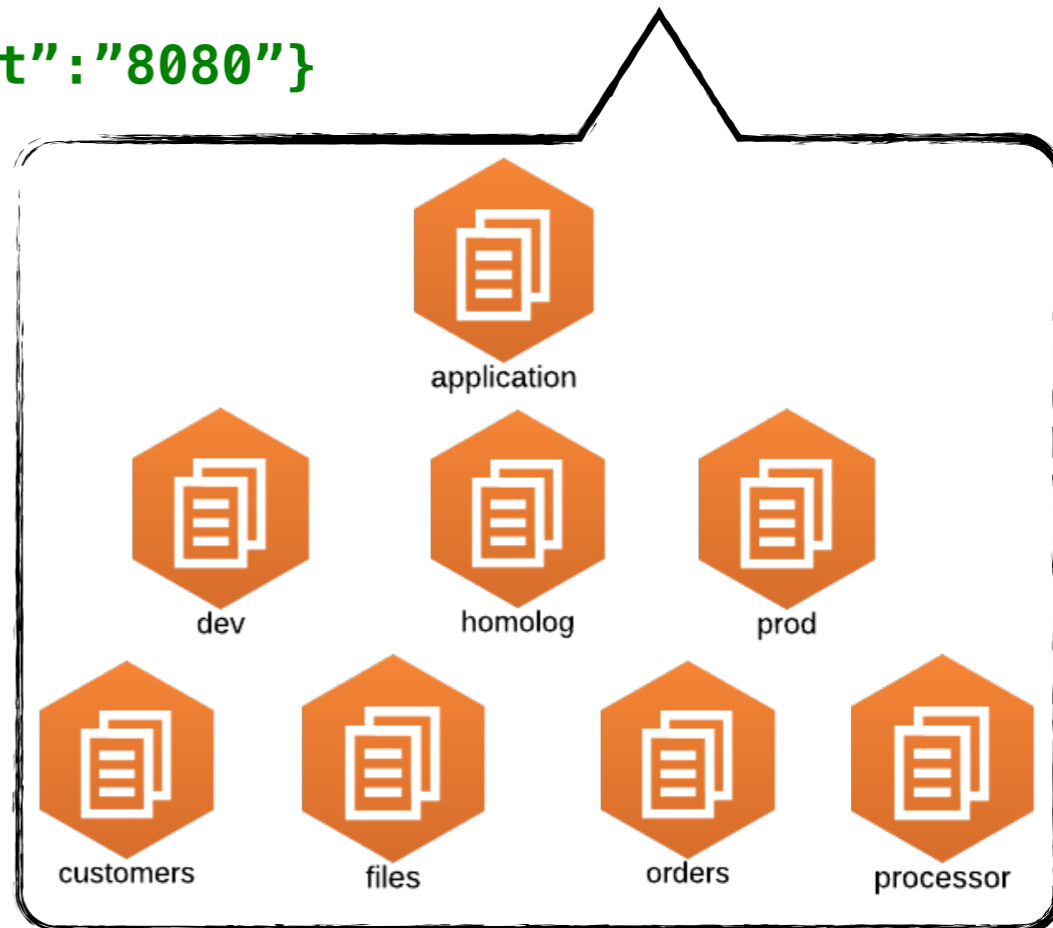
customers

name / profiles

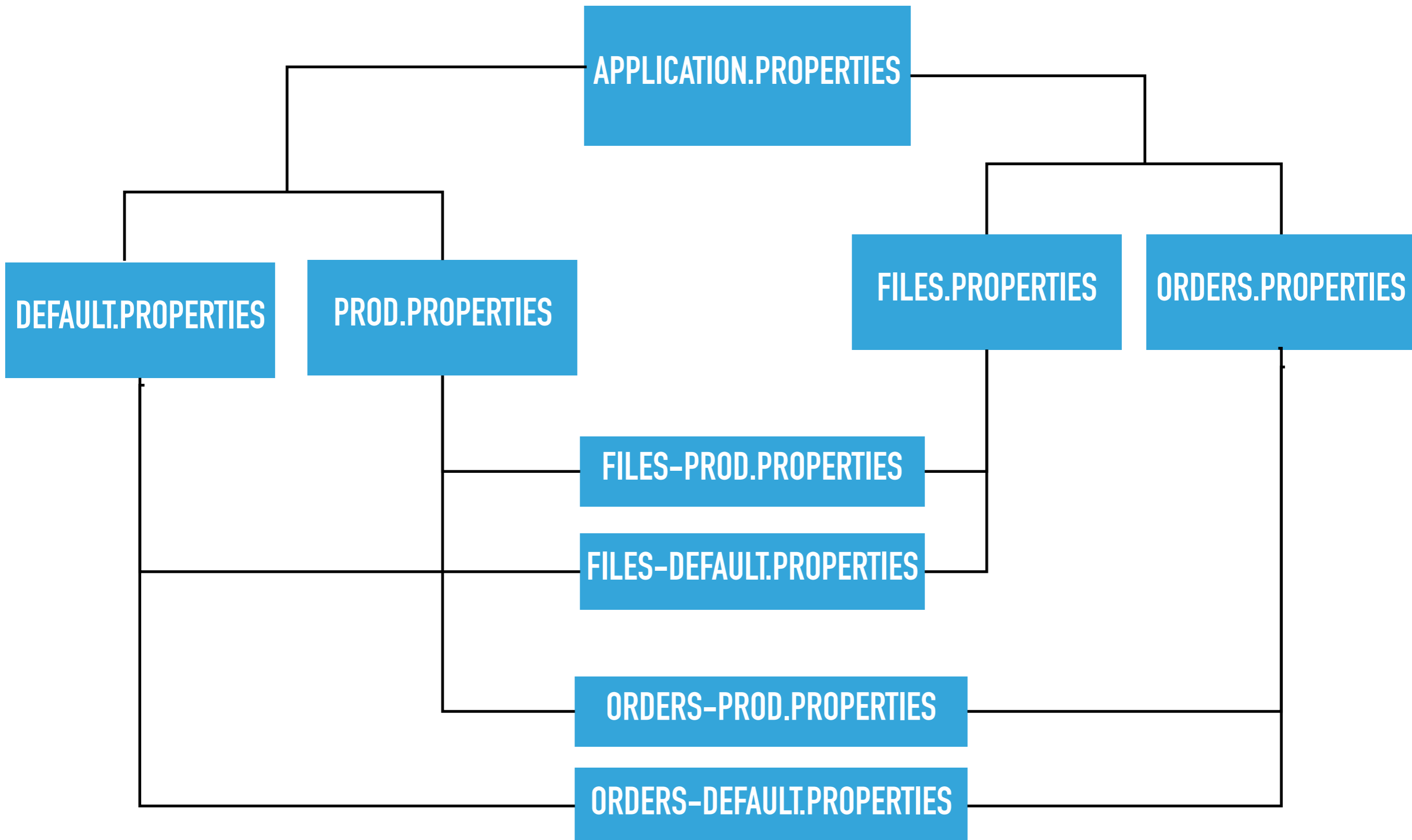


`{"server.port": "8080"}`

Config Server

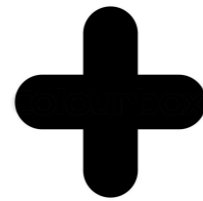


HIERARQUIA DE CONFIGURAÇÕES



SPRING CLOUD CONFIG + FEATURE TOGGLE

Config Server



Feature Toggle



```
feature.clients.active=true
```

```
@Component
@ConfigurationProperties
@RefreshScope
public class RefreshProperties {

    @Value("${feature.toggle.feature.hello.active}")
    private boolean helloApiActive;

}
```

```
curl -d {} http://localhost:8090/actuator/refresh
```

```
@RestController
@RequestMapping("/web")
public class WebController {

    @GetMapping(value = "/hello")
    public String getHello() {
        if(!refreshProperties.isHelloFeatureActive()) {
            throw new InactiveFeatureException("hello inactive");
        }

        return "Web API";
    }
}
```



Config Server



```
curl -d {} http://.../actuator/refresh
```

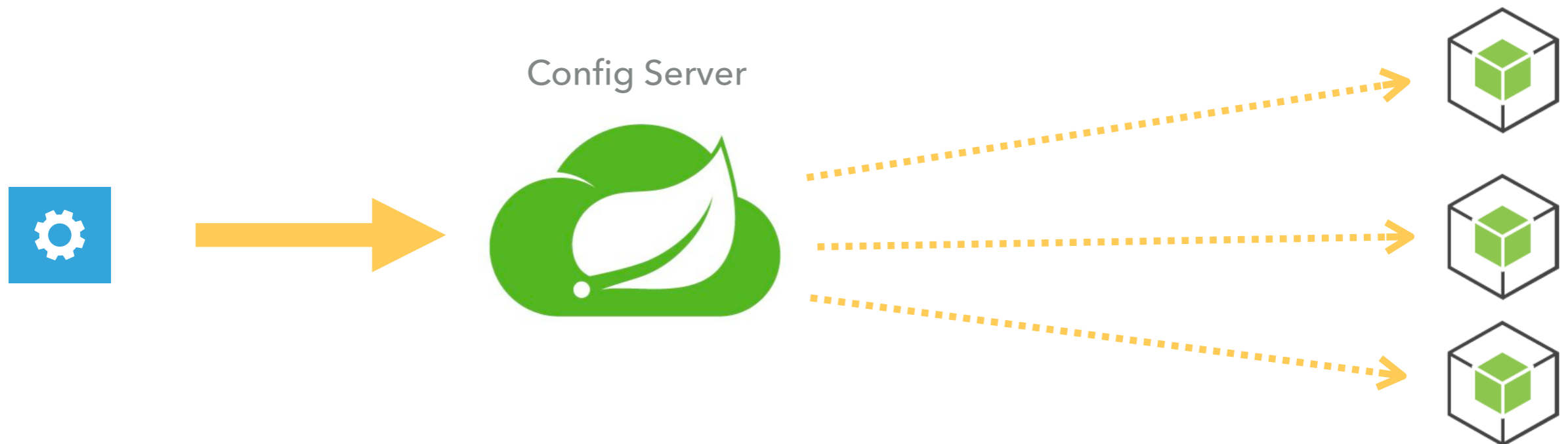


```
curl -d {} http://.../actuator/refresh
```

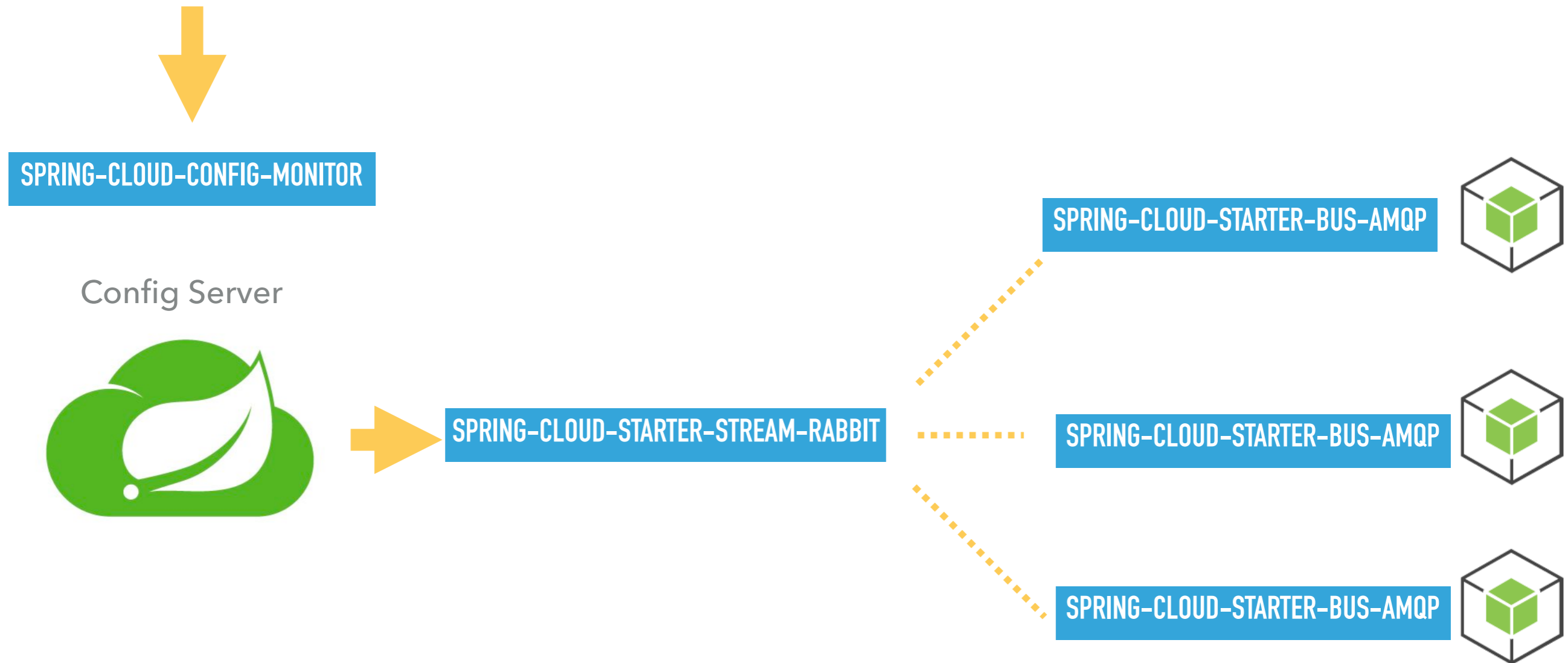


```
curl -d {} http://.../actuator/refresh
```

ATUALIZAÇÕES AUTOMÁTICAS



SPRING CLOUD STACK

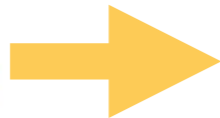


ATUALIZAÇÕES AUTOMÁTICAS



`feature.clients.active=false`

Config Server



`feature.clients.active=false`

`feature.clients.active=true`



`feature.clients.active=false`

`feature.clients.active=true`



`feature.clients.active=false`

`feature.clients.active=true`



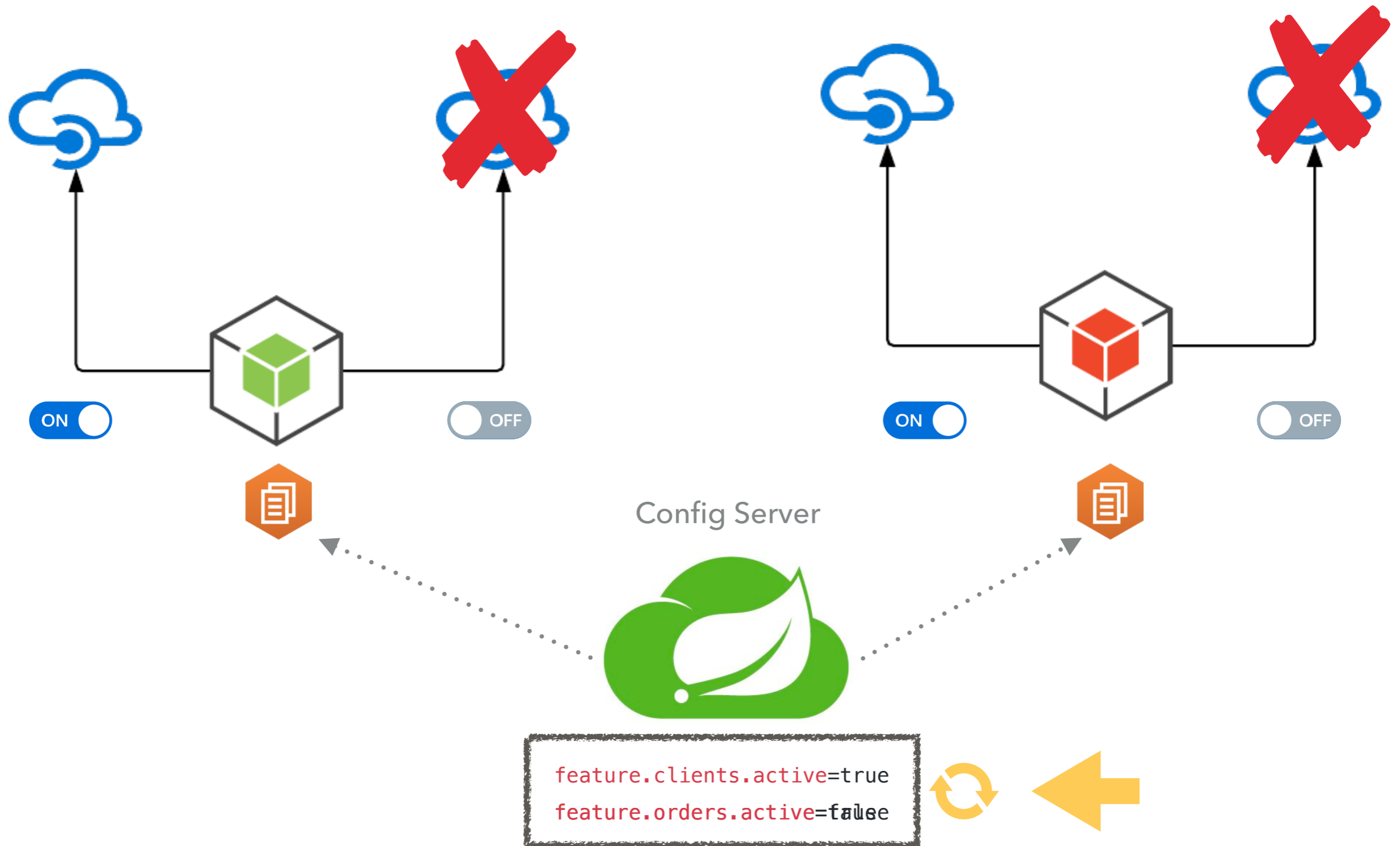
`feature.clients.active=false`

```
@GetMapping(value = "/hello")
@FeatureToggle(
    enableKey = "feature.toggle.feature.hello.active",
    nameKey = "feature.toggle.feature.hello.name")
public String getHello() {
    return "Hello Feature Toggle";
}
```

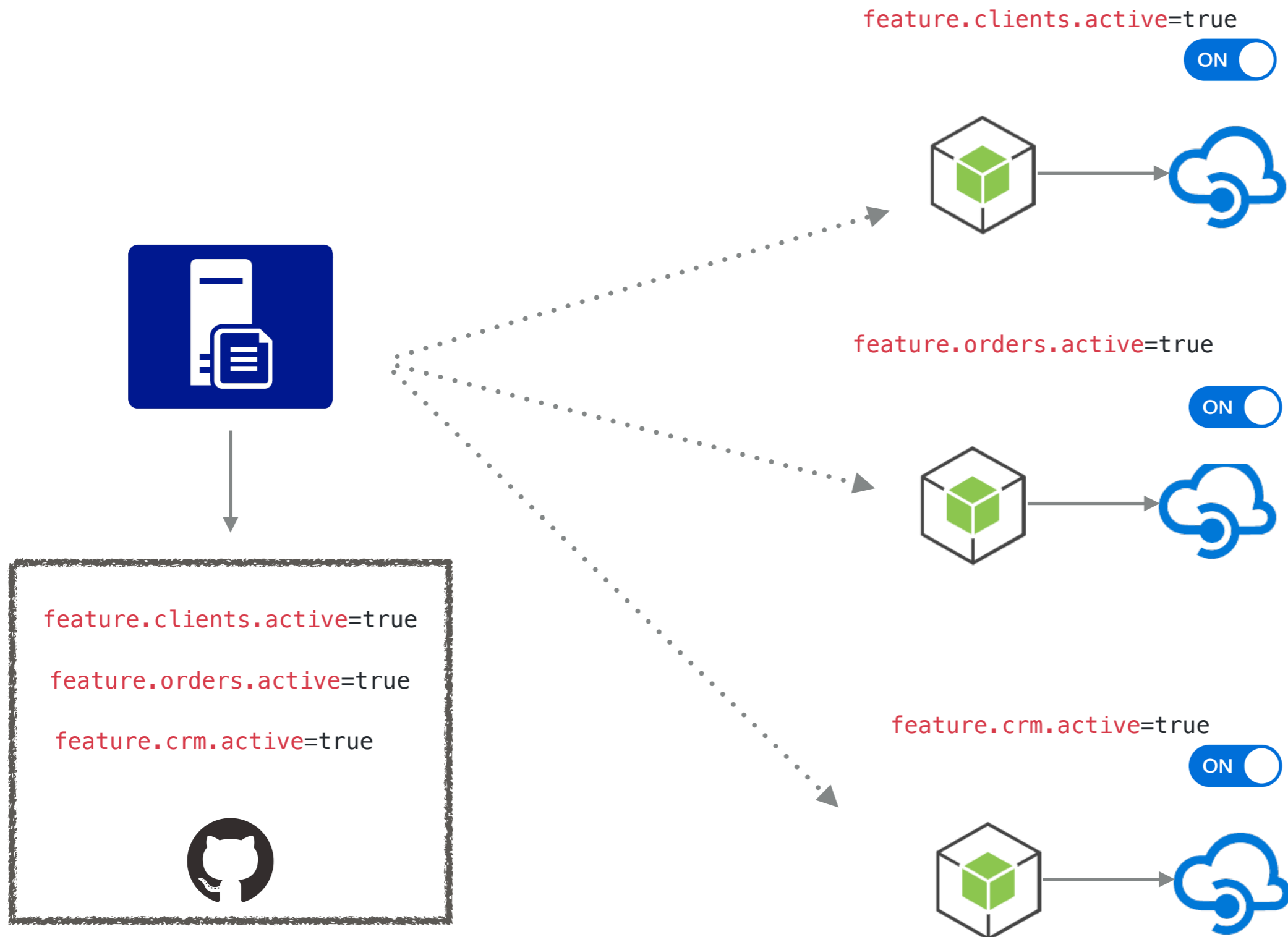
ConfigServer/application.properties

```
feature.toggle.feature.hello.name=Hello
feature.toggle.feature.hello.active=true
```

PREPARANDO



FEATURE TOGGLE + CONFIGURAÇÕES DISTRIBUÍDAS



FRAMEWORKS FEATURE TOGGLES



- ▶ Strategy-based Toggling
- ▶ Role-based Toggling
- ▶ Auditing
- ▶ Database Support
- ▶ Web console

<https://github.com/emmanuelneri/fault-tolerant-features>

OBRIGADO!



@emmanuelnerii



github.com/emmanuelneri



www.linkedin.com/in/emmanuelnerisouza



www.emmanuelneri.com.br



emmanuelnerisouza@gmail.com