

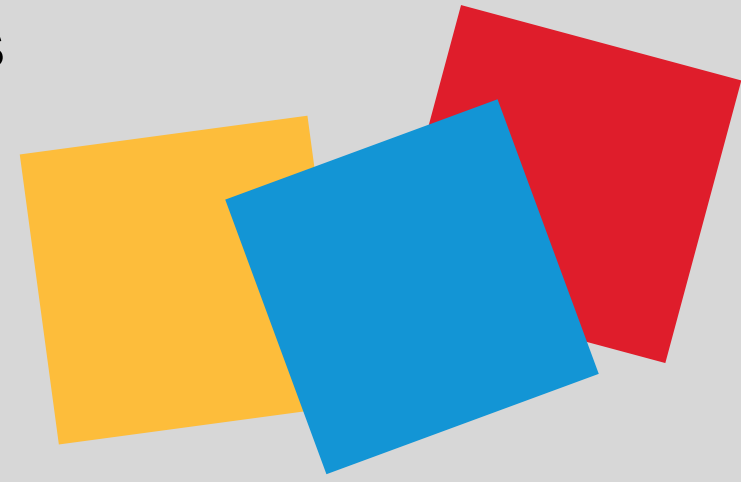
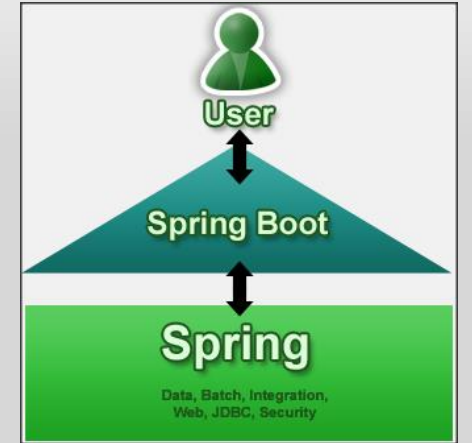
# **B.evezetés a Spring Boot rejtelmeibe**

**Microservice architektúra kialakítása  
JAVA alapokon**



# Introduction

- A tool for getting started very quickly with Spring
- Single point of focus (as opposed to large collection of spring-\* projects)
- Common non-functional requirements for a "real" application
- Exposes a lot of useful features by default
- Gets out of the way quickly if you want to change defaults



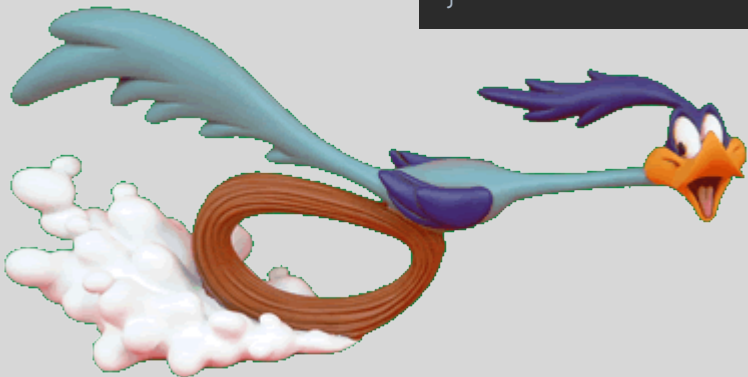
# Spring Boot features

- Create standalone Spring applications
- Automatically configure Spring whenever possible
- production-ready features such as metrics, health checks, externalized configs
- Absolutely no code generation and no requirement for XML configuration



# Spring Boot Hello World

```
@SpringBootApplication
@ComponentScan(useDefaultFilters = false)
@Controller
public class Application {
    @RequestMapping("/")
    @ResponseBody
    String sayHello() {
        return "Hello World!";
    }
    public static void main(String[] args) throws Exception {
        SpringApplication.run(Application.class, args);
    }
}
```

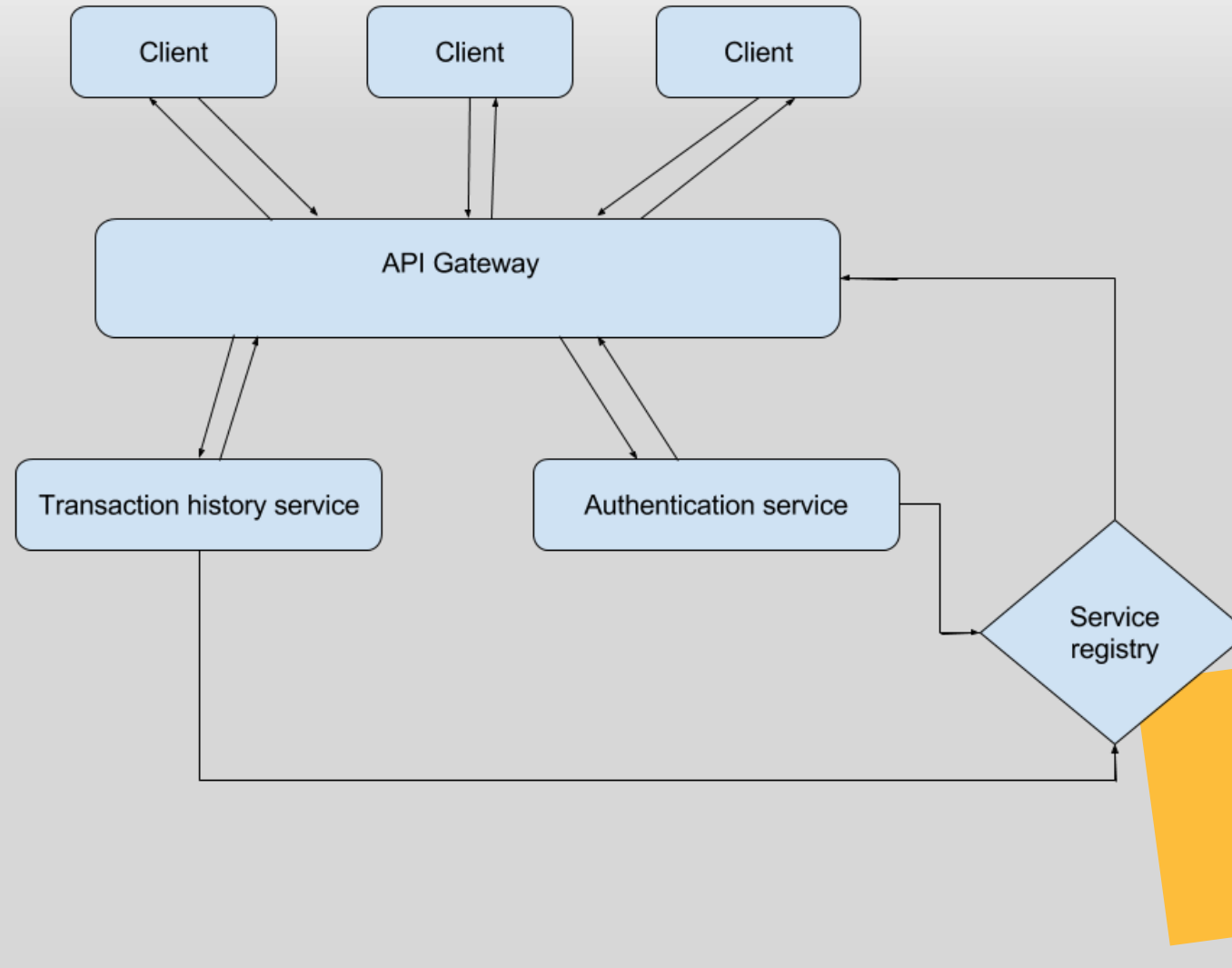


# Spring Boot and Spring Cloud

- An open-source library makes it easy to develop JVM applications for the cloud.
- It provides tools for developers to quickly build some of the common patterns in distributed systems
- e.g. service registration and discovery, load balancing etc.
- Spring Cloud is built on Spring Boot and utilizes parent and starter POMs.

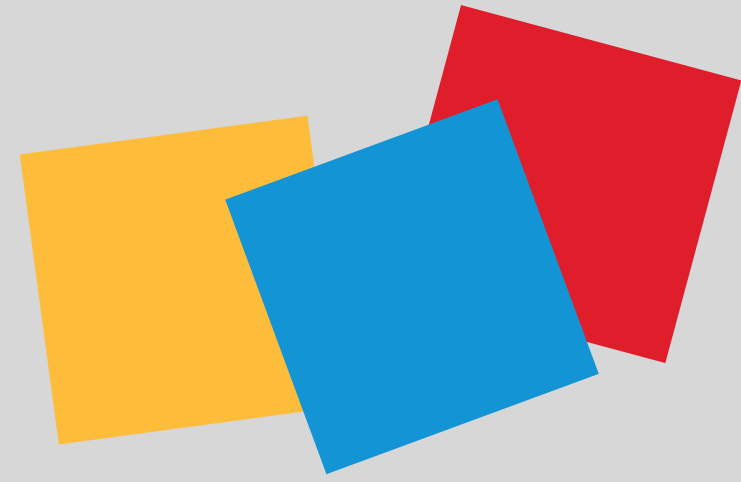


# API Gateway pattern



## Demo

- Netflix Eureka service registry
- API Gateway instance
- Authentication service
- Transaction history service



# T.hank you!

