

Effortless Spring HTTP Interfaces with Client Registry



Rossen Stoyanchev

Spring engineering team

Spring Framework, Spring for GraphQL



GET /path

Content-Type: application-json

Host: localhost:8080

RestClient

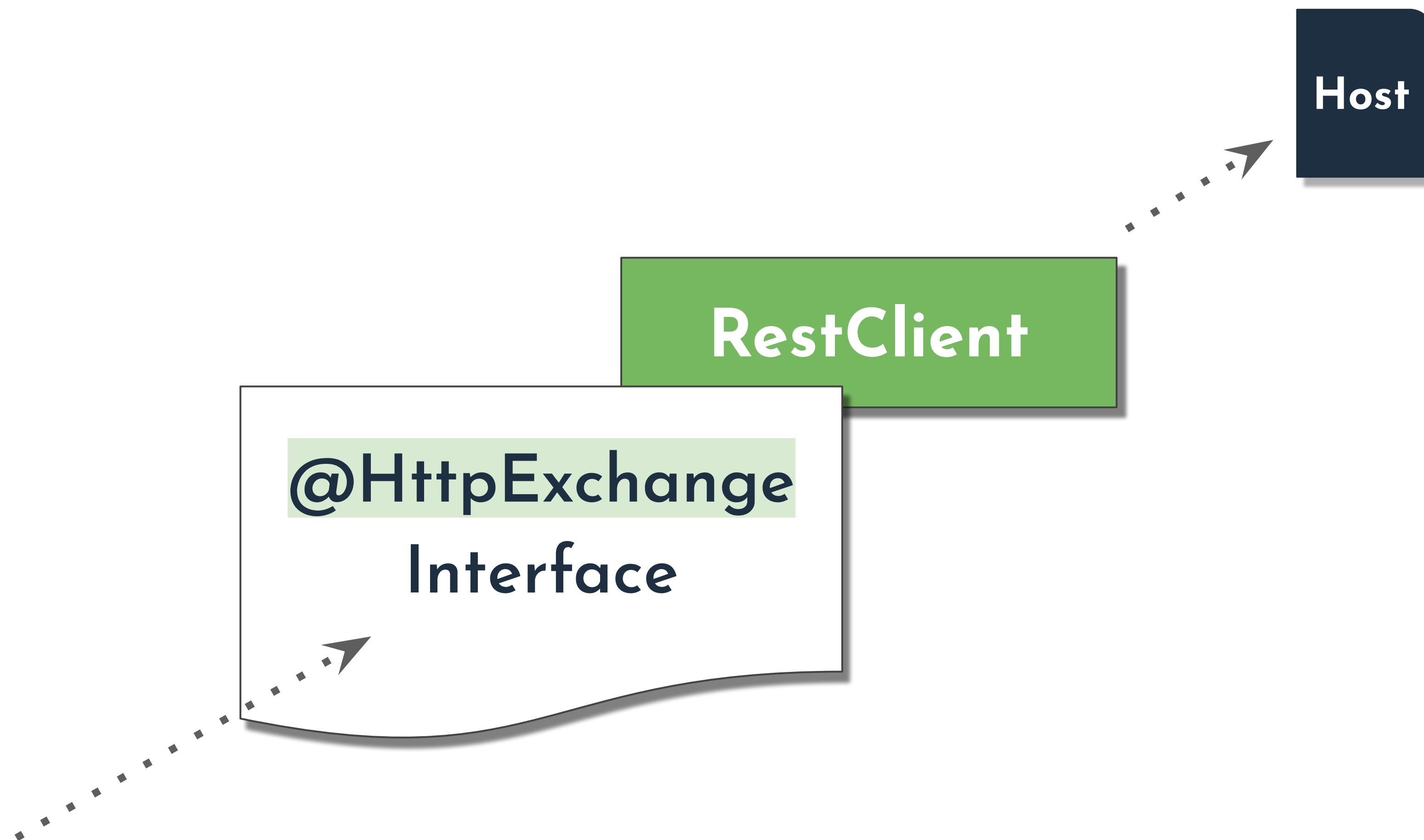
Host

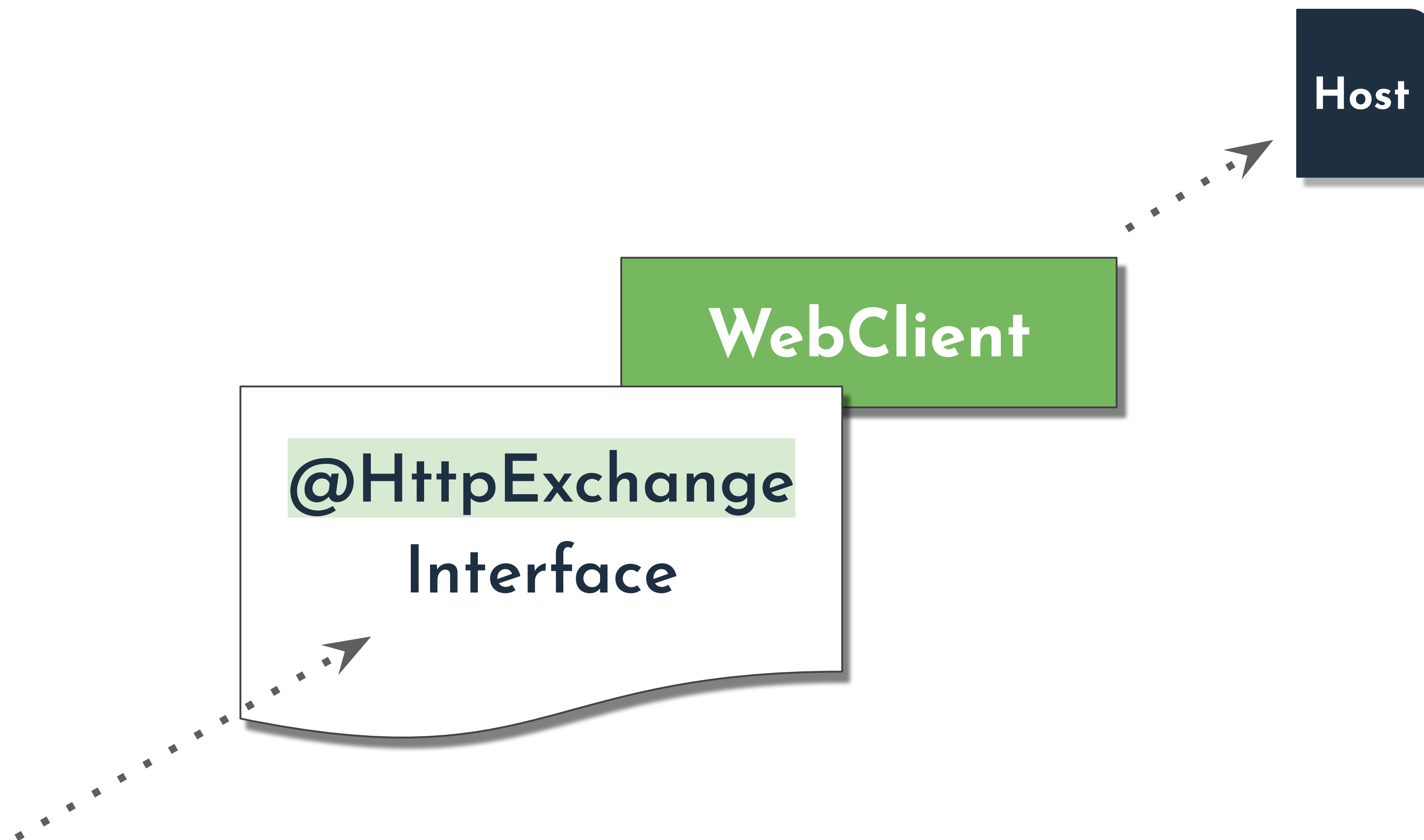


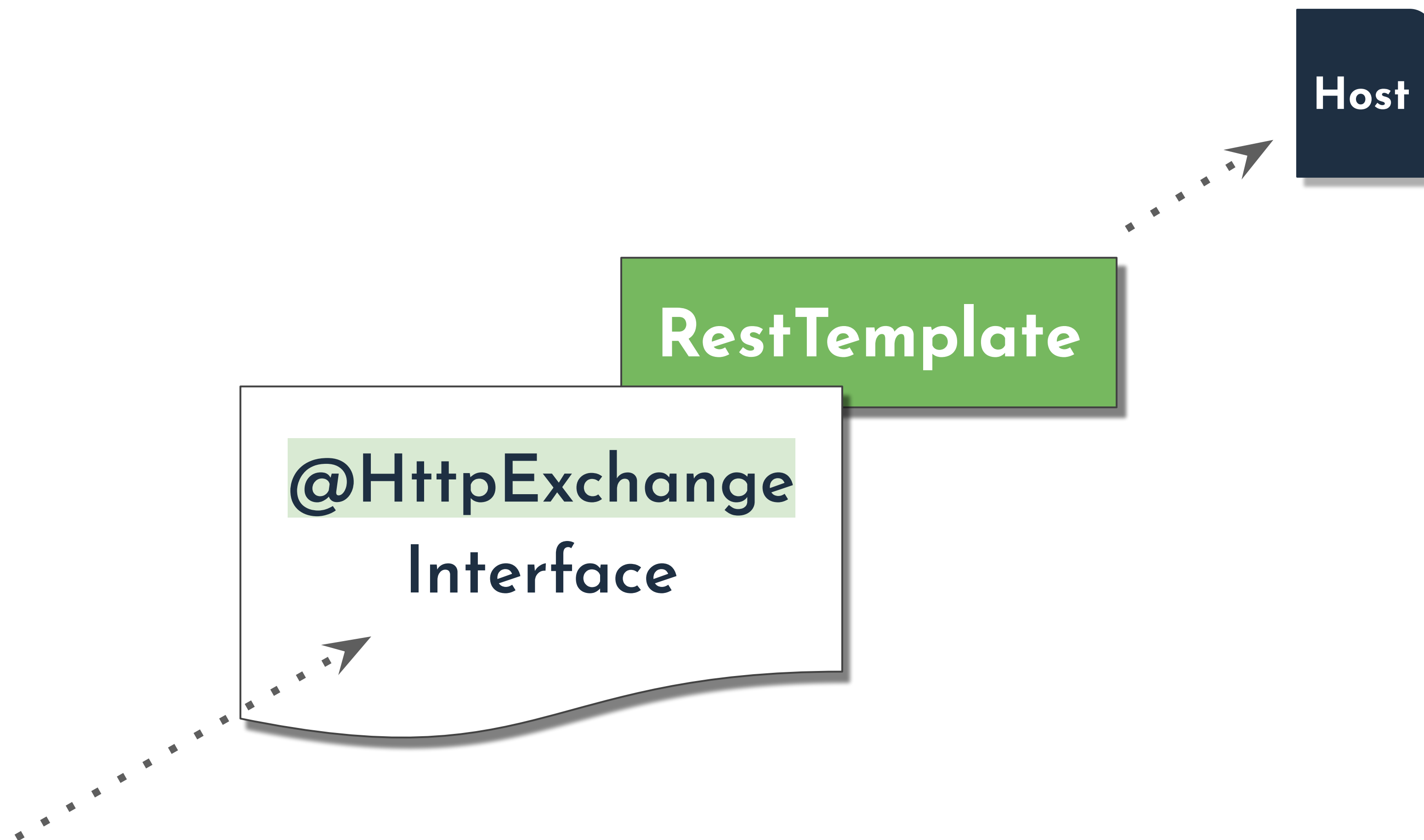
HTTP Service

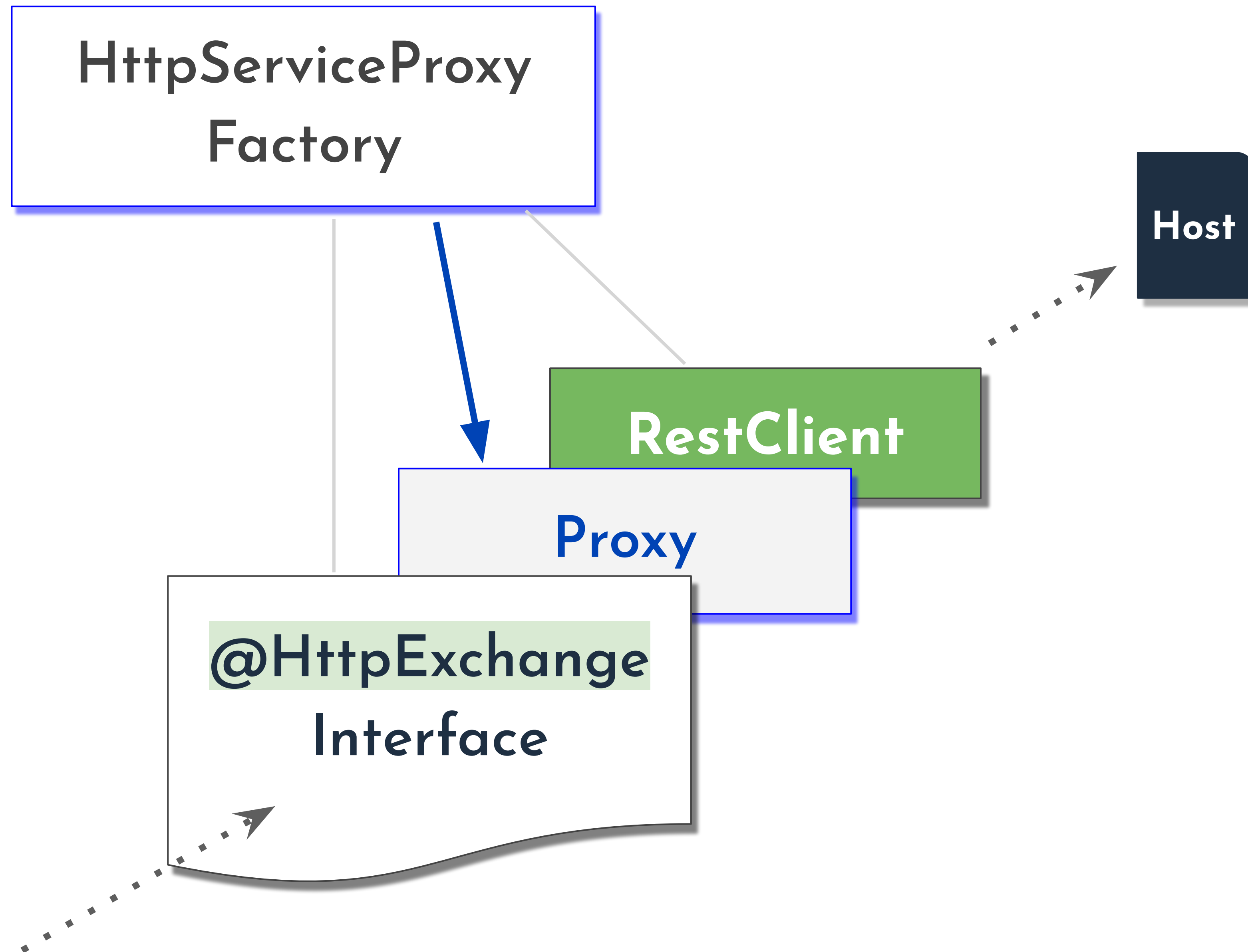
Spring Framework 6

```
public interface MilestoneService {  
  
    @GetExchange(🌐✓ "/repos/{org}/{repo}/milestones")  
    List<Milestone> getMilestones(  
        @PathVariable String org, @PathVariable String repo,  
        @RequestParam State state);  
  
}
```









@Configuration

public class DemoConfig {

 @Bean

 MilestoneService milestoneService(HttpServiceProxyFactory factory) {
 return factory.createClient(MilestoneService.class);
 }

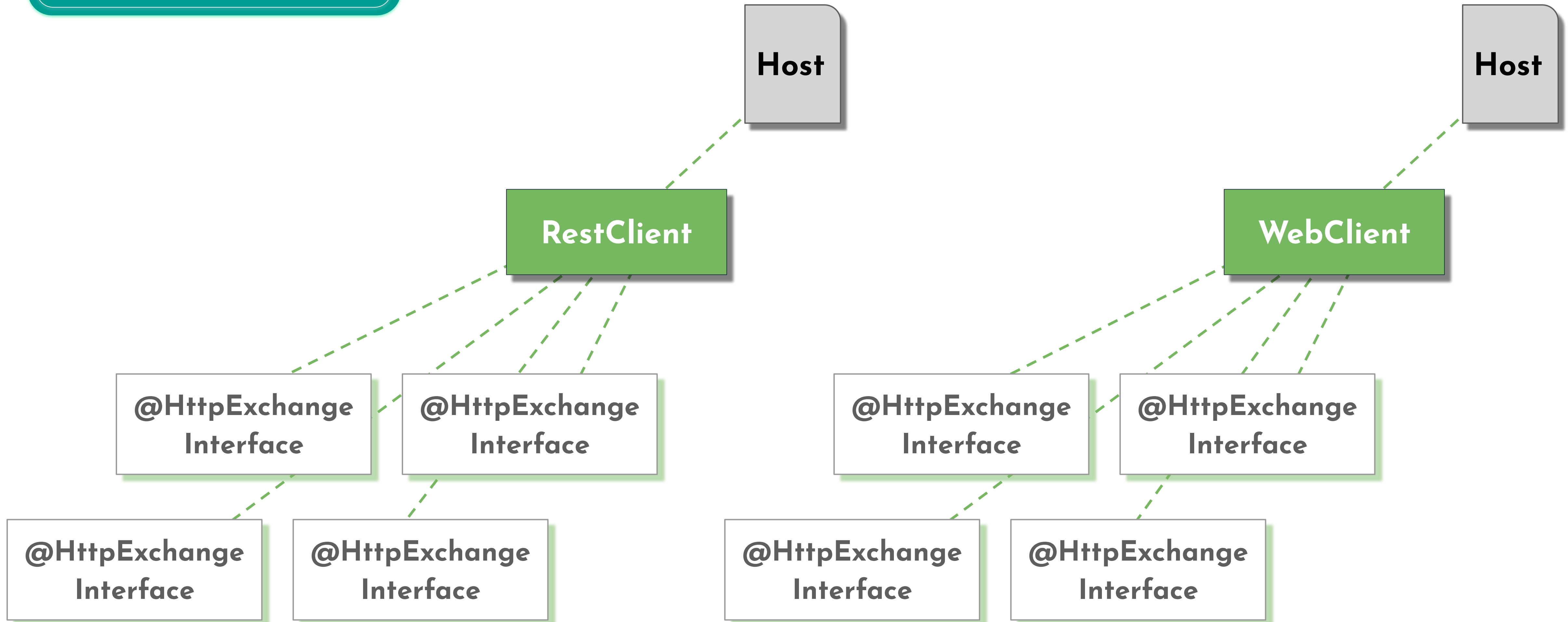
 @Bean

 HttpServiceProxyFactory httpServiceProxyFactory(RestClient client) {
 return HttpServiceProxyFactory.builder()
 .exchangeAdapter(RestClientAdapter.create(client))
 .build();
 }

}



Configuration Challenge





boilerplate

@Configuration

```
public class DemoConfig {
```

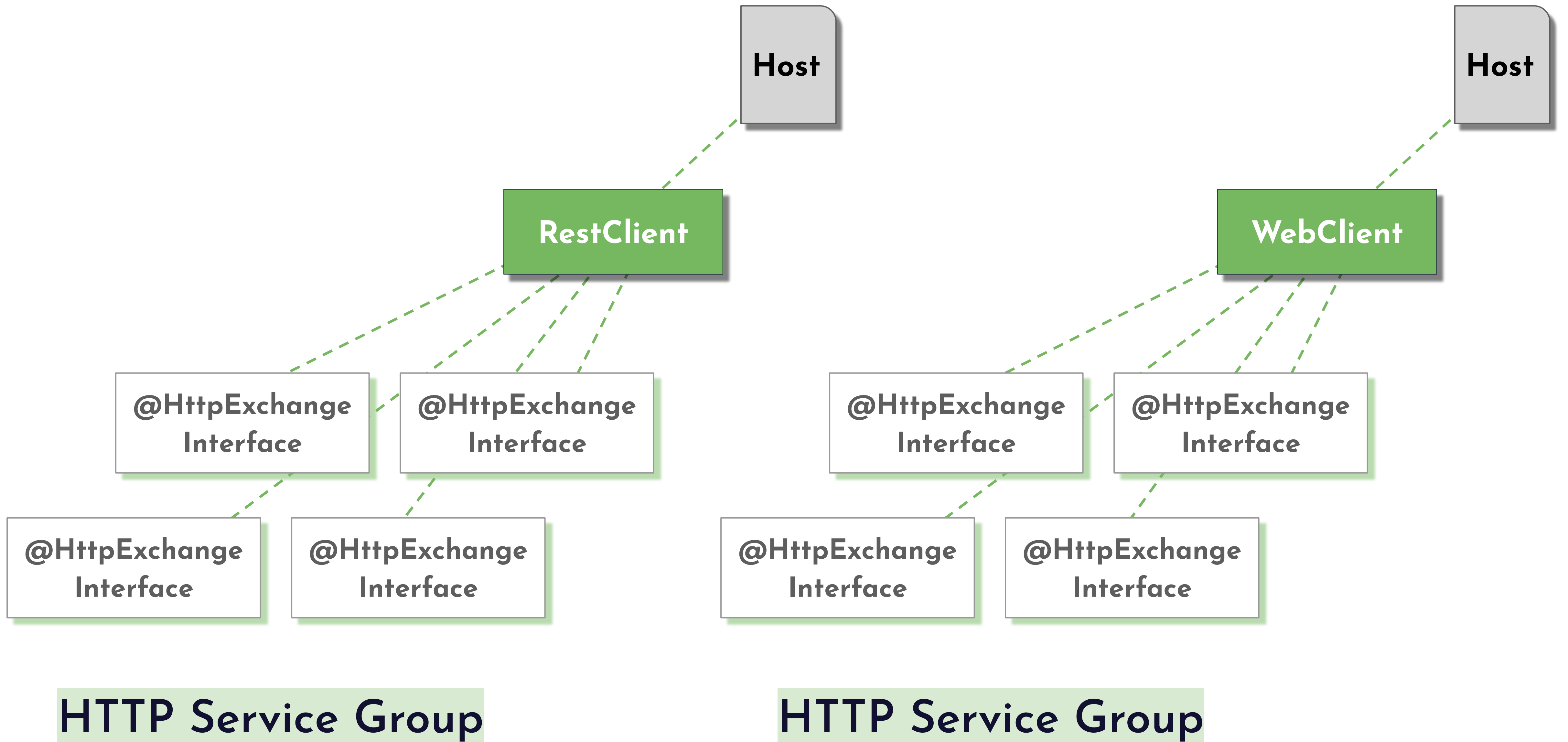
```
    @Bean
```

```
    MilestoneService milestoneService(HttpServiceProxyFactory factory) {  
        return factory.createClient(MilestoneService.class);  
    }
```

```
    @Bean
```

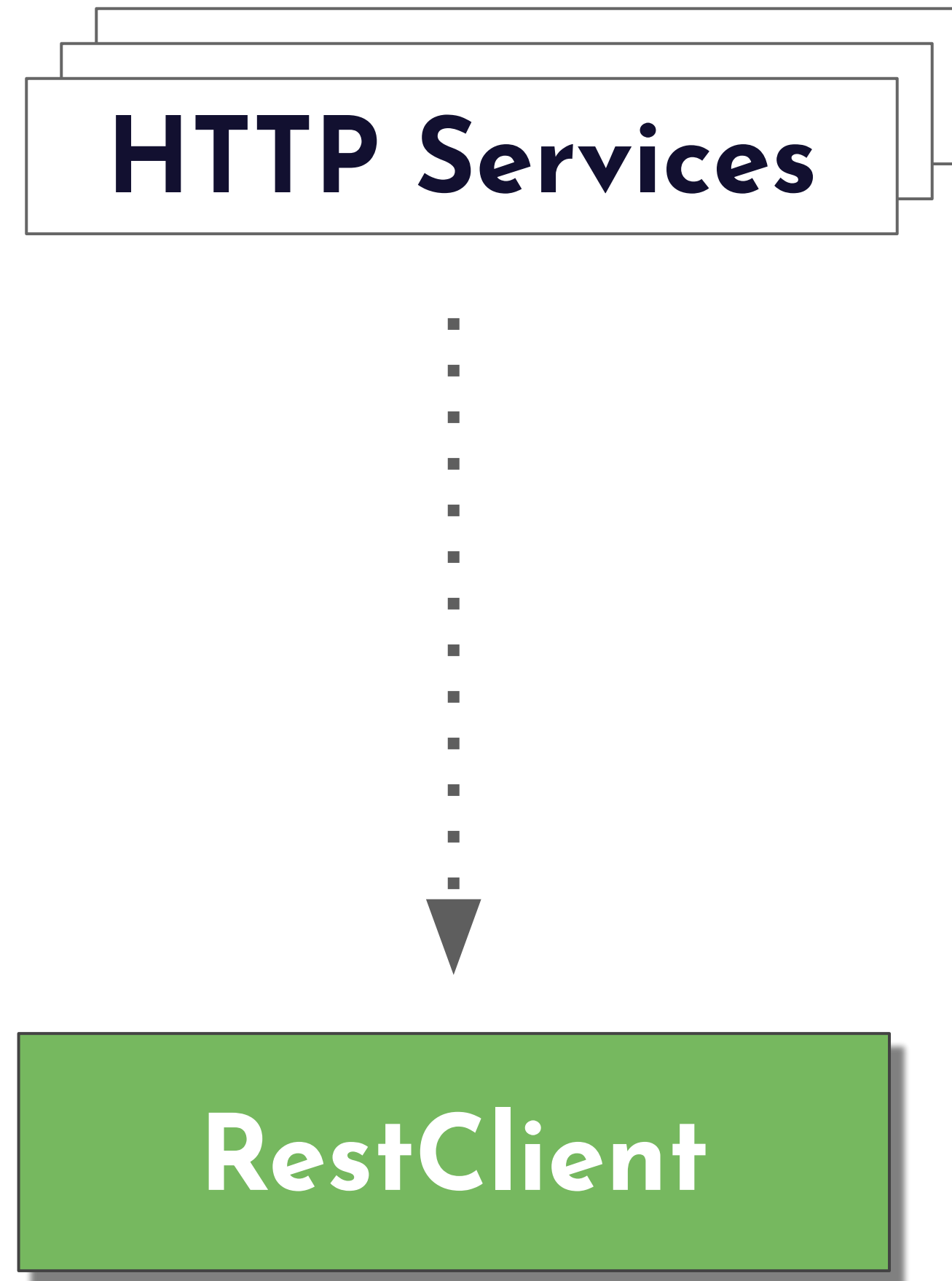
```
    HttpServiceProxyFactory httpServiceProxyFactory(RestClient client) {  
        return HttpServiceProxyFactory.builder()  
            .exchangeAdapter(RestClientAdapter.create(client))  
            .build();  
    }
```

```
}
```



HTTP Service Group

1. Identify services



HTTP Service Group

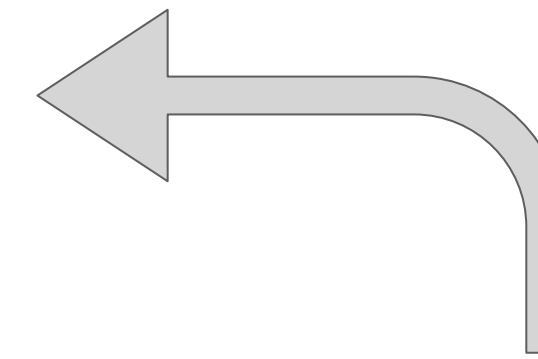
1. Identify services

HTTP Services



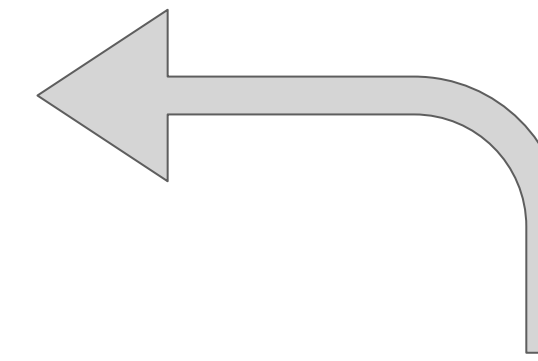
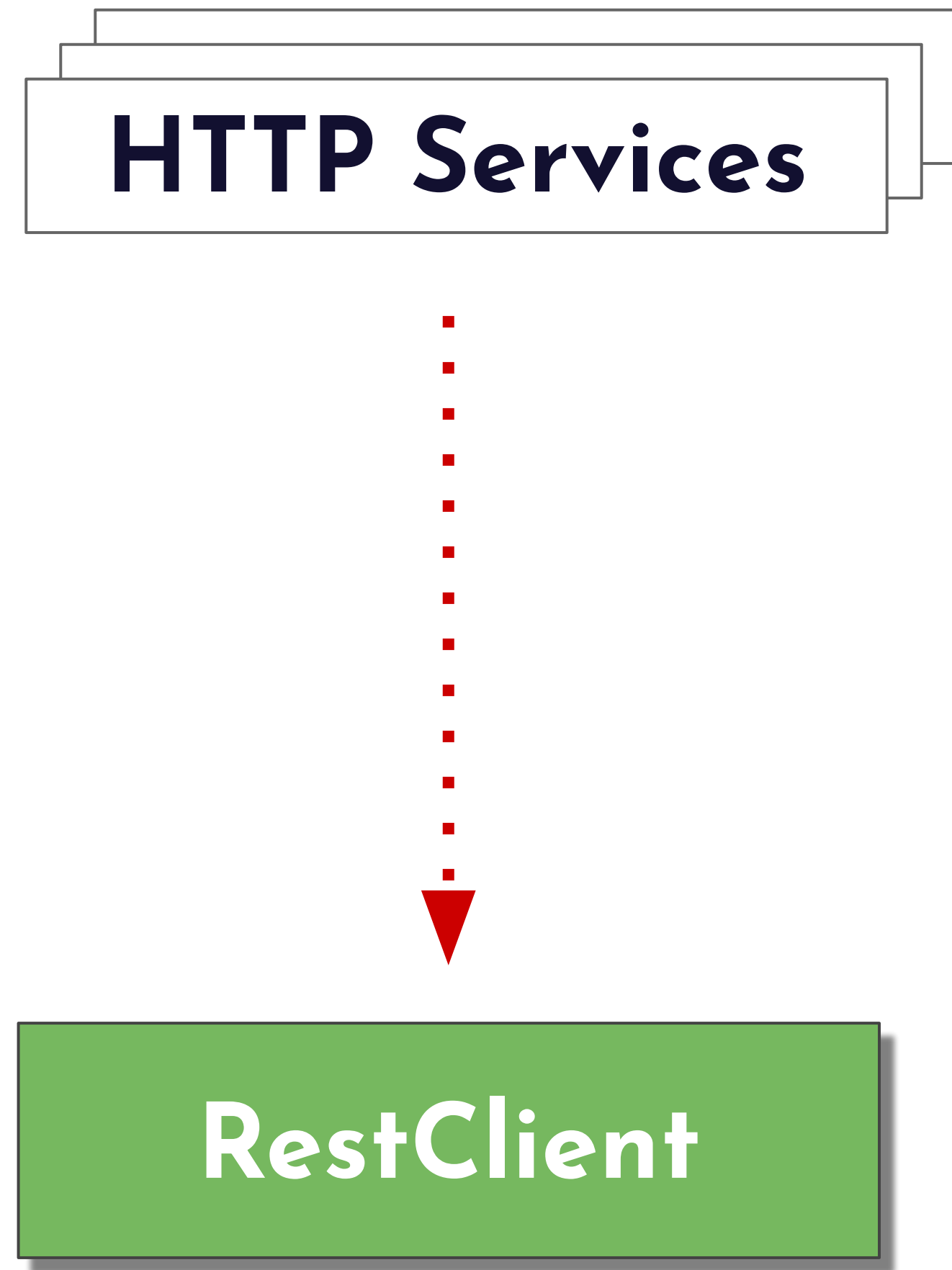
*2. Configure client
(cloud, security)*

RestClient

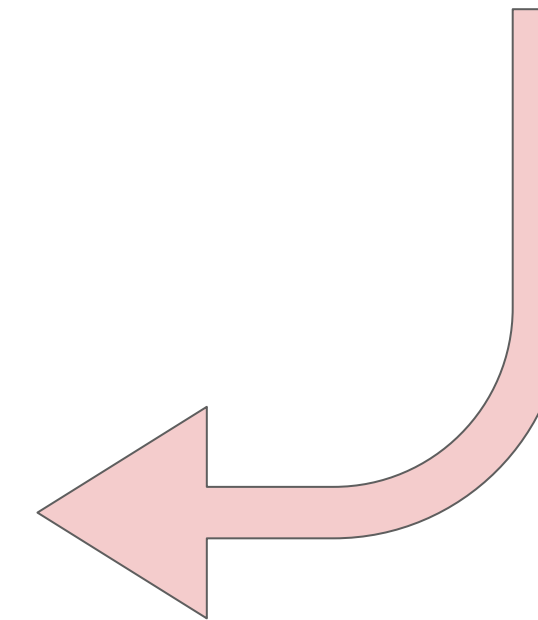


*Create proxies and
declare beans*

Lifecycle Issue



*Create proxies and
declare beans*



depends on

Deferred bean definitions
for HTTP services

*Early Phase
(import registrar)*

Configure clients &
create proxies

*HTTP Service Proxy Registry
(FactoryBean)*

Demo



Import phase

Scan or list via `@ImportHttpServices`

Programmatic registration

Client config

HTTP Service group configurers

Common or individual (per client) config

Spring Boot properties

Spring Security Demo



Security client config

OAuth through

@ClientRegistrationId on HTTP Service

Spring Cloud Demo



Spring Cloud client config

Load balancing

Resources

[springio25-service-registry](#) demo

Spring Cloud [sample](#)

Q & A



THANK YOU

