Microservices
with Spring Cloud

Spencer Gibb, Pivotal

start

Pivotal®

CLOUD FOUNDRY

spring
by Pivotal™

RabbitMQ™
Messaging that just works

redis

APACHE GEODE

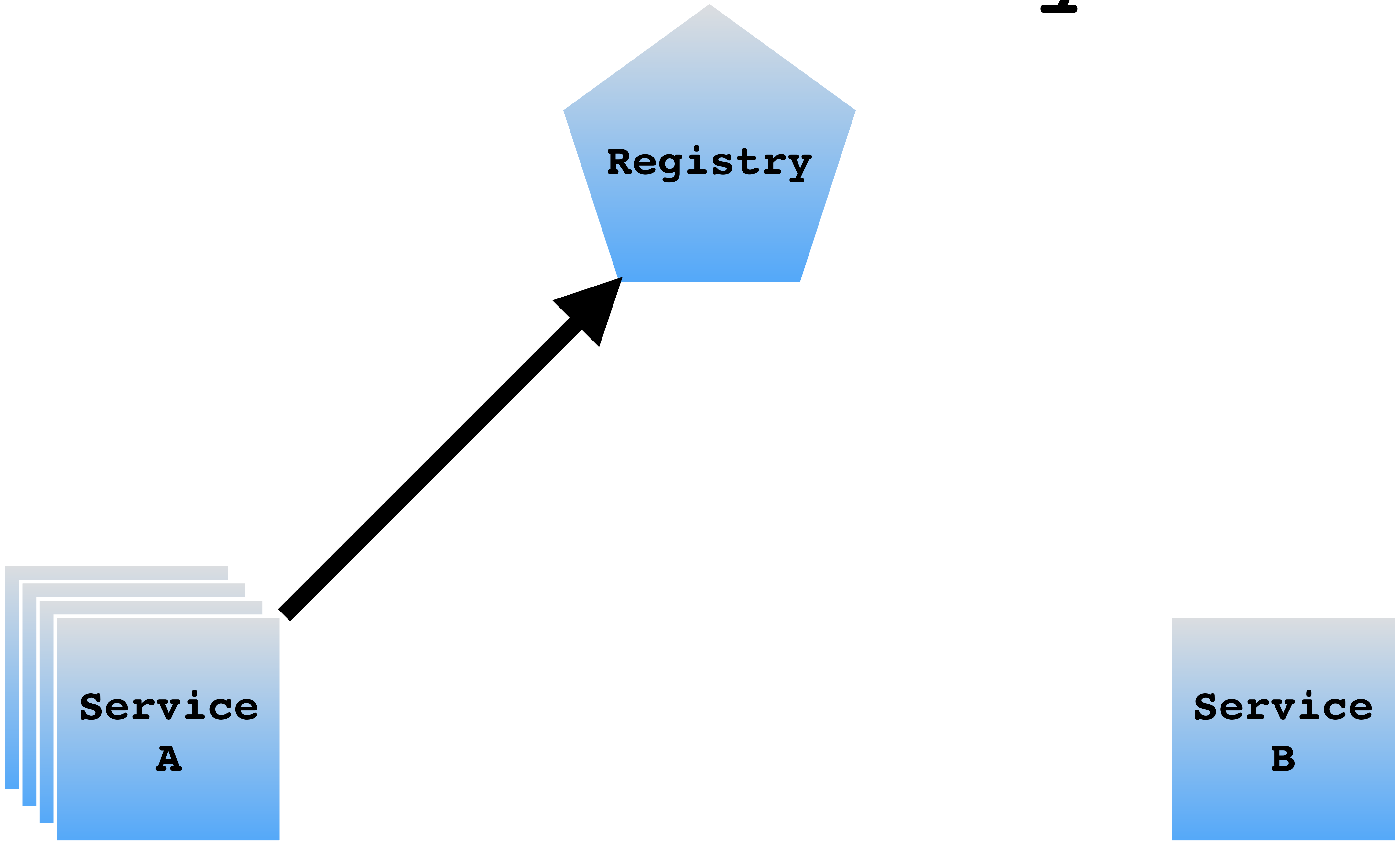
GREENPLUM DATABASE

next

Table of Contents

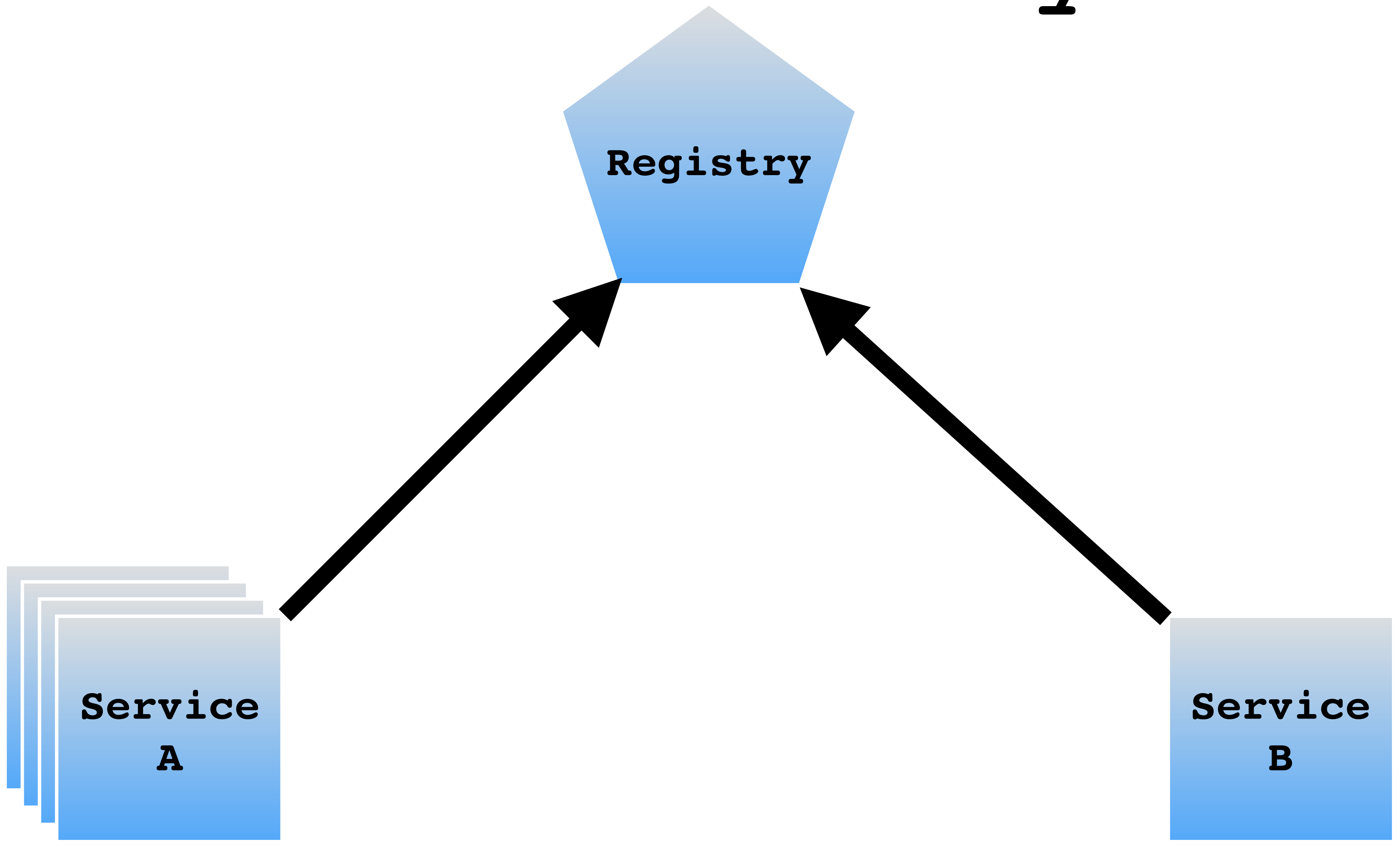
next

Service Registration & Discovery



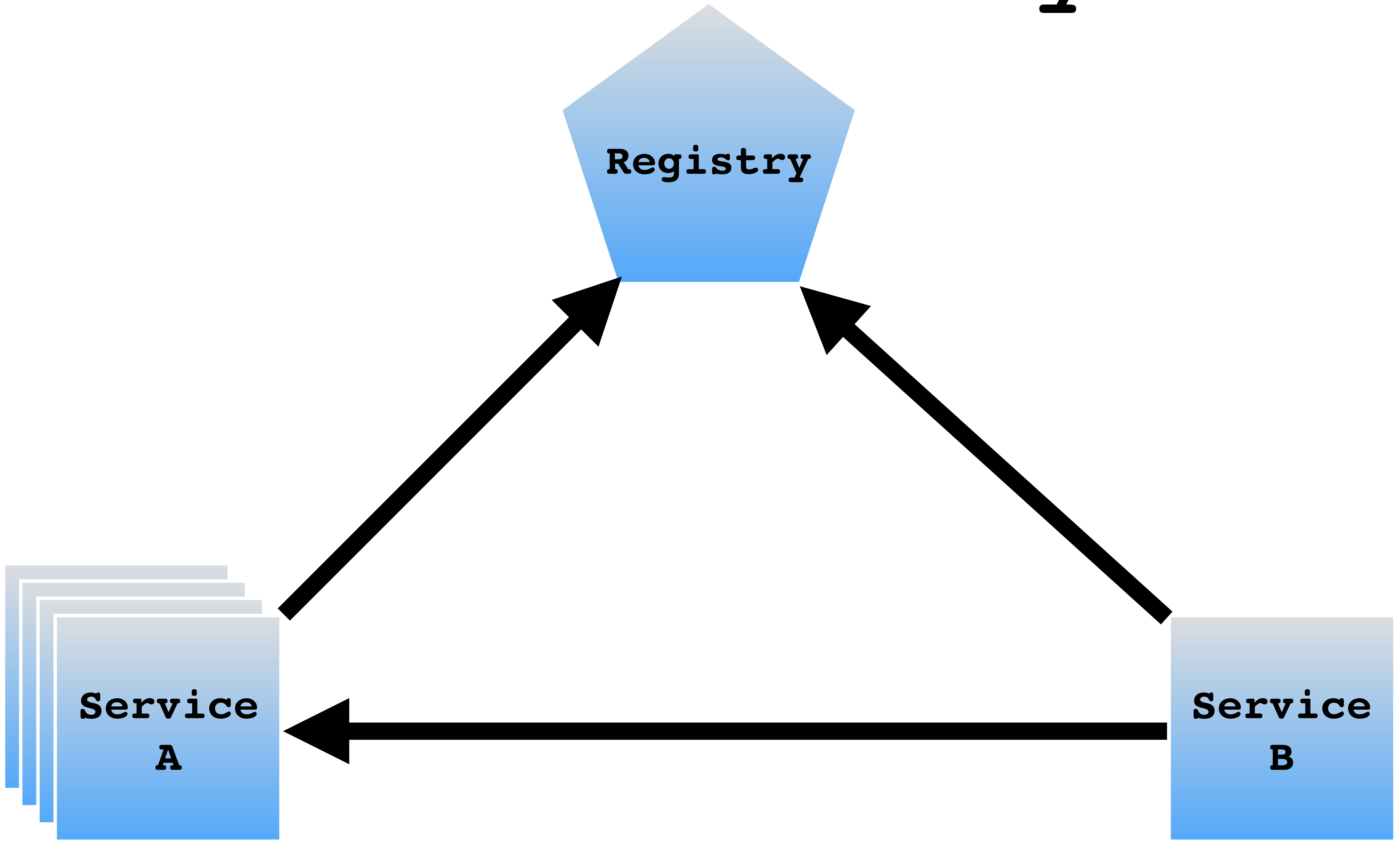
next

Service Registration & Discovery



next

Service Registration & Discovery

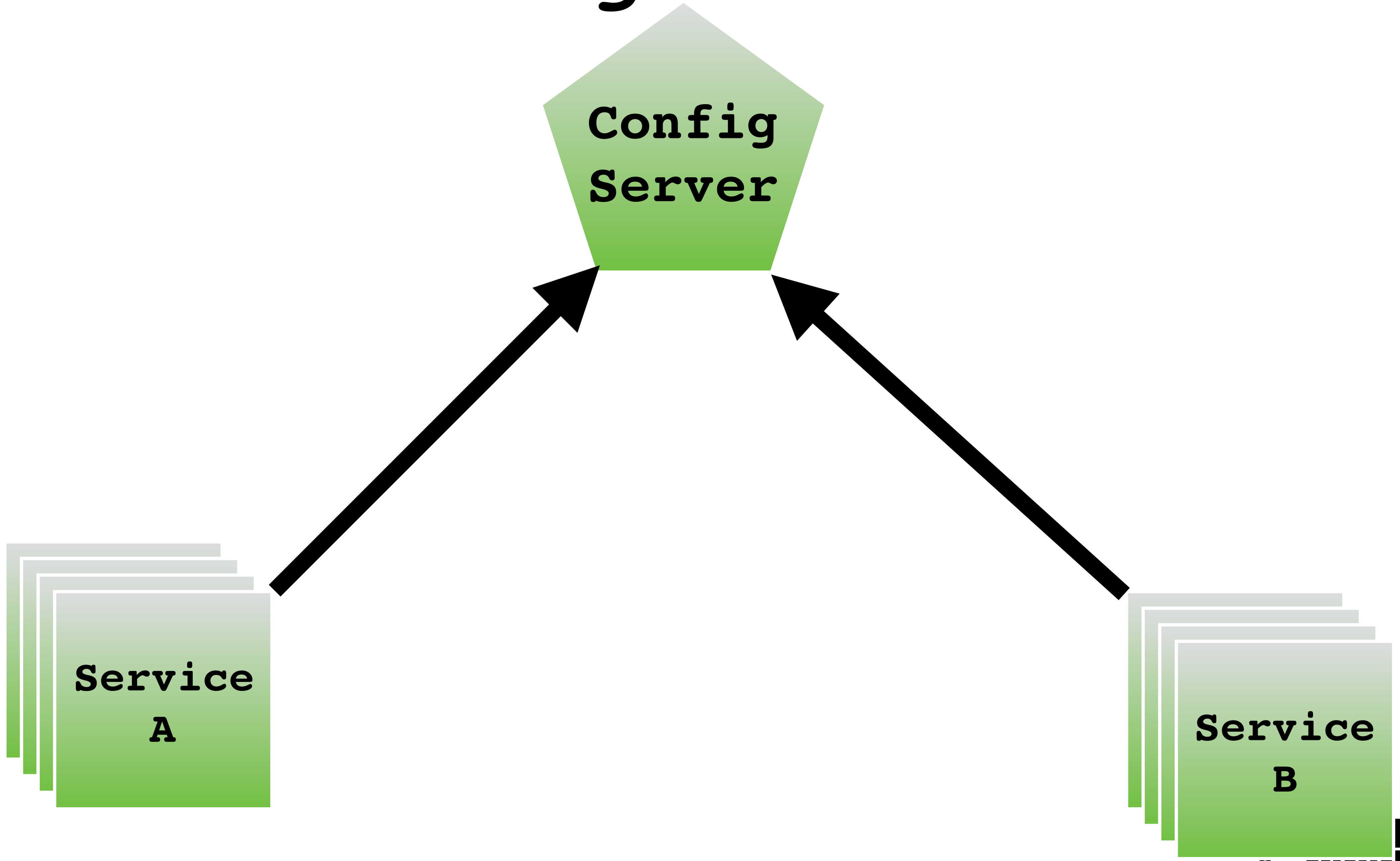


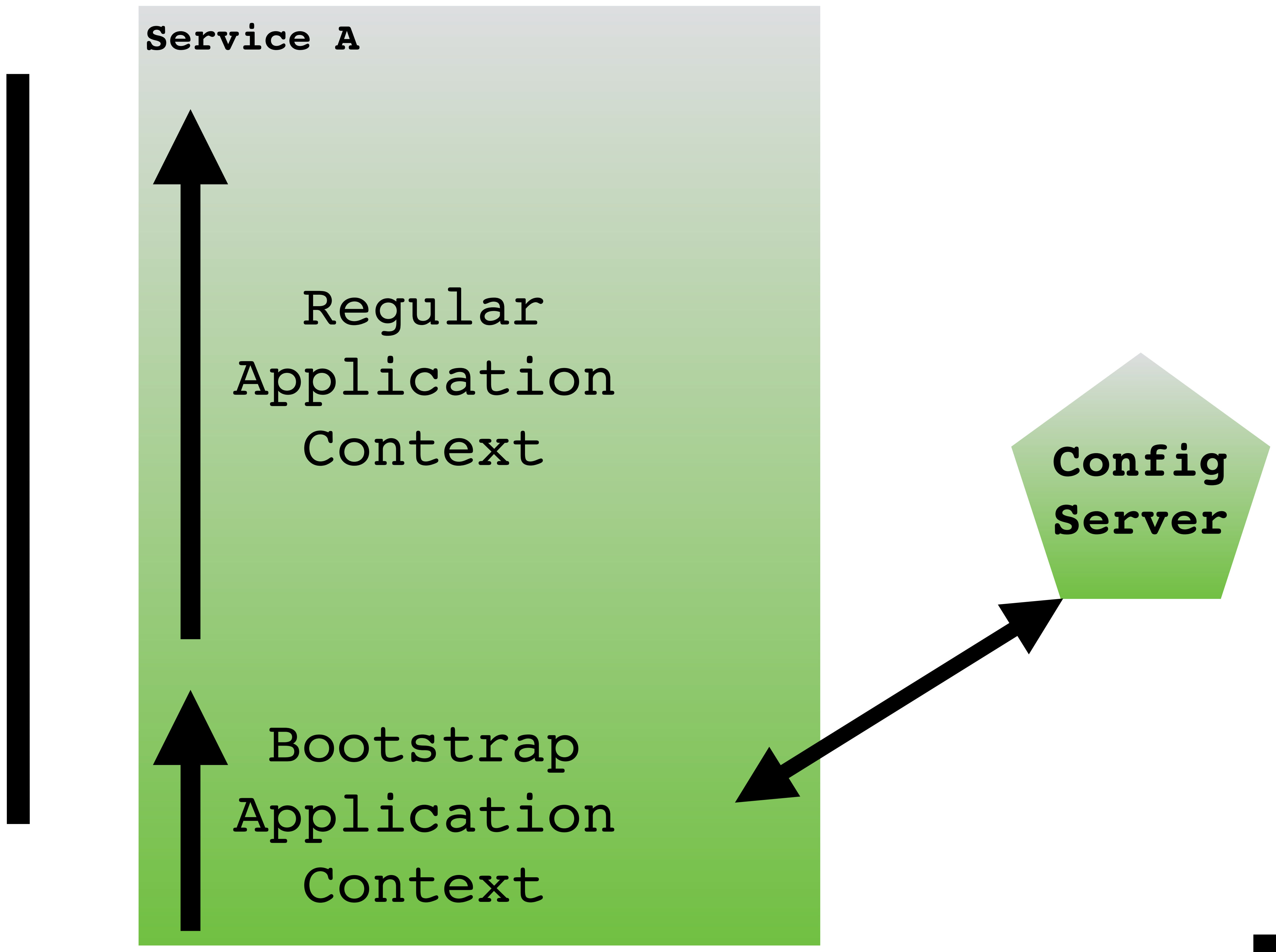
DiscoveryClient

```
@EnableDiscoveryClient

ServiceInstance si =
discoveryClient.choose(“serviceId”)
```

Distributed Configuration



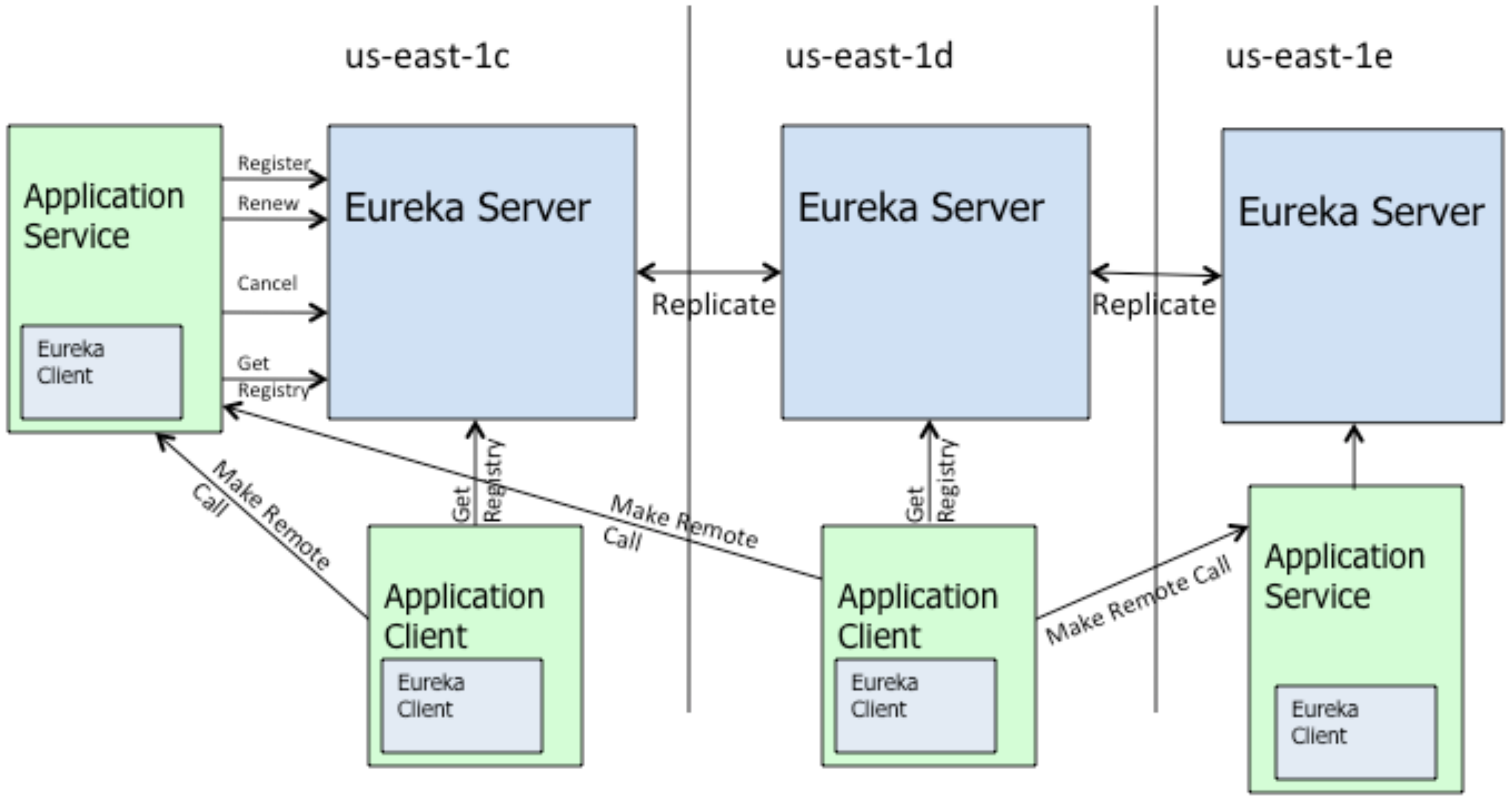

Service A
Regular Application Context
Bootstrap Application Context
Config Server

next

Environment

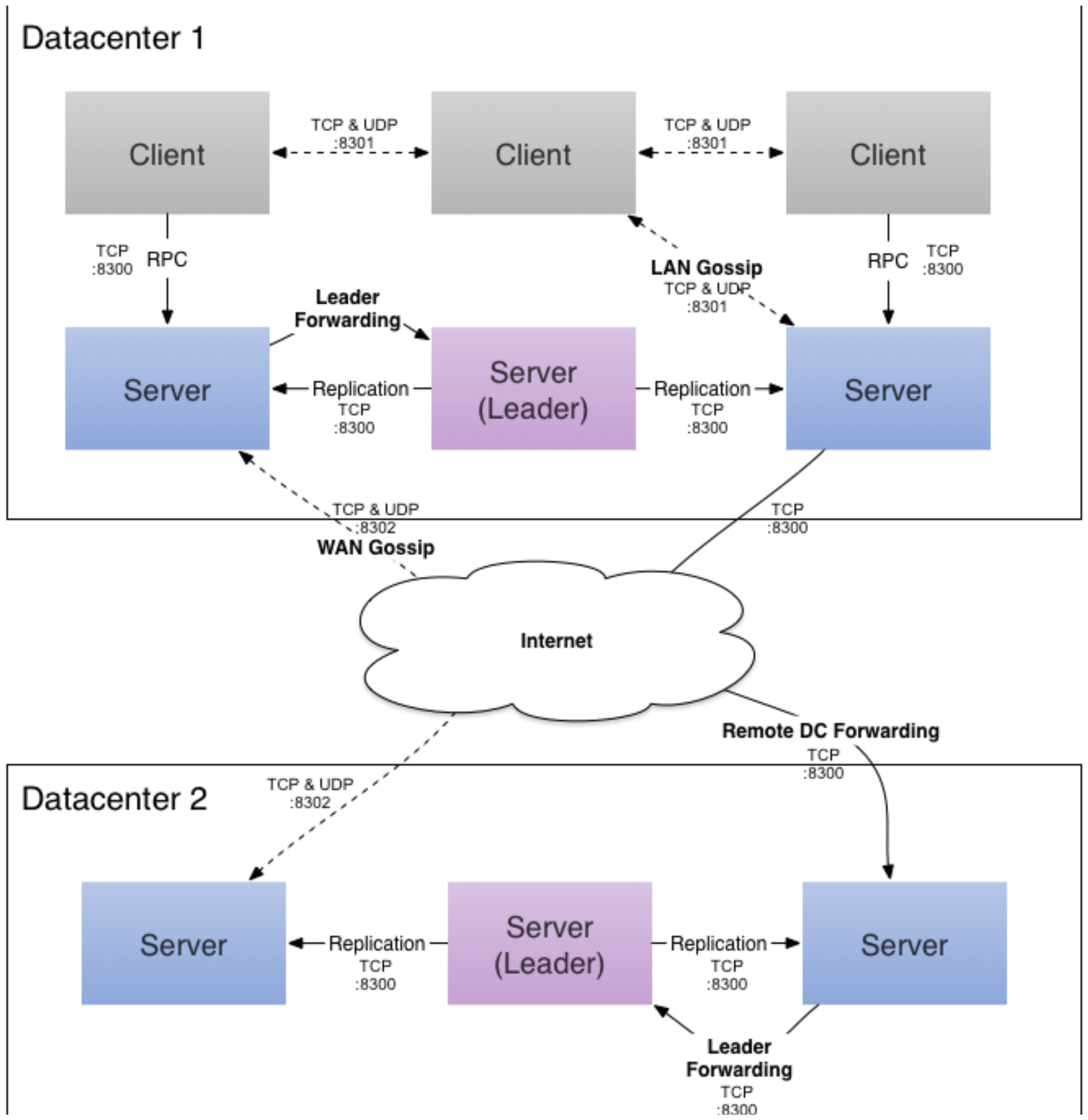
@ConfigurationProperties

@Value



Eureka

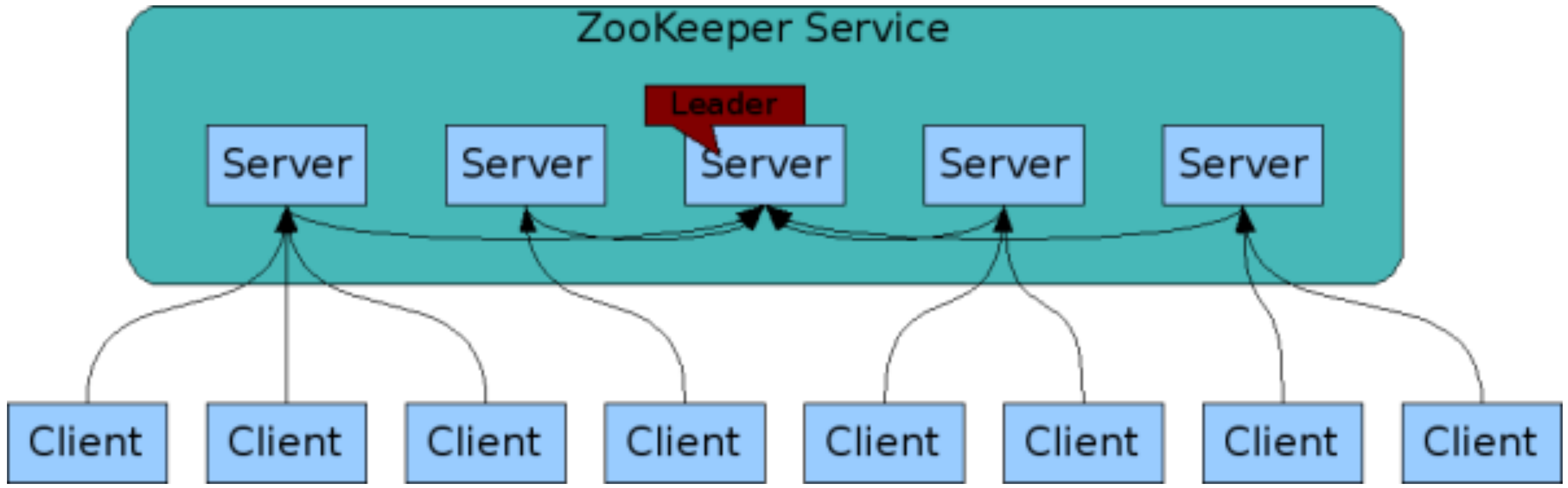
NETFLIX



Consul

HASHICORP™

next



Zookeeper

next

Spring Cloud Sleuth

- Via Josh Long @starbuxman
- Sleuth is a distributed tracing framework: propagate correlation IDs across processes to understand request path
- Sleuth has traces (aggregate journey of a request) and Spans (each hop in journey from egress to ingress point)
- Sleuth Stream marshals captured Sleuth Spans over a Spring Cloud Stream binder (RabbitMQ, Kafka, etc.)
- Stream Zipkin takes marshaled Spans & writes to Zipkin DB for analysis
- Once you have instrumented nodes emitting Spans via Sleuth Stream to Zipkin Stream server, fire up https://github.com/openzipkin/zipkin/tree/master/zipkin-web