Introduction to Spring Cloud Components

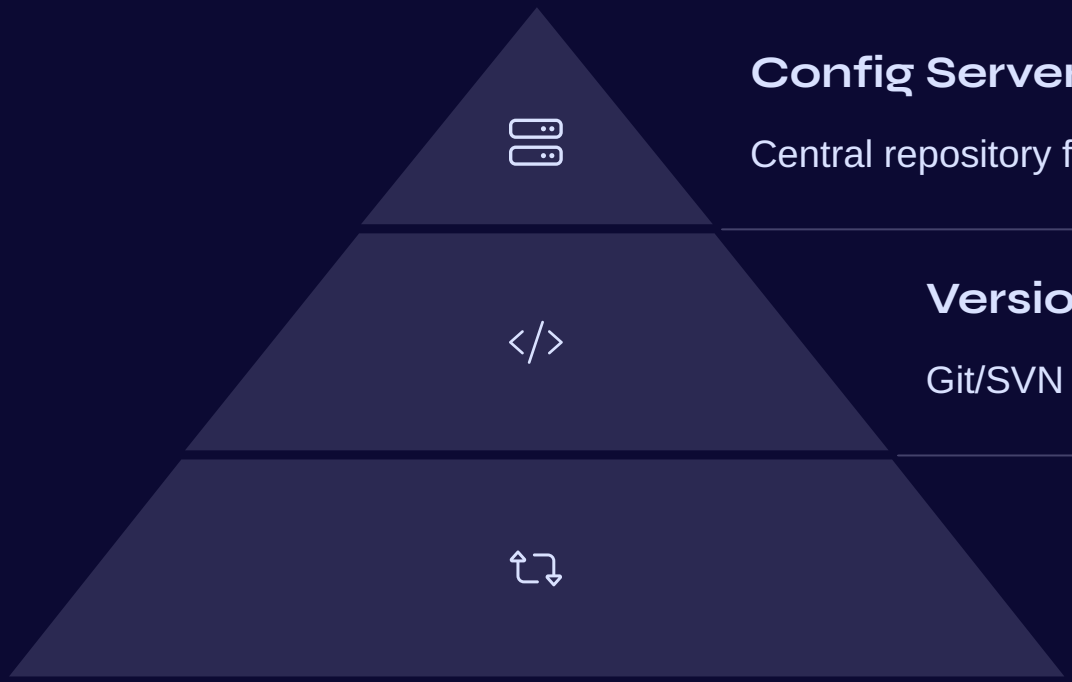
Spring Cloud provides a comprehensive suite for building scalable, cloud-ready applications.

It solves common microservices challenges through ready-to-use components.

These components integrate seamlessly with Spring Boot for rapid development.

S **by Saratha Natarajan**

Centralized Configuration: Spring Cloud Config



Config Server

Central repository for configuration

Version Control

Git/SVN backing for config history

Dynamic Refresh

Runtime property updates without restarts

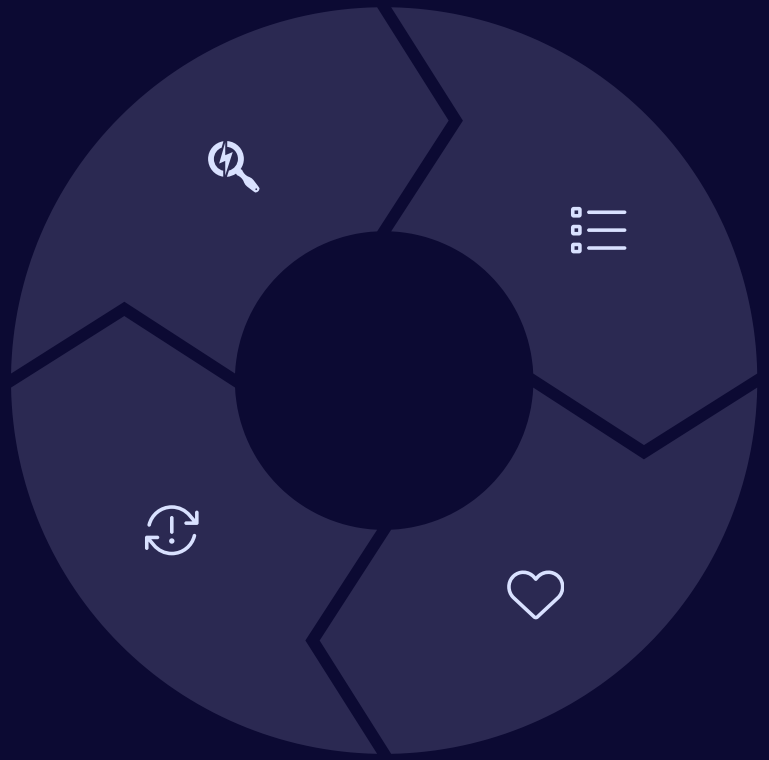
Service Discovery: Eureka, Consul, Zookeeper

Register

Services announce themselves

Discover

Find available service instances



Update

Maintain registry consistency

Monitor

Track health and availability

Client-Side Load Balancing: Ribbon & Spring Cloud LoadBalancer

Client Request

Application initiates service call

Load Balancer

Applies distribution strategy

Service Selection

Chooses optimal instance

Request Handling

Processes and returns response

Intelligent Routing: Spring Cloud Gateway & Zuul

Spring Cloud Gateway

- Built on WebFlux
- Non-blocking architecture
- Supports websockets
- Route predicates

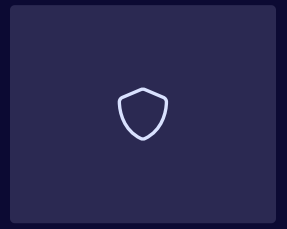
Netflix Zuul

- JVM-based router
- Synchronous architecture
- Request filtering
- Load shedding

Common Features

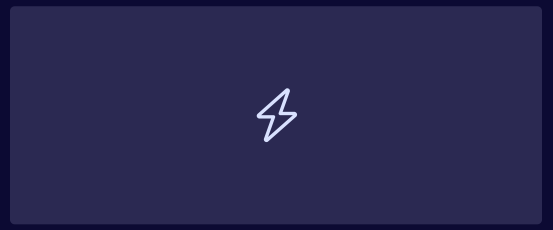
- Security
- Monitoring
- Dynamic routing
- Request transformation

Communication & Resilience: Feign, RestTemplate, Circuit Breakers



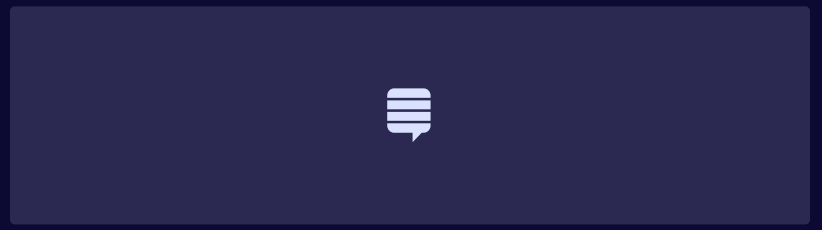
Resilience4j/Hystrix

Prevent cascading failures



WebClient

Reactive, non-blocking API calls

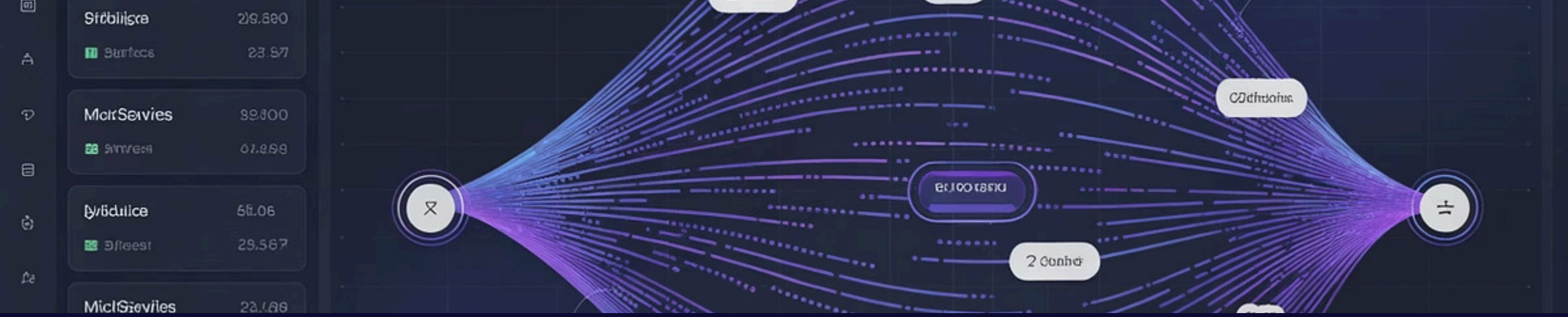


RestTemplate

Synchronous REST communication

Feign Client

Declarative REST clients



Distributed Messaging, Tracing, and Logging

Spring Cloud Stream

Integrates with Kafka/RabbitMQ for event-driven architecture and message processing.

Spring Cloud Sleuth

Adds trace IDs to log entries to track requests across multiple services.

Zipkin Integration

Visualizes request traces, identifies bottlenecks, and analyzes latency issues.

Centralized Monitoring

Consolidates logs and metrics for better system-wide visibility.

Benefits of Spring Cloud Components

Infrastructure Abstraction

Handles complex distributed system patterns without custom code.

Provides consistent APIs across different cloud environments.

Developer Focus

Reduces boilerplate code for common distributed patterns.

Allows teams to concentrate on core business logic.

Enterprise Ready

Built-in resilience, monitoring, and scaling capabilities.

Backed by Spring's active community and commercial support.