# Microservices + Spring Cloud Notes

SPRING CLOUD FEATURES FOR MICROSERVICES

# Agenda



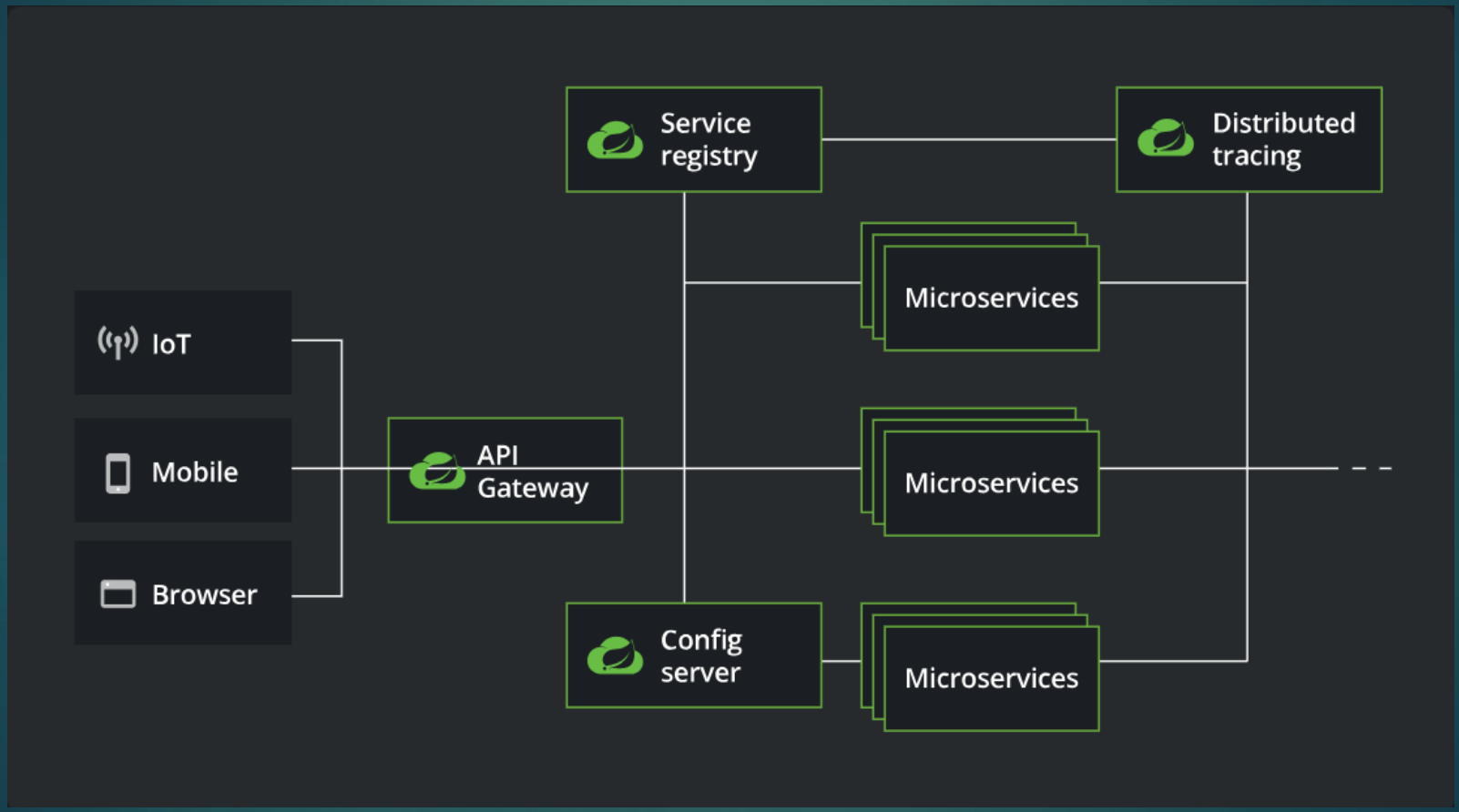
By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/

Concept explanation and Spring specific details for following:

- Service discovery
- Load balancing
- External Configuration
- Circuit Breakers
- Messaging
- Tracing
- Other important features

# Spring Microservices Architecture






By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/

# Need of Spring Cloud



- Distributed systems of microservices demand greater interaction between services.
- Making your code 'cloud-native' means dealing with **12-factor** issues such as external configuration, statelessness, logging, and connecting to backing services.
- The Spring Cloud suite of projects contains many of the services you need to make your applications run in the cloud.

By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/

# Service discovery - What is it ?



- In a cloud based environment, the IP and ports of the microservice instances can change anytime
- The number of instances of a service can change anytime
- In such scenario, there needs to be a service discovery mechanism that knows the location and count of instances of any microservice
- Microservice to Microservice calls can be done by making use of service discovery mechanism to locate the target microservice

By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/

# Load balancing – What is it ?



- Load balancer aims to distribute traffic to a microservice among its multiple instances so that no single instance gets overwhelmed by traffic
- Load balancer can make user of service discovery to find out about available service instances
- Load balancing can be done on client side or server side

By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/

# Spring - Service discovery and load balancing



By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/

Provides **DiscoveryClient** implementation for following service discovery tools

- **Eureka** – Eureka client and Eureka server support
- **Consul** - Instances can be registered with the Consul agent and clients can discover the instances using Spring-managed beans
- **Zookeeper** – Client side load balancing solution. Supports OpenFeign
- **Kubernates** - Client side loadbalancing via Netflix Ribbon. DiscoveryClient implementation.

# Spring - Service discovery and load balancing



By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/

Important to know annotation

- @EnableDiscoveryClient - Enables discovery client as per available dependency like Eureka, consul, zookeeper

- Update: This annotation is no longer needed to be added explicitly in the code. Having a DiscoveryClient implementation on your classpath will cause spring boot application to register itself to service discovery server

# External Configuration – What is it ?



- Store configuration of a microservice in an external location

This enables

- Run application in multiple environments without modification or recompilation
- Update application configuration without changing application code

By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/

# Spring – External configuration



By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/

Spring Cloud Config provides server and client-side support for externalized configuration in a distributed system. Spring supports integration to following external configuration tools

- Consul – Uses consul's key/value store
- Zookeeper – Use zookeeper as a data store
- Kubernates - PropertySource objects configured via ConfigMaps

**Important annotation**

- @RefreshScope – Reinitializes the bean when there is a configuration change

# Messaging – What is it ?



- Enables asynchronous communication between microservices through separate application managing the message queues
- Loose coupling due to asynchronous messaging
- Makes services scalable
- Improves microservices availability

By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/

# Spring cloud support for messaging and streaming



Spring cloud Bus and Spring cloud Stream projects provide messaging and streaming support for following:

- Kafka
- RabbitMQ
- Pulsar

By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/

# Circuit Breaker – What is it ?



- Circuit breakers gracefully degrade functionality when a method call fails
- Use of the Circuit Breaker pattern can allow a microservice to continue operating when a related service fails, preventing the failure from cascading and giving the failing service time to recover.

By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/

# Spring cloud support for circuit breakers



By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/

Provides an abstraction across different circuit breaker implementations.

Supported implementations:

- Retry
- Resiliance4j
- Sentinel

Note:- Hystrix is deprecated

Important annotations and classes:

- @EnableCircuitBreaker
- CircuitBreakerFactory – Use this to create a circuit breaker in your code

# Tracing – What is it ?



- Adds ability for tracing a request flowing through multiple microservices
- Tracing libraries instrument your application code so that each request have unique identifier that can be used to trace the request flow

By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/

# Spring support for tracing



- Supports Spring cloud sleuth with zipkin
- Adds trace and span ids to Slf4J MDC (Mapped Diagnostic Context) – This can be used to trace a request in a log aggregator
- Instruments common ingress and egress points from Spring applications (servlet filter, rest template, scheduled actions, message channels, feign client).

By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/

# Other important features



- Spring Actuator endpoints to view and manipulate the Environment
- TextEncryptor – Used to encrypt text data
- Spring cloud Task – Support for short lived microservice
- Cloud specific service support for
  - AWS
  - Azure
  - Alibaba
  - GCP

By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/

# Save this PDF for quick reference



- Source: Official Spring documentation
- Follow Amol Limaye to more see such content in your feed

https://www.linkedin.com/in/amolrlimaye/

By Amol Limaye:
https://www.linkedin.com/in/amolrlimaye/