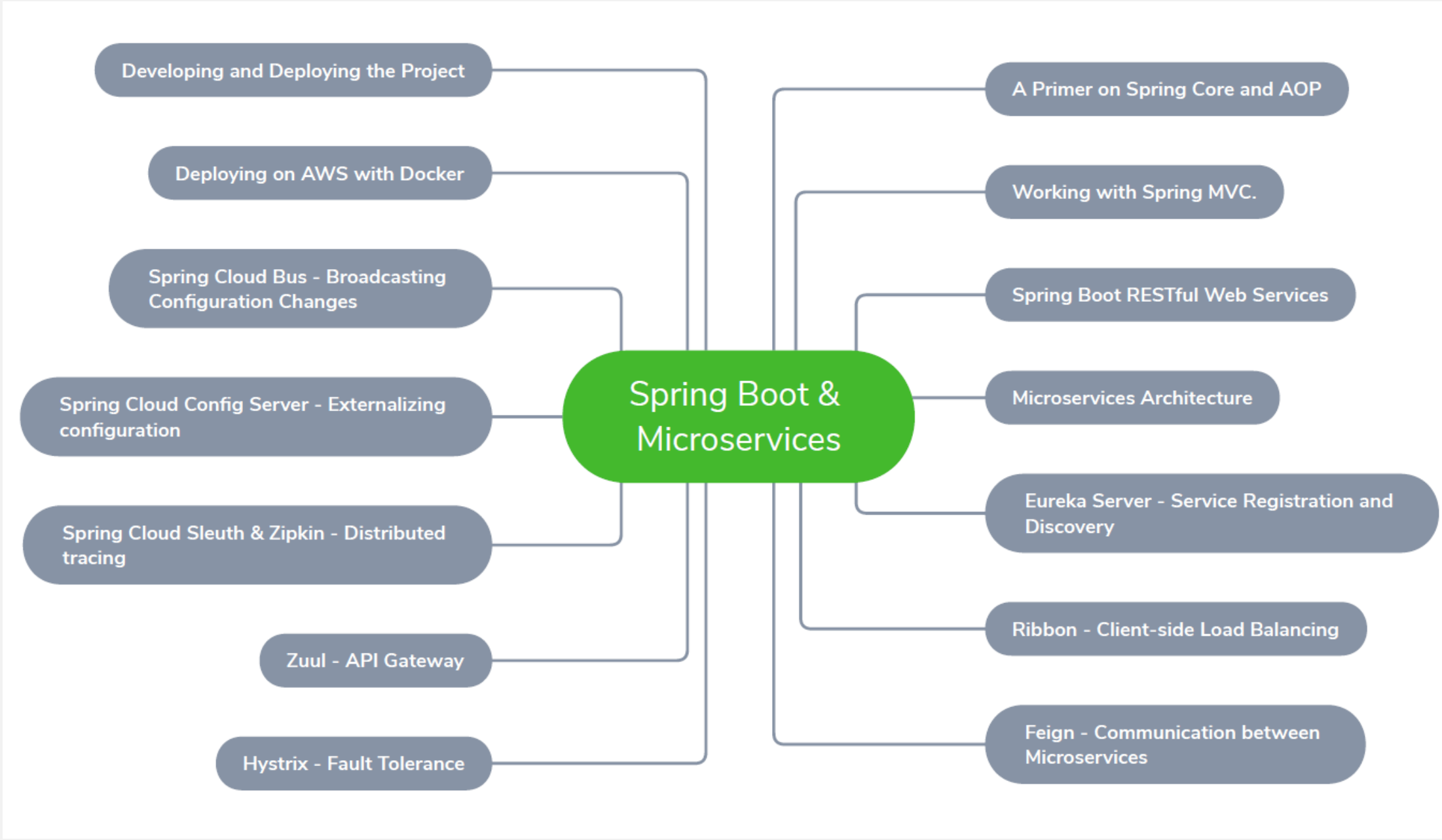
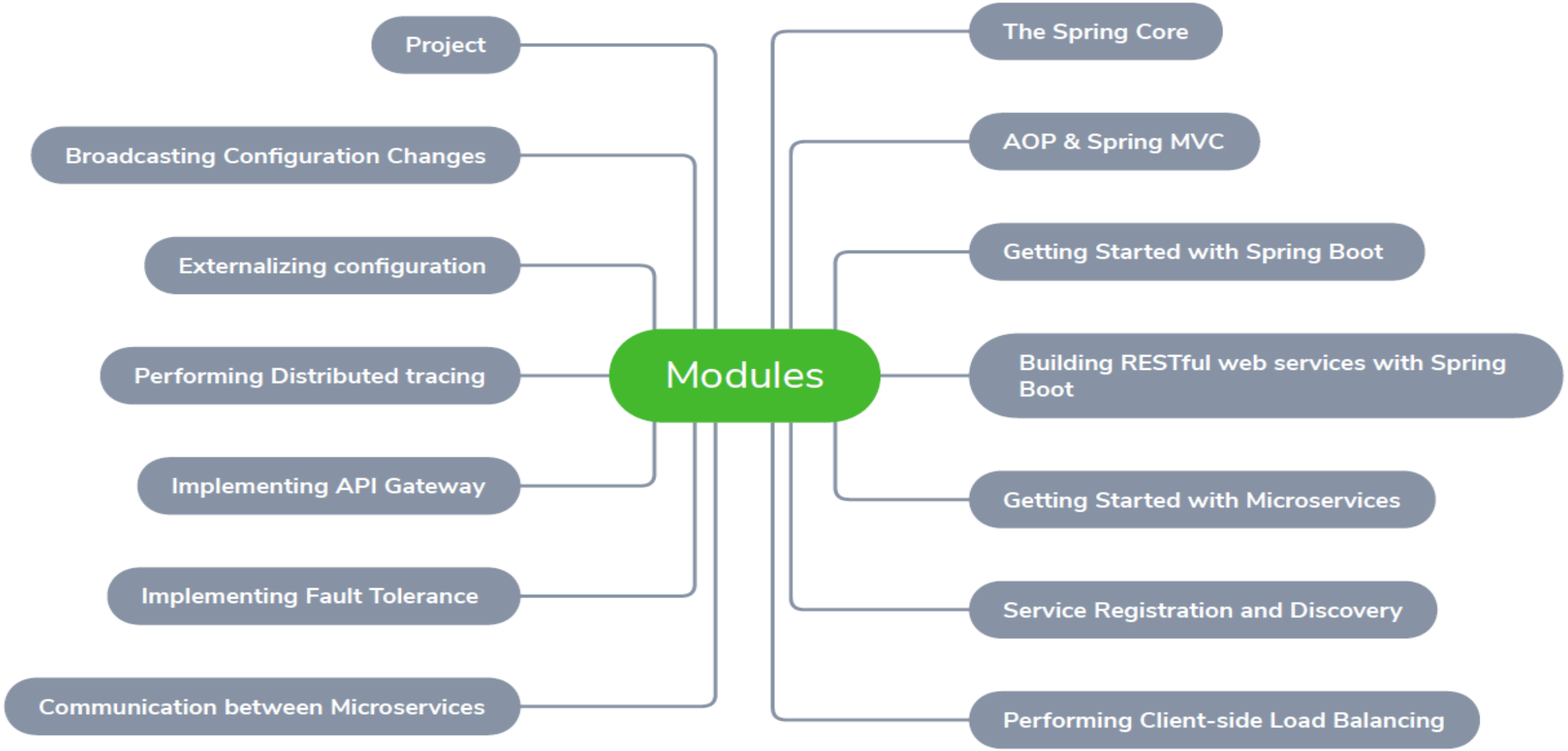
Cognixia

A Primer on Building & Coordinating Microservices

Spring Boot and Spring Cloud


Spring Boot & Microservices
Developing and Deploying the Project
Deploying on AWS with Docker
Spring Cloud Bus - Broadcasting Configuration Changes
Spring Cloud Config Server - Externalizing configuration
Spring Cloud Sleuth & Zipkin - Distributed tracing
Zuul - API Gateway
Hystrix - Fault Tolerance
A Primer on Spring Core and AOP
Working with Spring MVC.
Spring Boot RESTful Web Services
Microservices Architecture
Eureka Server - Service Registration and Discovery
Ribbon - Client-side Load Balancing
Feign - Communication between Microservices


Project
Broadcasting Configuration Changes
Externalizing configuration
Performing Distributed tracing
Implementing API Gateway
Implementing Fault Tolerance
Communication between Microservices
Modules
The Spring Core
AOP & Spring MVC
Getting Started with Spring Boot
Building RESTful web services with Spring Boot
Getting Started with Microservices
Service Registration and Discovery
Performing Client-side Load Balancing

CREATE STAND-ALONE, PRODUCTION-GRADE SPRING BASED APPLICATIONS THAT YOU CAN "JUST RUN".

Getting Started with Spring Boot

- An Opinionated Framework
- Enables Rapid Development
- Allows Creating Stand Alone Apps
- Ensures Productivity with Spring Boot Starters
- Provides Embedded Tomcat Server

MICROSERVICES

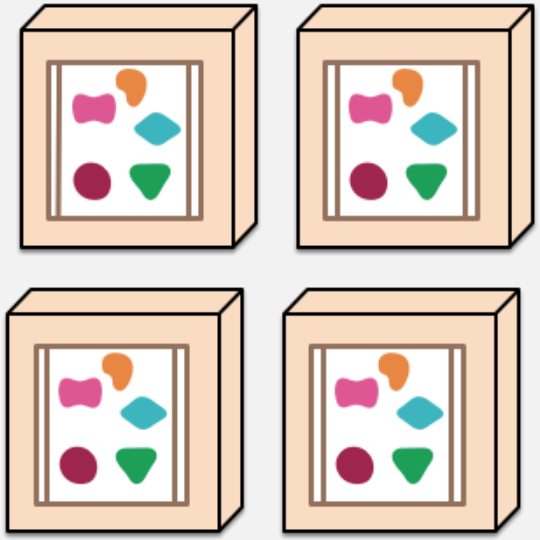
BUILDING SMALL,
SELF-CONTAINED,
READY TO RUN APPLICATIONS

GREAT FLEXIBILITY AND
ADDED RESILIENCE
TO YOUR CODE

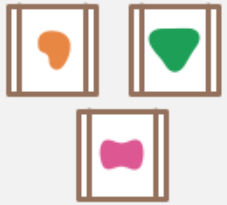
Monolithic vs Microservices Architecture

A monolithic application puts all its functionality into a single process...

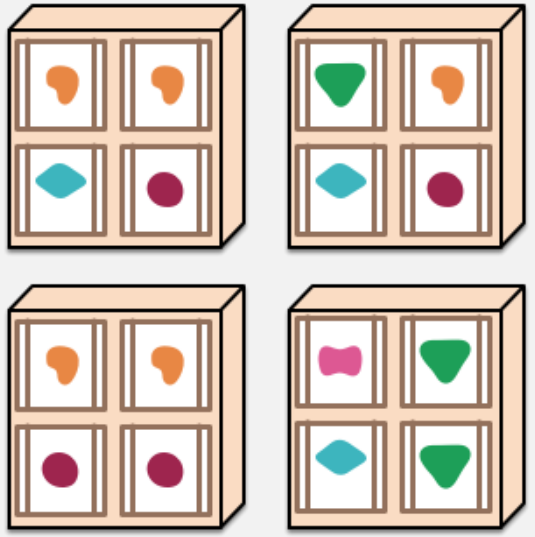
... and scales by replicating the monolith on multiple servers



A microservices architecture puts each element of functionality into a separate service...



... and scales by distributing these services across servers, replicating as needed.



Microservice Architectural Style

approach to developing a single application

as a **suite of small services**

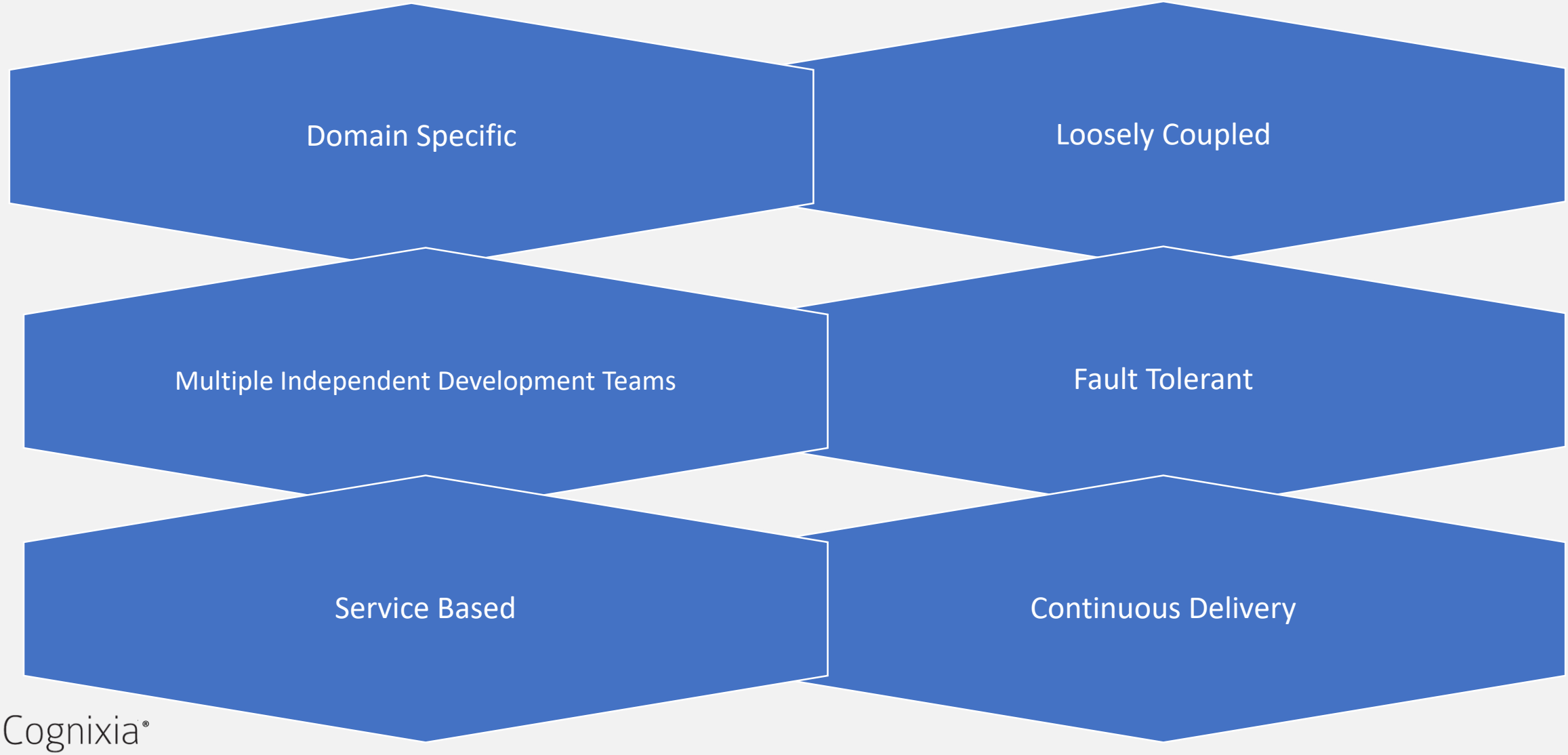
each **running in its own process** and

communicating with lightweight mechanisms (HTTP)

Microservice Architectural Style

- services are **built around business capabilities**
- **independently deployable**
- **bare minimum of centralized management**
- may be written in different programming languages
- use different data storage technologies.

Microservice Characteristics



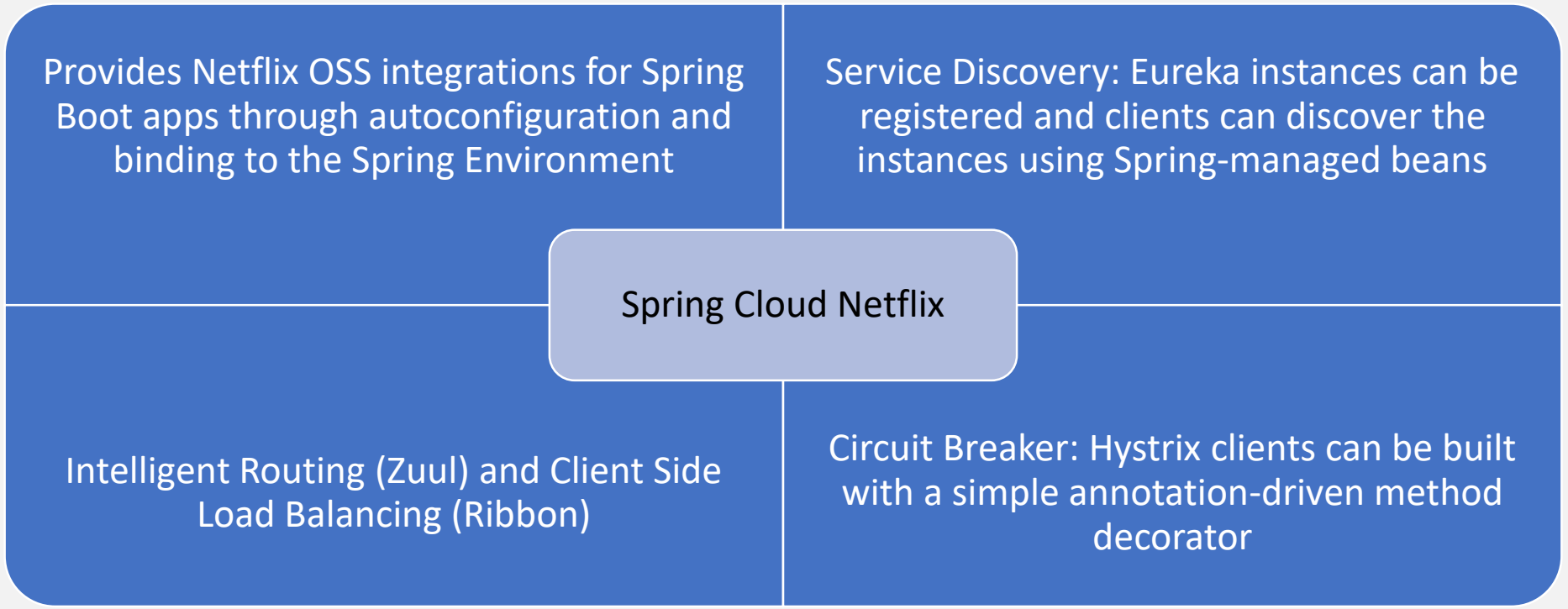
TOOLS FOR DEVELOPERS

TO QUICKLY BUILD SOME OF THE

COMMON PATTERNS IN DISTRIBUTED SYSTEMS

(E.G. CONFIGURATION MANAGEMENT, SERVICE DISCOVERY)

Spring Cloud Projects



Spring Cloud Projects

Spring Cloud Config

- Provides server and
- client-side support for
- externalized configuration in a
- distributed system.
- Backed up by Git

Spring Cloud Sleuth

- Provides Spring Boot auto-configuration for
- Distributed tracing

- DEVELOPING ENTERPRISE APPLICATIONS USING MICROSERVICES ARCHITECTURE
- BUILDING AND DEPLOYING CLOUD-BASED, SCALABLE, AND FAULT-TOLERANT APPLICATIONS
- CONTAINERIZING APPLICATIONS USING DOCKER
- DEPLOYING SPRING BOOT APPLICATIONS TO AWS CLOUD