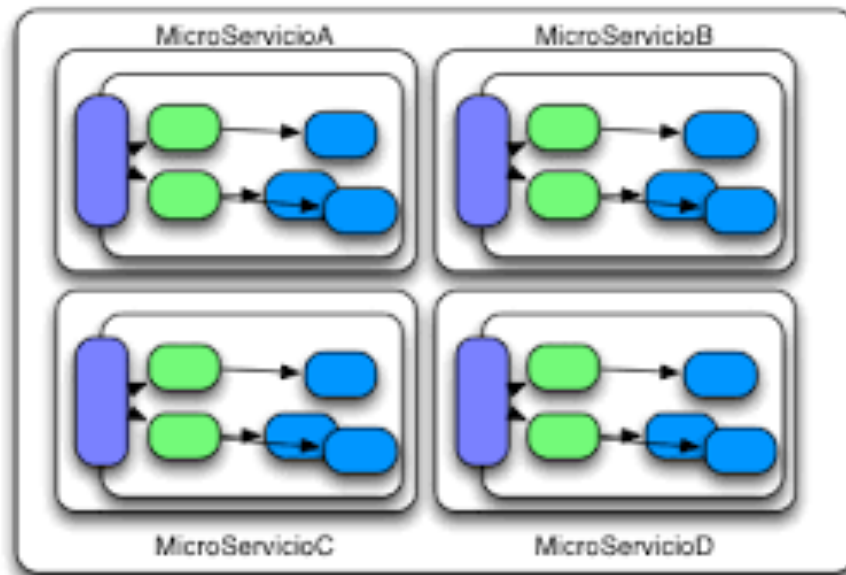


MICROSERVICIOS



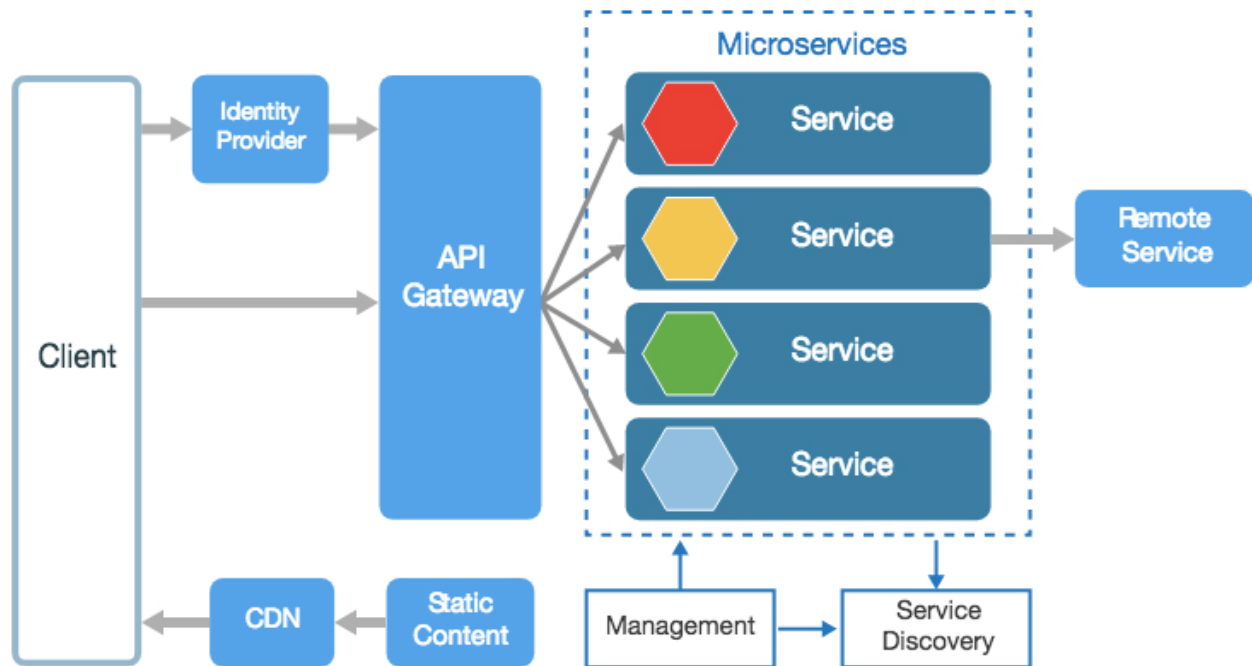
Introducción.

Los micro servicios son una arquitectura de construcción de software que genera aplicaciones estructurados en aplicaciones más pequeñas, autónomas y con un propósito muy específico. Dichas “mini-aplicaciones” se ejecutan por si mismas, y se comunican con otros microservicios mediante protocolos como HTTP/REST y cmensajería basada en colas. En el caso de Java, se usa también APIs Restful y Event-Driver Arquitectura (emisión y recepción de eventos).

Esto permite que el desarrollo de una plataforma completa pueda ser trabajado de manera paralela por diferentes equipos de trabajo, por lo que tiene un alto nivel de escalabilidad y facilidad para mantenimiento del software. Esta arquitectura está altamente relacionada con el cloud computing debido a su idoneidad para el entorno conocido normalmente como “nube”, pues este tipo de entorno está plenamente pensado en usar solo los recursos necesarios, por ende es necesario desfragmentar lo que en otros entornos en una aplicación completa, en muchas aplicaciones pequeñas para poder seleccionar que es útil al usuario/cliente y trabajar únicamente con ello.

Esta arquitectura requiere consideraciones de seguridad especiales, pues dado que cada servicio requiere comunicarse con otros es importante que la información llegue integra, por lo que se requiere de firewalls, políticas de acceso, autenticación y autorización.

Microservicios en Java



En Java, la arquitectura de microservicios es muy popular dada su aceptación en el mundo del desarrollo de software en general, pues es posible usar variedad de herramientas frameworks para construir cada una de las aplicaciones necesarias.

Para el desarrollo de microservicios en Java está ampliamente extendido el uso de 2 tecnologías llamadas Docker y Kubernetes, que contienen herramientas y plataformas adecuadas para la simplificación de la gestión de la infraestructura.

Ambos (Docker y Kubernetes) son “contenedores”, que proporciona un ambiente idóneo de ejecución del micro servicio que se quiere probar o implementar. Actúa similar a una máquina virtual, pero proporcionando los recursos necesarios y descartando aquellos que no lo son. Docker proporciona la herramienta en sí de los contenedores donde se llevará a cabo el desarrollo generalmente en entornos locales, mientras que Kubernetes orquesta estos contenedores para gestionar la implementación en producción en dichas aplicaciones.

Los principios del desarrollo consisten en Descomponer la Aplicación en componentes independientes, Separar Responsabilidades para que cada microservicio lleve a cabo sus funciones y Autonomía entre ellos, sin que eso signifique ausencia de comunicación, generalmente con protocolos.