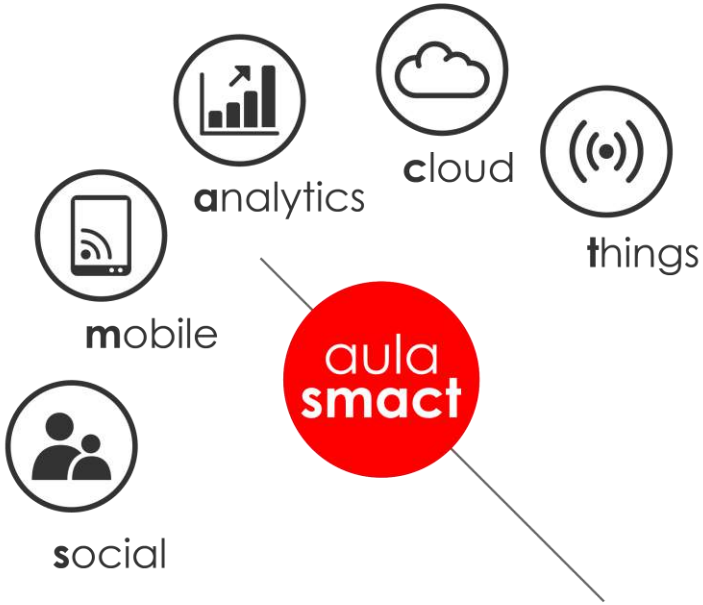
Contenedores y Microservicios

avanttic_day

Ángel Mogollón Gonzálz

Solution Specialist

14 de diciembre de 2020



Escuela Superior de Informática

UCLM
UNIVERSIDAD DE CASTILLA-LA MANCHA

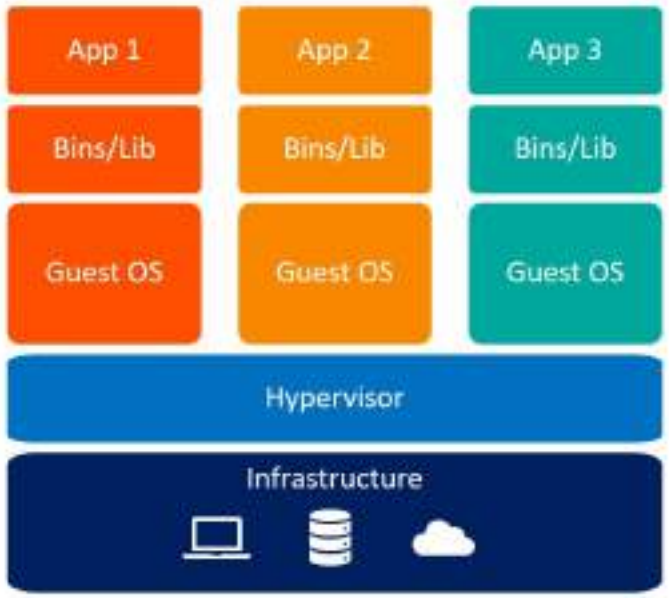
Contenedores

avanttic_day

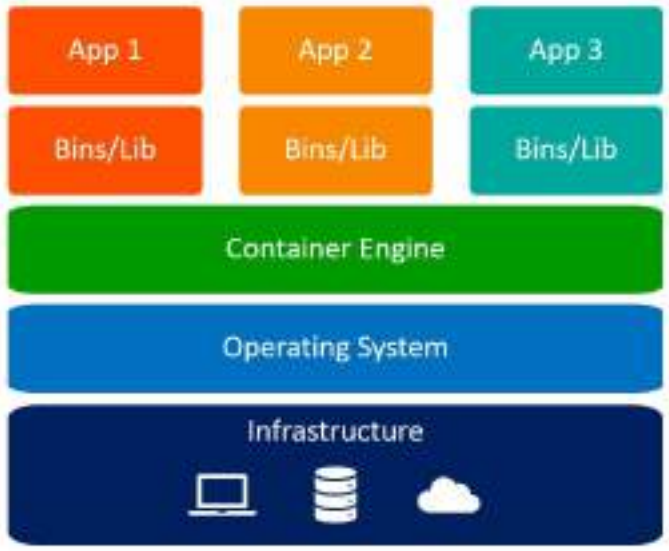
Contenedores y Microservicios

Contenedores

- Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema y código.



Virtual Machines



Containers

Orquestador de contenedores

avanttic_day

Orquestador de contenedores

- Es una herramienta para la automatización de despliegues, escalado y administración de contenedores.
- Kubernetes agrupa los contenedores que conforman una aplicación en unidades lógicas (pods) para una fácil administración, descubrimiento y escalado de las mismas.
- Kubernetes es código abierto, lo que le brinda la libertad de aprovechar infraestructura propia (on-premises), híbrida o de nube pública, lo que le permite mover las cargas de trabajo sin esfuerzo.

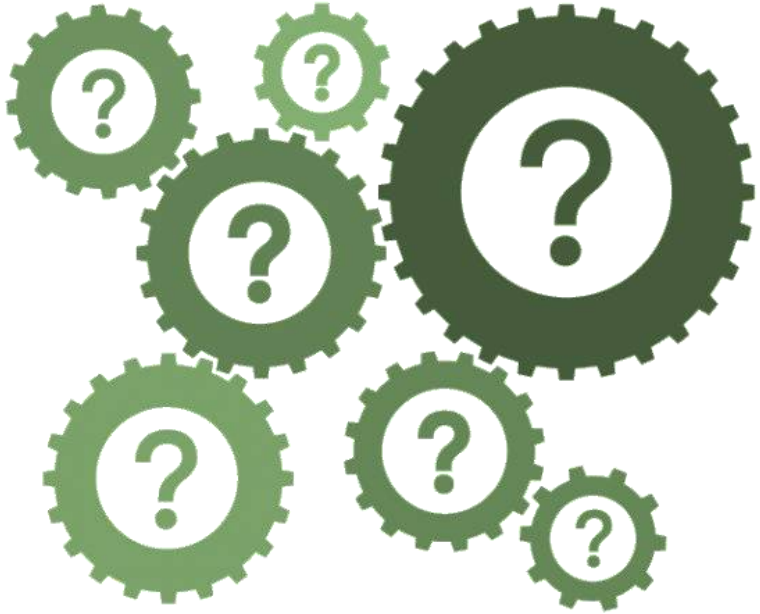
¿Qué son los microservicios?

avanttic_day

¿Qué son los microservicios?

¿Qué son los microservicios?

- Es un **método de desarrollo de aplicaciones** que funciona como un **conjunto de servicios que se ejecutan de manera independiente y autónoma, proporcionando una funcionalidad de negocio completa.** En ella, **cada microservicio es un código** que puede estar en un lenguaje de programación diferente, y **que desempeña una función específica**.

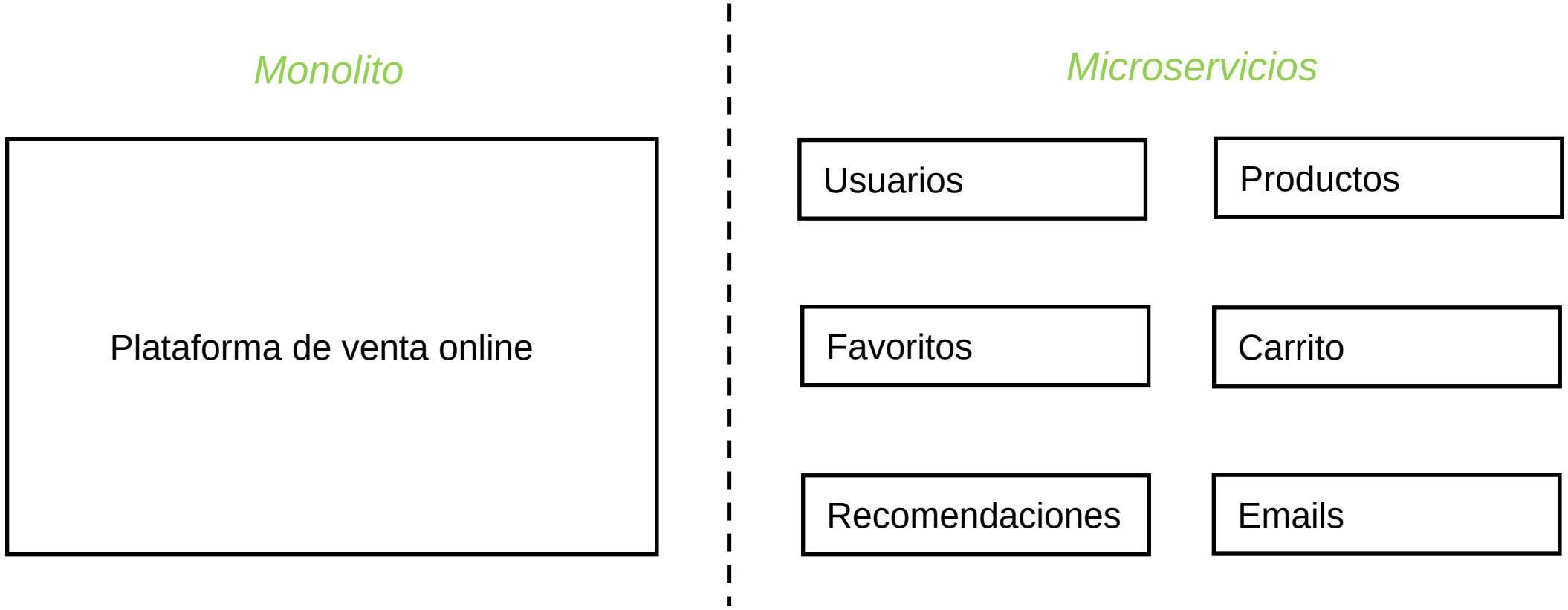


¿Qué diferencías hay con una arquitectura monolítica?

avanttic_day

Diferencias Monolito vs Microservicios

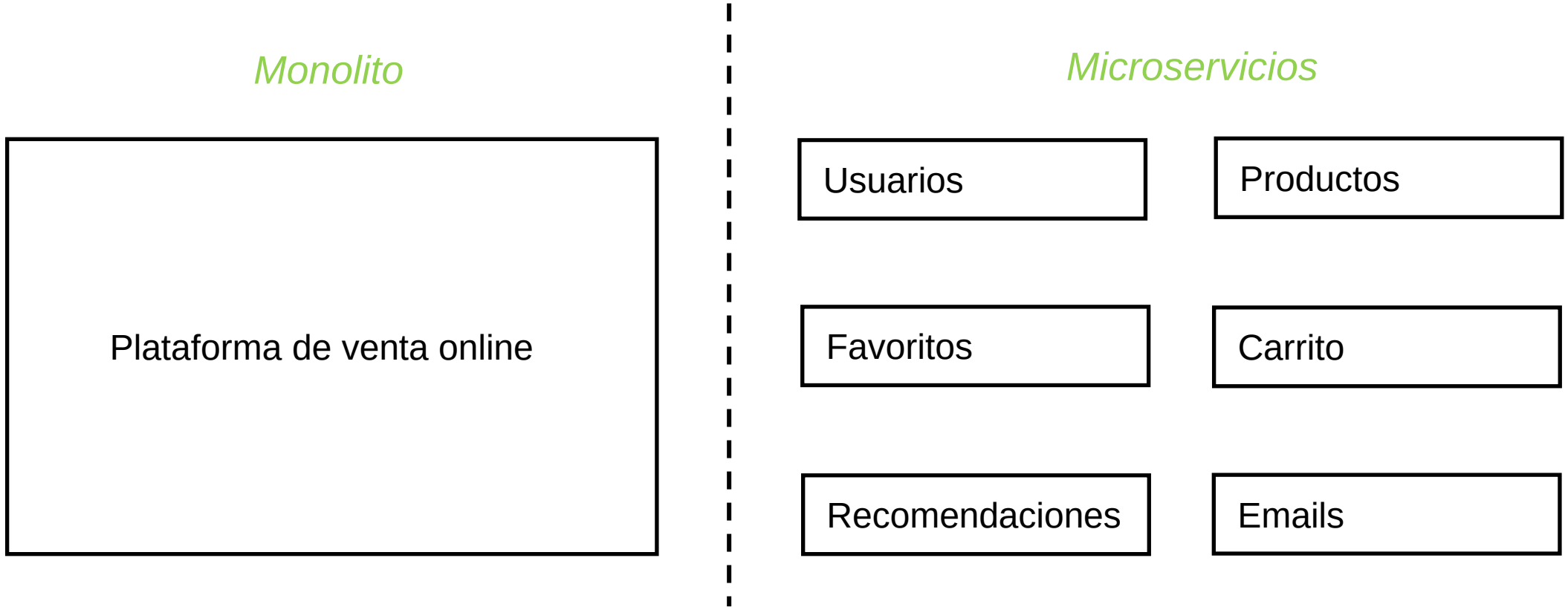
- Veamos como se implementaría una hipotética aplicación de venta online, primero en forma monolítica y luego en forma de microservicios:



Contenedores y Microservicios

Diferencias Monolito vs Microservicios

- La diferencia entre los dos sistemas es que el primero es una gran unidad, la segunda es un conjunto de servicios específicos con un papel funcional concreto.



Ventajas

avanttic_day

Contenedores y Microservicios

Ventajas

- **Modularidad**: al tratarse de servicios autónomos, se pueden desarrollar y desplegar de forma independiente. Además un error en un servicio no debería afectar la capacidad de otros servicios para seguir trabajando según lo previsto.

Contenedores y Microservicios

Ventajas

- **Escalabilidad**: como es una aplicación modular, se puede escalar horizontalmente cada parte según sea necesario, aumentando el escalado de los módulos que tengan un procesamiento más intensivo.

Ventajas

- **Versatilidad**: se pueden usar diferentes tecnologías y lenguajes de programación. Lo que permite adaptar cada funcionalidad a la tecnología más adecuada y rentable.

Contenedores y Microservicios

Ventajas

- **Rapidez de actuación**: Una vez tenemos lista la infraestructura, el reducido tamaño de los microservicios permite un desarrollo menos costoso, así como el uso de “contenedores” permite que el despliegue de la aplicación se pueda llevar a cabo rápidamente.

Retos

avanttic_day

Contenedores y Microservicios

Retos

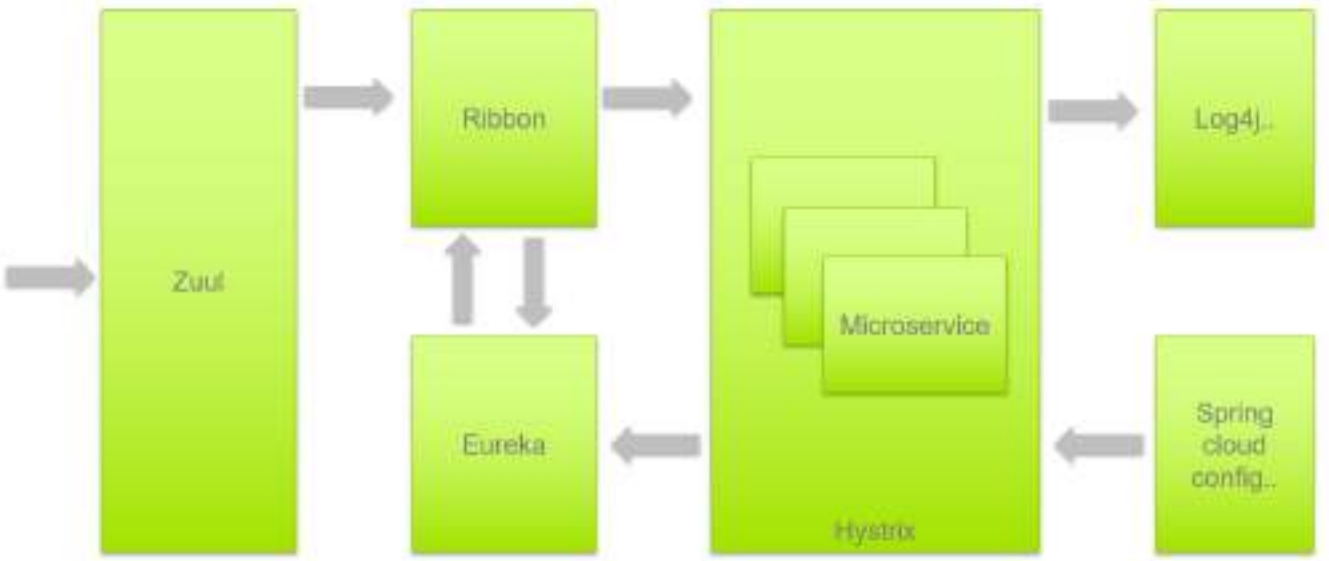
- **Alto consumo de recursos**: al tener cada microservicio sus propios recursos y bases de datos, consumen más memoria y CPU.
- **Inversión de tiempo inicial**: al crear la arquitectura, se necesita más tiempo para poder fragmentar los distintos microservicios.
- **Complejidad en la gestión**: si contamos con un gran número de microservicios, será más complicado controlar la gestión e integración de los mismos, aunque esto lo suplimos con orquestadores.
- **Tráfico interno**: al separar la aplicación en varias piezas se genera tráfico que antes no teníamos al comunicarse entre ellas.
- **Perfil de desarrollador**: los microservicios requieren desarrolladores experimentados con conocimientos en DevOps.

Evolución de los microservicios

avanttic_day

Evolución de los microservicios

- Con el paso del tiempo la arquitectura ha ido madurando, al principio alcanzó mucha popularidad el stack de Netflix adoptado por Spring como su principal solución para Cloud, donde las tareas de balanceo, configuración, registro, gestión de errores recaen en la propia aplicación a través de los diferentes microservicios que a parte tienen que ser desplegados.

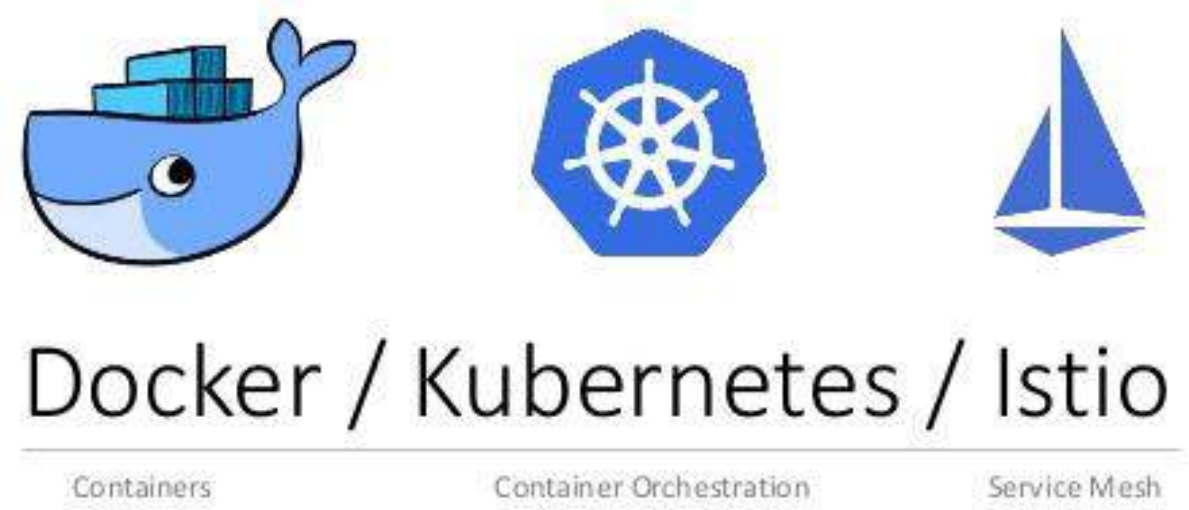


Evolución de los microservicios

- Al poco tiempo, aparece Kubernetes y se traslada la funcionalidad que veíamos anteriormente (balanceo, registro, configuración…) a las plataformas que nos ofrecen soluciones gestionadas de Kubernetes(OKE, AWS ,Kubernetes Engine de Google).

Evolución de los microservicios

- Por último, aparecen los Service Mesh, que nos permiten abstraer la funcionalidad de balanceo, registro, configuración…, tanto de la aplicación como del orquestador.



Istio

- Se trata de una plataforma para hacer ***server mesh***, es decir, añadir componentes a nuestros servicios (microservicios, en este caso) sin tener que modificar ni tocar nada en ellos.

- **Proxy Envoy**: Es la pieza encargada de interceptar todo el tráfico de entrada y salida al contenedor de la aplicación, se comunica con el resto de piezas para llevar a cabo funcionalidades como son el balanceo, circuit breaking, gestión de timeouts…

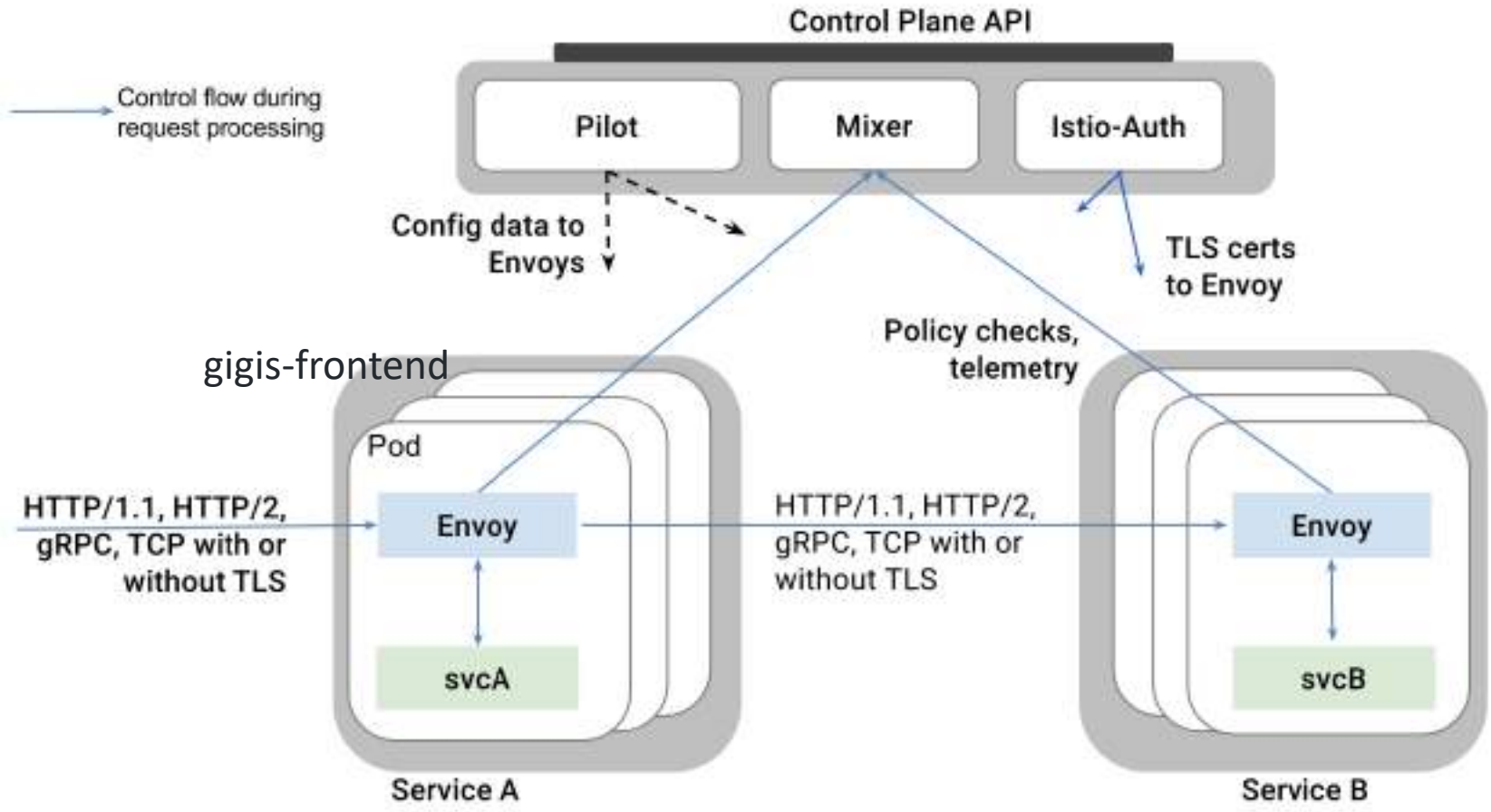
- **Mixer**: Se encarga del control de acceso y de recibir las métricas generadas por las llamadas en los proxy Envoy su finalidad es mover la integración con el control de acceso, gestión de cuota, logging y similares fuera del propio servicio (aplicación)

- **Pilot**: Es la pieza encargada de gestionar todo lo referente a registro de servicios, enrutado y control de errores (circuit breaking, timeouts, reintentos…)

- **Istio-Auth**:Este último componente, permite comprobar credenciales, utilizar comunicación TLS mutua, comprobación de identidad, etc.

Contenedores y Microservicios

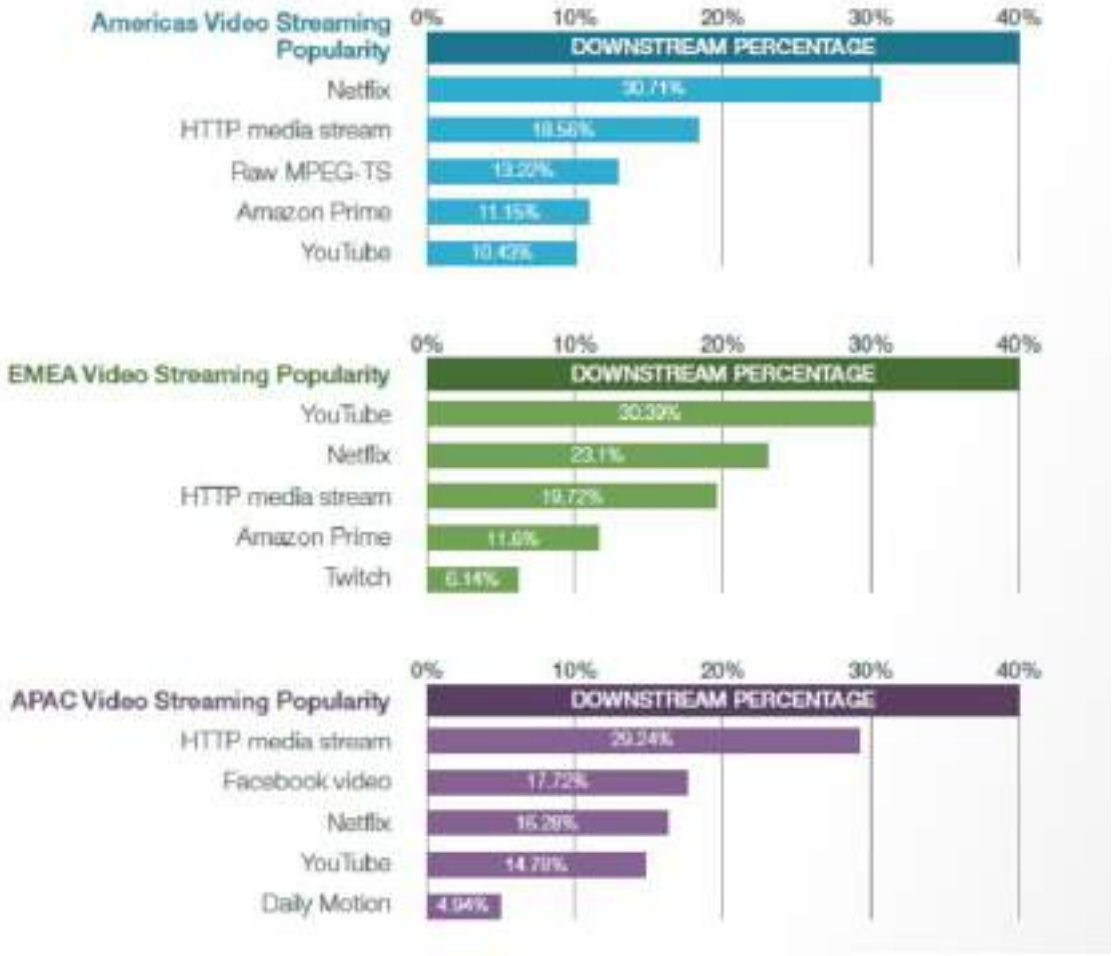
Istio



Contenedores y Microservicios

Caso de uso

NETFLIX

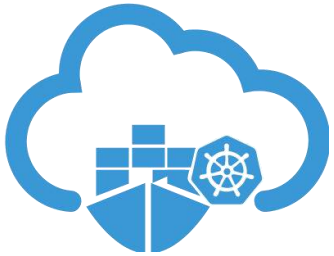


Demo

avanttic_day

Contenedores y Microservicios

Oracle Cloud Native Services



Oracle Cloud Infrastructure Registry

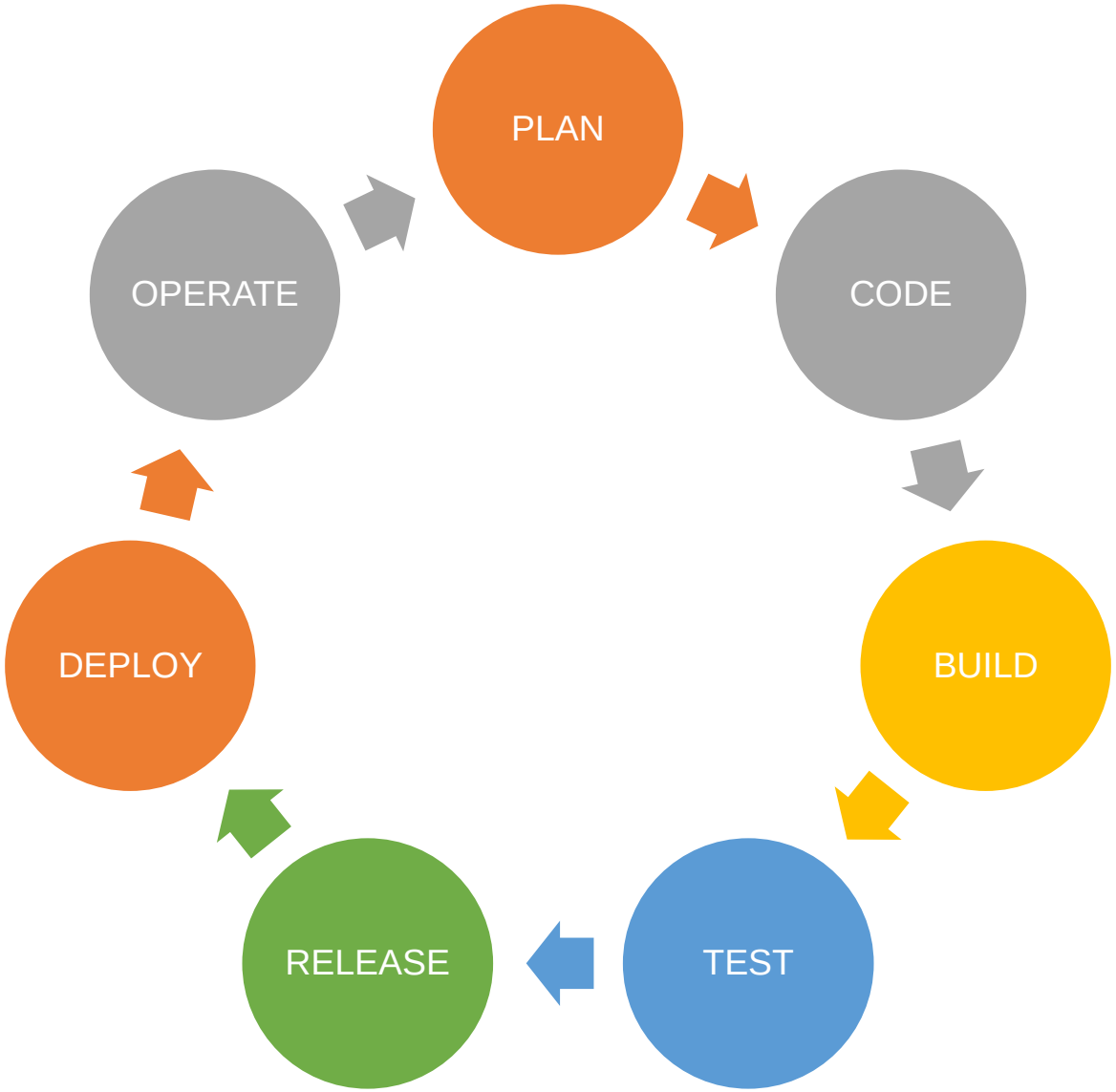
kubernetes

ORACLE

Oracle Kubernetes Engine

Contenedores y Microservicios

Ciclo de vida



Contenedores y Microservicios

Ciclo de vida – Oracle Visual Builder Studio

Control del código

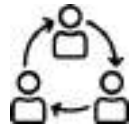
- Repositorio GIT
- Repositorio Maven
- Control de ramas, commits..etc
- Revisión de código

Integración continua

- Lanzar y trackear jobs
- Desplegar en cualquier sitio
- Lanzar pipelines

Control de incidencias

- Crear tareas, defectos o nuevas funcionalidades
- Estimar y asignar tareas

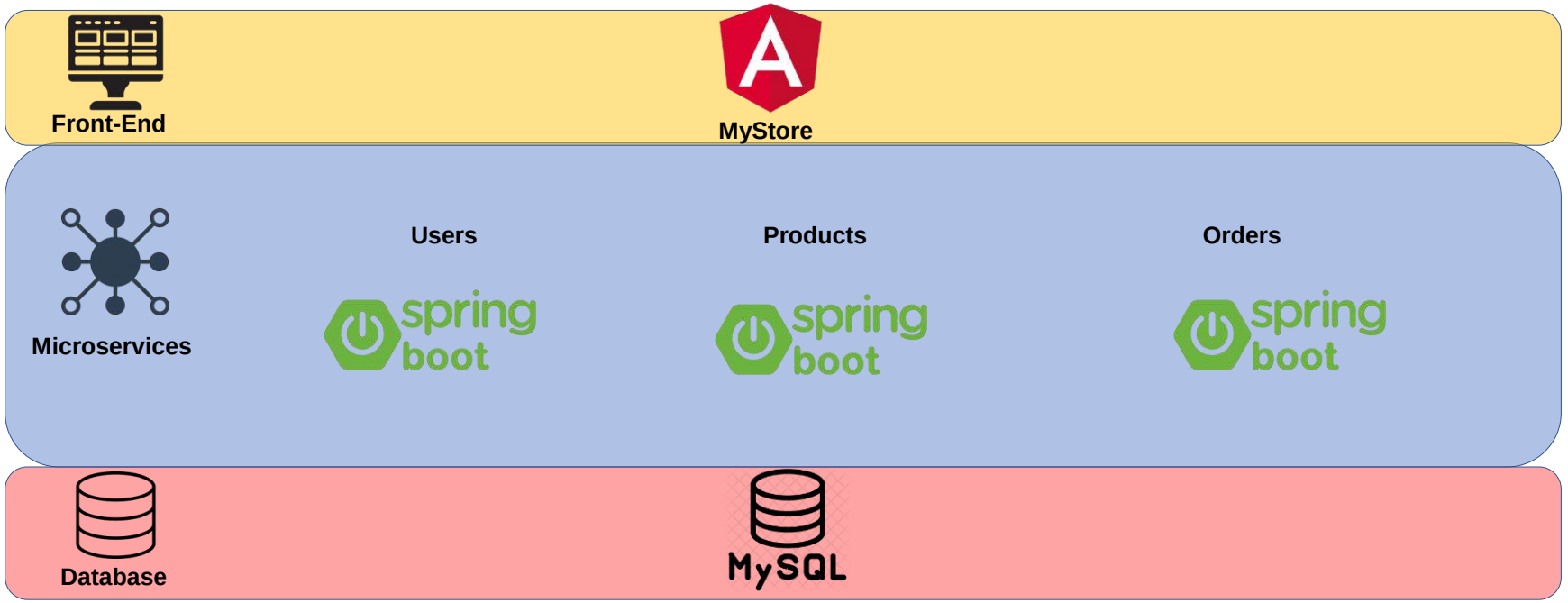


Colaboración

- Soporte a proyectos Agile (SCRUM)
- Wiki

Contenedores y Microservicios

Arquitectura de la demo



avanttic el partner

Ángel Mogollón González

Solution Specialist

angel.mogollon@avanttic.com

Escuela Superior de Informática

UCLM
UNIVERSIDAD DE CASTILLA-LA MANCHA