



Serverless

Kevin Barbón García UO263779

Thalía Cuetos Fernández UO264545

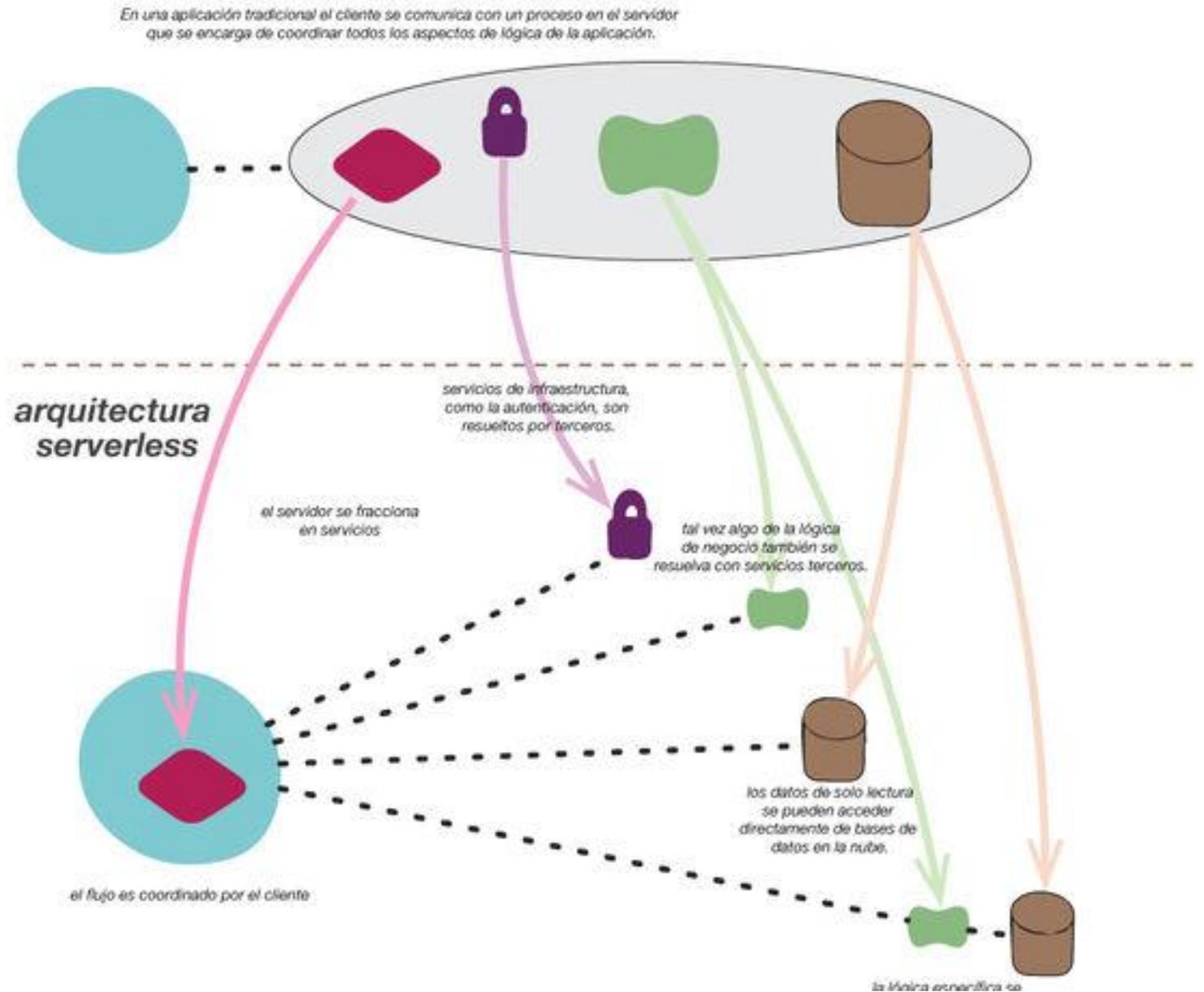
Antecedentes

- ¿Qué tamaño de servidores son adecuados?
- ¿Qué sistema operativo deberían tener?
- ¿Cuándo debería decidir aumentar mis servidores?
- ¿Cómo soportará la aplicación fallos en el hardware?
- ¿Cómo puedo controlar su acceso?
- ¿Quién hará los parches del SO?
- ¿Qué usuarios deben tener acceso?

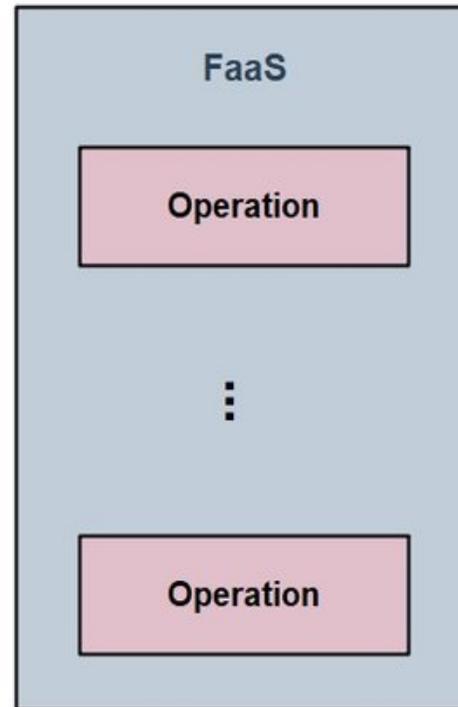
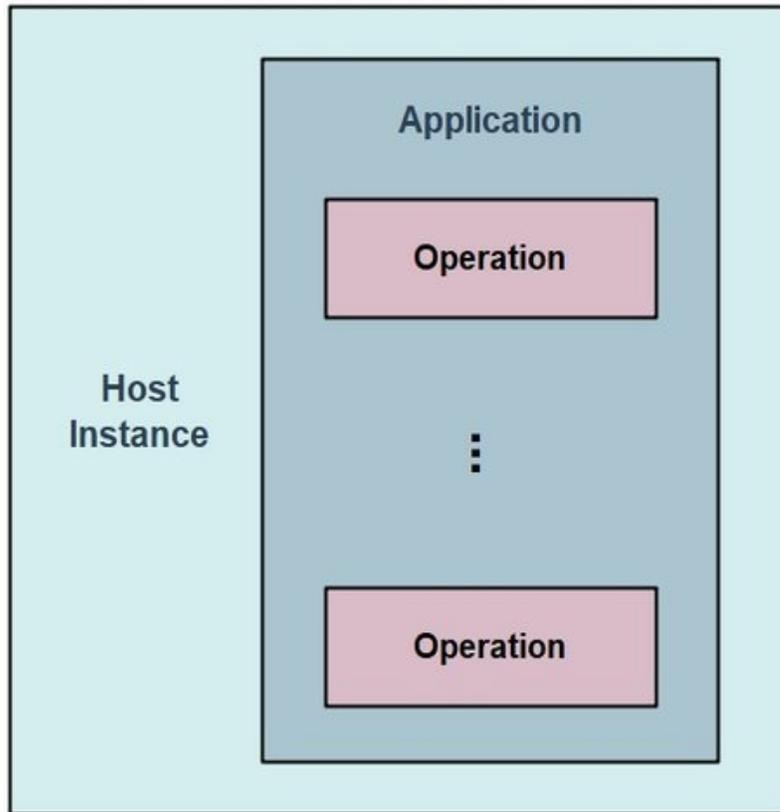
Arquitectura Serverless ("sin servidor")

Elimina la necesidad de manejar, aprovisionar y escalar servidores y bases de datos

FaaS
BaaS



¿Qué es el modelo FaaS?



- Funciones como servicio
- Contenedores temporales y sin estado
- Coste por tiempo de ejecución
- Escalado automático

¿Qué es el modelo BaaS?

Backend como servicio

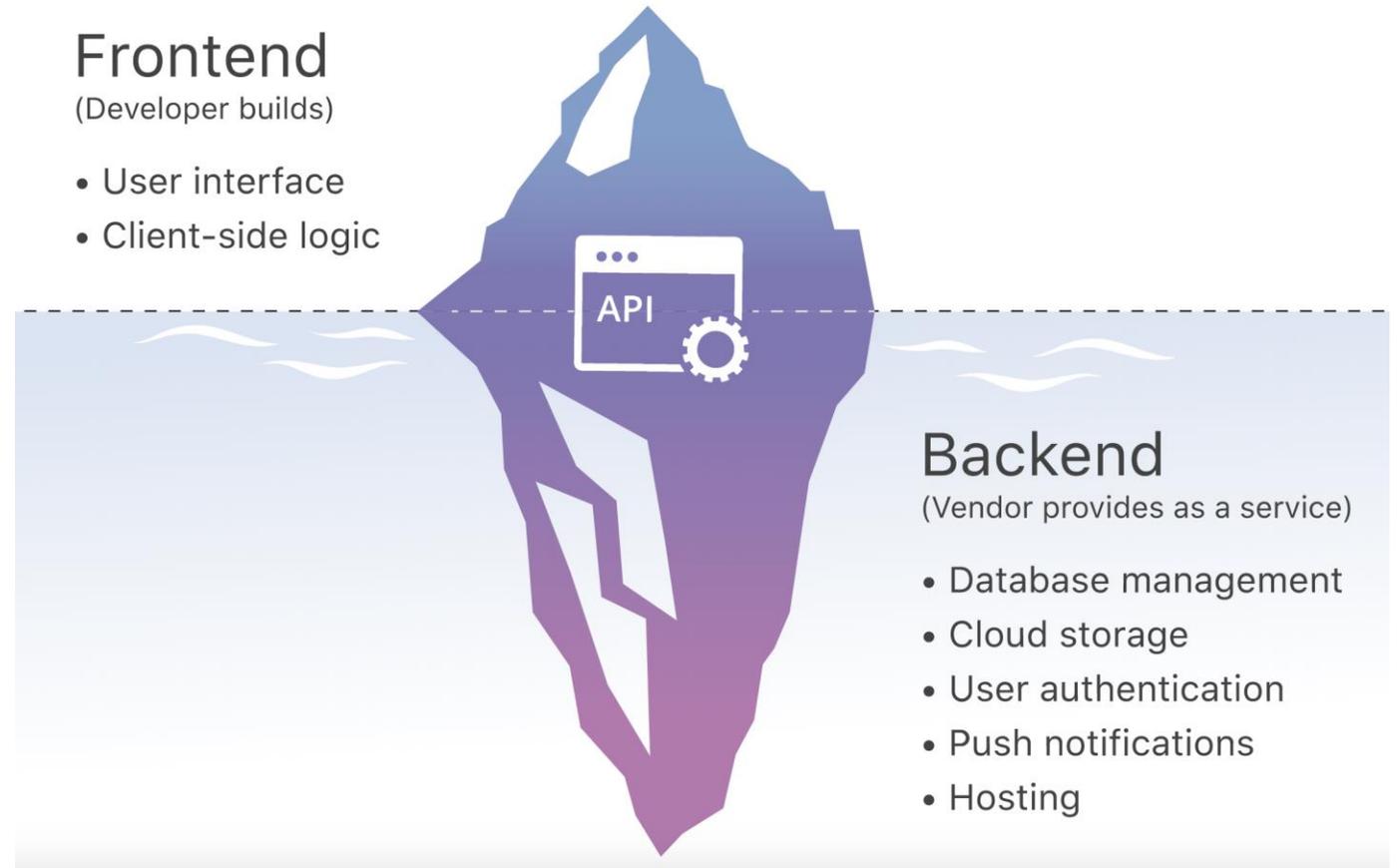
API propia

Modelo de pago *freemium*

Frontend

(Developer builds)

- User interface
- Client-side logic



Backend

(Vendor provides as a service)

- Database management
- Cloud storage
- User authentication
- Push notifications
- Hosting

Principales proveedores



AWS Lambda



Google
Cloud
Functions

Ventajas

Costo operativo reducido

Costo de desarrollo reducido

Costos de escalado

Gestión operativa más sencilla

Desventajas

- Alojamiento compartido
- Dependencias del proveedor
- Pérdida de optimizaciones del servidor
- Contenedores sin estado
- Latencia de inicio y "arranques en frío"
- Poco monitoreo

¿Cuándo se puede usar?

- Backends para móviles
- Backends de aplicaciones web
- Procesamiento de datos
- Ejecutar tareas programadas
- IoT

El futuro de serverless

- Mitigar los inconvenientes
- Arquitecturas híbridas
- Crecimiento de la comunidad



Conclusión

- No es el enfoque correcto para todos los problemas
- Costos operativos y de desarrollo reducidos
- Administración operativa más sencilla
- Menor tiempo de creación de nuevos componentes de aplicaciones.