



*Innovación para la  
transformación digital.*

# NETFLIX

*Netflix, hizo que ya casi no queden videoclubes*



# *Arquitectura de referencia para nube híbrida*



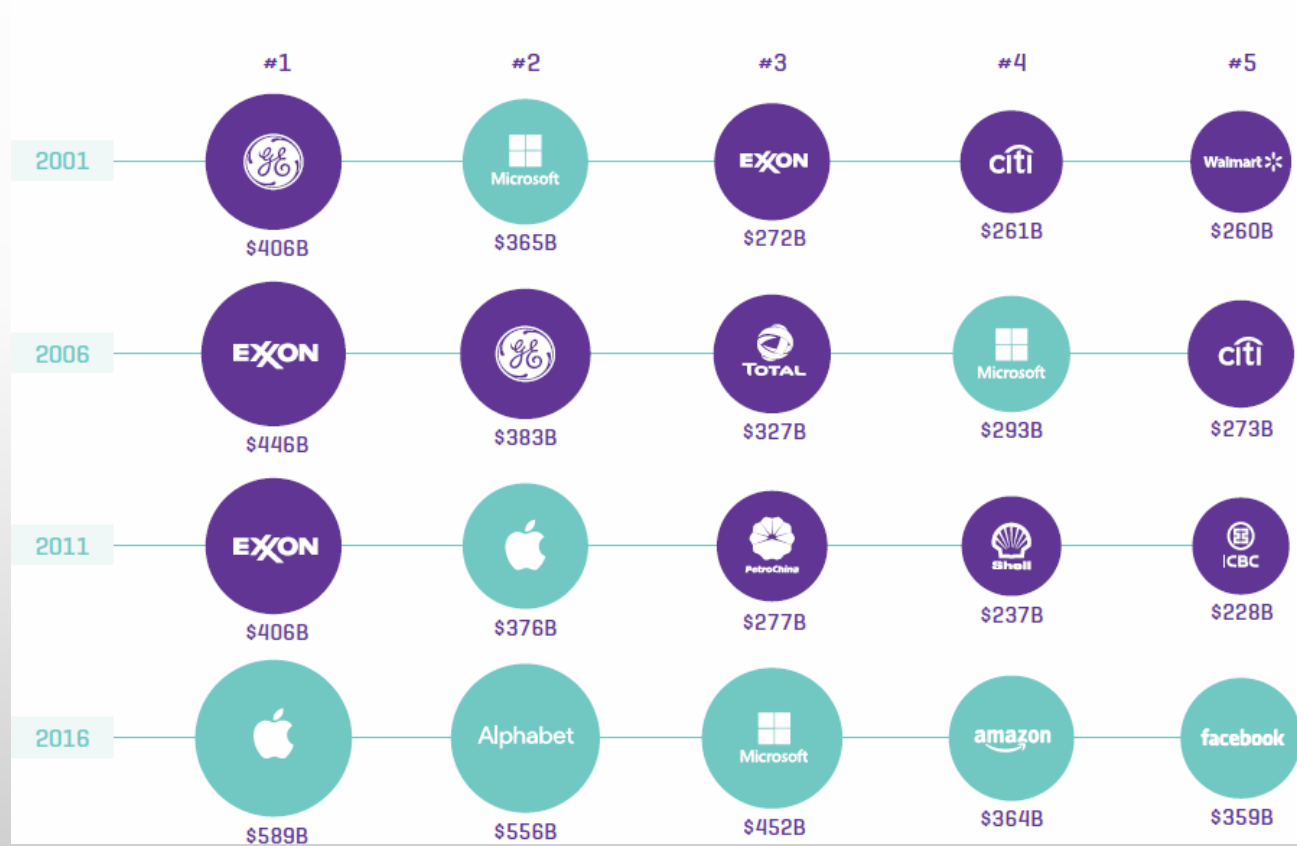
# *Web APIs*



# Empresas por capitalización de mercado.

The oil barons have been replaced by the whiz kids of Silicon Valley

● Tech ● Other



## *APIs - Enfoque*

- La arquitectura orientada a servicios (SOA) se refiere a ahorro de costos, NO a generar nuevos ingresos.
- *El enfoque de APIs no está en el reuso y ahorro de costos que había en SOA.*
  - *Más bien, el foco está en la consumibilidad y la competencia en la economía de APIs. Las APIs son un producto vendible.*

## *Características de las APIs*

- APIs necesitan portales que permitan que los desarrolladores descubran y experimenten con las APIs
- Deben existir mecanismos para que los desarrolladores se registren para usar y pagar por los APIs
- Los proveedores de APIs necesitan contar con planes de pagos que se ajusten al uso de los APIs
- Dado que los APIs están expuestos públicamente, el gateway que los expone debe tener capacidades robustas de seguridad.
- Todo lo anterior debe ser autoservicio y simple.... API Management



# *Evolución de arquitectura SOA → APIs & Microservicios*

# Avance progresivo de exposición de funciones de negocio / APIs

**MAINFRAME & MINI COMPUTING**  
1960s-1980s

APIs: COBOL, VT100



**CLIENT-SERVER & DISTRIBUTED COMPUTING**  
1980s-1990s

APIs: CORBA, SQL



**WEB & E-COMMERCE**  
1990s-2000s

APIs: WEB SERVICES



**CLOUD, SOCIAL, MOBILE**  
2000s-2010s

APIs: REST, JSON



**IOT / INDUSTRIAL INTERNET**  
2010s +

APIs: SENSORTHINGS, ...



# Evolución en la construcción de soluciones

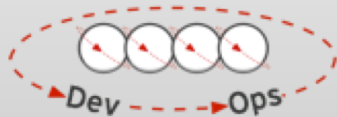
## Development Process

Waterfall



Agile

**DevOps**



## Application Architecture

Monolithic



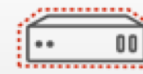
N-Tier

**Microservices**



## Deployment & Packaging

Physical Servers



Virtual Servers

**Containers**



## Application Infrastructure

Datacenter



Hosted

**Cloud**





# *Microservicios*



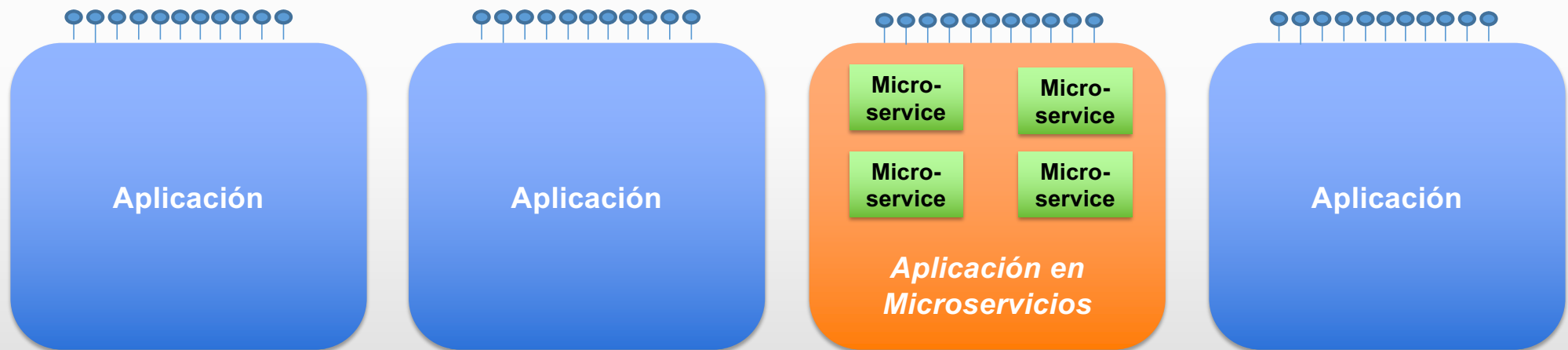
## *Qué es arquitectura de microservicios (MSA)?*

- MSA es una alternativa de arquitectura
- Arquitectura orientada a que las operaciones, funcionalidades son realizadas por conjuntos de servicios autónomos e independientes.
- Los microservicios son mini aplicaciones que cumplen cada uno su propio ciclo de vida y despliegue
- El prefijo “micro” se refiere la granularidad de sus componentes internos y no a la granularidad de su interfaz expuesta



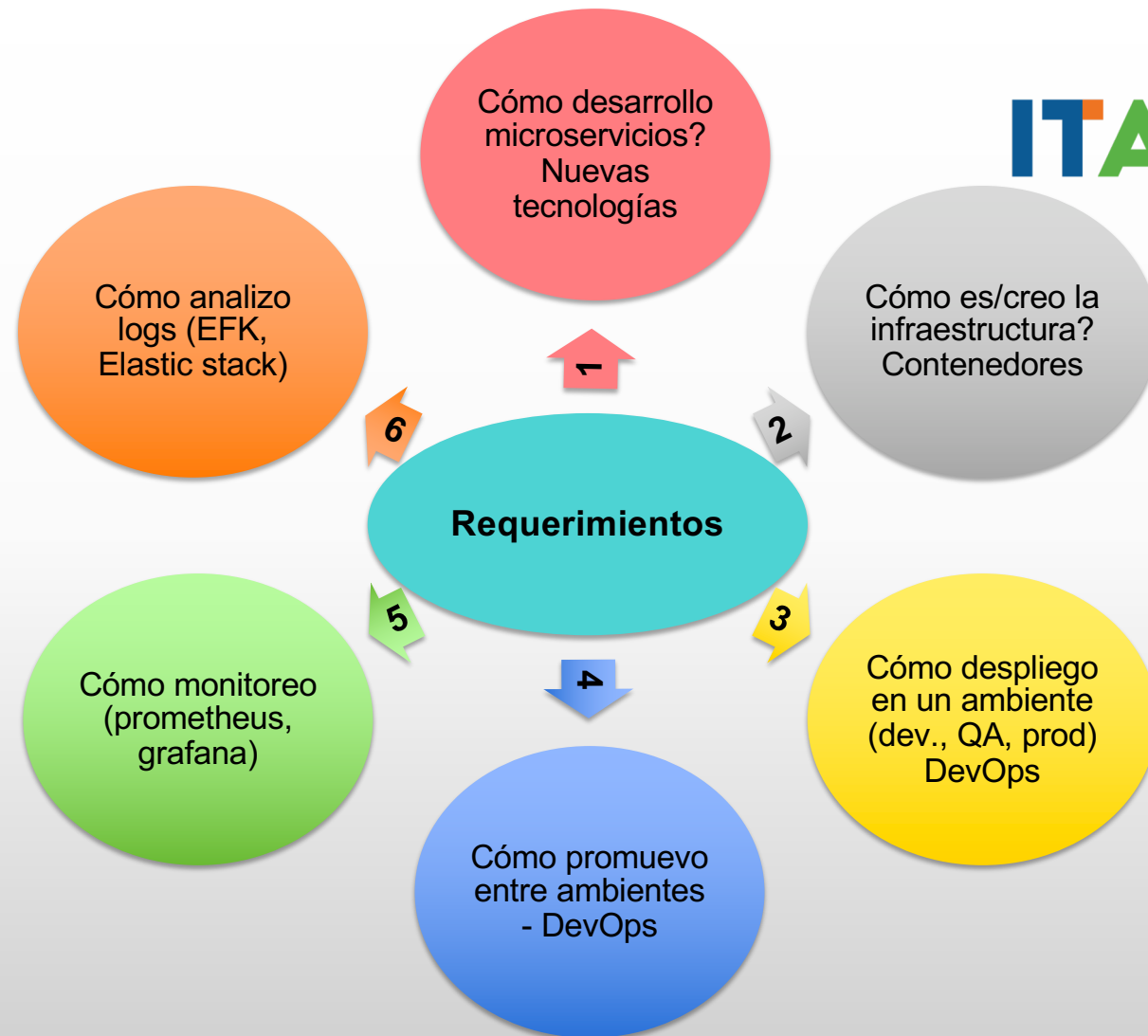
# Diferencias entre una arquitectura de microservicios y SOA

SOA se refiere a exposición de servicios a nivel empresarial

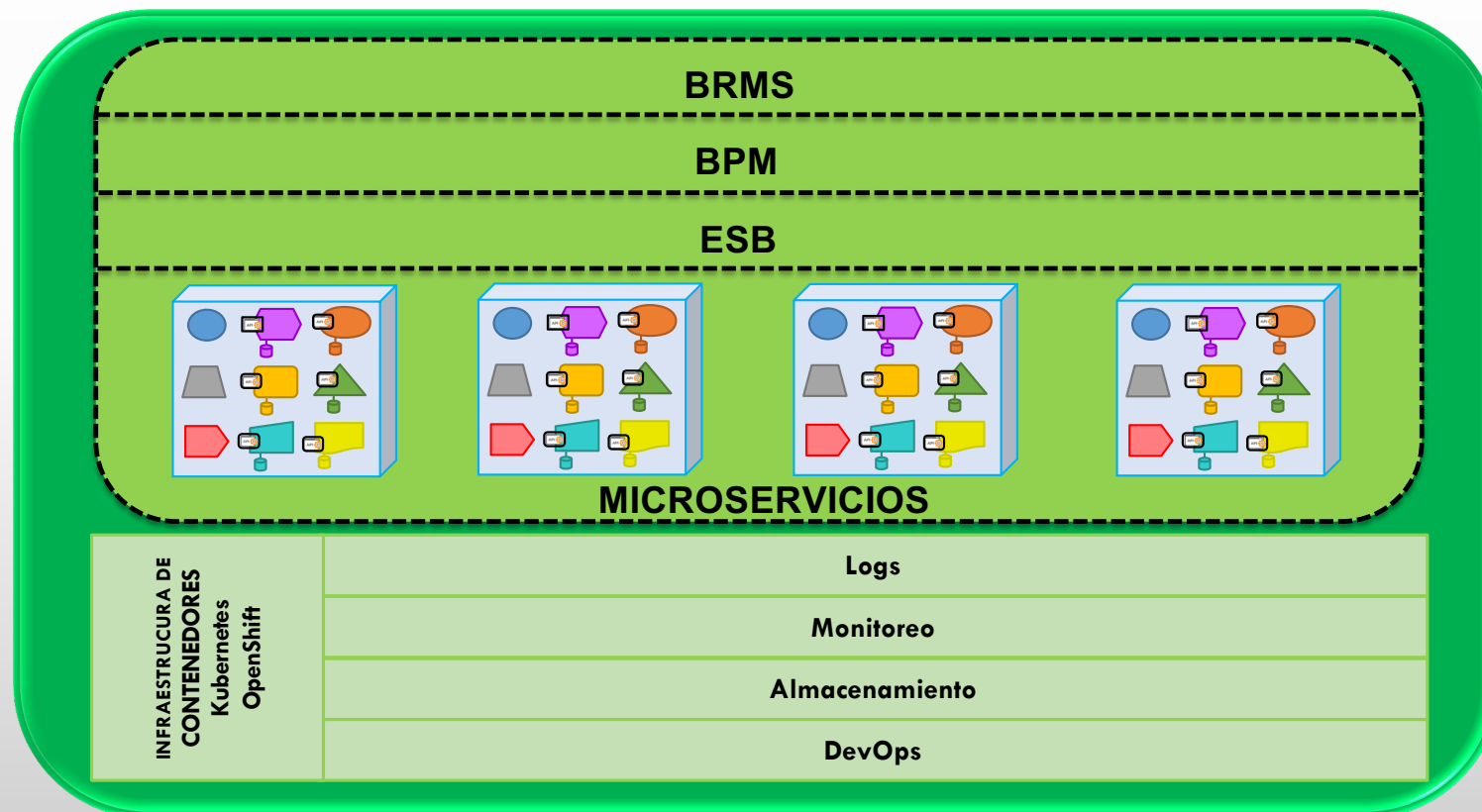


Microservicios se refiere a **arquitectura de aplicación**

# Requerimientos para implementar Microservicios



# Cómo es/creo la infraestructura de Contenedores?

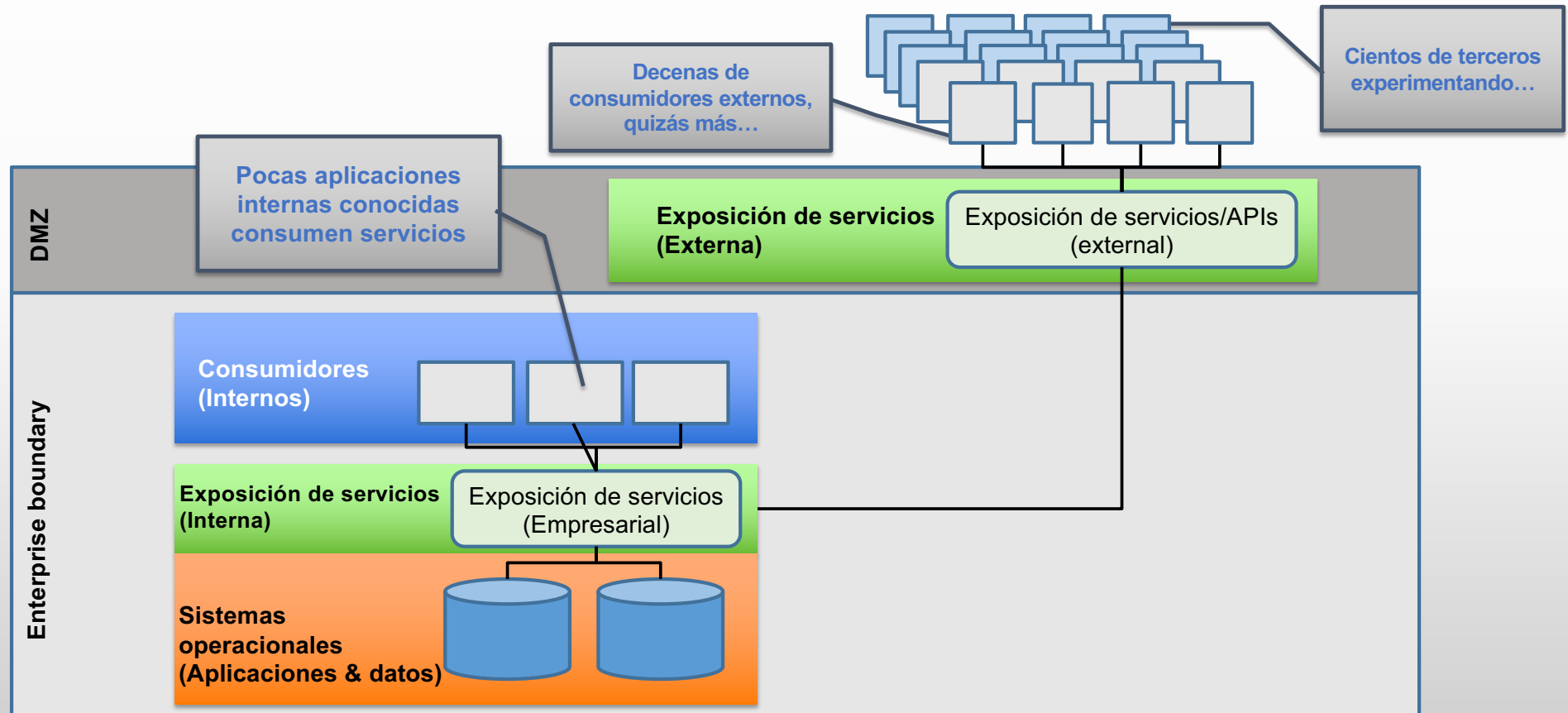




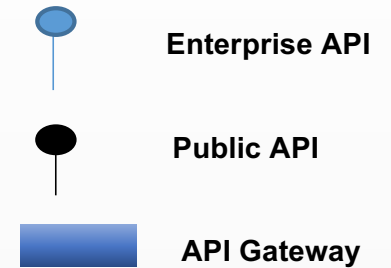
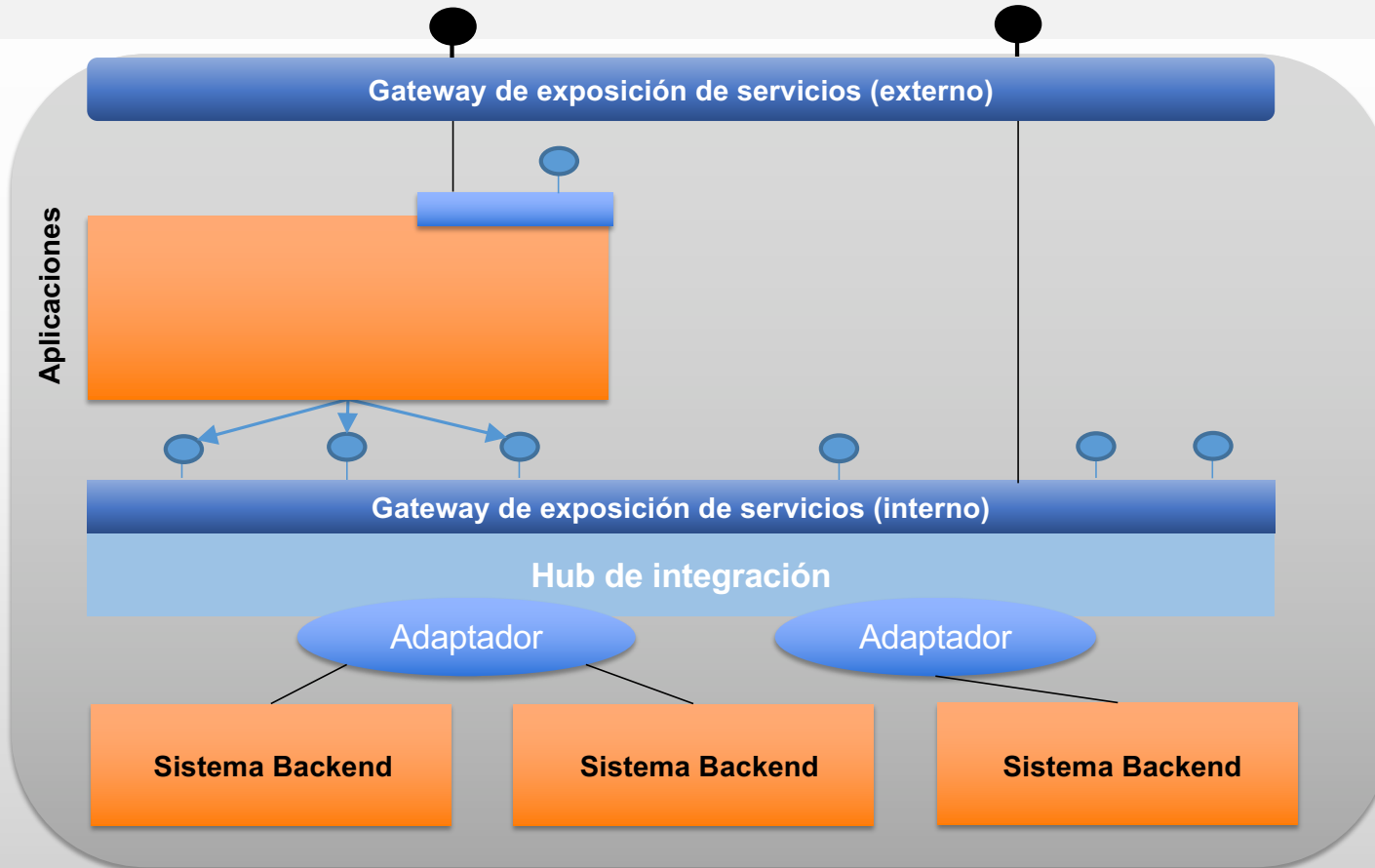
## *Beneficios de Microservicios*

- *Agilidad y productividad, reducción de tiempos de desarrollo, mejora en oportunidad de entrega de soluciones al mercado.*
- *Escalabilidad, facilidad para escalamiento horizontal*
- *Resiliencia: Gracias al “runtime” separado, la tolerancia a fallas es independiente de errores en otros componentes.*

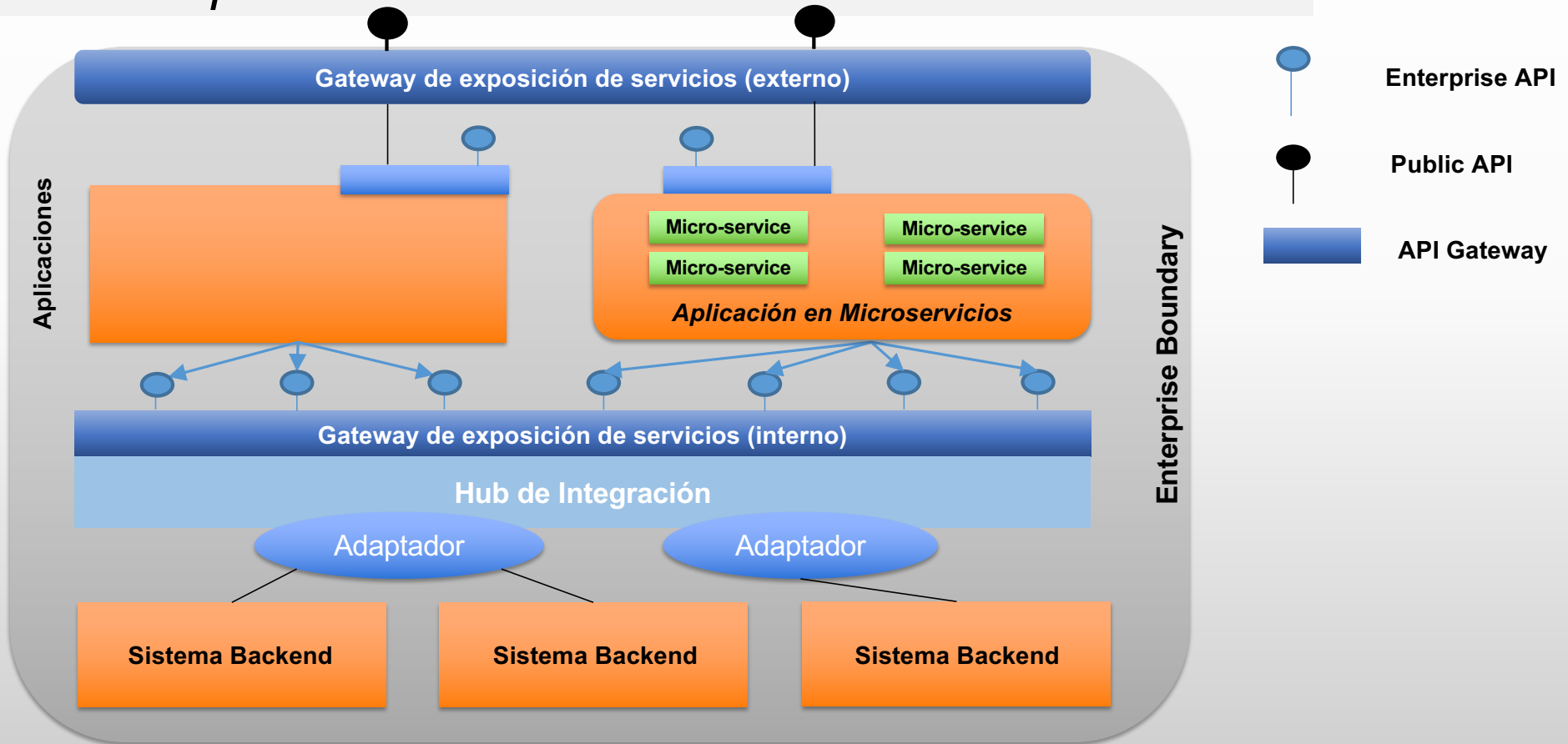
# Nuevos factores que afectan la exposición de servicios fuera de la empresa



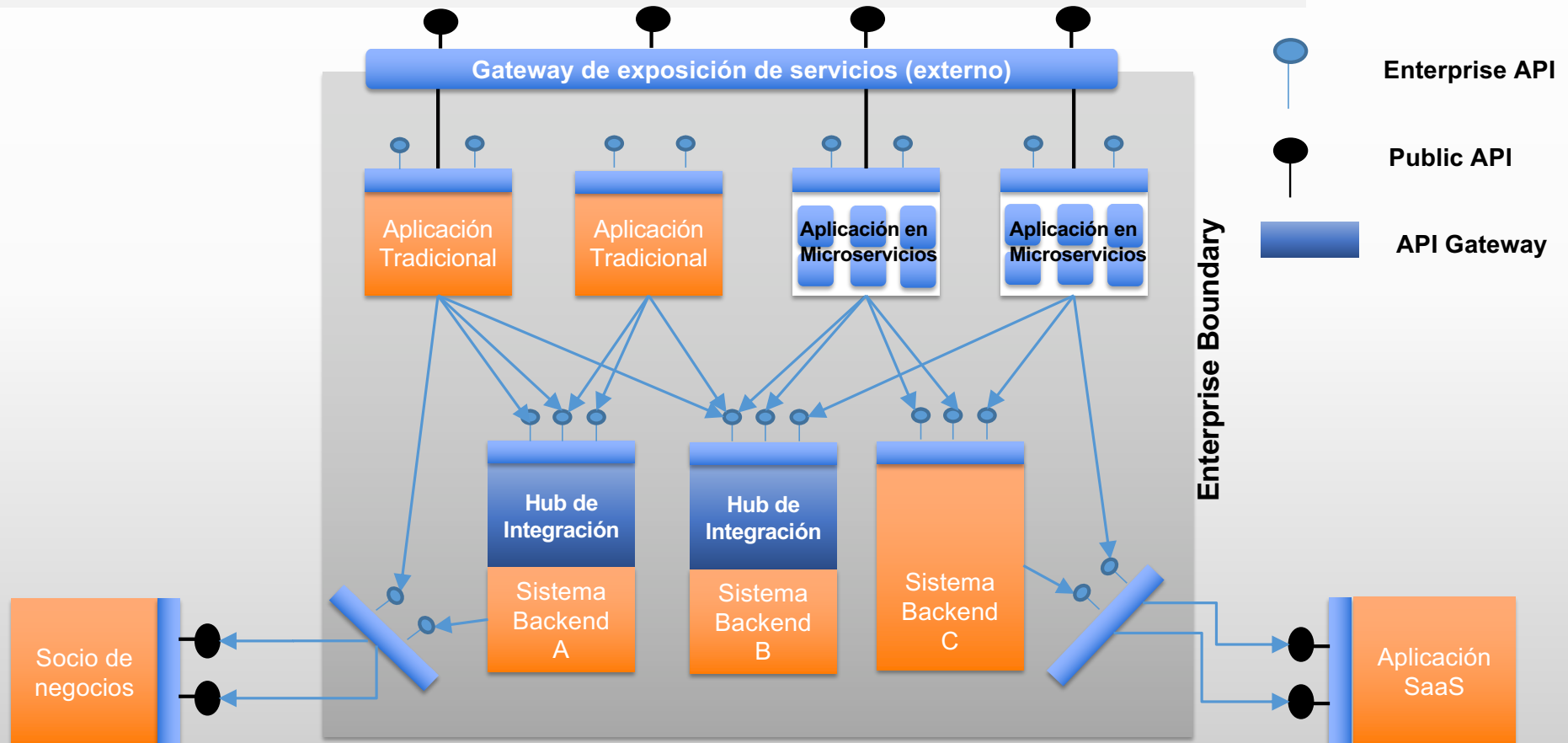
# Exposición de APIs interna y externamente



# Gran empresa con microservicios como parte de su arquitectura



# Microservices, SOA y APIs combinados







# *Arquitectura de referencia para nube híbrida*



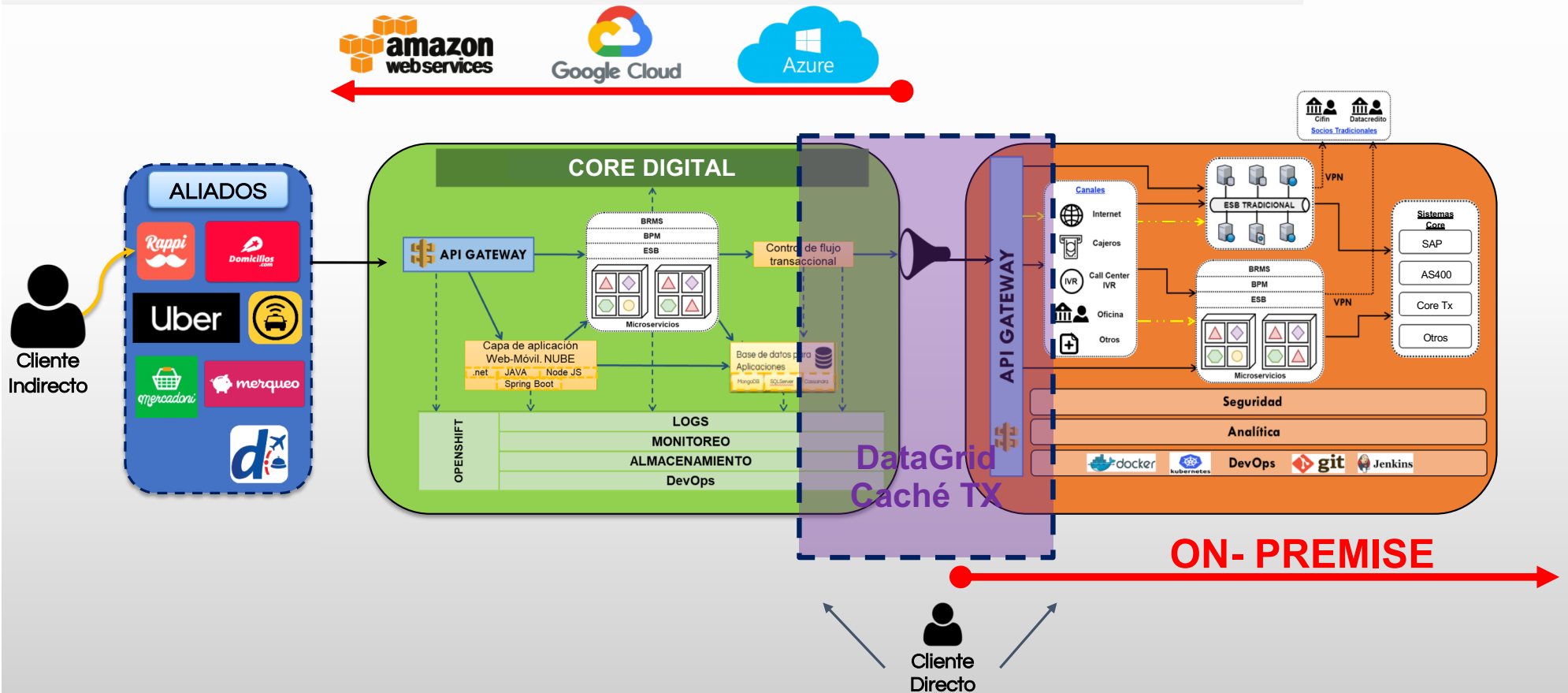
## *Atributos, cualidades o características Arquitectónicas requeridas*

- La arquitectura debe estar desplegada en alguna nube pública/privada*
- Las funcionalidades deben desplegarse en Contenedores*
- El Manejo de Logs debe ser centralizado*
- El Monitoreo debe ser centralizado*
- La arquitectura debe aplicar el modelo DevOps*
- La arquitectura debe adherirse a las regulaciones y políticas de seguridad del Cliente*

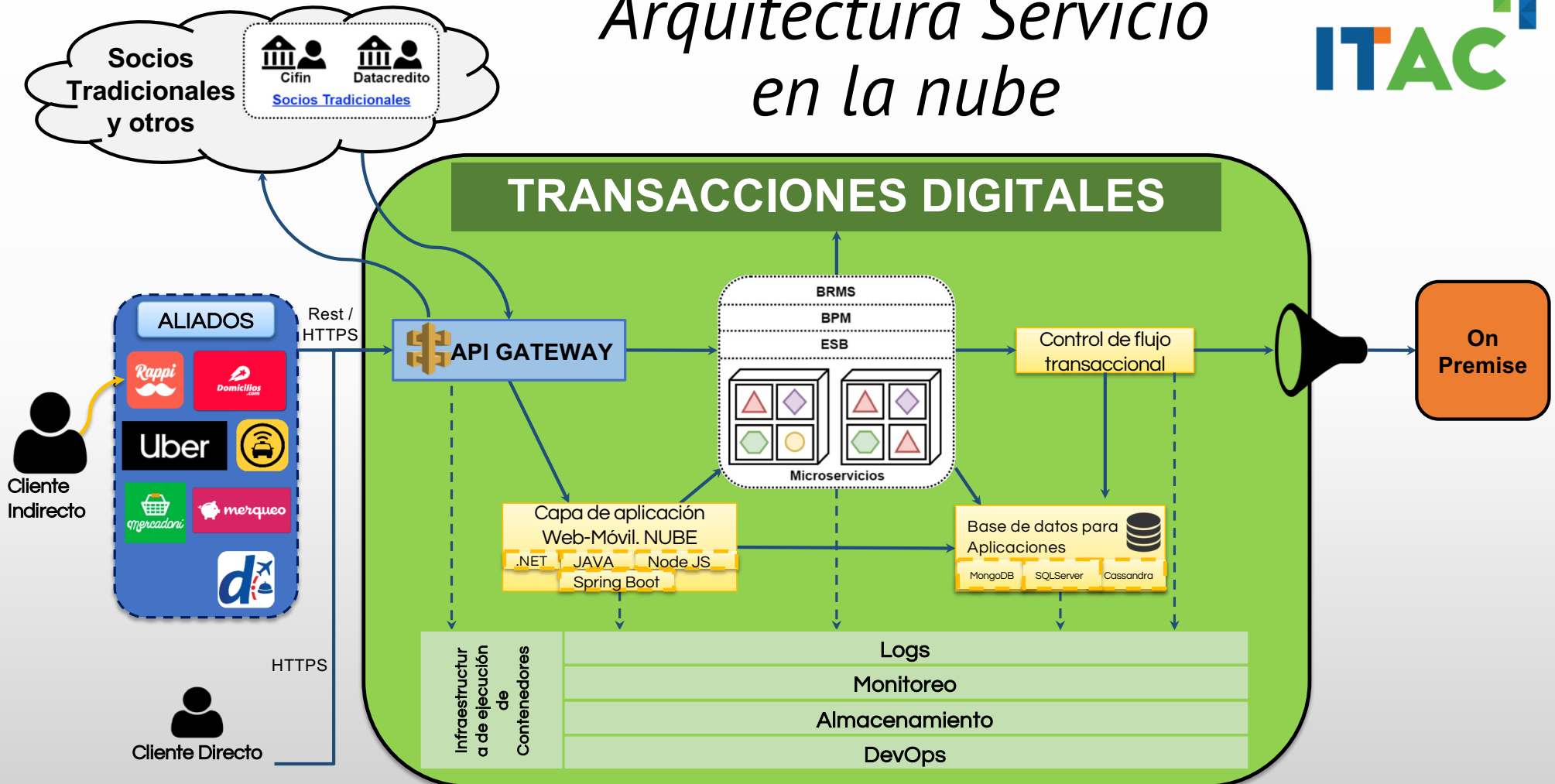
## *Atributos, cualidades o características Arquitectónicas requeridas*

- *La arquitectura debe proveer crecimiento elástico*
- *La arquitectura debe estar en capacidad de regular el flujo transaccional que se envíe hacia los sistemas OnPremise (Backends)*
- *Las funcionalidades deben exponerse como API o Contenido Web*
- *La arquitectura debe permitir portabilidad entre nubes (públicas y/o privadas)*
- *La arquitectura debe ser efectiva en costos (Costo de salida de datos desde las nubes).*
- *Idealmente que se puedan gestionar las diferentes nubes de forma centralizada*

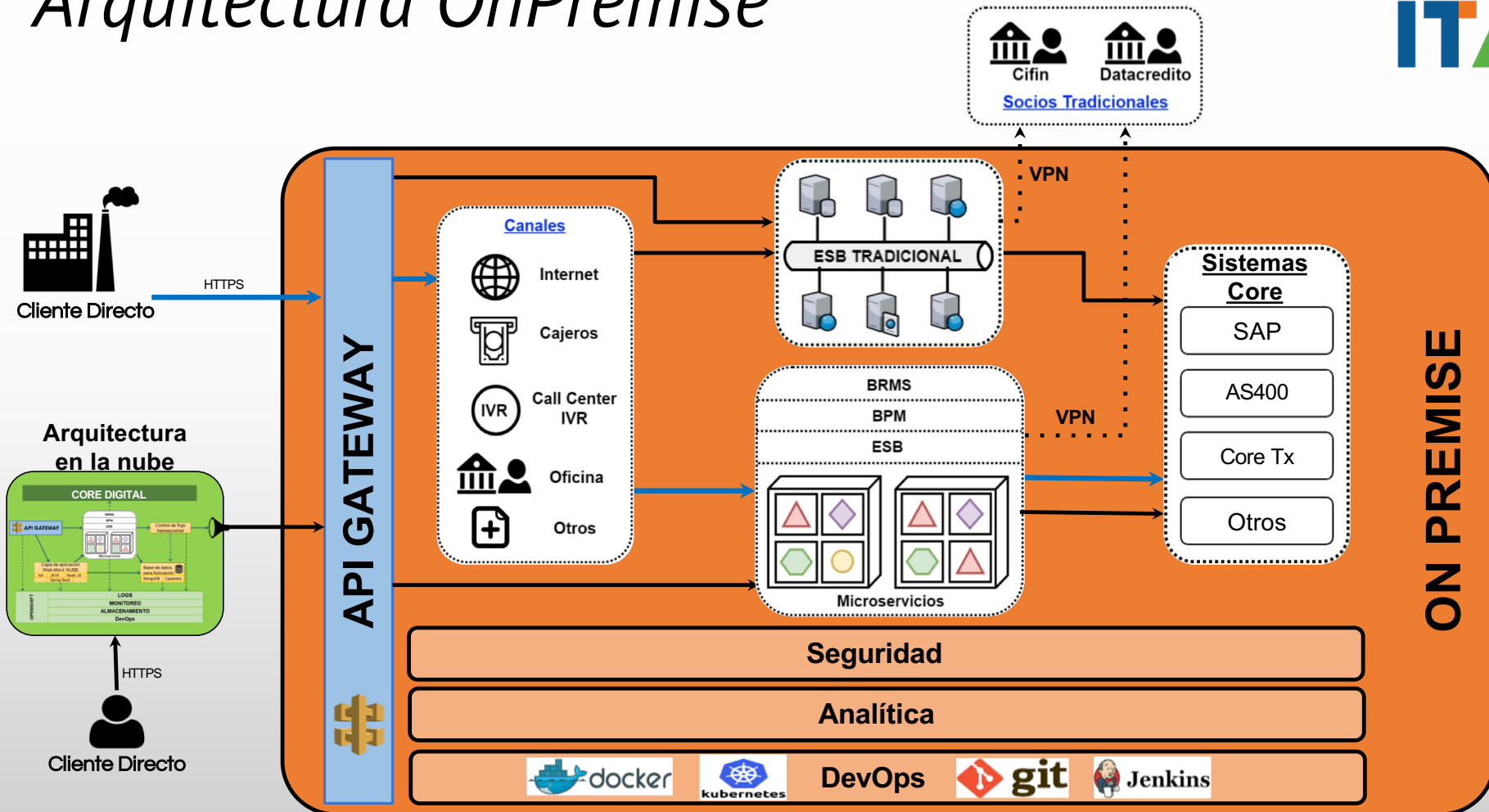
# Visión arquitectónica completa



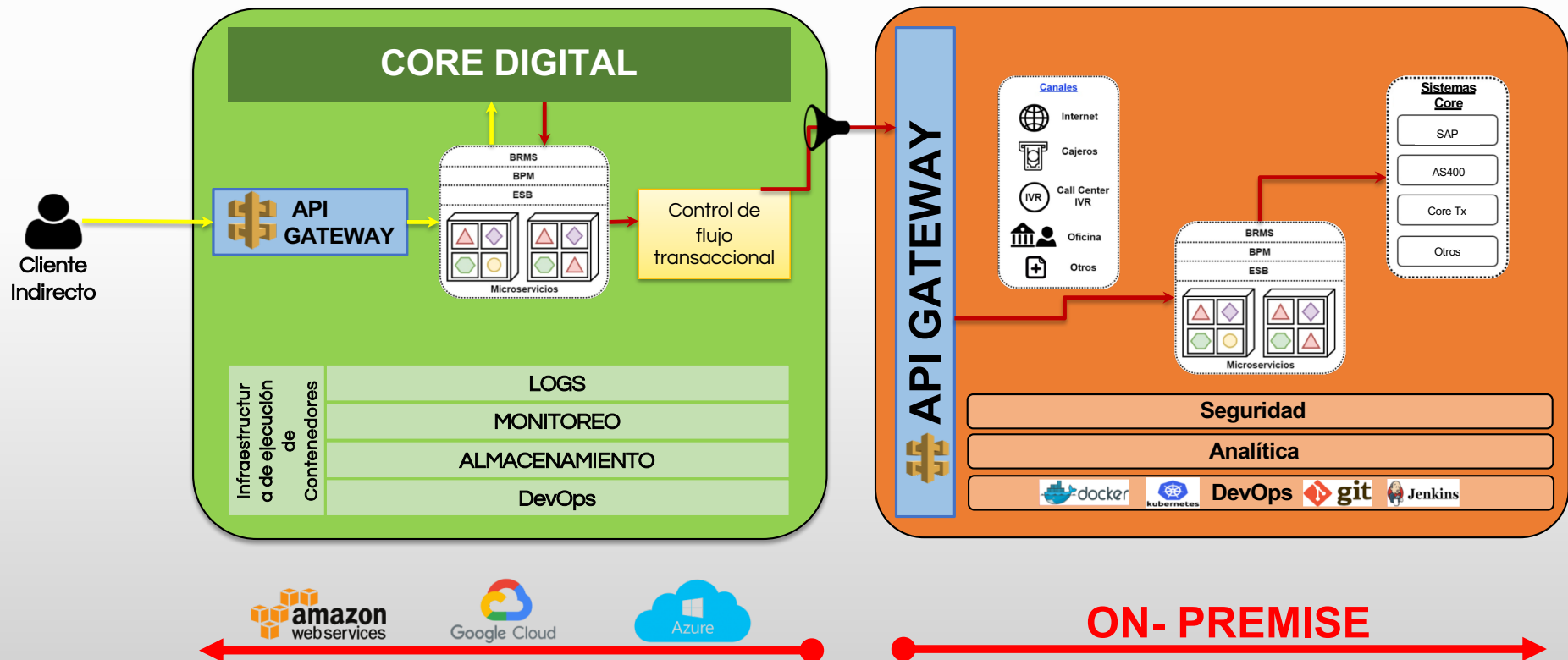
# Arquitectura Servicio en la nube



# Arquitectura OnPremise

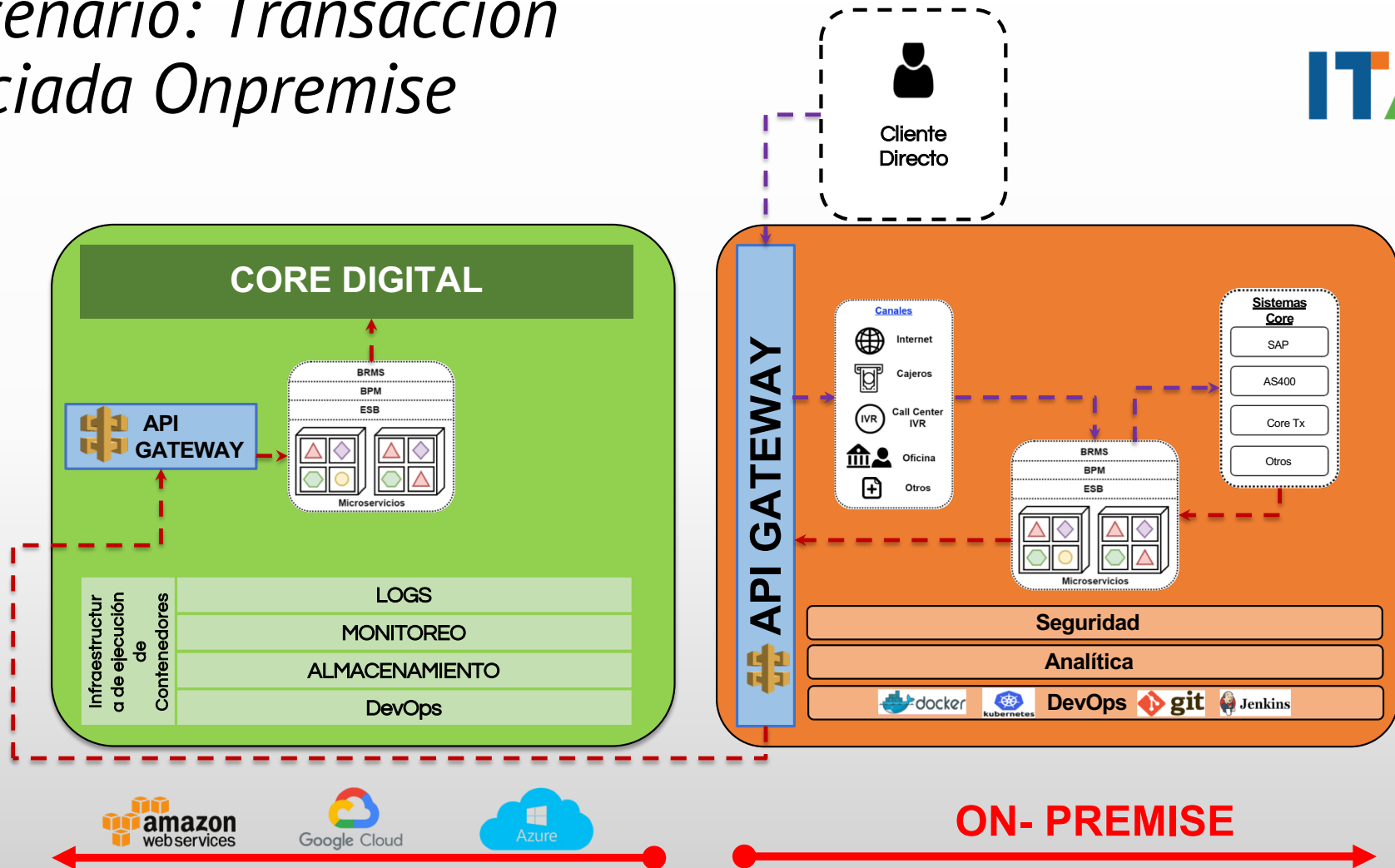


# Escenario: Transacción iniciada en nube



**ON- PREMISE**

# Escenario: Transacción iniciada Onpremise



**ON-PREMISE**



## *Pensamientos finales*

- SOA se refiere a ahorro de costos, NO a generar nuevos ingresos.
- El enfoque de APIs está en la consumibilidad y la competencia en la economía de APIs. Las APIs son un producto vendible... Generación de ingresos
- La arquitectura de microservicios es un enfoque alternativo para estructurar aplicaciones

## Invitación...

*“¿Cuánto tiempo crees que va a durar tu empleo en su forma normal?”*

*“Te sugiero que te reinventes diariamente, para continuar en este “juego” llamado vida.”*

*“Siempre yendo para adelante. No porque detrás venga gente.... Sino porque ya hay mucha gente adelante de nosotros, que nos lleva ventaja.”*





## *Referencias*

# Referencias

- Microservices, SOA, and APIs: Friends or enemies?
  - [https://www.ibm.com/developerworks/websphere/library/techarticles/1601\\_clark-trs/1601\\_clark.html](https://www.ibm.com/developerworks/websphere/library/techarticles/1601_clark-trs/1601_clark.html)
- Integration architecture: Comparing web APIs with service-oriented architecture and enterprise application integration
  - [https://www.ibm.com/developerworks/websphere/library/techarticles/1503\\_clark/1305\\_clark.html](https://www.ibm.com/developerworks/websphere/library/techarticles/1503_clark/1305_clark.html)
- Microservices vs SOA: What's the Difference?
  - <https://www.bmc.com/blogs/microservices-vs-soa-whats-difference/>
- An Enterprise Architect's Guide to API Integration for ESB and SOA
  - <https://www.ca.com/content/dam/ca/us/files/ebook/ca-enterprise-architects-guide-to-api-integration-for-esb.pdf>
- API Strategy and Architecture: A Coordinated Approach
  - <https://www.ca.com/content/dam/ca/us/files/ebook/api-strategy-and-architecture-a-coordinated-approach.pdf>
- Arquitectura de microservicios - Parte 1: Introducción
  - <http://enmilocalfunciona.io/arquitectura-microservicios-1/>
- Richards, M. Software Architecture Patterns. O'Reilly Media, Inc. (2015).



*¡Gracias!*



*Innovación para la  
transformación digital.*