

# Arquitecturas Empresariales y de Integración

***Sesión #13\_14: Estilo de Arquitectura SOA & identificación de servicios***

Departamento de Ingeniería  
de Sistemas y Computación

Jorge Arias



***“Para el año 2008, SOA será la práctica prevaleciente de ingeniería de software, finalizando así 40 años de dominio de las prácticas monolíticas de hacer software” Gartner 2005***

# Reflexionemos (1)

- Gary Hamel en su best seller “ The Future of Management” nos habla que **el futuro de la administración** se define en torno a dos palabras claves: **Innovación y Agilidad** ¿Pero cómo lograrlo? ¿ Qué tiene ver todo esto con IT?
- **Gartner concluye:** “El negocio es un **objetivo en movimiento**” ¿Cómo podemos ofrecer **plataformas, enfoques y estrategias** que soporten un **negocio cambiante**?
- La **tendencia** es bien clara hoy en día:
  - **Presupuestos de tecnología cada día mas reducidos** (Post dot Com, y efectos de la caída de la economía mundial)
  - **Más requerimientos menos presupuesto** (más con menos)

## Reflexionemos (2)

- Sí SOA es parte de la respuesta ¿Cómo lograrlo? ¿ Acaso SOA no es un “hyper” que promete mucho y está entregando poco?

# Agenda



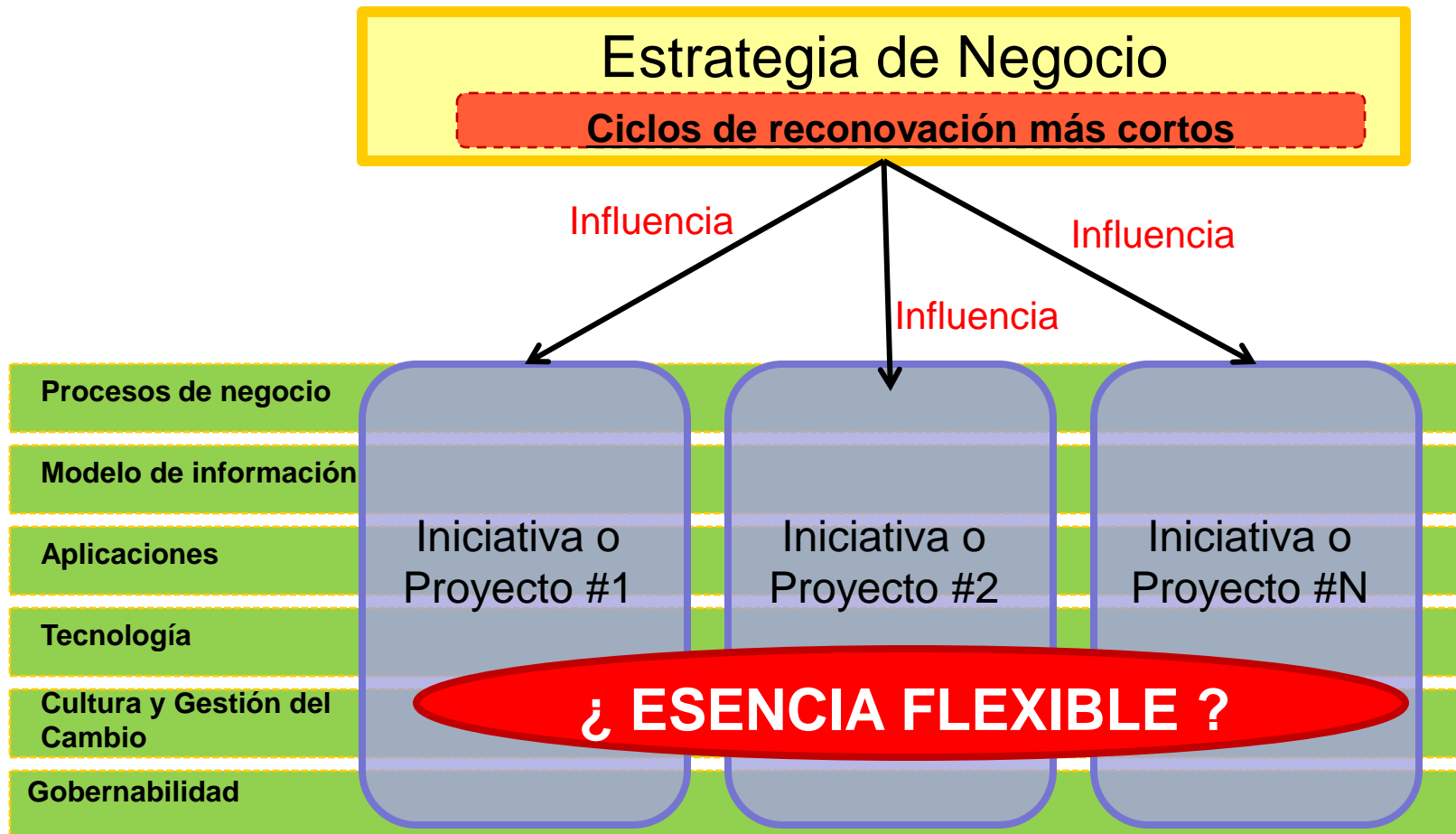
1. Motivación & Contexto

2. SOA: Estructura Conceptual

3. EA: Descubriendo servicios

4. Conclusiones

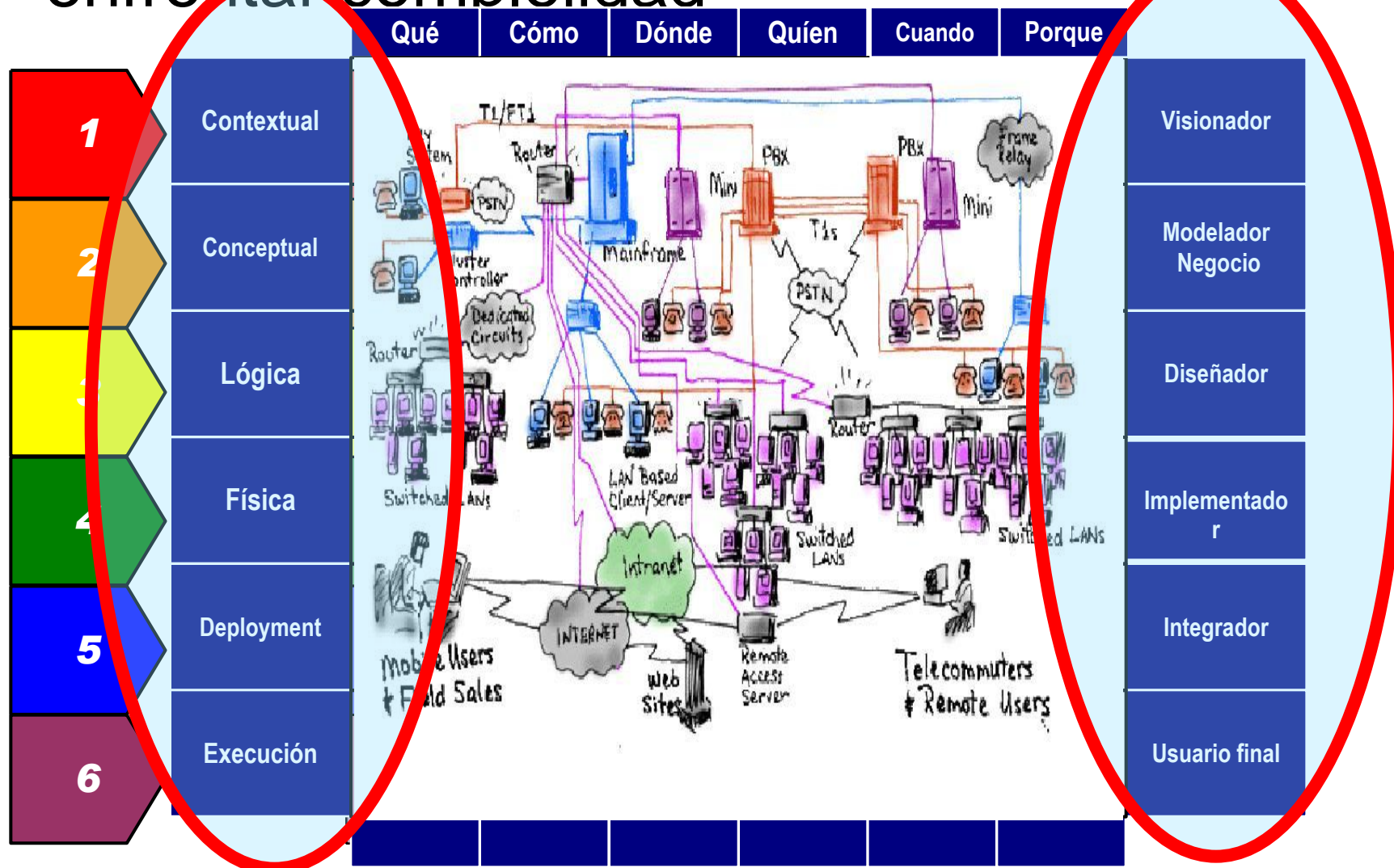
**Contexto:** La estrategia de negocio define la esencia y estructura de todo proyecto



# Contexto: Estrategía de negocio y esencia de cada proyecto

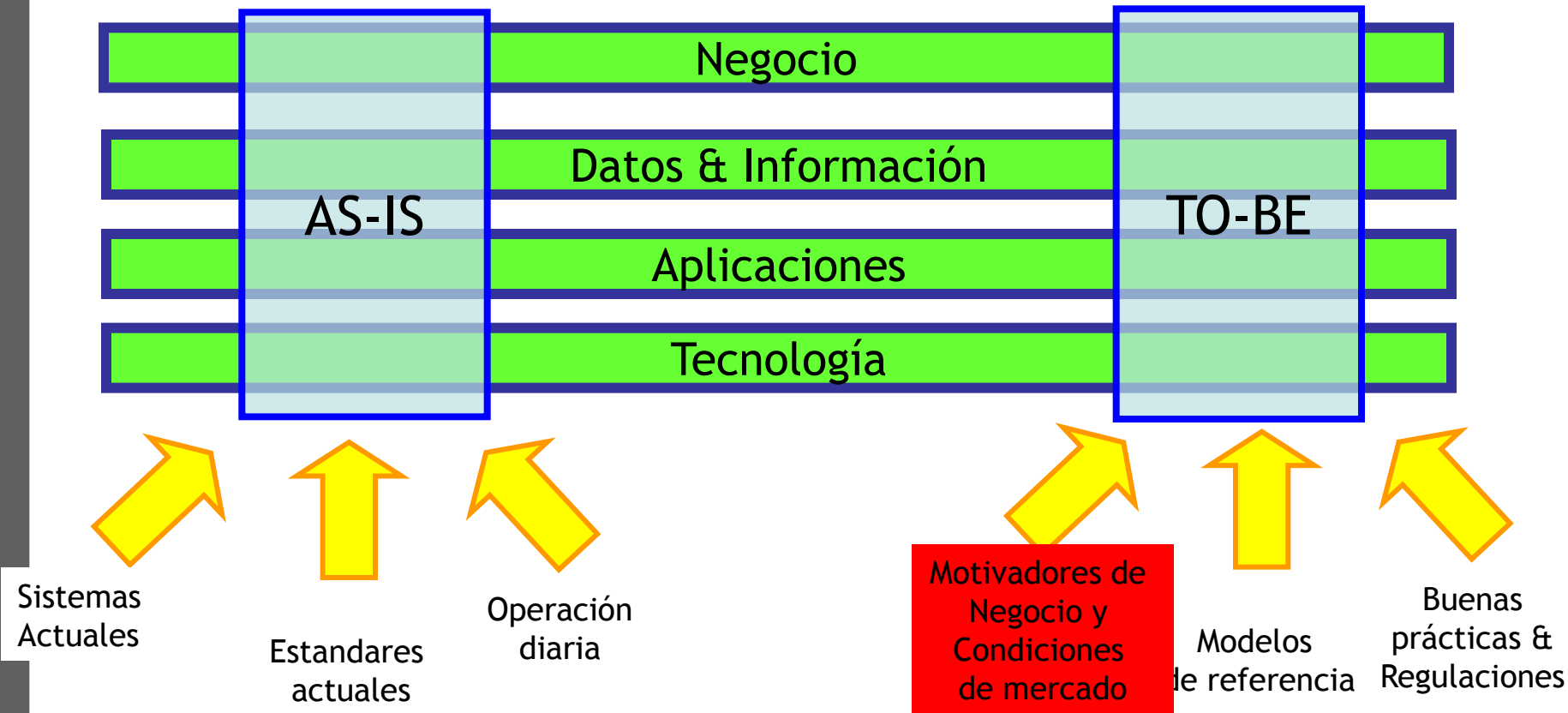
- Cualquier iniciativa o proyecto de negocio/IT obedece a una respuesta a la estrategia de negocio
- La complejidad de un proyecto tiene que ser gestionada desde diferentes dimensiones. Considerar una visión TI simplista sólo terminará en fracasos
- Necesitamos de un estilo de arquitectura lo suficientemente flexible que permita soportar dicha estrategia de negocio cambiante.

# Contexto: EA Como herramienta para enfrentar complejidad

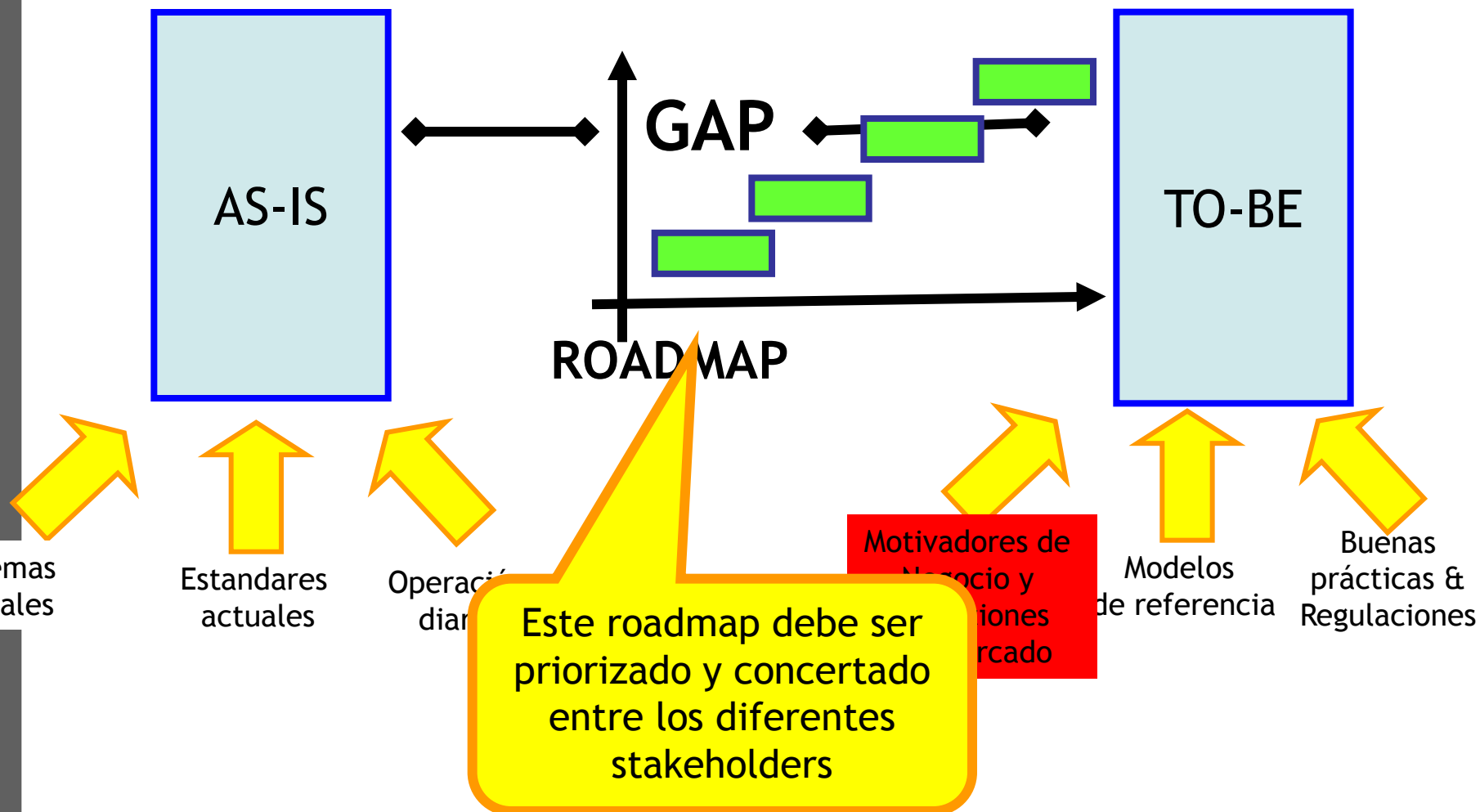




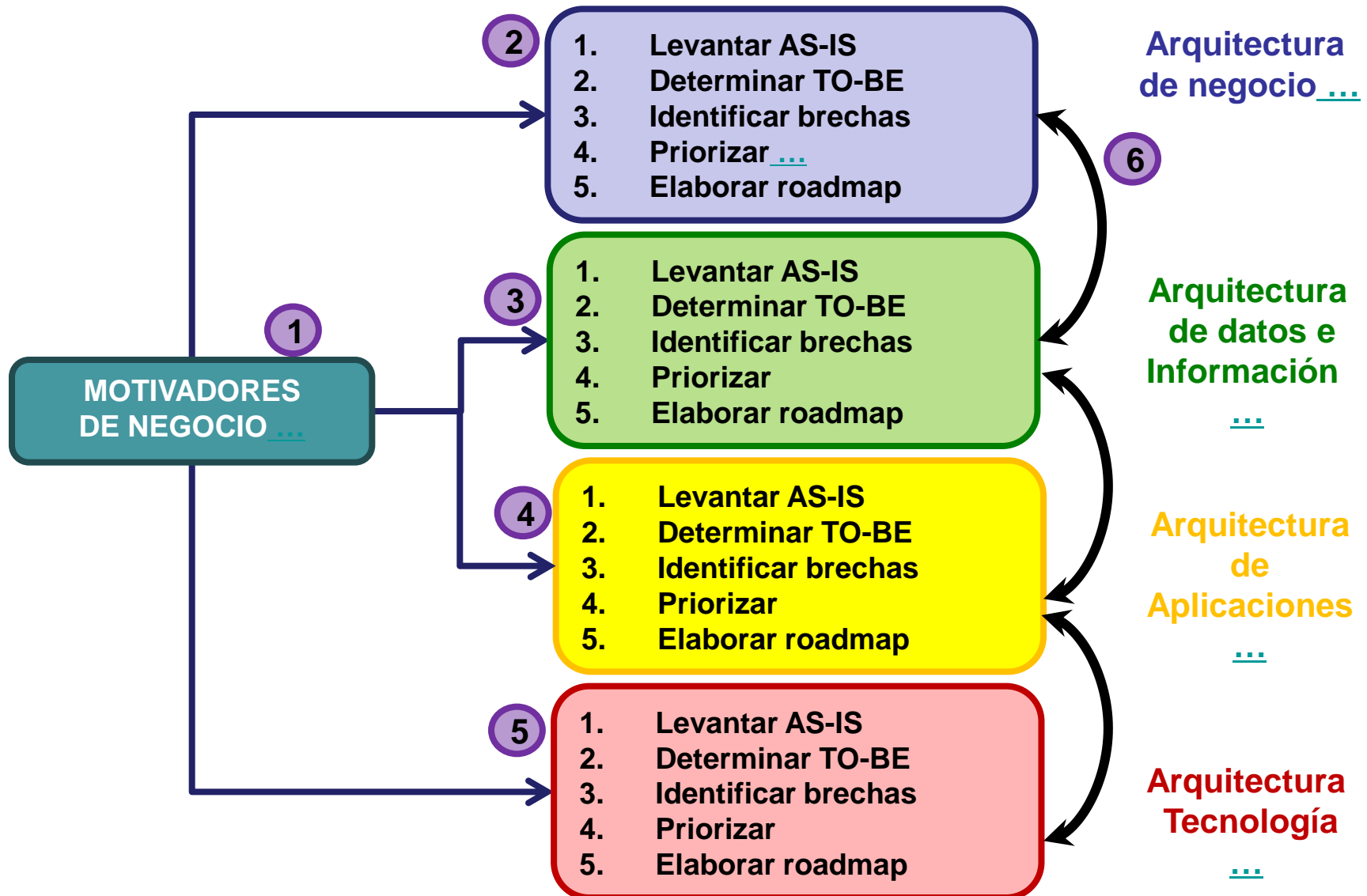
# Contexto: Modelo metodológico de Operación de EA



# Contexto: Modelo metodológico de Operación de EA (Parte 2)



# Contexto: En resumen EA opera...

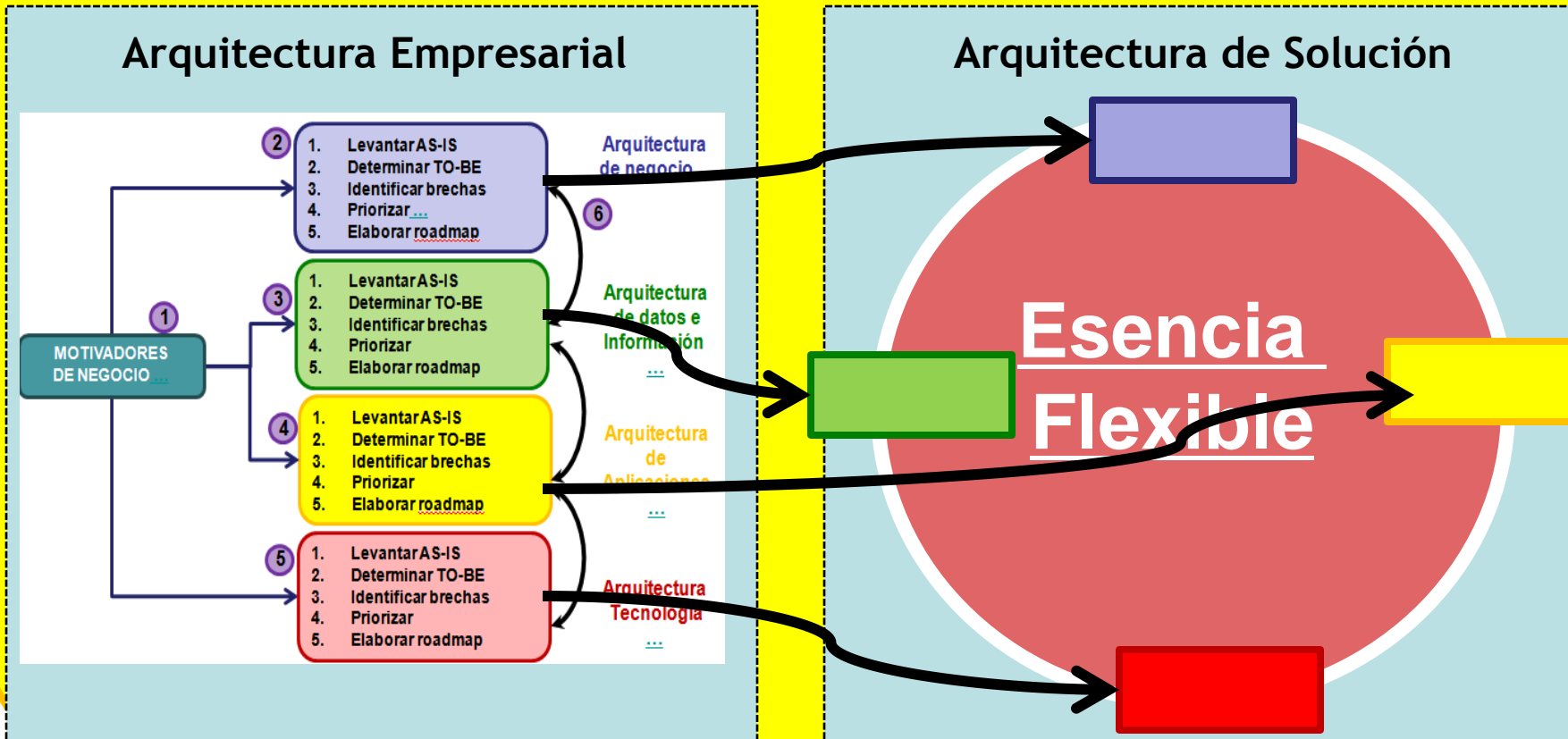


# Contexto: EA + Esencia Flexible

## Iniciativa o Proyecto Corporativo

**¿QUÉ?**

**¿CÓMO?**



# Contexto: SOA cómo esencia Flexible

- **SOA es:**

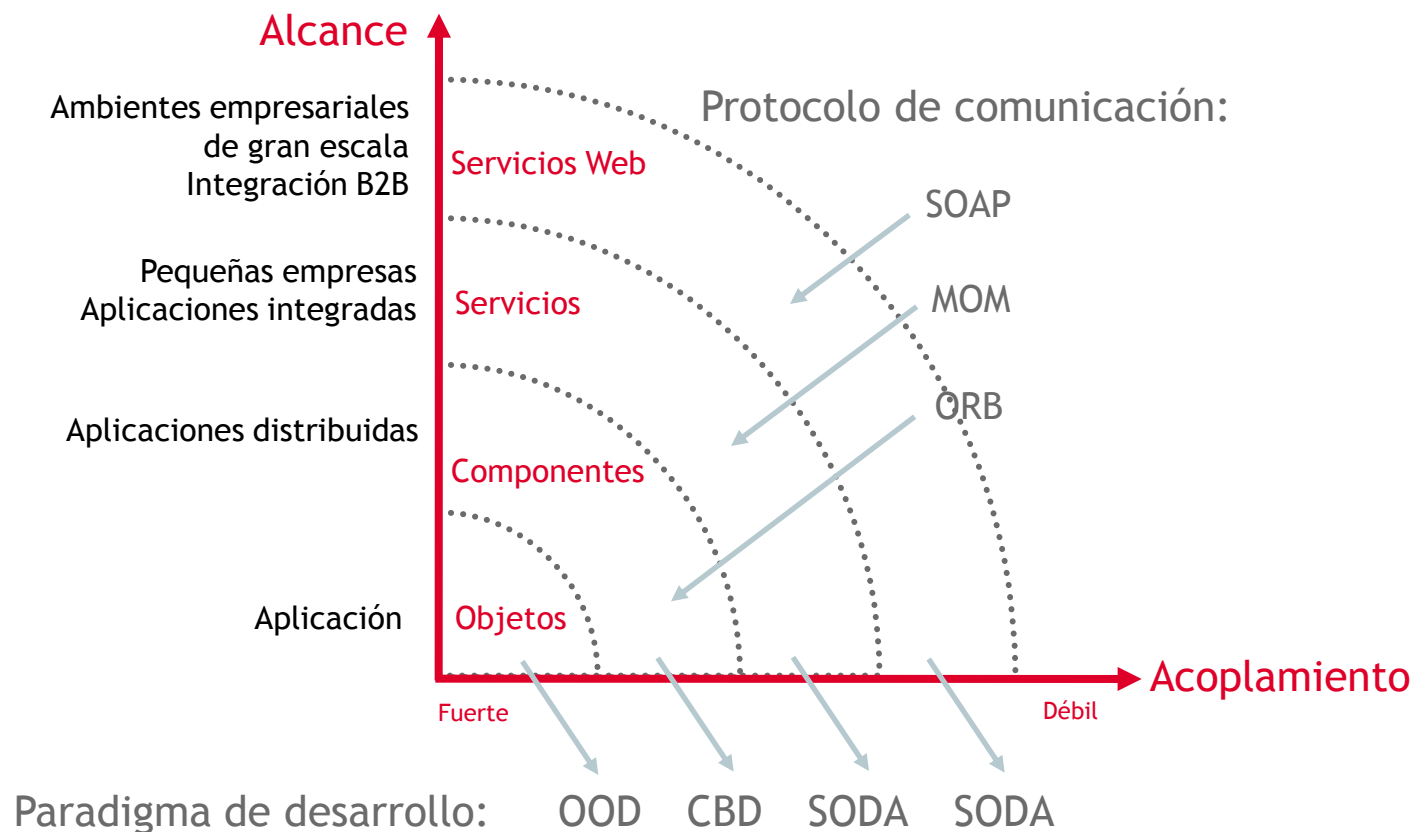
- Un **estilo de arquitectura** definido en términos de **varios principios de diseño**, los cuales buscan **implementar unidades de negocio, información e infraestructura flexibles**, reusables e interoperables.
- **Enfoque arquitectónico** que busca **alinear Negocio y Tecnología** a través **de piezas de negocio** bajamente acopladas y reutilizables que se **componen en procesos de negocio flexibles y medibles** contra una estrategia de negocio.

- **SOA no es:**

- **Tecnología**
- ESB: Enterprise Services Bus
- **Webservices**

**Sockets, RMI,  
Webservices, Remoting,  
Any RCP pueden ser  
empleadas como  
tecnologías de  
implementación SOA**

# Contexto: SOA cómo esencia Flexible - Evolución



Fuente: Gartner Group

# Contexto: SOA cómo esencia Flexible (2)

- Algunas otras definiciones comerciales de SOA:

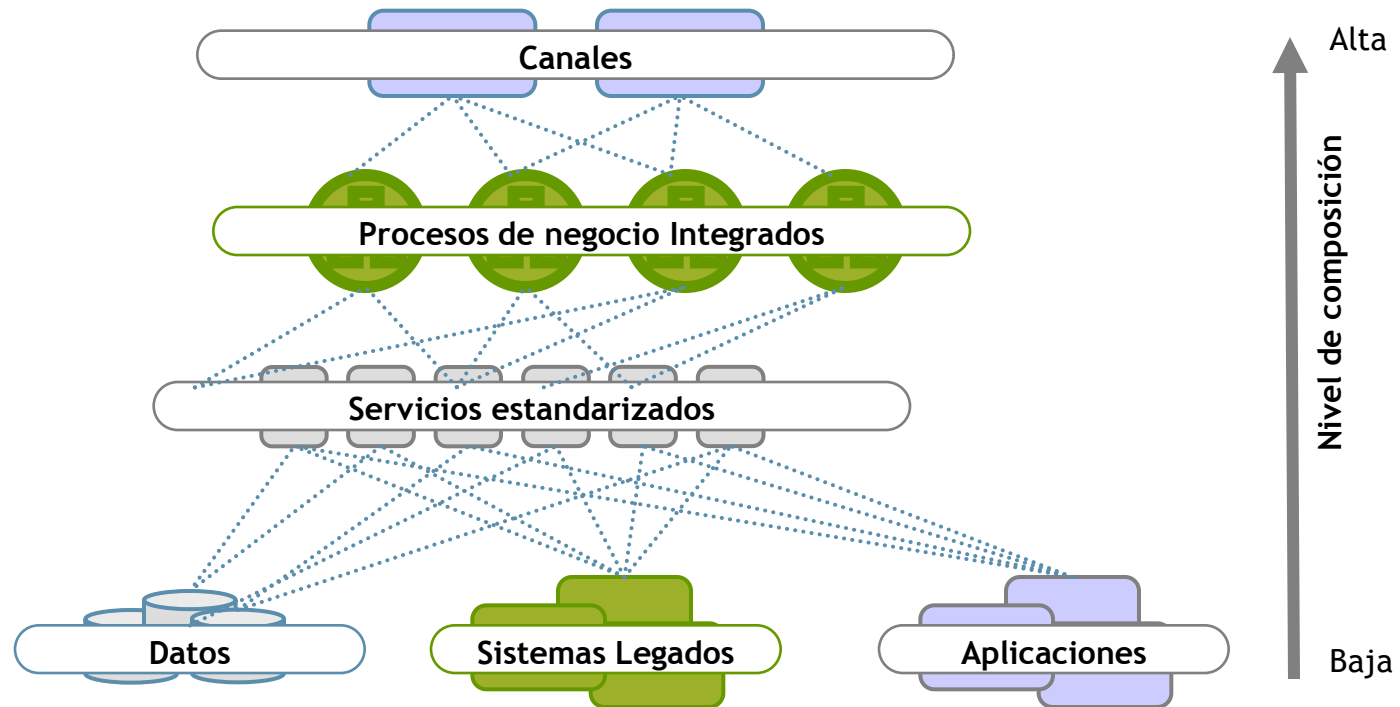
“Es un enfoque de diseñar y construir soluciones de software a partir de componentes **débilmente acoplados** que exponen **funciones de negocio** como **servicios accesibles programáticamente por otras** aplicaciones a través de interfaces de publicación y descubrimiento”

IBM

“**Políticas, prácticas y marcos de referencias** que habilitan que una **funcionalidad de una aplicación** sea **provista y consumida como un conjunto de servicios publicados con una granularidad relevante** para el consumidor, los cuales son abstraídos de su implementación utilizan una forma de interfaces única, basada en estándares”

CBDI Forum

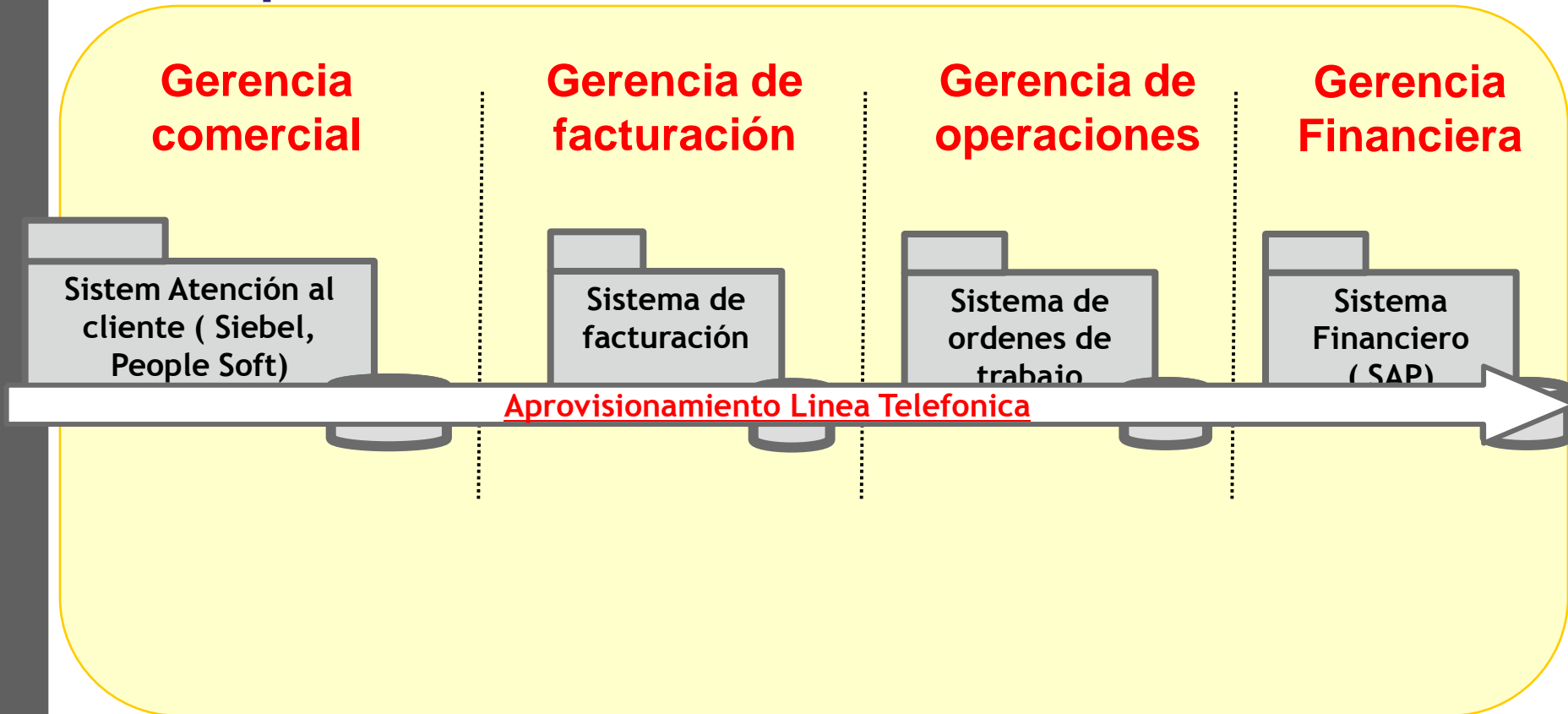
# Contexto: SOA cómo esencia Flexible y de Composición

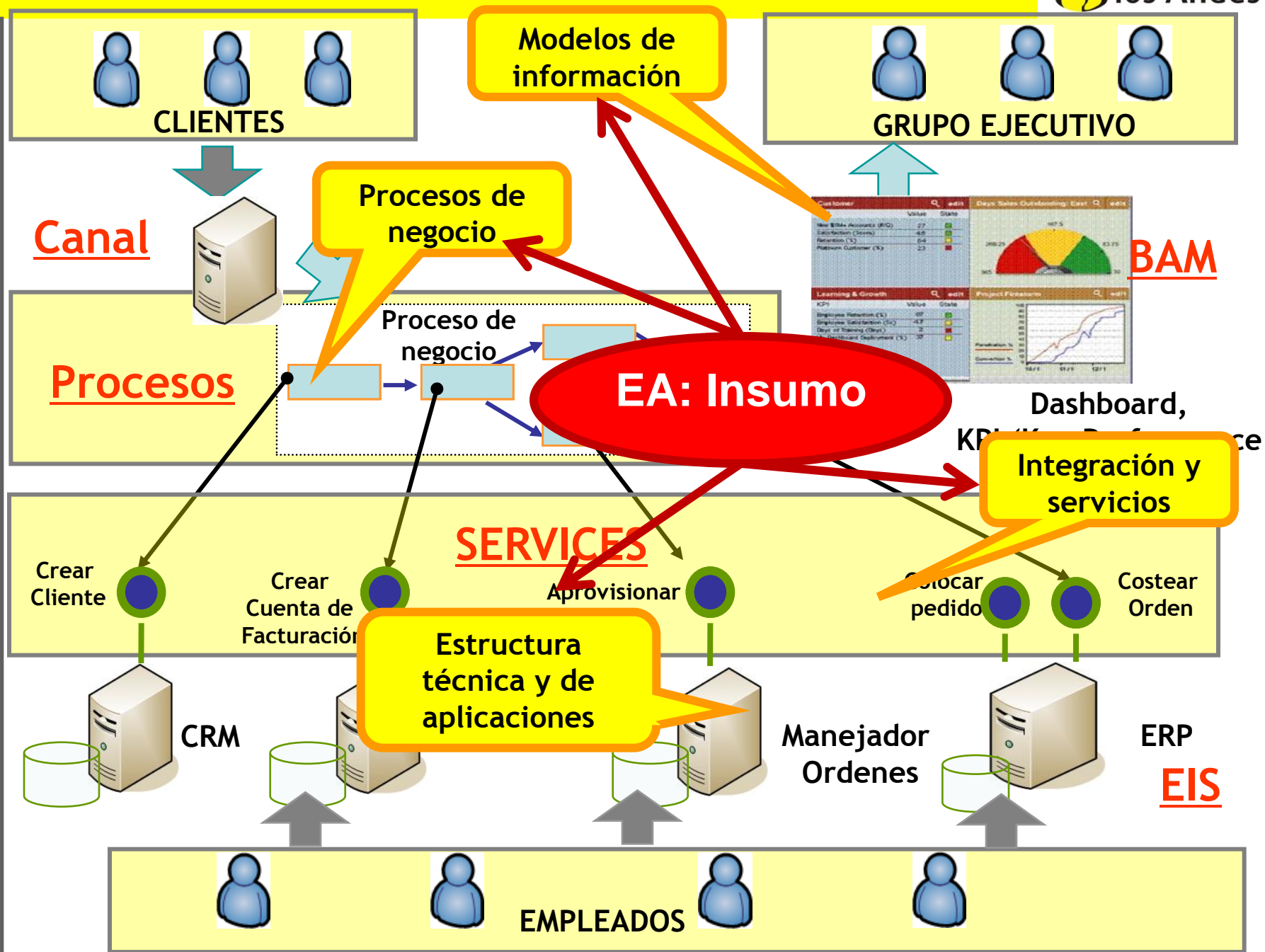




Contexto: SOA habilita la orientación a procesos de la organización vía esencia Flexible

## Compañía Telco ABC





# Contexto: En resumen

- El estilo de arquitectura que propone **SOA** permite estructurar una **arquitectura de Solución** a un proyecto en particular **apoyado en**:
  - **Unidades de negocio flexibles**, reutilizables e interoperables llamadas **servicios**.
  - **Composiciones de unidades** de negocio reutilizables (Procesos de negocio o servicios más gruesos)
  - Especificaciones y definiciones de **elementos estructurales considerados** dentro de una **análisis de EA**.
    - Procesos de negocio
    - Entidades de negocio
    - Portafolio de aplicaciones y funcionalidades
    - Mapa de integraciones
    - Frameworks de infraestructura técnica
    - Motivadores de negocio.

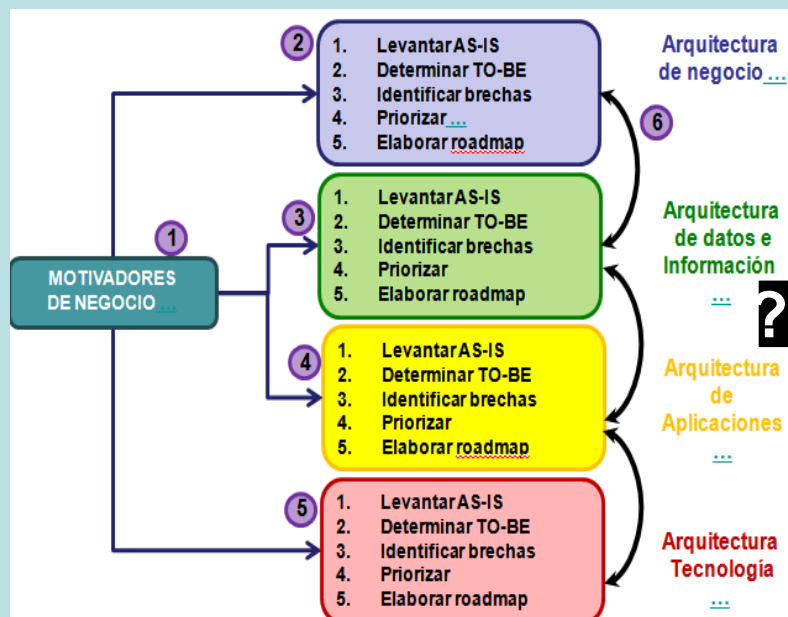
# Contexto: EA + SOA = SOA exitoso

## Iniciativa o Proyecto Corporativo

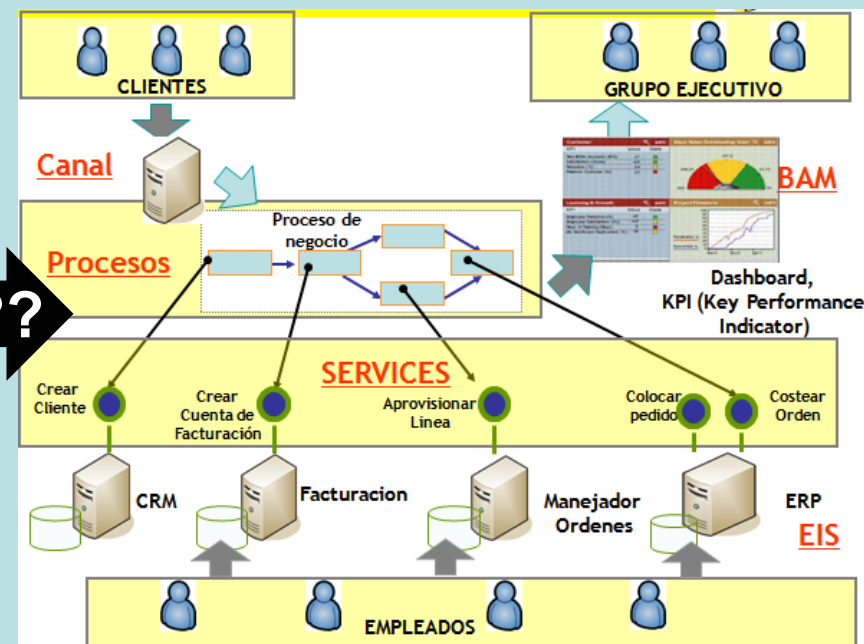
¿QUÉ?

¿CÓMO?

### Arquitectura Empresarial

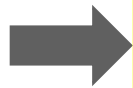


### SOA: Arquitectura Solución



# Agenda

1. Motivación & Contexto

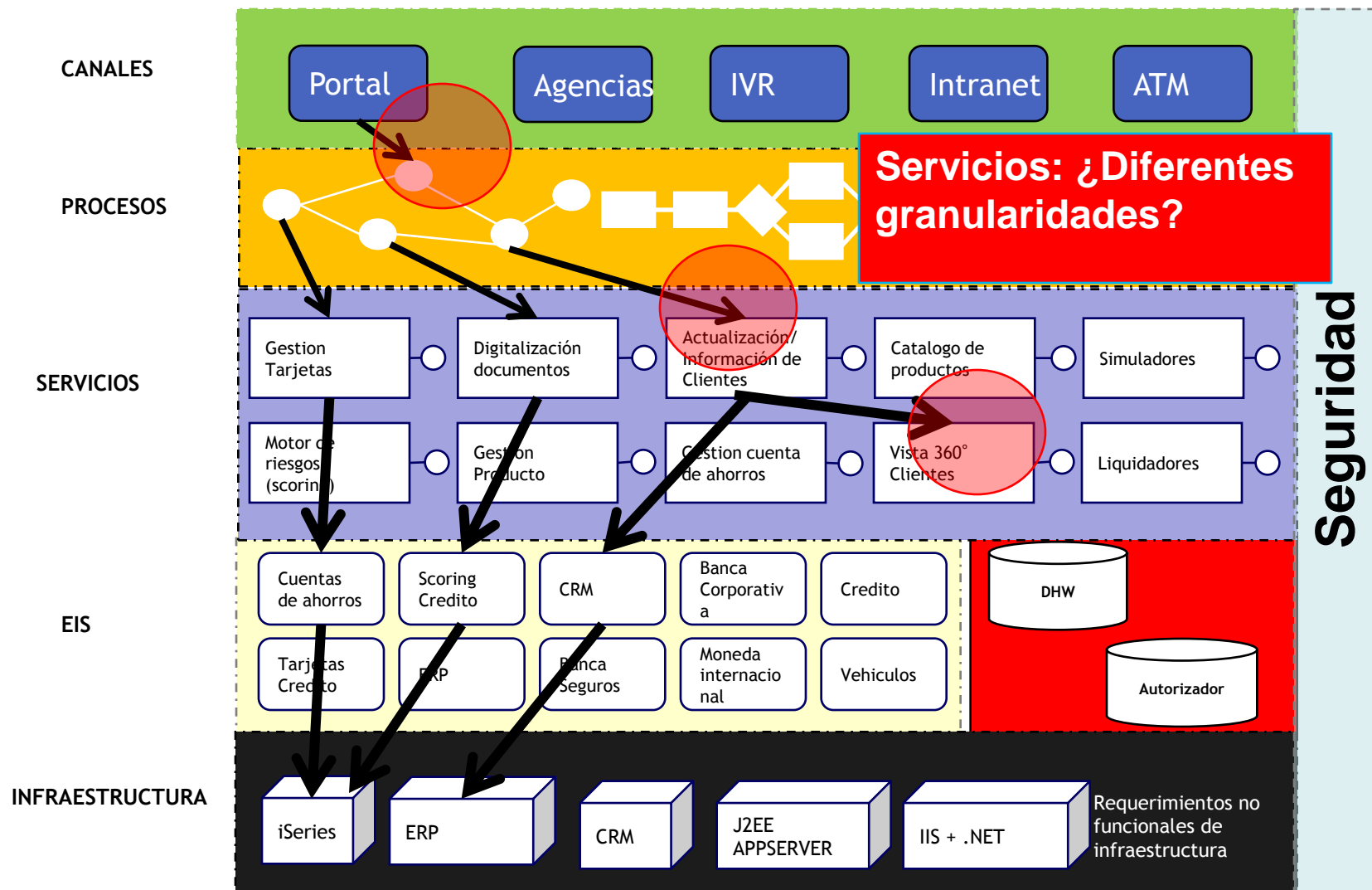


2. SOA: Estructura Conceptual

3. EA: Descubriendo servicios

4. Conclusiones

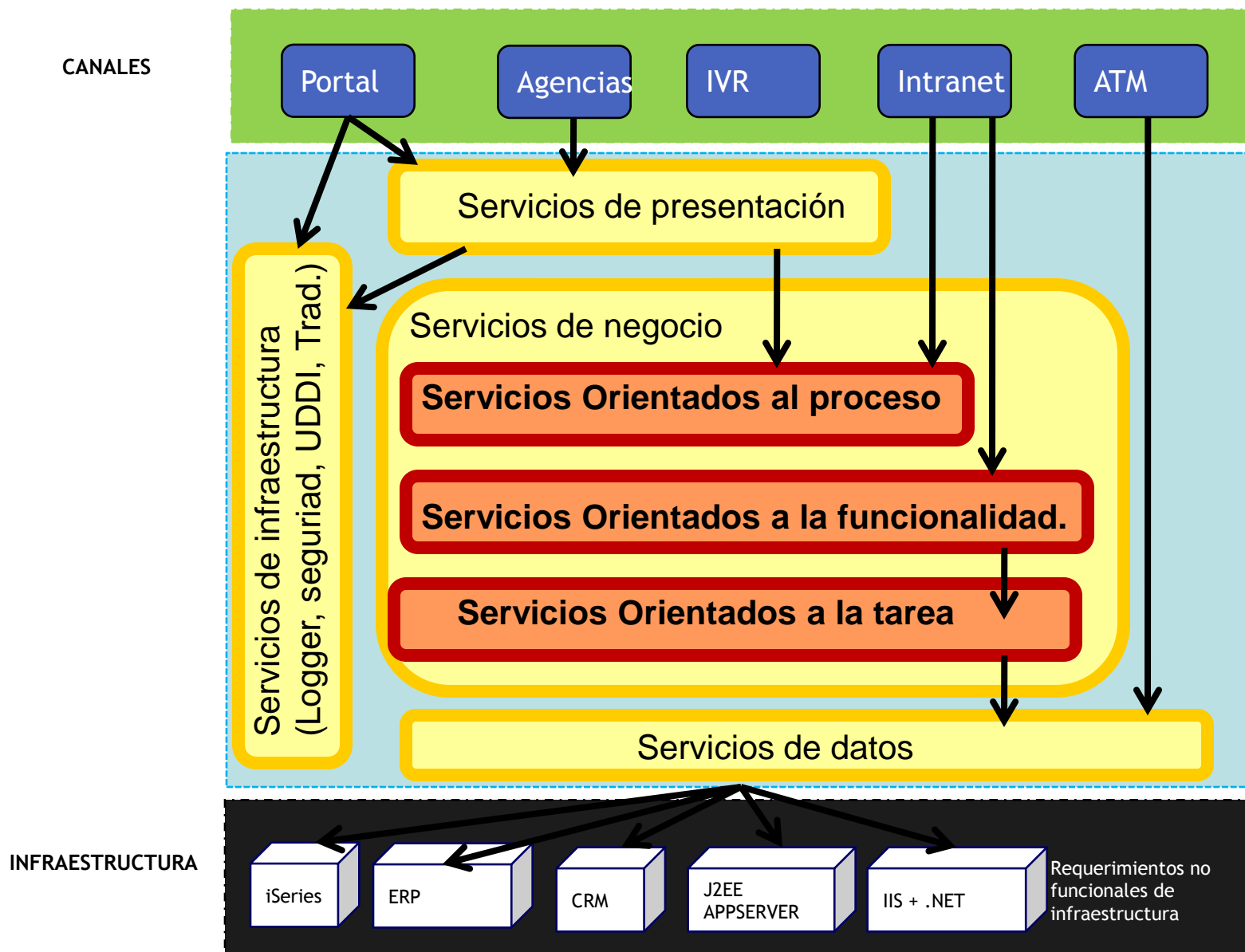
# SOA: Modelo conceptual Multicapas



# SOA: Características del modelo multicapas

- Las **aplicaciones y sistemas de información** deben construirse con una **visión multi-canal** (La misma funcionalidad o servicio para todos los canales)
  - En los Bancos es bien común que se **tenga implementada funcionalidad de negocio x canal**.
  - Los canales consumen procesos y servicios
- **Como consecuencia del estilo de arquitectura** SOA se obtienen: Procesos & Servicios
- Las funcionalidades que se publican como **servicios están al interior de las aplicaciones** que actualmente soportan la operación de la empresa.
- La gestión de **seguridad se da a todos lo niveles**.

# SOA: Servicios x Nivel de granularidad



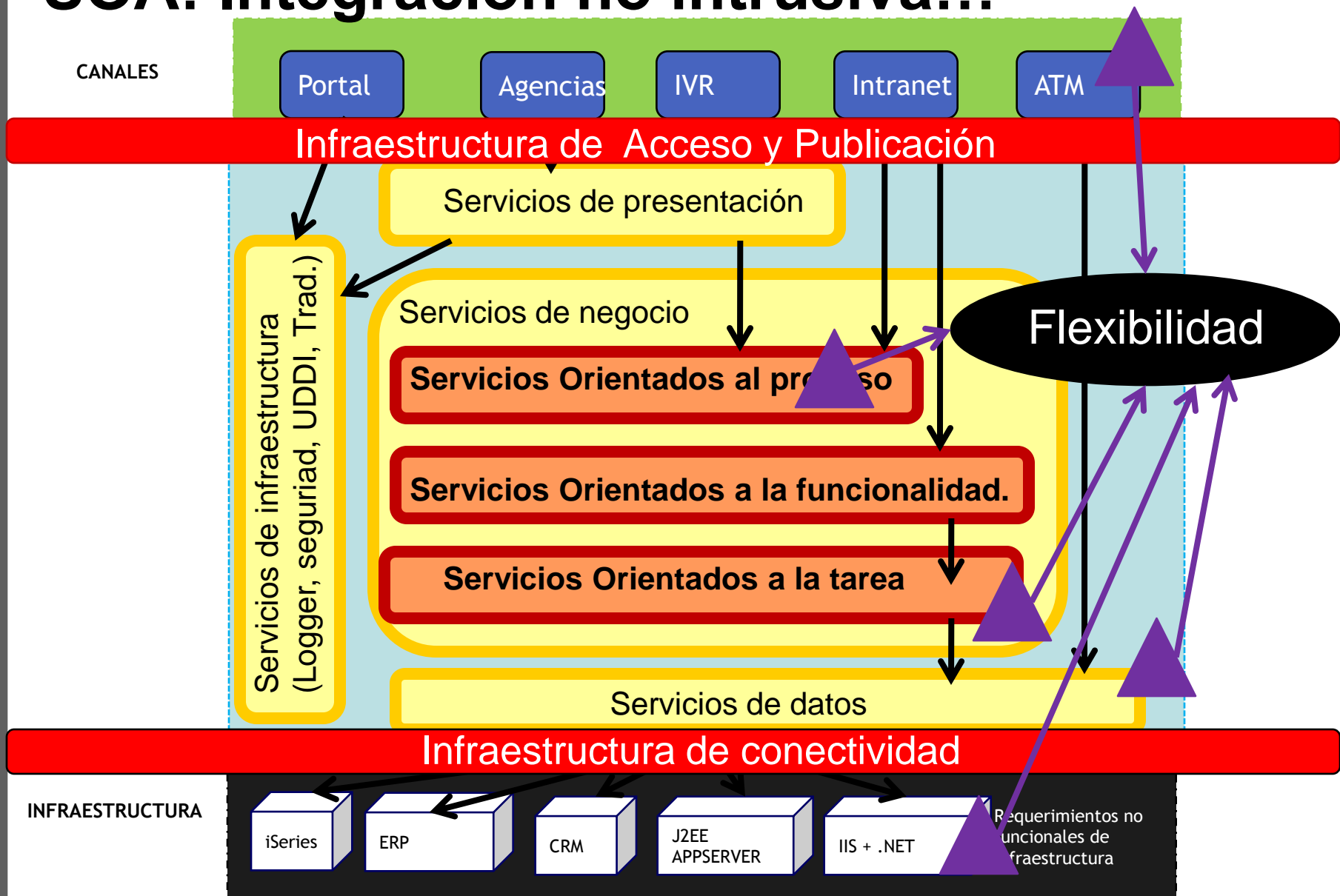


# SOA: Integración no intrusiva

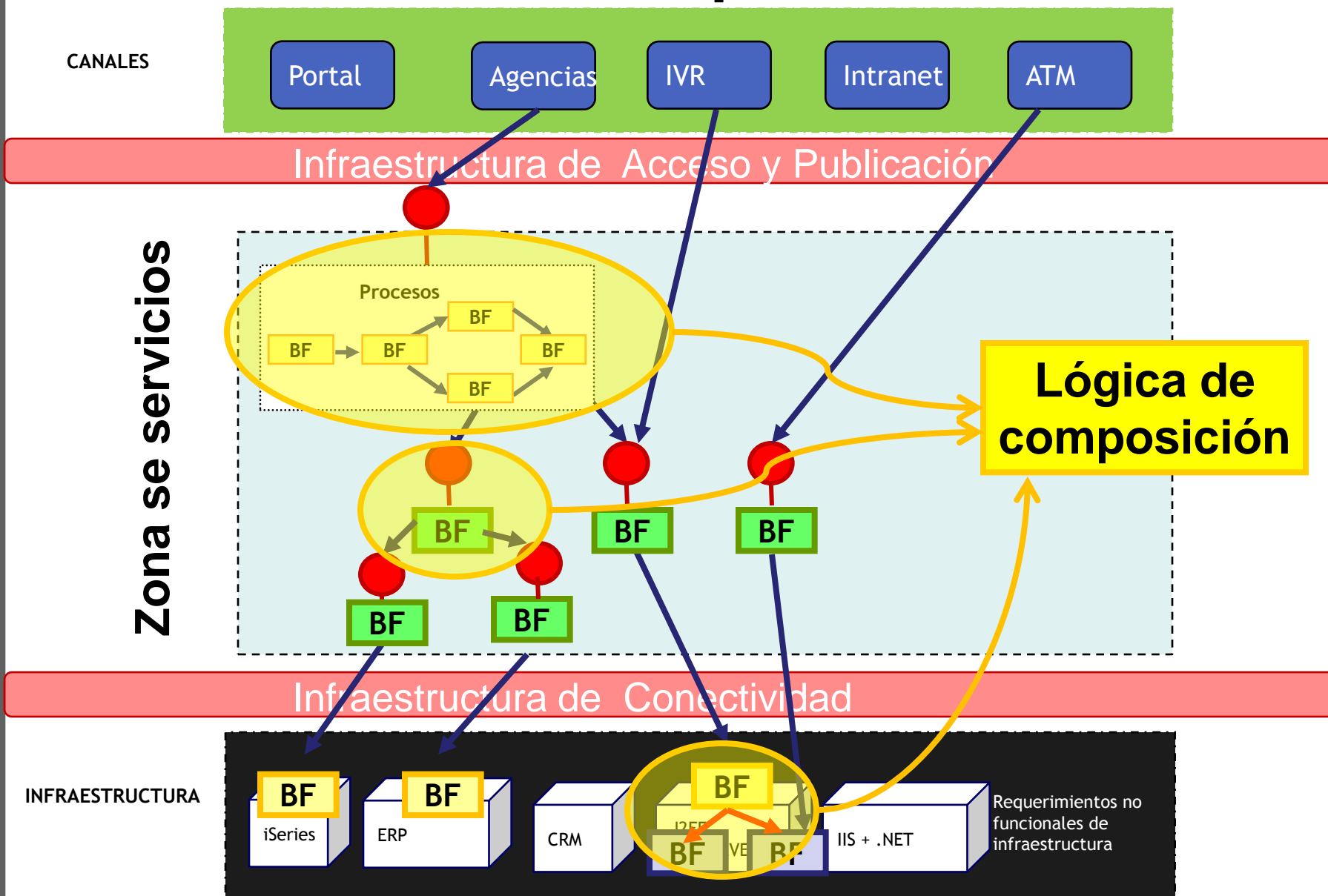
*Necesitamos de **un modelo de integración** entre **canales, servicios y aplicaciones proveedoras** **no intrusivo**. Los canales no deben ser afectados al momento de consumir servicios. Las aplicaciones proveedoras no deben ser afectadas al momento de publicar sus funcionalidades.*

***¿COMO LOGRARLO?***

# SOA: Integración no intrusiva...



# SOA: Modelos de composición



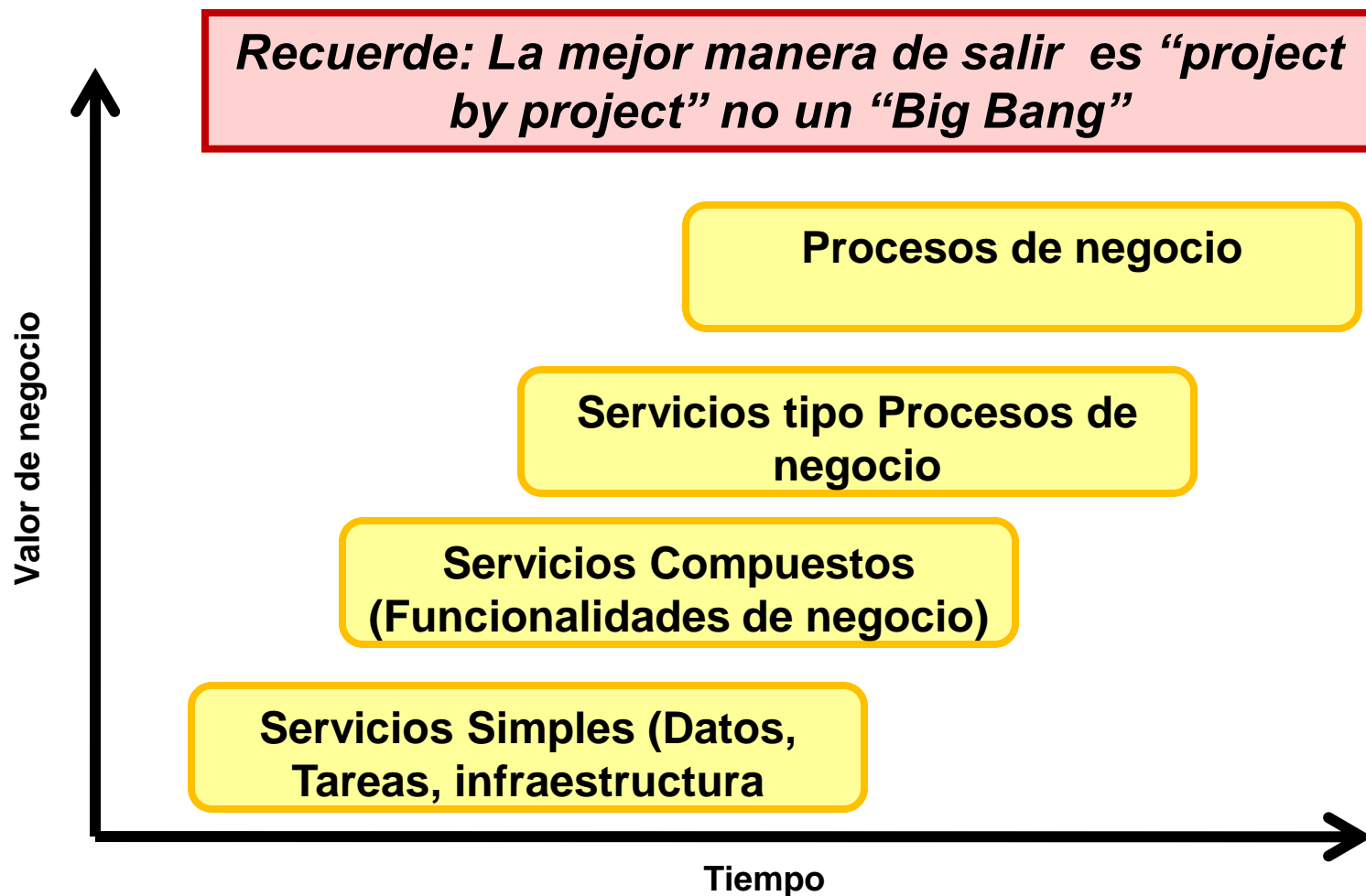
## SOA: Modelos de composición(2)

- La **composición** de se puede dar a nivel de: **Proceso de negocio, servicios de negocio** y **a nivel del proveedor del servicio**.
- La composición a nivel **de proceso** se debe expresar en lenguajes de orquestación tales como **BPEL**
- La composición a nivel de **servicios de negocio** se puede expresar en **lenguajes de orquestación propietarios** o **BPEL**
- La composición a **nivel de proveedor de servicio** por lo general se da a nivel de **programación (Java, C#, etc.)**

## SOA: Modelos de composición(2)

- Finalmente es importante anotar que si **como parte de una misma transacción** se llaman **más de un servicio** provistos por **la misma plataforma**, se debe **construir al interior** de esta ultima **un servicio de mayor nivel de granularidad** que los agrupe (Evita overhead – chatty, manejo transaccional y excepciones)
- Necesitamos de una **infraestructura** que nos **ofrezca un contenedor de ejecución** de **composiciones notadas en BPEL** o lenguajes similares.

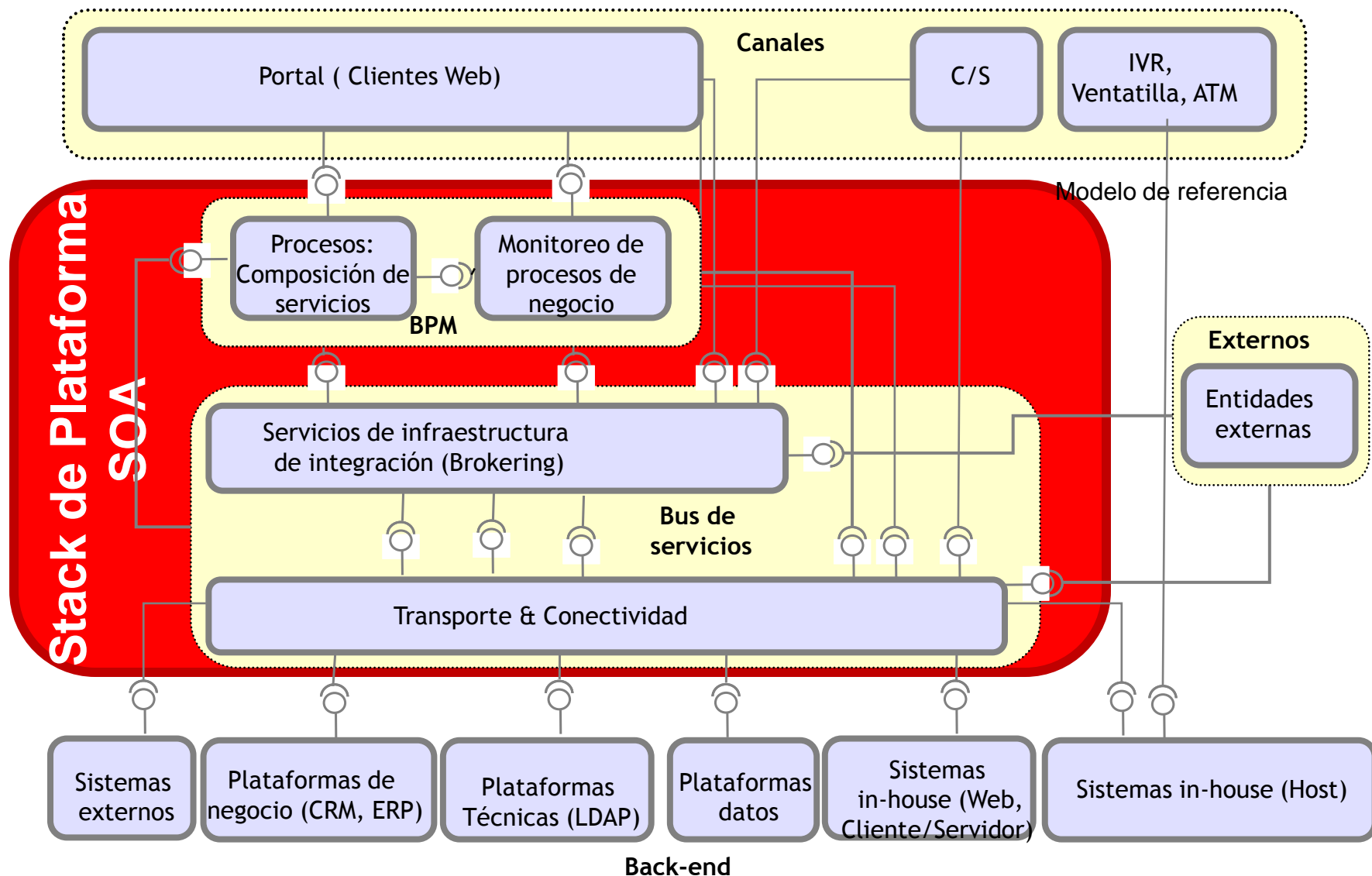
# SOA: Roadmap implementación servicios



# SOA: Hasta ahora que llevamos ...

- SOA debe verse dentro del contexto de un **modelo Conceptual** multicapas (Canales, procesos, servicios, proveedores)
- Existen **diferentes categorías de servicios** (Presentación, negocio, datos, infraestructura) de acuerdo su función y **granularidad**.
- Existen diferentes tipos de **composición de servicios**.
- La **flexibilidad se presenta en cualquier nivel**.
- Necesitamos de una **infraestructura** promueva la **integración y publicación de servicios** de manera **no intrusiva**, y ejecute **composiciones de servicios en notaciones BPEL** potenciando de esta manera el estilo de arquitectura SOA.

# SOA: Modelo Lógico multicapas

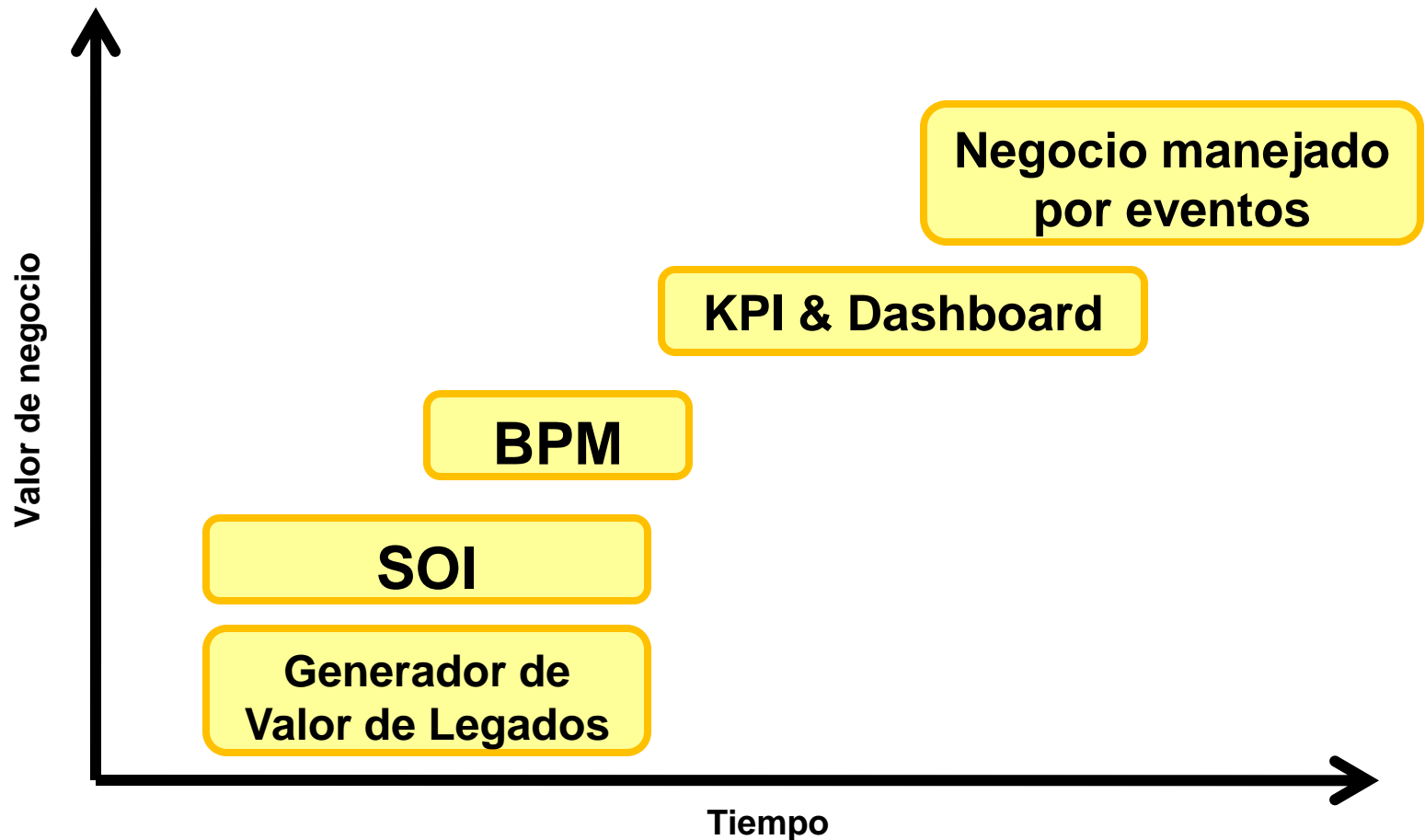




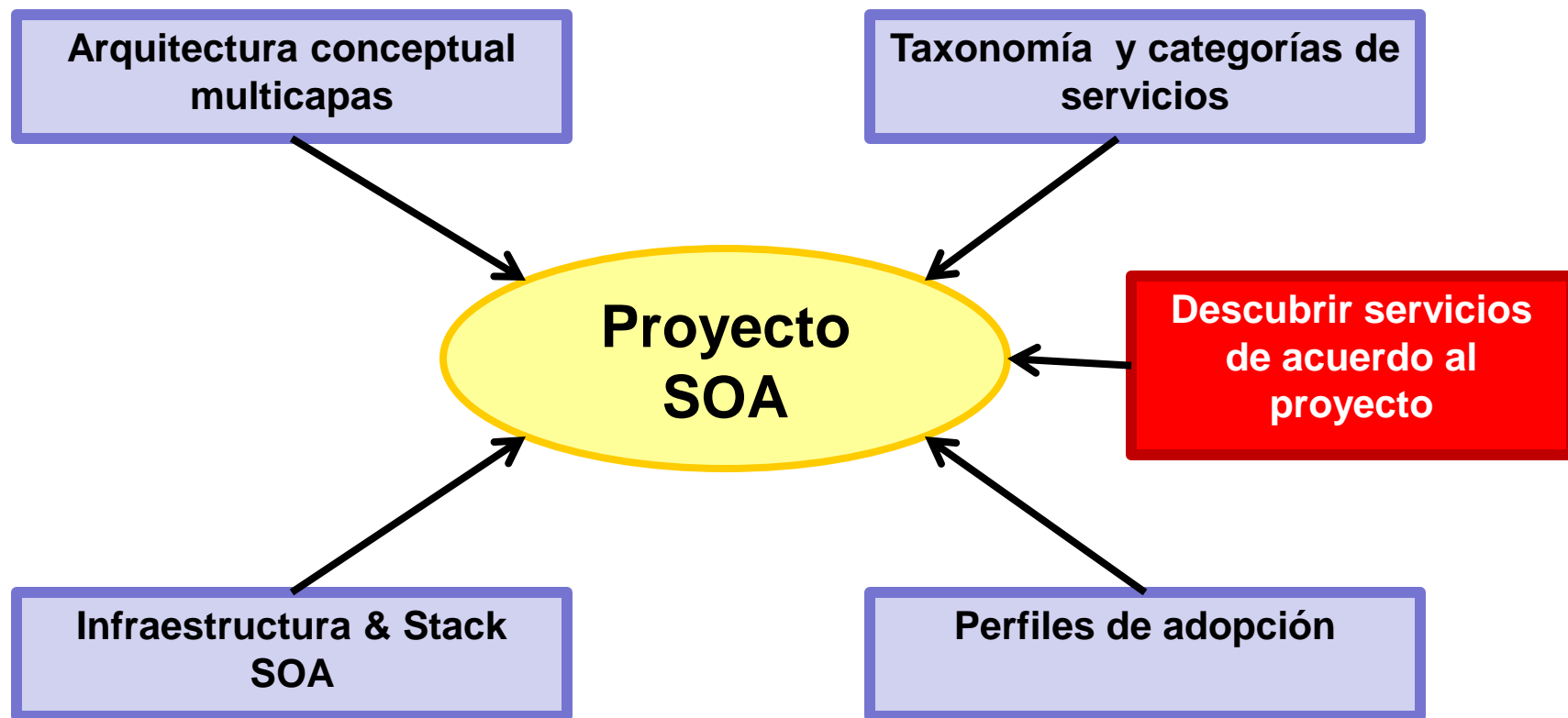
# SOA: Con estos modelos podemos soportar los diferentes perfiles de implementación SOA

- **SOI (Services Oriented Integration)**
  - Integración orientada a servicios
- **BPM (Business Process Management) & Composite Apps.**
  - Composición de aplicaciones via procesos de negocios
  - Necesidad de portafolio
- **Generador de valor para aplicaciones legadas**
  - Adaptadores a plataformas legadas (Visión bottom-up)
- **Medición y valoración de procesos**
  - KPI, Dashboard
  - BAM vs. BI
- **Sistema nervioso generador de eventos**
  - Clase mundial, Reaccionar a eventos de negocio

# SOA: Roadmap tradicional de adopción



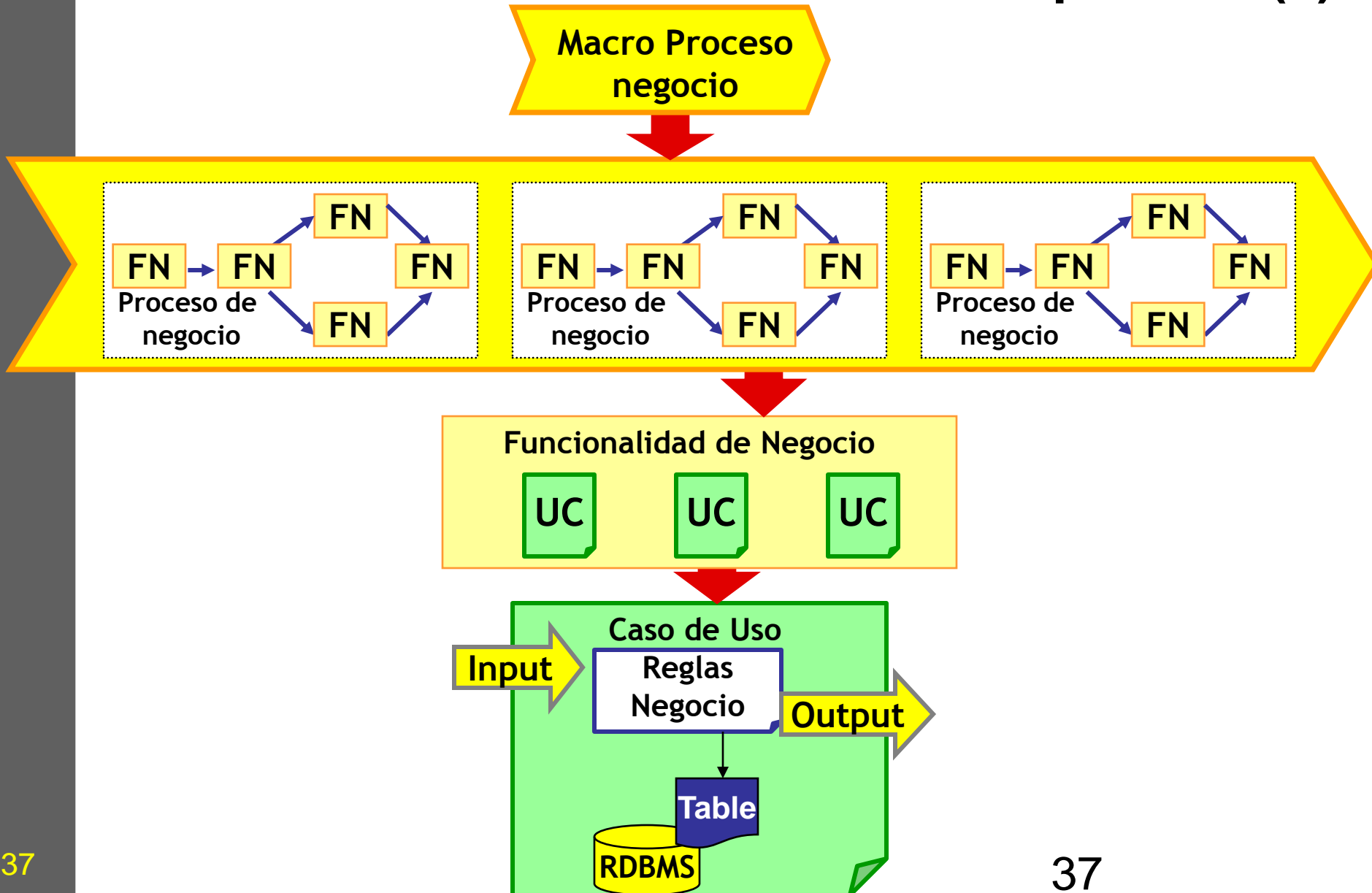
# SOA: ¿Qué nos falta?



# SOA: Los servicios son el combustible de una infraestructura SOA.

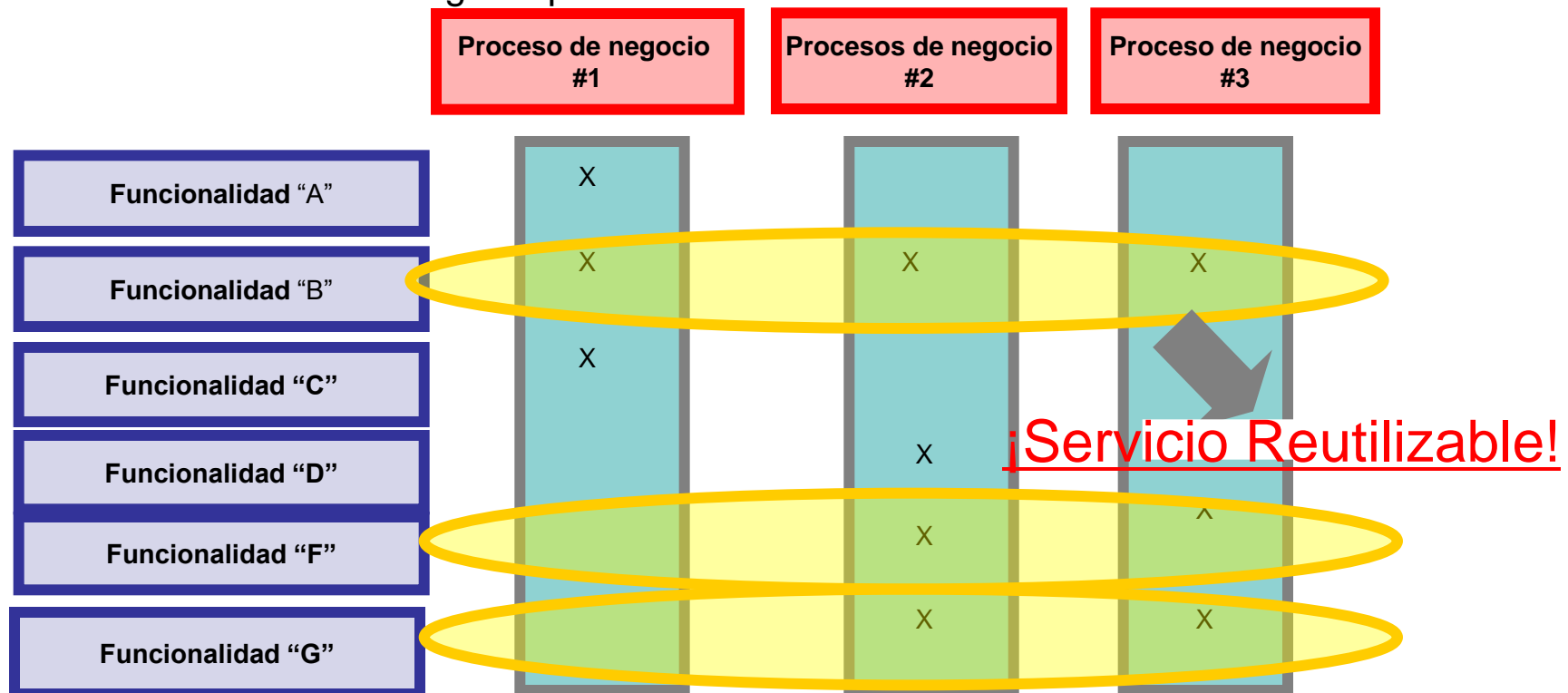
*De nada sirve tener el mejor enfoque de arquitectura, la mejor plataforma SOA sino se cuenta con un enfoque metodológico que permita descubrir servicios que estén alineados a la estrategia de negocio.*

# SOA: Descubrimiento de servicios Top-Down (1)



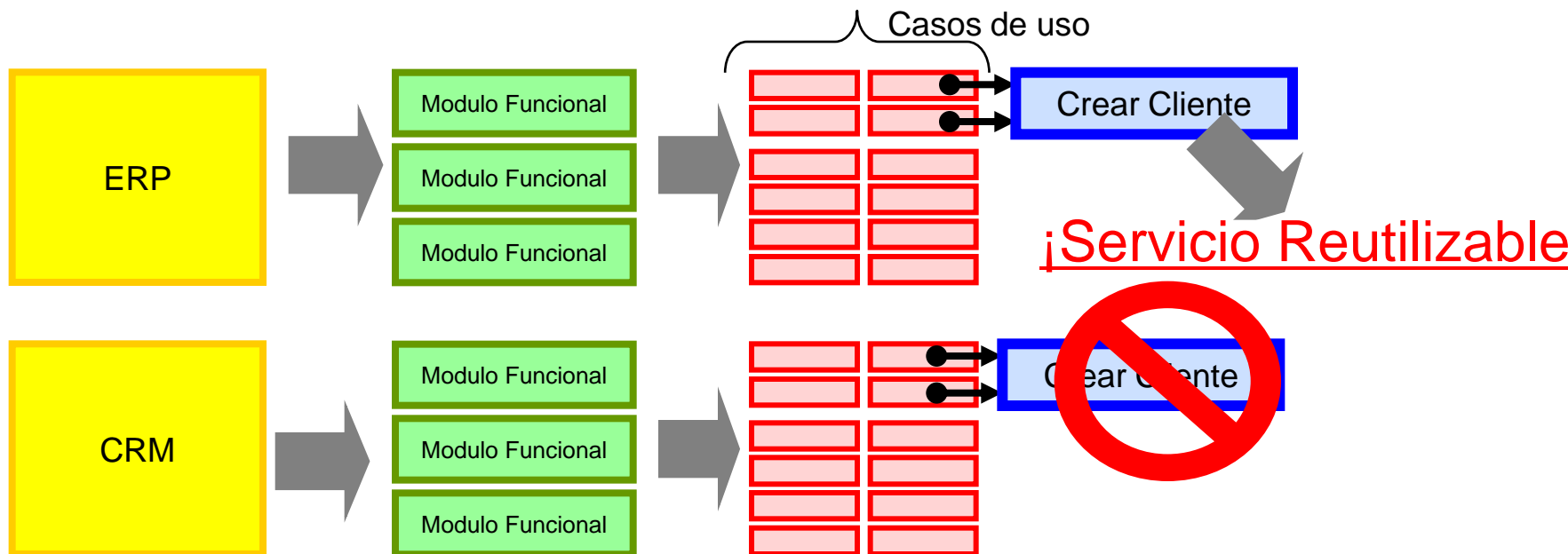
# SOA: Descubrimiento de servicios Top-Down (2)

- Parte desde los **procesos de negocio**
- Considera el mapeo: **Business processes** → **Business Functionalities** → **Requirements** → **Use cases**
- Este enfoque aplica bastante bien para iniciativas **BPM** basadas en la orquestación de **funcionalidades** de negocio publicadas como servicios

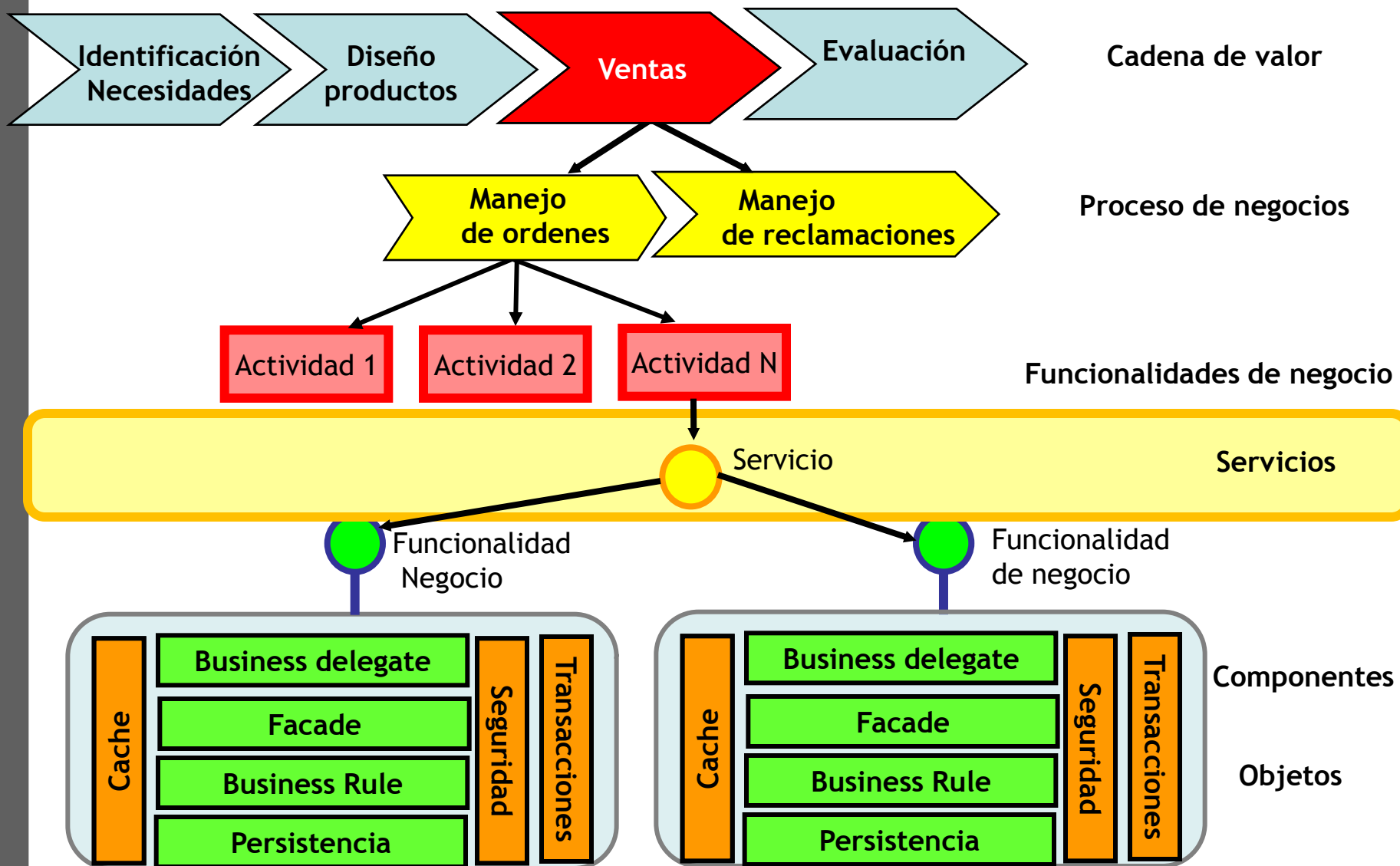


# SOA: Descubrimiento de servicios Bottom-up

- Parte desde el **ecosistema de aplicaciones de la empresa**: CRM, ERP, Legacy Systems,
- Considera el mapeo: **Information System** → **Modulo funcionals** → **use cases** → **Business Functionalities**
- Este enfoque funciona bastante bien cuando se desea **iniciar un proyecto de integración basado en servicios** o en la consolidación de un **portafolio de servicios**.
- Este enfoque permite **identificar y eliminar funcionalidades duplicadas** (Promueven inconsistencia de datos y complican la mantenibilidad del sistema)

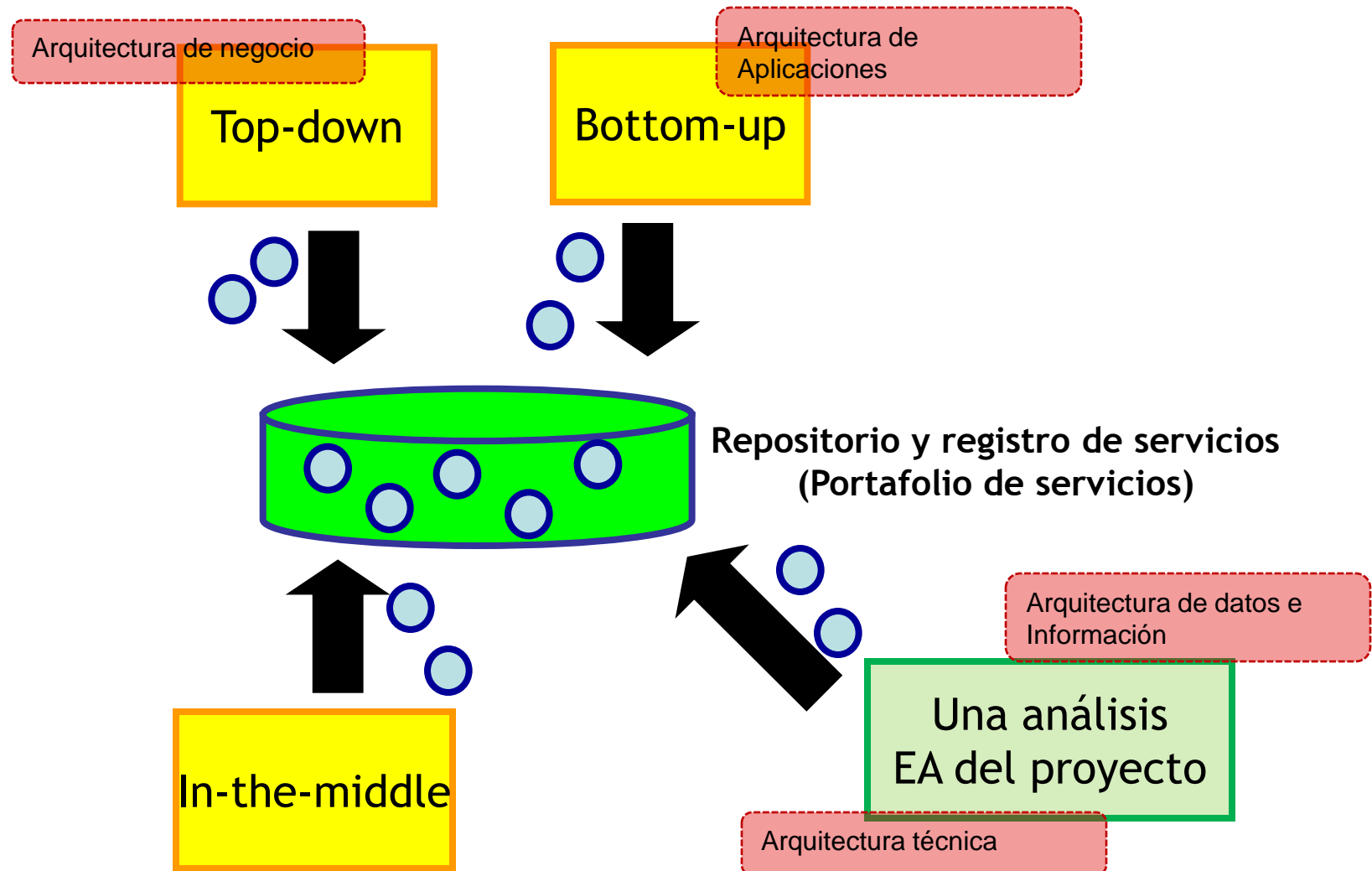


# SOA: Descubrimiento de servicios in-the-middle





# SOA: ¿Top-Down, Bottom-Up & In-the middle serán suficientes para descubrir servicios?



# SOA: By the way... No se olvide de los contratos SOA.

## CONTRATO DE UN SERVICIO

### Información general

- Version
- Dueño
- RACI ( Responsable, Accountable, consulted, informed)
- Tipo ( Presentación, Proceso, Datos, Funcionalidad, integración, etc.)

### Requerimientos funcionales

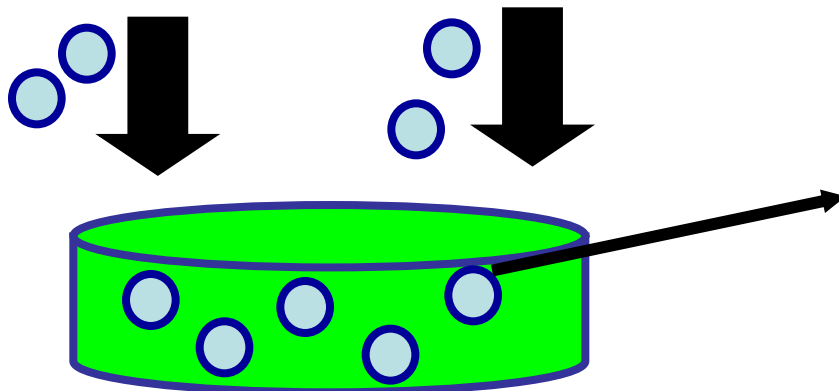
- Precondiciones
- Postcondiciones
- Operaciones
- Reglas de negocio

### Requerimientos no funcionales

- Seguridad
- Transacciones
- SLA
- QoS
- Semánticas

Top-down

Bottom-up

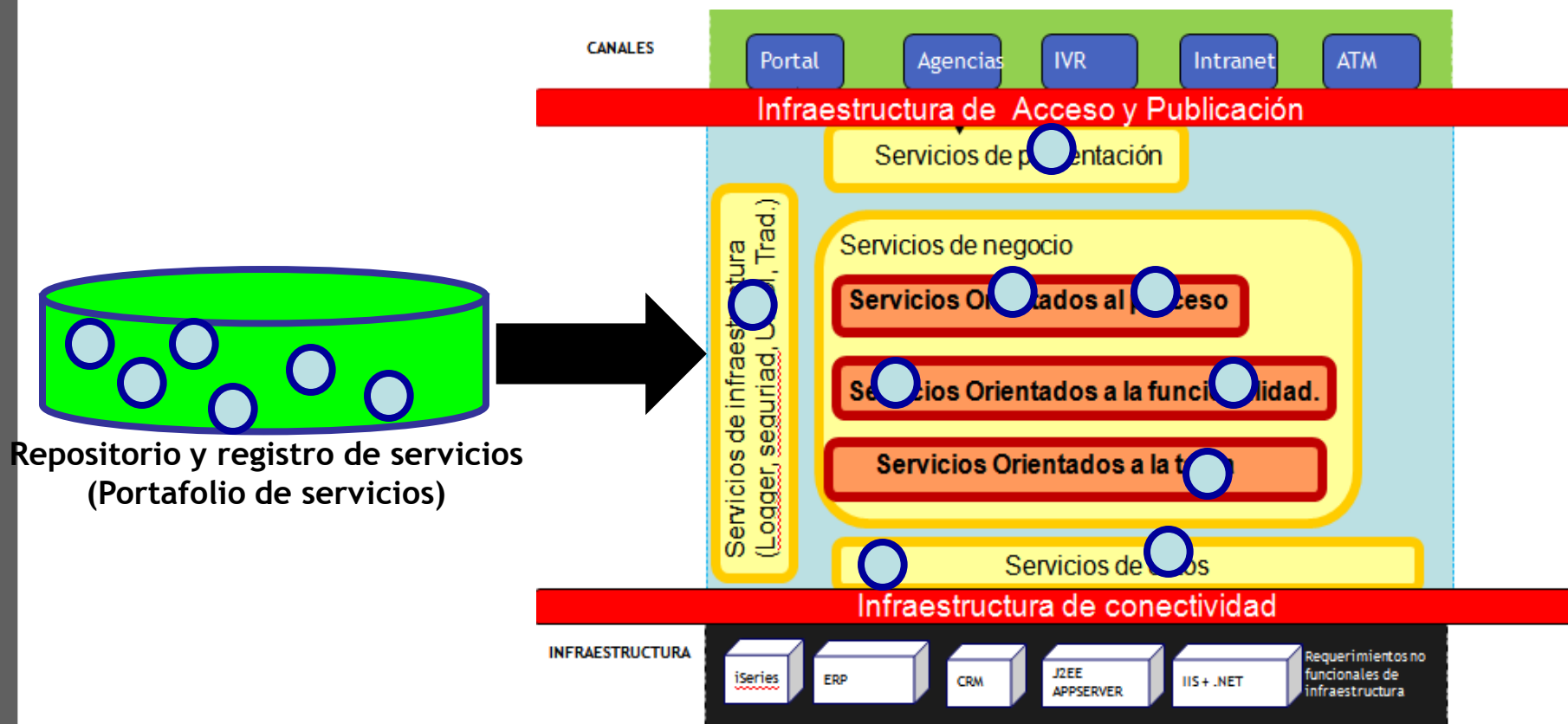


Repositorio y registro de servicios

No se deje llevar por la simplicidad de los SOAP/ webservices ( Qué no le pase a usted



# SOA: Llegando a SOA desde un descubrimiento de servicios basado en EA



# Agenda

1. Motivación & Contexto

2. SOA: Estructura Conceptual



3. EA: Descubriendo servicios

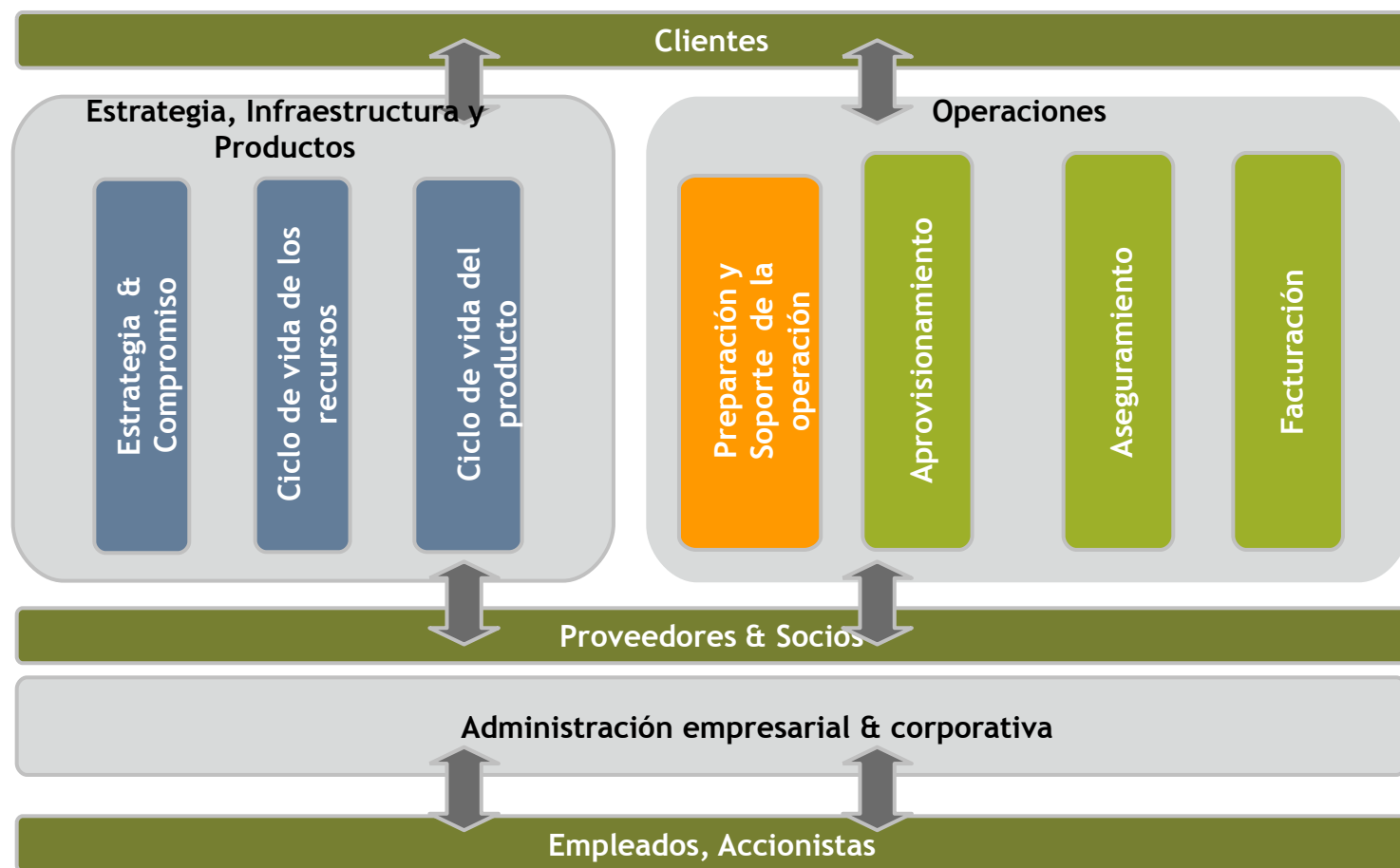
4. Conclusiones

# Descubrimiento de Servicios: Arquitectura de negocio

- Funcionalidades de negocio comunes a varios procesos

	Proceso de negocio #1	Proceso de negocio #2	Proceso de negocio #3	Proceso de negocio #N
Funcionalidad 1	X			
Funcionalidad 2		X		
Funcionalidad 3		X		
Funcionalidad 4	X	X	X	
Funcionalidad 5				X
Funcionalidad 6	X	X	X	
Funcionalidad 7				X
Funcionalidad N	X	X		X

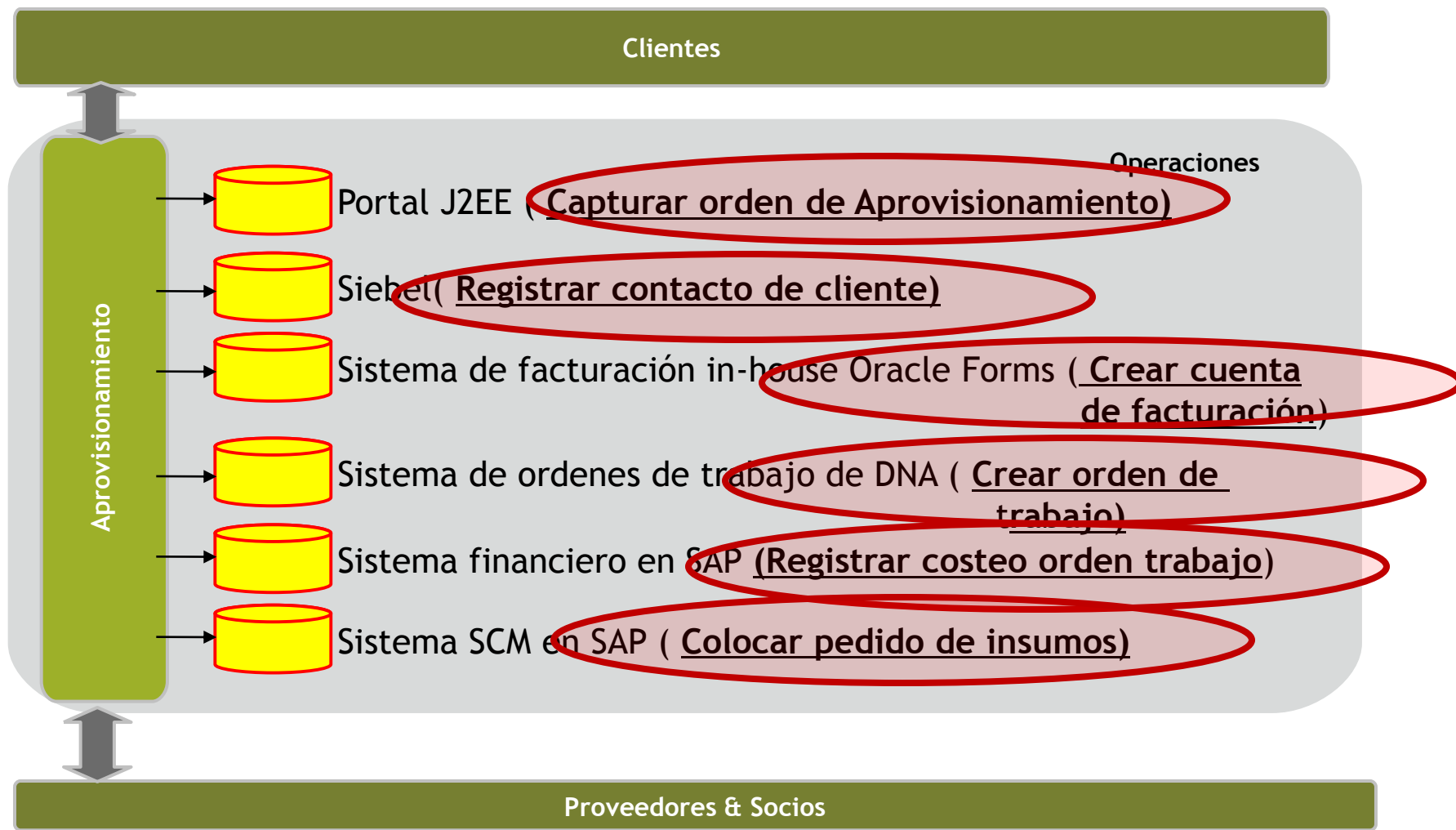
# Descubrimiento de Servicios: Arquitectura de negocio (Recordemos el modelo de referencia)



Idea tomada y derechos reservados desde Telemanagement Forum

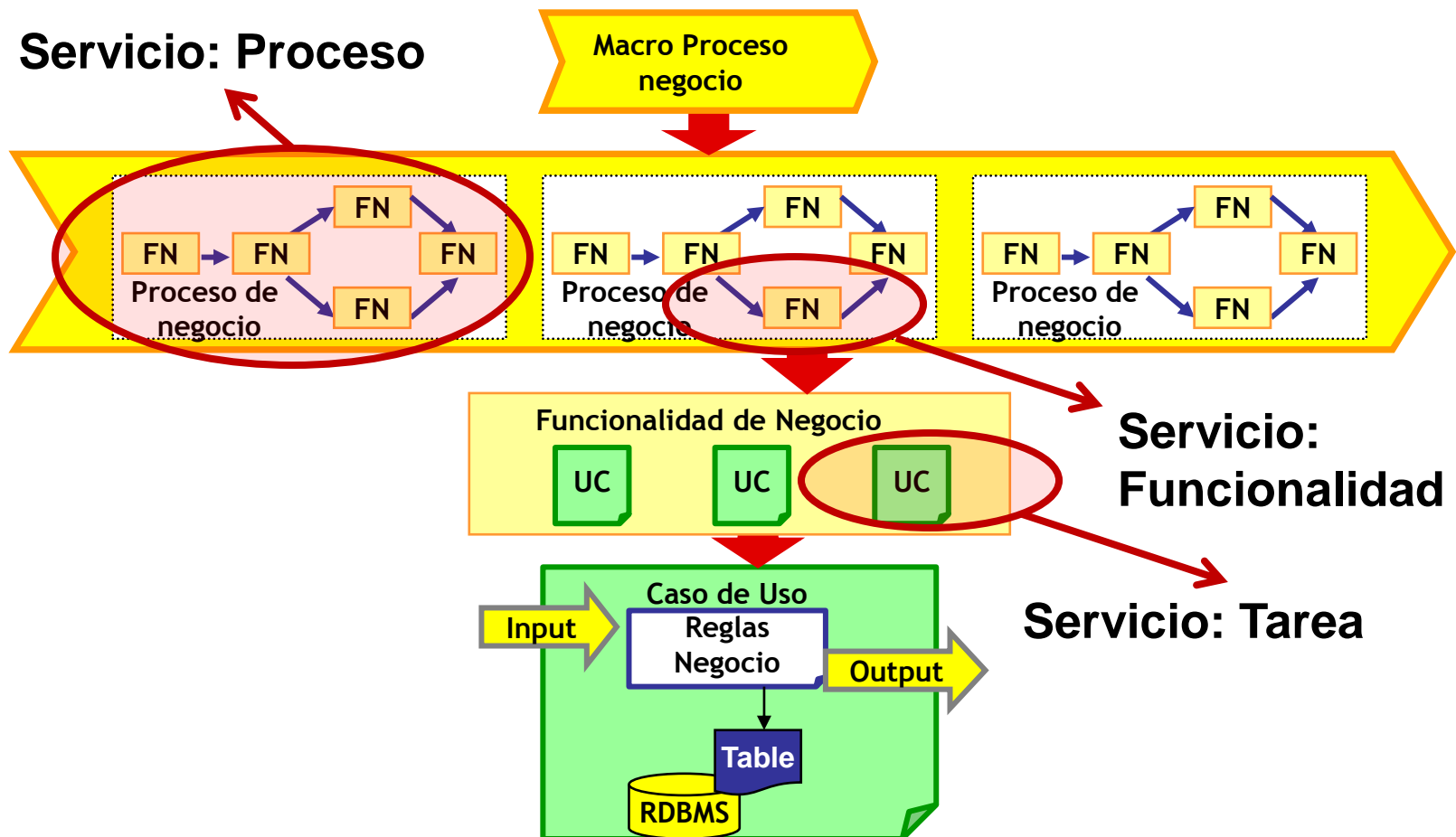
# Descubrimiento de Servicios: Arquitectura de negocio

- Integración orientada y basada en procesos de negocios



# Descubrimiento de Servicios: Arquitectura de negocio

- Descomposición funcional





# Descubrimiento de negocio: Arquitectura de información y datos

- Identificación de sistemas dueños de datos. **Sistemas de información**

	Siebel	SAP	Vision +	FACT	Systematics	Remedy Trouble Ticket
Cliente	RW	RW	RW	R	RW	R
Cuentas	R		RW	RW	RW	
Tarjetas			RW	RW	RW	
Facturas	R	RW	RW	R	R	
Pagos	R	RW	RW		RW	

**Entidades de negocio**

Más de un sistema es dueño de la misma entidad: Problemas de calidad e inconsistencia de datos.

# Descubrimiento de negocio: Arquitectura de información y datos

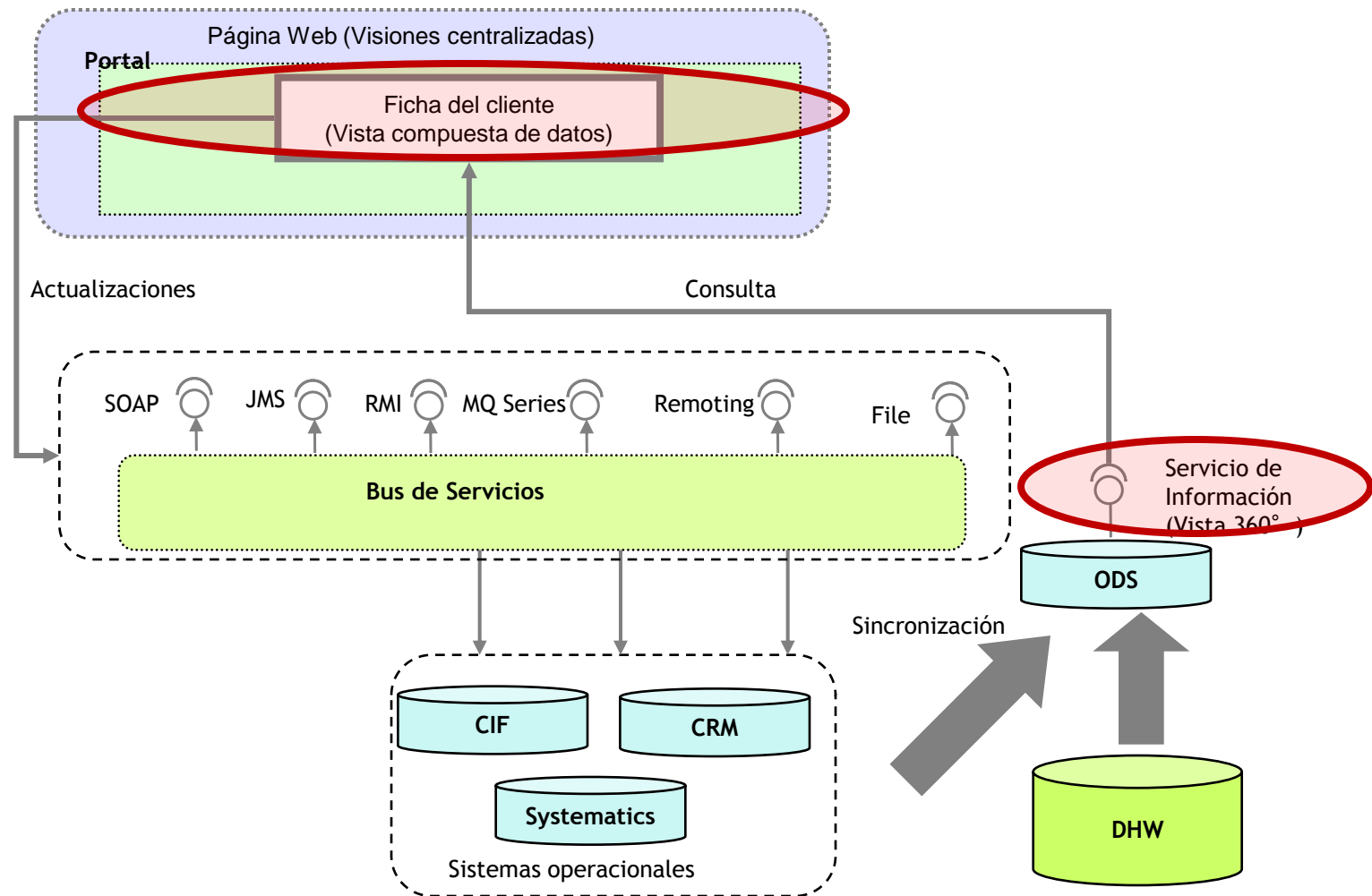
- Identificación de sistemas dueños de datos.

	Siebel	SAP	Vision +	FACT	Systematics	Remedy Trouble Ticket
Cliente	RW	R	R	R	R	R
Cuentas	R		RW	R	R	
Tarjetas			RW	R	R	
Facturas	R	RW	R	R	R	
Pagos	R	RW	R		R	

**Un sólo dueño de la entidad: SAP debe publicar un servicio de datos que permite manejar el ciclo de pagos**

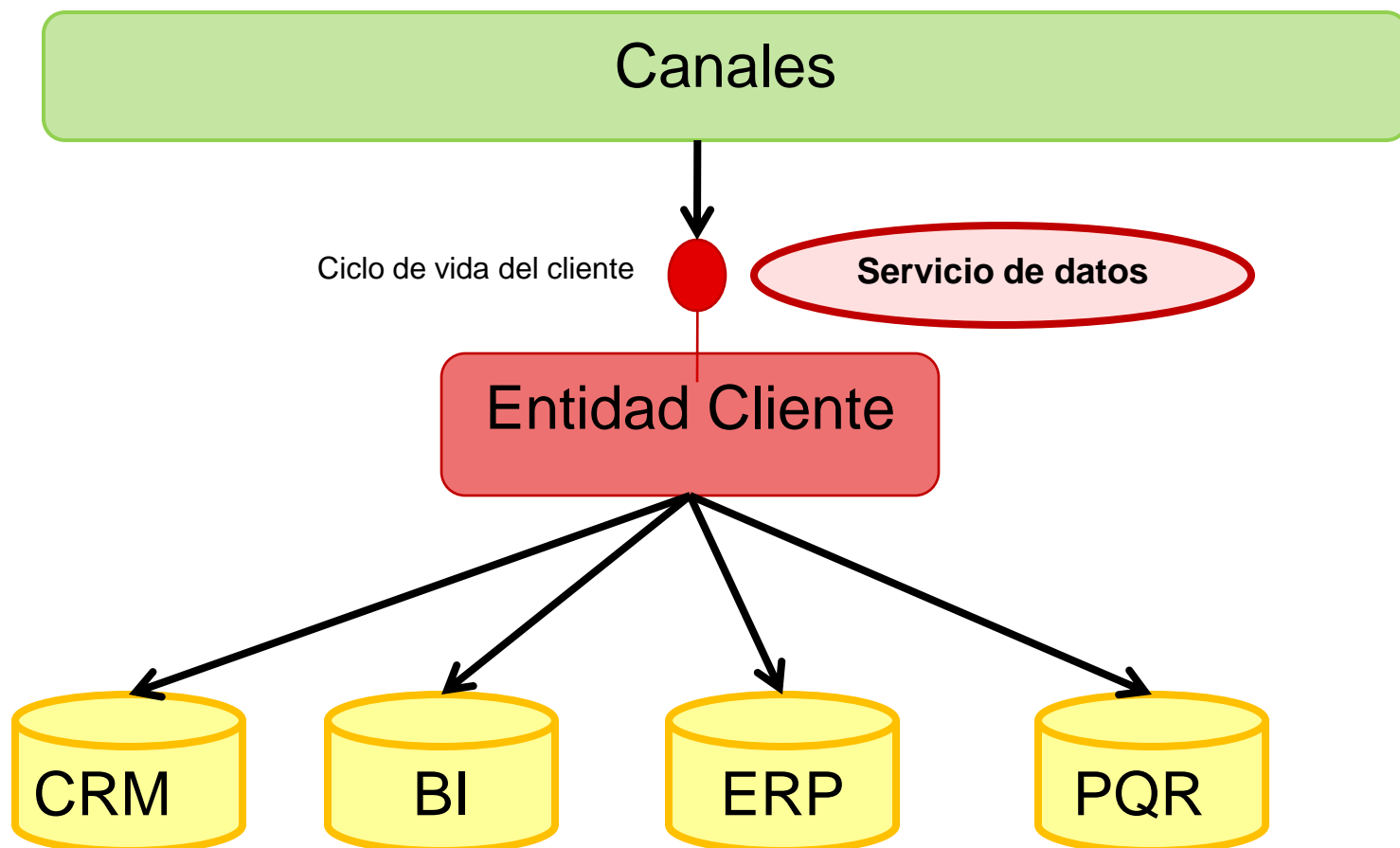
# Descubrimiento de negocio: Arquitectura de información y datos

- Vistas consolidadas de datos (Clientes, Empleado, etc.)



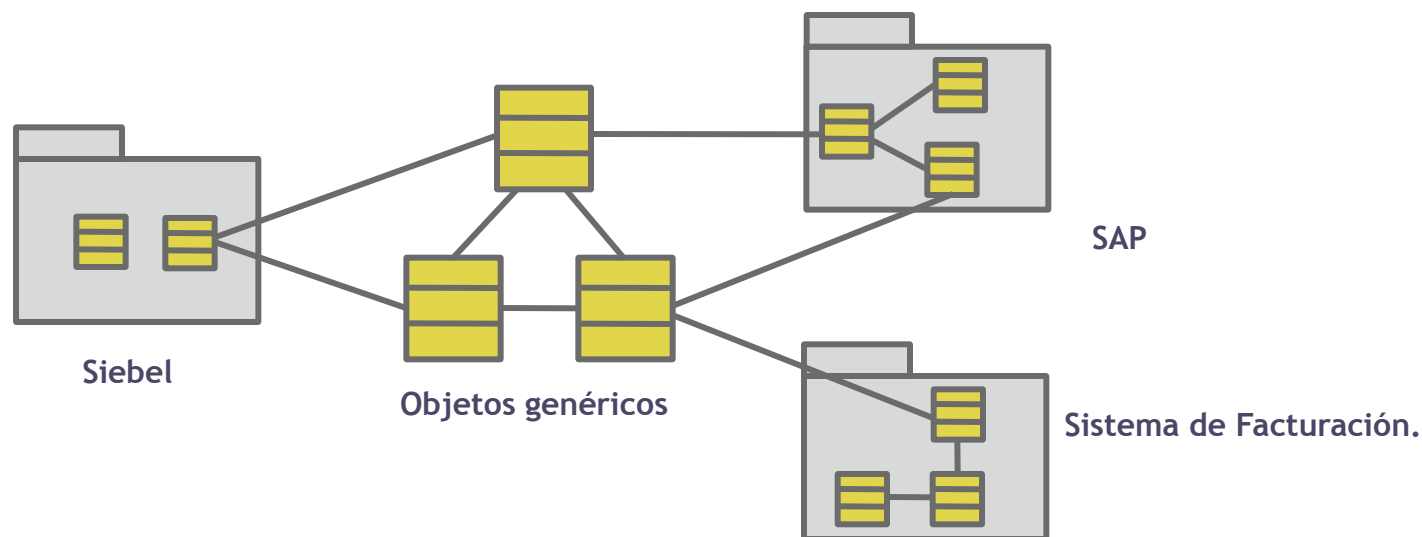
# Descubrimiento de negocio: Arquitectura de información y datos

- Agregación virtual de datos



# Descubrimiento de negocio: Arquitectura de información y datos

- Modelo de objetos genéricos (Canónicos)



**Implementar servicios de transformación a canónicos reutilizables**

# Descubrimiento de negocio: Arquitectura Aplicación

- Identificación de aplicaciones dueñas de funcionalidades

Funcionalidad	Siebel	SAP	Vision +	FACT	Systematics	Remedy Trouble Ticket
Actualizar Cliente	X	X	X			X
Actualizar Cuenta	X		X	X		
Consultar Tarjeta			X	X		
Anular Factura		X	X			
Aplicar Nota credito			X	X	X	

**Funcionalidades de negocio**

**Más de un sistema es dueño de la misma funcionalidad: Problemas de mantenibilidad y consistencia.**

# Descubrimiento de negocio: Arquitectura Aplicación

- Identificación de aplicaciones dueñas de funcionalidades

Funcionalidad	Siebel	SAP	Vision +	FACT	Systematics	Remedy Trouble Ticket
Actualizar Cliente	X					
Actualizar Cuenta			X			
Consultar Tarjeta			X			
Anular Factura		X				
Aplicar Nota credito			X			

**El dueño de la funcionalidad debe publicar un servicio**

# Descubrimiento de negocio: Arquitectura Aplicación

- Identificación de servicio a partir de la matriz de integración (Funcionalidades Similares con nombres diferentes)

	Siebel	SAP	Vision+	Systematics	Fact
Siebel	NA	Crear Pedido (ETL), Generar Factura (SOAP)	Actualizar Cliente(JT400)	Crear Cuenta ahorros ( MQ Series)	
SAP		NA			
Fact		Crear Pedido (BAPI)	Actualizar Cuenta Habiente (MQ Series)	Crear Cuenta Ahorros (Sockets)	
Systematics		Generar Factura (ETL)	Aplicar Novedad Cliente(ETL)	NA	
Vision+			Modificar teléfono cliente ( Sockets)		

**La misma funcionalidad bajo diferente estructura y nombre**



# Descubrimiento de negocio: Arquitectura Aplicación

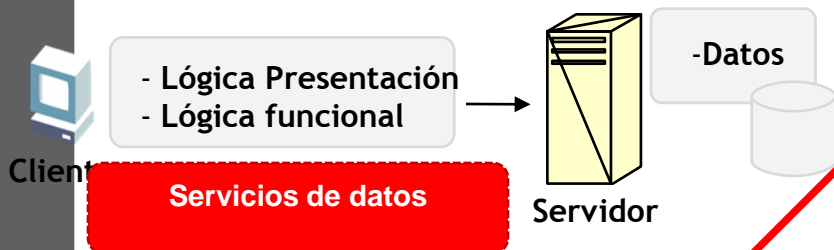
- Identificación de servicio a partir de la matriz de integración (Unificar tecnologías de acceso a las funcionalidades)

	Siebel	SAP	Vision+	Systematics	Fact
Siebel	NA	SOAP, BAPI, ETL	Sockets, JT400, MQ Series	ETL, Sockets	
SAP		NA			
Fact		ETL	DataQueue, JMS, Sockets	Sockets, SOAP	
Systematics		Sockets + BAPI	ETL	NA	
Vision+			Sockets	DBLink, Sockets	NA

Por Gobernabilidad se debería tratar de unificar métodos de acceso

# Descubrimiento de negocio: Arquitectura Aplicación

## Modelo Cliente/Servidor dos niveles



## Modelo host



1990

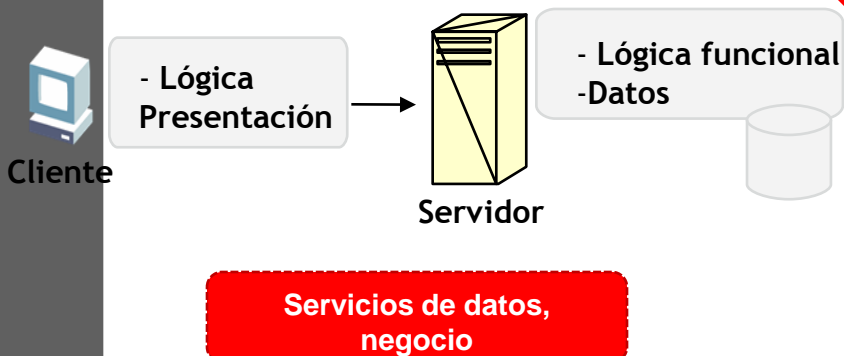
2008

1980

1970

2005

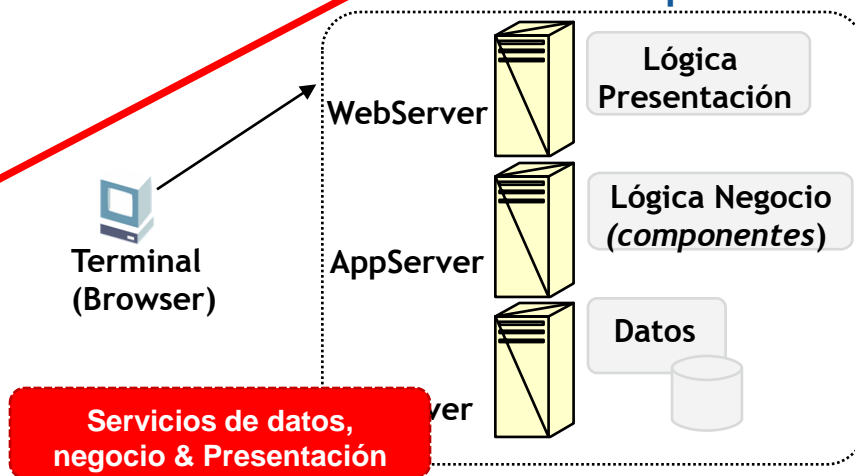
## Modelo Cliente/Servidor modificado



1998

Terminal  
(Browser)

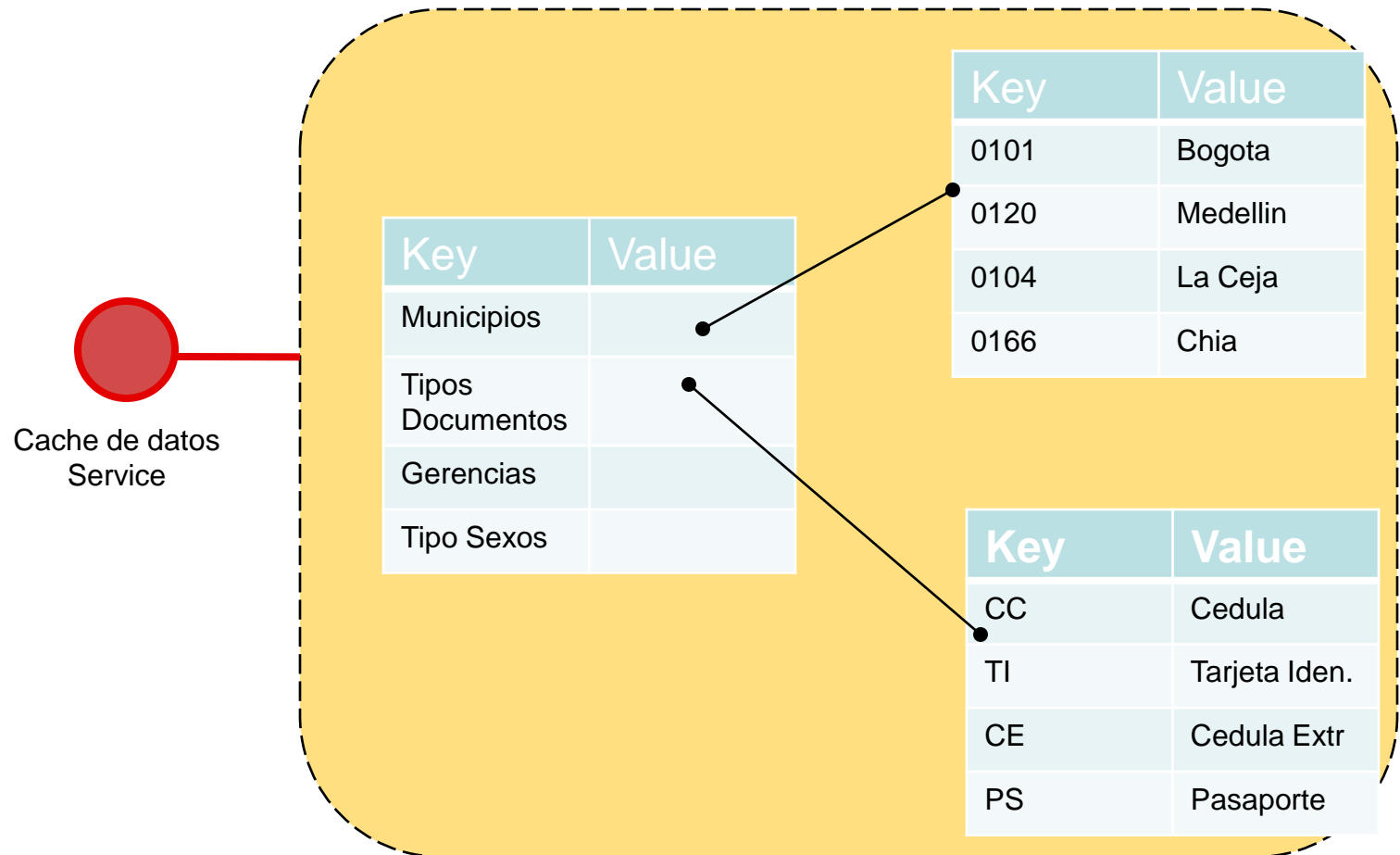
## Multicapas





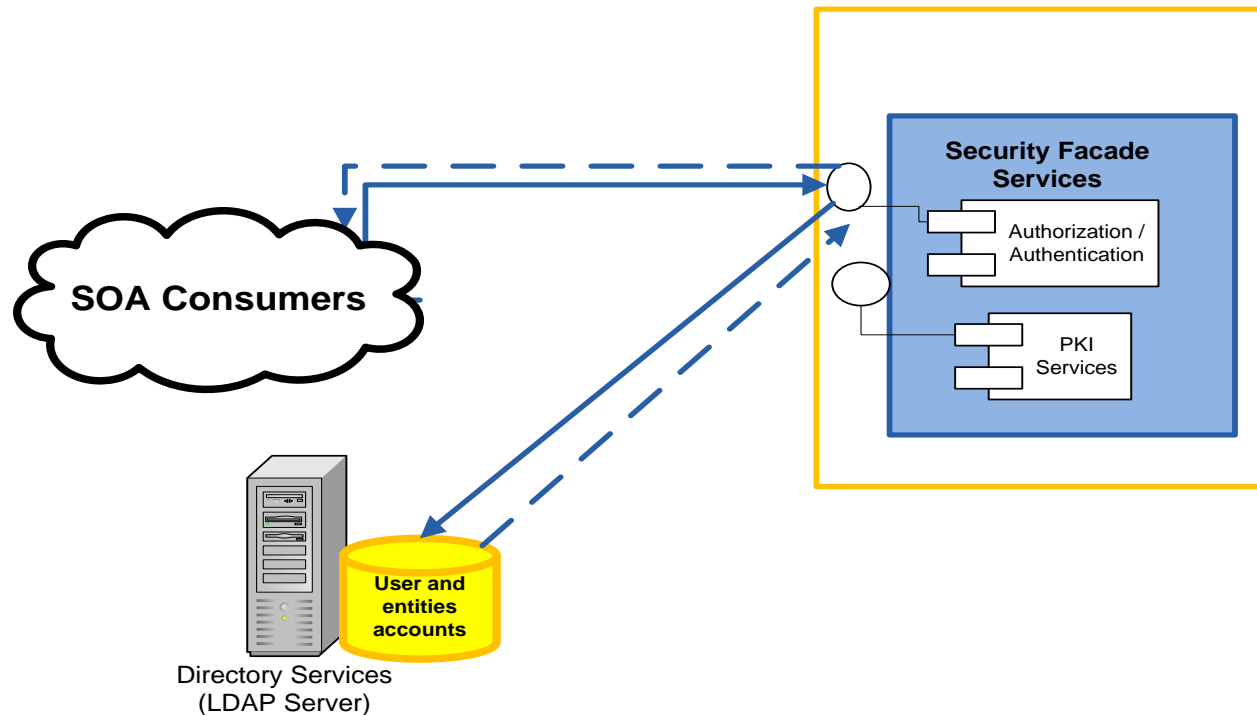
# Descubrimiento de negocio: Arquitectura Técnica

- Servicio caches



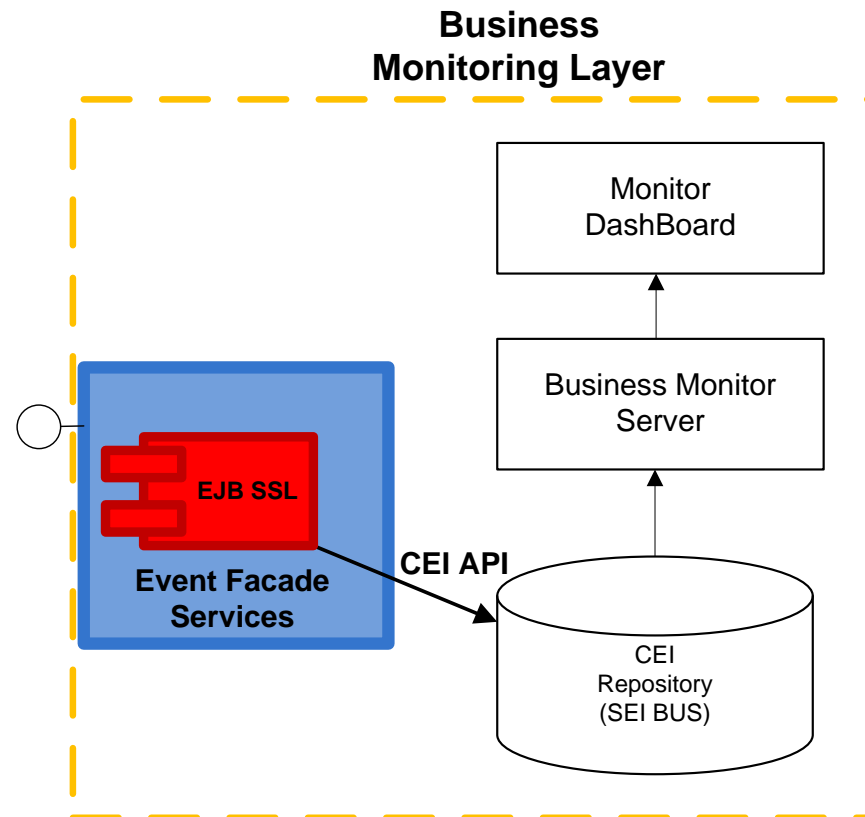
# Descubrimiento de negocio: Arquitectura Técnica

- Servicios de seguridad



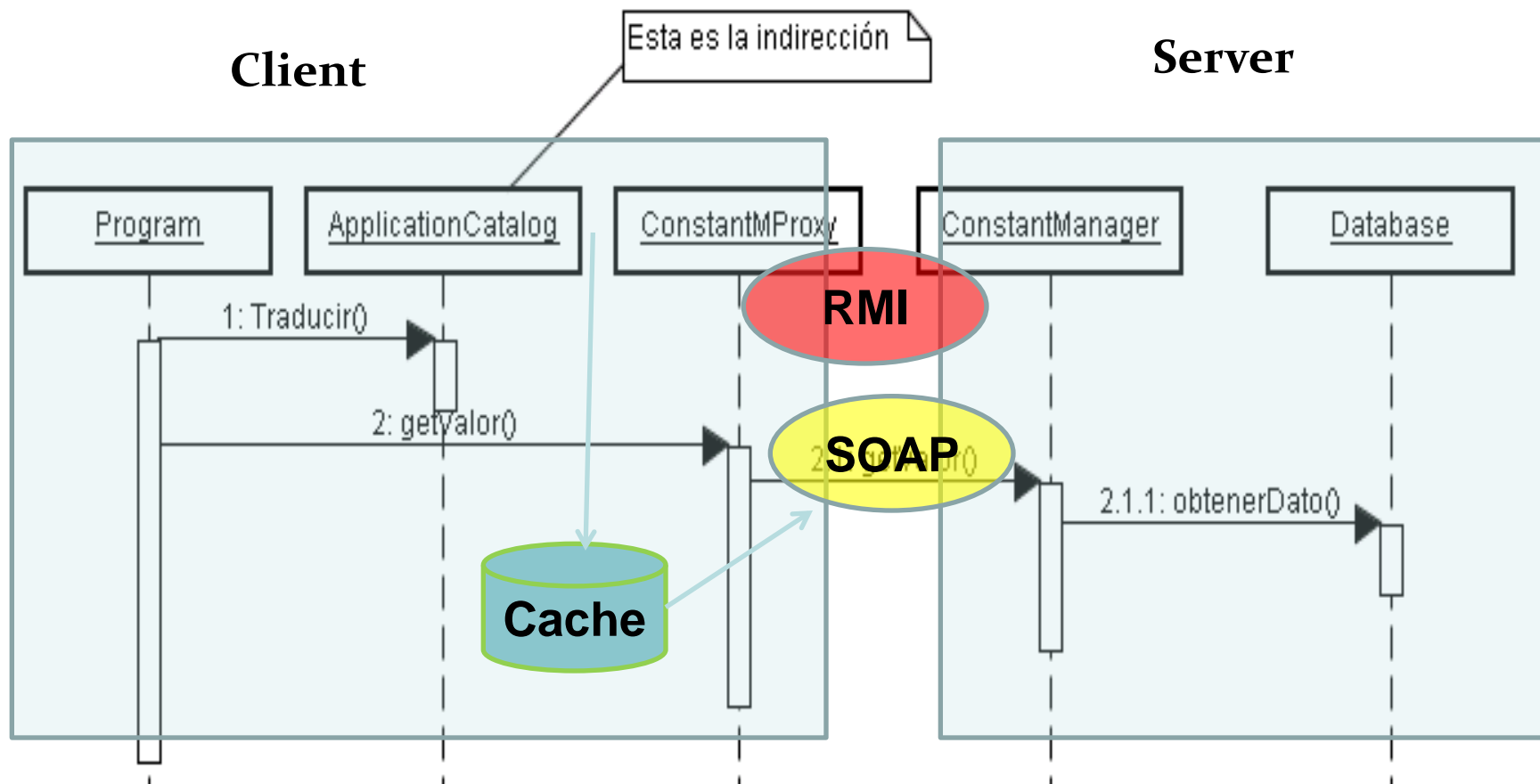
# Descubrimiento de negocio: Arquitectura Técnica

- Servicios de Monitoreo



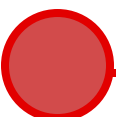
# Descubrimiento de negocio: Arquitectura Técnica

- Servicio Manejador de Constantes



# Descubrimiento de negocio: Arquitectura Técnica

- Servicio Traductor de mensajes

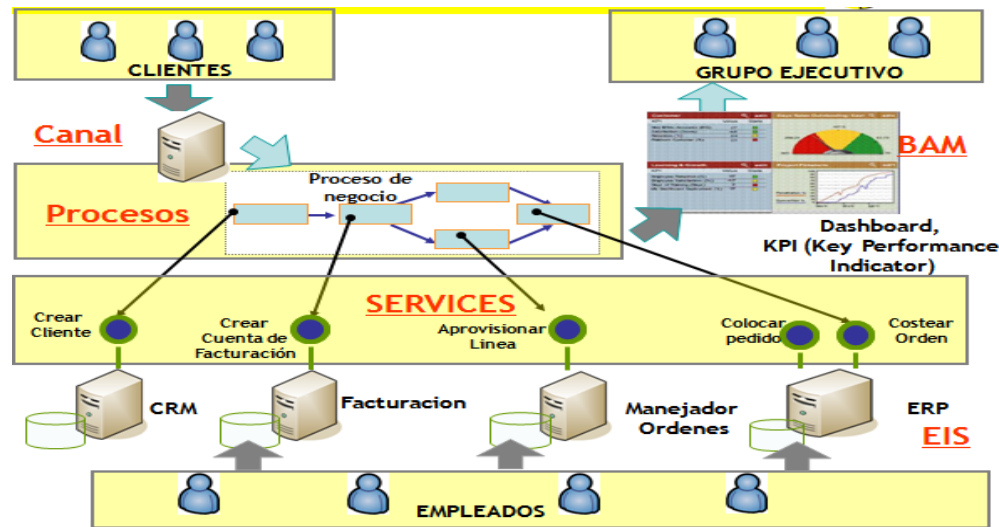


Traductos Mensajes

Aplicacion	Codigo	Descripcion	Tipo
Siebel	SB-0001	Cliente no corportativo	Funcional
SAP	SP-U89	BAPI no correctamente registrado	Tecnico



# Descubrimiento de negocio: Arquitectura Técnica



Seguridad

Caches

Auditoria

Monitoreo

Constantes

Traductor

**Servicios de infraestructura**

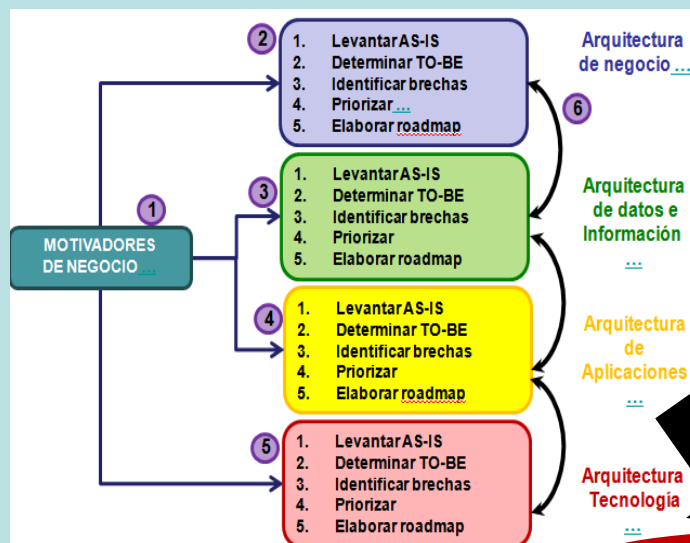
# Contexto: EA + SOA = SOA exitoso

## Iniciativa o Proyecto Corporativo

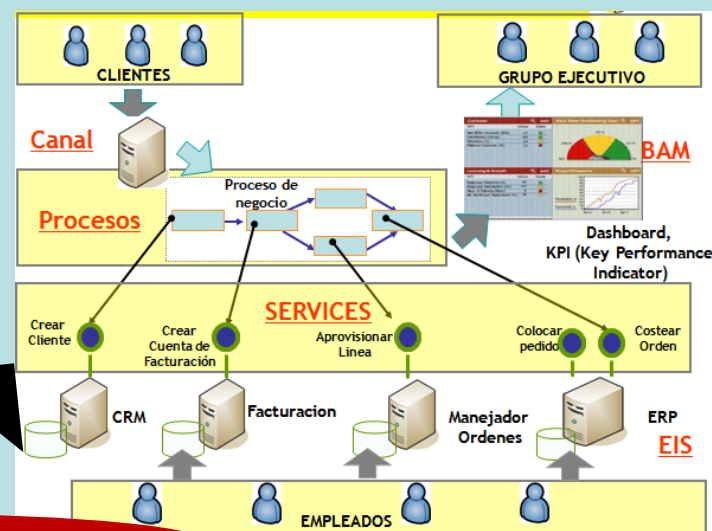
**¿QUÉ?**

**¿CÓMO?**

### Arquitectura Empresarial



### SOA: Arquitectura de Solución



**Portafolio de servicios alineado al negocio**

# Agenda

1. Motivación & Contexto

2. SOA: Estructura Conceptual

3. EA: Descubriendo servicios

➔ 4. Conclusiones

# Conclusiones

- **SOA** permite crear un **arquitectura de solución** altamente **flexible** y **alineada a la estrategia de negocio** cuando **parte de una análisis EA** previamente realizado por la organización.
- **EA es un excelente enfoque para descubrir servicios**, por lo tanto debe ser el punto de partida de cualquier proyecto SOA, independientemente del **perfil de implementación**.
- **Los servicios** no sólo se limitan **a funcionalidades de negocio reutilizables**, en contra parte debe tomarse ventaja de **servicios de presentación, infraestructura, datos; todos ellos también reutilizables**.
- **El estilo de arquitectura** que propone SOA + **Infraestructura** SOA + **Enfoque metodológico** para **descubrir servicios alineado al negocio** + **Arquitectura de referencia** + Gobernabilidad + Metodología = Éxito
- SOA < > Webservices < > Tecnología < > ESB < > BPM

# ¿Preguntas?