



HEINSOHN
BUSINESS TECHNOLOGY

SOA

Arquitectura Orientada a Servicios con WCF



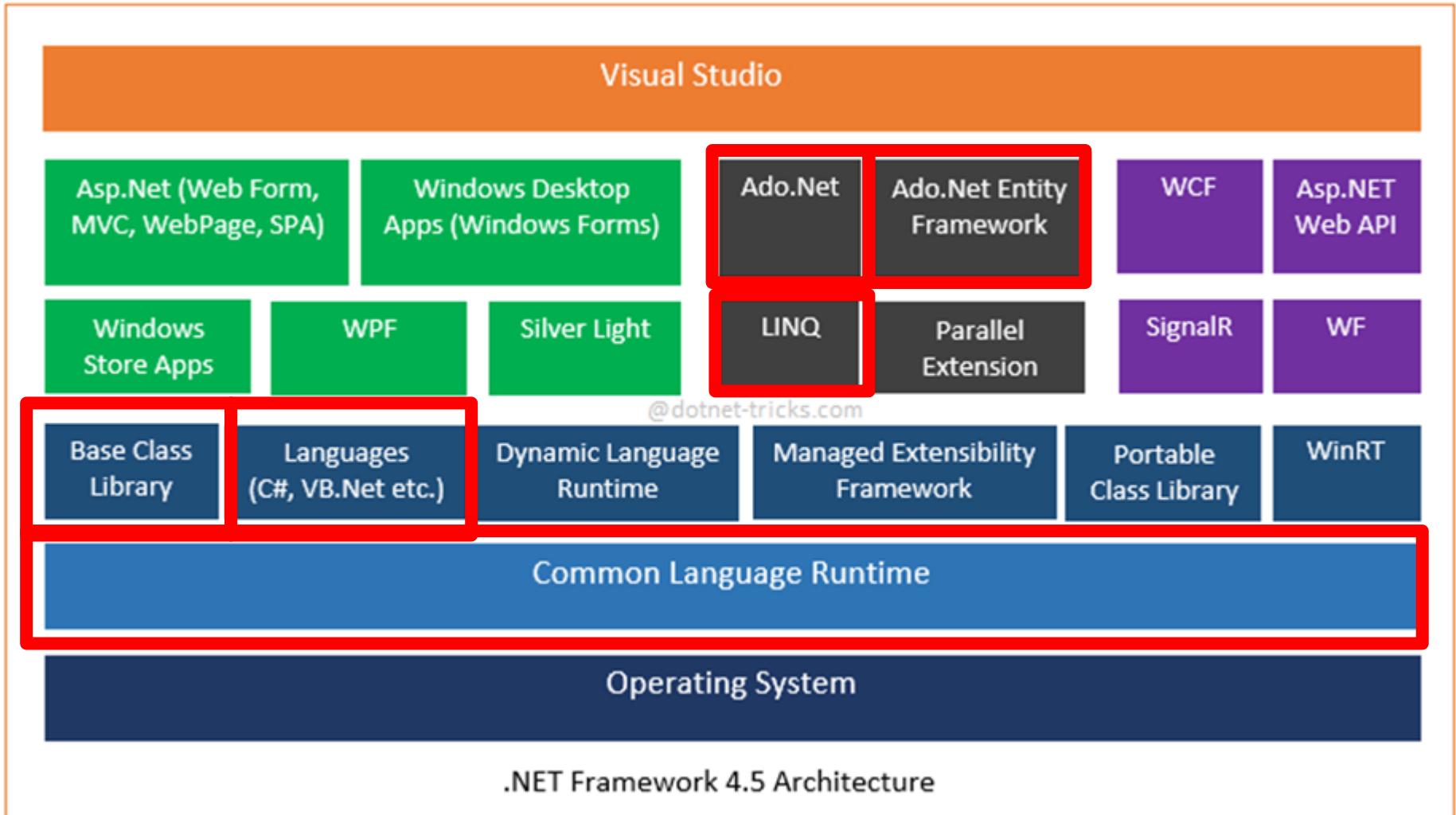
Microsoft Partner

Gold Software Development

Gold Web Development

Gold Software Asset Management

.Net Framework 4.5 Architecture



- ¿Cómo es el área de TI en su organización?
- ¿Las diferentes áreas de su organización son **islas**?
- ¿Usted ha realizado **grandes inversiones** en proyectos de tecnología sin obtener los **resultados** esperados?
- ¿Ha sentido que la tecnología le trae más **problemas** que **soluciones** en su organización?

Realidad en las Organizaciones

- En una organización encontramos:
 - Aplicaciones cliente-servidor
 - ERPS
 - Aplicaciones WEB

Realidad en las Organizaciones

- Encontramos islas
 - Cada negocio cuenta con **un sistema** para solucionar una necesidad específica.
- La historia de TI
 - Proyectos **independientes**
 - **Falta de comunicación** entre áreas de negocio

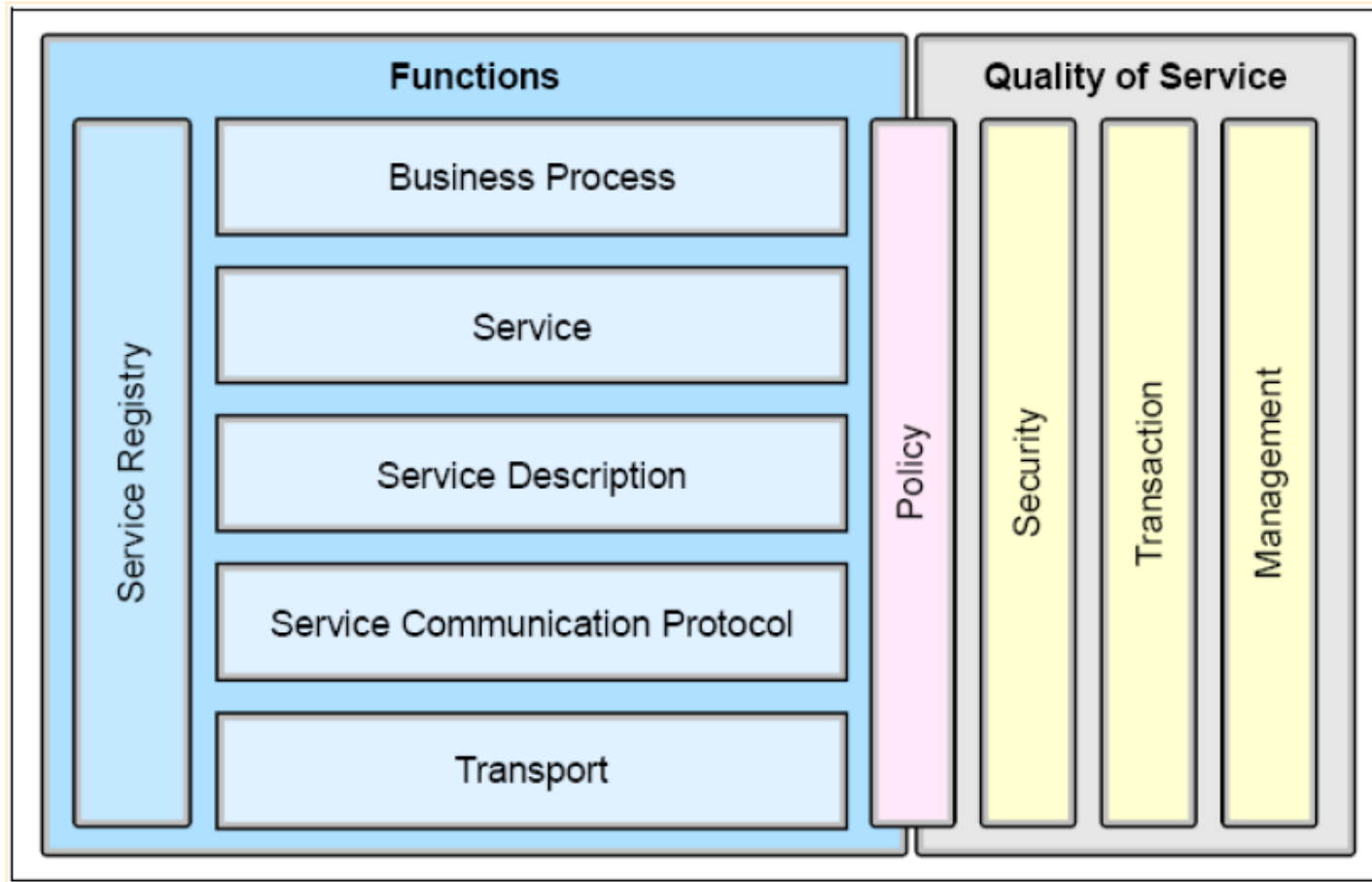
¿Qué es SOA?

- SOA significa: **Arquitectura Orientada a Servicios** (Service Oriented Architecture). Como su nombre indica el objetivo es proporcionar funcionalidad mediante servicios a las necesidades del negocio.
- SOA no es una tecnología, es una forma de hacer las cosas.
- SOA es un modelo de diseño basado en servicios.

Otras definiciones de SOA

- 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.

Componentes Principales de SOA



- **Roles, Entidades y Capacidades Evolucionan Agilmente**
 - *SO: Los Servicios deben ser Autónomos*
- **Exposición y Consumo de Servs realizado a propósito**
 - *SO: Las fronteras deben ser explícitas*
- **Las Capacidades se describen, negocian y acuerdan**
 - *SO: La Comunicación se basa en el Contrato*
- **Las Capacidades son Gobernadas por Demandas y Exigencias**
 - *SO: El Comportamiento de los servicios se basa en las Demandas y requerimientos (Políticas, etc.)*

SOA...

- Reduce el nivel de acoplamiento.
- Interoperabilidad.
- Mantenimiento.
- Reutilización.
- Escalabilidad.
- Permite un mapeo directo entre los procesos y los sistemas.

¿Qué es un Servicio?

- Un **servicio de negocio** es un componente reutilizable de software, con significado funcional completo, y que está compuesto por:
 - **Contrato:** especificación de la finalidad, funcionalidad, forma de uso y restricciones del servicio.
 - **Interfaz:** mecanismo de exposición del servicio a los usuarios.
 - **Implementación:** debe contener la lógica o el acceso a datos.



Tipos de Servicio

- **Servicios de proceso:** Representan procesos de negocio de larga duración, con posibles flujos complejos e interacción con usuarios.
- **Servicios de Actividad:** Coordina las operaciones de varios servicios de entidad en una sola operación atómica.
- **Servicios de Entidad:** Representan operaciones sencillas y atómicas. Pueden consultar, almacenar datos sobre una base de datos, componente, externo.

- Fue creado con el fin de permitir una programación rápida de sistemas distribuidos y el desarrollo de aplicaciones basadas en arquitecturas orientadas a servicios (también conocido como SOA), con una API simple.
- Integrado en el Framework de .NET desde la versión 3.0
- Permite enviar datos como mensajes asincrónicos de un extremo de servicio a otro.
- Un extremo puede formar parte de un servicio disponible continuamente hospedado por IIS, o puede ser un servicio hospedado en una aplicación.
- El otro extremo puede ser un cliente de un servicio que solicita datos de un extremo de servicio.





Orientado a Objetos



1980s

- Polimorfismo
- Encapsulamiento
- Sub Clases

Basado en Componentes



1990s

- Basado en Interfaces
- Carga Dinámica
- Runtime de MetaDatos

Orientado a Servicios



2000s

- Basado en mensajes
- Schema+Contract
- Binding via Policy

- **Mensajería y extremos**

- Se basa en la noción de comunicación basada en mensajes, y cualquier cosa que se pueda modelar como un mensaje.
- El modelo distingue entre clientes, que son aplicaciones que inician la comunicación y servicios, que son aplicaciones que esperan a que los clientes se comuniquen con ellos y respondan a esa comunicación.
- Los mensajes se envían entre extremos (Endpoint).
- Un extremo describe de una manera basada en estándar dónde se deberían enviar los mensajes, cómo se deberían enviar y qué aspecto deberían tener los mensajes.
- Un servicio puede exponer esta información como metadatos que los clientes pueden procesar para generar clientes WCF adecuados y pilas de comunicación.

- **Protocolos de comunicaciones**

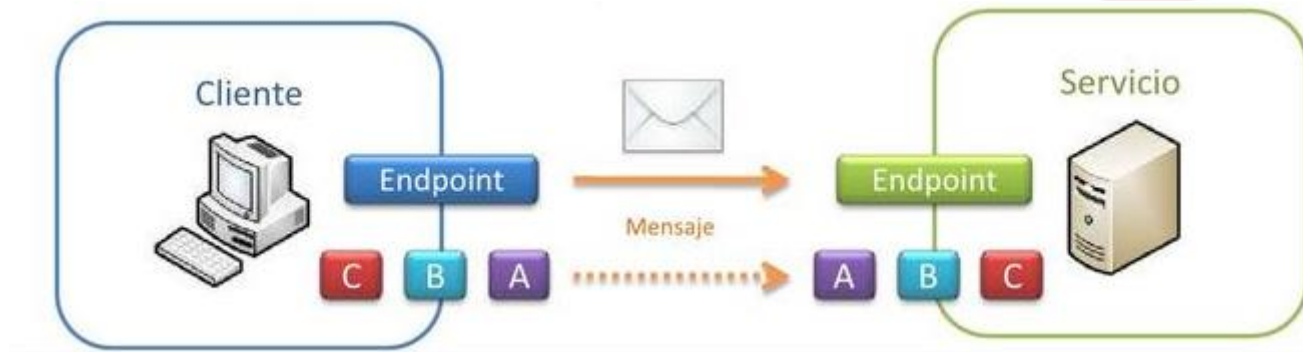
- Un elemento requerido de la pila de la comunicación es el protocolo de transporte.
 - Comunes: HTTP y TCP
 - Otros: Message Queue Server (MSMQ)
- Otro elemento necesario en la pila de comunicación es la codificación que especifica cómo se da formato a cualquier mensaje determinado. WCF proporciona las siguientes codificaciones:
 - Codificación de texto.
 - Codificación Mecanismo de optimización de transmisión de mensajes (MTOM).
 - Codificación binaria para una transferencia eficaz.

- **Patrones de mensajes**

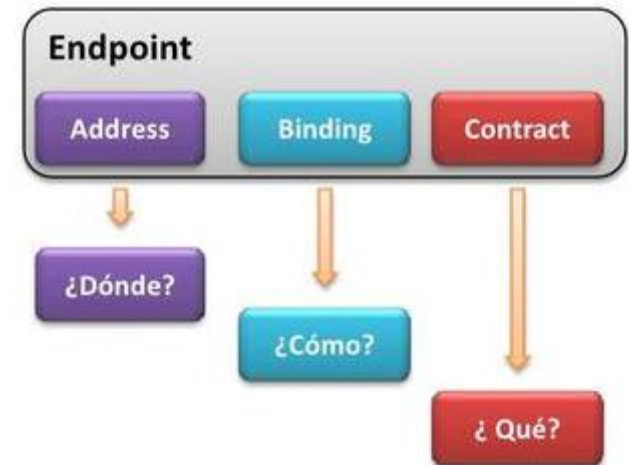
- WCF admite varios patrones de mensajería, incluida la comunicación de solicitud-respuesta unidireccional y dúplex.

- **ENDPOINT**

- Se define como un portal para la comunicación con todo el mundo.
- Es un recurso en la red, al cual pueden ser enviados los mensajes.
- Consiste en 3 componentes (el “ABC” de un endpoint):
 - Address (dirección).
 - Binding (Canal).
 - Contract (Contrato).



- **ADDRESS**
 - Dirección de la red, donde está alojado el “Endpoint” del servicio.
- **BINDING**
 - Define el canal y los parámetros de comunicación.
 - Consta de:
 - Protocolos: TCP, HTTP, MSMQ
 - Codificación: Texto o binario.
 - Seguridad: SSL, Seguridad SOAP
 - Referencia al servicio a exponer
- **CONTRACT**
 - Interfaz (Contrato del Servicio)
 - Define las capacidades ofrecidas por el “Endpoint”
 - Tipos:
 - Contrato de Servicios.
 - Contrato de Operación.
 - Contrato de Mensaje.
 - Contrato de Error.



- **CONTRATOS**

- Definen varios aspectos del sistema de mensajes y lo que puede hacer un servicio.
- Describe cada parámetro que constituye cada mensaje que un servicio puede crear o utilizar.
- Existen 3 tipos de contratos:
 - Contrato de datos
 - Contrato de mensaje
 - Contrato de servicio
- Las directivas y enlaces estipulan las condiciones exigidas para comunicarse con un servicio.



- **Tiempo de ejecución y servicios**
 - Permite ejecutar y cargar los servicios.
 - Instancia los nuevos servicios.
 - Contiene los comportamientos que sólo se producen durante la operación actual del servicio, es decir, los comportamientos en tiempo de ejecución del servicio.



- **Mensajería**
 - **Compuesta por canales.**
 - **Un canal es un componente que procesa un mensaje de alguna manera, por ejemplo, autenticando un mensaje.**
 - **Existen 2 tipos de canales:**
 - **Canales de transporte -> Leen y escriben mensajes de la red**
 - **Canales de protocolo -> Implementan protocolos de procesamiento de mensajes**
 - **La capa de la mensajería muestra los posibles formatos y modelos de intercambio de los datos.**



- **Activación y Alojamiento**
 - En su forma final, un servicio es un programa.
 - Como otros programas, un servicio se debe ejecutar en un ejecutable. Esto se conoce como un servicio con host propio.
 - También se pueden hospedar o ejecutar en un ejecutable administrado por un agente externo, como IIS .
 - Los servicios también se pueden ejecutar manualmente como ejecutables (archivos .exe)
 - También se puede ejecutar automáticamente como un servicio de Windows.

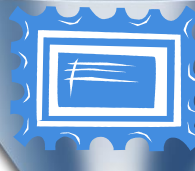


Datos



Modelo de Servicio

Address, Binding,
Contract y Behaviors

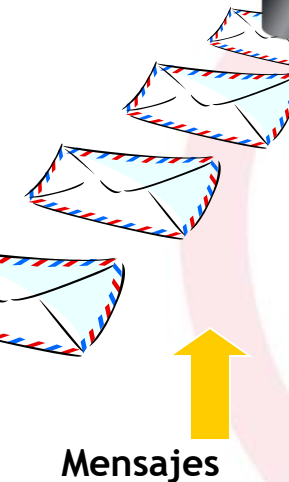
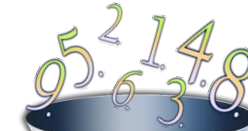
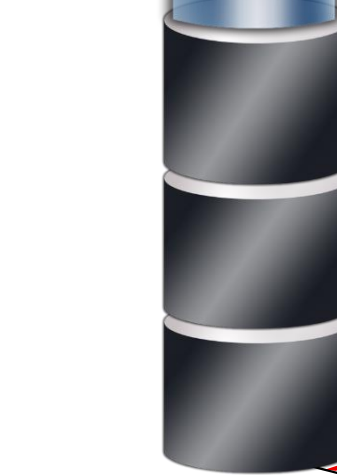


Metadatos



Channel Layer

Protocolos,
Encoders y
Transportes



Mensajes



- **Mostrar como se crean servicios WCF en Visual Studio 2017.**
- **Mostrar los contratos, Markup del Servicio y la implementación.**
- **Hacer un Servicio para insertar un Pedido por medio de SqlCommand en Northwind.**
- **Consumir el servicio creado desde la aplicación consola.**

- Realizar un servicio Web que devuelva los datos de un cliente con base en el Identificador y cree clientes en la base de datos Northwind. (Entity Framework)
- Debe ser consumido desde una aplicación de consola.

We are a global
business company.
We create solutions
through technology.

