



Tiziano Ferrandi

Plataforma tecnológica Wonderware
para aplicaciones de supervisión e
integración de planta

¿Qué nos solicitan nuestros clientes?

Entorno de
gestión
unificado
Plataforma

Reutilización de
componentes

Arquitecturas
flexibles

Gestión de
información de
instalaciones

Integración de
sistemas

Redundancia
nativa

¿Quién son nuestros clientes?



¿Quién son nuestros clientes?



¿Qué nos solicitan nuestros clientes?

Entorno de
gestión
unificado.
Plataforma

Reutilización de
componentes

Arquitecturas
flexibles

Gestión de
información de
instalaciones

Integración de
sistemas

Redundancia
nativa

Plataformas Software

Los principales fabricantes lanzan Plataformas Software con vocación de convertirse en entornos de desarrollo estándar.



Microsoft
.net



POWERED BY
SAP NetWeaver



ARCHITECTURE BY
ARCHESTRA
*Every system in your plant,
working in concert.*

¿Qué es ArchestrA?

ArchestrA es la Plataforma Software basada en .NET de Microsoft, desarrollada por Wonderware para facilitar e impulsar la Integración de Dispositivos y Sistemas a distintos niveles.



.Net y ArchestrA



Plataforma de Sistemas

**Gestión de
Aplicación
Única**

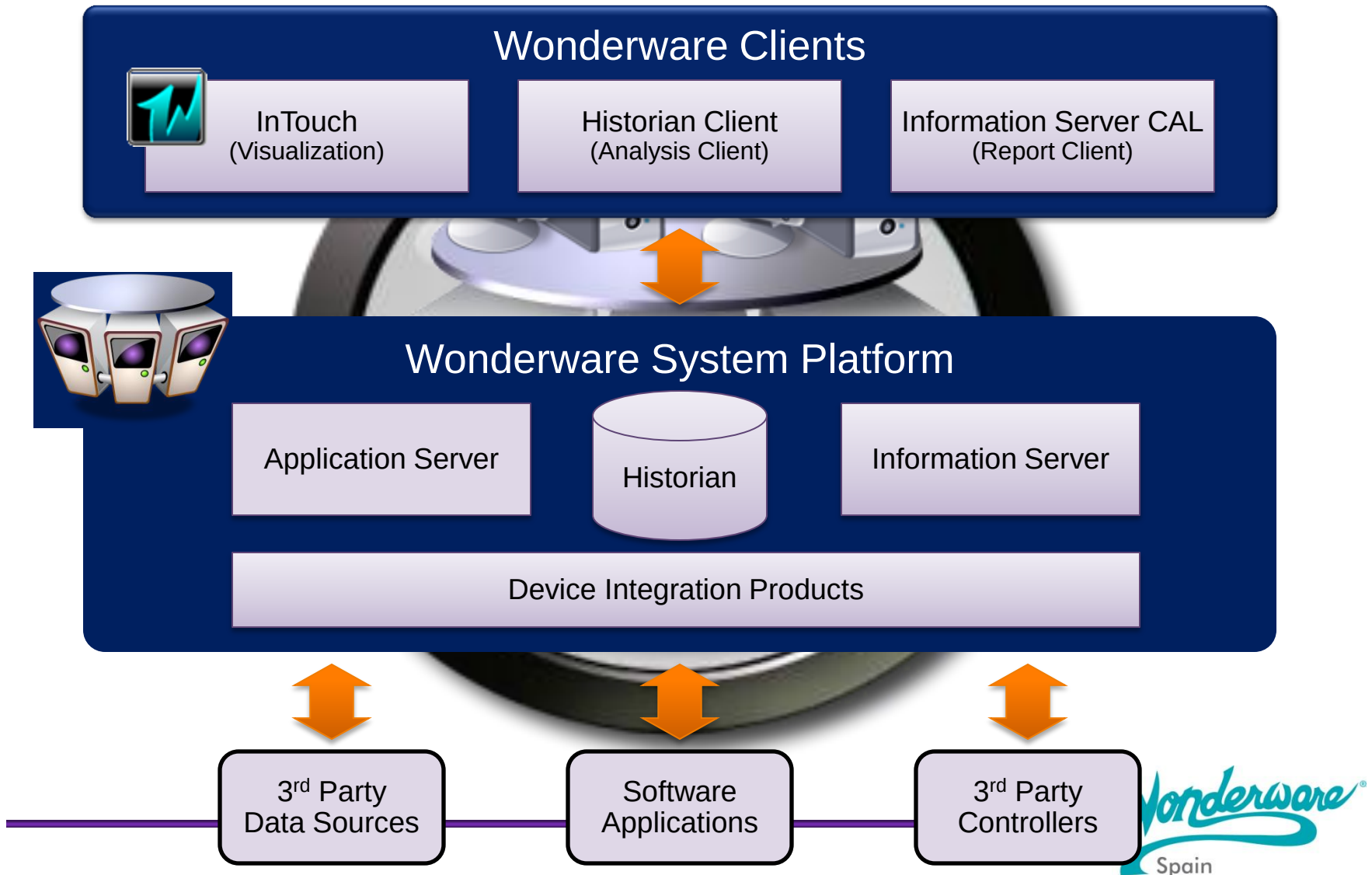
**Gestión
Entornos
Distribuidos**

**Integración de
Aplicaciones**

**Redundancia
Comunicaciones
Aplicaciones
Históricos
Visualización**

**Reducción TCO
(Total Cost of
Ownership)**

Introducción Wonderware System Platform



¿Qué nos solicitan nuestros clientes?

Entorno de
gestión
unificado.
Plataforma

Reutilización de
componentes

Arquitecturas
flexibles

Gestión de
información de
instalaciones

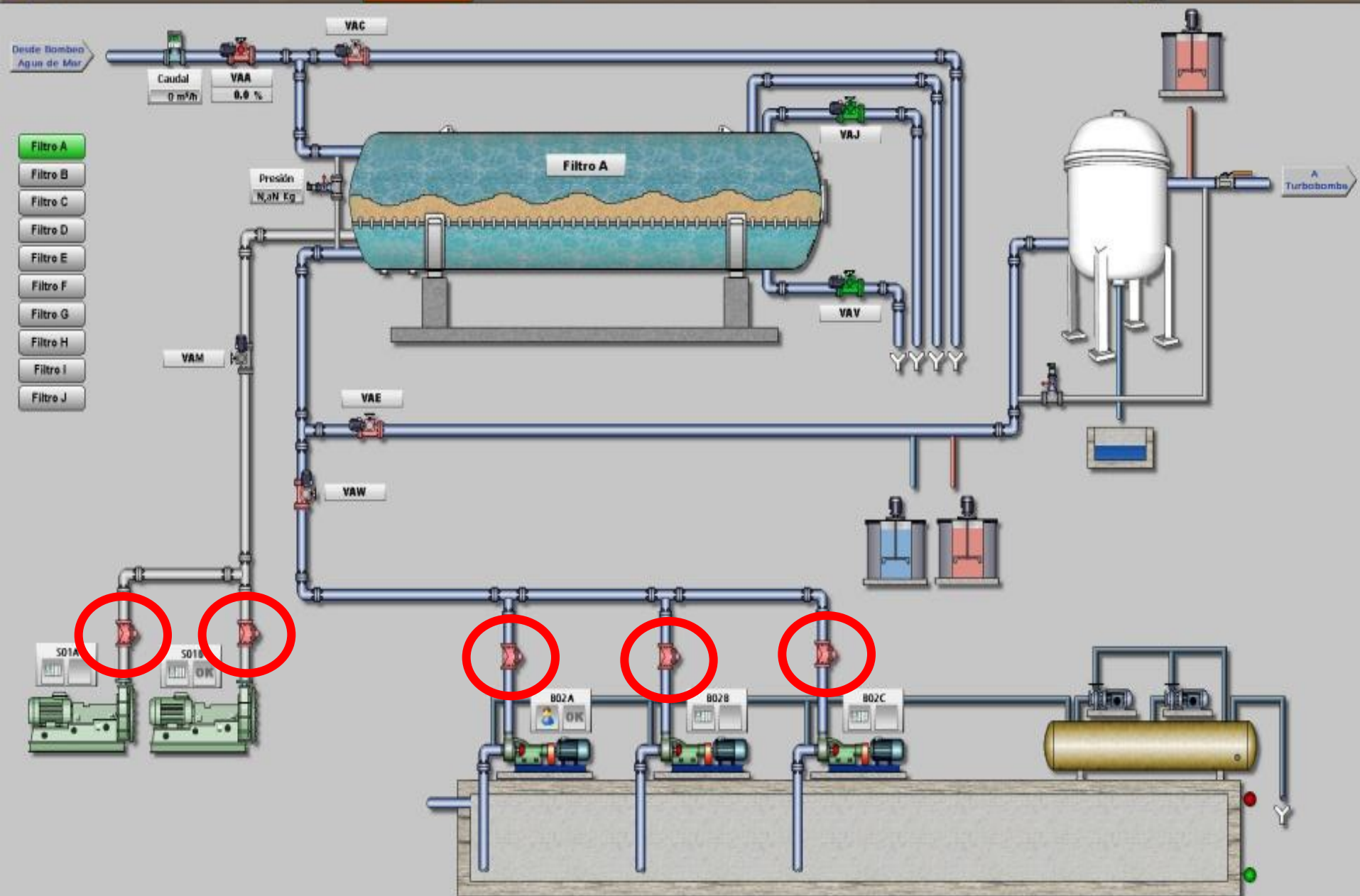
Integración de
sistemas

Redundancia
nativa

Programación orientada a objetos

■ Creación de objetos. Plantillas y Herencia





La definición de nuestro cliente...

Objetos

El entorno de desarrollo de basa en Objetos que pueden contener a su vez más Objetos...

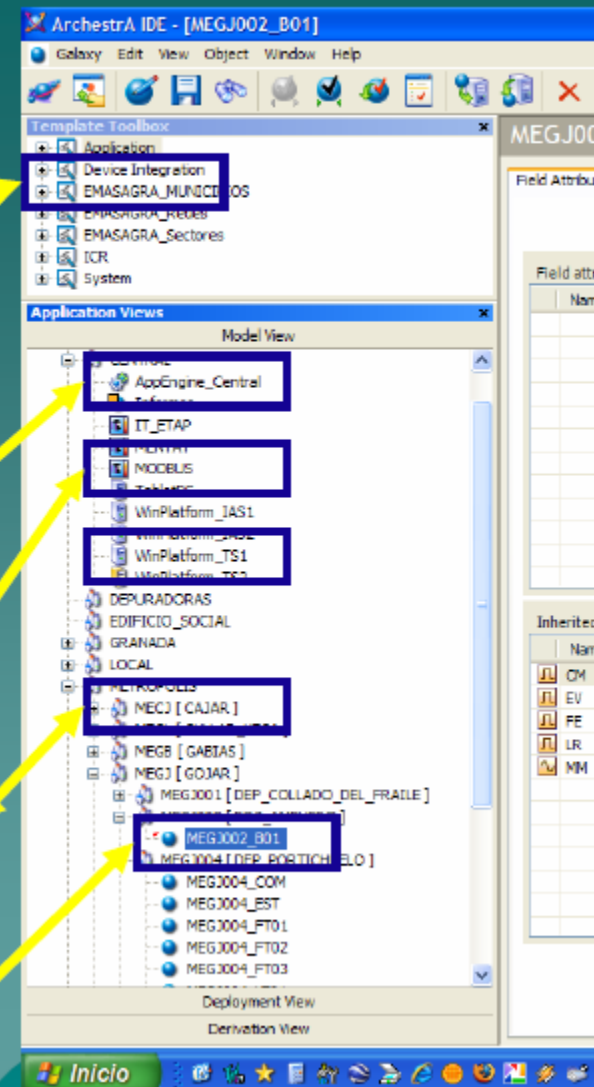
Templates: Plantillas para generar Objetos.

AppEngine: Uno o más por ordenador donde se ejecutan los objetos. Según tiempos de ciclo de ejecución de procesos...

Driver: Todos son Clientes Suite Link

WinPlatform: Define los ordenadores donde se van a ejecutar los AppEngine, el Servidor de Históricos, el Proveedor de Alarmas, etc...

Instancias: Materialización de una plantilla = Objeto.



¿Qué nos solicitan nuestros clientes?

Entorno de
gestión
unificado.
Plataforma

Reutilización de
componentes

Arquitecturas
flexibles

Gestión de
información de
instalaciones

Integración de
sistemas

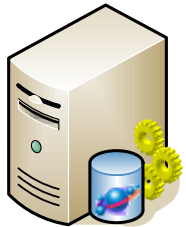
Redundancia
nativa

Escalabilidad, Arquitecturas Flexibles

Small System



Large System



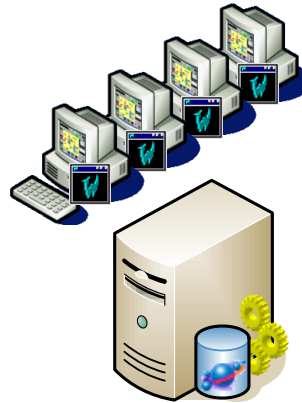
Application Object
Server Only



Integrated
Single Node



Distributed
Peer to Peer



Single Server
Multi Client



Distributed Multi Server
Multi Client

Escalabilidad, Arquitecturas Flexibles

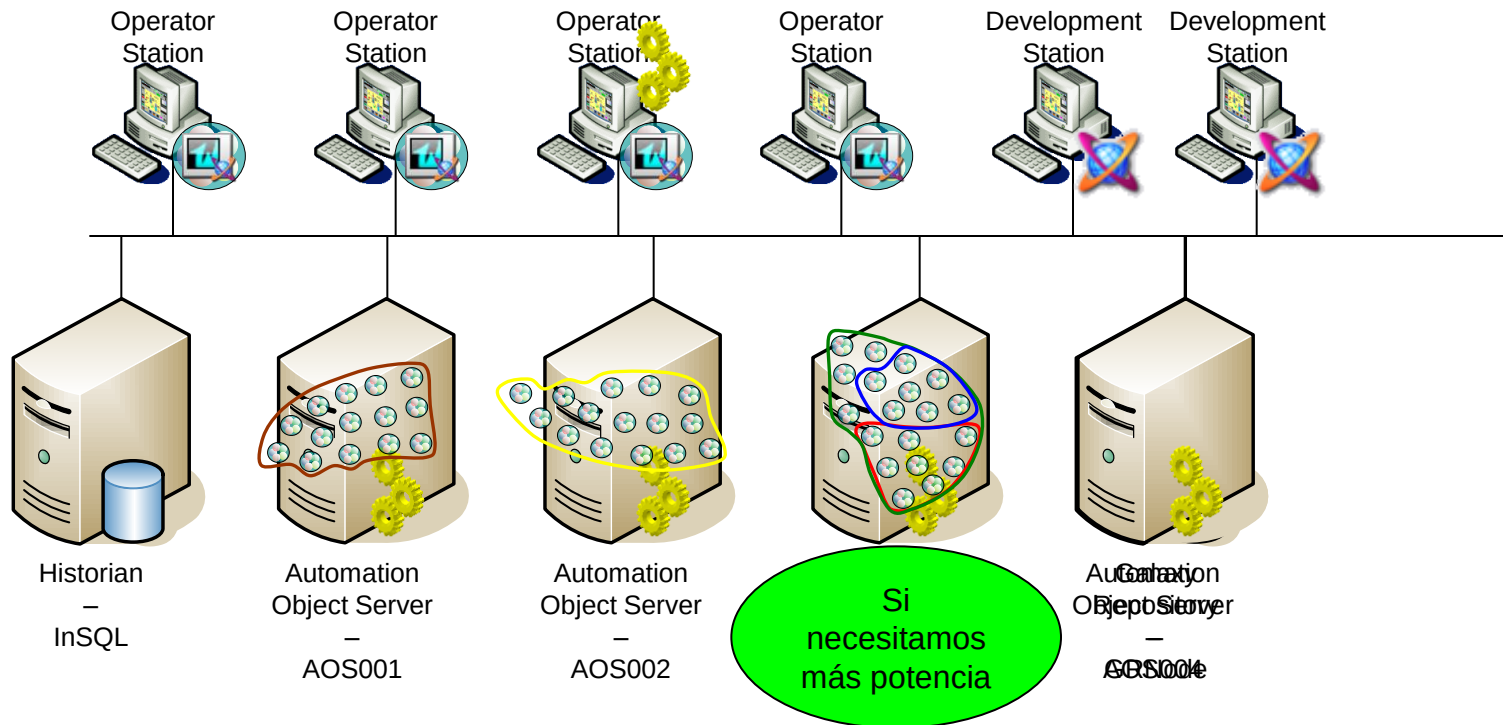
Señales e Históricos

Servidores y Clientes

Incremento de Funcionalidades

Expansión del Sistema.

Redistribución de la Carga de trabajo



Redistribución de la Carga en Tiempo de Ejecución. Alta Disponibilidad.

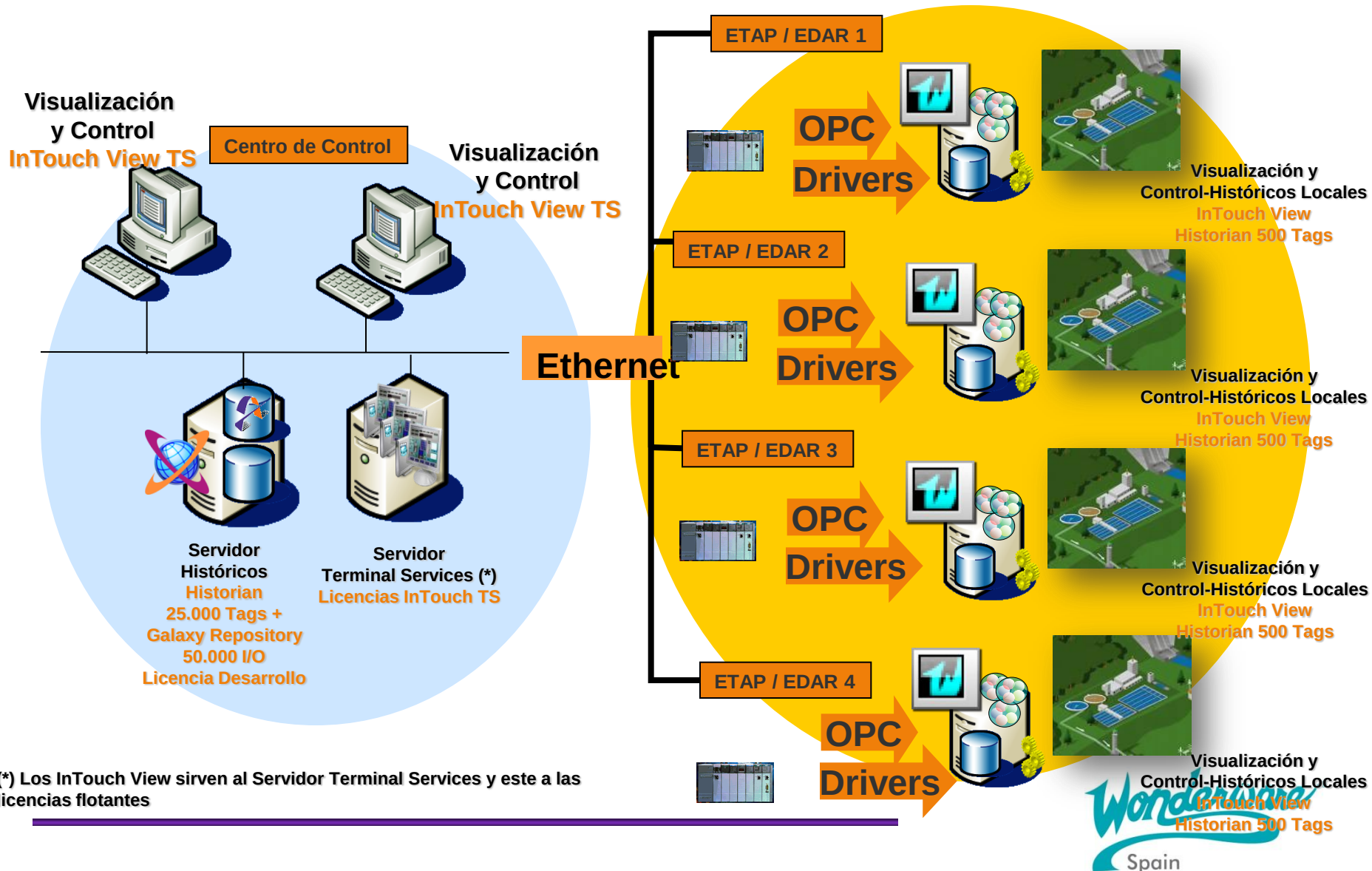
Ampliación de Servidores para dotar de más Potencia al Sistema. Ampliación. Escalabilidad

Ampliación de Puestos Clientes sin necesidad de “tocar” la Aplicación Principal. Reducción TCO.

Los Clientes leen los objetos distribuidos entre Servidores. No es necesario definir dirección. Flexibilidad.

Arquitectura System Platform

Acceso Terminal Services, Históricos Centrales y Locales



¿Qué nos solicitan nuestros clientes?

Entorno de
gestión
unificado.
Plataforma

Reutilización de
componentes

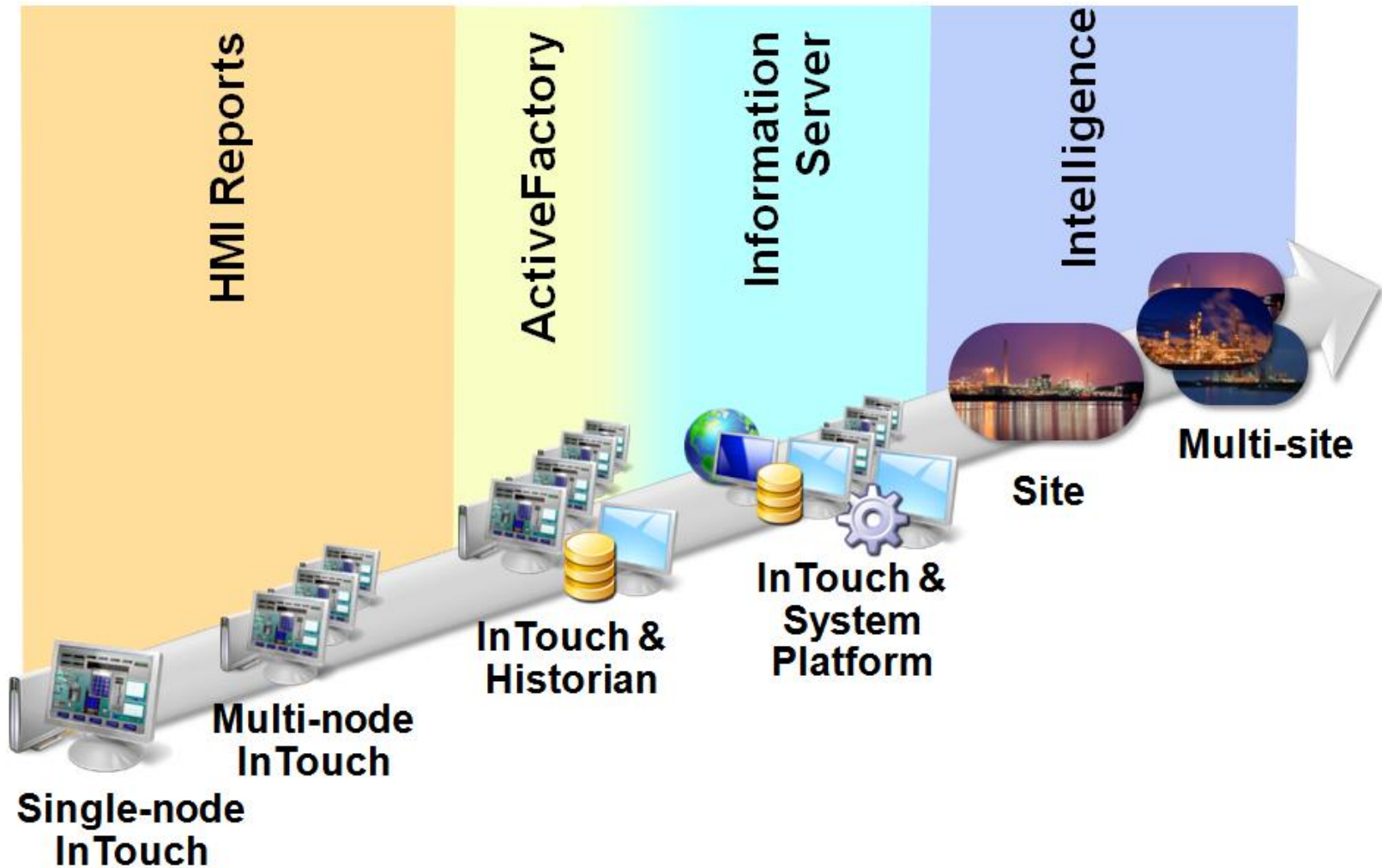
Arquitecturas
flexibles

Gestión de
información de
instalaciones

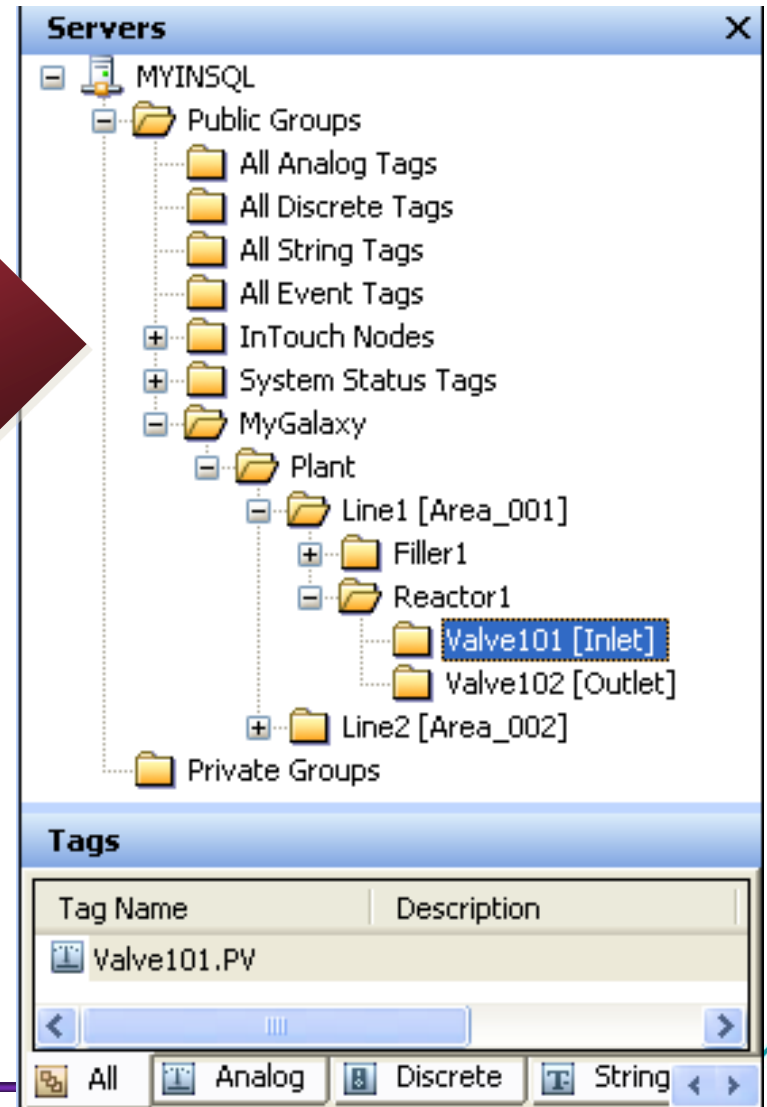
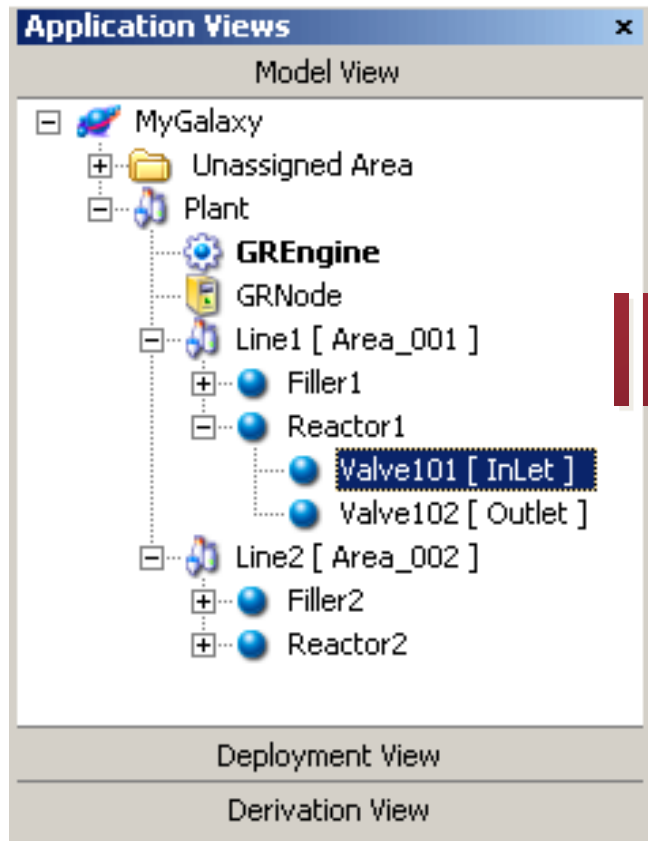
Integración de
sistemas

Redundancia
nativa

Opciones de acceso a la información

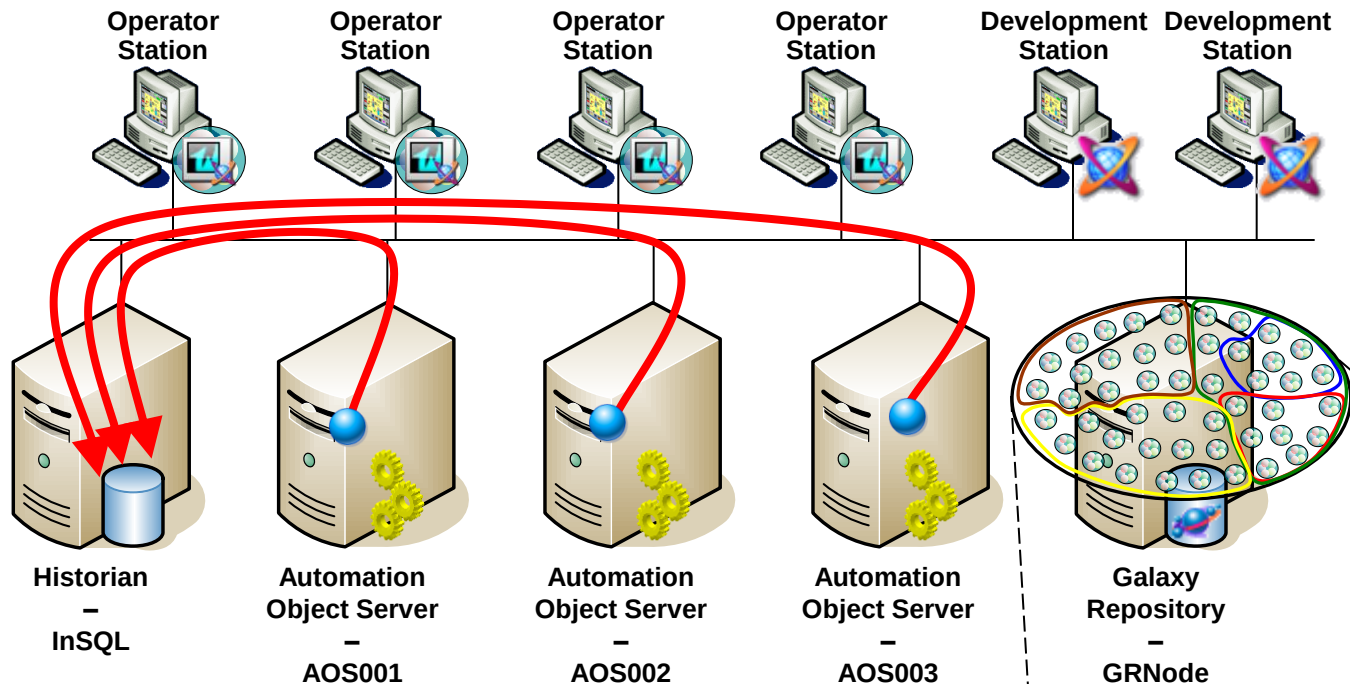


El Modelo creado en la Plataforma configura la Estructura de la Base de Datos. MODELO DE INFORMACION



- Creado Automáticamente
- Despliegue de Objetos

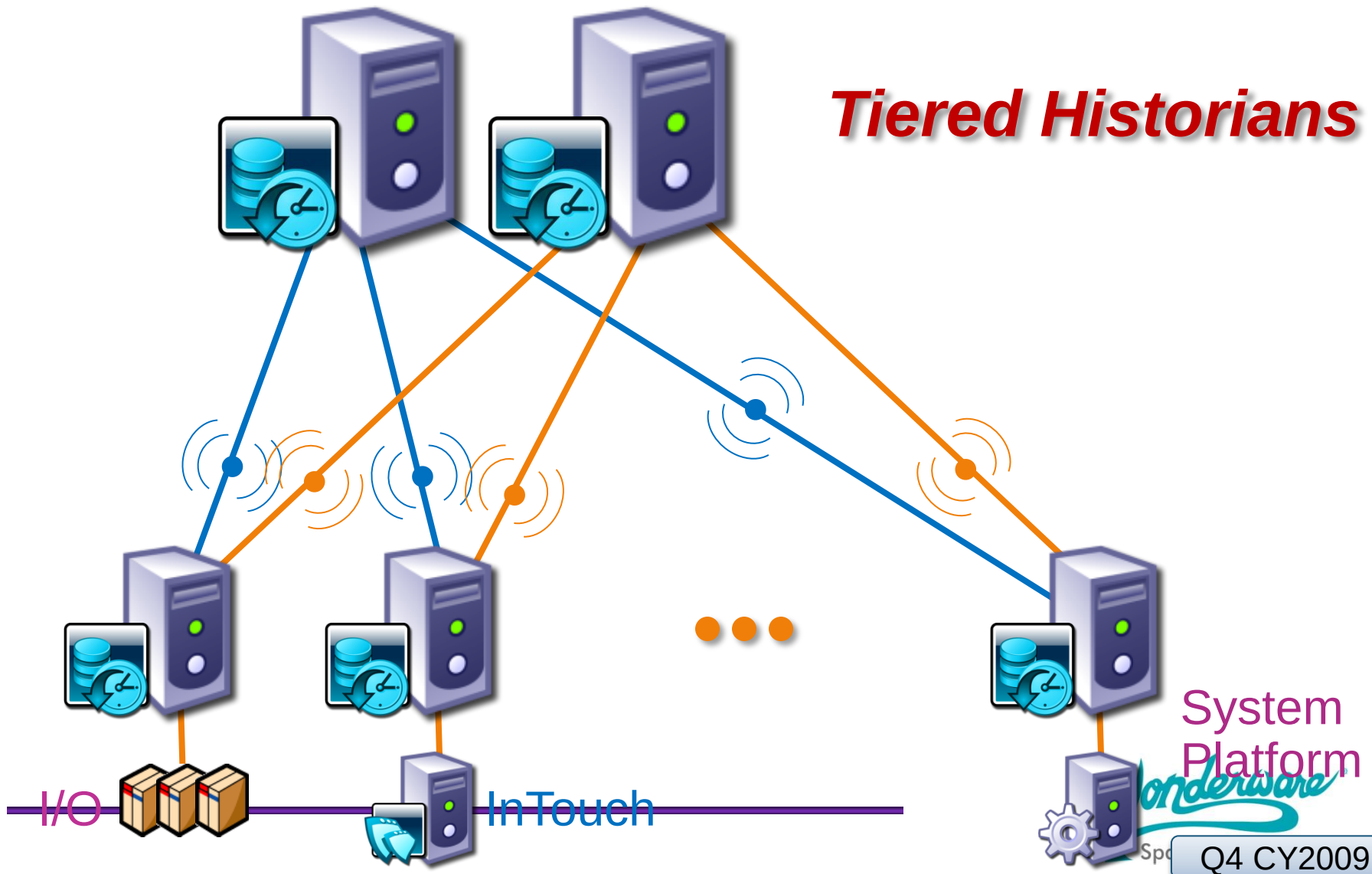
Configuración del Historian



☒ History extension

Force storage period:	0	ms		
Engineering units:	m/s			
Value deadband:	0.1	EU		
Trend high:	10.0	EU		
Trend low:	0.0	EU		

Alta disponibilidad de datos Históricos



¿Qué nos solicitan nuestros clientes?

Entorno de
gestión
unificado.
Plataforma

Reutilización de
componentes

Arquitecturas
flexibles

Gestión de
información de
instalaciones

Integración de
sistemas

Redundancia
nativa

Disponibilidad a nivel de Aplicación

Configuración :

The screenshot displays the Archestra IDE interface for configuring an application engine. The window title is "Archestra IDE - [AppEngine_001]". The menu bar includes "Galaxy", "Edit", "View", "Object", "Window", and "Help". The toolbar contains various icons for file operations and system management.

The left pane, titled "Application Views", shows a tree structure under "Deployment View". The tree includes "WaterDemoRedundancy", "Unassigned Host", "NodeA", "AppEngine_001" (selected), "CommonEquip [CommonEquip]", "Filters [Filters]", "SystemArea", "Tank100 [Tank100]", "Tank200 [Tank200]", "Tank300 [Tank300]", "Tank400 [Tank400]", "WaterPlant", "NodeB", and "AppEngine_001 (Backup)".

The right pane, titled "AppEngine_001", shows the configuration for the selected engine. It includes a "Keep Checked Out" checkbox and a tabbed interface with "General", "Redundancy", "Alarms", "Scheduler History", "Engine History", and "Object Information". The "Redundancy" tab is active, showing the following settings:

- ☒ Enable redundancy
- Forced failover timeout: 30000
- Maximum checkpoint deltas buffered: 0
- Maximum alarm state changes buffered: 0
- Standby engine heartbeat period: 1000
- Active engine heartbeat period: 1000
- Maximum consecutive heartbeats missed from Active engine: 5
- Maximum consecutive heartbeats missed from Standby engine: 5

The status bar at the bottom shows "Ready", "DefaultUser", and "WaterDemoRedundancy on VM" with a green checkmark icon.

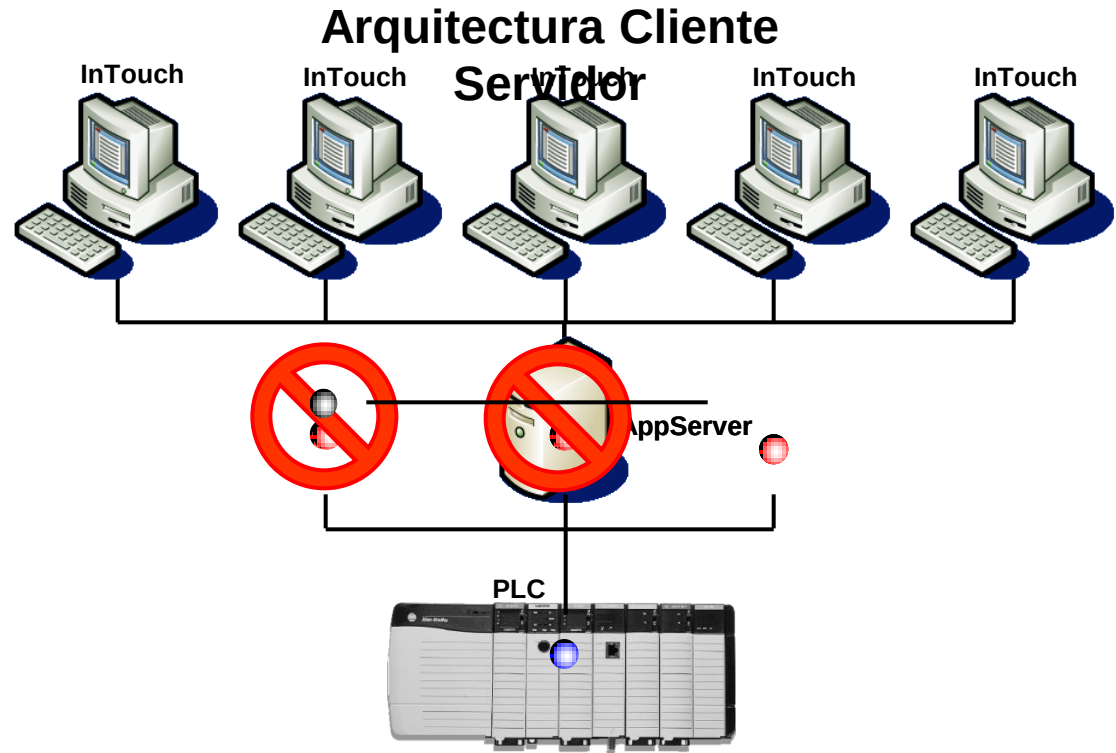
Disponibilidad a nivel de Aplicación

Sistema no redundante

1. El servidor maneja todos los datos.
2. Si cae el servidor los clientes pierden la conexión con los datos.
 - InTouch
 - Historian
 - Otros AppServers

Sistema redundante

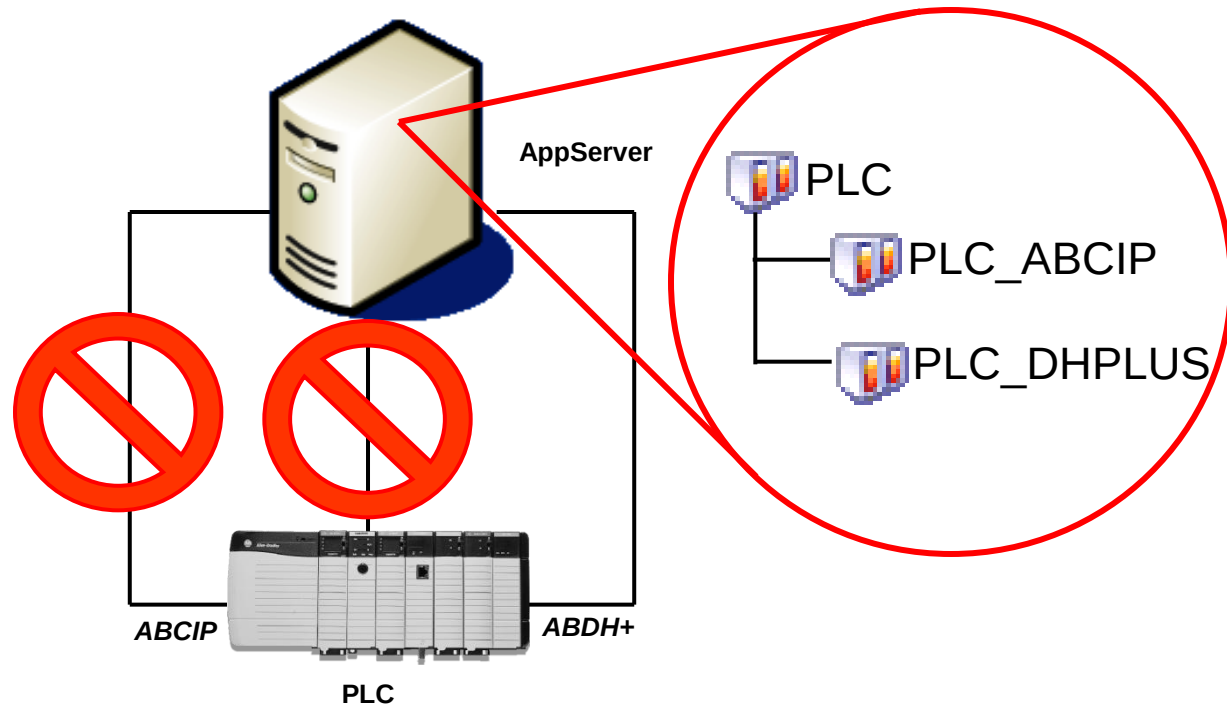
1. Los servidores redundantes mantienen sincronizados los datos a través de una red dedicada.
2. Si el servidor primario se cae el servidor de repuesto se encargará de gestionar lo que el primario no hizo. La conmutación es transparente para los equipos clientes.
3. La configuración se hace seleccionando una check box en el AppEngine.



Vamos a ver como se realiza esta sincronización.

Redundancia Nivel Comunicación

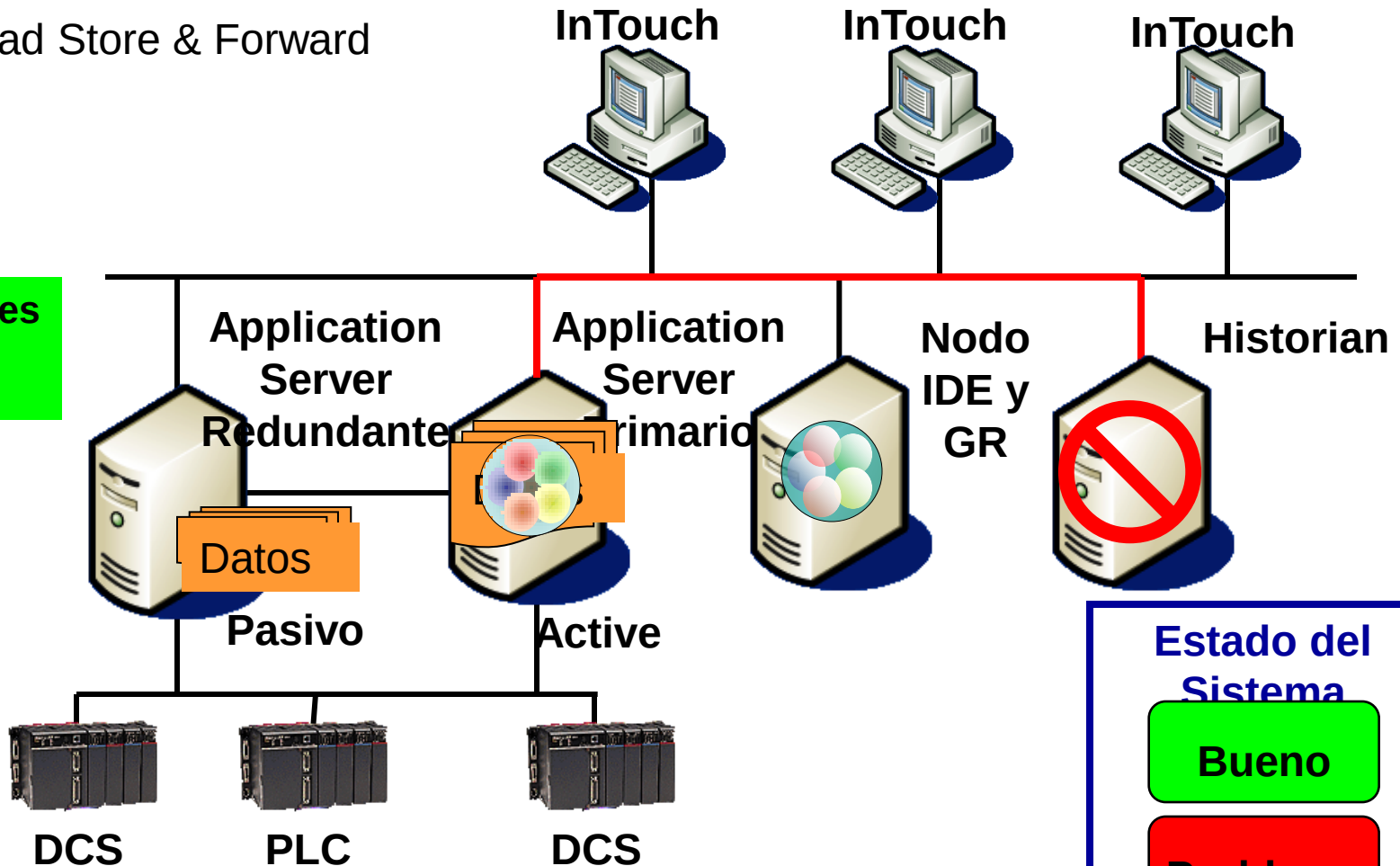
1. Las comunicaciones con los dispositivos de campo también necesitan ser tolerantes a fallos.
2. La pérdida de la red de campo puede causar la pérdida de conexión con los datos.
3. El DI Object redundante permite comunicaciones a través de otra red si la primera se cae.
4. El DI Object redundante permite a los objetos clientes suscribirse a una sola dirección que puede coger datos de las dos redes.



Alta disponibilidad Históricos. Store & Forward

- Funcionalidad Store & Forward

Comunicaciones
resistentes a
fallos



Estado del
Sistema

Bueno

Problema

MES & EMI SOFTWARE SOLUTION

**SYSTEM
INTEROPERABILITY**

MANUFACTURING OPERATIONS MANAGEMENT

**INFORMATION /
VISIBILITY**

ERP
(SAP, Oracle,
Microsoft..)



Database
Applications



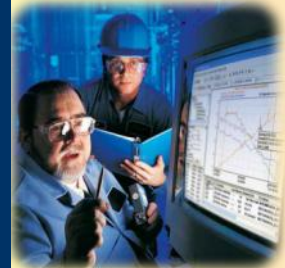
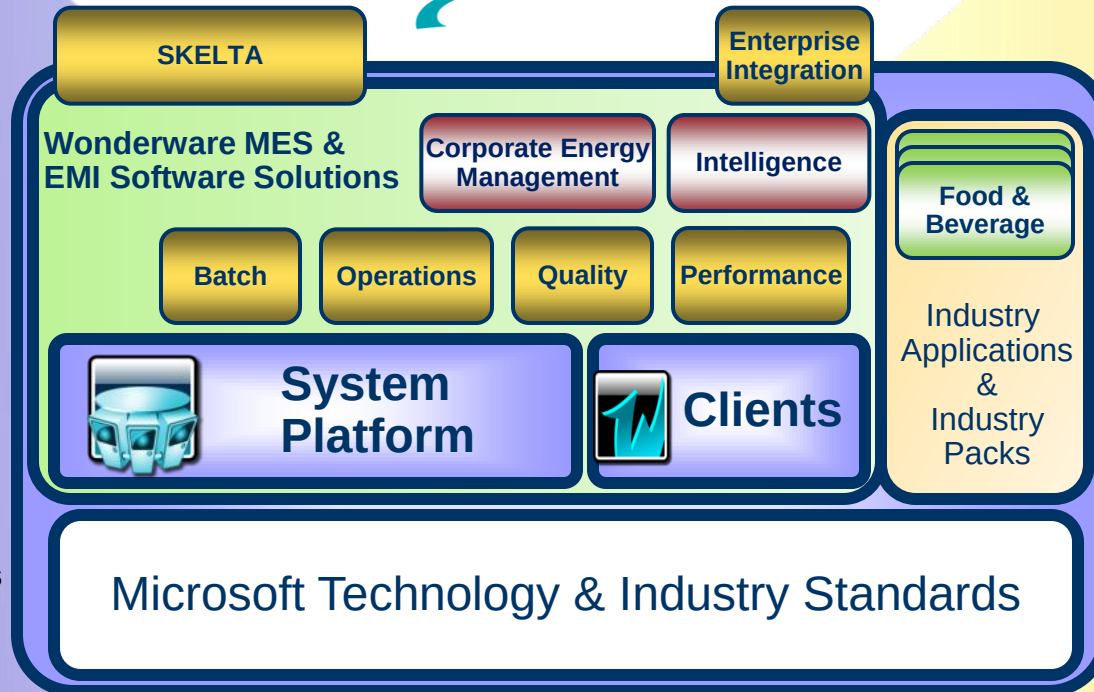
Asset
Management



LIM Systems



Devices



Empowering Success of Production Workers



Customer First



Customer FIRST



¿Preguntas?



tiziano.ferrandi@wonderware.es