



# OPC

## Ole for Process Control

Controllo Digitale

Alessia Ciuffo

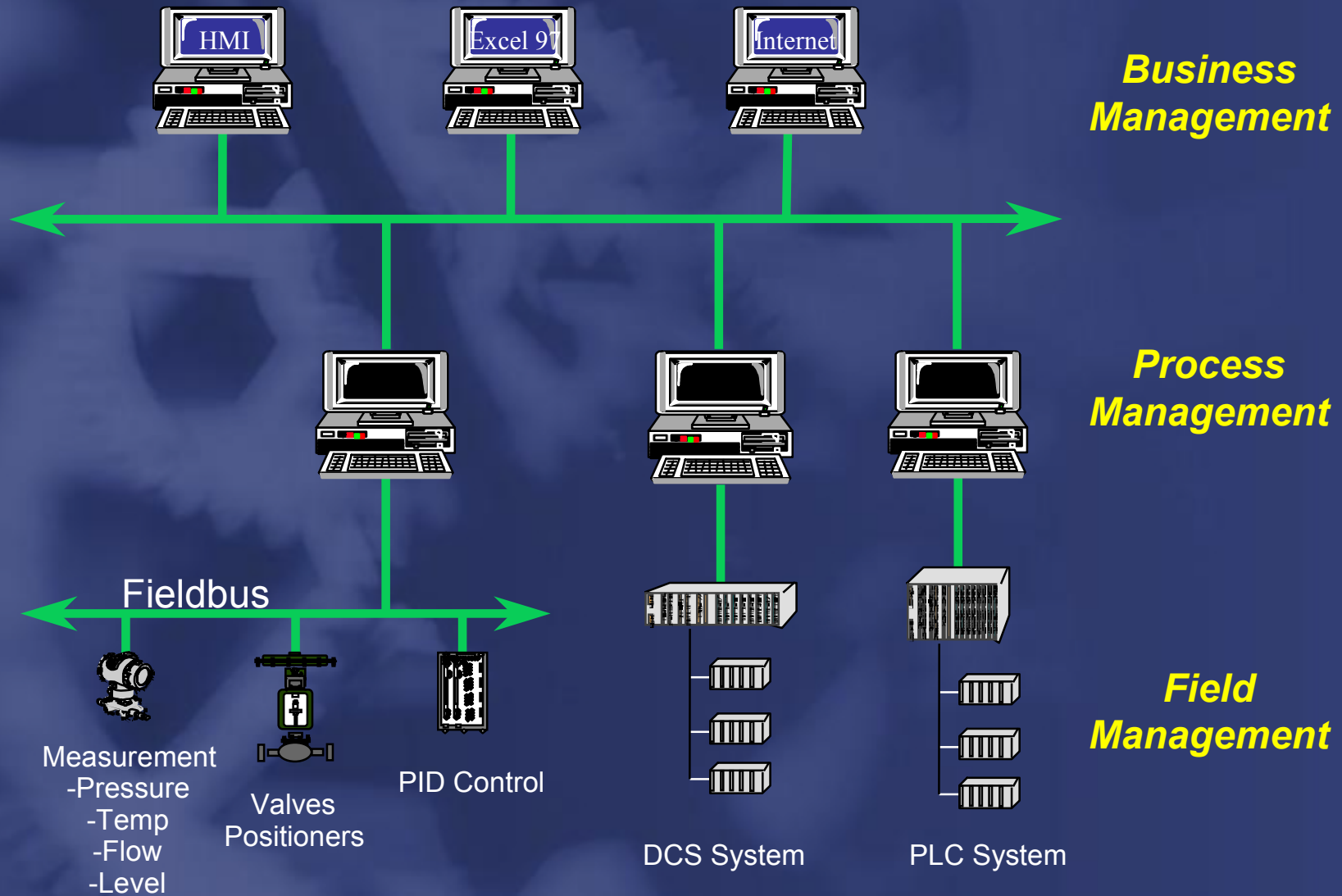
4/10/2000

# Aspetti trattati

- ◆ OLE Background
- ◆ Soluzione: OLE per il Controllo di Processi
- ◆ OPC: Aspetti tecnici
- ◆ OPC: Vantaggi

# OLE Background

Architettura del processo di controllo



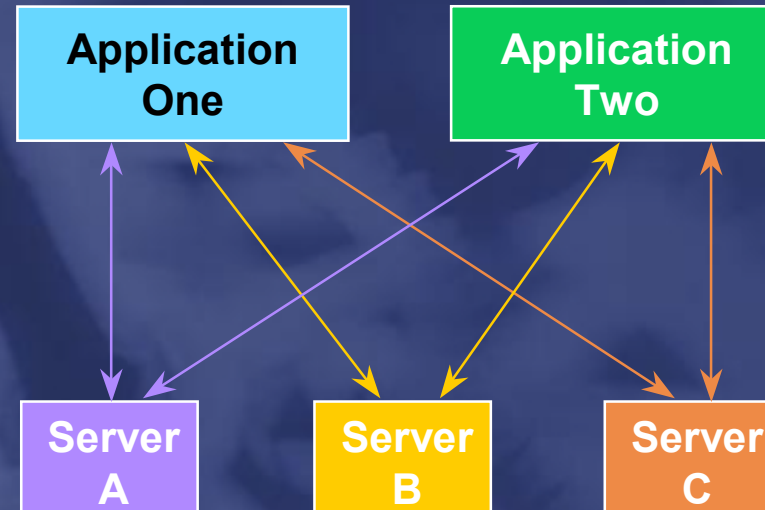
# OLE Background

**Field Management:** L'uso di dispositivi a campo intelligente, fornisce dati quali la salute del dispositivo stesso, i suoi parametri di configurazione, i materiali di costruzione ecc. Tutte queste info devono essere presentate allo user e ad ogni applicazione che questo utilizza, in modo consistente.

**Process Management:** I processi da monitorare e da controllare producono dati disponibili elettronicamente.

**Business Management:** Per poter integrare le info collezionate dal processo in un sistema di gestione è necessario l'accesso ai dati del livello più basso e la successiva integrazione con i sistemi di gestione già esistenti.

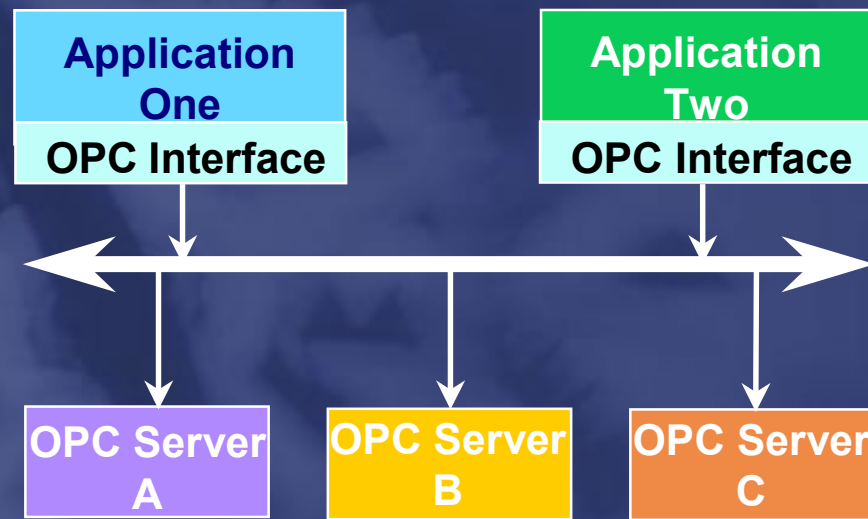
# Problemi riscontrati



- ◆ Incompatibilità delle interfacce di comunicazione
- ◆ Ogni applicazione sviluppa interfacce private
- ◆ Dispositivi venduti necessitano di un server per ogni applicazione
- ◆ Risultato: esistono centinaia di driver per permettere ai controllori di processo di comunicare con i dispositivi periferici.
- ◆ Problemi: inconsistenza tra i driver venduti da ditte diverse, conflitti e necessità di modifiche in caso di sostituzione di componenti Hardware.

# Soluzione:

## OLE per il Controllo di Processo



OPC definisce un'interfaccia aperta tramite la quale componenti software per PC sono in grado di scambiare dati.

SCOPO: consentire l'interoperabilità tra le applicazioni .

“Si può affermare che il Plug&Play è diventato realtà nell'automazione”.

# SOLUZIONE

## OLE per il Controllo di Processo

“Raramente vi è stato un così forte cambiamento nel mondo dell’automazione, come quello provocato dalle specifiche sulla comunicazione”.

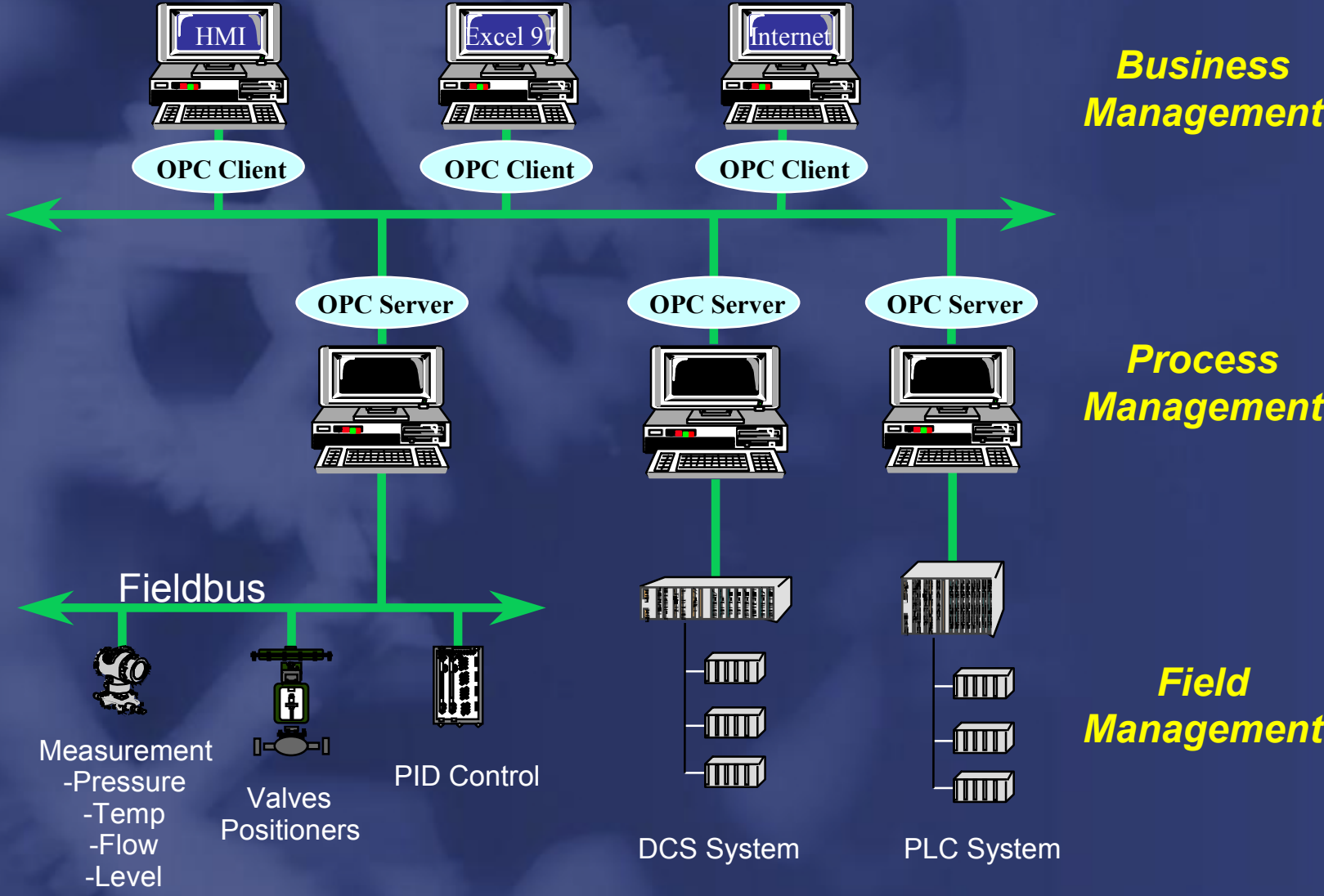
OPC è diventato oggi uno standard necessario a qualsiasi costruttore di componenti per l’automazione.

Basato sulla tecnologia Windows **OLE** ( Object Linking and Embedding), **COM** ( Component Object Model) e **DCOM** ( Distributed COM). Queste tecnologie definiscono oggetti standard, metodi e proprietà per server real-time come sistemi di processi distribuiti, controllori a logica programmabile.

La prima stesura di OPC mette a fuoco tre punti:

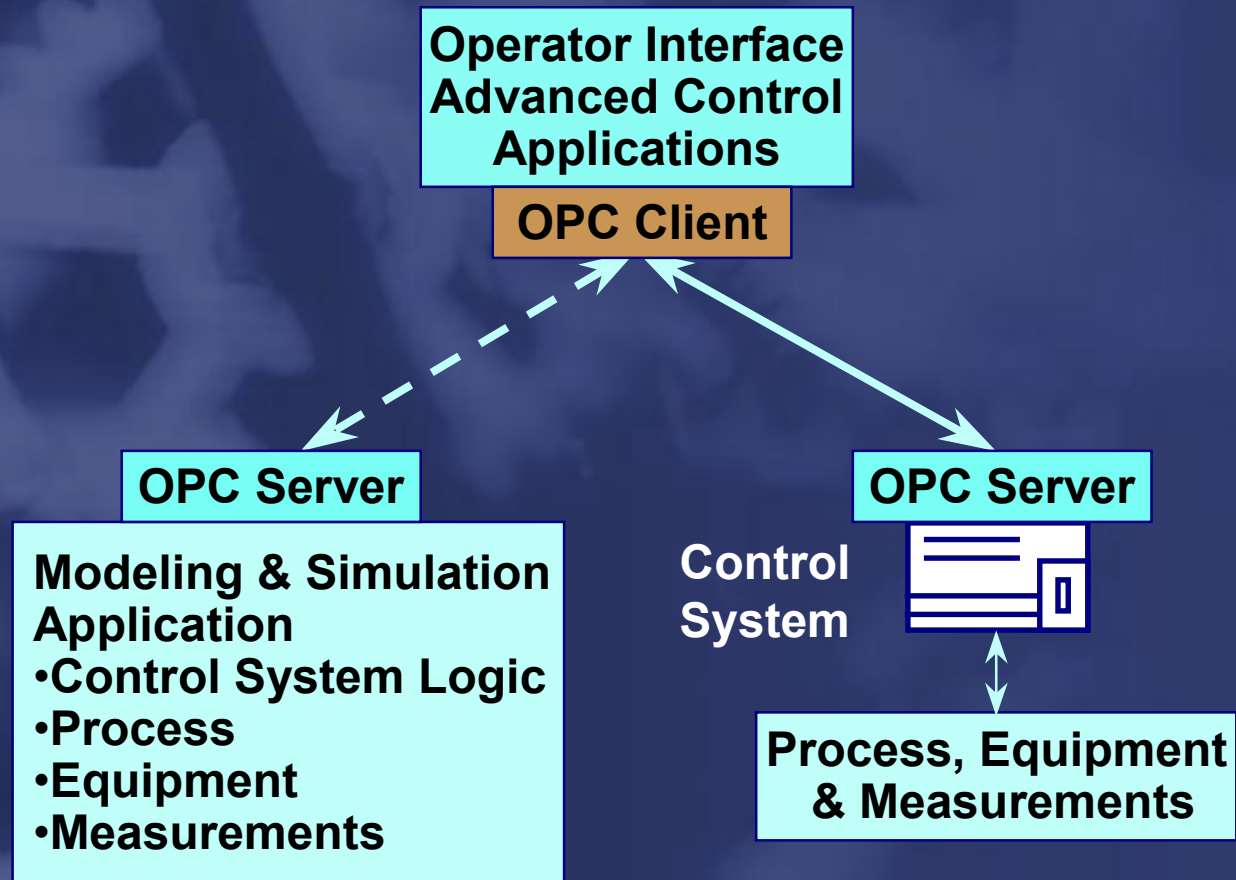
- ◆ Accesso dati on-line (più flessibile e efficiente)
- ◆ Allarmi ed Eventi : l’ Interfaccia di un Client OPC fornisce un meccanismo di avviso circa il verificarsi di specifici eventi o condizioni di allarme.

# Applicazione dell' interfaccia OPC



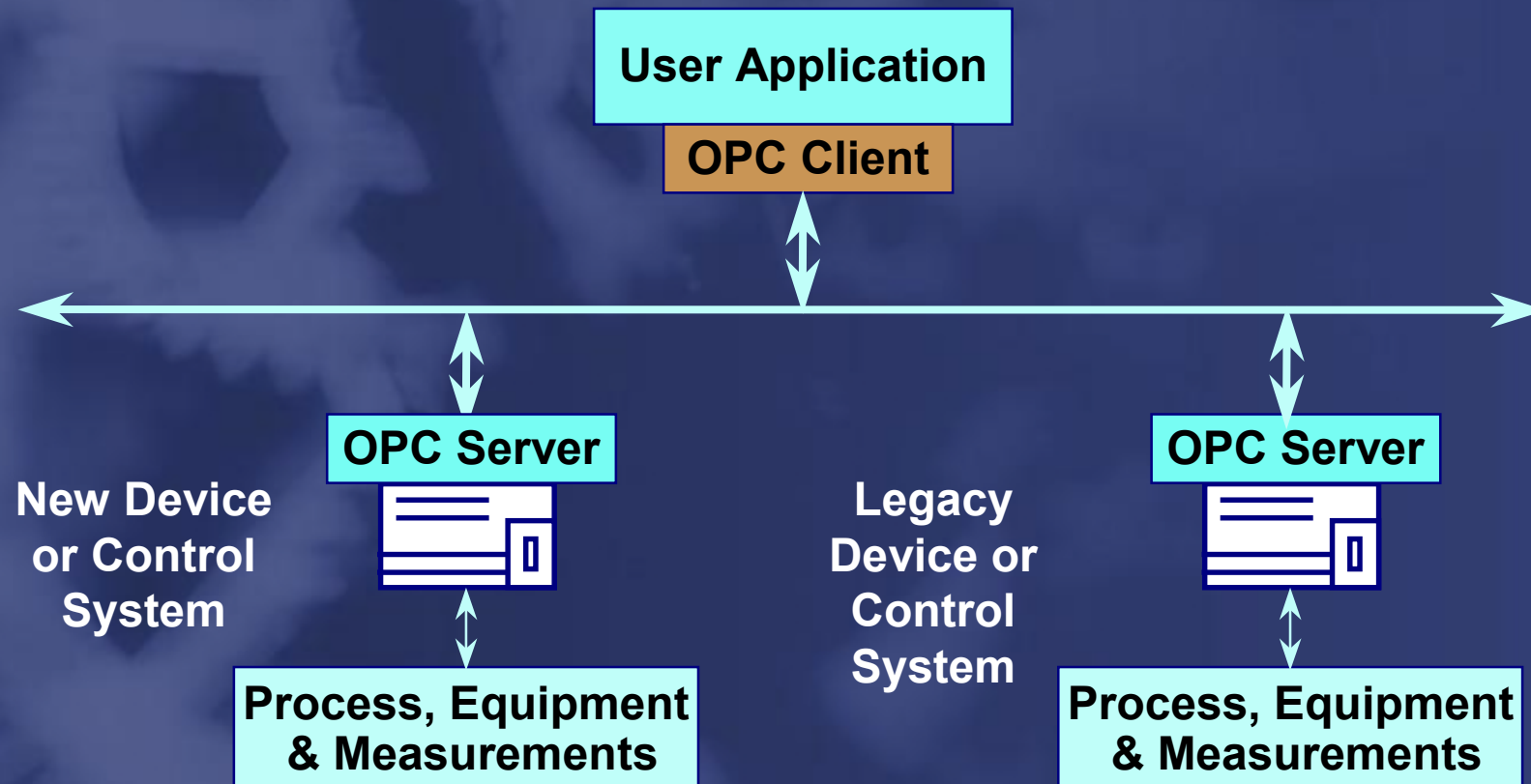


# OPC per Simulazione e Control Avanzato



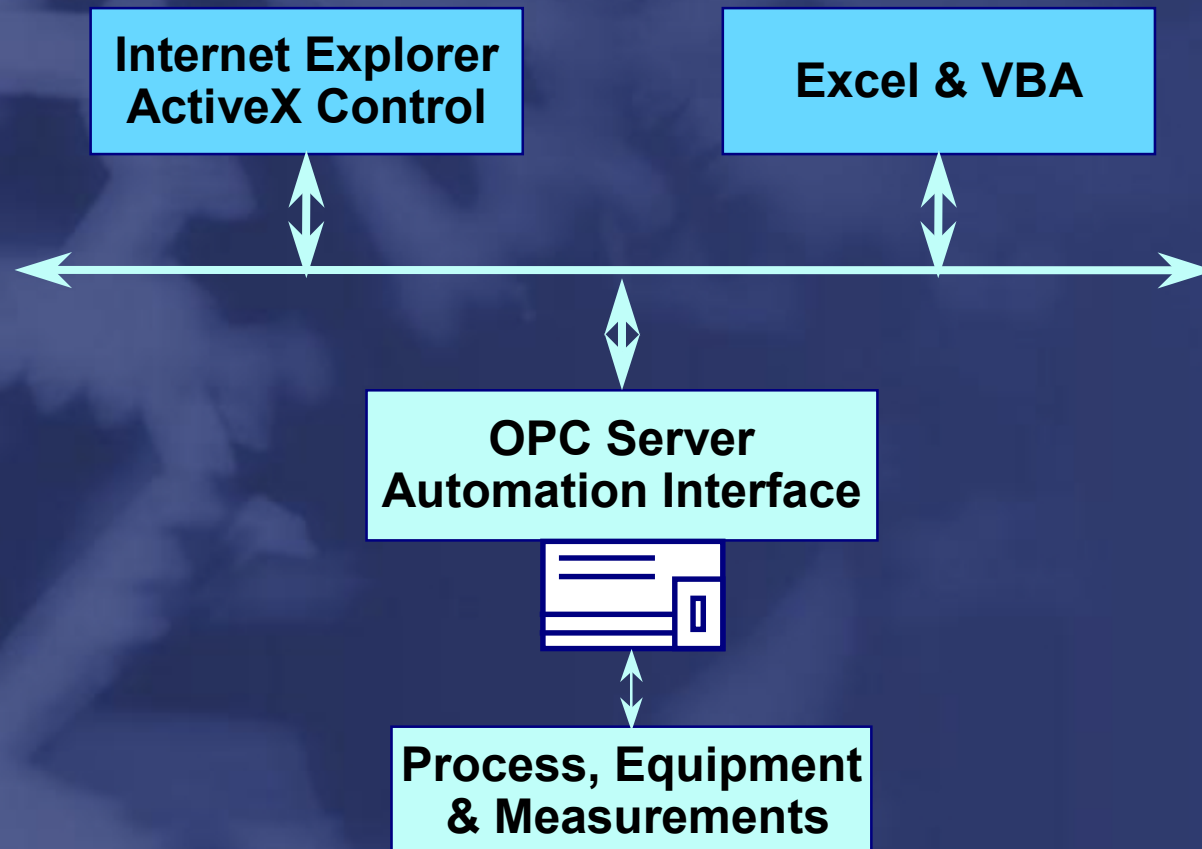
# OPC Connessione di Applicazioni

- ◆ User può usare OPC per spartire informazioni tra vecchi e nuovi sistemi

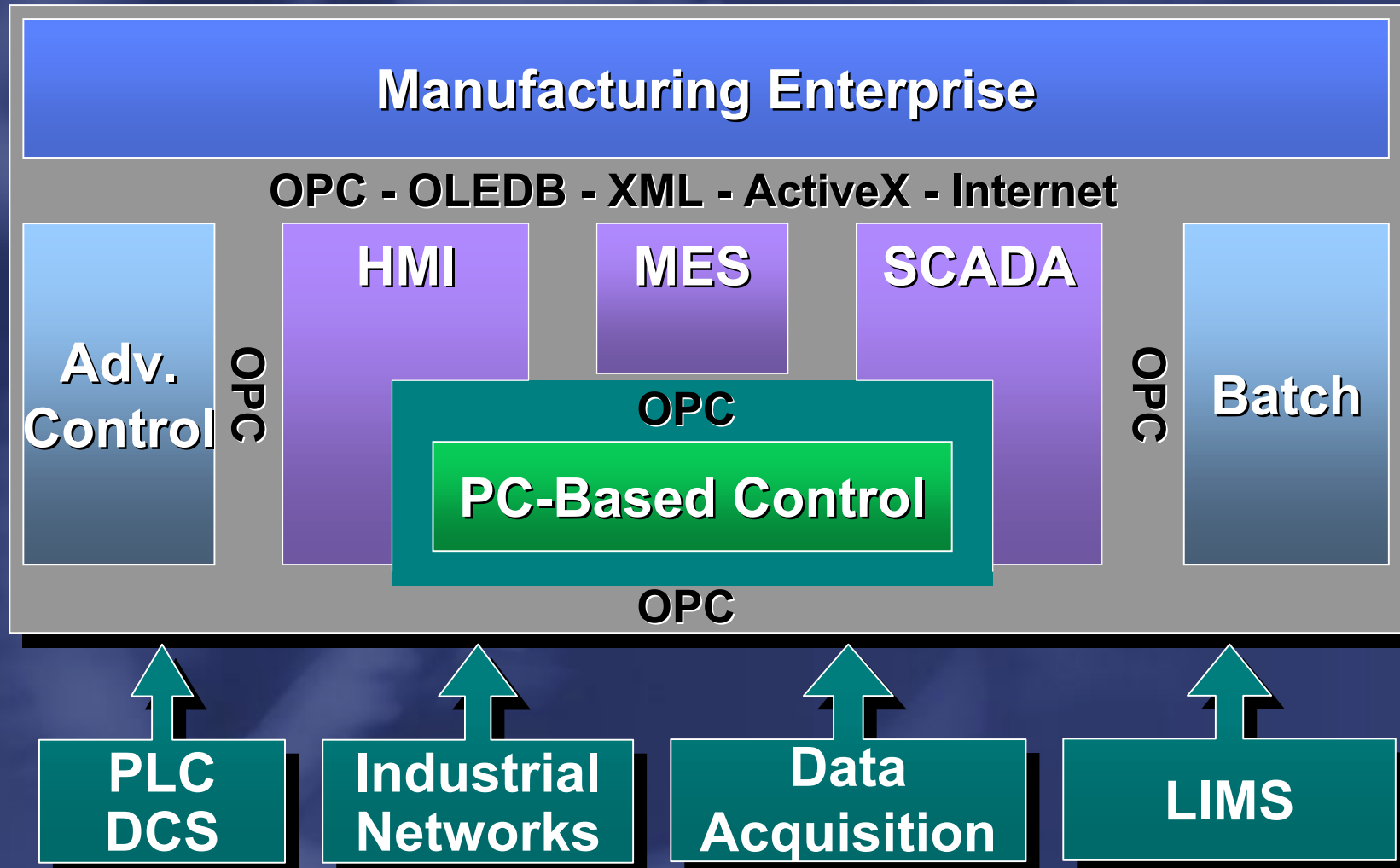


# OPC, ActiveX e Internet

- ◆ User può accedere ai dati OPC da applicazioni VB e da Internet



# OPC permette Interoperabilità

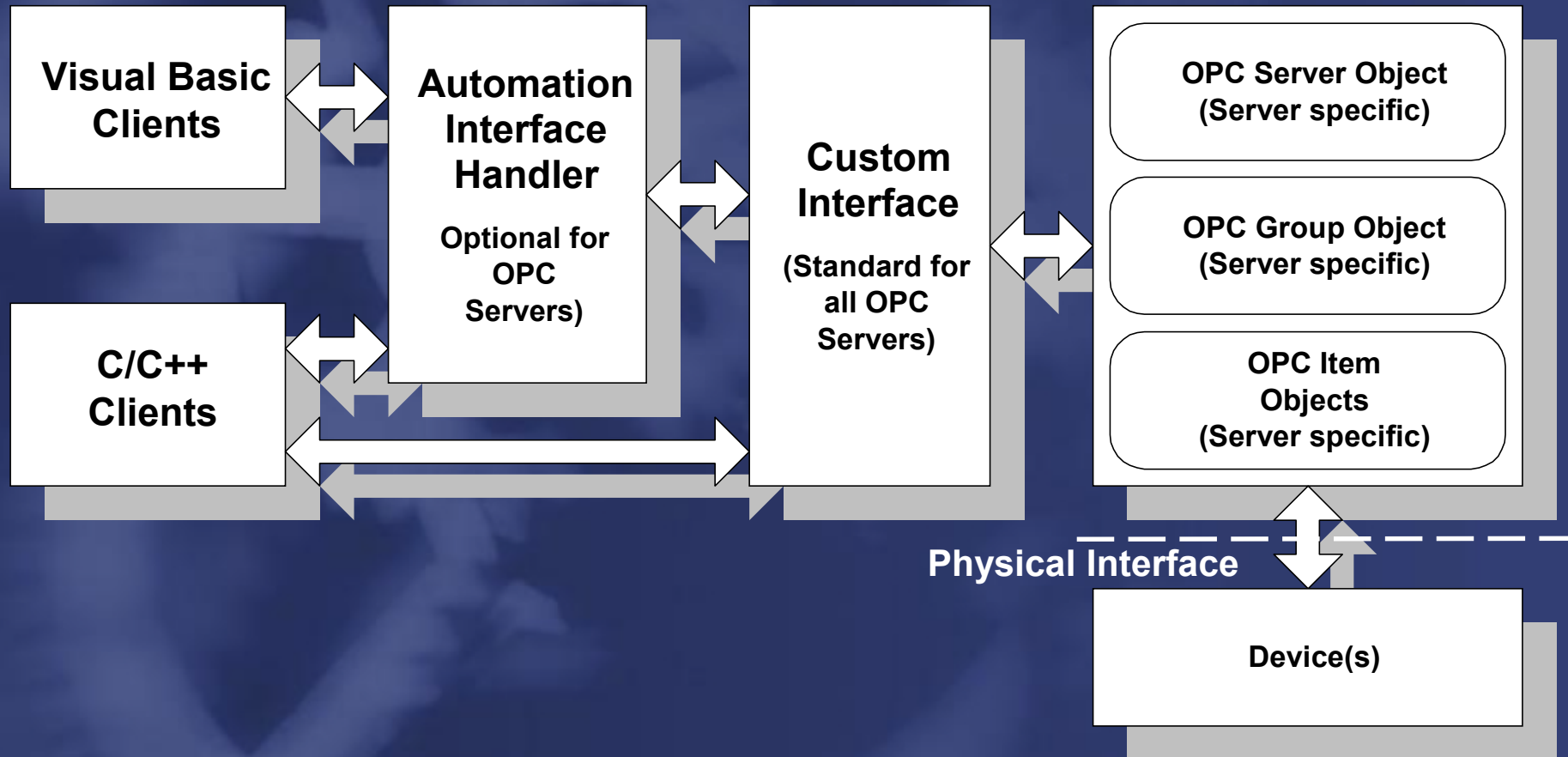


# Quali dati controlla OPC?

- ◆ Sensori Real time: dati - temp, pressione, flusso
- ◆ Controllo parametri - open, close, run, stop
- ◆ Status information (or Update)
  - Status della connessione hardware
  - Status del software locale e dei sottosistemi
- ◆ Allarmi su sensori dati - temp, pressione, flusso
- ◆ Allarmi su parametri di controllo - open, close, run, stop
- ◆ OPC può esporre ogni dato disponibile

OPC Interfaces			
Sensor Data	Comm Status	System Performance	OS Statistics

# Architettura OPC Server



# Specifiche OPC

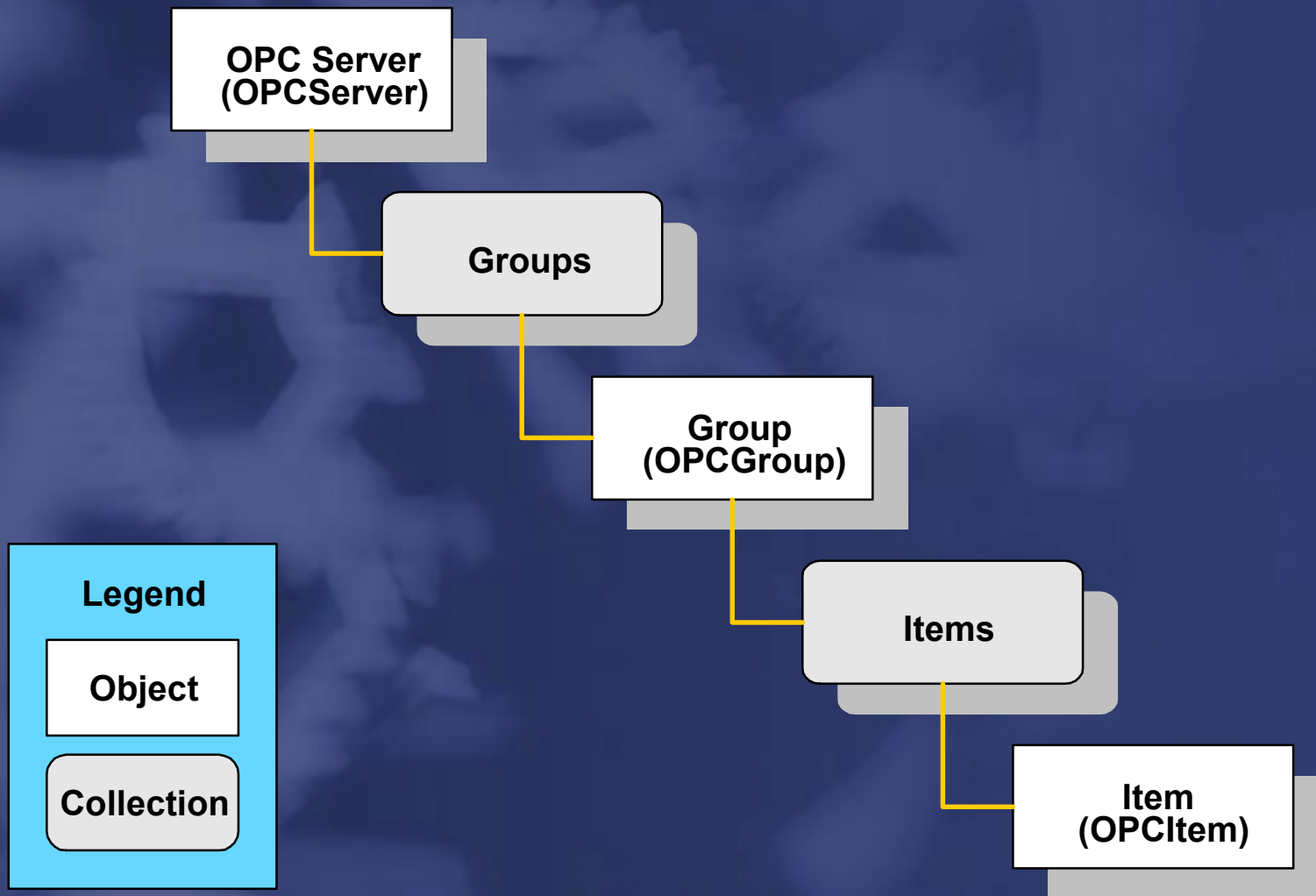
- ◆ 2 set di interfacce.
- ◆ **Custom Interface**
- ◆ **Automation Interface**
- ◆ OPC deve tener conto della tendenza di sviluppare applicazioni in ambienti diversi.
- ◆ L'intento della Microsoft è quello di facilitare l'evoluzione di Server OPC in C++ e quello dei Client OPC in altri linguaggi.
- ◆ le specifiche OPC non danno informazioni sull'implementazione delle Interfacce ma solo sulle loro caratteristiche.
- ◆ Come tutte le implementazioni Com , l'architettura OPC è un modello Client-Server in cui OPC Server procura una interfaccia a oggetti OPC e li gestisce.
- ◆ Il Server OPC è costituito da diversi oggetti quali: il **Server** , il **Gruppo** , l'**Item**.

# Interfaccia Server OPC

- ◆ **Tutte le funzionalità si basano su Application Programming Interfaces (APIs)**
- ◆ **E' difficile provvedere ad una singola API che lavori bene sia per C++ che per VB**
- ◆ **L'interfaccia OPC Custom è ottimizzata per l'uso di Client C++**
- ◆ **L'interfaccia Automation è ottimizzata per l'uso di Client VB**
- ◆ **Tipicamente OPC fornisce l' Interfaccia Custom come sua primitiva.**
- ◆ **L'Interfaccia Automation è costruita come strato superiore della Custom Interface.**



# OPC Object Model



# OPC Server Object

- ◆ **Gestisce, controlla e ottimizza l'accesso ai dati per clients multipli**
- ◆ **Ottiene informazioni sullo stato del server**
- ◆ **Contiene (aggiunge ed elimina) OPC Groups**

# L'interfaccia Server

## Campi principali :

- ◆ IOPCServer
- ◆ IOPCBrowseServerAddressSpace (optional)
- ◆ IOPCCommon **SetLocaleID, GetLocaleID, QueryAvailableLocaleIDs, GetErrorString, SetClientName**
- ◆ IOPCItemProperties(2.0)
- ◆ IConnectionPointContainer **EnumConnectionPoints FindConnectionPoint (Provides access to the connection point for IOPCShutdown)**



# OPC Group Object

- ◆ meccanismo conveniente per i clients per organizzare i dati
- ◆ Gruppi differenti possono essere usati da porzioni differenti di una applicazione
  - aggiornamenti multipli
- ◆ Contiene info sugli elementi e provvede alla loro organizzazione.
- ◆ Accede ai dati in modo sincrono e asincrono
- ◆ Si distinguono in pubblici e privati.
- ◆ I gruppi pubblici sono condivisi da più client e possiedono delle particolari interfacce.
- ◆ I privati sono gruppi esclusivi.

# L'interfaccia Group

- ◆ IOPCGroupStateMgt
- ◆ IOPCAsyncIO2 (2.0 - replaces IOPCAsyncIO)
- ◆ IOPCItemMgt
- ◆ IOPCSyncIO
- ◆ IConnectionPointContainer (2.0 replaces IDataObject)



# OPC Item Object

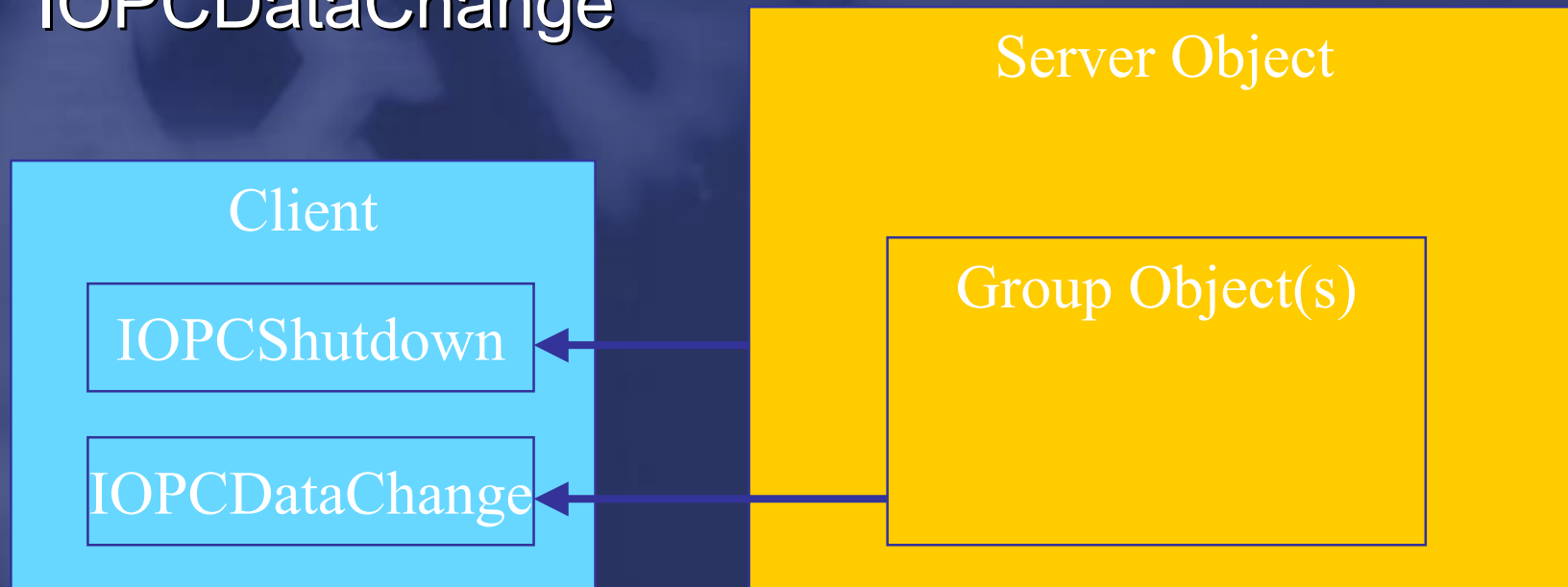
- ◆ Definiscono il punto I/O di un dispositivo.
- ◆ L'Item OPC non è accessibile come un oggetto da un client OPC e non possiede una Interfaccia esterna.
- ◆ L'accesso avviene tramite il Gruppo OPC in cui l'Item è contenuto o in cui è definito.
- ◆ Ad ogni Item è associato:
  - Valore corrente
  - qualità/status
  - timestamp

**Nota:** Gli elementi non sono dati,ma sono ad essi connessi.

**Gli elementi OPC devono essere pensati come una specifica sull'indirizzo dei dati.**

# Interfaccia dal lato Client

- ◆ Il Client ha 2 interfacce che il server può contattare
- ◆ IOPCShutdown: quando il server deve scaricare dati
- ◆ IOPCDataChange



# Vantaggi OPC

- ◆ **Riduce i costi di sviluppo** per i fornitori; riduce i costi di integrazione per gli utenti finali.
- ◆ I produttori di Hardware implementano un solo **driver, il server OPC**, per i loro componenti.
- ◆ Ai produttori software è richiesto solo un driver di interfaccia ,l'interfaccia **OPC Client**.
- ◆ **Maggiore flessibilità**: l'utilizzatore può combinare attraverso OPC qualsiasi sistema di visualizzazione o di controllo con qualsiasi hardware.
- ◆ **Incremento di qualità** dei driver e dei Server OPC:i costruttori possono concentrarsi su un singolo Server OPC e dato il minor numero di driver da sviluppare , ne conseguono **migliori caratteristiche di funzionalità,prestazioni e facilità di utilizzo**.
- ◆ Attraverso l'interfaccia OPC ora è possibile accedere al server OPC da più OPC Client



# Conclusioni

- ◆ OPC offre enormi vantaggi ai produttori di componenti per l'automazione.
- ◆ Gli utenti possono essere flessibili nella scelta dei moduli hardware e software.
- ◆ Attraverso la standardizzazione delle interfacce di comunicazione, i prodotti dei vari fornitori possono essere combinati e possono interagire senza modifica dei programmi.
- ◆ OPC consente di implementare il Plug&Play nelle applicazioni d'automazione e permette l'integrazione di bus di campo diversi.

# Funzionalità di OPC

- ◆ **“Namespace browsing”**: usato per la configurazione del sistema di acquisizione dati. I dati di processo vengono mostrati nel server OPC ,come si è abituati a fare in un menu di windows , e possono essere raccolti nel programma OPC Client attraverso la funzione drag&drop. Le caratteristiche del sistema rimangono così nascoste all'utilizzatore.
- ◆ Altra potenzialità di OPC è la **velocità di trasmissione dati** quando le variabili dinamiche di processo sono in numero elevato. I valori di 5000 variabili (numero realistico per i sistemi di visualizzazione) possono essere trasmessi tra il Server OPC ed il Client OPC in meno di un secondo sia localmente che usando due computer. Con sole 500 variabili di processo , il tempo si riduce a 100 ms.Sono interamente basati su OPC applicazioni fortemente critiche dal punto di vista del tempo,

# Esempi di applicazioni

- ◆ HVAC System @ National Instruments Austin, TX Headquarter
- ◆ sostituzione driver proprietari con OPC drivers, CERN (Ginevra)
- ◆ Supervisione Impianto Macinazione Concimi, CAGI (Varese)
- ◆ Monitoraggio variabili durante collaudo motore c/o FIAT AVIO, ARTS (Napoli)

