

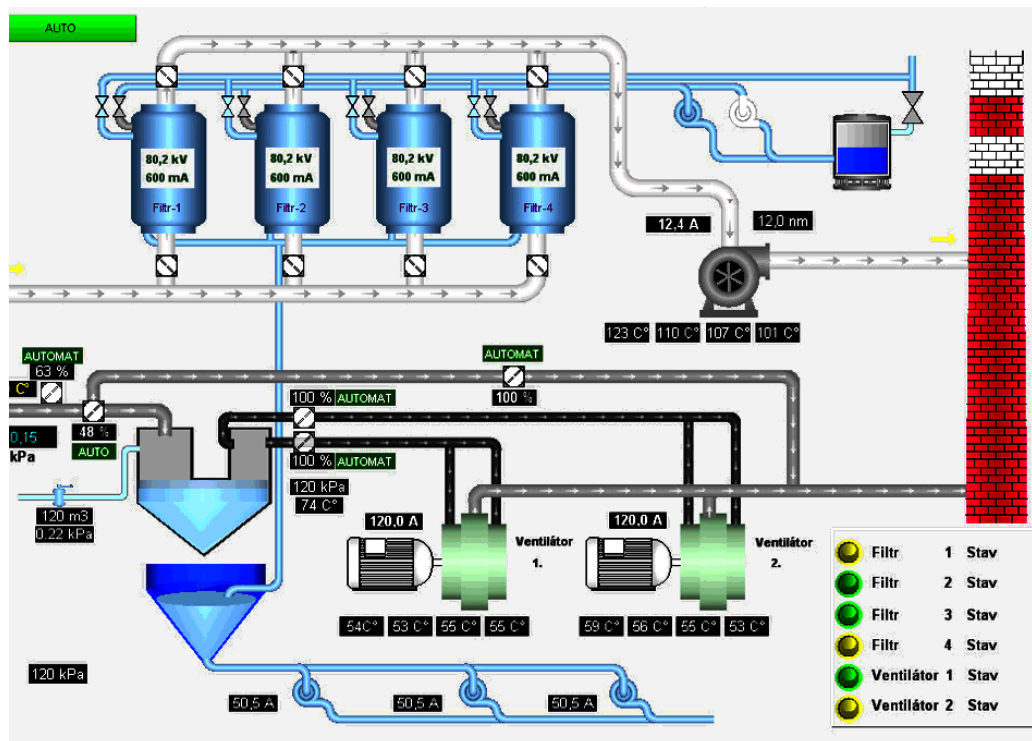


Lezione #4

Sistemi SCADA

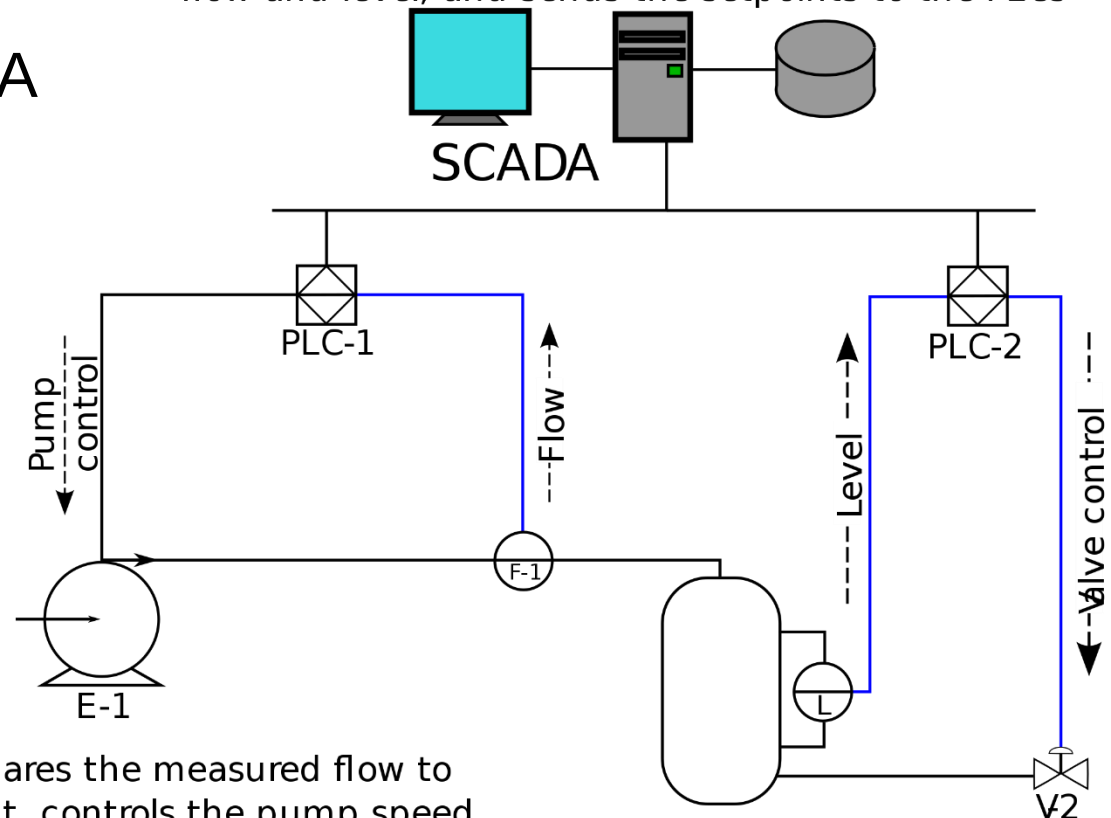
- Introduzione ai sistemi SCADA
- Principali funzionalità
- Esempio: DTT

- Uno **SCADA (Supervisory Control And Data Acquisition)** è un software che consente all'operatore di interfacciarsi con il processo attraverso i sistemi di controllo



- Schema di un semplice SCADA

The SCADA system reads the measured flow and level, and sends the setpoints to the PLCs



PLC1 compares the measured flow to the setpoint, controls the pump speed as required to match flow to setpoint

PLC2 compares the measured level to the setpoint, controls the flow through the valve to match level to setpoint

- **Acquisizione dati**

- Per poter fornire informazioni sul processo, è necessario che il sistema **acquisisca dati** da esso mediante opportuni **driver di comunicazione**
- Questi dati vengono raccolti in una **base dati**, che costituisce il nucleo del sistema SCADA
- Ai dati grezzi possono essere aggiunte altre informazioni come l'identificativo del sensore, stringhe informative, unità di misura e/o conversione del valore in unità standard...

- **Controllo di supervisione**

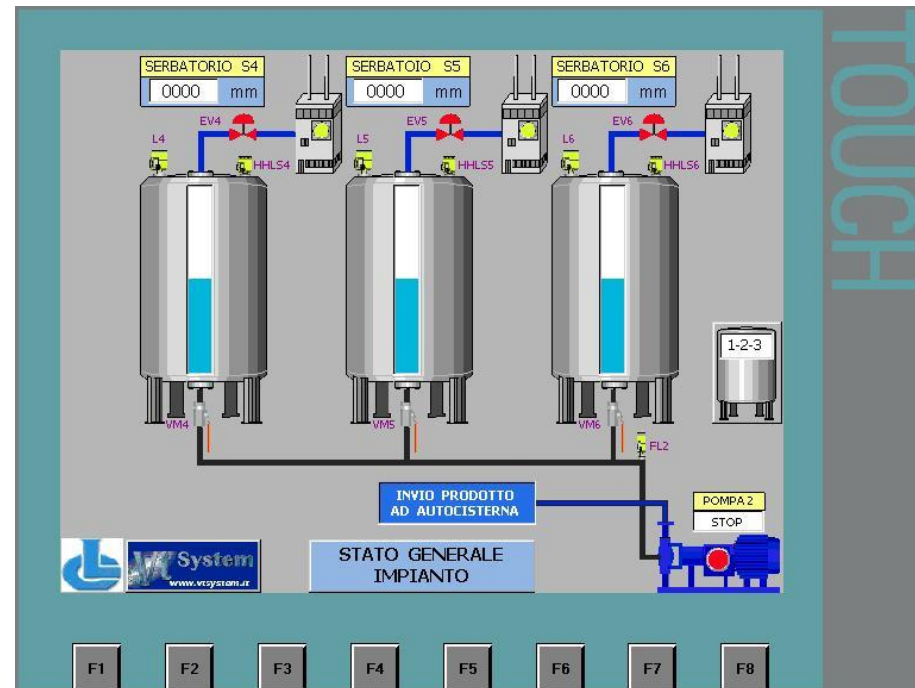
- Tramite uno SCADA, l'operatore può **supervisionare il processo** ed intervenire su di esso se necessario
- Questa funzione si realizza tipicamente tramite una **Human-Machine Interface (HMI)** di tipo grafico

Oltre a queste, uno SCADA moderno in genere assolve ad altre funzionalità

Rappresentazione dei dati

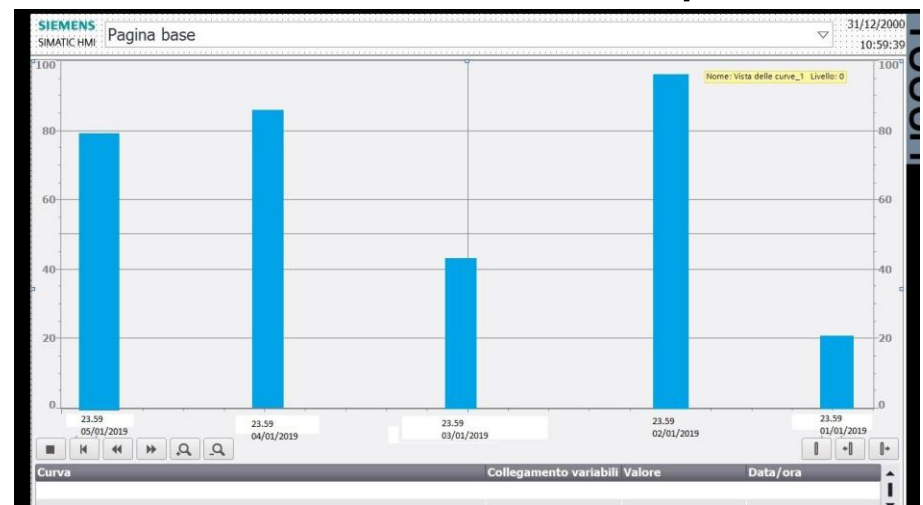
Spesso lo SCADA presenta i dati all'operatore tramite una HMI, che in genere comprende:

- **Quadri sinottici**, con elementi statici e dinamici (a seconda che l'aspetto grafico vari in base al dato)
- **Pannelli di controllo**, consentono all'operatore di interagire con l'impianto in maniera intuitiva (manopole, pulsanti, slider...)



Storicizzazione dei dati

- Spesso, gli SCADA possono anche restituire **serie storiche di dati** per fare analisi di tipo gestionale e tenere traccia dei ritmi di produzione (efficientamento del processo, gestione di criticità...)
- Possono anche generare automaticamente dei *report* a intervalli prefissati



Gestione di allarmi

- Lo SCADA consente di programmare la condizione che causa l'insorgenza di un allarme
- All'insorgere di una condizione di allarme, lo SCADA provvede a **informare l'operatore** (lampeggianti, sirene, e-mail, SMS...)



Gestione di allarmi

- Tipi di allarmi:
 - **A insorgenza**
Permangono fino a una reinizializzazione del sistema
 - **A insorgenza e riconoscimento**
Permangono fino a quando l'operatore non ne comanda la disattivazione (*riconoscimento*)
 - **A insorgenza, riconoscimento e rientro**
Permangono finché non avviene il riconoscimento e la condizione di attivazione viene meno



Gestione di ricette

- Gli SCADA possono eseguire particolari sequenze di operazioni programmate dall'utente (dette *ricette*, *lotti* o *batch*)
- Tali ricette possono essere eseguite:
 - Al verificarsi di una **scadenza temporale** prefissata
 - All'insorgere di **eventi** particolari (ad esempio in risposta a un allarme)
 - Su **richiesta** dell'operatore



Supporto alla manutenzione

- La manutenzione può essere **correttiva** o **preventiva**
- Lo SCADA può fornire all'utente dei **piani di manutenzione**
- Può anche aiutare a coordinare tali operazioni, che potrebbero dover essere eseguite su un **numero elevato** di dispositivi



Interazione con sistemi di livello superiore

- Agli SCADA moderni è spesso richiesto anche di interfacciarsi con altri sistemi
- Esempi:
 - *Domain Controller* per la gestione degli accessi e l'autenticazione degli utenti
 - Sistemi MES (*Manufacturing Execution System*) per gestire il dispaccio di ordini, le scorte di magazzino e in generale per gestire e controllare la produzione
 - Software gestionali ERP (*Enterprise Resource Planning*)

Evoluzione dei sistemi SCADA

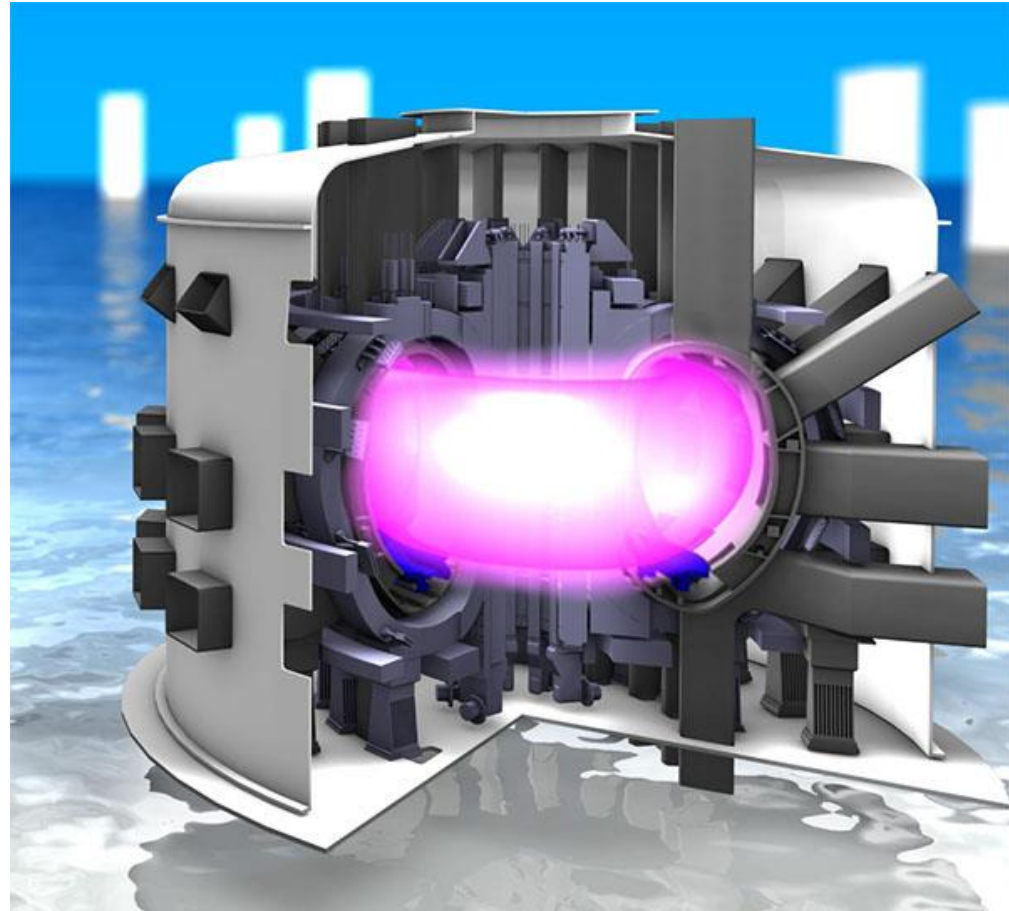
- Nel corso del tempo, i sistemi SCADA hanno subito una significativa evoluzione
- I primi SCADA erano spesso realizzati su mainframe o [minicomputer](#) con sistemi di comunicazione proprietari
- Oggi si va nella direzione di sistemi **distribuiti**, spesso **web-based**
- La connessione dei sistemi SCADA a reti aziendali o a internet ha aumentato i **rischi per la sicurezza** in tali sistemi (si pensi ad esempio al caso di [Stuxnet](#))

Esempio: DTT

Il **Divertor Tokamak Test facility** (DTT) è un prototipo di reattore a fusione nucleare che dovrebbe essere costruito nei prossimi anni nei laboratori di ENEA Frascati

I **Tokamak** sono macchinari complessi, in cui le funzioni di controllo e monitoraggio svolgono un ruolo critico

Nell'ultimo Interim Design Report è incluso anche il design concettuale di un sistema SCADA



Esempio: DTT

DICS: DTT Instrumentation and Control System

“

DTT Instrumentation and Control System, DICS, is a combination of systems and subsystems that collectively allow automating the operation of the DTT device.

- DTT Interim Design Report, 2019

Esempio: DTT

Componenti:

- **Central control**
 - Central Command and Control System (**CCCS**)
 - Central Machine Protection Systems (**CMPS**)
 - Central Occupational Safety Systems (**COSS**)
- **Local control**
Plant System Controllers, organizzati in Plant System Groups (**PSGs**)
- **Distributed control**
Real Time Control Systems (**RTCS**)
- **Reti di interconnessione**

CCCS

- Funzioni di controllo e coordinazione **non critiche e non real-time**
- Interfacce per gestire le operazioni, terminali utente, server per gestire il database dell'impianto e le ricette, meccanismi di comunicazione agli altri sistemi

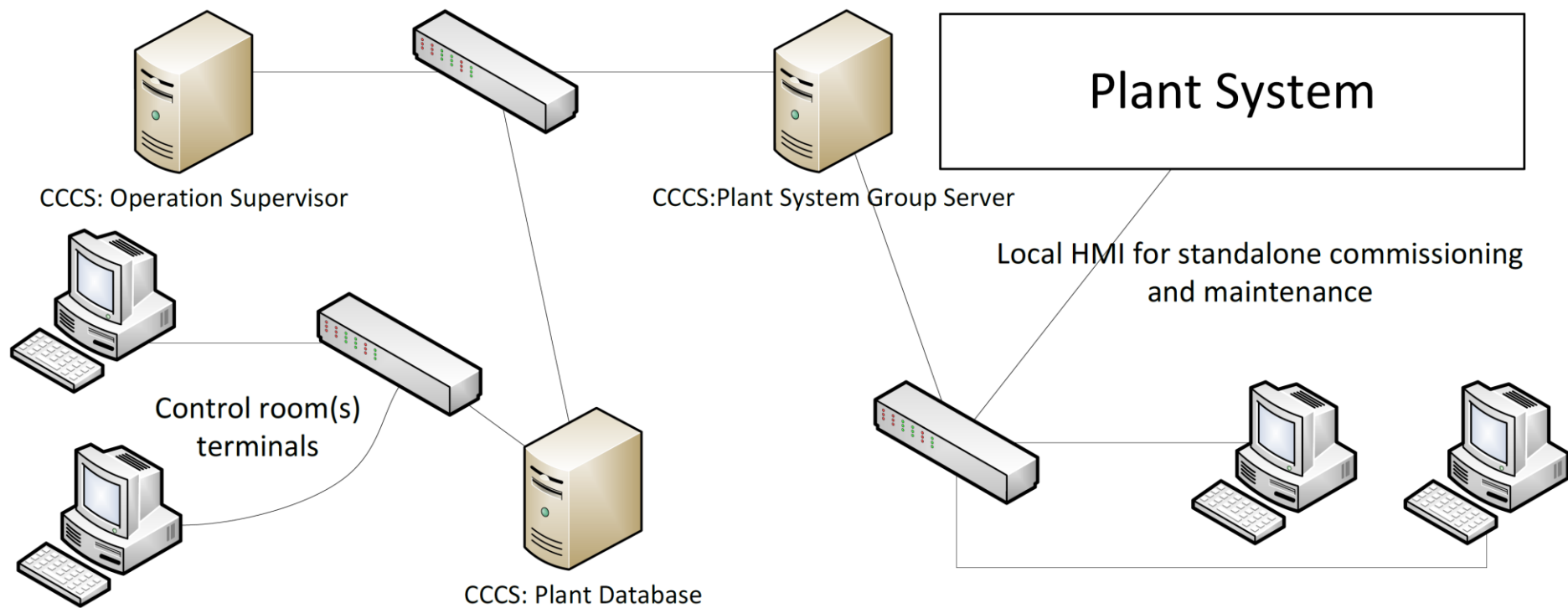
CCCS

- Database
- Interfacce operatore e quadri sinottici
- Programmazione degli esperimenti
- Accesso ai dati sperimentali e Historical Experiment Database **(HED)** (→ MDS+)
- Coordinazione automatica dei sottosistemi **(Operation Supervisor)**

Esempio: DTT

Central Control System

CCCS



CMPS

- Funzioni di controllo e coordinazione **critiche e real-time**
- Gestisce guasti e malfunzionamenti per minimizzarne gli effetti negativi
- Gestisce anche il clock, le reti di comunicazione RT e le reti per la Machine Protection
- Le strategie di gestione effettive sono implementate a livello dei *Plant Systems*, e vengono coordinate dal CMPS

CMPS

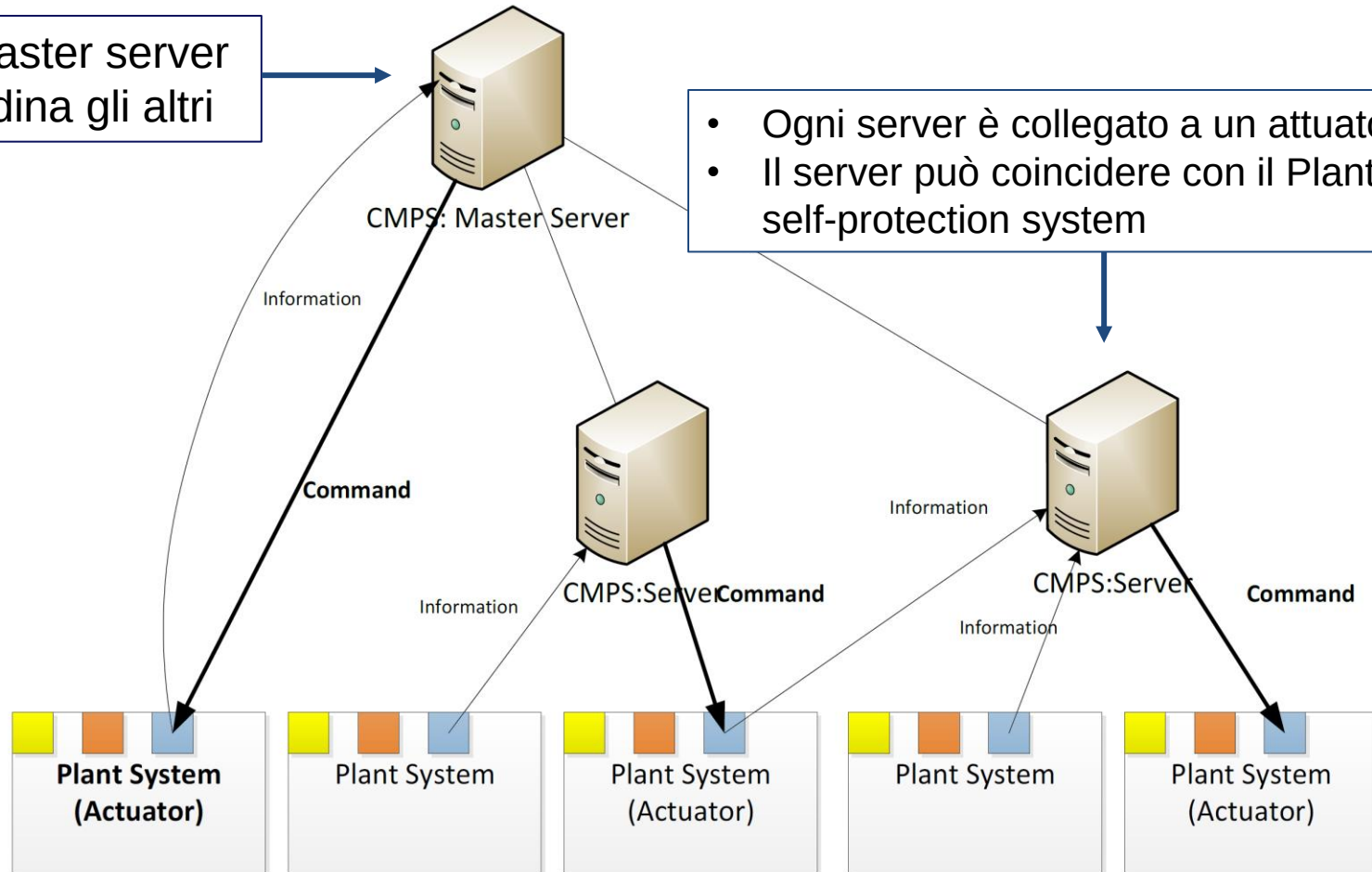
Critical vs non-critical protection functions

- sono considerate critiche le condizioni che possono causare gravi ritardi nelle operazioni o costi di manutenzione/riparazione elevati
- *quality-control* necessario su tutte le componenti coinvolte nella gestione di queste evenienze

Esempio: DTT

Central Control System

Un master server coordina gli altri



- Ogni server è collegato a un attuatore
- Il server può coincidere con il Plant-System self-protection system

Il **numero** e i **compiti** di ogni server sono scelti in modo da **minimizzare la complessità del sistema!**

COSS

- Funzioni di controllo e coordinazione **critiche e real-time**
- Gestisce gli eventi che possono causare danno a persone in prossimità dell'impianto
- Include il controllo degli accessi ad aree potenzialmente pericolose, funzioni di *interlock* (ad esempio per sorgenti di alta tensione...), ecc.

Plant Systems

- **Standard Industrial Control Platform: PLC**
 - Main+backup
 - Possibili framework per lo sviluppo di SW in-house sono UNICOS e ITER Industrial Control Library
 - *‘Whenever a tender will be placed, some level of contractual pressure will be placed so that tenderers will adopt this platform’*

Plant Systems

- **Standard Computer Platform: Linux**
 - Distro standard per il CCCS
 - Installazioni minimali con kernel semplificati per i Plant System (disponibilità ↑ - vulnerabilità ↓)
 - Real-Time dove necessario
 - Framework [MARTe2](#) per lo sviluppo di applicazioni di controllo RT

Plant Systems

- **Data Acquisition Hardware**
 - **PLC remote I/O cards** per performance «moderate» (<1kHz)
 - **National Instruments C-RIO + Linux PC** nei casi in cui sono necessarie performance più spinte (con l'uso di FPGA)
 - Piattaforma per prestazioni elevate **ancora da definire**

Plant Systems

- **Interfaccia col CCCS** - soluzioni «preferite»:
 - Native interface to the chosen Supervisory Control and Data Acquisition (SCADA)
 - [OPC-UA](#)
 - [REST](#)ful

RTCS

- Funzioni perlopiù **non critiche**
- Controllori, sistemi di diagnostica e attuatori specifici per l'applicazione considerata
- Framework MARTe2 + Linux per lo sviluppo

Networks

- **Command and control network**
 - Ethernet-based
 - Comunicazione tra CCCS e server PSG, terminali in control-room, partecipanti remoti
 - *Field network* basata su un mix di IP e *fieldbus*
- **Data Acquisition Network**
 - TCP/IP-based
 - Usata per trasmissioni a frequenze $>100\text{Hz}$
 - I dati non vengono conservati localmente nei Plant System

Networks

- **Real Time Network**
 - Protocollo Point-to-multipoint (MULTICAST) UDP
 - Traffico stazionario: tutti i nodi contribuiscono con **carichi noti** e con **pattern periodici** sincronizzati al *central timing system*
 - Topologia: **switched multi-star**
- **RTN Simulation Support**
 - La rete RT deve anche consentire test *system-in-the-loop* per testare combinazioni di Plant System usando input sintetici
 - Simulazioni offline per testare il software real-time dei Plant System su piattaforme PC (in **Simulink**)

Esempio: DTT

Local Control System

- La tecnologia SCADA da adottare è in fase di definizione
- Preferenza per sistemi **industriali** e **scalabili** che possano essere facilmente integrati con altri prodotti (e.g. [WINCC-OA](#))

Riferimenti e Risorse

- [3] Cap. 12
- [What is SCADA?](#)@RealPars 
- [Modbus](#)@RealPars (→ ScadaBR) 
- [Lista di software SCADA](#) open source
- [Note su SCADA](#) (S. Bimbo e E. Colaiacovo)
- [ERP, MES, PLM](#)
- [DTT Interim Design Report](#)



Fine Lezione #4

Sistemi SCADA