

# Scenari di integrazione

## 1. Integrazione col mondo dell'automazione

Le macchine automatiche sono controllate da sistemi che non sono dei computer ma dei **PLC**. L'integrazione con il resto del mondo concerne spesso l'esportazione **real-time** di dati di sensori della macchina, su cui si opera con vari sistemi di analisi, big data, intelligenza artificiale. L'integrazione richiede la conoscenza di protocolli di comunicazione specifici per l'ambiente dell'automazione ed i meccanismi di sicurezza.

## 2. Sistemi a micro-servizi

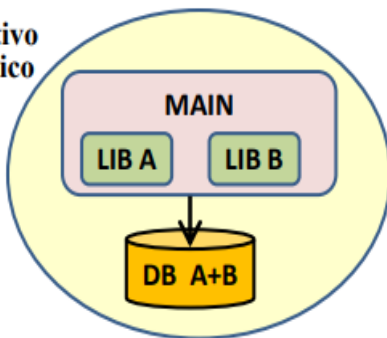
**Applicativi monolitici** (applicativi costituiti da un solo pezzo o al massimo 2)

- l'applicazione in Java, o il servizio web
- il servizio web + il tier del DB

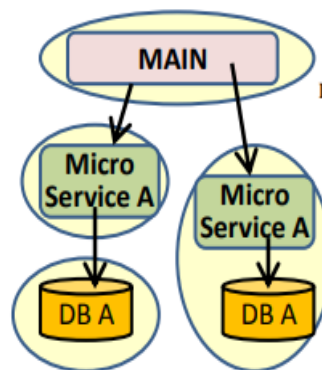
**Applicativo a micro-servizi**

- più componenti separati e customizzabili, detti **micro-servizi**
- in esecuzione sullo stesso host o più host distribuiti in rete
- componenti che interagiscono tra loro **scambiandosi messaggi** mediante **interfacce software e protocolli standard**
- consentono il **riutilizzo del software**

applicativo monolitico



applicativo a micro-servizi



N.B -> al posto di main ci potrebbe essere un servizio web

### Vantaggi

- **Riusabilità:** le componenti (trasformate in micro-servizi) possono essere utilizzate da diversi applicativi
- **Scalabilità:** vengono replicati e bilanciati solo i micro-servizi che necessitano
- **Deployment:** può essere realizzato sulla stessa macchina che su macchine differenti, riconfigurando i servizi di comunicazione per utilizzare gli opportuni indirizzi
- **Cloud:** i micro-servizi sono il modo ideale per affrontare lo sviluppo di applicativi orientati nativamente al cloud computing

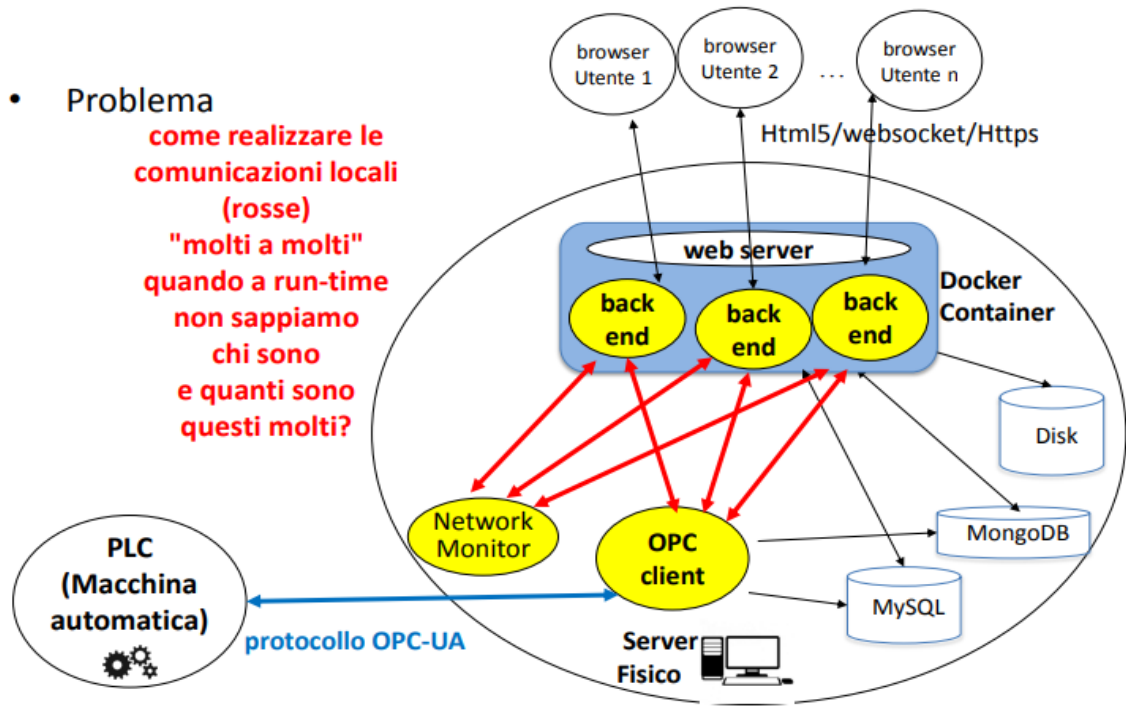
### Svantaggi

- i micro-servizi devono utilizzare soltanto interfacce software e protocolli di comunicazione diffusi e condivisi (**standard de facto**)
- progettare le applicazioni stesse affinché utilizzino solo questi standard

- assicurare che la rete di comunicazione permetta il passaggio di questi messaggi. In fase di progettazione e dispiegamento inserire opportune infrastrutture e servizi per superare il **firewall** e il **NAT**

- **Problema**

come realizzare le comunicazioni locali (rosse)  
"molti a molti"  
quando a run-time non sappiamo chi sono e quanti sono questi molti?



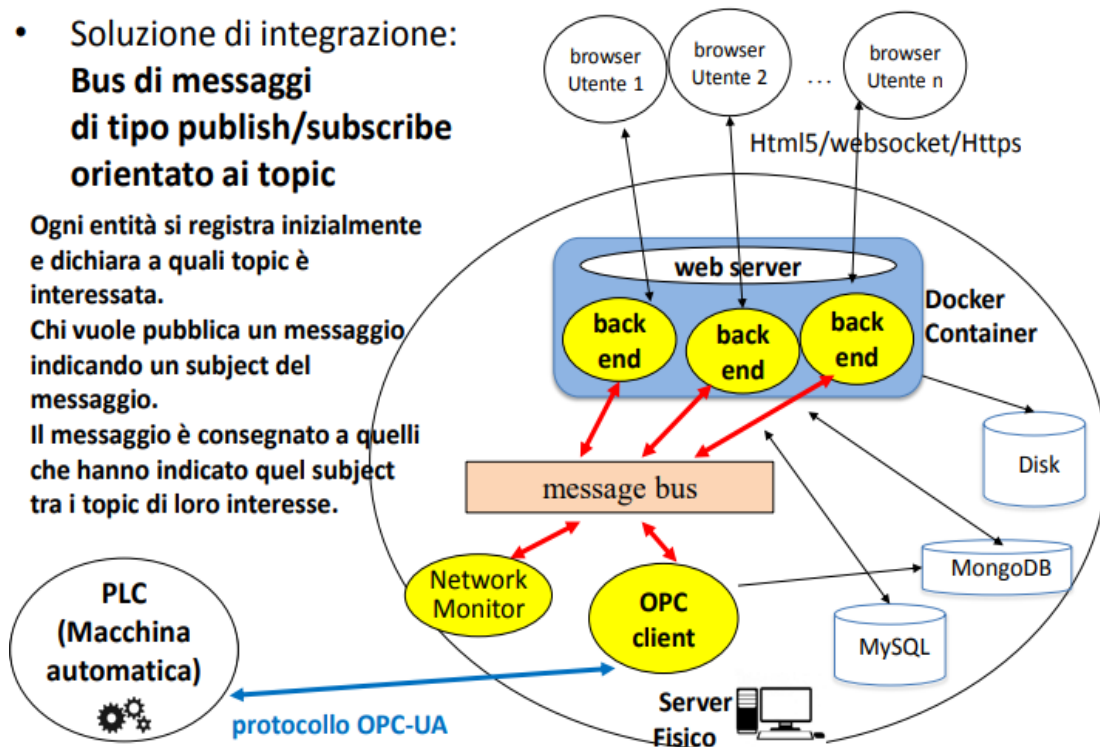
**OPC client:** è un servizio offerto dal server che ha il compito di ricevere i dati dal PLC e di comunicarli ai vari **DB**. Per comunicare con il PLC sfrutta un protocollo di comunicazione **OPC-UA**

- **Soluzione di integrazione:**  
**Bus di messaggi di tipo publish/subscribe orientato ai topic**

Ogni entità si registra inizialmente e dichiara a quali topic è interessata.

Chi vuole pubblica un messaggio indicando un subject del messaggio.

Il messaggio è consegnato a quelli che hanno indicato quel subject tra i topic di loro interesse.



## Bus di messaggi (Message Broker)

Esistono diversi protocolli per lo scambio di messaggi, tra i principali:

- AMQP
- MQTT

Hanno il compito di definire un formato per i messaggi trasportati, che include un **body**. Permettono di scegliere le modalità di consegna dei messaggi --> (affidabile, non affidabile, con ricevuta di consegna, con timeout, ...). Mettono a disposizione API per diversi linguaggi, anche a base web e per diversi OS.

Esempi di implementazioni dei bus di messaggi: **RabbitMQ, Microsoft Azure Service Bus Messaging**

## 3. Sistemi a micro-servizi dispiegati in cloud

Un sistema può appoggiarsi parzialmente o completamente a micro-servizi operanti su infrastrutture **cloud**.

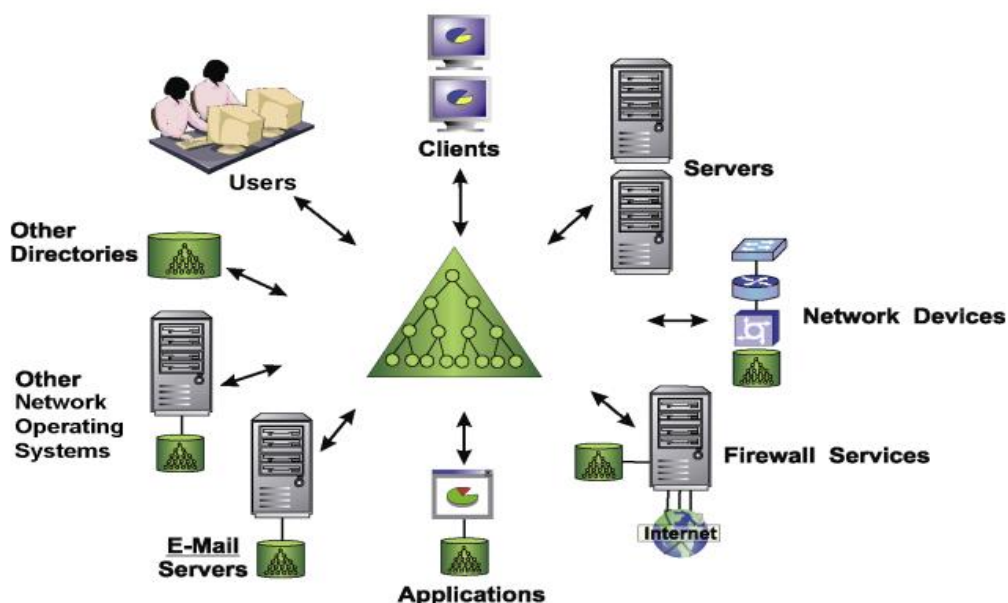
Vedi esempio di scenario sulla slide **Scenari di Integrazione** a pagina 21-22.

## 4. Sicurezza nei sistemi informatici

La **sicurezza** è un **criterio base di progettazione** trasversale a tutti i sistemi. Molto spesso, un requisito di progettazione è la centralizzazione in un servizio unico delle funzionalità di **autenticazione** e **autorizzazione** all'uso di risorse.

Quest'ultimo requisito è esso stesso un **fattore di sicurezza** poiché limita i punti di attacco e favorisce il controllo. → I servizi di **Directory** si occupano anche di autenticazione e autorizzazione.

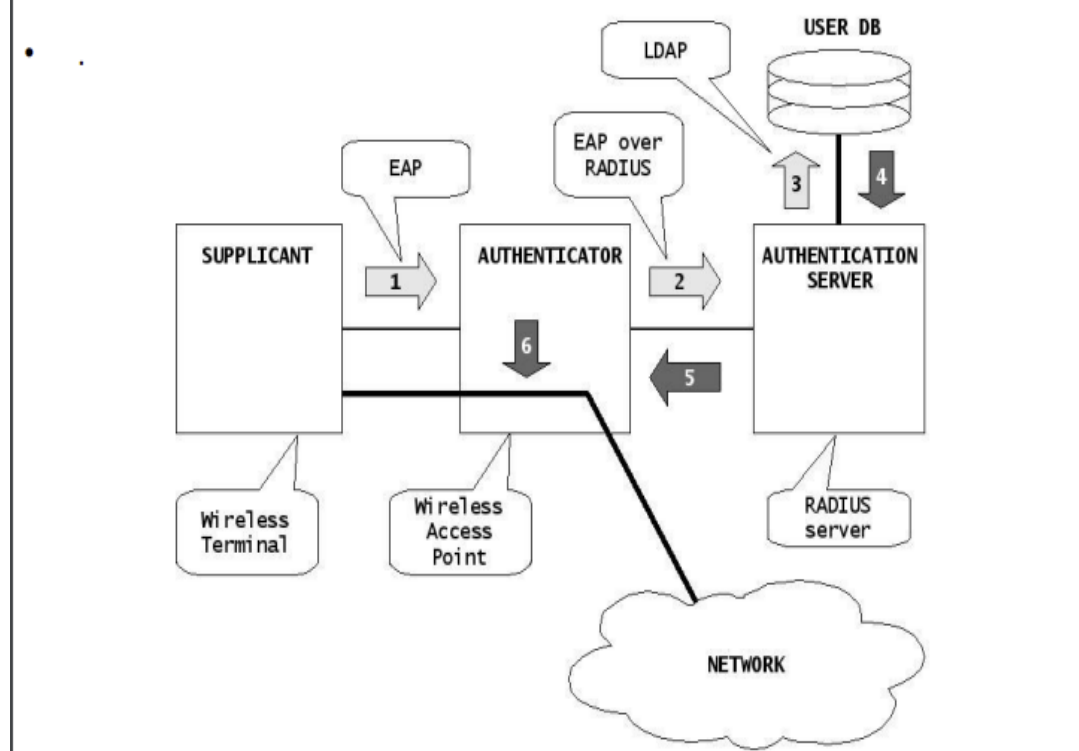
## 5. Directory Service



Vedere esempio sulla slide **Scenari di integrazione** a pagina 27-28.

## Autenticazione in canali wireless basata su Directory Service

### Autenticazione per accesso a AP WiFi e retrostante rete basata su **Server Radius** (WPA2, 802.1x e Active Directory)



Lo standard **WPA2** utilizzato nelle reti wireless per la sicurezza prevede di utilizzare 802.1x per gestire l'autenticazione. → **IEEE 802.1x** è uno standard per l'autenticazione e l'autorizzazione in rete basato sul protocollo **EAP** ( Extensible Authentication Protocol) per l'autenticazione.

Prevede tre entità:

- **Authenticator:** chiede l'autenticazione prima di fornire il servizio
- **Supplicant:** vuole accedere al servizio e deve essere autenticato
- **Authentication server:** verifica le credenziali del supplicant a nome dell'autenticator

**Single-Sign-On** → meccanismo che concede all'utente l'autorizzazione all'uso di più applicazioni/servizi con una sola richiesta di credenziali. L'implementazione del **SSO** può essere fatta in tanti modi diversi a seconda di:

- Dove si trova il computer su cui opera l'utente
- Quale sistema operativo opera sul computer dell'utente
- Se le applicazioni da utilizzare sono a **base web o no**.

Tra i sistemi di SSO per servizi web il più usato è **Shibboleth**.

Vedi esempi sulla slide **Scenari di Integrazione** a pagina 35-36.

## 6. Dispiegamento di servizi scalabili e isolati mediante Virtualizzazione e Containerizzazione

**Virtualizzazione:** è la creazione di una rappresentazione virtuale e non fisica di qualcosa.

**Virtualizzazione del server** → tecnologia che permette di eseguire un sistema operativo **ospitato** isolandolo all'interno di una macchina che non è fisica bensì ottenuta da uno strato software denominato **Virtual Machine** o **Hypervisor**.


Il pregio della Virtualizzazione è di **isolare un'applicazione all'interno del S.O installato sulla VM**.

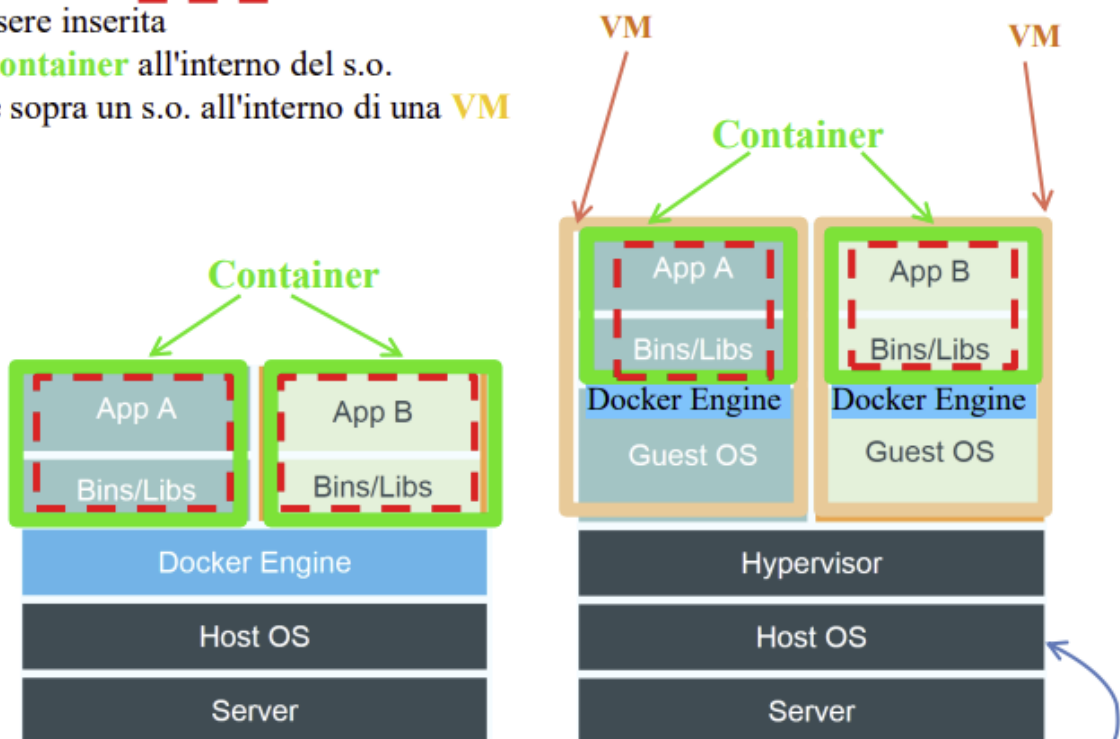
**Containerizzazione:** questa tecnologia consiste nel **virtualizzare l'applicazione**, creando per l'applicazione un contesto di esecuzione che non è più tutto un server.

I **Container** realizzano un sottoinsieme delle risorse offerte dal S.O e le mettono a disposizione alle applicazioni che vengono eseguite all'interno dei **Container**.

I Container isolano le applicazioni all'interno del S.O ospitato, in questo modo le applicazioni sono dipendenti dal container ma non dalla versione e configurazione del S.O ospitato nella VM.

### Container vs. Virtual Machine

L'applicazione   
può essere inserita  
in un **container** all'interno del s.o.  
oppure sopra un s.o. all'interno di una **VM**



NB: alcuni hypervisor si sostituiscono al sistema operativo, quindi qui l' Host OS potrebbe non esserci

## Cloud e categorie di servizi

Il **Cloud** di un provider è essenzialmente un gruppo di datacenter in ciascuno dei quali vengono forniti dei micro-servizi.

- I micro-servizi più essenziali sono macchine virtuali per le quali è possibile richiedere certe tipologie di CPU e caratteristiche HW, ammontare di memoria, spazio su disco etc...
- Su micro-servizi base possono essere costruiti altri micro-servizi più complessi
- Alcuni micro-servizi più raffinati permettono di gestire ridondanza dei dati e failure recovery verso altri data center

### Tipologia dei servizi cloud

- **Software as a Service (SaaS)**: un servizio applicativo che non vede né sistema operativo né host su cui lavora. Es → google documents
- **Platform as a Service (PaaS)**: fornisce delle API di sviluppo per comporre servizi più complessi. E' scalabile in modo trasparente. Es → CosmosDB
- **Infrastructure as a Service (IaaS)**: fornisce un'infrastruttura su cui installare servizi  
Es → macchina virtuale

## 7. Progettazione comunicazioni in reti protette da NAT e Firewall

### VPN (Virtual Private Network)

- **Site-to-site**: sono VPN instaurate tra due reti, per esempio tra due sedi distaccate di una stessa azienda per le quali si vuole usare una rete privata unica.
- **Remote access VPN**: collegano un singolo computer ad una rete privata tramite VPN.
- **Problema di sicurezza**: la compromissione di un computer esterno, collocato in una rete di dominio grazie a una VPN, rischia di compromettere tutta la rete del dominio.

Vedi esempi di scenari su NAT e Firewall nella slide **Scenari di Integrazione** da pagina 48.