**Scenari di integrazione**

1. **Integrazione col mondo dell’automazione**

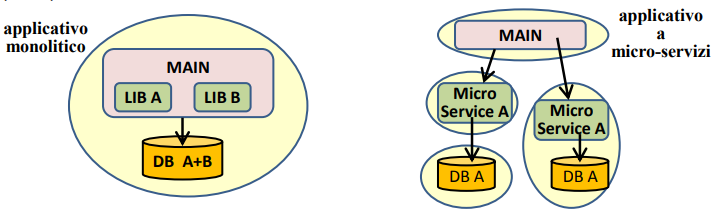
Le macchine automatiche sono controllate da sistemi che non sono dei computer ma dei **PLC**. L’integrazione con il resto del mondo concerne spesso l’esportazione **real-time** di dati di sensori della macchina, su cui si opera con vari sistemi di analisi, big data, intelligenza artificiale. L’integrazione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

1. **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**

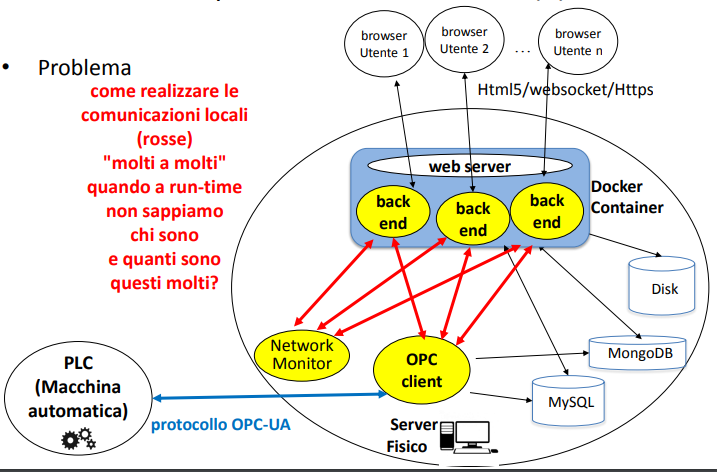
* più componenti separati e customizzabili, detti **micro-servizi**
* in esecuzione sullo stesso host o più host distribuiti in rete
* componenti che interagiscono tra loro **scambiandosi messaggi** mediante **interfacce software e protocolli standard**
* ****consentono il **riutilizzo del software**

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

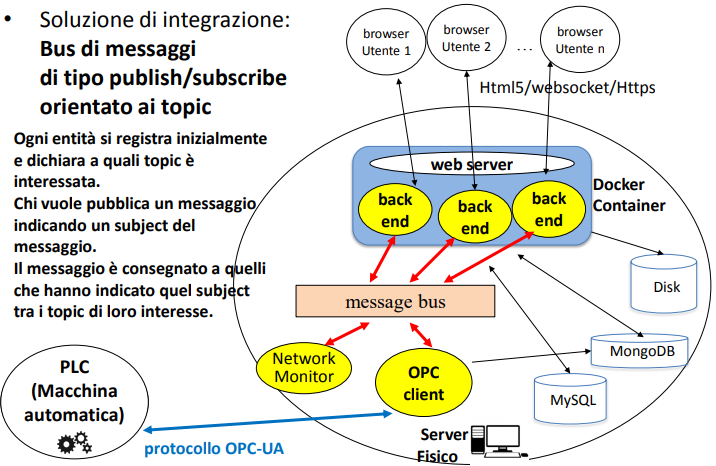
**Vantaggi**

* **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 microservizi sono il modo ideale per affrontare lo sviluppo di applicativi orientati nativamente al cloud computing

**Svantaggi**

* i microservizi devono utilizzare soltano interfacce software e protocolli di comunicazione diffusi e condivisi (**standard de facto**)
* progettare le applicazioni stesse affinchè 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**

**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**

****

**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**