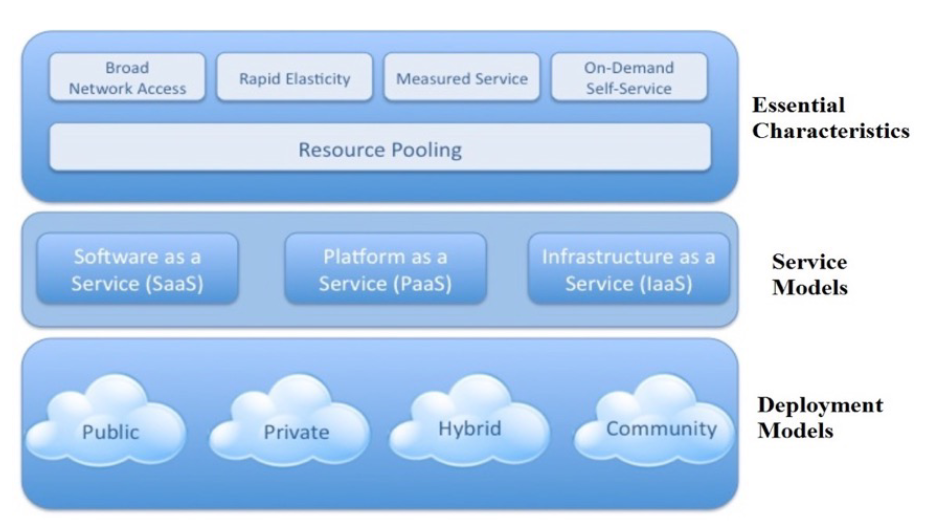
# Definizione di Cloud Computing

L'edge computing è un modello/tecnica per l’accesso alla rete che sta alla base del IoT. È perfetto per applicazioni che richiedono risposte in tempo reale.

* **Elaborazione Locale**: I dati vengono elaborati vicino alla loro fonte di origine, riducendo la necessità di trasferirli a grandi distanze per l'elaborazione.
* **Riduzione della Latenza**:
* **Efficienza dei Dati**: Non tutti i dati raccolti dai dispositivi periferici sono necessari per il processamento centrale. L'edge computing consente di filtrare e aggregare i dati localmente, inviando solo quelli rilevanti al cloud per ulteriori analisi.
* **Sicurezza e Privacy**: Mantenere i dati vicino alla loro origine può ridurre i rischi associati al trasferimento di dati sensibili attraverso la rete.
* **Capacità di Elaborazione Distribuita**: Utilizza nodi di elaborazione distribuiti (edge nodes) che collaborano con il cloud e le infrastrutture di rete, offrendo una soluzione più scalabile e resiliente rispetto ai modelli centralizzati tradizionali.

Il cloud computing è l’esatto opposto dell’edge computing. È meno sicuro (invio dei dati in rete) ma più scalabile ed efficiente (non dipende dai dispositivi IoT ed è meno costoso). Più adatto per elaborazioni su grande scala e backup.

Il Cloud Computing può essere spiegato attraverso il modello NIST:

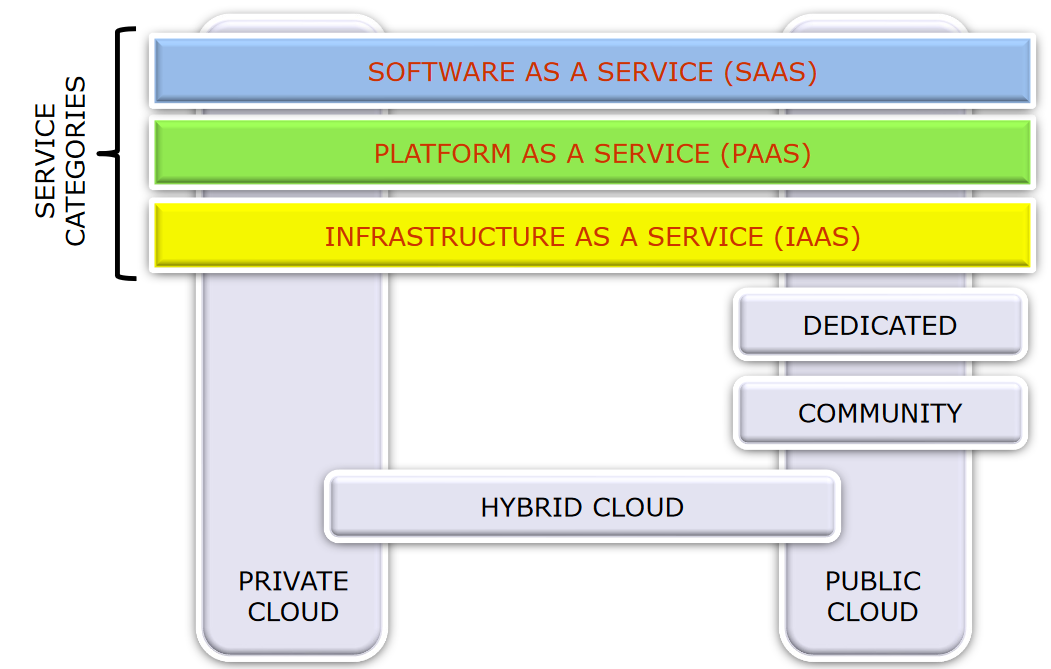


### Caratteristiche Essenziali del Cloud Computing

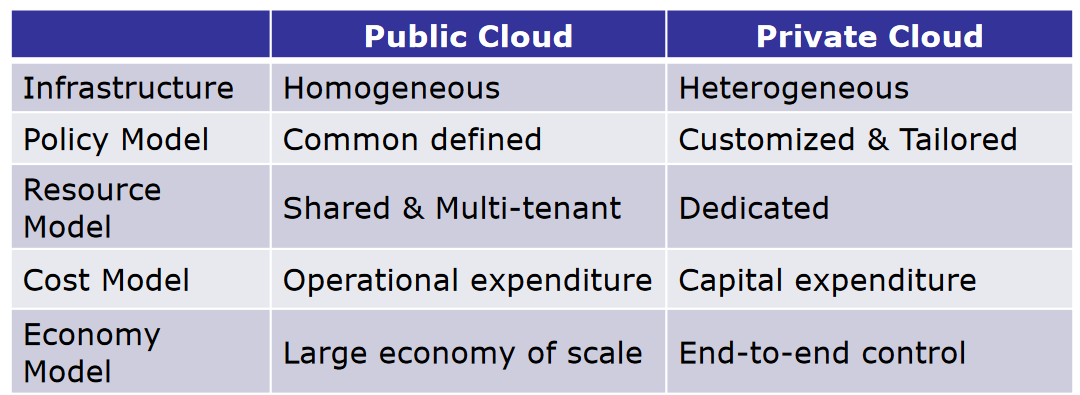
* **Self-service su richiesta**: senza richiedere un interazione umana.
* **Accesso alla rete su vasta scala**: Disponibile sulla rete e dispositivi diversi.
* **Elasticità rapida**: Scalare rapidamente verso l'alto o verso il basso in maniera automatica.
* **ti spiano**

### Modelli di Servizio - SPI

* **Software as a Service (SaaS)**: tipo google drive
* **Platform as a Service (PaaS)**: Consente agli utenti di sviluppare, eseguire e gestire applicazioni senza preoccuparsi dell'infrastruttura sottostante.
* **Infrastructure as a Service (IaaS)**: tipo i DB di amazon



### Modelli di Deployment

* **Cloud Pubblico**: di proprietà di un'organizzazione che vende/affitta servizi cloud | infrastruttura omogenea e risorse condivise. È offerta a terze parti con mobilità standard.
* **Cloud Privato**: È usato esclusivamente da un'organizzazione | risorse dedicate, infrastruttura eterogenea.
* **Cloud Ibrido**: Combina cloud pubblici e privati
* **Cloud Comunitario**: condiviso da diverse organizzazioni con interessi comuni.

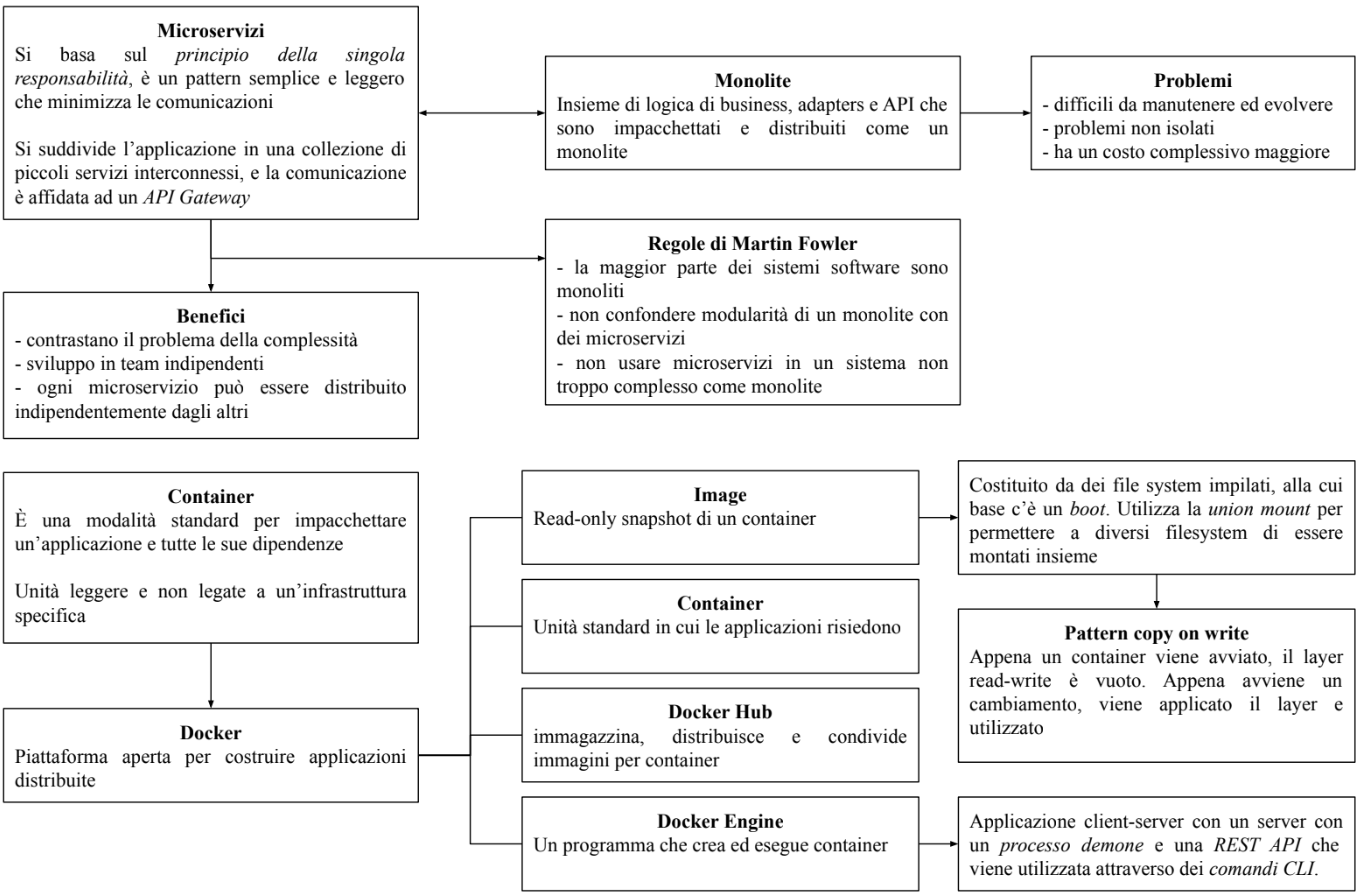
### Virtualizzazione

La virtualizzazione permette di eseguire più sistemi operativi e applicazioni contemporaneamente, indipendentemente dall'hardware sottostante. I vantaggi includono l'isolamento dei fallimenti, l'introduzione di nuove capacità senza aggiungere complessità e il miglioramento delle prestazioni complessive.

semplice da installare e overhead

* **Process Virtual Machine**: usata solo per un singolo processo che viene compilato nel suo codice macchina e poi emulato (faccio finta che sia windows) oppure interpretato (java) in base alle necessità.
* **Virtual Machine Monitor**: capace di fornire VM a tanti tipi di programmi diversi, usando una sola CPU. Usato dagli SD, in quanto ogni programma è isolato e gli errori non si propagano.

### MicroServizi



L'architettura a microservizi è uno stile di progettazione del software (simile a [SOA](https://docs.google.com/document/u/2/d/1ruSroQeQ3snfko6p_1AJ1igje-men5yjP1AtSlNqZ-c/edit))

* **Servizi Indipendenti**: Ogni micro servizio è autonomo rispetto agli altri. Ognuno dei quali esegue una funzione specifica
* **Distribuzione Decentralizzata**: I microservizi possono essere distribuiti su diverse piattaforme e infrastrutture. Ogni team può scegliere la tecnologia e gli strumenti più adatti per il proprio microservizi.
* **Scalabilità**: Se un particolare servizio ha un picco di richieste, può essere scalato senza dover scalare l'intera applicazione.
* **Resilienza**: Grazie al loro isolamento, un errore in un micro servizio non influisce necessariamente sugli altri servizi.
* **Contenitori**: I container forniscono un ambiente isolato e replicabile per l'esecuzione dei microservizi, ovvero ogni micro servizio corrisponde ad una VM.

**Vantaggi dei Microservizi**

* Agilità: Facilitano lo sviluppo e la distribuzione rapida di nuove funzionalità.
* Flessibilità: Permettono l'uso di diverse tecnologie e linguaggi di programmazione per diversi servizi.
* Manutenibilità, Scalabilità, Resilienza

**Svantaggi dei Microservizi**

* Gestire molti servizi indipendenti richiede una gestione avanzata del deployment, del monitoraggio e della comunicazione.
* La comunicazione tra i servizi può introdurre latenze
* Mantenere la consistenza dei dati è complicato

**Esempio di Utilizzo**

Applicazione di e-commerce. Invece di avere un'unica applicazione monolitica (composta da Adapter, business logic, etc) l'applicazione può essere suddivisa in microservizi distinti come gestione degli ordini, inventario, pagamento, catalogo prodotti, e autenticazione.

### Containers

Un container è un'unità standard di software che racchiude il codice e tutte le sue dipendenze, permettendo all'applicazione di funzionare in modo rapido e affidabile da un ambiente all'altro. I container forniscono un ambiente consistente per l'esecuzione del software.

Vengono usati per creare servizi PaaS e SaaS.

* **Isolamento**: persino il file system è isolato ed unico per ogni container. Eseguono diverse applicazioni sullo stesso host senza conflitti.
* **Portabilità**: I container includono tutto ciò che è necessario per eseguire un'applicazione, come codice, runtime, librerie e configurazioni di sistema, rendendo l'applicazione portatile tra diversi ambienti, sia locali che cloud.
* **Consistenza**: Forniscono un ambiente di esecuzione coerente
* **Efficienza**: Condividono il kernel del SO sottostante, rendendoli più leggeri e meno esigenti in termini di risorse rispetto alle VM. Questo consente di eseguire molti più container sulla stessa infrastruttura.
* **Velocità**: I container possono essere avviati e arrestati più rapidamente delle VM, permettendo scalabilità dinamica.

Vantaggi dei container = distribuzione di un applicazione su più nodi della rete e unità standard di deployment.

Il container si occupa della comunicazione con il web server.

**Svantaggi dei Container**

* **Sicurezza**: Richiedono una gestione attenta della sicurezza, specialmente in ambienti multi-tenant.
* **Persistenza dei Dati**: La gestione dei dati persistenti può essere complicata, poiché i container sono progettati per essere effimeri.
* **Complessità Operativa**: La gestione di molti container può diventare complessa, richiedendo strumenti di orchestrazione come Kubernetes.

**Esempio di Utilizzo**

Prendiamo lo sviluppo di applicazioni microservice. Ogni micro servizio può essere eseguito in un container separato, consentendo di sviluppare, testare e distribuire ogni micro servizio indipendentemente dagli altri.

