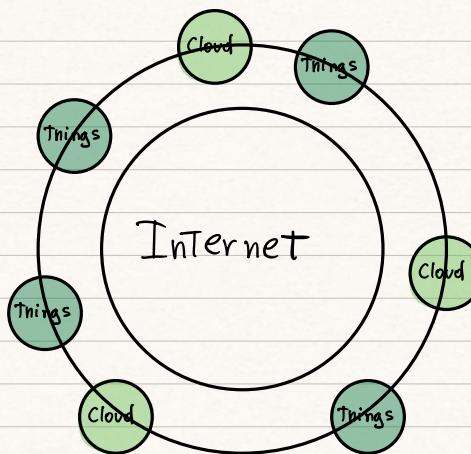
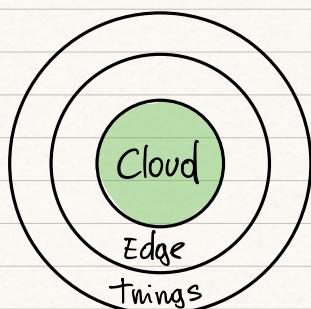


Quando parliamo di computazione pervasiva parliamo di una computazione che si inserisce in tutti gli ambiti della nostra vita (ad esempio con sensori, attuatori) che ci consentono di essere presenti nel mondo computazionale.

I sistemi odierni sono organizzati nel seguente modo:



Il cloud è il posto dove avvengono le decisioni, inoltre è qui che abbiamo i server, e quindi "l'intelligenza", dove la maggior parte delle computazioni avviene.

Quando parliamo di **edge** parliamo degli **edge del nostro cloud**, dove i nostri oggetti si collegano ed interagiscono. Questo ci costringe a portare i dati all'interno del cloud. Dopo che questi dati vengono elaborati, vengono nuovamente riportati in "periferia" (edge) dalle "things".

**Perché trasferiscono i dati?**

Il posto dove vengono prodotti e consumati i dati, è diverso da dove i dati vengono processati. Se andiamo a vedere dove i dati vengono effettivamente elaborati, noteremo che l'unico posto utile è **solo il Cloud**. Oggi giorno, se dobbiamo processare un gran numero di dati, ci rivolgiamo ad un provider che ci fornisce dei server ai quali darremo **trasferire i nostri dati**.

La banda necessaria per trasferire tutti i dati necessari, è **estremamente grande**, e quindi **NON scalabile**.

## Computazione Pervasiva

Concepiamo la C.P. come un ambiente in cui abbiano una capacità di calcolo "pervasiva" ovvero così diffusa nell'ambiente tanto da essere non più visibile.

E' fondamentale, per la C.P. supportare gli aspetti di mobilità. Tutto ciò che riguarda la C.P. assume ed ingloba la mobile computing.

Quando (storicamente) si è passati dai sistemi isolati a sistemi connessi, si sono iniziati ad affrontare tutta la serie dei problemi dei sistemi distribuiti. Si è quindi dovuto sviluppare un gran numero di meccanismi per risolvere i vari problemi introdotti con le innovazioni.

Dopo i sistemi distribuiti altri problemi sono sorti con l'avvento dei **sistemi mobili** come ad esempio il dover gestire il frequente cambio della rete. Un altro punto importante è sicuramente la gestione delle limitate risorse, sia energetiche che computazionali.

Come far sì che l'enorme mole di sensori ed attuatori venga usata in maniera efficace ed efficiente?

**Scalabilità localizzata**: in questi contesti possono avere meccanismi di scalabilità locale che possono essere usati per realizzare applicazioni molto complesse.

**Riuscire a gestire la diversità**: proprio perché abbiano degli ambienti contenenti dispositivi completamente diversi tra loro, dobbiamo fare in modo che tutto funzioni in modo corretto, e che l'utente abbia un'esperienza "senza intoppi".

## Uso efficiente degli Smart Spaces

Possiamo concepire uno Smart Space come uno spazio con un'infrastruttura di computazione; l'aspetto interessante è che in uno smart space vengono fusi due elementi che finora erano stati completamente disgiunti: **Spazio fisico** e **Spazio Virtuale** (della computazione). In uno Smart space questi due elementi vengono fusi insieme.

**Esempio:** Regolazione automatica di temperatura in una stanza. Questo "problema" può essere gestito grazie ai diversi sensori di Umidità, Temperatura, etc. In un sistema di questo tipo, quando il sistema rileva che nella stanza vi sono esseri umani, e sussistono alcune condizioni (e.g. luminosità) vi è una regolazione automatica di temperatura. La rilevazione di umani, deve essere fatta in modo da rendere il tutto trasparente.

## Invisibilità

L'invisibilità è un aspetto fondamentale che non è ancora stato totalmente risolto, proprio perché richiede dei livelli di integrazione che non sono ancora disponibili. Finché siamo capaci di integrare ad un livello di astrazione adeguato sensori-attuatori, app mobile e logica computazionale che siano capaci di "creare" fa sì che le persone non debbano modificare il loro comportamento per adeguarsi ai sistemi computazionali. Questo è la vera sfida della computazione pervasiva.

Quindi, se i sistemi sono "invisibili" è solo perché si interfacciano al nostro stesso livello.

AM 00:27 Lez 1

## Scalabilità localizzata

I problemi di scalabilità locale riguardano quanto siano in grado di gestire i problemi in modo da avere dei sistemi in grado di gestire sempre un maggior numero di interazioni e di comportamenti.

Nell'ambito dell'avanzamento e ricerca per quanto riguarda la scalabilità, molto spesso i sistemi precedenti hanno ignorato il concetto di **Distanza Fisica**: I server, solitamente non memorizzano molte informazioni riguardo ai client, al massimo delle info di posizione geografica per capire quale server DNS controllare. I server hanno quindi un unico obiettivo: rispondere a quante più richieste possibile nel minor tempo possibile; la distanza fisica viene ignorata.

Nella C.P. Il "mondo fisico" e "mondo computazionale" sono strettamente collegati; di conseguenza quando si interagisce con sistemi di questo tipo è importante tenere conto della posizione dei veri "oggetti" che compongono la C.P.

La maggior parte delle reti, nelle C.P., le interazioni sono in **Locale**, o nel nostro "ricinato".

E' quindi opportuno sfruttare la capacità computazionale locale.

## Mascherare la diversità

Proprio perché ci troviamo a lavorare con sistemi estremamente vari, è necessario attuare una serie di azioni che permettano di mantenere invariate l'esperienza d'uso. Potremmo dover passare da un sistema in cui la capacità computazionale è "alta" ad uno dove invece è molto scarsa. Dobbiamo quindi attivare azioni che possono sopprimere le determinate menzogne.

### Uniformità

Questo obiettivo non è ancora stato ottenuto, ed è l'obiettivo dei prossimi sistemi.

**Perché è importante?** L'uniformità è importante perché questo aspetto può impattare sulla vita delle persone; l'obiettivo è quindi diminuire la quantità di variazioni che l'utente deve subire. Se ad esempio sono su un **sistema desktop** e sto effettuando delle operazioni, e mi sposto su un **sistema mobile**, non potrò effettuare le stesse operazioni per un problema legato alla non uniformità.

**Esempio di sistema uniforme:** facciamo finta che mi trovo in un ufficio con copertura internet e sto usando un programma di posta. Se mi sposto in un edificio che non dispone di copertura internet, il sistema potrebbe scaricare preventivamente la posta in modo da potermi permettere di leggere la posta anche senza connessione ad internet.

**Quindi** Vedremo come costruire un'app mobile che consuma e produce dati e che si interfaccia con uno o più servizi in cloud e consente di mantenere e produrre informazioni anche tra più dispositivi.

Per ogni "problema" visto precedentemente sceglieremo una tecnologia/protocollo che ci consenta di comprendere la soluzione ad uno specifico tramite dei **Pattern** ben precisi.

### Modalità di esame

Progetto sviluppo applicazione Android