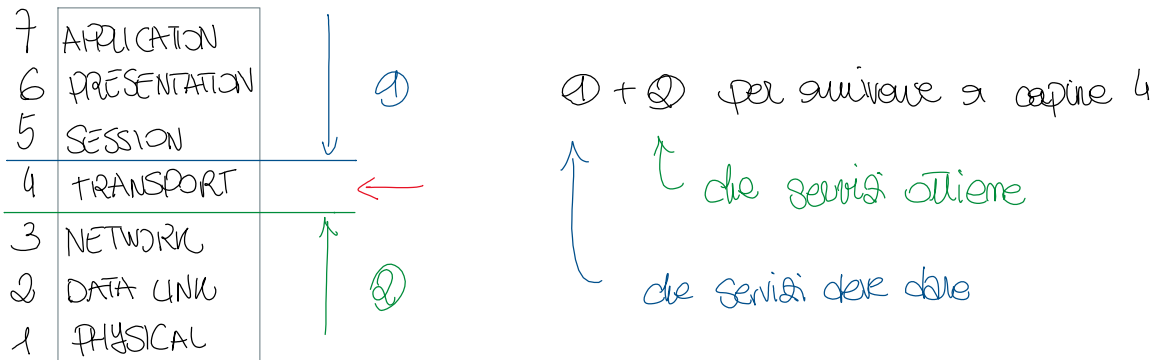


Dati, controlli e management

↳ Arrivano in continuazione

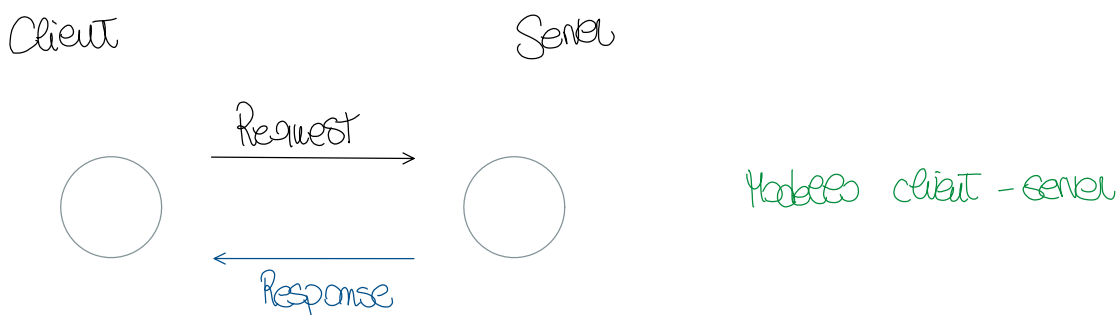
Management: in caso di configurazione o modifica apparati e lettura di variabili di sistema.



LIBRO: Reti di Calcolatori e Internet (un approccio Top down)

APPLICATION:

Modello applicazioni di Rete:

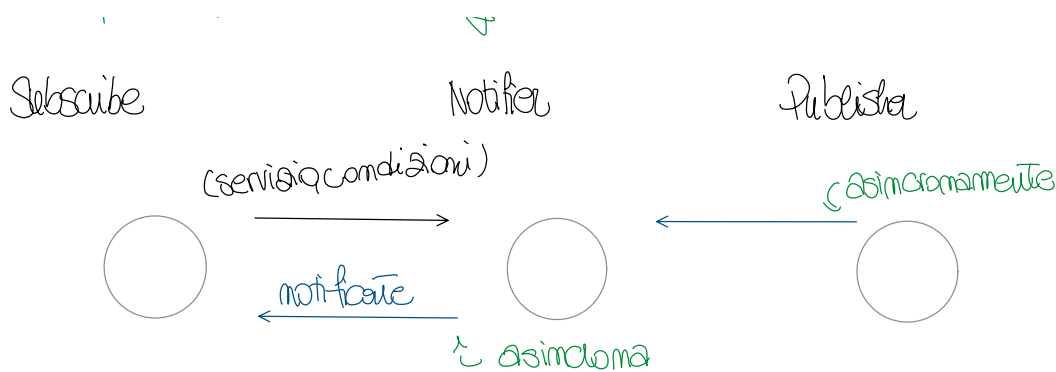


Generalmente posti su host diversi  
 ↳ macchine

Il server deve rispondere subito (dopo l'elaborazione)

L'iniziativa (di mandare un dato) è del client sempre.

Modello publish, subscribe, notify.



Notifier e Publisher spesso collassate in un'unica entità

Subscribe  $\approx$  client perché fa una richiesta ma rimane in attesa finché non gli notificano che ciò che ha chiesto è disponibile

Si può emulare con client-server facendo il **polling**: chiedere spesso i server finché non ha la risposta.  
 è un impegno che ha un costo

Ci sarebbe anche il **modello peer to peer**: vincente perché i dati vengono da diversi client invece che da un server.

Utile per liberare la banda del server, è divisa tra i client.

## LINGUAGGIO C:

Il C disciplina sia l'output del programma che l'uso della memoria.

```
int main()
```

```
{
```

```
char c;
```

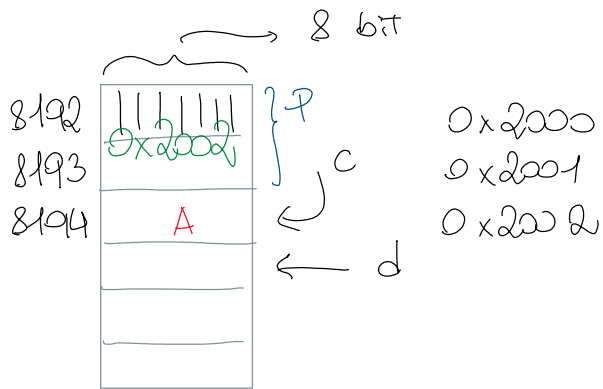
```
c = 'A';
```

```
printf("%d", c);
```

```
}
```

↳ stringa per il formato per visualizzare il dato

$$2^{10} = 1024 \quad 2^{13} = 8192$$



2 | 0 | 0 | 0  
10 | 0000 | 0000 | 0000

Siccome A è un carattere, occorre una transcodifica (ASCII)

A : 0100 | 0001

0x41 65 decimale

%d = decimale, il programma stampa 65

```
int main() {
    char c, d;
    c = 'A';
    printf("%d", c);
    printf("%x", &c);
}
```

decimale  
esadecimale

Il programma stampa l'esadecimale

che è l'indirizzo della cella di memoria di c.

& : operatore prefisso unario per

→ 16 bit in questo caso per contenere un ind

p è puntatore a c : contiene l'indirizzo di c

Quindi

printf("%c", \*p) Stampa A, il contenuto di  
↳ carattere

Se facis :

\*p = 'B';

'B'

`printf("%d", c);` stampa 66

ho modificato `c` senza farlo esplicitamente.

Però, se faccio:

`p = p + 1;`

`*p = 'E';`

`printf("%c", d);` stampa E, inserito in `d` (non era nemmeno inizializzato)

`int c` 32 bit = 8 byte

`long int c` 32 bit

`long long c` 64 bit