

PARADIGMA PUBLISH & SUBSCRIBE

→ Pattern o Modalità con la quale si interagisce in un'applicazione

Cattura un fenomeno fisico, Prenderlo lì e Poi Portare l'Elaborazione Altrove, dove c'è più Potenza di Calcolo

CLIENT-SERVER Nasce a Livello Applicativo

Protocolli Applicativi per capire come ci parliamo, uno possibile è CLIENT-SERVER, che è MOLTO RIGIDO

→ Ordine TEMPORALE Ben Stabilito visto che Client fa il Primo Passo e Server il SECONDO.

Mentre MODELLO PUB/SUB (che sta prendendo piede nell'IOT)

Ci Sono 3 Ruoli Definiti (3 TIPI di "ATTORI")

■ **PUBLISH** → Possono Essere +, pubblica DATO Sul BROKER
Pubblicano copie <DATO, TOPIC>
Etichettato con TOPIC
Semantica del dato ↗

SubscriveR \Rightarrow Si ABBONA al BROKER, Intecagisce:

① Abbonandosi ad un TOPIC (Mostra Interesse per Quel Topic, ma non c'è detta che ci Sia Già il DATO)

Quando Qualche PUBLISHER Pubblicherà un DATO Con Quel TOPIC Verranno NOTIFICATI i SUBSCRIBER

BROKER duplicherà il DATO a TUTTI i Subscriber Interessati (SORTA DI MULTICAST)

BROKER \Rightarrow Deve Essere Unico

È una BACHECA DEGLI ANNUNCI, c'è chi ATTACCA e chi Legge e soprattutto non Si CONOSCONO DIRETTAMENTE

Dato un CLIENT-SERVER posso Portarlo su PUB-SUB con dei GUADAGNI:

Host Processo SERVER deve ESSERE SEMPRE ACCESO (Cicli ∞) ma deve anche avere un IP PUBBLICO, perché non posso Metterlo come i CLIENT dietro ad un NAT.

Si CREA anche una VULNERABILITÀ dato che è un PROCESSO in Attesa con una PORTA APERTA

Sposta la Complessità nel BROKER, che è GENERALISTA

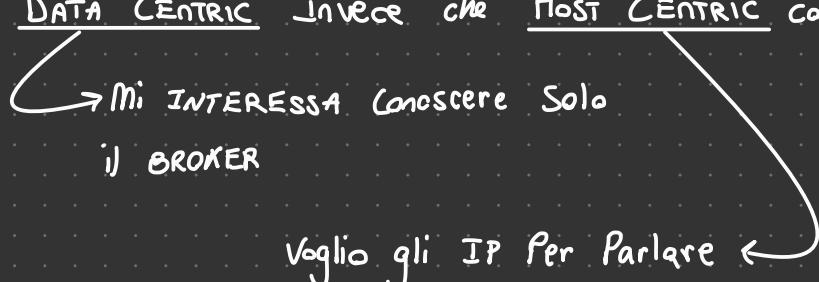
ESEMPIO DI ESAME \Rightarrow Trasformare CLIENT-SERVER to PUB-SUB

È un ARCHITETTURA multi a multi e non multi a Uno
PUB-SUB CLIENT-SERVER

Aggiungere un DISPLAY ad Esempio da un SENSORE con PUB-SUB
devo Solamente Aggiungere un SUBSCRIBER Mentre con Client/Srv
Avrei dovuto RISCRIVERE il Server.

PUB-SUB è più SCABILE ed ESTENDIBILE

PUB-SUB è DATA CENTRIC Invece che HOST CENTRIC come C/S

Mi INTERESSA Conoscere Solo
il BROKER

Voglio gli IP Per Parlare

A livello di REALIZZAZIONI di PUB-SUB Abbiamo: (PROTOCOLLI)

- # MQTT Mosquitto (c/c++) o Paho (Python)
- # AMQP RabbitMQ
- # Kafka

E poi ci Sono le diverse IMPLEMENTAZIONI

Display può Essere SOLO un SUBSCRIBER, Ci Sono Altri Casi
dove un dispositivo può essere Sia SUBSCRIBER che PUBLISHER

Ad Esempio chi RAFFINA dei DATI ←
Per PRODURNE Altri (MEDIA,...)

MQTT :

Message Queuing Telemetry Transport

Posso Avere PUBLISHER e SUBSCRIBER Implementati con Linguaggi
diversi ma a NOI Interessa però il PROTOCOLLO con il quale
Si Parlano

Posso Aggiungere Tra MQTT e TCP un Layer di TLS/SSL per
Renderlo sicuro

Non ha DIRITTI Sui TOPIC. (Chiunque può leggere tutto, a patto che
riesca a Collegarsi)

Username e PSW Servono per Autenticare i "CLIENT" ma con Kafka
ad Esempio ci Sono La Gestione degli UTENTI

