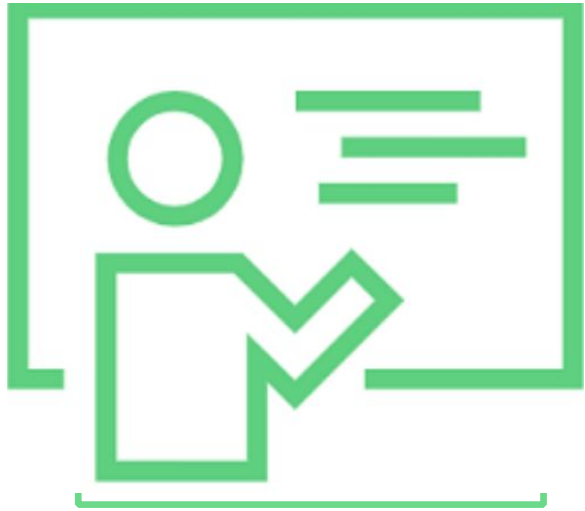




Nätverk & Systemintegration Kursintroduktion

Joakim Englund, Systemmentor AB

Översikt

- Nätverk - mest repetition av datorkommunikation + säkerhet
- Systemintegration - väldigt tl;dr så blir det mycket API

Tentamen

Kort sagt kommer den täcka kunskapsmålen från kursplanen
(KUN27-KUN32)

Inlämningsuppgift

- Kommer täcka kompetens- och färdighetskraven från kursplanen (KOM19-KOM20 & F28-F33)
- Läggs ut snart, den färdiga ligger på Linux partitionen som jag inte kommer åt just nu

Dagens mål

- Repetition och slå bort lite ringrost från sommarlovet
- Förstå rollen för TCP/IP, UDP, MQTT och LoRa i IoT
- Förstå IP-adress, port, broker, publisher och subscriber
- Bygga ett första MQTT-flöde med JSON-data
- Dokumentera topologi och projekt i Git

Kursens röda tråd

- Nätverk + MQTT
- ↓
- API + JSON/XML
- ↓
- Systemintegration
- ↓
- Säkerhet + övervakning
- ↓
- Felsökning + dokumentation

Dagens IoT-kedja

- Sensor eller simulator (Producent)
↓ publicerar
 - MQTT-broker
↓ förmedlar
 - Mottagare/subscriber (Konsument)
-
- Sensor/simulator = datakälla
 - Broker = nav för meddelanden
 - Subscriber = tar emot data

TCP/IP: hur hittar data fram?

- **IP-adress**: identifierar en enhet i nätverket
- **TCP**: tillförlitlig, anslutningsorienterad transport
- **Port**: identifierar tjänsten på en IP-adress

MQTT kör normalt över *TCP/IP*

Klient: TCP, port 1883 -> Broker

TCP och UDP

<i>TCP</i>	<i>UDP</i>
Anslutningsorienterat	Anslutningslöst
Tillförlitlig leverans	Låg overhead
Ordning och kvittens	Ingen garanti för leverans/ordning
Vanligt för MQTT	Kan passa tidskritiska datagram

MQTT i tre roller

<i>Roll</i>	<i>Gör</i>
Publisher	Producerar meddelanden
Broker	Tar emot och förmedlar meddelanden
Subscriber	Konsumerar meddelanden

Topics: adressen för MQTT-meddelanden

Under kursen när ni jobbar två och två tycker jag att ni bör hitta på något namn för eran grupp så kan vi alla köra på samma bas och stil på våra topics: **iot25/<grupp>/sensor/<typ av sensor><nummer om ni har flera>**

Ex: **iot25/main-characters/sensor/temperatur3** för tredje temp-sensorn för gruppen 'main-characters'

- Topics är hierarkiska namn
- Subscriber väljer vilket topic den vill följa
- `#` kan användas som wildcard i en prenumeration

Ex: **iot25/#** för att subscriba på alla sub-topics under 'iot25'

Leverans: QoS och retained message

- QoS 0: högst en leverans, ingen kvittens
- QoS 1: minst en leverans, kan ge dubletter, mer trafik, kräver mer prestanda
- QoS 2: exakt en leverans, ännu högre trafik, kräver ännu mer prestanda
- Retained: brokern sparar senaste värdet för nya prenumeranter

Payload: själva datat/meddelandet

```
{  
  "sensorId": "temp-01",  
  "value": 21.7,  
  "unit": "C"  
}
```

- Payload: meddelandets innehåll
- JSON: strukturerat och lättläst dataformat
- Samma fältnamn och betydelse ska användas konsekvent

LoRa i en IoT-arkitektur

Sensor - via LoRa -> Gateway - via IP/MQTT -> (MoIn)tjänster

- LoRa: lång räckvidd och låg energiförbrukning
 - Passar små mängder data från batteridrivna enheter
- MQTT: applikationsprotokoll för meddelanden

LoRa och MQTT kan förekomma i samma lösning, men har olika roller

Dagens laboration

1. Skapa Git-repository och README
2. Dokumentera topologi, broker, port och topic
3. Starta en subscriber
4. Publicera ett JSON-meddelande
5. Visa att meddelandet kommer fram
6. Dokumentera ett testat fel och gör en commit

Demonstration

1. Subscriber lyssnar på ett topic
2. Publisher skickar JSON
3. Brokern förmedlar meddelandet
4. Subscriber visar topic + payload

Guide för mosquitto på Windows: [Mosquitto Guide](#)