1. **MQTT Bridge (Server Side)**

Start → Initialise MQTT Client → Connect to Broker

↓ Subscribe to Topics(Meter Data, Events, Tamper Alerts)

↓(Message Received?)

Yes → Forward to DLMS Parser/Store in DB

No→ Discard & Log Error

No→ Keep Listening

↓(Connection Lost?)

Yes → Reconnect Procedure

No → Continue Operation

1. **DLMS Request Handling (Sever Side)**

Start → Application / User Requests Data from Meter

↓Build DLMS APDU (Get/Set/Action/Read/Write)

↓Apply Security (Authentication Token + Encryption)

↓ Send APDU via MQTT to Meter

↓Wait for Response (ACK/Data/Error)

↓ (Response Valid?)

Yes → Decrypt & Parse DLMS Respinse

→ Update DB/Notify Client

No → Retry or Mark as Faled

**3. Security Authentication / Encryption & Integrity (Server Side)**  
Start → Connection Request to Meter  
   ↓ Initiate Authentication (HLS: Challenge-Response)  
   ↓ Generate Session Key (AES/HMAC)  
   ↓ Encrypt DLMS APDU  
   ↓ Send to Meter via MQTT  
   ↓ Receive Response → Decrypt & Verify Integrity  
       ↓ (Auth Passed?)  
          Yes → Session Active  
          No  → Reject / Retry / Raise Alert  
  
  
**4. Data Storage & Processing (Additional – Server Side)**  
Start → Receive Valid Meter Data  
   ↓ Parse DLMS Objects (OBIS Codes: Voltage, Energy, Tamper Events)  
   ↓ Store in Database (Time-Series / SQL / NoSQL)  
   ↓ Trigger Business Logic:  
       - Billing Engine  
       - Load Forecasting  
       - Demand Response  
   ↓ Archive Historical Data  
  
  
**5. Analytics & Reporting (Additional – Server Side)**  
Start → Daily/Hourly Scheduler Trigger  
   ↓ Fetch Data from DB (e.g., last 45 days)  
   ↓ Run Analytics (Consumption Patterns, Grid Import/Export, Voltage Issues)  
   ↓ Generate Reports (PDF/Excel/JSON)  
   ↓ Send to Stakeholders via Email/MQTT/Portal  
  
  
**6. Event & Alert Handling (Additional – Server Side)**  
Start → Receive Event Message (Tamper, Power Failure, Over Voltage)  
   ↓ Decrypt & Verify  
   ↓ Log Event into DB  
   ↓ Notify Stakeholders (SMS/Email/Mobile App)  
   ↓ If Critical → Trigger Control Action (e.g., Remote Disconnect)