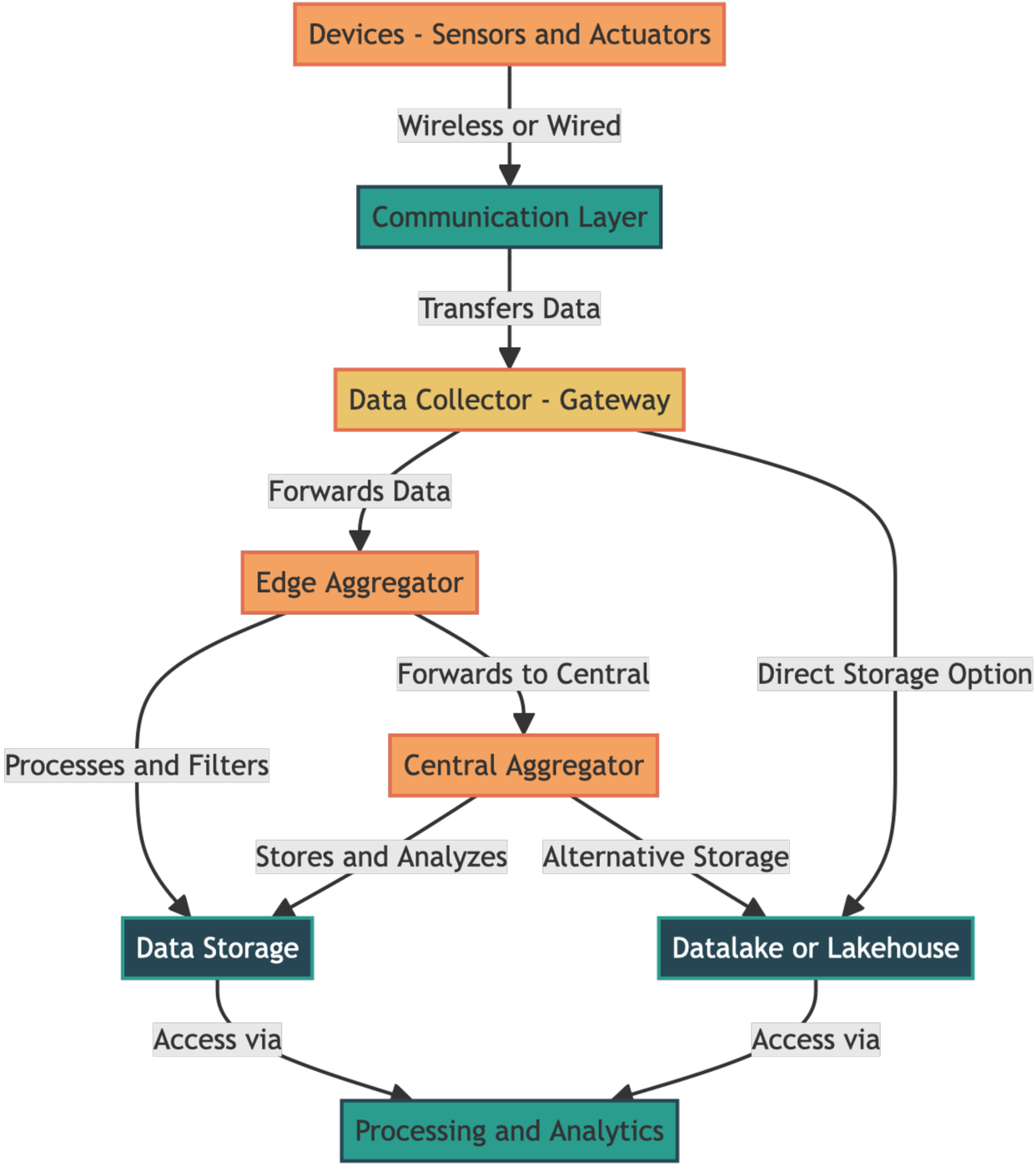
IoT-arkitektur och datakommunikation

Sammanfattning 2025-03-17

Mikael Wallin - mikael.x.wallin@gmail.com


Devices - Sensors and Actuators
Wireless or Wired
Communication Layer
Transfers Data
Data Collector - Gateway
Forwards Data
Edge Aggregator
Forwards to Central
Direct Storage Option
Processes and Filters
Central Aggregator
Stores and Analyzes
Alternative Storage
Data Storage
Datalake or Lakehouse
Access via
Access via
Processing and Analytics

1. **Describe how communication from a web browser to a server and back works, covering all layers from ARP to port opening.**
    - Explain the journey of a request from a browser to a server, including DNS resolution, TCP/IP, ARP, HTTP, and response handling.

2. **Compare IoT architectural demands with real-time system (RTS) requirements.**
    - Discuss how constraints like timing, reliability, and computational requirements differ between IoT and RTS, and how they impact system design.

3. **Describe the most important data storage options for IoT.**
    - Compare relational databases, time-series databases, object storage, and distributed storage in terms of scalability, performance, and use cases.

4. **Describe the challenges of data aggregation in IoT systems, from the device level to edge computing and central cloud storage.**
    - Discuss trade-offs in processing power, latency, bandwidth usage, and decision-making when handling data at different points in an IoT architecture.

5. **Explain the role and impact of different communication protocols in IoT networks.**
    - Compare MQTT, CoAP, HTTP, and WebSockets in terms of efficiency, scalability, and suitability for real-time communication.

6. **Analyze security challenges in IoT networks and propose mitigation strategies.**
    - Identify risks such as data breaches, device spoofing, and denial-of-service attacks, and suggest best practices for encryption, authentication, and secure communication.

https://github.com/miwashi-edu/edu-auth-flask