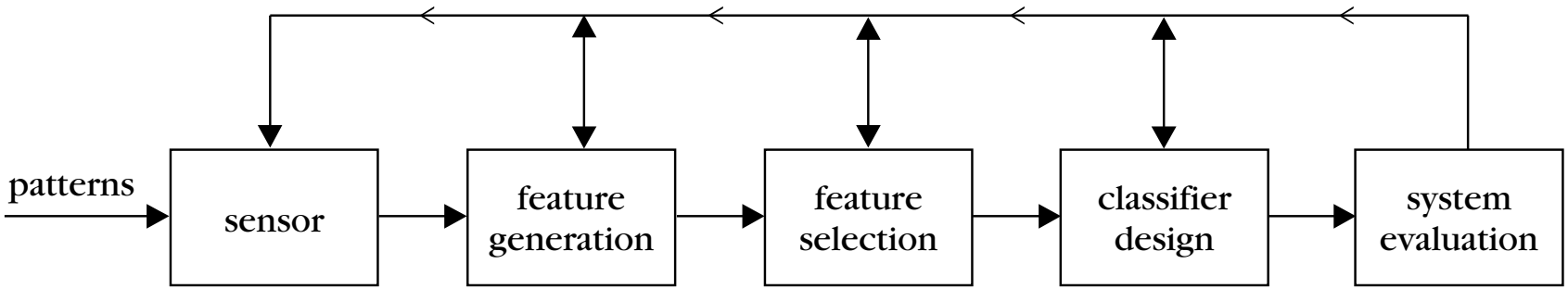
Software Architecture for Sensor-based Human Monitoring

(Scenario: Activity Monitoring)

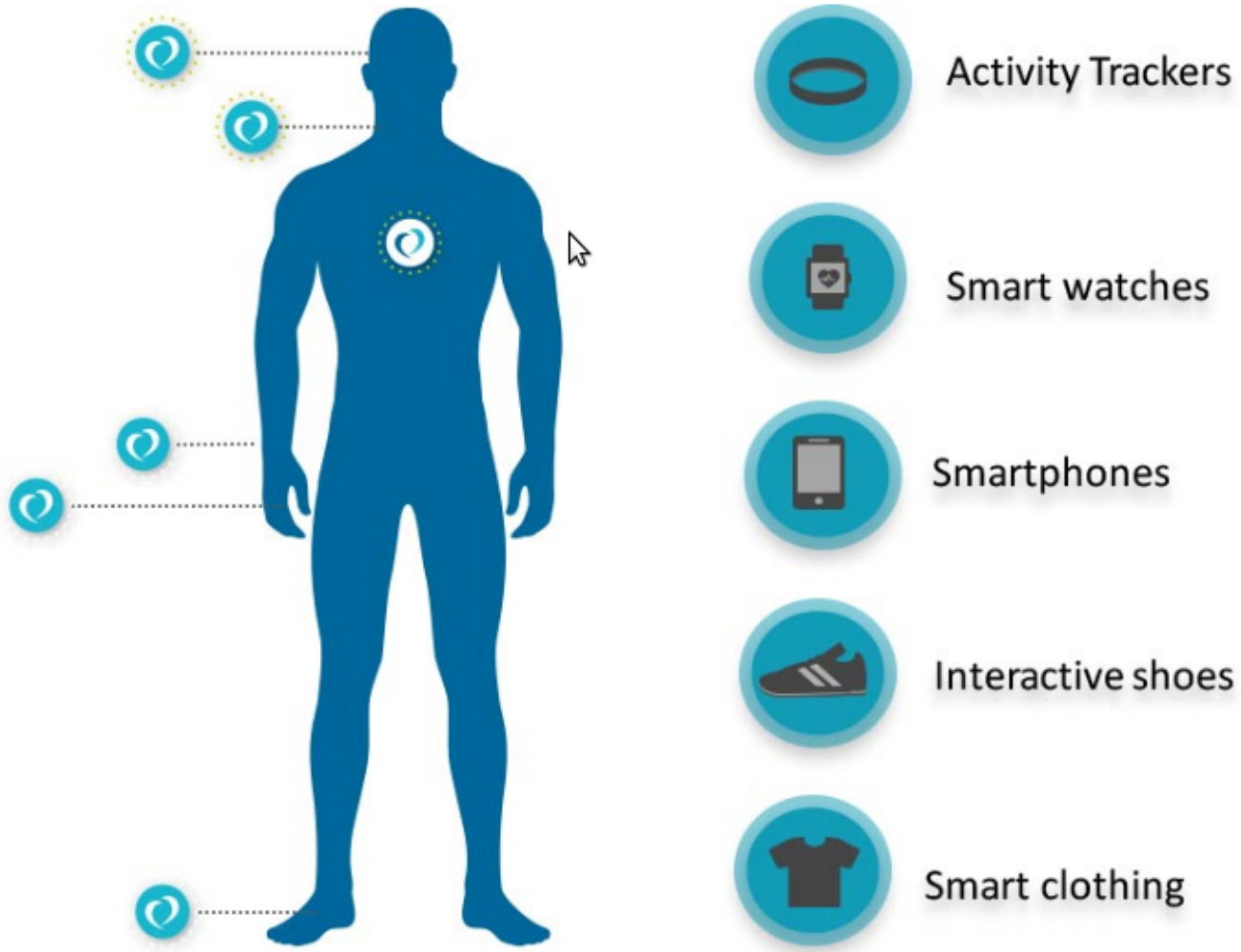
Dr. Muhammad Adeel Nisar
(Courtesy: Prof. Dr. Marcin Grzegorzek)

Department of Information Technology
University of the Punjab, Lahore

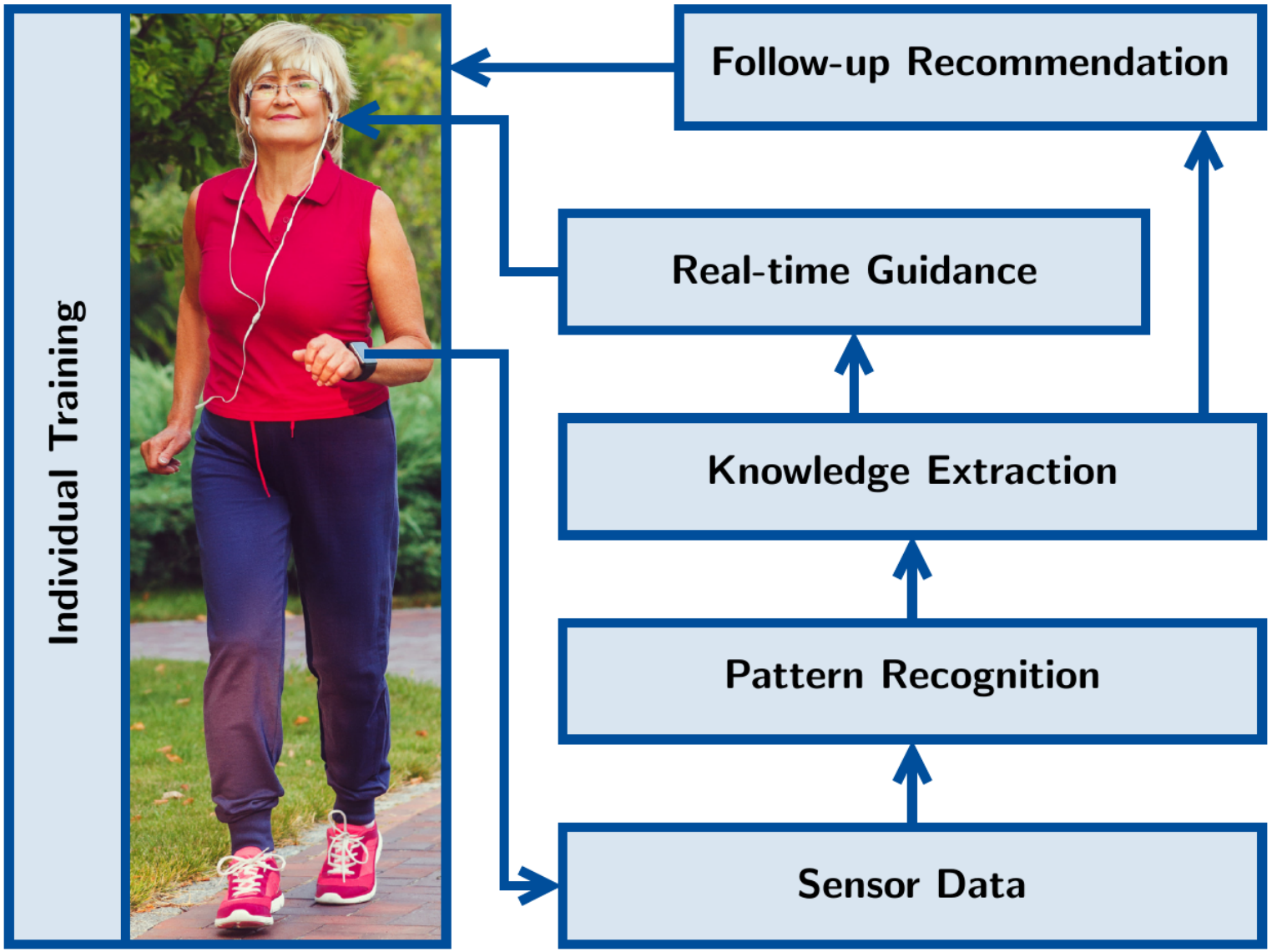
Basic Stages of Pattern Analysis



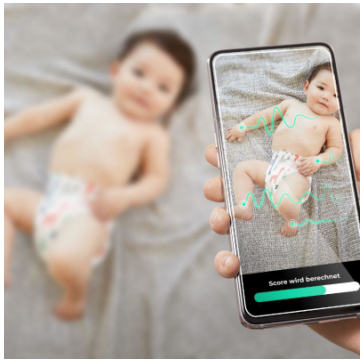
Human Observed by Sensors

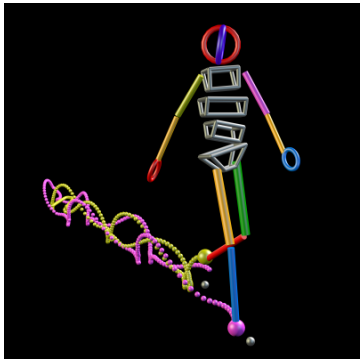


Real-time Guidance and Individual Training

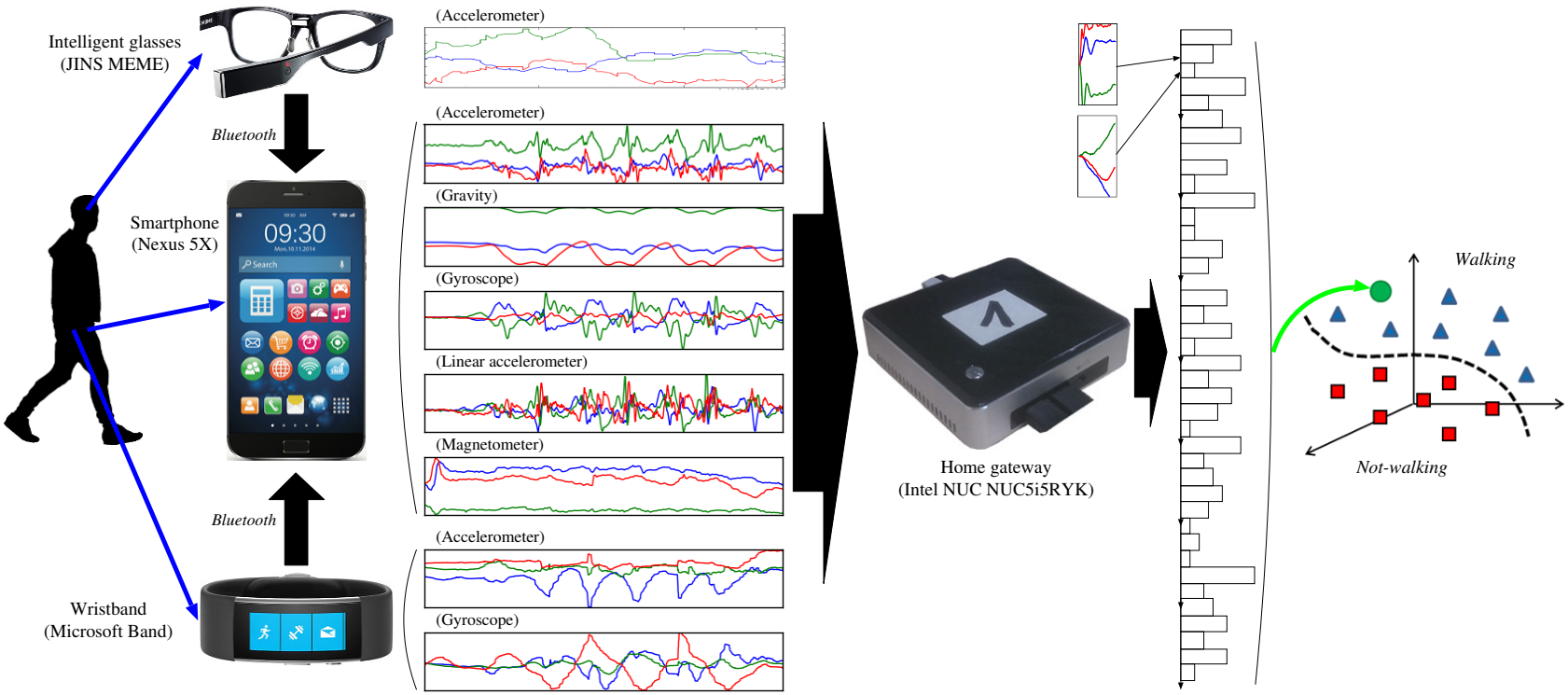


DS4AHT – Applications



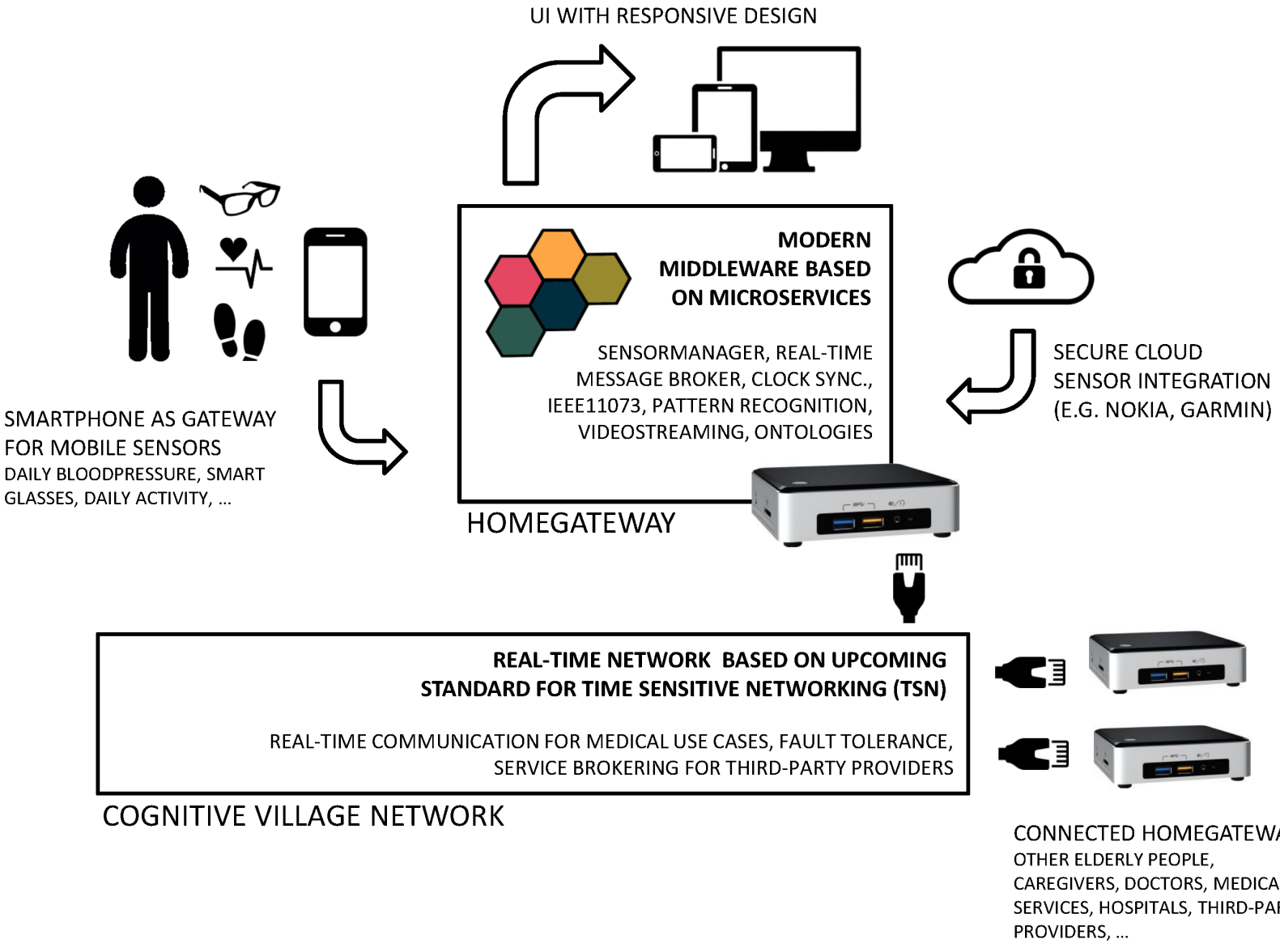


Activity Recognition Platform – Data Flow



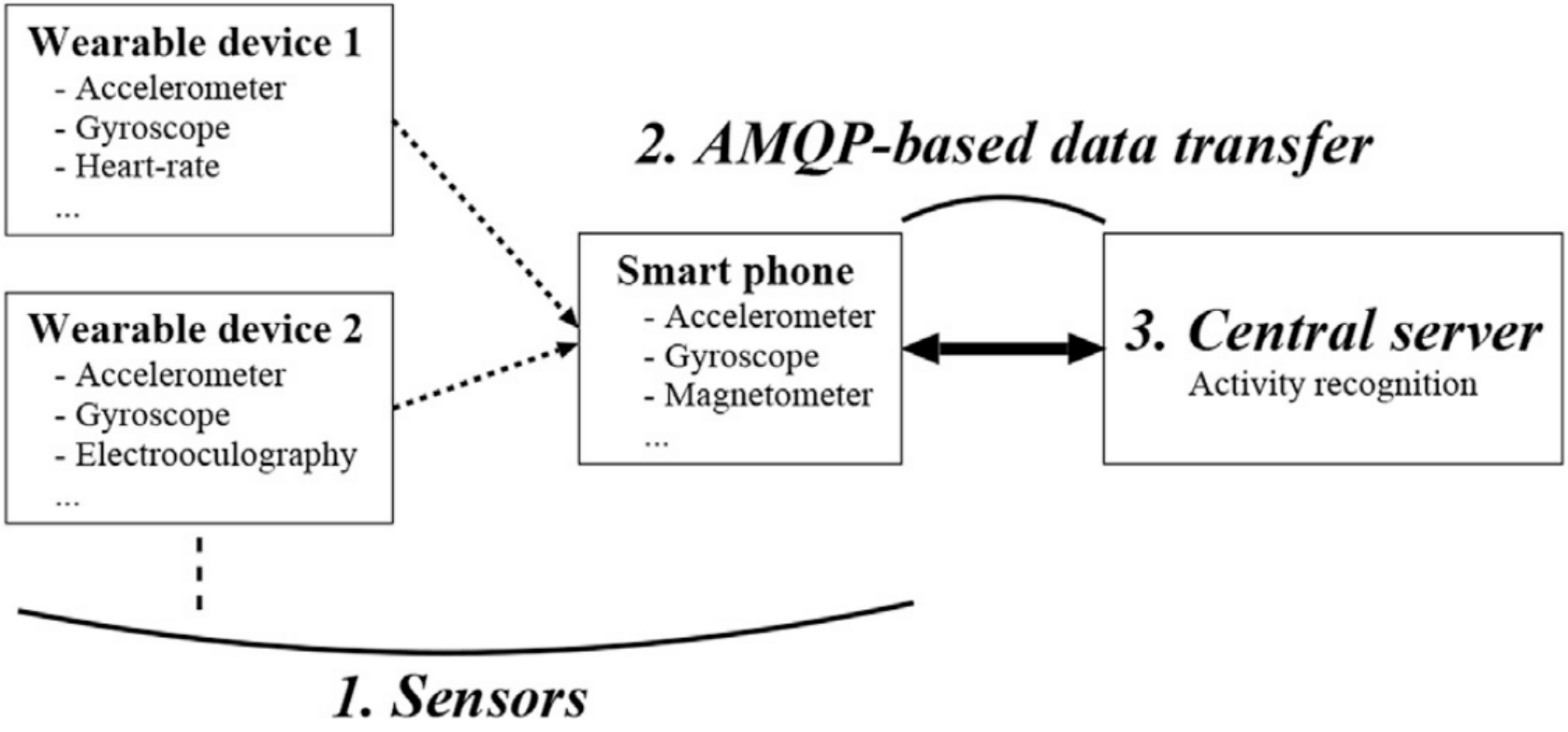
https://www.youtube.com/watch?v=J6WaO7jFOtU
https://www.youtube.com/watch?v=73qRIVP3kfU

Activity Recognition Platform – Technical Components

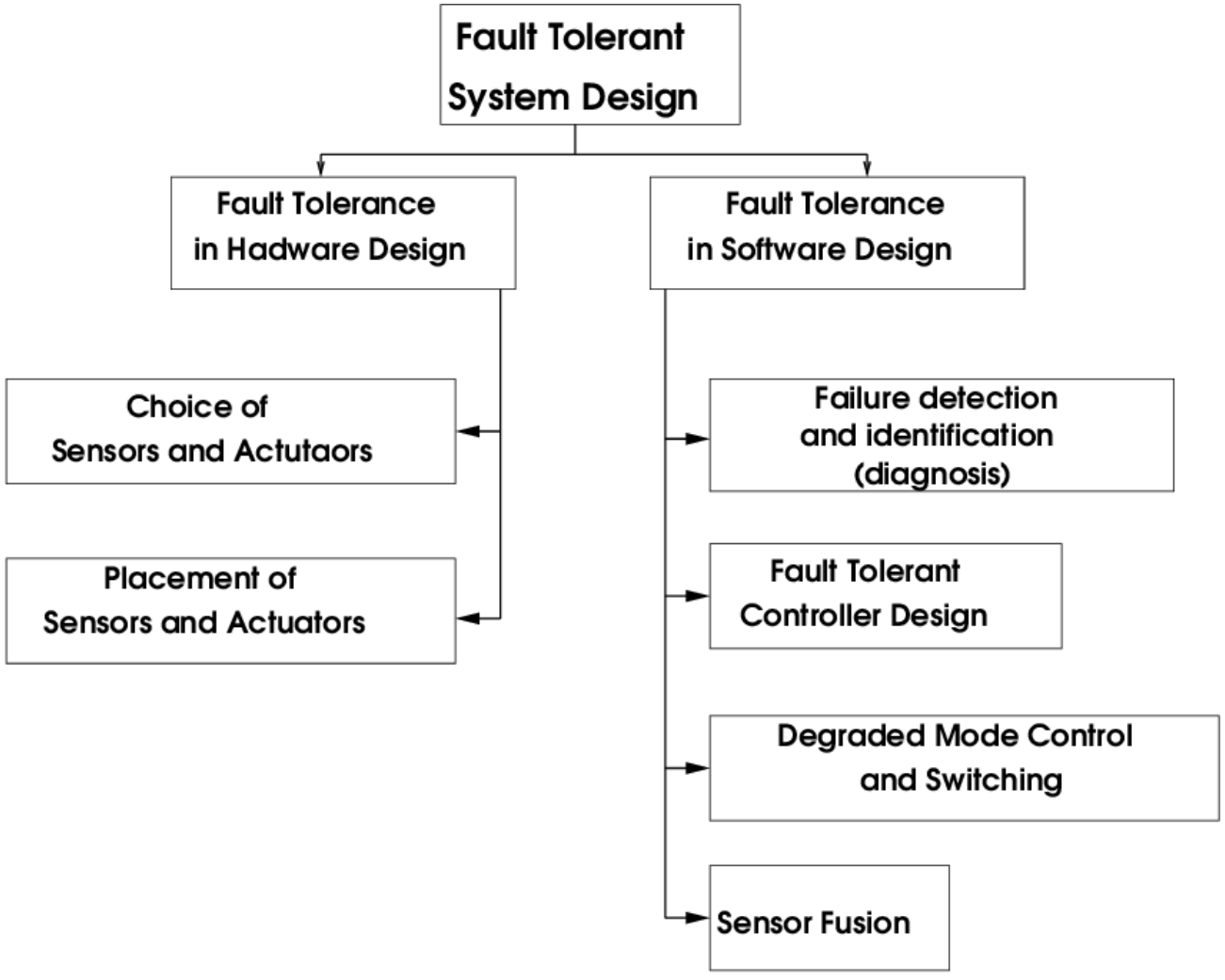


Activity Recognition Platform – Simplified Architecture

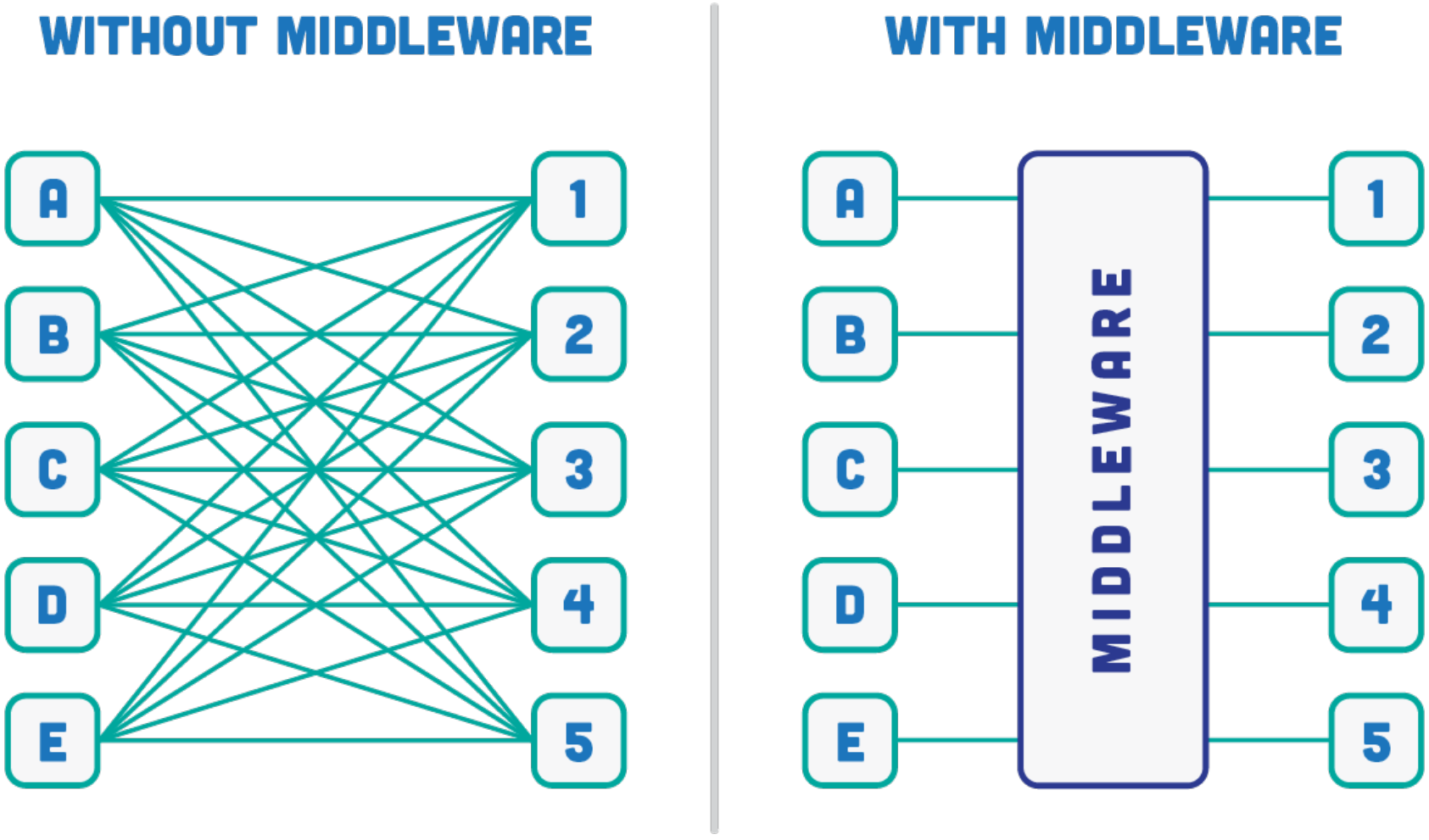
Advanced Message Queuing Protocol (AMQP)



Activity Recognition Platform – Fault-tolerant Design

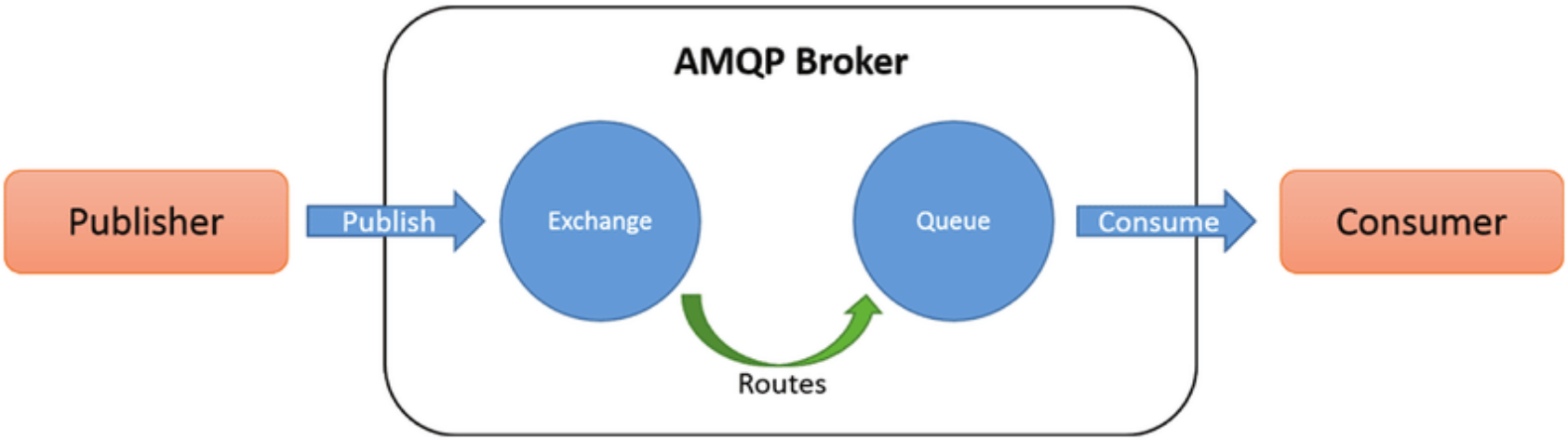


Middleware – Overall Concept



Message Broker – General Idea

Advanced Message Queuing Protocol



Message Broker Realised with RabbitMQ

RabbitMQ by Pivotal

Features | Get Started | Support

RabbitMQ Tutorials

These tutorials cover the basics of creating messaging applications using RabbitMQ. You need to have the RabbitM
tutorials, please see the installation guide or use the Docker image. Code examples of these tutorials are open so

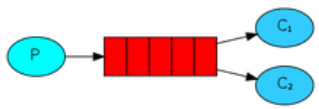
1 "Hello World!"

The simplest thing that does *something*

- Python
- Java
- Ruby

2 Work queues

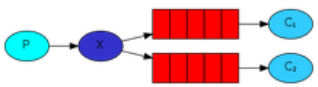
Distributing tasks among workers (the competing consumers pattern)



- Python

3 Publish/Subscribe

Sending messages to many consumers at once



- Python
- Java

https://www.rabbitmq.com/getstarted.html
https://www.youtube.com/watch?v=deG25y_r6OY

Final Statements

- Multiple sensors need to be integrated into a consistent and **reliable platform** in order to be efficiently used for human monitoring.
- Real-time capability, fault-tolerant design, openness with regard to integrating new sensors are **desirable properties** of such a platform.