RAIS

Real-time Analytics for Internet of Sports

Marie Curie European Training Network

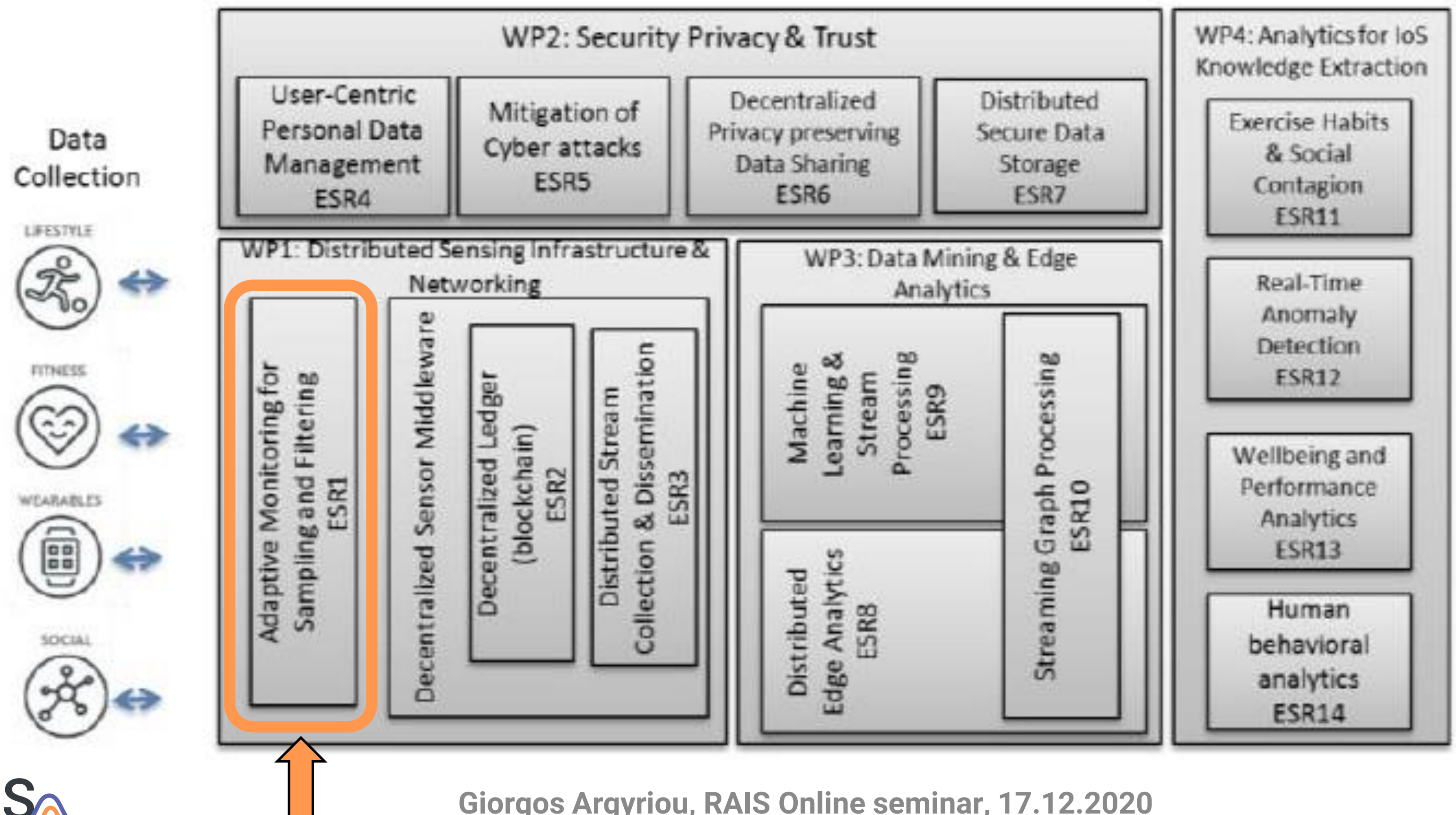
LOWERING DATA DISSEMINATION ON THE EDGE OF THE NETWORK

Giorgos Argyriou, University of Cyprus (UCy)

ESR1: Adaptive monitoring framework for Wearable Devices

University of Cyprus

LInC



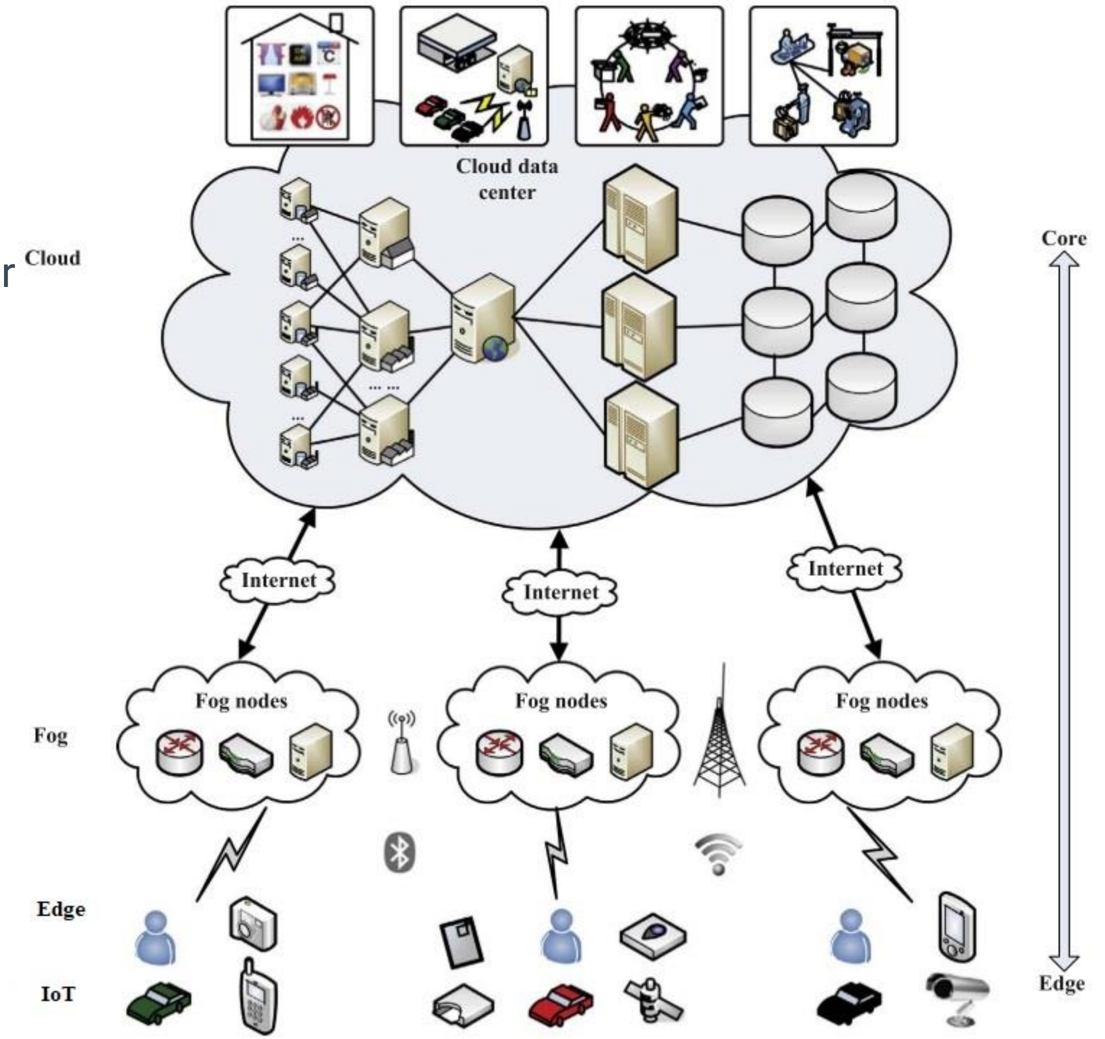
Cloud - Fog - Edge - IoT

Key characteristics

- Cloud
  - Since ~2005
  - Unlimited processing power
  - Unlimited storage
  - Thousand devices

- IoT
  - Since 1999, supply chain management
  - Healthcare, home, transport…
  - Billion devices

- Edge/Fog
  - Base stations, routers, smartphones, etc



IoT (1)

- Since 1999, supply chain management
- Healthcare, home, transport…
- Billion devices the last 5 years

2008: 6 Technologies with potential Impacts in the US [1]

"Internet of Things is the general idea of things, especially everyday objects, that are readable, recognizable, locatable, addressable, and controllable via the Internet - whether via RFID, wireless LAN, wide-area network, or other means"

2012: Singularity Summit in San Francisco [2]

"It is estimated that already 5% of human-constructed objects had embedded microprocessors"

[1] *National Intelligence Council. Disruptive Technologies Global Trends 2025. Six Technologies with Potential Impacts on US Interests Out to 2025. 2008. Available online: http://www.fas.org/irp/nic/disruptive.pdf (accessed on 14 December 2020)*

[2] *Vinge, V. Who's Afraid of First Movers? The Singularity Summit, San Francisco, CA, USA, 13–14 October 2012. Available online: https://vimeo.com/54718577 (accessed on 15 December 2020).*

IoT (2)

IoT components

- One sensor (at least)
- Microcontroller or microprocessor (usually ARM)
- Energy (plugged or battery)

Basic design factors

- **What** physical quantity is going to be monitored
- **Where** is going to be placed

    - Size
    - Shape
    - Energy source

- ❖ Wearables
- ❖ On clothes
- ❖ On electric appliances
- ❖ On transportation
- ❖ On buildings
- ❖ …

Fog or Edge? (1)

Categorization is indeed **foggy**!

Cloud → Data Center

Fog ? Edge → Routers, Smartphones, Cloudlets, Gateways

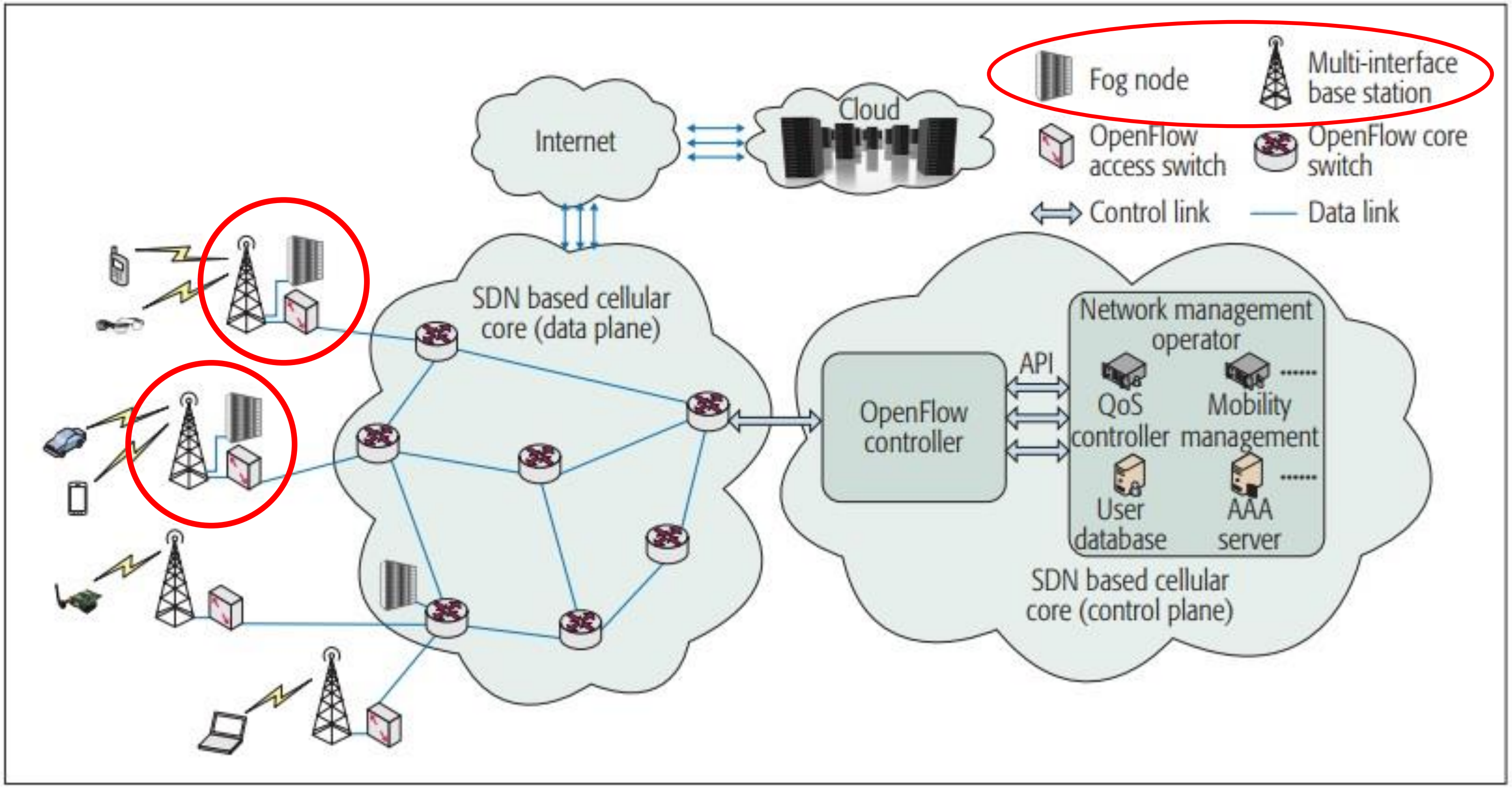
IoT → Wearables

Q: Categorization? **A**: IT DEPENDS !

Q: On what? **A**: Researchers and use cases

EdgeIoT

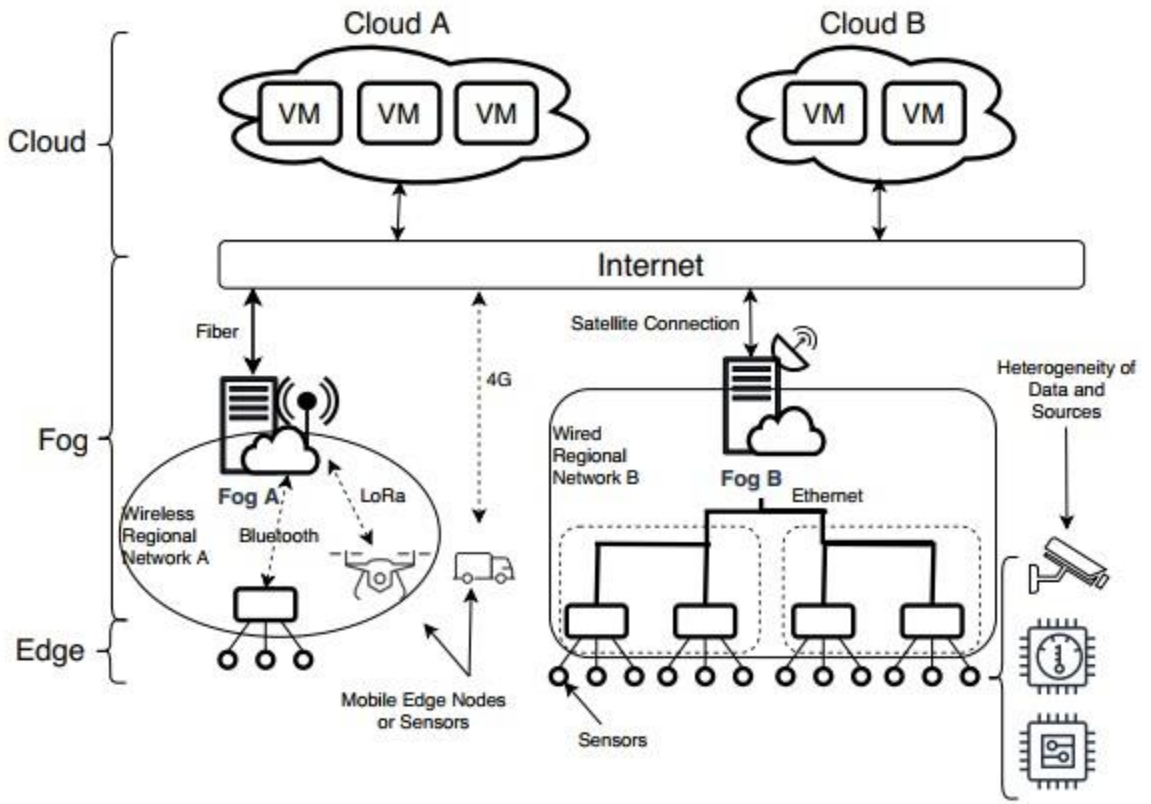
- Concept: Equip Base Stations (mobile telephony) with computing nodes
- Terminology: Every computing node (physical or virtual) is called **Fog node**



Sun, X., & Ansari, N. (2016). ***EdgeIoT: Mobile edge computing for the Internet of Things****. IEEE Communications Magazine, 54(12), 22-29.*

Fogify

- <u>Concept</u>: an emulator that eases the modeling, deployment and
- large-scale experimentation of fog and edge testbeds.
- <u>Terminology</u>: Every device with computing capabilities is a **Fog node**



Symeonides, M., Georgiou, Z., Trihinas, D., Pallis, G., & Dikaiakos, M. ***Fogify: A fog computing emulation framework****. In Proceedings of the 5th ACM/IEEE Symposium on Edge Computing, ser. SEC (Vol. 20).*

Fog or Edge? (2)

- Smartphone -> IoT? Edge? Fog?

  EdgeIoT: IoT device
  Fogify: Fog node

- Industrial IoT Gateway -> IoT? Edge? Fog?

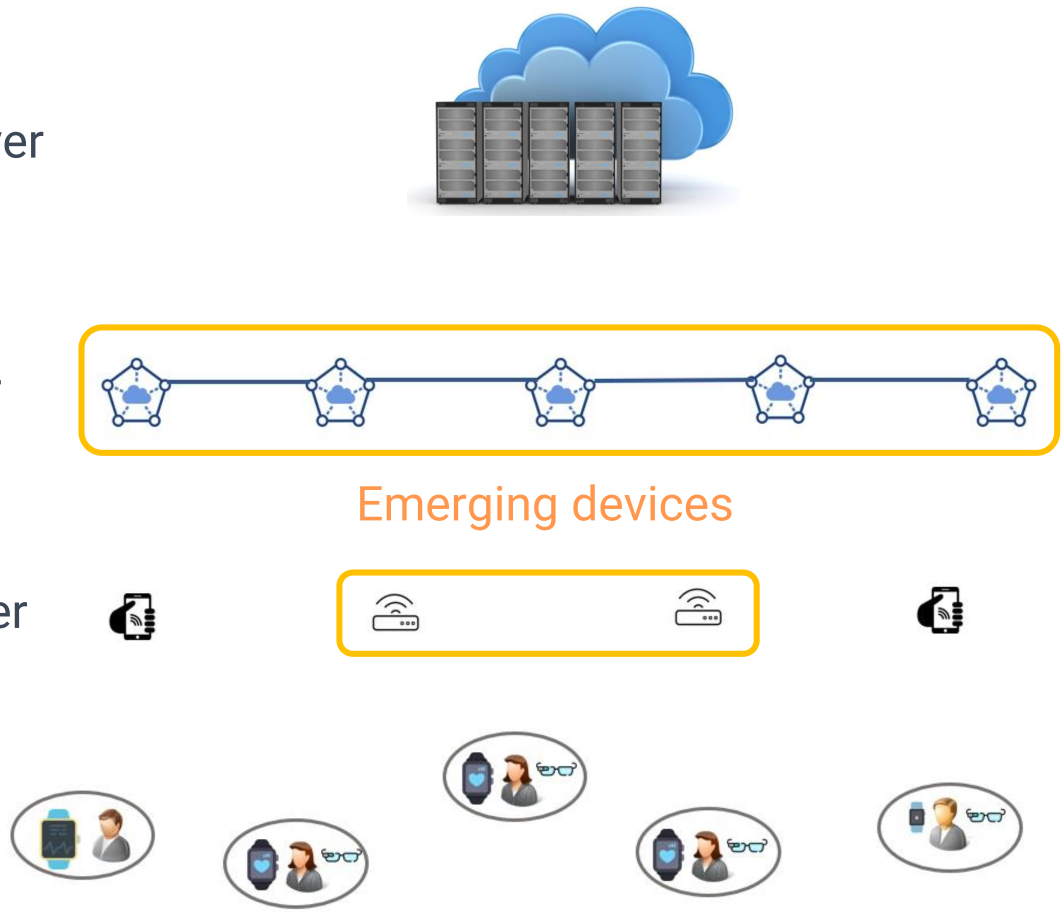
  EdgeIoT: Fog node
  Fogify: Fog node

Who's right ? Who's wrong? → NO ONE

Every researcher/author is right within in their research limits

4 layers

Cloud layer
Fog layer
Edge layer
IoT layer



- Data Centers
- Cloudlet
- Smartphones
- Enhanced routers
- Wearables

➢ Capabilities of resources
➢ How to organize
➢ Requirements

Resources on the Edge of network

Computing resources as a cloudlet (Fog layer)

- Fixed position.
- Large bandwidth (Ethernet or WiFi)
- Plugged in power network

Challenge: How to organize such infrastructure
Suggested approach: Serverless programming model

Standalone computing resources (Edge layer)

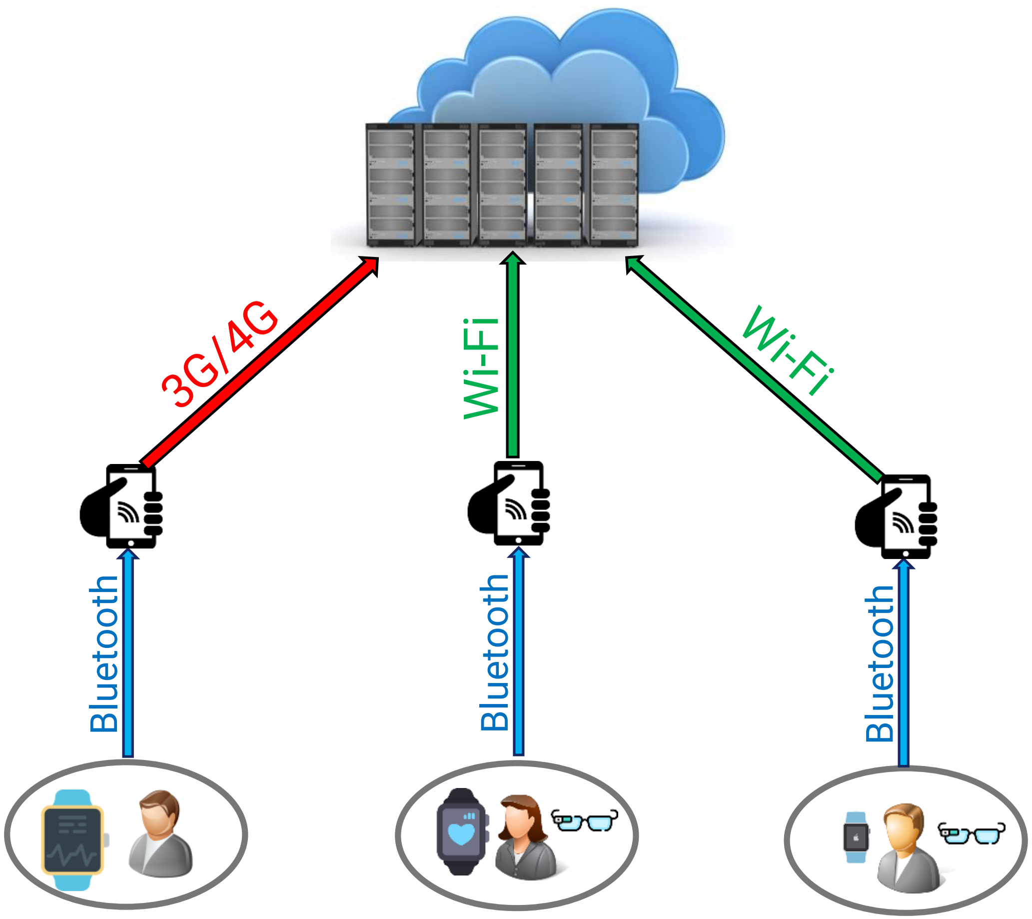
- Mobile devices
- Connectivity issues (intermittent, high cost)
- Battery powered

Challenge: How to lower power consumption
Suggested approach: Lowering data dissemination

Internet of Sports use case

3 layers
- Cloud layer
- Edge layer
- IoT layer

Data flow
- Data produced on IoT
- Data transmission to Edge
  - Wi-Fi
  - Bluetooth
- Cloud
  - Long term storage
  - Further processing



My current research

Focus: Lower power consumption on IoT devices

How they are used:

- Low capabilities
- Paired with a stronger device (on Edge layer)
- Bluetooth Low Energy (BLE) connection
- Gather data -> Unload data -> Lightweight -> Gather data

"It is estimated that 30% of ALL IoT devices are using BLE as the enabling communication protocol" *[3]*

3 main activities:

- Computation
- Sensing
- Data transmission

Energy consumption

- Data transmission is the most consuming activity
- Transmitting data over BLE costs tens of milliwatts [4]
- Computing at full power costs tens of microwatts [4]

[3] *Garcia-Espinosa, E., et al. (2018). Power Consumption Analysis of Bluetooth Low Energy Commercial Products and Their Implications for IoT Applications. Electronics, 7(12), 386.*

[4] *Blalock, D., Madden, S., & Guttag, J. (2018). Sprintz: Time series compression for the internet of things. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2(3), 1-23.*

Argument: **Lowering data dissemination will improve energy sufficiency of IoT devices**

Univariate vs Multivariate metric streams

Univariate metric streams

- Scalar physical quantities
- Sample format: (timestamp, value) = (t, v)
- Value is scalar

Sensors in wearables*

- Heart rate sensor (scalar) -> univariate metric stream : (t, v)
- Accelerometer (3-axis) -> multivariate metric stream: (t, v_1, v_2, v_3)
- Gyrometer (3-axis) -> multivariate metric stream: (t, v_1, v_2, v_3)

* GPS, thermometer, altimeter and SpO2 sensors are also used

IoT as an entity

- All streams in a multivariate one
- Number and type of sensors -> Number of total dimensions
  - e.g. 1 accelerometer + 1 heart rate sensor = 4-dimensional multivariate metric stream

System requirements

<u>Focus</u>: Exploit correlations of dimensions in a multivariate metric stream for lowering the amount of data.

System model requirements define solution form

- Storing capability
  - ➢ Solution related to data compression
- Data instantly disseminated
  - ➢ Solution that work instantly
- Processing capabilities of IoT devices
  - ➢ Run algorithm on IoT vs Split algorithm between IoT and Edge

The proposed solution should guarantee

- Less power consumption of IoT
- Data reduction will maintain a certain level of data accuracy

Data compression on univariate metric stream

Sprintz [4] is a time-series compression algorithm that achieves state-of-the-art compression ratios with minimum memory and adding virtually no latency.

- Reduce data size without sacrificing the quality
- Design requirements
  - Small block size
  - High decompression speed
  - Lossless
- Performance
  - Less than 1KB memory needed
  - Decompression > 3GB/s in single thread
- High speed forecasting algorithm
  - Temporal correlations on value and variance on one variable
  - Parallelization across different variables
- Online training
- Limitation: only on integers

[4] Blalock, D., Madden, S., & Guttag, J. (2018). **Sprintz: Time series compression for the internet of things**. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *2*(3), 1-23.

Data compression on multivariate metric stream

Energy efficient approach for IoT data collection and analysis [5]

- Fast error-bounded lossy compressor prior to transmission
- Rebuild data on edge node
- Eventually data are processed using supervised machine learning
- The SZ compression technique
  - Compression on multivariate time series
  - Controls information loss by employing an error bound technique
- Intelligent vehicle systems case
  - Vital signs data are collected
  - Monitoring driver behavior
- Evaluation
  - Data are reduced by up to 103 times
  - Quality of medical data is not affected
  - Driver stress level detection is not affected

[5] Azar, J., Makhoul, A., Barhamgi, M., & Couturier, R. (2019). **An energy efficient IoT data compression approach for edge machine learning**. *Future Generation Computer Systems*, *96*, 168-175.

Adaptive sampling on univariate metric stream

AdaM [6] is a lightweight adaptive monitoring framework for smart battery powered-devices with limited capabilities.

- Adapts monitoring intensity in place
  - Adaptive sampling
  - Adaptive filtering
- Provides one-step ahead estimation
  - Adjusts sampling rate
  - Adjusts filter range
- Identifies abrupt transient changes
- Runs on the source device

[6] *Trihinas, D., Pallis, G., & Dikaiakos, M. D. (2015, October).* ***AdaM: An adaptive monitoring framework for sampling and filtering on IoT devices.*** *In 2015 IEEE International Conference on Big Data (Big Data) (pp. 717-726). IEEE.*

Adaptive monitoring dissemination

ADMin [7] is a low-cost IoT framework that reduces on device energy consumption and the volume of data disseminated across the network

- Adapts the rate of dissemination
- Run-time knowledge of
  - Stream evolution
  - Variability
  - Seasonal behavior
- Sends update of the estimation model instead of the stream
- Takes into account the seasonality
- Runs on source device. The receiver must be tuned

[7] Trihinas, D. et al. (2017, May). ADMin: Adaptive monitoring dissemination for the Internet of Things. In *IEEE INFOCOM 2017-IEEE conference on computer communications* (pp. 1-9). IEEE.

AdaM vs ADMin

AdaM

- Each new sample -> sampling period
- Variable sampling -> adaptive sampling according to each sample
- Output: When samples are filtered the last sample that has been sent is taken into account

ADMin

- Cycle of: samples used for training -> estimation model creation -> send model
- Cycle is interrupted when there is an unexpected sample
- What is sent: training and unexpected samples + estimation model (1st degree equation)
- Output: When samples are filtered the model is taken into account

Data to be processed -> AdaM

Data to be disseminated -> ADMin

Evaluation

User-defined parameters

- Tmax upper limit of sampling period
- g acceptable imprecision of a reconstructed metric stream

$$c_i = 1 - \frac{|\hat{\sigma}_i - \sigma_i|}{\sigma_i}$$

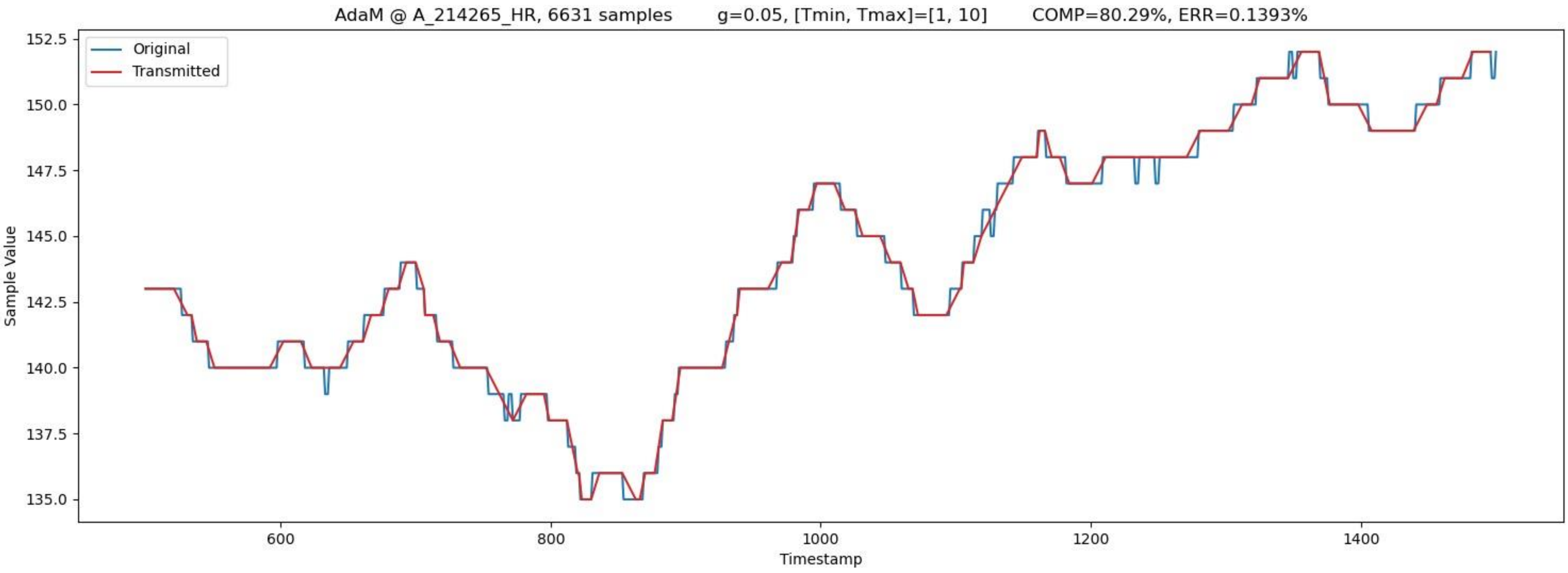
$$T_{i+1} = \begin{cases} T_i + \lambda \cdot (1 + \frac{c_i - \gamma}{c_i}), & c_i \geq 1 - \gamma \\ T_{min}, & else \end{cases}$$

Performance metrics

- AdaM's output -> compressed timeseries
- Compression = $\frac{\text{amount of transmitted samples}}{\text{amount of original samples}}$ %
- Error -> Mean Absolute Percentage Error (MAPE) %
    - Accuracy = 1 – Error %

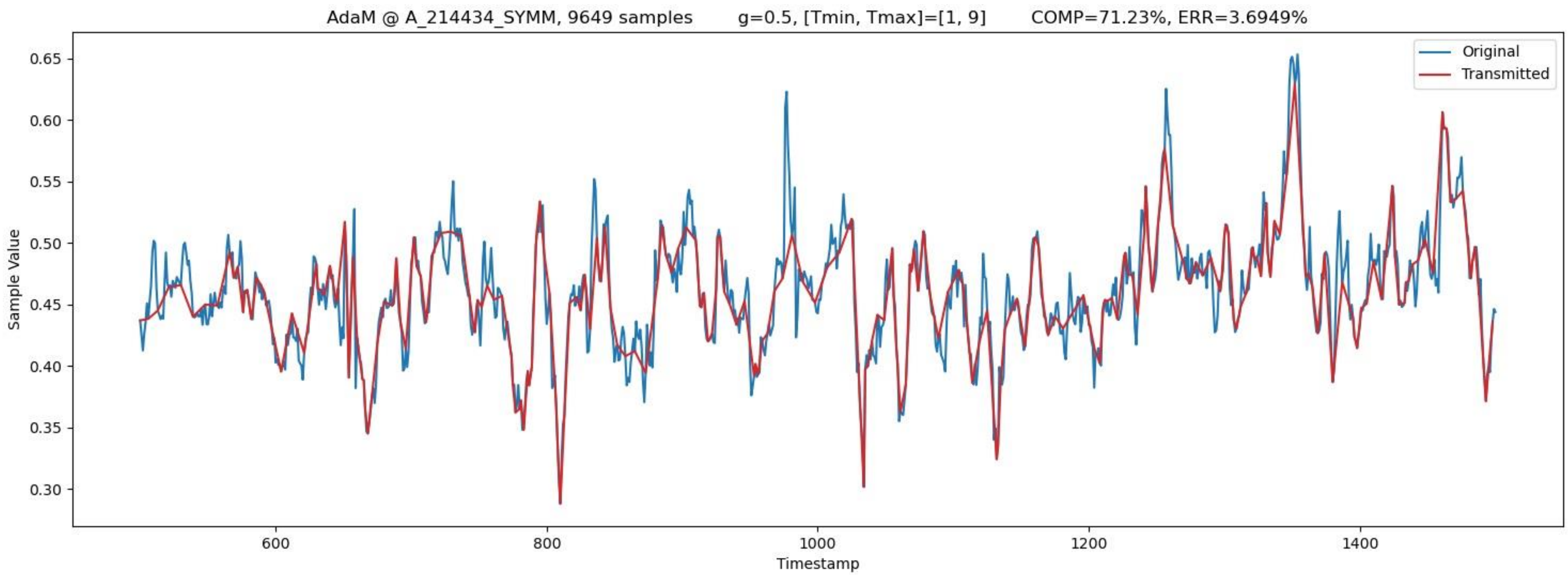
AdaM performance on Heart-Rate datasets

- Compression levels ~80%
- Error levels ~0.15%
    - *Error was never more than 0.5% even with the most extreme settings*
    - *Error is calculated using MAPE*



AdaM performance on SYMM datasets

- Compression levels ~70%
- Error levels ~3.5%
  - *Error was never more than 4% even with the most extreme settings*
  - *Error is calculated using MAPE*



THANK YOU!