Cyber-Physical Systems

Dr. Jonathan Jaramillo

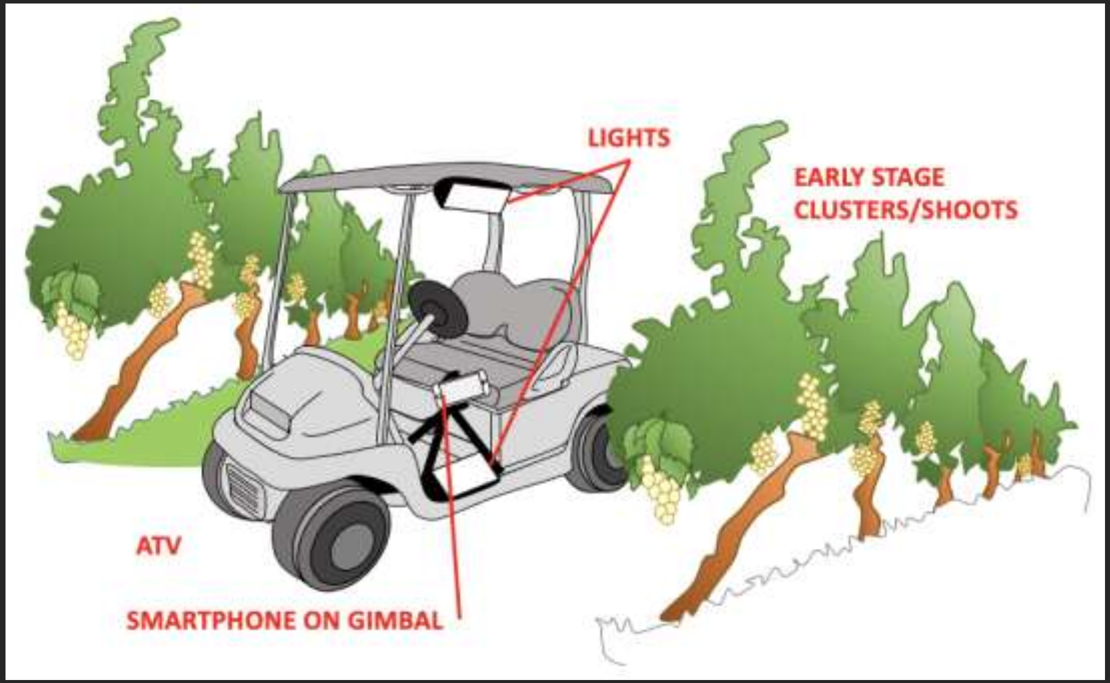
Lecture Outline

- About me
- Course Introduction
- Assignment Overview
- Module 0 – CPS Architectures and Frameworks

About Me

- From Ithaca, Ny
- BS. In Physics and Computer Science from Houghton University
- Systems Engineer at Lockheed Martin
  - Situational awareness, sensor fusion, user interface
- Half an M. Eng. Degree in systems engineering
- PhD in ECE from Cornell University
  - Computer Vision, Digital Agriculture, Robotics

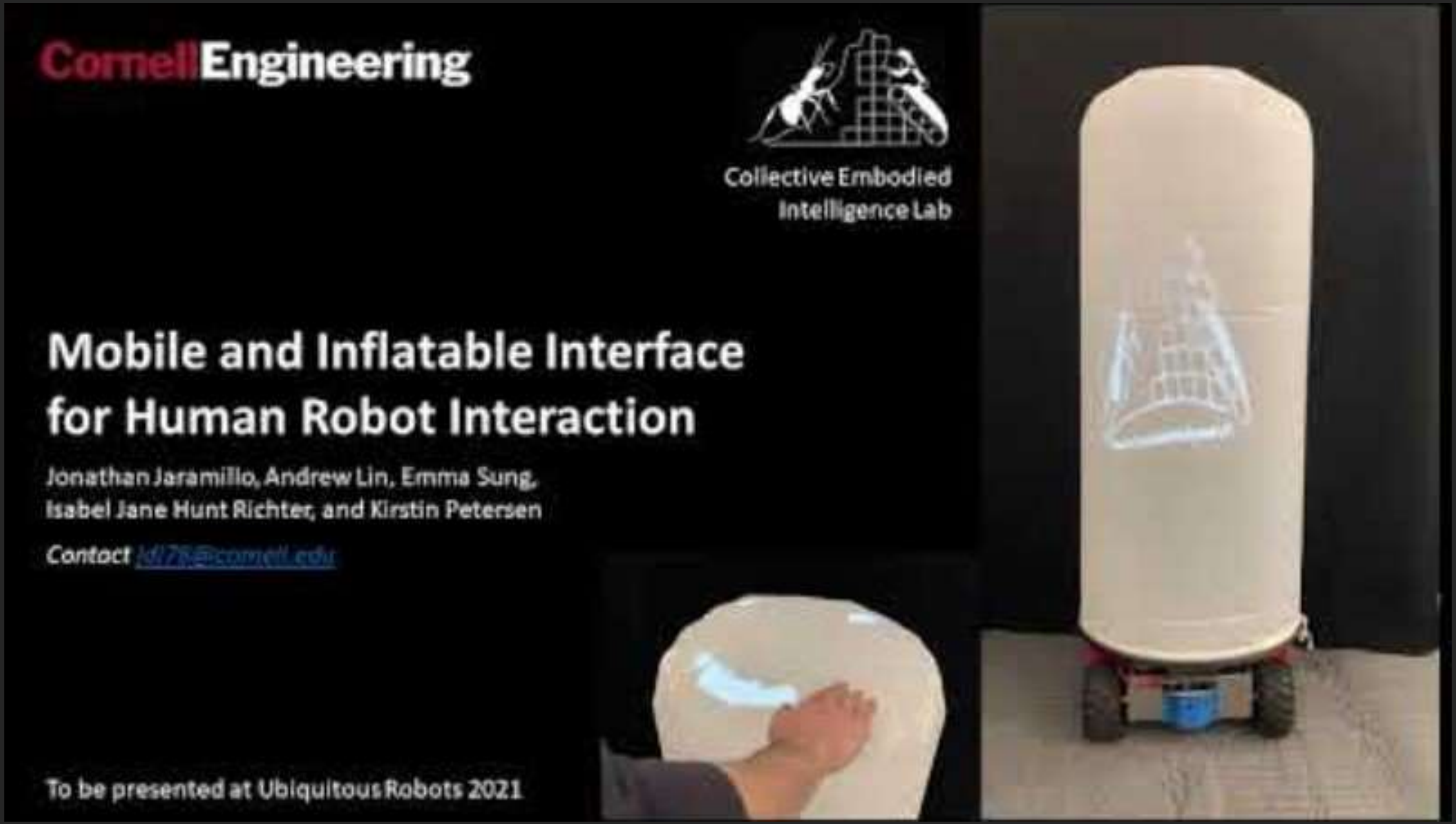
Yield Estimation in Vineyards



Pruning Weight Estimation in Vineyards

Crop Coefficient Estimation in Vineyards

Robotics

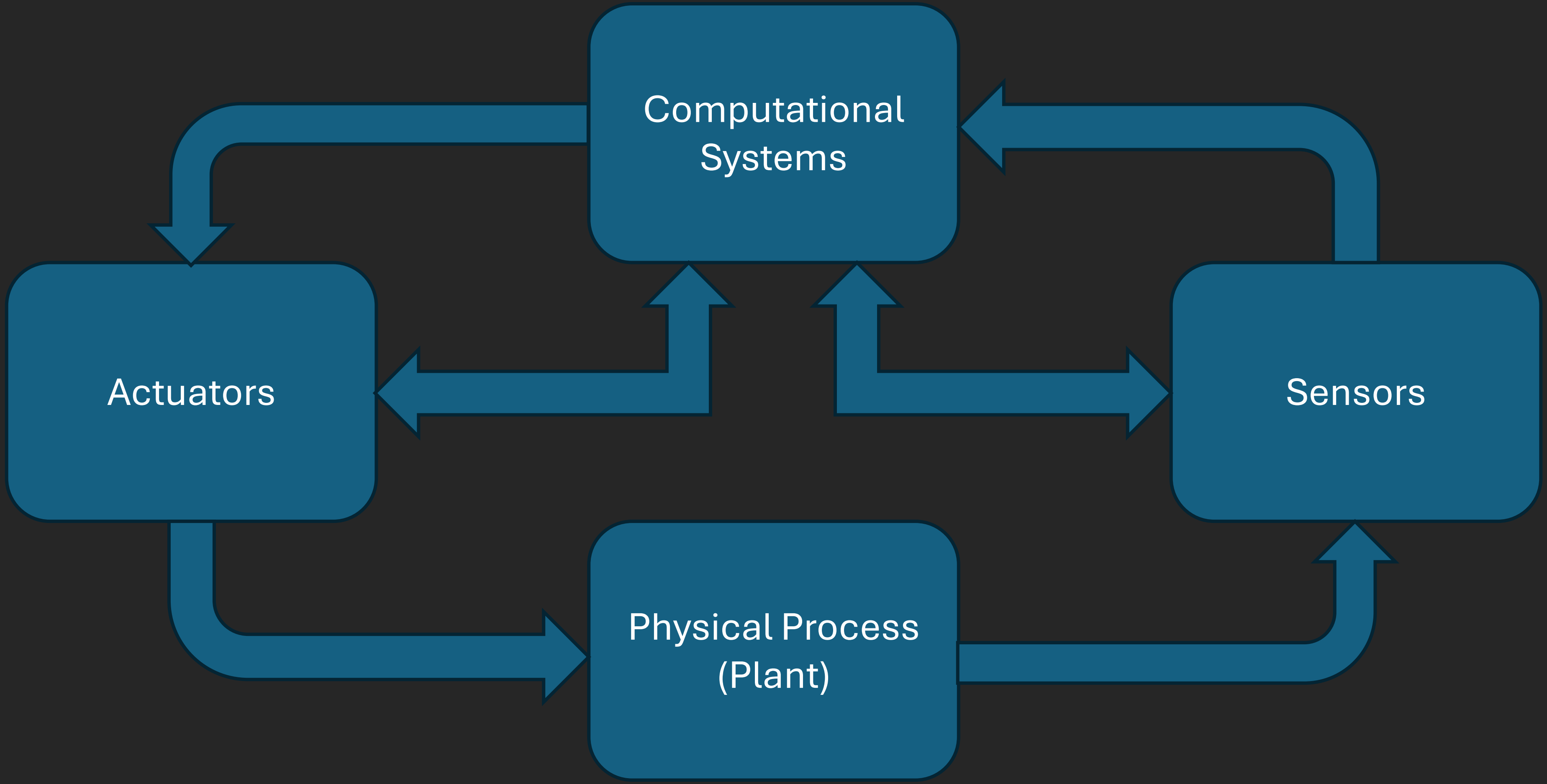

CornellEngineering
Collective Embodied Intelligence Lab
Mobile and Inflatable Interface for Human Robot Interaction
Jonathan Jaramillo, Andrew Lin, Emma Sung, Isabel Jane Hunt Richter, and Kirstin Petersen
Contact
To be presented at Ubiquitous Robots 2021

Course Introduction

What are Cyber-Physical Systems?

- Physical Components
  - Devices, machine, sensors, actuators, ...
- Cyber/Computation Components
  - Software, algorithms, data processing
- Communication
  - Physical/computational components facilitating data transfer
- Feedback Loops and Automation
  - Bidirectional data transfer that facilitates control and decision making

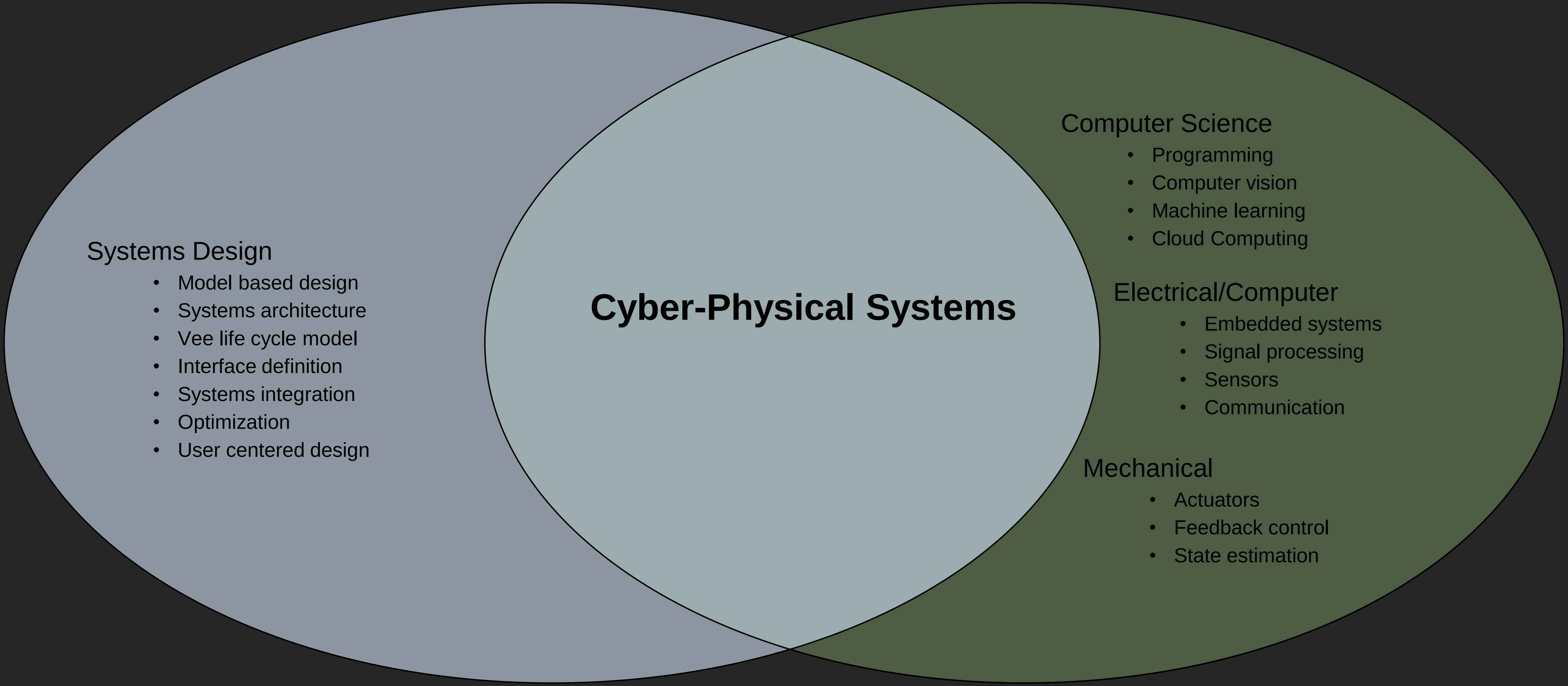
What are Cyber-Physical Systems?



Definition

Integrated systems that combine computational algorithms with physical processes, enabling real-time monitoring, control, and interaction between the digital and physical worlds.

Who Should Take this Class?



Who Should Take this Class?

Who Should Take this Class?

Survey – Student Backgrounds

- Majors
  - Computer Science
  - Mechanical Engineering
  - Electrical & Computer Engineering
  - Physics
  - Math
  - Industrial engineering & Operations Research
- Industry Experience

Questions?

Assignments

Assignments

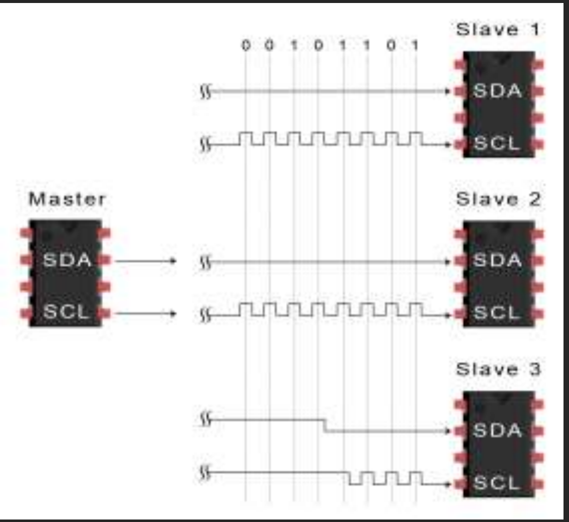
- No prelims!
- Labs –60%
  - 12 total
  - Weekly reports
- Case Studies – 30%
  - 4 total
  - Group submissions
- Participation & Quizzes – 10%
  - 1 per week
  - 5 minutes each

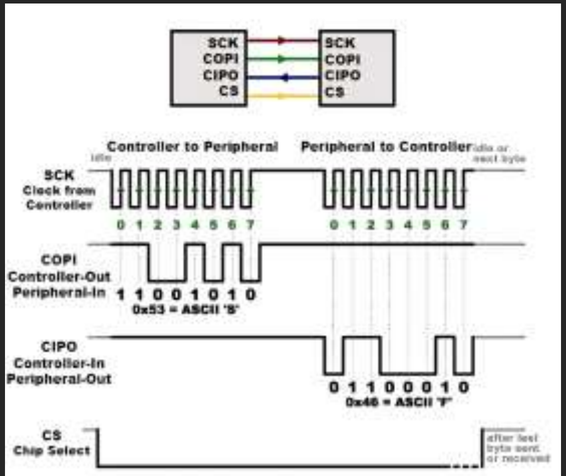
Lab 1 – Programming Languages (1/27)

- Introduction to Embedded Systems
  - Raspberry Pi Pico
  - Flashing the board, GPIO, DACs, Serial communication
- Programming Languages
  - C++/C
  - Python
  - MicroPython
- Learning Outcomes
  - Learn how to select the appropriate programming paradigm
  - Familiarize with programming

Lab 2 – Wired Communication Protocols (2/3)

- Bus Communication
  - I2C - addressing
  - Serial
- Compression
  - Computational resources
  - Baud rates Vs Transfer rates
- Learning Outcomes
  - Quantify data transfer rates
  - Understand tradeoffs between compression and transfer





Lab 3 – Wireless Communication (2/10)

- Bluetooth Vs Wifi
  - Range
  - Transfer Speeds
  - Protocols
- HTTPS vs MQTT
  - Transfer speeds
- Learning Outcomes
  - Compare wireless communication protocols
  - Understand ISO networking model layers

Lab 4 – Wireless Communication (Continued)

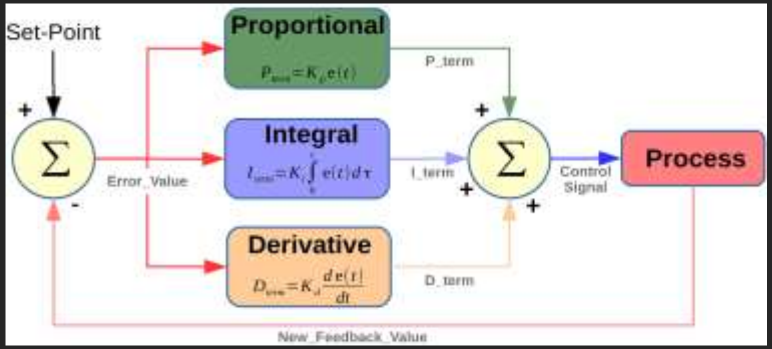
- Bluetooth Vs Wifi
  - Range
  - Transfer Speeds
  - Protocols
- HTTPS vs MQTT
  - Transfer speeds
- Learning Outcomes
  - Compare wireless communication protocols
  - Understand ISO networking model layers

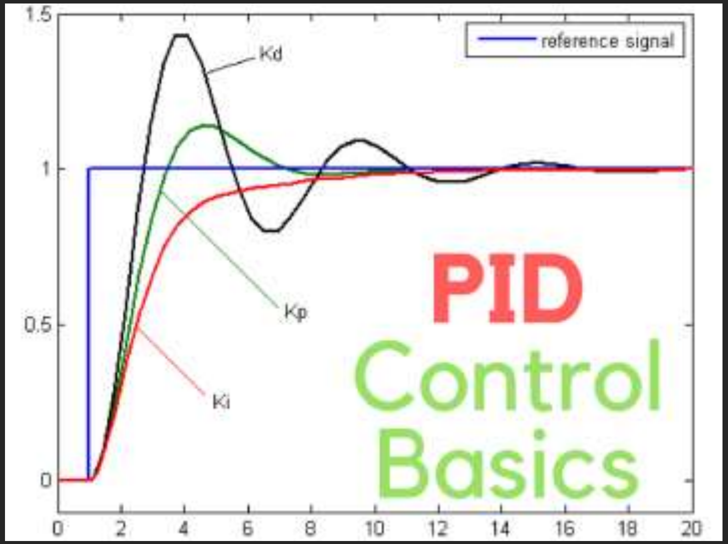
Lab 5 – Actuators (2/24)

- Motors, servos, ADC
  - PWM
  - Read/Write GPIOs
- Encoders
  - Polling
  - Hardware interrupts
- Learning Outcomes
  - Understand DC motors and encoders
  - Understand difference between polling and hardware interrupts

Lab 6 – Feedback Control (3/3)

- PID Controllers
  - Wheel speed
  - Angular position
  - Wall following
- Servos
- Learning Outcomes
  - Implement and tune PID controllers
  - Implement nested PID controllers
  - Test Servos



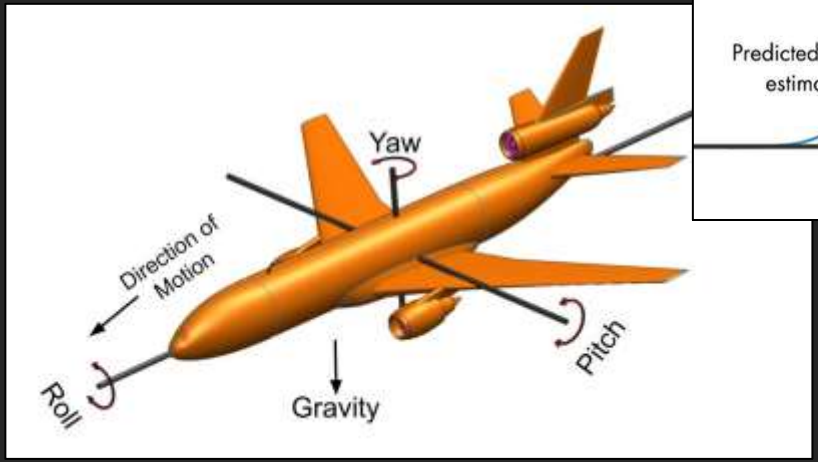


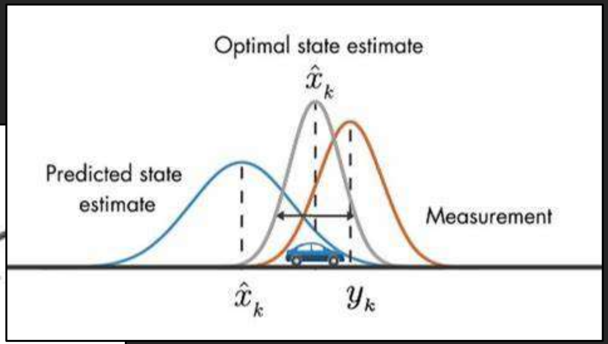
Lab 7 – Sensors & Filtering (3/10)

- Ultrasonic Range Finder Vs ToF Sensor
  - Characterize accuracy, noise, sample rates
- Gyroscope
  - Bias reduction
- Accelerometer
  - Low-pass filter
- Learning Outcomes
  - Implement and characterize sensor integration
  - Compare sensor modalities

Lab 8 – Sensor Fusion (Kalman Filter) (3/17)

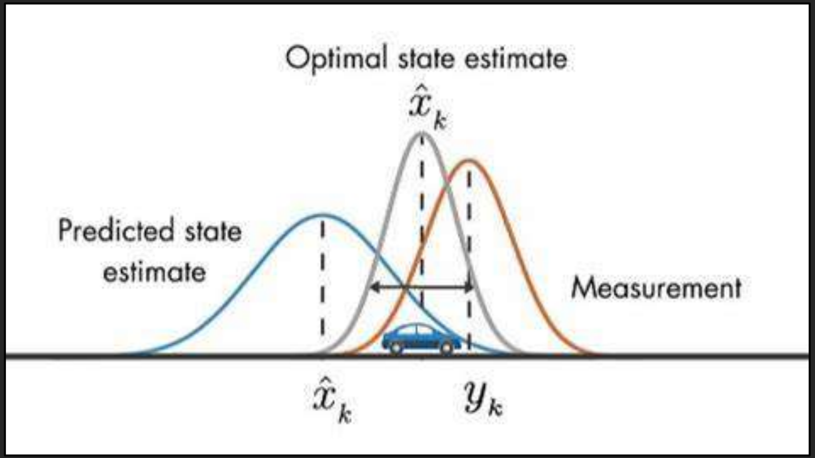
- Compute roll, pitch, and yaw
  - Characterize sensor noise
  - Complimentary filter
  - Kalman Filter
  - Data visualization
- Learning Outcomes
  - Compare trade-offs between alpha value and lag
  - Compare low-pass filter, complementary filter, and Kalman filter

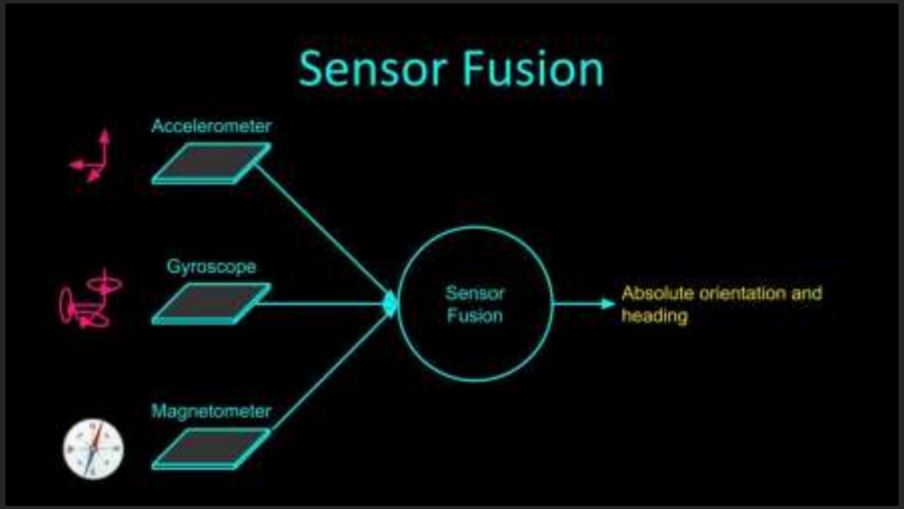




Lab 9 – Sensor Fusion (Continued) (3/24)

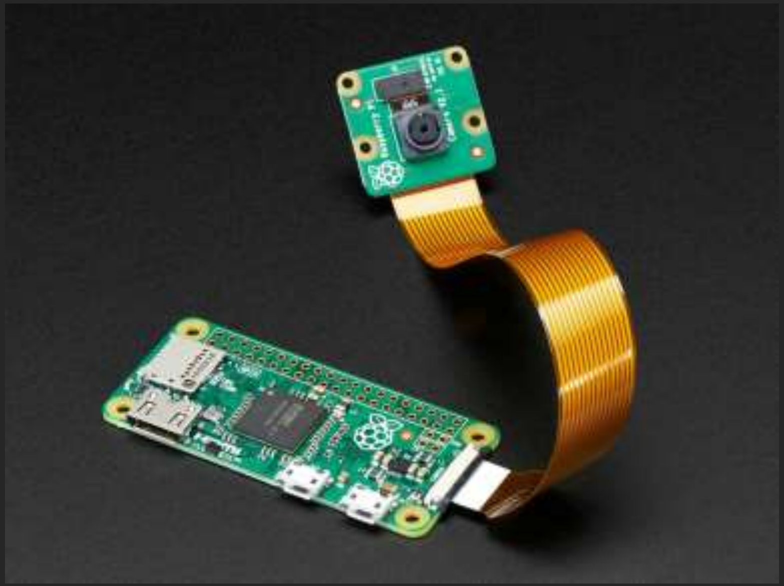
- Kalman filter sensor fusion for position
  - Characterize sensor noise
  - IMU data
  - Encoder data
  - Visualize data
- Learning Outcomes
  - Characterized performance of state estimation algorithms

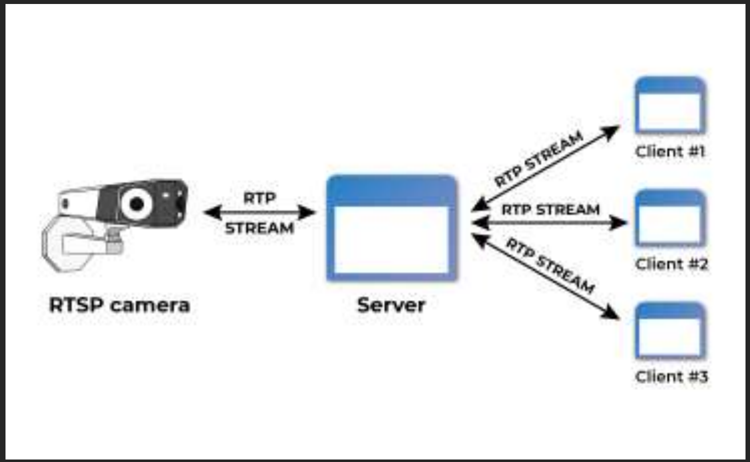




Lab 10 – Cameras and Computer Vision (4/7)

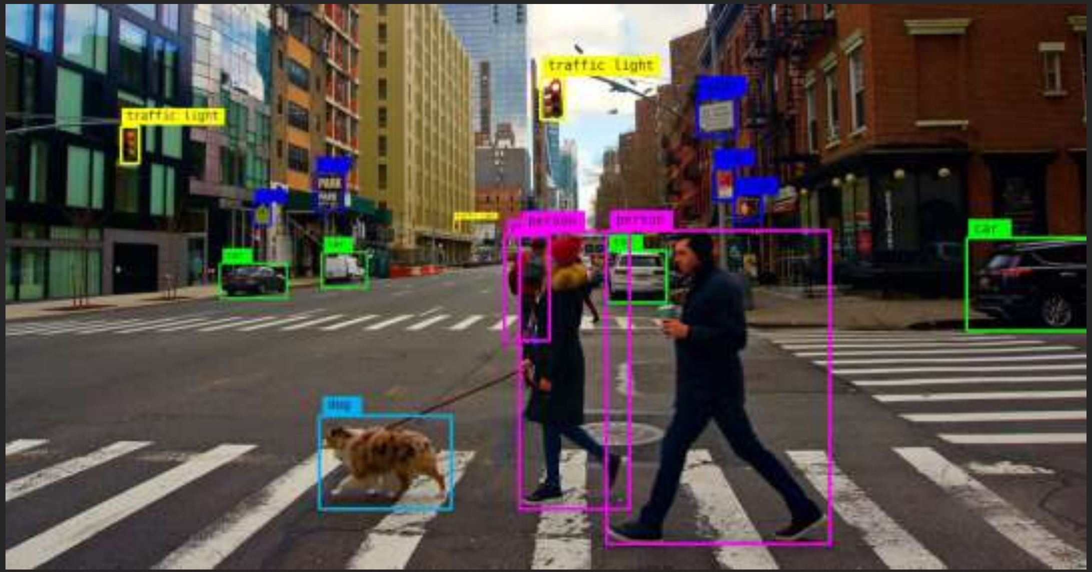
- Configure Raspberry Pi Zero
  - RTSP
  - HTTPS
- Machine learning
  - Object Detection and Tracking
  - YOLO
- Learning Outcomes
  - Basics of computer vision and perception
  - Understand video compression/codecs
  - Real time streaming protocols





Lab 11 – Cloud Computing (4/14)

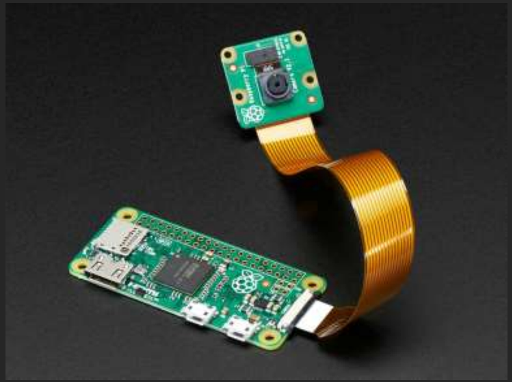
- Implement Google Vision API
  - Facial Expression Detection
  - Image Labeling
  - Object Detection
  - Text Detection
  - Logo Detection
  - Landmark detection

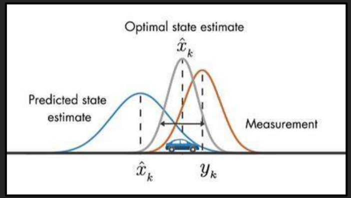


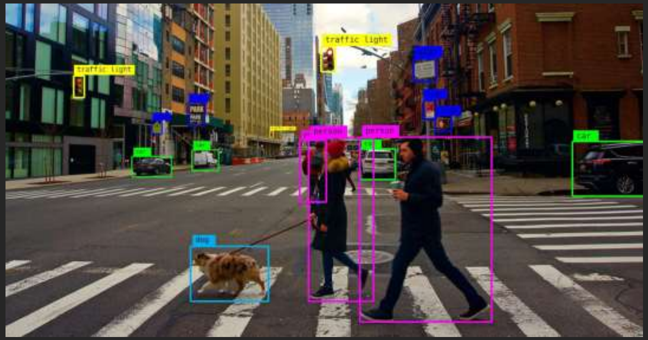
- Learning Outcomes
  - Use Google Vision API for high level tasks
  - Design computer vision systems

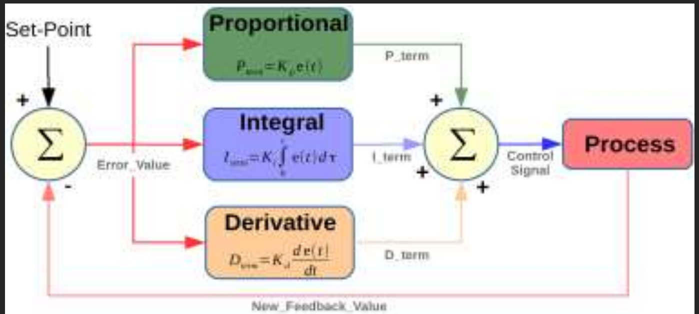
Lab 12 – Cyber-physical System (4/21)

- Put it all together









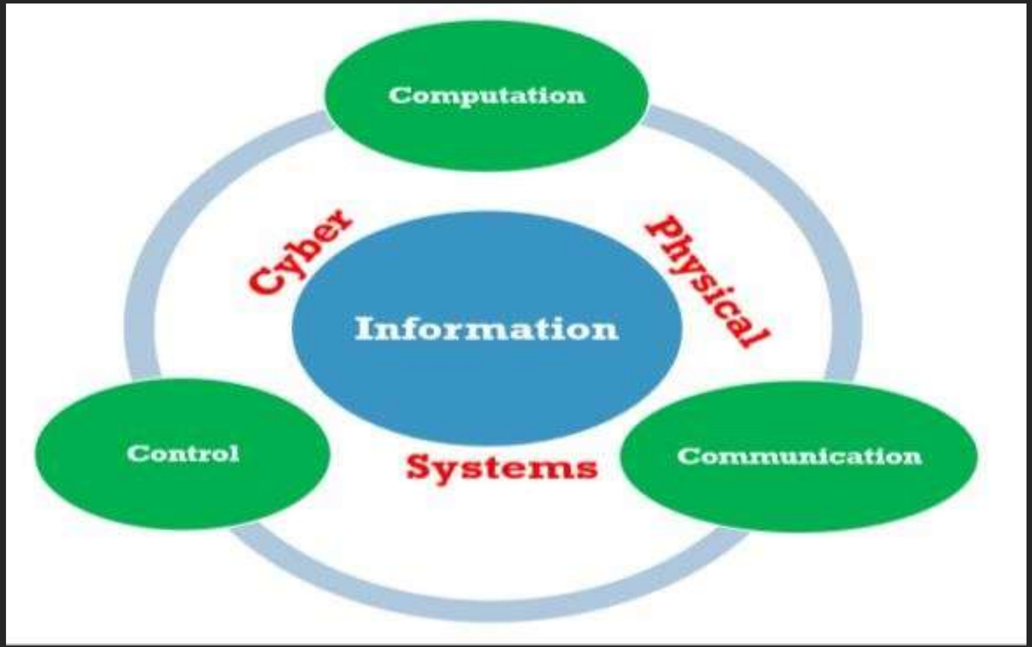
Architectures & Frameworks

CPS Architectures & Frameworks

- 3C Architecture
- 5C Architecture
- IoT Architecture
- NIST Architecture
- Edge/Fog/Cloud Computing Framework
- Digital Twin Framework

3C Architecture

- Computation
  - Data processing and decision making
- Communication
  - Transmission of data between entities
- Control
  - Actions taken by the system to manipulate the physical environment



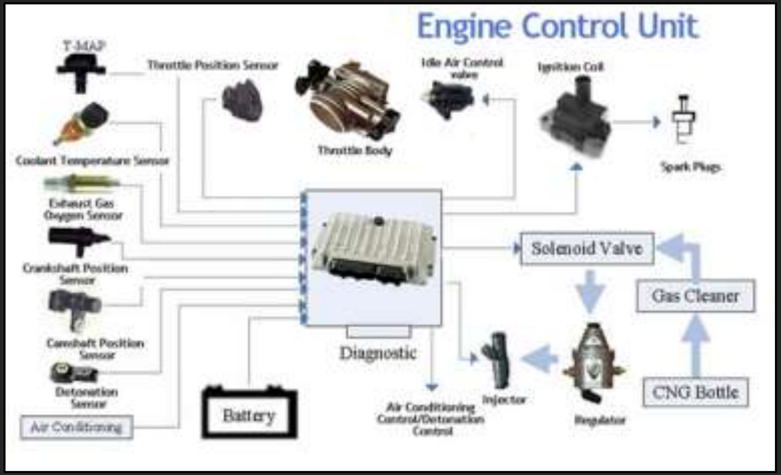
Ateş, Emre, Erkan Bostancı, and Mehmet Güzel. "Security evaluation of industry 4.0: understanding industry 4.0 on the basis of crime, big data, internet of thing (IoT) and cyber physical systems." *Güvenlik Bilimleri Dergisi* International Security Congress Special Issue (2020): 29-50.

3C Arc – Example: Modern Car

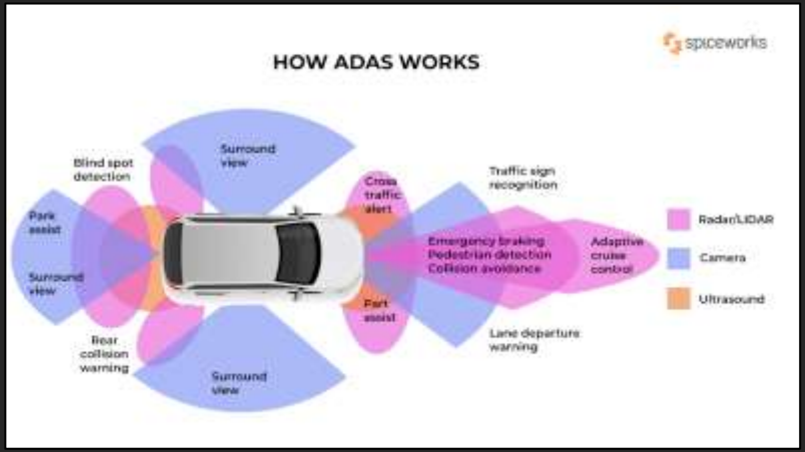
- Computational
- Communication
- Control

3C Arc – Example: Modern Car

- Computational
  - ECU
  - Advanced Driver-Assistance System
  - Infotainment System
- Communication

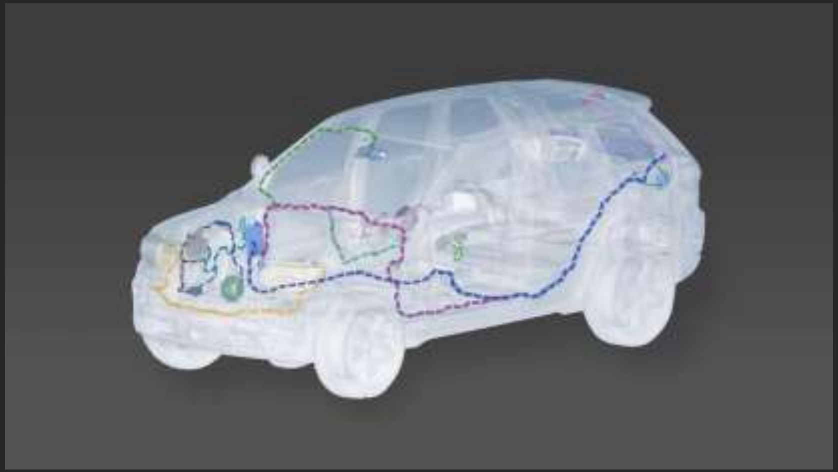


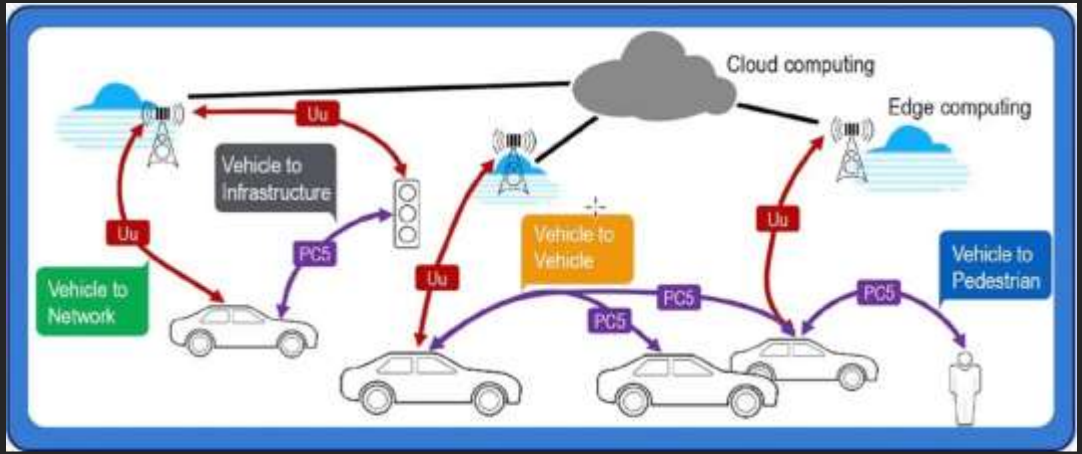
- Control



3C Arc – Example: Modern Car

- Computational
  - ECU
  - Advanced Driver-Assistance System
  - Infotainment System
- Communication
  - Intra-Vehicle Networks (CAN bus)
  - Vehicle-to-Vehicle Networks (V2V)
  - Telematics Systems
- Control





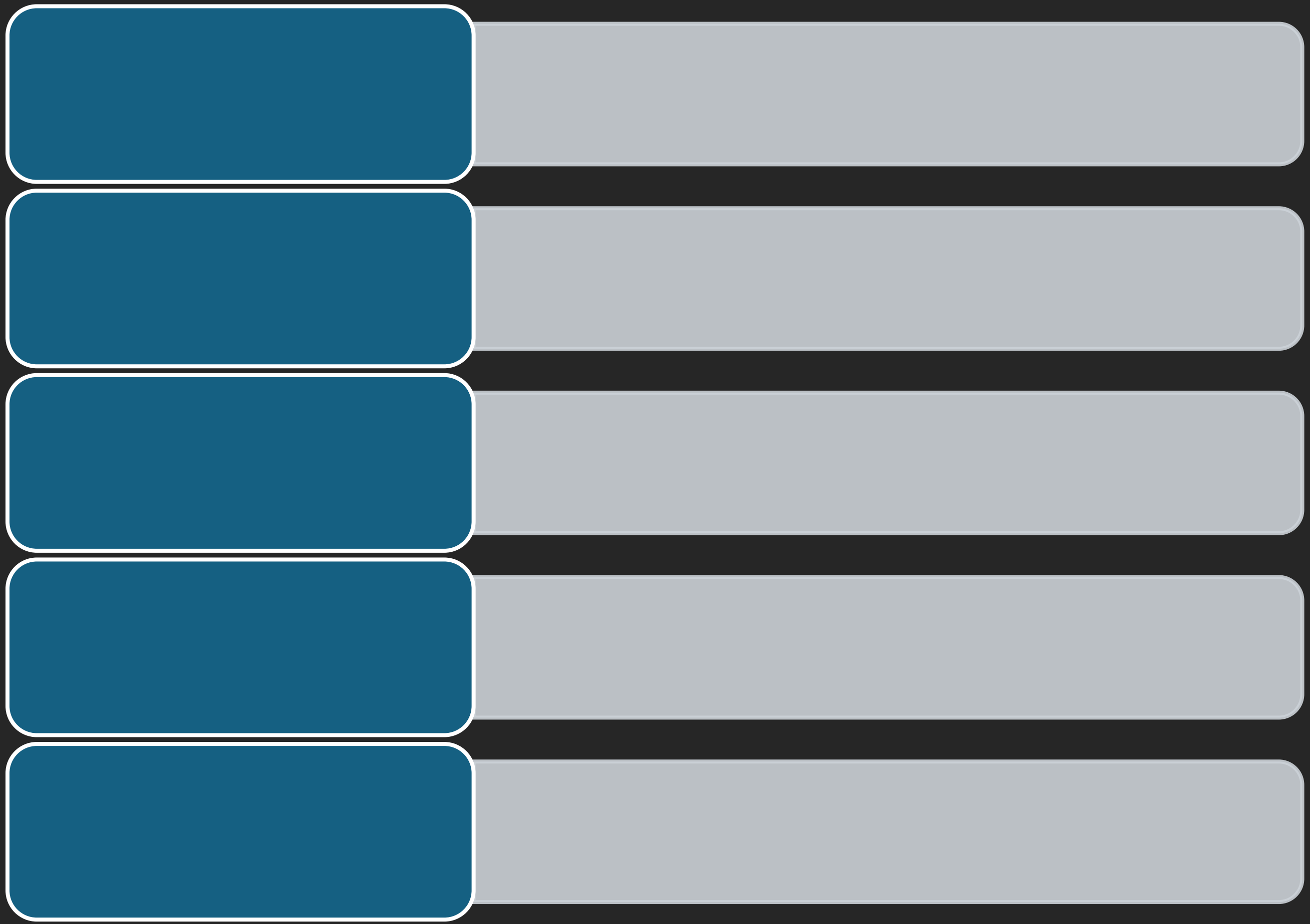
3C Arc – Example: Modern Car

- Computational
  - ECU
  - Advanced Driver-Assistance System
  - Infotainment System
- Communication
  - Intra-Vehicle Networks (CAN bus)
  - Vehicle-to-Vehicle Networks (V2V)
  - Telematics Systems
- Control
  - Actuators (steering, breaking, throttle, suspension)
  - Stability Control Systems
  - Climate Control Systems

When to use the 3C Architecture

- Advantages
  - Simple – easy to use
- Disadvantages
  - Simple
  - Doesn’t capture functionality of component
- Use: *when simplicity suffices*

5C Architecture



5C Architecture

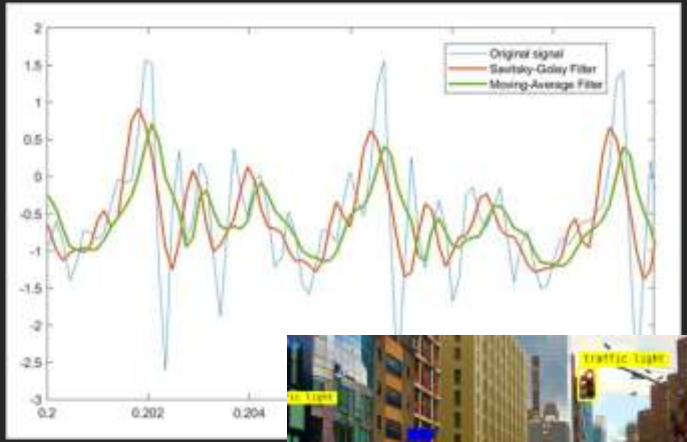
Connection

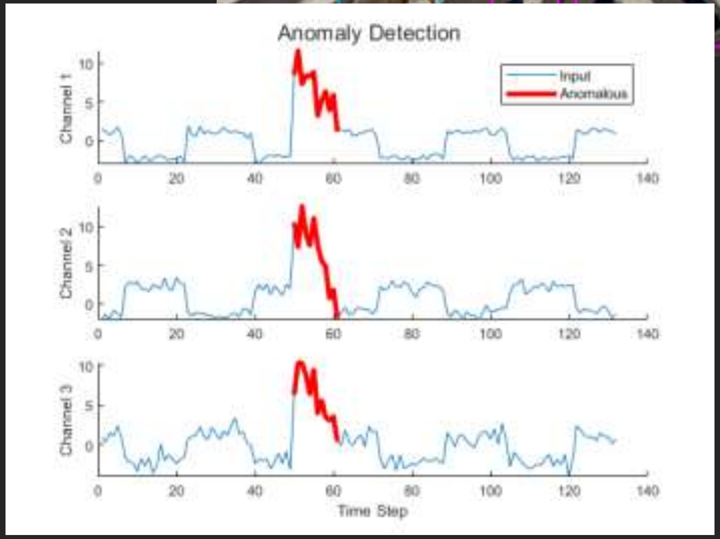
- Foundational layer comprised of sensors and actuators

5C Architecture

Connection
- Foundational layer comprised of sensors and actuators

Conversion
- Processes and converts raw data





5C Architecture

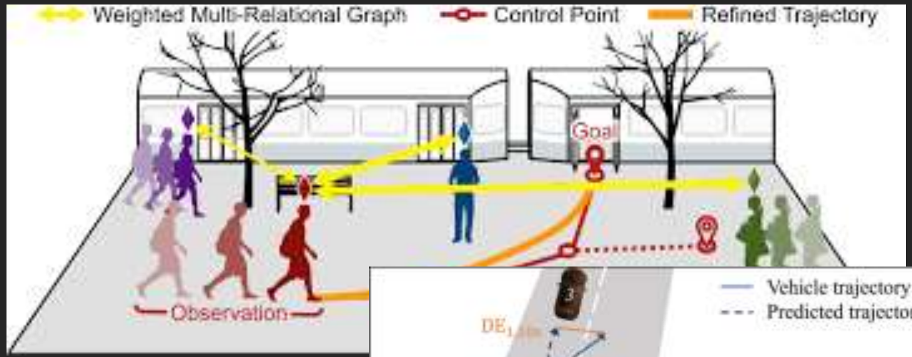
Connection
- Foundational layer comprised of sensors and actuators

Conversion
- Processes and converts raw data

Cyber
- Digital representation of physical system

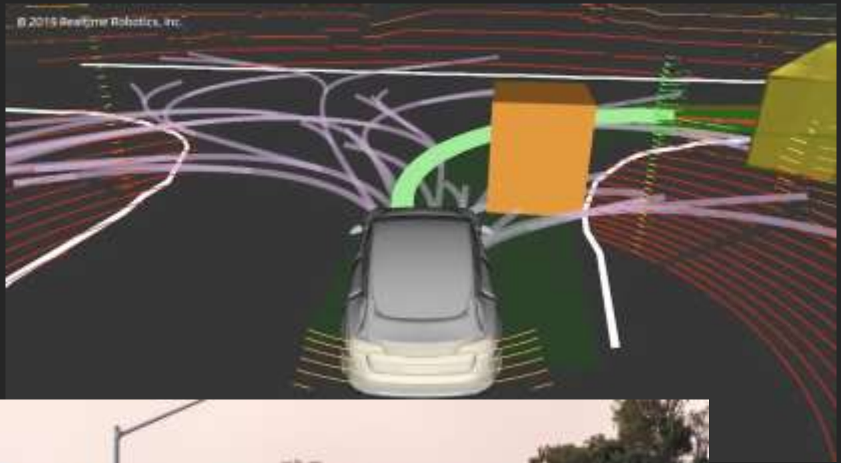
5C Architecture

Connection	• Foundational layer comprised of sensors and actuators
Conversion	• Processes and converts raw data
Cyber	• Digital representation of physical system
Cognition	• Interprets data



5C Architecture

Connection	• Foundational layer comprised of sensors and actuators
Conversion	• Processes and converts raw data
Cyber	• Digital representation of physical system
Cognition	• Interprets data
Configuration	• Decisions regarding manipulating the environment



5C Arch – Example: Smart Wearable

Connection	
Conversion	
Cyber	
Cognition	
Configuration	

5C Arch – Example: Smart Wearable

Connection	• Wearable devices collect health data and connect to smartphones
Conversion	
Cyber	
Cognition	
Configuration	

5C Arch – Example: Smart Wearable

Connection	• Wearable devices collect health data and connect to smartphones
Conversion	• Data is preprocessed on the device or phone
Cyber	
Cognition	
Configuration	

5C Arch – Example: Smart Wearable

Connection	• Wearable devices collect health data and connect to smartphones
Conversion	• Data is preprocessed on the device or phone
Cyber	• Cloud services create a digital profile of the user's health
Cognition	
Configuration	

5C Arch – Example: Smart Wearable

Connection	• Wearable devices collect health data and connect to smartphones
Conversion	• Data is preprocessed on the device or phone
Cyber	• Cloud services create a digital profile of the user's health
Cognition	• Analytics provide health insights and activity recommendations
Configuration	

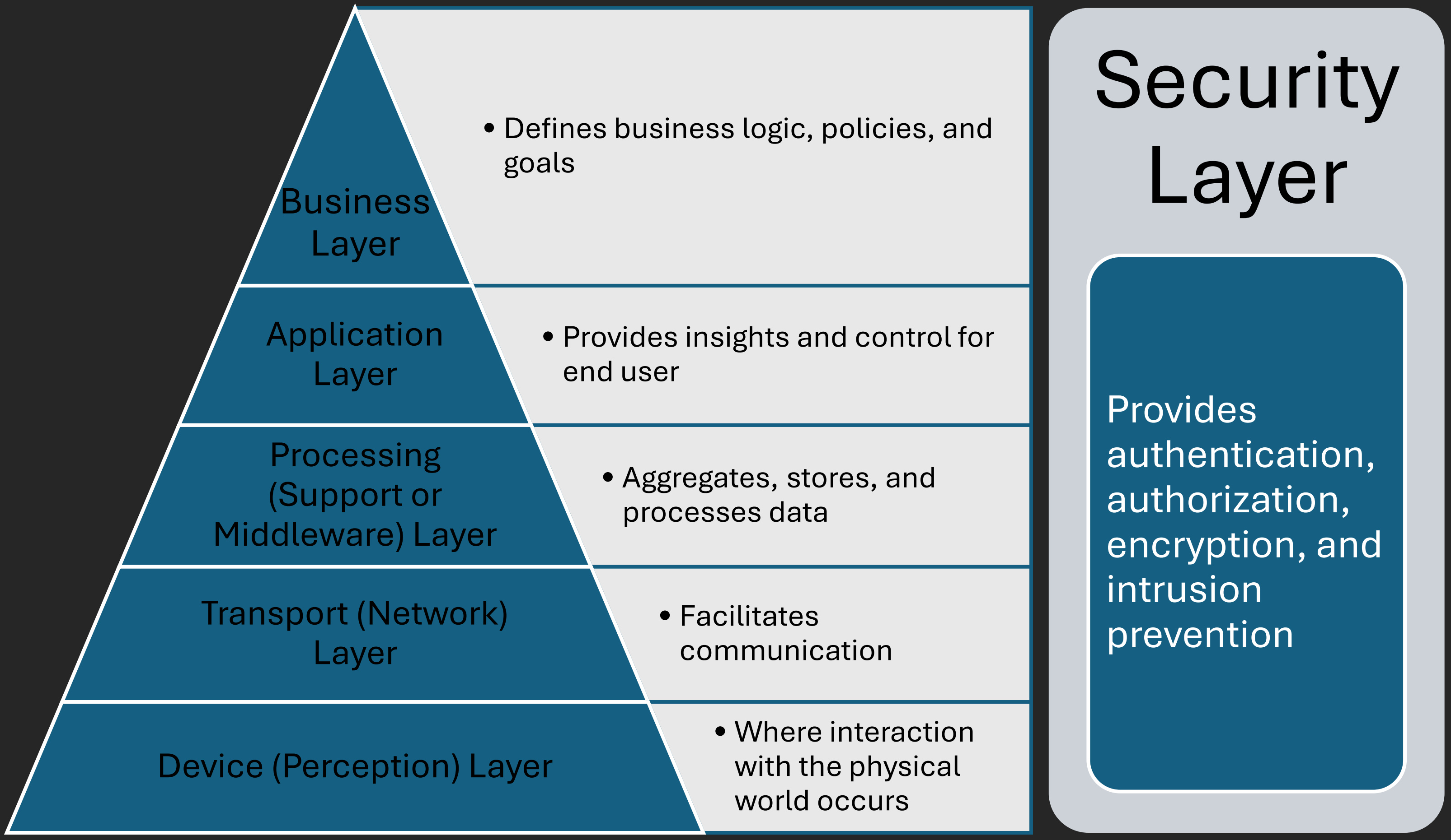
5C Arch – Example: Smart Wearable

Connection	• Wearable devices collect health data and connect to smartphones
Conversion	• Data is preprocessed on the device or phone
Cyber	• Cloud services create a digital profile of the user's health
Cognition	• Analytics provide health insights and activity recommendations
Configuration	• Devices adjust settings or prompt users to act (e.g., stand up, stretch).

When to use the 5C Architecture

- Advantages
  - Data-centric framework
  - More nuanced than 3C
- Disadvantages
  - Poor conceptualization of communication and physical systems
  - Assumes digital twin model
- Use: *for data-intensive operations* and *intelligent decision making*

Internet of Things Architecture



IoT Arch – Example: Smart Traffic City

- Device (Perception) Layer
  - Cameras, inductive loop detectors, traffic lights, variable speed limits, express lanes
- Transport (Network) Layer
  - Wired connections and wireless networks
- Processing (Support or Middleware) Layer
  - Data processing, databases (historical traffic patterns), edge computing devices (CV)
- Application Layer
  - Dashboards for monitoring, mobile apps for traffic updates
- Business Layer
  - Policy (traffic management), revenue (tolls, funding budgets)
- Security Layer
  - Encryption, authentication and authorization, intrusion detection systems

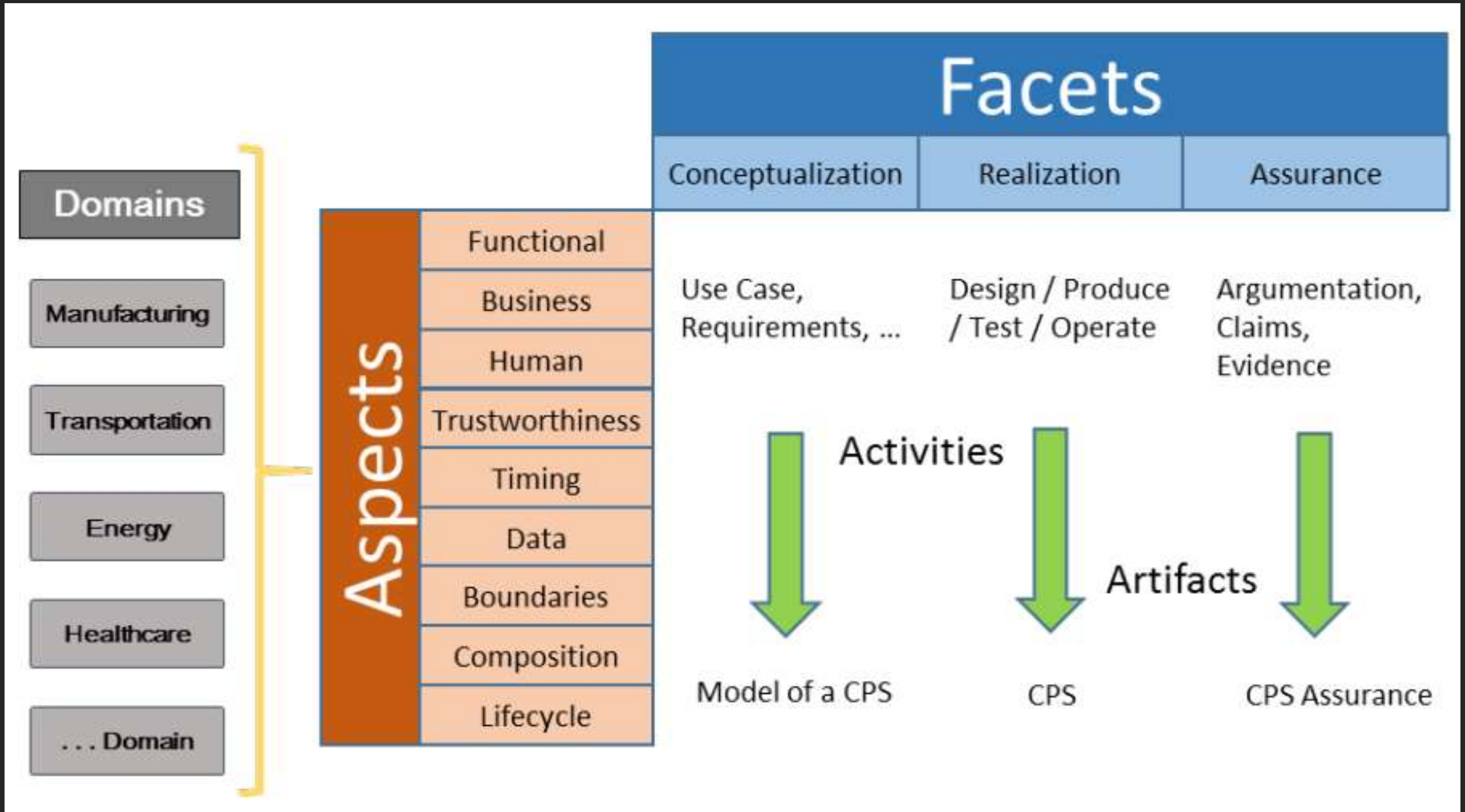
When to use the IoT Architecture

- Advantages
  - Strong functional categorization
  - Incorporates business models
- Disadvantages
  - Business model might not be defined
- Use: *for consumer products*

NIST Architecture

- Domains – Specific application or environment
- Facets – Stages of engineering process
  - Conceptualization – define requirements and high-level goals
  - Realization – design, production, implementation
  - Assurance – verification and validation
- Aspects – cross-cutting concerns for the entire system
  - Functional, business, human, trustworthiness, timing, data, boundaries, composition, and lifecycle

NIST Architecture



When to use the IoT Architecture

- Advantages
  - Wholistic modeling approach
  - Adds context to
- Disadvantages
  - Loose sight of specific functional engineering components
- Use: *projects with largescale public stakeholdership*

Edge, Fog, Cloud Computing Framework

- Edge Computing – Distributed computing paradigm that brings limited computation and data storage closer to source of data
  - Example – embedded processing performing signal process
- Fog Computing – non-centralized, semi-distributed computing
  - Example – Cellular base stations have computational units for signal processing, or ISP distributing video streaming services
- Cloud Computing – Centralized computing paradigm that uses server farms for scalable computational and storage services
  - Examples - Azure, Google cloud, AWS

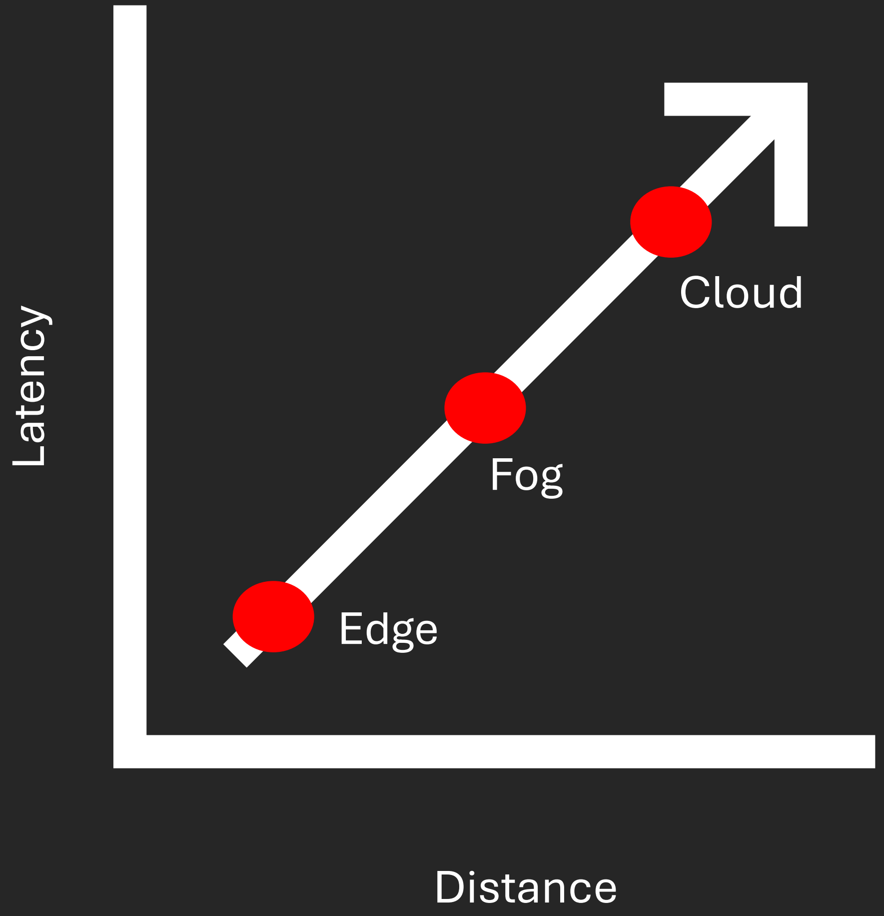
Edge, Fog, Cloud Computing Framework

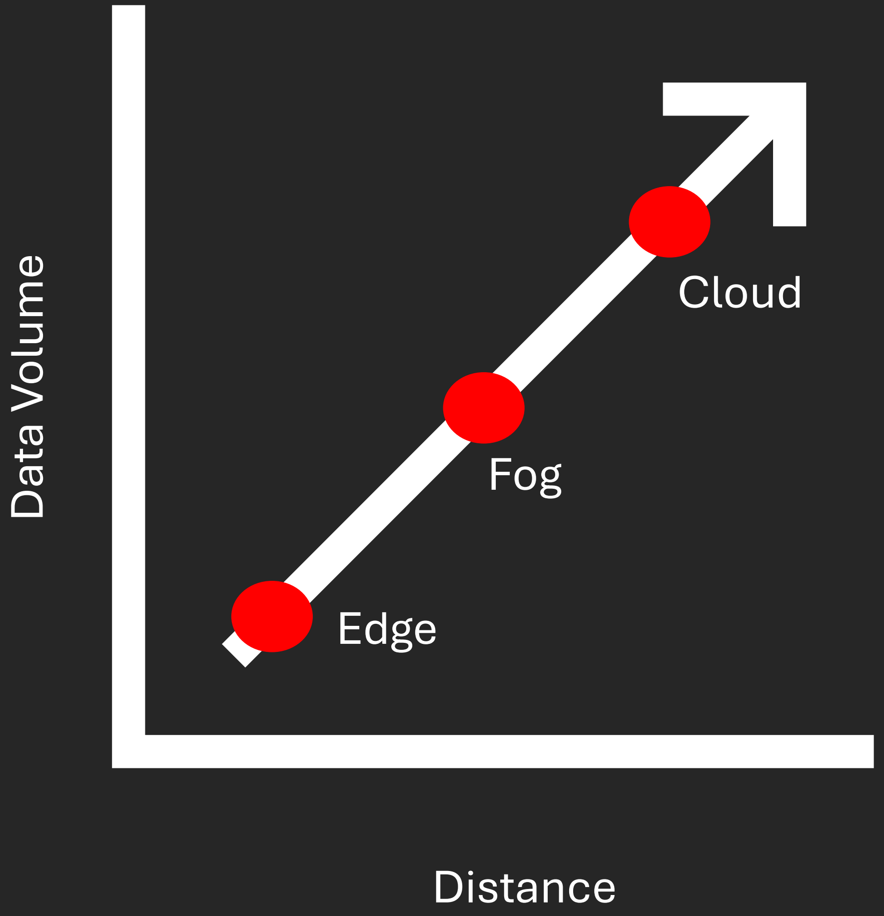
- Proximity to Data Source
  - Edge: Closest
  - Fog: Between (network gateways or routers)
  - Cloud: Furthest (remote data centers)
- Latency
  - Edge: Lowest latency
  - Fog: Moderate latency
  - Cloud: Higher latency

Edge, Fog, Cloud Computing Framework

- Data Processing
  - Edge: Processes data locally on the devices or nearby servers.
  - Fog: Processes data partially, filtering or aggregating before sending it to the cloud.
  - Cloud: Centralized processing in large-scale data centers.
- Data Volume
  - Edge: Handles smaller volumes of data (localized).
  - Fog: Handles intermediate volumes of data.
  - Cloud: Designed to handle large volumes of data for in-depth analysis and storage.

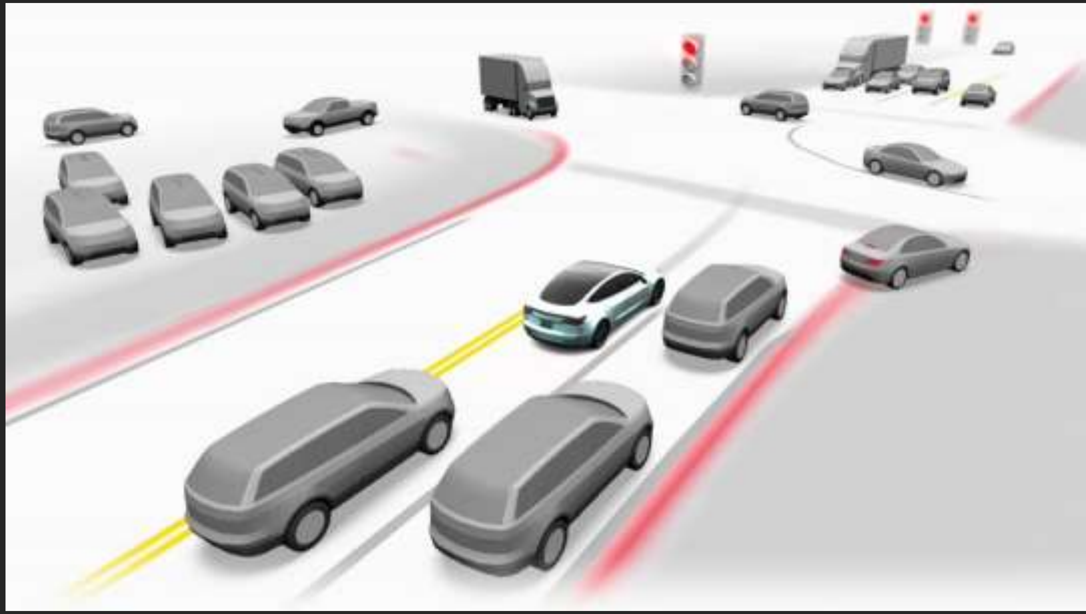
Edge, Fog, Cloud Computing Framework



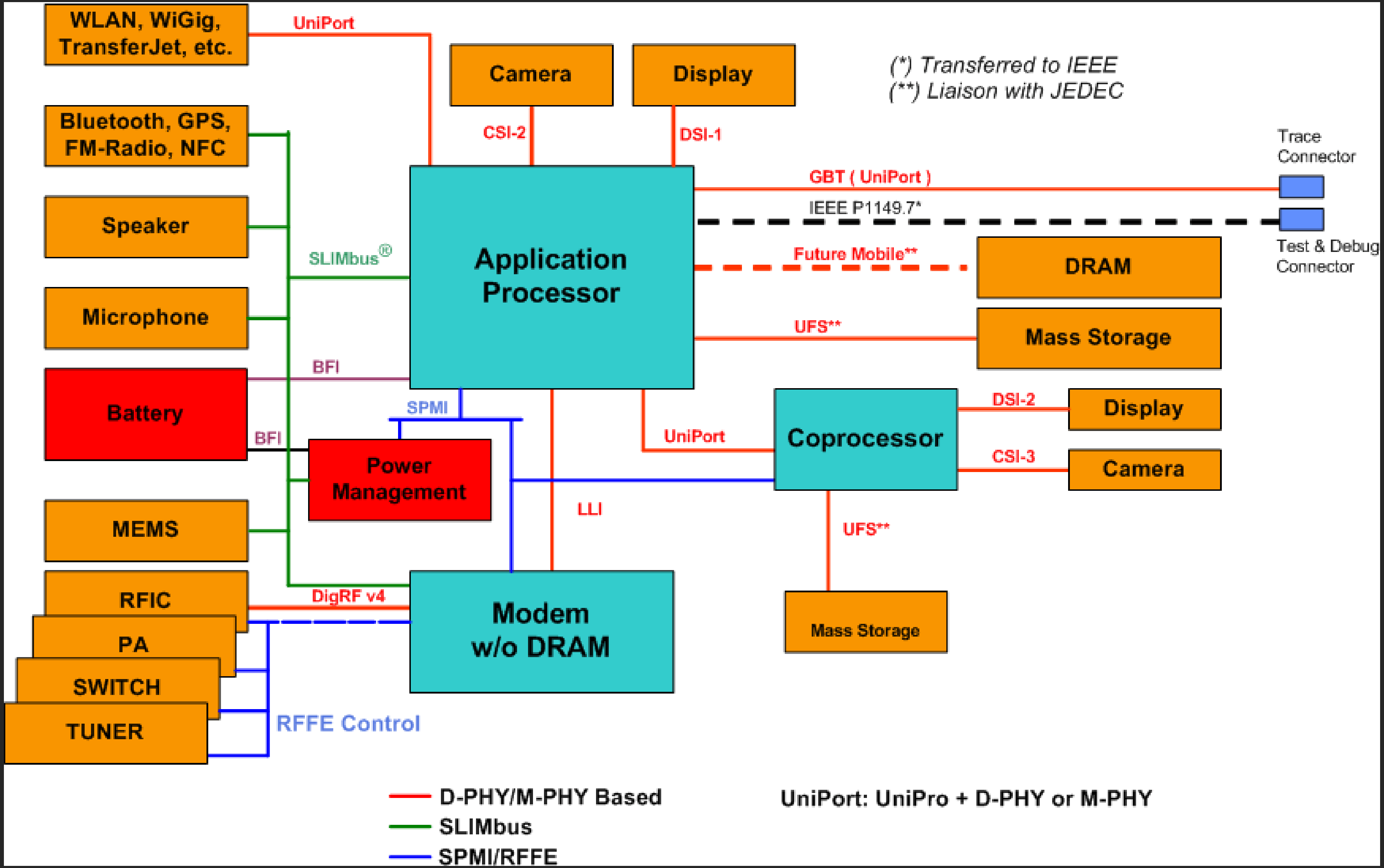


Digital Twin Framework

- Definition – Model of a physical system
  - Continuously updated with real-time data
  - Mirrors, simulates, and analyzes the system
  - Predict issues and optimize performance
- Examples
  - Aircraft maintenance schedules
  - Amazon warehouse stock
  - Wind farm modeling

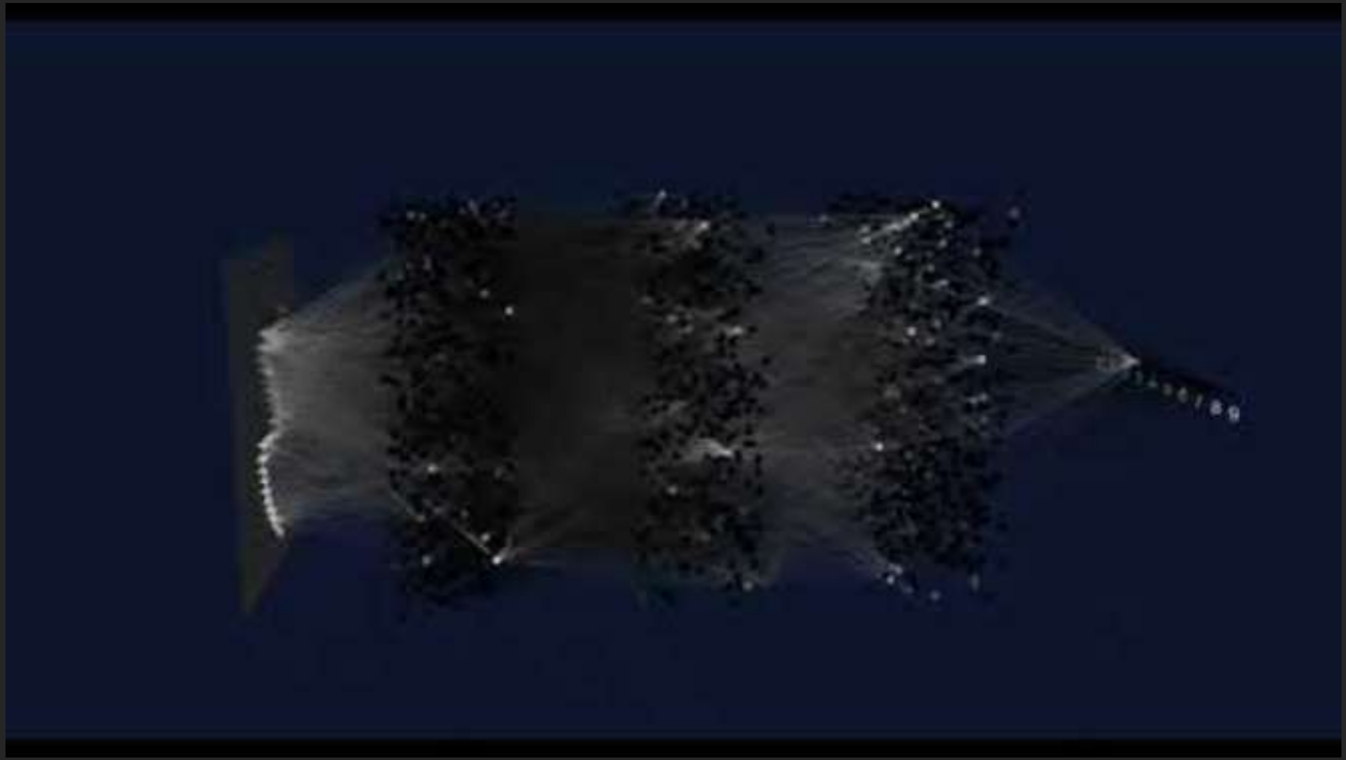


Referential Transparency & System Diagrams


WLAN, WiGig, TransferJet, etc.
UniPort
Camera
Display
(*) Transferred to IEEE
(**) Liaison with JEDEC
Bluetooth, GPS, FM-Radio, NFC
CSI-2
DSI-1
Trace Connector
GBT (UniPort)
IEEE P1149.7*
Speaker
SLIMbus®
Application Processor
Future Mobile**
DRAM
Test & Debug Connector
Microphone
UFS**
Mass Storage
BFI
Battery
SPMI
DSI-2
Display
BFI
UniPort
Coprocessor
Power Management
CSI-3
Camera
LLI
MEMS
UFS**
RFIC
DigRF v4
Modem w/o DRAM
PA
Mass Storage
SWITCH
TUNER
RFFE Control
D-PHY/M-PHY Based
SLIMbus
SPMI/RFFE
UniPort: UniPro + D-PHY or M-PHY

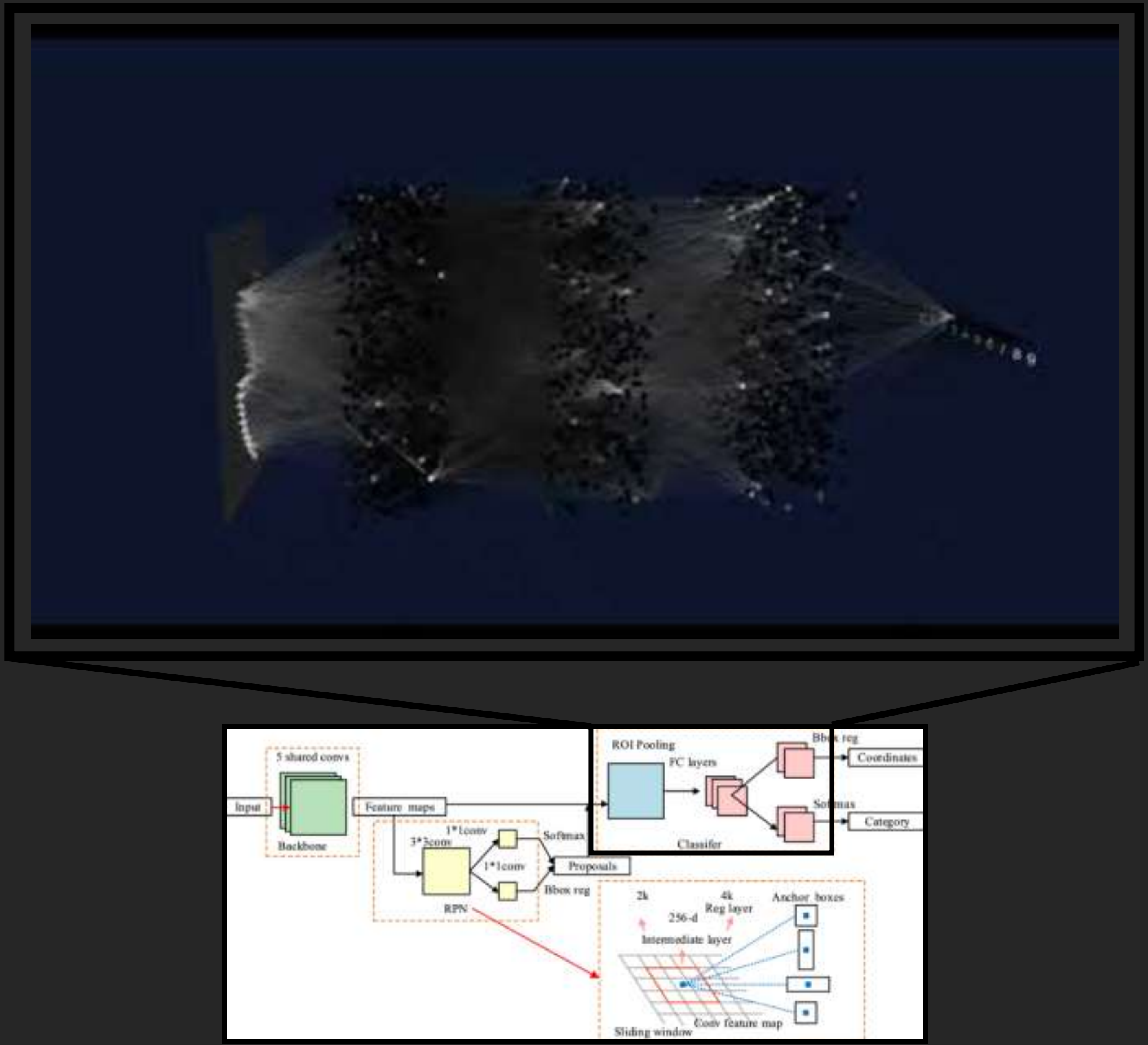
Abstraction in CPS

- Why use abstraction?



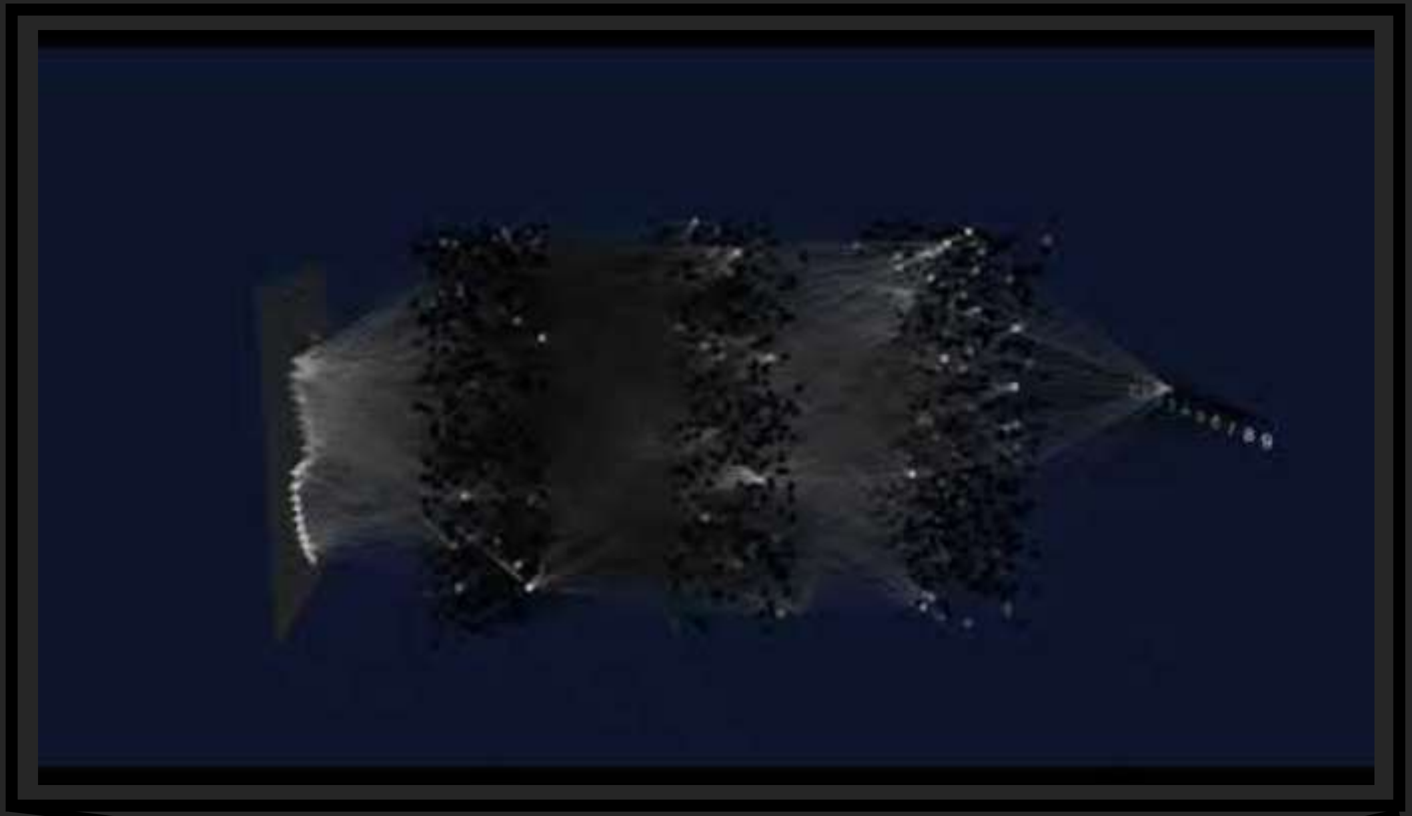
Abstraction in CPS

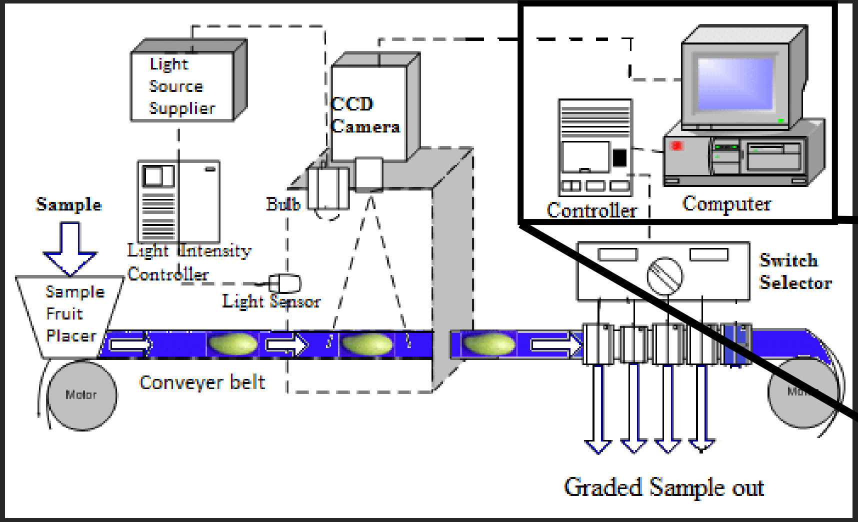
- Why use abstraction?

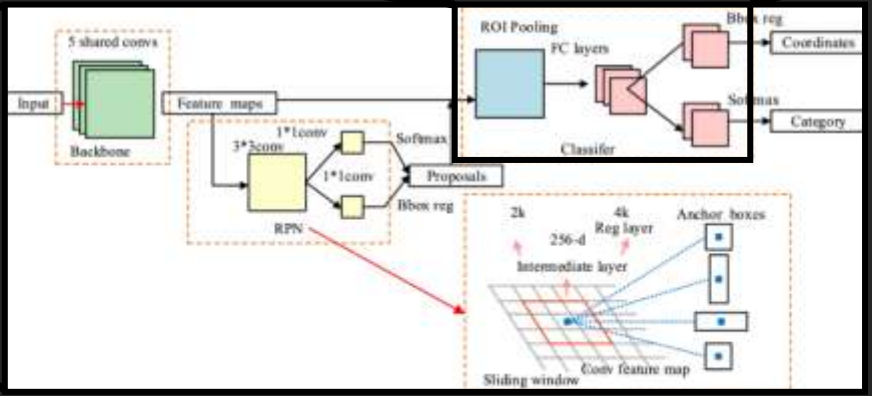


Abstraction in CPS

- Why use abstraction?







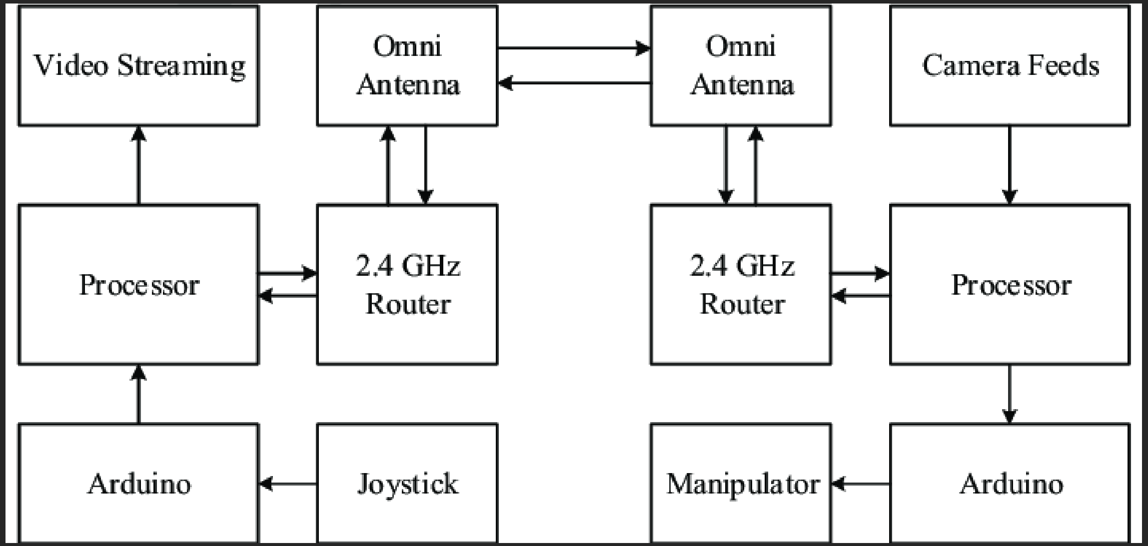
Key Principles of Abstraction

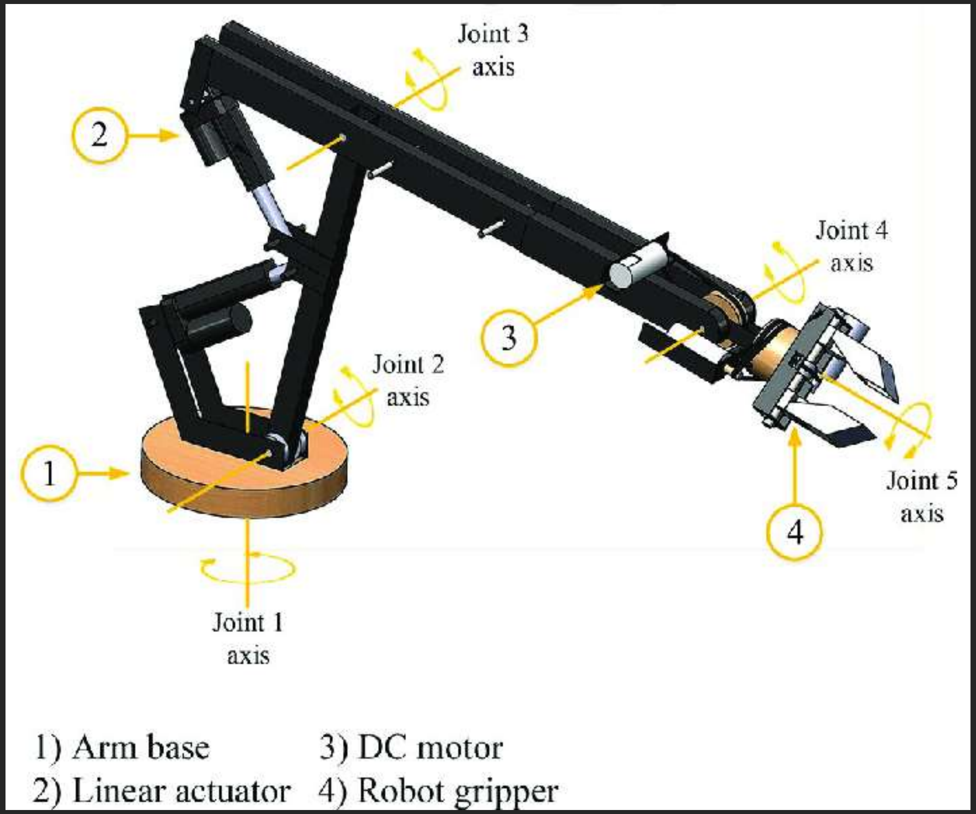
- Precisely define interfaces
  - Communication protocols, data structures, endianness, units, etc. ...
- Precisely define functionality
  - Performance, mathematical function, environmental manipulation

Precise Interfaces + **Precise Functionality** = **Referential Transparency**

Referential Transparency

- Modeling and development of a five DoF vision based remote operated robotic arm with transmission control protocol





Referential Transparency

- Improved reasoning and complexity management
- Enhanced testability
- Simplified debugging
- Modularity and useability