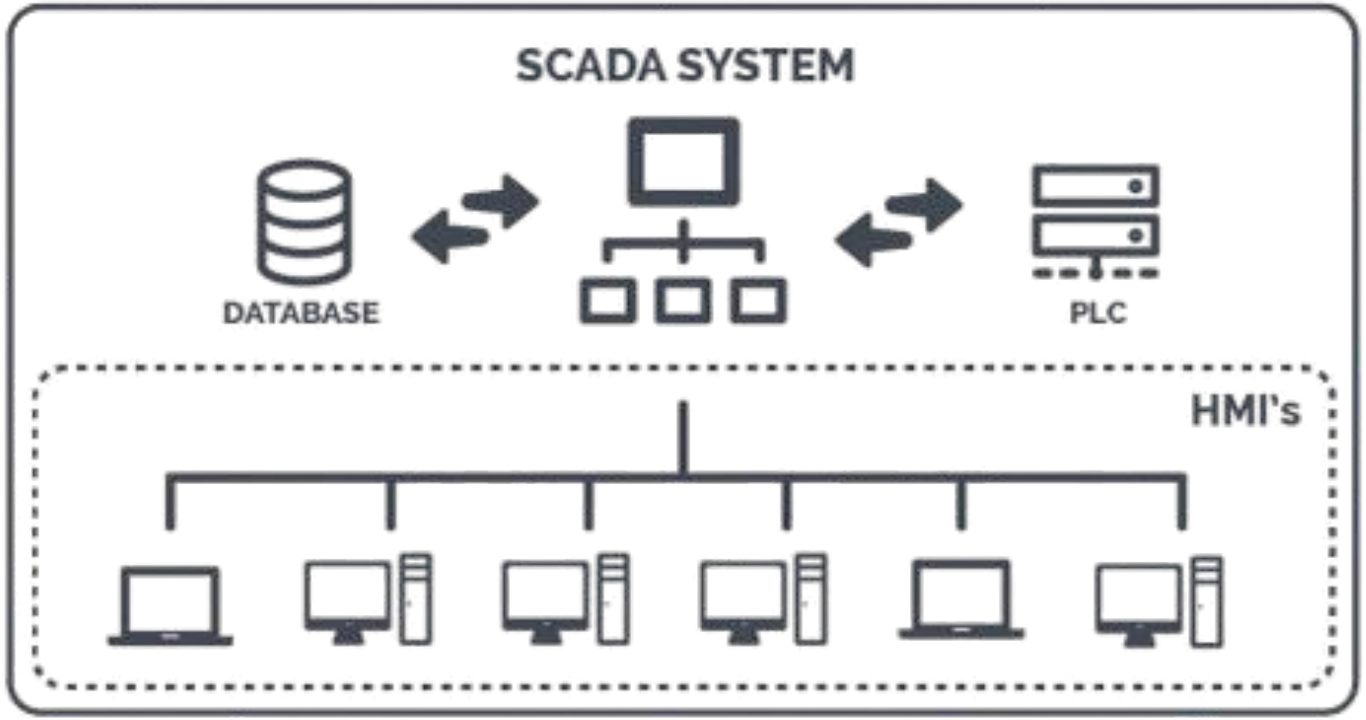
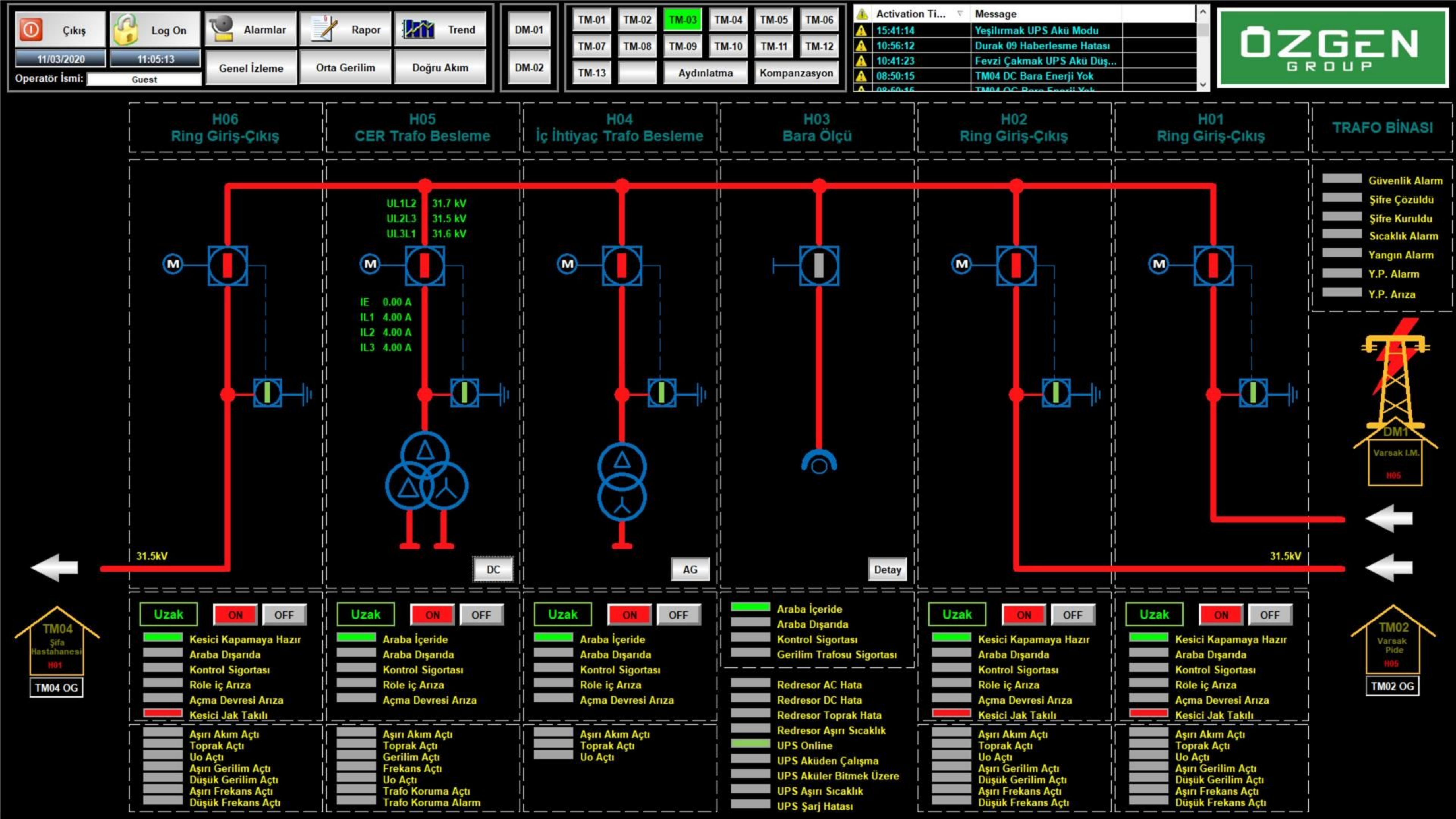
SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

Adaptive Trust Models for Securing SCADA Systems



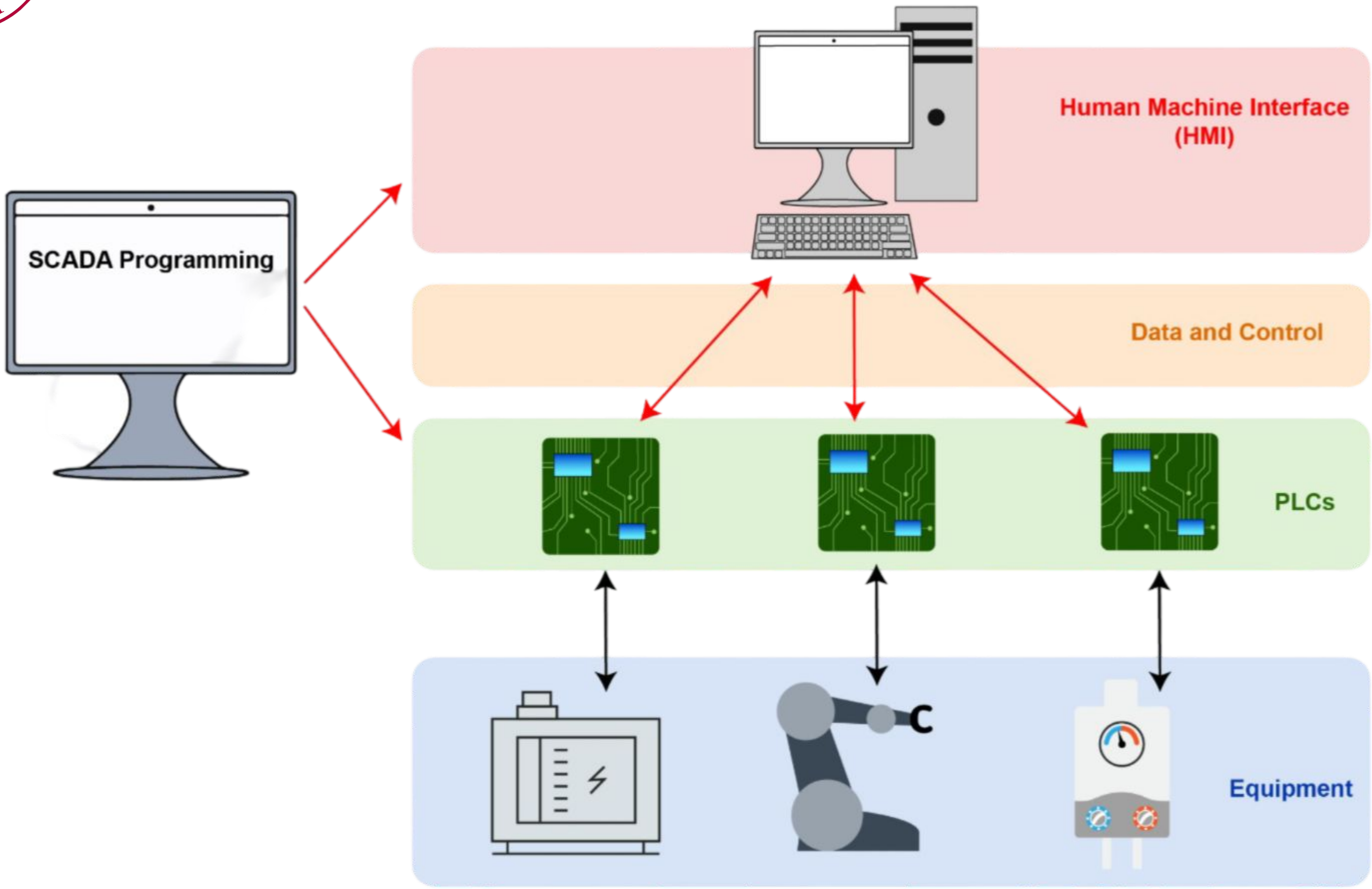
Marley Willyoung ⬩ CSEN353 ⬩ Fall 2024

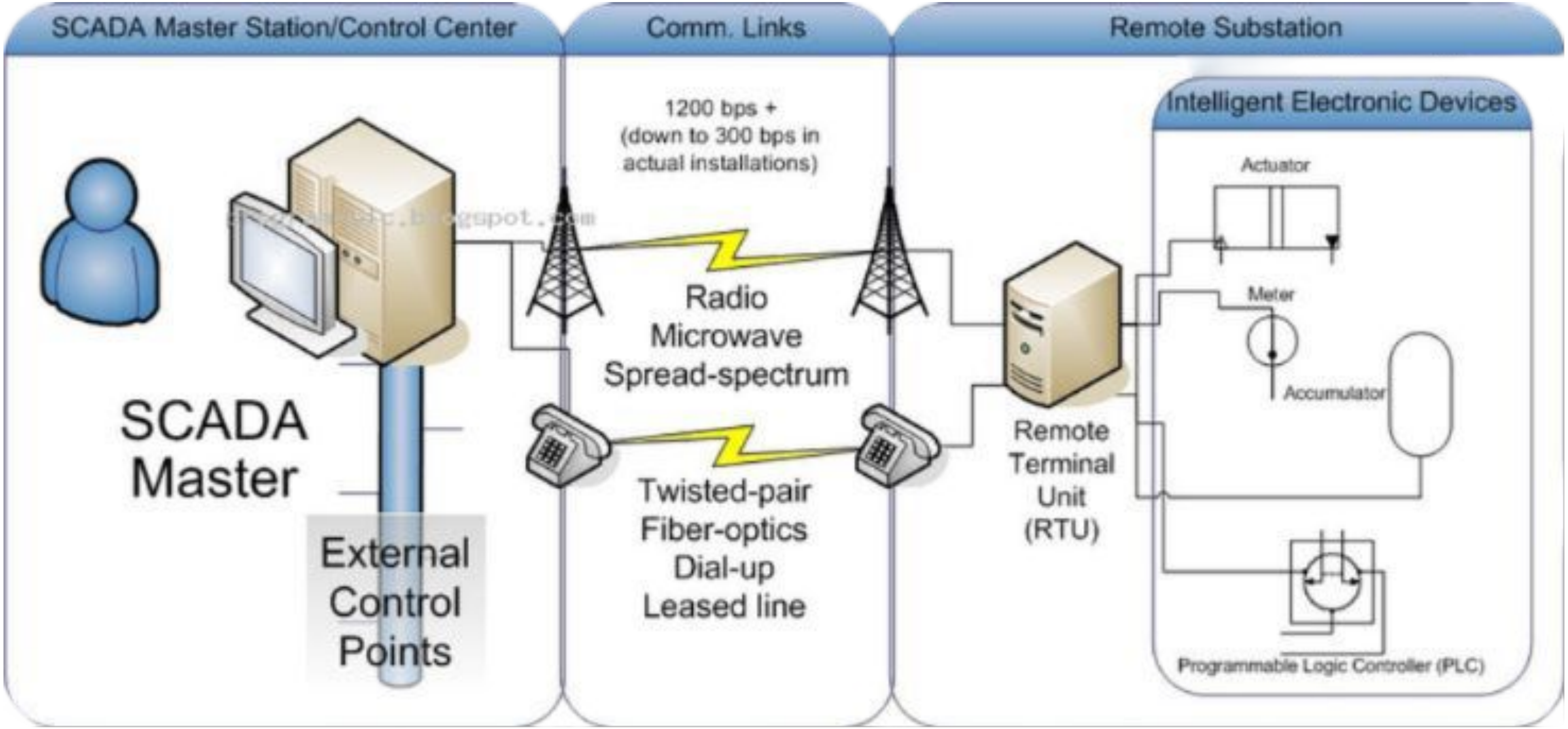


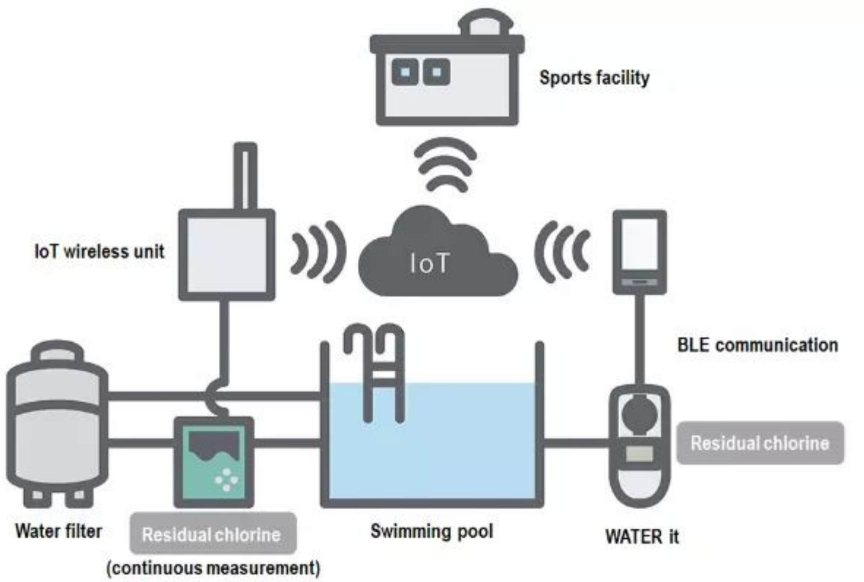
Çıkış
Log On
Alarmlar
Rapor
Trend
11/03/2020
11:05:13
Genel İzleme
Orta Gerilim
Doğru Akım
Operatör İsmi:
Guest
DM-01
DM-02
TM-01 TM-02 TM-03 TM-04 TM-05 TM-06
TM-07 TM-08 TM-09 TM-10 TM-11 TM-12
TM-13
Aydınlatma
Kompanzasyon
Activation Ti...
Message
15:41:14 Yeşilırmak UPS Akü Modu
10:56:12 Durak 09 Haberlesme Hatası
10:41:23 Fevzi Çakmak UPS Akü Düş...
08:50:15 TM04 DC Bara Enerji Yok
ÖZGEN GROUP
H06 Ring Giriş-Çıkış
H05 CER Trafo Besleme
H04 İç İhtiyaç Trafo Besleme
H03 Bara Ölçü
H02 Ring Giriş-Çıkış
H01 Ring Giriş-Çıkış
TRAFO BİNASI
Güvenlik Alarm
Şifre Çözüldü
Şifre Kuruldu
Sıcaklık Alarm
Yangın Alarm
Y.P. Alarm
Y.P. Arıza
UL1L2 31.7 kV
UL2L3 31.5 kV
UL3L1 31.6 kV
IE 0.00 A
IL1 4.00 A
IL2 4.00 A
IL3 4.00 A
M
31.5kV
DC
AG
Detay
DM1
Varsak İ.M.
H05
TM04
Şifa Hastahanesi
H01
TM04 OG
TM02
Varsak Pide
H05
TM02 OG
Uzak ON OFF
Kesici Kapamaya Hazır
Araba Dışarıda
Kontrol Sigortası
Röle iç Arıza
Açma Devresi Arıza
Kesici Jak Takılı
Aşırı Akım Açtı
Toprak Açtı
Uo Açtı
Aşırı Gerilim Açtı
Düşük Gerilim Açtı
Aşırı Frekans Açtı
Düşük Frekans Açtı
Araba İçeride
Gerilim Açtı
Frekans Açtı
Trafo Koruma Açtı
Trafo Koruma Alarm
Gerilim Trafosu Sigortası
Redresor AC Hata
Redresor DC Hata
Redresor Toprak Hata
Redresor Aşırı Sıcaklık
UPS Online
UPS Aküden Çalışma
UPS Aküler Bitmek Üzere
UPS Aşırı Sıcaklık
UPS Şarj Hatası

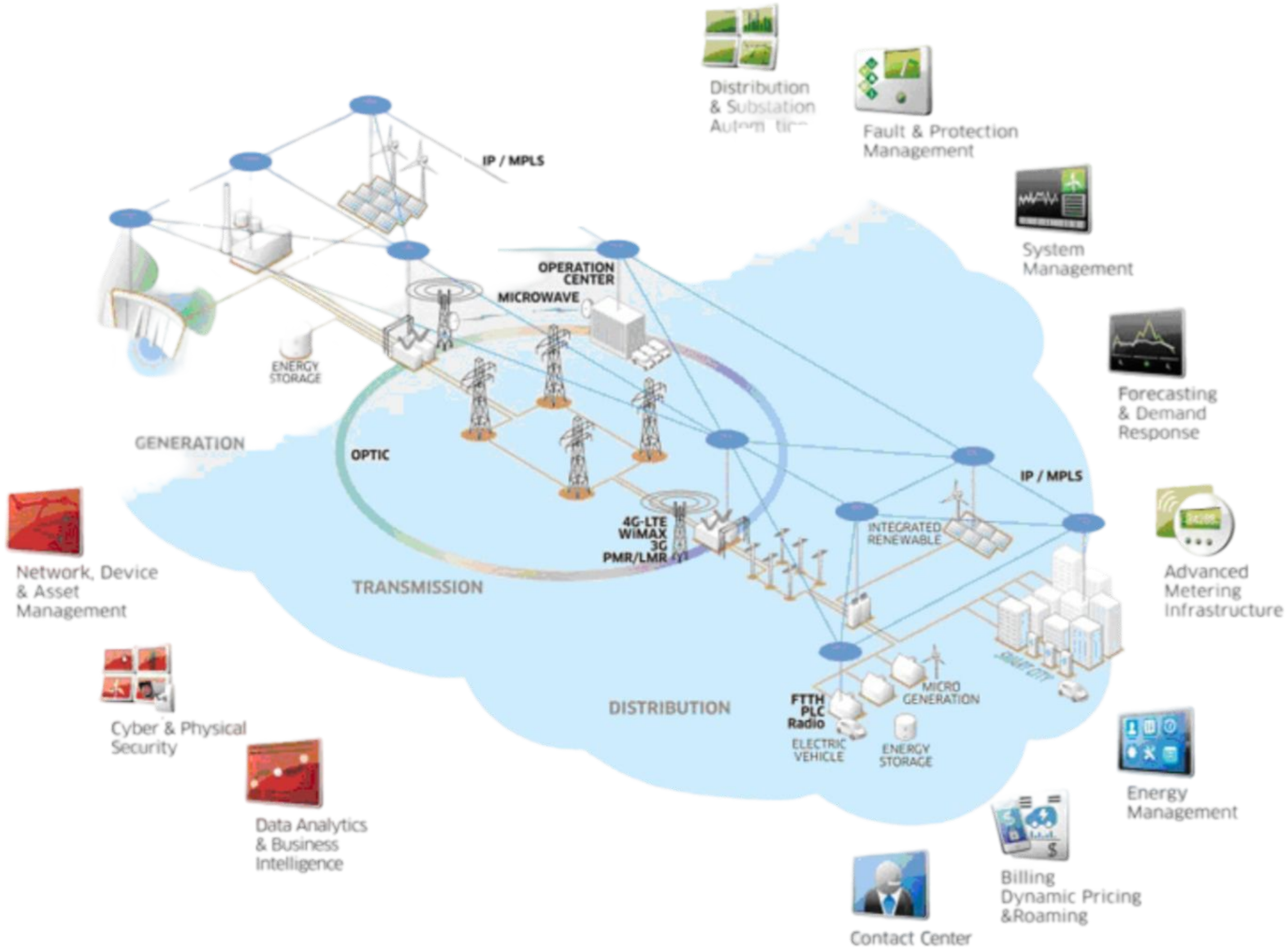
What are SCADA Systems?

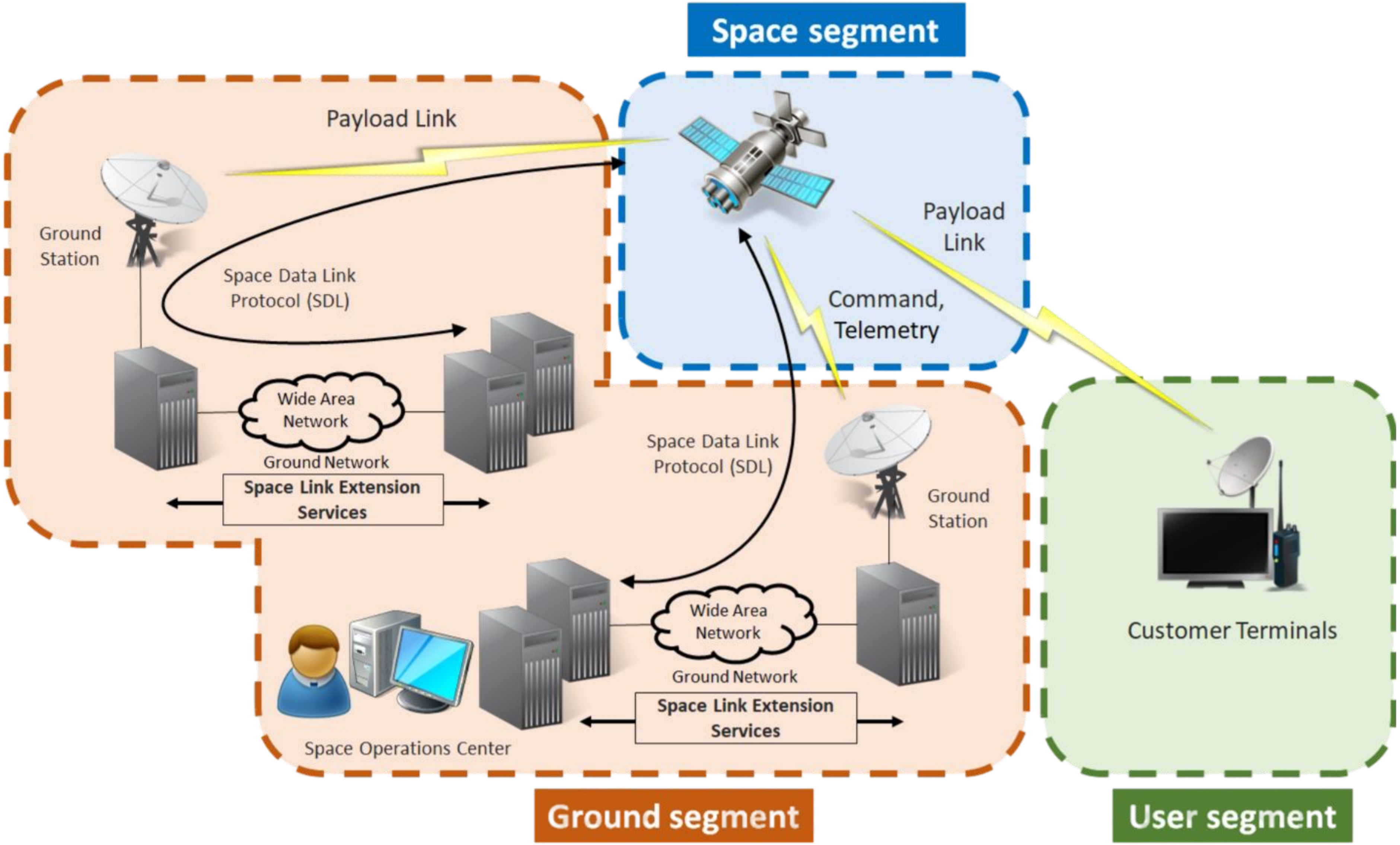
- **SCADA (Supervisory Control and Data Acquisition):** A system used for monitoring, controlling, and analyzing industrial processes in real-time across multiple locations using **M2M (Machine-to-Machine)** communication with field devices.
- **Features:** Provides centralized data visualization, alarms, and control, enabling operators to monitor trends, detect faults, and make adjustments remotely. SCADA systems integrate with **HMI (Human-Machine Interface)** software for data interpretation and support protocol standards.
- **Difference from PLCs:** While **PLCs** handle direct, local control and automation of equipment via M2M communication, SCADA systems act as a supervisory layer, aggregating data from multiple PLCs, analyzing performance, and providing remote monitoring and control over distributed systems.

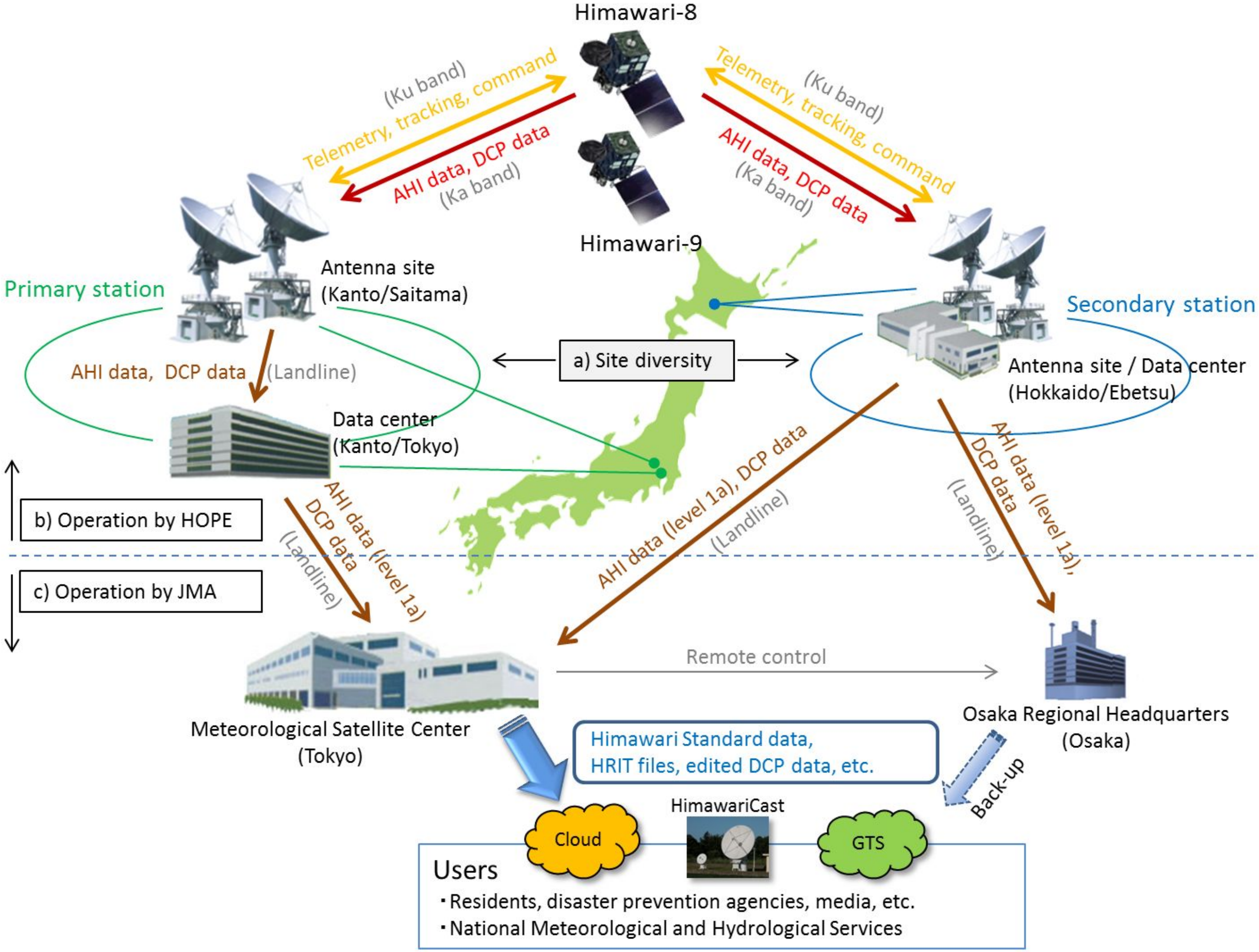

SCADA Programming
Human Machine Interface
(HMI)
Data and Control
PLCs
Equipment

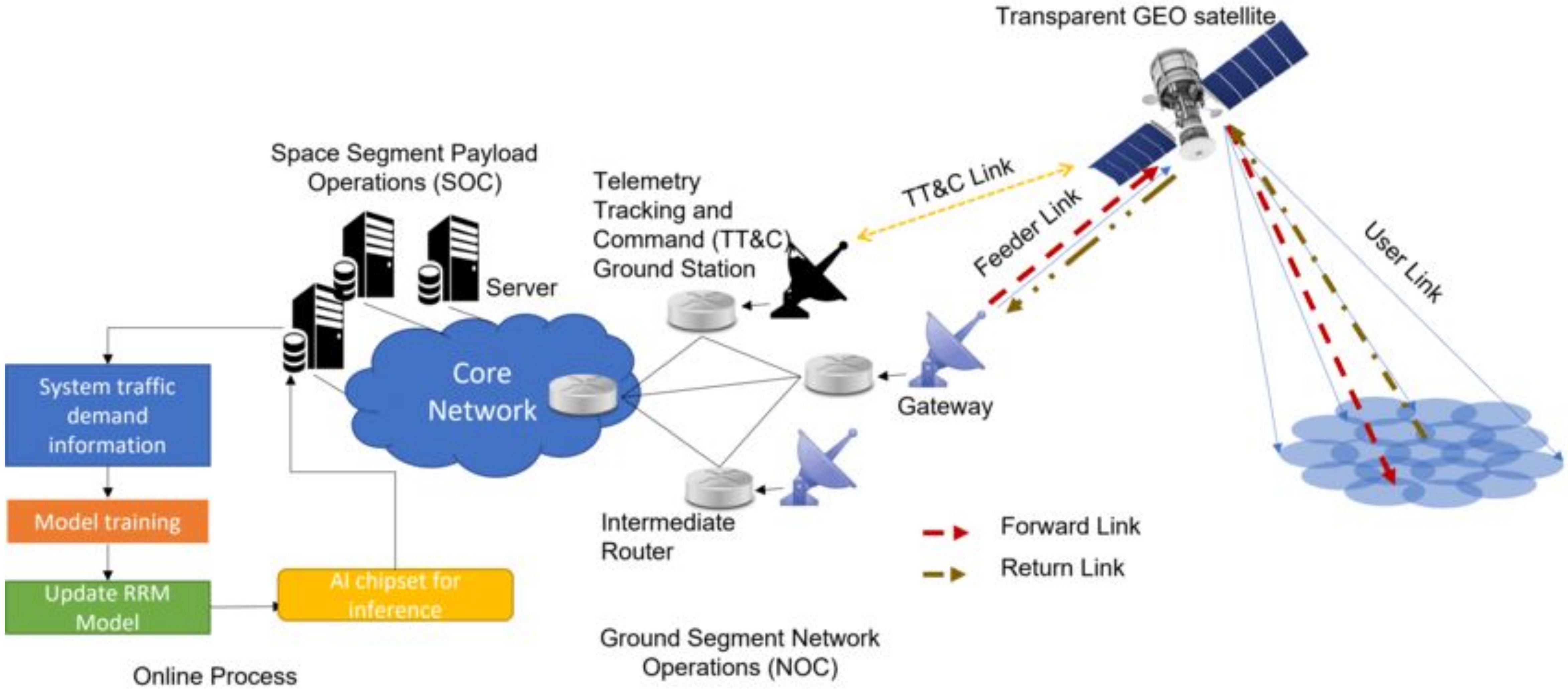

SCADA Master Station/Control Center
Comm. Links
Remote Substation
1200 bps +
(down to 300 bps in
actual installations)
Radio
Microwave
Spread-spectrum
Twisted-pair
Fiber-optics
Dial-up
Leased line
SCADA
Master
External
Control
Points
Remote
Terminal
Unit
(RTU)
Intelligent Electronic Devices
Actuator
Meter
Accumulator
Programmable Logic Controller (PLC)


Sports facility
IoT wireless unit
IoT
BLE communication
Water filter
Residual chlorine
(continuous measurement)
Swimming pool
WATER it
Residual chlorine

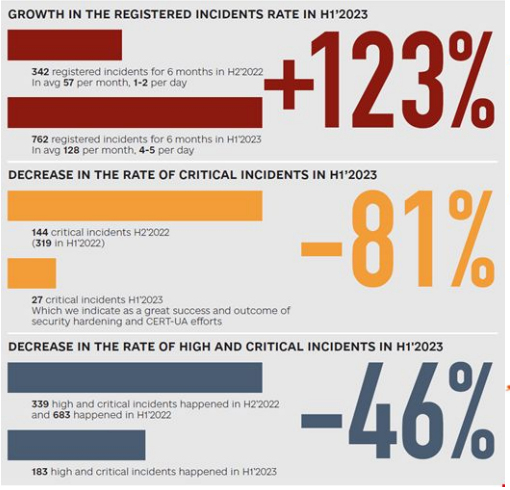

Distribution
& Substation
Fault & Protection
Management
System
Management
Forecasting
& Demand
Response
Advanced
Metering
Infrastructure
Energy
Management
Billing
Dynamic Pricing
&Roaming
Contact Center
Data Analytics
& Business
Intelligence
Cyber & Physical
Security
Network, Device
& Asset
Management
IP / MPLS
OPERATION
CENTER
MICROWAVE
ENERGY
STORAGE
GENERATION
OPTIC
4G-LTE
WiMAX
3G
PMR/LMR
INTEGRATED
RENEWABLE
IP / MPLS
TRANSMISSION
DISTRIBUTION
FTTH
PLC
Radio
ELECTRIC
VEHICLE
ENERGY
STORAGE
MICRO
GENERATION

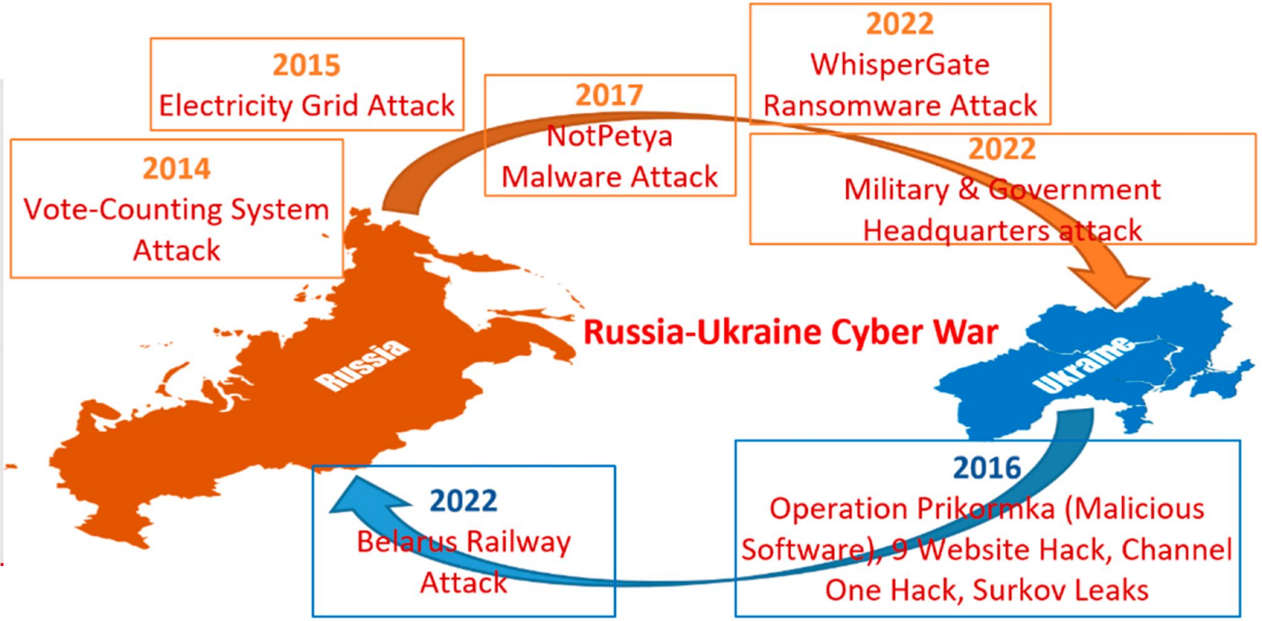

Space segment
Payload Link
Ground Station
Space Data Link Protocol (SDL)
Payload Link
Command, Telemetry
Wide Area Network
Ground Network
Space Link Extension Services
Space Data Link Protocol (SDL)
Ground Station
Customer Terminals
Wide Area Network
Ground Network
Space Link Extension Services
Space Operations Center
Ground segment
User segment

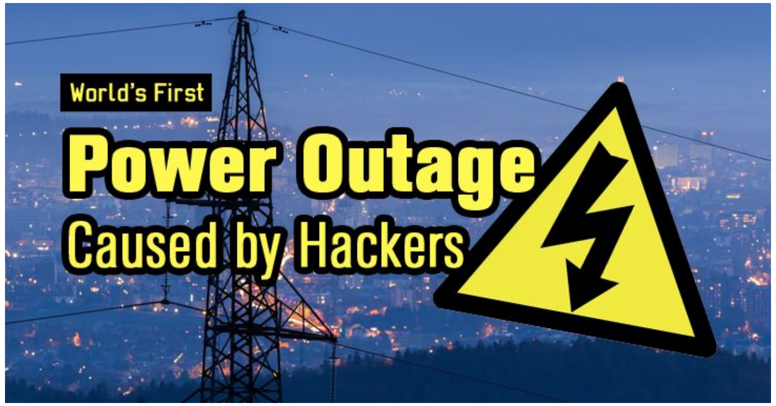

Himawari-8
Himawari-9
(Ku band)
Telemetry, tracking, command
AHI data, DCP data
(Ka band)
Primary station
Antenna site
(Kanto/Saitama)
AHI data, DCP data
(Landline)
Data center
(Kanto/Tokyo)
a) Site diversity
Secondary station
Antenna site / Data center
(Hokkaido/Ebetsu)
b) Operation by HOPE
c) Operation by JMA
AHI data (level 1a),
DCP data
AHI data (level 1a), DCP data
Remote control
Meteorological Satellite Center
(Tokyo)
Osaka Regional Headquarters
(Osaka)
Himawari Standard data,
HRIT files, edited DCP data, etc.
Back-up
Cloud
HimawariCast
GTS
Users
・Residents, disaster prevention agencies, media, etc.
・National Meteorological and Hydrological Services

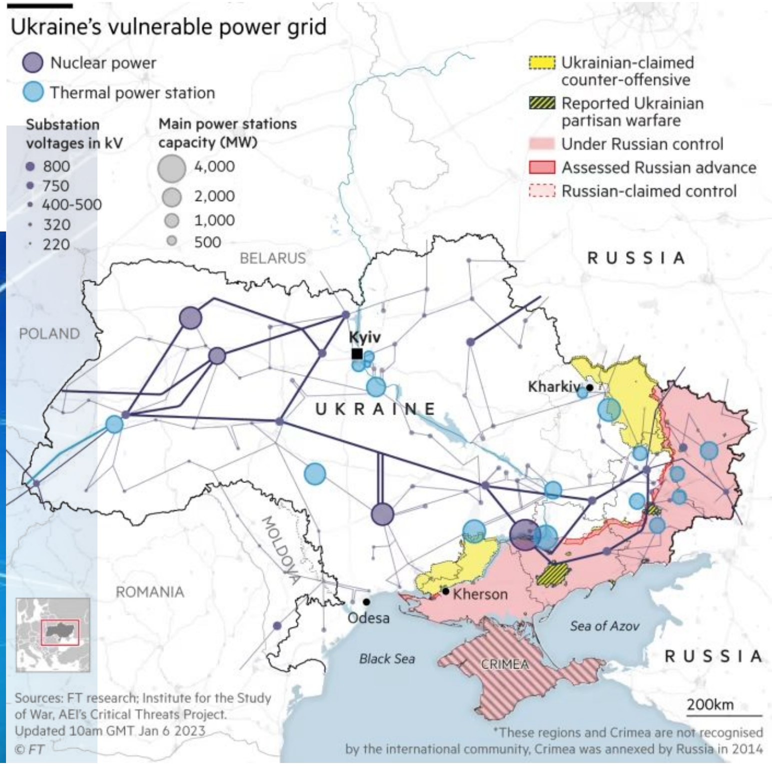

Transparent GEO satellite
Space Segment Payload Operations (SOC)
Server
Telemetry Tracking and Command (TT&C) Ground Station
TT&C Link
Feeder Link
User Link
Core Network
Gateway
Intermediate Router
System traffic demand information
Model training
Update RRM Model
AI chipset for inference
Online Process
Forward Link
Return Link
Ground Segment Network Operations (NOC)

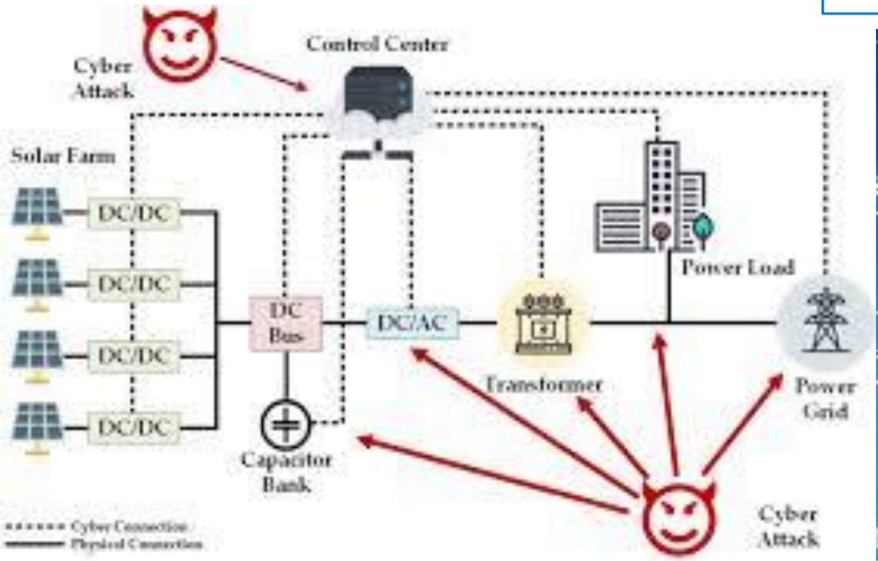
SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING











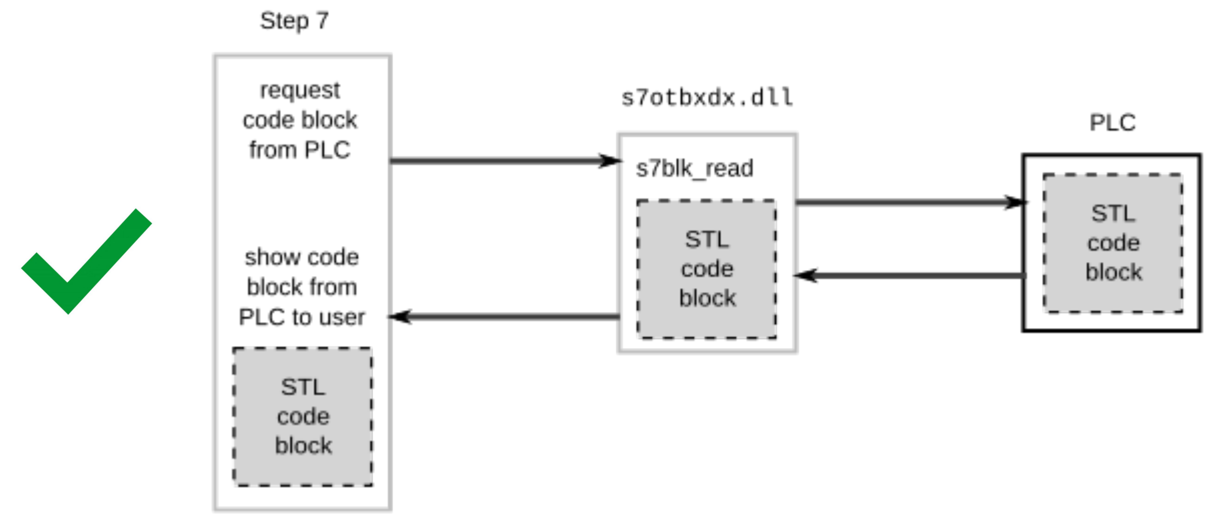
Some Vulnerabilities

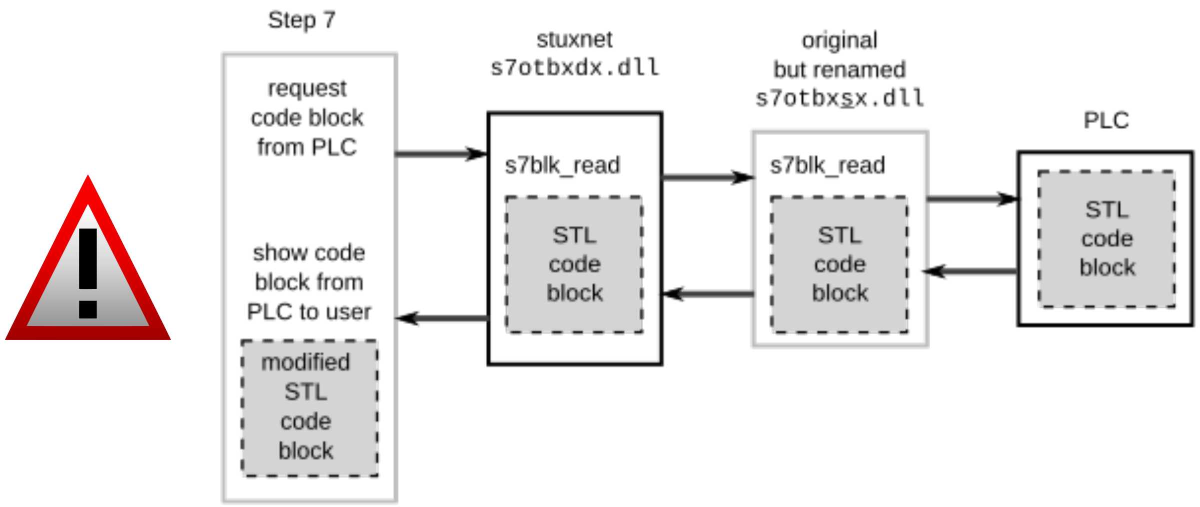
- **Static Trust Models:** Many SCADA systems rely on static trust relationships, which cannot adapt to evolving threats or compromised devices.
- **Lack of Authentication:** Devices may accept commands or data without verifying their source, increasing susceptibility to attacks like spoofing or command injection.
- **Unencrypted Communication:** Insecure communication protocols like Modbus allow attackers to intercept or alter data in transit and break trust between devices.
- **Behavioral Anomalies:** Without real-time trust monitoring, abnormal device behaviors perhaps something like an unexpected command frequency may go unnoticed.
- **Limited Privacy Protections:** Sensitive operational data transmitted between SCADA components is often unprotected, risking exposure during cyberattacks.

Stuxnet

Breaking Device Trust: Stuxnet targeted trust relationships by injecting malicious code into Siemens PLCs. The PLCs executed this code while appearing normal to operators.

Manipulating Behavior: The worm altered centrifuge operations, causing physical damage while feeding false data to SCADA systems to maintain the illusion of normalcy.





Common Attacks in SCADA Systems

- **On-Off Attack:** Alternating good and bad behavior to maintain trust above the threshold (intermittently sending false data from sensors to manipulate load-shedding strategies or fault detection in power grids).
- **Conflicting Behavior Attack:** Acting inconsistently toward different groups (a compromised IED sending accurate data to one SCADA controller while misleading another, disrupting synchronization and fault recovery).
- **Sybil Attack:** Creating multiple fake identities to influence decisions or avoid detection (introducing fake sensor nodes to flood SCADA systems with false readings, causing resource mismanagement).
- **Meter Tampering:** Manipulating smart meter data to reduce power bills or inject false usage patterns (affecting billing accuracy, load forecasting, and demand-response programs).

Other Attacks in SCADA Systems

- **Replay Attack:** Intercepting and replaying legitimate commands or data (replaying a command to open or close circuit breakers, leading to unsafe grid conditions or equipment damage).
- **Man-in-the-Middle Attack (MITM):** Intercepting and altering communication between SCADA components (altering voltage regulation commands from SCADA HMI to substations, compromising fault-clearing).
- **Denial of Service (DoS) Attack:** Overwhelming SCADA servers or networks (targeting substation communication networks to delay critical operations like circuit breaker activations).
- **Data Injection Attack:** Injecting false data into the SCADA network to manipulate behavior (falsified transformer load data triggering whatever they want).

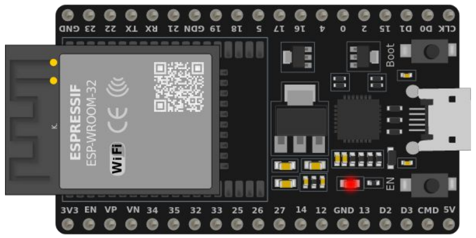
Current Research Gaps

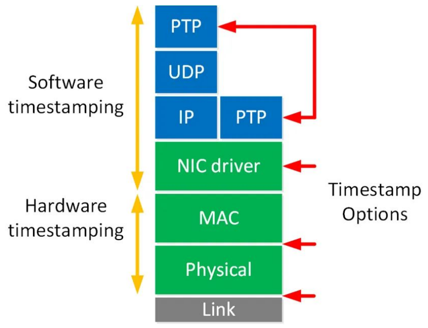
- **Reliance on Static Models:** Many trust models depend on pre-defined parameters, lacking the adaptability to respond to dynamic changes in device or network behavior.
- **Inability to Address Insider Threats:** Static models are not context-aware or flexible, making them ineffective against insider threats and novel and evolving attacks.
- **Challenges in Heterogeneous Environments:** Propagating trust relationships across diverse devices (sensors, meters, control systems) and networks remains difficult, leading to inconsistencies and reduced accuracy.

Constraints

- **Trust computation:** in smart grids is resource-intensive, which is problematic given the limited computational capacity of many grid components (smart meters, sensors).
- **Propagation delays:** could impact real-time responsiveness, especially in systems requiring instantaneous decisions (fault protection).
- **Lack of standardized:** frameworks for trust management, which leads to fragmented and inconsistent implementations across industries.

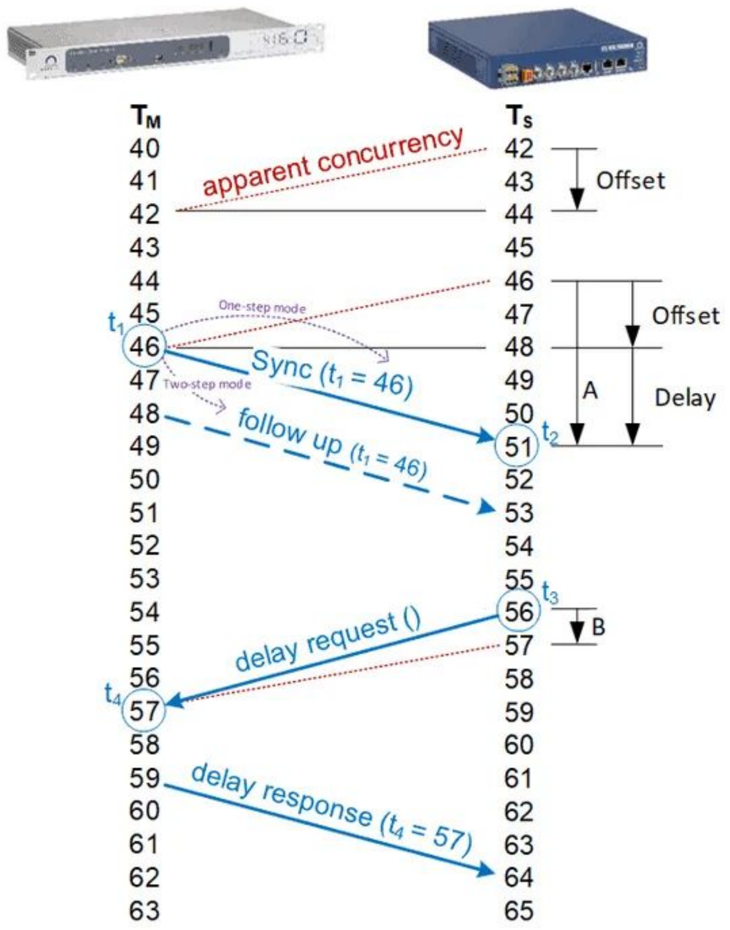
freeRTOS





Timing in Low-Power Agents

- **Timing Mismatches:** Low-power components running trust algorithms may lag, missing critical events in real-time protocols which require nanosecond-level precision.
- **Impact on GOOSE Protocol:** Processing delays in trust decisions could prevent timely circuit breaker activation during faults risking equipment damage or cascading failures.
- **Precision Timing and Time to Wake (TWT):** Trust computation lag or delays in wake timing disrupt synchronization in protocols like PTP leading to instability across substations.



Offset and delay calculations

Theory:

$$A = t_2 - t_1 = Delay + Offset$$
$$B = t_4 - t_3 = Delay - Offset$$

$$Delay = \frac{A + B}{2}$$

$$Offset = \frac{A - B}{2}$$

Example:

$$A = 51 - 46 = 5$$
$$B = 57 - 56 = 1$$

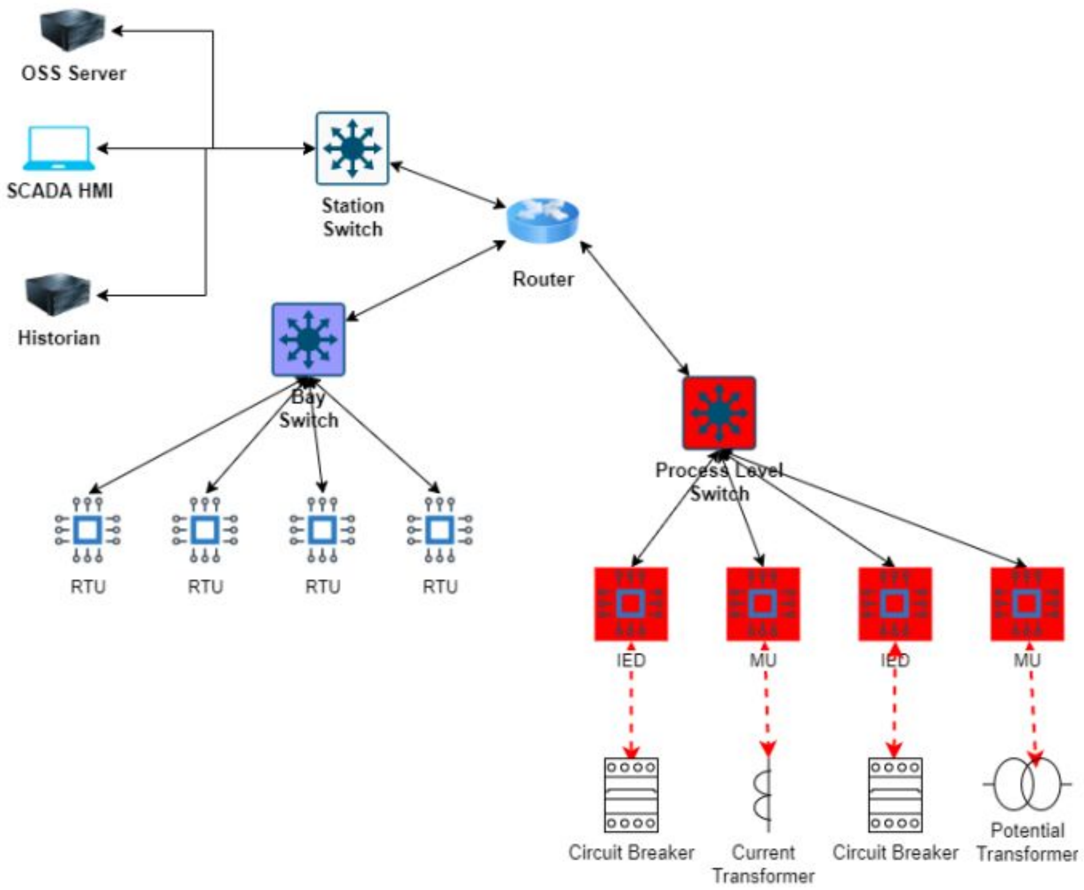
$$Delay = \frac{5 + 1}{2} = 3$$

$$Offset = \frac{5 - 1}{2} = 2$$

State of the Art Overview

- **Agent Selection in Subnets:** Most researchers agree that the critical challenge in improving privacy and trust models in SCADA systems lies in determining what constitutes an "agent" within a subnet. Treating each sensor as an agent increases accuracy but introduces significant computational overhead and downtime, whereas treating larger subnets as an agent reduces latency but sacrifices precision.
- **Lack of Real-World Testing:** Few systems are tested under real-world trust-related threats, leaving vulnerabilities unaddressed. Trust computation often relies solely on immediate parameters, overlooking historical data that could reveal deeper malicious patterns.
- **Fragmented Trust Management:** The absence of standardized frameworks for trust management in especially wireless components results in fragmented and inconsistent implementations across industries and hardware specifications.
- **Lack of Real Data:** Most PLC logs are proprietary. Many researches have resorted to hardware setups to simulate small SCADA systems in a lab or others generate artificial data. This is a large research bottleneck.

"A Trust-Influenced Smart Grid: A Survey and a Proposal"



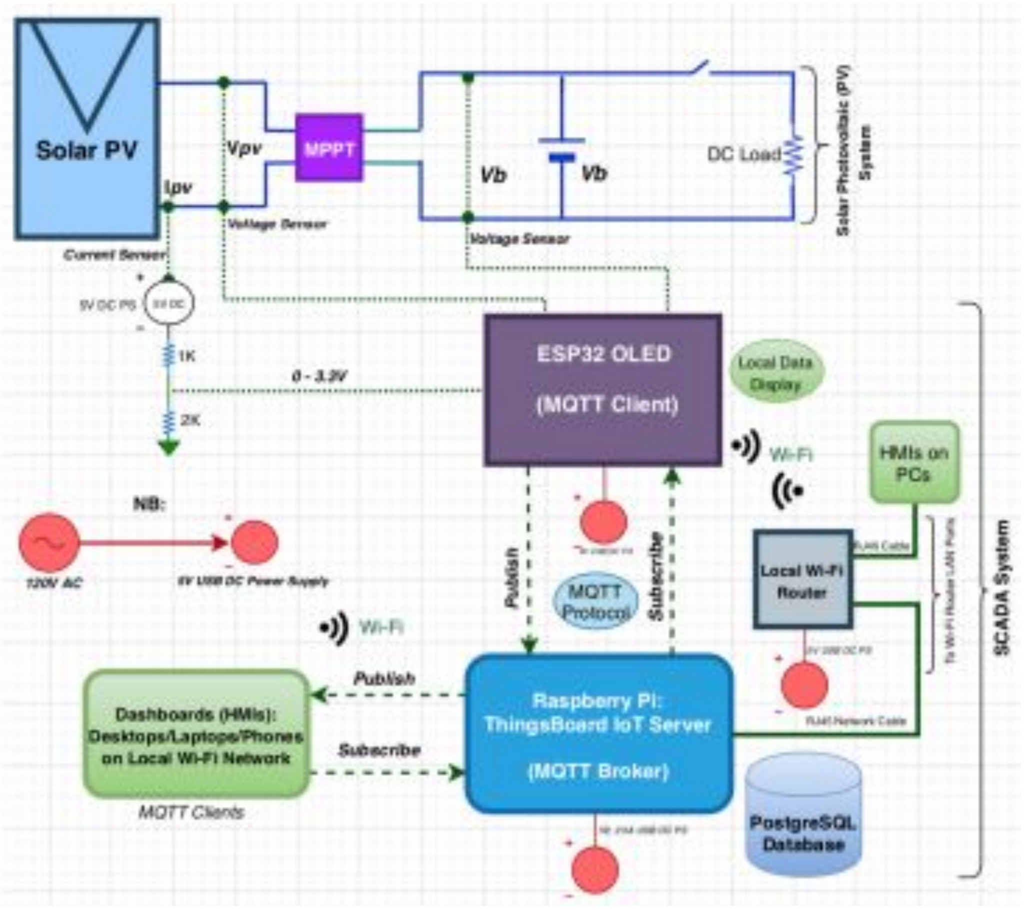
(Research Paper Reference #1)

- **System Design:** The proposed system models trust in substation automation systems (SAS) using multiple hierarchical levels: the station, bay, and process levels. Data from Intelligent Electronic Devices (IEDs) and SCADA logs are used to compute trust values for devices and commands.
- **Expanded Model:** Trust computation includes direct and indirect trust, incorporating a **familiarity score** that evaluates device behavior based on consistency, exposure frequency, and communication history. However, risk was not quantified, and some attacks (on-off) could evade detection due to static thresholds.
- **Testing and Conclusions:** Trust models were simulated using machine learning and fuzzy logic to identify threats within substation communications. Despite promising results the lack of real-world testing and unaddressed risk components limit defense against sophisticated attacks.

Boakye-Boateng, K., Ghorbani, A., & Lashkari, A. (2022). A Trust-Influenced Smart Grid: A Survey and a Proposal. *J. Sens. Actuator Networks*, 11, 34. https://doi.org/10.3390/jsan11030034

"IoT-Based SCADA System Design"

- **Low-Cost Architecture:** The system uses an ESP32 microcontroller (MQTT client) and a Raspberry Pi (MQTT broker) to create a lightweight, scalable, and open-source SCADA framework.
- **Real-Time Monitoring:** Data from sensors (voltage, current) is published and subscribed via MQTT, with visualization handled through dashboards and local HMIs.
- **Edge-Level Processing:** By placing monitoring and processing algorithms closer to edge devices (ESP32) latency is reduced.



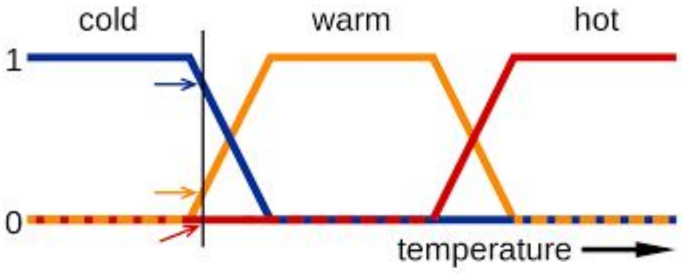
Aghenta, L., & Iqbal, M. (2019). Design and implementation of a low-cost, open source IoT-based SCADA system using ESP32 with OLED, ThingsBoard, and MQTT protocol. *AIMS Electronics and Electrical Engineering*. https://doi.org/10.3934/electreng.2020.1.57

(Research Paper Reference #2)

Proposed Solution

Tools:

- **Datasets for Network Modeling:** Cisco data on wireless networks of computing hosts for simulation and testing of trust algorithms, with ground truth options included (Madani et al., 2022). Perfect considering most SCADA systems are wireless.
- **Fuzzy Logic Frameworks:** Xfuzzy 3.5 enables design, verification, and implementation of fuzzy logic systems for trust modeling and decision-making.
- **Python Libraries:** scikit-fuzzy, NumPy, and Pandas are used for fuzzy logic algorithms data processing in trust simulations.



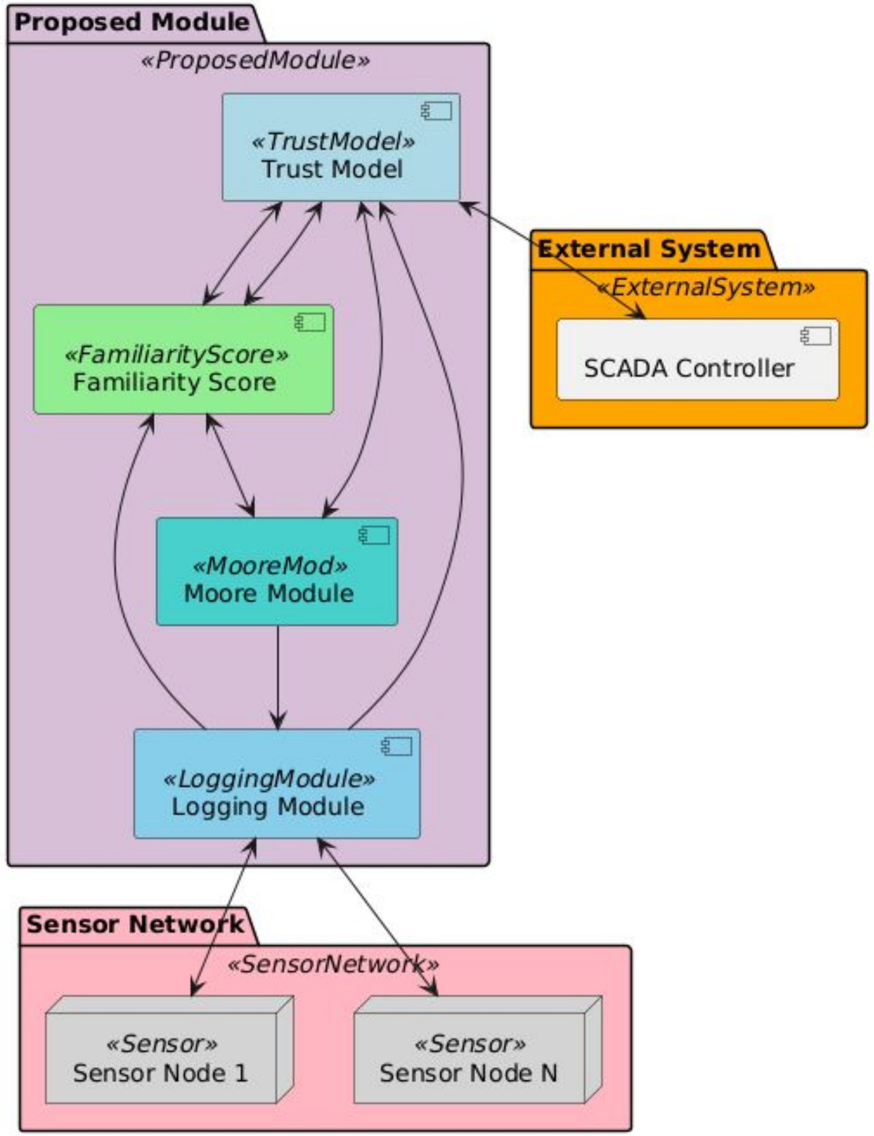
References:

Madani, O., Averineni, S. A., & Gandham, S. (2022). *A Dataset of Networks of Computing Hosts. Proceedings of the 2022 ACM on International Workshop on Security and Privacy Analytics*, 100-104. https://snap.stanford.edu/data/cisco-networks.html.

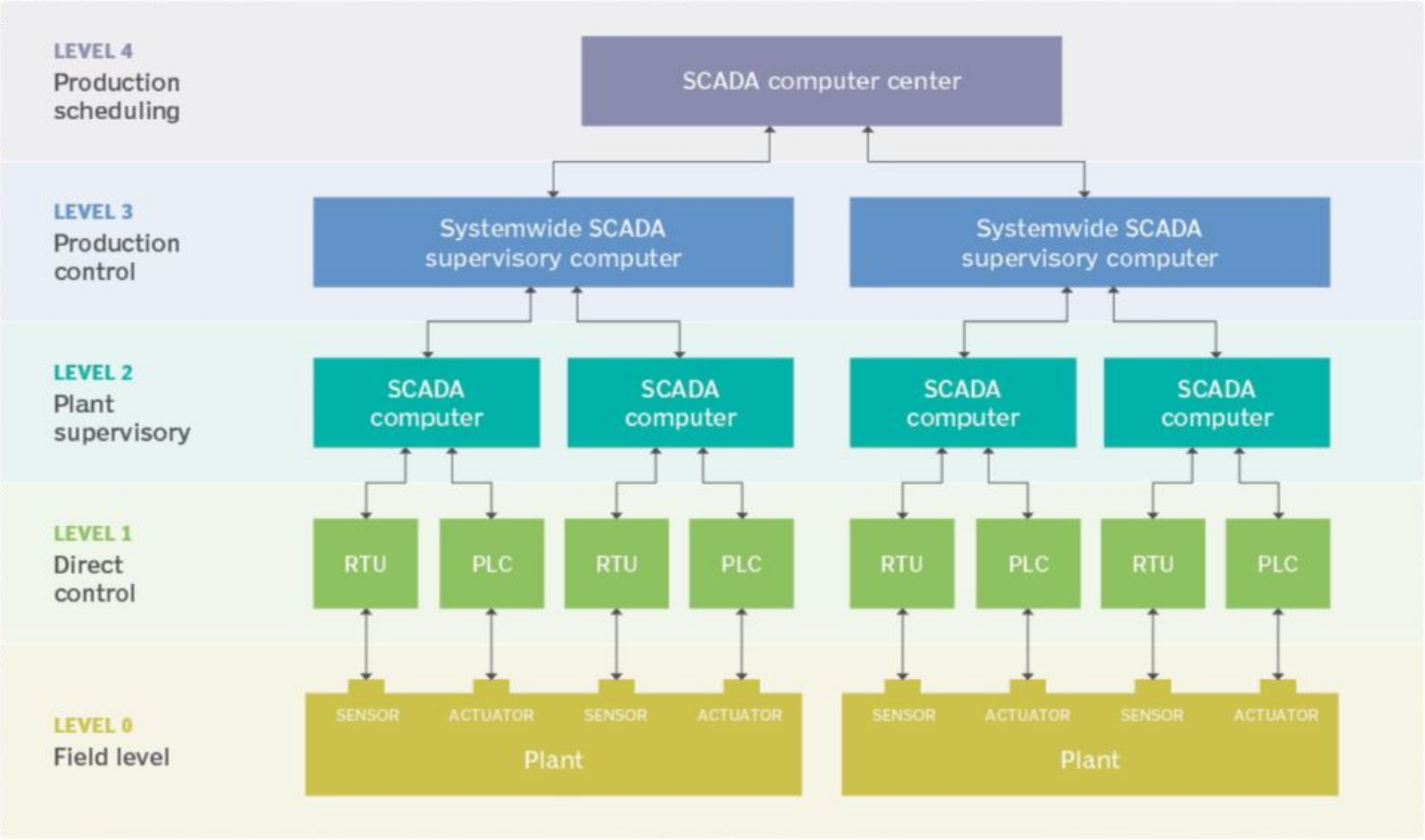
Proposed Solution

Refined Fuzzy and Familiarity Scores: Optimize a fuzzy logic-based trust evaluation model for a **sweet spot in PLC subnet size and communication patterns**.

- **Enhanced Trust Evaluation Accuracy:** Extract communication features (message intervals, frequency, similarity patterns) and combining familiarity metrics and temporal behavior analysis into fuzzy logic.
- **Optimized Computational Efficiency for SCADA:** Modular with lightweight calculations so the system remains scalable and performant for real-time evaluations.
- **Adjusting definition of Agents:** Find ratios and patterns of fuzzy logic accuracy alongside ideal subnet or agent size.
- **Modular Adjustments of Parameters:** Find the best parameters from the dataset to detect anomalies with the lowest number of false positives.

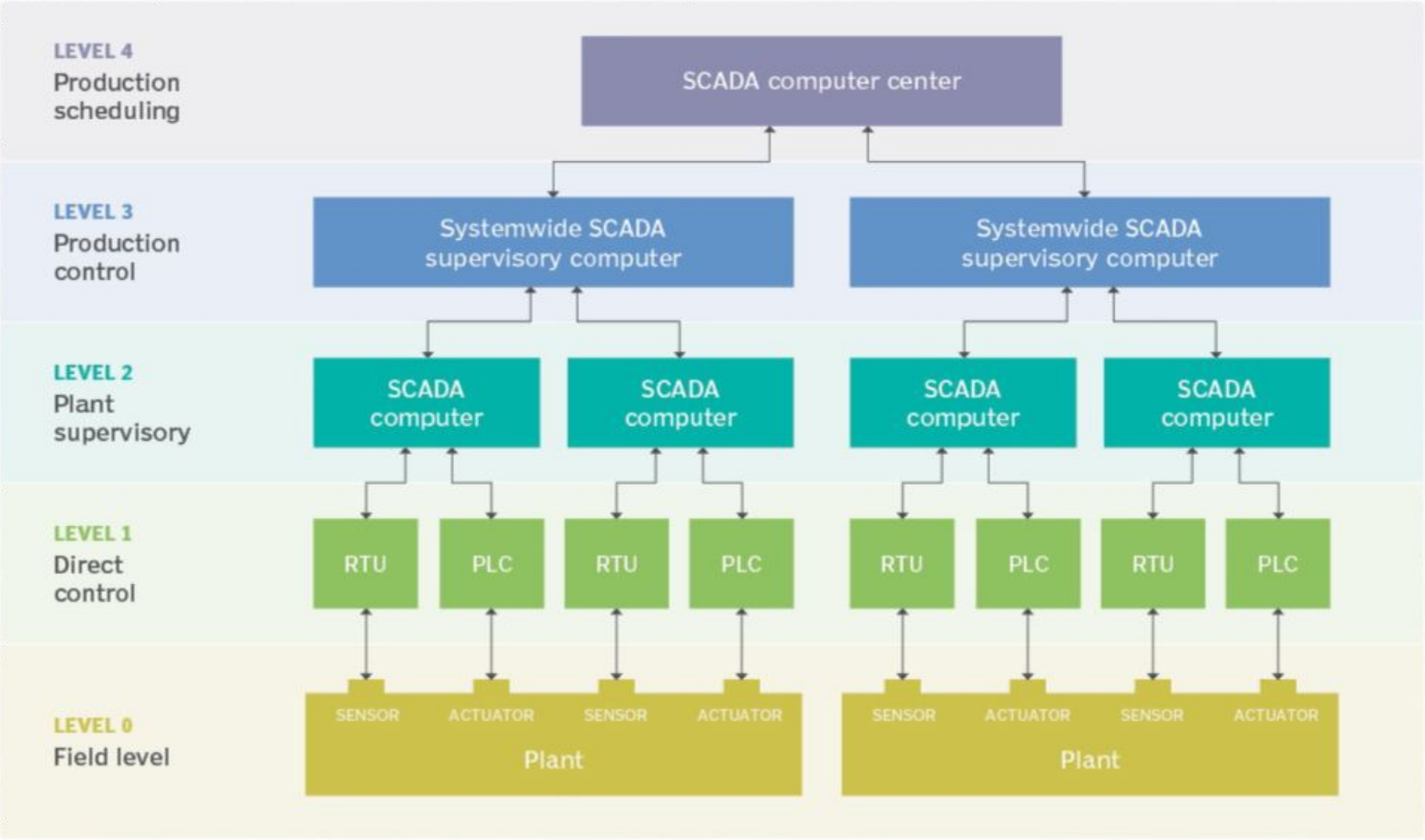


Layers of the SCADA system architecture



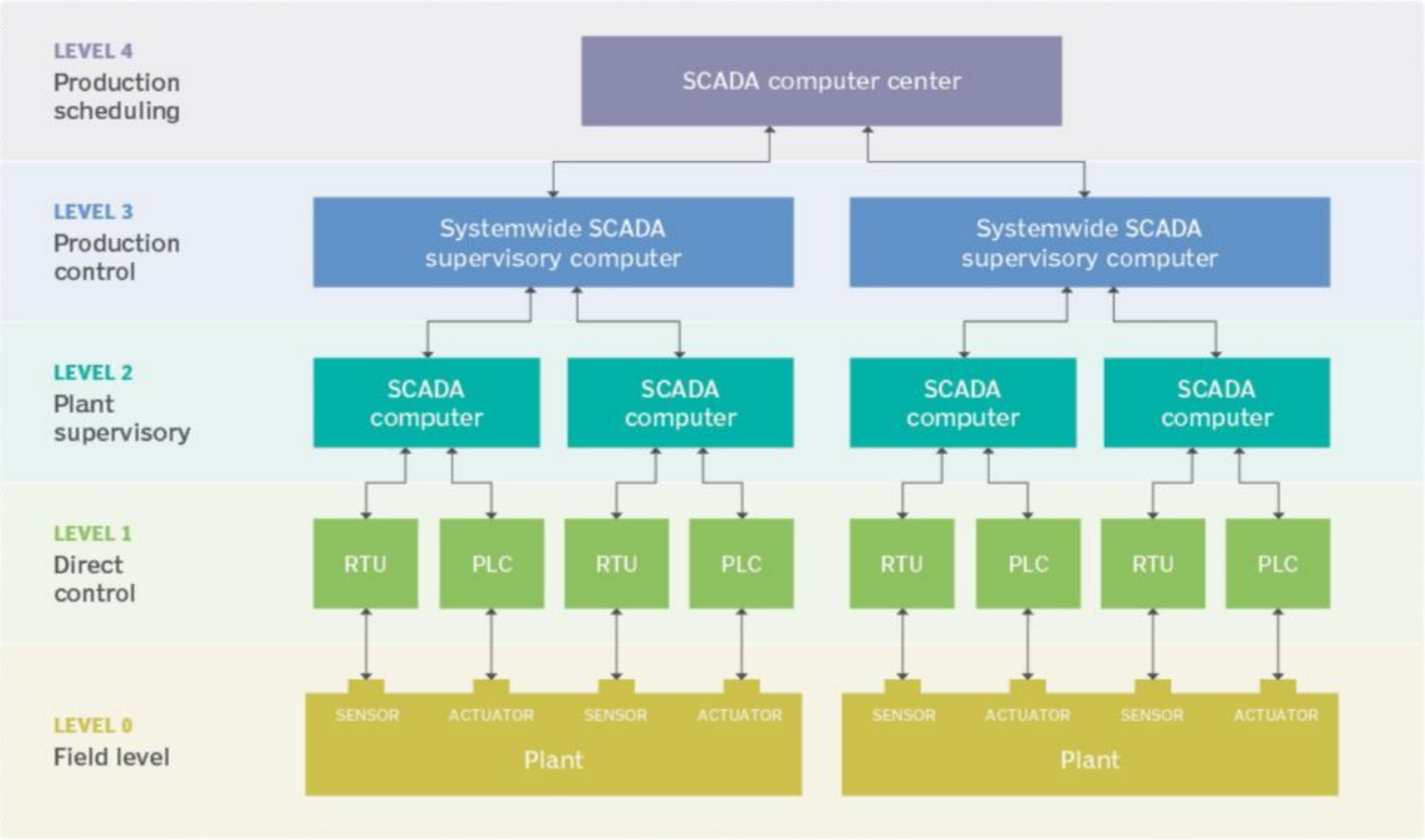
Layers of the SCADA system architecture

Trust Model Layer 3



Layers of the SCADA system architecture

Trust Model Layer 1.5



- **Dynamic Trust Scoring:** Compute trust scores (E_i, E_f, E_s) dynamically using fuzzy logic, based on:
  - Familiarity metrics (F) from communication features (e.g., message intervals ζ_{qq}, ζ_{qr}).
  - Temporal behavior metrics (T) from Moore machine states.
- **Trust Score Formula:**

$$\text{Trust Score} = \mu_E(E_i) + \mu_F(E_f) + \mu_S(E_s)$$

  where μ_E, μ_F, and μ_S are fuzzy membership functions for intensity, frequency, and similarity exposure.
- **Enhanced Temporal Behavior Analysis:** The Moore machine captures sequential patterns of communication (e.g., ζ_{qq}, ζ_{qr}) and deviations (ζ_{to}), providing a predictive dimension to trust modeling.
- **Final Trust Evaluation:**

$$\text{Final Trust} = w_1 \cdot \text{FuzzyLogic}(F) + w_2 \cdot \text{MooreLogic}(T)$$

  where w_1, w_2 balance real-time efficiency and accuracy.
- **Expected Output:** Real-time trust values (Trust Score $\in [0,1]$), anomaly alerts, and organized logs sent to the SCADA system for actionable insights.

Example: Anomaly Detection with Moore Machine in a Power Grid

- **State Representation:** The Moore machine tracks states based on historical message intervals:

$$\rho = \{\rho_{\text{normal}}, \rho_{\text{warning}}, \rho_{\text{anomaly}}\}$$

  Where:

  - ρ_{normal}: Represents normal operation with $\zeta_{qq} \in [450, 550]$ hours.
  - ρ_{warning}: Represents minor deviations with $\zeta_{qq} \in [300, 450] \cup [550, 700]$ hours.
  - ρ_{anomaly}: Represents significant deviations with $\zeta_{qq} < 300$ or $\zeta_{qq} > 700$ hours.

- **Transition Function:** Transitions between states occur based on observed intervals (ζ_{qq}) and thresholds:

$$\delta(\rho, \sigma) \to \rho'$$

  For an observed $\zeta_{qq} = 48$:

$$\delta(\rho_{\text{normal}}, \zeta_{qq}) = \rho_{\text{anomaly}}$$

- **Deviation Calculation:** The Moore machine uses historical averages ($\mu_{\zeta_{qq}}$) and standard deviations ($\sigma_{\zeta_{qq}}$) to compute a deviation score:

$$\text{Deviation Score} = \frac{|\zeta_{qq} - \mu_{\zeta_{qq}}|}{\sigma_{\zeta_{qq}}}$$

  Where:

$$\mu_{\zeta_{qq}} = 500, \quad \sigma_{\zeta_{qq}} = 50, \quad \zeta_{qq} = 48$$

  Substituting:

$$\text{Deviation Score} = \frac{|48 - 500|}{50} = \frac{452}{50} = 9.04$$

- **Anomaly Detection:** If Deviation Score > 3 (3 standard deviations from the mean), the event is flagged as an anomaly:

$$9.04 > 3 \quad \Rightarrow \quad \text{Anomaly Detected}$$

- **Trust Score Computation:** The Moore machine outputs a temporal metric ($T = \delta(\rho, \sigma)$), which is passed to the fuzzy logic module to compute the trust score:

$$\text{Final Trust} = w_1 \cdot \mu_E(E_i) + w_2 \cdot \mu_F(E_f) + w_3 \cdot \mu_S(E_s)$$

  Example weights and membership values:

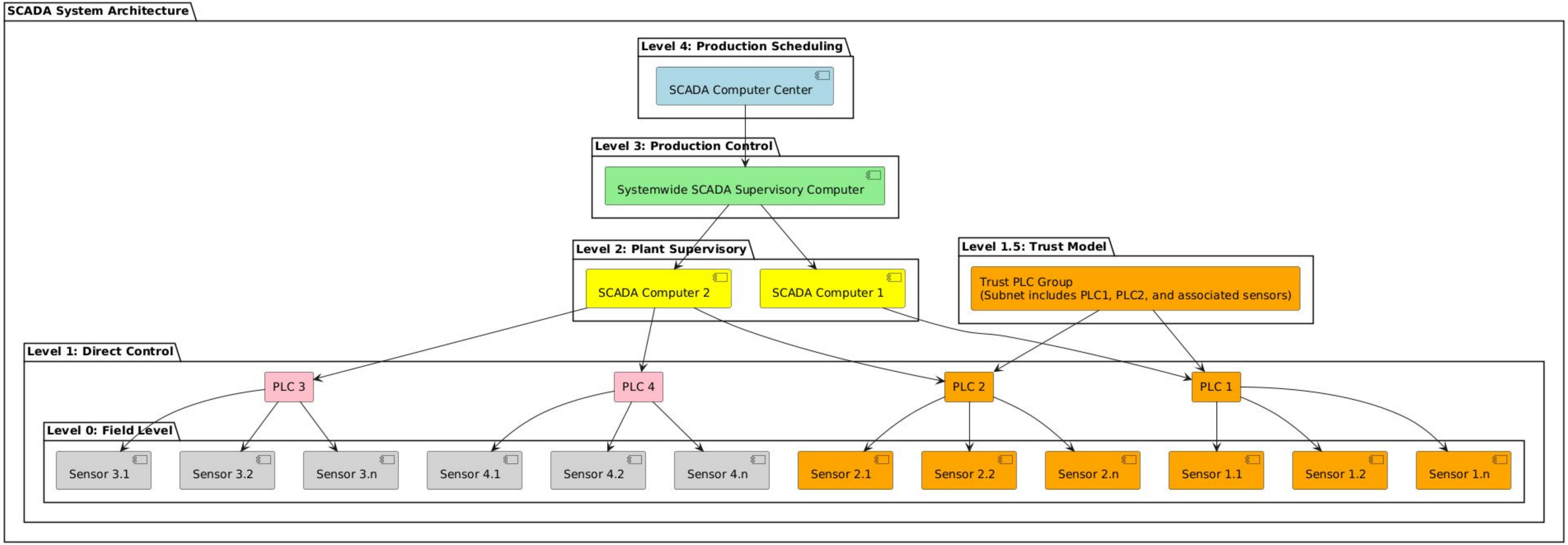
$$w_1 = 0.4, \quad w_2 = 0.3, \quad w_3 = 0.3$$

$$\mu_E(E_i) = 0.2, \quad \mu_F(E_f) = 0.1, \quad \mu_S(E_s) = 0.0$$

  Substituting:

$$\text{Final Trust} = 0.4 \cdot 0.2 + 0.3 \cdot 0.1 + 0.3 \cdot 0.0 = 0.08 + 0.03 + 0.0 = 0.11$$

- **Output to SCADA System:** The trust score, 0.11, is sent to the SCADA system for further action.


SCADA System Architecture
Level 4: Production Scheduling
SCADA Computer Center
Level 3: Production Control
Systemwide SCADA Supervisory Computer
Level 2: Plant Supervisory
SCADA Computer 2
SCADA Computer 1
Level 1.5: Trust Model
Trust PLC Group
(Subnet includes PLC1, PLC2, and associated sensors)
Level 1: Direct Control
PLC 3
PLC 4
PLC 2
PLC 1
Level 0: Field Level
Sensor 3.1
Sensor 3.2
Sensor 3.n
Sensor 4.1
Sensor 4.2
Sensor 4.n
Sensor 2.1
Sensor 2.2
Sensor 2.n
Sensor 1.1
Sensor 1.2
Sensor 1.n

Evaluation Plan

- **Comparison with Baseline Trust Models:** Compare the trust scores generated by your system with those generated by traditional fuzzy logic models or in other research. Analyze the accuracy and precision of anomaly detection under varying network conditions.
- **Error Rate Analysis in Real-World Simulations:** Use synthetic or real SCADA data to measure the false-positive and false-negative rates of anomaly detection. See how well the new model differentiates between legitimate fluctuations in communication patterns and actual threats.
- **Timing from Our Module:** The simulated SCADA top level controller will track when the request was sent and when it was received to measure speed.

```
├── README.md
├── requirements.txt
├── run.py
├── dataset/
│   ├── cisco_22_networks/
│   │   ├── dir_20_graphs/
│   │   ├── dir_g21_small_workload_with_gt/
│   │   ├── dir_g22_extra_graph_with_gt/
│   │   ├── read_graphs.py
│   │   ├── read_gt.py
│   │   └── README
│   ├── synthetic_data/
│   └── processed_data/
│
├── src/
│   ├── __init__.py
│   ├── scadacontroller.py
│   ├── trustevaluation.py
│   ├── utils.py
│   ├── simulate_attack.py
│   ├── moore.py
│   ├── familiarity.py
│   └── fuzzymodel/
│       ├── xfuzzy_interface.py
│       └── ruleset.fcl
```

Our Data Set

- **Dataset Description:** Contains 22 disjoint graphs representing network communications in distributed applications. Data includes anonymized nodes (IPs), port numbers, and detailed communication patterns over multiple time periods. Includes ground truth groupings in graphs g21 and g22 for validation.
- **Why It Works:** Provides realistic SCADA emulation with ground-truth groupings to test trust evaluation models and detect anomalies. Captures temporal communication patterns and diverse port statistics to model SCADA-like behaviors effectively. Supports temporal behavior analysis with detailed message intervals. Scales well with graphs ranging from small (52 nodes) to large (278,739 nodes).
- **Limit:** Most SCADA datasets are proprietary.

O. Madani, S. A. Averineni, S. Gandham, *A Dataset of Networks of Computing Hosts*, IWSPA 2022.

SNAP

Properties
Number of graphs: 22
Directed: Yes
Node features: No
Edge features: Yes
Graph labels: No
Temporal: Yes

Stats	Min	Max
Nodes	86	278,739
Edges	155	2,158,346

Results

- **Trust Scores Analysis:** The system effectively categorized nodes into "Normal" and "Anomalous" based on their trust scores, with normal scores averaging above 0.5 and anomalous scores typically below 0.3.
- **Anomaly Detection Performance:** The enhanced model detected an average of around 6 anomalies per test scenario, achieving a detection accuracy in the high 90% range. The use of temporal data improved the system's ability to discern anomalous patterns over time.
- **Comparison with Basic Fuzzy Logic:** Compared to the basic fuzzy logic approach, the advanced model showed a noticeable improvement in detection accuracy (over 10% on average) with a slight increase in processing time, staying within sub-millisecond performance.

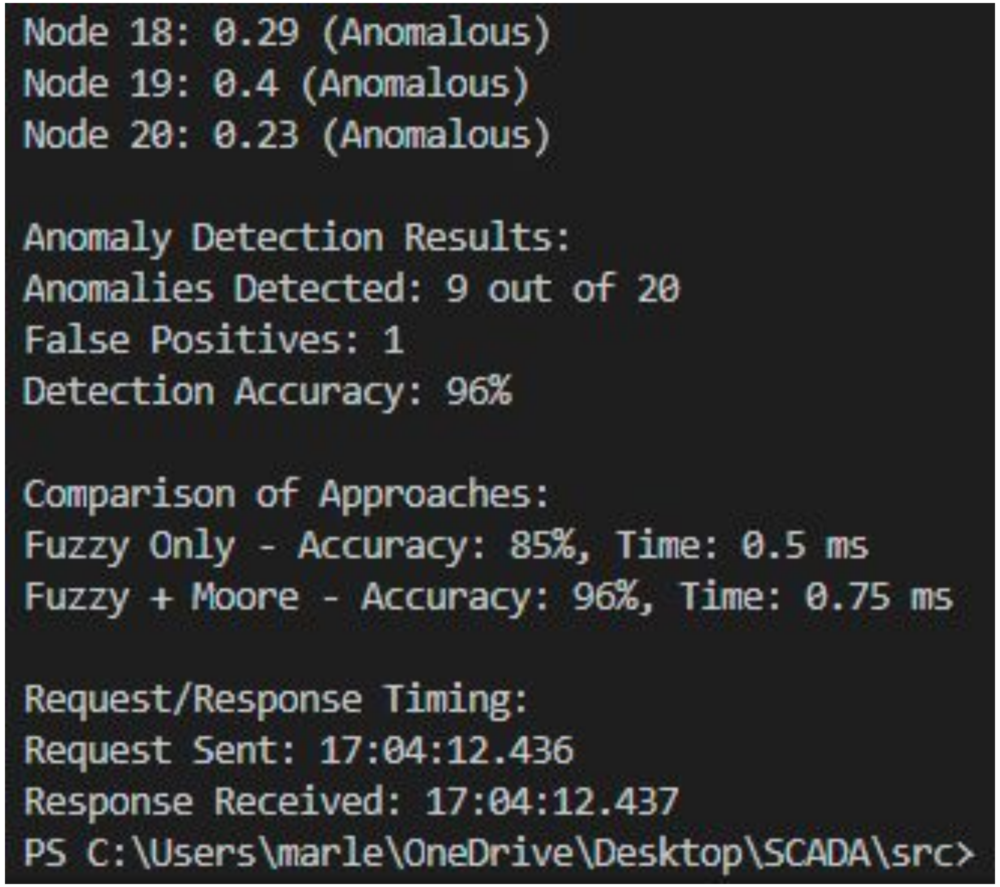
```
Anomaly Detection Results:
Anomalies Detected: 6 out of 20
False Positives: 1
Detection Accuracy: 96%

Comparison of Approaches:
Fuzzy Only - Accuracy: 85%, Time: 0.5 ms
Fuzzy + Moore - Accuracy: 96%, Time: 0.75 ms

Request/Response Timing:
Request Sent: 16:59:37.329
Response Received: 16:59:37.329
PS C:\Users\marle\OneDrive\Desktop\SCADA\src>
```

Results

- **Trust Scores**: During simulated attack conditions, the system identified more nodes as anomalous, with trust scores averaging below 0.4 for malicious nodes. This shows some effectiveness in detecting more subtle attack behaviors.
- **Improved Accuracy**: On average, the system flagged 9 to 10 anomalies per scenario, with detection accuracy improving by around 10% compared to basic fuzzy logic models. This improvement came with a marginal increase in processing time, still under 1 millisecond.
- **Baseline**: The inclusion of temporal metrics from the Moore module enhanced the system's ability to detect complex attack patterns, such as delayed responses and unusual communication intervals, which were often missed by simpler models.



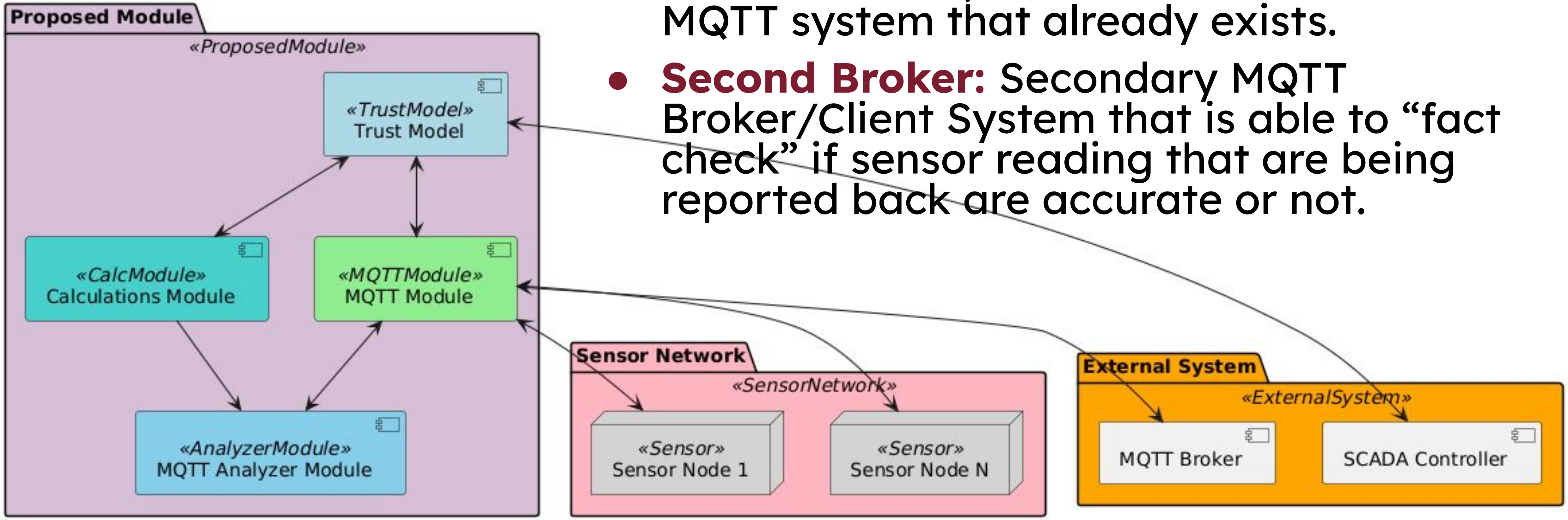
Conclusion & Takeaways

- **Results:** We were able to increase accuracy with a delay. While this delay is minor, its potential impact on other operations within the PLC, especially if the model runs on the PLC itself warrants further investigation. **We should emulate known RTOS protocols within testing.** Aim to define the delay is too high, switch back to regular fuzzy model. Verify that different attacks do act as real world attacks.
- **Scalability Challenges:** As the system grows models need to adapt to larger and more complex environments. Defining what constitutes an "agent" is a grey area, as trust scores must account for differences in behaviors and roles within distributed systems. **More tests need to be developed when applying a model to be monitoring different amounts of PLC subnets.**
- **Need more refinement in my parameters:** Keep refining membership functions, all modules of the model and experiment with different agent and subnet sizes and definitions of anomalies. Customize it to different systems, everyone is strapped for real **M2M SCADA** datasets.

Future Work and Expansion

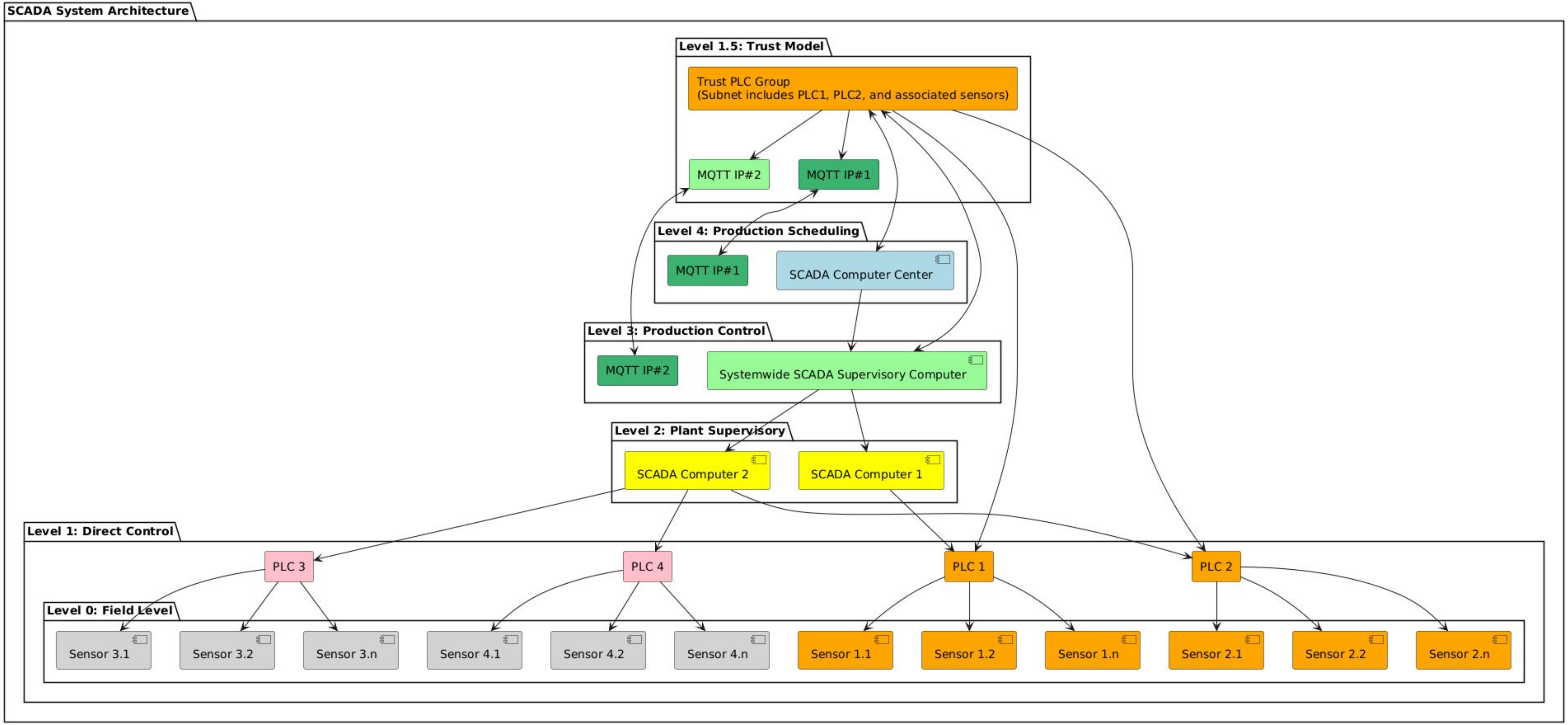
- **Standardization of Trust Evaluation Metrics:** Develop standardized benchmarks and methodologies for evaluating trust in SCADA and PLC systems. This will create consistency and comparability across different implementations.
- **Advanced Anomaly Detection:** More ML, such as neural networks or ensemble methods, to improve the detection of complex, multi-step anomalies that might not be captured by Moore models or fuzzy logic alone, if delay is small.
- **Continue to Refine Parameters:** Keep refining my suggest parameters in the familiarity score such as port hopping, timing, along with size of the subnet or agent and the familiarity and moore modules as well. Perhaps have advanced AI/ML run these simulations for us and compare all the different simulations to find the sweet spots?

An Abstract Proposed Solution



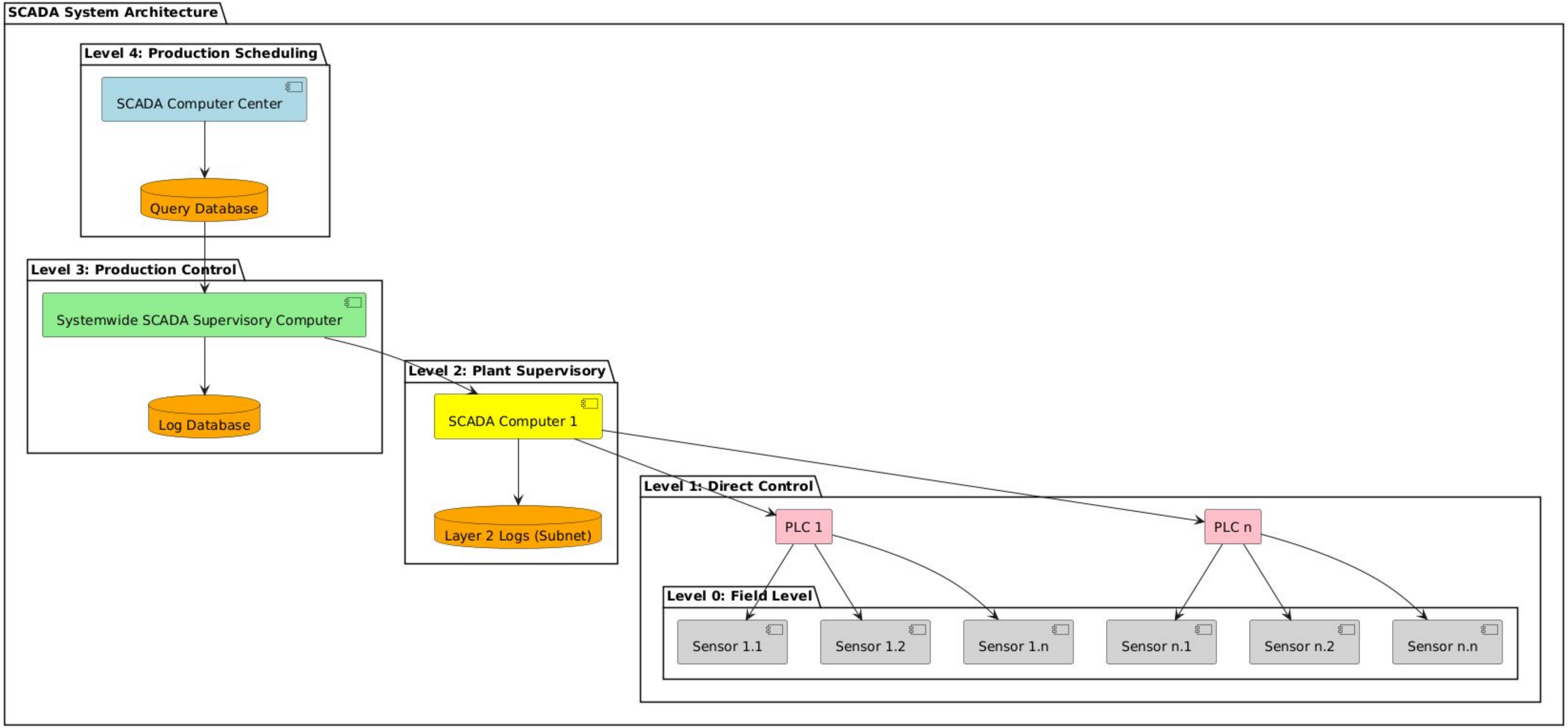
Approach #2: Cross Referencing Sensors

- **MQTT:** Have a separate MQTT module close to the sensors checking their real measurements, and check with the main MQTT system that already exists.
- **Second Broker:** Secondary MQTT Broker/Client System that is able to “fact check” if sensor reading that are being reported back are accurate or not.

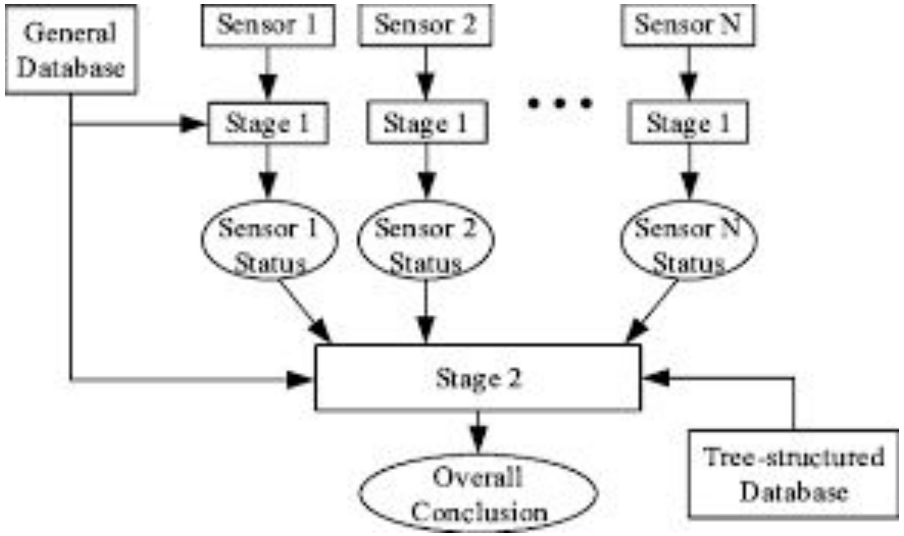

SCADA System Architecture
Level 1.5: Trust Model
Trust PLC Group
(Subnet includes PLC1, PLC2, and associated sensors)
MQTT IP#2
MQTT IP#1
Level 4: Production Scheduling
MQTT IP#1
SCADA Computer Center
Level 3: Production Control
MQTT IP#2
Systemwide SCADA Supervisory Computer
Level 2: Plant Supervisory
SCADA Computer 2
SCADA Computer 1
Level 1: Direct Control
PLC 3
PLC 4
PLC 1
PLC 2
Level 0: Field Level
Sensor 3.1
Sensor 3.2
Sensor 3.n
Sensor 4.1
Sensor 4.2
Sensor 4.n
Sensor 1.1
Sensor 1.2
Sensor 1.n
Sensor 2.1
Sensor 2.2
Sensor 2.n

Additional Abstract Proposed Exploration

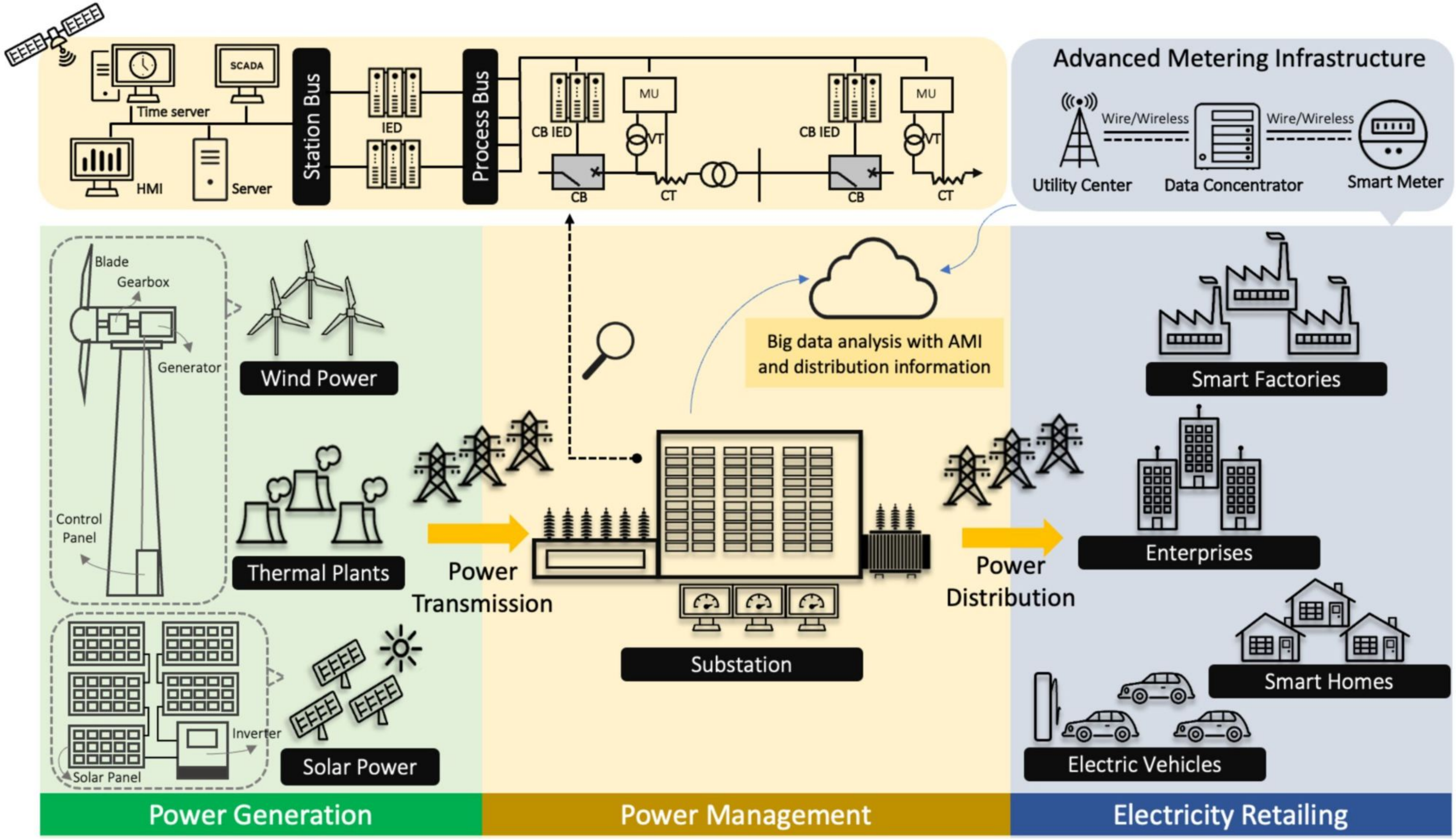
- **Implementing K-Anonymity in M2M Communication:** Use k-anonymity principles to obscure exact machine-to-machine (M2M) communication data in layered SCADA systems. Mask sensitive identifiers or granular details to prevent attackers from pinpointing specific operations or devices.
- **Layered Data Aggregation for SCADA Systems:** Aggregate machine logs and system events at higher SCADA levels (Layer 3 or 4) before exposing data to external queries. Generalized and anonymized information minimizes the risk of attackers learning critical system behaviors.
- **Dynamic Query Filters and Noise Injection:** Introduce dynamic query filters and controlled noise injection for real-time queries on Layer 3 logs by Layer 4 systems. This ensures sensitive data patterns remain protected, making it harder for attackers to analyze and exploit system behavior.

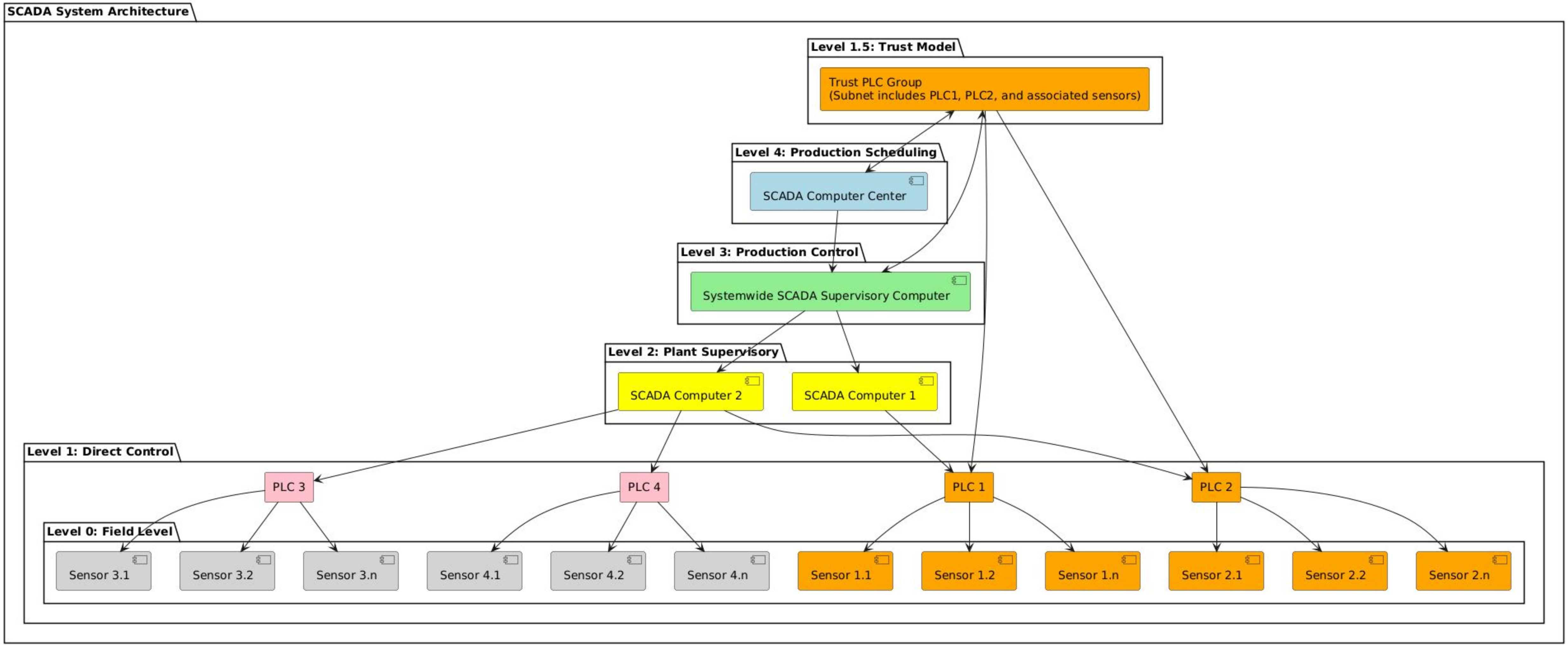

SCADA System Architecture
Level 4: Production Scheduling
SCADA Computer Center
Query Database
Level 3: Production Control
Systemwide SCADA Supervisory Computer
Log Database
Level 2: Plant Supervisory
SCADA Computer 1
Layer 2 Logs (Subnet)
Level 1: Direct Control
PLC 1
PLC n
Level 0: Field Level
Sensor 1.1
Sensor 1.2
Sensor 1.n
Sensor n.1
Sensor n.2
Sensor n.n

Thank you!



Do you have any questions?


Time server
SCADA
HMI
Server
Station Bus
IED
Process Bus
CB IED
MU
VT
CB
CT
Advanced Metering Infrastructure
Wire/Wireless
Utility Center
Data Concentrator
Smart Meter
Blade
Gearbox
Generator
Control Panel
Wind Power
Thermal Plants
Solar Panel
Inverter
Solar Power
Power Transmission
Big data analysis with AMI and distribution information
Substation
Power Distribution
Smart Factories
Enterprises
Smart Homes
Electric Vehicles
Power Generation
Power Management
Electricity Retailing


SCADA System Architecture
Level 1.5: Trust Model
Trust PLC Group
(Subnet includes PLC1, PLC2, and associated sensors)
Level 4: Production Scheduling
SCADA Computer Center
Level 3: Production Control
Systemwide SCADA Supervisory Computer
Level 2: Plant Supervisory
SCADA Computer 2
SCADA Computer 1
Level 1: Direct Control
PLC 3
PLC 4
PLC 1
PLC 2
Level 0: Field Level
Sensor 3.1
Sensor 3.2
Sensor 3.n
Sensor 4.1
Sensor 4.2
Sensor 4.n
Sensor 1.1
Sensor 1.2
Sensor 1.n
Sensor 2.1
Sensor 2.2
Sensor 2.n