DATA-PRIVACY FOCUSED FEDERATED LEARNING FRAMEWORK FOR INDUSTRIAL IOT

SLIIT UNI

THE KNOWLEDGE UNIVERSITY

Final Presentation

R25 - 039

PROJECT GROUP

Supervisors

Mr. Amila Nuwan Senarathne
Supervisor

Mr. Tharaniwarma Kyumaralingam
Co-Supervisor

Team

Nanayakkara Y.D.T.D
IT21826368

Mendis H.R.M
IT21822612

Dissanayaka K.D.A.R.A
IT21828348

Weerasinghe K.M
IT21831904

FRAMEWORK COMPONENT

Attack Defense And Resilience Module

Privacy Preservation Module

Secure Aggregation Module

Secure Communication and Protocol Enforcement Module

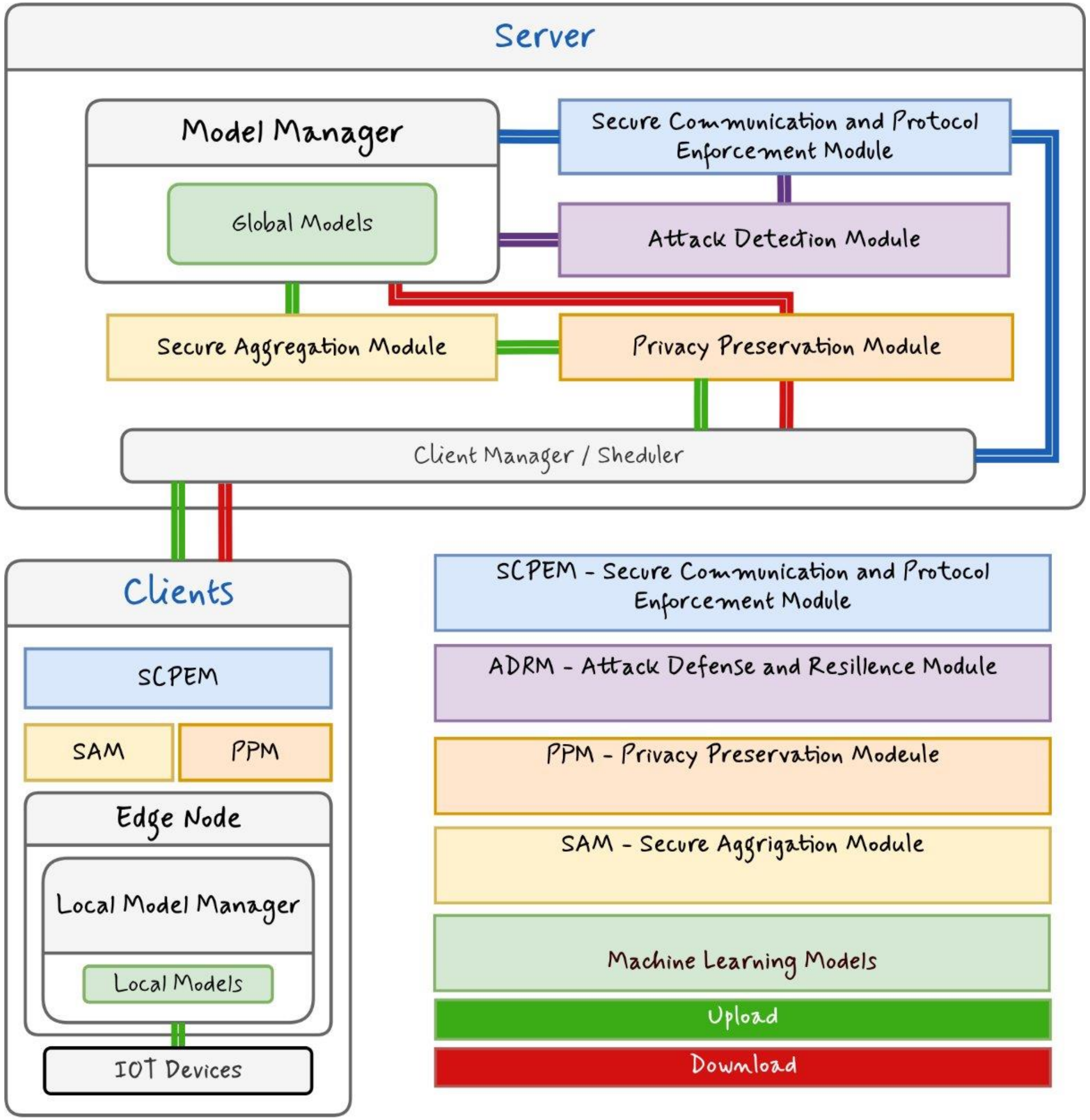
PROBLEMS

Standard Fl systems fall short in IIoT, lacking a unified approach to security , Privacy and efficiency. They remain vulnerable to poisoning and byzantine attacks, and security failures in IIoT can have severe consequences.

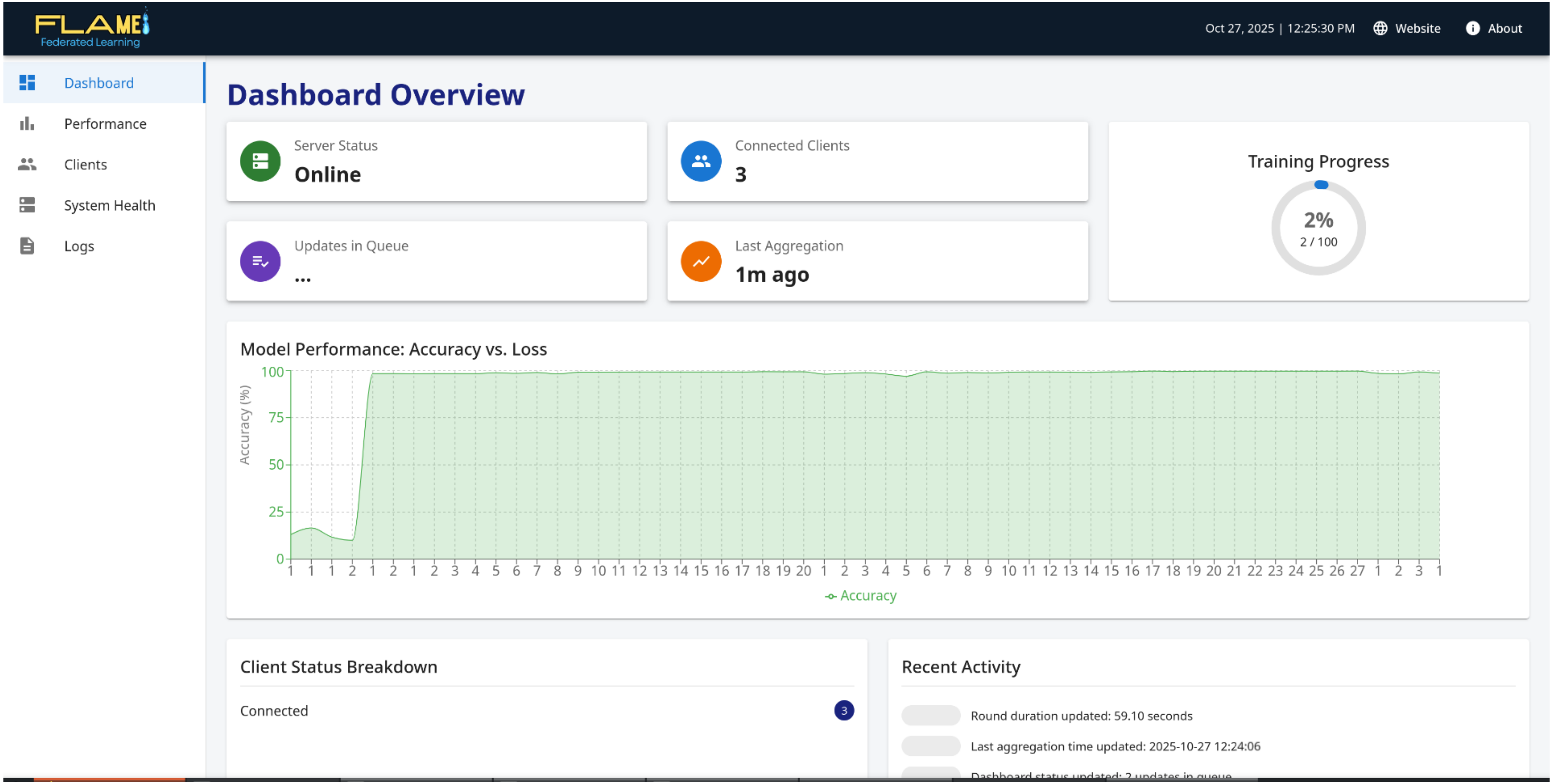
SOLUTIONS

- A modular framework enhancing FL with:
  - Hybrid privacy preservation (DP & HE),
  - Real-time attack detection
  - Lightweight secure aggregation (Shamir’s Secret Sharing)
  - Secure Communication and Protocol Enforcement Module

SYSTEM ARCHITECTURE


Server
Model Manager
Global Models
Secure Communication and Protocol Enforcement Module
Attack Detection Module
Secure Aggregation Module
Privacy Preservation Module
Client Manager / Sheduler
Clients
SCPEM
SAM
PPM
Edge Node
Local Model Manager
Local Models
IOT Devices
SCPEM - Secure Communication and Protocol Enforcement Module
ADRM - Attack Defense and Resillence Module
PPM - Privacy Preservation Modeule
SAM - Secure Aggrigation Module
Machine Learning Models
Upload
Download

System Web Portal



Dashboard Overview

Server Status
Online

Connected Clients
3

Training Progress
2%
2 / 100

Updates in Queue
...

Last Aggregation
1m ago

Model Performance: Accuracy vs. Loss

Client Status Breakdown

Connected	3

Recent Activity

Round duration updated: 59.10 seconds

Last aggregation time updated: 2025-10-27 12:24:06

Dashboard status updated: 2 updates in queue

Attack Detection and Resilience Module

IT21826368 | Nanayakkara Y.D.T.D

RESEARCH PROBLEM

- How can we design and integrate a lightweight, scalable, and effective attack detection and resilience system directly into the Federate Learning process to protect the global model's integrity in real-time?

RESEARCH GAP

- **Limitations**: Existing FL defenses are siloed, Targeting single threats, and its also too resource heavy for scalable IIoT
- High Overhead makes many solutions impractical for IIoT. Also, There are not any unified lightweight approach

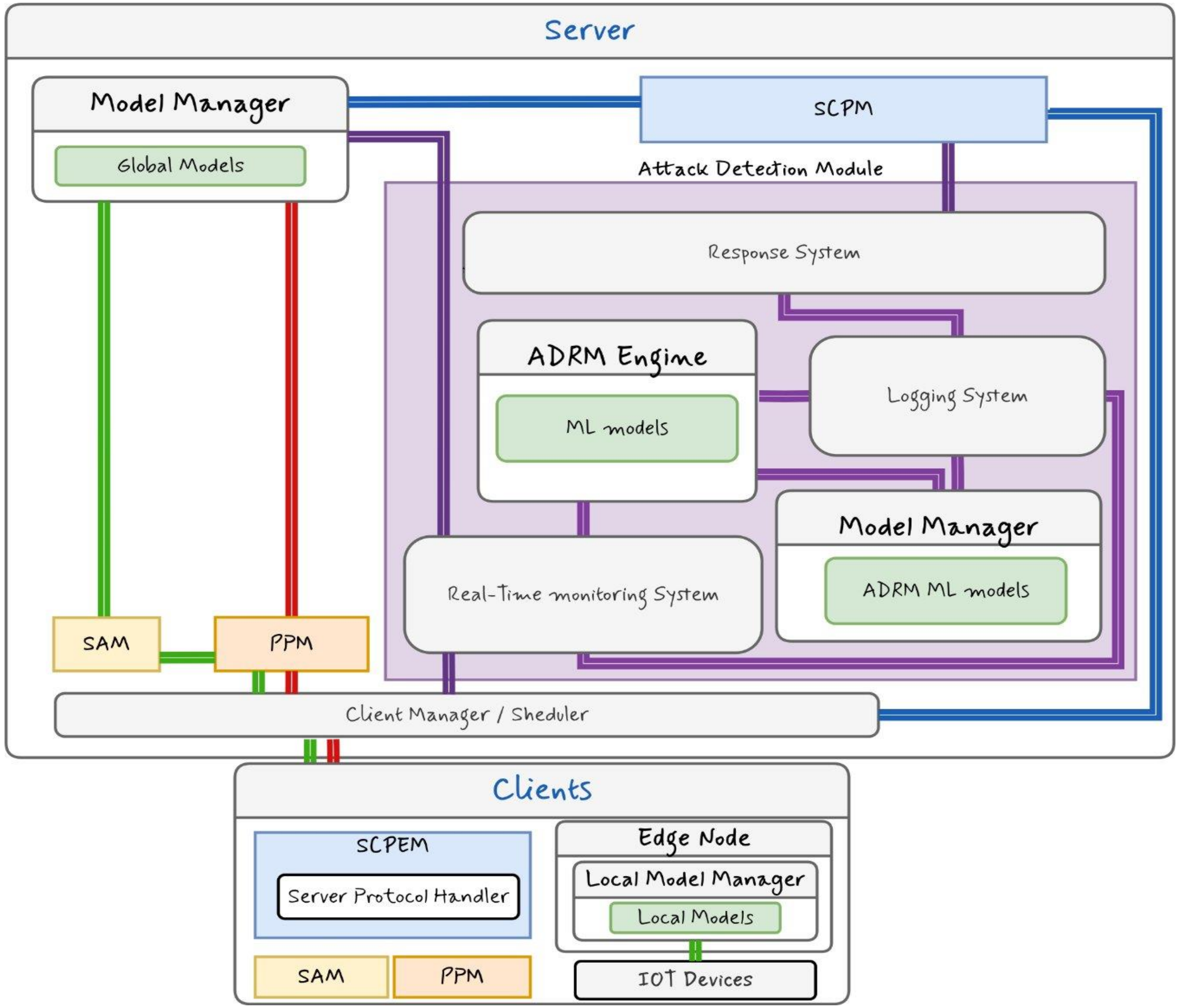
Background

- Federated learning is a promising privacy preserving solution for IIoT security, but its decentralized nature create a new and significant attack surface for malicious actors
- **Significance** : Without active defenses, Global model integrity is at risk,making robust security vital for industrial applications

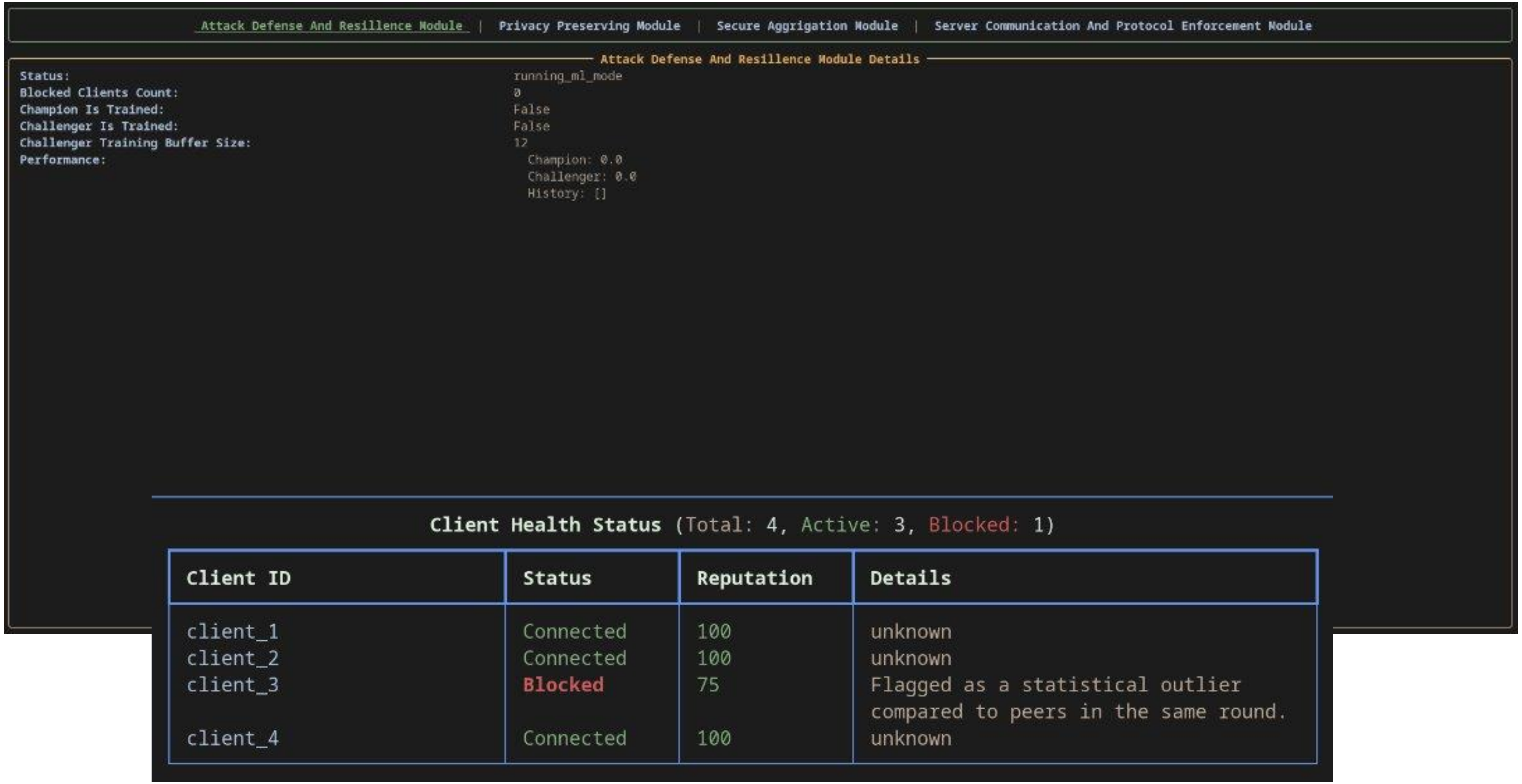
SOLUTION: ATTACK DETECTION & RESILIENCEMODULE

- **Main Objective:** To develop a scalable and lightweight module that enhances the security of FL in IIoT by detecting and mitigating malicious activities in real-time
- The Multi-Layered Approach:
  - **Real-Time Anomaly Detection:** Detects malicious updates with unsupervised learning
  - **Enhanced Resilience:** Ensures continuity with recovery mechanisms like rollbacks
  - **Client reputation grading system** – Reduce client points based on detection

ADRM ARCHITECTURE



Terminal User Interface



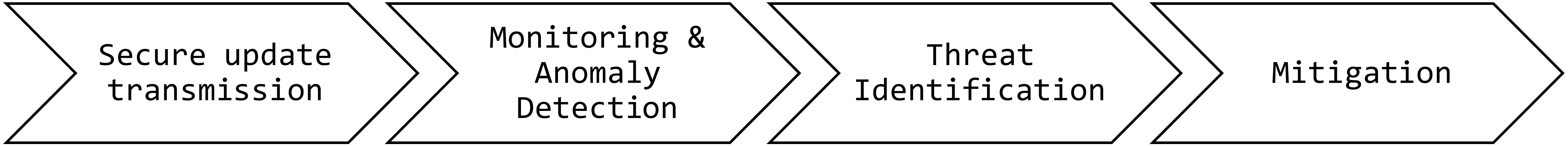
```
Attack Defense And Resillence Module | Privacy Preserving Module | Secure Aggrigation Module | Server Communication And Protocol Enforcement Module

Attack Defense And Resillence Module Details
Status:                               running_ml_mode
Blocked Clients Count:                0
Champion Is Trained:                  False
Challenger Is Trained:                False
Challenger Training Buffer Size:      12
Performance:
                                        Champion: 0.0
                                        Challenger: 0.0
                                        History: []
```

Client Health Status (Total: 4, Active: 3, Blocked: 1)

Client ID	Status	Reputation	Details
client_1	Connected	100	unknown
client_2	Connected	100	unknown
client_3	Blocked	75	Flagged as a statistical outlier compared to peers in the same round.
client_4	Connected	100	unknown

Image1: reputation grading and client blocking

DETECTION & MITIGATION PROCESS FLOW



Secure update transmission

- Clients send encrypted updates via secure channels

Monitoring & Anomaly Detection

- ADRM analyzes gradients, flags deviations

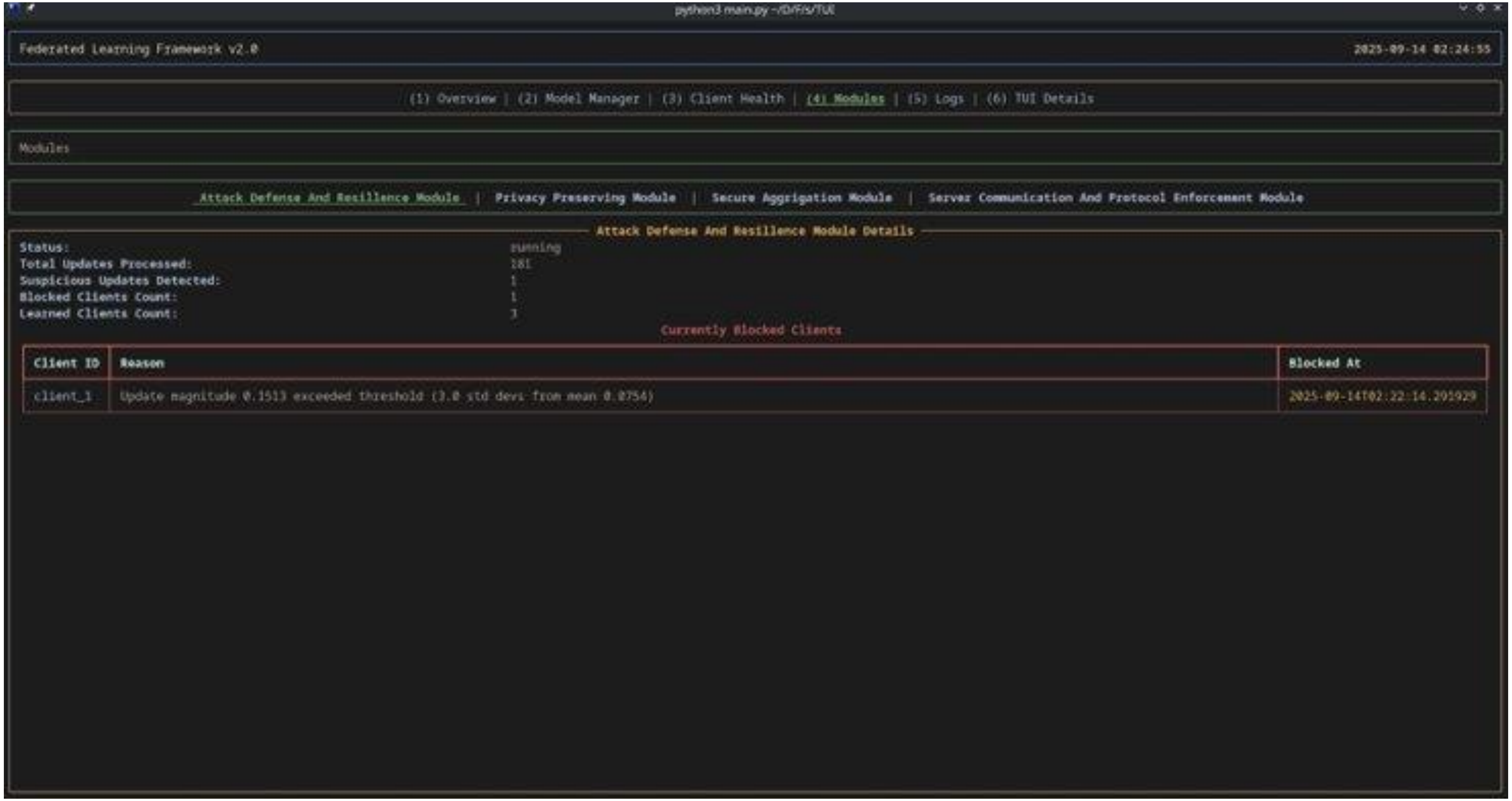
Threat Identification

- Detects model anomalies

Mitigation

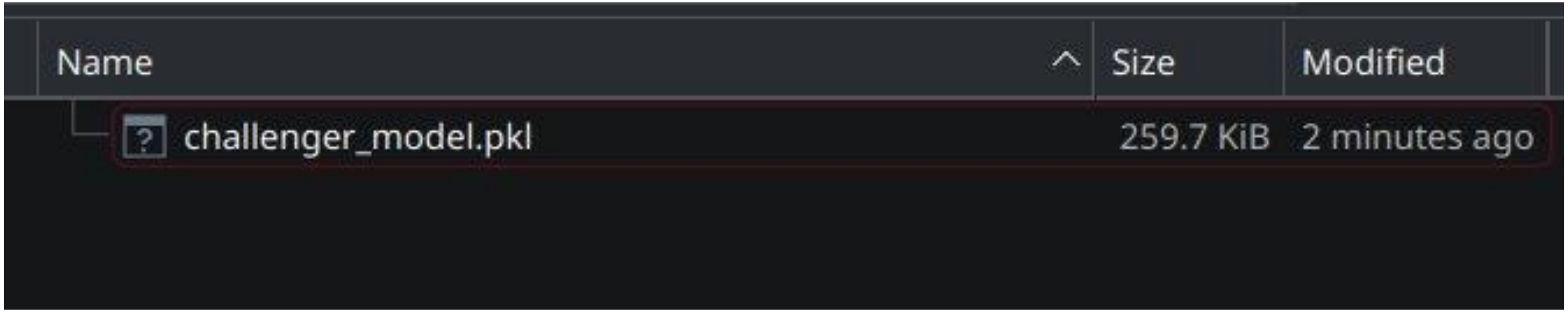
- Isolates/filter malicious clients; only verified updates aggregated
- Blocks and reduce reputation

Attack detection and Defense



```
python3 main.py ~/D/F/s/TUI

Federated Learning Framework v2.0                                                                 2025-09-14 02:24:55

             (1) Overview | (2) Model Manager | (3) Client Health | (4) Modules | (5) Logs | (6) TUI Details

Modules

   Attack Defense And Resilience Module | Privacy Preserving Module | Secure Aggrigation Module | Server Communication And Protocol Enforcement Module

──────────────────────────────── Attack Defense And Resilience Module Details ────────────────────────────────
Status:                                  running
Total Updates Processed:                 181
Suspicious Updates Detected:             1
Blocked Clients Count:                   1
Learned Clients Count:                   3
                                                 Currently Blocked Clients
 ┌───────────┬──────────────────────────────────────────────────────────────────────────┬────────────────────────────┐
 │ Client ID │ Reason                                                                   │ Blocked At                 │
 ├───────────┼──────────────────────────────────────────────────────────────────────────┼────────────────────────────┤
 │ client_1  │ Update magnitude 0.1513 exceeded threshold (3.0 std devs from mean 0.0754) │ 2025-09-14T02:22:14.291929 │
 └───────────┴──────────────────────────────────────────────────────────────────────────┴────────────────────────────┘
```

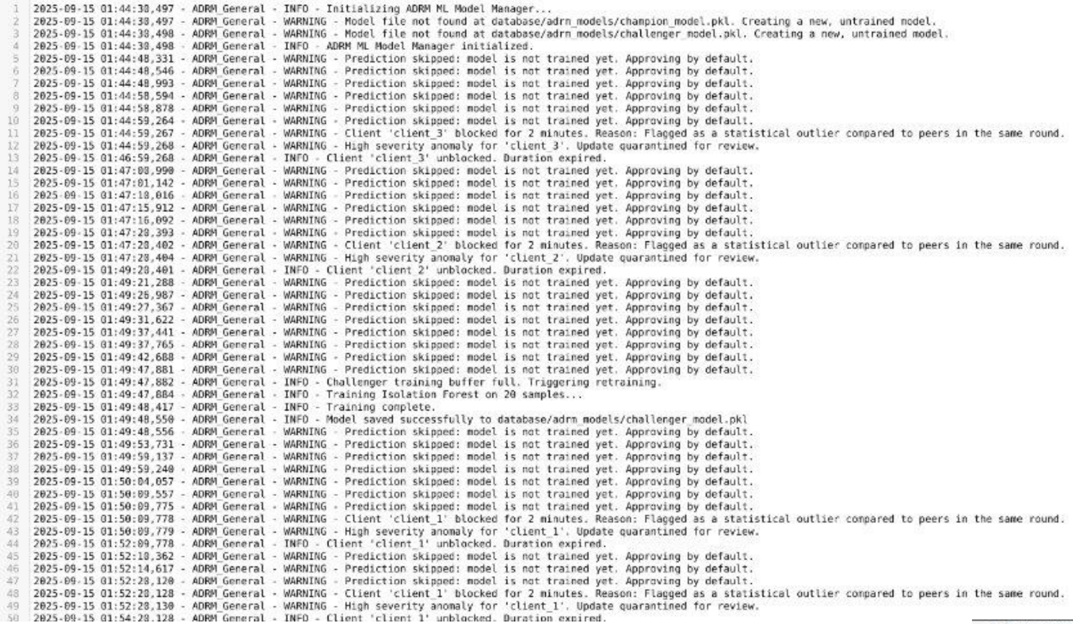
DETECTION METHOD



Name	Size	Modified
challenger_model.pkl	259.7 KiB	2 minutes ago

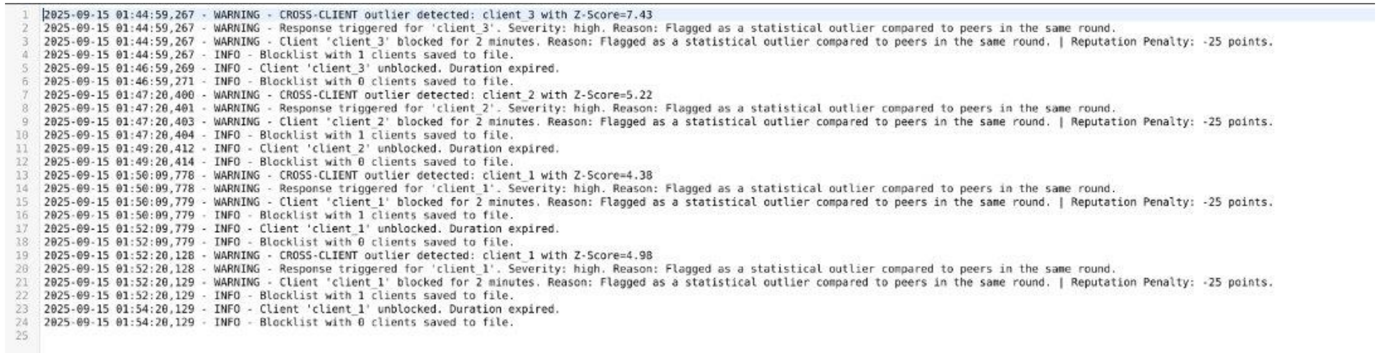
Created ML Model for anomaly detection

Logging



```
2025-09-15 01:44:30,497 - ADRM_General - INFO - Initializing ADRM ML Model Manager...
2025-09-15 01:44:30,497 - ADRM_General - WARNING - Model file not found at database/adrm_models/champion_model.pkl. Creating a new, untrained model.
2025-09-15 01:44:30,498 - ADRM_General - WARNING - Model file not found at database/adrm_models/challenger_model.pkl. Creating a new, untrained model.
2025-09-15 01:44:30,498 - ADRM_General - INFO - ADRM ML Model Manager initialized.
2025-09-15 01:44:40,331 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:44:40,546 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:44:40,993 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:44:50,594 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:44:50,878 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:44:59,264 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:44:59,267 - ADRM_General - WARNING - Client 'client_3' blocked for 2 minutes. Reason: Flagged as a statistical outlier compared to peers in the same round.
2025-09-15 01:44:59,268 - ADRM_General - WARNING - High severity anomaly for 'client_3'. Update quarantined for review.
2025-09-15 01:46:59,268 - ADRM_General - INFO - Client 'client_3' unblocked. Duration expired.
2025-09-15 01:47:00,990 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:47:01,142 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:47:10,016 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:47:15,912 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:47:16,092 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:47:20,393 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:47:20,402 - ADRM_General - WARNING - Client 'client_2' blocked for 2 minutes. Reason: Flagged as a statistical outlier compared to peers in the same round.
2025-09-15 01:47:20,404 - ADRM_General - WARNING - High severity anomaly for 'client_2'. Update quarantined for review.
2025-09-15 01:49:20,401 - ADRM_General - INFO - Client 'client_2' unblocked. Duration expired.
2025-09-15 01:49:21,288 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:49:26,987 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:49:27,367 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:49:31,622 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:49:37,441 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:49:37,765 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:49:42,688 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:49:47,881 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:49:47,882 - ADRM_General - INFO - Challenger training buffer full. Triggering retraining.
2025-09-15 01:49:47,884 - ADRM_General - INFO - Training Isolation Forest on 20 samples...
2025-09-15 01:49:48,417 - ADRM_General - INFO - Training complete.
2025-09-15 01:49:48,550 - ADRM_General - INFO - Model saved successfully to database/adrm_models/challenger_model.pkl
2025-09-15 01:49:48,556 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:49:53,731 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:49:59,137 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:49:59,240 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:50:04,057 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:50:09,557 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:50:09,775 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:50:09,778 - ADRM_General - WARNING - Client 'client_1' blocked for 2 minutes. Reason: Flagged as a statistical outlier compared to peers in the same round.
2025-09-15 01:50:09,779 - ADRM_General - WARNING - High severity anomaly for 'client_1'. Update quarantined for review.
2025-09-15 01:52:09,778 - ADRM_General - INFO - Client 'client_1' unblocked. Duration expired.
2025-09-15 01:52:10,362 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:52:14,617 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:52:20,120 - ADRM_General - WARNING - Prediction skipped: model is not trained yet. Approving by default.
2025-09-15 01:52:20,128 - ADRM_General - WARNING - Client 'client_1' blocked for 2 minutes. Reason: Flagged as a statistical outlier compared to peers in the same round.
2025-09-15 01:52:20,130 - ADRM_General - WARNING - High severity anomaly for 'client_1'. Update quarantined for review.
2025-09-15 01:54:20,128 - ADRM_General - INFO - Client 'client_1' unblocked. Duration expired.
```

Model creation based on dataset in the current FL system



```
2025-09-15 01:44:59,267 - WARNING - CROSS-CLIENT outlier detected: client_3 with Z-Score=7.43
2025-09-15 01:44:59,267 - WARNING - Response triggered for 'client_3'. Severity: high. Reason: Flagged as a statistical outlier compared to peers in the same round.
2025-09-15 01:44:59,267 - WARNING - Client 'client_3' blocked for 2 minutes. Reason: Flagged as a statistical outlier compared to peers in the same round. | Reputation Penalty: -25 points.
2025-09-15 01:44:59,267 - INFO - Blocklist with 1 clients saved to file.
2025-09-15 01:46:59,269 - INFO - Client 'client_3' unblocked. Duration expired.
2025-09-15 01:46:59,271 - INFO - Blocklist with 0 clients saved to file.
2025-09-15 01:47:20,400 - WARNING - CROSS-CLIENT outlier detected: client_2 with Z-Score=5.22
2025-09-15 01:47:20,401 - WARNING - Response triggered for 'client_2'. Severity: high. Reason: Flagged as a statistical outlier compared to peers in the same round.
2025-09-15 01:47:20,403 - WARNING - Client 'client_2' blocked for 2 minutes. Reason: Flagged as a statistical outlier compared to peers in the same round. | Reputation Penalty: -25 points.
2025-09-15 01:47:20,404 - INFO - Blocklist with 1 clients saved to file.
2025-09-15 01:49:20,412 - INFO - Client 'client_2' unblocked. Duration expired.
2025-09-15 01:49:20,414 - INFO - Blocklist with 0 clients saved to file.
2025-09-15 01:50:09,778 - WARNING - CROSS-CLIENT outlier detected: client_1 with Z-Score=4.38
2025-09-15 01:50:09,778 - WARNING - Response triggered for 'client_1'. Severity: high. Reason: Flagged as a statistical outlier compared to peers in the same round.
2025-09-15 01:50:09,779 - WARNING - Client 'client_1' blocked for 2 minutes. Reason: Flagged as a statistical outlier compared to peers in the same round. | Reputation Penalty: -25 points.
2025-09-15 01:50:09,779 - INFO - Blocklist with 1 clients saved to file.
2025-09-15 01:52:09,779 - INFO - Client 'client_1' unblocked. Duration expired.
2025-09-15 01:52:09,779 - INFO - Blocklist with 0 clients saved to file.
2025-09-15 01:52:20,128 - WARNING - CROSS-CLIENT outlier detected: client_1 with Z-Score=4.98
2025-09-15 01:52:20,128 - WARNING - Response triggered for 'client_1'. Severity: high. Reason: Flagged as a statistical outlier compared to peers in the same round.
2025-09-15 01:52:20,129 - WARNING - Client 'client_1' blocked for 2 minutes. Reason: Flagged as a statistical outlier compared to peers in the same round. | Reputation Penalty: -25 points.
2025-09-15 01:52:20,129 - INFO - Blocklist with 1 clients saved to file.
2025-09-15 01:54:20,129 - INFO - Client 'client_1' unblocked. Duration expired.
2025-09-15 01:54:20,129 - INFO - Blocklist with 0 clients saved to file.
```

Attack detection

Privacy Preservation Module

IT21822612 | Mendis H.R.M

BACKGROUND

- Federated Learning enables decentralized model training but remains vulnerable to inference attacks from untrusted servers. Combining **Differential Privacy (DP)**for noise and **Homomorphic Encryption (HE)** for secure communication provides a hybrid solution to protect client data.

RESEARCH GAP

- Existing Federated Learning methods use either Differential Privacy or Homomorphic Encryption alone, failing to provide full protection without compromising accuracy or efficiency, especially in resource-limited IIoT settings.

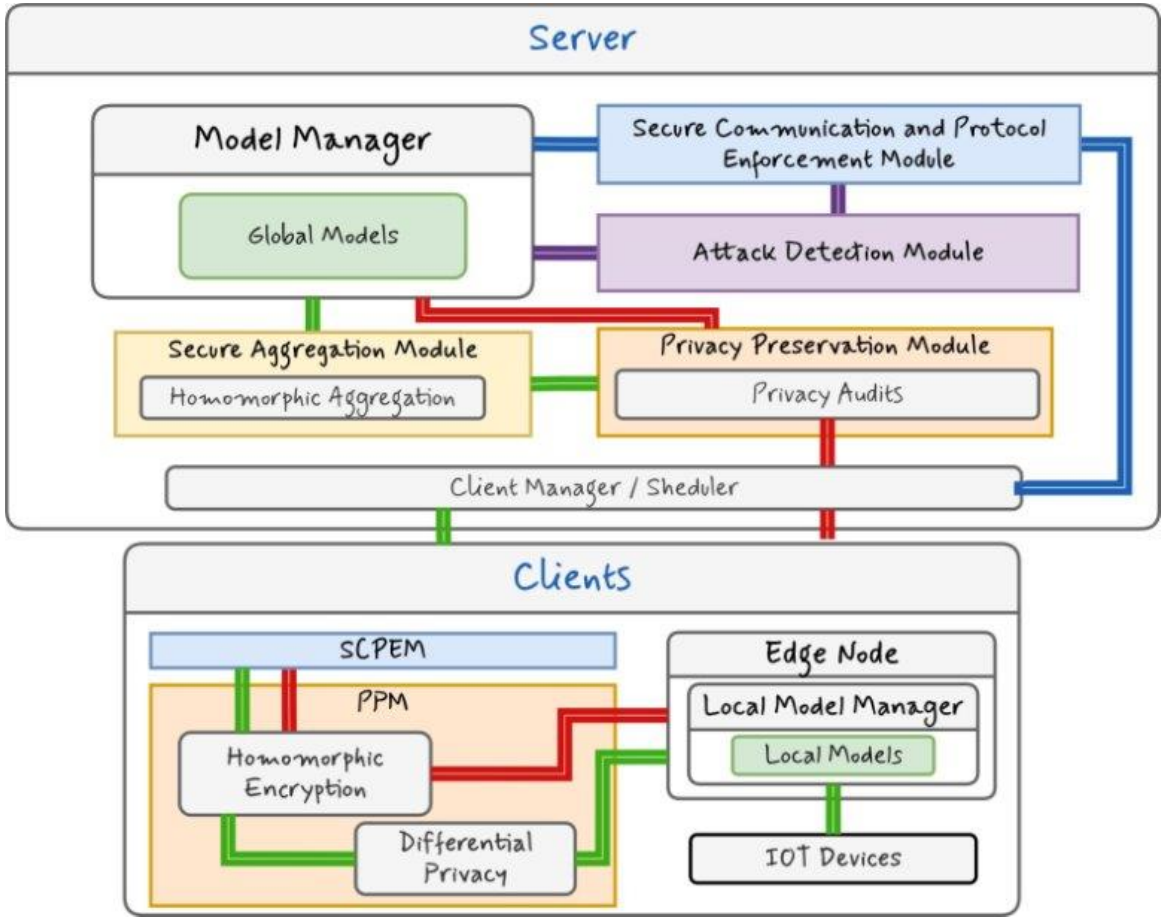
RESEARCH PROBLEM

- How can we design a hybrid Federated Learning framework combining Differential Privacy and Homomorphic Encryption to protect client data from inference attacks by untrusted servers while maintaining model accuracy and communication efficiency?

SOLUTION:

- A hybrid privacy-preserving Federated Learning framework that combines **Differential Privacy (DP)** to add noise during training and **Homomorphic Encryption (HE)** to secure communication of model updates. This approach protects client data from inference attacks by untrusted servers while balancing model accuracy and communication efficiency, tailored for resource-constrained Industrial IoT environments.

PPM Architecture



METHODOLOGY

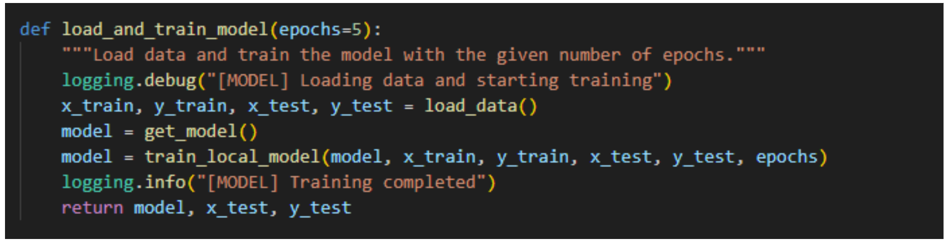
Approach:

- Analyze existing FL privacy vulnerabilities.
- Combine HE and DP for enhanced privacy.
- Optimize techniques for IIoT-specific constraints.
- Validate Using real-world Datasets

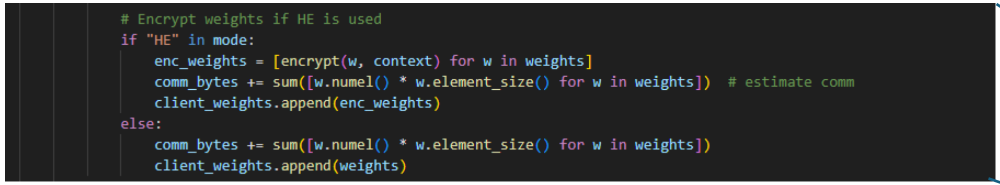
Key Techniques:

- **Homomorphic Encryption (HE)**: Encrypts gradients, allowing computations on encrypted data without decrypting it. Prevents data leakage even if adversaries intercept communications.
- **Differential Privacy (DP)**: Ensure that individual data points cannot be separated by adding controlled noise to gradients. Balances model accuracy with privacy

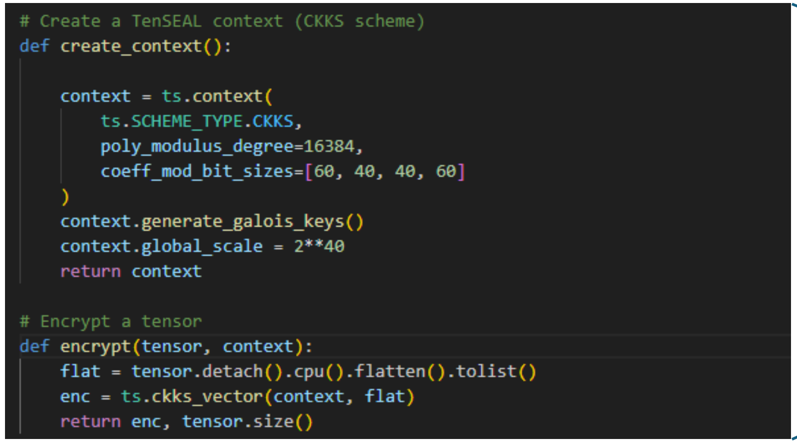
FUNCTIONAL UNITS & IMPLEMENTATIONS (Client)



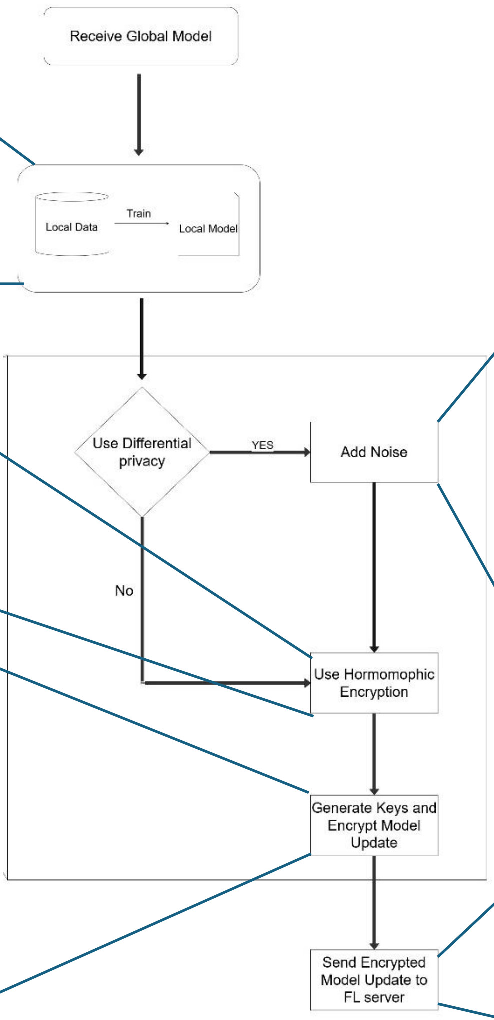
```python
def load_and_train_model(epochs=5):
    """Load data and train the model with the given number of epochs."""
    logging.debug("[MODEL] Loading data and starting training")
    x_train, y_train, x_test, y_test = load_data()
    model = get_model()
    model = train_local_model(model, x_train, y_train, x_test, y_test, epochs)
    logging.info("[MODEL] Training completed")
    return model, x_test, y_test
```

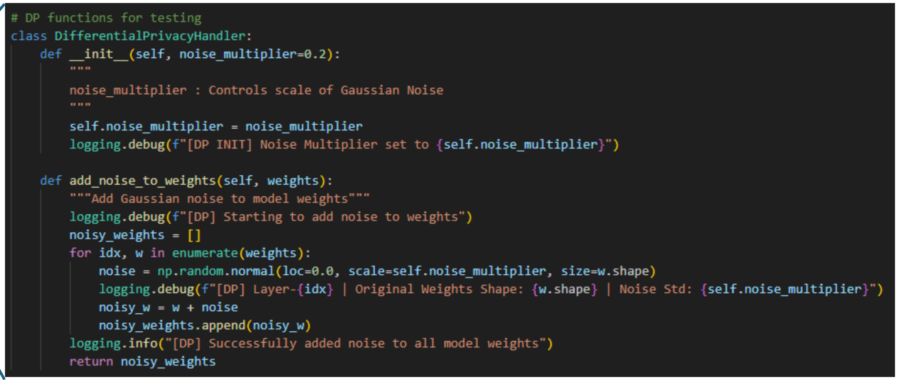


```python
            # Encrypt weights if HE is used
            if "HE" in mode:
                enc_weights = [encrypt(w, context) for w in weights]
                comm_bytes += sum([w.numel() * w.element_size() for w in weights])  # estimate comm
                client_weights.append(enc_weights)
            else:
                comm_bytes += sum([w.numel() * w.element_size() for w in weights])
                client_weights.append(weights)
```

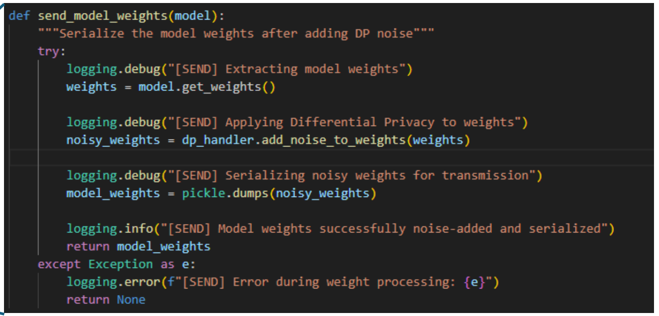


```python
# Create a TenSEAL context (CKKS scheme)
def create_context():

    context = ts.context(
        ts.SCHEME_TYPE.CKKS,
        poly_modulus_degree=16384,
        coeff_mod_bit_sizes=[60, 40, 40, 60]
    )
    context.generate_galois_keys()
    context.global_scale = 2**40
    return context

# Encrypt a tensor
def encrypt(tensor, context):
    flat = tensor.detach().cpu().flatten().tolist()
    enc = ts.ckks_vector(context, flat)
    return enc, tensor.size()
```


Receive Global Model
Local Data
Train
Local Model
Use Differential privacy
YES
Add Noise
No
Use Hormomophic Encryption
Generate Keys and Encrypt Model Update
Send Encrypted Model Update to FL server

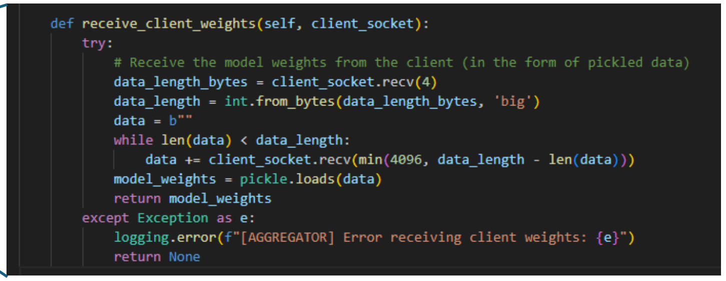


```python
# DP functions for testing
class DifferentialPrivacyHandler:
    def __init__(self, noise_multiplier=0.2):
        """
        noise_multiplier : Controls scale of Gaussian Noise
        """
        self.noise_multiplier = noise_multiplier
        logging.debug(f"[DP INIT] Noise Multiplier set to {self.noise_multiplier}")

    def add_noise_to_weights(self, weights):
        """Add Gaussian noise to model weights"""
        logging.debug(f"[DP] Starting to add noise to weights")
        noisy_weights = []
        for idx, w in enumerate(weights):
            noise = np.random.normal(loc=0.0, scale=self.noise_multiplier, size=w.shape)
            logging.debug(f"[DP] Layer-{idx} | Original Weights Shape: {w.shape} | Noise Std: {self.noise_multiplier}")
            noisy_w = w + noise
            noisy_weights.append(noisy_w)
        logging.info("[DP] Successfully added noise to all model weights")
        return noisy_weights
```

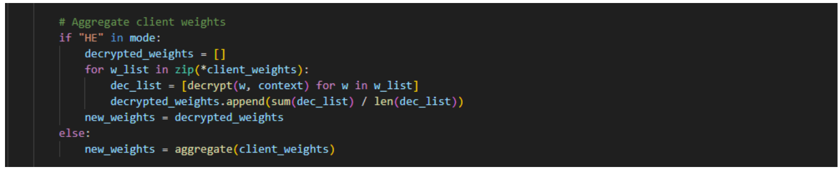


```python
def send_model_weights(model):
    """Serialize the model weights after adding DP noise"""
    try:
        logging.debug("[SEND] Extracting model weights")
        weights = model.get_weights()

        logging.debug("[SEND] Applying Differential Privacy to weights")
        noisy_weights = dp_handler.add_noise_to_weights(weights)

        logging.debug("[SEND] Serializing noisy weights for transmission")
        model_weights = pickle.dumps(noisy_weights)

        logging.info("[SEND] Model weights successfully noise-added and serialized")
        return model_weights
    except Exception as e:
        logging.error(f"[SEND] Error during weight processing: {e}")
        return None
```

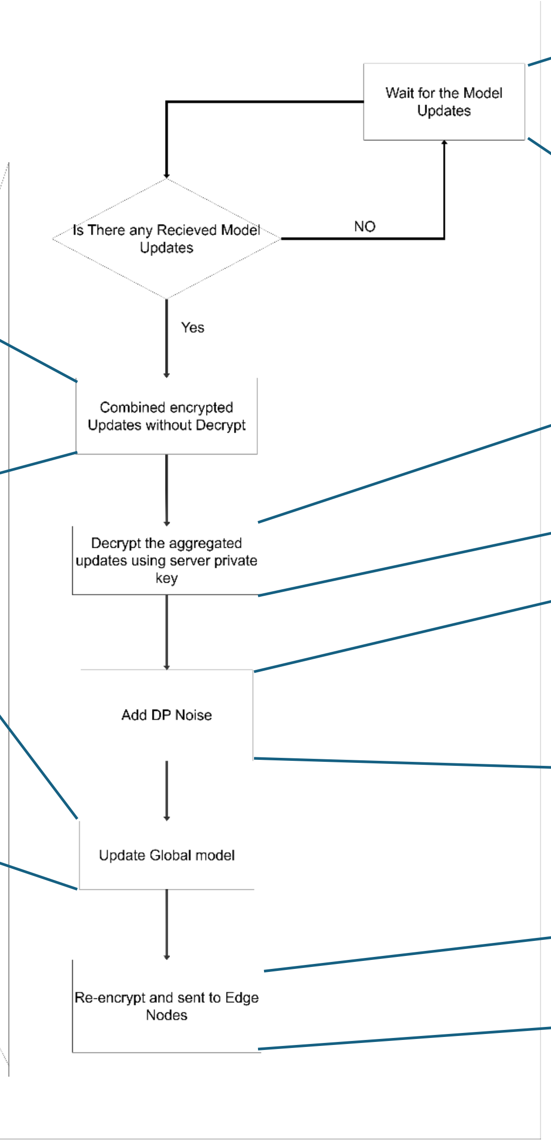
FUNCTIONAL UNITS & IMPLEMENTATIONS (Server)



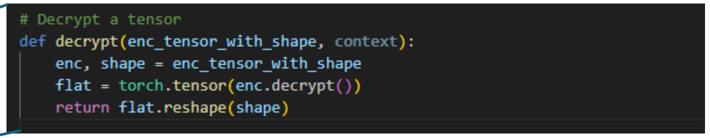
```
def receive_client_weights(self, client_socket):
    try:
        # Receive the model weights from the client (in the form of pickled data)
        data_length_bytes = client_socket.recv(4)
        data_length = int.from_bytes(data_length_bytes, 'big')
        data = b""
        while len(data) < data_length:
            data += client_socket.recv(min(4096, data_length - len(data)))
        model_weights = pickle.loads(data)
        return model_weights
    except Exception as e:
        logging.error(f"[AGGREGATOR] Error receiving client weights: {e}")
        return None
```



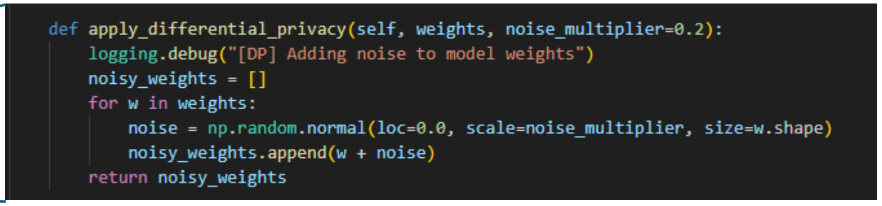
```
# Aggregate client weights
if "HE" in mode:
    decrypted_weights = []
    for w_list in zip(*client_weights):
        dec_list = [decrypt(w, context) for w in w_list]
        decrypted_weights.append(sum(dec_list) / len(dec_list))
    new_weights = decrypted_weights
else:
    new_weights = aggregate(client_weights)
```



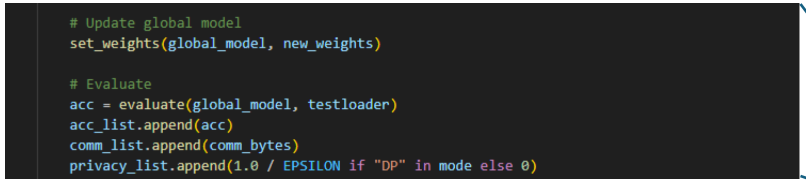
Wait for the Model Updates
Is There any Recieved Model Updates
NO
Yes
Combined encrypted Updates without Decrypt
Decrypt the aggregated updates using server private key
Add DP Noise
Update Global model
Re-encrypt and sent to Edge Nodes



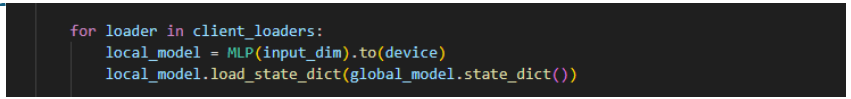
```
# Decrypt a tensor
def decrypt(enc_tensor_with_shape, context):
    enc, shape = enc_tensor_with_shape
    flat = torch.tensor(enc.decrypt())
    return flat.reshape(shape)
```



```
def apply_differential_privacy(self, weights, noise_multiplier=0.2):
    logging.debug("[DP] Adding noise to model weights")
    noisy_weights = []
    for w in weights:
        noise = np.random.normal(loc=0.0, scale=noise_multiplier, size=w.shape)
        noisy_weights.append(w + noise)
    return noisy_weights
```



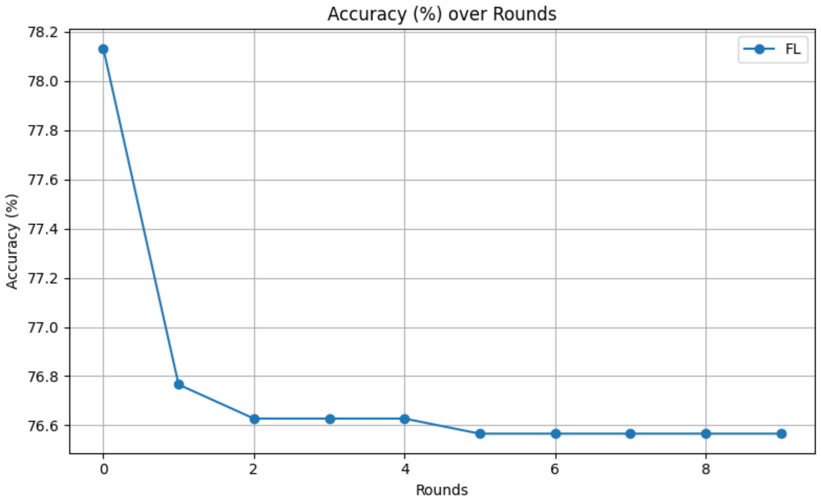
```
    # Update global model
    set_weights(global_model, new_weights)

    # Evaluate
    acc = evaluate(global_model, testloader)
    acc_list.append(acc)
    comm_list.append(comm_bytes)
    privacy_list.append(1.0 / EPSILON if "DP" in mode else 0)
```



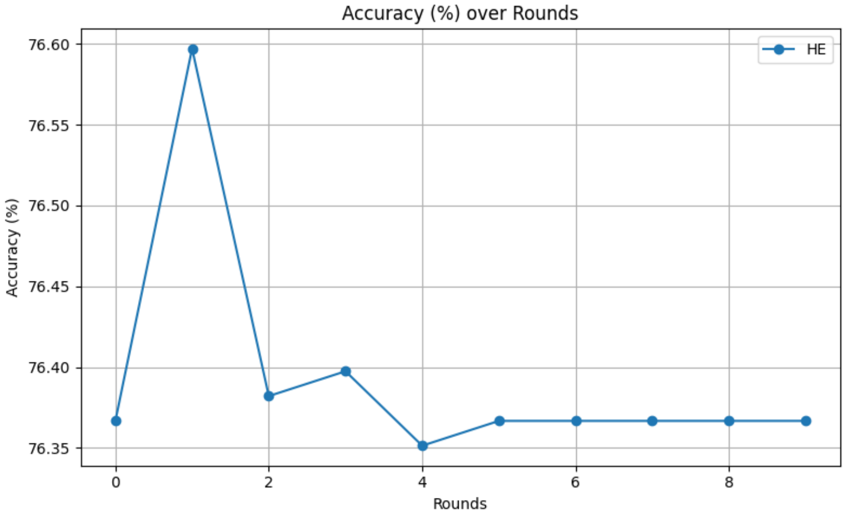
```
    for loader in client_loaders:
        local_model = MLP(input_dim).to(device)
        local_model.load_state_dict(global_model.state_dict())
```

PROOFS & RESULTS

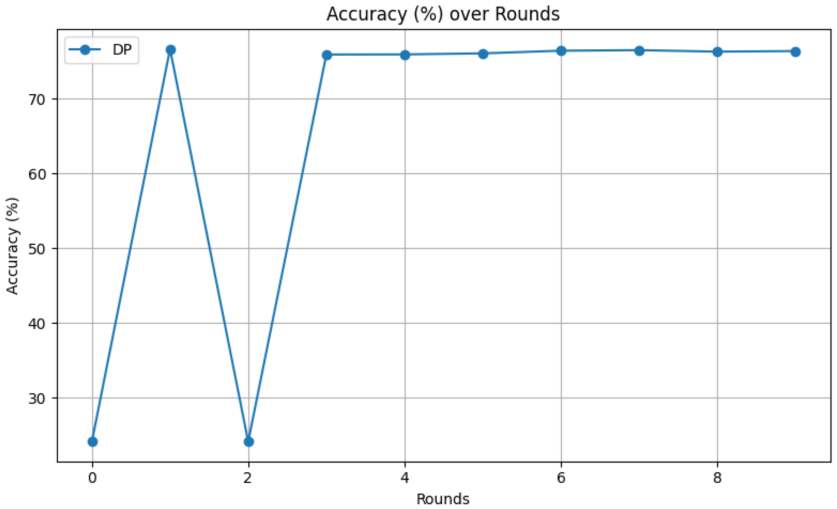
- **Accuracy Curves**



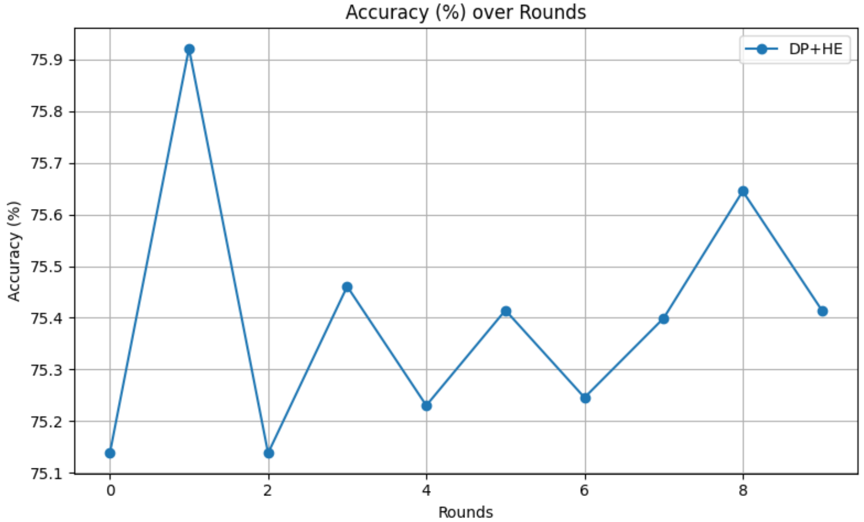
FL



FL + HE



FL + DP



FL + DP + HE

Note: Testing was done using UCI Adult Dataset.

PROOFS & RESULTS

Average Accuracy over Rounds

Mode	Average Accuracy (%)	Remarks
FL	76.81	Highest accuracy because no extra privacy mechanisms are applied.
FL + DP	65.87	~11% Lower accuracy due to Differential Privacy adds **noise** to gradients to protect individuals.
FL + HE	76.39	Differential Privacy adds **noise** to gradients/weights to protect individuals. Homomorphic Encryption protects updates **without changing the data** (no noise).
FL + DP + HE	75.34	Accuracy only slightly lower than FL (~1.5% drop)

FL + DP + HE is the best choice, balancing utility and privacy. FL alone has the highest accuracy but no privacy. DP offers strong privacy with lower accuracy, while HE protects only communication. Combining DP and HE gives strong privacy with minimal accuracy loss, making it both practical and secure.

Secure Communication and Protocol Enforcement

IT21828348 | Dissanayaka KDARA

BACKGROUND

- In IIoT, communication between clients and server is a prime target for threats like eavesdropping , tampering, and MITM attacks. Standar FL focused on efficiency but lacked strong defenses. TLS provides a baseline, but its one-sided authentication leaves gaps.

RESEARCH GAP

- Current solutions lack an integrated security framework for IIoT. Standard TLS often validates only the server, leaving no assurance that connected devices are legitimate. A More strict, two-way validation protocol is needed.

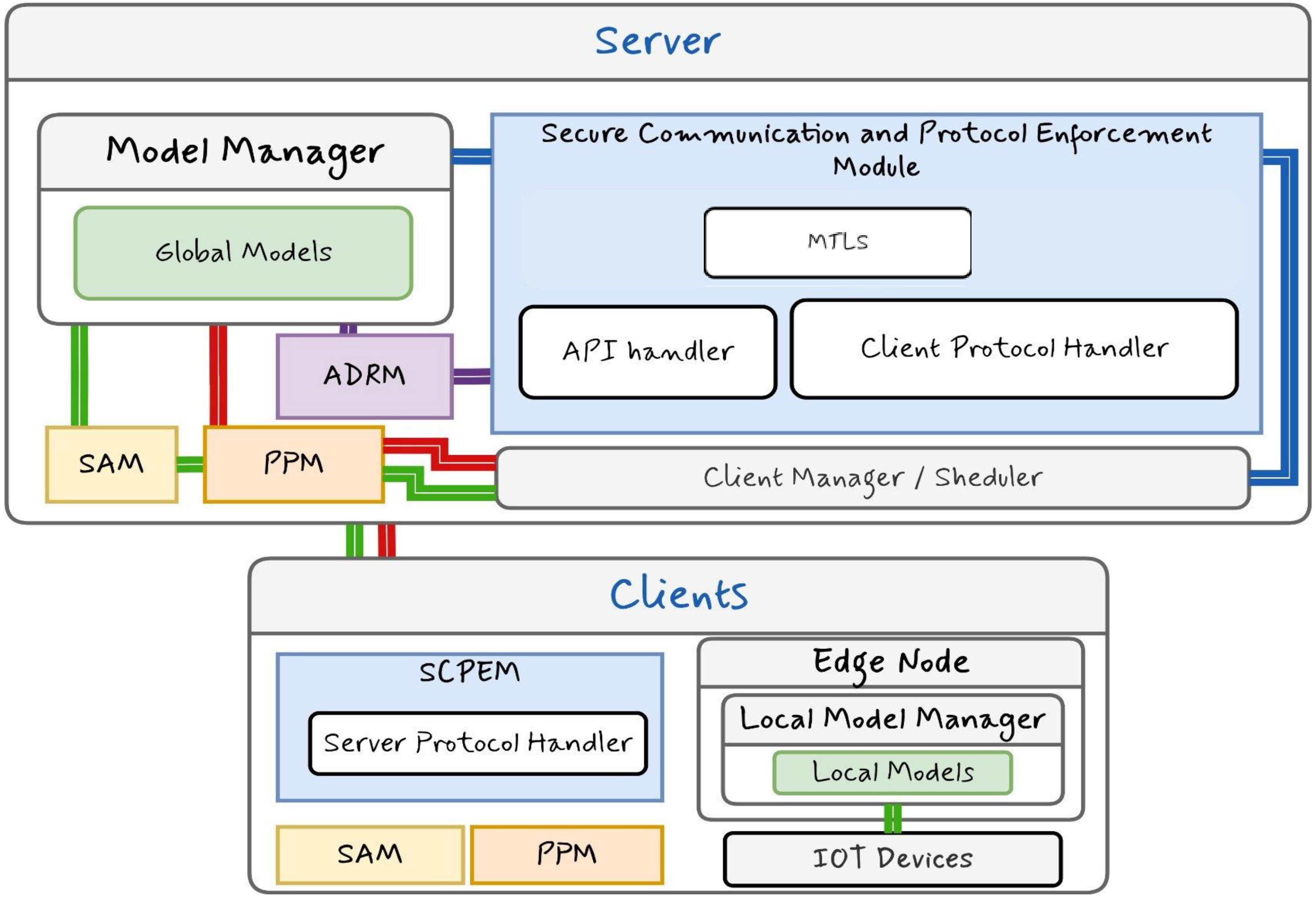
RESEARCH PROBLEM

- Insecure FL channels risk unauthorized access, tampered updates, fake servers, and undetected data changes.

SOLUTION:

- Implement a secure communication protocol that ensures only authorized IIoT devices can participate, protects model updates from tampering, verifies the legitimacy of the server, and adds integrity checks beyond encryption to guarantee unaltered data.

Module Architecture



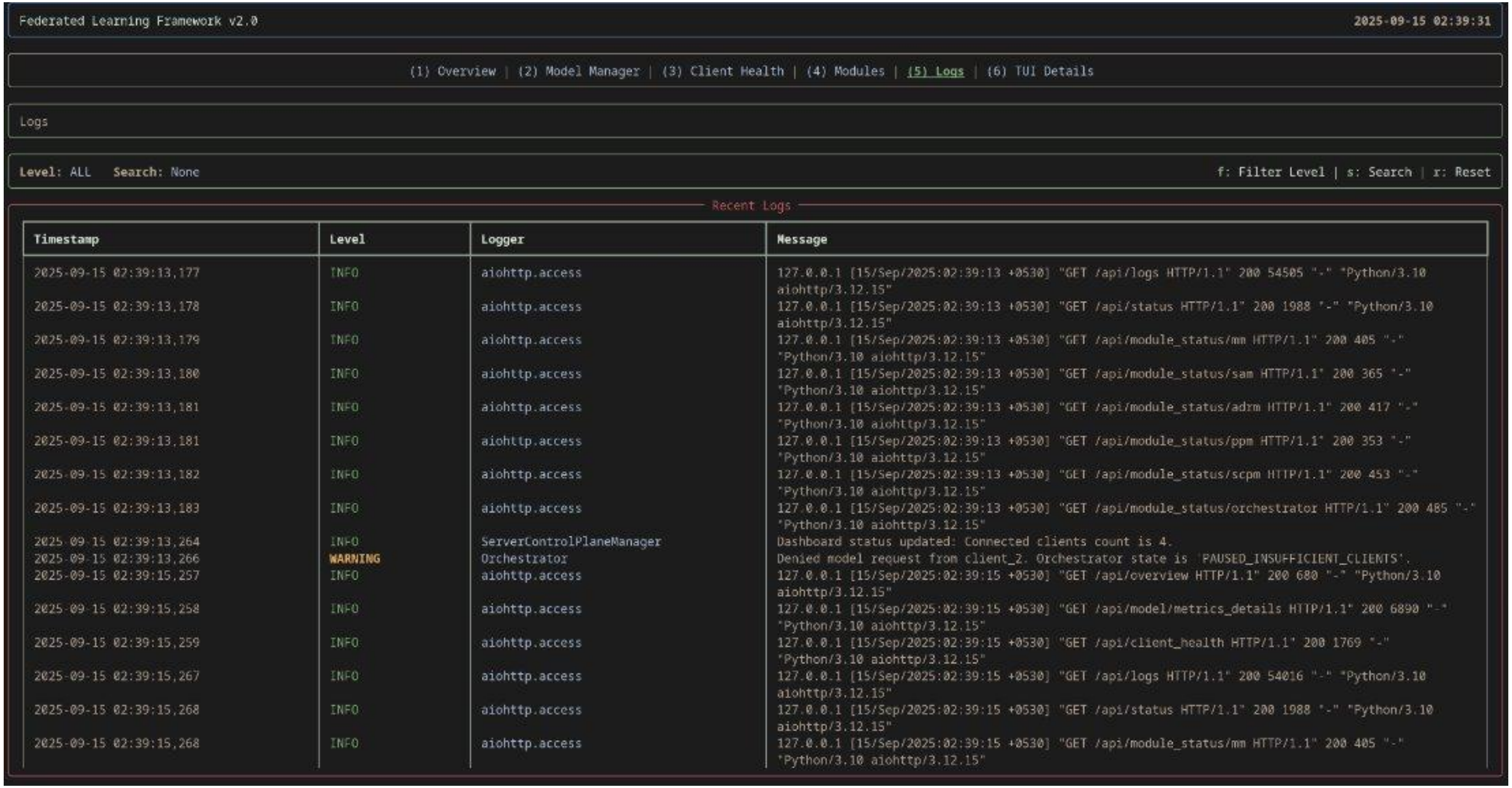
SERVER COMPONENTS

- **Model Manager:** Distributes and updates the global model.
- **SCPEM:** Secure channel using mTLS (two-way authentication) +HMAC for integrity.
- **Client Manager/Scheduler:** Selects and manages participating clients.

OVERALL WORKFLOW

1. Server selects clients for training.
2. Global model sent via **SCPEM** (secure channel).
3. Clients train locally on IoT data.
4. Updates protected using **PPM & SAM**.
5. Protected updates sent back via **SCPM**.
6. Server validates updates with **ADRM**.
7. **SAM** aggregates updates securely.
8. **Model Manager** updates global model.
9. Cycle repeats.

Log manager



Federated Learning Framework v2.0 2025-09-15 02:39:31

(1) Overview | (2) Model Manager | (3) Client Health | (4) Modules | (5) Logs | (6) TUI Details
Logs
f: Filter Level | s: Search | r: Reset
Recent Logs

Level: ALL **Search:** None

Timestamp	Level	Logger	Message
2025-09-15 02:39:13,177	INFO	aiohttp.access	127.0.0.1 [15/Sep/2025:02:39:13 +0530] "GET /api/logs HTTP/1.1" 200 54505 "-" "Python/3.10 aiohttp/3.12.15"
2025-09-15 02:39:13,178	INFO	aiohttp.access	127.0.0.1 [15/Sep/2025:02:39:13 +0530] "GET /api/status HTTP/1.1" 200 1988 "-" "Python/3.10 aiohttp/3.12.15"
2025-09-15 02:39:13,179	INFO	aiohttp.access	127.0.0.1 [15/Sep/2025:02:39:13 +0530] "GET /api/module_status/mm HTTP/1.1" 200 405 "-" "Python/3.10 aiohttp/3.12.15"
2025-09-15 02:39:13,180	INFO	aiohttp.access	127.0.0.1 [15/Sep/2025:02:39:13 +0530] "GET /api/module_status/sam HTTP/1.1" 200 365 "-" "Python/3.10 aiohttp/3.12.15"
2025-09-15 02:39:13,181	INFO	aiohttp.access	127.0.0.1 [15/Sep/2025:02:39:13 +0530] "GET /api/module_status/adrm HTTP/1.1" 200 417 "-" "Python/3.10 aiohttp/3.12.15"
2025-09-15 02:39:13,181	INFO	aiohttp.access	127.0.0.1 [15/Sep/2025:02:39:13 +0530] "GET /api/module_status/ppm HTTP/1.1" 200 353 "-" "Python/3.10 aiohttp/3.12.15"
2025-09-15 02:39:13,182	INFO	aiohttp.access	127.0.0.1 [15/Sep/2025:02:39:13 +0530] "GET /api/module_status/scpm HTTP/1.1" 200 453 "-" "Python/3.10 aiohttp/3.12.15"
2025-09-15 02:39:13,183	INFO	aiohttp.access	127.0.0.1 [15/Sep/2025:02:39:13 +0530] "GET /api/module_status/orchestrator HTTP/1.1" 200 485 "-" "Python/3.10 aiohttp/3.12.15"
2025-09-15 02:39:13,264	INFO	ServerControlPlaneManager	Dashboard status updated: Connected clients count is 4.
2025-09-15 02:39:13,266	WARNING	Orchestrator	Denied model request from client_2. Orchestrator state is 'PAUSED_INSUFFICIENT_CLIENTS'.
2025-09-15 02:39:15,257	INFO	aiohttp.access	127.0.0.1 [15/Sep/2025:02:39:15 +0530] "GET /api/overview HTTP/1.1" 200 680 "-" "Python/3.10 aiohttp/3.12.15"
2025-09-15 02:39:15,258	INFO	aiohttp.access	127.0.0.1 [15/Sep/2025:02:39:15 +0530] "GET /api/model/metrics_details HTTP/1.1" 200 6890 "-" "Python/3.10 aiohttp/3.12.15"
2025-09-15 02:39:15,259	INFO	aiohttp.access	127.0.0.1 [15/Sep/2025:02:39:15 +0530] "GET /api/client_health HTTP/1.1" 200 1769 "-" "Python/3.10 aiohttp/3.12.15"
2025-09-15 02:39:15,267	INFO	aiohttp.access	127.0.0.1 [15/Sep/2025:02:39:15 +0530] "GET /api/logs HTTP/1.1" 200 54016 "-" "Python/3.10 aiohttp/3.12.15"
2025-09-15 02:39:15,268	INFO	aiohttp.access	127.0.0.1 [15/Sep/2025:02:39:15 +0530] "GET /api/status HTTP/1.1" 200 1988 "-" "Python/3.10 aiohttp/3.12.15"
2025-09-15 02:39:15,268	INFO	aiohttp.access	127.0.0.1 [15/Sep/2025:02:39:15 +0530] "GET /api/module_status/mm HTTP/1.1" 200 405 "-" "Python/3.10 aiohttp/3.12.15"

Secure Aggregation

IT21831904 | Weerasinghe K.M

BACKGROUND

- **Main Function:** Secure aggregation ensures the server can compute an aggregate of client Updates without seeing any individual Private contribution
- **Requirement**: It protects client data privacy during aggregation, even if the central server is curious or malicious

RESEARCH GAP

- **Limitations**: Many Secure aggregation protocols are either too heavy for resource constrained IIoT devices and some only exist in theory, and do not account for diverse IIoT environments
- **Challenges**: They often lack robustness to network disruptions or client failures, which can cause and entire training round to collapse

RESEARCH PROBLEM

- How to design a secure Aggregation protocol that is both adaptive and robust enough for scalable IIoT deployments.

SOLUTION:

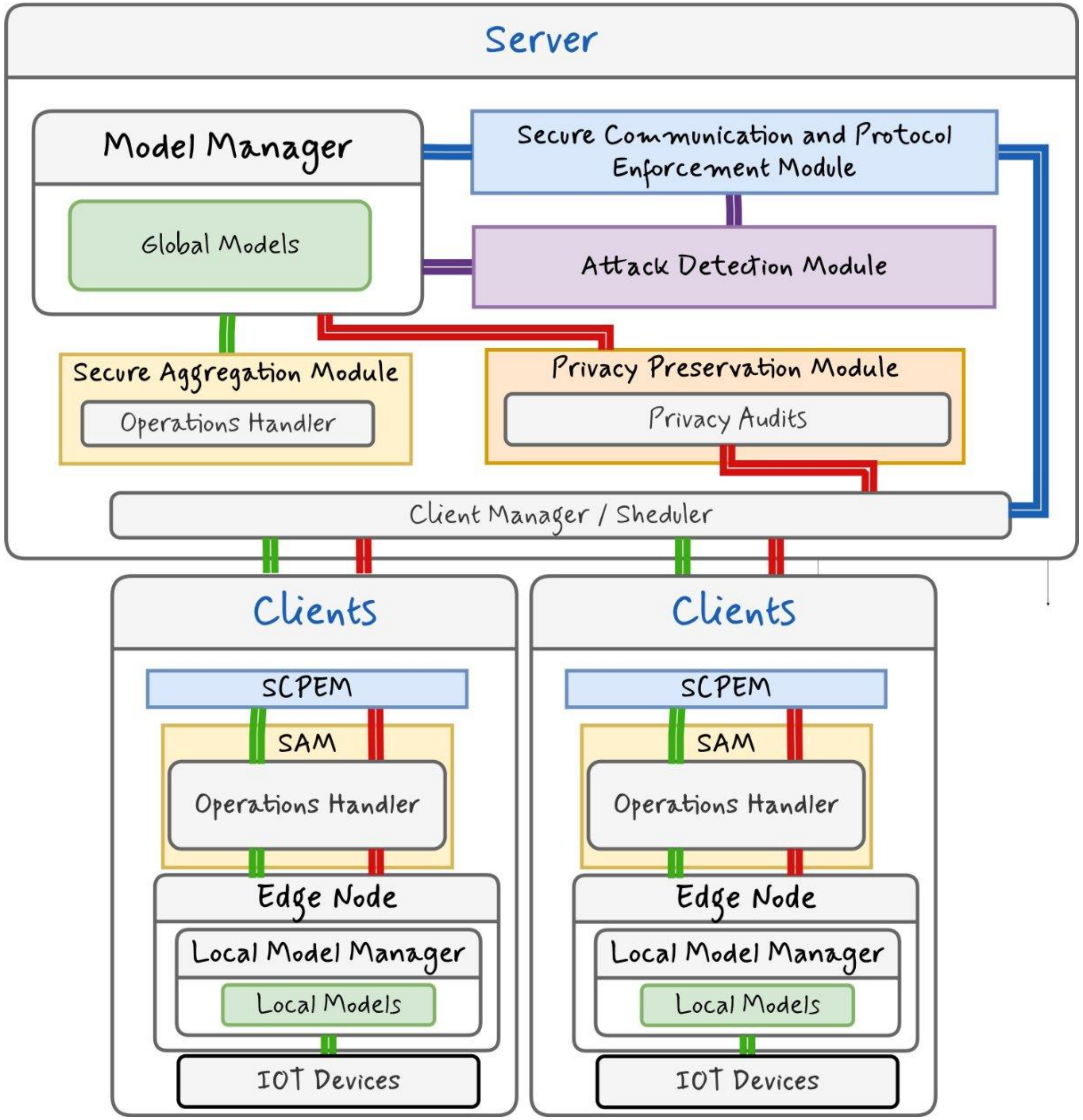
- **Main Objective:** To aggregate client updates securely, ensuring the server learns only the final sum, not individual contributions, while being resilient to client dropouts.
- **Our approach:**
  - **Shamir's Secret Sharing (SSS):** Each client's model update is split into multiple shares. No single share reveals any information.
  - **Threshold Cryptography:** The original secret (the aggregated sum) can only be reconstructed if a minimum number of shares (the "threshold") are combined.
- **Benefits:**
  - **server Blindness:** The server cannot reconstruct any individual update.
  - **Fault Tolerance:** The aggregation succeeds even if some client failures

METHODOLOGY

- **Client-Side Operations Handler:** Splits the local model update into encrypted shares using SSS.

- **Server-Side Operations Handler:** Receives and stores shares from all clients

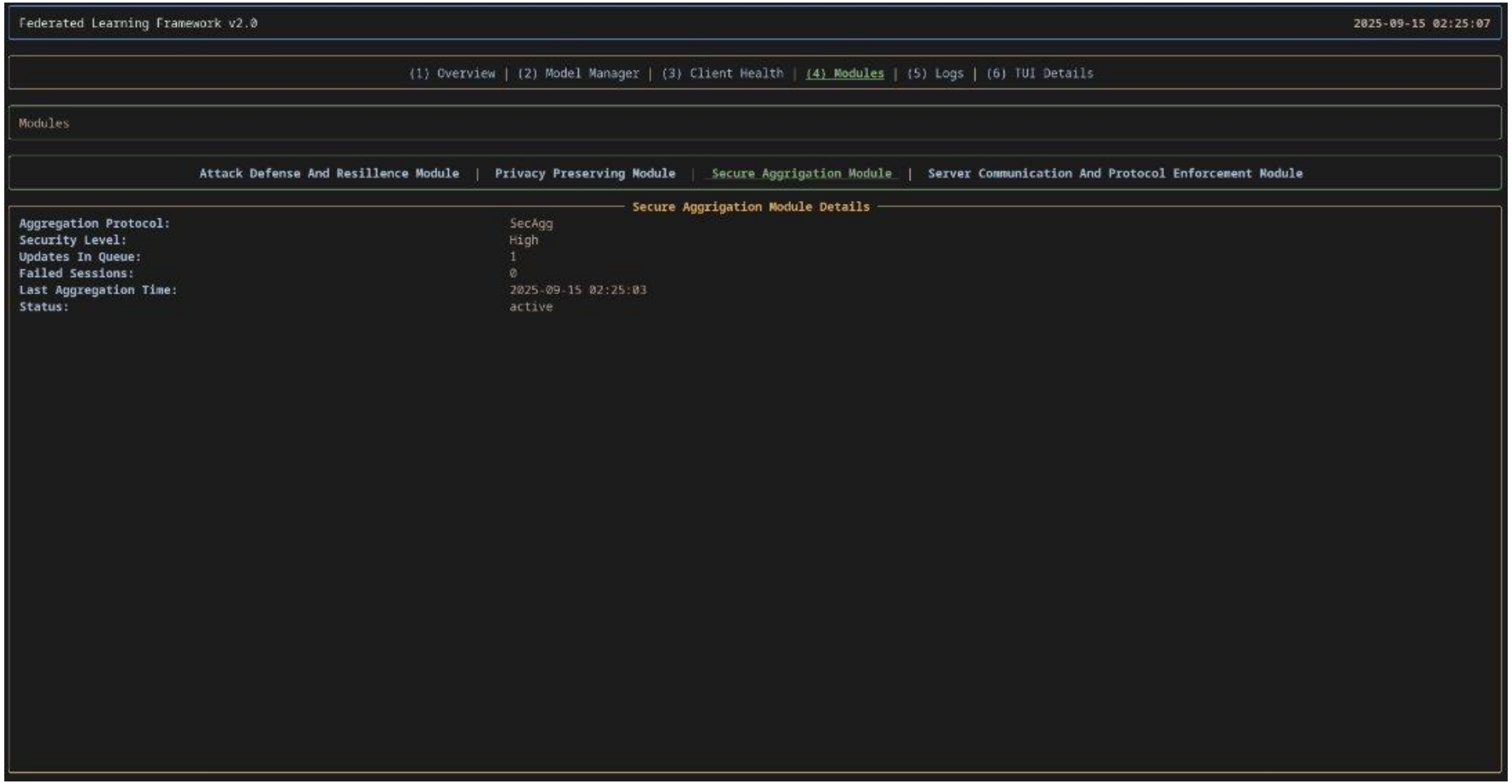
- **Threshold Reconstructor:** A server function that combines shares to reconstruct the final aggregated model update once the required threshold is reached.



SECURE AGGREGATION PROCESS FLOW

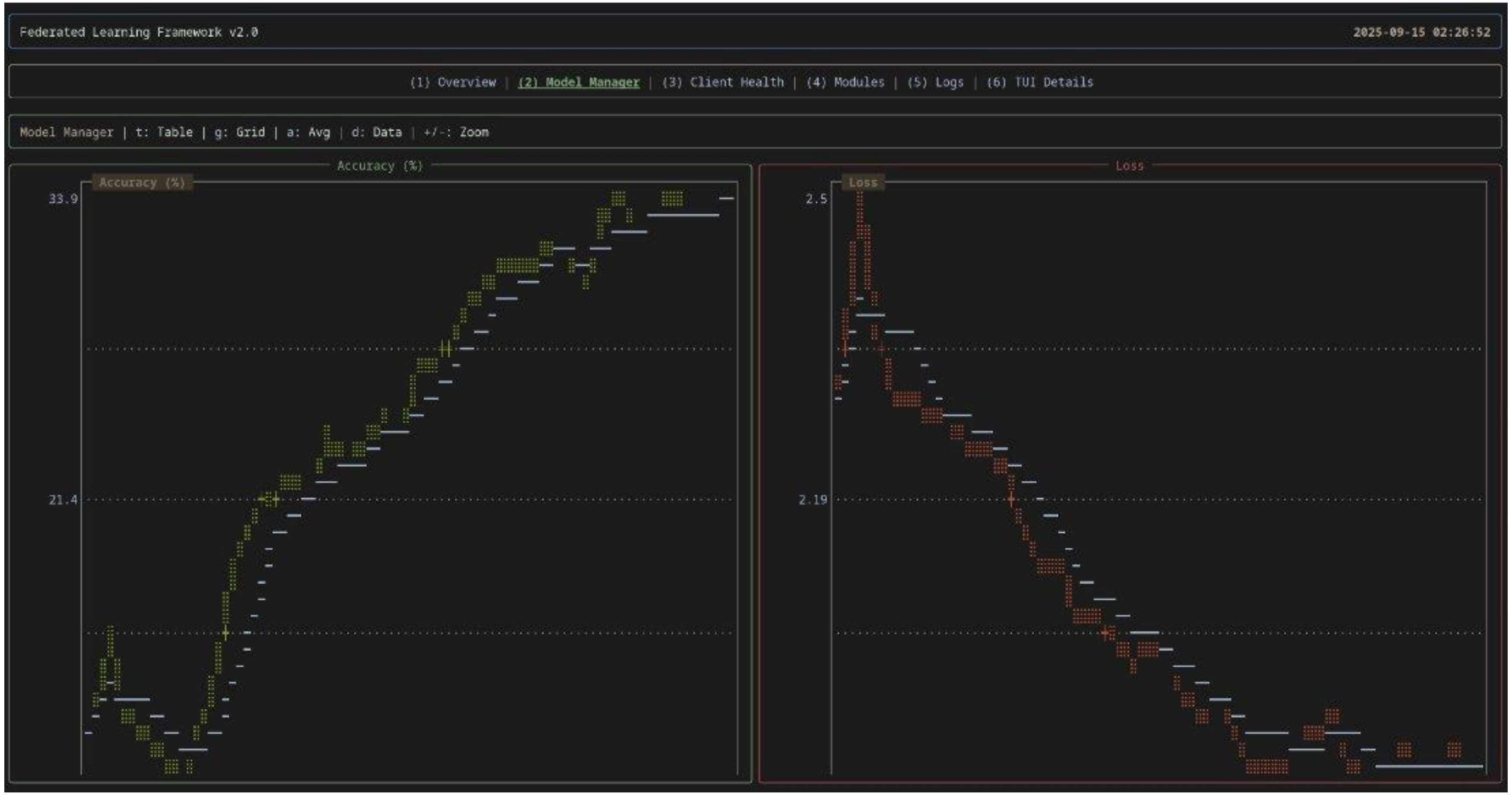
- Each client splits its update into n no of shares.
- Shares are distributed securely to the server.
- The server aggregates the corresponding shares from all clients.
- Once at least t clients have submitted, the server reconstructs the final aggregated update.

SAM MODULE TUI INTERFACE


Federated Learning Framework v2.0
2025-09-15 02:25:07
(1) Overview | (2) Model Manager | (3) Client Health | (4) Modules | (5) Logs | (6) TUI Details
Modules
Attack Defense And Resillence Module | Privacy Preserving Module | Secure Aggrigation Module | Server Communication And Protocol Enforcement Module

```
─────────────── Secure Aggrigation Module Details ───────────────
Aggregation Protocol:                            SecAgg
Security Level:                                  High
Updates In Queue:                                1
Failed Sessions:                                 0
Last Aggregation Time:                           2025-09-15 02:25:03
Status:                                          active
```

AGGREGATION PROCESS



Federated Learning Framework v2.0
2025-09-15 02:26:52
(1) Overview | (2) Model Manager | (3) Client Health | (4) Modules | (5) Logs | (6) TUI Details
Model Manager | t: Table | g: Grid | a: Avg | d: Data | +/-: Zoom
Accuracy (%)
Accuracy (%)
33.9
21.4
Loss
Loss
2.5
2.19

USER TESTING

TECHNOLOGIES

- BACKEND
  - Python
  - Asyncio
- COMMUNICATION
  - GRPC
  - Protocol buffers
  - AIOHTTP
  - AIOHTTP-CORS
- ML & DATA PROCESSING
  - PyTorch
  - Torch Vision
  - NumPy
- CRYPTOGRAPHY & SECURITY
  - PyCryptodrome
  - Secret-sharing
  - X.509 Certificates
- FRONTEND & VALIDATION
  - Rich
  - Prompt Toolkit
- LOGGING AND MONITORING
  - Python-JSON-Logger
  - Logstash

COMMERCIALIZATION

Open-Source

We are Pleased to present the Framework to the industrial IOT

Why Open Source?

- **Transparent by Design**
  Code, roadmap, and issues are fully open and inspectable.
- **Security**
  Privacy and protection mechanisms can be independently audited.
- **For Community-Driven Hardening**
  faster vulnerability discovery and fixes.
- **Faster Adoption & Innovation**
  Teams can test, fork, and contribute improvements.

Commercialization Plan

Aspect	Summary
Product	Open-source IIoT federated learning platform with built-in security, privacy, and real-time defense.
Market Need	Industrial data can’t leave sites; networks are unstable; FL is exposed to tampering and dropouts.
Solution	Secure, privacy-preserving, and dropout-robust FL stack with mTLS, RBAC, DP, HE, and anomaly defense.
Differentiation	IIoT-focused design with governance, lifecycle control, privacy modes, and robust aggregation.
Business Model	Open core

Demonstration

THANK YOU