

# Logbook – Privacy Preservation Module of R25 -039



Project ID: R25 - 039

## **Project Title: Data-Privacy Focused Federated Learning Framework for Industrial IoT**

Student Details:	
Names:	Student IDs:
Mendis H.R.M	IT21822612

Supervisor: Mr. Amila Seneratha

Co-Supervisor: Mr. Tharaniyawarma Kumaralingam

Date of Submission: 2025

## Contents

1. Group Details.....	3
2. Project Details.....	3
3. Communication Methods .....	4
4. Meetings With Supervisors .....	6
5. Task Details .....	7
5.1 Personal task Assigning and Completion .....	7
6. System Details .....	8
6.1 System completion status.....	8
6.2 System Design .....	9
I. System Architecture.....	9
II. Module Architecture .....	10
6.3 System Testing.....	11
6.4 System Codes.....	12
7. GitHub Upload .....	14
8. Documentation.....	14
8.1 Proposal.....	14
8.2 Presentation 1 .....	15
8.3 Presentation 2.....	15
8.4 Final Presentation.....	16
8.4 Final Product .....	17
8.5 Research Paper.....	18
I. Conference Appetence .....	18
9. CDAP upload.....	19
10. Website .....	20
10.1 Development .....	20
10.2 Finalize.....	23

## 1. Group Details

Student Details:		
Names:	Student IDs:	Research Component
Nanayakkara Y.D.T. D	IT21826368	Attack Defense and Resillience
Mendis H.R.M	IT21822612	Privacy Preseravation
Weerasinghe K.M	IT21831904	Secure Aggrigation
Dissanayaka K.D.A.R. A	IT21828348	Secure Communicaiton and Protocol Enforcement

## 2. Project Details

Topic - Data-Privacy Focused Federated Learning Framework for Industrial IoT

Aim – To develop a product that going to full fill the research

Deliverables – Federated Learning Framework designed for industrial internet of things

This project was initiated to develop a secure and private **Federated Learning (FL) framework** specifically for **Industrial IoT (IIoT)** environments.

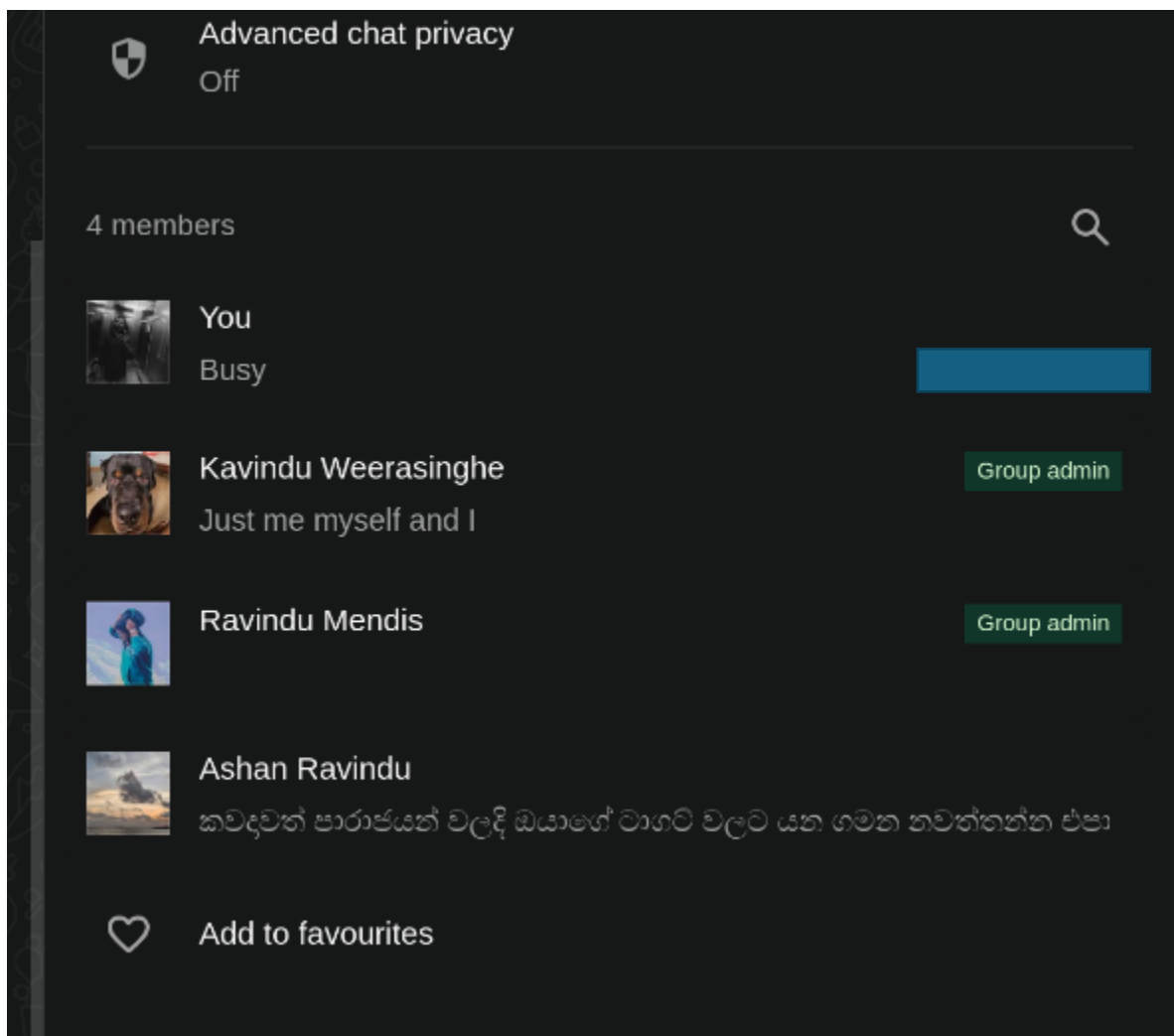
**The Challenge:** Traditional AI methods require centralizing sensitive factory data, which poses major **privacy risks** and clashes with the distributed nature of industrial operations. Existing FL solutions are insufficient because they fail to simultaneously provide robust security, data privacy, and efficient operation on **resource-limited IIoT devices**.

**The Solution:** The developed framework is a multi-layered system that provides **end-to-end protection**.

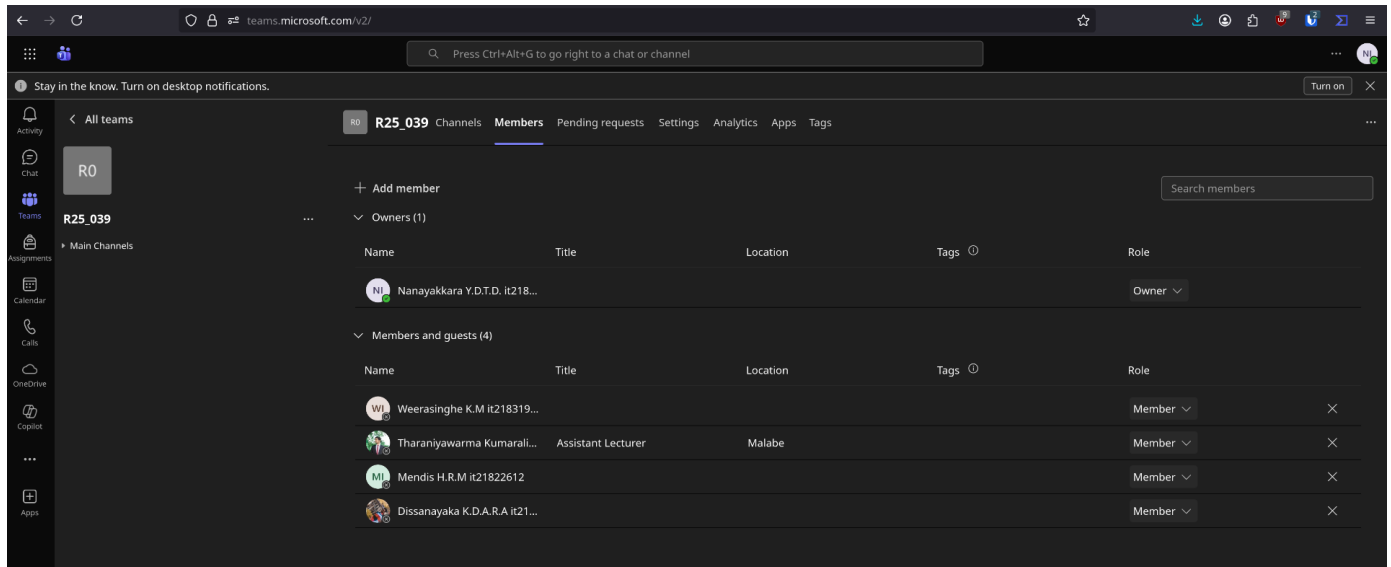
- It uses techniques like **Differential Privacy (DP)** and **Homomorphic Encryption (HE)** to guarantee data confidentiality.
- It implements a robust protocol that uses **client/server validation** to actively block cyber threats such as **Model Poisoning and Byzantine Attacks**.
- The system is optimized for **efficiency** to reduce overhead on IIoT devices.

### 3. Communication Methods

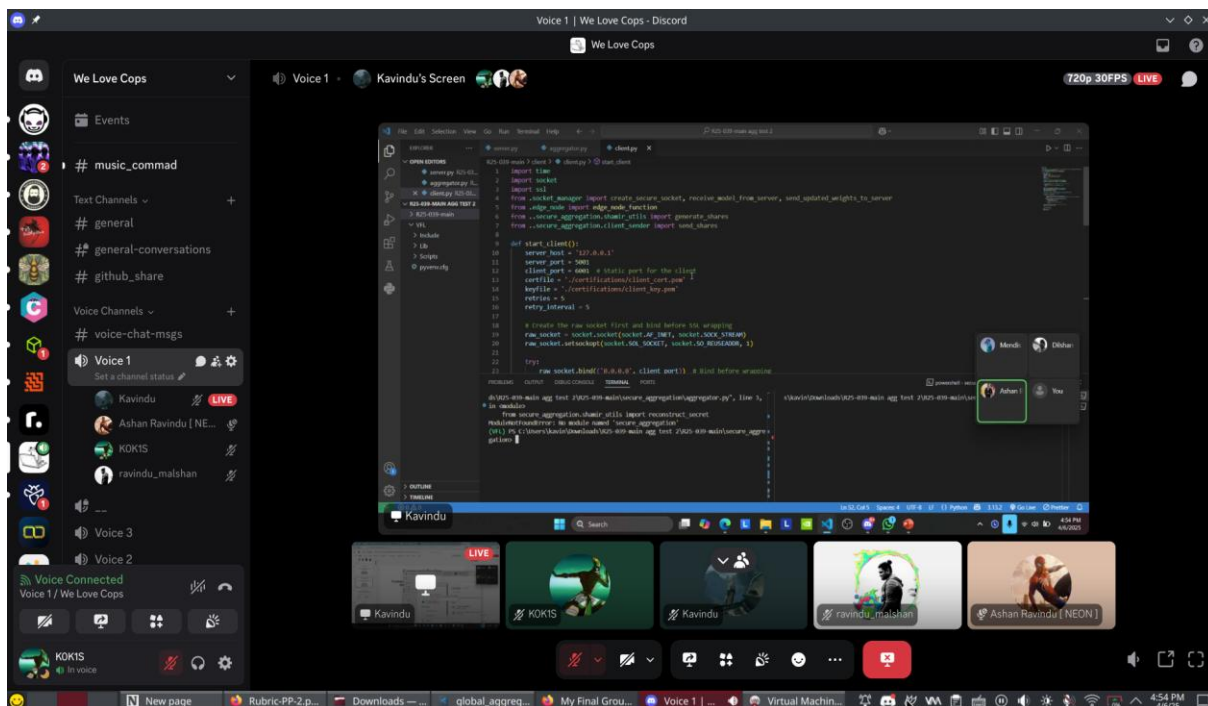
WhatsApp Group – Team



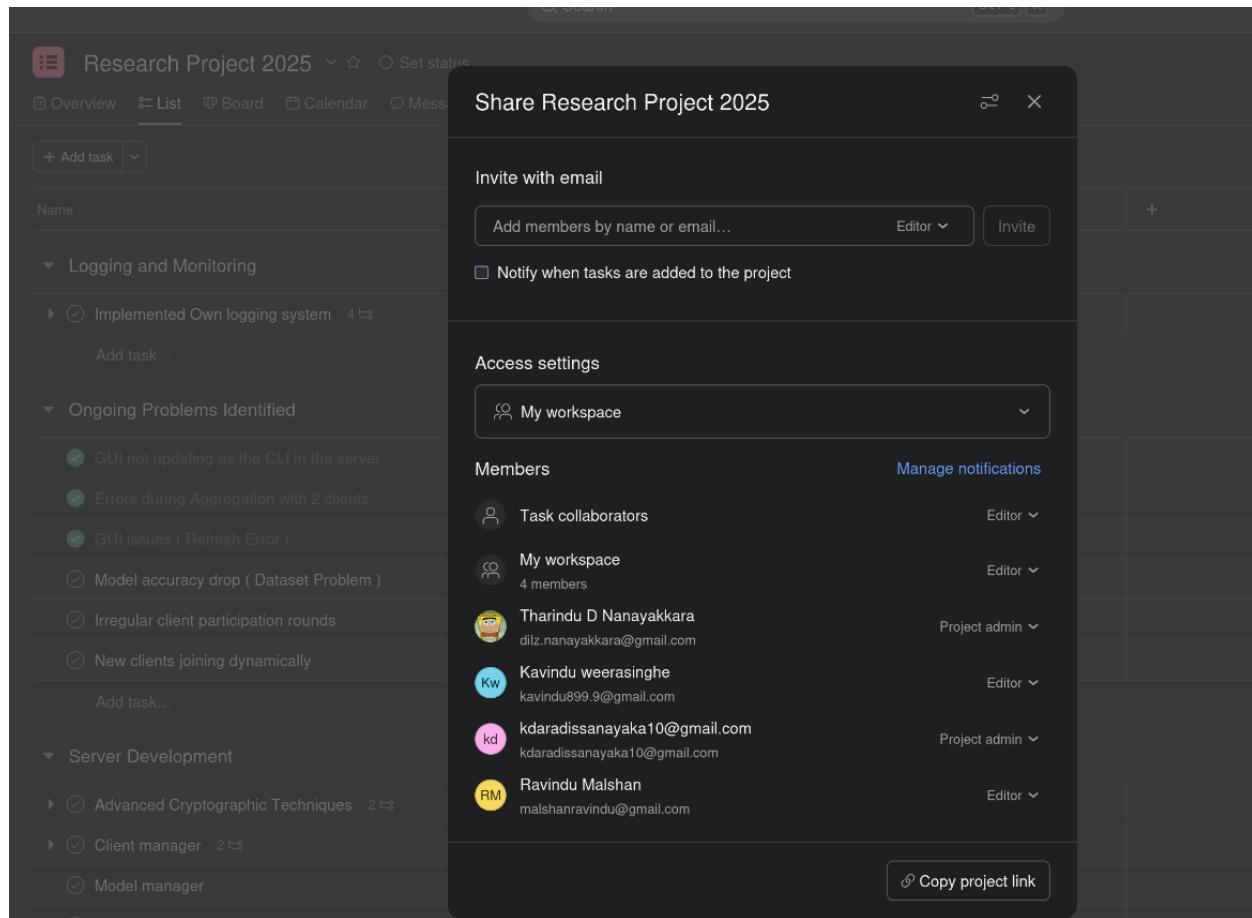
Microsoft Teams - All



## Group Meetings – Discord – Team



## Asana – Task Assigning – Team



## 4. Meetings With Supervisors

All the meetings were conducted in person and only WhatsApp calls were taken to organize the meeting

## 5. Task Details

Research Project 2025					Set status
Overview	List	Board	Calendar	Messages	Note
+ Add task					Filter Sort Group Options
Name	Assignee	Due date	Priority		
Logging and Monitoring					
Implemented Own logging system					
Add task...					
Ongoing Problems Identified					
GUI not updating as the CLI in the server					
Errors during Aggregation with 2 clients					
GUI issues ( Refresh Error )					
Model accuracy drop ( Dataset Problem )					
Irregular client participation rounds					
New clients joining dynamically					
Add task...					
Server Development					
Advanced Cryptographic Techniques					
Client manager					

Privacy Preservation Module was assigned.

### 5.1 Personal task Assigning and Completion

Add task...				
Privacy Preservation Module (PPM)				
Differential Privacy values reported ( $\epsilon$ , $\delta$ ).	RM Ravindu Ma...			
Homomorphic Encryption functionality demonstrated.	RM Ravindu Ma...			
Accuracy comparison: With vs. Without privacy methods.	RM Ravindu Ma...			
Convergence curve plots included.	RM Ravindu Ma...			
Memory & time benchmarks of hybrid DP+HE scheme.	RM Ravindu Ma...			
Communication overhead metrics.	RM Ravindu Ma...			
graph or table showing privacy utility trade off	RM Ravindu Ma...			

## 6. System Details

### 6.1 System completion status

Finished

PPM TUI

```
2025-04-07 00:54:03.110 - DEBUG - [DP DMT] Noise Multiplier set to 0.2
connected to server at 127.0.0.1:5801 from static port 6001.
2025-04-07 00:54:15.729 - INFO - Waiting to receive initial model from the server.
2025-04-07 00:54:15.733 - INFO - Expecting 678660 bytes of model data.
2025-04-07 00:54:15.738 - INFO - Successfully received 678660 bytes of model data.
2025-04-07 00:54:15.739 - INFO - [RECEIVE] Successfully deserialized model weights (bytes)
2025-04-07 00:54:15.739 - INFO - [MODEL] Loading data and starting training
2025-04-07 00:54:17.836 - DEBUG - [MODEL] Creating new model instance

C:\Users\User\Videos\FI_DP_Tabular\FI_DP_Tabular>python main.py --mode FL

--- Running mode: FL ---
Round 1/10 | Mode: FL | Acc: 76.13% | Comm: 215000 bytes
Round 2/10 | Mode: FL | Acc: 76.77% | Comm: 215000 bytes
Round 3/10 | Mode: FL | Acc: 76.63% | Comm: 215000 bytes
Round 4/10 | Mode: FL | Acc: 76.63% | Comm: 215000 bytes
Round 5/10 | Mode: FL | Acc: 76.63% | Comm: 215000 bytes
Round 6/10 | Mode: FL | Acc: 76.57% | Comm: 215000 bytes
Round 7/10 | Mode: FL | Acc: 76.57% | Comm: 215000 bytes
Round 8/10 | Mode: FL | Acc: 76.57% | Comm: 215000 bytes
Round 9/10 | Mode: FL | Acc: 76.57% | Comm: 215000 bytes
Round 10/10 | Mode: FL | Acc: 76.57% | Comm: 215000 bytes

C:\Users\User\Videos\FI_DP_Tabular\FI_DP_Tabular>python main.py --mode HE

--- Running mode: HE ---
Round 1/10 | Mode: HE | Acc: 76.37% | Comm: 215000 bytes
Round 2/10 | Mode: HE | Acc: 76.60% | Comm: 215000 bytes
Round 3/10 | Mode: HE | Acc: 76.30% | Comm: 215000 bytes
Round 4/10 | Mode: HE | Acc: 76.40% | Comm: 215000 bytes
Round 5/10 | Mode: HE | Acc: 76.35% | Comm: 215000 bytes
Round 6/10 | Mode: HE | Acc: 76.37% | Comm: 215000 bytes
Round 7/10 | Mode: HE | Acc: 76.37% | Comm: 215000 bytes
Round 8/10 | Mode: HE | Acc: 76.37% | Comm: 215000 bytes
Round 9/10 | Mode: HE | Acc: 76.37% | Comm: 215000 bytes
Round 10/10 | Mode: HE | Acc: 76.37% | Comm: 215000 bytes

C:\Users\User\Videos\FI_DP_Tabular\FI_DP_Tabular>python main.py --mode DPHE

warnings.warn(
C:\Users\User\Videos\FI_DP_Tabular\FI_DP_Tabular\utils\train.py:31: UserWarning: Full backward hook is
Inputs require gradients. See https://docs.pytorch.org/docs/main/generated/torch.nn.Module.html#torch.
loss.backward()
Round 1/10 | Mode: DPHE | Acc: 75.34% | Comm: 215000 bytes
Round 2/10 | Mode: DPHE | Acc: 75.92% | Comm: 215000 bytes
Round 3/10 | Mode: DPHE | Acc: 75.43% | Comm: 215000 bytes
Round 4/10 | Mode: DPHE | Acc: 75.43% | Comm: 215000 bytes
Round 5/10 | Mode: DPHE | Acc: 75.23% | Comm: 215000 bytes
Round 6/10 | Mode: DPHE | Acc: 75.43% | Comm: 215000 bytes
Round 7/10 | Mode: DPHE | Acc: 75.25% | Comm: 215000 bytes
Round 8/10 | Mode: DPHE | Acc: 75.40% | Comm: 215000 bytes
Round 9/10 | Mode: DPHE | Acc: 75.04% | Comm: 215000 bytes
Round 10/10 | Mode: DPHE | Acc: 75.43% | Comm: 215000 bytes

C:\Users\User\Videos\FI_DP_Tabular\FI_DP_Tabular>

C:\Users\User\Videos\FI_DP_Tabular\FI_DP_Tabular>python main.py --mode DP

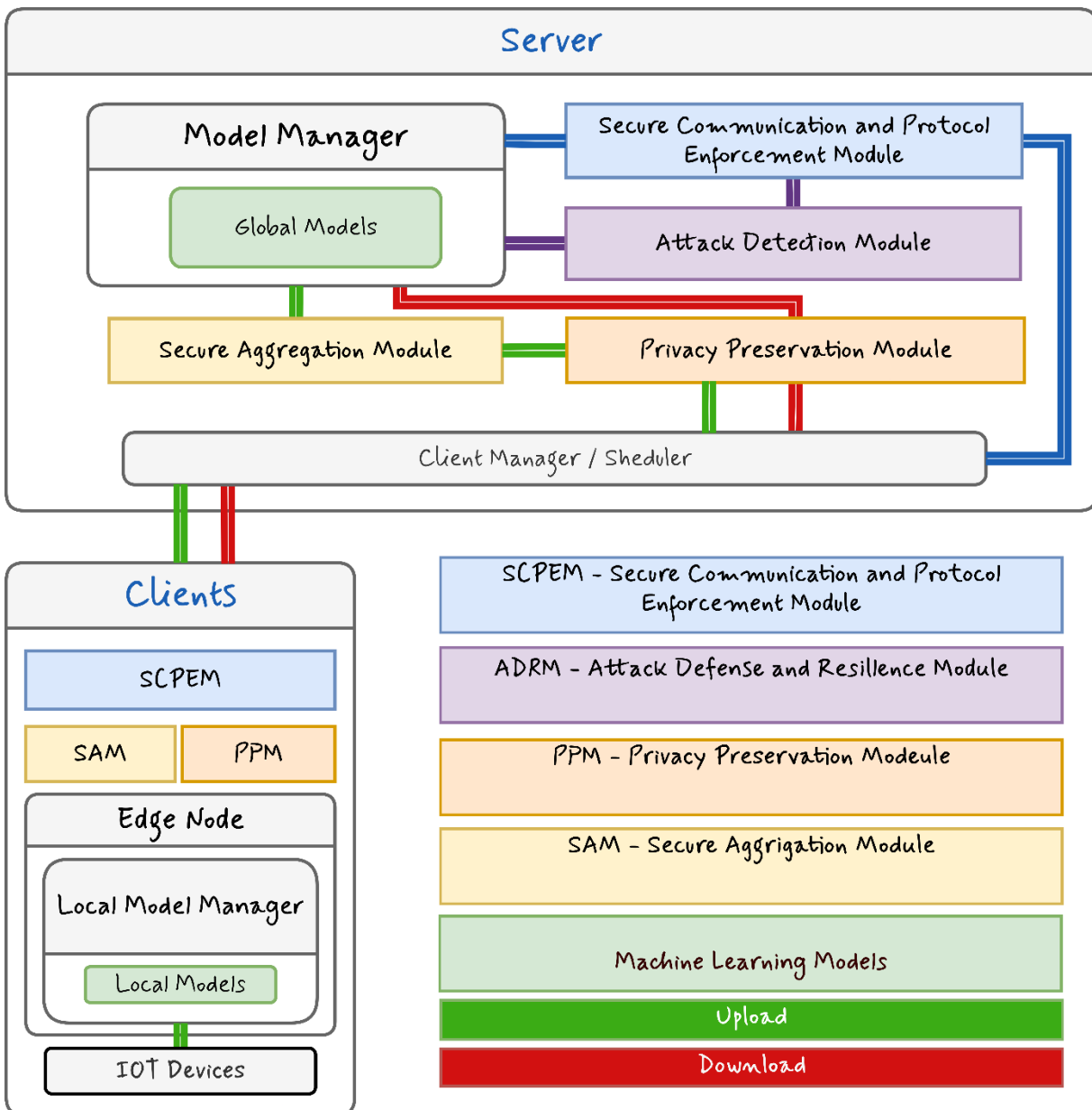
--- Running mode: DP ---
C:\Users\User\Videos\FI_DP_Tabular\FI_DP_Tabular\venv\Lib\site-packages\opacus\privacy_engine.py:56
n as it allows for much faster training performance, but remember to turn it on and retrain one la
warnings.warn(
C:\Users\User\Videos\FI_DP_Tabular\FI_DP_Tabular\utils\train.py:31: UserWarning: Full backward hook
Inputs require gradients. See https://docs.pytorch.org/docs/main/generated/torch.nn.Module.html#torch.
loss.backward()
Round 1/10 | Mode: DP | Acc: 24.12% | Comm: 215000 bytes
Round 2/10 | Mode: DP | Acc: 76.60% | Comm: 215000 bytes
Round 3/10 | Mode: DP | Acc: 24.12% | Comm: 215000 bytes
Round 4/10 | Mode: DP | Acc: 75.91% | Comm: 215000 bytes
Round 5/10 | Mode: DP | Acc: 75.92% | Comm: 215000 bytes
Round 6/10 | Mode: DP | Acc: 76.80% | Comm: 215000 bytes
Round 7/10 | Mode: DP | Acc: 76.41% | Comm: 215000 bytes
Round 8/10 | Mode: DP | Acc: 76.49% | Comm: 215000 bytes
Round 9/10 | Mode: DP | Acc: 76.29% | Comm: 215000 bytes
Round 10/10 | Mode: DP | Acc: 76.37% | Comm: 215000 bytes
```

Federated Learning Framework v2.0		2025-04-15 02:21:18
(1) Overview   (2) Model Manager   (3) Client Health   (4) Modules   (5) Logs   (6) TUI Details		
Modules		
Attack Defense And Resilience Module   <u>Privacy Preserving Module</u>   Secure Aggregation Module   Server Communication And Protocol Enforcement Module		
Privacy Preserving Module Details		
Module Name:	PPM	
Status:	active	
Version:	1.0	
Is On:	True	
Is Active:	True	
Description:	Privacy Preserving Mechanism for policy auditing.	

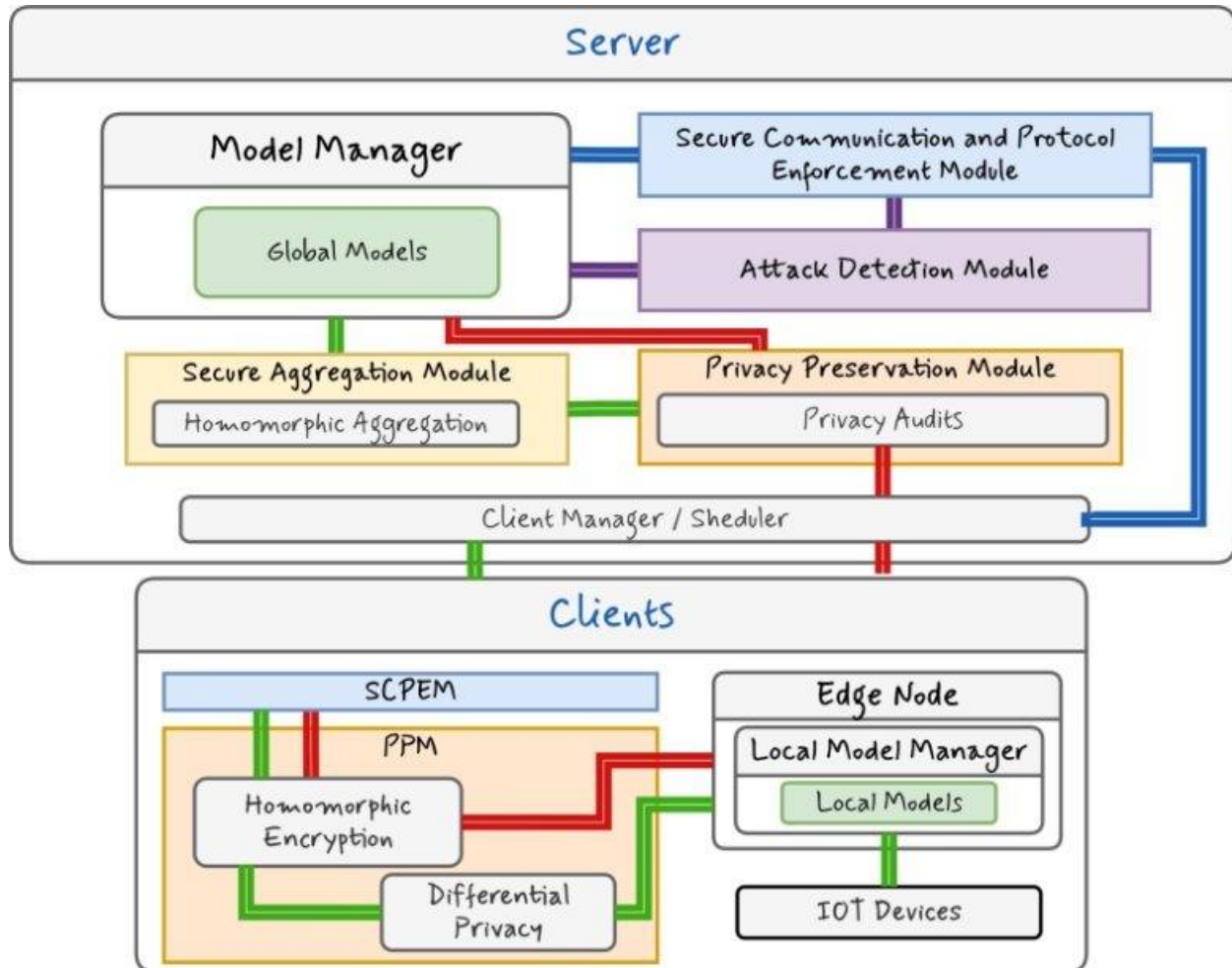


## 6.2 System Design

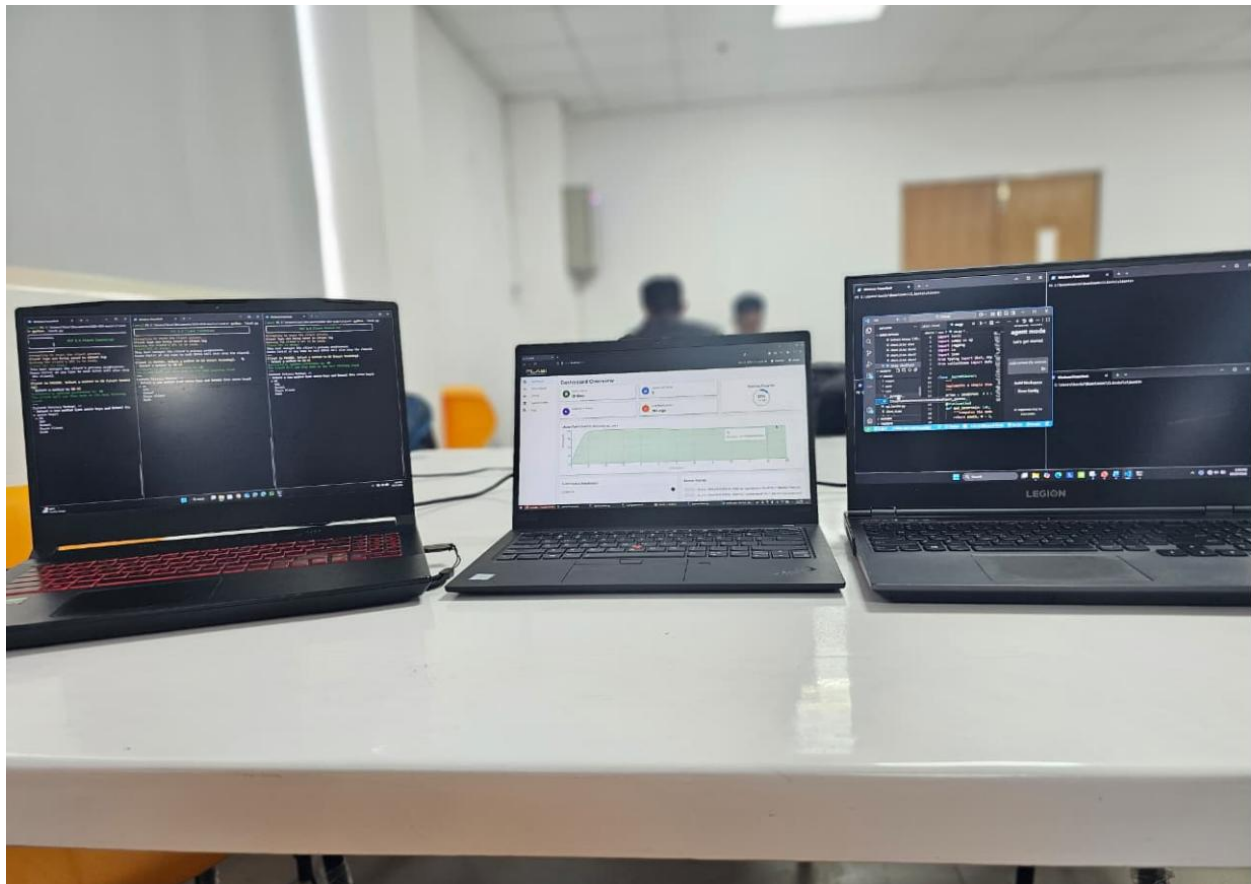
### I. System Architecture



## II. Module Architecture



### 6.3 System Testing



## 6.4 System Codes

### PPM – Server

```

1 # server/ppm/ppm.py
2
3 import logging
4 from typing import Dict, Any
5
6 from log_manager.log_manager import ContextAdapter
7
8
9 class PPM:
10     """
11     Privacy-Preserving Mechanism (PPM) for Federated Learning.
12     This class is responsible for auditing privacy protocols and guiding the
13     server's aggregation strategy based on policy, not for adding noise itself.
14     """
15
16     def __init__(self, cfg: Dict[str, Any]):
17         """
18         Initializes the PPM with configuration parameters for both DP and HE.
19
20         Args:
21             cfg (Dict[str, Any]): The configuration dictionary, expected to contain
22                 'privacy' sub-keys for 'dp' and 'he'.
23         """
24         # Load DP parameters from config, with safe fallbacks
25         dp_cfg = cfg.get("privacy", {}).get("dp", {})
26         self.epsilon = dp_cfg.get("epsilon", 1.0)
27         self.delta = dp_cfg.get("delta", 1e-5)
28
29         # Load HE parameters from config
30         he_cfg = cfg.get("privacy", {}).get("he", {})
31         self.he_active = he_cfg.get("active", False)
32
33         self.logger = logging.getLogger(self.__class__.__name__)
34         self.logger = ContextAdapter(self.logger, {"component": self.__class__.__name__})
35         self.logger.info(
36             f"PPM initialized. DP: (epsilon={self.epsilon}, delta={self.delta}). "
37             f"HE active: {self.he_active}"
38         )
39
40     def check_aggregation_policy(self) -> bool:
41         """
42         Performs a privacy audit to determine if homomorphic aggregation should be used.
43         The policy checks if Homomorphic Encryption is active.
44
45         Returns:
46             bool: True if the policy allows for homomorphic aggregation, False otherwise.
47         """

```

### Homomorphic Aggregation

```

1 import logging
2 import torch
3 import io
4 from typing import Dict, Any
5
6 class HomomorphicEncryption:
7     """
8     A placeholder class for Homomorphic Encryption (HE) on the server side.
9     This module is responsible for decrypting model updates received from clients.
10
11     This mock version now correctly handles binary data by serializing/deserializing
12     PyTorch state dictionaries to and from byte streams.
13     """
14
15     def __init__(self):
16         self.logger = logging.getLogger(self.__class__.__name__)
17         self.logger.info("Server-side HomomorphicEncryption module initialized.")
18
19     def decrypt_model_state(self, encrypted_bytes: bytes) -> Dict[str, Any]:
20         """
21         Decrypts an encrypted model state received from a client.
22
23         This mock implementation deserializes the byte stream back into a PyTorch
24         state dictionary, simulating decryption.
25
26         Args:
27             encrypted_bytes (bytes): The encrypted model state from the client.
28
29         Returns:
30             Dict[str, Any]: The decrypted model state dictionary.
31         """
32         self.logger.info("Decrypting model state from client.")
33
34         buffer = io.BytesIO(encrypted_bytes)
35         # **FIX:** Added weights_only=False to allow loading the full state dictionary.
36         state_dict = torch.load(buffer, map_location='cpu', weights_only=False)
37
38         self.logger.info("Model state decrypted successfully.")
39         return state_dict

```

### PPM – Clients

## Differential Privacy

```

newclients > admrcient > ppm > dp.py
1 # dp.py
2
3 import torch
4 import numpy as np
5 import logging
6 from typing import Dict, Any
7
8 class DifferentialPrivacy:
9     """
10     Applies local differential privacy to model updates by adding Gaussian noise.
11     """
12     def __init__(self, epsilon: float, clipping_norm: float = 1.0):
13         If epsilon <= 0 or clipping_norm <= 0:
14             raise ValueError("Epsilon and clipping norm must be positive values.")
15
16         self.epsilon = epsilon
17         self.clipping_norm = clipping_norm
18         self.std_dev = self.clipping_norm / self.epsilon
19         self.logger = logging.getLogger(self.__class__.__name__)
20         self.logger.info("DifferentialPrivacy initialized with epsilon={self.epsilon}, clipping_norm={self.clipping_norm}, std_dev={self.std_dev:.4f}")
21
22     def add_noise(self, state_dict: Dict[str, Any]) -> Dict[str, Any]:
23         """
24         Clips the norm of the entire model update and then adds Gaussian noise
25         to each parameter.
26         """
27         self.logger.info("Applying differential privacy: Clipping and adding Gaussian noise.")
28
29         # 1. Flatten all parameters into a single vector to calculate the L2 norm
30         flat_params = torch.cat([p.flatten() for p in state_dict.values()])
31         total_norm = torch.linalg.norm(flat_params)
32
33         # 2. Calculate the clipping factor to scale the update if its norm exceeds the threshold
34         clip_factor = min(1.0, self.clipping_norm / (total_norm + 1e-6))
35
36         self.logger.info(f"Update norm: {total_norm:.4f}, Clipping factor: {clip_factor:.4f}")
37
38         noisy_state_dict = {}
39         for key, param in state_dict.items():
40             # 3. Apply the clipping factor to each parameter
41             clipped_param = param * clip_factor
42
43             # 4. Add Gaussian noise scaled to the clipping norm
44             noise = torch.randn_like(clipped_param) * self.std_dev

```

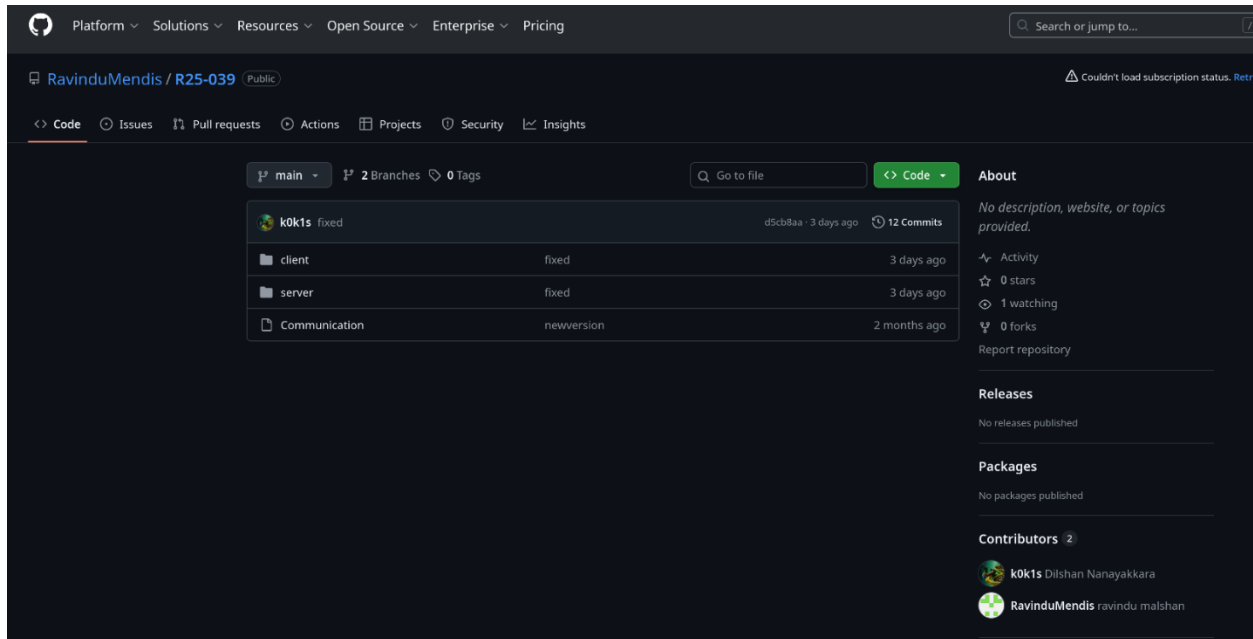
## Homomorphic Encryption

```

newclients > admrcient > ppm > he.py
1 # he.py
2
3 import torch
4 import logging
5 import io
6 from typing import Dict, Any
7
8 class HomomorphicEncryption:
9     """
10     A placeholder class for Homomorphic Encryption (HE).
11     This mock version serializes/deserializes PyTorch state dictionaries
12     to and from byte streams, simulating the process.
13     """
14     def __init__(self):
15         self.logger = logging.getLogger(self.__class__.__name__)
16         self.logger.info("HomomorphicEncryption module initialized.")
17
18     def encrypt_model_state(self, state_dict: Dict[str, Any]) -> bytes:
19         """
20         Mocks the encryption of a model state dictionary.
21         """
22         self.logger.info("Encrypting model state with Homomorphic Encryption.")
23         buffer = io.BytesIO()
24         torch.save(state_dict, buffer)
25         encrypted_bytes = buffer.getvalue()
26
27         self.logger.info("Model state encrypted. Sending to server.")
28         return encrypted_bytes
29
30     def decrypt_model_state(self, encrypted_bytes: bytes) -> Dict[str, Any]:
31         """
32         Mocks the decryption of an encrypted model state.
33         """
34         self.logger.info("Decrypting model state from server.")
35
36         buffer = io.BytesIO(encrypted_bytes)
37         state_dict = torch.load(buffer, map_location='cpu')
38
39         self.logger.info("Model state decrypted successfully.")
40         return state_dict

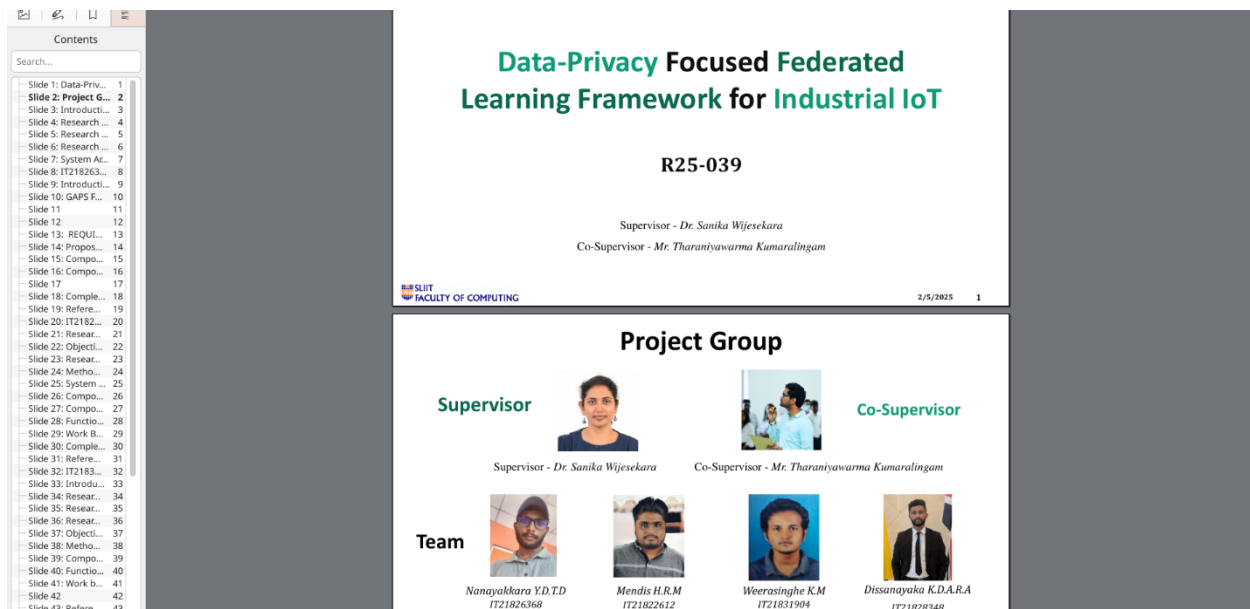
```

## 7. GitHub Upload



## 8. Documentation

### 8.1 Proposal



## 8.2 Presentation 1

Contents

Search...

- Slide 1: Data-Priv... 1
- Slide 2: Project Gr... 2
- Slide 3: Introduct... 3
- Slide 4: Research... 4
- Slide 5: Research... 5
- Slide 6: Research... 6
- Slide 7: System Ar... 7
- Slide 8: IT2182612... 8
- Slide 9: Introduct... 9
- Slide 10: GAPS F... 10
- Slide 11... 11
- Slide 12... 12
- Slide 13: REQU... 13
- Slide 14: Propo... 14
- Slide 15: Compo... 15
- Slide 16: Compo... 16
- Slide 17... 17
- Slide 18: Comple... 18
- Slide 19: Refere... 19
- Slide 20: IT2182... 20
- Slide 21: Resear... 21
- Slide 22: Object... 22
- Slide 23: Resear... 23
- Slide 24: Metho... 24
- Slide 25: System... 25
- Slide 26: Compo... 26
- Slide 27: Compo... 27
- Slide 28: Functio... 28
- Slide 29: Work B... 29
- Slide 30: Comple... 30
- Slide 31: Refere... 31
- Slide 32: IT2183... 32
- Slide 33: Introdu... 33
- Slide 34: Resear... 34
- Slide 35: Resear... 35
- Slide 36: Resear... 36
- Slide 37: Object... 37
- Slide 38: Metho... 38
- Slide 39: Compo... 39
- Slide 40: Functio... 40
- Slide 41: Work b... 41

Real-world testing

There is a lack of large scale IoT datasets for testing privacy preserving techniques under real-world conditions. Existing research often relies on synthetic data, limiting the generalizability of results.

Regulatory compliance

Privacy preserving methods need to be compatible with existing data protection laws like GDPR, but there's a gap in evaluating how these solutions perform in compliance contexts.

Resistance to privacy attacks

While current privacy preserving methods aim to protect data, the robustness of these methods against evolving privacy attacks in IIoT remains insufficiently addressed.

Energy consumption and efficiency

Many privacy-preserving methods are computationally intensive, posing significant challenges to resource-constrained IIoT devices, affecting their energy efficiency.

**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.

**System Architecture**



## 8.3 Presentation 2

Contents

Search...

- > OVERALL 1
- > ADMIN COMPONENT... 15
- > COMPONENT 2 PRL... 23
- > COMPONENT 3 SEC A... 33
- > COMPONENT 4 SCPM... 40
- > GENERAL END SLIDES 48

# DATA-PRIVACY

## FOCUSED

# FEDERATED LEARNING FRAMEWORK


## FOR

# INDUSTRIAL IOT


R25 - 039

**PROJECT GROUP**

**Team**




Nanayakkara Y.D.T.D  
IT21826368



Mendis H.R.M  
IT21822612

**Supervisors**



Mr. Amila Nuwan Senarathne  
Supervisor

## 8.4 Final Presentation

Contents

Search...

- Slide 1: DATA-PR... 1
- Slide 2: PROJECT ... 2**
- Slide 3: FRAMEW... 3
- Slide 4: PROBLEMS 4
- Slide 5: SOLUTIONS 5
- Slide 6: SYSTEM A... 6
- Slide 7: System W... 7
- Slide 8: Attack De... 8
- Slide 9: RESEARC... 9
- Slide 10: SOLUTL... 10
- Slide 11: ... 11
- Slide 12: Termin... 12
- Slide 13 ... 13
- Slide 14: Attack... 14
- Slide 15: DETECT... 15
- Slide 16: Logging 16
- Slide 17: Privacy ... 17
- Slide 18 ... 18
- Slide 19: SOLUTL... 19
- Slide 20: PPM Ac... 20
- Slide 21: METHO... 21
- Slide 22: FUNCTL... 22
- Slide 23: FUNCTL... 23
- Slide 24: PROOF... 24
- Slide 25 ... 25
- Slide 26: Secure ... 26
- Slide 27 ... 27
- Slide 28 ... 28
- Slide 29: Module... 29
- Slide 30: SERVER... 30
- Slide 31: OVERA... 31
- Slide 32: Log ma... 32
- Slide 33: Secure ... 33
- Slide 34 ... 34
- Slide 35 ... 35
- Slide 36: METHO... 36
- Slide 37 ... 37
- Slide 38: SAM M... 38
- Slide 39: AGGRE... 39
- Slide 40: USER T... 40
- Slide 41 ... 41
- Slide 42: We are ... 42
- Slide 43: We are ... 43

**DATA-PRIVACY**

**FOCUSED**

**FEDERATED LEARNING FRAMEWORK**

**FOR**

**INDUSTRIAL IOT**



**Final Presentation**

R25 - 039

**PROJECT GROUP**

**Supervisors**



Mr. Anila  
Nuwan  
Senarathne  
Supervisor



Mr. Tharaniwanna  
Kumaraalingam  
Co-Supervisor

**Team**



Nanayakkara Y.D.T.D  
IT21826368



Mendis H.R.M  
IT21822612



Dissanayaka K.D.A.R.A  
IT21828348

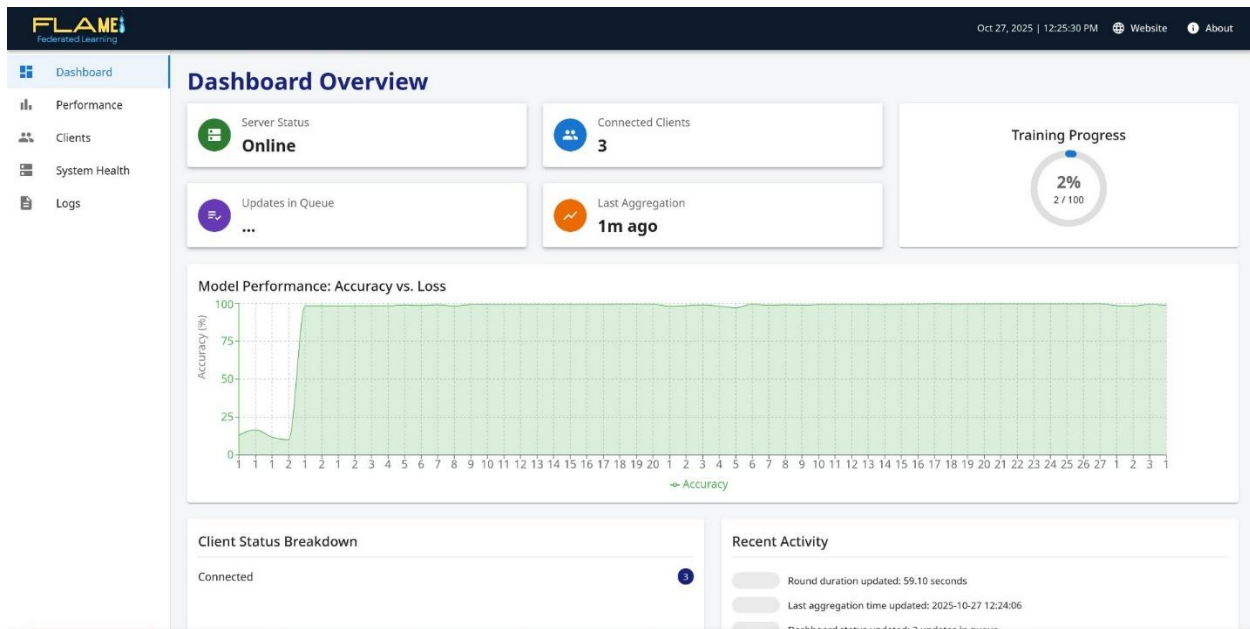


Weerasinghe K.M  
IT21831904

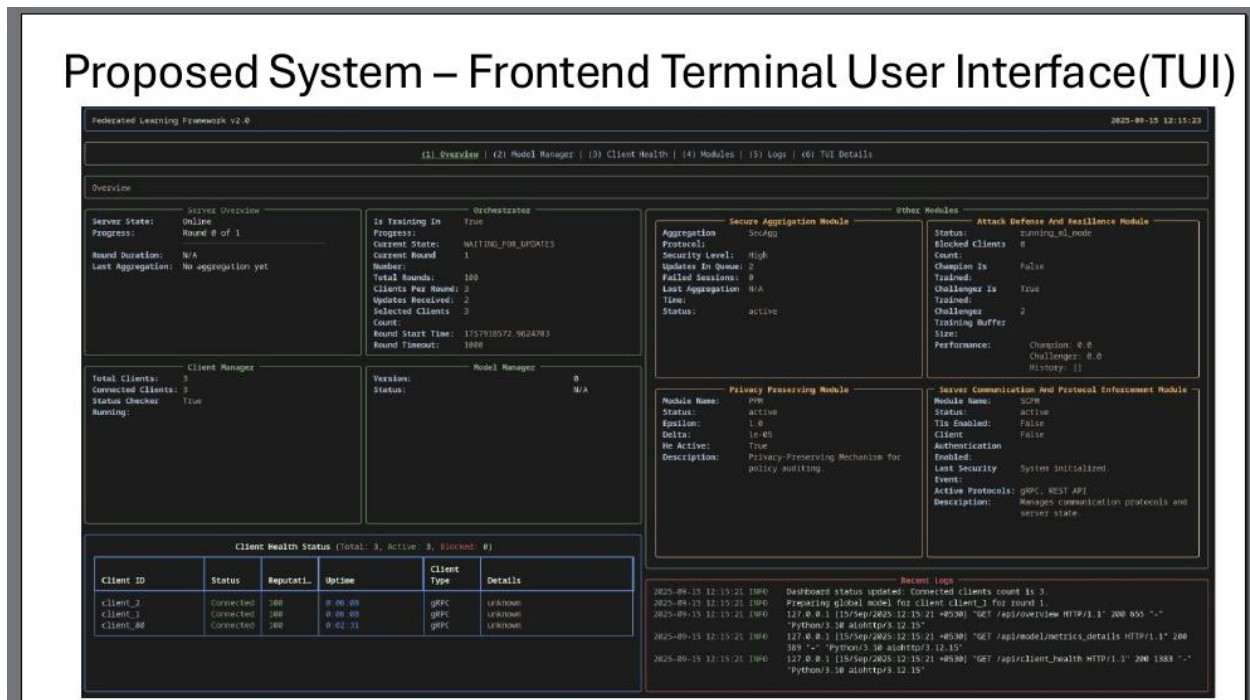


## 8.4 Final Product

### Web Portal Frontend



### Terminal User interface (Frontend)



## 8.5 Research Paper

## I. Conference Appetence

To Tharindu D Nanayakkara <dilz.nanayakkara@gmail.com> 

10/29/25, 11:27 AM

## Acceptance Notification

Dear Tharindu D Nanayakkara,

Congratulations! We are pleased to inform you that your paper has been accepted as a regular paper to be presented at the 7th International Conference on Advancements in Computing 2025.

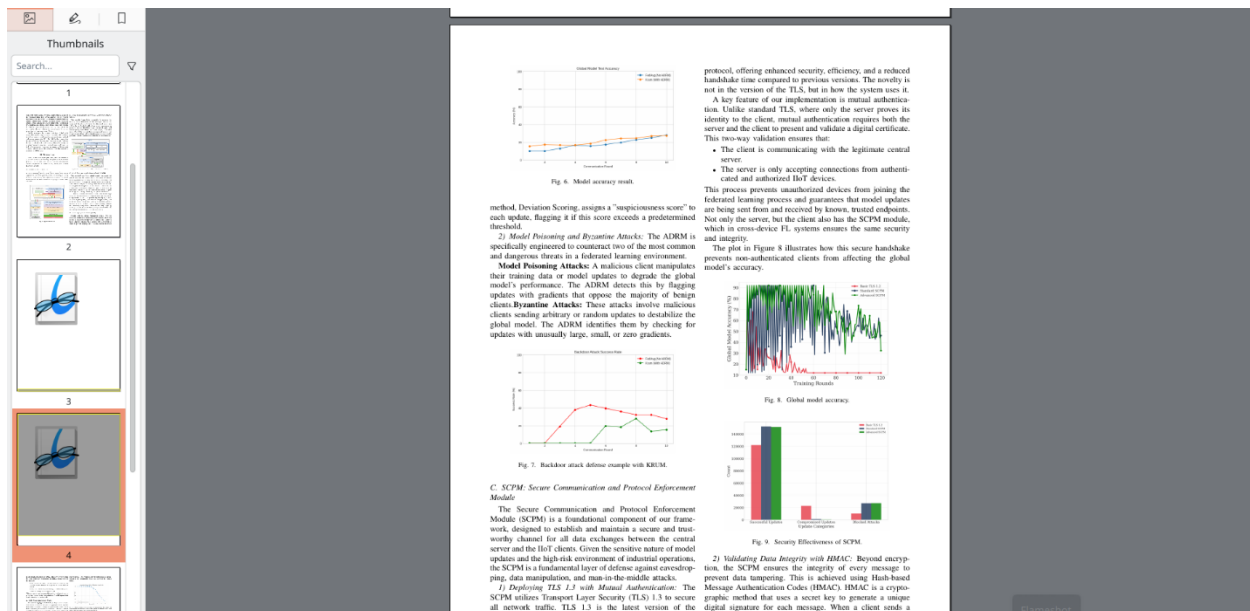
Paper ID: 469

Paper Title: Data-privacy based Federated Learning Framework for Industrial IOT

Please visit <https://cm33.research.microsoft.com/7ICAC2025/Submission/Index> to view the reviews given during the double-blind review process.


When preparing the camera-ready version of your paper, please address all the review comments and follow the camera-ready guidelines given in the <https://icac.lk/for-authors>

Please note that the camera-ready deadline is 10th November 2025 and camera-ready submission portal on CMT will be available starting from 22nd October 2025.















## 9. CDAP upload



**CDAPSubmissionCloud**   
Private group

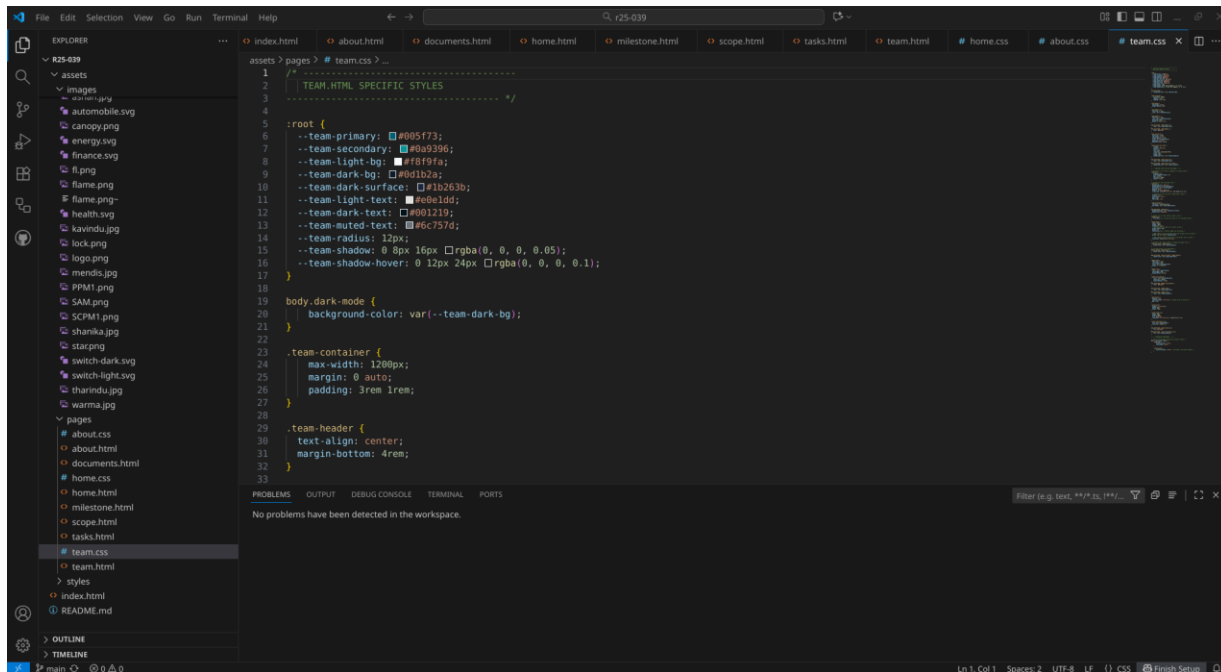
[+ New](#)
[Upload](#)
[Edit in grid view](#)
[Share](#)
[Copy link](#)
[Add shortcut to OneDrive](#)
[Download](#)
[Export to Excel](#)
[Automate](#)
[Integrate](#)
[Sync](#)

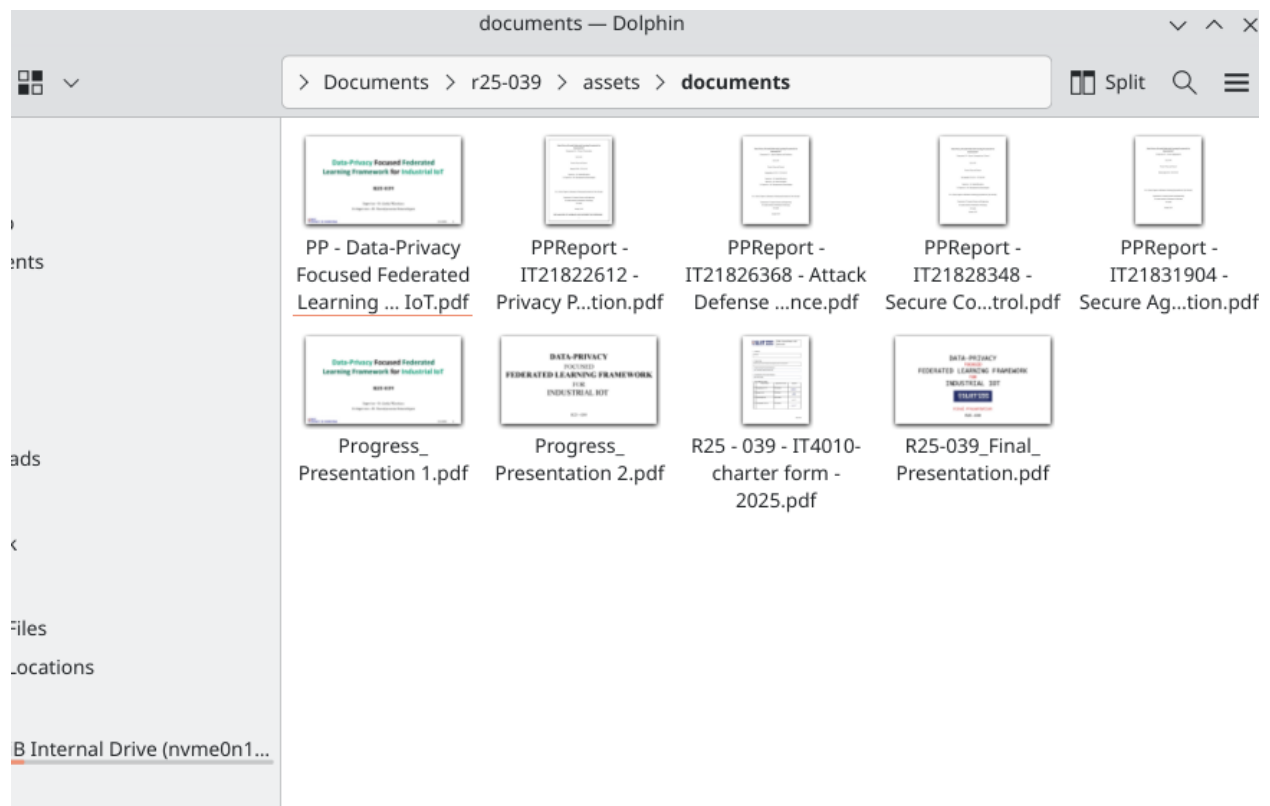
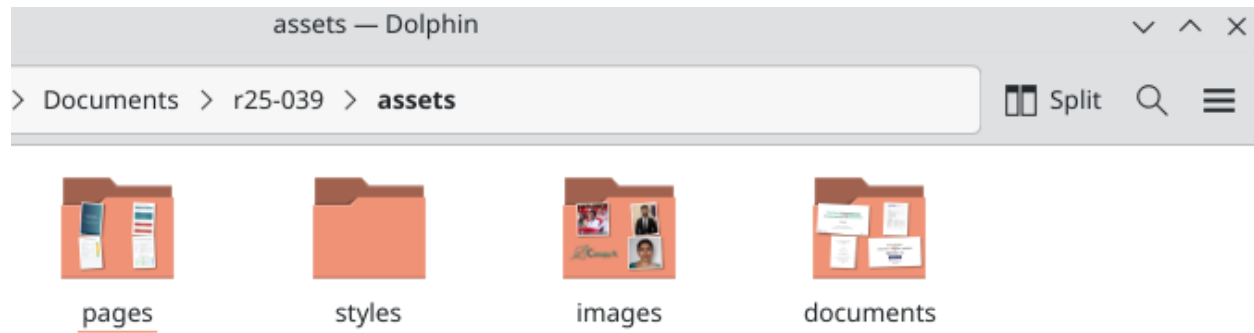
2025RegCloud > **R25-039-Students**

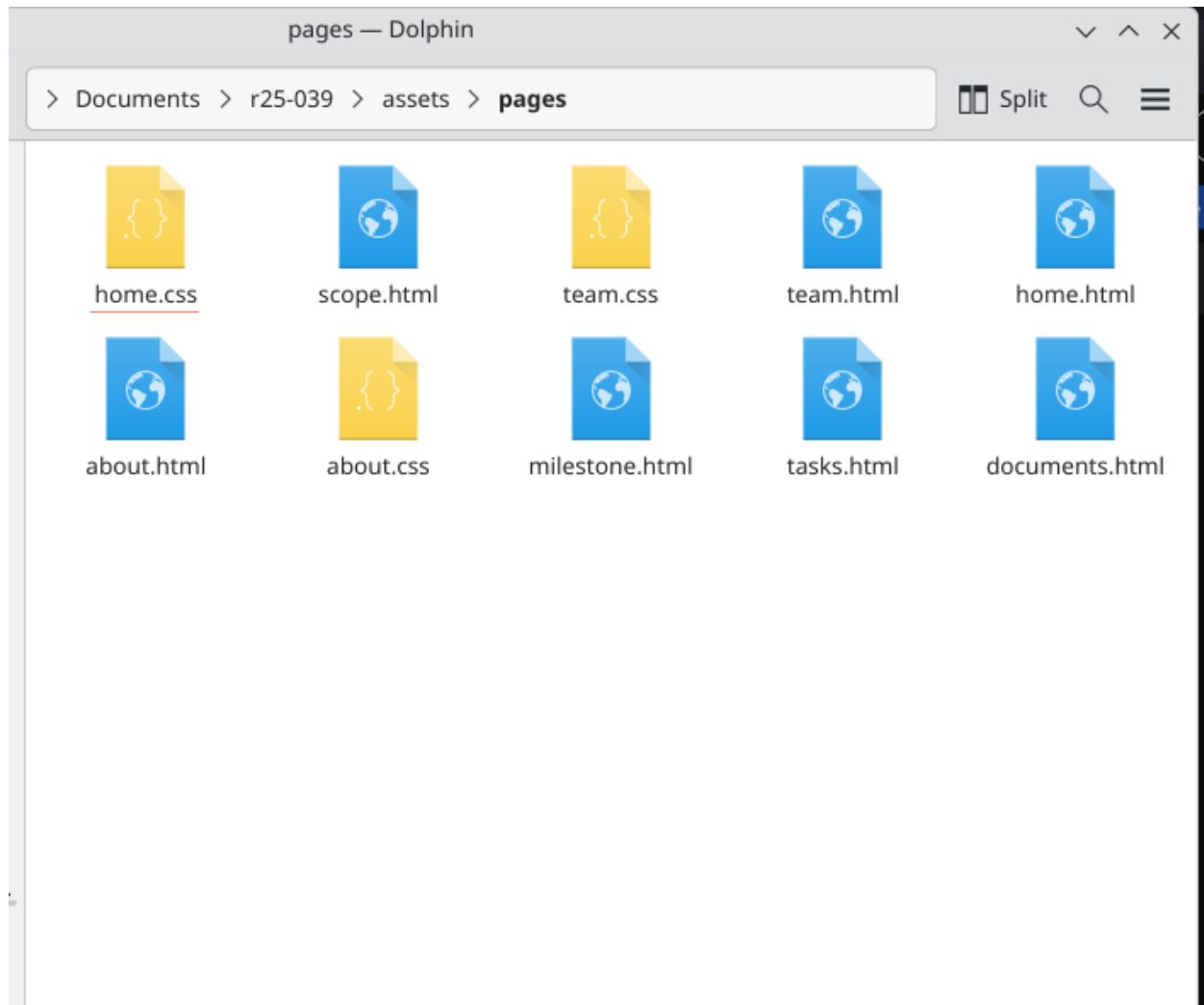
 Name	Modified	Modified By
 1. Project Proposal	January 27	Tharaniyawarma Kumaralingam
 2. Progress Presentation - 1	January 27	Tharaniyawarma Kumaralingam
 3. Progress Presentation - 2	January 27	Tharaniyawarma Kumaralingam
 4. Research Paper	January 27	Tharaniyawarma Kumaralingam
 5. Final Report & Presentation	January 27	Tharaniyawarma Kumaralingam
 6. Check List Documents	April 29	CDAP SLIIT
 7. Website	January 27	Tharaniyawarma Kumaralingam
 8. Log Book	January 27	Tharaniyawarma Kumaralingam
 Marking Schemes	January 27	Tharaniyawarma Kumaralingam
 Project Registration Documents	January 27	Tharaniyawarma Kumaralingam
 Panel Comments for the Students.xlsm	September 20	CDAP SLIIT

## 10. Website


### 10.1 Development







## 10.2 Finalize



Home About Scope Milestones Tasks Docs Team

# A Privacy-First Federated Learning Framework

Building secure, resilient, and efficient ML models without compromising data privacy. Designed for the demands of the Industrial IoT.

**F . L . A . M . E**

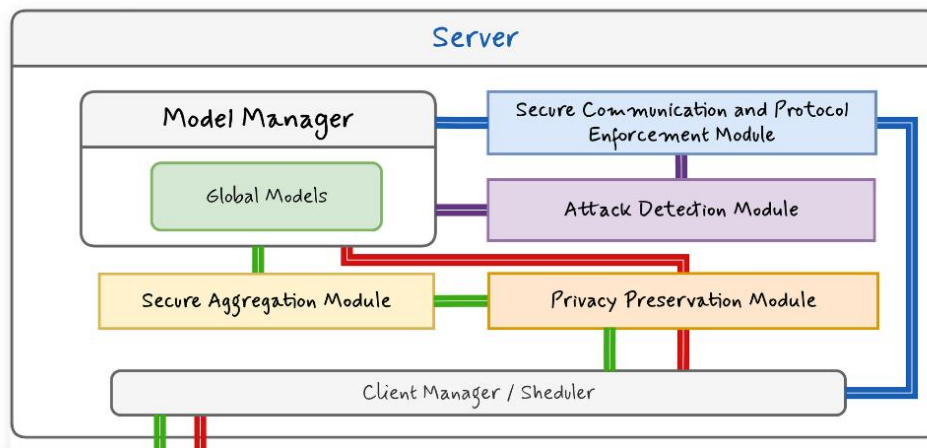
**Federated Learning Aggrigated Modular Environment**

[Explore Features](#) [View on GitHub](#)

# Privacy-Enhanced Federated Learning Framework

Our comprehensive Federated Learning (FL) System Framework is engineered to significantly augment the privacy, security, and operational resilience of machine learning models deployed in decentralized and distributed environments. The framework is composed of four interconnected core modules, collectively guaranteeing data integrity, defense against adversarial attacks, and authenticated inter-component communication.

## Framework Overview and Architecture







**Mr. Amila Senerathne**

Supervisor



**Dr. Sanika Wijesekara**

External Supervisor



**Mr. T. Kumaralingam**

Co-Supervisor

