FederatedAI System Design Description

Abstract

The ‘FederatedAI‘ system provides a framework for collaborative AI model training using distributed resources in a secure and privacy-preserving manner. It leverages heterogeneous computing resources across a federation of systems while ensuring data privacy and efficiency.

Contents

1 Overview

The ‘FederatedAI‘ system facilitates collaborative training of an AI model. ensuring data privacy and secure model updates. It trains the model locally and the weights and gradients are shared/produced but the raw input data is not. In Section 2, we describe the implementation details of the system, including the coding languages, tools, and database used, as well as how the system manages the training and secure communication between systems.

2 Implementation

This implementation is based on the SysD document for the ‘FederatedAI‘ system.

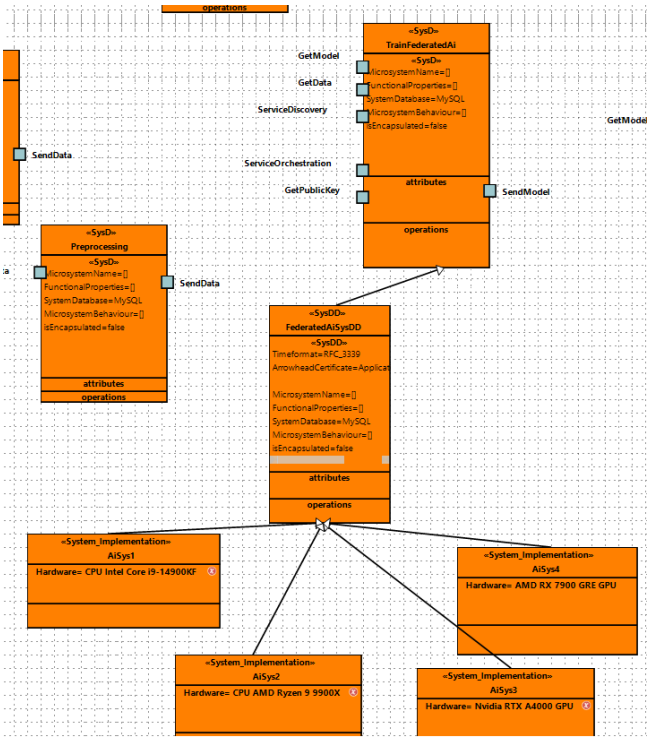

Figure 1: SysD (FederatedAisystem) document and several other systems and system implementations

2.1 Implementation language and tools

- Coding language: Python 3.9
- IDE: PyCharm; Compiler: Python’s standard interpreter
- Libraries:
  - PyTorch - for AI model training and gradient computation
  - Flask - for setting up API endpoints for communication between systems
  - OpenSSL - for securing communication between systems using TLS

2.2 Functional properties implementation

- Data handled and eventually stored by the system includes:
  - Weights and updates (gradients) at train federated ai system.
  - Metadata such as system status, task assignments, and model versions.

2.3 Non functional properties implementation

2.3.1 Security

- The system uses TLS encryption for all communications between systems to protect against eavesdropping and ensure secure data exchange.

- Authentication is managed using public key infrastructure (PKI) where each system has a unique certificate for verification.

2.3.2 Power management

- The system dynamically scales the workload based on available resources and power consumption data from systems, optimizing the allocation of tasks to balance performance and power usage.

2.3.3 Internal monitoring

- A monitoring service tracks system performance, model accuracy, and system health, providing real-time feedback and logs for maintenance and debugging.

2.3.4 Configuration

- The system accepts configuration data such as system addresses, model types, and communication protocols during initialization.

3 Services

The implementation services are based on the following SD and IDD documents:

- SD: FederatedAI System Design
- IDD: FederatedAI IDD

Table 1: References to documentation for services produced and consumed.

Services produced/consumed	SysD ref	SD ref	IDD ref
FederatedModelUpdate	SysD FederatedAI	SD FederatedAI	IDD FederatedAI

4 References

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	2024-10-19	1.0	Initial version of ‘FederatedAI‘ SysDD document	Adam Epstein

5.2 Quality Assurance

No.	Date	Version	Approved by
1	2024-10-19	1.0	