AiGym

System Description

Abstract
This is the template for System of Systems Description (SoSD document) according to the Eclipse Arrowehad documentation structure.

Contents

1 Overview

This document describes the AiGym system of systems (SoS), which provides training of a AI model and testing of that model using training and test data, the AI gym use 4 machines with different processors to achieve the result. Prefereable make use of SysML/UML use case diagram.

The rest of this document is organized as follows. In Section **??**, we reference major prior art capabilitites of the SoS. In Section 1.1, we the intended usage of the SoS. In Section 1.2, we describe fundamental properties provided by the SoS. In Section 1.3, we describe de-limitations of capabilities ofn the SoS. In Section 2, we describe the microsystem (abstract level with references to their SysDs) which constitutes the SoS. In Section 3, we describe the security capabilitites of the SoS.

1.1 How This SoS Is Meant to Be Used

Describe intended usage of the SoS. Usage scenarios shall be described.

The usage is applicable to any machine learning style system where we learn a model from training data and test on a disjunct dataset to see the performance in a general case/setting.

Preferable a SysML/UML block diagram of the System should be provided. See the SysML profile and library (github.com/eclipse-arrowhead/profile-library-sysml) for support on how such block diagram should look like in eclipse papyrus.

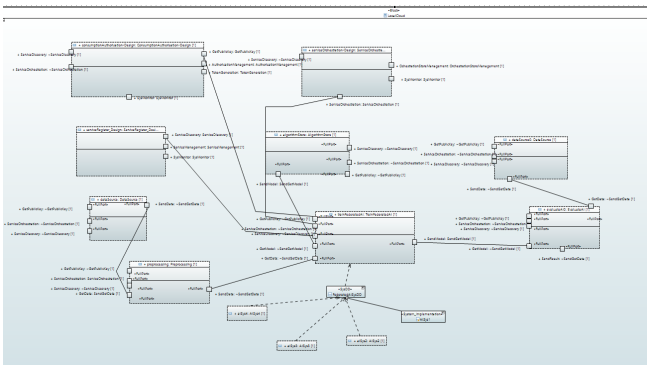

Figure 1: The system of system in its entirety, the Sos in embedded in a local cloud

1.2 SoS functionalities and properties

Narrative describe system functionalities and properties (no implmentation details) like e.g.:

1.2.1 Functional properties of the SoS

Primary Mission Functionality: The main purpose of the SoS. My SoS is designed for training AI models across multiple computing resources, the core functionality is to distribute and manage AI training tasks dynamically.

The SoS facilitates communication among subsystems. It enables real-time data exchange and synchronization between systems.

1.2.2 Configuration of SoS properties

-Mission Objective: Enable dynamic resource allocation for AI training tasks across multiple computing nodes.

-Performance Metrics: Average latency should not exceed 200ms; resource utilization should be above 80% during peak times.

-Data Format: JSON for configuration exchange

-Scalability: Support for dynamic addition/removal of computational resources.

1.2.3 Data stored by the individual microsystem

Brief overview of data stored to achive the functionality of the SoS.

Training Data: There are systems that store the datasets, which could be images/text or preprocessed data. Model Parameters: Storing model weights and configurations such as learning rates and layers as they are updated during training. Training Performance Logs: Storing logs on the progress of model training, including accuracy, loss rates, and time taken per epoch.

1.2.4 Non functional properties

The information is sent using TCP/ IP which is a secure way of transmitting information. This method ensures reliable and orderly delivery, of information packets across the network(s). The method can use the IP of various systems to know where to send the packages. TCP is comparable with all OS, which is a good thing since we have 4 heterogeneous machines which we work with. In our case the information will be transmitted over a local cloud. The SoS should maintain low latency (under 200ms) when allocating AI training tasks across machines.

1.2.5 Stateful or stateless

The microsystems within the AI gym SoS can be either stateful or stateless, depending on their specific role:

Hybrid Configuration: The AI gym combines both approaches, using stateful microsystems for managing training tasks and stateless microsystems for short-lived or real-time operations, optimizing the overall performance of the SoS.

1.3 Important Delimitations

The delimitations are centered around the computational power of the 4 machines that are utilized for the training of the AI model, it will determine the speed of training.

2 Services

This section describes consumed and produced service. In particular, each subsection names a prodiuced or consumed service indicating the different capabilities and associated interfaces of the service. Reference to the appropriate SD document shall be made.

At the endpoints the system consume data for training/test and at the other endpoint the system returns result measures (The service GetSendData is used), internally the service GetSendModel is also used.

2.1 Produced service

SendGetData (GetData) (SD document) and the IDD doc (SendGetDataIDD)

2.2 Consumed services

SendGetData (ResultData) (SD document) and the IDD doc (SendGetDataIDD)

3 Security

The system is connected to the Arrowhead core microsystems which provide service registry, orchestration, and authorisation. The information is sent using TCP/ IP which is a secure way of transmitting information. This method ensures reliable and orderly delivery, of information packets across the network(s). The method can use the IP of various systems to know where to send the packages. TCP is comparable with all OS, which is a good thing since we have 4 heterogeneous machines which we work with. All data exchanged between microsystems is encrypted using Transport Layer Security (TLS) to protect against eavesdropping and man-in-the-middle attacks. Local cloud is used to create a local network to protect against outside threats.

4 References

5 Revision History

5.1 Amendments

Revision history and Quality assurance as per examples below

No.	Date	Version	Subject of Amendments	Author
1	2024-10-19	4.6.2		Adam Epstein
2				
3				

5.2 Quality Assurance

No.	Date	Version	Approved by
1	2022-01-10	4.6.2	