ENI ISG - PoC Proposal

1 PoC Project Details

1.1 PoC Project

PoC Number (assigned by ETSI): PoC#21

PoC Project Name: Validation of LLM for Network OAM Application on Generic Computing Platform

PoC Project Host: China Telecom

Short Description: This PoC intends to demonstrate the feasibility and capability of network OAM LLM application running on generic computing platform aka X86 based platform instead of a GPU platform, with special attention to the lower cost and power comsumption aspects, in the context defined by ENI.

In particular, this PoC solves the adaptation of generic computing platform e.g. X86 to replace part or all of a GPU platform for a reduced power and cost consumption. As demand for GPU increase dramatically due to the rise of AIGC applications, service providers having difficulties to provide sufficient computing power for AIGC related applications. The X86 platform on the other hand, cumulated large amoung of computing power by stable service provider investiment. The overall CPU workload for service provider often very low, less than 50%. To solve the unmatch problem of computing power and demand, it is a possible solution to employ AIGC applications on X86 resouces. This PoC will demonstrate the computing ability by X86 platform, which realizes the balance between the CPU workload and lowering the power and budget cost.

Note that we begin with to demonstrate the use case [Use Case #1-3: Energy optimization using AI] and [Use Case#4-2: Assurance of Service Requirements] discussed in GS ENI 001 [1]. In general, we aim to achieve these objectives based on a X86 platform with open source toolbox. Thus, this ENI implementation can also be applicable to other user cases.

1.2 PoC Team Members

Table 1.1

	Organization name	ISG ENI participant (yes/no)	Contact (Email)	PoC Point of Contact (see note 1)	Role (see note 2)	PoC Components
1	China Telecom	Yes	Yu Zeng (zengyu@chinatelecom.cn)	X	Service Provider	- User Stories / Use Cases definition - PoC development - PoC documentation - PoC demos
2	Intel	Yes	Haining Wang (haining.wang@intel.com)		Vendor	-Help with the architecture design, implementation of algorithm, testbed setup
3	China Unicom	Yes	Bingming Huang (huangbm7@chinaunicom.cn)		Service Provider	-Participation in project discussions
4	Huawei UK	Yes	Aldo Artigiani (aldo.artigiani@huawei.com)		Vendor	-Participation in project discussions
5						

NOTE 1: Identify the PoC Point of Contact with an X.
NOTE 2: The Role will be network operator/service provider, infrastructure provider, application provider or other as given in the Definitions of ETSI Classes of membership.

All the PoC Team members listed above declare that the information in this proposal is conformant to their plans at this date and commit to inform ETSI timely in case of changes in the PoC Team, scope or timeline.

1.3 PoC Project Scope

1.3.1 PoC Goals

The PoC will demonstrate aspects of various Use Cases that were identified by in GS ENI 001, namely:
Use Case #1-3: Energy optimization using AI
Use Case #4-2: Assurance of Service Requirements

This PoC intends to demonstrate a method of employing AIGC related applications on X86 platform. The detailed goals include:

 PoC Project Goal #1: AIGC application on X86 Platform. Demonstrate how to support AIGC application on X86 platform, support AIGC infrerence for LLM services, and realize the functionality and capability to provide identical AIGC request.

 PoC Project Goal #2: Distributed AIGC computing optimization. Demonstrate organizing multiple computing node to provide a distributed solution for AIGC application.

1.3.2 PoC Topics

PoC Topics identified in this clause need to be taken for the PoC Topic List identified by ISG ENI and publicly available, i.e. the three topics identified in clause 4.5 of the ENI PoC Framework. PoC Teams addressing these topics commit to submit the expected contributions in a timely manner.

Table A.2

PoC Topic Description (see note)	Related WI	Expected Contribution	Target Date
Network Operations -> Intelligent Network application	GS ENI 001 Use Cases (release 4)	1.Functional blocks for this PoC. 2.Test results for LLM OAM applications	30/10/2024
NOTE: This column should be filled according to the contents of table 1.			

1.4 PoC Project Stages/Milestones

Table A.4

PoC Milestone	Stages/Milestone description	Target Date	Additional Info
P.S	PoC Project Start	03/2024	Proposal approved for PoC reviewing during #ENI 29
P.D1	PoC Demo 1	08/2024	Venue, F2F / Webinar
P.D1	PoC Demo 1	09/2024	Venue, F2F / Webinar
…	…		
P.C1	PoC Expected Contribution 1	10/2024	contributions to ENI requirements ENI-002v411.
P.C2	PoC Expected Contribution 2	10/2024	contributions to ENI use case ENI001v411.
…	…		
P.R	PoC Report	12/2024	PoC-Project-End Feedback
P.E	PoC Project End	01/2025	Presented to ISG ENI for information
NOTE: Milestones need to be entered in chronological order.			

1.5 Additional Details

2 PoC Technical Details

2.1 PoC Overview

An x86-based LLMapplication is a software program designed to facilitate language learning and processing tasks on computers with x86 architecture. The x86 architecture refers to a family of instruction set architectures (ISAs) based on the Intel 8086 CPU. It includes both 32-bit (x86) and 64-bit (x86-64 or x64) processors, which are widely used in personal computers, servers, and embedded systems.

An LLM application on this architecture leverages the capabilities of x86 processors to perform complex natural language processing (NLP) tasks, including but not limited to language translation, sentiment analysis, text summarization, and language generation. These applications utilize advanced machine learning models, particularly those in the realm of deep learning, to understand, interpret, and generate human language in a way that is meaningful and contextually relevant.

Key Components of an x86-based LLM Application:

1. **Language Models**: At the core of an LLM application are the language models themselves. These models are trained on vast amounts of text data to learn the statistical properties of languages. They can predict the likelihood of a sequence of words or generate new text based on a given prompt.

2. **Processing Unit**: Given the intensive computational demands of LLMs, especially those involving deep learning, the x86 CPUs, and optionally GPUs (Graphics Processing Units), are utilized to process the data efficiently. Modern x86 processors are capable of parallel processing, significantly speeding up the computations required for training and inference.

3. **Software Frameworks**: The application relies on software frameworks and libraries optimized for machine learning, such as TensorFlow, PyTorch, or ONNX. These frameworks provide the tools and functions necessary for model development, training, and deployment, taking full advantage of the x86 architecture for performance optimization.

4. **Data Preprocessing and Management**: Effective language learning requires clean and structured data. The application includes components for data preprocessing, such as tokenization, stemming, and lemmatization, as well as data management tools for handling large datasets.

5. **User Interface (UI)**: An intuitive UI is crucial for facilitating interaction between the user and the LLM application. This could range from simple command-line interfaces to sophisticated graphical user interfaces (GUIs), depending on the application's complexity and target audience.

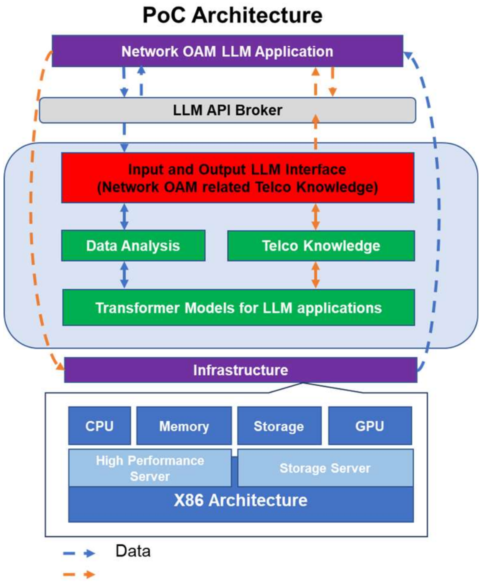
6. **Security and Privacy**: Given the sensitive nature of language data, x86-based LLM applications incorporate security features to protect user data and ensure privacy. This includes data encryption, secure data storage, and compliance with data protection regulations.

In summary, an x86-based LLM application represents a powerful tool for a wide range of language-related tasks, leveraging the computational capabilities of x86 processors to deliver advanced NLP functionalities. As technology progresses, these applications are expected to become even more efficient, accurate, and versatile, further enhancing our ability to interact with and process human languages.

This PoC intends to demonstrate the feasibility and capability of network OAM LLM application running on generic computing platform aka X86 based platform instead of a GPU platform, with special attention to the lower cost and power comsumption aspects, in the context defined by ENI.

2.2 PoC Architecture

2.2.1 The diagram represented below shows the framework of the PoC mapping to the ENI reference architecture.



In order to validate the performace of LLM applications, the PoC framework consists of the flowing stages:

1. The first stage is data processing of the LLM, the infrastructure layer contains two types of server structure: the high performance and storage. The high performace server will perform the inference LLM tasks, and the storage server perform less computing exhausting tasks e.g. backup, data exchange.

2. The second stage is model pre-training, which maps to Cognition Framework, Knowledge Management and Context-Aware Management FBs of the ENI System. In this stage, training datasets are fed into LLM algorithms (e.g. transformer based) to generate the models used to provide Telco OAM applications.

3. The third stage is model tuning, the X86 servers can be scaled to meet the demands of the tuning process. This can involve adding more servers to form a cluster or enhancing existing servers with more powerful CPUs. This scalability ensures that the infrastructure can keep up with the increasing computational demands of model tuning, particularly as models grow in size and complexity.

4. The fourth stage is model inferencing. In this stage, the X86 architecture can be used in inferencing tasks, the task of request handling from upper system layers can be performed using X86 server and software.

Based on the above stages, validation tasks can be carried out by comparing the performace between generic computing platform and GPU platform.

2.3 PoC Success Criteria

Explain how the proposal intends to verify that the goals are presented in clause A.3.1 have been met.

EXAMPLE: Functional (demonstration shown validation of generic computing platform of the PoC proposal worked), Performance (comparing to GPU platform, the generic computing platform can meet the network service requirements), Availability(can be improved by scaling optimization).

2.4 Additional information

[1] RGS/ENI-008 (GS ENI 001), "Experiential Networked Intelligence (ENI); ENI use cases", v3.1.15, Sec 5.3.
[2] RGS/ENI-007 (GS ENI 002), "Experiential Networked Intelligence (ENI); ENI requirements", v3.2.0.