AI in PLM Support Processes

Traditional support structures that leave access to information entirely at the user's discretion will not be sufficient for dynamic engineering environments under time pressure.

AI Transformation

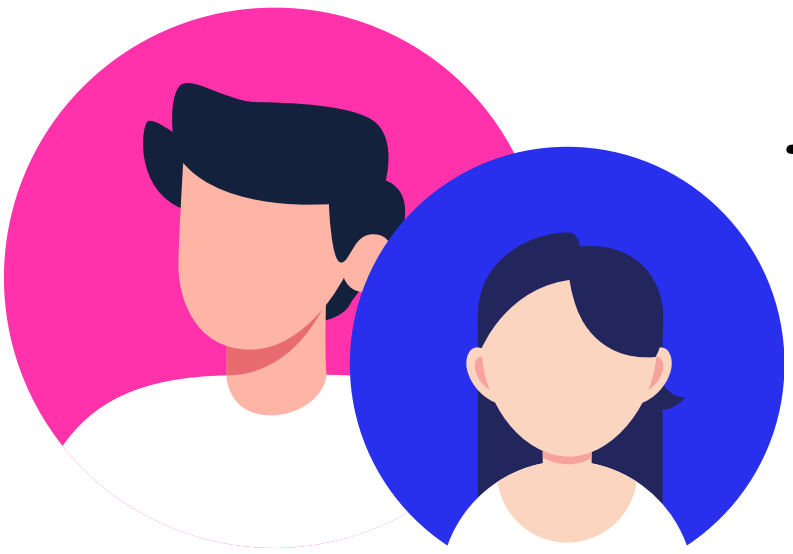
Therefore, could an AI-powered context-aware bir
——**digital assistant**
approach that is activated the moment an error occurs be adopted?

Solution Approach: –**AI-powered** guidance at the moment of error

System Features

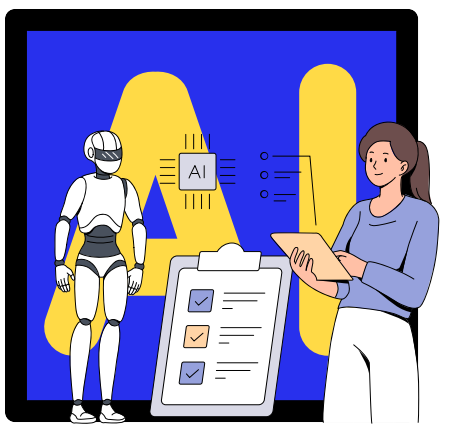
Operates as a **browser extension**.
Automatically detects error messages in the **3DEXPERIENCE Platform** interface by listening to the DOM.
Sends the error message and related object information to an AI LLM model specifically trained for **3DEXPERIENCE.**
If necessary, initiates the process via a Jira ticket or email. The user no longer needs to search for a solution.
The solution appears on the screen precisely when the need arises.

Users



ERROR

Auto Detection



Solution

Jira-email

Simple Scenario : ‘modify frozen data’ error

When the user attempts to assign an Enterprise Item Number to a FROZEN data object in the 3DEXPERIENCE platform, they receive the error message:
“EDITION IS NOT ALLOWED…”

How the System Works?

Detection
The Chrome/Edge extension automatically detects on-screen error messages by listening to the **DOM** (Document Object Model) .

Context Gathering
Information such as the object type where the error occurred, lifecycle state, and user role is collected.
Relevant chunks from the **3DEXPERIENCE Users Guide** database are added to the context via **RAG**.

Analysis & Inference & Response Generation
The fine-tuned **LLM** analyzes the cause using the error message and context.

Solution Suggestion
Based on the fact that Enterprise Item Number assignment is not allowed on **frozen** data, root cause analysis and solution suggestions are presented to the user.

Additional Support Automation
If the user wishes:
The error screen and object information are automatically attached.
A Jira ticket is created.
An email is sent to the relevant support team.

Solution Components
RAG + LLM + Supervised Fine-tuning

🧠 LLM (Large Language Model)
Understands the error messages in the 3DEXPERIENCE interface and generates explanatory and guiding responses.
Since it is trained with 3DEXPERIENCE error messages, it can provide platform-specific interpretations.

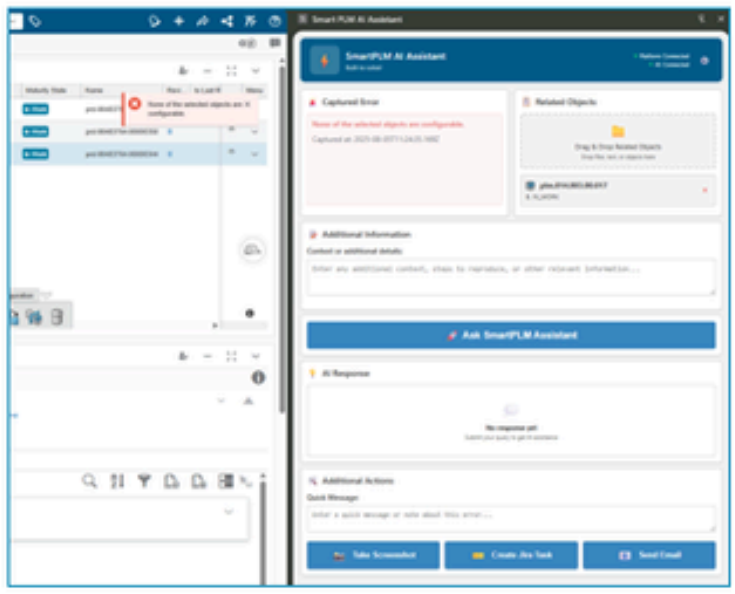
🔎 RAG (Retrieval-Augmented Generation)
Retrieves in real time the parts of the 3DEXPERIENCE Users Guide document that may be related to the error from the FAISS database, strengthening the accuracy and context of the model's response.
Provides a way to easily add new context without the need to retrain the model.

📚 3DEXPERIENCE Platform Domain Specific Train Data
The model was trained with a dataset of 7,500 examples containing real 3DEXPERIENCE errors and their solutions.
As a result, the responses are not generic but specific to the system's content and targeted.

🤖 Integration Layer
When the solution suggestion is insufficient, the system can transfer the findings and the current screenshot to a Jira ticket or a support email.
It can be easily integrated with other help desk systems.

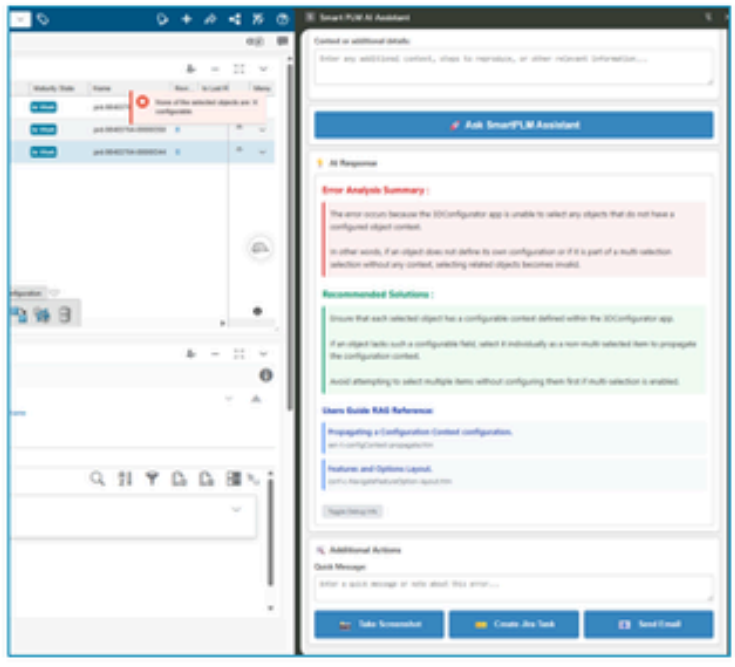
Solution Components
INFRASTRUCTURE

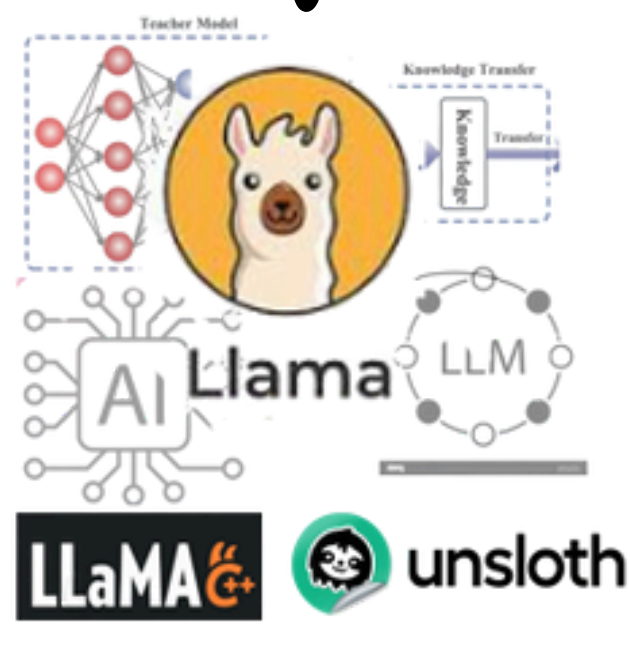
The open-source LLaMA3 1B Instruct model was trained with **unsloth**. Thanks to its small size, it can run even on a **local CPU**.
The 1-billion-parameter model enables usage **without** the need for high-end hardware, in **on-premise** environments, **on cloud** infrastructures, and even directly on **client computers.**
The architecture is flexible enough to integrate with both **closed networks and cloud systems.**



Retrieval Augmented Generation

Smart PLM Assistant
Browser Extension





Technical Details

The training set was prepared by combining error messages from the 3DEXPERIENCE Platform's on-premise language(nls) files with the 3DEXPERIENCE Users Guide.

The unsloth library was used for the fine-tuning process.
Training was completed in approximately 1–2 hours on Google Colab using an L4 GPU.
Base Model: LLaMA 1B Instruct
Dataset: A 7,500-record dataset containing 3DEXPERIENCE-specific error messages extracted from NLS files and synthetically augmented was used.
Format: Prepared in the Alpaca structure (instruction, input, output).

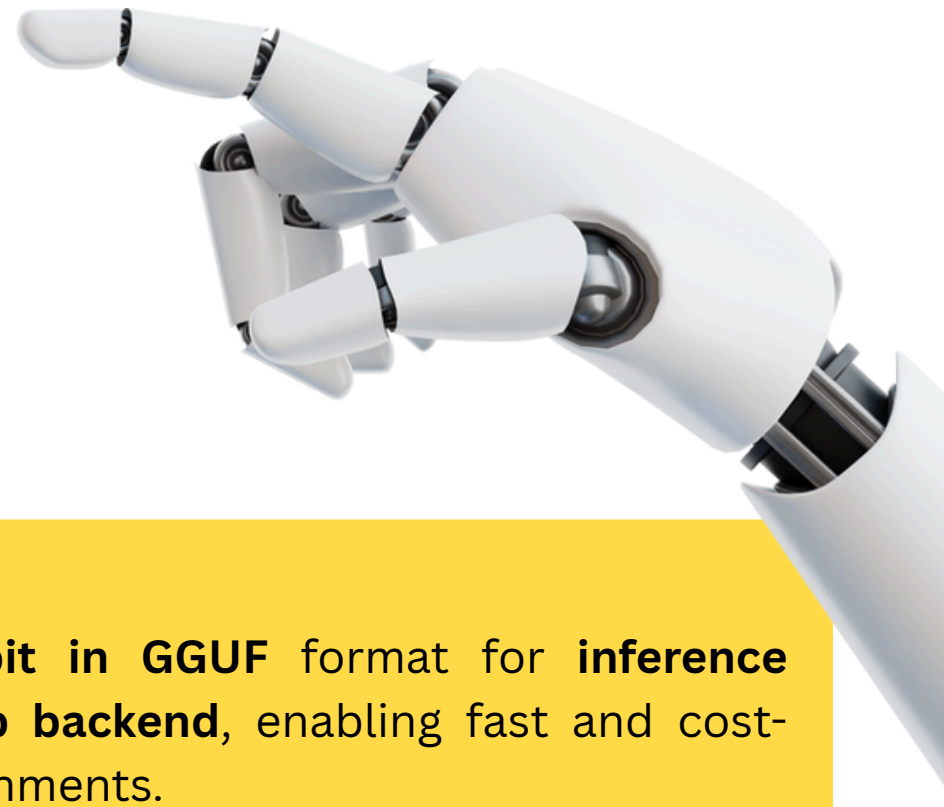
SFT (SUPERVISED FINE-TUNING)

llama 3.1-1B-Instruct

Supervised Finetuning Training
3DEXPERIENCE
Errors & Solutions

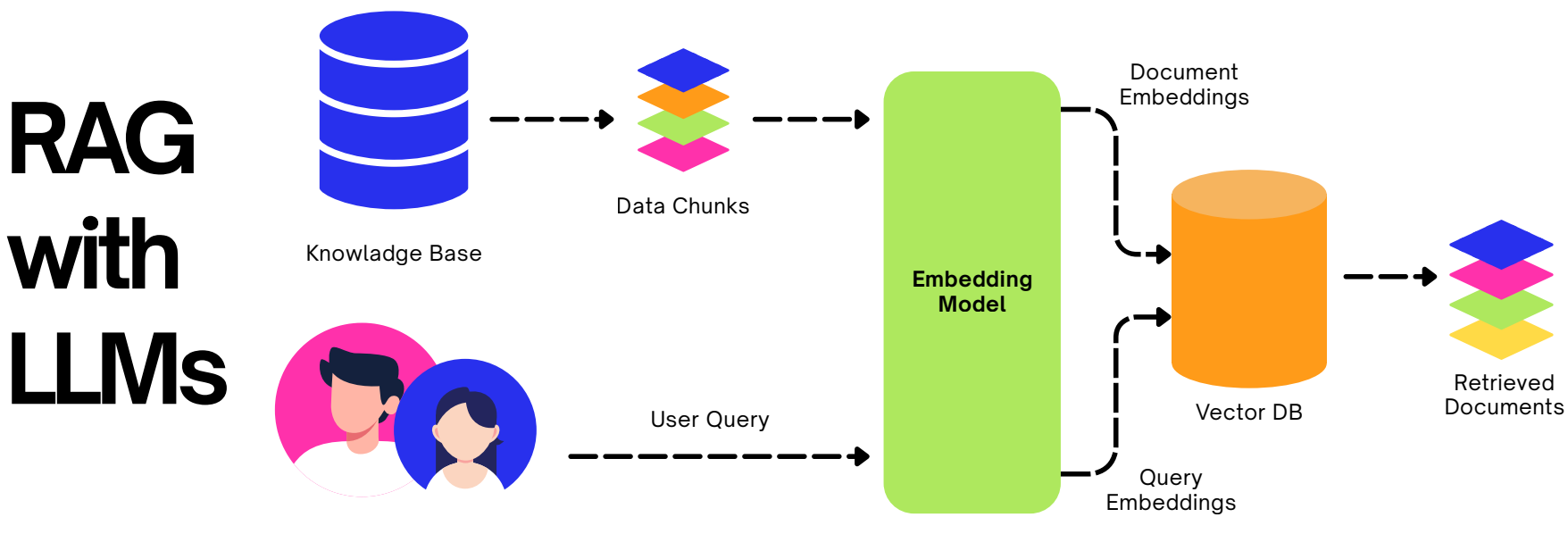
1b-Fine Tuned
3DEXPERIENCE Model-
4bit- UUGF

Technical Details



Local Inference Optimization
The fine-tuned model was **quantized to 4-bit in GGUF** format for **inference optimization** and deployed with the **llama.cpp backend**, enabling fast and cost-efficient execution on both **CPU and GPU** environments.
The system is designed to operate entirely in **offline environments**.
Training can be performed in the cloud, while inference can run on **local machines or within isolated networks.**

RAG (Retrieval-Augmented Generation)
Allows the **AI model** to retrieve the most **up-to-date** and accurate information from **external knowledge sources** (documents, databases, etc.) to generate responses, thereby improving both accuracy and **contextual consistency.**



Future Perspective Vision

Approach & Conclusions

Instead of developing new models, training small models(**slm)** with the **high quality data** is much more efficient.

Low-parameter models like **LLaMA 1B** can deliver high performance with **fine-tuning.**

By training even lower-parameter models, it is possible to have systems that can run on a **CPU**.

The most critical factor: creating a diverse and high-quality **training dataset.**

Expansion Possibilities

Not only solving problems, but also automating processes through **AI agent** and **tool** usage.

With **continuous training** adaptation, the model has been further trained using responses to support requests.

Aiming to improve the model's **accuracy** by **scoring** responses with an additional text classification model.

Technical Aspect

Training **reasoning**-supported multi-step models to ensure complete solutions.

Agentic AI – Tool usage, **LangChain** implementation.

This structure is the first step toward context-aware, AI-powered guidance in PLM systems.

for screen video please click