Blockathon Presentation Script

github:

> https://github.com/soonding/Powerblock

dockerhub:

docker push tnsghd22/powerblock:latest

Slide 1: Title Slide

Good afternoon everyone, and thank you for joining this session.

Today, I'm honored to present our project titled *"Web3-based Privacy-Enhanced AI Chatbot Framework with Clio-X Integration,"* developed under the name **POWERBLOCK**.

This presentation will cover our motivations, system architecture, technical details, and key innovations that differentiate our approach.

Slide 2: Motivation and Objectives

The rapid adoption of AI chatbots has brought clear benefits for user interaction and automation.

However, this convenience often comes at the cost of privacy: users' personal data can be collected, stored, and even shared without adequate safeguards or consent mechanisms.

合适的，proper

At the same time, Web3 technologies are offering novel ways to empower individuals with decentralized, self-sovereign identity management and data ownership.

有主权的

Our objective was clear: to design a chatbot framework that integrates privacy enhancement mechanisms while leveraging **Clio-X**, a Web3-native solution, ensuring users can securely authenticate and control their personal information.

The core goal is not just an AI chatbot—but a privacy-first, user-centric conversational platform.

Slide 3: System Architecture Overview

This slide illustrates the overall architecture of our framework.

On the left, we see the **User Input Interface**, where users submit their queries.

These inputs are authenticated through **Clio-X**, which handles Web3-based identity verification and consent management.

Next, inputs pass through a **Local Privacy Enforcement Layer**, implemented within a Dockerized algorithm container.

Personally Identifiable Information

This layer performs automatic **PII masking** using natural language processing techniques before the sanitized query is sent for processing.

FAISS is a tool library specifically designed for efficient and fast similarity search in large-scale vector data.

The core processing layer integrates **vector similarity search using FAISS**, a high-performance nearest-neighbor search library, to retrieve relevant contextual information from curated datasets, such as Enron and Cameroon government records.

Finally, the system generates a response using an **LLM backend**, served via **Ollama**, running externally but securely interfaced.

The response is saved locally as a JSON object, ensuring full auditability and reproducibility.

Slide 4: Technical Details and Implementation

Let's look deeper into key technical elements.

We implemented the algorithm container using **Python 3.11 with a slim Docker base image**, optimizing for fast builds and reproducibility.

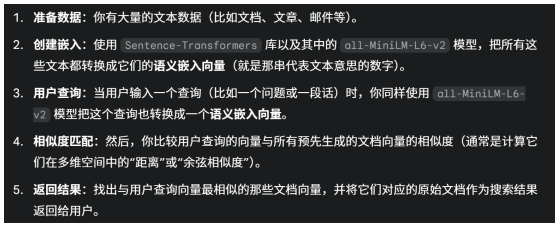
spaCy 是一个 NLP 库

Named Entity Recognition

The PII masking module relies on **spaCy's pre-trained NER model,** `en_core_web_sm`, which automatically detects sensitive entities such as names, locations, and email addresses.

生成句子、段落甚至短文档的优质嵌入向量

For semantic search, we integrated `sentence-transformers` **with the** `all-MiniLM-L6-v2` **model**, encoding user queries into dense embeddings for fast similarity search using **FAISS** indexes.



为了让整个系统**更灵活**（方便管理、独立升级、多应用共享）并且**更高效地利用昂贵的 GPU 资源**（因为 LLM 运算高度依赖 GPU），我们选择将运行大型语言模型（LLM）的 Ollama 服务单独部署在一台独立的电脑或服务器上，而不是和主要的应用程序混合在一起运行。这就像把一个耗费大量计算资源的核心大脑（LLM）放在一个专门的、高性能的机房里，其他部分需要时就去调用它。

Contextual documents, indexed beforehand, are dynamically queried to provide the chatbot with relevant grounding context before LLM inference.

这句话就是在解释，我们如何利用 Docker 的环境变量机制，让运行在 Docker 容器里的应用程序 `OLLAMA_URL`，能够方便、灵活且不费力地找到并使用外部（宿主机或远程）的 Ollama 服务。

The **Ollama server** runs externally for flexibility and GPU utilization.

Within Docker, we explicitly configured environment variables such as `OLLAMA_URL` to enable seamless connectivity to the host machine or remote Ollama service.

Slide 5: Privacy Enhancement Mechanism

A key innovation in our system is the **real-time PII masking before any user query reaches the language model backend**.

This approach ensures that even if the LLM backend or external API were compromised, it would not receive unmasked sensitive data.

Our solution goes beyond traditional privacy-by-design principles by enforcing automatic masking without requiring manual user intervention.

Moreover, Clio-X ensures that authentication metadata itself remains under user control and can be cryptographically verified.

1. **输入文本：** 示例输入文本为“Summarize Cameroon decrees mentioning Mr. John Doe from Paris and email john.doe@example.com”。其中包含了人名、地点和电子邮件等敏感信息。
2. **spaCy NER处理：** 系统首先利用spaCy的NER功能对输入文本进行处理。根据规则，如果识别出的实体标签是PERSON或GPE，则将其替换为“[MASKED]”。在这个例子中，“Mr. John Doe”和“Paris”将被识别并遮蔽。
3. **Regex电子邮件增强：** 接下来，系统应用正则表达式对电子邮件进行进一步的增强遮蔽。预定义的 `email_pattern` 会匹配符合电子邮件格式的字符串，并将其替换为“[MASKED]”。这样，“john.doe@example.com”也会被遮蔽。
4. **处理后的输出：** 最终，经过这些步骤处理后，输出的文本变为“Summarize Cameroon decrees mentioning [MASKED] from [MASKED] and email [MASKED]”。所有敏感信息都已被成功遮蔽。

1. The example input text contains sensitive information such as names, locations, and email addresses.
2. The system first processes the input text using spaCy's NER functionality. According to the rules, if the identified entity label is PERSON or GPE, it is replaced with "[MASKED]".
3. Next, the system applies regular expressions for further enhanced masking of emails to replace them with "[MASKED]". 4. Finally, all sensitive information has been successfully masked.

Slide 6: Web3 and Clio-X Integration

In this section, we emphasize the integration of **Clio-X as the core Web3 component**.

Clio-X allows users to authenticate via decentralized credentials and grants them granular control over what information is shared during the session.

1. a user who possesses a Decentralized Identifier
2. Once the user initiates an action, the system immediately performs a blockchain verification
3. Upon successful blockchain verification, clio-x event triggers are activated.
4. Following the Clio-X event trigger, the relevant Algorithm container is activated.

The consent and authentication flow leverages blockchain primitives for transparency and auditability, ensuring that the system adheres to the highest privacy standards.

This integration provides trust, decentralization, and eliminates reliance on centralized identity providers.

Slide 7: Results and Output Example

Here we present an example output from our system.

Given the input query: *“Summarize Cameroon decrees mentioning Mr. John Doe from Paris and email john.doe@example.com,”*

the **sanitized query automatically masked PII entities**, resulting in:

“Summarize Cameroon decrees mentioning Mr. [MASKED] from [MASKED] and email [MASKED].”

The system correctly retrieved relevant Cameroon government decrees and returned a privacy-protected summary, ensuring no sensitive information was leaked or processed by the language model backend.

Slide 8: Deployment and Containerization

这页好像不存在

To maximize portability, we containerized the entire algorithm using **Docker**, with a clear separation of dependencies, input/output volumes, and environment configurations.

Developers can easily run `docker-compose` locally or in cloud-native environments.

Further, we prepared the image for **publishing on Docker Hub for easy deployment and sharing**, ensuring reproducibility and rapid onboarding for new environments or stakeholders.

Slide 9: Conclusion and Future Work

In summary, POWERBLOCK demonstrates that it is feasible to design an **AI chatbot that rigorously enforces privacy while offering Web3-native user control and transparency**.

By integrating **Clio-X decentralized identity management** with PII masking and contextual grounding using FAISS, we provide both strong privacy guarantees and high-quality AI-driven responses.

For future work, we envision extending this framework to support **end-to-end encryption of responses**, deeper integration with blockchain-based data marketplaces, and expanding multilingual capabilities to serve a broader user base.

Slide 10: Thank You

Thank you all for your attention.

We would be happy to answer any questions and discuss how this framework can be adapted for different use cases or integrated into enterprise-grade solutions.