January 2024

Tenant Data Security for LLM Applications in Multi-Tenancy Environment

Assaf Namer

Jim Miller

Prashant Kulkarni

Hauke Vagts

Jason Bisson

See next page for additional authors

Inventor(s)
Assaf Namer, Jim Miller, Prashant Kulkarni, Hauke Vagts, Jason Bisson, and Brandon Maltzman

Tenant Data Security for LLM Applications in Multi-Tenancy Environment

ABSTRACT

Large language models (LLMs) and other types of generative artificial intelligence can be used in a wide variety of business applications. However, there is a possibility of data leakage from LLM responses when an LLM is used in shared multi-tenant environments where each tenant has respective private datasets. Deploying individual adapter layers for each tenant can provide data isolation. However, such implementations can be complex and costly. This disclosure describes techniques to create and maintain a single model that can serve multiple tenants, with security controls for multi-tenancy services to isolate customer data efficiently. Data for different tenants is signed with their respective tenant-specific keys and is then appended with the tenant-specific signature prior to training/tuning a model or use by the model at inference time. When a business application of a particular tenant requests a response from the LLM, the response is generated using the adapter layer. The response includes data citations that are verified prior to the response being provided to the business application. The verification is based on the tenant-specific signature in the citation to ensure that only data that belongs to the particular tenant that requested the response is included.

KEYWORDS

- Large language model (LLM)
- LLM security
- Data leakage
- Data isolation
- Response citation
- Data citation
- Grounding
- Adapter model
- Fine-tuned model
- Parameter efficiency turning (PET)
- CI/CD

BACKGROUND

Large language models (LLMs) and other types of generative artificial intelligence (Gen AI) are used in a wide variety of business applications, such as text content generation, classification of data, chatbots, etc., that can improve business efficiency. Since LLMs are trained on large text datasets that may belong to multiple entities, the use of an LLM to generate an answer can cause risk of data leakage or sensitive information disclosure, e.g., if the LLM reproduces a portion of the training data in the response or if the LLM response includes data for which the querying entity does not have access rights.

In scenarios where multiple users access a shared environment, the likelihood of data exposure can be higher. For example, consider a SaaS (Software-as-a-Service) provider that offers an LLM-based service to customers such as a chatbot. To isolate data between different tenants (i.e., customers), the SaaS provider would need to create individually tuned models, e.g., based on the respective customer's data. While such model fine tuning and separation can ensure the highest isolation of data, the implementation can be complex. Also, it can be costly to manage and update each of the models.

In current multi-tenancy environments, data isolation for customers can be achieved through various techniques, e.g., encryption domains and per-tenant keys. However, when dealing with machine learning datasets, the separation of keys may not be a feasible solution, as these datasets exist in cleartext to be usable by the LLM or other models. Employing data masking and obfuscation techniques for securing data may not provide sufficient isolation of data between tenants. For example, if encrypted data values may be visible to multiple customers, it may be possible to identify patterns even if the data itself is not readable.

DESCRIPTION

This disclosure describes techniques to create and maintain a single model (e.g., an LLM or another generative AI model type) that can serve multiple customers, with security controls for multi-tenancy services to isolate customer data efficiently. The techniques are based on the principle of data citation. As per the principle, data citations are included in the response generated from LLMs as evidence of source data.

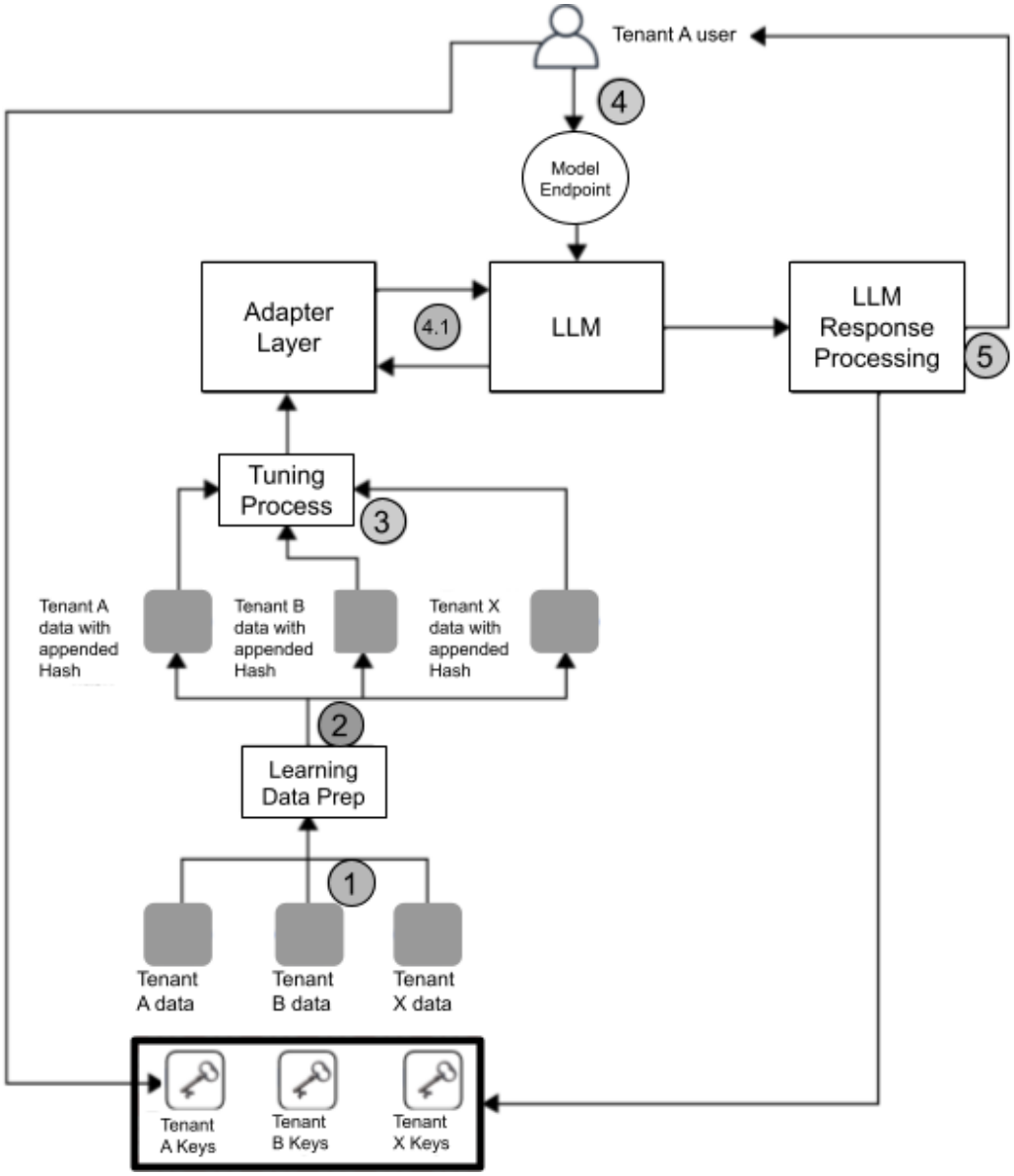

Fig. 1: Data isolation for LLM use in a multi-tenancy environment

Fig. 1 illustrates data isolation for LLM use in multi-tenancy environments with logical data separation and security control, per techniques of this disclosure. As shown in Fig. 1, in the multi-tenancy environment, every tenant (customer) has an individual set of data and keys (1). The tenant data and keys are isolated from those of other tenants with appropriate security controls. A unique, private hash is generated for each tenant using the tenant's key and tenant-ID. This hash serves as a unique signature for the tenant. For example, a hash, 'hash-a-,' can be created for tenant ID 'A'. This process can be performed using any suitable cryptographic function. This process is also known as hash-based message authentication code (HMAC).

The data artifacts of each tenant are prefixed with the tenant's signature (2) as depicted in Fig. 1 during data preparation. For example, if tenant A's hash is "hash-a-," and data includes a file named 'financial_report.pdf,' then the file is renamed to the concatenation of the hash and the original file name, that is '*hash-a*-financial_report.pdf'. Similarly, the files for all tenants are renamed. The hash can be appended to the file as a prefix, suffix, or at another suitable point in the filename. A data artifact can be structured data, such as a database table, or unstructured data, such as a document or PDF. With appropriate user permissions, a combination of different files for a tenant can be used to build training datasets required for model training.

A large language model (LLM, also referred to as foundation model) is tuned (3) on the dataset comprising data from the different tenants. The adapter layer, in combination with a standard LLM, can receive (4) queries from different users or applications (tenants) via a model endpoint and generates responses (4.1). The tuned weights can be stored in an adapter layer which is used during LLM inference to influence the LLM response.

When the user (e.g., a particular application of a particular tenant) provides a query to the model via the model endpoint, the unique and private hash of the tenant is provided as part of the query. The generated LLM response is processed (LLM Response Processing, 5) using the hash that was provided by the tenant application. The processing includes verifying citations in the LLM response as belonging to the tenant's dataset. The verification is done by verifying the appended signature on each artifact in the citation in the LLM response. After verification (and the removal of any information for which the citation cannot be verified), the response is provided to the tenant application. This process ensures that tenant data is not cross-contaminated, e.g., revealed to other tenants.

As private unique keys are used to separate different tenants' datasets and LLM responses, multiple tenants can utilize a single model without suffering data leakage. Further, tenants can control artifact selection for specific prompts or applications via different keys or different hashes while using a single adapter layer. The techniques described can be applied to any model architecture where response source citation is available in the responses generated by the model.

By using the described techniques, with appropriate permissions from tenants, data from different tenants can be leveraged to train and/or tune generative AI models while maintaining security and confidentiality of tenants' data. Since a single model can serve multiple customers in a multi-tenancy environment, the overall cost of using generative models is lower than techniques that require the creation of individual adapter layers for each tenant.

The additional security and isolation of data provided by the described techniques can prevent data leakage. Tenants can also use multiple different keys to segregate data within their own environment. Model providers, e.g., cloud service providers that offer LLMs or other

generative AI models, can provide the models to their customers while meeting compliance requirements.

CONCLUSION

This disclosure describes techniques to create and maintain a single model that can serve multiple tenants, with security controls for multi-tenancy services to isolate customer data efficiently. Data for different tenants is signed with their respective tenant-specific keys and is then appended with the tenant-specific signature prior to training/tuning a model or use by the model at inference time. When a business application of a particular tenant requests a response from the LLM, the response is generated using the adapter layer. The response includes data citations that are verified prior to the response being provided to the business application. The verification is based on the tenant-specific signature in the citation to ensure that only data that belongs to the particular tenant that requested the response is included.

REFERENCES

1. "Responsible AI - Citation metadata," available online at https://cloud.google.com/vertex-ai/docs/generative-ai/learn/responsible-ai#citation_metadata accessed Dec 12, 2023.
2. "OWASP Top 10 for Large Language Model Applications," available online at https://owasp.org/www-project-top-10-for-large-language-model-applications/ accessed Dec 12, 2023
3. "Transform your applications into multi-tenant architecture," available online at https://cloud.google.com/saas#section-5 accessed Dec 12, 2023.

4. “Adaptation of Large Foundation Models” available online at https://services.google.com/fh/files/misc/adaptation_of_foundation_models_whitepaper_google_cloud.pdf accessed Dec 12, 2023.
5. “Grounding Large Language Models — Arion Research LLC” available online at https://www.arionresearch.com/blog/grounding-large-language-models accessed Jan 11, 2024.