GOVTEXT

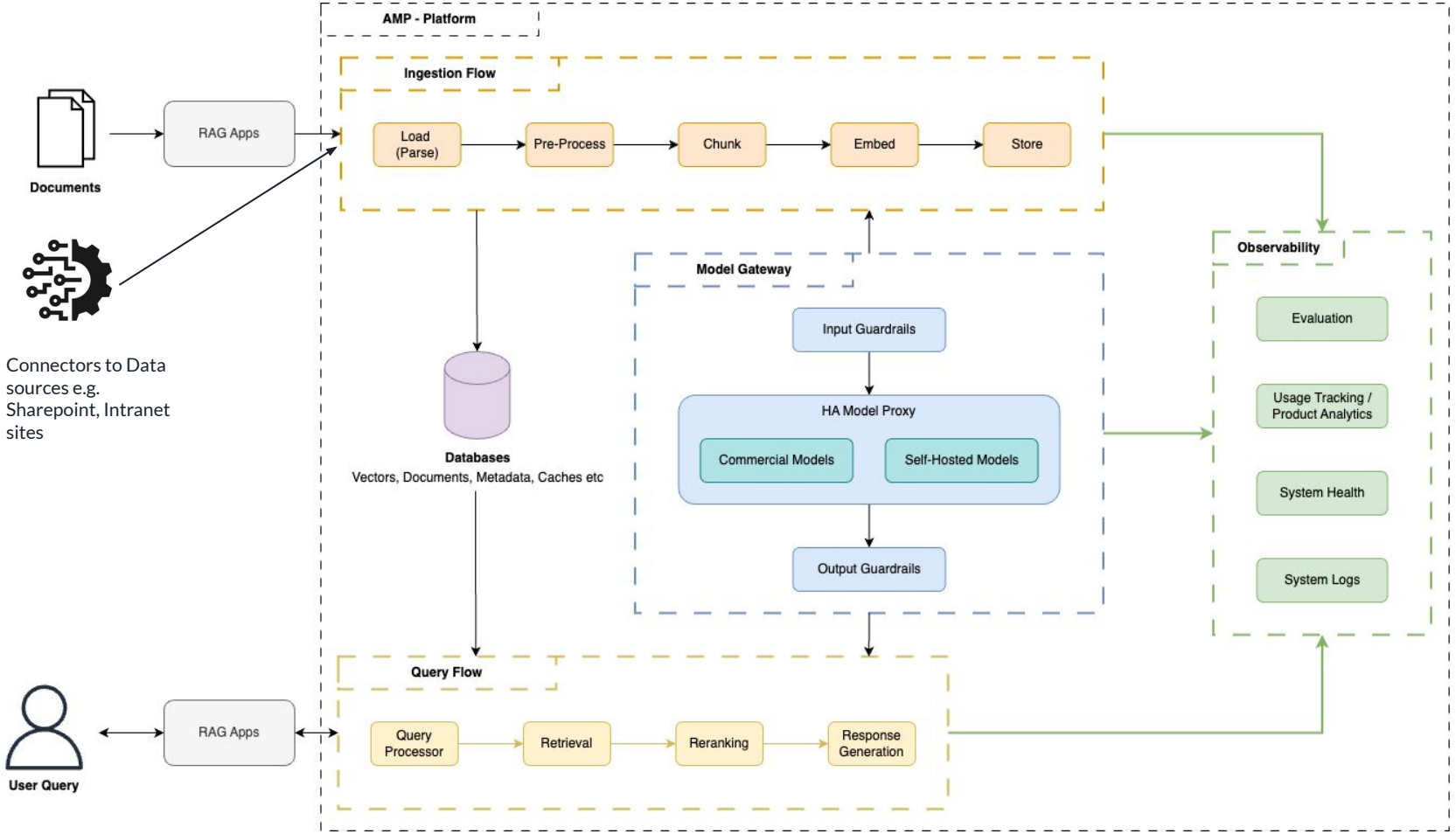
LLM-as-a-Service

Introducing GovText's Platform Services

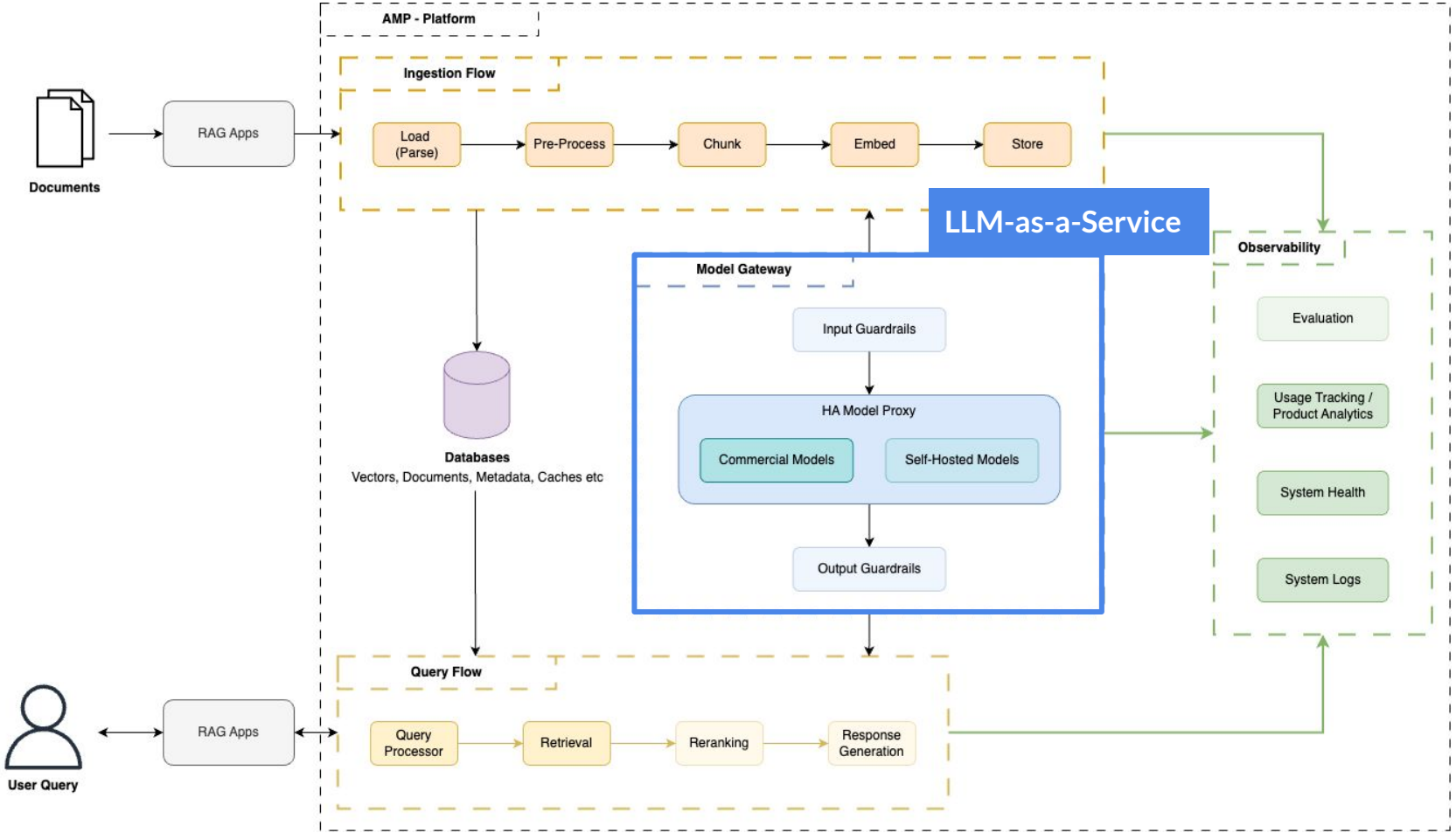
- Retrieval-Augmented Generation-as-a-Service (RAGaaS)
  - Scalable and configurable ingestion pipeline, query pipeline and storage through a unified platform.
  - Embedding and generation models incorporated in pipelines.
  - Other RAG pipeline services e.g. Evaluation, guardrails

- Large Language Model-as-a-Service (LLMaaS)
  - Proxy creation and management of API keys for both cloud service providers (CSPs) and self-hosted models.
  - Dynamic load balancing to maintain high token generation speed, and overcome CSP rate limitations, such as hidden queues or allocated tokens per minute.
  - Easy access to multiple LLMs and embedding models without the complexities of setting up subscriptions, and configurations to handle higher-class data in line with SNG requirements.
  - Tools to track and monitor usage and billing efficiently.

RAGaaS - Medium-term to-be Architecture



RAGaaS - Current as-is Architecture



Origin of GovText's LLM-as-a-Service (LLMaaS)

GovText is a product in GovTech's Data and AI Platforms program and GovText's mission is to accelerate AI adoption across WOG by developing text-related Reusable AI Services.

LLMaaS is currently one of the two foundational AI Services in GovText.

LLMaaS is part of the next evolution of LLM Stack.

- LLM Stack was first developed in May of 2023 to facilitate innovation with Gen AI
  - Meant to support Low Code No Code development of Gen AI applications
  - Not built for stability and scale

- LLM Stack joined GovText in Aug 2024, at a time when realizing the benefits of Gen AI in production is becoming more important relative to just innovation
  - Performance, Stability, Scalability + Security, Safety and Compliance becomes key for getting the benefits of Gen AI in production

Why GovText LLM-as-a-service?

Identified Problems

1. Duplicative Efforts

It is ineffective, inefficient and costly, to duplicate efforts across WOG to set up and access common LLM services and self-hosted models in a safe, secure, performant, and policy-compliant manner.

-> Doing so will result in lowered ROIs, and delays in time-to-value and adoption of AI-enabled applications across WOG.

2. Potential Learning Gaps

As a DAIP platform for Reusable AI Services, we need to be intimately in tune with the use of LLMs across government, so that we can continually develop 'on-point' Reusable AI Services in line with the demands across WOG.

Solution

GovText LLM-as-a-service (LLMaaS)

Build once, Serve all.

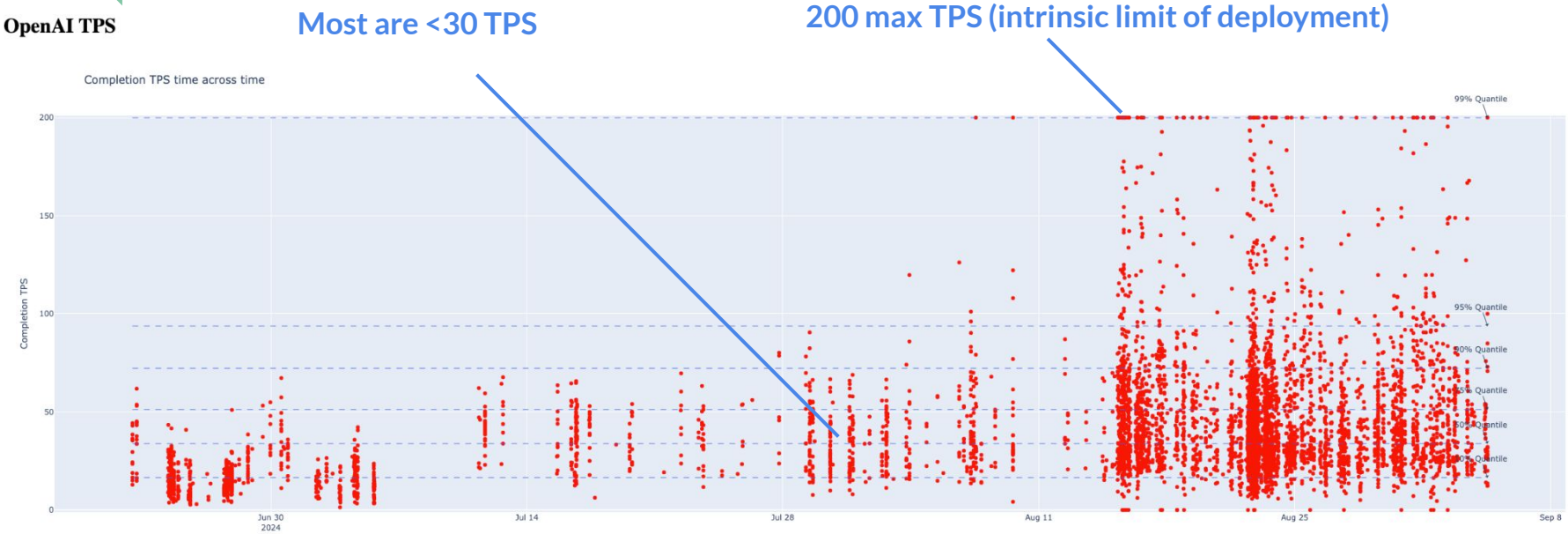
- Fastest tokens per second generation achievable globally
- Central non data retention config + CSPs agreement
- Input and Output guardrails with DAIP sister team - AI Guardian
- DAIP + AI Practice curated self-hosted models for different use cases + handle CCE

Serve all, Learn more, Build right.

- Central AI use-cases consolidation, LLM-usage observability across WOG
- Identify AI patterns to invest efforts

Unstable performance of single-region deployments is a problem

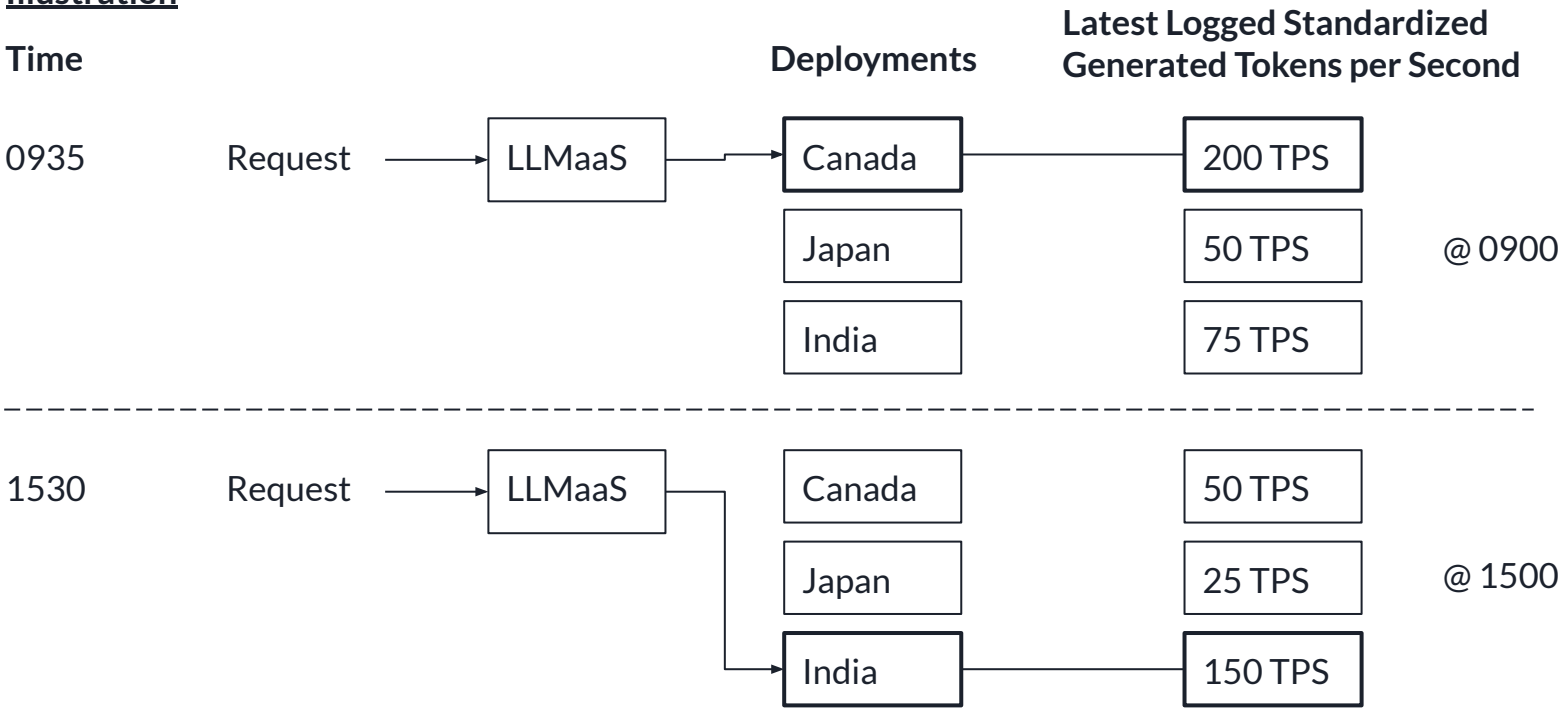
OpenAI TPS



From CareerKaki, Canada east deployment, GPT3.5T

We solve this by continuously determining the fastest deployment globally to send requests

Illustration



We eliminate the uncertainty of policy-compliance and lengthy process needed to achieve it

To use managed service LLMs for Restricted SN data, e.g. Azure Open AI, an agency has to…

Requirements	Agencies' awareness	Estimated time to achieve
1. Get agency's legal department to secure written agreement from Microsoft to agency, indicating that no data will be retained for any reasons if compliance logging is turned off	Low	Weeks - Infinity [Legal takes long + Azure may not respond quickly]
2. Use a specific form to apply for modified abuse-monitoring and content-filtering of Azure Open AI services for a specific Azure subscription	Low	Days - Weeks [Used to be not approved unless we help to push Azure, now may be better]
3. Ensure that Azure Open AI deployments are ONLY spun up in Non-GDPR regions	Extremely Low	Agency don't know, so not done

But if we took out the out-of-the-box guardrails to handle higher classed data… how will we ensure safe and secure usage?

=

Work with DAIP sister team ‘AI Guardian’ (evolution of AI Verify) to install our own guardrails.

[Works well too when we avail through APIs, self-hosted models that, intrinsically, have no out-of-the-box guardrails]

So many open-sourced models, how to choose which to self-host and avail?

=

Collaborative curation across DAIP and AI Practice

DAIP value-generation - e.g. GovText, Transcribe
[Experience using models with real use cases e.g. Phi-3, LLaMA3.1]
+
DAIP value-protection - e.g. AI Guardian
[Evaluation and benchmarking of models e.g. safety, biasness, vulnerabilities]
+
AI Practice
[Experimentation of latest models]

Users of LLMaaS

General Usage

Closed-beta keys issued to 7 teams

- CPFB
- MCCY
- Sentosa
- SSG
- Locus
- SHIP-HATS
- Transcribe

>10 more requested to access, pending issuance

- EDB, DOS, MDDI, MINDEF, MFA, URA, Judiciary, CareerKaki, …

Used as a component within AI Engines of key central products

In Progress

- AI Bots
- Query@Gov
- VICA

Future

- Enterprise Search

Experience with LiteLLM thus far, and future explorations

Challenges & Fixes:

- LiteLLM has numerous issues (Full list here in Annex)
  - Rate limiting issues by model on same key
  - Allowing agencies to view own usage & spend
  - Various UI and key management issues
- Other Issues fixed (worked directly with LiteLLM team)
  - API key bug - unable to call specific embedding model
  - Spend & rate limit exceeding settings

Concurrent Explorations:

- GovText is strategically working closely with Azure to test out APIM
- Evaluating in parallel other proxy gateway solutions and “AI Gateways” e.g. APIM, Kong AI, Cloudflare
- Load balancing across subscriptions, agency self-management of API keys and viewing of usage and spend amounts

Upcoming Product Roadmap

LLM as a Service (Closed Beta)

- Handling load balancing across subscriptions and regions
- Limits on API (Budget, rate)
- OpenAI Models (LLMs & embedding)

Integrate with RAG Service, handle CCE

- Integrate to enable Generation for RAG-as-a-Service
- Self-hosted models (Llama 3.1, Mistral, Phi3)
- Evaluation of other gateway solutions / migrate if necessary
- Balancing across subscriptions
- Onboarding CSP models (to align)
- Fine-tuned models for use cases (e.g. GPT4o-mini / strawberry)

Future Developments

- Account management to view usage and billing
- Dev Portal to create account, request for keys
- Input / Output guard rails
- Reserved instances for higher TPS

Thank You

Annex

Link to experience of LiteLLM thus far

LiteLLM General Issues

UI Issues:

- UI unable to assign keys to diff users (can be done via curl)
- Unable to log out within UI
- On the usage page, changing time range will reset key selection despite UI indicating key is still selected
- On Model Analytics tab in the Models page, in order to view new metrics properly, once has to refresh the page to re-select model. The refresh button to update the tables does not work - you will be greeted with a bunch of errors (see ss in next slide).

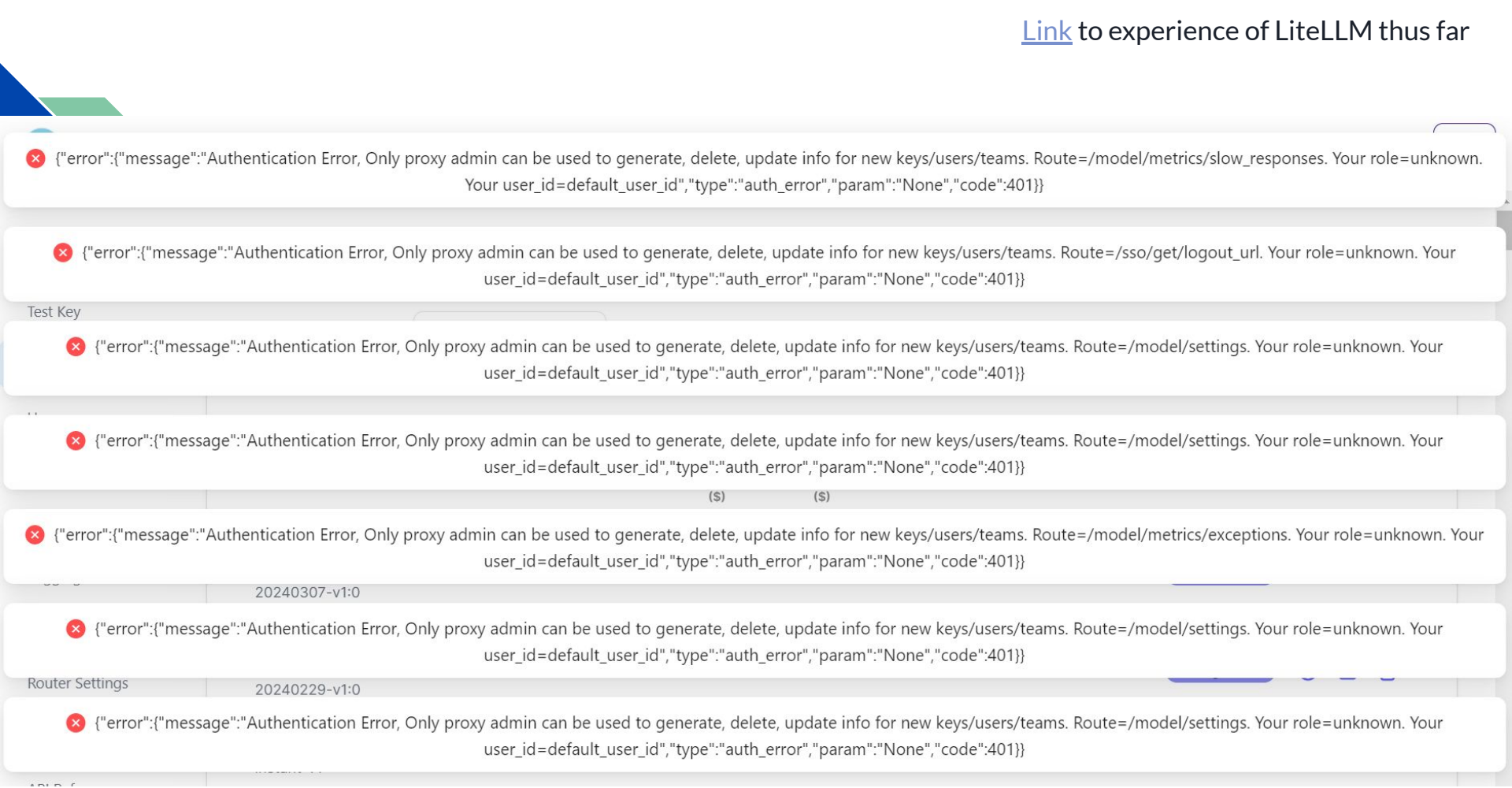
User Management issues:

- There seems to be some minor difference between the role set at user creation vs update role afterwards (i.e. if original user role was admin, they can view UI and get all spend info via endpoints. If the role is updated to internal user they can no longer view the UI but can still get all spend info (requires admin perms) via endpoints).
- Unable to delete users

Other issues:

- Inconsistent behaviour of API key hashing for Prometheus metrics (some ‘hashed’ api keys still show up in plain text), potentially due to different versions of liteLLM being used
- Unable to send user invite link for them to set their own password

Link to experience of LiteLLM thus far



Link to experience of LiteLLM thus far

LiteLLM Internal User & Admin Issues

Internal user issues:

- internal users cannot view spend
- internal users cannot properly delete self created key, but can no longer view key after refreshing page

Admin user issues:

- admin (view) seems to be able to see different UI but still no keys/spend.
- there's also inconsistent admin role behaviour (assigning full admin role b4 reverting to admin view seems to provide some limited viewing permissions, can see total but not breakdown by key)
- removing admin role from an account/api key still seems to allow API calls to access all keys' spending

TIdr: atm does not seem possible for users to view their own spend via UI. The API endpoints show inconsistent behaviour as well.

OpenAI Deployments Fluctuates (Within the Day)

From CareerKaki, Canada east deployment, GPT3.5T

OpenAI Deployments Fluctuates (Across Days)

From CareerKaki, Canada east deployment, GPT3.5T