LLMBot

AI Inference and Document Intelligence Service for GSI/FAIR

Alexey Rybalchenko
Software Development for Experiments Group
Matteo Dessalvi
Cluster Group

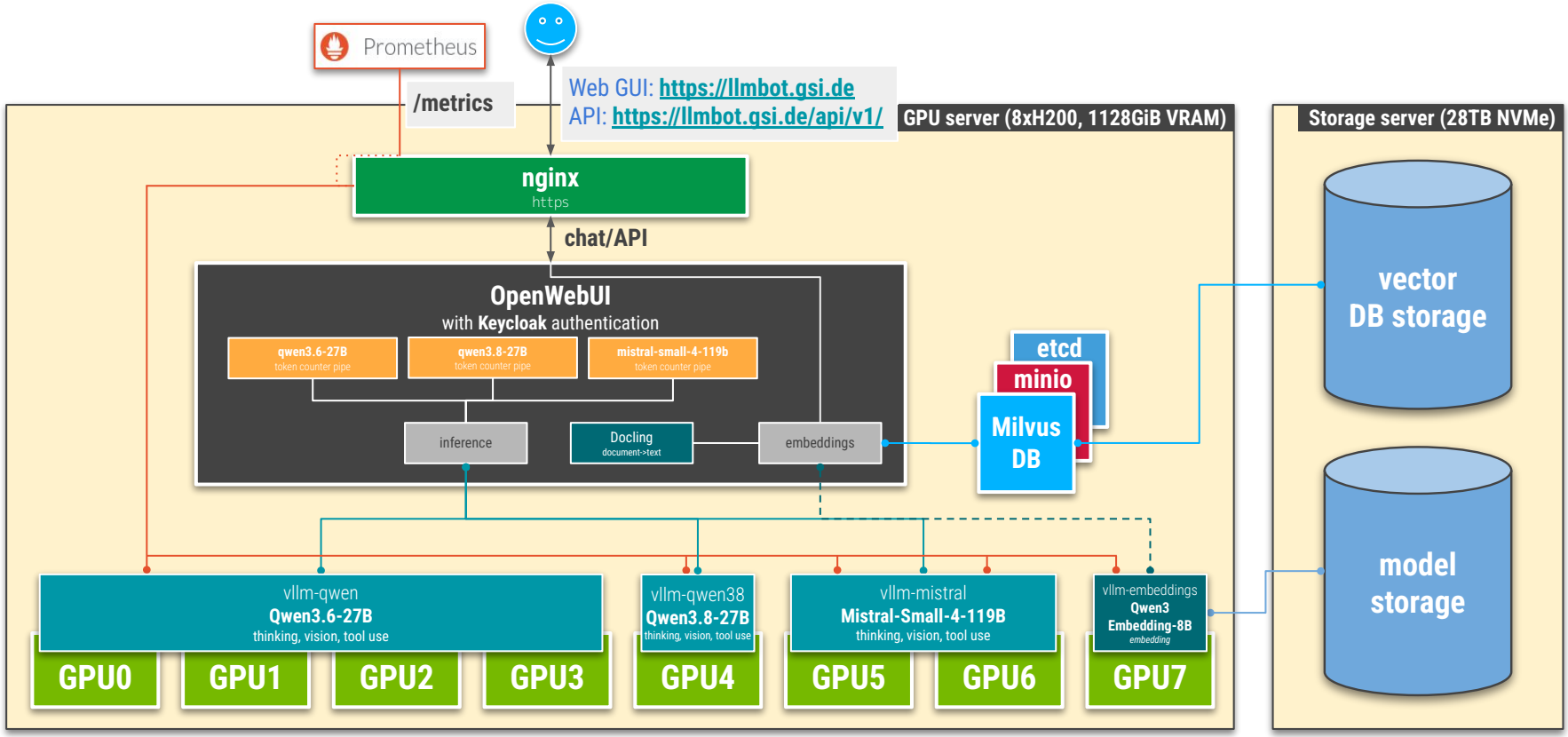
GSI Helmholtz Centre for Heavy Ion Research

2nd GSI/FAIR AI Workshop
GSI, August 31, 2026

Goal

- **On-premises** LLM inference service.
- Full control over data and infrastructure - **chats and documents never leave GSI/FAIR**.
- **Best open-weights models**
  (that fit on the available hardware with sufficient parallelism budget).
- Authentication with **KeyCloak**: id.gsi.de.
- Access via **website** and **API**.
- Local-only (**no web access for the model**) to avoid data leaks.
- Access from **GSI network only**.

System Architecture



Software stack: vLLM, MilvusDB, Docker Compose, OpenWebUI, nginx, Docling

Model Fleet & GPU Allocation

for one 8xH200 server

MODEL	GPUs	PARAMETERS	CONTEXT	QUANTIZATION	FEATURES
Qwen3.6 27B	4 x H200	27B	262k	FP8	thinking + vision + tool use
Qwen3.8 27B **(in testing)**	1 x H200	27B	262k	-	thinking + vision + tool use
Mistral-Small-4 119B	2 x H200	119B-A6B	128k	NVFP4	thinking + vision + tool use
Qwen3 Embedding 8B	1 x H200	8B	-	-	vectorization

Model availability will change over time as newer and more capable models are evaluated and deployed.

Optimised for a cost/performance trade-off.

Second server will be added in the near future, allowing to scale up the service.

Document Intelligence

Tools, RAG

How it works:

1. User uploads document in chat or in a knowledge base.
2. Document is split into chunks that are vectorized & stored in Milvus vector DB.
3. User query triggers retrieval of *n* relevant chunks. Directly via the RAG pipeline or from LLM's tool call.
4. (optional) tools may be used to refine the answer.
5. LLM answers with the context.

Features:

- Dedicated embedding GPU - no contention with chat models.
- Shared knowledge bases - curated collections accessible for all users.
- Private workspaces - per-user/group knowledge bases for sensitive documents.
- Vector deduplication - same file doesn't duplicate storage.

API access allows users to build custom solutions independently and offload heavy inference and embedding work to LLMBot.

→ Optimize vector retrieval settings for specific use case rather than a general one-fits-all.

Integration Pilot Projects

Users can leverage system-managed knowledge bases, add their own, define custom models settings or access the API to build their own solutions. Current pilot projects:

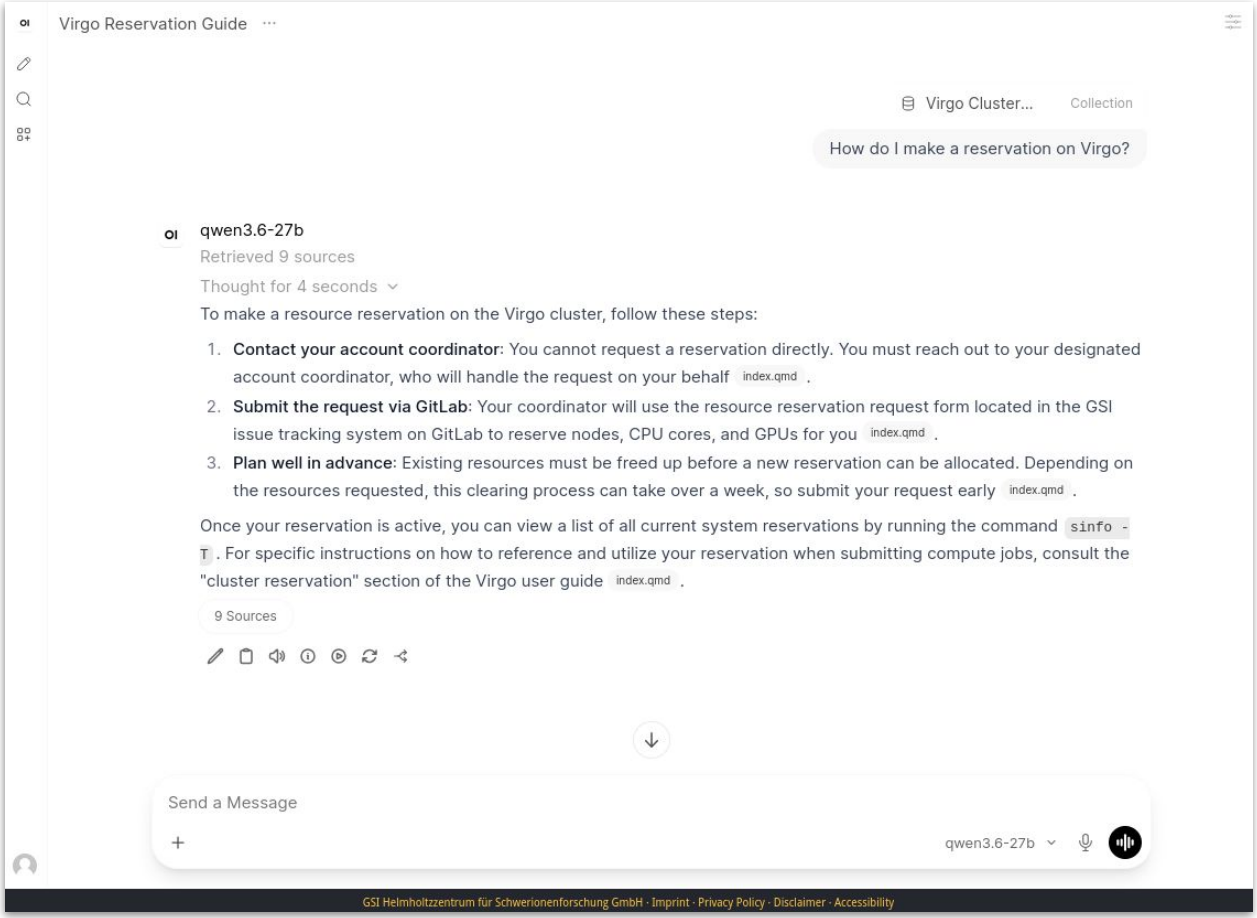
1. **GSI Virgo computing cluster documentation.** knowledge base centrally managed
   Centrally-managed RAG pipeline.

2. **GSI travel assistant.** knowledge base user-managed custom model settings
   Custom model configuration with user-defined knowledge base and tools attached to it.

3. **ELOG-RAG - AI-Assisted Knowledge Search for Electronic Logbooks.** custom solution via **API** for inference and embeddings
   A fully on-premise hybrid RAG system that makes electronic logbook entries searchable in natural language, surfacing expert knowledge that keyword search misses.

"ELOG-RAG - AI-Assisted Knowledge Search for Electronic Logbooks"
Alexander Warth, GSI Magnet Testing & Technology group
2:00 PM

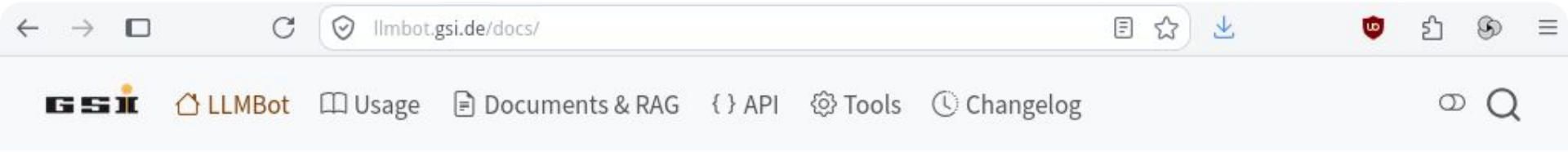
Example Web Session



Documentation

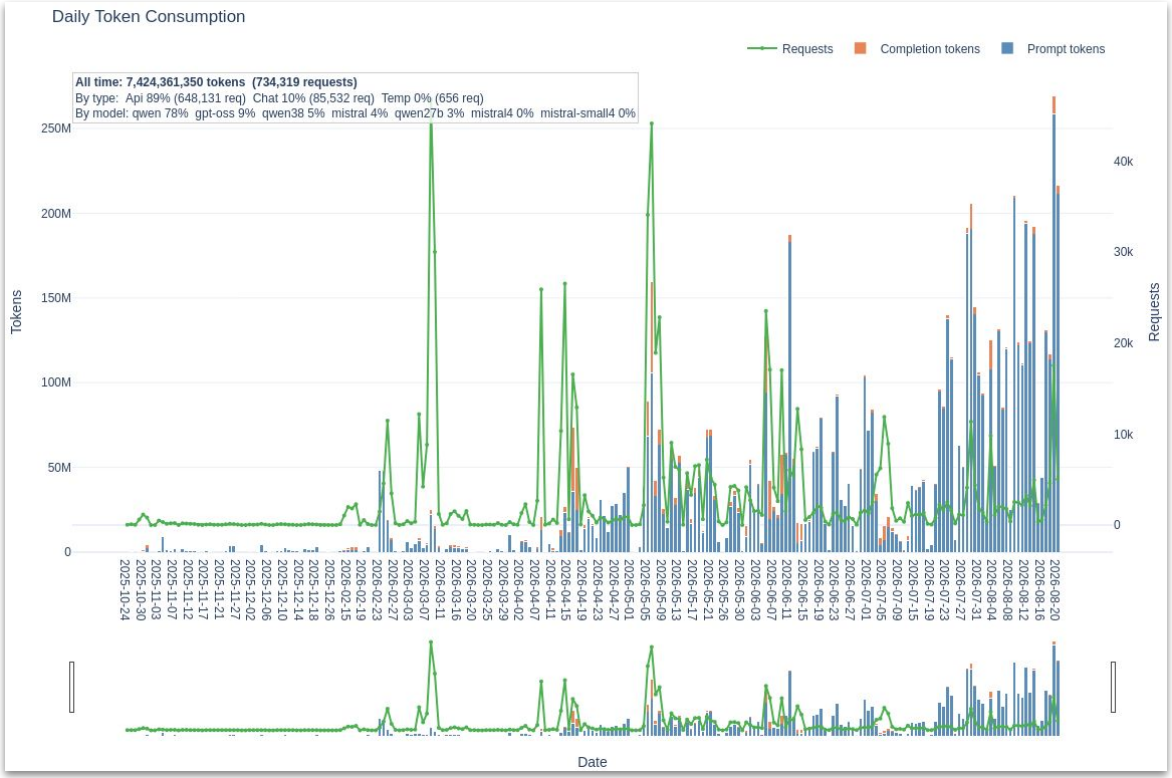
https://llmbot.gsi.de/docs/ covers:

- Current model list and capabilities.
- Usage guide.
- Token efficiency guide.
- Documents guide.
- API documentation.
- Config examples for common tools (OpenCode).
- Changelog.



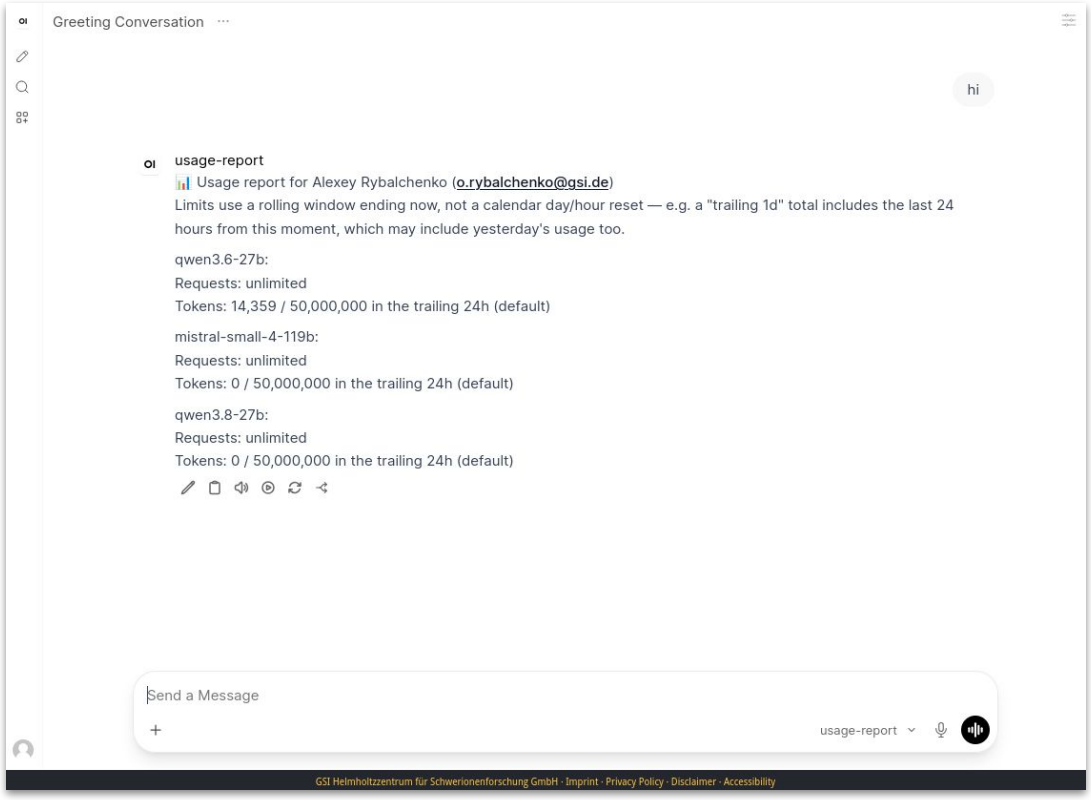
Token Usage Monitoring

Custom pipe function proxies the requests and records **token usage**.
Additionally, **request/token rate limits** are applied. CLI query tool: recent requests, top users, summaries.

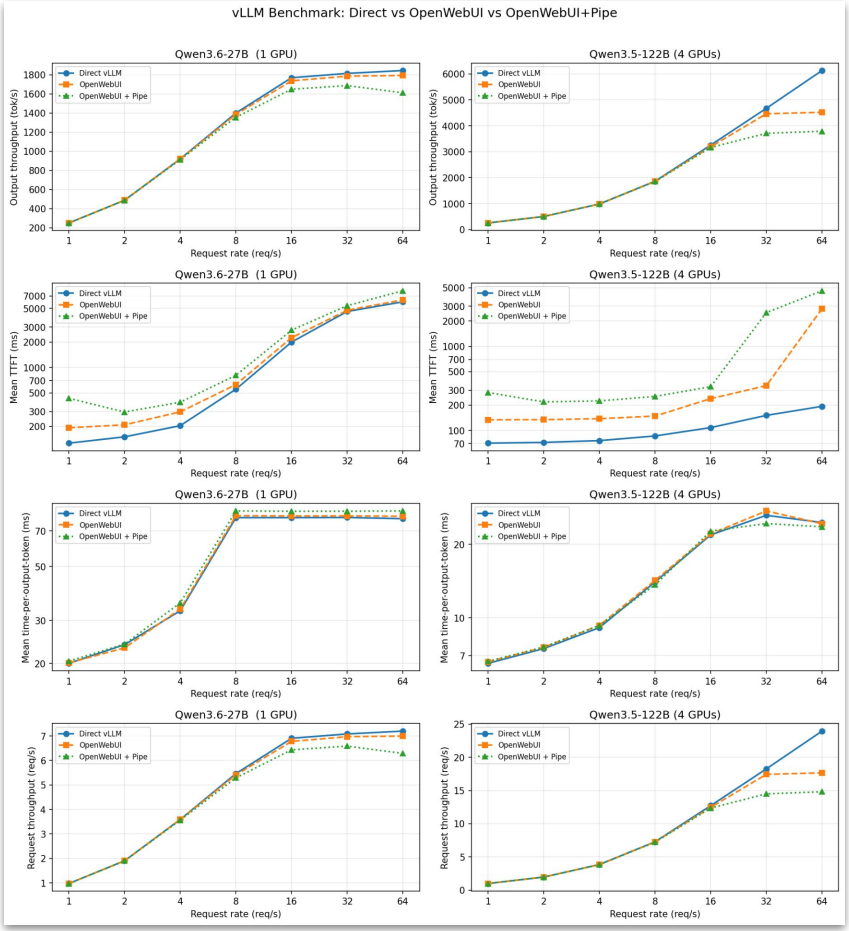


Rate Limit Status

Daily limit of **50,000,000 tokens per day per model**.
Not set in stone, will evolve with the requirements and the available resources.

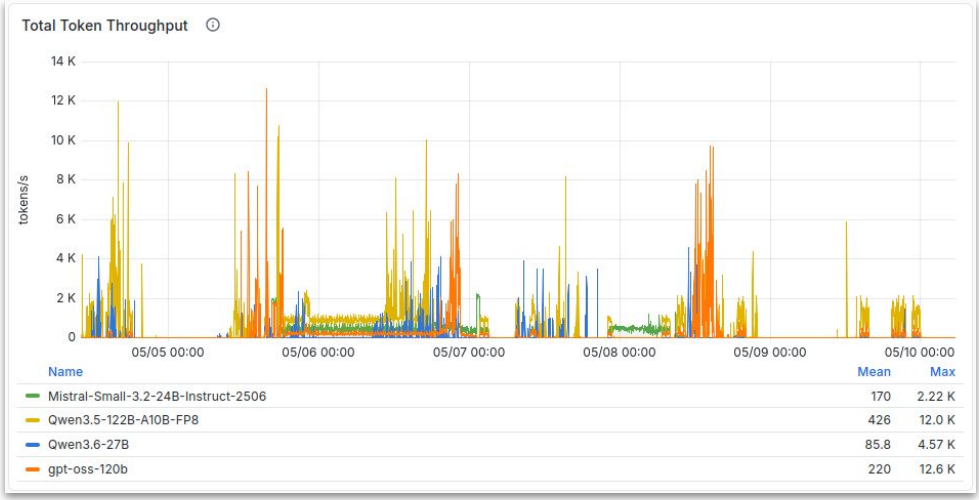


Performance & Monitoring



- Stay up to date with vLLM developments, recommended model settings, optimizations for our hardware & use cases.
- Measure and minimize overhead from frontend and custom functions.
- Monitoring via Prometheus+Grafana:

  request rates, token throughput, input/output sizes, request phase timings, E2E latency, time to first token, inter-token latency, KV cache utilization, cache hit rate, GPU temperatures/load/memory, etc.



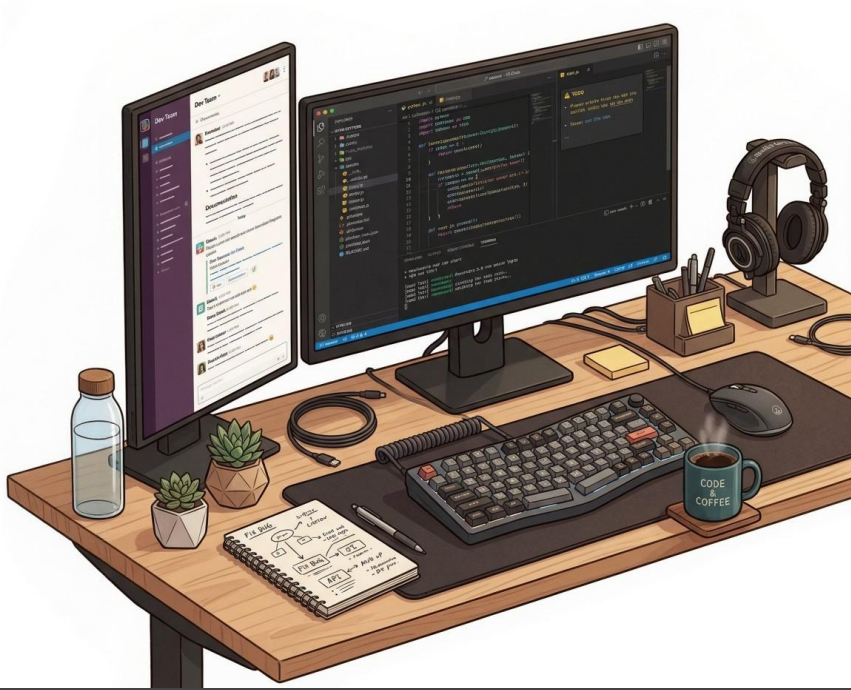
Development & Testing

Development setup uses production docker compose file, with a few overrides for dev models/settings where needed. Otherwise identical software stack. This allows extensive testing of fast-moving dependencies.

- Minimum of one inference model + embedding model.
- Model selection determines VRAM floor.
- Compatibility stubs for other models.

Simplified dev infrastructure:

- Local storage - no NFS mounts or fixed UIDs required.
- No TLS - HTTP on port 80 via nginx-dev.conf.
- No OAuth - local login form with auto-admin signup.
- Single-command start after ./setup-dev.sh.



Current Focus

- Evaluate new models.
- Improve document processing - conversion and/or inference with XLS, PPT, Images, etc.
- Explore integration of different systems into LLMBot - e.g. support tickets, active codebases.
- Evaluate more harnesses/tools and document how they can be connected to LLMBot.
- Guide users on how to optimally enable their workflow with LLMBot (website vs. API).
- Keep track of dependency changes and avoid premature/broken features.
- Contribute back to used projects, where meaningful (OpenWebUI, vLLM, Docling - issues/PRs).

The service is currently still in a test phase - contact Mohammad Al-Turany for access request.

For support contact **ai-service@gsi.de**.