

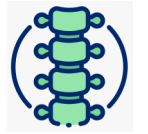
Home-lab-2-prod-ai-golden-path



September 2025

1- Objective

- Play with AI with easy entry level concepts from SRE - Platform or Data Engineer – Architect – DevOps **eyes**.
- Create a imperfect POC to play like a **backbone** that can be enhanced little by little. ****Use case****: ingest your intranet and local documents into a RAG database an connect a local LLM/SLM. We test MCP architecture development against our local database as well.
- Create a **golden path**, opinionated shortcut, that can be reused from home lab to PROD and can ease your way to start playing with AI.
- Mini state of the art: from silo databases towards MCP agents.

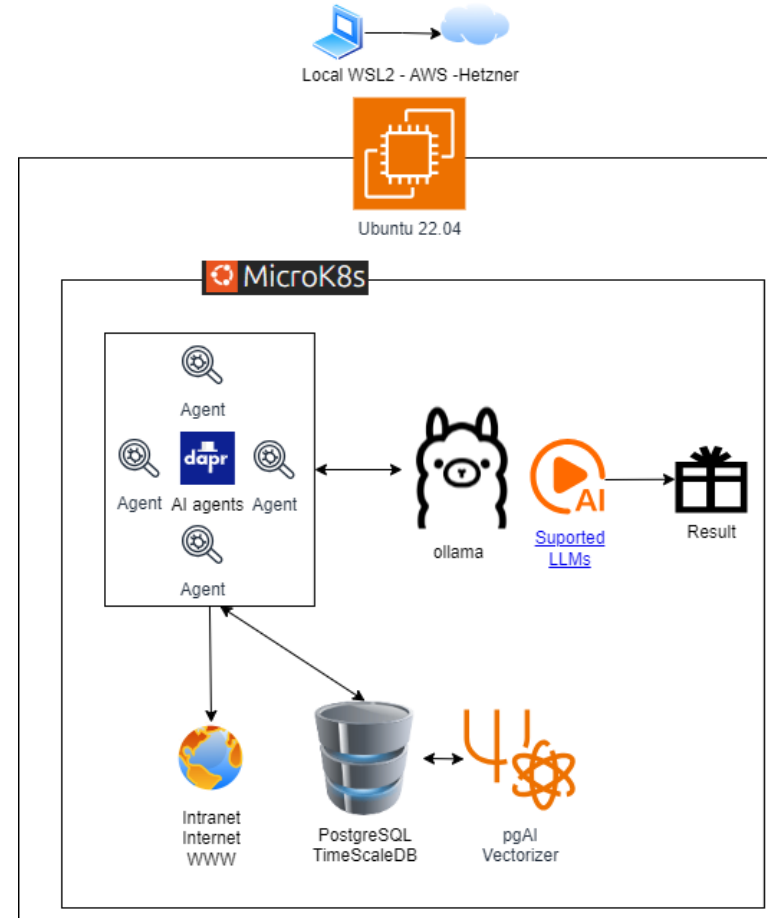


1- Objective: golden path 1st step

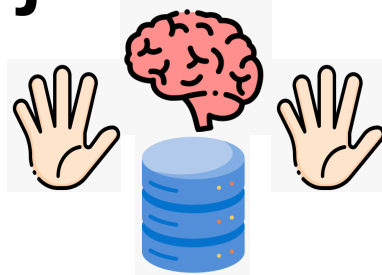


- Select and integrate OSS projects curated by CNCF to respect vendor neutrality.
- Privacy: local AI and local data.
- Technological independence, Europe first.

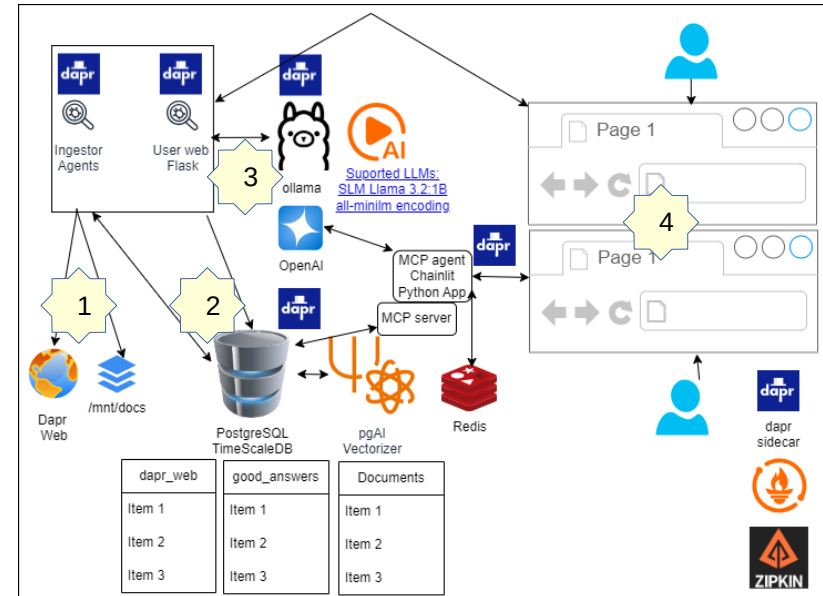
Tariffs 15 %



1- Objective: create a 1st POC



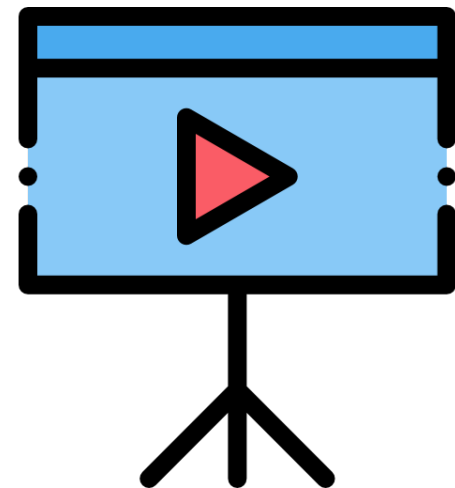
- Broad definition of **AI agents + RAG** : AI Agents (hands, gets data) - LLM/SLM (brain, process data and answers) – RAG database (authoritative knowledge base).
- 1 • Python program scraper reads web content and save into PostgreSQL DB . Another agent reads & saves documents.
- 2 • **Pg AI** chunks and vectorizes data **automatically**, this is our RAG database where semantic search is done. Pg AI can vectorize sources from local or S3 or PostgreSQL and can integrate with Ollama or OpenAI LLM.
- 3 • **Ollama** allows you to run large language models (LLM & SLM) locally on your own computer. No GPU needed.
- 4 • Python Flask web where user can write prompt and receive and answer based on local RAG database and local Ollama LLM/SLM, optionally configured with OpenAI as well. Good answers can be saved back to RAG database. MCP app & architecture querying PostgreSQL database with OpenAI.





2- Demo time

- LLM (Large Language Model) vs SLM (Small): trained with less parameters so we need less hardware resources to run it. Demo can be run in a 5 years old laptop WSL2 Ubuntu with 10GB RAM & 4 cores assigned.
- Ollama LLM/SLM selection: **llama3.2:1b** is an SLM which **license** is not fully OSS but is very permissive, supports all-minilm encoding, honesty temperature parameter / prompt, can use RAG (authoritative knowledge base) and supports different language support (original RAG data language matters in semantic search). **TinyLlama (1.1B)** is fully OSS but performance is a little behind **llama3.2:1b**.
- TimescaleDB/PostgreSQL RAG database vectorizer encoding is setup to all-minilm which is OSS and maps sentences & paragraphs to a dimensional dense vector space and can be used for semantic search. **Performance:** Ollama CPU, chunk size, original data language, save contexts and metrics (tokens, time ...) to integrate with Prometheus & Zipkin.
- Dapr microservices **building blocks** – maximizes standardization and code reuse changing configuration files only. **Dapr is also evolving into an AI Agent framework**
- **Dapr eases Open Telemetry monitoring** Zipkin tracing – Prometheus metrics.





2- Demo time: project structure

- Github – MIT license. "Built with Llama" extra license. TimescaleDB license is **against cloud Hiperscalers only**.
- Detailed explanations in README.md in each folder.
- 1-IaC: shell script to install in WSL2 - OpenTofu IaC for Hetzner (U know, Europe first) & AWS. OpenTofu supports **many providers**.



```
— 1-IaC: we choose kubernetes microK8S implementation as the neutral vendor platform to run differents POCs
  |— AWS: TF/OpenTofu scripts to run it in AWS Ubuntu 22.04
  |— WSL2: steps/shell script how to install it locally in WSL2 - Ubuntu 22.04
  |— OVH-Hetzner: TF/OpenTofu scripts to run it in Hetzner Ubuntu 22.04 European provider
— 2-mandatory-k8s-services: mandatory services to install in K8S
  |— dapr: distributed application runtime helping microservices standardization and agent implementation
  |— ollama: local LLM/SLM offering an API
  |— timescaleDB: postgresSQL with vector database support, our RAG database using pgAI vectorizer
— 3-dapr-microservices-agents: microservices and agents inserting data into database and passing it through to LLM/SLM
  |— 1-user-web
  |— 2-injection-agent-web-dapr
  |— 3-injection-agent-docs
  |— 4-MCP
— 4-optional-k8s-services: monitoring with Zipkin, Prometheus & Grafana - DONE.
  |— Jupyter Notebooks, MLflow vs KubeFlow, ArgoCD - Pending.
— Docs: some documents and references to help to have an holistic/global view on AI
```





3- From this POC to ...

- You will find that getting the right data samples and use cases are key.
- Depending on your data type: text or language data can work better with semantic data search (understand meaning of natural language better with LLM& SLM) vs image similarity & numerical data & anomaly detection sensor data (could work better with ML or deep learning).
- This POC is suitable for Lab/Dev environments & Educational environments to gain hands-on experience.
- This POC might work in Raspberry Pi industrial environments on the edge.
- You can also play with Machine Learning installing **MLFlow** or **Kubeflow** or **both** on microK8S (MLFlow better for tracking experiments VS Kubeflow better for pipelines).
- The AI agent concept is a rapidly evolving trend and Dapr offers its **Dapr AI Agents framework** and **MCP Quickstarts**, see **MCP architecture image**.
- SLMs are ideal for specific domain knowledge and the **future of Agentic AI according to NVIDIA**. Install and play, have your own internal RAG database filled with your web/intranet or documents or other data and expose it to your own users. Reinforce feedback & learning.
- **RAG architectures** and **AI Agents vs Agentic AI and AI design patterns**.
- AI agents are 10% AI and 90% software engineering **iceberg** or **7 layers**.



4- AI strategy: golden path vs Luke



- MIT **report 95% of GenAI projects are getting zero return**. Systems do not learn. Focus on ROI business cases. User context & remembering.
- From lab to PROD imperfect path: IMHO technological change management best practice should be this way, see **image**.
- Data quality is the most important and still may be an undervalued factor in AI. **There is a huge Diogenes syndrome in companies data**. Do you want to be a data engineer hero? - **AI and synthetic data pros and cons**.
- Quality AI services are paid + Cloud **Technofeudalism** + 15% tariffs. Are Cloud repatriation & OSS the medicine we need?  
- AI twister: too many AI news, all is AI. Impostor syndrome. Cognitive overload. **Gartner's Hype Cycle**. We must invest in AI strategies suitable for every company's different state.
- Make your own path - Use the force, Luke!
 - What you hear, you will forget - What you do, you will understand - Explain to others, you will understand even deeper - Do not be a perfectionist and fail fast to learn fast. Play hard to work smart.
 - If you want to be fast, do it alone. If you want to go far, do it with company. If the company is not good, it is better to go alone.



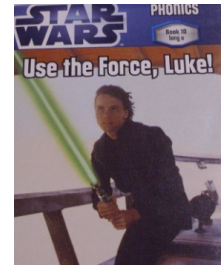
Scarecrow: he seeks a brain, believing he lacks intelligence. **AI ML & LLM & SLM are not intelligent, they need good quality data, otherwise they misinform.**

Tin Woodman/ Tin man: He desires a heart, feeling he is incapable of love and emotion. **IT & AI must ease everybody's work and not make it more difficult. Why are we so stressed?**

Dorothy Gale: she feels out of place in OZ, wants to return to Kansas. **Probably like you in this AI twister.**

Cowardly Lion: He wants courage, thinking he is too fearful to be brave. **Accept that many strategic IT implementations need a change and start doing them right, no excuses.**

Ultimately, the journey reveals that they often possess the qualities they seek within themselves, even before they meet the Wizard of OZ. **Do you really need any AI wizard/guru?**



5 - Q&A - Contribute



- Share your experience or thoughts.
- Tell me if you need any help with your lab.
- This project is a imperfect learning by doing experience - If something is wrong or there is an error, comment it but do not be a hater ;)
- Contact points: this [YouTube video](#) or [GitHub](#) or [LinkedIn](#).





6- Some images references

- Most of the icons in this presentation are from flaticom.com, flaticom license (Free for personal and commercial purpose with attribution)
- Trump's image [link](#)
- Wizard of Oz 1st slide image [link](#)
- Use the force Luke image [link](#)