# History, Design and Future

Why the heck should we need an SDK on top of FastAPI, you may think or ask yourself.

> What's the reason for this project? [...]

Here's a story of the reasons which drive me to build this.

## Background

I have been creating Software Applications, Solutions and APIs with complex requirements for decades (Mainly for AI Systems based on ML/DL and NLP/NLU, in distributed systems, asynchronous, with NoSQL and SQL databases), leading several Software companies and Development Teams.

As part of that, we always had to build Showcases, PoC's, MVP's and finally a Production Version.

While we used for decades Java with the Eclipse Vert.x library, since 2018 we fell in love with Python for AI Use Cases and replaced java with Python3.

As the development stack and architecture shifted 100% to RestAPI, we stumbled over **FastAPI** and tiangolo's work. Hies ideas and concepts was a match made in heaven.

The history of **FastAPI** is in great part similar to our history with Python and is now our core Framework.

We were only missing some major feature and functions we had with Vert.x, there was still a big hole in our new stack.

Then we found **dapr** and a final Vert.x replacement that even fits better our needs now with docker, kubernetes etc.

Our customers run on Azure, AWS, Google etc., basically they all run on private cloud but more and more need native cloud solutions.

## Micro Services Architecture with Hybrid AI and a kind of Hybrid Service Oriented Architecture

In the past we usually did build monolitic solutions with an RestAPI endpoint.

Today, we use a Micro Service Architecture to provide a SOA layer. This added new challenges and risks. With all those 100's of services and the freedom of the developer building them, the security issues were increasing.

So we used a Boilerplate approach to ensure we always use the same stack of tools and libraries, but as we all know today this doesn't really solve the problem on the long run.

Also the UI challange wasn't solved using Data Analyst Tools like Dash, Streamlit etc. for PoC's they were not efficient enough to give the customer the right impression of possible UI's etc., esp. for a final solution impression.

So there was and is a need to build some sophisticated modern Web UI to showcase even a simple API feature set, we developers are fine with OpenAPI/Swagger but customers mostly are not.

You mostly also have some structured data in SQL type of DB's, with a need for some CRUD UI's and a document oriented DB like MongoDB to store Model structures.

Also for FastAPI we always have the same need for Middleware, Profiling etc.

If you create many PoC's, MVP's and finally the Services, you repeat a lot of things over and over again, sounds like a old know fact but get's repeated with any new technology coming along.

## Design

The design approach was very pragmatic, leave out the AI based libraries and condense to an SDK which holds what we use and need in 80% of our API's and Micro Services.

So we just made a list of what the minimum stuff is we want:

- Pure Python 3 with as much as possible Type hints., no java bridges etc.
- **dapr** as our distributed runtime
- Well for sure **FastAPI** with Pydantic, orjson, Jinja2 Template Engine, gunicorn, uvicorn and uvloop
- Starlette extensions and Middleware
- SQLModel with SQLAlchemy
- The major aio libraries: aiomultiprocess, aiofiles, aiosqlite, aioredis etc.
- Loguru for logging
- The usual datatype parsers for xml, html, dates etc.
- pandas, numpy, pillow, matplotlib... the usual stuff
- For the UI we wanted a python lib to create React/Vue.js UI's, so we decided to use Amis
- mkdocs with mkdocstrings and the mkdocs-material template for our documentation needs, esp. the automation of code reference documentation

So we collected all the stuff from past projects, did some major re-factoring (we needed to ensure that when we overload or forked stuff that we won't have name conflicts).

### Finally...

Beside the need we had for our own work, we decided to also give back to the community as we heavily use open source and community help aswell.

There is also a tiny hope that some of you may find this helpfull and maybe even want to contribute and join us, yes that is an invitation...

---

Last update: September 13, 2022
Created: September 11, 2022