Write and prod an efficient REST API with API-Hour in 5 minutes

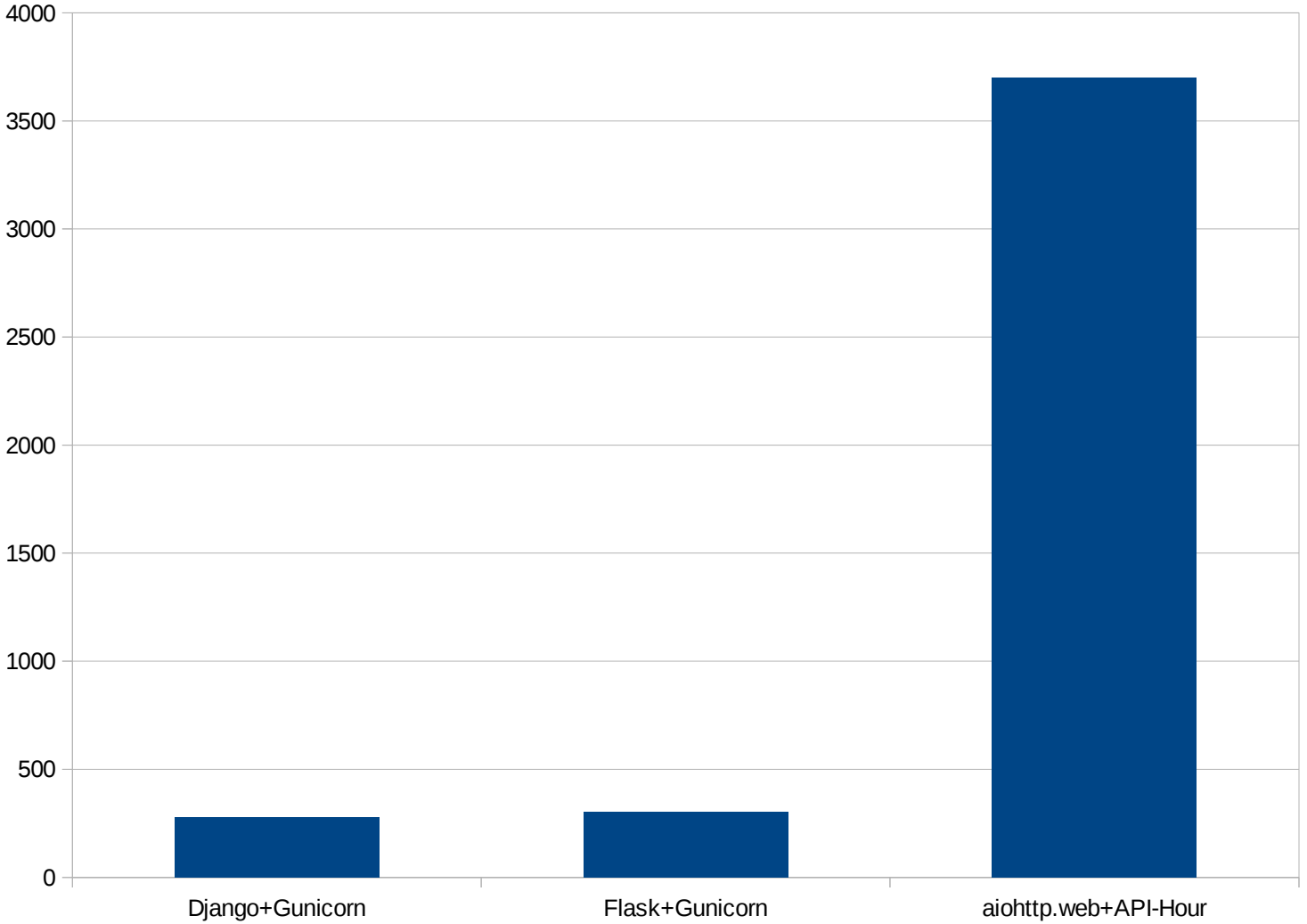
WHY ???

We don't like to suffer

Day	Task	like ?
1	Meeting with clients	Easy as a pie
20	Write business logic	Boring, simple, easy
200	Deployment	*No !*
200,00001	Debug low level issues	**Wait... WHAT ?**
290	Handling heavy load	**WTF ????**

Concretely for my business ?

- Handles a lot more HTTP requests with Python (See: http://blog.gmludo.eu/2015/02/)



What is API-Hour ?

- API-Hour enables easily multiprocessing for AsyncIO daemons
- Based on Gunicorn with a custom worker
- AsyncIO + API-Hour = Performant

API-Hour philosophy

- KISS philosophy: Easy to understand and deploy
- Use AsyncIO libraries directly
- A Starter-Kit to create quickly your HTTP/REST daemon ready for production
  (with system config files + Ansible playbook)

API-Hour Container Architecture

- **Container**: An object that represents your application with everything inside: routing...
- **Endpoints**: Simple Python coroutines called when your Application received requests (HTTP, SSH, Async AGI, AMI events...)
- **Engines**: Data source providers for Services. Example: PostgreSQL, Asterisk, CouchDB...
- **Services**: Where you transform data for Endpoints. It represents your business logic and your internal Python API.