# Hecate2
## Electrical and Computer Engineering

 github.com/Hecate2  @ hecate2@qq.com   SECRET_PHONE

##  EDUCATION

| | |
|---|---|
| 2016-2020 | **Fudan University**, Bachelor of Science, Electronic and Information Science and Technology |
| 2020-2021 | **Georgia Institute of Technology**, Master, Electrical and Computer Engineering<br>(2-year program finished in 1 year) |

##  COMPETENCES

| | |
|---|---|
| Languages | <u>Python</u>, **C#**, **Go**, JavaScript (TypeScript), **Solidity**, C(++), **MATLAB**, **verilog**, HTML, Java, … |
| Fields of Competence | blockchain (VM, DeFi, …), c100k concurrency, web scraping, data-intricate web backend |
| Fields of Interest | open-source, ML&DL, ETL, c10m concurrency, compiler, AUTOSAR, ASIC, medical imaging, … |
| Insania Fast Learner | Never fear knowledge! I swallow docs & source codes & papers and learn insanely fast! |

##  EXPERIENCES

**May 2021 TILL NOW** — Blockchain developer, SMART CONTRACTS, INFRASTRUCTURES & SECURITY AUDITS, neo.org
- Revolutionized testing & debugging tools for Neo blockchain : <u>Fairy</u> (chain node plug-in for RPC-based testing & debugging) with an automated <u>client</u>. Similar to but more powerful than <u>hardhat</u> & <u>truffle</u>.
- Miscellaneous docs & <u>tutorials</u> for debugging Neo blockchain at deepest source-code level.
- Developed miscellaneous DeFi & DAO smart contracts : <u>NeoBurger</u>, <u>ruler protocol</u>, <u>Divisible NFT</u> standard & loan, etc. Researched on MINLP algorithms to maximize profits.
- Helped find VM instruction-level issues in Neo's compilers "neo3-boa" (Python) & "nccs" (C#).
- Helped analyze Neo's VM, garbage collection, etc. , which lead to reveals of security vulnerabilities.
- Performed on-chain data analysis; <u>monitored</u> some popular DeFi protocols on Ethereum.
- Helped build a webpage-based functional demo utilizing Neo's features : NeoFS, digital ID system, oracle & smart contracts. Analyzed the front end (using React) and smart contract codes.

`Neo blockchain` `smart contract` `DeFi` `neo-vm` `compiler` `tester` `debugger` `security audit` `NeoFS` `Neo ID` `Solidity` `C#` `node.js` `React`

**June 2019** — Ignareo, HTTP SPIDER CORE OF ULTIMATE CONCURRENCY, 150+ stars
- Spiders packed in asynchronous tornado servers
- Distributed broker architecture : aspects packaged as micro–service nodes
- Covering shortages of scrapy : non-blocking sleep, user codes of better cohesion, freedom and tidiness; scalability with customized load–balancing
- Seamless compatibility with anything based on asyncio or gevent
- Combat-proven performance and stability

`distributed` `web spider` `asyncio` `aiohttp` `gevent` `requests` `tornado`

**February 2020 June 2020** — Intern backend developer, FEDERATED LEARNING BACKEND WITH DATA PRIVACY, www.points.org
A web service for cross–organization federated machine learning. Based on distributed datasets from different organizations, the training process should still keep strict data privacy of each data owner. The service is principally based on gRPC instead of HTTP.
- Basic but complete experience of commercial development flow : designing, coding, deploying, testing, pull request and code review. 111052++, 86888– lines of code commits in the main repository.
- Designed and wrote a user and API authorization backend. Wrote a data-usage approval backend.
- Referring to horizontal learning APIs, developed vertical learning APIs. These APIs mostly involve data manipulation, but do not include core encryption and secret–sharing algorithms for learning.
- Wrote a basic service to retrieve real-time logs from docker containers and feed them to the frontend.
- Developed basic automatic cross-system process tests for the APIs described above.
- Implemented a mechanism to stop distributed vertical or horizontal training.
- Built a basic client for on-line model inference service stress test, using Ignareo.
- Tentatively designed a RESTful service for data providers to manage data, using flask and swagger.
- Participated in software copyright registration and federated learning product standards.
- Wrote docs and user manuals.

`protobuf` `gRPC` `sqlalchemy` `redis (message queue)` `tensorflow-federated` `dockerfile` `docker-compose` `git` `Flask` `swagger` `Go`

| August 2021 | Intern backend developer, Web Spider & Flask Backend for OpenMMLab Operation Service, SenseTime × Shanghai AI Lab |
|---|---|
| September 2021 | › Maintained a system to fetch information from OpenMMLab's Github repositories and display info on a web page; send weekly e-mail reports based on fetched Feishu docs. Fixed miscellaneous bugs and implemented back end features. |
| | › Designed a process to migrate data from SQLite to MySQL (using DataGrip). |
| | › Designed code for runtime multi-thread operations in Flask. |
| | DataGrip  docker-compose  Flask  SQLAlchemy  apscheduler  requests |

| July 2019 | Intern developer, High Concurrency Web Spider for Monitoring, Johnson Controls |
|---|---|
| August 2019 | › Monitored tens of thousands of IoT devices with help of Ignareo. |
| | › Analyzed anomaly and output results to Excel sheets. |
| | › Provided a web service to browse and download the Excel files. |
| | Ignareo  requests  tornado  xlwings |