

ASKIN CIHAN KILIC

MID / SENIOR PYTHON DEVELOPER | AI SYSTEMS ENGINEER

Warsaw, Poland | +48 501 255 736 | askincihanpl@gmail.com

PROFESSIONAL SUMMARY

Mid/Senior Python Developer and AI Systems Engineer focused on production-grade local AI systems, backend orchestration, concurrency-safe state management, GPU inference pipelines, and reliability engineering. Hands-on ownership across architecture, Python implementation, Linux integration, adversarial testing, deployment safety, observability, and rollback design. Strong at debugging complex interactions between Python services, model runtimes, plugins, process lifecycles, SQLite, and infrastructure.

CORE TECHNICAL STACK

Python Engineering	Python 3, stdlib, pathlib, subprocess, threading, sqlite3, AST, JSON/YAML, unittest/mock
AI / LLM Systems	Hermes Agent, Ollama, local LLM orchestration, tool/plugin hooks, model routing, delegated agents
GPU / Image Inference	PyTorch, diffusers, Transformers, CUDA, BF16, FLUX.2 Klein, sequential CPU offload
Infrastructure / Ops	Linux, Bash, PowerShell, Windows Server, networking, SQL, monitoring, automation, Git/GitHub

SELECTED ENGINEERING PROJECT

OMNI CORE - Private Local AI Operating System	2026
Lead Python / AI Systems Engineer Warsaw / Local GPU Infrastructure	
- Architected a multi-model local AI platform with a permanent MAIN intelligence, delegated CODER specialist, and on-demand IMAGE specialist, integrated through Hermes and Ollama without replacing working infrastructure. - Designed and implemented a production Resource Manager in Python with SQLite as the authoritative state store, transactional leasing, BEGIN IMMEDIATE concurrency control, exactly-one-winner acquisition, heartbeat renewal, dead/zombie process recovery, and monotonic lifecycle transition logging. - Integrated specialist lifecycle management through native Hermes plugin hooks (pre_tool_call, post_tool_call, subagent_stop, post_llm_call) while keeping vendor delegation code untouched and preserving rollback safety. - Built lifecycle adapters for real CODER and IMAGE workloads, including model switching, resource ownership, completion/failure handling, MAIN restoration, and production DB auditability. - Developed a local FLUX.2 Klein image runtime using PyTorch/diffusers with BF16, CUDA GPU isolation, sequential CPU offload, atomic output writes, metadata persistence, offline model loading, and artifact/path validation. - Created independent adversarial acceptance harnesses with Python unittest/mock, temporary SQLite databases, hard process watchdogs, hash-based immutability gates, model/GPU monitoring, rollback automation, and whole-system regression checks. - Validated the integration with an 83-test adversarial suite and real production lifecycle evidence for native CODER and IMAGE execution, including correlated leases/jobs, heartbeats, completion, and MAIN restoration. - Applied defensive subprocess and filesystem practices including shell=False, constrained output roots, JSON contract validation, PNG validation, timeout handling, fail-closed behavior, and no secret exposure to frontend paths.	

PROFESSIONAL EXPERIENCE

IT Specialist / Python Developer
Technology Company | Warsaw, Poland

Jul 2024 - Present

- Develop Python-based automation and internal tooling while supporting Linux/Windows environments and day-to-day technical operations.
- Work across server administration, networking fundamentals (TCP/IP, DNS, DHCP, subnetting), scripting, databases, monitoring, troubleshooting, and process automation.
- Translate operational requirements into repeatable scripts and tools, with emphasis on reliability, traceability, and minimizing manual intervention.

ENGINEERING STRENGTHS

- **Python backend & systems programming:** State machines, lifecycle orchestration, process control, safe subprocess execution, filesystem contracts, configuration handling.
- **Concurrency & persistence:** SQLite transactional coordination, lease ownership, heartbeat systems, race-condition testing, recovery and integrity checks.
- **LLM / agent orchestration:** Local model routing, native tool delegation, specialist isolation, plugin hooks, model swap workflows, context/runtime validation.
- **Reliability engineering:** Adversarial tests, regression gates, production immutability checks, checkpoint/rollback plans, failure injection, evidence-driven acceptance.
- **GPU inference engineering:** CUDA-aware runtime setup, local model loading, memory-aware offload, artifact validation, real workload monitoring.
- **Infrastructure debugging:** Linux services, shell scripting, process inspection, logs, networking, monitoring, Windows Server and infrastructure fundamentals.

TECHNICAL SKILLS

Advanced / Primary Python, Linux, Bash, SQLite, automation, backend integration, debugging, test engineering

AI / ML Systems Ollama, Hermes Agent, PyTorch, diffusers, Transformers, FLUX, local LLM deployment and orchestration

Backend / Data SQLite, SQL, JSON, YAML, REST-style integrations, process and filesystem APIs

Systems / Infrastructure Windows Server, Active Directory / Group Policy fundamentals, TCP/IP, DNS, DHCP, subnetting, VMware, AWS fundamentals

Tooling Git/GitHub, PowerShell, monitoring concepts, Nagios background, Ansible background

EDUCATION & TRAINING

Computer Engineering
Management Academy of Applied Sciences in Warsaw | Online studies

- Computer Engineering studies with a software, systems, and infrastructure focus.

Python Developer Training - Turkish Aeronautical Association University

LANGUAGES

Turkish - Native | **English** - Advanced