Adam Horacek

SENIOR / LEAD iOS / macOS ENGINEER · AI-NATIVE APPLE PLATFORM & AGENTIC SYSTEMS ENGINEER

Swift 6 • SwiftUI • Structured Concurrency • OpenAI Agents SDK • Agent Builder • ChatKit • Codex • On-Device AI

- Tel: +420 775 622 608
- E-mail: adam.horacek@me.com
- GitHub: https://github.com/adamhoracek
- LinkedIn: https://linkedin.com/in/adamhoracek
- PDF: https://adamhoracek.com/Adam-Horacek-CV.pdf

SUMMARY

Senior iOS/macOS engineer with **15+ years** building and shipping Apple-platform software across fintech, security-sensitive, and performance-critical environments. Strong in **modern Swift**, **Swift 6 / strict concurrency**, SwiftUI, architecture, CI, debugging, and production releases.

Currently focused on **AI-native Apple-platform and agentic systems engineering**: combining senior mobile architecture with **OpenAI agent workflows**, local/on-device inference, and privacy-conscious LLM-assisted delivery. Comfortable across both major implementation paths: **hosted Agent Builder + ChatKit** workflows for product-facing agent chat experiences, and **local/code-first OpenAI Agents SDK** implementations when the application needs to own orchestration, tools, approvals, state, storage, deployment, and runtime behavior.

Practical with **Codex**, **Claude**, CLI agents, repo-local instructions, skills, plugins, MCP tools, worktrees, review loops, and verification commands for implementation, refactoring, migration planning, documentation, debugging, and code review.

Hands-on ML background including **custom neural networks and classifiers**, TensorFlow 2 coursework/certification, and local inference experiments using **Core ML**, **ML Kit**, and Apple-platform runtime integration. Interested in AI systems that are not just demos, but maintainable, testable, privacy-aware features embedded into real applications.

Systems thinker with strong **pattern recognition**, enabling fast end-to-end mental models, **early** detection of hidden coupling, and design for real operational constraints.

PROFESSIONAL EXPERIENCE

Independent Contractor

Senior/Lead iOS/macOS Engineer · AI-Native Apple Platform & Agentic Systems Engineer

Contract • Remote • 07/2023–Present

Freelance/contract engineering across Apple-platform codebases, with recent focus on modern Swift, AI-native development workflows, OpenAI agent product architecture, and local/on-device AI integration.

- Contract delivery across iOS/macOS/**tvOS** codebases

- **Swift 6 migration**: resolved **Sendable** fallout, actor-isolation refactors, strict concurrency issues, and race conditions
- Designed and used **agentic development workflows** with **Codex**, **Claude**, CLI agents, repo-local instructions, skills, plugins, MCP tools, and task-specific orchestration
- Used Codex workflows including **Plan mode** and **Goal mode** where useful for scoped planning, longer-running tasks, review loops, and verification
- Worked with current **OpenAI agent product concepts**: **OpenAI Agents SDK**, **Agent Builder**, **ChatKit**, tool calling, MCP/connectors, state, guardrails, handoffs, traces, evals, and human review
- Mapped and prototyped both **hosted** and **local/code-first** agent paths: **Agent Builder workflows + ChatKit embeds** for OpenAI-hosted product chat surfaces, and **OpenAI Agents SDK** implementations when the application owns orchestration, tools, approvals, state, storage, deployment, and runtime behavior
- Orchestrated AI agents for implementation, refactoring, code review, migration planning, documentation, debugging, and verification
- Built AI-assisted workflows around explicit context, scoped instructions, reproducible commands, tests, review loops, and auditable changes
- Worked with **local/on-device inference** patterns, including connecting models into app architecture through **Core ML** and **ML Kit**
- Built personal/R&D ML systems including **custom neural networks and classifiers**; TensorFlow 2 certified
- Explored local inference architecture for privacy-conscious AI features close to the device/runtime boundary
- Introduced **DTO boundaries** to decouple UI/domain layers
- Integrated **Kotlin Multiplatform (KMP)** shared modules into an iOS app
- Modernized SwiftUI stack using **Observation** and a **VM-less MV (Model–View) approach**
- Implemented **Swift Charts** (with Metal overlays)
- Strengthened unit, **integration**, **snapshot**, and UI tests; wired into CI

Deutsche Bank

Lead iOS Engineer

Contract • Remote • Frankfurt am Main, Germany • 09/2020–06/2023

Greenfield, modular, white-label mobile banking + signing iOS applications.

- Designed and implemented a **modular Clean Architecture** with **Coordinators** and **RxSwift**
- Owned critical user flows: **authentication/session lifecycle** and **transaction** verifications
- Implemented **SwiftPM-level build customization** (custom code exclusion/feature gating + secure config/secrets injection at build time)
- Built **CI / environment configuration** patterns: secure secrets handling, env-specific configuration, and safe release pipelines
- Drove security hardening work (threat-model mindset, defensive coding)
- Mentored engineers via code reviews and architecture guidance
- Worked in regulated delivery environments (CI, SDLC, SAFe)

Apps shipped / contributed to

- Deutsche Bank — https://apps.apple.com/us/app/deutsche-bank/id1040475847
- Postbank — https://apps.apple.com/us/app/postbank/id6444706327
- FYRST — https://apps.apple.com/us/app/fyrst/id6447359205
- Postbank BestSign — https://apps.apple.com/us/app/postbank-bestsign/id1442251022

Deutsche Bank

Senior iOS Developer

Contract • On-site/Remote • Frankfurt am Main, Germany • 02/2018–03/2019

Enterprise iPad applications (internal distribution; used by advisors in branches).

- Built features using **Clean Swift (VIP)** architecture
- Collaborated with Apple's Enterprise Design Labs
- Diagnosed and resolved **concurrency/data race** issues to improve stability and performance
- Improved developer experience: deprecated brittle binary dependencies, introduced Workspaces, and reduced build friction across teams
- Strengthened testing culture (unit/UI tests) and high-signal code reviews

---

VEON

Senior iOS Developer

Contract • On-site • Amsterdam, Netherlands • 12/2017–01/2018

- Modernized a high-traffic messaging/VoIP app by migrating Objective-C components to Swift
- Refactored toward VIPER + SOLID to improve maintainability and long-term velocity

---

Sapient Nitro

Senior iOS Developer

Contract • London, UK / New Delhi, India • 04/2017–06/2017

- Delivered UI/UX improvements and new iPad features; improved iPhone app quality in collaboration with an offshore team
- Raised engineering standards via code reviews and pragmatic best practices

Apps shipped / contributed to

- European Tour — https://apps.apple.com/us/app/european-tour/id573521629

Aegon

Senior iOS Developer

Contract • On-site • The Hague, Netherlands • 11/2016–02/2017

- Re-architected UI layers (MVC/MVVM) and implemented client-side auth (JWT/SSO)
- Strengthened security posture through crypto-oriented improvements and defensive coding

---

Qredo

Senior iOS Developer

On-site • London, UK • 10/2015–12/2015

- Consolidated Objective-C/Swift iOS SDK
- Improved stability by fixing hundreds of failing tests
- Optimized multi-threading/message queues and strengthened secure communications (binary protocols, Security framework)

Farm At Hand

Senior iOS Developer

On-site • Vancouver, Canada • 02/2014–07/2014

- Built a farm management iOS app (offline multi-user support & synchronization)
- Contributed to API design, DevOps, and hiring process

Apps shipped / contributed to

- Farm At Hand — https://apps.apple.com/us/app/farm-at-hand/id1089000986

CORE STRENGTHS

- **Swift 6 & concurrency correctness**: async/await, actors, Sendable, strict concurrency checks, TaskGroups, cancellation; race condition debugging
- **SwiftUI stack**: SwiftUI, **Observation (@Observable)**, SwiftData; UIKit/AppKit interoperability
- **Charts & visualization**: **Swift Charts**, custom rendering where needed; performance tuning (Instruments, signposts), Metal when beneficial
- **Synchronization**: **Synchronization framework (Mutex)** for safe shared state where actors aren't the right trade-off
- **Architecture**: MVVM(+C) + Coordinator, Clean Architecture, Clean Swift (VIP), VIPER, **MV (Model–View) SwiftUI (view-model-less / Observation-driven)**
- **Systems thinking (pattern-first cognitive style)**: tends to process information differently, surfacing hidden coupling, edge cases, and second-order effects early; translates that into pragmatic architecture and clean interfaces.
- **Quality & delivery**: Swift Testing / XCTest, UI tests, snapshot tests, integration tests, CI/CD, code review, SDLC/SAFe environments
- **Security mindset**: auth/session design, secure config handling, reverse-engineering awareness

AI-native / agentic development

- **OpenAI agent stack**: hosted Agent Builder + ChatKit workflows, local/code-first OpenAI Agents SDK implementations, Responses API concepts, tool calling, MCP/connectors, guardrails, handoffs, traces, evals, and human review
- **Agentic workflows**: Codex, Claude, CLI agents, repo-local instruction files, skills, plugins, subagent/parallel exploration, and task-specific orchestration
- **LLM workflow design**: context shaping, prompt/instruction design, multi-step task decomposition, tool routing, workflow state, verification loops, and human-reviewed output

- **Software delivery with agents**: using LLM agents for implementation, refactoring, test creation, documentation, migration planning, debugging, and review support
- **On-device AI**: Core ML, ML Kit, local inference architecture, Apple-platform model integration
- **ML foundations**: custom neural networks/classifiers, TensorFlow 2, model training/evaluation basics
- **Privacy-first practice**: no secrets/client-data leakage, minimal/redacted context, policy-compliant usage, and reviewed/tested output before integration

SKILLS

- **Languages**: Swift, Objective-C, C/C++, Python, JavaScript, Bash
- **Platforms**: iOS, iPadOS, macOS, tvOS
- **UI**: SwiftUI, UIKit, AppKit
- **Modern Swift**: Swift 6, Structured Concurrency (async/await, actors, Sendable), Synchronization (Mutex), GCD (legacy/interop)
- **Frameworks**: Observation, Swift Charts, SwiftData, Core Data, Combine/RxSwift (interop), Swift-NIO (when applicable), Metal
- **AI / Agentic development**: OpenAI Agents SDK, Agent Builder, ChatKit, hosted agent workflows, local/code-first agent orchestration, Codex, Claude, CLI agents, skills/plugins, MCP, LLM workflows, prompt/instruction design, verification loops
- **Codex workflow control**: **Plan mode**, **Goal mode**, worktrees, repo-local instructions, custom skills/plugins, MCP tools, review workflows, subagent/parallel task decomposition
- **AI / ML**: Core ML, ML Kit, local/on-device inference, TensorFlow 2, neural networks, classifiers
- **Testing**: Swift Testing, XCTest, UI Testing, snapshot tests, integration tests
- **Architecture**: MVVM(+C) + Coordinator, Clean Architecture, Clean Swift (VIP), VIPER, **MV (Model–View) SwiftUI (view-model-less / Observation-driven)**
- **Tooling**: Xcode, Instruments, Swift Package Manager, Git, Charles Proxy, Wireshark, Figma
- **CI/DevOps**: GitHub/GitLab, CI pipelines, secure environment config

SELECTED PROJECTS (PERSONAL / R&D)

OpenAI Agentic Workflow / Product R&D

R&D around AI-native product architecture and agentic developer workflows, with emphasis on making agents useful, auditable, and shippable inside real software projects.

- Prototyped both **hosted Agent Builder + ChatKit** workflows and **local/code-first OpenAI Agents SDK** patterns around tool calling, MCP/connectors, guardrails, state, traces, evals, and human-in-the-loop review
- Focused on practical agent UX and architecture tradeoffs: when to use OpenAI-hosted Agent Builder + ChatKit for product chat surfaces, when to keep orchestration local/code-first with the Agents SDK, and how to keep product behavior testable and maintainable

Local AI / On-Device Inference R&D

Apple-platform R&D around local model integration, privacy-conscious inference, and app/runtime boundaries.

- Connected local models into Apple-platform app architecture using **Core ML** and **ML Kit**
- Explored local inference pipelines, model boundaries, runtime constraints, and offline-first AI behavior
- Built ML foundations through custom **neural networks and classifiers** using TensorFlow 2
- Focused on privacy, device-local execution, explicit data boundaries, and maintainable integration into real app architecture

Quantitative Trading (Personal Project)

macOS trading workstation focused on high-performance visualization and backtesting.

- Real-time chart rendering, indicators, and backtesting engine
- Broker API integrations (architecture-ready for multiple providers)
- **Tech**: Swift, SwiftUI, AppKit, SwiftData, Swift Concurrency, Metal, Swift Charts

BitTorrent + DHT (Personal Project)

Custom BitTorrent client with a DHT implementation for decentralized peer discovery.

- **Tech**: Swift, Swift-NIO, networking protocols, distributed systems concepts

---

EDUCATION

- Masaryk University — Czechia (Parallel and Distributed Systems) — 2010–2011
- Charles University — Czechia (General Computer Science) — 2007–2009

COURSES

- Claude Code in Action, Anthropic — https://verify.skilljar.com/c/ynuxkpki9wj6
- Getting started with TensorFlow 2, Imperial College London — https://www.coursera.org/account/accomplishments/verify/2SNBXJ9KDZBM

LANGUAGES

- English — C2
- Czech / Slovak — C2
- German — B1