WILLIAM ASTLEY

+1 (518) 902-5805 • Toronto, ON, Canada (US & Canadian Citizen - Willing to relocate)

wrastley@gmail.com • LinkedIn • GitHub • PyPI • Website

Summary:

- Agentic Systems Engineer who independently built architectural frameworks for dynamic multi-agent AI - where agents create other agents, self-modify, self-replicate, and orchestrate emergent teams with scalable memory systems. Developed 100+ implementations from complex simulations (games, debates, social dynamics) to production systems across diverse domains.
- Breakthrough: eliminated rigid architectures through dynamic composition and A2A protocols, enabling truly adaptive AI.
- Self-taught innovator with 20+ open source frameworks spanning autonomous systems and theoretical mathematics

---

Professional Experience

2024-Present: Sole/Lead Developer of Haive

- Long term memory systems with context memory with graph DB and RAG
- Developed RAG architecture that greatly improves performance for large (gigabytes) and diverse (200+ sources) by autonomously optimizing the preprocessing into the vector database via fetch, load, store, transform, annotated, embed, store, and retrieve processes.
- Agentic AI workflows that can self modify, replicate and evolve dynamically to secure optimal tooling and coordination with other autonomous agents.
- Proven and tested these in games, finance, social media conversations, oxford debates, and other applications.

June 2022-Present: Sr. Quantitative Risk Developer, Corpay *Toronto*

- Built financial modeling tools that utilize mathematics, algorithms, AI, & software architecture.
- Designed and implemented automated stress testing pipeline, real time dashboards, capital allocation strategies
- Designed and implemented a proprietary compliance agent-architecture leveraging dynamic RAG and reflection (without observation), accelerating onboarding by 90%. Achieved a 0% error rate and 100% case coverage
- Built GARCH/LLM forecasting models and automated stress testing pipelines, improving forecast accuracy by 35% and cutting execution time by 75%, enabling proactive risk management across a $500M portfolio and

preventing tens of millions of dollars in potential losses during post-COVID volatility.
- Developed a real-time treasury dashboard integrating 10+ predictive models, informing capital allocation and liquidity strategies that contributed to 40% company growth (vs -10% industry benchmark) and supported S&P 125 inclusion

2022-2023 Founder & Lead Developer, Algebraic Wealth Technologies
Toronto, Canada

- Founded and led a 10-person team building one of the first LLM-driven financial intelligence platforms, integrating technical analysis, economic data mining, and agent-based trading architectures.
- Advanced to Y Combinator interview stage based on technical innovation and early traction in pioneering AI agent applications.
- Developed core systems including proprietary document ingestion frameworks, real-time research generation, and sentiment-driven trading agents (details in *Projects*). [WILL HYPERLINK]

TECHNICAL SKILLS

The Agentic AI systems that I have implemented in my entrepreneurial efforts are built using the following technical capabilities.

Multi-Agent Architecture: Dynamic agent composition, Agent-to-Agent (A2A) protocols, Runtime modification, Agent self-replication, Multi-agent coordination, Emergent team orchestration, Distributed consensus

AI/ML Frameworks: LangChain, LangGraph, AutoGen, CrewAI, OpenAI Agents SDK, Hugging Face Transformers, PyTorch, TensorFlow, Vector databases (Pinecone, Weaviate), RAG architecture

Agentic AI Development: Autonomous agents, Self-modifying systems, Agent communication protocols, Multi-agent reinforcement learning (MARL), Agent memory systems, Tool-agnostic agents

Programming & Infrastructure: Python(pydantic/typing), TypeScript, FastAPI/UviCorn, Docker, Kubernetes, PostgreSQL, Redis, WebSockets, Distributed systems, CI/CD (GitHub Actions),

EDUCATION

2018-2021 Mathematics Specialist Program Honors BSc, University of Toronto 2018–2021

Recruited into industry before degree completion based on technical achievement and open-source contributions.

Example AI Projects developed within my start-ups, Y combinator,

and self directed learning
Links in blue direct to my website and Github repositories

Agent Framework

Haive Framework A comprehensive AI agent framework enabling dynamic workflow composition and runtime modification through a robust classification, registration, and serialization system. Allowing for expansive agentic teams, swarms, memory, and allowing agents to create other agents. Built on Pydantic and Langgraph, Haive organizes AI capabilities into standardized, interchangeable components with consistent interfaces that can be composed and modified at runtime without requiring code changes.

• haive-core **Conceived and implemented an AI framework foundation with DynamicGraph builder enabling runtime workflow modification**; implemented auto-derived state schemas, node factory system, and reusable pattern registry, creating a structured, dynamic A2A-like protocol for connecting various agents. Developed intelligent state management, persistence, state schema composition and modification.

• haive-agents **Developed comprehensive agent ecosystem with dynamically modifiable architectures**. These include reasoning agents (Tree of Thoughts, LATS,), planning systems (Plan-Execute, ReWOO, LLM Compiler, Hierarchical Context Planning & Parallelizable Execution), multi-agent orchestration with Dynamic Supervisor for team coordination and emergent capabilities, RAG implementations (15+ variants with runtime component switching), structured memory pipelines with parallel knowledge graph construction for long-term retention, FLATSER pipeline (Fetch-Load-Annotate-Transform-Store-Embed-Retrieve) for document processing, and smart summarization techniques utilizing time-weighted-retrievers, Text mining taxonomy, complex data extraction, knowledge graph creation, and knowledge graph RAG for context length management.

• haive-tools **Engineered a standardized tool interface with 100+ implementations across domains** (search, finance, development); implemented dynamic tool discovery and runtime composition; created specialized toolkits for shell automation, Python code analysis, and API orchestration with consistent error handling and retry mechanisms.

• haive-prebuilt **Developed 50+ domain-specific agent implementations** leveraging the core framework, including OpenPerplexity research system with multi-source integration, Web Navigator for browser automation, Content Intelligence for text analysis, and specialized agents for contract analysis, scientific research, and disaster management with minimal configuration requirements.

• haive-games (Staidium) **Created an open agent evaluation framework with 20+ competitive environments** for benchmarking AI capabilities; supports both head-to-head competitions (Chess, Poker, Monopoly) and solo challenges with leaderboards; designed as a community platform where developers

can contribute games and individuals can participate without coding expertise; implements "King of the Hill" tournaments and specialized competitions that incentivize creation of best-in-class models for specific domains like research, creativity, or strategic reasoning; provides standardized metrics and visualization tools for comprehensive analysis of agent performance.

- haive-dataflow & haive-mcp **Created FastAPI serialization layer with middleware integration and Supabase support for persistent agent deployment**; implemented secure MCP adaptor connectors for integrating various MCPs; enables seamless API exposure of agent capabilities, stateful interaction management, and database integration for enterprise deployment scenarios.

Algebraic Wealth - Financial modeling AI Architecture

awt-quant Full-stack quantitative finance platform integrating stochastic PDE simulations (GBM, Heston, CIR, OU, MJD), portfolio optimization, risk management, multifactor analysis, macroeconomic forecasting, and volatility modeling. Supports LLM-based forecasting via TimeGPT pipelines and Lag-Llama, traditional ML time series forecasting with AutoTS and GARCH volatility models, stress testing, backtesting, and autonomous research agent pipelines.

awt-ti Developed a Python library offering 110+ technical indicators, integrated backtesting, options analytics, and strategy building tools, enabling rapid creation, testing, and optimization of trading algorithms.

awt-dcf Fundamental valuation toolkit for automating discounted cash flow (DCF) analysis, WACC computation, sensitivity and scenario analysis, and Monte Carlo simulation pipelines for probabilistic intrinsic valuation.

equity-expert Central controller for the Algebraic Wealth Technologies (AWT) finance suite, integrating awt quant, awt-ti, and awt-dcf modules to provide AI agents with unified access to stochastic simulations, portfolio optimization, technical analysis, fundamental valuation (DCF, WACC, Monte Carlo), and 800k+ macroeconomic time series spanning equities, bonds, and futures, through a FastAPI orchestration layer.

Other AI Tools and Solutions

notex-ai AI-powered tool converting handwritten notes, PDFs, and images into structured LaTeX documents. Combines OCR, LLM-based LaTeX code generation, autonomous error reflection and patching, and long-context handling to create accurate, high-quality LaTeX outputs. Adopted by math and CS students.

agent-logic Framework for symbolic logic, proof systems, and SAT-solving in LLM-based agents, using type-safe Pydantic models for integration with LangGraph

reasoning architectures.

prompt-injections

djangautomate CLI tool and Python library for generating Django apps, models, and views from SQLAlchemy databases, accelerating development and ensuring consistent codebases.

repo-sanitize CLI and Python library for cleaning Git repositories of sensitive files and secrets before sharing or open-sourcing.

credit-default Built a credit risk prediction pipeline using KNN, Logistic Regression, and XGBoost with engineered feature preprocessing and evaluation via ROC-AUC and feature importance metrics.

openchain Decentralized marketplace for AI model serving, enabling open-source LLM deployment through peer-to-peer model sharing (libtorrent) and distributed compute contribution (Ray), with Ethereum-based incentives for hosting and resource sharing (Solidity). Built to extend the Hugging Face ecosystem into a scalable, community-driven alternative to centralized AI infrastructure.

WILLIAM ASTLEY

+1 (518) 902-5805 • Toronto, ON, Canada (US & Canadian Citizen - Willing to relocate)

wrastley@gmail.com • LinkedIn • GitHub • PyPI • Website

Summary:

- Agentic Systems Engineer who independently built architectural frameworks for dynamic multi-agent AI - where agents create other agents, self-modify, self-replicate, and orchestrate emergent teams with scalable memory systems. Developed 100+ implementations from complex simulations (games, debates, social dynamics) to production systems across diverse domains.
- Breakthrough: eliminated rigid architectures through dynamic composition and A2A protocols, enabling truly adaptive AI.
- Self-taught innovator with 20+ open source frameworks spanning autonomous systems and theoretical mathematics

---

Professional Experience

2024-Present: Sole/Lead Developer of Haive

- Long term memory systems with context memory with graph DB and RAG
- Developed RAG architecture that greatly improves performance for large (gigabytes) and diverse (200+ sources) by autonomously optimizing the

preprocessing into the vector database via fetch, load, store, transform, annotated, embed, store, and retrieve processes.
- Agentic AI workflows that can self modify, replicate and evolve dynamically to secure optimal tooling and coordination with other autonomous agents.
- Proven and tested these in games, finance, social media conversations, oxford debates, and other applications.

June 2022-Present: Sr. Quantitative Risk Developer, Corpay *Toronto*

- Built financial modeling tools that utilize mathematics, algorithms, AI, & software architecture.
- Designed and implemented automated stress testing pipeline, real time dashboards, capital allocation strategies
- Designed and implemented a proprietary compliance agent-architecture leveraging dynamic RAG and reflection (without observation), accelerating onboarding by 90%. Achieved a 0% error rate and 100% case coverage
- Built GARCH/LLM forecasting models and automated stress testing pipelines, improving forecast accuracy by 35% and cutting execution time by 75%, enabling proactive risk management across a $500M portfolio and preventing tens of millions of dollars in potential losses during post-COVID volatility.
- Developed a real-time treasury dashboard integrating 10+ predictive models, informing capital allocation and liquidity strategies that contributed to 40% company growth (vs -10% industry benchmark) and supported S&P 125 inclusion

2022-2023 Founder & Lead Developer, Algebraic Wealth Technologies *Toronto, Canada*

- Founded and led a 10-person team building one of the first LLM-driven financial intelligence platforms, integrating technical analysis, economic data mining, and agent-based trading architectures.
- Advanced to Y Combinator interview stage based on technical innovation and early traction in pioneering AI agent applications.
- Developed core systems including proprietary document ingestion frameworks, real-time research generation, and sentiment-driven trading agents (details in *Projects*). [WILL HYPERLINK]

TECHNICAL SKILLS

The Agentic AI systems that I have implemented in my entrepreneurial efforts are built using the following technical capabilities.

Multi-Agent Architecture: Dynamic agent composition, Agent-to-Agent (A2A) protocols, Runtime modification, Agent self-replication, Multi-agent coordination, Emergent team orchestration, Distributed consensus

AI/ML Frameworks: LangChain, LangGraph, AutoGen, CrewAI, OpenAI Agents SDK, Hugging Face Transformers, PyTorch, TensorFlow, Vector databases (Pinecone, Weaviate), RAG architecture

Agentic AI Development: Autonomous agents, Self-modifying systems, Agent communication protocols, Multi-agent reinforcement learning (MARL), Agent memory systems, Tool-agnostic agents

Programming & Infrastructure: Python(pydantic/typing), TypeScript, FastAPI/UviCorn, Docker, Kubernetes, PostgreSQL, Redis, WebSockets, Distributed systems, CI/CD (GitHub Actions),

EDUCATION

2018-2021 Mathematics Specialist Program Honors BSc, University of Toronto 2018–2021

Recruited into industry before degree completion based on technical achievement and open-source contributions.

Example AI Projects developed within my start-ups, Y combinator, and self directed learning

Links in blue direct to my website and Github repositories

Agent Framework

Haive Framework A comprehensive AI agent framework enabling dynamic workflow composition and runtime modification through a robust classification, registration, and serialization system. Allowing for expansive agentic teams, swarms, memory, and allowing agents to create other agents. Built on Pydantic and Langgraph, Haive organizes AI capabilities into standardized, interchangeable components with consistent interfaces that can be composed and modified at runtime without requiring code changes.

- haive-core **Conceived and implemented an AI framework foundation with DynamicGraph builder enabling runtime workflow modification**; implemented auto-derived state schemas, node factory system, and reusable pattern registry, creating a structured, dynamic A2A-like protocol for connecting various agents. Developed intelligent state management, persistence, state schema composition and modification.

- haive-agents **Developed comprehensive agent ecosystem with dynamically modifiable architectures**. These include reasoning agents (Tree of Thoughts, LATS,), planning systems (Plan-Execute, ReWOO, LLM Compiler, Hierarchical Context Planning & Parallelizable Execution), multi-agent orchestration with Dynamic Supervisor for team coordination and emergent capabilities, RAG implementations (15+ variants with runtime component switching), structured memory pipelines with parallel knowledge graph construction for long-term retention, FLATSER pipeline (Fetch-Load-Annotate-Transform-Store-Embed-Retrieve) for document processing, and smart summarization techniques

utilizing time-weighted-retrievers, Text mining taxonomy, complex data extraction, knowledge graph creation, and knowledge graph RAG for context length management.

• haive-tools **Engineered a standardized tool interface with 100+ implementations across domains** (search, finance, development); implemented dynamic tool discovery and runtime composition; created specialized toolkits for shell automation, Python code analysis, and API orchestration with consistent error handling and retry mechanisms.

• haive-prebuilt **Developed 50+ domain-specific agent implementations** leveraging the core framework, including OpenPerplexity research system with multi-source integration, Web Navigator for browser automation, Content Intelligence for text analysis, and specialized agents for contract analysis, scientific research, and disaster management with minimal configuration requirements.

• haive-games (Staidium) **Created an open agent evaluation framework with 20+ competitive environments** for benchmarking AI capabilities; supports both head-to-head competitions (Chess, Poker, Monopoly) and solo challenges with leaderboards; designed as a community platform where developers can contribute games and individuals can participate without coding expertise; implements "King of the Hill" tournaments and specialized competitions that incentivize creation of best-in-class models for specific domains like research, creativity, or strategic reasoning; provides standardized metrics and visualization tools for comprehensive analysis of agent performance.

• haive-dataflow & haive-mcp **Created FastAPI serialization layer with middleware integration and Supabase support for persistent agent deployment**; implemented secure MCP adaptor connectors for integrating various MCPs; enables seamless API exposure of agent capabilities, stateful interaction management, and database integration for enterprise deployment scenarios.

Algebraic Wealth - Financial modeling AI Architecture

awt-quant Full-stack quantitative finance platform integrating stochastic PDE simulations (GBM, Heston, CIR, OU, MJD), portfolio optimization, risk management, multifactor analysis, macroeconomic forecasting, and volatility modeling. Supports LLM-based forecasting via TimeGPT pipelines and Lag-Llama, traditional ML time series forecasting with AutoTS and GARCH volatility models, stress testing, backtesting, and autonomous research agent pipelines.

awt-ti Developed a Python library offering 110+ technical indicators, integrated backtesting, options analytics, and strategy building tools, enabling rapid creation, testing, and optimization of trading algorithms.

awt-dcf Fundamental valuation toolkit for automating discounted cash flow (DCF) analysis, WACC computation, sensitivity and scenario analysis, and

Monte Carlo simulation pipelines for probabilistic intrinsic valuation.

equity-expert Central controller for the Algebraic Wealth Technologies (AWT) finance suite, integrating awt quant, awt-ti, and awt-dcf modules to provide AI agents with unified access to stochastic simulations, portfolio optimization, technical analysis, fundamental valuation (DCF, WACC, Monte Carlo), and 800k+ macroeconomic time series spanning equities, bonds, and futures, through a FastAPI orchestration layer.

Other AI Tools and Solutions

notex-ai AI-powered tool converting handwritten notes, PDFs, and images into structured LaTeX documents. Combines OCR, LLM-based LaTeX code generation, autonomous error reflection and patching, and long-context handling to create accurate, high-quality LaTeX outputs. Adopted by math and CS students.

agent-logic Framework for symbolic logic, proof systems, and SAT-solving in LLM-based agents, using type-safe Pydantic models for integration with LangGraph reasoning architectures.

prompt-injections

djangautomate CLI tool and Python library for generating Django apps, models, and views from SQLAlchemy databases, accelerating development and ensuring consistent codebases.

repo-sanitize CLI and Python library for cleaning Git repositories of sensitive files and secrets before sharing or open-sourcing.

credit-default Built a credit risk prediction pipeline using KNN, Logistic Regression, and XGBoost with engineered feature preprocessing and evaluation via ROC-AUC and feature importance metrics.

openchain Decentralized marketplace for AI model serving, enabling open-source LLM deployment through peer-to-peer model sharing (libtorrent) and distributed compute contribution (Ray), with Ethereum-based incentives for hosting and resource sharing (Solidity). Built to extend the Hugging Face ecosystem into a scalable, community-driven alternative to centralized AI infrastructure.