Malakai Spann

MalakaiSpann@gmail.com • (718) 593-1969 • MalakaiSpann.com

A full-performance, devoted software engineer focused on data processing, developing scalable, cross-platform software, and enhancing development processes through formal methods, principles, and modern technologies.

QUALIFICATIONS

Florida Institute of Technology

Bachelor of Science, Computer Science

Top Secret/Secret Compartmented Information Clearance

Polygraph & Full-Scope Background Investigation
January 2024

Certificates/Training

AWS Certified Developer Associate[1], Nov 2025

TECHNICAL EXPERIENCE

Programming

Languages: Java, Python, C/C++, Java/TypeScript, Zig, Bash

Libraries/Frameworks: Spring, JUnit, Pydantic, Numpy, Pandas, Hono, React, Tailwind, PyTest

Tools: AWS, Bun, Git, Github Actions, Gitlab CI, Docker, Jenkins, Pip, UV, Maven

Fundamentals

Strong background in the formal SDLC and Object-Oriented Programming. Also experienced with data processing, embedded systems, software security, developer operations (DevOps), and web-based technologies.

[1] Validation Number: 19c354b8ddd1482a8fb702503829aa67

PROFESSIONAL EXPERIENCE[2]

Red Alpha, Columbia, MD
Software Engineer, Data Tradecraft & Artificial Intelligence (AI), Nov 2024 – Current

- Collaborate with SMEs and technical leads to gather, clarify, and refine (non) functional requirements for web services.
- Architect and implement Python and Typescript-based REST micro services, including request/response models, middleware, and validation logic
- Design and enforce JSON-based schemas using JSON Schema, Pydantic, and Zod.
- Create comprehensive unit and integration tests using Pytest, Bun, and Jest to maintain coverage and validate application logic.
- Author internal and user-facing documentation detailing service design, implementation, and operational workflows.
- Diagnose and resolve critical issues in Java microservices using log analysis
- Implement REST controllers and handlers in Java using Spring
- Refactor Java libraries to improve modularity, testability, and extensibility
- Implement scalable indexing and retrieval workflows using NoSQL databases like Elasticsearch and Python-based SDKs
- Build automated test harnesses and validation frameworks in Python3 to support CI-driven verification
- Convert a centralized Python/pandas pipeline into a distributed PySpark workflow to increase performance and scalability
- Architect and implement multi-stage GitLab CI/CD pipelines with automated build, test, security scanning, and artifact packaging using Bash scripts and containerized environments.
- Build and deploy AWS Lambda functions for data processing using Python and pandas.

Department of Defense, Fort Meade, MD
Software Engineer, Computer Network Operations, Mar 2024 – Nov 2024

- Collaborate with senior technical leaders, team members, and external stakeholders to ensure customer satisfaction and product quality.
- Lead effort to improve automated development processes through the use of containerized environments and Jenkins CI
- Develop Python utilities to generate sanitized, compressed transfer bundles for cross-domain deployments
- Modernize legacy CMake build scripts to stabilize and streamline C++ application builds
- Execute and document comprehensive unit tests using Google Test to validate application logic

[2] Prepublication Review Case Number: RES-2025-16821

- Utilize industry standard tools to create/track, prioritize, and resolve customer issues, feature requests, and improvement work.
- Implement high-performance data extraction, transformation, and storage utilities in C++
- Develop and debug applications in Linux environments using command-line tools such as grep, tail, and ssh

Lockheed Martin, Denver, CO
Software Engineer Intern, Enterprise Flight Software, May 2022 – Dec 2023

- Develop safety-critical, configurable embedded software for spacecraft data-processing systems
- Use data-transport protocols (I2C, SRIO, MS1553-B, SpaceWire) to build embedded device communication networks
- Collaborate with internal and external teams to design, develop, test, and review C/C++ code
- Integrate Docker, GitLab CI, and related tooling into development workflows to streamline build, test, and deployment processes
- Contribute to iterative improvement of embedded software through code reviews and test-cycle participation

PROJECTS

Crab Compensation: A PySpark application that analyzes ~390,000 transactions from the Maryland state government between FY2008 and 2024 to provide insights on vendor compensation. Generates JSON-formatted data including total spending, transaction statistics, and top vendor rankings. Designed for scalable deployment and tested on AWS EMR clusters with distributed file management using Hadoop Distributed File System. This project also features automated quality control workflows and comprehensive testing.

HTML Parser: A purpose-built HTML parser written in Zig – accurate, performant, robust.

Adjustable Image Recognition Keyboard System (AirKeys): A proof-of-concept, projection-based virtual keyboard utilizing Computer Vision to detect and transmit user input to multiple devices. The project heavily focuses on image recognition, hard/software integration, embedded development, and the development process. Serves as my undergraduate capstone project. Written in Python.

- Project Pitch (Slideshow)
- Project Website (In Progress)
- Development process includes custom docker image/development environment using Visual Studio Dev Containers, custom scripts, and industry standard tools such as Sphinx and PyTest for documentation & quality assurance.

SemMed – Neo4j: A data processing workflow built to ingest, transform, and visualize large quantities of data from the National Library of Medicine's Semantic Medline Database. Written in Python, using pandas and the Neo4j SDK. Data stored and visualized using Neo4j, a graphical database.

Tic-Tac-Toe w/ AI: Uses a combination of the minmax (including an alpha-beta pruning version) algorithm and OOP concepts to create a Tic-Tac-Toe game in Python.

Traveling Salesman Problem (TSP) w/ AI: A set of solutions and test runs for the classic Traveling Salesman Computing Problem using Python and the Genetic & Ant Swarm Optimization Algorithms.

- Genetic Algorithm Version : An implementation of a genetic algorithm that explores the various decision-making methods applicable to the TSP while providing in-depth explanations of the design choices and their effects on the implementation.
- Ant Swarm Optimization Version : An implementation similar to the genetic algorithm version that uses the ant swarm optimization algorithm instead.