ContainerShip

AI-Powered Docker Optimization Platform

Uriel Buitrago & Shane Aung

Advanced Programming Tools - Summer 2025

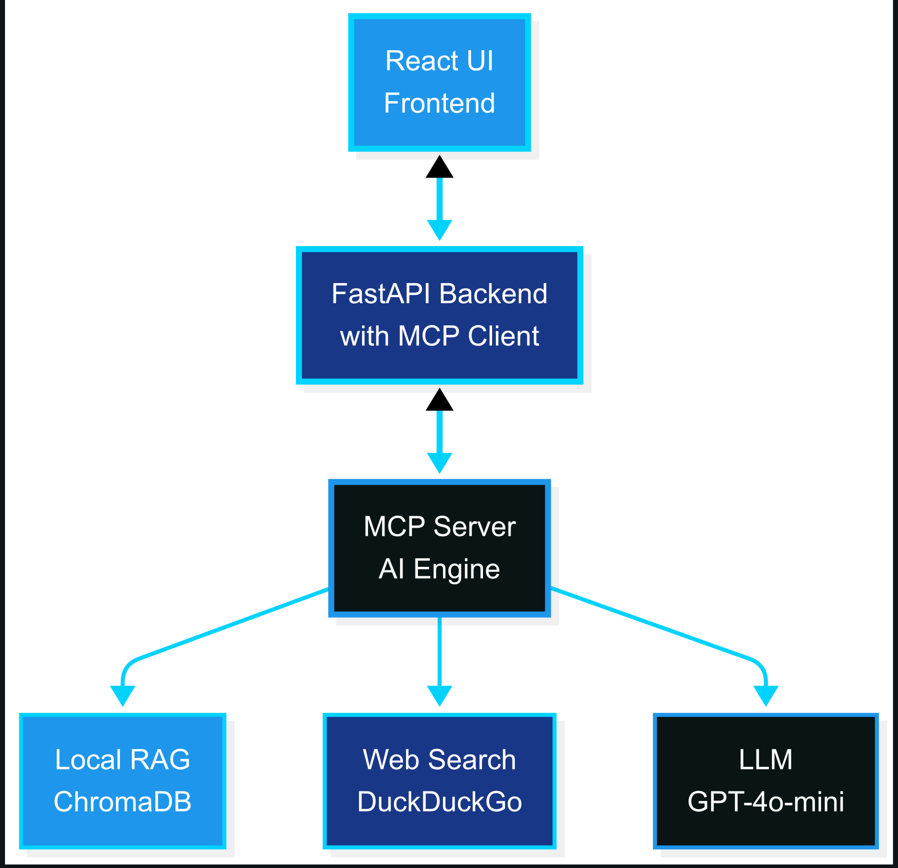
The Problem with Current Docker Optimization

- **Static analysis tools** lack contextual understanding
- **Commercial platforms** operate as "black boxes" with vendor lock-in
- **Generic AI tools** don't understand containerization specifics
- **Developers struggle** with evolving best practices
- **Security vulnerabilities** often go undetected until runtime

ContainerShip Solution Overview

- **Multi-LLM AI optimization** with OpenAI GPT & Google Gemini support
- **Enhanced hybrid knowledge**: Local docs + DuckDuckGo + Tavily intelligence
- **Integrated vulnerability scanning** for Docker images and packages
- **Technology-aware analysis** tailored to your specific stack
- **Interactive web interface** with real-time analysis & security assessment
- **Extensible MCP architecture** for continuous improvement

System Architecture



Architecture Components

Frontend: React TypeScript UI

• Real-time Dockerfile editor • Interactive analysis visualization • **Integrated vulnerability scanner**

Backend: FastAPI Server

• **Multi-LLM support** (GPT + Gemini) • MCP client for AI communication • Technology detection pipeline • Clause parsing

AI Engine: MCP Server

• Multi-tool architecture for specialized optimization • **Enhanced web search** with Tavily • Hybrid knowledge coordination

Knowledge Sources

• **ChromaDB**: Local Docker documentation (RAG) • **DuckDuckGo**: Privacy-focused web intelligence • **Tavily API**: Premium security & threat intelligence

Model Context Protocol (MCP) Integration - Core Tools

docker_docs

RAG system with comprehensive Docker documentation & ChromaDB

web_search_docker

Multi-provider intelligence: DuckDuckGo + Tavily APIs

optimize_dockerfile

Multi-layered analysis with technology-specific strategies

Model Context Protocol (MCP) Integration - Security Tools

check_security_best_practices

Enhanced vulnerability assessment with web-based threat intelligence

search_dockerfile_examples

Community-validated containerization patterns

search_security_vulnerabilities

Dedicated CVE & image vulnerability scanning

User Experience & Dockerfile Analysis Workflow

Validaton & Technology Detection

Automatic stack identification (Python Flask, Node.js, Java Spring, Go)

Vulnerability Analysis

Automated security assessment of images and packages

Interactive Results

Side-by-side comparison with **vulnerability panels** & recommendation cards

Real-time Streaming

Synchronous updates with visual progress indicators

Language Model Integration & AI Capabilities

Exploration of Different LLMs

Flexible OpenAI GPT & Google Gemini model selection
GPT-4o-mini: Higher quality recommendations & faster results
Gemini-2.5-Flash: Cost-effective with some verbosity trade-offs

Enhanced Prompt Engineering

RAG-enhanced templates prioritizing local docs + web integration

Context Management

Seamless blending of local + **multi-source** web intelligence

Technology Awareness

Framework-specific optimization strategies

Live Product Demo

Industry Impact & Results

Developer Productivity Enhancement

Intelligent automation delivering contextually relevant guidance

Security Posture Improvement

Proactive vulnerability identification with integrated CVE scanning

Cost Optimization Benefits

Systematic image size reduction & performance improvements

Knowledge Accessibility

Making **containerization & security expertise** accessible

Continuous Learning Capability

Platform recommendations remain current with ecosystem evolution

Future Possibilities & Roadmap

Extended Multi-LLM Ecosystem

Integration with **Claude, Llama**, and emerging models

CI/CD Pipeline Integration

Automated optimization **& vulnerability scanning** in development workflows

Team Collaboration Features

Shared optimization templates **& security policies** for enterprises

Conclusion

Proven Architecture

Scalable, extensible, and maintainable

Real Impact

Measurable improvements in security, performance, and productivity

Questions & Discussion

Thank you for your attention!