Mahdi Mamashli

bitsgenix@gmail.com | github.com/m-mdy-m | m-mdy-m.github.io | Iran

Summary

Backend Developer specializing in event-driven architecture and modular system design. Creator of Gland framework with hands-on experience building scalable backend applications using TypeScript/Node.js. Strong focus on performance optimization, caching strategies, and clean architectural patterns. Active open-source contributor with 15+ published technical articles on algorithms and system design.

Technical Skills

Languages: TypeScript, JavaScript, Python
Backend: Node.js, Bun, Express.js, event-driven architectures, WebSocket, pub/sub patterns
Databases: PostgreSQL, MongoDB, Redis (caching strategies, eviction policies)
System Design: Event brokers, modular architecture, CQRS patterns, dependency injection
Tools: Docker, Git, Linux/Unix, shell scripting, CI/CD fundamentals
Specializations: Performance optimization, algorithm complexity analysis, technical writing

Professional Experience

Open Source Developer **Jan 2023 – Present**
Independent Contributor *Remote*

- Designed and implemented Gland, an event-driven backend framework with broker-based architecture supporting protocol-agnostic communication (HTTP, WebSocket) and modular component isolation
- Built QIKS caching system achieving 1M+ ops/sec with O(1) complexity, implementing LRU/LFU/MRU eviction policies, cascade invalidation via dependency graphs, and TTL-based expiration
- Developed TideityIQ complexity analyzer in C, parsing JavaScript AST to calculate Big O/Theta/Omega notations for recursive algorithms
- Authored 15+ technical articles on Dev.to and Medium covering event-driven patterns, caching strategies, algorithm analysis, and backend architecture
- Contributed to open-source projects through code reviews, documentation improvements, and community support

Key Projects

Gland Framework | *TypeScript, Event-Driven Architecture* |

- Event broker system with radix tree-based routing, namespace isolation, and wildcard pattern matching for dynamic event propagation
- Dependency injection container with reflection-based metadata scanning for automatic controller/channel registration
- Protocol adapters (Express, WebSocket planned) connecting via central broker with strategy-based execution (first/all/race)
- Lifecycle hooks (onModuleInit, onAppBootstrap, etc.) for systematic initialization and graceful shutdown

QIKS Caching System | *TypeScript, Performance Engineering* |

- In-memory cache with Map/WeakMap adapters achieving microsecond-level operations through optimized data structures
- Multiple eviction policies (LRU, LFU, MRU) with hybrid expiration (TTL, idle timeout) and configurable capacity management
- Dependency graph for cascade invalidation, event system for monitoring (Set/Get/Delete/Expire), and namespace support
- Pattern matching utilities for key filtering and batch operations with functional programming interfaces

TideityIQ | *C, Compiler Design* |

- CLI tool parsing JavaScript source files to analyze recursive function complexity using AST traversal
- Implemented pattern recognition for divide-and-conquer, linear recursion, and branching recursion with recurrence relation solving

ARLIZ Book | *LaTeX, Technical Writing* |

- Comprehensive guide to data structures and algorithms covering historical context, mathematical foundations, and practical implementations
- Part I explores computing history from ancient counting to modern systems; Part II covers mathematical prerequisites for algorithm analysis

Additional Projects

- **AGAS**: Bun-powered HTTP client with event-driven request/response lifecycle and CLI interface
- **Cop**: Telegram bot for automated group moderation using Grammy framework and PostgreSQL
- **NExa**: Python CLI tool fetching and displaying dev.to articles with Docker deployment support

Education

Azad University **Sep 2022 – Present**
Bachelor of Science in Computer Engineering *Iran*

Recognition

Top 10 GitHub Contributor in Iran
15+ Published Technical Articles on Dev.to and Medium
Framework Creator – Gland framework with modular event-driven architecture