Li Yongchun

Male | Chinese | Position: Backend Developer | +61 0450789126 | +86 18351977789 |
leeyc.official@gmail.com | github.com/LiWinston | bitsleep.cn | WeChat: ycLee_Official

Education

The University of Melbourne *Master of Software Engineering (Distributed Systems)* *December 2025*
DB, Java, Algorithms, Process Management, Operating Systems, Requirements Engineering, Distributed Systems, Testing, Web Design Patterns, Cluster & Cloud Computing, Automata, etc.

Southeast University *Bachelor of Automation (Intelligent Information Processing)* 985 & 211 *July 2022*
C, C++, Data Structures, Microcomputer Systems and Interfaces, Linux Embedded Systems, Digital Image Processing, Pattern Recognition, Introduction to Artificial Intelligence, etc.

Work Experience

China Railway Construction Bridge Engineering Group - Digital Platform | OA Workflow Subsystem
Innovation Research Institute | Software Division *June 2024 - August 2024*
A project management system based on SpringCloud(Alibaba), Flowable, TTL, and Ruoyi-Vue, transforming the original monolithic project into a service-oriented architecture. Supports project customization, publishing, and task assignment; enables custom workflows with enterprise WeChat integration. Managers can monitor project/task progress through real-time ECharts dashboards, achieving streamlined and intelligent project management.

- Implemented custom SpringCloud Gateway global filters for unified authentication and API timing metrics; used custom annotations + AOP for service-level authentication and internal certification. Utilized Skywalking for performance monitoring and distributed tracing, integrated with DingTalk and email notifications for complete system and business monitoring loops
- Implemented Redis + Lua counter-based rate limiting for whitelist APIs (login, etc.), with optional IP-based limiting to effectively prevent API abuse
- Used RocketMQ for project and task approval workflow messages, achieving asynchronous decoupling and improving system processing efficiency by 20%
- Implemented custom fallback service degradation with OpenFeign + Sentinel to reduce service load; integrated Sentinel with Gateway for rate limiting to prevent service overload while ensuring availability
- Created annotation-based distributed locks using Redisson with dynamic SpEL lock identifiers to ensure workflow concurrent update consistency
- Adopted Seata AT mode for distributed transactions, ensuring task approval status consistency and preventing cross-service transaction inconsistencies
- Implemented Cache Aside pattern for timely cache invalidation after database updates, maintaining data consistency between database and cache
- Developed custom request header interceptor using TransmittableThreadLocal, encapsulating user data in thread variables to reduce database queries by 40% with automatic session refresh; preserved request headers in Feign calls to maintain call chain state
- Used MySQL-persisted Nacos as registration and configuration center to prevent config loss on restart; simplified deployment with Docker Compose

Personal Projects

HuskyAI - Multi-Model AI Chatbot Full Stack - Unimelb Client Project
Based on Spring Boot and React JS, using Mybatis/PostgreSQL (user information and conversation summaries), MongoDB (conversation history, conversation sharing, model access permissions, login locations) and Redis caching. Supports synchronous/WebFlux streaming conversations and user-level message preference settings.

- Implemented keyword full-text search, history lazy loading and search result pagination using PageHelper; JWT + TTL for interface authentication and auto-login, storing uuid and page context to support cache async double deletion and multi-thread context propagation
- Used Spring Cache annotation-based caching to reduce database access frequency for 30+ hotspot queries, implemented interface anti-shake for repeated queries; reduced complex join query time from 1.9s to 0.1s for relatively static data like conversation-user relationships
- AOP-based remote login reporting, model fault failover, conversation summary generation (dual interception points + model avoidance + CompletableFutureMap for parallel summaries, stream can carry titles midway), follow-up query prediction generation and SSE pushing
- Implemented timestamp-based NoSQL synchronization and recovery, ensuring messages are ordered, with correct parity, no duplication or omission; lightweight message queue based on Redis ZSet+Hash for reliable

task delivery and tracking, supporting failure retry and progress recovery; triple synchronization with MQ push + incremental distance + scheduled two-way sync ensuring both performance and completeness
- Context limiter based on token decomposition and slow-start progressive cache reading, efficiently trimming conversations while reducing cache pressure and API overhead
- Implemented time-limited email confirmation and index-based conversation sharing using Redis/Mongo TTL, supporting extension/cancellation of share links
- Developed weighted evaluation-based title/follow-up model selector (success rate/time cost/random factor/usage balance), can disable underperforming models
- Admin console for managing models, user usage limits, adjusting prompts and strategies; multi-level admin permission annotations with time-period exemptions
- Implemented dual-layer rate limiting with Sentinel+Redis/Lua scripts, ensuring balanced QPS and resource usage control with stackable IP/user/resource limitations
- Integrated embedding vector similarity detection + user fact summaries + clustering for deduplication, implementing memory generation/retrieval/knowledge networks, enhancing cross-conversation associations; integrated LangChain4j, researched custom LangSearch engine adaptation, utilizing AiService structured output for search keyword optimization
- Developed AI-powered resume optimization system using OCR recognition for PDF/image parsing, JsonSchema for structured data extraction, semantic search and RAG for tailored improvement suggestions; created interview coaching and simulation tools based on user application history analysis; built job recommendation system using Flink+Kafka+ElasticSearch pipeline with Selenium+WebClient for data crawling, real-time processing, and profile-based matching

Multi-user Whiteboard Based on gRPC (Protobuf) and Java Swing Unimelb Course Design

Supports real-time collaborative stroke/text/graphic drawing/text insertion/erasure, includes chat rooms, online lists, real-time editor indicators.
- Used gRPC streaming calls for instant preview of strokes/draggable graphics, used thread-safe collections to store temporary graphics and user states, ensuring multi-thread safety. Implemented high-performance two-phase graphic overlap detection algorithm based on Rectangle2D and Area, rejecting later conflicting drawings through concurrent drawing overlap conflict detection
- Used Sentinel for broadcast rate limiting to implement preview frame rate control; integrated JWT for identity verification, ensuring privileged operation security

BitSleep - Personal Homepage ReactJS/i18n/Tailwind|NodeJs|SpringBoot/Redis/Mongo/Pgsql

Implemented weather forecasts/coordinate navigation/email forwarding/instant-effective WebSocket banning and unbanning (IP/fingerprint/batch)/real-time announcements/file S3 upload-download, preview, resume hot updates - file eTag cache validity judgment ensures client-side updates while saving traffic, timeliness. Sa-Token-based single sign-on system for sub-services, supporting password, email verification code, and Oauth2 third-party login.

Midnight Chapter - Unity3D C# Swordsmanship Game Prototype Lead Programmer | System Design | Technical Optimization
- **Polymorphic Projectiles**: Designed skill system supporting thrown and fixed-point casting modes through strategy pattern for polymorphic effects (explosion, continuous, ricochet); implemented landing point preview and validity verification with ray detection and LineRenderer; supports fan-shaped multi-emission skills that scale with level; optimized resource management and damage growth with object pooling and curve editors
- **Status Effects**: Designed interface-based status framework (IFreezable etc.) implementing complex states like freezing, meditation, and Golden Bell Shield; managed status duration and synchronization via coroutines and event systems; supports dynamic status bar stacking with gradient visual feedback
- **AI Intelligence**: Implemented NavMesh pathfinding with state machines for multi-mode AI behavior (pursuit/attack/wandering); handled dynamic obstacles through ray detection; supports remote tracking and group coordination; implemented specialized Boss skills and behavior patterns
- **Combat System**: Created collision-based hit detection supporting combos, critical hits, and blocking mechanics; decoupled damage calculation and effect lifecycle through event systems; achieved high cohesion and low coupling in unit interactions
- **Performance Optimization**: Managed monster/item spawning and projectile/effect creation/destruction through custom object pools; processed continuous effects via coroutines; implemented Ray-Tracing based dynamic camera switching and occlusion optimization ensuring stable 100+ FPS in WebGL and high refresh rates on desktop
- **Development Tools**: Created curve-based universal value component supporting non-linear configuration of skill intensity/leveling/experience, accelerating game design workflow

Other

- Java Pac-Man, Poker Games, K-V Dictionary Client, Servlet Club Management Program, C Language IMAP Agent and other course projects
- IEEE Computer Mouse Maze Competition: Second Prize at University Level, responsible for PID control parameter optimization
- Design of an Oceanic Organism Detection Algorithm Based on Lightweight YOLO BEng Thesis
  Improved lightweight YOLOv5 by introducing attention mechanisms/optimizing Neck fusion/utilizing Transformers and deformable convolutions, enhancing detection performance for multi-scale targets under varying viewpoints and specific target detection rates, achieving 2x inference speed improvement on MHz-level devices with accuracy decrease of less than 0.1.

Skills

- Proficient with GitHub Desktop and Git command line; familiar with Maven dependency management and build tools; experienced with Gradle
- Familiar with JUnit testing, Jmeter load testing, Swagger API documentation, Postman API testing, and Jenkins CI/CD pipelines
- Solid computer science fundamentals with strong knowledge of data structures and core network protocols (IP, UDP, TCP, HTTPS)
- Proficient in Java, object-oriented programming, and practical application of design patterns (Singleton, Factory, Strategy, etc.)
- Mastered Spring Boot, Spring Framework, and MyBatis-Plus; understanding of auto-configuration principles, IoC container, and AOP
- Strong concurrent programming skills with experience using thread pools, locks, and concurrent utilities (synchronized, CountDownLatch, AtomicReference); understanding of container implementations like HashMap and CopyOnWriteArrayList
- Proficient with Redis data structures, replication, Sentinel mode, and persistence strategies (AOF, RDB); awareness of cache-related challenges including penetration, breakdown, and avalanche; familiar with MongoDB document storage, indexing, and aggregation queries
- Solid understanding of SQL databases including indexing, transactions, locking mechanisms, storage engines, and MVCC
- Basic knowledge of C/C++, STL, GDB, Makefile, and Linux development environment
- Proficient with Docker multi-stage builds, containerized deployment, and Dockerfile optimization
- Familiar with HTML/CSS/JavaScript basics and traditional Servlet/JSP development with JDBC
- Experience with Matlab, Python, TensorFlow, and PyTorch for modeling and research; comfortable with Anaconda environment setup
- Strong English proficiency (TOEFL, CET-6); skilled at technical documentation research; proficient with AI tools like GPT and GitHub Copilot to enhance development efficiency
- Self-motivated problem solver with strong communication skills and consistent code contributions in team projects
- **Available for internship: June 2025 - August 2025**