

KHALA v2.0: COMPLETE IMPLEMENTATION DOCUMENTATION

Master Index & Navigation Guide

Project: KHALA (Knowledge Hierarchical Adaptive Long-term Agent)

Version: 2.0

Framework: Agno + SurrealDB

Date: November 2025

Total Strategies: 57 (22 Core + 35 Advanced)

DOCUMENTATION SUITE OVERVIEW

This is the **master index** for the complete KHALA v2.0 implementation documentation. All documentation follows a structured, numbered format for easy navigation and reference.

Document Structure

The documentation is organized into **12 primary documents**, numbered sequentially:

1. [01-plan.md](#) - Project planning and strategy
2. [02-tasks.md](#) - Complete task breakdown (350+ tasks)
3. [03-architecture.md](#) - Technical architecture
4. [04-database.md](#) - Database schema and design
5. [05-api.md](#) - API specifications
6. [06-deployment.md](#) - Deployment guide
7. [07-testing.md](#) - Testing strategy
8. [08-monitoring.md](#) - Monitoring and observability
9. [09-security.md](#) - Security architecture
10. [10-troubleshooting.md](#) - Troubleshooting guide
11. [11-contributing.md](#) - Contributing guidelines
12. [12-roadmap.md](#) - Roadmap and future plans

Total Pages: ~180 pages of comprehensive documentation

QUICK START

New Team Member Onboarding

1. **Start Here:** Read this index (5 min)
2. **Understand Vision:** Read [01-plan.md](#) (30 min)
3. **Review Architecture:** Read [03-architecture.md](#) (1 hour)
4. **Setup Environment:** Follow [01-plan.md](#) "Resources & Dependencies" + [06-deployment.md](#) (2 hours)
5. **First Task:** Select from [02-tasks.md](#) Module 01 (start coding!)

Developer Quick Reference

- **Finding a Task:** See [02-tasks.md](#), search by module/priority
- **Architecture Question:** See [03-architecture.md](#)
- **Database Query:** See [04-database.md](#)
- **API Usage:** See [05-api.md](#)
- **Deployment Issue:** See [06-deployment.md](#) or [10-troubleshooting.md](#)
- **Test Guidance:** See [07-testing.md](#)

Manager/Stakeholder Quick Reference

- **Project Overview:** [01-plan.md](#) "Executive Summary"
- **Progress Tracking:** [02-tasks.md](#) "Task Status"
- **Metrics & KPIs:** [01-plan.md](#) "Success Metrics"
- **Roadmap:** [12-roadmap.md](#)
- **Risk Management:** [01-plan.md](#) "Risk Management"

DOCUMENT SUMMARIES

[01-plan.md](#) (Project Plan)

Purpose: High-level project planning and strategy

Audience: All team members, stakeholders

Length: ~40 pages

Status: ✓ Complete

Contents:

- Executive Summary (vision, objectives, deliverables)
- Project Scope (all 57 strategies listed)
- Technical Architecture (stack selection, system layers)

- Implementation Strategy (10 modules, DDD approach)
- Resources & Dependencies (team, infrastructure, APIs)
- Success Metrics (performance, quality, cost, reliability)
- Risk Management (7 major risks with mitigation)
- Quality Assurance (testing strategy, code standards)
- Deployment Strategy (dev/staging/production models)
- Documentation Requirements (14 doc types)

Key Sections:

- Section 2: Complete list of all 57 strategies with descriptions
- Section 4: 10-module implementation breakdown with effort estimates
- Section 6: Detailed success metrics and targets
- Section 7: Risk management with contingency plans

02-tasks.md (Task Breakdown)

Purpose: Complete task list for implementation

Audience: Developers, project managers

Length: ~50 pages

Status: ✓ Complete

Contents:

- Task Organization System (numbering, priorities, statuses)
- Module 01-10: Detailed task breakdowns (25-40 tasks per module)
- Deployment Tasks (15 tasks)
- Documentation Tasks (25 tasks)
- Testing Tasks (18 tasks)
- Task Dependencies (critical path, parallel streams)
- Task Tracking Guide

Total Tasks: 350+

Task Format:

```

**M{module}.{category}.{task}** [Priority] Description
- Detailed requirements
- **Deliverable**: What to produce
- **Reference**: Links to docs/code
- **Expected Impact**: Quantified benefit
- **Status**: TODO/IN_PROGRESS/DONE

```

Key Sections:

- Modules 01-05: Core implementation (critical path)
- Modules 06-07: Cost optimization + quality assurance (high ROI)
- Modules 09-10: Production features + advanced capabilities
- Section "Task Dependencies": Critical path visualization

03-architecture.md **(Technical Architecture)**

Purpose: Detailed technical architecture and design

Audience: Senior developers, architects

Length: ~30 pages

Status: To be created

Planned Contents:

1. System Overview

- High-level architecture diagram
- Component interaction map
- Data flow end-to-end

2. Technology Stack Details

- Agno framework integration patterns
- SurrealDB configuration (WebSocket, namespaces, RBAC)
- Gemini API usage (all 3 models)
- Redis architecture (L2 cache)
- GPU setup (CUDA + ONNX)

3. Module Architecture (10 modules)

- Per-module component diagrams
- Class diagrams
- Sequence diagrams for key workflows
- API contracts between modules

4. Data Flow Diagrams

- Memory storage flow (verification → embedding → storage)
- Search flow (query → intent → hybrid search → ranking)
- Consolidation flow (decay → merge → deduplicate → archive)
- Multi-agent flow (debate → consensus → decision)

5. Component Interactions

- Domain layer (pure business logic)
- Infrastructure layer (technical implementations)
- Application layer (services, orchestration)

- Interface layer (CLI, MCP, API)

6. API Specifications

- Internal APIs (Python classes)
- External APIs (REST, MCP, WebSocket)
- Event contracts (LIVE subscriptions)

7. Security Architecture

- Authentication flow
- Authorization (RBAC model)
- Data encryption (at-rest, in-transit)
- API security (rate limiting, validation)

8. Performance Architecture

- Caching strategy (L1/L2/L3)
- Indexing strategy (HNSW, BM25, multi-index)
- Query optimization
- GPU acceleration

9. Scalability Design

- Horizontal scaling (distributed consolidation)
- Vertical scaling (GPU nodes)
- Database sharding (if needed)
- Load balancing

04-database.md (Database Schema)

Purpose: Complete database schema and design

Audience: Database administrators, backend developers

Length: ~25 pages

Status: To be created

Planned Contents:

1. SurrealDB Schema Complete

- Namespace and database definitions
- Table definitions (20+ tables)
- Relationship definitions (graph edges)
- Permissions and RBAC

2. Core Tables

- `memory` table (primary memory storage)
 - Fields: `user_id`, `content`, `embedding`, `tier`, `importance`, etc.

- Indexes: HNSW vector, BM25 full-text, user_id, tier, etc.
- entity table (extracted entities)
- relationship table (graph edges)
- skill table (skill library)
- audit_log table (compliance)
- multimodal_memory table (images/tables/code)
- cost_tracking table (LLM costs)
- debate_consensus table (agent debate results)

3. All Indexes (50+ indexes)

- Vector indexes (HNSW)
- Full-text indexes (BM25)
- B-tree indexes (user_id, tier, created_at)
- Composite indexes (hot path optimization)

4. Custom Functions (10+ functions)

- fn::decay_score(age_days, half_life) - Exponential decay
- fn::should_promote(age, access, importance) - Promotion logic
- fn::similarity_threshold(embedding1, embedding2, threshold) - Vector similarity
- fn::content_hash(content) - SHA256 hashing
- fn::days_ago(days) - Date math

5. ER Diagrams

- Entity-Relationship diagrams
- Graph structure visualization
- Memory tier flow diagram

6. Query Examples (100+ queries)

- Basic CRUD operations
- Vector similarity search
- BM25 full-text search
- Hybrid queries (vector + BM25 + metadata)
- Graph traversal (multi-hop)
- Aggregations and analytics
- Temporal queries (decay scoring)

7. Performance Optimization

- Index selection guide
- Query optimization tips
- Explain plan analysis

- Caching strategies

8. Backup Strategy

- Daily incremental backups
- Weekly full backups
- Point-in-time recovery
- Disaster recovery procedures

05-api.md (API Specifications)

Purpose: Complete API documentation

Audience: Frontend developers, integration engineers

Length: ~20 pages

Status: To be created

Planned Contents:

1. REST API Endpoints

- `POST /memory/store` - Store new memory
- `GET /memory/retrieve` - Retrieve similar memories
- `POST /memory/consolidate` - Trigger consolidation
- `GET /memory/context` - Get assembled context
- `GET /health` - Health check
- `GET /metrics` - Prometheus metrics

2. MCP Tools Documentation

- `store_memory(content, tags, importance)` - Store memory via MCP
- `retrieve_memory(query, top_k, min_relevance)` - Retrieve via MCP
- `search_graph(entity, depth, relation_types)` - Graph traversal
- `consolidate(user_id)` - Manual consolidation trigger
- `get_context(query, max_tokens)` - Context assembly

3. WebSocket API

- LIVE subscription protocol
- Event types (`memory_created`, `memory_promoted`, etc.)
- Real-time updates

4. Authentication/Authorization

- API key authentication
- JWT token support
- RBAC permissions
- Namespace isolation

5. Request/Response Examples

- Complete examples for each endpoint
- Success responses
- Error responses

6. Error Handling

- Error code reference (4xx, 5xx)
- Error message formats
- Retry logic guidance

7. Rate Limiting

- Rate limits per endpoint
- Burst allowances
- Throttling strategies

8. Versioning

- API versioning strategy (/v1/, /v2/)
- Backward compatibility
- Deprecation policy

06-deployment.md (Deployment Guide)

Purpose: Complete deployment instructions

Audience: DevOps engineers, system administrators

Length: ~15 pages

Status: To be created

Planned Contents:

1. Deployment Architecture

- Development environment
- Staging environment
- Production environment
- High-availability setup

2. Environment Configuration

- Environment variables (.env)
- Configuration files (config.yaml)
- Secrets management

3. Docker Setup

- Dockerfile for application
- docker-compose.yml (SurrealDB + Redis + App)

- Container orchestration

4. **Kubernetes Configuration** (optional)

- Kubernetes manifests
- Helm charts
- Service mesh integration

5. **CI/CD Pipeline**

- GitHub Actions workflow
- Build pipeline
- Test automation
- Deployment automation

6. **Monitoring Setup**

- Prometheus deployment
- Grafana deployment
- Alerting configuration
- Log aggregation (ELK stack)

7. **Backup & Recovery**

- Backup procedures
- Restore procedures
- Testing backup validity

8. **Disaster Recovery**

- RTO/RPO targets
- Failover procedures
- Incident response plan

07-testing.md **(Testing Strategy)**

Purpose: Testing methodology and examples

Audience: QA engineers, developers

Length: ~12 pages

Status: To be created

Planned Contents:

1. **Testing Strategy**

- Testing pyramid (unit → integration → E2E)
- Test coverage targets (>80%)
- Continuous testing in CI/CD

2. **Unit Test Examples**

- pytest setup
- Mock external dependencies
- Property-based testing
- Example test cases

3. Integration Test Examples

- Database integration tests
- API integration tests
- Multi-module integration

4. Load Testing Methodology

- Load testing tools (Locust, k6)
- Performance benchmarks
- Scalability testing

5. Performance Benchmarks

- Baseline metrics
- Target metrics
- Regression detection

6. Test Coverage Requirements

- Coverage tools (pytest-cov)
- Coverage reports
- Coverage thresholds

7. CI/CD Integration

- Automated test execution
- Test result reporting
- Quality gates

8. Quality Gates

- Pre-merge checks
- Pre-deployment checks
- Production validation

08-monitoring.md (Monitoring Guide)

Purpose: Monitoring and observability setup

Audience: SRE engineers, operations team

Length: ~10 pages

Status: To be created

Planned Contents:

1. Monitoring Architecture

- Metrics collection
- Log aggregation
- Distributed tracing
- Alerting

2. Prometheus Metrics

- Application metrics
- Infrastructure metrics
- Custom metrics
- Metric naming conventions

3. Grafana Dashboards

- System overview dashboard
- Performance dashboard
- Cost dashboard
- Error dashboard

4. Alerting Rules

- Critical alerts (pager)
- Warning alerts (email/Slack)
- Alert escalation
- On-call rotation

5. Log Aggregation

- Log format standards
- Log levels
- Log retention
- Log analysis

6. Performance Monitoring

- Latency tracking (p50, p95, p99)
- Throughput tracking
- Resource utilization
- Bottleneck detection

7. Cost Tracking

- LLM cost tracking
- Infrastructure cost
- Cost optimization alerts

8. SLA Tracking

- Uptime tracking
- SLA compliance
- SLI/SLO definitions

09-security.md (**Security Guide**)

Purpose: Security architecture and best practices

Audience: Security engineers, compliance team

Length: ~12 pages

Status: To be created

Planned Contents:

1. Security Architecture

- Defense in depth
- Zero trust principles
- Security boundaries

2. Authentication Methods

- API key authentication
- JWT tokens
- OAuth 2.0 support
- Multi-factor authentication

3. Authorization (RBAC)

- Role definitions
- Permission model
- Namespace isolation
- Row-level security

4. Data Encryption

- Encryption at rest (database)
- Encryption in transit (TLS)
- Key management

5. API Security

- Input validation
- SQL injection prevention
- Rate limiting
- CORS configuration

6. Network Security

- Firewall rules

- VPN access
- DMZ setup

7. Compliance

- GDPR compliance (data deletion, portability)
- SOC 2 compliance
- HIPAA considerations
- Audit logging

8. Security Best Practices

- Secure coding guidelines
- Dependency scanning
- Vulnerability management
- Security testing

10-troubleshooting.md (Troubleshooting Guide)

Purpose: Common issues and solutions

Audience: Support team, developers, operations

Length: ~15 pages

Status: To be created

Planned Contents:

1. Common Issues

- Connection errors
- Performance degradation
- Memory leaks
- Database issues

2. Debug Procedures

- Enabling debug logging
- Reading stack traces
- Using debuggers
- Profiling tools

3. Performance Tuning

- Query optimization
- Index optimization
- Cache tuning
- Resource allocation

4. Log Analysis

- Finding errors in logs
- Correlating events
- Log search patterns

5. Database Troubleshooting

- Connection pool exhaustion
- Slow queries
- Index issues
- Replication lag

6. Network Issues

- WebSocket disconnections
- Timeout errors
- DNS issues

7. Memory Issues

- Out of memory errors
- Memory leak detection
- Garbage collection tuning

8. FAQ

- Frequently asked questions
- Quick reference
- Known limitations

11-contributing.md (Contributing Guide)

Purpose: Contributing guidelines for open source

Audience: External contributors, community

Length: ~8 pages

Status: To be created

Planned Contents:

1. Contributing Guidelines

- Code of conduct
- How to contribute
- Issue reporting
- Feature requests

2. Code Standards (PEP 8)

- Python style guide
- Naming conventions

- Documentation standards
- Type hints

3. **Git Workflow**

- Branching strategy
- Commit message format
- Git best practices

4. **Pull Request Process**

- PR template
- Review process
- Merge criteria

5. **Code Review Guidelines**

- Review checklist
- Feedback etiquette
- Approval process

6. **Testing Requirements**

- Test coverage requirements
- Running tests locally
- CI/CD checks

7. **Documentation Requirements**

- When to update docs
- Doc formats
- Doc review

8. **Community Guidelines**

- Communication channels
- Getting help
- Recognition

12-roadmap.md **(Roadmap)**

Purpose: Future plans and version history

Audience: All stakeholders

Length: ~6 pages

Status: To be created

Planned Contents:

1. **Version History**

- Version 1.0 (baseline)

- Version 2.0 (current - 57 strategies)

2. **Current Version (2.0)**

- Features included
- Known issues
- Release notes

3. **Planned Features**

- Version 2.1 (minor improvements)
- Version 2.2 (additional optimizations)
- Version 3.0 (major enhancements)

4. **Community Requests**

- Top requested features
- Feature voting

5. **Research Directions**

- Cutting-edge research integration
- Experimental features

6. **Breaking Changes**

- Planned breaking changes
- Migration guides

7. **Deprecation Plan**

- Deprecated features
- Timeline for removal

8. **Long-term Vision**

- 5-year vision
- Strategic direction

NAVIGATION GUIDE

By Role

Developers:

1. Start: [01-plan.md](#) (understand vision)
2. Tasks: [02-tasks.md](#) (find your module)
3. Architecture: [03-architecture.md](#) (understand design)
4. Database: [04-database.md](#) (understand schema)
5. Testing: [07-testing.md](#) (write tests)

DevOps/SRE:

1. Deployment: [06-deployment.md](#) (deploy the system)
2. Monitoring: [08-monitoring.md](#) (set up observability)
3. Troubleshooting: [10-troubleshooting.md](#) (solve issues)
4. Security: [09-security.md](#) (secure the system)

Managers:

1. Plan: [01-plan.md](#) (overview, metrics, risks)
2. Tasks: [02-tasks.md](#) (track progress)
3. Roadmap: [12-roadmap.md](#) (future plans)

Contributors:

1. Contributing: [11-contributing.md](#) (how to contribute)
2. Architecture: [03-architecture.md](#) (understand design)
3. Tasks: [02-tasks.md](#) (find good first issues)

By Topic

Setup & Installation:

- [01-plan.md](#) "Resources & Dependencies"
- [06-deployment.md](#) "Environment Configuration"

Architecture & Design:

- [03-architecture.md](#) (complete architecture)
- [04-database.md](#) (database design)

Development:

- [02-tasks.md](#) (what to build)
- [03-architecture.md](#) (how to build)
- [11-contributing.md](#) (code standards)

Testing & Quality:

- [07-testing.md](#) (testing strategy)
- [01-plan.md](#) "Quality Assurance"

Operations:

- [06-deployment.md](#) (how to deploy)
- [08-monitoring.md](#) (how to monitor)
- [10-troubleshooting.md](#) (how to fix)

Security & Compliance:

- [09-security.md](#) (security architecture)
- [04-database.md](#) "RBAC"

Future Planning:

- [12-roadmap.md](#) (what's next)
- [01-plan.md](#) "Success Metrics"

QUICK REFERENCE

Key Metrics (from [01-plan.md](#))

Performance Targets:

- Search latency p95: <100ms
- Embedding generation: >1000/sec
- Memory precision@5: >90%
- Cache hit rate: >70%

Quality Targets:

- Verification pass rate: >70%
- Deduplication accuracy: >90%
- Entity extraction accuracy: >85%

Cost Targets:

- LLM cost per memory: <\$0.03
- Monthly LLM cost: <\$500 (1M memories)

Reliability Targets:

- System uptime: >99.95%
- Database availability: >99.99%

Module Summary (from [01-plan.md](#))

- **Module 01:** Foundation (SurrealDB, schemas)
- **Module 02:** Search System (hybrid search, intent)
- **Module 03:** Memory Lifecycle (3-tier, consolidation)
- **Module 04:** Processing & Analysis (entities, skills)
- **Module 05:** Integration & Coordination (MCP, multi-agent)
- **Module 06:** Cost Optimization (LLM cascading)
- **Module 07:** Quality Assurance (verification, debate)
- **Module 08:** Advanced Search (multi-index, patterns)

- **Module 09:** Production Features (audit, distributed, GPU)
- **Module 10:** Advanced Capabilities (multimodal, dashboards)

All 57 Strategies (from 01-plan.md)

Core (22):

1-5: Storage & Indexing
6-8: Search & Retrieval
9-12: Memory Management
13-17: Processing & Analysis
18-22: Integration

Advanced (35):

23-25: Cost Optimization
26-28: Quality Assurance
29-30: Search Enhancement
31-37: Memory & Search Optimization
39-48: Production Features
49-57: Advanced Capabilities

Tech Stack (from 01-plan.md)

- **Agent Framework:** Agno
- **Database:** SurrealDB 2.0+
- **Embedding:** gemini-embedding-001 (768d)
- **LLM Fast:** gemini-1.5-flash
- **LLM Medium:** gpt-4o-mini
- **LLM Smart:** gemini-2.5-pro
- **Cache:** Redis 7+
- **Language:** Python 3.11+

DOCUMENT STATUS

Completed ✓

- 01-plan.md (40 pages)
- 02-tasks.md (50 pages)
- This index document

In Progress ☐

- None (awaiting team assignment)

Planned ☐

- [03-architecture.md](#) (30 pages)
- [04-database.md](#) (25 pages)
- [05-api.md](#) (20 pages)
- [06-deployment.md](#) (15 pages)
- [07-testing.md](#) (12 pages)
- [08-monitoring.md](#) (10 pages)
- [09-security.md](#) (12 pages)
- [10-troubleshooting.md](#) (15 pages)
- [11-contributing.md](#) (8 pages)
- [12-roadmap.md](#) (6 pages)

Total Planned: 153 pages

NEXT STEPS

1. **Review this index** to understand the documentation structure
2. **Read [01-plan.md](#)** to understand the project vision and strategy
3. **Read [02-tasks.md](#)** to understand the complete task breakdown
4. **Create remaining documents** (03-12) following the outlined structure
5. **Begin implementation** starting with Module 01 tasks

SUPPORT & FEEDBACK

Questions? See [10-troubleshooting.md](#) (when complete) or create an issue

Suggestions? See [11-contributing.md](#) (when complete) or submit a PR

Roadmap? See [12-roadmap.md](#) (when complete)

Status: Documentation Suite v2.0 Ready

Last Updated: November 2025

Maintained By: KHALA Development Team

☐ **Ready to build the world's best agent memory system!**