

Shadow AI - The Enterprise Architect

Complete Agent Capabilities Reference Guide

Version: 13.0.0

Total Services: 480

Build Status: Passing

Last Updated: December 17, 2025

Table of Contents

- [Executive Summary](#)
- [Architecture Overview](#)
- [Core Capabilities](#)
- [Service Categories](#)
- [AI/ML Capabilities](#)
- [DevOps & Infrastructure](#)
- [Enterprise Features](#)
- [Development Tools](#)
- [Security Features](#)
- [Areas for Development](#)
- [Recommendations](#)

Executive Summary

Shadow AI is a comprehensive autonomous coding agent with **480 specialized services** designed to assist developers across the entire software development lifecycle. The agent operates as an Electron desktop application with a React frontend and Node.js backend, integrating with multiple AI providers including OpenAI, Anthropic, Google, and local models via Ollama.

Key Differentiators

- Multi-Model Support:** Seamless switching between AI providers
- Autonomous Execution:** Self-planning, self-debugging, self-healing capabilities
- Full-Stack Coverage:** Frontend, backend, DevOps, AI/ML, security, and business intelligence
- Enterprise-Ready:** Authentication, payments, analytics, multi-tenancy support
- Self-Improving:** Recursive self-improvement and evolution capabilities

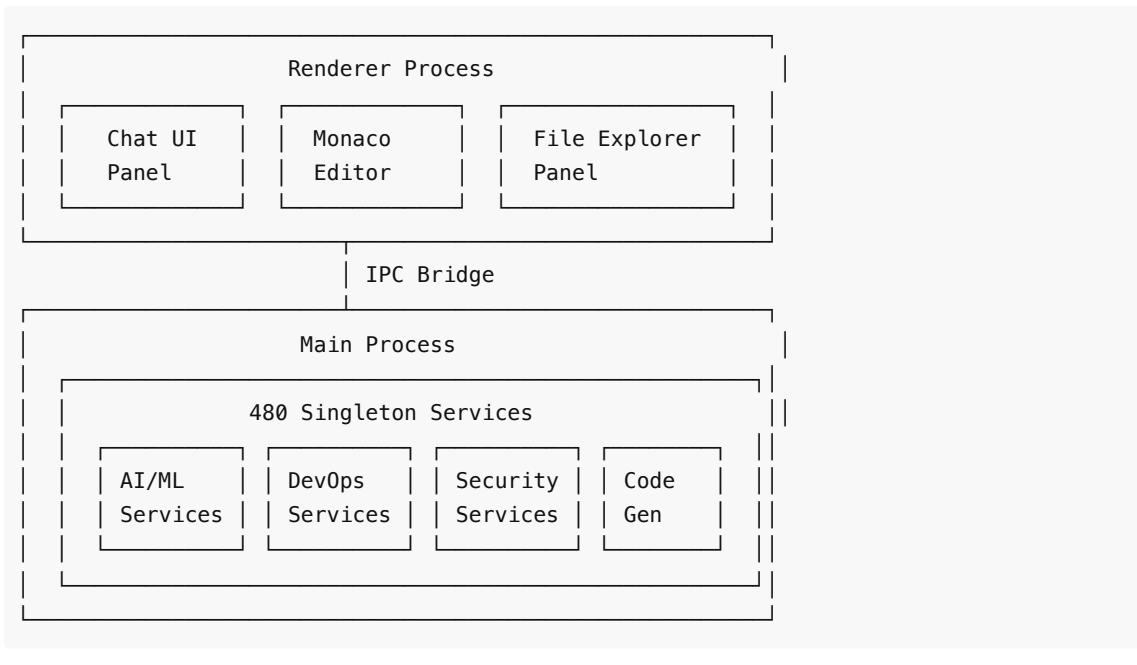
Architecture Overview

Technology Stack

Layer	Technologies
Frontend	React 18, TypeScript, Tailwind CSS, Monaco Editor
Backend	Electron, Node.js, TypeScript

AI Integration	OpenAI, Anthropic Claude, Google Gemini, Ollama
State Management	React Context, Event Emitters
IPC Communication	Electron IPC with 2700+ handlers

Core Architecture Pattern



Core Capabilities

1. Code Generation (150+ Services)

Shadow AI can generate code in virtually any programming language for any purpose:

Languages Supported

- **Web:** JavaScript, TypeScript, HTML, CSS, React, Vue, Angular, Svelte
- **Backend:** Node.js, Python, Go, Rust, Java, C#, Ruby, PHP
- **Mobile:** React Native, Flutter, Swift, Kotlin
- **Systems:** C, C++, Rust, Assembly
- **Data:** SQL, Python, R, Julia
- **Infrastructure:** Terraform, Kubernetes YAML, Docker, Bash, PowerShell

Code Generation Services

Service	Capability
FullStackGeneratorService	Complete full-stack application scaffolding
ComponentGeneratorService	React/Vue/Angular component generation

APIGeneratorService	REST/GraphQL/gRPC API generation
MicroserviceGeneratorService	Microservice architecture generation
DatabaseSchemaService	SQL/NoSQL schema generation
TestGeneratorService	Unit, integration, E2E test generation
DocumentationService	README, API docs, JSDoc generation

2. Autonomous Agent Capabilities

The agent can operate autonomously with minimal human intervention:

Planning & Execution

Service	Capability
GoalDecomposerService	Break complex goals into executable tasks
TaskPlannerService	Create step-by-step execution plans
SelfReflectionService	Evaluate decisions and outcomes
ErrorRecoveryService	Automatic error detection and healing
RecursiveSelfImprovementService	Improve own code and capabilities

Memory & Context

Service	Capability
AgentMemoryService	Long-term memory with embeddings
TemporalMemoryService	Time-aware memory with decay
CognitiveGraphService	Knowledge graph construction
ContextManagerService	Conversation context management
ProjectDNAService	Deep project understanding

3. Multi-Model AI Integration

Supported Providers

Provider	Models	Use Cases
OpenAI	GPT-4, GPT-4 Turbo, GPT-3.5	General coding, reasoning
Anthropic	Claude 3 Opus, Sonnet, Haiku	Long context, analysis
Google	Gemini Pro, Gemini Ultra	Multimodal, large context
Ollama	Llama 3, CodeLlama, Mixtral	Local, private, offline

Kimi K2	Kimi K2	Specialized coding tasks
DeepSeek	DeepSeek V3	Code optimization

AI Services

Service	Capability
ModelRoutingService	Intelligent model selection
CostTrackerService	Usage and cost monitoring
StreamingResponseService	Real-time streaming responses
FunctionCallingService	Structured tool execution
MultiAgentService	Coordinated agent teams

Service Categories

Complete Service Inventory (480 Services)

Category Breakdown

Category	Count	Description
Code Generation	150+	Language-specific generators, templates
AI/ML	50+	Neural networks, NLP, computer vision
DevOps	40+	CI/CD, containers, cloud infrastructure
Security	30+	Authentication, encryption, scanning
Testing	25+	Unit, integration, E2E, performance
Enterprise	40+	Payments, analytics, multi-tenancy
Documentation	15+	Docs, diagrams, comments
Debugging	20+	Error detection, profiling
UI/UX	20+	Figma, prototyping, accessibility
Research	10+	ArXiv, paper-to-code
Total	480	

AI/ML Capabilities

Neural Network Design

Service	Capability
NeuralNetworkDesignerService	Design CNN, RNN, Transformer architectures
FeatureEngineeringService	Automated feature extraction and selection
ModelTrainingService	Training pipelines for PyTorch/TensorFlow

Natural Language Processing

Service	Capability
NLPPipelineService	Text classification, NER, sentiment analysis
EmbeddingService	Generate and manage text embeddings
SemanticSearchService	Semantic code and document search

Computer Vision

Service	Capability
ComputerVisionService	Object detection, OCR, segmentation
ImageProcessingService	Image manipulation and optimization

Recommendation Systems

Service	Capability
RecommendationEngineService	Collaborative and content-based filtering
AnomalyDetectionService	Outlier detection in data and behavior

DevOps & Infrastructure

Container Orchestration

Service	Capability
KubernetesArchitectService	K8s manifests, Helm charts, Kustomize
DockerGeneratorService	Dockerfiles, docker-compose
ServiceMeshService	Istio, Linkerd configuration

Infrastructure as Code

Service	Capability

TerraformDesignerService	AWS, GCP, Azure Terraform modules
CloudFormationService	AWS CloudFormation templates
PulumiService	Pulumi infrastructure code

CI/CD Pipelines

Service	Capability
CICDPipelineService	GitHub Actions, GitLab CI, Jenkins
DeploymentService	Blue-green, canary, rolling deployments

Observability

Service	Capability
MonitoringStackService	Prometheus, Grafana dashboards
LoggingPipelineService	ELK, Loki configuration
TracingService	Distributed tracing setup
AlertingService	Alert rules and escalation

Enterprise Features

Authentication & Authorization

Service	Capability
AuthenticationDesignerService	JWT, OAuth, SSO, MFA implementation
RBACService	Role-based access control
APIKeyService	API key management

Payments & Billing

Service	Capability
PaymentIntegrationService	Stripe, PayPal, crypto payments
SubscriptionService	Recurring billing, trials, upgrades
InvoicingService	Invoice generation and tracking

Analytics & Business Intelligence

Service	Capability
AnalyticsPipelineService	Event tracking, funnels, cohorts
BusinessIntelligenceService	Revenue analysis, PMF, SEO
ABTestingDesignerService	Feature flags, experiments

Multi-Tenancy

Service	Capability
MultiTenancyService	Tenant isolation, routing
FeatureFlagsService	Per-tenant feature toggles

Development Tools

Language-Specific Tools

Service	Capability
CompilerDesignerService	Custom DSLs, lexers, parsers
RegexWizardService	AI-powered regex generation
ShellScriptingService	Bash, PowerShell, cross-platform scripts

Database Tools

Service	Capability
DatabaseDesignerService	Schema design, ERD, migrations
QueryOptimizerService	SQL query optimization
CacheStrategyService	Redis, Memcached configuration

Search & Messaging

Service	Capability
SearchEngineDesignerService	Elasticsearch, Algolia, vector search
MessageQueueDesignerService	RabbitMQ, Kafka, SQS
WebhookDesignerService	Webhook infrastructure

Security Features

Code Security

Service	Capability
SecurityScannerService	Static analysis, SAST
VulnerabilityScannerService	Dependency vulnerability scanning
PenTestSimulatorService	OWASP attack simulation
ThreatModelingService	STRIDE threat analysis

Data Security

Service	Capability
EncryptionDesignerService	E2E encryption, KMS integration
SBOMGeneratorService	Software Bill of Materials
DataMaskingService	PII redaction and masking

Compliance

Service	Capability
ComplianceScannerService	GDPR, SOC2, HIPAA checks
LicenseCheckerService	Open source license compliance

Areas for Development

Based on the current architecture, here are opportunities for improvement:

High Priority Improvements

1. UI/UX Enhancement

- **Current State:** Basic React UI with Monaco Editor
- **Opportunity:** Modern glassmorphism design, animations, dark mode polish
- **Recommended Files:** `src/renderer/App.tsx` , `src/renderer/index.css`

2. Real-Time Collaboration

- **Current State:** Single-user operation
- **Opportunity:** WebSocket-based real-time collaboration
- **Recommended Services:** `RealTimeSyncService` , `PairProgrammingService`

3. Plugin System UI

- **Current State:** Service-based plugins, no UI
- **Opportunity:** Visual plugin marketplace, installation wizard

- **Recommended Services:** `PluginMarketplaceService`

4. Voice Interface

- **Current State:** `VoicePrototypingService` exists
- **Opportunity:** Continuous voice-driven coding experience
- **Integration:** Web Speech API, Whisper API

Medium Priority Improvements

5. Performance Optimization

- **Issue:** Large bundle size (691 KB)
- **Solution:** Code splitting, lazy loading, tree shaking
- **Files:** `vite.config.ts`, component imports

6. Offline Capabilities

- **Current State:** Requires internet for most AI models
- **Opportunity:** Enhanced Ollama integration, local caching
- **Services:** `OllamaService`, `LocalModelManagerService`

7. Testing Coverage

- **Current State:** Limited test coverage
- **Opportunity:** Comprehensive unit and E2E tests
- **Services:** `TestGeneratorService`

8. Mobile Companion App

- **Current State:** Desktop only
- **Opportunity:** React Native companion app
- **Services:** `MobileFirstService`

Low Priority Enhancements

9. Metaverse Development

- **Current State:** `MetaverseDevService` placeholder
- **Opportunity:** Full VR/AR development support

10. Quantum Computing

- **Current State:** `QuantumComputingService` placeholder
- **Opportunity:** Qiskit/Cirq integration

Recommendations

For New Developers

1. **Start Here:** Explore `src/main/services/` to understand service patterns
2. **Key Files:**
 - `src/main/ipc/kimiK2Handlers.ts` - All IPC handlers
 - `src/main/services/*.ts` - 480 service implementations
 - `src/renderer/App.tsx` - Main UI component

For Feature Development

1. Adding New Services:

- o Create service in `src/main/services/`
- o Follow singleton pattern
- o Add IPC handler in `kimiK2Handlers.ts`
- o Run `npm run build` to verify

2. Modifying UI:

- o Components in `src/renderer/components/`
- o Styles in `src/renderer/index.css`
- o State management via React Context

For Production Deployment

1. **Build:** `npm run build`

2. **Package:** `npm run package`

3. **Distribute:** Electron Builder for macOS, Windows, Linux

Appendix: Service Index

Quick Reference

All 480 services follow this pattern:

```
export class ExampleService extends EventEmitter {
    private static instance: ExampleService;
    private constructor() { super(); }

    static getInstance(): ExampleService {
        if (!ExampleService.instance) {
            ExampleService.instance = new ExampleService();
        }
        return ExampleService.instance;
    }

    generate(): string {
        // LLM-powered code generation
        return generatedCode;
    }
}

export const exampleService = ExampleService.getInstance();
```

Service Directories

Directory	Purpose
-----------	---------

src/main/services/	All 480 service implementations
src/main/ ipc/	IPC handlers for service communication
src/main/providers/	AI model providers
src/renderer/components/	React UI components

Shadow AI v13.0.0 - THE ENTERPRISE ARCHITECT 480 Services / Autonomous / Enterprise-Ready / Self-Improving

© 2025 Shadow AI Project