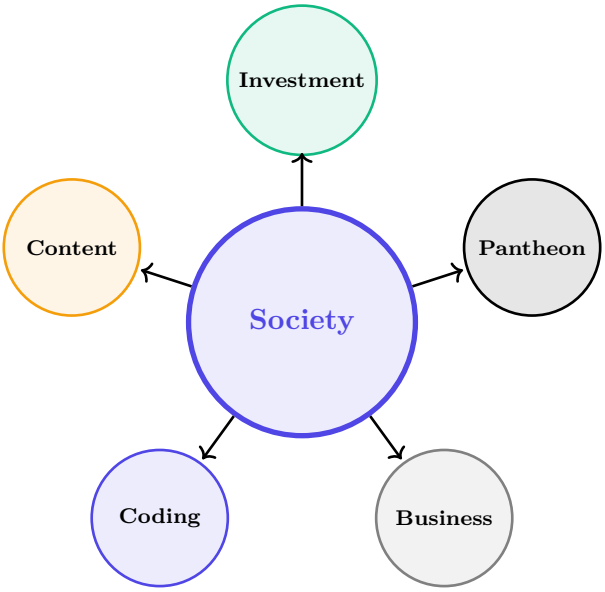
Miyabi Society

Multi-Agent Autonomous System

Technical Specification Document



Version: 1.0.0
Date: December 5, 2025
Status: Draft
Author: Miyabi Development Team

Contents

1 Executive Summary

1.1 Vision

Miyabi Society は、**完全自律型 AI エージェント集団**によるビジネス・開発オペレーションの自動化を目指すプラットフォームです。

- **21+ AI Agents**: 専門分野に特化したエージェント群
- **28+ MCP Servers**: Model Context Protocol によるツール統合
- **Multi-Society**: ドメイン別の自律組織構造
- **A2A Protocol**: エージェント間の標準化通信

1.2 Key Objectives

1. **Autonomous Operation**: 人間の介入を最小化した 24/7 稼働
2. **Scalable Architecture**: 200+ 並列エージェントへのスケーリング
3. **Domain Specialization**: 9 つの専門ドメイン Society
4. **Quality Assurance**: 自動レビュー・品質ゲート

1.3 Document Scope

本仕様書は以下を定義します：

- Society アーキテクチャ設計
- エージェント構成と役割
- 通信プロトコル仕様
- データベーススキーマ
- 実装ロードマップ

2 Architecture Overview

2.1 System Architecture

2.2 Design Principles

1. **Coordinator-Centric**: 各 Society に Coordinator を配置
2. **Context Isolation**: Subagent ごとに独立したコンテキスト窓
3. **Lazy Loading**: 必要時のみ Context をロード
4. **Checkpoint Recovery**: 任意時点への状態復帰
5. **MCP First**: 外部ツール連携は MCP 経由を優先

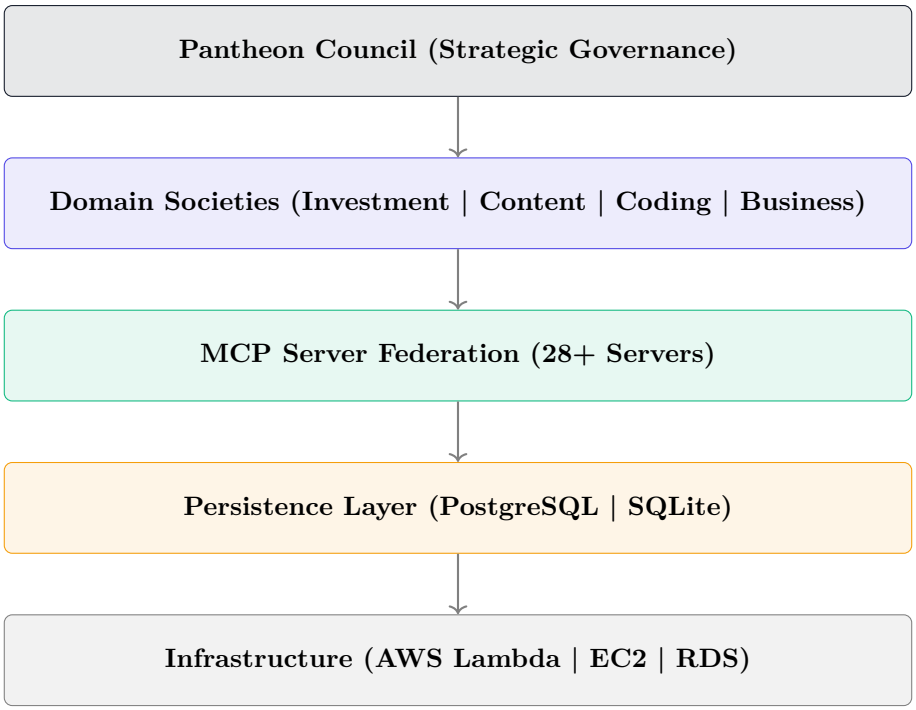


Figure 1: Miyabi Society 5-Layer Architecture

3 Society Structure

3.1 Society Types

3.2 Investment Society (21 Agents)

3.2.1 Hierarchical Structure

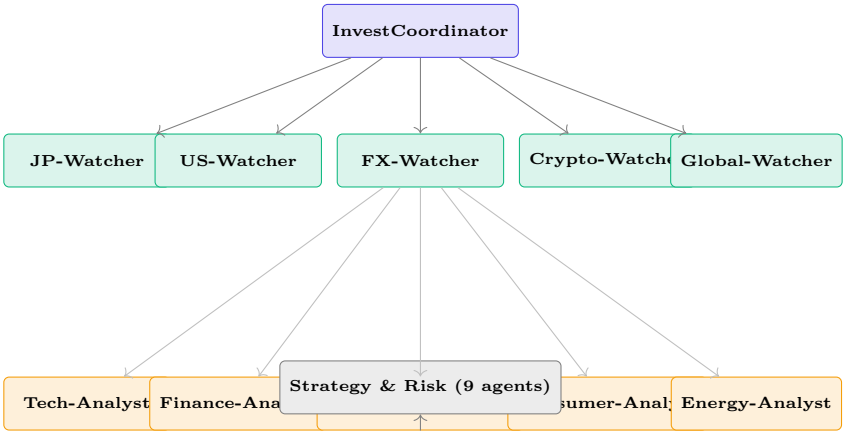


Figure 2: Investment Society Hierarchy

3.2.2 Agent Roles

Society	Mission	Agents	Status
Investment	投資分析・ポートフォリオ管理	21	Active
Content	コンテンツ制作・SNS 戦略	5	Design
Coding	ソフトウェア開発自動化	7	Active
Business	ビジネス戦略・マーケティング	14	Active
Phase 2 (2026)			
Finance	財務・会計自動化	9	Planned
HR & People	採用・人事管理	9	Planned
Sales	営業・リード管理	9	Planned
Customer Success	顧客満足度最大化	8	Planned
Legal	法務・コンプライアンス	8	Planned

Table 1: Society Types and Status

Layer	Agent	Responsibility
Command	InvestCoordinator	タスク分配・最終意思決定
Watchers	JP-Watcher	日経 225/TOPIX 監視
	US-Watcher	S&P500/NASDAQ 監視
	FX-Watcher	主要通貨ペア監視
	Crypto-Watcher	BTC/ETH/主要コイン
	Global-Watcher	世界指数・コモディティ
Analysts	Tech-Analyst	テクノロジーセクター分析
	Finance-Analyst	金融セクター分析
	Healthcare-Analyst	ヘルスケアセクター分析
	Consumer-Analyst	消費財セクター分析
	Energy-Analyst	エネルギーセクター分析
	Industrial-Analyst	産業セクター分析
Strategy	Momentum-Trader	モメンタム戦略立案
	Value-Hunter	バリュー投資銘柄発掘
	Quant-Analyzer	定量分析・バックテスト
	Risk-Manager	リスク評価・VaR 計算
	Executor	売買執行・タイミング最適化

3.3 Content Society (5 Agents)

Agent	Name	Responsibility
かぞえるん	Analytics	メトリクス収集・KPI 追跡
つぶやくん	SNS Strategy	X/Twitter 分析・投稿最適化
どうがくん	YouTube	動画分析・SEO・サムネイル
かくちゃん	Content Creator	レポート・記事・投稿文生成
ひろめるん	Marketing	拡散戦略・インフルエンサー連携

Table 3: Content Society Agents

3.4 Coding Society (7 Agents)

Agent	Name	Responsibility
しきるん	Coordinator	タスク分解・DAG 構築・エージェント指揮
つくるん	CodeGen	AI 駆動コード生成・テスト作成
めだまん	Review	品質スコア算出・セキュリティ検

4 Communication Protocol

4.1 A2A Message Format

Listing 1: A2A Message Structure

```rust
pub struct A2AMessage {
    pub id: Uuid,
    pub session_id: Uuid,
    pub from_agent: AgentId,
    pub to_agent: AgentId,
    pub message_type: MessageType,
    pub content: serde_json::Value,
    pub priority: Priority,
    pub timestamp: DateTime<Utc>,
}

pub enum MessageType {
    Signal,     // 売買シグナル
    Alert,      // リスクアラート
    Data,       // 市場データ
    Report,     // 分析レポート
    Sync,       // 死活監視
    Command,    // コマンド発行
    Response,   // 応答
}

pub enum Priority {
    Critical,   // 即時処理必須
    High,       // 優先処理
    Medium,     // 通常処理
    Low,        // バッチ処理可
}
```

4.2 Communication Layers

Layer	Protocol	Latency	Use Case
L1: tmux	send-keys	∼100ms	ローカル開発
L2: MCP	JSON-RPC/SSE	∼500ms	Claude Code 統合
L3: A2A	HTTP/gRPC	∼1-2s	外部エージェント連携

Table 5: Communication Layer Comparison

4.3 Inter-Society Routing

Listing 2: Routing Rules

```yaml
routing_rules:
  - trigger: "market_alert"
    from: "investment-society"
    to: "content-society"
    action: "generate_market_report"

  - trigger: "content_ready"
```

```
    from: "content-society"
    to: "coding-society"
    action: "commit_to_obsidian"

  - trigger: "deployment_complete"
    from: "coding-society"
    to: "all"
    action: "notify_stakeholders"
```

5 Database Schema

5.1 Entity Relationship

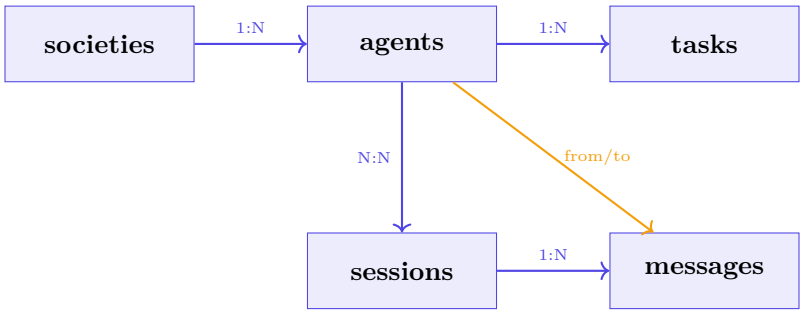


Figure 3: Entity Relationship Diagram

5.2 Core Tables

Listing 3: PostgreSQL Schema

```sql
-- Society Management
CREATE TABLE societies (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    name VARCHAR(100) UNIQUE NOT NULL,
    coordinator_id UUID REFERENCES agents(id),
    config JSONB DEFAULT '{}',
    status VARCHAR(50) DEFAULT 'active',
    created_at TIMESTAMPTZ DEFAULT NOW()
);

-- Agent Registry
CREATE TABLE agents (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    name VARCHAR(255) NOT NULL,
    agent_type VARCHAR(100) NOT NULL,
    society_id UUID REFERENCES societies(id),
    status VARCHAR(50) DEFAULT 'active',
    config JSONB DEFAULT '{}',
    created_at TIMESTAMPTZ DEFAULT NOW(),
    updated_at TIMESTAMPTZ DEFAULT NOW()
);

-- Task Management
CREATE TABLE tasks (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
```

```
    title VARCHAR(500) NOT NULL,
    description TEXT,
    agent_id UUID REFERENCES agents(id),
    status VARCHAR(50) DEFAULT 'pending',
    priority VARCHAR(20) DEFAULT 'medium',
    issue_number INTEGER,
    result JSONB,
    created_at TIMESTAMPTZ DEFAULT NOW(),
    updated_at TIMESTAMPTZ DEFAULT NOW()
);

-- Session Tracking
CREATE TABLE sessions (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    name VARCHAR(255),
    status VARCHAR(50) DEFAULT 'active',
    agents_count INTEGER DEFAULT 0,
    metadata JSONB DEFAULT '{}',
    started_at TIMESTAMPTZ DEFAULT NOW(),
    ended_at TIMESTAMPTZ
);

-- A2A Messages
CREATE TABLE messages (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    session_id UUID REFERENCES sessions(id),
    from_agent UUID REFERENCES agents(id),
    to_agent UUID REFERENCES agents(id),
    message_type VARCHAR(50) NOT NULL,
    content JSONB NOT NULL,
    priority VARCHAR(20) DEFAULT 'medium',
    created_at TIMESTAMPTZ DEFAULT NOW()
);

-- Indexes
CREATE INDEX idx_agents_society ON agents(society_id);
CREATE INDEX idx_tasks_agent ON tasks(agent_id);
CREATE INDEX idx_tasks_status ON tasks(status);
CREATE INDEX idx_messages_session ON messages(session_id);
CREATE INDEX idx_messages_timestamp ON messages(created_at);
```

6 MCP Integration

6.1 Server Inventory

6.2 Tool Examples

Listing 4: Investment Society Tools

```
const investmentTools = [
    "invest_market_overview",      // 市場概要取得
    "invest_quote",                // 株価取得
    "invest_screen_stocks",        // 銘柄スクリーニング
    "invest_technical_analysis",   // テクニカル分析
    "invest_fundamental_analysis", // ファンダメンタル分析
    "invest_risk_metrics",         // リスク指標
    "invest_portfolio_analysis",   // ポートフォリオ分析
```

Category	Server	Tools
Core	miyabi-mcp	15
	miyabi-github	12
	miyabi-tmux	8
AI	gemini3-general	6
	gemini-image-gen	5
	miyabi-openai	4
Society	miyabi-investment-society	11
	miyabi-content-society	5 (planned)
Business	lark-mcp-enhanced	10
	miyabi-commerce	8

Table 6: MCP Server Inventory (Total: 28+)

```
    "invest_news_stock",          // 株別ニュース
    "invest_news_market",         // 市場ニュース
    "invest_news_search",         // ニュース検索
    "invest_analyze",             // 総合分析
];
```

7 Implementation Roadmap

7.1 Phase Overview

Phase	Duration	Deliverable	Cost Est.
P0: Infrastructure	1-2 days	RDS/Lambda 稼働	$150
P1: Coordinator	1 week	3 Society Coordinator	$500
P2: Content Society	2 weeks	5 Agent 実装	$1,000
P3: A2A Integration	1 month	Society 間通信	$2,000
P4: 9 Domains	2 months	Full Society	$4,000
Total	**~3 months**		**$7,650**

Table 7: Implementation Phases

7.2 Phase 0: Infrastructure (1-2 days)

1. AWS RDS PostgreSQL プロビジョニング
   - Instance: db.t3.small
   - Engine: PostgreSQL 15.4
   - Storage: 20GB gp3
2. Lambda Function デプロイ
3. マイグレーション実行

7.3 Phase 1: Coordinator-Centric (1 week)

1. Subagent 定義ファイル作成
   - investment-coordinator.md
   - content-coordinator.md
   - coding-coordinator.md
2. Society Manager 拡張
3. Checkpoint 統合

7.4 Phase 2: Content Society (2 weeks)

1. miyabi-content-society MCP Server 実装
2. 5 Agent 定義・プロンプト作成
3. API 連携（Twitter, YouTube, Gemini）
4. 統合テスト

7.5 Phase 3: A2A Integration (1 month)

1. Inter-Society Protocol 実装
2. tmux オーケストレーション
3. エラーハンドリング・リトライ
4. パフォーマンス最適化

7.6 Phase 4: 9 Domain Expansion (2 months)

1. Finance Society
2. HR & People Society
3. Sales & BizDev Society
4. Customer Success Society
5. Legal & Compliance Society
6. 残り 4 ドメイン

8 Quality Assurance

8.1 Auto-Loop Pattern (Nacho's Pattern)

8.2 Test Coverage Requirements

9 Security Considerations

9.1 MCP Security

- **Prompt Injection**: 入力サニタイズ必須

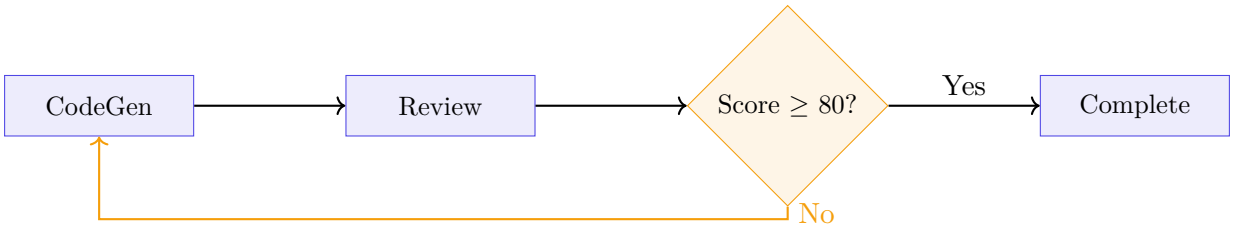


Figure 4: Auto-Loop Quality Gate (Max 3 iterations)

Test Type	Current	Target
Unit Tests	98.1%	≥ 95%
Integration Tests	96.7%	≥ 90%
E2E Tests	94.2%	≥ 85%

Table 8: Test Coverage Requirements

- **Tool Chaining**: 意図しないツール連鎖を防止
- **Credential Management**: 環境変数経由、コードに埋め込み禁止
- **Audit Logging**: 全 MCP 呼び出しをログ記録

9.2 Data Protection

Listing 5: .gitignore Security

```
# Secrets
.env
.env.local
.env.production
credentials.json
private-keys/
*.pem
*.key

# Audit
audit-logs/
```

10 Appendix

10.1 Environment Variables

Listing 6: Required Environment Variables

```
# Database
DATABASE_URL=postgresql://user:pass@host:5432/miyabi

# Authentication
JWT_SECRET=<secret>
GITHUB_CLIENT_ID=<id>
GITHUB_CLIENT_SECRET=<secret>

# AWS
```

```
AWS_REGION=ap-northeast-1
AWS_ACCESS_KEY_ID=<key>
AWS_SECRET_ACCESS_KEY=<secret>

# AI APIs
OPENAI_API_KEY=<key>
GOOGLE_AI_API_KEY=<key>
ANTHROPIC_API_KEY=<key>
```

10.2 References

- Claude Code Documentation: https://docs.anthropic.com/claude-code
- Model Context Protocol: https://modelcontextprotocol.io
- A2A Protocol: https://github.com/google/a2a-spec
- Miyabi Repository: Internal

End of Document
Generated: December 5, 2025