***SutazAI System Architecture***Blueprint

Authoritative Technical Architecture Based on Verified Components. Mandatory Compliance Required for All Development Activities.

Created: August 2025Classification: AUTHORITATIVE TECHNICAL SPECIFICATIONStatus: MANDATORY COMPLIANCE REQUIRED

**Database Layer**

PostgreSQL 16.3 with 14 tables, Redis 7 for caching, Neo4j 5 for graph relationships

**Application Layer**

FastAPI v17.0.0 backend, Streamlit frontend, 70+ API endpoints

**AI Services**

Ollama with TinyLlama (637MB), 7 agent services (stubs), vector databases

**Monitoring**

Prometheus, Grafana, Loki, AlertManager for comprehensive observability

**Current State Assessment**

**Verified Operational Components**

**Core Infrastructure (100% Operational)**

* • PostgreSQL 16.3 (Port 10000) - 14 tables created
* • Redis 7 (Port 10001) - Cache and session store
* • Neo4j 5 (Ports 10002-10003) - Graph database
* • FastAPI Backend v17.0.0 (Port 10010)
* • Streamlit Frontend (Port 10011)

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Service Mesh (Minimal Configuration)**

* • Kong Gateway 3.5 (Ports 10005, 8001) - RUNNING
* • Consul (Port 10006) - Service discovery
* • RabbitMQ 3.12 (Ports 10007, 10008) - Message queue

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Critical Gaps & Deficiencies**

**Immediate Issues**

* • Model Mismatch: Backend expects gpt-oss, TinyLlama loaded
* • Agent Implementation: All agents are stubs without logic
* • Service Mesh Configuration: Kong has no routes defined
* • Vector Database Integration: Running but not integrated

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Architectural Gaps**

* • No Orchestration Layer: Agents cannot coordinate tasks
* • Missing Authentication: No proper auth/authz implementation
* • Limited Scalability: Single-node deployment only
* • No Backup Strategy: Manual backups only

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Documentation vs Reality Analysis**

| **Claimed** | **Reality** | **Status** |
| --- | --- | --- |
| 69 intelligent AI agents | 7 Flask stubs with health endpoints | GAP |
| AGI/ASI capabilities | Basic LLM text generation with TinyLlama | GAP |
| Production ready | 20% complete proof-of-concept | PARTIAL |

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Technical Architecture**

**System Topology**

**Current Network Architecture**

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Target Architecture (Production)**

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Data Architecture**

**Current Data Flow**

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Target Data Flow**

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**PostgreSQL Schema Overview**

User Management

* • users
* • sessions
* • api\_usage\_logs

AI Services

* • agents
* • tasks
* • agent\_executions
* • model\_registry

Knowledge & Monitoring

* • chat\_history
* • knowledge\_documents
* • vector\_collections
* • system\_metrics
* • system\_alerts

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Implementation Roadmap**

**1**

**Phase 1: Foundation (Immediate - 7 Days)**

Stabilize core infrastructure and fix critical issues

**Week 1 Objectives**

* **Day 1:** Fix model configuration mismatch
* **Day 2:** Database schema verification
* **Days 3-4:** Service mesh configuration
* **Days 5-6:** Implement basic agent logic
* **Day 7:** Fix ChromaDB connection

**Deliverables**

* • Working LLM integration
* • Configured service mesh
* • One functional agent
* • Vector database integration

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**2**

**Phase 2: Integration (30 Days)**

Implement core functionality and agent orchestration

**Weeks 2-3: Core Integration**

* • Agent communication framework (RabbitMQ)
* • Authentication system (JWT, RBAC)
* • API Gateway configuration

**Weeks 3-4: Advanced Features**

* • Vector database integration
* • Enhanced monitoring (custom dashboards)
* • Agent implementation (all stubs → functional)

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**3**

**Phase 3: Production Readiness (60 Days)**

Performance optimization, scaling, and security hardening

**Weeks 5-8: Performance & Scaling**

* • Performance optimization
* • Horizontal scaling (Kubernetes)
* • High availability (replication)

**Weeks 9-12: Security & Operations**

* • Security hardening (TLS, secrets management)
* • CI/CD pipeline implementation
* • Documentation and training

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Implementation Timeline**

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Technical Specifications**

**API Specifications**

**RESTful API Standards**

POST

/api/v1/auth/login

User authentication with JWT tokens

GET

/api/v1/agents

List available agents and capabilities

POST

/api/v1/tasks

Submit new task for processing

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**WebSocket Endpoints**

ws://host/ws/chat

Real-time chat interactions

ws://host/ws/updates

System status and task updates

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Agent Specifications**

**Agent Interface Standard**

class BaseAgent(ABC):

"""Standard interface for all agents"""

def \_\_init\_\_(self, agent\_id: str, config: Dict[str, Any]):

self.agent\_id = agent\_id

self.config = config

self.status = "initializing"

@abstractmethod

async def process(self, task: Dict[str, Any]) -> Dict[str, Any]:

"""Process task and return results"""

@abstractmethod

async def health\_check(self) -> Dict[str, Any]:

"""Return health status"""

@abstractmethod

async def get\_capabilities(self) -> List[str]:

"""Return list of capabilities"""

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Agent Capabilities Matrix**

Text Processing Agenttext\_generation, summarization

Data Analysis Agentstatistical\_analysis, forecasting

Task Orchestration Agenttask\_routing, resource\_allocation

Knowledge Management Agentsemantic\_search, relationship\_mapping

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Operational Requirements**

**Performance Requirements**

**Response Time Targets**

* • API Endpoints: P99 < 500ms
* • Database Queries: < 100ms
* • Agent Processing: < 30 seconds
* • Page Load: < 2 seconds

**Throughput Requirements**

* • Concurrent Users: 1000
* • Requests per Second: 500
* • Tasks per Minute: 100
* • Messages per Second: 1000

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Security Requirements**

**Authentication**

* • JWT Token Authentication (RS256)
* • 15-minute access token expiry
* • 7-day refresh token expiry
* • Token rotation on refresh

**Encryption Standards**

* • Data at Rest: AES-256 encryption
* • Data in Transit: TLS 1.3 minimum
* • Secrets Management: HashiCorp Vault
* • Key rotation every 90 days

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Reliability Requirements**

**Availability Targets**

* • System Availability: 99.9%
* • API Availability: 99.95%
* • Database Availability: 99.99%
* • Error rate: < 0.1%

**Failure Recovery**

* • RTO: 1 hour
* • RPO: 15 minutes
* • MTTR: 30 minutes
* • MTBF: 720 hours

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Compliance & Governance**

**Architecture Principles**

**Design Patterns**

* **Microservices Architecture**

Service isolation, independent deployment, fault isolation

* **Event-Driven Architecture**

Loose coupling, asynchronous processing, event sourcing

* **API-First Design**

OpenAPI specification, versioning, backward compatibility

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Technology Standards**

Languages

* • Python 3.11+
* • TypeScript 5.0+
* • Bash, Python scripts

Databases

* • PostgreSQL 15+
* • Redis 7+
* • Neo4j 5+

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Compliance Framework**

**Regulatory Requirements**

* **Data Protection**

GDPR compliance (EU), CCPA compliance (California)

* **Security Standards**

OWASP Top 10 mitigation, CIS benchmarks, NIST framework

* **Industry Standards**

ISO 27001, SOC 2 Type II, PCI DSS (if applicable)

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Internal Policies**

Code Quality

Minimum 80% test coverage, code review mandatory, static analysis required

Security Policies

Vulnerability scanning, penetration testing quarterly, incident response plan

Operational Policies

Change management process, capacity planning reviews, performance benchmarking

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Appendices**

**Service Registry**

**Core Infrastructure Services**

| **Service** | **Port** | **Status** |
| --- | --- | --- |
| PostgreSQL | 10000 | WORKING |
| Redis | 10001 | WORKING |
| Backend API | 10010 | WORKING |
| Kong Gateway | 10005 | RUNNING |

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Configuration Standards**

**Environment Variables**

# Database Configuration

POSTGRES\_HOST=localhost

POSTGRES\_PORT=10000

POSTGRES\_DB=sutazai

# Application Configuration

ENVIRONMENT=development|staging|production

DEBUG=true|false

LOG\_LEVEL=DEBUG|INFO|WARNING|ERROR

# Security Configuration

JWT\_SECRET\_KEY=<secure\_key>

JWT\_ALGORITHM=RS256

**Feature Flags**

* **ENABLE\_VECTOR\_SEARCH:** false
* **ENABLE\_AGENT\_ORCHESTRATION:** false
* **ENABLE\_ADVANCED\_MONITORING:** true
* **ENABLE\_AUTHENTICATION:** false

[[1]](blob:https://www.kimi.com/694a066d-5300-447c-8f22-2e3edfac4c98#citation-1)

**Document Control**

**Version History**

| **Version** | **Date** | **Author** |
| --- | --- | --- |
| 1.0 | August 2025 | Sr. System Architect |

**Review Schedule**

* • Technical Review: Monthly
* • Architecture Review: Quarterly
* • Compliance Review: Annually
* • Full Revision: As needed

**Distribution**

* • Development Team: Full access
* • Operations Team: Full access
* • Management: Executive summary
* • External Auditors: Upon request

# ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>SutazAI System Architecture Blueprint</title>

<script src="https://cdn.tailwindcss.com"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/mermaid/10.6.1/mermaid.min.js"></script>

<link href="https://fonts.googleapis.com/css2?family=Playfair+Display:ital,wght@0,400;0,700;1,400&family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

<script>

tailwind.config = {

theme: {

extend: {

colors: {

primary: '#1e293b',

secondary: '#475569',

accent: '#3b82f6',

muted: '#64748b',

background: '#f8fafc',

surface: '#ffffff'

},

fontFamily: {

serif: ['Playfair Display', 'serif'],

sans: ['Inter', 'sans-serif']

}

}

}

}

</script>

<style>

.gradient-overlay {

background: linear-gradient(135deg, rgba(30, 41, 59, 0.9) 0%, rgba(59, 130, 246, 0.7) 100%);

}

.citation-link {

color: #3b82f6;

text-decoration: none;

font-size: 0.75rem;

vertical-align: super;

margin-left: 2px;

}

.citation-link:hover {

text-decoration: underline;

}

.mermaid-container {

display: flex;

justify-content: center;

min-height: 300px;

max-height: 800px;

background: #ffffff;

border: 2px solid #e5e7eb;

border-radius: 12px;

padding: 30px;

margin: 30px 0;

box-shadow: 0 8px 25px rgba(0, 0, 0, 0.08);

position: relative;

overflow: hidden;

}

.mermaid-container .mermaid {

width: 100%;

max-width: 100%;

height: 100%;

cursor: grab;

transition: transform 0.3s ease;

transform-origin: center center;

display: flex;

justify-content: center;

align-items: center;

touch-action: none;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

}

.mermaid-container .mermaid svg {

max-width: 100%;

height: 100%;

display: block;

margin: 0 auto;

}

.mermaid-container .mermaid:active {

cursor: grabbing;

}

.mermaid-container.zoomed .mermaid {

height: 100%;

width: 100%;

cursor: grab;

}

.mermaid-controls {

position: absolute;

top: 15px;

right: 15px;

display: flex;

gap: 10px;

z-index: 20;

background: rgba(255, 255, 255, 0.95);

padding: 8px;

border-radius: 8px;

box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);

}

.mermaid-control-btn {

background: #ffffff;

border: 1px solid #d1d5db;

border-radius: 6px;

padding: 10px;

cursor: pointer;

transition: all 0.2s ease;

color: #374151;

font-size: 14px;

min-width: 36px;

height: 36px;

text-align: center;

display: flex;

align-items: center;

justify-content: center;

}

.mermaid-control-btn:hover {

background: #f8fafc;

border-color: #3b82f6;

color: #3b82f6;

transform: translateY(-1px);

}

.mermaid-control-btn:active {

transform: scale(0.95);

}

/\* Enhanced mermaid styling for better contrast and unified appearance \*/

.mermaid svg {

max-width: none !important;

height: auto !important;

font-family: 'Inter', sans-serif !important;

}

/\* Improve contrast for different node colors \*/

.mermaid .node rect,

.mermaid .node circle,

.mermaid .node ellipse,

.mermaid .node polygon {

stroke: #1e293b !important;

stroke-width: 2px !important;

filter: drop-shadow(0 2px 4px rgba(0,0,0,0.1));

}

.mermaid .node .label {

color: #1e293b !important;

font-weight: 600 !important;

font-size: 14px !important;

text-shadow: 0 1px 2px rgba(255,255,255,0.8);

}

/\* Specific styling for different node types to ensure contrast \*/

.mermaid .node[style\*="fill:#e3f2fd"] .label,

.mermaid .node[style\*="fill:#f3e5f5"] .label,

.mermaid .node[style\*="fill:#e8f5e8"] .label,

.mermaid .node[style\*="fill:#fff3e0"] .label,

.mermaid .node[style\*="fill:#fce4ec"] .label {

color: #1e293b !important;

font-weight: 700 !important;

}

.mermaid .edgePath .path {

stroke: #64748b !important;

stroke-width: 2.5px !important;

filter: drop-shadow(0 1px 2px rgba(0,0,0,0.1));

}

.mermaid .arrowheadPath {

fill: #64748b !important;

filter: drop-shadow(0 1px 2px rgba(0,0,0,0.1));

}

/\* Timeline specific styling \*/

.mermaid .timeline-section {

font-weight: 600 !important;

}

.mermaid .timeline-event {

background: rgba(255,255,255,0.95) !important;

border: 2px solid #e2e8f0 !important;

border-radius: 8px !important;

padding: 12px !important;

box-shadow: 0 4px 6px -1px rgba(0, 0, 0, 0.1) !important;

}

.mermaid .timeline-title {

color: #1e293b !important;

font-weight: 700 !important;

font-size: 16px !important;

}

.mermaid .timeline-text {

color: #475569 !important;

font-size: 14px !important;

line-height: 1.4 !important;

}

/\* Flowchart specific improvements \*/

.mermaid .flowchart-link {

stroke: #64748b !important;

stroke-width: 2.5px !important;

}

.mermaid .marker {

fill: #64748b !important;

}

/\* Ensure all text has proper contrast \*/

.mermaid text {

fill: #1e293b !important;

font-weight: 600 !important;

text-shadow: 0 1px 2px rgba(255,255,255,0.8);

}

@media (max-width: 1024px) {

.mermaid-control-btn:not(.reset-zoom) {

display: none;

}

.mermaid-controls {

top: auto;

bottom: 15px;

right: 15px;

}

}

</style>

</head>

<body class="bg-background font-sans text-primary leading-relaxed overflow-x-hidden">

<!-- Fixed Table of Contents -->

<nav class="fixed left-0 top-0 h-screen w-80 bg-surface border-r border-gray-200 overflow-y-auto z-40 shadow-lg hidden lg:block">

<div class="p-6">

<h3 class="font-serif text-lg font-bold mb-6 text-primary">Table of Contents</h3>

<ul class="space-y-3 text-sm">

<li>

<a href="#hero" class="block py-2 px-3 rounded-lg hover:bg-gray-50 transition-colors">Executive Summary</a>

</li>

<li>

<a href="#current-state" class="block py-2 px-3 rounded-lg hover:bg-gray-50 transition-colors">Current State Assessment</a>

</li>

<li>

<a href="#technical-architecture" class="block py-2 px-3 rounded-lg hover:bg-gray-50 transition-colors">Technical Architecture</a>

</li>

<li>

<a href="#implementation-roadmap" class="block py-2 px-3 rounded-lg hover:bg-gray-50 transition-colors">Implementation Roadmap</a>

</li>

<li>

<a href="#technical-specifications" class="block py-2 px-3 rounded-lg hover:bg-gray-50 transition-colors">Technical Specifications</a>

</li>

<li>

<a href="#operational-requirements" class="block py-2 px-3 rounded-lg hover:bg-gray-50 transition-colors">Operational Requirements</a>

</li>

<li>

<a href="#compliance-governance" class="block py-2 px-3 rounded-lg hover:bg-gray-50 transition-colors">Compliance & Governance</a>

</li>

<li>

<a href="#appendices" class="block py-2 px-3 rounded-lg hover:bg-gray-50 transition-colors">Appendices</a>

</li>

</ul>

</div>

</nav>

<!-- Main Content -->

<main class="lg:ml-80">

<!-- Hero Section -->

<section id="hero" class="relative min-h-screen bg-gradient-to-br from-primary via-secondary to-accent">

<div class="absolute inset-0 gradient-overlay"></div>

<div class="relative z-10 min-h-screen flex items-center">

<div class="max-w-7xl mx-auto px-4 md:px-8 py-20">

<div class="grid grid-cols-1 gap-12 items-center">

<!-- Title Section -->

<div>

<h1 class="font-serif text-3xl md:text-4xl lg:text-6xl font-bold text-white mb-6 leading-tight">

<em>SutazAI System Architecture</em>

<br>

<span class="text-xl md:text-2xl lg:text-4xl font-normal">Blueprint</span>

</h1>

<p class="text-base md:text-xl text-gray-200 mb-8 max-w-full md:max-w-2xl">

Authoritative Technical Architecture Based on Verified Components.

Mandatory Compliance Required for All Development Activities.

</p>

<div class="flex flex-wrap items-center gap-4 text-sm text-gray-300">

<span class="flex items-center">

<i class="fas fa-calendar mr-2"></i>

Created: August 2025

</span>

<span class="flex items-center">

<i class="fas fa-shield-alt mr-2"></i>

Classification: AUTHORITATIVE TECHNICAL SPECIFICATION

</span>

<span class="flex items-center">

<i class="fas fa-exclamation-triangle mr-2"></i>

Status: MANDATORY COMPLIANCE REQUIRED

</span>

</div>

</div>

<!-- Key Highlights Grid -->

<div class="grid grid-cols-1 md:grid-cols-2 gap-6 mt-8 md:mt-0">

<div class="bg-white/10 backdrop-blur-sm rounded-xl p-6 border border-white/20">

<div class="flex items-center mb-4">

<i class="fas fa-database text-2xl text-blue-300 mr-3"></i>

<h3 class="font-semibold text-white">Database Layer</h3>

</div>

<p class="text-gray-200 text-sm">PostgreSQL 16.3 with 14 tables, Redis 7 for caching, Neo4j 5 for graph relationships</p>

</div>

<div class="bg-white/10 backdrop-blur-sm rounded-xl p-6 border border-white/20">

<div class="flex items-center mb-4">

<i class="fas fa-server text-2xl text-green-300 mr-3"></i>

<h3 class="font-semibold text-white">Application Layer</h3>

</div>

<p class="text-gray-200 text-sm">FastAPI v17.0.0 backend, Streamlit frontend, 70+ API endpoints</p>

</div>

<div class="bg-white/10 backdrop-blur-sm rounded-xl p-6 border border-white/20">

<div class="flex items-center mb-4">

<i class="fas fa-brain text-2xl text-purple-300 mr-3"></i>

<h3 class="font-semibold text-white">AI Services</h3>

</div>

<p class="text-gray-200 text-sm">Ollama with TinyLlama (637MB), 7 agent services (stubs), vector databases</p>

</div>

<div class="bg-white/10 backdrop-blur-sm rounded-xl p-6 border border-white/20">

<div class="flex items-center mb-4">

<i class="fas fa-chart-line text-2xl text-yellow-300 mr-3"></i>

<h3 class="font-semibold text-white">Monitoring</h3>

</div>

<p class="text-gray-200 text-sm">Prometheus, Grafana, Loki, AlertManager for comprehensive observability</p>

</div>

</div>

</div>

</div>

</div>

</section>

<!-- Current State Assessment -->

<section id="current-state" class="py-20 bg-surface">

<div class="max-w-6xl mx-auto px-4 md:px-8">

<h2 class="font-serif text-4xl font-bold mb-12 text-primary">Current State Assessment</h2>

<div class="grid grid-cols-1 lg:grid-cols-2 gap-12 mb-16">

<div>

<h3 class="font-serif text-2xl font-bold mb-6">Verified Operational Components</h3>

<div class="space-y-6">

<div class="bg-green-50 border-l-4 border-green-400 p-6 rounded-r-lg">

<h4 class="font-semibold text-green-800 mb-2">Core Infrastructure (100% Operational)</h4>

<ul class="text-sm text-green-700 space-y-1">

<li>• PostgreSQL 16.3 (Port 10000) - 14 tables created</li>

<li>• Redis 7 (Port 10001) - Cache and session store</li>

<li>• Neo4j 5 (Ports 10002-10003) - Graph database</li>

<li>• FastAPI Backend v17.0.0 (Port 10010)</li>

<li>• Streamlit Frontend (Port 10011)</li>

</ul>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

<div class="bg-blue-50 border-l-4 border-blue-400 p-6 rounded-r-lg">

<h4 class="font-semibold text-blue-800 mb-2">Service Mesh (Minimal Configuration)</h4>

<ul class="text-sm text-blue-700 space-y-1">

<li>• Kong Gateway 3.5 (Ports 10005, 8001) - RUNNING</li>

<li>• Consul (Port 10006) - Service discovery</li>

<li>• RabbitMQ 3.12 (Ports 10007, 10008) - Message queue</li>

</ul>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

</div>

</div>

<div>

<h3 class="font-serif text-2xl font-bold mb-6">Critical Gaps & Deficiencies</h3>

<div class="space-y-6">

<div class="bg-red-50 border-l-4 border-red-400 p-6 rounded-r-lg">

<h4 class="font-semibold text-red-800 mb-2">Immediate Issues</h4>

<ul class="text-sm text-red-700 space-y-1">

<li>• Model Mismatch: Backend expects gpt-oss, TinyLlama loaded</li>

<li>• Agent Implementation: All agents are stubs without logic</li>

<li>• Service Mesh Configuration: Kong has no routes defined</li>

<li>• Vector Database Integration: Running but not integrated</li>

</ul>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

<div class="bg-orange-50 border-l-4 border-orange-400 p-6 rounded-r-lg">

<h4 class="font-semibold text-orange-800 mb-2">Architectural Gaps</h4>

<ul class="text-sm text-orange-700 space-y-1">

<li>• No Orchestration Layer: Agents cannot coordinate tasks</li>

<li>• Missing Authentication: No proper auth/authz implementation</li>

<li>• Limited Scalability: Single-node deployment only</li>

<li>• No Backup Strategy: Manual backups only</li>

</ul>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

</div>

</div>

</div>

<!-- Reality vs Claims Comparison -->

<div class="bg-gray-50 rounded-xl p-8">

<h4 class="font-serif text-xl font-bold mb-6">Documentation vs Reality Analysis</h4>

<div class="overflow-x-auto">

<table class="w-full text-sm">

<thead>

<tr class="border-b border-gray-300">

<th class="text-left py-3 px-4 font-semibold">Claimed</th>

<th class="text-left py-3 px-4 font-semibold">Reality</th>

<th class="text-left py-3 px-4 font-semibold">Status</th>

</tr>

</thead>

<tbody class="divide-y divide-gray-200">

<tr>

<td class="py-3 px-4">69 intelligent AI agents</td>

<td class="py-3 px-4">7 Flask stubs with health endpoints</td>

<td class="py-3 px-4"><span class="bg-red-100 text-red-800 px-2 py-1 rounded text-xs">GAP</span></td>

</tr>

<tr>

<td class="py-3 px-4">AGI/ASI capabilities</td>

<td class="py-3 px-4">Basic LLM text generation with TinyLlama</td>

<td class="py-3 px-4"><span class="bg-red-100 text-red-800 px-2 py-1 rounded text-xs">GAP</span></td>

</tr>

<tr>

<td class="py-3 px-4">Production ready</td>

<td class="py-3 px-4">20% complete proof-of-concept</td>

<td class="py-3 px-4"><span class="bg-yellow-100 text-yellow-800 px-2 py-1 rounded text-xs">PARTIAL</span></td>

</tr>

</tbody>

</table>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

</div>

</section>

<!-- Technical Architecture -->

<section id="technical-architecture" class="py-20 bg-background">

<div class="max-w-6xl mx-auto px-4 md:px-8">

<h2 class="font-serif text-4xl font-bold mb-12 text-primary">Technical Architecture</h2>

<!-- System Topology -->

<div class="mb-16">

<h3 class="font-serif text-2xl font-bold mb-8">System Topology</h3>

<div class="bg-surface rounded-xl p-8 shadow-sm border border-gray-200 mb-8">

<h4 class="font-semibold mb-6">Current Network Architecture</h4>

<div class="mermaid-container">

<div class="mermaid-controls">

<button class="mermaid-control-btn zoom-in" title="放大">

<i class="fas fa-search-plus"></i>

</button>

<button class="mermaid-control-btn zoom-out" title="缩小">

<i class="fas fa-search-minus"></i>

</button>

<button class="mermaid-control-btn reset-zoom" title="重置">

<i class="fas fa-expand-arrows-alt"></i>

</button>

<button class="mermaid-control-btn fullscreen" title="全屏查看">

<i class="fas fa-expand"></i>

</button>

</div>

<div class="mermaid" id="mermaid-1">

graph TB

subgraph "Docker Host (Single Node)"

subgraph "Docker Network: sutazai-network"

PG["PostgreSQL

<br />:10000"]

Redis["Redis

<br />:10001"]

Neo4j["Neo4j

<br />:10002-03"]

Backend["Backend API

<br />:10010"]

Frontend["Frontend

<br />:10011"]

Ollama["Ollama

<br />:10104"]

Kong["Kong Gateway

<br />:10005, 8001"]

Consul["Consul

<br />:10006"]

RabbitMQ["RabbitMQ

<br />:10007-08"]

Agents["7 Agent Stubs

<br />Ports 8xxx"]

VectorDBs["Vector DBs

<br />Qdrant, FAISS, ChromaDB"]

Prometheus["Prometheus

<br />:10200"]

Grafana["Grafana

<br />:10201"]

end

end

Frontend --> Backend

Backend --> PG

Backend --> Redis

Backend --> Neo4j

Backend --> Ollama

Agents --> Backend

VectorDBs --> Backend

Prometheus -.-> All

Grafana --> Prometheus

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

<div class="bg-surface rounded-xl p-8 shadow-sm border border-gray-200">

<h4 class="font-semibold mb-6">Target Architecture (Production)</h4>

<div class="mermaid-container">

<div class="mermaid-controls">

<button class="mermaid-control-btn zoom-in" title="放大">

<i class="fas fa-search-plus"></i>

</button>

<button class="mermaid-control-btn zoom-out" title="缩小">

<i class="fas fa-search-minus"></i>

</button>

<button class="mermaid-control-btn reset-zoom" title="重置">

<i class="fas fa-expand-arrows-alt"></i>

</button>

<button class="mermaid-control-btn fullscreen" title="全屏查看">

<i class="fas fa-expand"></i>

</button>

</div>

<div class="mermaid" id="mermaid-2">

graph TB

LB["Load Balancer (HA)"]

KongCluster["Kong API Gateway Cluster

<br />Authentication, Rate Limiting"]

BackendCluster["Backend Cluster"]

AgentCluster["Agent Services Cluster"]

FrontendCluster["Frontend Cluster"]

DataLayer["Data Layer (HA)"]

PG["PostgreSQL

<br />Primary/Replicas"]

Redis["Redis Cluster"]

Neo4j["Neo4j Cluster"]

LB --> KongCluster

KongCluster --> BackendCluster

KongCluster --> AgentCluster

KongCluster --> FrontendCluster

BackendCluster --> DataLayer

AgentCluster --> DataLayer

DataLayer --> PG

DataLayer --> Redis

DataLayer --> Neo4j

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

</div>

<!-- Data Architecture -->

<div class="mb-16">

<h3 class="font-serif text-2xl font-bold mb-8">Data Architecture</h3>

<div class="grid grid-cols-1 lg:grid-cols-2 gap-8">

<div class="bg-surface rounded-xl p-6 shadow-sm border border-gray-200">

<h4 class="font-semibold mb-4">Current Data Flow</h4>

<div class="mermaid-container">

<div class="mermaid-controls">

<button class="mermaid-control-btn zoom-in" title="放大">

<i class="fas fa-search-plus"></i>

</button>

<button class="mermaid-control-btn zoom-out" title="缩小">

<i class="fas fa-search-minus"></i>

</button>

<button class="mermaid-control-btn reset-zoom" title="重置">

<i class="fas fa-expand-arrows-alt"></i>

</button>

<button class="mermaid-control-btn fullscreen" title="全屏查看">

<i class="fas fa-expand"></i>

</button>

</div>

<div class="mermaid" id="mermaid-3">

graph LR

User["User Request"] --> Frontend["Frontend"]

Frontend --> Backend["Backend API"]

Backend --> Database["Database"]

Backend --> Ollama["Ollama Service"]

Ollama --> Backend

Backend --> Response["Response → User"]

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

<div class="bg-surface rounded-xl p-6 shadow-sm border border-gray-200">

<h4 class="font-semibold mb-4">Target Data Flow</h4>

<div class="mermaid-container">

<div class="mermaid-controls">

<button class="mermaid-control-btn zoom-in" title="放大">

<i class="fas fa-search-plus"></i>

</button>

<button class="mermaid-control-btn zoom-out" title="缩小">

<i class="fas fa-search-minus"></i>

</button>

<button class="mermaid-control-btn reset-zoom" title="重置">

<i class="fas fa-expand-arrows-alt"></i>

</button>

<button class="mermaid-control-btn fullscreen" title="全屏查看">

<i class="fas fa-expand"></i>

</button>

</div>

<div class="mermaid" id="mermaid-4">

graph LR

User["User Request"] --> APIGW["API Gateway"]

APIGW --> LB["Load Balancer"]

LB --> Backend["Backend Service"]

Backend --> TaskQ["Task Queue (RabbitMQ)"]

TaskQ --> Orchestrator["Agent Orchestrator"]

Orchestrator --> AgentPool["Agent Pool

<br />Parallel Processing"]

AgentPool --> ResultAgg["Result Aggregation"]

ResultAgg --> ResponseQ["Response Queue"]

ResponseQ --> Backend

Backend --> User

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

</div>

</div>

<!-- Database Schema -->

<div class="bg-surface rounded-xl p-8 shadow-sm border border-gray-200">

<h4 class="font-semibold mb-6">PostgreSQL Schema Overview</h4>

<div class="grid grid-cols-1 md:grid-cols-3 gap-6">

<div>

<h5 class="font-medium text-accent mb-3">User Management</h5>

<ul class="text-sm space-y-1 text-secondary">

<li>• users</li>

<li>• sessions</li>

<li>• api\_usage\_logs</li>

</ul>

</div>

<div>

<h5 class="font-medium text-accent mb-3">AI Services</h5>

<ul class="text-sm space-y-1 text-secondary">

<li>• agents</li>

<li>• tasks</li>

<li>• agent\_executions</li>

<li>• model\_registry</li>

</ul>

</div>

<div>

<h5 class="font-medium text-accent mb-3">Knowledge & Monitoring</h5>

<ul class="text-sm space-y-1 text-secondary">

<li>• chat\_history</li>

<li>• knowledge\_documents</li>

<li>• vector\_collections</li>

<li>• system\_metrics</li>

<li>• system\_alerts</li>

</ul>

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

</div>

</section>

<!-- Implementation Roadmap -->

<section id="implementation-roadmap" class="py-20 bg-surface">

<div class="max-w-6xl mx-auto px-4 md:px-8">

<h2 class="font-serif text-4xl font-bold mb-12 text-primary">Implementation Roadmap</h2>

<div class="space-y-12">

<!-- Phase 1 -->

<div class="bg-green-50 border border-green-200 rounded-xl p-8">

<div class="flex items-center mb-6">

<div class="bg-green-500 text-white rounded-full w-10 h-10 flex items-center justify-center font-bold mr-4">1</div>

<div>

<h3 class="font-serif text-xl font-bold">Phase 1: Foundation (Immediate - 7 Days)</h3>

<p class="text-green-700 text-sm">Stabilize core infrastructure and fix critical issues</p>

</div>

</div>

<div class="grid grid-cols-1 md:grid-cols-2 gap-6">

<div>

<h4 class="font-semibold mb-3">Week 1 Objectives</h4>

<ul class="space-y-2 text-sm">

<li class="flex items-start">

<i class="fas fa-check-circle text-green-500 mt-1 mr-2"></i>

<span><strong>Day 1:</strong> Fix model configuration mismatch</span>

</li>

<li class="flex items-start">

<i class="fas fa-check-circle text-green-500 mt-1 mr-2"></i>

<span><strong>Day 2:</strong> Database schema verification</span>

</li>

<li class="flex items-start">

<i class="fas fa-check-circle text-green-500 mt-1 mr-2"></i>

<span><strong>Days 3-4:</strong> Service mesh configuration</span>

</li>

<li class="flex items-start">

<i class="fas fa-check-circle text-green-500 mt-1 mr-2"></i>

<span><strong>Days 5-6:</strong> Implement basic agent logic</span>

</li>

<li class="flex items-start">

<i class="fas fa-check-circle text-green-500 mt-1 mr-2"></i>

<span><strong>Day 7:</strong> Fix ChromaDB connection</span>

</li>

</ul>

</div>

<div>

<h4 class="font-semibold mb-3">Deliverables</h4>

<ul class="space-y-2 text-sm text-green-700">

<li>• Working LLM integration</li>

<li>• Configured service mesh</li>

<li>• One functional agent</li>

<li>• Vector database integration</li>

</ul>

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

<!-- Phase 2 -->

<div class="bg-blue-50 border border-blue-200 rounded-xl p-8">

<div class="flex items-center mb-6">

<div class="bg-blue-500 text-white rounded-full w-10 h-10 flex items-center justify-center font-bold mr-4">2</div>

<div>

<h3 class="font-serif text-xl font-bold">Phase 2: Integration (30 Days)</h3>

<p class="text-blue-700 text-sm">Implement core functionality and agent orchestration</p>

</div>

</div>

<div class="grid grid-cols-1 md:grid-cols-2 gap-6">

<div>

<h4 class="font-semibold mb-3">Weeks 2-3: Core Integration</h4>

<ul class="space-y-2 text-sm">

<li>• Agent communication framework (RabbitMQ)</li>

<li>• Authentication system (JWT, RBAC)</li>

<li>• API Gateway configuration</li>

</ul>

</div>

<div>

<h4 class="font-semibold mb-3">Weeks 3-4: Advanced Features</h4>

<ul class="space-y-2 text-sm">

<li>• Vector database integration</li>

<li>• Enhanced monitoring (custom dashboards)</li>

<li>• Agent implementation (all stubs → functional)</li>

</ul>

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

<!-- Phase 3 -->

<div class="bg-purple-50 border border-purple-200 rounded-xl p-8">

<div class="flex items-center mb-6">

<div class="bg-purple-500 text-white rounded-full w-10 h-10 flex items-center justify-center font-bold mr-4">3</div>

<div>

<h3 class="font-serif text-xl font-bold">Phase 3: Production Readiness (60 Days)</h3>

<p class="text-purple-700 text-sm">Performance optimization, scaling, and security hardening</p>

</div>

</div>

<div class="grid grid-cols-1 md:grid-cols-2 gap-6">

<div>

<h4 class="font-semibold mb-3">Weeks 5-8: Performance & Scaling</h4>

<ul class="space-y-2 text-sm">

<li>• Performance optimization</li>

<li>• Horizontal scaling (Kubernetes)</li>

<li>• High availability (replication)</li>

</ul>

</div>

<div>

<h4 class="font-semibold mb-3">Weeks 9-12: Security & Operations</h4>

<ul class="space-y-2 text-sm">

<li>• Security hardening (TLS, secrets management)</li>

<li>• CI/CD pipeline implementation</li>

<li>• Documentation and training</li>

</ul>

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

</div>

<!-- Timeline Visualization -->

<div class="mt-16 bg-surface rounded-xl p-8 shadow-sm border border-gray-200">

<h4 class="font-semibold mb-6">Implementation Timeline</h4>

<div class="mermaid-container">

<div class="mermaid-controls">

<button class="mermaid-control-btn zoom-in" title="放大">

<i class="fas fa-search-plus"></i>

</button>

<button class="mermaid-control-btn zoom-out" title="缩小">

<i class="fas fa-search-minus"></i>

</button>

<button class="mermaid-control-btn reset-zoom" title="重置">

<i class="fas fa-expand-arrows-alt"></i>

</button>

<button class="mermaid-control-btn fullscreen" title="全屏查看">

<i class="fas fa-expand"></i>

</button>

</div>

<div class="mermaid" id="mermaid-5">

timeline

title "SutazAI Implementation Roadmap"

Q3 2025 : "Phase 1 - Foundation"

: "Week 1: Model & DB Fixes"

: "Week 2: Service Mesh Config"

Q4 2025 : "Phase 2 - Integration"

: "Weeks 3-4: Authentication"

: "Weeks 5-6: Agent Framework"

: "Weeks 7-8: Vector DB Integration"

Q1 2026 : "Phase 3 - Production"

: "Weeks 9-12: Performance"

: "Weeks 13-16: Security"

: "Weeks 17-20: CI/CD Pipeline"

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

</div>

</section>

<!-- Technical Specifications -->

<section id="technical-specifications" class="py-20 bg-background">

<div class="max-w-6xl mx-auto px-4 md:px-8">

<h2 class="font-serif text-4xl font-bold mb-12 text-primary">Technical Specifications</h2>

<div class="grid grid-cols-1 lg:grid-cols-2 gap-12">

<!-- API Specifications -->

<div>

<h3 class="font-serif text-2xl font-bold mb-8">API Specifications</h3>

<div class="space-y-6">

<div class="bg-surface rounded-xl p-6 shadow-sm border border-gray-200">

<h4 class="font-semibold mb-4 text-accent">RESTful API Standards</h4>

<div class="space-y-3">

<div class="flex items-start">

<span class="bg-green-100 text-green-800 px-2 py-1 rounded text-xs font-mono mr-3">POST</span>

<div>

<code class="text-sm">/api/v1/auth/login</code>

<p class="text-xs text-secondary mt-1">User authentication with JWT tokens</p>

</div>

</div>

<div class="flex items-start">

<span class="bg-blue-100 text-blue-800 px-2 py-1 rounded text-xs font-mono mr-3">GET</span>

<div>

<code class="text-sm">/api/v1/agents</code>

<p class="text-xs text-secondary mt-1">List available agents and capabilities</p>

</div>

</div>

<div class="flex items-start">

<span class="bg-yellow-100 text-yellow-800 px-2 py-1 rounded text-xs font-mono mr-3">POST</span>

<div>

<code class="text-sm">/api/v1/tasks</code>

<p class="text-xs text-secondary mt-1">Submit new task for processing</p>

</div>

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

<div class="bg-surface rounded-xl p-6 shadow-sm border border-gray-200">

<h4 class="font-semibold mb-4 text-accent">WebSocket Endpoints</h4>

<div class="space-y-2">

<div>

<code class="text-sm bg-gray-100 px-2 py-1 rounded">ws://host/ws/chat</code>

<p class="text-xs text-secondary mt-1">Real-time chat interactions</p>

</div>

<div>

<code class="text-sm bg-gray-100 px-2 py-1 rounded">ws://host/ws/updates</code>

<p class="text-xs text-secondary mt-1">System status and task updates</p>

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

</div>

</div>

<!-- Agent Specifications -->

<div>

<h3 class="font-serif text-2xl font-bold mb-8">Agent Specifications</h3>

<div class="bg-surface rounded-xl p-6 shadow-sm border border-gray-200 mb-6">

<h4 class="font-semibold mb-4 text-accent">Agent Interface Standard</h4>

<div class="bg-gray-50 rounded-lg p-4 font-mono text-sm overflow-x-auto">

<pre class="text-xs">class BaseAgent(ABC):

"""Standard interface for all agents"""

def \_\_init\_\_(self, agent\_id: str, config: Dict[str, Any]):

self.agent\_id = agent\_id

self.config = config

self.status = "initializing"

@abstractmethod

async def process(self, task: Dict[str, Any]) -> Dict[str, Any]:

"""Process task and return results"""

@abstractmethod

async def health\_check(self) -> Dict[str, Any]:

"""Return health status"""

@abstractmethod

async def get\_capabilities(self) -> List[str]:

"""Return list of capabilities"""</pre>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

<div class="bg-surface rounded-xl p-6 shadow-sm border border-gray-200">

<h4 class="font-semibold mb-4 text-accent">Agent Capabilities Matrix</h4>

<div class="space-y-3 text-sm">

<div class="flex justify-between items-center p-2 bg-blue-50 rounded">

<span class="font-medium">Text Processing Agent</span>

<span class="text-blue-700">text\_generation, summarization</span>

</div>

<div class="flex justify-between items-center p-2 bg-green-50 rounded">

<span class="font-medium">Data Analysis Agent</span>

<span class="text-green-700">statistical\_analysis, forecasting</span>

</div>

<div class="flex justify-between items-center p-2 bg-purple-50 rounded">

<span class="font-medium">Task Orchestration Agent</span>

<span class="text-purple-700">task\_routing, resource\_allocation</span>

</div>

<div class="flex justify-between items-center p-2 bg-orange-50 rounded">

<span class="font-medium">Knowledge Management Agent</span>

<span class="text-orange-700">semantic\_search, relationship\_mapping</span>

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

</div>

</div>

</div>

</section>

<!-- Operational Requirements -->

<section id="operational-requirements" class="py-20 bg-surface">

<div class="max-w-6xl mx-auto px-4 md:px-8">

<h2 class="font-serif text-4xl font-bold mb-12 text-primary">Operational Requirements</h2>

<div class="grid grid-cols-1 lg:grid-cols-3 gap-8">

<!-- Performance Requirements -->

<div class="bg-white rounded-xl p-6 shadow-sm border border-gray-200">

<h3 class="font-serif text-lg font-bold mb-6 text-accent">Performance Requirements</h3>

<div class="space-y-4">

<div>

<h4 class="font-semibold mb-2">Response Time Targets</h4>

<ul class="text-sm space-y-1">

<li>• API Endpoints: P99 < 500ms</li>

<li>• Database Queries: < 100ms</li>

<li>• Agent Processing: < 30 seconds</li>

<li>• Page Load: < 2 seconds</li>

</ul>

</div>

<div>

<h4 class="font-semibold mb-2">Throughput Requirements</h4>

<ul class="text-sm space-y-1">

<li>• Concurrent Users: 1000</li>

<li>• Requests per Second: 500</li>

<li>• Tasks per Minute: 100</li>

<li>• Messages per Second: 1000</li>

</ul>

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

<!-- Security Requirements -->

<div class="bg-white rounded-xl p-6 shadow-sm border border-gray-200">

<h3 class="font-serif text-lg font-bold mb-6 text-accent">Security Requirements</h3>

<div class="space-y-4">

<div>

<h4 class="font-semibold mb-2">Authentication</h4>

<ul class="text-sm space-y-1">

<li>• JWT Token Authentication (RS256)</li>

<li>• 15-minute access token expiry</li>

<li>• 7-day refresh token expiry</li>

<li>• Token rotation on refresh</li>

</ul>

</div>

<div>

<h4 class="font-semibold mb-2">Encryption Standards</h4>

<ul class="text-sm space-y-1">

<li>• Data at Rest: AES-256 encryption</li>

<li>• Data in Transit: TLS 1.3 minimum</li>

<li>• Secrets Management: HashiCorp Vault</li>

<li>• Key rotation every 90 days</li>

</ul>

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

<!-- Reliability Requirements -->

<div class="bg-white rounded-xl p-6 shadow-sm border border-gray-200">

<h3 class="font-serif text-lg font-bold mb-6 text-accent">Reliability Requirements</h3>

<div class="space-y-4">

<div>

<h4 class="font-semibold mb-2">Availability Targets</h4>

<ul class="text-sm space-y-1">

<li>• System Availability: 99.9%</li>

<li>• API Availability: 99.95%</li>

<li>• Database Availability: 99.99%</li>

<li>• Error rate: < 0.1%</li>

</ul>

</div>

<div>

<h4 class="font-semibold mb-2">Failure Recovery</h4>

<ul class="text-sm space-y-1">

<li>• RTO: 1 hour</li>

<li>• RPO: 15 minutes</li>

<li>• MTTR: 30 minutes</li>

<li>• MTBF: 720 hours</li>

</ul>

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

</div>

</div>

</section>

<!-- Compliance & Governance -->

<section id="compliance-governance" class="py-20 bg-background">

<div class="max-w-6xl mx-auto px-4 md:px-8">

<h2 class="font-serif text-4xl font-bold mb-12 text-primary">Compliance & Governance</h2>

<div class="grid grid-cols-1 lg:grid-cols-2 gap-12">

<div>

<h3 class="font-serif text-2xl font-bold mb-8">Architecture Principles</h3>

<div class="space-y-6">

<div class="bg-surface rounded-xl p-6 shadow-sm border border-gray-200">

<h4 class="font-semibold mb-4 text-accent">Design Patterns</h4>

<ul class="space-y-3 text-sm">

<li class="flex items-start">

<i class="fas fa-cubes text-blue-500 mt-1 mr-3"></i>

<div>

<strong>Microservices Architecture</strong>

<p class="text-secondary mt-1">Service isolation, independent deployment, fault isolation</p>

</div>

</li>

<li class="flex items-start">

<i class="fas fa-bolt text-yellow-500 mt-1 mr-3"></i>

<div>

<strong>Event-Driven Architecture</strong>

<p class="text-secondary mt-1">Loose coupling, asynchronous processing, event sourcing</p>

</div>

</li>

<li class="flex items-start">

<i class="fas fa-code text-green-500 mt-1 mr-3"></i>

<div>

<strong>API-First Design</strong>

<p class="text-secondary mt-1">OpenAPI specification, versioning, backward compatibility</p>

</div>

</li>

</ul>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

<div class="bg-surface rounded-xl p-6 shadow-sm border border-gray-200">

<h4 class="font-semibold mb-4 text-accent">Technology Standards</h4>

<div class="grid grid-cols-2 gap-4 text-sm">

<div>

<h5 class="font-medium mb-2">Languages</h5>

<ul class="space-y-1 text-secondary">

<li>• Python 3.11+</li>

<li>• TypeScript 5.0+</li>

<li>• Bash, Python scripts</li>

</ul>

</div>

<div>

<h5 class="font-medium mb-2">Databases</h5>

<ul class="space-y-1 text-secondary">

<li>• PostgreSQL 15+</li>

<li>• Redis 7+</li>

<li>• Neo4j 5+</li>

</ul>

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

</div>

</div>

<div>

<h3 class="font-serif text-2xl font-bold mb-8">Compliance Framework</h3>

<div class="space-y-6">

<div class="bg-surface rounded-xl p-6 shadow-sm border border-gray-200">

<h4 class="font-semibold mb-4 text-accent">Regulatory Requirements</h4>

<ul class="space-y-3 text-sm">

<li class="flex items-start">

<i class="fas fa-shield-alt text-blue-500 mt-1 mr-3"></i>

<div>

<strong>Data Protection</strong>

<p class="text-secondary mt-1">GDPR compliance (EU), CCPA compliance (California)</p>

</div>

</li>

<li class="flex items-start">

<i class="fas fa-lock text-red-500 mt-1 mr-3"></i>

<div>

<strong>Security Standards</strong>

<p class="text-secondary mt-1">OWASP Top 10 mitigation, CIS benchmarks, NIST framework</p>

</div>

</li>

<li class="flex items-start">

<i class="fas fa-certificate text-green-500 mt-1 mr-3"></i>

<div>

<strong>Industry Standards</strong>

<p class="text-secondary mt-1">ISO 27001, SOC 2 Type II, PCI DSS (if applicable)</p>

</div>

</li>

</ul>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

<div class="bg-surface rounded-xl p-6 shadow-sm border border-gray-200">

<h4 class="font-semibold mb-4 text-accent">Internal Policies</h4>

<div class="space-y-3 text-sm">

<div>

<h5 class="font-medium mb-1">Code Quality</h5>

<p class="text-secondary">Minimum 80% test coverage, code review mandatory, static analysis required</p>

</div>

<div>

<h5 class="font-medium mb-1">Security Policies</h5>

<p class="text-secondary">Vulnerability scanning, penetration testing quarterly, incident response plan</p>

</div>

<div>

<h5 class="font-medium mb-1">Operational Policies</h5>

<p class="text-secondary">Change management process, capacity planning reviews, performance benchmarking</p>

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

</div>

</div>

</div>

</div>

</section>

<!-- Appendices -->

<section id="appendices" class="py-20 bg-surface">

<div class="max-w-6xl mx-auto px-4 md:px-8">

<h2 class="font-serif text-4xl font-bold mb-12 text-primary">Appendices</h2>

<div class="grid grid-cols-1 lg:grid-cols-2 gap-8">

<!-- Service Registry -->

<div class="bg-white rounded-xl p-6 shadow-sm border border-gray-200">

<h3 class="font-serif text-xl font-bold mb-6 text-accent">Service Registry</h3>

<div class="space-y-4">

<div>

<h4 class="font-semibold mb-2">Core Infrastructure Services</h4>

<div class="overflow-x-auto">

<table class="w-full text-sm">

<thead>

<tr class="border-b border-gray-200">

<th class="text-left py-2 px-2">Service</th>

<th class="text-left py-2 px-2">Port</th>

<th class="text-left py-2 px-2">Status</th>

</tr>

</thead>

<tbody class="divide-y divide-gray-100">

<tr>

<td class="py-2 px-2">PostgreSQL</td>

<td class="py-2 px-2">10000</td>

<td class="py-2 px-2"><span class="bg-green-100 text-green-800 px-2 py-1 rounded text-xs">WORKING</span></td>

</tr>

<tr>

<td class="py-2 px-2">Redis</td>

<td class="py-2 px-2">10001</td>

<td class="py-2 px-2"><span class="bg-green-100 text-green-800 px-2 py-1 rounded text-xs">WORKING</span></td>

</tr>

<tr>

<td class="py-2 px-2">Backend API</td>

<td class="py-2 px-2">10010</td>

<td class="py-2 px-2"><span class="bg-green-100 text-green-800 px-2 py-1 rounded text-xs">WORKING</span></td>

</tr>

<tr>

<td class="py-2 px-2">Kong Gateway</td>

<td class="py-2 px-2">10005</td>

<td class="py-2 px-2"><span class="bg-yellow-100 text-yellow-800 px-2 py-1 rounded text-xs">RUNNING</span></td>

</tr>

</tbody>

</table>

</div>

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

<!-- Configuration Standards -->

<div class="bg-white rounded-xl p-6 shadow-sm border border-gray-200">

<h3 class="font-serif text-xl font-bold mb-6 text-accent">Configuration Standards</h3>

<div class="space-y-4">

<div>

<h4 class="font-semibold mb-2">Environment Variables</h4>

<div class="bg-gray-50 rounded-lg p-4 font-mono text-xs overflow-x-auto">

<pre># Database Configuration

POSTGRES\_HOST=localhost

POSTGRES\_PORT=10000

POSTGRES\_DB=sutazai

# Application Configuration

ENVIRONMENT=development|staging|production

DEBUG=true|false

LOG\_LEVEL=DEBUG|INFO|WARNING|ERROR

# Security Configuration

JWT\_SECRET\_KEY=&lt;secure\_key&gt;

JWT\_ALGORITHM=RS256</pre>

</div>

</div>

<div>

<h4 class="font-semibold mb-2">Feature Flags</h4>

<div class="bg-blue-50 rounded-lg p-4 text-sm">

<ul class="space-y-1">

<li><strong>ENABLE\_VECTOR\_SEARCH:</strong> false</li>

<li><strong>ENABLE\_AGENT\_ORCHESTRATION:</strong> false</li>

<li><strong>ENABLE\_ADVANCED\_MONITORING:</strong> true</li>

<li><strong>ENABLE\_AUTHENTICATION:</strong> false</li>

</ul>

</div>

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

</div>

<!-- Document Control -->

<div class="mt-12 bg-white rounded-xl p-8 shadow-sm border border-gray-200">

<h3 class="font-serif text-xl font-bold mb-6 text-accent">Document Control</h3>

<div class="grid grid-cols-1 md:grid-cols-3 gap-8">

<div>

<h4 class="font-semibold mb-3">Version History</h4>

<table class="w-full text-sm">

<thead>

<tr class="border-b border-gray-200">

<th class="text-left py-2">Version</th>

<th class="text-left py-2">Date</th>

<th class="text-left py-2">Author</th>

</tr>

</thead>

<tbody>

<tr class="border-b border-gray-100">

<td class="py-2">1.0</td>

<td class="py-2">August 2025</td>

<td class="py-2">Sr. System Architect</td>

</tr>

</tbody>

</table>

</div>

<div>

<h4 class="font-semibold mb-3">Review Schedule</h4>

<ul class="text-sm space-y-1">

<li>• Technical Review: Monthly</li>

<li>• Architecture Review: Quarterly</li>

<li>• Compliance Review: Annually</li>

<li>• Full Revision: As needed</li>

</ul>

</div>

<div>

<h4 class="font-semibold mb-3">Distribution</h4>

<ul class="text-sm space-y-1">

<li>• Development Team: Full access</li>

<li>• Operations Team: Full access</li>

<li>• Management: Executive summary</li>

<li>• External Auditors: Upon request</li>

</ul>

</div>

</div>

<a href="#citation-1" class="citation-link">[1]</a>

</div>

</div>

</section>

<!-- Citations -->

<section class="py-12 bg-background border-t border-gray-200">

<div class="max-w-6xl mx-auto px-4 md:px-8">

<h3 class="font-serif text-xl font-bold mb-6">Citations</h3>

<div class="text-sm text-secondary space-y-2">

<p id="citation-1"><strong>[1]</strong> SutazAI System Architecture Blueprint - Authoritative Technical Specification, August 2025. Document Version: 1.0.</p>

</div>

</div>

</section>

</main>

<script>

// Initialize Mermaid with enhanced configuration for better display

mermaid.initialize({

startOnLoad: true,

theme: 'base',

themeVariables: {

primaryColor: '#ffffff',

primaryTextColor: '#1e293b',

primaryBorderColor: '#1e293b',

lineColor: '#64748b',

secondaryColor: '#f1f5f9',

tertiaryColor: '#e2e8f0',

background: '#ffffff',

mainBkg: '#ffffff',

secondBkg: '#f8fafc',

tertiaryBkg: '#f1f5f9',

// Enhanced contrast colors for different node types

cScale0: '#ffffff',

cScale1: '#e3f2fd',

cScale2: '#f3e5f5',

cScale3: '#e8f5e8',

cScale4: '#fff3e0',

cScale5: '#fce4ec',

// Text colors for better readability

cScaleLabel0: '#1e293b',

cScaleLabel1: '#1e293b',

cScaleLabel2: '#1e293b',

cScaleLabel3: '#1e293b',

cScaleLabel4: '#1e293b',

cScaleLabel5: '#1e293b',

// Additional contrast improvements

nodeBorder: '#1e293b',

clusterBkg: '#f8fafc',

clusterBorder: '#64748b',

defaultLinkColor: '#64748b',

titleColor: '#1e293b',

edgeLabelBackground: '#ffffff',

// Timeline specific colors

cScale6: '#f8fafc',

cScale7: '#f1f5f9',

cScale8: '#e2e8f0',

cScale9: '#cbd5e1'

},

flowchart: {

useMaxWidth: false,

htmlLabels: true,

curve: 'basis',

padding: 20

},

timeline: {

useMaxWidth: false,

padding: 20

},

fontFamily: 'Inter, sans-serif',

fontSize: 14,

// Ensure proper sizing and centering

gantt: {

useMaxWidth: false

}

});

// Initialize Mermaid Controls for zoom and pan

function initializeMermaidControls() {

const containers = document.querySelectorAll('.mermaid-container');

containers.forEach(container => {

const mermaidElement = container.querySelector('.mermaid');

let scale = 1;

let isDragging = false;

let startX, startY, translateX = 0, translateY = 0;

// 触摸相关状态

let isTouch = false;

let touchStartTime = 0;

let initialDistance = 0;

let initialScale = 1;

let isPinching = false;

// Zoom controls

const zoomInBtn = container.querySelector('.zoom-in');

const zoomOutBtn = container.querySelector('.zoom-out');

const resetBtn = container.querySelector('.reset-zoom');

const fullscreenBtn = container.querySelector('.fullscreen');

function updateTransform() {

mermaidElement.style.transform = `translate(${translateX}px, ${translateY}px) scale(${scale})`;

if (scale > 1) {

container.classList.add('zoomed');

} else {

container.classList.remove('zoomed');

}

mermaidElement.style.cursor = isDragging ? 'grabbing' : 'grab';

}

if (zoomInBtn) {

zoomInBtn.addEventListener('click', () => {

scale = Math.min(scale \* 1.25, 4);

updateTransform();

});

}

if (zoomOutBtn) {

zoomOutBtn.addEventListener('click', () => {

scale = Math.max(scale / 1.25, 0.3);

if (scale <= 1) {

translateX = 0;

translateY = 0;

}

updateTransform();

});

}

if (resetBtn) {

resetBtn.addEventListener('click', () => {

scale = 1;

translateX = 0;

translateY = 0;

updateTransform();

});

}

if (fullscreenBtn) {

fullscreenBtn.addEventListener('click', () => {

if (container.requestFullscreen) {

container.requestFullscreen();

} else if (container.webkitRequestFullscreen) {

container.webkitRequestFullscreen();

} else if (container.msRequestFullscreen) {

container.msRequestFullscreen();

}

});

}

// Mouse Events

mermaidElement.addEventListener('mousedown', (e) => {

if (isTouch) return; // 如果是触摸设备，忽略鼠标事件

isDragging = true;

startX = e.clientX - translateX;

startY = e.clientY - translateY;

mermaidElement.style.cursor = 'grabbing';

updateTransform();

e.preventDefault();

});

document.addEventListener('mousemove', (e) => {

if (isDragging && !isTouch) {

translateX = e.clientX - startX;

translateY = e.clientY - startY;

updateTransform();

}

});

document.addEventListener('mouseup', () => {

if (isDragging && !isTouch) {

isDragging = false;

mermaidElement.style.cursor = 'grab';

updateTransform();

}

});

document.addEventListener('mouseleave', () => {

if (isDragging && !isTouch) {

isDragging = false;

mermaidElement.style.cursor = 'grab';

updateTransform();

}

});

// 获取两点之间的距离

function getTouchDistance(touch1, touch2) {

return Math.hypot(

touch2.clientX - touch1.clientX,

touch2.clientY - touch1.clientY

);

}

// Touch Events - 触摸事件处理

mermaidElement.addEventListener('touchstart', (e) => {

isTouch = true;

touchStartTime = Date.now();

if (e.touches.length === 1) {

// 单指拖动

isPinching = false;

isDragging = true;

const touch = e.touches[0];

startX = touch.clientX - translateX;

startY = touch.clientY - translateY;

} else if (e.touches.length === 2) {

// 双指缩放

isPinching = true;

isDragging = false;

const touch1 = e.touches[0];

const touch2 = e.touches[1];

initialDistance = getTouchDistance(touch1, touch2);

initialScale = scale;

}

e.preventDefault();

}, { passive: false });

mermaidElement.addEventListener('touchmove', (e) => {

if (e.touches.length === 1 && isDragging && !isPinching) {

// 单指拖动

const touch = e.touches[0];

translateX = touch.clientX - startX;

translateY = touch.clientY - startY;

updateTransform();

} else if (e.touches.length === 2 && isPinching) {

// 双指缩放

const touch1 = e.touches[0];

const touch2 = e.touches[1];

const currentDistance = getTouchDistance(touch1, touch2);

if (initialDistance > 0) {

const newScale = Math.min(Math.max(

initialScale \* (currentDistance / initialDistance),

0.3

), 4);

scale = newScale;

updateTransform();

}

}

e.preventDefault();

}, { passive: false });

mermaidElement.addEventListener('touchend', (e) => {

// 重置状态

if (e.touches.length === 0) {

isDragging = false;

isPinching = false;

initialDistance = 0;

// 延迟重置isTouch，避免鼠标事件立即触发

setTimeout(() => {

isTouch = false;

}, 100);

} else if (e.touches.length === 1 && isPinching) {

// 从双指变为单指，切换为拖动模式

isPinching = false;

isDragging = true;

const touch = e.touches[0];

startX = touch.clientX - translateX;

startY = touch.clientY - translateY;

}

updateTransform();

});

mermaidElement.addEventListener('touchcancel', (e) => {

isDragging = false;

isPinching = false;

initialDistance = 0;

setTimeout(() => {

isTouch = false;

}, 100);

updateTransform();

});

// Enhanced wheel zoom with better center point handling

container.addEventListener('wheel', (e) => {

e.preventDefault();

const rect = container.getBoundingClientRect();

const centerX = rect.width / 2;

const centerY = rect.height / 2;

const delta = e.deltaY > 0 ? 0.9 : 1.1;

const newScale = Math.min(Math.max(scale \* delta, 0.3), 4);

// Adjust translation to zoom towards center

if (newScale !== scale) {

const scaleDiff = newScale / scale;

translateX = translateX \* scaleDiff;

translateY = translateY \* scaleDiff;

scale = newScale;

if (scale <= 1) {

translateX = 0;

translateY = 0;

}

updateTransform();

}

});

// Initialize display

updateTransform();

});

}

// Initialize Mermaid with enhanced configuration for better display

mermaid.initialize({

startOnLoad: true,

theme: 'base',

themeVariables: {

primaryColor: '#ffffff',

primaryTextColor: '#1e293b',

primaryBorderColor: '#1e293b',

lineColor: '#64748b',

secondaryColor: '#f1f5f9',

tertiaryColor: '#e2e8f0',

background: '#ffffff',

mainBkg: '#ffffff',

secondBkg: '#f8fafc',

tertiaryBkg: '#f1f5f9',

// Enhanced contrast colors for different node types

cScale0: '#ffffff',

cScale1: '#e3f2fd',

cScale2: '#f3e5f5',

cScale3: '#e8f5e8',

cScale4: '#fff3e0',

cScale5: '#fce4ec',

// Text colors for better readability

cScaleLabel0: '#1e293b',

cScaleLabel1: '#1e293b',

cScaleLabel2: '#1e293b',

cScaleLabel3: '#1e293b',

cScaleLabel4: '#1e293b',

cScaleLabel5: '#1e293b',

// Additional contrast improvements

nodeBorder: '#1e293b',

clusterBkg: '#f8fafc',

clusterBorder: '#64748b',

defaultLinkColor: '#64748b',

titleColor: '#1e293b',

edgeLabelBackground: '#ffffff',

// Timeline specific colors

cScale6: '#f8fafc',

cScale7: '#f1f5f9',

cScale8: '#e2e8f0',

cScale9: '#cbd5e1'

},

flowchart: {

useMaxWidth: false,

htmlLabels: true,

curve: 'basis',

padding: 20

},

timeline: {

useMaxWidth: false,

padding: 20

},

fontFamily: 'Inter, sans-serif',

fontSize: 14,

// Ensure proper sizing and centering

gantt: {

useMaxWidth: false

}

});

// Initialize Mermaid Controls for zoom and pan

function initializeMermaidControls() {

const containers = document.querySelectorAll('.mermaid-container');

containers.forEach(container => {

const mermaidElement = container.querySelector('.mermaid');

let scale = 1;

let isDragging = false;

let startX, startY, translateX = 0, translateY = 0;

// 触摸相关状态

let isTouch = false;

let touchStartTime = 0;

let initialDistance = 0;

let initialScale = 1;

let isPinching = false;

// Zoom controls

const zoomInBtn = container.querySelector('.zoom-in');

const zoomOutBtn = container.querySelector('.zoom-out');

const resetBtn = container.querySelector('.reset-zoom');

const fullscreenBtn = container.querySelector('.fullscreen');

function updateTransform() {

mermaidElement.style.transform = `translate(${translateX}px, ${translateY}px) scale(${scale})`;

if (scale > 1) {

container.classList.add('zoomed');

} else {

container.classList.remove('zoomed');

}

mermaidElement.style.cursor = isDragging ? 'grabbing' : 'grab';

}

if (zoomInBtn) {

zoomInBtn.addEventListener('click', () => {

scale = Math.min(scale \* 1.25, 4);

updateTransform();

});

}

if (zoomOutBtn) {

zoomOutBtn.addEventListener('click', () => {

scale = Math.max(scale / 1.25, 0.3);

if (scale <= 1) {

translateX = 0;

translateY = 0;

}

updateTransform();

});

}

if (resetBtn) {

resetBtn.addEventListener('click', () => {

scale = 1;

translateX = 0;

translateY = 0;

updateTransform();

});

}

if (fullscreenBtn) {

fullscreenBtn.addEventListener('click', () => {

if (container.requestFullscreen) {

container.requestFullscreen();

} else if (container.webkitRequestFullscreen) {

container.webkitRequestFullscreen();

} else if (container.msRequestFullscreen) {

container.msRequestFullscreen();

}

});

}

// Mouse Events

mermaidElement.addEventListener('mousedown', (e) => {

if (isTouch) return; // 如果是触摸设备，忽略鼠标事件

isDragging = true;

startX = e.clientX - translateX;

startY = e.clientY - translateY;

mermaidElement.style.cursor = 'grabbing';

updateTransform();

e.preventDefault();

});

document.addEventListener('mousemove', (e) => {

if (isDragging && !isTouch) {

translateX = e.clientX - startX;

translateY = e.clientY - startY;

updateTransform();

}

});

document.addEventListener('mouseup', () => {

if (isDragging && !isTouch) {

isDragging = false;

mermaidElement.style.cursor = 'grab';

updateTransform();

}

});

document.addEventListener('mouseleave', () => {

if (isDragging && !isTouch) {

isDragging = false;

mermaidElement.style.cursor = 'grab';

updateTransform();

}

});

// 获取两点之间的距离

function getTouchDistance(touch1, touch2) {

return Math.hypot(

touch2.clientX - touch1.clientX,

touch2.clientY - touch1.clientY

);

}

// Touch Events - 触摸事件处理

mermaidElement.addEventListener('touchstart', (e) => {

isTouch = true;

touchStartTime = Date.now();

if (e.touches.length === 1) {

// 单指拖动

isPinching = false;

isDragging = true;

const touch = e.touches[0];

startX = touch.clientX - translateX;

startY = touch.clientY - translateY;

} else if (e.touches.length === 2) {

// 双指缩放

isPinching = true;

isDragging = false;

const touch1 = e.touches[0];

const touch2 = e.touches[1];

initialDistance = getTouchDistance(touch1, touch2);

initialScale = scale;

}

e.preventDefault();

}, { passive: false });

mermaidElement.addEventListener('touchmove', (e) => {

if (e.touches.length === 1 && isDragging && !isPinching) {

// 单指拖动

const touch = e.touches[0];

translateX = touch.clientX - startX;

translateY = touch.clientY - startY;

updateTransform();

} else if (e.touches.length === 2 && isPinching) {

// 双指缩放

const touch1 = e.touches[0];

const touch2 = e.touches[1];

const currentDistance = getTouchDistance(touch1, touch2);

if (initialDistance > 0) {

const newScale = Math.min(Math.max(

initialScale \* (currentDistance / initialDistance),

0.3

), 4);

scale = newScale;

updateTransform();

}

}

e.preventDefault();

}, { passive: false });

mermaidElement.addEventListener('touchend', (e) => {

// 重置状态

if (e.touches.length === 0) {

isDragging = false;

isPinching = false;

initialDistance = 0;

// 延迟重置isTouch，避免鼠标事件立即触发

setTimeout(() => {

isTouch = false;

}, 100);

} else if (e.touches.length === 1 && isPinching) {

// 从双指变为单指，切换为拖动模式

isPinching = false;

isDragging = true;

const touch = e.touches[0];

startX = touch.clientX - translateX;

startY = touch.clientY - translateY;

}

updateTransform();

});

mermaidElement.addEventListener('touchcancel', (e) => {

isDragging = false;

isPinching = false;

initialDistance = 0;

setTimeout(() => {

isTouch = false;

}, 100);

updateTransform();

});

// Enhanced wheel zoom with better center point handling

container.addEventListener('wheel', (e) => {

e.preventDefault();

const rect = container.getBoundingClientRect();

const centerX = rect.width / 2;

const centerY = rect.height / 2;

const delta = e.deltaY > 0 ? 0.9 : 1.1;

const newScale = Math.min(Math.max(scale \* delta, 0.3), 4);

// Adjust translation to zoom towards center

if (newScale !== scale) {

const scaleDiff = newScale / scale;

translateX = translateX \* scaleDiff;

translateY = translateY \* scaleDiff;

scale = newScale;

if (scale <= 1) {

translateX = 0;

translateY = 0;

}

updateTransform();

}

});

// Initialize display

updateTransform();

});

}

// Initialize Mermaid with enhanced configuration for better display

mermaid.initialize({

startOnLoad: true,

theme: 'base',

themeVariables: {

primaryColor: '#ffffff',

primaryTextColor: '#1e293b',

primaryBorderColor: '#1e293b',

lineColor: '#64748b',

secondaryColor: '#f1f5f9',

tertiaryColor: '#e2e8f0',

background: '#ffffff',

mainBkg: '#ffffff',

secondBkg: '#f8fafc',

tertiaryBkg: '#f1f5f9',

// Enhanced contrast colors for different node types

cScale0: '#ffffff',

cScale1: '#e3f2fd',

cScale2: '#f3e5f5',

cScale3: '#e8f5e8',

cScale4: '#fff3e0',

cScale5: '#fce4ec',

// Text colors for better readability

cScaleLabel0: '#1e293b',

cScaleLabel1: '#1e293b',

cScaleLabel2: '#1e293b',

cScaleLabel3: '#1e293b',

cScaleLabel4: '#1e293b',

cScaleLabel5: '#1e293b',

// Additional contrast improvements

nodeBorder: '#1e293b',

clusterBkg: '#f8fafc',

clusterBorder: '#64748b',

defaultLinkColor: '#64748b',

titleColor: '#1e293b',

edgeLabelBackground: '#ffffff',

// Timeline specific colors

cScale6: '#f8fafc',

cScale7: '#f1f5f9',

cScale8: '#e2e8f0',

cScale9: '#cbd5e1'

},

flowchart: {

useMaxWidth: false,

htmlLabels: true,

curve: 'basis',

padding: 20

},

timeline: {

useMaxWidth: false,

padding: 20

},

fontFamily: 'Inter, sans-serif',

fontSize: 14,

// Ensure proper sizing and centering

gantt: {

useMaxWidth: false

}

});

// Initialize Mermaid Controls for zoom and pan

function initializeMermaidControls() {

const containers = document.querySelectorAll('.mermaid-container');

containers.forEach(container => {

const mermaidElement = container.querySelector('.mermaid');

let scale = 1;

let isDragging = false;

let startX, startY, translateX = 0, translateY = 0;

// 触摸相关状态

let isTouch = false;

let touchStartTime = 0;

let initialDistance = 0;

let initialScale = 1;

let isPinching = false;

// Zoom controls

const zoomInBtn = container.querySelector('.zoom-in');

const zoomOutBtn = container.querySelector('.zoom-out');

const resetBtn = container.querySelector('.reset-zoom');

const fullscreenBtn = container.querySelector('.fullscreen');

function updateTransform() {

mermaidElement.style.transform = `translate(${translateX}px, ${translateY}px) scale(${scale})`;

if (scale > 1) {

container.classList.add('zoomed');

} else {

container.classList.remove('zoomed');

}

mermaidElement.style.cursor = isDragging ? 'grabbing' : 'grab';

}

if (zoomInBtn) {

zoomInBtn.addEventListener('click', () => {

scale = Math.min(scale \* 1.25, 4);

updateTransform();

});

}

if (zoomOutBtn) {

zoomOutBtn.addEventListener('click', () => {

scale = Math.max(scale / 1.25, 0.3);

if (scale <= 1) {

translateX = 0;

translateY = 0;

}

updateTransform();

});

}

if (resetBtn) {

resetBtn.addEventListener('click', () => {

scale = 1;

translateX = 0;

translateY = 0;

updateTransform();

});

}

if (fullscreenBtn) {

fullscreenBtn.addEventListener('click', () => {

if (container.requestFullscreen) {

container.requestFullscreen();

} else if (container.webkitRequestFullscreen) {

container.webkitRequestFullscreen();

} else if (container.msRequestFullscreen) {

container.msRequestFullscreen();

}

});

}

// Mouse Events

mermaidElement.addEventListener('mousedown', (e) => {

if (isTouch) return; // 如果是触摸设备，忽略鼠标事件

isDragging = true;

startX = e.clientX - translateX;

startY = e.clientY - translateY;

mermaidElement.style.cursor = 'grabbing';

updateTransform();

e.preventDefault();

});

document.addEventListener('mousemove', (e) => {

if (isDragging && !isTouch) {

translateX = e.clientX - startX;

translateY = e.clientY - startY;

updateTransform();

}

});

document.addEventListener('mouseup', () => {

if (isDragging && !isTouch) {

isDragging = false;

mermaidElement.style.cursor = 'grab';

updateTransform();

}

});

document.addEventListener('mouseleave', () => {

if (isDragging && !isTouch) {

isDragging = false;

mermaidElement.style.cursor = 'grab';

updateTransform();

}

});

// 获取两点之间的距离

function getTouchDistance(touch1, touch2) {

return Math.hypot(

touch2.clientX - touch1.clientX,

touch2.clientY - touch1.clientY

);

}

// Touch Events - 触摸事件处理

mermaidElement.addEventListener('touchstart', (e) => {

isTouch = true;

touchStartTime = Date.now();

if (e.touches.length === 1) {

// 单指拖动

isPinching = false;

isDragging = true;

const touch = e.touches[0];

startX = touch.clientX - translateX;

startY = touch.clientY - translateY;

} else if (e.touches.length === 2) {

// 双指缩放

isPinching = true;

isDragging = false;

const touch1 = e.touches[0];

const touch2 = e.touches[1];

initialDistance = getTouchDistance(touch1, touch2);

initialScale = scale;

}

e.preventDefault();

}, { passive: false });

mermaidElement.addEventListener('touchmove', (e) => {

if (e.touches.length === 1 && isDragging && !isPinching) {

// 单指拖动

const touch = e.touches[0];

translateX = touch.clientX - startX;

translateY = touch.clientY - startY;

updateTransform();

} else if (e.touches.length === 2 && isPinching) {

// 双指缩放

const touch1 = e.touches[0];

const touch2 = e.touches[1];

const currentDistance = getTouchDistance(touch1, touch2);

if (initialDistance > 0) {

const newScale = Math.min(Math.max(

initialScale \* (currentDistance / initialDistance),

0.3

), 4);

scale = newScale;

updateTransform();

}

}

e.preventDefault();

}, { passive: false });

mermaidElement.addEventListener('touchend', (e) => {

// 重置状态

if (e.touches.length === 0) {

isDragging = false;

isPinching = false;

initialDistance = 0;

// 延迟重置isTouch，避免鼠标事件立即触发

setTimeout(() => {

isTouch = false;

}, 100);

} else if (e.touches.length === 1 && isPinching) {

// 从双指变为单指，切换为拖动模式

isPinching = false;

isDragging = true;

const touch = e.touches[0];

startX = touch.clientX - translateX;

startY = touch.clientY - translateY;

}

updateTransform();

});

mermaidElement.addEventListener('touchcancel', (e) => {

isDragging = false;

isPinching = false;

initialDistance = 0;

setTimeout(() => {

isTouch = false;

}, 100);

updateTransform();

});

// Enhanced wheel zoom with better center point handling

container.addEventListener('wheel', (e) => {

e.preventDefault();

const rect = container.getBoundingClientRect();

const centerX = rect.width / 2;

const centerY = rect.height / 2;

const delta = e.deltaY > 0 ? 0.9 : 1.1;

const newScale = Math.min(Math.max(scale \* delta, 0.3), 4);

// Adjust translation to zoom towards center

if (newScale !== scale) {

const scaleDiff = newScale / scale;

translateX = translateX \* scaleDiff;

translateY = translateY \* scaleDiff;

scale = newScale;

if (scale <= 1) {

translateX = 0;

translateY = 0;

}

updateTransform();

}

});

// Initialize display

updateTransform();

});

}

// Initialize Mermaid with enhanced configuration for better display

mermaid.initialize({

startOnLoad: true,

theme: 'base',

themeVariables: {

primaryColor: '#ffffff',

primaryTextColor: '#1e293b',

primaryBorderColor: '#1e293b',

lineColor: '#64748b',

secondaryColor: '#f1f5f9',

tertiaryColor: '#e2e8f0',

background: '#ffffff',

mainBkg: '#ffffff',

secondBkg: '#f8fafc',

tertiaryBkg: '#f1f5f9',

// Enhanced contrast colors for different node types

cScale0: '#ffffff',

cScale1: '#e3f2fd',

cScale2: '#f3e5f5',

cScale3: '#e8f5e8',

cScale4: '#fff3e0',

cScale5: '#fce4ec',

// Text colors for better readability

cScaleLabel0: '#1e293b',

cScaleLabel1: '#1e293b',

cScaleLabel2: '#1e293b',

cScaleLabel3: '#1e293b',

cScaleLabel4: '#1e293b',

cScaleLabel5: '#1e293b',

// Additional contrast improvements

nodeBorder: '#1e293b',

clusterBkg: '#f8fafc',

clusterBorder: '#64748b',

defaultLinkColor: '#64748b',

titleColor: '#1e293b',

edgeLabelBackground: '#ffffff',

// Timeline specific colors

cScale6: '#f8fafc',

cScale7: '#f1f5f9',

cScale8: '#e2e8f0',

cScale9: '#cbd5e1'

},

flowchart: {

useMaxWidth: false,

htmlLabels: true,

curve: 'basis',

padding: 20

},

timeline: {

useMaxWidth: false,

padding: 20

},

fontFamily: 'Inter, sans-serif',

fontSize: 14,

// Ensure proper sizing and centering

gantt: {

useMaxWidth: false

}

});

// Initialize Mermaid Controls for zoom and pan

function initializeMermaidControls() {

const containers = document.querySelectorAll('.mermaid-container');

containers.forEach(container => {

const mermaidElement = container.querySelector('.mermaid');

let scale = 1;

let isDragging = false;

let startX, startY, translateX = 0, translateY = 0;

// 触摸相关状态

let isTouch = false;

let touchStartTime = 0;

let initialDistance = 0;

let initialScale = 1;

let isPinching = false;

// Zoom controls

const zoomInBtn = container.querySelector('.zoom-in');

const zoomOutBtn = container.querySelector('.zoom-out');

const resetBtn = container.querySelector('.reset-zoom');

const fullscreenBtn = container.querySelector('.fullscreen');

function updateTransform() {

mermaidElement.style.transform = `translate(${translateX}px, ${translateY}px) scale(${scale})`;

if (scale > 1) {

container.classList.add('zoomed');

} else {

container.classList.remove('zoomed');

}

mermaidElement.style.cursor = isDragging ? 'grabbing' : 'grab';

}

if (zoomInBtn) {

zoomInBtn.addEventListener('click', () => {

scale = Math.min(scale \* 1.25, 4);

updateTransform();

});

}

if (zoomOutBtn) {

zoomOutBtn.addEventListener('click', () => {

scale = Math.max(scale / 1.25, 0.3);

if (scale <= 1) {

translateX = 0;

translateY = 0;

}

updateTransform();

});

}

if (resetBtn) {

resetBtn.addEventListener('click', () => {

scale = 1;

translateX = 0;

translateY = 0;

updateTransform();

});

}

if (fullscreenBtn) {

fullscreenBtn.addEventListener('click', () => {

if (container.requestFullscreen) {

container.requestFullscreen();

} else if (container.webkitRequestFullscreen) {

container.webkitRequestFullscreen();

} else if (container.msRequestFullscreen) {

container.msRequestFullscreen();

}

});

}

// Mouse Events

mermaidElement.addEventListener('mousedown', (e) => {

if (isTouch) return; // 如果是触摸设备，忽略鼠标事件

isDragging = true;

startX = e.clientX - translateX;

startY = e.clientY - translateY;

mermaidElement.style.cursor = 'grabbing';

updateTransform();

e.preventDefault();

});

document.addEventListener('mousemove', (e) => {

if (isDragging && !isTouch) {

translateX = e.clientX - startX;

translateY = e.clientY - startY;

updateTransform();

}

});

document.addEventListener('mouseup', () => {

if (isDragging && !isTouch) {

isDragging = false;

mermaidElement.style.cursor = 'grab';

updateTransform();

}

});

document.addEventListener('mouseleave', () => {

if (isDragging && !isTouch) {

isDragging = false;

mermaidElement.style.cursor = 'grab';

updateTransform();

}

});

// 获取两点之间的距离

function getTouchDistance(touch1, touch2) {

return Math.hypot(

touch2.clientX - touch1.clientX,

touch2.clientY - touch1.clientY

);

}

// Touch Events - 触摸事件处理

mermaidElement.addEventListener('touchstart', (e) => {

isTouch = true;

touchStartTime = Date.now();

if (e.touches.length === 1) {

// 单指拖动

isPinching = false;

isDragging = true;

const touch = e.touches[0];

startX = touch.clientX - translateX;

startY = touch.clientY - translateY;

} else if (e.touches.length === 2) {

// 双指缩放

isPinching = true;

isDragging = false;

const touch1 = e.touches[0];

const touch2 = e.touches[1];

initialDistance = getTouchDistance(touch1, touch2);

initialScale = scale;

}

e.preventDefault();

}, { passive: false });

mermaidElement.addEventListener('touchmove', (e) => {

if (e.touches.length === 1 && isDragging && !isPinching) {

// 单指拖动

const touch = e.touches[0];

translateX = touch.clientX - startX;

translateY = touch.clientY - startY;

updateTransform();

} else if (e.touches.length === 2 && isPinching) {

// 双指缩放

const touch1 = e.touches[0];

const touch2 = e.touches[1];

const currentDistance = getTouchDistance(touch1, touch2);

if (initialDistance > 0) {

const newScale = Math.min(Math.max(

initialScale \* (currentDistance / initialDistance),

0.3

), 4);

scale = newScale;

updateTransform();

}

}

e.preventDefault();

}, { passive: false });

mermaidElement.addEventListener('touchend', (e) => {

// 重置状态

if (e.touches.length === 0) {

isDragging = false;

isPinching = false;

initialDistance = 0;

// 延迟重置isTouch，避免鼠标事件立即触发

setTimeout(() => {

isTouch = false;

}, 100);

} else if (e.touches.length === 1 && isPinching) {

// 从双指变为单指，切换为拖动模式

isPinching = false;

isDragging = true;

const touch = e.touches[0];

startX = touch.clientX - translateX;

startY = touch.clientY - translateY;

}

updateTransform();

});

mermaidElement.addEventListener('touchcancel', (e) => {

isDragging = false;

isPinching = false;

initialDistance = 0;

setTimeout(() => {

isTouch = false;

}, 100);

updateTransform();

});

// Enhanced wheel zoom with better center point handling

container.addEventListener('wheel', (e) => {

e.preventDefault();

const rect = container.getBoundingClientRect();

const centerX = rect.width / 2;

const centerY = rect.height / 2;

const delta = e.deltaY > 0 ? 0.9 : 1.1;

const newScale = Math.min(Math.max(scale \* delta, 0.3), 4);

// Adjust translation to zoom towards center

if (newScale !== scale) {

const scaleDiff = newScale / scale;

translateX = translateX \* scaleDiff;

translateY = translateY \* scaleDiff;

scale = newScale;

if (scale <= 1) {

translateX = 0;

translateY = 0;

}

updateTransform();

}

});

// Initialize display

updateTransform();

});

}

// Initialize Mermaid with enhanced configuration for better display

mermaid.initialize({

startOnLoad: true,

theme: 'base',

themeVariables: {

primaryColor: '#ffffff',

primaryTextColor: '#1e293b',

primaryBorderColor: '#1e293b',

lineColor: '#64748b',

secondaryColor: '#f1f5f9',

tertiaryColor: '#e2e8f0',

background: '#ffffff',

mainBkg: '#ffffff',

secondBkg: '#f8fafc',

tertiaryBkg: '#f1f5f9',

// Enhanced contrast colors for different node types

cScale0: '#ffffff',

cScale1: '#e3f2fd',

cScale2: '#f3e5f5',

cScale3: '#e8f5e8',

cScale4: '#fff3e0',

cScale5: '#fce4ec',

// Text colors for better readability

cScaleLabel0: '#1e293b',

cScaleLabel1: '#1e293b',

cScaleLabel2: '#1e293b',

cScaleLabel3: '#1e293b',

cScaleLabel4: '#1e293b',

cScaleLabel5: '#1e293b',

// Additional contrast improvements

nodeBorder: '#1e293b',

clusterBkg: '#f8fafc',

clusterBorder: '#64748b',

defaultLinkColor: '#64748b',

titleColor: '#1e293b',

edgeLabelBackground: '#ffffff',

// Timeline specific colors

cScale6: '#f8fafc',

cScale7: '#f1f5f9',

cScale8: '#e2e8f0',

cScale9: '#cbd5e1'

},

flowchart: {

useMaxWidth: false,

htmlLabels: true,

curve: 'basis',

padding: 20

},

timeline: {

useMaxWidth: false,

padding: 20

},

fontFamily: 'Inter, sans-serif',

fontSize: 14,

// Ensure proper sizing and centering

gantt: {

useMaxWidth: false

}

});

// Initialize Mermaid Controls for zoom and pan

function initializeMermaidControls() {

const containers = document.querySelectorAll('.mermaid-container');

containers.forEach(container => {

const mermaidElement = container.querySelector('.mermaid');

let scale = 1;

let isDragging = false;

let startX, startY, translateX = 0, translateY = 0;

// 触摸相关状态

let isTouch = false;

let touchStartTime = 0;

let initialDistance = 0;

let initialScale = 1;

let isPinching = false;

// Zoom controls

const zoomInBtn = container.querySelector('.zoom-in');

const zoomOutBtn = container.querySelector('.zoom-out');

const resetBtn = container.querySelector('.reset-zoom');

const fullscreenBtn = container.querySelector('.fullscreen');

function updateTransform() {

mermaidElement.style.transform = `translate(${translateX}px, ${translateY}px) scale(${scale})`;

if (scale > 1) {

container.classList.add('zoomed');

} else {

container.classList.remove('zoomed');

}

mermaidElement.style.cursor = isDragging ? 'grabbing' : 'grab';

}

if (zoomInBtn) {

zoomInBtn.addEventListener('click', () => {

scale = Math.min(scale \* 1.25, 4);

updateTransform();

});

}

if (zoomOutBtn) {

zoomOutBtn.addEventListener('click', () => {

scale = Math.max(scale / 1.25, 0.3);

if (scale <= 1) {

translateX = 0;

translateY = 0;

}

updateTransform();

});

}

if (resetBtn) {

resetBtn.addEventListener('click', () => {

scale = 1;

translateX = 0;

translateY = 0;

updateTransform();

});

}

if (fullscreenBtn) {

fullscreenBtn.addEventListener('click', () => {

if (container.requestFullscreen) {

container.requestFullscreen();

} else if (container.webkitRequestFullscreen) {

container.webkitRequestFullscreen();

} else if (container.msRequestFullscreen) {

container.msRequestFullscreen();

}

});

}

// Mouse Events

mermaidElement.addEventListener('mousedown', (e) => {

if (isTouch) return; // 如果是触摸设备，忽略鼠标事件

isDragging = true;

startX = e.clientX - translateX;

startY = e.clientY - translateY;

mermaidElement.style.cursor = 'grabbing';

updateTransform();

e.preventDefault();

});

document.addEventListener('mousemove', (e) => {

if極isDragging && !isTouch) {

translateX = e.clientX - startX;

translateY = e.clientY - startY;

updateTransform();

}

});

document.addEventListener('mouseup', () => {

if (isDragging && !isTouch) {

isDragging = false;

mermaidElement.style.cursor = 'grab';

updateTransform();

}

});

document.addEventListener('mouseleave', () => {

if (isDragging && !isTouch) {

isDragging = false;

mermaidElement.style.cursor = 'grab';

updateTransform();

}

});

// 获取两点之间的距离

function getTouchDistance(touch1, touch2) {

return Math.hypot(

touch2.clientX - touch1.clientX,

touch2.clientY - touch1.clientY

);

}

// Touch Events - 触摸事件处理

mermaidElement.addEventListener('touchstart', (e) => {

isTouch = true;

touchStartTime = Date.now();

if (e.touches.length === 1) {

// 单指拖动

isPinching = false;

isDragging = true;

const touch = e.touches[0];

startX = touch.clientX - translateX;

startY = touch.clientY - translateY;

} else if (e.touches.length === 2) {

// 双指缩放

isPinching = true;

isDragging = false;

const touch1 = e.touches[0];

const touch2 = e.touches[1];

initialDistance = getTouchDistance(touch1, touch2);

initialScale = scale;

}

e.preventDefault();

}, { passive: false });

mermaidElement.addEventListener('touchmove', (e) => {

if (e.touches.length === 1 && isDragging && !isPinching) {

// 单指拖动

const touch = e.touches[0];

translateX = touch.clientX - startX;

translateY = touch.clientY - startY;

updateTransform();

} else if (e.touches.length === 2 && isPinching) {

// 双指缩放

const touch1 = e.touches[0];

const touch2 = e.touches[1];

const currentDistance = getTouchDistance(touch1, touch2);

if (initialDistance > 0) {

const newScale = Math.min(Math.max(

initialScale \* (currentDistance / initialDistance),

0.3

), 4);

scale = newScale;

updateTransform();

}

}

e.preventDefault();

}, { passive: false });

mermaidElement.addEventListener('touchend', (e) => {

// 重置状态

if (e.touches.length === 0) {

isDragging = false;

isPinching = false;

initialDistance = 0;

// 延迟重置isTouch，避免鼠标事件立即触发

setTimeout(() => {

isTouch = false;

}, 100);

} else if (e.touches.length === 1 && isPinching) {

// 从双指变为单指，切换为拖动模式

isPinching = false;

isDragging = true;

const touch = e.touches[0];

startX = touch.clientX - translateX;

startY = touch.clientY - translateY;

}

updateTransform();

});

mermaidElement.addEventListener('touchcancel', (e) => {

isDragging = false;

isPinching = false;

initialDistance = 0;

setTimeout(() => {

isTouch = false;

}, 100);

updateTransform();

});

// Enhanced wheel zoom with better center point handling

container.addEventListener('wheel', (e) => {

e.preventDefault();

const rect = container.getBoundingClientRect();

const centerX = rect.width / 2;

const centerY = rect.height / 2;

const delta = e.deltaY > 0 ? 0.9 : 1.1;

const newScale = Math.min(Math.max(scale \* delta, 0.3), 4);

// Adjust translation to zoom towards center

if (newScale !== scale) {

const scaleDiff = newScale / scale;

translateX = translateX \* scaleDiff;

translateY = translateY \* scaleDiff;

scale = newScale;

if (scale <= 1) {

translateX = 0;

translateY = 0;

}

updateTransform();

}

});

// Initialize display

updateTransform();

});

}

// Smooth scrolling for navigation links

document.querySelectorAll('a[href^="#"]').forEach(anchor => {

anchor.addEventListener('click', function (e) {

e.preventDefault();

const target = document.querySelector(this.getAttribute('href'));

if (target) {

target.scrollIntoView({

behavior: 'smooth',

block: 'start'

});

}

});

});

// Highlight active section in navigation

const sections = document.querySelectorAll('section[id]');

const navLinks = document.querySelectorAll('nav a[href^="#"]');

function highlightActiveSection() {

let current = '';

sections.forEach(section => {

const sectionTop = section.offsetTop;

const sectionHeight = section.clientHeight;

if (pageYOffset >= sectionTop - 200) {

current = section.getAttribute('id');

}

});

navLinks.forEach(link => {

link.classList.remove('bg-accent', 'text-white');

if (link.getAttribute('href') === `#${current}`) {

link.classList.add('bg-accent', 'text-white');

}

});

}

window.addEventListener('scroll', highlightActiveSection);

highlightActiveSection(); // Initial call

</script>

</body>

</html>