SOLID.AI Framework

A Formal Specification for Strategic, Organized, Layered, Intelligent, Data-Driven Artificial Intelligence

Whitepaper v1.0 — Stable

Published: November 2025

A comprehensive architectural specification for building AI-native organizations that scale human intelligence through structured collaboration between people and artificial intelligence. This whitepaper provides the complete technical specification, architectural patterns, and governance principles required to implement production-ready AI-native systems.

Abstract

Status: Version: 1.0

Citation

If you use SOLID.AI in your research or project, please cite:

```
@techreport{freitas2025solidai,
  title        = {SOLID.AI: Strategic Organization Leveraging Intelligent
                  Design for Artificial Intelligence},
  author       = {Freitas, Gustavo},
  year         = {2025},
  month        = {November},
  institution  = {Midora Education Labs},
  type         = {Technical Whitepaper},
  version      = {1.0.0},
  url          = {https://gusafr.github.io/midora-solid-ai/whitepaper/}
}
```

APA Style:

> *Freitas, G. (2025). SOLID.AI: Strategic organization leveraging intelligent design for artificial intelligence (Version 1.0.0). Midora Education Labs. https://gusafr.github.io/midora-solid-ai/whitepaper/*

Chicago Style:

> *Freitas, Gustavo. 2025. "SOLID.AI: Strategic Organization Leveraging Intelligent Design for Artificial Intelligence." Version 1.0.0. Technical Whitepaper. Midora Education Labs.*

IEEE Style:

> *G. Freitas, "SOLID.AI: Strategic Organization Leveraging Intelligent Design for Artificial Intelligence," Midora Education Labs, Tech. Rep., Nov. 2025, ver. 1.0.0.*

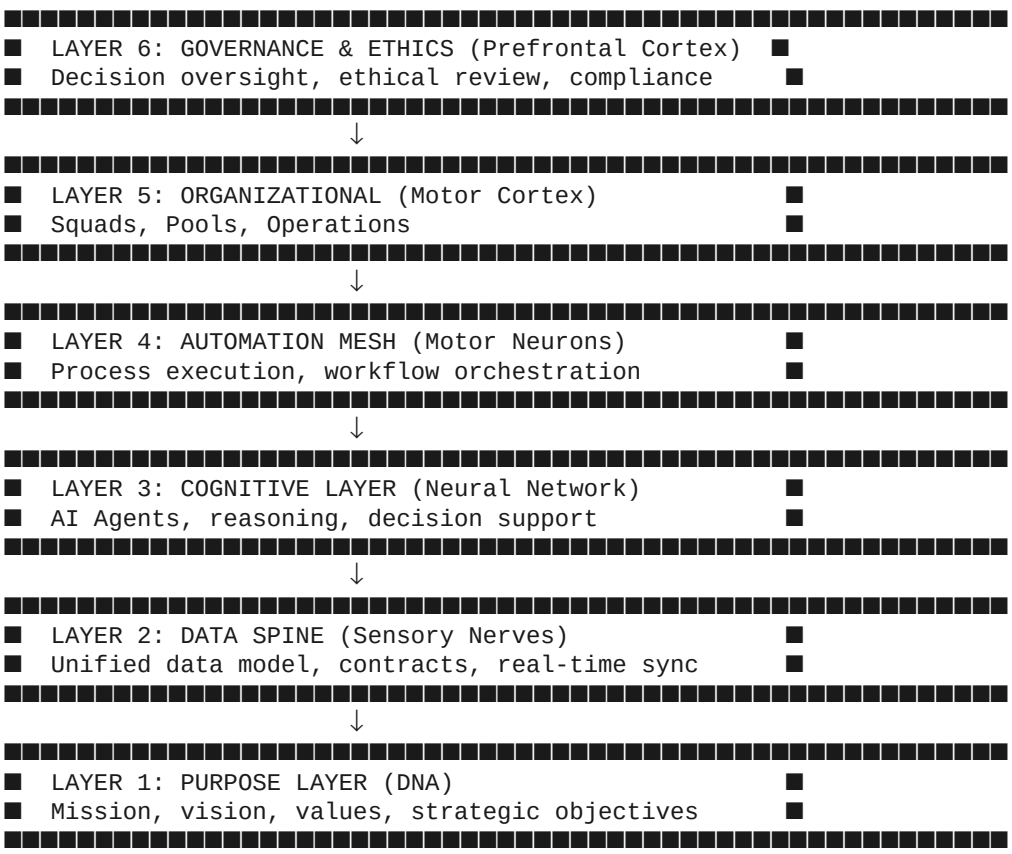
Navigation: Architecture →

Architecture

Status: Version: 1.0

Six-Layer Architecture

As shown in Figure 1 (see Diagrams), SOLID.AI employs a biological-inspired architecture analogous to an organizational nervous system:



***See:** Interactive Layer Model Diagram for detailed visualization*

Layer 1: Purpose Layer (DNA)

Biological Analogy: DNA encoding the organism's fundamental blueprint

Function: Defines the organization's immutable core identity and strategic direction

Components:

- Mission statement
- Vision and strategic goals
- Core values and principles
- Success metrics (OKRs)
- Ethical boundaries

Key Characteristics:

- Rarely changes (only through formal RFC process)
- Informs all decisions across layers
- Accessible to all humans and AI agents
- Machine-readable format (YAML/JSON)

Layer 2: Data Spine (Sensory Nerves)

Biological Analogy: Sensory nervous system transmitting information to the brain

Function: Unified, real-time data infrastructure serving as single source of truth

As shown in Figure 3 (see Diagrams), the Data Spine provides <5 second latency with 99.9% uptime SLA.

Components:

- Canonical data models
- Data contracts between systems
- Event-driven synchronization
- Data quality monitoring
- Analytics and metrics dashboards

Key Characteristics:

- Schema-first design with strict contracts
- Real-time propagation (<5 second latency)
- Immutable event logs (audit trail)
- Bi-directional sync across all systems

See: Specification → Data Spine | Data Spine Topology Diagram

Layer 3: Cognitive Layer (Neural Network)

Biological Analogy: Brain processing information and generating insights

Function: AI agents providing reasoning, decision support, and autonomous actions

Components:

- AI Agent definitions (capabilities, constraints, interfaces)
- Reasoning engines (LLM orchestration)
- Context management (memory, session state)
- Decision logs (transparency)

Agent Types:

- **Analytical Agents:** Data analysis, pattern recognition, forecasting
- **Operational Agents:** Process execution, workflow orchestration
- **Advisory Agents:** Strategic recommendations, risk assessment
- **Collaborative Agents:** Team coordination, meeting facilitation

See: Specification → Cognitive Layer

Layer 4: Automation Mesh (Motor Neurons)

Biological Analogy: Motor nervous system executing coordinated movements

Function: Process execution layer translating decisions into actions

As shown in Figure 2 (see Diagrams), the Automation Mesh coordinates all AI-driven actions through event-driven orchestration connecting agents, business services, and external systems.

Components:

- SIPOC process definitions
- Workflow orchestration (temporal.io, Airflow)
- Integration adapters (APIs, webhooks)

- Monitoring and observability

Key Patterns:

- **SIPOC Automation:** Supplier → Input → Process → Output → Customer
- **Event-Driven Workflows:** Trigger → Validate → Execute → Verify
- **Human-in-the-Loop:** Approval gates for critical decisions

See: Specification → Automation Mesh | Automation Mesh Diagram

Layer 5: Organizational Layer (Motor Cortex)

Biological Analogy: Motor cortex coordinating complex movements

Function: Human team structures optimized for AI-native collaboration

Organizational Patterns:

1. Squads

- **Purpose:** Cross-functional product/feature teams
- **Size:** 5-9 people (Dunbar's limit for tight collaboration)
- **Structure:** Product Manager, Engineers, Designer, Data Analyst
- **AI Integration:** Embedded agents for specific squad functions
- **Ownership:** Business service accountability (P&L responsibility)
- **Lifecycle:** Persistent teams aligned to long-term product areas

2. Pools

- **Purpose:** Flexible specialist communities supporting multiple squads
- **Examples:** Data Science Pool, Security Pool, UX Research Pool
- **Model:** Pull-based engagement (squads request support)
- **AI Integration:** Pool-specific specialized agents
- **Governance:** Community lead coordinates allocation

3. Operations

- **Purpose:** Stable, repeatable business processes
- **Examples:** Payroll, Compliance, Customer Support

- **Model:** High automation (80%+ AI-driven)
- **Human Role:** Exception handling, oversight, continuous improvement
- **Metrics:** Throughput, error rate, cycle time

See: Specification → Organizational Layer

Layer 6: Governance & Ethics (Prefrontal Cortex)

Biological Analogy: Prefrontal cortex providing judgment and ethical reasoning

Function: Decision oversight ensuring alignment with values and compliance

Components:

- RFC (Request for Comments) process for major decisions
- ADR (Architecture Decision Records) documenting choices
- Ethical review board (human + AI advisors)
- Compliance monitoring (SOC2, GDPR, HIPAA, etc.)
- Incident response protocols

Key Mechanisms:

- **Impact Analysis:** Assess risks before changes
- **Approval Workflows:** Tiered authorization based on risk
- **Audit Trails:** Complete decision lineage
- **Feedback Loops:** Retrospectives driving improvement

See: Governance → Implementation

Organizational Scalability Model

SOLID.AI enables exponential growth through AI multiplication:

```
Traditional Organization:
  Revenue: $10M → $50M (+400%)
  Headcount: 100 → 500 people (+400%)
  Ratio: 1:1 scaling

AI-Native Organization (SOLID.AI):
  Revenue: $10M → $50M (+400%)
```

```
Headcount: 100 → 150 people (+50%)
AI Agents: 0 → 350 equivalent roles
Ratio: 1:0.5 scaling (humans), 1:3.5 (AI multiplication)
```

Economic Case:

- **Traditional $50M Company:** 500 employees × $100K = $50M payroll (100% of revenue)
- **AI-Native $50M Company:** 150 employees × $100K = $15M payroll (30% of revenue)
- **Savings:** $35M/year reallocated to R&D, market expansion, or profit
- **Quality:** Error rates <1% (vs. 5-10% traditional), faster time-to-market

Navigation: ← Abstract | Specification → | ■ Diagrams

Specification

Status: Version: 1.0

1. Core Entities

1.1 Actor

Definition: A human participant with decision-making authority and accountability within the system.

Attributes:

- `actor_id`: Unique identifier (UUID)
- `role`: Organizational role (e.g., Product Manager, Compliance Officer)
- `authority_level`: Decision boundary scope (tactical, strategic, governance)
- `authentication_context`: Identity verification state
- `session_metadata`: Active context and preferences

Constraints:

- MUST have unique identity across all system boundaries
- MUST be traceable through audit logs
- MUST operate within defined authority boundaries
- MAY delegate execution to AI Agents but CANNOT delegate accountability

Example:

```
actor:
  actor_id: "a7f3c8b1-4e5d-6f7a-8b9c-0d1e2f3a4b5c"
  role: "Product Manager"
  authority_level: "strategic"
  authentication_context:
    method: "SSO"
    verified_at: "2025-11-29T14:30:00Z"
  session_metadata:
    workspace: "Q4-Planning"
    active_context: ["sales-analysis", "budget-review"]
```

1.3 Event

Definition: A state change or occurrence within the system that triggers downstream processing.

Attributes:

- `event_id`: Unique identifier (UUID)
- `event_type`: Classification (business, system, governance, audit)
- `timestamp`: ISO 8601 timestamp with timezone
- `source`: Originating entity (Actor, AI Agent, External System)
- `payload`: Event data conforming to schema
- `correlation_id`: Parent event or transaction identifier
- `causation_chain`: Full lineage of triggering events

Constraints:

- MUST be immutable after creation
- MUST include complete causation chain
- MUST be persisted to event store
- MUST propagate through Automation Mesh
- MAY trigger zero or more downstream Actions

Example:

```
event:
  event_id: "evt-2025-11-29-14-30-001"
  event_type: "business"
  timestamp: "2025-11-29T14:30:15.234Z"
  source:
    type: "external_system"
    system_id: "salesforce-prod"
  payload:
    event_name: "opportunity_closed_won"
    opportunity_id: "opp-2025-Q4-1234"
    amount: 250000
    customer_id: "cust-enterprise-456"
  correlation_id: "txn-2025-11-29-001"
  causation_chain:
    - "evt-2025-11-29-14-25-001" # opportunity_updated
    - "evt-2025-11-29-14-28-003" # approval_granted
```

1.5 Policy

Definition: A declarative rule that governs system behavior, access control, and decision-making.

Attributes:

- `policy_id`: Unique identifier (UUID)
- `policy_name`: Human-readable name
- `policy_type`: Classification (access*control, approval*workflow, data_governance, compliance)
- `scope`: Applicability (global, domain-specific, agent-specific)
- `conditions`: Logical expressions for policy activation
- `enforcement_action`: Required behavior when policy triggers
- `priority`: Execution order when multiple policies apply

Constraints:

- MUST be versioned
- MUST be auditable
- MUST support conflict resolution via priority
- MAY be overridden by governance layer
- MUST be evaluated before action execution

Example:

```
policy:
  policy_id: "pol-budget-approval-001"
  policy_name: "Budget Allocation Approval Workflow"
  policy_type: "approval_workflow"
  scope:
    domain: "finance"
    applies_to: ["budget-allocation", "cost-center-transfer"]
  conditions:
    - "action.amount > 50000"
    - "action.executor.type == 'ai_agent'"
  enforcement_action:
    type: "require_human_approval"
    approver_roles: ["CFO", "Finance Director"]
    timeout: "4h"
  priority: 100
```

1.7 Data Domain

Definition: A logical grouping of related data entities with consistent governance, ownership, and quality standards.

Attributes:

- `domain_id`: Unique identifier (UUID)

- `domain_name`: Human-readable name
- `owner`: Accountable Actor or team
- `schema_registry`: Data structure definitions
- `quality_requirements`: Validation rules and SLAs
- `access_control`: Authorization policies
- `lineage_tracking`: Data provenance metadata

Constraints:

- MUST have designated owner
- MUST define schema contracts
- MUST enforce quality requirements
- MUST maintain lineage metadata
- MAY federate across multiple storage systems

Example:

```
data_domain:
  domain_id: "dom-sales-performance-001"
  domain_name: "Sales Performance Analytics"
  owner:
    actor_id: "a7f3c8b1-4e5d-6f7a-8b9c-0d1e2f3a4b5c"
    role: "VP Sales Operations"
  schema_registry:
    - entity: "opportunity"
      version: "v2.1"
      fields: ["id", "amount", "stage", "close_date", "probability"]
    - entity: "sales_forecast"
      version: "v1.3"
      fields: ["period", "amount", "confidence", "updated_at"]
  quality_requirements:
    completeness: ">= 0.98"
    freshness: "<= 60s"
    accuracy: ">= 0.95"
  access_control:
    read: ["sales_team", "executive_team", "agent-sales-analyst-*"]
    write: ["salesforce-prod", "sales_automation_agents"]
```

2. System Behaviors

2.1 Event Propagation

Description: The mechanism by which Events flow through the Automation Mesh, triggering downstream Actions and maintaining causation chains.

Behavior Specification:

- **Event Publication**
- Event is created with immutable attributes
- Event is published to Automation Mesh event bus (Kafka topic)
- Event correlation*id and causation*chain are preserved

- **Event Routing**
- Automation Mesh evaluates event_type and payload
- Subscribed AI Agents and services receive event notifications
- Routing respects Boundary constraints

- **Event Processing**
- Consumers process event within SLA targets (P95 < 5s)
- Processing generates new Events and Actions
- Causation chain is extended with new event IDs

- **Event Storage**
- All events persisted to event store (immutable log)
- Events retained per data retention policy
- Events indexed for audit and replay

Guarantees:

- Events are delivered at-least-once
- Event ordering preserved within partition key (correlation_id)
- No event is lost (durability via replication)
- Full causation chain always reconstructable

Example Flow:

```
[CRM System] ---> [Event: opportunity_closed_won]
                          |
                          v
                [Automation Mesh: Kafka Topic]
                          |
                          +---> [AI Agent: Sales Analyst]
                          |           |
                          |           v
                          |     [Action: update_forecast]
                          |
                          +---> [Data Spine: Opportunity Index]
                          |           |
```

```
|          v
|     [Event: forecast_updated]
|
+---> [Governance Layer: Audit Log]
```

2.3 Human Override

Description: The capability for Actors to intervene in automated workflows, overriding AI Agent recommendations or halting in-progress Actions.

Behavior Specification:

- **Override Trigger**
- Actor issues override_request Event
- Request specifies target Action or decision
- Override reason and justification captured
- **Immediate Halt**
- Target Action transitions to "suspended" status
- Downstream Actions blocked
- System state snapshot captured
- **Actor Decision**
- Actor reviews context, data, and AI recommendation
- Actor approves, modifies, or cancels Action
- Decision rationale recorded in audit log
- **Execution Resume**
- System applies Actor's decision
- Workflow continues with modified parameters
- Override Event propagated to audit and governance layers

Guarantees:

- Human override ALWAYS takes precedence over automation
- Override latency < 100ms (real-time responsiveness)
- Full context preserved for Actor decision-making
- Override logged with Actor identity and justification

Example Override:

```
override_event:
  event_id: "evt-override-2025-11-29-001"
  event_type: "governance"
  timestamp: "2025-11-29T15:45:00Z"
  source:
    type: "actor"
    actor_id: "a7f3c8b1-4e5d-6f7a-8b9c-0d1e2f3a4b5c"
  payload:
    override_type: "action_modification"
    target_action_id: "act-budget-allocation-789"
    original_parameters:
      amount: 250000
      allocation: "Q4-marketing-expansion"
    modified_parameters:
      amount: 200000
      allocation: "Q4-marketing-expansion"
    justification: "Market conditions shifted; reducing spend by 20% to preserve cash reser
  result:
    action_status: "resumed"
    modified_execution: true
```

2.5 Audit Trail Registration

Description: The comprehensive logging of all system activities, decisions, and state changes for compliance, debugging, and forensic analysis.

Behavior Specification:

- **Event Capture**
- All Events, Actions, Actor interactions logged
- Logs include full context and causation chain
- Logs signed with cryptographic integrity
- **Structured Storage**
- Audit logs stored in immutable append-only log
- Logs partitioned by domain and time
- Logs replicated for durability (3x replication)
- **Retention Management**
- Logs retained per regulatory requirements (e.g., 7 years for SOX)
- Automated archival to cold storage after hot period
- Legal hold capability for litigation
- **Query and Analysis**

- Audit logs queryable via API
- Full-text search and structured filters
- Anomaly detection via ML models

Guarantees:

- 100% of system activities logged (no gaps)
- Log integrity verifiable via cryptographic signatures
- Log retention meets all compliance requirements
- Logs never modified or deleted (immutable)

Example Audit Entry:

```
audit_entry:
  entry_id: "aud-2025-11-29-14-30-001"
  timestamp: "2025-11-29T14:30:15.234Z"
  entry_type: "action_executed"
  actor_or_agent:
    type: "ai_agent"
    agent_id: "agent-sales-analyst-001"
  action:
    action_id: "act-2025-11-29-14-30-002"
    action_type: "update_forecast"
    target: "revenue-forecasting-service"
  context:
    triggering_event: "evt-opportunity-closed-won"
    correlation_id: "txn-2025-11-29-001"
    causation_chain: ["evt-2025-11-29-14-25-001", "evt-2025-11-29-14-28-003"]
  result:
    status: "completed"
    outcome: "forecast_updated"
    duration_ms: 1234
  integrity:
    signature: "sha256:a3f5d8c9b2e1f4a7..."
    previous_entry_hash: "sha256:9f2e4b7c8a3d1..."
```

3.2 Traceability

Guarantee Statement: Every system output, decision, and state change is traceable to its originating inputs, context, and reasoning chain.

Specification:

- **Full Lineage:** All Events and Actions linked via causation chains
- **Provenance Tracking:** Data transformations preserve origin metadata
- **Decision Explanation:** AI Agents provide reasoning for recommendations
- **Audit Completeness:** 100% of activities logged to immutable audit trail

Implementation Requirements:

- **Causation Chain Propagation:**
- Every Event includes full ancestry
- Actions reference triggering Events
- Chains preserved across system boundaries
- **Explainability:**
- AI Agents output reasoning alongside recommendations
- Reasoning includes data sources, context factors, and logic
- Explanations human-readable and technically precise
- **Audit Coverage:**
- All Actor interactions logged
- All AI Agent decisions logged
- All system Events logged

Example Trace:

```
trace:
  query: "Why did the Q4 forecast increase to $2.45M?"
  trace_result:
    action_id: "act-2025-11-29-14-30-002"
    action_type: "update_forecast"
    executor: "agent-sales-analyst-001"
    triggering_event:
      event_id: "evt-opportunity-closed-won"
      payload:
        opportunity_id: "opp-2025-Q4-1234"
        amount: 250000
    reasoning:
      - "Opportunity opp-2025-Q4-1234 closed won for $250K"
      - "Current Q4 forecast: $2.2M"
      - "Adding $250K to forecast: $2.2M + $250K = $2.45M"
      - "New forecast within target range ($2.3M - $2.7M)"
    data_sources:
      - domain: "sales_performance"
        entity: "opportunities"
        freshness: "12s"
    causation_chain:
      - "evt-2025-11-29-14-25-001" # opportunity_updated
      - "evt-2025-11-29-14-28-003" # approval_granted
      - "evt-2025-11-29-14-30-001" # opportunity_closed_won
```

3.4 Observability Coverage

Guarantee Statement: All system components, behaviors, and performance metrics are observable, measurable, and alertable in real-time.

Specification:

- **Metrics Collection:** Performance, latency, throughput, error rates
- **Logging:** Structured logs for all Events, Actions, and decisions
- **Tracing:** Distributed traces across service boundaries
- **Alerting:** Real-time alerts for SLA violations and anomalies

Implementation Requirements:

- **Metrics Instrumentation:**
- Prometheus metrics for all services
- Custom metrics for AI Agent performance (accuracy, latency)
- SLA tracking (P50, P95, P99 latencies)
- **Distributed Tracing:**
- OpenTelemetry spans for all operations
- Trace IDs propagated across system boundaries
- Trace visualization via Jaeger or Tempo
- **Dashboards:**
- Real-time system health dashboard
- AI Agent performance dashboard
- Compliance and governance dashboard
- **Alerting:**
- SLA breach alerts (e.g., P95 latency > 5s)
- Policy violation alerts
- Anomaly detection via ML models

Key Metrics:

| Metric | Target | Alert Threshold |

|--------|--------|----------------|

| Event Processing Latency (P95) | < 5s | > 10s |

| Data Spine Freshness | < 60s | > 120s |

| AI Agent Accuracy | ≥ 0.95 | < 0.90 |

| System Availability | ≥ 99.9% | < 99.5% |

| Audit Log Completeness | 100% | < 99.9% |

| Policy Evaluation Latency (P95) | < 100ms | > 500ms |

Example Observability Stack:

```
observability:
  metrics:
    collector: "Prometheus"
    retention: "90d"
    scrape_interval: "15s"

  logging:
    system: "ELK Stack (Elasticsearch, Logstash, Kibana)"
    structured_format: "JSON"
    retention: "7y" # SOX compliance

  tracing:
    system: "OpenTelemetry + Jaeger"
    sampling_rate: "100%" # Full trace coverage
    retention: "30d"

  dashboards:
    - name: "System Health"
      url: "/dashboards/system-health"
    - name: "AI Agent Performance"
      url: "/dashboards/ai-agents"
    - name: "Compliance Status"
      url: "/dashboards/compliance"

  alerting:
    system: "PagerDuty"
    channels: ["email", "slack", "sms"]
    escalation_policy: "on-call-rotation"
```

5. References

5.1 Related Specifications

- **Architecture** — Six-layer system design
- **Technical Specification** — Component implementation details
- **Governance** — Implementation roadmap and compliance
- **Diagrams** — Visual architecture references

5.2 External Standards

- **GDPR:** General Data Protection Regulation (EU 2016/679)
- **SOX:** Sarbanes-Oxley Act (2002)
- **HIPAA:** Health Insurance Portability and Accountability Act (1996)
- **ISO 27001:** Information Security Management (2013)
- **FedRAMP:** Federal Risk and Authorization Management Program
- **OpenTelemetry:** Cloud-native observability framework

7. License

This specification is released under the MIT License. See LICENSE for details.

Navigation:

- ← Back to Whitepaper Index
- View Architecture →
- View Technical Specification →

Implementation Guide

Status: Version: 1.0

This section provides detailed technical specifications for each layer of the SOLID.AI architecture.

Cognitive Layer (Layer 3)

Overview

As shown in Figure 4 (see Diagrams), AI agents operate as organizational members with defined roles, capabilities, and accountability through a continuous collaboration loop with humans.

> ***See:*** *Human-AI Collaboration Loop Diagram for complete interaction flow*

Agent Definition Schema

```
agent:
  id: sales_analyst_001
  name: Sales Performance Analyzer
  type: analytical
  version: 2.1.0

  purpose: |
    Analyze sales pipeline data, identify trends, and provide
    actionable recommendations to sales leadership.

  capabilities:
    - pipeline_forecasting
    - deal_risk_assessment
    - win_loss_analysis
    - competitor_intelligence

  data_access:
    read:
      - customers (all)
      - opportunities (all)
      - contracts (all)
      - interactions (type="sales_call")
    write:
      - forecasts (own)
      - recommendations (own)

  interfaces:
    input:
```

```
      - slack_channel: "#sales-analytics"
      - api_endpoint: "/agents/sales_analyst"
      - scheduled_triggers: ["daily 8am", "weekly monday"]
    output:
      - slack_notifications: true
      - dashboard_updates: "sales_dashboard"
      - email_reports: sales_leadership@company.com

  constraints:
    execution_time: <60 seconds
    cost_per_run: <$0.50
    accuracy_threshold: >95%

  human_oversight:
    approval_required: false
    audit_frequency: weekly
    escalation_conditions:
      - forecast_deviation >20%
      - deal_risk_score >8/10

  ethical_boundaries:
    - no_customer_discrimination
    - transparent_scoring_methodology
    - human_review_for_contract_termination
```

Agent Lifecycle

- **Definition:** RFC process for new agents
- **Development:** Build and test in sandbox
- **Validation:** Human review + test cases
- **Deployment:** Gradual rollout with monitoring
- **Operation:** Continuous execution + logging
- **Evolution:** Feedback-driven improvements
- **Retirement:** Deprecation with migration plan

Reasoning Patterns

Chain-of-Thought:

```
User Query: "Why is Q4 forecast down 15%?"

Agent Reasoning:
1. Retrieve Q4 pipeline data
2. Compare to Q3 pipeline at same point
3. Identify closed-lost deals (reasons)
4. Analyze new deal velocity (slower)
5. Assess stage progression rates (delayed)
6. Synthesize findings into explanation

Output: "Q4 forecast is down 15% due to: (1) 3 large
```

```
enterprise deals slipped to Q1 ($450K total), (2) new
pipeline generation 20% below target, and (3) slower
progression from Discovery → Proposal (avg 14 days vs.
9 days in Q3). Recommendation: Focus on accelerating
mid-stage deals and launching Q1 demand gen campaign."
```

Human-AI Collaboration Model:

- AI performs analysis (speed, scale)
- Human validates conclusions (judgment)
- AI implements decisions (execution)
- Human monitors outcomes (oversight)

Organizational Layer (Layer 5)

Squad Specification

Charter Template:

```
squad:
  name: Checkout Experience Squad
  mission: Optimize conversion and revenue at checkout

  business_service:
    name: E-Commerce Checkout
    metrics:
      - conversion_rate (current: 68%, target: 75%)
      - cart_abandonment (current: 32%, target: 25%)
      - revenue_per_session (current: $45, target: $55)

  team:
    product_manager: alice_johnson
    tech_lead: bob_chen
    engineers: [carol_lopez, dave_kumar, eve_taylor]
    designer: frank_williams
    data_analyst: grace_martinez

  ai_agents:
    - checkout_optimizer (A/B test orchestration)
    - fraud_detector (transaction risk scoring)
    - personalization_engine (offer recommendations)

  dependencies:
    upstream:
      - Product Catalog Squad (inventory data)
      - Pricing Squad (promotional rules)
    downstream:
      - Order Fulfillment Squad (order handoff)
      - Customer Support Squad (checkout issues)
```

```
ceremonies:
  sprint_length: 2 weeks
  planning: Monday 9am
  daily_standup: Daily 10am (15 min)
  review: Friday 2pm
  retrospective: Friday 3pm

decision_authority:
  autonomous: [UI changes, A/B tests, bug fixes]
  requires_approval: [pricing strategy, payment provider]
  forbidden: [PCI compliance changes without Security]
```

Navigation: ← Architecture | Governance → | ■ Diagrams

Governance

Status: Version: 1.0

Implementation Methodology

SOLID.AI transformation follows a phased approach balancing speed with organizational change management. As shown in Figure 1 (see Diagrams), the complete architecture is built incrementally, starting with the Data Spine (Figure 3) and Cognitive Layer (Figure 4), then scaling the Automation Mesh (Figure 2) across the organization.

Three-Phase Roadmap

Phase 1: Foundation (Months 1-3)

Objectives:

- Establish Data Spine infrastructure
- Define Purpose Layer (mission, values, OKRs)
- Select pilot business service
- Form first AI-native squad

Deliverables:

- [ ] Canonical data models documented
- [ ] Data contracts between 3+ systems
- [ ] First AI agent deployed (low-risk use case)
- [ ] RFC/ADR governance process established
- [ ] Ethical review board formed

Success Metrics:

- Data Spine operational (99% uptime)
- <5 second data propagation latency
- First agent achieving >90% accuracy

- Zero ethical violations

Pilot Candidates:

- Sales pipeline analysis (low risk, high value)
- Customer support ticket routing
- Invoice processing automation
- Marketing campaign performance analysis

Phase 2: Pilot & Learn (Months 4-9)

Objectives:

- Scale to 3-5 squads across functions
- Deploy 10-15 production AI agents
- Validate organizational patterns
- Refine governance processes

Deliverables:

- [ ] 3 business services AI-native
- [ ] Cross-functional squad coordination proven
- [ ] Agent marketplace established (reusable agents)
- [ ] Observability dashboards operational
- [ ] First retrospective-driven improvements

Success Metrics:

- 50% reduction in cycle time (pilot services)
- 80% automation rate for operational tasks
- Employee satisfaction >4.0/5.0
- Zero compliance incidents

Common Challenges:

- Resistance to change (address with training)
- Data quality issues (invest in cleanup)
- Integration complexity (prioritize key systems)
- Unclear roles (define RACI matrices)

Phase 3: Scale (Months 10-24)

Objectives:

- Whole-organization transformation
- 50+ AI agents in production
- All functions operating AI-native
- Self-sustaining continuous improvement

Deliverables:

- [ ] 100% business services AI-enabled
- [ ] Agent autonomy increasing (80%+ decisions)
- [ ] Organizational scalability demonstrated
- [ ] Documented playbooks for new entrants
- [ ] Open-source contributions to framework

Success Metrics:

- 10x improvement in time-to-market
- Revenue growth without linear headcount scaling
- <1% error rates across processes
- Industry recognition (case studies, awards)

Compliance Management

Regulatory Frameworks

SOLID.AI supports compliance with:

| Framework | Scope | Key Requirements |

|---|---|---|

| **GDPR** | EU data protection | Consent, data minimization, right to erasure |

| **CCPA** | California privacy | Disclosure, opt-out, non-discrimination |

| **SOC 2** | Security controls | Access control, encryption, audit logs |

| **HIPAA** | Healthcare data | PHI protection, access logging, encryption |

| **ISO 27001** | Information security | Risk assessment, incident response |

| **FedRAMP** | US government cloud | Enhanced security controls, continuous monitoring |

Compliance Architecture

Data Classification:

```
data_classification:
  public:
    examples: [marketing_content, blog_posts]
    encryption: optional
    access: all

  internal:
    examples: [roadmaps, financial_models]
    encryption: required
    access: employees_only

  confidential:
    examples: [customer_contracts, employee_salaries]
    encryption: required (AES-256)
    access: role_based
    audit: all_access_logged

  restricted:
    examples: [PHI, PII, financial_transactions]
    encryption: required (AES-256 + tokenization)
    access: explicit_approval
    audit: all_access_logged + reviewed
    retention: auto_delete_after_90_days
```

Agent Compliance Controls:

```
agent: customer_support_assistant
compliance:
  data_access:
    - customer_name (public)
    - email (confidential, masked: j***@example.com)
    - order_history (confidential)
    - payment_info (FORBIDDEN - restricted)

  retention:
    conversation_logs: 90_days
    sensitive_data: 30_days
    audit_trail: 7_years

  encryption:
    in_transit: TLS 1.3
    at_rest: AES-256

  monitoring:
```

```
access_logging: enabled
anomaly_detection: enabled
compliance_alerts: pii_exposure, unauthorized_access
```

Continuous Improvement

Feedback Loops

Agent Performance Review (Weekly):

- Accuracy metrics vs. baseline
- Cost per execution
- User satisfaction ratings
- Error analysis

Squad Retrospective (Biweekly):

- What went well?
- What needs improvement?
- Action items (captured as RFC/ADR)

Organizational Health Check (Quarterly):

- Employee engagement survey
- AI trust metrics
- Ethical incident review
- Scalability assessment

Annual Framework Audit:

- Purpose Layer relevance
- Architecture evolution needs
- Governance effectiveness
- Industry benchmark comparison

Conclusion

SOLID.AI provides the architectural blueprint for building **Intelligent Hybrid Organizations**—enterprises where humans and AI collaborate as peers under ethical governance. The framework is:

■ **Comprehensive:** Six layers covering purpose → execution

■ **Practical:** Battle-tested patterns and templates

■ **Flexible:** Technology-agnostic, adaptable to any industry

■ **Ethical:** Governance and compliance built-in

■ **Open Source:** MIT license, community-driven evolution

The Transformation Imperative:

> *You cannot compete in the AI-native era with a bipolar organization. Whole-organization transformation is not optional—it's existential.*

Next Steps:

- Read the Quick Start Guide
- Assess your AI maturity using the Maturity Model
- Join the community on GitHub
- Start your pilot (Month 1-3)

License

THE FRAMEWORK IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE FRAMEWORK OR THE USE OR OTHER DEALINGS IN THE FRAMEWORK.

Plain Language Summary:

■ **Commercial Use Allowed:** Use SOLID.AI in for-profit organizations

■ **Modification Allowed:** Adapt to your specific needs

■ **Distribution Allowed:** Share with colleagues, clients, partners

■ **Private Use Allowed:** Internal implementation without disclosure

■■ **Attribution Required:** Credit original authors in derivative works

■■ **No Warranty:** Use at your own risk; authors not liable for outcomes

Whitepaper Diagrams

Status: Version: 1.0

1. SOLID.AI Architecture Layer Model

```
graph TB
    subgraph L6["Layer 6: Governance & Ethics"]
        G1[RFC Process]
        G2[Ethical Review]
        G3[Compliance Monitoring]
        G4[Audit Trail]
    end

    subgraph L5["Layer 5: Organizational Layer"]
        O1[Squads<br/>5-9 people]
        O2[Pools<br/>Specialists]
        O3[Operations<br/>Stable processes]
    end

    subgraph L4["Layer 4: Automation Mesh"]
        A1[SIPOC Workflows]
        A2[Process Orchestration]
        A3[Integration Adapters]
        A4[Monitoring]
    end

    subgraph L3["Layer 3: Cognitive Layer"]
        C1[AI Agents]
        C2[Reasoning Engines]
        C3[Context Management]
        C4[Decision Logs]
    end

    subgraph L2["Layer 2: Data Spine"]
        D1[Canonical Models]
        D2[Data Contracts]
        D3[Event Streaming]
        D4[Quality Framework]
    end

    subgraph L1["Layer 1: Purpose Layer"]
        P1[Mission & Vision]
        P2[Core Values]
        P3[Strategic OKRs]
        P4[Ethical Boundaries]
    end

    L6 --> L5
    L5 --> L4
    L4 --> L3
    L3 --> L2
    L2 --> L1
```

```
    L1 -.->|Informs| L2
    L2 -.->|Powers| L3
    L3 -.->|Executes| L4
    L4 -.->|Organizes| L5
    L5 -.->|Overseen by| L6

    style L6 fill:#e8f4f8,stroke:#0d9488,stroke-width:2px
    style L5 fill:#f0fdf4,stroke:#10b981,stroke-width:2px
    style L4 fill:#fef3c7,stroke:#f59e0b,stroke-width:2px
    style L3 fill:#ede9fe,stroke:#8b5cf6,stroke-width:2px
    style L2 fill:#dbeafe,stroke:#3b82f6,stroke-width:2px
    style L1 fill:#fee2e2,stroke:#ef4444,stroke-width:2px
```

<figure markdown>

<figcaption>Figure 1 — SOLID.AI Six-Layer Architecture

Overview of the structural layers from alignment to governance, establishing the foundation for hybrid intelligent organizations. The six-layer architecture creates an organizational nervous system where the Purpose Layer (DNA) defines immutable identity, the Data Spine (sensory nerves) provides real-time information, the Cognitive Layer (brain) generates insights, the Automation Mesh (motor neurons) executes processes, the Organizational Layer (motor cortex) coordinates human teams, and the Governance Layer (prefrontal cortex) ensures ethical oversight.</figcaption>

</figure>

Reference Implementation: See Midora Topology (ADR-0003) for a concrete mapping of this diagram into a real AI-native education platform.

3. SOLID.AI Data Spine Topology

```
graph TB
    subgraph Sources["Data Sources"]
        S1[CRM<br/>Customers]
        S2[ERP<br/>Finance]
        S3[Support<br/>Tickets]
        S4[Product<br/>Usage]
        S5[HR<br/>Employees]
    end

    subgraph Spine["Data Spine Core"]
        subgraph Ingestion["Ingestion Layer"]
            I1[Change Data Capture]
            I2[API Polling]
            I3[Webhook Receivers]
        end

        subgraph Stream["Event Streaming"]
            KAFKA[Apache Kafka<br/>Event Bus]
        end
```

```
        subgraph Models["Canonical Models"]
            M1[Customer Entity]
            M2[Order Entity]
            M3[Ticket Entity]
            M4[Employee Entity]
        end

        subgraph Storage["Storage"]
            DB[(PostgreSQL<br/>Canonical Store)]
            WAREHOUSE[(Data Warehouse<br/>Analytics)]
        end

        subgraph Quality["Data Quality"]
            Q1[Schema Validation]
            Q2[Business Rules]
            Q3[Completeness Check]
        end
    end

    subgraph Consumers["Data Consumers"]
        C1[AI Agents<br/>Cognitive Layer]
        C2[Dashboards<br/>Analytics]
        C3[Workflows<br/>Automation Mesh]
        C4[APIs<br/>External Access]
    end

    Sources -->|Raw Data| Ingestion
    Ingestion -->|Events| KAFKA
    KAFKA -->|Stream| Models
    Models -->|Validate| Quality
    Quality -->|Write| DB
    DB -->|Replicate| WAREHOUSE
    DB -->|Read| Consumers
    WAREHOUSE -->|Query| Consumers

    style Spine fill:#dbeafe,stroke:#3b82f6,stroke-width:3px
    style Stream fill:#e0f2fe,stroke:#0284c7,stroke-width:2px
    style Models fill:#ddd6fe,stroke:#7c3aed,stroke-width:2px
```

<figure markdown>

<figcaption>Figure 3 — Data Spine Domain Model

Unified data backbone enabling clean, derived, and real-time data flows across the organization. The Data Spine ingests data from multiple sources (CRM, ERP, Support, Product, HR) via CDC/APIs/webhooks, streams events through Kafka, maps to canonical entity models, validates quality, stores in PostgreSQL (transactional) and Data Warehouse (analytics), and serves all consumers with <5 second latency and 99.9% uptime SLA. Target SLO: P95 latency < 5s, availability ≥ 99.9%, data freshness < 60s for real-time entities.</figcaption>

</figure>

Reference Implementation: See Midora Topology (ADR-0003) for a concrete mapping of this diagram into a real AI-native education platform.

Diagram Usage Guidelines

In Academic Citations

When referencing these diagrams in papers:

> *"Figure 1 shows the SOLID.AI six-layer architecture (Freitas, 2025), where each layer serves a distinct biological function in the organizational nervous system."*

In Implementation

These diagrams should be used during:

- **Executive presentations** - Use Layer Model to explain transformation scope
- **Technical architecture reviews** - Reference Data Spine and Automation Mesh for infrastructure design
- **Team onboarding** - Show Human-AI Collaboration Loop to clarify roles
- **Vendor evaluations** - Map vendor capabilities to specific layers

Diagram Formats

All diagrams are available in multiple formats:

- **Mermaid (source)** - Editable, version-controlled `.mmd` files
- **SVG (web)** - Rendered automatically in browser, scalable for presentations
- **PNG (print)** - High-resolution exports for documentation and papers
- **PDF (publication)** - Vector format for academic submissions

---

Navigation: ← Governance | Abstract →

Architecture Overview

Status: Version: 1.0

Figure 1 — SOLID.AI Six-Layer Architecture

Overview of the structural layers from alignment to governance, establishing the foundation for hybrid intelligent organizations.

```
graph TB
    subgraph L6["Layer 6: Governance & Ethics"]
        G1[RFC Process]
        G2[Ethical Review]
        G3[Compliance Monitoring]
        G4[Audit Trail]
    end

    subgraph L5["Layer 5: Organizational Layer"]
        O1[Squads<br/>5-9 people]
        O2[Pools<br/>Specialists]
        O3[Operations<br/>Stable processes]
    end

    subgraph L4["Layer 4: Automation Mesh"]
        A1[SIPOC Workflows]
        A2[Process Orchestration]
        A3[Integration Adapters]
        A4[Monitoring]
    end

    subgraph L3["Layer 3: Cognitive Layer"]
        C1[AI Agents]
        C2[Reasoning Engines]
        C3[Context Management]
        C4[Decision Logs]
    end

    subgraph L2["Layer 2: Data Spine"]
        D1[Canonical Models]
        D2[Data Contracts]
        D3[Event Streaming]
        D4[Quality Framework]
    end

    subgraph L1["Layer 1: Purpose Layer"]
        P1[Mission & Vision]
        P2[Core Values]
        P3[Strategic OKRs]
        P4[Ethical Boundaries]
    end

    L6 --> L5
```

```
    L5 --> L4
    L4 --> L3
    L3 --> L2
    L2 --> L1

    L1 -.->|Informs| L2
    L2 -.->|Powers| L3
    L3 -.->|Executes| L4
    L4 -.->|Organizes| L5
    L5 -.->|Overseen by| L6

    style L6 fill:#e8f4f8,stroke:#0d9488,stroke-width:2px
    style L5 fill:#f0fdf4,stroke:#10b981,stroke-width:2px
    style L4 fill:#fef3c7,stroke:#f59e0b,stroke-width:2px
    style L3 fill:#ede9fe,stroke:#8b5cf6,stroke-width:2px
    style L2 fill:#dbeafe,stroke:#3b82f6,stroke-width:2px
    style L1 fill:#fee2e2,stroke:#ef4444,stroke-width:2px
```

<figure markdown>

<figcaption>Figure 1 — SOLID.AI Six-Layer Architecture

The six-layer architecture creates an organizational nervous system where the Purpose Layer (DNA) defines immutable identity, the Data Spine (sensory nerves) provides real-time information, the Cognitive Layer (brain) generates insights, the Automation Mesh (motor neurons) executes processes, the Organizational Layer (motor cortex) coordinates human teams, and the Governance Layer (prefrontal cortex) ensures ethical oversight.</figcaption>

</figure>

Key Insight: Each layer serves a distinct biological function, creating a self-regulating system where strategy flows downward (command) and information flows upward (feedback).

Figure 3 — Data Spine Domain Model

Unified data backbone enabling clean, derived, and real-time data flows across the organization.

```
graph TB
    subgraph Sources["Data Sources"]
        S1[CRM<br/>Customers]
        S2[ERP<br/>Finance]
        S3[Support<br/>Tickets]
        S4[Product<br/>Usage]
        S5[HR<br/>Employees]
    end

    subgraph Spine["Data Spine Core"]
        subgraph Ingestion["Ingestion Layer"]
            I1[Change Data Capture]
            I2[API Polling]
            I3[Webhook Receivers]
```

```
        end

        subgraph Stream["Event Streaming"]
            KAFKA[Apache Kafka<br/>Event Bus]
        end

        subgraph Models["Canonical Models"]
            M1[Customer Entity]
            M2[Order Entity]
            M3[Ticket Entity]
            M4[Employee Entity]
        end

        subgraph Storage["Storage"]
            DB[(PostgreSQL<br/>Canonical Store)]
            WAREHOUSE[(Data Warehouse<br/>Analytics)]
        end

        subgraph Quality["Data Quality"]
            Q1[Schema Validation]
            Q2[Business Rules]
            Q3[Completeness Check]
        end
    end

    subgraph Consumers["Data Consumers"]
        C1[AI Agents<br/>Cognitive Layer]
        C2[Dashboards<br/>Analytics]
        C3[Workflows<br/>Automation Mesh]
        C4[APIs<br/>External Access]
    end

    Sources -->|Raw Data| Ingestion
    Ingestion -->|Events| KAFKA
    KAFKA -->|Stream| Models
    Models -->|Validate| Quality
    Quality -->|Write| DB
    DB -->|Replicate| WAREHOUSE
    DB -->|Read| Consumers
    WAREHOUSE -->|Query| Consumers

    style Spine fill:#dbeafe,stroke:#3b82f6,stroke-width:3px
    style Stream fill:#e0f2fe,stroke:#0284c7,stroke-width:2px
    style Models fill:#ddd6fe,stroke:#7c3aed,stroke-width:2px
```

<figure markdown>

<figcaption>Figure 3 — Data Spine Domain Model

The Data Spine ingests data from multiple sources (CRM, ERP, Support, Product, HR) via CDC/APIs/webhooks, streams events through Kafka, maps to canonical entity models, validates quality, stores in PostgreSQL (transactional) and Data Warehouse (analytics), and serves all consumers with <5 second latency and 99.9% uptime SLA. Target SLO: P95 latency < 5s, availability ≥ 99.9%, data freshness < 60s for real-time entities.</figcaption>

</figure>

Key Insight: The Data Spine eliminates data silos by providing a single, real-time, canonical view of organizational truth, accessible to both humans and AI agents through unified interfaces.

System Integration

These four figures form an integrated architectural view:

- **Figure 1** establishes the **structural foundation** — six layers creating an organizational nervous system
- **Figure 2** details the **execution layer** — how workflows orchestrate across systems
- **Figure 3** reveals the **information backbone** — canonical data flowing in real-time
- **Figure 4** demonstrates the **operational cycle** — humans and AI collaborating through defined control modes

Together, they specify a complete AI-native organization where:

- **Strategy flows down** (Purpose → Data → Cognitive → Automation → Organization → Governance)
- **Feedback flows up** (Execution results inform strategic adjustments)
- **Humans and AI collaborate** at appropriate control points (in-the-loop, on-the-loop, outside-the-loop)
- **All actions are auditable** through immutable event logs and decision trails

Use Cases

This architecture overview is designed for:

| Audience | Use Case | Focus Figures |

|----------|----------|---------------|

| **Executives** | Business transformation roadmap | Figure 1, 4 |

| **Architects** | System design and integration | Figure 2, 3 |

| **Data Engineers** | Data infrastructure planning | Figure 3 |

| **AI Engineers** | Agent deployment and orchestration | Figure 2, 4 |

| **Product Managers** | Feature prioritization and workflows | Figure 1, 2, 4 |

| **Compliance Officers** | Governance and audit requirements | Figure 1, 4 |

Navigation: ← Diagrams | Abstract → | Index