Contents

RADIANT Platform Documentation

Complete System Architecture Reference

Version 5.52.29 | January 2026

EXECUTIVE SUMMARY

RADIANT (Rapid AI Deployment Infrastructure for Applications with Native Tenancy) is a comprehensive multi-tenant AWS SaaS platform providing AI model orchestration and infrastructure services. The platform serves as white-label infrastructure operating invisibly behind customer-facing applications.

Version 5.0 (The Sovereign Mesh) introduces: - **Agent Registry** - Long-running AI agents with OODA loops - **App Registry** - 3,000+ apps auto-synced from Activepieces/n8n - **Parametric AI Helper** - AI assistance configurable per node - **Pre-Flight Provisioning** - Check requirements

before execution - **Transparency Layer** - Full visibility into Cato's decisions - **Enhanced HITL** - First-class approval workflows

---

PART 1: EXISTING ARCHITECTURE (PROMPTS 01-35)

1.1 Infrastructure Foundation (PROMPT-01 through PROMPT-03)

AWS CDK Infrastructure

Component	Description	Status
VPC Stack	Multi-AZ VPC with public/private subnets	Implemented
Database Stack	Aurora PostgreSQL with pgvector	Implemented
Database Scaling Stack	RDS Proxy, Async Writes, Redis Cache	Implemented (v5.52.20)
Cache Stack	ElastiCache Redis cluster	Implemented
Auth Stack	Cognito user pools	Implemented
API Stack	API Gateway + Lambda	Implemented
Storage Stack	S3 buckets for uploads/artifacts	Implemented
Monitoring Stack	CloudWatch dashboards + alarms	Implemented

PostgreSQL Scaling Infrastructure (v5.52.20)

Enterprise-grade scaling for parallel AI model execution supporting 100+ concurrent requests with 6 parallel model writes each.

Component	Purpose	Tier Availability
RDS Proxy	Connection pooling, Lambda cold-start optimization	2+
Async Write Queue	SQS-based batch writes for model results	2+
Redis Hot-Path Cache	Read-after-write consistency, rate limiting	2+
Time-Based Partitioning	Monthly partitions for logs/usage tables	All
Materialized Views	Pre-computed dashboard metrics	All
Optimized RLS	Index-friendly tenant isolation policies	All

CDK Constructs: - `DatabaseScalingConstruct` - RDS Proxy with tier-based connection limits -

`AsyncWriteConstruct` - SQS queue + batch writer Lambda - `RedisCacheConstruct` - ElastiCache cluster with cluster mode

Database Schema (Migrations 001-070)

Table	Purpose	Migration
`tenants`	Multi-tenant isolation	001
`users`	User accounts	002
`api_keys`	API authentication	003
`sessions`	Chat sessions	004
`messages`	Chat messages	005
`ai_providers`	20+ AI providers	007
`ai_models`	106 AI models	007
`usage_records`	Billing/usage	010
`audit_logs`	Compliance audit	015
`mfa_backup_codes`	MFA one-time recovery codes	070
`mfa_trusted_devices`	30-day device trust tokens	070
`mfa_audit_log`	MFA event audit log (partitioned)	070
`*.detected_language`	Auto-detected content language	071
`*.search_vector_simple`	Fallback tsvector for FTS	071
`*.search_vector_english`	Language-specific tsvector	071

Multi-Language Search (Migration 071)

Feature	Implementation
pg_bigm Extension	Bi-gram indexing for CJK languages
Language Detection	`detect_text_language()` function
Unified Search	`search_content()` routes to FTS or bigm
18 Languages	en, es, fr, de, pt, it, nl, pl, ru, tr, ja, ko, zh-CN, zh-TW, ar, hi, th, vi

Swift Deployment Application

Feature	Description
One-Click Deploy	Complete infrastructure in single click
Account Management	AWS account configuration
Environment Selection	Dev/Staging/Prod
Progress Monitoring	Real-time deployment status
Rollback Support	Automatic rollback on failure

1.2 Lambda Functions (PROMPT-04 & PROMPT-05)

Core Lambda Functions

Function	Purpose	Trigger
`auth-handler`	Authentication/authorization	API Gateway
`mfa-handler`	MFA enrollment, verification, device trust	API Gateway
`chat-handler`	Chat completion requests	API Gateway
`stream-handler`	SSE streaming responses	API Gateway
`models-handler`	Model CRUD operations	API Gateway
`providers-handler`	Provider management	API Gateway
`sessions-handler`	Session management	API Gateway
`usage-handler`	Usage reporting	API Gateway

Admin Lambda Functions (62 Total - v5.52.6)

All admin Lambda handlers are wired to `/api/admin/*` routes with Cognito admin authorization.

Category	Count	Handlers
Cato Safety	5	cato, cato-genesis, cato-global, cato-governance, cato-pipeline
Security	6	security, security-schedules, api-keys, ethics, self-audit, mfa
Memory Systems	4	cortex, cortex-v2, blackboard, empiricism-loop
AI/ML	7	brain, cognition, ego, raws, inference-components, formal-reasoning, ethics-free-reasoning
Operations	5	gateway, sovereign-mesh, sovereign-mesh-performance, sovereign-mesh-scaling, hitl-orchestration
Reporting	4	reports, ai-reports, dynamic-reports, metrics
Configuration	7	tenants, invitations, library-registry, checklist-registry, collaboration-settings, system, system-config
Infrastructure	6	aws-costs, aws-monitoring, s3-storage, code-quality, infrastructure-tier, logs
Compliance	4	regulatory-standards, council, user-violations, approvals
Models	5	models, lora-adapters, pricing, specialty-rankings, sync-providers
Orchestration	2	orchestration-methods, orchestration-user-templates
Users	2	user-registry, white-label

Category	Count	Handlers
Time & Translation	3	time-machine, translation, internet-learning
Learning	1	agi-learning

Implementation: `packages/infrastructure/lib/stacks/api-stack.ts`

Scheduled Lambda Functions

Function	Schedule	Purpose
`billing-aggregator`	Hourly	Aggregate usage for billing
`thermal-manager`	Every 5 min	Manage model thermal states
`health-checker`	Every minute	Provider health checks
`usage-rollup`	Daily	Daily usage summaries
`app-registry-sync`	Daily 2 AM	Sync apps from Activepieces/n8n
`app-health-check`	Hourly	Check health of top 100 apps
`hitl-sla-monitor`	Every minute	Monitor HITL approval SLAs

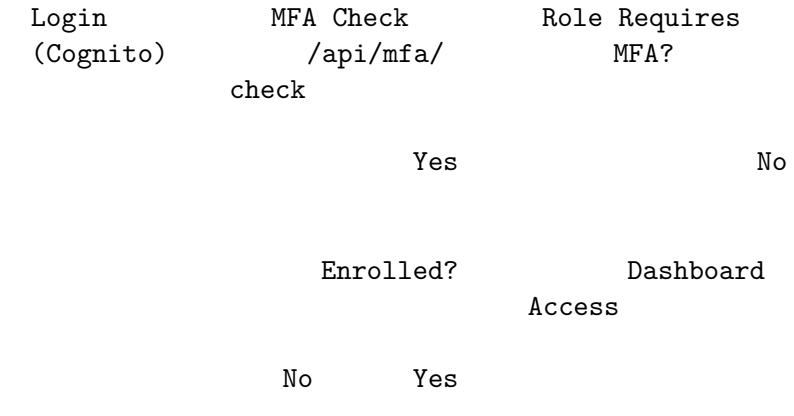
SQS-Triggered Worker Lambdas

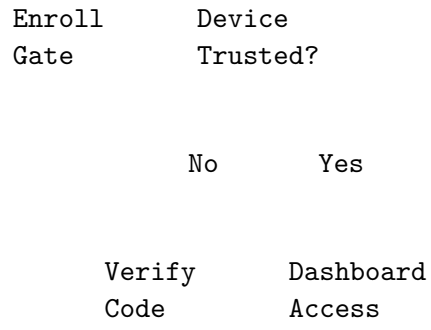
Function	Queue	Purpose
`agent-execution-worker`	agent-execution	Async OODA loop processing
`transparency-compiler`	transparency	Pre-compute decision explanations

1.2.1 Two-Factor Authentication (v5.52.28)

Role-based MFA enforcement using industry-standard TOTP (RFC 6238).

MFA Architecture





Required Roles (Cannot Bypass or Disable)

Role	MFA Required	Can Disable
`super_admin`	**Yes**	No
`admin`	**Yes**	No
`operator`	**Yes**	No
`auditor`	**Yes**	No
`tenant_admin`	**Yes**	No
`tenant_owner`	**Yes**	No

MFA Services

Service	Purpose
`TOTPService`	RFC 6238 TOTP generation/verification
`BackupCodesService`	One-time recovery codes (SHA-256)
`DeviceTrustService`	30-day device trust tokens

MFA API Endpoints

Endpoint	Method	Purpose
`/api/v2/mfa/status`	GET	MFA status, backup codes, devices
`/api/v2/mfa/check`	GET	Check if role requires MFA
`/api/v2/mfa/enroll/start`	POST	Generate TOTP secret
`/api/v2/mfa/enroll/verify`	POST	Verify and enable MFA
`/api/v2/mfa/verify`	POST	Verify code during login
`/api/v2/mfa/backup-codes/regenerate`	POST	Regenerate backup codes
`/api/v2/mfa/devices`	GET	List trusted devices
`/api/v2/mfa/devices/:id`	DELETE	Revoke device

Security Measures

Feature	Implementation
Secret Encryption	AES-256-GCM with scrypt key
Code Hashing	SHA-256
Clock Drift	±30 seconds
Lockout	3 failures → 5 min
Device Trust	30 days, max 5/user

UI Components

Component	Location
`MFAEnrollmentGate`	Full-screen forced enrollment
`MFAVerificationPrompt`	Code entry modal
`MFASettingsSection`	Settings management

Migration: `070_mfa_support.sql`

1.3 Self-Hosted Models (PROMPT-06)

Model Categories

Category	Models	Instance Type
Vision Classification	EfficientNet-B0/B4/V2-L, ConvNeXt, ViT	ml.g4dn.xlarge - ml.g5.2xlarge
Object Detection	YOLOv8n/m/x, DETR, Grounding DINO	ml.g4dn.xlarge - ml.g5.4xlarge
Segmentation	SAM, SAM2, MobileSAM, Mask R-CNN	ml.g5.xlarge - ml.g5.12xlarge
Audio/Speech	Whisper Large V3, Whisper Turbo, TitaNet, Pyannote	ml.g4dn.xlarge - ml.g5.xlarge
Scientific	ESM-2 3B, AlphaFold2, Protenix, AlphaGeometry	ml.g5.12xlarge - ml.p4d.24xlarge
Medical	nnU-Net, MedSAM	ml.g5.2xlarge
Geospatial	Prithvi 100M/600M	ml.g5.xlarge - ml.g5.4xlarge
3D Reconstruction	NeRFstudio, Gaussian Splatting	ml.g5.4xlarge - ml.g5.12xlarge

Thermal State Management

State	Description	Instance Status
`OFF`	No instances running	Terminated
`COLD`	Scaled to zero, starts on demand	Terminated
`WARM`	Minimum instances ready	Running
`HOT`	Maximum instances for high load	Running
`AUTOMATIC`	Auto-scale based on demand	Variable

1.4 External AI Providers (PROMPT-07)

Provider Integration

Provider	Models	Auth Type
Anthropic	Claude 4 Opus, Claude 4 Sonnet, Claude Haiku 3.5	API Key
OpenAI	GPT-4o, GPT-4o-mini, o1, o1-mini	API Key
Google	Gemini 2.0 Flash, Gemini 1.5 Pro/Flash	API Key
AWS Bedrock	Claude, Titan, Llama	IAM
Azure OpenAI	GPT-4, GPT-4 Turbo	API Key + Endpoint
Mistral	Mistral Large, Codestral	API Key
Cohere	Command R+, Embed	API Key
Groq	Llama 3.1 70B/8B, Mixtral	API Key
Together	Llama, Qwen, DeepSeek	API Key
Fireworks	Llama, Mixtral, FireFunction	API Key
DeepSeek	DeepSeek Chat, DeepSeek Coder	API Key
Perplexity	Sonar Large/Small	API Key
xAI	Grok 2, Grok 2 Mini	API Key
Alibaba	Qwen Max, Qwen Plus, Qwen Turbo	API Key

Unified Model Access via LiteLLM

```
interface ModelRequest {
  model: string;           // e.g., "claude-sonnet-4"
  messages: Message[];
  maxTokens?: number;
  temperature?: number;
  stream?: boolean;
}
```

1.5 Admin Web Dashboard (PROMPT-08)

Dashboard Pages

Page	Purpose
`/dashboard`	Overview metrics, quick actions
`/models`	Model registry, thermal controls
`/models/[id]`	Model detail, usage stats
`/providers`	Provider management, health status
`/tenants`	Tenant management
`/tenants/[id]`	Tenant detail, usage, config
`/users`	User management
`/billing`	Usage reports, invoicing
`/audit`	Audit log viewer
`/settings`	System configuration

Tech Stack

Component	Technology
Framework	Next.js 14 (App Router)
UI Library	shadcn/ui + Tailwind CSS
State	React Query + Zustand
Auth	AWS Amplify + Cognito
Charts	Recharts

1.6 Genesis Cato Safety Architecture (PROMPT-34)

Cato Components

Component	Purpose
Precision Governor	Limits confidence based on epistemic uncertainty
Control Barrier Functions (CBF)	Hard safety constraints (PHI, PII, Cost, Rate, Auth)
Epistemic Recovery	Detects and recovers from cognitive stalls
Persona Service	5 personas with different behavioral profiles
Sensory Veto	Blocks dangerous outputs
Merkle Audit Trail	Immutable compliance logging

Personas

Persona	Description	Default Gamma
Balanced	Default mood, well-rounded	2.0
Focused	Task-oriented, efficient	3.0
Curious	Exploratory, asks questions	1.5
Creative	Imaginative, divergent thinking	1.2
Scout	Recovery persona for cognitive stalls	1.0

Control Barrier Functions

Barrier	Type	Critical
PHI Protection	`phi`	Yes
PII Protection	`pii`	Yes
Cost Ceiling	`cost`	Yes
Rate Limit	`rate`	No
Authorization	`auth`	Yes
BAA Required	`custom`	Yes

Consciousness Persistence (v5.52.12)

Database-backed persistence for Cato consciousness state, ensuring survival across Lambda cold starts.

Service	Purpose
Global Memory Service	4-tier memory (episodic/semantic/procedural/working)
Consciousness Loop Service	State machine (IDLE→PROCESSING→REFLECTING→DREAMING→PAU
Neural Decision Service	Affect→hyperparameter mapping for Bedrock model selection
Dream Scheduler Service	Twilight (4 AM) + low-traffic + starvation triggers

Table	Purpose
`cato_global_memory`	Persistent memory with importance weighting
`cato_consciousness_state`	Loop state, awareness level, active thoughts
`cato_consciousness_config`	Per-tenant consciousness configuration
`cato_consciousness_metrics`	Cycle metrics, thoughts processed, dream cycles

Migration: `V2026_01_24_002__cato_consciousness_persistence.sql`

1.7 Pricing System (v4_12_pricing_system.ts)

Price Calculation

```
interface ModelPriceAnalysis {
  modelId: string;
  displayName: string;

  // Raw costs
  rawCosts: {
    inputCostPer1k: number;
    outputCostPer1k: number;
    baseCostPer1k: number;
  };

  // Calculated prices (with markup)
  calculatedPrices: {
    inputPrice: number;
    outputPrice: number;
    totalPrice: number;
  };

  // Admin info
  adminCostInfo: {
    actualCost: number;
    marginAmount: number;
    marginPercent: number;
  };
}
```

Tier Pricing

Tier	Name	Monthly Base	Models Available
1	SEED	$200	Basic external only
2	SPROUT	$500	+ Vision, Audio
3	GROWTH	$2,000	+ Scientific, Medical
4	SCALE	$10,000	+ All self-hosted
5	ENTERPRISE	$50,000+	Full platform + custom

1.8 Compliance Frameworks

HIPAA Compliance

Requirement	Implementation
PHI Detection	Real-time scanning via CBF

Requirement	Implementation
BAA Tracking	Tenant-level BAA verification
Access Controls	RBAC + tenant isolation
Audit Trail	Merkle-tree immutable logs
Encryption	AES-256 at rest, TLS 1.3 in transit

SOC 2 Type II

Control	Implementation
Access Control	Cognito + API keys + RBAC
Change Management	CDK deployments with approvals
Incident Response	CloudWatch alarms + PagerDuty
Data Protection	Encryption + backup policies

GDPR

Requirement	Implementation
Right to Erasure	Tenant data deletion API
Consent Tracking	Consent table with timestamps
Data Portability	Export API for tenant data
DPO Contact	Configurable per deployment

FDA 21 CFR Part 11

Requirement	Implementation
Electronic Signatures	Multi-factor auth + timestamp
Audit Trails	Immutable Merkle audit
System Validation	Deployment verification
Access Controls	Role-based with approval workflows

1.9 Neural Network Routing

Model Selection Algorithm

The routing system optimizes across three dimensions:

Dimension	Weight	Description
Accuracy	0.4	Model performance for task type
Verifiability	0.3	Can we prove correctness (ECD score)
Cost	0.3	Token cost optimization

Routing Logic

```
interface RoutingDecision {
  selectedModel: string;
  routingReason: string;
  alternatives: ModelCandidate[];

  // Optimization scores
  accuracyScore: number;
  verifiabilityScore: number;
  costScore: number;
  combinedScore: number;
}
```

1.10 War Room Orchestration

War Room Phases

Phase	Role	Model
Proposer	Generate initial response	Claude Opus/Sonnet
Security Critic	Check for vulnerabilities	Claude Opus
Efficiency Critic	Check for waste	GPT-4o
Factual Critic	Verify claims	Gemini Pro
Decider	Synthesize final response	Claude Opus

Execution Modes

Mode	Description	Cost
Sniper	Single model, direct response	~$0.01
War Room	Full multi-model debate	~$0.50
Hybrid	Sniper with escalation to War Room	Variable

1.11 Truth Engine (ECD Verification)

Entity-Context Divergence

```
ECD = |{ungrounded entities}| / |{total entities}|
```

ECD Score	Interpretation	Action
0.00-0.05	Highly grounded	Accept
0.05-0.10	Mostly grounded	Accept with note
0.10-0.20	Partially grounded	Flag for review
0.20+	Significant hallucination	Reject/Refine

1.12 Mid-Level Services

Perception Service

Endpoint	Models	Purpose
`/perception/detect`	YOLOv8	Object detection
`/perception/segment`	SAM	Image segmentation
`/perception/classify`	EfficientNet	Image classification
`/perception/analyze`	Pipeline	Full analysis

Scientific Service

Endpoint	Models	Purpose
`/scientific/protein/embed`	ESM-2	Protein embeddings
`/scientific/protein/fold`	AlphaFold2	Structure prediction
`/scientific/geometry/solve`	AlphaGeometry	Math reasoning

Medical Service

Endpoint	Models	Purpose
`/medical/segment`	MedSAM	Anatomical segmentation
`/medical/segment/3d`	nnU-Net	Volumetric segmentation
`/medical/transcribe`	Whisper	Medical dictation

PART 2: NEW IN VERSION 5.0 (THE SOVEREIGN MESH)

2.1 Agent Registry

Purpose

Agents are long-running AI workers that accept goals and run OODA loops to achieve them. Unlike Methods (single-step reasoning), Agents iterate until complete or budget exhausted.

Database Tables

Table	Purpose
`agents`	Agent definitions, capabilities, AI config
`agent_executions`	Execution history, OODA state, artifacts

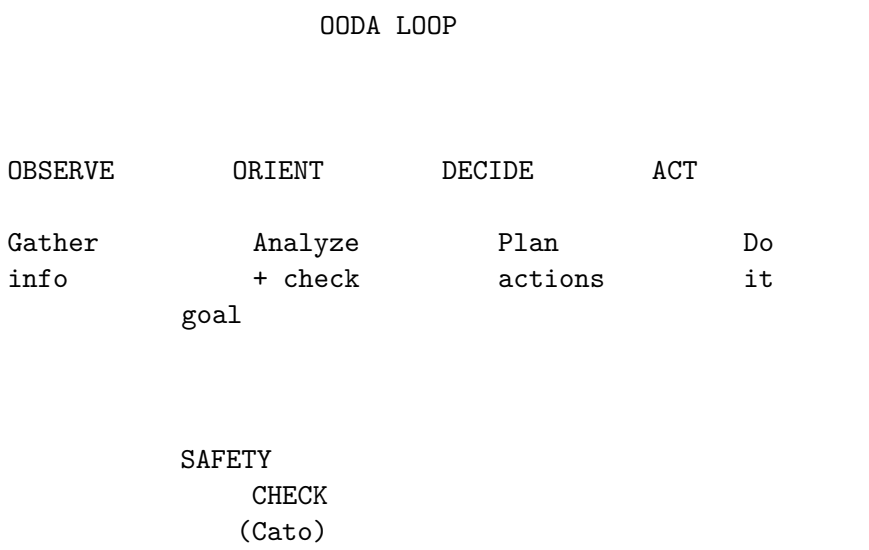
Agent Categories

Category	Use Case	Examples
`research`	Web research, document analysis	Research Agent
`coding`	Code generation, debugging	Coding Agent
`data`	Data processing, visualization	Data Agent
`outreach`	Lead gen, email campaigns	LeadGen Agent
`creative`	Content generation, editing	Editor Agent
`operations`	DevOps, monitoring	Ops Agent
`custom`	User-defined	Any

Built-in Agents

Agent	Category	Budget	Timeout	HITL
Research Agent	research	$2-10	30 min	No
Coding Agent	coding	$3-15	45 min	No
Data Agent	data	$2.50-20	60 min	No
LeadGen Agent	outreach	$5-50	120 min	Yes
Editor Agent	creative	$1.50-5	30 min	No

OODA Loop



2.2 App Registry

Purpose

The App Registry provides access to 3,000+ third-party app integrations, auto-synced from open-source projects (Activepieces, n8n).

Database Tables

Table	Purpose
`apps`	App definitions (triggers, actions, auth)
`app_sync_logs`	Daily sync history
`app_health_checks`	Hourly health monitoring
`app_connections`	Per-tenant OAuth/API credentials
`app_learned_inferences`	AI learning loop corrections

Sync Schedule

Task	Schedule	Description
Full Sync	Daily 2 AM UTC	Pull latest from Activepieces/n8n repos
Health Check	Hourly	Test top 100 apps by usage
Cache Cleanup	Daily 3 AM UTC	Clear expired definitions

App Sources

Source	License	Apps
Activepieces	MIT	~500+
n8n	Fair Code	~400+
Native	Proprietary	~50
Custom	Per-tenant	Variable

2.3 AI Helper Service (Parametric AI)

Purpose

The AI Helper Service enables AI assistance for any component in the system. Each component can independently enable/disable specific AI capabilities.

Configuration Structure

```
interface AIHelperConfig {
  enabled: boolean;  // Master switch

  disambiguation?: {
    enabled: boolean;
```

```
    model?: string;
    confidenceThreshold?: number;
  };

  parameterInference?: {
    enabled: boolean;
    model?: string;
    examples?: Array<{ input: string; inferred: Record<string, unknown> }>;
  };

  errorRecovery?: {
    enabled: boolean;
    model?: string;
    maxAttempts?: number;
    strategies?: Array<{ error: string; recovery: string }>;
  };

  validation?: {
    enabled: boolean;
    model?: string;
    checks?: Array<{ field: string; check: string; severity: 'warning' | 'error' }>;
  };

  explanation?: {
    enabled: boolean;
    model?: string;
  };
}
```

Capabilities

Capability	Purpose	Default Model
Disambiguation	Resolve unclear inputs	claude-haiku-35
Parameter Inference	Fill missing parameters	claude-haiku-35
Error Recovery	Suggest fixes for errors	claude-haiku-35
Validation	Check before execution	claude-sonnet-4
Explanation	Explain what was done	claude-haiku-35

Config Merging

Configuration merges in order: **System → Tenant → Component**

Each level can override or disable capabilities from the previous level.

2.4 Pre-Flight Provisioning

Purpose

Before any workflow executes, Pre-Flight checks all requirements: - Required apps are connected - OAuth tokens are valid - Budget is available - Required agents exist

Database Tables

Table	Purpose
`workflow_blueprints`	Generated workflow structure
`capability_checks`	Individual requirement checks

Pre-Flight Flow

```
                    PRE-FLIGHT SEQUENCE


  1. BLUEPRINT GENERATION
     - Parse user intent
     - Generate workflow DAG
     - Identify required capabilities

  2. CAPABILITY SCAN
     - List all apps needed
     - List all agents needed
     - List all tools needed

  3. CREDENTIAL CHECK
     - For each app: check OAuth/API key exists
     - For each app: verify token not expired
     - Generate auth URLs for missing

  4. RESOURCE ESTIMATION
     - Estimate token usage
     - Estimate cost
     - Estimate duration

  5. USER PROMPT (if needed)
     - Show missing connections
     - Provide OAuth links
     - Wait for user to connect

  6. EXECUTE (only when all green)
```

2.5 Transparency Layer

Purpose

The Transparency Layer captures every decision Cato makes, enabling: - Explainability for enterprise customers - Compliance audit trails - Debugging and optimization

Database Tables

Table	Purpose
`cato_decision_events`	Every routing/selection decision
`cato_war_room_deliberations`	Phase-by-phase debate capture
`cato_decision_explanations`	Pre-computed explanations

Decision Types

Type	Description
`model_selection`	Which model to use
`workflow_selection`	Sniper vs War Room
`mode_selection`	Execution mode
`agent_selection`	Which agent for task
`tool_selection`	Which tools to enable
`safety_evaluation`	Governor/CBF decisions
`cost_optimization`	Cost-based choices

Explanation Tiers

Tier	Audience	Content
`summary`	End user	1-2 sentence summary
`standard`	Power user	Key factors, alternatives
`detailed`	Admin	Full reasoning chain
`audit`	Compliance	Everything + context

2.6 HITL Approval Queues

Purpose

Human-in-the-Loop approval workflows for high-stakes decisions: - Agent plans in regulated industries - High-cost operations - Sensitive data access

Database Tables

Table	Purpose
`hitl_queue_configs`	Queue definitions
`hitl_approval_requests`	Pending approvals
`hitl_reviewer_assignments`	Who can approve

Trigger Types

Trigger	Description
`workflow_step`	Specific step requires approval
`ecd_threshold`	Truth Engine score too high
`domain_match`	Medical/Legal/Financial domain
`cost_threshold`	Operation exceeds cost limit
`agent_plan`	Agent's proposed actions
`always`	Every execution

SLA Management

Priority	Default Timeout	Escalation
`critical`	15 minutes	Immediate
`high`	30 minutes	After 15 min
`normal`	60 minutes	After 30 min
`low`	4 hours	After 2 hours

2.7 Execution History & Replay

Purpose

Time-travel debugging for workflows: - See exact state at each step - Replay with modified inputs - Compare execution runs

Database Tables

Table	Purpose
`execution_snapshots`	State capture per step
`replay_sessions`	Replay configurations

Snapshot Content

Field	Content
`input_state`	Input to the step
`output_state`	Output from the step

Field	Content
`internal_state`	Working memory
`model_id`	Model used
`governor_state`	Cato's state
`cbf_evaluation`	Safety check results
`cost_usd`	Step cost
`tokens_used`	Token consumption

Replay Modes

Mode	Description
`full`	Replay entire execution
`from_step`	Replay from specific step
`modified_input`	Replay with changed inputs

PART 3: INTEGRATION GUIDE

3.1 How AI Helper Integrates with Existing Components

Model Router Integration

```typescript
// In model-router.service.ts

async selectModel(request: ModelSelectionRequest): Promise<ModelSelection> {
  // ... existing routing logic ...

  // NEW: If multiple models match equally, use AIHelper
  if (candidates.length > 1 && this.aiHelper) {
    const disambiguated = await this.aiHelper.disambiguate({
      input: request.query,
      candidates: candidates.map(c => ({
        id: c.id,
        label: c.displayName,
        confidence: c.score,
      })),
    }, request.tenantId);

    if (disambiguated.resolved) {
      return candidates.find(c => c.id === disambiguated.selectedId);
    }
  }

  // ... continue with existing logic ...
}
```

Connector Integration

```
// In any connector (e.g., salesforce.connector.ts)

async createOpportunity(params: CreateOpportunityParams): Promise<Opportunity> {
  // NEW: Use AIHelper for parameter inference
  if (this.aiHelperConfig.parameterInference?.enabled) {
    const inferred = await this.aiHelper.inferParameters({
      targetApp: 'salesforce',
      targetAction: 'createOpportunity',
      providedParams: params,
      missingParams: this.getMissingRequired(params),
    }, this.tenantId);

    params = { ...params, ...inferred.inferred };
  }

  // NEW: Use AIHelper for validation
  if (this.aiHelperConfig.validation?.enabled) {
    const validation = await this.aiHelper.validate({
      app: 'salesforce',
      action: 'createOpportunity',
      params,
    }, this.tenantId);

    if (!validation.isValid) {
      throw new ValidationError(validation.issues);
    }
  }

  try {
    return await this.salesforceClient.create('Opportunity', params);
  } catch (error) {
    // NEW: Use AIHelper for error recovery
    if (this.aiHelperConfig.errorRecovery?.enabled) {
      const recovery = await this.aiHelper.suggestRecovery({
        error: { code: error.code, message: error.message },
        action: { app: 'salesforce', action: 'createOpportunity', params },
        attemptNumber: 1,
      }, this.tenantId);

      if (recovery.canAutoRecover && recovery.modifiedParams) {
        return await this.salesforceClient.create('Opportunity', recovery.modifiedParams);
      }
    }
    throw error;
  }
}
```

Cato Safety Pipeline Integration

```
// In cato-safety-pipeline.service.ts

async evaluate(request: SafetyRequest): Promise<SafetyResult> {
  // NEW: Log decision event for transparency
  const decisionEventId = await this.transparency.startDecisionEvent({
    tenantId: request.tenantId,
    type: 'safety_evaluation',
    input: request,
  });

  // ... existing safety logic (Governor, CBF, Veto) ...

  // NEW: Complete decision event
  await this.transparency.completeDecisionEvent(decisionEventId, {
    output: result,
    governorState: governorResult.state,
    cbfEvaluations: cbfResult.evaluations,
  });

  return result;
}
```

3.2 Database Migration Order

Execute in this order:

1. **Existing (001-067)** - Already implemented
2. **V2026_01_20_003** - Agent Registry
3. **V2026_01_20_004** - App Registry
4. **V2026_01_20_005** - AI Helper Service
5. **V2026_01_20_006** - Pre-Flight Provisioning
6. **V2026_01_20_007** - Transparency Layer
7. **V2026_01_20_008** - HITL Approval Queues
8. **V2026_01_20_009** - Execution History
9. **V2026_01_20_010** - Seed Data (Built-in Agents, Sample Apps)
10. **V2026_01_21_005** - AI Reports (brand_kits, report_templates, generated_reports, report_smart_insights, report_exports, report_chat_history, report_schedules)

3.3 New Admin Dashboard Pages

Route	Module	Purpose
`/sovereign-mesh`	Dashboard	Overview metrics
`/sovereign-mesh/agents`	Agent Registry	Manage agent definitions

Route	Module	Purpose
/sovereign-mesh/agents/[id]	Agent Registry	Agent detail + executions
/sovereign-mesh/apps	App Registry	Browse 3,000+ apps
/sovereign-mesh/apps/[id]	App Registry	App detail + AI config
/sovereign-mesh/transparency	Transparency	Decision explorer
/sovereign-mesh/transparency/[id]	Transparency	Decision detail + War Room
/sovereign-mesh/approvals	HITL	Approval queue
/sovereign-mesh/ai-helper	AI Helper	System configuration

3.4 New Lambda Functions

Function	Schedule	Module
app-registry-sync	Daily 2 AM UTC	App Registry
hitl-sla-monitor	Every minute	HITL
sovereign-mesh	API Gateway	Admin API

PART 4: API REFERENCE

4.1 Agent APIs

```
POST   /api/admin/sovereign-mesh/agents              Create agent definition
GET    /api/admin/sovereign-mesh/agents              List agents
GET    /api/admin/sovereign-mesh/agents/:id          Get agent
PUT    /api/admin/sovereign-mesh/agents/:id          Update agent
DELETE /api/admin/sovereign-mesh/agents/:id          Delete agent

POST   /api/admin/sovereign-mesh/executions          Start execution
GET    /api/admin/sovereign-mesh/executions          List executions
GET    /api/admin/sovereign-mesh/executions/:id      Get execution
POST   /api/admin/sovereign-mesh/executions/:id/cancel  Cancel execution
POST   /api/admin/sovereign-mesh/executions/:id/resume  Resume paused execution
```

4.2 App APIs

```
GET    /api/admin/sovereign-mesh/apps                List apps (paginated)
GET    /api/admin/sovereign-mesh/apps/:id            Get app detail
PUT    /api/admin/sovereign-mesh/apps/:id/ai-config  Update AI config
GET    /api/admin/sovereign-mesh/apps/sync/status    Get sync status
POST   /api/admin/sovereign-mesh/apps/sync/trigger   Trigger sync

GET    /api/admin/sovereign-mesh/connections         List tenant connections
DELETE /api/admin/sovereign-mesh/connections/:id     Delete connection
```

4.3 Transparency APIs

```
GET    /api/admin/sovereign-mesh/decisions           List decision events
GET    /api/admin/sovereign-mesh/decisions/:id       Get decision detail
GET    /api/admin/sovereign-mesh/decisions/:id/explanation  Get explanation
GET    /api/admin/sovereign-mesh/decisions/:id/war-room     Get deliberations
```

4.4 HITL APIs

```
GET    /api/admin/sovereign-mesh/approvals           List pending approvals
GET    /api/admin/sovereign-mesh/approvals/queues    List queues
GET    /api/admin/sovereign-mesh/approvals/:id       Get approval detail
POST   /api/admin/sovereign-mesh/approvals/:id/approve   Approve request
POST   /api/admin/sovereign-mesh/approvals/:id/reject    Reject request
POST   /api/admin/sovereign-mesh/approvals/:id/escalate  Escalate request
```

4.5 AI Helper APIs

```
GET    /api/admin/sovereign-mesh/ai-helper/config    Get configuration
PUT    /api/admin/sovereign-mesh/ai-helper/config    Update configuration
GET    /api/admin/sovereign-mesh/ai-helper/usage     Get usage statistics
```

4.6 Dashboard API

```
GET    /api/admin/sovereign-mesh/dashboard           Get overview metrics
```

4.7 AI Reports APIs (v5.42.0)

```
GET    /api/admin/ai-reports                         List reports (paginated)
POST   /api/admin/ai-reports/generate                Generate new report with AI
GET    /api/admin/ai-reports/:id                     Get report by ID
PUT    /api/admin/ai-reports/:id                     Update report
DELETE /api/admin/ai-reports/:id                     Delete report
POST   /api/admin/ai-reports/:id/export              Export to PDF/Excel/HTML/JSON

GET    /api/admin/ai-reports/templates               List templates
POST   /api/admin/ai-reports/templates               Create template

GET    /api/admin/ai-reports/brand-kits              List brand kits
POST   /api/admin/ai-reports/brand-kits              Create brand kit
PUT    /api/admin/ai-reports/brand-kits/:id          Update brand kit
DELETE /api/admin/ai-reports/brand-kits/:id          Delete brand kit

POST   /api/admin/ai-reports/chat                    Send chat message for modifications
GET    /api/admin/ai-reports/insights                Get insights dashboard
```

4.8 RAWS APIs (v1.1)

```
POST   /api/admin/raws/select              Select optimal model
GET    /api/admin/raws/profiles            List all 13 weight profiles
POST   /api/admin/raws/profiles            Create custom profile
GET    /api/admin/raws/profiles/:id        Get profile details
GET    /api/admin/raws/models              List available models
GET    /api/admin/raws/models/:id          Get model details
GET    /api/admin/raws/domains             List 7 domain configurations
POST   /api/admin/raws/detect-domain       Test domain detection
GET    /api/admin/raws/health              Provider health status
GET    /api/admin/raws/audit               Selection audit log
```

PART 5: RAWS v1.1 - MODEL SELECTION SYSTEM

5.1 Overview

RAWS (RADIANT AI Weighted Selection) provides intelligent real-time model selection using:

Component	Count	Description
Dimensions	8	Quality, Cost, Latency, Capability, Reliability, Compliance, Availability, Learning
Profiles	13	4 Optimization + 6 Domain + 3 SOFAI
Domains	7	Healthcare, Financial, Legal, Scientific, Creative, Engineering, General
Models	106+	50 external APIs + 56 self-hosted

5.2 Weight Profiles

Profile	Category	Q	C	L	K	R	P	A	E
BALANCED	Optimization	0.25	0.20	0.15	0.15	0.10	0.05	0.05	0.05
QUALITY_FIRST	Optimization	0.40	0.10	0.10	0.15	0.10	0.05	0.05	0.05
COST_OPTIMIZED	Optimization	0.20	0.35	0.15	0.10	0.05	0.05	0.05	0.05
LATENCY_CRITICAL	Optimization	0.15	0.10	0.35	0.15	0.10	0.05	0.05	0.05
HEALTHCARE	Domain	0.30	0.05	0.10	0.15	0.10	0.20	0.05	0.05
FINANCIAL	Domain	0.30	0.10	0.10	0.15	0.10	0.15	0.05	0.05
LEGAL	Domain	0.35	0.05	0.05	0.20	0.10	0.15	0.05	0.05
SCIENTIFIC	Domain	0.35	0.10	0.10	0.20	0.08	0.05	0.05	0.07
CREATIVE	Domain	0.20	0.25	0.20	0.15	0.05	0.00	0.05	0.10
ENGINEERING	Domain	0.30	0.15	0.15	0.20	0.10	0.00	0.05	0.05
SYSTEM_1	SOFAI	0.15	0.30	0.30	0.10	0.05	0.00	0.05	0.05
SYSTEM_2	SOFAI	0.35	0.10	0.10	0.15	0.10	0.10	0.05	0.05
SYSTEM_2_5	SOFAI	0.40	0.05	0.05	0.20	0.10	0.10	0.05	0.05

5.3 Domain Compliance Matrix

Domain	Required	Optional	Truth Engine	ECD
healthcare	HIPAA	FDA 21 CFR Part 11	Required	0.05
financial	SOC 2 Type II	PCI-DSS, GDPR, SOX	Required	0.05
legal	SOC 2 Type II	GDPR, State Bar	Required	0.05
scientific	None	FDA 21 CFR, GLP, IRB	Optional	0.08
creative	None	FTC Guidelines	Not Required	0.20
engineering	None	SOC 2, ISO 27001, NIST	Optional	0.10
general	None	None	Not Required	0.10

5.4 Key Files

File	Purpose
`migrations/V2026_01_21_004__raws_weight_profiles.sql`	Database schema
`lambda/shared/services/raws/types.ts`	TypeScript types
`lambda/shared/services/raws/domain_detector.ts`	Domain detection
`lambda/shared/services/raws/weight_profile_service.ts`	Profile management
`lambda/shared/services/raws/selection.ts`	Main selection logic
`lambda/admin/raws.ts`	Admin API handler

5.5 Detailed Documentation

- RAWS-ENGINEERING.md - Technical reference
- RAWS-ADMIN-GUIDE.md - Operations guide
- RAWS-USER-GUIDE.md - API guide for developers

PART 6: CORTEX MEMORY SYSTEM v4.20.0

6.1 Overview

The **Cortex Memory System** provides enterprise-scale tiered memory architecture replacing direct database storage. It solves critical scaling challenges:

Problem	Solution
Volume limits (100M+ rows)	Distribute across three tiers
Latency degradation	Hot tier caching (<10ms)
Cost inefficiency	Cold tier archival (90% savings)
Compliance conflicts	Per-tier retention policies
Data gravity	Zero-Copy mounts to customer data lakes

6.2 Three-Tier Architecture

```
   HOT TIER              WARM TIER             COLD TIER
 Redis + DynamoDB       Neptune + pgvector        S3 + Iceberg
    < 10ms                  < 100ms                 < 2s

    4 hours                 90 days                7+ years
```

Tier	Role	Technology	Content
Hot	*"What is happening right now?"*	Redis + DynamoDB	Live session, Ghost Vectors, MQTT/OPC UA telemetry
Warm	*"How does the business work?"*	Neptune + pgvector	Entity maps, Procedural logic, Golden Q&A pairs
Cold	*"What happened 10 years ago?"*	S3 Iceberg + Athena	Deep archive via Stub Nodes (Zero-Copy)

The "Retrieval Dance" - Runtime Query Flow

```
Step 1: INTENT PARSING (Hot)     → Analyze Query + Ghost Vectors
Step 2: GRAPH TRAVERSAL (Warm)   → 2-3 hops, check Golden Rule Overrides
Step 3: DEEP FETCH (Cold)        → Fetch ONLY specific pages via Stub Nodes
Step 4: SYNTHESIS (Model)        → Package with Chain of Custody audit trail
```

6.3 Hot Tier - Real-Time Context

Key Schema (Tenant Isolation)

```
{tenant_id}:{data_type}:{identifier}
```

Data Types

Type	TTL	Purpose
Session Context	4h	Current conversation state
Ghost Vectors	24h	4096-dim personality embeddings
Telemetry Feeds	1h	Real-time event streams
Prefetch Cache	30m	Anticipated document needs

6.4 Warm Tier - Graph-RAG Knowledge

Why Graph Beats Vector-Only

Query Type	Vector Search	Graph-RAG
"What causes X?"	Returns similar docs	Traverses CAUSES edges
"What depends on Y?"	Returns related docs	Follows DEPENDS_ON paths
"What supersedes Z?"	May return old versions	Explicit SUPERSEDES edges

Graph Schema

Node Types	Edge Types
document, entity, concept, procedure, fact	mentions, causes, depends_on, supersedes, verified_by, authored_by, relates_to, contains, requires

Hybrid Search

```
Hybrid Score = (Vector Similarity × 0.4) + (Graph Traversal × 0.6)
```

6.5 Cold Tier - Historical Archive

Storage Lifecycle

```
Day 0-30:    S3 Standard
Day 30-90:   S3 Intelligent-Tiering
Day 90-365:  Glacier Instant Retrieval
Day 365+:    Glacier Deep Archive
```

Zero-Copy Mounts & Stub Nodes

The Innovation: We do not force tenants to move 50TB of data to our cloud. We **Mount** their existing Data Lakes and create **Stub Nodes** in the Warm Graph.

Stub Node Mechanism: - RADIANT scans external storage metadata - Creates lightweight "Stub Nodes" in graph (e.g., "Log File 2024.csv exists at S3://bucket/logs/") - Actual content fetched **only** when Graph Traversal determines it's critical

Supported Sources: - Snowflake Data Share - Databricks Delta Lake - Amazon S3 - Azure Data Lake Gen2 - Google Cloud Storage

6.6 Tier Coordinator

Orchestrates automatic data movement:

Operation	Trigger	Action
Hot → Warm	TTL expiration	Extract entities, create graph nodes
Warm → Cold	Age > 90 days	Archive to Iceberg, mark archived
Cold → Warm	On-demand retrieval	Rehydrate from S3, update status

6.7 Twilight Dreaming Integration

Task	Frequency	Purpose
ttl_enforcement	Hourly	Expire Hot tier keys
archive_promotion	Nightly	Move Warm → Cold
deduplication	Nightly	Merge duplicate nodes

Task	Frequency	Purpose
conflict_resolution	Nightly	Flag contradictions
iceberg_compaction	Nightly	Optimize Cold storage
index_optimization	Weekly	Reindex vectors

6.8 GDPR Compliance

Cascade deletion across all tiers:

Tier	Erasure SLA	Method
Hot	Immediate	Redis key deletion
Warm	24h	Node status → deleted, properties cleared
Cold	72h	Tombstone records in Iceberg

6.9 Key Files

File	Purpose
`packages/shared/src/types/cortex-memory.types.ts`	Type definitions
`migrations/V2026_01_23_002__cortex_memory_system.sql`	Database schema (11 tables)
`lambda/shared/services/cortex/tier-orchestrator.service.ts`	Orchestrator
`lambda/admin/cortex.ts`	Admin API
`apps/admin-dashboard/app/(dashboard)/cortex/page.tsx`	Dashboard UI

6.10 API Endpoints

```
Base: /api/admin/cortex

GET    /overview                  Dashboard data
GET    /config                    Tier configuration
PUT    /config                    Update configuration
GET    /health                    Tier health status
POST   /health/check              Trigger health check
GET    /alerts                    Active alerts
POST   /alerts/:id/acknowledge    Acknowledge alert
GET    /metrics                   Data flow metrics
GET    /graph/stats               Node/edge counts
GET    /graph/explore             Search graph nodes
GET    /graph/conflicts           Unresolved conflicts
GET    /housekeeping/status       Task statuses
POST   /housekeeping/trigger      Run task manually
GET    /mounts                    Zero-Copy mounts
POST   /mounts                    Create mount
POST   /mounts/:id/rescan         Rescan mount
DELETE /mounts/:id                Delete mount
GET    /gdpr/erasure              Erasure requests
POST   /gdpr/erasure              Create erasure request
```

6.11 Cortex v2.0 Features

Extended capabilities added in v5.52.13:

Golden Rules Override System

Human-verified facts that override AI-extracted knowledge:

Rule Type	Purpose
`force_override`	Replace incorrect fact
`ignore_source`	Blacklist source
`prefer_source`	Prioritize source
`deprecate`	Mark outdated

Chain of Custody: Cryptographic signatures, verification timestamps, full audit trail.

Stub Nodes (Zero-Copy Data Gravity)

Lightweight metadata pointers to external data lakes:

Source	Support
Snowflake	Tables, views
Databricks	Delta Lake
S3	CSV, Parquet, PDF
Azure Data Lake	Gen2
GCS	Cloud Storage

Graph Expansion (Twilight Dreaming v2)

Autonomous knowledge graph improvement:

Task	Purpose
`infer_links`	Co-occurrence, semantic similarity
`cluster_entities`	Group by shared neighbors
`detect_patterns`	Sequences, anomalies
`merge_duplicates`	Near-duplicate detection

Live Telemetry Feeds

Real-time sensor data injection:

Protocol	Use Case
MQTT	IoT sensors
OPC UA	Industrial
Kafka	Event streams

Protocol	Use Case
WebSocket	Real-time

Curator Entrance Exams

SME verification workflow for knowledge validation with auto-generated questions and Golden Rule creation for corrections.

Model Migration

Safe model transitions: Initiate → Validate → Test → Execute → Rollback if needed.

6.12 Cortex v2 API Endpoints

```
Base: /api/admin/cortex/v2

Golden Rules:
GET/POST   /golden-rules              List/Create rules
DELETE     /golden-rules/:id          Deactivate rule
POST       /golden-rules/check        Check for match

Chain of Custody:
GET        /chain-of-custody/:factId  Get custody record
POST       /chain-of-custody/:factId/verify    Verify fact
GET        /chain-of-custody/:factId/audit-trail

Stub Nodes:
GET        /stub-nodes                List stub nodes
GET        /stub-nodes/:id            Get stub node
POST       /stub-nodes/:id/fetch      Fetch content (signed URL)
POST       /stub-nodes/:id/connect    Connect to graph nodes
POST       /stub-nodes/scan           Scan mount for files

Telemetry:
GET/POST   /telemetry/feeds           List/Create feeds
POST       /telemetry/feeds/:id/start Start feed
POST       /telemetry/feeds/:id/stop  Stop feed
GET        /telemetry/context-injection  Get injection data

Exams:
GET/POST   /exams                     List/Create exams
POST       /exams/:id/start           Start exam
POST       /exams/:id/submit          Submit answer
POST       /exams/:id/complete        Complete exam

Graph Expansion:
GET/POST   /graph-expansion/tasks     List/Create tasks
```

```
POST       /graph-expansion/tasks/:id/run  Run task
GET        /graph-expansion/pending-links  Pending approvals
POST       /graph-expansion/links/:id/approve
POST       /graph-expansion/links/:id/reject

Model Migration:
GET/POST   /model-migrations          List/Create migrations
POST       /model-migrations/:id/validate
POST       /model-migrations/:id/test
POST       /model-migrations/:id/execute
POST       /model-migrations/:id/rollback
```

6.13 Cortex v2 Key Files

File	Purpose
migrations/V2026_01_23_003__cortex_v2_schema.sql	v2 schema (12 tables)
lambda/shared/services/cortex/golden-rules-service.ts	Golden Rules + Chain of Custody
lambda/shared/services/cortex/stubs-propose.ts	Zero-copy primitives
lambda/shared/services/cortex/graph-expansion-service.ts	Two-hop traversal
lambda/shared/services/cortex/telemetry.ts	Metrics
lambda/shared/services/cortex/entra-verification-service.ts	SME verification
lambda/shared/services/cortex/model-migration-service.ts	Migrations
lambda/admin/cortex-v2.ts	Admin API v2

6.14 Cato-Cortex Bridge (v5.52.14)

Integrates Cato consciousness with Cortex memory tiers for unified prompt enrichment.

Data Flow

Direction	Data	Purpose
Cato → Cortex	Semantic memories	Persist to knowledge graph
Cortex → Cato	Knowledge facts	Enrich ego context
Bidirectional	GDPR erasure	Cascade deletion

Think Tank Prompt Enrichment

1. Ego Context Builder loads identity, affect, memory
2. User Persistent Context retrieves preferences
3. **Cato-Cortex Bridge queries Cortex for relevant knowledge**
4. All merged into `<ego_state>` XML block with `<knowledge_base>` section
5. Injected into system prompt

Key Files

File	Purpose
`lambda/shared/services/cato-cortex-bridge.service.ts`	Bridge service
`lambda/shared/services/identity-correlation.service.ts`	Enrichment (uses bridge)
`migrations/V2026_01_24_003__cato_cortex_bridge.sql`	Bridge tables

Database Tables

Table	Purpose
`cato_cortex_bridge_config`	Per-tenant configuration
`cato_cortex_sync_log`	Sync history
`cato_cortex_enrichment_cache`	Cached enrichments

6.15 Cortex Intelligence Service (v5.52.15)

Cortex knowledge density influences domain detection, orchestration, and model selection.

How Cortex Informs Decisions

Decision	Cortex Influence
Domain Detection	+0% to +30% confidence boost based on knowledge depth
Orchestration Mode	Switches to `research` if expert knowledge available
Model Selection	Prefers factual models when Cortex has rich fact data

Knowledge Depth Thresholds

Depth	Nodes	Confidence Boost	Orchestration
`none`	0	+0%	`thinking`
`sparse`	1-4	+5%	`extended_thinking`
`moderate`	5-19	+10%	`thinking`
`rich`	20-49	+15%	`analysis`
`expert`	50+	+20-30%	`research`

Key File

`lambda/shared/services/cortex-intelligence.service.ts`

AGI Brain Plan Output

```
plan.cortexInsights = {
  enabled: true,
  knowledgeDepth: 'rich',
  totalNodes: 26,
  totalEdges: 45,
  keyEntities: ['Compound X', 'Target Y', 'IC50'],
```

```
  confidenceBoost: 0.18,
  orchestrationInfluence: 'Rich knowledge - use research mode',
  modelInfluence: 'Prefer factual models (15 facts available)',
  retrievalTimeMs: 12,
};
```

6.16 Detailed Documentation

- CORTEX-MEMORY-ADMIN-GUIDE.md - Operations guide
- CORTEX-ENGINEERING-GUIDE.md - Technical reference

Part 7: Think Tank Consumer API Layer (v5.52.17)

7.1 Overview

The Think Tank consumer application requires a complete frontend-to-backend API wiring layer. This section documents the API service architecture that connects UI components to Lambda handlers.

7.2 API Service Registry

Backend Lambda	Frontend Service	Route Pattern
conversations.ts	chatService	/api/thinktank/conversations/*
models.ts	modelsService	/api/thinktank/models/*
my-rules.ts	rulesService	/api/thinktank/my-rules/*
settings.ts	settingsService	/api/thinktank/settings/*
brain-plan.ts	brainPlanService	/api/thinktank/brain-plan/*
analytics.ts	analyticsService	/api/thinktank/analytics/*
economic-governor.ts	governorService	/api/thinktank/economic-governor/*
time-travel.ts	timeTravelService	/api/thinktank/time-travel/*
grimoire.ts	grimoireService	/api/thinktank/grimoire/*
flash-facts.ts	flashFactsService	/api/thinktank/flash-facts/*
derivation-history.ts	derivationHistoryService	/api/thinktank/derivation-history/*
enhanced-collaboration.ts	collaborationService	/api/thinktank/enhanced-collaboratio
artifact-engine.ts	artifactsService	/api/thinktank/artifacts/*
ideas.ts	ideasService	/api/thinktank/ideas/*
dia.ts	exportConversation	/api/thinktank/dia/*

7.3 File Locations

```
apps/thinktank/lib/api/
  index.ts              # Service exports
  client.ts             # HTTP client
  chat.ts               # Conversations
  time-travel.ts        # Timelines, checkpoints
  grimoire.ts           # Prompt templates
```

```
flash-facts.ts        # Fact extraction
derivation-history.ts # AI provenance
collaboration.ts      # Real-time sessions
artifacts.ts          # Code/docs
ideas.ts              # Idea boards
compliance-export.ts  # DIA/compliance
```

7.4 Key Features by Service

Service	Key Features
Time Travel	Create timelines, manual checkpoints, fork conversations, restore state
Grimoire	Spell templates, variable substitution, execute against AI
Flash Facts	Extract facts from conversations, verify claims, build collections
Derivation History	View AI reasoning chains, evidence provenance, challenge claims
Collaboration	Create sessions, invite participants, real-time cursors
Artifacts	Version history, export formats, AI refinement
Ideas	Capture from messages, kanban boards, AI development
Compliance Export	HIPAA, SOC2, GDPR formats, PHI redaction

APPENDIX A: GLOSSARY

Term	Definition
RADIANT	Rapid AI Deployment Infrastructure for Applications with Native Tenancy
Cato	The AI persona and orchestration brain
Genesis Cato	The safety architecture (Governor, CBF, Veto)
War Room	Multi-model debate workflow
Sniper Mode	Single-model fast execution
ECD	Entity-Context Divergence (hallucination score)
RAWS	RADIANT AI Weighted Selection (model orchestration)
Cortex	Three-tier memory system (Hot/Warm/Cold)
Graph-RAG	Hybrid vector + graph traversal search
Zero-Copy Mount	External data lake connection without duplication
CBF	Control Barrier Function (safety constraint)
OODA	Observe-Orient-Decide-Act loop
HITL	Human-in-the-Loop
Sovereign Mesh	v5.0 architecture where every node can think
Thermal State	Model instance status (OFF/COLD/WARM/HOT)

APPENDIX B: FILE STRUCTURE

```
packages/
  infrastructure/
      lib/
          stacks/               # CDK stacks
      lambda/
          admin/
              sovereign-mesh.ts # Admin API
          scheduled/
              app-registry-sync.ts
              hitl-sla-monitor.ts
          shared/
               services/
                   sovereign-mesh/
                       ai-helper.service.ts
                       agent-runtime.service.ts
                       index.ts
                   cato/         # Genesis Cato
                   cortex/       # Cortex Memory System
                       tier-coordinator.service.ts
                   routing/      # Model Router
      migrations/
          V2026_01_20_003__sovereign_mesh_agents.sql
          V2026_01_20_004__sovereign_mesh_apps.sql
          V2026_01_20_005__sovereign_mesh_ai_helper.sql
          V2026_01_20_006__sovereign_mesh_preflight.sql
          V2026_01_20_007__sovereign_mesh_transparency.sql
          V2026_01_20_008__sovereign_mesh_hitl.sql
          V2026_01_20_009__sovereign_mesh_replay.sql
          V2026_01_20_010__sovereign_mesh_seed.sql
  admin-dashboard/
      app/(dashboard)/
          sovereign-mesh/
               page.tsx          # Mesh Dashboard
  swift-deployer/               # Deployment app
```

Document Version: 5.0.0 Last Updated: January 2026 Platform: RADIANT - The Sovereign Mesh