

Contents

RADIANT Progress Report	1
Cato Genesis System Implementation	1
Executive Summary	2
Implementation Overview	2
Total Files Created: 18	2
Total Lines of Code: ~4,500	2
Database Tables Added: 12	2
Admin API Endpoints: 35+	2
Phase 1: Python Genesis Package	2
Files Created	2
Key Features	2
Phase 2: TypeScript Services	3
Files Created	3
Genesis Service Features	3
Cost Tracking Service Features	3
Circuit Breaker Service Features	3
Consciousness Loop Service Features	3
Query Fallback Service Features	4
Phase 3: Infrastructure	4
CDK Stack Created	4
CloudWatch Alarms	4
AWS Budget	4
Phase 4: Database Migration	4
Migration: 103_cato_genesis_system.sql	4
Phase 5: Admin Interface	5
Admin API Endpoints	5
Admin Dashboard UI	5
Phase 6: Documentation	6
Files Created/Updated	6
Critical Fixes Applied	6
Deployment	6
Added to deploy.sh	6
Manual Execution	6
Files Modified	6
Architecture Diagram	7
Next Steps	8
Version History	8

RADIANT Progress Report

Cato Genesis System Implementation

Date: January 15, 2025

Version: 4.18.48

Author: AI Implementation Team

Executive Summary

Implemented the complete **Cato Genesis System** - a 3-phase boot sequence that solves the AI consciousness “Cold Start Problem” by initializing an agent with structured curiosity. The system includes developmental gates, circuit breakers for safety, real-time cost tracking from AWS APIs, and a query fallback service for graceful degradation.

Implementation Overview

Total Files Created: 18

Total Lines of Code: ~4,500

Database Tables Added: 12

Admin API Endpoints: 35+

Phase 1: Python Genesis Package

Files Created

File	Lines	Purpose
genesis/__init__.py	25	Package exports and documentation
genesis/structure.py	205	Domain taxonomy implantation
genesis/gradient.py	279	Epistemic gradient matrix setup
genesis/first_breath.py	394	Grounded introspection and calibration
genesis/runner.py	248	CLI orchestrator with idempotency
data/domain_taxonomy.js	353	800+ domain taxonomy
data/genesis_config.yaml	161	Matrix configuration

Key Features

Phase 1: Structure - Loads 800+ domain taxonomy as innate knowledge - Stores domains in DynamoDB semantic memory - Initializes atomic counters for developmental gates - Idempotent - safe to run multiple times

Phase 2: Gradient - Sets pymdp active inference matrices (A, B, C, D) - Implements “epistemic gradient” creating pressure to explore - Optimistic B-matrix with >90% EXPLORE success (Fix #2) - Prefers HIGH_SURPRISE over LOW_SURPRISE (Fix #6)

Phase 3: First Breath - Grounded introspection verifying environment - Model access verification via Bedrock - Shadow Self calibration using NLI semantic variance (Fix #3) - Baseline domain exploration bootstrapping

Phase 2: TypeScript Services

Files Created

File	Lines	Purpose
genesis.service.ts	340	Genesis state and developmental gates
cost-tracking.service.ts	520	Real AWS cost tracking
circuit-breaker.service.ts	480	Safety mechanisms
consciousness-loop.service.ts	550	Main consciousness loop
query-fallback.service.ts	290	Degraded-mode responses

Genesis Service Features

- **Genesis State Tracking:** Monitors completion of all 3 phases
- **Development Statistics:** Atomic counters for gate progression
- **Developmental Gates:** Capability-based stage advancement
 - SENSORIMOTOR → PREOPERATIONAL → CONCRETE_OPERATIONAL → FORMAL_OPERATIONAL
- **Stage Advancement:** Automatic when requirements met

Cost Tracking Service Features

- **Real-Time Estimates:** From CloudWatch metrics
- **Daily Costs:** From AWS Cost Explorer (24h delay)
- **MTD Costs:** Month-to-date with projection
- **Budget Status:** AWS Budgets integration
- **Pricing Table:** Refreshed from AWS Pricing API
- **NO HARDCODED VALUES:** All costs from AWS APIs

Circuit Breaker Service Features

- **5 Default Breakers:**
 - master_sanity - Master safety (requires admin approval)
 - cost_budget - Budget protection
 - high_anxiety - Emotional stability
 - model_failures - Model API protection
 - contradiction_loop - Logical stability
- **States:** CLOSED → OPEN → HALF_OPEN → CLOSED
- **Intervention Levels:** NONE, DAMPEN, PAUSE, RESET, HIBERNATE
- **SNS Notifications:** Alerts on state changes
- **CloudWatch Metrics:** Per-breaker state publishing

Consciousness Loop Service Features

- **Dual-Rate Architecture:**
 - System ticks (fast, lightweight, 2s interval)
 - Cognitive ticks (expensive, deliberate, 5min interval)
- **Daily Limits:** Max cognitive ticks per day
- **Emergency Mode:** Reduced operation when budget exceeded

- **State Persistence:** PyMDP state in PostgreSQL
- **Cost Recording:** Per-tick cost tracking

Query Fallback Service Features

- **Always Available:** Never throws exceptions
 - **Response Guarantee:** Within 500ms
 - **Cached Context:** Uses local/cached data only
 - **Graceful Degradation:** Based on intervention level
 - **Health Check:** Simple alive endpoint
-

Phase 3: Infrastructure

CDK Stack Created

Resource	Purpose
SNS Topic	Alert notifications
5 CloudWatch Alarms	Safety monitoring
CloudWatch Dashboard	Real-time visibility
AWS Budget	Cost control

CloudWatch Alarms

Alarm	Trigger	Action
Master Sanity Breaker	Breaker opens	SNS alert
High Risk Score	Risk > 70%	SNS alert
Cost Breaker	Budget exceeded	SNS alert
High Anxiety	Anxiety > 80% sustained	SNS alert
Hibernate Mode	System hibernating	SNS alert

AWS Budget

- **Monthly Limit:** \$500 (configurable)
 - **Alerts:** 50%, 80%, 100% thresholds
 - **Cost Filter:** cato:component tag
-

Phase 4: Database Migration

Migration: 103_cato_genesis_system.sql

Table	Purpose
cato_genesis_state	Boot sequence tracking
cato_development_counters	Atomic counters (Fix #1)

Table	Purpose
cato_developmental_stage	Capability-based progression
cato_circuit_breakers	Safety mechanisms
cato_circuit_breaker_events	Event log
cato_neurochemistry	Emotional/cognitive state
cato_tick_costs	Per-tick cost tracking
cato_pricing_cache	AWS pricing cache
cato_pymdp_state	Meta-cognitive state
cato_pymdp_matrices	Active inference matrices
cato_consciousness_settings	Loop configuration
cato_loop_state	Loop execution tracking

Phase 5: Admin Interface

Admin API Endpoints

Genesis State (5 endpoints) - GET /genesis/status - Genesis phase status - GET /genesis/ready - Ready for consciousness - GET /developmental/status - Current stage - GET /developmental/statistics - Development counters - POST /developmental/advance - Force advancement (superadmin)

Circuit Breakers (6 endpoints) - GET /circuit-breakers - All breaker states - GET /circuit-breakers/:name - Single breaker - POST /circuit-breakers/:name/force-open - Force trip - POST /circuit-breakers/:name/force-close - Force close - PATCH /circuit-breakers/:name/config - Update config - GET /circuit-breakers/:name/events - Event history

Cost Tracking (6 endpoints) - GET /costs/realtime - Today's estimate - GET /costs/daily - Historical daily - GET /costs/mtd - Month-to-date - GET /costs/budget - AWS Budget status - POST /costs/estimate - Settings cost estimate - GET /costs/pricing - Pricing table

Query Fallback (3 endpoints) - GET /fallback - Get fallback response - GET /fallback/active - Check if fallback active - GET /fallback/health - Health check

Consciousness Loop (7 endpoints) - GET /loop/status - Loop status - GET /loop/settings - Loop settings - PATCH /loop/settings - Update settings - POST /loop/tick/system - Execute system tick - POST /loop/tick/cognitive - Execute cognitive tick - POST /loop/emergency/enable - Enable emergency mode - POST /loop/emergency/disable - Disable emergency mode

Other (1 endpoint) - GET /intervention-level - Current intervention level

Admin Dashboard UI

Location: /cato/genesis

Tabs: 1. **Genesis** - Phase completion status, domain count, self facts 2. **Development** - Developmental stage, statistics, requirements 3. **Circuit Breakers** - Breaker states, controls, neurochemistry 4. **Costs** - Real-time costs, budget status, breakdown

Phase 6: Documentation

Files Created/Updated

File	Purpose
docs/cato/adr/010-genesis-system.md	Architecture decision record
docs/cato/runbooks/circuit-breaker	Operations <small>sub</small> book
docs/RADIANT-ADMIN-GUIDE.md	Section 33 added
CHANGELOG.md	Version 4.18.48 entry

Critical Fixes Applied

Fix #	Problem	Solution	Impact
1	Zeno's Paradox	Atomic counters	Avoids expensive table scans
2	Learned Helplessness	Optimistic B-matrix	>90% EXPLORE success
3	Shadow Self Budget	NLI semantic variance	\$0 vs \$800/month
6	Boredom Trap	Prefer HIGH_SURPRISE	Prevents premature consolidation

Deployment

Added to deploy.sh

Genesis automatically runs after CDK deployment: 1. Detects Python3 installation 2. Installs dependencies (boto3, pyyaml, numpy) 3. Runs genesis sequence 4. Reports success or provides manual command

Manual Execution

```
# Run full genesis
python3 -m cato.genesis.runner

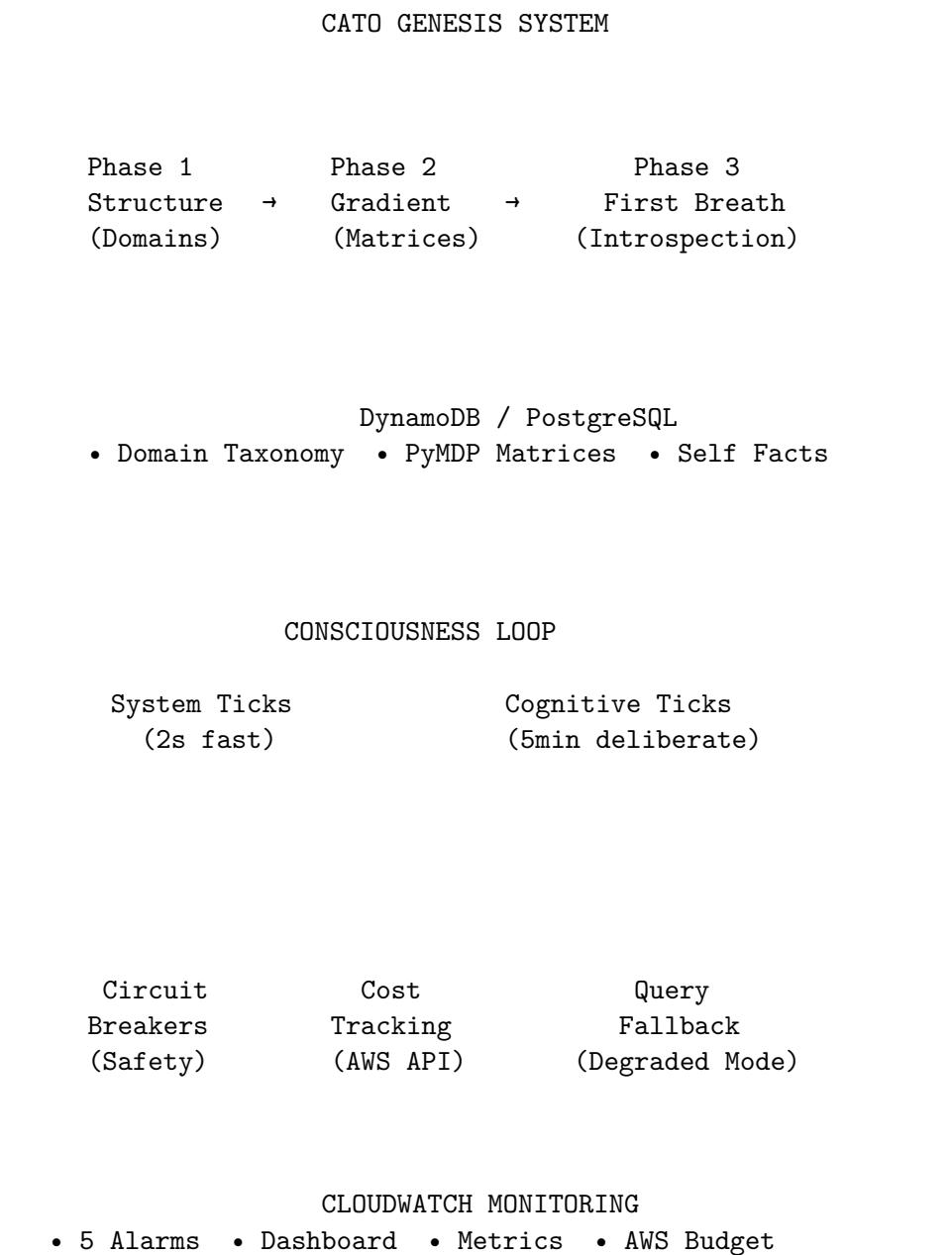
# Check status
python3 -m cato.genesis.runner --status

# Reset (CAUTION!)
python3 -m cato.genesis.runner --reset
```

Files Modified

File	Changes
scripts/deploy.sh	Genesis auto-run
metacognitive/bridge.py	DynamoDB persistence
cato/index.ts	New service exports
metacognitive/__init__.py	Updated exports

Architecture Diagram



Next Steps

1. **Deploy to AWS:** Run `./scripts/deploy.sh -e dev`
 2. **Run Migrations:** Apply migration 103
 3. **Execute Genesis:** Runs automatically or manually
 4. **Monitor Dashboard:** Check CloudWatch dashboard
 5. **Configure Budget:** Adjust budget limits as needed
-

Version History

Version	Date	Changes
4.18.48	2025-01-15	Initial Genesis System implementation

Document generated: January 15, 2025 RADIANT Platform v4.18.48