



# RADIANT Progress Report

## Bobble Genesis System Implementation

**Date:** January 15, 2025

**Version:** 4.18.48

**Author:** AI Implementation Team

### Executive Summary

Implemented the complete **Bobble 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

**18**

Files Created

**~4,500**

Lines of Code

**12**

Database Tables

**35+**

API Endpoints

## 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.json	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

### Circuit Breaker Service

Breaker	Purpose	Auto-Recovery
master_sanity	Master safety - requires admin approval	No
cost_budget	Budget protection	No (24h)
high_anxiety	Emotional stability	Yes (10 min)
model_failures	Model API protection	Yes (5 min)
contradiction_loop	Logical stability	Yes (15 min)

## Intervention Levels

Level	Condition	Effect
NONE	All breakers closed	Normal operation
DAMPEN	1 breaker open	Reduce cognitive frequency
PAUSE	2+ breakers open	Pause consciousness loop
RESET	3+ breakers open	Reset to baseline state
HIBERNATE	master_sanity open	Full shutdown

## Phase 3: Infrastructure

---

### CDK Stack: bobble-genesis-stack.ts

Resource	Purpose
SNS Topic	Alert notifications
5 CloudWatch Alarms	Safety monitoring
CloudWatch Dashboard	Real-time visibility
AWS Budget	Cost control (\$500/month default)

### 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

## Phase 4: Database Migration

---

### Migration: 103\_bobble\_genesis\_system.sql

Table	Purpose
bobble_genesis_state	Boot sequence tracking
bobble_development_counters	Atomic counters (Fix #1)
bobble_developmental_stage	Capability-based progression
bobble_circuit_breakers	Safety mechanisms
bobble_circuit_breaker_events	Event log
bobble_neurochemistry	Emotional/cognitive state
bobble_tick_costs	Per-tick cost tracking
bobble_pricing_cache	AWS pricing cache
bobble_pymdp_state	Meta-cognitive state
bobble_pymdp_matrices	Active inference matrices
bobble_consciousness_settings	Loop configuration
bobble_loop_state	Loop execution tracking



## Phase 5: Admin Interface

---

### Admin API Endpoints (35+)

Category	Endpoints	Count
Genesis State	status, ready, developmental status/statistics/advance	5
Circuit Breakers	list, get, force-open, force-close, config, events	6
Cost Tracking	realtime, daily, mtd, budget, estimate, pricing	6
Query Fallback	fallback, active, health	3
Consciousness Loop	status, settings, tick/system, tick/cognitive, emergency	7
Other	intervention-level	1

### Admin Dashboard UI

**Location:** /bobble/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

 **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

## Architecture Diagram





## Deployment

Genesis automatically runs after CDK deployment via `scripts/deploy.sh`

### Manual Execution

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

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

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



### 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

*Document generated: January 15, 2025*

**RADIANT Platform v4.18.48**