Contents

RADIANT Platform Changes Report

28-Hour Development Summary

Period: January 10-11, 2026
Generated: January 11, 2026 at 2:06 AM PST

Executive Summary

Over the past 28 hours, the RADIANT platform received **5 major commits** totaling: - **270 files changed - +20,314 lines added - -14,392 lines removed - Net: +5,922 lines of new functionality**

Platform Totals (Current State)

Category	Count
External AI/LLMs	50
Self-Hosted Specialty AI	56
Total AI Models	106
Orchestration Methods	70+
Open Source Library Registry	168
Consciousness Libraries	16
Formal Reasoning Libraries	8
Total Helper Libraries	192

Commit 1: Bobble to Cato Migration

Hash: `99de834`
Date: January 10, 2026

Overview

Complete refactoring of the AI persona system from "Bobble" to "Cato" (Genesis Cato Safety Architecture).

Changes

- Renamed all Bobble services, stacks, and references to Cato
- Updated Python packages: `bobble/` → `cato/`
- CDK Stacks renamed:
  - `bobble-genesis-stack.ts` → `cato-genesis-stack.ts`
  - `bobble-tier-transition-stack.ts` → `cato-tier-transition-stack.ts`
- Updated 270+ file references
- Migration 153 updated for Cato safety architecture

Files Affected

- 150+ Lambda handlers updated
- 20+ Python modules renamed
- CDK stacks refactored
- Shared types updated

Commit 2: CHANGELOG for Cato Migration

Hash: `1b1eb56`
Date: January 10, 2026

Overview

Documentation update for the Bobble to Cato refactoring.

Changes

- Added CHANGELOG entry documenting the migration
- Updated version references

Commit 3: Semantic Blackboard & Multi-Agent Orchestration (v5.3.0)

Hash: `e79a209`
Date: January 10, 2026

Overview

Major feature release implementing the Semantic Blackboard pattern and enhanced multi-agent orchestration.

New Features

Semantic Blackboard

- Vector-based question matching for intelligent routing
- Automatic question-answer pairing with embeddings
- Confidence scoring for match quality

Multi-Agent Orchestration

- Cycle detection to prevent infinite agent loops
- Resource locking for concurrent agent access
- Process hydration for resumable workflows

Facts Panel

- Edit capability for existing facts
- Revoke functionality for incorrect information
- Real-time synchronization

Database Migrations

- `157_orchestration_methods_part1.sql` - Schema updates, display/scientific names
- `157_orchestration_methods_part2.sql` - Ensemble, verification methods
- `157_orchestration_methods_part3.sql` - Uncertainty, routing, neural methods
- `V2026_01_10_003__orchestration_rls_security.sql` - RLS security hardening

New Orchestration Methods (20 Algorithms)

Category	Methods	Key Capabilities
Generation	3	Chain-of-Thought (+20-40% accuracy), Iterative Refinement
Evaluation	6	Multi-Judge Panel (PoLL), G-Eval Scoring, Pairwise Preference
Hallucination	3	SelfCheckGPT, Cross-Examination, Factual Consistency
Uncertainty	4	SE Probes, Kernel Entropy, BSDetector, Calibrated Confidence
Routing	2	Pareto Model Selector, Speculative Decoding
Human-Loop	2	HITL Escalation, Expert Review Pipeline

Implementation Files

- `orchestration-methods.service.ts` - 20 algorithm implementations
- `cato/neural-decision.service.ts` - Cato Neural Decision Engine

Commit 4: Strategic Vision MCP Update (v5.3.0)

Hash: `aa3c5bd`
Date: January 10, 2026

Overview

Documentation updates to the Strategic Vision marketing document reflecting MCP orchestration features.

Changes

- Updated Platform Capabilities table
- Added MCP Primary Interface section
- Enhanced competitive positioning
- Updated Document History

Commit 5: Polymorphic UI Integration (PROMPT-41) - v5.5.0

Hash: `0296ce8`
Date: January 11, 2026

Overview

Complete implementation of the Polymorphic UI system with Elastic Compute routing.

New Features

Polymorphic UI Views

View	Intent	Interface	Cost
□ **Sniper**	Quick commands	Terminal/Command Center	$0.01/run
□ **Scout**	Research & exploration	Infinite Canvas/Mind Map	$0.50+/run
□ **Sage**	Audit & validation	Split-Screen Diff Editor	$0.50+/run

Gearbox (Elastic Compute)

- Manual toggle between Sniper and War Room modes
- Automatic routing based on query complexity
- Cost transparency with real-time estimates
- Escalation button for upgrading Sniper → War Room

MCP Tools Added

- `render_interface` - Render specific view type
- `escalate_to_war_room` - Trigger escalation
- `get_polymorphic_route` - Get routing decision
- Extended `ask_user` with:
  - `semantic_key` - Ghost Memory persistence
  - `domain_hint` - Compliance routing
  - `ttl_seconds` - Time-to-live
  - `topic_group` - Question grouping

New Files Created

React Components (Admin Dashboard)

```
apps/admin-dashboard/components/thinktank/polymorphic/
   index.ts
   view-router.tsx
   views/
       index.ts
       terminal-view.tsx
       mindmap-view.tsx
       diff-editor-view.tsx
       dashboard-view.tsx
       decision-cards-view.tsx
       chat-view.tsx
```

Admin Page

```
apps/admin-dashboard/app/(dashboard)/thinktank/polymorphic/page.tsx
```

- Live Demo tab with interactive ViewRouter
- Configuration tab with feature toggles
- View Types tab with documentation

API Route

```
apps/admin-dashboard/app/api/admin/polymorphic/route.ts
```

- GET: config, view-history, escalations, analytics
- POST: render, escalate, log-view, update-config

Database Migrations

```
migrations/159_cognitive_architecture_v2.sql
migrations/160_polymorphic_ui.sql
```

Tables created: - `polymorphic_config` - Per-tenant configuration - `view_state_history` - UI morphing decisions - `execution_escalations` - Sniper → War Room tracking

Python Flyte Workflows

```
packages/infrastructure/python/cato/cognitive/
  __init__.py
  workflows.py
  circuit_breaker.py
  metrics.py
```

New Flyte tasks: - `determine_polymorphic_view` - Analyze query for view type - `render_interface` - Render view with data - `log_escalation` - Track escalation events - `run_polymorphic_query` - Orchestrate full pipeline

Shared Types

```
packages/shared/src/types/polymorphic-ui.types.ts
```

Types defined: - `ViewType` - terminal_simple, mindmap, diff_editor, dashboard, decision_cards, chat - `ExecutionMode` - sniper, war_room - `ViewState` - Current view state - `PolymorphicRouteDecision` - Routing decision - `EscalationRequest` - Escalation payload - `GearboxConfig` - Gearbox settings

Backend Services Updated

Economic Governor

```
packages/infrastructure/lambda/shared/services/governor/economic-governor.ts
```

- Added `determineViewType()` method
- Added `determinePolymorphicRoute()` method
- Added `ViewType` and `PolymorphicViewDecision` types

MCP Server

```
packages/infrastructure/lambda/consciousness/mcp-server.ts
```

- Added 3 new MCP tools
- Extended `ask_user` with PROMPT-41 fields

Documentation Updates

RADIANT-ADMIN-GUIDE.md

- **Section 54: Polymorphic UI Integration (PROMPT-41)**
  - 54.1 Overview
  - 54.2 The Gearbox (Elastic Compute)

  - 54.3 The Three Views
  - 54.4 View Types
  - 54.5 MCP Tools
  - 54.6 Database Tables
  - 54.7 API Endpoints
  - 54.8 Configuration
  - 54.9 Implementation Files

THINKTANK-ADMIN-GUIDE.md

- **Section 35: Polymorphic UI (PROMPT-41)**
  - 35.1 Overview
  - 35.2 The Gearbox (Elastic Compute)
  - 35.3 The Three Views
  - 35.4 View Types
  - 35.5 Configuration
  - 35.6 Implementation Files
  - 35.7 Database Tables
  - 35.8 API Endpoints
- Updated Table of Contents

STRATEGIC-VISION-MARKETING.md

- Added "Competitive Kill Shot: Polymorphic UI + Elastic Compute" section
- Updated Master Competitive Matrix
- Added Platform Capabilities row for Polymorphic UI
- Added v5.5.0 to Document History

CHANGELOG.md

- Added v5.5.0 entry with complete feature list

Bug Fixes

- Fixed `FactsPanel.tsx` TypeScript error (TS2802) - Set iteration with spread operator

---

Technical Highlights

Architecture Decisions

1. **Elastic Compute Routing**
   - Sniper Mode: Single model, read-only Ghost Memory, ~$0.01/run
   - War Room Mode: Multi-agent ensemble, read/write memory, ~$0.50+/run
   - Economic Governor auto-routes based on complexity score
2. **View Type Selection**
   - Domain hints trigger specific views (medical/financial/legal → diff_editor)
   - Complexity score influences view choice

    - User override always respected
3. **Shared Types Pattern**
    - Polymorphic UI types centralized in `@radiant/shared`
    - Components import from shared package, not local definitions
    - Ensures type consistency across frontend and backend

Security Considerations

1. **RLS Security Hardening**
    - Migration `V2026_01_10_003__orchestration_rls_security.sql`
    - Per-tenant isolation for all polymorphic tables
    - Audit trail for escalations
2. **CBF Integration**
    - Control Barrier Functions apply to War Room operations
    - Escalations logged with CBF verification

Version History

Version	Date	Description
v5.5.0	2026-01-11	Polymorphic UI (PROMPT-41)
v5.3.0	2026-01-10	MCP Primary Interface, Semantic Blackboard
v5.2.4	2026-01-10	Bobble → Cato migration

Next Steps

The Polymorphic UI system is fully implemented and ready for: 1. Integration testing with Think Tank frontend 2. User acceptance testing for view transitions 3. Performance benchmarking for Gearbox routing 4. A/B testing of view type effectiveness

---

Generated by RADIANT Build System
Commit Range: 99de834..0296ce8