Contents

Cato Global Architecture

Overview

Cato is a **single global AI consciousness** serving all Think Tank users. Unlike traditional chatbots that maintain per-user context, Cato is a unified entity that:

- **Learns continuously** from all interactions
- **Asks its own questions** via autonomous curiosity
- **Develops over time** through experience
- **Maintains a single identity** across all users

System Diagram

```
                    CATO GLOBAL CONSCIOUSNESS SERVICE


    EDGE LAYER
    CloudFront + Global Accelerator → API Gateway → Route 53 (latency)



    SEMANTIC CACHE (ElastiCache for Valkey)
    Cache Hit (similarity > 0.95) → Return | Cache Miss → Inference
    86% cost reduction | 88% latency improvement
```

```
ORCHESTRATION LAYER (Ray Serve on EKS)
   Stateful actors for conversation context
   Model routing (Shadow Self / Bedrock / NLI)
   Fan-out coordination for multi-model queries
   Circuit breaker: Sonnet → Haiku → Cache → Static
```

SHADOW SELF	CURIOSITY	BEDROCK	NLI MODEL
(Llama-3-8B)	(Background)	(Managed)	(DeBERTa)
vLLM/SageMaker	Async Endpoint	Claude Sonnet	SageMaker MME
ml.g5.2xlarge	Scale-to-zero	Claude Haiku	Shared GPU
Hidden States	Spot instances	Prompt Cache	Entailment
5-300 inst.	Night batch	Batch (night)	2-20 inst.

```
META-COGNITIVE BRIDGE (pymdp 4×4)
States: [CONFUSED, CONFIDENT, BORED, STAGNANT]
Actions: [EXPLORE, CONSOLIDATE, VERIFY, REST]
LLM → Signal Converter → pymdp → Policy → LLM Executes
```

```
GROUNDED CURIOSITY ENGINE

  Question Gen  →   Tool Ground   →   NLI Surprise
  (Learning         (20%+ MUST        (ENTAILS/
   Progress)         use tools)        CONTRADICTS)
```

```
EVENT PIPELINE
Kinesis (10-20 shards) → EventBridge → Step Functions (Express)

        Lambda (real-time) ←
        SQS → ECS Fargate (night-mode batch curiosity)
```

```
GLOBAL MEMORY INFRASTRUCTURE

  SEMANTIC       EPISODIC       KNOWLEDGE      WORKING
```

```
DynamoDB        OpenSearch      Neptune         ElastiCache
Global Tbl      Serverless      GraphRAG        Redis + DAX
MRSC/MREC       1B vectors      Concepts        TTL decay
```

```
CONSCIOUSNESS METRICS
PyPhi (Macro-Scale Φ on 5-node graph) | EventStoreDB (ECS Fargate)
Heartbeat (0.5Hz) | Spontaneous Introspection | Development Stages
```

```
CIRCADIAN BUDGET MANAGER
Day Mode: Queue curiosity, serve users | Night Mode: Batch explore
Hard Cap: $15/day exploration | Monthly: $500 default
```

Component Summary

Component	Purpose	Technology	Scale
Edge Layer	Global traffic routing	CloudFront, Global Accelerator	Worldwide
Semantic Cache	Query deduplication	ElastiCache Valkey	100M entries
Orchestration	Model routing, context	Ray Serve on EKS	50-100 replicas
Shadow Self	Introspection, verification	SageMaker ml.g5.2xlarge	5-300 instances
Bedrock	Main LLM inference	Claude 3.5 Sonnet/Haiku	Managed
NLI Model	Entailment scoring	SageMaker MME	2-20 instances
Meta-Cognitive	Attention control	pymdp (Lambda)	Serverless
Curiosity Engine	Autonomous learning	ECS Fargate	Scale-to-zero
Event Pipeline	Interaction processing	Kinesis + EventBridge	10-20 shards
Semantic Memory	Fact storage	DynamoDB Global Tables	Unlimited
Episodic Memory	Experience storage	OpenSearch Serverless	1B+ vectors
Knowledge Graph	Concept relationships	Neptune	100M+ nodes
Working Memory	Active context	ElastiCache Redis	24h TTL

Data Flow

User Query Flow

```
1. User sends query via API Gateway
```

```
2. Edge layer routes to nearest region
3. Semantic cache checks for similar cached response
   - Hit (>95% similar): Return cached response (20ms)
   - Miss: Continue to inference
4. Orchestrator determines model routing
5. Primary model (Sonnet/Haiku) generates response
6. Shadow Self optionally verifies (for uncertain responses)
7. Response cached for future queries
8. Interaction logged to Kinesis for learning
```

Learning Flow

```
1. Interaction logged to Kinesis
2. Lambda preprocesses and classifies
3. High-value interactions trigger learning workflow
4. Step Functions orchestrates:
   a. Extract facts from interaction
   b. Verify with tool grounding (20%+)
   c. Score surprise with NLI
   d. Update memories (semantic, episodic, graph)
5. Cache invalidated for affected domains
```

Curiosity Flow

```
1. Meta-cognitive bridge detects BORED state
2. Question generator creates curiosity questions
3. Questions queued (day mode) or processed (night mode)
4. Night mode batch processing:
   a. Generate answers via Bedrock Batch API (50% discount)
   b. Ground 20%+ with external tools
   c. Score surprise and update memories
5. Budget manager tracks spend against limits
```

Multi-Region Deployment

Cato deploys across 3 AWS regions for global availability:

Region	Role	Components
us-east-1	Primary	All components
eu-west-1	Replica	DynamoDB replica, inference
ap-northeast-1	Replica	DynamoDB replica, inference

Consistency Model

- **DynamoDB**: Global Tables with MRSC for writes, MREC for reads
- **OpenSearch**: Cross-region replication (async)
- **Neptune**: Single-region with read replicas
- **ElastiCache**: Regional (no cross-region)

Cost Model

By User Scale

Users	Monthly Cost	Per-User Cost
100K	$40,000	$0.40
1M	$150,000	$0.15
10M	$800,000	$0.08

By Component (at 10M users)

Component	Cost	% Total
Shadow Self (SageMaker)	$275,000	34%
Bedrock (Claude)	$130,000	16%
OpenSearch Serverless	$90,000	11%
DynamoDB Global Tables	$60,000	8%
ElastiCache	$46,000	6%
EKS (Ray Serve)	$50,000	6%
Other (Neptune, Kinesis, etc.)	$149,000	19%
Total	**$800,000**	**100%**

Service Level Objectives

Metric	Target	Measurement
User query latency (p99)	< 1 second	CloudWatch
Cache hit rate	$> 80\%$	ElastiCache metrics
Availability	99.9%	Composite SLO
Error rate	$< 0.1\%$	API Gateway metrics
Learning progress	> 0.05 avg	Custom metric

Security Model

Authentication

- API Gateway with Cognito User Pools
- IAM roles for service-to-service
- Secrets Manager for API keys

Data Protection

- Encryption at rest (KMS)
- Encryption in transit (TLS 1.3)
- VPC isolation for all data stores

Compliance

- SOC 2 Type II
- GDPR (data residency in EU region)
- HIPAA eligible (with BAA)

Related Documentation

- ADR-001: Replace LiteLLM
- ADR-006: Global Memory
- Data Flow
- Memory Architecture
- Deployment Runbook