Contents

Think Tank Admin - API Gap Analysis

Generated: v4.18.0
Purpose: Map Think Tank Admin UI pages to Radiant backend APIs and identify gaps

Summary

Category	Count
Total Pages	23
APIs Fully Working (pre-existing)	8
APIs Correctly Wired	5
APIs Newly Implemented	5
Remaining Low Priority	5

Implementation Status

- **5 new Lambda handlers created**
- **1 database migration created** (`120_thinktank_admin_tables.sql`)
- **18 of 23 pages now have working APIs**

Page-by-Page Analysis

Fully Working (8 pages)

Page	UI Path	API Endpoints	Status
Users	`/users`	`GET /api/admin/thinktank/users`, `GET /api/admin/thinktank/users/stats`	`thinktank/users.ts`
Conversations	`/conversations`	`GET /api/admin/thinktank/conversations`, `GET /api/admin/thinktank/conversations/stats`, `GET /api/admin/thinktank/conversations/:id/messages`	`thinktank/conversations.ts`
Domain Modes	`/domain-modes`	`GET/PUT /api/admin/thinktank/domain-modes`, `GET /api/admin/thinktank/taxonomy-domains`, `POST /api/admin/thinktank/domain-modes/detect`	`thinktank/domain-modes.ts`
Model Categories	`/model-categories`	`GET /api/admin/thinktank/model-categories`, `PATCH /api/admin/thinktank/model-categories/:id`	`thinktank/model-categories.ts`
Ego	`/ego`	`GET /api/admin/ego/dashboard`, `GET/PUT /api/admin/ego/config`, `GET/PUT /api/admin/ego/identity`, etc.	`admin/ego.ts`
Workflow Templates	`/workflow-templates`	`GET /api/admin/orchestration/user-templates`, `GET /api/admin/orchestration/workflows`	`admin/orchestration-*.ts`
Structure from Chaos	`/structure-from-chaos`	`GET/PUT /api/thinktank/chaos/config`, `GET /api/thinktank/chaos/metrics`	`thinktank/structure-from-chac`
Magic Carpet	`/magic-carpet`	Demo page - no API needed	N/A

Already Correctly Wired (5 pages)

These pages are using the correct API paths that match their backend handlers.

Page	UI Uses	Backend Handler	Status
Concurrent Execution	`/api/thinktank/concurrent/*`	`thinktank/concurrent-execution.ts`	Working
Governor	`/api/mission-control/governor/*`	Needs verification	Check path
Grimoire	`/api/thinktank/grimoire/*`	`thinktank/grimoire.ts`	Working
Delight	`/api/admin/delight/*`	`delight/handler.ts`	Working

Page	UI Uses	Backend Handler	Status
Collaborate	`/api/thinktank/collaborate/sessions/:id`	`thinktank/enhanced-collaboration.ts`	Check path

NEWLY IMPLEMENTED APIs (5 pages)

Page	UI Path	API Endpoints	Lambda File
Dashboard	`/`	`GET /api/thinktank-admin/dashboard/stats`	`thinktank-admin/dashboard.ts`
Analytics	`/analytics`	`GET /api/admin/thinktank/analytics?days=X`	`thinktank/analytics.ts`
Settings	`/settings`	`GET /api/admin/thinktank/status`, `GET/PATCH /api/admin/thinktank/config`	`thinktank/settings.ts`
My Rules	`/my-rules`	`GET/POST/PUT/DELETE /api/admin/my-rules/*`	`thinktank/my-rules.ts`
Shadow Testing	`/shadow-testing`	`GET /api/admin/shadow-tests`, `GET/PUT /api/admin/shadow-tests/settings`, `POST /api/admin/shadow-tests/:id/start|stop|promote`	`thinktank/shadow-testing.ts`

Remaining Missing APIs (5 pages)

Page	UI Path	Expected API Endpoints	Priority
Delight Statistics	`/delight/statistics`	`GET /api/admin/delight/statistics` - **Already exists in delight/handler.ts**	Done
Artifacts	`/artifacts`	`GET /api/admin/artifact-engine/dashboard` - needs admin routes added	**MEDIUM**
Polymorphic	`/polymorphic`	`GET/PUT /api/admin/polymorphic/config` - static UI, low priority	**LOW**
Compliance	`/compliance`	Placeholder page - components to be migrated	**LOW**
Enhanced Collaborate	`/collaborate/enhanced`	Uses component, needs session ID from query	**LOW**

Detailed Gap Specifications

1. Dashboard Stats API (HIGH PRIORITY)

File to create: packages/infrastructure/lambda/thinktank-admin/dashboard.ts

```
// GET /api/thinktank-admin/dashboard/stats
interface DashboardStats {
  activeUsers: number;
  activeUsersChange: number;      // % change from last period
  conversations: number;
  conversationsChange: number;
  userRules: number;
  userRulesChange: number;
  apiRequests: number;
  apiRequestsChange: number;
}
```

2. Analytics API (HIGH PRIORITY)

File to create: packages/infrastructure/lambda/thinktank/analytics.ts

```
// GET /api/admin/thinktank/analytics?days=7|30|90
interface Analytics {
  overview: {
    totalUsers: number;
    activeUsers: number;
    totalConversations: number;
    totalMessages: number;
    avgMessagesPerUser: number;
    avgSessionDuration: number;
    totalTokensUsed: number;
    totalCost: number;
  };
  trends: Array<{
    date: string;
    users: number;
    conversations: number;
    messages: number;
    tokens: number;
    cost: number;
  }>;
  modelUsage: Array<{
    modelId: string;
    modelName: string;
    requests: number;
    tokens: number;
    cost: number;
    avgLatency: number;
  }>;
}
```

3. Think Tank Settings/Status API (HIGH PRIORITY)

File to create: `packages/infrastructure/lambda/thinktank/settings.ts`

```typescript
// GET /api/admin/thinktank/status
interface ThinkTankStatus {
  installed: boolean;
  version: string | null;
  dataRetained: boolean;
  health: {
    status: 'healthy' | 'degraded' | 'unhealthy';
    latencyMs: number;
    activeUsers: number;
    activeConversations: number;
    errorRate: number;
  } | null;
}

// GET/PATCH /api/admin/thinktank/config
interface ThinkTankConfig {
  maxUsersPerTenant: number;
  maxConversationsPerUser: number;
  maxTokensPerConversation: number;
  enabledModels: string[];
  rateLimits: {
    requestsPerMinute: number;
    tokensPerMinute: number;
  };
  features: {
    collaboration: boolean;
    voiceInput: boolean;
    codeExecution: boolean;
    fileUploads: boolean;
    imageGeneration: boolean;
  };
}
```

4. My Rules API (MEDIUM PRIORITY)

File to create: `packages/infrastructure/lambda/thinktank/my-rules.ts`

```typescript
// GET /api/admin/my-rules
// POST /api/admin/my-rules
// PUT /api/admin/my-rules/:id
// DELETE /api/admin/my-rules/:id
// GET /api/admin/my-rules/presets
interface UserRule {
  id: string;
  content: string;
  category: string;
  isActive: boolean;
  timesApplied: number;
  createdAt: string;
  updatedAt: string;
}
```

5. Shadow Testing API (MEDIUM PRIORITY)

File to create: `packages/infrastructure/lambda/thinktank/shadow-testing.ts`

```
// GET /api/admin/shadow-tests
// GET/PUT /api/admin/shadow-tests/settings
// POST /api/admin/shadow-tests/:id/start
// POST /api/admin/shadow-tests/:id/stop
// POST /api/admin/shadow-tests/:id/promote
interface ShadowTest {
  id: string;
  testName: string;
  baselineTemplateName: string;
  candidateTemplateName: string;
  testMode: string;
  trafficPercentage: number;
  minSamples: number;
  samplesCollected: number;
  baselineAvgScore: number | null;
  candidateAvgScore: number | null;
  winner: string | null;
  confidenceLevel: number | null;
  status: 'pending' | 'running' | 'completed' | 'promoted' | 'rejected';
  startedAt: string | null;
  completedAt: string | null;
}

interface ShadowSettings {
  defaultTestMode: string;
  autoPromoteThreshold: number;
  autoPromoteConfidence: number;
  maxConcurrentTests: number;
  notifyOnCompletion: boolean;
  notifyOnAutoPromote: boolean;
}
```

6. Delight Statistics API (MEDIUM PRIORITY)

Add to: `packages/infrastructure/lambda/delight/handler.ts`

```
// GET /api/admin/delight/statistics
interface DelightStatistics {
  overview: {
    totalMessagesShown: number;
    totalAchievementsUnlocked: number;
    totalEasterEggsDiscovered: number;
    totalSoundsPlayed: number;
    totalActiveUsers: number;
    firstActivityDate: string | null;
    lastActivityDate: string | null;
    daysWithActivity: number;
  };
  topMessages: Array<MessageStats>;
  achievementStats: Array<AchievementStats>;
  easterEggStats: Array<EasterEggStats>;
  weeklyTrends: Array<WeeklyTrend>;
}
```

7. Artifact Engine Admin API (MEDIUM PRIORITY)

Add to: `packages/infrastructure/lambda/thinktank/artifact-engine.ts`

```
// GET /api/admin/artifact-engine/dashboard
// PUT /api/admin/artifact-engine/validation-rules/:id
// POST /api/admin/artifact-engine/allowlist
// DELETE /api/admin/artifact-engine/allowlist/:packageName
```

Recommended Implementation Order

1. **Phase 1 - Critical** (enables basic admin functionality)
   - Dashboard Stats API
   - Settings/Status API
   - Analytics API
2. **Phase 2 - Path Fixes** (make existing APIs work)
   - Add admin route aliases OR update UI paths for:
     - Concurrent Execution
     - Governor
     - Grimoire
     - Delight
     - Collaborate
3. **Phase 3 - Feature APIs**
   - My Rules API
   - Shadow Testing API
   - Delight Statistics API
   - Artifact Engine Admin API
4. **Phase 4 - Polish**
   - Polymorphic config API

- Compliance page integration

Database Tables Required

Most APIs can use existing tables. New tables needed for:

1. **My Rules**: `user_rules` table (may already exist for "User Persistent Context")
2. **Shadow Testing**: `shadow_tests`, `shadow_test_samples`, `shadow_test_settings`

Next Steps

1. Create missing Lambda handlers in order of priority
2. Add API Gateway routes in CDK stack
3. Update UI API paths where needed OR add route aliases
4. Add database migrations if needed
5. Test end-to-end integration