Contents

RADIANT App Isolation Architecture

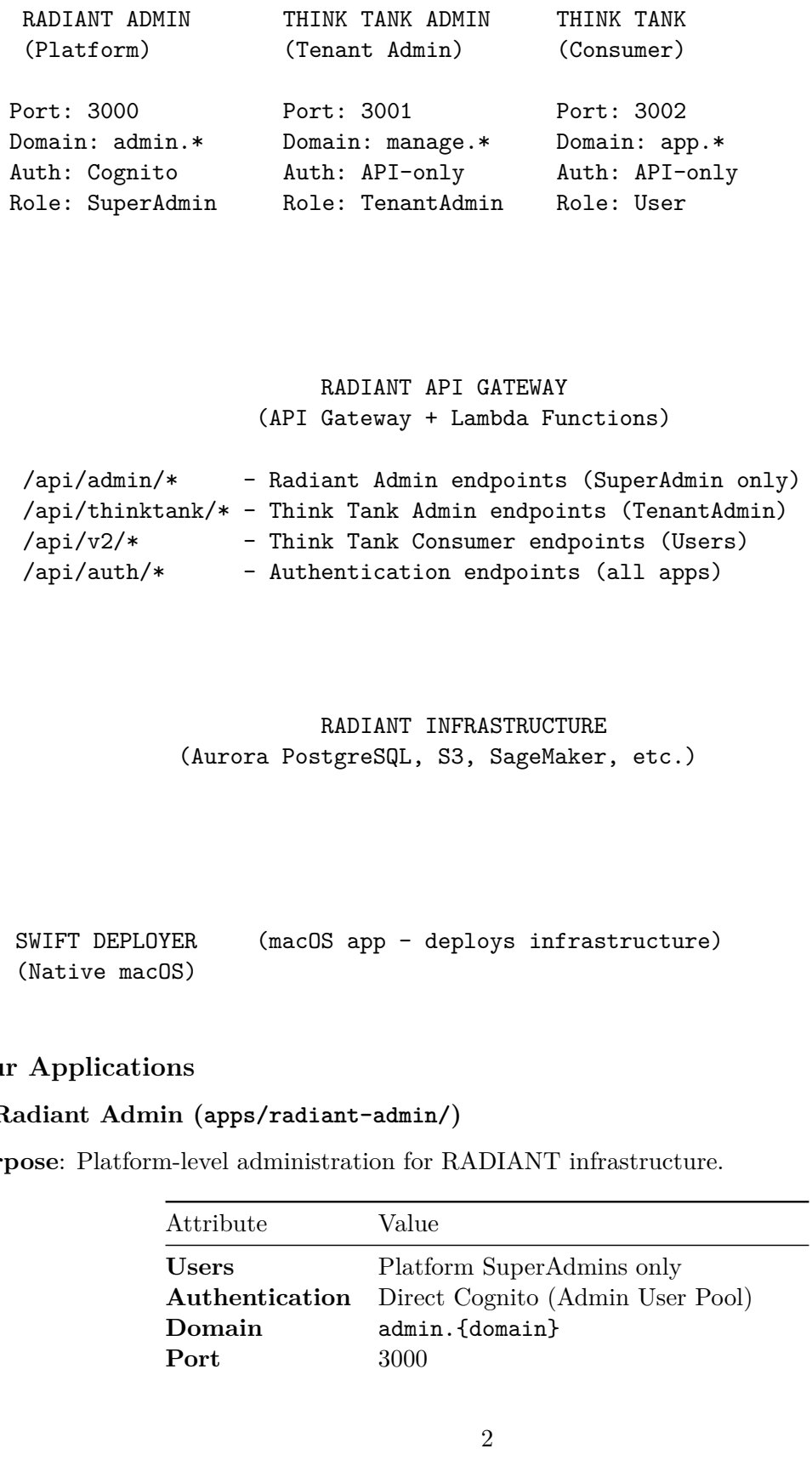
Version: 4.18.0
Status: MANDATORY - No exceptions
Last Updated: 2026-01-18

Executive Summary

RADIANT consists of **four completely isolated applications**. Think Tank MUST NOT share any web server, authentication context, or direct resource access with Radiant Admin. Period.

Architecture Overview

```
RADIANT PLATFORM
```



Four Applications

1. Radiant Admin (`apps/radiant-admin/`)

Purpose: Platform-level administration for RADIANT infrastructure.

Attribute	Value
Users	Platform SuperAdmins only
Authentication	Direct Cognito (Admin User Pool)
Domain	`admin.{domain}`
Port	3000

Attribute	Value
Access	Direct to Radiant resources (privileged)

Features: - Tenant management (create, suspend, delete) - Global model registry and pricing - Provider configuration (AWS Bedrock, OpenAI, etc.) - Infrastructure monitoring - Global billing and costs - Security alerts and compliance - Multi-region configuration - System-wide settings

What it does NOT contain: - Any Think Tank user features - Any Think Tank admin features - Conversation management - User rules management - Delight system configuration - Domain mode configuration

---

2. Think Tank Admin (`apps/thinktank-admin/`)

Purpose: Tenant-level administration for Think Tank.

Attribute	Value
Users	Tenant Administrators ONLY (TenantAdmin, SuperAdmin roles)
Authentication	API-only (via `/api/auth/admin/*` endpoints)
Domain	`manage.{tenant}.{domain}`
Port	3001
Access	API-only - NO direct Radiant access

CRITICAL: Admin Role Validation

Think Tank Admin uses **admin-only authentication endpoints** that validate user roles:

```
POST /api/auth/admin/login    - Validates TenantAdmin or SuperAdmin role
POST /api/auth/admin/refresh  - Re-validates admin role on every refresh
GET  /api/auth/admin/session  - Validates admin role for session checks
```

Non-admin users (regular Users) CANNOT access Think Tank Admin. No exceptions.

- Server-side: Lambda validates role is in `['SuperAdmin', 'TenantAdmin', 'super_admin', 'tenant_admin', 'admin']`
- Client-side: Double-validation in auth client (defense in depth)
- On 403 ADMIN_ACCESS_DENIED: Immediate redirect to login with clear error message

Features: - Think Tank user management (within tenant) - Conversation monitoring and moderation - User rules templates - Delight system configuration - Domain mode configuration - Model category preferences - Shadow testing configuration - Ego system configuration - Collaboration settings - Compliance within tenant - Tenant-level analytics

Security Rules: - All data access via Radiant API - Tenant isolation enforced by API - No direct database access - No direct S3 access - No Cognito SDK usage - Cannot access other tenants - Cannot modify platform settings

---

3. Think Tank Consumer (`apps/thinktank/`)

Purpose: End-user AI chat application.

Attribute	Value
Users	End users (consumers)
Authentication	API-only (via `/api/auth/*`)
Domain	`app.{tenant}.{domain}` or `{tenant}.thinktank.ai`
Port	3002
Access	API-only - NO direct Radiant access

Features: - AI chat interface - Conversation history - My Rules (personal preferences) - Brain plan visualization - Artifacts (generated content) - Collaboration (real-time sessions) - Domain selection - Model preferences - File uploads (via API) - Export/share conversations

Security Rules: - All data access via Radiant API - User can only access own data - Tenant isolation enforced by API - Rate limiting enforced by API - No direct database access - No direct S3 access - No Cognito SDK usage - No admin functionality

---

4. Swift Deployer (`apps/swift-deployer/`)

Purpose: macOS application for deploying RADIANT infrastructure.

Attribute	Value
Users	DevOps / Platform Engineers
Authentication	AWS credentials (local)
Platform	macOS 13.0+
Access	AWS APIs for deployment

No changes to this app.

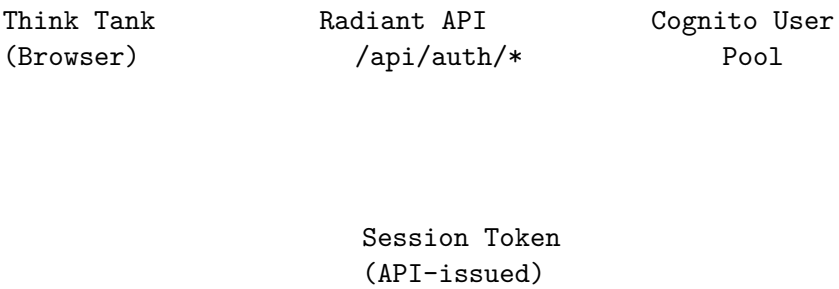
Authentication Architecture

Radiant Admin (Direct Cognito)

```
Radiant Admin        Cognito Admin         Radiant API
(Browser)              User Pool               (Privileged)
```

- Uses `@aws-amplify/auth` directly
- Admin User Pool (separate from Think Tank users)
- Returns Cognito tokens
- API validates Cognito JWT directly

Think Tank Apps (API-Only Auth)



- NO Cognito SDK in browser
- All auth via `/api/auth/*` endpoints:
  - `POST /api/auth/login` - Email/password login
  - `POST /api/auth/register` - User registration
  - `POST /api/auth/refresh` - Refresh session
  - `POST /api/auth/logout` - End session
  - `POST /api/auth/forgot-password` - Password reset
  - `POST /api/auth/verify-email` - Email verification
- API returns opaque session tokens (not Cognito tokens)
- Sessions stored server-side (Redis/DynamoDB)
- Tokens are httpOnly cookies or short-lived JWTs

Directory Structure

```
apps/
  radiant-admin/              # Platform administration
      app/
          (dashboard)/        # Admin pages
              tenants/
              models/
              providers/
              billing/
              security/
              compliance/
              orchestration/
              consciousness/
              brain/
              cato/
              ...
          api/                # BFF routes (proxy to Lambda)
      lib/
          auth/               # Cognito auth (direct)
          api/                # API client
      components/
```

```
thinktank-admin/            # Think Tank tenant administration
    app/
        (dashboard)/
            users/
            conversations/
            rules/
            delight/
            domain-modes/
            model-categories/
            shadow-testing/
            ego/
            collaborate/
            compliance/
            settings/
            analytics/
        api/                # BFF routes (proxy to Lambda)
    lib/
        auth/               # API-only auth
        api/                # API client
    components/

thinktank/                  # Think Tank consumer app
    app/
        (chat)/             # Main chat interface
        (settings)/         # User settings
            my-rules/
            preferences/
            profile/
        collaborate/        # Collaboration sessions
        artifacts/          # Generated artifacts
        api/                # BFF routes (proxy to Lambda)
    lib/
        auth/               # API-only auth
        api/                # API client
    components/

swift-deployer/             # macOS deployer (unchanged)
```

API Endpoint Segregation

Platform Admin Endpoints (`/api/admin/*`)

Only accessible by Radiant Admin with SuperAdmin role:

```
/api/admin/tenants/*
/api/admin/models/*
```

```
/api/admin/providers/*
/api/admin/billing/*
/api/admin/security/*
/api/admin/compliance/*
/api/admin/orchestration/*
/api/admin/consciousness/*
/api/admin/brain/*
/api/admin/cato/*
/api/admin/infrastructure/*
/api/admin/multi-region/*
```

Think Tank Admin Endpoints (`/api/thinktank/*`)

Accessible by Think Tank Admin with TenantAdmin role:

```
/api/thinktank/users/*
/api/thinktank/conversations/*
/api/thinktank/rules/*
/api/thinktank/delight/*
/api/thinktank/domain-modes/*
/api/thinktank/model-categories/*
/api/thinktank/shadow-testing/*
/api/thinktank/ego/*
/api/thinktank/collaborate/*
/api/thinktank/compliance/*
/api/thinktank/settings/*
/api/thinktank/analytics/*
```

Think Tank Consumer Endpoints (`/api/v2/*`)

Accessible by Think Tank users:

```
/api/v2/chat/*
/api/v2/conversations/*
/api/v2/my-rules/*
/api/v2/preferences/*
/api/v2/brain-plan/*
/api/v2/artifacts/*
/api/v2/collaborate/*
/api/v2/domain-taxonomy/*
/api/v2/models/*
/api/v2/upload/*
/api/v2/export/*
```

Auth Endpoints (`/api/auth/*`)

Used by Think Tank apps (not Radiant Admin):

```
POST /api/auth/login
POST /api/auth/register
```

```
POST /api/auth/refresh
POST /api/auth/logout
POST /api/auth/forgot-password
POST /api/auth/verify-email
POST /api/auth/change-password
GET  /api/auth/session
```

Security Invariants

MUST NEVER

1. **Think Tank apps MUST NEVER**:
   - Import `@aws-amplify/auth` or any Cognito SDK
   - Access AWS resources directly (S3, DynamoDB, etc.)
   - Have database connection strings
   - Have AWS credentials
   - Access Radiant Admin endpoints
   - Share cookies/sessions with Radiant Admin
2. **Radiant Admin MUST NEVER**:
   - Serve Think Tank consumer pages
   - Serve Think Tank admin pages
   - Share authentication with Think Tank

MUST ALWAYS

1. **Think Tank apps MUST ALWAYS**:
   - Authenticate via `/api/auth/*` endpoints only
   - Access data via Radiant API only
   - Include tenant ID in all requests
   - Handle API errors gracefully
2. **API MUST ALWAYS**:
   - Validate session tokens server-side
   - Enforce tenant isolation
   - Rate limit requests
   - Log all access attempts

Deployment

Separate Domains

App	Production Domain	Staging Domain
Radiant Admin	`admin.radiant.ai`	`admin.staging.radiant.ai`
Think Tank Admin	`manage.{tenant}.radiant.ai`	`manage.{tenant}.staging.radiant.ai`
Think Tank	`{tenant}.thinktank.ai`	`{tenant}.staging.thinktank.ai`

App	Production Domain	Staging Domain

CDK Stacks

```
// Separate CloudFront distributions
new RadiantAdminStack(app, 'RadiantAdmin', { ... });
new ThinkTankAdminStack(app, 'ThinkTankAdmin', { ... });
new ThinkTankConsumerStack(app, 'ThinkTankConsumer', { ... });
```

Migration Plan

Phase 1: Create New Apps

1. Create `apps/thinktank-admin/` with API-only auth
2. Create proper `apps/thinktank/` consumer app
3. Implement `/api/auth/*` Lambda endpoints

Phase 2: Migrate Features

1. Move Think Tank admin pages to `thinktank-admin/`
2. Move Think Tank consumer features to `thinktank/`
3. Update all API calls to use proper endpoints

Phase 3: Clean Radiant Admin

1. Remove all Think Tank code from `admin-dashboard/`
2. Rename `admin-dashboard/` to `radiant-admin/`
3. Update sidebar and navigation

Phase 4: Deploy & Verify

1. Deploy all three apps to separate domains
2. Verify authentication isolation
3. Verify API access controls
4. Security audit

Compliance

This architecture is REQUIRED for: - **SOC 2 Type II**: Separation of duties - **HIPAA**: Access controls and audit trails - **GDPR**: Data isolation and consent management - **PCI DSS**: Network segmentation (if processing payments)

Document History

Date	Version	Change
2026-01-18	1.0.0	Initial architecture document