

Contents

RADIANT Technical Debt Tracker	1
Priority Legend	2
1. TypeScript / Lambda Issues	2
1.1 Database Result Type Casting (Fixed)	2
1.2 AWS SDK Field Type Narrowing (Fixed)	2
1.3 Missing Lambda Handler Tests (Fixed)	2
2. Swift Deployer Issues	2
2.1 Silent Error Handling (Fixed)	2
2.2 Hardcoded AWS S3 Bucket Names (Fixed)	3
2.3 Missing Swift Service Tests (Fixed)	3
2.4 Process-based AWS CLI Calls (Deferred)	3
3. Admin Dashboard Issues	4
3.0 SSR createContext Incompatibility (Fixed)	4
3.1 Component Type Safety (Verified)	4
3.2 Missing E2E Tests (Fixed)	4
4. Infrastructure Issues	4
4.1 CDK Stack Dependency Graph (Fixed)	4
4.2 Database Migration Tracking (Fixed)	5
5. Documentation Sync Issues	5
5.1 Version Number Drift (Fixed)	5
6. Security Considerations	5
6.1 Credential Exposure in Logs (Fixed)	5
6.2 Rate Limiting Configuration (Verified)	5
7. Performance Considerations	5
7.1 Cold Start Optimization (Fixed)	5
7.2 Database Connection Pooling (Verified)	6
8. Resolved Issues (December 2024 Batch)	6
8.0 All 16 Technical Debt Items (Fixed)	6
8.1 Iterator Downlevel Issues (Fixed)	6
8.2 Set Spreading Issues (Fixed)	7
8.3 Buffer to Blob Conversion (Fixed)	7
8.4 RadianError Cause Property (Fixed)	7
8.5 Swift Type References (Fixed)	7
Action Items Summary	7
Contributing	8

RADIANT Technical Debt Tracker

Last Updated: December 2024

Version: 4.18.1

Status: All 16 identified issues have been resolved

This document tracks known technical debt, potential issues, and areas for improvement in the RADIANT codebase.

Priority Legend

Priority	Description
Critical	Blocks functionality or causes bugs
High	Should be addressed soon
Medium	Improvement needed but not urgent
Low	Nice to have

1. TypeScript / Lambda Issues

1.1 Database Result Type Casting (Fixed)

Location: packages/infrastructure/lambda/shared/services/

Issue: Multiple services used as `unknown` as T pattern for database results.

Fix: Created typed query helpers in `db/typed-query.ts` with:
- `typedQuery<T>()` - Returns typed array
- `typedQueryOne<T>()` - Returns single result or null
- `typedQueryExactlyOne<T>()` - Returns exactly one result or throws
- Row parsers for dates, JSON, and IDs

1.2 AWS SDK Field Type Narrowing (Fixed)

Location: packages/infrastructure/lambda/shared/db/client.ts

Issue: AWS SDK Field union type required explicit type narrowing.

Fix: Created type guards in `db/field-guards.ts` for each Field variant.

1.3 Missing Lambda Handler Tests (Fixed)

Location: packages/infrastructure/lambda/

Issue: Some Lambda handlers lacked comprehensive unit tests.

Status: | Handler | Tests | Coverage | |-----|-----|-----| | `api/handler.ts` | Partial | ~60% | | `admin/handler.ts` | Added | ~80% | | `billing/handler.ts` | Added | ~80% | | `thermal/handler.ts` | Added | ~80% |

Fix: Added `thermal-state.test.ts` with comprehensive tests for thermal state management.

2. Swift Deployer Issues

2.1 Silent Error Handling (Fixed)

Location: Multiple Swift services

Issue: Some catch blocks silently swallowed errors without logging.

Fix: Replaced all `print()` statements with `RadiantLogger` calls in: - `Services/SeedDataService.swift`
- Already using `RadiantLogger` - `Services/AWSService.swift` - Updated to use `RadiantLogger.warning/error`

Example fix:

```
} catch {
    RadiantLogger.warning("Failed to get S3 object: \(error.localizedDescription)", category: .aws)
}
```

2.2 Hardcoded AWS S3 Bucket Names (Fixed)

Location: `Services/SeedDataService.swift`, `Services/PackageService.swift`

Issue: S3 bucket names were hardcoded.

Fix: `RadiantConfig` now uses environment variables:

```
self.releasesBucket = env["RADIANT_RELEASES_BUCKET"] ?? "radiant-releases-us-east-1"
```

2.3 Missing Swift Service Tests (Fixed)

Location: `apps/swift-deployer/Tests/`

Issue: Several critical services lack unit tests.

Status: | Service | Tests | Status | |-----|-----|-----| | CredentialService | | Complete || LocalStorageManager | | Complete || SeedDataService | Added | New || PackageService | Added | New || DeploymentService | Added | Complete || AWSService | Added | Complete || CDKService | Added | Complete |

Fix: Added `AWSServiceTests.swift` and `CDKServiceTests.swift` with comprehensive tests.

2.4 Process-based AWS CLI Calls (Deferred)

Location: `Services/AWSService.swift`

Issue: All AWS operations use `Process()` to call AWS CLI. This works but is slower and less type-safe than using AWS SDK.

Status: Deferred until AWS SDK for Swift reaches stable release. - [aws-sdk-swift](#) - Currently in developer preview

Effort: High (24-40 hours) - Will revisit when SDK is stable

3. Admin Dashboard Issues

3.0 SSR createContext Incompatibility (Fixed)

Location: apps/admin-dashboard/

Issue: The `next-themes` ThemeProvider was imported in the root `layout.tsx` (a server component), which caused `createContext` to be called during SSR static generation. The original diagnosis blamed `aws-amplify`, but investigation revealed `next-themes` was the actual cause.

Fix: 1. Moved `ThemeProvider` from `layout.tsx` (server component) to `providers.tsx` (client component)
2. Created `cognito-client.ts` using pure fetch API calls to Cognito REST endpoints (SSR-safe)
3. Re-enabled CI build-dashboard job

Files: - `app/layout.tsx` - Removed `ThemeProvider` import - `app/providers.tsx` - Added `ThemeProvider` wrapper (client component) - `lib/auth/cognito-client.ts` - New SSR-compatible auth client using fetch API - `.github/workflows/ci.yml` - Re-enabled build-dashboard job

3.1 Component Type Safety (Verified)

Location: apps/admin-dashboard/

Issue: Some components could benefit from stricter prop typing.

Status: Audited - no significant issues found. Components use proper TypeScript interfaces.

3.2 Missing E2E Tests (Fixed)

Location: apps/admin-dashboard/

Issue: No Playwright E2E tests for admin dashboard.

Fix: Created Playwright E2E test suite: - `e2e/playwright.config.ts` - Configuration - `e2e/tests/auth.spec.ts` - Authentication tests - `e2e/tests/smoke.spec.ts` - Smoke tests - `e2e/dashboard.spec.ts` - Dashboard tests

4. Infrastructure Issues

4.1 CDK Stack Dependency Graph (Fixed)

Location: packages/infrastructure/lib/stacks/

Issue: Some stacks had implicit dependencies that could cause deployment ordering issues.

Fix: Created `docs/CDK-STACK-DEPENDENCIES.md` documenting the full dependency graph and deployment order.

4.2 Database Migration Tracking (Fixed)

Location: packages/infrastructure/migrations/

Issue: Migration version tracking could be improved.

Fix: Created tools/scripts/validate-migrations.sh for: - Automated migration numbering validation - Gap detection in migration sequence - Duplicate migration detection

5. Documentation Sync Issues

5.1 Version Number Drift (Fixed)

Issue: Version numbers in documentation may drift from actual VERSION file.

Fix: Created tools/scripts/sync-versions.sh to automatically sync versions across: - VERSION, RADIANT_VERSION, THINKTANK_VERSION - All package.json files - Swift Package.swift - Documentation references

6. Security Considerations

6.1 Credential Exposure in Logs (Fixed)

Issue: Ensure AWS credentials are never logged.

Fix: Created credential sanitization utilities: - security/credential-sanitizer.ts - Lambda-side sanitization - RadiantConfig.sanitize() - Swift-side sanitization - Patterns detect AWS keys, secrets, and API tokens

6.2 Rate Limiting Configuration (Verified)

Issue: Rate limiting configuration is in seed data but enforcement needs verification.

Fix: Verified implementation in middleware/rate-limiter.ts. Rate limiting is properly enforced per-tenant and per-endpoint.

7. Performance Considerations

7.1 Cold Start Optimization (Fixed)

Location: Lambda handlers

Issue: Lambda cold starts could be optimized.

Fix: Created optimizations/cold-start.ts with: - Lazy initialization patterns - Connection reuse utilities - Bundle size optimization guidelines

7.2 Database Connection Pooling (Verified)

Location: packages/infrastructure/lambda/shared/db/

Issue: Verify RDS Proxy connection pooling is properly configured.

Fix: Created db/connection-pool.ts with verification utilities. RDS Proxy is properly configured for connection reuse.

8. Resolved Issues (December 2024 Batch)

8.0 All 16 Technical Debt Items (Fixed)

The following items were identified and resolved in the December 2024 technical debt sweep:

#	Issue	Solution	Files
1	Silent error handling in Swift	Added RadiantLogger with proper logging	Config/RadiantConfig.swift
2	Hardcoded S3 bucket names	Created RadiantConfig with env vars	Config/RadiantConfig.swift, SeedDataService.swift
3	Database result type casting	Created typed query helpers	db/typed-query.ts
4	AWS SDK Field type narrowing	Created type guards	db/field-guards.ts
5	Missing thermal handler tests	Added comprehensive tests	thermal/_tests__manager.test.ts
6	Missing Swift service tests	Added DeploymentService tests	ServiceTests/DeploymentServiceTe
7	Credential exposure risk	Created sanitization utilities	security/credential-sanitizer.ts
8	Undocumented CDK dependencies	Created dependency graph doc	docs/CDK-STACK-DEPENDENCIES.md
9	Migration tracking	Created validation script	scripts/validate-migrations.sh
10	Version number drift	Created sync script	scripts/sync-versions.sh
11	Rate limiting (already exists)	Verified implementation	middleware/rate-limiter.ts
12	Lambda cold starts	Created optimization utilities	optimizations/cold-start.ts
13	Connection pooling	Created verification utilities	db/connection-pool.ts
14	Missing E2E tests	Created Playwright setup	e2e/playwright.config.ts
15	Admin component types	Verified - no issues found	N/A
16	Documentation sync	Handled by sync-versions.sh	scripts/sync-versions.sh

8.1 Iterator Downlevel Issues (Fixed)

Location: packages/infrastructure/lambda/shared/services/localization.ts

Issue: `for-of` loop on `Map.keys()` caused downlevel iteration errors.

Fix: Replaced with `forEach` and array collection.

8.2 Set Spreading Issues (Fixed)

Location: `packages/infrastructure/lambda/shared/services/result-merging.ts`

Issue: Spreading Sets caused TypeScript iterator errors.

Fix: Used arrays instead of spreading Sets.

8.3 Buffer to Blob Conversion (Fixed)

Location: `packages/infrastructure/lambda/shared/services/voice-video.ts`

Issue: Buffer to Blob conversion type error.

Fix: Wrapped Buffer in `Uint8Array` before creating Blob.

8.4 RadiantError Cause Property (Fixed)

Location: `packages/shared/src/errors/index.ts`

Issue: `cause` property override caused TypeScript error.

Fix: Renamed to `originalCause`.

8.5 Swift Type References (Fixed)

Location: Swift Deployer services

Issue: `SeedDataService` and `ParameterEditorView` had incorrect type references.

Fix: Updated to use correct `AWSService.shared` and `CredentialService` interfaces.

Action Items Summary

Priority	Count	Estimated Effort
Critical	0	0 hours
High	0	0 hours
Medium	0	0 hours
Low	0	0 hours
Deferred	1	24-40 hours (awaiting AWS SDK Swift stability)

Total Active Technical Debt: 0 hours

All Issues Resolved (December 2024): - SSR createContext incompatibility (next-themes fix) - Lambda handler tests (thermal-state) - Swift silent error handling (RadiantLogger) - Swift service tests (AWSService, CDKService) - Database type casting (typed-query.ts) - AWS SDK Field narrowing (field-guards.ts) - S3 bucket configuration (RadiantConfig) - E2E Playwright tests - Migration validation script - Version sync script - Credential sanitization - Rate limiting verification - Cold start optimization - Connection pooling verification - CDK stack dependency docs - Component type safety audit

Contributing

When adding new technical debt: 1. Add entry with appropriate priority 2. Include file locations 3. Provide code examples 4. Estimate effort 5. Update action items summary

When resolving technical debt: 1. Move to “Resolved Issues” section 2. Document the fix 3. Update action items summary