Contents

User Data Service (UDS) Administration Guide **2**
Table of Contents . . . 2
1. Overview . . . 2
Why UDS vs Cortex? . . . 3
2. Architecture . . . 3
2.1 System Overview . . . 3
2.2 Database Tables . . . 4
3. Tiered Storage . . . 4
3.1 Tier Overview . . . 4
3.2 Automatic Transitions . . . 5
3.3 Configuration . . . 5
3.4 Manual Operations . . . 5
4. Data Types . . . 5
4.1 Conversations . . . 5
4.2 Messages . . . 6
4.3 Uploads . . . 6
5. Encryption . . . 7
5.1 Architecture . . . 7
5.2 Algorithm Details . . . 7
5.3 Key Rotation . . . 8
5.4 Configuration . . . 8
6. Audit System . . . 8
6.1 Features . . . 8
6.2 Audit Entry Structure . . . 8
6.3 Event Categories . . . 9
6.4 Merkle Verification . . . 9
6.5 Export . . . 10
7. Upload Management . . . 10
7.1 Upload Flow . . . 10
7.2 Upload States . . . 10
7.3 API Endpoints . . . 10
8. GDPR Compliance . . . 11
8.1 Right to Erasure . . . 11
8.2 Erasure Scopes . . . 12
8.3 Create Erasure Request . . . 12
8.4 Erasure Status . . . 12
8.5 Verification . . . 12
9. Admin API Reference . . . 13
Dashboard . . . 13
Configuration . . . 13
Conversations . . . 13
Uploads . . . 13
Audit . . . 13
Tiers . . . 13
Erasure . . . 14
Encryption . . . 14

Statistics . . . 14
10. Admin Dashboard . . . 14
10.1 Overview Tab . . . 14
10.2 Audit Log Tab . . . 14
10.3 GDPR Erasure Tab . . . 15
10.4 Configuration Tab . . . 15
11. Configuration . . . 15
11.1 Environment Variables . . . 15
11.2 Per-Tenant Configuration . . . 15
12. Monitoring . . . 15
12.1 Key Metrics . . . 15
12.2 CloudWatch Alarms . . . 15
12.3 Housekeeping . . . 16
13. Troubleshooting . . . 16
13.1 Common Issues . . . 16
13.2 Support . . . 16
Document History . . . 16

User Data Service (UDS) Administration Guide

Version: 1.0.0
Last Updated: January 24, 2026
RADIANT Version: 5.52.18

Table of Contents

1. Overview
2. Architecture
3. Tiered Storage
4. Data Types
5. Encryption
6. Audit System
7. Upload Management
8. GDPR Compliance
9. Admin API Reference
10. Admin Dashboard
11. Configuration
12. Monitoring
13. Troubleshooting

---

1. Overview

The User Data Service (UDS) is RADIANT's dedicated system for storing, managing, and securing user-generated content at scale (1M+ concurrent users). It provides:

- **Tiered Storage**: Hot → Warm → Cold → Glacier automatic data lifecycle
- **End-to-End Encryption**: AES-256-GCM with KMS key management
- **Tamper-Evident Audit**: Merkle chain for compliance verification
- **GDPR Compliance**: Right-to-erasure with multi-tier deletion
- **File Handling**: Virus scanning, text extraction, semantic search

Why UDS vs Cortex?

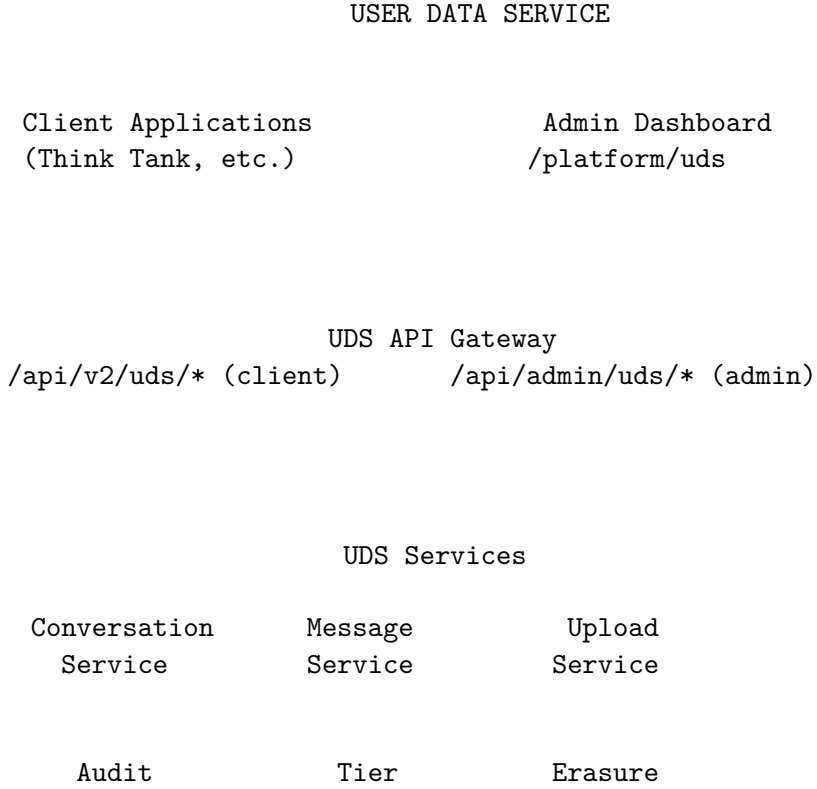
System	Purpose	Data
Cortex	AI Memory	Knowledge graphs, semantic memory, ghost vectors
UDS	User Data	Conversations, messages, uploads, audit logs

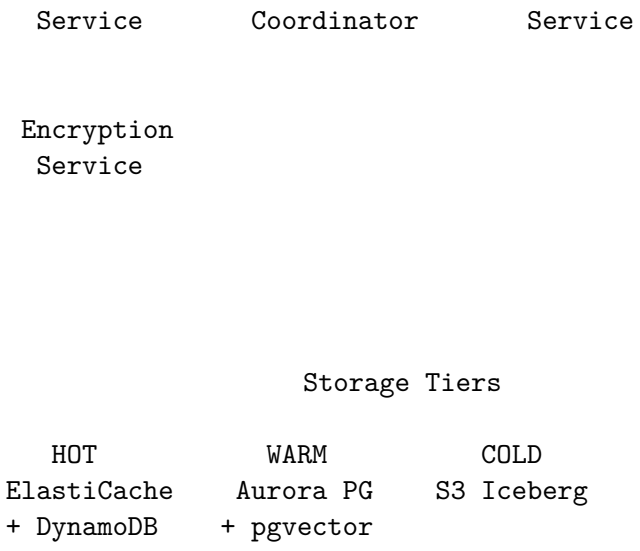
UDS is optimized for **time-series CRUD** operations, while Cortex is optimized for **graph queries** and **semantic search**.

---

2. Architecture

2.1 System Overview





2.2 Database Tables

Table	Purpose
`uds_config`	Per-tenant configuration
`uds_encryption_keys`	Encryption key registry
`uds_conversations`	Conversation metadata
`uds_messages`	Encrypted message content
`uds_message_attachments`	Inline attachments
`uds_uploads`	File upload metadata
`uds_upload_chunks`	Chunked upload tracking
`uds_audit_log`	Tamper-evident audit trail
`uds_audit_merkle_tree`	Merkle tree checkpoints
`uds_export_requests`	Compliance data exports
`uds_erasure_requests`	GDPR deletion requests
`uds_tier_transitions`	Data movement history
`uds_data_flow_metrics`	Tier health metrics
`uds_search_index`	Full-text + semantic search

---

3. Tiered Storage

3.1 Tier Overview

Tier	Storage	Retention	Access Pattern	Latency
Hot	ElastiCache + DynamoDB	0-24 hours	Real-time	<10ms
Warm	Aurora PostgreSQL	1-90 days	Active	<100ms
Cold	S3 Iceberg	90 days - 7 years	Rare	1-10s

Tier	Storage	Retention	Access Pattern	Latency
Glacier	S3 Glacier	7+ years	Archive only	1-12h

3.2 Automatic Transitions

Data automatically moves between tiers based on access patterns:

```
Hot (24h) → Warm (90d) → Cold (7y) → Glacier
                ↑_____________|
                   (retrieval)
```

Transition Rules: - **Hot → Warm**: Conversation not accessed for 24 hours - **Warm → Cold**: Conversation archived AND not accessed for 90 days - **Cold → Glacier**: Data older than 7 years (compliance retention) - **Cold → Warm**: Manual retrieval request

3.3 Configuration

```
// Per-tenant tier configuration
{
  hotSessionTtlSeconds: 14400,      // 4 hours default session TTL
  hotMessageTtlSeconds: 86400,      // 24 hours default message TTL
  warmRetentionDays: 90,            // 90 days in warm tier
  coldRetentionYears: 7,            // 7 years compliance retention
}
```

3.4 Manual Operations

Trigger Hot → Warm Promotion:

```
POST /api/admin/uds/tiers/promote
```

Trigger Warm → Cold Archival:

```
POST /api/admin/uds/tiers/archive
```

Retrieve from Cold to Warm:

```
POST /api/admin/uds/tiers/retrieve
Content-Type: application/json

{
  "resourceIds": ["uuid-1", "uuid-2"]
}
```

---

4. Data Types

4.1 Conversations

Conversations are the primary container for user interactions.

```typescript
interface Conversation {
  id: string;
  tenantId: string;
  userId: string;
  title: string;
  modelId: string;
  messageCount: number;
  totalInputTokens: number;
  totalOutputTokens: number;
  totalCostCredits: number;
  status: 'active' | 'archived' | 'deleted';
  currentTier: 'hot' | 'warm' | 'cold' | 'glacier';
  // Time Machine support
  parentConversationId?: string;
  forkPointMessageId?: string;
  branchName?: string;
  // Collaboration
  isShared: boolean;
  sharedWithUserIds: string[];
}
```

Features: - **Time Machine**: Fork conversations at any message - **Checkpoints**: Save named snapshots - **Collaboration**: Share with other users - **Tagging**: Custom metadata tags

4.2 Messages

Messages are encrypted at rest and contain the actual conversation content.

```typescript
interface Message {
  id: string;
  conversationId: string;
  role: 'system' | 'user' | 'assistant' | 'tool';
  content: string;  // Decrypted on read
  sequenceNumber: number;
  inputTokens: number;
  outputTokens: number;
  costCredits: number;
  // Editing
  isEdited: boolean;
  editCount: number;
  // Feedback
  userRating: number;  // 1-5
  flagged: boolean;
}
```

4.3 Uploads

Uploads support multiple file formats with automatic processing.

Supported Content Types:

Category	Extensions
Documents	pdf, docx, doc, xlsx, xls, csv, txt, md, json, xml
Images	png, jpg, jpeg, gif, webp, svg, bmp, tiff
Audio	mp3, wav, ogg, m4a
Video	mp4, webm, mov
Archives	zip, tar, gz

Processing Pipeline: 1. **Quarantine**: File uploaded to quarantine bucket 2. **Virus Scan**: ClamAV Lambda checks for malware 3. **Promotion**: Clean files moved to main bucket 4. **Text Extraction**: Textract/Tika extracts content 5. **Embedding**: Vector embedding for semantic search 6. **Thumbnail**: Generate preview images

---

5. Encryption

5.1 Architecture

UDS uses **envelope encryption** with AWS KMS:

```
                    ENCRYPTION HIERARCHY


    AWS KMS Master Key
    (alias/radiant-uds-master)




    Data Encryption Keys (DEKs)
    - Per-tenant key (default)
    - Per-user key (optional, high-security)
    - Rotated every 90 days




    Encrypted Data
    - Messages: AES-256-GCM with per-message IV
    - Uploads: S3 SSE-KMS
    - Attachments: AES-256-GCM
```

5.2 Algorithm Details

- **Algorithm**: AES-256-GCM

- **IV Length**: 96 bits (12 bytes)
- **Auth Tag Length**: 128 bits (16 bytes)
- **Key Spec**: AES_256

5.3 Key Rotation

Automatic Rotation: - Keys are automatically rotated every 90 days - Old keys remain available for decryption - New data uses the latest key version

Manual Rotation:

```
POST /api/admin/uds/encryption/rotate
Content-Type: application/json

{
  "userId": "optional-user-id-for-per-user-key"
}
```

5.4 Configuration

```
{
  encryptionEnabled: true,
  encryptionAlgorithm: 'AES-256-GCM',
  perUserEncryptionKeys: false,  // Enable for high-security tenants
}
```

---

6. Audit System

6.1 Features

- **Append-Only**: Entries cannot be modified or deleted
- **Merkle Chain**: Each entry links to previous via hash
- **Tamper-Evident**: Verification detects any modification
- **Compliance Ready**: GDPR, HIPAA, SOC2 compatible

6.2 Audit Entry Structure

```
interface AuditEntry {
  id: string;
  tenantId: string;
  userId: string;

  // Event
  eventType: string;           // e.g., 'conversation_created'
  eventCategory: string;       // e.g., 'conversation'
  eventSeverity: string;       // 'debug' | 'info' | 'warning' | 'error' | 'critical'

  // Resource
  resourceType: string;
```

```
  resourceId: string;

  // Action
  action: string;              // 'create' | 'read' | 'update' | 'delete'
  actionDetails: object;

  // Merkle Chain
  merkleHash: string;          // SHA-256 hash of entry + previous hash
  previousMerkleHash: string;
  sequenceNumber: number;

  // Request Context
  requestId: string;
  ipAddress: string;
  userAgent: string;

  createdAt: Date;
}
```

6.3 Event Categories

Category	Events
`auth`	login, logout, token_refresh
`conversation`	created, updated, deleted, forked, archived
`message`	created, updated, deleted, flagged
`upload`	initiated, completed, downloaded, deleted
`gdpr`	erasure_requested, erasure_completed
`system`	tier_transition, housekeeping

6.4 Merkle Verification

Verify Chain Integrity:

```
POST /api/admin/uds/audit/verify
Content-Type: application/json

{
  "fromSequence": 1,
  "toSequence": 1000
}
```

Response:

```
{
  "isValid": true,
  "treeRoot": "a1b2c3...",
  "entriesVerified": 1000,
  "errors": []
}
```

6.5 Export

Export Audit Log:

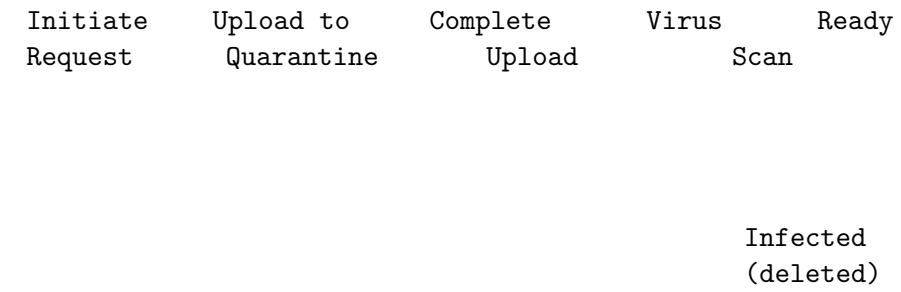
```
POST /api/admin/uds/audit/export
Content-Type: application/json

{
  "startDate": "2025-01-01T00:00:00Z",
  "endDate": "2025-12-31T23:59:59Z",
  "format": "json"  // or "csv"
}
```

---

7. Upload Management

7.1 Upload Flow



7.2 Upload States

Status	Description
`pending`	Presigned URL generated, awaiting upload
`scanning`	Virus scan in progress
`clean`	Passed virus scan, being processed
`infected`	Failed virus scan, file deleted
`processing`	Text extraction/thumbnail in progress
`ready`	Fully processed, available for download
`failed`	Processing failed
`deleted`	Soft deleted by user/admin

7.3 API Endpoints

Initiate Upload:

```
POST /api/v2/uds/uploads/initiate
Content-Type: application/json

{
```

```
  "originalFilename": "document.pdf",
  "mimeType": "application/pdf",
  "fileSizeBytes": 1048576,
  "conversationId": "optional-uuid"
}
```

Response:

```
{
  "uploadId": "uuid",
  "presignedUrl": "https://s3...",
  "expiresAt": "2025-01-24T08:00:00Z",
  "maxSizeBytes": 104857600
}
```

Complete Upload:

```
POST /api/v2/uds/uploads/{uploadId}/complete
Content-Type: application/json

{
  "sha256Hash": "abc123..."
}
```

Get Download URL:

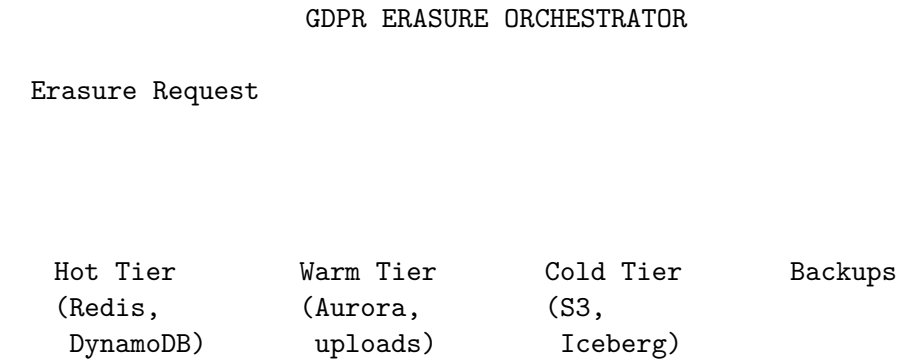
```
GET /api/v2/uds/uploads/{uploadId}/download
```

---

8. GDPR Compliance

8.1 Right to Erasure

UDS implements GDPR Article 17 (Right to Erasure) with multi-tier deletion:



```
Verification
Hash Generated
```

8.2 Erasure Scopes

Scope	Description
`user`	Delete all data for a specific user
`conversation`	Delete a specific conversation
`tenant`	Delete all data for entire tenant

8.3 Create Erasure Request

```
POST /api/admin/uds/erasure
Content-Type: application/json
```

```
{
  "scope": "user",
  "userId": "uuid-of-user",
  "eraseConversations": true,
  "eraseMessages": true,
  "eraseUploads": true,
  "eraseAuditLog": false,     // Usually keep for compliance
  "eraseFromBackups": false,   // Expensive, requires manual intervention
  "anonymizeRemaining": true,  // Anonymize data we can't delete
  "legalBasis": "gdpr_article_17",
  "legalReference": "User request #12345"
}
```

8.4 Erasure Status

Status	Description
`pending`	Request created, not yet started
`processing`	Actively deleting data
`completed`	All tiers processed successfully
`failed`	Error occurred, may be partially complete
`partial`	Some tiers completed, others pending

8.5 Verification

Each completed erasure generates a **verification hash** that proves: - What was deleted - When it was deleted - The scope of deletion

This hash is stored in the audit log for compliance proof.

---

9. Admin API Reference

Base URL: `/api/admin/uds`

Dashboard

Method	Endpoint	Description
GET	`/dashboard`	Full dashboard with health, stats, config

Configuration

Method	Endpoint	Description
GET	`/config`	Get tenant configuration
PUT	`/config`	Update tenant configuration

Conversations

Method	Endpoint	Description
GET	`/conversations`	List conversations with filters
GET	`/conversations/{id}`	Get conversation details
DELETE	`/conversations/{id}`	Delete conversation
GET	`/conversations/{id}/messages`	List messages

Uploads

Method	Endpoint	Description
GET	`/uploads`	List uploads with filters
GET	`/uploads/{id}`	Get upload details
DELETE	`/uploads/{id}`	Delete upload

Audit

Method	Endpoint	Description
GET	`/audit`	List audit entries with filters
POST	`/audit/verify`	Verify Merkle chain integrity
POST	`/audit/export`	Export audit log
GET	`/audit/merkle-trees`	List Merkle trees

Tiers

Method	Endpoint	Description
GET	`/tiers`	Get tier health status
GET	`/tiers/metrics`	Get tier metrics
POST	`/tiers/promote`	Trigger Hot → Warm promotion
POST	`/tiers/archive`	Trigger Warm → Cold archival
POST	`/tiers/retrieve`	Retrieve from Cold to Warm
POST	`/tiers/housekeeping`	Run housekeeping tasks

Erasure

Method	Endpoint	Description
GET	`/erasure`	List erasure requests
POST	`/erasure`	Create erasure request
GET	`/erasure/{id}`	Get erasure request details
DELETE	`/erasure/{id}`	Cancel pending erasure

Encryption

Method	Endpoint	Description
GET	`/encryption/keys`	List encryption keys
POST	`/encryption/rotate`	Rotate encryption key

Statistics

Method	Endpoint	Description
GET	`/stats`	Get UDS statistics

---

10. Admin Dashboard

Access the UDS Admin Dashboard at: **Admin Dashboard → Platform → UDS**

10.1 Overview Tab

- **Tier Health**: Real-time status of all storage tiers
- **Quick Actions**: Promote, archive, housekeeping buttons
- **Statistics**: Conversation, message, upload, audit counts
- **Distribution**: Visual breakdown of data across tiers

10.2 Audit Log Tab

- **Filterable Log**: Filter by category, event type, user
- **Merkle Verification**: Visual indicator of chain integrity
- **Export**: Download audit log for compliance

10.3 GDPR Erasure Tab

- **Request List**: All erasure requests with status
- **Create Request**: Form to initiate new erasure
- **Progress Tracking**: Per-tier deletion status

10.4 Configuration Tab

- **Tier Settings**: TTL and retention configuration
- **Security Settings**: Encryption, virus scanning status
- **Upload Settings**: Size limits, allowed types
- **GDPR Settings**: Auto-delete, anonymization settings

---

11. Configuration

11.1 Environment Variables

Variable	Description	Default
`UDS_KMS_KEY_ALIAS`	KMS master key alias	`alias/radiant-uds-master`
`UDS_UPLOAD_BUCKET`	Main upload S3 bucket	`radiant-uds-uploads`
`UDS_QUARANTINE_BUCKET`	Quarantine S3 bucket	`radiant-uds-quarantine`
`UDS_HOT_TTL_SECONDS`	Default hot tier TTL	`86400`
`UDS_WARM_RETENTION_DAYS`	Default warm retention	`90`
`UDS_COLD_RETENTION_YEARS`	Cold tier retention	`7`

11.2 Per-Tenant Configuration

All settings can be overridden per-tenant via the `uds_config` table or Admin API.

---

12. Monitoring

12.1 Key Metrics

Metric	Description	Alert Threshold
`uds.hot.item_count`	Items in hot tier	>10,000
`uds.warm.storage_bytes`	Warm tier storage	>100GB
`uds.cold.storage_bytes`	Cold tier storage	>1TB
`uds.hot.cache_hit_rate`	Cache efficiency	<90%
`uds.tier.transition_errors`	Failed transitions	>0
`uds.upload.scan_failures`	Virus scan failures	>0

12.2 CloudWatch Alarms

Recommended alarms: - Hot tier item count exceeds threshold - Tier transition error rate - Upload quarantine backup - Audit chain verification failure

12.3 Housekeeping

Run housekeeping regularly to: - Promote data between tiers - Clean up deleted items - Update tier metrics

Manual Trigger:

`POST /api/admin/uds/tiers/housekeeping`

Scheduled: EventBridge rule runs hourly

---

13. Troubleshooting

13.1 Common Issues

Upload Stuck in "Scanning": - Check ClamAV Lambda health - Verify quarantine bucket permissions - Check CloudWatch logs for scan errors

High Hot Tier Item Count: - Verify TTL configuration - Check if promotion job is running - Review access patterns (frequently accessed data stays hot)

Merkle Chain Verification Failed: - Do NOT attempt to fix manually - Contact security team immediately - Preserve audit log for investigation

Erasure Request Failed: - Check per-tier status for specific failure - Review CloudWatch logs - Retry with smaller scope if needed

13.2 Support

For UDS issues: 1. Check CloudWatch logs: `/aws/lambda/radiant-uds-*` 2. Review tier health in Admin Dashboard 3. Contact platform team with request ID

Document History

Version	Date	Changes
1.0.0	2026-01-24	Initial release

This document is part of the RADIANT Platform Documentation.