Audit Architecture

Audit Architecture

Version: 1.0 **Date:** 2026-02-21 **Status:** Approved **Owner:** Database Architect

---

Overview

Drop's audit system records all significant user actions for compliance (PSD2, GDPR, AML) and security monitoring. The `audit_log` table is the central audit store, designed for append-only writes with indexed queries for investigation and reporting.

Regulatory drivers:

- **PSD2 (Betalingstjenesteloven):** Art. 94 requires payment service providers to maintain records of payment transactions for at least 5 years
- **GDPR (Personopplysningsloven):** Art. 5(2) accountability principle -- demonstrate compliance
- **AML (Hvitvaskingsloven):** Section 30 requires retention of all customer due diligence and transaction records
- **Finansavtaleloven:** Section 3-53 requires complaint handling audit trail

---

Audit Log Table Design

Schema

```
CREATE TABLE audit_log (
    id          TEXT PRIMARY KEY,           -- Prefixed ID (e.g., 'aud_a1b2c3...')
    timestamp   TEXT DEFAULT CURRENT_TIMESTAMP,
    user_id     TEXT REFERENCES users(id),  -- NULL for unauthenticated events
    action      TEXT NOT NULL,              -- Event type (e.g., 'auth.login')
    resource_type TEXT,                     -- Entity type (e.g., 'transaction')
```

```
    resource_id TEXT,                       -- Entity ID (e.g., 'tx_abc123')
    details     TEXT,                       -- JSON string with event-specific data
    ip_address  TEXT,                       -- Client IP (from X-Forwarded-For)
    user_agent  TEXT,                       -- Browser/app user agent
    request_id  TEXT                        -- Correlation ID for multi-event requests
);

CREATE INDEX idx_audit_log_user ON audit_log(user_id);
CREATE INDEX idx_audit_log_action ON audit_log(action);
-- Note: idx_audit_log_timestamp is planned but not yet implemented in db.ts
```

Design Rationale

Column	Design Decision
`id`	TEXT with prefix for consistency with all other Drop tables
`timestamp`	TIMESTAMPTZ (PostgreSQL 16 — ADR-014; all environments)
`user_id`	Nullable FK -- some events (failed login, rate limit hit) occur before authentication
`action`	Dot-notation namespace (e.g., `auth.login`, `transaction.create`) for hierarchical filtering
`resource_type` + `resource_id`	Generic resource reference avoids polymorphic FKs while enabling resource-specific queries
`details`	JSON TEXT for flexible, event-specific metadata without schema changes per event type
`request_id`	Correlates multiple audit entries from a single API request (e.g., transaction creation generates audit + notification)

Audit Event Flow

```
flowchart TD
    A[User Action] --> B[API Route Handler]
    B --> C{Action Type}

    C -->|Authentication| D[Auth Events]
    C -->|Transaction| E[Financial Events]
    C -->|Settings/Profile| F[Account Events]
```

```
    C -->|Admin/Compliance| G[Admin Events]

    D --> H[Write audit_log]
    E --> H
    F --> H
    G --> H

    H --> I[Primary Indexes]
    I --> J[idx_audit_log_user<br/>User investigation]
    I --> K[idx_audit_log_timestamp<br/>Time-range queries]
    I --> L[idx_audit_log_action<br/>Event type filtering]

    H --> M{Severity Check}
    M -->|Critical| N[AML Alert Pipeline]
    M -->|Normal| O[Stored for review]
    N --> P[aml_alerts table]
```

```
sequenceDiagram
    participant User
    participant API as API Handler
    participant Audit as Audit Logger
    participant DB as Database
    participant AML as AML Monitor

    User->>API: POST /transactions/remittance
    API->>API: Validate request
    API->>DB: BEGIN transaction
    API->>DB: UPDATE bank_accounts (debit)
    API->>DB: INSERT transactions
    API->>Audit: Log 'transaction.create'
    Audit->>DB: INSERT audit_log
    API->>DB: COMMIT

    Audit->>AML: Check transaction patterns
    AML->>DB: Query recent transactions for user
    alt Suspicious pattern detected
        AML->>DB: INSERT aml_alerts
    end
```

Audit Event Types

Action	Category	Trigger	Logged Details
`auth.login`	Authentication	Successful BankID login	`{method: "bankid", provider: "bankid"}`
`auth.login.failed`	Authentication	Failed login attempt	`{reason: "invalid_credentials", email: "..."}`
`auth.logout`	Authentication	User logout	`{sessions_revoked: N}`
`auth.session.created`	Authentication	New session created	`{session_id: "ses_..."}`
`auth.session.revoked`	Authentication	Session revoked	`{session_id: "ses_..."}`
`auth.token.refreshed`	Authentication	JWT token refreshed	`{new_session_id: "ses_..."}`
`transaction.create`	Financial	Remittance or QR payment created	`{type, amount, currency, fee, recipient_id/merchant_id}`
`transaction.complete`	Financial	Transaction marked completed	`{transaction_id: "tx_..."}`
`transaction.fail`	Financial	Transaction marked failed	`{transaction_id, reason}`
`qr_payment.create`	Financial	QR payment executed	`{amount, merchant_id, fee}`
`bank_account.link`	Account	Bank account linked via AISP	`{bank_name, last4_account}`
`bank_account.balance_sync`	Account	Balance refreshed from AISP	`{bank_account_id, balance}`
`recipient.create`	Account	New recipient added	`{country, currency}`
`recipient.delete`	Account	Recipient removed	`{recipient_id}`
`settings.update`	Account	User settings changed	`{changed_fields: ["currency","language"]}`
`merchant.register`	Account	Merchant profile created	`{business_name, org_number}`
`kyc.status_change`	Compliance	KYC status updated	`{old_status, new_status, method}`
`consent.granted`	Compliance	GDPR consent given	`{consent_type, ip_address}`
`consent.withdrawn`	Compliance	GDPR consent withdrawn	`{consent_type, ip_address}`
`dsar.export`	Compliance	Data export request completed	`{request_id}`
`dsar.erasure`	Compliance	Account deletion requested	`{request_id}`
`complaint.created`	Compliance	Customer complaint filed	`{category, complaint_id}`
`complaint.resolved`	Compliance	Complaint resolved	`{complaint_id, resolution_days}`

Action	Category	Trigger	Logged Details
`aml.alert_created`	AML	Suspicious activity detected	`{alert_type, severity, transaction_id}`
`aml.alert_resolved`	AML	AML alert investigated and resolved	`{alert_id, resolution}`
`str.filed`	AML	STR submitted to authorities	`{str_id, reference_number}`
`screening.completed`	AML	PEP/sanctions screening done	`{screening_type, result}`
`user.deleted`	Account	User account deleted (GDPR)	`{reason: "gdpr_erasure"}`
`rate_limit.exceeded`	Security	Rate limit hit	`{endpoint, ip_address, limit}`
`card.created`	Account	Card created (FUTURE)	`{type, last_four}`
`card.frozen`	Account	Card frozen (FUTURE)	`{card_id}`
`card.cancelled`	Account	Card cancelled (FUTURE)	`{card_id}`

Tamper Detection

Hash Chain Mechanism

To ensure audit log integrity (detect unauthorized modifications or deletions), the audit system should implement a hash chain:

```
flowchart LR
    E1[Entry 1<br/>hash = SHA256<br/>data + genesis] --> E2[Entry 2<br/>hash = SHA256<br/>data + E1.hash]
    E2 --> E3[Entry 3<br/>hash = SHA256<br/>data + E2.hash]
    E3 --> E4[Entry N<br/>hash = SHA256<br/>data + E(N-1).hash]
```

Proposed implementation:

Add a `chain_hash` column to `audit_log`:

```
ALTER TABLE audit_log ADD COLUMN chain_hash TEXT;
```

Each entry's hash is computed as:

```
chain_hash = SHA256(
    timestamp || user_id || action || resource_type ||
    resource_id || details || previous_chain_hash
)
```

Verification: Walk the chain from the first entry, recomputing each hash. A mismatch indicates tampering. This can be run as a periodic integrity check job.

Current status: Not yet implemented. The audit table stores events without tamper detection. Hash chain is a Phase 3 enhancement (pre-production launch).

Alternative: Append-Only with Write-Once Storage

For production, audit logs should be replicated to write-once storage (AWS S3 Object Lock / Glacier Vault Lock) within minutes of creation. This provides:

- Immutability guarantee independent of database access
- External verification point
- Compliance with regulatory requirements for tamper-evident audit trails

---

Compliance Requirements

PSD2 Audit Trail

Requirement	Implementation	Status
Record all payment transactions	`transaction.create` event with full details	Implemented
Record authentication events	`auth.login`, `auth.logout` events	Implemented
Record consent actions	`consent.granted`, `consent.withdrawn` events	Implemented
5-year retention	Retention policy defined (see data-lifecycle.md)	Policy defined
Tamper-evident	Hash chain proposed, not yet implemented	Planned

GDPR Audit Trail

Requirement	Implementation	Status
Record data access requests	`dsar.export`, `dsar.erasure` events	Implemented
Record consent changes	`consent.granted`, `consent.withdrawn` events	Implemented
Record data deletion	`user.deleted` event	Implemented
Demonstrate accountability	Full audit trail queryable by user_id	Implemented

AML Audit Trail

Requirement	Implementation	Status
Record suspicious activity alerts	`aml.alert_created` event	Implemented
Record STR filings	`str.filed` event	Implemented
Record screening results	`screening.completed` event	Implemented
Record KYC status changes	`kyc.status_change` event	Implemented

Log Retention and Searchability

Query Patterns

Query	Use Case	Index Used	Example SQL
All events for a user	Investigation, DSAR	`idx_audit_log_user`	`SELECT * FROM audit_log WHERE user_id = ? ORDER BY timestamp DESC`
Events in time range	Compliance reporting	`idx_audit_log_timestamp`	`SELECT * FROM audit_log WHERE timestamp BETWEEN ? AND ?`
Events by type	Pattern analysis	`idx_audit_log_action`	`SELECT * FROM audit_log WHERE action = 'transaction.create'`
Events for a resource	Transaction audit trail	Sequential scan (consider composite index)	`SELECT * FROM audit_log WHERE resource_type = 'transaction' AND resource_id = ?`
Recent events globally	Dashboard, monitoring	`idx_audit_log_timestamp`	`SELECT * FROM audit_log ORDER BY timestamp DESC LIMIT 50`

Retention Tiers

Tier	Age	Storage	Access
Hot	0-3 months	Primary database, fully indexed	Real-time queries
Warm	3-12 months	Primary database, indexed	Standard queries
Cold	1-5 years	Archive storage (S3)	Restored on demand
Purge	5+ years	Deleted	Not available

Cross-References

- **Audit log schema:** DATABASE-SCHEMA.md (audit_log section)
- **Data lifecycle:** data-lifecycle.md (retention periods)
- **Indexing strategy:** indexing-strategy.md (audit log indexes)
- **Compliance status:** COMPLIANCE.md
- **Security architecture:** SECURITY-ARCHITECTURE.md

Revision #6
Created 2026-02-21 05:59:07 UTC by John
Updated 2026-05-23 10:57:19 UTC by John