LLD: Withdrawal Flow

Withdrawal Request Flow (Angrerett)

Purpose

Implements the user's right of withdrawal (angrerett) as required by Norwegian consumer protection law (angrerettloven). Users can submit a withdrawal request to cancel their account or service agreement within the statutory cooling-off period.

Sequence Diagram

```
sequenceDiagram
    participant U as User (App)
    participant API as Drop API
    participant Auth as Auth Middleware
    participant DB as PostgreSQL
    participant Audit as Audit Log

    U->>API: POST /withdrawal { reason, comment }
    API->>Auth: Validate JWT token
    Auth-->>API: user context

    alt Invalid JSON body
        API-->>U: 400 bad_request
    end

    API->>API: Sanitize reason (max 100 chars)
    API->>API: Sanitize comment (max 1000 chars)
    API->>API: Validate reason against VALID_REASONS
```

```
    alt Invalid reason
        API-->>U: 400 validation_error
    end

    API->>DB: INSERT INTO withdrawal_requests (id, user_id, reason, comment)
    DB-->>API: OK

    API->>Audit: Log WITHDRAWAL_REQUEST action
    Audit->>DB: INSERT INTO audit_log

    API-->>U: 201 { success: true, id }
```

Database Schema

withdrawal_requests table

Column	Type	Constraints
id	TEXT	PRIMARY KEY (prefix: `wr_`)
user_id	TEXT	NOT NULL, REFERENCES users(id)
reason	TEXT	Nullable
comment	TEXT	Nullable
status	TEXT	DEFAULT 'pending', CHECK IN ('pending','processing','completed','rejected')
created_at	TIMESTAMPTZ	DEFAULT NOW()

Index: `idx_withdrawal_requests_user` on `user_id`.

Valid Withdrawal Reasons

Value	Description
`not_needed`	User no longer needs the service
`alternative`	User found an alternative service
`not_satisfied`	User is not satisfied with the service

Value	Description
`other`	Other reason (details in comment field)
`""` (empty)	No reason provided

Request Processing

1. **Authentication** -- request must include a valid JWT token (authMiddleware).
2. **Input validation** -- reason is checked against the allowlist; both reason and comment are sanitized via `sanitizeText` with length limits.
3. **Record creation** -- a new `withdrawal_requests` row is inserted with status `pending`.
4. **Audit logging** -- an audit log entry is created with action `WITHDRAWAL_REQUEST`, including the reason and the requester's IP address.

Status Lifecycle

```
pending --> processing --> completed
                      \-> rejected
```

- **pending** -- initial state after user submits request.
- **processing** -- staff/admin has begun reviewing the request.
- **completed** -- withdrawal has been executed, account closed or service cancelled.
- **rejected** -- request was denied (e.g., outside cooling-off period, regulatory hold).

Error States

Scenario	HTTP Status	Error Code
Missing/invalid JWT	401	unauthorized
Malformed JSON body	400	bad_request
Invalid reason value	400	validation_error
Database write failure	500	internal_error

Edge Cases

- **Duplicate requests** -- no uniqueness constraint on user_id; a user can submit multiple withdrawal requests. Business logic should handle deduplication at the review stage.

- **Already deleted user** -- the foreign key on user_id ensures the user must exist. If the user record has `deleted_at` set, the auth middleware should reject the request before it reaches this route.
- **AML retention** -- even after withdrawal is completed, transaction records and AML-related data must be retained for 5 years per hvitvaskingsloven. The data retention cron (`/cron/retention`) handles anonymization after the retention period expires.

Cross-References

- **Angrerettloven** -- Norwegian Act on the Right of Withdrawal (consumer protection).
- **Data retention** -- See `src/drop-api/src/routes/cron.ts` retention endpoint and `docs/architecture/lld/flow-kyc-aml.md` for AML retention requirements.
- **Audit logging** -- See `src/drop-api/src/lib/audit.ts` for audit log implementation.

Revision #6
Created 2026-02-23 11:28:53 UTC by John
Updated 2026-05-23 10:56:25 UTC by John