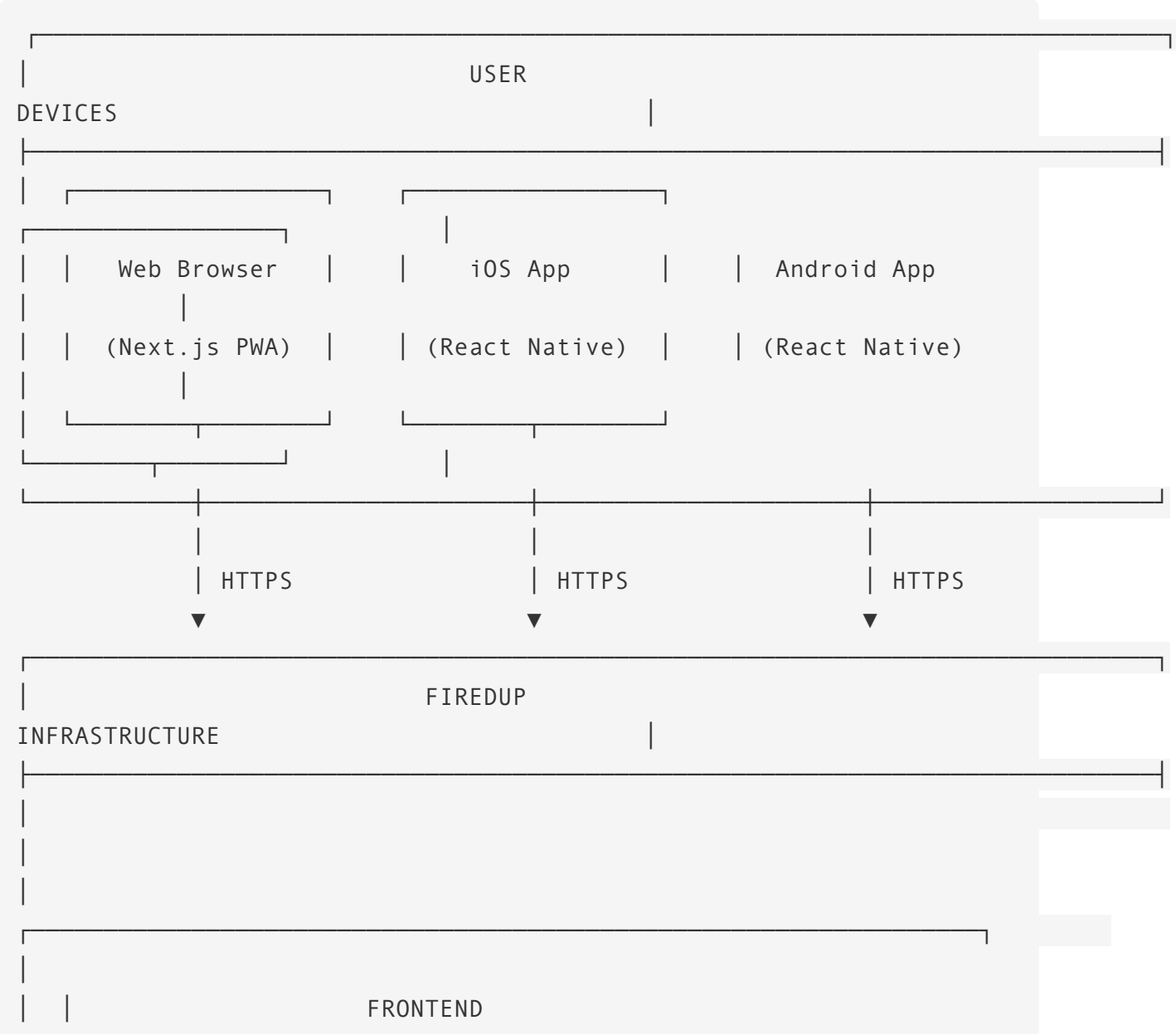
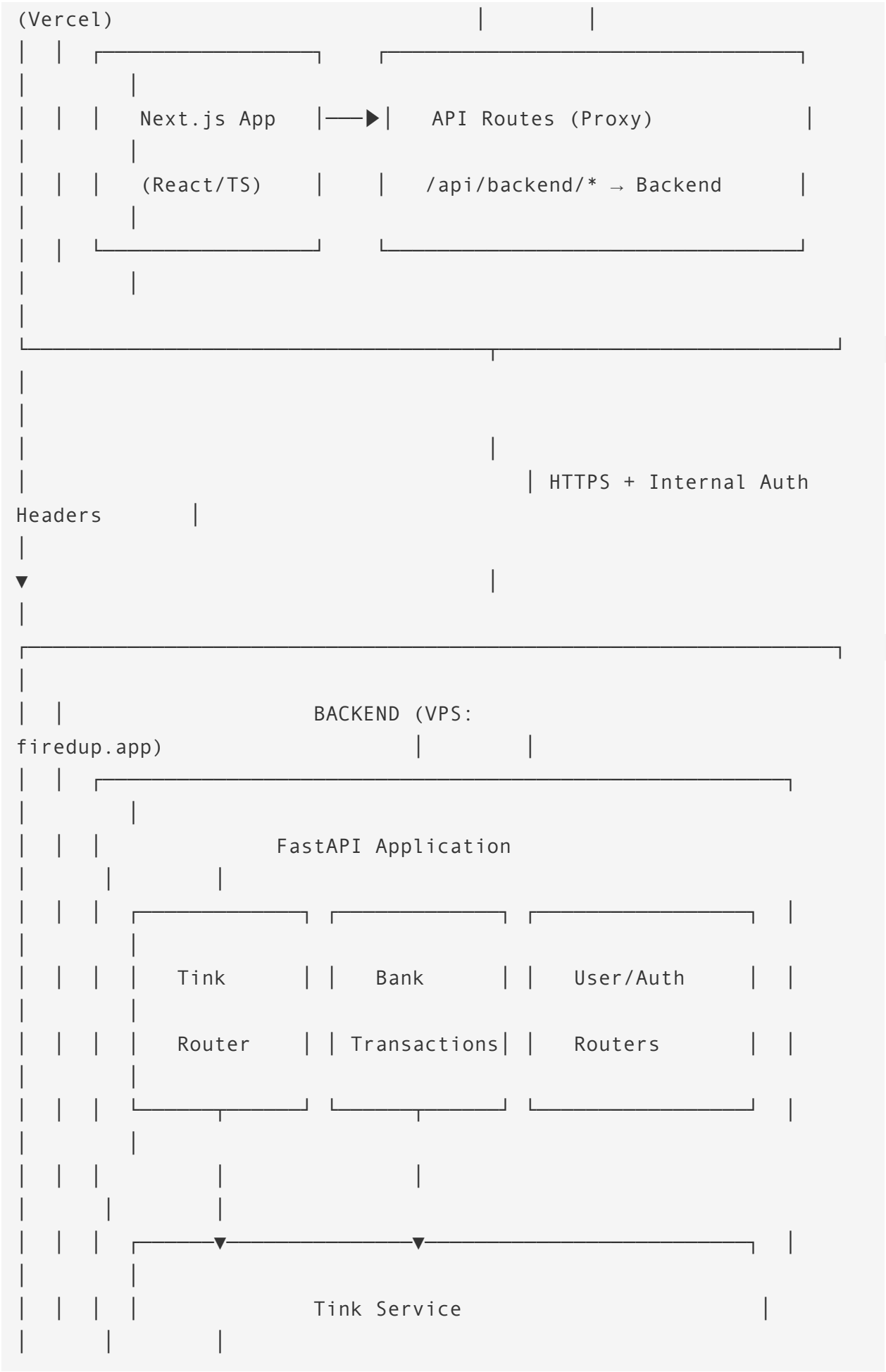
Technical Architecture

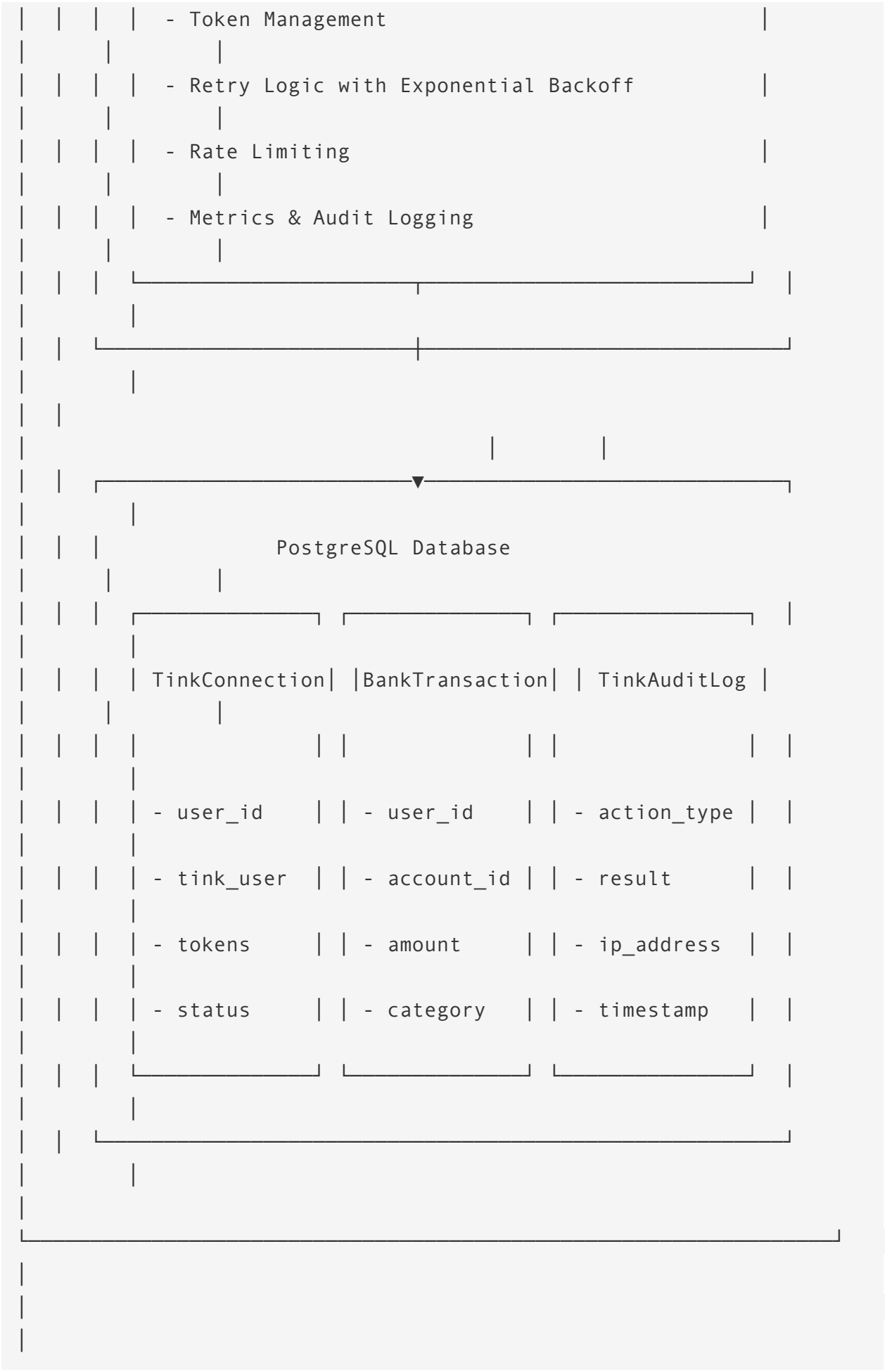
System Overview

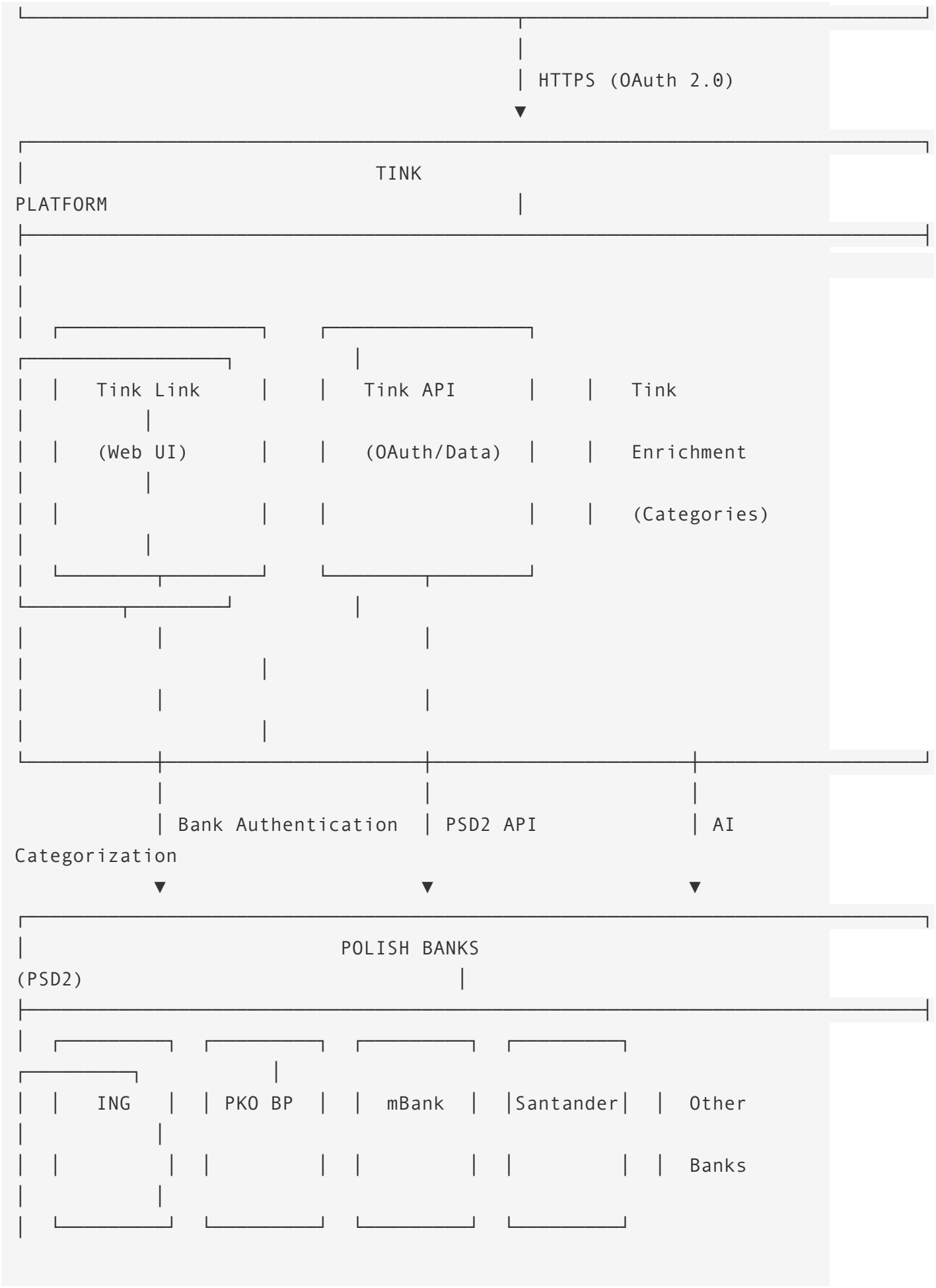
FiredUp is built as a modern SaaS application with a clear separation between frontend, backend, and third-party integrations. The Tink integration follows Tink's recommended Platform model using the Connectivity API v1.

Architecture Diagram




(Vercel)
Next.js App
(React/TS)
API Routes (Proxy)
/api/backend/* → Backend
HTTPS + Internal Auth
Headers
BACKEND (VPS:
firedup.app)
FastAPI Application
Tink
Router
Bank
Transactions
User/Auth
Routers
Tink Service


- Token Management
- Retry Logic with Exponential Backoff
- Rate Limiting
- Metrics & Audit Logging
PostgreSQL Database
TinkConnection
BankTransaction
TinkAuditLog
- user_id
- tink_user
- tokens
- status
- user_id
- account_id
- amount
- category
- action_type
- result
- ip_address
- timestamp


HTTPS (OAuth 2.0)
TINK
PLATFORM
Tink Link
(Web UI)
Tink API
(OAuth/Data)
Tink
Enrichment
(Categories)
Bank Authentication
PSD2 API
AI
Categorization
POLISH BANKS
(PSD2)
ING
PKO BP
mBank
Santander
Other
Banks

```
└──────────┘              |
└──────────────────────────────────────────────────────────────────────────────┘
```

Data Flow: Bank Connection

Step 1: Initiate Connection

```
 User                    FiredUp Web           FiredUp Backend
Tink API
  |                         |
|                         |
  |  Click "Connect"        |
|                         |
  |─────────────────────────▶|
|                         |
  |                         |  POST /banking/tink/
connect                  |
  |
|─────────────────────────▶|                        |
  |                         |                          |  Get Client
Token    |
  |                         |
|──────────────────────▶|
  |                         |
|◀──────────────────────|
  |                         |                          |  Create Tink
User    |
  |                         |
|──────────────────────▶|
  |                         |
|◀──────────────────────|
  |                         |                          |  Get Auth
Code       |
  |                         |
|──────────────────────▶|
  |                         |
|◀──────────────────────|
  |                         |                          |  Store
pending_auth |
```

```
 |                        |                        |  (DB with
state)    |
 |                        |  { tink_link_url }
|                      |
 |
|◀──────────────────────|                       |
 |  Redirect to Tink     |
|                      |
 |◀──────────────────────|
|                      |
```

Step 2: Bank Authentication (via Tink Link)

```
User                  Tink Link             Bank Website
 |                        |                       |
 |  Select Bank           |                       |
 |──────────────────────▶|                        |
 |                        |  Redirect to Bank     |
 |                        |──────────────────────▶|
 |                        |◀──────────────────────|
 |  Login to Bank         |                       |
 |──────────────────────────────────────────────▶|
 |◀──────────────────────────────────────────────|
 |  Grant Consent         |                       |
 |──────────────────────▶|                        |
 |  Redirect back         |                       |
 |◀──────────────────────|                        |
```

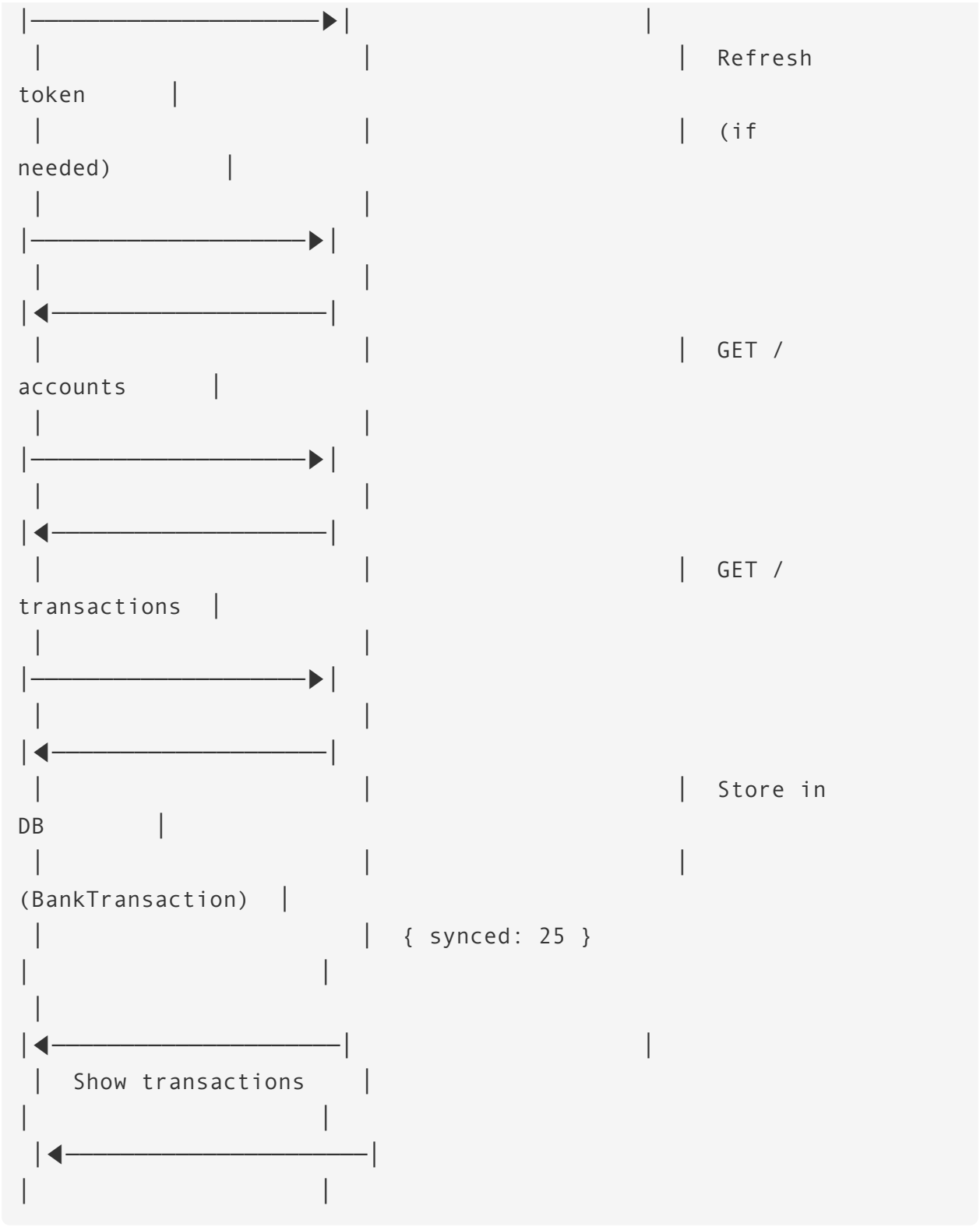
Step 3: Complete Connection

```
User                  FiredUp Web          FiredUp Backend
Tink API
 |                        |
|                      |
 |  Callback with state  |
|                      |
 |──────────────────────▶|
|                      |
 |                        |  POST /banking/tink/
callback               |
```

```
 |                        |  { state: "..." }
|                       |
 |
|──────────────────────▶|                      |
 |                        |                      |  Verify state
(DB)  |
 |                        |                      |  Get stored
auth    |
 |                        |                      |  Exchange for
token |
 |                        |
|──────────────────────▶|
 |                        |
|◀──────────────────────|
 |                        |                      |  Store
TinkConnection|
 |                        |                      |  (encrypted
tokens) |
 |                        |  { success: true }
|                       |
 |
|◀──────────────────────|                      |
 |  Show success          |
|                       |
 |◀──────────────────────|
|                       |
```

Step 4: Data Synchronization

```
 User                  FiredUp Web          FiredUp Backend
Tink API
 |                        |
|                       |
 |  Click "Sync"          |
|                       |
 |──────────────────────▶|
|                       |
 |                        |  POST /banking/transactions/
sync           |
 |
```


Refresh token (if needed)
GET / accounts
GET / transactions
Store in DB (BankTransaction)
{ synced: 25 }
Show transactions

Key Components

Backend Services

Service	File	Responsibility
TinkService	`backend/app/services/tink_service.py`	OAuth flow, token management, API calls
TinkMetricsService	`backend/app/services/tink_metrics_service.py`	Request metrics, performance monitoring
AuditService	`backend/app/services/audit_service.py`	Security audit logging

Database Models

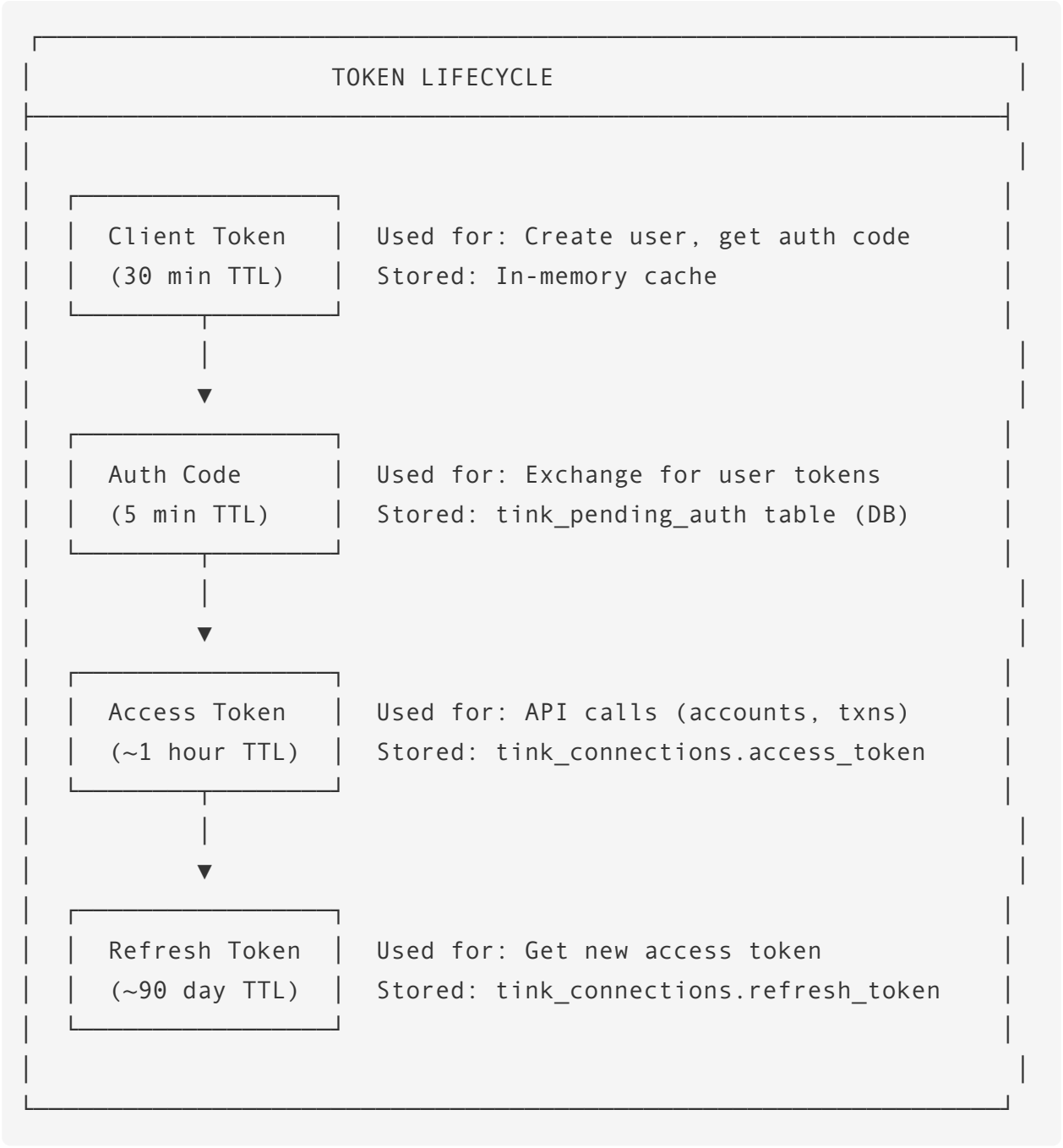
Model	Table	Purpose
TinkConnection	`tink_connections`	Store user's bank connection and tokens
TinkPendingAuth	`tink_pending_auth`	Temporary auth state during OAuth flow
BankTransaction	`bank_transactions`	Imported transactions from banks
TinkAuditLog	`tink_audit_logs`	Security and compliance audit trail

API Endpoints

Endpoint	Method	Purpose
`/banking/tink/connect`	POST	Initiate bank connection
`/banking/tink/callback`	POST	Handle OAuth callback
`/banking/tink/connections`	GET	List user's connections
`/banking/tink/connections/{id}`	DELETE	Disconnect bank
`/banking/transactions/sync`	POST	Sync transactions from Tink
`/banking/transactions`	GET	List imported transactions

Security Architecture

Token Management



State Token Protection (CSRF)

The state token used in the OAuth flow is protected using HMAC-SHA256:

```
 # Generation
state = secrets.token_urlsafe(32)  # Random 32-byte token
signature = hmac.new(
    secret_key.encode(),
    state.encode(),
    hashlib.sha256
).digest()
signed_state = base64.urlsafe_b64encode(state + signature)

# Verification
# 1. Decode and split state and signature
# 2. Recompute HMAC with stored secret
# 3. Compare signatures (constant-time)
# 4. Verify state exists in tink_pending_auth table
# 5. Mark state as used (prevent replay)
```

Infrastructure

Production Environment

Component	Provider	Location
Frontend	Vercel	EU (fra1)
Backend	VPS	EU (Germany)
Database	PostgreSQL	Same VPS
Domain	Cloudflare	DNS/CDN

Environment Variables

Variable	Purpose	Storage
`TINK_CLIENT_ID`	Tink app identifier	Server env
`TINK_CLIENT_SECRET`	Tink app secret	Server env (encrypted)
`TINK_REDIRECT_URI`	OAuth callback URL	Server env

Variable	Purpose	Storage
`DATABASE_URL`	PostgreSQL connection	Server env
`SENTRY_DSN`	Error monitoring	Server env

Monitoring & Observability

Error Monitoring

- **Sentry** for exception tracking
- Automatic alert on error rate spikes
- Stack traces with request context

Metrics

- TinkMetricsService tracks:
  - Request counts by endpoint
  - Response times (p50, p95, p99)
  - Error rates by type
  - Token refresh frequency

Audit Logging

- TinkAuditLog captures:
  - All connection attempts (success/failure)
  - Token refresh operations
  - Data sync operations
  - Disconnection events

Scalability Considerations

Current Design (100-1,000 users)

- Single backend instance
- Direct database writes
- On-demand synchronization

Future Scaling Options (1,000+ users)

- Add background job queue (Celery + Redis)
- Implement scheduled sync jobs
- Add read replicas for database
- Horizontal scaling of API servers

Document Revision

Version	Date	Changes
1.0	February 2026	Initial version