

Contents

SAAS_DECISIONS_EXITON_TRADING.md	1
EXITON 自動投資システム — SaaS 設計方針（決定事項） v0.1	1
1. 現フェーズ（2025時点）の前提	1
SD-TR-001 実行環境	1
SD-TR-002 ユーザー管理	1
2. 将来のSaaS構成（ターゲット像）	2
SD-TR-010 全体構成イメージ	2
3. 認証・認可（Auth）	2
SD-TR-020 認証基盤の候補	2
SD-TR-021 最初の選択肢（優先候補）	2
SD-TR-022 認可	2
4. データベース・永続化	2
SD-TR-030 当面のデータ保存方針	2
SD-TR-031 保存対象	2
5. ログ・監視	3
SD-TR-040 初期段階	3
SD-TR-041 SaaS 化時	3
6. 環境分離・設定管理	3
SD-TR-050 環境レイヤ	3
SD-TR-051 設定管理	3
7. 有料化・プラン設計（ラフな方針）	3
SD-TR-060 収益化の方向性	3
SD-TR-061 課金モデル（候補）	3
8. 今やらないこと（スコープ外宣言）	3
SD-TR-070 現時点で行わないもの	3

SAAS_DECISIONS_EXITON_TRADING.md

EXITON 自動投資システム — SaaS 設計方針（決定事項） v0.1

本ドキュメントは、SAAS_STANDARD_v0.1.md に基づき、EXITON 自動投資システム（ai-signal-chart）における具体的な SaaS 設計方針・当面の決定事項をまとめたものである。

1. 現フェーズ（2025時点）の前提

SD-TR-001 実行環境

- ・現時点では、**ローカル環境での個人利用**を前提とする。
 - Mac mini 上で FastAPI + Streamlit を実行し、ブラウザからローカルネットワーク経由でアクセス。

SD-TR-002 ユーザー管理

- ・現フェーズではユーザーアカウント・ログイン機能は実装しない。
- ・将来的に SaaS 化する際には、外部認証基盤の利用を前提とする（後述）。

2. 将来のSaaS構成（ターゲット像）

SD-TR-010 全体構成イメージ

- フロントエンド：
 - Next.js / React ベース（候補）
 - ホスティング：Vercel
 - バックエンドAPI：
 - Python（FastAPI）を継続利用
 - ホスティング：Railway / Render / Fly.io / Supabase Functions 等を比較検討
 - データベース：
 - ユーザー設定・戦略プリセット・バックテスト履歴：PostgreSQL系（Supabase / Neon）
 - ログ・監視：
 - Sentry / Better Stack 等のログ集約サービスを利用する。
-

3. 認証・認可（Auth）

SD-TR-020 認証基盤の候補

- 自前認証は行わず、以下のいずれかを採用する方針：
 - Supabase Auth
 - Auth0
 - Firebase Auth

SD-TR-021 最初の選択肢（優先候補）

- 小規模個人SaaSとしては **Supabase Auth** が第一候補。
 - DB とセットで運用しやすい
 - ロールベースアクセス・メールログイン等がひととおり揃う

SD-TR-022 認可

- 将来的に有料プランを導入する場合：
 - プラン情報を DB に持ち、
 - 「同時バックテスト数」や「履歴保存期間」等で差別化する。
 - 当面は単一プランを想定し、認可ロジックはシンプルに保つ。
-

4. データベース・永続化

SD-TR-030 当面のデータ保存方針

- ローカル開発中は：
 - シンプルな SQLite / JSON ファイル保存でもよい。
- SaaS 化フェーズでは：
 - Supabase（PostgreSQL）または Neon を採用する方向で検討。

SD-TR-031 保存対象

- ユーザー別に保存が必要になるもの：
 - 戦略プリセット
 - バックテスト設定（銘柄・期間・戦略ID等）
 - バックテスト結果のメタ情報（run_id・サマリ）
 - フルのエクイティカーブ全データは、必要に応じて再計算可能な設計とし、run_id ベースで再実行できるようにしておく
-

5. ログ・監視

SD-TR-040 初期段階

- ・ ローカル開発では：
 - 標準出力ログ+ローカルファイルに保存で十分。
- ・ ログ形式は NFR_EXITON_TRADING_v0.1.md に準拠した構造化ログを意識する。

SD-TR-041 SaaS 化時

- ・ 本番環境では：
 - Sentry などのサービスを利用してエラー収集する。
 - 致命的エラーが一定回数以上発生した場合、通知（メール / Slack / Discord 等）を送る仕組みを検討する。
-

6. 環境分離・設定管理

SD-TR-050 環境レイヤ

- ・ 将来的な環境構成：
 - dev：開発者向け（こうすけ用）
 - staging：本番相当の検証環境
 - prod：一般ユーザー向け本番環境

SD-TR-051 設定管理

- ・ 各環境の設定（APIキー・DB接続文字列・機能フラグ等）は、
 - 環境変数および .env 系ファイルで管理する。
 - ・ ソースコードに機密情報を埋め込まない。
-

7. 有料化・プラン設計（ラフな方針）

SD-TR-060 収益化の方向性

- ・ 自動投資システムそのものを「完全自動売買ツール」として販売するのではなく、
 - バックテスト・戦略設計・リスク可視化のための学習ツール / 研究ツールとして有料化する方針。
- ・ 法的にグレーな領域（投資一任・助言）には踏み込まない。

SD-TR-061 課金モデル（候補）

- ・ 月額サブスクリプション（例：月¥X,XXX）
- ・ 無料プラン+有料プラン（バックテスト長さ・保存期間・戦略数などで差別化）

本ドキュメントでは詳細金額までは決めず、方向性のみを示す。

8. 今やらないこと（スコープ外宣言）

SD-TR-070 現時点で行わないもの

- ・ 証券会社APIとの直接連携（自動発注）
- ・ 顧客の実口座残高・個別ポジション情報の取得・保存
- ・ KYC / 本人確認等の法的に重いフロー

これらを必要とする場合は、新たに DOMAIN_RULES_EXECUTION.md や LEGAL_COMPLIANCE.md を作成し、別フェーズのプロジェクトとして切り出す。

以上。

本ドキュメントは、EXITON 自動投資システムが「どこまでSaaSとして踏み込むか / どこから先はやらないか」を明確にするための方針書である。