

SportEase の情報の取り扱い

佐藤佑作

2026 年 2 月 11 日

1 個人情報保護方針

本システムにおける個人情報の取り扱いは、以下の通り厳格に管理する。

1.1 Google 認証で収集する個人情報

利用する Google アカウントから取得する個人情報はメールアドレスのみである。

メールアドレスは本システム内でユーザーを一意に識別する ID として使用する。

1.2 データの保存期間と表示

- **保存期間:** 過去のイベント結果（トーナメント表、順位、得点等）は、学校行事の記録として永続的に保存する。
- **表示ポリシー:**
 - 現在進行中のイベントについては、ユーザー名（表示名）を表示する。
 - 終了した過去のイベントについては、プライバシー保護のためユーザー名を非表示とし、クラス単位またはチーム単位の結果のみを表示する。
 - 卒業生の個人データ（メールアドレス等）はホワイトリストから削除されるため、システムへのログインは不可能となる。

1.3 アクセス制御

データベースへのアクセスは以下の通り厳密に制限する。

- **Root (システム管理者):** 全データへのアクセス権限を持つ。
- **Admin (行事委員):** 運営に必要な最小限のデータ（試合結果、出欠状況）のみ参照・更新可能。

- Student (一般学生): 自身のプロフィールおよび公開された競技結果のみ閲覧可能。他者のメールアドレス等は閲覧不可。
- DB 直接アクセス (システム外からのアクセス): 定められたシステム管理者 (1~2名) のみが、SSH 接続を通してのみアクセス可能。

1.4 情報の使用用途

表 1 情報の使用用途

| 情報 | 用途 |
|---------|--|
| メールアドレス | ユーザー認証 (OAuth2)、システムからの重要なお知らせ (Web Push 通知) |
| クラス・学年 | トーナメントのチーム分け、クラス対抗戦の集計 |

ユーザーの表示名に関しては、初回ログイン時に以下の画面で設定し、その後自由に変更可能にできる。



図1 初回ログイン時の表示

2 外部設計

2.1 ユーザーロール

本システムでは以下の 3 つのユーザーロールを定義し、権限管理を行っている。必要に応じてロールを増やすことはできるが、権限管理はこの 3 つのベースロールで行っている。

表 2 ユーザーロール一覧

| ロール名 | 概要 |
|----------------|--|
| Student (一般学生) | 主に情報の閲覧（対戦表、試合結果、クラスの進捗）および通知の受信を行う。 |
| Admin (行事委員) | イベント全体の管理、試合結果の入力・承認、スケジュールの変更、MVP 投票の管理など、運営に必要な全機能にアクセス可能。 |
| Root (システム管理者) | システム設定、ホワイトリスト管理、マスターデータ（スポーツ、クラス等）の管理を行う最高権限。 |

2.2 機能要件

各ロールにおける利用可能な機能の詳細は以下の通りである。

2.2.1 Student (一般学生)

一般的な学生ユーザーに割り当てられるロールであり、以下の機能を利用可能である。

- アカウント管理:

- Google アカウントを用いたログイン・ログアウト
- プロフィール設定（表示名、所属クラスの登録）

- 情報閲覧:

- 開催中のイベント情報の閲覧
- 各競技のトーナメント表・対戦組み合わせの閲覧
- 試合結果および勝敗のリアルタイム確認
- クラス別の総合得点・進捗状況・順位の閲覧

- 昼企画（リレー・綱引き等）の結果閲覧
- チーム管理（自クラスのみ）：
 - 所属クラスのチームメンバー編成機能（メンバーの追加・削除）
 - チームメンバーの確認
- 通知・その他：
 - 試合開始・結果等の Web Push 通知の受信
 - MVP 投票への参加（投票権がある場合）

2.2.2 Admin (行事委員)

イベント運営の実務を担当するロールであり、Student の全機能に加え、以下の管理機能を利用可能である。

- 試合進行管理：
 - 試合開始時間の変更・調整
 - 試合結果（スコア・勝者）の入力・確定
 - 試合ステータスの更新（試合中、終了など）
 - 昼企画（リレー・綱引き等）の予選・決勝結果の入力
- 運営管理：
 - 前日・当日の出欠確認・登録
 - 各競技の参加定員数の調整・変更
 - ユーザーの表示名修正（不適切な名前の修正等）
 - 運営用画像・PDF 資料（要項・ルールブック等）のアップロード
 - MVP 投票状況の確認・集計
 - 行事委員権限の付与・剥奪

2.2.3 Root (システム管理者)

システムの全権限を持つロールであり、マスターデータの管理を含む以下の高度な機能を利用可能である。

- システム・イベント設定：
 - ログイン許可リスト（ホワイトリスト）の管理（追加・削除・一括インポート）
 - 新規イベントの作成およびアクティブラベルの切り替え
 - 雨天モード（スケジュール短縮・変更時）の切り替えおよび設定管理
- マスターデータ管理：

- 競技種目の新規作成・削除
- クラス情報の管理および在籍学生数の登録 (CSV インポート対応)
- 昼企画のグループ分け・テンプレート管理
- 全競技のトーナメント表の一括自動生成・プレビュー
- **全体通知:**
 - 全ユーザーまたは特定のロールに向けた任意のお知らせ通知の配信

2.3 画面構成・遷移

本システムは Web ブラウザ上で動作する SPA (Single Page Application) であり、以下の画面構成を持つ。

- **ログイン画面:** Google アカウントによる認証エントリポイント。
- **ダッシュボード (ホーム):** ログイン後の初期画面。ユーザーロールに応じたメニューを表示。
- **学生向けメニュー (Student):**
 - **マイページ:** プロフィール設定、クラス情報確認
 - **競技・イベント情報:** トーナメント表閲覧、競技詳細、点数一覧
 - **昼企画:** 昼競技の結果確認
 - **その他:** QR コード表示、通知確認、通知申請
- **運営管理者メニュー (Admin):**
 - **試合管理:** 試合結果入力、トーナメント進行管理、昼競技結果入力
 - **参加管理:** クラス・チーム編成管理、QR コード読み取り・参加確認、出席登録
 - **運営設定:** ロール管理、競技詳細設定、MVP 投票管理
- **システム管理者メニュー (Root):**
 - **マスター管理:** イベント作成、競技種目設定、ホワイトリスト管理
 - **全体設定:** 雨天モード設定、トーナメント一括生成、クラス人数設定
 - **コンテンツ管理:** 競技要項アップロード、MVP 確認

2.4 動作環境

- **クライアント端末:** スマートフォン (iOS/Android)、タブレット、PC
- **推奨ブラウザ:** Google Chrome, Safari, Microsoft Edge (各最新版)
- **ネットワーク:** インターネット接続必須 (学内 Wi-Fi またはモバイルネットワーク)

2.5 セキュリティ対策

本システムでは、以下の多層的なセキュリティ対策を実装している。

2.5.1 アプリケーションセキュリティ

- **SQL インジェクション対策:** すべてのデータベース操作において、Go 言語のプレースホルダ機能（Prepared Statements）または ORM を使用し、SQL インジェクションを無効化している。
- **CSRF 対策:** セッションベースの認証（HttpOnly, SameSite=Lax Cookie）および CORS による Origin 検証により、不正なクロスサイトリクエストを防止している。
- **XSS 対策:** SvelteKit のオートエスケープ機能により、ユーザー入力のスクリプト実行を防止している。Markdown 表示箇所については、サニタイズ処理を行っている。
- **API 認可:** 全てのエンドポイントにおいてミドルウェア（Middleware）による認証・認可チェックを行い、権限のないリソースへのアクセスを遮断している。
- **ファイルアップロード:** 画像および PDF ファイルのアップロード時は、拡張子だけでなく MIME タイプの検証を行い、実行可能ファイルの混入を防ぐ。

2.5.2 インフラストラクチャ・通信

- **通信の暗号化:** Let's Encrypt を用いた SSL/TLS 証明書の自動更新により、全通信を HTTPS 化している。
- **ポート制限:** 外部（インターネット）には HTTP(80) と HTTPS(443) のみを公開し、DB ポート (3306) や SSH ポートは外部から遮断している。
- **コンテナ隔離:** Docker によりアプリケーションとホスト環境を分離し、コンテナ侵害時の影響を最小限に抑えている。

2.6 運用管理規定

2.6.1 管理体制

- **運用責任者:** 行事委員会
- **システム管理者:** 佐藤佑作
- **サーバー管理者:** 林航平、佐藤佑作

2.6.2 引継ぎ

- **卒業時の処理:** システム管理者が卒業する際は、アクセス権限（サーバーログイン情報、DB パスワード）を変更し、新管理者へ安全に引き継ぐ。旧管理者のアカウントは即座に無効化する。

3 内部設計

3.1 システムアーキテクチャ

本システムは Docker コンテナを用いたマイクロサービス構成を採用している。

- **Reverse Proxy (Traefik):** エントリーポイントとして HTTPS 通信の終端、Let's Encrypt による SSL 証明書の自動更新、および Frontend/Backend へのルーティングを行う。
- **Frontend (Frontapp):** SvelteKit (Node.js) を用いた SSR/CSR ハイブリッド構成。ユーザーインターフェースを提供。
- **Backend (Backapp):** Go 言語 (Gin Framework) による REST API サーバー。ビジネスロジックおよび WebSocket によるリアルタイム配信を担当。
- **Database (DB):** MySQL 8.0。永続化データを管理。

3.2 技術スタック

表 3 採用技術一覧

| カテゴリ | 技術 |
|---------|------------------------------------|
| フロントエンド | SvelteKit, TailwindCSS, TypeScript |
| バックエンド | Go 1.24, Gin, Gorilla WebSocket |
| データベース | MySQL 8.0 |
| インフラ | Docker, Docker Compose, Traefik v2 |
| 認証 | Google OAuth2 (OpenID Connect) |

3.3 データモデル設計

主なエンティティとその役割は以下の通りである。

- **Users**: ユーザー ID、メールアドレス、ロール、表示名を管理。
- **Events**: 球技大会等のイベント単位。アクティブなイベントの設定が可能。
- **Sports / Classes**: 競技種目および参加クラスのマスターデータ。
- **Teams / Entries**: クラス・競技ごとのチーム編成およびエントリー情報。
- **Tournaments / Matches**: トーナメント構造および個々の試合データ（開始時刻、スコア、勝者）。
- **Notifications**: Web Push 通知の購読情報および通知履歴。

3.4 データベース設計

詳細なデータベース設計は以下の通りである。

3.4.1 Core Tables

基本となるユーザー、イベント、競技、クラス等のテーブル群。

表 4 users (ユーザー)

| Column | Type | Description |
|---------------------|--------------|--------------------|
| id | CHAR(36) | UUID (Primary Key) |
| email | VARCHAR(255) | メールアドレス (Unique) |
| display_name | VARCHAR(255) | 表示名 |
| class_id | INT | 所属クラス ID (FK) |
| is_profile_complete | BOOLEAN | プロフィール登録完了フラグ |
| created_at | TIMESTAMP | 作成日時 |
| updated_at | TIMESTAMP | 更新日時 |

表 5 roles (ロールマスタ)

| Column | Type | Description |
|--------|-------------|-----------------------------|
| id | INT | Primary Key |
| name | VARCHAR(50) | ロール名 (root, admin, student) |

表 6 user_roles (ユーザー・ロール割り当て)

| Column | Type | Description |
|----------|----------|------------------------|
| user_id | CHAR(36) | ユーザー ID (FK) |
| role_id | INT | ロール ID (FK) |
| event_id | INT | イベント ID (FK, Nullable) |

表 7 events (イベント・大会)

| Column | Type | Description |
|------------------------|--------------|----------------------|
| id | INT | Primary Key |
| name | VARCHAR(255) | イベント名 |
| year | INT | 年度 |
| season | ENUM | 時期 (spring, autumn) |
| start_date | DATE | 開始日 |
| end_date | DATE | 終了日 |
| active | BOOLEAN | 現在有効なイベントかどうか |
| competition_guidelines | TEXT | 競技要項 (Markdown/Text) |

表 8 classes (クラス)

| Column | Type | Description |
|---------------|--------------|--------------|
| id | INT | Primary Key |
| event_id | INT | イベント ID (FK) |
| name | VARCHAR(255) | クラス名 |
| student_count | INT | 学生数 |
| attend_count | INT | 出席数 |

表 9 sports (競技マスター)

| Column | Type | Description |
|--------|--------------|-------------|
| id | INT | Primary Key |
| name | VARCHAR(255) | 競技名 |

表 10 event_sports (イベント採用競技)

| Column | Type | Description |
|-------------|------|-------------------------------|
| event_id | INT | イベント ID (FK) |
| sport_id | INT | 競技 ID (FK) |
| description | TEXT | 説明 |
| rules | TEXT | ルール |
| location | ENUM | 場所 (gym1, gym2, ground, etc.) |
| staff_count | INT | 運営スタッフ数 |

表 11 whitelisted_emails (ログイン許可リスト)

| Column | Type | Description |
|----------|--------------|--------------------|
| email | VARCHAR(255) | メールアドレス |
| role | ENUM | 初期ロール |
| event_id | INT | イベント ID (NULLable) |

3.4.2 Competition Tables

トーナメント、試合、チーム、得点等の競技運営に関わるテーブル群。

表 12 teams (チーム)

| Column | Type | Description |
|----------|--------------|--------------|
| id | INT | Primary Key |
| name | VARCHAR(255) | チーム名 |
| class_id | INT | クラス ID (FK) |
| sport_id | INT | 競技 ID (FK) |
| event_id | INT | イベント ID (FK) |
| capacity | INT | チーム定員 |

表 13 `team_members` (チームメンバー)

| Column | Type | Description |
|--------------|----------|--------------|
| team_id | INT | チーム ID (FK) |
| user_id | CHAR(36) | ユーザー ID (FK) |
| is_confirmed | BOOLEAN | 参加確定フラグ |

表 14 `tournaments` (トーナメント)

| Column | Type | Description |
|----------|--------------|--------------|
| id | INT | Primary Key |
| name | VARCHAR(255) | トーナメント名 |
| event_id | INT | イベント ID (FK) |
| sport_id | INT | 競技 ID (FK) |

表 15 matches (試合)

| Column | Type | Description |
|-----------------------|--------------|--|
| id | INT | Primary Key |
| tournament_id | INT | トーナメント ID (FK) |
| round | INT | ラウンド番号 |
| match_number_in_round | INT | ラウンド内試合順 |
| team1_id | INT | チーム 1 ID (FK) |
| team2_id | INT | チーム 2 ID (FK) |
| team1_score | INT | チーム 1 スコア |
| team2_score | INT | チーム 2 スコア |
| winner_team_id | INT | 勝者チーム ID (FK) |
| next_match_id | INT | 次戦 ID (FK) |
| status | VARCHAR(50) | 試合状態 (scheduled, in_progress, completed) |
| start_time | VARCHAR(255) | 開始予定時刻 |
| rainy_mode_start_time | VARCHAR(255) | 雨天時開始時刻 |
| court_number | VARCHAR(255) | コート番号 |
| is_bronze_match | BOOLEAN | 3 位決定戦フラグ |
| loser_bracket_block | VARCHAR(50) | 敗者復活戦ブロック |

表 16 class_scores (クラス得点集計)

| Column | Type | Description |
|----------------------|------|--------------|
| id | INT | Primary Key |
| event_id | INT | イベント ID (FK) |
| class_id | INT | クラス ID (FK) |
| initial_points | INT | 初期点 |
| attendance_points | INT | 出席点 |
| noon_game_points | INT | 昼企画点 |
| mvp_points | INT | MVP 点 |
| total_points_overall | INT | 総合得点 |
| rank_overall | INT | 総合順位 |

表 17 score_logs (得点履歴)

| Column | Type | Description |
|-----------------|-----------|--------------|
| id | INT | Primary Key |
| event_id | INT | イベント ID (FK) |
| class_id | INT | クラス ID (FK) |
| points | INT | 加算/減算ポイント |
| reason | TEXT | 理由 |
| source_match_id | INT | 対象試合 ID (FK) |
| created_at | TIMESTAMP | 作成日時 |

表 18 mvp_votes (MVP 投票)

| Column | Type | Description |
|--------------------|----------|----------------|
| id | INT | Primary Key |
| event_id | INT | イベント ID (FK) |
| voter_user_id | CHAR(36) | 投票者 ID (FK) |
| voted_for_class_id | INT | 被投票クラス ID (FK) |
| reason | TEXT | 理由 |
| points | INT | 投票ポイント |

表 19 check_ins (チェックイン・出席)

| Column | Type | Description |
|---------------|-----------|-----------------------------|
| id | INT | Primary Key |
| user_id | CHAR(36) | ユーザー ID (FK) |
| event_id | INT | イベント ID (FK) |
| purpose | ENUM | 目的 (opening_ceremony, etc.) |
| checked_in_at | TIMESTAMP | チェックイン日時 |

3.4.3 Noon Game Tables

昼企画（リレー、綱引き等）に関わるテーブル群。

表 20 noon_game_sessions (昼企画セッション)

| Column | Type | Description |
|----------------------|--------------|-----------------------------|
| id | INT | Primary Key |
| event_id | INT | イベント ID (FK) |
| name | VARCHAR(255) | 企画名 |
| mode | ENUM | 対戦モード (class, group, mixed) |
| win/loss/draw_points | INT | 勝敗別ポイント設定 |

表 21 noon_game_groups (昼企画グループ)

| Column | Type | Description |
|------------|--------------|-----------------|
| id | INT | Primary Key |
| session_id | INT | セッション ID (FK) |
| name | VARCHAR(255) | グループ名 (例: 1年合同) |

表 22 noon_game_template_runs (テンプレート実行)

| Column | Type | Description |
|--------------|-------------|--------------------------|
| id | INT | Primary Key |
| session_id | INT | セッション ID (FK) |
| template_key | VARCHAR(50) | テンプレート識別子 (year_relay 等) |

3.4.4 Notification Tables

通知機能に関わるテーブル群。

表 23 notifications (通知)

| Column | Type | Description |
|------------|-----------|--------------|
| id | INT | Primary Key |
| title | TEXT | タイトル |
| body | TEXT | 本文 |
| created_by | CHAR(36) | 作成者 (FK) |
| event_id | INT | イベント ID (FK) |
| created_at | TIMESTAMP | 送信日時 |

表 24 notification_recipients (通知受信者)

| Column | Type | Description |
|------------------|----------|---------------------|
| notification_id | INT | 通知 ID (FK) |
| user_id/class_id | CHAR/INT | 受信対象 (ユーザー個別/クラス全体) |

表 25 push_subscriptions (WebPush 購読)

| Column | Type | Description |
|---------------------|--------------|-------------------|
| id | INT | Primary Key |
| user_id | CHAR(36) | ユーザー ID (FK) |
| endpoint | VARCHAR(500) | Push サービスのエンドポイント |
| keys (auth, p256dh) | VARCHAR | 暗号化キー |

3.5 API 設計

RESTful API のエンドポイント一覧。

3.5.1 Authentication / User

表 26 Auth API

| Method | Path | Description |
|--------|---------------------------|---------------------|
| GET | /api/auth/google/login | Google OAuth ログイン開始 |
| GET | /api/auth/google/callback | Google OAuth コールバック |
| GET | /api/auth/dev-login | 開発用ログイン (Dev 環境のみ) |
| GET | /api/auth/user | ログインユーザー情報取得 |
| POST | /api/auth/logout | ログアウト |
| PUT | /api/user/profile | プロフィール更新 |

3.5.2 General / Student

学生権限以上でアクセス可能な API。

表 27 General API

| Method | Path | Description |
|--------|------------------------|-------------------|
| GET | /api/events/active | 現在アクティブなイベント取得 |
| GET | /api/events/:id/sports | イベント採用競技一覧取得 |
| GET | /api/classes | クラス一覧取得 |
| GET | /api/scores/class | クラス得点情報取得 |
| GET | /api/qrcode/teams | 自分の所属チーム QR コード取得 |
| POST | /api/qrcode/generate | QR コード生成 |
| POST | /api/qrcode/verify | QR コード検証 |

表 28 Student API (/api/student)

| Method | Path | Description |
|--------|----------------------------|--------------|
| GET | .../class-progress | クラス進捗状況取得 |
| GET | .../events/:id/tournaments | トーナメント表取得 |
| GET | .../noon-game/session | 昼企画セッション情報取得 |
| GET | .../notification-requests | 通知リクエスト一覧 |
| POST | .../notification-requests | 通知リクエスト作成 |

3.5.3 Admin

行事委員 (Admin) 以上でアクセス可能な API。

表 29 Admin API (/api/admin)

| Method | Path | Description |
|---------|---------------------------|---------------|
| GET | .../events | イベント一覧取得 |
| POST | .../events/:id/sports | 競技をイベントに追加 |
| DELETE | .../events/:id/sports/:id | イベントから競技を削除 |
| GET/PUT | .../events/.../details | 競技詳細設定 |
| PUT | .../result, /status | 試合結果・ステータス更新 |
| POST | .../attendance/register | 出席登録 |
| POST | .../images, /pdfs | 画像・PDF アップロード |
| GET | .../users | ユーザー一覧検索 |
| PUT/DEL | .../users/role | ユーザーロール変更・削除 |

表 30 Class/Team Management API (/api/admin/class-team)

| Method | Path | Description |
|--------|-------------------------|-------------|
| GET | .../managed-class | 担当クラス情報取得 |
| GET | .../classes/:id/members | クラスメンバー取得 |
| POST | .../assign-members | チームメンバー割り当て |
| DELETE | .../remove-member | チームメンバー解除 |

3.5.4 Root

システム管理者 (Root) のみアクセス可能な API。

表 31 Root API (/api/root)

| Method | Path | Description |
|----------|-------------------------------|------------------|
| GET/POST | .../whitelist | ホワイトリスト管理 |
| POST | .../whitelist/csv | ホワイトリスト CSV 一括登録 |
| POST | .../events | イベント新規作成 |
| PUT | .../events/active | アクティブイベント切り替え |
| POST | .../tournaments/generate-all | 全トーナメント一括生成 |
| PUT | .../classes/student-counts | クラス人數設定 |
| POST | .../sports | 競技マスター作成 |
| POST | .../notifications | 全体通知作成 |
| POST | .../notification-requests/... | 通知リクエスト承認・否認 |

3.6 処理フロー例 (トーナメント進行)

試合結果が入力され、トーナメント表が更新されるまでのデータフローは以下の通りである。

1. **結果入力:** Admin ユーザーが管理画面から試合のスコアを入力し、確定ボタンを押下する。
2. **API リクエスト:** フロントエンドから POST /api/admin/match/result が送信される。
3. **バックエンド処理:**
 - matches テーブルの該当レコード（スコア、勝者 ID、ステータス）を更新する。
 - 勝者が決定した場合、トーナメントの構造に基づき、次戦 (next_match_id) の対戦チームとして勝者を自動設定する。
 - WebSocket を通じて、接続中の全クライアントに対して「試合更新イベント」をブロードキャストする。
4. **リアルタイム更新:**
 - 各クライアント (Student/Admin 画面) が WebSocket メッセージを受信する。
 - トーナメント表コンポーネントが再レンダリングされ、リロードなしで最新の勝敗結果が反映される。

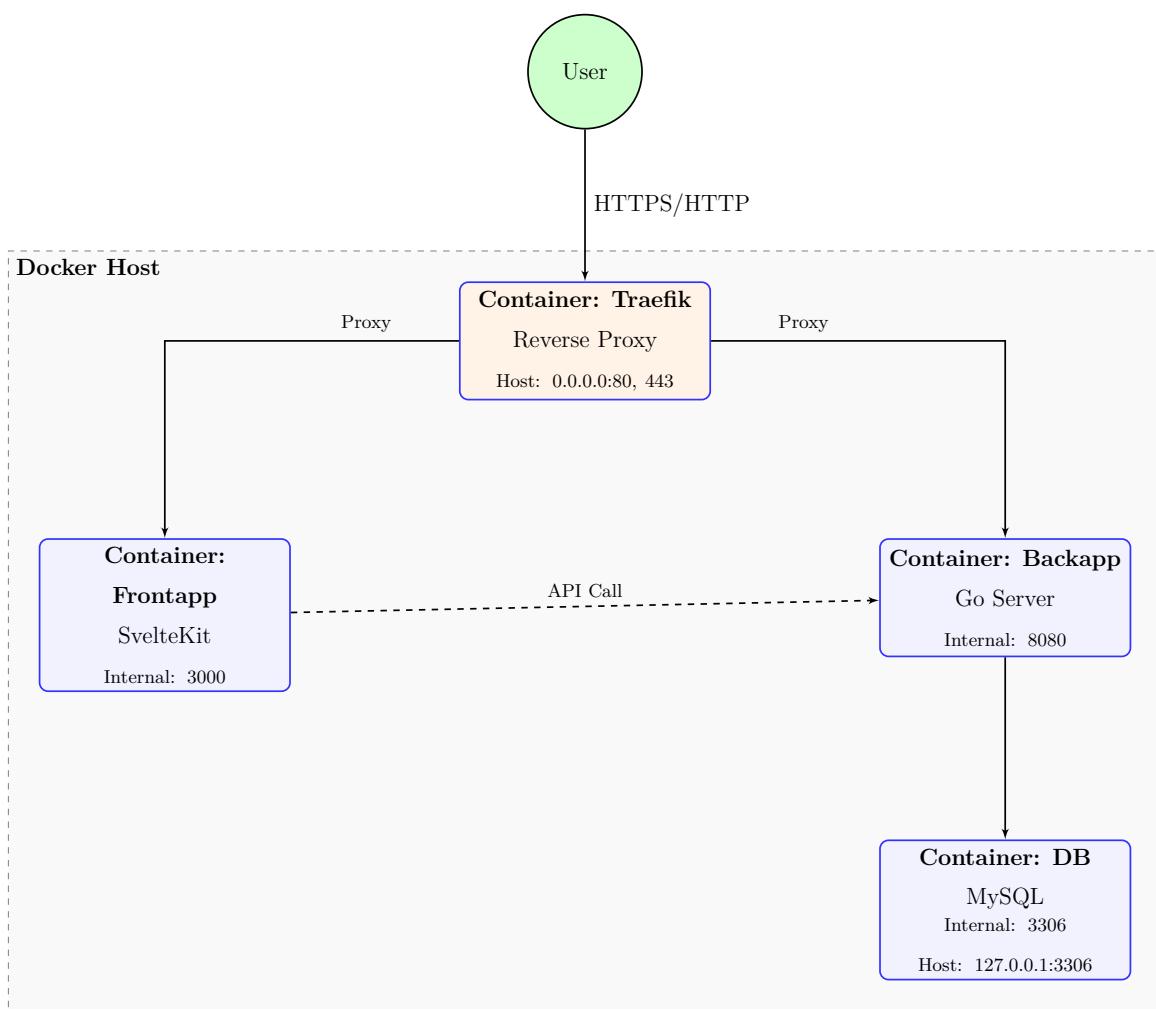


図2 システム構成図（Docker コンテナ構成）