インフラ設計書

Tascal（タスカル） - 社員タスク管理・カレンダーアプリケーション

項目	内容
ドキュメントID	IF-001
バージョン	1.0
作成日	2026-02-26
プロジェクト名	SVN Dashboard App (Tascal)
ステータス	初版

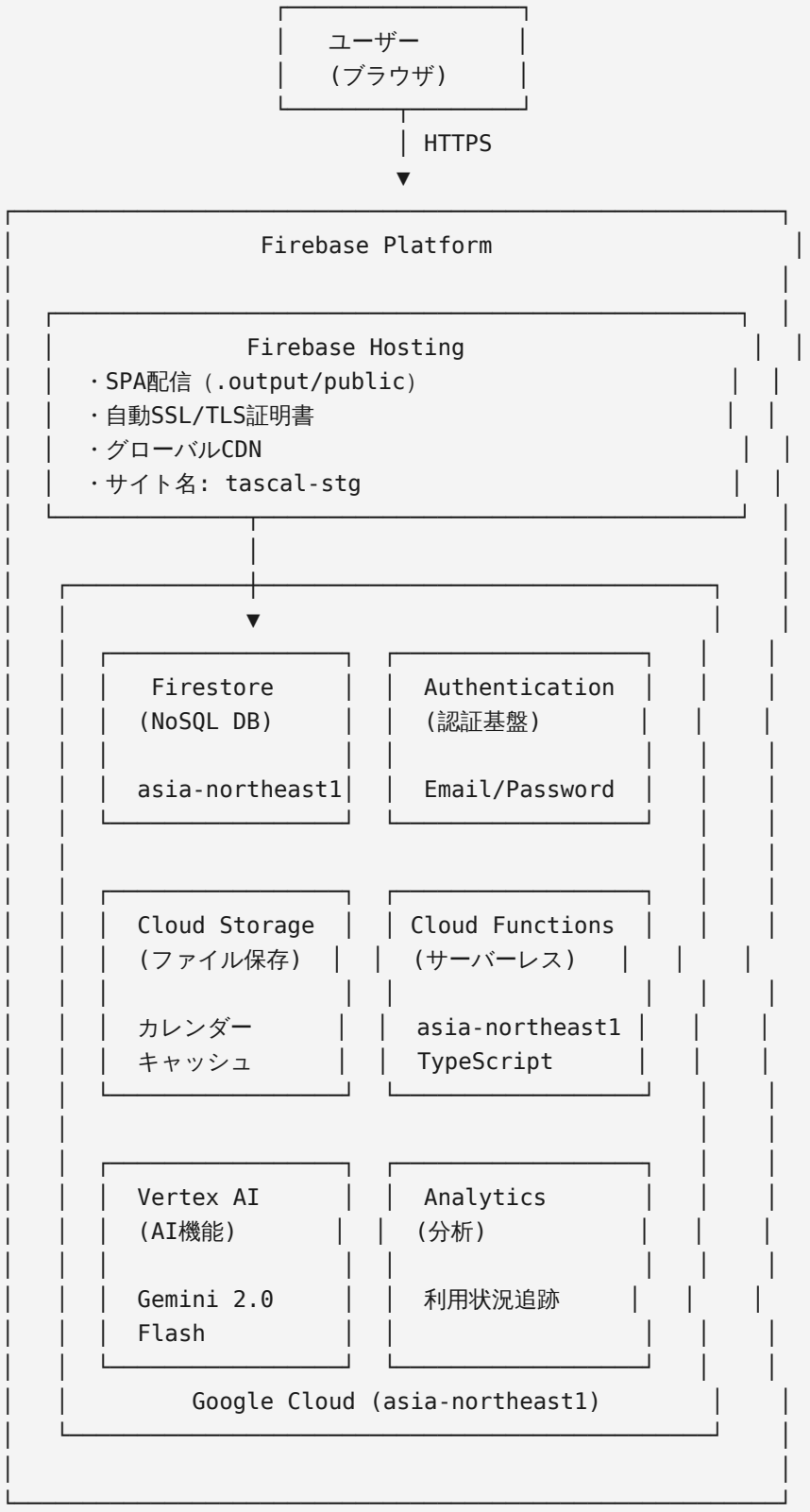
---

1. インフラ概要

1.1 インフラ構成方針

本アプリケーションは **Firebase** のマネージドサービスを全面的に採用し、サーバーレスアーキテクチャで構築する。自前のサーバー管理は不要であり、Google Cloudのインフラ上でスケーラブルに動作する。

1.2 全体構成図


ユーザー
(ブラウザ)
HTTPS
Firebase Platform
Firebase Hosting
・SPA配信（.output/public）
・自動SSL/TLS証明書
・グローバルCDN
・サイト名: tascal-stg
Firestore
(NoSQL DB)
asia-northeast1
Authentication
(認証基盤)
Email/Password
Cloud Storage
(ファイル保存)
カレンダー
キャッシュ
Cloud Functions
(サーバーレス)
asia-northeast1
TypeScript
Vertex AI
(AI機能)
Gemini 2.0
Flash
Analytics
(分析)
利用状況追跡
Google Cloud (asia-northeast1)

2. Firebase プロジェクト構成

2.1 プロジェクト情報

項目	値
プロジェクトID	tascal-app-e3c28
プロジェクト名	Tascal App
デフォルトリージョン	asia-northeast1（東京）
認証ドメイン	tascal-app-e3c28.firebaseapp.com
ストレージバケット	tascal-app-e3c28.firebasestorage.app
Messagingセンダー ID	（設定済み）
App ID	（設定済み）
Measurement ID	（設定済み）

2.2 利用サービス一覧

サービス	用途	プラン	リージョン
Firebase Hosting	SPA静的ファイル配信	Blaze (従量課金)	グローバルCDN
Cloud Firestore	データベース	Blaze	asia-northeast1
Firebase Authentication	ユーザー認証	Blaze	-
Cloud Storage for Firebase	ファイル保存	Blaze	asia-northeast1
Cloud Functions for Firebase	サーバーレスバックエンド	Blaze	asia-northeast1
Firebase Analytics	アクセス分析	無料	-
Vertex AI for Firebase	AI機能（Gemini）	Blaze	asia-northeast1

---

3. Hosting設計

3.1 ホスティング設定

```
{
  "hosting": {
    "site": "tascal-stg",
    "public": ".output/public",
    "ignore": [
      "firebase.json",
      "**/.*",
      "**/node_modules/**"
    ]
  }
}
```

3.2 デプロイ方式

項目	内容
ビルドツール	Nuxt 3 (Vite)
ビルドコマンド	`npx nuxi generate`
出力ディレクトリ	`.output/public`
デプロイコマンド	`firebase deploy --only hosting`
SSR	無効（`ssr: false`）
レンダリング	クライアントサイドレンダリング（CSR）

3.3 配信アーキテクチャ

```
ビルド → 静的ファイル生成 → Firebase Hosting → グローバルCDN → ユーザー

[Nuxt Build]
    │
    ├── index.html (SPA エントリーポイント)
    ├── _nuxt/ (JavaScriptバンドル)
    ├── assets/ (CSS、画像等)
    └── api-mocks/ (モックデータ)
         │
         ▼
[Firebase Hosting CDN]
    │
    ├── 自動SSL/TLS
    ├── HTTP/2
    ├── Brotli圧縮
    └── グローバルエッジキャッシュ
```

3.4 CORS設定

```
[
  {
    "origin": ["特定ドメイン"],
    "method": ["GET"],
    "maxAgeSeconds": 3600
  }
]
```

---

4. Cloud Firestore設計

4.1 データベース構成

項目	値
データベースID	(default)
ロケーション	asia-northeast1
モード	ネイティブモード
セキュリティルール	認証済みユーザーのみ読み書き可能

4.2 パフォーマンス設計

最適化項目	対策	効果
初回クエリ数削減	重複loadData()排除・watch統合	18 → 3〜4クエリ
マスターデータキャッシュ	TTLベースメモリキャッシュ	ロード時間50%削減
ビュー別データ取得	monthly/dailyは自ユーザーのみ	データ転送量90%削減
In-flight重複排除	同一リクエストのPromise共有	不要なクエリ防止
バッチ書き込み	複数ドキュメント一括書き込み	書き込み性能向上
チャンク分割クエリ	IN句30件制限対応	大量ID検索対応

4.3 インデックス構成

7つの複合インデックスを定義（詳細はデータベース設計書参照）。

主要なクエリパターン:

- 日付範囲によるイベント検索
- 参加者IDによるイベント検索
- 施設/設備IDによるイベント検索
- ステータス別ユーザー検索

4.4 セキュリティルール

```
認証チェック: request.auth != null
  ├── 認証済み → 全コレクションの読み書き許可
  └── 未認証 → 全操作拒否
```

---

5. Firebase Authentication設計

5.1 認証方式

認証プロバイダ	状態	備考
メール/パスワード	有効	主要認証方式
Twitter	準備済み	プロバイダ設定あり

5.2 認証フロー

```
[新規登録]
  ├── createUserWithEmailAndPassword()
  ├── メール確認送信（オプション）
  └── Firestoreにユーザープロフィール作成

[ログイン]
  ├── signInWithEmailAndPassword()
  ├── onAuthStateChanged でユーザー状態検知
  ├── IDトークンから admin claim 確認
  └── ユーザープロフィール取得・status確認

[ログアウト]
  ├── signOut()
  ├── ステート初期化（user, userProfile = null）
  └── /signin にリダイレクト
```

5.3 カスタムクレーム

クレーム	型	説明
admin	boolean	管理者権限フラグ

---

6. Cloud Storage設計

6.1 バケット構成

項目	値
バケット名	tascal-app-e3c28.firebasestorage.app
リージョン	asia-northeast1

6.2 ディレクトリ構成

```
tascal-app-e3c28.firebasestorage.app/
├── calendar-cache/            # カレンダーキャッシュ
│   ├── {year}-{week}-cache.json
│   └── ...
├── sample/                    # サンプルファイル
│   └── {id}
└── users/                     # ユーザー別ファイル
    └── {uid}/
        └── ai/
            └── {aid}/
                └── files/
                    └── {fid}
```

6.3 セキュリティ

- Firebase Authenticationと連携したセキュリティルール
- ダウンロードURLはトークン付きで生成

---

7. Cloud Functions設計

7.1 Functions構成

項目	値
ソースディレクトリ	functions/
言語	TypeScript
ランタイム	Node.js
リージョン	asia-northeast1
ビルドコマンド	`npm --prefix "$RESOURCE_DIR" run build`
デプロイコマンド	`firebase deploy --only functions`

7.2 デプロイ設定

```
{
  "functions": [
    {
      "source": "functions",
      "codebase": "default",
      "predeploy": [
        "npm --prefix \"$RESOURCE_DIR\" run build"
      ]
    }
  ]
}
```

---

8. ネットワーク設計

8.1 通信プロトコル

通信	プロトコル	暗号化
ブラウザ → Hosting	HTTPS (TLS 1.2+)	自動SSL証明書
ブラウザ → Firestore	HTTPS (gRPC-Web)	TLS暗号化
ブラウザ → Auth	HTTPS	TLS暗号化
ブラウザ → Storage	HTTPS	TLS暗号化
ブラウザ → Functions	HTTPS	TLS暗号化

8.2 エンドポイント

サービス	エンドポイント
Hosting	https://tascal-stg.web.app
Firestore	https://firestore.googleapis.com
Auth	https://identitytoolkit.googleapis.com
Storage	https://firebasestorage.googleapis.com

---

9. 環境設計

9.1 環境構成

環境	用途	Hosting Site	備考
ステージング（STG）	開発・検証	tascal-stg	現在の主要環境
本番（PROD）	本番運用	（別途設定）	将来構築予定

9.2 環境別設定

設定項目	STG	PROD
Firebase Project	tascal-app-e3c28	別プロジェクト（予定）
SSR	無効	無効
デバッグログ	有効	無効
Firestoreプロファイリング	有効	無効

---

10. ビルド・デプロイ設計

10.1 ビルドパイプライン

```
[ソースコード]
     │
     ▼
[npm install]
  依存パッケージインストール
     │
     ▼
[npx nuxi generate]
  Nuxt 3 静的サイト生成
  ├── TypeScript コンパイル
  ├── Vue SFC コンパイル
  ├── Vite バンドル
  ├── アセット最適化
  └── .output/public/ に出力
     │
     ▼
[firebase deploy]
  Firebase Hosting にデプロイ
  ├── --only hosting  (フロントエンドのみ)
  ├── --only functions (バックエンドのみ)
  └── (全体デプロイ)
```

10.2 デプロイ手順

```
# 1. 依存パッケージインストール
npm install

# 2. 静的サイト生成
npx nuxi generate

# 3. Firebase Hosting にデプロイ
firebase deploy --only hosting

# 4. Cloud Functions にデプロイ（必要な場合）
firebase deploy --only functions

# 5. Firestore ルール/インデックスのデプロイ（必要な場合）
firebase deploy --only firestore
```

---

11. 監視・運用設計

11.1 監視項目

監視対象	監視方法	備考
Hosting アクセス	Firebase Analytics	ページビュー、ユーザー数
Firestore 利用量	Firebase コンソール	読み取り/書き込み数、ストレージ使用量
Authentication	Firebase コンソール	アクティブユーザー数、認証エラー
Cloud Functions	Firebase コンソール	呼び出し回数、エラー率、レイテンシー
Storage 使用量	Firebase コンソール	ストレージ使用量、転送量

11.2 アプリケーション内監視

監視機能	実装箇所	内容
Firestoreクエリプロファイラ	useFirestore.ts	クエリ回数・レイテンシー計測
デバッグサマリー	printFirestoreDebugSummary()	コレクション別クエリ統計出力

11.3 ログ設計

ログ種別	出力先	条件
クエリプロファイル	ブラウザコンソール	開発時
Firebase SDK エラー	ブラウザコンソール	常時
Cloud Functions ログ	Google Cloud Logging	常時
認証エラー	ブラウザコンソール	常時

---

12. セキュリティ設計

12.1 セキュリティ対策一覧

脅威	対策	実装箇所
通信盗聴	HTTPS/TLS強制	Firebase Hosting
不正アクセス	Firebase Authentication必須	middleware/router.global.ts

脅威	対策	実装箇所
XSS	DOMPurify によるHTMLサニタイズ	MarkdownRenderer.vue
不正データアクセス	Firestoreセキュリティルール	firestore.rules
CSRF	Firebase SDKのトークン認証	Firebase SDK
情報漏洩（プライベートイベント）	参加者以外へのマスキング	useCalendar.ts
権限昇格	カスタムクレームによるadmin判定	firebase.client.ts
非アクティブユーザー	ステータスチェック・強制ログアウト	firebase.client.ts

12.2 データ保護

項目	方針
保存時暗号化	Google Cloudのデフォルト暗号化（AES-256）
転送時暗号化	TLS 1.2以上
アクセス制御	Firebase Auth + Firestoreルール
バックアップ	Firestoreの自動バックアップ機能

---

13. 可用性・災害復旧

13.1 可用性

項目	仕様
Hosting SLA	99.95%（Firebase Hosting）
Firestore SLA	99.999%（マルチリージョン）/ 99.99%（リージョン）
Authentication SLA	Firebase プラットフォーム SLA に準拠

13.2 災害復旧

項目	方針
データバックアップ	Firestoreの自動バックアップ
フロントエンド復旧	ソースコードから再ビルド・再デプロイ
認証データ	Firebase Authentication のマネージド管理

項目	方針
RTO（目標復旧時間）	1時間以内（再デプロイ）
RPO（目標復旧時点）	Firestoreの自動バックアップ頻度に依存

---

14. コスト設計

14.1 Firebase 料金体系（Blaze プラン）

サービス	無料枠	超過時の料金目安
Hosting	10 GB/月 転送量	$0.15/GB
Firestore 読み取り	50,000件/日	$0.06/100,000件
Firestore 書き込み	20,000件/日	$0.18/100,000件
Firestore ストレージ	1 GB	$0.18/GB
Authentication	50,000 MAU	超過分は有料
Cloud Storage	5 GB	$0.026/GB
Cloud Functions	200万回/月	$0.40/100万回

14.2 想定月額コスト

小規模利用（社員100名以下）の場合、Firebase無料枠内で運用可能な見込み。 利用規模の拡大に応じて従量課金が発生する。