MUSUBI PLATFORM

Musubi プラットフォーム 方式設計書（基本設計）

システム方式・技術アーキテクチャ — IPA 共通フレーム/基本設計の構成に準拠（実装の実態に基づく）

版数 **v0.3（ドラフト）**
作成日 **2026-06-29**
対象 **管理/ポータル/公開サイト/連携**
関連 **要件定義書 v0.3**

WHAT=要件定義書／HOW=本書。多観点レビュー(IPA準拠/図表/技術正確性)を反映し図・表・状態を整備。

CONTENTS

目次（IPA 基本設計の構成）

1. DOCUMENT

文書情報・用語・凡例

- 目的：Musubi 事業システムの技術方式を体系化し、開発・保守・引き継ぎ・合意形成の基準とする。
- 位置づけ：要件定義書(WHAT)の実現方式(HOW)。要件 v0.3 の確定/未決と対応（14章トレーサビリティ）。
- 対象：管理アプリ・顧客ポータル・公開サイト群・定期処理・外部連携。

用語：ストア=店舗(stores) ／ owner=店主 ／ staff=運営 ／ content=サイト内容(jsonb) ／ デプロイ=公開生成 ／ 在庫=空き枠

状態凡例

確定 ：実装・運用済み

仕掛 ：一部実装／検討中

検討漏れ/未確定 ：未実装・要対応

図の凡例：実線=同期・破線=非同期/Cron・赤=無認証経路

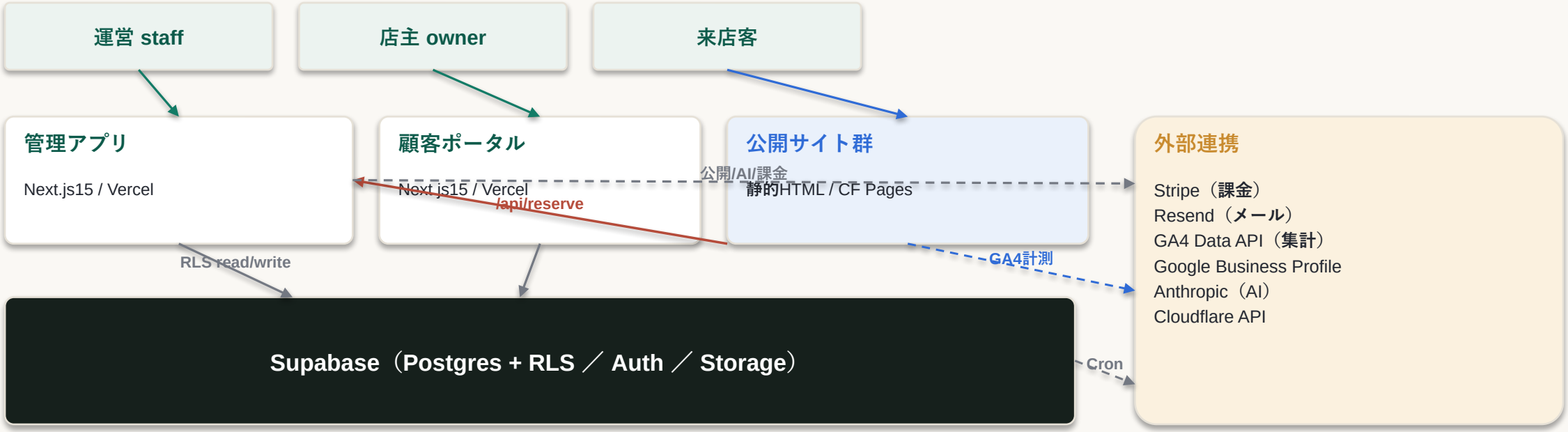
2. PREMISE

前提・制約条件

区分	前提・制約	状態
基盤	マネージドSaaS(Vercel/Cloudflare/Supabase/GCP)前提。自前サーバ・OSは持たない	確定
ドメイン	musubiweb.com の DNS は Cloudflare 管理。公開サブドメインはこれに依存	確定
公開サイト	静的HTML（サーバ処理なし）。動的処理は予約API・Cron・管理/ポータルに集約	確定
外部依存	AI/メール/課金/集計/口コミは各SaaSのキー・利用承認に依存	確定
口コミ運用	Google Business Profile API の利用承認が前提（取得・投稿は承認待ち）	未確定
法令	個人情報・特商法・電通法(外部送信規律)等の整備が前提（要件 ADDENDUM）	未確定

3.1 ARCHITECTURE

システム全体構成（論理構成図）



要点：UIとデータは Supabase に集約。公開サイトのみ静的分離（管理系障害が公開を止めない）。赤=無認証の公開予約API、破線=非同期/Cron。

3.2 DEPLOYMENT

物理配置図（デプロイ・信頼境界）

信頼境界：管理/ポータルは認証必須(緑)。公開サイト・公開予約API・Cron・Stripe Webhookは無認証経路(赤/青)で、Bearer照合・CORS+store検証・署名検証で個別防御。

3.3 NETWORK / DNS

ネットワーク・ドメイン構成

レコード/経路	内容	状態
NS	musubiweb.com → emma/miguel.ns.cloudflare.com（Squarespaceから移管済）	確定
apex / www	主サイト（A×4 / CNAME ext-sq.squarespace.com）	確定
MX / TXT	Google Workspace（MX smtp.google.com・SPF・DKIM・DMARC=要明文化）	仕掛
<店舗>	CNAME → <project>.pages.dev（proxied）＋ Pages 独自ドメイン割当・SSL自動	確定
send	send.musubiweb.com（Resend・SPF/DKIM設定済・DMARC方針は課題）	仕掛
CORS	公開予約API は Access-Control-Allow-Origin=* ＋ store(site_project)検証	確定

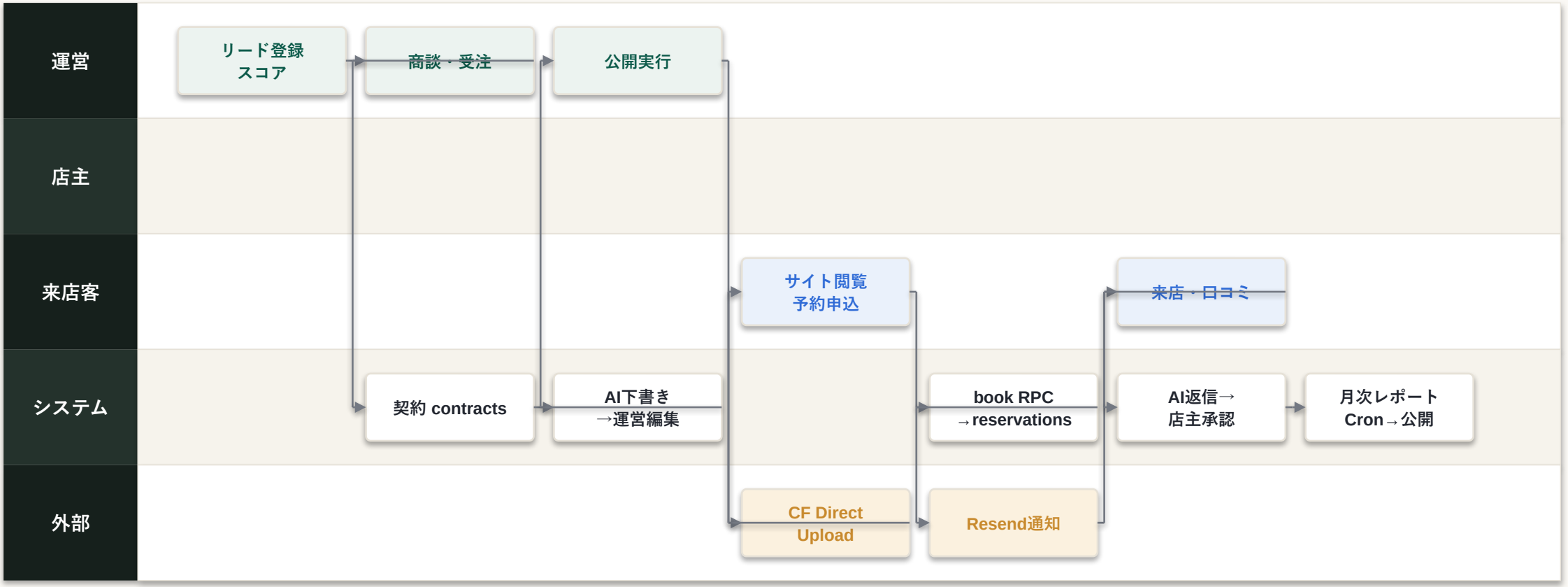
4. STACK

ソフトウェア構成（技術スタック）

区分	採用技術・方式
フロント/サーバ	Next.js 15(App Router)・React 19・TypeScript。Server Actions＋Route Handlers。@supabase/ssr
スタイル	共通 globals.css の独自クラス／CSS Module（Tailwindは出力未配線のため原則不使用）
データ	Supabase Postgres(Tokyo)・RLS・Storage(site-assets)・Auth(staff/owner)
レンダリング	自作テンプレエンジン render.ts（{{}}/{{#each}}/{{#if}}）。プレビューと公開生成で共用
AI	Anthropic Messages API（tool-use）。制作=sonnet-4-6／口コミ=haiku（env切替）
集計/連携	@google-analytics/data（GA4）・stripe・Resend(fetch)・Cloudflare API・GBP(予定)
実行基盤	Vercel(管理/ポータル/Cron)・Cloudflare(Pages/DNS)・GCP(WIF)・Node実行環境

5. BUSINESS FLOW

業務処理フロー（スイムレーン）



分岐（◇）：空き無し→電話導線 ／ 低評価→人間レビュー必須 ／ 支払失敗→Resend案内（要件 ADDENDUM）。承認ゲート＝口コミ投稿・公開前確認。

6. FUNCTION

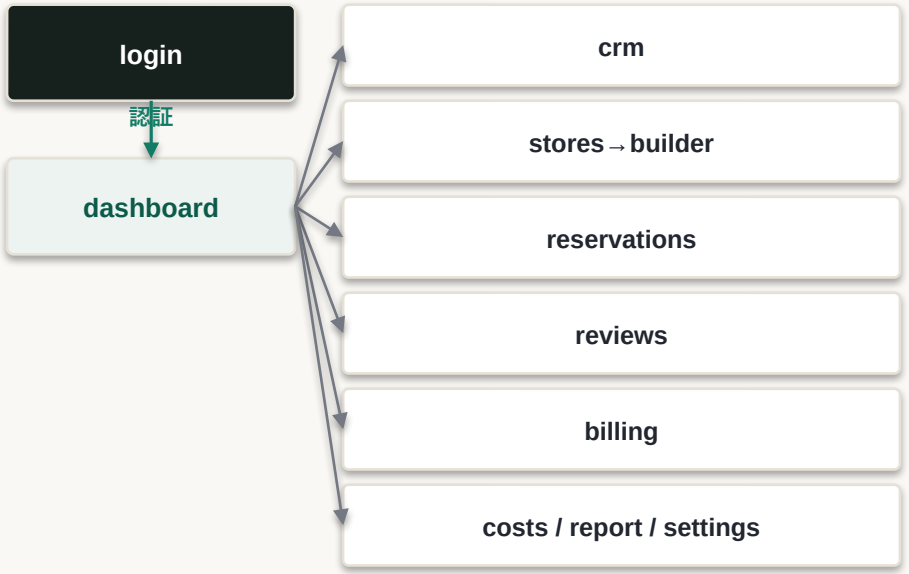
機能構成・機能一覧（機能ID）

ID	機能 / サブシステム	状態
FA	管理：dashboard / crm(営業) / stores(顧客) / settings	確定
FB	制作：builder（テンプレ・フォーム/JSON・画像アップロード・AI下書き・プレビュー・公開/削除）	確定
FC	予約：reservations 管理 ／ 公開サイトのネット予約（在庫型）	確定
FD	口コミ：reviews（AI下書き・監視）／ ポータル承認 ／ GBP取得・投稿	仕掛
FE	レポート：GA4集計→reports→自動公開・通知（アクセス数）	確定
FF	課金：billing（Stripe 3プラン・Webhook同期・初期費）	仕掛
FG	運用：costs（コスト監視）・監視・バックアップ	確定
FH	認証・認可：Supabase Auth＋RLS（staff/owner）	確定

7. SCREEN

画面方式（画面遷移図）

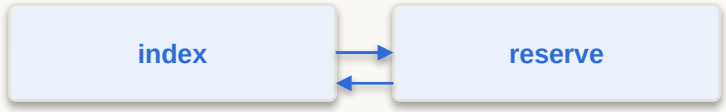
管理アプリ（staff）



顧客ポータル（owner）

自店スコープ(RLS)

公開サイト（無認証）

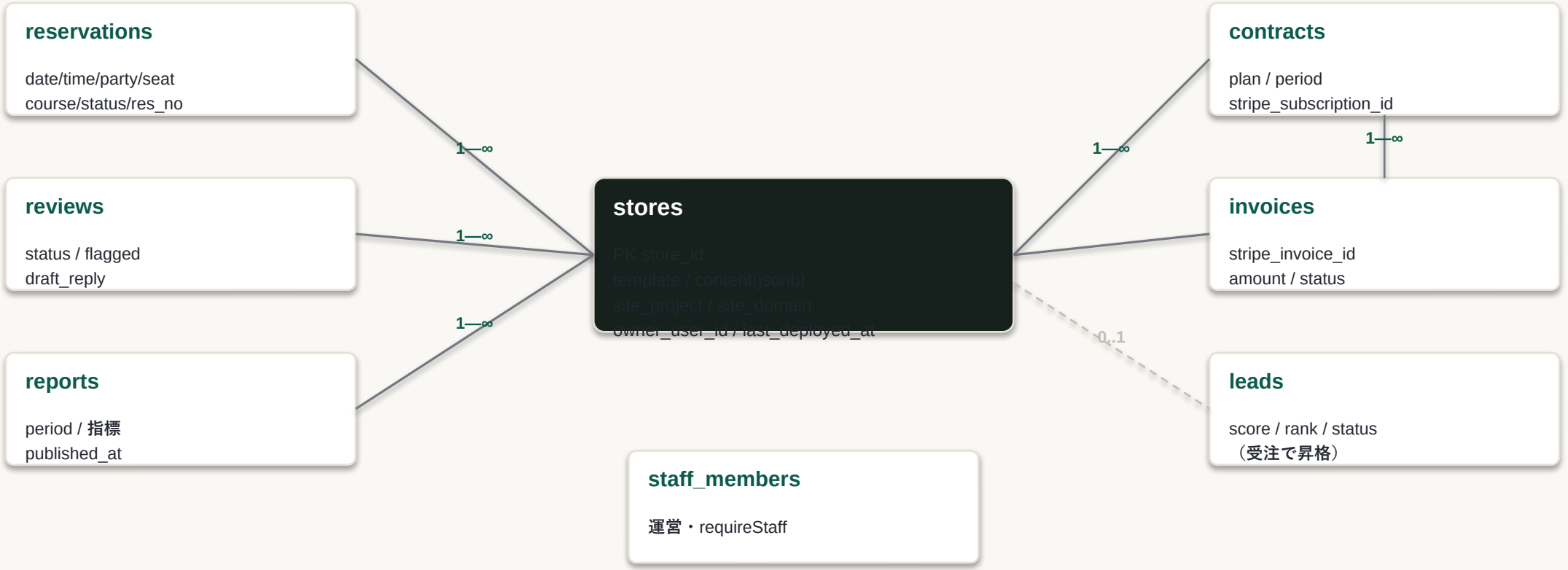


reserve：カレンダー→スロット→入力→確定

★GA4イベント(電話/予約)は未実装＝検討漏れ
予約未接続時はローカルfallback

8.1 DATA / ER

データ方式：ER概要図（stores 中心）



注：content(jsonb)＝サイト内容の集約点。在庫=定員－reservations確定人数（算出値・非保存）。owner_user_id/staff_members=Auth.usersと対応（RLS基点）。

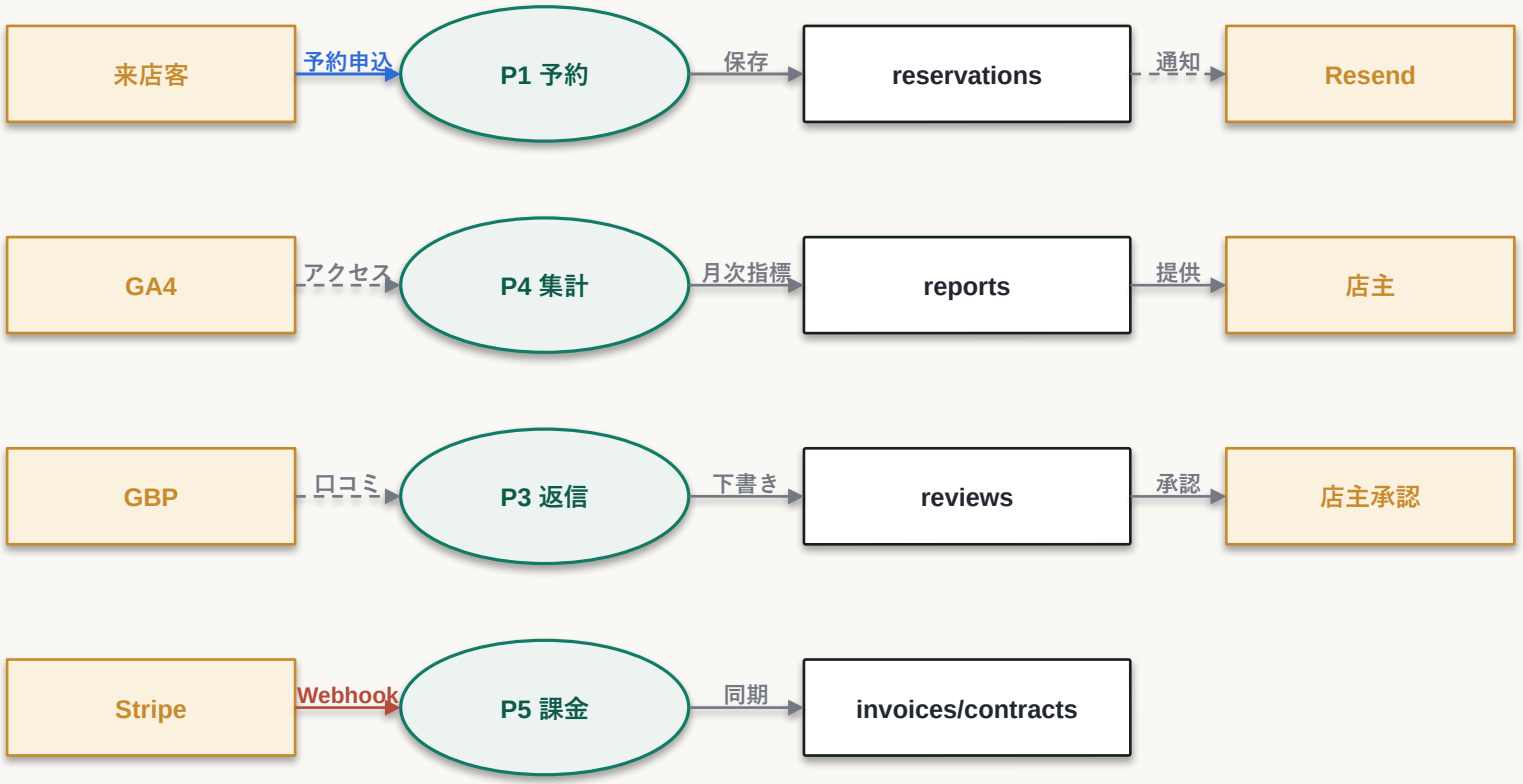
8.2 DATA / TABLES

データ方式：テーブル定義（主なもの）

テーブル	主な列 / 値域	状態
leads	氏名/連絡先・score・rank・status(未着手/アポ/商談/受注/除外)。staff限定RLS	確定
stores	template・content(jsonb)・site_project・site_domain・owner_user_id・last_deployed_at	確定
reservations	date/time/party/seat/course_id/course_name・total・status(confirmed/cancelled)・res_no	確定
reviews	rating・comment・draft_reply・status(pending/drafted/approved/posted/skipped)・flagged	確定
reports	period・access/phone/reservation 指標・comment・published_at	仕掛
contracts / invoices	plan・stripe_subscription_id／stripe_invoice_id・amount・支払status	仕掛
staff_members / cost_alerts	運営アカウント／コスト警告(period,service,level でunique・冪等)	確定

注：コアテーブル(stores/leads/reviews/reports/contracts/invoices/staff_members/cost_alerts)はRLSをマイグレーション(0002/0003/0007)で定義済み。reservationsはRLS有効だが明示ポリシー無し（anon拒否・APIはservice role運用）。残課題＝reservationsのポリシー追加とテナント分離の自動テスト（14章）。

データフロー図（レベル1）



注記

・PII境界：tel/email は P3(AI)へ流さない

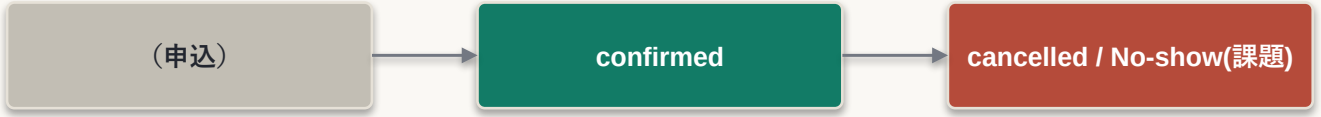
・アクセス数=GA4 / 予約数=reservations から集計（GA4イベント未実装のため）

・公開系：運営/AI→stores.content→Cloudflare→公開サイト

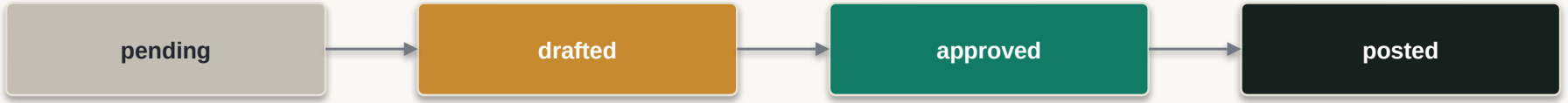
8.4 DATA / STATE

コード設計（状態遷移）

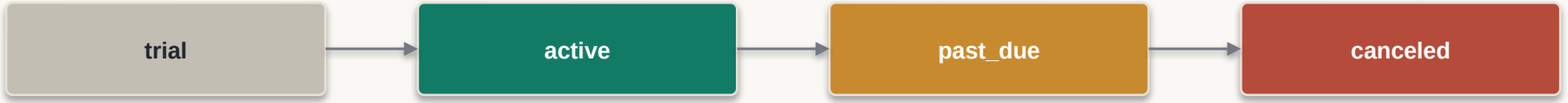
予約 reservations.status



口コミ reviews.status



契約/請求（Stripe同期）



注：予約の No-show・変更/キャンセル、契約状態に連動した公開/予約/口コミの自動ON/OFF（状態機械）は未実装＝検討漏れ（要件 ADDENDUM O1/O4）。

9.1 PROCESSING

処理方式（オンライン/バッチ/認証境界）

同期：Server Actions

管理/ポータルのフォーム
保存・公開・取消・承認
RLS or 認可ラッパで実行

同期：Route Handlers

/builder/upload・ai-fill
/api/reserve/availability・book
CORS・認証は経路ごと

非同期：Cron（Vercel）

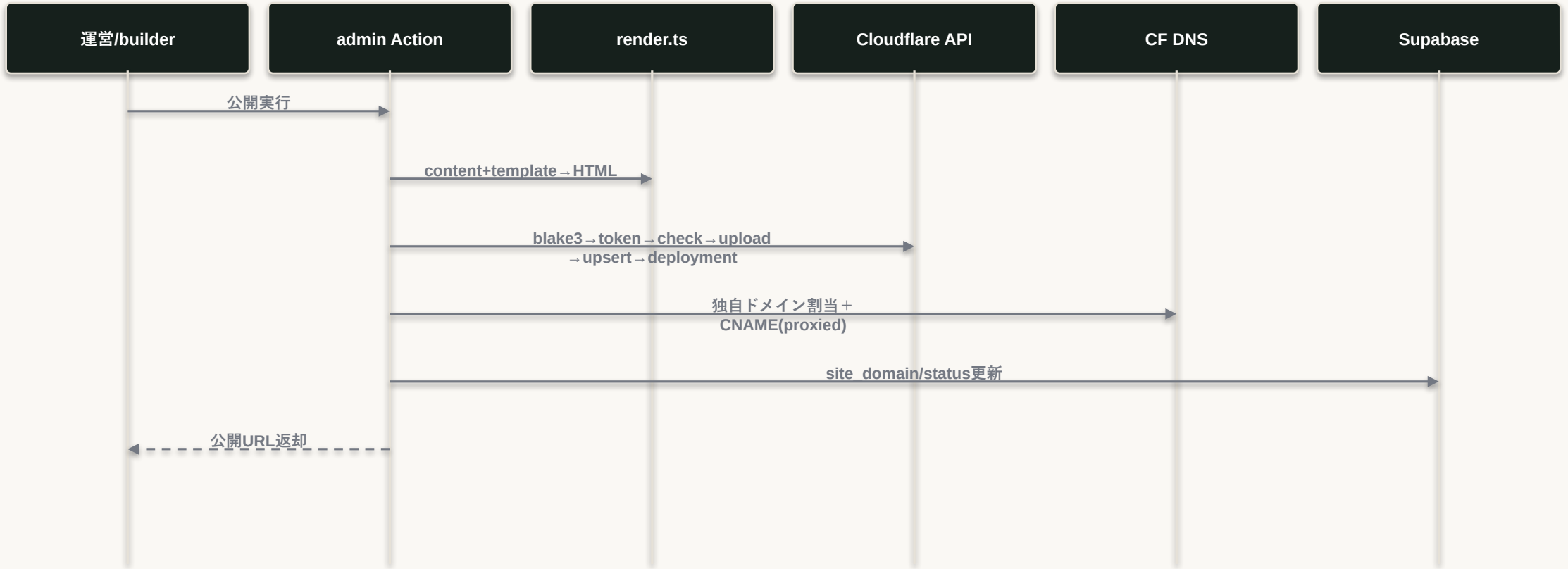
ingest-ga4 月次(0 3 1 * *)
cost-alert 日次
generate-drafts（未登録）

認証境界（middleware）

- 全リクエストで getUser()。未認証は /login へ。例外：/api/cron/*（CRON_SECRET の Bearer 照合）と /api/reserve/*（公開・CORS）。
- 検討漏れ：/api/stripe/webhook は middleware スキップ対象外＝未認証POSTが /login へリダイレクトされる恐れ。スキップ追加が要修正。

9.3 SEQUENCE (1)

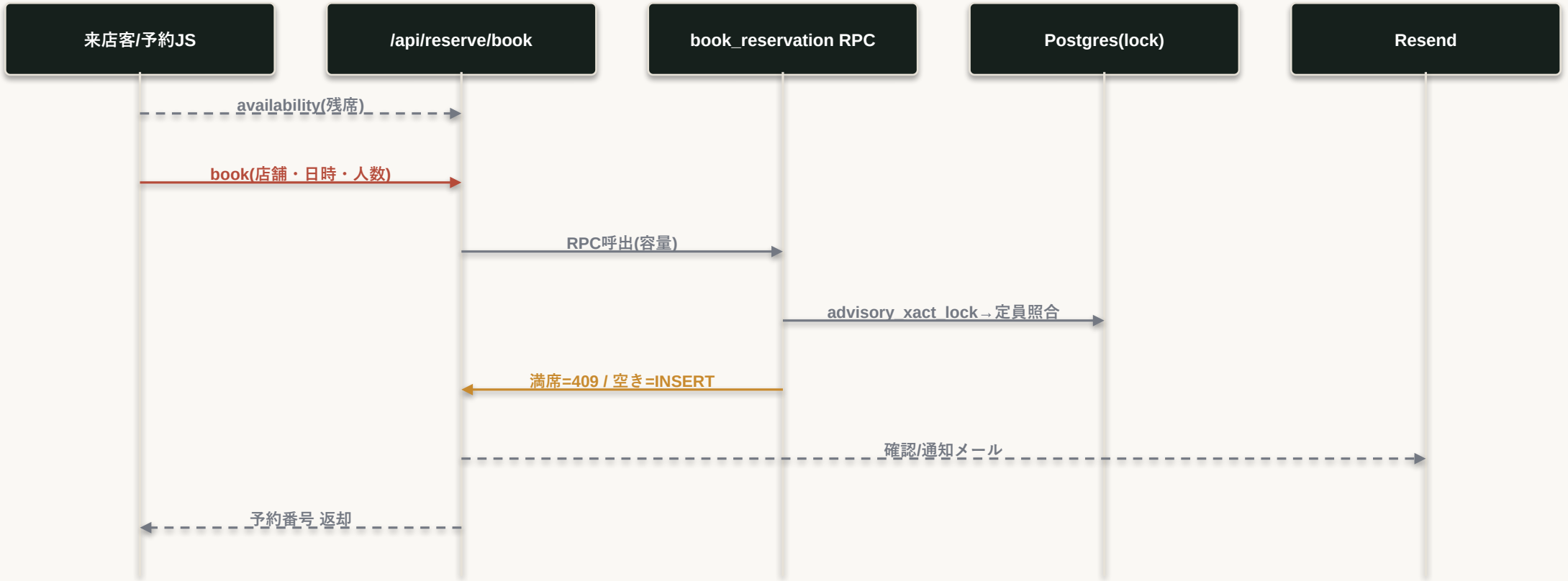
処理シーケンス：サイト公開デプロイ



差分アップロード(ハッシュ照合)で冪等。削除時は Pages＋CNAME を撤去。

9.3 SEQUENCE (2)

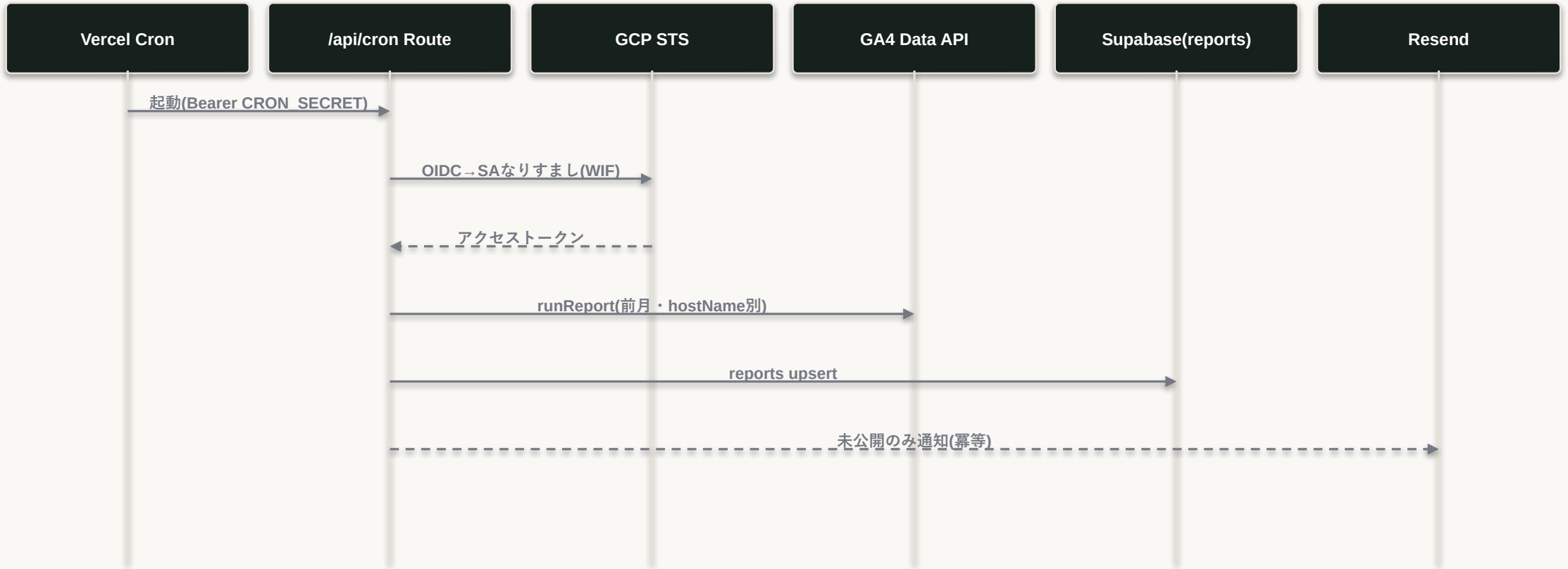
処理シーケンス：ネット予約確定



無認証・CORS、store は site_project で検証(service role)。未注入プレビューはローカルfallback。状態機械・冪等キーは課題。

9.3 SEQUENCE (3)

処理シーケンス：月次レポート集計



鍵レスWIF。予約数はGA4イベント未実装のため reservations から集計すべき＝要再設計（検討漏れ）。

排他・トランザクション方式

- 予約確定：book_reservation RPC 内で pg_advisory_xact_lock(店舗+日時のキー) を取得し、定員照合→INSERT を同一トランザクションで原子的に実行（オーバーブッキング防止）。
- レポート：reports は upsert（period＋店舗で一意）。未公開のみ published_at セットで二重通知を防止（冪等）。
- 課金：invoices は stripe_invoice_id で冪等 upsert。subscription系イベントの重複・順序は未対策＝仕掛。
- 検討漏れ：予約の冪等キー(二重送信)・同一枠の確定上限のDB制約・状態機械は未整備（要件 ADDENDUM D6/O4）。

9.5 AI METHOD

AI方式（Claude）

用途	方式	状態
口コミ返信下書き	draftReply(haiku)。口コミを<review>で隔離(インジェクション対策)。低評価flagged・失敗時テンプレ	確定
サイト下書き生成	aiFill(sonnet-4-6)。参考URL取得→HTML除去＋画像抽出→tool強制でcontent構造化。画像は候補URLのみ	確定
安全方式	SSRF(DNS解決で私的IP拒否＋リダイレクト不追従)・出力エスケープ・PII(tel/email)非送信	確定
人間関与	AI生成物は公開/投稿前に人手承認を前提（口コミは店主承認）	確定

9.5 REPORT METHOD

レポート方式（GA4）※要再設計

- 集計：GA4 Data API（@google-analytics/data）。1プロパティ集約・hostName で店舗識別。月次Cron(0 3 1 * *)→reports upsert→未公開のみ自動公開＋通知。
- 認証：鍵レス WIF（VERCEL_OIDC_TOKEN→GCP STS→サービスアカウント）。本番で集計の実通し確認済み。
- アクセス数：GA4 の hostName 別セッションから取得（確定）。
- 予約数：現状 ingest-ga4 は GA4 の eventCount を見るが、公開サイトTSテンプレに 予約送信/電話タップ イベントが未実装＝計測されない（検討漏れ）。予約数は reservations テーブルから集計する方式へ是正が必要。
- GBP由来の指標（口コミ件数等）は API 承認まで手入力で代替（仕掛）。

9.5 AUTH

認証・認可方式

層	方式	状態
プラットフォーム	Vercel Deployment Protection（管理/ポータルの前段ゲート）	確定
認証	Supabase Auth。middleware が getUser() で検証、未認証は /login	確定
認可(運営)	staff_members 照合の requireStaff()。課金等を staff 限定	確定
認可(店主)	顧客ポータルで owner_user_id と store の一致を検証（口コミ承認の IDOR ガード）。管理側の取消は staff/RLS ベース	確定
分離	RLSはコアテーブルでマイグレーション定義済み（reservationsはポリシー追加・分離テストが課題）	仕掛
公開/Webhook	予約API=無認証＋CORS＋store検証。Stripe Webhook=署名検証だが middleware スキップ漏れ＝要修正	検討漏れ

10. INTERFACE

外部インタフェース方式（IF定義）

連携先	方向 / 方式 / 認証 / 冪等・エラー	状態
Cloudflare API	admin→CF。Pages Direct Upload・DNS。APIトークン。失敗時はリトライ/通知が課題	確定
Supabase	双方向。@supabase/ssr(JWT) / service role。RLS。PostgREST・RPC	確定
Anthropic	admin→API。Messages/tool-use。APIキー。失敗時フォールバック	確定
GA4 Data API	Cron→GA4。runReport。WIF(鍵レス)。クォータ・TZは課題	仕掛
Stripe	Stripe→admin(Webhook 署名検証)・admin→Stripe(API)。invoice冪等／subscription順序は課題	仕掛
Resend	admin/API→送信。APIキー。バウンス/DMARC・到達監視は課題	仕掛
Google Business Profile	双方向（取得/投稿）。API 利用承認待ち＝未接続	未確定

11. NON-FUNCTIONAL

非機能方式（性能・可用・運用・保守）

観点	方式 / 現状	状態
性能	公開サイトは静的CDN配信。プレビューはクライアント生成。数値目標は未定	仕掛
可用性	マネージドSaaSに依拠。公開とデータ層を分離。SLA/RTO/RPOは未定	仕掛
拡張性	サーバレス水平拡張。テンプレ/モジュール追加。content(jsonb)で店舗差分吸収	確定
運用・監視	CI/CD(GitHub→Vercel)。UptimeRobot3点・コストアラート日次・ランブック	確定
保守	構築ログ・ロードマップを随時更新。監査ログ・相関IDは未整備	仕掛
バックアップ	論理BU(全テーブルJSON)復元手順。PITR/暗号化保管は課題	仕掛

11. SECURITY

セキュリティ方式

観点	方式	状態
出力(XSS)	メール/HTML出力でユーザ入力をエスケープ	確定
アップロード	画像は拡張子allowlistでContent-Type決定（client MIME不採用・SVG除外）	確定
SSRF	AI取得URLをDNS解決して私的/予約IP拒否＋リダイレクト不追従	確定
IDOR	予約取消・口コミ承認で店舗スコープ＋本人確認	確定
公開API防御	レート制限/CAPTCHA/冪等/メール実在検証は未実装＝検討漏れ	検討漏れ
RLS/監査	コアテーブルのRLSは定義済み。reservationsポリシー・分離テスト・監査ログが未整備	仕掛

12. MIGRATION

移行方式

- DNS移行：musubiweb.com を Squarespace→Cloudflare へNS切替（既存A/www/MX/SPF/DKIM/DMARCを複製してから切替・メール無断絶を死守）＝実施済み。
- DBマイグレーション：Supabase Management API（個人アクセストークン）でDDLを本番適用。適用済み版を記録。
- サイト移行：既存デモ「さくら亭」をテンプレ化済。既存サイトからの取り込み範囲・並行稼働・301切替は未定（課題）。
- 解約時：公開停止/サブドメイン回収・データ(HTML/画像/予約/口コミ)のエクスポート・削除手順は未整備＝検討漏れ。
- 方式上の課題：本番反映前のステージング・公開前承認・ロールバック(Instant Rollback)手順の整備。

13. DEV

開発標準・環境

- リポジトリ：musubi-admin / musubi-portal（アプリ・private・master）、musubi-demos（ハブ・public・main）、local-web-sales（ドキュメント）。公開サイトは Cloudflare Pages プロジェクト。
- デプロイ：master push→Vercel 本番自動デプロイ。アプリ反映はユーザ指示時に push、ドキュメントは随時 push。
- 品質：実装時に tsc(型)・build を通す。UIは実描画(ヘッドレス)で確認。数値は捏造せず未接続は「—」。
- 環境：本番(Vercel)中心。dev/stg は未整備。秘密情報は Vercel 環境変数で管理（ローテーション運用は課題）。
- 検討漏れ：自動テスト(RLS分離・予約整合・回帰)・ステージング・命名/コーディング規約の標準化。

14.1 ISSUES

課題管理表（状態別）

ID	課題	状態
C1	公開予約APIの多層防御（レート制限/CAPTCHA/冪等/メール検証）	検討漏れ
C2	reservationsのRLSポリシー追加＋テナント分離の自動テスト（コアRLSは定義済み）	仕掛
C3	/api/stripe/webhook の認証スキップ漏れ（middleware）	検討漏れ
C4	予約数のレポート集計の是正（GA4イベント未実装→reservationsから集計）	検討漏れ
C5	課金⇔公開/予約/口コミの状態機械（自動連動）・subscription冪等	仕掛
C6	SLO/RTO/RPO・PITR・TZ統一・Cron/Stripe監視と突合	仕掛
C7	在庫モデル拡張・content版管理・監査ログ・解約オフボーディング	仕掛

14.2 TRACEABILITY

要件トレーサビリティ（要件v0.3 ↔ 方式）

要件	対応する方式（本書）	状態
F4-6 制作/公開/予約	5章業務フロー・8章サイト生成・9章予約シーケンス	確定
F8-10 口コミ	9.5 AI方式・状態遷移（GBP取得/投稿は未接続）	仕掛
F11 レポート	9.5 レポート方式（予約数の集計是正が必要）	仕掛
F12 課金	DFD P5・外部IF Stripe（webhookスキップ・冪等が課題）	仕掛
非機能(IPA6)	11章 非機能方式（目標値未定）	仕掛
ADDENDUM 検討漏れ	14.1 課題管理表 C1〜C7 に対応	検討漏れ

SUMMARY

方式設計の要点

- UIとデータは Supabase に集約、公開サイトのみ静的分離（障害波及を抑制）。
- 公開＝テンプレ→content→TSレンダリング→Cloudflare Pages Direct Upload で内製・1操作化。
- 予約は在庫型(RPC＋advisory lock)、AI/レポートはサーバ・Cronに分離、認証は middleware で集中。
- 本書は IPA 基本設計の構成に準拠し、図(構成/配置/業務/ER/DFD/シーケンス/遷移)と表で実態を正規化。
- 確定方式の上に、予約API防御・RLSコード化・webhookスキップ・予約数集計是正・状態機械を順次対応。

改訂履歴：v0.1（初版）→ v0.2（2026-06-29 IPA準拠の章立て・図表整備・多観点レビュー反映）。関連：要件定義書 v0.3。