TravelQ Hotel Admin — Kiến trúc microservice

Đề xuất self-host Holitech Linux · thiết kế only · 07/11/2026 (Asia/Saigon)

Trạng thái: Bản thiết kế đề xuất – chưa triển khai / chưa nghiệm thu

Mã tài liệu	TravelQ-HotelAdmin-Kien-truc-Microservice-20261107
Đối tượng	anh Long · đội kỹ thuật Holitech
Phạm vi	Thiết kế kiến trúc · không mã ứng dụng · không deploy · không credential
Hạ tầng mục tiêu	Self-host Linux Holitech · **không** Cloudflare Pages/Functions/D1/TURN mint
Nguồn đối chiếu	Hợp đồng Spa/KT OFFLINE; audit Linux 07/11; manifest nguồn TEST t368 (250/250, aggregate `c3ca7ab08ca8a17f`) — đường dẫn tương đối trong kho tài liệu, không nhúng secret

> Mọi mốc thời gian/công sức là **ước lượng lập kế hoạch**, không phải cam kết giao hàng hay chi phí. Không tuyên bố chứng nhận danh tính TQID/provider; gateway TQID là ranh giới ngoài kèm kiểm thử tích hợp hiện có. Tài liệu **không** nêu số liệu hồ sơ TQID hay khẳng định dữ liệu thật/giả của provider.

Giả định nguồn của bản này: đọc hợp đồng Spa/KT + đề xuất transport OFFLINE; audit khả thi Linux + provenance hash; đối chiếu nghiệp vụ từ cây nguồn TEST đã pin. Không đọc/chép secret. Bản này dùng sự thật đã có lúc soạn.

1. Mục tiêu điều hành và phạm vi

1.1 Mục tiêu

- Tách miền nghiệp vụ thành dịch vụ có **chủ sở hữu dữ liệu riêng** (database-per-service), giảm điểm nghẽn monolith hiện tại (một Worker + D1 dùng chung).
- Giữ **hợp đồng nghiệp vụ đã biết** (HV/TA, một tài khoản nhận tiền KS, bộ phận, hiển thị TQID, xuất pháp lý kiểm soát) khi đổi kiến trúc.
- Chạy trên máy chủ Linux Holitech tự quản; TLS Nginx/Caddy; bí mật qua Docker/K8s secrets — không phụ thuộc Cloudflare.

1.2 Trong / ngoài phạm vi

Trong phạm vi

Catalog ~14–16 miền; ma trận sở hữu dữ liệu; luồng bắt buộc; bảo mật; vận hành; lộ trình & rủi ro. Phân biệt UI bộ phận với backend miền — UI riêng không đồng nghĩa một microservice mỗi tab.

Ngoài phạm vi

Viết mã, migration chạy thật, deploy, provisioning, đọc secret, gửi tin ngoài. Chứng nhận cổng thanh toán / khuôn mặt / encoder / attestation (chưa nghiệm thu). App TravelQ ID trung tâm thuộc biên ngoài.

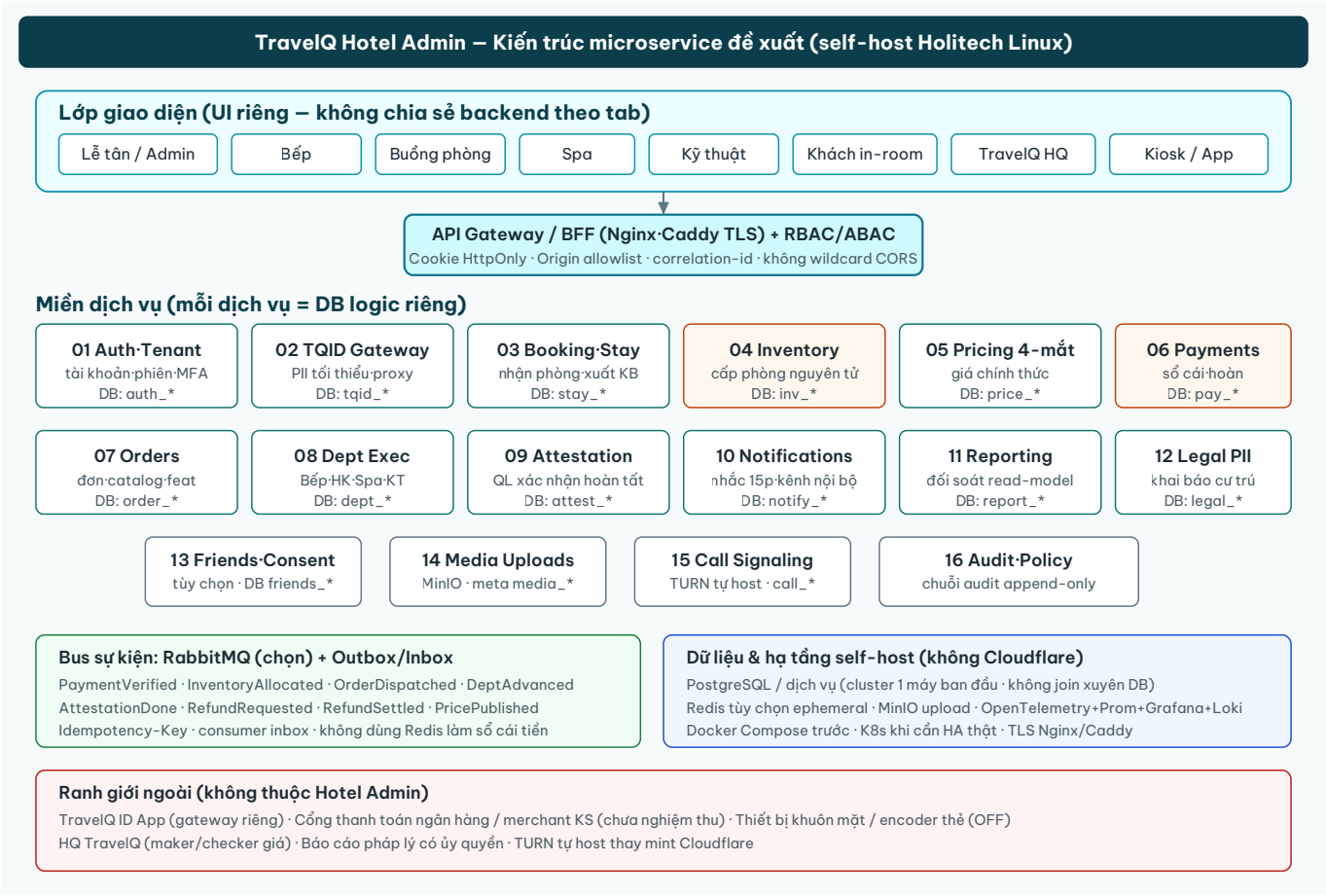
2. Hiện trạng đã xác minh vs đề xuất

Sự thật kỹ thuật từ audit 07/11/2026 (chỉ đọc). Đề xuất audit cũ Node + SQLite một máy phục vụ port monolith tối thiểu — **bị thay** bởi quyết định viết lại microservice; stack dưới chọn PostgreSQL per-service.

Hạng mục	Hiện trạng đã kiểm	Đề xuất microservice
Kiến trúc	SPA + CF Pages Functions; D1 SQLite; chưa microservice	API Gateway/BFF + dịch vụ miền

Hạng mục	Hiện trạng đã kiểm	Đề xuất microservice
API / schema	4 tiền tố API; 10 bảng core; Friends DB tùy chọn	OpenAPI theo dịch vụ; schema tách
KV / R2 / DO / Queues / Telegram	Không thấy phụ thuộc bắt buộc trong cây đã rà	Không giả định; kiểm kê nếu phát hiện thêm
Thanh toán / hoàn	Chưa gateway verified; mô phỏng fail-closed	Ledger riêng + webhook; OFF đến khi có contract
Spa / KT least-priv	Ứng viên OFFLINE đã review core; chưa live / GUI E2E	UI riêng + Dept Exec + Attestation
Attestation QL	Chưa có endpoint / contract triển khai	Dịch vụ Attestation; fail-closed trước đó
TQID / Face / Encoder	Proxy ngoài; face/encoder OFF	Gateway tối thiểu; thiết bị giữ OFF
TURN	Mint Cloudflare	TURN tự host (coturn / provider duyệt)
Frontend nghiệp vụ	Check-in / booking / đơn / bảng bộ phận / báo cáo / cổng xuất: có theo hợp đồng nguồn	Không đồng nghĩa E2E production đã đạt

3. Sơ đồ kiến trúc đề xuất



Hình 1. Lớp UI riêng · Gateway/BFF · 16 miền dịch vụ · RabbitMQ outbox/inbox · Postgres per-service · ranh giới ngoài (TQID, cổng TT, thiết bị). SVG vector nhúng — không phụ thuộc Mermaid ngoài.

4. Catalog miền (~16) và sở hữu dữ liệu

Nguyên tắc: mỗi dịch vụ sở hữu schema, credential, backup riêng. Ban đầu có thể một cluster Postgres trên một máy nhưng **cấm** bảng dùng chung và **cấm** SQL join xuyên database. Giao tiếp liên miền bằng API phiên bản hóa + sự kiện.

#	Dịch vụ	DB logic	Lệnh / sự kiện	Trách nhiệm chính	Không làm
01	Auth · Tenant	auth_*	Login, revoke, version	RBAC+ABAC hotel/dept; MFA admin	PII khách; tiền
02	TQID Gateway	tqid_*	Resolve, proxy scoped	PII tối thiểu; fail-closed	Lưu hồ sơ đầy đủ; chứng nhận provider
03	Booking · Stay	stay_*	CheckIn, Assign, Checkout	Holder policy; gọi cổng Legal	Cấp quỹ; thu tiền
04	Inventory	inv_*	AllocateOnPaid	Cấp phòng nguyên tử; chống oversell	User-hold; hoàn HV
05	Pricing 4-mắt	price_*	Propose / Approve / Publish	Hai tài khoản cá nhân độc lập, không tự duyệt; KS chỉ xem	KS tự duyệt giá phòng
06	Payments · Refunds	pay_*	Intent, Verify, Refund	1 tài khoản KS; mã động; ledger	Mark hoàn sớm; áp hoàn DV lên HV
07	Orders · Catalog	order_*	Create, Cancel(cho), Feat	Snapshot giá; tắt feat chỉ đơn mới	Dispatch khi chưa paid_verified (prepaid)
08	Dept Execution	dept_*	AdvanceStep	Bếp / HK / Spa / KT; expectedStep CAS	Chốt attestation; thanh toán
09	Attestation	attest_*	AttestCompletion	Đề xuất: cổng hoàn tất dịch vụ (không xoá ledger paid)	Bypass bằng login manager; không thay sổ cái tiền
10	Notifications	notify_*	ReminderDue	Nhắc 15 phút	Tự bịa kênh Telegram (chưa có)
11	Reporting	report_*	ProjectFromEvents	Read-model đối soát	Ghi đè ledger
12	Legal · Residence	legal_*	Upsert, AuthorizedExport	PII mã hóa; audit xuất	Full PII trên UI thường
13	Friends · Consent	friends_*	OptIn / Out	Scope tùy chọn	Trộn DB vận hành KS
14	Media	media_* + MinIO	PutObjectMeta	Upload có phép	Log body; lưu định danh thô không mã hóa
15	Call Signaling	call_*	Offer / Answer meta	P2P + TURN tự host	Mint Cloudflare
16	Audit · Policy	audit_*	Append, VerifyChain	Append-only; correlation-id	UPDATE / DELETE audit

4.1 Biên nhất quán mạnh (không tách sai)

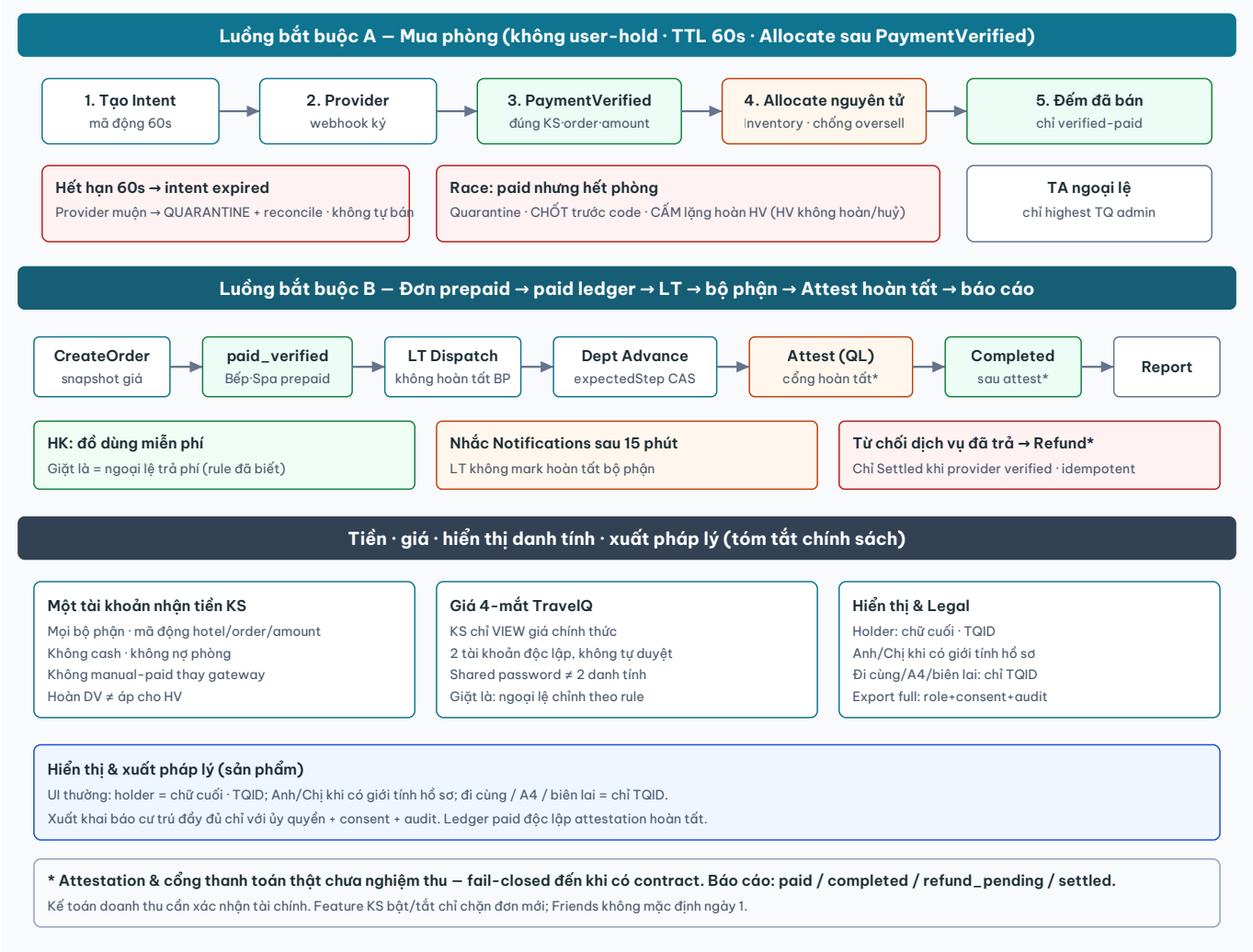
- **Inventory ↔ Payments:** chỉ Allocate sau `PaymentVerified`. Race "tiền về / hết phòng" → quarantine + reconcile. **Không** lặng hoàn HV (HV cấm hoàn/huỷ). Đây là câu hỏi thiết kế mở cần chốt trước code.
- **Orders ↔ Payments ↔ Dept:** prepaid (đồ ăn/uống, Spa): paid_verified (ledger tiền đã nhận) → LT Dispatch → Advance non-terminal → Attest (cổng hoàn tất dịch vụ, đề xuất). Báo cáo tách paid / completed / refund_pending / settled. Kế toán doanh thu cần xác nhận tài chính — không khẳng định mọi "revenue" chỉ sau attestation. Không double-pay / double-count.
- **Pricing ↔ Orders:** đơn giữ snapshot; đổi giá không sửa đơn cũ. Ngoại lệ **giặt là** theo rule đã biết — không suy rộng.
- **UI bộ phận** là client; một bounded context Dept Execution + ABAC đủ ngày 1 trừ khi cần scale tách sau.

5. Stack đề xuất (self-host)

Thành phần	Lựa chọn	Lý do
Runtime	Node.js LTS + TypeScript	Gần logic JS hiện có; OpenAPI nhanh với đội nhỏ
API	REST + OpenAPI 3; gRPC nội bộ tùy chọn	gRPC chỉ khi đo được lợi ích latency/streaming
DB	PostgreSQL per-service	Đa instance / concurrency; PITR. Không SQLite cho microservice đa tiến trình. (Đề xuất SQLite audit = port monolith cũ — không áp dụng ở đây.)
Bus	**RabbitMQ** + Outbox / Inbox	Hàng đợi bền, DLQ, pattern tài chính quen thuộc trên Compose; chọn thay NATS vì đội nhỏ ưu tiên độ chín saga tiền
Cache	Redis tùy chọn	Ephemeral (rate-limit, cache); không chân lý tiền
Object	MinIO (S3-compatible)	Upload nội bộ DC Holitech
TLS / Secret	Nginx hoặc Caddy; Docker / K8s secrets	Không Cloudflare; không secret trong image / log / PDF
Orchestration	Docker Compose trước; K8s khi cần HA thật	Một host ≠ HA
Quan sát	OpenTelemetry → Prometheus + Grafana + Loki	Correlation-id; redact PII
Backup	PG PITR + MinIO mã hóa + restore drill	Mục tiêu đề xuất: RPO ≤ 15 phút · RTO ≤ 4 giờ — chưa đạt, chỉ mục tiêu

Outbox ghi cùng transaction miền; consumer Inbox theo idempotency key — chống double-pay / double-allocate. Tên biến môi trường (names-only, không giá trị): ví dụ `AUTH_DATABASE_URL`, `PAY_DATABASE_URL`, `RABBITMQ_URL`, `MINIO_ENDPOINT` — Holitech chọn sau khi chốt runtime. Docker/Compose chưa cài trên máy audit hiện tại — cần môi trường riêng khi triển khai sau này.

6. Luồng nghiệp vụ bắt buộc (sơ đồ)



Hình 2. Luồng A (mua phòng) và Luồng B (đơn prepaid → ledger paid → attestation hoàn tất). HV tuyệt đối không hoàn/huỷ.

6.1 Mua phòng — không user-hold

1. Khởi tạo thanh toán → mã động TTL **60 giây**, không hold quỹ theo phiên user.
2. Hết hạn → intent expired; provider đến muộn → **quarantine** + reconcile; không tự coi đã bán.
3. Chỉ `PaymentVerified` (chữ ký webhook, đúng merchant KS, đúng amount/order, chống replay) → Inventory Allocate nguyên tử.
4. Verified-paid mới đếm quỹ đã bán. Paid nhưng hết phòng → quarantine; xử lý theo policy TA/dịch vụ nếu được phép — **tuyệt đối không hoàn/huỷ HV**. Đây là design question mở, không "lặng hoàn".

HV: không hoàn, không huỷ. **TA:** ngoại lệ chỉ highest TravelQ admin (không manager/LT). Quy tắc hoàn dịch vụ **không** áp blanket lên HV.

6.2 Đơn → tiền (ledger) → LT → bộ phận → attestation hoàn tất → báo cáo

1. Tạo đơn (snapshot giá server).
2. Prepaid (Bếp đồ ăn/uống; Spa): `paid_verified` trước LT dispatch. HK: đồ dùng miễn phí; giặt là trả phí theo rule.
3. LT dispatch → Dept Advance + `expectedStep`; non-terminal. LT không đánh dấu hoàn tất bộ phận. Nhắc sau 15 phút.
4. Đề xuất: Attestation QL là cổng **hoàn tất dịch vụ** — không xoá / không thay sổ cái đã paid_verified. Hiện chưa có contract → fail-closed cho bước hoàn tất.

5. Dịch vụ đã trả bị từ chối: RefundRequested → gọi provider → RefundSettled chỉ khi verified; retry idempotent; không mark sớm; không double-count.
6. Báo cáo / đối soát tách: paid · completed · refund_pending · settled. Ghi nhận kế toán doanh thu cần xác nhận tài chính.

Mọi khoản về **một tài khoản nhận tiền đích KS**. Không cash / nợ phòng / manual-paid. Mỗi giao dịch: mã động hotel / order / amount. KS bật/tắt nhóm đơn: chỉ đơn mới; giữ lịch sử. KS VIEW giá; TravelQ maker/ checker = **hai tài khoản cá nhân độc lập, không tự duyệt** (mật khẩu dùng chung ≠ hai tài khoản độc lập).

6.3 Hiển thị danh tính & xuất pháp lý

- UI thường: holder = **chữ cuối · TQID**; Anh/Chị chỉ khi có giới tính hồ sơ. Người đi cùng / A4 / biên lai: chỉ TQID.
- Xuất khai báo cư trú: full info chỉ với ủy quyền + consent + audit; UI thường không full PII.

7. Bảo mật (đề xuất)

- **RBAC + ABAC:** mọi request gắn hotel / department / `account-version` đã ký phía server; role không do trình duyệt chọn.
- **MFA** cho admin và cặp 4-mắt giá.
- **Service identity:** mTLS (danh tính dịch vụ) **và** phân đoạn mạng giữa service — phân đoạn mạng không thay thế danh tính; webhook provider: chữ ký + timestamp + nonce chống replay.
- Truy vấn tham số hóa; PII hạn chế mã hóa at-rest; retention tách; audit pháp lý.
- Phiên khách in-room: gắn phòng, TTL ngắn, khóa khi checkout.
- CSRF: SameSite + Origin allowlist; cookie **HttpOnly Secure**; **cấm** wildcard origin.
- Không secret / hash verifier / token / số định danh cá nhân trong tài liệu hay ví dụ.

8. Vận hành · quan sát · sao lưu

- Compose: gateway, services, Postgres (logic tách), RabbitMQ, MinIO, (Redis), OTel collector, Prometheus, Grafana, Loki.
- Log: correlation-id bắt buộc; không PII; không body thanh toán.
- Backup: WAL PITR Postgres từng DB logic; object MinIO phiên bản + mã hóa; **test restore** định kỳ.
- Mục tiêu (chưa đạt): RPO ≤ 15 phút · RTO ≤ 4 giờ trên single-host. Failover HA là giai đoạn sau khi có nhu cầu thật đã đo.

9. Lộ trình · ước lượng · rủi ro

Giả định đội: **2–3 engineer** full-stack + **0.5 ops** quen Node/Postgres; song song với xác nhận nghiệp vụ từ anh Long. Đơn vị: **person-week (pw)**. Không hứa ngày calendar / chi phí tiền. Không bịa số liệu test/CI hiện tại.

Giai đoạn	Deliverable	Acceptance (không %)	Ước lượng
G0 Chốt	Biên dịch vụ + quyết định race HV + app con ngày 1	Biên bản xác nhận viết	1–2 pw
G1 Nền	Compose, Gateway, Auth, Audit, OTel	Health + login fail-closed + audit append	3–5 pw
G2 Phòng & tiền	Inventory + Payments ledger (adapter lab kiểm soát)	Không oversell test đua; HV không hoàn	5–8 pw
G3 Ở & đơn	Booking/Stay + Orders + Dept + Notify	Prepaid + reminder 15p + CAS	5–8 pw
G4 Giá & BC	Pricing 4-mắt + Reporting	Hai tài khoản độc lập, không tự duyệt; KS view-only	2–4 pw
G5 PII & ngoài	Legal cổng; TQID gateway; TURN tự host	Export có audit; TQID fail-closed	3–5 pw
G6 Cứng hóa	Attestation (cổng hoàn tất); chaos/ idempotency; restore drill	Hoàn tất dịch vụ có attest khi bật; ledger paid độc lập; restore đạt mục tiêu giấy	3–5 pw

Tổng thô: ~22–37 pw (ước lượng lập kế hoạch). Với giả định đội 2–3 engineer, quy đổi thô ~**8–19 tuần** trước các trễ bên ngoài (gateway/TQID/pháp lý/onboarding) — không phải cam kết ngày giao.

9.1 Rủi ro chính

1. Onboarding cổng thanh toán / merchant từng KS.
2. Contract TQID & thay TURN khỏi Cloudflare.
3. Khuôn mặt / encoder — giữ OFF đến khi có hardware + contract.
4. Xuất pháp lý & retention PII.
5. Race inventory ↔ payment (đặc biệt HV).
6. Migration JSON-state monolith → DB tách / nhất quán phân tán.
7. Ops single-host (backup / restore / disk / secrets).

9.2 Phải xác nhận nghiệp vụ trước khi code

- Xử lý "paid nhưng hết phòng" với HV (không hoàn) so với TA / dịch vụ.
- Phạm vi ngoại lệ giá giặt là.
- Attestation: ai ký, SLA 15 phút, có ủy quyền không.
- App con ngày 1 (Friends không mặc định).
- Ai cấp manager đầu trên Holitech (kênh bảo mật — không qua PDF).

10. Thuật ngữ

Thuật ngữ	Nghĩa trong tài liệu
Microservice	Dịch vụ deploy độc lập, sở hữu DB logic riêng, giao tiếp API / sự kiện
Outbox / Inbox	Ghi sự kiện cùng transaction; consumer khử trùng idempotent
paid_verified	Thanh toán đã xác minh bởi provider / webhook — không phải tự khai UI
ABAC	Kiểm soát truy cập theo thuộc tính (hotel, dept, account-version)
4-mắt	Maker và checker phải là hai tài khoản cá nhân độc lập; không tự duyệt
HV / TA	Hạng quỹ phòng theo nghiệp vụ TravelQ; HV không hoàn / huỷ
RPO / RTO	Mục tiêu mất dữ liệu tối đa / thời gian phục hồi tối đa (chưa đạt)
Fail-closed	Thiếu contract / secret / provider → từ chối, không mock production
Quarantine	Trạng thái giữ kết quả tiền/phòng chưa quyết — chờ reconcile, không tự bán / tự hoàn HV

11. Kết luận

Đây là **bản thiết kế đề xuất** để Holitech self-host theo hướng microservice, kế thừa hợp đồng nghiệp vụ đã biết từ nguồn Hotel Admin hiện có, đồng thời **không** tuyên bố đã triển khai, đã nghiệm thu cổng thanh toán, khuôn mặt, encoder, attestation, hay chứng nhận danh tính TQID. Sự thật đã kiểm (monolith CF+D1, 4 API, 10 bảng core, không KV/R2 bắt buộc trong phạm vi rà, payment/face/encoder/attestation chưa xong) được tách rõ khỏi phần đề xuất.

Bước tiếp theo: anh Long và Holitech chốt các câu hỏi mục §9.2, rồi mới mở giai đoạn G0 / G1.

— Hết tài liệu đề xuất · không chứa secret · không dữ liệu cá nhân —