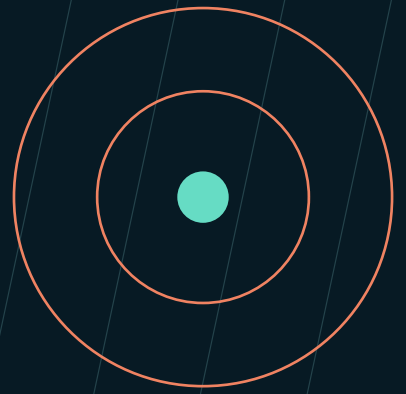




WORKING PRODUCT CONCEPT

Foxes Travel OS

The travel-business operating system built from one brief.

Product definition, experience architecture, chat control, technical architecture, governance model, connector catalogue and MVP roadmap for DMCs and tour operators.

Audience	Product owners, technical leads and pilot operators
Status	Concept definition - no production or provider readiness implied
Date	September 2026 - v0.2 (adds chat control, storefront preview, connector catalogue)

01 / PRODUCT DEFINITION

A complete digital business system for travel operators

Foxes Travel OS begins with the operator's business model, not a list of software features. The output is a connected operating system tailored to the operator's products, markets, channels, rules and brand.

The customer

DMCs, tour operators and experience businesses that sell direct or through agents, coordinate suppliers and support travellers across web, mobile, messaging and voice.

The core promise

Describe the business once. Receive a launch blueprint and an integrated system for selling, booking, collecting money, coordinating delivery and supporting the guest.

Product thesis

Travel businesses do not need six disconnected implementation projects. They need one travel-native operating model expressed consistently across customer channels, commerce, CRM, AI roles, payments and operations.

One business brief becomes one governed travel platform.

What the concept is - and is not

It is	It is not
A travel-business configuration and orchestration platform	A generic no-code app builder
A shared travel core across channels and workflows	A website with disconnected plug-ins
A governed AI operating team with predefined travel roles	An unrestricted chatbot or agent graph
A platform the owner changes by talking to it, inside approval rules	A tool that needs a developer for every price, date or page change
A phased product vision that starts with one revenue loop	A claim that every module already exists or is provider-connected

02 / EXPERIENCE

From business brief to an operating company

The user journey must make a complex implementation feel guided without hiding consequential choices. The system proposes; the owner reviews, connects and approves.

Stage	What happens
1. Describe	Destinations, products, markets, languages, channels, payment rules, suppliers, brand and team.
2. Blueprint + preview	Foxes Travel OS maps journeys, modules, data, roles, integrations, policies and launch dependencies, and renders a first storefront preview from the brief.
3. Configure	The operator reviews brand, catalogue, rates, policies, roles, content and the approval matrix.
4. Connect	Each external service passes identity, credential, webhook, sandbox and production-canary gates.
5. Launch	One measurable revenue loop goes live before optional channels and modules are enabled.
6. Operate by chat	The owner changes prices, dates, pages and rules by telling the workspace; content applies with undo, money and policy wait for approval.

The launch blueprint

Business model
Brand, markets, languages, currency, products, rate structure, policies, supplier model and commercial goals.

System model
Customer journeys, operator workflows, data entities, AI roles, approvals, connectors, readiness gates and phased backlog.

Generated surfaces
A brand site, mobile experience, booking flow, operator workspace, CRM views, AI-team configuration and communications templates.

Required human decisions
Pricing policy, cancellation policy, supplier authority, data ownership, payment account, legal terms, brand approval and go-live acceptance.

Design principle: progressive commitment

The first interaction may generate a useful draft blueprint without credentials or external side effects. Sensitive data, provider credentials, messaging, money movement and supplier commitments are introduced only when their exact destination and consequence are clear.

02 / EXPERIENCE CONTINUED

Change the platform by talking to it

Once live, the operator does not open six admin panels. An instruction in plain words becomes one scoped change that is interpreted, previewed on every surface, decided against policy, recorded and reversible. Vague instructions get a question back, never a guess.

Outcome	What it covers	Examples
Applied now, logged	Text, photos, translations, availability, capacity and assistant tone. Live at once with a before/after record and one-tap undo.	Close Saturday departures for Cairo day tours in August. Translate the balloon ride page into German.
Waits for approval	Prices, promotions, deposits, refund policy, retiring a product and access. Previewed on every surface, then approved or rejected by the named role.	Add a 10% Eid discount on all Luxor day tours until 20 April. Raise the deposit to 30%.
Asks first	Any instruction missing an amount, a product or a period. Nothing changes until the detail is supplied.	Run a promotion on Nile cruises in August (amount missing).

Interpretation contract

- Interpret: classify the change and extract scope, amounts, dates and languages.
- Scope: resolve the products touched; no money change without an amount, no availability change without dates.
- Preview: list every surface that changes and what travellers will see; confirmed bookings and sent quotes are never rewritten.
- Decide: apply, wait for the named approver, or ask. Owners can add gates; the defaults below cannot be removed.
- Record and undo: instruction, interpreted change, approver, surfaces touched, and a one-tap rollback everywhere at once.

Default human gates every pilot starts with

Action	Approved by	Recorded
Send a price outside the published rules	Owner or sales lead	Draft price, rule breached, decision
Confirm a booking	Verified payment event	Payment reference, amount, time
Issue a refund or accept a cancellation	Owner or finance	Policy applied, amount, reason
Commit a supplier, guide or driver	Operations lead	Request, confirmation, cost
Answer without an approved source	Handed to a person	Question, missing source, who took it
Change a price, deposit or policy from chat	Owner or finance	Instruction, interpreted change, approver
Connect an external system	Owner, after a verified test	Connector, test result, date

03 / PRODUCT MODULES

Six modules, one travel context

Each module can be phased independently, but none should fork the core product, traveller, trip or financial truth.

Module	Primary capabilities	Shared context
Brand and channels	Website, mobile, content, web enquiry, messaging and voice	Business profile, products, policies, traveller context
Travel commerce	Products, rates, availability, extras, cart, booking and vouchers	Catalogue, price rules, inventory and trip
AI operating team	Sales, itinerary, pricing, booking, operations and guest-care roles	Scoped knowledge, tools, approval matrix
CRM and operations	Enquiries, travellers, trips, tasks, suppliers and service delivery	Conversation, itinerary, commitments, evidence
Payments and revenue	Deposits, balances, refunds, commissions, margin and reconciliation	Verified events, ledger entries and exceptions
Governance and evidence	Identity, tenant isolation, policies, approvals, audit and readiness	Actor, source, status, decision and outcome

Cross-module invariants

- Every record belongs to one tenant workspace and carries a stable identifier.
- A product, policy or rate has an effective version; customer output names the version used.
- A draft, approved action, queued action and verified external outcome are different states.
- Money and supplier commitments require policy checks plus a named approval where configured.
- Every external event is authenticated, idempotent and attributable to a connector and tenant.
- Customer-facing AI always identifies itself and hands off when approved evidence is absent or conflicting.

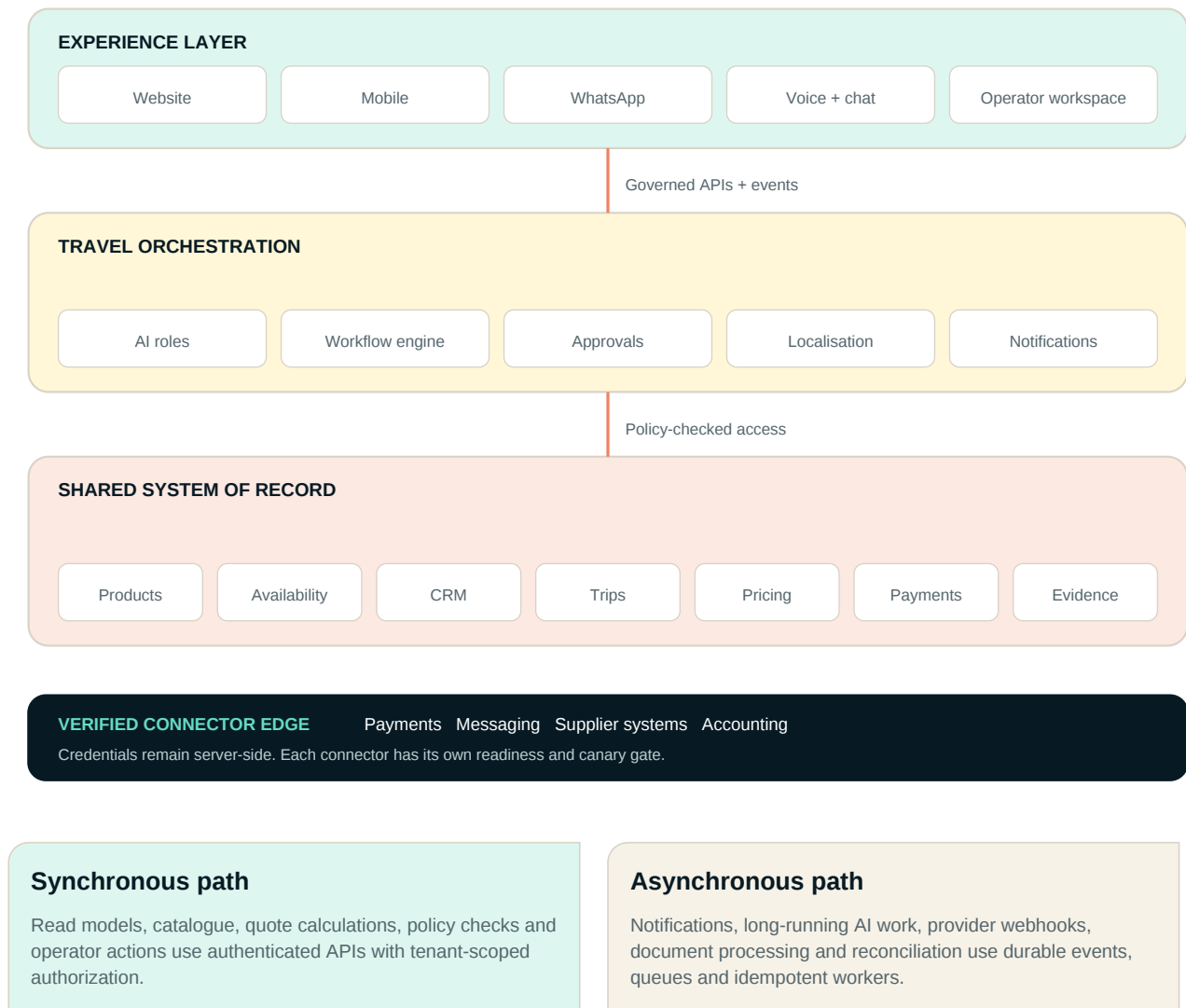
Shared travel core
Product, availability, policy, traveller, trip, proposal, payment, supplier work and evidence are canonical domain records.

Composable experiences
Web, mobile, messaging, voice and operator workspace are clients of the same policies and domain services - not separate business systems.

04 / REFERENCE ARCHITECTURE

Layered for speed, isolation and verifiable execution

The reference architecture separates customer experience, travel orchestration, canonical data and external provider execution. This keeps generated surfaces flexible while business truth remains controlled.



04 / ARCHITECTURE CONTINUED

Implementation boundaries

Clear boundaries keep generated experiences flexible while canonical business truth, side effects and provider credentials remain controlled.

Boundary	Responsibility
Experience apps	Web, installable/mobile surfaces and operator workspace. No provider secrets and no direct database access.
API gateway / BFF	Identity, tenant context, rate limits, validation and client-oriented response composition.
Travel domain services	Catalogue, pricing, CRM, trip, booking, supplier, payment and evidence state machines.
Orchestration	Workflow state, AI-role dispatch, human tasks, approval rules, retries and compensation.
Connector workers	Provider credentials, outbound calls, webhook verification, idempotency and readiness canaries.
Data platform	Transactional store, object storage, search/retrieval index, event log and analytics projections.

Command boundary

All consequential changes enter through authenticated, tenant-scoped domain commands with policy, state and idempotency checks.

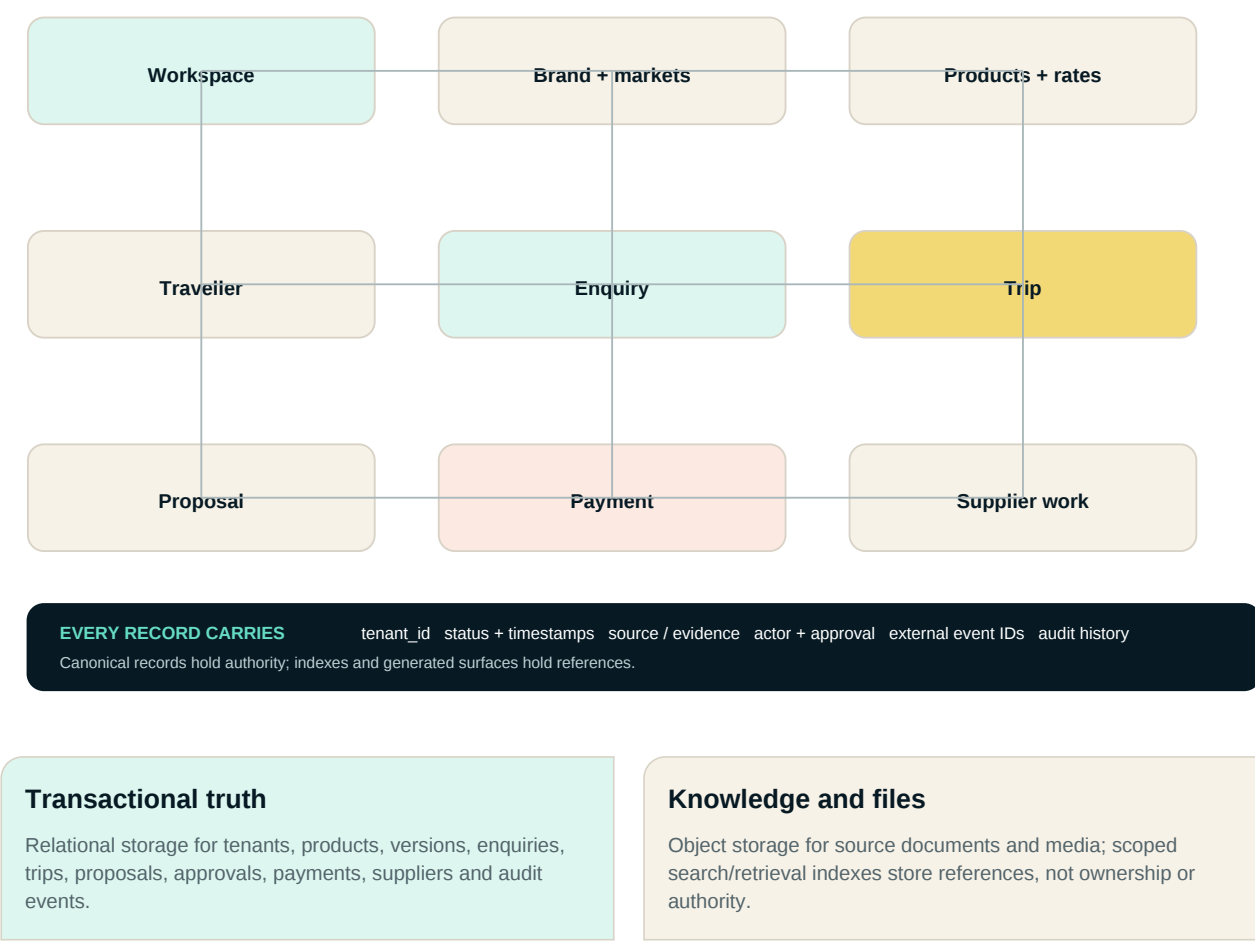
Evidence boundary

Provider responses and webhooks are normalised into internal events before a trip, payment or commitment changes state.

05 / CANONICAL DATA

One trip record connects demand, money and delivery

The data model should make the golden workflow observable end to end. The key design decision is to keep commercial, operational and evidence records linked rather than copying them into channel-specific stores.



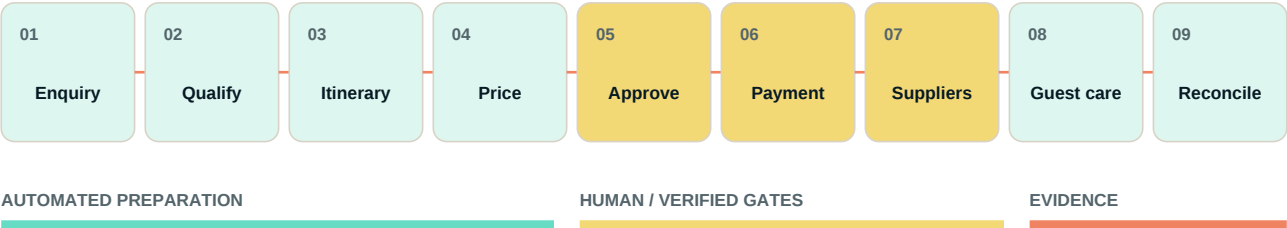
State-machine examples

Record	Illustrative states	Terminal evidence
Proposal	draft -> reviewed -> approved -> sent -> accepted / expired	Approved version, actor, timestamp and delivery receipt
Payment	created -> pending -> verified / failed / refunded	Signed provider event plus internal ledger entry
Supplier commitment	requested -> internally approved -> sent -> confirmed / declined	Supplier-origin confirmation or signed contract
AI action	requested -> grounded -> policy-checked -> approved -> executed / blocked	Sources, policy result, actor and external outcome

06 / AI TEAM AND WORKFLOW

Specialised travel roles sharing one governed context

The product should deploy predefined travel roles and workflows. Operators configure business scope, tone, languages and approved knowledge; they do not assemble an unrestricted technical agent graph.



Role	Owns	Hard boundary
Sales	Qualify enquiries, prepare grounded replies, capture structured constraints	No unsupported rates or availability
Itinerary	Assemble plans from approved products and destination knowledge	No invented inventory
Pricing	Calculate cost, markup, tax, commission and margin	Below-floor price requires exception approval
Booking	Prepare proposals, deposit requests, vouchers and traveller records	Payment is complete only after verified event
Operations	Coordinate suppliers, tasks, transfers, documents and service delivery	Internal status is not supplier commitment
Guest care	Answer pre-trip and in-destination questions with context	Hand off when evidence or authority is missing

Agent execution contract

- Input: tenant, authenticated actor, trip/enquiry context, permitted tools, scoped knowledge and requested action.
- Plan: a structured intent and required evidence, never an invisible chain of business commitments.
- Check: grounding, authorization, policy, budget/margin, approval need and connector readiness.
- Execute: through idempotent domain commands and connector workers, not direct free-form provider calls.
- Record: sources, policy result, approval, tool result, external identifiers and customer-visible output.

07 / SECURITY AND GOVERNANCE

Safety is built into the architecture

Safety is not a prompt property. The platform handles personal data, money, supplier commitments and third-party credentials, so controls belong in identity, policy, state machines, encryption, execution and evidence layers.

Control	Implementation intent	Applies to
Identity	Verified user identity, workspace membership, role and session assurance	Every request and approval
Tenant isolation	tenant_id enforcement, row-level access policy, scoped object paths and isolated retrieval	Queries, files, search and events
Secrets	Server-side encrypted connector credentials; no secret values returned to clients	Connector configuration and workers
Authorization	Capability and object-level checks plus step-up approval for high-risk commands	Money, discounts, refunds, suppliers
Webhooks	Signature verification, replay window, idempotency key and tenant/provider binding	Payments, messaging and partner events
AI governance	Approved source scope, tool allow-list, policy result and deterministic high-risk gates	Every agent turn with side effects
Audit	Append-only actor, source, decision, action, result and external evidence references	All material state changes
Privacy	Data minimisation, retention controls, export/deletion workflow and redacted logs	Personal and conversational data

Readiness is explicit and multi-dimensional

Local / sandbox readiness

The module compiles, tests pass, sample data works and provider calls are simulated or sandboxed. This is not customer delivery proof.

Production readiness

Identity, billing, secrets, provider account, webhooks, canaries, telemetry, rollback and owner acceptance each pass separately.

08 / INTEGRATIONS

Connect through verified adapters, not UI seams

A settings screen is not an integration. Each adapter needs an explicit contract, encrypted credentials, sandbox verification, signed inbound events, idempotent outbound work and a production canary.

Adapter	Contract	Go-live gates
Payments	Create payment requests, verify events, refunds and reconciliation	Account ownership; webhook; sandbox; live canary; ledger match
Messaging	Templates, inbound threads, approved sends and delivery evidence	Sender ownership; templates; signatures; opt-in; delivery canary
Supplier / inventory	Products, availability, reservations, cancellations and confirmations	Contracted API; mapping; retry; source timestamps; confirmation evidence
Maps / geocoding	Places, route context and transfer calculations	Quota; privacy; cache; fallback; accuracy review
Accounting	Invoices, receipts, payouts, commissions and ledger export	Chart mapping; tax rules; idempotency; reconciliation
Identity	Operator sign-in, workspace membership and invitations	Email/domain policy; MFA; callback validation; recovery
Analytics	Funnel, operations and financial projections	Event schema; consent; PII rules; backfill and monitoring

Connector state model

State	Meaning
not_configured	No credentials or account relationship exists.
configured	Encrypted configuration saved; no functional proof yet.
sandbox_verified	Contract exercised against sandbox or simulated provider.
production_pending	Account and configuration exist; canary or final approval missing.
live_enabled	Production canary passed and monitoring/rollback are active.
degraded / blocked	Evidence shows an operational failure or policy stop; execution fails closed.

Failure rule

If readiness is lost, preserve the trip, record the attempt, retry only when safe and route the operator to a manual path. Never promote queued or simulated work to delivered, paid or confirmed.

08 / INTEGRATIONS CONTINUED

Connector catalogue by phase

Names show the connectors intended for the pilot and for later phases so an operator can see where current tools fit. They describe the plan, not existing partnerships; each one is switched on only after a verified test. Logos and names are trademarks of their owners.

Area	Pilot	Later
Payments	Stripe, Paymob	Fawry, PayPal, Apple Pay, Google Pay
Messaging	WhatsApp Business API, Email (Mailgun), Website chat	Gmail and Outlook inboxes, SMS and voice
Marketplaces and OTAs	-	Viator, GetYourGuide, Tripadvisor Experiences, Klook, Expedia
Partners and channel managers	Partner portal (built in)	Hotel desk logins, Bokun, Rezdy
Operations	Google Maps and pickup routing, supplier requests by email and API	Driver and guide app, WhatsApp supplier groups
Finance and back office	Google Sheets export, CSV and accounting export	Xero, QuickBooks
Marketing and analytics	Google Analytics 4	Meta Pixel, Google Ads conversions, Mailchimp or Brevo, review requests
Identity and data	Import from spreadsheets, full data export, daily backups	Google and Apple sign-in

How a connector goes live

1. Credentials go in the vault

Keys and tokens are stored encrypted per operator. Nobody, including the AI roles, sees them in plain text.

2. A verified test proves it

A real test payment, message or sync runs and its result is recorded. If it fails, the connector stays off.

3. Switched on, with evidence

Only then does the connector go live. The date, the test result and who approved it stay on the audit trail. Readiness lost later fails closed and routes the operator to a manual path.

09 / MVP ROADMAP

Prove one revenue loop before building the full vision

The highest-risk mistake is attempting the entire platform before one operator can reliably convert and operate a real booking. The MVP should test the shared-core thesis with the smallest commercially complete workflow.

Phase	Scope	Exit evidence
Phase 0 - Blueprint	Owner brief, product catalogue model, policy pack, journey map, role map, architecture and acceptance criteria	Signed pilot definition
Phase 1 - Revenue loop	Web enquiry/booking, CRM, AI sales + itinerary, pricing/margin, operator approval, payment request, chat-driven content and availability changes, evidence trail	A real enquiry reaches a verified deposit with full evidence; the owner changes dates and pages by chat without a developer
Phase 2 - Delivery loop	Supplier tasks, vouchers, guest care, balance collection and reconciliation	A pilot trip completes without loss of context or unexplained margin
Phase 3 - Channel expansion	Messaging, mobile/PWA, voice reply, partner/supplier adapters, chat-driven pricing and policy changes with approval, localisation	Each channel uses the same trip and policy truth; a second channel adds no second copy of the trip
Phase 4 - Platform scale	Templates, tenant onboarding, marketplace/adapters, analytics and multi-market controls	Repeatable launch with measurable readiness

Recommended Phase 1 inclusions

- Responsive customer website with product discovery, enquiry and one bookable flow.
- Operator workspace with catalogue, rates, policies, enquiries, proposals, trips and evidence.
- Sales and itinerary AI roles grounded only in approved pilot knowledge.
- Deterministic pricing and margin policy with a named human approval stop.
- One verified payment connector, one messaging path and production-readiness dashboards.
- Automated tests for tenant isolation, authorization, state transitions, signatures and idempotency.

Defer until the loop is proven

Native mobile apps, broad supplier marketplace, unrestricted custom workflows, many payment/messaging providers, complex B2B distribution and predictive optimisation should follow evidence from the pilot - not precede it.

10 / NON-FUNCTIONAL ARCHITECTURE

Built to remain trustworthy under real operations

Quality targets should be measurable before implementation. The exact numbers belong in the pilot SLO agreement; the following defines the capability areas that must exist.

Quality area	Required capabilities
Availability	Health checks, graceful degradation, queue buffering, retry budgets and runbooks for critical booking/payment paths
Performance	Fast catalogue reads, predictable quote calculation, async AI work, CDN/static delivery and query/index budgets
Reliability	Idempotent commands, workflow checkpoints, deduplicated webhooks, compensation and reconciliation jobs
Security	Least privilege, encryption, dependency and secret scanning, audit review, abuse limits and incident response
Privacy	Purpose limitation, retention, export/deletion, redaction and region/vendor assessment
Observability	Tenant-aware traces, structured events, error budgets, workflow state, connector health and business outcome metrics
Accessibility	Keyboard/touch support, semantic structure, readable contrast, reduced motion and multilingual/RTL verification
Deployability	Environment parity, migrations, preview environments, canaries, rollback and explicit local-vs-production evidence

Delivery architecture

Application delivery
Versioned web/operator applications, infrastructure-as-code, database migrations, feature gates and reversible releases.

Evidence delivery
Automated tests, contract tests, security checks, browser/device captures, provider canaries and production telemetry archived per release.

Definition of done

A module is complete only when the intended role can finish the workflow, negative and authorization paths pass, applicable provider outcomes are verified, telemetry and rollback exist, and the owner accepts the evidence and limitations.

11 / DECISIONS AND NEXT ACTIONS

What is good enough for now - and what must be decided next

This document establishes a coherent current direction without pretending the product, provider stack or brand are final.

Decisions captured in this concept

- Brand identity: Foxes Travel OS. Product architecture remains modular so visual identity can evolve without forking the operating model.
- Primary buyer: DMCs and tour operators with multi-channel sales and supplier coordination.
- Primary product motion: business brief -> blueprint -> configured platform -> phased launch.
- Architecture: experience layer over governed travel orchestration and a shared canonical system of record.
- AI model: predefined travel roles with scoped knowledge, deterministic high-risk gates and evidence.
- MVP: one end-to-end revenue loop, followed by delivery operations and channel expansion.
- Chat control: content, translation, availability, capacity and assistant tone apply immediately with undo; prices, promotions, deposits, refund policy, retiring products and access wait for a named approval.
- Connector catalogue: the pilot set is Stripe, Paymob, WhatsApp Business API, Mailgun email, website chat, Google Maps, Sheets and CSV export, Google Analytics 4, the built-in partner portal, spreadsheet import, full export and daily backups.

Open owner decisions

Decision	Why it matters
Pilot operator and market	Determines products, language, regulation, channels, suppliers and proof criteria.
Business model	Subscription, setup fee, usage, transaction revenue or hybrid affects tenancy, billing and support.
Mobile strategy	PWA-first versus native affects MVP cost, release operations and device capabilities.
Initial providers	Payment, messaging, identity and supplier choices shape integration and compliance work.
Ownership boundary	Clarify who owns content, traveller data, connector accounts, domains and generated assets.
Pilot acceptance	Define the real booking, operational and reliability evidence required to expand.

Immediate next actions

Product
Run a structured discovery with one pilot operator; turn their real catalogue, policies and enquiry-to-delivery workflow into the first blueprint and clickable product flow.

Technical
Write Phase 1 domain contracts, state machines, tenant/auth model, connector interfaces, acceptance tests and a delivery plan with explicit provider and production gates.

12 / THE CONCEPT SITE

What is demonstrated today, and what is not built

The public concept site at foxes-travel-os.netlify.app is the working companion to this document. Everything on it runs in the visitor's browser from fixed rules; nothing is provisioned, stored or sent.

Demonstrated on the site	How it works today	In the product
Brief to storefront preview and launch blueprint	A rule-based engine reads the brief and renders a storefront draft (brand, catalogue, booking widget, languages, channels) plus systems, AI roles, human gates, pilot sequence, open questions and a coverage score. Copy or download as Markdown.	The same brief provisions a real site on the operator's domain, connected to their payment and messaging accounts after verified tests.
Chat control	An instruction becomes a scoped change with surfaces, preview, decision (apply / wait for approval / ask), evidence and undo, with a working approve, reject and undo lifecycle.	The same instruction is interpreted by the AI roles against the live tenant and gated by the same rules.
Connector orbit and catalogue	Pilot and later connectors around a shared travel core, one tile per brand.	Each connector passes the vault, verified test and switch-on gates before use.
Guardrails, workflow, roadmap, FAQ	Explains the operating model, the nine-step golden workflow, the default human gates and phase exit criteria.	Unchanged: these are the contract the build is held to.

Not built yet

No platform is generated or deployed from the site. Mobile apps, the AI voice agent, live payments, CRM and WhatsApp are described, previewed and planned, not connected. Building the generator that provisions a tenant across Foxes Technology's existing products is a separate project with its own plan, gates and evidence.