Web Suite System Architecture and Trust Boundaries

1. Purpose

This document defines the structural shape of the Web Suite and the non negotiable trust boundaries between services.

It exists to prevent architectural drift and accidental boundary violations.

---

2. Services

Surface Detail

Type: Browser based web client

Exposure: Public

Environment: Runs in user browser

Responsibilities:

- Render UI
- Authenticate users via Auth0
- Call public API endpoints on Polite Intervention
- Never access internal services directly

Constraints:

- No internal secrets
- Any VITE_ environment variable is publicly exposed

Polite Intervention

Type: Public edge API service

Exposure: Public

Environment: Server

Responsibilities:

- Validate Auth0 bearer tokens
- Authorize user access
- Expose public API endpoints required by Surface Detail
- Call Considered Response via server to server communication
- Enforce error envelope consistency

Constraints:

- Holds INTERNAL_SERVICE_SECRET
- Must never expose internal secrets to clients
- Must not mock internal services

Considered Response

Type: Internal domain service

Exposure: Internal only

Environment: Server

Responsibilities:

- Domain logic
- Data access
- Deterministic domain behavior

Constraints:

- Must not be callable directly from browsers
- Must require server to server authentication
- Must not rely on browser tokens
- Must fail fast if required internal configuration is missing

---

3. Call Direction

The only allowed call path is:

Surface Detail → Polite Intervention → Considered Response

Surface Detail must never call Considered Response.

Considered Response must never be reachable from a browser.

4. Authentication Model

User Authentication

- Surface Detail authenticates users via Auth0.
- Auth0 bearer tokens are sent to Polite Intervention.
- Polite Intervention validates Auth0 tokens.

Considered Response does not validate Auth0 tokens directly unless explicitly required by architecture.

Internal Service Authentication

- Polite Intervention calls Considered Response using a shared secret.
- Header name: X-Internal-Secret
- Secret value: INTERNAL_SERVICE_SECRET
- Secret stored only in server environments.
- Considered Response rejects all requests without valid internal secret.

5. Secrets Policy

Browser

- No internal secrets.
- No service to service secrets.
- No reliance on hidden client side values.

Polite Intervention

- Stores INTERNAL_SERVICE_SECRET.
- Stores CONSIDERED_RESPONSE_BASE_URL.
- Never exposes internal secret in responses.

Considered Response

- Stores INTERNAL_SERVICE_SECRET.
- Validates internal secret on all routes.

6. Error and Response Rules

- All services return JSON error envelopes.
- No HTML error pages.
- No stack trace leakage.
- Internal failures propagate as structured errors.

7. Non Negotiable Constraints

- No direct browser access to internal services.
- No internal secrets in client code.
- No mock upstream services in production paths.
- No silent fallback data when upstream fails.
- Considered Response must fail at startup if INTERNAL_SERVICE_SECRET is missing.

8. Observability Boundaries

- Correlation IDs propagate across all service calls.
- Errors are logged at each layer with service context.
- Internal service failures are visible in the public edge logs.

This document defines system shape.

Feature specifications must conform to it.

If a slice contradicts this document, the slice must change, not the boundary.