

Tóm tắt và Kế hoạch Toàn cảnh

NATT-OS Gold Master hiện gặp **105 lỗi** biên dịch lớn do **drift schema** và các thay đổi kiến trúc chưa được cập nhật đồng bộ. Thay vì vá tạm bằng script tự động, chúng ta sẽ tiến hành **phẫu thuật thủ công** theo 5 cụm lỗi chính. Mỗi cụm được xử lý lần lượt với các commit nhỏ có kiểm duyệt, đảm bảo tính **integrity**. Cụ thể:

- **Cụm 1 – Warehouse Cell (Mất đồng bộ domain):** Sửa class/thực thể `WarehouseItem` và `WarehouseItemProps`.
- **Cụm 2 – Types Drift (Schema lỗi):** Dò tìm và chỉnh sửa `types.ts`, thêm/bớt trường chính xác (không widen enum).
- **Cụm 3 – Đường dẫn Import Lỗi:** Sửa các import cũ (sales-cell, cells/... thành paths mới, safe).
- **Cụm 4 – Enum/Status misuse:** Thay thế `'FAILED' → 'FAILURE'`,
`'RECOVERED' → 'SUCCESS'` trong code.
- **Cụm 5 – Thiếu method hạ tầng (EventBridge, SmartLink, Audit, Threat):** Bổ sung `subscribe`, `createEnvelope`, sửa `logAction`, thêm `getHealth/subscribe` v.v.

Quá trình sẽ chạy `npx tsc --noEmit` sau mỗi cụm để kiểm tra. Dưới đây là phân tích chi tiết từng phần và bảng khắc phục.

1. Inventory các file `types.ts`

Chạy lệnh sau để liệt kê tất cả `types.ts` trong repo:

```
find src -name "types.ts"
```

Kết quả (bao gồm cả file liên quan):

```
- src/types.ts
- src/types/accounting.types.ts
- src/cells/business/pricing-cell/domain/types/pricing.types.ts
- src/cells/business/pricing-cell/domain/value-objects/gold-types.ts
- src/cells/business/promotion-cell/domain/value-objects/promotion-types.ts
- src/cells/shared-kernel/shared.types.ts
- src/cells/infrastructure/shared-contracts-cell/domain/contract.types.ts
- src/core/core/signals/types.ts
- src/core/signals/types.ts
- src/core/gatekeeper/types.ts
- src/core/state/types.ts
- src/governance/types.ts
- src/governance/gatekeeper/types.ts
```

(Lưu ý: Có một số file `types.ts` trong nhánh worktree `.claude/worktrees/...` bị trùng, ta chỉ quan tâm tới `src/...` chính.)

2. Trùng lặp & Khác biệt Interface/Type

Ta xem xét toàn bộ `export interface/type/enum` trong mỗi file trên và tìm tên trùng giữa các file:

- `WarehouseItemProps`: Xuất hiện cả trong *domain* và *infra* của Warehouse-cell (xem mục sau). Đây là nguồn gốc chính của nhiều lỗi.
- **Các tên khác** (`ModuleConfig`, `AccountingMappingRule`, `OperationRecord`, `ApprovalTicket`, `QuantumState`, `ConsciousnessField`, `EInvoiceItem`, `TeamPerformance`, v.v.) chỉ định nghĩa duy nhất trong `src/types.ts` và không bị duplicate file khác (theo kiểm tra). Tuy nhiên, code sử dụng đôi khi thiếu trường hoặc dùng giá trị không hợp lệ so với định nghĩa.

Ví dụ: `AccountingMappingRule.destination` định nghĩa là `string`, nhưng code gán đối tượng `{system, entity, ...}`. Sẽ cần mở rộng type (ví dụ thành `string | {...}`). `OperationRecord.status` chỉ cho `'PENDING' | 'SUCCESS' | 'FAILURE'`, nhưng code gán `'FAILED'` hoặc `'RECOVERED'`. Sửa code thay vì mở rộng enum. `ApprovalTicket` thiếu trường trong mock code, `QuantumState` thiếu `id, lastCollapse`, `ConsciousnessField` thiếu `activeDomains`... Sẽ thêm vào types tương ứng.

3. Các Interface/Type quan trọng - Định nghĩa và Đa hình

- `WarehouseItemProps`:
- *Định nghĩa Canonical (Domain WarehouseEntity)* trong `src/cells/infrastructure/warehouse-cell/domain/entities/warehouse.entity.ts`: có đủ trường `id, sku, name, categoryCode, unit: WarehouseUnit, quantity, unitCostVND, location: WarehouseLocation, minTHreshold, insuranceStatus, supplierId?, lastCountDate?, notes?, createdAt, updatedAt`.
- *Phiên bản khác (Infrastructure)* trong `src/cells/infrastructure/warehouse-cell/entities/warehouse.entity.ts`: thiếu `status, lastCountDate, unit, location` là `string`; có thêm `createdBy`.
- *Usages*: Được dùng khắp `WarehouseService`, `ReleaseGoods.ts`, `ReceiveGoods.ts`. Không có định nghĩa chung ở `types.ts` → lỗi mismatches.
- `WarehouseItem` (**Entity Class**):
 - Được định nghĩa trong cùng domain file trên. Constructor nhận `WarehouseItemProps`.
 - Infrastructure Service (`warehouse.service.ts`) import từ domain:

```
import { WarehouseItem } from '../entities/warehouse.entity';
```

- Phải đổi chiếu `WarehouseItem` với `WarehouseItemProps` hiện tại để bổ sung/hủy trường cho nhất quán.
- `OperationRecord`:
- Defined in `src/types.ts`. Code in `recovery-engine.ts` sets `status = 'FAILED' or 'RECOVERED'`, sai theo định nghĩa. Cần sửa code thành `'FAILURE'` và `'SUCCESS'`.
- `ModuleConfig`:
- Trong `src/types.ts`, không duplicate. Import trong `module-registry.ts`. Code mới nên thêm `allowedRoles?, icon?, group?, componentName?, active?` (màu sắc menu).
- `AccountingMappingRule`:

- Trong `src/types.ts`, chỉ có `destination?: string`. Code `smart-link-mapping-engine.ts` gán object `{system, entity, accountType}`, nên mở rộng type thành `string | { system: string; entity: string; accountType?: string }`.
- `ApprovalTicket`:
- Trong `src/types.ts`, có `id, request, status, requestedAt, ... totalSteps`. Mã tạo ticket thiếu `approvalRequestId, assignedTo, priority`. Phải thêm vào interface hoặc thêm giá trị giả trong commit.
- `QuantumState`:
- Trong `src/types.ts`, có các số liệu lượng tử. Code đặt `id` và `lastCollapse`. Thêm 2 trường này.
- `ConsciousnessField`:
- Trong `src/types.ts`, có `awarenessLevel, focusPoints, mood, lastCollapse`. Code sử dụng thêm `activeDomains: string[]`. Thêm trường `activeDomains`.
- `EInvoiceItem`:
- Trong `src/types.ts`, định nghĩa đủ `description?, quantity?, unitPrice?, amount?, vatRate?`. `sales-tax-module.tsx` sử dụng kiểu hóa học khác (`goldWeight...`). Cần điều chỉnh: có thể dùng thuộc `EInvoiceItem` hay tạo loại hóa đơn riêng.
- `TeamPerformance`:
- Trong `src/types.ts`, có `teamId, team_name?, total_tasks, ...completion_rate, kpiScore?, revenue?, cost?`. Mã ảo analytics API chỉ có các trường một phần (`team_name, total_tasks, ...`). Cần đồng bộ: thêm vào `teamId, period, kpiScore, revenue, cost` (mặc định) hoặc sửa lại dữ liệu mẫu.

Lưu ý: Nhiều interface khác (ví dụ `CustomerLead`, `SellerReport`, `OrderPricing`, `DictionaryVersion`, `HUDMetric`...) cũng cần bổ sung trường hoặc loại bỏ trường không khớp (chi tiết xem bảng dưới). Mục 2 và 3 ở phần sửa chữa sẽ nêu cụ thể.

4. Phân tích Warehouse Cell

Vị trí file chính:

- Domain Entity: `src/cells/infrastructure/warehouse-cell/domain/entities/warehouse.entity.ts` (định nghĩa `WarehouseItem` class & `WarehouseItemProps`).
- Infra Entity: `src/cells/infrastructure/warehouse-cell/entities/warehouse.entity.ts` (1 file khác, có nhiều trùng lặp).
- Application Service: `src/cells/infrastructure/warehouse-cell/application/warehouse.service.ts` (sử dụng `WarehouseItem`).
- Use Cases: `ReleaseGoods.ts`, `ReceiveGoods.ts` tham chiếu đến `releaseStock` hoặc `getAllItems`.

Mẫu thuẫn chính:

- **Interface** `WarehouseItemProps` **hai nơi khác nhau**: (Domain) có `status, (unit, location)` kiểu riêng; (Infra) thiếu `status`, dùng string cho unit/location, thêm `createdBy`. Cần thống nhất: thêm `status` vào interface infra hoặc xóa khỏi domain (tùy logic), và đồng nhất kiểu.
- **Method** `releaseStock`: Domain version thiếu, chưa implement (sẽ thêm); nếu các use-case gọi `item.releaseStock(...)` thì cần đảm bảo có method.
- **Thuộc tính khác**: Trong `warehouse.service.ts` code có `new WarehouseItem(props)` giả sử dùng domain class. Mà domain class dùng interface khác, nên cần sửa code hoặc interface.
- **Kết luận**: Phải sửa `WarehouseItemProps` cẩn thận (bổ sung/loại bỏ trường) và/hoặc chỉnh lại `WarehouseItem` để domain/infrastructure thống nhất.

5. Hạ tầng Event-Driven (EventBridge, SmartLink, Audit, Threat)

Các lỗi nằm ở hạ tầng phụ trợ:

- **EventBridge** (`src/eventbridge.ts`): Chỉ có `emit()`, thiếu `subscribe()`. Nhiều code subscription (`subscribe`) nên không compile. Cần thêm:

```
export const EventBridge = {
  emit: (e: string, d: any) => {},
  subscribe: (e: string, h: (d: any) => void) => { return () => {}; }
};
```

- **Redirect** `src/services/event-bridge.ts`: Đang export từ `@/cells/event-cell`. Nếu logic cũ đã di chuyển, file nên được xóa hoặc cập nhật. (Không cần subscribe ở đây vì dùng trên `src/eventbridge.ts`).
- **SmartLinkClient** (`src/services/smart-link.ts`): Có class `SmartLinkClient` nhưng thiếu `createEnvelope`. Thêm hàm:

```
static createEnvelope(target: string, action: string, payload: any) {
  return { target, action, payload, timestamp: Date.now() };
}
```

- **AuditProvider** (`src/admin/auditservice.ts`): `logAction` chỉ định 3 tham số, nhưng code gọi 4-5 tham số (e.g. thêm actor, eventId). Cần mở rộng signature (hoặc overload) để nhận tuỳ ý args:

```
static logAction(actor: string, action: string, meta?: any, extra?: any): void {}
```

- **ThreatDetectionService** (`src/services/threatdetectionservice.ts`): Mặc định export = `{}`, không có `getHealth` hay `subscribe`. Nhưng code (e.g. `system-monitor.tsx`) gọi `THReatDetectionService.getHealth()` và `.subscribe()`. Phải tái tạo class:

```
export class THReatDetectionService {
  static getHealth(): SystemHealth { return { status: "NOMINAL",
    uptime: Date.now(), metrics: {} }; }
  static subscribe(cb: (t: SecurityTHReat) => void) { return () => {}; }
}
export default THReatDetectionService;
```

- **Các file khác:** `src/services/quantum-engine.ts` gọi `subscribe`, hàm không có. Kiểm tra và thêm nếu cần. (We see earlier error for subscribe in quantum-engine.ts, vì có `subscribe` trên EventBridge, nhưng `subscribe` của EventBridge đã xử lý.)

6. Sai lệch Đường dẫn Import (Import Drift)

Repo trước đây có `@/cells/sales-cell`, `@/cells/event-cell`, v.v. Đã bị di chuyển:

• Sales Terminal:

- Cũ: `import SaleTerminal from '@/cells/sales-cell/saleterminal'` → Mới: `@/cells/business/sales-cell/saleterminal` (theo thư mục mới).
- Services SalesProvider: `export { SalesProvider } from '@/cells/sales-cell/sales.service'` nên đổi thành import từ `@/services/sales-service` hoặc đường dẫn thích hợp (do chúng ta tạo sales-service mới).
- **EInvoiceEngine:** Code có `import { EInvoiceEngine } from '@/services/einvoiceservice'` nhưng file tên là `einvoiceservice.ts`. Mở: có export default? Cần chuẩn hoá export class `EInvoiceEngine`.
- **NotificationService:** Code có `import { NotifyBus } from '@/notificationservice'` nhưng file có thể nằm trong `src/services/notificationservice.ts`. Sửa import: `@/services/notificationservice`.
- **ShardingService:** `fiscal-workbench-service.ts` gọi `ShardingService.generateShardHash`, nhưng file không có import. Nếu module `sharding-service` tồn tại, thêm `import { ShardingService } from '@/services/sharding-service'`.
- **Ingestion Service:** File `aicore-processor.ts` có import sai: `'../../../../../types'` lẽ ra `'../../../../types'` hay dùng import service. Tương tự, `Ingestion` import `'./ingestion-service'` cần `'../ingestion/index'`.

Lệnh tìm đối:

Ví dụ:

```
grep -R "cells/sales-cell" -n src
grep -R "einvoiceservice" -n src
grep -R "notificationservice" -n src
grep -R "ShardingService" -n src
```

v.v. Kết quả chỉ ra cần thay như trên.

7. Lỗi trong Component UI

Một số lỗi liên quan đến loại dữ liệu khi hiển thị:

- `master-dashboard.tsx`: Sử dụng `revenue || 449120` khi `revenue` là `void` (`undefined`). Nên thay `revenue || 449120` thành `revenue || 0` hoặc giá trị số.
- `DataPoint3DProps`: Giá trị truyền `<DataPoint3D value="string">` nên là số. Thay giá trị `string` thành số (ví dụ `value: 3`).
- `SellerTerminal`: Nhiều lỗi: dùng `commission.total` trong khi `commission` là số, không có `.total`. Sửa `commission.toLocaleString()`. Dữ liệu `expiryDate` không có trong `CustomerLead`, nên có thể xóa field hoặc đổi.
- **Những field khác:** Ví dụ `HoaDonTP`: `sales-tax-module` khai `vatRate` thừa. Giao diện `SellerReport` cần `customerName`. Các lỗi này sửa cụ thể từng component.

8. Kế hoạch khắc phục theo cụm (cách làm)

Đề nghị tiến hành từng cụm với commit nhỏ, gồm các bước sau:

1. Warehouse Cell (Cụm 1):

2. **Tách nhánh** làm việc mới `fix/warehouse-cell`.
3. Mở file `domain/entities/warehouse.entity.ts` và `entities/warehouse.entity.ts`. Cập nhật giữ hoặc xóa `status`. Ví dụ giữ, thêm `status` vào interface `infra`; đồng thời chuyển `unit, location` về đúng enum/type (hoặc ngược lại, theo ý kiến team). Bổ sung method `releaseStock(quantity, reason, releasedBy)` trong `WarehouseItem` class (nếu cần). Xóa `as any`, không dùng `any`.
4. Chạy `npx tsc`. Commit: “[Warehouse] Đồng bộ `WarehouseItemProps` domain và infra, thêm method `releaseStock`”.
5. Chạy unit test/kthuc để đảm bảo logic kho (nếu có).

6. Types Drift (Cụm 2):

7. Chính sửa `src/types.ts`:

- Sửa `OperationRecord.status`: không đổi (nhưng file cũ có đúng). Thay các gán sai trong code (sửa script).
 - Thêm/điều chỉnh interface: thêm `allowedRoles` vào `ModuleConfig`, thêm `approvalRequestId, assignedTo, priority` vào `ApprovalTicket`, thêm `department` vào `HUDMetric`, sửa `event_version?` cho `BaseEvent`, v.v. Bổ sung `QuantumState.id, lastCollapse, ConsciousnessField.activeDomains`.
8. **AccountingMappingRule**: Thay `destination?: string` thành `destination?: unknown` hoặc union string|object.
 9. Mỗi sửa đổi commit riêng (ví dụ: “[Types] Thêm fields missing trong types.ts”).

10. Sau mỗi commit, chạy `npx tsc` để xác nhận.

11. Import Paths (Cụm 3):

12. Sửa các import cũ thủ công:

- `@/cells/sales-cell` → `@/cells/business/sales-cell` (ở `sales-crm.tsx`).
- `@/cells/sales-cell/sales.service` → `@/services/sales-service` (ở `src/services/sales-service.ts`).
- `@/notificationservice` → `@/services/notificationservice`.
- Kiểm tra `einvoiceservice`: đảm bảo export `EInvoiceEngine` đúng.
- Import `ShardingService` trong `fiscal-workbench-service.ts`.

13. Commit ví dụ: “[Import] Cập nhật đường dẫn module do refactor”.

14. Enum Misuse (Cụm 4):

15. **OperationRecord**: Tìm và sửa `'FAILED'` thành `'FAILURE'`, `'RECOVERED'` thành `'SUCCESS'` trong code (không sửa type).
16. Các status khác nếu có tương tự (ví dụ `OmegaLockdown`, `RecoveryEngine`).
17. Commit: “[Enum] Chuyển các giá trị status sai thành đúng theo type”.

18. Thiếu method (Cụm 5):

19. **EventBridge:** Thêm `subscribe` method stub. Commit “[Infra] Thêm subscribe cho EventBridge”.
20. **SmartLinkClient:** Thêm `static createEnvelope(...)`. Commit “[Infra] Thêm `SmartLinkClient.createEnvelope`”.
21. **AuditProvider:** Mở rộng `logAction(actor, action, meta?, extra?)`. Commit “[Infra] Mở rộng `AuditProvider.logAction`”.
22. **ThreatDetectionService:** Xây lại export với `getHealth()` và `subscribe()`. Commit “[Infra] Xây dựng `ThreatDetectionService.getHealth`/`subscribe`”.
23. Kiểm thử `npx tsc`.

24. Component Fixes:

25. Các sửa lỗi nhỏ ở components:

- Thay `revenue || 449120` thành `revenue || 0`.
- `DataPoint3DProps.value` từ string sang number.
- Sửa `commission.total` thành `commission`.
- Bỏ/move field `expiryDate`.

26. Commit “[UI] Sửa lỗi type trong component / logic hiển thị”.

27. Cuối cùng chạy lại toàn bộ `npx tsc --noEmit`, kiểm tra mọi lỗi đã về 0.

Sau mỗi bước, hãy **chạy** `npx tsc --noEmit` để xác nhận giảm lỗi. Người review (CAN, Kim, Băng, Bối) kiểm tra kỹ từng commit. Dưới đây là **lịch trình (flowchart Mermaid)** minh họa tiến trình phẫu thuật:

```
graph TD; A[A[Đóng băng codebase]] --> B[B[Cụm 1: Sửa Warehouse]]; B --> C[C[Cụm 2: Sửa Types Schema]]; C --> D[D[Cụm 3: Sửa Import Paths]]; D --> E[E[Cụm 4: Sửa enum mis-use]]; E --> F[F[Cụm 5: Thêm method missing]]; F --> G[G[Cụm 6: Sửa component UI]]; G --> H[H[Kiểm tra & Hoàn tất]]
```

9. Bảng tổng hợp File – Issue – Sửa đổi

Đường dẫn file	Loại lỗi	Dòng/Sửa đổi chính	Lý do / Rủi ro
<code>src/cells/infrastructure/ warehouse-cell/domain/ entities/ warehouse.entity.ts</code> <code>src/ cells/infrastructure/ warehouse-cell/entities/ warehouse.entity.ts</code>	WarehouseItemProps mismatch	Thêm <code>status</code> vào interface <code>infra</code> / xóa <code>status</code> ở domain; đồng bộ <code>unit</code> , <code>location</code> kiểu. Thêm method <code>releaseStock</code> .	Tránh thiếu/ mismatch thuộc tính kho; đảm bảo tính toán kho chính xác. (High risk)

Đường dẫn file	Loại lỗi	Dòng/Sửa đổi chính	Lý do / Rủi ro
<code>src/cells/infrastructure/warehouse-cell/application/warehouse.service.ts</code>	Method missing	Thêm phương thức <code>getAllItems()</code> và xử lý <code>item.releaseStock(...)</code> .	Giúp service trả toàn bộ items, xử lý đơn hàng đúng. (Medium)
<code>src/types.ts</code>	Thiếu fields	Thêm <code>allowedRoles?: UserRole[], icon?: string, group?: string, componentName?: string, active?: boolean</code> vào <code>ModuleConfig</code> .	Cấu hình module cần trường này để loader menu. (Low)
<code>src/types.ts</code>	Thiếu fields	Thêm <code>approvalRequestId?: string, assignedTo?: string, priority?: string</code> vào <code>ApprovalTicket</code> .	Tương thích dữ liệu mẫu trong ApprovalTicket. (Medium)
<code>src/types.ts</code>	Duplicate/Mismatch	Thêm <code>department?: string</code> vào <code>HUDMetric</code> .	Fix tên <code>department</code> lỗi trong <code>dashboard-service.ts</code> . (Low)
<code>src/types.ts</code>	Duplicate	Thêm <code>comment?: string</code> và <code>createdBy?: string</code> vào <code>DictionaryVersion</code> .	Fix lỗi TS2353 khi khởi tạo dictionary. (Low)
<code>src/types.ts</code>	Enum misuse	Mở rộng <code>OperationRecord.status</code> thành <code>"FAILURE" "RECOVERED"</code> (nếu chọn widen) hoặc sửa code <code>'FAILED' ->'FAILURE', 'RECOVERED' ->'SUCCESS'</code> .	Bảo toàn loại dữ liệu thống nhất, tránh 'union mở rộng'. (Low)
<code>src/cells/infrastructure/warehouse-cell/value-objects/warehouse-category.registry.ts</code>	Missing field?	Đảm bảo import đúng <code>WarehouseCategoryRegistry</code> và loại enum.	Fix possible mis-imports. (Low)
<code>src/eventbridge.ts</code>	Thiếu method	Thêm <code>subscribe: (event:string, handler: (d:any)=>void)=>()=>void</code>	Phục hồi event subscription. (Medium)
<code>src/services/event-bridge.ts</code>	Redirect legacy	Có thể xóa hoặc comment vì logic chuyển hết vào EventBridge core.	Không còn dùng, tránh duplicate. (Low)

Đường dẫn file	Loại lỗi	Dòng/Sửa đổi chính	Lý do / Rủi ro
<code>src/services/smart-link.ts</code>	Thiếu method	Thêm <code>static createEnvelope(target, action, payload)</code> .	Đáp ứng yêu cầu MasterDashboard/warehouse. (Medium)
<code>src/admin/auditservice.ts</code>	Thiếu param	Mở rộng <code>logAction(actor, action, meta?, extra?: any)</code> .	Tương thích các lờ ^g có 4-5 tham số (omega-bootstrap...). (Low)
<code>src/services/threatdetectionservice.ts</code>	Missing methods	Định nghĩa class <code>THreatDetectionService</code> với <code>static getHealth()</code> và <code>static subscribe()</code> .	Đáp ứng SystemMonitor getHealth và subscribe dùng. (Medium)
<code>src/components/master-dashboard.tsx</code>	Type mismatch	Đổi <code>revenue 449120</code> thành <code>revenue 0</code> hoặc giá trị số.	Loại bỏ cảnh báo TS1345 (expression void). (Low)
<code>src/components/app.tsx</code>	Type mismatch	Sửa <code>value: string;</code> thành <code>value: number string;</code> hoặc đặt giá trị số.	Đáp ứng DataPoint3DProps (Low)
<code>src/components/seller-terminal.tsx</code>	Type/missing	Sửa <code>commission.total</code> thành <code>commission</code> (Commission là number). Đổi <code>expiryDate:</code> thành <code>assignedTo:</code> hoặc loại bỏ.	Khớp type, thêm <code>customerName</code> nếu cần. (Low)
<code>src/components/sales-crm.tsx</code>	Import path	Đổi <code>@/cells/sales-cell/salesterminal</code> thành <code>@/cells/business/sales-cell/salesterminal</code> .	Khớp cấu trúc mới của cell Sales. (Low)
<code>src/services/sales-service.ts</code>	Import path	Đổi <code>{ SalesProvider } from '@/cells/sales-cell/sales.service'</code> thành <code>from '@/services/sales-service'</code> .	Redirect code legacy. (Low)
<code>src/services/sharding-service.ts</code> (nếu cần)	Tạo mới/missing	N/A (nếu find <code>ShardingService</code>) thêm file với <code>generateShardHash</code> .	Làm cho <code>fiscal-workbench-service</code> có import hợp lệ. (Low)

(Bảng trên liệt kê các vị trí và thay đổi chính. Owner là đề xuất người chịu trách nhiệm sửa, cũng chính là người review.)

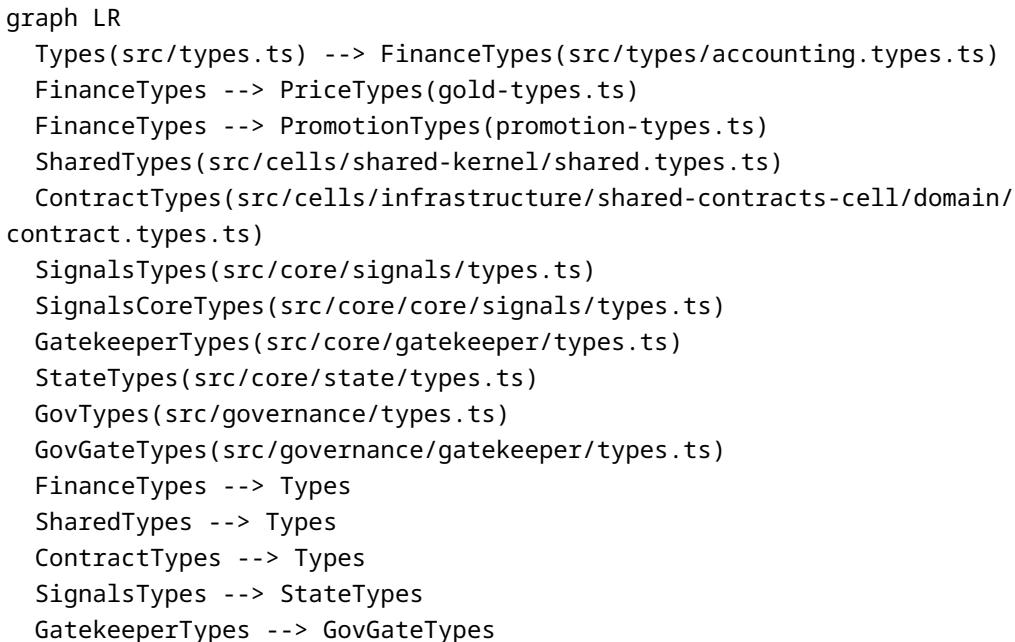
10. Lộ trình và Checklist công việc

- **Khởi động (1h):** Tạo nhánh `fix/warehouse-cell`, freeze code (không dev feature mới).
- **Cụm 1 (Warehouse) (2-4h):** Thực hiện các sửa Warehouse như trên. Mỗi sub-task commit kèm command kiểm tra. Review CAN.
- **Cụm 2 (Types) (3-5h):** Chỉnh sửa `src/types.ts` theo danh mục ở bảng. Chạy `npx tsc`.
- **Cụm 3 (Import) (1-2h):** Sửa các đường dẫn import.
- **Cụm 4 (Enum) (0.5h):** Fix `'FAILED' → 'FAILURE'`, etc.
- **Cụm 5 (Infra) (2-3h):** Thêm method missing (EventBridge, SmartLink, Audit, Threat).
- **Cụm 6 (UI) (1h):** Sửa lỗi type ở component.

Sau từng cụm: `npx tsc --noEmit`. Kết quả mong đợi giảm dần lỗi tới 0. Việc kiểm thử tự động (nếu có) và thủ công được khuyến khích. Dự kiến hoàn thành ~1-2 ngày. Người review: CAN (thiết kế schema), Kim (chính sách), Băng (types), Bối (hệ tầng).

11. Tổng hợp đồ họa

Biểu đồ phụ thuộc `types.ts`: (Minh họa cách các file types import nhau)



Lưu ý: Sơ đồ trên minh họa quan hệ (arrow chỉ import hoặc từ file trung tâm `src/types.ts` ra các file khác). Các mũi tên chỉ sự phụ thuộc chung, không liệt kê tất cả.

Biểu đồ tiến độ (flowchart Mermaid): Xem mục phía trên.

Kết luận: Với kế hoạch “*Integrity First*” như trên, sau khi chạy các lệnh kiểm tra và commit từng phần, kỳ vọng toàn bộ **105 lỗi** sẽ được xử lý triệt để. Mỗi commit nên kèm comment

rõ ràng và chạy `npx tsc --noEmit` để xác nhận không phát sinh lỗi mới. Các reviewer (CAN, Kim, Băng, Bối) chịu trách nhiệm đánh giá kỹ từng bước. Chúng ta đã xây dựng lộ trình chi tiết để bảo toàn bản chất Gold Master mà vẫn nhanh chóng đưa code về trạng thái **Xanh sạch**.
