# Fractal Silk Route Hub — Агентная архитектура и первый модуль (TerraPedia)

**Версия:** 0.9 (рабочая) **Формат:** Фрактальный модуль / TerraPedia **Языки интерфейса:** RU (основной), EN/DE/UZ/CN/AR/FA/ES/PT — подготовлены метаданные для перевода TerraLex

# Краткое описание

Документ содержит детализированную структуру первого крупного рабочего блока — **структурной модели агентов (фрактальных модулей)** и их взаимодействий для проекта *Fractal Silk Route Hub*. Материал оформлен в формате TerraPedia: иерархичный, рекурсивный, готовый для трансформации в интерактивный модуль портала.

# Содержание

1. Цели и принципы
2. Определения и номенклатура (L0–L7)
3. Описание агентов: роли и API
4. PLT-протокол: concept pulses, trace vectors, inject hashes, silence blocks
5. Информационная модель (схема данных) — сущности и связи (ER-диаграмма)
6. Кластеризация и определение центров фрактальной связанности
7. Критерии выбора площадок-хабов в Узбекистане (рекомендации)
8. Алгоритмы маршрутизации и распределения ресурсов (псевдокод)
9. Интерфейсы TerraPedia и UI-компоненты (Terra Design System)
10. План этапов реализации и контрольные маяки
11. Список артефактов для следующей итерации

## 1. Цели и принципы

* Построить автономную, самоподобную сетевую систему агентов, обеспечивающую управление грузоперевозками по коридору США—Европа—ЦА.
* Соблюдать принципы FMP / NULLO: фрактальная организация данных, голографическая связность, поддержка PLT и TerraLex.
* Обеспечить многоязычность и проверяемость данных (inject hashes) на всех уровнях.

## 2. Определения и номенклатура (L0–L7)

* **L0** — семантическое ядро (ТН ВЭД-элемент, смысловой прототип товара)
* **L1** — единица груза / упаковочная единица
* **L2** — локальный поставщик / склад / транспортное средство
* **L3** — локальный хаб / терминал
* **L4** — региональный хаб (например, Самарканд, Навоий)
* **L5** — национальный узел (Ташкент, транспортно-логистическая сеть)
* **L6** — транзитный коридор (Европа–ЦА, маршруты через КЗ/РФ/КНР)
* **L7** — глобальные интеграции и политико-экономические слои

Каждый уровень содержит метаданные для перевода и PLT-ссылки на родительские/дочерние элементы.

## 3. Описание агентов: роли и API

### 3.1. Обозреватель (Observer)

* Задачи: сбор телеметрии, проверка статусов, входная валидация данных.
* Input: сенсорные данные, EDI/JSON от партнёров, таможенные бирки.
* Output: верифицированные события, уведомления о нарушениях.
* API: GET /observer/status?agent\_id=, POST /observer/event.

### 3.2. Анализатор (Analyzer)

* Задачи: прогнозирование спроса, оптимизация маршрутов, симуляции сценариев.
* Инструменты: ML-модули (прогнозирование объёмов), симулятор перколяции фрактальных кластеров.
* API: POST /analyzer/plan, GET /analyzer/forecast?corridor=.

### 3.3. Резольвер (Resolver)

* Задачи: согласование целей между агентами, разрешение конфликтов, арбитраж.
* Механизм: протокол переговоров (auction / consensus / negotiated replan).
* API: POST /resolver/propose, POST /resolver/resolve.

### 3.4. Организатор (Organizer)

* Задачи: назначение ресурсов, создание фракталов, масштабирование агентов.
* API: POST /organizer/deploy, DELETE /organizer/retire.

### 3.5. Репортер (Reporter)

* Задачи: формирование отчётов, агрегирование KPI, экспорт в стандарты (UN/CEFACT, ISO).
* API: GET /reporter/kpi?period=, GET /reporter/export?format=pdf.

*Каждый агент имеет цифровой аватар — сущность в TerraPedia с уникальным inject hash.*

## 4. PLT-протокол (технический каркас)

* **Concept Pulse (ψ)** — уникальный смысловой вектор для сущности (например, HS-код + семантические признаки). Хранится в каноническом виде.
* **Trace Vector (v)** — временная траектория состояния агента (логи, события, координаты).
* **Inject Hash (h\_inject)** — SHA-256( canonical\_representation || v ) — защищает целостность версии знания.
* **Silence Block (□)** — мета-плейсхолдер для отрицательных/отсутствующих данных: используется как семантический маркер «пре-условия отсутствуют». Нужен для PLT-валидности.

PLT-операция: при обновлении сущности система генерирует новое h\_inject и публикует событие PLT/UPDATE в TerraPedia с подписанным верификатором.

## 5. Информационная модель (ER-схема) — сущности и связи

Ключевые сущности:

* Goods (L0–L1) {hs\_code, description, weight, volume, value, ψ}
* Shipment {shipment\_id, origin, destination, goods[], current\_agent, trace\_vector v}
* Agent {agent\_id, type, location, status, h\_inject}
* Hub {hub\_id, geo, capacity, services}
* Route {route\_id, legs[], cost\_model, time\_windows}
* Event {event\_id, shipment\_id, timestamp, type, payload}

Связи: Goods ↔ Shipment ↔ Route ↔ Hub ↔ Agent. Каждая связь имеет PLT-метаданные.

## 6. Кластеризация и определение центров фрактальной связанности

Методика:

1. Собрать тепловую матрицу перемещений M\_{i,j} по узлам в течение T периодов.
2. Вычислить индекс плотности R\_j = Σ\_i M\_{i,j} \* w\_i (веса — экономическая ценность, срочность).
3. Применить алгоритм DBSCAN / OPTICS в геопространственном + товарном пространстве с метрикой, смешивающей расстояние и семантическое сходство (cosine(ψ\_i, ψ\_j)).
4. Центры с наибольшим R\_j — кандидаты на региональные хабы; затем проводится экономическая оценка ROI.

## 7. Критерии выбора площадок-хабов в Узбекистане (рекомендации)

Критерии ранжирования:

* Транспортная доступность: близость к ж/д узлам, автомагистралям, аэропортам
* Пропускная способность существующей инфраструктуры
* Таможенные режимы и совместимость процессов
* Стоимость земли и возможностей для строительства хабов
* Наличие промышленного парка и кадрового резерва
* Экологические и социальные ограничения

Рекомендуемые стартовые площадки (приоритет):

1. Навоий (индустриальный потенциал, близость к транзитным ж/д направлениям)
2. Самарканд (географическая центровка, туристический/логистический потенциал)
3. Ташкент (административный центр, крупные склады и аэропорт)
4. Андижан / Фергана — для восточного коридора

Для каждой площадки нужно подготовить ТЭО с расчётом пропускной способности и ROI.

## 8. Алгоритмы маршрутизации и распределения ресурсов (псевдокод)

### 8.1. Функция быстрой перенацелировки (fastReplan)

function fastReplan(shipment S):  
 candidates = getAvailableRoutes(S.origin, S.destination)  
 for r in candidates:  
 score[r] = costModel(r, S) \* riskFactor(r) \* fractalAffinity(S.goods, r)  
 choose r\* with min(score)  
 proposeResolution = Resolver.propose(S, r\*)  
 if Resolver.resolve(proposeResolution) == OK:  
 Organizer.deploy(S, r\*)  
 else:  
 escalateToAnalyzer(S)

fractalAffinity — измеряет согласие товара и маршрута по ψ-векторам (cosine similarity).

## 9. Интерфейсы TerraPedia и UI-компоненты

* Главная панель: фрактальный навигатор (вверх/вниз по уровням L0–L7)
* Карта: слой маршрутов, хабов, активных шипментов с возможностью 3D-режима
* Агрегированные KPI: throughput, dwell time, customs delays, ROI
* Компонент «Аватар агента» — карточка с PLA-подписью и историей trace vectors

Дизайн-элементы — согласно Terra Design System (цвета, кнопки, карточки, анимации).

## 10. План этапов реализации и контрольные маяки

**Шаг 0 (2 недели):** Подготовка TerraPedia-ядра, шаблоны сущностей, PLT-инструментарий

**Шаг 1 (6 недель):** Реализация базовых агентов (Observer/Reporter), загрузка пилотных данных

**Шаг 2 (8 недель):** Разработка Analyzer/Resolver/Organizer, простая маршрутизация и fastReplan

**Шаг 3 (6 недель):** Интеграция картографии, 3D визуалов и TerraLex переводчика (базовый словарь)

**Шаг 4 (12 недель):** Пилот в Узбекистане: Навоий + Ташкент — тестирование потоков, KPI, TCO/ROI

**Шаг 5:** Масштабирование на регион (КЗ, Таджикистан) и интеграция с внешними API (Customs, UNCTAD)

## 11. Список артефактов для следующей итерации

* ER-диаграмма в виде файла .svg/.drawio
* Прототип UI (Figma) главной панели и карты
* Набор тестовых данных (фейковых, но реалистичных) для пилота
* Скрипты кластеризации и расчёта R\_j (Python)
* Шаблон ТЭО для хаба (Навоий)

# Заключение

Этот документ — готовый рабочий модуль, который можно загрузить в TerraPedia и использовать как основу для разработки прототипа платформы. Следующие шаги — генерация артефактов (ER-диаграмма, UI, тестовые данные) и развёртывание пилота.

*Подготовил: Fractal Silk Route Hub — TerraPedia Team (ассистент).*