

# AuraPass — Data Flow (End-to-End)

Berikut adalah alur lengkap bagaimana data bergerak di dalam sistem AuraPass, mulai dari aktivitas on-chain pengguna hingga menjadi skor kredit, ZK-proof, dan oracle yang dapat dibaca protokol DeFi.

---

## 1. User On-chain Activity

Wallet melakukan aktivitas on-chain: - Transfer token - Interaksi DeFi (swap, lend, borrow) - Minting NFT - Bridging - Staking / farming

**Output:** Raw blockchain events (transaction, logs, receipts).

---

## 2. Indexer Layer

**Komponen:**

- **The Graph Subgraph:** mengindeks event standar.
- **Custom Indexer (WebSocket RPC):** menangkap event real-time.

**Flow:** Blockchain → Websocket RPC → Parser → Event JSON

**Output:** Event terstruktur (**normalized raw event**).

---

## 3. Event Stream / Message Queue

Event dikirim ke message broker: - Kafka / Redis Streams

**Tujuan:** - Menjamin urutan event - Anti-lost messages - Membuat fan-out ke beberapa worker

**Output:** Event stream siap diproses.

---

## 4. ETL (Extract → Transform → Load)

Worker memproses event menjadi features.

**Transformasi meliputi:**

- Hitung tx\_count
- Hitung inflow/outflow
- Hitung stablecoin ratio
- Identifikasi protokol yang digunakan
- Normalisasi token value (USD equivalent)
- Identifikasi pola perilaku (time-between-tx, cluster interaction)

**Output:** Feature Vector → disimpan di Feature Store.

---

## 5. Feature Store & Time-series DB

Sumber utama untuk ML & heuristic scoring.

Menyimpan: - snapshot fitur - time-window features (1 hari, 7 hari, 30 hari) - historical trends

**Output:** Data siap digunakan Scoring Engine.

---

## 6. Reputation Engine / Credit Scoring

Dua engine berjalan paralel:

### 6.1 Heuristic Scoring

- Rule-based
- Deterministic

### 6.2 ML Model

- Gradient boosting model (XGBoost)
- Mengambil feature vector → output probabilitas & skor

**Output:** - score (0–1000) - tier (A–F) - tags ("stable saver", "risky", "active user") - confidence value

Skor ini dikirim ke DB dan ke ZK Service.

---

## 7. ZK-Proof Generation Layer

Menerima: - score - score\_hash - nullifier

**Proses:**

- Buat commitment (hash)
- Generate ZK proof (Circos/Noir)
- Simpan proof artifact

**Output:** - proof.json - publicSignals.json - commitment hash

---

## 8. Credit Passport Smart Contract (SBT)

Menerima hash hasil scoring: - score\_hash - last\_update - metadata

Kontrak menyimpan:

```
wallet → score_hash  
wallet → nullifier
```

**Output:** On-chain canonical identity & credit object.

---

## 9. API Gateway

Backend menyediakan: - GET score - GET history - GET proof - POST verify

**Output:** REST API untuk integrator & dashboard.

---

## 10. Oracle Adapter (On-chain Consumer)

Protokol DeFi memanggil: - Verifier.sol untuk cek proof - ScoreOracle.sol untuk cek score\_hash - Off-chain API untuk full score

**Output:** Protokol dapat mengambil keputusan (approve loan / set collateral).

---

## 11. Dashboard (Front-end)

Menampilkan: - Credit score - History - ZK-proof status - On-chain identity badges

---

## SUMMARY FLOW (Ringkas)

```
User On-chain Activity  
↓  
Indexer (Subgraph + Custom)  
↓  
Event Queue (Kafka)  
↓  
ETL / Feature Extraction  
↓  
Feature Store (ClickHouse)  
↓  
Scoring Engine (Heuristic + ML)  
↓  
ZK-Proof Generation (Circom/Noir)  
↓  
ScoreOracle + CreditPassport (Smart Contract)  
↓  
API Gateway / Dashboard
```

↓  
DeFi Protocol Integration (Oracle)

---

Jika kamu mau, saya bisa lanjutkan dengan **diagram visual (SVG/PNG)** untuk data flow ini atau membuat **sequence diagram** untuk 1 user flow lengkap (misal: peminjaman DeFi).