

**API AVENGERS 1.0**

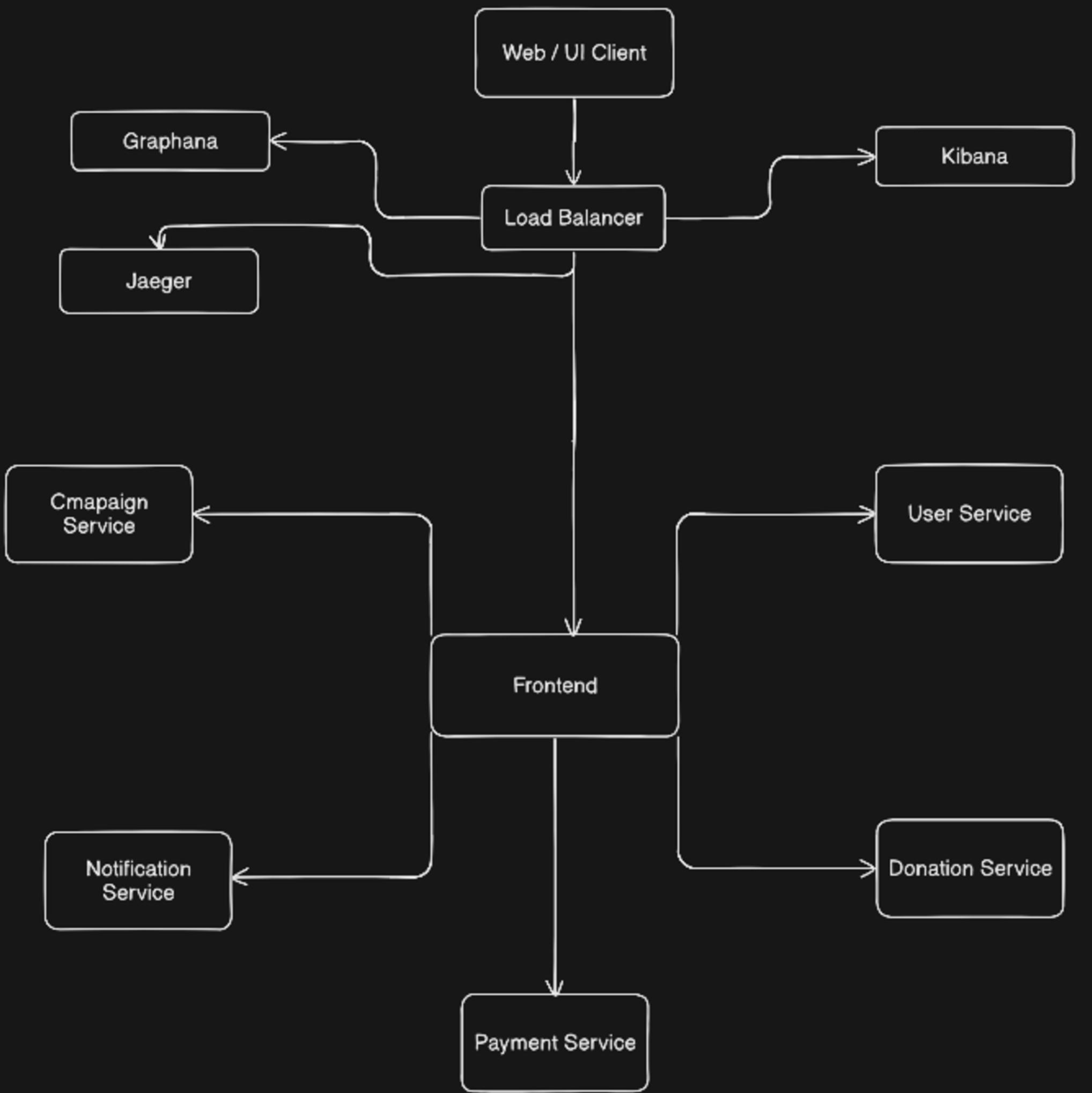
# **Care For All: Microservices Solution**

**Presented by:**

**Team BUGS**

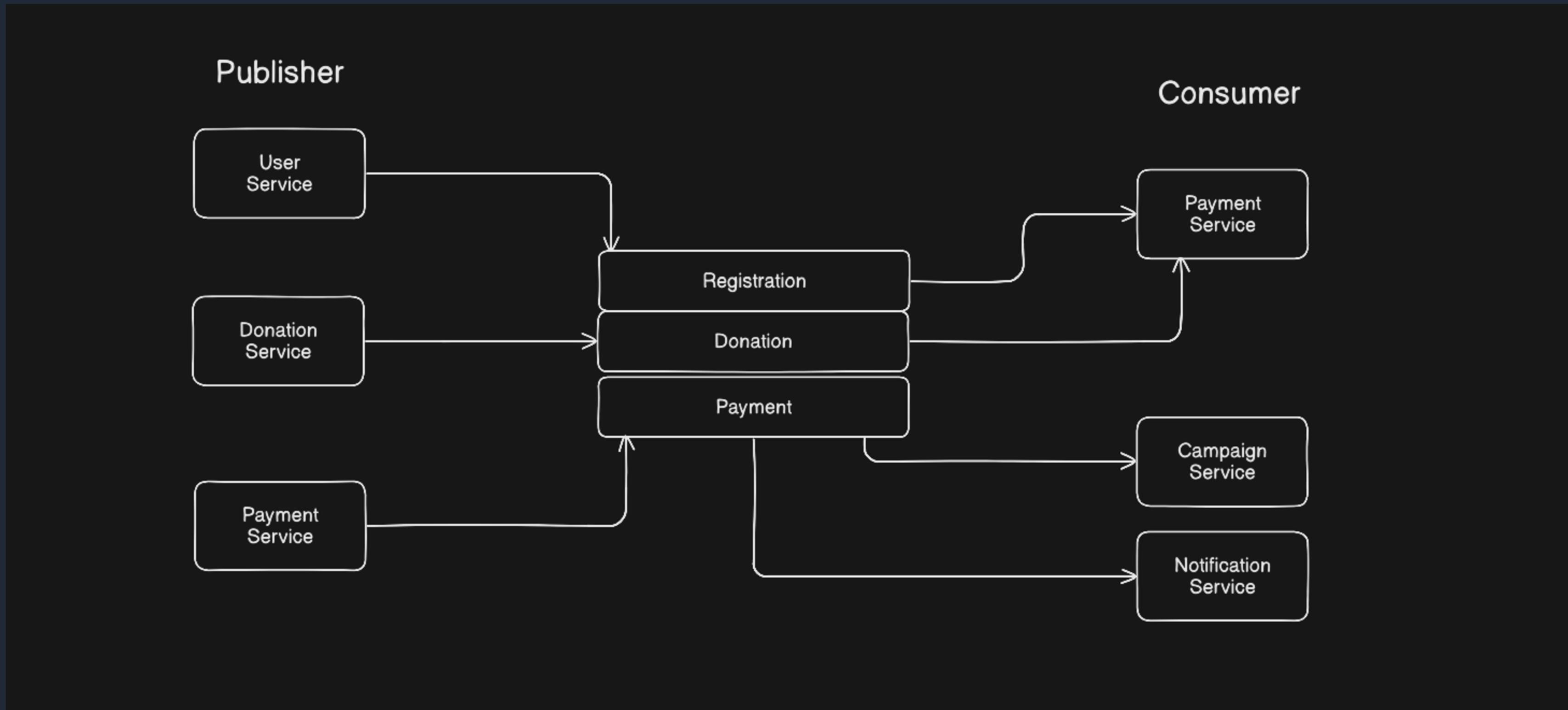
- Ali Asif Khan**
- Nafis Nahian**
- Estyak Ahamed**

# System design

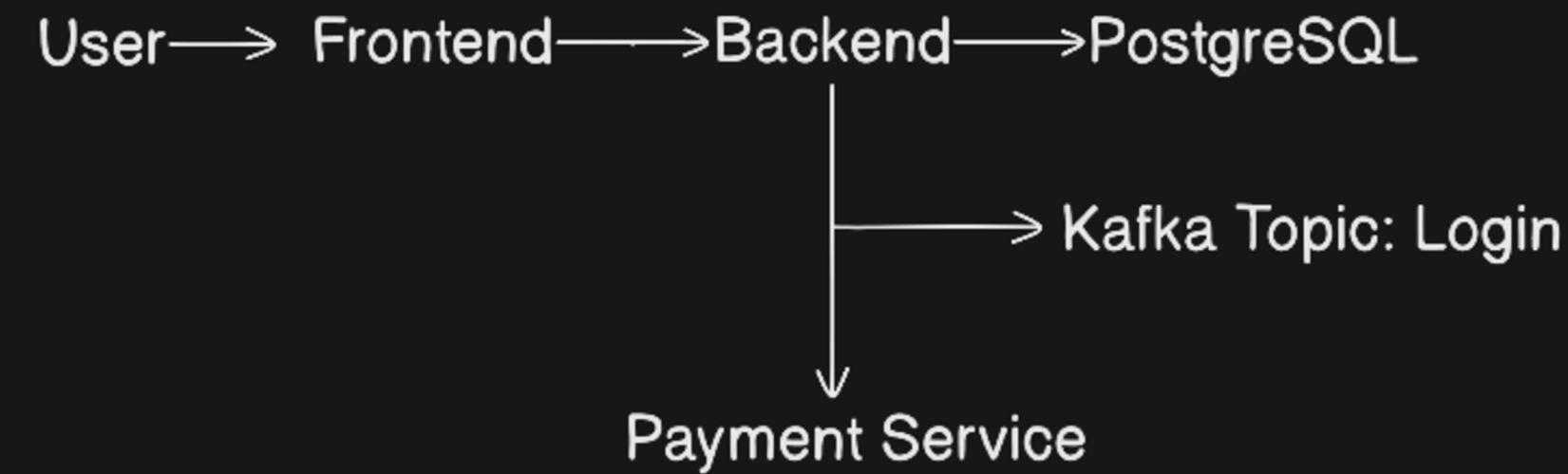


# Event Driven Architecture

Decoupling & Resilience



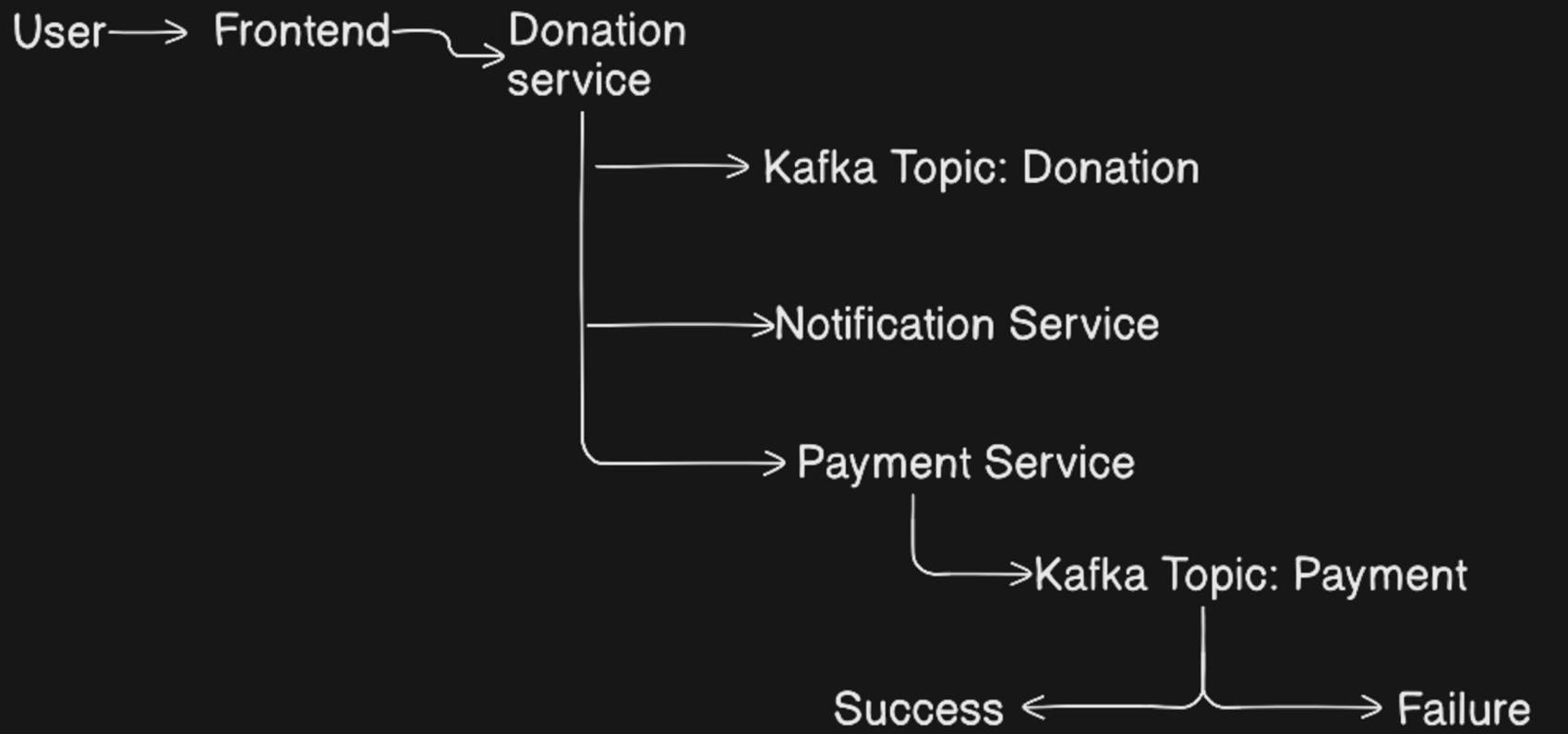
# User Registration Flow



The screenshot shows a "NEW USER REGISTRATION" interface. The user details listed are:

- User ID: 1
- Email: asif1@gmail.com
- Username: asif1

A success message at the bottom indicates: "Created user account: asif1 (ID: 1) with balance \$1000.00".

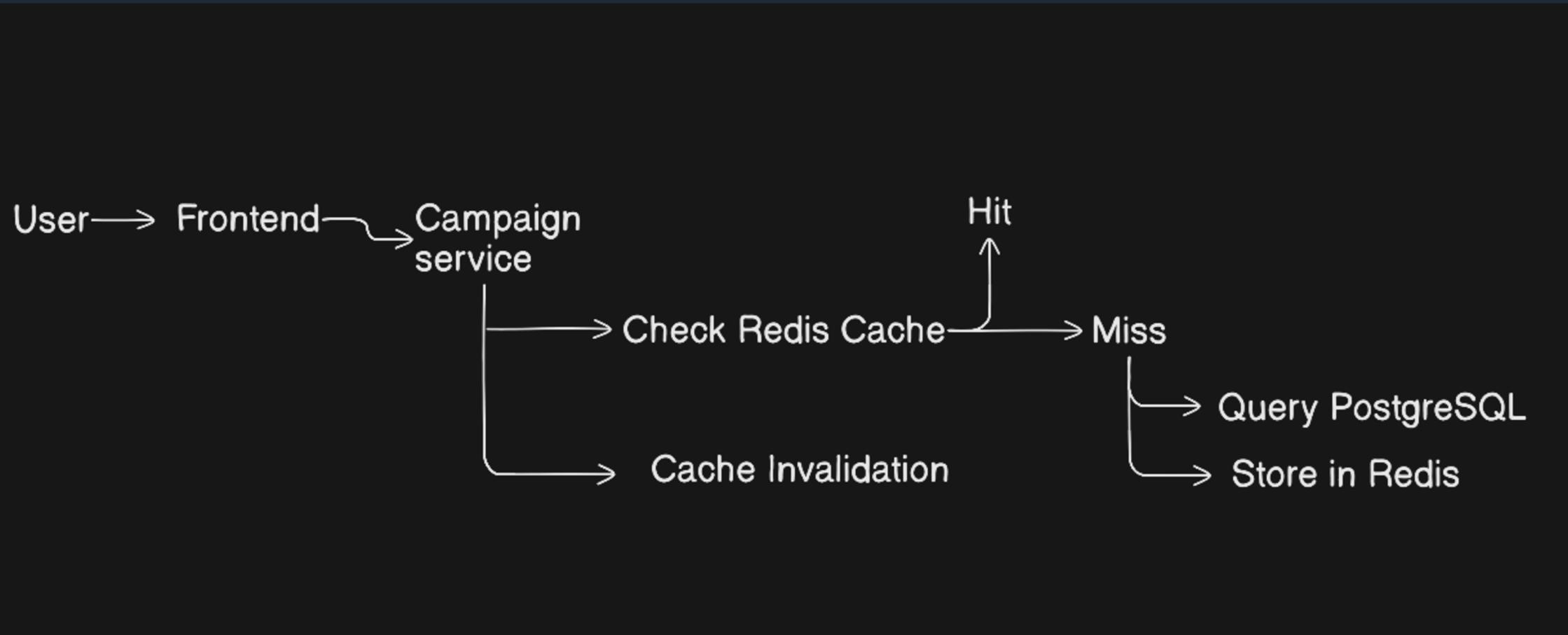


# Donation Flow

```
$ Ready to accept donations at POST /api/donate
$ Processing NEW donation from asif1@gmail.com for campaign 1: $100
$ Idempotency Key: 1-1-1763721808755-vr...
$ Kafka producer connected
$ ✓ Donation event published to Kafka: {
$   donationId: 'DON-1763721808850-0mdvtlevu',
$   campaignId: 1,
$   amount: 100,
$   userEmail: 'asif1@gmail.com'
$ }
```

# Campaign Query Flow

```
CACHE MISS FOR CAMPAIGN 1 - RETRIEVING FROM DB
Received payment event: {
    donationId: 'DON-1763721808850-0ndvtlevu',
    userId: 1,
    userEmail: 'asif1@gmail.com',
    campaignId: 1,
    amount: 100,
    timestamp: '2025-11-21T10:43:28.892Z',
    status: 'success',
    previousBalance: 1000,
    newBalance: 900,
    idempotencyKey: '1-1-1763721808755-vreavji28ys'
}
✓ Campaign updated: Clean Water Initiative - New total: $125100.00
```



# Notification Logs

 NEW DONATION RECEIVED!

---

 **Donation Details:**

- Donation ID: DON-1763721815857-9ua0uw3v7
- Campaign ID: 1
- Amount: \$50.00 USD
- Donor Email: asif1@gmail.com
- User ID: 1
- Status: pending
- Timestamp: 2025-11-21T10:43:35.857Z

 **Message Metadata:**

- Topic: donation
- Partition: 2
- Offset: 1
- Event Type: donation.created
- Source: donation-service

 **[Notification System]**

- Email notification would be sent to: asif1@gmail.com
- Thank you message for donation of \$50.00
- Campaign confirmation for Campaign #1



# Key Architectural Foundations

- **Idempotency:** Multiple clicks but single transaction
- **Reliable Events:** Events stored in kafka message queue
- **Observability:** Prometheus & Grafana
- **Logging:**
  - Filebeat → Collects Logs
  - Elastic Search
  - Kibana
- **Retries**

# Idempotency Log

```
Donation service listening on port 6000
$ Ready to accept donations at POST /api/donate
Metrics available at GET /metrics
Processing NEW donation from nafisnahian71@gmail.com for campaign 1: $500
  Idempotency Key: 1-1-1763722798541-uv...
Kafka producer connected
  ✓ Donation event published to Kafka: {
    donationId: 'DON-1763722798639-d7g89jsl6',
    campaignId: 1,
    amount: 500,
    userEmail: 'nafisnahian71@gmail.com'
  }
  ✓ Donation processed and cached with idempotency key
    Cache size: 1 entries
  Processing NEW donation from nafisnahian71@gmail.com for campaign 1: $100
  Idempotency Key: 1-1-1763723091398-1x...
  ✓ Donation event published to Kafka: {
    donationId: 'DON-1763723091433-wp84y7oqe',
    campaignId: 1,
    amount: 100,
    userEmail: 'nafisnahian71@gmail.com'
  }
  ✓ Donation processed and cached with idempotency key
    Cache size: 2 entries
  ⚡ Idempotent request detected for key: 1-1-1763723091398-1x...
  Returning cached response for user: nafisnahian71@gmail.com
```

# Prometheus

The screenshot shows the Prometheus Alerting interface with two main sections:

- cache-alerts** (`/etc/prometheus/alert.rules.yml`): This section contains four alerts:
  - LowCacheHitRate: Status is PENDING (1)
  - CriticalCacheHitRate: Status is PENDING (1)
  - RedisDown: Status is INACTIVE (1)
  - RedisMemoryHigh: Status is FIRING (1)
- container-alerts** (`/etc/prometheus/alert.rules.yml`): This section contains four alerts:
  - HighCPUUsage: Status is INACTIVE (4)
  - CriticalCPUUsage: Status is INACTIVE (4)
  - HighMemoryUsage: Status is INACTIVE (4)
  - CriticalMemoryUsage: Status is INACTIVE (4)

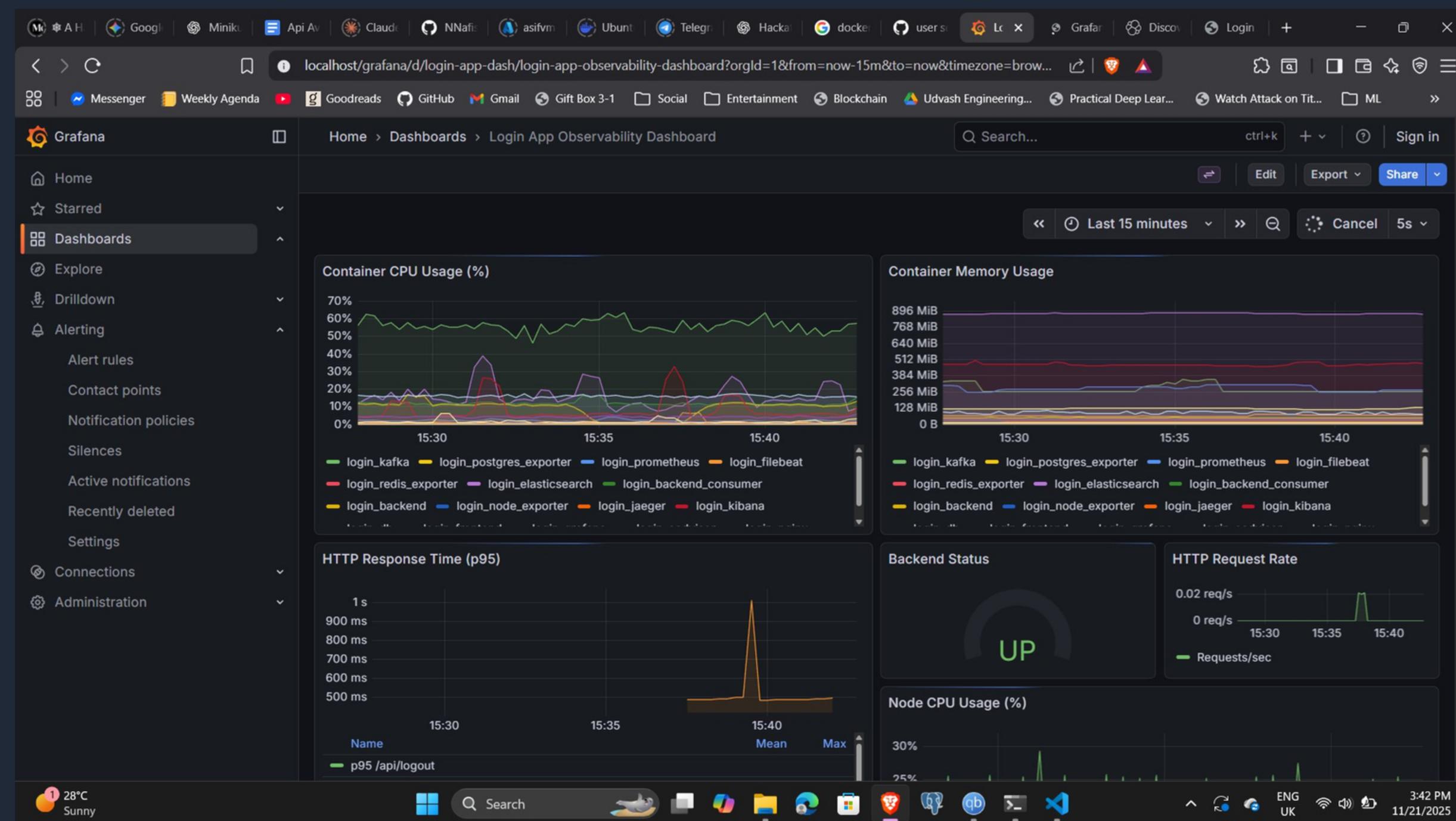
# Kibana Logging Interface

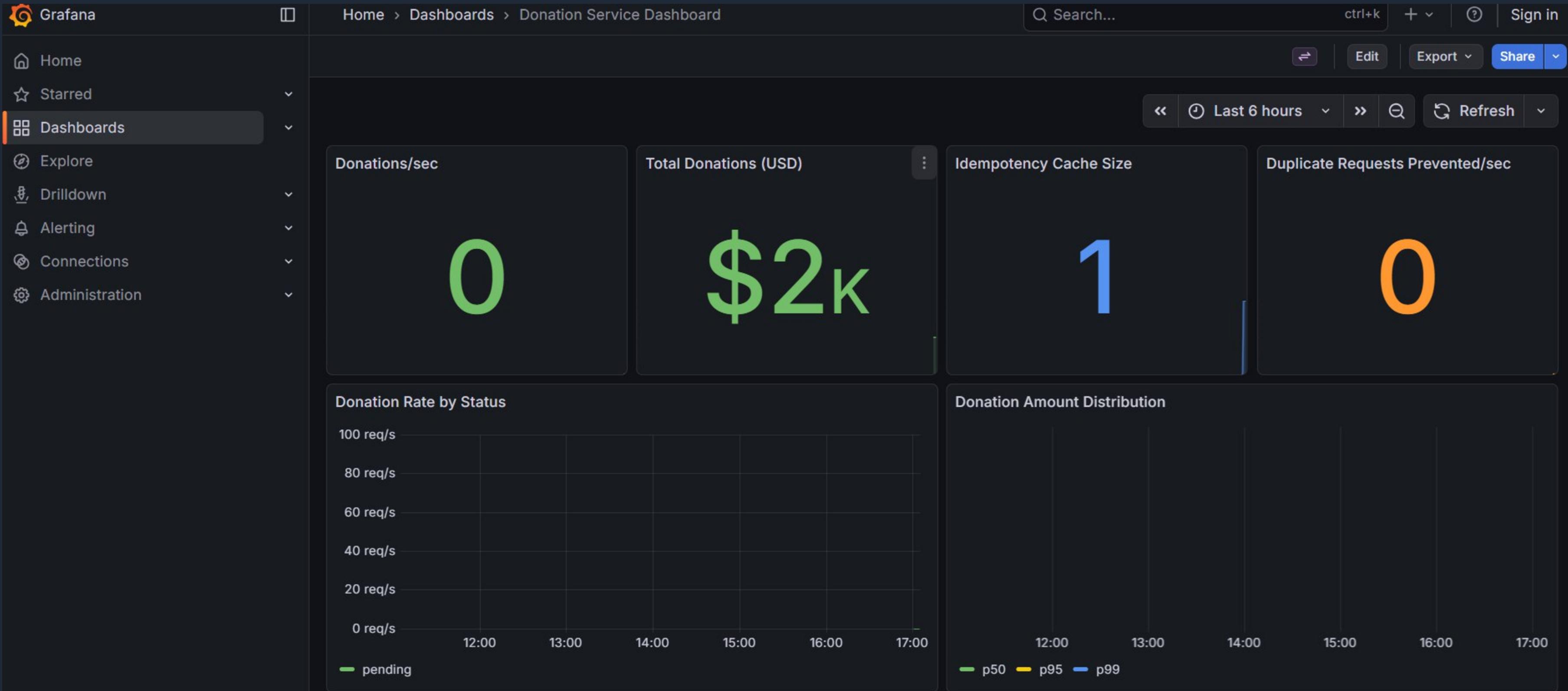
The screenshot shows the Kibana Discover interface with the following details:

- Header:** elastic logo, search bar "Find apps, content, and more.", navigation tabs: New, Open, Share, Alerts, Inspect, Save.
- Toolbar:** Filter dropdown set to "payment logs", search icon, KQL syntax input, time range "Last 15 minutes", Refresh button.
- Result Summary:** 54 hits.
- Available Fields:** A sidebar listing 52 fields including @timestamp, agent.id, container.labels.com\_docker\_compose\_config\_hash, etc.
- Visualizations:** Two stacked bars at the top showing event counts over time. The left bar spans from 16:46 to 16:52 on November 21, 2025. The right bar spans from 16:58 to 17:00.
- Document List:** A table showing log entries. One entry is expanded to show its full JSON structure:

```
],  
"host.name.keyword": [  
    "93cbea97f5da"  
],  
"container.labels.  
    com_docker_compose_service": [  
        "payment-service"  
    ],  
"container.id": [  
    "3cf230e1e0d8aa1ea1851efecd2e70ff3d6ce  
24a2 agent.id c2e42222-e13b-4ce6-a4be  
d e3adfcdf805ced5185cae480ee552851  
.labels.architecture x86_6...  
Copy value
```
- Bottom Navigation:** Rows per page: 100, page number 1.

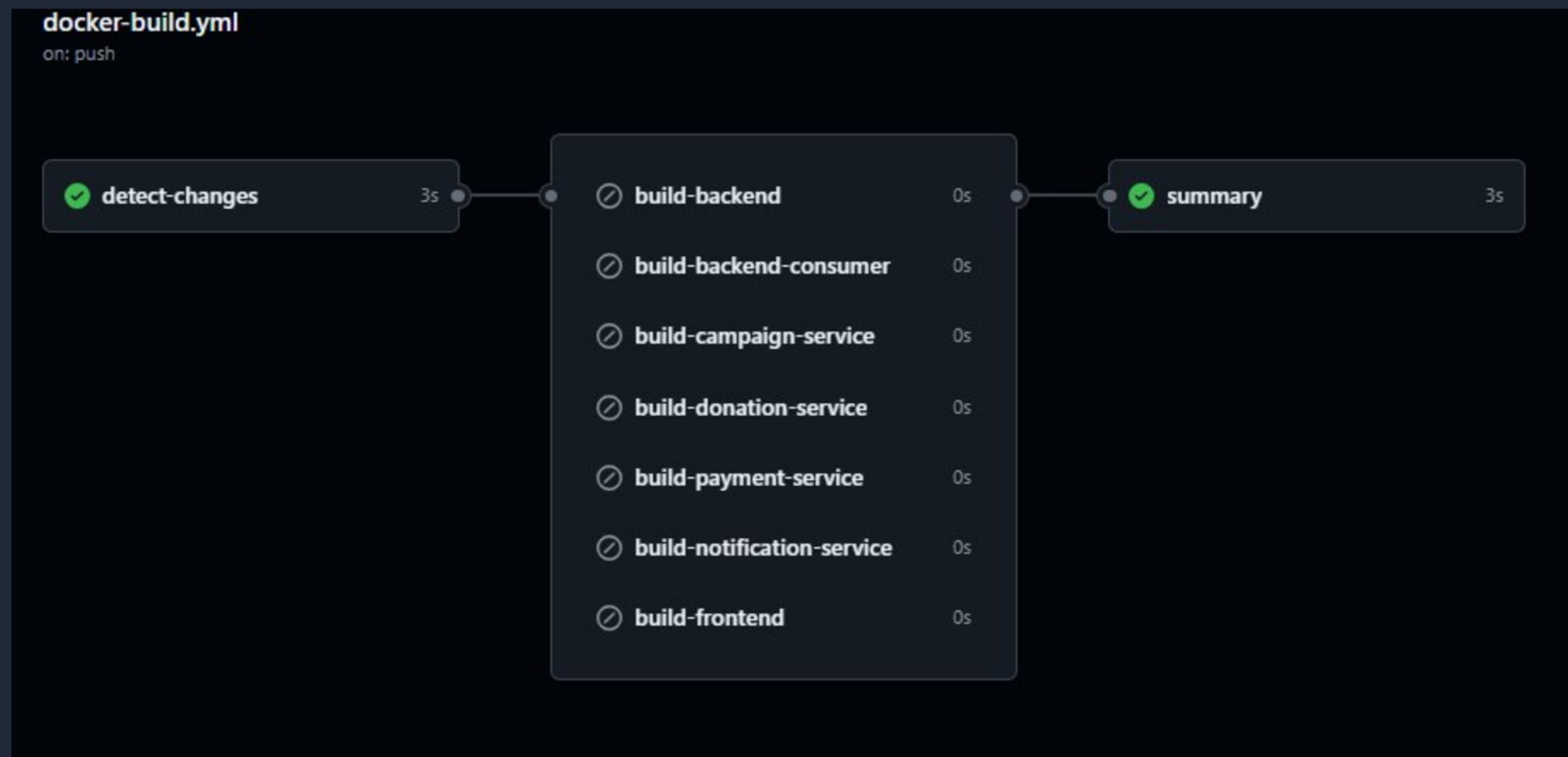
# Graphana Logs





# Graphana Dashboard

# CI Pipeline



# Tech Stack

- **Frontend**
  - React.js
  - Vite
- **Backend**
  - Nodejs
  - express
- **Message Broker**
  - Apache Kafka
- **Databases**
  - PostgreSQL
  - Redis
- **Monitoring**
  - Prometheus
  - Grafana
  - Jaeger
  - Elastic Search
  - Filebeat

# Load Balancing & Rate Limiting

## Load Balancing Features:

- Upstream server groups for backend, campaign, and donation services
- Least connections algorithm distributes load to server with fewest active connections
- Health checks