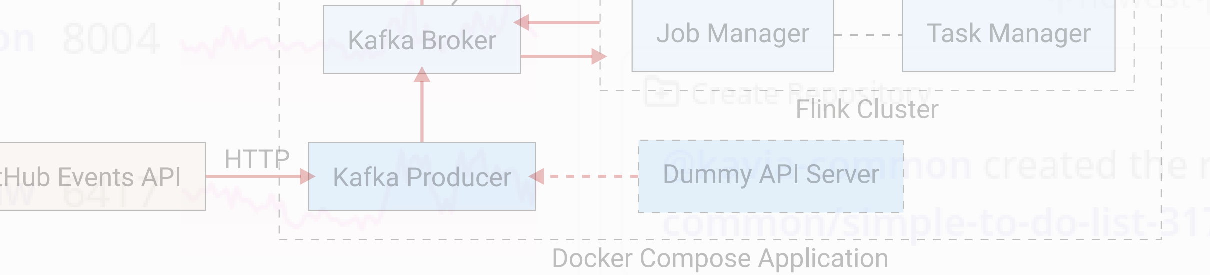


Final Presentation – Real-Time Big Data Processing 2024/2025

Real-Time Analysis of Public GitHub Activity

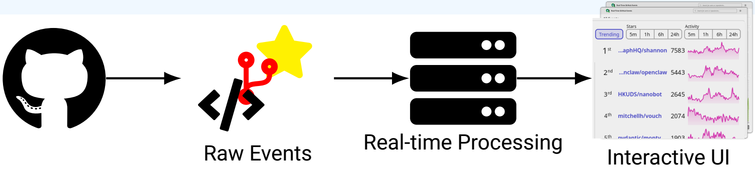
Roland Bernard

Febuary 2026

Application Domain

- Motivation and Context:
  - → GitHub hosts millions of developers and repositories.
  - → Generates large volumes of public events.
  - → Most existing tools focus mainly on historical analysis.

- Project Objectives:
  - → Collect GitHub events continuously.
  - → Process data in real time.
  - → Provide interactive dashboard.
  - → Ensure low latency and reliability.



1st	...aphHQ/shannon	7583
2nd	...nclaw/openclaw	5443
3rd	HKUDS/nanobot	2645
4th	mitchellh/vouch	2074
5th	pydantic/monty	1903

Description of the Data Sources

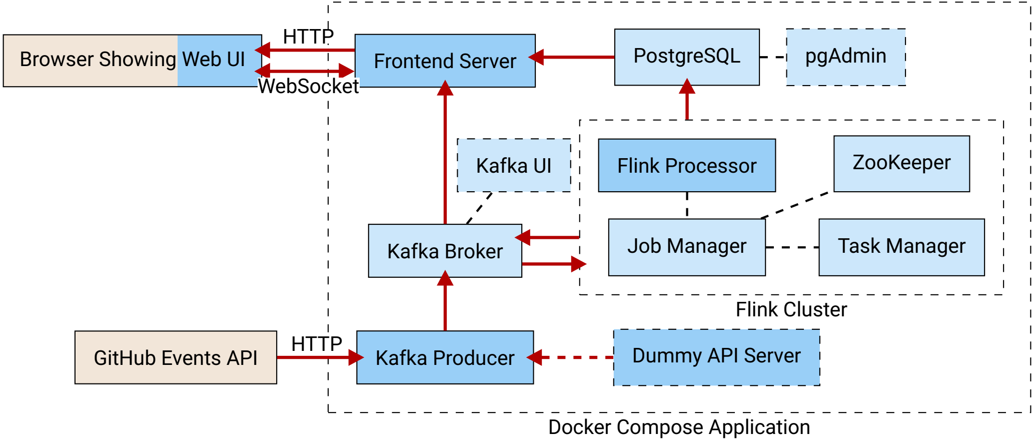
- GitHub Events API:
  - → Endpoint: `https://api.github.com/events`.
  - → Up to 300 most recent public events in JSON format.
  - → Rate limited, i.e., access token required.

- Key Properties:
  - → Variety: many event types, e.g., push, issues, stars, etc.
  - → Velocity: up to 120 events/sec, millions of events per day.
  - → Veracity: out-of-order and delayed events.
  - → Latency: API around 5 minutes behind real time.

```
{
  "id": "8251545586",
  "type": "PushEvent",
  "actor": {
    "id": 121951544,
    "login": "movieflixgr",
    "display_login": "movieflixgr",
    "gravatar_id": "",
    "url": "https://api.github.com/users/[
    "avatar_url": "https://avatars.github[
  },
  "repo": {
    "id": 1126450259,
    "name": "movieflixgr/Subtitles",
    "url": "https://api.github.com/repos/[
  },
  "payload": {
    "repository_id": 1126450259,
    "push_id": 30546526697,
    "ref": "refs/heads/main",
    "head": "d5e986993aa47729cb18a82ac9d1[
    "before": "6bf91f2d0600084cc7218c1934[
  },
  "public": true,
  "created_at": "2026-02-
08T20:09:50Z"
}
```

System Architecture

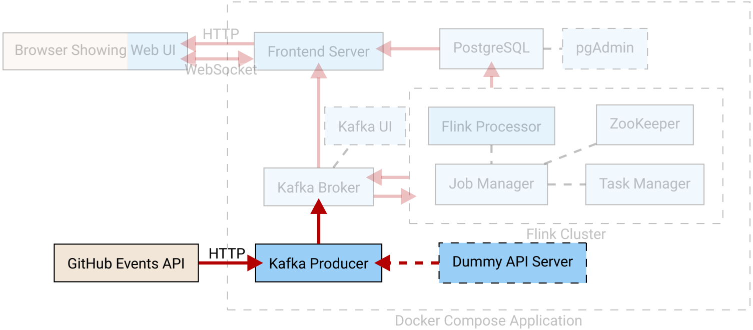
- Technology Stack:
  - → Backend: Java + Maven; RxJava; Apache Kafka; Apache Flink; PostgreSQL + TimescaleDB
  - → Frontend: Single Page Application with React + TypeScript; Vite; Recharts; RxJS
  - → Deployment: Docker and Docker Compose



System Architecture – Data Ingestion

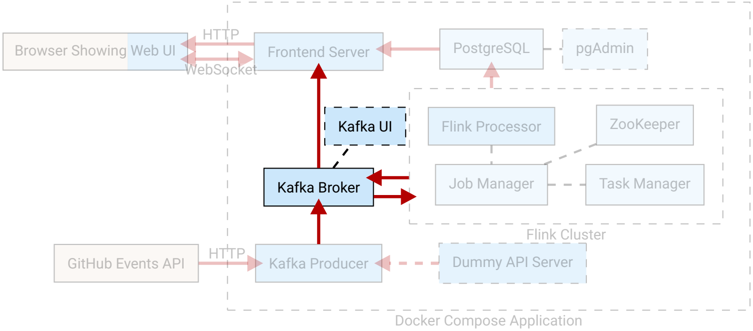
- Kafka Producer:
  → Polls GitHub API periodically, respecting rate limits.
  → Retrieves overlapping windows, then deduplicates events.
  → Publishes deduplicated events to Kafka topic.

- Testing Mode:
  → Dummy API server to simulate real GitHub API with configurable throughput.



System Architecture – Messaging

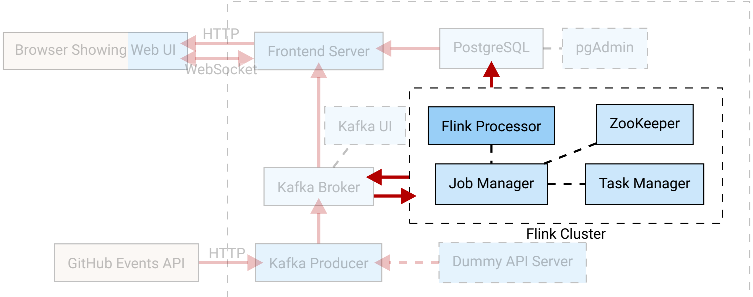
- Apache Kafka:
  → Decouples the ingestion from the processing.
  → Decouples the processing from the frontend.
  → Kafka message retention enables restarting Flink job without losing events.



System Architecture – Stream Processing

- Core Processing Tasks:
  - → Retrieve raw events from Kafka and parse JSON.
  - → Event-time processing using Watermarks with 10s maximum out-of-orderness.
  - → Writes results to PostgreSQL and Kafka.

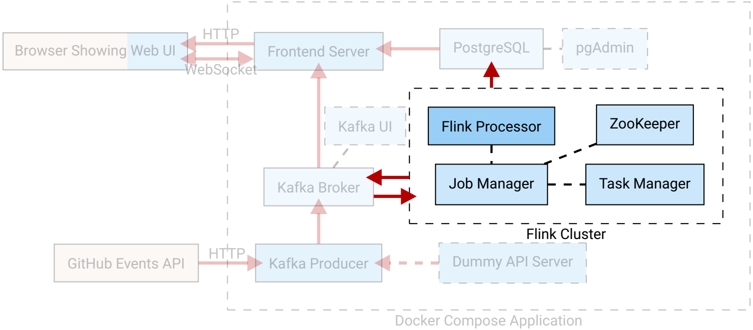
- Windowed Aggregations:
  - → Tumbling windows with 10s and 5min window sizes.
  - → Custom sliding windows with 5min, 1h, 6h, and 24h length, updated every second.



System Architecture – Stream Processing

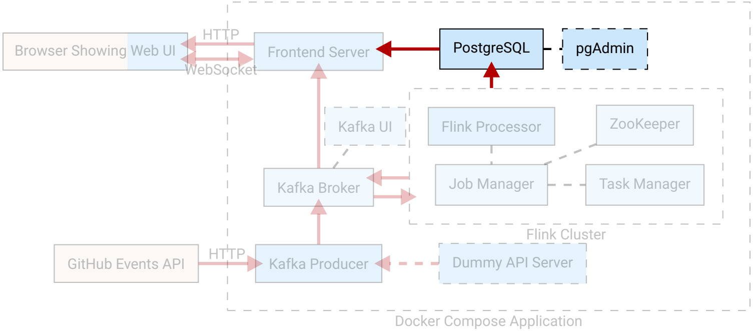
- Leaderboards:
  - → User and repository activity ranking.
  - → Based on latest values from sliding window aggregations.
  - → Optimization: Emits only changed rows and frontend reconstructs ranking.

- Trending Score:
  - → Weighted sum of recent stars: $t_{\text{score}} = 10s_{5\text{m}} + 5s_{1\text{h}} + 2s_{6\text{h}} + s_{24\text{h}}$.



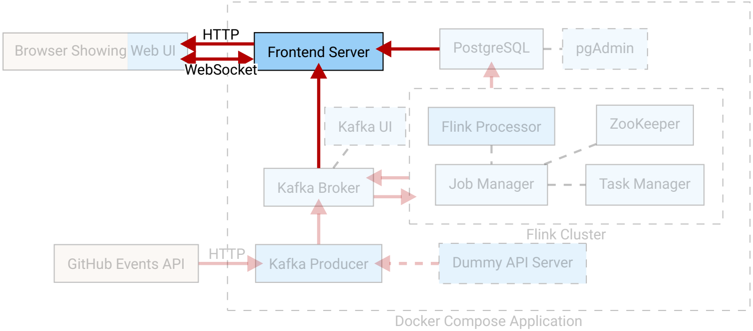
System Architecture – Storage

- PostgreSQL + TimescaleDB:
  → Persistent storage for results of Flink processor.
  → Handles queries for initial snapshot and historical data.
  → Makes use of TimescaleDB hypertables and retention policy.
  → Database commit before Kafka publish to ensure consistency.



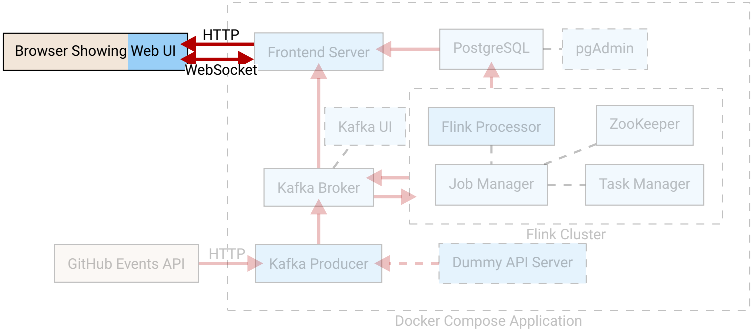
System Architecture – Data Serving

- Frontend Server:
  - → HTTP server for serving static files of the client web application.
  - → WebSocket server to serve API.
  - → Combines database queries for snapshots and Kafka streams for real-time updates.
  - → Manages client subscriptions to route events received from Kafka.



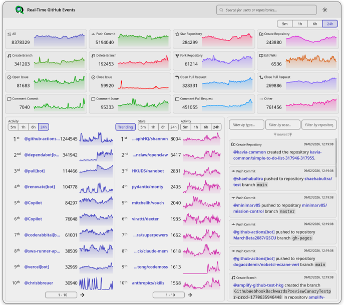
System Architecture – Frontend

- Web Application:
  - → A filterable live event stream.
  - → WebSocket server to serve API.
  - → Real-time counters for event volumes.
  - → User and repository leaderboards.



Functionalities and Demo

- Live Demo: `https://rtgh.rolandb.com`



Real-Time GitHub Events

Event type	Count
All	8378329
Push Commit	5194040
Star Repository	284299
Create Repository	243880
Create Branch	341203
Delete Branch	192453
Fork Repository	61214
Edit Wiki	6536
Open Issue	81683
Close Issue	59920
Open Pull Request	328331
Close Pull Request	269886
Comment Commit	7040
Comment Issue	95333
Comment Pull Request	451055
Other	761456

1st	@github-actions...	1244545	1st	...aphHQ/shannon	8004
2nd	@dependabot[b...	341942	2nd	...nclaw/openclaw	6417
3rd	@pull[bot]	114466	3rd	HKUDS/nanobot	2831
4th	@renovate[bot]	104778	4th	pydantic/monty	2405
5th	@Copilot	84297	5th	mitchellh/vouch	2040
6th	@Copilot	76048	6th	virattt/dexter	1935
7th	@coderabbitai[b...	61001	7th	...ra/superpowers	1662
8th	@swa-runner-ap...	38509	8th	...ck/claude-mem	1618
9th	@vercel[bot]	32969	9th	...tong/codemoss	1613
10th	@chrisbbreuer	30940	10th	anthropics/skills	1568

Create Repository 09/02/2026, 12:19:08
@kavia-common created the repository kavia-common/simple-to-do-list-317946-317955.

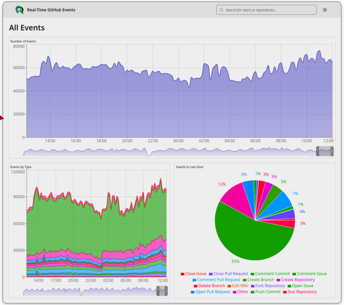
Push Commit 09/02/2026, 12:19:08
@shaehabultra pushed to repository shaehabultra/test branch `main`

Push Commit 09/02/2026, 12:19:08
@minimarv85 pushed to repository minimarv85/mission-control branch `master`

Push Commit 09/02/2026, 12:19:08
@github-actions[bot] pushed to repository MarchBeta2087/GSCU branch `gh-pages`

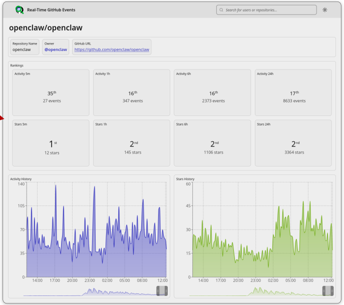
Push Commit 09/02/2026, 12:19:08
@github-actions[bot] pushed to repository dogaozdemir/nobetci-eczane-veri branch `main`

Create Branch 09/02/2026, 12:19:08
@amplify-github-test-hkg created the branch `GithubWebhookBackwardsPreviewCanaryTestpr-prod-1770635946448` in repository amplify-



Real-Time GitHub Events

All Events



Real-Time GitHub Events

openclaw/openclaw

Repository Name	Owner	GitHub URL
openclaw	@openclaw	https://github.com/openclaw/openclaw

Rankings

Activity 5m	Activity 1h	Activity 6h	Activity 24h
35th 27 events	16th 347 events	16th 2373 events	17th 8633 events

Stars 5m	Stars 1h	Stars 6h	Stars 24h
1st 12 stars	2nd 145 stars	2nd 1106 stars	2nd 3364 stars

Conclusion

- Successes:
  - → Robust architecture able to handle 70-120 events per second.
  - → Docker and Docker Compose simplifies deployment.
  - → Responsive UI with leaderboards and charts updated in real-time.

- Challenges:
  - → Real-time rankings and long sliding window efficiency.
  - → API changes made by GitHub reduced information.
  - → Debugging Flink operator logic and state management.

- Future Improvements:
  - → More advanced models to detect trending repositories.
  - → Social media integration to see where repositories or users are mentioned.
  - → Sentiment analysis on the text content of certain event types.