KAKAO CORP.

APACHE S2GRAPH (INCUBATING) AS A USER EVENT HUB

ABSTRACT

Apache S2Graph (incubating) is a graph database designed to handle transactional graph processing at scale.

Its API allows you to store, manage and query relational information using edge and vertex representations in a fully asynchronous and non-blocking manner.

However, at Kakao Corp., where the project was originally started, we believe that it could be so much more. There have been efforts to utilize S2Graph as the centerpiece of Kakao's event delivery system taking advantage of its strengths such as;

- flexibility of seamless bulk loading, AB testing, and stored procedure features,
- multitenancy that allows interoperability among different services within the company,
- and most of all, the ability to run various operations ranging from basic CRUD to multi-step graph traversal queries in realtime with large volumes.

We would like to share the story behind this rather unconventional choice of technology with the Apache world.

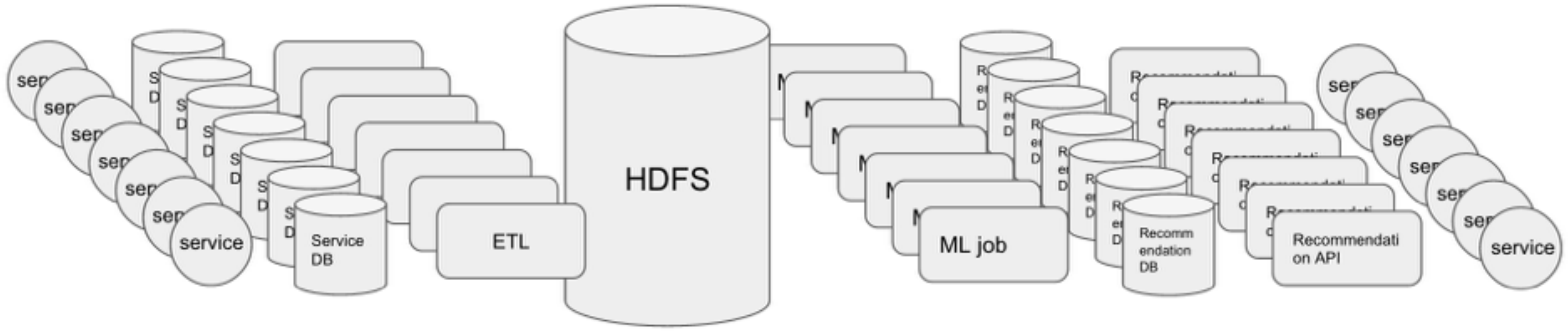
KAKAO

- #1 Messenger (with 93% M/S, KakaoTalk became a verb!!)
- #1 Music Streaming (AND #4!!)
- #1 O2O Taxi Service
- #2 Search Engine
- #2 Portal
- + social network, webtoon, video streaming, and so on.
- eying to become a “mobile lifestyle platform”

APACHE S2GRAPH (INCUBATING)

- Property Graph Model: Vertices + Edges + Properties
- S2Graph = Property Graph Model + Scalability + Fast CRUD Operations
- Graph-processing layer atop HBase
- Designed for distributed and fault-tolerant management of highly interconnected data at web scale
- Optimized for serving versatile ranking queries on higher-order relationships within graph data
- Features: idempotency, eventual consistency, low latency, high concurrency, and a powerful set of graph query APIs for massive graph data processing in real-time

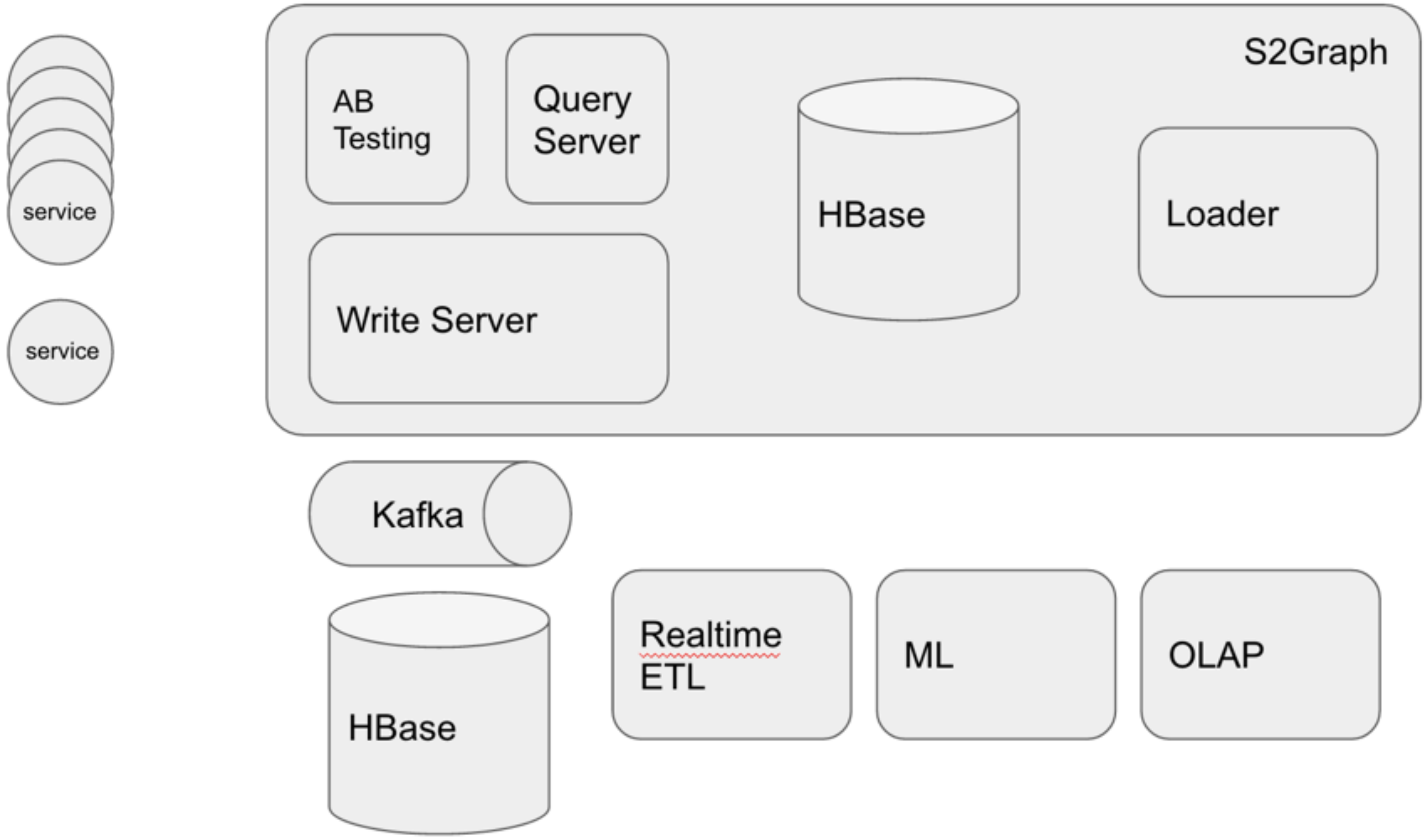
BEFORE S2GRAPH



BEFORE S2GRAPH

- Batch: Daily updates
- Fragmentation: N different services -> n different DB schemas, log formats, ETLs, ML jobs, and even APIs
- Redundancy: Mostly ctrl c ctrl v (or command c command v)
- No or little feedback: one and done!
- Inefficient!

AFTER S2GRAPH

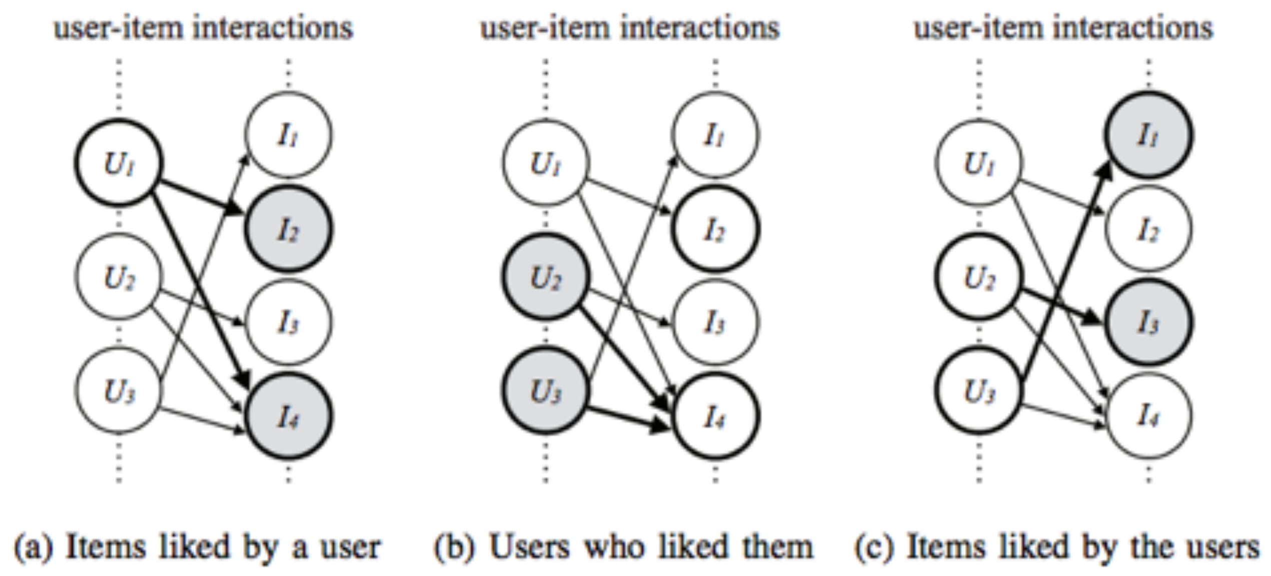


AFTER S2GRAPH

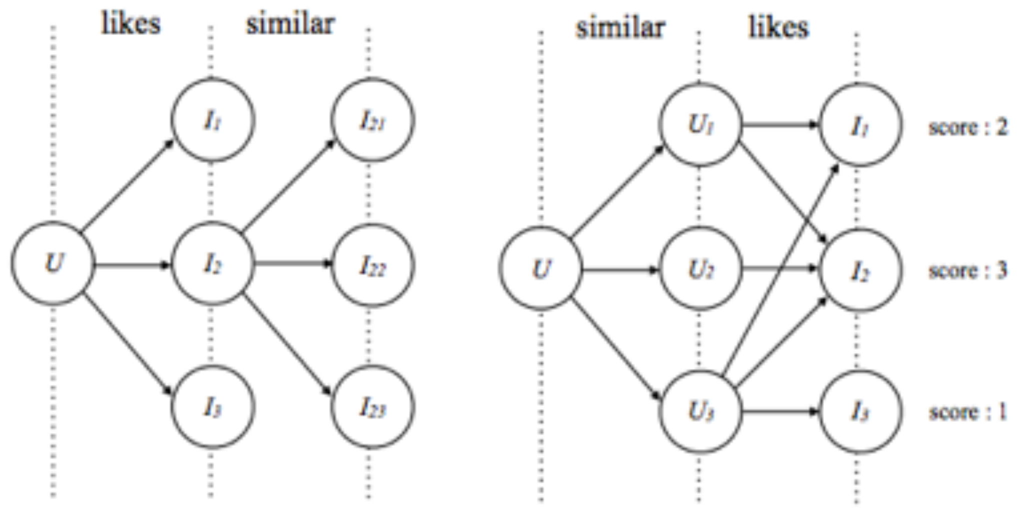
- Real-time
- Flexibility
- Unified Schema + API
- Interoperability among different services

AFTER S2GRAPH: REAL-TIME

- OLTP graph traversal



(a) Items liked by a user (b) Users who liked them (c) Items liked by the users

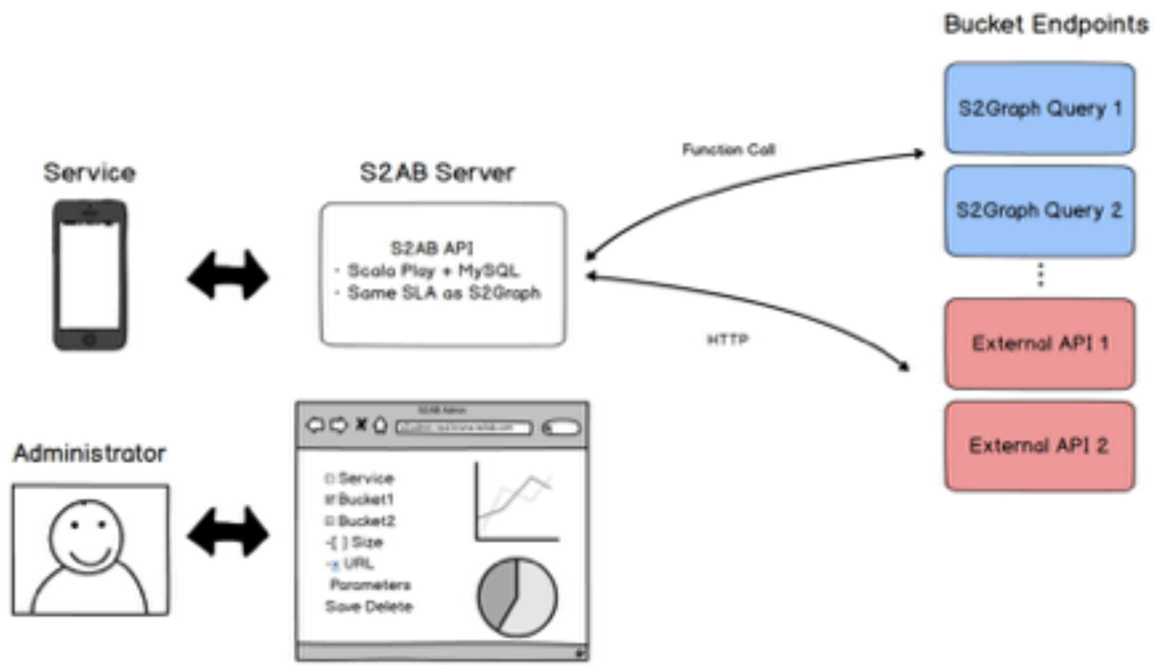


(a) Item-based CF (b) User-based CF

- Streamlined ETL

AFTER S2GRAPH: FLEXIBILITY

- AB testing



- ML + bulk loading

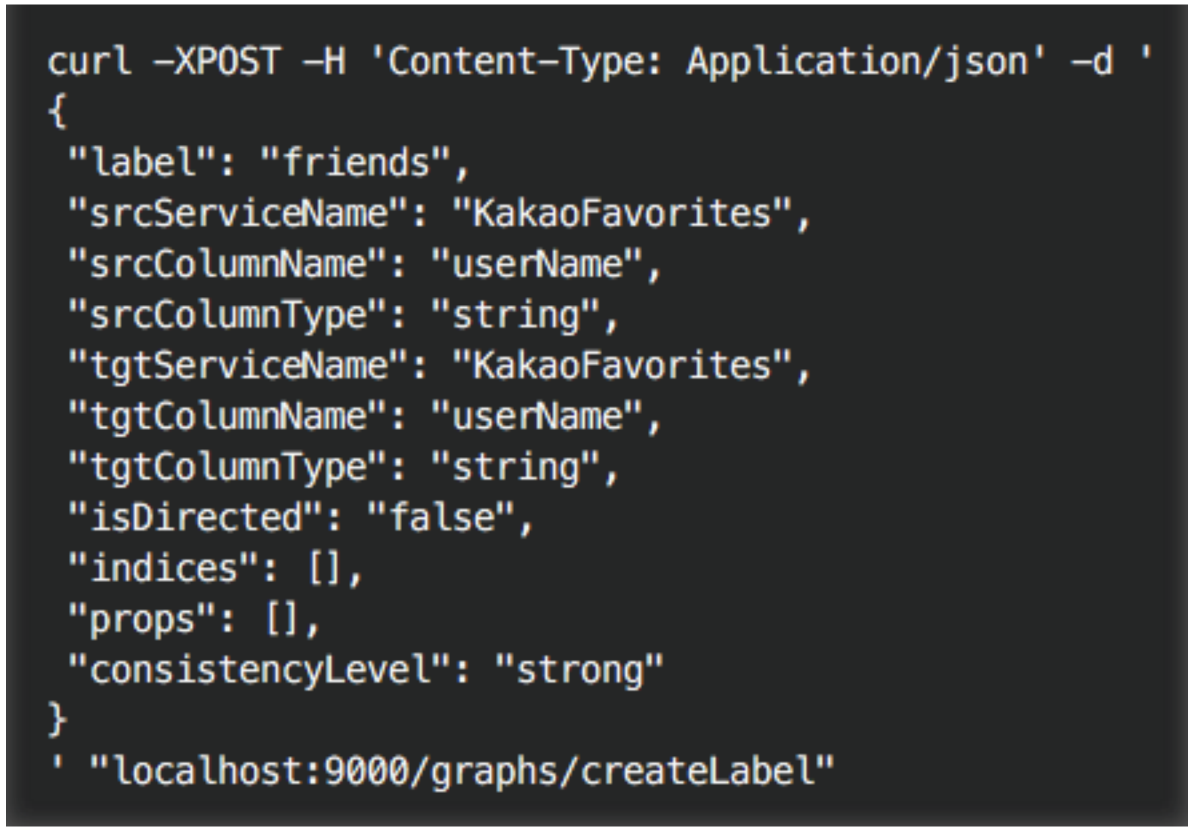
APACHE Spark™

+

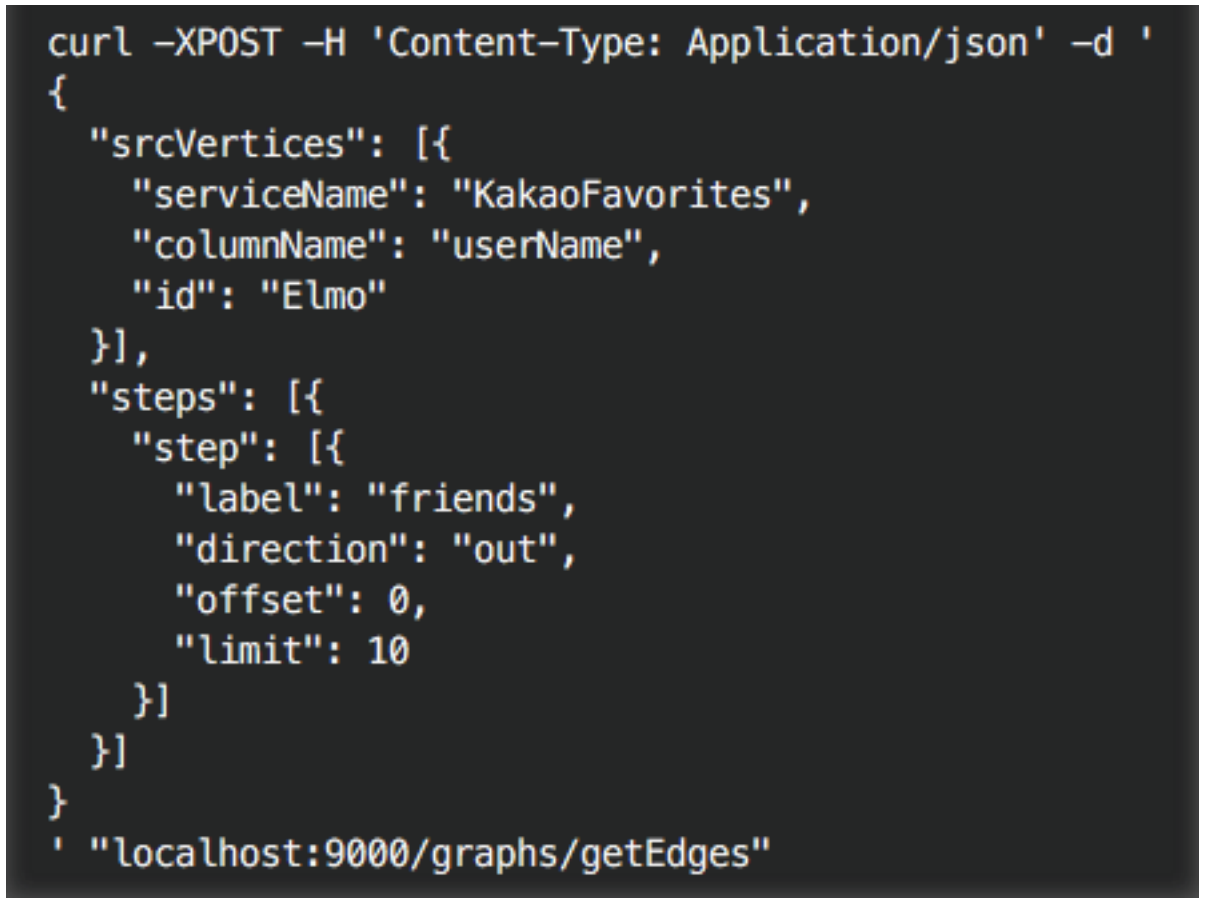
S2 GRAPH Loader

AFTER S2GRAPH: UNIFIED SCHEMA + API

- Edge format
- Graph API



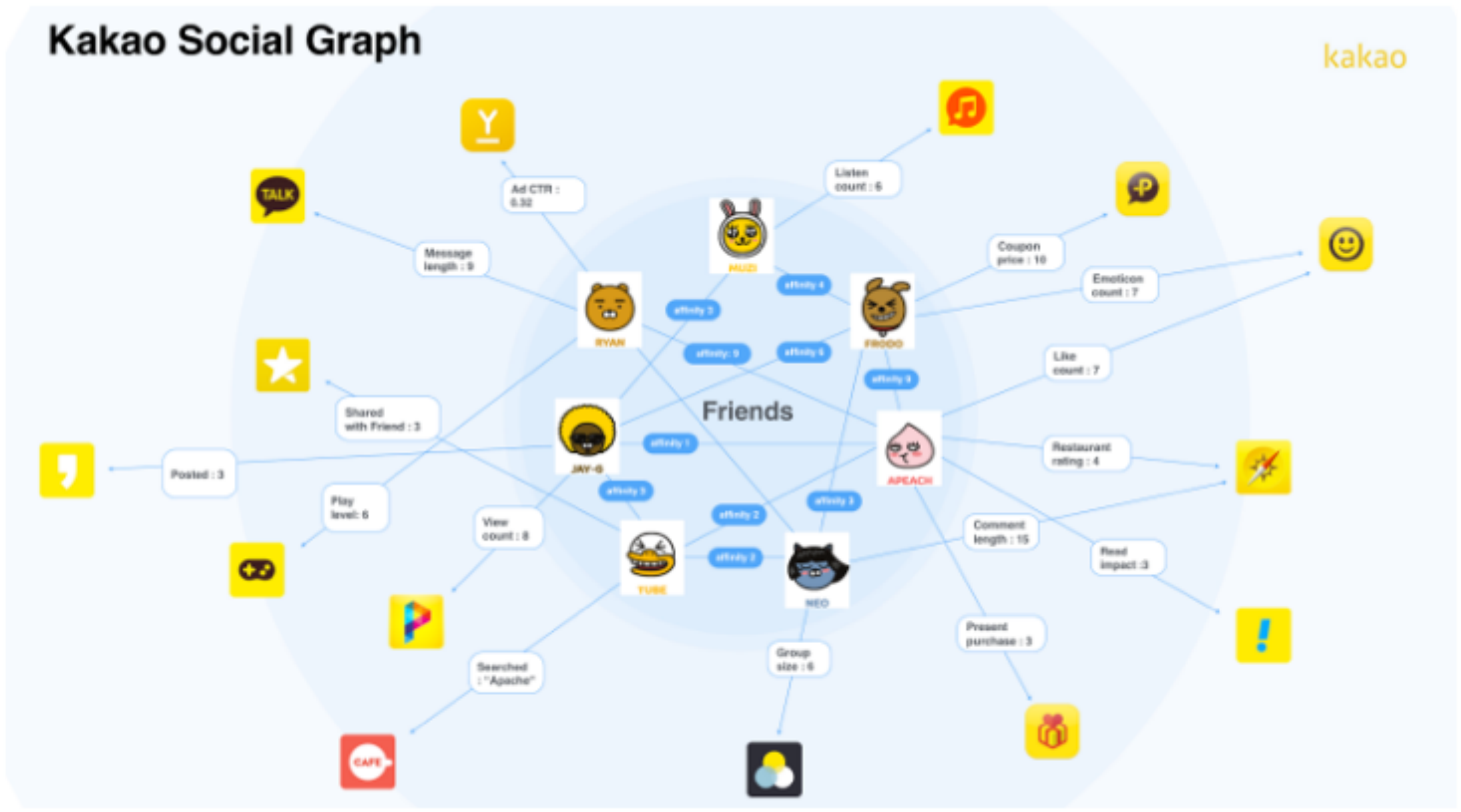
```
curl -XPOST -H 'Content-Type: Application/json' -d '
{
 "label": "friends",
 "srcServiceName": "KakaoFavorites",
 "srcColumnName": "userName",
 "srcColumnType": "string",
 "tgtServiceName": "KakaoFavorites",
 "tgtColumnName": "userName",
 "tgtColumnType": "string",
 "isDirected": "false",
 "indices": [],
 "props": [],
 "consistencyLevel": "strong"
}
' "localhost:9000/graphs/createLabel"
```



```
curl -XPOST -H 'Content-Type: Application/json' -d '
{
  "srcVertices": [{
    "serviceName": "KakaoFavorites",
    "columnName": "userName",
    "id": "Elmo"
  }],
  "steps": [{
    "step": [{
      "label": "friends",
      "direction": "out",
      "offset": 0,
      "limit": 10
    }]
  }]
}
' "localhost:9000/graphs/getEdges"
```

AFTER S2GRAPH: INTEROPERABILITY

- Multitenancy
- Universal (cross-service) recommendations



THE PROJECT

- Website: https://s2graph.incubator.apache.org/
- Mail list: dev-subscribe@s2graph.incubator.apache.org, users-subscribe@s2graph.incubator.apache.org