SKOD: A Framework for Situational Knowledge on Demand

Servio Palacios[1] and KMA Solaiman[1], Pelin Angin[2], Alina Nesen[1], Bharat Bhargava[1], Zachary Collins[3], Aaron Sipser[3], Michael Stonebraker[3], James MacDonald[4]

VLDB Workshop: POLY'19
Los Angeles, California
August 30, 2019

1 PURDUE UNIVERSITY
2 METU
3
4 NORTHROP GRUMMAN

Outline

- Objectives
- Problem Statement
- State-of-the-art
- SKOD Functionalities
- Datasets
- SKOD Architecture

- Architecture Modules
  - Data Streaming
  - Feature Extraction
  - PostgreSQL Database
  - Graph-based Indexing Layer
  - Front End

Objectives

- Retrieve **knowledge** for multiple users *changing* needs and mission
- Relate **multi-modal** data and update the **knowledge** for users
- Integrate new *streaming data* with knowledge queries already used by mission
- Complete the unfulfilled data needs for missions
- Discover new **knowledge** that can benefit mission

…

Objectives

- Research **transfer learning**, reinforcement learning, active learning and apply to NG large databases (sensors, signals, text, phone calls, videos, images, voice)
- Make system practical and responsive and efficient by using systems and tools already available and used in industry

Problem Statement

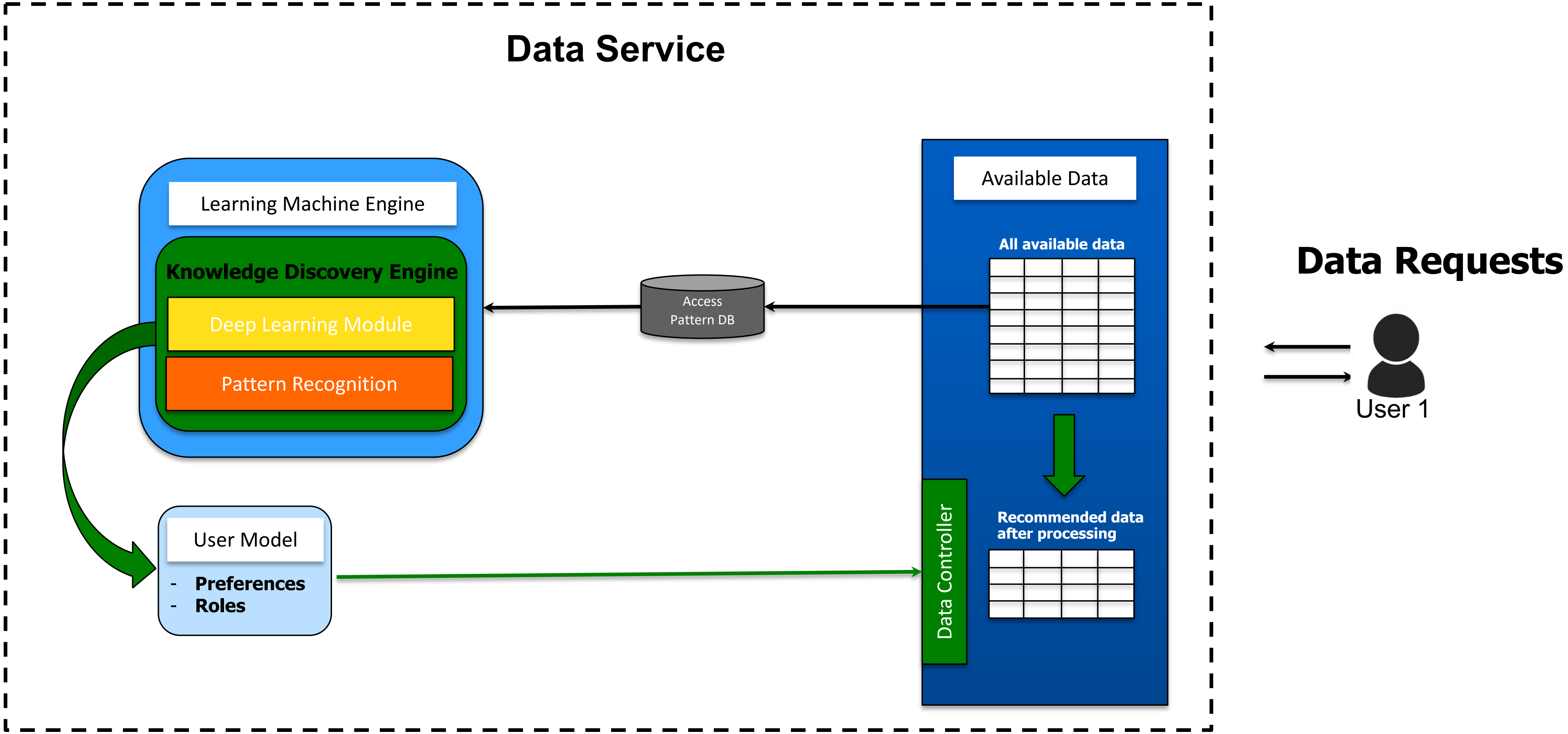
Determine ***relevant*** *objects* according to data at rest and *heterogeneous* data streams utilizing **knowledge** built on top of a relational database. **Cache** appropriate data and queries to improve performance.

Related Work

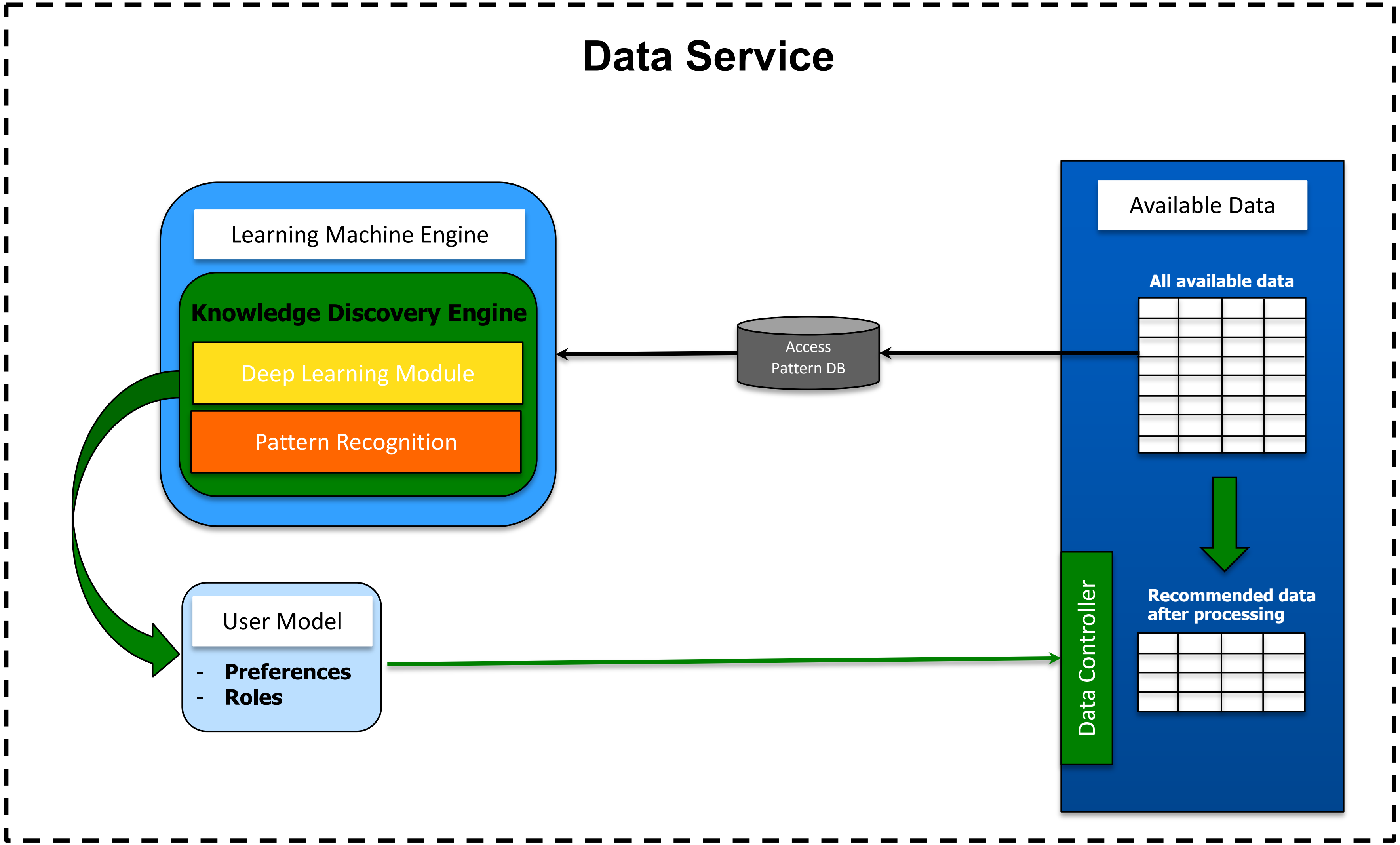
Specific domains **multi-modal** fusion of data [KBZ18, MVK17, AHD+15, FFV15].

Evolving mission requirements with heterogeneous user interests

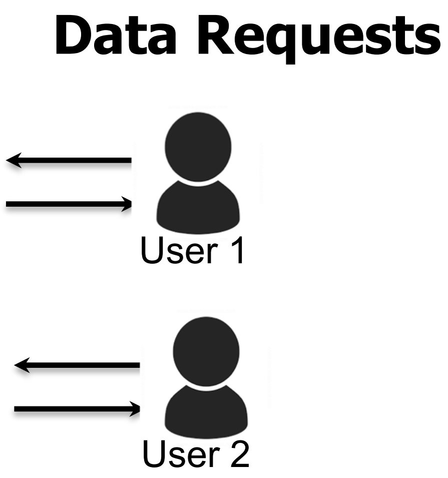
SKOD Functionalities



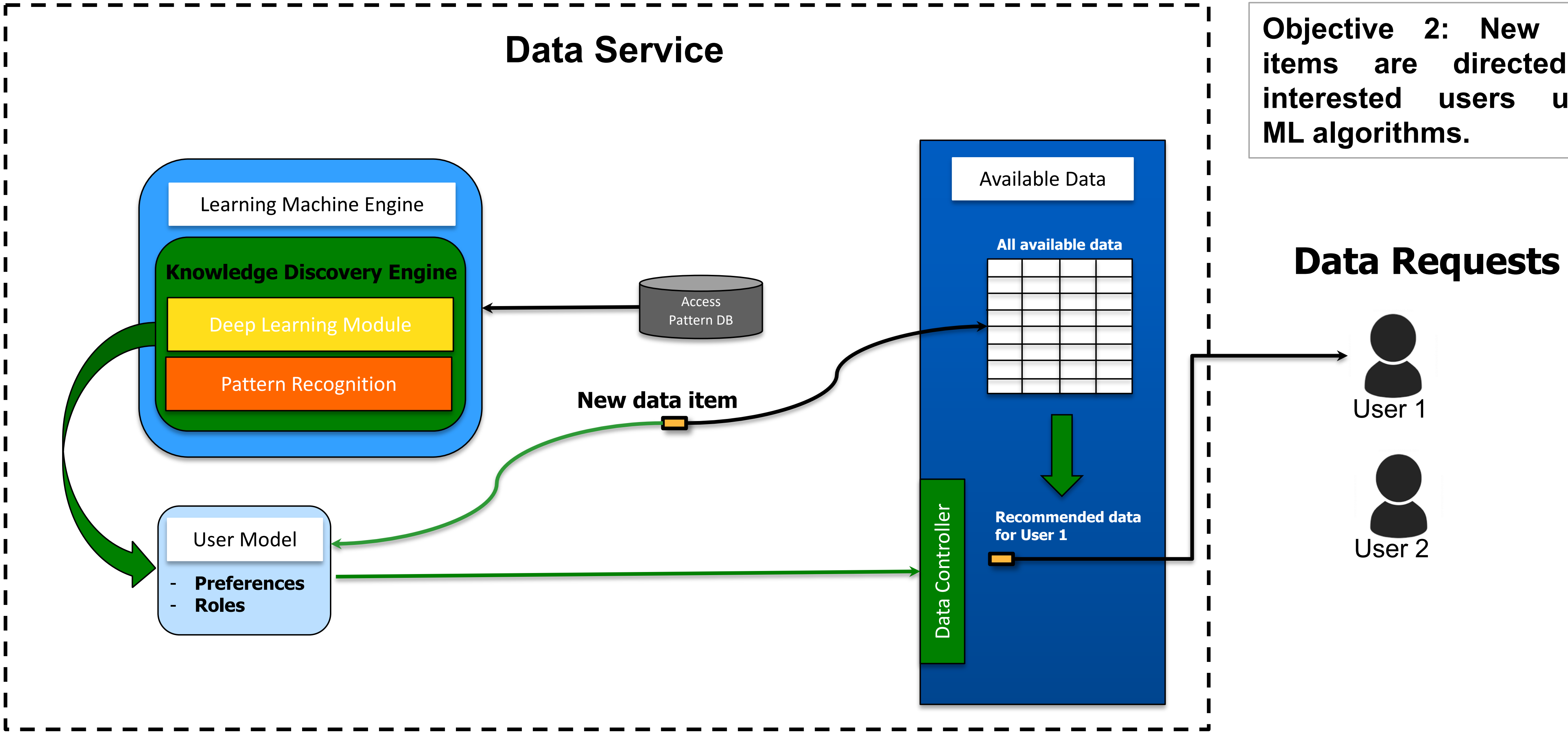
SKOD Functionalities



Objective 1: Relevant data is efficiently passed to users based on their requests

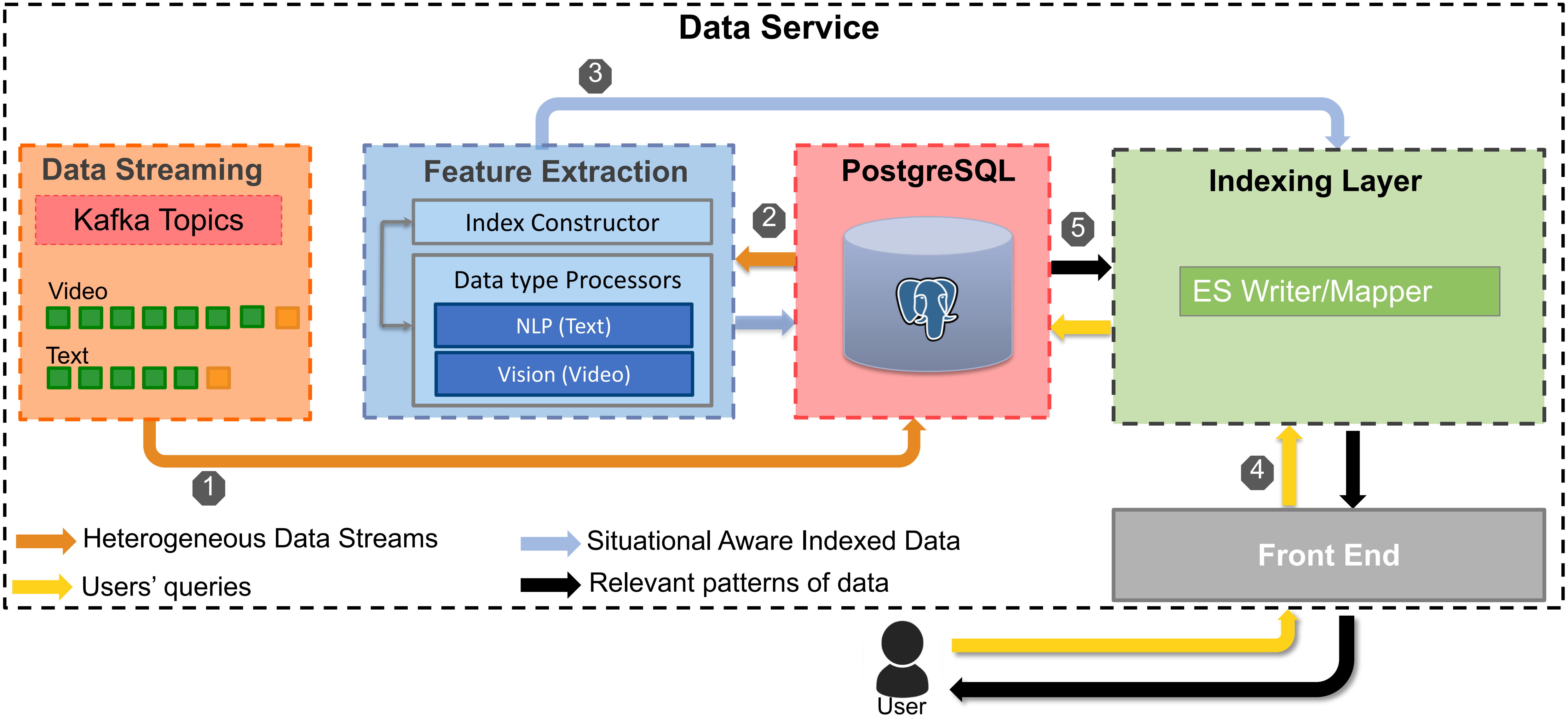


SKOD Functionalities



Objective 2: New data items are directed to interested users using ML algorithms.

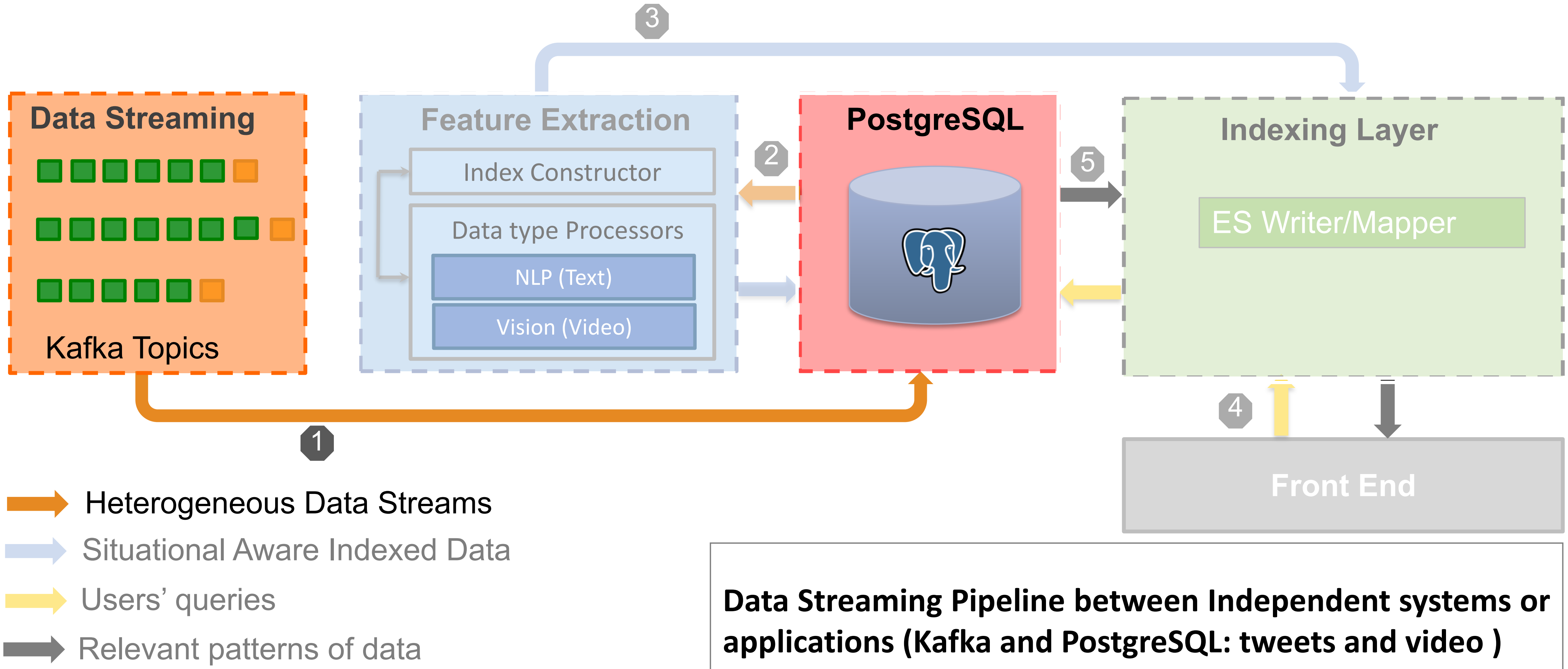
Proposed Architecture



Data Streaming Module

An Apache Kafka based Solution

Data Streaming Module

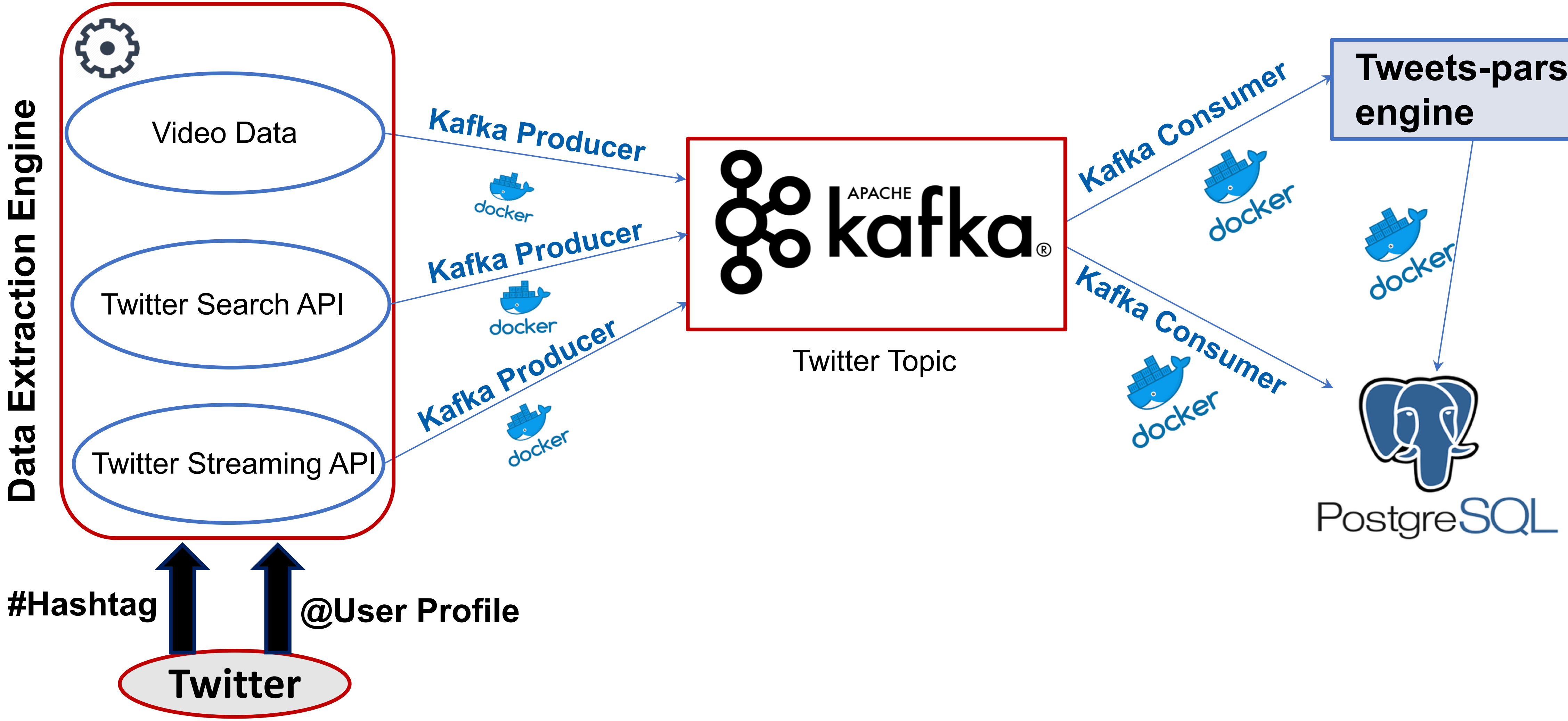


Data Streaming Pipeline between Independent systems or applications (Kafka and PostgreSQL: tweets and video)

Problem Statement

- Retrieve **RESTFUL** and **Streaming** Tweets
- ***Parse** collected metadata* to **extract targeted information** and *store in Postgres*
- Build a Data Processing Pipeline
- Microservice for production

Research Approach



Tweets-Parser-Engine

- Parses metadata to extract
  - Full tweet text
  - User Information
  - Hashtags, URLs, User mentions
  - Geolocation (latitude, longitude)
- Differentiates and processes
  - Original tweets
  - Retweets
  - Quoted tweets

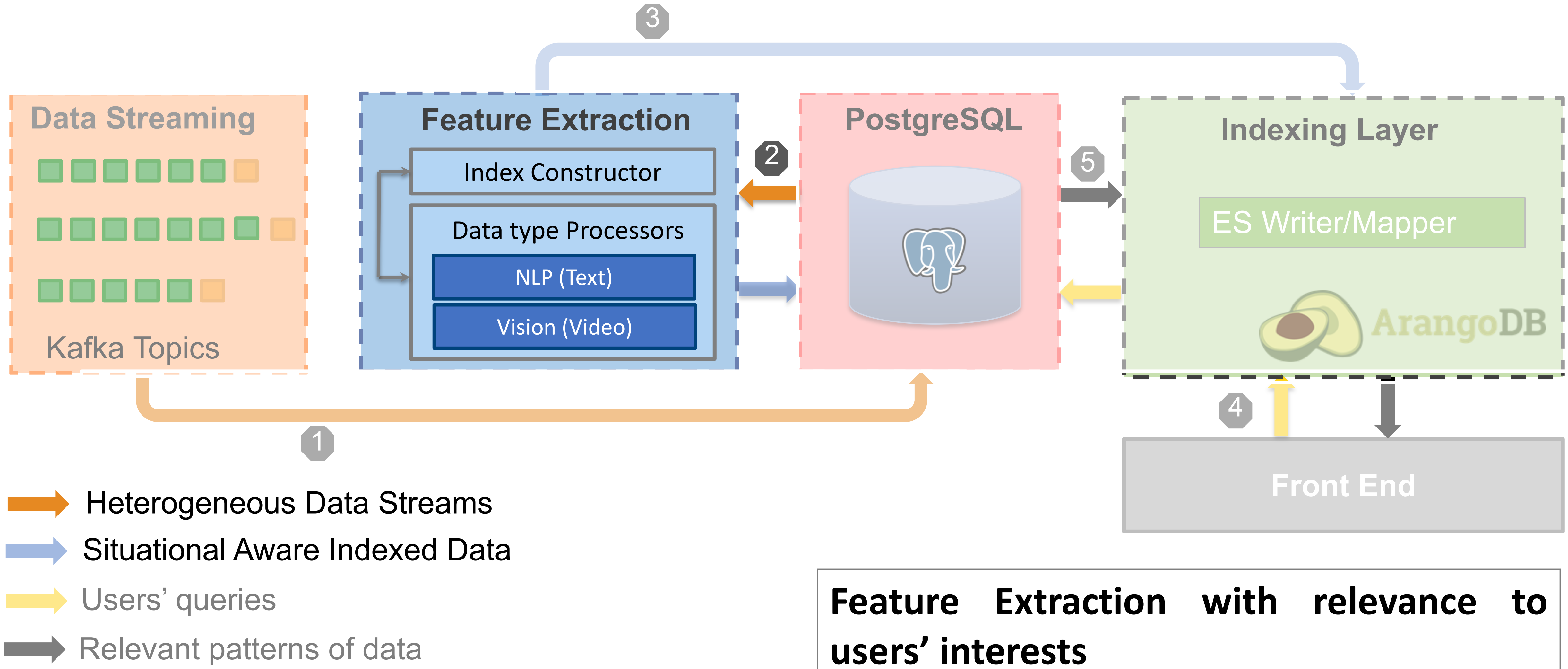
Benefits

- Provides dataset for use cases – **Unattended Child Identification, Crime Detection and prediction**
- A pipeline to **consume** data from **heterogeneous sources** and then **dispatch** to disparate systems
  - Fault-tolerant
  - Scalable
  - Asynchronous
  - Capable of stream processing

Feature Extraction Module

A Machine Learning and Deep Learning Solution

Feature Extraction Module



Feature Extraction with relevance to users' interests

Datasets : Unstructured Text

- For unstructured text, we are collecting **Twitter data**
- Final target: **1 million tweets about Cambridge, MA**
- As data source, official hashtags and user profiles has been chose
- Currently **140K tweets** has been collected
- Every week the repository is updated (new data is collected)
- **Parsed and processed tweets** can be retrieved from MIT database:
  - contains metadata with actual Tweet text
  - Twitter object and User objects are stored in different tables

Preprocessing Tweets

- Social media text has **jargon**, misspellings, special slangs, emojis

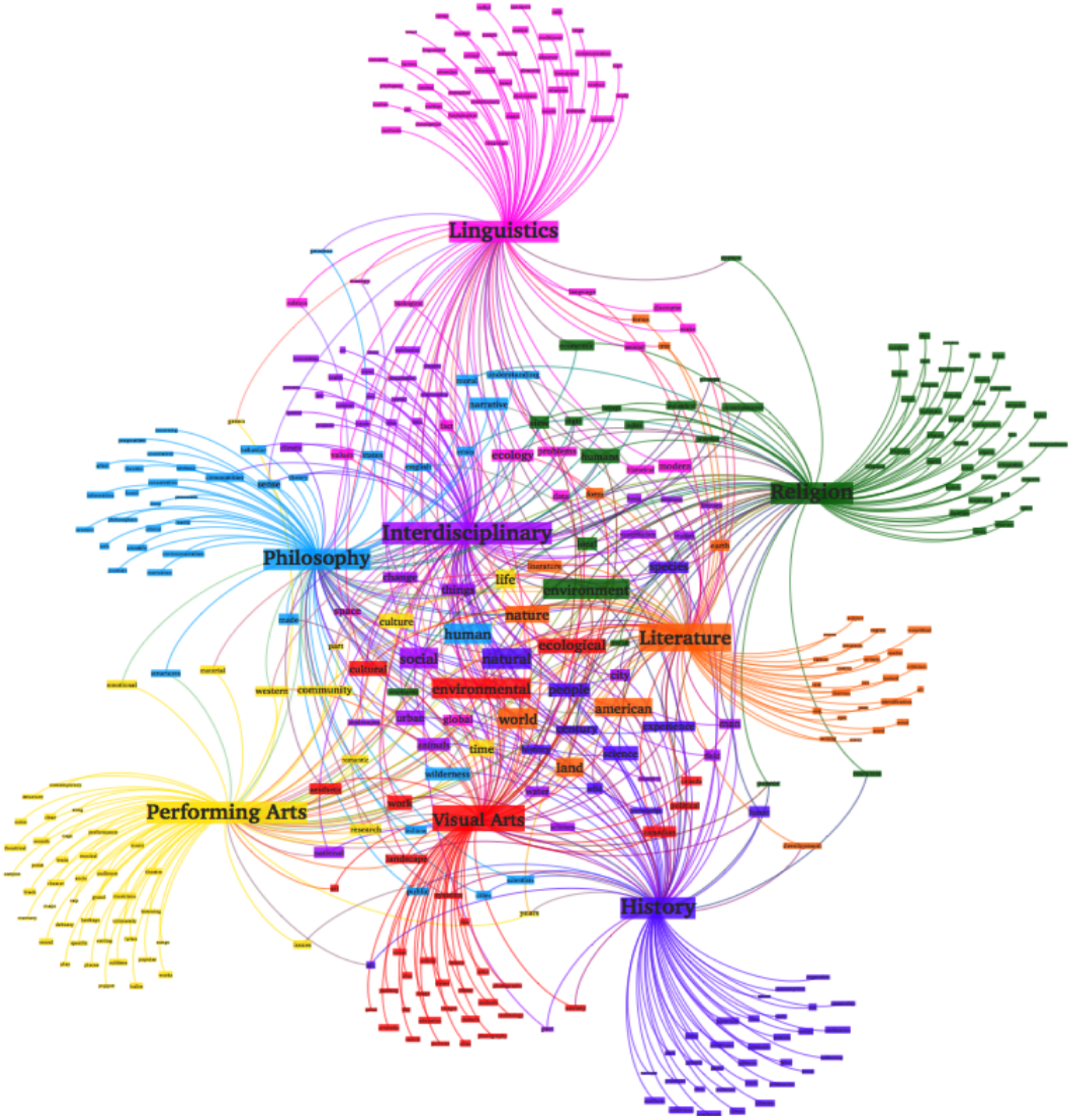
15:45 I luv my <3 iphone & you’re awsm apple, love you 3XXX. DisplayIsAwesome, sooo happppppy 🙂 🙏 http://www.apple.com #apple @sjobs

- **Cleaning process –**
  - HTML decoding
  - Expanding Contractions
  - Removing URL, Emoji, Reserved words, Smiley, User-mentions (or replace), hashtags
- **Preprocessing before tokenization**
  - Remove punctuation, space, stop word

Preprocessing Tweets

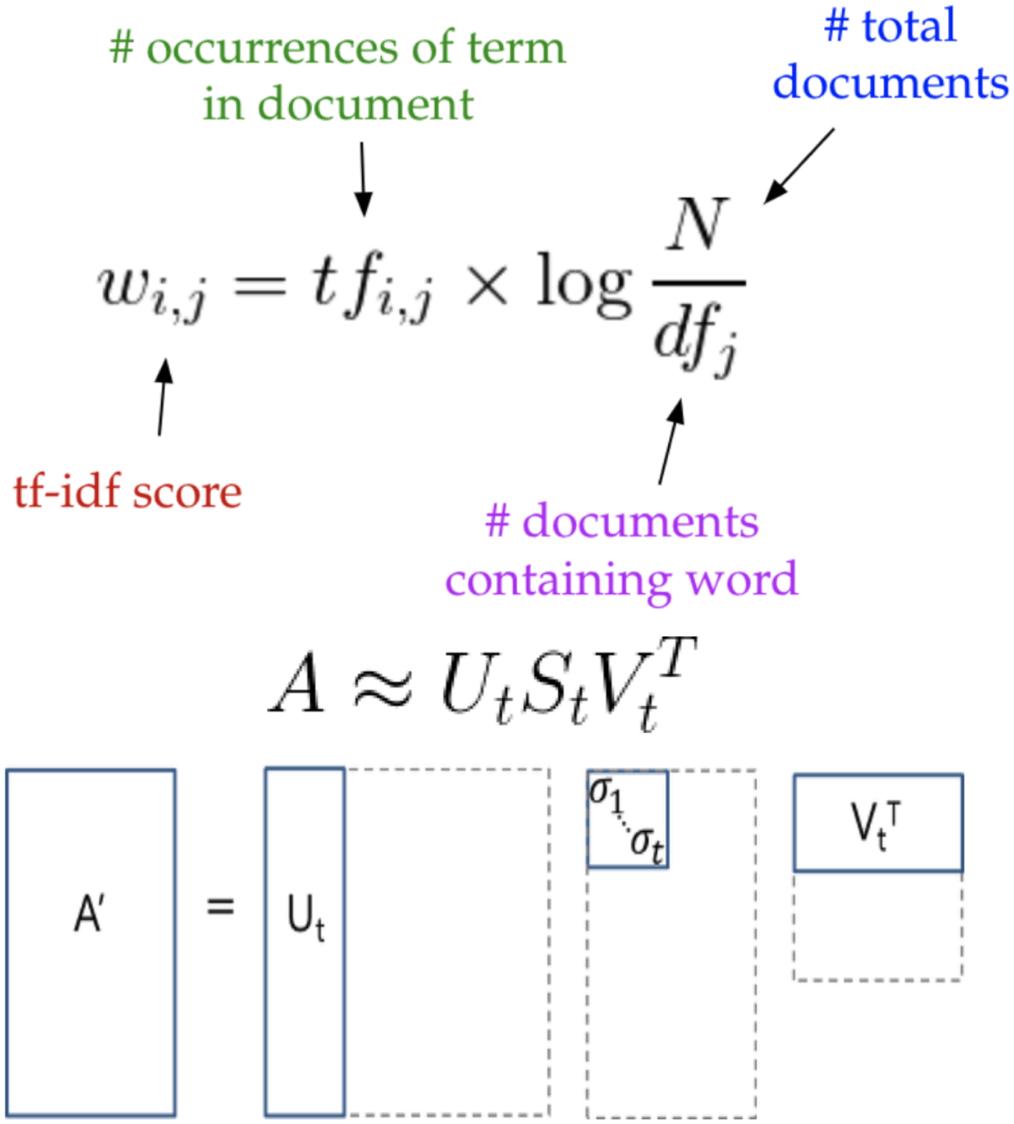
- **Future tasks**
  – Normalization of Noisy Text
  – **Awsm -> awesome, luv -> love**
  – Methodologies
    i. Lexical normalization
    ii. Normalization with edit scripts and recurrent neural embeddings
    iii. Find balance between precision and recall

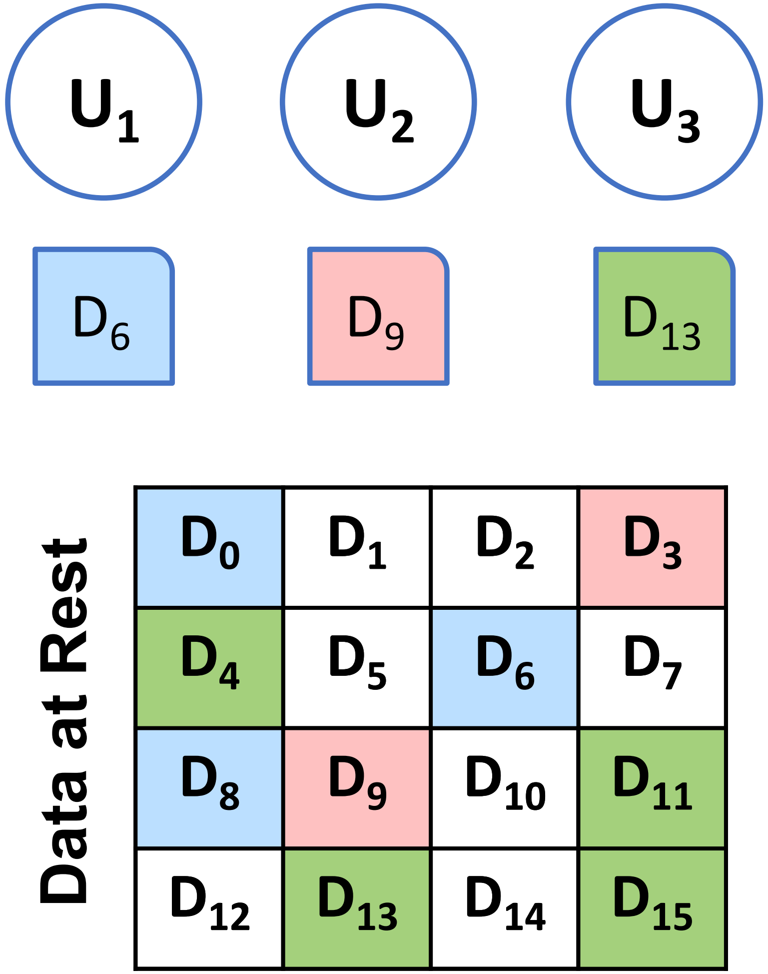
Feature Extraction



Feature Extraction: Topic Modeling

- **Latent Semantic Analysis**, or **LSA**
  - Find document-term matrix with tf-idf
  - Topics are latent
  - Dimensionality reduction with SVD, gives our term-topic matrix
- Apply **cosine similarity** to evaluate:
  - the similarity of terms (or “queries”) and documents (we want to retrieve passages most relevant to our search query).





Finding text data
similar to user-query
DEMO
File: relevant-doc-query-restful.mp4

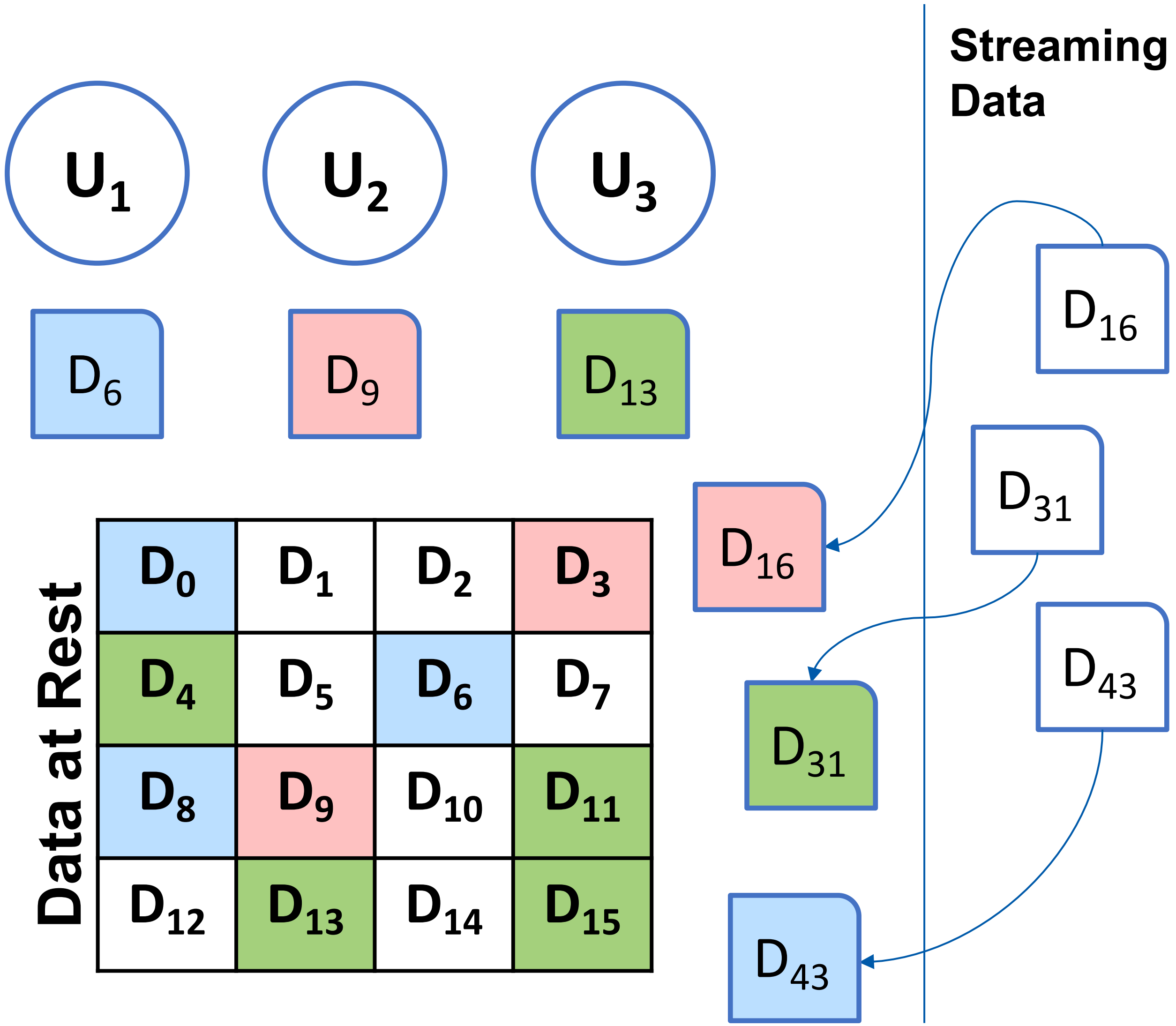
https://github.com/purdue-gask

Feature Extraction: Ongoing Research

- Even though LSA *finds* similar documents to user query, it has *less efficient* representation for topics.
- **Topics** are necessary for **ontologies** while **building our knowledge graph**
- **LDA** (Latent Dirichlet Allocation)
  - Generative Model
  - Uses **Dirichlet priors** for the document-topic and word-topic distributions
  - **Results in better generalization for new documents**
  - Allows online learning

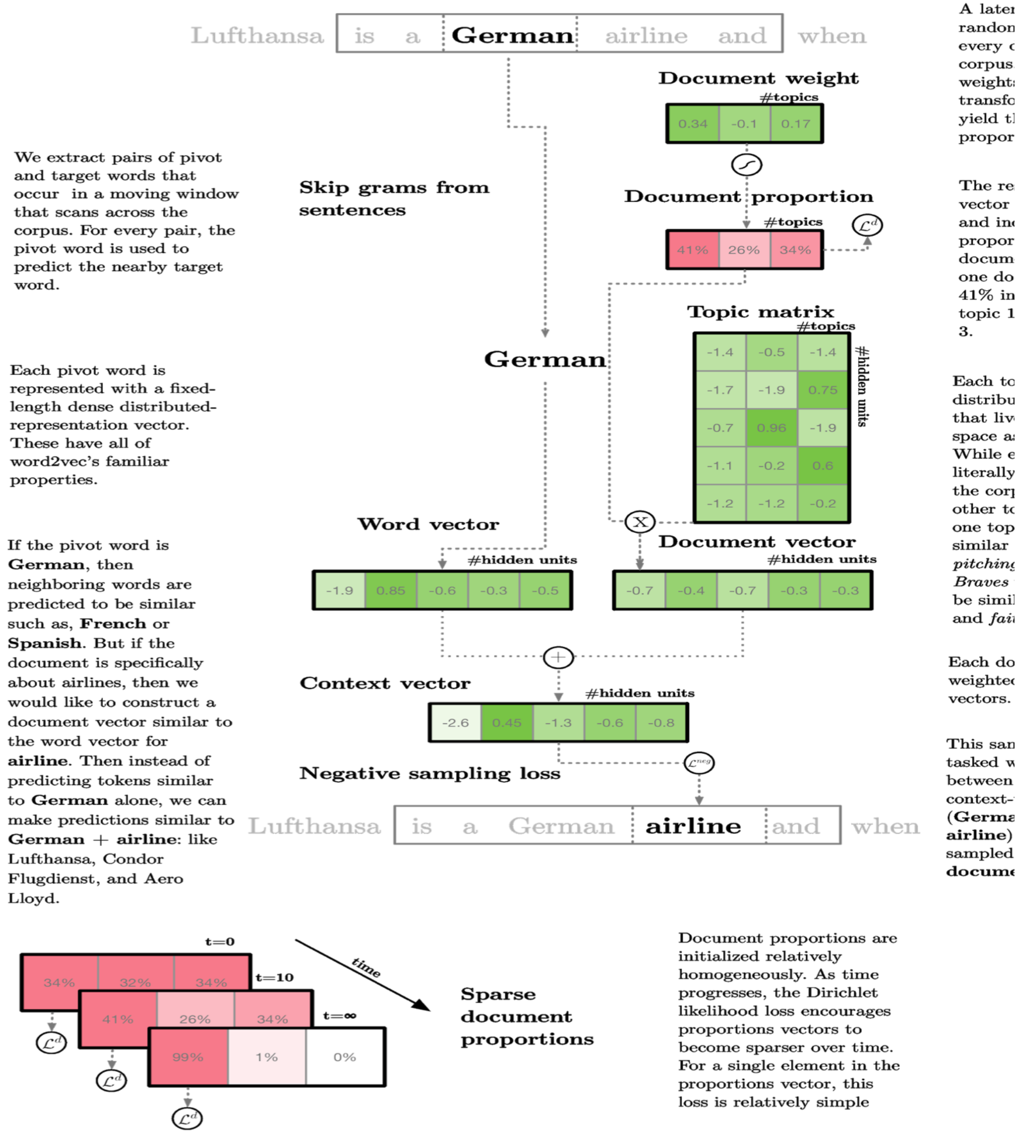
Feature Extraction: Ongoing Research

- Extract human-interpretable topics from a document corpus
- Each topic is characterized by words most strongly associated with.
- documents as **mixtures of topics** that spit out words with certain probabilities.

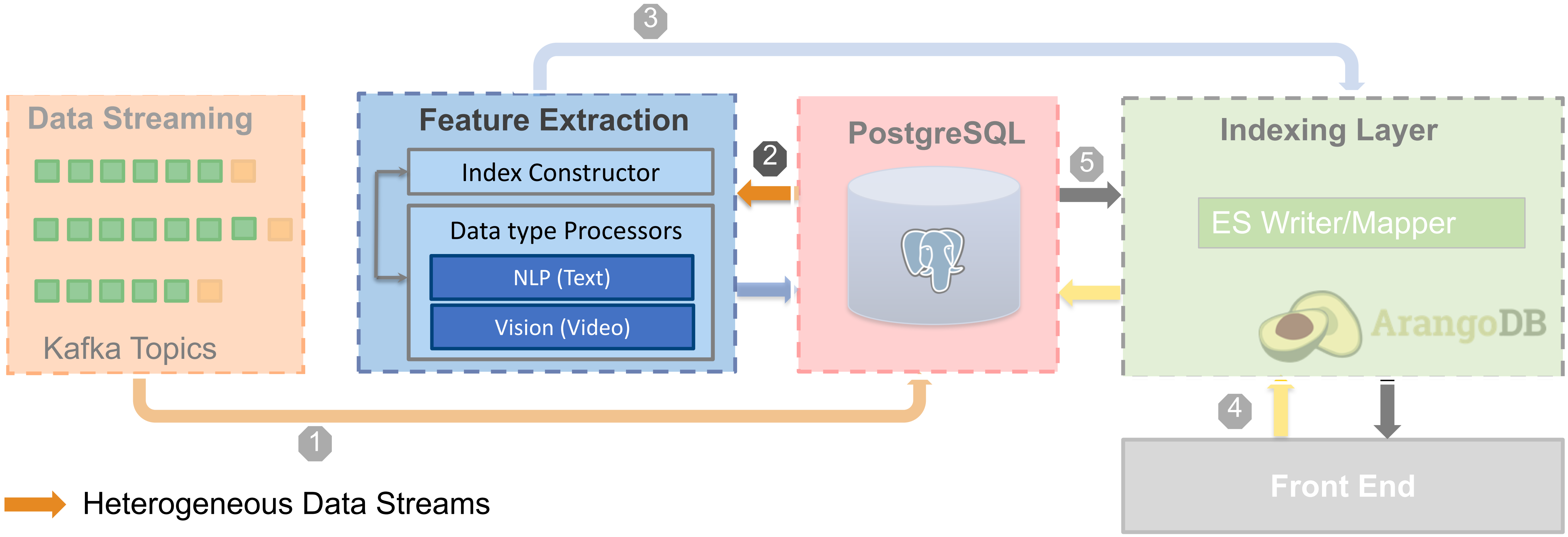


Information Retrieval : Ongoing Research

- **Deep Learning model: lda2vec**
- With lda2vec, leverages a ***context vector*** to make the predictions.
- Context : sum of the **word vector** and the **document vector**.
- learns word embeddings (and context vector embeddings) for words, and topic representations and document representations



Feature Extraction Module



Heterogeneous Data Streams

Situational Aware Indexed Data

Users' queries

Relevant patterns of data

Feature extraction from videos using transfer learning

Video Datasets

- Video represents a new modality of the data, essentially different from the text and structured numerical dataset
- **100+ hours** of *dashcam video* collected at MIT which is being further explored together with *tweets* from the same vicinity and Cambridge public datasets of structured data
- **Raw video** can be retrieved from MIT database:
  - Split into chunks of 30 seconds
  - Metadata collected: geolocation and timestamp for each 30 seconds video file
  - Metadata and videos are stored in different tables

Extracting Information Through Transfer Learning

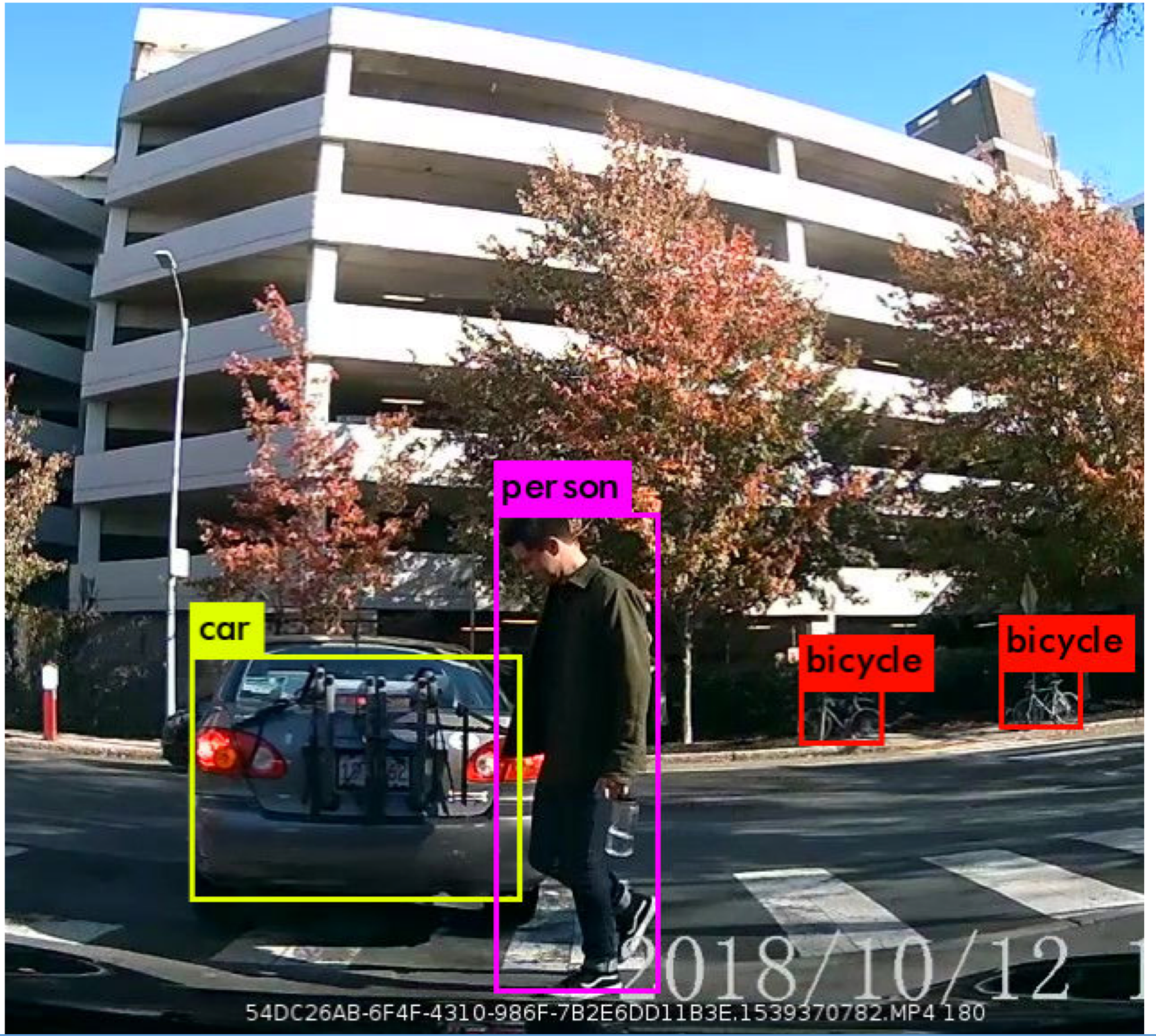
- **Object detection and classification**: best result achieved with deep learning architectures:
  - Faster RCNN
  - YOLO
  - SSD
- Manual annotation and labeling
  - **Time-consuming** and expensive for large datasets
  - Outsourced human labor can be employed (**MTurk**)
- We use *pre-trained* **YOLO** neural network to extract knowledge, detect and label objects in video

Neural Network For Object Detection and Classification

- Detects **100+** classes
- Our raw video dataset contains about 15 of the objects from these classes

Neural network object detection algorithm

1. Regions of interests/**region** proposals are generated
2. For each region, features are extracted and classified with *Convolutional Neural Network*
3. Apply non-maximum suppression: all candidate regions where probability of certain object detection is not max are dismissed



YOLO (You Only Look Once) v3 Neural Network Architecture

1. The image is split into an *SxS* grid of cells.

2. Each grid predicts *B* bounding boxes with *C* class probabilities

- *SxSxBx5* outputs in total

3. Conditional class probabilities are predicted *Pr(Class(i)/Object):*

- *SxSxC* class probabilities
- *SxSx(B*5+C)* output tensor
- *S=7, B=2, C=20 => (7,7,30)*
- Train a CNN to predict (7,7,30) tensor

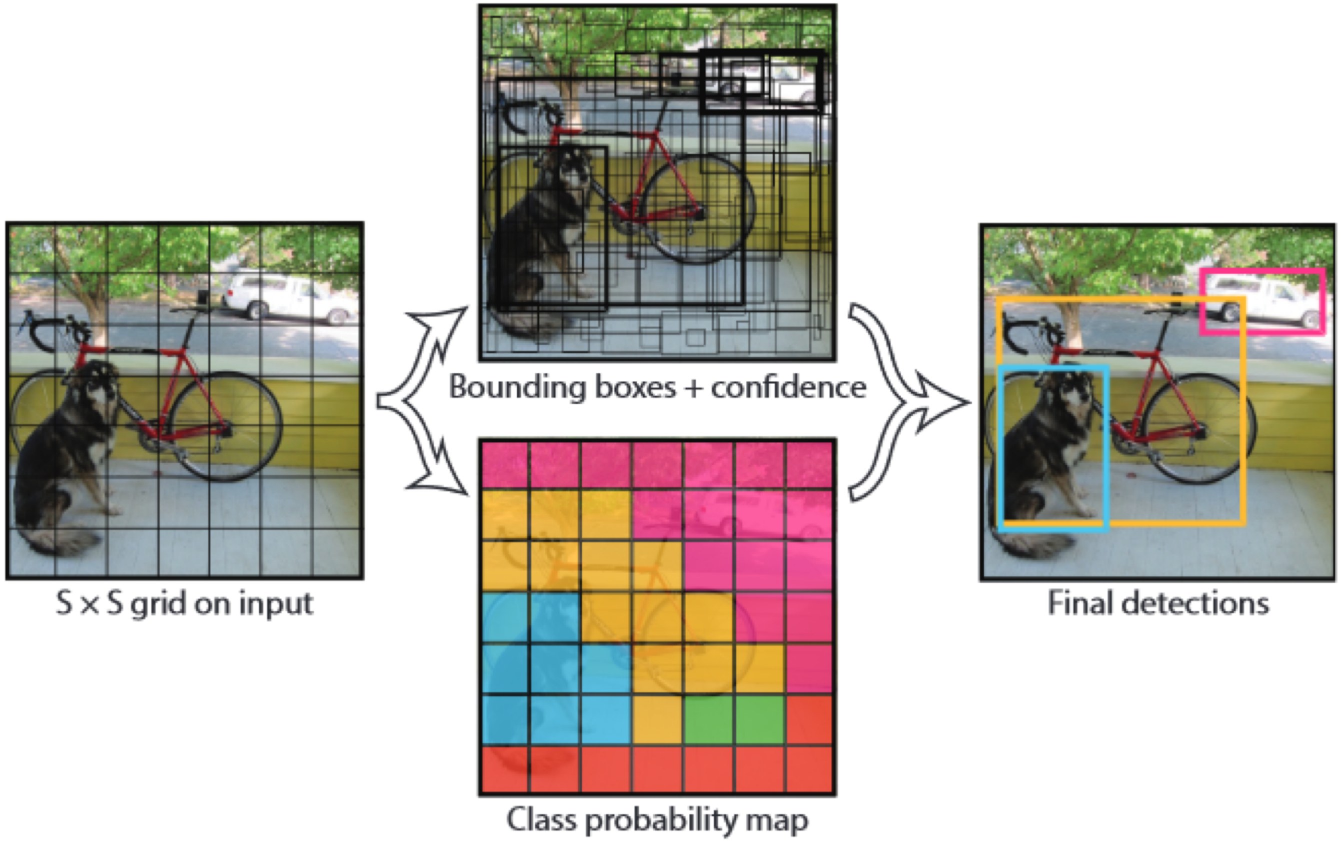


Image source: **You Only Look Once: Unified, Real-Time Object Detection**
[Joseph Redmon](), [Santosh Divvala](), [Ross Girshick](), [Ali Farhadi]()
https://arxiv.org/abs/1506.02640

Detected Classes In the MIT Video Dataset

CAR

TRUCK

PERSON

BICYCLE

TRAFFIC LIGHT

STOP SIGN

FIRE HYDRANT

PARKING METER

… AND MORE!

Knowledge Extraction From MIT Video Data

Frame-by-frame recognition for every 1 minute video

Saving: FrameID, list of objects' bounding boxes coordinates, list of corresponding object tags, confidence level for each object, list of unique objects per frame and per video up to the current moment

Each frame is saved in JPEG format with the bounding boxes

Saving the extracted information with the video and frame IDs as indices

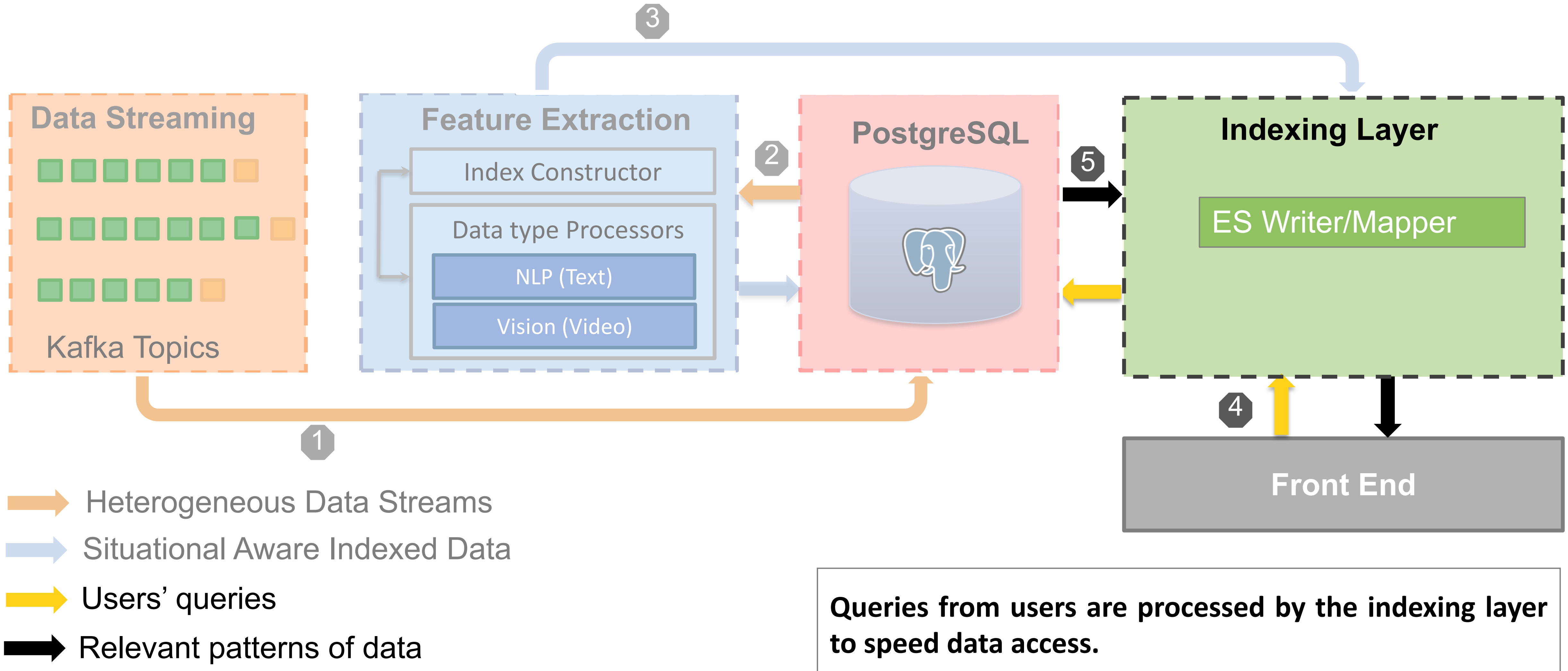
JSON Data Schema

```
{
 "id": "FRAME_ID", //String
 "name": "VIDEO_NAME/TITLE", //String
 "shortDescription": "SHORT_DESCRIPTION", //String
 "description": "LONG_DESCRIPTION", //String
 "originalVideoUrl": "VIDEO_ORIGINAL_URL", //String
 "storageType": "STORAGE_TYPE", //String
 "encodingProfileId": "ENCODING_PROFILE_ID", //String
 "listObjectsFrame": [], //Array of Unique Objects
 "listObjectsVideo": [], //Array of Unique Objects
 "clickUrl": "CLICK_VIDEO_URL", //String
 "supportedPlayerTypes": ["SUPPORTED_PLAYER"], //Array of strings
 "objectTags": {} //Object of tags, includes confidence level
 "category": [ "CATEGORY_TAG” ], //Array of strings
 "keywords": [ "KEYWORD” ], //Array of strings
 "copyright": "COPYRIGHT", //String
}
```

Indexing Layer and Front End

Elasticsearch and Graph-based Solution

Indexing Layer and Front End

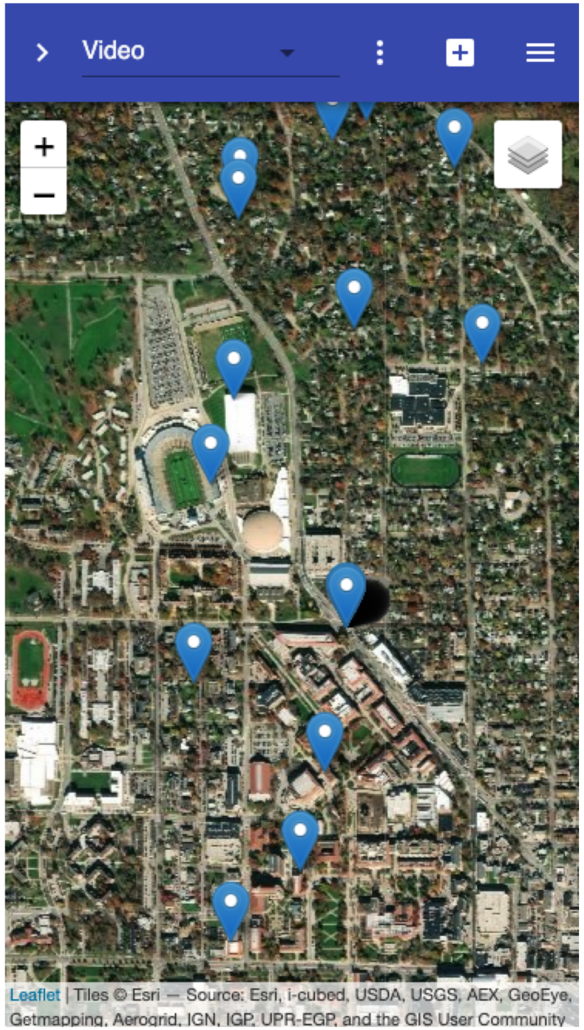


Queries from users are processed by the indexing layer to speed data access.

SKOD – Framework

Summary

- Extract data (text, video, sound) from Heterogeneous Sources and expose data via **Apache Kafka Topics**
- Consume data from Apache Kafka Microservice and populate the **RDBMS** and the Index Layer (**Elasticsearch** and *Graph Database)*
- Utilizing geolocation (**GeoJSON**) of text data (**Twitter**) visualize real-time streams on Leaflet map
- Analyze data relationships through graph analytics (**clustering**)



kafka

Trellis
www.trellisframework.org

React

JS

docker

JWT

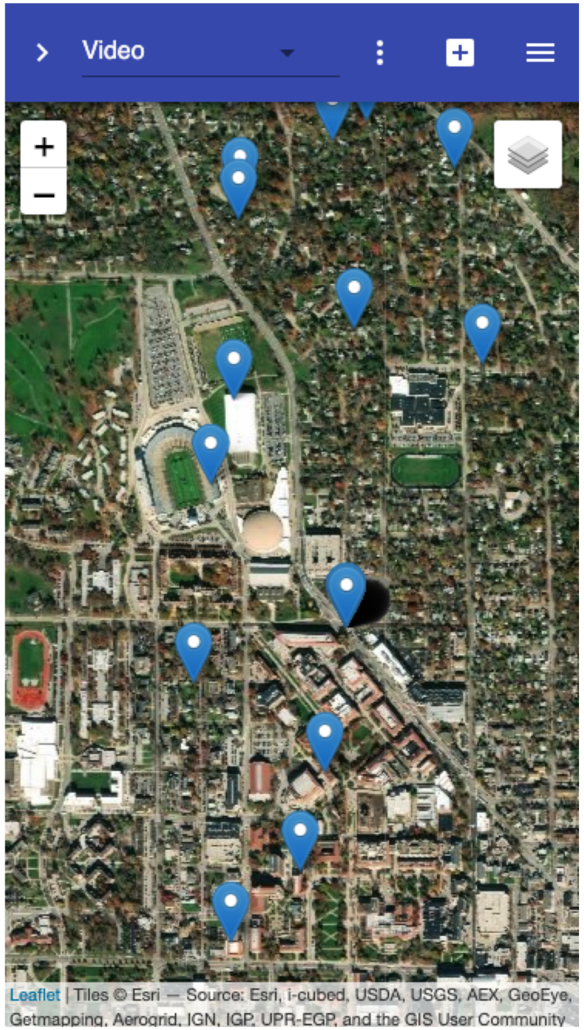
We utilize the OADA/Trellis framework to build the PoC of the Web App.

SKOD – Framework

Features

- Open source @
- Distributed Compute Engine (Apache Spark **GraphX**) and **Motif** analysis
- ArangoDB Graph Database
- **Multiple layers of Cache** (PouchDB) [PSB+19] @
- Easy to setup (using Docker containers)
- **React/Cerebral** based Analytics Web-UI

@ https://github.com/purdue-gask/skod/



@ https://github.com/OADA/oada-cache

kafka
Trellis www.trellisframework.org
React
JS
docker
JWT

Concluding remarks

- There are numerous **users** with different **missions**
- **Missions** with various **needs** for information
- **SKOD** is an end-to-end system to empower such **users** with **relevant knowledge** from *streaming* or *stored* data
- **SKOD** demo in October at **Northrop Grumman Tech Fest**

https://github.com/purdue-gask

https://github.com/OADA

References

[KBZ18] Kang, D., Bailis, P., Zaharia, M.: Blazeit: Fast exploratory video queries using neural networks. CoRRabs/1805.01046(2018).

[MVK17] Meditskos, G., Vrochidis, S., Kompatsiaris, I.: Description logics and rules for mul-timodal situational awareness in healthcare. In: MMM (1). Lecture Notes in Computer Science, vol. 10132, pp. 714–725. Springer (2017)

[AHD+15] Adjali, O., Hina, M.D., Dourlens, S., Ramdane-Cherif, A.: Multimodal fusion, fission and virtual reality simulation for an ambient robotic intelligence. In:ANT/SEIT. Procedia Computer Science, vol. 52, pp. 218–225. Elsevier (2015)

[FFV15] Foresti, G.L., Farinosi, M., Vernier, M.: Situational awareness in smart environments: socio-mobile and sensor data fusion for emergency response to disasters. J.Ambient Intelligence and Humanized Computing6(2), 239–257 (2015).

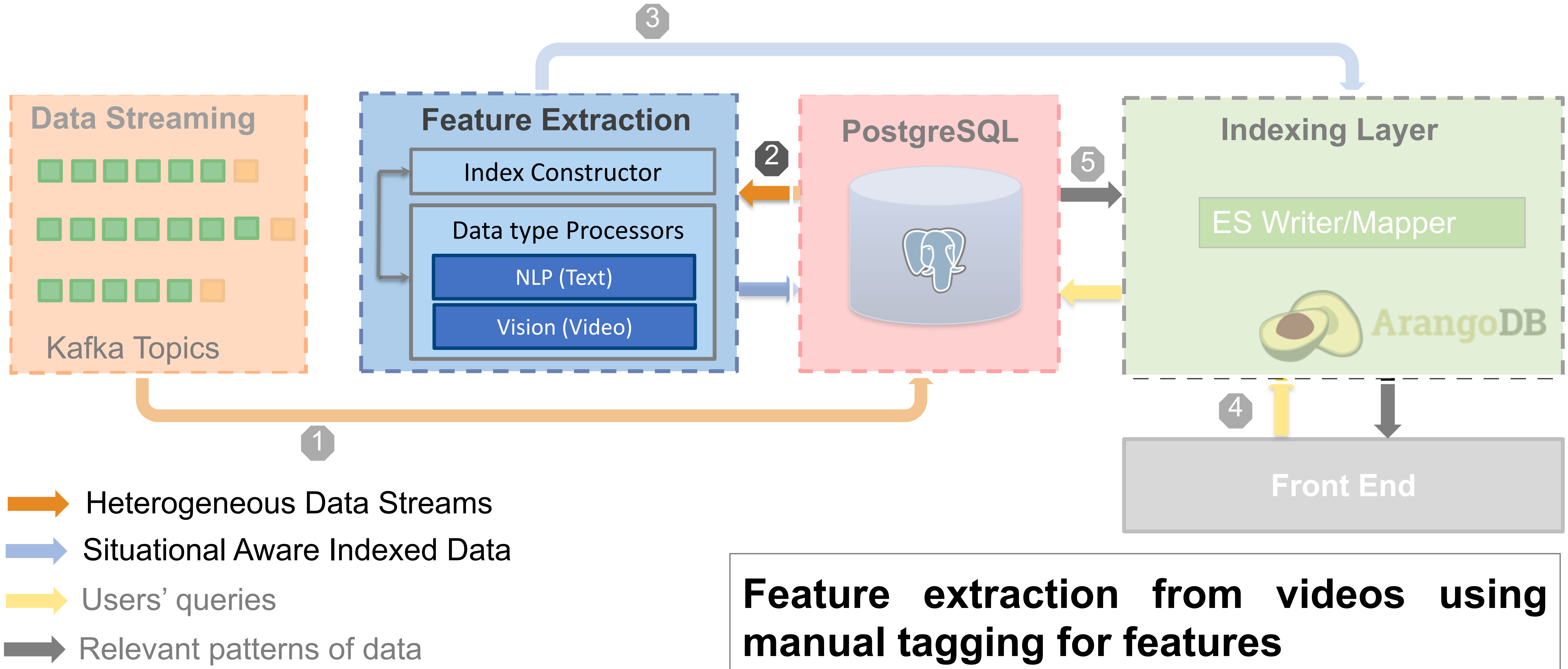
[PSB+19] Palacios, S., Santos, V., Barsallo, E., Bhargava, B.K., (2019). MioStream: A Peer-to-Peer Distributed Live Media Streaming on the Edge. Proceedings of the Multimedia Tools and Applications (MTA). Published.

[PAL+19] Palacios, S., Ault, A., Layton, A., Krogmeier, J., Buckmaster, D., Bhargava, B.K., (2019). Trellis++: A Practical, Auditable, Privacy-Preserving Certification Framework with Oblivious Smart Contracts. Proceedings of the Transactions ASABE. In Preparation.

[PSA+19] Palacios, S., Solaiman, KMA, Angin, P., Nesen, A., Stonebraker, M., Bhargava, B.K., (2019). SKOD: A Framework for Situational Knowledge on Demand. Proceedings of the VLDB Workshop POLY. Accepted.

Backup slides

Feature Extraction Module



Feature extraction from videos using manual tagging for features

Manual Feature Extraction from Videos

- Features targeted
  - Objects in Video
  - Attributes of the objects
- Amazon Mechanical Turk (Mturk)
  - For task design
  - For annotation collection
  - For task distribution
- Steps
  - Run Object detection algorithms
  - Segment video into frames
  - Modify the existing annotations

Task Design Sample: Instance Segmentation

Instructions ×

View full instructions

View tool guide

Use the tools to label each instance of the requested items in the image

Color in each instance of the requested items in the image

Labels ×

Choose a class below to add its instance(s).

▶ **Car**

▼ **Fire Hydrant**

Fire Hydrant #1 1

Add instance

▶ **Turn signals**

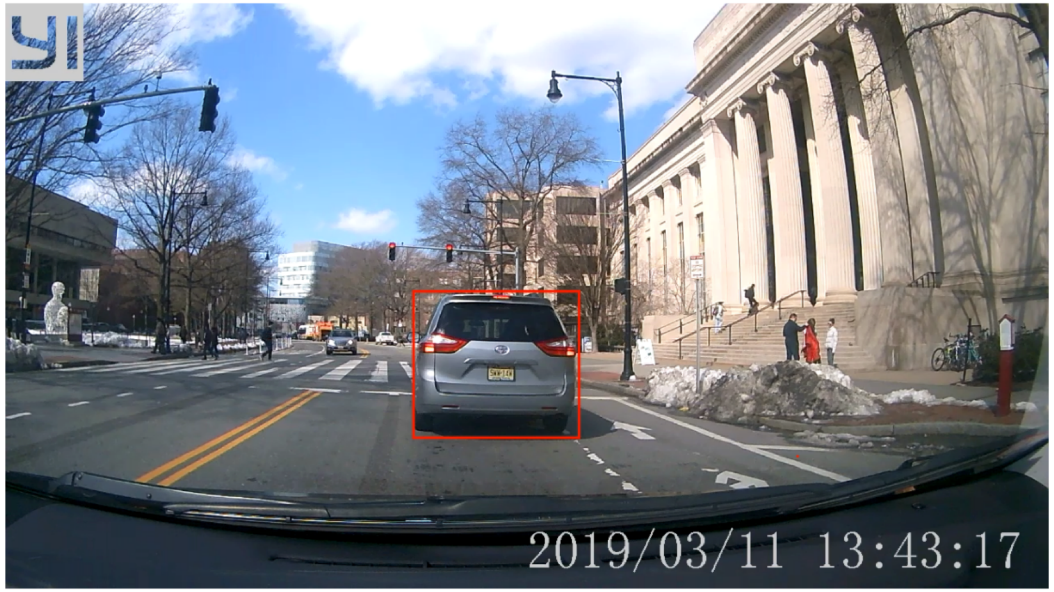
Polygon Brush Eraser Dimmer Undo Redo Zoom in Zoom out Move Fit image

☐ Nothing to label **Submit**

Task Design Sample: Attribute Tagging

Instructions: Given a frame, describe the attributes of the marked object in the bounding box.

Attributes can include number plate, color of car, street name that can be used to describe the object.



Word/phrase 1

Number plate/SWW-14W

Word/phrase 2