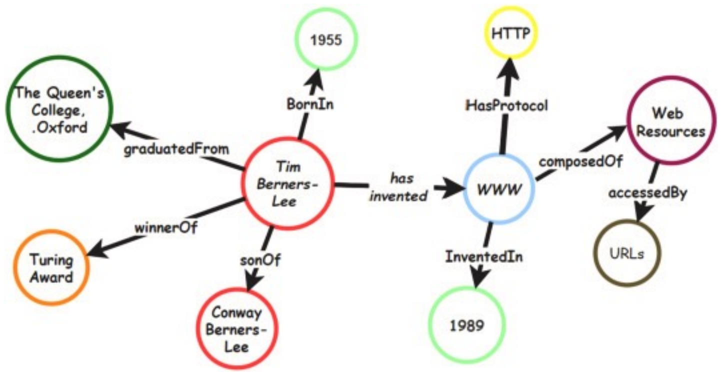
Stream2Graph: Dynamic Knowledge Graph for Online Learning Applied in Large-scale Network

데이터사이언스학과 빅데이터 관리 및 응용 연구실
석사과정 김민선

2023-07-05

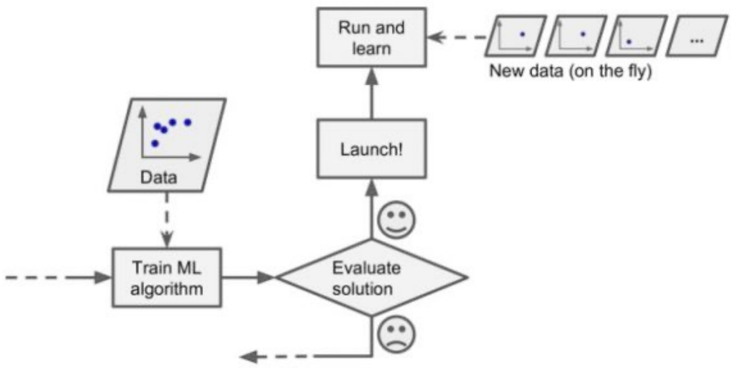
Preliminaries

- Knowledge Graph



> ❝
> *…a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent potentially different relations between these entities.*

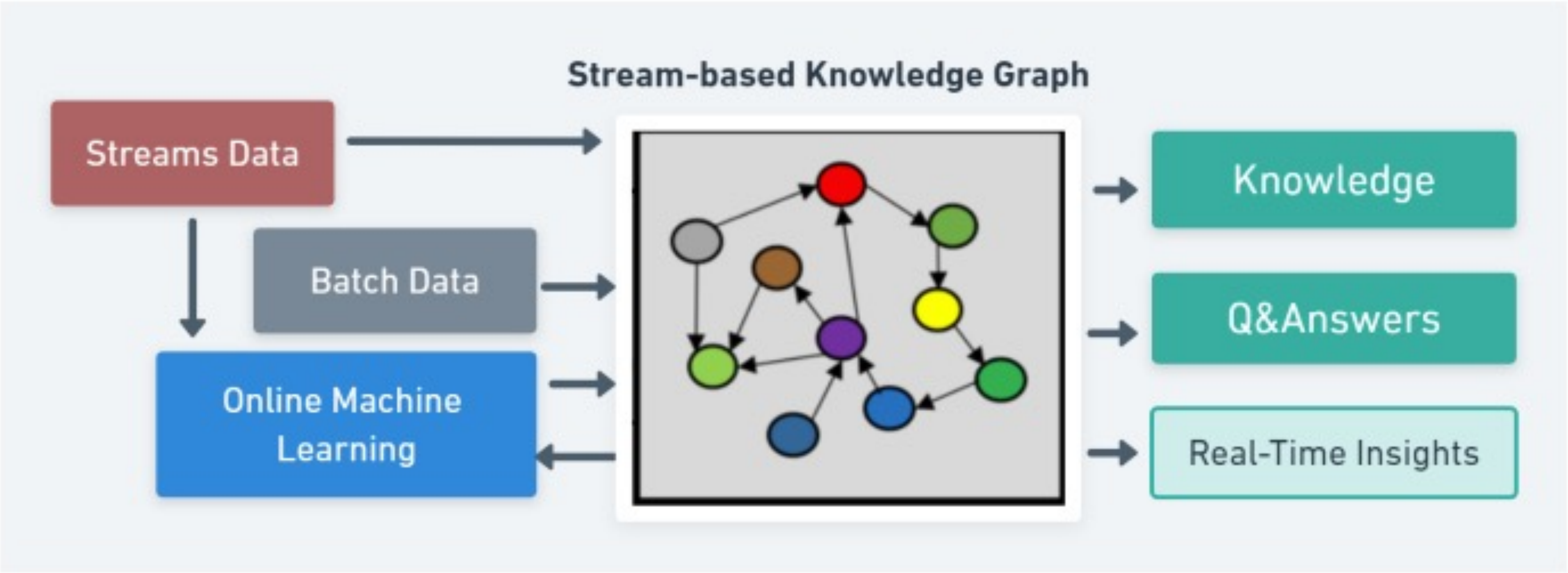
- Online Learning



> ❝
> *… to maximize the accuracy of the sequence of predictions made By the online learner, given the knowledge of correct answers to Previous prediction tasks and possibly additional information*

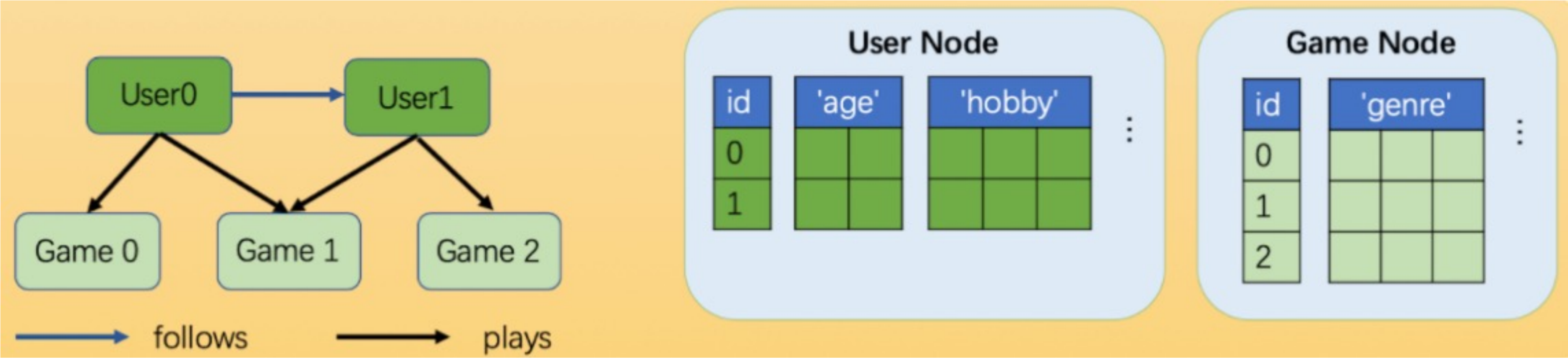
Stream2Graph

A domain-agnostic system to easily build and operationalize stream-based knowledge graphs and combine them with online learning applications



Desiderata

1. Heterogeneous data
2. Training and deployment of predictive models on evolving data
3. Data pipelines for updating and maintaing the KG

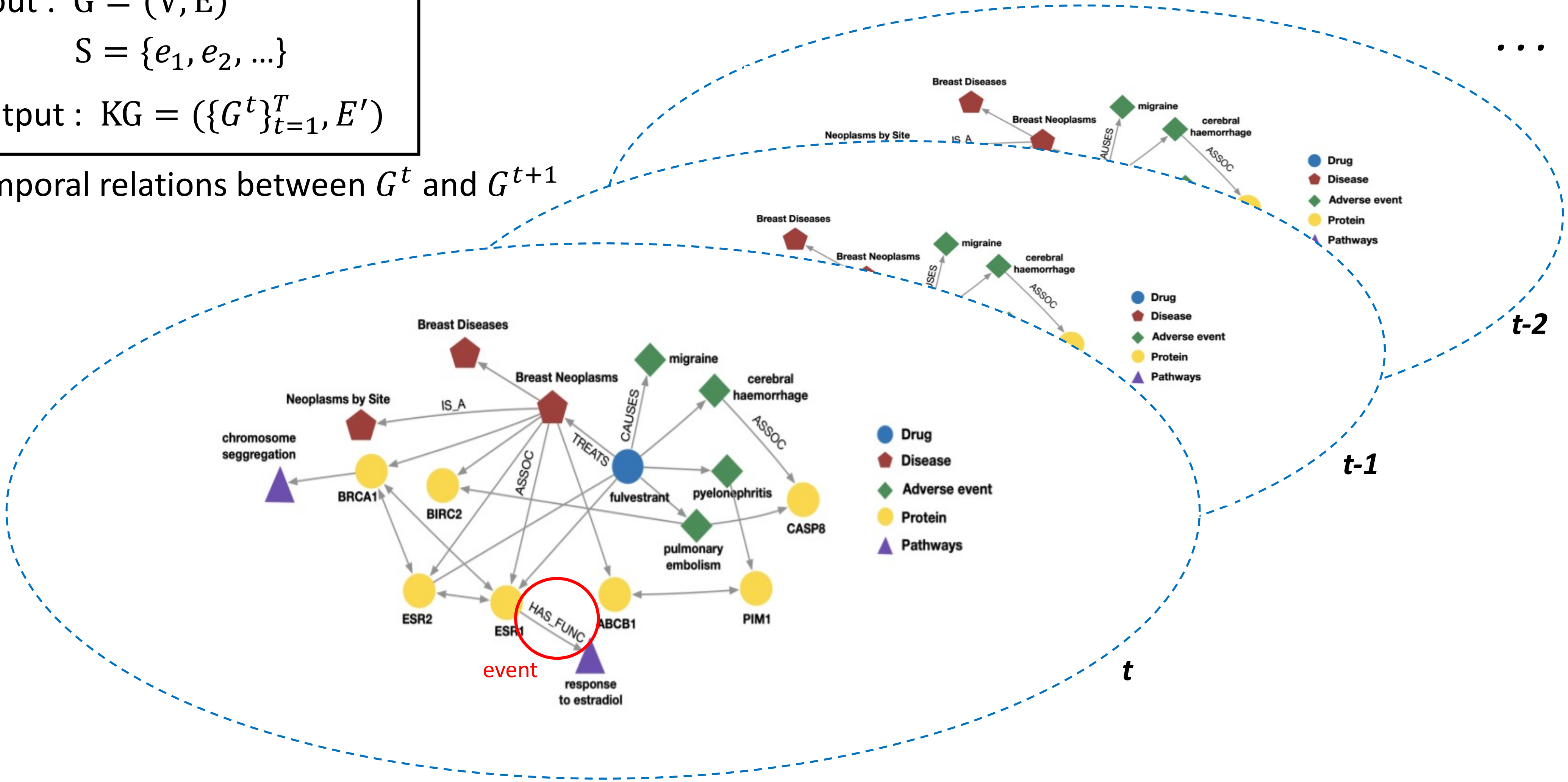


** Heterogeneous graph*

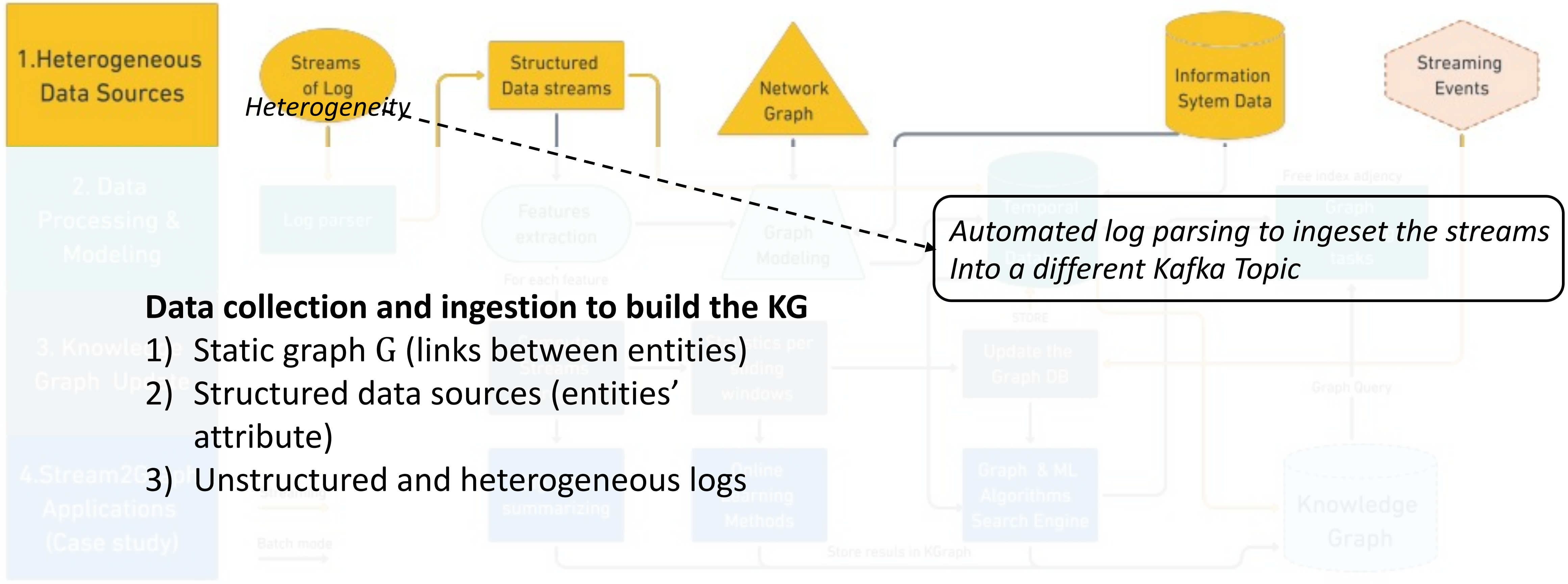
Problem definition: Graph

- Input : $\mathrm{G} = (\mathrm{V}, \mathrm{E})$
  $\mathrm{S} = \{e_1, e_2, \ldots\}$
- Output : $\mathrm{KG} = (\{G^t\}_{t=1}^{T}, E')$

E' : temporal relations between G^t and G^{t+1}
$e_i \in E$



1. Knowledge Collcetion from multiple sources



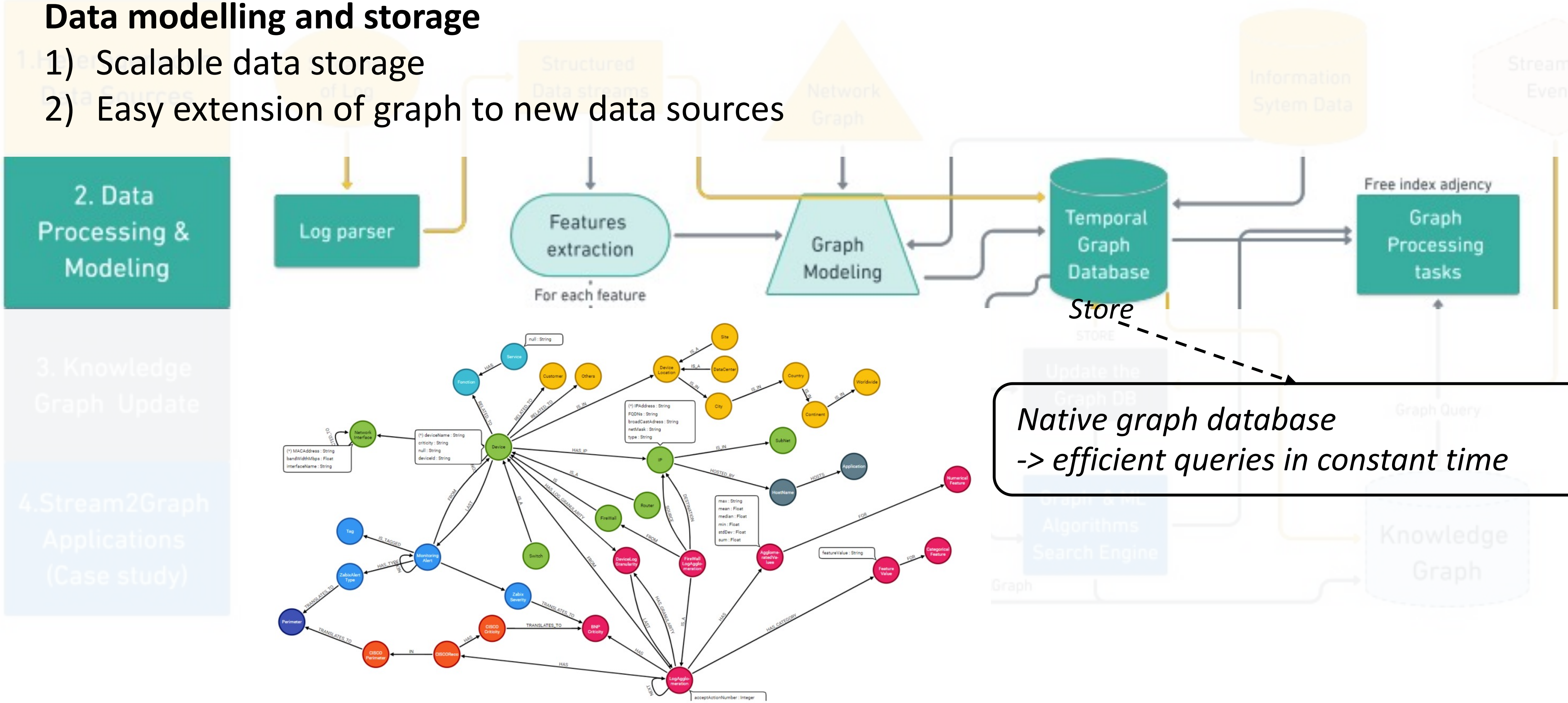
Data collection and ingestion to build the KG

1) Static graph G (links between entities)
2) Structured data sources (entities' attribute)
3) Unstructured and heterogeneous logs

2. Knowledge modelling, enrichment, and mapping

Data modelling and storage

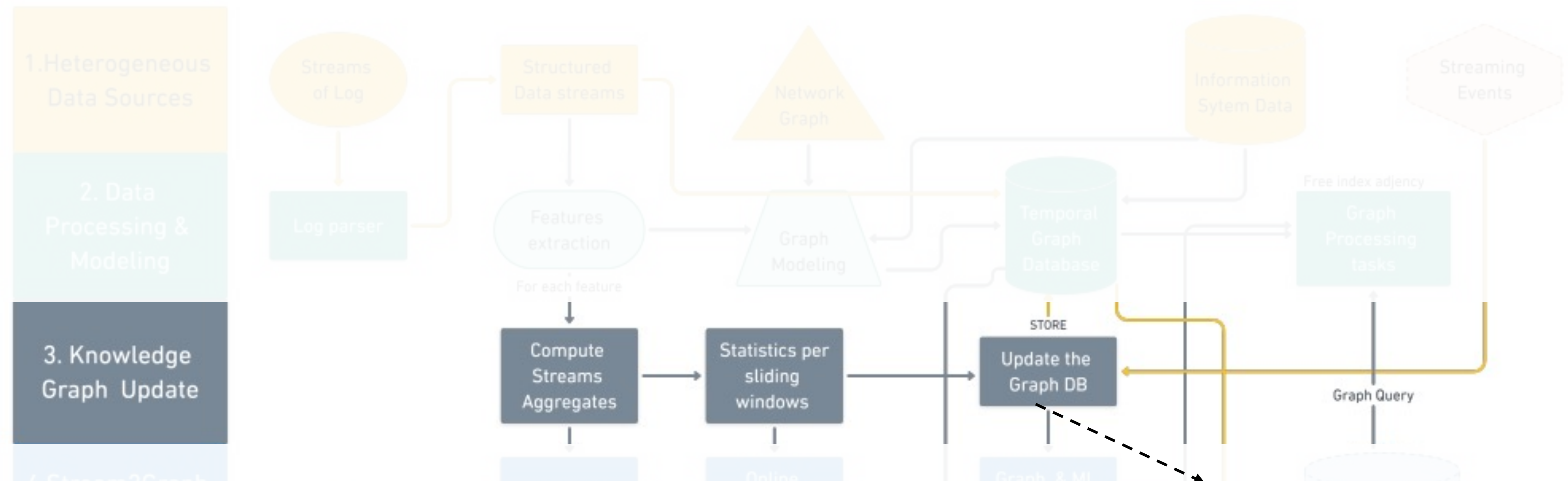
1) Scalable data storage
2) Easy extension of graph to new data sources



Store

Native graph database
-> efficient queries in constant time

3. Knowledge Graph incremental update

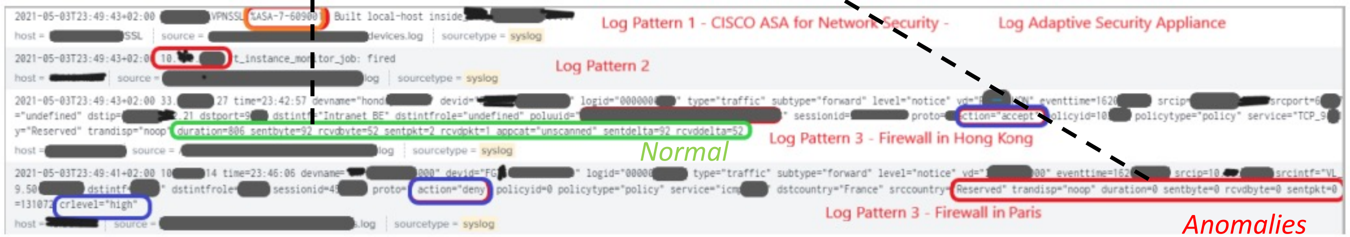


1) Extract features from event stream
2) Compute statistics and trends for each feature type
3) Update current graph with new nodes, edges and recent statistics

Incremental update
-> online learning settings

Online Knowledge Graph Update

- Compute feature vector for each node based on <u>categorical and numerical features</u> from raw log data



**Log patterns related to network traffic on devices*

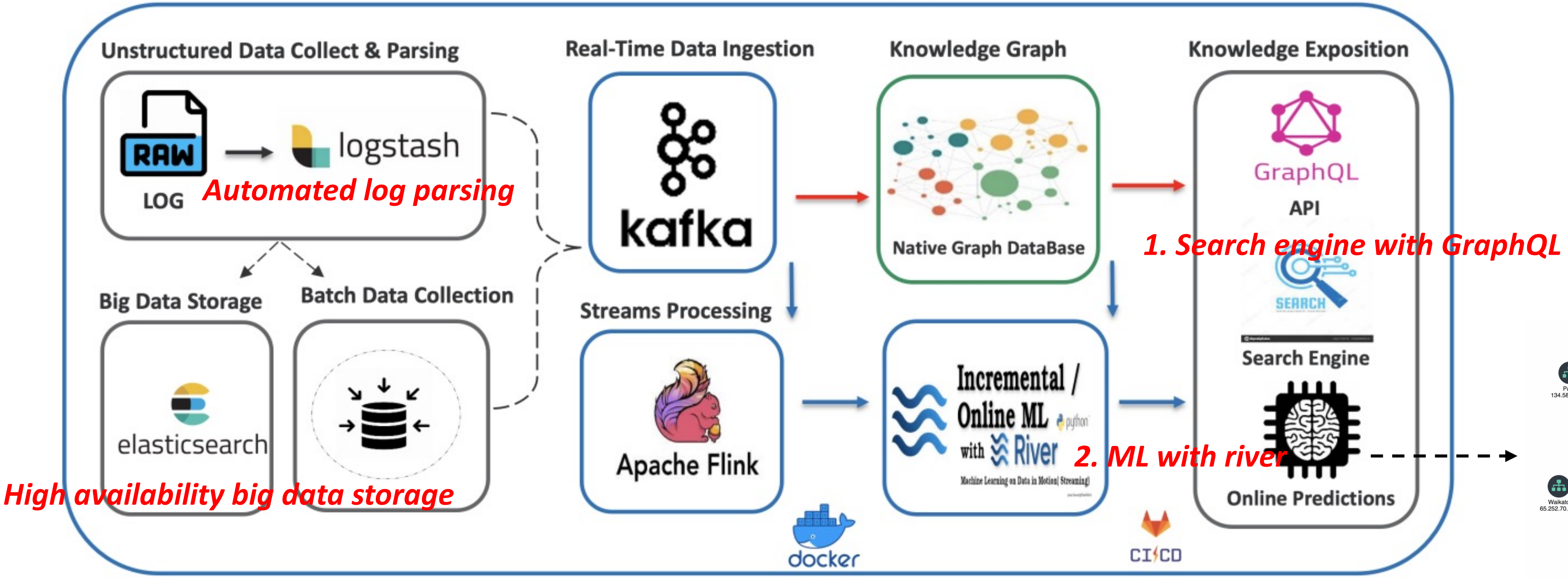
Algorithm 1: Online Knowledge Graph Update

Input: Event stream $S = \{e_1, \ldots, e_n \ldots\}$ over time t
Knowledge Graph $G^t = (V^t, E^t)$, multivariate vector r_i
u_i, v_i source and destination node of event e_i
η_i set of numerical attributes of $e_i \in G^t$,
$\tilde{\eta}_{ij}$ set of statistics of attribute η_{ij},
ς_i set of categorical attributes of e_i,
$\tilde{\varsigma}_{ij}$ count of attribute ς_{ij}, r_{ij} value of feature j in r_i

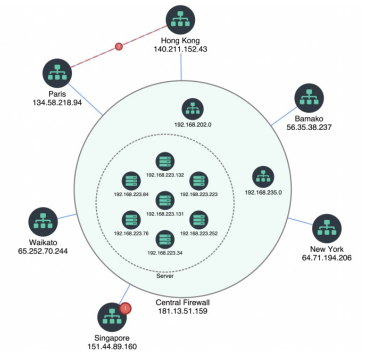
Output: Updated Knowledge Graph $\{G^t\}_{t=1}^{T} = (\mathscr{V}, \mathscr{E})$

```
while new record e_i = (u_i, v_i, r_i, t_i) arrives do
    V^{t_i} ← V^t ∪ u_i ∪ v_i
    for w_ij in e_i do
        if IsNumericalAttribute(r_ij)
            η_i ← η_i ∪ r_ij
        else if IsCategoricalAttribute(w_ij)
            ς_i ← ς_i ∪ r_ij
    ▷ Update V with numerical statistics
    for η_ij in η_i do
        η̃_ij ← ComputeStats(η_ij)
        V^{t_i} ← V^{t_i} ∪ η_ij
    ▷ Update V with categorical counts
    for ς_ij in ς_i do
        ς̃_ij ← UpdateCount(ς_ij)
        V^{t_i} ← V^{t_i} ∪ ς_ij
    E^{t_i} ← E^t ∪ e_i
    G^{t_i} ← Graph(V^{t_i}, E^{t_i})
    G^T ← G^T ∪ G^{t_i}
    ▷ Output updated graph G^T
```

4. Architecture & operationalization of a dynamic knowledge graph



Collection -> Ingestion -> Moelling -> Exposition

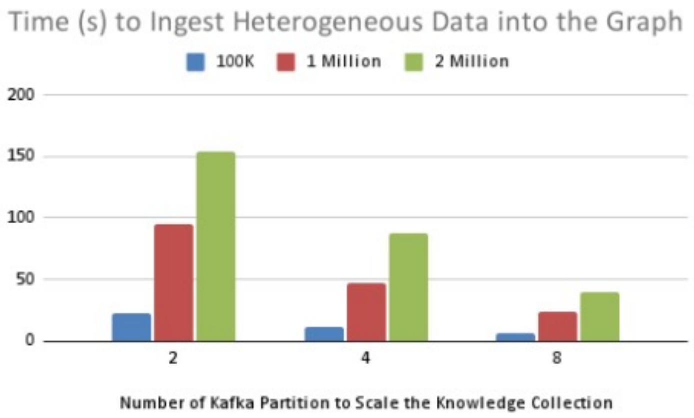


Detecting anomalies

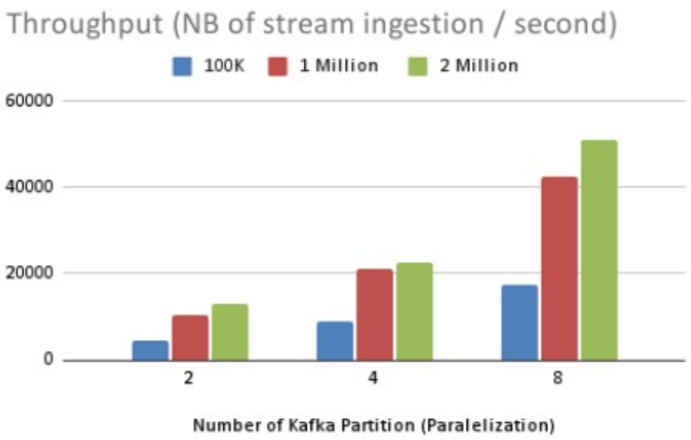
Experimental evaluation

1. Scalability in knowledge collection

-> Scale the number of instance-varying Kafka partitions from 2 to 8



(a) Knowledge ingestion Time scales linearly with the number of processing instances



(b) Throughput (Number of events/sec) scales linearly with parallelization.

Experimental evaluation

2. Latency in knowledge ingestion and enrichment to processing billions of events

-> 22.000 events/sec

(c) High-velocity Data (22,000 events/sec) from industrial big data (21 Billions of events emitted)

Experimental evaluation

3. Improvement of ML performance on high-velocity data streams

1. No redundant information
2. Low-dimensional while keeping relevant information

-> easier to train online ml model with time improvement (while preserving performance)

Table III: time (s) and ROC AUC for events classification on real data. Before (4 million instances) vs After (4,000 records aggregated /seconds) enriched with KG

	Raw network data		Data with KG Features		
	ROC AUC	Time (s)	ROC AUC	Time (s)	Speed up
ARF	0.93	10,623	**0.94**	**5.19**	2000×
AdaBoost	0.98	3,994	0.98	**7.29**	547×
HT	0.96	2,861	0.96	**1.46**	1900×