Navigating the Visual Fog: Analyzing and Managing Visual Data from Edge to Cloud

Ragaad Altarawneh, Christina Strong, Luis Remis, Pablo Muñoz, Addicam Sanjay, Srikanth Kambhatla

Intel

Why Visual Data

Motivation:

- Over 70 exabytes of live video expected to be flowing over the Internet by 2021
- Video analytics is becoming an increasingly important source for deriving actionable insights in various use cases like digital surveillance, retail analytics, factory automation, and smart cities.

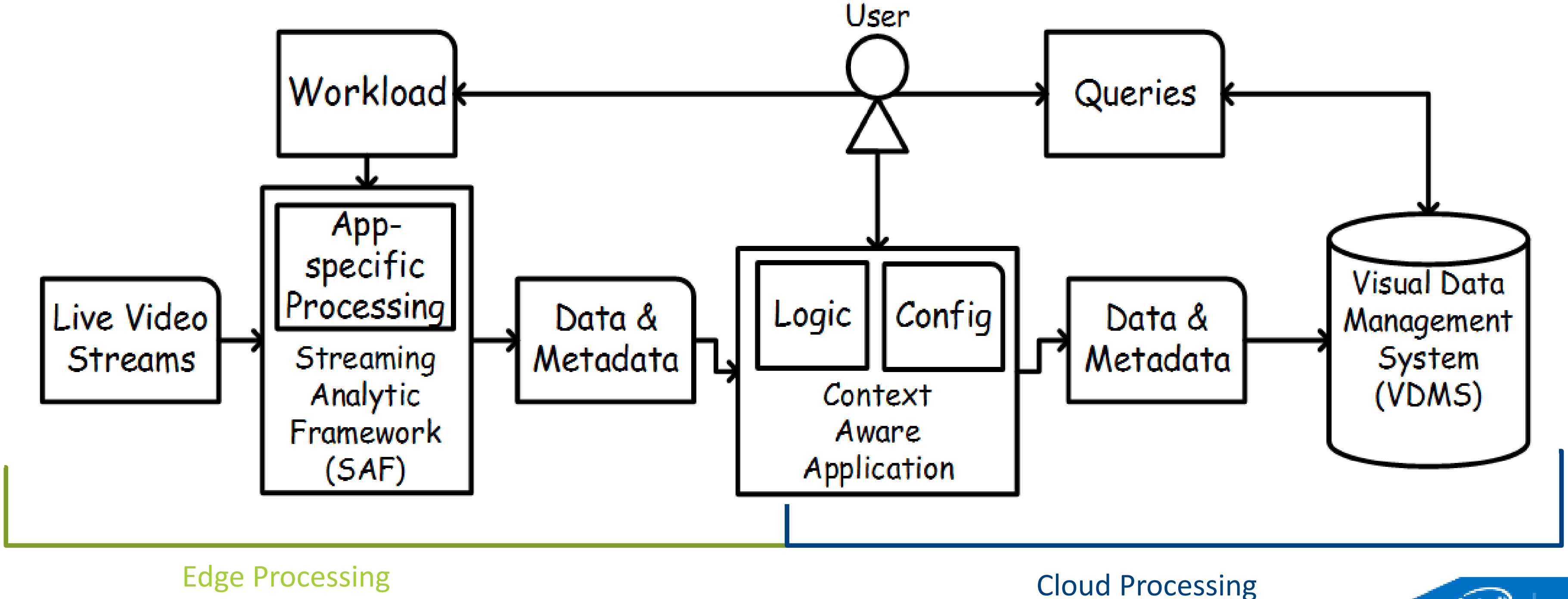
Challenges:

- Despite of its abundance and ubiquity, visual data is *still significantly under-utilized due to the computational requirements and the outstanding volume of collected data.*
- Today's solutions have made advances in converting video data into actionable information at scale, these solutions often have fixed function capabilities and are limited to the use of cloud infrastructure

Clouds vs Edge Processing (Retail Analytics as an example)

- Providing shopper insights requires far more functionality, as well as flexibility, than simply counting people (e.g. summarize a list of interesting areas for a certain group of people. E.g. females vs males).
- However, because of the limited bandwidth from edge data sources such as cameras, it is often not feasible to send all video streams to the cloud.
- While this presents a problem in using cloud infrastructure alone, it is also impractical to rely solely on edge devices, due to the significant computational resource requirements of visual data.
- Edge computing has the additional drawback of not being economically scalable, as having on-premise servers translates to recurring expenses to the users.

Edge-to-cloud design of E2E streaming data management framework.

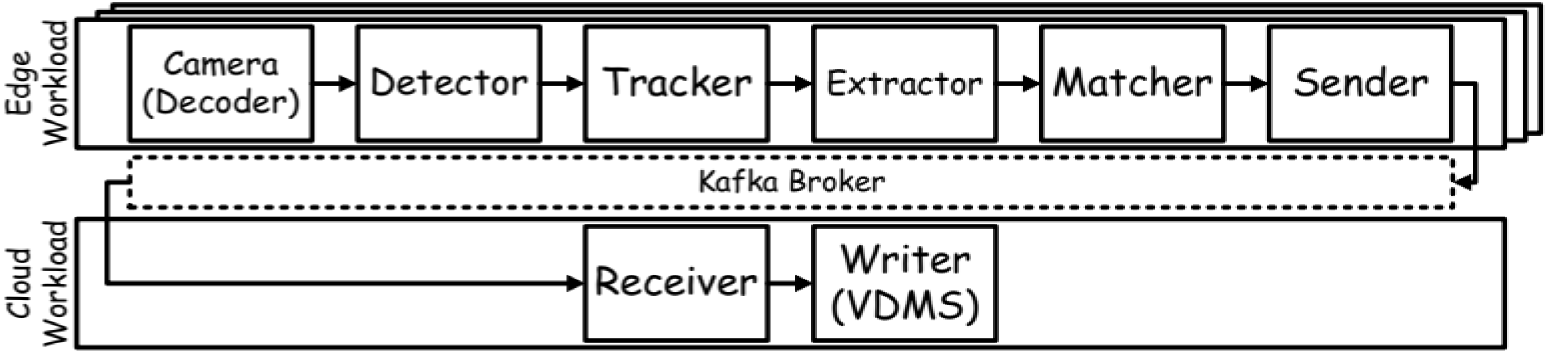


Edge Side : Streaming Analytics Framework (SAF)

https://github.com/viscloud/saf

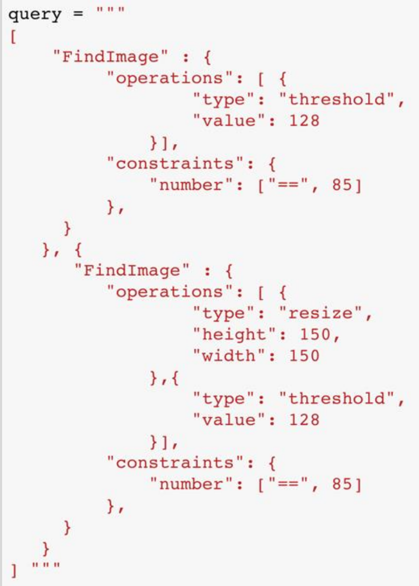
SAF is an open source framework for *creating* and *running* video analytics workloads that provides low-latency video processing by incorporating *edge-to-cloud* deployment with E2E scheduling.

- Bottom up programming for easy workload partitioning and scheduling
  - Processor modules based on OpenVINO (aka CVSDK), Caffe, TensorFlow, etc.
- Build on CVSDK, NCSDK, MKL, etc. for optimized Intel hardware utilization
-



Cloud Side: Visual Data Management System (VDMS)

https://github.com/IntelLabs/vdms



```
query = """
[
    "FindImage" : {
         "operations": [ {
                 "type": "threshold",
                 "value": 128
             }],
         "constraints": {
             "number": ["==", 85]
         },
     }
   }, {
      "FindImage" : {
         "operations": [ {
                 "type": "resize",
                 "height": 150,
                 "width": 150
             },{
                 "type": "threshold",
                 "value": 128
             }],
         "constraints": {
             "number": ["==", 85]
         },
     }
   }
] """
```

Efficient completion of complex metadata queries

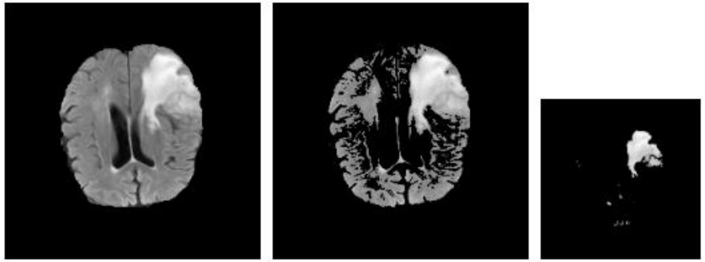
- Using our in-house Graph Database
- Metadata stored in (persistent) memory

Efficient visual data retrieval + pre-processing

- Images
  - Threshold, crop, resize, or basic augmentation on images on the server side
- Visual Descriptors
  - Similarity search performed on the fly
- Videos

Straightforward client API to enable both metadata and data retrieval

- Queries submitted as JSON (using Python or C++)
- General Purpose



Original image (left) and the result of query

Edge + Clouds: E2E Context Aware Application

While SAF can perform analytics and VDMS can store data and metadata, it is also necessary to have an application that provides context to the analyzed data.

For SAF, the application should be responsible for indicating which operators are needed and how they should be connected in order to achieve its objective.

The application should identify what its metadata schema would be in order to add entities with appropriate labels and properties to VDMS (e.g. Customer, Product)

Further, the application can provide additional functionality that the basic versions of SAF and VDMS may not support (FV summarizations, FV filteration)

Smart Retail Application Logic (User Queries)

We start the project by identifying a list of queries that drive the implementation

This step was important because we aimed to define the required schema in which we will store the data.

Examples of such queries:

- Visits per Store: provides retailers with information on how many people visited the store in a given time range.
- Trip Summary: lists all the areas a customer visited in a given trip. This can help with more targeted advertisements or understanding behaviors
- Hotspot Identification: identifies areas in which the largest number of customers have gathered. This can help store managers with ways to rearrange the store and spread out products to enable ease of viewing.
- Returned Customer Identification: based on a given FV, identify if this person is a returned customer or not.

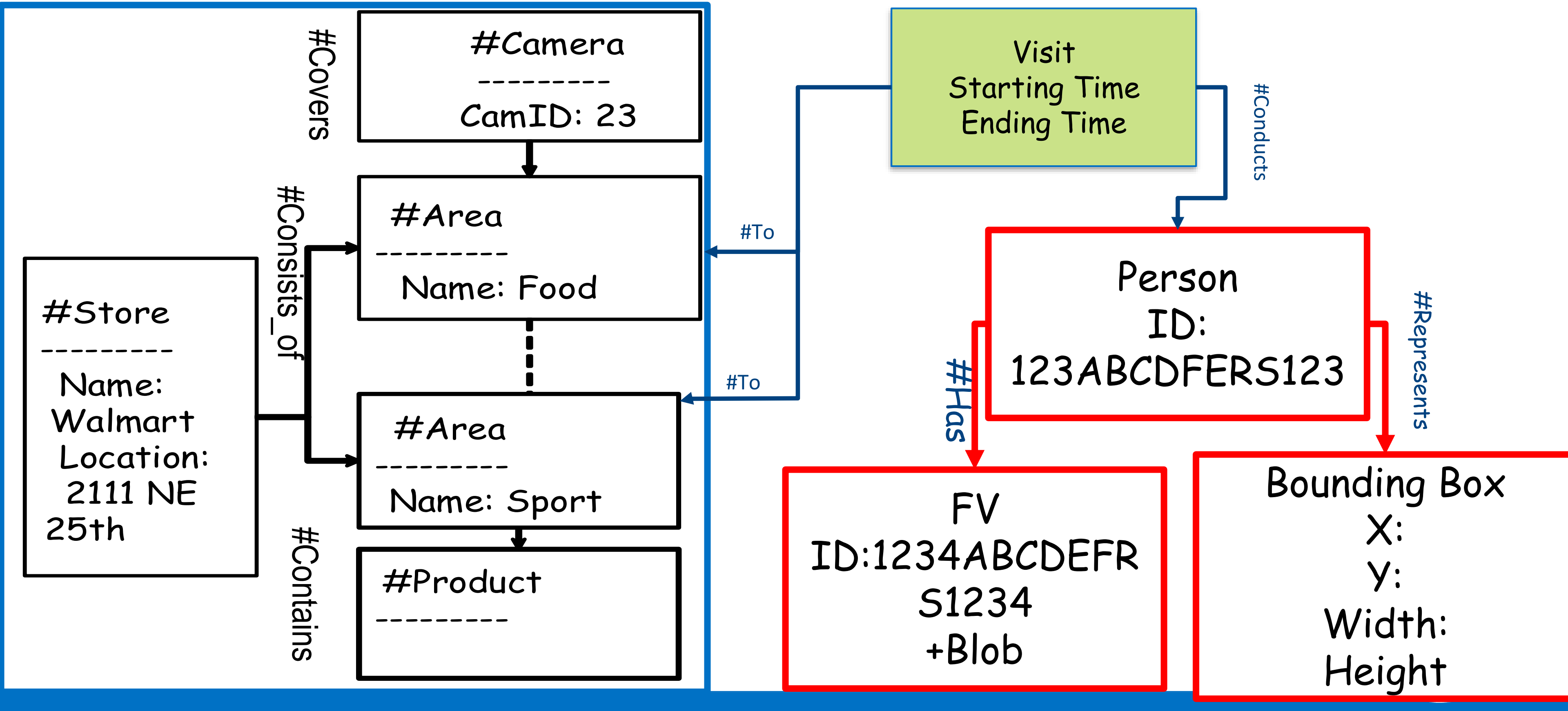
End to End Context Aware Application- Data Types

Constructed Data : related to the store configuration, number of cameras, and list of areas, association between cameras and areas.

Dynamic Data: this is the data that being generated by the SAF pipeline. The data will be fed to VDMS in real-time. It is the job of the retail application logic to make sense of this data and build the schema that VDMS understands.

Enhanced Data: this data is created by the application logic. It is very important to link between the contextual data and the dynamic data. Example of that is the Visit node in our schema.

Retail Smart Application Logic: Building the Schema



Visit Node Building through Re-Identification of a Person

Re-identification is the process by which a collection of feature vectors is searched for a *query feature vector*, usually corresponding to a person or object, to establish if there is a similar feature vector in the collection.

The similarity is usually expressed and measured in terms of geometric distances (euclidean distance, inner product, etc). This operation is also called *feature matching*, and the similarity search is often performed using a k-nearest neighbor (*knn*) algorithm.

VDMS natively supports the ability to perform efficient *feature matching*, thus enabling this application to perform re-identification of a person based on extracted feature vectors from SAF

Re-Identification of a Person

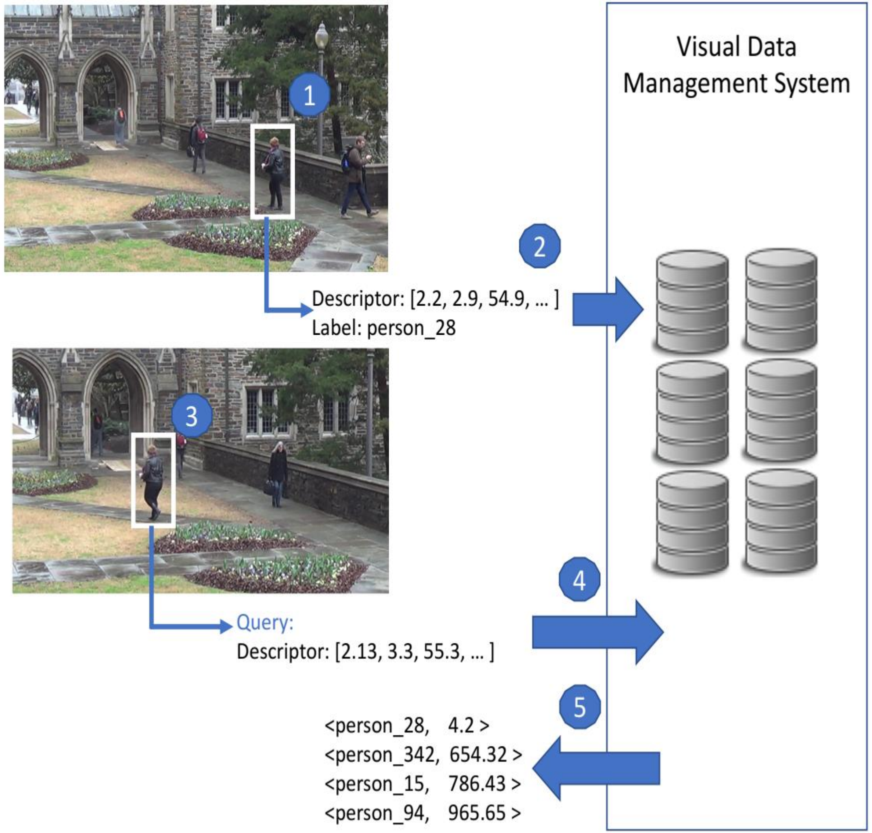
For each new FV, find_Fv similar(Saf_FV)

If (_distance <=threshold)

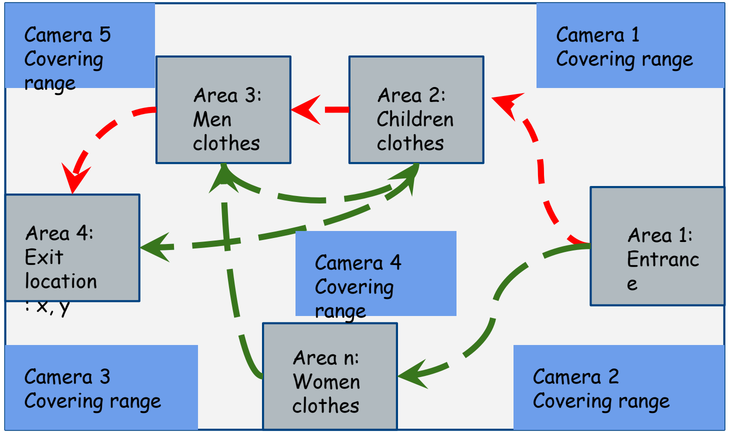
Add FV as alias to the stored person

If(_distance>threshold)

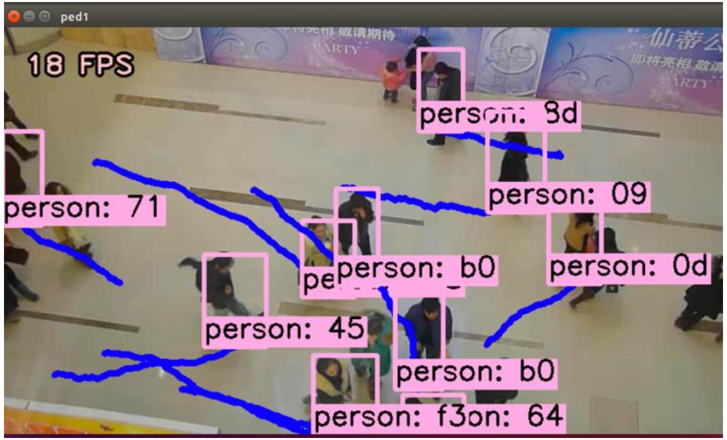
Add new person and new FV node



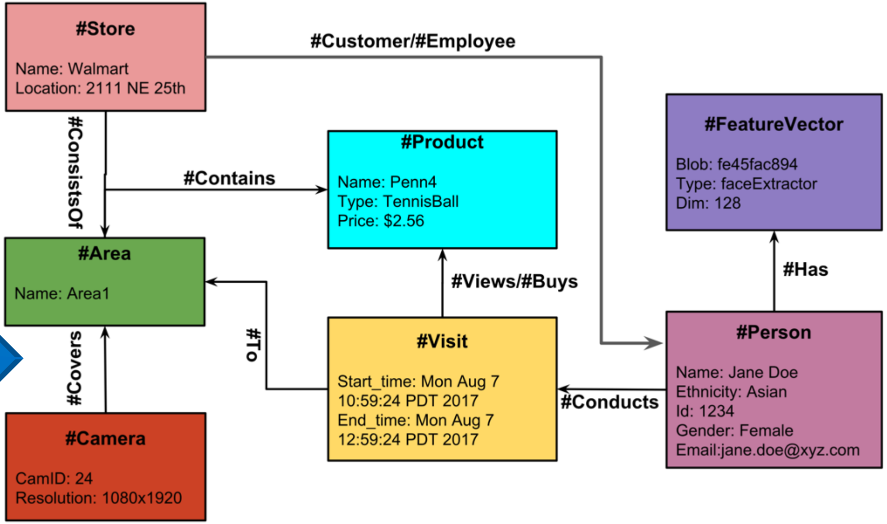
Shopper Insights: SAF & VDMS



Store configured



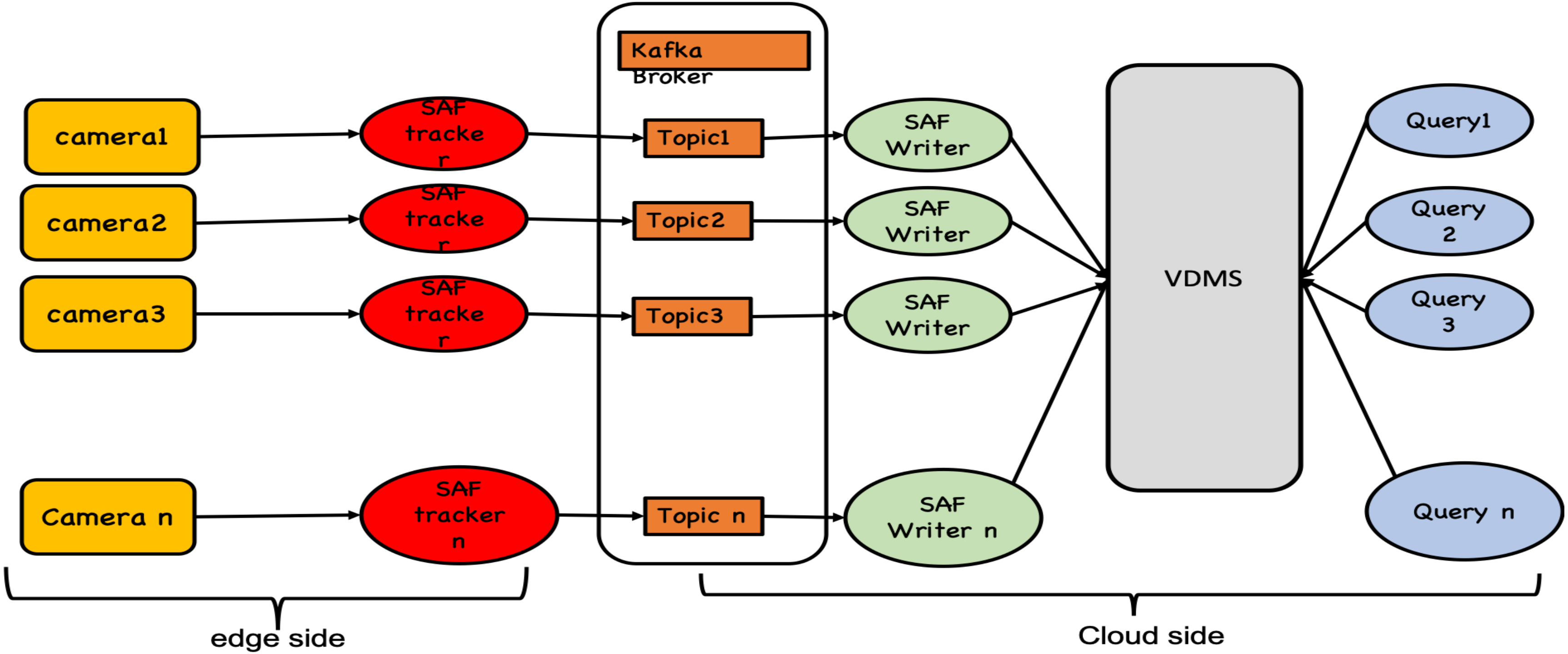
Shoppers are tracked by SAF



Store and Customer's metadata in VDMS

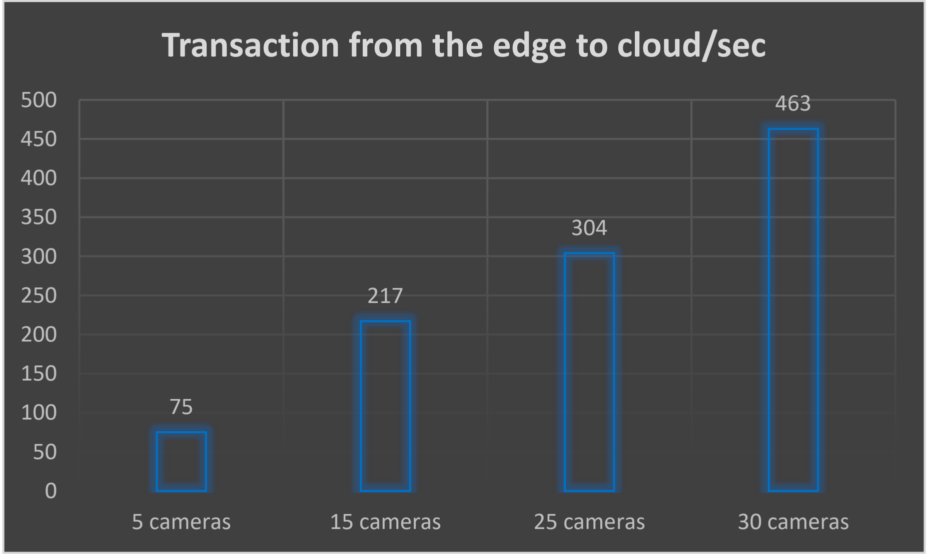
Feature vectors as data in VDMS

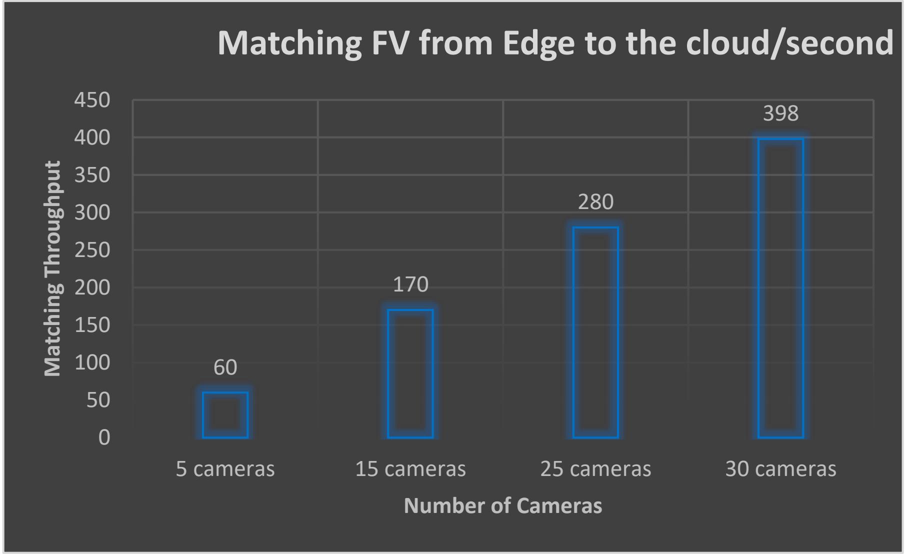
Retail PoC – Currnet Evaluation Results



- 30 SAF pipelines running simultaneously for 2 hours, and writing metadata and extracted FV blob to VDMS.

Results:





*Platform: Broadwell – 2x Xeon Processor E5-2699 v4 – 22 Core per processor

Discussion and Open Issues

- Feature vector summarizing techniques, where it should happen?
- Re-identification of a person identity based on changing feature vectors
- Preserve privacy: collecting meta-data about people including there feature vectors together with the actual frame of the person might expose extra information about the person