

Investigating Multi-Stream FaaS Applications in Geo-Distributed Edge Environments

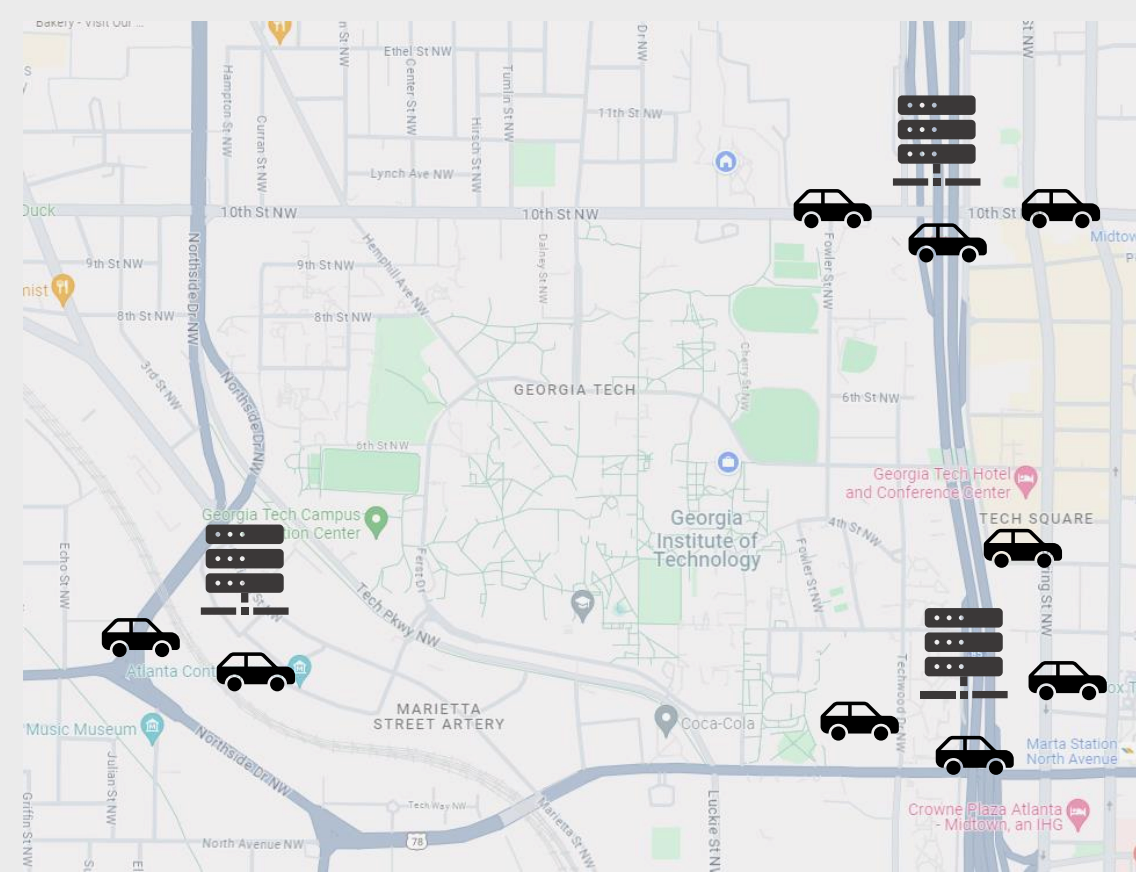
Jinsun Yoo¹, Anirudh Sarma¹, Difei Cao², Kartik Sinha¹, Myungjin Lee³, Umakishore Ramachandran¹

School of Computer Science [1], School of Electrical and Computer Engineering [2], Cisco [3]



Georgia Tech College of Computing
Center for Research into
Novel Computing Hierarchies

Introduction

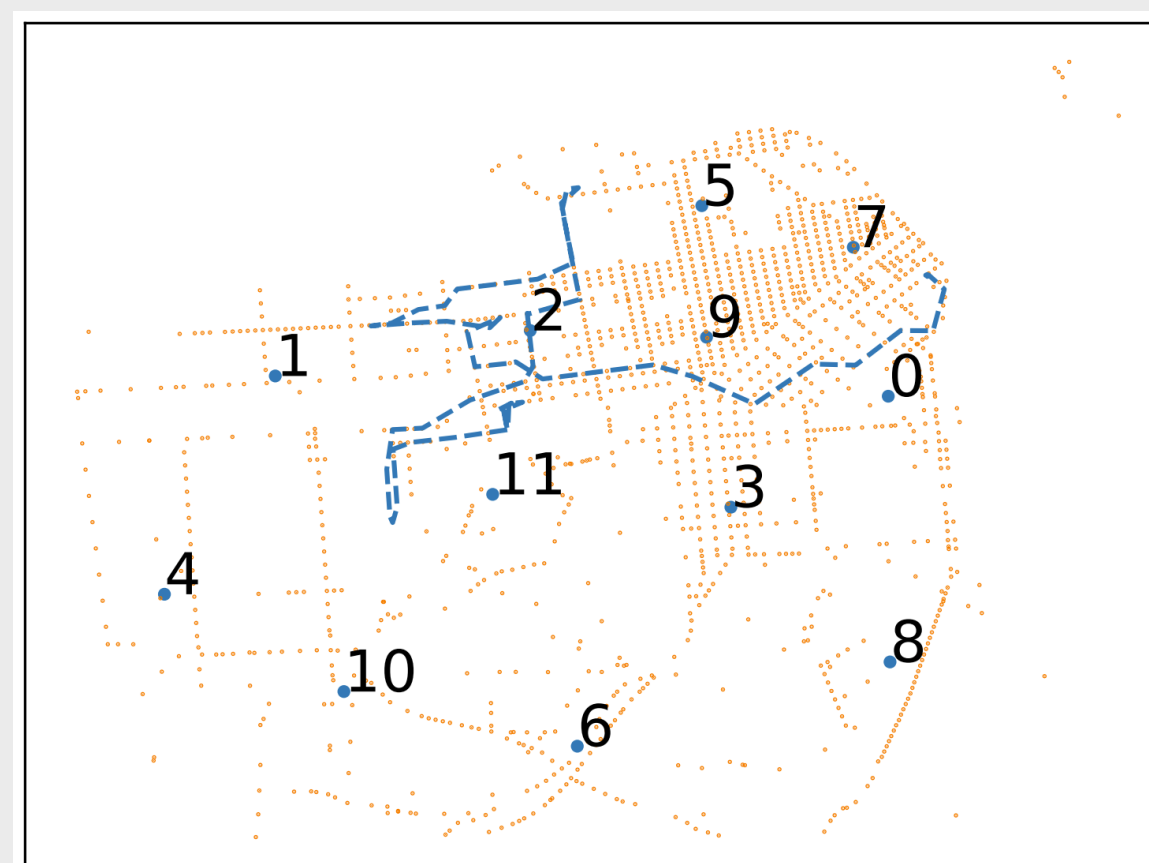


Function as a Service (FaaS) is gaining interest for **Situation Awareness Applications**

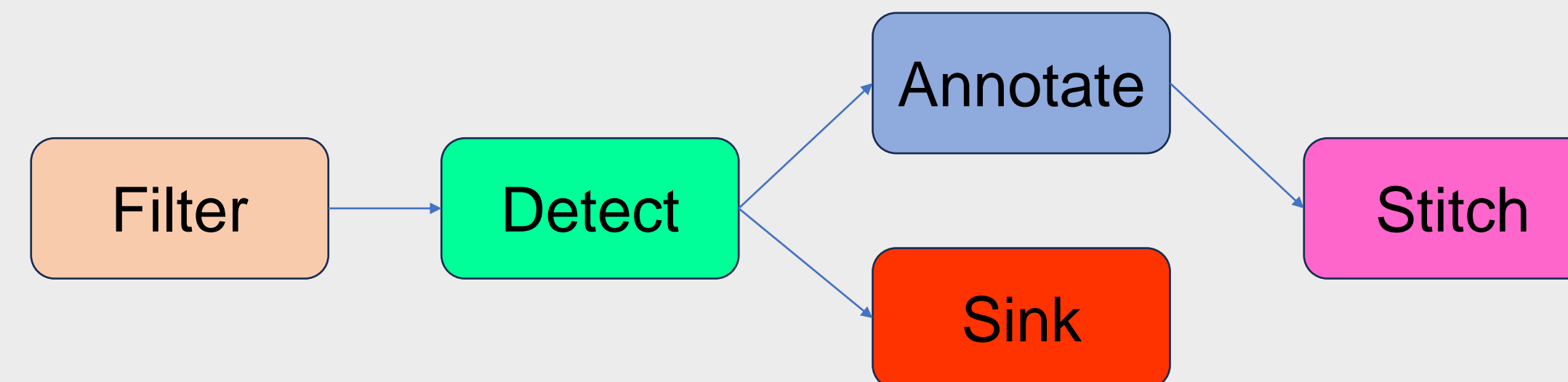
- Limited Resources on the Edge / Pay-as-you-Go
- Swift scaling capabilities

Current FaaS platforms do not provide the abstractions necessary for a **geo-distributed multi-stream** applications

- Combining insight from multiple clients
- Deploying across geo-distributed edge sites to match client defined SLO



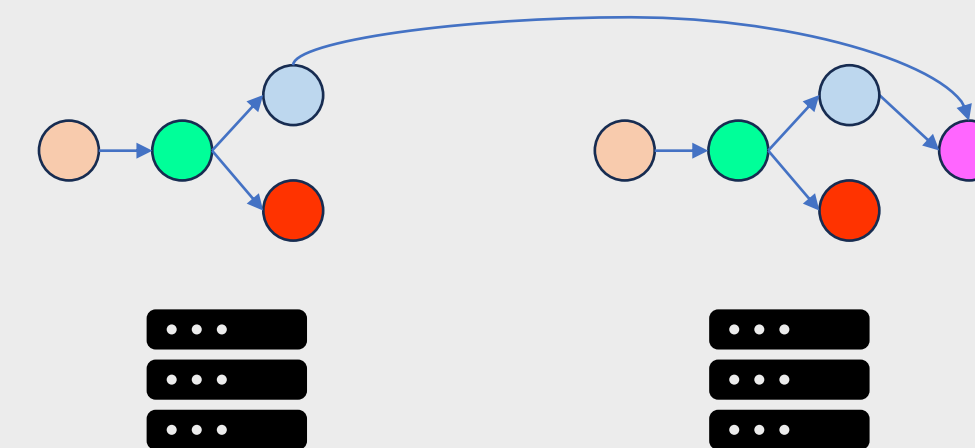
Application Model



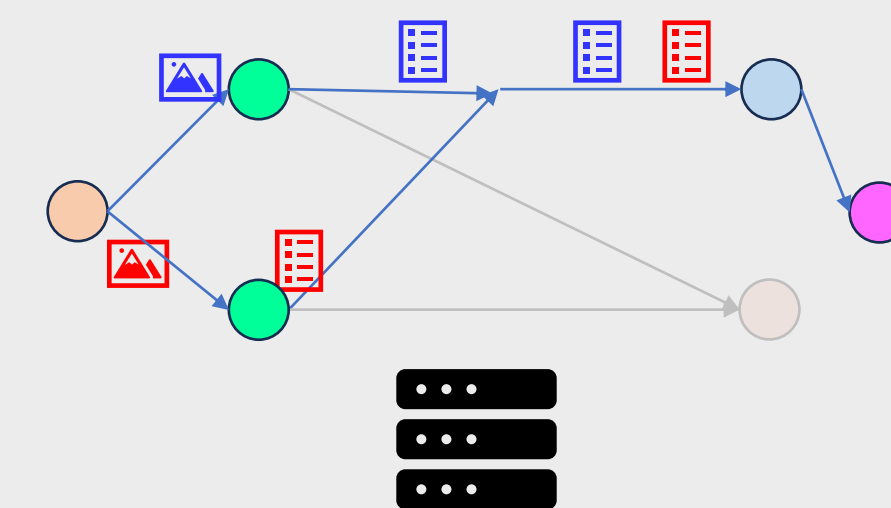
Detect objects in line of sight and share information

- **Stitch** function aggregates objects from multiple vehicles in an Area of Interest
- Functions can be **reused** between different vehicles

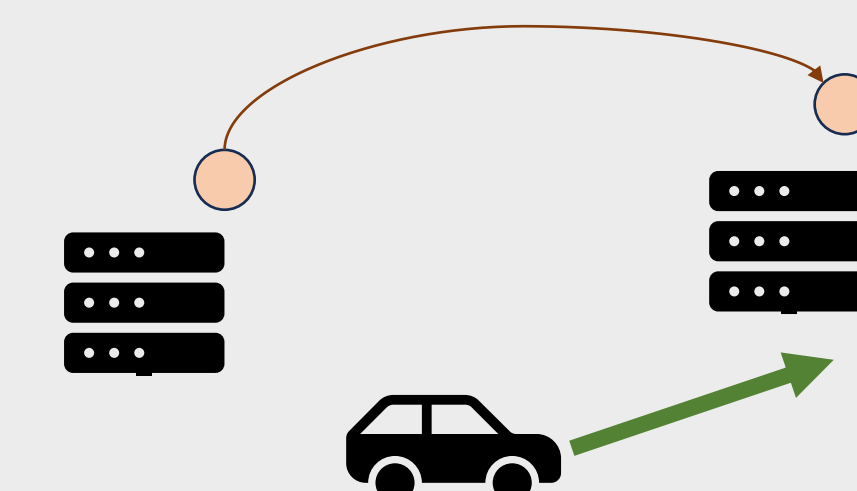
Challenges



Spatial Correctness: Invoke the correct function attributed to this stream



Temporal Correctness: Input from different stream of same timestamp, Ordered input process



Client Mobility Aware Scheduling: Preemptive and Reactive adjustment from latency and resource constraints

Our Programming Model

Sample User Input

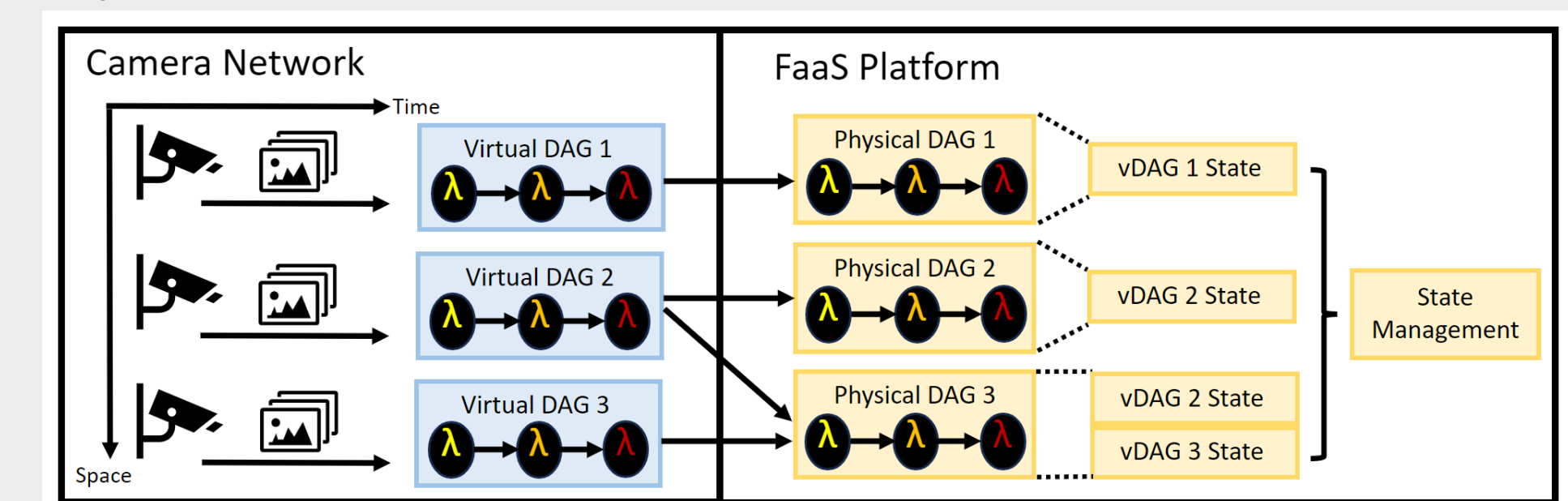
```
k2e_latency_slo_ms: 500
functions:
- name: filter
  image: docker.com/jinsun-yoo/filter
  downstream_connection: ['detect']
- name: detect
  image: docker.com/jinsun-yoo/detect
  downstream_connection: ['annotate', 'sink']
- name: annotate
  image: docker.com/jinsun-yoo/filter
  downstream_connection: ['stitch']
- name: sink
  image: docker.com/jinsun-yoo/filter
  downstream_connection: []
- name: stitch
  image: docker.com/jinsun-yoo/filter
  downstream_connection: []
temporal_ordering: true
```

```
connections:
- functions: ['annotate', 'stitch']
  connection_type: join_vdag
  scope: region_of_interest
  deadline: 100ms

regions_of_interest:
- coords:
  - x: 125.00
  - y: 38.00
  - x: 150.00
  - y: 40.00
- coords:
  - x: 150.00
  - y: 38.00
  - x: 175.00
  - y: 40.00
```

User only provides the **minimal description** of the DAG and **no changes to the function** is necessary
The system takes care of ensuring correctness & orchestration

Underlying DAG Concept



User specifies DAG of the application with set of inputs
System takes care of mapping user's DAG respecting **spatio-temporal correctness** and **SLO** requirements of the applications

Future Work

Algorithmic Study & Evaluation

- Consensus to maintain mapping across distributed Edge sites
- Timestamp scheme to ensure temporal correctness
- Enhanced scheduling considering client movement and DAG topology