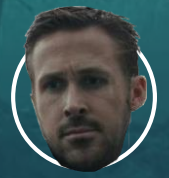
Eywa

Eugene Siow | Thanassis Tiropanis | Wendy Hall

BLADE RUNNER 2049



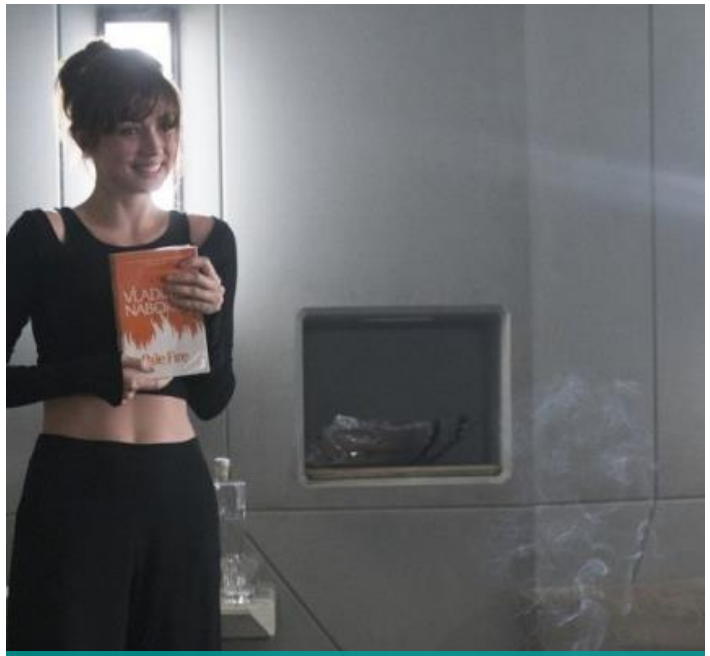
Ryan Gosling / K
Replicant

Ana de Armas / Joi
AI

A movie about **Fog Computing**

Fog Computing Sci-Fi

Blade Runner 2049's **Joi** lives in the Fog



Smart Home Hologram

Joi lives on a **console** in K's home rather than the cloud. She can control all actuators in the house.

Emanator

Joi can reside on the **portable emanator** and move around with K. A Fog Computing device?

K's Spinner Car

Joi is connected to the car. When the spinner goes down, she loses the ability to project herself.

The **Internet** of **Things**

Cloud

Dynamic provisioning of scalable resources e.g. analytics on a **huge volume** of historical data.

Fog Computing

An emerging technology that bridges the gap, deployed **close to the source**.

Things

Connected sensors and actuators producing **streams of time-series data.**

Challenges for **Fog Computing**

01 Interoperability

Heterogeneity of device, platform and data

02 Stream Processing

Performant and scalable processing of multiple streams in real-time

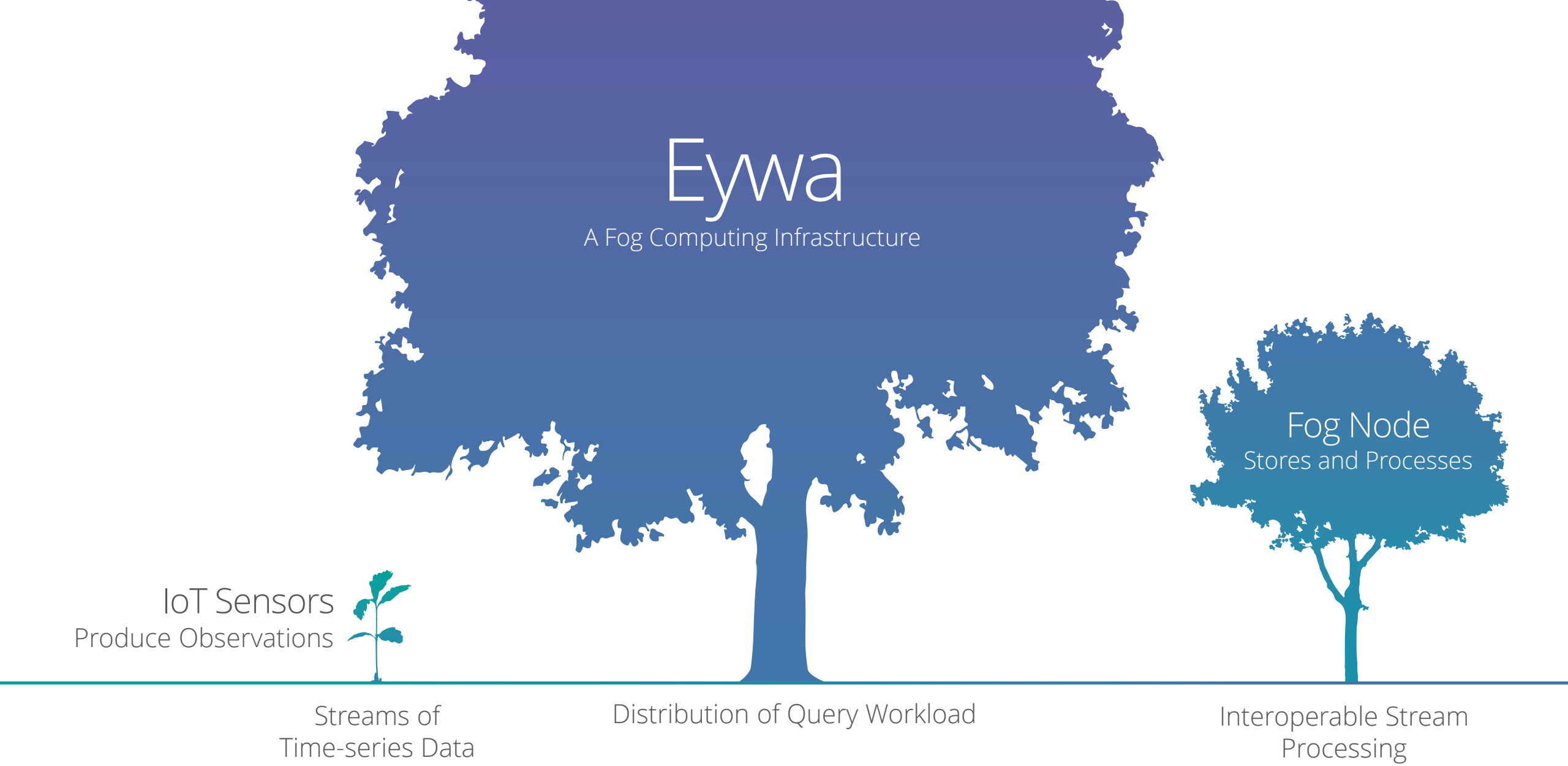
03 Distribution

Provisioning of resources and distribution of work load

Eywa is like a huge biological internet; the trees are fog computing nodes that store and process information and sensors are connected flora and fauna

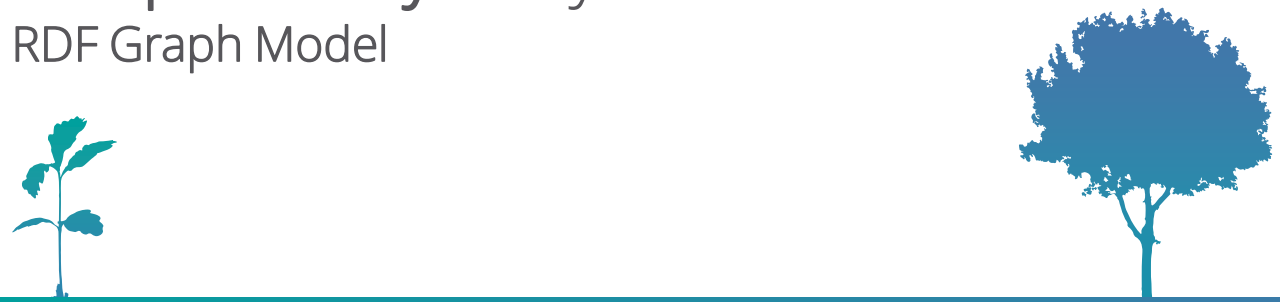
Avatar

By James Cameron


Eywa
A Fog Computing Infrastructure
Fog Node
Stores and Processes
IoT Sensors
Produce Observations
Streams of
Time-series Data
Distribution of Query Workload
Interoperable Stream
Processing

Semantic Interoperability in Eywa

Using a Common RDF Graph Model



IoT Domains for Things and Apps

Common Structure
Widely-used, flexible model

Data Integration
Stores Rich Metadata

Graph Querying
Powerful SPARQL Graph Queries

Siow, E., Tiropanis, T., Hall, W. (2016). **Interoperable and Efficient: Linked Data for the Internet of Things.** The 3rd International Conference on Internet Science.

Deliver

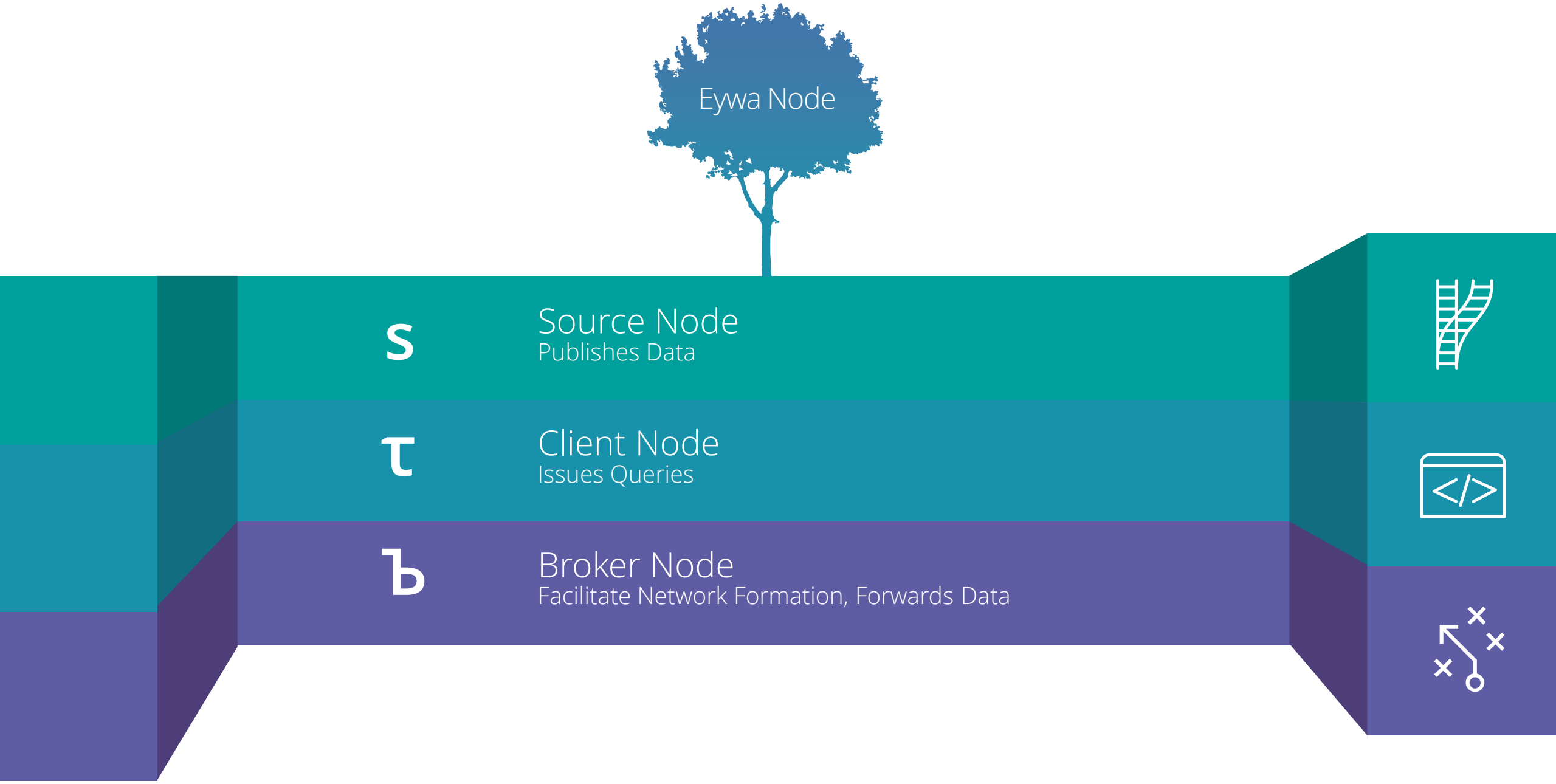
Inverse-publish-subscribe

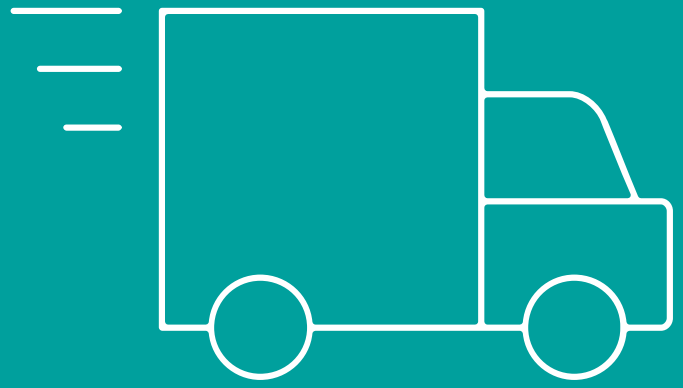
Distribute

Workload Distribution by Projection Pushdown

Process

Stream Processing by Query Translation


Eywa Node
Source Node
Publishes Data
Client Node
Issues Queries
Broker Node
Facilitate Network Formation, Forwards Data



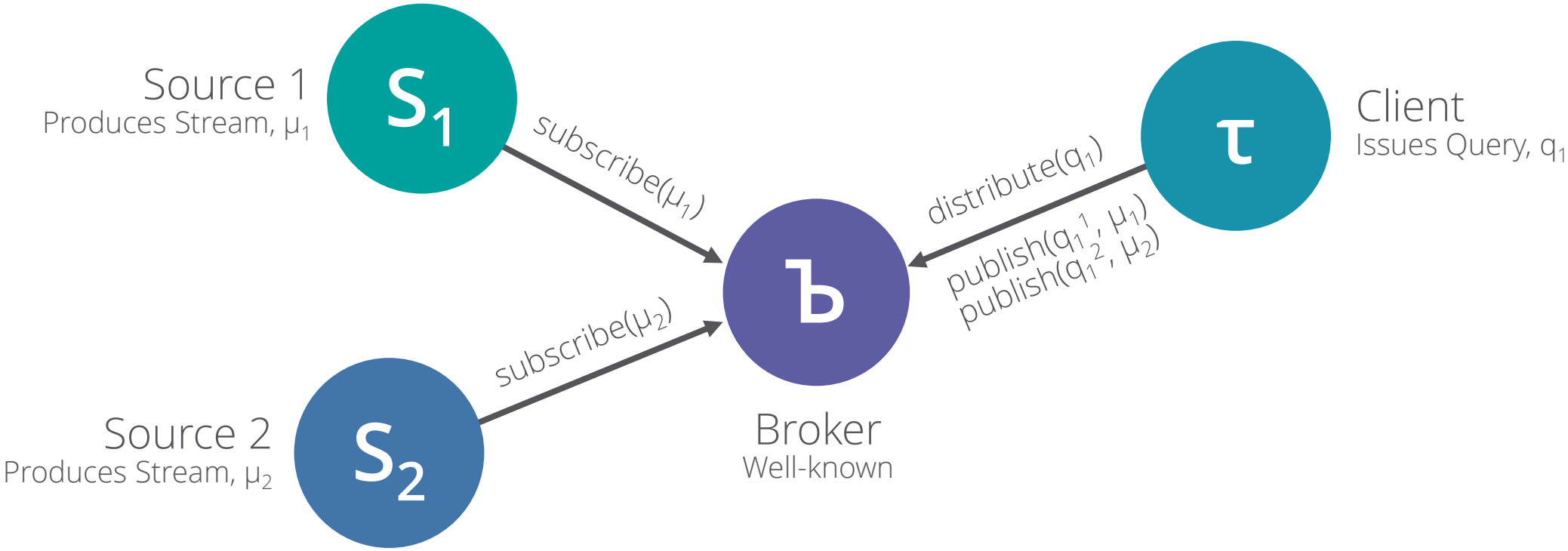
Deliver Queries

Inverse-Publish-Subscribe

A novel system for **best-effort, co-operative query delivery** in fog computing *control-plane*

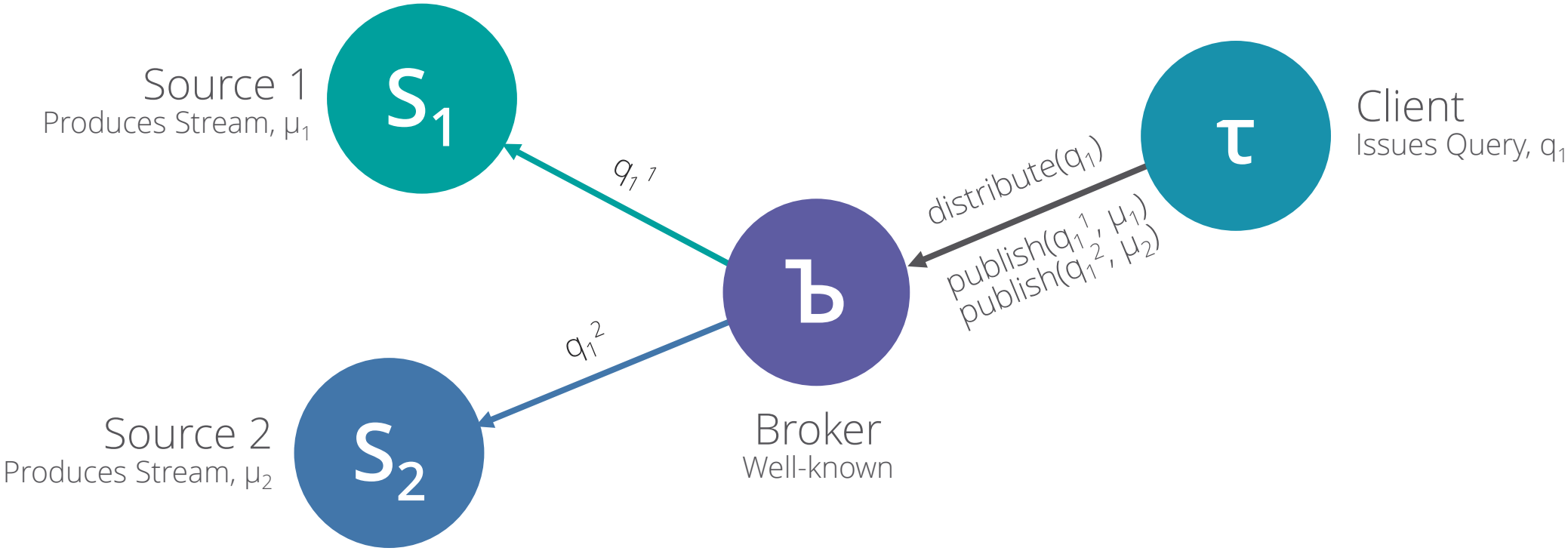
Deliver Queries

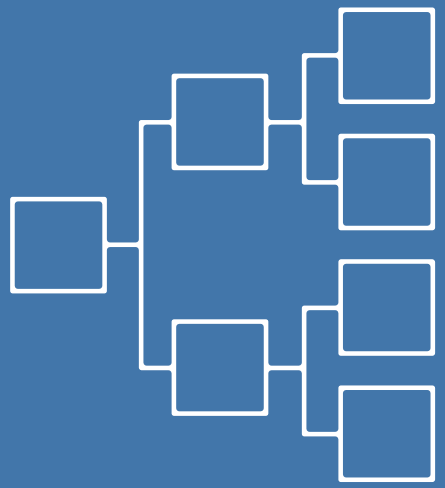
Inverse Publish-Subscribe (1)



Deliver Queries

Inverse Publish-Subscribe (2)





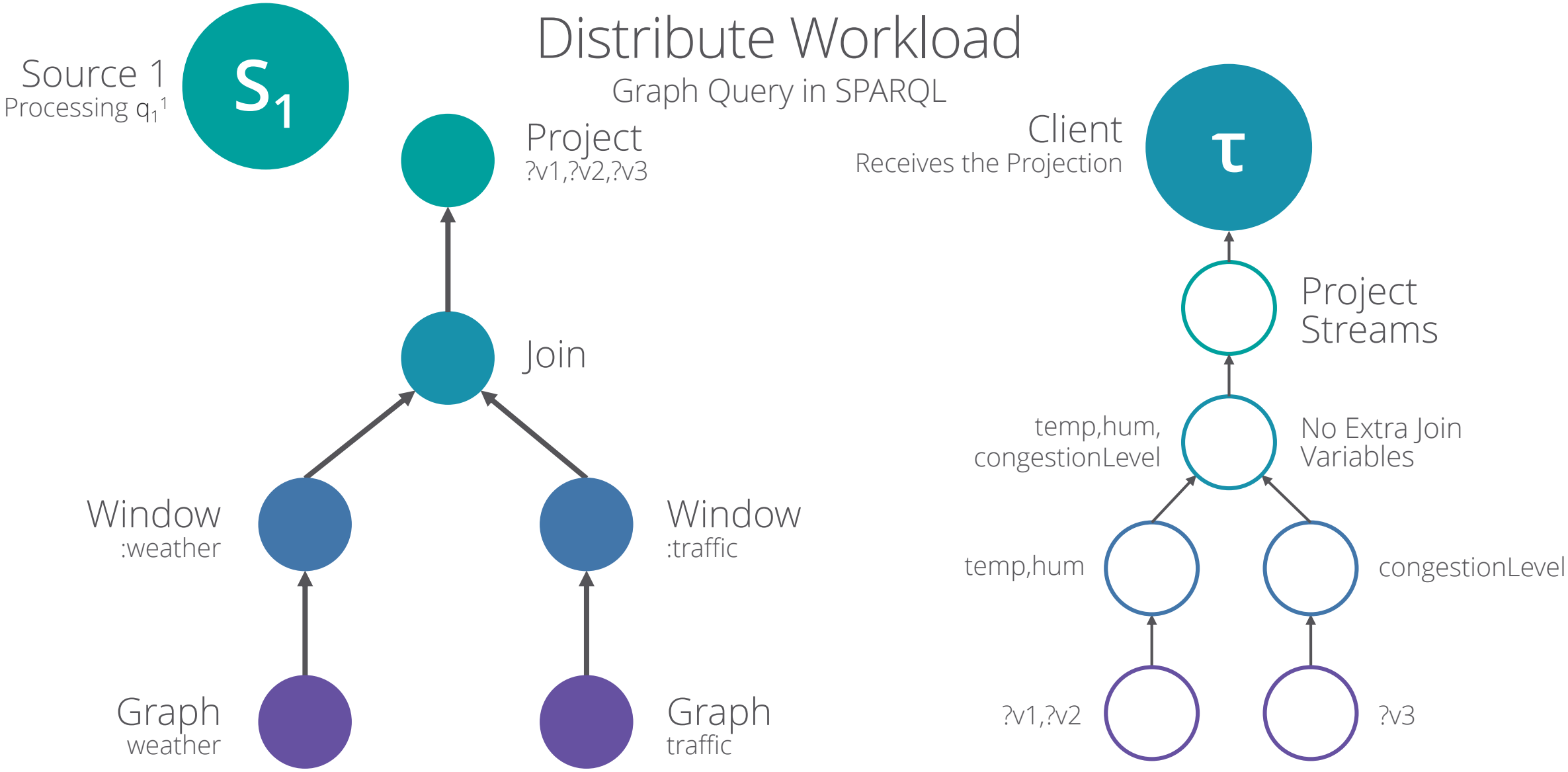
Distribute Workload

Projection Pushdown

Efficient streaming of only **projected data** in the fog computing *data-plane*

Distribute Workload

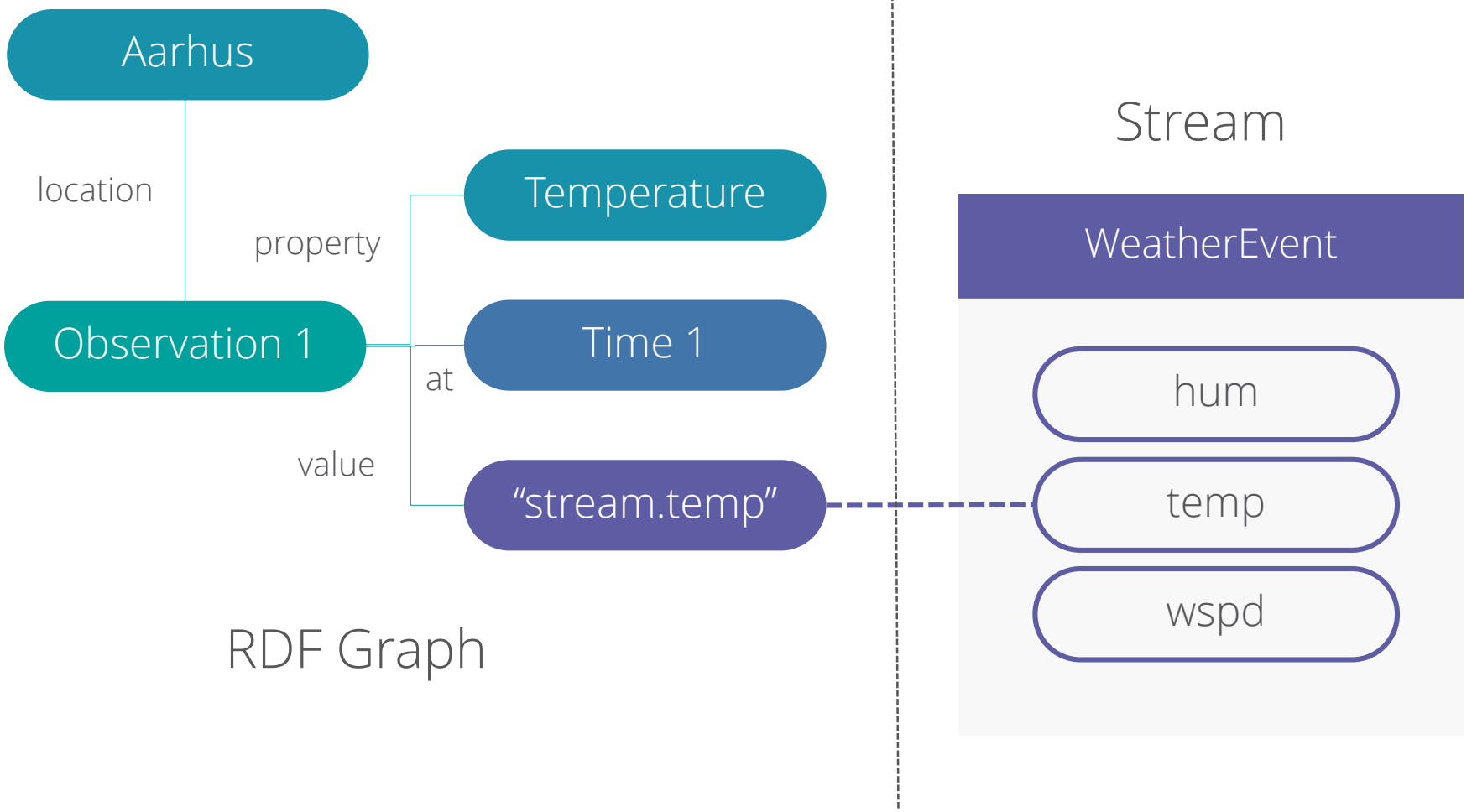
Graph Query in SPARQL



From **CityBench** (Smart City Streams) Query 2. **Finding the traffic congestion level and weather conditions of my planned journey.**

Distribute Workload

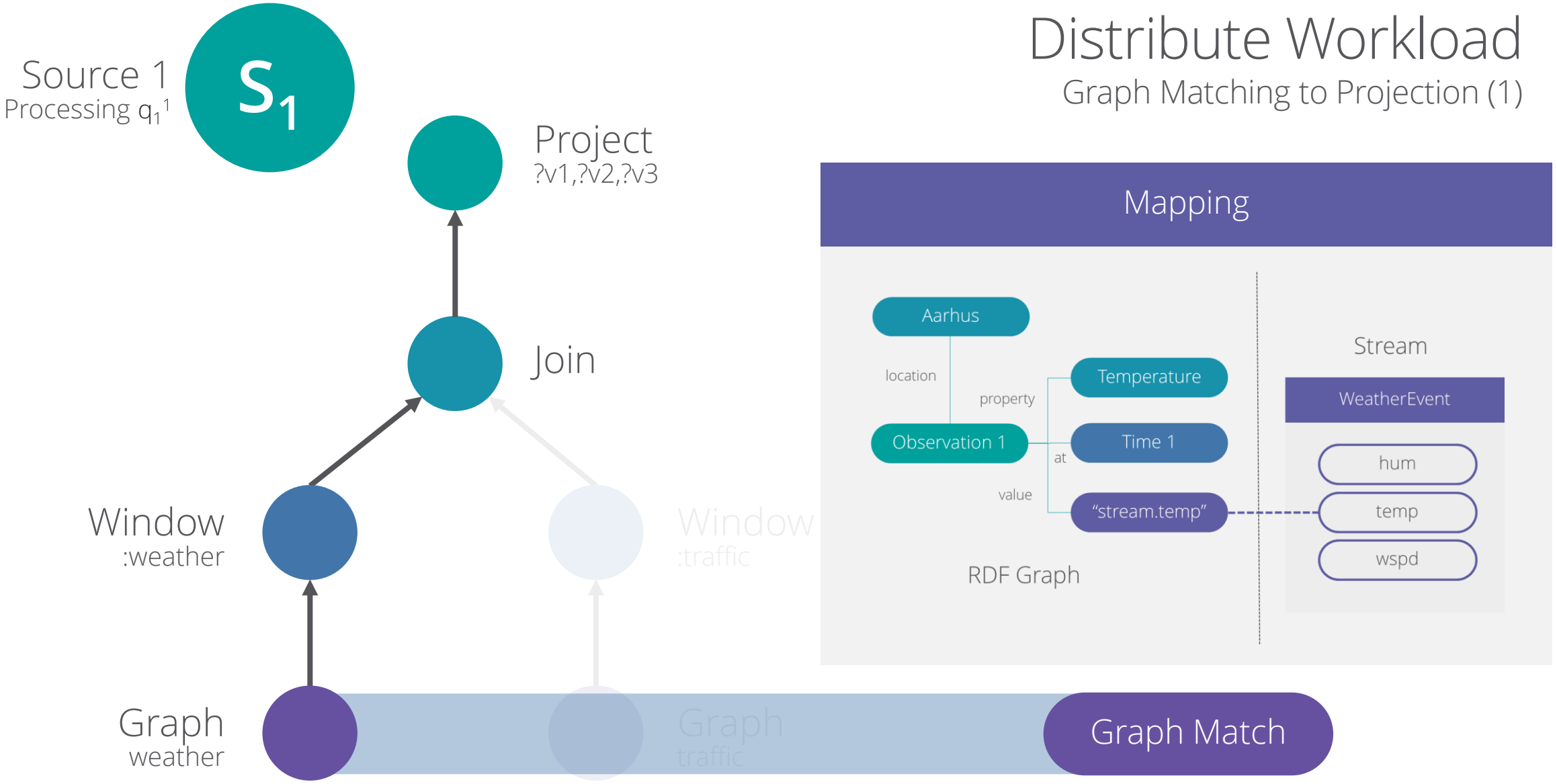
Efficient Mappings for RDF Stream Processing



Siow, E., Tiropanis, T. and Hall, W. (2016) SPARQL-to-SQL on internet of things databases and streams. ISWC2016: The 15th International Semantic Web Conference

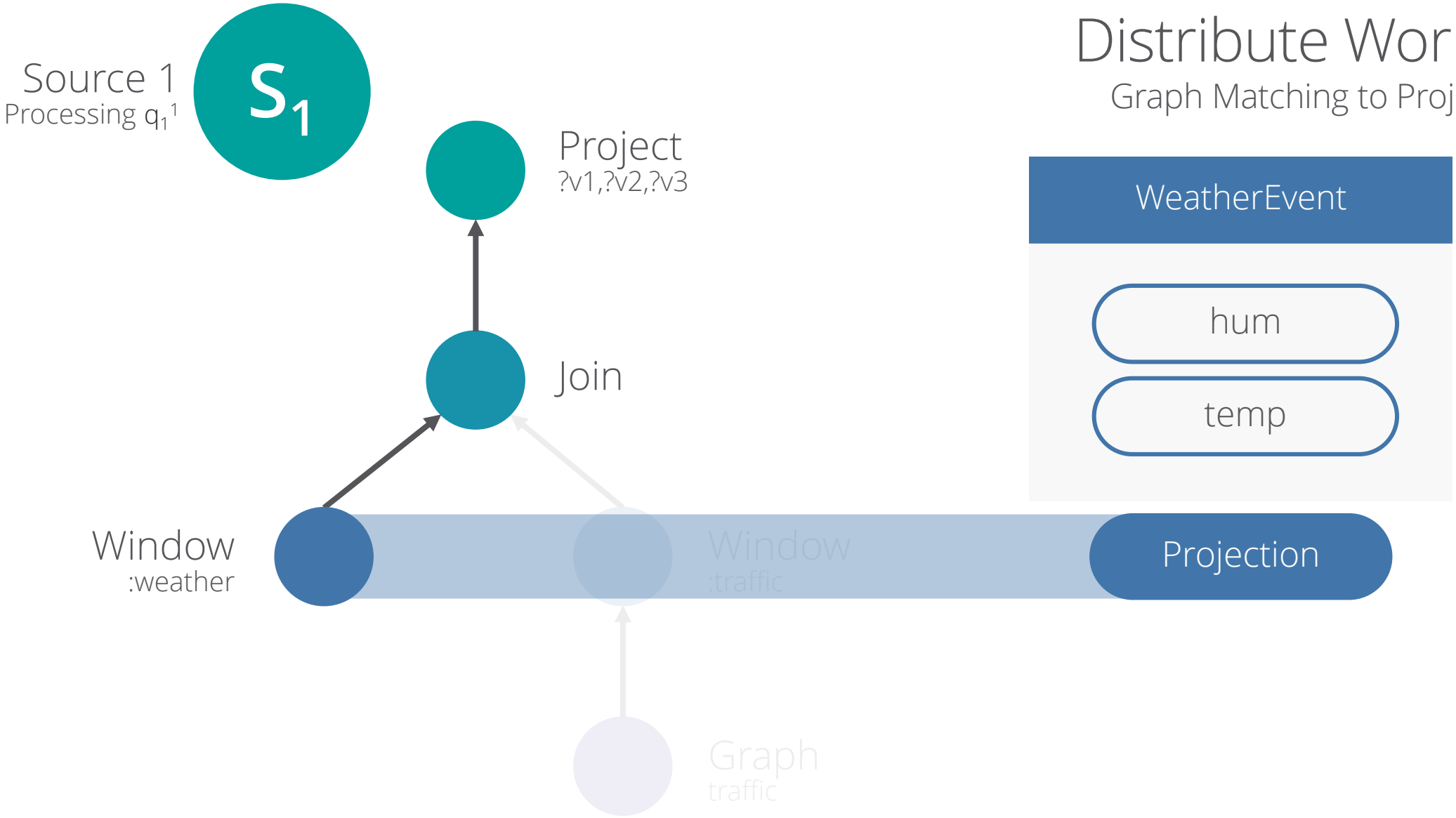
Distribute Workload

Graph Matching to Projection (1)



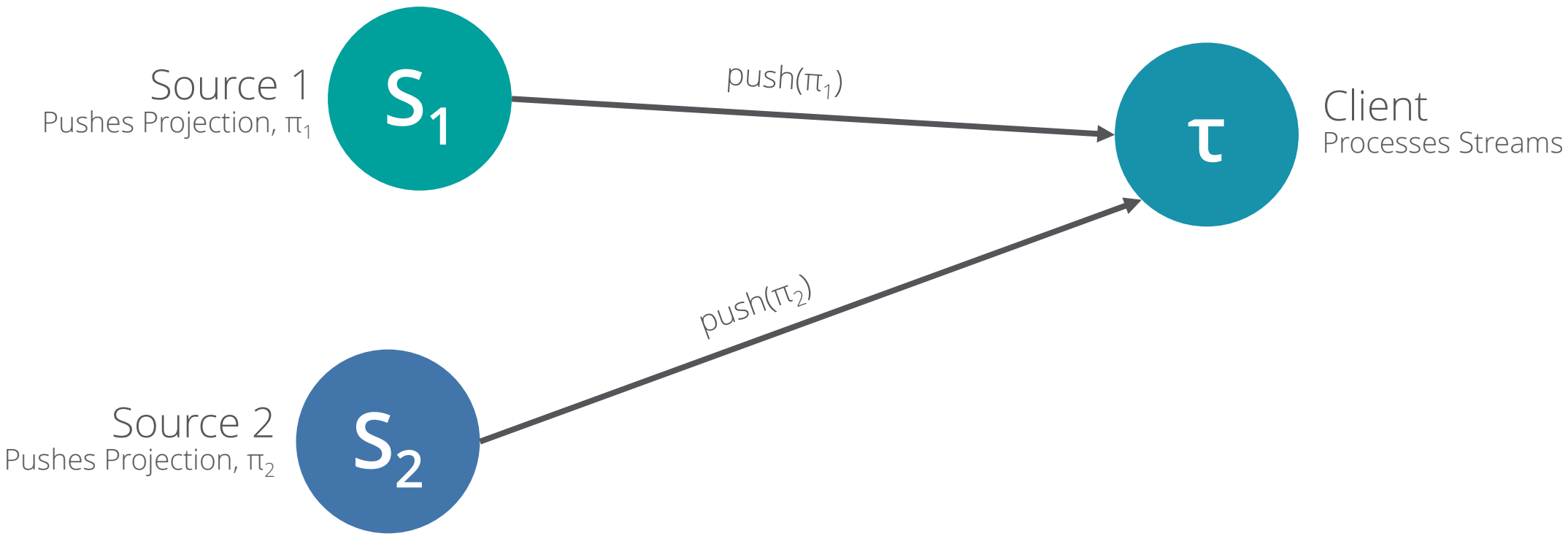
Distribute Workload

Graph Matching to Projection (2)



Distribute Workload

Projection Pushdown, Push Projection



Process

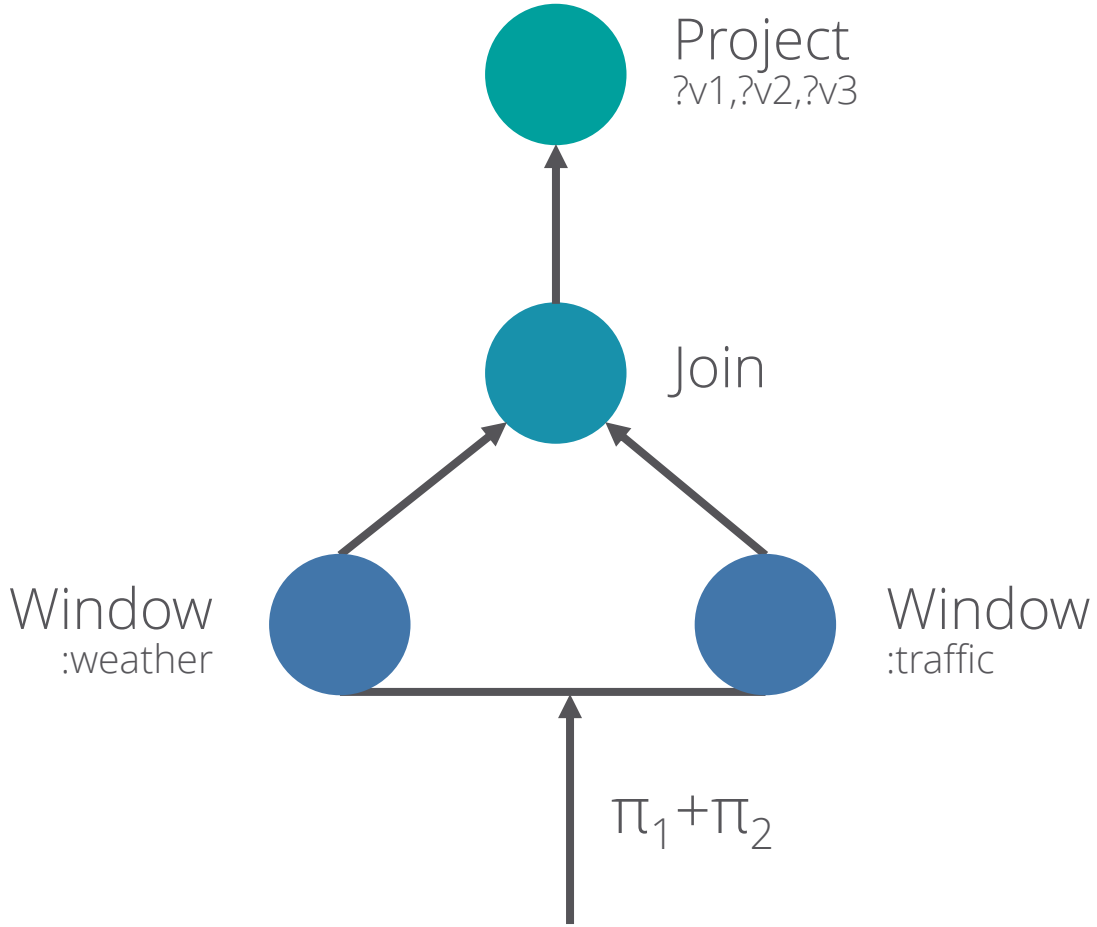
Query Translation

Efficient stream processing of **graphs for IoT time-series** in the fog computing *data-plane*

Processing Streams

Query Translation for Stream Processing

Client
Processes Query, q_1



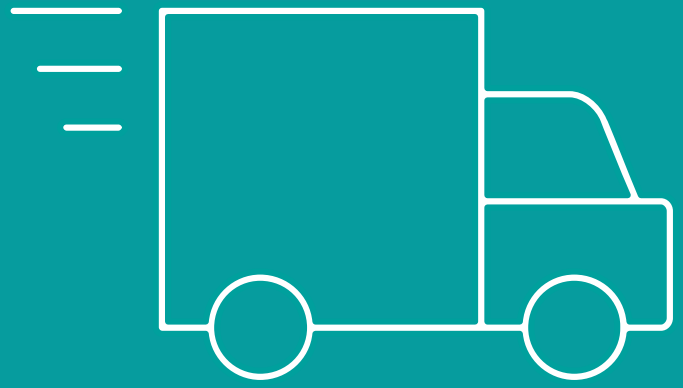
Event Processing Language Query

SELECT

- temp AS v1
- hum AS v2
- congestionLevel AS v3

FROM

- weather(3s)
- congestionLevel(3s)



Evaluation

CityBench Smart City Benchmark

Latency and Scalability

Evaluation on 3 Stream Processing Engines

Smart City RDF Streams

CITYBENCH

Real-time streams (e.g. vehicle traffic, parking, weather, pollution)

Based on smart city applications (e.g. parking space finder, admin console)

Run on resource-constrained Raspberry Pis as Fog Nodes (~500mhz CPU, 512mb ram, SD CARD)

01

C-SPARQL

Barbieri et al. "C-SPARQL: SPARQL for continuous querying." WWW2009.

02

CQELS

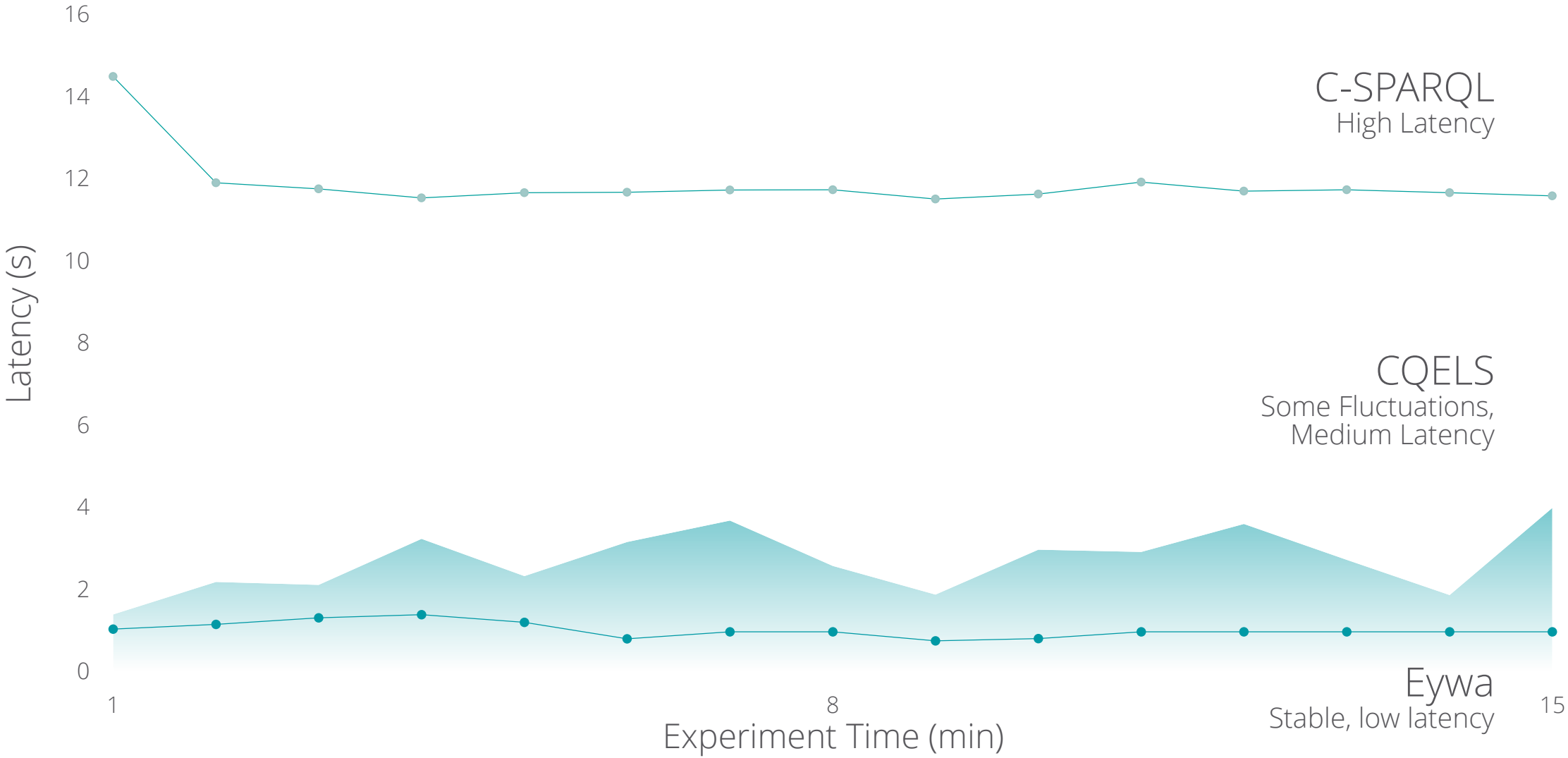
Le-Phuoc et al. "A native and adaptive approach for unified processing of linked streams and linked data." ISWC2011

03

Eywa

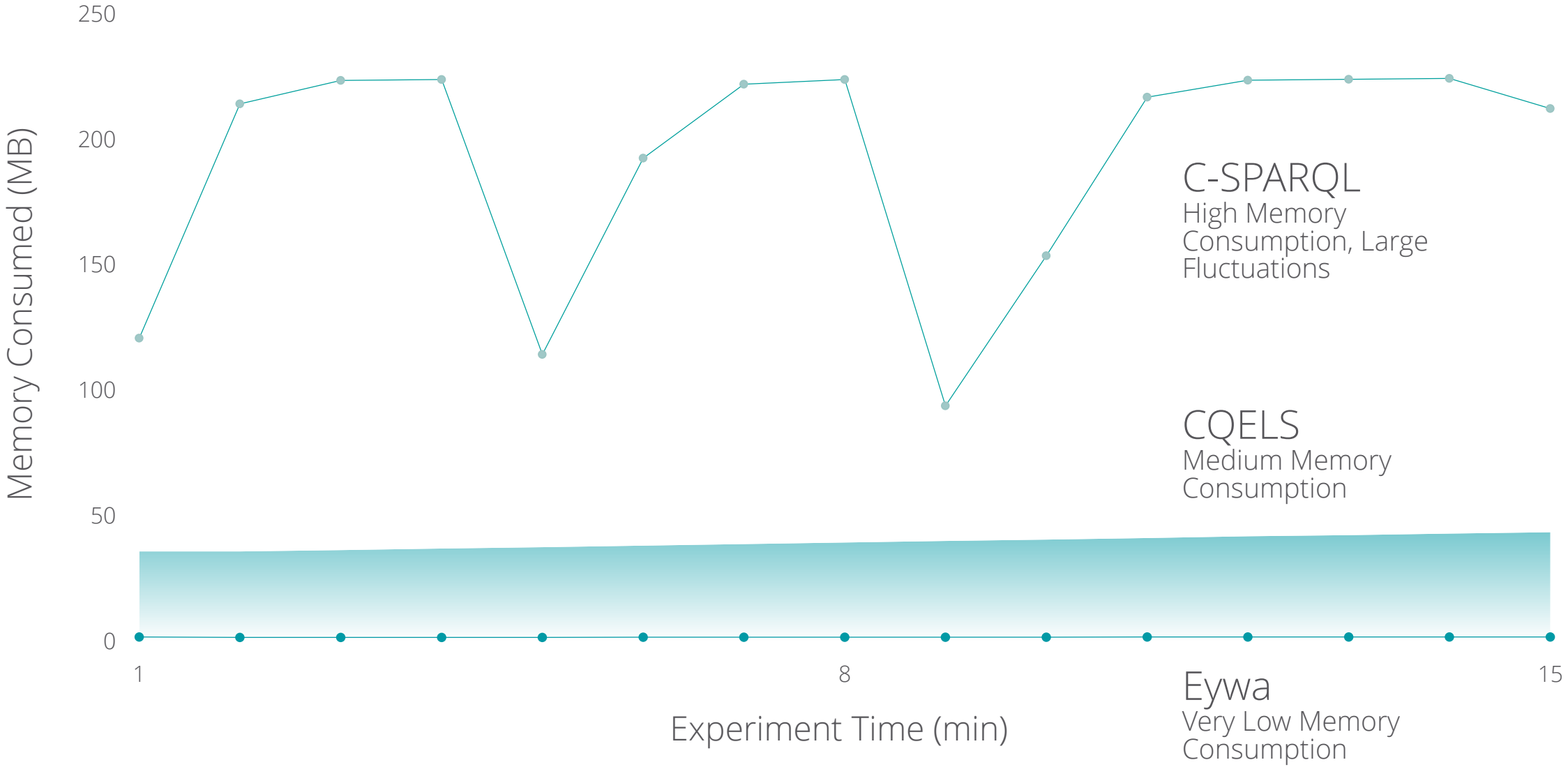
Latency Evaluation

CityBench Query 1 (traffic congestion level on two roads)



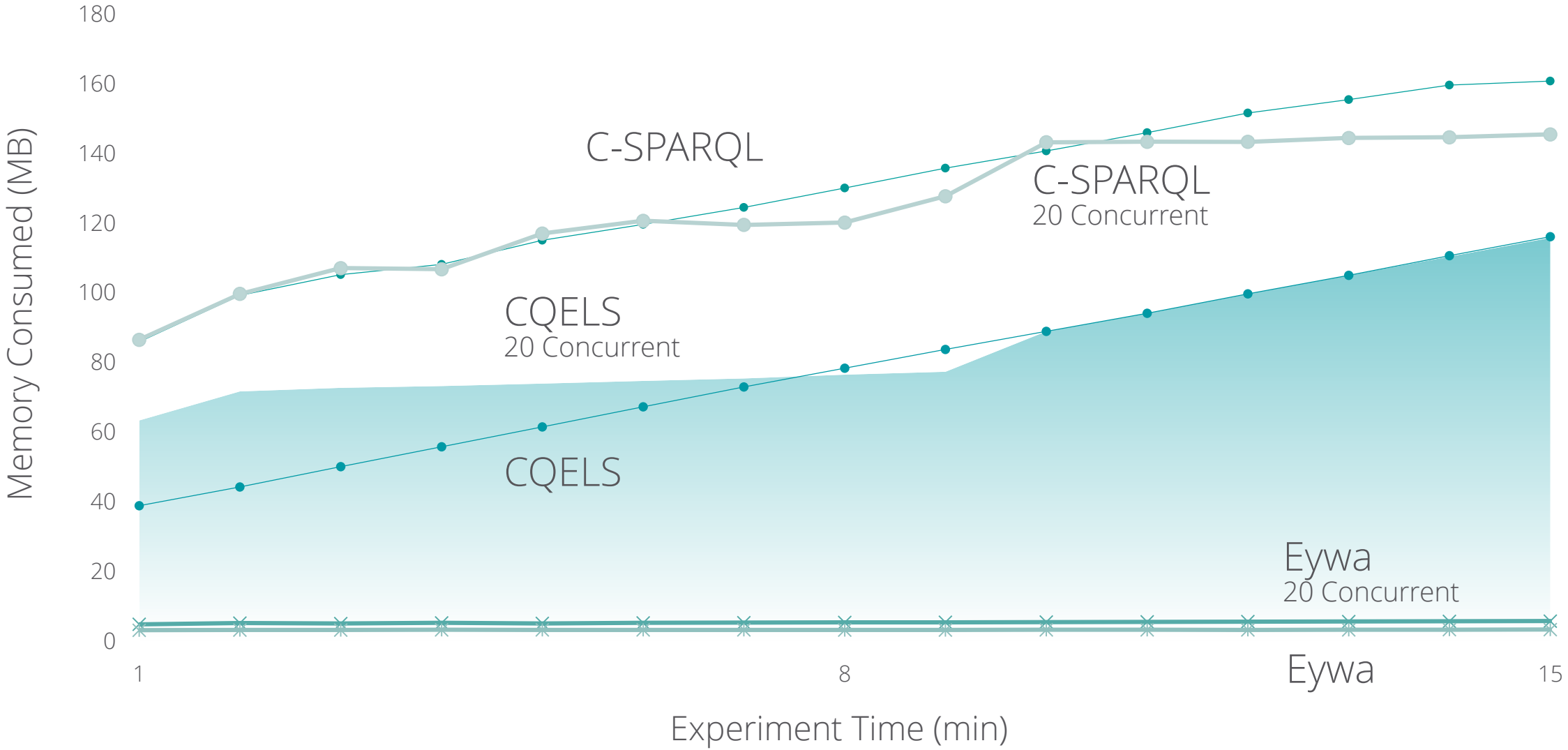
Scalability Evaluation

CityBench Query 2 (traffic congestion level and weather)



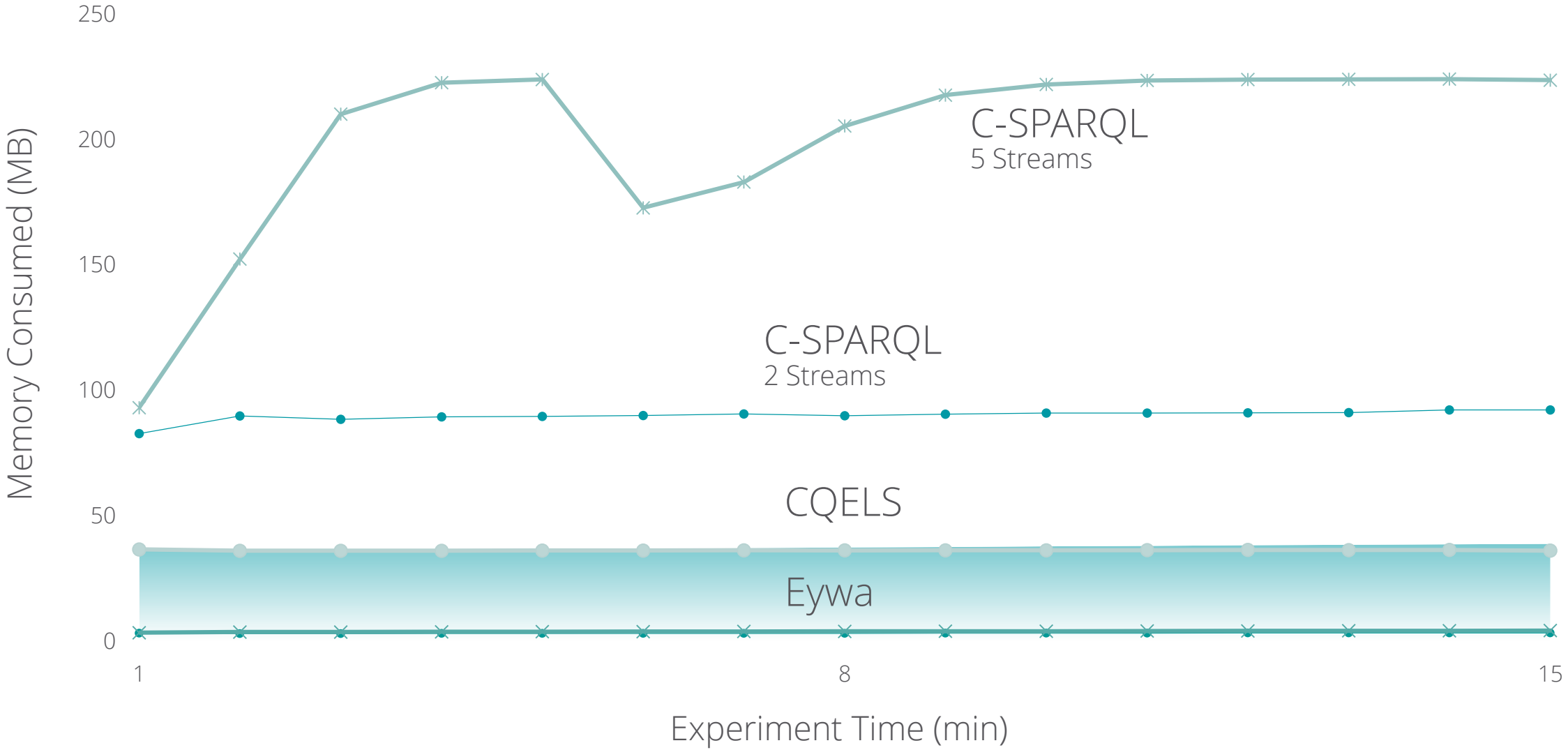
Scalability Evaluation

CityBench Query 5 (traffic congestion where event is happening)



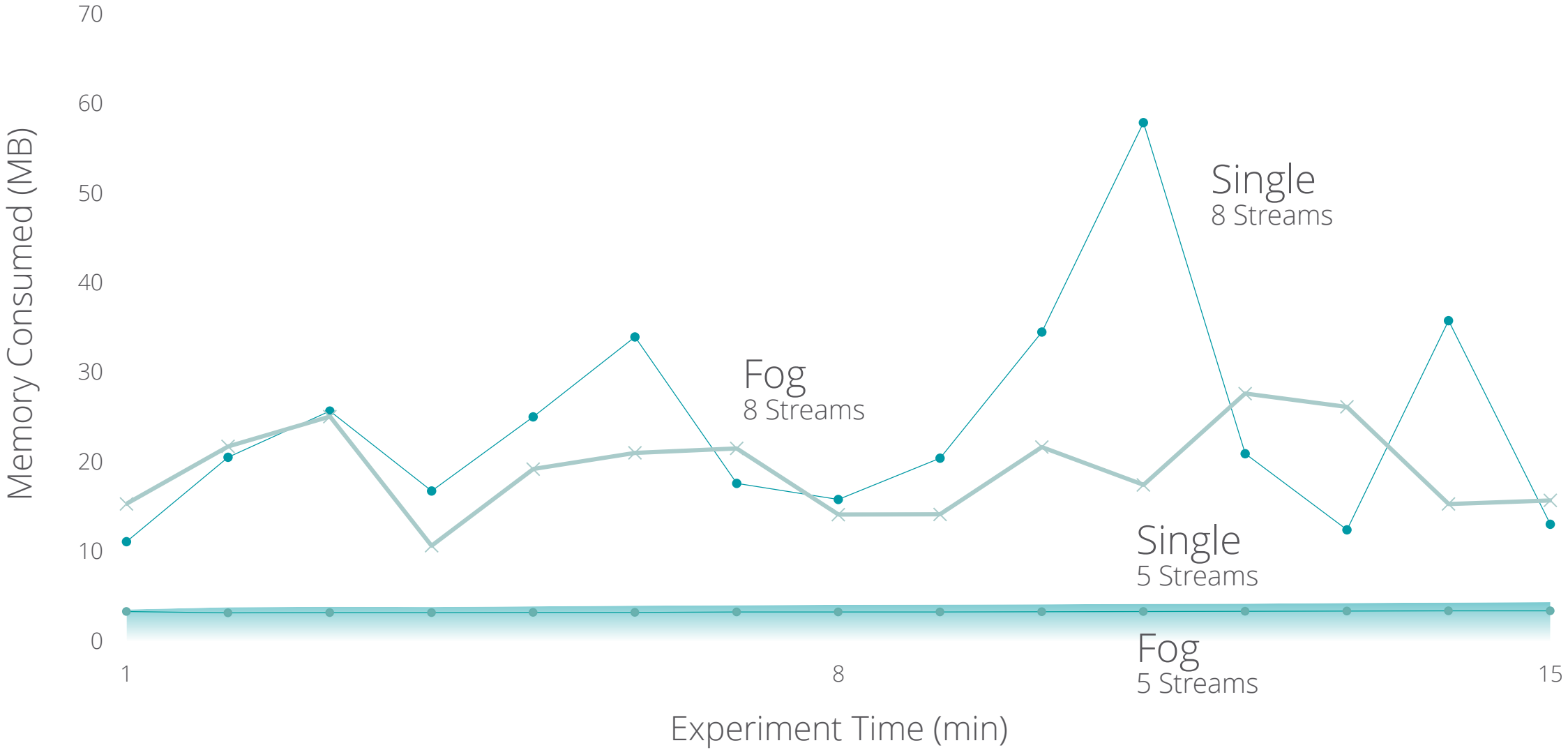
Scalability Evaluation

CityBench Query 10 (most polluted area in the city in real-time)



Fog Scalability Evaluation

CityBench Query 10 (most polluted area in the city in real-time)



Conclusions

Latency
01

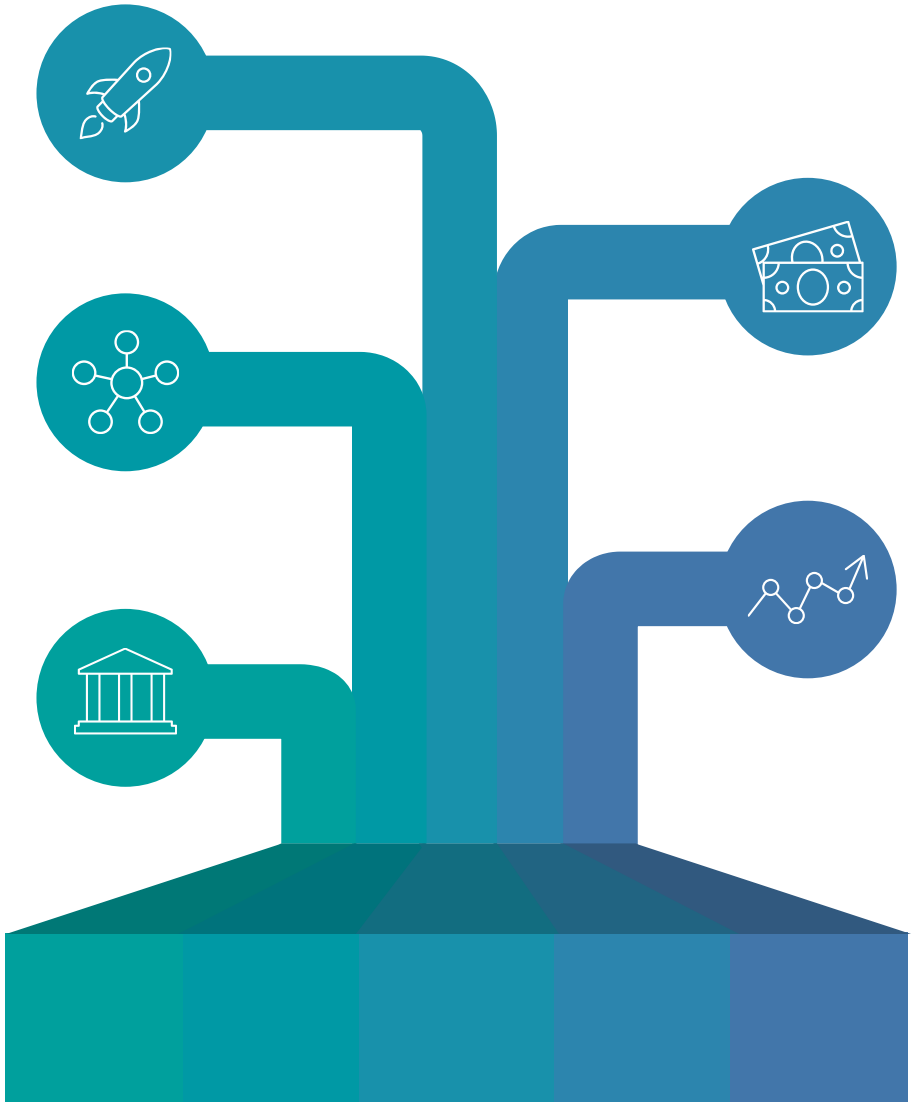
Eywa is fast and performant at stream processing

Interoperability
03

From the RDF graph model for metadata and data

Utility for the IoT
05

Data locality, offline access, data ownership



Scalability
02

Eywa Fog consumes less memory than other engines

Distributed
04

Eywa is a means for distributed Fog Computing