FACULTY OF ENGINEERING AND ARCHITECTURE

DEPARTMENT INTEC
RESEARCH GROUP IDLab

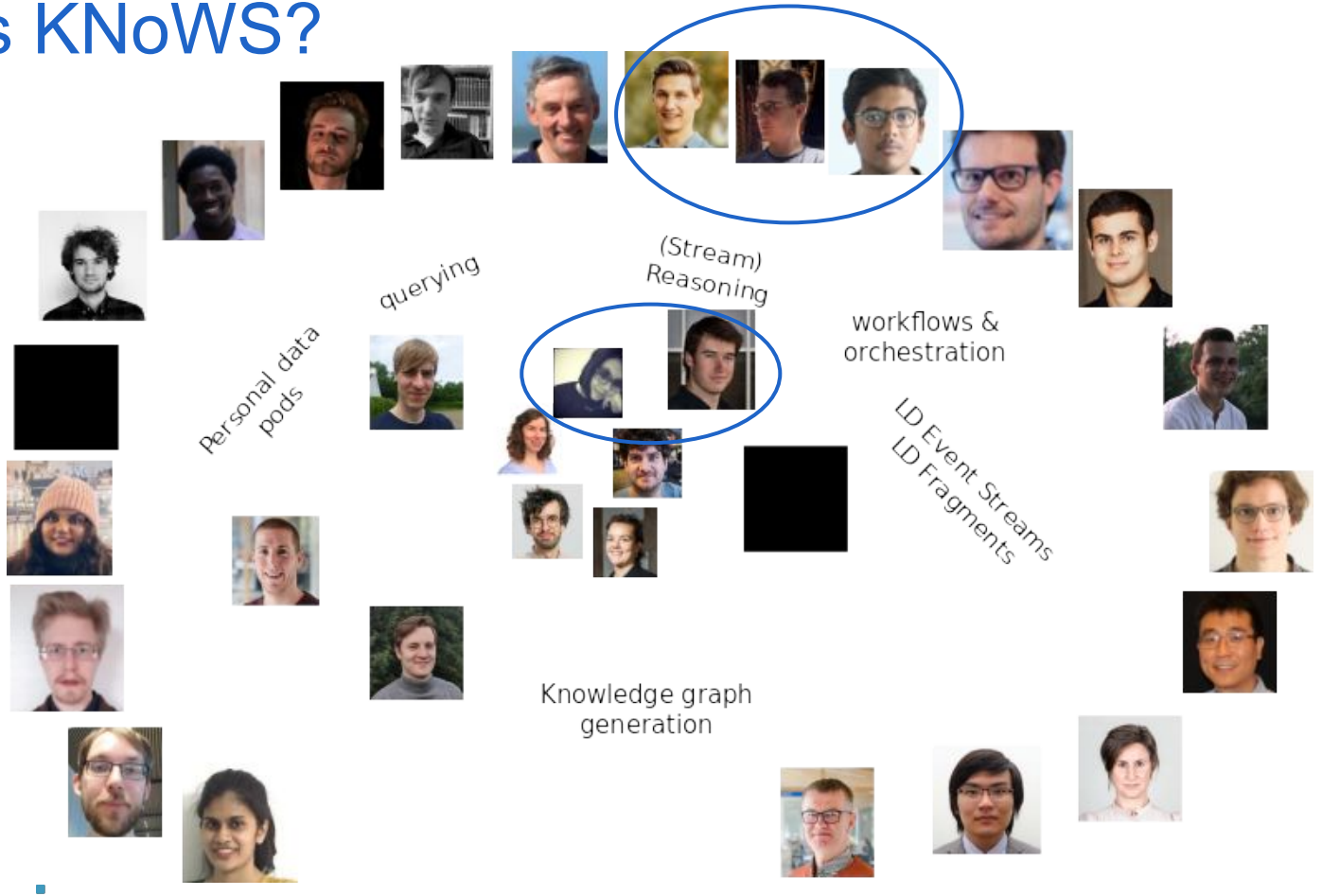
Reasoning in SOLID

Kushagra Singh Bisen & Maarten Vandenbrande & Mathijs van Noort

GHENT UNIVERSITY

imec

Who is KNoWS?



GHENT UNIVERSITY

imec

Overview

- Maarten
  - Incremental Query Aggregators in SOLID
- Kushagra
  - Aggregating Sensitive Health Data streams in SOLID
- Mathijs
  - Towards a Unifying Logic for Linked Data Streams

Incremental Query Aggregators in SOLID

GHENT UNIVERSITY

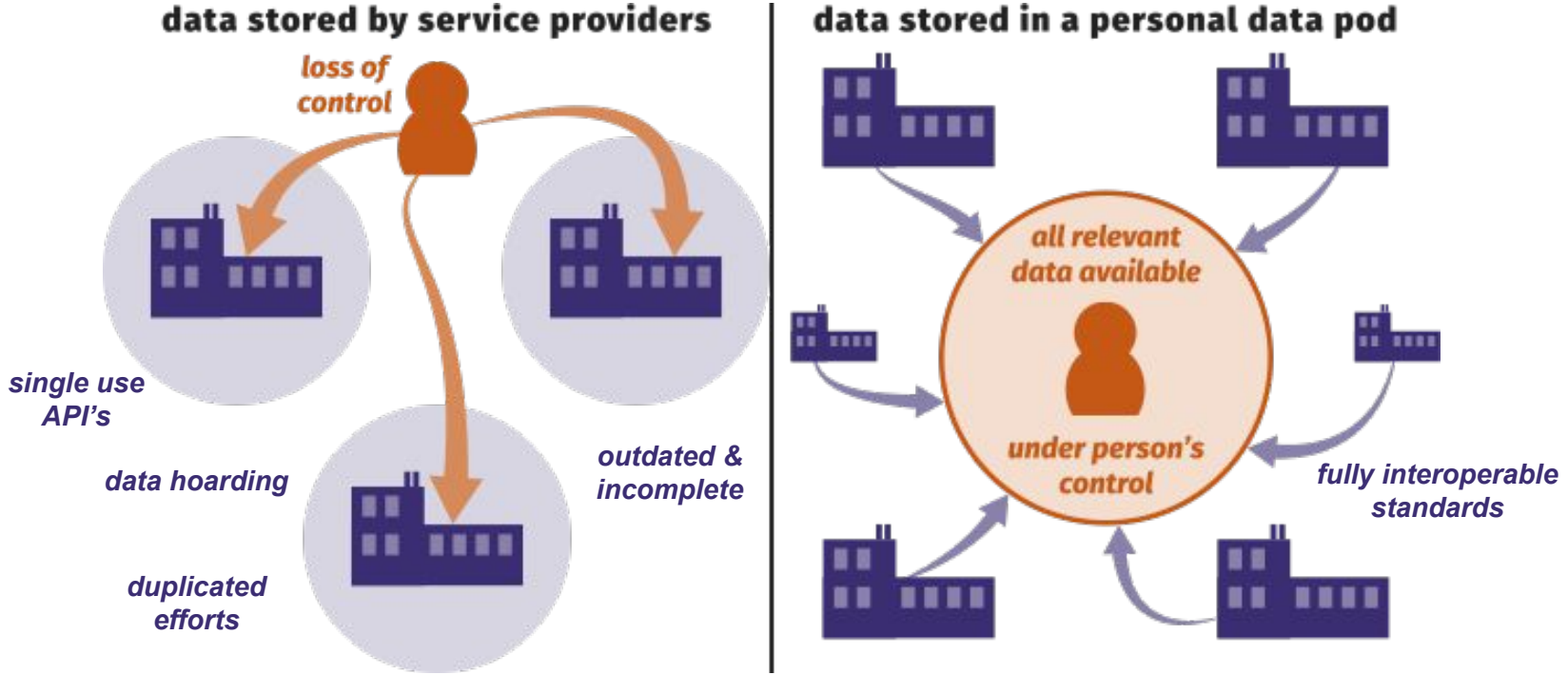
imec

SOLID

GHENT UNIVERSITY

imec

Why SOLID?

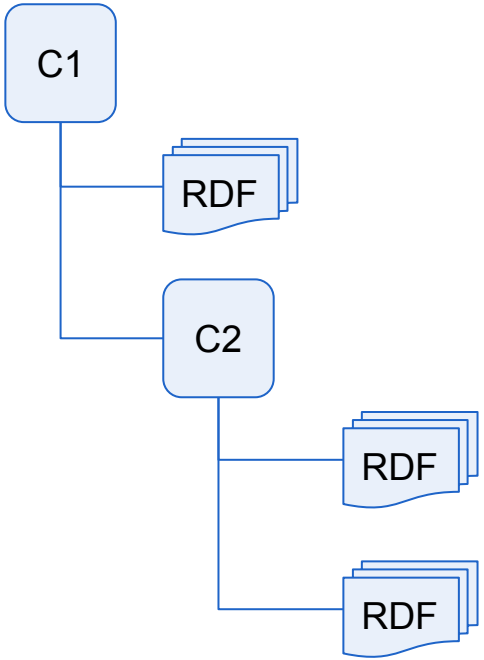


GHENT UNIVERSITY

imec

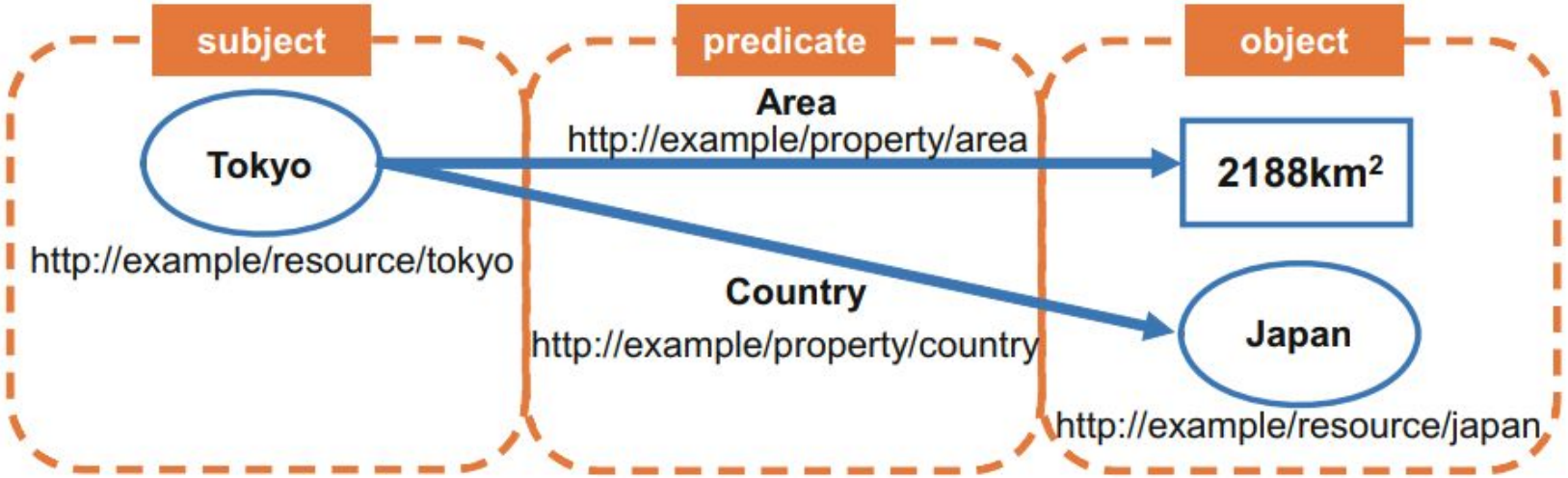
The SOLID pod

- Data vault
- Knowledge graph
- LDP: Fragmented => Files
- Add semantics to data
- RDF



RDF

<http://example/resource/tokyo> <http://example/property/area> “2188km²” .
<http://example/resource/tokyo> <http://example/property/country> <http://example/resource/japan> .

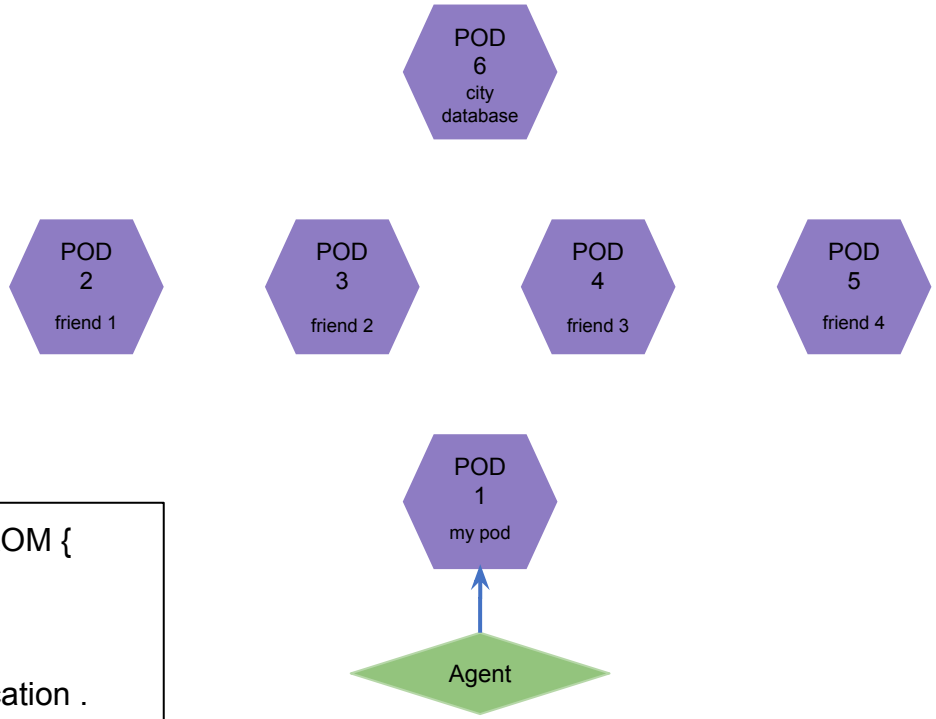


GHENT UNIVERSITY

imec

Incremental Query Aggregators

Problem with decentralized ecosystems

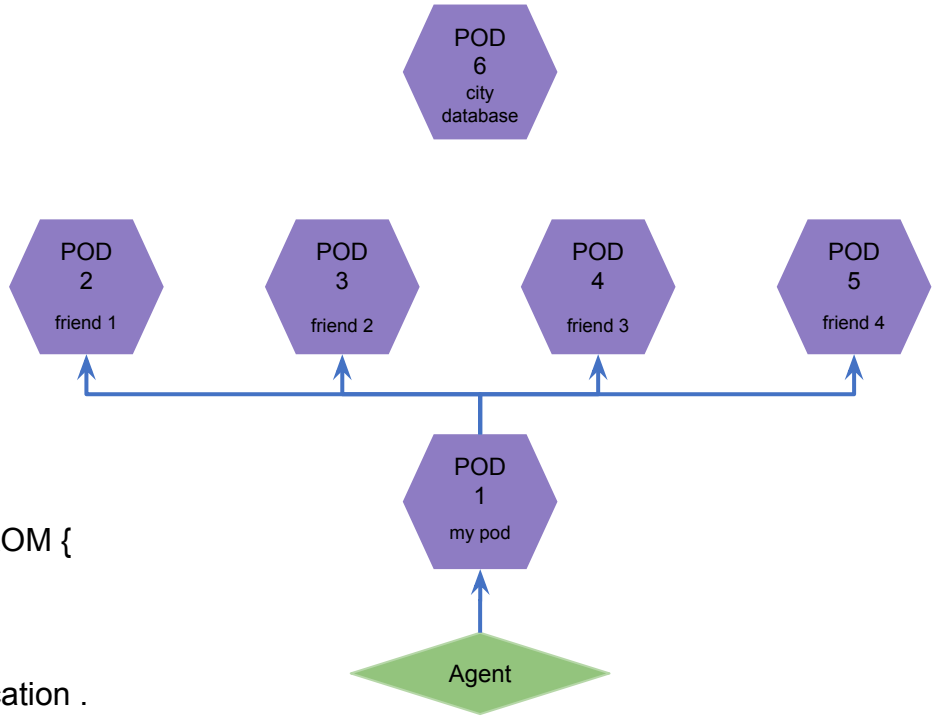



```
SELECT ?name ?cityLocation FROM {
        ?me :knows ?friend .
        ?friend :name ?name .
        ?friend :livesIn ?city .
        ?city :hasLocation ?cityLocation .
}
```

GHENT UNIVERSITY

imec

Problem with decentralized ecosystems

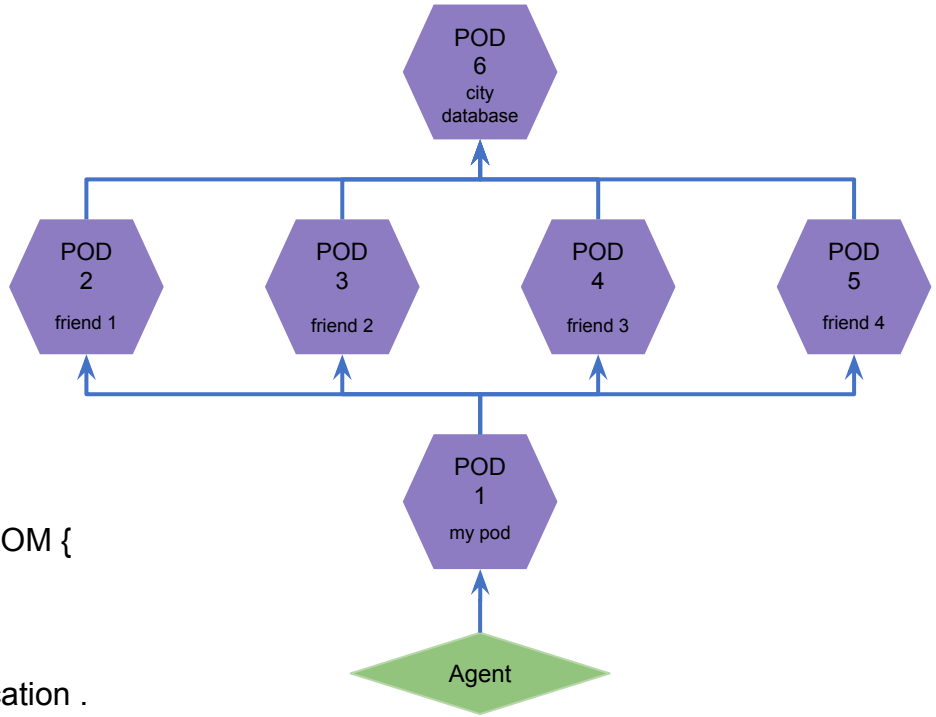



```
SELECT ?name ?cityLocation FROM {
        ?me :knows ?friend .
        ?friend :name ?name .
        ?friend :livesIn ?city .
        ?city :hasLocation ?cityLocation .
}
```

GHENT UNIVERSITY

imec

Problem with decentralized ecosystems

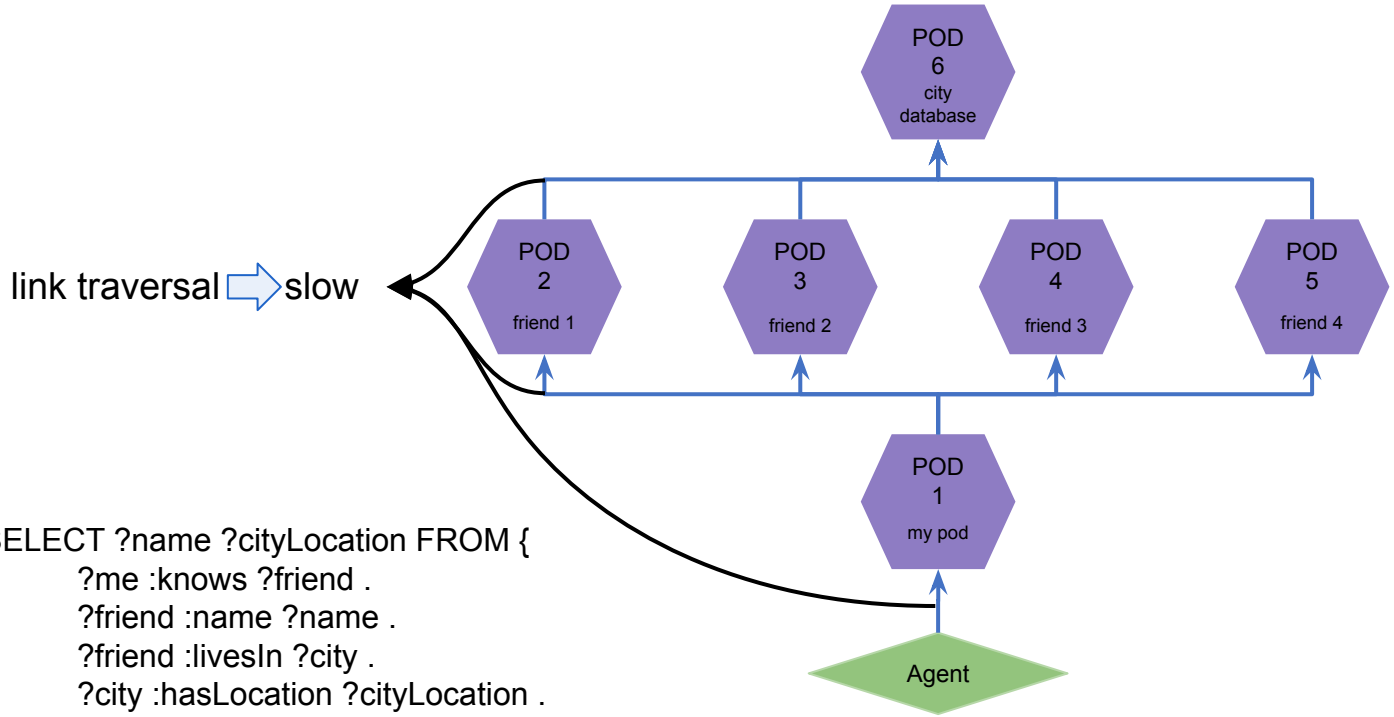



```
SELECT ?name ?cityLocation FROM {
        ?me :knows ?friend .
        ?friend :name ?name .
        ?friend :livesIn ?city .
        ?city :hasLocation ?cityLocation .
}
```

GHENT UNIVERSITY

imec

Problem with decentralized ecosystems

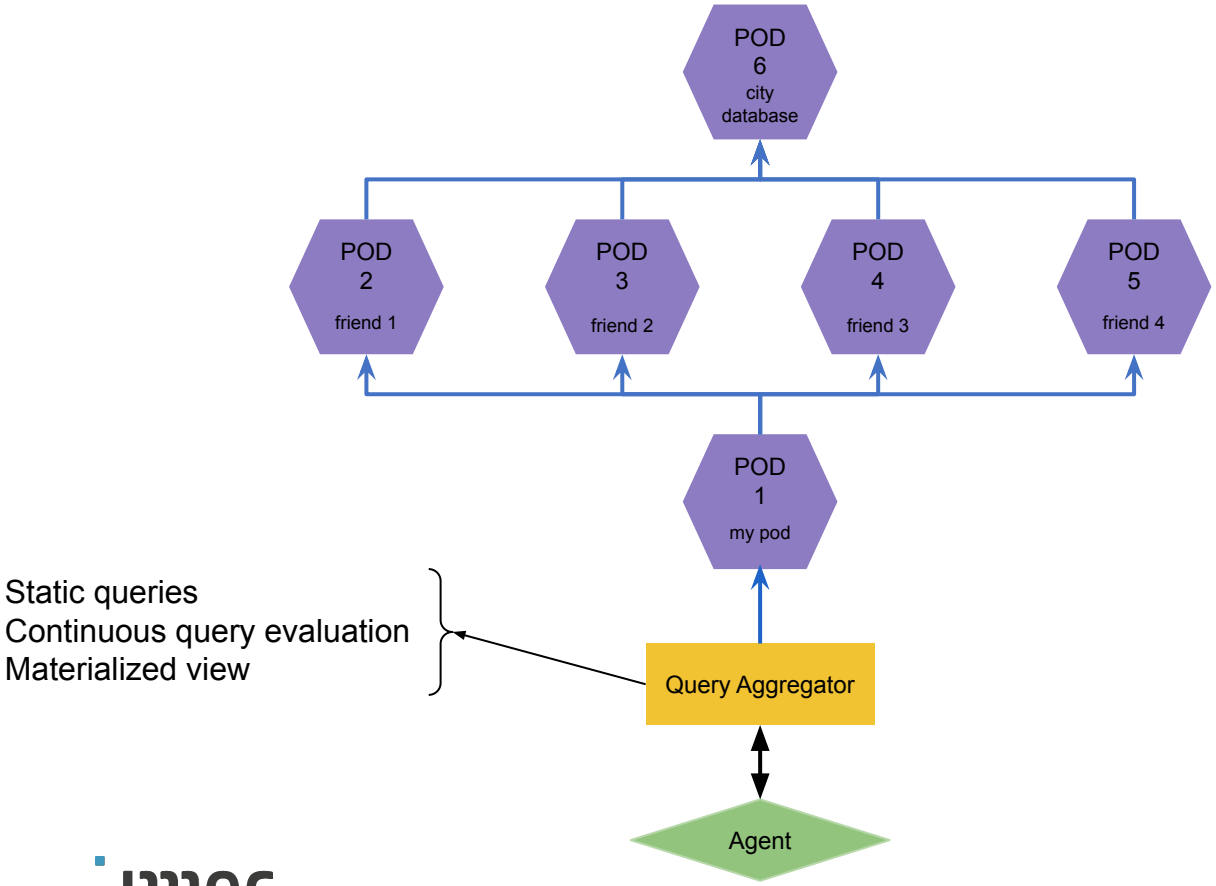



```
SELECT ?name ?cityLocation FROM {
        ?me :knows ?friend .
        ?friend :name ?name .
        ?friend :livesIn ?city .
        ?city :hasLocation ?cityLocation .
}
```

GHENT UNIVERSITY

imec

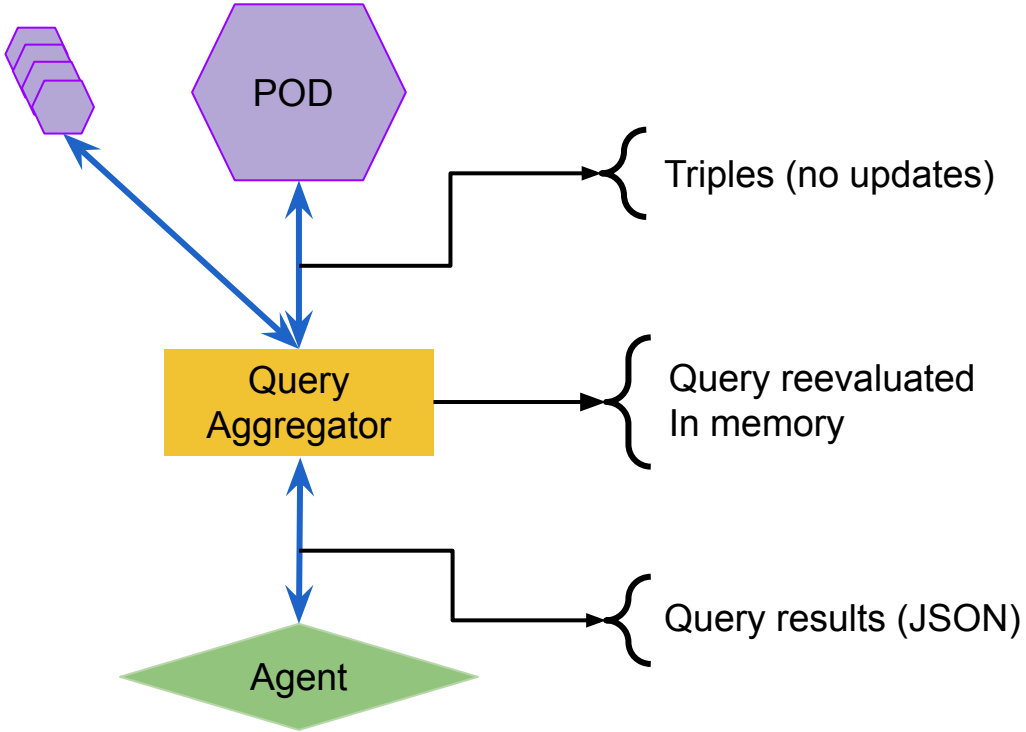
Why query aggregators?



GHENT UNIVERSITY

imec

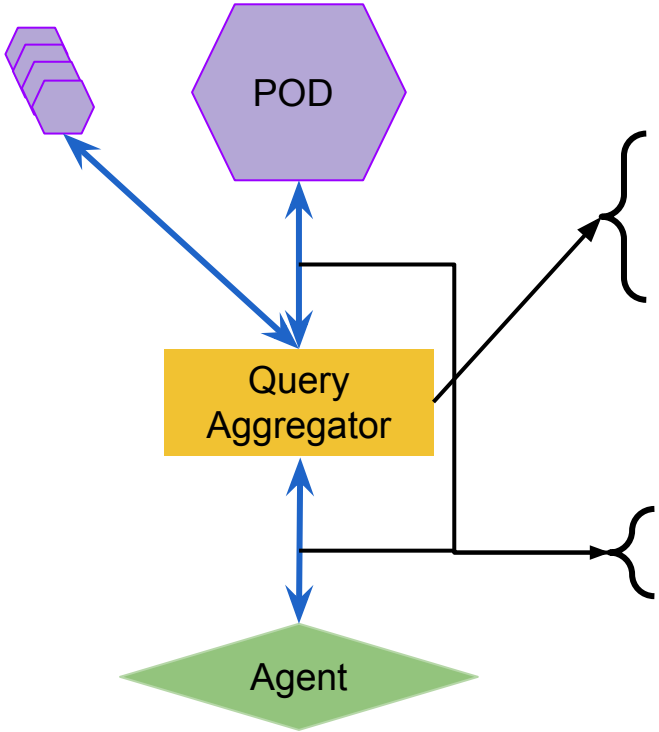
Query aggregators: Current state



GHENT UNIVERSITY

imec

Query aggregators: Future work



Incremental query evaluation (additions/deletions)
Schema alignment (reasoning)
Caching intermediate results (save query state to disk)

Universal ontology for result/updates
Description of data

GHENT UNIVERSITY

imec

Conclusion

- Decentralized ecosystems ⇨ slow to query
- Query aggregators
  - Incremental query evaluation

FACULTY OF ENGINEERING AND ARCHITECTURE

Maarten Vandenbrande

PhD Researcher
IDLab

E maarten.vandenbrande@ugent.be
M +32 494335741

www.ugent.be

GHENT UNIVERSITY

imec

FACULTY OF ENGINEERING AND ARCHITECTURE

DEPARTMENT INTEC
RESEARCH GROUP IDLAB

Stream Reasoning Workshop 2022, Amsterdam

Aggregating Sensitive Health Data Streams in SOLID

Kushagra Singh Bisen, Pieter Bonte and Femke Ongenae

GHENT UNIVERSITY

imec

Solid is amazing for personal data, but what about streams?

GHENT UNIVERSITY

imec

Wearables with SOLID : A use-case

Wearables produce sensitive data streams.

Makes sense to store them with SOLID.

streaming data

Pod 1

Person 1

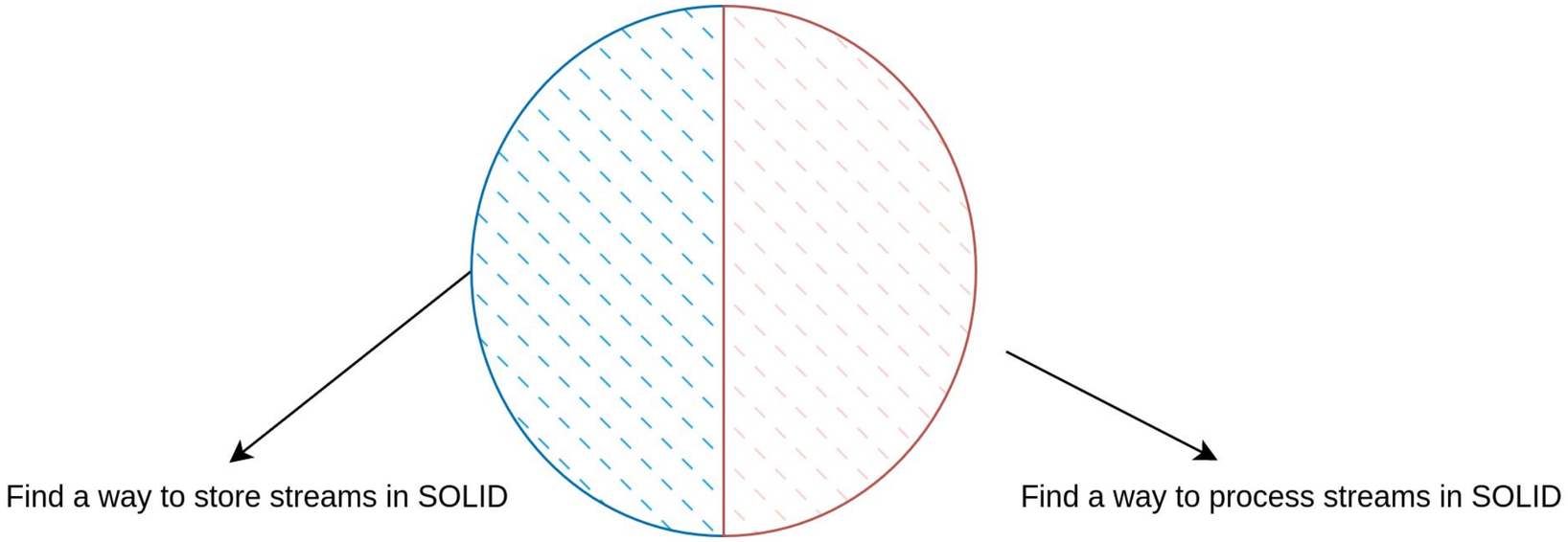
streaming data

Pod 2

Person 2

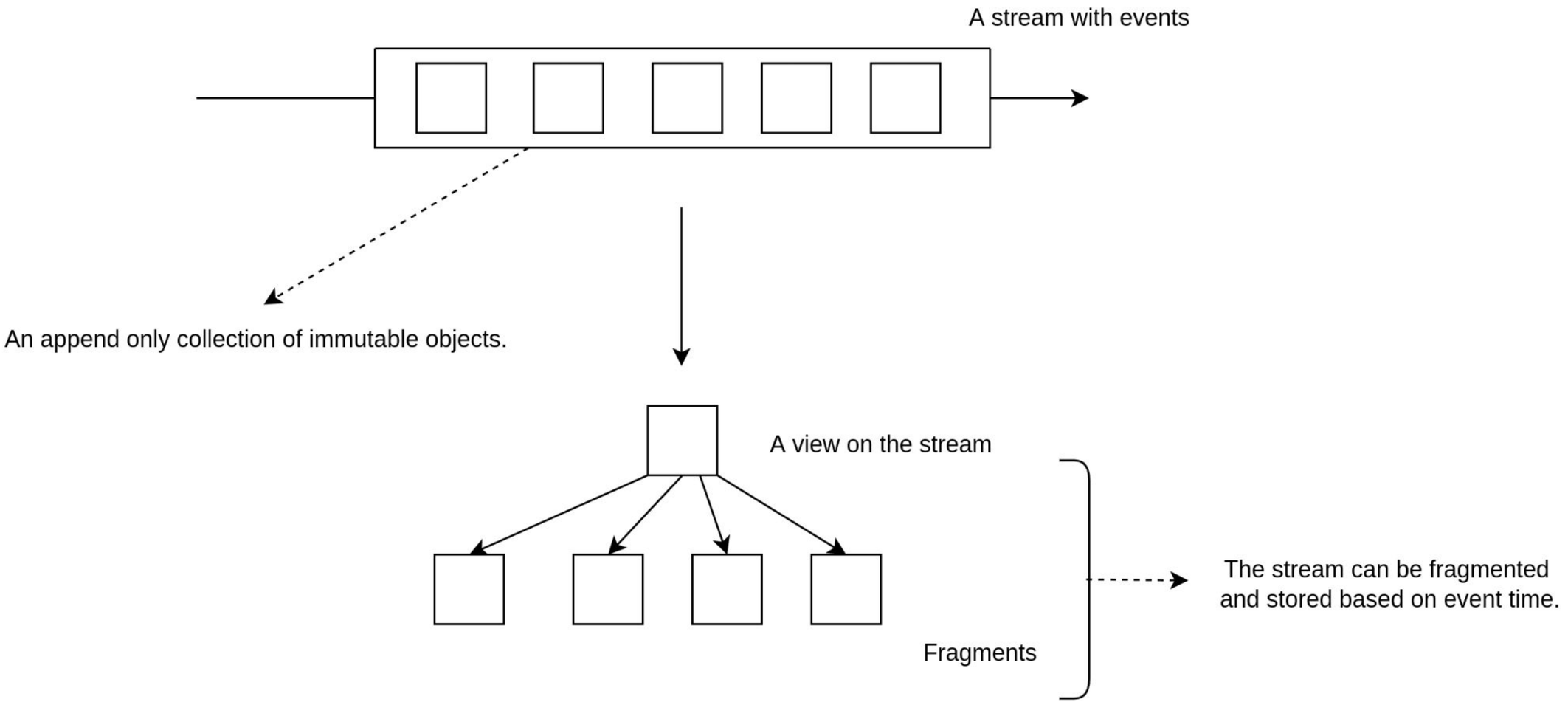
To complete the circle to accomplish this,

We will need both parts of the circle.



Storing Streams in Solid.

Linked Data Event Streams (LDES)



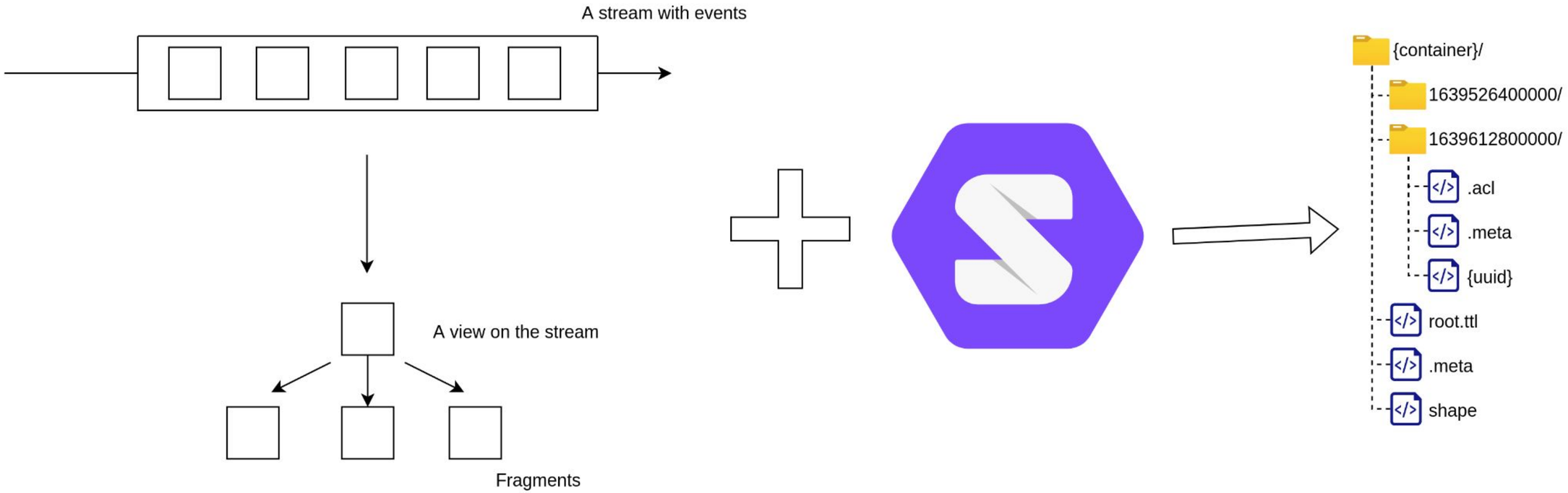
GHENT UNIVERSITY

imec

Storing Streams in Solid.

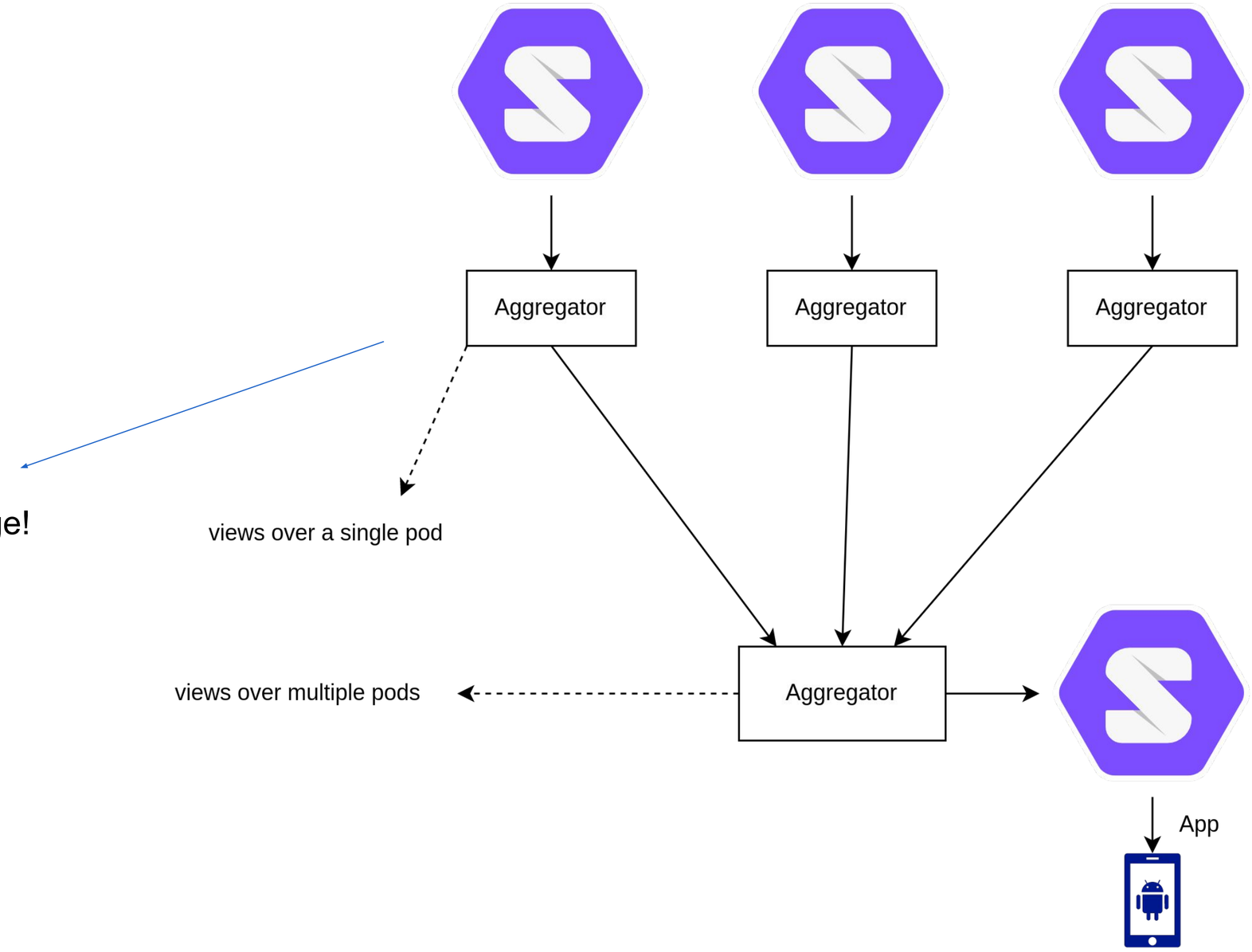
Solid is currently file based.

Solid uses LDES with LDP (Linked Data Platform)

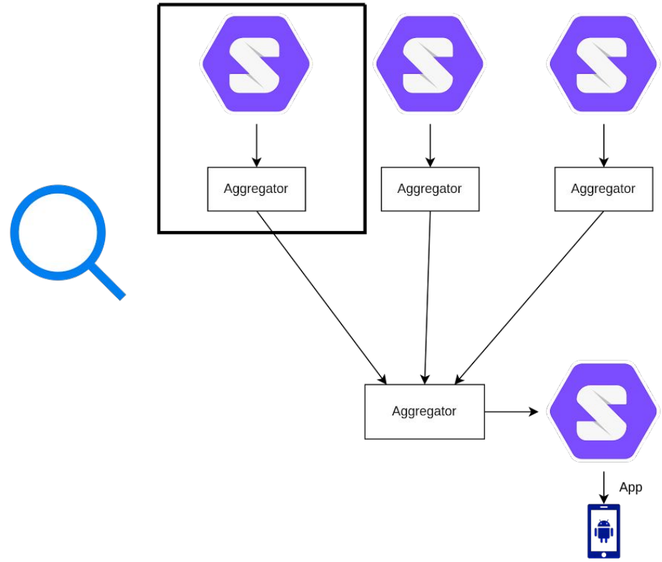


Aggregating Streams with SOLID.

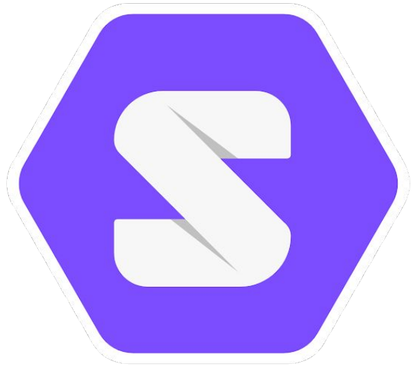
Let’s explain this image!



Local Aggregations



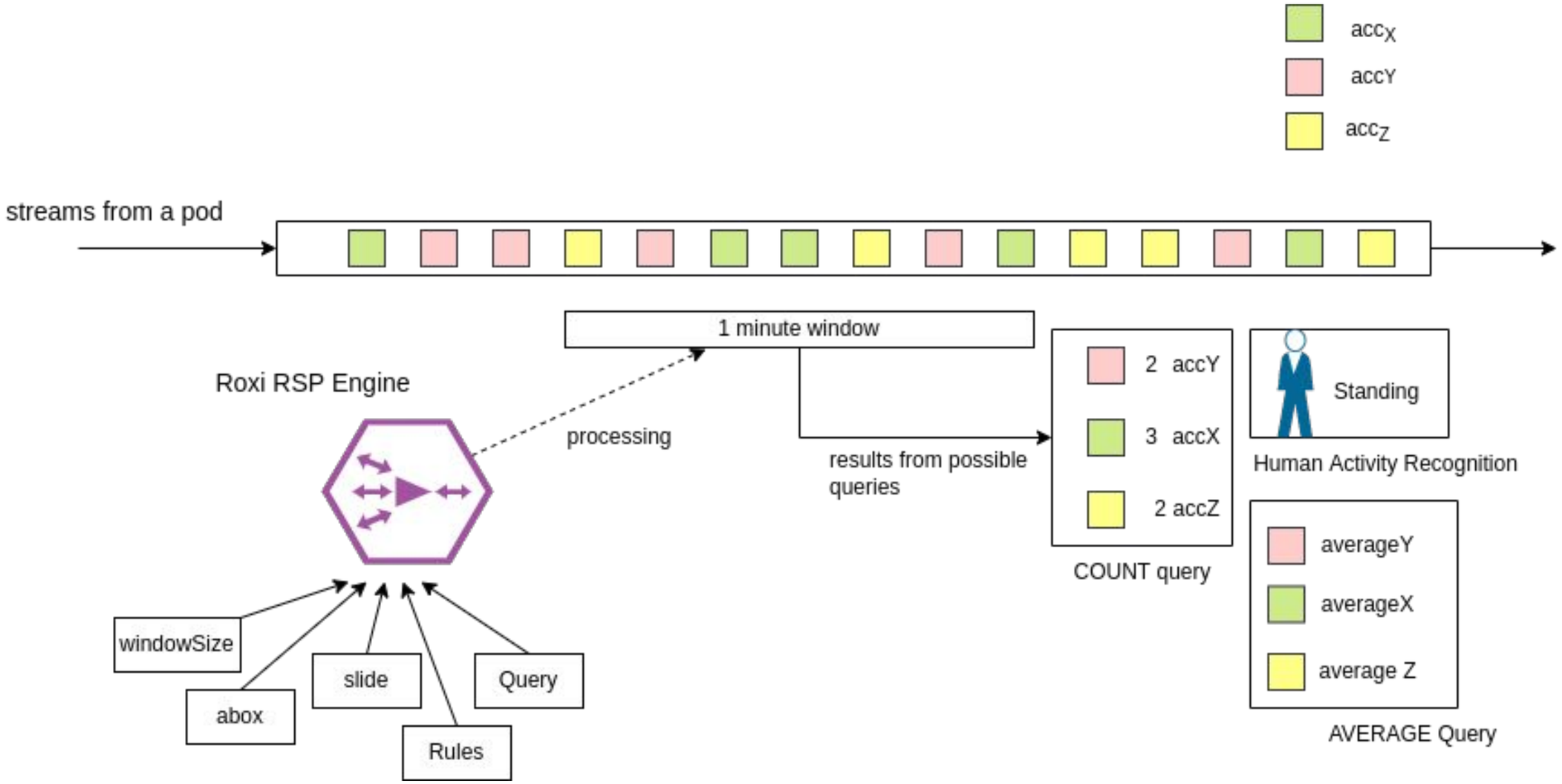
:EventStream tree:member :obsX
:obsX dct:isVersionOf saref:Measurement;
saref:hasTimestamp time;
saref:hasValue "0.456";
saref:measurementMadeBy :Wearable1
saref:relatesToProperty :accelerationX .



Aggregator

:EventStream tree:member :aggX
:aggX :isA :Aggregation;
:position **:Standing;**
:fromPod PodX .

Handling the Stream.

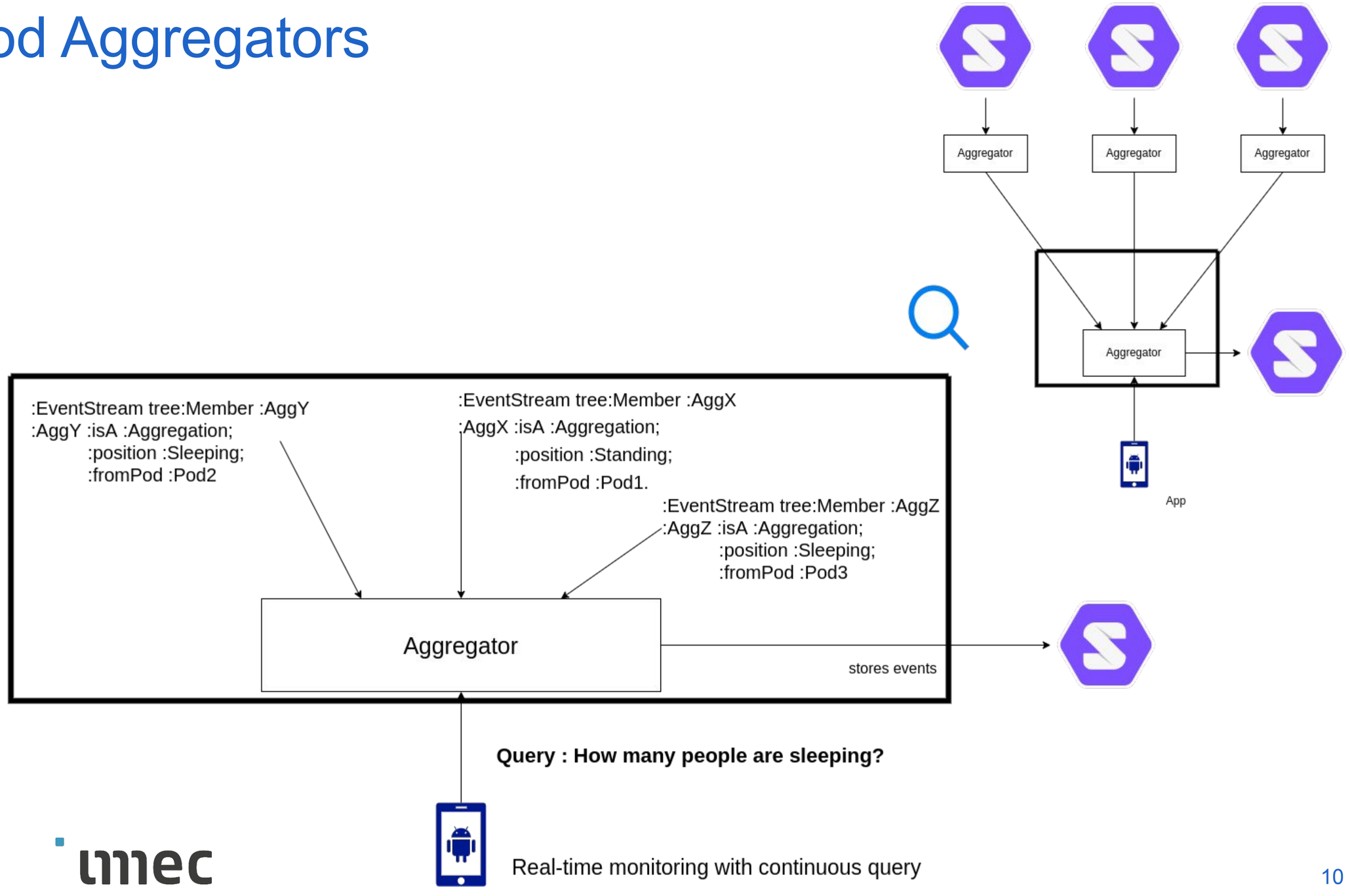


https://github.com/pbonte/roxi

GHENT UNIVERSITY

imec

Multipod Aggregators



Real-time monitoring with continuous query

Future Steps

My research will focus on,

- an efficient optimal aggregator.
- scalability.

Other challenges our lab is focussing,

- synchronization of the pods.
- specifying policies for aggregators.
- dealing with contradicting data.

GHENT UNIVERSITY

imec

FACULTY OF ENGINEERING
AND ARCHITECTURE

Kushagra Singh Bisen

PhD researcher
Internet Technology and Data Science Lab

E: kushagrasingh.bisen@ugent.be

@argahsuknesib

argahsuknesib.github.io

GHENT
UNIVERSITY

imec

FACULTY OF ENGINEERING AND ARCHITECTURE

STREAM REASONING WORKSHOP 2022 — AMSTERDAM

TOWARDS A UNIFYING LOGIC FOR LINKED DATA STREAMS

Mathijs van Noort — Mathijs.vanNoort@UGent.be

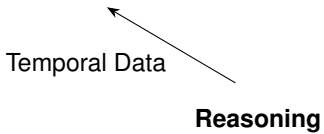
GHENT UNIVERSITY

imec

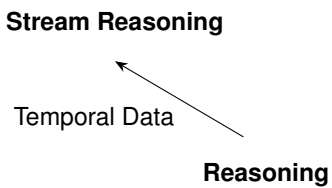
Our Context

Reasoning

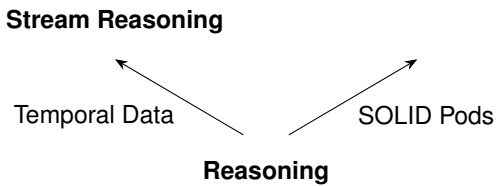
Our Context



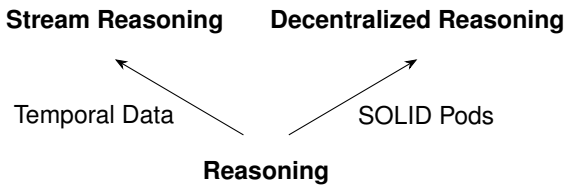
Our Context



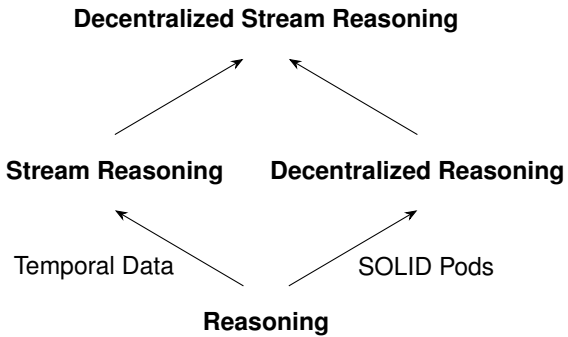
Our Context



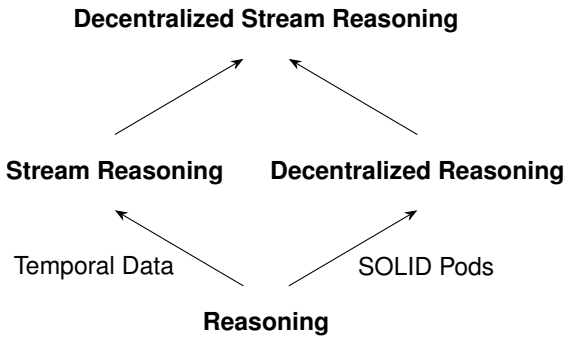
Our Context


Stream Reasoning
Decentralized Reasoning
Temporal Data
SOLID Pods
Reasoning

Our Context

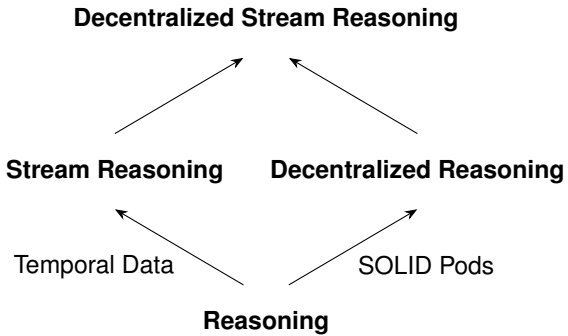

Decentralized Stream Reasoning
Stream Reasoning
Decentralized Reasoning
Temporal Data
SOLID Pods
Reasoning

Our Context



$\Rightarrow$ **What logical framework can accommodate these complex needs?**

Our Context

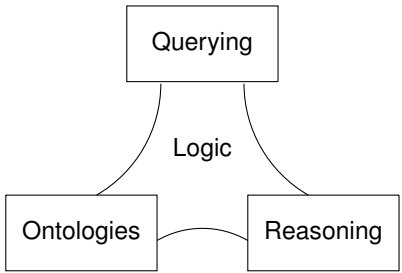


⇒ **What logical framework can accommodate these complex needs?**

*RDF has no negation, so preferably a framework without negation

Why One Logic Framework for All ?

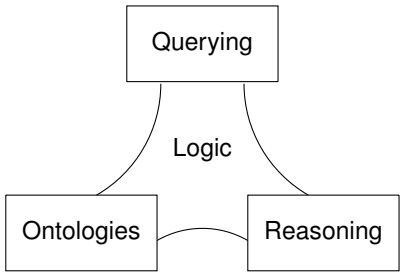
Interoperability!



WHY ONE LOGIC FRAMEWORK FOR ALL ?

Interoperability!

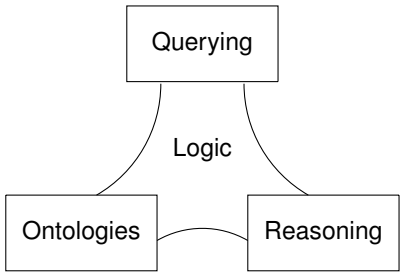
- Different Pods $\Rightarrow$ Different Ontologies



WHY ONE LOGIC FRAMEWORK FOR ALL ?

Interoperability!

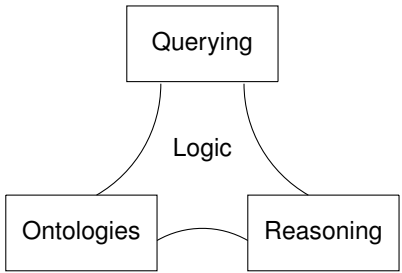
- Different Pods $\Rightarrow$ Different Ontologies
- Different Agents $\Rightarrow$ Different Querying & Reasoning



WHY ONE LOGIC FRAMEWORK FOR ALL ?

Interoperability!

- Different Pods $\Rightarrow$ Different Ontologies
- Different Agents $\Rightarrow$ Different Querying & Reasoning
- Logic to mediate between different languages/assumptions/...



Expressing Time in Logic

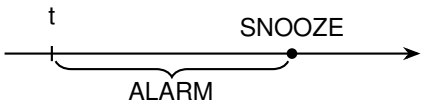
Some Semantics

Expressing Time in Logic

Some Semantics

Point-based

"My ALARM goes of until I hit SNOOZE"



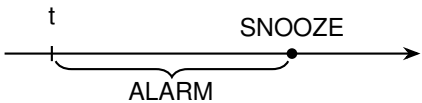
ALARM until SNOOZE

Expressing Time in Logic

Some Semantics

Point-based

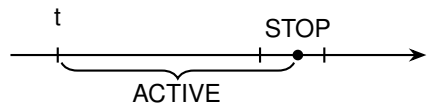
"My ALARM goes of until I hit SNOOZE"



ALARM until SNOOZE

Interval-based

"My alarm stays ACTIVE until I hit STOP after it starts ringing again"



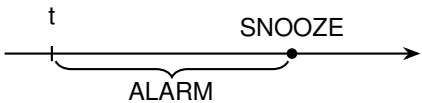
ACTIVE $\text{until}_{[5,6]}$ STOP

Expressing Time in Logic

Some Semantics

Point-based

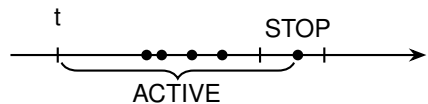
"My ALARM goes of until I hit SNOOZE"



ALARM until SNOOZE

Interval-based

"My alarm stays ACTIVE until I hit STOP after it starts ringing again"



ACTIVE $\text{until}_{[5,6]}$ STOP

Temporal Operators in Temporal Reasoning

Pointwise Semantics

since and **until** operators $\mathcal{S}, \mathcal{U}$

$\diamondplus\, \psi \equiv \top \mathcal{U} \psi$

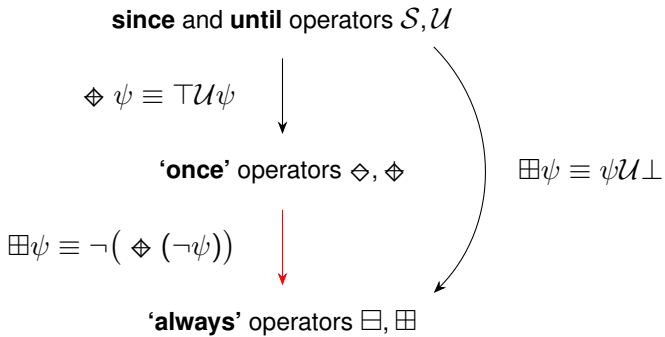
'once' operators $\diamondminus$, $\diamondplus$

$\boxplus \psi \equiv \neg\big(\diamondplus\, (\neg\psi)\big)$

'always' operators $\boxminus$, $\boxplus$

Temporal Operators in Temporal Reasoning

Pointwise Semantics



Temporal Operators in Temporal Reasoning

Interval-based Semantics

since and **until** operators $\mathcal{S}_I, \mathcal{U}_I$

$\diamondplus_I \psi \equiv \top \mathcal{U}_I \psi$

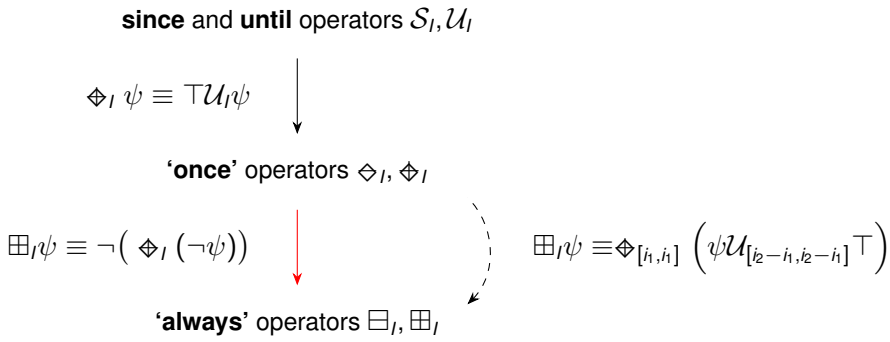
'once' operators $\diamondminus_I, \diamondplus_I$

$\boxplus_I \psi \equiv \neg\big(\diamondplus_I (\neg\psi)\big)$

'always' operators $\boxminus_I, \boxplus_I$

Temporal Operators in Temporal Reasoning

Interval-based Semantics



Future Leads and Ambitions

- Build logic framework for SR without negation
- Decentralized Reasoning
- Combine Stream and Decentralized Reasoning frameworks

FACULTY OF ENGINEERING
AND ARCHITECTURE

Mathijs van Noort

PhD Researcher

Mathijs.vanNoort@UGent.be

Mathijs van Noort

@MathijsNoort

Ghent University – Internet Technology and Data Science Lab

GHENT UNIVERSITY

imec