

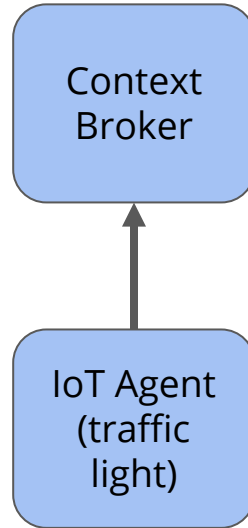
Comparing a polling and push-based approach for live Open Data interfaces

*2nd February – COOCK workshop
Connecting, storing and publishing sensor data*

Brecht Van de Vyvere
IDLab (IMEC – Ghent University)

Publish live data from traffic light “sensor”?

Let's use a context broker!





Principes

Onder deze principes is meer achtergrondinformatie terug te vinden, samen met de types data waarop dit charter betrekking kan hebben. In [deze begeleidende presentatie](#) wordt elk principe in meer detail toegelicht en worden er voorbeelden gegeven. Het staat geïnteresseerde besturen vrij deze presentatie te hergebruiken.

1. § **Open by default:** data die door, in of over de stad gecapteerd worden, worden standaard als open data ter beschikking gesteld voor hergebruik.
2. § **Comply or explain:** wanneer data niet opengesteld worden, dient er een beargumenteerde verklaring te worden gegeven waarom dit zo is.
3. § **Only once:** gegevens worden slechts één keer verzameld en zo dicht mogelijk bij de bron gepubliceerd (decentraal).



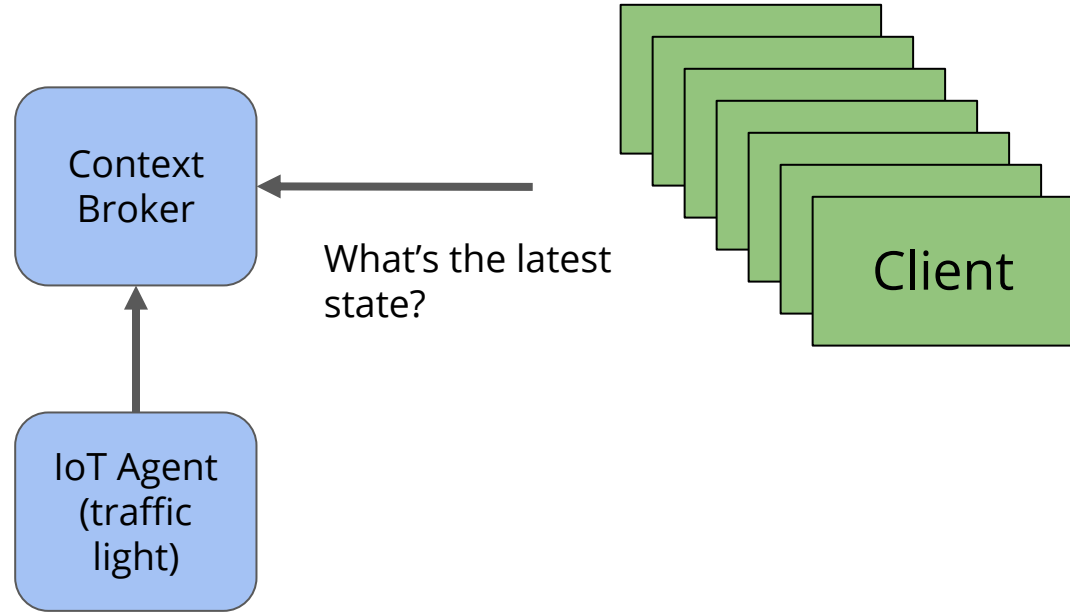
Principes

Onder deze principes is meer achtergrondinformatie terug te vinden, samen met [deze begeleidende presentatie](#) wordt elk principe in meer detail toegelicht en worden deze presentatie te hergebruiken.

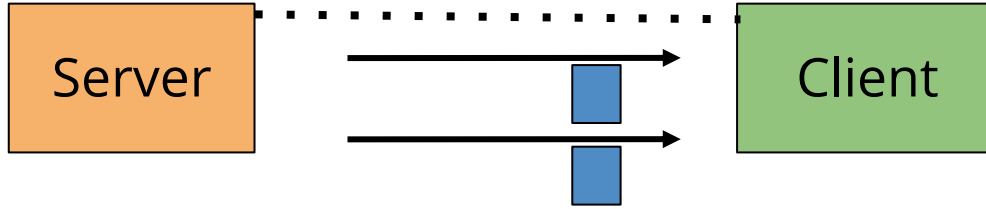
No API keys or tokens
-> explosion of
clients/applications

1. § **Open by default:** data die door, in of over de stad gecapteerd worden, worden standaard als open data ter beschikking gesteld voor hergebruik.
2. § **Comply or explain:** wanneer data niet opengesteld worden, dient er een beargumenteerde verklaring te worden gegeven waarom dit zo is.
3. § **Only once:** gegevens worden slechts één keer verzameld en zo dicht mogelijk bij de bron gepubliceerd (decentraal).

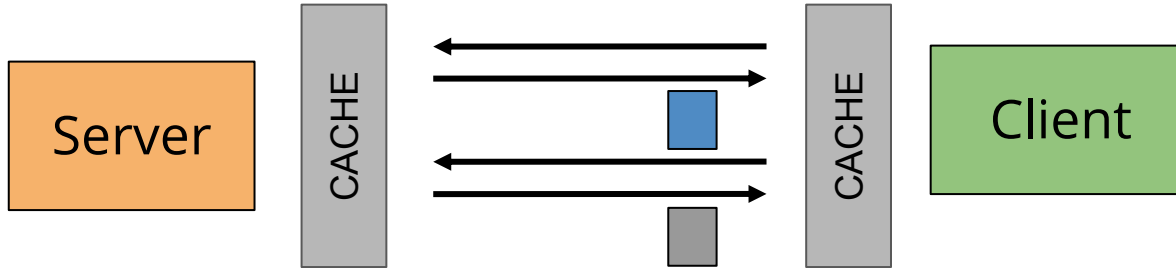
How does context broker scale in an open environment?



Publish/subscribe



HTTP polling



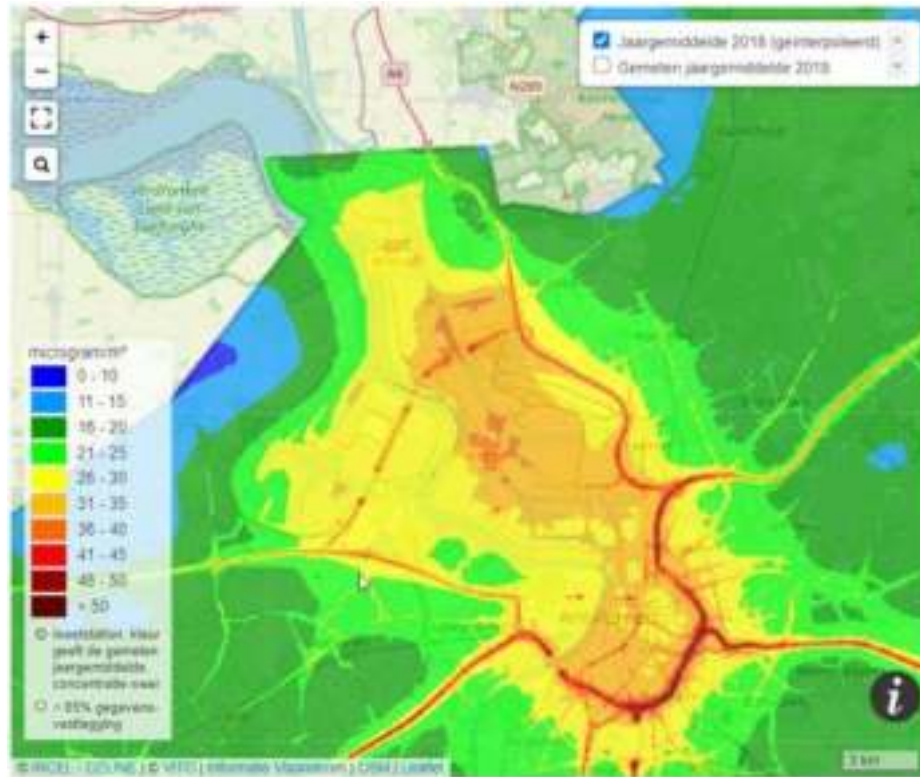
■ = latest update

■ = latest already retrieved


**Answer in chat:
would you use pushing or polling?**



**Answer in chat:
would you use pushing or polling?**



Answer in chat: would you use pushing or polling?

 **78** Zevergem - De Pinte - Gent - Lochristi - Lokeren

[Other directions](#) [Save line](#) [Receive email updates](#)

[Journeys](#) [Timetable](#) [Diversions](#)

Journey on Tuesday at 11:33

[Refresh](#)

Zwinaarde	Bollebergen	↓
Zwinaarde	Klosse	↓
Zevergem	Grote Steenweg	↓
Zevergem	Den Beer	15'
De Pinte	Mieregoedstraat	18'
De Pinte	Minnestede	19'
De Pinte	Europaplein	20'

Few push-based APIs available on Open Data portals

Country	Datasets	Update interval	Update mechanism
Belgium	Vehicles position (Public transport MIVB)	20s	Polling
Belgium	Bicycle counter	realtime	Polling
Belgium	Park+rides	realtime	Polling
France	Parking and bicycle stations availability	60s	Polling
Sweden	Notifications about Lightning Strikes	realtime	Push-based
Ireland	weather station information, the Irish National Tide Gauge Network	3600s	Polling
UK	River level data	900s	Polling
UK	Cycle hire availability & arrival predictions (Transport for London Unified API)	300s	Polling
UK	Arrival predictions (Transport for London Unified API)	realtime	Push-based
U.S.	Real-Time Traffic Incident Reports of Austin-Travis County	300s	Polling
U.S.	True Time API (arrival information and location of public transport vehicles)	realtime	Polling
U.S.	Current Bike Availability by Station (Nextbike)	300s	Polling
U.S.	USGS Streamflow Stations	24h	Polling
U.S.	NOAA water level (tidal) data of 205 Stations for the Coastal United States and Other Non-U.S. Sites	360s	Polling
U.S.	National Renewable Energy Laboratory [20]	60s	Polling
U.S.	RTC MetStation real time data	360s	Polling
U.S.	Seattle Real Time Fire 911 Calls	300s	Polling

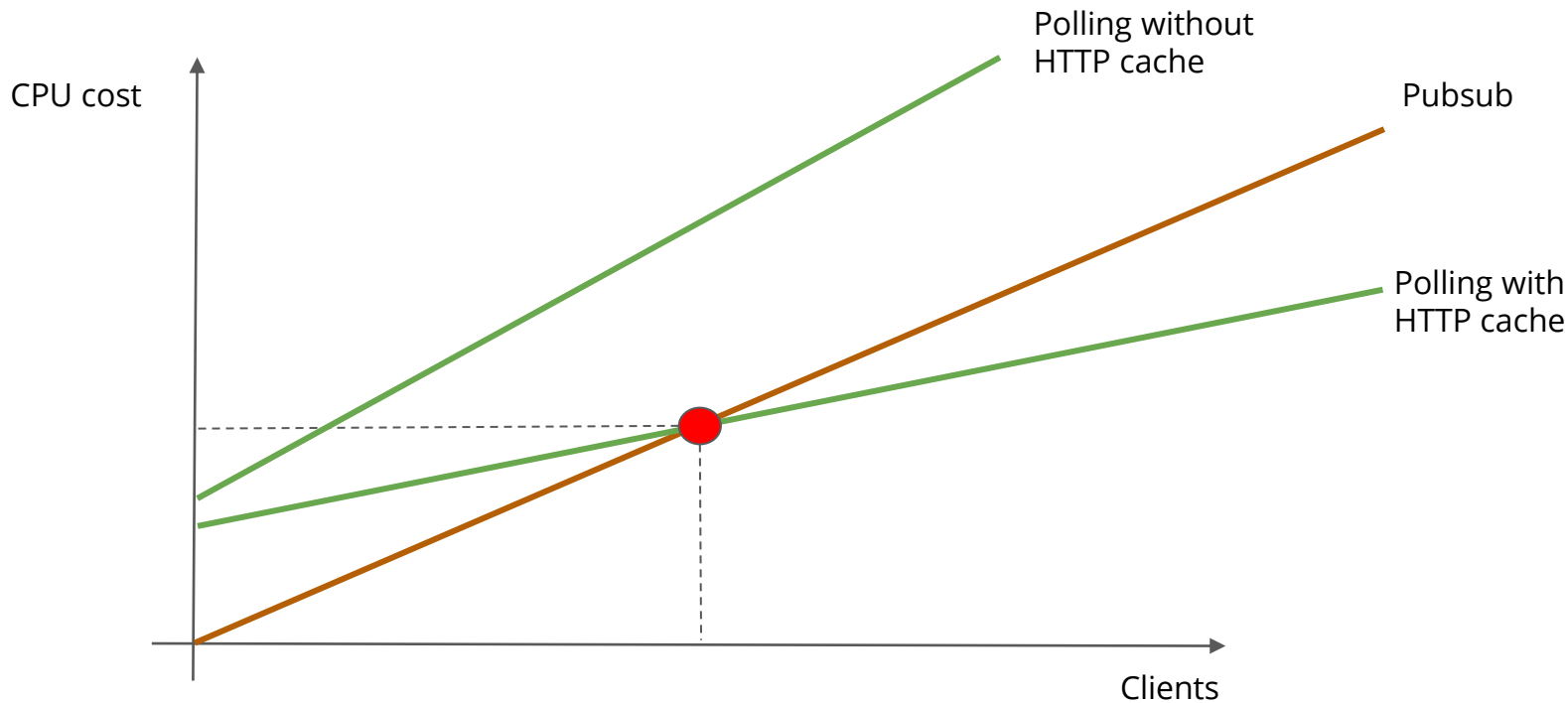
Use pushing for lowest latency*, but...

How does the server scale with thousands of clients?

Is polling cheaper for the server than pushing?

* Pimentel, V., Nickerson, B.G.: Communicating and Displaying Real-Time Data with WebSocket. IEEE Internet Computing. 16, 45–53 (2012).

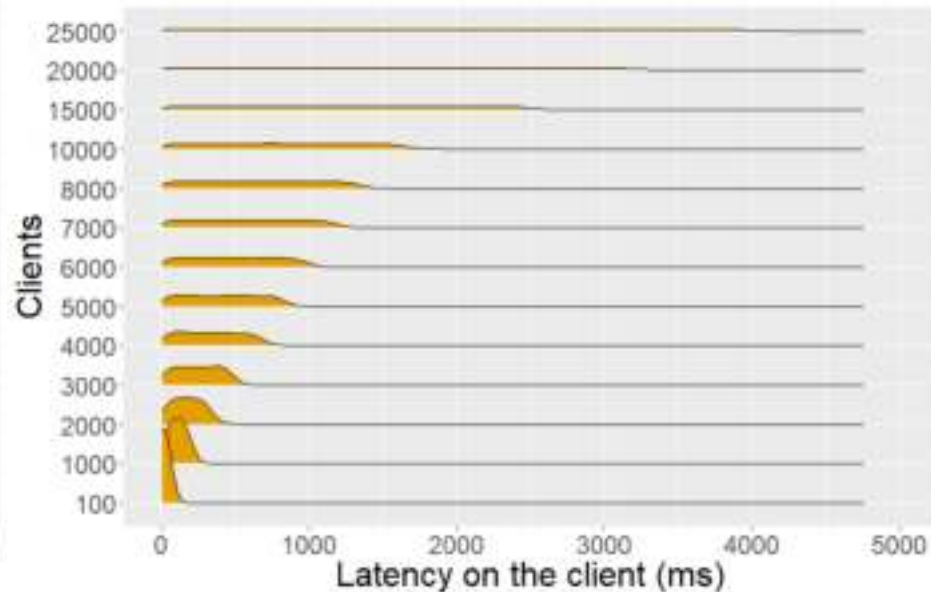
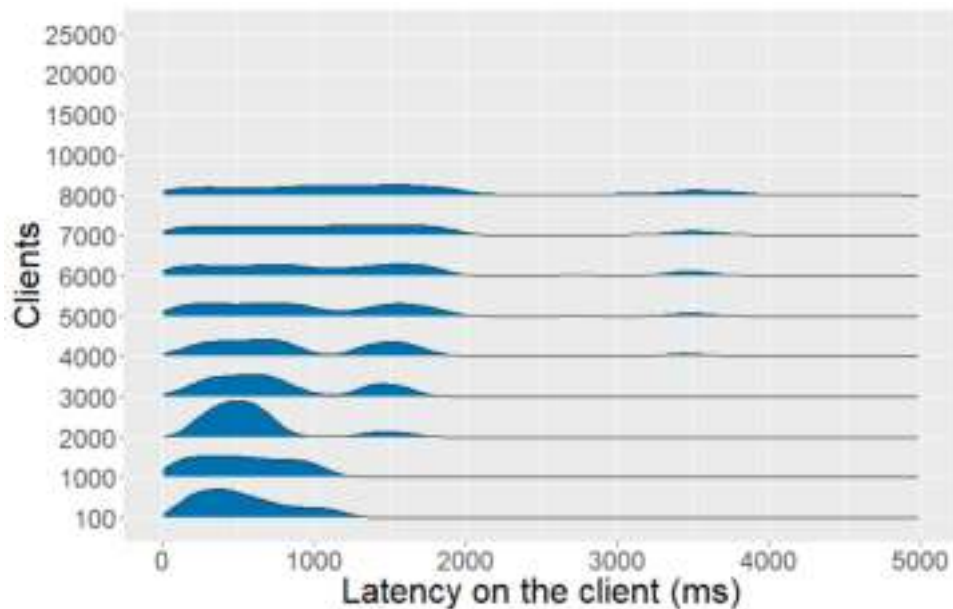
Will HTTP caching outperform pushing?



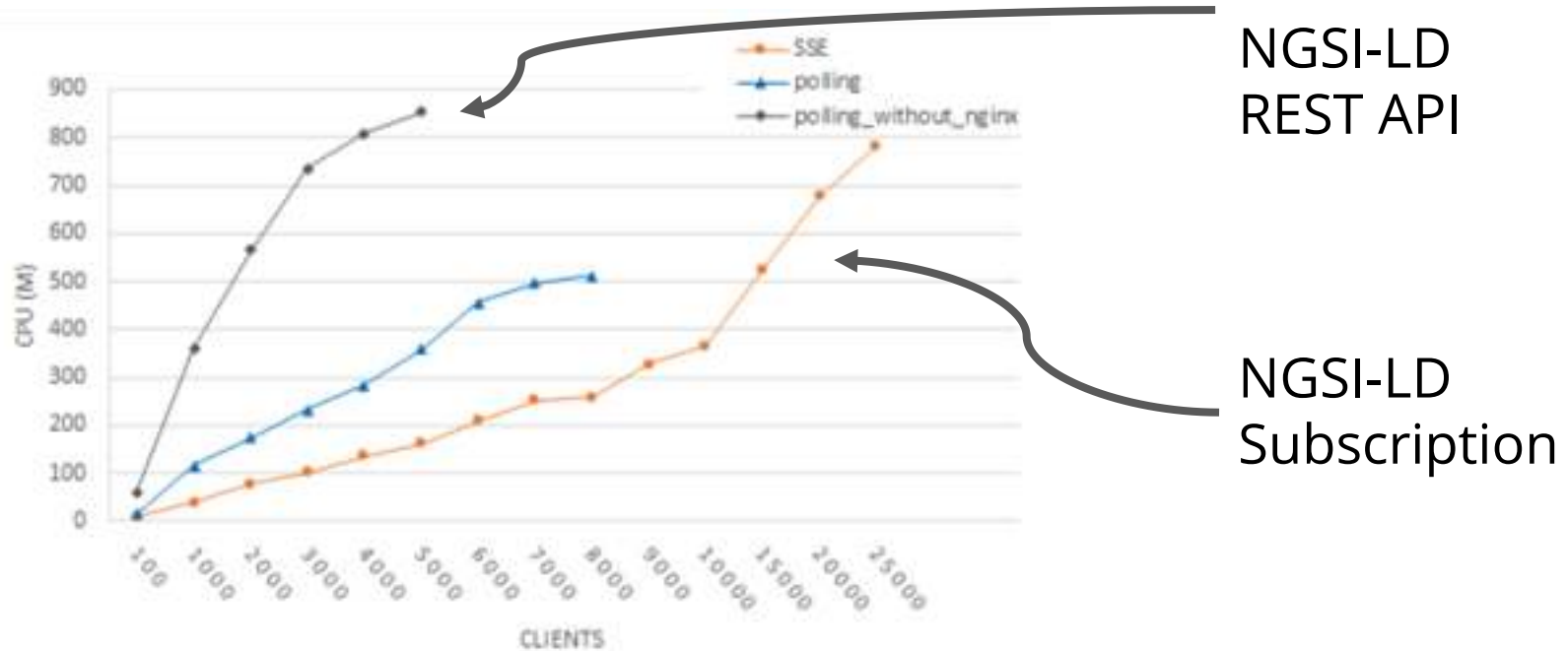
Pushing has higher scalability and lower maximal latency

Polling with HTTP cache

Pushing

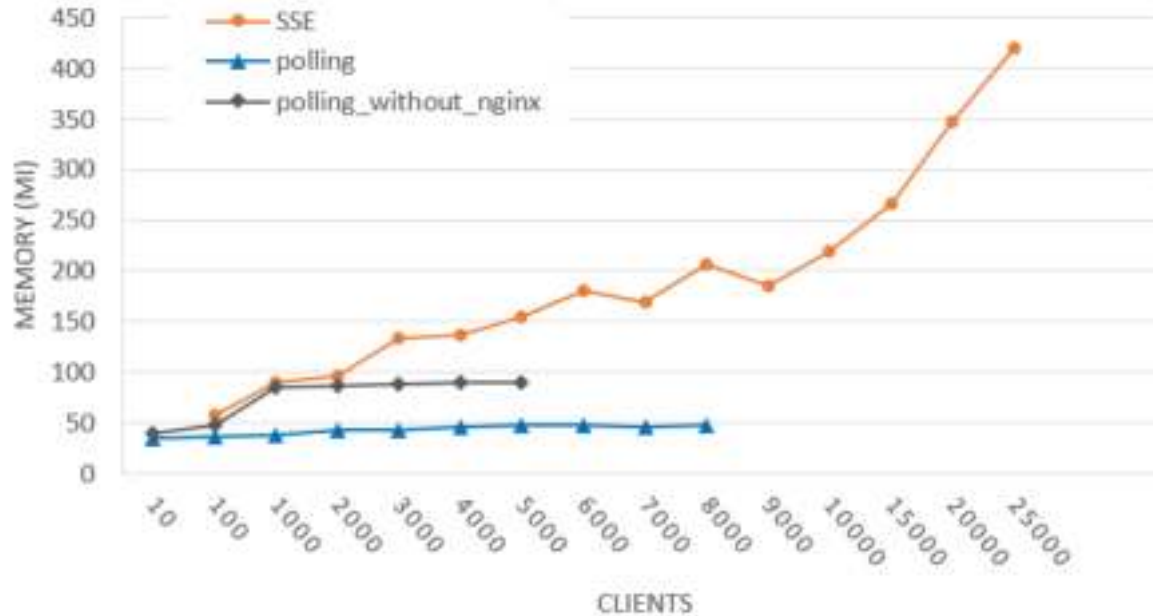


Pushing requires less CPU than polling



HTTP **polling with cache does not outperform**
and grows apart from pushing

Polling requires less memory



Pushing is favoured when the latency on the client must be as low as possible

But what if the user finds a higher latency still acceptable?

Polling supports the same number of clients (25k) when all users have a Maximal Acceptable Latency (MAL) of 10s

$$Requests_{max\ supported}/s \geq Users * request/MAL$$

2.52k requests/s
wrk HTTP benchmark

10s

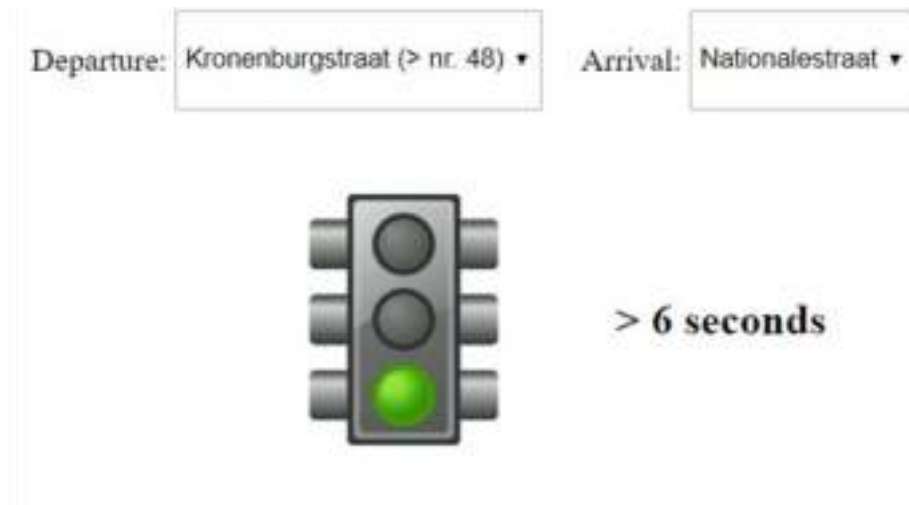
Key takeaways

When users expect **latency** < 10s,
choose pushing, e.g. **subscription** mechanism
Serve more clients, lower CPU cost, lower latency

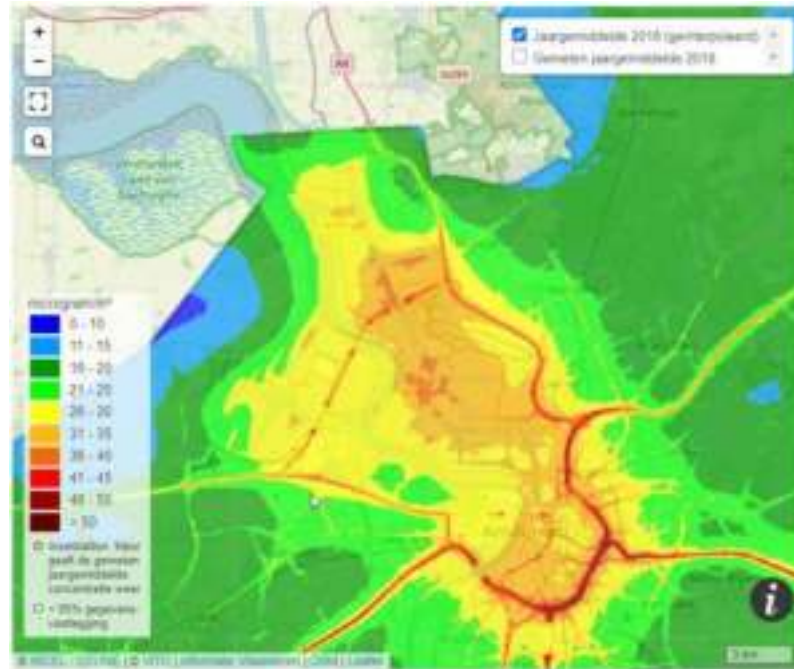
When users have a maximal acceptable **latency** > 10s,
choose polling, e.g. **REST API**
Spread of requests, more clients served

***Stream* live data to users (user latency < 10s)**


With Server-Sent Events, Websockets, MQTT...



Use polling (user latency > 10s)



Depends on the users' expectations












 **78** Zevergem - De Pinte - Gent - Lochristi - Lokeren

[Other directions](#) [Save line](#) [Receive email updates](#)

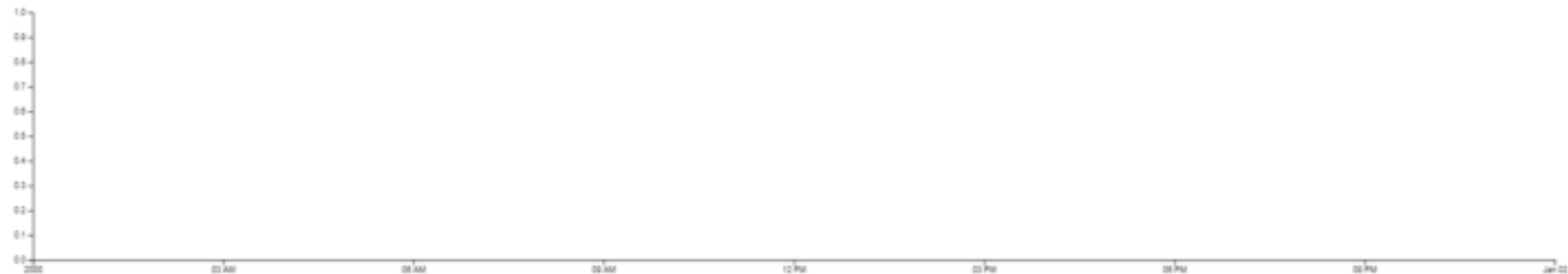
[Journeys](#) [Timetable](#) [Diversions](#)

Journey on Tuesday at **11:33**

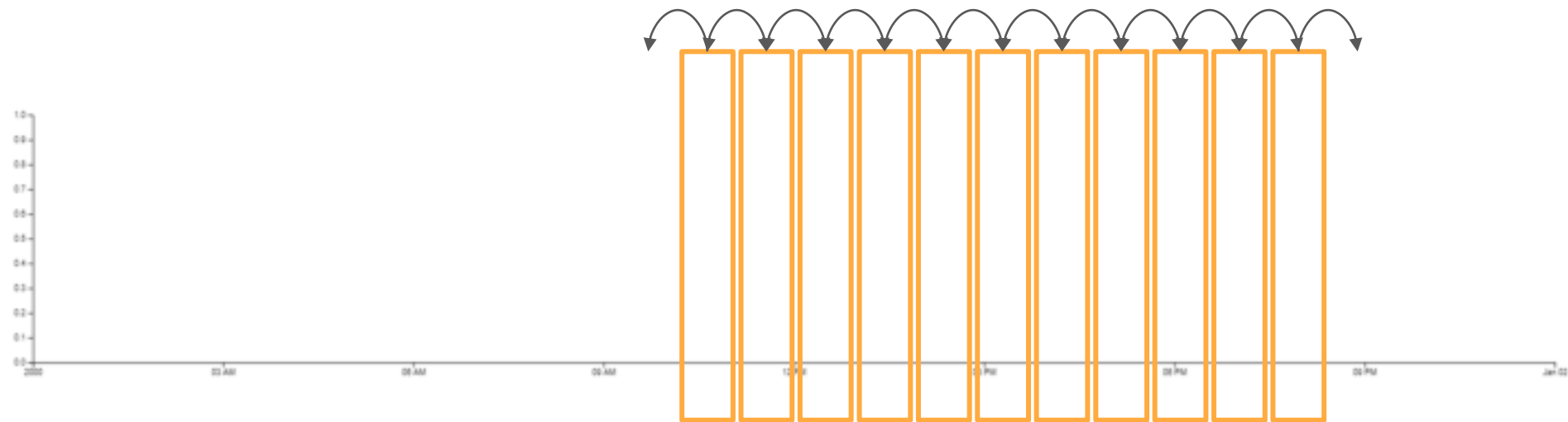
[Refresh](#)

Zwinaarde		Bollebergen	
Zwinaarde		Klosse	
Zevergem		Grote Steenweg	
Zevergem		Den Beer	 16'
De Pinte		Mieregoedstraat	18'
De Pinte		Minnestede	19'
De Pinte		Europaplein	20'

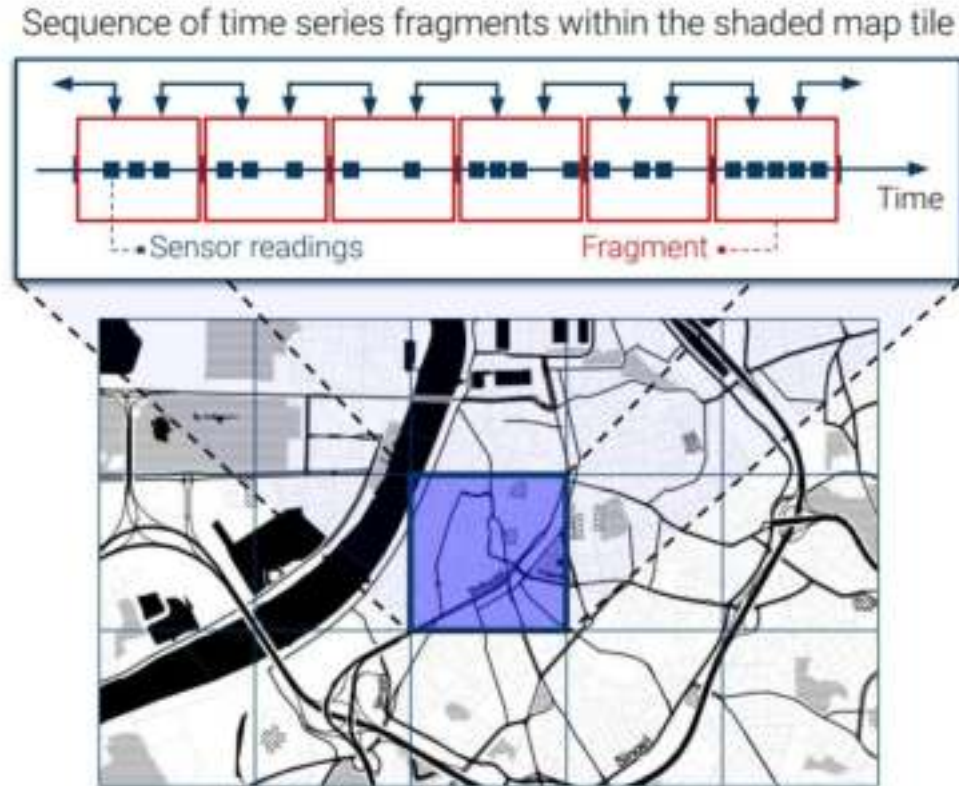
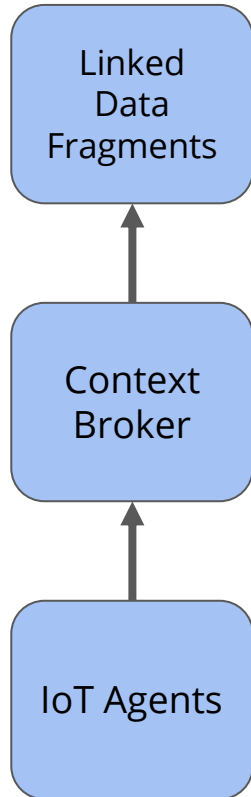
What's best for older sensor data?



Historic data in cacheable, linked fragments



Context Broker fit for historic Open Data publishing with NGSI-LDF



Interested in learning more about NGSI-LDF?

Join the “Publishing and using sensor data as linked open data” COOCK workshop

When? 23/02/2021

https://vloca-kennishub.vlaanderen.be/vloca-kennishub/Coock_Open_Stad

Interoperability on a Flemish level with Open Standards for Linked Organizations (OSLO)



Interested in learning more about OSLO?

Join the “Open City and its Citizens” COOCK workshop

When? 09/02/2021

https://vloca-kennishub.vlaanderen.be/vloca-kennishub/Coock_Open_Stad/Open_City_and_its_Citizens

Comparing a polling and push-based approach for live Open Data interfaces

*2nd February – COOCK workshop
Connecting, storing and publishing sensor data*

Brecht Van de Vyvere
IDLab (IMEC – Ghent University)

Evaluation design

200 bare-metal servers

eight 64-bit ARMv8 cores at 2.4 GHz,
64 GB of RAM, 120 TB of SATA flash storage

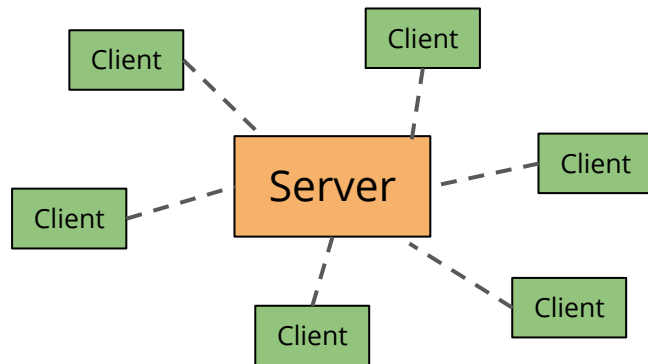
Single-threaded (Nginx and Node.js)

Kubernetes orchestration

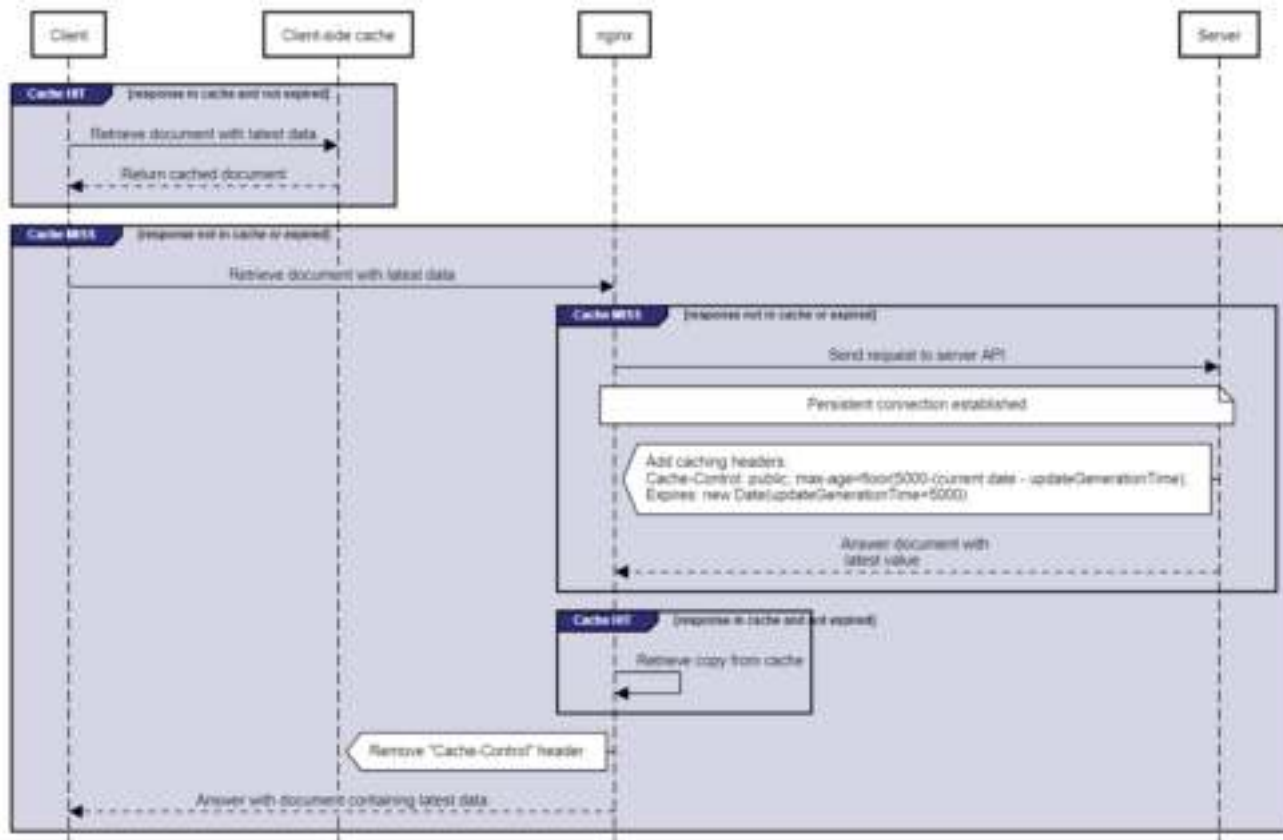
Every 5 seconds a new update (5.2 kB JSON)

Polling interval of 500ms

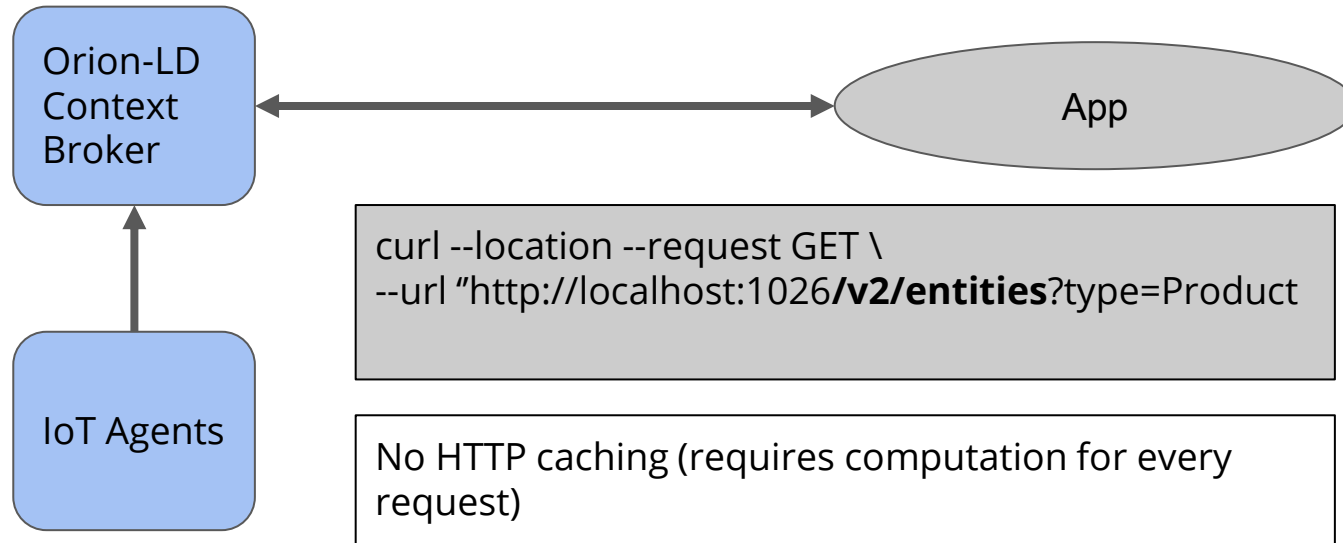
Pushing with Server-Sent Events implementation



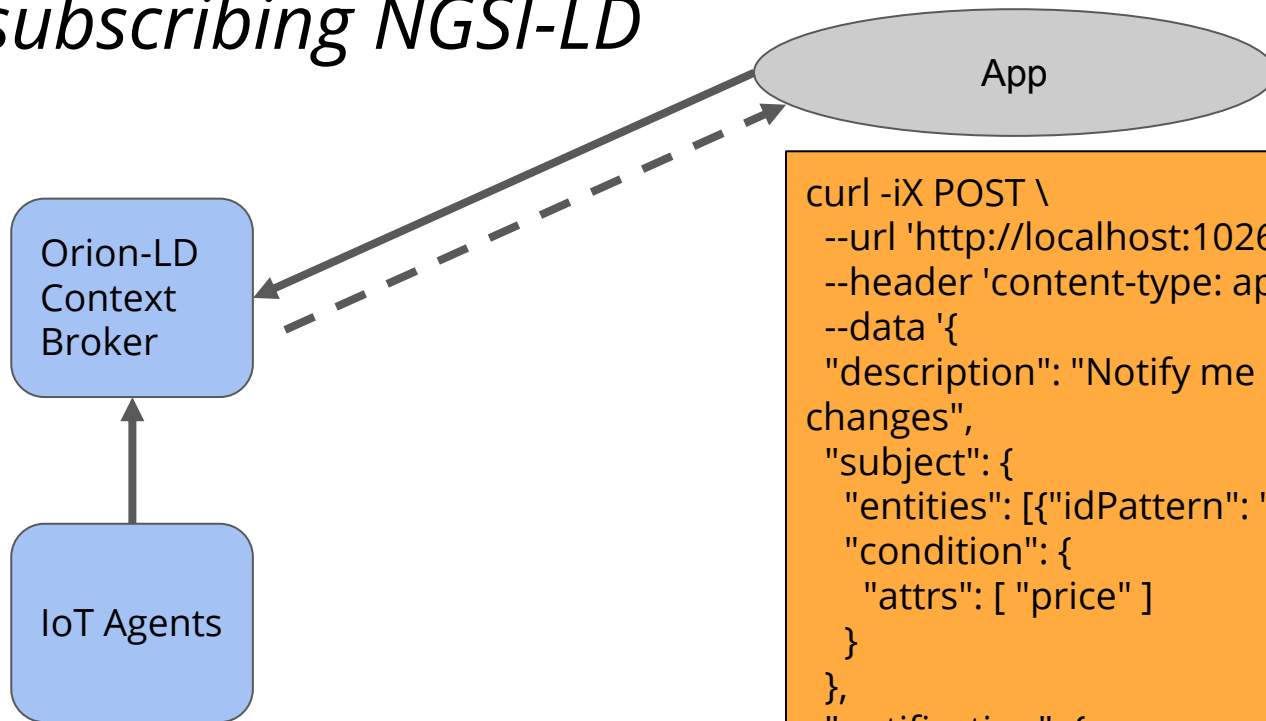
Setup polling



Retrieving real time last-value context data *by polling NGSI-LD*



Retrieving real time last-value context data *by subscribing NGSI-LD*



```
curl -iX POST \
  --url 'http://localhost:1026/v2/subscriptions' \
  --header 'content-type: application/json' \
  --data '{
    "description": "Notify me of all product price
changes",
    "subject": {
      "entities": [{"idPattern": ".*", "type": "Product"}],
      "condition": {
        "attrs": [ "price" ]
      }
    },
    "notification": {
      "http": {
        "url": "http://tutorial:3000/subscription/price-
change"
      }
    }
  }'
```

Retrieving real time last-value context data *by polling or subscribing NGSI-LDF*

