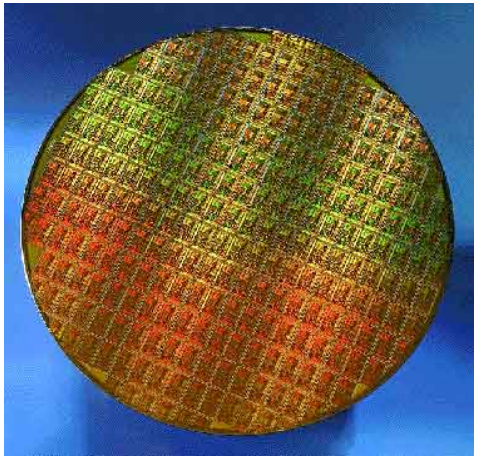
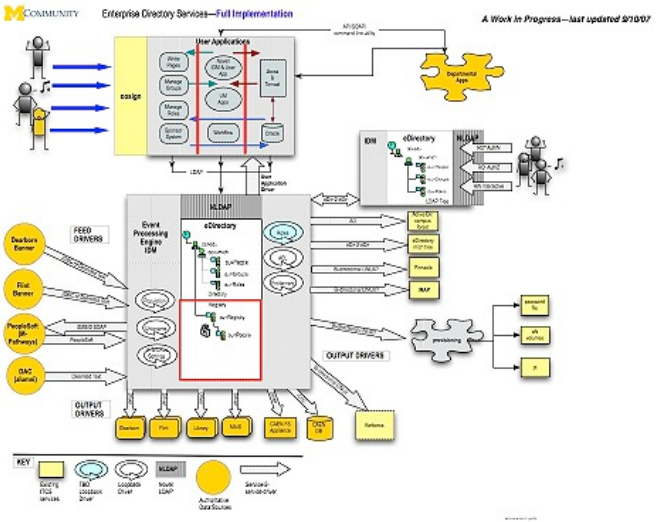
Tasty Topics!

Novel approaches using Topic Filtering

QCon
LONDON

It’s all about me





tom.fairbairn@solace.com
dev.solace.com
https://www.linkedin.com/in/tomfairbairn

solace

Pub/Sub revision

- Distributed
- Decoupled
- Fanin/Fanout
- Persistence
- Register interest in *Topic*

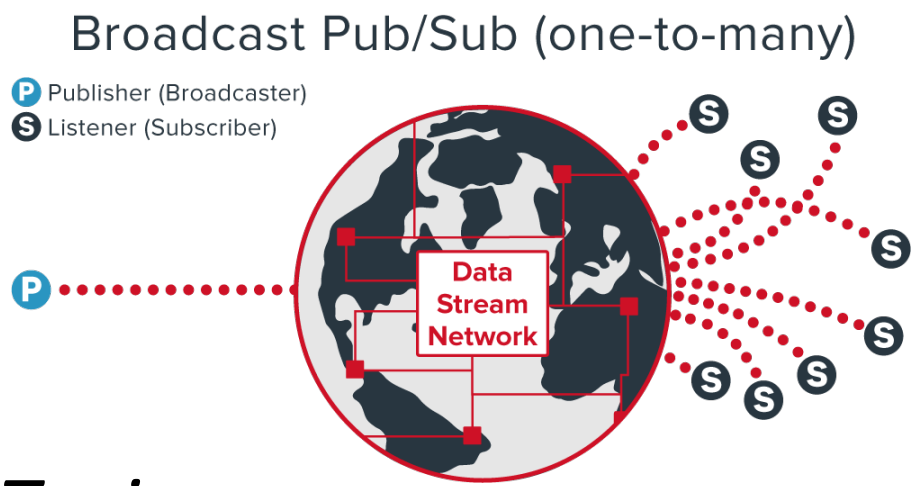
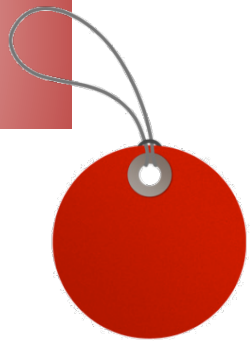


Image credit: pubnub.com

Topics

Topic ≠ Tag!



food/apple/slices

//slices

=>

food/apple/slices

food/ham/slices

food_apple_slices

List {food_apple_slices, food_ham_slices}

String search?

Who Cares?

- Simpler
- Consistent
- Reduces unnecessary data copies
  - E.g. In IoT reduces unnecessary sensor reads

Use Case 1

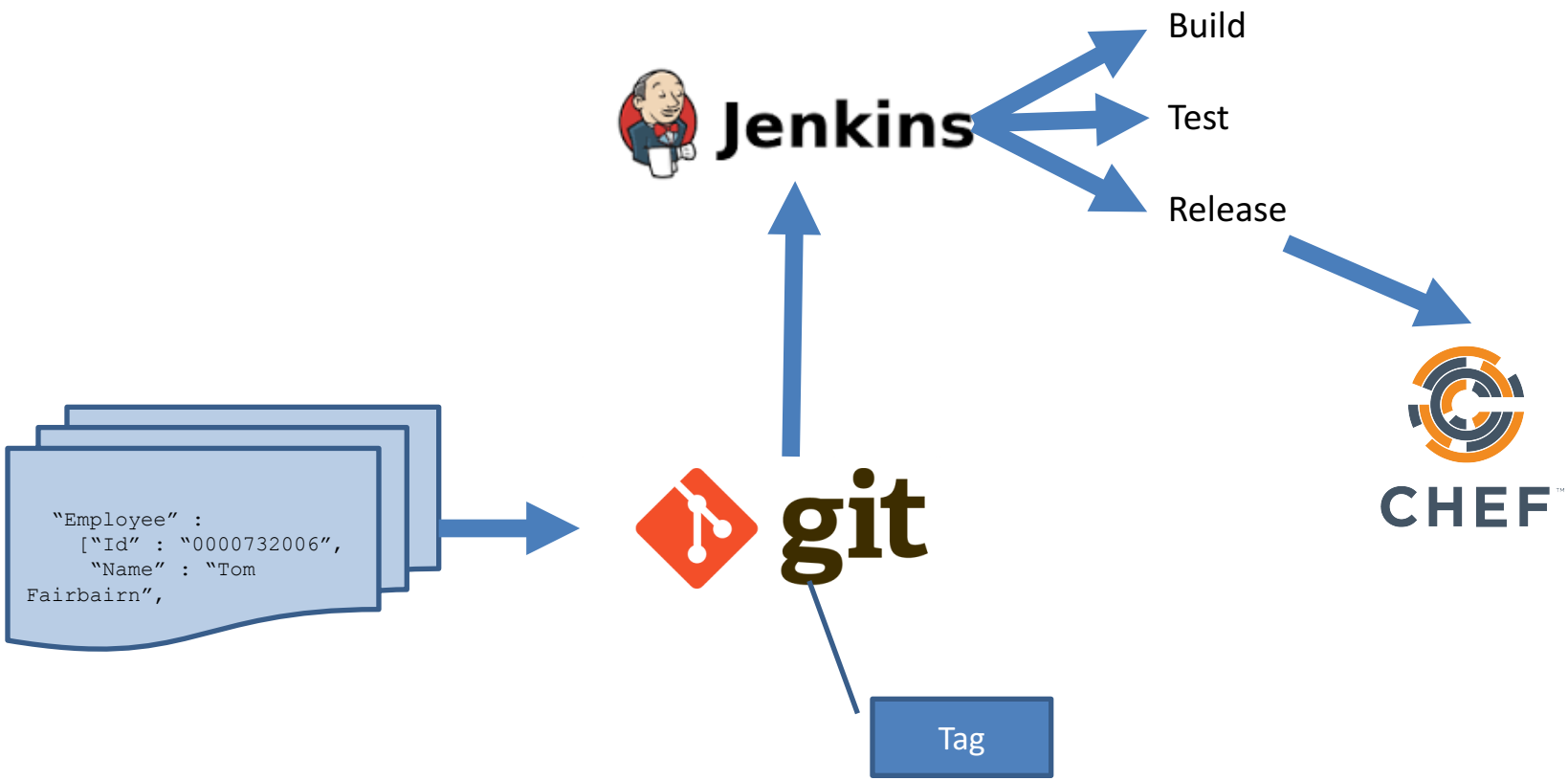
Migrating Your Data Format

Case 1: Migrating Your Data Format

```
{
  “Person” : “Tom”,
  “Team” : “Magicians”,
  “Mobile” : “07746 244422”,
  “EmployeeId” : 6
}
```

```
{
  “Employee” :
    [“Id” : “0000732006”,
     “Name” : “Tom Fairbairn”,
     “PhoneNum” : “+44(0)7746244422”,
     “DirectReports”: [],
     ”ReportsTo”: “Ben Taieb”
    ]
}
```

CI/CD


Jenkins
Build
Test
Release
CHEF
"Employee" :
["Id" : "0000732006",
"Name" : "Tom
Fairbairn",
git
Tag

Data Format – read/write

```
Gson gson = new Gson();
empolyeeData = gson.fromJson(data, employee.class);
```

```
public class employee {
  private String Person;
  private String Team;
  private String Mobile;
  public int EmployeeId;
…
}
```

Tag: v1.0

```
public class employee {
  private class employeeData {
    private String Id;
    private String Name;
    private String PhoneNum;
    private String[] DirectReports;
    private String ReportsTo;
  }
…
}
```

Tag: v2.0

Data Format topic

```
private String versionedTopic =
      “london/employee/json/$GIT_TAG_NAME/[…]”;

session.subscribe(versionedTopic);

producer.send(message, versionedTopic);
```

Use Case 2

Monitoring

Monitoring

A quick diversion

{ REST }

https://ip/endpoint

MQTT.ORG

AMQP

Websockets

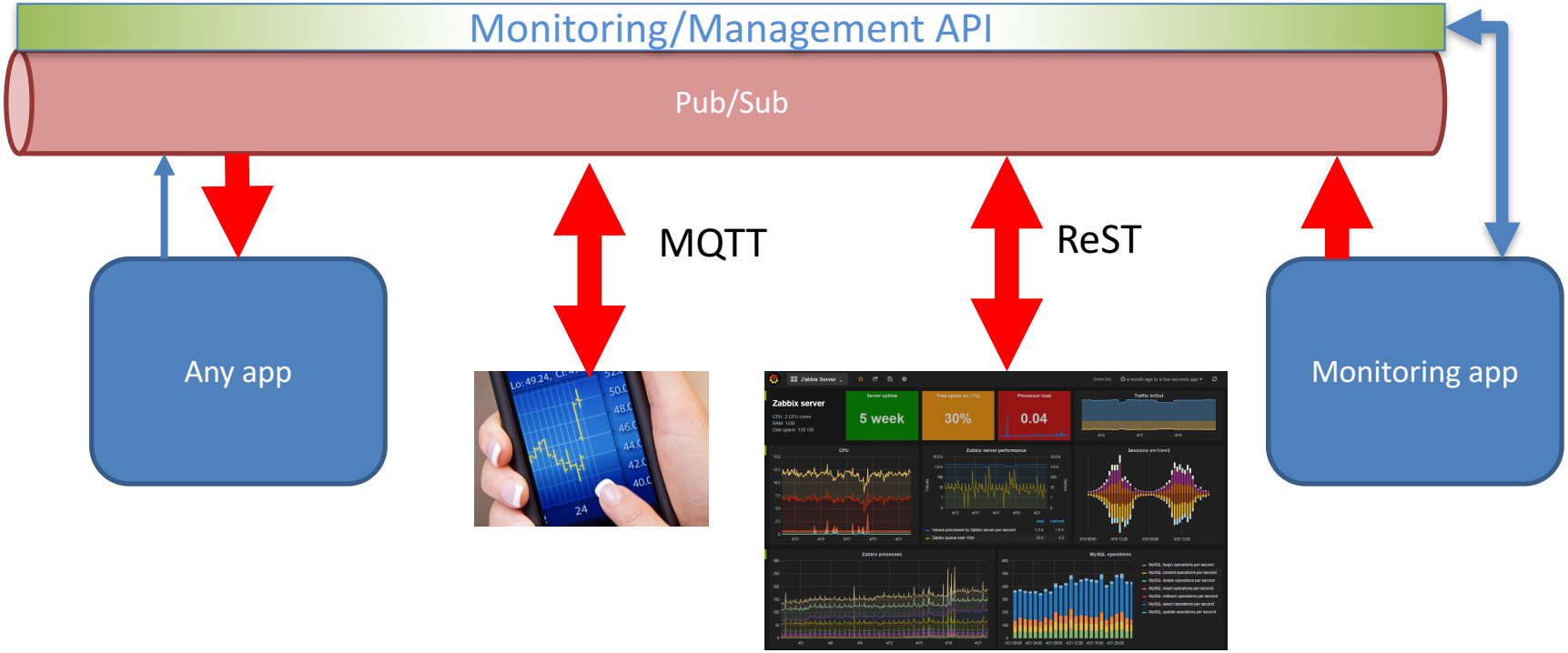
A quick diversion -again

V 1

V 5

V 3

Pub/Sub Monitoring over Pub/Sub!



Solace Stats 2

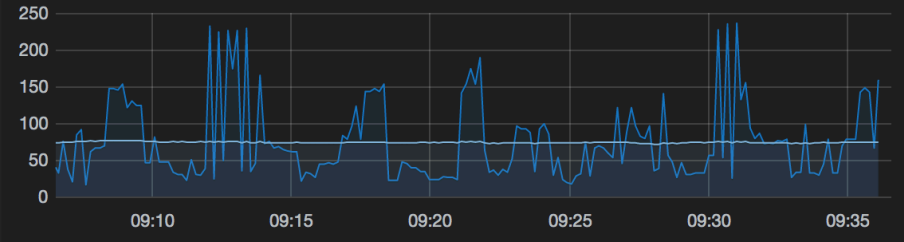
Zoom Out | Last 30 minutes | Refresh every 10s

Time Reso: auto

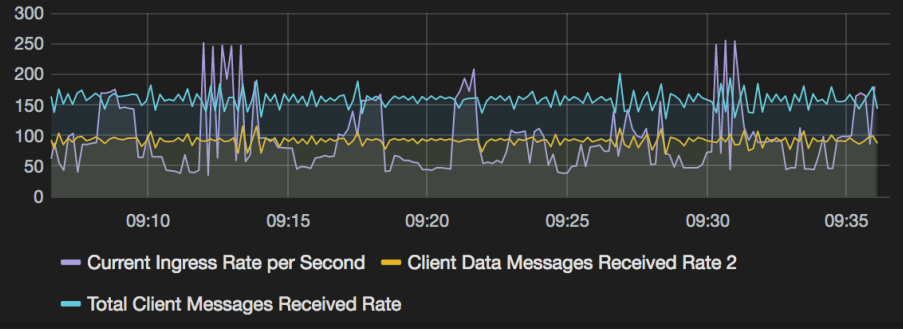
emea1 Connections	Subscriptions	emea1 Config-Sync	emea1 Redundancy	emea1 Operational Status	NAB Buffer Load Factor
361	28.2 K	Up	Up	AD-Active	0%

Current Ingress Data Rate	Average Ingress Data Rate	Current Egress Data Rate	Average Egress Data Rate
43.6 kbps	92 bps	42.7 kbps	17.9 kbps

emea1 Egress Message Rates



emea1 Ingress Message Rates



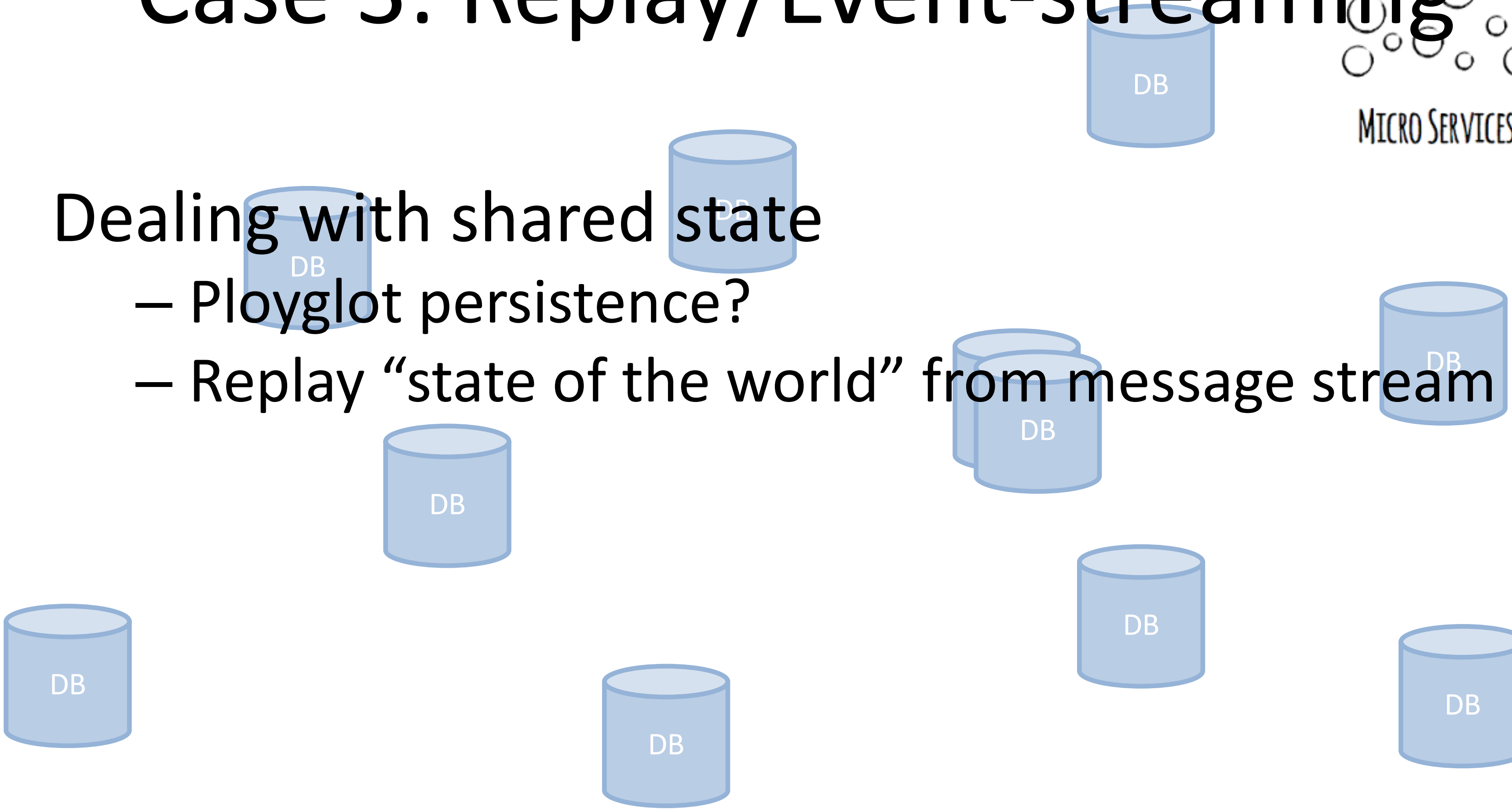
Use Case 3

Replay

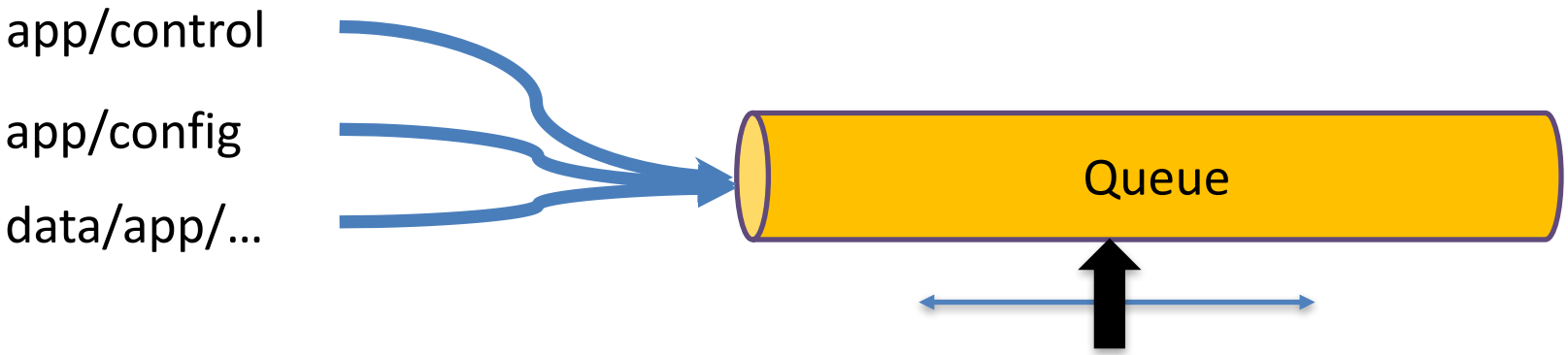
Case 3: Replay/Event-streaming

Dealing with shared state

- Ploygot persistence?
- Replay "state of the world" from message stream



Replay – queues that can subscribe

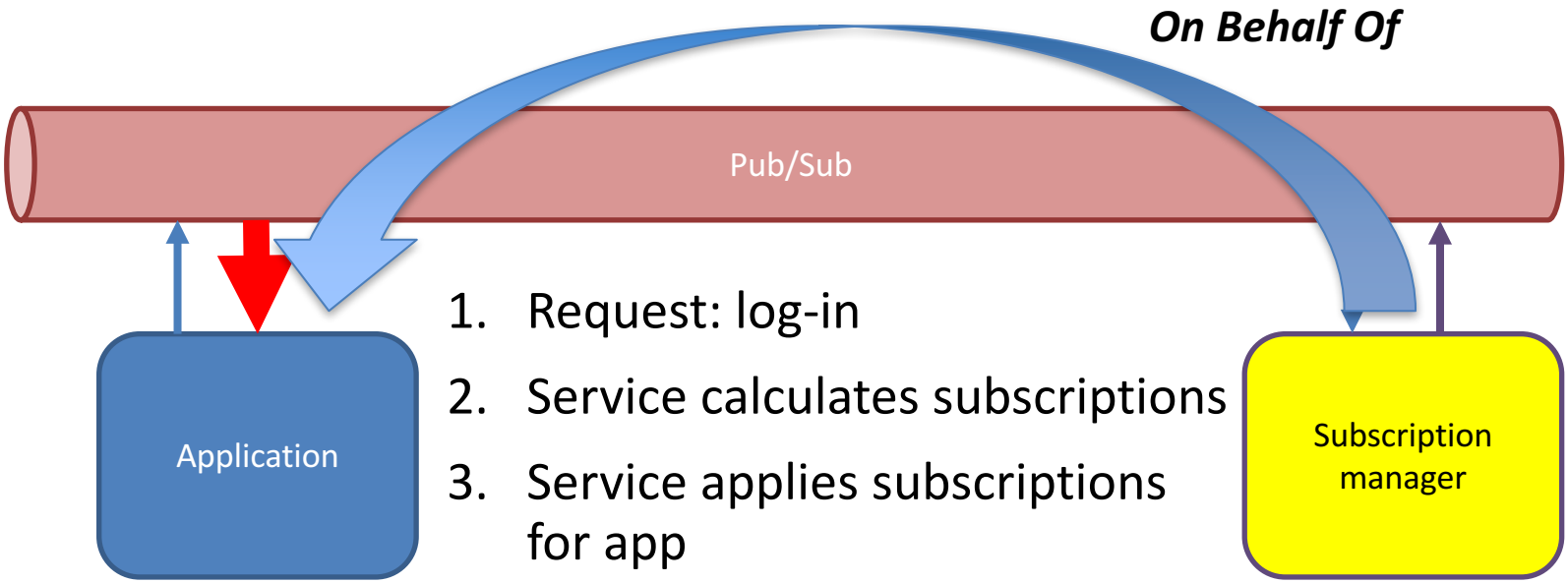


- Queue Browser
- TTL
- LVQ

Use Case 4

Authorisation

Case 4: Authorisation

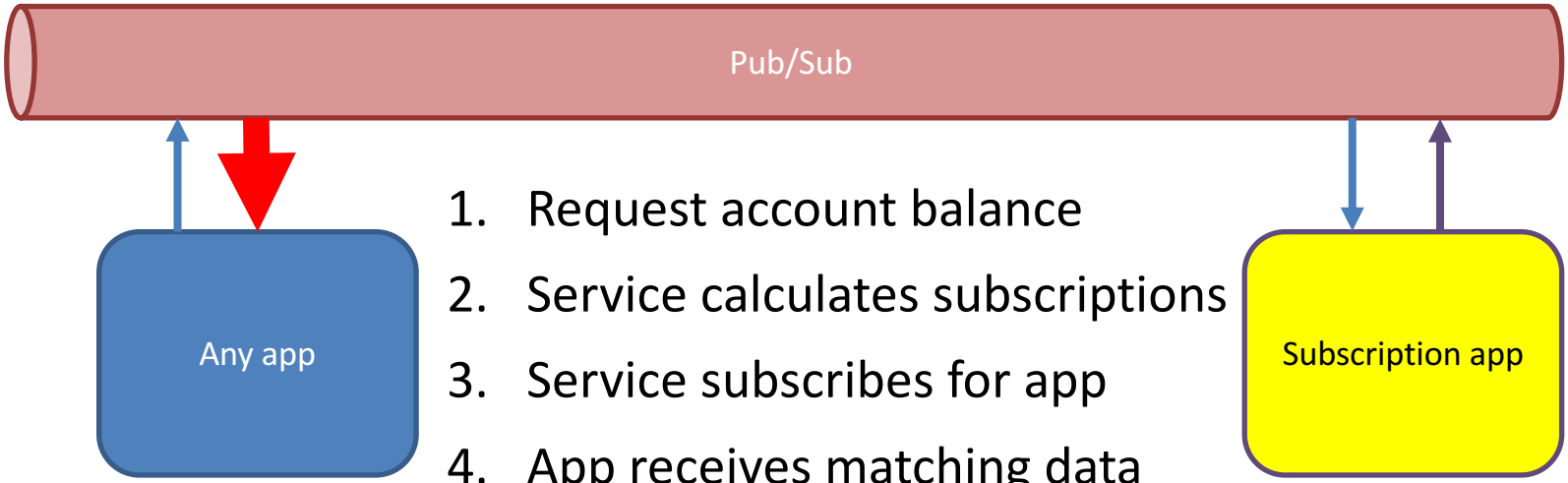


1. Request: log-in
2. Service calculates subscriptions
3. Service applies subscriptions for app
4. Reply with OK
5. App receives matching data

On Behalf Of

- Client has no awareness of topics/services
  - No chance to guess other services
  - No work/exposure at client
- Fully pluggable architecture

Authorisation



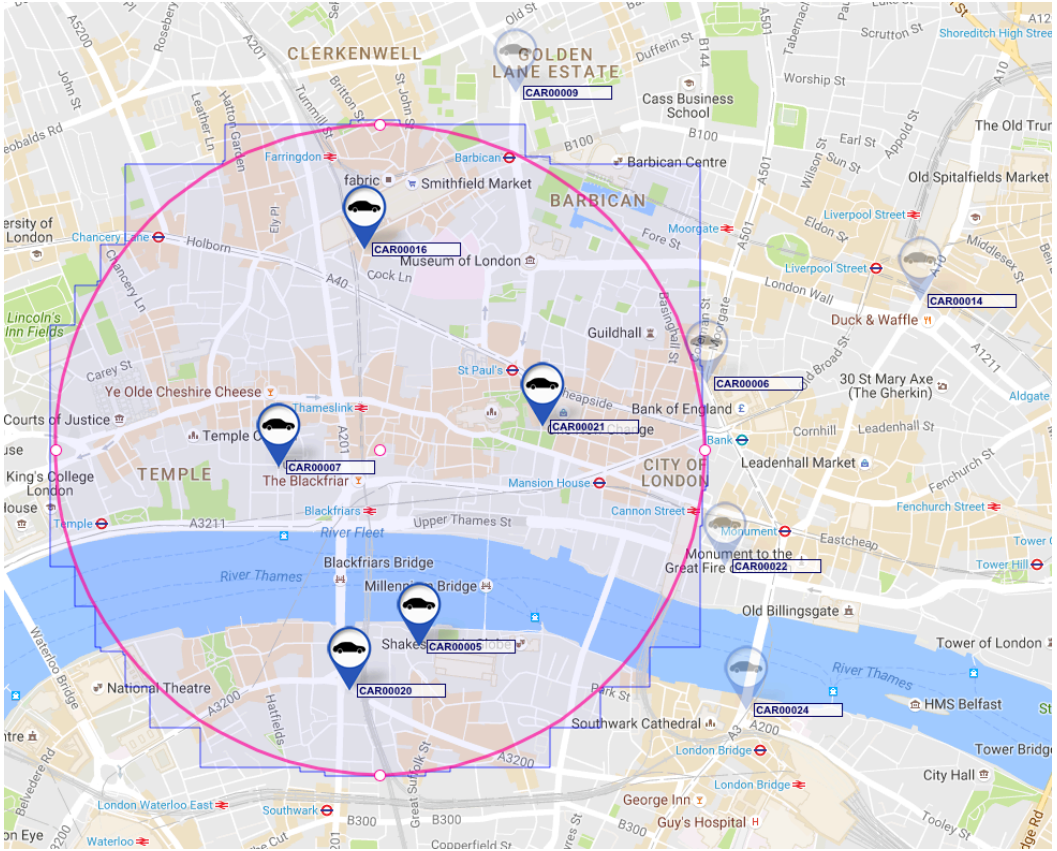
1. Request account balance
2. Service calculates subscriptions
3. Service subscribes for app
4. App receives matching data

Use Case 5

Find the nearest…

Geo-location using topics

Case 2:Find The Nearest… In Real Time



Source: ICOMP 2016, A.L. Lee, Ranged Filtering of Streaming Numeric Data… using Topic-Based Pub/Sub Messaging

Geo-filtering topic

Publish to topic with location:

```
<app>/<type>/<lat>/<long>/<vehicle>/<id>
geo/sim/51.520150/-00.097330/CAR/00021
```

Where is CAR00021?

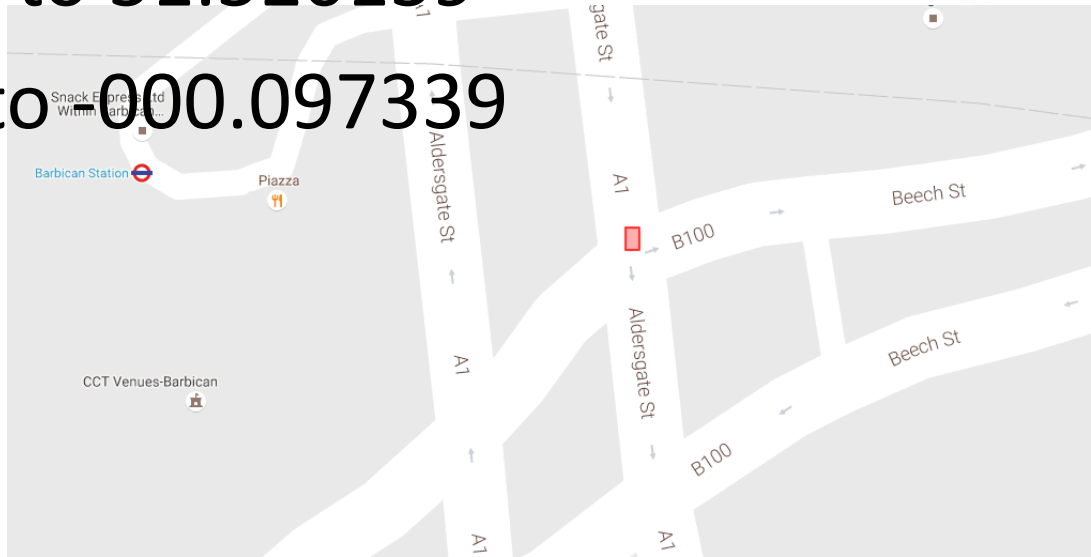
```
subscribe(“geo/sim/*/*/CAR/00021”);
```

Geo-filtering location

```
subscribe("geo/sim/51.52015*/-00.09733*/>");
```

Match: lat 51.520150 to 51.520159

long -000.097330 to 000.097339

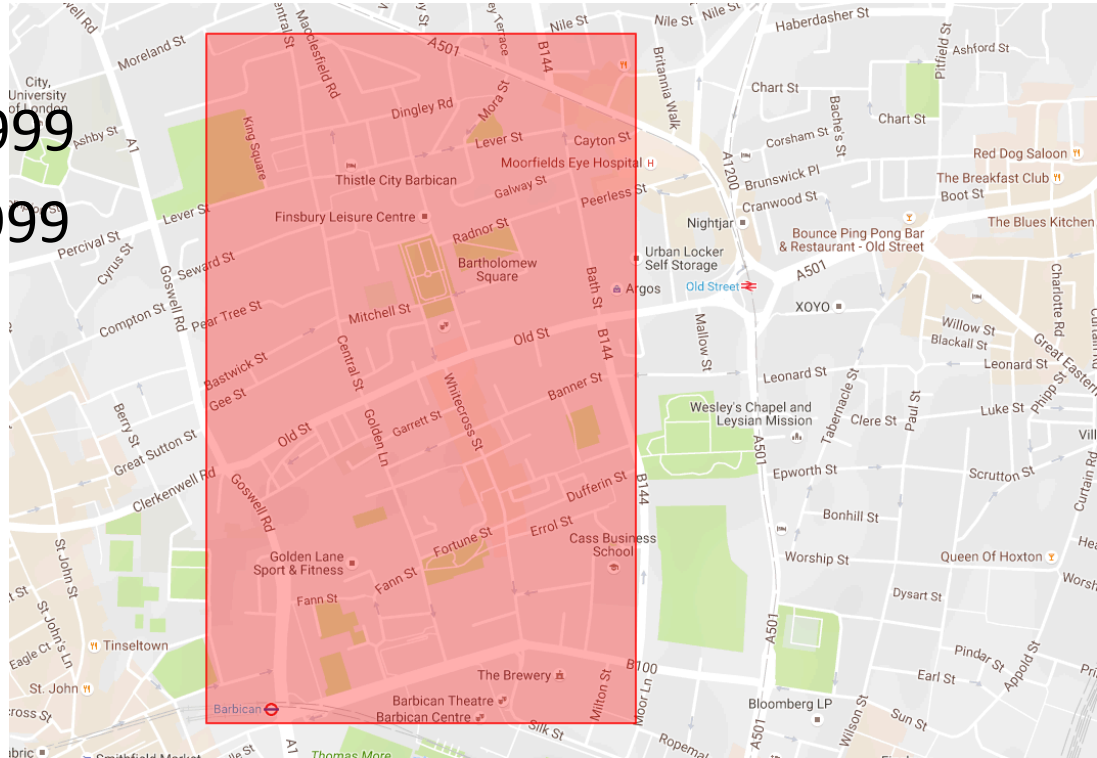


Geo-filtering location

```
subscribe("geo/sim/51.52*/-00.09*/>");
```

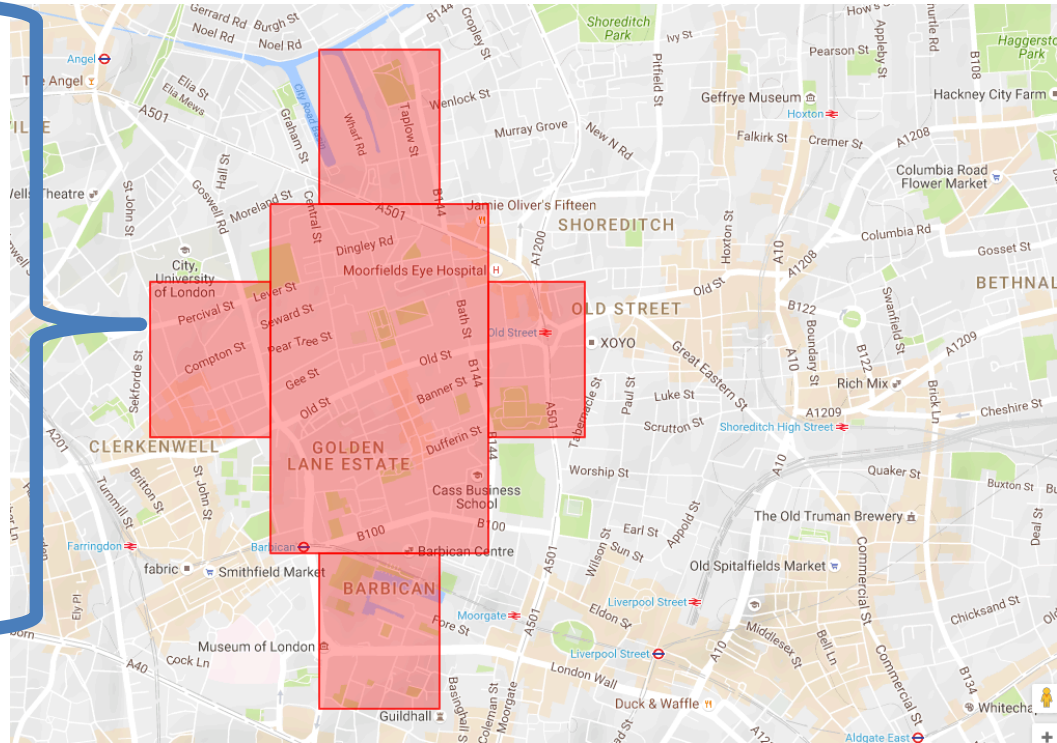
Match: lat 51.520 to 51.529999

long -0.090 to -0.099999



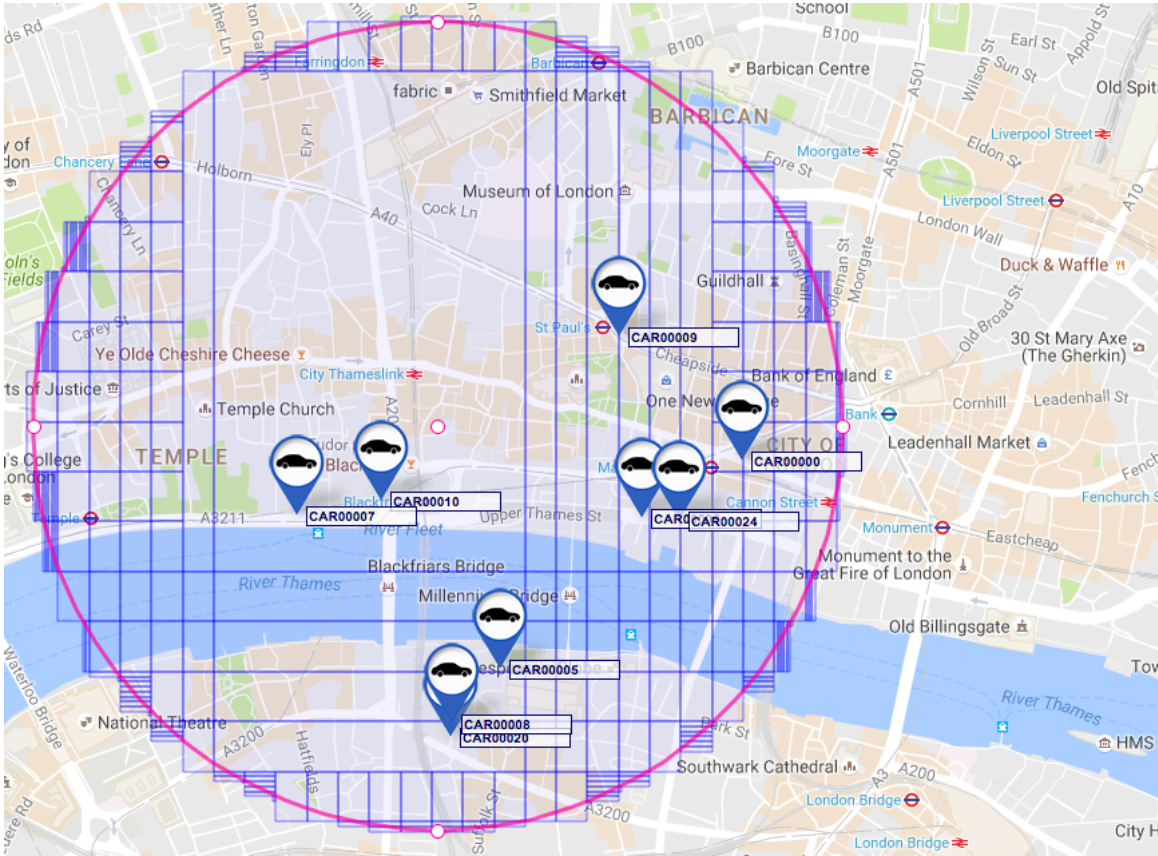
Geo-filtering location

```
subscribe("geo/sim/51.52*/-00.09*/>",

"geo/sim/51.516*/-00.092*/>",
"geo/sim/51.516*/-00.093*/>",
"geo/sim/51.516*/-00.094*/>",
"geo/sim/51.516*/-00.096*/>",

"geo/sim/51.517*/-00.092*/>",
"geo/sim/51.517*/-00.093*/>",
"geo/sim/51.517*/-00.094*/>",
"geo/sim/51.517*/-00.096*/>",

// repeat for 51.518 and .519
);
```



Geo-filtering location

- Create any polygon
  - Accuracy at metre level
  - Circles, arcs…
- Subscriptions generated *once*
- Matches then stream in with no extra computation

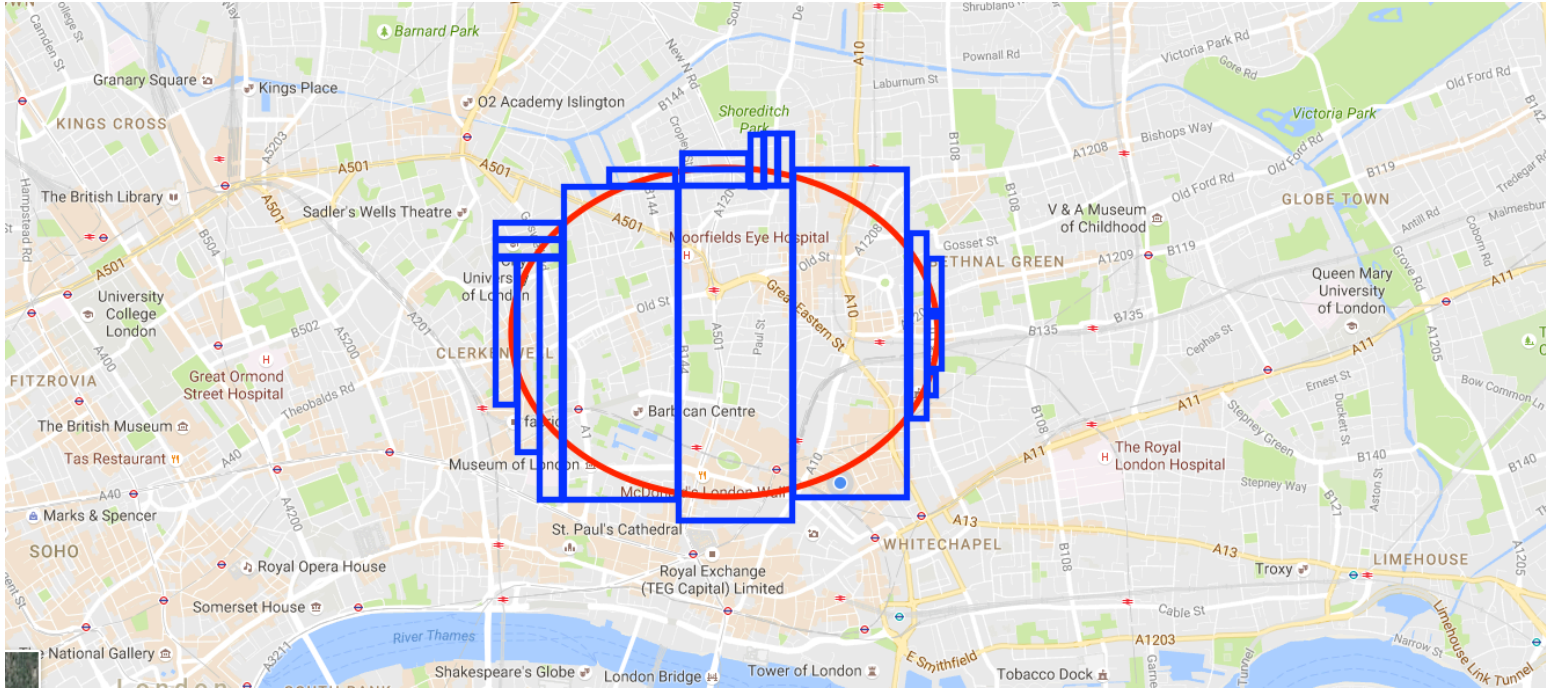


Geo-filtering location algorithm

- Divide space into rectangles aligned to subscriptions
- Throw away rectangles with no match

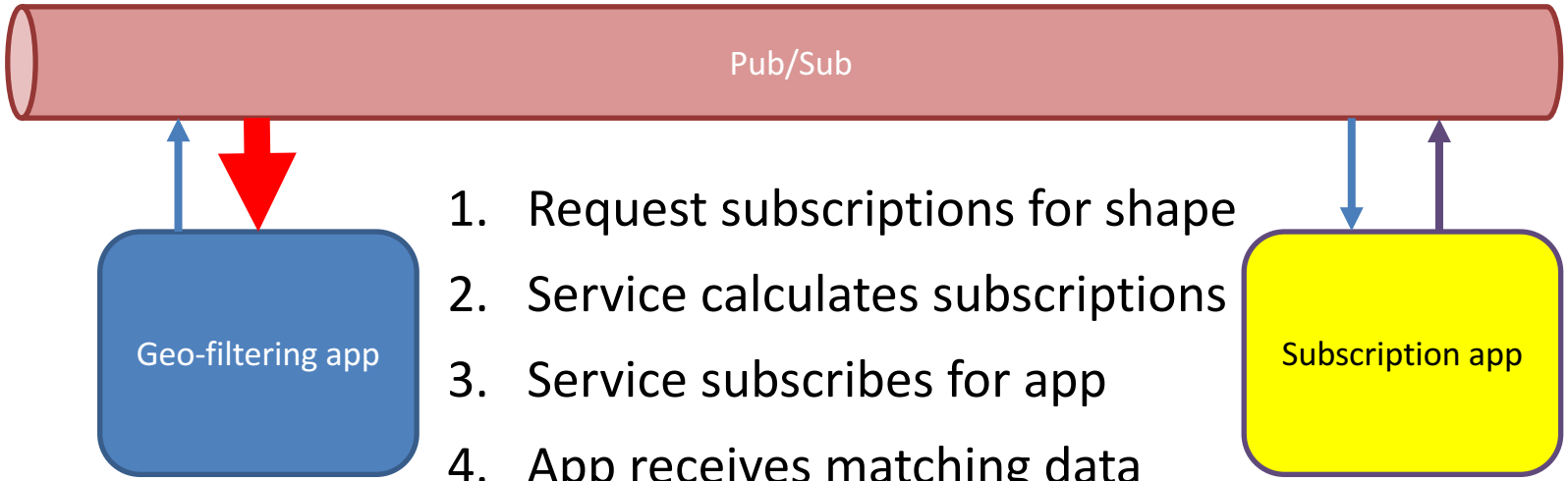
Geo-filtering location algorithm

- Repeat: divide remaining rectangles by 10
- Throw away rectangles with no match



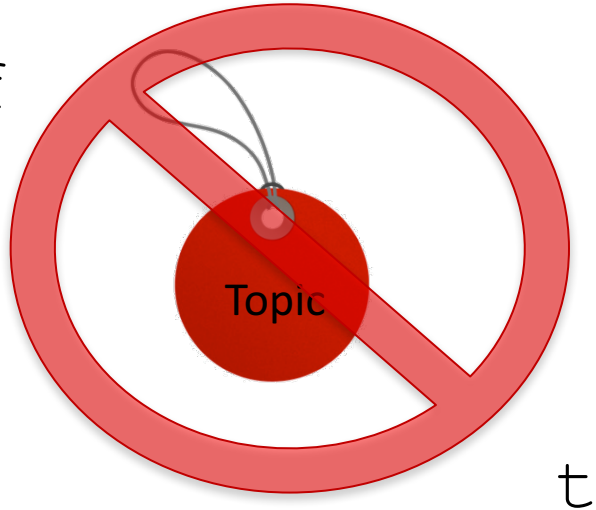
Geo-filtering location algorithm deployment

- Library?

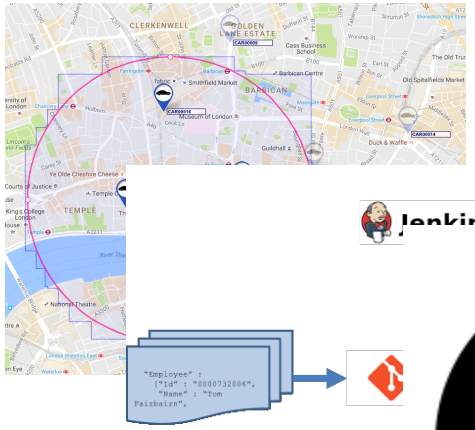


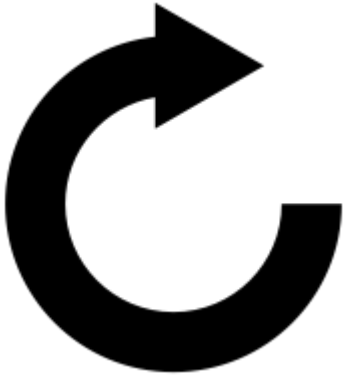
Last code snippet

if



then





Come and see us at our Booth!