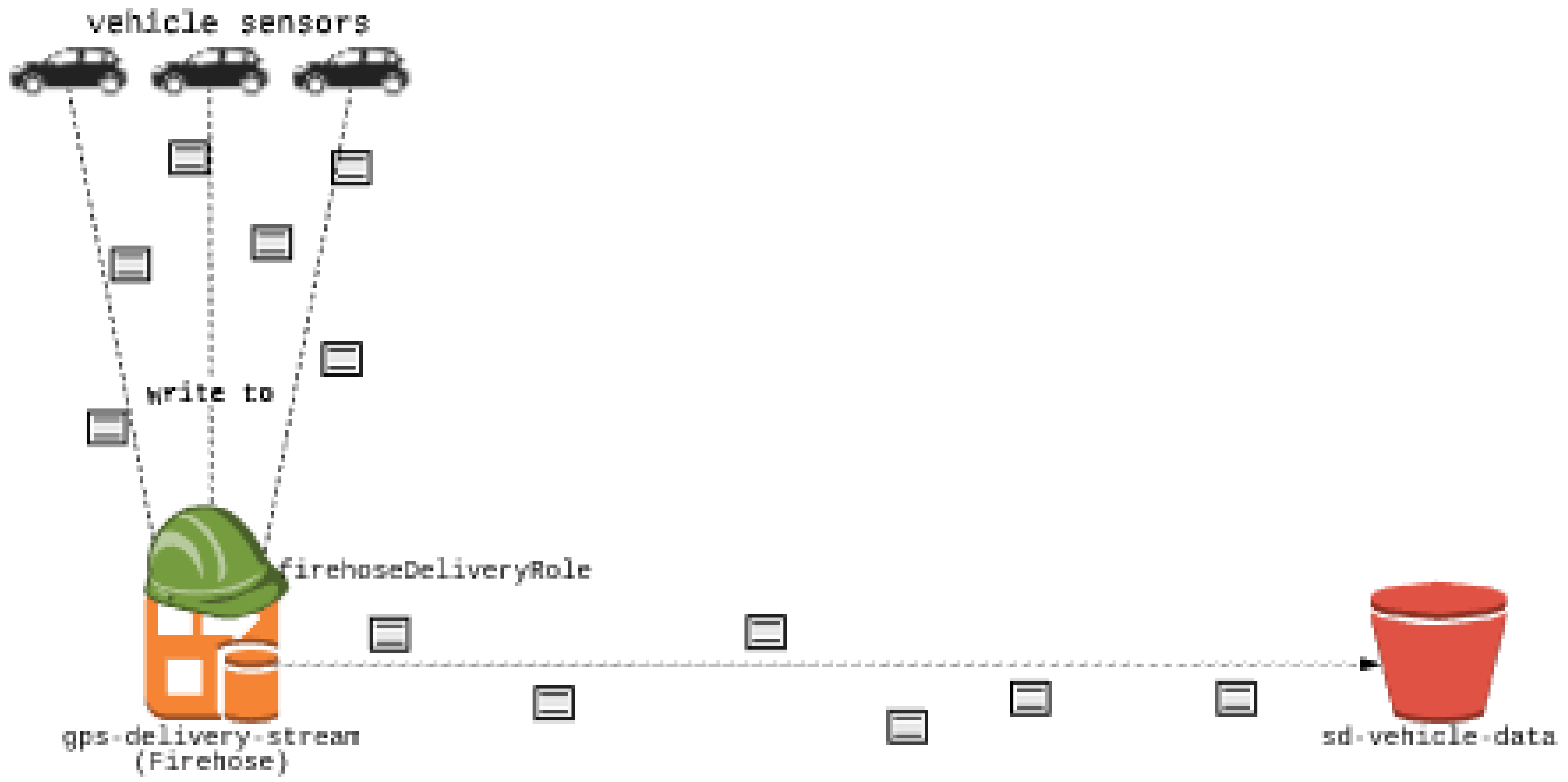
A transformational Lambda

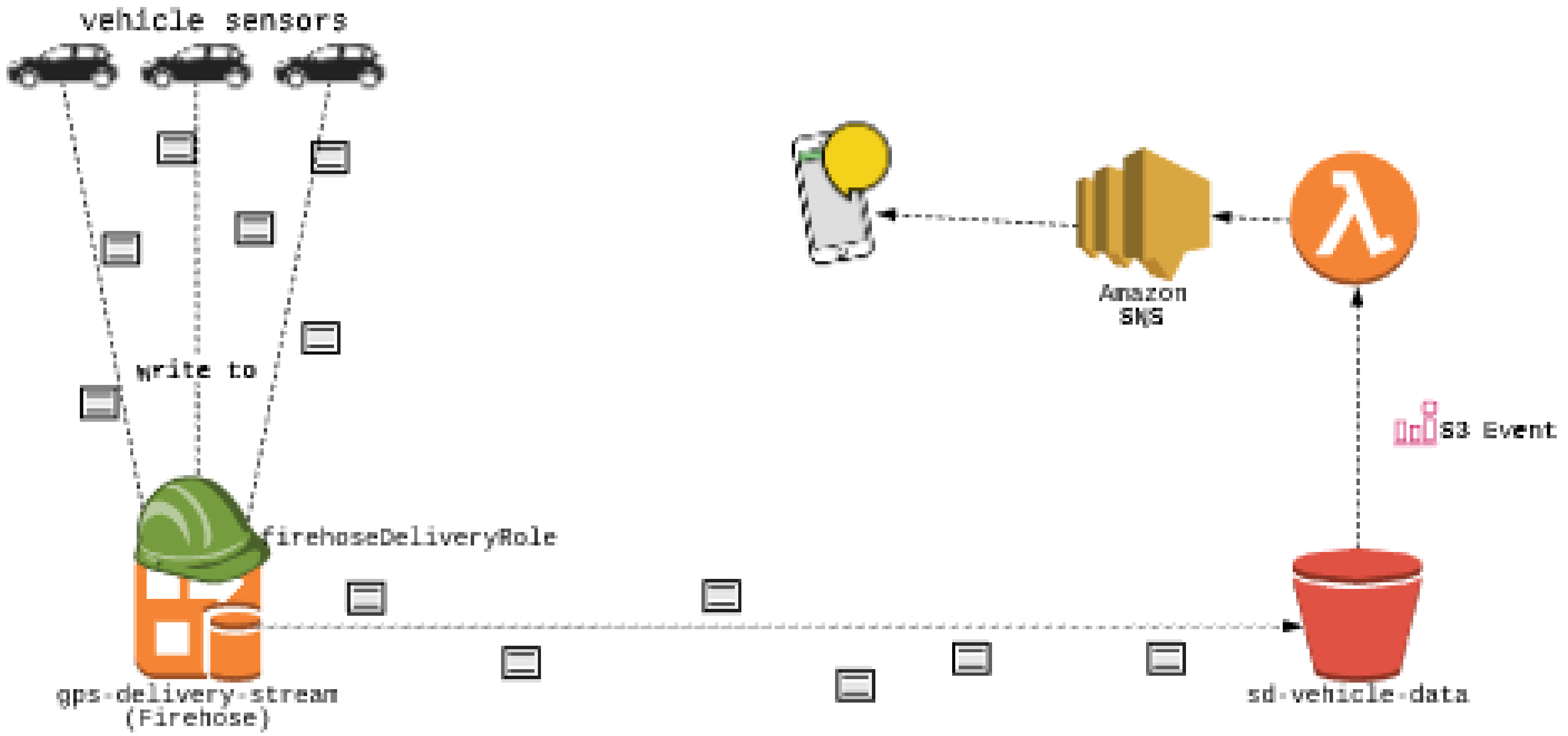
STREAMING DATA WITH AWS KINESIS AND LAMBDA

Maksim Pecherskiy
Data Engineer

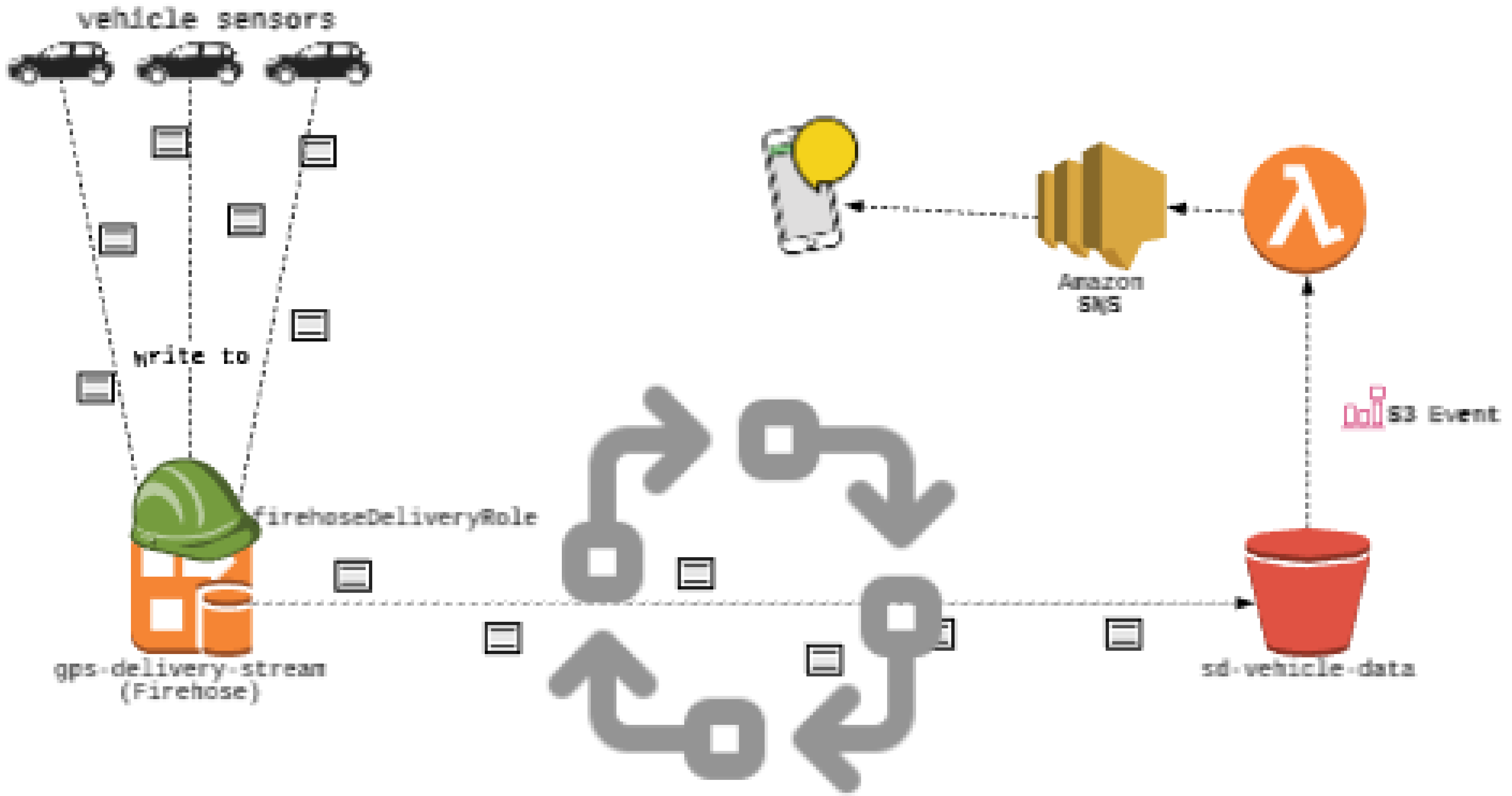
In chapter 1...


vehicle sensors
write to
firehoseDeliveryRole
gps-delivery-stream
(Firehose)
sd-vehicle-data

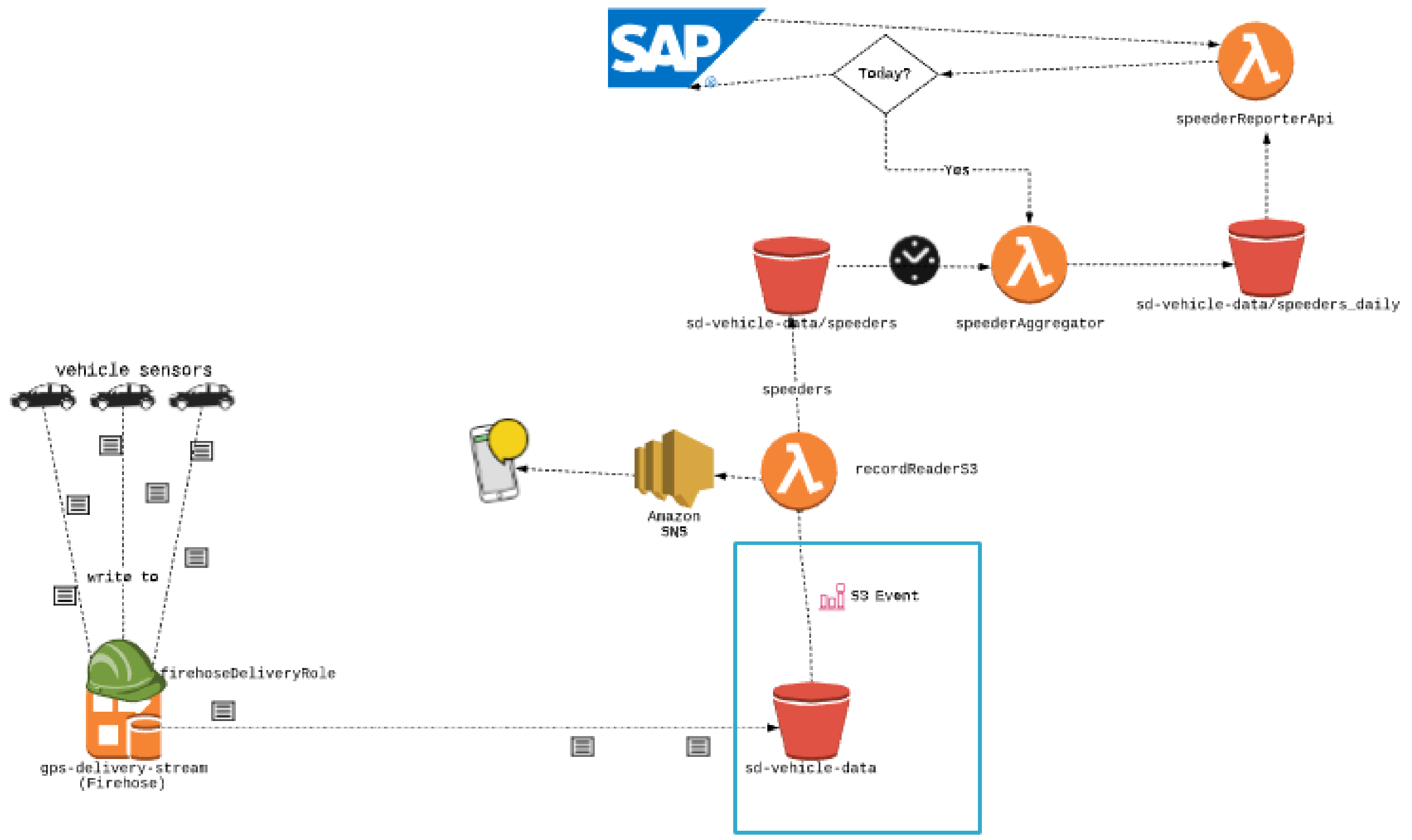
In chapter 2...


vehicle sensors
write to
firehoseDeliveryRole
gps-delivery-stream (Firehose)
sd-vehicle-data
S3 Event
Amazon SNS

In this chapter...



Previous approach



Processing once in S3 vs Lambda transform

Processing once in S3

- Uses a lambda function fired on object write in S3
- Longer delay until data can be transformed
- Raw data is stored before cleaning
- Requires Firehose destination to be S3

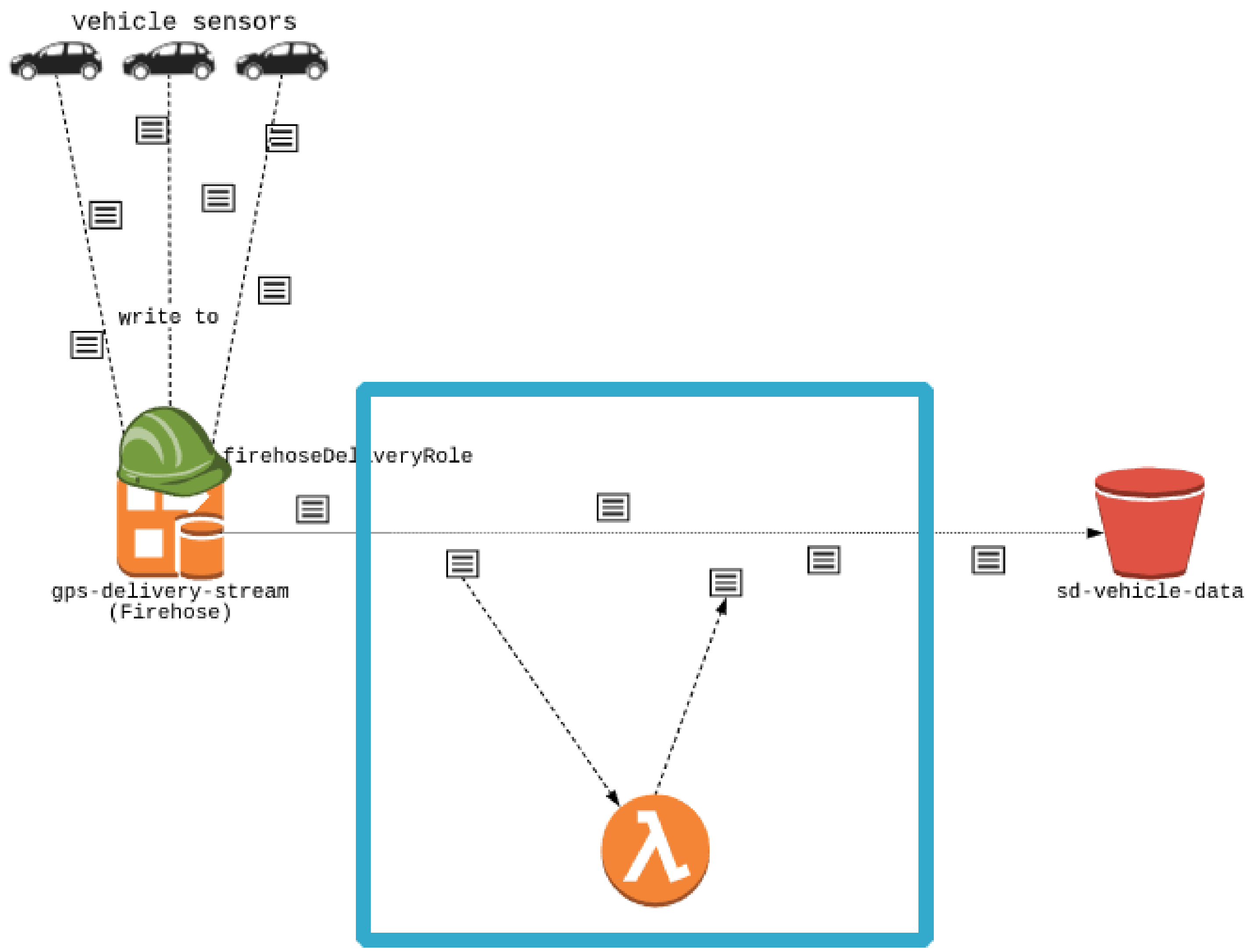
Processing via Lambda transform

- Uses a lambda function fired mid-firehose-stream
- Transformation is immediate
- Only cleaned data is stored
- Allows for other destination in Firehose

Incoming data

record_id	timestamp	vin	lon	lat	speed
939ed1d1-1740-420c-8906-445278573c7f	4:25:06.000	4FTEX4944AK844294	106.9447146	-6.3385652	25
f29a5b3d-d0fa-43c0-9e1a-e2a5cdb8be7a	8:10:47.000	3FTEX1G5XAK844393	108.580681	34.79925	37
ff8e7131-408d-463b-8d07-d016419b0656	20:26:44.000	2LAXX1C8XAK844292	114.392392	36.097577	90

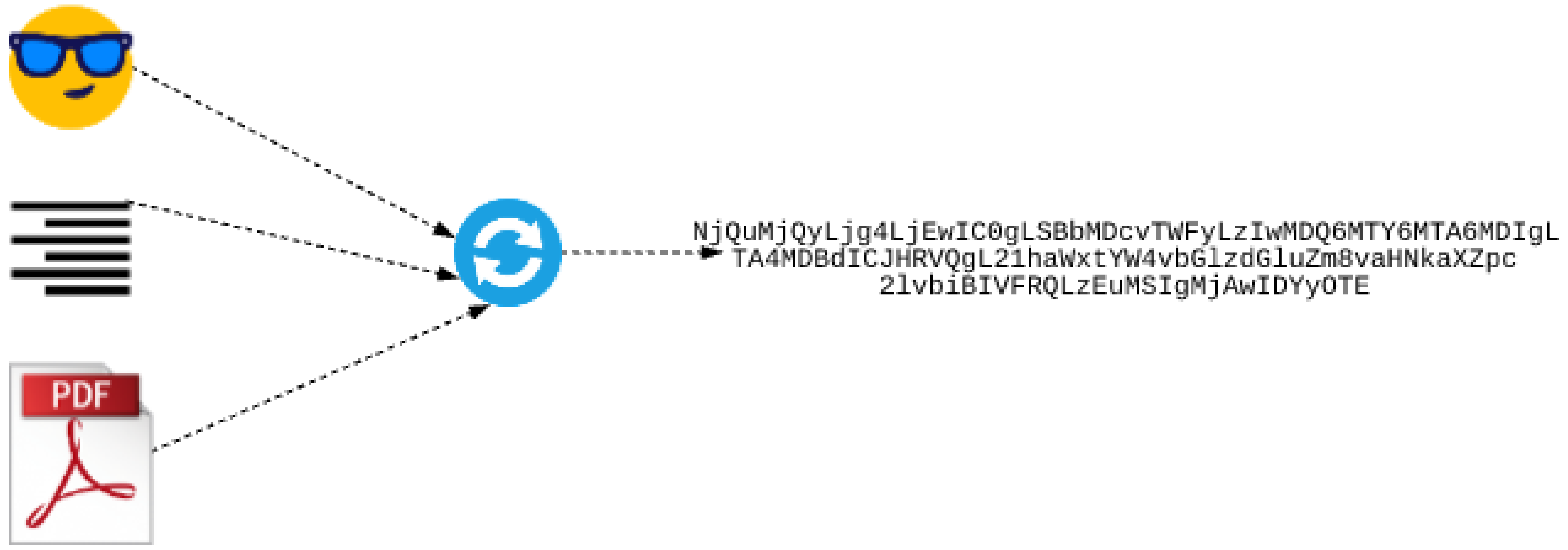
Transformational Lambda



Sample event

```
{
  "invocationId": "invocationIdExample",
  "deliveryStreamArn": "arn:aws:firehose:us-east-1:458912630:deliverystream/gps-delivery-stream",
  "region": "us-east-1",
  "records": [
    {
      "recordId": "49546986683135544286507457936321625675700192471156785154",
      "approximateArrivalTimestamp": 1495072949453,
      "data": "NjQuMjQyLjg4LjEwIC0gLSBbMDcvTWFyLzIwMDQ6MTY6MTA6MDIgLTA4MDBdICJHRVQgL21haWxtYW4vbGlz
    }
  ]
}
```

Base64



NjQuMjQyLjg4LjEwIC0gLSBbMDcvTWFyLzIwMDQ6MTY6MTA6MDIgL
TA4MDBdICJHRVQgL21haWxtYW4vbGlzdGluZm8vaHNkaXZpc
2lvbiBIVFRQLzEuMSIgMjAwIDYyOTE

Processing the data

```
import base64
from datetime import datetime as dt
def convert_timestamp(record_time_val):
    # Get today's date as string
    today = dt.today().strftime("%Y-%m-%d")
    # Combine today's date with the record's time and make datetime object
    new_ts = dt.strptime(f"{today} {record_time_val}", "%Y-%m-%d %H:%M:%S.%f")
    # Convert the datetime object to a nicely formatted string
    return new_ts.strftime("%Y-%m-%dT%H:%M:%S")
```

Processing the data

```
def lambda_handler(event, context):
    output = []
    for record in event['records']:
        payload = base64.b64decode(record['data'])
        payload = payload.decode()
        payload = payload.split(" ")
        payload[1] = convert_timestamp(payload[1])
```

Processing the data

```
for record in event['records']:
    ...
    payload = " ".join(payload)
    payload_enc = base64.b64encode(payload.encode())
    output.append({
      'recordId': record['recordId'],
      'result': 'Ok',
      'data': payload_enc
    })
return {'records': output}
```

A review

```python
import base64
def lambda_handler(event, context):
    output = []
    for record in event['records']: # Iterate over the records
        payload = base64.b64decode(record['data']).decode() # Decode the payload
        # Modify it
        payload_enc = base64.b64encode(payload.encode()) # Re-encode it
        output.append({ # Put it in a dictionary
            'recordId': record['recordId'], 'result': 'Ok', 'data': payload_enc,
        })
    return {'records': output}
```

Creating Lambda in AWS

Script for create lambda

https://drive.google.com/file/d/1fTzlp2mgWt04nVdILc7ClyAMOF1yNxkv/view?usp=sharing

Let's practice!

STREAMING DATA WITH AWS KINESIS AND LAMBDA

Transforming data inside a stream

STREAMING DATA WITH AWS KINESIS AND LAMBDA

Maksim Pecherskiy
Data Engineer

Let's practice!

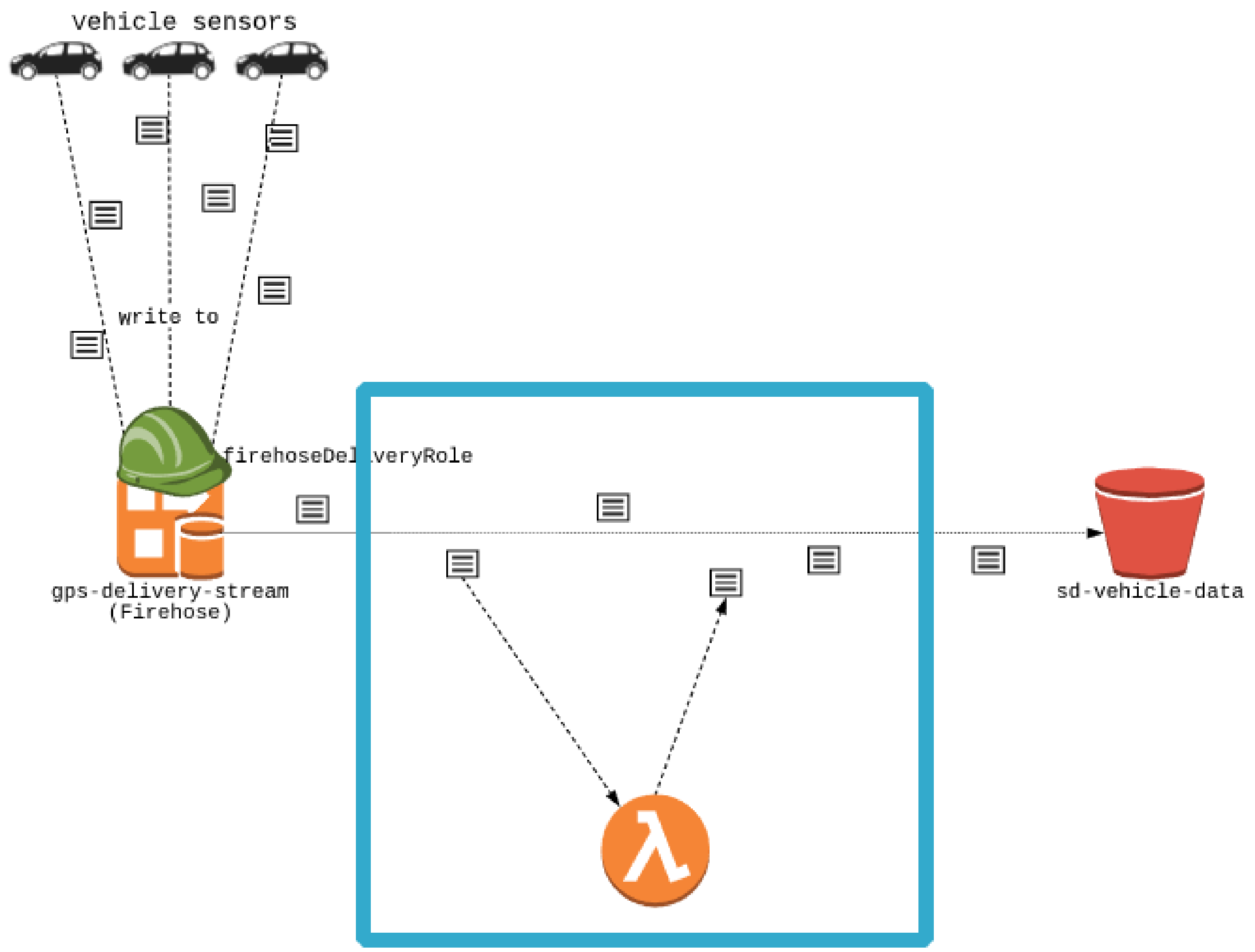
STREAMING DATA WITH AWS KINESIS AND LAMBDA

Analyzing data in the stream

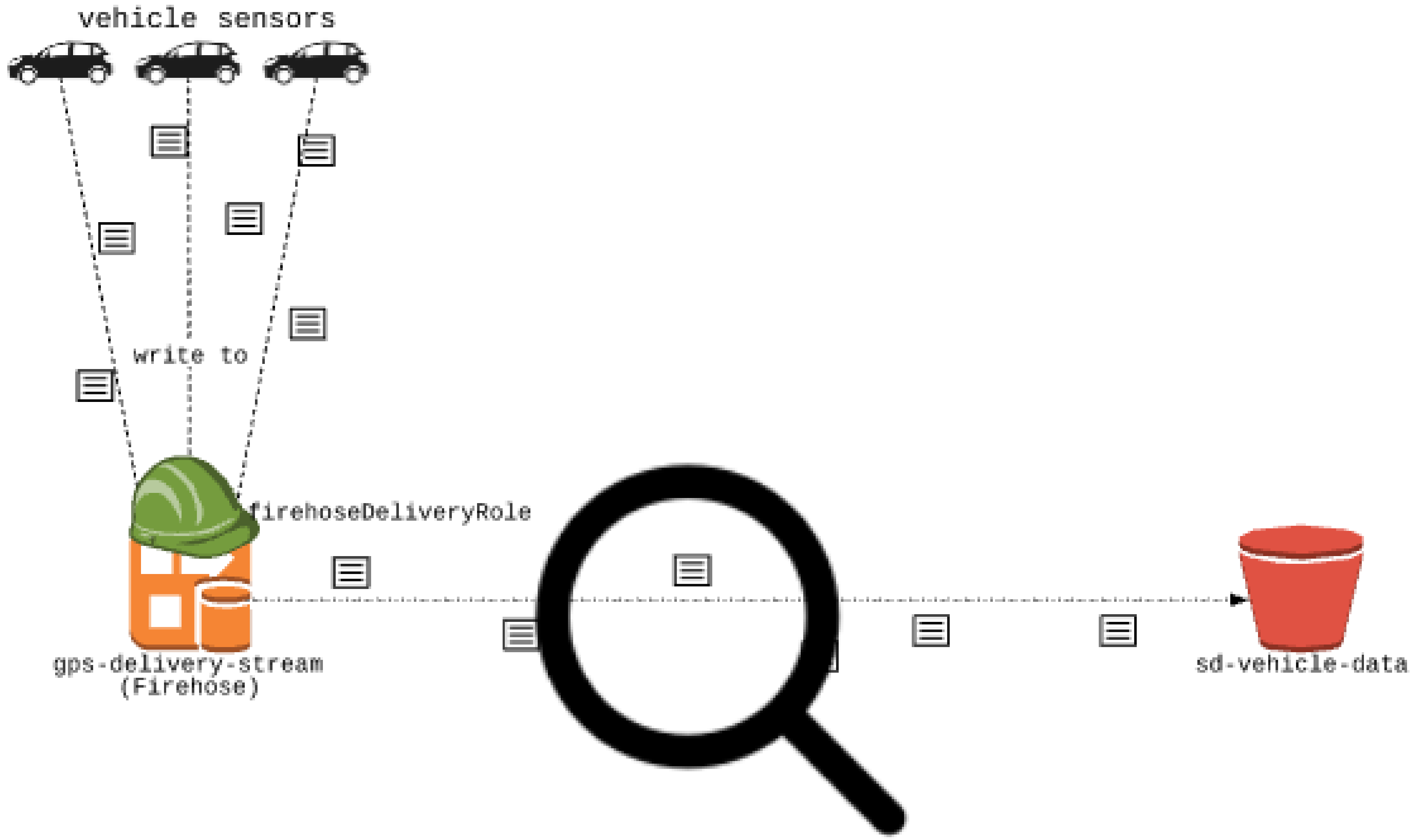
STREAMING DATA WITH AWS KINESIS AND LAMBDA

Maksim Pecherskiy
Data Engineer

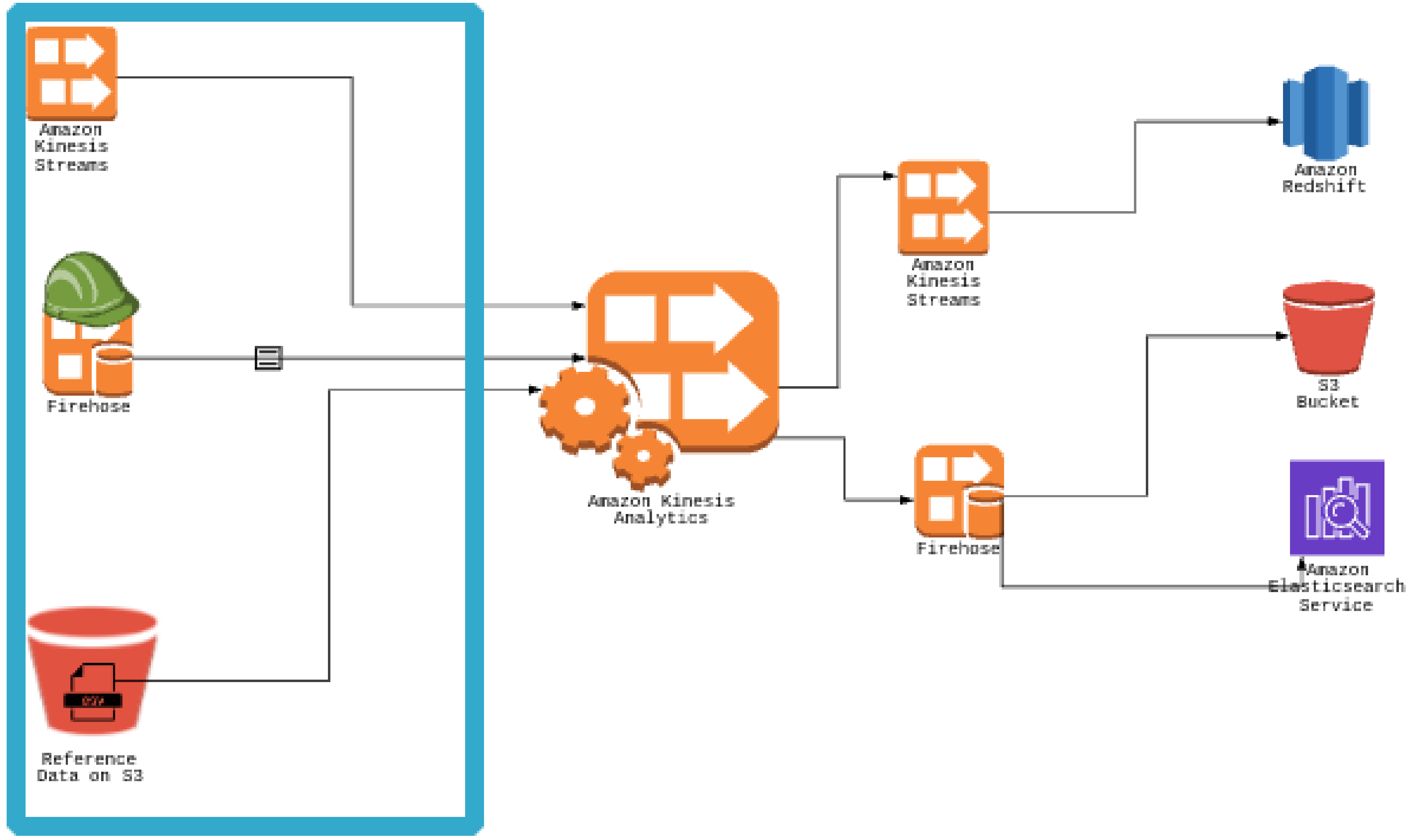
Last lesson


vehicle sensors
write to
firehoseDeliveryRole
gps-delivery-stream
(Firehose)
sd-vehicle-data

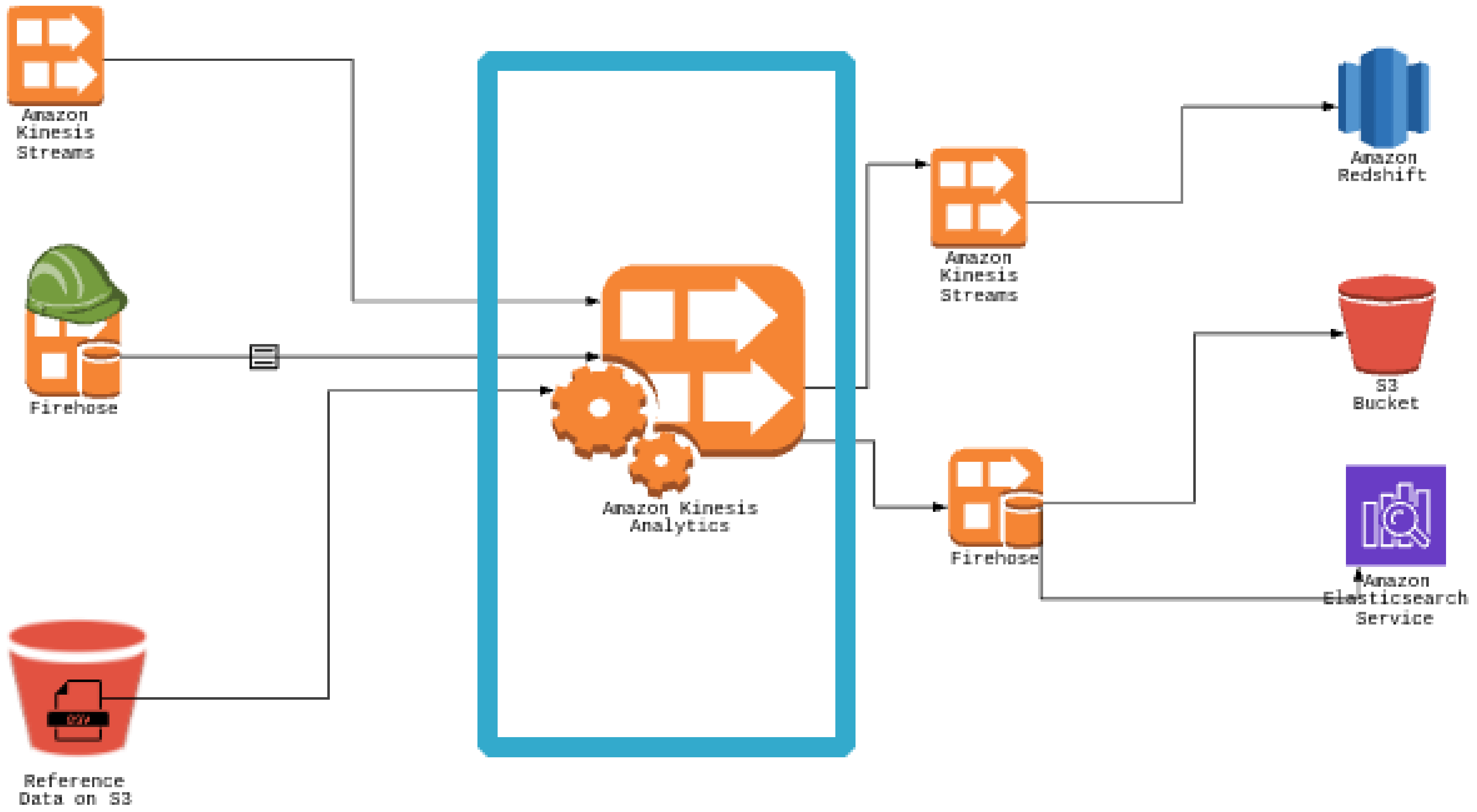
This lesson


vehicle sensors
write to
firehoseDeliveryRole
gps-delivery-stream (Firehose)
sd-vehicle-data

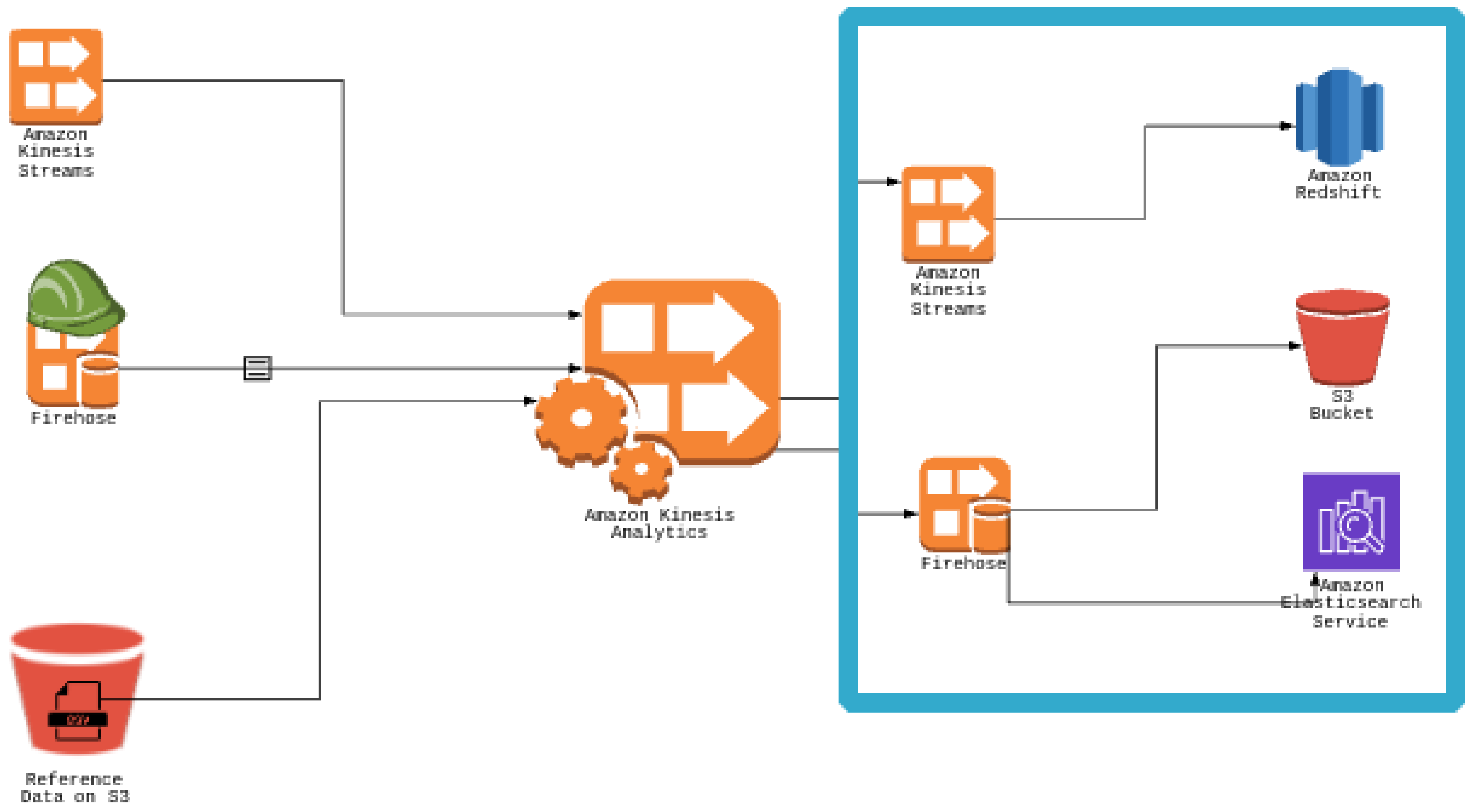
Kinesis Data Analytics



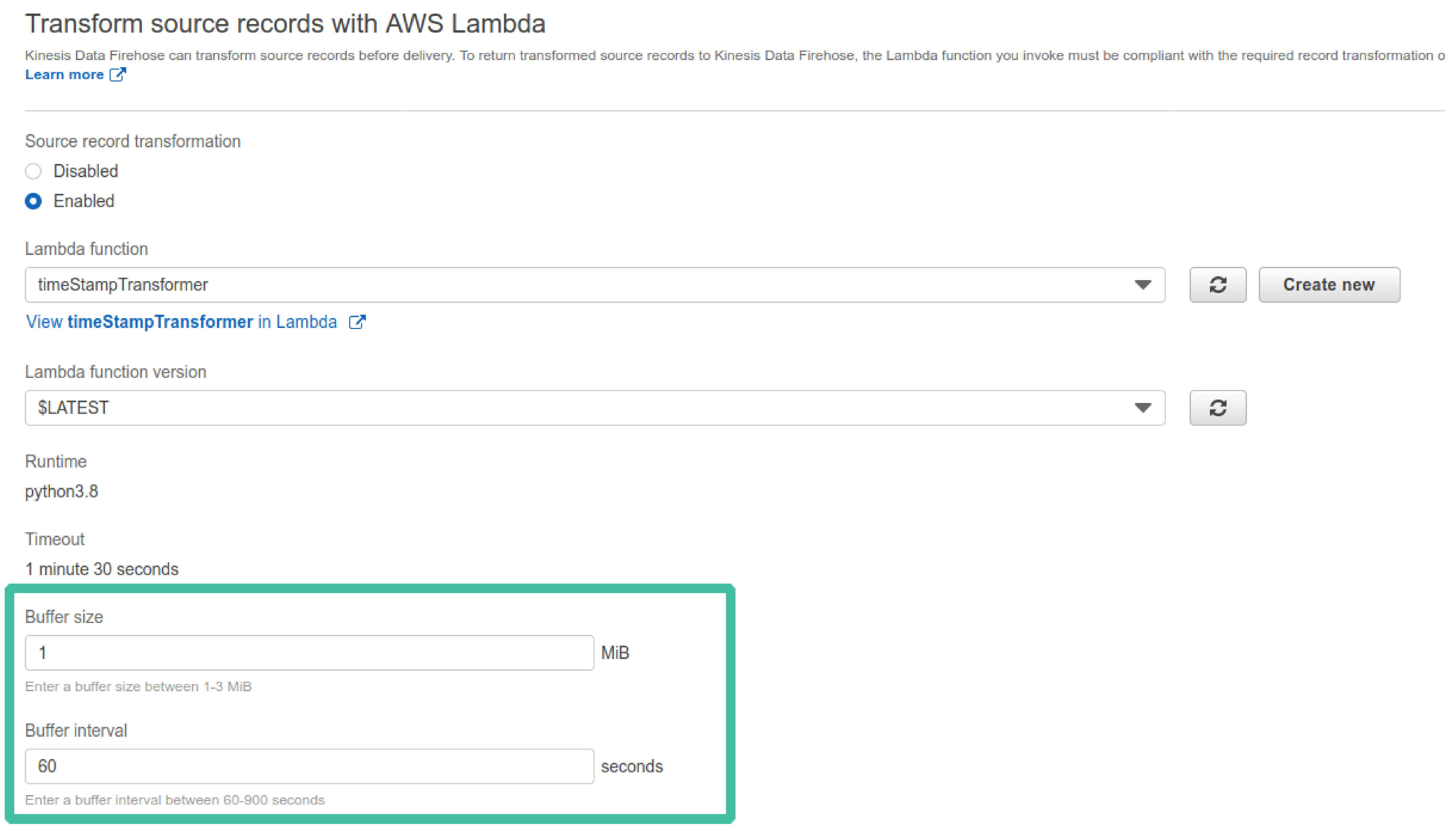
Kinesis Data Analytics



Kinesis Data Analytics



Why Kinesis Data Analytics



Transform source records with AWS Lambda

Kinesis Data Firehose can transform source records before delivery. To return transformed source records to Kinesis Data Firehose, the Lambda function you invoke must be compliant with the required record transformation o

Learn more

Source record transformation

Disabled

[✓ Enabled]

Lambda function

timeStampTransformer

View **timeStampTransformer** in Lambda

Lambda function version

$LATEST

Runtime

python3.8

Timeout

1 minute 30 seconds

Buffer size

1 MiB

Enter a buffer size between 1-3 MiB

Buffer interval

60 seconds

Enter a buffer interval between 60-900 seconds

Kinesis Data Analytics vs transformation Lambdas

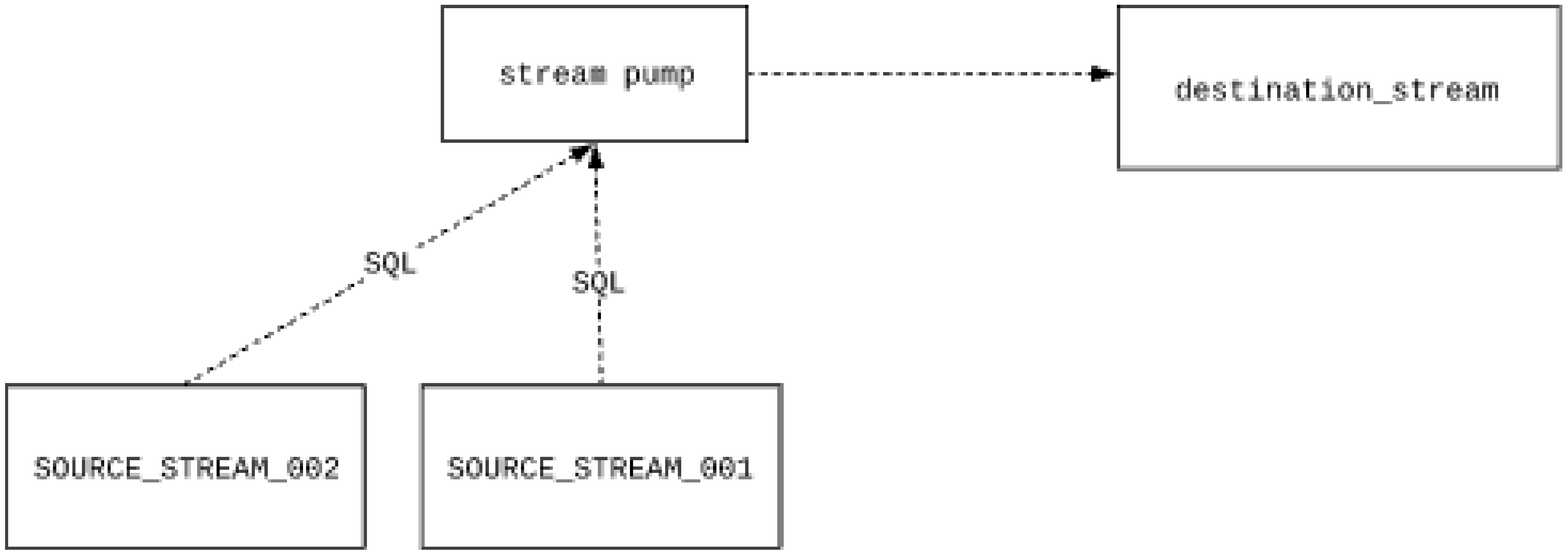
Transformational Lambda

- Python + Pandas
- Filter / aggregate
- Fixed window
- Great for data transformations per item
- Cannot combine multiple streams
- Not the best way to send output to another destination

Kinesis Data Analytics

- SQL
- Filter / aggregate
- We control the window
- Lets us look at the stream in chunks
- Can combine multiple streams
- Can send output to another stream or other destinations

Kinesis Data Analytics SQL



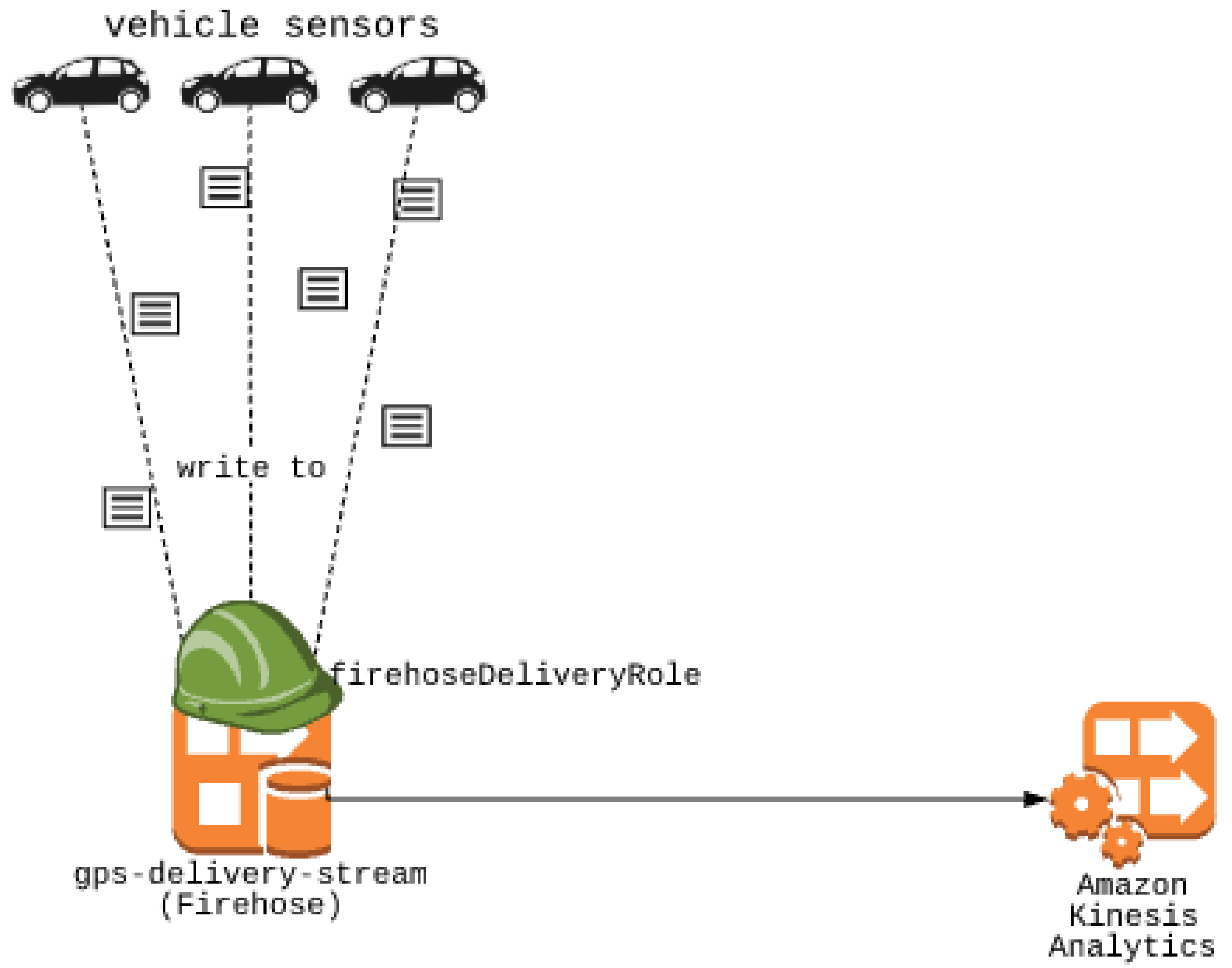
Kinesis Data Analytics SQL

```sql
-- Create destination SQL stream
CREATE OR REPLACE STREAM "DESTINATION_SQL_STREAM"
(ITEM VARCHAR(1024), ITEM_COUNT DOUBLE);
-- Create the pump
CREATE OR REPLACE PUMP "STREAM_PUMP" AS INSERT INTO "DESTINATION_SQL_STREAM"
-- Pump the results
SELECT * FROM "SOURCE_SQL_STREAM__001"
```

Some options

- Join multiple streams
- Enrich data from a static list using a join
- Find anomalies in the data
- Continuously filter data
- Find the top x repeating items in a timeframe

Finding overpingers



Let's practice!

STREAMING DATA WITH AWS KINESIS AND LAMBDA

Creating a Kinesis data analytics application

STREAMING DATA WITH AWS KINESIS AND LAMBDA

Maksim Pecherskiy
Data Engineer

Let's practice!

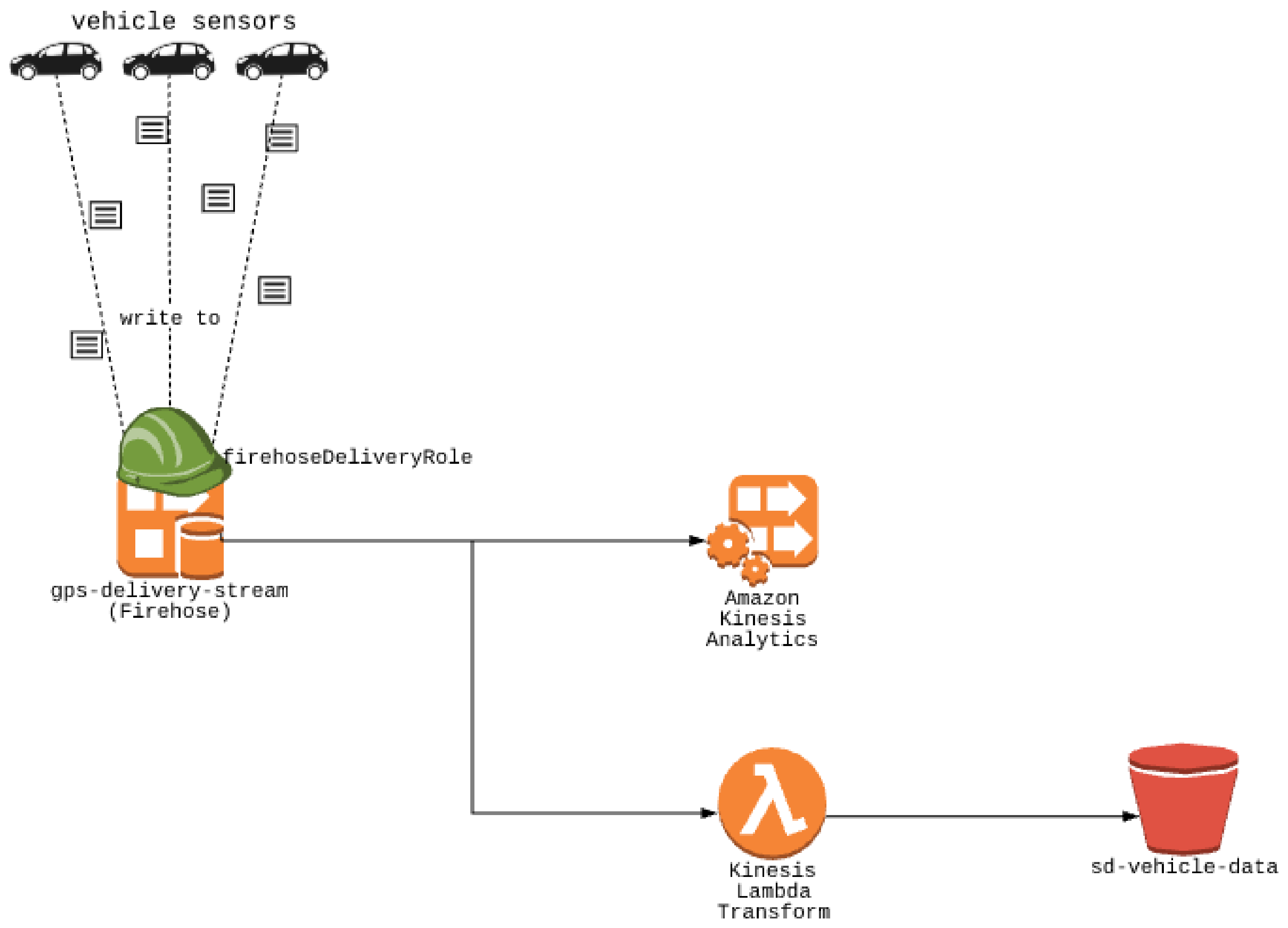
STREAMING DATA WITH AWS KINESIS AND LAMBDA

Using multiple streams

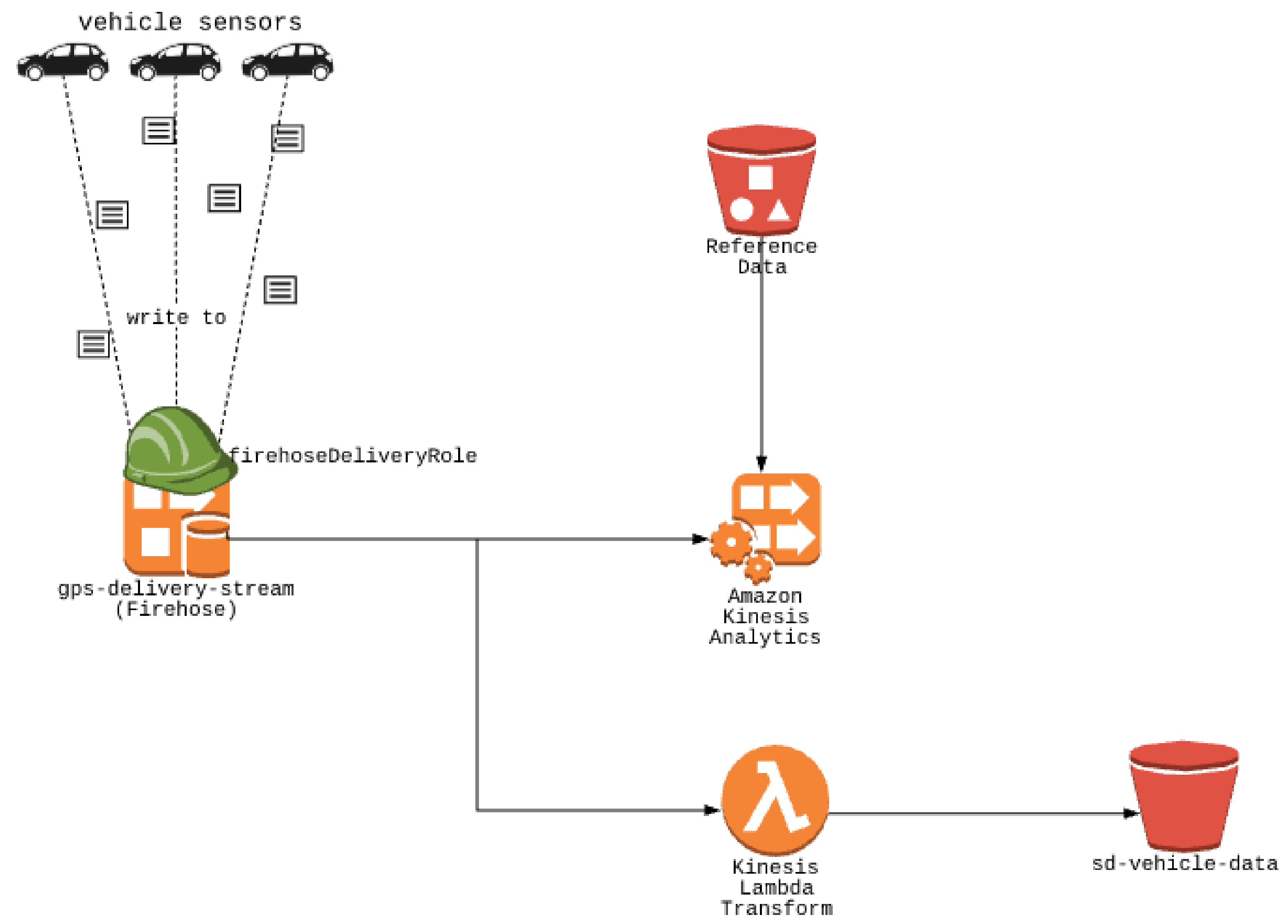
STREAMING DATA WITH AWS KINESIS AND LAMBDA

Maksim Pecherskiy
Data Engineer

In this lesson



Adding reference data



Adding reference data

vin	sensor_id
4FTEX4944AK844294	22885604061367
3FTEX1G5XAK844393	45832995035180
2LAXX1C8XAK844292	67875841488278
5FTEX1MAXAK844295	55623002258298
1FTEX1C8XAK855191	69566364579353

Let's practice!

STREAMING DATA WITH AWS KINESIS AND LAMBDA

Delivering data from Kinesis Analytics

STREAMING DATA WITH AWS KINESIS AND LAMBDA

Maksim Pecherskiy
Data Engineer

Let's practice!

STREAMING DATA WITH AWS KINESIS AND LAMBDA