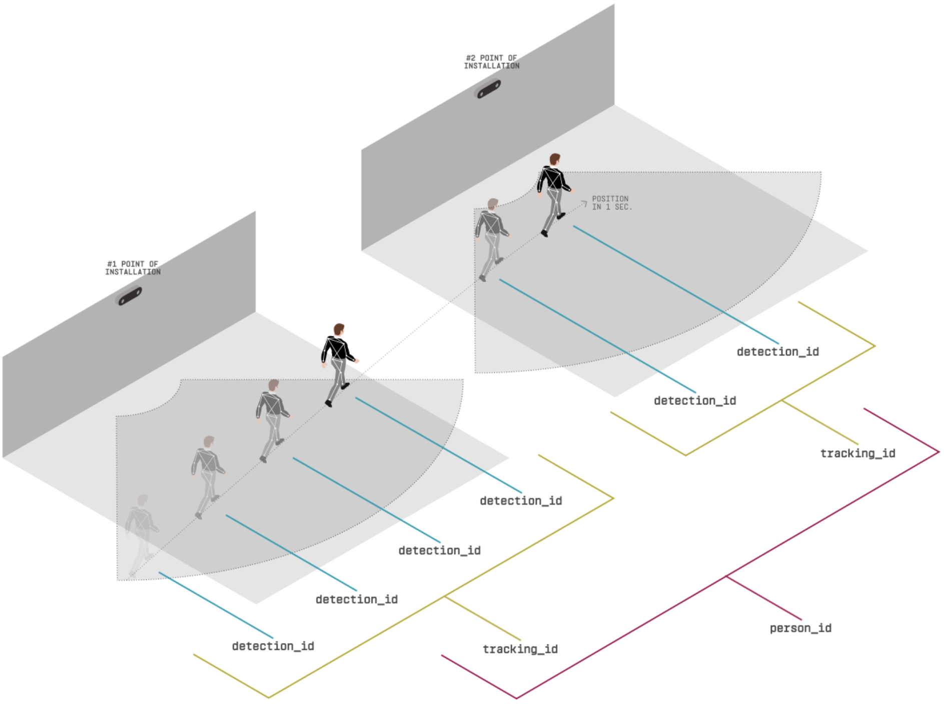
Advertima Intern (NodeJS/JavaScript) task

Advertima operates a number of edge devices with connected sensors (cameras). Each device has the ability to calculate analytics in real-time.

The sensors scan the environment in front of the screen and constantly calculates metrics about the detected persons.

For example, one of the record types is called *Tracking*.
When a person is filmed, each camera frame is analyzed and creates one detection (anset of information that describes the person). Detections are produced as long as the person is in the camera's field of view.
As soon as the person is *gone* (not in the field of view of the camera), detections will not be produced anymore for this person.
At this moment, we generate a *Tracking* record for this person.



Here is a simplified example of this record:

```json
{
  "trackingId": "97804e07-7c77-4b6d-a4bc-0cfe9b30a5f8",
  "personId": "d107bc94-a97b-4164-b08f-63259ecdd12f",
  "age": 45,
  "gender": "female",
  "startLocalTimestamp": 1537362301000,
  "endLocalTimestamp": 1537362303000,
  "totalViewTime": 1500,
  "coordinates": {
    "x": 0.393,
    "y": 1.234,
    "z": 0
  }
}
```

The purpose of this task is to evaluate candidates' code writing, solutions delivery, and problem-solving skills by giving them the possibility to implement such a calculation.

Task

Build a **node.js application** responsible for **calculating the Tracking records**.

The component responsible for creating detections from the video stream will be replaced with a given JSON file that you will need to read from.

The records will be produced in realtime and emitted via WebSocket.

The server will expose an HTTP endpoint `/metrics` that returns the number of total trackings and the average tracking time since the server started.

You need to

- Read from the given JSON file `data.json` (you're free to decide which strategy to use in order to simulate the interval between the events)
- Calculate the Tracking records
- Emit these records to a WebSocket
- Expose an HTTP endpoint that returns the number of total trackings and the average tracking

Note: you do not need any database or persistence, everything can be done in memory

What we expect from you

- a working node.js application

- the project is a git repo (either pushed in GitHub or a zipped git repo) with proper commits
- a `README.md` explaining...
  - how to run the application
  - your approach, choices, assumptions

Nice to have

- your code has unit tests
- the app is written in TypeScript if needed, you used RxJS to handle event streams
- your application has a Dockerfile