Example architecture schematic

As a GCP managed Neo4j service

Load balancer
e.g.,Kubernetes

The slice of architecture represented by this technical test

Routing
Express.js

Server side
Nest.js REST API

Client app
e.g., React.js

Requests

Frequently made requests

Cache

Auth

What data might we cache?
- Data loaded by everyone (e.g., promoted events on the front page)
- Profiles for popular users or events (e.g., celebs)

What tables might we have?
- Metadata / Users / Events

Big Query
Tabular db

Read & Write

Read

Replication / Sharding

Neo4j
Network db

What relationship data might we have?
- Interactions between users
- Events users have attended
- Similar events/users tags

Why don't we store everything as a network db?
Neo4j is well suited for highly related data, but it does not represent highly complex properties, or information about the nodes and relationships in a system.

New writes

Scheduler
Trigger every *n* seconds /events

CRON job
Aggregate recent data

Add new writes to network db

What is this?
Neo4j is not built for very high data throughput services, it would be slower and more unreliable when we have lots of concurrent writes.

Therefore, we will need a way of aggregating recent changes into one larger batch that can be processed at a lower frequency.

NB. *We would likely also need blob storage for unstructured data like files and images e.g., buckets*