**Technical task response:**

Solution have been built in Mulesoft API development platform. Which provides a reliable, secure, scalable environment to create API gateways and implement backend business logic in underlying mule ESB flows to process requests and generate the corresponding requests.

Version: Mule server **3.8.1** EE

Solution comprised below high level steps:

1. Develop RAML in any point platform (API Design)
2. Create mule project with API dev tool kit. This creates required artefacts as mocking console and main api flow.
3. Develops mule flow to process request in mule server.
4. Configure required parameter as open weather API details, API Key, city lists, threshold values.
5. Deploy the mule application over cloudhub.

APIs can be accessible at following URL

<http://rocketo-weather-apic.us-e2.cloudhub.io/weather?city=Perth> (For selected city)

<http://rocketo-weather-apic.us-e2.cloudhub.io/weather> (For all cities)

**Configuration:**

1. City list configuration:

All cities (Melbourne, Darwin, Hobart, and Perth) has been configured in configuration file along with the threshold values. Which can be changed later. For instance, we can add another city in configuration file without changing the code.

The configured values are being read in mule flows dynamically and perform the required requests.

1. City location code configuration. This allows user to enter the city name rather than code in URL
2. There are fields like open weather related details are configured in configuration file to avoid any code change.

**Code reusability:**

The mule flows have been divided in four parts as main flow and sub flows. All sub flows can be called in other flows if required with no code change.

1. Fetch the threshold values from configuration file. (sub flow)
2. Find city location code (sub flow)
3. Call to open weather API (sub flow)
4. Main flow

The above 3 flows are working independently and can be utilized from other mule flows if required.

**Security-**

Currently the weather API is deployed on Mule any point platform and the backend code (mule application) is deployed over cloud hub. API Gateway protects the underlying backend APIs and also limit the number of access using policies.

The APIs can be protected through various policies at run time. For instance, currently the following policies are enabled.

1. Throttling (which allows only 2 request per 10 sec)

\*Besides this we can add OAuth2.0 mule component for authentication of user.

**Scalability-**

Solution deploy over cloud are auto scalable. More processing power can be added on the fly. Moreover, the auto scaling can be triggered by policies as CPU & memory usage of the server.

**Performance-**

Mule flows and the message processor activities are highly optimized piece of code. However, there are various way to tune the mule application as data weave operations, remove the redundant design activities, configuration settings etc. Moreover, the underlying JVM also can be tuned for heap memory and garbage collection settings.

Automation-

Versioning-