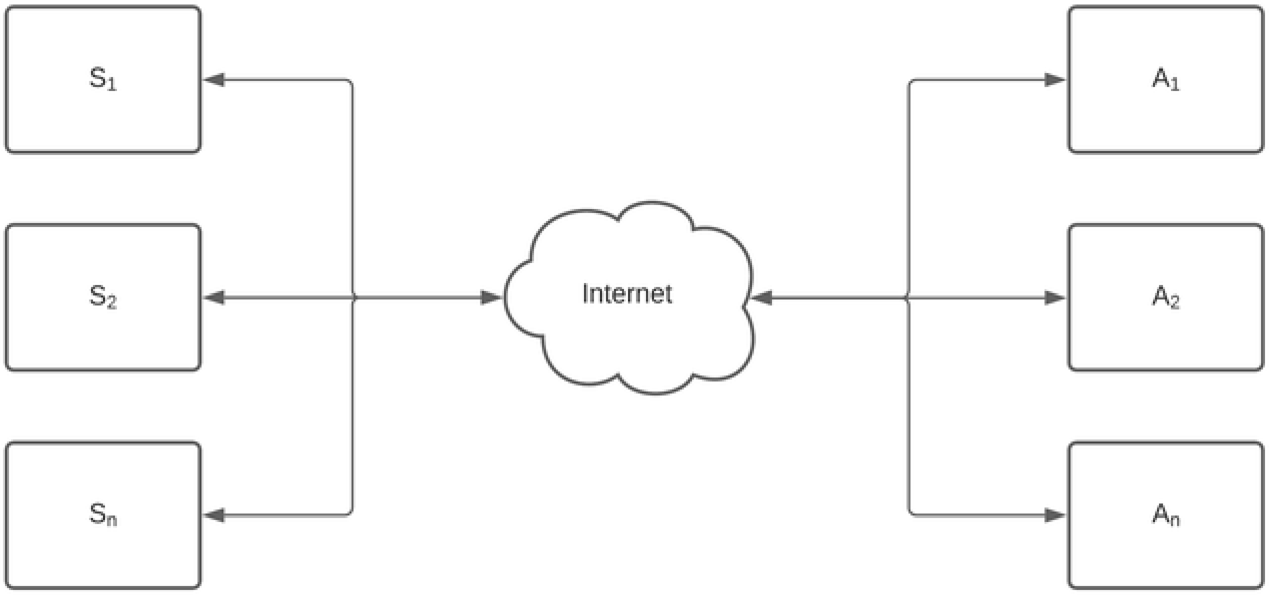
Software development proficiency test

Problem definition

A large number (~2000) of client services (S_1, S_2, …, S_n) need to communicate with a handful of cloud applications (A_1, A_2, …, A_n) over the public internet. Suggest at least one method for the applications to be able to authenticate requests coming from the services and implement the best solution you can using Python 3.



Requirements

- Different client services (S_1, S_2, …, S_n) can have different permissions in application (A_n).
- Client service (S_n) can have different permissions in different applications (A_1, A_2, …, A_n).
- Ability to add, update and remove client permissions without changing any code.

Example

Consider application (A_1) with an endpoint called `https://app-A1-domain/my-secure-endpoint` and service (S_1) that wants to call that endpoint and do something with secret data it expects to receive. How would you implement a secure method for the application to authenticate that the service should be allowed to receive the response with the secret data? The method needs to scale well with all clients and applications as well as fulfill the requirements above.

Code Examples

Client service (S_1) example:

```
import requests

res = requests.get(“https://app-A1-domain/my-secure-endpoint”)

do_something(res)
```

Cloud application (A_1) example:

```
from flask import Flask
app = Flask(__name__)

@app.route('/my-secure-endpoint')
def my_secure_endpoint():
    return secret_data
```