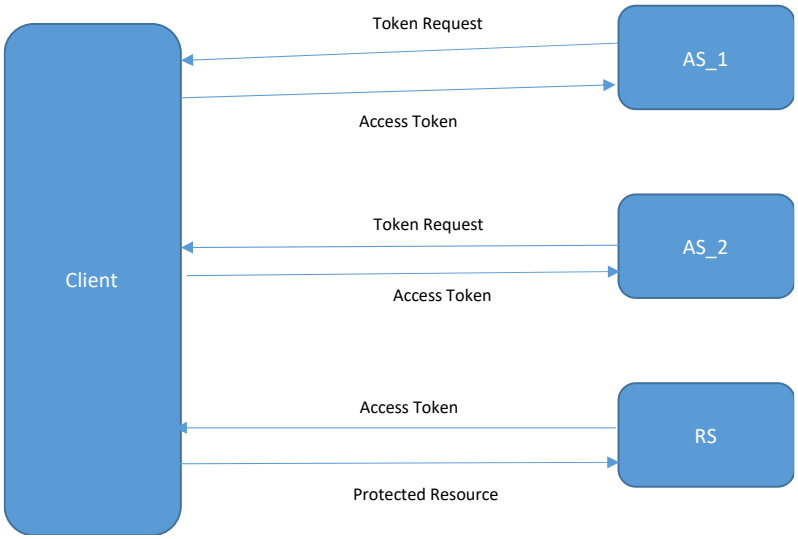
CS 45231 Internet Engineering
Final Project

In this Final project, you will write a security application in python, this application will implement the access control that manages the access to sensitive resources.

Background:

Access control is a fundamental security mechanism by which we control who access what resource.



Assignment

You have to use python to implement the following:

1. Peer.py (a class from which you will instantiate four objects: AS_1 , AS_2, RS, and Client)
2. Communication protocol.py
3. Resolution resolver.py
4. Message.py

- **Peer class has the following components:**
    1. **Resource data structure (L)** : a JSON format file: { "C1": "discount 20%"
       "C2": "Credit Score 780"
       "C3": "Alex.kent.edu",
       "C4": "ID81023456"
       }
    2. **Policy data structure**: a JSON format file { "C1":[["C2", "C3"],
       "C2": [True],
       "C3": ["C4"],
       "C4": "TRUE"
       }

3. send a message method: call the message class and create a message to other peers
4. Receive a message method:
5. Resolution Resolver: an algorithm that will be invoked when a message m is received. It extracts the requested credential (C) from the message, check the policy and send new messages to request all the item in the policy from different parties. See the example below

The Peer Resolution Resolver Algorithm is:

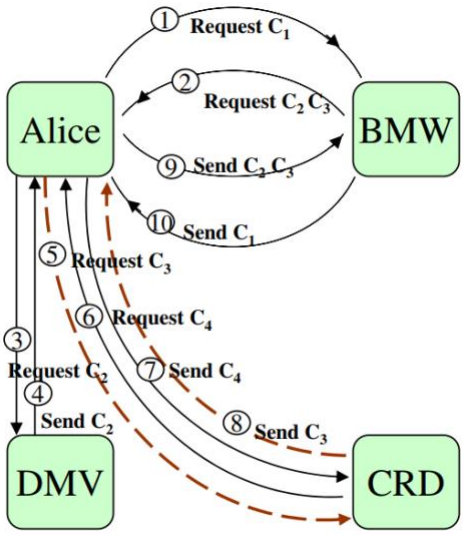
```
1. Resolution Resolver (M_received, M_sent, ) {
    Let Pthis be the current peer
2.  m = the latest message in Mreceived
3.  Qsent = set of credentials Pthis requested from others
4.  Qreceived = set of credentials others requested from Pthis
5.  Qnew=∅
6.  Dsent = set credentials Pthis sent to others
7.  Dreceived = set of credentials Pthis received from others
8.  Dnew=∅
9.  if m is an offer of credential C  then
    /* Calculate new credentials Dnew that Pthis will send to other parties */
10. Dunlocked = all credentials unlocked by C and other credentials in Dreceived
11.  Dnew = Dunlocked ∩ Qreceived − Dsent
12. else if m is a request for credential C then if C is already unlocked then
13. Dnew ={C}
14. else
    /* Calculate new credentials Qnew( based on the policy)  that Pthis will request
    from others */
    Drelevant = all relevant remote credentials for C # need to be requested form other
    Qnew = Drelevant − Dreceived − Qsent
Return the list of messages M composed of credentials in Dnew and requests for credentials
in Qnew
        }
```

- **The communication algorithm:** an algorithm that manages the sending and receiving of messages between peers.

```
Participate In OAuth () {
    while willing to participate do
        if the incoming message queue is empty then
            Wait a short period of time for new messages
        if there is a new message in the incoming message queue then
            Choose such a message m and remove it from the queue
            Record m's receipt time as the current time
            ProcessMessage(m)}
/* message handler */

ProcessMessage (m) {
/* Mreceived and Msent store received and sent messages respectively */
Mreceived = Mreceived ∪ {m}
If m is an offer, add m to the local knowledge base L ( resources_vault)
Apply the resolution resolver algorithm with parameters Mreceived, Msent, and L,
which returns a list of messages M
/* send the messages to their intended recipients */
if M is not empty then
    for every message k in M do
        Send k to its specified recipient
        Record k's sending time as the current time
        Msent = Msent ∪ {k}
}
```

Example of message flow between parties:




```
C1: BMW PremierDiscount(Alice)
C2: DMV DriversLicense(Alice)
C3: CRD GoodCredit(Alice)
C4: Alice PermitRelease(CRD)
C5: CRD GoodCreditScore(Alice)
```