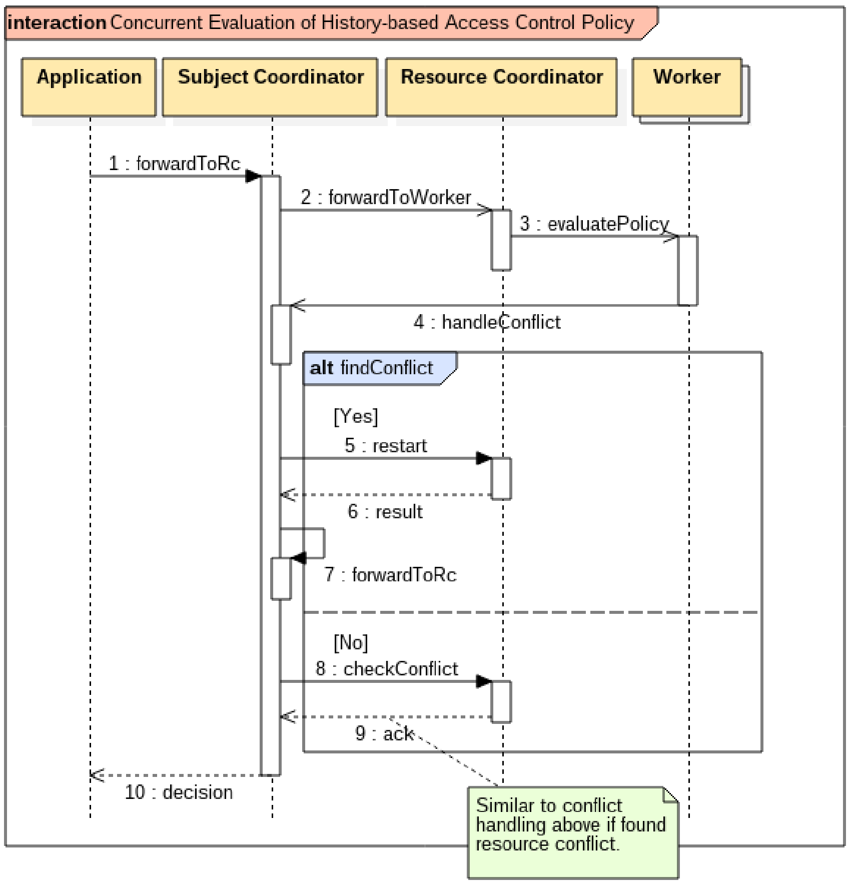
Name: Zhitao Fan, Chen Dai

The whole processing sequence is illustrated as followed.



(I) Application

- Step 1: Determines the coordinator that is responsible for the subject in question based on the id of the subject and the list of coordinators
- Step 2: Sends a policy evaluation request to this coordinator

```
forwardToSC(req):
    sc = lookup(req.subject)    # step 1
    send(sc, req)               # step 2
    resp = await()
```

(II) Subject Coordinator

- Step 3: Assigns a globally unique id to this evaluation, - Sets up the administration for the subject, - Adds any tentatively updated attributes to the request, - Determines the coordinator responsible for the resource in question - If it dectes a conflicting update of a subject attribute, - it immediately restarts the evaluation.
- Step 4: Forwards the authorization request to that coordinator
- Step 9: Checks whether the evaluation employed subject attributes - that were updated in the mean while. - If not, it tentatively executes the updates of subject attributes
- Step 10: Asks the coordinator responsible for the resource - whether there are conflicts for resource attributes
- Step 13: Executes the tentative attribute updates, - Clears its administration for this evaluation
- Step 14: Passes the decision along to the application

```
forwardToRC(req):
    assignId(req)               #step 3
    setupAdmin(req)
    addUpdatedAttr(req)
    rc = lookup(req.resource)
    send(rc, self, req)         #step 4

handleConflict(worker, resp):
    if findConflict(resp)               # step 9: Optimistic locking
        restart(resp.id)
    else:
        add(tentativeAttrs, resp.wAttr) # Update to block conflicting requests
        send(rc, self, req)             # step 10
        wait(rc, ack)
        if ack == success:
            commit(DB, resp.wAttr)      # step 13
            clearAdmin(resp.scAdmin)
            send(app, resp)             # step 14
        else:
            restart(resp.id)

assignId(req):
    req.id = generate()

setupAdmin(req):
    req.scAdmin = createAdmin()

addUpdatedAttr(req):
    req.tAttr = tentativeAttrs

findConflict(resp):
    return isIn(tentativeAttrs, resp.wAttr) or
            isIn(tentativeAttrs, resp.rAttr)

restart(requestId):
    req = getRequestById(requestId)
    clearAdmin(req.scAdmin)
    forwardToRC(req)
```

(III) Resource Coordinator

- Step 5: Sets up the administration for the resource
- Step 6: Assigns the request to a worker
- Step 11: checks its administration, if there are no conflicts, - executes the updates of resource attributes, - clears its administration for this evaluation - If it detects a conflicting update of a resource attribute, - it notifies the sc.
- Step 12: Acknowledges success to the first coordinator

```
forwardToWorker(sc, req):
    setupAdmin(req)             # step 5
    worker = getFromPool()
    send(worker, sc, req)       # step 6

checkConflict(sc, req):         # step 11
    if (hasDetectedAConflicting()):
        nofityResourceConflict(sc)
    else:
        if checkAdmin(req.rcAdmin):
            updateResAttr(DB)
        clearAdmin(req)
        ackSuccess(sc)          # step 12

setupAdmin(req):
    req.rcAdmin = createAdmin()

restart(req):
    clearAdmin(req.rcAdmin)
```

(IV) Worker

- Step 7: Evaluates the policy for this request
- Step 8: Sends the result to the coordinator responsible for the subject

```
evaluatePolicy(sc, req):
    decision = doEvaluate(DB, req)
    rAttr = getReadAttr(DB, req)
    wAttr = getUpdateAttr(DB, req)
    resp = createResponse(req, decision, rAttr, wAttr)   # step 7
    send(sc, resp)                                       # step 8
```

(V) High level Data Structure

- 1.req: request for policy evaluation
    - subject: subject Id
    - resource: resource Id
    - id: assigned global Id
    - app: application instance
    - scAdmin: subject administration Id
    - rcAdmin: resource administration Id
    - tAttr: tentatively updated attribute list
- 2.scMap: mapping from subjectId to Subject Coordinator instance
- 3.rcMap: mapping from resourceId to Resource Coordinator instance
- 4.tentativeAttrs: Tentatively updated attributes
- 5.resp: response of evaluation
    - id: assigned global Id
    - app: application instance
    - scAdmin: subject administration Id
    - rcAdmin: resource administration Id
    - wAttr: attributes that should be updated
    - rAttr: attributes that were read