Control Loop Coordinator (CLC)

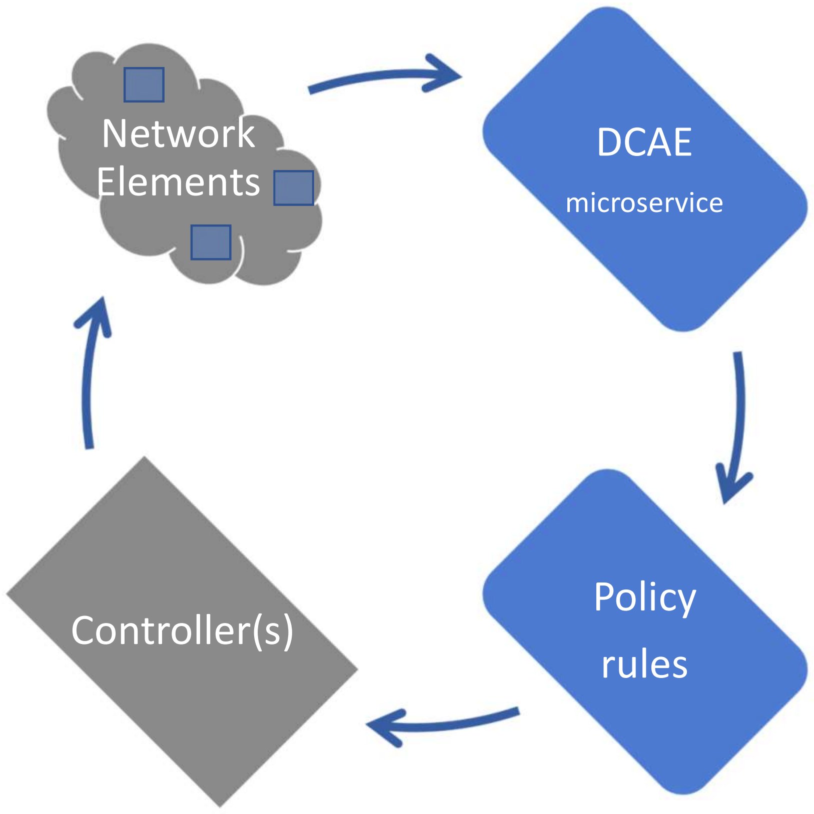
Joshua Reich

AT&T

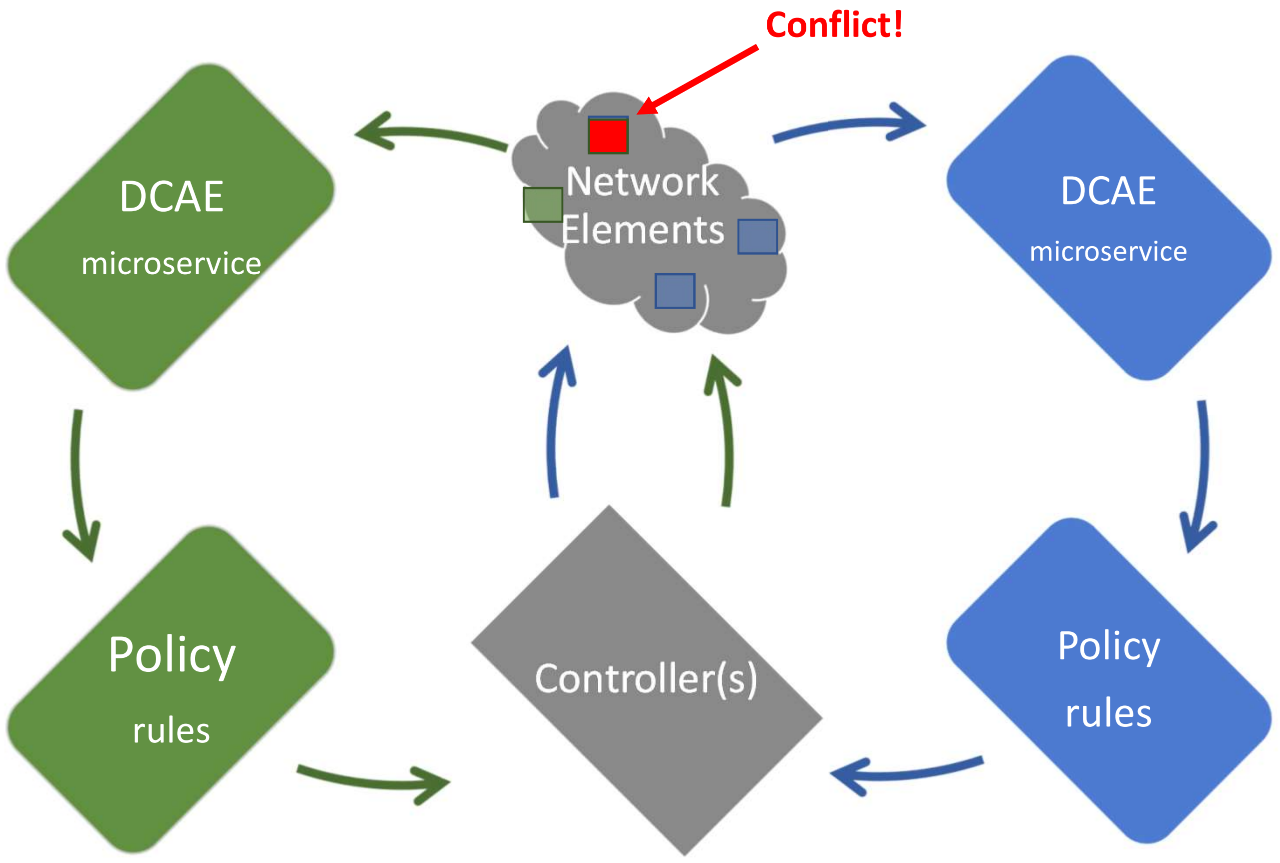
Summary

- Problem & proposed solution (functional view)
- Example coordination patterns provided by CLC
- Architectural view (& scaling)
- Codebase summary
- Implementation details

ONAP Control Loop (Functional)

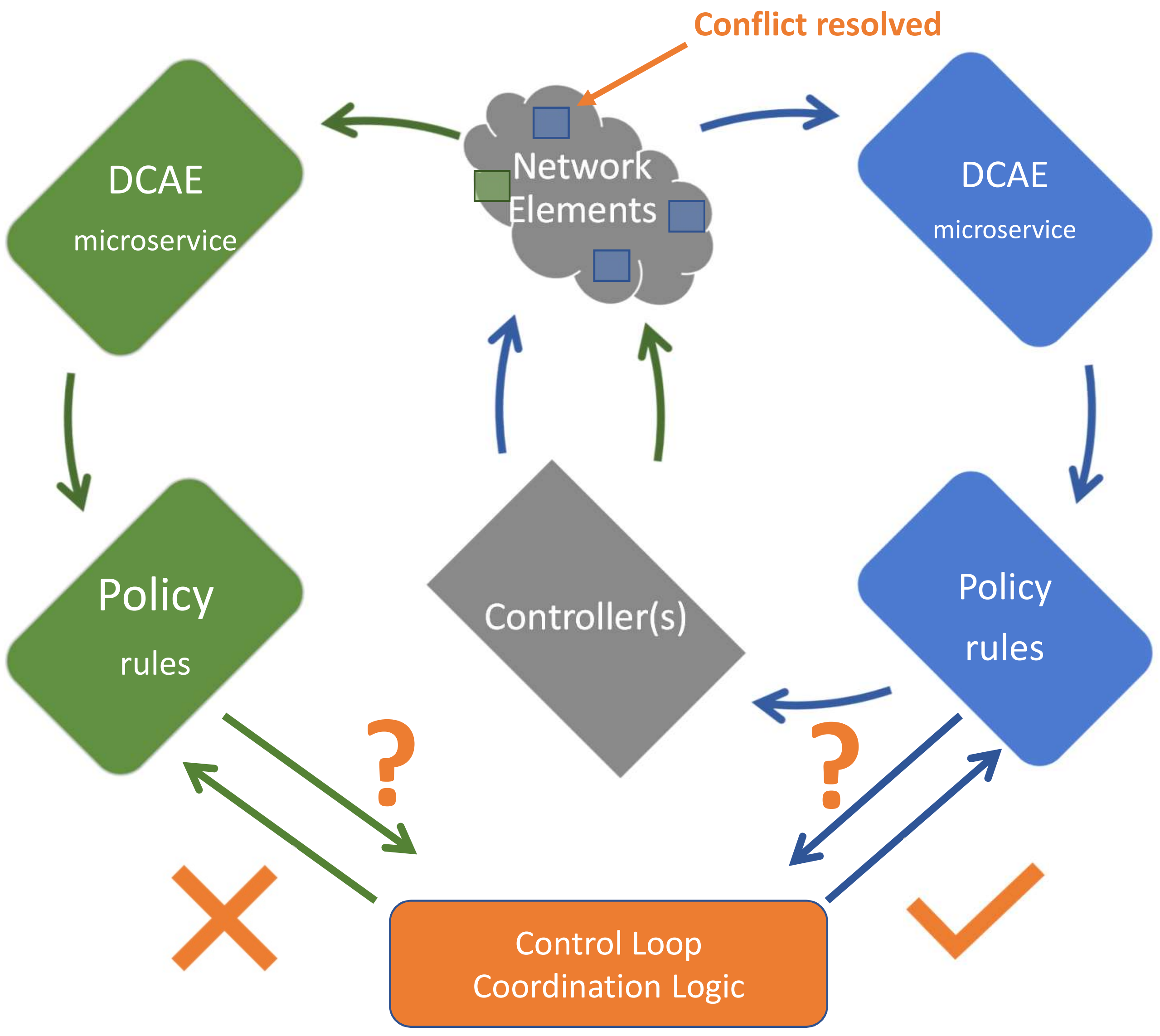


Two Loops (Functional)



CLC (Functional)

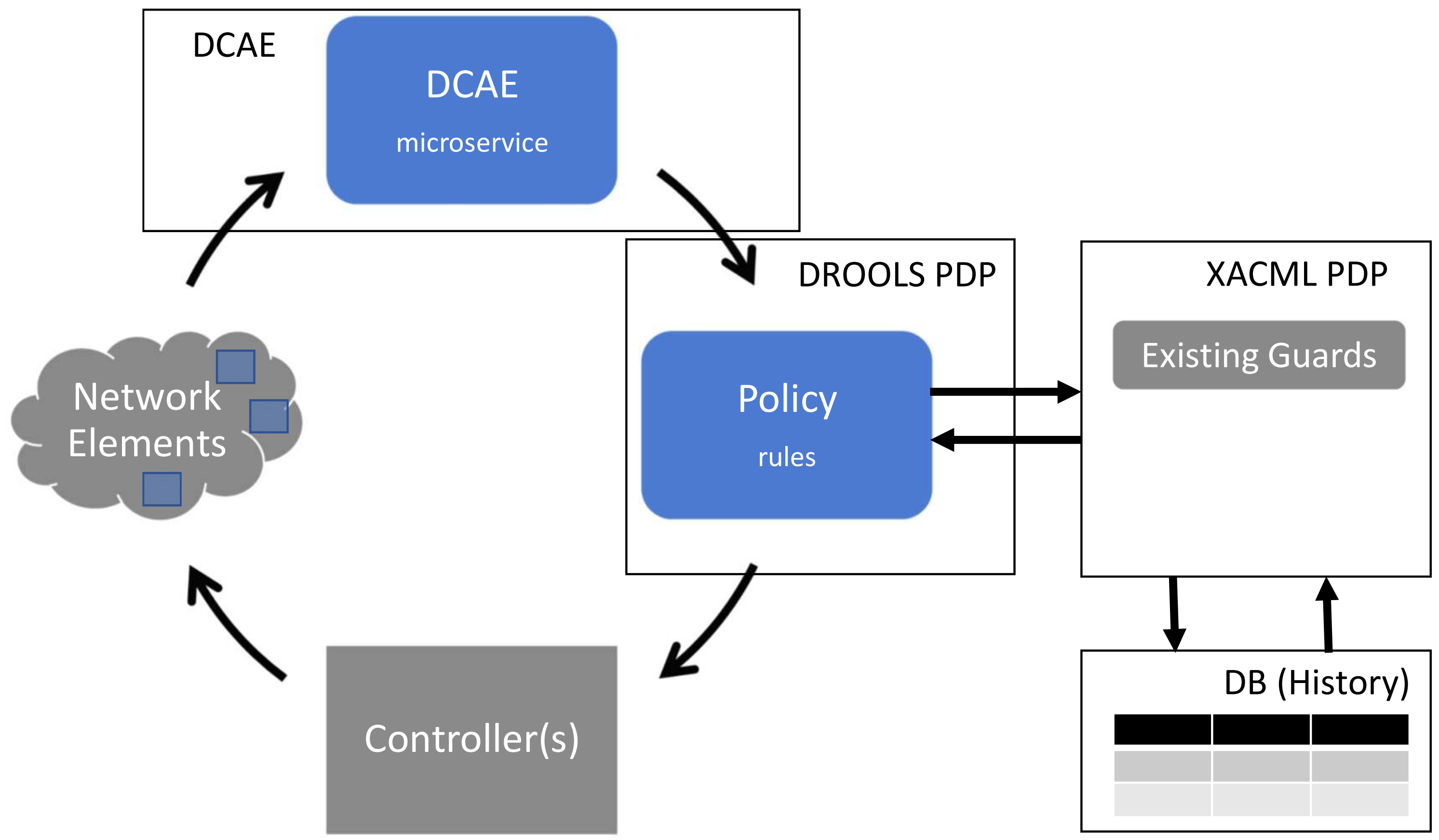
Directives to controllers dependent on CLC response e.g., under coordination pattern ‘prefer **blue**’



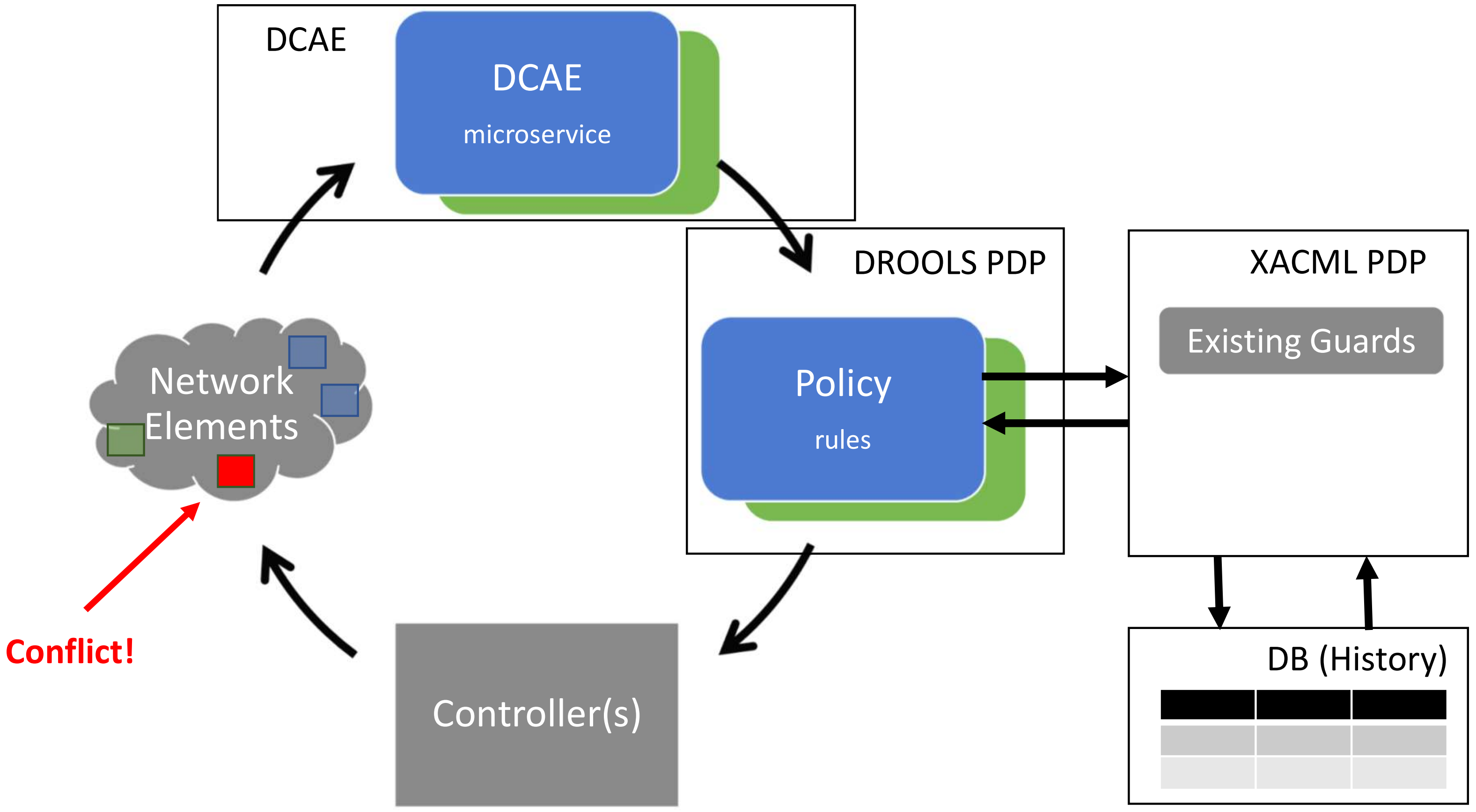
Example Coordination Patterns

```
• enable(G,B)  :=                                 permit G

• disable(G,B) :=   if B open actions,            deny G
                    else                          permit G

• block(G,B,t) :=   if B open actions  OR
                       now - B.closed < t,        deny G
                    else                          permit G

• preempt(G,B) :=   if B open actions,            cancel B
                                                  permit G
                    else                          permit G
```

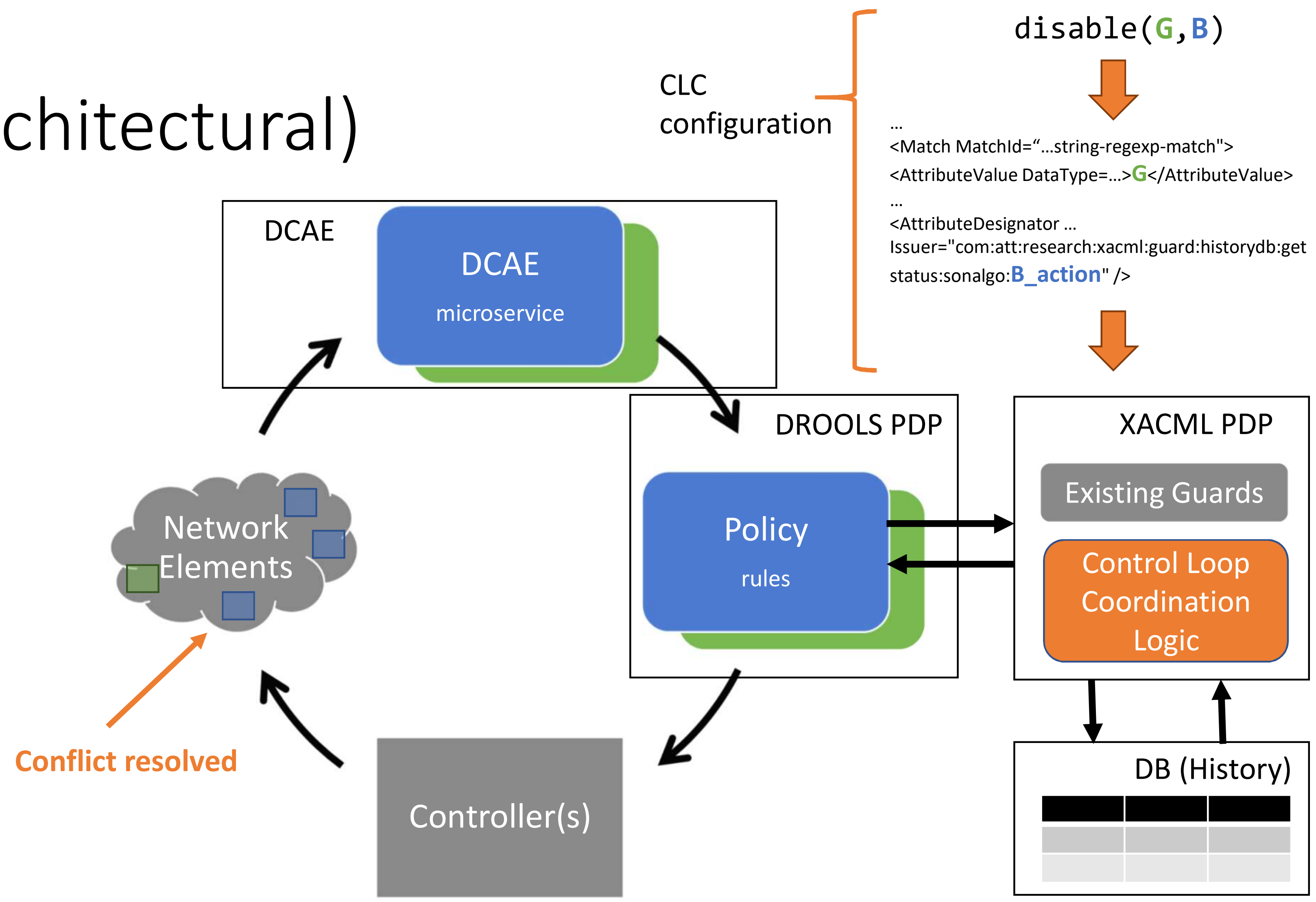
ONAP Control Loop (Architectural)



Two Loops (Architectural)

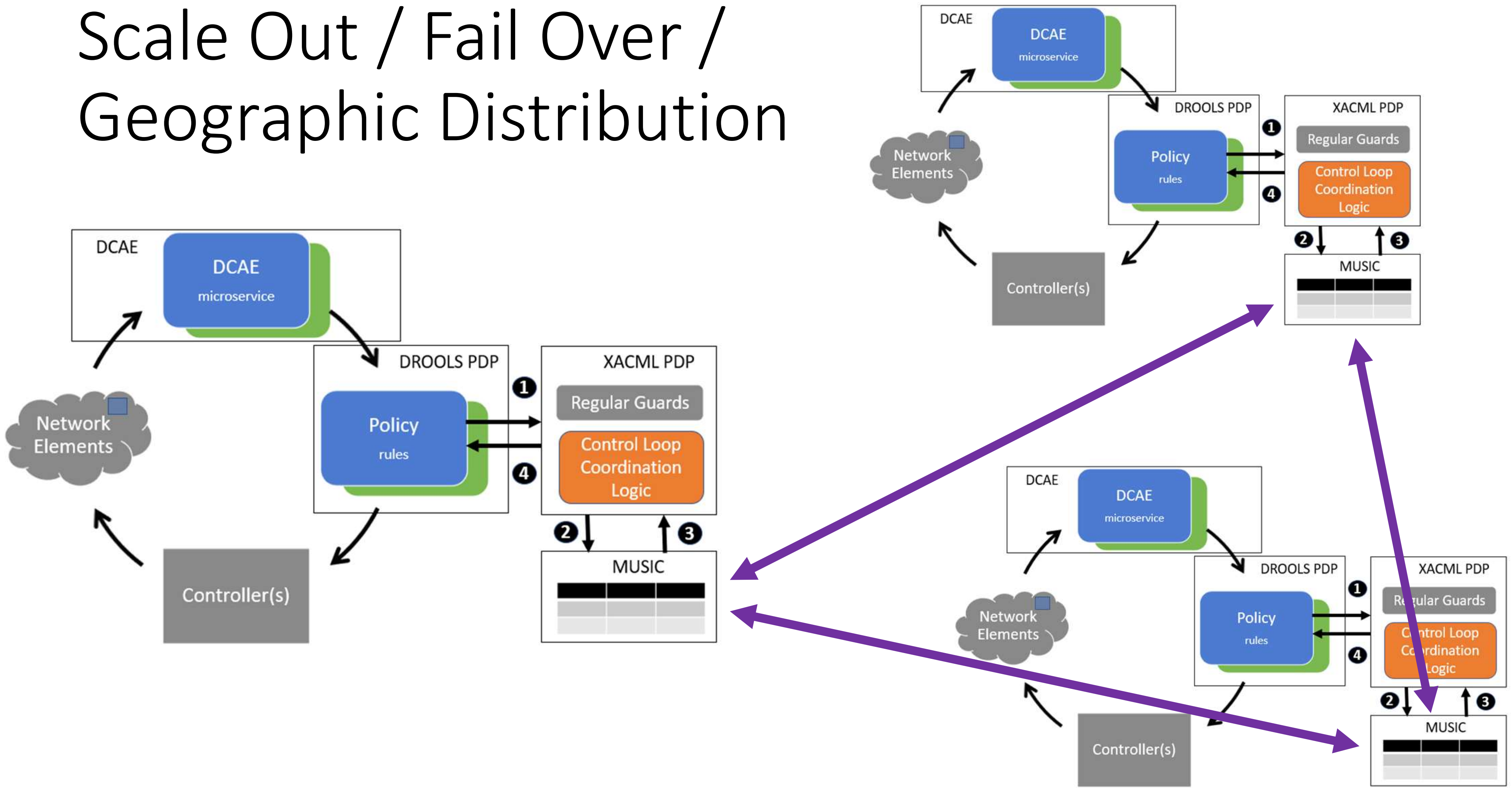


CLC (Architectural)



```
…
<Match MatchId="…string-regexp-match">
<AttributeValue DataType=…>G</AttributeValue>
…
<AttributeDesignator …
Issuer="com:att:research:xacml:guard:historydb:get
status:sonalgo:B_action" />
```

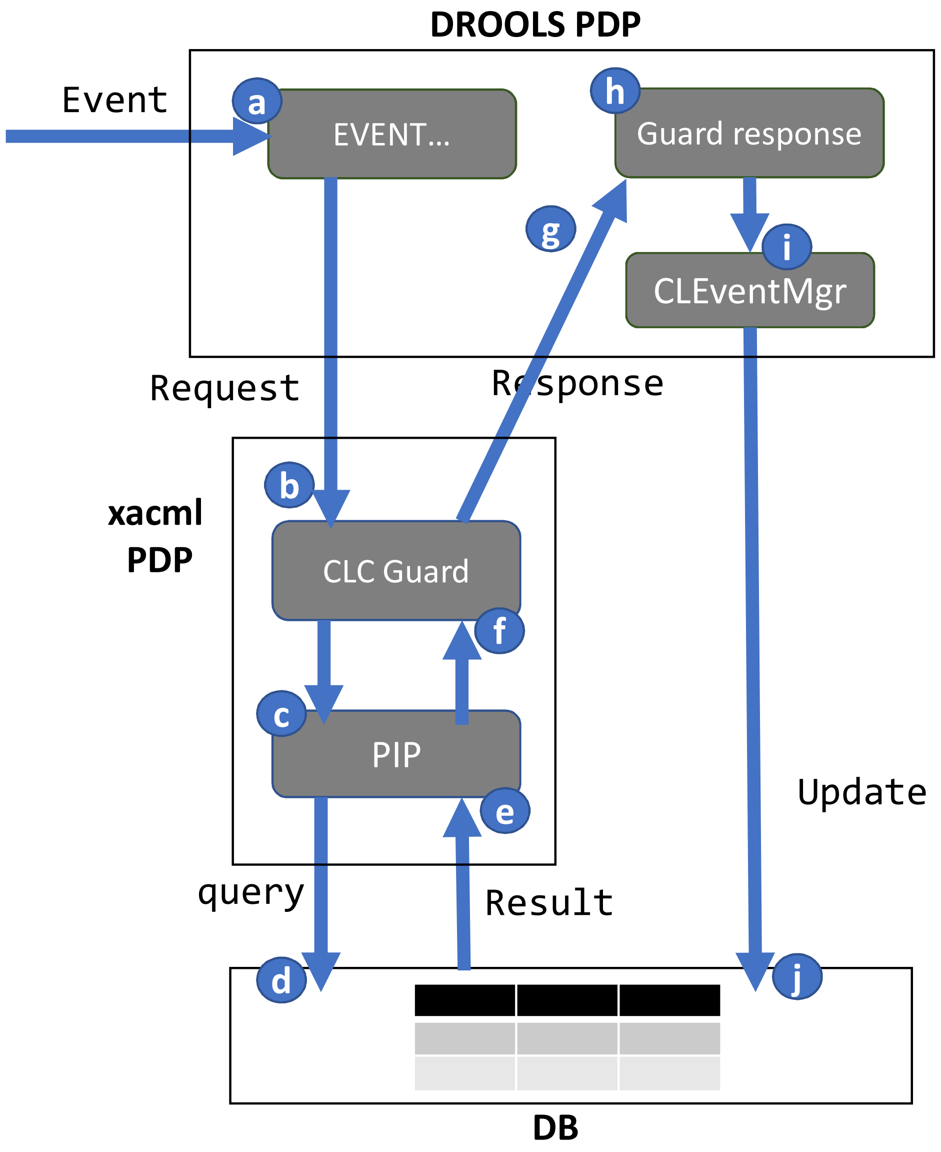
Scale Out / Fail Over / Geographic Distribution



Detailed DROOLS/CLC Flows

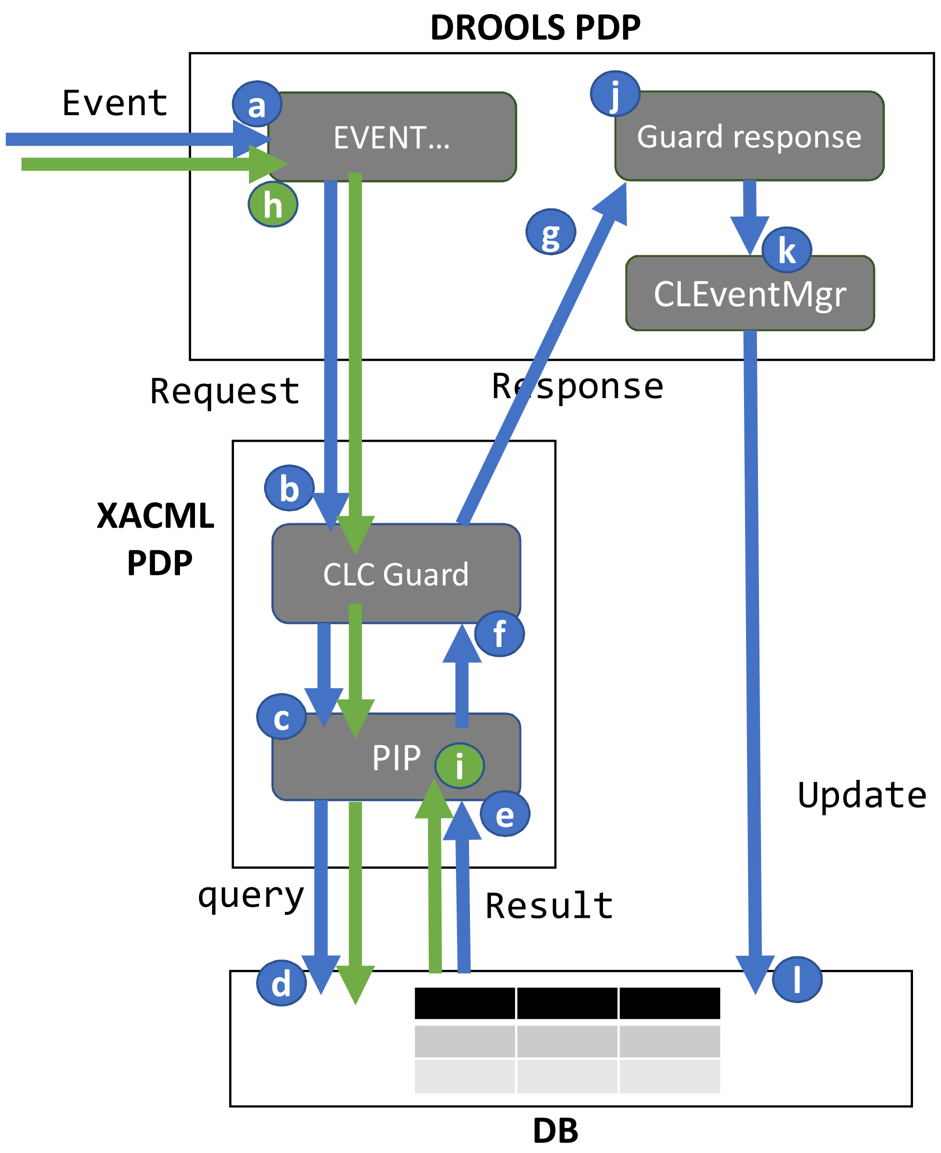
Detailed Flow

a. onset or abated event arrives
b. CLC guard triggered
c. xacml PDP requests variable
d. PIP issues SQL (JPA) queries
e. PIP receives results, calcs value
f. xacml PDP issues deny/permit
g. DROOLS rcvs, if action approved
h. Action rules triggered
i. summary logged via EventMgr
j. Action written to MariaDB



Race Condition!

- Second loop's onset event arrives
- & triggers DB query before
- CLEventManager updates



Even More Detailed Flow w/ CLC and DB locks

DROOLS PDP

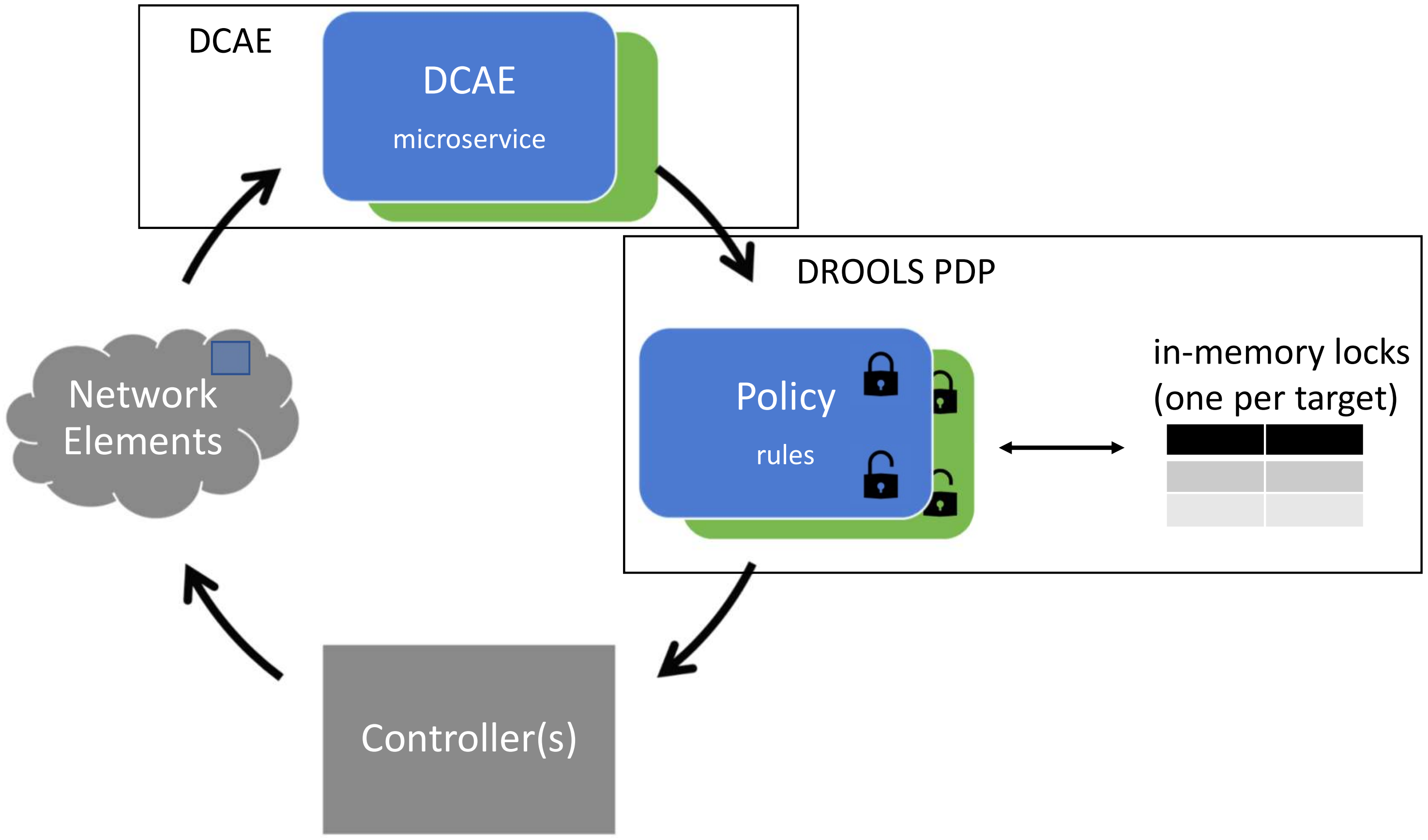
1. EVENT
   - ControlLoopEventManager created
2. EVENT.MANAGER
   - If new Onset
     - obtain ClosedLoopOperationManager
     - insert to memory
   - If Abated
     - Retract from memory
     - Call ControlLoopOperationManager.completeOperation()
     - Which then calls storeOperation() containing JPA code for DB write
3. EVENT.MANAGER.OPERATION.GUARD_NOT_BET_QUERIED
   - Determine action via operation.Policy.getRecipe
   - **send XACML request on new thread (obtain DB lock(s), read(s))**
4. GUARD.RESPONSE
5. EVENT.MANAGER.OPERATION.GUARD_DENIED
   - Release DB lock
6. EVENT.MANAGER.OPERATION.GUARD_PERMITTED
   - Get request to send: request = operation.startOperation($event)
7. Controller.RESPONSE
   - Calls ControlLoopOperationManager.completeOperation()
   - Which then calls storeOperation() containing JPA code for DB write
   - Release DB lock

XACML PDP

1. Request received,
2. Activate all applicable rules in parallel
3. For each applicable CLC rule
   a. request VariableReferences from PIPs
   b. PIPs
      - Obtain entry locks
      - Issue DB reads
   c. VariableReferences populated
   d. Functions applied
   e. Decision returned
4. Rule Decisions combined
5. XACML response sent

Comments on Target Locking

Target Locking (as currentlB implemented)



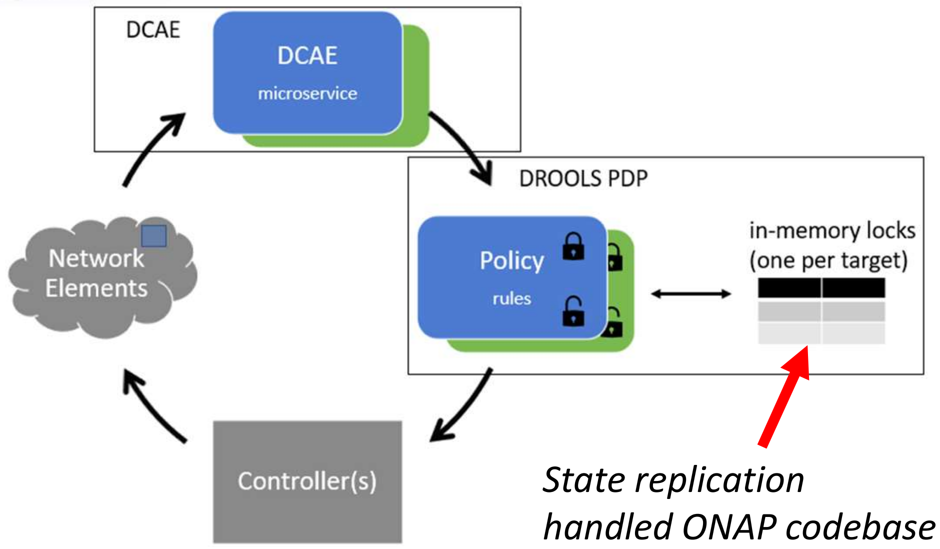
Target Locking vs. CLC

Supports only one, hardcoded, coordination pattern

```
• enable(G,B)   :=                              permit G
• disable(G,B) :=   if B open actions,          deny G
                    else                        permit G;
• block(G,B,t) :=   if B open actions   OR
                       now - B.closed < t,      deny G
                    else                        permit G;
• preempt(G,B) :=   if B has open actions,      cancel;
                                                permit G;
```

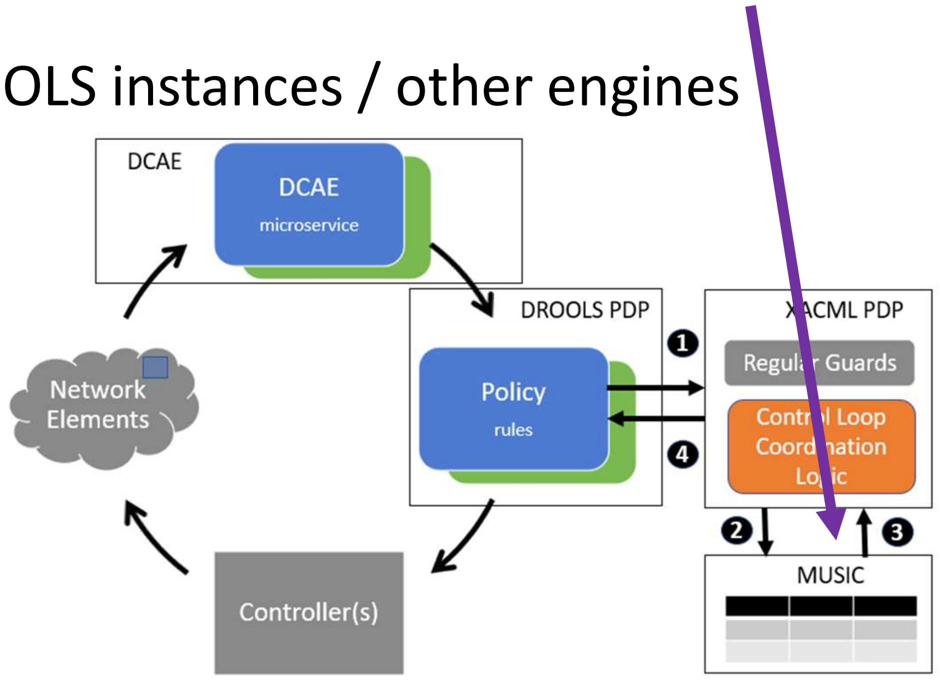
Target Locking vs. CLC

- Less flexibility
- More code complexity
- More work to handle multiple DROOLS instances / other engines



State replication handled ONAP codebase

State replication handled by MUSIC


DCAE
DCAE
microservice
DROOLS PDP
Policy
rules
XACML PDP
Regular Guards
Control Loop Coordination Logic
MUSIC
1
2
3
4
Network Elements
Controller(s)