An idea for access control enhancement on the EPICS

Accelerator Group
J-PARC, JAERI

April 2004

Why enhancement is required?

- Requirement from high-intensity proton machine: (Mega-Watt class machine)
  - There is a target value for effective dose in J-PARC. (0.25μSv/hour)
  - In our 50GeV synchrotron, effective dose when 100% loss will be 49μSv/pulse. Loss of a beam pulse causes waste of machine time for 200 hours.
  - We are not allowed to mistake on operation at all. We need anything to reduce possibility of operation error as low as possible.
  - So, access control enhancement by using exclusive operational right is required to achieve more safe operation.

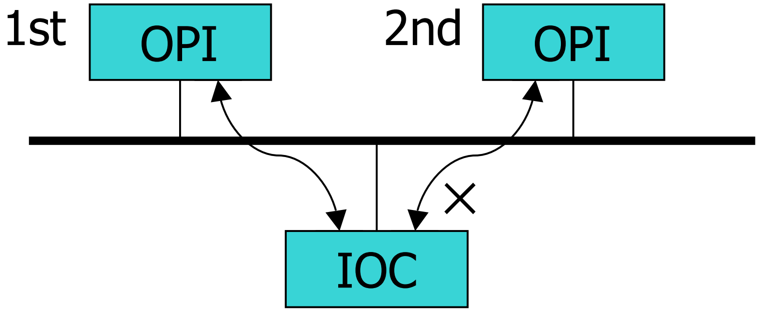
What is the exclusive operational right?

First come, first serve...

① Each critical records have a property of access control called “exclusive operational right.”

② A client connected to a record at first have an exclusive operational right (right to write) and can full access.

③ A client connected on and after second have no exclusive operational right and can not write access to a record. (but can read access)



Exclusive operational right

- Requirements:
  - An operation (write access) to a record must be permitted only for one operator at a time.
  - An operational right must be passed to another operator.
  - An operational right must be force released by supervisor if need.
  - Status of operational right should be shown for others.

A plan to implement

- Step by step strategy:
  - Phase 1
    Minimum modification to make full use of existing EPICS resources.
  - Phase 2
    Full spec implementation.

Phase 1:
A tentative implementation

- Outline:
  - Changes of operational right is notified by using CA_PROTO_ACCCESS_RIGHTS packet.
  - Modify the CA Server program a bit
  - No modification on CA protocol
  - No modification on AS files
- Merit
  - No compatibility problems
  - No need to modify existing clients
- Demerit
  - All of records on an IOC may take effect
  - No legal way to know who has the right

Phase 1 implementation: procedure to pass a right

- Points:
  - Pass the right to another client when right holding channel was disconnected. (Because existing clients could not take the right themselves.)
- Procedure:
  1. CAS (CA Server) choose a client within current connected channels to pass the right when right holder client was disconnected.
  2. CAS send an ACCESS_RIGHTS packet to the client.

Phase 1 implementation: trick to force pass a right

- Points:
  - Assign an user who has privilege to force release operational rights.
  - Force release current operational right when channel is connected from privileged user. (Because existing clients could not return the right themselves.)
- Procedure:
  1. CAS send an ACCESS_RIGHTS packet to right holder client when the right was released.
  2. CAS choose a client within current connected channels to pass the right.
  3. CAS send an ACCESS_RIGHTS packet to the client.

Phase 2:
A proposal implementation

- Outline:
  - Add a new command to CA protocol
    - new CA_PROTO_RIGHT_CONTROL
    - existing CA_PROTO_ACCESS_RIGHTS also used
  - Add new keywords in AS file
    - to specify record need or not need operational exclusion
    - to specify privileged user
- Merit
  - realize all of requirements
- Demerit
  - right control dialogue are desired on GUI's
  - too hard to update all of existing clients

Phase 2 implementation:
new CA_PROTO_RIGHT_CONTROL packet

cmd (28)	size (variable)
type (variable)	count (0)
cid (variable)	
sid (variable)	
(“username\0hostname\0”)	

the ‘type’ field is abused for sub-command

query(0)	query current status of right. replay packet have payload
request(1)	request for a right
return(2)	return/pass a right
deny(3)	deny request for a right
force(4)	request for a right (force)

Phase 2 implementation: procedure to pass a right

① A following client send a RIGHT_CONTROL/request packet to CAS.

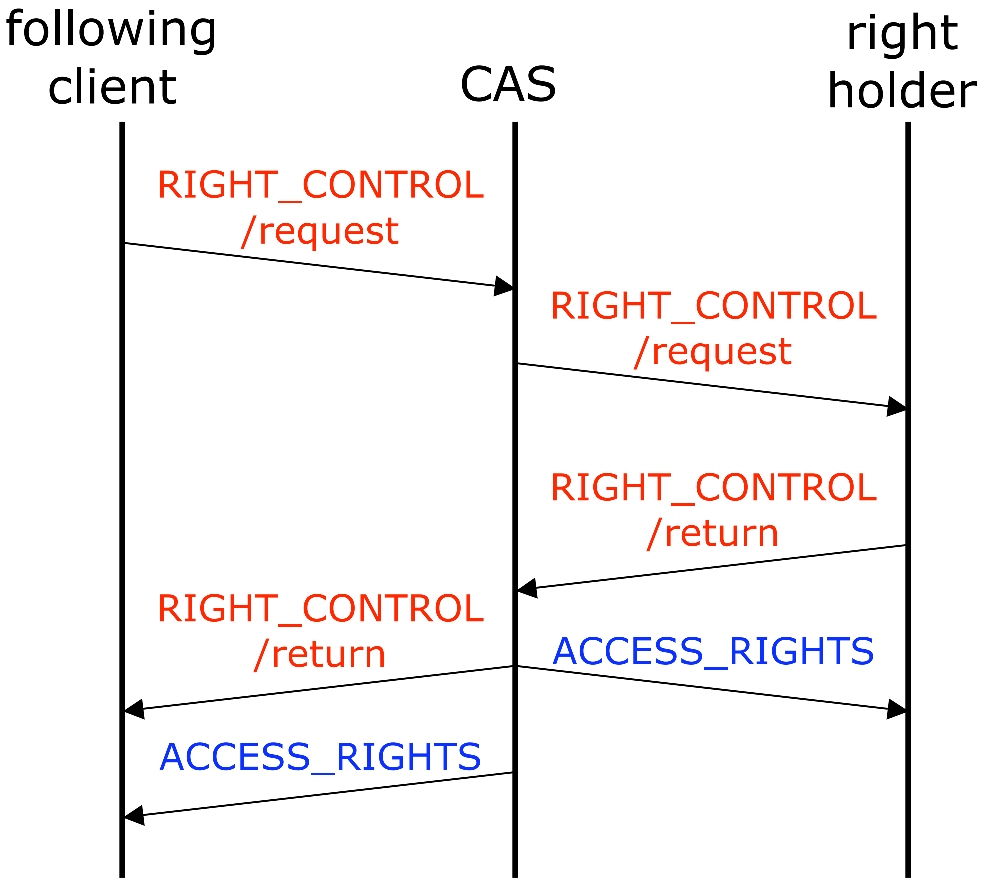
② CAS relay the packet to a client who has operational right. Note that Channel Access has no way of inter-client communication, so CAS should relay it.

③ The right holder client returns a RIGHT_CONTROL/return or /deny reply packet according to decision of permit or deny.
Note that this decision possibly done by operator.

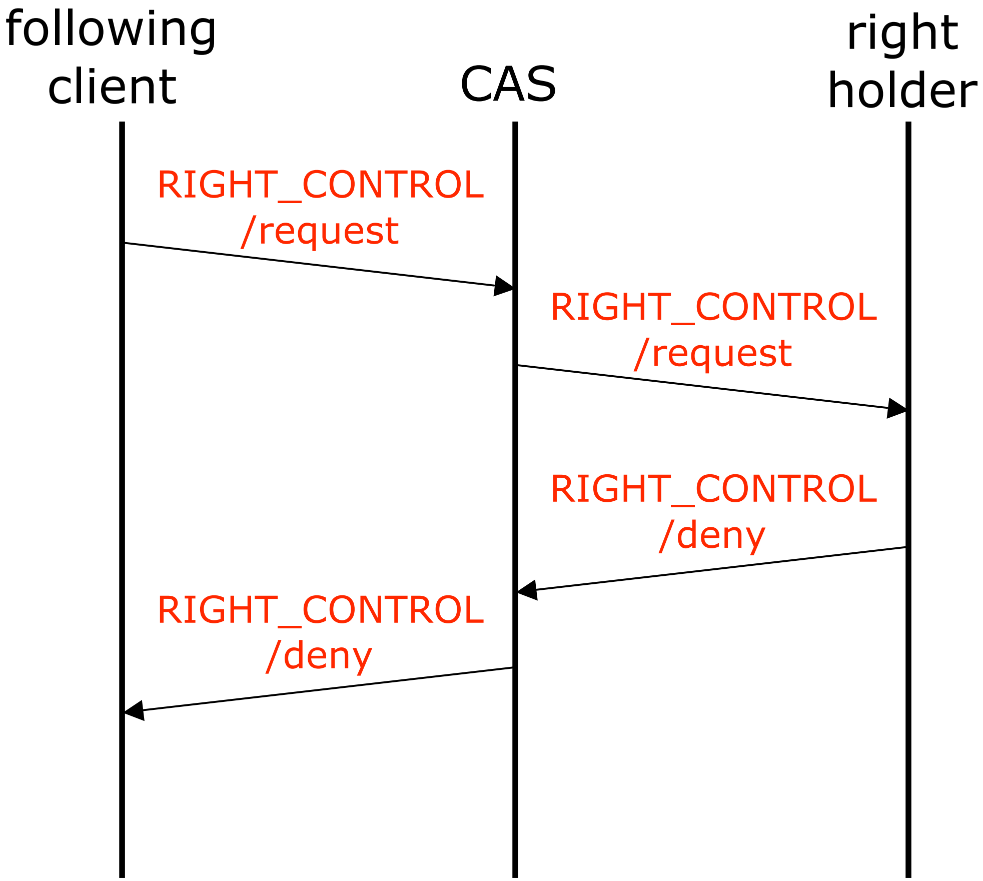
④ After right returned, CAS send an ACCESS_RIGHT packet to the previous client.

⑤ CAS relay the RIGHT_CONTROL packet to following client. And also send an ACCESS_RIGHT packet when operational right is passed.

Phase 2 implementation: procedure to pass a right (accept)



Phase 2 implementation: procedure to pass a right (deny)



Phase 2 implementation: procedure to force release a right

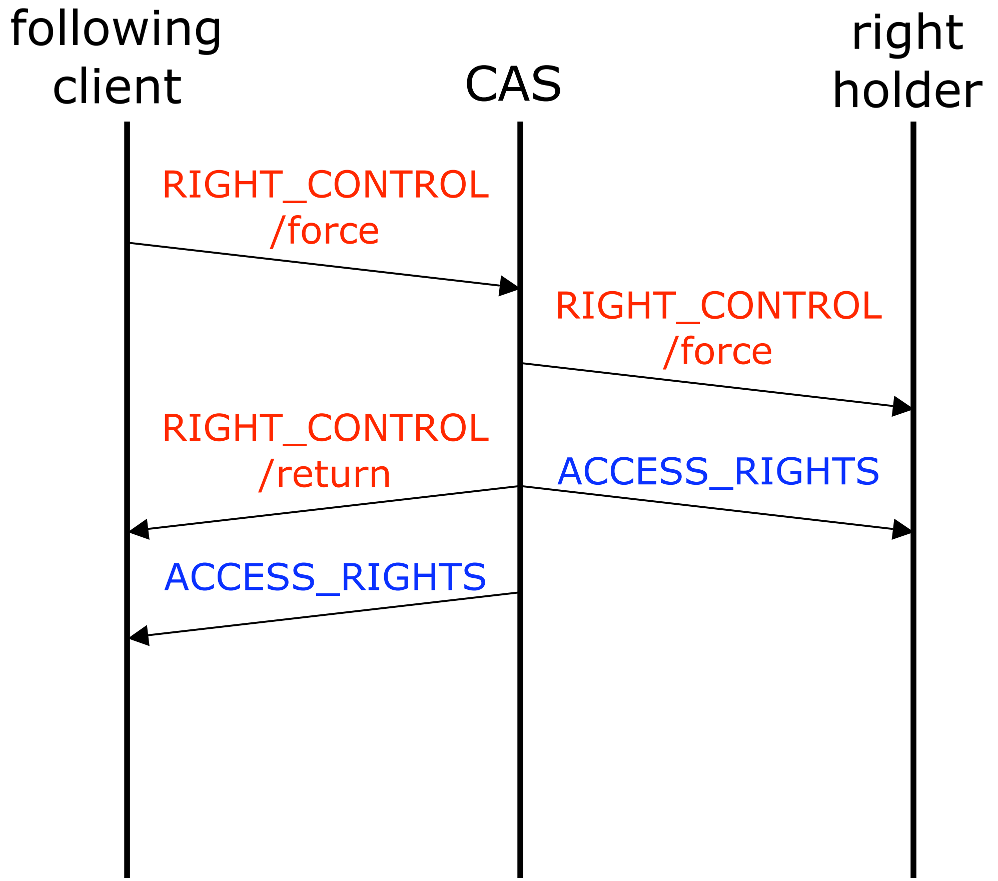
① A following client send a RIGHT_CONTROL/force packet to CAS.
Note that following client must have privilege.

② CAS relay the packet to a right holder client.
No reply from the right holder client at this point.

③ CAS send an ACCESS_RIGHT packet to the right holder client.

④ CAS send a RIGHT_CONTROL/return packet and an ACCESS_RIGHT packet to following client.

Phase 2 implementation: procedure to force release a right

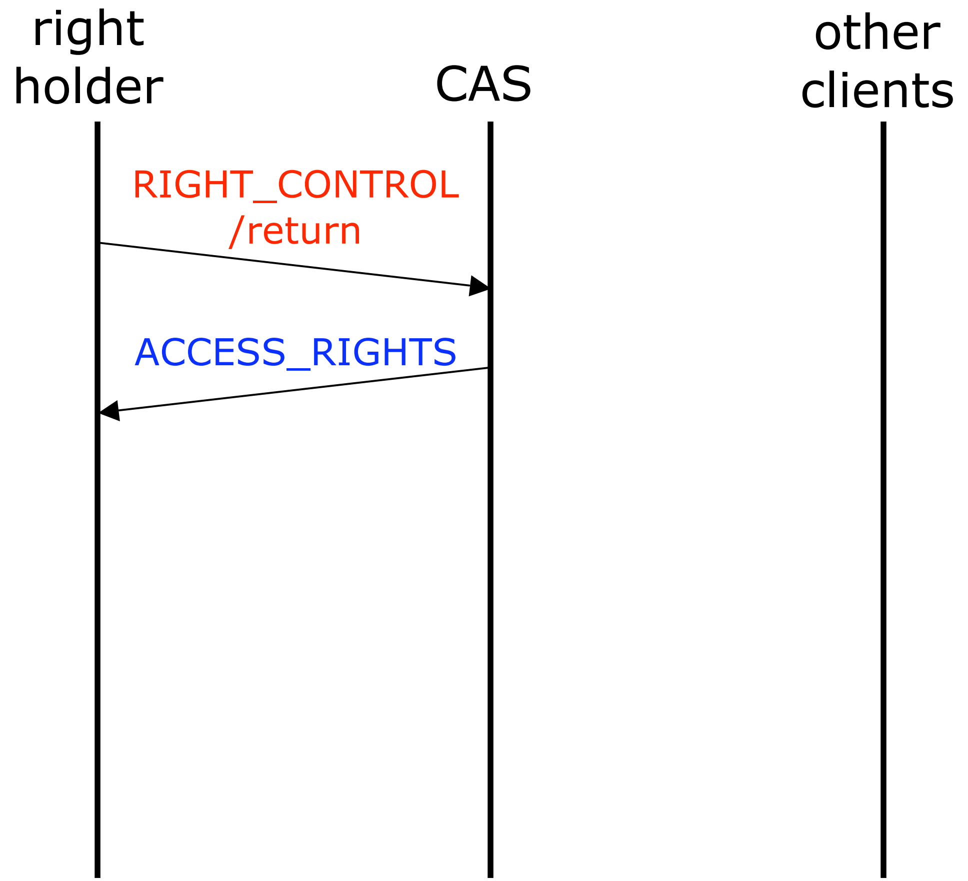


Phase 2 implementation: procedure to return a right

① Right holder client send a RIGHT_CONTROL/return packet to CAS.

② CAS send an ACCESS_RIGHT packet to client.

Phase 2 implementation: procedure to return a right

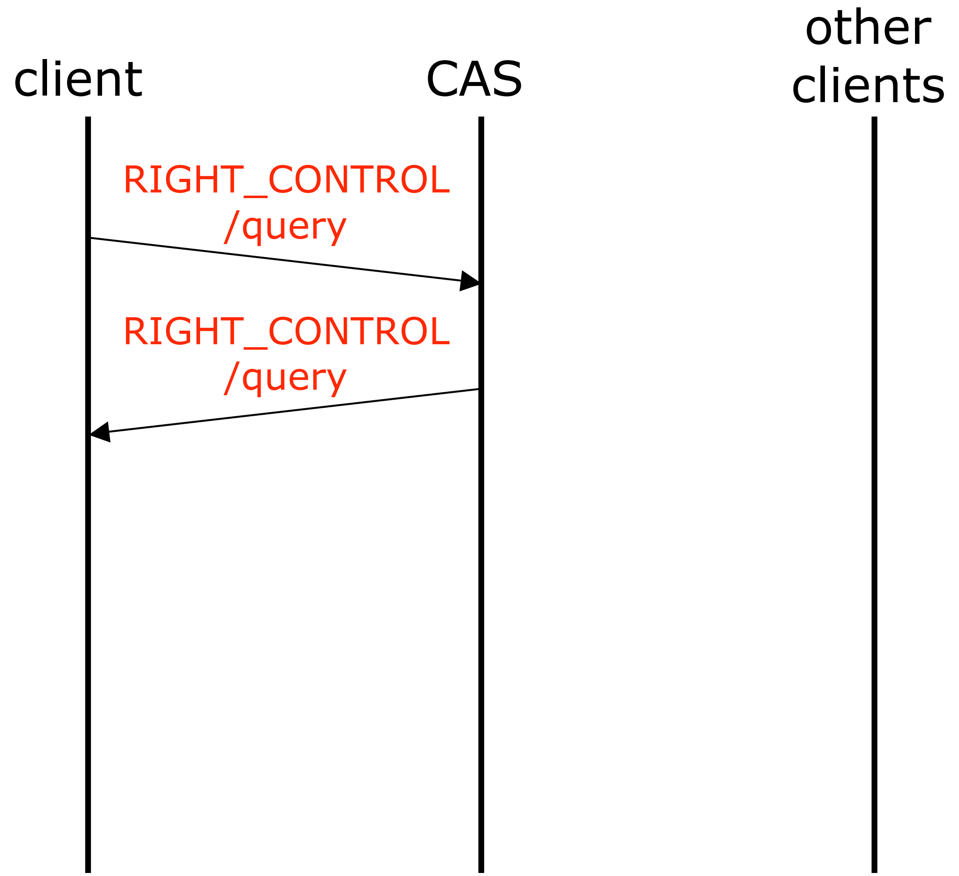


Phase 2 implementation:
procedure to query status of a right

① A client send a RIGHT_CONTROL/query packet to CAS.

② CAS returns a RIGHT_CONTROL/query packet with right holder information.

Phase 2 implementation: procedure to query status of a right


client
CAS
other clients
RIGHT_CONTROL /query
RIGHT_CONTROL /query