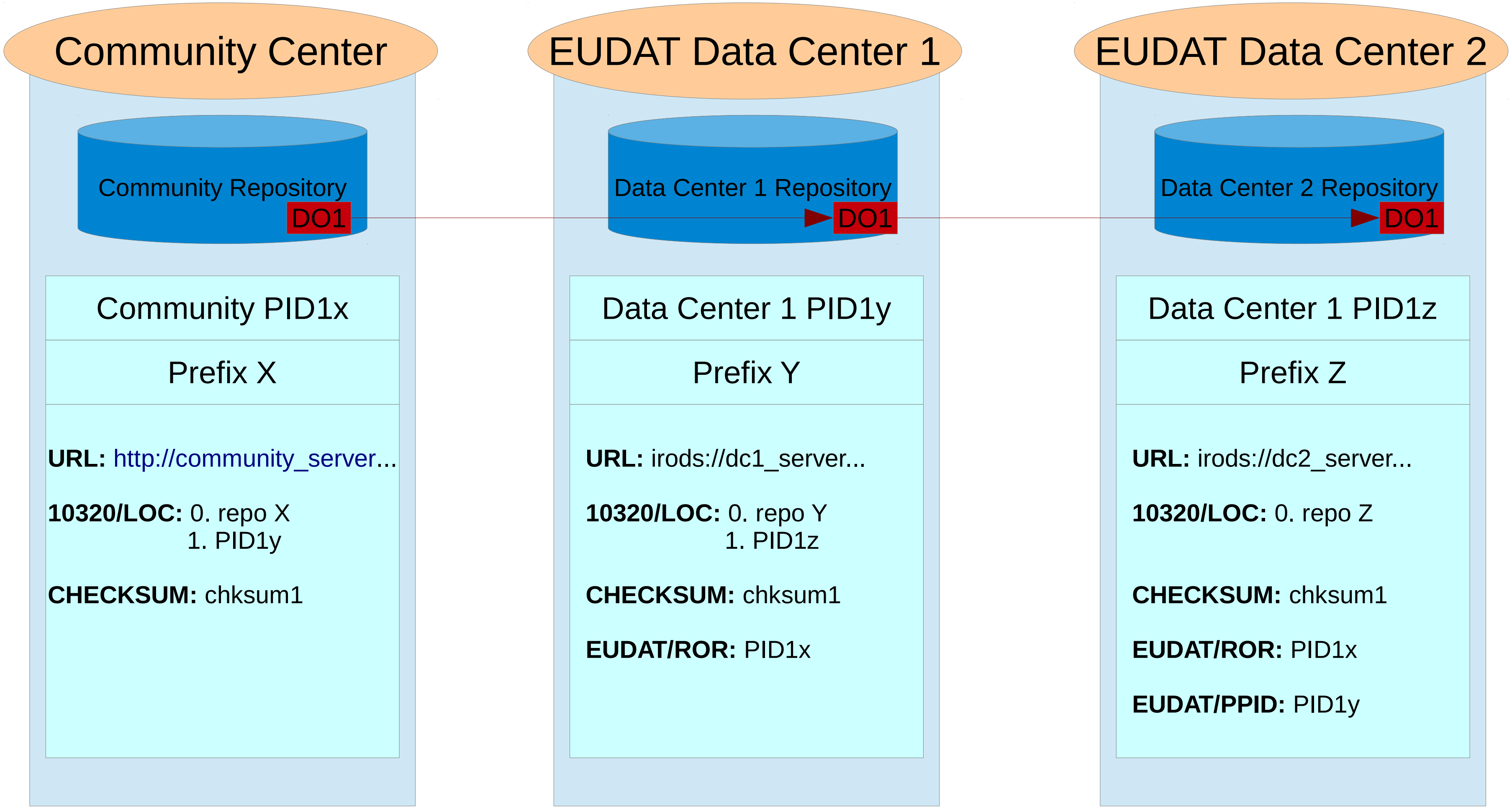
PID Interlinking Scheme



Community Center

Community Repository

DO1

Community PID1x

Prefix X

URL: http://community_server...

10320/LOC: 0. repo X
1. PID1y

CHECKSUM: chksum1

EUDAT Data Center 1

Data Center 1 Repository

DO1

Data Center 1 PID1y

Prefix Y

URL: irods://dc1_server...

10320/LOC: 0. repo Y
1. PID1z

CHECKSUM: chksum1

EUDAT/ROR: PID1x

EUDAT Data Center 2

Data Center 2 Repository

DO1

Data Center 1 PID1z

Prefix Z

URL: irods://dc2_server...

10320/LOC: 0. repo Z

CHECKSUM: chksum1

EUDAT/ROR: PID1x

EUDAT/PPID: PID1y

PID Interlinking Implementation

```
eudat.re:
processPIDCommandFile(*cmdPath):
...
foreach (*item_LIST in *list) {
        if      ( *counter == 0 ) { *pidAction   = *item_LIST ; }
        else if ( *counter == 1 ) { *parent      = *item_LIST ; }
        else if ( *counter == 2 ) { *destination = *item_LIST ; }
        else if ( *counter == 3 ) { *ror         = *item_LIST ; }
        *counter = *counter + 1;
    }
...
 if(*pidAction == "create") {
            if(*ror=="None") {
                *ror = *parent;
                *parent = "None";
                logInfo("ror = *ror, parent = *parent");
            }
...
```

PID Interlinking Implementation

eudat.re:

```
createPID(*parent_pid, *path, *ror, *newPID, *iCATCache):
…
 # add RoR to PID record if there is one defined

    if((*ror == "None") && (*parent_pid != "None")) {
        *ror = *parent_pid;
        *parent_pid = "None";
    }

    if(*ror != "None") {
        # add RoR to PID record
        if(*epicDebug > 1) {
            logDebug("epicclient.py *credStoreType *credStorePath modify *newPID ROR *ror");
        }

        *list0 = split(*ror, "/");
        *first = elem(*list0,0);
        if(*first=="http:") {
            msiExecCmd("epicclient.py","*credStoreType *credStorePath modify *newPID EUDAT/ROR *ror",
"null", "null", "null", *out4);
            msiGetStdoutInExecCmdOut(*out4, *response4);
            logDebug("modify handle response = *response4")
        }
        else {
            msiExecCmd("epicclient.py","*credStoreType *credStorePath modify *newPID EUDAT/ROR
*epicApi*ror", "null", "null", "null", *out2);
            msiGetStdoutInExecCmdOut(*out2, *response2);
            logDebug("modify handle response = *response2");
        }
    }
...
```

PID Interlinking Implementation

eudat.re:

```
createPID(*parent_pid, *path, *ror, *newPID, *iCATCache):
…
if(*parent_pid != "None") {
    # add PPID to PID record
        if(*epicDebug > 1) {
            logDebug("epicclient.py *credStoreType *credStorePath modify *newPID EUDAT/PPID
*parent_pid");
        }

        *list0 = split(*parent_pid, "/");
        *first = elem(*list0,0);
        if(*first=="http:") {
            msiExecCmd("epicclient.py","*credStoreType *credStorePath modify *newPID EUDAT/PPID
*parent_pid", "null", "null", "null", *out44);
            msiGetStdoutInExecCmdOut(*out44, *response44);
            logDebug("modify handle response = *response44")
        }
        else {
            msiExecCmd("epicclient.py","*credStoreType *credStorePath modify *newPID EUDAT/PPID
*epicApi*parent_pid", "null", "null", "null", *out22);
            msiGetStdoutInExecCmdOut(*out22, *response22);
            logDebug("modify handle response = *response22");
        }
    }
...
```

PID Interlinking Implementation

```
eudat.re:

searchPIDROR(*ror, *existing_pid) {
    logInfo("search pid for *ror");
    getEpicApiParameters(*credStoreType, *credStorePath, *epicApi,
*serverID, *epicDebug);
    #check if PID already exists
    if(*epicDebug > 1) {
        logInfo("epicclient.py *credStoreType *credStorePath search
EUDAT/ROR *ror");
    }
    msiExecCmd("epicclient.py", "*credStoreType *credStorePath search
EUDAT/ROR *ror", "null", "null", "null", *out);
    msiGetStdoutInExecCmdOut(*out, *existing_pid);
}
```

PID Interlinking Implementation

```
eudat.re:

searchPIDROR(*ror, *existing_pid) {
    logInfo("search pid for *ror");
    getEpicApiParameters(*credStoreType, *credStorePath, *epicApi,
*serverID, *epicDebug);
    #check if PID already exists
    if(*epicDebug > 1) {
        logInfo("epicclient.py *credStoreType *credStorePath search
EUDAT/ROR *ror");
    }
    msiExecCmd("epicclient.py", "*credStoreType *credStorePath search
EUDAT/ROR *ror", "null", "null", "null", *out);
    msiGetStdoutInExecCmdOut(*out, *existing_pid);
}
```