iRODS – Advanced user training

FEDERATIONS AND RULES – S4R WORKSHOP

Christine Staiger

Agenda

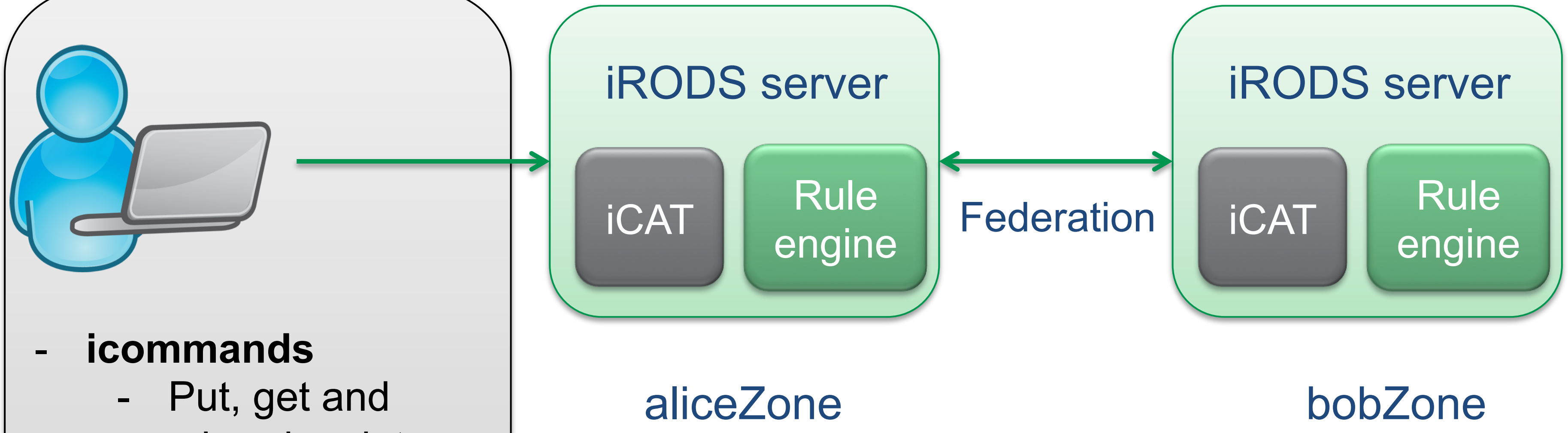
10.00-10.30 Recap of icommands

10.30-12.00 iRODS Federations and data replication

12.00-13.00 Lunch

13.00-17.00 Rules, rules rules

Training Setup



- **icommands**
  - Put, get and synchronise data
  - Execute rules
- Folder 'exampleRules'

User Interface machine

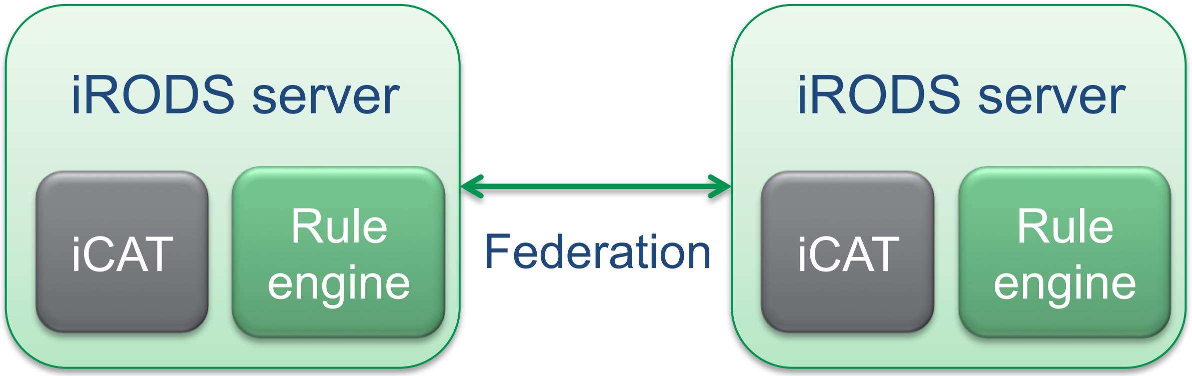
Login User Interface

ssh <uname>@<server>
Password:

iRODS aliceZone

<uname>
Password:

iRODS Federations



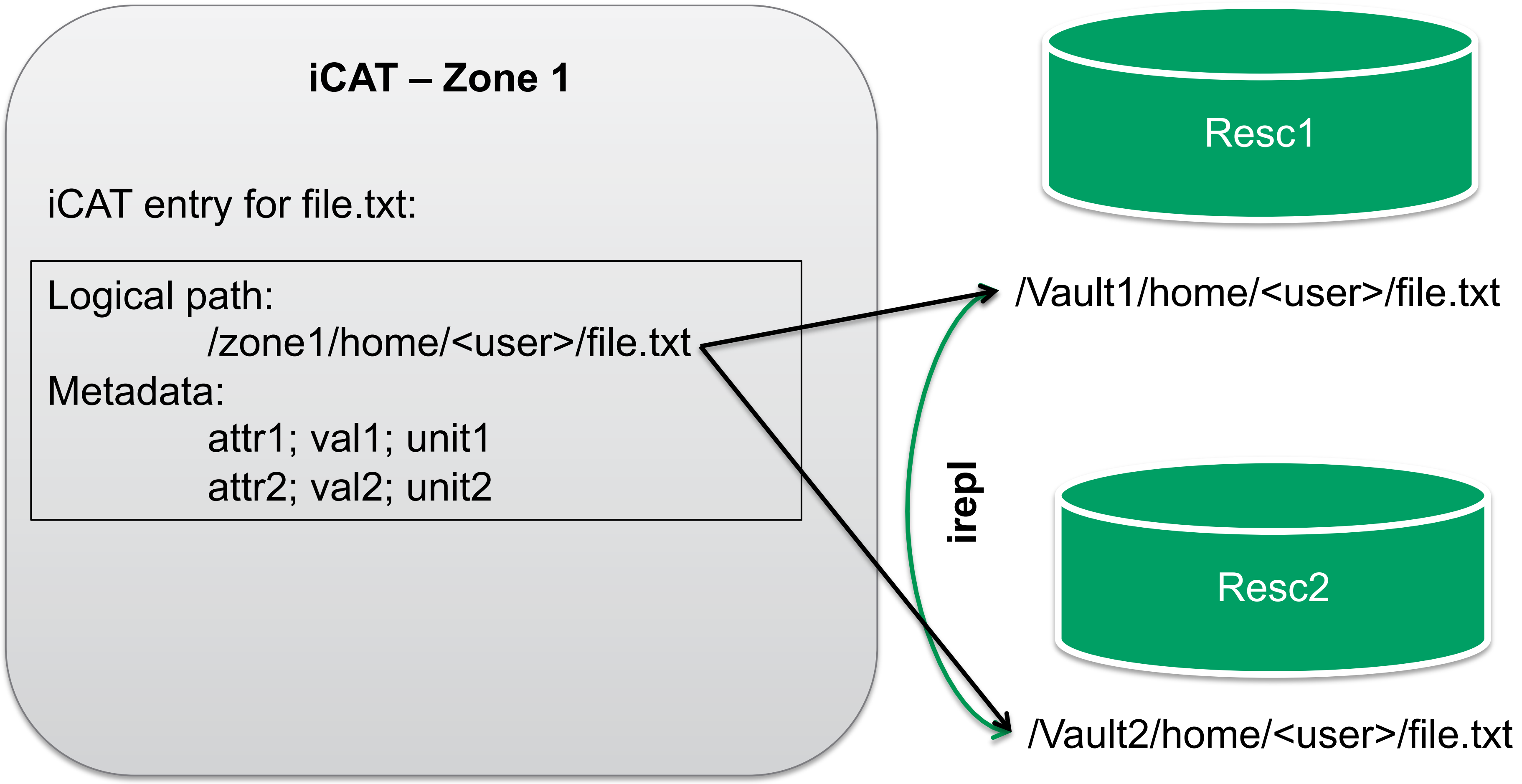
- Two independent iRODS zones, own rule engine and different rulebases
- Federation on system level
- iRODS admins give access to certain users

User

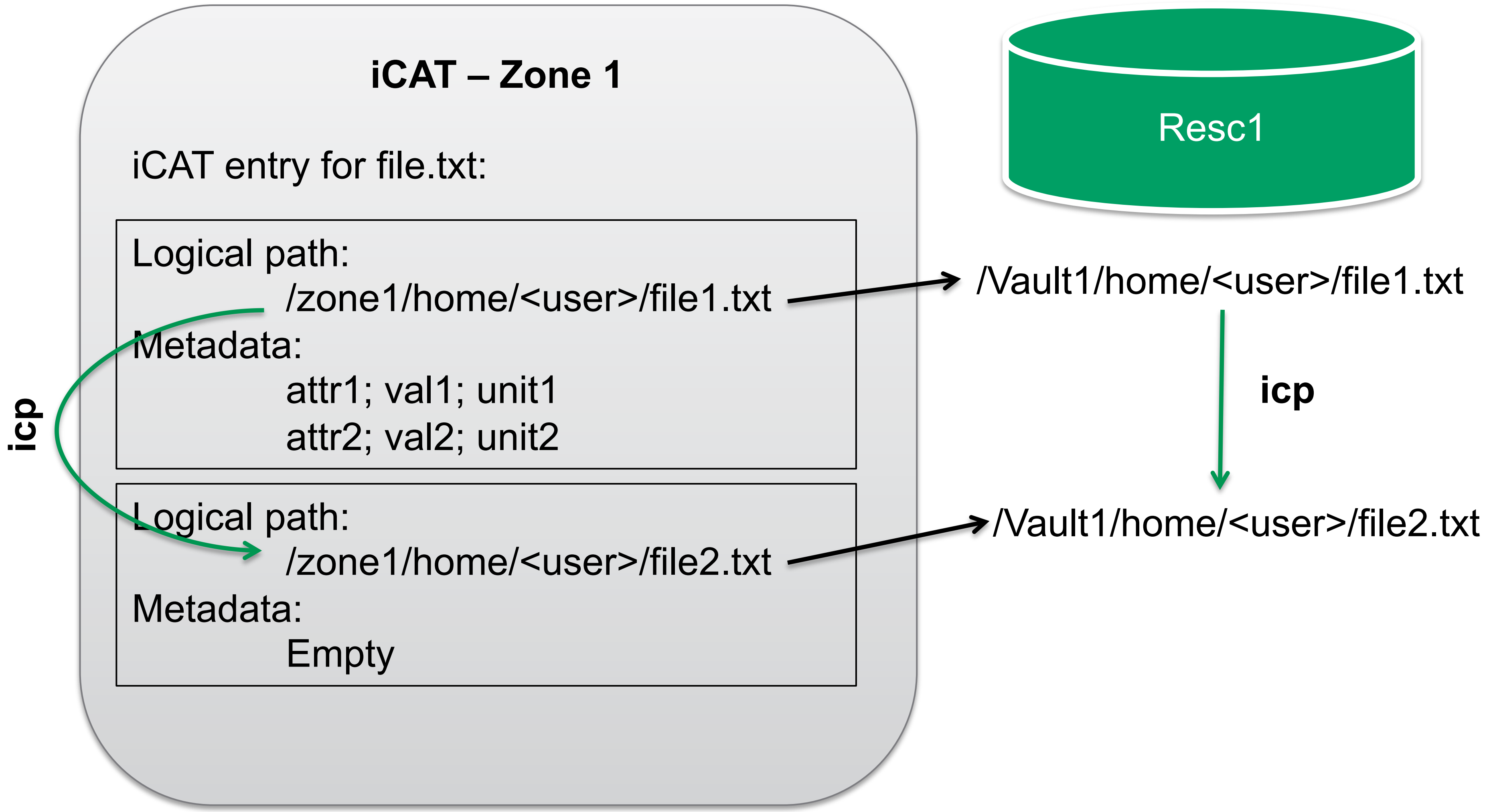
- Authentication with home iRODS zone
- If acknowledges user: Access to federated zone
  /otherIRODSzone/home/user#homeIRODSzone

Data – metadata relations with imv, icp and irepl

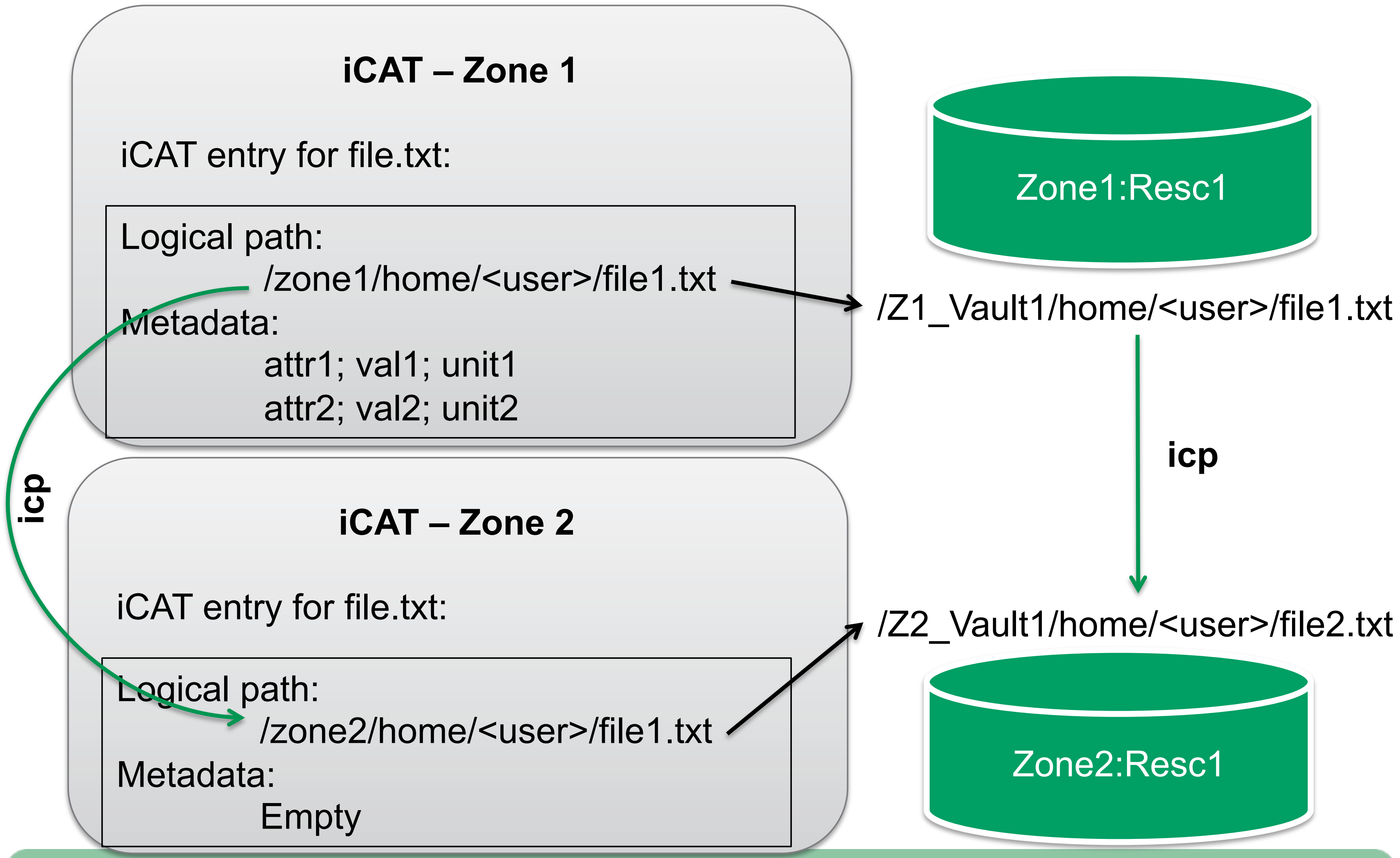
irepl


iCAT – Zone 1
iCAT entry for file.txt:
Logical path:
/zone1/home/<user>/file.txt
Metadata:
attr1; val1; unit1
attr2; val2; unit2
Resc1
/Vault1/home/<user>/file.txt
irepl
Resc2
/Vault2/home/<user>/file.txt

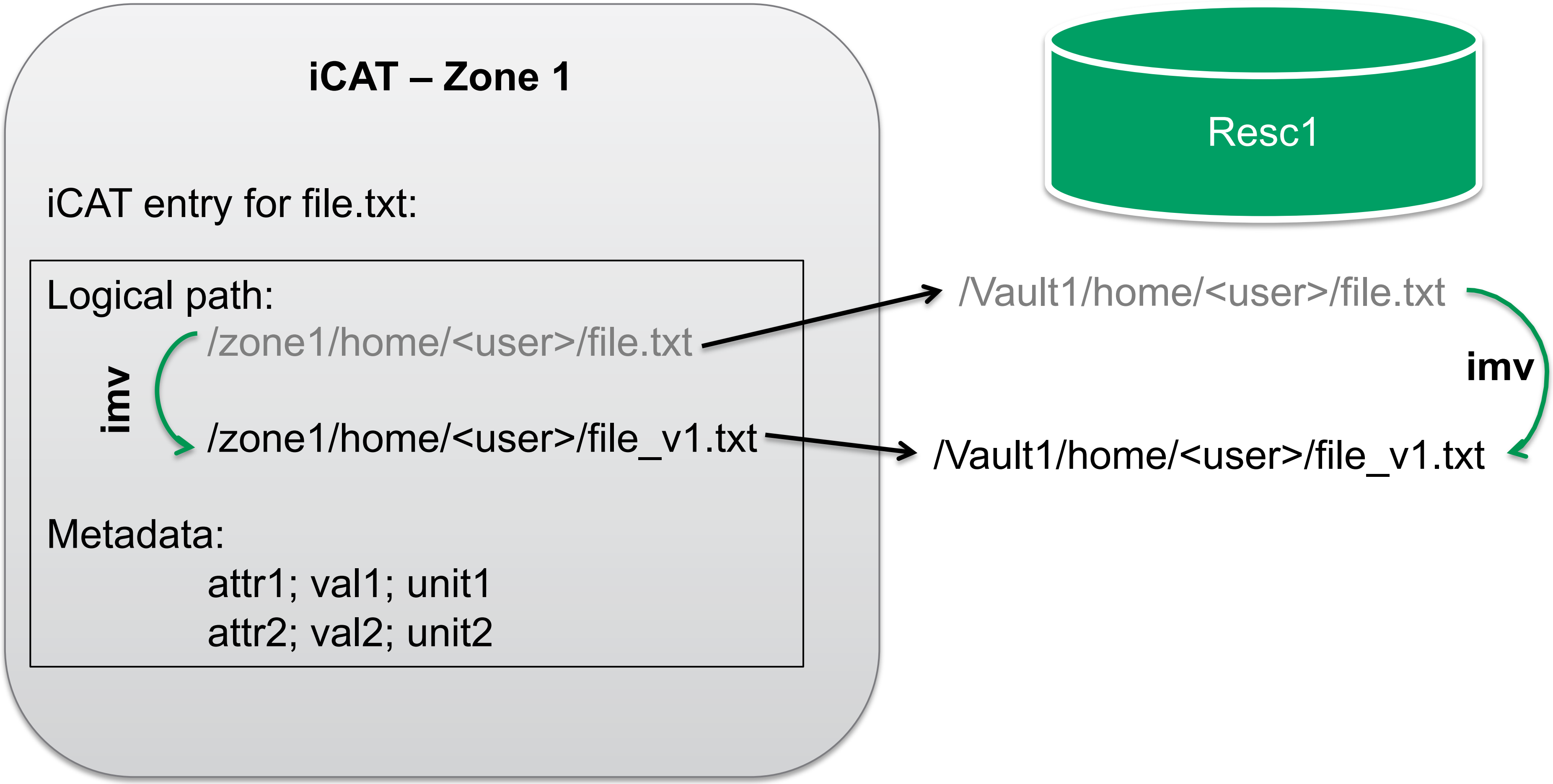
icp – in one zone



icp – across zones



imv

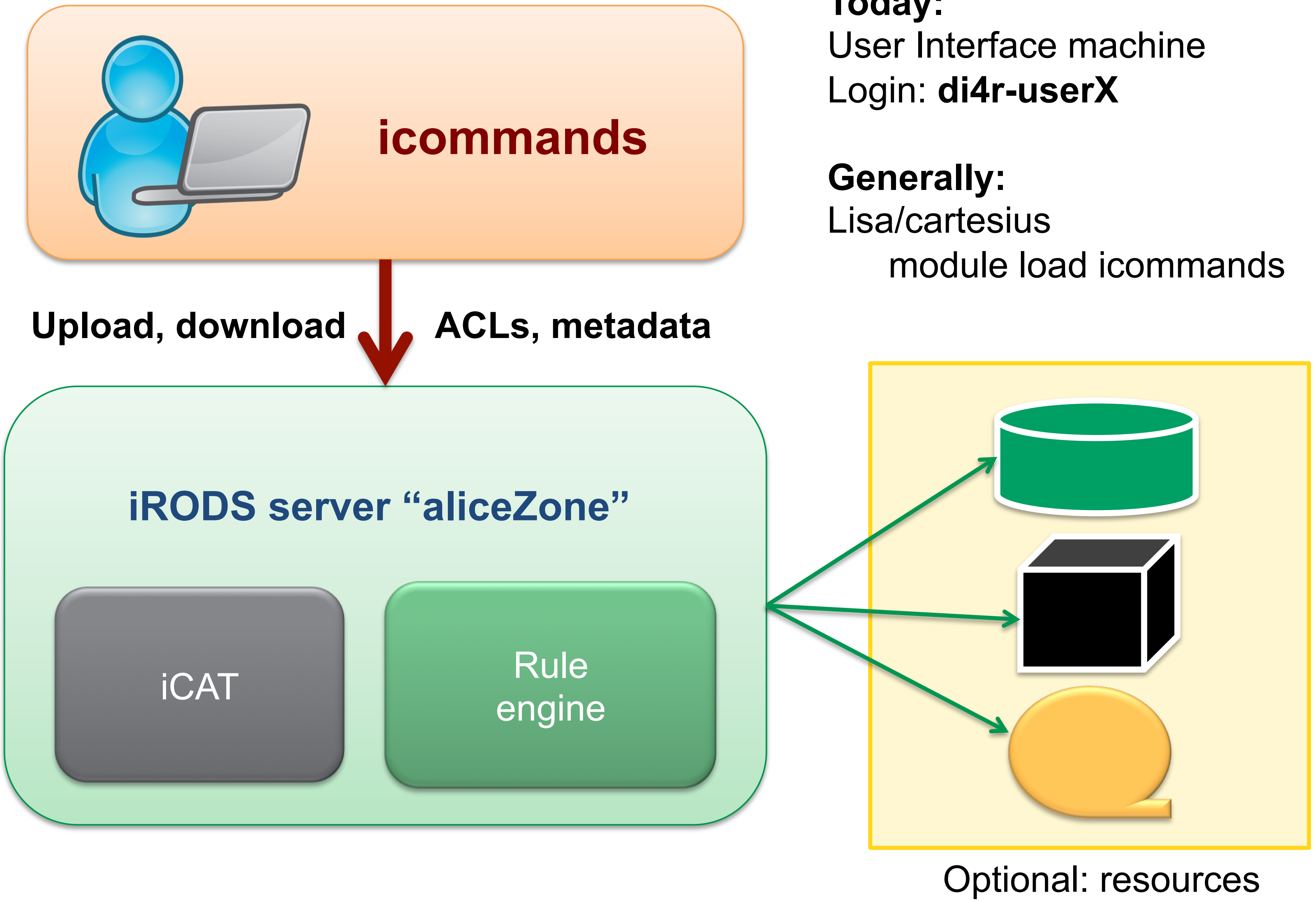

iCAT – Zone 1
iCAT entry for file.txt:
Logical path:
/zone1/home/<user>/file.txt
imv
/zone1/home/<user>/file_v1.txt
Metadata:
attr1; val1; unit1
attr2; val2; unit2
Resc1
/Vault1/home/<user>/file.txt
imv
/Vault1/home/<user>/file_v1.txt

Not possible to do an imv across Zones:

Metadata entry in Zone1 while data resides on resource in Zone 2

Today:
User Interface machine
Login: **di4r-userX**

Generally:
Lisa/cartesius
module load icommands



Rules and micro services

iRODS micro services

- Define actions on data, resources and users → atomic
- C++ functions, calling external libraries
- Used and combined in workflows and policies → iRODS rules

- Predefined microservices
  http://docs.irods.org/4.1.10/doxygen
- Example: msiCollRsync → synchronises two iRODS collections from different zones

- Own micro services:
  - Written in C++
  - Need to be installed on the iRODS server → root or iRODS service account rights
  - Example: Automatic metadata extraction from HDF5 files

iRODS rules

- iRODS rule engine → built-in interpreter for own language
- Automatise data management tasks
- Standard set of pre-implemented rules constitutes default data policies

- Trigger execution of rules by
  - irule → User
  - Delayed or scheduled execution → User & iRODS admin
  - Actions and policy enforcement points extending and overlaying the default rule base → sysadmin

```
HelloWorld{
        writeLine("stdout", "Hello *name!");
}
INPUT *name="World"
OUTPUT ruleExecOut, *name
```

iRODS standard data policies

- Event hooks are triggered by actions
  - E.g. put data (client interaction - iput)
  - acPostProcForPut - Rule for post processing the put operation.

```
acPostProcForPut {msiSysChksumDataObj;
          msiSysReplDataObj("demoResc","all"); }
```

- Policy enforcement points (PEPs) are executed by the rule engine

```
pep_api_data_obj_put_post(
    *COMM, *DATAOBJINP, *BUFFER, *PORTAL_OPR_OUT)
    { acPostProcForPut; }
```

Extending the standard core.re

- Predefined core.re and also pretty empty in standard setup
  - Placeholder for all event hooks and PEPs
  - Placeholder for own general data management rules

- Place your (carefully tested) rules directly into core.re

  → bad idea

- Write an own policy.re and configure server

  "re_rulebase_set":[{"filename":”policy”}, {"filename":"core”}]

  → policy.re and core.re build the rule set for this iRODS instance

  → Order matters

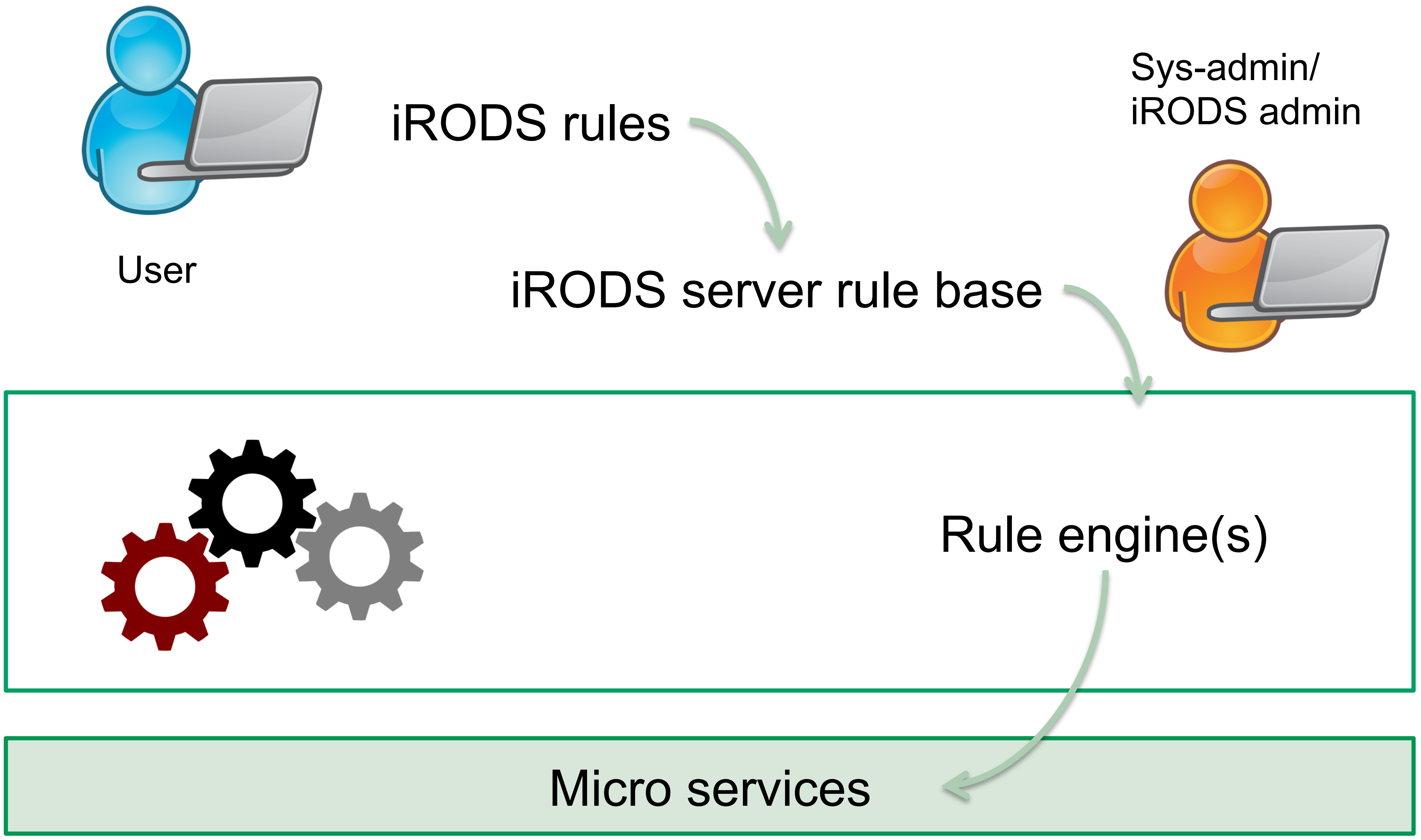
Rules: Order matters

- No namespaces!
- First rule that matches (name and variables) will be executed
- Event hooks and PEPs follow the syntax of rules

Workflow for developing policies/rules

- Write a local rule as iRODS user → irule <file>
  - → Debugging
- Put rule on top of all rules in the configured rule set
  - → Does it still work?
  - → Which rules does it inhibit from being executed
- Bit by bit find the right spot for the rule in the rule base

The Hierarchy



Write your own data archiving policy/rule