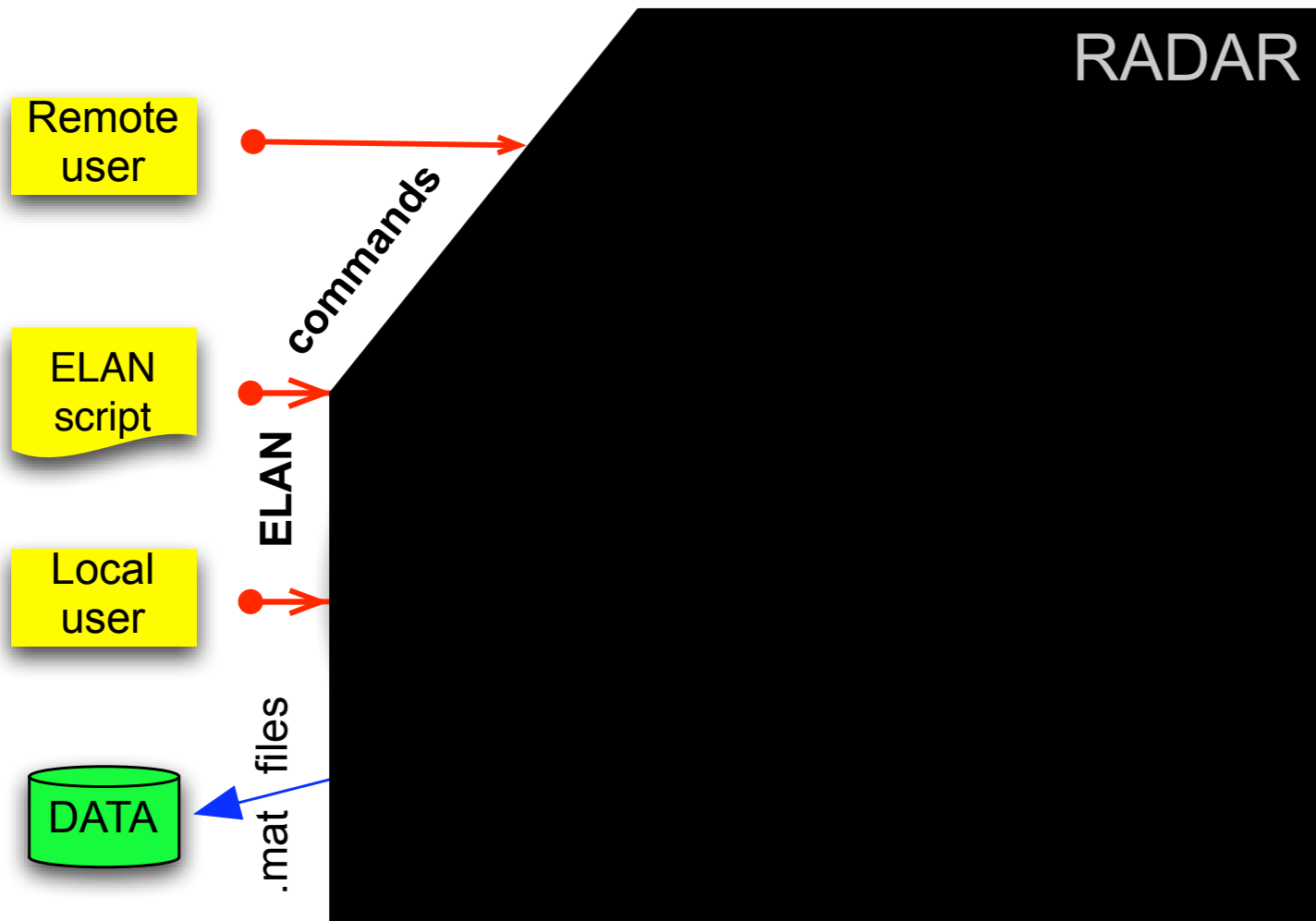


16. 15 The EROS user interface (JM):

- EROS internal structure
- how EROS communicates with drivers
- how the user communicates with EROS
- the EROS simulator

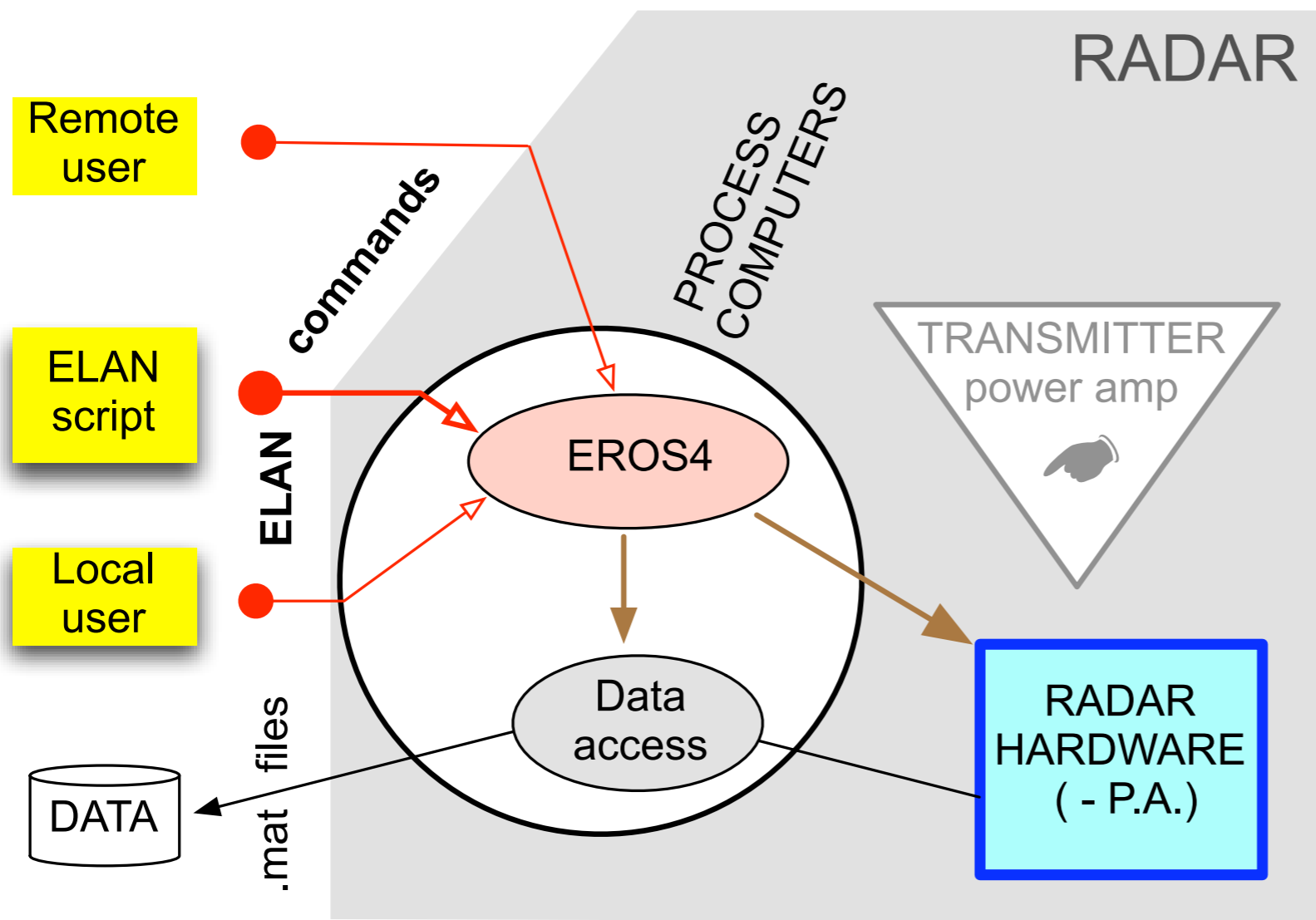
17.30 Hamburger party at the Space Campus

Operating an EISCAT radar



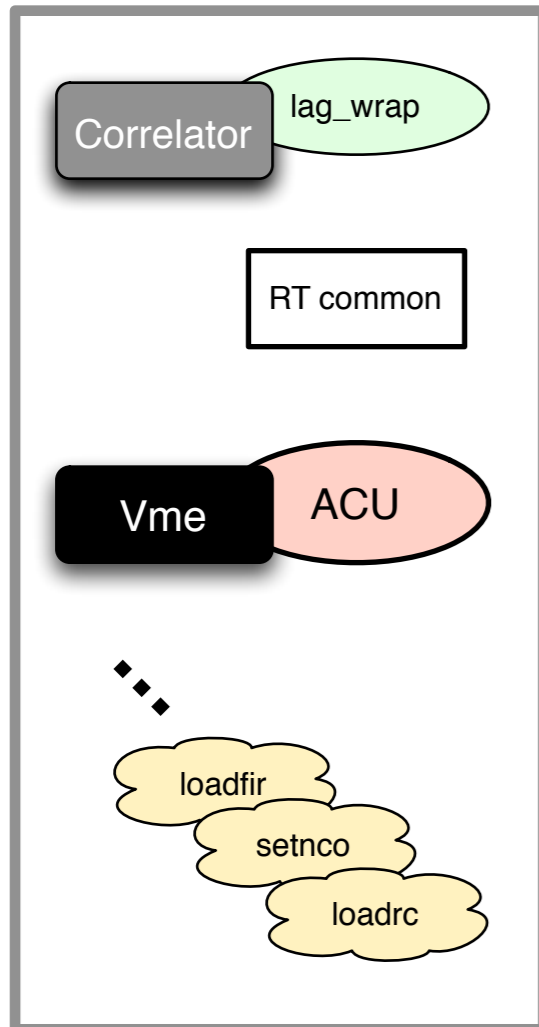
EISCAT radar operation = **ELAN commands in, raw data out.**
(The single required ELAN command to perform a full-fledged EISCAT experiment can often be copy-pasted from the EISCAT schedule.)

EROS4 – the middle man

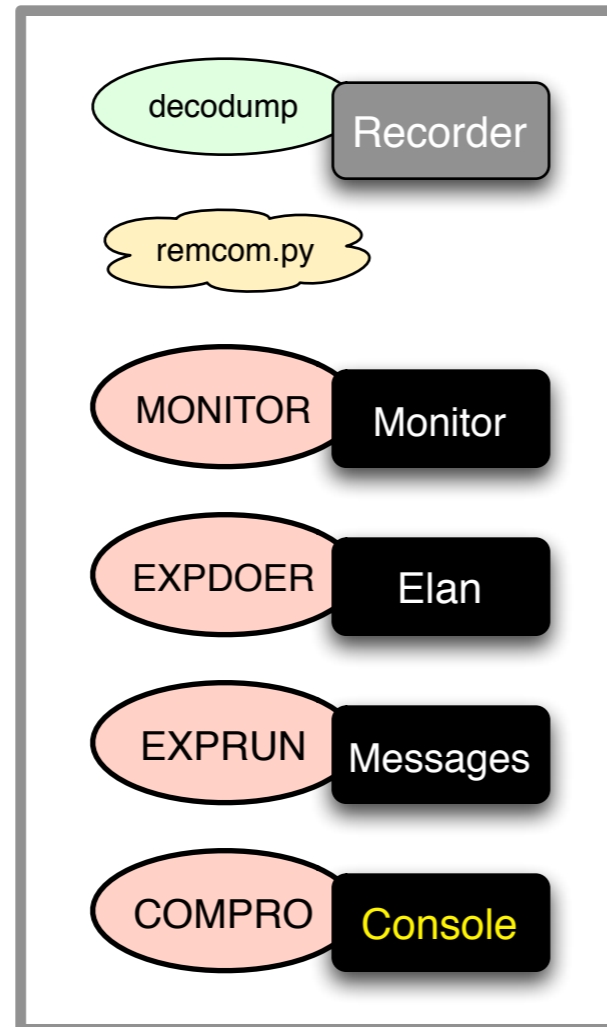


EROS4 processes and associates (Kiruna site)

Rec VME computer k5011

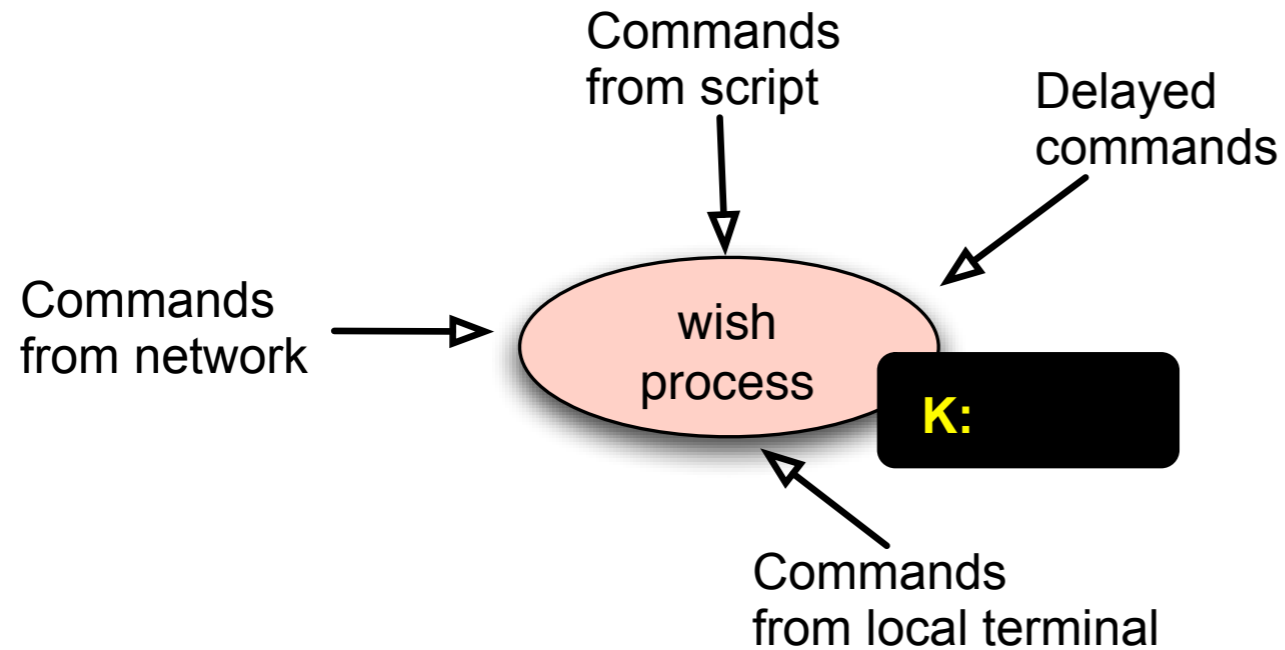


Server k2501



- eros (red oval) Always active
- data acc (green oval) Long-lived
- unix (yellow cloud) Short-lived

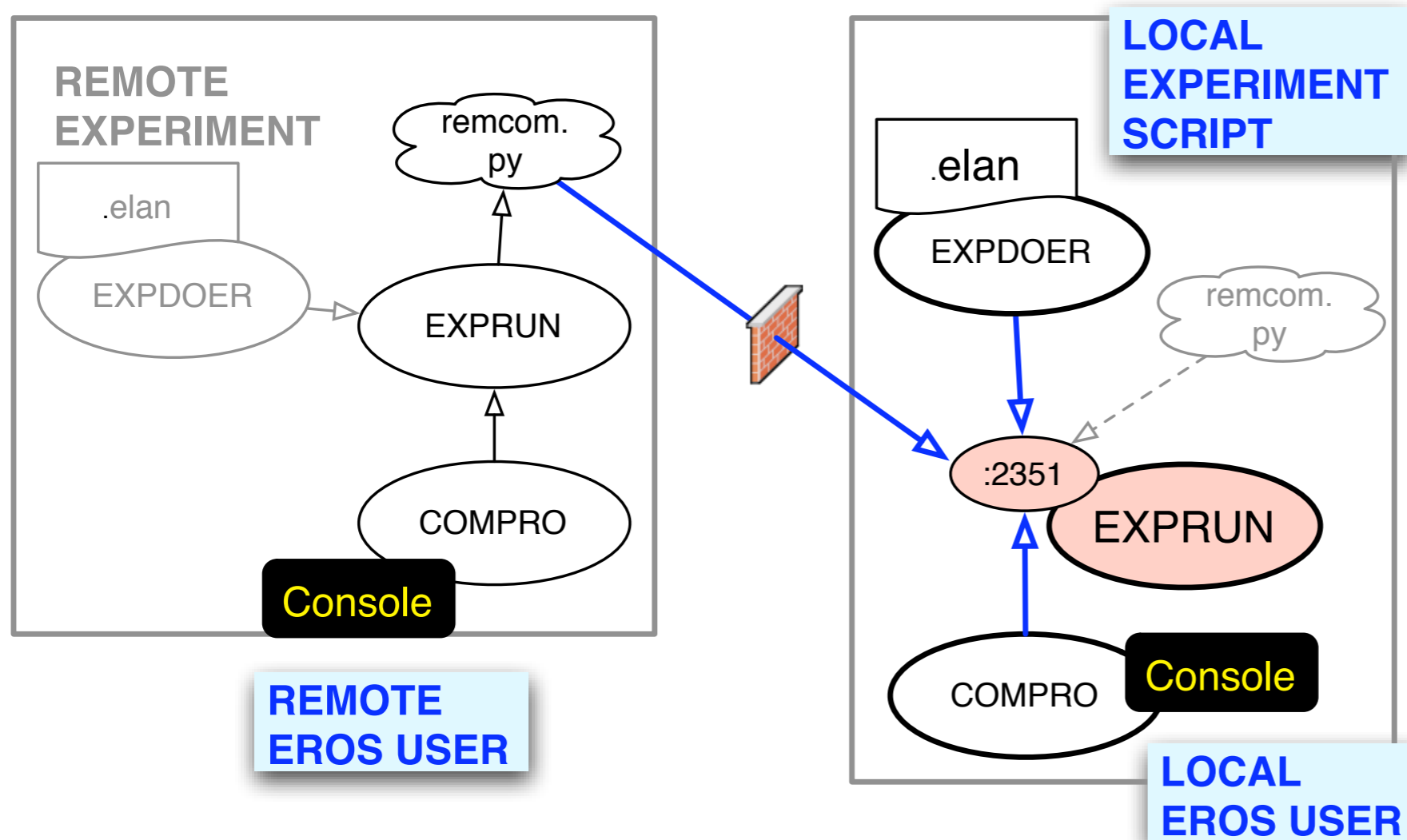
Wish command sources



A wish process has a single thread only, but can still serve asynchronous commands from several sources via its built-in event loop.

- ▶ **COMPRO**-wise receives and parameter-checks local interactive commands and sends them over to EXPRUN.
- ▶ **EXPRUN**-wise is the central engine. It receives all user-level commands and organizes the required actions, mostly by giving commands to ACU, EXPDOER and MONITOR. It interacts directly with the decodump. EXPRUN maintains the radar state information – all the printXXX commands are served directly by EXPRUN. EXPRUN maintains the EROS4 system log, and displays in the "Messages" window error messages and user messages. EXPRUN does all UHF pointing geometry computations.
- ▶ **ACU**-wise executes in the receiver VME crate computer. It receives commands from EXPRUN to invoke various UNIX-level routines (loadrc) to access the hardware over the VME-bus. In KST, ACU also accesses the antenna (pointxxx), and takes care of celestial tracking.
- ▶ **EXPDOER**-wise runs an ELAN file simply by sourcing it. EXPDOER itself executes the SYNC, AT, BLOCK, CALL, DO and DISP-commands. Most other commands EXPDOER sends back to EXPRUN, who handles them as if they were interactive user commands.
- ▶ **MONITOR**-wise runs explicitly a tight loop, which checks periodically various things like the current antenna position, which it updates in RTcommon, or if there is an experiment in the run queue, or a block in the gotoblock queue, or if it is time to put the UHF antenna to standby.

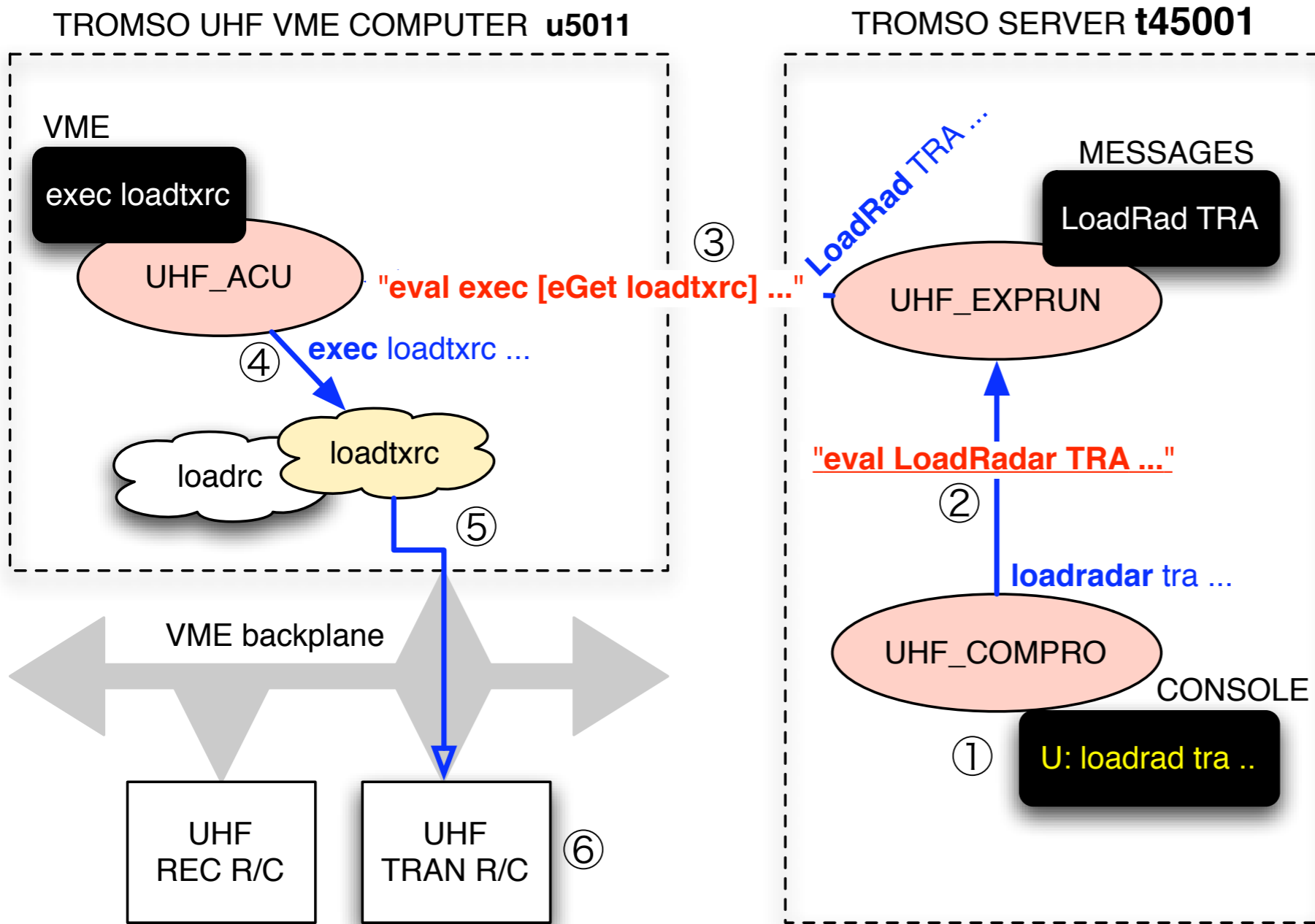
EROS4 command sources



All commands, both local and remote, are first send over the network to EXPRUN, which distributes them to the appropriate server processes: ACU, decodump, etc.

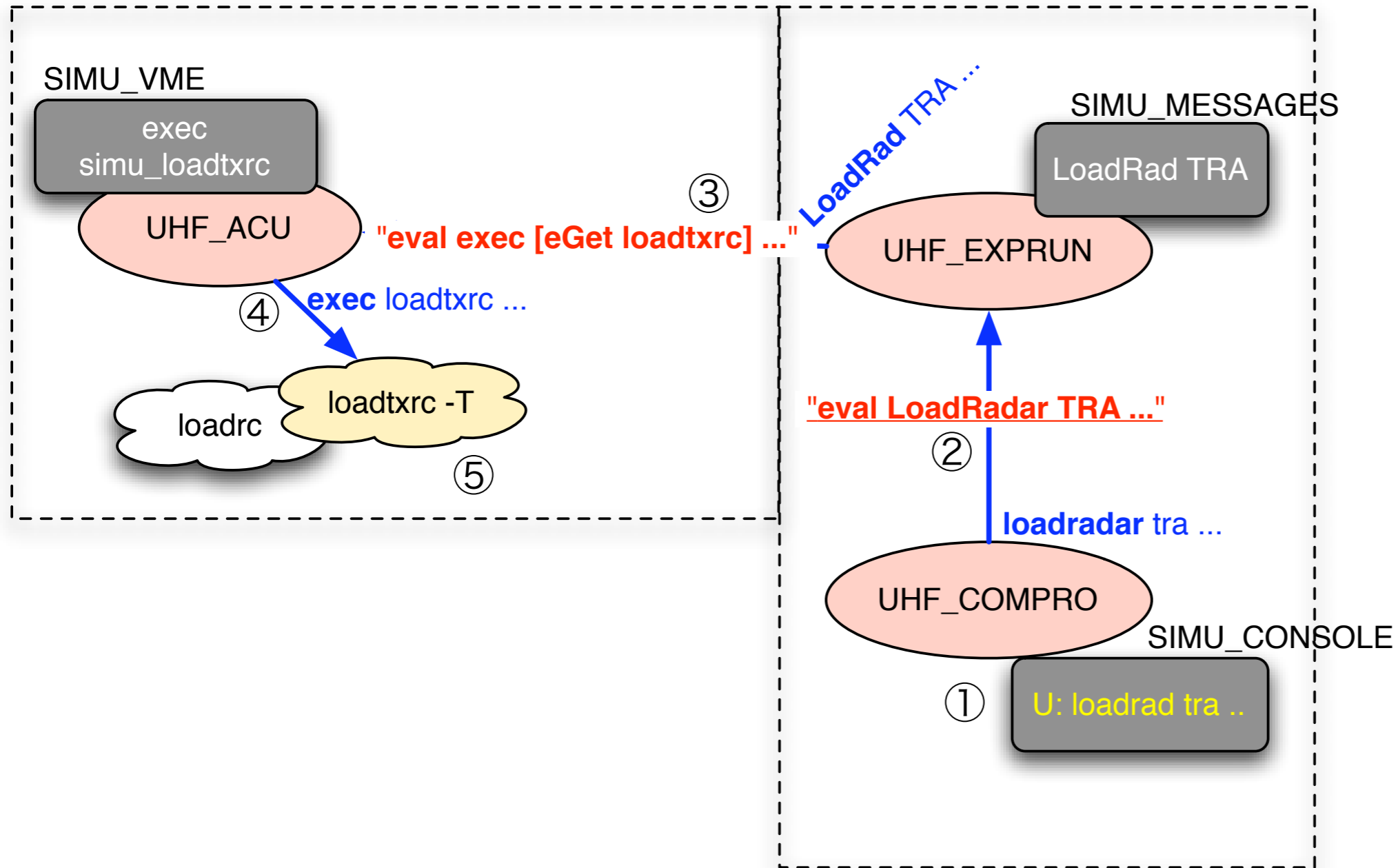
This arrangement is meant to ensure the integrity of the hardware access.

EROS4 internal command passing



EROS4 simulator, internal command passing

your workstation



EROS4 - what little is left of the "R"

E-TIME

B-TIME

C-TIME

U-TIME

```
EROS4 console (SOD)
S: printexp
EXPERIMENT (SOD) 27-Jul-2005 04:37:42.2
-----
Exp file   : tau2p1.e1an
dir        : /kst/exp/tau2p1
state      : RUNNING since 26-Jul 22:57:36.5
E-time     : 26-Jul 22:55:00.0
Block      : o1dcp1 tau2p1 292.9
launch     : 26-Jul 22:57:42.0
B-time     : 26-Jul 22:55:00.0
Cont.at    : 27-Jul 04:38:40.0
loop       : 450.0 s
Rad contr  : started 26-Jul 22:57:45.0
Latest     : /data1/tau2p1_r_o1dcp1_1.10_FR@sod/20050727_04/17901455.mat
           : 27-Jul 04:37:35 (7 sec ago)
S: █
```

Implementation of SYNC and AT commands

file:/kst/eros4/lib/expdoer.tcl

```
proc SYNC { Nsec } {  
    global C-TIME  
    set C-TIME [expr $C-TIME + $Nsec]  
    while { [UT] < $C-TIME } {  
        Serve_background_commands  
    }  
}
```

```
proc AT { UTspec } {  
    global C-TIME  
    set C-TIME $UTspec  
    while { [UT] < $C-TIME } {  
        Serve_background_commands  
    }  
}
```

Time-synchronisation in an ELAN program

