

COACK MULTI-SERVER SYSTEM WITH STARS

T.Kosuge, Y.Saito, K.Nigorikawa, H.Katagiri

A.Shirakawa, H.Nakajima, K.Ito

J.Kishiro and S.Kurokawa KEK

Isamu Abe

The logo for COACK features the word "COACK" in a bold, italicized, light green font. Behind the text are several thick, diagonal brush strokes in shades of blue, green, and yellow. A thin horizontal line is positioned below the text.

COACK

Contents

- Multi server system in COACK
 - Beamline interlock and central control system---
- Development of multi-server
- Installation and result
- Summary



Multi server system in COACK

- Security(read only)
- Mirroring
- Firewall
- Load sharing

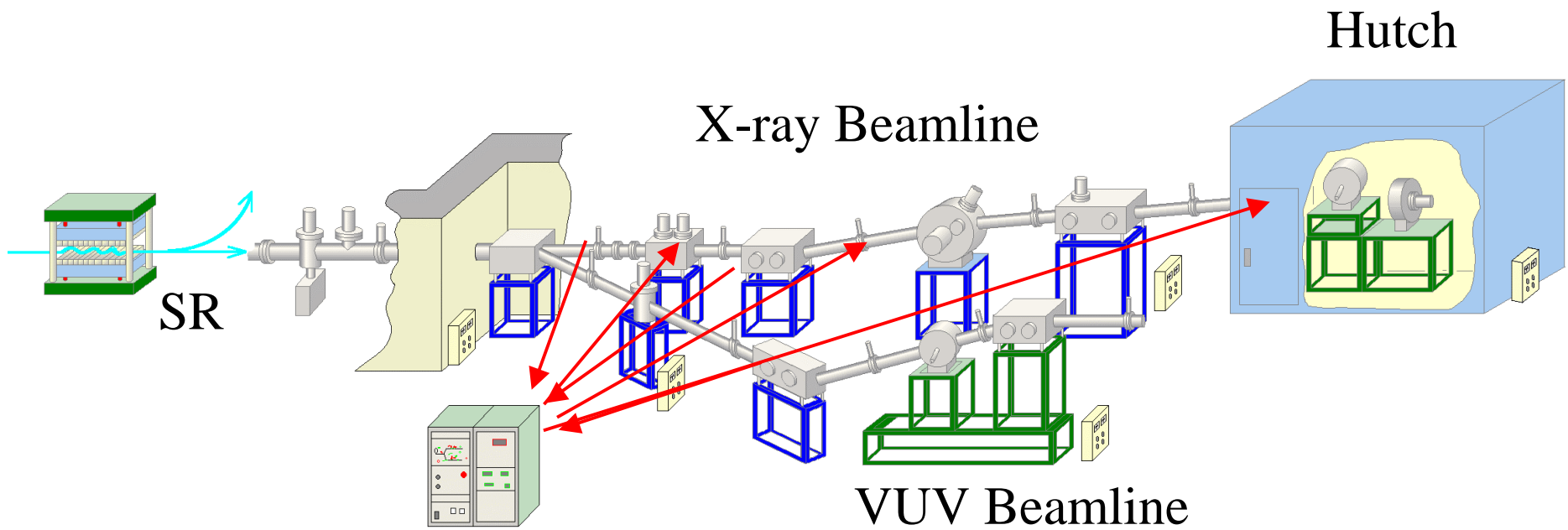


Beamline interlock and central control system

- Radiation safety
- Vacuum interlock
- Protect beamline components against radiation damage

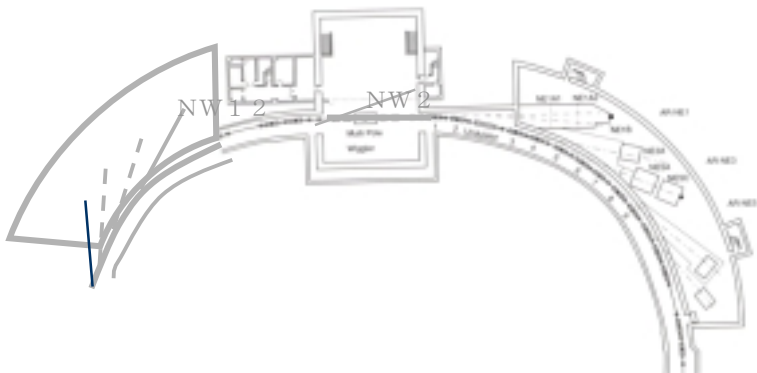


Beamline and Beamline Interlock System



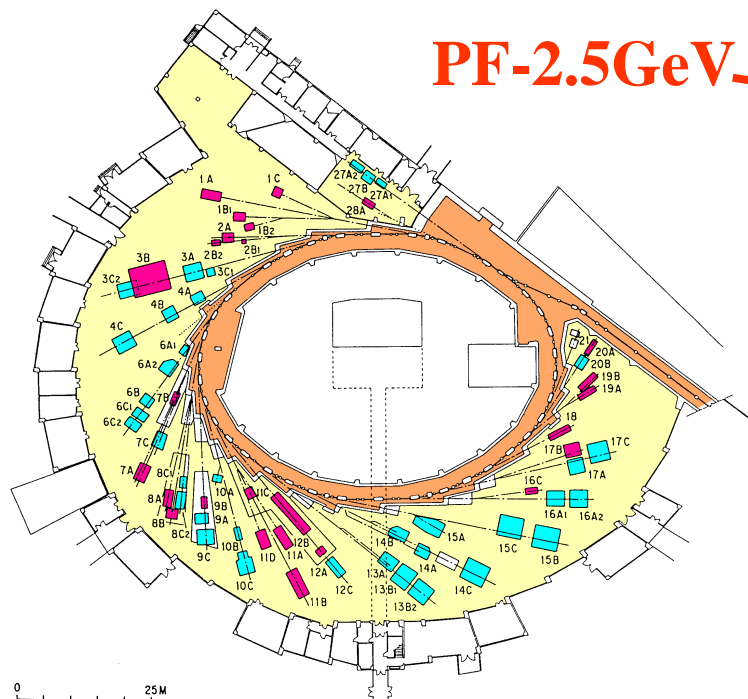
Interlock System

*Beamline components are controlled by
PLC (Programmable Logic Controller)*

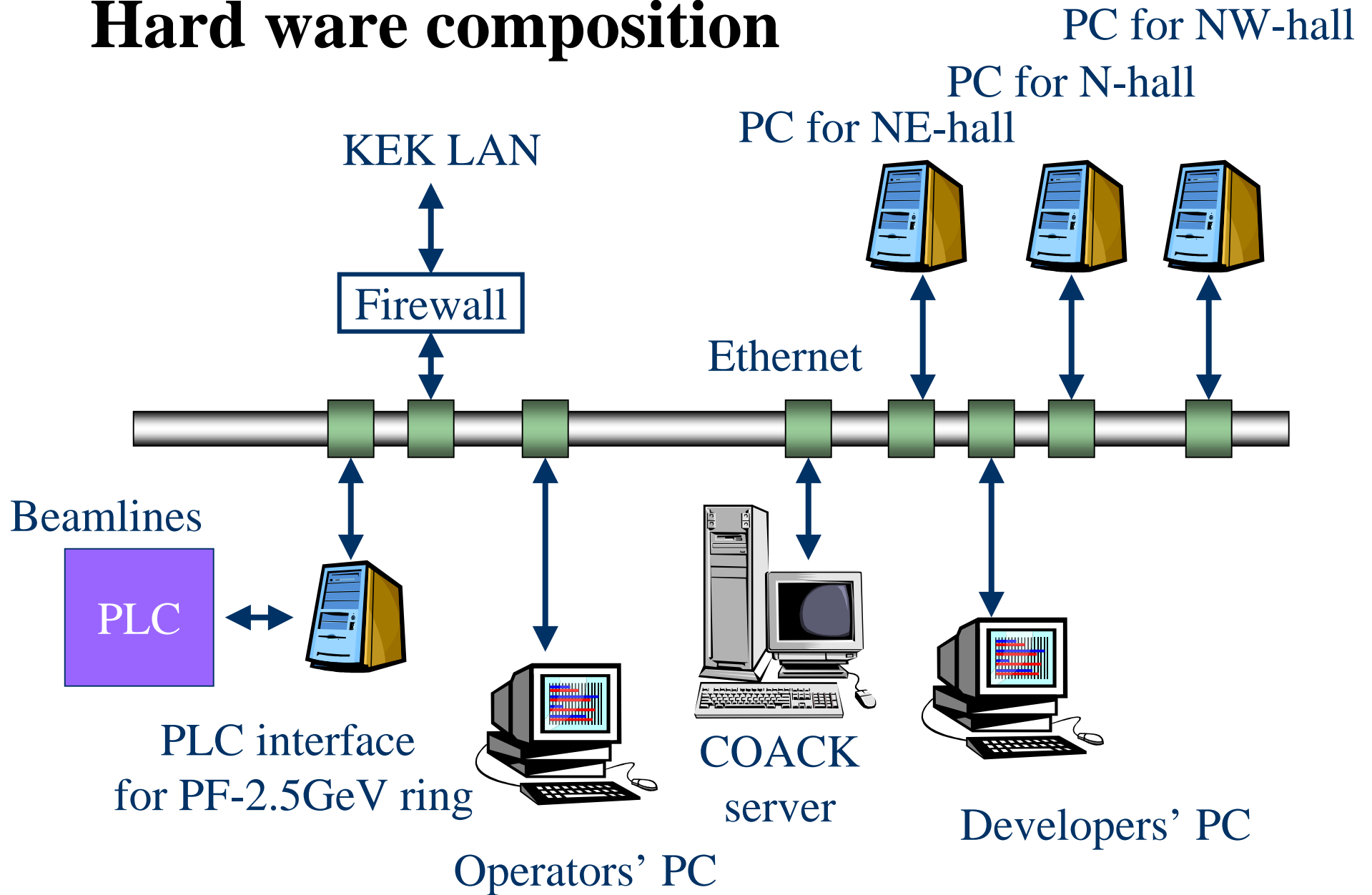


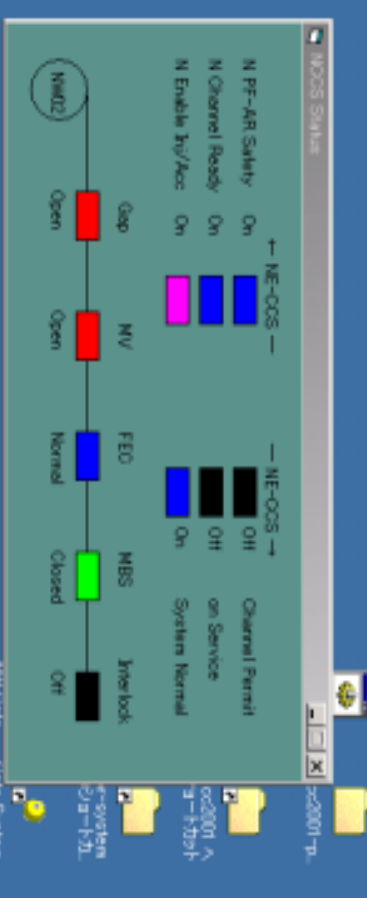
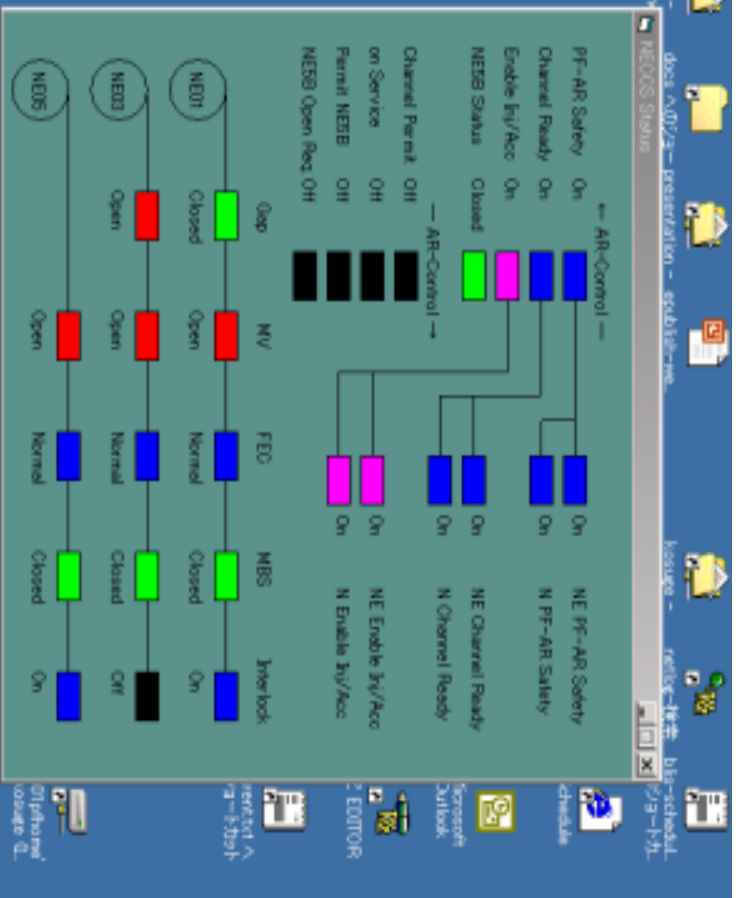
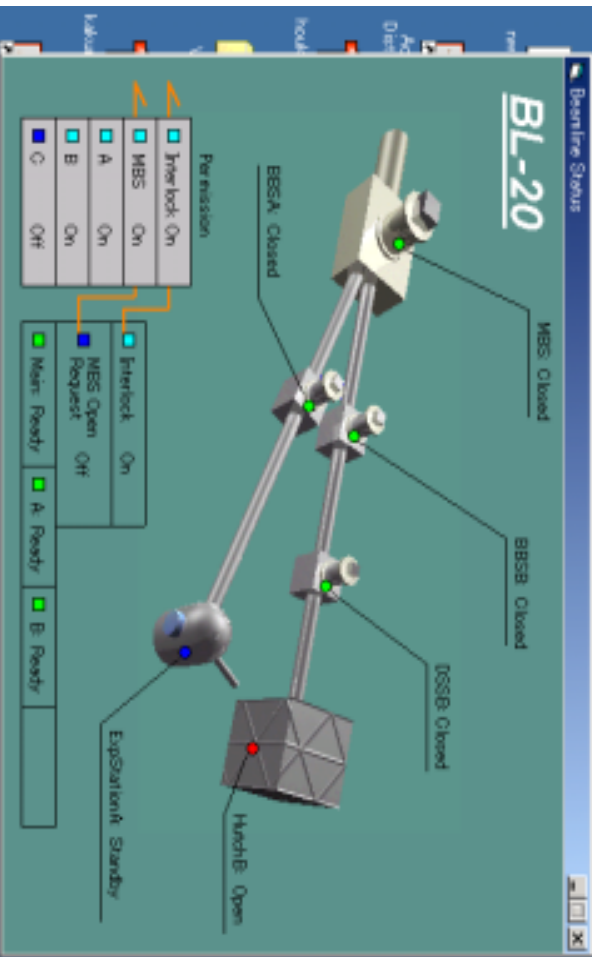
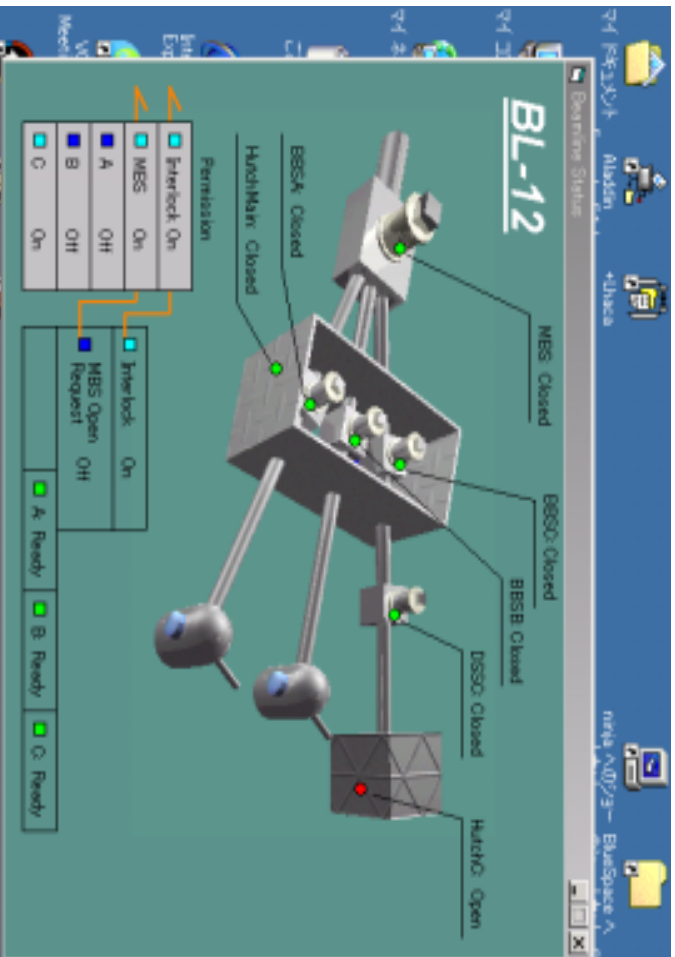
PF-AR

PF-2.5GeV

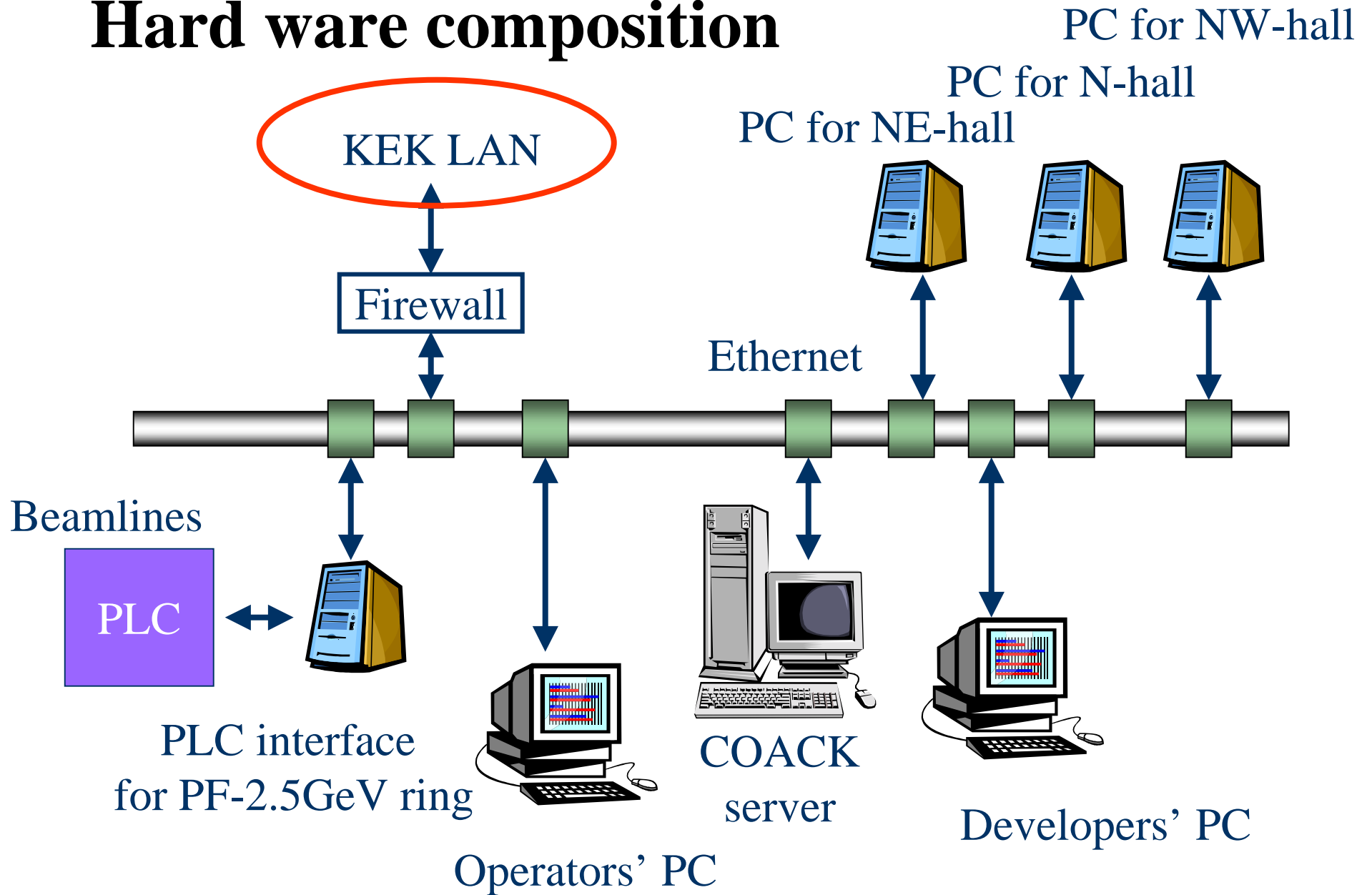


Hard ware composition

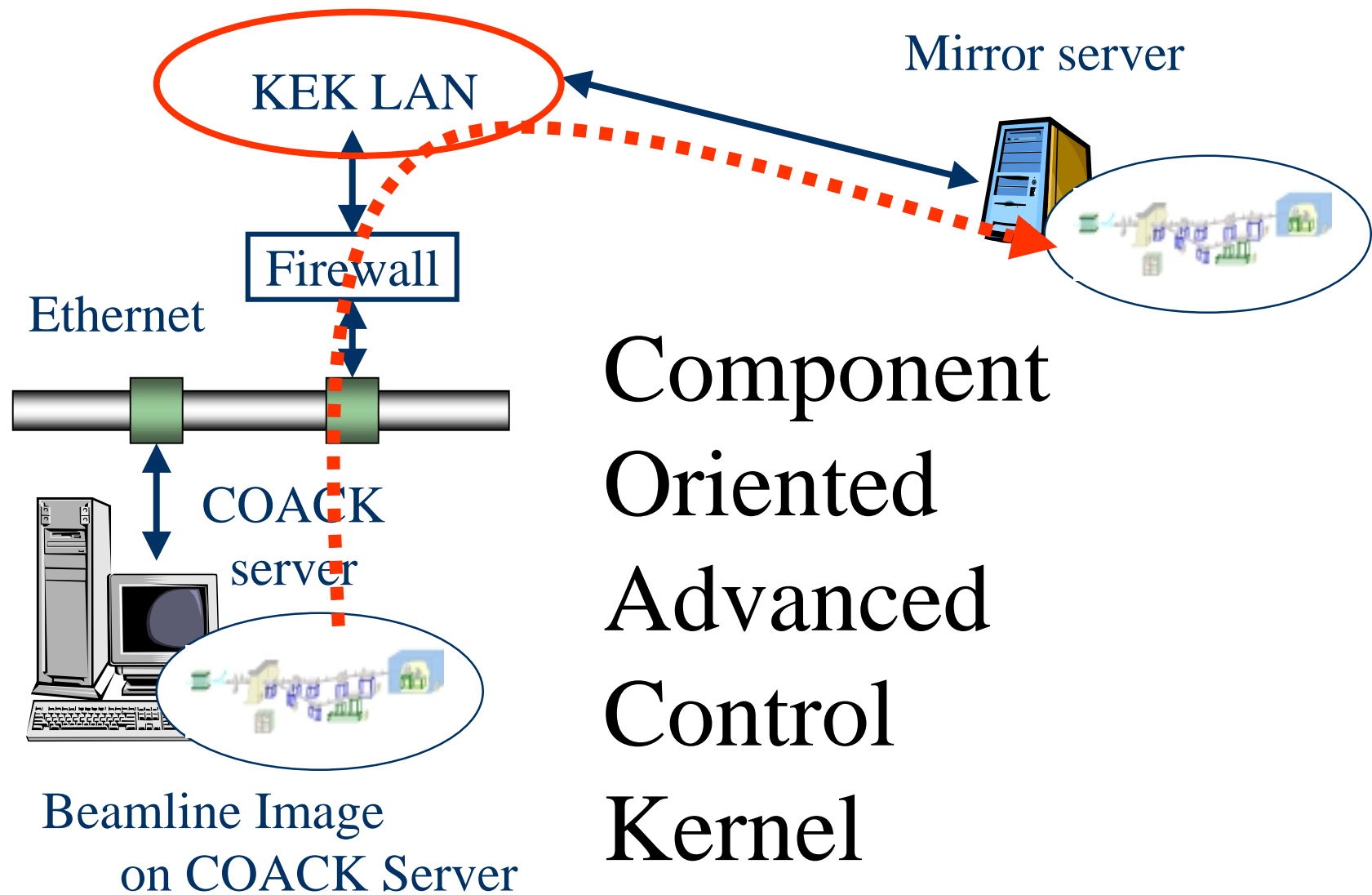




Hard ware composition



Requirements and solution



Development of multi-server

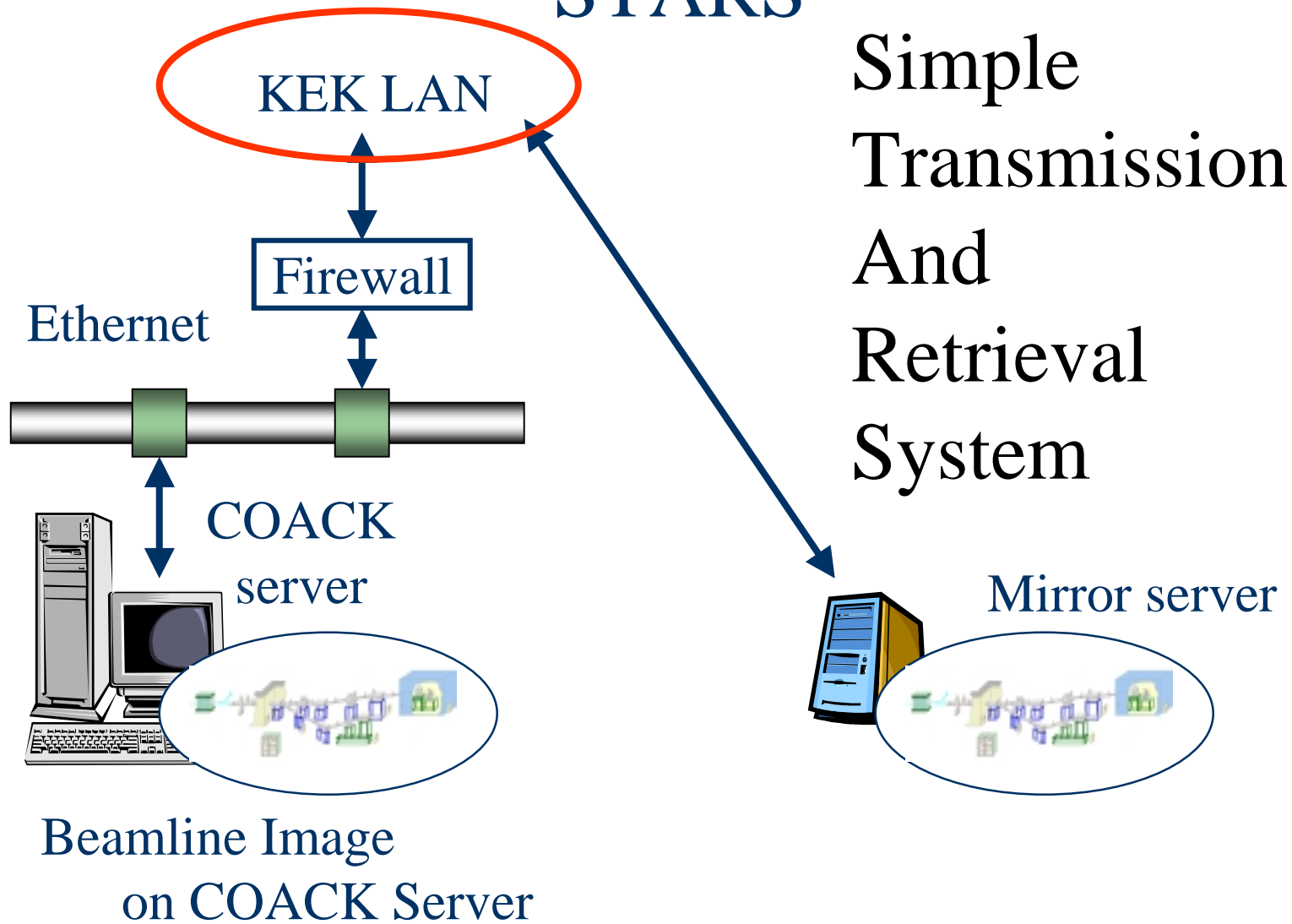
Requirements

- Load sharing
- Short construction time
- Easy to implement

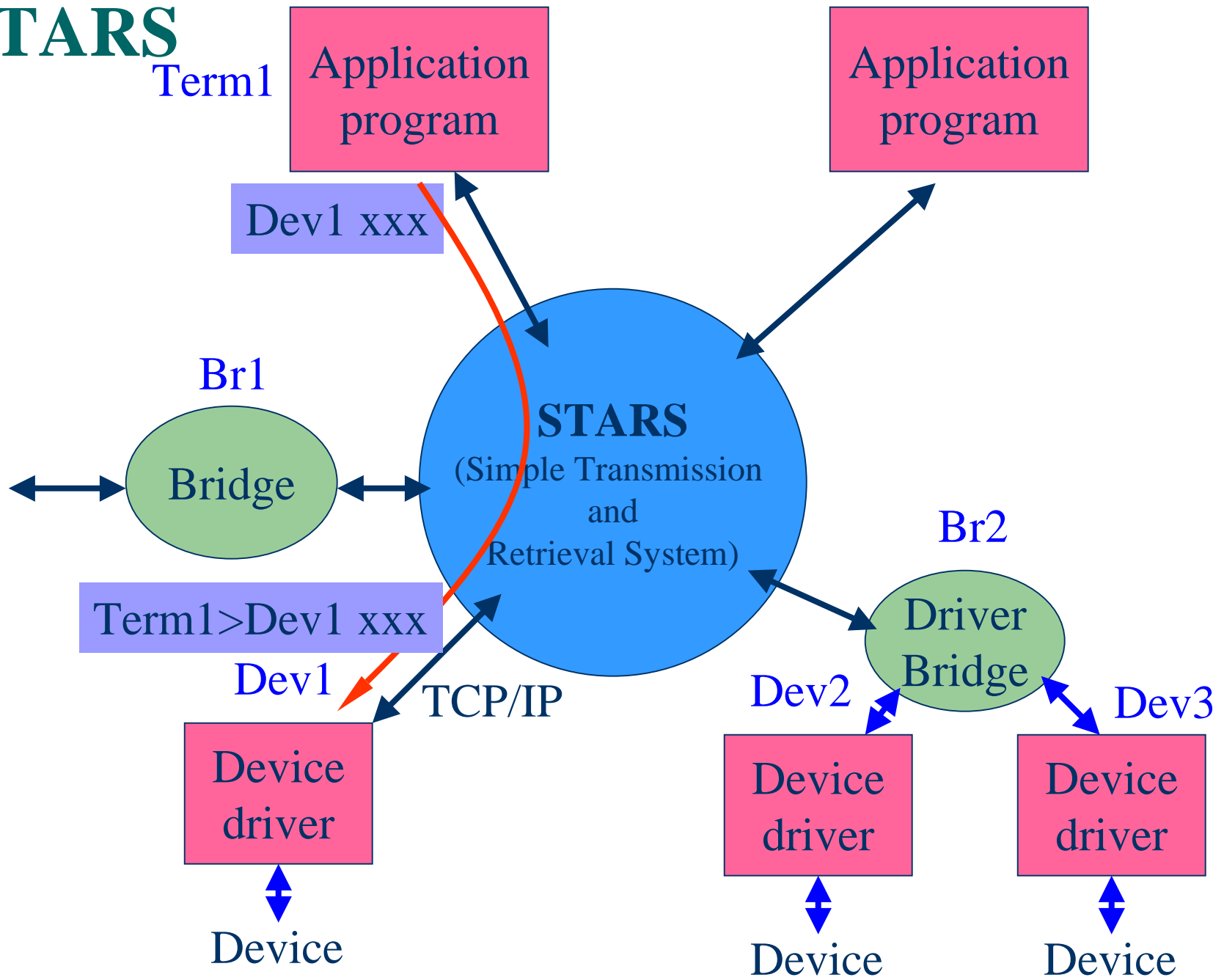


Mirroring

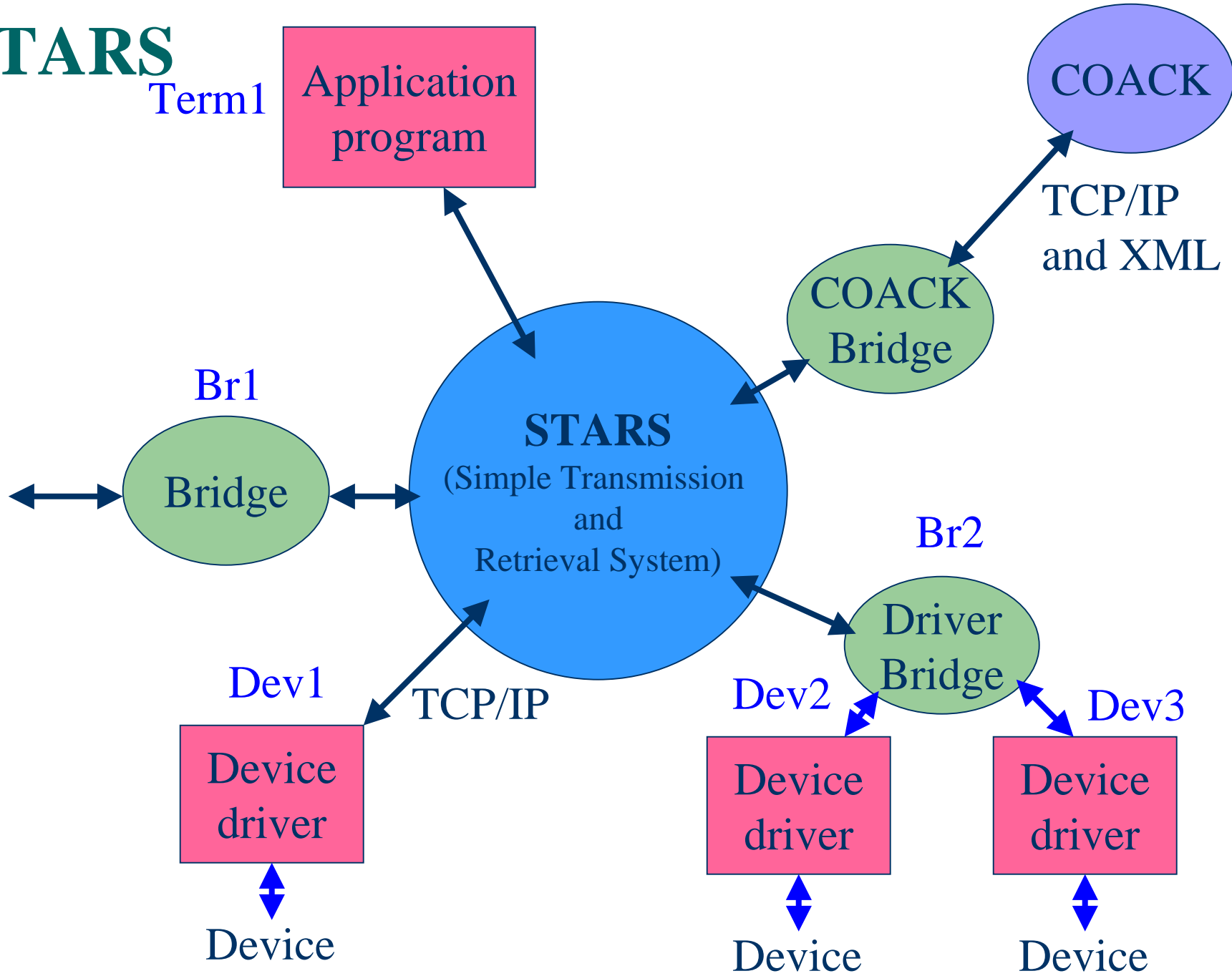
STARS

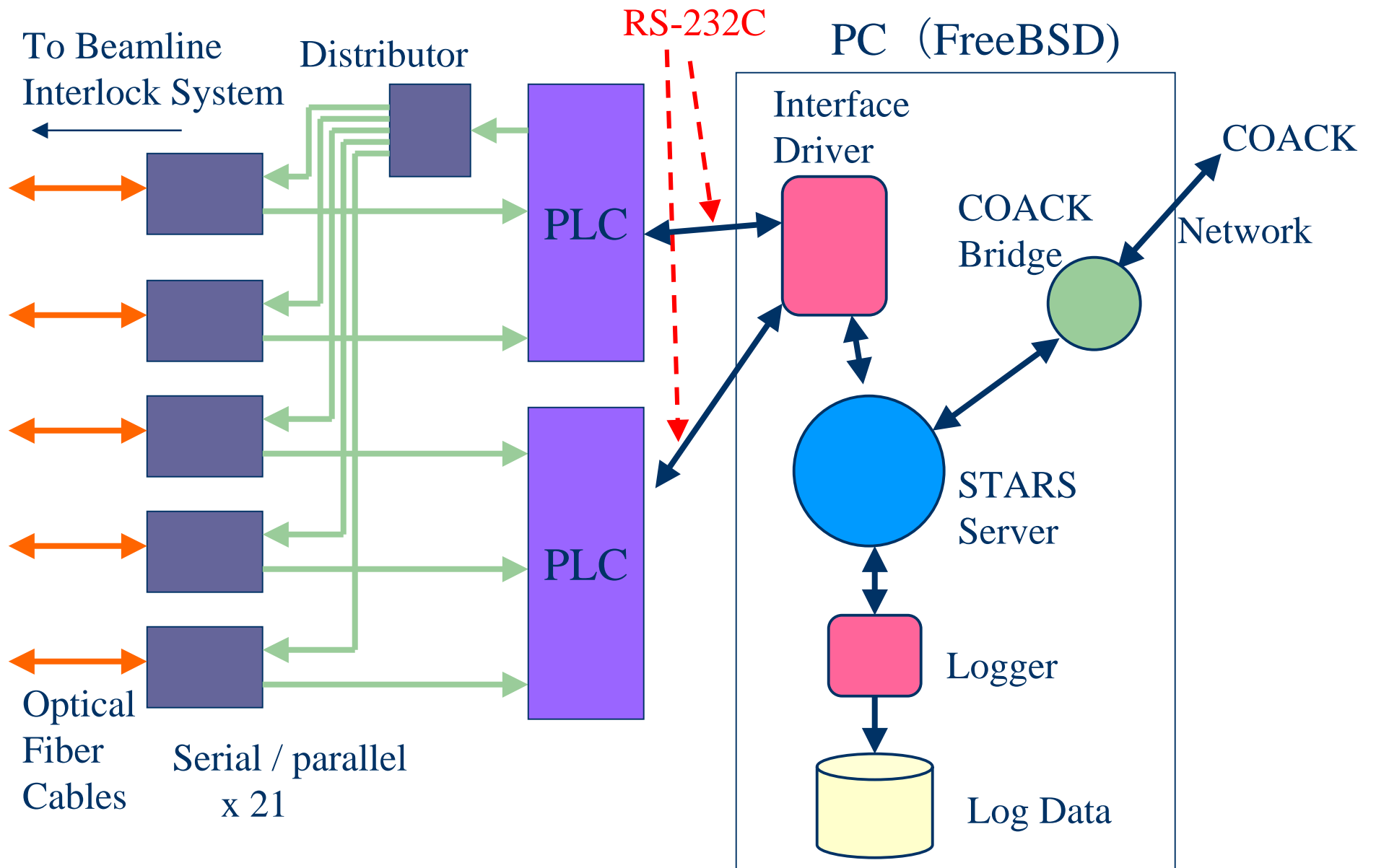


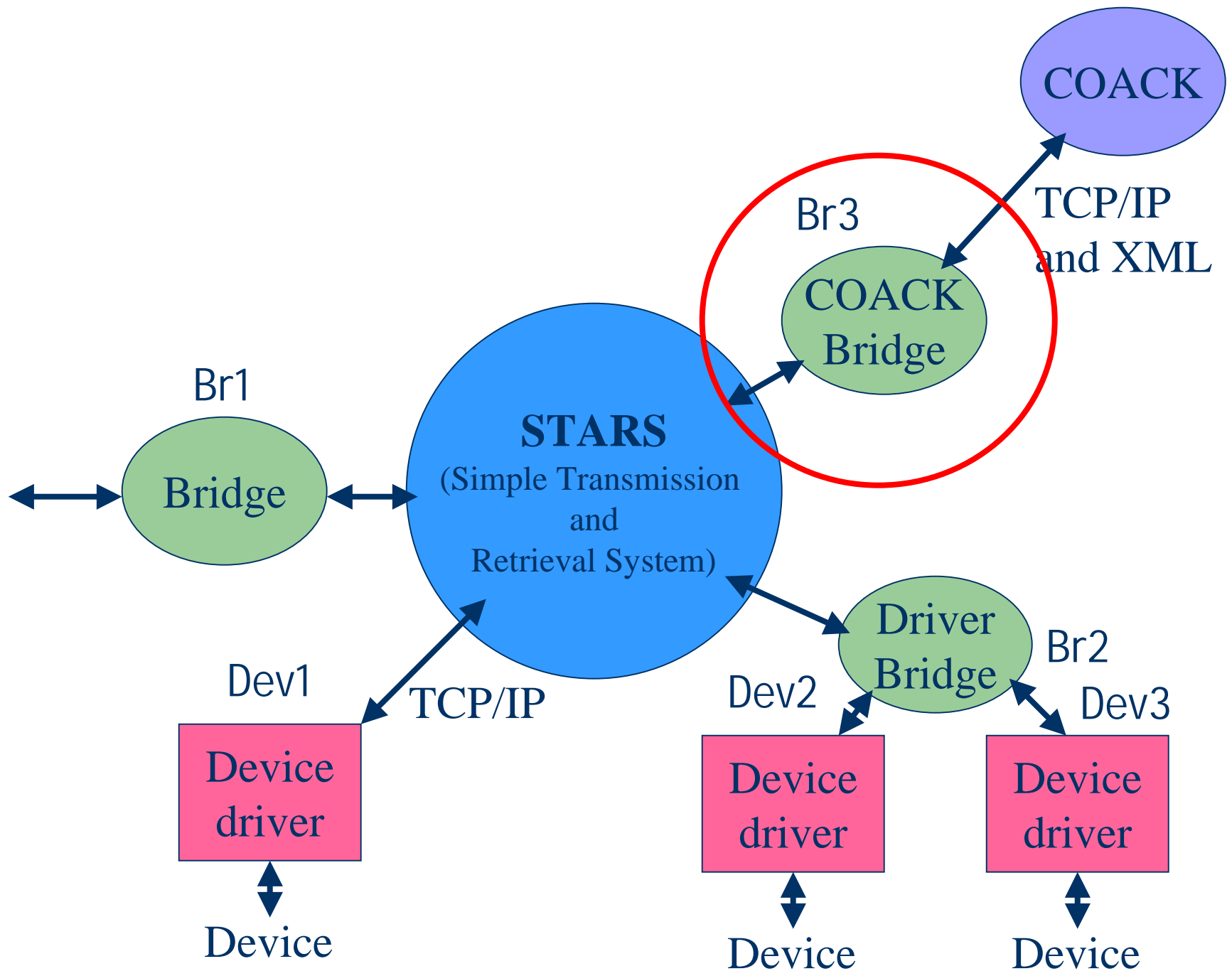
STARS

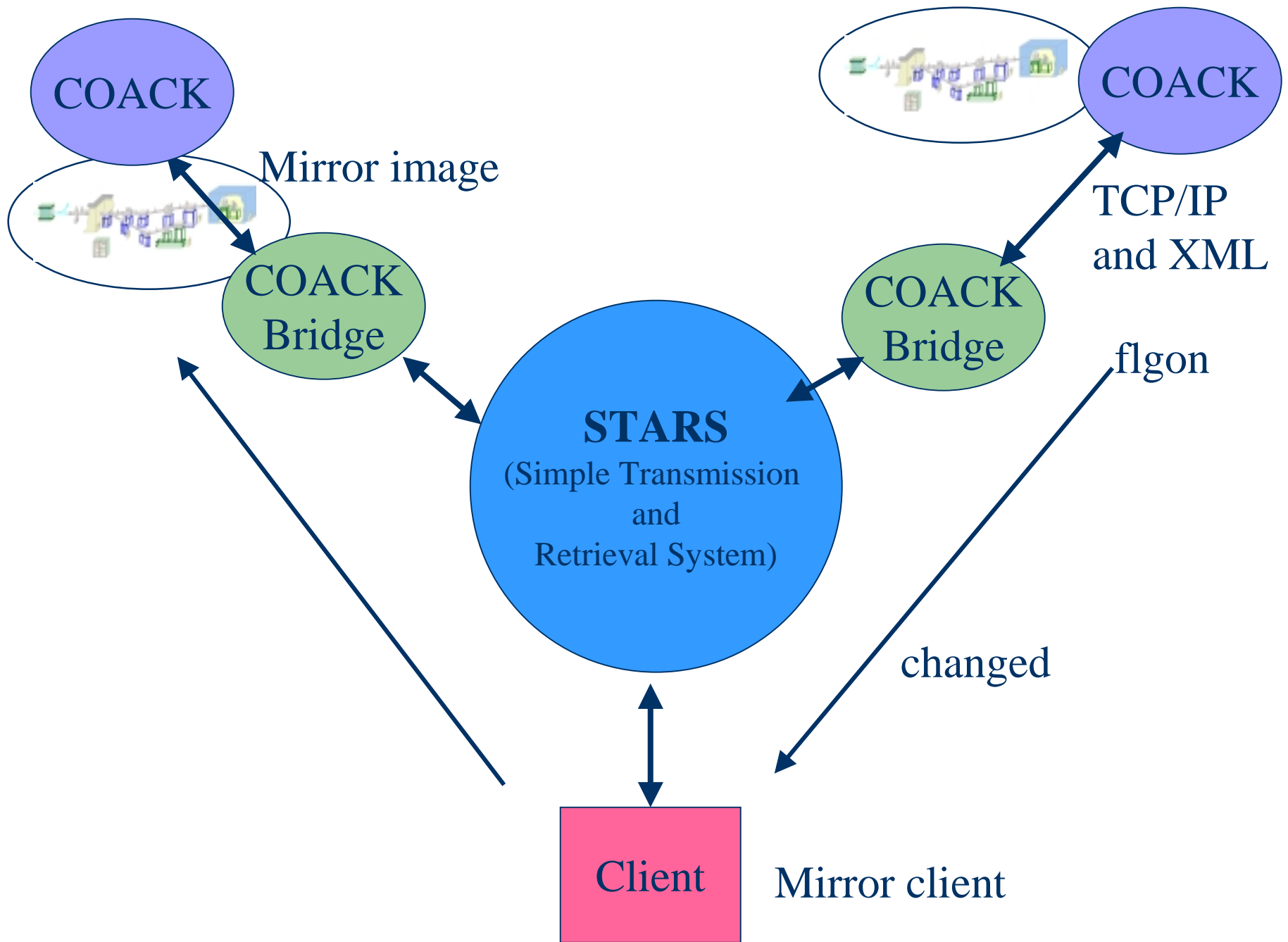


STARS









```

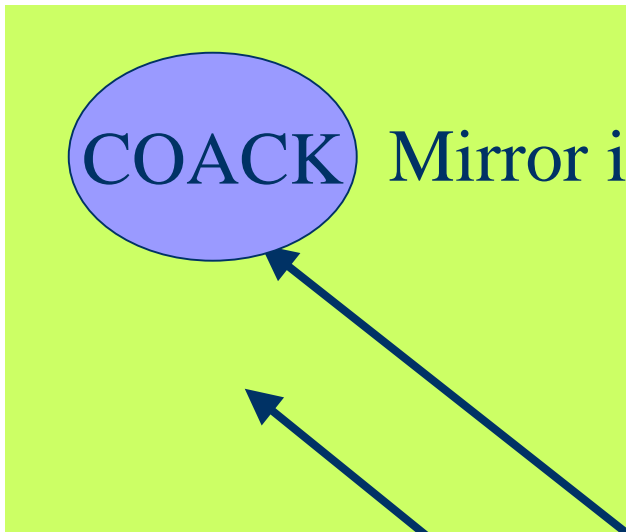
#!/usr/bin/perl
use strict;
use Getopt::Long;
use stars;
#####
# mirrorman
#####
## ToDo: Set parameters here.
$::NodeName = 'mirrorman';    #Default node name.
$::Server = 'localhost';    #Default stars server.
$::RealImage = 'coast';    #Bridge name of Real COACK
$::MirrorImage = 'coastmir';    #Bridge name of Mirror COACK
require 'mirrorman.cfg';
#####
## ToDo: You can set option switches. See help 'Getopt::Long'.
GetOptions(
'node=s' => \$::NodeName,
'h' => \&usage,
) or die "Bad switch.\n";
if($_ = shift(@ARGV)){ $::Server = $_; }
## Open Stars server. $::tak is a Stars object.
$::tak = stars->new($::NodeName, $::Server)
    or die "Could not connect Stars server";
$::tak->addcallback(\&handler);
my $i;
my $mnode;
my $mproperty;
## Initialize ##
$::tak->Send("create", $::RealImage);
$::tak->Send("create", $::MirrorImage);
for $i (@::flgon){ $::tak->Send("$::RealImage.$i _flgon"); }
for $i (@::mflgon){ $::tak->Send("$::RealImage.$i _mflgon"); }
for $i (@::getcache){ $::tak->Send("$::RealImage.$i getcache"); }
for $i (@::mgetcache){
    ($mnode, $mproperty) = split("_", $i);
    $::tak->Send("$::RealImage.$mnode mgetcache $mproperty");
}
stars->Mainloop();
exit(1);

```

```

# Print usage. -----
sub usage{
## ToDo: Please modify help message for "-h" option.
    print "Usage: mirrorman [-h] [-node MyNodeName] [StarsServer]\n";
    exit(0);
}
# Command handler from Stars server -----
sub handler{
## ToDo: Please modify handler sub routine.
## (The handler sub routine will be called when client
## receives a message from a Stars server.)
    my ($from, $to, $mess) = @_ ;
    if($mess =~ /^_ WatchDataChanged\s+(\S+)\s+(\S+)/){
        set_mirror($from, $1, $2);
    }elseif($mess =~ /^_@GetCache\s+(\S+)\s+(\S+)/){
        set_mirror($from, $1, $2);
    }elseif($mess eq 'hello'){ $::tak->Send("@hello nice to meet you.", $from); }
    elseif($mess eq 'help'){ $::tak->Send("@help hello", $from); }
    elseif($mess =~ /^[_@]/){ return; }
    else{ $::tak->Send("@$mess Er: Bad command or parameter", "$from"); }
}
}
sub set_mirror{
    my $from = shift;
    my $property = shift;
    my $value = shift;
    unless($from =~ s/^\$::RealImage\.//){ return; }
    $::tak->Send("$::MirrorImage.$from _setcache $property $value");
}

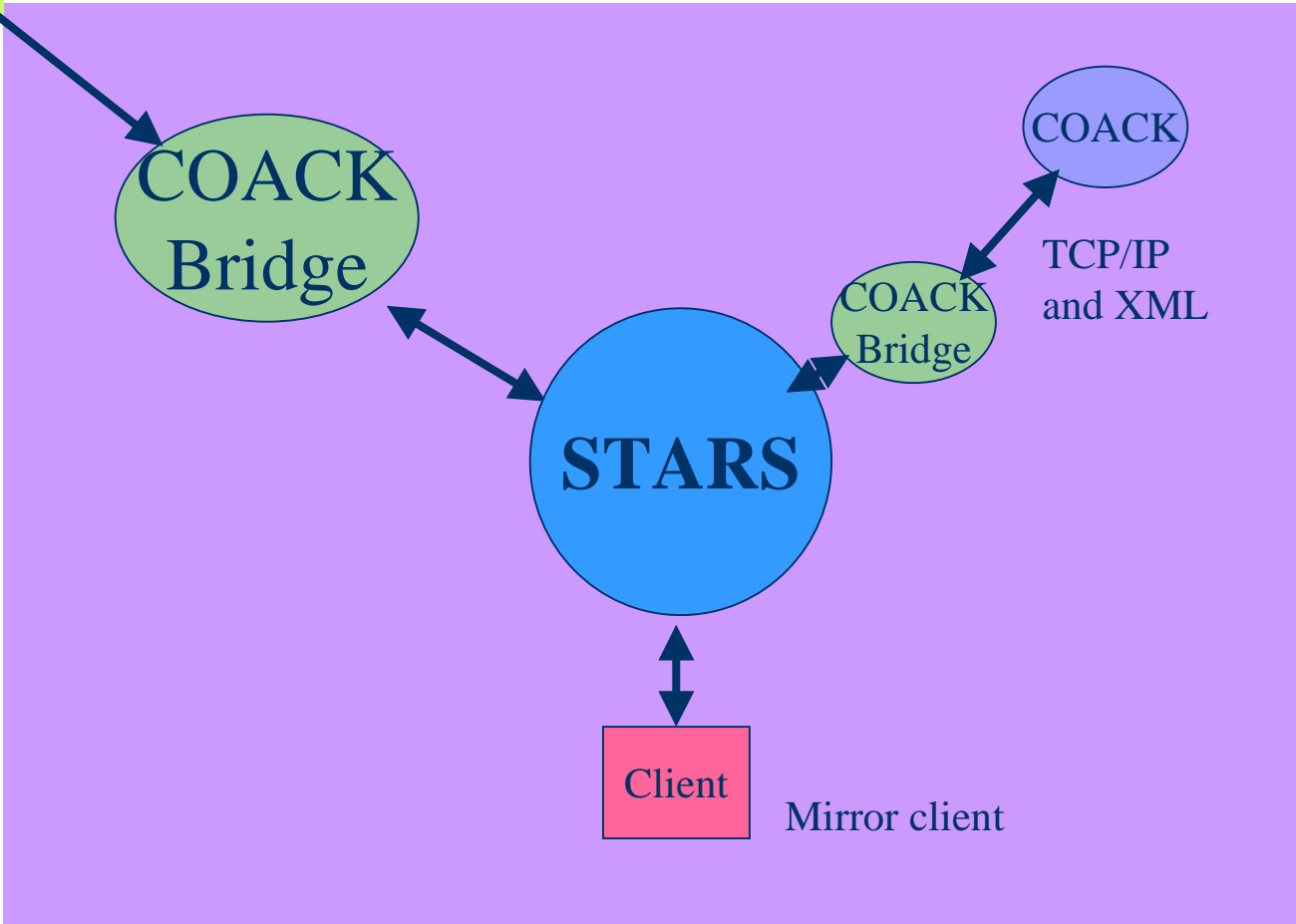
```



COACK Mirror image

Bi-direction is not required

Connection



COACK Bridge

STARS

COACK Bridge

COACK

TCP/IP and XML

Client

Mirror client

Advantages

- Load sharing
- Read only server
- Easy construction
- Available through Firewall

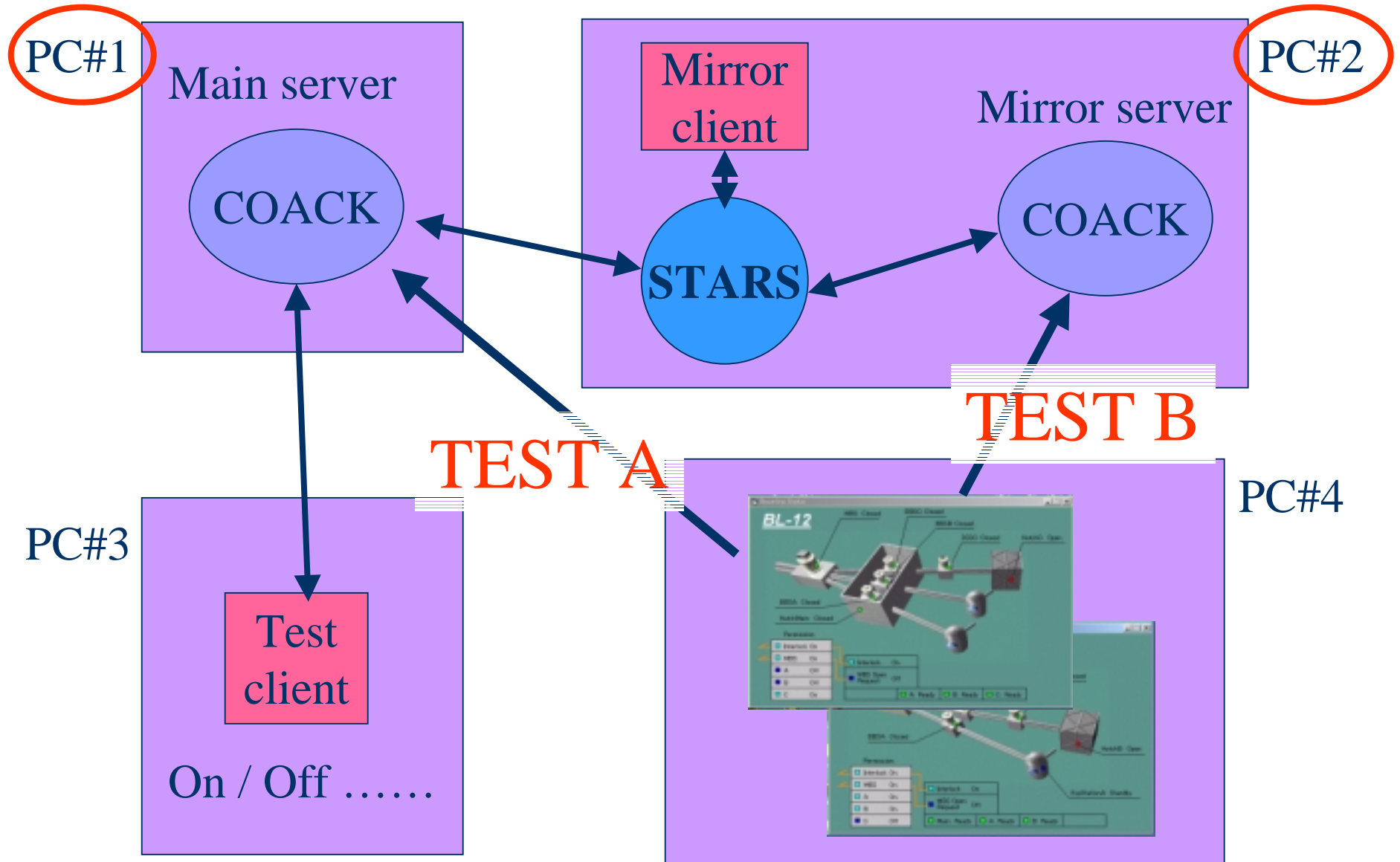


Installation and result

- Installed into central control system of beamline interlock
- Working stably



Test of load sharing



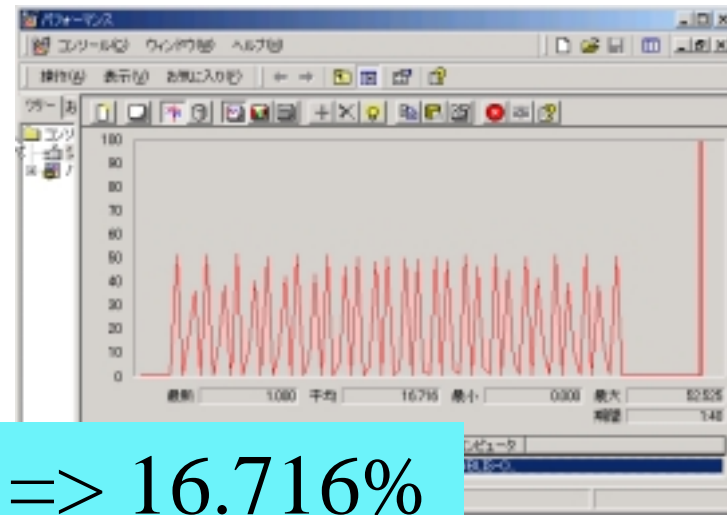
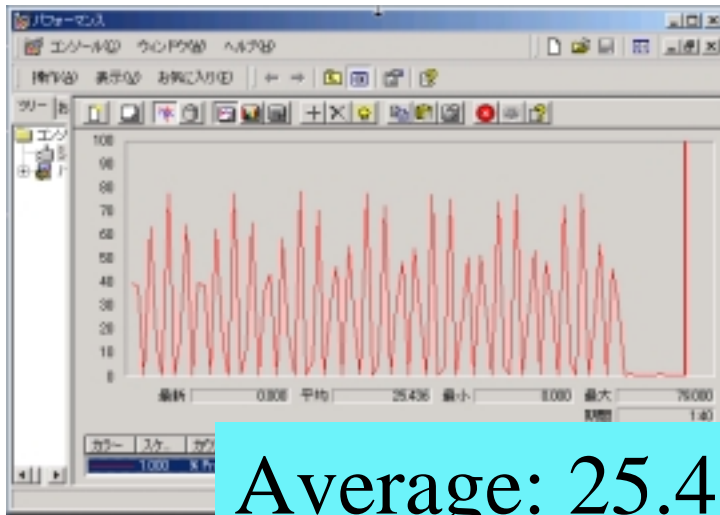
CPU load

TEST A

TEST B

Main server

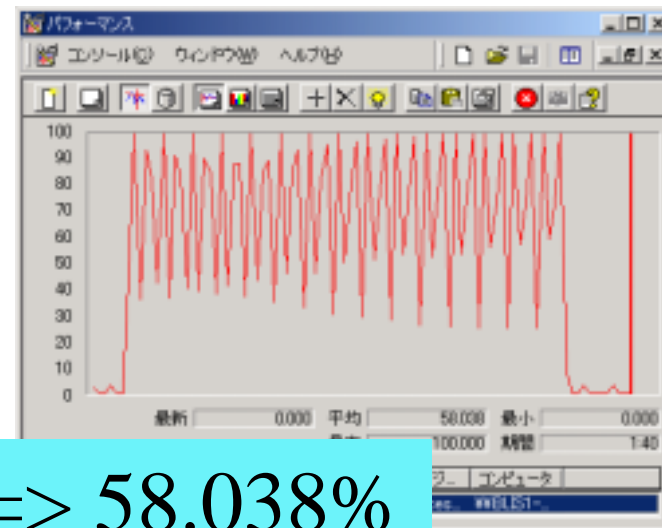
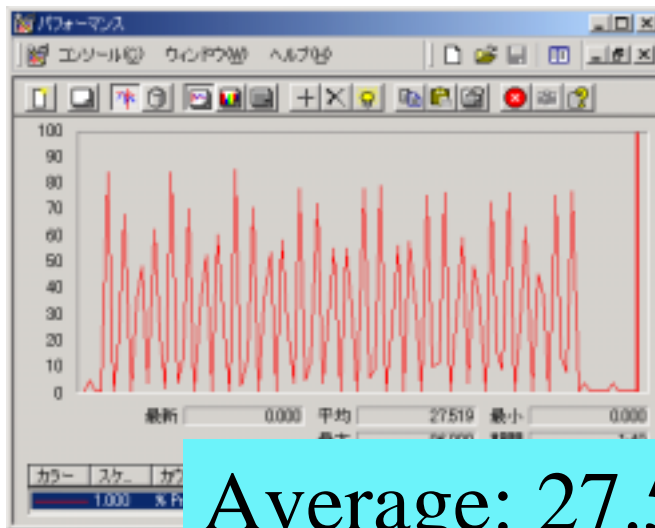
PC#1



Average: 25.436% => 16.716%

Mirror server

PC#2



Average: 27.519% => 58.038%

Summary

- We developed COACK multi server system with STARS
- Mirroring with STARS is effective
- We demonstrated the effectiveness of the multi server system in COACK
- COACK will be upgraded to support multi server system

