Server Redirection as a tool in the **TINE** Control System

S. Herb, MSC, DESY

Tine Workshop, Sept. 2007

Introduction

What is ‘Redirection’ and what is it good for?

The TINE (and DOOCS) control system addresses are organized as *tree structures* in which the **server processes** play a central role.

Facility / **DeviceGroup (~ *server*)** / DeviceInstance / Property
‘Hera’ / **‘BPM’** / ‘WL334MX’ / ‘Intensity’

(assumes that all BPMs are controlled by one server?)

These trees may provide a good description of the control system structure, but **not** such a good **object-oriented view** of the accelerator, since, especially for a large facility, group devices (and their properties) may be spread over multiple servers.

Introduction

For Example

- BPM readout could be distributed over many servers
- BPM calibration data might be on a DB server
- BPM property histories could be on a history server

But as CS users, we would like to have simple commands

get (HERA / BPM / WL117MX / Intensity.History, …)

which hide the underlying multiple server structure

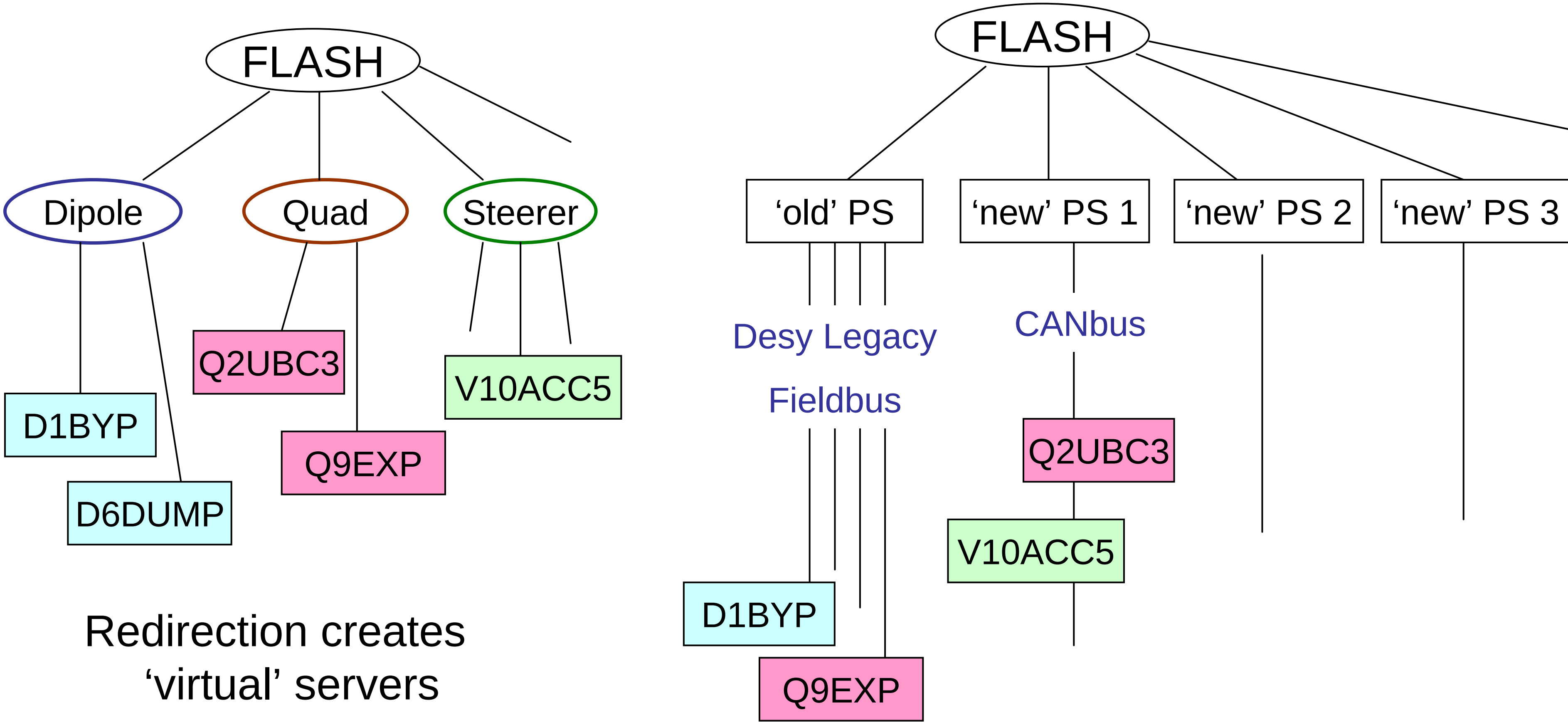
So this is really a **Name Resolution** and **Naming** issue, and **Redirection** represents a set of tools within TINE for handling it.

I will describe a specific use, involving control of Magnet PSs for **FLASH**

FLASH Magnet PS (~ 180 ‘old’ , 70 ‘new’)

What they want to see (and do)

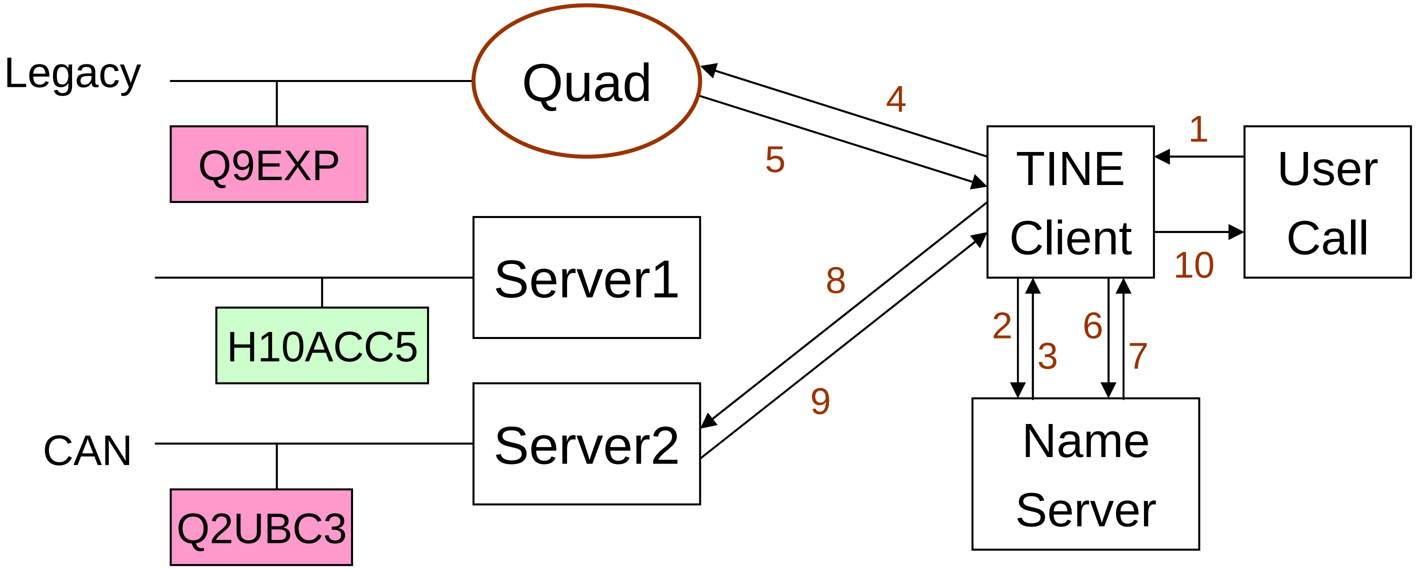
What’s really there



Redirection creates
‘virtual’ servers

The Server structure **really** does not
mirror the accelerator structure!

How does it work ? (for this case)



get (“FLASH / QUAD / Q2UBC3 / DCVOLTS”)

- First call results in Nameserver query: “where is server Quad?”
- Client sends to Quad/Q2UBC3, receives ‘this device is on Server2 gets the new address from Nameserver, and sends the call to Server2.
- For the next call, this information has already been **cached** at the TINE client level and the call proceeds directly to ‘Server2’

Is this billiards game with RPC calls really necessary?

- Control System Naming is a problem with **no** ‘good’ solution
- The underlying CS structure **will** be complicated.
- So it is worth some effort to present the user with an understandable hierarchical representation based on ‘real’ accelerator devices
- The above mapping could also have been achieved entirely within the Nameserver, by storing complete addresses there:

  FLASH/QUAD/Q9ACC1/DCVOLTS => IP# X

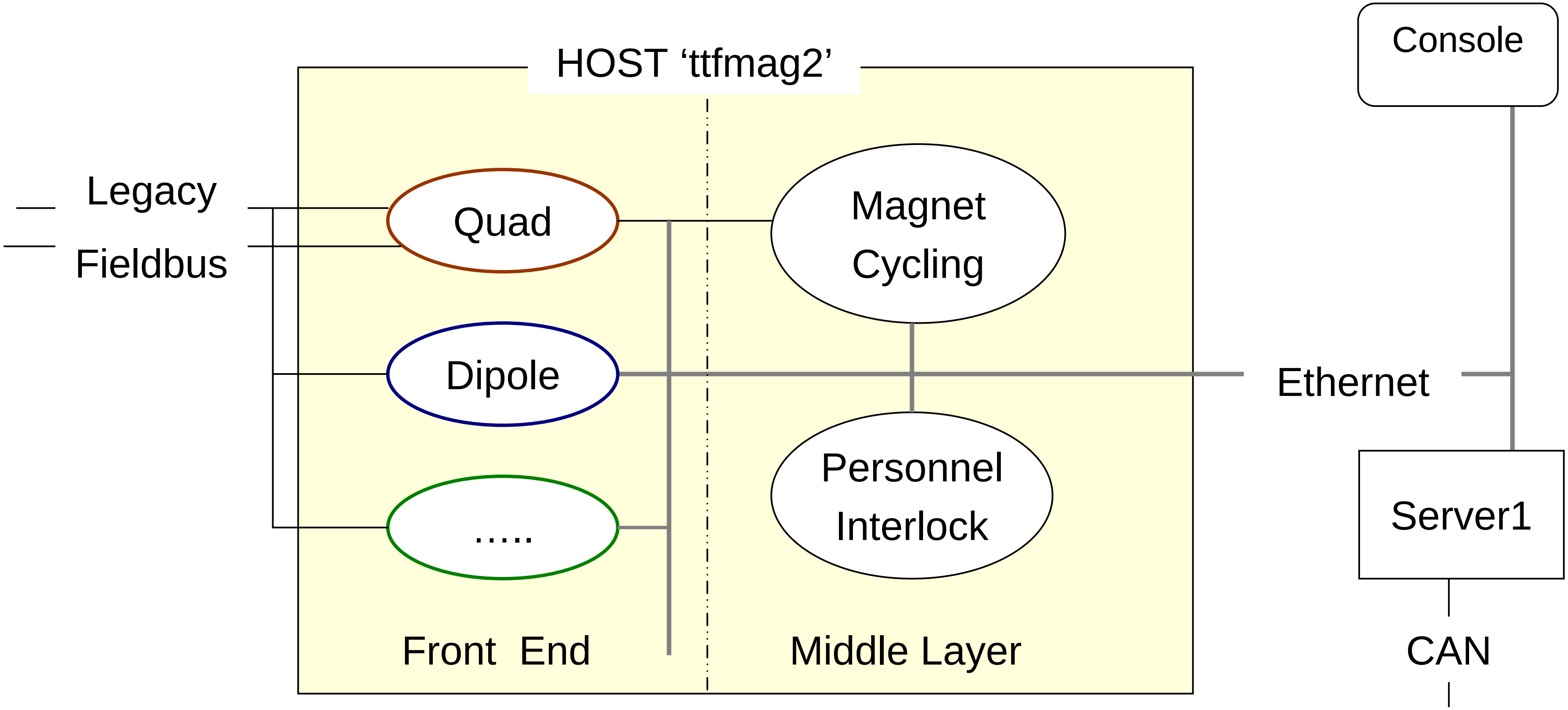
  FLASH/QUAD/Q9ACC1/DCVOLTS.History => IP# Y

  How to do the mapping is a design decision for the control system

- There is also a need for dealing with groups of distributed devices

  => “middle layer consolidation”

Flash ‘*Old PS Server(s)*’ revisited

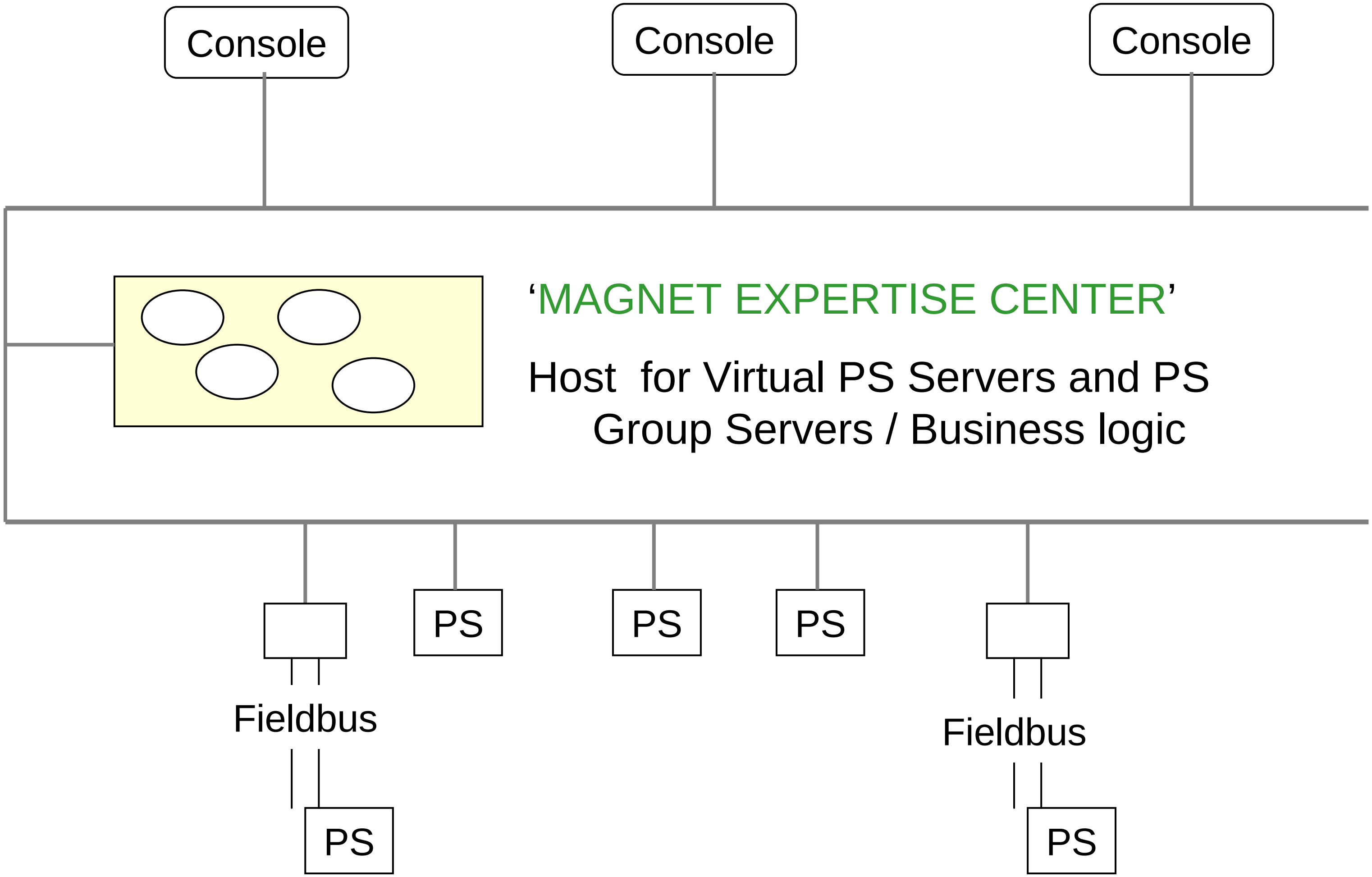


Middle Layer Servers handle activities spanning magnets on multiple hosts

They encapsulate ‘Business Logic’ which would otherwise be at console level

Groups can be defined at a single location, rather than console by console

Possible Distributed CS Topology


Console
Console
Console
‘MAGNET EXPERTISE CENTER’
Host for Virtual PS Servers and PS
Group Servers / Business logic
PS
PS
PS
Fieldbus
Fieldbus
PS
PS