PVSS based LHCb Testbeam Run Control

Old system

Why change?

Current system

Next year plans

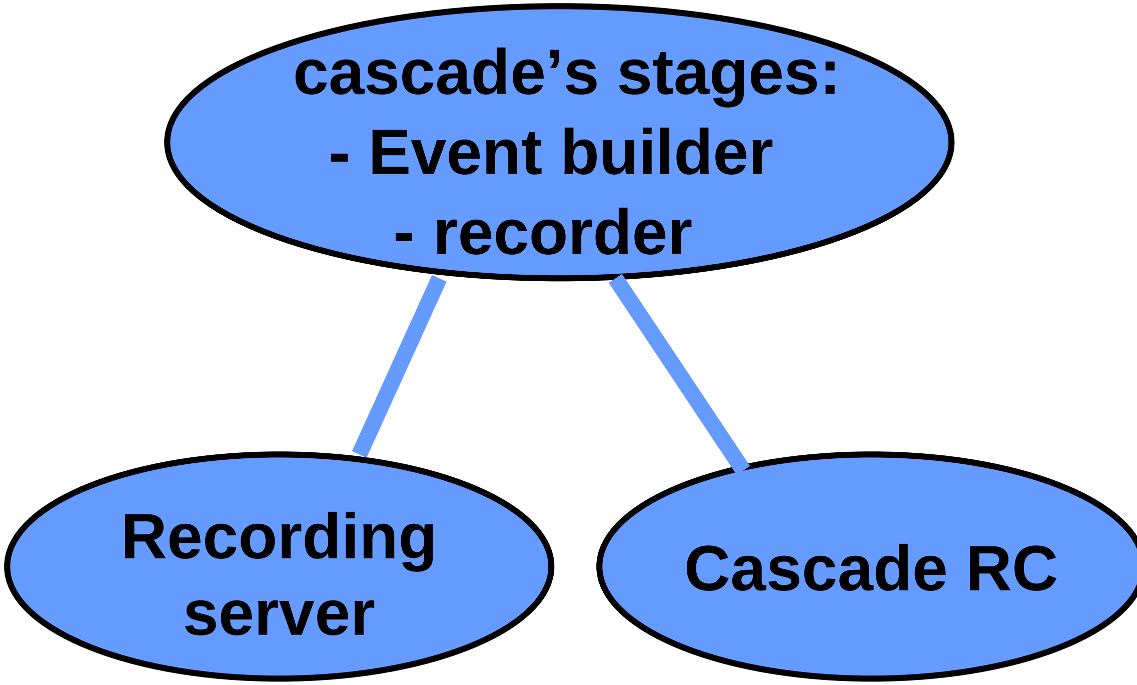
Old system configuration

Hard (OS):

VME: LynxOs

Station (HP)

Soft:



cascade’s stages:
- Event builder
- recorder

Recording server

Cascade RC

Why change?

- HP are not supported any more,
- Cascade Run Control Configuration is quite hard,
- We need a multi-user & wide-accessible control,
- A uniform system is easy to maintain & support,
- Run files management has to be ‘clear’,
- All testbeam operations have to be logged,
- A central support allows quick debug.

Why PVSS?

- This system will be used in the future LHC experiments,
- Testbeam Run Control requirements are strong:
  - 24h/24h system,
  - Error-recovery.

  -> It's not just a PVSS test, it's a PVSS real application.
- PVSS is used here for:
  - User interface,
  - All system's configuration,
  - All user's parameters,
  - Support,
  - All system monitoring.

Current System

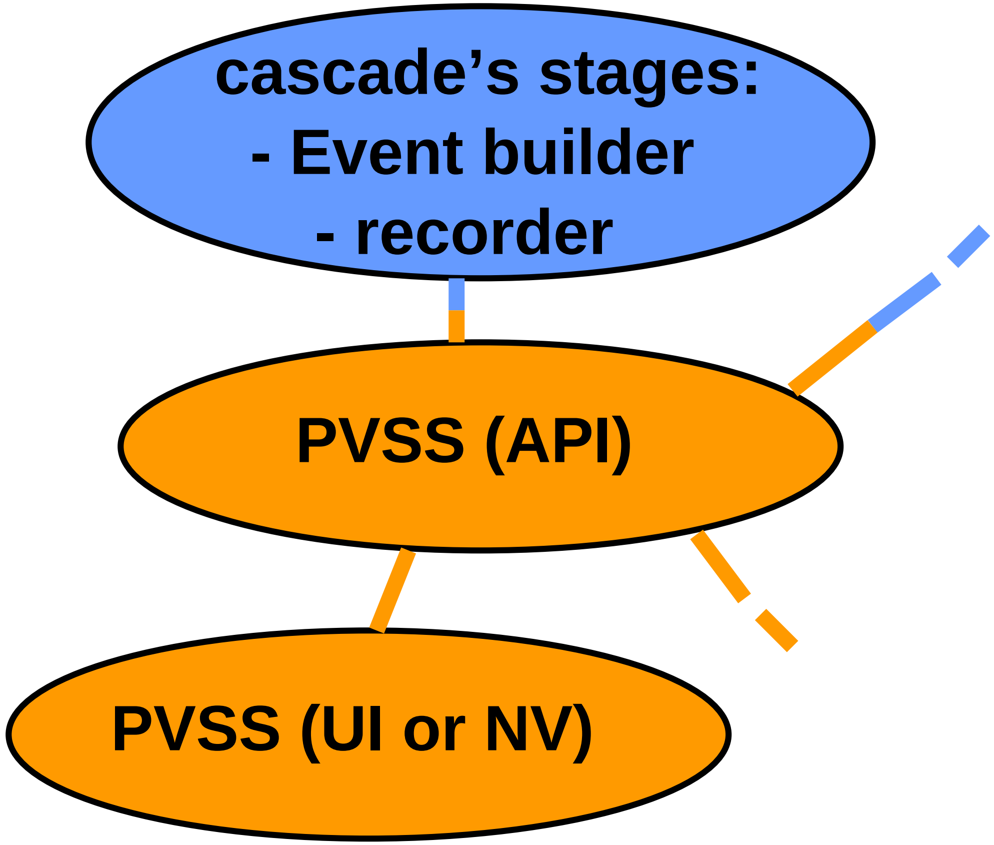
Hard (OS):

Soft:

VME: LynxOs

Central PC (Linux)

PC (Linux or Windows)



cascade's stages:
- Event builder
- recorder

PVSS (API)

PVSS (UI or NV)

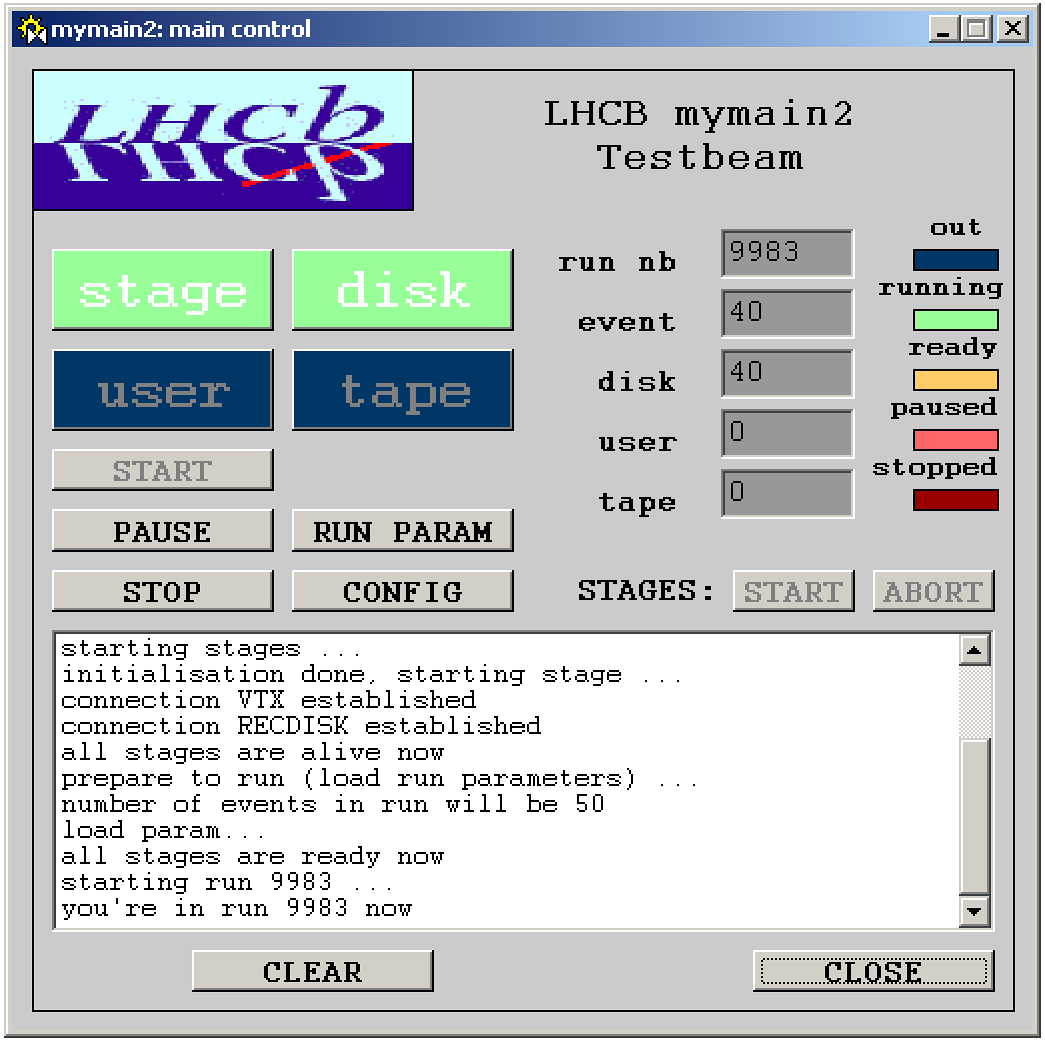
Details...

- API:
  - One per user, to avoid confusion and allow a like-standalone control,
  - Managed by a ctrl script (check status, restart if died, …),
  - Based on only one user dp, no interaction between different users,
  - Generate a complete log file,
  - Can be 'accessed' by many people from the same sub-detector by specifics panels.
- PVSS:
  - Managed by a crontab script:
    - clean kill / restart if problems.
  - Own 'emergency' status:
    - warns users, wait and restart all.

From the user point of view

- A single script allows user to access this system,
- All user’s process (display) run on the main control PC,
- All user configuration accessible by panels,
- No actions are link to panels (only the APIs really do something), to increase stability,
- Anybody can ‘have a look’ at the testbeam Run Control,
- Support is common.

Panels...



run nb 9983

event 40

disk 40

user 0

tape 0

```
starting stages ...
initialisation done, starting stage ...
connection VTX established
connection RECDISK established
all stages are alive now
prepare to run (load run parameters) ...
number of events in run will be 50
load param...
all stages are ready now
starting run 9983 ...
you're in run 9983 now
```

- Allow all actions & configuration
- All status display
- Info messages

New version

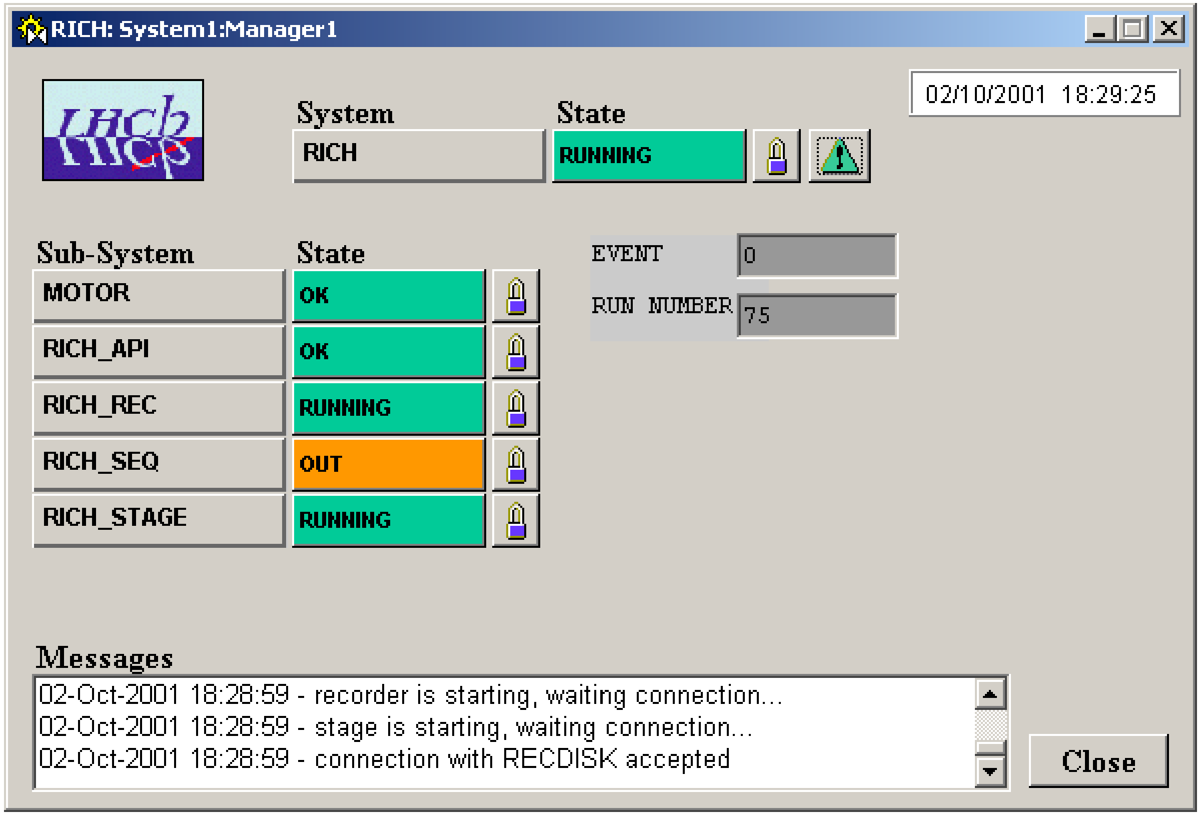
- PVSS works well as a standalone system this year,
- A PVSS Framework will be used for LHC experiments, and is now in his ‘development’ version.

  -> We are including it into the testbeam Run Control System.
- The new concepts are:
  - Notion of ‘Devices’,
  - Finite State Machines used for actions & status.

Already done

- The entire Cascade Run Control is now included:
  - Stages have become ‘devices’,
  - We used an ‘API’ device to control them.
- New utilities / devices have been implemented:
  - EMU utility (Cascade error message routing),
  - PC monitoring (processes, CPU, …),
  - Accelerator info display (based on Sascha Schmeling’s DIM server),
  - Calorimeter motors control (based on Vladimir Novikov’s OPC server).
- For the moment, both systems are running in the main control PC.

Panels...



02/10/2001 18:29:25

System	State
RICH	RUNNING

Sub-System	State
MOTOR	OK
RICH_API	OK
RICH_REC	RUNNING
RICH_SEQ	OUT
RICH_STAGE	RUNNING

EVENT 0

RUN NUMBER 75

Messages

```
02-Oct-2001 18:28:59 - recorder is starting, waiting connection...
02-Oct-2001 18:28:59 - stage is starting, waiting connection...
02-Oct-2001 18:28:59 - connection with RECDISK accepted
```

- **Mainly the same actions,**
- **Add / remove devices,**
- **Soon available.**

New planned devices

- High voltage control,
- Castor file utility control (at the moment managed by the API and castor scripts, which, after each run, ‘sends’ the file to Castor),
- ‘Online’ reconstruction (Gaudi) control (for example: if run stopped, start a Gaudi process with the run files as one of its arguments),
- Gas systems.

Conclusion / Demo

- PVSS has passed the ‘testbeam test’,
- An PVSS-API is flexible and can do almost everything,
- PVSS Framework allows to easy integrate many devices.