

Level-3 Build Infrastructure

Dave Waters for the Level-3 Group

- Level-3 Build Infrastructure – Overview
- Changes Since the Last Review
- Turn Around Times
- Validation
- Calibrations

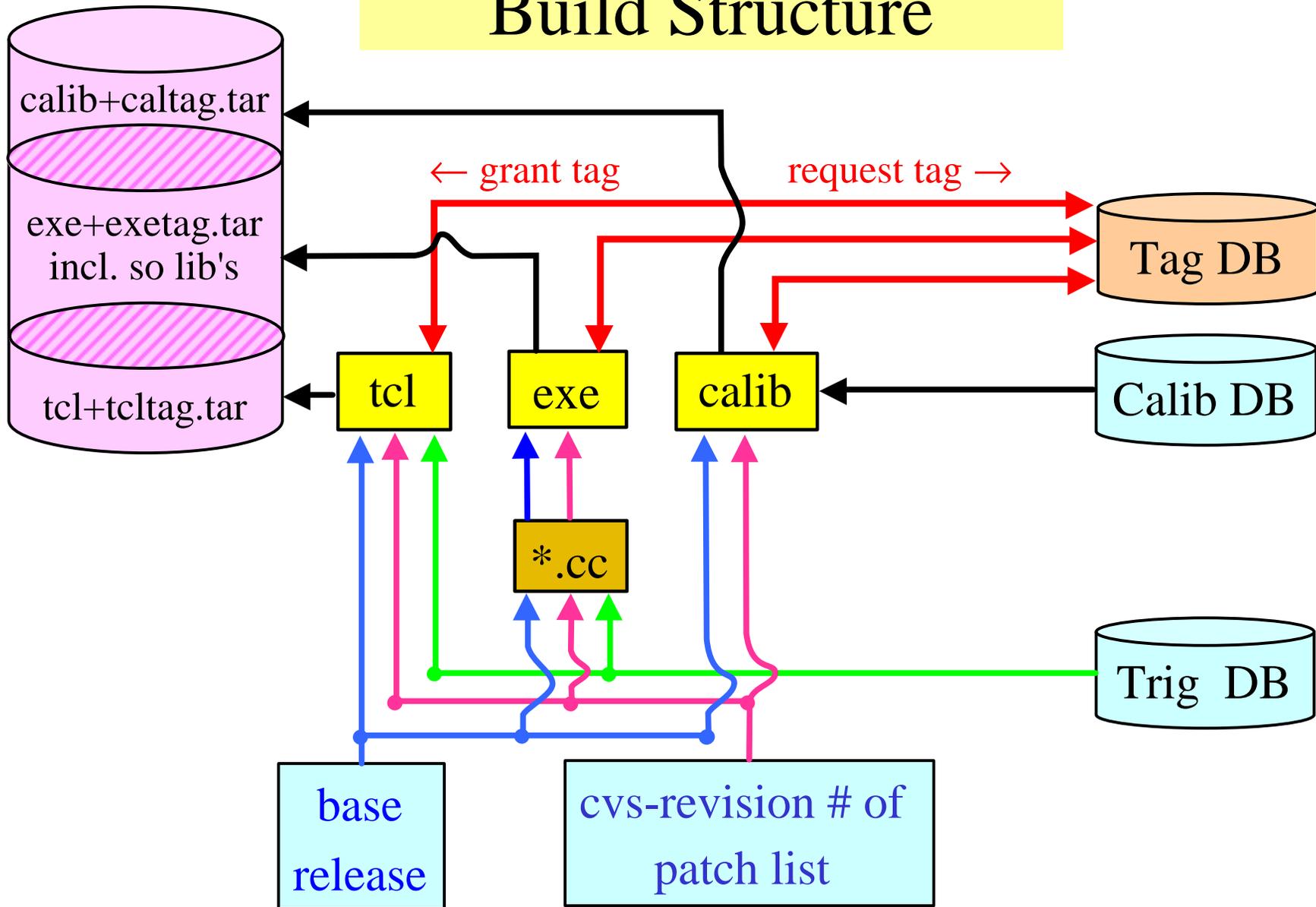
Level-3 Build Infrastructure - Overview

Reminder of Key Requirements :

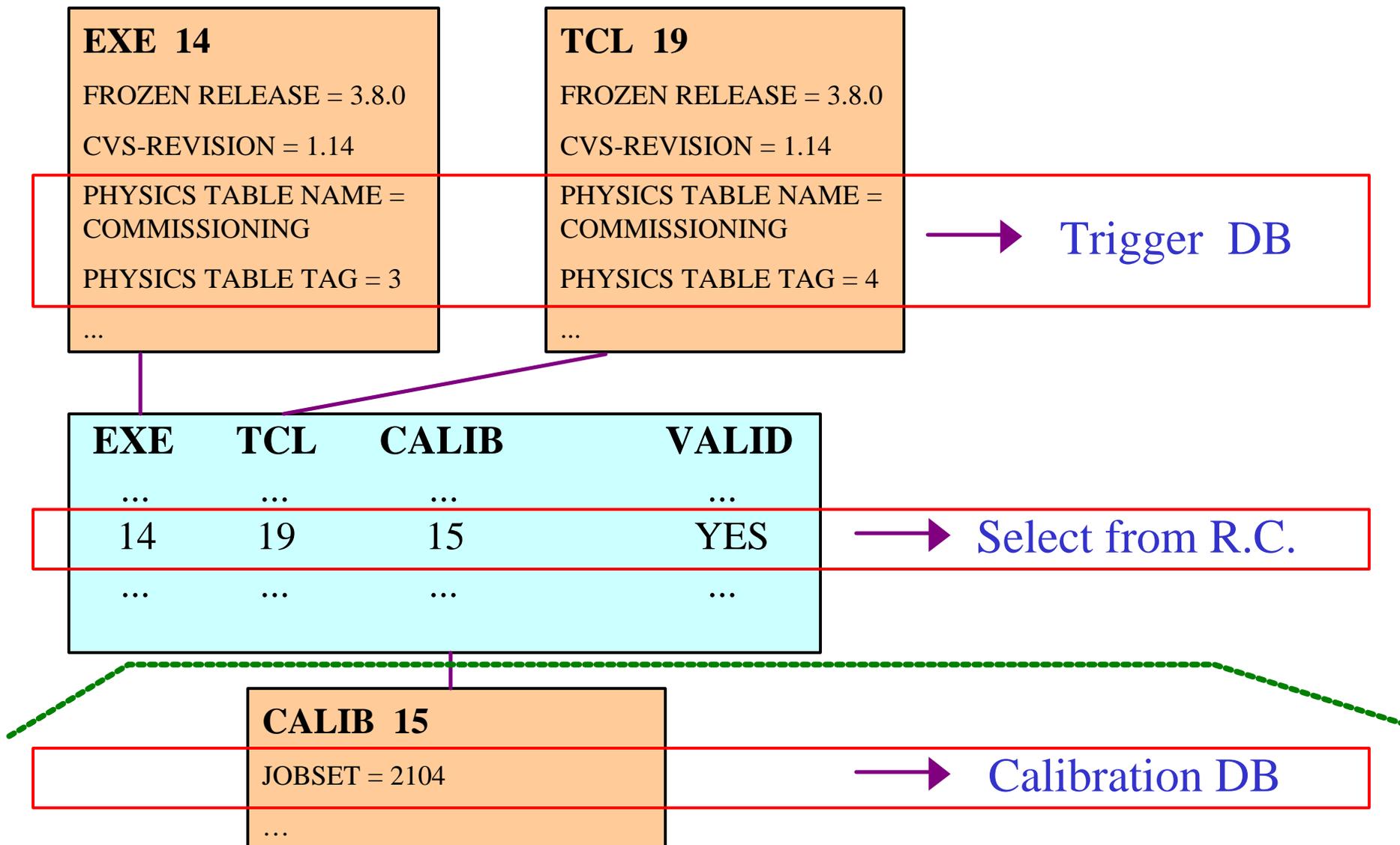
- Strict reproducibility. If **L3exe+47** crashes or is discovered to have some features, we can rebuild it and investigate these at any future time.
 - ▀ Reminder of patching and tagging scheme.
- Provision of executables, shared libraries, tcl-files and calibrations in a format suitable for distribution to the Level-3 worker nodes.
- Easy building of Level-3 executables by any collaboration member for testing and validation purposes.
- Procedures for validation of executables prior to their release.
- Procedures for reporting of errors and debugging of Level-3 core files.
- Provision of full documentation of these procedures :

[Level3/doc/level3.instructions](#)

Build Structure



Valid Tag Sets



Changes Since the Last Review (I)

- The Commissioning Run was extremely useful in shaking down the whole build procedure. Based on our experience we have made improvements in many areas :

- Build Infrastructure :
 - Use of Tag Tables in Trigger DB rather than Toy DB.
 - Automated entering of tag information during the Level-3 build.
 - **usable-exe-tag** mechanism for avoiding unnecessary executable builds. If a previously built executable can do the job (i.e. includes the necessary modules) it will be used.
 - Easier reproducibility of Level-3 executables – build scripts take optional arguments to specify previously used sets of patches.
 - Better automated tag-set validation procedures (see later slide).

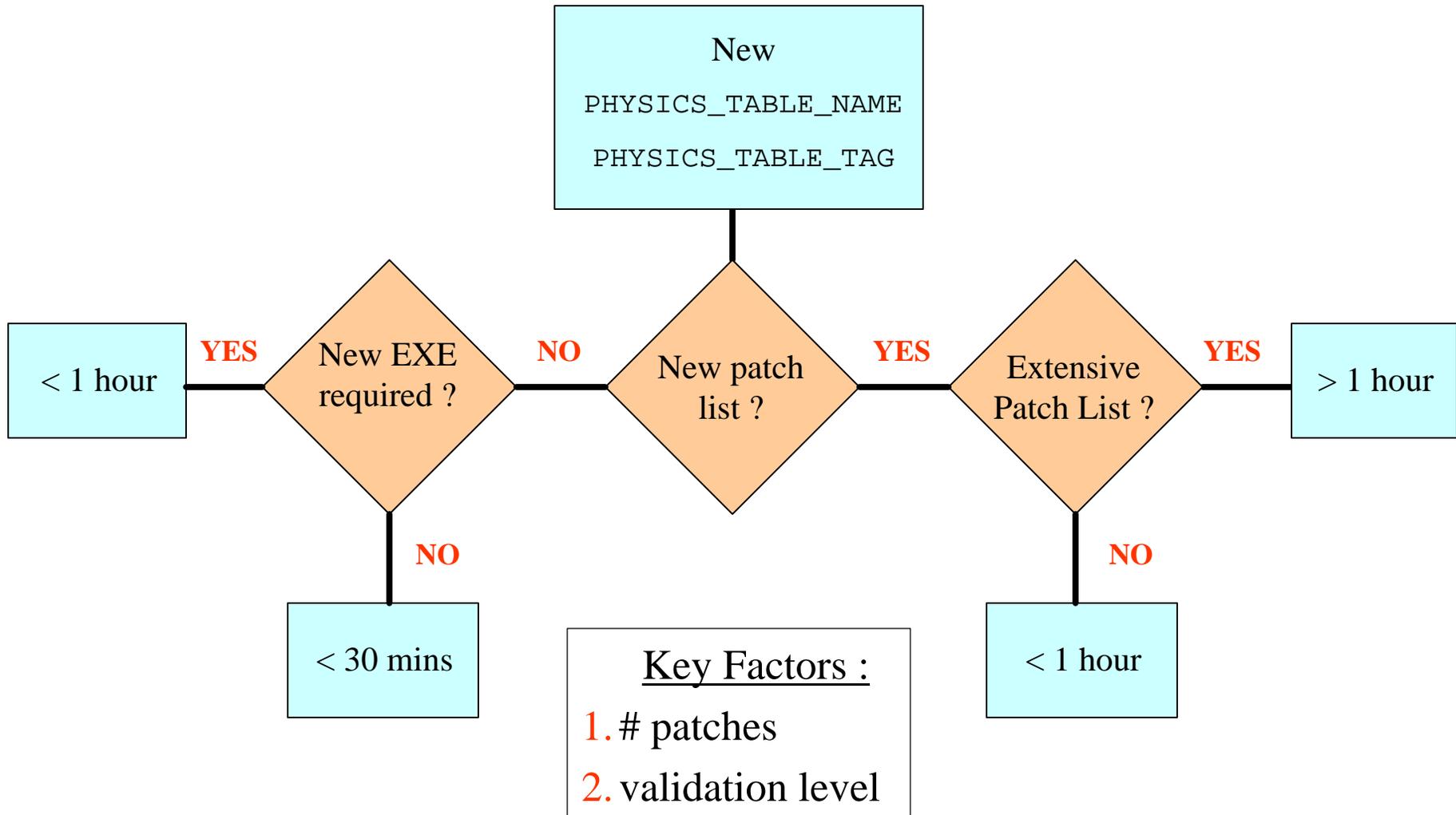
Changes Since the Last Review (II)

- Patching Scheme :
 - Use centrally maintained scripts (themselves based on the original Level-3 scripts) for the creation of patched test releases.
 - A separate patch list file for each frozen release :
`Release/patches/<base release>_level3.patches`
 - Easier sharing of patches between Level-3 and Consumers.
 - Easier dependency checking in case of patched header files.
- Build Platforms :
 - Moved from **oxpc01** to **b0dap30** . Plan to move to new Level-3 4-way server machine.
 - Level-3 builds on IRIX6 as well as Linux (main issue was **perl_dbd_oracle**). Enables use of IRIX code-testing tools and easier Level-3 code development by the collaboration.

Turn Around Times (I)

- Why does building Level-3 executables take time ?
 - We build executables and tcl files in test releases.
 - Building a test release based on patches to a frozen release takes time. In the CR we had instances of patched header files and consequently several large packages having to be rebuilt.
 - Consistency checking. A test release is thoroughly checked for local modifications and inconsistencies before new Level-3 tags are issued. This ensures true reproducibility.
 - Validation. A number of tests are performed on any newly generated executables and tcl files before they are released.
- ➡ *Assuming the trigger information is already in the Trigger DB :*

Turn-Around Times (II)



Validation

- A series of tests are performed on new exe-tcl combinations prior to their release for real running. The Level-3 executable is run in an offline mode over recent data and the following are checked :
 - Error reports or extraneous print-out by the Level-3 exe.
 - **Edm_ObjectLister** and **AC++Dump** from the base release can read the output. Event sizes and object content briefly checked.
 - Consumers from **cdfsoftb0** or **kor** can successfully run on the output file :
TrigMon, Xmon, Ymon, PhysMon, Stage0, evd
- Not checked every build (too time consuming) :
 - Memory leak status, validation histograms from each module.
- ➡ Fundamental trade-off :

Level of Validation \hat{U} Time Taken

Calibrations (I) – Flat File Generation

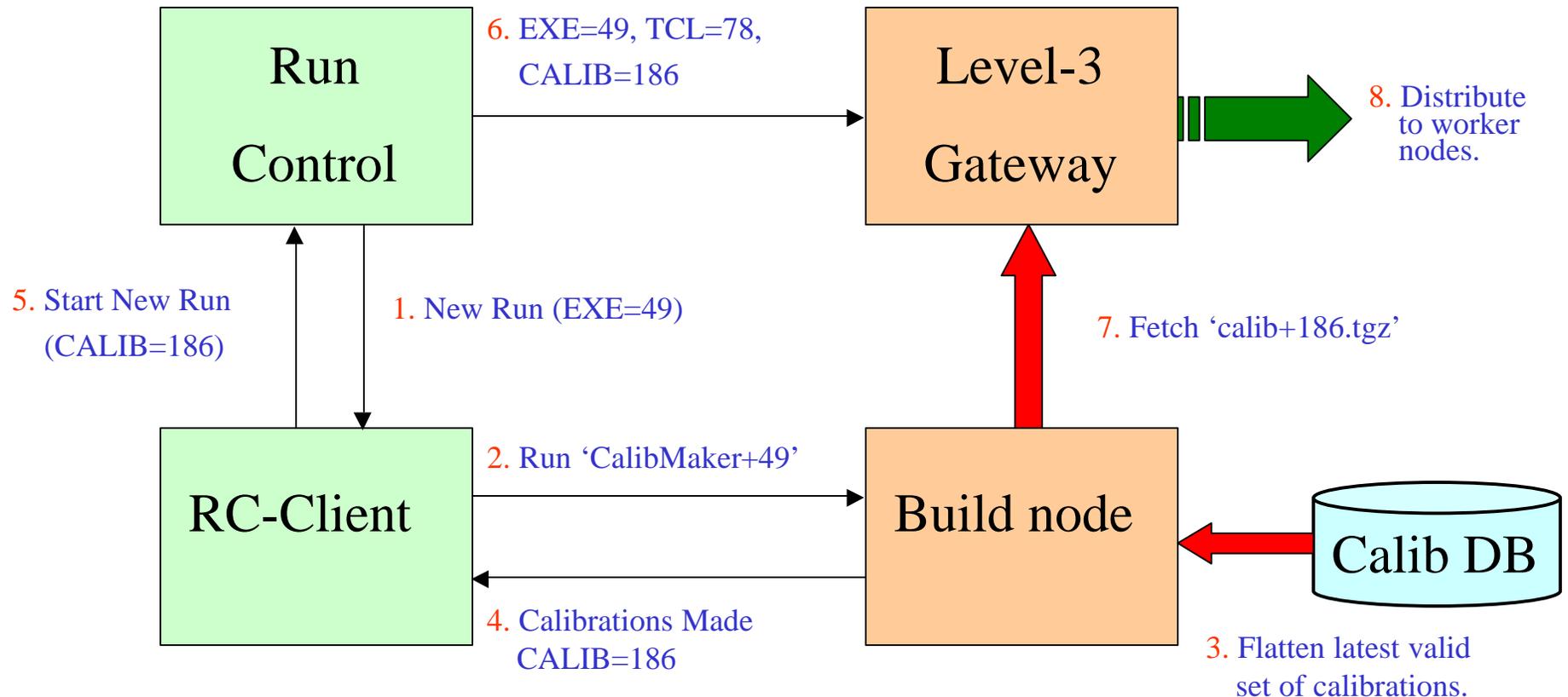
Basic Considerations :

- It is not feasible for all Level-3 worker nodes to be accessing the Oracle Calibrations Database online.
- Calibrations should be extracted from the database in flat file format and shipped to the worker nodes when required.
- This should be done in a fully tagged and reproducible way.
- This use of flat files should be completely transparent to calibration client code.

Commissioning Run :

- All the above was achieved during the commissioning run. However the calibrations were updated so infrequently that that the flat file building could be triggered by hand (**gmake Level3.tbin**).
- ➡ Main challenge for the real run is to dramatically increase the frequency with which flat files are generated from *validated* sets of calibrations.

Calibrations (II) – Cold Start



Calibrations (III) – Warm Start

Possible Solutions :

- I. Synchronous build : execute the same procedure as for the Cold Start. This is only feasible if the building of calibrations is fast enough.
 - Need to benchmark the extraction and flattening of large sets of calibrations.
- II. Asynchronous build (A) : query the Calibration DB for new “good” calibrations every few minutes. If any have appeared, build a new set of calibrations *using the build tools tagged with the current EXE tag.*
- III. Asynchronous build (B) : whenever new calibrations are flagged as “good”, Oracle broadcasts a message which is received by Run Control (or a client), triggering a preparatory build as in (II).

Action Points :

- Creation of calibration flat file building scripts and associated tools (Oxford – close to completion).
- Run Control client to coordinate calibration flat file builds and provide requisite Run Control messages (Rochester).
- Investigate Oracle message broadcasting (Jim K. & co.).

Summary and Conclusions

- We learned a lot during the Commissioning Run about :
 - the logistics of building executables from patched frozen releases.
 - the mechanisms needed for this to be truly and easily reproducible.
 - the importance of validation and testing of new executables.
- We have implemented a series of modifications to our build procedure in the light of this experience.
- The creation of reproducible and validated Level-3 executables and tcl files does take time. This is determined largely by the size of the patch lists and the level of validation required.
- We have a set of validation tests that will catch 90% of problems with Level-3 executables. We will never catch them all.
- Procedures for building calibration flat files at Cold Start have been designed. Implementation is well underway. Proposals exist for Warm Start procedures and are being investigated.