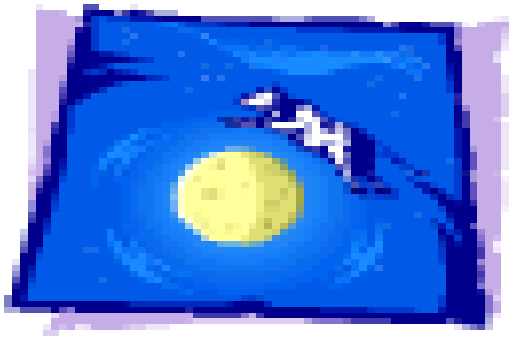


Building Configurations for Initial Charge-injection Calibration Runs

J. Bogart

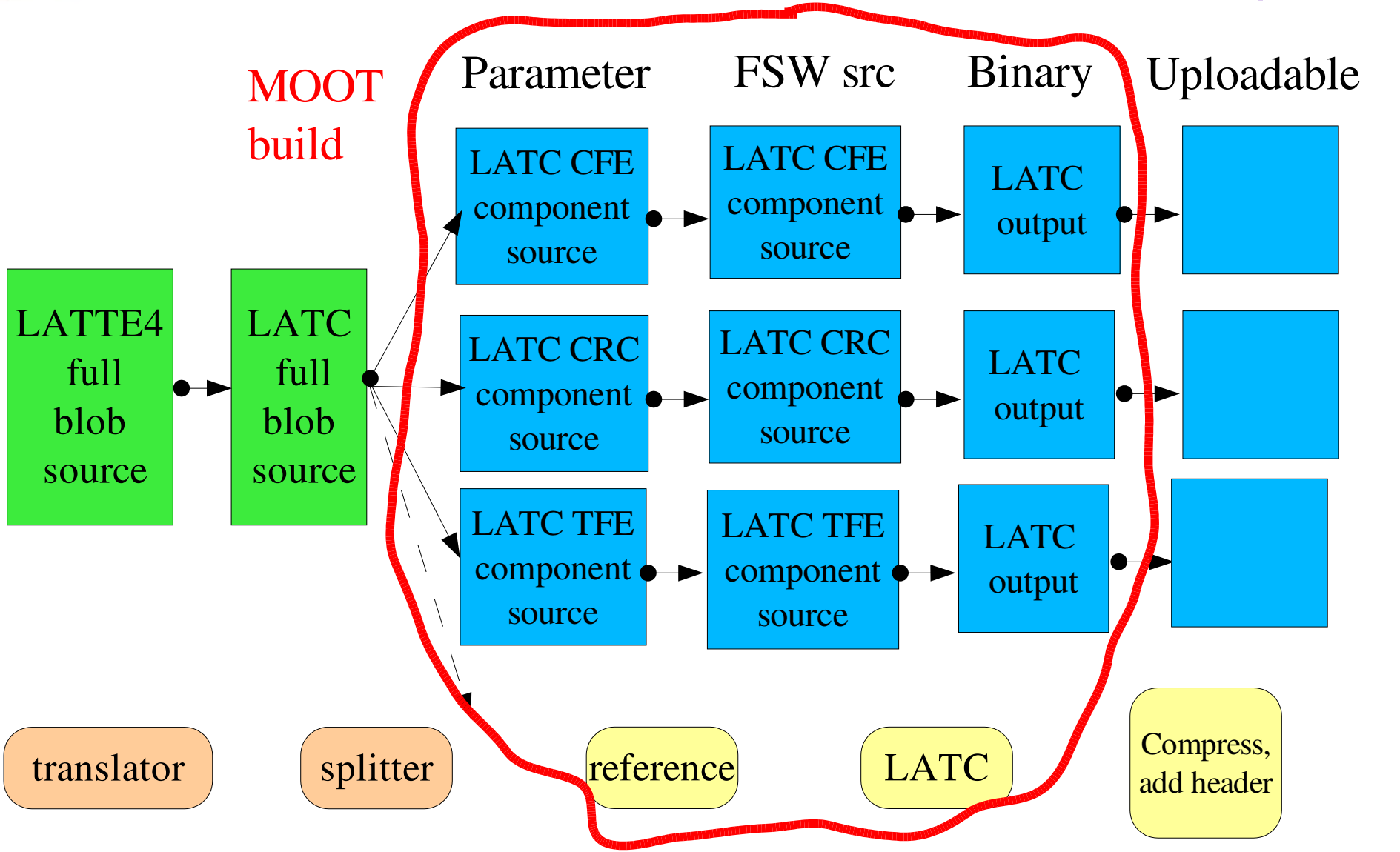
Conceptual Stages Involving MOOT

- Define and create a configuration. End product is collection of binaries.
- Determine destination, upload.
- Select among uploaded configurations; run.
- Analyze data

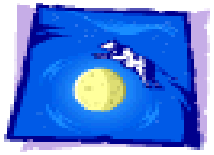
Config Contents

- Input to LATC
- Input to LCI for calibration runs
- “Ignore” file

Pipeline for Calib Runs

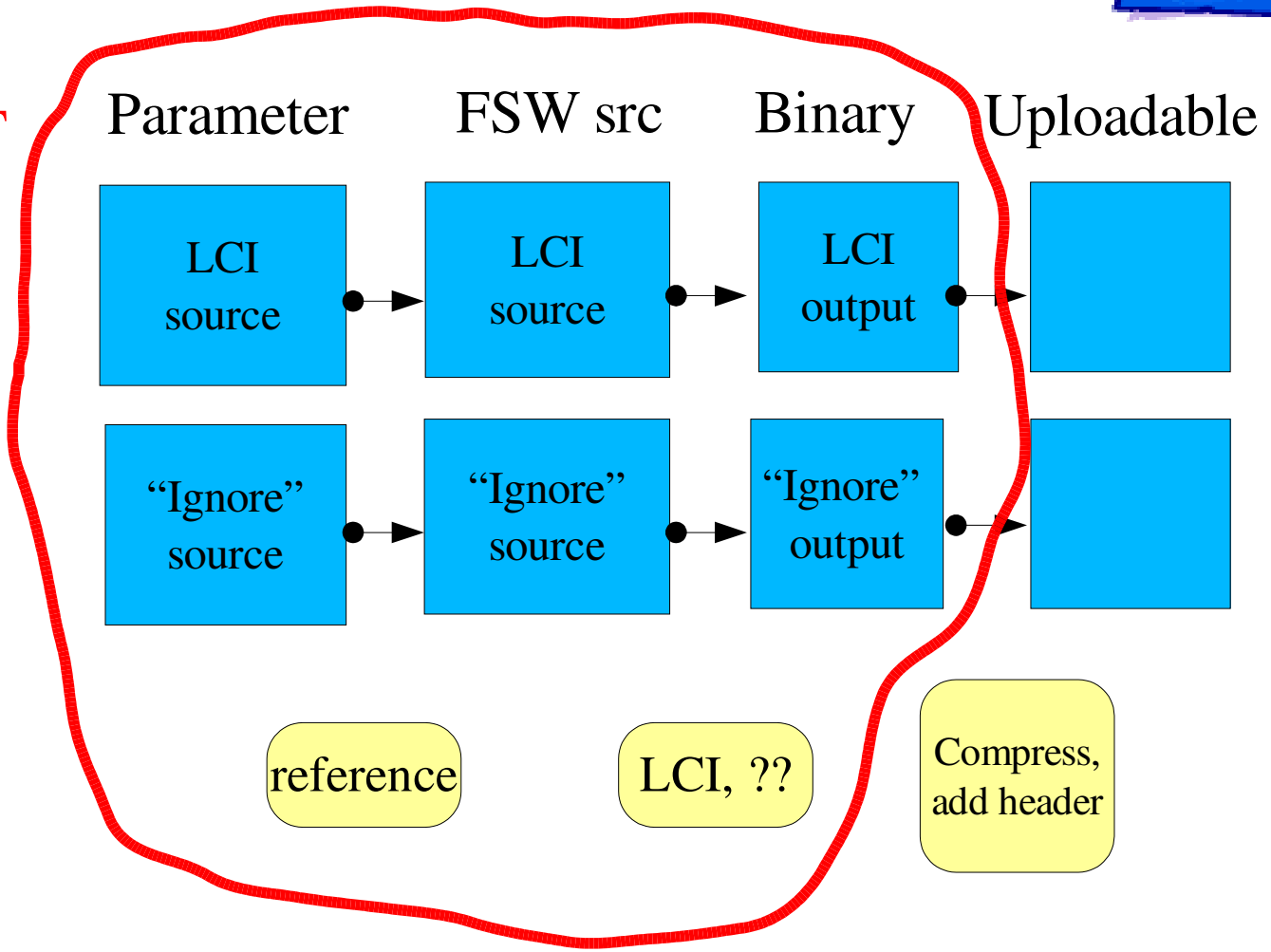


Pipeline for Calib Runs



MOOT build

Handle non-LATC input similarly, but simpler



Status

Excerpt from test program:

```
MOOT::MoodConnection* moodCon =
   new MOOT::MoodConnection(host, user, pw, dbname);

 MOOT::MootBuild moot(moodCon);

 std::vector<ParmDescrip> pd;
 pd.reserve(3);

 pd.push_back(ParmDescrip("$(MOOTROOT)/data/daq_aem.xml", "AEM_parm"));
 pd.push_back(ParmDescrip("$(MOOTROOT)/data/daq_gem.xml", "GEM_parm"));
 pd.push_back(ParmDescrip("$(MOOTROOT)/data/daq_dft.xml", "DFT_parm"));

 unsigned configKey = moot.simpleConfig(pd, "AnotherFirstConfig",
                                        MOOT::MootBuild::MODE_data,
                                        "emptyAlg");
```

..results in creation of many entries in dbs, including a row in the Configs table, and three binary files