Goals for this meeting?

- Create a to do list that addresses concerns from all I&T Depts with the intent to help prioritize the work
  - SVAC needs to have some information available to be able to draw the Collaboration into the Data Analysis effort
- The aim is to focus on tower A only
  - Discuss implementation of all scripts needed for I&T
    - Implementation affects Online, IFCT, Particle Tests and SVAC
  - trade-offs to be considered
    - Minimize effect on LATTE releases
    - Minimize procedure rewriting and operator errors
    - Minimize effects on offline pipeline
    - Minimize burden on people analyzing the data

What are the constraints?

- Constraints to be addressed for implementation trade-offs
  - Minimize effect on LATTE
    - Online needs to move on into LATTE 5
  - Minimize procedure rewriting and operator errors
    - IFCT needs to finalize the write-ups (procedures, AIDS)
  - Minimize effects on offline pipeline
    - The pipeline is not very stable
    - SVAC suffers from the fact that we have to support subsystems and I&T (just like the Online group does)
      - » Adding anything in the pipeline and or its products will be big deal for quite a while
  - Minimize burden on people analyzing the data
    - People outside I&T are not expected to have the same knowledge as we do and they are essential for the data analysis
    - Elog is not optimized for data analysis queries

Data Analysis

- Data Analysis includes analysis of data from
  - all tests
    - Aliveness
    - LPT
    - CPT
    - Trigger tests
    - E2E
    - SVAC
  - Different particle types
    - Charge injection
    - Cosmics
    - Photons
    - Other
  - Different hardware types
    - Single tower outside and inside grid
    - Single and multiple towers in the grid
    - Engineering Model
    - Standalone TKR, CAL and ACD

Where do we find information?

- To analyze data produced by these tests we need to know
  - Test Description
    - Pat's Document LAT-TD-002370
      - » Need to know what is being done for trigger tests and whether that matches the description in LAT-TD-002370
    - Gary's Document LAT-TD-004136
      - » Matrix is not convenient for a general user.
      - » SVAC will write it in a form that is easier to read, but how do we keep track of changes?
  - Test Configuration
    - Driven by scripts/LATTE with and without operator inputs
      - » There had been some issues with subsystem tests only since their tags do not necessarily conform with I&T ones (more on this later)
  - Instrument configuration
    - Parsed from LATTE output files at the end of the run
    - SVAC would like to have list of which external information (trigger rates, etc) is being parsed into the rcReport.out and what are the names used.

Questions concerning scripts

- Should have a discussion on the choice of algorithms for some of the tests involving I&T and subsystems
  - For example CALIBTHRES and CAL muTrig are similar. Online is implementing the former and SVAC asked for the latter. Are we going to use both?
    - What is driving the decision?
  - Are there differences between the timing in scripts from the Trigger groups compared to those from the CAL and TKR?
    - May have effects when comparing the results
      - » Cal mu Shape is used by CAL for timing tests. How was the information in mu Shape transferred into the Trigger group scripts that I&T will use later?
  - Would like to understand what is the process for changing settings inside a script for the E2E and SVAC tests
    - Some settings for the E2E and SVAC scripts depend on results from other scripts (e.g trigger)
    - How are we reviewing that and implementing the new settings?
- How do we verify that once a register is set that the output agrees with the input
  - Need to verify all settings for the E2E and SVAC tables worked correctly through the snapshot files
  - Do all scripts produce PASS/FAIL reports that are logged and archived by the pipeline?
  - There are tracker data taking scripts producing errors (they may not be a big deal) but we need to know how we are handling those during I&T
    - this will affect the PASS/FAIL criteria for E2E and SVAC tests

Procedures for Data Taking

- Operator inputs
  - Operator inputs and information coded in the scripts are used for queries to search data
    - As part of the validation of scripts SVAC would like to review the contents of scripts and outputs since pipeline depends on it
      - » For example the pipeline will NOT reconstruct EM data if a particular tag is not set correctly.
      - » We have find inconsistencies between values within a report. If a value is not used it should not appear in the report
      - » There is a need for dummy tags for development of scripts, we already created dummy cosmic runs for tests (Online is addressing this now)
      - » Need to review all test suites to ensure that they provide all necessary tags for offline processing
- Connection with subsystems
  - Subsystems must be informed of the rules for the pipeline chain in case they want to benefit from it.
    - Hopefully the I&T rules will not make their life so miserable since it is to our benefit to be able to compare data
    - TKR provided runID in different format than I&T, and this affected the pipeline, so runs were not processed.

Runs Database

- SVAC would like to clean it up the ORACLE elog because users are confused with all that information.
- We propose to use the
  - Top of page for people doing data analysis
  - Bottom of page for I&T and subsystem experts

- Offline tag
  - Can we add an offline tag in the scripts to minimize logic for the queries?
    - offline == "recon“, ‘norecon”, “nodogi”
- Serial number tag
  - EM has to be added to all scripts for EM
    - So that EM data can be reconstructed with the right code
    - There will be no search by pipeline on the phase
    - EM2Cal', 'MiniTkr' , 'EM2'. These are the "unique" identifiers and nothing else, right?
  - FM tags have to be added to all scripts
    - So that FM data can be reconstructed with the right code
    - There will be no search by the pipeline on instrument type to reconstruct data
      - » However instrument type is there for backward compatibility and is growing so tag must be recorded so that reports are accurate
  - MiniCal is not reconstructed

CAL Calibrations

- Official CAL calibrations will be stored in SAS database and taken when Tower is inside the Flight Grid (of course we will exercise them outside the grid)
  - muTrig suite or I&T equivalent to set thresholds (muons)
    - Flight settings
    - Muon settings
      - » High energy muon gain (4 range, auto range, FLE 100 MeV)
  - calibGen suite (charge injection)
    - Flight settings
    - High energy muon gain (4 range, auto range, FLE 100 MeV)
  - calibDAC suite (charge injection)
    - FLE,FHE,ULD,LAC operational thresholds
  - SVAC B-10
    - Main cosmic ray for partially populated LAT with CAL high energy muon gain
  - SVAC B-2
    - Main cosmic ray for partially populated LAT with CAL flight settings

TKR Calibrations

- **Official TKR calibrations will be stored in SAS database and taken when Tower is inside the Flight Grid (of course we will exercise them outside the grid)**
  - **Charge injections**
    - **TE601 - TkrThresholdCal.py**
    - **TE602 - TkrTotGain.py**
    - **TE604 - TkrThrDispersion.py**
  - **TKR subsystem data taking script is replaced by those SVAC scripts shown below during the I&T phase**
    - **TE603 - TkrDataTaking.py**
  - **SVAC B-10**
    - **Main cosmic ray for partially populated LAT with CAL high energy muon gain**
  - **SVAC B-2**
    - **Main cosmic ray for partially populated LAT with CAL flight settings**

Back up

Instrument type vs. Serial no. (1)

- At the moment, subsystem tests put serial no. info into both instrument type and serial no tag in the rcReport
  - For CAL:
    - Instrument type: CALFM103
    - In rcReport, <SerialNos>('calinstrument':'FM103')<SerialNos>
  - For TKR:
    - Instrument type: TKRFMA
    - In rcReport, <SerialNos>('tkr':'TkrFMA')<SerialNos>
- User can use eLog to query data in both ways:
  - Select from insturment type drop down menu
  - Type in TKR/CAL serial no.(e.g. FM103 or FMA)

Instrument type vs. Serial no. (2)

- It may be better to store serial no. info only in the serial no tag in the rcReport for the following reasons:
  - Operator has to manually select the instrument type from a drop down menu in run control while the serial no. is written to the rcReport by script. Less human intervention reduces risk of potential errors.
  - Redundant info increases risk of potential errors.
  - The drop down menu for the instrument type will grow very long as new hardware comes in.

CAL CPT

LAT-MD-01370-02 CAL Functional Test Definition Page 7 of 15

The CPT shall be comprised of the following test procedures:

1. CALU_INIT: Initialize the Calorimeter.
2. CALF_EXR_P01: Exercise registers.
3. (optional) Margin Test: Test voltage and frequency margins.
4. CALU_INIT: Initialize the Calorimeter.
5. CALF_PEDESTALS_CI: Compute pedestals.
6. CALU_INIT: Redefine pedestal file.
7. CALF_MU_OPTICAL: Collect cosmic muon data under CAL self-trigger.
8. CALF_MU_TREND: Analyze muon data for changes in relative optical performance.
9. CALF_SHP_P01: Calibrate slow shaper and determine optimal charge-injection Tack time delay under Timed
10. CALF_GAIN_P01: Calibrate relative electronic gains with charge injection.
11. CALU_INIT: Redefine relative gain table.
12. CALU_COLLECT_CI_SINGLEX16: Collect charge injection data (for front-end linearity test).
13. CALF_ADC_P02: Analyze data from charge injection to determine front-end integral non-linearity and noise.
14. CALU_COLLECT_CI: Collect charge-injection data (for front-end droop test).
15. CALF_ADC_P05: Evaluate front-end droop.
16. CALU_COLLECT_CI_SINGLEX16: Collect charge injection data (for front-end calibration).
17. CALF_TRG_P01: Test CAL-LO and CAL-HI trigger enable/disable with charge injection.
18. CALF_TRG_P04: Characterize FLE and FHE trigger times.
19. CALF_TRG_P03: Characterize FLE and FHE DAC settings with charge injection.
20. CALF_SUPP_P01: Characterize LAC DAC settings with charge injection.
21. CALF_SUPP_P02: Determine lowest effective LAC DAC settings.
22. CALU_INIT: Redefine LAC table, setting LAC to its lowest effective setting.
23. CALF_RNG_P01: Characterize ULD DAC settings with charge injection.
24. CALF_OVR_P01: Evaluate overload recovery with charge injection.
25. CALF_DTM_P01: Estimate event dead-time with charge injection.

CAL LPT

LAT-MD-01370-02 CAL Functional Test Definition Page 8 of 15

The LPT shall be comprised of the following test procedures:

1. CALU_INIT: Initialize the Calorimeter.
2. CALF_EXR_P01: Exercise registers.
3. (optional) Margin Test: Test voltage and frequency margins.
4. CALU_INIT: Initialize the Calorimeter.
5. CALF_PEDESTALS_CI: Compute pedestals.
6. CALU_INIT: Redefine pedestal file.
7. CALF_MU_OPTICAL: Collect cosmic muon data under CAL self-trigger.
8. CALF_MU_TREND: Analyze muon data for changes in relative optical performance.