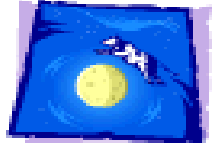


MOOT

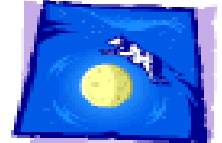
Configuration Tracking, SAS Perspective



Goals

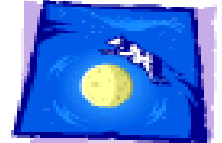


- **Reconstruct the configuration used for a run**
(only possible if MOOT participates in creation and uploading)
- Avoid creating and uploading redundant information
- Quickly implement something which satisfies above for upcoming calibration runs and smoothly evolves to handle test data runs and, ultimately, flight.



Conceptual Stages Involving MOOT

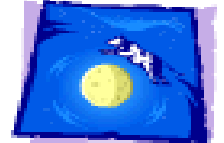
- Define and create a configuration. End product is collection of binaries.
- Determine destination, upload.
- Select among uploaded configurations; run.
- **Analyze data**



Implementation

- Identify information needed to work backwards to intent*
- Design suitable database structure
- Implement services needed at each stage

***Intent** is expressed in application user terms; e.g., “use 150 MeV calorimeter thresholds”



Config Contents

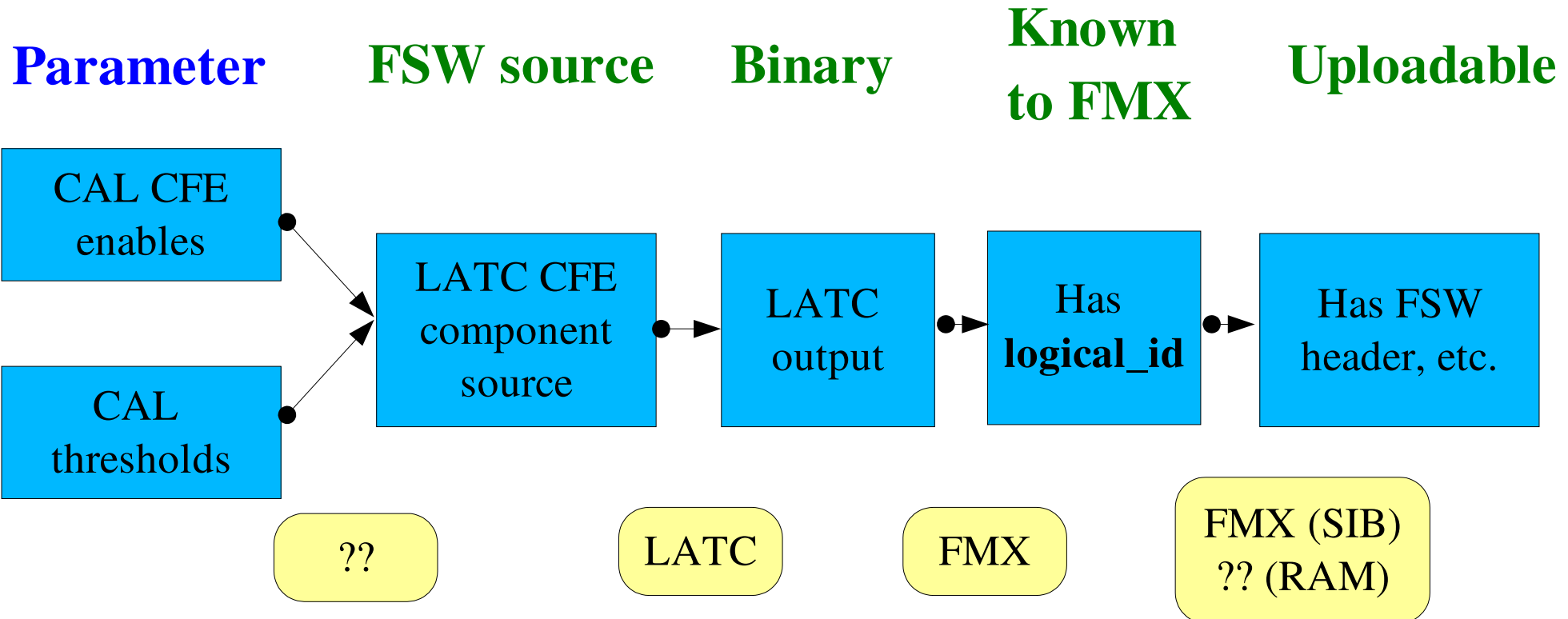
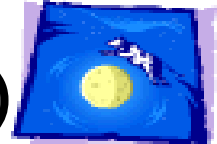
- Input to LATC¹
- Input to LCI² for calibration runs
- Input to Filter, etc., for data runs
- For now, at least, MOOT will not keep track of thermal parameters, power-on state, etc. These can be added later if needed.
- MOOT will not keep track of code modules.

¹ **LATC** is a Flight Software facility which handles most of the LAT register settings, including such things as thresholds, channel masks, trigger configuration,... User input is xml file.

² **LCI** is a Flight Software facility which supports definition and execution of calibration runs. User input is xml file.



Config Creation Pipeline (MOOT view)

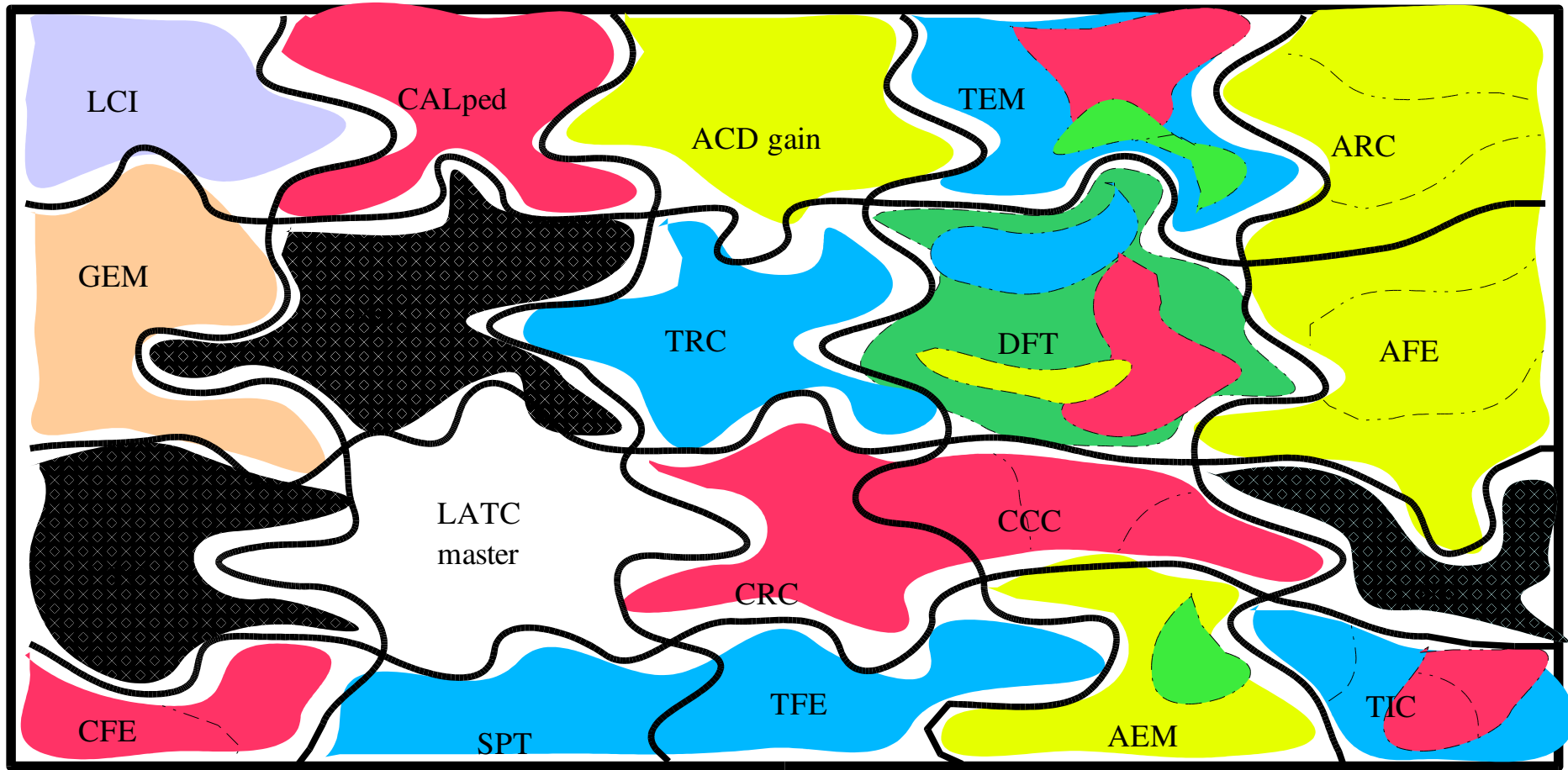
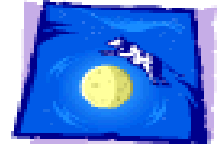


Only first stage may be many-to-one; all others are one-to-one.
 Imagine parallel pipeline for each LATC component and also for other kinds of configuration.

Yellow boxes indicate mechanism to get from stage to stage.



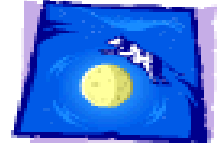
FSW Files/Delegates/Parameters Overlay



Extra dotted lines indicate how information from different parameters may be merged into a single FSW file. Note that different colors (indicating different delegates) are always separated by lines.



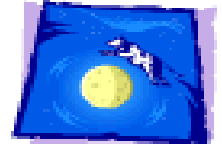
Database design



- Primary tables to represent **Parameters**, **FSW inputs**, and **Configs**.
- Secondary tables to relate Parameter file instances to FSW input instances; FSW input instances to **Configs**.
- Secondary tables to describe classes of parameter files and classes of FSW inputs.



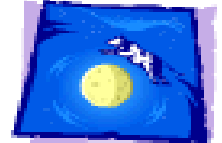
Primary Tables



Tables	
Configs	
Parameters	
Parameter_class	
FSW_inputs	
FSW_class	
Columns	
<input checked="" type="checkbox"/>	parm_key
<input checked="" type="checkbox"/>	class_fk
<input checked="" type="checkbox"/>	source
<input checked="" type="checkbox"/>	source_fmt
<input checked="" type="checkbox"/>	quality
<input checked="" type="checkbox"/>	flavor
<input checked="" type="checkbox"/>	archive
<input checked="" type="checkbox"/>	vstart
<input checked="" type="checkbox"/>	vend
<input checked="" type="checkbox"/>	instrument
<input checked="" type="checkbox"/>	precursor
<input checked="" type="checkbox"/>	description
<input checked="" type="checkbox"/>	checksum
<input checked="" type="checkbox"/>	size
<input checked="" type="checkbox"/>	creation_time
<input checked="" type="checkbox"/>	creator

Tables	
Configs	
Parameters	
Parameter_class	
FSW_inputs	
FSW_class	
Columns	
<input checked="" type="checkbox"/>	FSW_input_key
<input checked="" type="checkbox"/>	class_fk
<input checked="" type="checkbox"/>	FSW_id
<input checked="" type="checkbox"/>	status
<input checked="" type="checkbox"/>	source
<input checked="" type="checkbox"/>	source_fmt
<input checked="" type="checkbox"/>	flavor
<input checked="" type="checkbox"/>	archive
<input checked="" type="checkbox"/>	description
<input checked="" type="checkbox"/>	sib_dest
<input checked="" type="checkbox"/>	ram_dest
<input checked="" type="checkbox"/>	checksum
<input checked="" type="checkbox"/>	size
<input checked="" type="checkbox"/>	creation_time
<input checked="" type="checkbox"/>	creator

Tables	
Configs	
Parameters	
Parameter_class	
FSW_inputs	
FSW_class	
Columns	
<input checked="" type="checkbox"/>	config_key
<input checked="" type="checkbox"/>	name
<input checked="" type="checkbox"/>	status
<input checked="" type="checkbox"/>	mode
<input checked="" type="checkbox"/>	algorithm
<input checked="" type="checkbox"/>	alg_step
<input checked="" type="checkbox"/>	instrument
<input checked="" type="checkbox"/>	description
<input checked="" type="checkbox"/>	creation_request_time
<input checked="" type="checkbox"/>	creation_end_time
<input checked="" type="checkbox"/>	creator



Recovering Configuration

- Data stream contains **logical_ids** of config files in use
- Look them up in **FSW_inputs** table
- For each, find one or more related Parameter files by querying **FSW_to_Parameter** table.
- Look up characteristics of found files in **Parameters** table.