The container interface is incompatible with the pipeline interface

What is the problem and why do we care?

- At job startup, the pipeline interface is invoked with the command: `bash pipeline_wrapper` (see for example: /sdf/data/fermi/n/u41/L1/logs/DEV/AAA-testVerifyS3df-no-wrap/5.9/verify/000/slurm_pilot)
- If we invoke the container with **<job executable="${preamble} ${container_wrap} python --version" batchOptions="${extra}" />** (as in Brian's example), the pipeline interface defined in `pipeline_wrapper` is NOT available inside the container
- **This is a showstopper:** L1Proc and other complex tasks use extensively the pipeline interface defined in `pipeline_wrapper`, to set/read pipeline variables, to create substreams, etc.

Possible solution (suggested by Brian on 20 Sep 2023)

- Write a lightweight interface to the pipeline environment, to obviate the need for invoking pipelineSet and pipelineCreateStream. Example here.
- **Implemented in GPLTools as pipeline.py. This seems to work as of 25 Jan 2024**

How we isolated and documented the issue

- Task AAA-testVerifyS3df-wrapper (see xml here) loads the container with `${container_wrap}` at line 50:
  - **<job executable="${preamble} ${container_wrap} ${python27} ${L1ProcROOT}/verify.py" batchOptions="${extra}" />**
- Thereby losing access to the pipeline interface. For example, we try accessing `pipelineSet` and fail (full log here, error at line 179):
  - ```
    pipelineSet=
    which: no pipelineSet in (/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin)
    ```

- Task AAA-testVerifyS3df-no-wrap (see xml here) defines the container, but does not invoke it at execution (line 50):
  - **<job executable="${python27} ${L1ProcROOT}/verify.py" batchOptions="${extra}" />**
  - (notably, the python errors in lines 173-213 of the logfile confirm that the container is not loaded correctly in the executable)
- And still has access to the pipeline interface. For example, we try accessing `pipelineSet` and succeed (full log here, lines 221-225):
  - ```
    pipelineSet=
    pipelineSet ()
    {
        echo "Pipeline.$1: $2" >> ${PIPELINE_SUMMARY}
    }
    ```