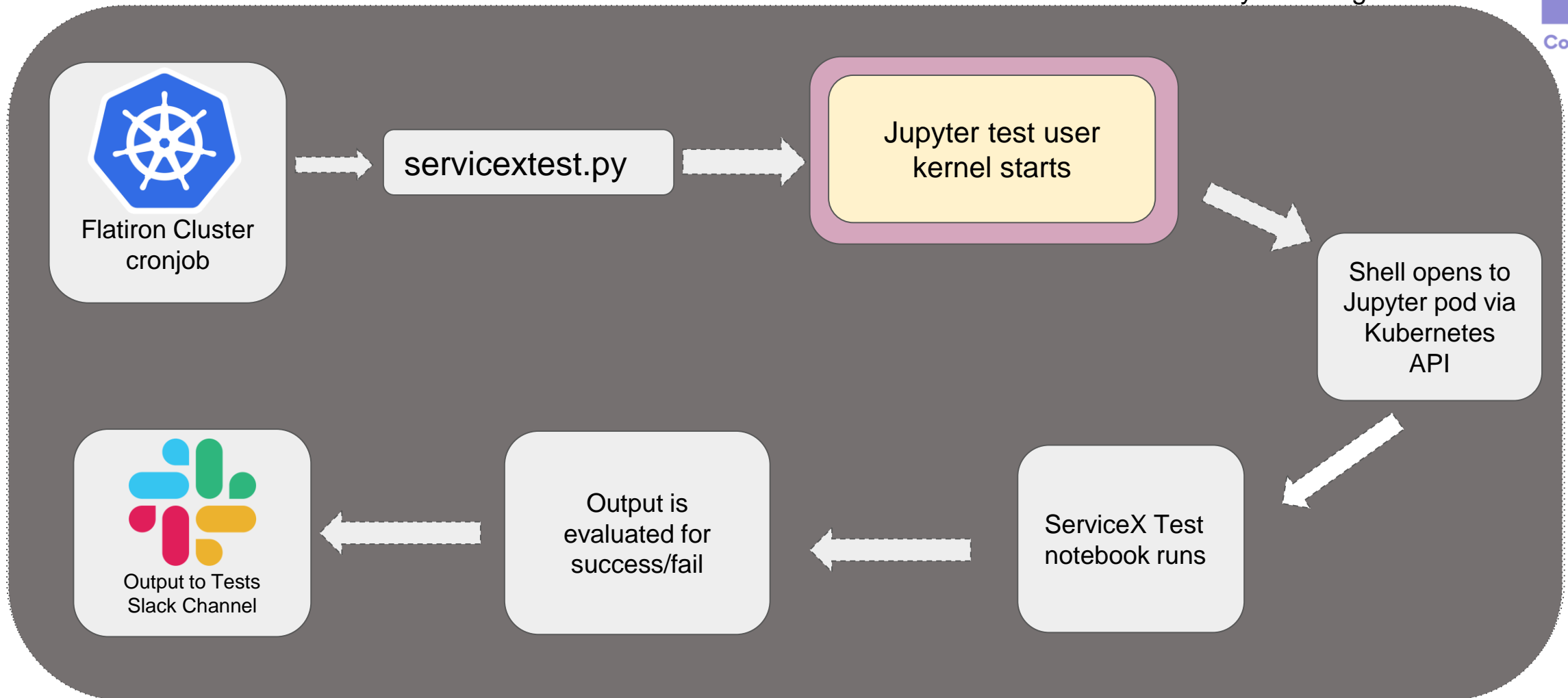


Coffea-Casa: Automated monitoring of ServiceX

Coffea-casa Analysis Facility

Coffea-casa facility @ UNL is co-located at U.S.CMS Tier-2 at University Nebraska-Lincoln and other instance is co-located at U.S.ATLAS Tier-3 at University UChicago



Testing Workflow Overview

- A Kubernetes cronjob builds a Python container and runs *servicetest.py*, which will run a test notebook on <https://coffea.casa/> analysis facility () on a daily schedule.
- The script will first use the *Jupyterhub API* to ensure that the test user has a Jupyter server spawned before running the ServiceX tests.
- After the pod has been successfully spawned the name/path of the notebook is passed into *servicetest.py* as a *sys arg*.
- *servicetest.py* opens a shell into the newly spawned test Jupyter server and runs the specified notebook.
- The output of the executed notebook is then evaluated by *servicetest.py*
- Test failure is determined by the "ename" error being present in the raw JSON notebook output. Test success is determined by the "text" field being present in the raw notebook JSON output.
- A message containing the *Failure/Success message* is outputted to the iris-hep coffea-casa tests slack channel.

Next Steps

- Extend tests to cover the CMS data access functionality through Xcache + ServiceX (as well simply through XCache) and to test Dask+Condor scaling
 - *The goal is to cover all coffea-casa components/dependencies* through multiple test combinations.
- Add the automatic discovery of *pytests in various repositories*, and add the AGC notebook as a nightly smoke test.
 - Add the automatic generation of cron jobs and config map pairs based on pytests in the repositories.
- Extend functionality with a test result visualisation through dashboard.