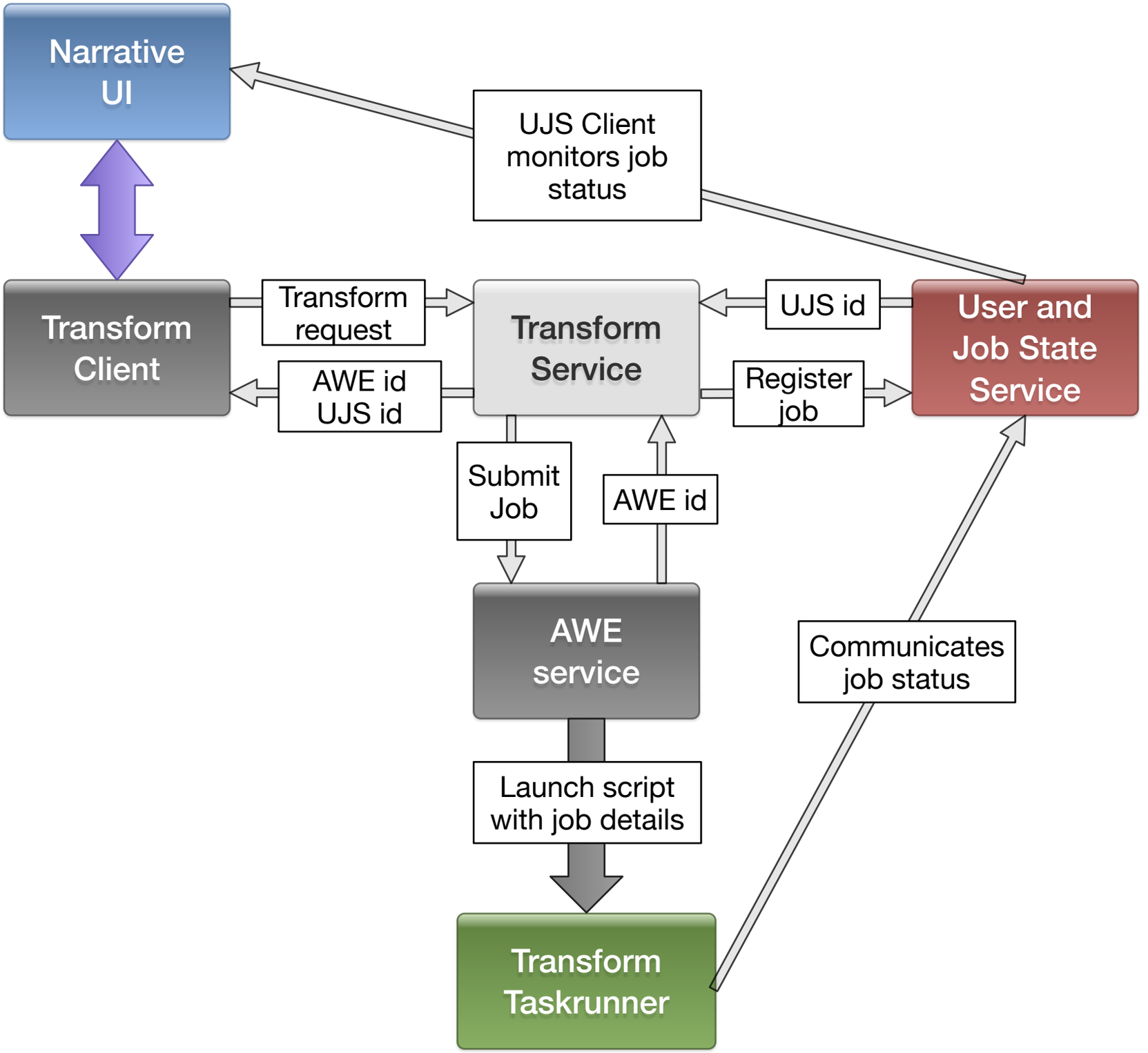
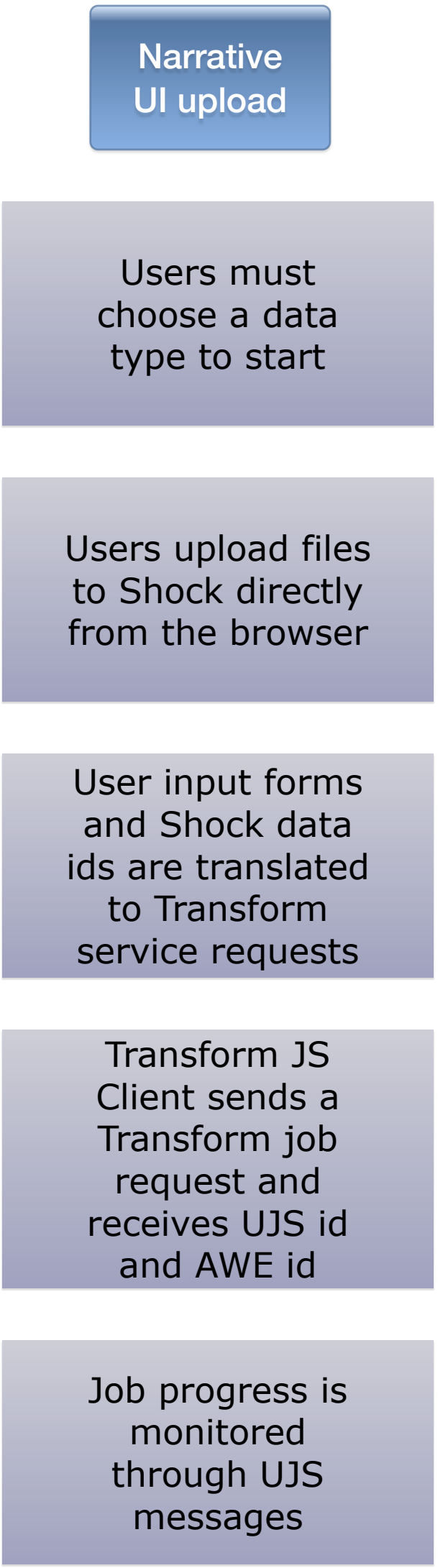

Narrative UI
UJS Client monitors job status
Transform Client
Transform request
Transform Service
UJS id
User and Job State Service
AWE id UJS id
Register job
Submit Job
AWE id
AWE service
Communicates job status
Launch script with job details
Transform Taskrunner


Narrative UI upload
Users must choose a data type to start
Users upload files to Shock directly from the browser
User input forms and Shock data ids are translated to Transform service requests
Transform JS Client sends a Transform job request and receives UJS id and AWE id
Job progress is monitored through UJS messages

Transform Taskrunner

Upload Taskrunner

Create working directory

Download input data

Decompress data

Run validator (if exists)

Run uploader

Save objects (if not done by converter)

Cleanup

Download Taskrunner

Create working directory

Run downloader

Record object provenance to file

Package files into zip/tarball

Upload zip/tarball to Shock

Send Shock URL to UJS

Cleanup

Convert Taskrunner

Create working directory

Run converter

Cleanup


Transform plugin
Executable that accepts command line options
JSON Config file that describes how to run it

Ideally should be able to develop anywhere, no need for a KBase environment.

Transform plugin development environment

Build virtualenv with minimal KBase clients using included script

Log to stdout/ stderr to have your plugin messages communicated back to the user.

Use a script tester (provided) to run the Taskrunner with your plugin executable and config. Develop your code and config in place in the repo.

Utility functions are provided with Transform for plugins to keep plugin behavior consistent and development simple.

Testing

Could wire up the upload_script_test.py to run each plugin for automated testing in Travis. This would have been the plan for a U/D sprint. Your tests are then easily constructed as configs to run the driver with.

For testing against any Transform/KBase deployment, there is an upload_client.py driver that can upload data to shock, trigger the Transform job request, monitor UJS, provide debugging output.