Figure 1: Conceptual diagram of data flow in Elixir. The arrows show the path that data about an image follows through the components of Elixir. Rounded rectangles represent programs, circles represent data products.

**RealTime Elixir Components**

Elixir has many useful diagnostic tools for evaluating images during observations, the Real Time component of Elixir. These tools in general have more aspects which are specific to the details of the CFH12K and the CFHT observing environment. Figure 1 shows schematically the different realtime components, and their relationship with other Elixir components and the telescope environment.

The connection between Elixir and the telescope is provided by a small shell script, `elixir.fork`, which is launched by the image acquisition software at the end of each exposure. This script simply calls all other needed Elixir scripts, passing them the names of the new image, along with the complete path and a flag telling the program the mode of the image (MEF/SPLIT). The `elixir.fork` script also sets up the appropriate unix PATH, umask, and other environment variables as needed.

There are two goals to wrapping the Elixir calls this script. First, by having a small script which launches other analysis scripts, we can choose to run the different processes on different machines as needed. Second, the small script can be extremely minimal and provide little impact on the observing process.

The `elixir.fork` script launches four main programs:

- **getseeing** script is launched for every science (OBJECT) image. This script measures the seeing in a subraster of a specific chip (chip 03) and writes the result, along with filter and image name, to a specific file. This measurement of the seeing can be finished within roughly 10 seconds of the image being obtained, giving the observers a quick look at basic image statistics. The resulting values are displayed the the user display tool `edisp`, and updated as new images are obtained.

- The second script launched by `elixir.fork` creates a binned jpeg image of the full mosaic. These jpeg images are only created for FOCUS and OBJECT data.

- The third `elixir.fork` script is `get.focus`, which is only applied to focus frames. This script performs a quick sextractor analysis on 4 of the 12 chips, identifies the stars in the focus sequence, finds the FWHM for stellar images from each focus step, and plots the measurements as a function of focus position. The plot also shows the median FWHM for each focus position and a set of parabolic fits for each of the four chips. We use four chips (01, 04, 07, 10) evenly spaced around the center of the mosaic to make a global measurements of the current seeing. This is an important issue since there can be significant image quality variations across the mosaic. By choosing four chips across the field-of-view, we can optimize the focus for the detector as a whole, without biases to a specific chip.

- The fourth script launched by `elixir.fork` is `elixir.grab`, which passes the image to the rest of the Elixir system. `elixir.grab` first makes a copy of the image to the Elixir raw-data disk. A copy is made for several reasons: it allows Elixir to operate on a version of the image in Waimea, it ensures that the copy Elixir knows about is available, in case the observers accidentally (or intentionally) delete it from the CFH12K account. The next thing `elixir.grab` does is to enter the image in the Image Registration database. Finally, `elixir.grab` passes a signal to the two real-time analysis elixirs, `ptolemy` and `imstats`, informing them of the image location.

Except for `elixir.grab`, these scripts run on a PIII machines at the summit, to avoid the I/O time of a copy to Waimea.