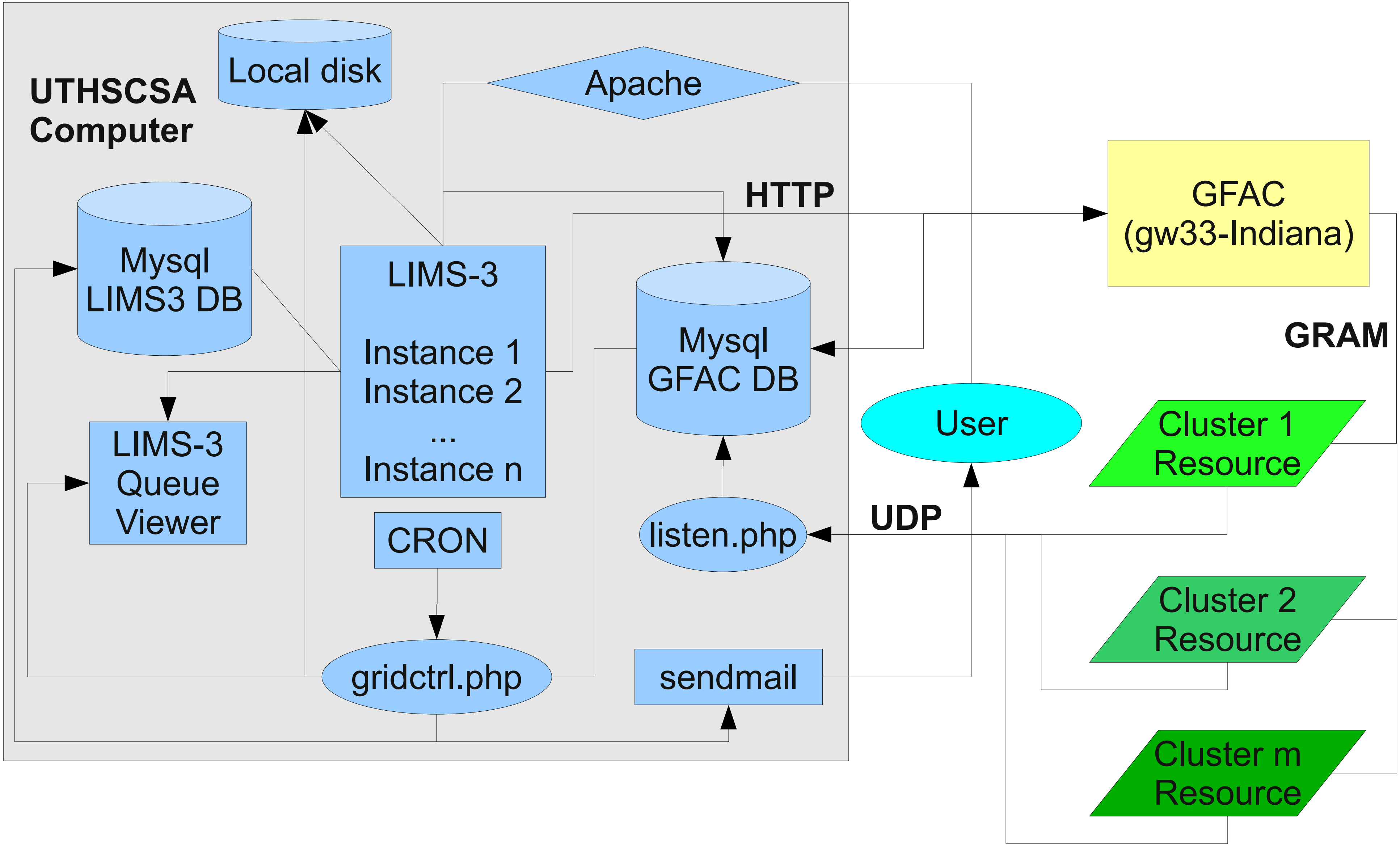
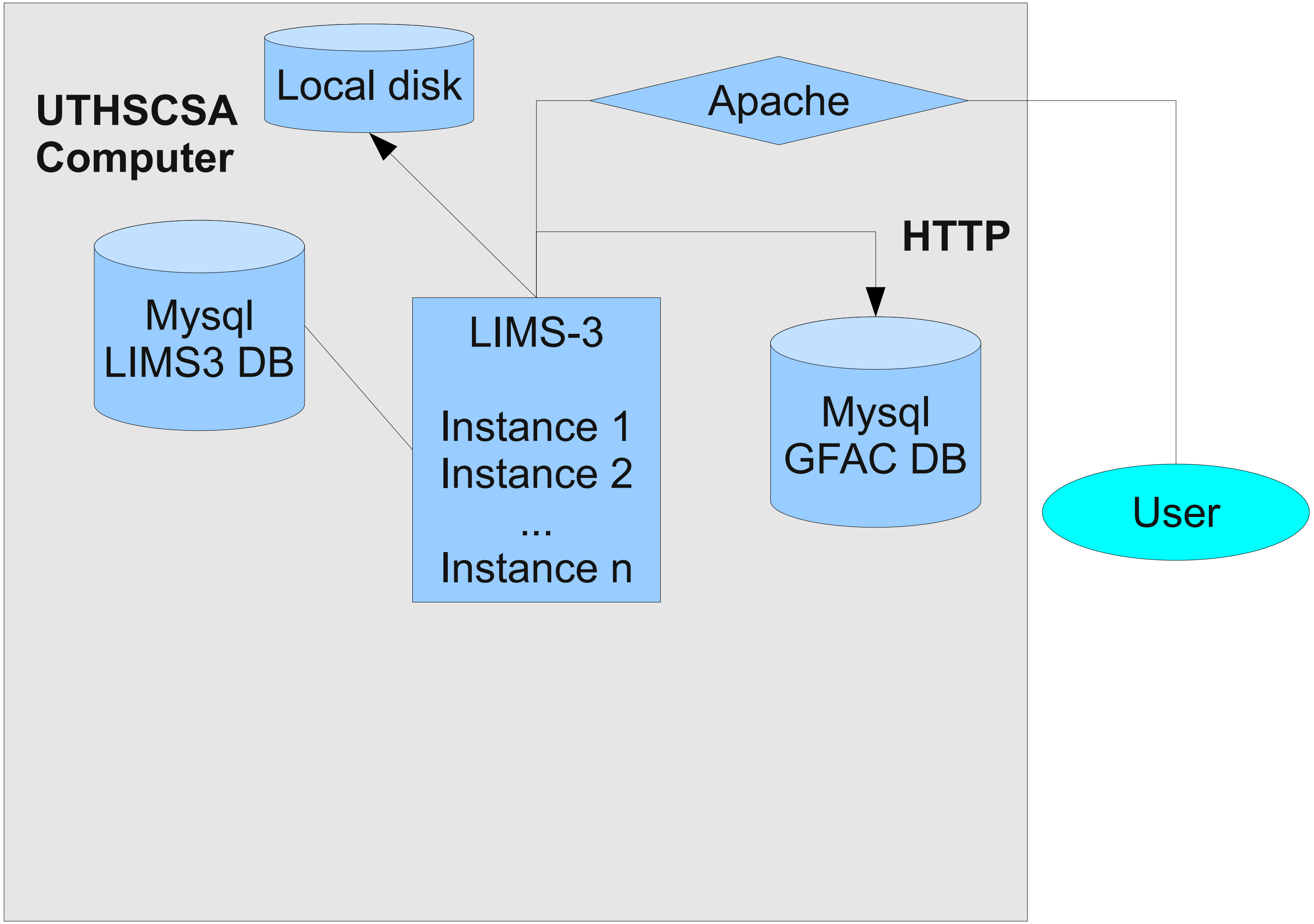
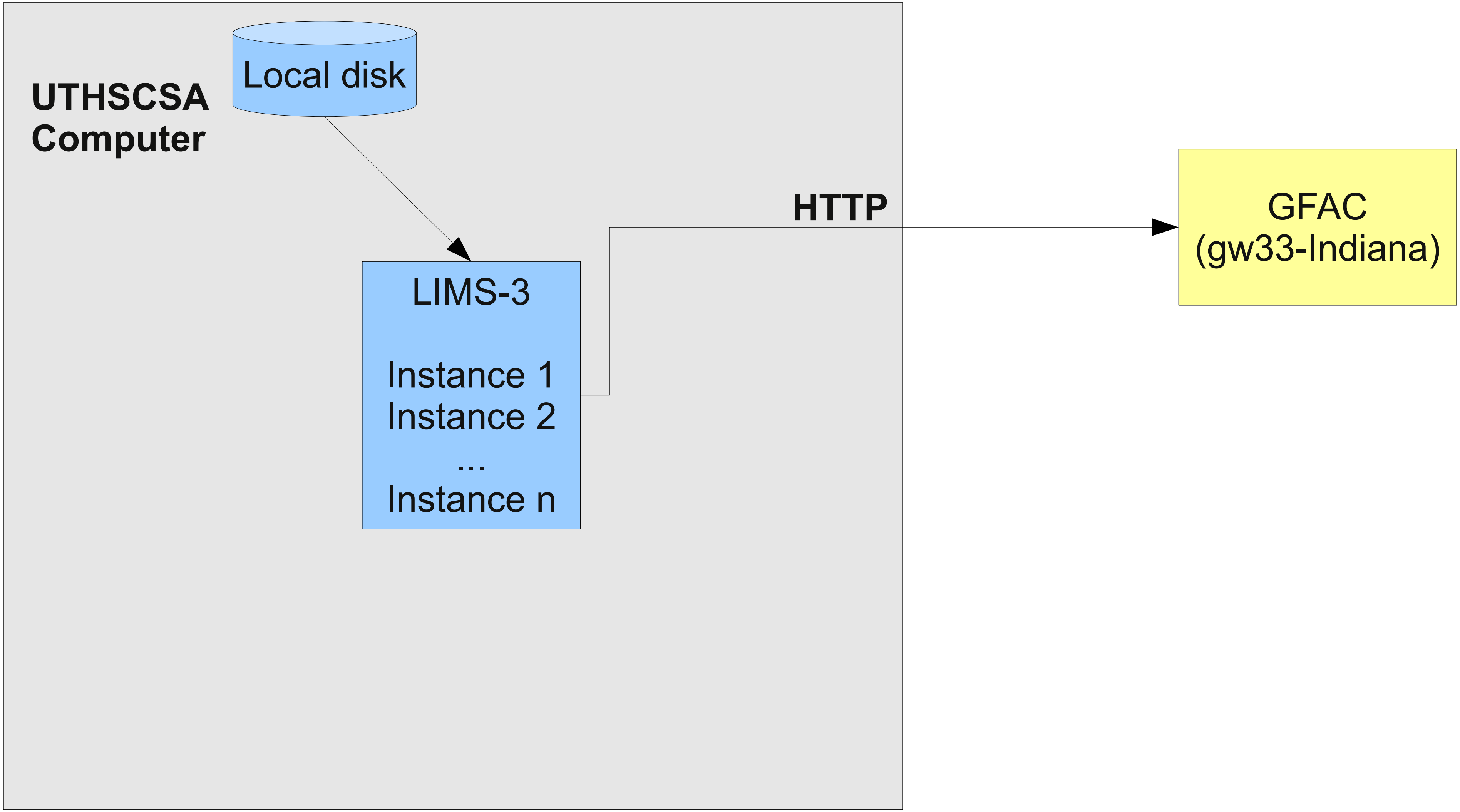
UltraScan LIMS3/GFAC Data Flow Overview



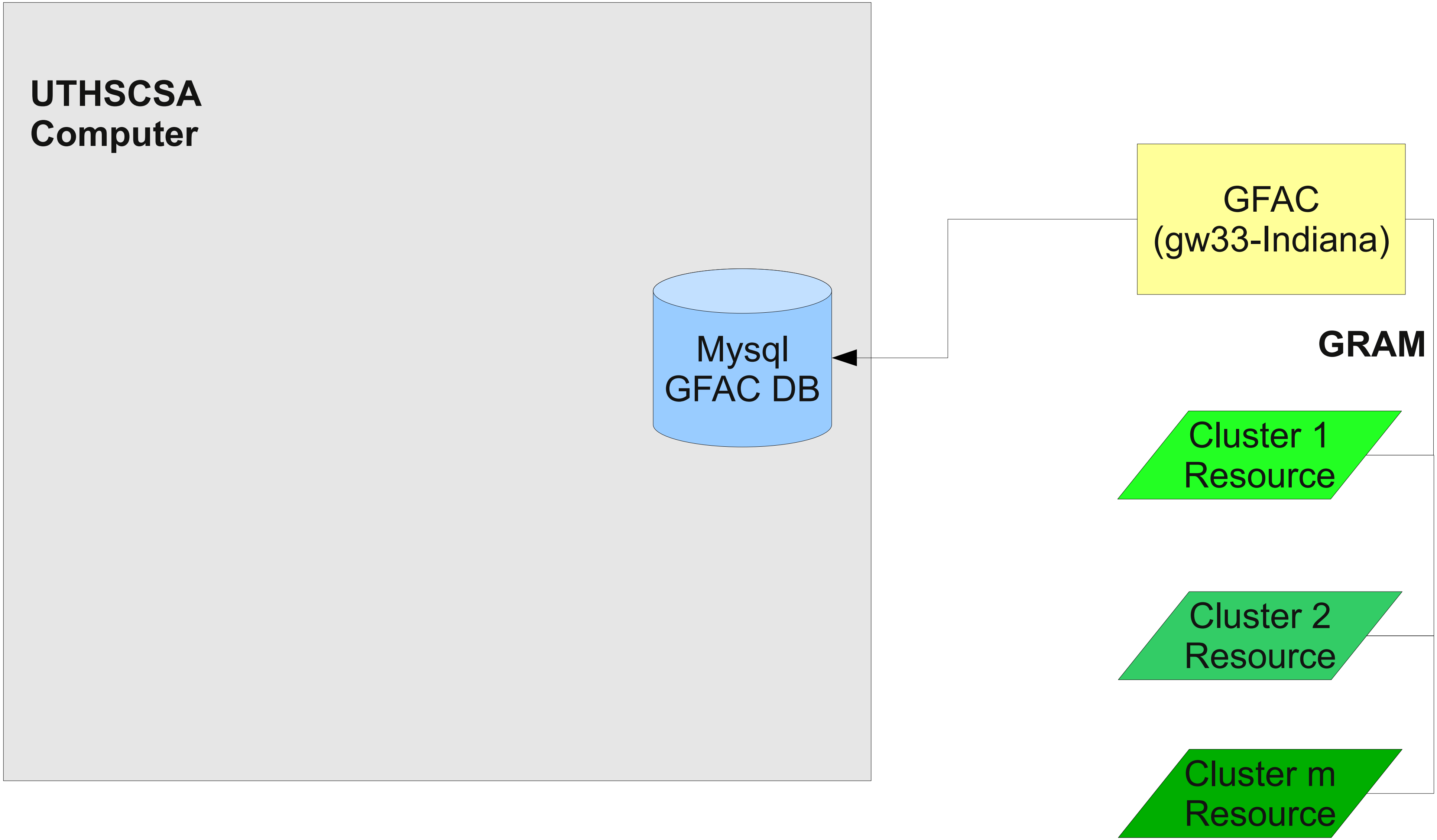
UltraScan LIMS3/GFAC Data Flow
Step 1



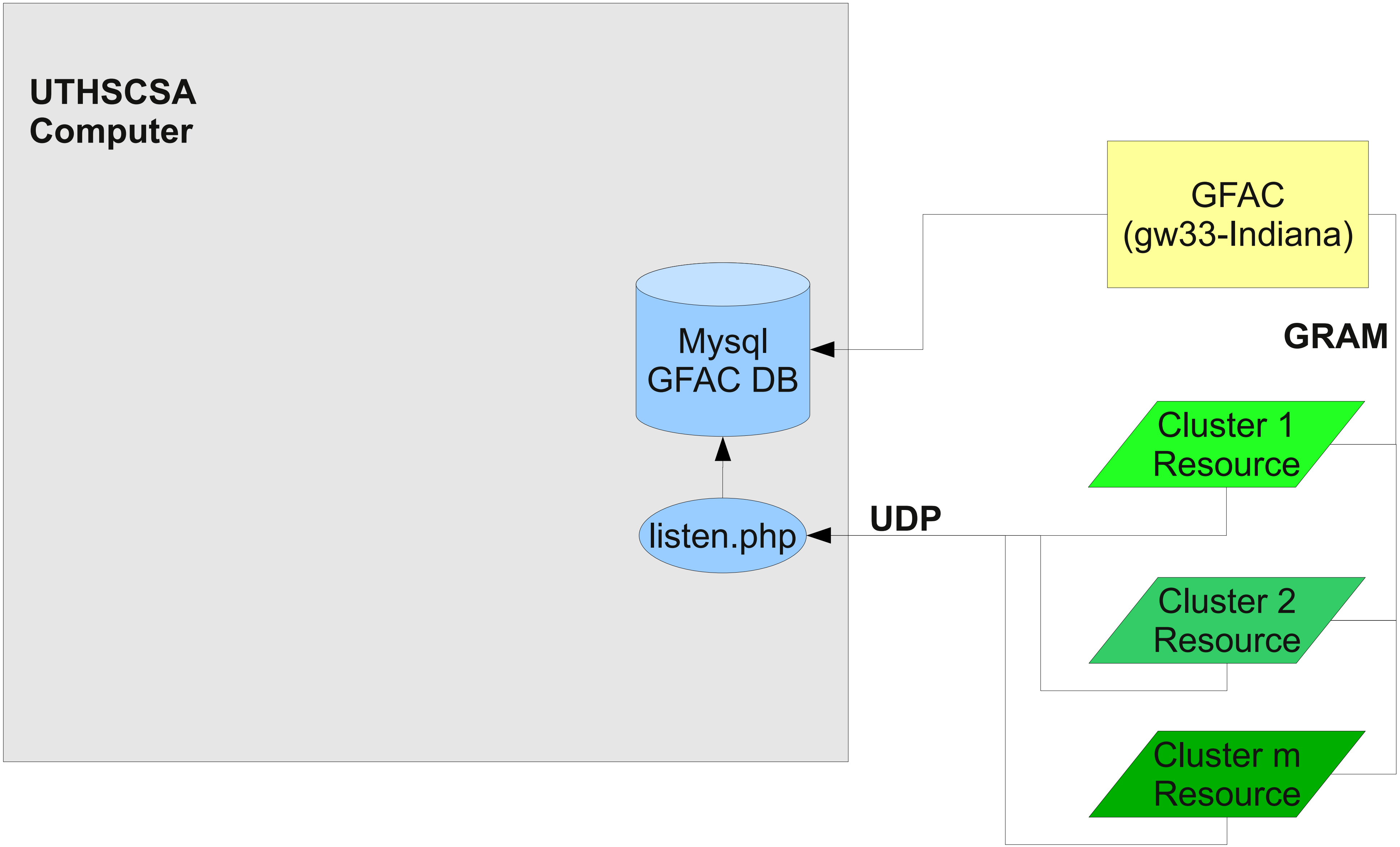
UltraScan LIMS3/GFAC Data Flow
Step 2



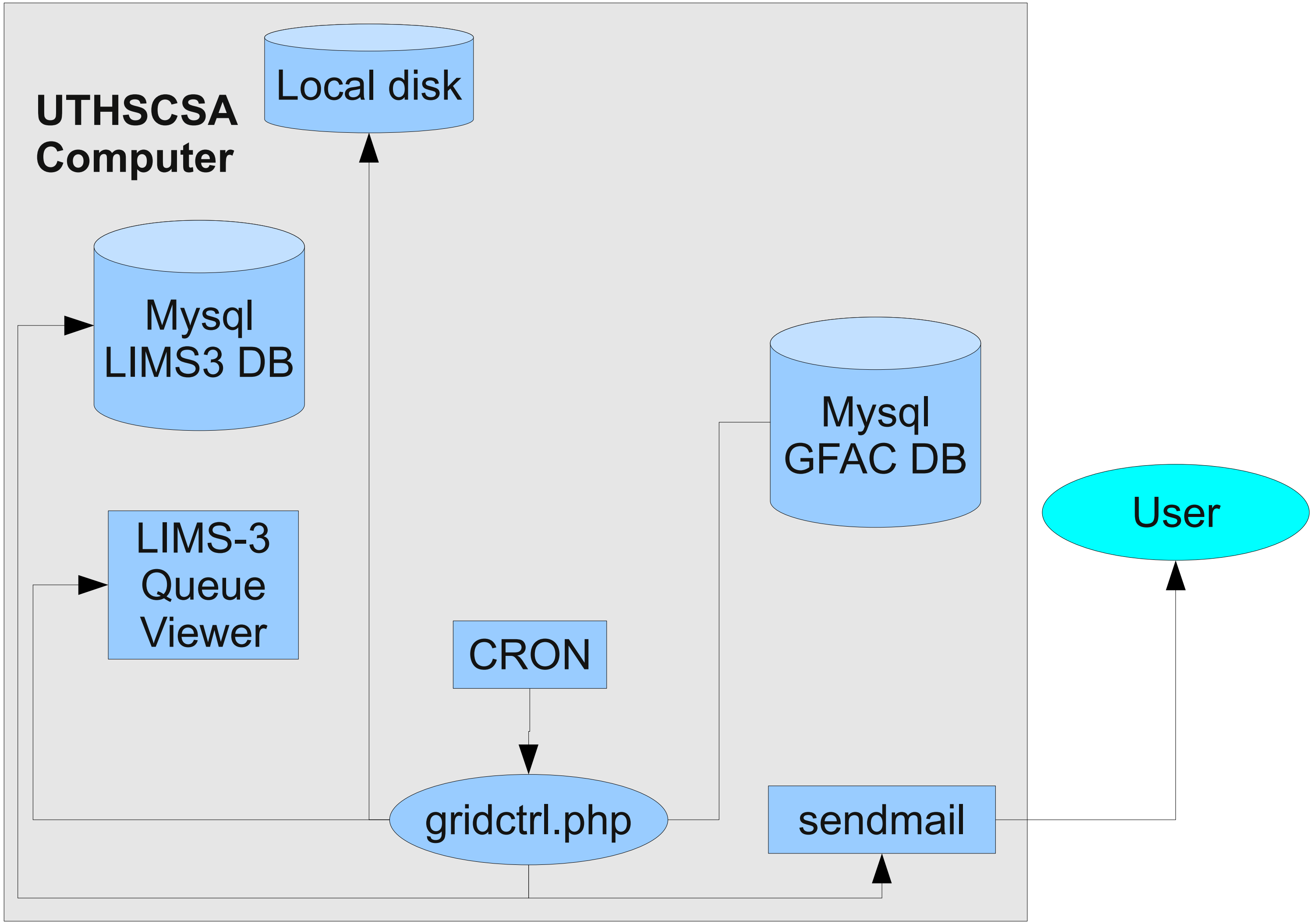
UltraScan LIMS3/GFAC Data Flow
Step 3

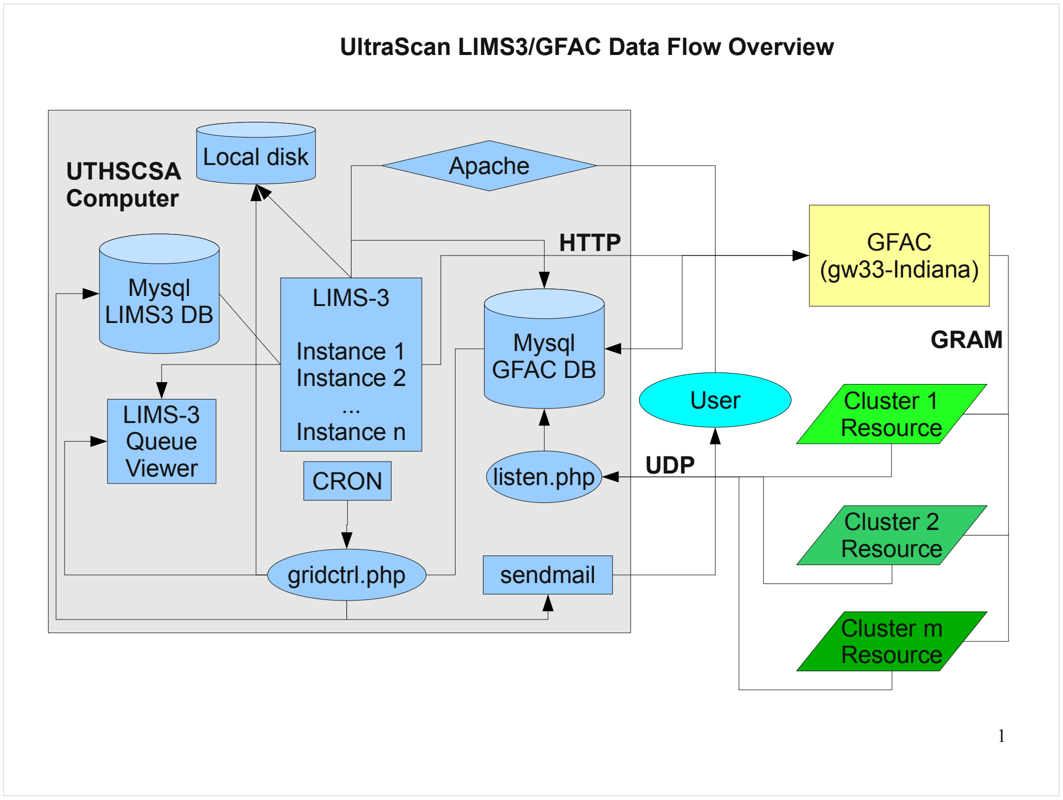


UltraScan LIMS3/GFAC Data Flow
Step 4



UltraScan LIMS3/GFAC Data Flow
Asynchronous Step





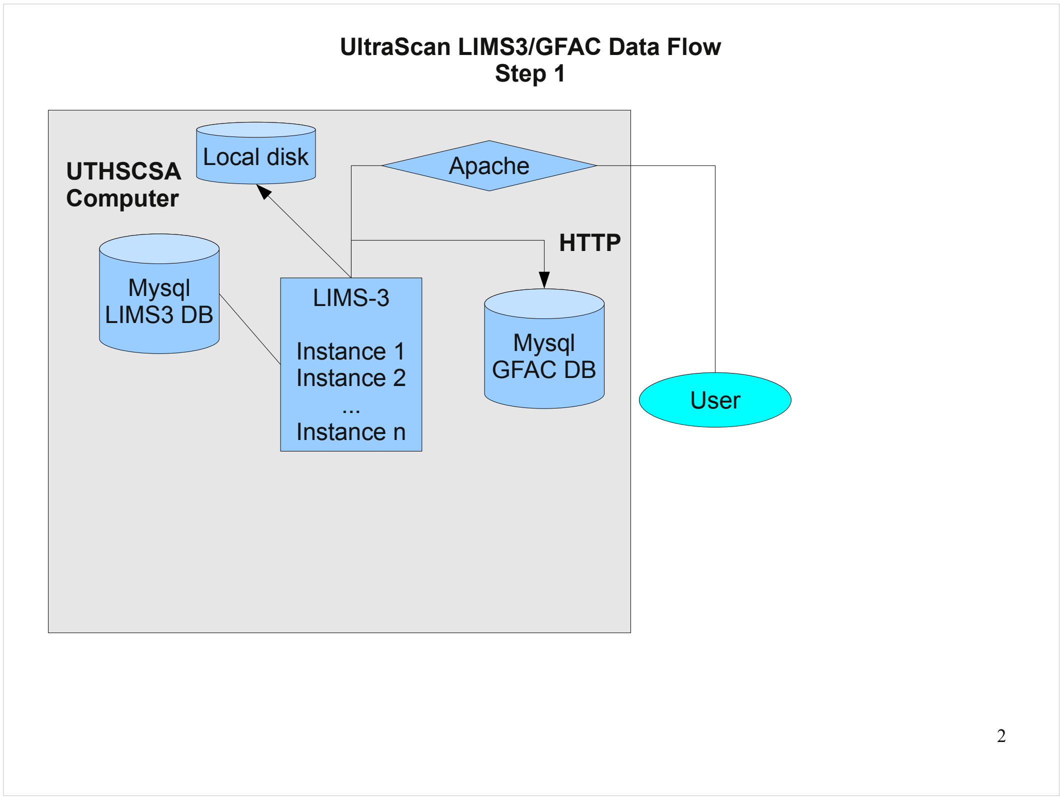
UltraScan LIMS3/GFAC Data Flow Overview

UltraScan-III/LIMS-III GFAC Data Flow Schema:

Components in blue reside on UTHSCSA computer (local disk is not shown)

green indicates cluster resource,

yellow indicates GFAC resource at Indiana U.

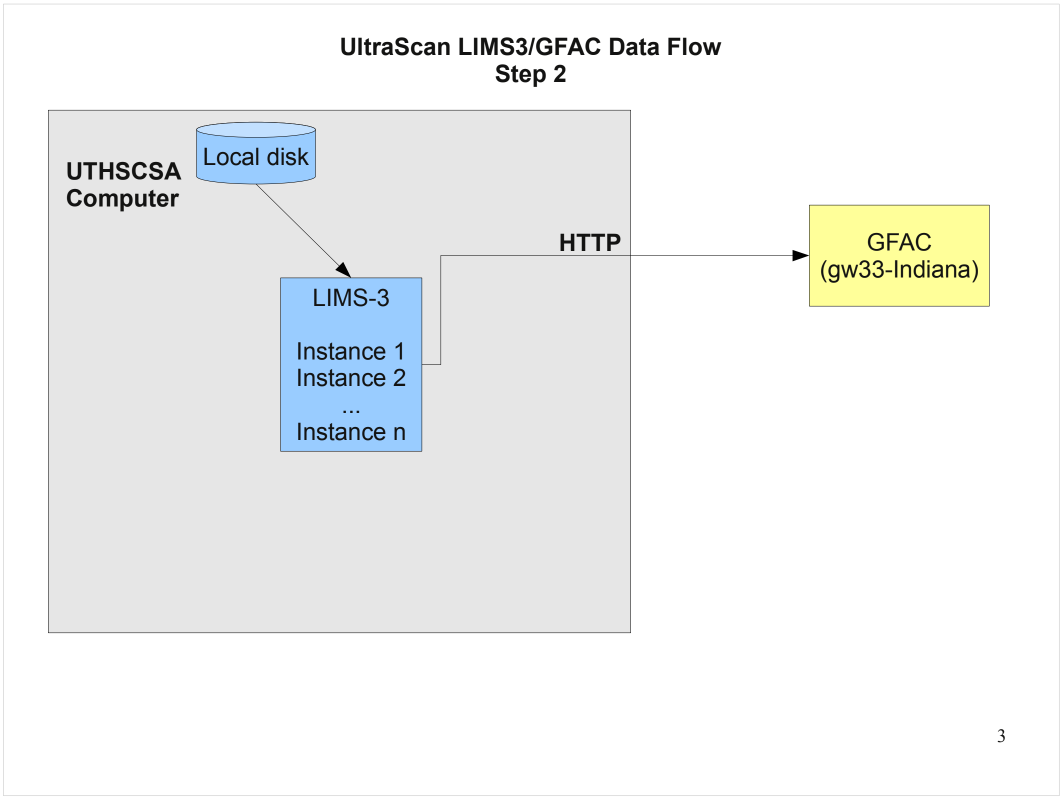


UltraScan LIMS3/GFAC Data Flow
Step 1

UltraScan-III/LIMS-III GFAC Data Flow Step 1:

User authenticates to LIMS3 instance via Apache on UTHSCSA computer, and selects their data (.auc file(s), edit profile(s), noise files) from the LIMS3 DB. User selects analysis method and analysis parameters, and cluster resource. LIMS3 issues a new job-GUID and creates a temporary local directory identified by the job-GUID name, and copies needed data into this directory and creates job tarfile.

LIMS3 then creates a new record in the GFAC DB. This record has the following fields: jobGUID, status1, status2, lims3DbName, submitterEmail, cluster, timeCreated, timeSubmitted, timeStarted, lastUpdated, analysisType, iteration, stderr, stdout, tarFile (blob), resultFile (blob). LIMS3 will update the following fields of this record: status1 field for this jobGUID is updated to “Job Created”, and lims3DbName, submitterEmail, cluster, timeCreated, analysisType are set to known values. Field “iteration” is set to zero, all other fields are left blank.



UltraScan LIMS3/GFAC Data Flow
Step 2

UTHSCSA
Computer

Local disk

LIMS-3

Instance 1
Instance 2
...
Instance n

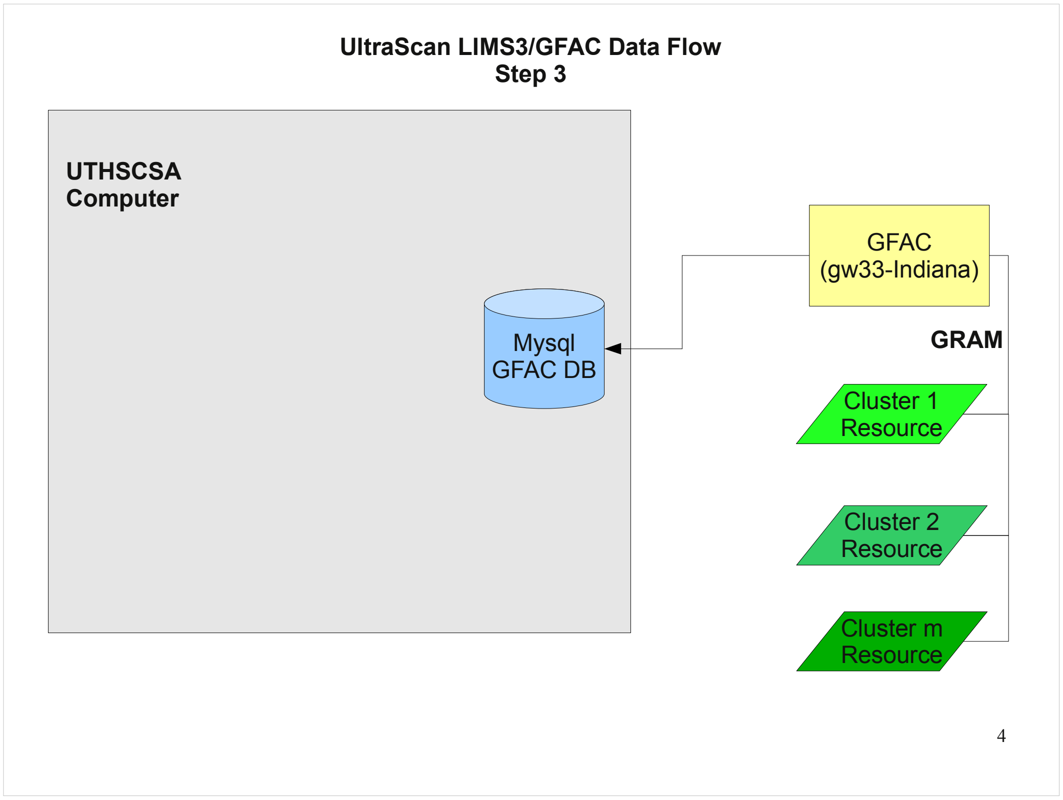
HTTP

GFAC
(gw33-Indiana)

3

UltraScan-III/LIMS-III GFAC Data Flow Step 2:

Tarfile is copied from Local disk and bin64 encoded and passed as http POST to GFAC resource along with a payload containing the following details: job GUID, desired cluster resource, predicted run time, number of processors requested, and estimated max memory. GFAC identifies this job from now on only by its jobGUID



UltraScan LIMS3/GFAC Data Flow
Step 3

UTHSCSA
Computer

Mysql
GFAC DB

GFAC
(gw33-Indiana)

GRAM

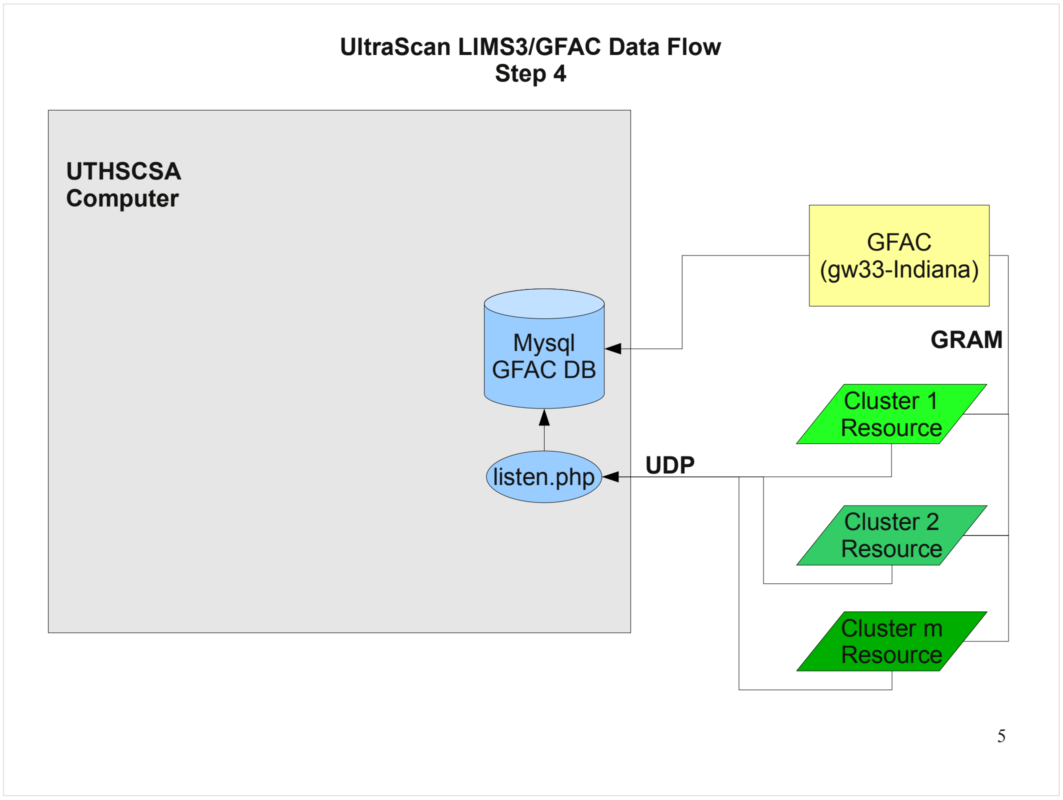
Cluster 1
Resource

Cluster 2
Resource

Cluster m
Resource

UltraScan-III/LIMS-III GFAC Data Flow Step 3:

GFAC creates a PBS (or similar) submission script for the jobGUID and sends the tarfile via GRAM to the desired cluster resource, where the tarfile is unpacked by the UltraScan backend and processed. GFAC then updates the jobGUID entry in the GFAC database and changes the field timeSubmitted to currentTime, and status1 to “Job Submitted”



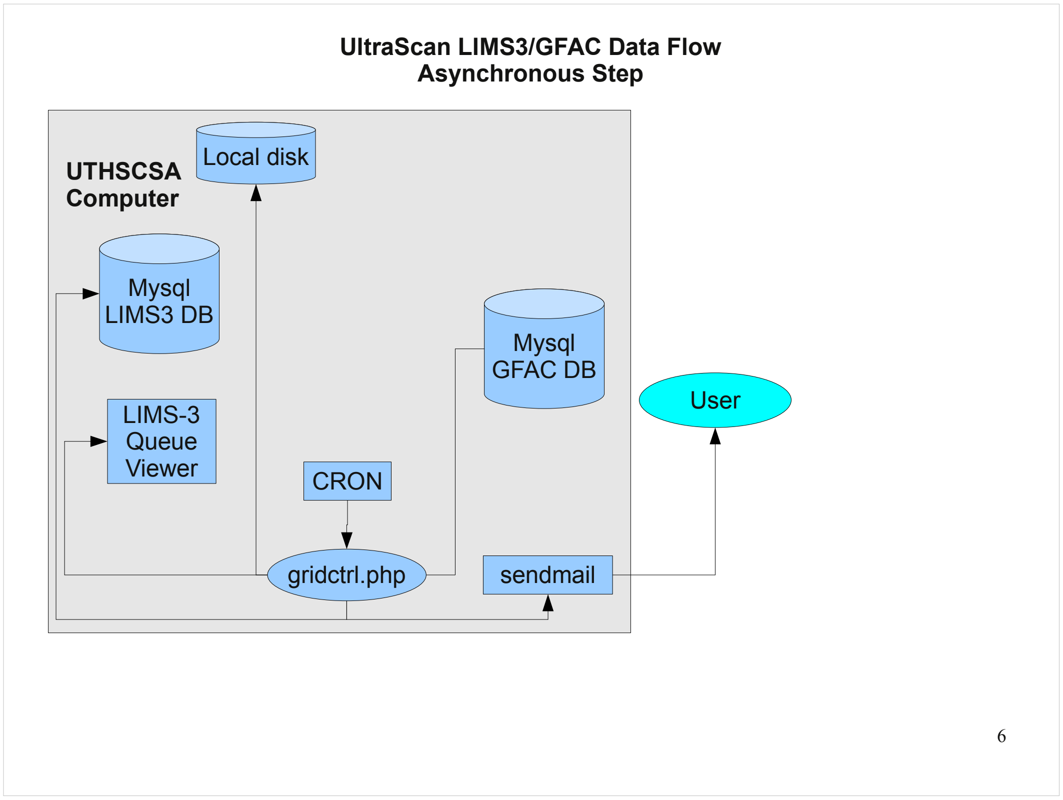
UltraScan LIMS3/GFAC Data Flow
Step 4

UltraScan-III/LIMS-III GFAC Data Flow Step 4:

While jobGUID is running, each cluster continually sends debug messages over a UDP channel to listen.php, which captures and interprets this stream. For example, it needs to capture the interation number of a Monte Carlo job, or the meniscus position and iteration currently being evaluated and update this information in the GFAC DB. Every time a UDP message is updated for a jobGUID, the lastUpdated field is also set to currentTime. If fields stdout and stderr are still blank, lastUpdated is set to currentTime by listen.php. On the UTHSCSA side, listen.php is the only device that can update entries in the GFAC DB, but it never creates or deletes records in GFAC DB (on the Indiana U side, GFAC also can update the GFAC DB).

GFAC will set the status1 field to the appropriate value based on information received via GRAM, and manage “Delete Job” requests received from LIMS3.

GFAC DB also has a queueMessage table that has the following fields: jobGUID, message, timestamp. These are updated by GFAC and hold the history of the GFAC status stamps.



UltraScan LIMS3/GFAC Data Flow
Asynchronous Step

UltraScan-III/LIMS-III GFAC Data Flow Asynchronous Step 1:

Under control of the cron daemon, gridctrl.php is executed periodically and will perform the following functions:

1. query GFAC DB for all existing jobs, and determine status and possible error condition. gridctrl.php is smart enough to interpret any possible error condition based on the fields status1, status2, timeCreated, timeSubmitted, timeStarted, stderr, stdout, and concatenates queueMessage entries into separate debug file. If fatal error is encountered, delete the job from GFAC DB, update queue viewer, and e-mail user & administrator.
2. If a job is completed, it sends e-mail notification to user only, and copies the relevant fields to LIMS3 DB and updates hpcAnalysisResults, and copies the stderr, stdout, result tarfile and emai message to the jobGUID directory, where they remain for 7 days to allow post-mortem analysis, then deletes all jobGUID entries from GFAC DB and removes job from queue viewer.
3. if job is not yet completed, it updates all changed fields in the queue viewer
4. a separate application allows administrator to see all fields in the GFAC DB for realtime troubleshooting.

gridctrl.php is the ONLY device that writes to the queue viewer, LIMS3 DB, and sendmail! On the UTHSCSA Computer, LIMS3 initially creates the GFAC DB, but after that only gridctrl.php can read from it. gridctrl.php never writes to GFAC DB, except when it deletes a record.