Kepler Science Operations Center Architecture

Christopher Middour*[a], Todd C. Klaus[a], Jon Jenkins[b], David Pletcher[c], Miles Cote[c], Hema Chandrasekaran[f], Bill Wohler[a], Forrest Girouard[a], Jay P. Gunter[a], Kamal Uddin[a], Christopher Allen[a], Jennifer Hall[a] , Khadeejah Ibrahim[a], Bruce Clarke[b], Jie Li[b] , Sean McCauliff [a], Elisa Quintana[b], Jeneen Sommers[a], Brett Stroozas[a], Peter Tenenbaum[b], Joseph Twicken[b], Hayley Wu[b], Doug Caldwell[b], Stephen Bryson[c], Paresh Bhavsar[c], Michael Wu[d], Brian Stamper[e], Terry Trombly[e], Christopher Page[e], Elaine Santiago[e]

[a] Orbital Sciences Corp., NASA Ames Research Center, M/S 244-30 Moffett Field, CA 94035 USA
[b] SETI Institute, NASA Ames Research Center, M/S 244-30 Moffett Field, CA 94035 USA
[c] NASA Ames Research Center, M/S 244-30 Moffett Field, CA 94035 USA
[d] Bastion Technologies, NASA Ames Research Center, M/S 244-30 Moffett Field, CA 94035 USA
[e] LOGYX, LLC, NASA Ames Research Center, M/S 244-30 Moffett Field, CA 94035 USA
[f] Lawrence Livermore National Laboratory, P.O. Box 808, L-478, Livermore, CA 94551 USA

ABSTRACT

We give an overview of the operational concepts and architecture of the *Kepler* Science Processing Pipeline. Designed, developed, operated, and maintained by the *Kepler* Science Operations Center (SOC) at NASA Ames Research Center, the Science Processing Pipeline is a central element of the *Kepler* Ground Data System. The SOC consists of an office at Ames Research Center, software development and operations departments, and a data center which hosts the computers required to perform data analysis. The SOC's charter is to analyze stellar photometric data from the *Keple*r spacecraft and report results to the *Kepler* Science Office for further analysis. We describe how this is accomplished via the *Kepler* Science Processing Pipeline, including the hardware infrastructure, scientific algorithms, and operational procedures. We present the high-performance, parallel computing software modules of the pipeline that perform transit photometry, pixel-level calibration, systematic error correction, attitude determination, stellar target management, and instrument characterization. We show how data processing environments are divided to support operational processing and test needs. We explain the operational timelines for data processing and the data constructs that flow into the *Kepler* Science Processing Pipeline.

Keywords: NASA, Kepler, transit photometry, architecture, Java, MATLAB, extrasolar, space telescope

1. INTRODUCTION

The *Kepler* spacecraft surveys a part of the Orion arm of our Milky Way galaxy. This neighborhood of the galaxy has enough potential targets and is far enough from the ecliptic so as not to be obscured by the Sun. Candidate objects for follow-up study are discovered by examining the amount of light emitted by each star, then looking for the periodic dimming that would be caused by a planet orbiting the star in an orientation that crosses between the star and the orbiting photometer.

The *Kepler* Science Operations Center (SOC) designed, developed, operates, and maintains the *Kepler* Science Processing Pipeline, a major analysis software component of the NASA *Kepler Mission*. This science data pipeline is used for analyzing photometric data from the *Kepler* spacecraft in support of the mission's search for Earth-like, extrasolar planets. During operations, the SOC analyzes stellar photometric data and reports the results to the *Kepler*

* chris.middour@nasa.gov

Science Office (SO). The pipeline performs five main functions: target observation planning (under direction of the SO), maintenance of target lists, generation of target and compression tables, photometer performance monitoring, and analysis of the stellar photometric data. The SOC runs the science data pipeline for transit searches, manages the database of science targets, provides target data, and is responsible for monitoring and reporting the photometer's status to the project for further evaluation. From a list of stellar targets delivered by the SO, the SOC generates target definitions, which specify to the spacecraft the photometer pixels to be used for a given quarter's observations. The pipeline software also includes science analysis tools used by project scientists to retrieve information supporting analysis.

The SOC consists of an office at Ames Research Center, a software development organization, an operations department, and a data center which hosts the computers required to do the data analysis[1]. The *Kepler Mission* has invested more than one hundred person-years in building a custom-designed transit photometry pipeline. The software has the capability to process 160,000+ stellar targets and provide instrument pixel and performance metrics. Mission requirements demand a high degree of software parallelism, scalability, and data storage throughput. The unique *Kepler* instrument's signatures require software calibration and systematic corrections which are not part of standard photometry packages. Existing commercial and open source photometry packages did not scale to 5.5 million pixels sampled every half hour. Responding to the unique needs of the mission, SOC staff designed a platform of widely adopted hardware and software upon which to implement the custom algorithms. The programming languages used are MATLAB, Java, and C++. Science algorithms are implemented in MATLAB, with data management functionality provided by Java code. The Java code retrieves inputs for a given unit of work from the database servers, passes these inputs to the pipeline algorithms, and stores the resulting outputs in the datastore once the algorithms complete. A small amount of C++ code is used to optimize performance in critical sections. Standard Intel-architecture servers running Linux are used to execute the software.

The *Kepler* Science Processing Pipeline receives raw instrument data from other mission ground segment elements. The main source of data is the Data Management Center (DMC) located at the Space Telescope Science Institute (STScI) at The Johns Hopkins University in Baltimore, Maryland. *Kepler* data consist primarily of 29.4-minute *long cadence* samples for 160,000+ stellar targets collected from 84 charge-coupled device (CCD) output amplifiers (referred to as CCD channels or module/outputs), resulting in nearly 20 GB of raw, long cadence pixel data received by the SOC per month. In addition, *Kepler* collects 58.9-second *short cadence* samples for up to 512 targets, resulting in an additional 4 GB of raw pixel data per month.

2. SOFTWARE ARCHITECTURE

The *Kepler* Science Processing Pipeline software consists of approximately one million lines of code, approximately half in Java and half in MATLAB. Functionality is spread across 25 software components, which cooperate via a flexible pipeline framework. High-level functions of the pipeline include:

1) Transform pixel data from the *Kepler* spacecraft into stellar light curves (flux time series).
2) Search each flux time series for signatures of transiting planets.
3) Fit physical parameters of planetary candidates, and calculate error estimates.
4) Perform statistical tests to reject false positives and establish accurate statistical confidence in each detection.
5) Manage target aperture and definition tables specifying which pixels in the spacecraft's CCD array are to be downlinked.
6) Manage the science data compression tables and parameters.
7) Report on the *Kepler* photometer's health and status semi-weekly after each low-bandwidth contact and monthly after each high-bandwidth science data downlink.
8) Monitor the pointing error and compute pointing tweaks when necessary to adjust the spacecraft pointing to ensure the validity of the uplinked science target tables.
9) Calibrate pixel data to remove instrument systematics.
10) Archive calibrated pixels, raw and corrected flux time series, and centroid location time series.

The *Kepler* Science Processing Pipeline software is configured and executed by the SOC Operations department[2]. Operations staff maintain software models containing thousands of parameters, update focal plane characterization models, perform data acceptance with the *Kepler* Science Office, perform data accounting, configure and execute processing pipelines, and generate archive products using *Kepler* Science Data Pipeline software.

3. HARDWARE ARCHITECTURE

The *Kepler* Science Processing Pipeline hardware architecture is built from commodity server and high-end storage components. The Fedora 11 or Red Hat Enterprise GNU/Linux operating system is used on all servers. A modular architecture allows the SOC to integrate additional computing power into the pipeline as needed, with a minimum of effort. The hardware is partitioned into separate clusters, which are configuration-managed and used for different purposes, such as monthly processing or quarterly transit searches. Firewalls isolate each cluster, and the clusters use separate user access lists based on the identified minimum set of users required.

A cluster nominally consists of a relational database server; a time series database server; and fourteen compute servers, which execute the SOC's science data processing algorithms and pipeline software. The servers are dual quad-core Xeon X54702 series processors (eight cores per server), containing 32 gigabytes of RAM and running at speeds of 2.6 to 3.0 GHz. Figure 1 shows the as-built system located within the SOC at NASA Ames Research Center.

Figure 1. *Kepler* Science Processing Pipeline hardware at NASA Ames Research Center.

4. SOFTWARE COMPONENTS

The *Kepler* Science Processing Pipeline is configured as multiple pipeline segments. Each pipeline segment is based on the particular type of dataset it processes and how frequently it runs, and each consists of a sequence of modules. In this context, a pipeline module is a coarse-grained task that implements a set of algorithms. The set of pipeline segments, the pipeline modules, and the module library are completely configurable. This architecture enables modules to be easily updated, added, or removed and minimizes code changes. Figure 2 graphically represents the Science Processing Pipeline software components and illustrates the data flow between them.

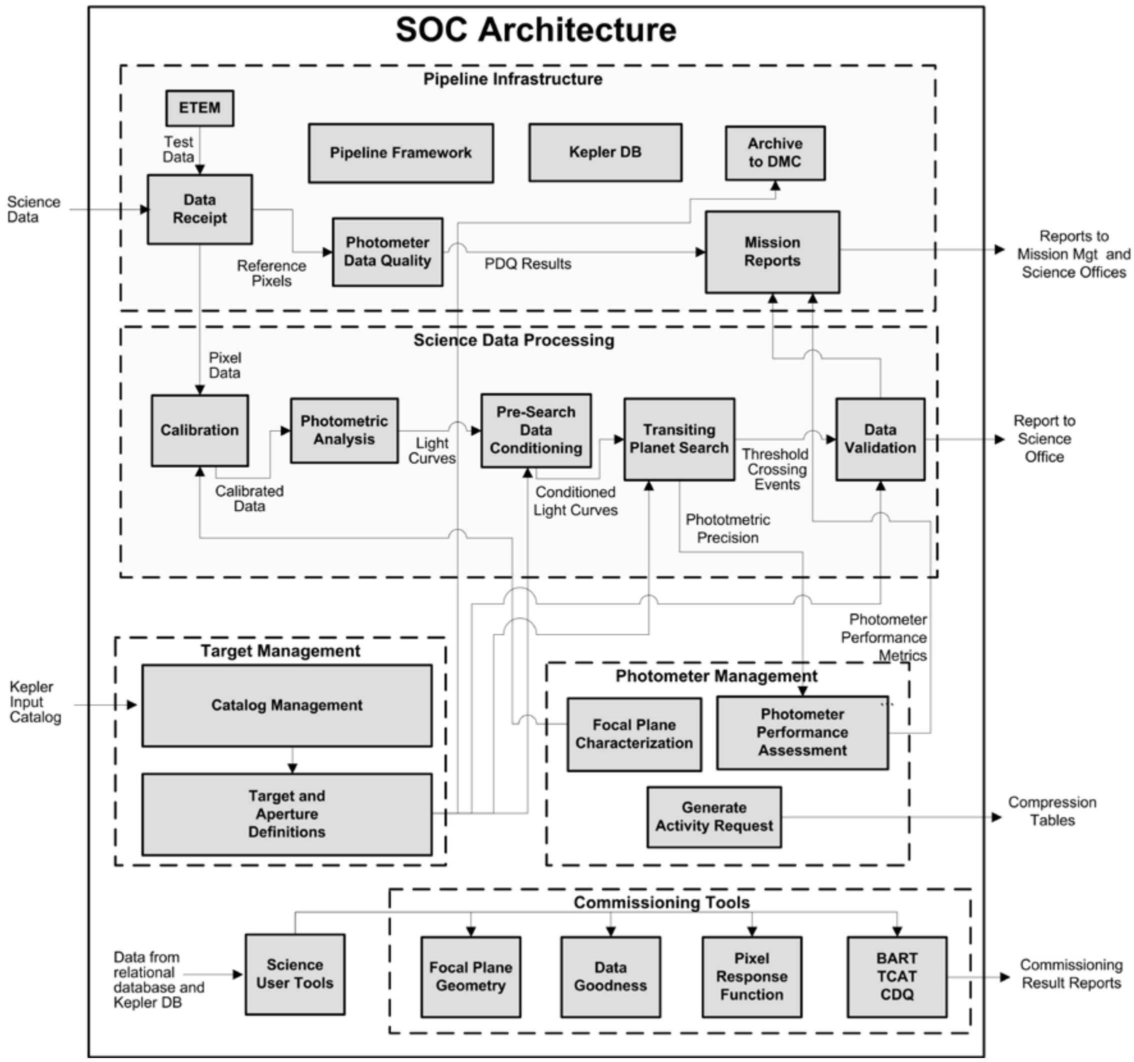

Figure 2. *Kepler* Science processing Pipeline software architecture.

4.1 Software infrastructure functionality

The infrastructure of the *Kepler* Science Processing Pipeline software provides basic functionality for data receipt, storage, and processing[3,4]. While developed for the *Kepler Mission*, none of the infrastructure software is *Kepler*-specific, and it could be used for other applications where these services are needed.

The pipeline infrastructure consists of the following modules:

- Data Receipt
- Pipeline Framework
- Kepler DB
- Archive to DMC
- Mission Reports
- End-to-End Module

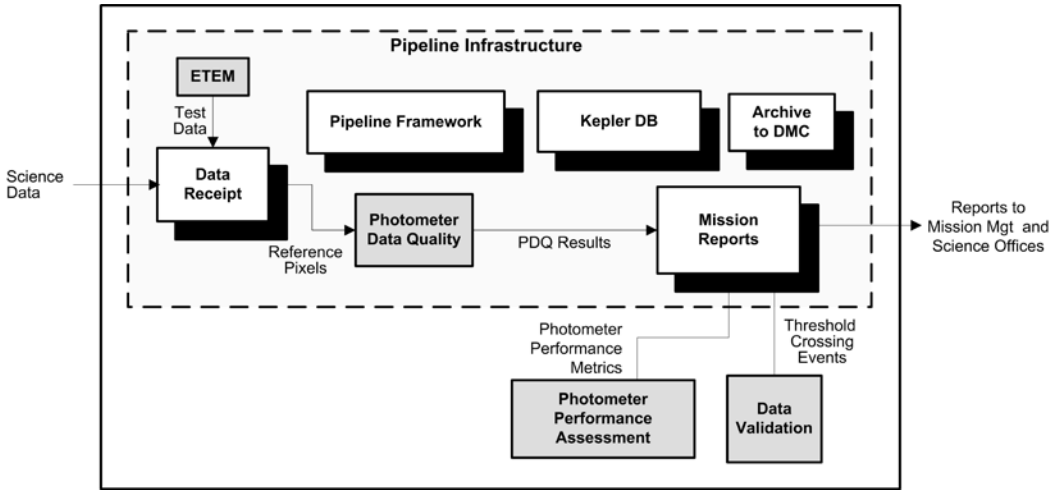

Figure 3. Pipeline infrastructure components.

The *Pipeline Framework* component[3] provides basic functionality for data communications, storage, retrieval, and manipulation. To support parallel processing requirements, the Pipeline Framework software partitions tasks into small pieces that can execute on a single CPU core. The pipeline transition logic is performed by the workers in a distributed fashion. Jobs are placed on a Java Message Service queue where worker threads can claim them when they are ready to accept a new job. As each job completes on a particular worker machine, the worker is responsible for executing the transition logic that generates the jobs for the next module. Since jobs are not pre-assigned to specific workers, the work is dynamically load-balanced across the cluster. The ways the data can be broken up depend on the nature of the pipeline module algorithms. For example, the Pre-search Data Conditioning (PDC) module needs data from all stars for a given CCD channel, but each month of data can be processed independently. In contrast, the Transiting Planet Search (TPS) module can operate on a single target at a time, but needs as many time samples as possible. The Pipeline Framework manages the distribution and synchronization of these tasks across more than 100 computer cores and provides a framework for each worker machine to call other software modules; it also provides common services like user authentication, alerts, and error handling.

The Pipeline Framework's graphical console is a Java/Swing application that displays the status of pipelines that are currently running, including how much data they have processed, what data are currently being processed, and the average throughput. The interface provides control over starting, stopping, and configuring pipelines[4].

To support development and normal operation, the Pipeline Framework software can scale to run on hardware configurations ranging from a single PC up to a full cluster of servers. A software developer can run the pipeline on a laptop during development, and then the same software can be deployed to the operational system without modification. The Pipeline Framework manages data integrity by coordinating storage transactions between the relational database and the Kepler DB software. To support management of the system, alerts to operators are displayed, and a performance metrics subsystem provides reports. The metrics subsystem also provides a simple API that can be used to instrument strategic locations in framework or application code for the purpose of collecting statistics on pipeline performance.

Data Receipt (DR) handles the receipt of science data that is sent from the DMC to the SOC. Data are parsed using XML, FITS, and binary parsers, depending on the type of data received. Depending on the type of data received, the parsed data are persisted into a popular relational database or to Kepler DB. DR also provides APIs for other pipeline components to access the persisted data.

The *Kepler DB*[5] is a custom-designed high-capacity, high-performance transactional database management system. It is used to store time series data and generic byte arrays. Kepler DB stores the vast majority of the data collected from the spacecraft, as well as final and intermediate data produced by the various pipeline modules.

Mission Reports (MR) is a web-based report viewing system. Data from other software modules display their reports via MR, which generates photometer performance assessment reports, science pipeline performance reports, and mission status reports for use by the SOC, Science Office, and Mission Management Office. MR generates some reports based on data persisted by other components and simply presents some reports (generic mission reports) generated by other components.

Archive to DMC (AR) compiles mission data (usually in the FITS format) for export to the permanent archive at the STScI.

The *End-to-End Model* (ETEM)[6] is a Monte Carlo approach to produce flight-like data for the *Kepler* photometer so the impacts of noise sources and systematic effects that are not amenable to direct analysis can be studied. The software produces simulated pixel data in exactly the same formats as exist onboard the spacecraft. After launch, ETEM has been used as a data source when ground truth is required for testing and debugging algorithms.

4.2 Target management functionality

The target management portion of the *Kepler* Science Processing Pipeline provides basic functionality for managing target observations. Due to storage and bandwidth constraints on the *Kepler* spacecraft, we can only downlink 5.44e6 of the ~95 megapixels collected on board. For this reason, the SOC must specify which pixels to downlink via target tables. Target management is one of the core SOC activities, consisting of target selection, generation of target and aperture definitions, uplink of target/aperture definition tables to the spacecraft, and performance verification for newly uplinked tables.

Once each quarter (every three months), the spacecraft is rolled 90 degrees to re-orient its solar panels toward the Sun. This also relocates the apparent position of stars on the photometer, so the SOC collaborates with the Science Office to produce new target tables that are adjusted to match the new stellar positions[7]. The target selection process provides lists of prioritized targets of various categories that are balanced to fill the final tables. Although the planetary target list accounts for the vast majority of the science targets, a number of lists are included to provide photometer performance information (e.g., stellar and image artifact targets), while others perform different scientific duties (e.g., comparison, Guest Observer, and eclipsing binary targets). The Science Office has developed a suite of MATLAB software tools to perform target selection, and to ensure that estimated target/pixel allocations remain within the operations constraints of the spacecraft and bandwidth available.

The target management portion of the pipeline consists of the following modules:

- Catalog Management
- Target and Aperture Definitions

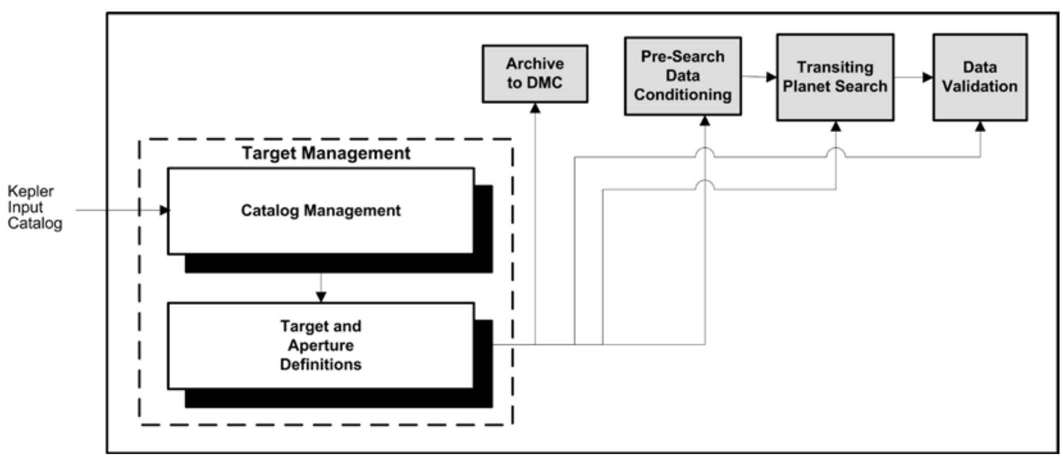

Figure 4. Target management components.

Catalog Management (CM) is a user interface for selecting or importing target lists and providing them to target management software. CM provides a graphical user interface to perform three distinct functions:

1. validate the *Kepler Input Catalog* (KIC) for correct format and missing data, load it into the SOC database, and generate KIC statistics and metrics;
2. assemble target lists specified by the Science Office into the *Kepler Target Catalog* and add observation start and stop time, target source, ID, and cadence; and
3. display or export data from the characteristics table or any of the *Kepler* catalogs based on science criteria used to select entries from that catalog.

Target and Aperture Definitions (TAD)[8] creates target definition tables for long and short cadence targets. A target definition is the selected aperture and its position on the focal plane. TAD first selects pixels that maximize the signal-to-noise ratio for each target based on a stellar catalog and measured optical and electronic properties of the *Kepler* photometer. These pixels are then assigned to one of 772 on-board pixel apertures. TAD also creates a target definition table for selected background and reference pixels.

4.3 Photometer management functionality

The photometer management portion of the *Kepler* Science Processing Pipeline provides the basic functionality for assessing photometer performance in order to maximize the scientific quality of the resulting data.

It includes the following components:

- Photometer Data Quality
- Focal Plane Characterization
- Photometer Performance Assessment
- Generate Activity Request

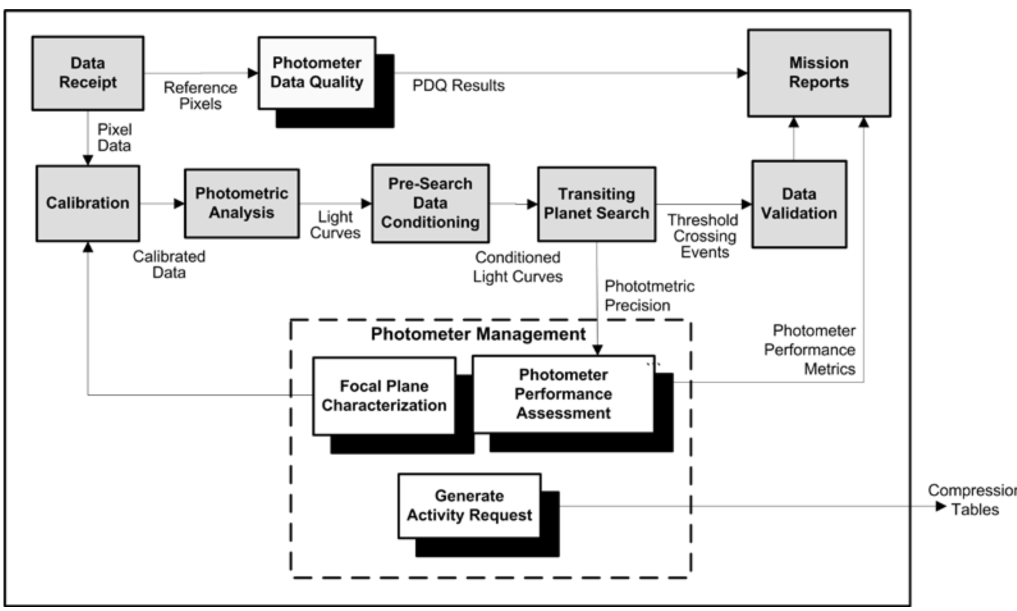

Figure 5. Photometer management components.

Photometer Data Quality (PDQ)[9] operates on a small pixel set from the spacecraft to produce metrics relating to spacecraft attitude, photometer brightness, and focus. Twice per week, the spacecraft sends a small number of pixels (called Reference Pixels) to Earth. PDQ is executed, and the resulting report is provided to the project for analysis.

Photometer Performance Assessment (PPA)[10] reports metrics on photometer performance, including spacecraft attitude, brightness, and focus. The PPA software is designed to have access to all downlinked pixels. This results in higher accuracy than the PDQ report, which is constrained to function with a smaller subset of pixels, but at a higher frequency than PPA.

Focal Plane Characterization (FC)[11] calculates physical attributes of the focal plane as measured during commissioning and computes point-spread functions based on pixel data taken between spacecraft slew maneuvers during commissioning. These are used to support target selection and focus analysis. Focal plane characteristics include: CCD geometry and pixel-response function, optical vignetting, and characterization of the noise in the electronics. Models are imported into FC when required. A history of the models imported is maintained, supporting data accountability needs and to support reprocessing older data. FC is capable of interpolating between points. For example, the 2D black model is occasionally updated and FC can interpolate the time interval between updates. FC models are retrieved directly by the pipeline at runtime for processing, but additional query tools, known as Science User Tools, are used by mission scientists for offline analysis and reporting.

Generate Activity Request (GAR) generates compression tables for upload to the spacecraft via the *Kepler* Mission Operations Center at the Laboratory for Atmospheric and Space Physics at the University of Colorado and the DMC.

4.4 Science Data Processing Functionality

The data analysis portion of the *Kepler* Science Processing Pipeline includes pixel-level calibration, light curve generation, systematic error correction, outlier identification, gap filling, identification of threshold crossing events, and validation[12]. This functionality is provided by the following components:

- Calibration
- Photometric Analysis
- Pre-search Data Conditioning
- Transiting Planet Search
- Data Validation

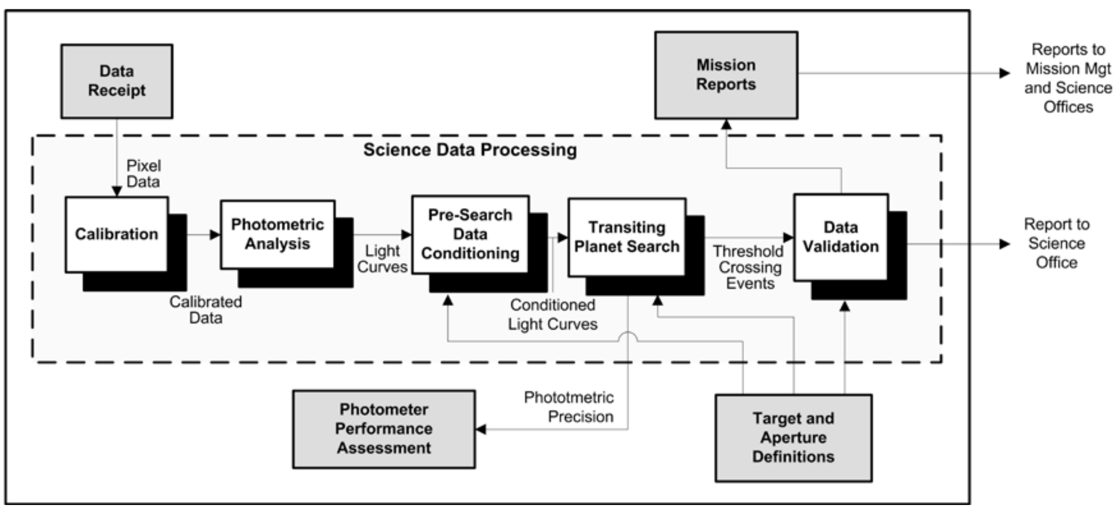

Figure 6. Science Data Processing Components.

Calibration (CAL)[13] is the first processing step applied to pixel data in the pipeline. CAL ingests uncalibrated data from DMC and calibrates it for a fixed-offset value, a mean black value (per CCD pair), a bias value, nonlinearity, gain, undershoot (an artifact introduced by a clamp circuit in the photometer electronics), smear (due to the lack of a shutter in the telescope), dark current (a thermally induced signal), and a standard flat field correction. The pipeline's CAL software removes cosmic rays from collateral data (the data from each CCD channel necessary to calibrate and interpret the respective pixel data and reach the required precision). Detecting cosmic rays in these types of pixels is done via a simple threshold algorithm. This is possible with these pixel types because they do not receive light input, and therefore cosmic rays stand out clearly over the otherwise low signal levels. To allow parallel processing and memory management, CAL processing is divided into units of work based on the source module/output, and time ranges. Collateral data are processed in the first invocation, so black and smear values will be available at the start of photometric pixel processing. Processing uncertainties or "errors" are calculated by CAL's Propagation of Uncertainties (POU) software [14]. Given the large data volumes pertaining to measuring errors in data processing, a novel approach was taken to provide uncertainty information within the computer resources available. Ideally, a software system would calculate errors in lock-step with data processing, but given the *Kepler* data volume, a solution like this would be impossible due to computer memory constraints. The POU solution includes storing the relatively small set of primitive data and metadata needed to reproduce pixel covariance information rather than storing the full covariance matrix. A singular value decomposition is performed, keeping only the highest-power components to further reduce the size of the stored information, with the added benefit of acting as a low-pass filter. Elements of the pixel covariance matrix are reconstructed to within 1 part in 1000 of the original value, which is within requirements for the propagated errors. This method significantly reduces the memory needed for execution.

Photometric Analysis (PA)[15] converts pixel values to target light curves. PA uses simple aperture photometry along with background-corrected, calibrated target pixel values to generate a flux time series for each target. Cosmic rays are identified and removed in both the background and target pixels. Detecting cosmic rays in photometric data (pixels which receive stellar flux) requires information over time, so cosmic ray removal from photometric pixels is performed in PA, which is capable of processing multiple time samples. Centroids are calculated for each target star on each frame. A number of metrics are calculated and persisted.

Pre-Search Data Conditioning (PDC)[16] is used to remove systematic errors prior to performing transiting planet searches. Error-corrected light curves are also exported to the Multimission Archive at STScI. PDC operates on raw flux light curves from PA and performs systematic error correction, outlier identification, and data gap filling. Artifacts in light curves caused by thermal changes are removed by cotrending with engineering data from the spacecraft, such as readings from temperature sensors. Spacecraft motion effects on light curves are addressed with the use of motion models produced by PA. Random discontinuities due, for example, to cosmic ray induced sensitivity changes are identified and corrected. Excess flux due to crowding in stellar apertures is removed. When systematic errors are corrected, an attempt is made to ensure that large transits (and other astrophysical events such as binary eclipses and flares) are left intact. Gaps in data caused by scheduled downlinks or safe modes onboard the spacecraft, are filled.

Transiting Planet Search (TPS)[17] applies a wavelet-based, adaptive, matched filter to identify transit-like features with durations of 1 to 16 hours. TPS makes use of the transit photometry method, examining the amount of light emitted by each star, then looking for periodic dimming caused by a planet orbiting the star in an orientation that crosses between the Earth and the star. At the moment the planet is directly between the telescope and the star, it dims the light the telescope sees by a small fraction. TPS measures the amount of dimming and the length of time over which it occurred. Light curves with transit-like features and a combined transit detection statistic exceeding 7.1σ for some trial period and epoch are designated as "threshold crossing events" (TCEs), which are then subject to analysis by the Data Validation module of the pipeline. This threshold ensures that no more than one false positive deriving from random fluctuations will occur over the life of the mission (assuming non-white, non-stationary Gaussian observation noise).

Data Validation (DV)[18] performs a suite of statistical tests to evaluate the confidence of a TCE, identify and reject false positives caused by background eclipsing binaries, and extract the physical parameters of each system for each candidate planet[19] (together with uncertainties and covariance matrices). Because other phenomena besides actual transits can cause a light curve signature to falsely appear to be a planet, the SOC compares the transit signatures with spacecraft engineering data (such as onboard temperatures and reaction wheel speeds) to eliminate false positives. The results are also compared to Kepler's laws of planetary motion, which is a further way to eradicate false positives. The results of this search are provided to the Follow-Up Observing Program, which schedules time on Earth-based telescopes to confirm the planetary candidates. After DV fits the planetary signature, DV removes it from the light curve, subjects the residual to a search for additional TCEs, and repeats the process until it identifies all TCEs. The SOC also provides the processed data to the Threshold Crossing Event Review Team, who evaluate and prioritize the TCEs for ground-based follow-up observations.

4.5 Spacecraft Commissioning Software

In Spring 2009, the mission conducted a thorough checkout of the spacecraft, ground station, ground software, and instruments. The SOC participated in this exercise using custom-built analysis tools. Once the spacecraft was in orbit, the commissioning phase began[20]. The photometer was turned on, and on-orbit science measurements compared to expected values based on ground-based tests. Based on the reports from the SOC, the spacecraft then ejected the dust cover. During commissioning, many full-frame images (FFIs) of all photometer pixels were sent to the SOC for analysis by the commissioning tool suite. Commissioning tools ran on both these FFIs and the dithered long cadence data used by FPG and PRF.

The tools in this suite, which were specifically designed for *Kepler* by the SOC, include:

- Focal Plane Geometry
- Data Goodness
- Pixel Response Function
- Black and Artifact Removal Tool
- Temperature Coefficient Analysis Tool
- Check Data Quality
- Science User Tools

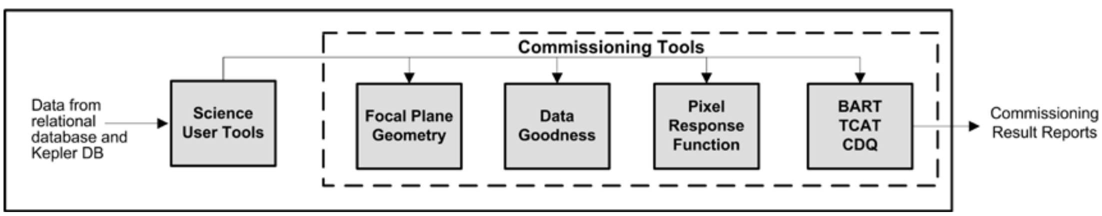

Figure 7. Commissioning tools.

Focal Plane Geometry (FPG)[21] fits the actual position of each CCD on the focal plane using star data. FPG measures the as-built positions and orientations of the CCDs on the focal plane of the *Kepler Mission*'s flight segment, using the night sky as a high-precision metrology tool. It is necessary to maximize the number of stellar targets which can be monitored for signs of planetary transits. Due to bandwidth and data processing limitations, this translates to a requirement to minimize the number of pixels devoted to each stellar target. To do this, the sky-to-pixel geometry must be as accurate as possible.

Data Goodness (DG) checks for data completeness or data corruption, possibly indicating a transmission or data handling problem. DG checks data content to see whether the right data were taken, and then checks data values to see if they indicate a potential problem with the hardware. A report is generated with all plots, histograms, and statistics of each individual detector array. This software tool continues its usefulness in normal operations, and is routinely executed on FFIs from the spacecraft.

Pixel Response Function (PRF)[22] fits the shape of the pixel response function for each module/output using star data. A PRF is defined as the optical point-spread function convolved with pixel structure and motion. The PRF tool produces a continuous model of each photometer detector array's responsivity to light. It also provides quality metrics for each PRF produced. This tool aids the SOC and Science Office in choosing pixels for optimal photometry, determining which pixels are downlinked, and determining the focus center (centroid) of each star. PRF works with the FPG tool in an iterative fashion, eventually coming up with an accurate geometry model. The default PRF pipeline is configured to loop back to FPG and then again to PRF over a configurable number of iterations. The iteration stops when the centroid change is below a configurable threshold. The computed PRFs are stored each iteration so there is an opportunity for examination by Science Office personnel. If the PRFs are approved, they are delivered to the SOC. If the PRFs are unacceptable, corrective action is determined and the process flow is restarted from an appropriate stage.

The *2D Black and Artifact Removal Tool* (BART) detects and models temperature-dependent image artifacts in pixel data[23]. The main purpose of BART is to support the decision to eject the spacecraft dust cover, during the commissioning phase of the mission. BART provides insight into whether the photometer data are consistent with pre-launch expectations regarding temperature variations. The spacecraft's data-gathering process is subject to a number of instrument artifacts, which can damage the quality of the mission's data if not properly managed. Of particular interest is the crosstalk from the spacecraft's fine guidance sensor (FGS) readout into the science CCD readout, which has a temperature-dependent intensity. In order to properly correct for this effect, it is necessary to determine the temperature dependence of the crosstalk intensity and to produce a 2D black model that incorporates the correct crosstalk intensity (i.e., to estimate the crosstalk intensity at the spacecraft's nominal operating temperature). The temperature dependence of the 2D black is studied via a series of FFIs acquired at different temperatures. Once the BART processing is complete, the CDQ and TCAT tools report further statistical analyses on BART's resulting data files.

The *Temperature Coefficient Analysis Tool* (TCAT) is a tool to study the thermal variations of pixels which are affected by crosstalk caused by the FGS. FGS Crosstalk is a major source of *Kepler* instrument noise. In order to calibrate pixels to remove FGS crosstalk, it is necessary to understand its thermal variation.

Check Data Quality (CDQ) checks and analyzes the RMS of data fitting residuals and thermal coefficients produced by BART for the pixels in the collateral regions of the photometer. The collateral pixel regions are areas that do not receive light, but are used for photometer diagnostics. CDQ reports the statistics on the RMS of data fitting residuals and thermal coefficients, and provides different types of plots.

The *Science User Tools* provide access to target tables, light curves, and other data products for use by mission scientists. They support read-only querying capability for use in data analysis.

5. SUMMARY

We have highlighted the software and hardware architecture of the SOC, discussed high-level functionality implemented, and described features of each software component. Interested readers will find a wealth of information about the SOC and the *Kepler* mission in the referenced papers, which are authored by the software developers and scientists who designed and implemented the software discussed here.

ACKNOWLEDGMENTS

Funding for the *Kepler Mission* is provided by NASA's Science Mission Directorate.

REFERENCES

[1] Jenkins, J., *et al.*, "Initial characteristics of *Kepler* long cadence data for detecting transiting planets," *AsJL* **713(2)**, L120–L125 (2010).

[2] Hall, J., *et al.*, "Kepler science operations processes, procedures, and tools," *Proc. SPIE* **7737**, in press (2010).

[3] Klaus, T., *et al.*, "The Kepler Science Operations Center pipeline framework," *Proc. SPIE* **7740**, in press (2010).

[4] Klaus, T., *et al.*, "The Kepler Science Operations Center pipeline framework extensions," *Proc. SPIE* **7740**, in press (2010).

[5] McCauliff, S., *et al.*, "The Kepler DB, a database management system for arrays, sparse arrays, and binary data," *Proc. SPIE* **7740**, in press (2010).

[6] Bryson, S. T., *et al.*, "The Kepler end-to-end model: creating high-fidelity simulations to test Kepler ground processing," *Proc. SPIE* **7738**, in press (2010).

[7] Batalha, N. M., *et al.*, "Selection, prioritization, and characteristics of Kepler target stars," *ApJL* **713(2)**, L109–L114 (2010).

[8] Bryson, S. T., *et al.*, "Selecting pixels for Kepler downlink," *Proc. SPIE* **7740**, in press (2010).

[9] Chandrasekaran, H., *et al.*, "Semi-weekly monitoring of the performance and attitude of Kepler using a sparse set of targets," *Proc. SPIE* **7740**, in press (2010).

[10] Li, J., et al., "Photometer performance assessment in Kepler science data processing," *Proc. SPIE* **7740**, in press (2010).

[11] Allen, C., Klaus, T., and Jenkins, J., "Kepler Mission's focal plane characterization models Implementation," *Proc. SPIE* **7740**, in press (2010).

[12] Jenkins, J. M., *et al.*, "Overview of the Kepler science processing pipeline," *ApJL* **713(2)**, L87–L91 (2010).

[13] Quintana, E. V., *et al.*, "Pixel-level calibration in the Kepler Science Operations Center pipeline," *Proc. SPIE* **7740**, in press (2010).

[14] Clarke, B. C., *et al.*, "A framework for propagation of uncertainties in the Kepler data analysis pipeline," *Proc. SPIE* **7740**, in press (2010).

[15] Twicken, J. D., *et al.*, "Photometric analysis in the Kepler Science Operations Center pipeline," *Proc. SPIE* **7740**, in press (2010).

[16] Twicken, J. D., *et al.*, "Presearch data conditioning in the Kepler Science Operations Center pipeline," *Proc. SPIE* **7740**, in press (2010).

[17] Jenkins, J., *et al.*, "Transiting planet search in the Kepler pipeline," *Proc. SPIE* **7740**, in press (2010).

[18] Wu, H., *et al.*, "Data validation in the Kepler Science Operations Center pipeline," *Proc. SPIE* **7740**, in press (2010).

[19] Tenenbaum, P., *et al.*, "An algorithm for the fitting of planet models to Kepler light curves," *Proc. SPIE* **7740**, in press (2010).

[20] Haas, M.R., *et al.*, "Kepler science operations," *ApJL* **713(2**), L115–L119 (2010).

[21] Tenenbaum, P., and Jenkins, J., "Focal plane geometry characterization of the Kepler mission," *Proc. SPIE* **7740**, in press (2010).

[22] Bryson, S. T., *et al.*, "The Kepler pixel response function," *ApJL* **713(2)**, L97–L102 (2010).

[23] Kolodziejczak, J., *et al.*, "Flagging and Correction of Pattern Noise in the Kepler Focal Plane Array." *Proc. SPIE* **7742**, in press (2010).