UNIVERSITY OF HAWAII AT MĀNOA
Institute for Astrononmy

Pan-STARRS Project Management System

Pan-STARRS PS-1 Image Processing Pipeline
System Concept Design

Grant Award No. : **F29601-02-1-0268**
Prepared For : **Pan-STARRS PMO**
Prepared By : **Eugene A. Magnier, Paul A. Price, Josh Hoblitt**

Document No. : **PSDC-430-011**
Document Date : **January 22, 2006**
Revision : **01**

DISTRIBUTION STATEMENT
Approved for Public Release – Distribution is Unlimited

2680 Woodlawn Drive, Honolulu, Hawaii 96822
An Equal Opportunity/Affirmative Action Institution

Submitted By:

[Insert Signature Block of Authorized Developer Representative] Date

Approved By:

[Insert Signature Block of Customer Developer Representative] Date

Revision History

Revision Number	**Release Date**	**Description**
DR.01	2004.01.01	First draft
DR.02	2004.03.05	Second draft
DR.03	2004.03.25	Section reorganization
DR.04	2004.04.13	Most sections fleshed out
DR.05	2004.04.29	Reorganization for consistency
DR.06	2004.10.21	Major revision in prep of PDR
00	2004.11.04	Incorporation of PDR Comments
01	2006.01.22	Name change (SSDD to SCD) before CDR

TBD Listing

Section No.	**Page No.**	**TBD No.**	**Description**
7	34	1	needs to be fleshed out substantially
7	34	2	needs to be fleshed out substantially

Contents

List of Figures

1 Scope

1.1 Identification

This document establishes Software Design Requirements for the Panoramic Survey Telescope and Rapid Response System (Pan-STARRS) Image Processing Pipeline (IPP) for the prototype telescope PS-1, and is a System-level controlled specification/design description document in the official Pan-STARRS engineering specification tree.

1.2 System Overview

The Institute for Astronomy at the University of Hawaii is developing a large optical synoptic survey telescope system, the Panoramic Survey Telescope and Rapid Response System (Pan-STARRS). The science goals, priorities, top-level concept of operations with associated operational requirements, and system performance drivers with associated system performance requirements are described in the Pan-STARRS Science Goals Statement (SGS). As described in this document, The system conceptual design for Pan-STARRS utilizes an array of four 1.8m telescopes each with a 7 degree^2 field of view, giving the system an étendue larger than all existing survey instruments combined (defined as the product of the collecting area A multiplied by the field-of-view solid angle Ω). Each telescope will be equipped with a 1 billion pixel CCD camera with low noise and rapid read-out, and the data will be reduced in near real time to produce both cumulative static sky and difference images from which transient, moving, and variable objects can be detected. Pan-STARRS will be able to survey up to $\approx 6,000$ degree^2 per night to a detection limit of approximately 24th magnitude. This unique combination of sensitivity and sky coverage will open up many new possibilities in time domain astronomy including a major goal of surveying the Potentially Hazardous Object (PHO) population down to a diameter of ≈ 300 meters. In addition, the Pan-STARRS data will be used to investigate a broad range of astronomical problems of extreme current interest concerning the Solar System, the Galaxy, and the Cosmos at large. A prototype single telescope system, PS-1, is being developed as a preliminary step before construction of the complete four telescope system.

Project sponsor:	AFRL, United States Air Force
Acquirer:	University of Hawaii Institute for Astronomy
User:	Astronomical community
Developer:	University of Hawaii Institute for Astronomy, participating institutions, and associated subcontractors

1.3 Document Overview

The Pan-STARRS IPP System Concept Definition (SCD) contains the complete design concepts of the Pan-STARRS PS-1 IPP in order to achieve the requirements specified by the Pan-STARRS PS-1 IPP Software Requirements Specification (SRS; PSDC-430-005). The requirements flow begun in the SGS and full Pan-STARRS SCD and continued in the SRS is used to guide the design presented here.

1.4 Requirements Definitions

The Pan-STARRS document naming scheme is PSDC-NNN-MMM-VV, where the VV entry specifies the document version number. Where documents are identified without the version number, the latest official version in that series is implied.

Open issues (TBDs) in this document are marked **in bold red**.

Quantities which should be reviewed (TBRs) are marked **in bold blue**.

1.4.1 “Shall”

When used in this specification, the word “shall” refers to an explicit requirement of a system component or the complete system.

1.4.2 “Should”

When used in this specification, the word “should” refers to a desired characteristic of a system component or the complete system.

1.4.3 “Will”

When used in this specification, the word “will” provides information about a characteristic of a related system component or a complete related system.

2 Referenced Documents

2.1 Internal Documents

Reference	**Title**
PSDC-230-001	PS-1 Design Reference Mission
PSDC-230-002	PS-1 System Concept Definition
PSDC-400-006	The Pan-STARRS IPP Computational Challenge
PSDC-430-004	Pan-STARRS IPP C Code Conventions
PSDC-430-005	Pan-STARRS IPP PS-1 Software Requirements Specification
PSDC-430-006	Pan-STARRS IPP Algorithm Design Document
PSDC-430-007	Pan-STARRS IPP PSLib Supplementary Design Requirements Specification
PSDC-430-010	Pan-STARRS IPP Perl Code Conventions
PSDC-430-012	Pan-STARRS IPP Modules Supplementary Design Requirements Specification
PSDC-430-014	Pan-STARRS IPP PS-1 Cluster Support

2.2 External Documents

Reference	**Title**
Posix Standard	Open Group Based Specifications Issue 6, IEEE Std 1003.1, 2003

3 Subsystem Overview

The Pan-STARRS Image Processing Pipeline (IPP) performs the image processing and data analysis tasks needed to enable the scientific use of the images obtained by the Pan-STARRS telescopes. The primary goals of the IPP are to process the science images from the Pan-STARRS telescopes and make the results available to other systems within Pan-STARRS. It also is responsible for combining all of the science images in a given filter into a single representation of the non-variable component of the night sky defined as the “Static Sky”. To achieve these goals, the IPP also performs other analysis functions to generate the calibrations needed in the science image processing and to occasionally use the derived data to generate improved astrometric and photometric reference catalogs. It also provides the infrastructure needed to store the incoming data and the resulting data products.

The IPP inherits lessons learned, and in some cases code and prototype code, from several other astronomy image analysis systems, including Imcat (Kaiser), the Sloan Digital Sky Survey (REF), the Elixir system (Magnier & Cuillandre), and Vista (Tonry). Imcat and Vista have a large number of robust image processing functions. SDSS has demonstrated a working analysis pipeline and large-scale database system for a dedicated project. The Elixir system has demonstrated an automatic image processing system and an object database system for operational usage.

The users of the IPP output are all systems internal to the Pan-STARRS project. They consist of: 1) the Preferred Science Clients, which receive specified data products on short timescales. 2) the Moving Object Processing System (MOPS), which is one of the Preferred Science Clients, but has the distinction of being a component funded by Pan-STARRS. It will receive the detections of non-stationary transient objects. 3) the Published Science Products Subsystem (PSPS), which will receive all data products of interest to the community external to the Pan-STARRS data processing systems, and will act as the long-term archive and publishing clearinghouse.

The IPP receives data from two Pan-STARRS subsystems: the Camera, from which it receives the large volume of image data, and OTIS (Observatory, Telescope and Infrastructure Subsystem), from which it receives metadata describing the images and the environmental conditions. The primary IPP hardware system on which the software operates will probably not be located at the summit. Instead, because of thermal, power, and space constraints, the hardware will likely be located in a facility off the mountain. A subset of processing tasks may eventually be assigned to machines at the summit if justified by the savings in data transfer time and cost.

The Pan-STARRS camera produces images consisting of multiple chips (Orthogonal Transfer Arrays or OTAs), each consisting of multiple cells (continuous set of pixels). The baseline design for the Pan-STARRS camera contains 64 chips each with 64 cells.

This document defines the design requirements of the IPP for the PS-1 prototype telescope. Much of the IPP design for PS-4 will be identical to or closely based on the PS-1 implementation. The software organization and the infrastructure systems are expected to be identical, with minor improvements in details. The type of analysis steps to be performed will be nearly identical, with some additional details added for PS-4 to improve the accuracy.

Although generally very similar, in terms of the IPP PS-1 differs from the complete PS-4 system in several specific ways. First, with only one telescope and camera, the data throughput rate is substantially reduced to a maximum of 1 64-OTA image per 40 seconds rather than 4. Since PS-1 is a prototype for testing the Pan-STARRS hardware and software subsystems, the observing strategy is not a fixed quantity. The PS-1 Design Reference Mission (PSDC-230-001) provides some guidelines for the types of observing tests which will probably be performed, including possibly starting an Astrometric and Photometric Survey which will eventually cover the entire 3π steradians of the sky accessible to PS-4. As a prototype, it is expected that much of the data collected by PS-1 will be processed multiple times to test and tune the analysis steps. Compare with PS-4, this difference in approach has implications for the storage required by PS-1: rather than delete images soon after they have been used, raw images from demonstration observations must be stored for at least the first two years of PS-1 operations. The PS-1 Design Reference Mission is used as an upper limit for these storage requirements to drive the hardware design.

3.1 System Design Decisions

Since Pan-STARRS is a survey project, all data from the telescopes will be uniformly analyzed by the Pan-STARRS Image Processing Pipeline (IPP), and the appropriate resulting data products made available to internal and external science analysis systems as they become available. The processing performed by the IPP on the science images will consist of detrending and object detection for the individual images, combination of multiple overlapping images and further object detection, subtraction of a reference (static-sky) image and detection of residual objects, update of the static sky images, and detailed object analysis of the static sky images. In addition, the IPP will produce improved astrometric and photometric reference catalogs on an as-needed basis. The output data products from the IPP consist of the calibration images, reduced images from the individual telescopes, combined images, difference images, the static sky image, object photometry, and reference astrometry and photometry.

The requirements for the IPP, as identified in the PS-1 IPP SRS (PSDC-430-005) fall into several broad categories: data analysis precision, throughput, system reliability, flexibility, testability, and traceability. The details of the analysis tasks are specified in order to achieve the precision. The architectural design as discussed below is motivated by the need for reliability and flexibility. The hardware organization and the distributed/parallel processing model is motivated by the throughput requirements. The need for flexibility and testability drives the software organization. The need for simple testing procedures drives both the software organization and the separation of the system architecture into different infrastructure elements.

3.2 Analysis Tasks and Stages

Specific programs are required to perform the processing steps listed above. These can be divided into well-defined analysis stages, each of which operates on a particular unit of data, such as a single OTA image or a collection of astronomical objects. Analysis tasks representing the different analysis stages are performed on the IPP computer cluster. Note the distinction between the generic analysis *stage* and a specific analysis *task*. An analysis stage represents a type of analysis which is performed, such as the basic image calibration and object detection analysis. An analysis task is a particular realization of an analysis stage, e.g., the analysis of OTA number 61 from exposure 654321 to produce a specific set of output data products. The analysis stages are discussed in detail in Section 5.

A particular stage may process individual images, collections of images, or derived data products. Because of the nature of the image data, many of the analysis stages can be run in parallel if needed to increase the processing throughput. For example, the analysis of a chip in one image does not depend on the results from another chip.

3.3 Architectural Components

In order to achieve the required functionality, the IPP provides an infrastructure within which the Analysis Stages described above are executed. In order to facilitate the subsystem testing, the IPP software infrastructure has been divided into a number of clearly-defined architectural software units as follows:

- **Image Server:** This component is a large data store for all images used by the IPP, including the raw images from the telescope, the master calibration images, the reference static-sky images, and any temporary image data products produced by the IPP. The Image Server accepts the incoming data and stores it until it is no longer needed by other portions of the IPP. The Image Server is not restricted to imaging data: it is capable of storing any large data files

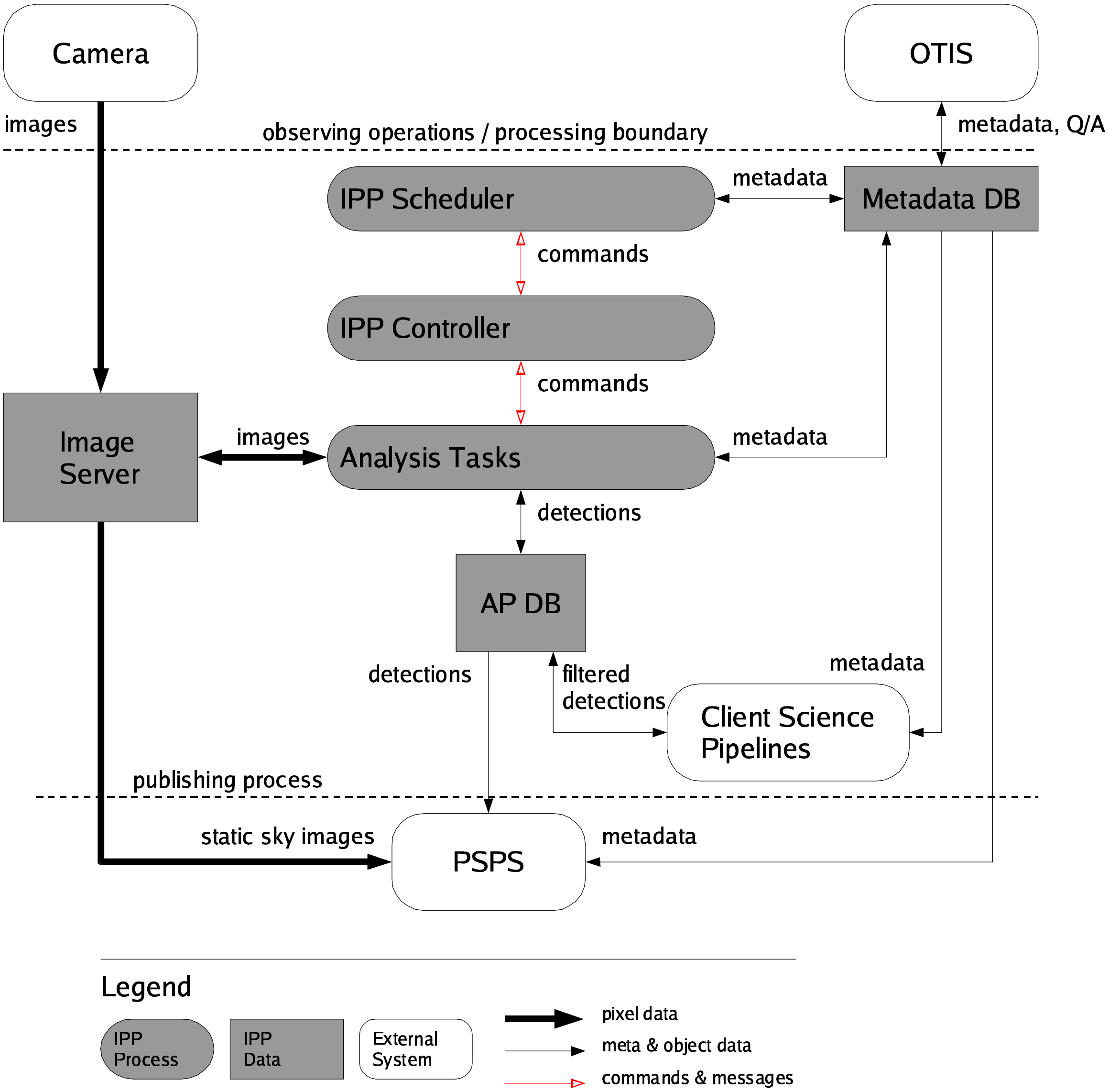

Figure 1: IPP System Overview

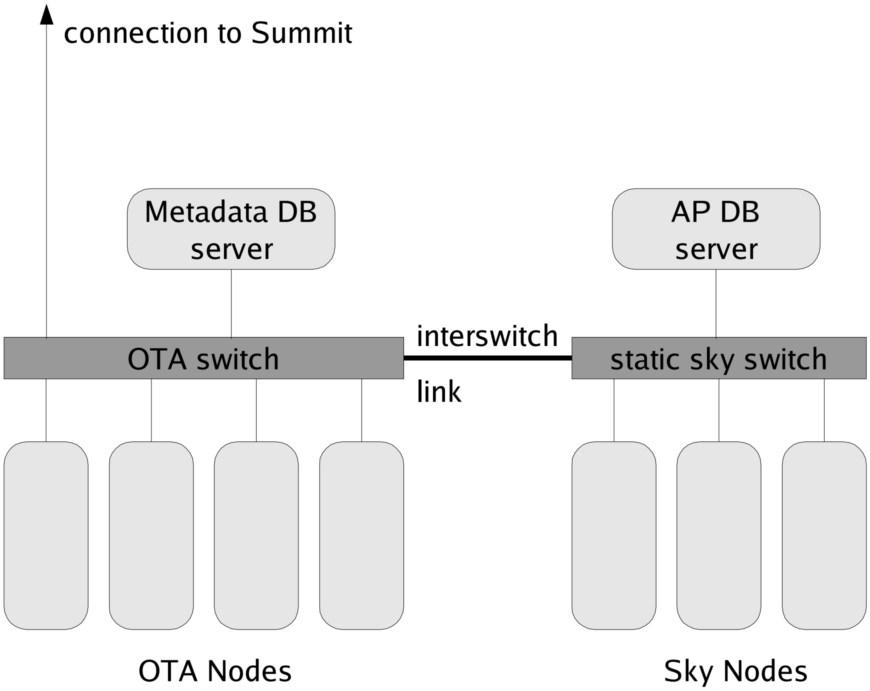

Figure 2: IPP Hardware Organization

which are not well-suited for inclusion in a more structured relational database, and for which access needs to be widely available beyond the individual process which created the file.

- **Metadata Database:** This component stores the data which is not directly related to images or astronomical objects, but which is needed to perform the IPP analyses. The metadata may include the summary weather information for each night, or details about the filters, camera, telescopes, etc. Note that the IPP Metadata Database is not required to retain all archival engineering data from all of Pan-STARRS; other Pan-STARRS subsystems use their own internal databases to store engineering metadata and only the necessary subset is transferred to the IPP Metadata Database.
- **Astrometry & Photometry Database (AP DB):** This component stores and manipulates astronomical objects detected in various images, as identified above, including individual measurements of objects on the images, the summary information about those objects, and reference object data. It also provides mechanisms for users to query and manipulate the objects and detections.
- **IPP Controller:** In order to achieve the required processing throughput for the IPP analysis stages, it is necessary to use distributed computing processes on a large number of computers. The IPP Controller manages the collection of analysis tasks performed on these machines.
- **IPP Scheduler:** This component is a decision-making mechanism which guides the operation of the IPP. It evaluates the currently available collection of data, identifies the necessary analysis, and assigns the analysis tasks to the IPP Controller.

The relationship between these software units is shown in Figure 1. This figure also shows the interactions between the IPP and other Pan-STARRS systems. The architectural components are discussed in detail in Section 4.

3.4 IPP Hardware Organization

The IPP will utilize substantial computer resources, both in terms of computational power and in terms of data storage and network bandwidth. The IPP requires relatively large amounts of data storage space, primarily for the image data. Image data is organized in two categories. First, there is the per-OTA data – data associated with specific OTAs, including the raw images, the calibration images, and temporary processed images at various stages. Second, there is the data associated with the static sky imagery, which is in turn organized into smaller sky-cell units. In addition to image data, there are the somewhat smaller data entities of the Metadata Database and AP Database.

The computer hardware is organized into nodes which provide both data storage and computational resources. The data storage nodes are divided into three classes: those which deal with the per-OTA image data, those that provide the storage for the static sky images, and those that provide the storage for the other data systems, the Metadata Database and the AP Database. In addition, the computational tasks related to the individual images take place on the per-OTA storage nodes and the processing of stacks of images takes place on the static sky storage nodes.

Figure 2 presents the basic concept for the hardware organization for the IPP. This diagram shows the two types of compute nodes: (1) OTA-level processing and storage nodes and (2) Static Sky processing and storage nodes. Also shown are two switches which divide the network into OTA and Static-Sky portions. In such an organization, the inter-switch communication must meet the throughput needs between these network portions (though a single switch may also be used if its backplane capacity is sufficient). The additional data systems (Metadata Database and AP Database) are also shown.

4 System Design : Architectural Components

4.1 IPP Image Server

4.1.1 Corresponding Requirements

The Image Server must meet the requirements specified in Section 3.4.1 of the Pan-STARRS PS-1 IPP SRS (PSDC-430-005). The specified design is chosen to meet requirements 3.4.1.3, and 3.4.1.5. The other three requirements (3.4.1.1, 3.4.1.2, and 3.4.1.4) depend on the volume and capabilities of the hardware, and are addressed in Section 10.

4.1.2 Image Server Overview

The IPP Image Server is a repository for all images and other large data files required by the IPP. Along with the storage hardware, it provides tools for managing the distribution of these large data files and for accessing the files. Data files stored by the IPP Image Server include the raw images, the calibration images, intermediate processing stage images as needed, final processed images, difference images, image subsections, and any large non-imaging data files needed by the IPP. The IPP Image Server must retain the files for as long as they are needed by the IPP.

The IPP Image Server is a parallel storage system. It stores data across a collection of computer nodes, each with their own data storage resources. Any single file is stored on only a single computer and storage device. In order to achieve the data throughput requirements, the IPP Image Server may distribute the images across the processor nodes in an organized fashion, i.e., associating specific machines with specific detectors. It is not the responsibility of the IPP Image Server to determine which computer should be associated with a specific data concept (Chip / region of sky), but it must enable the association of a particular file with a particular machine.

There are three data concepts relevant to the IPP Image Server:

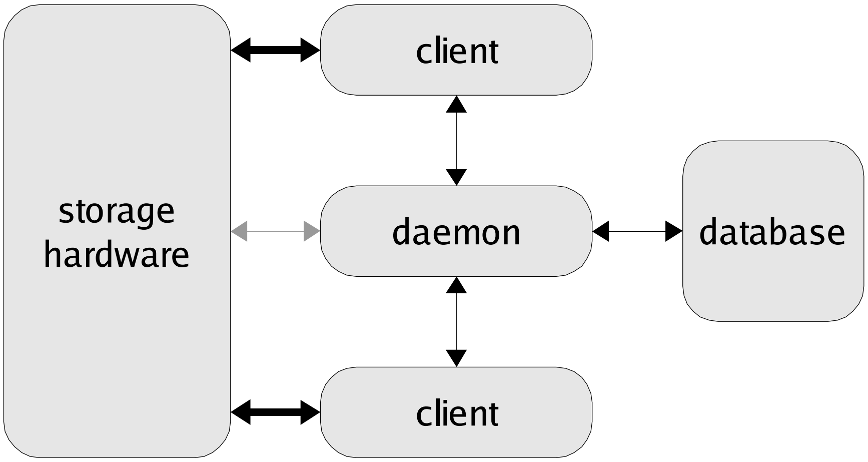

Figure 3: The components of the IPP Image Server.

- **Storage object:** This represents a single, unique data entity in the Image Server.
- **Instance:** A single copy of the storage object in the Image Server. In general, a given storage object may have several instances in the Image Server, normally on different computer nodes.
- **File ID:** This is the identifier of a particular storage object in the Image Server. The file ID is simply a unique string, equivalent to the filename in a UNIX file system.

The Image Server provides file pointers (in C), handles (in Perl or Python), or file names corresponding to the instances of the storage objects. The Image Server provides the data organization but does not define a file system; it assumes the existence of an appropriate file system which makes the files visible as local files. This may be done over many machines with a network file system such as NFS or GFS.

The IPP Image Server provides the storage and access mechanisms, but it does not include any logic or information about the data. The Image Server does not, e.g., monitor the age of images and delete them on some schedule.

As shown in Figure 3, the IPP Image Server consists of the following components:

- Image Server storage hardware
- Image Server database
- Image Server daemon
- Image Server client APIs
- Image Server maintenance tools (not shown)

4.1.3 IPP Image Server Client APIs

Clients interact with the IPP Image Server via a small number of C APIs. Bindings are also provided for Perl and Python and UNIX shell commands in some cases. The client commands are:

- `new object`: create a new storage object in the Image Server. This function takes as input the file ID and returns a C-style file pointer or a Perl file handle to the instance of the storage object. The arguments to the function include an optional node name on which the new storage object must be located. If this target is not given, the Image Server places the new storage object on an appropriate machine from the pool, though the details need to be specified.
- `open object`: open an instance of an existing storage object, as identified by the file ID. This function may also specify the node on which the object should be opened (if an instance of the object is not stored on that node, the function returns an error). On success, the function returns a file pointer.
- `find object`: return a list of filenames in the UNIX name space associated with the storage object identified by the given file ID. Since there are in general multiple instances for a given storage object, this function returns the collection of all available instances. These may be freely opened by the client server using the standard `fopen` functions.
- `stat object`: returns status information about the specified storage object, including the number of instances of the object.
- `replicate object`:a new instance of the given storage object. The target node may be optionally specified, otherwise an appropriate node is selected.
- `cull object`: removes one of the instances of the storage object. The input parameters may optionally specify the target machine to delete.
- `delete object`: deletes all instances of the storage object and sets the storage object status to `deleted`.

4.1.4 IPP Image Server Daemon

The Image Server client requests are mediated via the Image Server daemon. Communication between the clients and the server is via SOAP implementing the commands above. The identity of the machine on which Image Server daemon runs is part of the Image Server configuration information.

4.1.5 IPP Image Server Database

The IPP Image Server daemon uses a database to store the information about the data storage objects, their instances, and the available hardware resources. A `mysql` database engine is used to manage the database table. The database tables defined for the Image Server are listed in Table 2, and their contents are listed in Appendix A. This database engine need not be the same one used for other IPP subsystems.

Table 2: Image Server Database Tables

Table Name	Description
`storage_object`	all storage objects known to Image Server
`instance`	all instances of all storage objects
`volume`	data storage devices known to Image Server

4.1.6 IPP Image Server Storage Hardware

The IPP Image Server manages data across a collection of computers and possibly on multiple storage devices on those computer nodes. The Image Server maintains a table of the available data volumes. The Image Server tracks information about each volume such as the total capacity, the current capacity, the association between computer and data volume.

4.1.7 IPP Image Server Maintenance Tools

The IPP Image Server provides a collection of administration tools which allow for maintenance. These are operations which may be automatically scheduled by the IPP or which may be initiated by a human via a command-shell interface. The maintenance functions include migrating data between nodes to re-balance the available space (this would only occur for instances which have not been placed on a specific node by the client API). Other functions include checking for file corruption, which involves sweeping all files on a data volume and comparing the calculated file checksum to the currently recorded value.

4.2 Metadata Database

4.2.1 Corresponding Requirements

The Metadata Database must meet the requirements specified in Section 3.4.2 of the Pan-STARRS PS-1 IPP SRS (PSDC-430-005). The specified design is chosen to meet requirements 3.4.2.1, 3.4.2.2, 3.4.2.3, 3.4.2.4, 3.4.2.5.

4.2.2 Overview

The IPP Metadata Database acts as a repository for non-pixel data needed by the IPP subsystems. This includes the image metadata, the environmental data, system configuration data and system reference data. The Metadata Database is required to save the non-ephemeral data for the lifetime of the project for future reference and additional analysis. The Metadata Database may be used in close coupling with the analysis pipelines to store temporary data either within or between stages of the analysis. In this scenario, the analysis pipeline will interact directly with the database. However, database latency may make this scenario impractical, in which case the database may be used for long-term storage only. In this scenario, the data produced by analysis pipelines which is destined for the Metadata Database may be collected and inserted by a separate, dedicated process. Metadata which is large in volume or poorly structured may also be stored in an appropriate container file (FITS Table, FITS Header, XML File) in the Image Server with the Metadata DB providing pointers to these files.

The IPP Metadata Database is a simple database system, consisting of a number of simple tables without extensive inter-table links. The `mysql` database engine will be used to drive the database.

4.2.3 Metadata Tables

Table 3 lists the Metadata tables identified to date for the Metadata Database. The contents of these tables are outlined in Appendix B, with examples for the data entries and their data types in many cases. Additional tables will be added as necessary as the data analysis scripts are fleshed out in detail. The Metadata Database, with a flat data organization, is flexible enough to add additional information as it is identified.

4.2.4 Metadata Queries

The IPP provides simple queries to the Metadata Database tables using auto-coded APIs. These queries return a single row or a collection of rows based on the primary key. The format of the API is identical for all Metadata tables. New tables and APIs can be added to the IPP system by adding to the auto-code table description files. The auto-code API includes read and write access permissions to be set for each table independently. See Appendix E for further information.

4.3 AP Database

4.3.1 Corresponding Requirements

The AP Database must meet the requirements specified in Section 3.4.3 of the Pan-STARRS PS-1 IPP SRS (PSDC-430-005). The specified design is chosen to meet requirements 3.4.3.1 and 3.4.3.2. In order to meet the throughput requirements, the AP Database will be distributed across 10 Nodes independent of the Image Server Nodes. An alternative organization of the database which will be studied will have the AP Database co-located with the Image Server Phase 4 Nodes.

4.3.2 Overview

The AP (Astrometry & Photometry) Database is a CSCI which stores data related to astronomical objects derived from various sources with a variety of associations. The AP Database deals with two related concepts: *objects* and *detections*. The *objects* are descriptions of astronomical objects while the *detections* are the specific measurements of those objects, typically measured from astronomical images. A collection of *detections* may be used to derive average quantities which describe a particular *object*. A third class of measurement to be considered are those supplied by external references. Such measurements may be treated as *detections*, with the caveat that access to the raw measurements and metadata are usually unavailable: the reported measurements and errors must be accepted as they are reported.

The AP Database stores the collections of detections which were derived from specific images from any of the analysis stages. It provides a mechanism to determine the image from which a specific detection was derived, and in conjunction with the Image Server locate the corresponding data file. The AP Database also makes it possible to extract all detections derived from a specific image and to determine quantities such as the pixel coordinates of the detection on the image.

The AP Database also has the capability to associate multiple detections of a specific object. Several major classes of objects will be present, each of which must be handled correctly.

First, the most distant stars, compact galaxies, and QSOs will have nearly fixed locations relative to other distant stars, with only small deviations for individual measurements. The association between multiple detections of such objects is made on the basis of their coincident positions. The AP Database determines the average position of the object and the deviations of the individual detections from that average on the basis of the ensemble of individual detection.

Second, solar system objects do not have a fixed location. Detections of such objects are linked by their orbits, and depend on both the position and the time of the image. The AP Database does not attempt to make this link; this is the role of the MOPS system. However, it has the ability to accept identifications made externally with specified detections and to return the identifier of the moving object associated with the specific detections. These associations also include descriptive information such as the offset of the detection from the predicted location of the detection based on the orbit. This functionality is required to allow the AP Database to ignore known moving object detections from other types of queries.

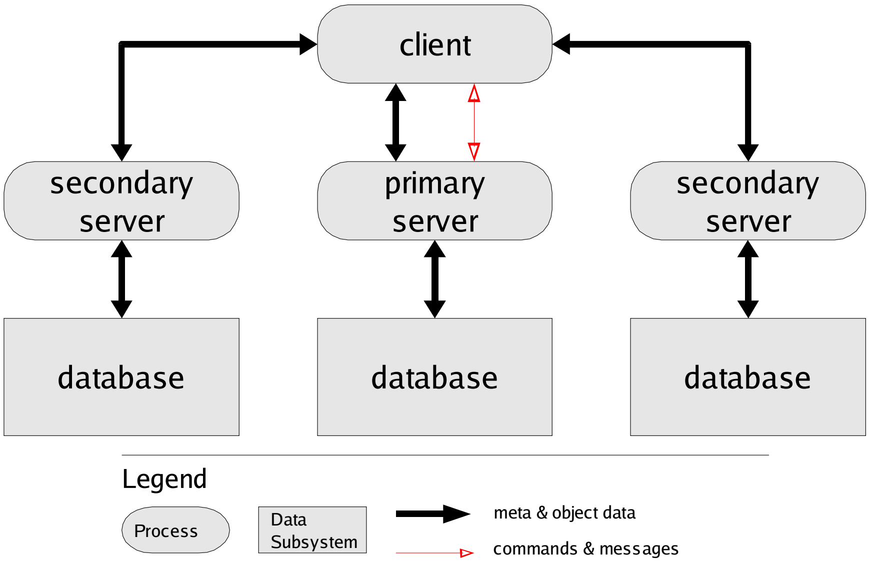

Figure 4: AP DB components

Third, objects in the general vicinity of the solar system fall in between these first two classes of objects. Their proper motion and parallax response is significant enough (> 0.2 arcsec in 1 year) that they are not well-described by an average location and a collection of offsets. These objects are described by a distance and a proper motion vector. The AP Database provides the association between the specific detections and an average object which includes finite parallax and proper motion.

Fourth, many detections, especially in their initial states, will not be associated with a specific astronomical object of any of the above classes and are treated as orphans. Most of these will be spurious (not representing real objects), some will be from solar system objects for which orbits are not yet determined, some will be from faint stars near the detection limits, and some will be from short-term transients which have only been detected once. The AP Database maintains these detections until they have been associated with one of the objects above. The AP Database provides mechanisms by which individual detections may be migrated back and forth between the orphan state and association with an astronomical object.

For every object, and all orphaned detections, the AP Database also provides the capability to determine the images containing the location of the object but for which no detection was made. The minimum set of information which must be carried for these non-detections is the image and the associated object or orphan.

The AP Database also stores the relationships between various photometric systems and the evolution of that relationship. It provides mechanisms to convert between the measured instrumental magnitude of a detection with a specific filter, detector, and telescope, and at a particular time and the implied magnitude in the average Pan-STARRS photometry system, given a determined set of calibrations. It also provides the capability to convert magnitudes in one system to the magnitudes in another system; an example of such a conversion is between the average Pan-STARRS filter systems and the various reference systems appropriate for those filters.

The AP Database provides interfaces to extract lists of objects and detections based on various query parameters. It provides the capability to extract all detections associated with a specific object, all non-detections of that object, all non-detections of an orphan, and summary statistics from these collections. It will also return all objects or detections within specified spatial regions including regions bounded by great circles (RA,DEC; GLAT,GLON; ELAT,ELON) and regions described by a location and a search radius. It will also return the image parameters associated with a specific detection including image coordinates of the detection, exposure time, time and date of the detection, etc.

As shown in Figure 4, the IPP AP Database consists of the following components:

- AP Database database tables
- AP Database database engine
- AP Database servers
- AP Database client APIs

4.3.3 AP Database Tables

Table 4 lists the tables used by the AP Database. The contents of these tables are outlined in Appendix C. Below, the use of these tables by the AP Database software is discussed below. Several of the tables are not just simple tables in the database but are instead table groups divided into many subtables, each of which represents a portion of the sky (a `region`). These subtables may also be distributed across different computers to distribute the processing load.

4.3.3.1 Images Table Group The `Images` table group lists all of the images which provided the data in the AP Database. These tables are subdivided by region on the sky. In general, the images listed in this table correspond to the Chips. This group of tables includes sufficient astrometric parameters to represent the coordinates of the detections to a sufficient accuracy. Parallel to the Images table is the Mosaic table. This table is very similar to the Images table, but defines the Mosaic which corresponds to a group of Images. The parameters include the astrometric information needed to define the camera distortion.

4.3.3.2 Image Overlaps Table Group The specific subtable of `Images` which contains a given image is the one which contains the center pixel of that image. An additional table group, `Image Overlaps` (with the same subtable organization as the `Images` subtables), lists images which overlap that specific subtable. Thus, given a particular coordinate, in order to find that images which overlap that coordinate, it is necessary to search the images in the `Images` subtable which includes that coordinate, and all images in the `ImageOverlaps` subtable for that coordinate.

4.3.3.3 Objects Table Group The `Objects` table group (also divided by region) stores the average parameters for each astronomical object. Certain details of this table have not yet been specified. In particular, objects with significant parallax and/or proper motion may potentially be stored in a distinct table. Solar system object identifications, to the extent average properties are maintained in the AP Database, will certainly be stored in a separate table.

4.3.3.4 Average Magnitudes Table Group A related table, also divided into the same regions, is the `Average Magnitudes` table. In this table, there are multiple rows per object, one for each of the primary filters of interest for which photometric averaging is performed. This organization makes the number of primary (averaged) filters a configurable value.

4.3.3.5 Matched Detections Table Group The `Matched Detections` table stores all of the measurements of astronomical objects on specific images. This table includes all detections associated with the average `Objects`. As discussed below, bright objects (above a configuration-specified signal-to-noise level) are defined object even if only one detection has been found at that position. Faint orphaned objects are not added to this list or the list of objects. The

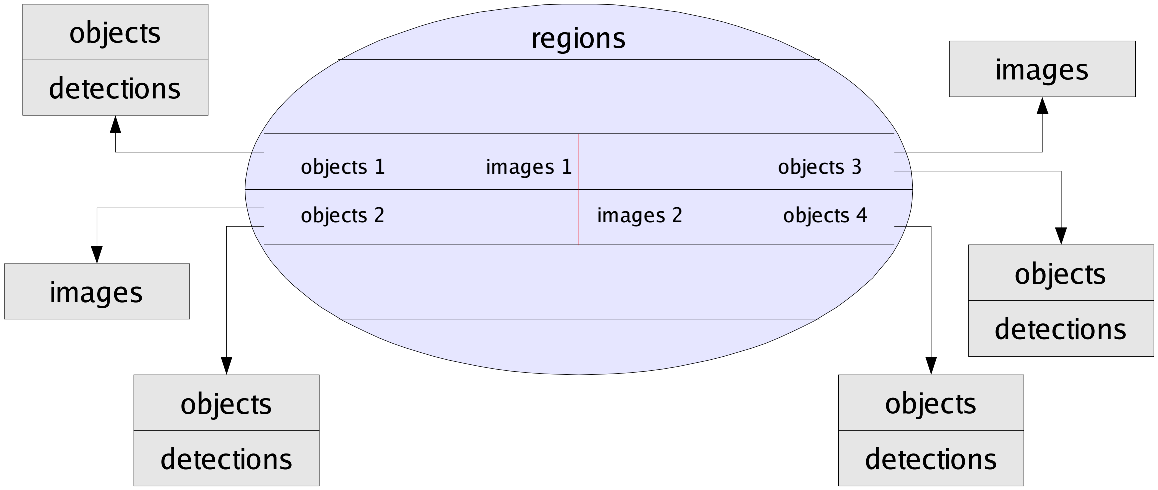

Figure 5: AP DB Regions and Image / Object tables

different types of detections (P2, P4Δ, P4Σ) are distinguished by their photometry codes. (This is only valid if the AP Database does not store different quantities for these types of detections.)

4.3.3.6 Orphaned Detections Table Group The `Orphaned Detections` table stores the detections which have not been correlated with an existing object. This table is only populated for objects below a configuration-specified signal-to-noise limit (e.g., 5σ). Bright orphaned detections are assigned an object and added to the `Matched Detections` table.

4.3.3.7 Non-detections Table Group The `Non-detections` table stores information about detection failures for each object. If an image is added to the database which overlaps an object but the object is not detected, an entry is made in this table. In practice, this table may store only the most recent non-detection and the total number, or a similar reduced set of non-detection statistics.

4.3.3.8 Regions Table The `Regions` table is used to subdivide the tables of images, objects, and detections, etc, as discussed above. The AP Database divides the sky into a hierarchy of regions (portions of the sky) each of which is in turn subdivided into smaller portions. Since nearly all interactions with the AP Database performed by the IPP are limited in spatial coverage, subdividing the tables allows a specific interaction to search only a small subset of the data. The table of images is the smallest of the three; the table of detections is likely to be the largest. As a result, the `Images` table group will be subdivided at a shallow hierarchical level, while the `Objects` and `Detections` are subdivided on deeper (more finely sampled) levels. The `Regions` table defines the boundaries of the sky regions and specifies if the region corresponds to an `Images` table, an `Objects` table, and/or a `Detections` table. It also specifies which regions in the next level of the hierarchy are contained by the region, and which parent region it belongs to. In addition to improving the spatial access to the image, object, and detection data, the `Regions` table allows for multiple computers to serve the database tables. The region file specifies the machine which stores the specific table. Figure 5 illustrates schematically the subdivision of the sky and the association between different levels of the hierarchy with different subtables.

4.3.3.9 Other Reference Tables The `Filters` table identifies all of the physical filters (specific pieces of glass) known to the system. A related table, `Photcodes`, defines relationships between photometry systems. A photometry system may consist of a detector, telescope, and specific filter, or it may be a derived photometry system. The `Database Machines` table identifies all of the computers available to the AP Database.

4.3.4 AP Database servers

The AP Database functions on a group of computers, with portions of the tables stored on separate machines, as described above. The association between a machine and the corresponding table or part of the sky is defined by the Region table. Each machine has a corresponding AP Database server which runs on that machine to interact with the tables available on that machine. Two possible interaction models are considered.

Option A: A client chooses one of the machines and sends its query or data to that machine. The server then uses the region table to determine which machines contain the relevant portion of the sky. Data to be added to the database is divided into corresponding region chunks and sent to the appropriate servers. Queries are redirected to the appropriate server(s). The original server may collect the results and return them to the original client.

Option B: The client downloads the region table and performs the division of the data into appropriate subsets. The subsets are then sent to the corresponding servers by the client.

The differences between these models is small. The first option may make the code more testable, placing all of the logic in the servers and making each server symmetric. The smaller tables (ie, Region, Filters, etc) could either be downloaded from a single server or replicated to all AP DB servers. For these reasons, Option A will be used for the PS-1 IPP..

4.3.5 AP Database engine

The backend database engine for the AP Database stores the tables and provides them to the servers on demand. The AP Database will use a `mysql` database engine for this function.

4.3.6 AP DB Client operations

The AP Database client interactions consist of a collection of basic queries of the database, along with more complex operations to perform particular tasks. The complex operations are listed below.

4.3.6.1 Insert Image & Detection Set (addstar) One of the most basic operations needed by the AP Database is to insert a collection of detections derived from a specific image, and add the definition of that image to the database. This operation is critical in terms of the processing throughput. After the detections have been assigned to the appropriate regions, they are matched against all objects in the `Objects` table. Matches are performed only on the basis of positional coincidence, using a matching radius which may depend on the image astrometry errors, or may be a fixed distance. Any matched detections are added to the `Matched Detections` table. Any unmatched detections brighter than the Faint Detection cut-off are specified as a new `Object` and also added to the `Matched Detections` table. Any faint unmatched detections are added to the `Orphaned Detections` table. This division is important because it allows the automatic association of new detections with existing bright objects while limiting the I/O volume required to make the detections. In general, there will be many fewer `Objects` than `Detections`, and there will be fewer bright orphans than faint orphans.

4.3.6.2 Insert Reference Objects (addrefs) This operation is very similar to the previous one. A collection of reference objects are added to the database as a collection of detections. The reference photometry should in general be given its own photometry code. The reference data is different from the image detection set because the associated image information is not included. Thus, no corresponding images are added to the database.

4.3.6.3 Determine Relative Photometry in region (relphot) This operation uses the overlaps of images and multiple observations of the same objects to determine the relative photometry zero-points for a collection of images. This is a task that wil be run much more infrequently than the object insertion tasks.

4.3.6.4 Determine Consistent Photometry Zero Points (uniphot) This operation uses the time history of relative photometry zero points for images and the spatial overlap information to determine a best set of image zero points which have a specific time scale for the atmospheric stability.

4.3.6.5 Determine Distortion and Static Astrometry Model (mosastro) This operation uses the reference and image detections to determine an optical distortion model for the camera and static astrometry model components. The astrometry model includes: (1) field distortion introduced by the telescope optics, which is a smoothly-varying function of the field position relative to the center of the telescope boresite coordinates. (2) focal plane geometry, which includes the chip positions and rotations in the focal relative to the boresite, along with chip-dependent plate-scale modifications needed to represent tilts or warps of the individual detectors relative to the ideal flat focal plane. .

4.3.7 Throughput

The AP Database design partly driven by the need to make the detection-object associations quickly and to processes the incoming detections at a sufficiently high rate to meet the throughput requirements. For each upload of the object detections from a complete FPA, the AP Database must match roughly 1.4×10^6 detections from an FPA with roughly 6.4×10^6 objects, including orphaned bright detections. This corresponds to roughly 640 MB, if each object uses 100 bytes for its descriptive informations (more than is currently specified in the Object table). With a throughput of 100 MB/s for reads from a RAID, the AP Database can perform the data read in a fraction of a second if the data is distributed across 10 computers.

4.4 Controller

4.4.1 Corresponding Requirements

The Controller must meet the requirements specified in Section 3.4.4 of the Pan-STARRS PS-1 IPP SRS (PSDC-430-005). The design must meet requirements 3.4.4.1 - 3.4.4.7. In particular, the Controller / Node Agent architecture is chosen to control the I/O flow between the Controller and the individual processes so that blocking on the I/O from many remote processes does not saturate the Controller processing.

4.4.2 Overview

The IPP uses a group of computers to store and process images and to manipulate collections of detections. These computers perform any of a large number of analysis stages or other processing tasks without significant interprocess

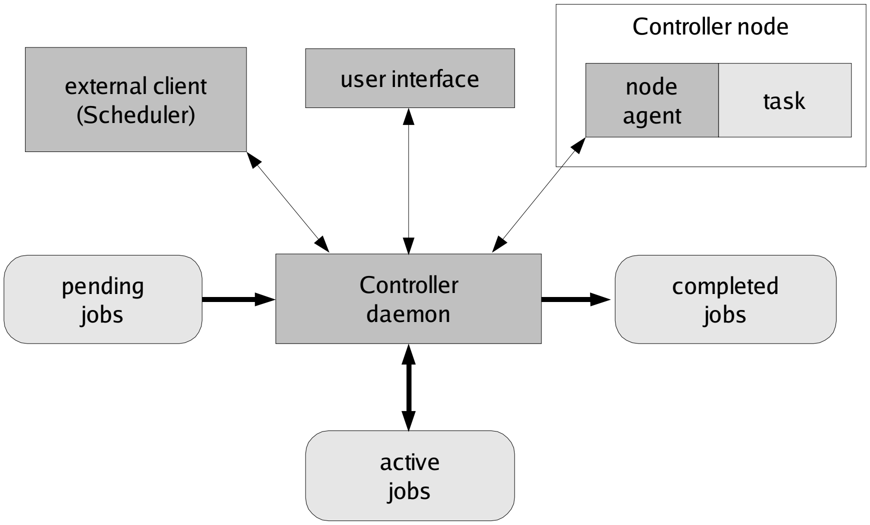

Figure 6: Schematic illustration of the Controller components

communication. It is necessary to have a mechanism which initiates computing tasks on the different computers, which monitors the tasks as they are executed, which handles the output and the errors from these tasks, and which reacts to the failure of any of the computing nodes. The system responsible for the tasks in the IPP is the IPP Controller.

The IPP Controller interacts with the collection of computers under its management and with other subsystems in the IPP. The IPP Controller receives a variety of inputs from other subsystems, described below, and initiates actions such as adding a new process to the queue of pending tasks. The IPP Controller also provides information to other subsystems on demand about its processing history and current state. Each physical computer may have multiple processors; since the IPP Controller is managing processing tasks, it treats each processor independently. It is up to the system configuration if each computer needs to reserve one of its CPUs to manage background tasks or if the IPP Controller should attempt to send one task per CPU and let the operating system handle the I/O load. The relationship between the different components of the Controller is illustrated in Figure 6 and discussed below.

4.4.3 Nodes

The Controller maintains a table of available processing computers (‘Nodes’) and tracks the status of these Nodes. Nodes managed by the IPP Controller are allowed to be in one of several states, and the IPP Controller must interact with it in an appropriate way for each of those states. A Node may be `alive`, `dead` or `off`. If the Node is `alive`, it responds to commands from the IPP Controller and may be used for tasks subject to other constraints. If it is `dead`, the Node is not responsive and must not be used for executing tasks. The IPP Controller must identify Nodes which have died (not responding) and occasionally test them to see if they are `alive` again. Nodes which are `off` are not available for tasks and must not be tested. Nodes may be set to the `off` or `dead` states by external subsystems; it is the responsibility of the IPP Controller to return a Node to the `alive` state if possible.

The IPP Controller must honor requests (normally from the users) to change the mode of any computing node on demand between `off` and `dead`. This would normally be done after a Node has been rebooted and is released to the IPP Controller for its use. It must also be able to change the list of allowed tasks as requested by external commands.

Two example scenarios illustrate the transition between these states, and the basic concept of operations for the IPP Controller. First, imagine a computer crashes. At this point the IPP Controller should detect that the Node is no longer responsive and mark it as `dead`. It should occasionally try to re-establish communication with the Node, potentially with longer and longer delays between attempts. A human could be notified if the Node seems to remain `dead` for a very long time. In another scenario, a person needs to work on a Node. They notify the IPP Controller that the machine is `off`, perhaps with a prior notification that the machine should be prepared to go off. When work on the machine is complete, it should be placed in the `dead` state. Only when the person is done working and testing the machine, and tells the IPP Controller that the machine is now `dead` can the IPP Controller attempt to re-start communications and re-new processing operations on that Node.

4.4.4 Node Agents

When the Controller starts, it attempts to launch a Node Agent on each of the available processing Nodes. Nodes which are not responsive are marked as `dead` so they may be re-tried. A Node Agent runs on each of the individual nodes to execute the tasks as directed by the Controller. The Node Agents communicate with the Controller via a socket connection.

A Node Agent (which is only running on a Node in the `alive` state) may be in one of four modes: `idle`, `busy`, `done`, `crash`. A Node Agent which is `busy` currently has a task assigned to it which is executing. The IPP Controller may only assign one task to a Node at a time. A Node Agent which is in the `idle` state may have a task assigned to it. When the Node Agent detects that a tasks has finished, it changes to either the `done` or `crash` states depending on the outcome of the process execution. The IPP Controller must also respect a list of task restrictions which may require specific tasks to run on specific CPUs or exclude specific tasks from specific CPUs.

A task being executed by the Node is run in the UNIX user space as a forked process. The Node Agent must monitor the standard error and standard output of the executing task and save them in separate buffers. If the process exits or dies, the Node Agent must detect this result and change state appropriately. The Node Agent must respond to various commands from the Controller, as follows:

4.4.4.1 Report status The Node Agent returns its state (`idle`, `busy`, `done`, `crash`) and the exit status of the current processing task, if available. The reported exit state, if the process has completed without crashing, is the UNIX exit state reported by the task: 0–256 with 0 indicating a successful completion.

4.4.4.2 Report stdout Send and flush the current stdout buffer. The Node Agent will return the complete contents of the stdout buffer via a buffered write and flush the buffer when it is finished. The Node Agent will not accept more data on the stdout buffer from the current processing task until the send is complete and the buffer is flushed. The daemon must accept all of the buffer output.

4.4.4.3 Report stderr Identical to ‘report stdout’, but for stderr.

4.4.4.4 Kill task The Node Agent should send a kill signal (`KILL` or `TERM`) to the current processing task. When the processing task has exited, the Node Agent should set its state to `crash`.

4.4.4.5 Clear task The Node Agent should set its state `idle`. If a processing stage is currently running, it should be killed (`KILL` or `TERM`) before the task is cleared.

4.4.4.6 Start processing stage The Node Agent forks a specified command. The command should be a standard UNIX command without command line redirection or backgrounding. The task is run with the same user ID as the Node Agent, which is also the same user ID as the Controller.

4.4.5 Tasks

The IPP Controller accepts tasks from other IPP subsystems. The task requests include the specific command to be executed and are in the form of a UNIX command which could be performed on any of the computing nodes. Any input or output data in the commands must be a valid resource regardless of the node on which the task is executed. Input and output data resources must be unique where necessary to avoid conflicts. It is the responsibility of the task to wait for network lags (ie, NFS delays). The IPP Controller gives each task a unique identifier, which is returned to the requesting entity. The requestor may then use that ID to obtain status information on that task or to send control signals to the specific task.

Task requests may specify a desired node for the task execution. The IPP Controller attempts to honor the request if the node is `alive`, but will execute it on another node if the requested one is `dead` or `off`. Even if a node is `alive`, the IPP Controller will choose another node if the specified task is not allowed on the requested node. In all other cases, the IPP Controller waits until the currently executing processes, and processes with higher priority, are completed before executing the specified task on the requested node.

Task requests may specify an urgency level. The IPP Controller determines the priority of the task on the basis of both the urgency and the age of the request. An executing task must be completed on a CPU before any new task is started on that CPU, regardless of priority. The urgency levels range from 0 to 2. Tasks with an urgency of 1 are scheduled whenever they reach the top of the stack. Tasks with an urgency of 2 are sent immediately to the top of the stack. Tasks assigned a priority of 0 are maintained in the queue and never executed.

It may be useful for the Controller to distinguish between tasks dominated by I/O and tasks dominated by data processing. It is possible that one of each of these types of tasks may be sent to the same node without significantly impacting the system performance. Alternatively, it may be necessary to limit a single machine with 2 CPUs to only one of each of these types of tasks (i.e., one processor will be working on I/O while the other is working on processing). Such details will be studied by the IfA IPP Team.

The IPP Controller monitors the output streams from the executing tasks and the exit status of the tasks. Each task is associated with a log file, to which all output is written. The status, including the exit status, of each task is maintained by the IPP Controller so that other subsystems may determine if specific tasks have started or completed.

4.4.6 Controller Interfaces

The IPP Controller must accept commands from other IPP subsystems. These commands include those which govern the processing of specified tasks, those which govern the behavior of specific computing nodes, and those which request information from the IPP Controller. The IPP Controller must be able to halt the execution of a specified task, delete an unexecuted task from the task list, change the priority of tasks, and change the requested nodes for tasks. The IPP Controller must also be able to stop the current execution of a task and push it to the end of the queue and also change its priority.

The IPP Controller must respond to informational requests regarding the collection of machines and their states as well as the collection of tasks and their states. The IPP Controller must monitor the execution times of the different tasks and provide summary statistics. Finally, the IPP Controller must respond to three top-level commands: `finish`, `stop` and `abort`. When `finish` is requested, no more new tasks are accepted on the stack of task, and when all tasks in the stack have completed, the IPP Controller must exit. When `stop` is requested, the currently executing tasks must be completed at which point the IPP Controller must exit, but tasks remaining in the stack which have not been started are flushed. When `abort` is issued, the IPP Controller immediately kills all executing tasks and exits.

The IPP Controller and the IPP Image Server have related needs for information from the combined storage-and-processing nodes regarding which nodes are available. It is not yet clear if this information is best stored in a single location (either IPP Controller or IPP Image Server), which provides the information to other systems on demand, or if both systems should maintain the information. Also, it may be necessary to distinguish nodes which are available for processing from those that are available to serve data as part of the IPP Image Server.

The Controller maintains three tables of processing jobs: pending stages, active stages, and completed stages. The pending stages are those which have not yet been performed. The active stages are those currently being performed on one of the remote nodes. The completed stages are those which have finished, either successfully or with an error state. The Controller daemon monitors the collection of remote clients and sends them new pending stages when they become free.

The IPP Controller provides a mechanism for users (either other programs or humans) to interact with it. The user interface provides commands to check the current processing job queues, the tables of successful and failed jobs, to stop or delete jobs, etc.

4.5 Scheduler

4.5.1 Corresponding Requirements

The Scheduler must meet the requirements specified in Section 3.4.5 of the Pan-STARRS PS-1 IPP SRS (PSDC-430-005). The design must meet requirements 3.4.5.1 - 3.4.5.7. In particular, the Task / Test division is chosen to prevent the Scheduler from blocking while an analysis process is performed. Scheduling requirements will be met by defining appropriate Test periods for the different Tasks.

4.5.2 Overview

The IPP is responsible for a variety of analysis jobs: processing of the science images through several stages; routine assessment of the detrend (instrumental calibration) images used in processing the science images; construction of replacement detrend images when needed; generation of astrometric and photometric reference catalogs based on the collected dataset; and the performance of test analysis programs. At any point, decisions need to be made about which of these tasks should be performed, based on an analysis of the contents of the metadata database, the requirements of the people monitoring the IPP, and the near-term observing plans. The IPP Scheduler is the mechanism that assesses these various inputs to guide the decisions and initiate the actions.

The IPP Scheduler acts as an interface between several components of the IPP and also between the IPP and external agents such as OTIS and the users who must monitor the behavior of the IPP. The IPP Scheduler may be viewed as the central brain of the IPP. Figure 7 illustrates the design of the IPP Scheduler.

4.5.3 Scheduler Tasks and Tests

The IPP Scheduler performs two types of actions. ’Tasks’ are long-running programs which are executed by the Controller. These are not only background tasks, but are distributed computing tasks. Examples of these include the science analysis

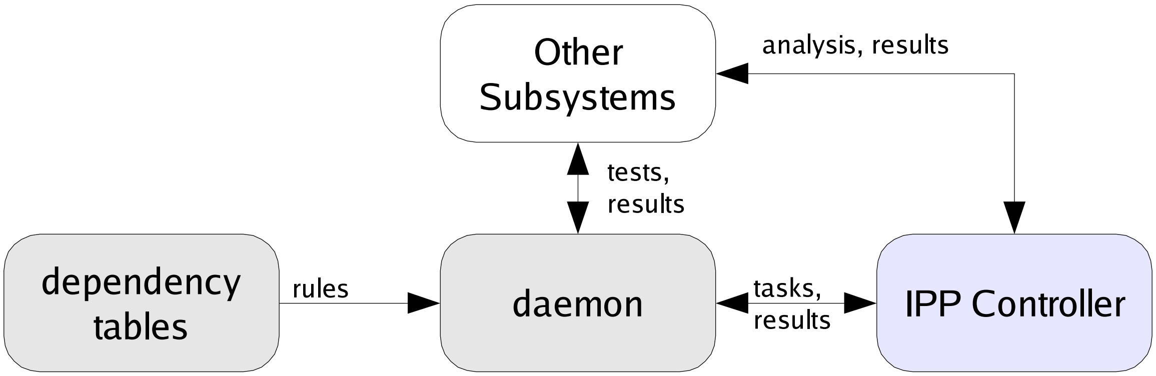

Figure 7: IPP Scheduler

tasks (eg, Phase 1, 2, 3, 4), the Calibration construction tasks, and data copy tasks (such as copying images and metadata from the summit system). 'Tests' are short-running programs which are used to decide which tasks should be run. Tests should be designed to return immediately ($< 100ms$) and are not run in the background; the Scheduler will block until the test is complete. The IPP Scheduler daemon, which runs continuously, performs tests (eg, queries of the IPP Metadata Database, queries of OTIS, checks of the IPP hardware status, etc). Based on these tests, the daemon defines appropriate tasks and sends them to the Controller. When tasks are completed, their results may be used by the Scheduler to update the external systems (update the Metadata Database), or the tasks themselves may send their results directly to the Metadata Database or other subsystems. Based on the successful completion (or not!) of the tasks, and the new state of entries in the Metadata Database, the Scheduler can define new tasks.

The IPP Scheduler sends tasks to the IPP Controller for execution. While the IPP Scheduler chooses the tasks to be performed, it is the IPP Controller's responsibility to manage the specific tasks executing on a given processing node. This division of responsibilities allows the different functionalities of the IPP Scheduler and the IPP Controller to be isolated and encapsulated. With this separation, the IPP Controller does not information about the details of the tasks it executes, while the IPP Scheduler does not need to monitor the computer hardware.

Communication between the IPP Scheduler and the IPP Controller is bi-directional; the IPP Scheduler sends tasks to the IPP Controller, while the IPP Controller informs the IPP Scheduler of the outcome of those tasks. For the PS-1 IPP, the IPP Scheduler and the IPP Controller are distinct, interacting software components. The interface mechanisms are described in Section 9.

4.5.4 Task Rules

The IPP Scheduler takes as input a collection of rules which define the dependency of tasks on certain tests. The IPP Scheduler must choose between several types of analysis tasks based on those rules and on results of the tests. The timescale on which different tasks (and their related tests) are executed may vary from 10s of seconds to hours, days, or even as long as a week. The list of tasks which the IPP Scheduler must decide between, and the relevant timescale, follow:

- moving data from the Summit pixel server ($\sim$ 30 second timescales)
- running the science analysis stages ($\sim$ 30 second timescales)
- testing the validity of the current detrend images ($\sim$ nightly)
- constructing new detrend images ($\sim$ weekly)

The scheduler may be viewed as a complex state machine. The goal is to design the scheduler so that rules may be specified independently from the engine which parses the rules to determine which specific jobs to send to the controller.

4.5.5 User Interface

The IPP Scheduler shall possess a user interface which allows a human operator, or other processes, to monitor the current state of the Scheduler. Users have the option to specify that a particular task or set of tasks is of higher or lower urgency (as defined in Section 4.4) than the norm, or to schedule a particular tasks on a different timescale from the basic rule.

The IPP Scheduler defines the operating state of the IPP and shares the same set of states:

- active state
- interactive state
- paused state

When the IPP Scheduler is in the *active state*, it performs the most appropriate of all possible tasks at a particular time. When the IPP Scheduler is in the *interactive state*, it performs only a specific requested action regardless of the outcome of the decision trees. In addition, in the interactive state, the IPP Scheduler must only perform the requested actions and not attempt to perform the other normally-required actions. The only exception to this exclusion is that, in the interactive state, data is still copied from the summit system. An additional IPP state is the *paused state*, intended for tests or maintenance, in which case the IPP Scheduler does not perform even the data copy tasks. Every task is performed on demand by the user. A user command sets the IPP Scheduler in one of these three states, *active*, *interactive*, and *paused*.

5 System Design : Science Analysis Tasks and Stages

This section describes the design of the science analysis stages which perform the fundamental image analysis steps of the IPP. The IPP science image processing stages perform analyses on the night-sky science images to extract the science data from these images. These consist of:

- Phase 1, the image processing preparation stage,
- Phase 2, the image reduction stage
- Phase 3, the exposure analysis stage
- Phase 4, the image combination stage.

These analysis tasks must process the images in a timely manner so that the incoming data stream will not overload the IPP Image Server. The decision to execute a specific pipeline for a specific dataset is made by the Scheduler, which sends the information to the Controller. The Controller executes the pipeline for the data on an appropriate machine and monitors the success or failure of the processing stage.

The analysis stages are written as UNIX commands, which may be executed by the IPP Controller, or may be executed individually by hand. This option makes testing of the complete analysis system much easier because the individual analysis stages may be tested independently of each other and the IPP infrastructure.

As part of this design model, the analysis stages have several methods for accepting and returning the input and output data and for defining optional choices in the analysis. All of the analysis stages load an analysis recipe, which defines the details of that analysis. The recipe includes the location of the data sources (from the metadata, from the image headers, from other external files, or supplied directly), which steps to employ, and how to assign optional parameters. For example, in the discussion of the Phase 2 analysis below, the recipe file may specify *if* a bias subtraction should be applied, *where* to find the overscan region and *which* bias image, *if any*, to apply.

The recipe is loaded as part of the runtime configuration information loaded when the analysis script starts. Four levels of runtime configuration information are defined. The `site` configuration defines values specific to the particular installation of the software. For example, the name of the machine which hosts the Metadata Database or a default path for data files could be part of the `site` configuration. Multiple installations or versions of the IPP software would need to have separate `site` configuration entries. For example, a version of the IPP installed at the IfA would use a different computer for the Image Server from the live IPP installation running on the Pan-STARRS cluster. The `base` configuration defines general data sources which may be needed by any portion of the IPP. The list of known telescopes or filters might be an example. The `camera` configuration consists of information which defines the parameters relevant to the cameras known by the IPP. For example, the default layout of the detectors or the names of specific header keyword values would be defined for each camera in a camera-specific configuration collection. Finally, each analysis script loads its own recipe. The location of this configuration information may be a collection of configuration files available on disk or some subset of the information may be stored in the Metadata Database. The source of these configuration entries can be overridden when the script is executed, and individual configuration values may also be specified on the command line. Examples of the recipe and other runtime configuration options are given in Appendix D.

5.1 Phase 1: image processing preparation

The Phase 1 analysis stage is performed on each science exposure (each complete FPA image) to calculate basic astrometric data needed by the later stages. Phase 1 uses the static (pre-determined) telescope distortion model and a table of nominal OTA positions and rotations, combined with the guide star pixel and celestial coordinates, to determine the correct telescope bore-sight, field rotation and magnification. The guide star coordinates are loaded from the Metadata database. These calculations are performed by comparing the observed guide star detector coordinates with the known astrometric positions of these same stars as reported by an external astrometric reference. The accuracy of the resulting astrometric solution is expected to be ~ 1 arcsec across the field, sufficient in later stages to match the vast majority of astrometric reference stars with their detections with minimal effort.

In some circumstances, science images may have no guide stars. This may occur in the Pan-STARRS system if the detectors are not run in OTA mode, for example for short snapshot images. This may also be the case if the IPP is being run on non-Pan-STARRS data. In such a circumstance, the Phase 1 stage uses the provided boresight coordinates, exposure time, and camera zero-point to predict the pixel coordinates of known bright stars expected to be found on the detectors. It then extracts a large box ($\sim 30 \times 30''$) around these locations and performs extremely basic object detection to determine the detector coordinates of those bright stars which are not saturated but which are significantly above the background level. By targeting known locations in the image files, only a small amount of data will have to be read.

If the image has invalid coordinates or no detectable bright stars, Phase 1 fails and reports a descriptive error.

Given the above astrometric solution, the Phase 1 analysis stage constructs a table of the overlaps between the science image to be processed and the static sky images that must be constructed. This table will be used to guide the processing of the static sky in Phase 4. The overlaps should be generously calculated so that small errors in astrometry at Phase 1 will not cause any valid static sky / science image pairs to be missed because of the astrometric error at this phase. It is acceptable for a small number of invalid overlaps to be identified as these will be excluded in Phase 4. Static Sky cells which do not have sufficient science image overlap ($< 5\%$) need not be processed because the few new measured pixels do not add significantly to the Static Sky.

5.1.1 Examples

Examples of Phase 1 as called from the command line, with different types of images:

```
Phase1 -file filename.fits [FPA is single fits file]
Phase1 -list filename.list [FPA is collection of files]
Phase1 -imdb ID            [FPA is single file in image server]
Phase1 -FPA  ID            [FPA identifier from metadata db]
```

5.2 Phase 2 : image reduction

5.2.1 Overview

Phase 2 processing within the Pan-STARRS image processing pipeline is the detrend stage, where the images from the detector are processed to remove instrumental signatures. This analysis is performed on individual chips, which can be identified as the data entity which has a single, continuous astrometric solution.

Phase 2 consists of the following operations, some of which as noted may be skipped by the recipe:

- Load science image
- Identify appropriate detrend images
- Load detrend images
- Form OT kernel
- Convolve detrend images with the OT kernel
- Bias/dark/overscan subtraction
- Mask bad pixels
- Trim overscan
- Non-linearity correction
- Flat-field
- Mask diffraction spikes and optical ghosts
- Subtract sky
- Find and photometer objects in the image
- Identify CRs by morphology
- Determine PSF model
- Improve astrometry
- Extract Bright object postage stamps

The steps are explained in detail below.

5.2.2 Load Images

The Phase 2 analysis must load the science image to be analyzed into memory, as well as the corresponding metadata (either from the image header and/or from the IPP Metadata Database). It must use the metadata for the image, along with information from the processing recipe, to determine the appropriate detrend images to be used for this analysis. The Metadata Database stores the information necessary to associate a specific science image with one of the registered master detrend images for each type. These images are also loaded by the Phase 2 analysis (note that the design of Phase 2 may perform the actual loading of pixels in blocks or groups to minimize the memory impact).

5.2.3 Form OT Kernel & Convolve with Detrend Images

Science images which have been obtained with Orthogonal-Transfer Guiding have had their pixel response smoothed by the image correction motion. For these images, some of the detrend images need to be convolved by the same OT kernel, so that they accurately represent the effective pixel response. The detrend images which must be convolved include: the flat-field image, the high-spatial-frequency fringe images, and dark images, if they are used. The appropriate kernel for each cell of an OTA must be determined from the guide star history, extracted from the IPP Metadata Database or from the image header. If the OT kernel is not available, but the image metadata notes that it should be, the convolution is skipped, with a warning.

The convolution method depends on the size and structure of the OT kernel. If the kernel is small ($< 5x5$ pixels), direct convolution may be employed. If the kernel is large, but may be decomposed using Gaussians, then it may be convolved using a decomposition method.

The module convolves each of the dark frame, flat-field, and the fringe frame(s) by the OT convolution kernel. Specific flags in the static bad pixel mask are also grown by the outline of the OT convolution kernel (see Section 5.2.4.0.1).

5.2.4 Bias Correction / Overscan Subtraction

The image bias must be subtracted. Since different detectors behave in different ways, several options for modeling the bias are available. The bias is measured from the image overscan region. The bias subtraction method must be capable of subtracting a single constant from the complete image, or to subtract a 1-D bias which varies as a function along the overscan. The function used to represent the overscan region may be a spline or a Chebychev polynomial derived from the data values along the overscan. The values used to determine both the single constant or the inputs to the spline and polynomial fits are derived from groups of pixels on the basis of one of several statistics, including the sample and robust mean, median, and modes. In the case of a single constant, all of the overscan pixel values are used in the calculation of this statistic. In the case of the 1-D functional representation, the input values to the fit must represent the coordinate along the overscan, with the statistic derived from the pixels in the perpendicular direction at each location. Sigma-clipping on the input data values must be an option.

5.2.4.0.1 Flag bad and saturated pixels A static bad pixel mask is used to identify pixels which are known to be bad in the camera. This mask is then processed with the science image. Bad pixels which are charge traps are grown by the extent of the OT convolution kernel. Bad pixels above a charge trap (i.e. bad columns) must not be grown, since they were not affected by pixel shifting, but only became bad at read-out.

Pixels which are saturated in the A/D converter, or with a signal level at which the response is excessively non-linear, must also be masked, and this area must be grown by an additional pixel to mask excess charge spillover.

The bad pixel mask is carried with the science images. Different bits are set to identify different reasons for masking the pixel. Flags are required for at least each of the following pixel attributes:

- The pixel is a charge trap;
- The pixel is a bad column;
- The pixel is saturated in the A/D converter;
- The pixel is non-positive in the flat-field;
- The pixel is part of a row that has excess noise; and
- The pixel is determined to be a cosmic ray, based on its morphology.

5.2.5 Trim

The image is trimmed to remove the non-imaging pixels, such as the overscan and any pre-scan pixels, along with those pixels near the edges that have been compromised due to OT operation. The definition of the imaging area of the detector is determined from the camera configuration data or from the metadata associated with the image, with the choice a user-configurable option.

The input science and mask frames are additionally trimmed by the extent of the OT convolution kernel in each direction ($+x$, $-x$, $+y$, $-y$). Within the PSLib image handling functions, the trim function is a virtual operation which simply marks the boundaries of the trimmed image but does not remove the corresponding memory. This allows the later corrections to work with untrimmed correction images and still apply the correct pixels. At the end of Phase 2, the only the trimmed portions of the output images are written out to disk.

5.2.6 Non-Linearity Correction

If required, the science image (after bias correction) must be corrected for the effects of non-linearity through a provided polynomial fit to the pixel data values or a numeric lookup table as a function of pixel flux. The choice to apply the correction must be set by the analysis recipe.

5.2.7 Flat field Correction

The science image (after bias correction and non-linearity correction) must be corrected for sensitivity variations as a function of position, dividing by a flat-field image. The mask is also modified for zero-valued pixels in the flat-field image.

5.2.8 Sky & Fringe subtraction

After the science image has been flat-fielded, the flux contribution of the sky (from both continuum emission and the line emission that causes fringing) must be subtracted from the image. The subtraction needs to remove background (technically, foreground) variations which are not celestial but generated in the atmosphere or by more localized scattering. This background should include the contribution from the zodiacal light. This background subtraction does not address the artifacts generated by bright stars: bleeding columns, ghosts, or other localized reflection effects. This process also produces a super-binned image of the background map which may be used as a debugging diagnostic.

This analysis step temporarily masks objects on the image using the astrometric solution from the boresight and fits for the sky background, consisting of a polynomial to model the continuum, and the fringe frame(s) to model the fringes from sky emission lines. If the concentration of objects in the image is too high to reliably fit the sky background, the background solution from an exposure close in time and airmass to the current object image is used. The output is the sky-subtracted object image.

5.2.9 Detect and Measure objects

After the image have been processed by the preceding steps, the Phase 2 analysis performs a basic object detection analysis. Objects on the flat-fielded object image are found, and general parameters are measured. Object detection is performed at several stages by the IPP, with different object parameters measured in each case. Table 5 gives a list of the different detection stages and the object parameters measured for those stages. For the Phase 2 analysis, the object parameters are: the object centroid and the position covariance matrix, the instrumental PSF magnitude and error, local background level and error, a measurement of the star-galaxy separation, and a measurement of the object shape ($\sigma_x, \sigma_y, \theta$). The detection threshold must be configurable, and be a function of the average background flux or the image noise map. Minimal object classification must be performed to distinguish objects which are consistent with a single PSF, objects which are inconsistent, and objects which are saturated. The resulting collection of detected objects are saved in the AP Database along with the relevant image metadata (i.e., filter, exposure time, etc). In addition, this process constructs a model of the point-spread-function (PSF) as a function of position in the image. This PSF model is saved as part of the image metadata.

5.2.10 Identify CRs by morphology

Charged particles in the detector frequently cause features which do not have the morphology of astronomical objects. In a well-sampled image, these may be easily identified by the sharpness of the image. In a near critically-sampled image, these ‘cosmic rays’ may be indistinguishable from stellar objects. This analysis step makes morphological identification of cosmic rays if the imaging data is sufficiently well sampled. The identified cosmic rays are masked with a configurable growth factor so that additional pixels beyond the detected pixels in the feature are also masked.

5.2.11 Perform Astrometry

The detected objects are matched with known astrometric reference objects, using reference object coordinates which have been adjusted for proper motion. The matches are then used to improve the astrometric parameters for the individual OTAs. The OTA astrometric parameters which are determined may include terms up to 3rd order in position, though the terms which are actually fitted are user-configurable. The Cell astrometric parameters are not allowed to vary at this stage. The fit must be robust, rejecting outlier matches (either stars with poorly determined proper motion or spurious matches). The resulting astrometric solution is consistent across the OTA field to within 1.0 arcsec.

5.2.12 Perform Photometry

If possible (if a local photometry reference exists), the Phase 2 analysis determines a photometry zero point for each image. To do this, it extracts the appropriate reference objects (from the AP Database) and matches the stars between the two samples. The zero-point is derived on the basis of a static atmospheric absorption model (eg, linear airmass slope).

5.2.13 Bright object postage stamps

The IPP must have the capability of extracting regions surrounding a specified subset of objects from the flattened images. These postage stamp images must be saved for additional use by client science pipelines. The identification of these objects must be on the basis of a set of rules applied to the object's magnitude and position. The postage stamps are not restricted in shape to simple rectangles, but may represent more complex regions. They are written to the Image Server. The outputs are these postage stamps and pixel masks, which are sent to the IPP Pixel Server.

5.3 Phase 3 : exposure analysis

The Phase 3 system operates on the combined Phase 2 results from an FPA to determine improved solutions for the image calibrations and to provide the parameters needed by Phase 4. The Phase 3 output is saved by the Metadata Database, and consists largely of improved values of the calibrations already determined by Phase 2. The analysis performed by this pipeline consists of:

- improved astrometric solution based on comparison between objects in the images and the astrometric reference.
- improved background model based on the full telescope field, or fields.
- photometric solution based on comparison to photometric standards
- FPA-wide PSF analysis

In the Phase 2 analysis, the astrometric solutions were determined independently for each chip. These solutions are limited by the assumption of a static distortion and by the accuracy of the astrometric reference. In the phase 3 analysis, the astrometric solutions of the complete FPA images are improved by combining the astrometry for all chips. The astrometry model consists of a projection of the celestial coordinates to the telescope boresite center, followed by a rotation to the average rotation of the FPA and adjustment for the central plate scale. The free parameters in this stage are the boresite coordinates (R_o, D_o), the field rotation (θ_o) and the plate scale (ρ_o), and are fitted in Phase 1. These tangent plane coordinates are then distorted by the optical distortion model, consisting of $N^{\rm th}$ order polynomials in two dimensions. Finally, the focal plane coordinates are mapped to the individual chip coordinates, including the chip position and rotation, as well as possible higher order terms representing warping of the individual detectors. A first pass at the chip positions is calculated in Phase 2, while the complete set of parameters is fitted as a whole during Phase 3. The fitting process is determined in a robust and stable way by fitting the gradient of the distortion as a function of field position, removing the degeneracy of the distortion coefficients on the chip position parameters.

In the Phase 2 analysis, the background is determined based only on the available sky in a single chip. However, the background structures are normally correlated on the scale of the FPA, so an improved background solution can be determined by combining the information from many chips. A high-order polynomial model of the background may be used and subtracted from all chips.

The Phase 3 photometric improvement is made by comparing the photometered objects from Phase 2 with the corresponding objects in a local reference catalog. This analysis may only be performed if a local reference is available. Note that improved relative photometry calculations may be performed in the absence of a reference catalog on the basis of image overlaps in the AP Database *after* the detections have been added to the Database. Such a relative photometry analysis is outside the scope of Phase 3 and will likely be performed as an independent analysis process. Given the presence of a local photometry reference, the zero point variations across the field may be measured, and possibly modeled. If the zero-point variations are excessive, then the image is marked as non-photometric by the analysis.

In the Phase 4 analysis, the N FPA images are optimally combined to create a single image of the sky with bad-pixel and cosmic-ray rejection. This combination requires the calculation of a set of PSF kernels to convert each of the input images to a single, common PSF. These PSF kernels are determined from the per-chip PSFs measured in Phase 2.

5.4 Phase 4 : image combination

5.4.1 Overview

Phase 4 is the image combination stage, in which multiple images of the same portion of the sky are merged and confronted with the static sky image. Phase 4 operates on the smallest data unit of the static sky, the sky cell, along with the associated pixels from a collection of images which have been processed through phases 1–3. The size and exact representation of a static sky cell are yet to be determined. The working concept is that the static sky cells contain roughly the same number of pixels as an OTA (4k x 4k) and represent a portion of a local tangent plane projection. In order to meet the image degradation requirements, the pixel scale of the static sky is planned to be $0.2''$, somewhat smaller than the $0.3''$ raw image pixel scale.

For each sky cell, the corresponding pixels are extracted from the exposures being processed and mapped to the projection of the sky cell. The pixels from the multiple input processed images are combined into a single, cleaned image. Outlier pixels may be optionally rejected at this stage (optionally, since moving objects will be rejected in images obtained over a wide range of times). This image is then confronted with the static sky cell data to produce a difference image. Residual objects in the difference image above a threshold are detected and excised from the original cleaned image. The remaining pixels are added to the existing static sky image. Object detection must be performed on the difference ($P4\Delta$) and cleaned ($P4\Sigma$) images.

5.4.2 Image Warping and Matching

The analysis first maps the detector images to the sky cell using the specified linear transformations, then combines the images with strong rejection criteria and uses the combined sky cell image to identify artifacts in the original detector images. It is desirable that the artifacts are masked in the detector plane (i.e. before mapping to the sky cell) so that they are not smeared out by the mapping; alternatively, the CR mask needs to be grown by an additional pixel. The mapped and masked detector images are then optimally combined using the specified weighting. Both sets of combinations use the photometric calibration for the images to set the relative scales of the input images. The limiting magnitude for the combined sky cell image should also be estimated.

5.4.3 Static Sky Subtraction

The corresponding static sky image is subtracted from the combined image stack. In this step, it is necessary to match the image kernel between the input image and the static sky image. This will be done by solving for a best-fit image kernel which minimizes the difference image using a technique equivalent to the Allard-Lupton method. One modification for the IPP is to represent the kernel as a combination of independent pixels rather than represent the components of the image difference kernel as a combination of Gaussians. This method also automatically determines a photometric match between the static sky image and the input science image.

5.4.4 Object Detection and Measurement

Objects in the difference image are detected and a specific set of object parameters are measured from these detections. Table 5 gives a list of the different detection stages and the object parameters measured for those stages. For the Phase 4 difference image (P4Δ), the measured object parameters consist of: the object centroid and the position covariance matrix, the instrumental PSF magnitude and error, local background level and error, a measurement of the star-galaxy separation, and a measurement of the object shape ($\sigma_x, \sigma_y, \theta$). The detection threshold must be configurable, and be a function of the average background flux or the image noise map. Detections must be performed for both positive and negative (static sky only) sources. Minimal object classification must be performed to distinguish objects which are consistent with a single PSF, objects which are inconsistent, and objects which are saturated. The resulting collection of detected objects are saved along with the relevant image metadata (i.e., filter, exposure time, etc).

Objects in the cleaned, summed image are detected and a specific set of object parameters are measured from these detections. Table 5 gives a list of the different detection stages and the object parameters measured for those stages. For the Phase 4 summed image (P4Σ), the measured object parameters consist of: the object centroid and the position covariance matrix, the instrumental PSF magnitude and error, local background level and error, a measurement of the star-galaxy separation, a measurement of the object shape ($\sigma_x, \sigma_y, \theta$), the Petrosian radius, magnitude, axis ratio, and angle; and the Sérsic radius, magnitude, axis ratio, angle, and parameter ν. The detection threshold must be configurable, and be a function of the average background flux or the image noise map. Minimal object classification must be performed to distinguish objects which are consistent with a single PSF, objects which are inconsistent, and objects which are saturated. The resulting collection of detected objects are saved along with the relevant image metadata (i.e., filter, exposure time, etc). In this measurement, objects at known positions will also be measured even if they have not been detected.

Objects which are detected in both of the Phase 4Σ and Phase 4Δ images are saved to the AP Database, along with the relevant image metadata (i.e., filter, exposure time, etc). In the process of adding these objects to the database, the transients which are correlated with previous detections of an object (and those which are not) will automatically be determined. A subset of these transient objects are sent, along with their associated metadata, to the MOPS and other preferred science client pipelines.

5.4.5 Static Sky Update

It is essential that the static sky image (which may have been painstakingly accumulated over many months) not be corrupted by adding in transient sources, or data that is of suspect quality (due, e.g., to an error upstream in the processing).

Object analysis of the static sky images is *not* a part of the Phase 4 analysis. This processing is envisioned to take place relatively infrequently (perhaps weekly or even monthly) and is scheduled as a separate analysis task, probably run during the day at a time when the computing infrastructure is not under significant load.

6 System Design : Calibration Image Processing

The Calibration Analysis Stages construct calibrations from the relevant input data. Some of these combine multiple raw input images together, after processing, to create a high-quality high-signal master calibration image. Some of the calibrations are used to correct other calibrations. Each of the calibration stages must also provide the tools to test the quality of the input data against existing master calibration data and to test the consistency of multiple measurements of the calibration.

The Calibration analysis stages may be performed on whatever timescales are appropriate and necessary to maintain the quality and relevance of the calibration images. The specific calibration data which must be constructed in the calibration analysis stages is listed below.

The IPP must generate basic calibration images using the raw bias, dark, and flat-field (dome or twilight) images obtained by the telescope as the input. The analysis of these images requires relatively simple stacking of the input set of images. Outlier rejection, both of complete input images as well as pixels within the input stack, must be performed. In addition, each type of image requires an appropriate normalization which may depend on the data levels in other detectors in the input set. Each of these calibration stages must be able to determine from the input stack if the relevant calibration image needs to be updated and perform an initial test to see which input images are consistent and valid.

6.1 Bias Images

Bias images may be needed to correct for structure in the bias. The IPP must have the capability of constructing a master bias image from a stack of raw bias frames. The input bias images, representing offsets from the overscan level, are processed by subtracting the overscan, including 1D structure if needed.

The master bias frame construction uses outlier image and outlier pixel rejection to construct a single high-quality bias frame. The statistic used to determine pixel values from the input stack can be set by the user to be one of the following: the sample mean, median, and mode, robust mean, median, and mode, and the clipped mean and median. Testing of the input images consists of constructing residual images, in which the master bias is applied to the input images. These images may be included or excluded from an additional iteration of the stack on the basis of their pixel-to-pixel statistics.

6.2 Dark Images

Dark images may be needed to correct for structure in the dark current. The IPP must have the capability of constructing a master dark image from a stack of raw dark frames. The input dark images are first corrected for the bias using whatever method is appropriate for the science images. Master dark frames depend on their exposure time. As such, the input dark frames must have a limited range of exposure times, and the output dark frame includes the exposure time as part of its associated metadata.

The master dark frame construction uses outlier image and outlier pixel rejection to construct a single high-quality dark frame. The statistic used to determine pixel values from the input stack can be set by the user to be one of the following: the sample mean, median, and mode, robust mean, median, and mode, and the clipped mean and median. Testing of the input images consists of constructing residual images, in which the master dark image is applied to the input images. These images may be included or excluded from an additional iteration of the stack on the basis of their pixel-to-pixel statistics. A collection of master dark frames with a range of exposure times are used to determine the scaling of the dark frame as a function of exposure time.

6.3 On-Off Dark Images for Light Leaks

A type of image which may be necessary for calibrations will be pairs of images taken at night with the shutter closed with and without the dome shutter closed. Such a pair of images can be used to determine any light-leak in the camera which may contribute additional flux across the mosaic.

6.4 Flat-Field Images

Master flat-field images must be constructed from a collection of input flat-field images. The input flat-field images may be obtained from any of the standard sources: the dome, the twilight sky, and the night-time sky. The choice of flat-field input image must be determined experimentally from observations during the commissioning phase of the telescope. The IPP flat-field construction system must be capable of handling any of these sources.

An appropriate set of input images is selected on the basis of their flux levels, time of observations, and the observing conditions. The input flat-field images are processed (bias and dark corrected if needed) and the resulting images are stacked. The master flat-field construction uses image and pixel outlier rejection to construct a single high-quality master flat-field frame. The statistic used to determine pixel values from the input stack can be set by the user to be one of the following: the sample mean, median, and mode, robust mean, median, and mode, and the clipped mean and median. Testing of the input images consists of constructing residual images, in which the master flat-field image is applied to the input images. These images may be included or excluded from an additional iteration of the stack on the basis of their pixel-to-pixel statistics.

6.5 Mask Images

Preliminary bad-pixel mask images are generated on the basis of comparison between raw flat-field images and a cleaned, stacked master. The mask creation system accepts a collection of flat-field images and identifies pixels which are consistently poorly flattened. Pixels which are under-responsive are also identified as pixels to be masked.

6.6 Sky & Fringe Frames

Fringe-correction frames must be generated to remove the fringe pattern caused by thin-film interference in the top layers of CCDs, particularly in the redder passbands. Fringe correction frames may be constructed on the basis of observations of the night-sky in the appropriate filters or on the basis of dome fringe lamp observations. The choice of the appropriate source will be determined experimentally on the basis of data obtained during the commissioning phase. The IPP must be capable of handing either source. The images are first flattened to remove the pixel-to-pixel sensitivity variations of the detector. The combination of multiple input fringe frames may not be simply stacked since the amplitude of the fringe pattern varies independently of other variations in the image. The amplitude of the fringe pattern in the input frames is measured and the images scaled to normalize the fringe amplitude to a consistent range (-1 to +1) for all input images before they are combined with one of the standard combination statistics (mean, median, mode, etc). The quality of the input frames is tested by flattening the input image and applying the master fringe-frame. The resulting residual image statistics are used to select or exclude specific input images.

6.7 Shutter Correction Map

Shutter correction map images may be generated based on the timing measurements of the shutter itself, or on the basis of dome-flat images of decreasing exposure times down to the shortest available exposures.

6.8 Low-k Sky Models

Large-scale background structure in images which is not caused by thin-film interference must also be detected and corrected. Models of this background structure may be a necessary input to the correction procedure. The IPP must have the capability of generating image models of the large-scale structure patterns observed with the telescope

6.9 Flat-Field Correction Frame

Flat-field images, whether constructed from the dome, twilight, or night-sky images, do not perfectly correct the detector response in a consistent fashion across the full field of the camera. The IPP must have the capability of generating flat-field photometric correction frames on the basis of the measured photometry of objects which are moved to a variety of locations on the detector in a sequence of images. The flat-field correction frames analysis stage makes use of targeted observations following a specified dither pattern, and extracts the photometered objects from the AP Database to determine the necessary photometric corrections. The resulting image is applied to the master flat-field image. Testing of the correction is performed by applying the correction to the basic master flat-field image, applying that flat-field image to the dithered photometry observations, and performing the object detections. Comparison of the photometry of individual stars at different locations on the mosaic will demonstrate the consistency of the flat-field image.

6.10 Non-Linearity Correction

The IPP must have the capability of constructing a correction for non-linearity in the detectors. These frames are constructed from exposures of a uniform source with a range of exposure times. The non-linearity correction frames provide polynomial correction coefficients or a lookup table describing the correction. There is likely to be a single non-linear correction for each OTA detector, or potentially for each Cell. The IPP must handle these two cases.

7 System Design : Miscellaneous Tasks

This section discusses additional operations which are performed by the IPP but which do not fall under the analysis of the science images or the creation of the calibration images.

7.1 Retrieval

The retrieval stage simply retrieves images from an external source (ordinarily OTIS at the Summit, but it could conceivably be some other external source) and store it in the Image Server.

7.2 Static Sky Analysis

The IPP is responsible for performing object detection and analysis on the static sky. This analysis is performed continuously (every day or week) on those portions of the sky within 15° of the sun. In this analysis, the object measurement is much more detailed than those performed in the real-time analysis. The currently envisioned parameters to be measured for every object are listed in Table 5. The parameters include the object centroid and the position covariance matrix, the instrumental PSF magnitude and error, local background level and error, a measurement of the star-galaxy separation, a measurement of the object shape ($\sigma_x, \sigma_y, \theta$), the Petrosian radius, magnitude, axis ratio, and angle; the Sérsic radius, magnitude, axis ratio, angle, and parameter ν, and a collection of annular aperture flux measurements, all of which are also measured for the P4Σ images. In addition, the galaxy-shape parameters $Gamma_1 \& \Gamma_2$, along with the complete 'polarization' terms are measured, as well as a collection of annular aperture flux and variance measurements. Another unique feature of the static sky analysis is that the object detection may be performed simultaneously on all filters, in which case the locations and other parameters may be more strongly constrained by simultaneously fitting between all bands. The analysis to be performed may be substantially more complex than the real-time analysis because of the relatively low data rate. Instead of needing to process thousands of images per night (~ 350Mpix per second), it is only necessary to process the complete sky in a year, or an average rate of $\sim$2 Mpix per second, or < 1% of the object analysis in the other analysis stages.

7.3 AstroRef: Astrometric Reference Catalog creation

needs to be fleshed out substantially (TBD)

This processing stage shall use many observations over a given time period to fit a consistent global astrometric solution, resulting in a high quality and internally-consistent astrometric catalog that may be published.

7.4 PhotoRef: Photometric Reference Catalog creation

needs to be fleshed out substantially (TBD)

This processing stage shall use many observations over a given time period to fit a consistent global photometric solution, resulting in a high quality and internally-consistent photometric catalog that may be published.

8 Software Hierarchy

In order to facilitate testing and development, and to encourage flexibility, the IPP will be built in a layered fashion. The lowest level functions will be written in C and collected together into a Pan-STARRS library. These library functions will be used to write more complex modules. The modules will be written in C but will make use of SWIG to make their functionality available within other languages. In particular, the modules can be tied together with a program in C or through the use of a high-level language such as Perl, Python, or Tcl employing the SWIG interfaces. For the high-level functions in the operational system, the IPP will make use of Perl as the scripting language to provide the required flow-control to tie the modules together.

This approach satisfies the requirement that complicated low-level analysis steps run fast, while preserving flexibility for coding the high-level wrappers for which the speed requirements are not so stringent.

8.1 External Libraries

Pan-STARRS will employ several external libraries to save duplicating functionality that is already available. These external libraries will be wrapped by the Pan-STARRS Library, insulating the project from the implementation details of the external libraries. Examples of the external libraries are FFTW and SLALib.

8.2 Pan-STARRS Library

The Pan-STARRS Library will consist of C structures describing the basic data types needed by the IPP and C functions which perform the basic data manipulation operations. Note that a subset of the library functions will be provided with SWIG interfaces as well to allow for their use in the creation of the processing stages. Examples of the Pan-STARRS

Library are Fourier transforms and transforming between pixel and celestial coordinates. The details of the Pan-STARRS Library are specified in the document Pan-STARRS IPP PSLib Supplementary Design Requirements Specification (PSDC-430-007), which also addresses coding requirements detailed in the IPP PS-1 SRS (PSDC-430-005), Section 3.3.

8.3 IPP Modules

The IPP analysis stages are broken down into modules which represent specific functional operations. The modules will be written in C using the Pan-STARRS Library functions and will be grouped into a Pan-STARRS Module Library. The modules will be provided with SWIG interfaces to all public APIs for their use in processing stages. Examples of modules are overscan subtraction and image combination. Some modules (e.g. find objects on an image) will be used by multiple stages. The details of the Pan-STARRS Modules are specified in the document Pan-STARRS IPP Modules Supplementary Design Requirements Specification (PSDC-430-012), which also addresses coding requirements detailed in the IPP PS-1 SRS (PSDC-430-005), Section 3.3.

8.4 IPP Stages

The major IPP processing tasks are organized into stages, which consist of multiple modules. Each stage represents a collection of complex operations performed on a single data entity. Each stage therefore represents the maximum amount of effort which can be performed in serial without interaction between parallel threads. The stages will be written in Perl, linking the modules together. Examples of stages are Phase 2 (detrend images) and Phase 4 (combine images from multiple telescopes and search for transients).

9 Interfaces

9.1 Internal Interfaces

Internal interfaces consist of interactions between the analysis scripts and the IPP Metadata Database, Image Server or AP Database. There are also interfaces between the IPP Scheduler, Controller, and the Metadata Database.

The science and calibration image processing pipelines make requests for images from the Image Server, metadata from the Metadata Database, and push their results back onto the Image Server and Metadata Database. The Scheduler specifies analysis tasks and sends them to the Controller, and determines the next action based on the contents of the Metadata Database. The various subsystems specify the API for client / server interactions, and are discussed in their individual section. Commands will be sent using either text-based commands, SOAP or an equivalent protocol. The format of the exchanged data may be in one of several forms discussed below.

FITS Images will be used to transport images between the components of the IPP. Non-standard FITS images representing triangular collections of pixels may be used to store the static sky images.

FITS Tables will be used to store and transport tabular data, especially large queries from database subsystems. The Auto-coding technique discussed in Appendix E is used to define many different table interactions.

XML files will be used to store and transport data which is not well-suited to the rectangular form of FITS Tables. Hierarchical data concepts and variable-length structures fall in this class. Examples include mosaic astrometry description information and configuration information.

SQL queries and C wrappers of SQL queries will be used as the direct interface to the databases.

Within IPP and Pan-STARRS in general, process-to-process communication will be defined through auto-coded APIs which support a limited and validated communication protocol. The APIs will be coded based on a table which defines the allowed command set and the grammar to be used. This mechanism will allow a single code block to define inter-process communication methods for many Pan-STARRS subsystems, including, within the IPP, the Scheduler-Controller communications.

9.2 External Interfaces

This subsection describes the interfaces between the IPP and other Pan-STARRS systems and the external clients. The interfaces are illustrated in Figure 1.

9.2.1 OTIS

The IPP Scheduler may query OTIS for a list of new images and metadata. The locations of those images in the Summit Pixel Server is sent back as a table, and all metadata may be sent to the IPP as a collection of FITS Tables. The IPP also may send quality assessment information for each FPA and major frame by writing out FITS tables and notifying OTIS of the presence of the new tables.

9.2.2 Camera

Images are pulled from the Summit Pixel Server, part of the Camera team's purview. The locations of the files are sent by OTIS. IPP may grab these via `http` commands or via `NFS` or another network file exchange protocol. The IPP notifies OTIS (and Camera) when each image has been received.

9.2.3 PSPS

Data will be sent to PSPS from the IPP as part of a daily or weekly analysis process on the Static Sky. The data will be pushed from the IPP to PSPS when they are available. The data to be transfered include:

- Static Sky images - to be transferred as FITS images or FITS triangular image regions.
- Postage Stamps - to be transferred as FITS images.
- Metadata tables - to be transferred as FITS tables
- Detections & Object associations - to be transferred as FITS tables.

9.2.4 MOPS

Data will be sent to MOPS from the IPP as part of the Phase 4 analysis. The data will be pushed from the IPP to MOPS when they are available. The data to be transfered include:

- Image Metadata tables - to be transferred as FITS tables
- Orphaned Detections - to be transferred as FITS tables

9.2.5 Other Preferred Client Science Pipelines

These cannot be completely defined until the Clients are defined and their requirements are specified. The expectation is that the data products will be the same as for the MOPS. The data will be pushed from the IPP to the Client Science Pipeline when they are available.

10 Computer Hardware

10.1 PS-1 Cluster Design

The PS-1 IPP computer system is designed as a cluster of ’fat bricks’: computers with both processing power and large amounts of local disk storage. These computers are large rack-mount boxes with space for 10s of disks (24 and 36 disk cases are available) and a motherboard with two CPUs and two Gig-E ethernet ports. One set of machines is specified for storage and processing of the individual OTAs up through Phase 2 (the ‘OTA nodes’), another set of machines are specified for storage of the Static Sky and processing of data from Phase 3 and Phase 4 (the ‘Sky nodes’). Other machines will be necessary to support the Metadata DB and the AP DB.

The IPP PS-1 SRS (PSDC-430-005) specifies the processing throughput requirements for the IPP. Benchmark tests of the IPP processing algorithms have been used to drive the design needed to achieve the throughput requirements. The details of this study are presented in the IPP Computational Challenge (PSDC-400-006), summarized here. The analysis measures the processing time (excluding I/O) for both Phase 2 and Phase 4 on an Intel Pentium 4 processor, and expresses the processing time in GHz-seconds, under the assumption that a machine with the same architecture and twice the processor speed will perform the same analysis in half the time. This is probably a valid assumption within a limited range on hardware using the same architecture. Independent tests show that 32-bit Pentium processors perform somewhat slower (up to a factor of 2) than equivalently rated 64 bit Opteron processors. This discrepancy makes the measured numbers somewhat conservative, and compensates for the simplified analysis performed. The benchmarks show that the Phase 2 analysis takes 12000 GHz-seconds for a complete major frame (4 FPAs) while the Phase 4 analysis takes 7800 GHz-seconds for the same major frame.

The total data I/O required for each processing node, both locally to disk and across the network to other machines, has also been measured. These numbers in turn depend on whether the data is optimally stored on the OTA nodes (raw images matched to their calibration images) or if the data are randomized across the storage nodes. There are also differences in the analysis for the number of bits per pixel and the number of calibration images used in the processing. For PS-1, the ‘minimal’ data set is appropriate, resulting in a total Phase 2 I/O of 21 GBs per major frame and a total Phase 4 I/O of 36 GBs. The randomized numbers are used as a conservative estimate, under the assumption the network, not local disk access, is the dominant I/O bottleneck.

The analysis assumes each CPU (rated at 2.2 GHz) is associated with one RAID array (maximum throughput 110 MB/sec) and one network controller (maximum throughput 70 MB/s). In this case, given the CPU load and I/O throughput above, Phase 2 will require a total of 190 seconds of I/O and 5500 seconds of processing distributed across the cluster. Likewise, the Phase 4 analysis will require a total of 330 sec of I/O and 3500 seconds of processing. Given the 160 seconds available per major frame, these numbers imply a total of 63 processors are needed to keep up with the processing and I/O load.

The other major driver on the IPP PS-1 cluster is the data storage requirements. It is necessary to store the raw images from the entire AP Survey, the MOPS Verification Program (MVP) and the IPP Verification Program (IVP), and to have storage enough to represent the Static Sky by the end of the two year mission. These storage requirements as a function of time are shown in Figure 8. Based on the PS-1 Design Reference Mission (PSDC-230-001), by the end of the second year, the total storage requirements for raw images and the Static Sky will be 850 TB, along with and an additional 55 TB needed for the AP DB storage

To meet these requirements, the IPP cluster is designed to use fat bricks which will be capable of holding 24 disks each. The 5U / 24 disk rack mount computer cases are one of the highest density solutions currently available. A 4U / 36 disk box is also available and will be considered. The disk purchases will be staggered in three waves. Before PS-1 goes on the sky, the first 1/3 of the disks (600 disks total) will be purchased. Since the lead time for disks is fairly short, the purchase will be made only when other portions of Pan-STARRS are clearly on a timeline to success. After 9 months (tentatively 2006 September), the next 1/3 of the disks will purchased, and the remaining disks 9 months after that (tentatively 2007 June). Using conservative estimates of the available disk sizes at these purchase dates (400 GB, 600 GB, and 900 GB), and allocating 1 of 12 disks to the RAID and 10% of the volume to file system and binary Gigabyte overheads, the disk purchases outlined above result in a total volume after the last purchase of 950 TB. This meets the requirements with 10% spare excess. The disk volume profile is also shown in Figure 8 and shows that the disk space will be available in the time it is required.

The total number of computers to be purchased is 80. This provides the 1800 disk slots and more than enough processors to meet the processing requirements. This also leaves 5 live spare machines.

There are two details which are not included in the analysis above: compression and replication. Compression of the older raw data will reduce the volume requirements by a factor of roughly two. However, replication of the data across the network is necessary to ensure the data against catastrophic failures on a single machine. Replication doubles the total data space needed. These two factors will tend to cancel each other, and are ignored in the calculations above.

The IPP PS-1 clusters will have the following allocations of computers from this cluster:

- Phase 2 Nodes: 32
- Phase 4 Nodes: 30
- AP Database: 10
- Metadata Database: 1
- Image Server Database: 1
- Controller / Scheduler: 1

This distribution meets the projections for computational power for each of these data systems, and leaves 5 computers as live spares for redundancy.

10.2 PS-1 Cluster Expected Reliability

With 80 computers and 1920 disks, component failures are inevitable. The cluster design and management must be chosen to minimize their impact on operations and data integrity.

There are several factors which reduce the cluster's exposure to hardware failures. First, the use of RAID controllers and RAID-5 striping of the data will protect the data on a single RAID set against the failure of a single disk in the array. Second, duplication of data across the cluster will protect against catastrophic failures of the array (loss of two disks, loss of the array controller card). Finally, the flexibility of the distributed computing plan minimizes the impact the loss of individual machines has on operations by making changes in the data and processing assignments on the cluster a trivial matter.

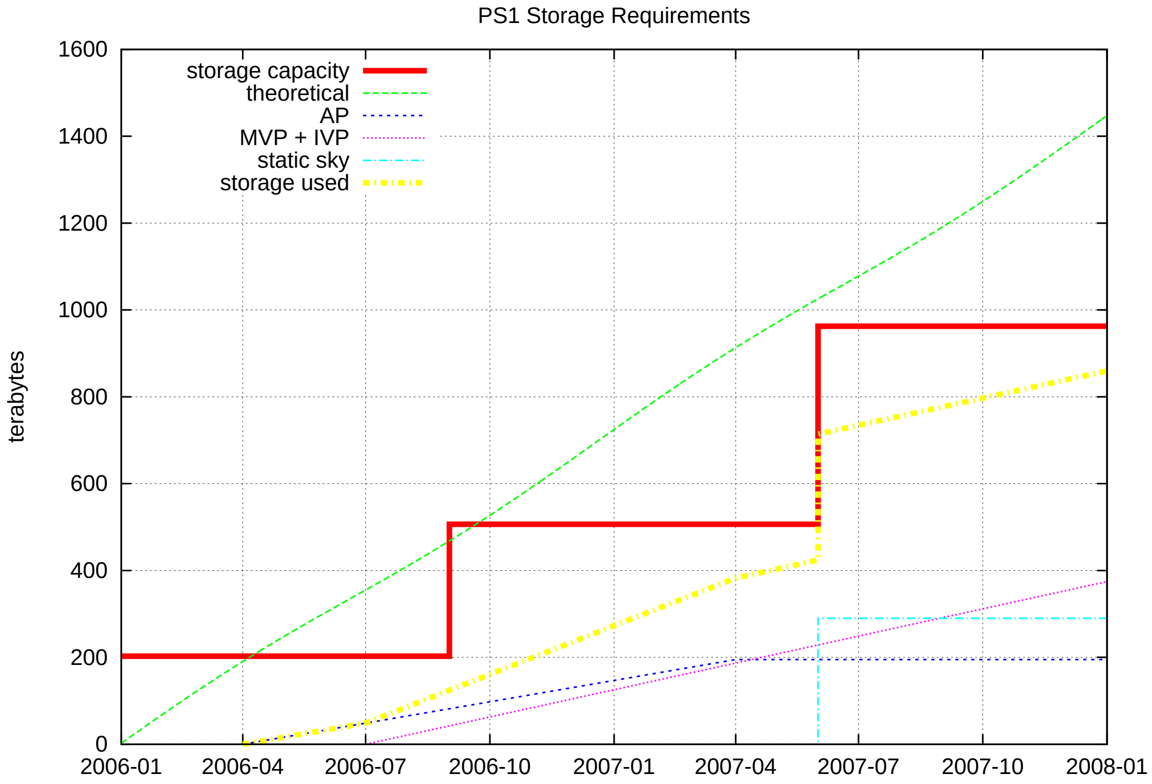

Figure 8: Storage Profile

The components which are most likely to fail in the experience of our team are, in order: hard drives, RAID controllers, ram, power supplies, and other components. The hard drive and RAID controller failure rates are by far the dominant concerns as they potentially affects the data integrity.

Most sources (REFS: UCSD article, Samsung White Paper) currently imply hard disk failure rates (MTBF) in the range 400,000 hours and 500,000 hours. These are used as an upper limit, with the more historically conservative value of 100,000 hours used instead. With 1920 disk, this MTBF implies a failure of one disk every 2.2 days. Since the disks are in a RAID which reports the disk failures immediately and drops the array into degraded mode, these failures will not have a huge impact on the operations, and recovery time is only 10s of minutes. This failure rate implies that the maintenance plan must include checks for hard disk failures on a daily basis, and should make use of email notification and early warning information (ie, SMART messages).

A catastrophic failure for the array would require two of the 12 disks to fail before the first failed disk is replaced. Assuming that maintenance is poor and it is possible for a disk to take 1 week to be replaced, the probability of a catastrophe is 1.8% each time the first disk fails. Combined with the disk failure rate, RAID catastrophes are expected 6 times over the 2 year operation of PS-1. These numbers can be used as a guideline for the level of support needed to avoid these RAID failures. Note that these 6 failures should not cause loss of data since the data is duplicated across the cluster, but they require over 1 day for recovery (as the entire array must be replicated across the network).

A detailed IPP computer cluster commissioning and maintenance plan is specified in the document ‘Pan-STARRS PS-1 IPP Cluster Support’ (PSDC-430-014).

Table 3: Metadata Database Tables

Table Name	**Description**
Weather	Details on the weather, including internal temperatures.
SkyProbe Transparency	Analysis of SkyProbe B & V data.
SkyProbe Absorption	Analysis of SkyProbe A data.
SkyProbe Emission	Analysis of SkyProbe E data.
DIMM	Summary of DIMM data analysis.
NIR	Summary statistics from NIR camera.
Dome Status	The time history of the dome status.
Telescope Status	The time history of the telescope status.
Raw FPAs	Information about the raw FPA exposures.
Pending Science Chips	Science images to be processed and status.
Processed Science Chips	Science images which have been migrated to the processed state.
Observation Group	Details about a group of associated observations.
Observation Frame	Major frame information.
Science Processing stats	Details on processed cells.
Chip / Sky overlaps	List of overlaps between sky cells and detectors.
Processed Sky-Cell stats	Details of the sky cell processing.

Table 4: AP Database Tables

Table Name	**Description**
Images	The images that have objects in the DB.
Image Overlaps	Image regions which are touched by specific images.
Objects	The objects — average properties of multiple detections of the same object.
Average Magnitudes	Average photometry in multiple filters
Solar System Objects	Identification of solar system objects
Matched Detections	Detections of sources in an image identified with an Object.
Orphaned Detections	Detections of sources in an image not identified with an Object.
Non-detections	Non-detections of objects in an image.
Regions	spatial distribution of tables
Filters	Filters understood by the system.
Photcodes	Transformations between different photometric systems
Zero Points	History of Zero-point & Airmass terms
Distortion Models	History of Optical Distortion terms
Database Hosts	computers used to store the tables

Table 5: AP Detection Classes & Object Parameters

Object Parameter	P2	P4S	P4D	SS
PSF x,y, covar, α, δ	+	+	+	+
PSF mag, $\sigma_{\rm mag}$	+	+	+	+
star/gal sep	+	+	+	+
σ_x, σ_y, θ	+	+	+	+
local sky data	+	+	+	+
Petrosian R, M, R_{50}, R_{90}	-	+	-	+
Sérsic R, M, AB, ϕ, ν	-	+	-	+
W.L. γ_1, γ_2, pol. terms	-	-	-	+
exp. spaced aps., Poisson noise, variance	-	-	-	+

A Image Server Database Table Contents

Tables 6 - 8 list the basic contents of the Image Server database tables.

Table 6: Storage Object Table Contents

Column Name	**Datatype**	**Description**
`so_id`	integer	internal storage object identifier
`ext_id`	string	external storage object identifier (file ID)
`comment`	string	user description of object
`epoch`	date/time	last date of access

Table 7: Instance Table Contents

Column Name	**Datatype**	**Description**
`ins_id`	integer	internal instance identifier
`so_id`	integer	key to storage object table
`uri`	string	location in hardware collection
`sha1sum`	string	checksum information
`assigned_location`	boolean	is location user-specified?
`epoch`	date/time	last date of access
`atime`	date/time	last date of access

Table 8: Volume Table Contents

Column Name	**Datatype**	**Description**
`vol_id`	integer	internal volume identifier
`uri`	string	node name

B Metadata Database Table Contents

Tables 9 – 23 list the basic contents of each of the Metadata Database tables listed in Section 4.2.

Table 9: Weather Table: some sample weather points

Column Name	**Datatype**	**Description**
Time	date/time	The time the weather information was measured.
Temperature 01	float	The external temperature
Temperature 02	float	The temperature at top of the dome
Temperature 03	float	The temperature on the primary mirror
Humidity	float	The relative humidity.
Pressure	float	The (external) atmospheric pressure.

Table 10: SkyProbe Transparency Table (sample entries)

Column Name	**Datatype**	**Description**
Time	date/time	The time the SkyProbe image was taken.
Filter	string	Filter used for SkyProbe image.
Transparency	float	The derived transparency.
Number of stars	int	The number of stars used to measure the transparency.
Astrometry	coords	The astrometry used on the SkyProbe image.
Exposure time	float	The exposure time of the SkyProbe image.
Sky brightness	float	The measured sky (surface) brightness, counts / second

Table 11: Skyprobe Line Absorption Table (sample entries)

Column Name	Datatype	Description
Time	date/time	The time the LRProbe observation was taken.
Disperser ID	string	ID of the dispersing element
Atm Component 1	float	The strength of the 1st atmospheric component.
Atm Component 2	float	The strength of the 2nd atmospheric component.
Atm Component 3	float	The strength of the 3rd atmospheric component.
Disperser ID	string	ID of the dispersing element
Number of stars	int	Number of stars used to measure the absorptions.
Astrometry	coords	The astrometry used on the LRProbe image.
Exposure time	float	The exposure time of the LRProbe image.
Sky brightness	float	The measured sky (surface) brightness, in physical units.

Table 12: Skyprobe Line Emission Table (sample entries)

Column Name	Datatype	Description
Time	date/time	The time the LRProbe observation was taken.
Disperser ID	string	ID of the dispersing element
Atm Component 1	float	The strength of the 1st atmospheric component.
Atm Component 2	float	The strength of the 2nd atmospheric component.
Atm Component 3	float	The strength of the 3rd atmospheric component.
Continuum	float	The strength of the continuum emission.
Disperser ID	string	ID of the dispersing element
Exposure time	float	The exposure time of the LRProbe image.

Table 13: DIMM Measurements Table

Column Name	Datatype	Description
Time	date/time	The time the DIMM observation was taken.
σ_x	float	Raw dispersion in x.
σ_y	float	Raw dispersion in y.
FWHM	float	Dervied seeing full width at half maximum.
RA	float	The coordinates of the measured star.
DEC	float	The coordinates of the measured star.
Exposure time	float	The exposure time of the DIMM observation.
Telescope ID	string	source of the DIMM data

Table 14: Near IR Wide-field Camera Results Table

Column Name	Datatype	Description
Time	date/time	The time the NIR observation was taken.
Sky brightness	float	The sky (surface) brightness in the NIR observation.
Sky variance	float	The variance in the sky (surface) brightness.
Astrometry	coords	The astrometry used on the NIR image.
FOV X	float	field width
FOV Y	float	field height

Table 15: Dome Status Table

Column Name	Datatype	Description
Time	date/time	The time for which the dome status is valid.
Azimuth	float	The azimuth of the dome.
Open status	boolean	Whether the dome is open or not.
Lights status	boolean	Whether lights are on in the dome or not.
Track status	boolean	Whether dome is tracking telescope or not.

Table 16: Telescope Status

Column Name	Datatype	Description
Time	date/time	The time for which the telescope status is valid.
Guide status	enum	The status of the guiding.
Altitude	float	The telescope altitude.
Azimuth	float	The telescope azimuth.
RA	float	The telescope Right Ascension (ICRS $\approx$ J2000).
Dec	float	The telescope Declination (ICRS $\approx$ J2000).

Table 17: Raw FPA Images

Column Name	Datatype	Description
ID	string	FPA image ID
RA	float	Coordinates of the boresight (i.e. telescope pointing).
DEC	float	Coordinates of the boresight (i.e. telescope pointing).
Filter	string	Filter used for the exposure.
Image Type	enum	image exposure type
Exposure time	float	Exposure time for the image.
Airmass	float	Airmass at which the image was taken.
ObsFrame ID	int	Observation frame identification number, ties FPAs into major frame
ObsGroup ID	int	Observation group identification number, ties FPAs into observing group
Observer	string	The name of the observer, or the version of the telescope scheduler software.
Program	string	The observing program being executed.
Nchips readout	int	Number of detector chips read out
Camera	string	Identification of camera source
Telescope	string	Telescope used for observation
Astrometry	coords	The astrometry used for the FPA.
Chip Metadata	string	metadata resource file
Cell Metadata	string	metadata resource file

Table 18: Pending Science Chips

Column Name	Datatype	Description
FPA ID	string	FPA image ID
Chip ID	string	Chip identification number.
Proc Status	enum	Current Processing Status.

Table 19: Processed Science Chips

Column Name	Datatype	Description
FPA ID	string	FPA Image ID
Chip ID	string	Chip identification number.
Status	enum	Current Processing Status.
Residual Stats	float	quality statistics.

Table 20: Observation Group Information

Column Name	Datatype	Description
ObsGroup ID	string	Identification number for the observation group.
Number of images	string	Number of images in the observation group.
Type	string	Type of observation.
Status	string	Status of the observation group.
etc		

Table 21: Observation Frame Information

Column Name	Datatype	Description
ObsFrame ID	string	Identification number for the observation frame.
Number of images	string	Number of images in the observation group.
Type	string	Type of observation.
Status	string	Status of the observation group.
etc		

Table 22: Science Processing Stats

Column Name	Datatype	Description
Chip ID	string	The chip identification number.
State	string	The state of the processing.
ObsFrame ID	string	The major frame the chip belongs to.
ObsGroup ID	string	The observation group the chip belongs to.
P1 astrom	string	The Phase 1 astrometry results file.
P2 astrom	string	The Phase 2 astrometry results file.
P3 astrom	string	The Phase 3 astrometry results file.
N guide stars	string	Number of guide stars used for the exposure.
Astrometry stats	string	Summary statistics for astrometry (number of stars, $sigma_x$, $sigma_y$)
Astrom catalog	string	The reference catalog that was used for the astrometry.
Bias method	string	Method used to correct the bias.
Bias stats	string	Summary statistics for bias
Flat-field image	string	The flat-field image that was applied.
Kernel data		A description of the OT kernel.
Flat-field stats		Summary statistics for flat-field (sigma of sky).
Mask image	string	The mask image that was applied.
Mask method	string	The algorithm used to mask the bad pixels.
Fringe images	string	The fringe model images that were used.
Fringe stats		Summary statistics for fringes (fringe amplitude, sky sigma)
Object stats		Summary statistics for object detection (number of objects, depth, other input parameters).
Photometry data		photometry information: magnitude zero point and other corrections.
Photometry stats		Summary statistics for the photometry (number of stars, $sigma_m$)
Photom catalog	string	The reference catalog that was used for the photometry.
PSF stats		Summary statistics of the PSF.
Software ver	string	Versions of each of the modules used in the processing.

Table 23: Chip / Sky overlaps

Column Name	Datatype	Description
Chip ID	string	The identification number of the chip.
Sky Cell ID	string	The identification number of the sky cell.
State	string	Processing state of overlap

Table 24: Processed Sky-Cell stats

Column Name	Datatype	Description
Input Chips	string	Identification numbers of the chips used to produce the sky cell.
PSF adjustments	string	Adjustments to the PSF.
CR rejection stats	string	Statistics from the CR rejection (number of CRs, distribution, limiting flux).
Image comb params	string	Parameters used for the image combination.
Diff image params	string	Parameters used for the image differencing.
Average weight	string	The weight of the reference image
P4D object stats	string	Summary statistics of the object detection
P4S object stats	string	Summary statistics of the object detection
Software versions	string	Software versions of modules used in the sky cell processing.
Processing stats	string	Summary statistics of the processing (CPU time, etc).

C AP Database Table Contents

Table 25: Images

Column Name	Datatype	Description
Image ID		
time/date		
Exposure Time		
Nstars		
NX		
NY		
photcode		
Mcal		
Mcal error		
Mcal chisq		
Airmass		
Astrometry		
PSF		
flags		
Camera		

Table 26: Image Overlaps

Column Name	Datatype	Description
Image ID		
Region Table		

D Software Runtime Configuration Issues

The IPP Software requires extensive runtime configuration information. This includes default parameters for analysis to be performed, descriptions of how a particular analysis is performed, locations of data sources, and so forth. The IPP may store this information in the Metadata Database or in configuration files available to the user. Both methods are implemented in the current design. In either method, the necessary parameters are identical. This section discusses the contents of specific portions of the runtime configuration.

D.1 Camera Definition Information

Every camera which may be analysed by the IPP has differences in how the data is represented. The IPP is built with the flexibility to handle data from many different cameras, not just the Pan-STARRS Gigapix cameras. This is partly to allow testing of the analysis system on data from other telescopes, such as MegaPrime on CFHT and Suprime on Subaru, but also to allow us to adapt to changes in the design of the Gigapix cameras themselves. It also means the IPP software may be used by astronomers for other analysis projects beyond the IPP.

Most cameras provide extensive descriptive information in the FITS image headers when the images are read out. Typically, the location and orientations of the individual detectors are defined by keywords such as DATASEC and DETSEC. Other variations on these words are used for cameras which place the pixels from multiple amplifiers in the same FITS data segment. Other parameters, such as astrometric information or exposure times, are stored in headers as well. It is possible to use these header keywords to guide the analysis software, but there are two difficulties.

First, it is very common for different keywords to be used by different cameras, sometimes even the same camera may use different keywords for the same information at different times (major readout software upgrades, for example, can be accompanied by keyword revisions). In addition, within Pan-STARRS and the IPP, it is necessary to have the capability to refer to the Metadata database as the authoratative sources of some of these entries rather than the image headers. Given this circumstance, it is at least necessary to define the appropriate source for a given data concept appropriate to data from a specific camera.

The second problem arises when actually performing an analysis. In many circumstances, the software needs to know what data to expect even when an appropriate camera image is not available. This is particularly true for a camera which is composed of multiple chips and multiple amplifiers. It is a frequent circumstance than some subset of the chips or amplifiers will either be unavailable or are invalid for one reason or another. It is important for the software to have a guide for what data should be available from a perfect readout of the given camera so decisions can be made how to handle data which is not complete. This is also important to validate that a particular dataset, which appears to be from a known camera, actually corresponds to that camera and has all of the necessary information where expected.

In order to facilitate the operation of the IPP with a variety of cameras, and to allow the software the flexibility to change the camera defintion dynamically, the IPP includes a collection of software runtime configuration information which defines a given camera. This information is represented below in the form of the PSLib Metadata Config file, but may be stored in the Metadata Database or in an alternate format as appropriate.

The a single camera is represented as a Focal Plane Array (FPA), divided into Chips, divided into Cells. For a single FPA, all imaging data is stored in a FITS file or a collection of FITS files. Software needs to know where in a given file or set of files to find a particular Cell, what Cells to expect, what chips to expect, and the relationships between those entities, etc.

A single camera configuration file (or dataset) represents the description of a complete FPA. In the configuration file, any parameters which are specific to the complete FPA are placed on their own lines. These include the definition of the keywords or database locations. An incomplete example is given below.

```
NCELL       S32    NN
NCHIP       S32    NN
EXPTIME-SRC STR    HD:EXPTIME # need to specify PHU vs EXTNAME
EXPTIME-KEY STR    EXPTIME
DATE-KEY    STR    DATE-OBS
DATE-FMT    STR    YYYY/MM/DD

TYPE        CELL   FILENAME           EXTNAME  CHIP      DATASEC        BIASSEC
CELL.nn     CELL   @ROOT@CELL         AMP00    CHIP.00   CF:[0,0:0,0]  HD:BIASSEC
CELL.01     CELL   @ID/@ID@CELL.fits  AMP01    CHIP.00   DB:???
```

D.2 Analysis Recipe Information

In order to maintain flexibility in the analysis details, the IPP uses recipes to define how a particular analysis is implemented. Each major analysis script (eg, Phase 2) has its own recipe configuration information, which may be stored in the Metadata Database or in the form of the PSLib Metadata Config file. This configuration information includes all of the user configurable parameters. Many of these may specify a specific value, or they may specify lookup methods (database locations, or header locations). The specifies of each depends on the context. Below is an example recipe file for the bias subtraction portion of Phase 2, giving several alternative options for certain entries. Note that, for example, the overscan subtraction may be specified as using a particular region given in the recipe file, or on the basis of a particular header keyword.

```
# BIAS:
BIAS.IMAGE                 STR    NONE
BIAS.IMAGE      STR    FILE:bias.fits
BIAS.IMAGE      STR    DB:BEST
BIAS.IMAGE      STR    DB:CLOSE

BIAS.OVERSCAN     STR    HD:BIASSEC
BIAS.OVERSCAN     STR    CF:[0,16:0,2048]
BIAS.OVERSCAN     STR    NONE

BIAS.OVERSCAN.STATS     STR    MEDIAN
BIAS.OVERSCAN.STATS     STR    MEAN

BIAS.OVERSCAN.FIT          STR    SPLINE
BIAS.OVERSCAN.FIT.NPTS     S32    5

BIAS.OVERSCAN.FIT          STR    POLYNOMIAL
BIAS.OVERSCAN.FIT.ORDER    S32    3
BIAS.OVERSCAN.FIT.NBIN     S32    5
```

E I/O Code Autogeneration

The IPP includes a number of data collections which have multiple representations. A software tool will be used to automatically generate code to provide I/O APIs to read and write these data and to define the data structures used to carry them within a program. Within the IPP, examples of these different data entities include database tables (ie, in the Metadata Database), FITS Tables (to exchange bulk data), and XML (to exchange more complete datasets).

I/O API Autocode template (example.def):

```
Name    Example
Table   EXAMPLE
EXTNAME EXAMPLE

KEY     XVALUE

# name  format   unit      comment
XVALUE  F32      pixels    "x coordinate"
BINNING S32      fraction  "binning factor"
NAME    STR[32]  string    "description of entry"
```

Running autocode on such a file would generate an output header and C files `example.h, example.c` with the following structure and APIs:

```
typedef struct {
  psF32 XVALUE;    // x coordinate
  psS32 BINNING;   // binning factor
  char  NAME[32];  // description of entry
} Example;

psMetadata *psFITSTableInitExample ();
psExample *psFITSTableLoadExample (char *filename, int *Nrows);
bool psFITSTableSaveExample (char *filename);

psMetadata *psDatabaseTableInitExample ();
```

```
psExample *psDatabaseTableLoadExample (char *filename, int *Nrows);
bool psDatabaseTableSaveExample (char *filename);
psExample *psDatabaseTableLoadExampleRow (char *filename, psF32 XVALUE);
```

Acronyms

CAN	Controller Area Network
CCD	Charge Coupled Device
CFHT	Canada-France-Hawaii Telescope
DC	Data Collection - Database or other data storage container.
DML	Device Meta-Language
DMT	Dark Matter Telescope
FOM	Figure of Merit
FOV	Field of View
FWHM	Full-Width at Half-Maximum
GPC	Giga-Pixel Camera
GS	Guide Star
GUI	Graphical User Interface
IAU	International Astronomical Union
IOD	Initial Orbit Determination
IPP	Image Processing Pipeline
KBO	Kuiper Belt Object
LAN	Local Area Network
LSN	Local Solar Neighborhood
LSS	Large-Scale Structure
LSST	Large Synoptic Survey Telescope
MBA	Main Belt Asteroid
MDS	Medium-Deep Survey
MOID	Minimum Orbital Intersection Distance - The minimum distance between two orbits.
MOPS	Moving Object Processing System
MPC	Minor Planet Center (of the IAU)
NEO	Near Earth Object - An asteroid or comet with perihelion ¡ 1.3AU.
OBS	Observation Seqencer
ODA	OTIS Data Archive
OOF	OTIS Observe File
OOT	OTIS Observation Tool
OTA	Orthogonal Transfer Array
OTF	Optical Transfer Function
OTIS	Observatory and Telescope System
OWS	OTIS Weather Server
PHO	Potentially Hazardous Object
PSDC	Pan-STARRS Document Control
PSF	Point Spread Function
PSPS	Published Science Products System
PTS	Pan-STARRS Telescope Scheduler
Pan-STARRS	Panoramic Survey Telescope and Rapid Response System
RFI	Radio Frequency Interference
SCD	System Concept Definition
SDSS	Sloan Digital Sky Survey
SGS	Science Goals Statement
SNe	Supernovae
SRS	Software Requirements Specification
SSS	Solar System Survey
TAC	Time Allocation Committee
TBD	To Be Determined
TBR	To Be Reviewed
TCS	Telescope Control System
TLA	Three Letter Acronym

Acronyms (con't)

TLR	Top Level Requirements
TNO	Trans-Neptunian Object
TTI	Transient Time Interval
UDS	Ultra-Deep Survey
UDS	Ultra-Deep Survey
UET	Unit Exposure Time
WFS	Wavefront Sensor
WFSS	Wavefront Sensor Star
WL	Weak Lensing
-	-
RA	right ascension
DEC	Declination
GLAT	Galactic Latitude
GLON	Galactic Longitude
ELAT	Ecliptic Latitude
ELON	Ecliptic Longitude

Glossary

Subaru	National Astronomical Observatory of Japan's 8.3m telescope
cadence	
Phase 1	IPP image processing preparation stage
Phase 2	IPP image reduction stage
Phase 3	IPP exposure analysis stage
Phase 4	IPP image combination stage
Detection	Identification of a source (real or not) in an image
Observation	In MOPS, a detection that corresponds to a real Solar System object
Designation	The identifying label assigned to newly identified Solar System objects
Orbit Identification	The identification of two separately determined orbits as representing the same object.
Attribution	The identification of a detection with a known orbit.
Linkage	The identification of sets of detections that allow an orbit determination for a Solar System object
Autonomous	operates at night without human intervention for a minimum of three nights out of seven. Daytime summit calibration and maintainance carried out four consecutive days a week (Monday-Thursday).
Robotic	operates at night without human intervention for a minimum of three nights out of seven. Only four days summit maintainence per week necessary, human intervention in calibration not required.
Remote	operates without human intervention at the summit at night.
Transient Time Interval	Time interval between two successive images of the same footprint in order to distinguish between stationary and non-stationary transient detections.
Trans-Neptunian Object	An asteroid or comet that spends most of its time outside Neptune's orbit. Include classical
Sweet Spots	
Potentially Hazardous Object	An asteroid or comet with MOID¡0.05AU with Earth's orbit
Observing Efficiency	ratio of shutter open time to total time in a night excluding weather loss

Table 27: Objects

Column Name	**Datatype**	**Description**
ID		
α		
δ		
μ_α		
μ_δ		
σ_α		
σ_δ		
χ^2 position		
$N_{\rm det}$		
$N_{\rm miss}$		
flags		

Table 28: Average Magnitudes

Column Name	**Datatype**	**Description**
object ID		
$M_{\rm int}$		
$M_{\rm ext}$		
$\chi^2_{\rm mag}$		
$\sigma_{\rm mag}$		
photcode		

Table 29: Solar System Objects

Column Name	**Datatype**	**Description**
SSO ID		
$N_{\rm det}$		

Table 30: Matched Detections

Column Name	Datatype	Description
α		
δ		
σ_α		
σ_δ		
$M_{\rm inst}$		
$M_{\rm cal}$		
$\sigma_{\rm mag}$		
photcode		
type		
flags		
time/date		
airmass		
σ_x		
σ_y		
θ		
object ID		
exptime		
sky		
$\sigma_{\rm sky}$		
etc		

Table 31: Orphaned Detections

Column Name	Datatype	Description
α		
δ		
σ_α		
σ_δ		
$M_{\rm inst}$		
$M_{\rm cal}$		
$\sigma_{\rm mag}$		
photcode		
type		
flags		
time/date		
airmass		
σ_x		
σ_y		
θ		
exptime		
sky		
$\sigma_{\rm sky}$		
etc		

Table 32: Non-detections

Column Name	Datatype	Description
object ID		
$N_{\mathrm{non-det}}$		
last time/date		
last mag		
faintest time/date		
faintest mag		

Table 33: Regions

Column Name	Datatype	Description
α_0		
α_1		
δ_0		
δ_1		
Region ID		
Parent ID		
Nchildren		
Images		
Objects		
Detections		

Table 34: Filters

Column Name	Datatype	Description
Filter ID		
Filter name		
Photcode		
λ_0		
δ_λ		
ϵ		
transmission curve		
time/date		

Table 35: Photcodes

Column Name	Datatype	Description
Photcode		
Telescope		
Camera		
Detector		
Filter		
Nominal ZP		
airmass terms		
color terms		
Target		

Table 36: Zero Point History

Column Name	Datatype	Description
Photcode		
start Time/date		
end Time/date		
Zero Points		
airmass		
color		
error		
N measurements		
N stars		
photom ref set		

Table 37: Distortion History

Column Name	Datatype	Description
Camera		
Telescope		
distortion terms		
time/date		
residuals / error		
N stars		
N images		
astrom ref set		

Table 38: Database Hosts

Column Name	Datatype	Description
machine name		
machine ID		