

LDAS/LAL Boot Camp

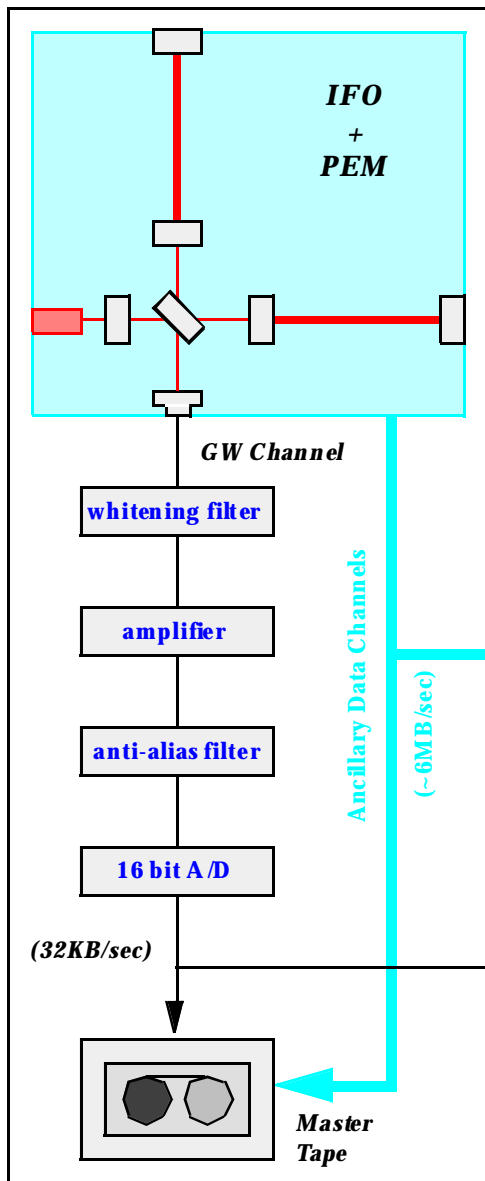
Kent Blackburn
LIGO Laboratory
Caltech
June 5th, 2001

G010228-00-E

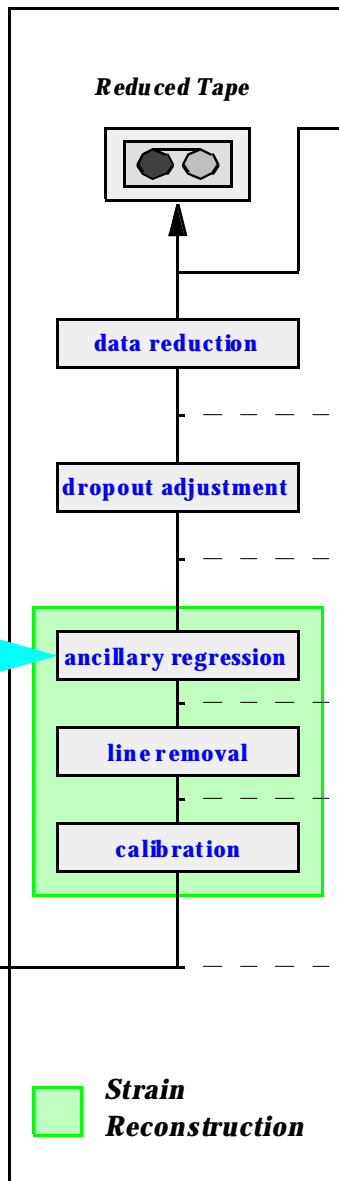
LDAS Usage Model

Historic Data Flow Model

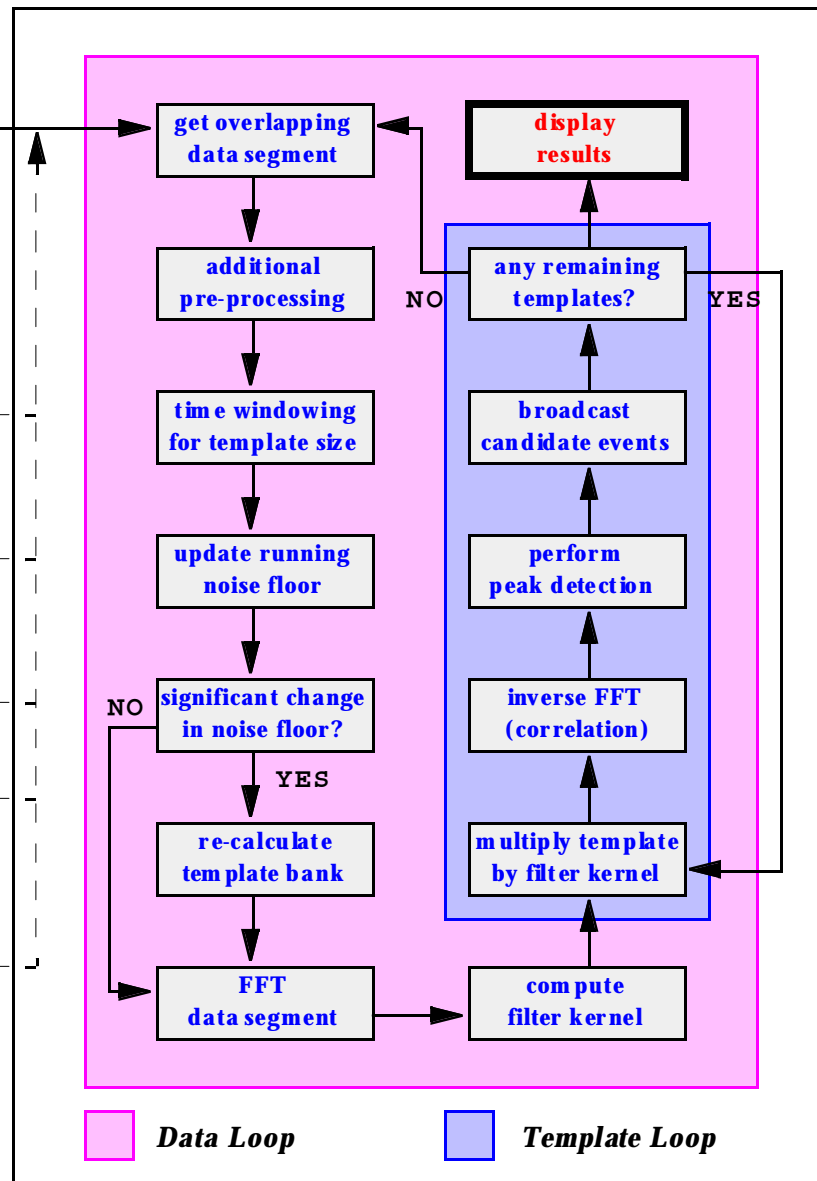
Data Acquisition



Data Processing



Binary Inspirial Data Analysis



What's Missing?

- Data Formats (Inputs, Outputs)
- Database (Metadata, Table Design)
- Data Characterization (Triggers, Vetos)
- User Interfaces
- Environment (Commands, Monitoring)

Data Formats

- **Frame Format: [LIGO-T970130-E](#)**
 - Specification developed in collaboration with VIRGO (Benoit Mours)
 - Closely resembles C structures
- **LIGO_LW: [LIGO-T990023-E](#)**
 - Specification developed in collaboration with CACR (Roy Williams)
 - Based on XML
- **ILWD: Internal LDAS Lightweight Data**
 - C++ objects which have been socket enabled
 - Has a file based representation which resembles other tagged data

LIGO Database

- Maintained/Managed by LDAS
- Original table design by Kent Blackburn; Peter Shawhan gets all the credit for making it useful (current form: [LIGO-T990101-E](#))
- Supports single/multiple interferometer searches, detector characterization, frame characterization, processes, process parameters, etc.

Detector Characterization

- Externally:
 - Trigger and Veto events generated by the DMT are added to the LIGO database using standard LDAS job command protocols and LIGO_LW.
- Internally:
 - dataConditionAPI generates PSDs and statistics on data sets through standard LDAS job commands.

User Interfaces

- LDAS Job command language (Philip).
 - Socket based (encryption coming soon).
 - Currently about a 10 individual ldas commands.
 - Uses email to notify users on completion.
 - External scripting languages interface well.
- Graphical User Interface: GUILD (Peter).
 - Powerful front-end to the LIGO database & frames.
 - Allows expert user to issue arbitrary commands.

LDAS Web Tools

www.lidas-dev.ligo.caltech.edu

LIGO Data Analysis System

Welcome to the LDAS Development Web Site

SOFTWARE
LDAS Software Index
LDAS/LSC Software Development
LDAS Problem Reporting System
User Access Tools
LDAS Bulletin Board

HARDWARE
Coming Soon

GETTING STARTED
How to Build LDAS
How to Configure LDAS
How to Test LDAS
LDAS Operator Commands
LDAS User Commands

DATA
Frame Archive
LDAS Database

LOG TOOLS
LDAS API Run Status
Electronic Logs

LDAS Development Log Files

	Current	Previous	Past
controlmonitor API	View	View	View
datacondition API	View	View	View
eventmonitor API	View	View	View
frame API	View	View	View
lightweight API	View	View	View
manager API	View	View	View
metadata API	View	View	View
mpi API	View	View	View
wrapper API	View	View	View

Shaded relief map courtesy Johns Hopkins University Applied Physics Laboratory

LIGO

[Contact us!](#)
Last modified on April 5, 2001

- Point and Click!
- Site-to-Site Access
- Running API Status
- LDAS Logs
- Access CVS
- Problem Reports
- Software Documents
- How-To ...

<http://www.lidas-sw.ligo.caltech.edu>

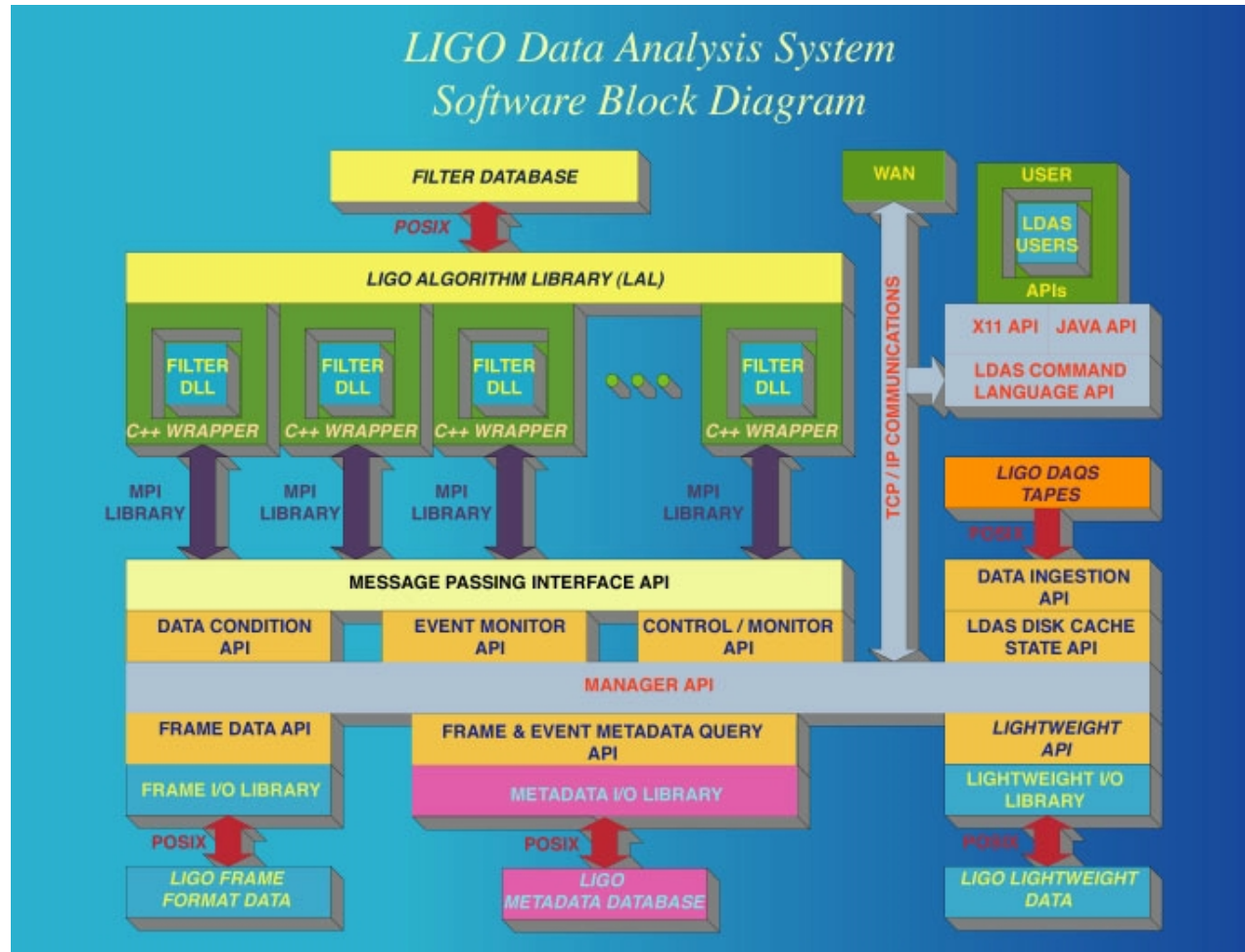
Anatomy

LDAS APIs

Top Level Comments

- **LDAS is a distributed computing environment mixing remote process control on servers with message passing on PC cluster.**
 - A TCL scripting layer is used to manage processing dynamics.
 - An underlying C++ extension to TCL provides the computational power.
- LDAS provides a framework for conducting scientific studies of LIGO data primarily through the concept of a “data-pipeline”.
 - LDAS user commands initiate a pipeline through a simple socket connection with username and password authentication.
- LDAS, uses dynamically loaded shared objects (dso) to customize the analysis code used on the LDAS PC cluster to each search.
 - Each node in a job loads search specific code at runtime.

Software Block Diagram



LIGO-T990101-E
LAL Specification

LIGO-T990097-E
wrapperAPI Requirements

www.lam.org
LAM version 5.6.1

LIGO-T990086-E
mpiAPI Requirements

LIGO-T990002-E
dataConditionAPI Requirements
FFTW version 2.3.1
CLAPACK version

LIGO-T980115-E
managerAPI Requirements

LIGO-T980117-E
frameAPI Requirements

LIGO-T970130-E
LIGO-VIRGO Frame Specification
FrameCPP Version 4

LIGO-T980119-E
metaDataAPI Requirements

dbEasy Library
ODBC Level 3

LIGO-T990101-E
LDAS Database Tables

LIGO-T990023-E
LIGO-Lightweight Format

GUILD
TCL/TK GUI

UNIX Sockets (TELNET)
User Commands (TCL)

LIGO-T010052-E
dataIngestionAPI Requirements
"In Development"

LIGO-T010051-E
diskCacheAPI Requirements
"In Development"

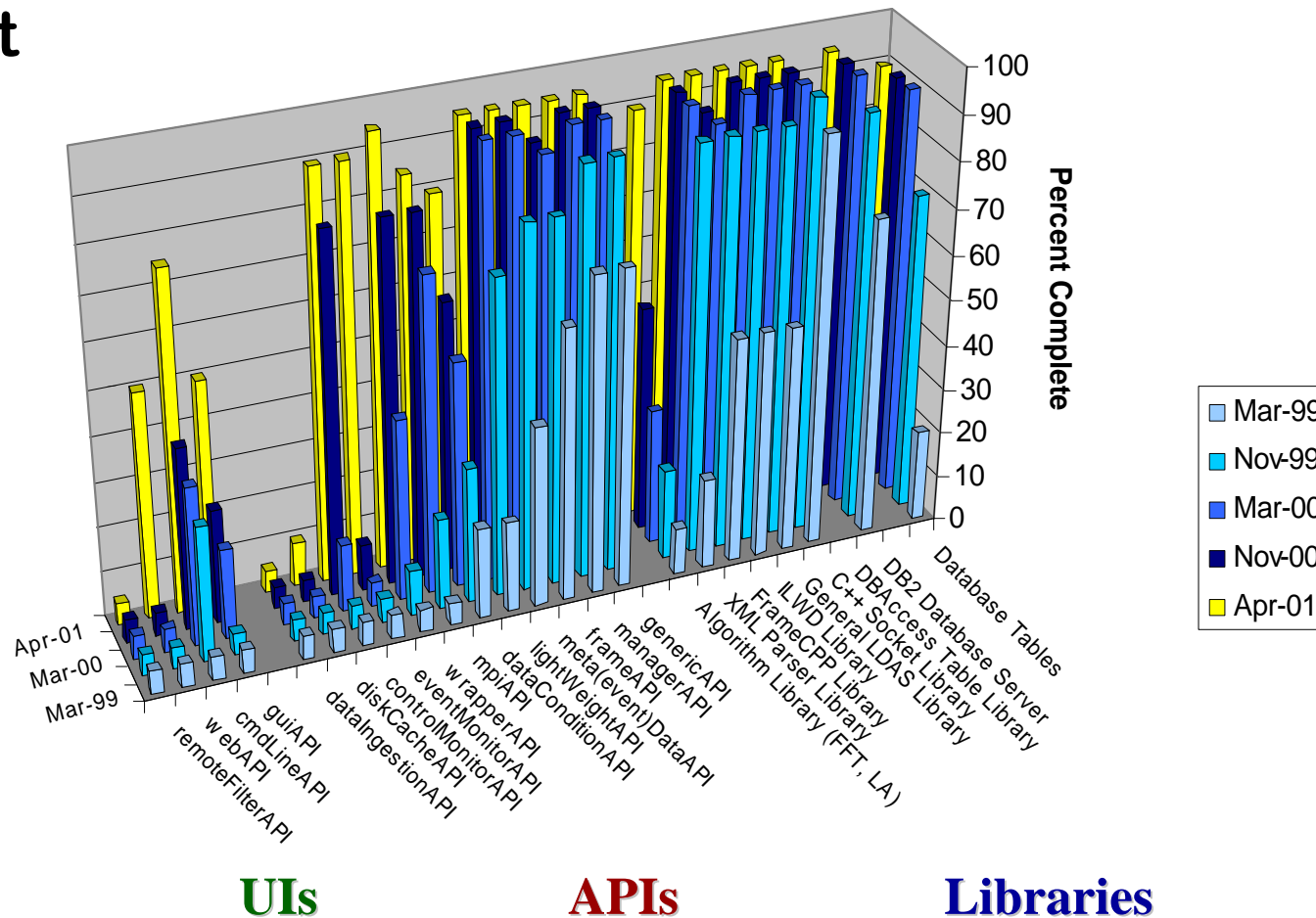
LIGO-T990037-E
lightWeightAPI Requirements

www.apache.org
C++ XML Parser Library

LDAS Component Evolution

LDAS Software Development Status:

Current

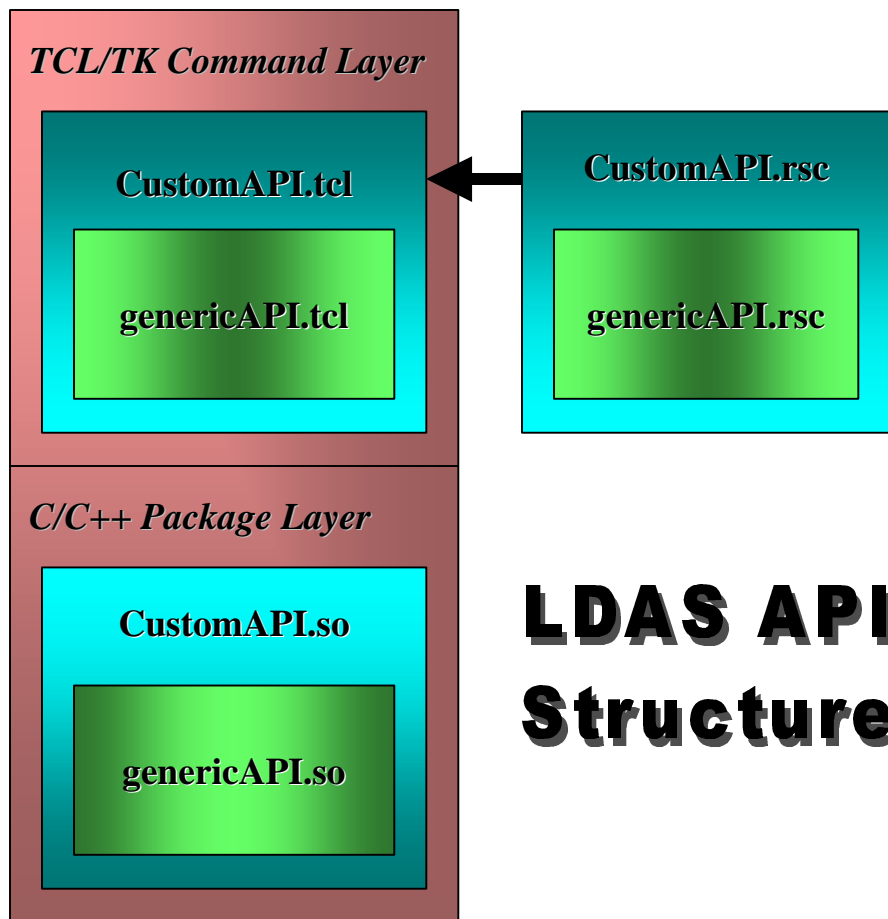


UIs

APIs

Libraries

LDAS API Layers



LDAS API Structure

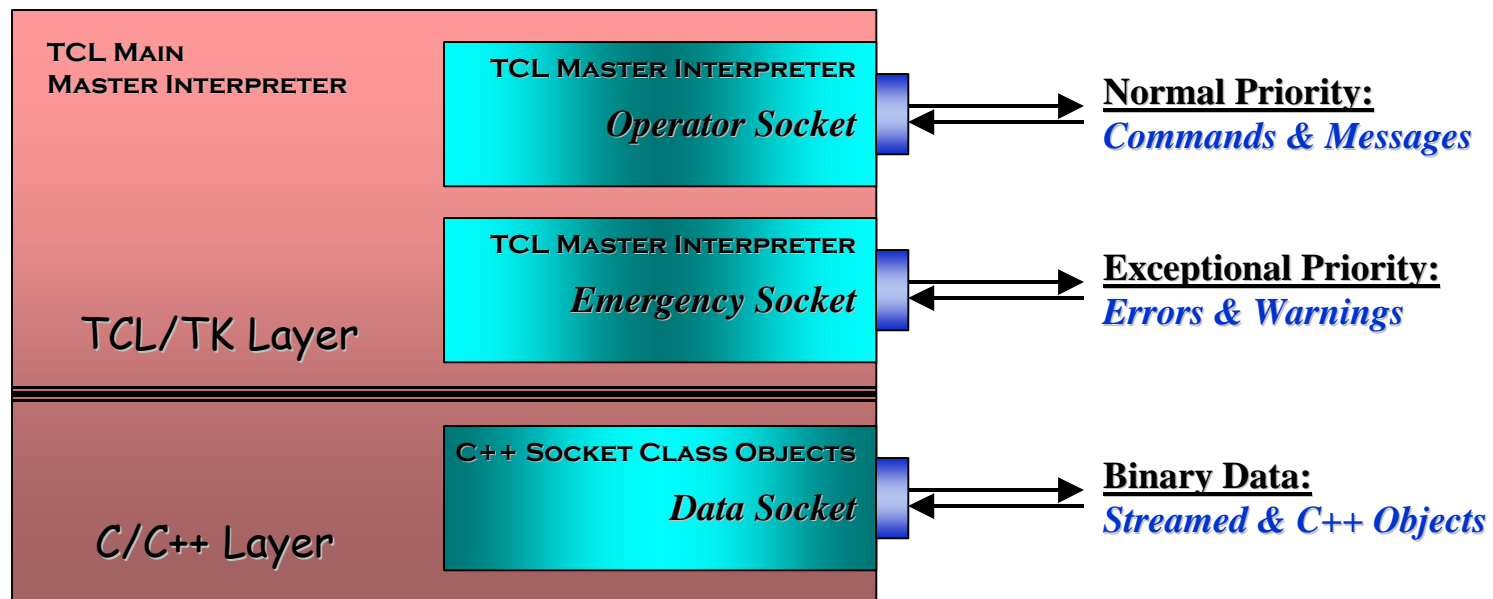
LDAS APIs:

- ① **Two Layers:**
 - ⇒ **TCL/TK**
 - ⇒ **C/C++ (extends TCL Language)**
 - ⇒ **SWIG Unifies Layers**
- ② **GenericAPI (core) Module:**
 - ⇒ **Communications**
 - ✓ **TCL <-> C++**
 - ✓ **API <-> API**
 - ⇒ **Common TCL proc s:**
 - ✓ **Help**
 - ✓ **Logging**
 - ✓ **Command Socket Management**
 - ✓ **Resource Management**
 - ⇒ **Common C/C++ methods:**
 - ✓ **Data Socket Management**
 - ✓ **Internal Data Management**
 - ✓ **Class Save & Restore**
- ③ **Custom (specialization) Module**

Inter-API Communications

3 Types of Socket Communications in APIs:

- ① Operator Sockets - Normal Inter-process Commands & Messages
- ② Emergency Sockets - Error & Warning Commands & Messages
- ③ Data Sockets - Binary Data in either Raw Streams or C++ Objects



managerAPI

- The brain of LDAS. (Sockets are analogous to the central nervous system).
- All user requests sent here.
- Assigns one of 10 assistant managers to carry out individual user requests (beyond 10 jobs get queued).
- ...these use macros specific to the ldas command to perform flow control.
- Reports completion or errors to users via email or socket.

frameAPI

- Reads in and writes out data in the frame format.
- Selects out the channels of interest to the analysis.
- Concatenates data into longer time series, useful to the particular type of analysis.
- Converts data into ILWD format for transmission to other LDAS APIs.

lightWeightAPI

- Reads in and writes out data in LIGO_LW format.
- Translates LIGO_LW to and from ILWD format.
- Sends ILWD data to other LDAS APIs.

metaDataAPI

- Receives table specific ILWDs from other APIs such as the dataConditionAPI, eventMonitorAPI or lightWeightAPI for insertion into LIGO database.
- Extracts data from LIGO database using standard SQL allowing datamining operations.
- Translates table data extracted from LIGO database into ILWD for transmission to other LDAS APIs.

dataConditionAPI

- Cares out pre-conditioning/filtering steps on LIGO data (first stage of any analysis)
 - Supports FFTs, Linear Filtering, Mixers, Power Spectral Estimation, Cross Spectral Estimation.
 - Planned support includes line removal, regression, time-frequency data construction.
- Includes an interpreted language similar to matlab for customization of pre-conditioning.

mpiAPI

- Initiates the mpirun command associated with the parallel searches.
- Supports multiple concurrent wrapperAPI jobs.
- Maintains a communication link with wrapperAPI for status, monitoring and load balancing.
- Maintains available node list for job sharing of a common resource (LDAS Beowulf).

wrapperAPI

- Carries out the MPI (Message Passing Interface Standard) parallel search.
- Reports status, progress, errors, efficiency to the mpiAPI.
- Sends results (data products) of the search to the eventMonitor for interpretation and forwarding.
- Dynamically loads search codes having the wrapperAPI - LALwrapper compliant interface.
- Stand-alone version available for development.

eventMonitorAPI

- Receives data products from the wrapperAPI.
- Manages data products concurrently from multiple wrapperAPI jobs.
- Parses these into their separate (table, user, state) components and forwards to appropriate locations.
 - Table data forwarded to metaDataAPI
 - User data forwarded to lightWeightAPI or frameAPI.
 - State information cached to disk for later wrapperAPI jobs.

controlMonitorAPi

- Consists of a server and a client.
- Client is GUI based (TCL).
- Can be used to monitor running jobs, log files, beowulf cluster.
- Control features require super user status.
- Custom filters can be used to simplify monitoring.

diskCacheAPI (TBD)

- Makes frame data sets available to the frameAPI, retrieving from archive, tape, disk when necessary.
- Manages many terabytes of spinning disk cache in LDAS.
 - Tracks most used data sets.
 - Currently cached data sets.
 - Disk space utilization.
 - Clean-up of disk space (purging rules).

dataIngestionAPI (TBD)

- Ingests data stored in frames from sites into the LIGO archive.
- Carries out any trivial rearrangement of data structure (e.g. duplication of data with different stripping)
- Writes custom (reduced data products) to tape.
- Supervisor of LDAS tape drives.

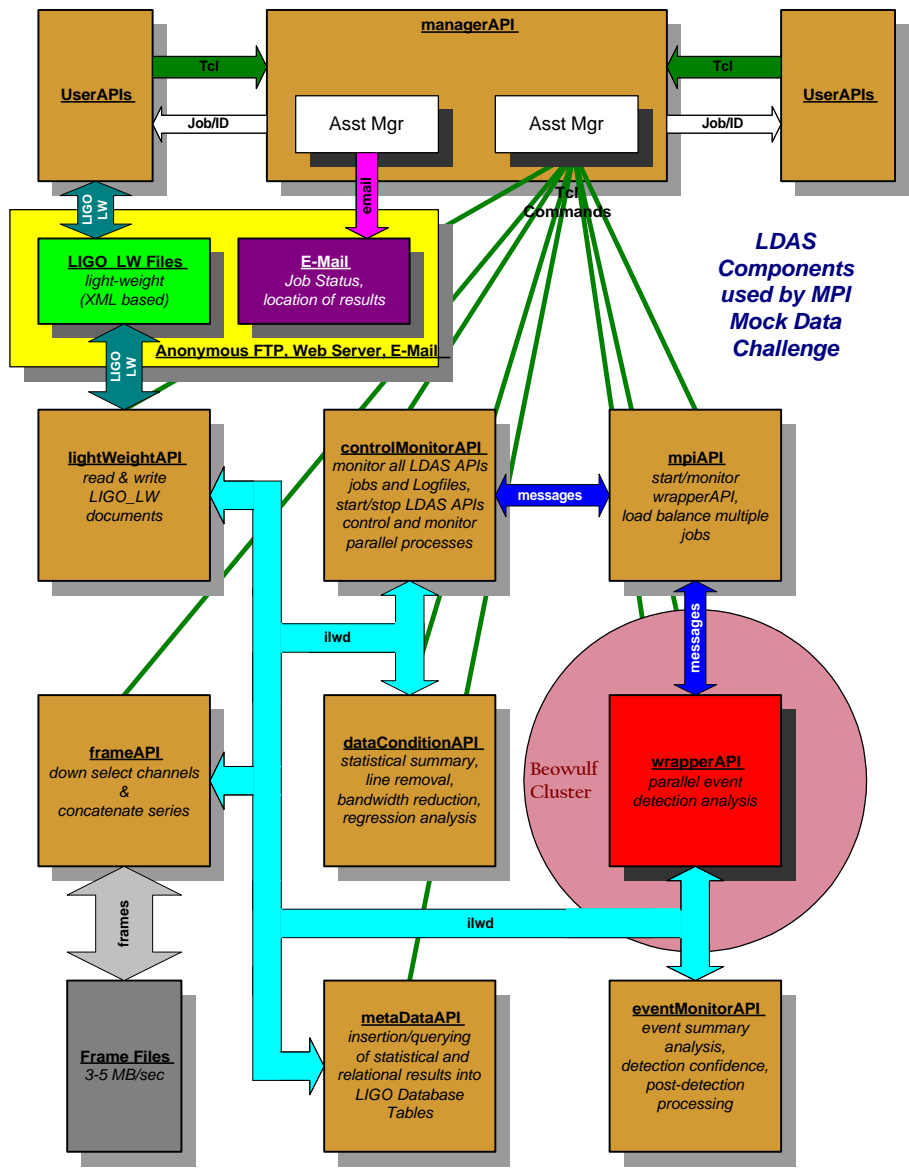
Status Summary

- LDAS Release 0.0.18 (later this week)
- We are still developing at the alpha level.
 - Two major APIs between us and beta LDAS.
 - diskCacheAPI (disk management / data availability)
 - dataIngestionAPI (tape control).
- Approximately 600,000 lines of code in place.
- Represents ~80-90% completion of LDAS.
- Evolution of LDAS usage will uncover new flow control models requiring new user commands.

LDAS/LAL Anatomy

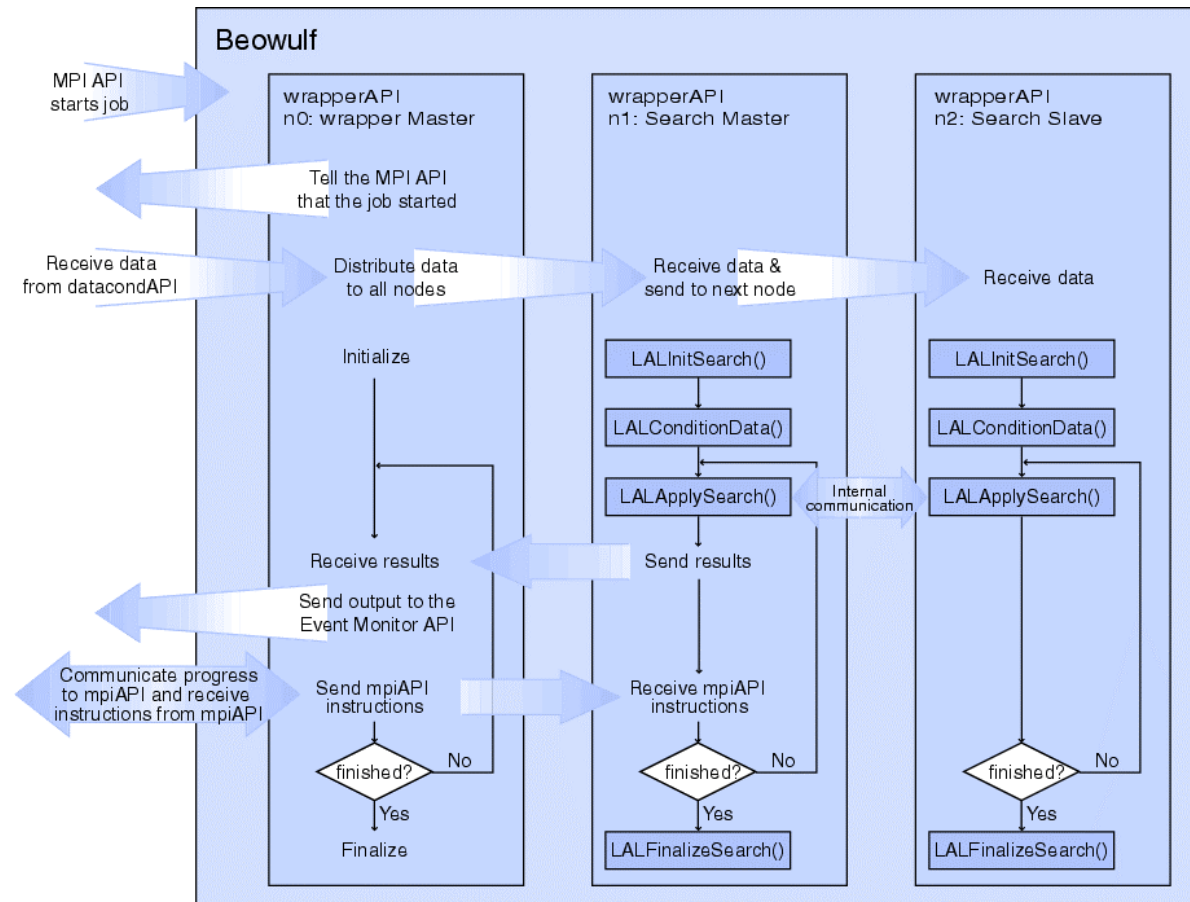
Data Flow Model
(putting it all together)

LDAS DataPipeline (Today's View)



- **UserAPI** connects to **managerAPI** and starts pipeline.
- **managerAPI** assigns *assistant manager* to control job.
- **frameAPI** reads *frames* into the LDAS system, downselects and sends data to the **dataConditionAPI**.
- **dataConditionAPI** pre-conditions data and forwards this to the **wrapperAPI** under the control of **mpiAPI**.
- **wrapperAPI** performs template based search using a dynamically loaded shared object.
- **eventMonitorAPI** receives search results and directs events to the **metaDataAPI** where they are placed in database.
- **eventMonitorAPI** sends user data products from search to **lightWeightAPI** where they are sent to user via FTP or URI commands.
- **managerAPI** notifies user via email of job completion.

WrapperAPI/LALwrapper/LAL



eventMonitorAPI

Output Structure:

- User data
- State data
- DB table data

Other Things to Look Into

- Surf the LDAS home page
 - Dozens of documents, presentations, source code can be found here.
- Learn to use the GNATS Problem Tracking System.
- Read the recommended LAL and LALwrapper documentation for this workshop.