The LIGO Data Analysis System: LDAS

"A Users' Perspective"

KENT BLACKBURN
LIGO LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
FEBRUARY 23RD, 1999
LIGO-G990011-00-E

The LDAS Team:

- Stuart Anderson
- Kent Blackburn
- Phil Ehrens
- Dave Farnham
- Xiao Hu
- Barbara Kratochwill
- Albert Lazzarini
- Walid Majid
- Ed Maros
- Tom Prince
- Bruce Sears
- Roy Williams

Overview of Talk:

- Primary Purpose, Goals & Demands
- Tour of Hardware Design
- Tour of Software Design
- Supporting Users
- Further Reading
- Closing Remarks

Primary Purpose:

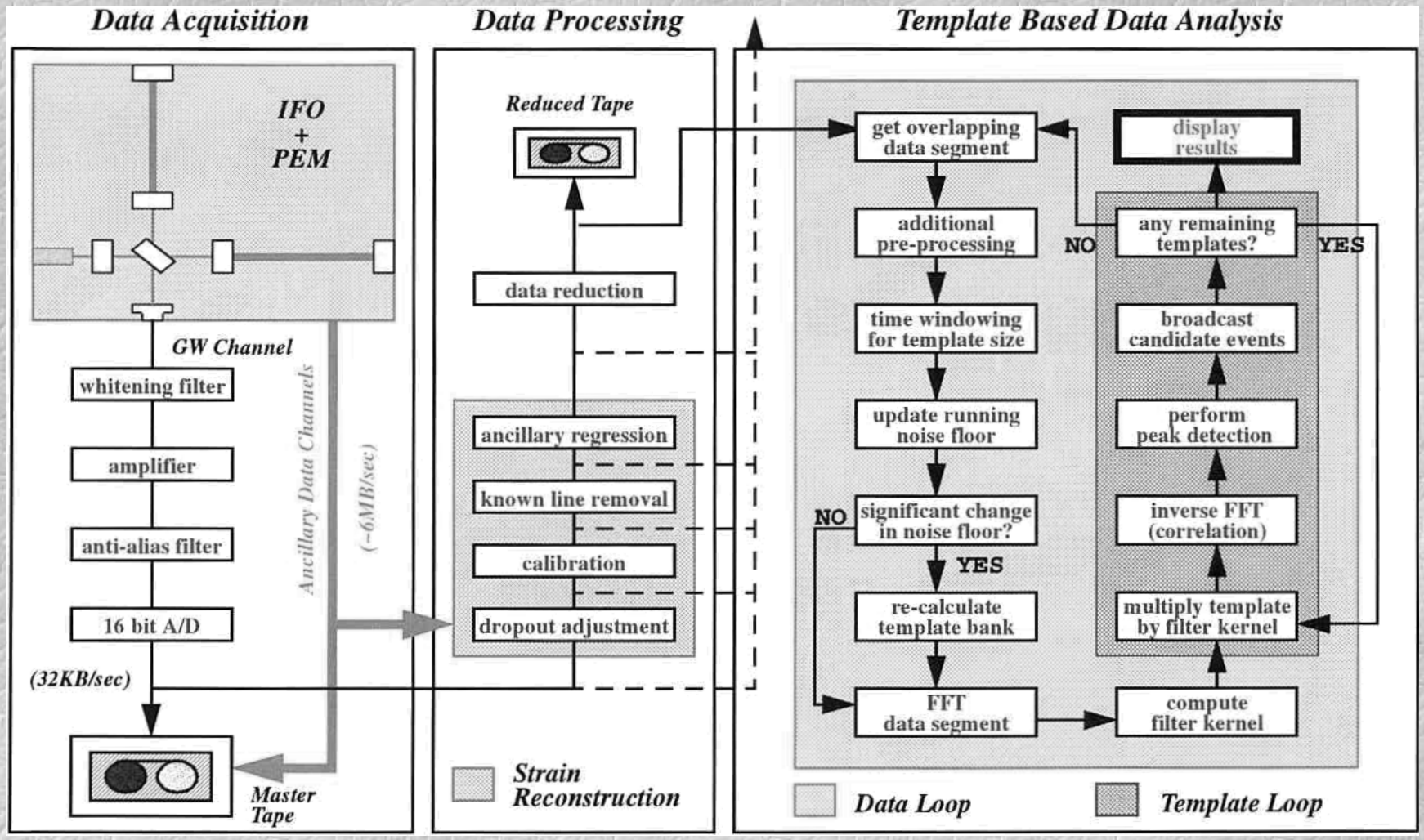
- First and foremost, the LIGO Data Analysis System is being implemented to detect and characterize gravitational waves from astrophysical sources.
- In addition, the LDAS will perform:
  1. raw frame data archival,
  2. database management functions for
     - raw frame data descriptors,
     - diagnostic trigger descriptors,
     - and astrophysical filter (template) event descriptors,
  3. data & metadata distribution services.

Principal Sources:

- Binary Inspiral of Compact Stellar Objects
  - neutron star - neutron star
  - black hole - neutron star
  - black hole - black hole

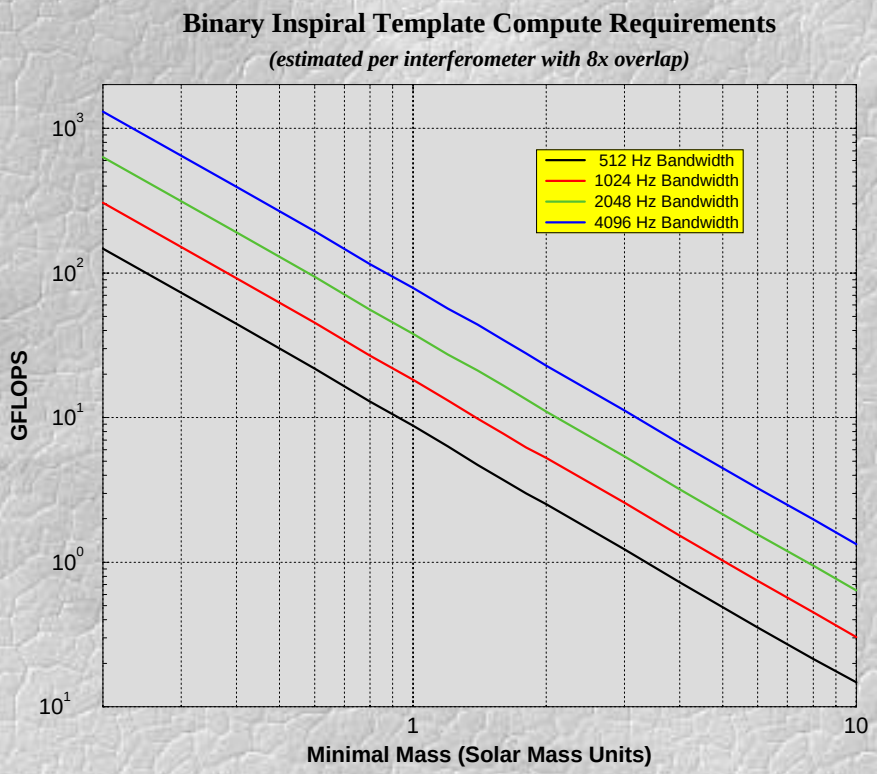
  } dominate LDAS requirements
- Burst Events
  - supernovae (requiring coincidence between sites)
- Periodic Sources
  - pulsars (all sky unlikely, but targeted searches easily carried out)
- Others
  - black hole ring-downs
  - black hole mergers
  - stochastic background
  - serendipitous discoveries!

Data Flow Model:

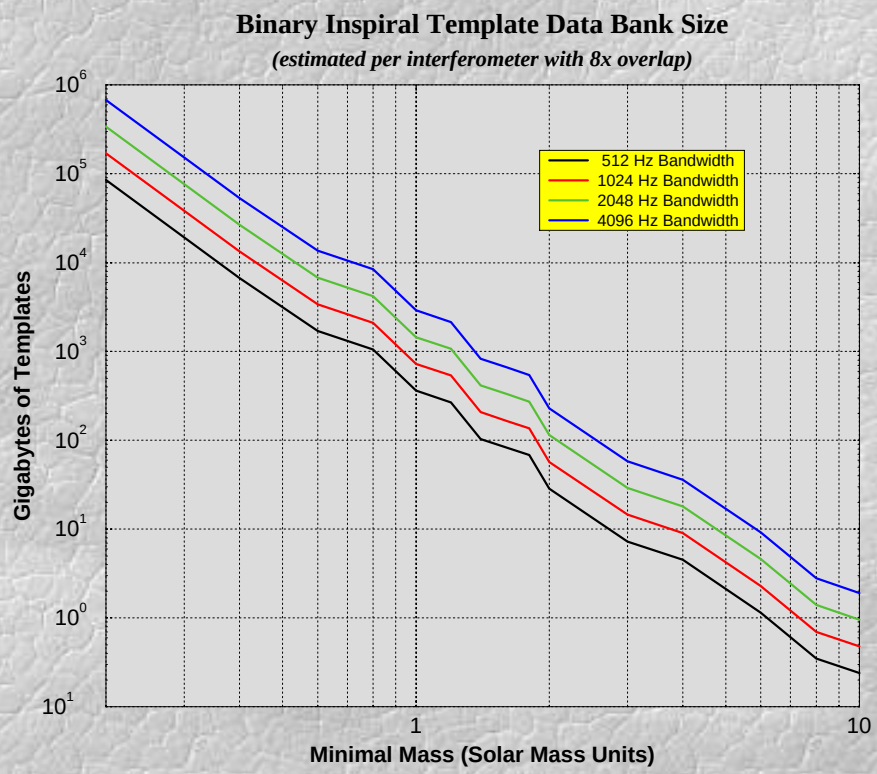


Optimal Filtering Demands:

Computation:



Templates:



Hardware Design Needs:

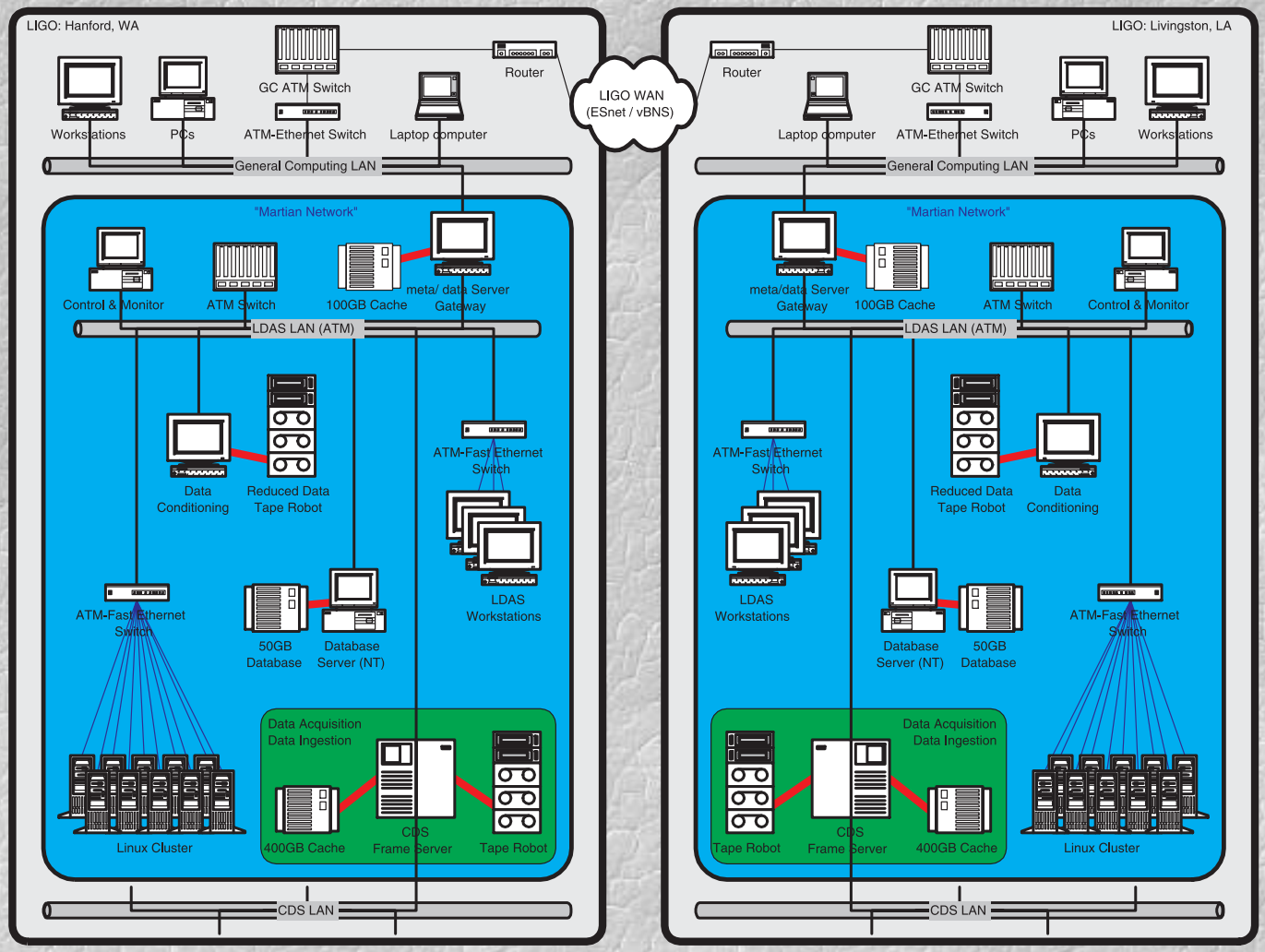
Requirements:

1. *10-100 GFLOPS of Compute Performance & 0.5-5 TB of uncompressed Templates*
2. *100-620 megabits per second point-to-point*
3. *500 GB of On-Line Disk Cache*
4. *50-500 TB Archived Data per Year*
5. *50-500 GB of metadata per Year*
6. *LAN & WAN Networks*

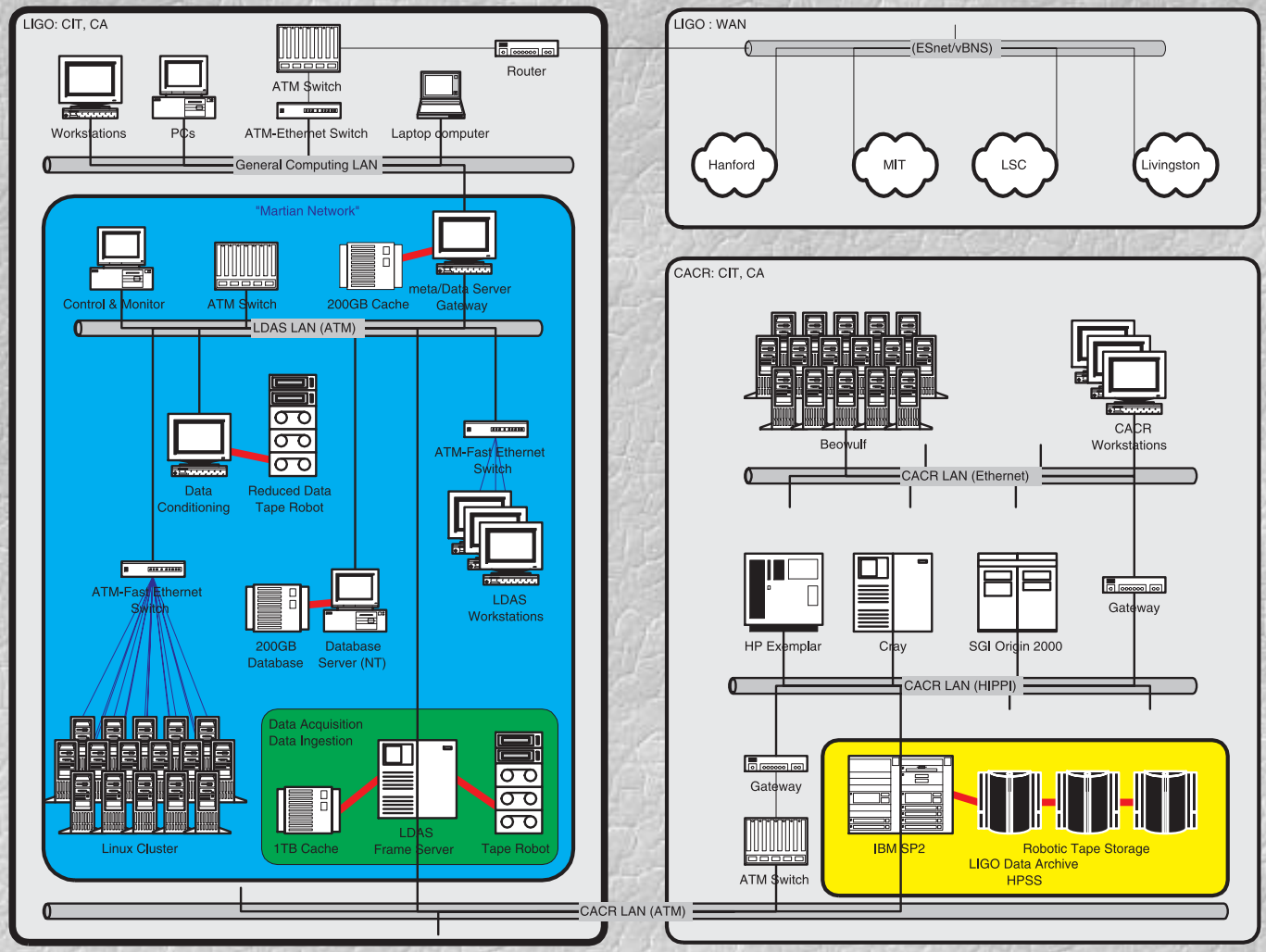
Technologies:

1. *Beowulf (PC-Linux Clusters)*
2. *Switched ATM & Fast-Ethernet for LANs*
3. *SCSI Hard Disk Storage*
4. *HPSS, Tapes (optical?)*
5. *Database Capable of handling this Volume*
6. *LIGO ATM & Fast-Ethernet LANs with ESNet and vBNS connectivity between the Observatories & Archive*

Hardware Architecture(online):



Hardware Architecture(offline):



Hardware Status:

- LDAS Hardware Design Relatively Complete
  1. Procurement being delayed/staged as much as possible
  2. PDR tentatively scheduled for March 11th
- WAN to Sites Established
  1. T1 links at both Hanford, WA & Livingston, LA
- First LAN Hardware (ATM Switch) Arrived at Hanford
- Compact Beowulf Cluster Received from ALTA Technologies for Prototyping
- meta/Data Server Hardware Identified in conjunction with CDS Frame-Builder
- Establishing Archival Storage Specifications in collaboration with CACR

Software Design Needs:

Requirements:

1. *Portability -*
   - POSIX for UNIX I/O
   - ANSI C/C++ for Compilers
   - TCL/TK for Steering
   - MPI for Parallel Computing
   - ODBC for Database Clients
2. *Extensibility -*
   - Modular/Reusable Code
   - OOP Design
3. *Maintainability -*
   - OO Languages Where Practical
   - CVS Source Code Management
4. *Flexibility -*
   - Class/Object Design
   - Modular Libraries
   - Distributed Processing

Components:

1. *Data Formats -*
   - IGWD Frames
   - Lightweight (Metadata, Events, Templates, Communications ...)
2. *Libraries -*
   - Supported Data Format I/O
   - Numerical (FFT's, filters, etc.)
   - POSIX & Socket interfaces
3. *LDAS API's -*
   - Supervisors to Data I/O
   - Control, Monitor, & Management
   - MPI Level Communications
   - Filtering and Analysis
4. *UI's -*
   - TCL/TK (Wish Shell) GUI
   - TCL Interpreter Command Line
   - TCLet Plug-ins to Web Browsers

Standardized Data Formats:

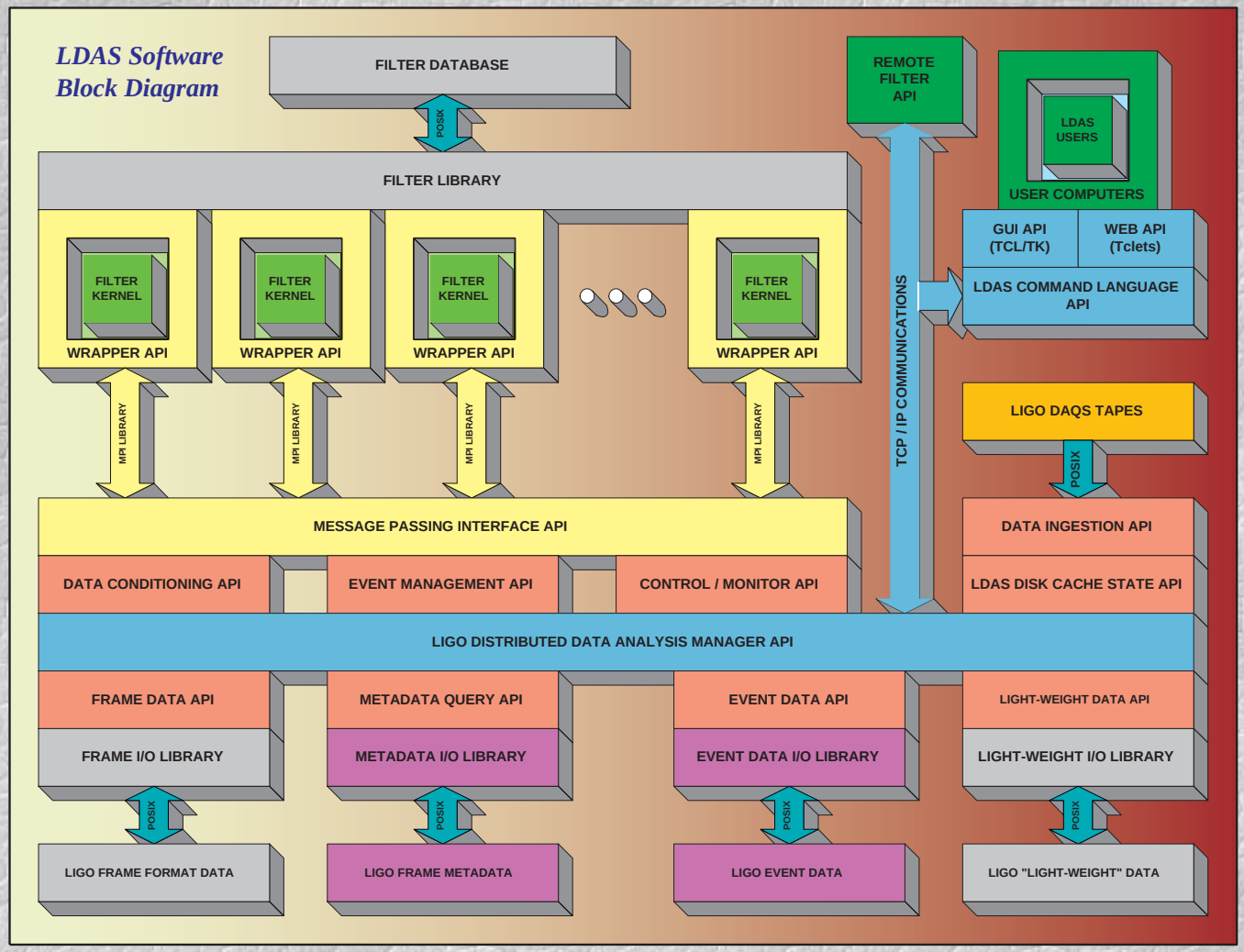
Frame:

1. Structured (C-like) Format
   - Samples of these structures:
     - ✓ ADC, Static, Detector, Trigger, ...
     - ✓ Simulated, History, Logs, ...
2. Jointly Developed with VIRGO
   - LIGO-T970130-B-E
   - VIRGO-SPE-LAP-5400-102
3. Original I/O Library in C
4. C++ Class I/O Library
5. Primary Uses:
   - Data Acquisition
   - Data Archival
   - *Subsystem for Diagnostics*

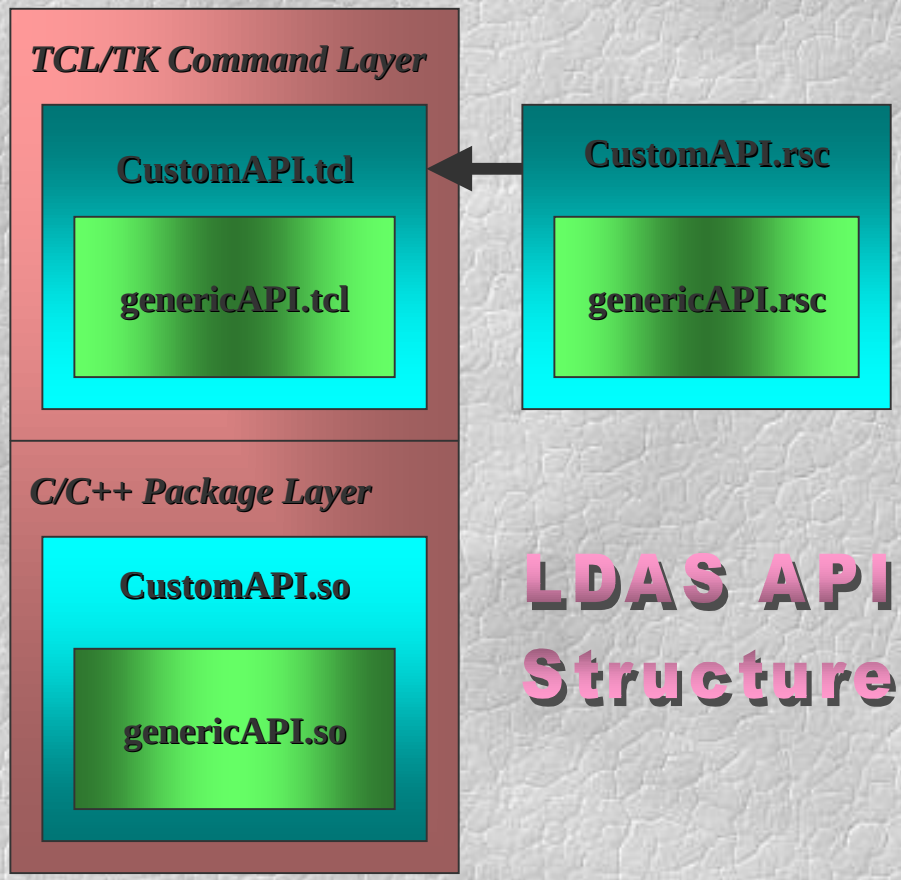
Lightweight:

1. Tagged (XML-based) Format
2. Easily Parsed *(and written)*
   - <int_s format="ascii">57, 7, 15, 12, 15, 31, 755</int_s>
3. LIGO Defined Objects
   - tables (n-tuplets),
   - arrays (matrix, vector),
   - vectors (time-series, power-spectra)
   - Some Revisions expected
4. Use Complement Frames!
   - LIGO Event data
   - LIGO Metadata
   - Spectra & Time-series data
   - Inter-process Communications

Software Architecture(block):



Layered LDAS API Design:



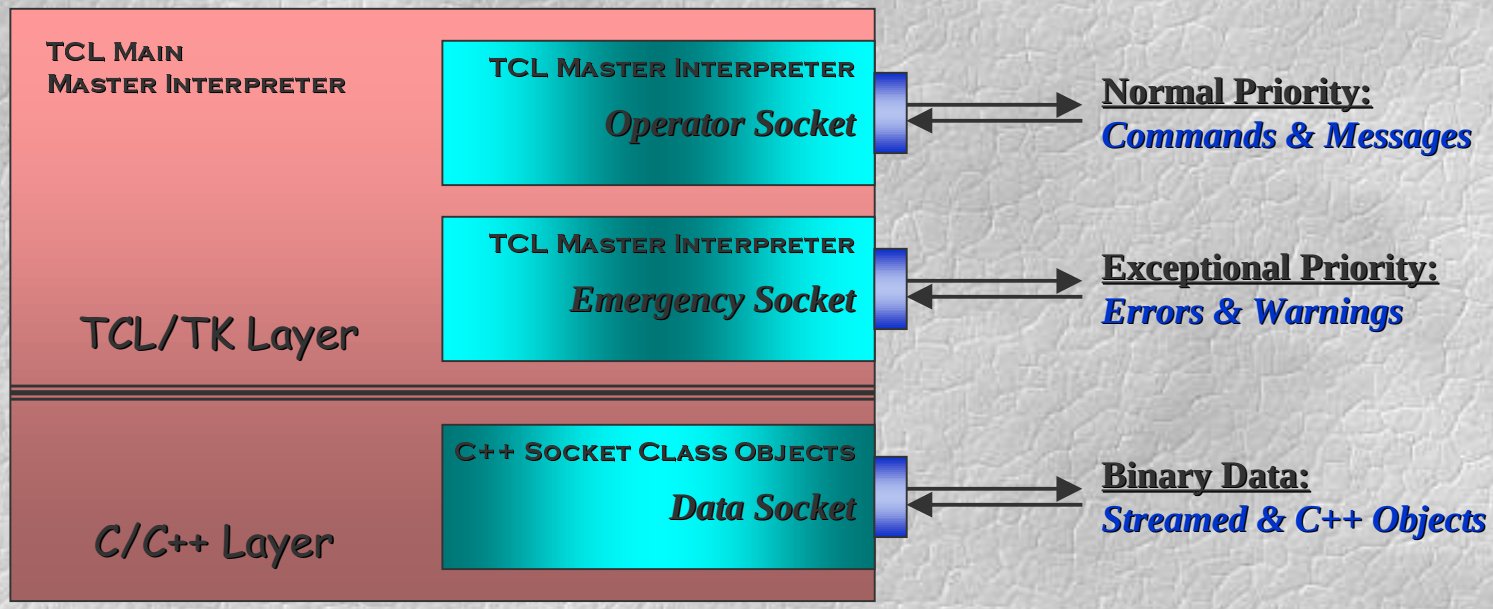
LDAS API Structure

LDAS API's:

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*

LDAS API Communications:

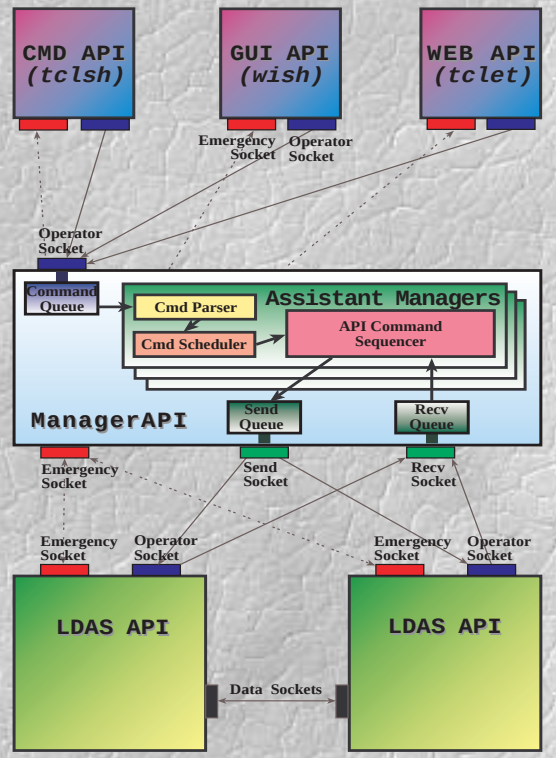
- **3 Types of Socket Communications in API's:**
  1. Operator Sockets - Normal Inter-process Commands & Messages
  2. Emergency Sockets - Error & Warning Commands & Messages
  3. Data Sockets - Binary Data in either Raw Streams or C++ Objects



Distributed API Supervisor:

Manager API:

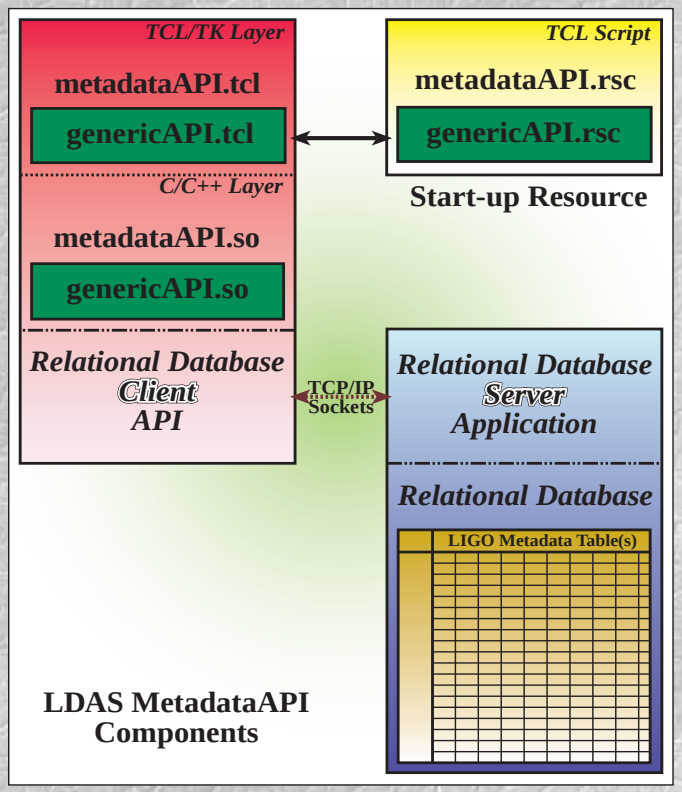
1. Two Supervisory Layers:
   - Single "top-level" Manager
   - Multiple (3-10) Assistant Managers
2. Manager Layer:
   - Communications
     - ✓ *Manages sockets & queues*
     - ✓ *Interface for User API's*
     - ✓ *Point of contact for All API's*
   - Exception Management
   - Log File Management
   - System Operator Interface
3. Assistant Manager Layer:
   - Command Execution:
     - ✓ *Parsing*
     - ✓ *Schedule Building*
     - ✓ *Command Sequencing*
   - Reports to Manager Layer



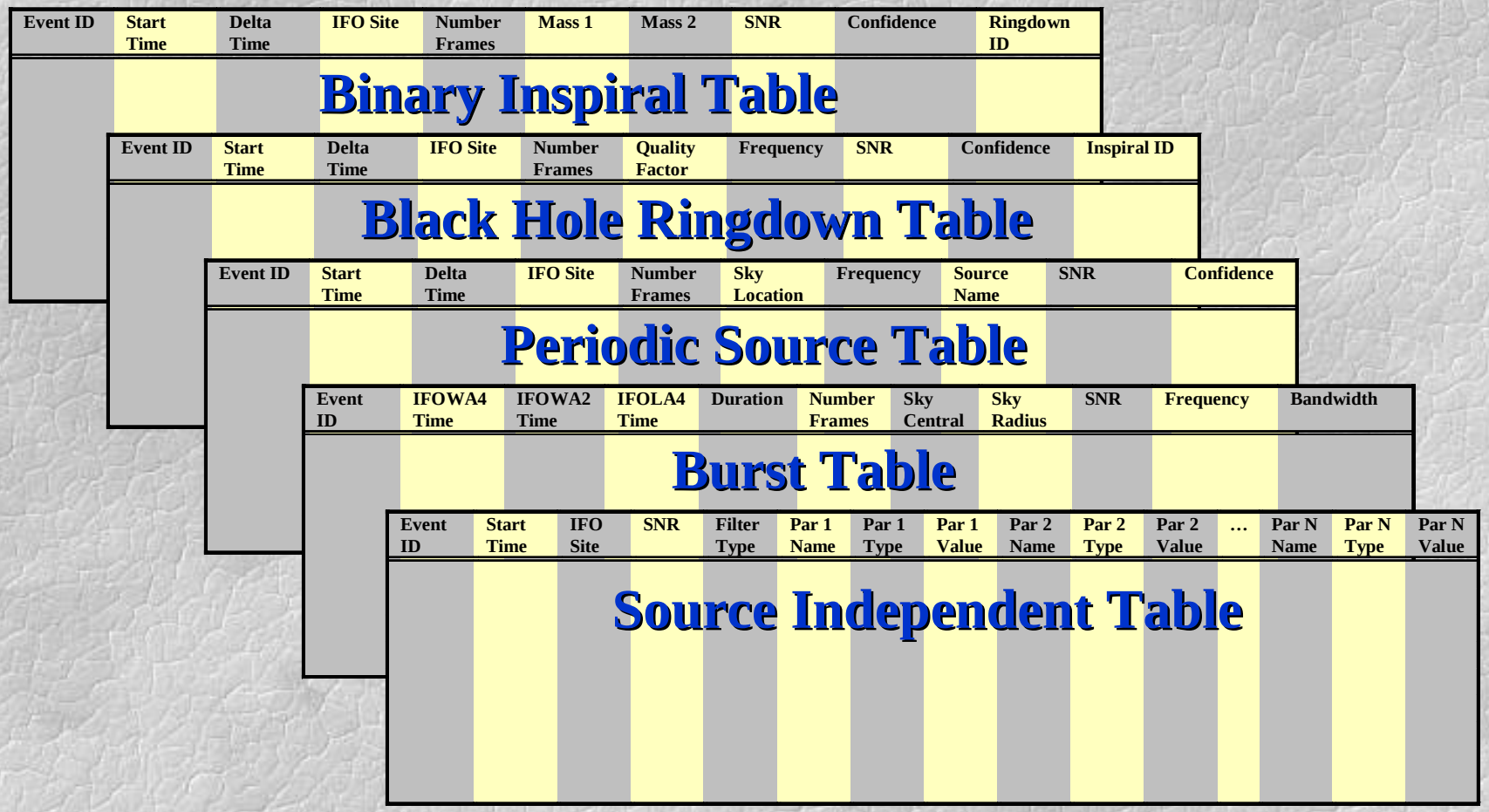
Distributed Database API:

Metadata API:

1. Relational Database:
   - Using ODBC Standard for Calls
   - Currently Developing with DB2
   - Server can be Unix or NT Based
2. Table Contents:
   - Frame Characterization:
     - ✓ *Descriptors*
     - ✓ *Simple Statistical Summary*
   - Event Characterization:
     - ✓ *Optimal Filter Results*
     - ✓ *Astrophysical parameters*
   - GDS Trigger Results
   - CDS State Vectors
   - LDAS Logs
3. Metadata Query Sevices
   - LDAS Processing Queries
   - Data-Mining Queries

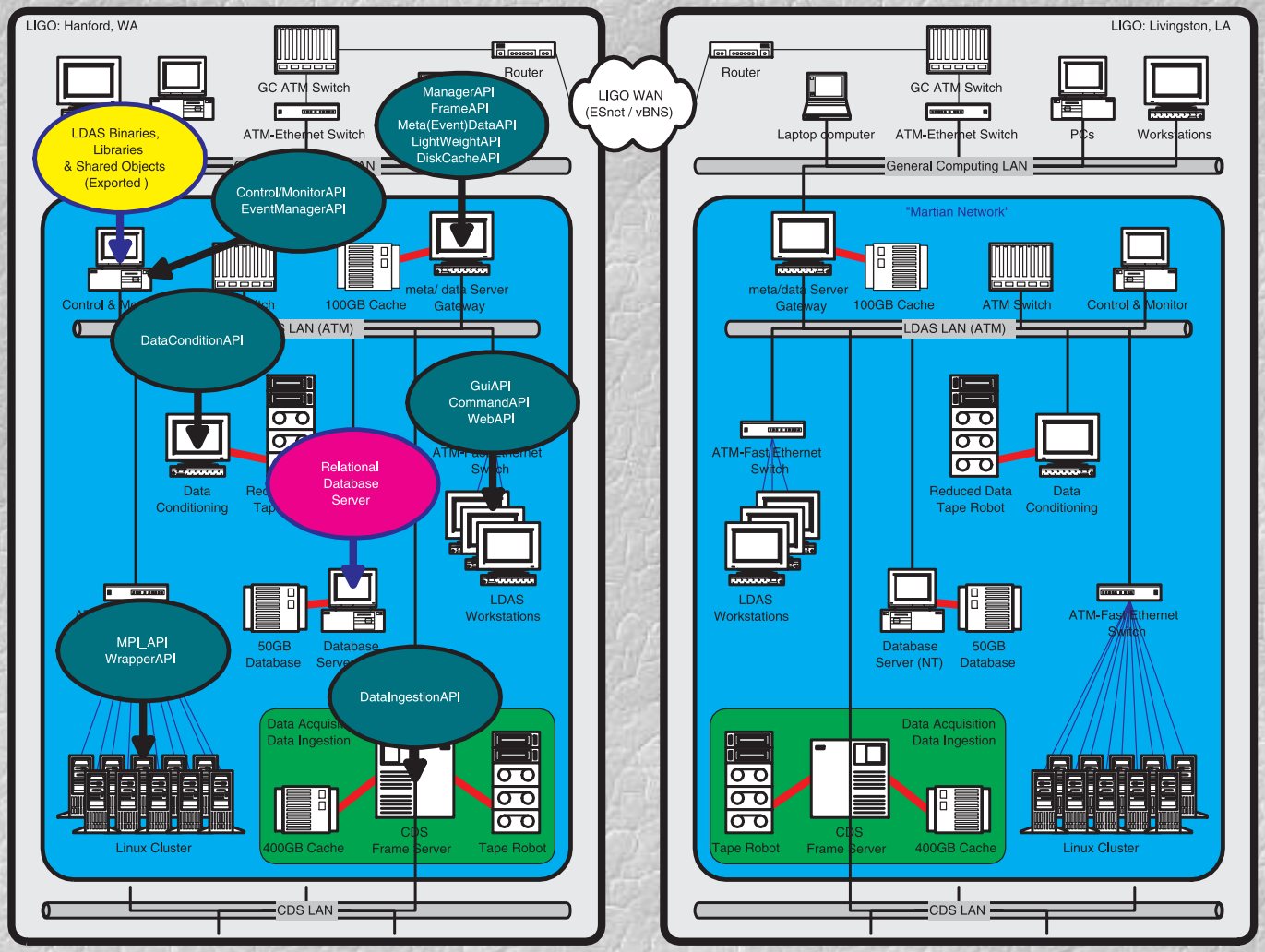


Database Event Tables:

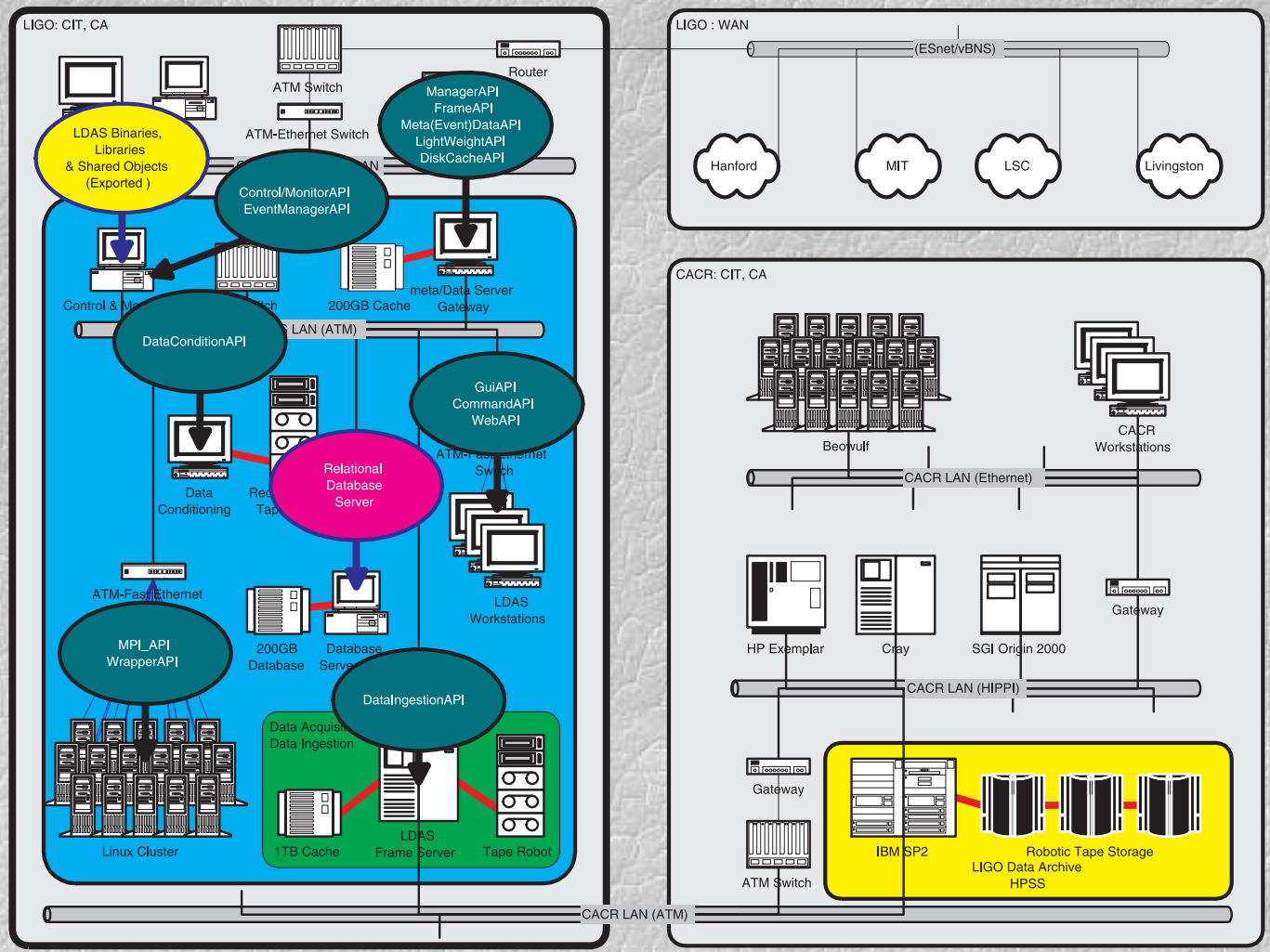


see URL: http://www.ligo.caltech.edu/~xhu/index.html for Event, Frame, GDS, Log, CDS Table Descriptions

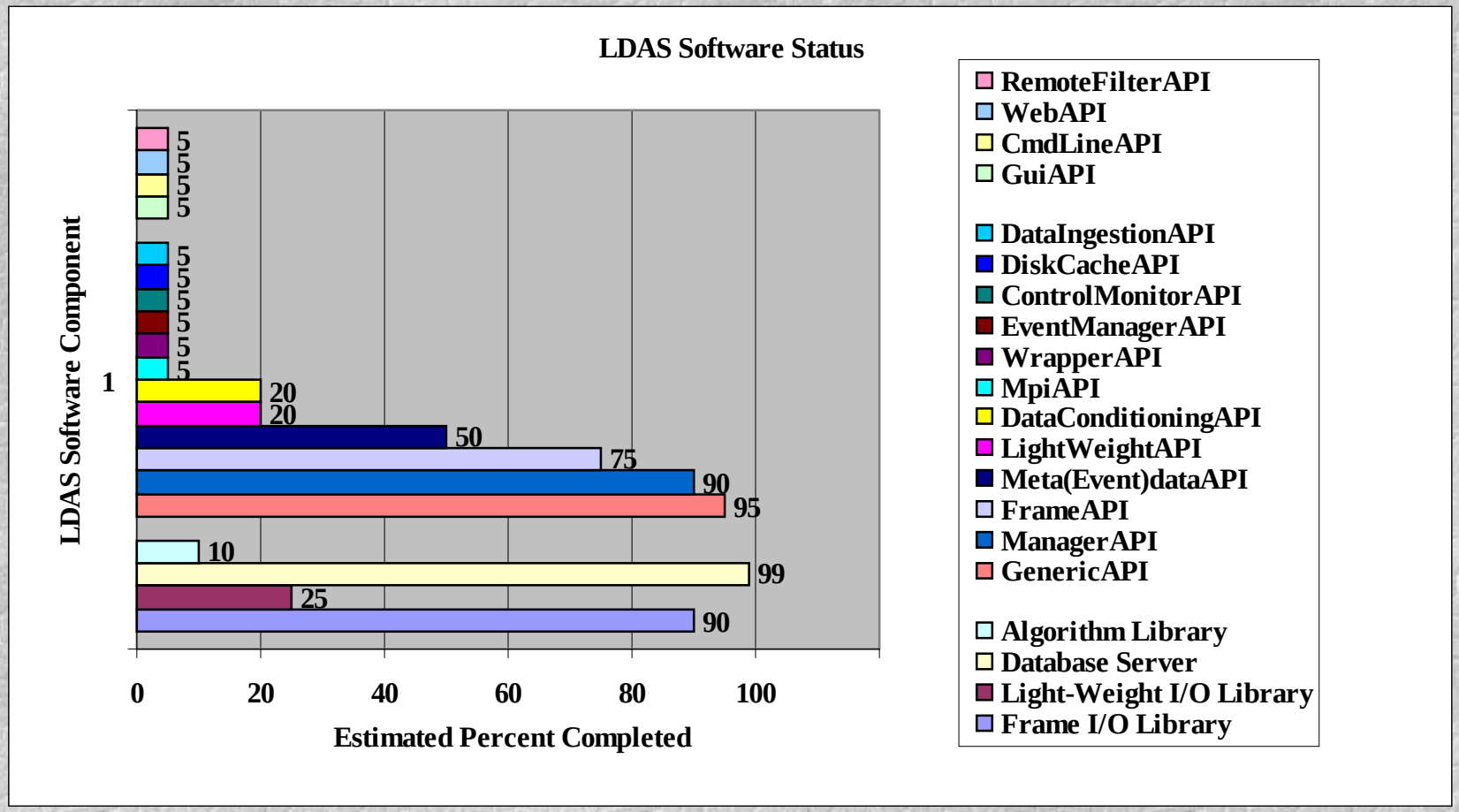
Software Mapping(online):



Software Mapping(offline):

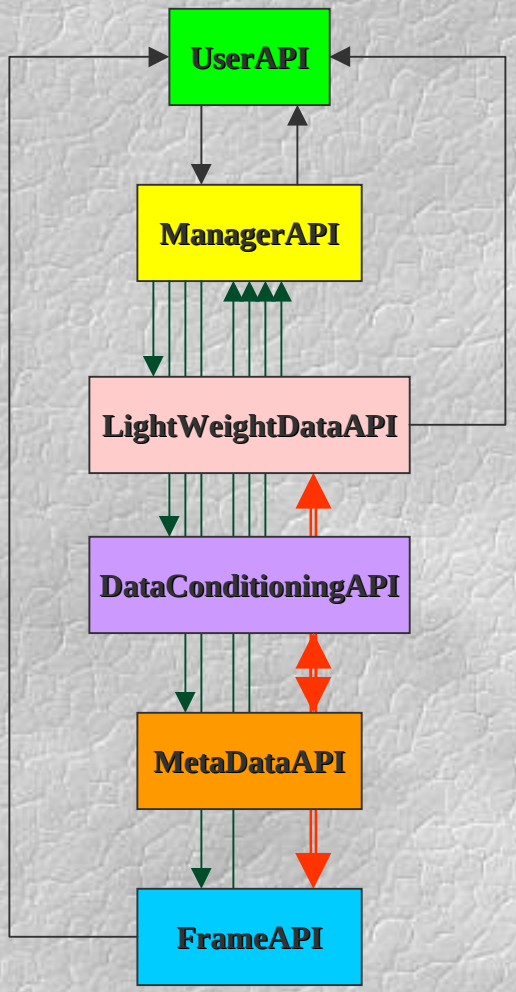


Software Status:



First Software Deliverable:

- **α-release Spring 99?**
  1. LDAS Components to be included:
     - GenericAPI
     - FrameAPI
     - Meta(Event)dataAPI
     - LightWeightDataAPI
     - DataConditioningAPI
     - ManagerAPI
     - Simple UserAPI
  2. Some "Quick Look" Capabilities
  3. Support of Site Installation Activities
  4. Provide User Base for Testing and Debugging System



LDAS User Classifications:

- Perusers - Browsers of Data Products
  1. Connection: Modem or Ethernet
  2. Tools of Choice: GUI & Web Browser
- Users - meta/Data Flow Manipulators *+ Perusers*
  1. Connection: Modem, Ethernet, LDAS LAN or WAN
  2. Tools of Choice: GUI or Command Line Interface
- Developers - Algorithm/Infra-structure Coders
  1. Connection: Ethernet, LDAS LAN or WAN
  2. Tools of Choice: Raw Data or Sockets Directly into LDAS Data Flow
- Processes - Distributed Processes of LDAS
  1. Connections: LDAS LAN or WAN
  2. Tools of Choice: LDAS Command and Data Sockets Protocols

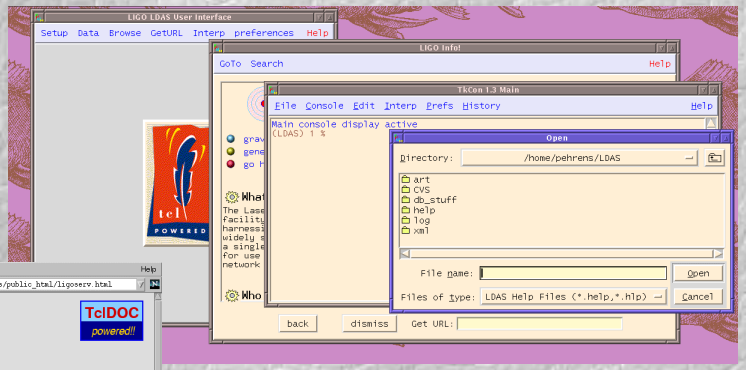
LDAS User Interfaces:

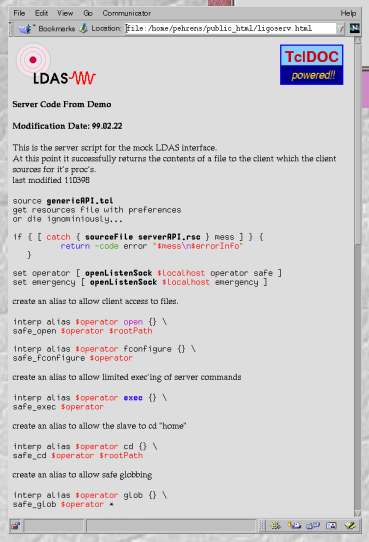
- **User Interface model based on TCL/TK**
  1. Doesn't preclude other software languages:
     - Any environment that can send LDAS Commands, properly formatted, to the ManagerAPI's Command Socket and receive LDAS Light-Weight Data Format can be made to work with LDAS!
  2. Portable between Unix / Windows / Mac Operating Systems
- **3 Types of User Interfaces Based on TCL/TK to be provided:**
  1. Command Line Interfaces - TCL shell scripts which communicate with the LDAS ManagerAPI sockets
  2. Graphical User Interfaces - TK Wish shell widgets which communicate with the LDAS ManagerAPI's sockets
  3. Web Browser Interfaces - TCLet plug-ins that display widgets in web browsers & communicate with the LDAS ManagerAPI's sockets

"Smart" User Interfaces:

Initial Limited UI

1. Client/Server Socket Links
2. Server sends TCL/TK Code

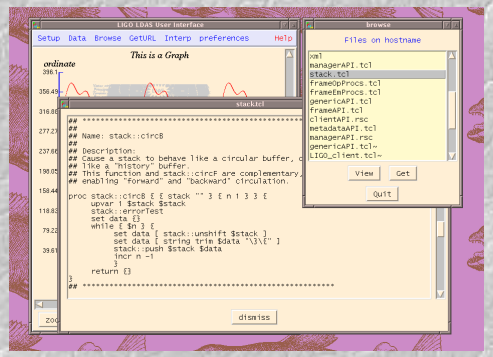


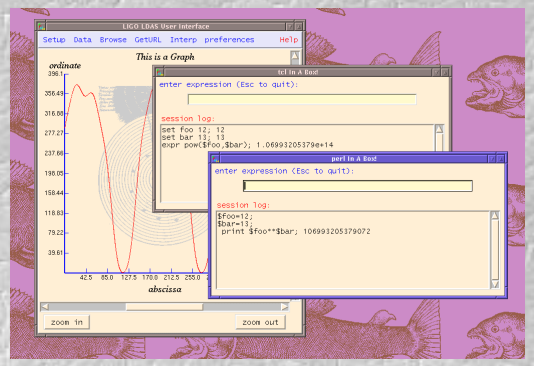


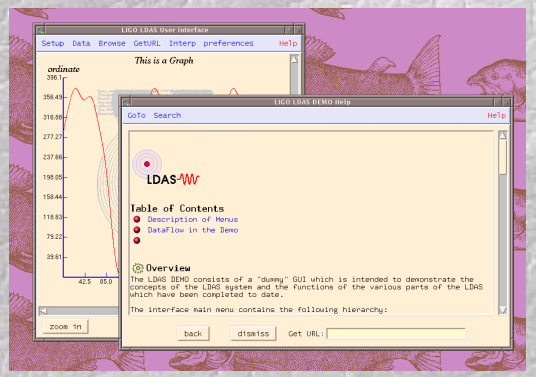
Socket Based Communications

Final Custom UI

1. Appended Functionality to UI
2. Provideds centralized UI Code Management







LDAS Documentation:

- On the Web: LIGO Data Working Group Bulletin Board
  1. http://www.ligo.caltech.edu/~prince/LDCG_lsc/LDCG.html
     - LDAS Technical Review Documents
     - LDAS Software Guidelines
     - LDAS Software Requirements
     - LDAS Software Specifications
     - LDAS Code Formatted as HTML
     - LDAS Presentations
     - Other Useful Documentation
  1. LDAS Source Code Presentation
     - C++ to HTML using DOC++
     - TCL/TK to HTML using TCLDoc
     - CVS Server to LDAS Software Repository (password protected)

Closing Remarks:

- Primary Purpose - Detection Of Gravitational Waves Using LIGO Data
- Design Is Portable, Extensible, Maintainable & Flexible
- Supports Users At Multiple Levels
  - e.g., Perusers, Users, Developers, Processes
- Powerful "Smart" User Interfaces Based On TLC/TK Have Been Prototyped
- Significant Amounts Of Coding Still Ahead!