


# LIGO Data Analysis System "LDAS"



*XXXIV<sup>th</sup> Rencontres de Moriond  
Gravitational Waves & Experimental Gravity  
January 23<sup>rd</sup>-30<sup>th</sup>, 1999*

*James "Kent" Blackburn  
LIGO Laboratory, CALTECH  
LIGO G990090-00-E*

# LDAS Primary Purpose

---

- First and for most, 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 data archival,
  - ② database management functions for
    - ⇒ raw data descriptors,
    - ⇒ diagnostic triggers,
    - ⇒ and astrophysical filter (template) events,
  - ③ data & metadata distribution services.

# LDAS Principal Sources

## ■ Binary Inspiral of Compact Stellar Objects

- ① neutron star - neutron star
- ② black hole - neutron star
- ③ black hole - black hole

} dominate LDAS requirements

## ■ Burst Events

- ④ supernovae (requiring coincidence between sites)

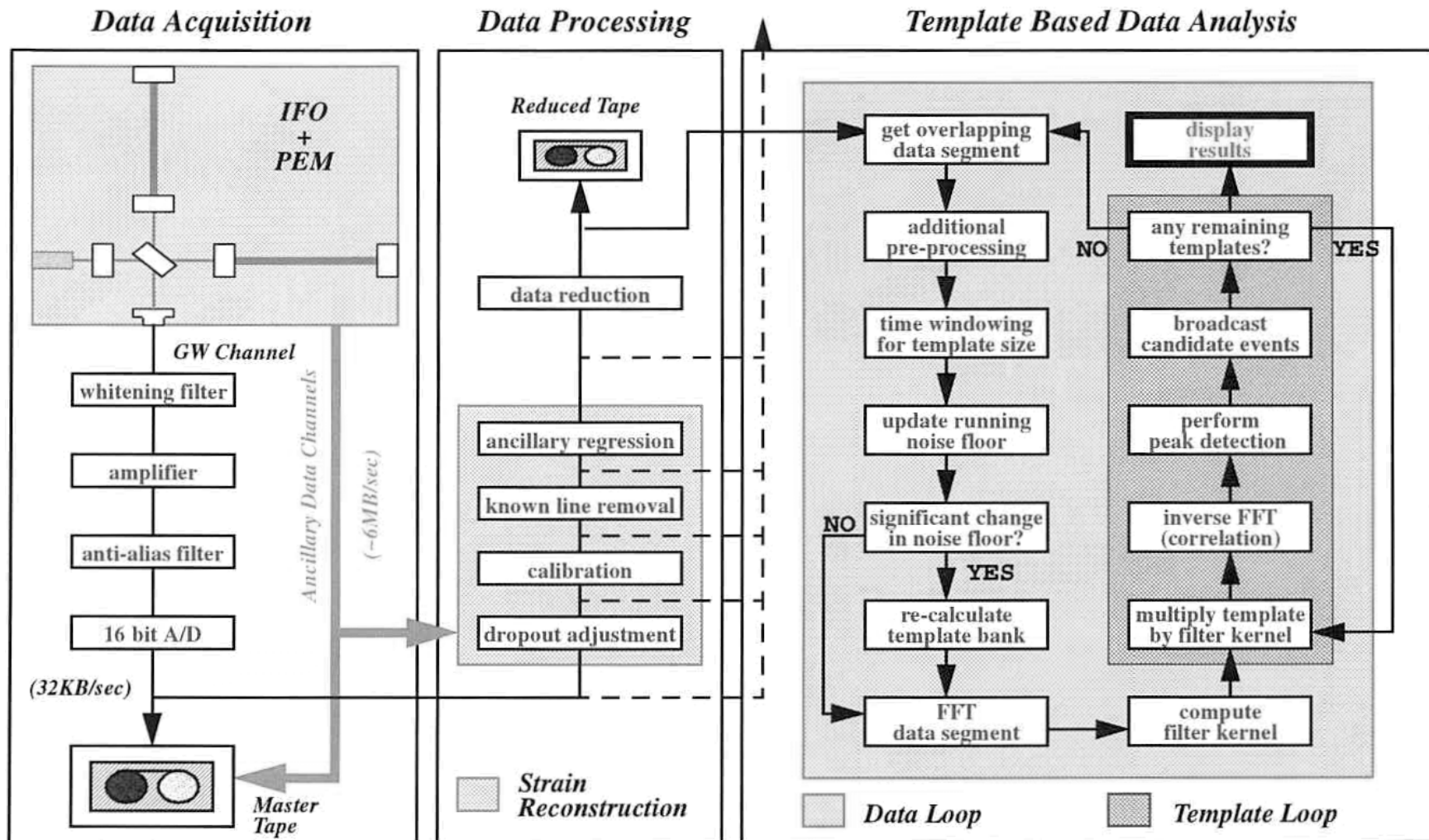
## ■ Periodic Sources

- ⑤ pulsars (all sky unlikely, but targeted searches easily carried out)

## ■ Others

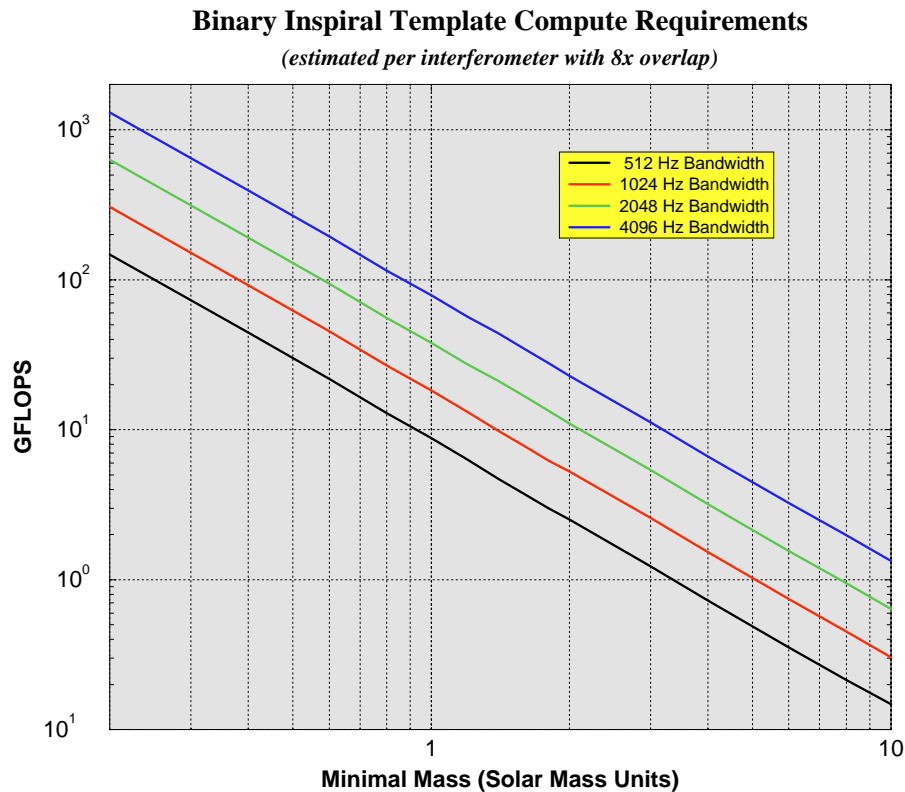
- ⑥ black hole ring-downs
- ⑦ black hole mergers
- ⑧ serendipitous discoveries!

# LDAS Data Flow Model

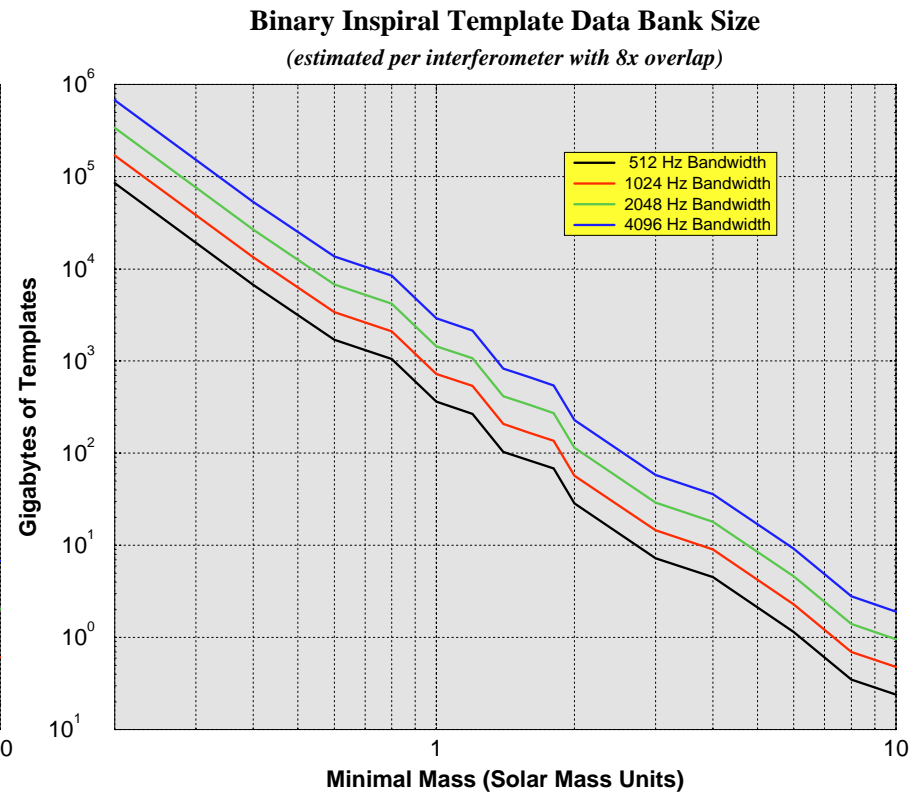


# LDAS Binary Inspiral Search "Optimal Filtering Demands"

## ■ Computation:



## ■ Templates:



# LDAS Hardware Design

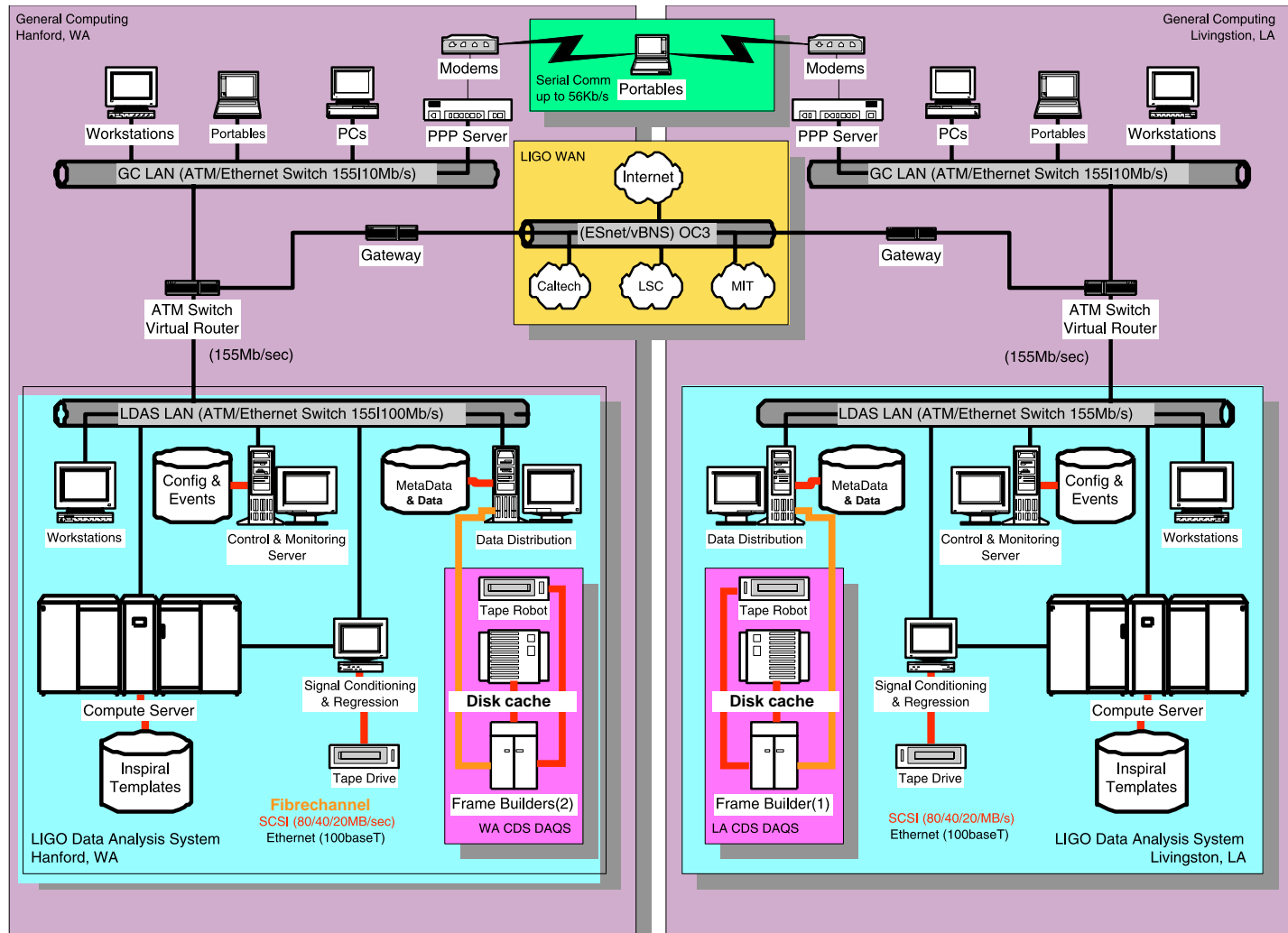
## ■ Requirements:

- 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*

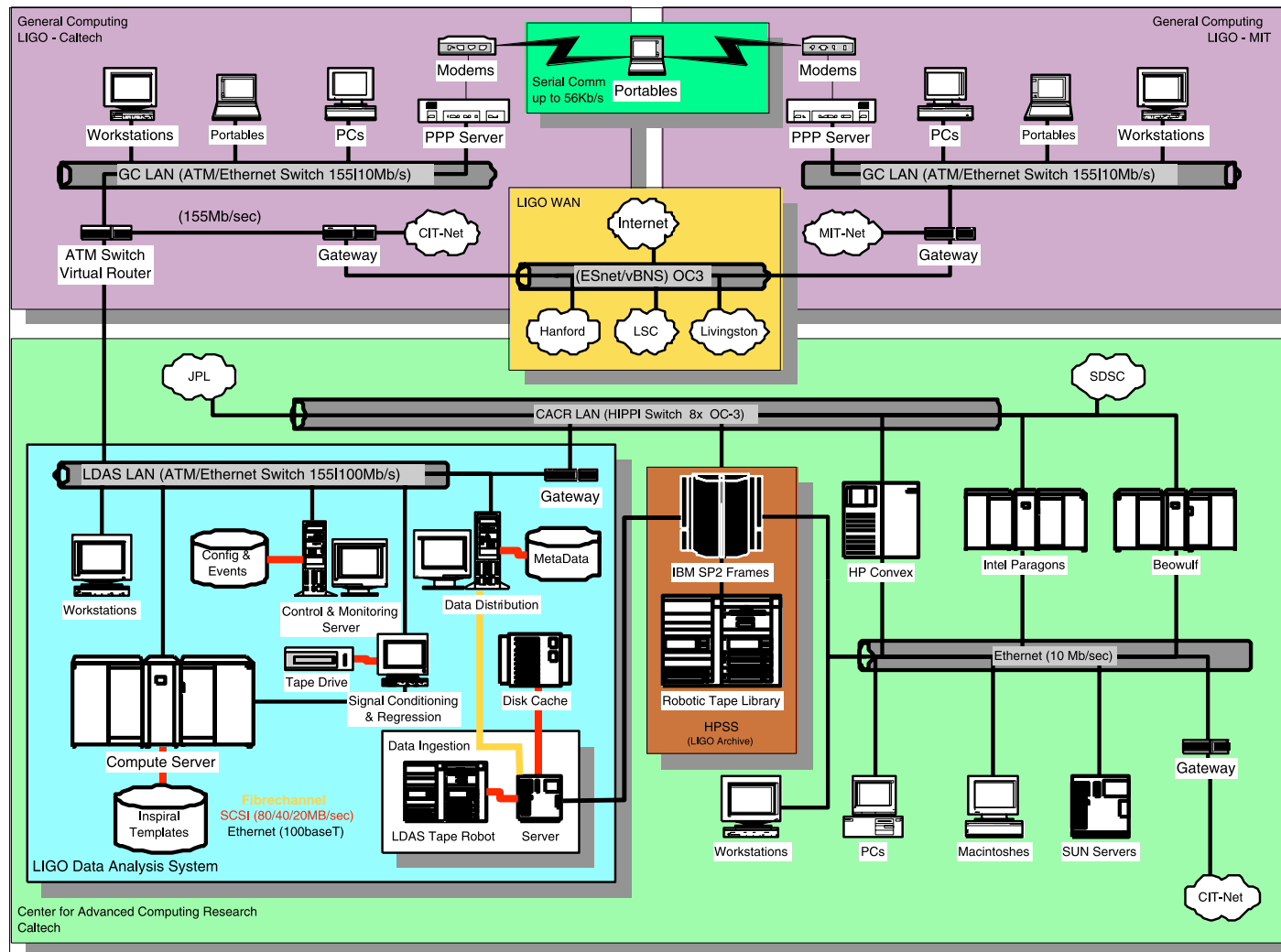
## ■ Components:

- ① *Beowulf (PC 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 ESNets and vBNS connectivity between the Observatories & Archive*

# LDAS On-Line Hardware Architecture



# LDAS Off-Line Hardware Architecture





# LDAS Software Design

## ■ Requirements:

- ① *Portability* -
  - ⇒ POSIX
  - ⇒ ANSI C/C++
  - ⇒ TCL/TK
  - ⇒ MPI for parallel computing
  - ⇒ ODBC for Databases
- ② *Extensibility* -
  - ⇒ Modular/Reusable Code
  - ⇒ OOP Design
- ③ *Maintainability* -
  - ⇒ Expressly Coded w/ OO Languages when possible
  - ⇒ CVS Source Code Management
- ④ *Flexibility* -
  - ⇒ Class 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 to OS
- ③ *API's* -
  - ⇒ High Level Supervisors to Data I/O
  - ⇒ Control, Monitor, and Management
  - ⇒ MPI Level Communications
  - ⇒ Filtering and Analysis
- ④ *UI's* -
  - ⇒ TCL/TK (Wish Shell) GUI
  - ⇒ TCL Interpreter Command Line
  - ⇒ TCLet Plug-ins for Web Browsers

# LDAS Data Formats

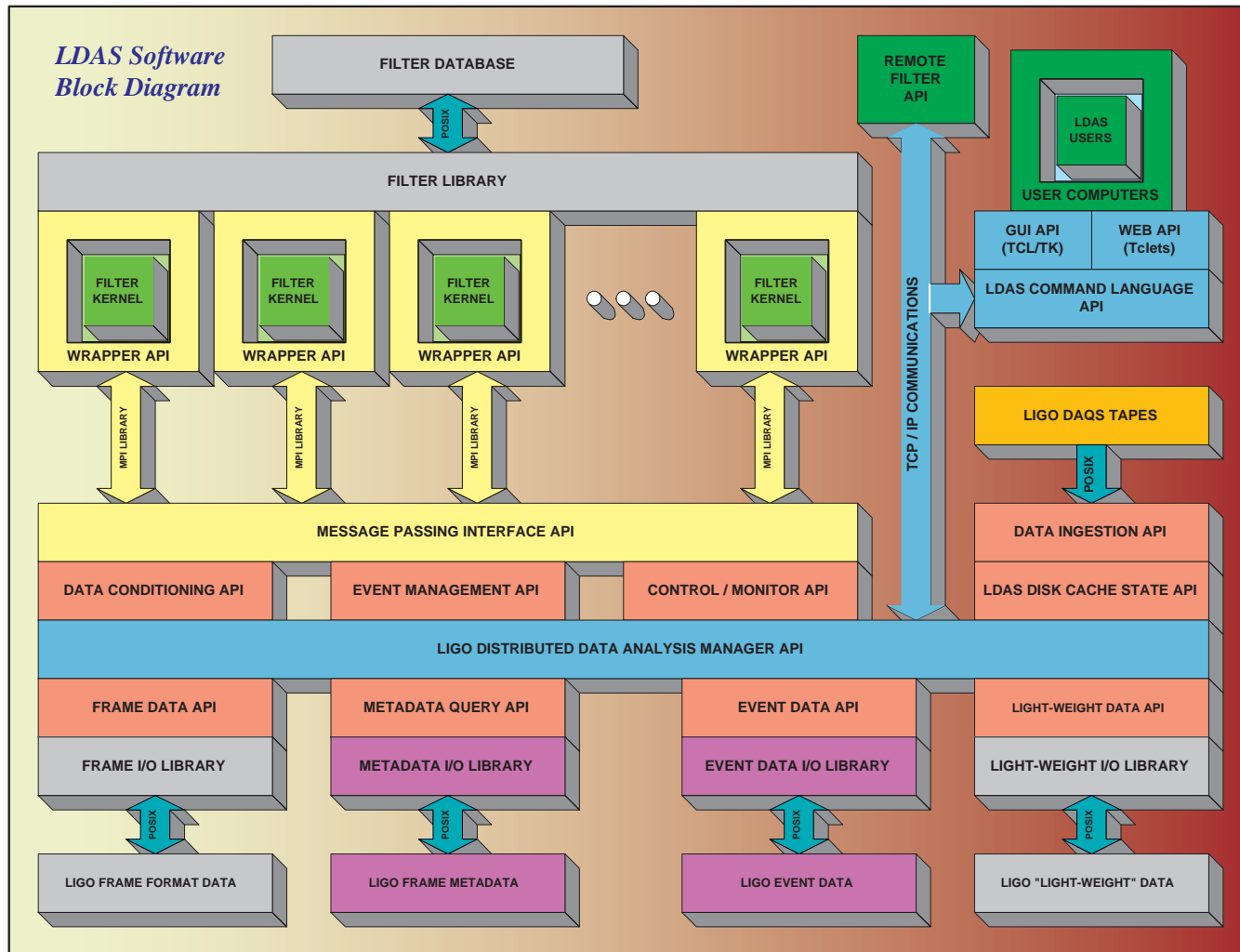
## ■ Frame Format:

- 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 Format:

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

# LDAS Software Block Architecture



# LDAS Layered API Design

## LDAS API's:

### ① Two Layers:

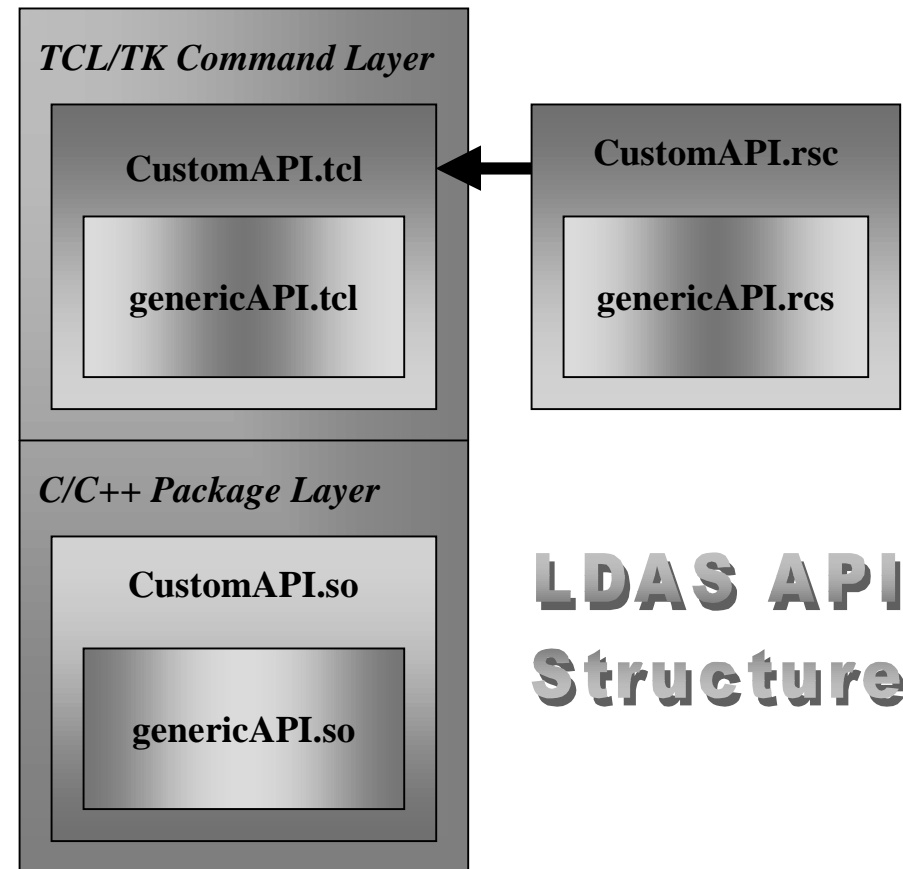
- ⇒ TCL/TK
- ⇒ C/C++ (*extends TCL Language*)
- ⇒ SWIG Unifies Layers

### 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 Light-Weight Data Management*
  - ✓ *Class Save & Restore*

### ③ Custom (*specialization*) Module:

- ⇒ I/O with local file system
- ⇒ Algorithms, filters, etc.

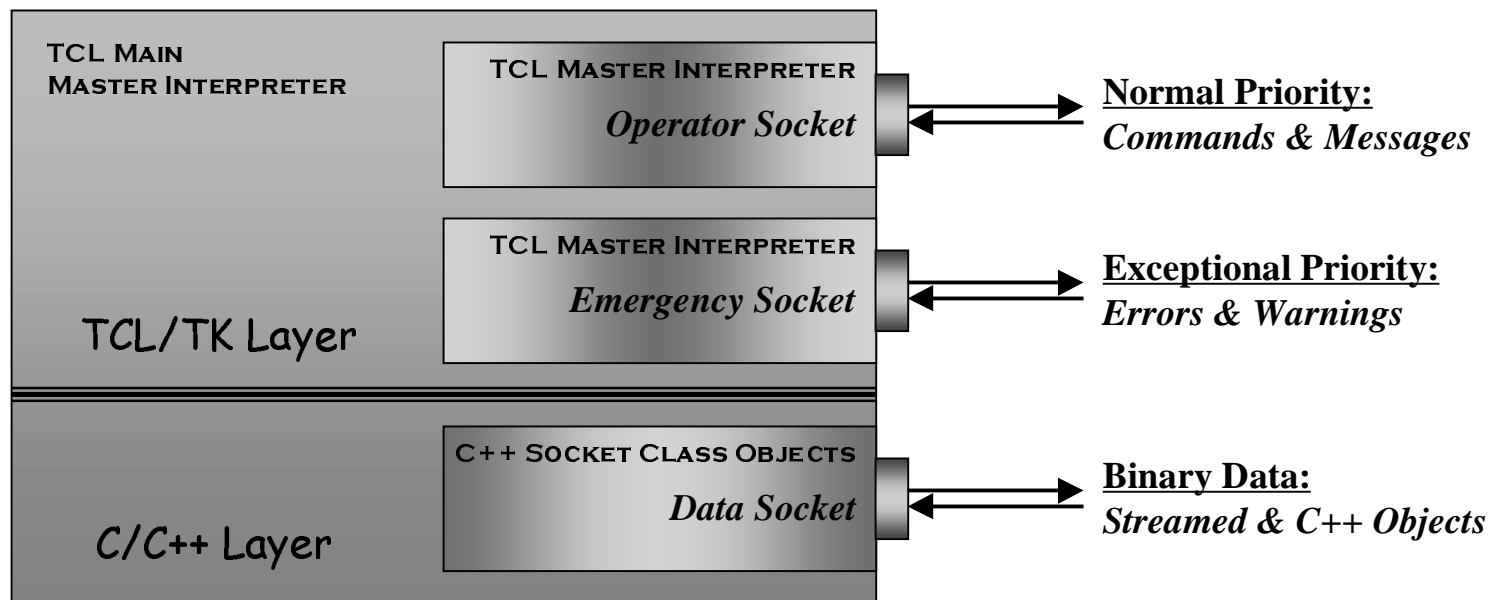


**LDAS API  
Structure**

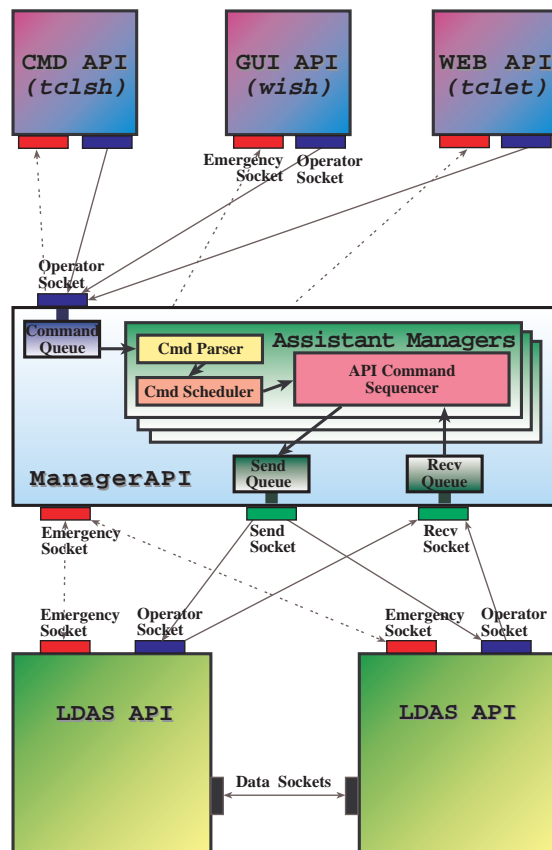
# LDAS API Communications

## ■ 3 Distinct 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



# LDAS Distributed API Supervisor



## ■ Manager API:

### ① Two Supervisory Layers:

- ⇒ Single "top-level" Manager
- ⇒ Multiple (3-10) Assistant Managers

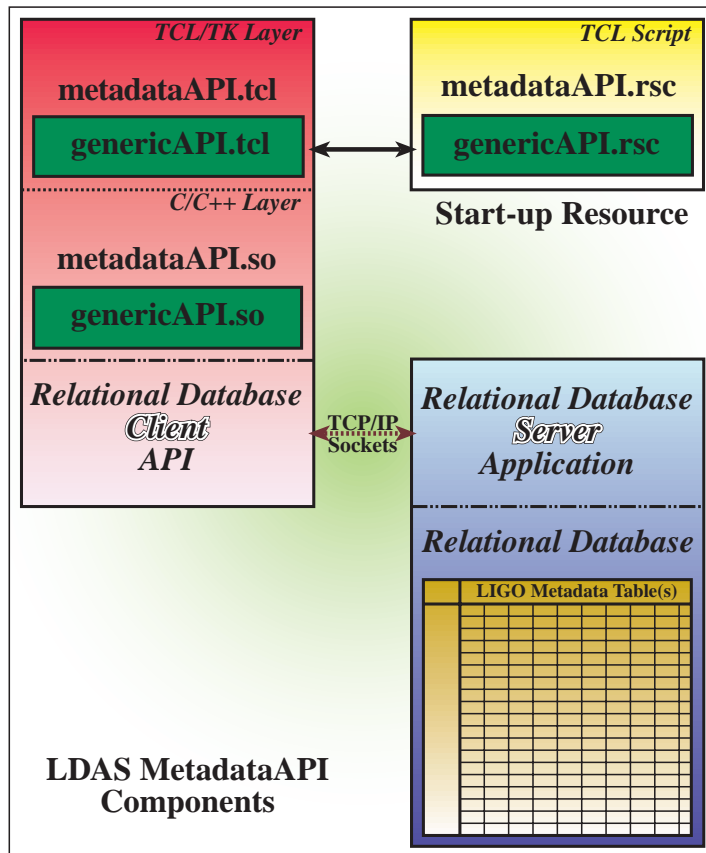
### ② Manager Layer:

- ⇒ 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*
- ⇒ Status Reports to Manager Layer

# LDAS Distributed Database API Components



## ■ Metadata API:

### ① Relational Database:

- ⇒ Using ODBC Standard for calls
- ⇒ Currently developing with DB2
- ⇒ Server can be Unix or Windows NT based

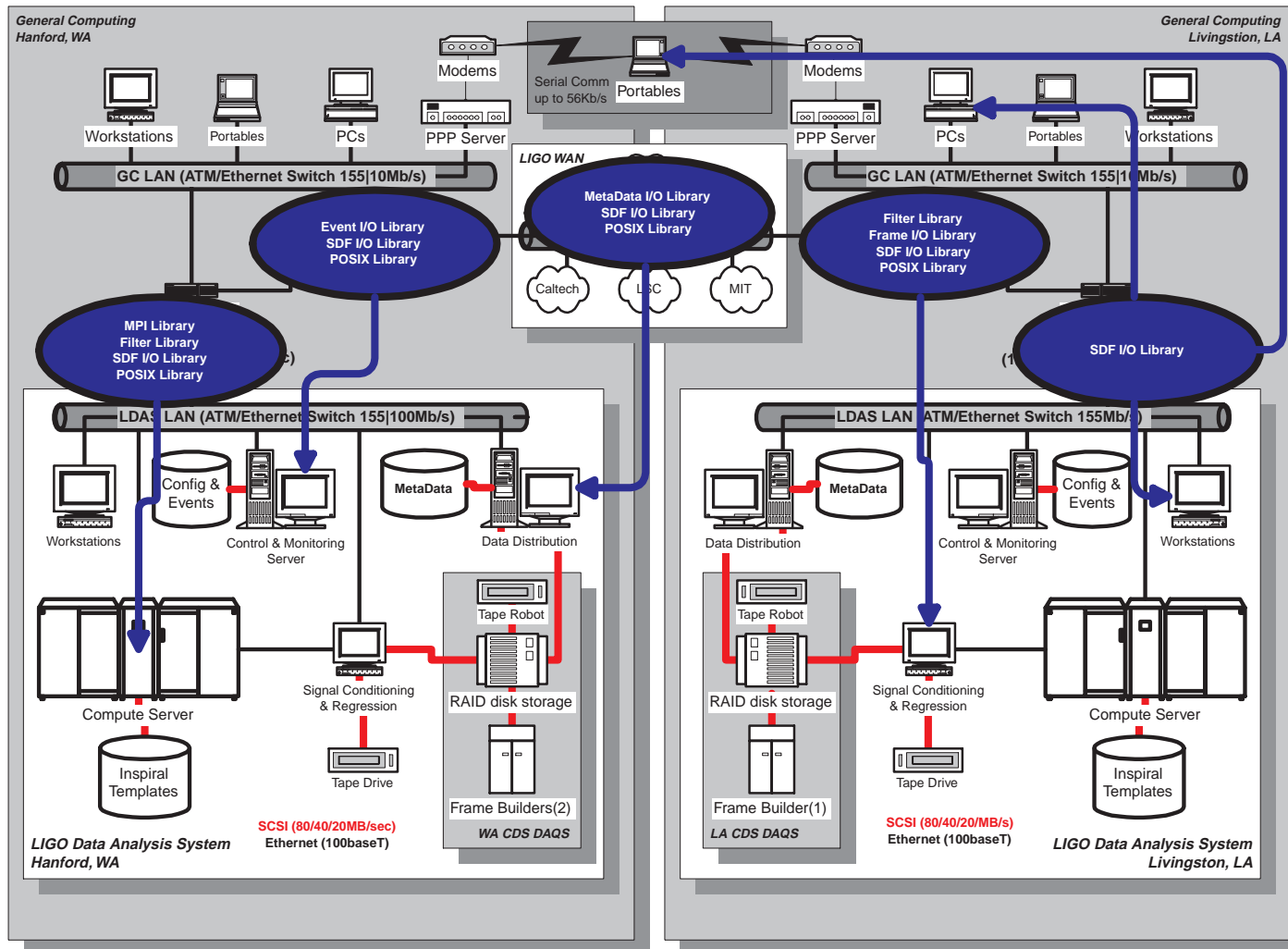
### ② Table Contents:

- ⇒ Frame Characterization
  - ✓ *Descriptors*
  - ✓ *Simple Statistical Summary*
  - ✓ *Diagnostic Reports (triggers)*
- ⇒ Event Characterization:
  - ✓ *Filter Results*
  - ✓ *Processing Information*
  - ✓ *Astrophysical parameters*

### ③ Metadata Query Services

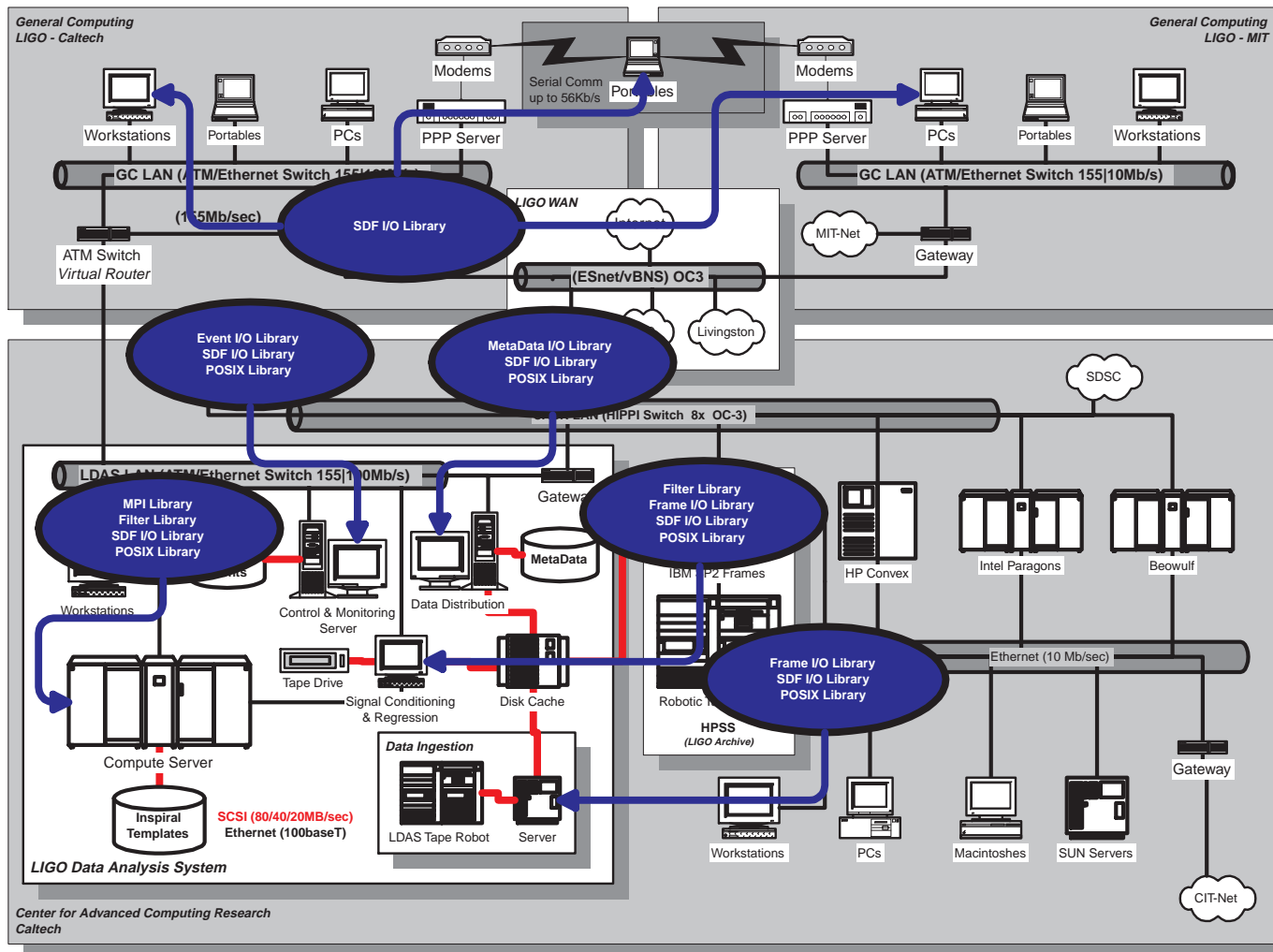
- ⇒ LDAS Processing Queries
- ⇒ User Data-Mining Queries

# LDAS Software/Hardware On-Line Library Map

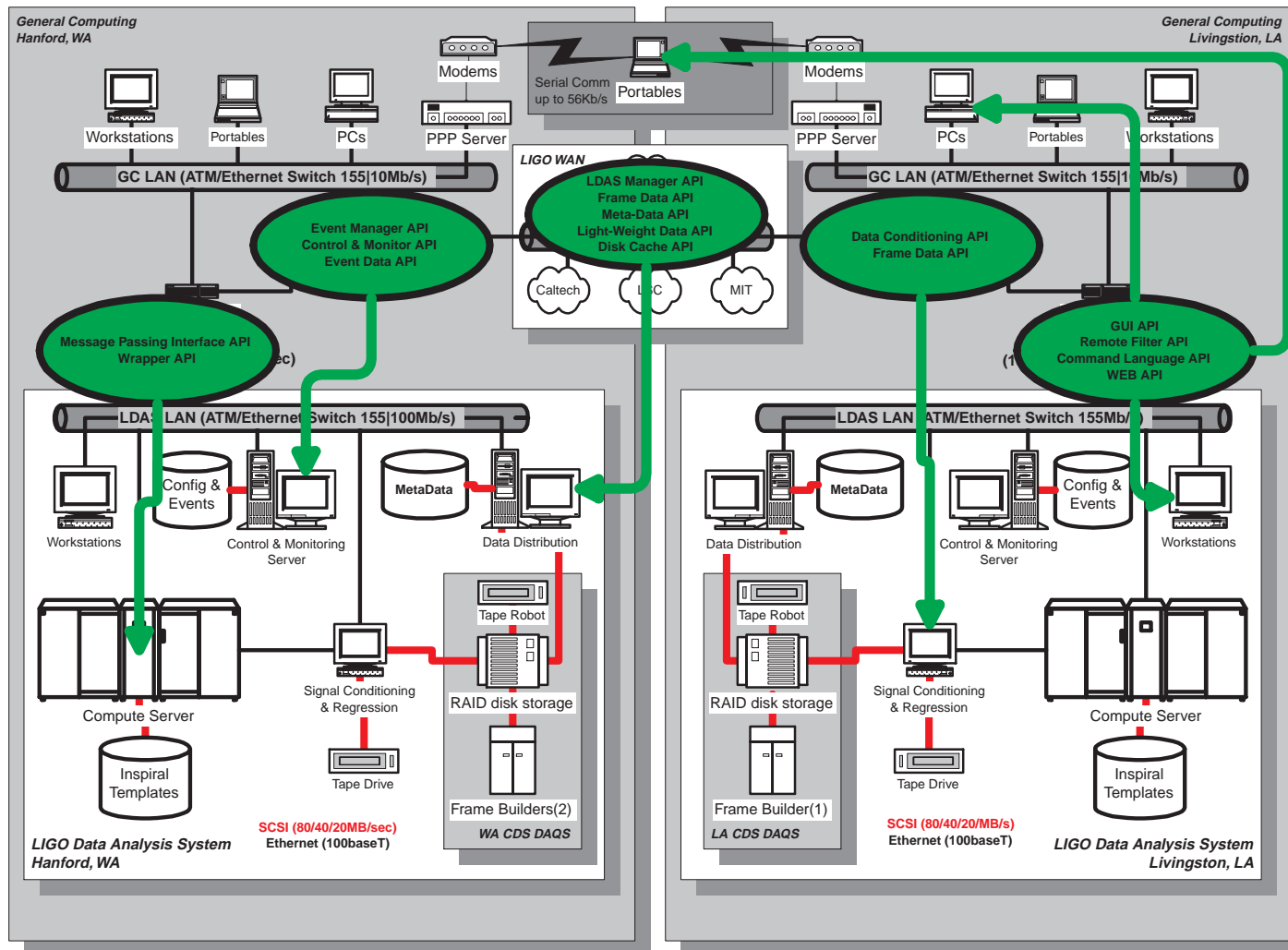




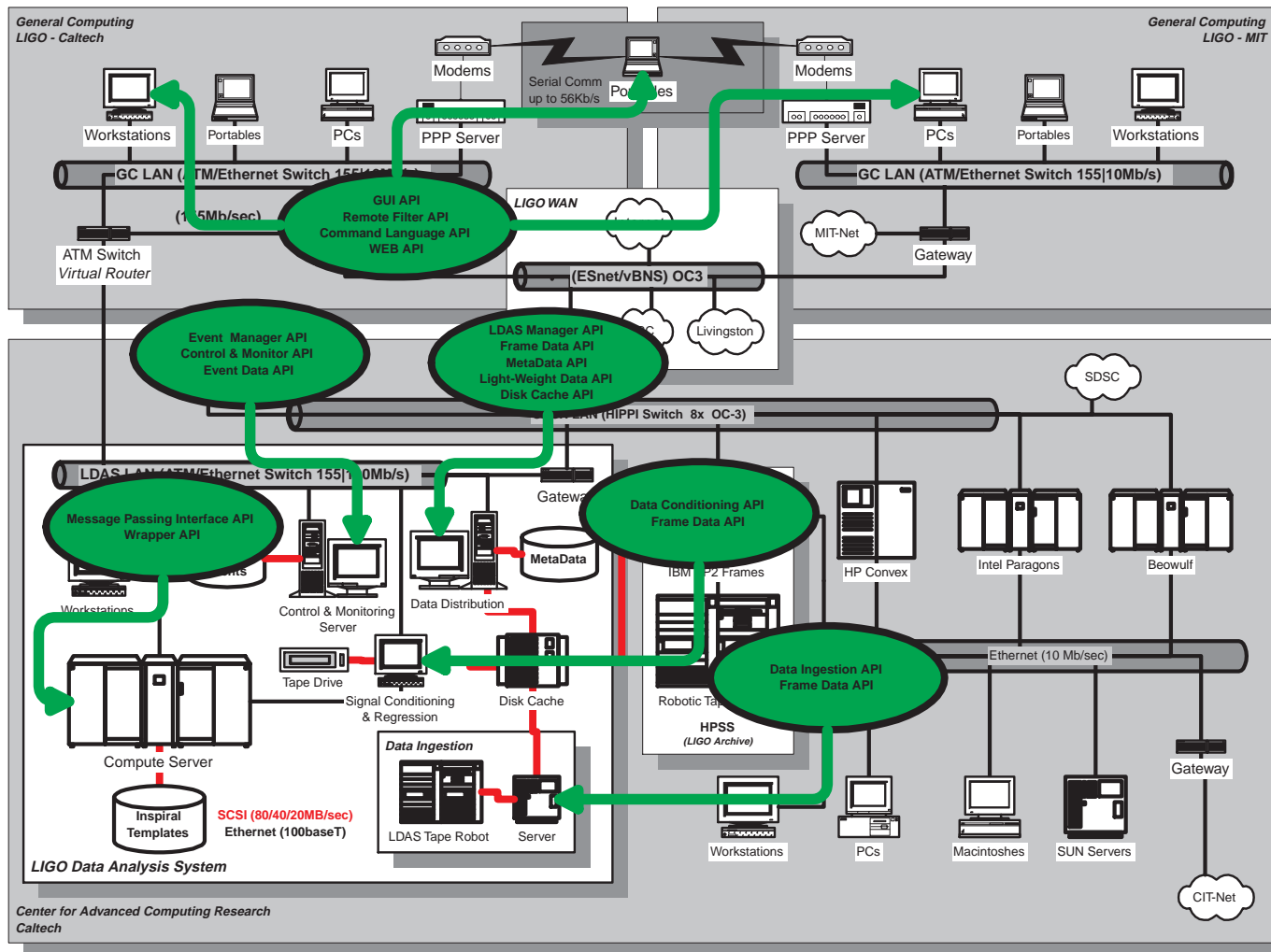
# LDAS Software/Hardware Off-Line Library Map



# LDAS Software/Hardware On-Line API Map



# LDAS Software/Hardware Off-Line API Map



# LDAS Software/Hardware Status Summary

## ■ Hardware Design Complete:

- ① Wide Area Network established to Sites
- ② Accepted delivery of "canned" Beowulf System
  - ⇒ Alta Technology's Altacluster 16 Pentium CPU w/ Extreme Linux
- ③ Data Distribution Server Hardware Identified

## ■ Software Design Complete:

- ① GenericAPI component developed
- ② FrameAPI, ManagerAPI being implemented
- ③ Expect to add DataConditioningAPI in time for Site Support
- ④ Myriad of Proto-typing activities underway
  - ⇒ Web-based Data Distribution
  - ⇒ Data-flow for Periodic Searches
  - ⇒ Frame-to-XML Translation & Viewing
- ⑤ MetadataAPI & EventdataAPI components based on ODBC using IBM's DB2!