



# Star (Traditional) Database Tasks & MySQL

---

1. Database Types & Operation Issues
2. Server & Database deployments
3. Tools with MySQL
4. Data definition
5. Star Database C++ API
6. Schema Evolution
7. XML



# Star Database Types

---

## Online:

Subsystem Conditions -> experimental monitoring

Subsystem Configurations -> detector run settings

RunLog -> run summaries

Event TagDb -> event level attributes

## Offline:

Calibrations

Geometry

Production, QA, Run-parameters

Event TagDb



# Operational Considerations

---

## Data Volumes:

Small -to- (*moderately*) Large

- RunLog: < ~Mbyte/day
- Conditions: ~10<sup>2</sup> Mbytes/day

## Updates:

frequent & real-time -to- prepared updates

- Conditions: ~updates/min + alarm generated
- Calibrations: ~production driven & previewed

## Record Types:

time-stamped & "versioning" protocols

- Conditions: ~essentially time-stamped
- Configurations: time-stamp + Catalog (versions)
- Calibrations: time-stamp + overwrite (versions)



# Server & Database Deployments

---

## STAR

- Online must be separable from Offline
- Online/Offline separate at the DB-Types
- Write responsibilities separate at the subsystem (domain) level

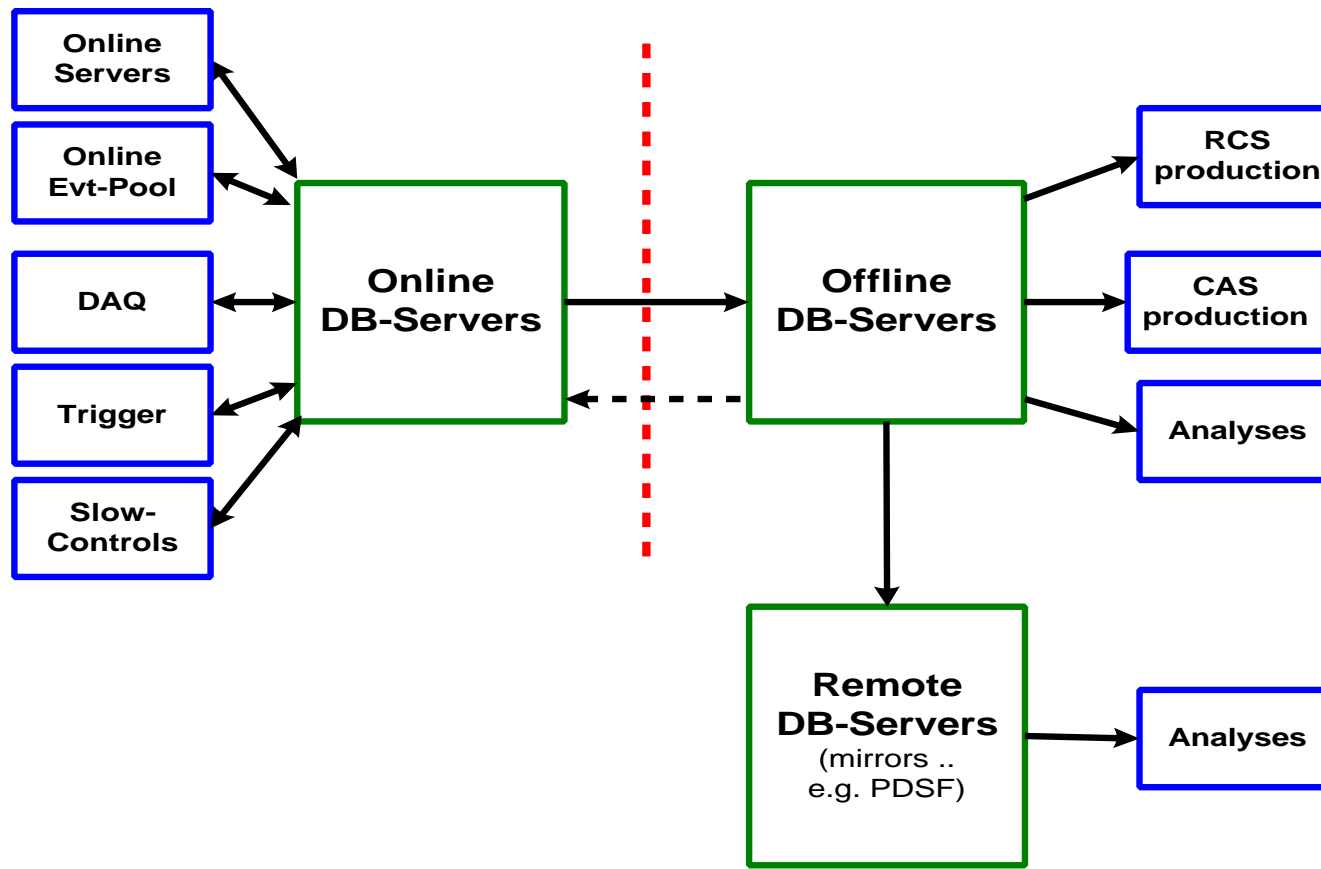
## MySQL

- A Server exists on a machine+port number
- A Server controls access to many DBs
- Write Access control is natural at the database level



# Server Deployment

4 Online Servers - 2 Primary & 2 development  
2 Offline Servers - 1 Primary & 1 Mirror





# Database Designations

---

## Database Types:

Conditions, Configurations, Calibrations,...

## Star Domains:

DAQ, Trigger, TPC, ...

## Star Database Names:

"Type\_domain"

Configurations\_trg, Conditions\_tpc,  
Calibrations\_emc,....



Online +  
Offline

## dbServers.xml Format

---

### Default Server Specification

```
<StDbServer>  
<server> duvall </server>  
<host> duvall.star.bnl.gov </host>  
<port> 3306 </port>  
<socket> /tmp/mysql.sock </socket>  
</StDbServer>
```

### Server By DataBase Name

```
<StDbServer>  
<server> db1 </server>  
<host> db1.star.bnl.gov </host>  
<port> 3306 </port>  
<socket> /tmp/mysql.sock </socket>  
<databases> Calibrations_tpc, Calibrations_emc </databases>  
</StDbServer>
```



# Tools used by STAR with MySQL

---

## php web tool

Very nice web administration tool  
Web-browser development (Sasha)

## Perl-API

Administration scripting  
Web-browsers

## Java-API

Online GUI: Run-Control & detector Configurations  
Strip-chart plotting tool for experimental monitoring

## C-API

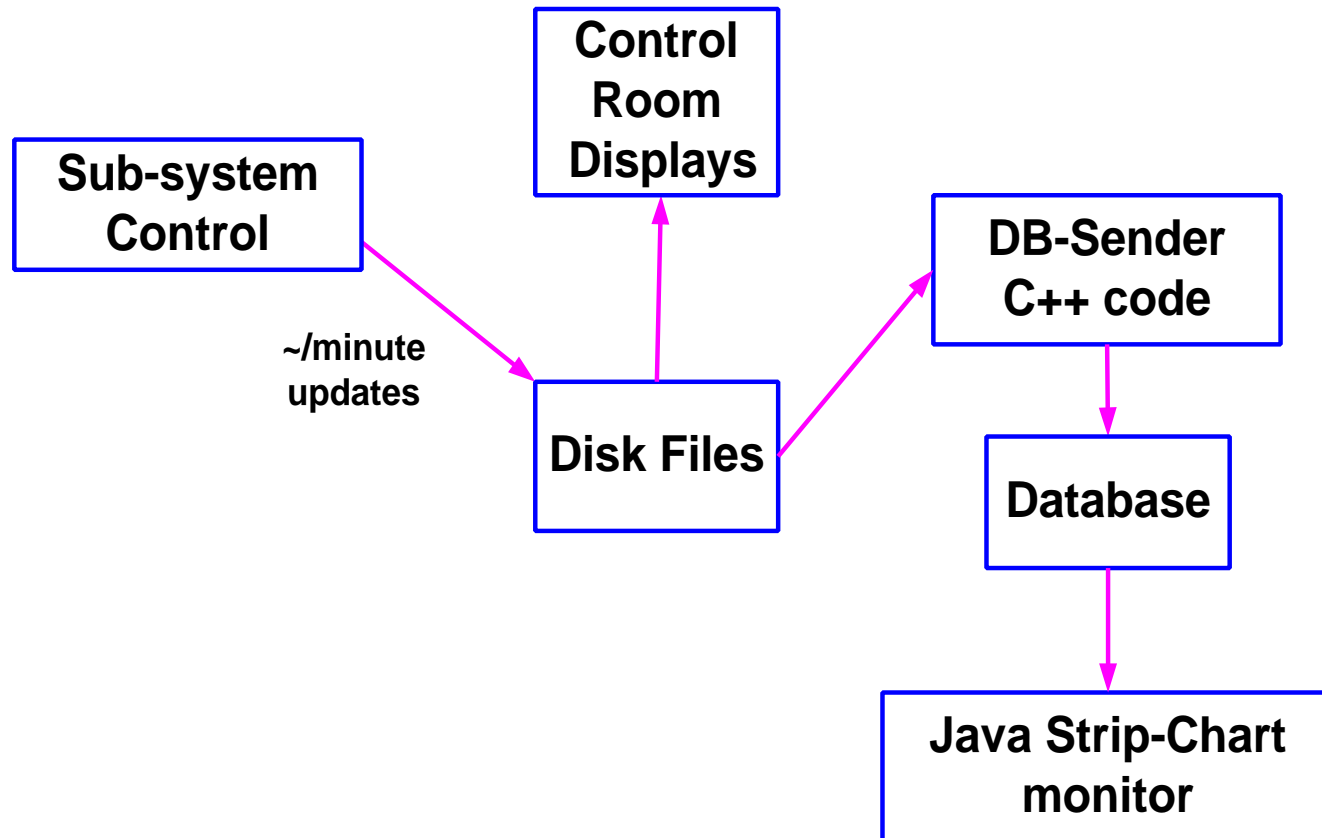
Low-level foundation of general Star C++ API





# Condition/DBs Processes

---





# Administration Tools

---

## Predominantly Perl Based

Define schema, storage structure,  
named-references, hierarchical structures  
& dump same to ASCII (XML)

Other perl scripts:

- replicate to mirrors

- perform backups & restores

- Log modifications



## Data Definition

---

### I/O protocol based on tables, c-structs

c-structs (\*.h files) are automatically translated into more descriptive XML files. Schema (XML) are stored in specified database

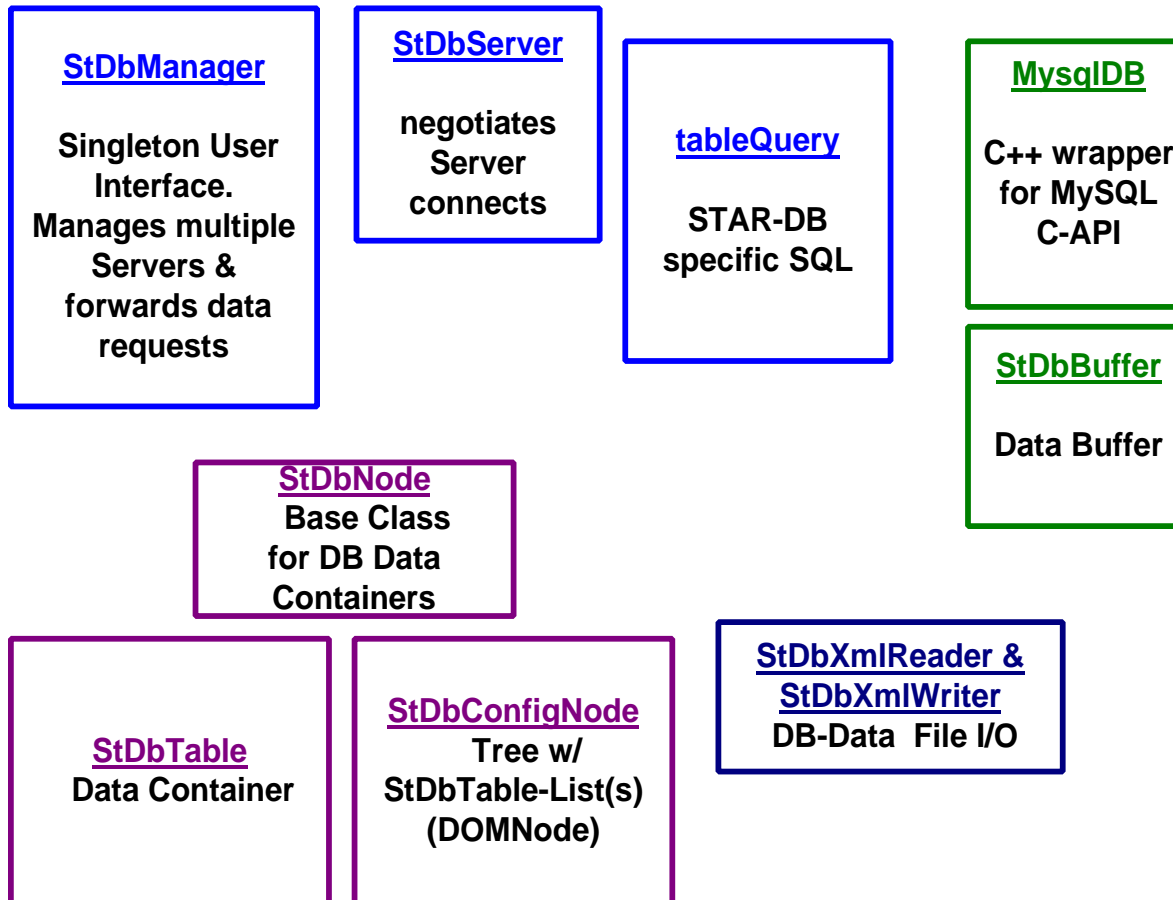
Database can be queried for c-struct schema (re)producing, \*.h, \*.idl, \*.xml

Database returns tables of data via C++ API



# Database C++ API Classes

---

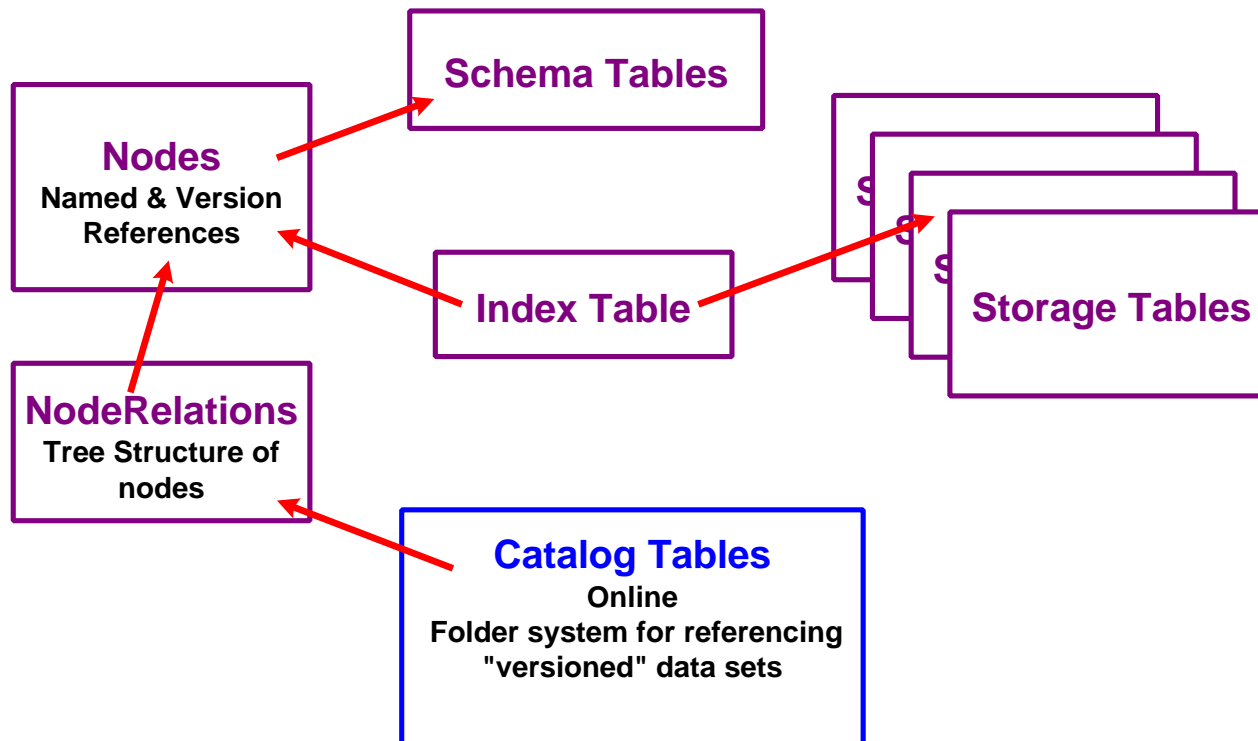




# Basic Table Set

---

General use DB-API hides SQL & requires  
~all databases to have the same structure





# Schema Evolution

---

**Database keeps information about schema. Default schema is last one loaded. Others can be requested by,**

- compiled table descriptor
- XML descriptor
- database schema ID



## XML

---

**Allows development of a flexible ASCII description of database information (both data and structural).**

Available Tools for XML support match those STAR uses with MySQL.

Perl-XML parser:

XML -> HTML translator

Java-XML parser:

XML -> Web & Online GUI

C++-XML parser:

XML -> DOMNode structure