

Use of Databases for real-time data

Chris Webster
NCAR / EOL

Background

- Previous real-time display used shared memory.
- Multiple copies of program X-displayed out.
- Difficult for other users/programs to have access.

Access to Data - onboard

- ASCII data feeds of scalar time-series
 - Network UDP broadcast
 - Serial feed; Digi SP-One (converts UDP to RS232)
- SQL Database/repository
 - Network readable and writeable by anyone
 - Permissions control
 - Easy to use and very common
 - Numerous language API's

PostgreSQL vs. MySQL

- PostgreSQL supports:
 - Arrays
 - Listen/notify mechanism
 - PostGIS extension.

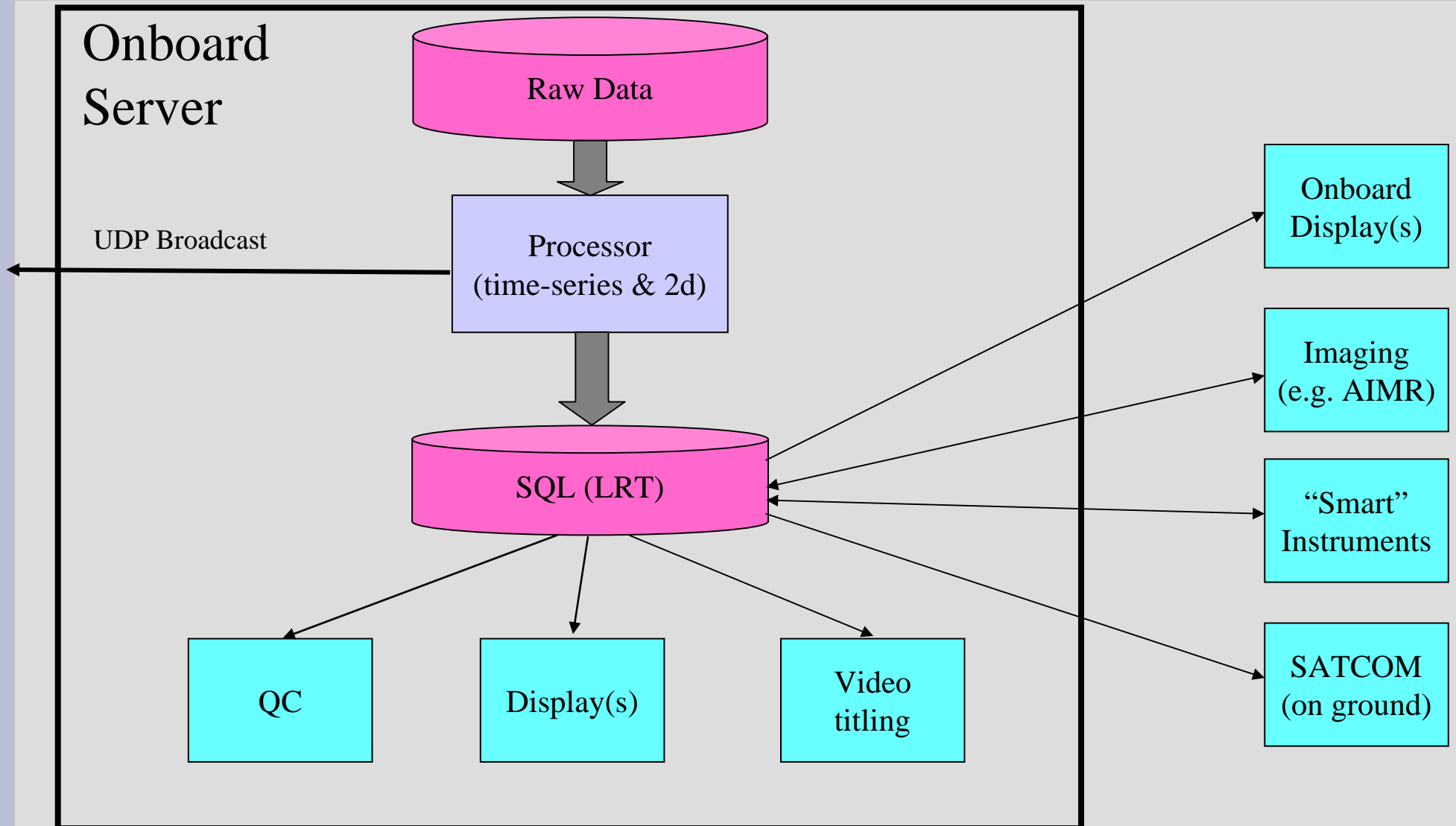
Table Layout

global_attributes

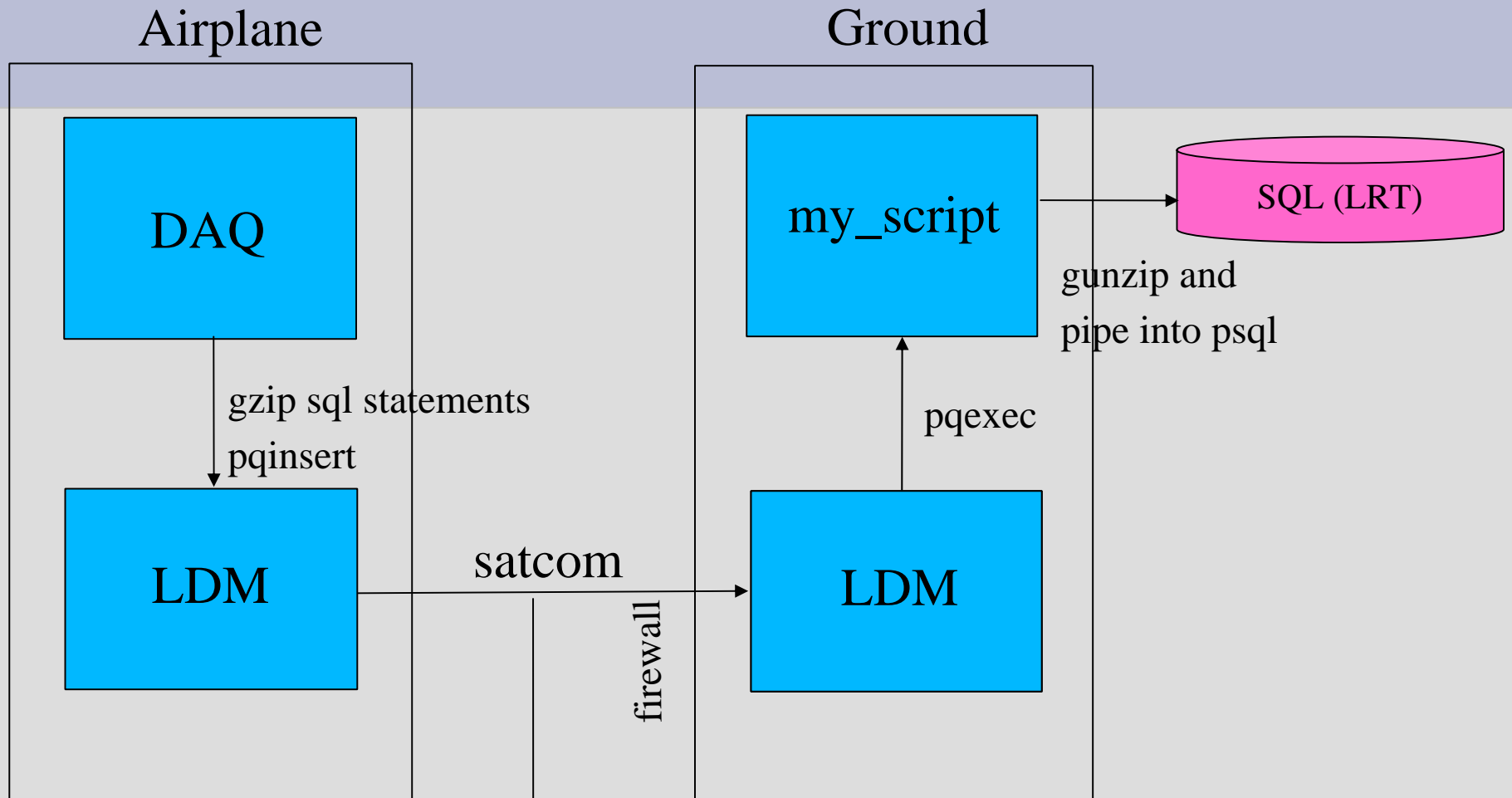
variable_list
and attributes

low_rate_table
Spreadsheet style
Typically 300 columns/variables
Ten hour flight will have 36,000 rows

Display System Architecture



Using Unidata's LDM to transfer files.



MPDS – ppoe – \$0.057/Kbyte - ~\$200/hour

or

Swift 64kbps – eth - \$10/min

Miscellaneous Notes

- Push/write 12,000-20,000 doubles per second.
- “ANALYZE” command needs to be run periodically.
 - Can cause a pause in database transactions
 - longer the bigger.
 - 2-30 seconds.
- Have not done image or non-standard data yet.
- Current databases are deleted per flight.

Pro's

- Network read/write
- FOTS Component
- Ubiquitous
- Really easy to use
 - “SELECT lat,lon,alt FROM raf_Irt where TAS > 20.0”
- Listen/Notify mechanism for real-time applications.
- Transfers in ASCII (platform independent).

Con's

- Can't “copy” a file and work offline.
- Transfers in ASCII (not compact).

Future

- We will continue to utilize and expand our database capabilities.
- User instruments writing directly into database
- Possible expansion:
 - Short and medium term storage and access of data.
 - Scan and Image data.