Supporting libraries for image processing

- Multiple libraries exists
  - cfitsio, WCS, ..
- They are not integrated to RTS2
  - .. nor they are (with cfitsio exception) provided as debian packages!
- Image processing tools..
  - ...IRAF, Midas, HEADAS, Root...
- ..are not libraries!
  - ..and are hard to install and maintain
  - usually hard to interface / reuse

RTS2 image library

- Rts2Image and child classes
- Based on cfitsio
  - Would like to escape this dependency (performance issues,..) and most probably use own implementation of FITS access
- Currently mainly FITS headers keywords routines

(My) goals

- Extend Rts2Image class that it will provide all operations needed for image and data processing
- Integrate calibrations within RTS2
- Write, document and distribute command line tools (binaries, shells, ..) based on operations with Rts2Image to
  - Calibrate image (→ calibration database)
  - Extract light curves
  - Search for transients

Who & what

- Maintain and improve Rts2Image library
- Search, design, develop and maintain operations on Rts2Image classes
- .. or provide and maintain another library I can use for image processing
  - That includes full documentation, .deb/.rpms packages, ..

Avoid the VO problem!

- Virtual Observatory people usually produces lots of useless paperwork on how thinks shall work
- .. but they are not open to develop code or share developed code
  - So shall we learn to compete with them?
  - And hence prove that VO is useless waste of money?
- → I would prefer to base our discussions on something that works, rather then building Potemkin villages

Avoid VO problems - examples

- STC (Space-Time-Coordinate system)
  - So complex that there is no library to deal with it
  - Non-standard XML
    - Very hard to use parsers to get useful information
  - → everybody uses subset of STC and is happy
- VOTable
  - Another way how to express table in XML
  - Does not solve elements which must be present,..

Avoid VO problems

- We need to define, code and use libraries for image processing
- We need to distribute those including source code
    - So other people will be able to use the code
    - Source code can be bound to accepting special license conditions and can be available only for personal use
        - But I prefer completely open developing cycle

Avoid VO problem

- We shall learn from VO documents
  - They are not so bad after all
- Significantly simplifies them
  - So they will fit only our cases
- And produce new simplified standards
  - Which we can make more complicated on the way
- Instead of top-bottom approach
  - We will try bottom-up one

(Dis) advantages of code sharing

- Distribution of work to participants
- Faster development
- More users, more bugs discovered
- More feature requests
- Feel that you are doing something other can use

- Same code, same bugs, same results
- Fear that somebody else can easily replicate your work and trash your time investment

→ we must deal some rules

- Who is interested to join
- Who will develop what
- Coordinate development and bug reporting
  - Sourceforge?
- So it will bring balanced benefits to all parties
  - Everybody will get something
  - Everybody will pay something