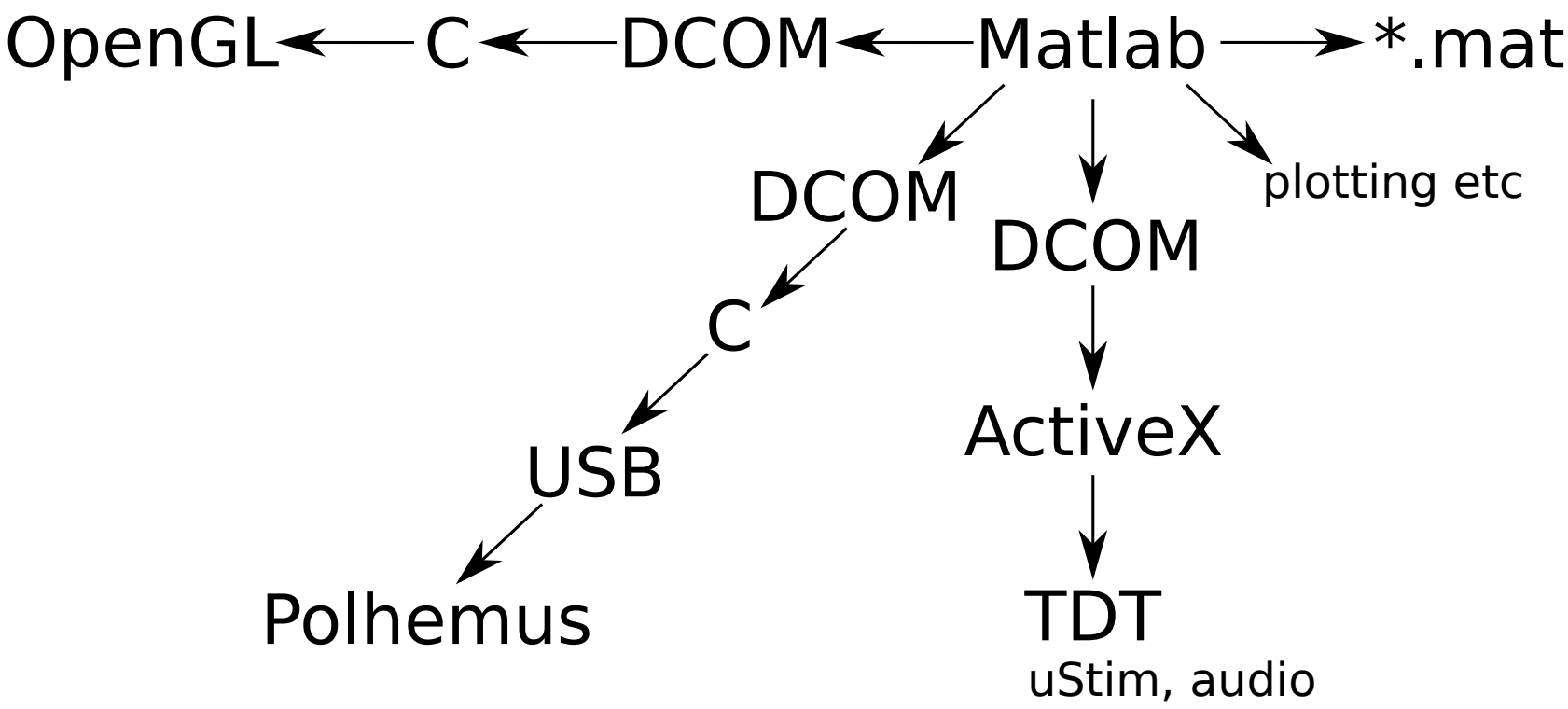
Sabes lab expermiental control software

UCSF

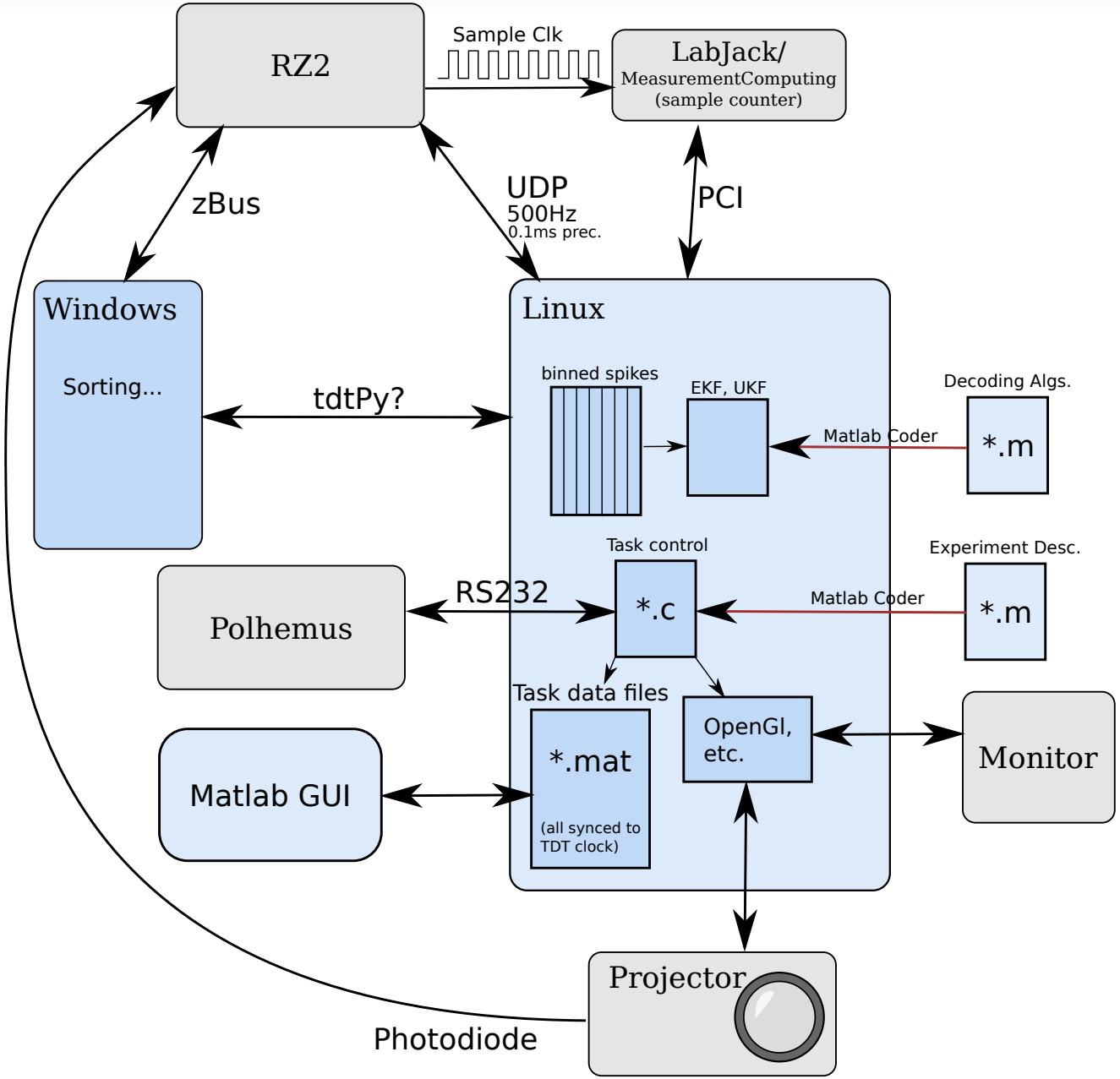
September 12 2012

To the best of my rather limited knowledge.

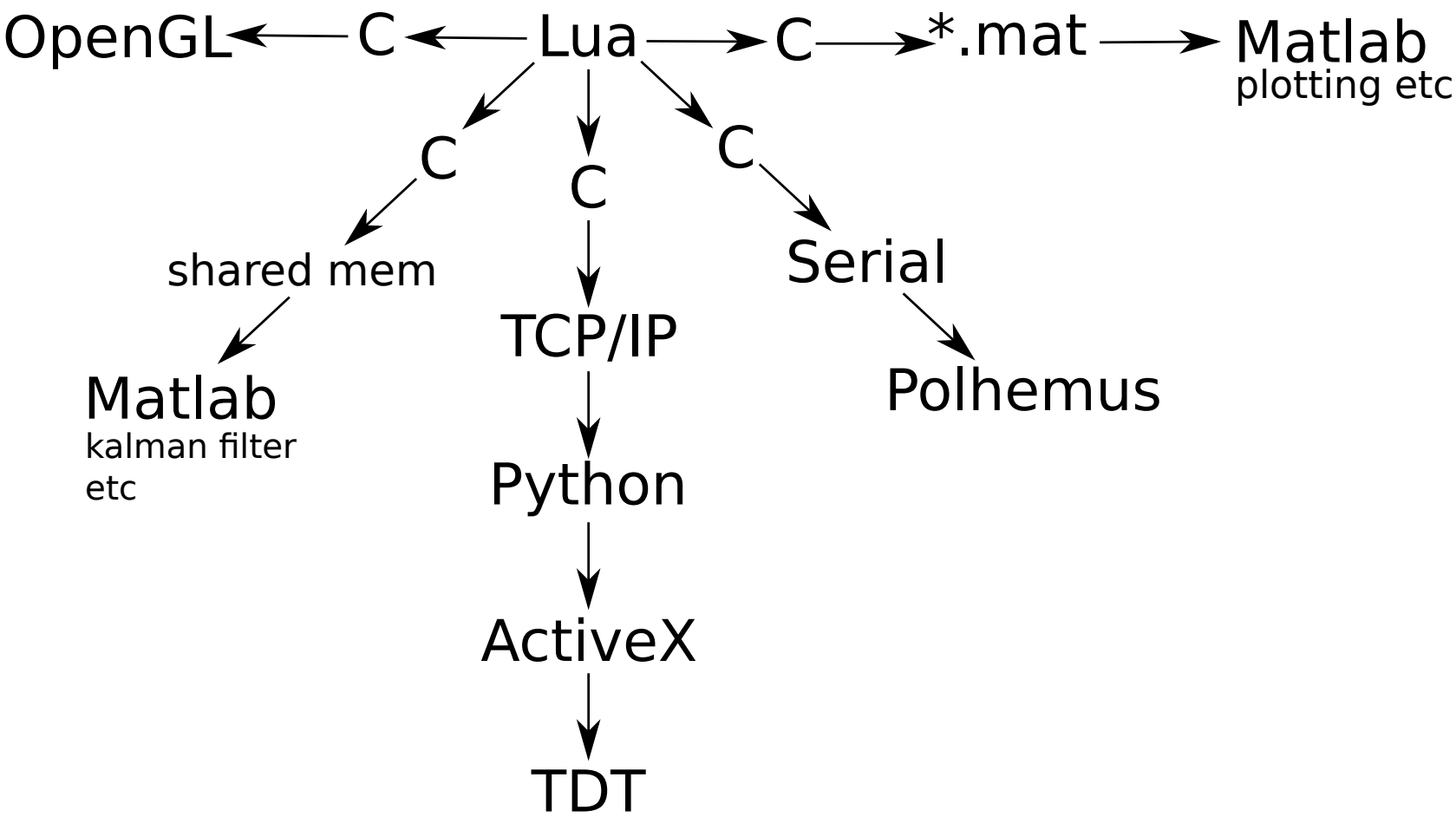
Present implementation:



Previous plan, August 15th 2012



New design:


OpenGL
C
Lua
C
*.mat
Matlab
plotting etc
C
C
C
shared mem
TCP/IP
Serial
Matlab
kalman filter
etc
Python
Polhemus
ActiveX
TDT

- Synchronization is an issue.
  - Count TDT sample clocks, tag all relevant events, and store.
  - Put the counter on the PCI bus so the transaction latency $\approx=$ clock period (40.96 μs)
  - USB bulk/control transaction latency 5.5 ms on Windows. [1] [2]

[1] `http://doc.utwente.nl/56344/1/Korver03adequacy.pdf`

[2] Lower for polled HID devices – 125Hz to 1kHz)

- Latency should be minimized.
  - Most of the latency is likely in the projector $\approx 16ms$ + Polhemus (3.5 ms)+ non-isochonous USB channel ($\approx$ 5.5 ms) + Windows networking latency ($\approx$ 0.5ms).
    - New 120Hz monitor and projector look *great*.
  - Lua *is* fast ($50\mu s$)... but is it worth the effort when most of the latency is elsewhere?

- It makes the most sense to use the native data structures & serialization of a given language.
  - Tables for Lua, *.mat for Matlab, binary files for C.
  - Could post-process lua tables into Matlab.
- Writing EKF / UKF / Wiener filters should be done in Matlab.
- Debugging embedded Lua is a bit of a PITA
  - Avoiding threads / continuations makes things easier.
- People in the lab already know Matlab ...