eyeris

Project Eyeris, May 13-20

TEAM

Will Bryan
Tyler Burnham
Scott Connell
Justin Derby
Kris Scott
Arjay Vander Velden

ADVISOR

Daji Qiao

CLIENT

Stephen Gilbert,VRAC

Client

- Virtual Reality Applications Center (VRAC)

Problem Statement

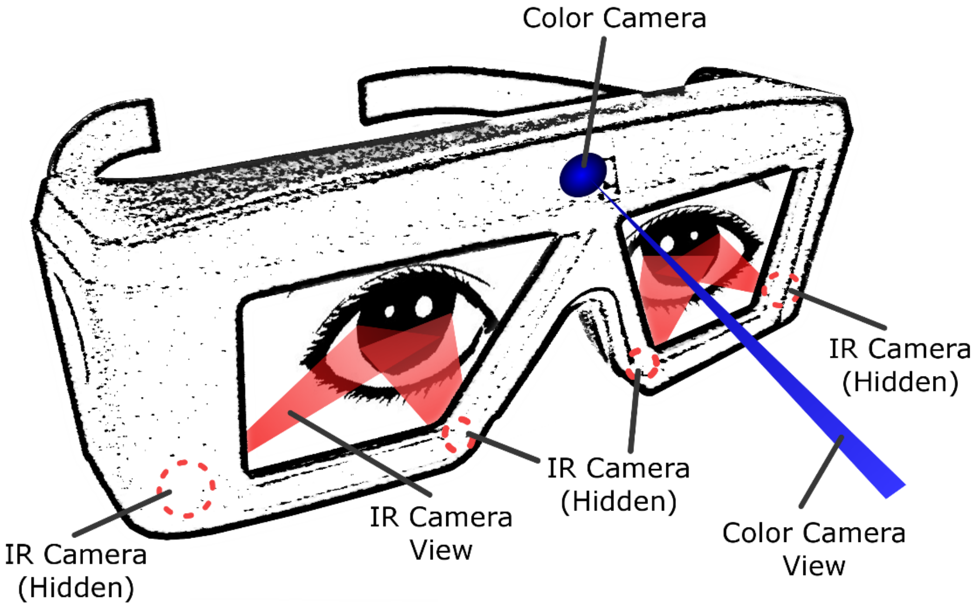
- No mobile solution tracks both eyes
- Need for real-time streaming
- Need solution to allow for viewers to analyze the data as the study is happening

System Description

Concept Sketch

- Embedded, real-time eye tracking system
- Stream outward video and eye tracking data wirelessly
- Will be used in the C6/MIRAGE for virtual reality applications

Design sketch-up



Functional Requirements

- Track both eyes for 3D depth
- Two-hour, onboard cache of world view video and eye data
- Real-time, wireless transmission of world view video
- Real-time, wireless transmission of eye tracking data
- Video and eye tracking data must be in-sync

Non-Functional Requirements

General Requirements

- Glasses will be active, stereoscopic glasses
- System must be unobstructive and nonintrusive
- Battery must last at least three hours

Non-Functional Requirements

Video Requirements

- World-view camera must be high definition (720p or 1080p)
- The sensor camera must be 640x480 at 30fps

Non-Functional Requirements

Physical Requirements

- Weight constraints: Glasses – 2.3 lbs.; Backpack – 5 lbs.
- Dimensional constraints: Glasses – no wider than 10 inches

Existing Products

Existing Market

Tobii ($45,000)

- No hardware adjustments necessary
- System guided calibration
- Lightweight & discrete
- Parallax compensation
- Millisecond data sync

Existing Market

SMI ($30,000)

- Real time & recording
- Audio

Project Justification

What can we improve upon?

- High resolution real-time streaming over Wi-Fi
- 3D vector tracking (adds depth to tracking location)
- Cost of the system

Implementation

Risks and Mitigations

Risk	Mitigation
IR exposure on an eye has not fully been tested.	Don’t allow direct IR exposure to the eye. Also pulse the IR LED instead of using a beam.
Synchronizing the cameras for both eyes with the outfacing camera doesn’t work.	Start working on the synchronization early and do prototyping.
Difficulty interfacing between the main board and the Gumstix boards.	Using well established communication protocol like UART.
Eye tracking algorithm does not work well with different sizes and shapes of eyes.	Use two cameras per eye to allow for 3D eye tracking and multiple video capture points.
Big project with a short schedule.	Make sure we are consistent with our schedule and make sure we try and actively stay ahead of it.
Implementing for different architectures and platforms (Windows, Linux, Mac).	Be conscious about what we program so that it can be compiled on multiple platforms and different hardware.

Resource and Cost Estimate

- **Resources**
  - 2 Gumstix Boards
  - 1 PandaBoard
  - 4 Inward-Facing Cameras
  - 1 Outward-Facing Camera
  - 1 Battery Pack
  - PCB
  - Wires
  - Stereo Glasses

- **Cost Estimate: Approx. ~$1,000**

Project Milestones

End of Fall semester:

- Finished Design Document
- Finished Project Plan
- Working 2D Eye-Tracking algorithm

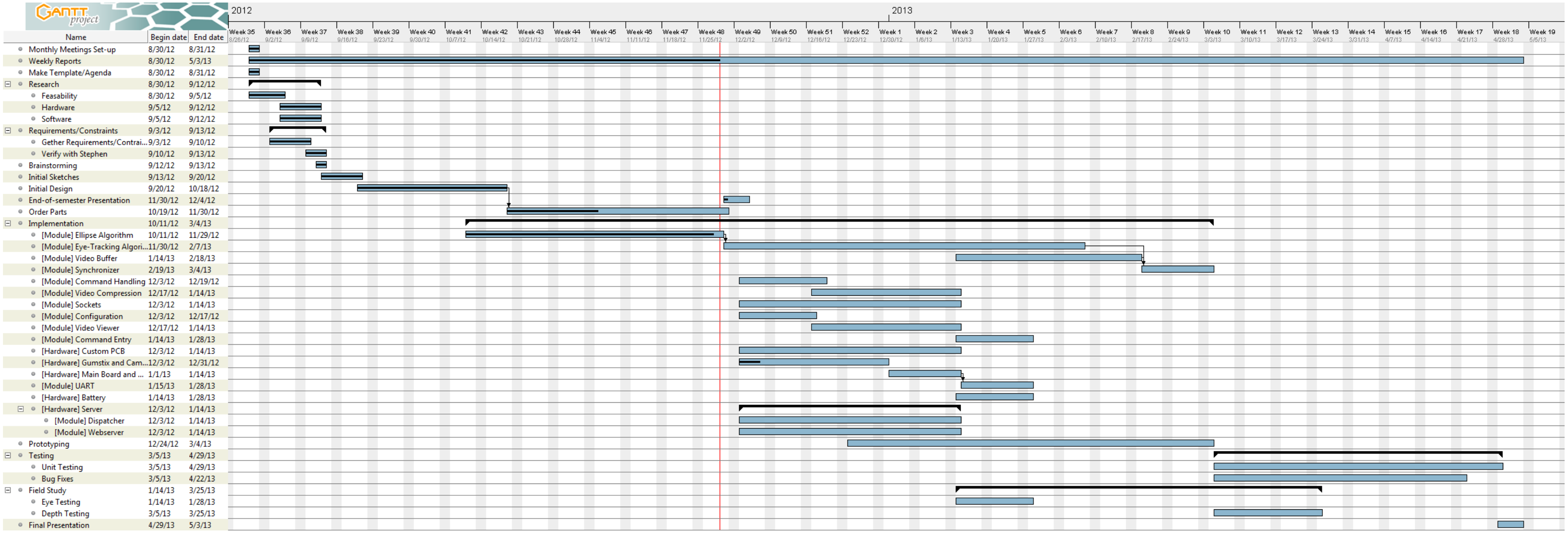
End of Winter break:

- ~50% of the modules implemented
- All hardware ordered

End of Spring semester:

- Fully working product

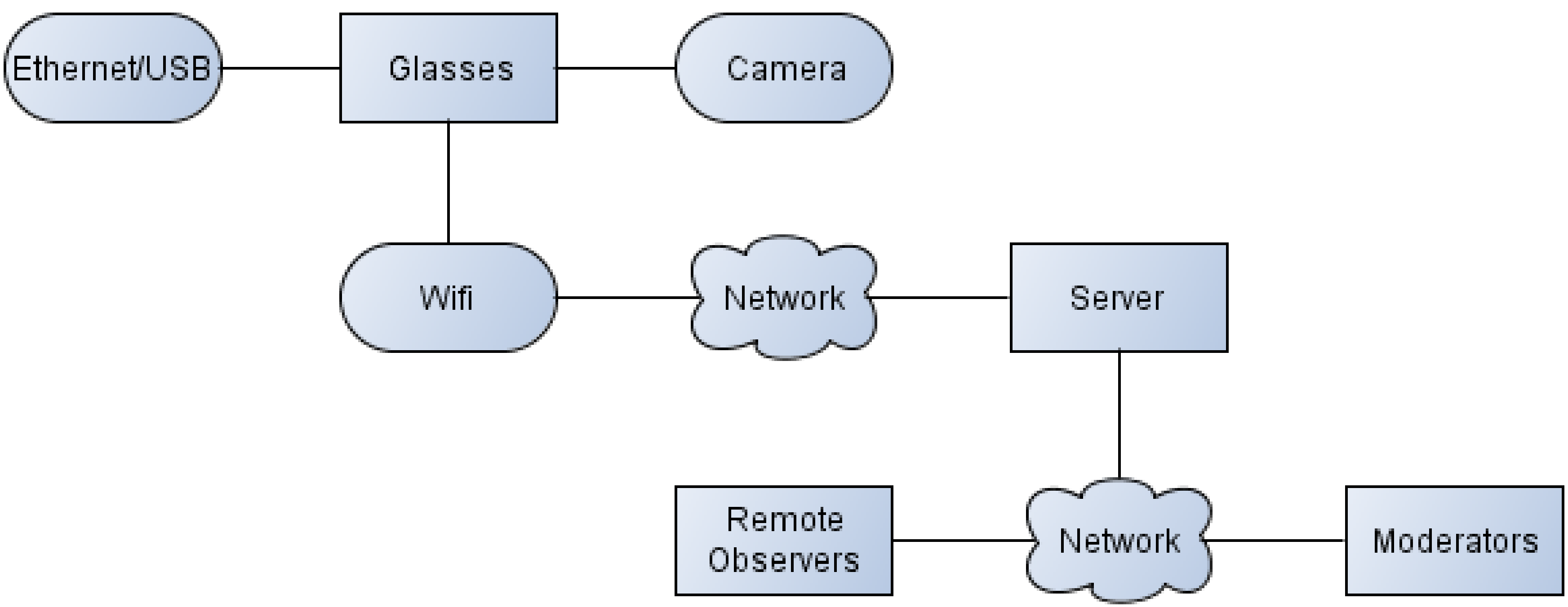
Project Schedule



Name	Begin date	End date
Monthly Meetings Set-up	8/30/12	8/31/12
Weekly Reports	8/30/12	5/3/13
Make Template/Agenda	8/30/12	8/31/12
⊟ Research	8/30/12	9/12/12
Feasability	8/30/12	9/5/12
Hardware	9/5/12	9/12/12
Software	9/5/12	9/12/12
⊟ Requirements/Constraints	9/3/12	9/13/12
Gether Requirements/Contrai...	9/3/12	9/10/12
Verify with Stephen	9/10/12	9/13/12
Brainstorming	9/12/12	9/13/12
Initial Sketches	9/13/12	9/20/12
Initial Design	9/20/12	10/18/12
End-of-semester Presentation	11/30/12	12/4/12
Order Parts	10/19/12	11/30/12
⊟ Implementation	10/11/12	3/4/13
[Module] Ellipse Algorithm	10/11/12	11/29/12
[Module] Eye-Tracking Algori...	11/30/12	2/7/13
[Module] Video Buffer	1/14/13	2/18/13
[Module] Synchronizer	2/19/13	3/4/13
[Module] Command Handling	12/3/12	12/19/12
[Module] Video Compression	12/17/12	1/14/13
[Module] Sockets	12/3/12	1/14/13
[Module] Configuration	12/3/12	12/17/12
[Module] Video Viewer	12/17/12	1/14/13
[Module] Command Entry	1/14/13	1/28/13
[Hardware] Custom PCB	12/3/12	1/14/13
[Hardware] Gumstix and Cam...	12/3/12	12/31/12
[Hardware] Main Board and ...	1/1/13	1/14/13
[Module] UART	1/15/13	1/28/13
[Hardware] Battery	1/14/13	1/28/13
⊟ [Hardware] Server	12/3/12	1/14/13
[Module] Dispatcher	12/3/12	1/14/13
[Module] Webserver	12/3/12	1/14/13
Prototyping	12/24/12	3/4/13
⊟ Testing	3/5/13	4/29/13
Unit Testing	3/5/13	4/29/13
Bug Fixes	3/5/13	4/22/13
⊟ Field Study	1/14/13	3/25/13
Eye Testing	1/14/13	1/28/13
Depth Testing	3/5/13	3/25/13
Final Presentation	4/29/13	5/3/13

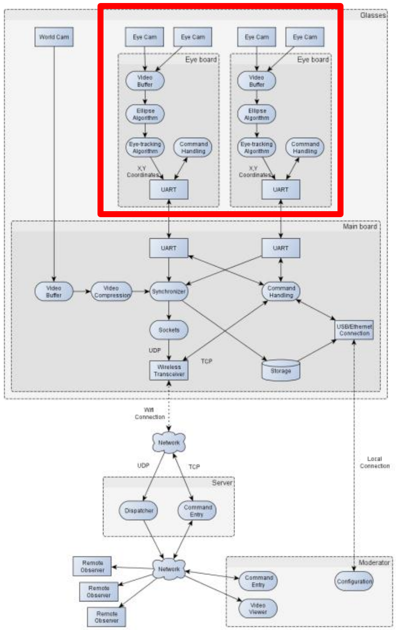
System Overview

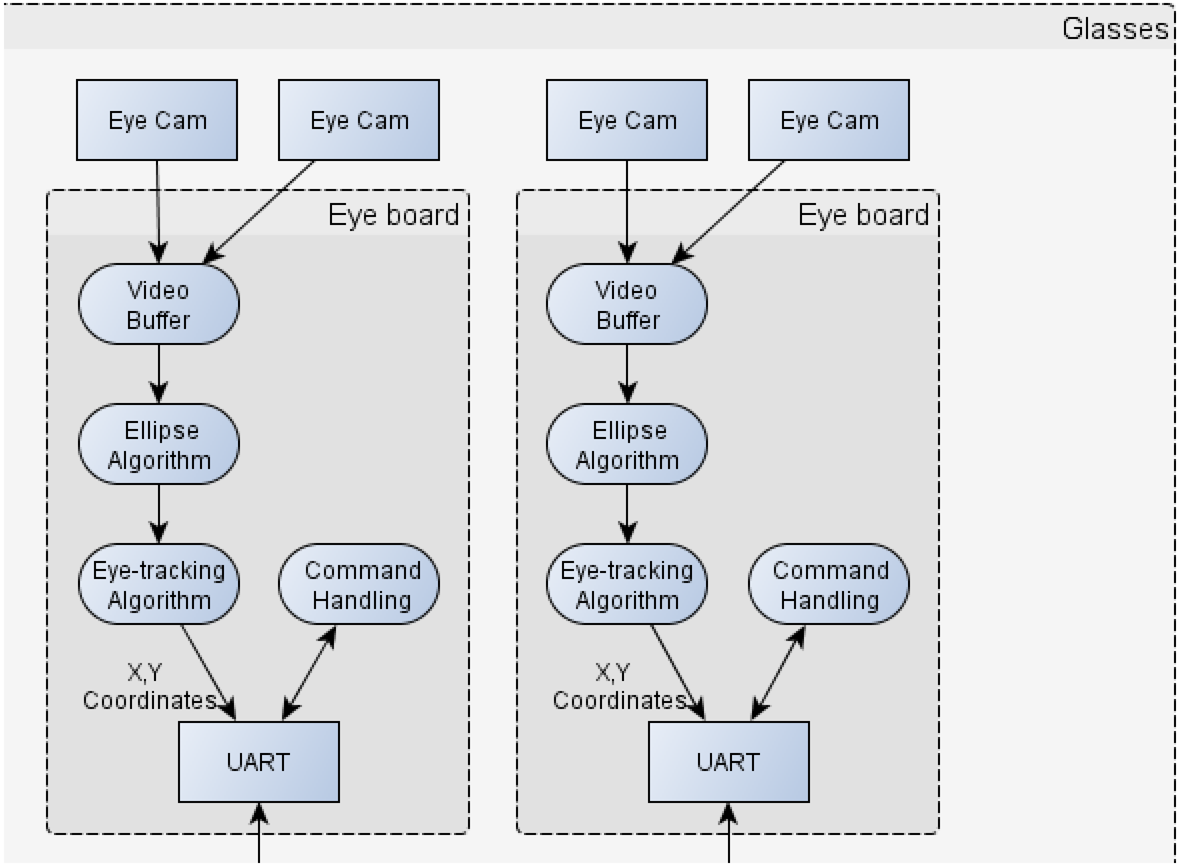
Functional Decomposition



Functional Decomposition

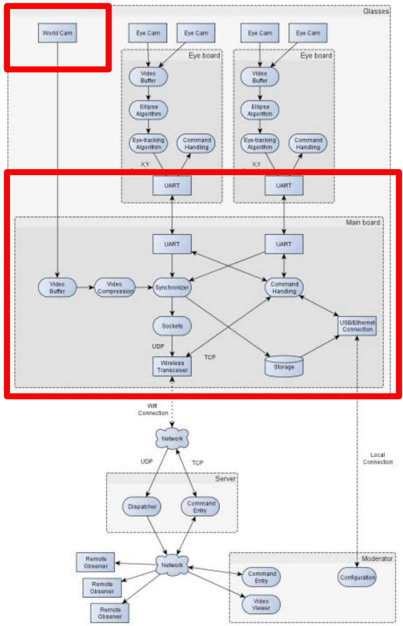
Glasses

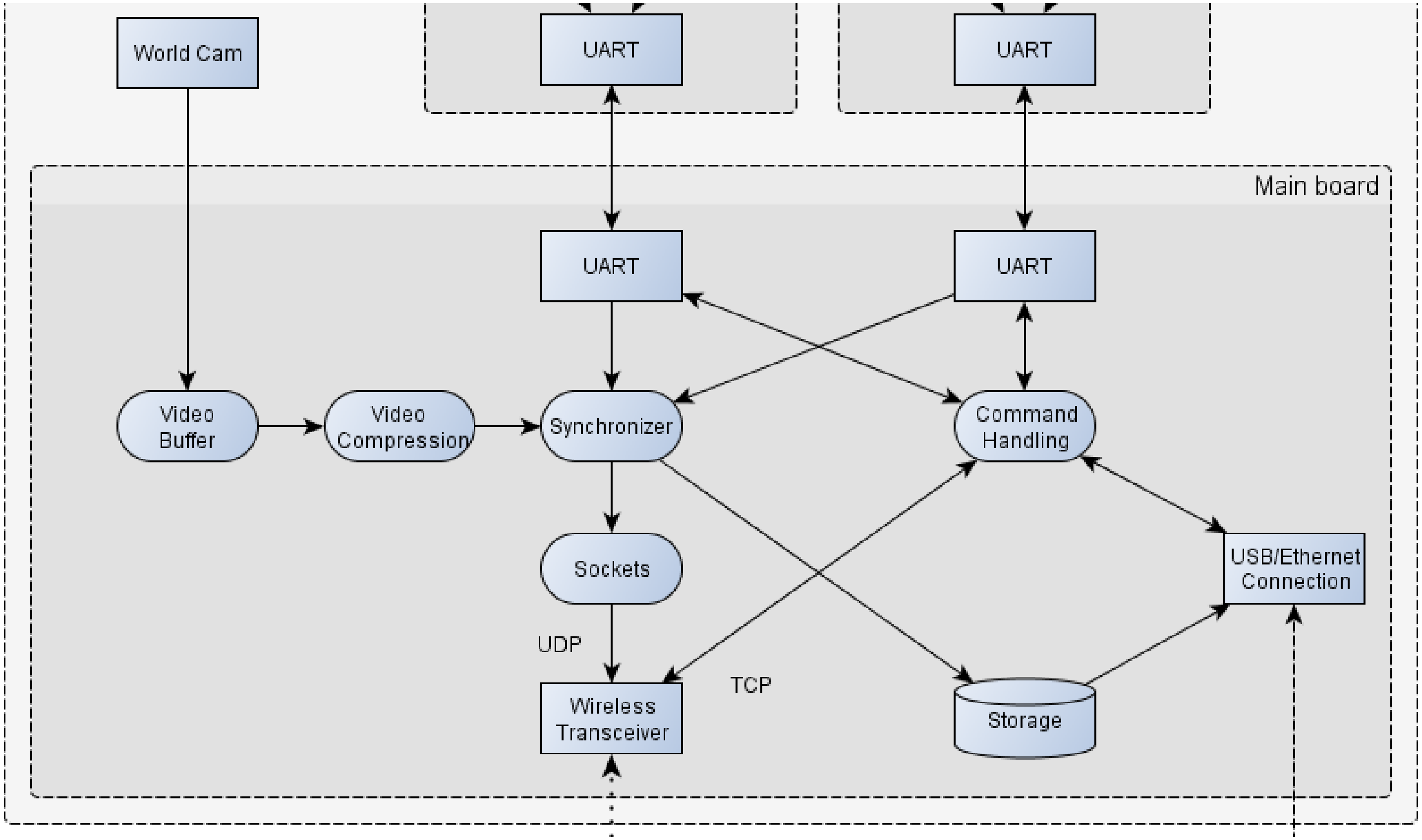




Functional Decomposition

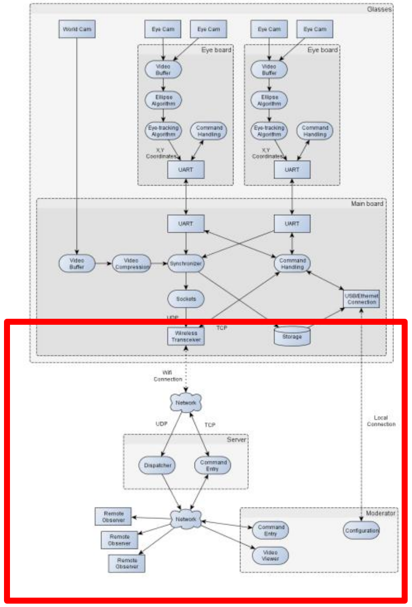
Mainboard

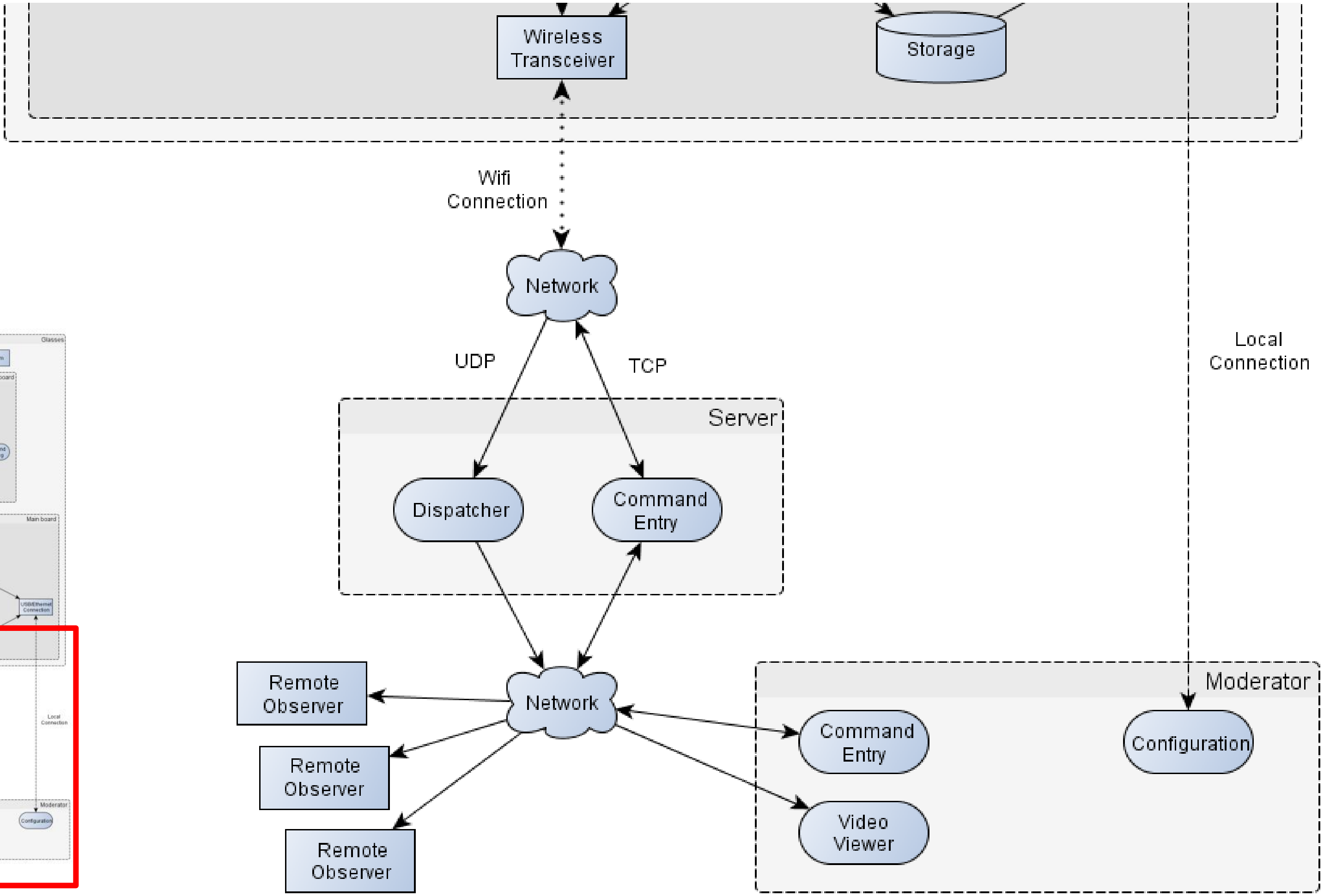




Functional Decomposition

Network





System Specifications

Hardware Used

- Gumstix Duo Vero
- PandaBoard ES
- Battery (TBD)
- CMOS Cameras
- Web Camera

Software Used

- OpenCV
- TBB (Threading Building Blocks)
- Boost (C++)
- Java Runtime

Platforms Used

- Linux
- Windows

Test Plan

- Unit testing
- Real-world Testing
  - Eye testing
  - Wireless testing
- Blackbox testing
- Whitebox testing

Metrics

- Network throughput
- Capture rate of cameras
- Accuracy of eye tracking
- Accuracy of depth

Next Steps

Prototype

Project Status

- First portion of eye tracking done
- Almost all hardware in
- Remaining hardware has been ordered
- Wrote scripts to compile software needed for the eye-tracking algorithm
- Can run programs on the Gumstix

Task Responsibility

- Justin Derby (SE)
  - Software development
  - Eye tracking
- Tyler Burnham (SE)
  - Software development
  - Eye tracking
- Arjay Vander Velden (CPR E)
  - Middle layer software development
- Scott Connell (CPR E)
  - Middle layer software development
- Will Bryan (CPR E)
  - Embedded Programming / Interfaces
- Kris Scott (EE)
  - Physical hardware

Plan for Next Semester

- Implementing
- Prototyping
- Testing
- Bug Fixing
- Testing again
- More bug fixing
- Finished product

Questions