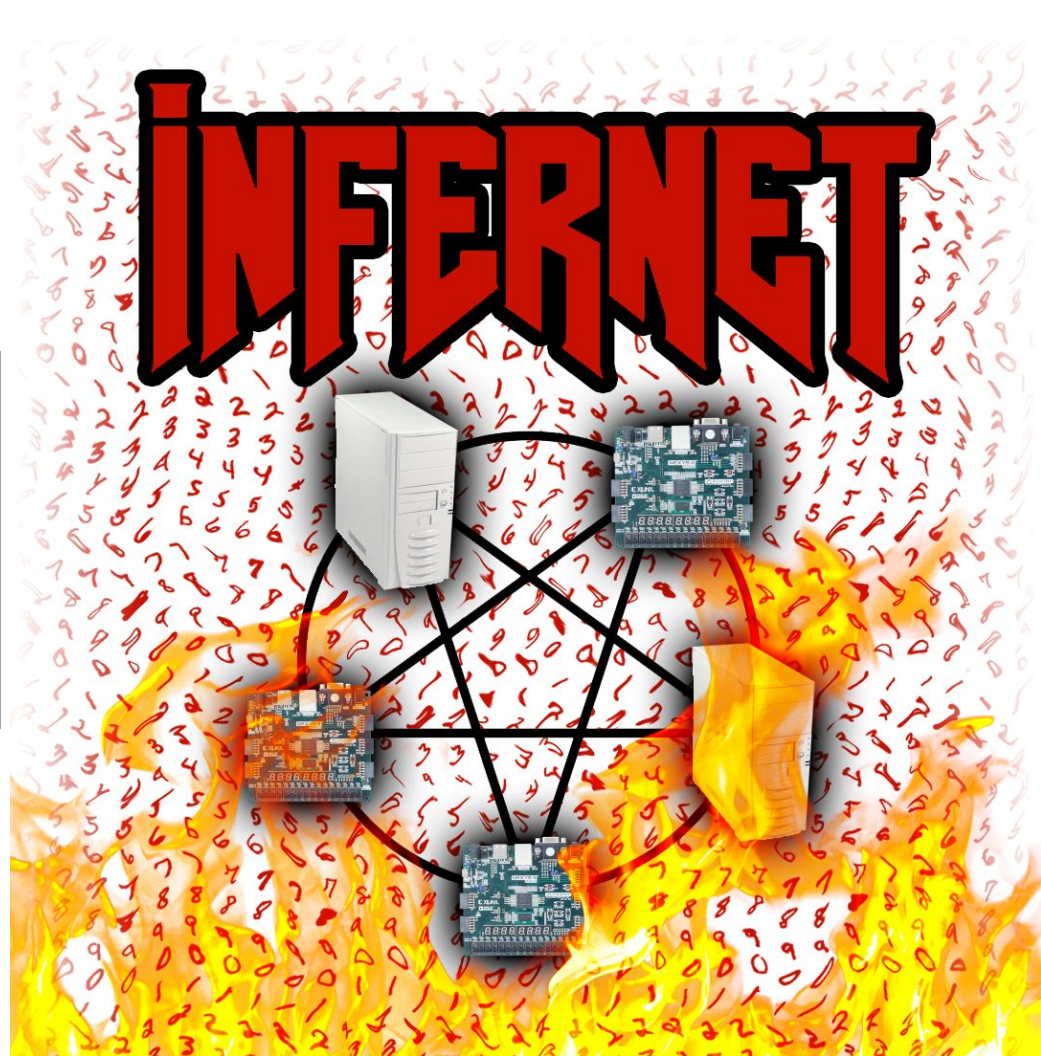


Team 6

ECE532 Project



Meet the team



Andrew



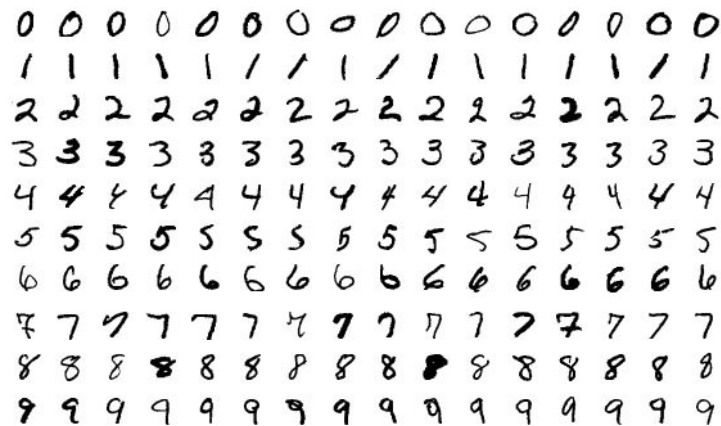
James



Quinn

What is Infernet?

- ❑ Distributed network of machine learning inference modules targeting the MNIST dataset
- ❑ Multiple networked FPGAs with hardware inferencers available to desktop PC clients
 - ❑ For this project: FPGAs and PCs on FPGANet



Sample images from the MNIST dataset

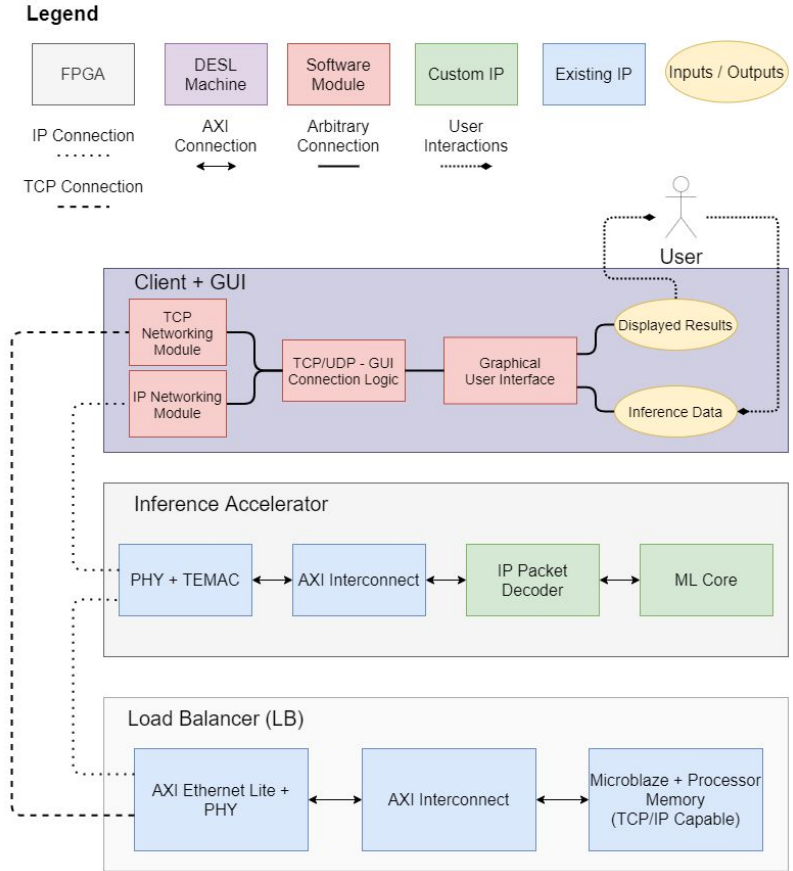
Project Goals

- ❑ Inferences done in pure hardware for speed
- ❑ Networking done in pure hardware on accelerator
- ❑ Concurrent clients and scalability facilitated through a load balancer



System Overview - Original

- ❏ One or more inference accelerators (IA)
- ❏ A load balancer (LB)
- ❏ A desktop client (client)
- ❏ A desktop GUI, which will be implemented within the client to display results



List of IP Created

- ❑ UDP Networking Module
- ❑ MNIST Inference Module

List of Software Created

- ❑ Load Balancer Code
- ❑ Client GUI Code
- ❑ Client Networking Code

List of IP/Software Used

All hardware designs

- ❑ Vivado MII to RMII
- ❑ Vivado Clocking Wizard

Accelerator

- ❑ Vivado Trimode Ethernet MAC (TEMAC)
- ❑ Vivado Reset Module
- ❑ Vivado Clocking Wizard
- ❑ Vivado MII to RMII
- ❑ Vivado AXI Lite TEMAC Controller
- ❑ Vivado DSP Builder
- ❑ Vivado BRAM Generator
- ❑ Vivado AXI-Stream FIFO Block

Load Balancer

- ❑ LWIP
- ❑ AXI Ethernet Lite
- ❑ AXI interconnect
- ❑ Microblaze

Problems

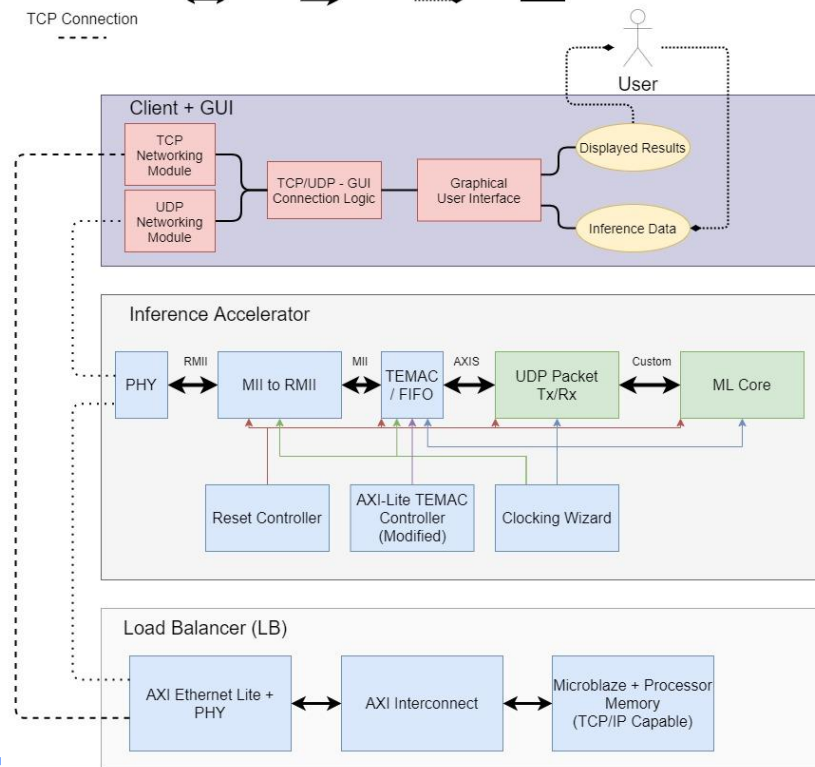
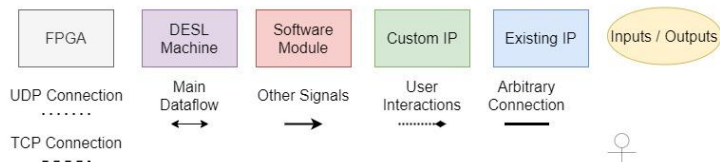
- ❑ Vivado RTOS + lwip doesn't work
- ❑ Raw IP Rx on LWIP was not well supported
- ❑ Integrations were very time consuming
- ❑ Very wide/parallel neural net not possible
- ❑ Tricky timing closure



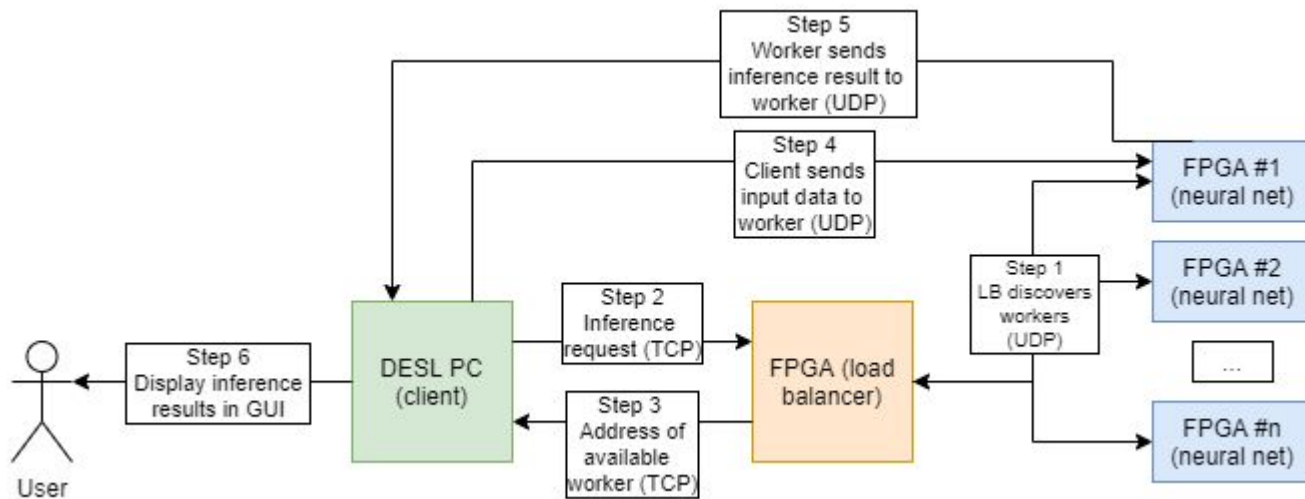
System Overview - Final Design

- Full complexity shown
- Additional logic for clocking, resets, TEMAC configuration

Legend



Final User Workflow



Final Results

- ❑ All initial goals achieved
- ❑ Adjusted network communication to work around roadblocks
- ❑ Pretty GUI
- ❑ ~93.5% inference accuracy in hardware :-)




Design Process - Team

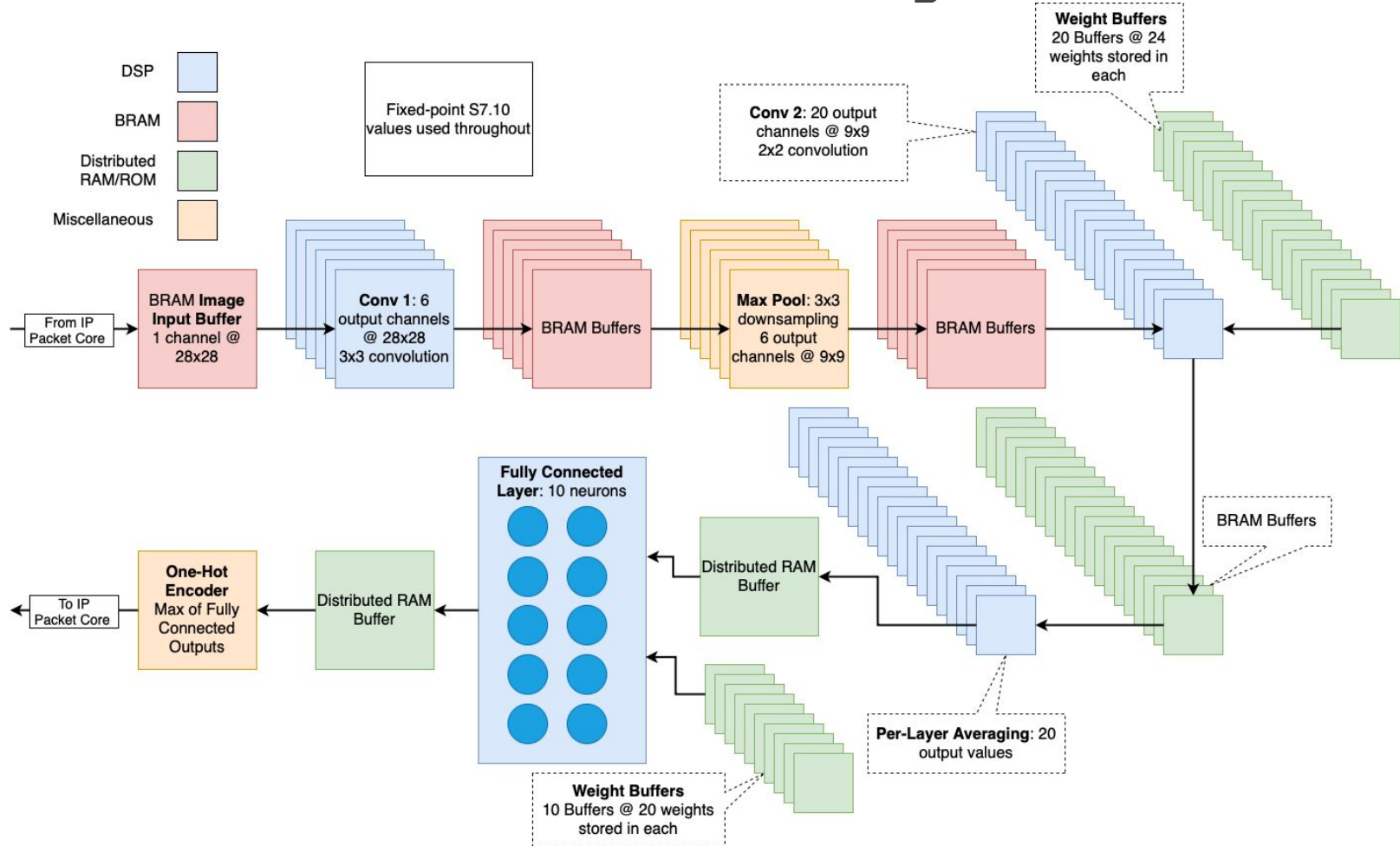
- ❑ Complimentary Skills
- ❑ Divided work optimizing for completion time and concurrency
- ❑ Started with more complex designs, simplified to core ideas
- ❑ Interface design done together




Design Process - Neural Net

- 
- ❑ Architected and developed neural net in Python first
 - ❑ Fast prototyping!
 - ❑ Very serialized due FPGA resource limitations
 - ❑ Constantly keeping in mind resource usage

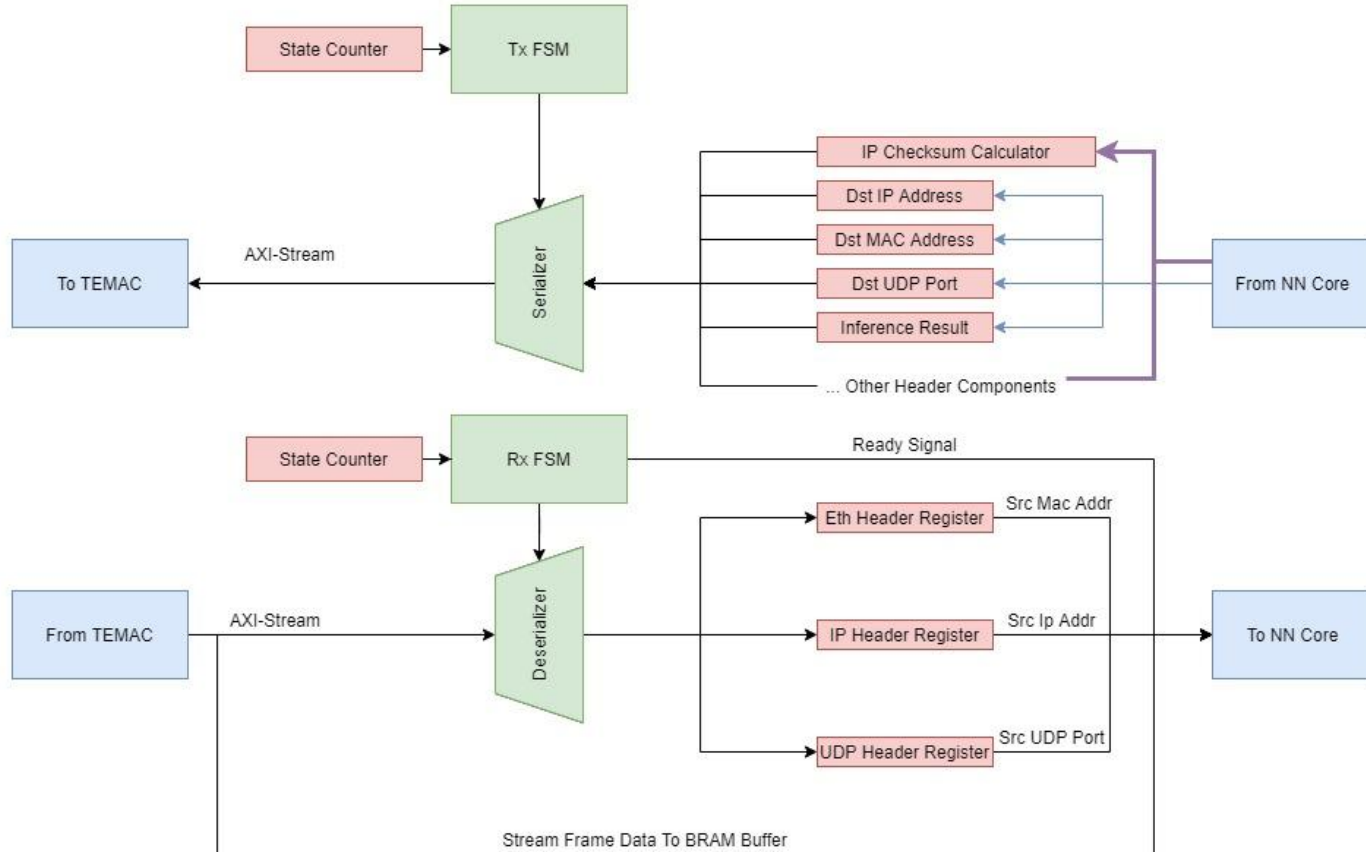
Neural Net Block Diagram



Design Process - UDP Tx/Rx

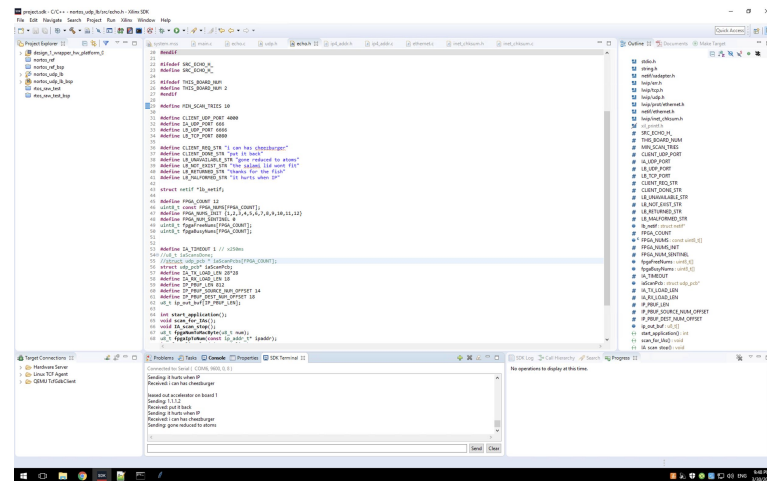
- 
- ❑ Conform to the standard or perish
 - ❑ Established communication protocol for inferences
 - ❑ Had to filter network garbage for accelerator
 - ❑ Had to configure TEMAC and its accompanying components

Design Process - Network IP

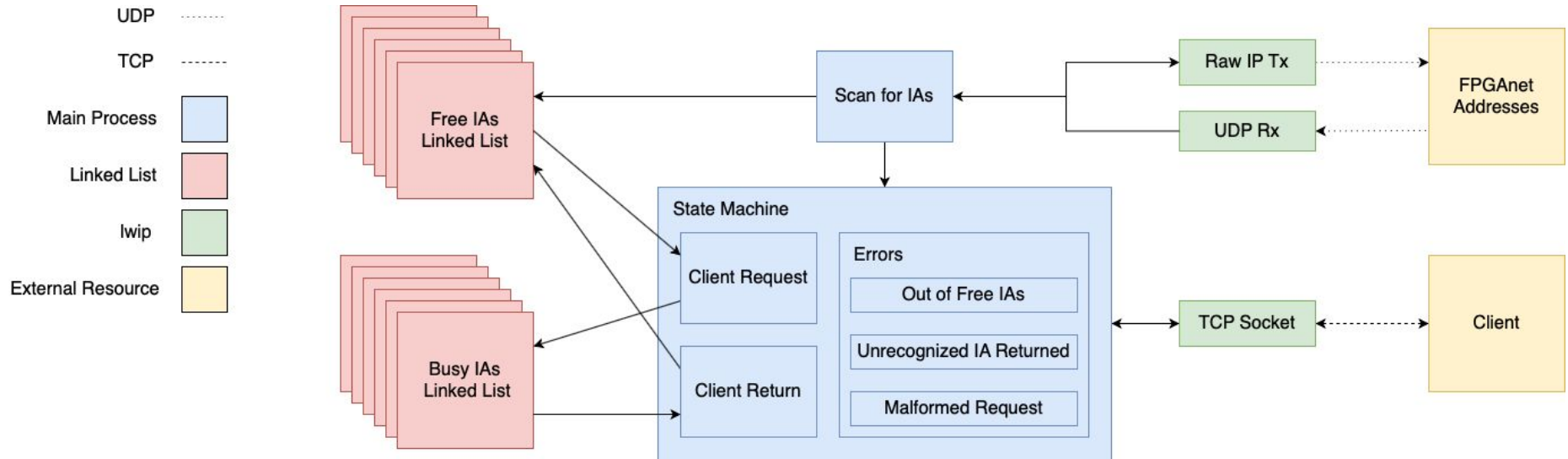


Page 10 of 10

- 

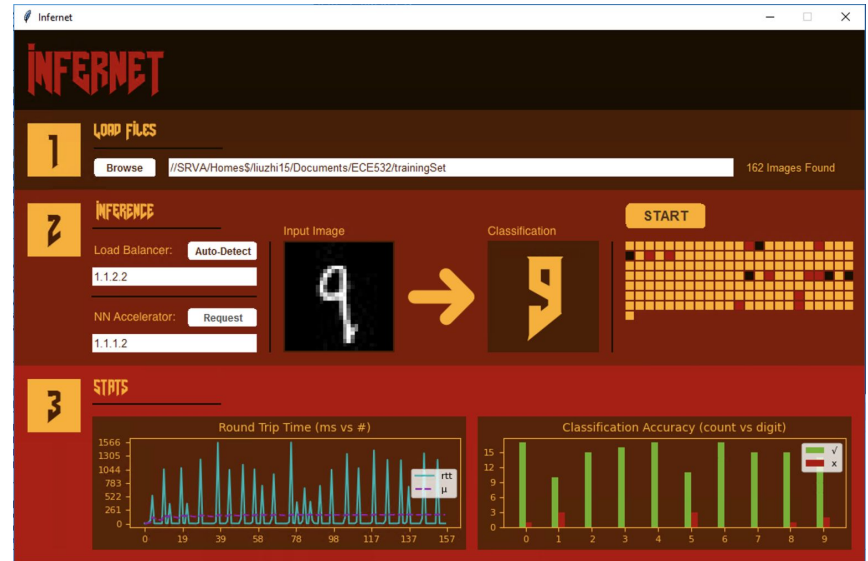


Load Balancer Block Diagram

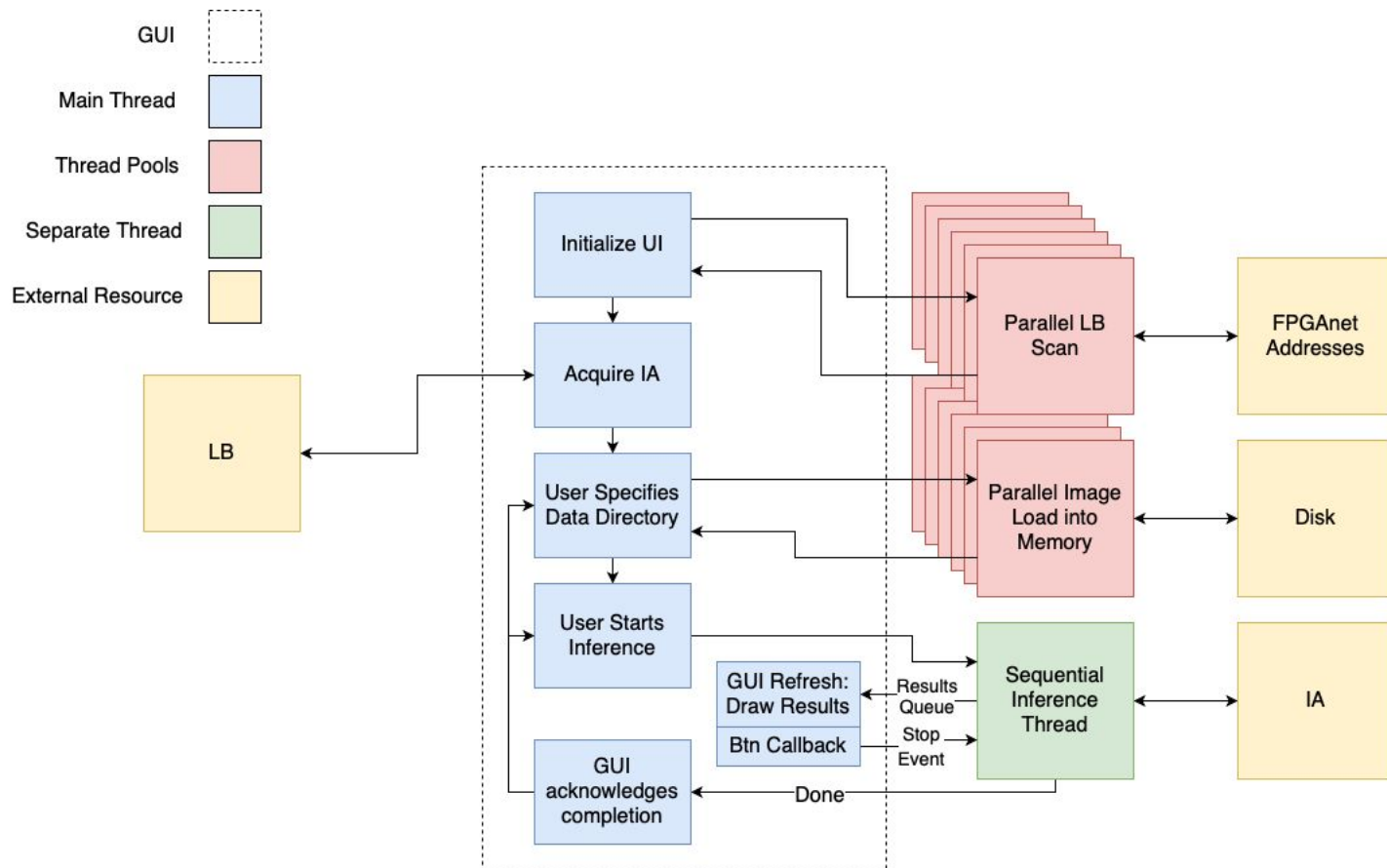


Design Process - GUI

- ❑ Threaded, asynchronous IO
- ❑ Modular python code
- ❑ Cross platform
- ❑ Automatic discovery
- ❑ On theme colour scheme
- ❑ tkinter abuse

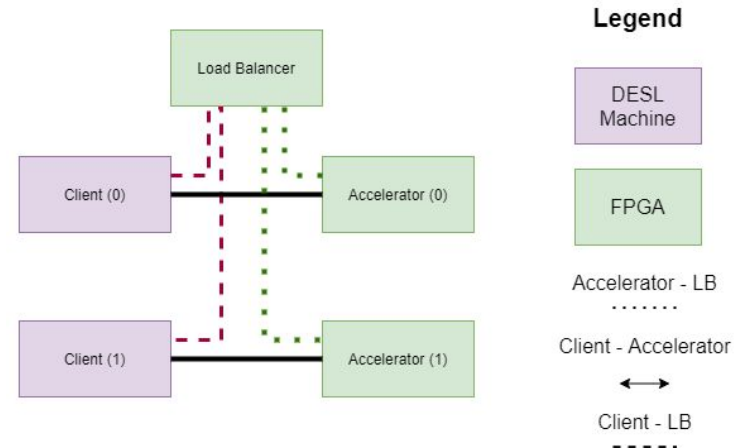


GUI Block Diagram

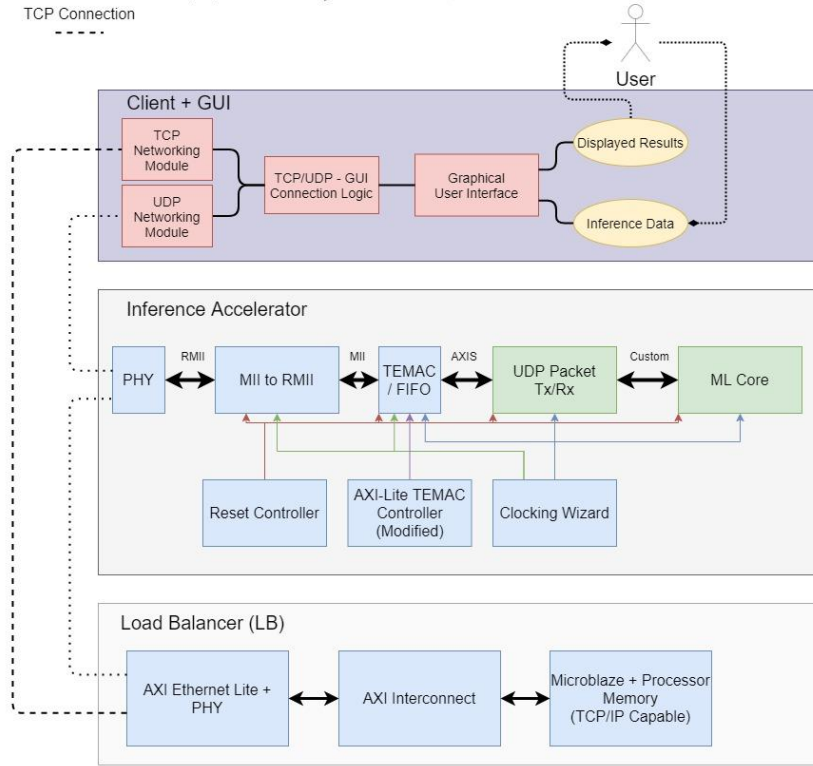
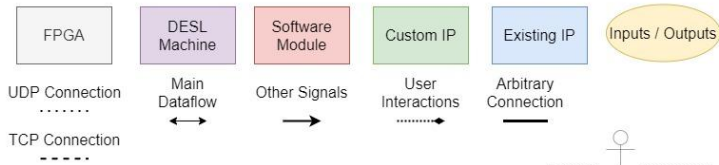


Final Demo

- Two clients
- Two IAs
- One LB
- Clients are concurrently running inferences through accelerators, all of this arbitrated by the LB.
- Stats including client side round trip time (rtt) and classification accuracy shown in graphs



Legend



Questions?

93.4%
accuracy

