Virtual Vaccine Passport Scanner for Remote Entry Approval

ECE 532 Group 3:

Mustafa Kanchwala

Guoxian Wu

Eduardo Stecca Ortenblad

Xuening Dong

Project Significance

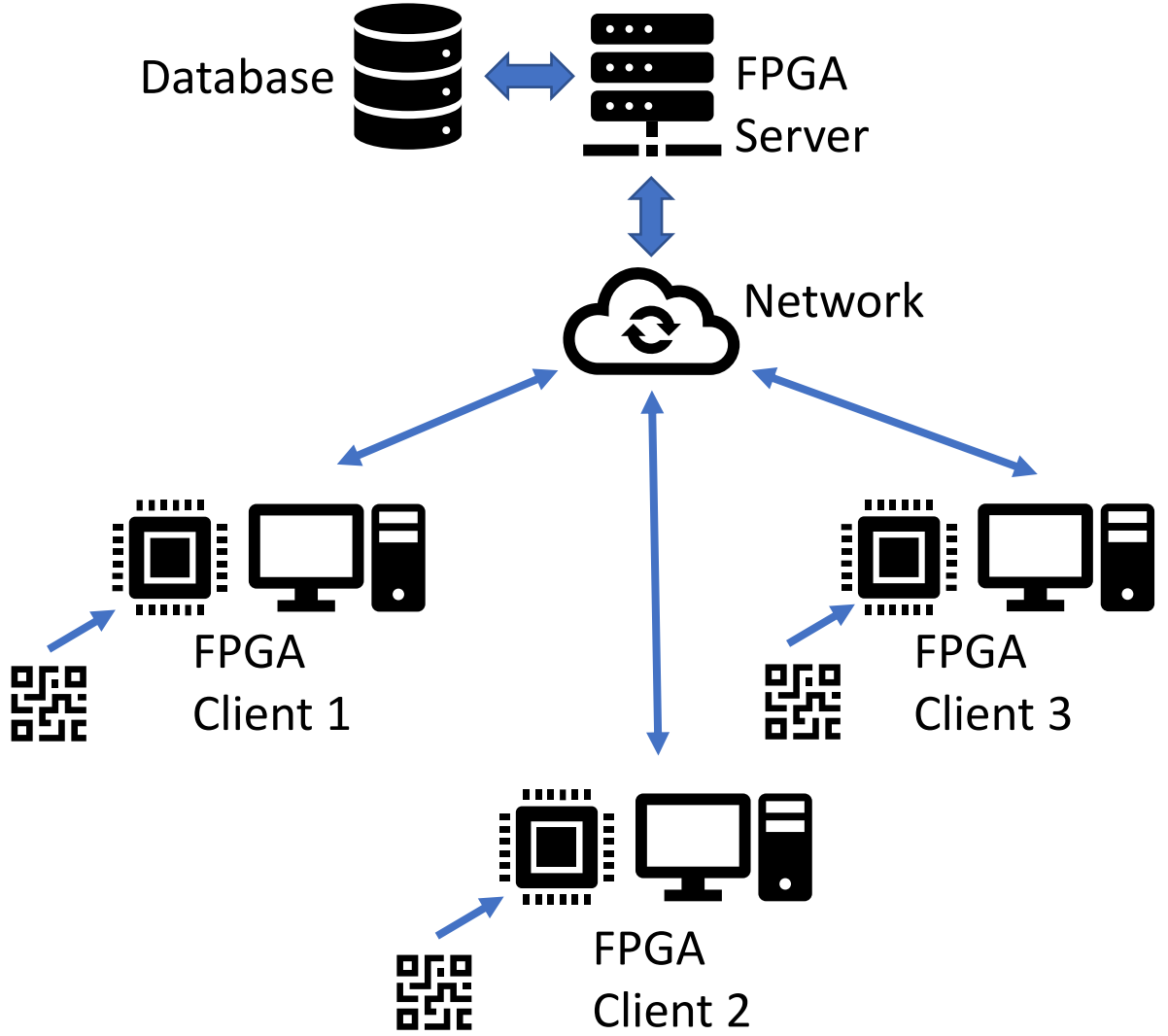
- What's out there:
  - A Human scans the vaccine passport and allows or denies entry
- Why that’s bad:
  - Risk of Exposure – Very High!
  - Conflicts arising from enforcement – High!
- What can be done:
  - A Hardware based - remote server-controlled scanning & verification

[1]

[2]

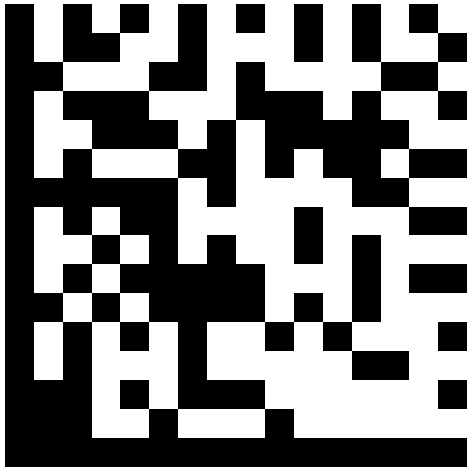
Data Center Aspect

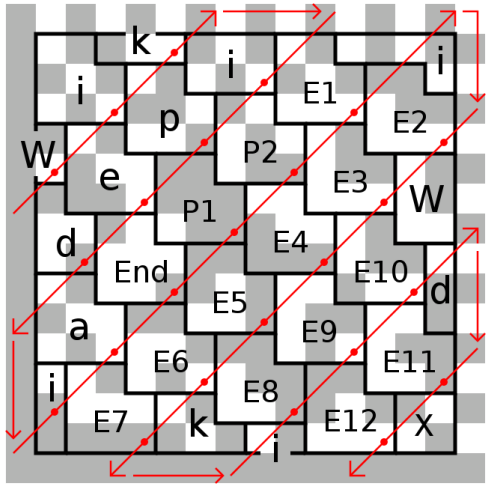
- Secure server stores vaccination information of:
  - Citizens - @ national level
  - Employees - @ corporation level
  - Residents - @ building level
- Client terminals are placed near entry points
  - Once passport is scanned, information is fetched and validated
  - Allow or Deny decision is made and displayed at the client



Background

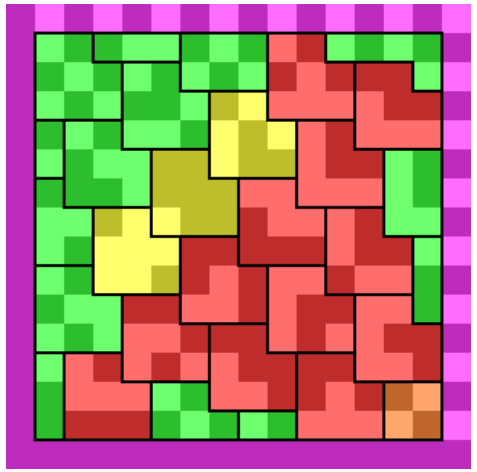
- Data Matrix:
  - Two-dimensional code
  - Black and white "cells"
  - Either a square or rectangular pattern





Data matrix of "Wikipedia"

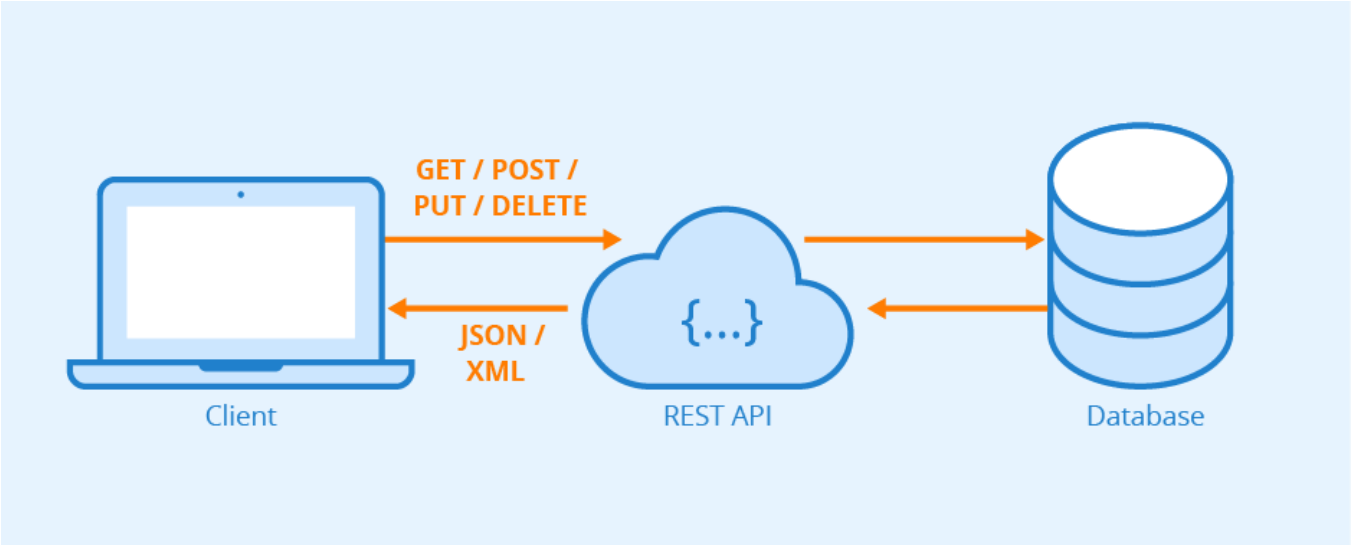
Purple: Finder pattern
Green: Data
Yellow: End of Massage & Padding
Red: Error Correction Code
Orange: Unused space



[3]

More Background

- REST APIs
  - GET – Retrieve data
  - POST – Create new data
  - PUT – Update data
  - DELETE – Delete data

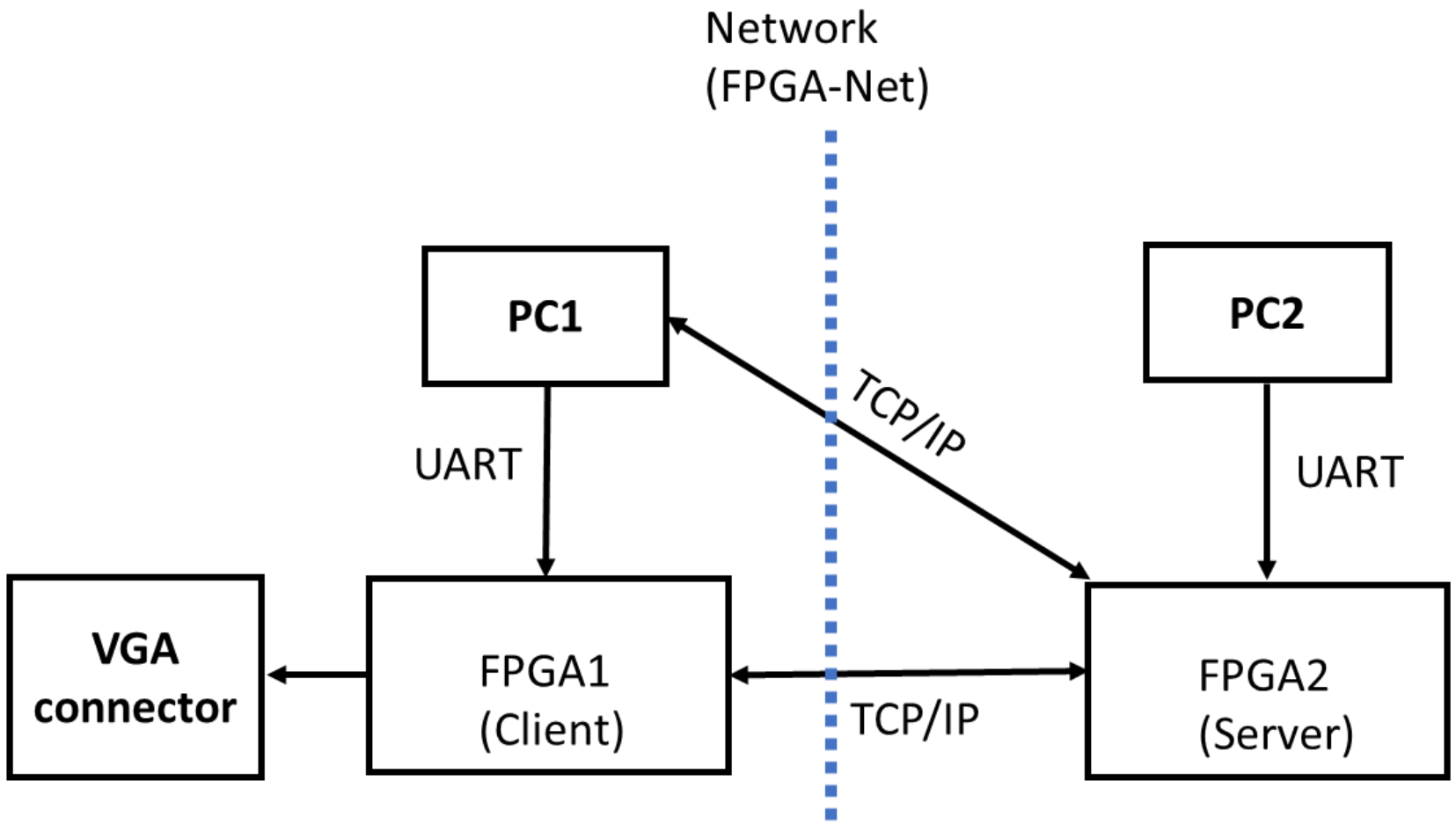


[4]

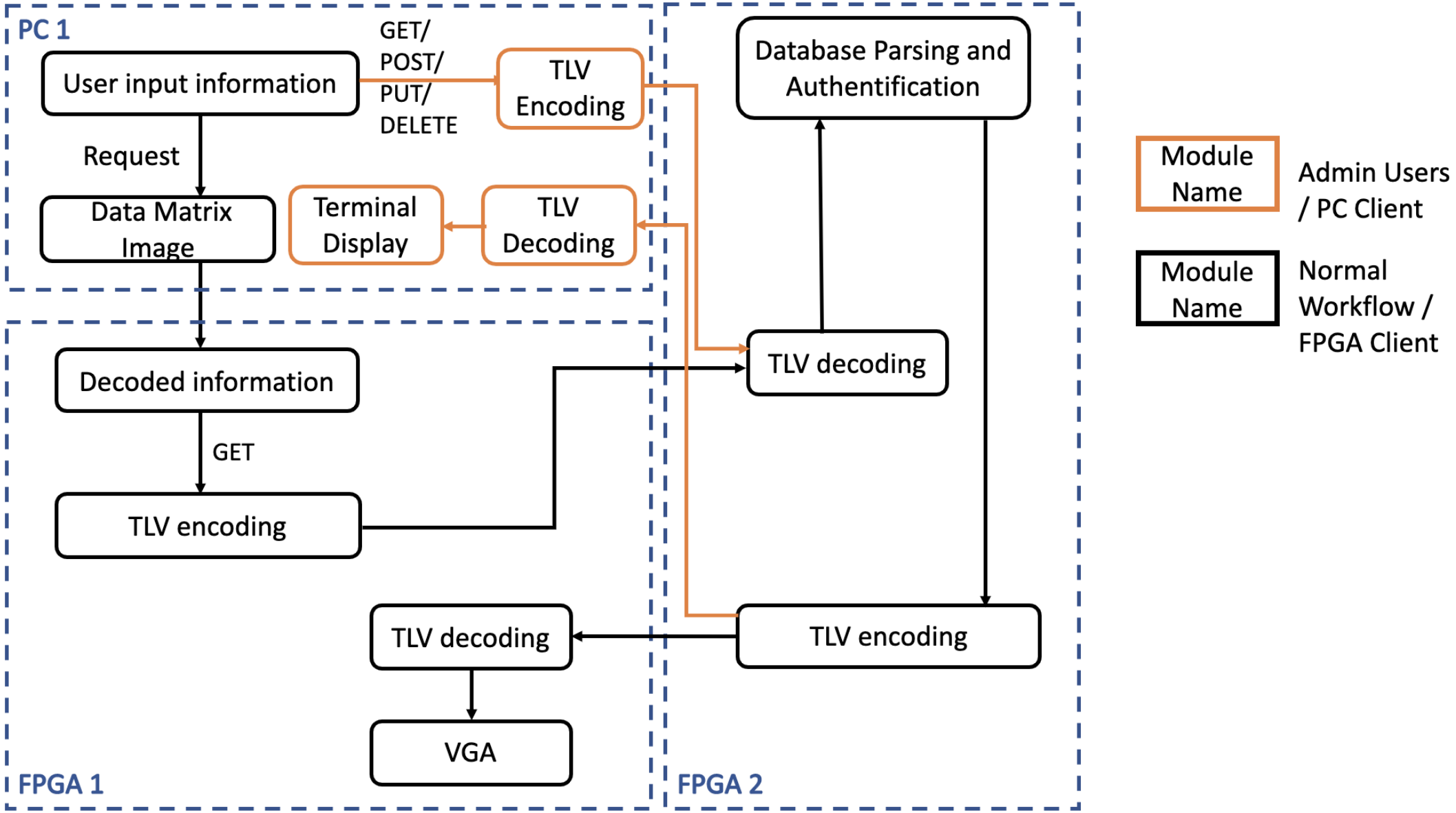
- TLV (**T**ype-**L**ength-**V**alue) Encoding:
  - Packets for network communication – up to 128 bytes
  - Encoding and Decoding

Type	Length	Value
1 Byte	1 Byte	Length Bytes

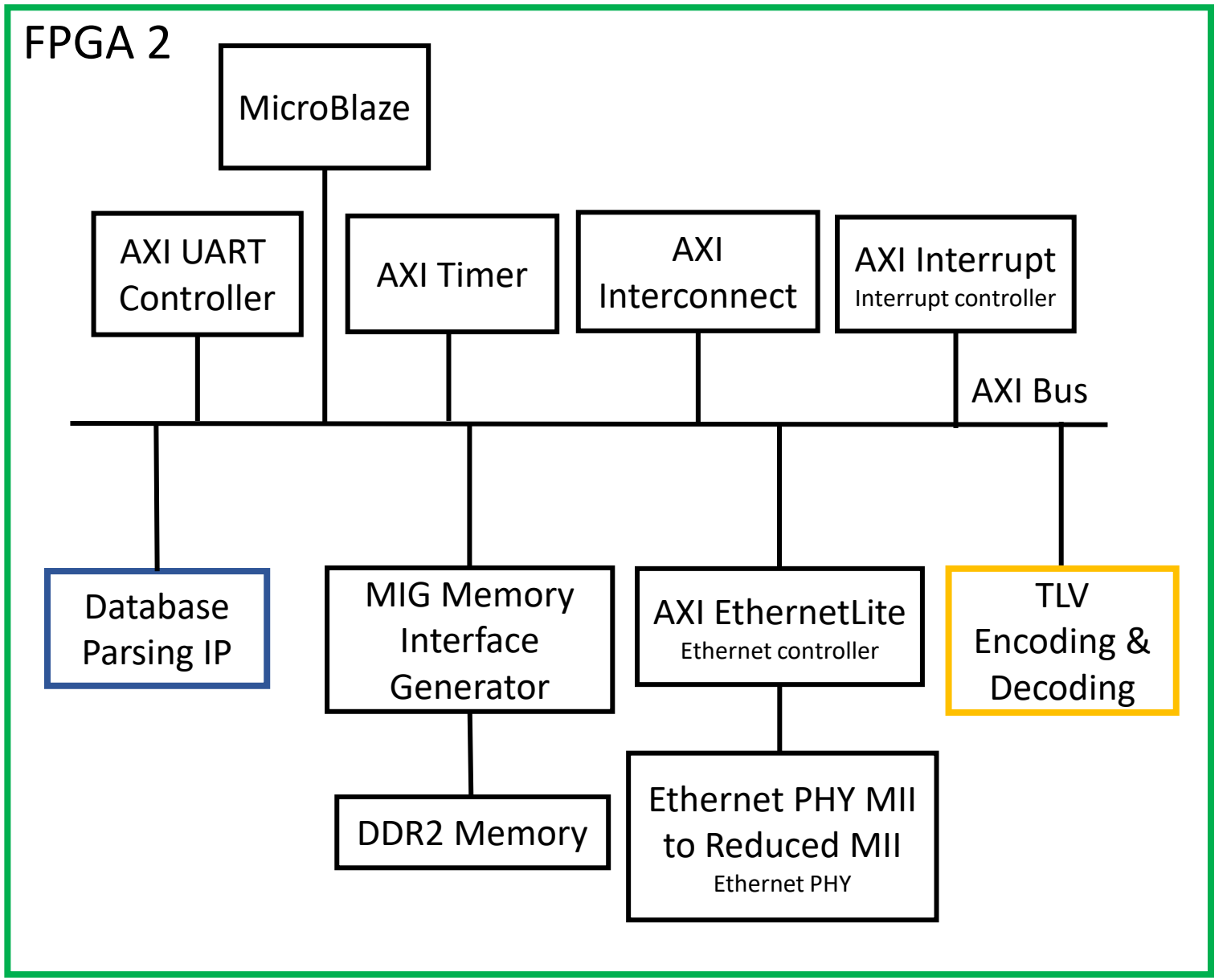
Proposed System



System Overview



Server

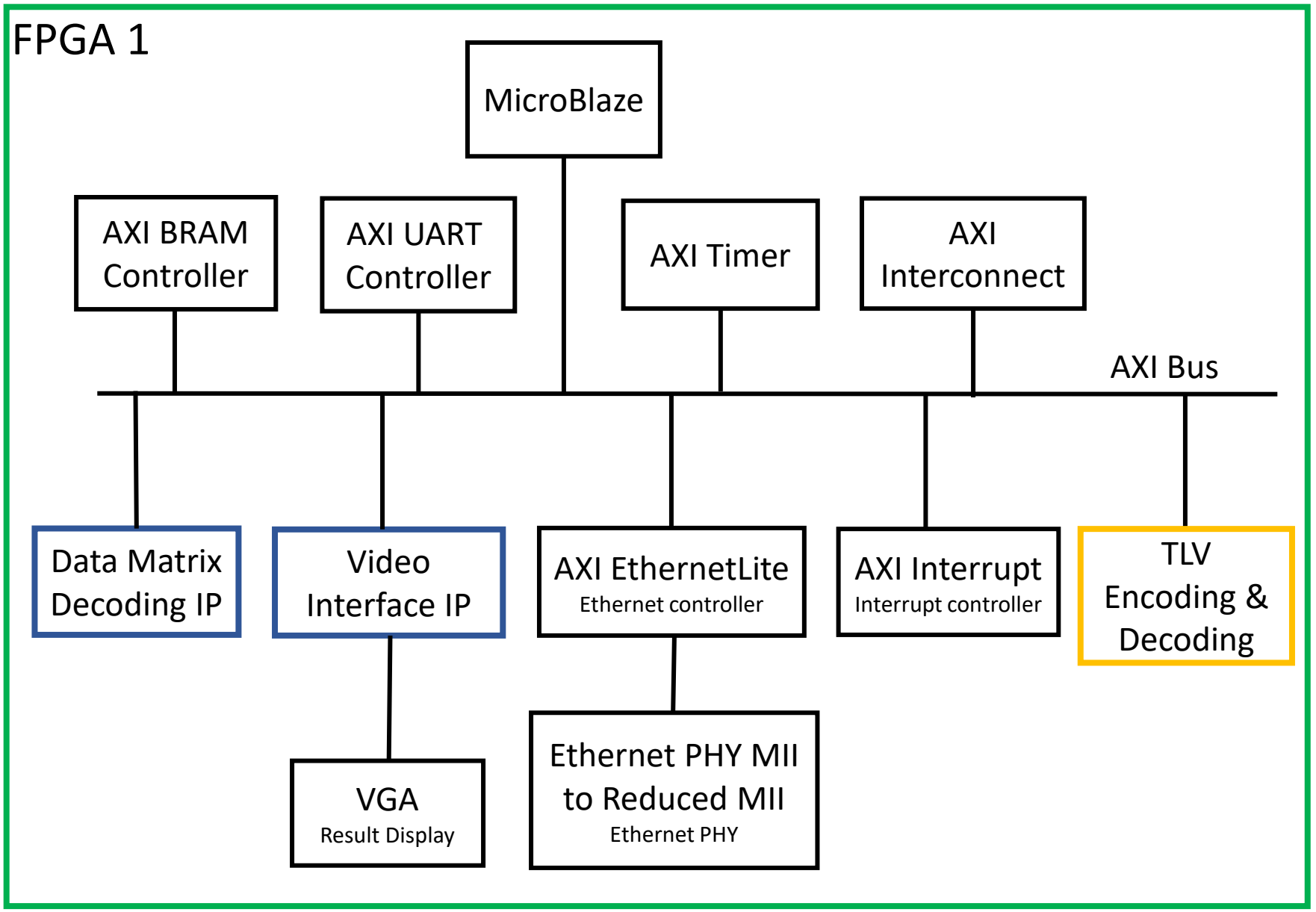


Module Name — Custom IP

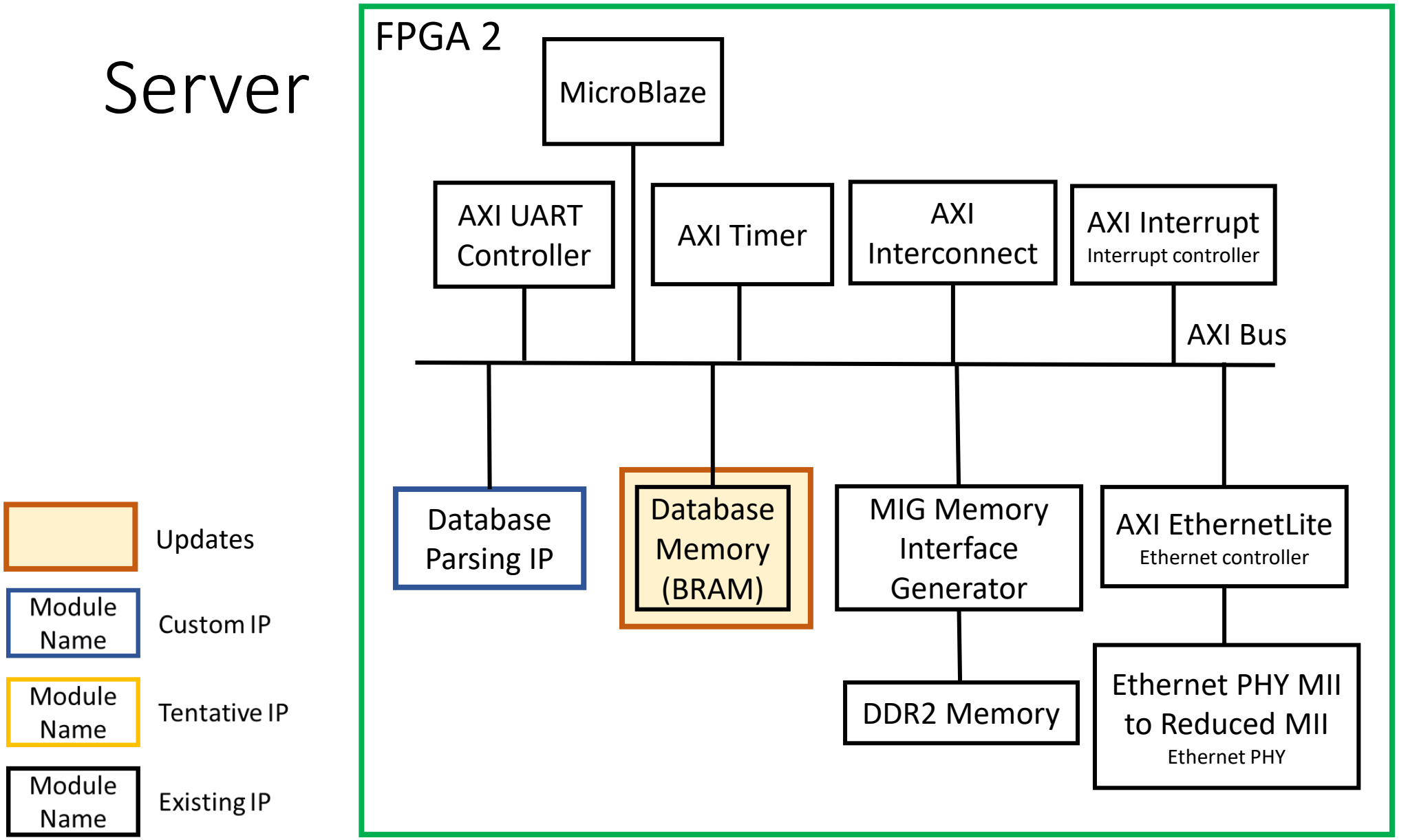
Module Name — Tentative IP

Module Name — Existing IP

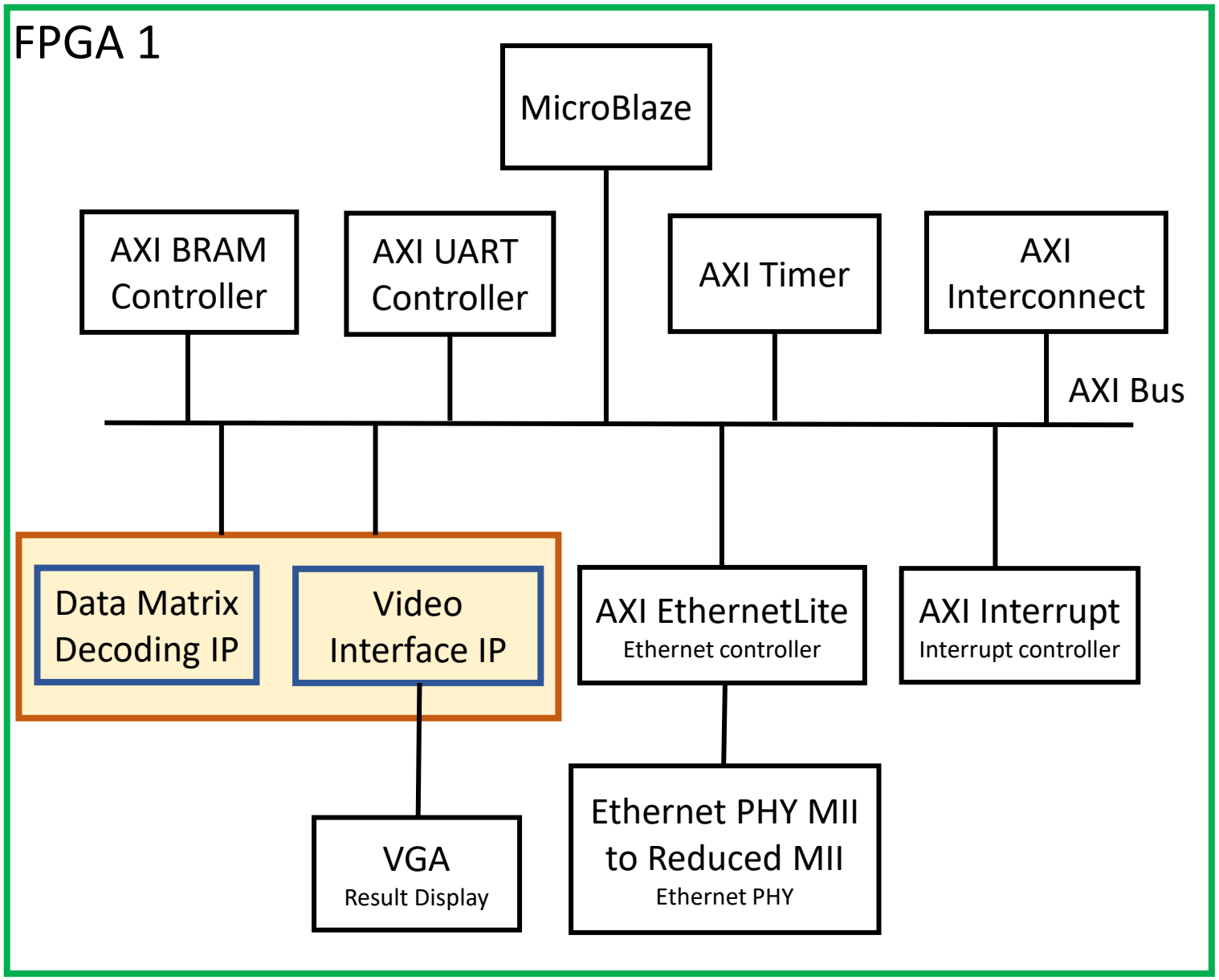
Client


FPGA 1
MicroBlaze
AXI BRAM Controller
AXI UART Controller
AXI Timer
AXI Interconnect
AXI Bus
Data Matrix Decoding IP
Video Interface IP
AXI EthernetLite
Ethernet controller
AXI Interrupt
Interrupt controller
TLV Encoding & Decoding
VGA
Result Display
Ethernet PHY MII to Reduced MII
Ethernet PHY
Module Name
Custom IP
Module Name
Tentative IP
Module Name
Existing IP

Server


FPGA 2
MicroBlaze
AXI UART Controller
AXI Timer
AXI Interconnect
AXI Interrupt
Interrupt controller
AXI Bus
Database Parsing IP
Database Memory (BRAM)
MIG Memory Interface Generator
AXI EthernetLite
Ethernet controller
DDR2 Memory
Ethernet PHY MII to Reduced MII
Ethernet PHY
Updates
Module Name
Custom IP
Module Name
Tentative IP
Module Name
Existing IP

Client


FPGA 1
MicroBlaze
AXI BRAM Controller
AXI UART Controller
AXI Timer
AXI Interconnect
AXI Bus
Data Matrix Decoding IP
Video Interface IP
AXI EthernetLite
Ethernet controller
AXI Interrupt
Interrupt controller
VGA
Result Display
Ethernet PHY MII to Reduced MII
Ethernet PHY
Module Name
Custom IP
Module Name
Tentative IP
Module Name
Existing IP

Implementation Plan - Partitioning

PC 1	FPGA 1 (Client)	FPGA 2 (Server)	PC 2
Python scripts to implement - Data Matrix Code Generation - UART data transfer - TCP Client	- Data Matrix IP - VGA IP - UART data transfer - TCP Client (Microblaze) - TLV	- Database parsing IP - TCP Server (Microblaze) - TLV	- UART connection to load database onto FPGA2

Implementation Plan

- **Milestone 1: Research and setup**
  - TLV/Data Matrix encoding/decoding rules defined
  - Database/VGA control setup
- **Milestone 2: Basic functionality implementation**
  - Python script for TLV encoding/decoding
  - TCP server and client
  - Database/VGA/data matrix decoder IP implementation
- **Milestone 3: Finalize most units**
  - TLV on MicroBlaze
  - Data Matrix decoder/database/VGA IP Testing
  - PC – FPGA communication

Implementation Plan

- **Milestone 4: Buffering**
  - Finalize IPs
  - Test on PC-FPGA connection
  - Catch up with any delayed milestone
- **Milestone 5 & Milestone 6: Integration**
  - Finalize all communications (PC-FPGA/ FPGA-FPGA)
  - Test the overall dataflow

Testing and Integration Plan

Testbench/ILA/Software observation:

- VGA display
- Data Matrix decoding
- Database parsing

Full System Integration Testing:

- Integrate and test main functional IP blocks
- System level testbench to simulate the connection between hardware blocks.
- Use ILA to monitor the hardware and GDB debug to monitor the software.
- Compare the information displayed on VGA with the input information.

Risks and Uncertainties

Sl no	Risk and Uncertainties	Contingency Plan
1	Challenges in hardware IP design - Database IP - VGA IP - DataMatrix Decode IP	Create a backup software implementation in MicroBlaze. Use this approach to debug and resolve issues with hardware implementation
2	Challenges encountered when using image input for DataMatrix decoding IP	Use PC to pre-process the data into bits of 0s & 1s
3	Run out of memory on Nexys DDR	Optimize memory allocation to the BRAM and use space on DDR2 or Switch to Nexys Video Board
4	Proposed tasks are not completed as planned	Use Milestone 4 as a Buffer week

This Photo by Unknown author is licensed under CC BY-NC-ND.

Reference

[1] “15 EU Countries & Iceland already issuing COVID-19 vaccine passports for travel - Finland & Hungary lagging behind,” *SchengenVisaInfo.com*, 16-Jun-2021. [Online]. Available: https://www.schengenvisainfo.com/news/15-eu-countries-iceland-already-issuing-covid-19-vaccine-passports-for-travel-finland-hungary-lagging-behind/. [Accessed: 08-Feb-2022].

[2] N. Wells, “B.C. residents will require 2nd vaccine card for travel, Province says,” *CBCnews*, 22-Oct-2021. [Online]. Available: https://www.cbc.ca/news/canada/british-columbia/bc-covid-travel-rules-1.6220473. [Accessed: 08-Feb-2022].

[3] “Data matrix,” *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/Data_Matrix. [Accessed: 08-Feb-2022].

[4] T. Naeem, “Rest API definition: What is a rest api integration (restful API)?,” *Astera*, 28-Jan-2020. [Online]. Available: https://www.astera.com/type/blog/rest-api-definition/. [Accessed: 08-Feb-2022].