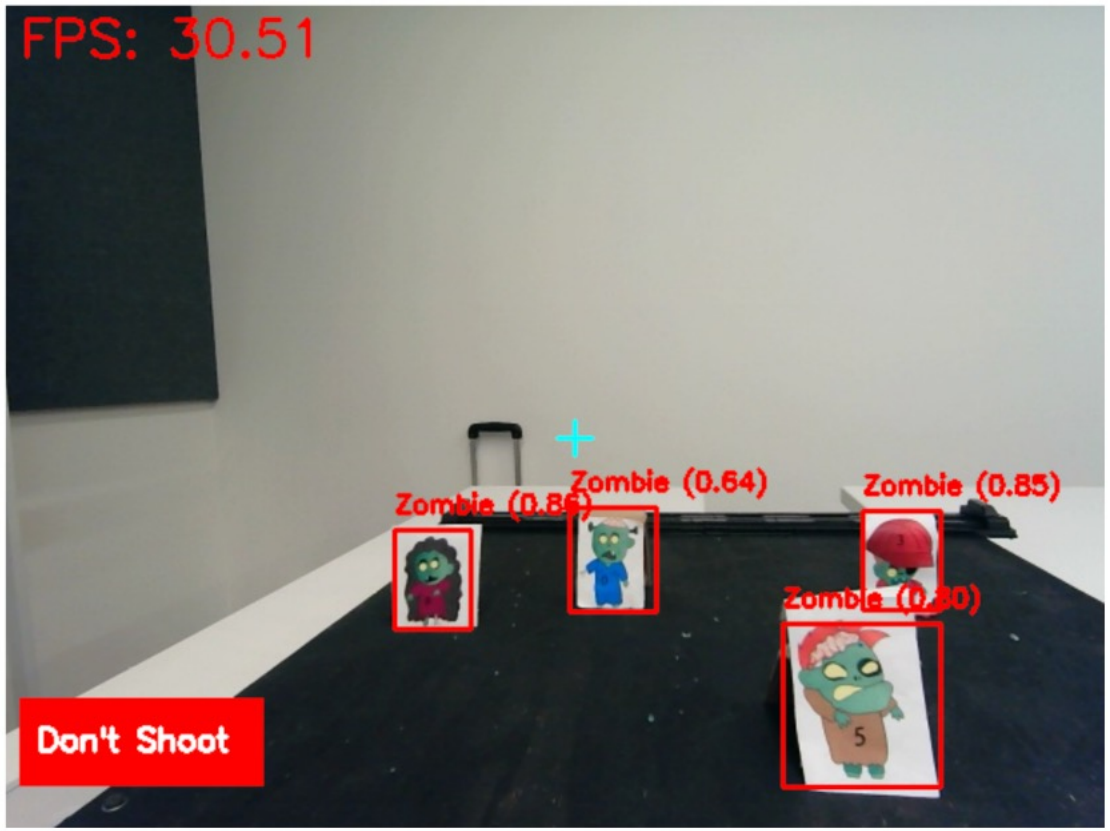
A Reliable AI Cannon System for Automated Target Prioritization and Engagement

Final Demonstration – Team 5 ROCAT (Robust Cannon Technology)

February 14, 2025

Key Features

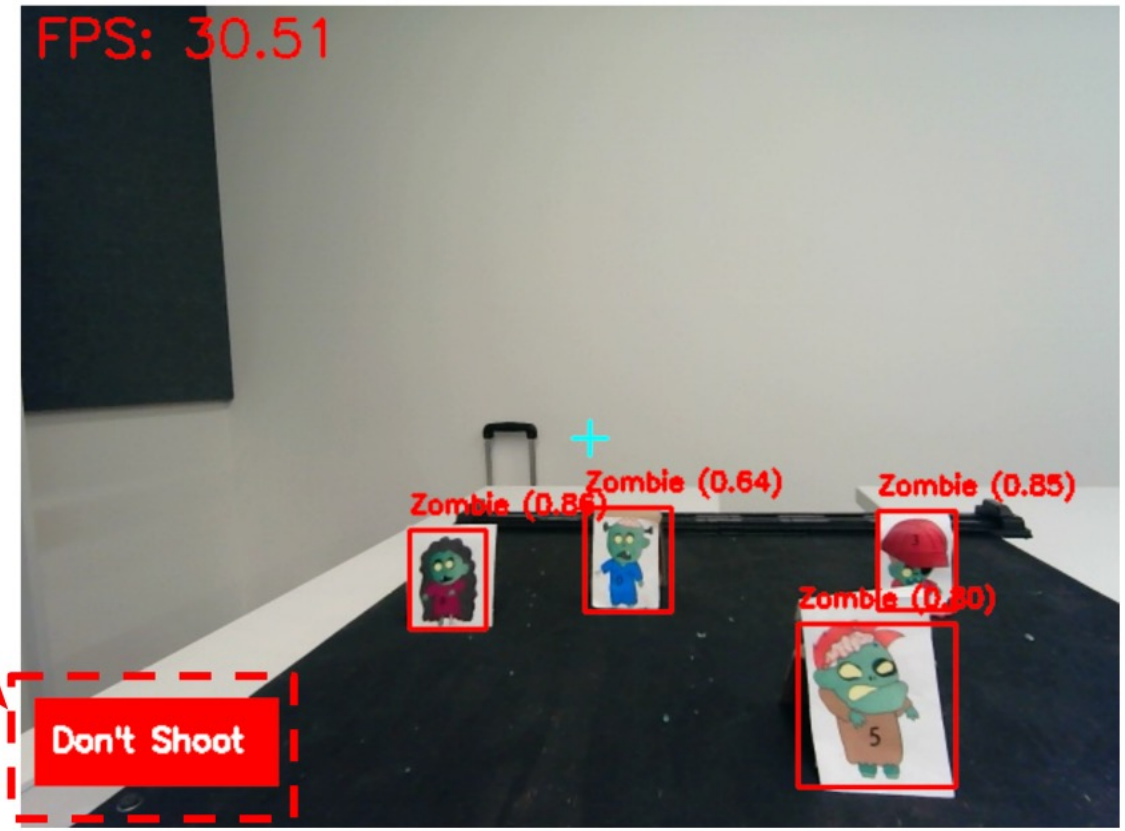
- We aim to:
  - ***Improve the reliability*** of the cannon system
  - Enable ***autonomous prioritization*** of targets
- Key features for enhanced reliability:
  1) ***Emergency Stop*** (GUI button Interaction)
  2) ***Nearest-Zombie*** Targeting
  3) ***Non-overlapping*** targets



< Cannon View (GUI) >

Key Features

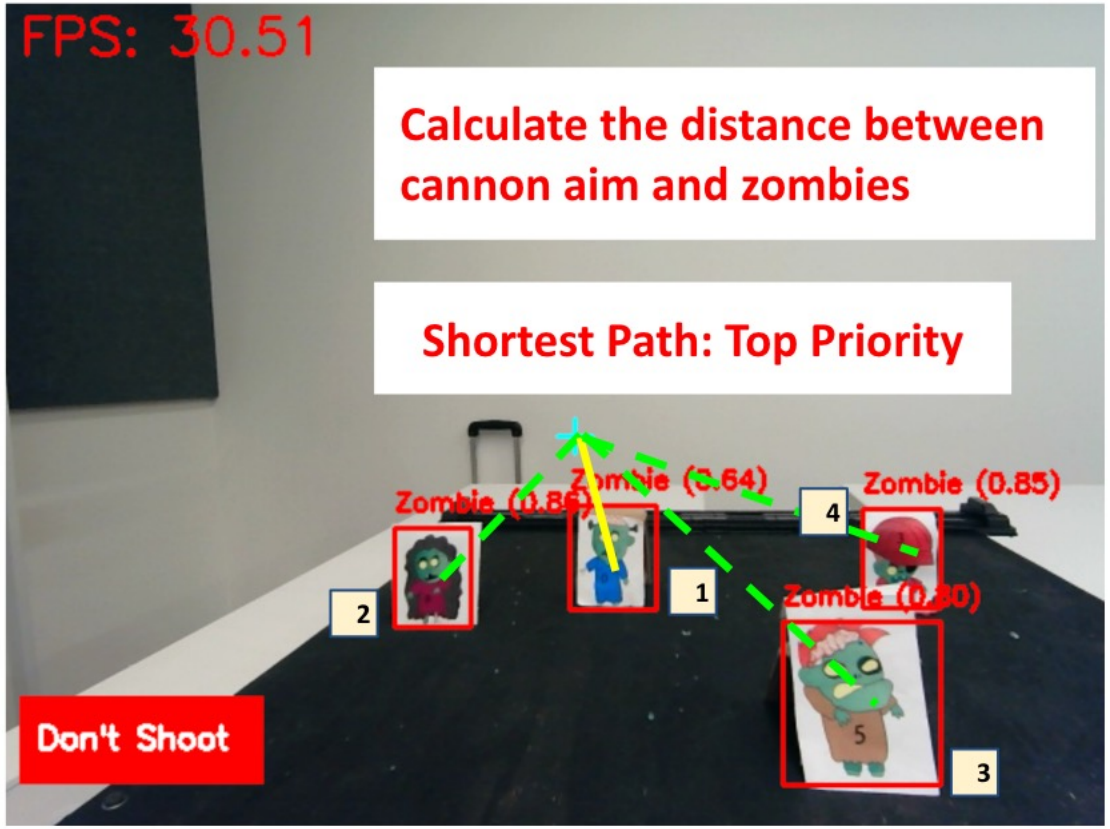
- We aim to:
  - ***Improve the reliability*** of the cannon system
  - Enable ***autonomous prioritization*** of targets
- Key features for enhanced reliability:
  1) ***Emergency Stop*** (GUI button Interaction)
  2) ***Nearest-Zombie*** Targeting
  3) ***Non-overlapping*** targets



< Cannon View (GUI) >

Key Features

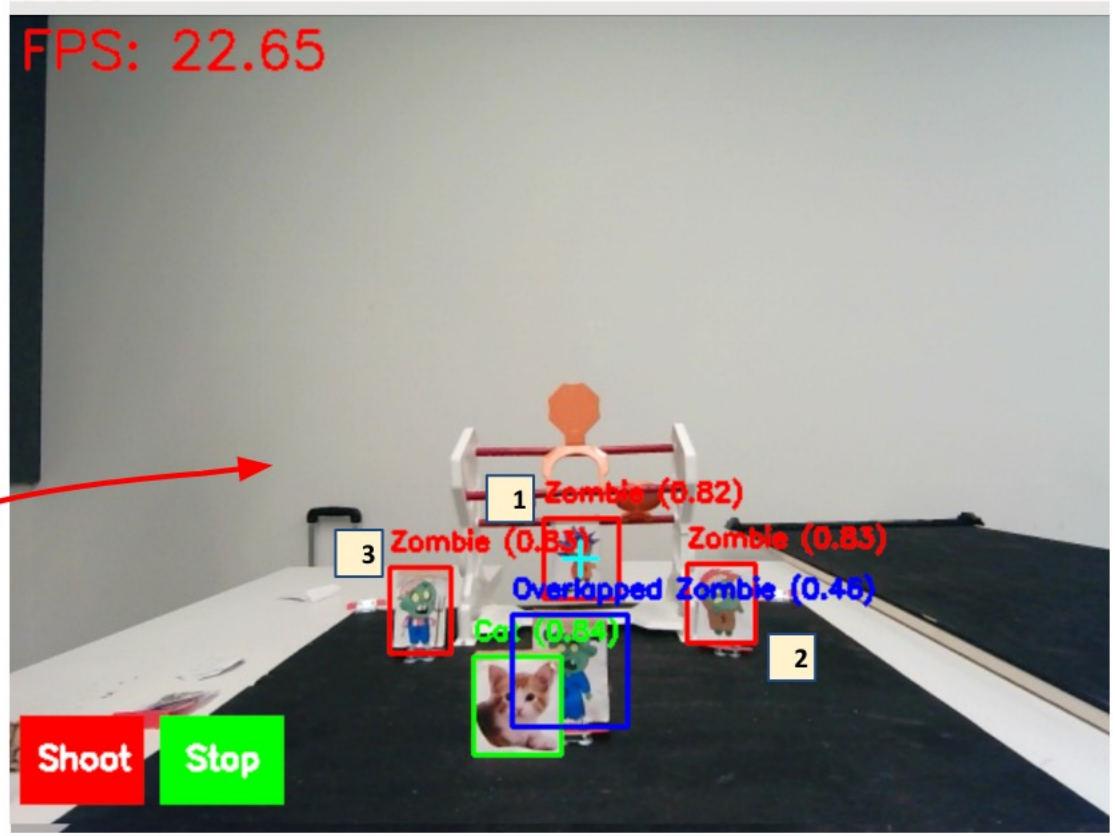
- We aim to:
  - ***Improve the reliability*** of the cannon system
  - Enable ***autonomous prioritization*** of targets
- Key features for enhanced reliability:
  1) ***Emergency Stop*** (GUI button Interaction)
  2) ***Nearest-Zombie*** Targeting
  3) ***Non-overlapping*** targets



< Cannon View (GUI) >

Key Features

- We aim to:
  - ***Improve the reliability*** of the cannon system
  - Enable ***autonomous prioritization*** of targets
- Key features for enhanced reliability:
  1) ***Emergency Stop*** (GUI button Interaction)
  2) ***Nearest-Zombie*** Targeting
  3) ***Non-overlapping*** targets



< Cannon View (GUI) >

Demonstration Scenarios

1) Stationary Target

Shoot (Don't Shoot) Logic
- Confidence score > 0.5
- Cat & Zombie IoU < 0.05

Nearest-Zombie Targeting
- Distance-based prioritization

"Track and ***bring down the moving target for the maximum times***"

2) Moving Target

Servo Control Optimization
- Optimized target tracking
- Simultaneous tracking & shooting

"***Shoot all of the non-overlapping zombies*** but not any of the cats"

GUI Interaction
- Emergency Stop

Shooting with ASR
- Shoot & stop commands
  - *fire, engage / wait*

Performance & Resources
- Frames per second (FPS)
- CPU usage (`htop`)
- GPU usage (`tegrastats`)

ROCAT

Robust Cannon Technology

We develop advanced AI cannon systems built for resilience.
Our technology ensures accurate detection, aiming, and firing with *maximal reliability*,
highlighting *automatic prioritization and engagement* of targets.

Vijay Sai
Advisor

Soo Min Jung
Team Leader

Chani Jung
Systems Administrator

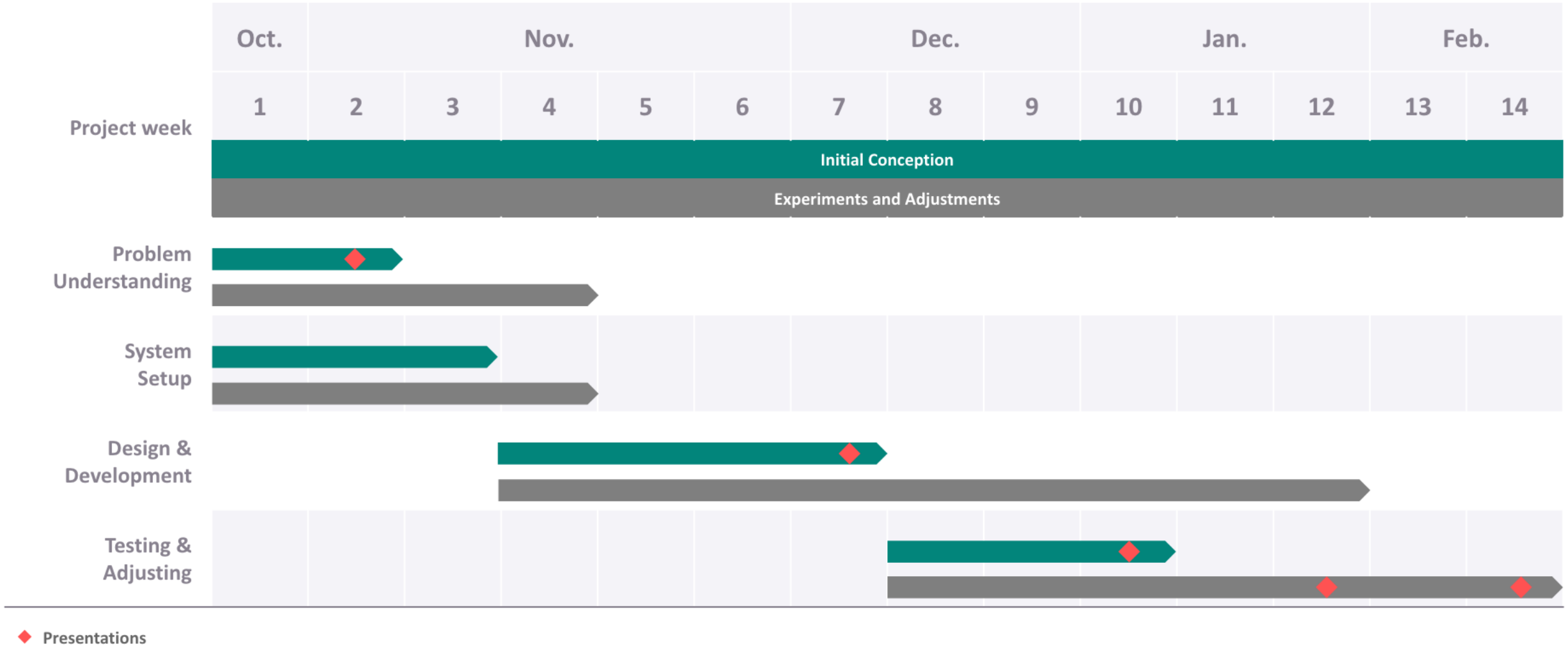
Seungjae Baek
Hardware Engineer

Kyungjin Kim
Process Lead

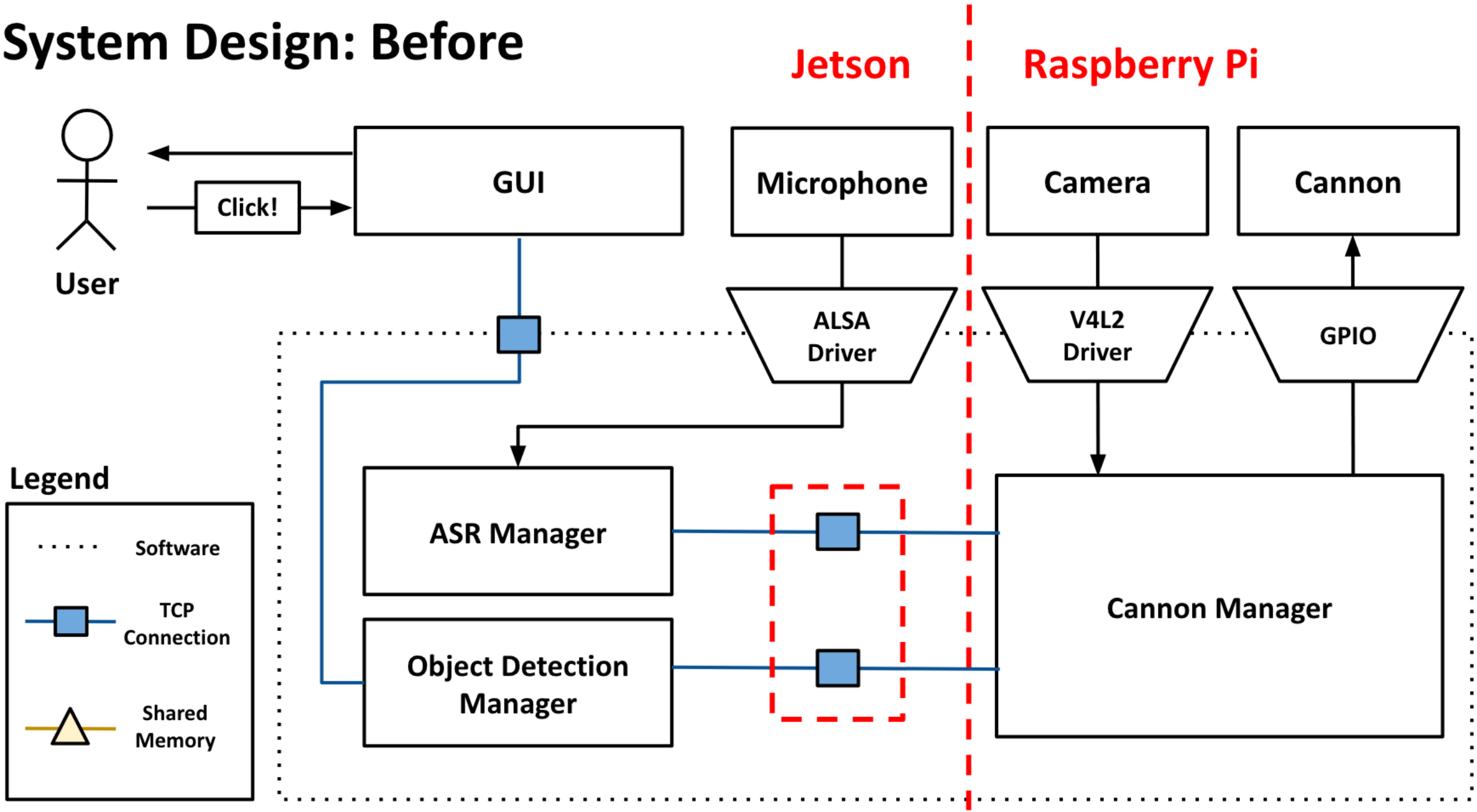
Dongjun Hwang
Integration Lead

Shiwon Kim
Project Manager

Project Timeline

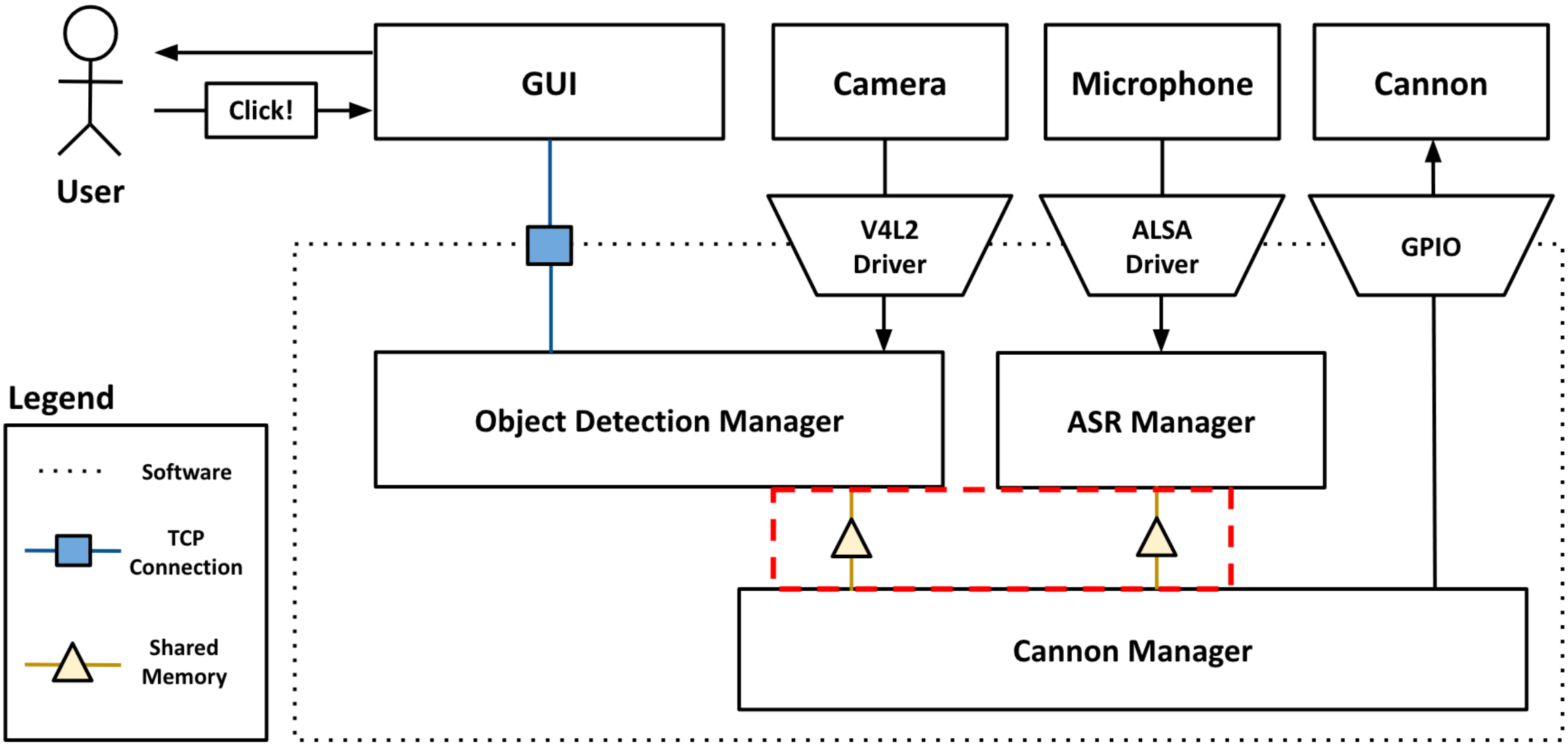


System Design: Before



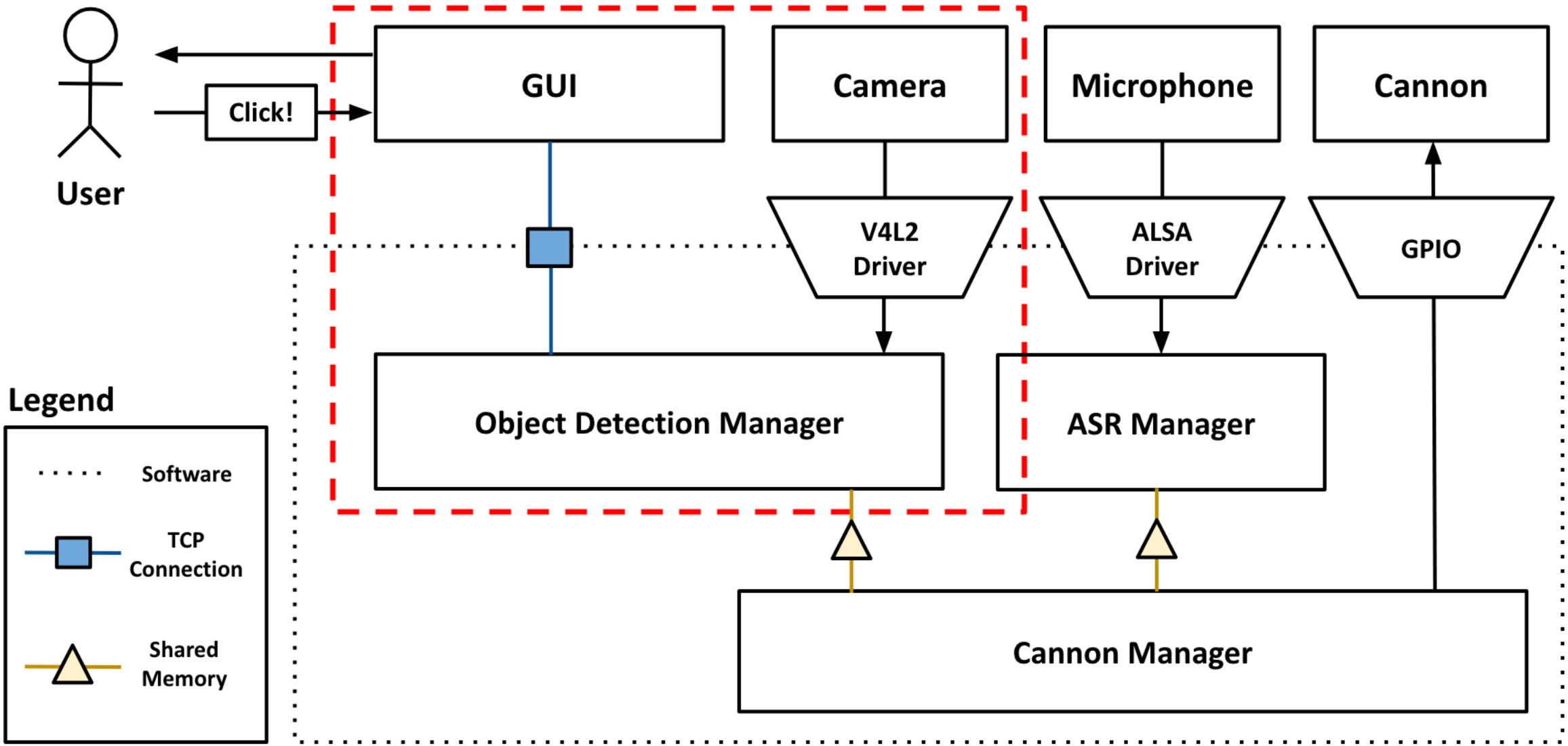
System Design: After

Only Jetson

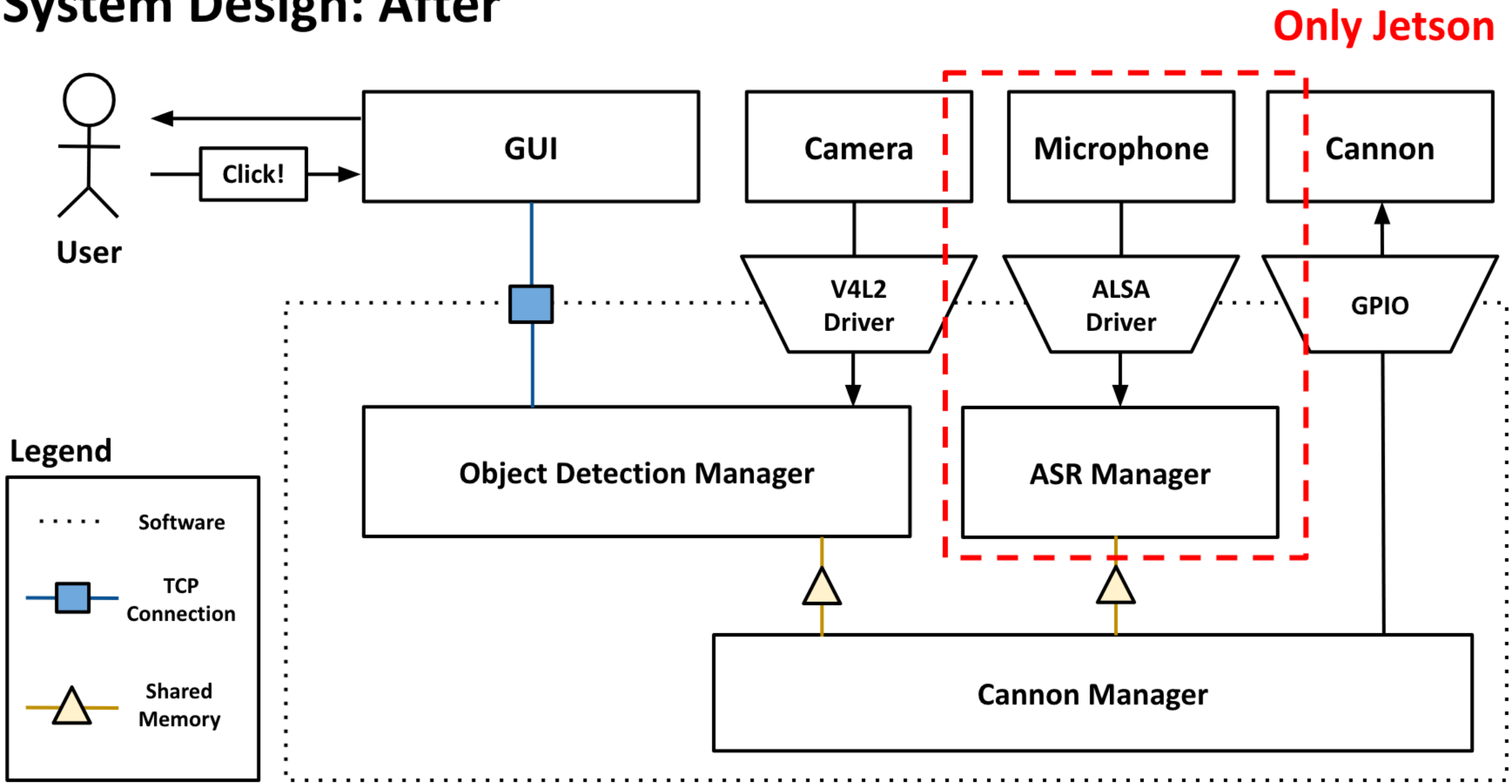


System Design: After

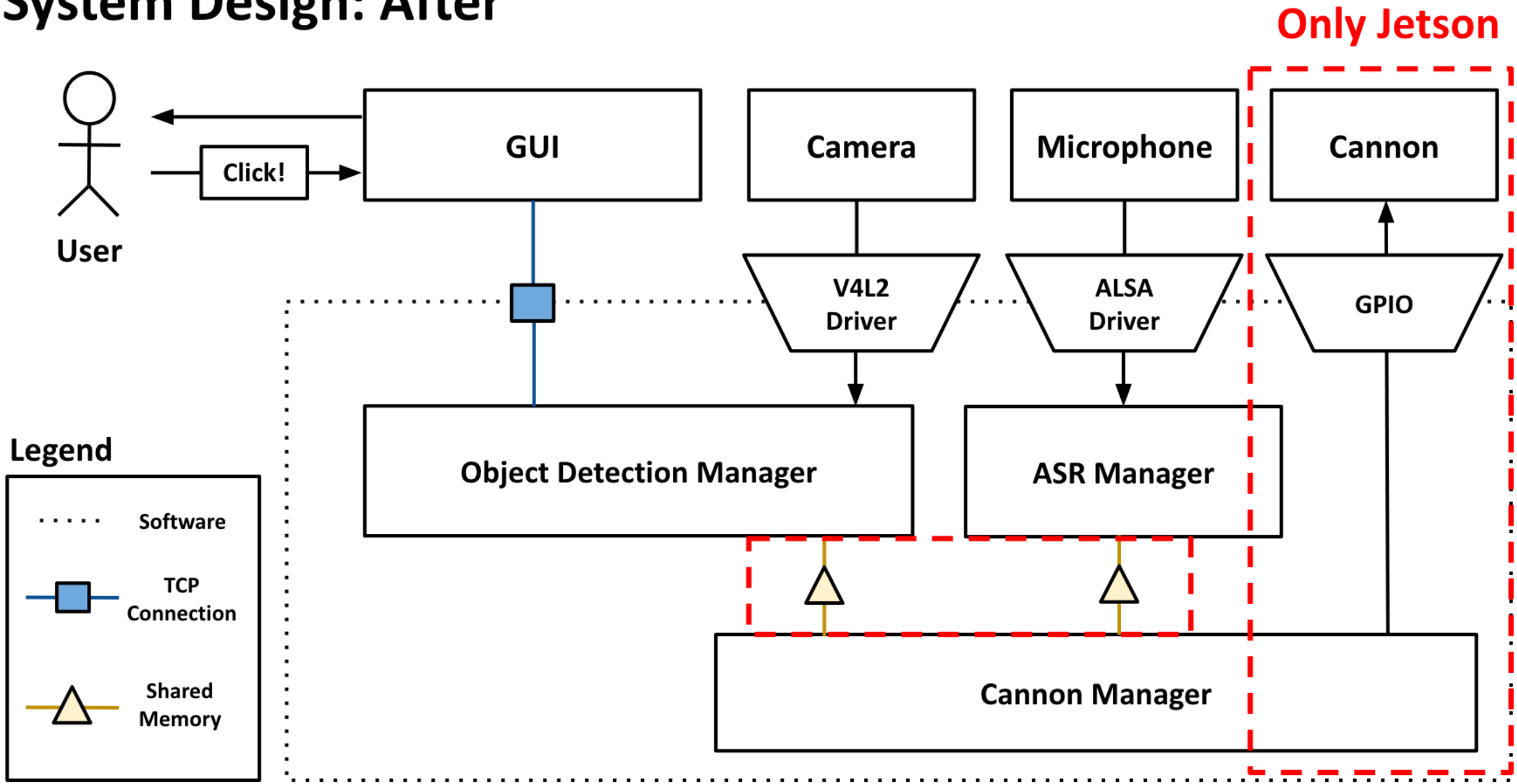
Only Jetson



System Design: After

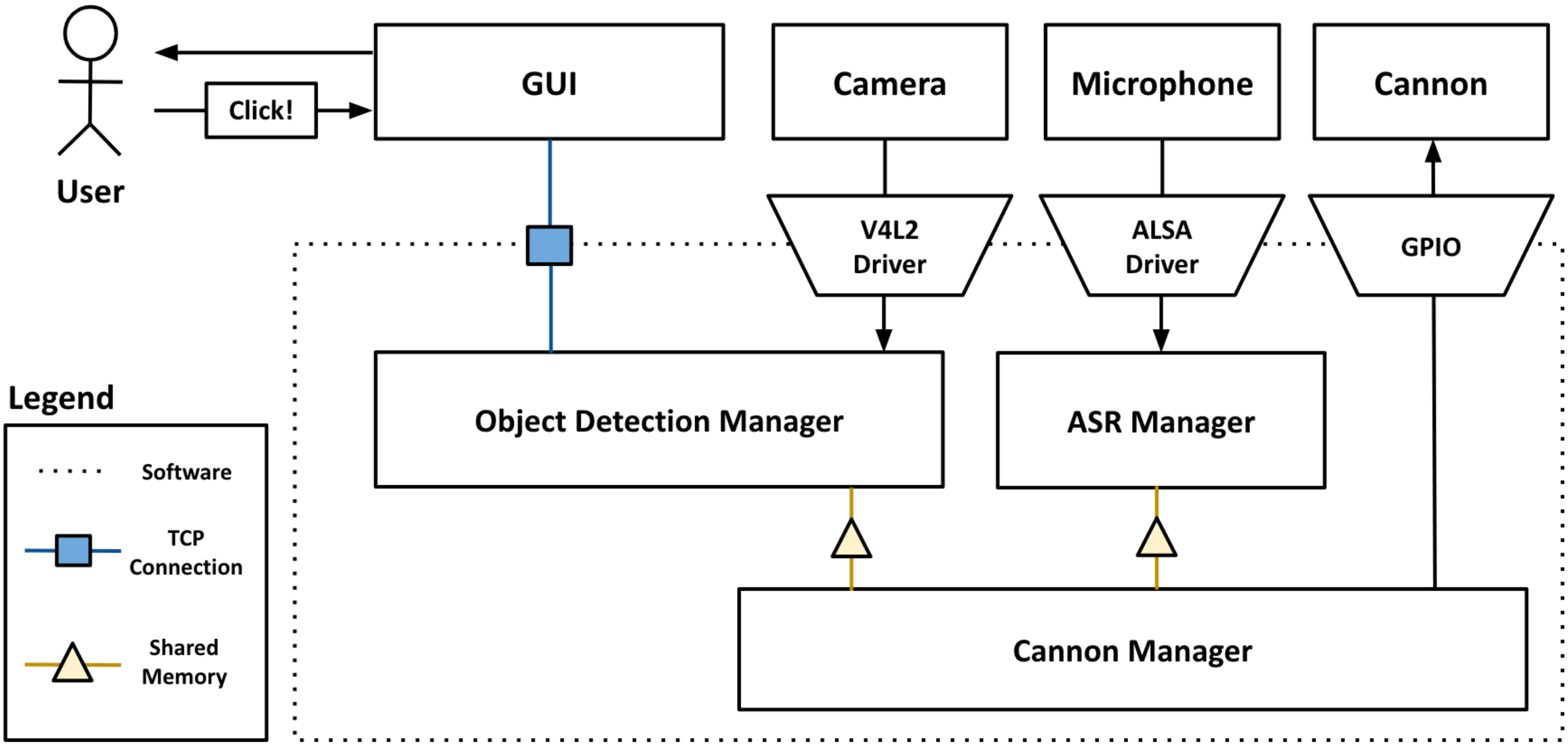


System Design: After



System Design: After

Only Jetson



Quality Attributes

	Scalability	Reliability	Accuracy
Object Detection	• Removed DeepStream usage • Leveraged multiprocessing for easy system replication	• Trained on 500 datasets for robust performance	• Fine-tuned for both static and dynamic targets
Speech Recognition	• Removed RIVA dependency to avoid CUDA version constraints	• Validated robustness with diverse voice inputs	• Better word recognition by discriminating synonyms
Cannon Control	• Hardware independence through elimination of the Raspberry Pi	• Integrated a dual-layer emergency stop system	• Calibrated for precise target engagement

Object Detection

Training Configuration

Model: ultralytics YOLO11

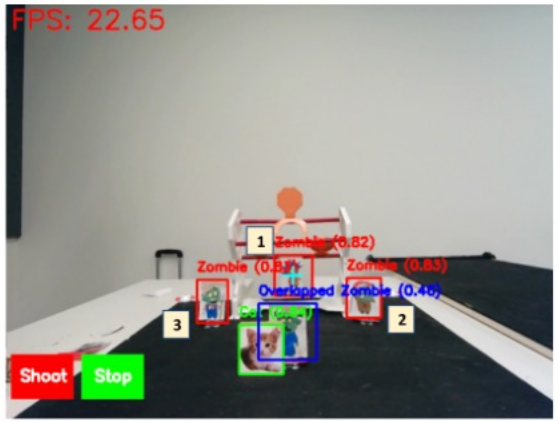
Task:  vs 

Data: 463 raw images from the Cannon (Train:Valid:Test = 7:1:2)

Augmentation:

Target Tracking

Nearest Zombie Targeting



Shoot (Don't Shoot) Logic

🔥 Confidence score > 0.5

	TP	FN
threshold ▼	FP	TN

threshold ▲

❎ Cat & Zombie IoU > 0.05

- **Single Zombie Class Training in Real-World Scenarios**
  - Enables autonomous targeting with priority assessment
- **Efficient Target Tracking via Euclidean Distance**
  - Optimizes movement
  - Minimizes cannon oscillation

Speech Command Recognition

Achieving High Accuracy with an Optimal Command Set

- Google Cloud API
- **Optimal commands selected through speech data analysis**: *fire, engage / wait*
  - Collected 120 voice samples from team members across 10 commands
  - Selected commands with the highest average accuracy

\+ Stopping control assisted by GUI interaction for safety!

Shooting Commands

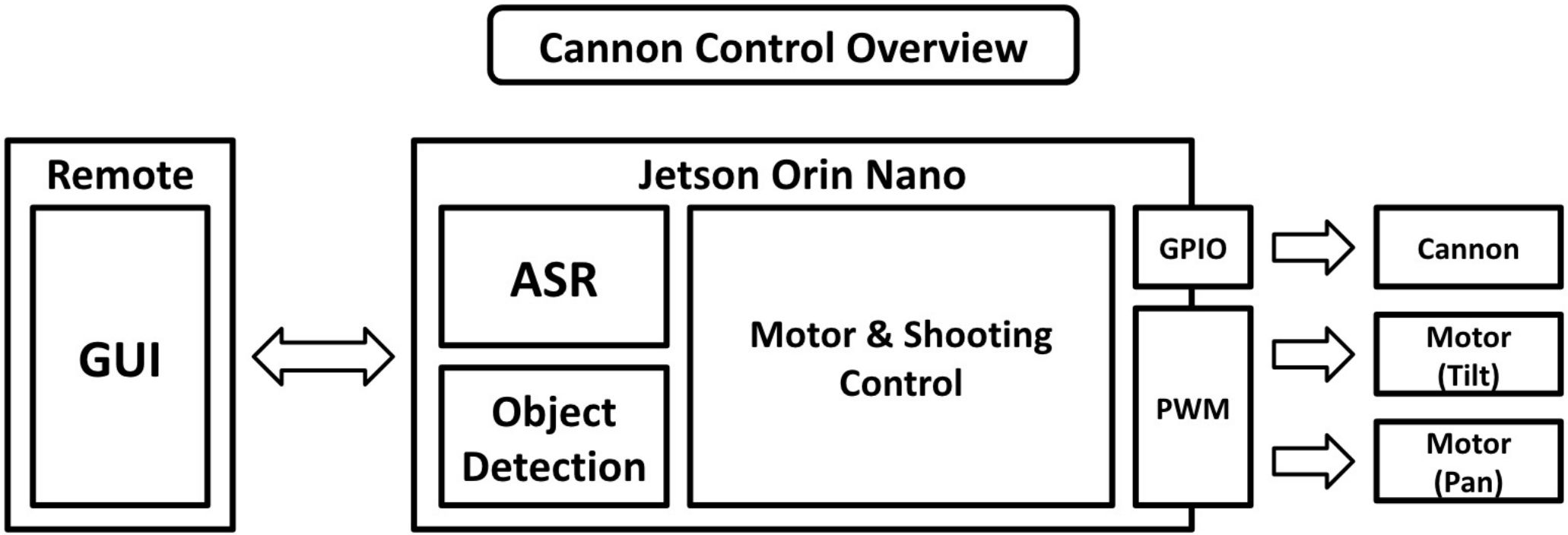
Command	Avg. Acc.	Std. Dev.
engage	0.750	0.274
fire	0.750	0.274
shoot	0.500	0.447
launch	0.333	0.516
attack	0.250	0.418

Stopping Commands

Command	Avg. Acc.	Std. Dev.
wait	0.500	0.548
stop	0.417	0.492
hold	0.250	0.274
pause	0.167	0.258
freeze	0.000	0.000

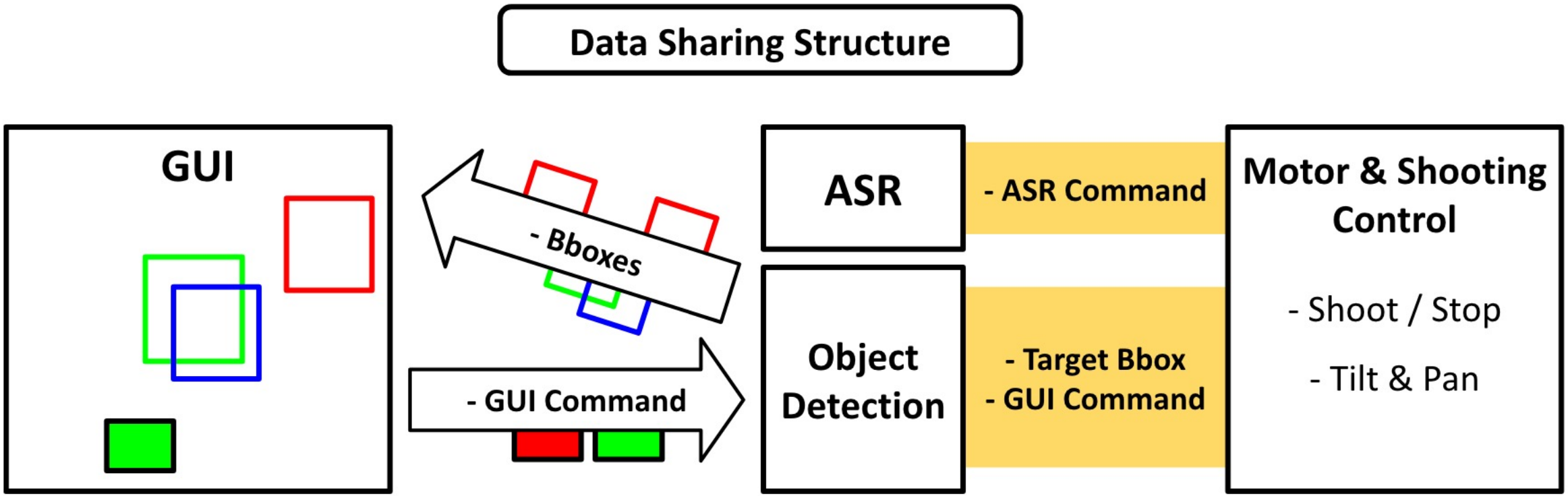
Cannon Control

- **Real-time Data Sharing:** Shared memory synchronizes object bounding boxes, GUI, and ASR commands.
- **Dynamic Targeting:** Pan and tilt adjust based on the target's bounding box center.
- **Hardware Simplification:** Powered cannon directly via Jetson's GPIO and a transistor, removing RPi.



Cannon Control

- **Real-time Data Sharing:** Shared memory synchronizes object bounding boxes, GUI, and ASR commands.
- **Dynamic Targeting:** Pan and tilt adjust based on the target's bounding box center.
- **Hardware Simplification:** Powered cannon directly via Jetson's GPIO and a transistor, removing RPi.



Cannon Control

- **Real-time Data Sharing:** Shared memory synchronizes object bounding boxes, GUI, and ASR commands.
- **Dynamic Targeting:** Pan and tilt adjust based on the target's bounding box center.
- **Hardware Simplification:** Powered cannon directly via Jetson's GPIO and a transistor, removing RPi.

Mode-Specific Config

	Stationary	Moving
Fire while Tracking	X	O
Fire Duration	50ms	400ms
SMOOTH_FACTOR*	70	30

ΔTilt = (Tilt - TiltError) / SMOOTH_FACTOR;

ΔPan = (Pan - PanError) / SMOOTH_FACTOR;

*Constraints Maximum Angle Adjustment (Step) for Aiming Stability

Cannon Control

- **Real-time Data Sharing:** Shared memory synchronizes object bounding boxes, GUI, and ASR commands.
- **Dynamic Targeting:** Pan and tilt adjust based on the target's bounding box center.
- **Hardware Simplification:** Powered cannon directly via Jetson's GPIO and a transistor, removing RPi.