

Advanced Vehicle Perception System

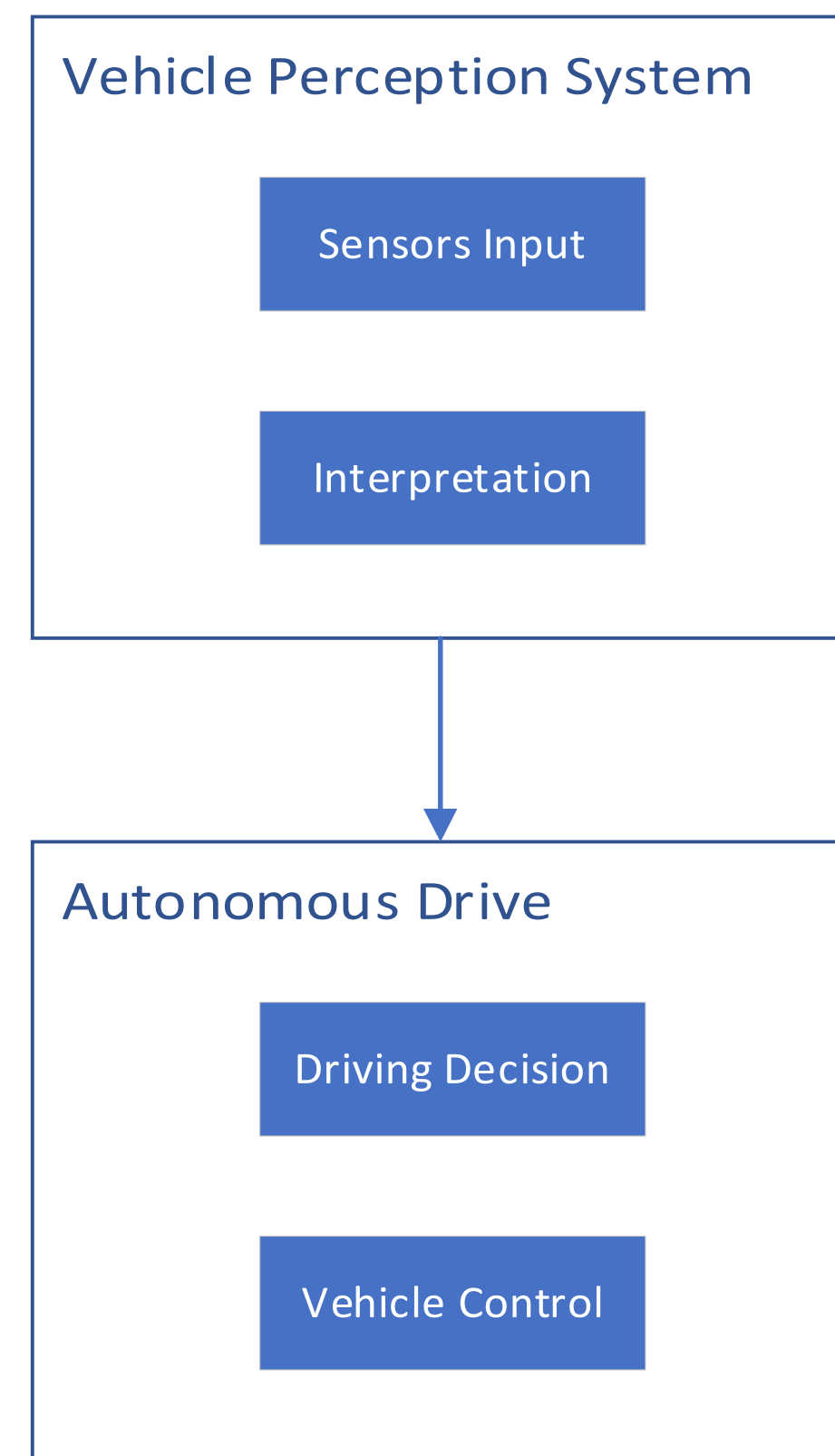
Centre for Mechatronics and Hybrid Technology

Department of Computing and Software Engineering, McMaster University

Ash (Chang) Liu

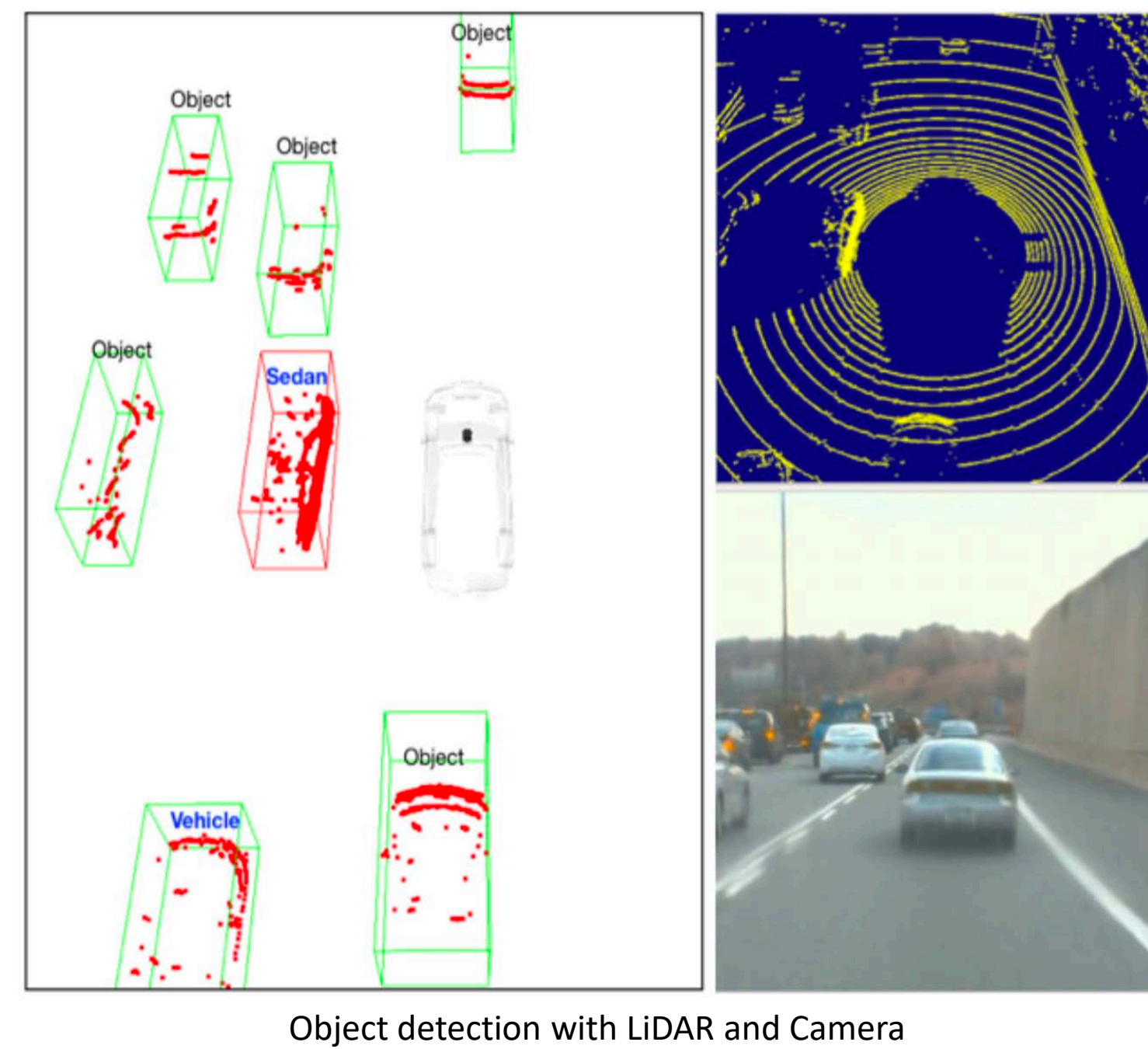
EECOMOBILITY (ORF) &
HEVPD&D CREATE

VEHICLE PERCEPTION SYSTEM

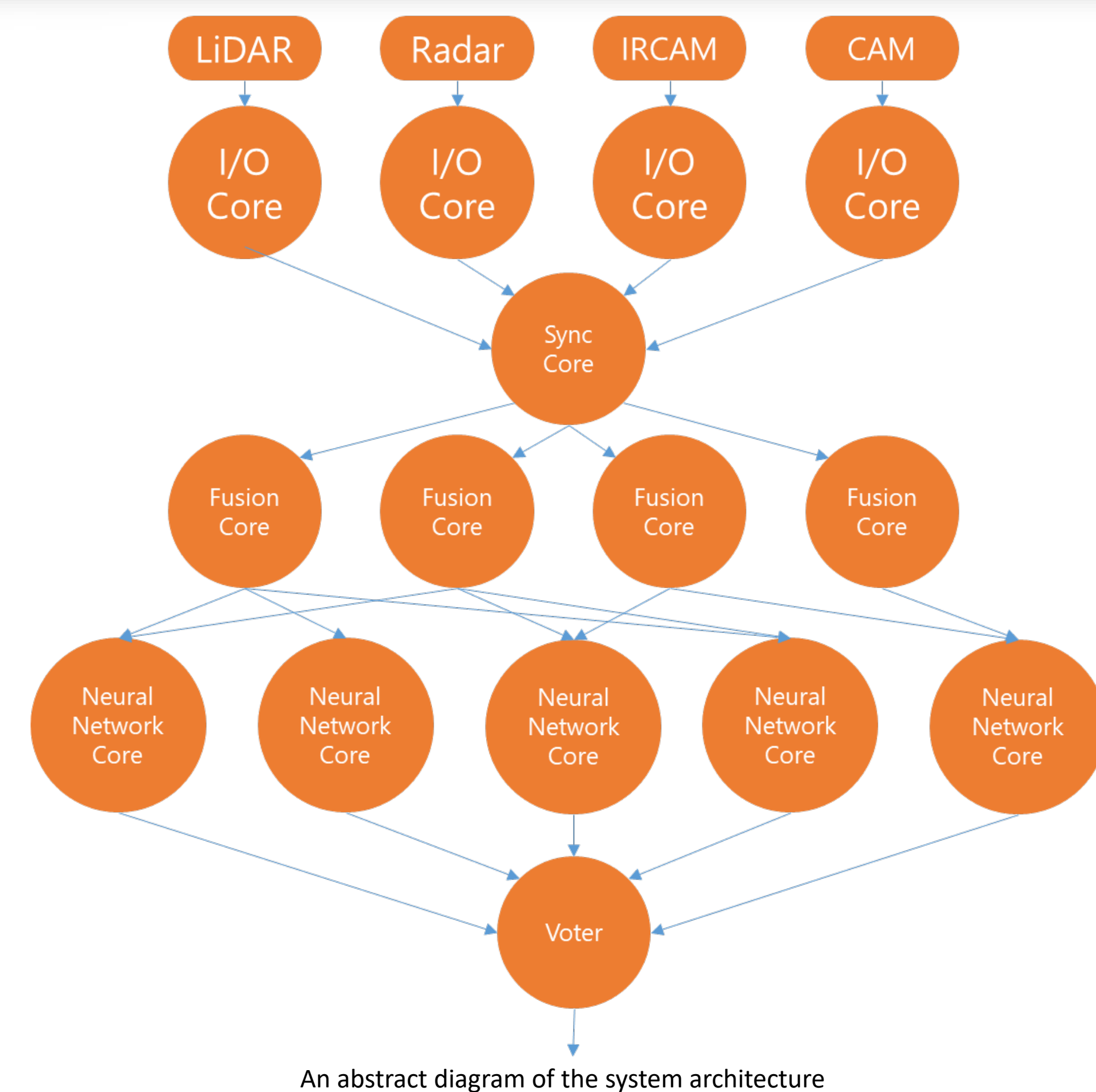


Just like our human needs to see, smell, hear, and feel to gather knowledges about the surrounding before we make decisions; an autonomous drive vehicle needs sensors to collect information from the surrounding, then extract information for the autonomous drive system to make any reasonable driving decision.

Taking raw information from the sensor is not a trivial problem: different type of sensors have different properties, and in some cases sensors can produce false information. Therefore a systematic approach to the sensor inputs is necessary to ensure information collected is accurate, and reliable.



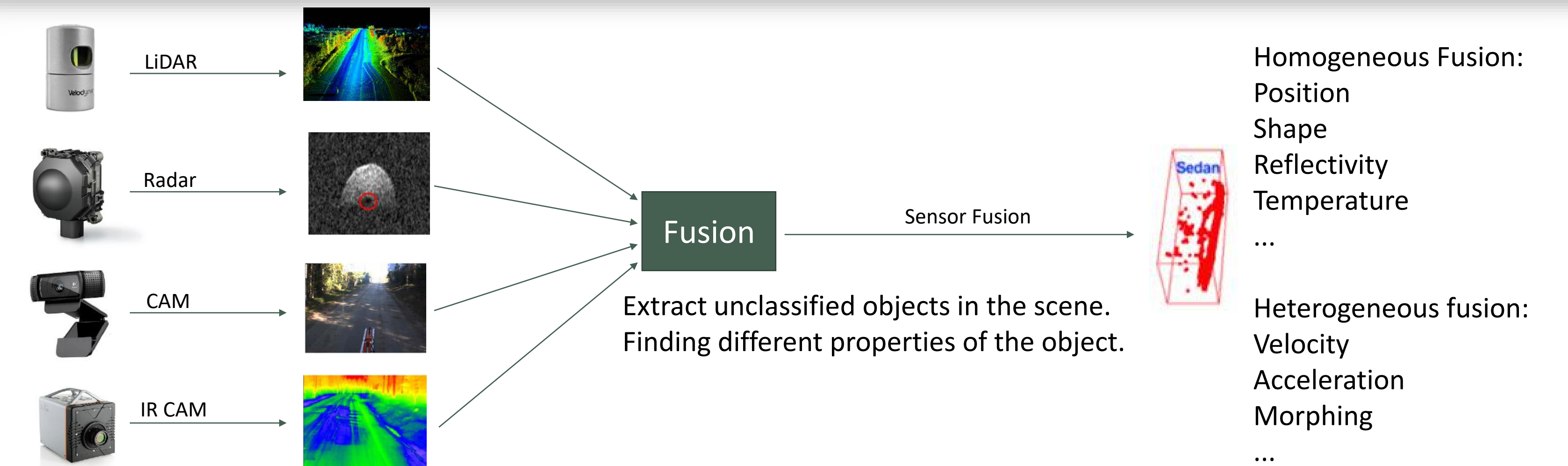
SYSTEM ARCHITECTURE



An abstract diagram of the system architecture

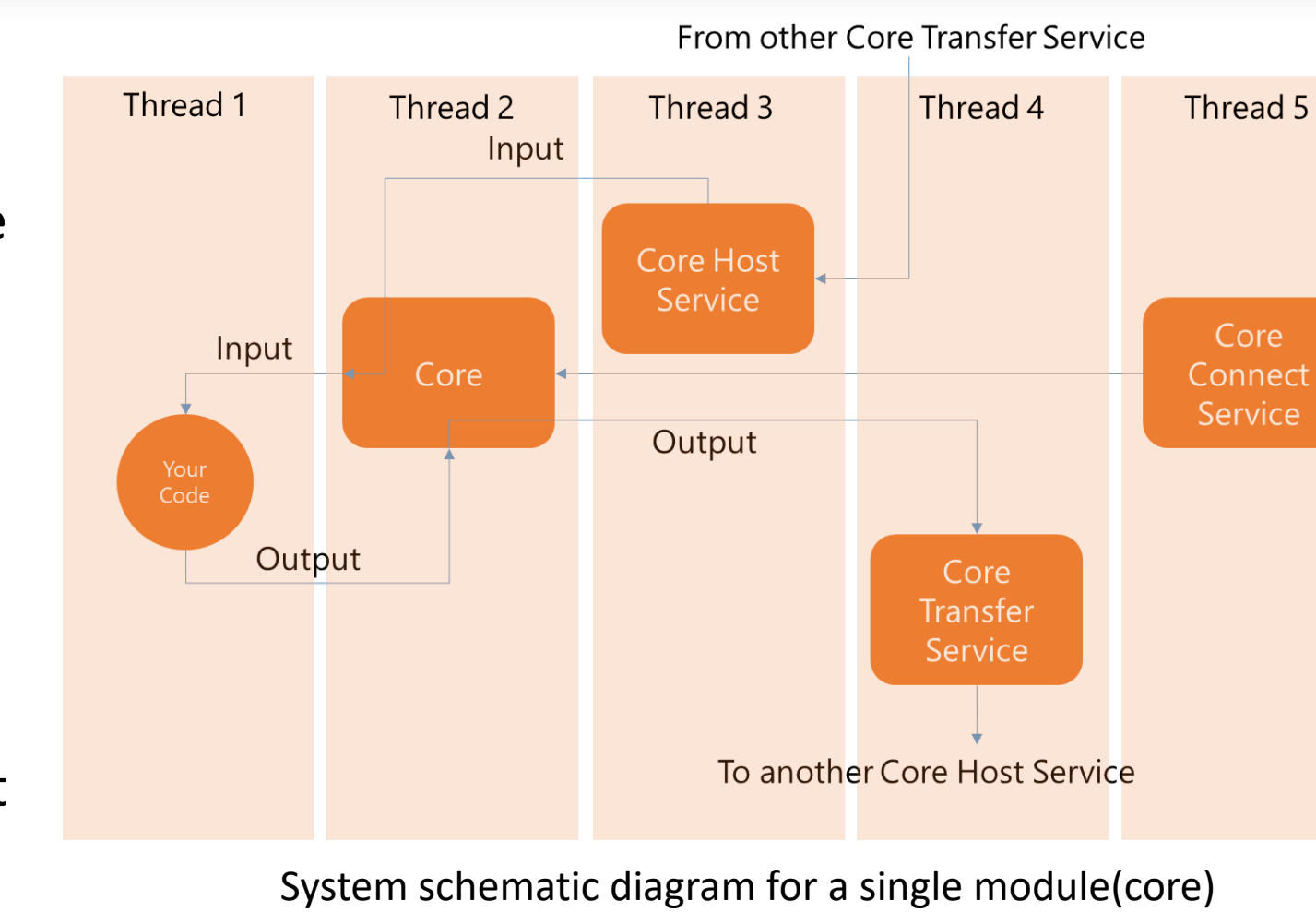
- Using modified classic blackboard architecture for decision making.
- A modulate design to allows us split works between different researchers, and running different module on different platforms. (Distributed System)
- System contains 4 layers: Input layer (I/O Cores & Sync Core), Fusion layer (Fusion Cores), Logic layer (Neural Network Cores), and Decision Layer (Voter).
- Input layer takes input from different sensor, and sync core synchronize the input.
- Fusion layer combines data retrieved from input layer to create a mix of unclassified objects, and assigning extracted information to the object. (Sensor fusion)
- Logic layer will take information from the fusion layer to perform classification and recognition.
- Decision layer took results different logic units (Neural Network Core) and determine what is the most accurate description of classified object.

SENSORS FUSION



DISTRIBUTED SYSTEM

Centralized system often faces the issue with scalability: the more sensors and modules you attached to the system, the longer response time it requires for the system to response, thus this directly reduce the sampling rate for the system and downgrade the overall system performance. To solve this issue, we used a decentralize design for our system, and developed our own version of distributed system package to support this architecture.



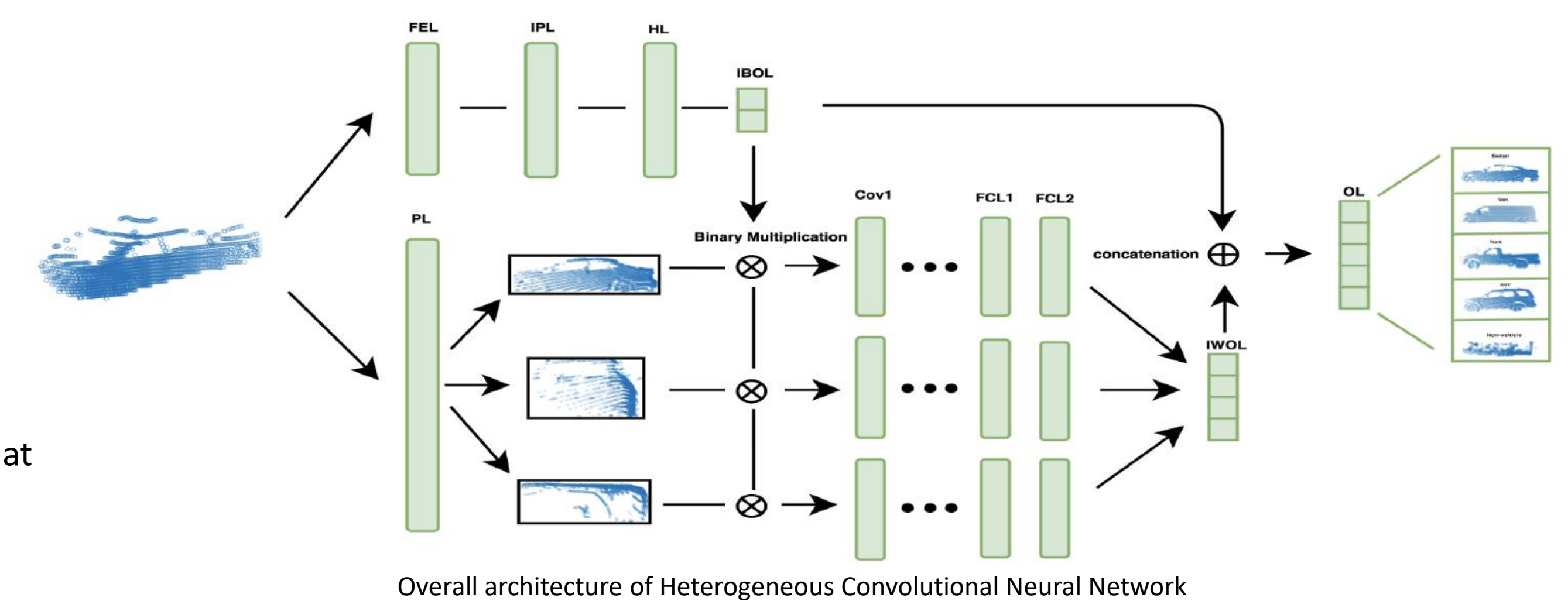
System schematic diagram for a single module(core)

Our software package is a C++ library package based on TCP/IP network interface to communicate between modules.

- Advantages:
- Hot plug module while system is running
 - Multi-platform/ multi-OS support
 - Vanilla.(No additional 3rd party library dependency)
 - Simple API
 - Real Time Support

HETEROGENEOUS CONVOLUTIONAL NEURAL NETWORK

Heterogeneous Convolutional Neural Network (HCNN) is a Deep Neural Network architecture we developed for vehicle classification and recognition. This architecture combines feature extraction with CNN to creates a robust neural network array. From our testing with collected road data, this neural network delivers superior performance at 94% precision.



Overall architecture of Heterogeneous Convolutional Neural Network