Requirement Collection and Analyses Document For *HUEBERT*

Matthew Ebert and Jamieson Fregeau

ECE 356, UVic

2021-10-11

Glossary

RP	Remote Player
LP	Local Player
OTS	Off The Shelf
HUE	The local system of HUEBERT which connects to a network and accesses the game environment and the local players
BERT	The remote system of HUEBERT which interfaces with the RP and send control instruction to HUE.
THORP	A system that connects to HUE and interacts with the physical pieces of the game environment. THORP **is not** included in the scope of this product

Introduction

This document completes all tasks outlined in “ECE 356 Requirement Collection & Analysis Document”.

Any word I*talicized* **or** CAPATALIZED **refers to Appendix A.**

1. Product Scope

HUEBERT will be designed as a framework to allow a remote player to play table-top games with local players. Over a network, a remote user can connect to and control a local robotic device. The local system, HUE, will have sensors and video to give the remote player feedback and relay the status of the game at any point. HUE will interact safely and non-disruptively with the local players such that normal gameplay can ensue.

The scope of this project includes a remote system, BERT, including a user interface, user input, and network communication, and a local system, HUE, composed of a network communication system, control and computation system, actuators and sensors, and a physical mechanical game interface, THORP. The details and abilities of each system must, at least, operate with base functionality to perform the requirements listed in this document.

An important definition of this product is that HUEBERT **does not** include the development of a THORP. THORP will be a separate series of product which HUEBERT will be designed to integrate with. HUEBERT **does** include the systems HUE and BERT.

A. Requirements

FUNCTIONAL	R01	The *Frame* must be able to move as instructed by the remote user within HUE's range with a precision of +/- 1mm.
	R02	HUE must have a minimum range of 400 mm.
	R03	The data and signals sent as RP *Input/Output* must relay accurate and complete information.
	R04	The user must have controls which allow instructions to be sent to the local system (HUE).
	R05	The HUEBERT system must support modular addition and configurability.
NON-FUNCTIONAL	R06	The *Frame* must move at a speed comparable to that of a human. The interaction speed of the manipulator must be at least 100mm/s.
	R07	The noise created at any point by HUEBERT must not exceed a certain decibel level. The HUE must create noise less than 70 decibels at any time and less than 60 decibels at rest.

	R08	The *RP Input* must be efficient and allow the user to respond to game play in real time. The response delay for any user single input must be less than 500ms.
	R09	All information sent to the RP from the HUE must have a latency of less than 150ms.
	R10	All electrical connections and circuits should be shielded from the LP's and RP.
	R11	No mechanical motion should occur without instruction from a user.
	R12	The actuators torque and speed should not be more than what is required for gameplay.
	R13	The RP's personal electronic data and identity must be secured and inaccessible by any unauthorised user or component of the HUEBERT system
	R14	All local networked devices must be secure from any attempted infiltration of the HUEBERT network device.
	R15	The HUEBERT system must be compatible with a variety of additional OTS products including Arduino, ESP32 and related components.
	R16	The HUEBERT mechanical and electrical components must support modification, replacement, and additions including additional sensors; additional 3D printed structural components; additional actuators.

B. RACI Matrix

The following matrix describes the roles of each of the product stakeholder who are defined as the following,

Stakeholder	Assigned
Client	Matthew Ebert
Product Owner	Jamieson Fregeau
Tester	Matthew Ebert/Jamieson Fregeau
Developer	Matthew Ebert/Jamieson Fregeau

	RACI MATRIX	Responsible	Accountable	Consulted	Informed
		ROLES			

#	TASKS	Developer	Tester	Product Owner	Client
1	Define product purpose, scope, environment, and users	I	I	C	R/A
2	Elicit Functional and Non-Functional Requirements	C	C	A	R
3	Research potential design and solutions	R	I	A	I
4	Define Hardware/Software/Communication Interfaces	R	R	A	I
5	Define User Interfaces	C	I	A	R
6	Asses Security and Safety Risks	C	I	R	I
7	Create and Rank User Stories and Scenarios	R	R	R	R
8	Create Project Documentation	C	C	R	A
9	Acquire and Develop Hardware/Software/Communication Systems	R	C	A	I
10	Test Hardware/Software/Communications Systems	I	R	A	C
11	Integrate Hardware, Communication, and Software systems	R	C	A	I
12	Test Integrated Product	I	R	A	C
13	Create User Documentation	C	C	R/A	I
14	Review of Product	A	C	R	I
15	Acceptance of Product	I	I	A	R
16	Release of Product	I	I	A	R
17	Ongoing Maintenance and Updates	R	R	A	C

C. Requirement Elicitation Techniques

Multiple elicitation techniques were utilized to determine the HUEBERT system's functional and non-functional requirements. The following techniques were found to be most effective:

- **Observation:** HUBERT's functional requirements were determined through the process of active observation. During this phase, an individual enacted all the necessary features that HUEBERT requires to effectively complete the scope of this project. During this process, an analyst observed the individual performing these tasks, stopped them to ask specific questions and determined the fundamental requirements for the HUEBERT system.
- An example of this technique in use is the following. An individual sits at one end of a chess board and grabs a bishop. The individual then picks the bishop up and moves it to the end of the board. In this scenario, the analyst determined that HEUBERT would be

required to move to the location of the bishop, to a precision that enabled it to interact with the piece. Then HEUBERT would have to be able to have a range that encompassed the entire game board and move in a path to this end location that did not disrupt the game environment.

- **Interviews:** HUBERT's non-functional requirements were determined through interviews of fellow engineering students. In these interviews, requirements such as HEUBERT's movement speed and noise levels, remote latency, security and safety hazards were determined. After the 4 interviews, the results were compiled, and exact non-functional requirements were agreed upon by client and product owner.
- An example of this technique for determining the minimum speed of the robot arm is as follows. In four interviews, the subjects were given the option to choose a minimum operating speed of either 50 mm/s, 75 mm/s, 100 mm/s and 125 mm/s. Each subject was given a demonstration of what these speeds resembled. 3 of the subjects chose 100 mm/s and 1 subject chose 125 mm/s. The client and product owner consulted on these results and chose to utilize 100 mm/s as a minimum operational speed.

D. User Stories and Ranking

The following table give User stories and Scenarios for each requirement. The ranking of each user story is found in **section H.**

R	*S*	*User Stories*	*Scenarios*
R1	S1	As a remote user, I want to control HUE's position such that I can pick up small game pieces and place them accurately within the game environment.	-Given a working HUEBERT system -When the RP sets a desired position for HUE -HUE moves to that position
	S2	As a remote user, I want to control Hue's motion such that I can dictate its path through space.	-Given a working HUEBERT system -When the RP sets a desired position for HUE and dictates a path for HUE to follow -Then HUE moves in accordance with the path to the desired locations
R2	S3	As a remote user, I want HUE to be able to access all areas of a standard sized game board (400 mm arm range).	-Given the HUE system and a *Game Environment* -When the RP asks HUE to the access the farthest end of the board -Then HUE can comply with the request
R3	S4	As the RP, I want to see and understand the Game Environment	-Given a *RP I/O* -When the RP requests sensor or video input from the *Game Environment* -Then the information is complete enough for the RP to understand the game status.
	S5	As the RP, I want to send instruction to HUE.	-Given a HUEBERT system and a Network connection -When the RP wants to send certain instruction to HUE and HUE is capable and allowed to follow the instruction -Then HUE can receive the enact the instruction from the RP
R4	S6	As the RP, I want to have controls that control HUE.	-Given BERT and a Network Connection -When the RP has a desired action for HUE

			-Then the RP can activate that action through the BERT controls over a network
R5	S7	As HUE, I want to be compatible with different THORP modules so that I may play a variety of games	-Given a HUE system and appropriate power -When the LP tries to install a THORP module -Then the THORP can easily be connected and integrated with HUE
	S8	As BERT, I want to be compatible with multiple RP control devices so that I may control different THORPS	-Given a BERT system and appropriate power -When a RP tries to install a control module for BERT -Then the RP can easily connect and integrate their controls with the BERT system
	S9	As the RP and LP, I want to equip different THORP and BERT add-ons to play a variety of games	-Given a HUEBERT system and a variety of THORP and BERT add-ons -When a player tries to modify and configure HUEBERT for different games -Then the player can do so quickly, safely, and without to much work.
R6	S10	As HUE, I want to be able to move at an expected and enjoyable speed for a user.	-Given a HUE system and a movement instruction -When HUE tries to move as instructed -Then it moves at reasonable speed reaching up to at least 100mm/s
R7	S11	As the LP, I want HUE to be quiet so that I can converse and play with other LPs without major disruption.	-Given an operational HUE system -When it acts on the control information sent to it from the RP -Then it should move at less than 70 decibels as to not disrupt the local game
R8	S12	As the RP, I want HUE to react to my inputs in a timely manner such that I can reasonably play my game of choice.	-Given an RP operating HUE -When they send control information through the networked connection to HUE -Then HUE should react accordingly in less than 500ms.
R9	S13	As the RP or LP, I want to be able to react to the feedback I receive from HUE in a timely manner without inhibiting delays.	-Given a HUE system in operation -When it is the systems turn to act -Then it should do so without large delays between information being sent and/or multiple movements.
R10	S14	As the RP or LP, I want to be safe when utilizing HUE, BERT or THORP and not be concerned about electrical hazards during gameplay.	-Given a functional HUEBERT system -When any player is interacting with the system -Then they should not be concerned about the electrical safety of the system or of any sparking or fires of occurring.
R11	S15	As the LP or RP, I want to safe when utilizing HUE, BERT or THORP and not be concerned about physical hazards during gameplay.	-Given a functional HUEBERT system -When any player is interacting with the system while in operation -Then they should not be concerned of the system harming them or being able to get caught any physical components
R12	S16	As an LP or RP, I want to feel and be safe around HUE, BERT, or THORP when they are in motion.	-Given a functional HUEBERT system -When any player interacts with the system while it moves

			-Then they should not be concerned about the system hitting them while in motion
R13	S17	As an RP, I want to choose what files or data is available to HUEBERT, THORP and LP's.	-Given an RP -When they connect to the HEUBERT system over the networked connection -Then their local data/files are not automatically available to the network
R14	S18	As an LP, I want to keep my private devices and IoT systems safe from malicious RP users.	-Given an LP -When they connect to the HUEBERT system with their device -Then their devices are not exposed to the network and are well protected from malicious intent
R15	S19	As THORP or BERT, I want to integrate with HUE seamlessly through industry standard processor or code.	-Given the THORP or BERT systems -When a user wants to operate the HUEBERT system and play a remote game -Then the HUE system responds as expected in accordance with industry standard
R16	S20	As a player, I want to create and modify my HUEBERT system so that I may increase the variety of playable games and the performance.	-Given a functional HUEBERT system -When a user wants to play a different variety of game -Then the HUEBERT system can easily be modified to fit my game environment of choice

E. Design Decisions

Design decisions were made based on the quality requirements.

1. Computing Hardware

To comply with the modularity and modifiability requirements, only Arduino, Raspberry Pi, and ESP32 will be used in the HUEBERT system for purposes of GPIO, computing, and network communications.

2. Frame and Fastening Hardware

The *Frame* will be comprised of or compatible with 3D printed components, NEMA mounting, metric fasteners, metric bearings, and metric axles.

3. Server Client Networking

A server client networking system will be used to comply with security requirements

4. RP Device Requirements

For communication, interface, and networking requirements, the RP needs a device such as a laptop or PC to work with the BERT module. The device must connect to a server, have a display, run the BERT program, and a USB input.

5. Code

Because of the computing hardware used, C++ and Python will run the HUE and BERT systems in compliance with Arduino and Raspberry Pi systems.

Other design decisions are implied in Appendix A

F. Safety and Security Hazards

Following the elicitation of the requirements, some potential hazards, risks, and threats were identified.

1. Safety Hazards

The following safety Hazards of HUEBERT were identified.

1. Electrical Hazards: The electrical supply and distribution of HUEBERT has the potential to cause harm to the user from the following sources.
    a. Exposed wire leads.
    b. Exposed circuits.
    c. Short circuits.
2. Mechanical Hazards: the motion of HUE and/or BERT has the potential to inflict injury on the users by
    a. Moving quickly and impacting a user's body
    b. Pinching the user with HEUBERT's mechanical joints
3. Heat Hazards: should HUEBERT be used for a sufficient period without proper cooling the following sources could burn the users.
    a. Stepper/DC/Servo Motors
    b. Motor drivers
    c. Exposed Processors
4. Noise Hazards: the actuator and/or sensor within HUEBERT may produce excessive noise. Should this noise be greater than 85 decibels damage to the users hearing could occur.
5. Device Safety: The HUEBERT system will directly connect multiple devices (user and system). A catastrophic failure of one device has the potential to damage or cause failure to another. For example, an electrical short in the BERT *RP Input* could cause an electrical surge through the RP's laptop and damage the components.

2. Security Threats

The following security threats of HUEBERT were identified.

1. RP data security: Since BERT's software is installed on a user device, BERT may have the ability to access the user's system and their data.

2. RP and LP network: HUEBERT must have access to both the RP and LP network to operate; Consequently, HUEBERT may be able to infiltrate either network
3. RP and LP device security: HUEBERT requires RP personal device and could require an LP personal device for certain functionality; Consequently, HUEBERT may have the ability to infiltrate nonauthorized systems of these devices
4. Authentication Security: HUE is controlled remotely over a network. There may be potential for an unauthorized user to connect to and control HUE.
5. Privacy: Since multiple user device may connect to HUEBERT, a malicious user might have the ability to access other user device through the HUEBERT system.

Both the safety hazards and security threats should be addressed and mitigated by any design of the HUEBERT product.

G. Risk and Threat Analysis

Understanding the risks and potential threats to the HUEBERT system is a fundamental design aspect that needs to be considered. The following figures describe the potential threats to the HUEBERT system, as well as standard mitigation techniques, from three differing points of view .

1. HUE System

1. **Malicious Software from BERT-HUE**: Should a malicious user gain access to BERT's local system and inject a bug into HUE's input stream, it is vital the HUE system has virus protection software in place.
2. **Malicious Software from Network-HUE**: To mitigate a malicious user from injecting a bug into the data stream from the network to HUE, it is vital that HUE verifies the source of all input data.
3. **Interception of Video Feed**: A malicious user could intercept data over the HUE-BERT connection NETWORK and view the live data stream from the RP and LP. This can be mitigated by properly encrypting the data stream to the regulated standard.
4. **Breach Local Network:** A malicious user sending data from BERT could use the connection to gain access to the LP's local network. A mitigation technique is to limit the access BERT must strictly the local game environment and prevent access to other devices or the local network.

2. BERT System

1. **Local Network Access:** Should a malicious user gain access to BERT's local network, the BERT system must be password protected to ensure they cannot pass malicious data on to HUE.
2. **Sniffing Attack:** To mitigate a user from intercepting the live data feed being sent from BERT to HUE, the networked connection must be encrypted, and the source must be verified.
3. **Hostile Takeover of HUE:** If a malicious user were to bypass HUE's security measure and gain access to the LP's system, it is vital that the LP has no ability to access the RP's system through the networked connection.

4. **Bug Injection:** Should a user gain control of HUE and pass a bug into the output, it is vital that the BERT system includes virus detection software to prevent the malicious user from passing a virus onto the BERT system.

3. Network

1. **Access Server Data:** To prevent a malicious user from accessing the server and all corresponding data it is vital that any user requires formal private-key certification to access server-based data.
2. **Session Hijacking:** To mitigate the ability of a user to hijack the networked connection, this connection must also require a private-key certification. Furthermore, the data must be encrypted to industry standard to that even if all security measures are bypassed a malicious user still must decrypt the data stream.

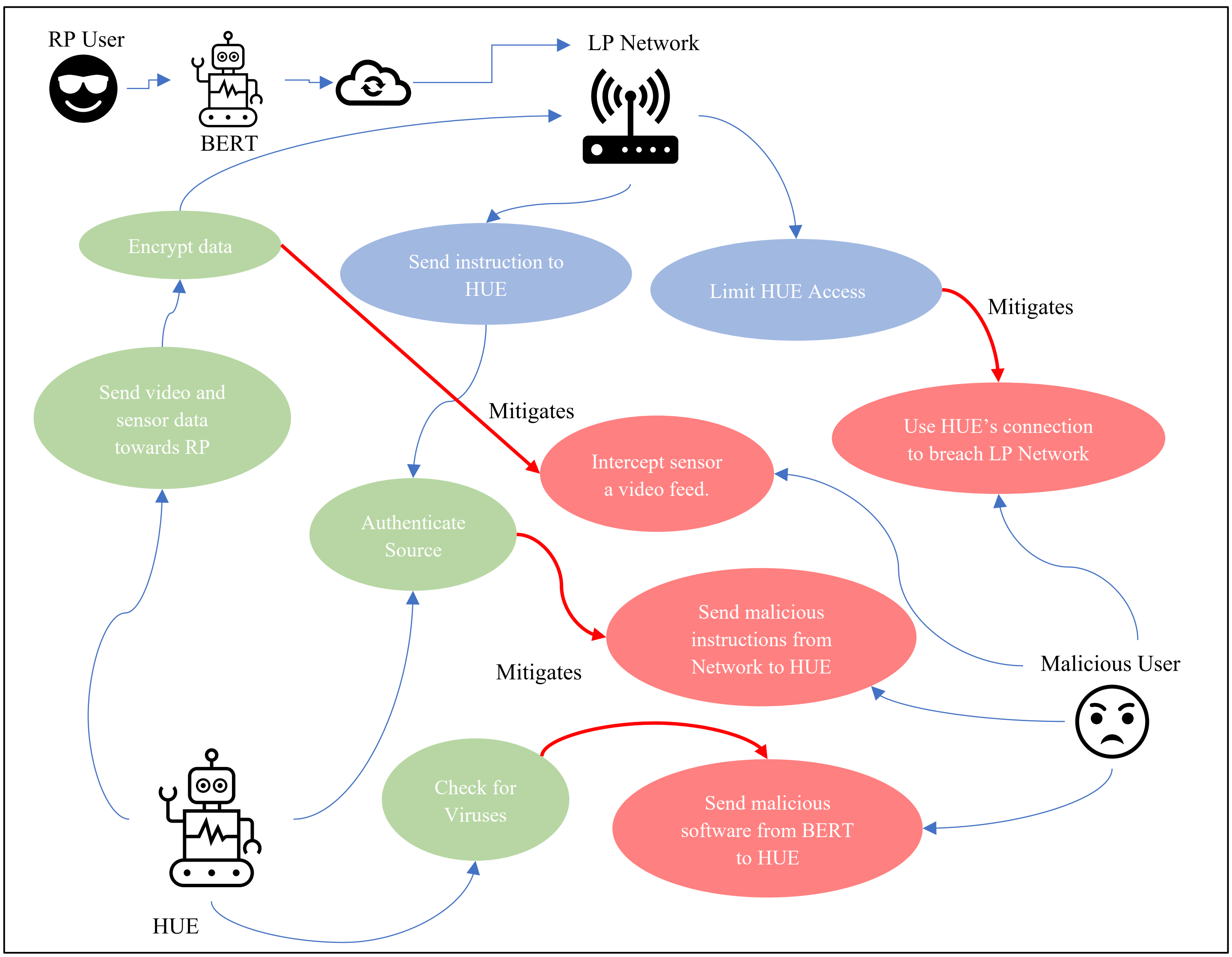

Figure 1: Hue threat analyses and workflow

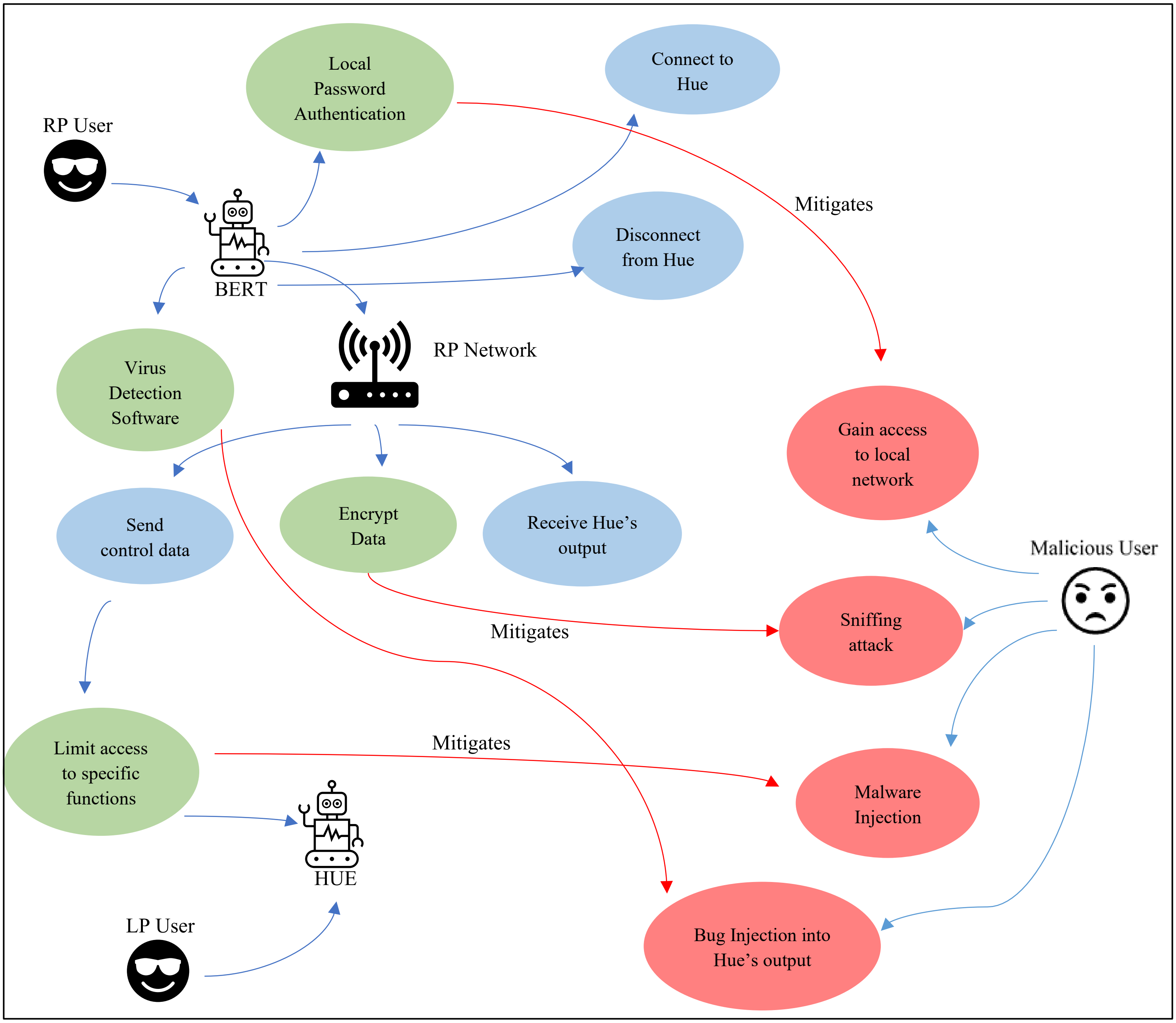

Figure 2: Bert threat analyses and workflows

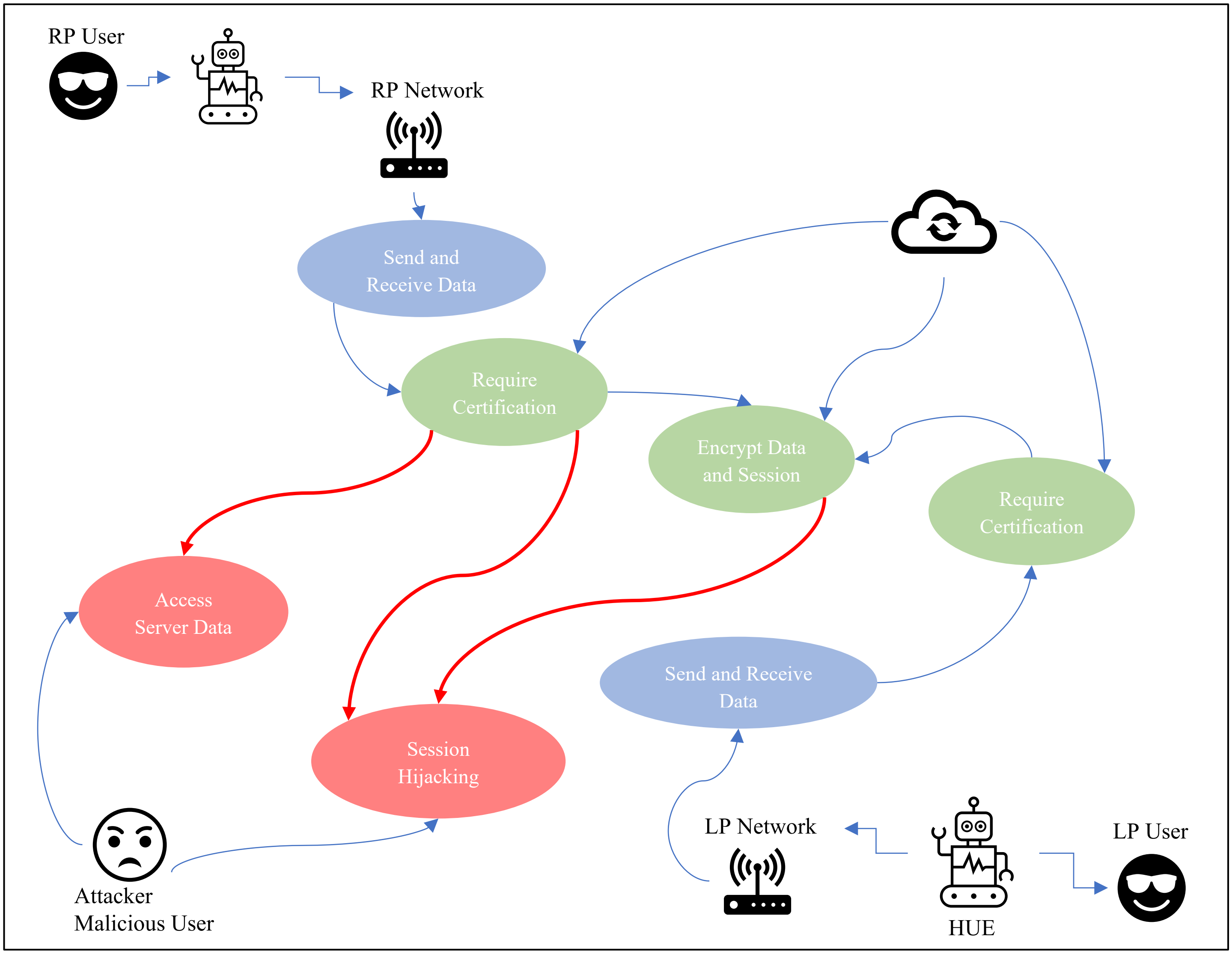

Figure 3: Network threat analyses and workflows

H. Prioritization

The user stories were ranked using a Fibonacci sequence scale from 1 to 8. Each stakeholder ranked the user story from their perspective. All ranks are summed to give the total rank. The total rank dictates the priority of each user story.

For each sprint, several stories are selected from highest to lowest priority. If a user story is dependent on a lower ranking user story, the lower ranked story will be selected preferentially. If any stories are not finished in a sprint, they will be re-added to the product backlog with their initial rank.

A minimum of 2 and a maximum of 7 user stories can be selected for a sprint. The length of a sprint will be between 1 and 4 weeks.

Product Backlog

HUEBERT

No.	User Stories	Client	Product Owner	Tester	Developer	Total
S14	As the RP or LP, I want to be safe when utilizing HUE, BERT or THORP and not be concerned about electrical hazards during gameplay.	8	8	8	3	27
S16	As an LP or RP, I want to feel and be safe around HUE, BERT, or THORP when they are in motion.	3	8	8	5	24
S7	As HUE, I want to be compatible with different THORP modules so that I may play a variety of games	8	5	5	5	23
S1	As a remote user, I want to control HUE's position such that I can pick up small game pieces and place them accurately within the game environment.	8	8	1	5	22
S18	As an LP, I want to keep my private devices and IoT systems safe from malicious RP users.	5	8	5	3	21
S4	As the RP, I want to see and understand the Game Environment	5	5	3	8	21
S2	As a remote user, I want to control Hue's motion such that I can dictate its path through space.	5	5	2	8	20
S20	As a player, I want to create and modify my HUEBERT system so that I may increase the variety of playable games and the performance.	2	2	8	8	20
S19	As THORP or BERT, I want to integrate with HUE seamlessly through industry standard processor or code.	3	5	3	8	19
S10	As HUE, I want to be able to move at an expected and enjoyable speed for a user.	5	5	2	5	17
S11	As the LP, I want HUE to be quiet so that I can converse and play with other LPs without major disruption.	5	5	2	5	17
S12	As the RP, I want HUE to react to my inputs in a timely manner such that I can reasonably play my game of choice.	5	5	2	5	17
S13	As the RP or LP, I want to be able to react to the feedback I receive from HUE in a timely manner without inhibiting delays.	5	5	2	5	17

S15	As the LP or RP, I want to safe when utilizing HUE, BERT or THORP and not be concerned about physical hazards during gameplay.	5	5	5	2	17
S17	As an RP, I want to choose what files or data is available to HUEBERT, THORP and LP's.	5	5	3	3	16
S5	As the RP, I want to send instruction to HUE.	5	5	2	3	15
S6	As the RP, I want to have controls that control HUE.	5	5	2	3	15
S3	As a remote user, I want HUE to be able to access all areas of a standard sized game board (400 mm arm range).	5	5	1	3	14
S8	As BERT, I want to be compatible with multiple RP control devices so that I may control different THORPS	5	3	3	3	14
S9	As the RP and LP, I want to equip different THORP and BERT add-ons to play a variety of games	5	3	3	3	14

Appendix A: Product Perspective and Systems

The following outlines the design decision and co-design of the HUEBERT system based on the collected requirement listed in this document.d

1. Product Perspective

HUEBERT is a new, self-contained product consisting of two systems: HUE and BERT. HUEBERT is designed to integrate and support a modular THORP project. A more detailed breakdown of each system can be found in Figure 5.

The remote system, named BERT, consists of the User I/O and interface. The local system, named HUE, includes a networked server, control and computation device, and an interface for a robotic *Game Interface*, named THORP.

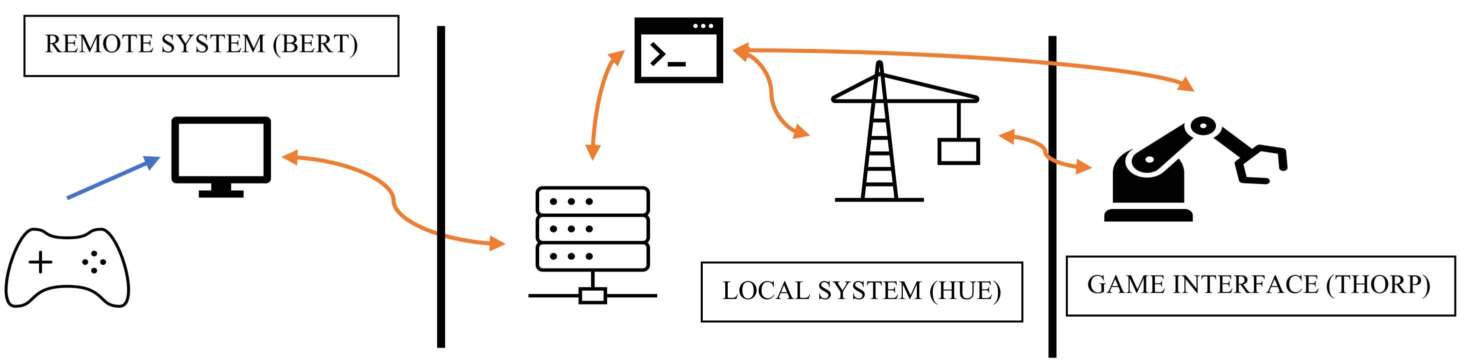

Figure 4: The major components of the product HUEBERT

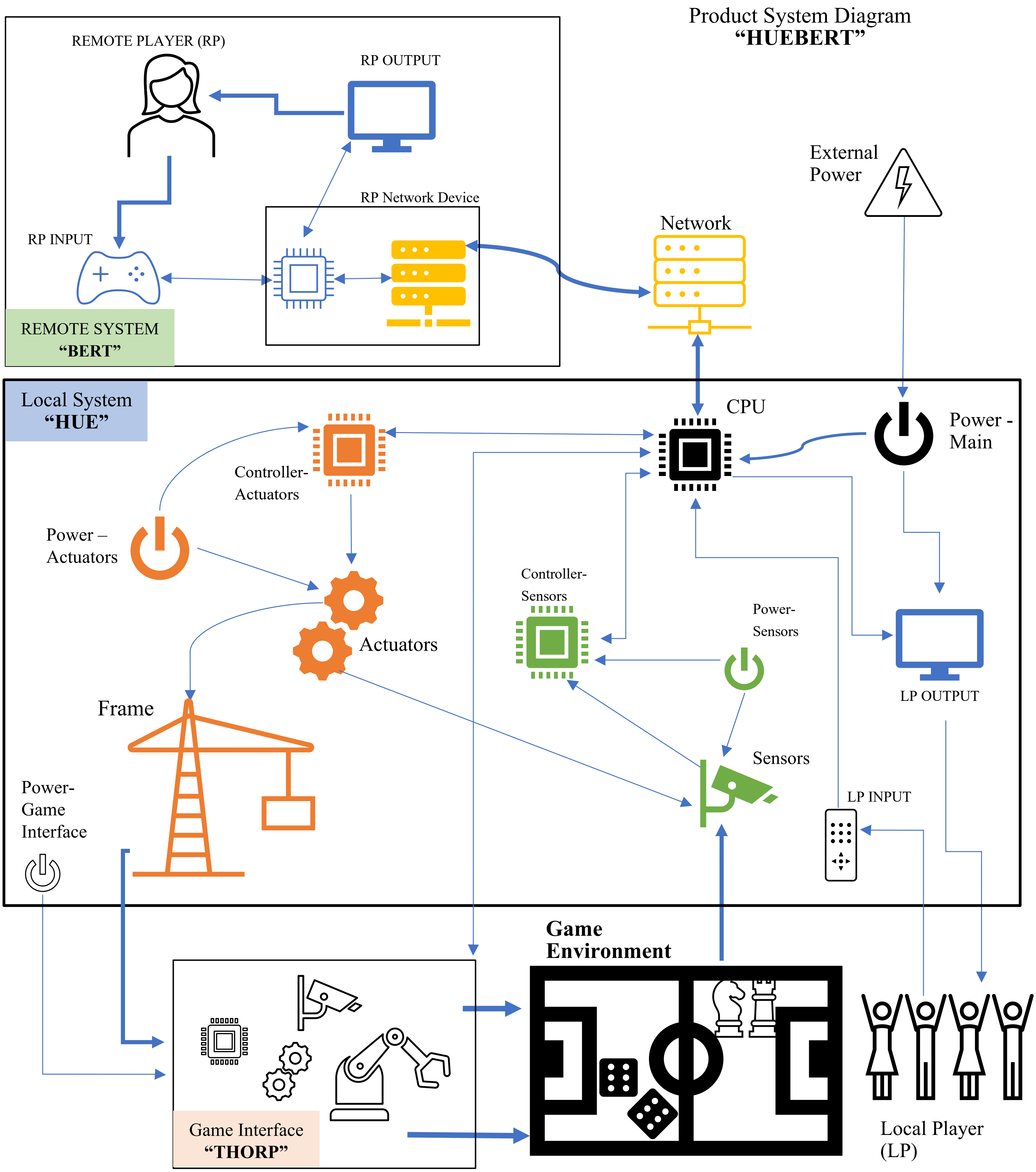

Figure 5: Detailed HUEBERT system diagram

2. User Classes, Functions, and Systems

The following sections summarize the discrete sections and classes of the product HUEBERT. A diagram relating all classes, functions and systems can be found in Figure 5.

H.2.1 HUE

The local system of HUEBERT is named HUE. HUE encompasses the controllers, CPU, sensors, power, and I/O in the local system. HUE also includes the THORP interface.

H.2.2 BERT

The remote system of HUEBERT is named BERT. BERT encompasses the RP I/O and the software running on the RP Network Device.

H.2.3 THORP

THORP is the name of the *Game Interface*. THORP is a modular, self-contained product that integrates with HUE. THORP physically interacts with the game environment. While closely related, THORP is not part of the HUEBERT product.

H.2.4 Game Environment

The *Game Environment* is any table-top game and environment. Some examples are Monopoly, Risk, DND, and Yahtzee.

H.2.5 Remote Player (RP)

The RP is a player of the game who is not physically able to access the game environment. The RP uses the HUEBERT system to play the game.

H.2.6 Remote Player Input

The *Remote Player Input* is part of BERT. This device can be changed and modified and contains a variety of controls which allow the RP to control HUE and THORP. Some possible examples of this device could be an Xbox controller, a smart phone, and a 3D printed custom controller.

H.2.7 Remote Player Output

The *Remote Player Output* is displayed on the RP Network device. It consists of a number of windows/monitors which displays sensor data and video feed.

H.2.8 Remote Player Network Device

The *Remote Player Network Device* is a separate system such as a laptop, tablet, or PC which can run the BERT software and interface with the *Remote Player I/O.*

H.2.9 Network

The *Network* system connects HUE and BERT. This system might be a server, P2P connections, or Bluetooth signal which send data and instruction from BERT to HUE and THORP.

H.2.10 CPU

The *CPU* is the main computation component of HUE. The *CPU* handles communication through the *Network*, *LP I/O,* mathematic computation, instruction for the *Controller-Actuator/Sensors,* and the communication with THORP.

H.2.11 Actuators

The *Actuators* will consist of OTS stepper, servo, and DC motors as well as electromagnets. These components will wield the Frame in response to input from the *Controller-Actuators*. These components must meet the speed, precision, noise, and torque requirements.

H.2.12 Sensors

The *Sensors* will give basic information about the game environment and will be mounted/integrated with the *Frame.* They may include, cameras, IR sensors, Ultrasonic sensors, Hall Sensors, and Potentiometers.

H.2.13 Frame

The *Frame* will allow for the *Actuators* to create desired motions and positioning of HUE. The *Frame* will be modular and allow for additions. This system may include 3D printed structural components, bearings, axles, springs, couplers, and fasteners

H.2.14 Controller – Actuators/Sensors

The *Controllers* will give input or manage output from the *Sensors* and *Actuators.* They may be separate devices and/or part of the CPU device.

H.2.15 External Power

The *External Power* is a 120V standard wall socket.

H.2.16 Power – Main/Actuators/Sensors/Game Interface

The *Power* modules provide electrical voltage and current to the *Actuators, Sensors*, *CPU, Controllers,* and THORP.

H.2.17 Local Player I/O

The *Local Player I/O* is a set of buttons and input as well as one or more displays with allow the LP's to configure and view the status of HUEBERT. It may also include a communications display to the RP.

3. Operating Environment

HUEBERT will operate in two physical environments: The *Game Environment* and the RP's environment. Both environments must be sheltered from elements such as rain, wind, and excessive dirt. The HUEBERT system will use Python, C++, and JavaScript. HUE main program will run on a single board computer (raspberry pi). HUE may interface with Arduino micro controller for motor and sensor input. The *Network* system may implement JavaScript for a client/server set up. BERTs main program will run on the *RP Network Device*. The *RP Input* may utilize Arduino's to handle controller input.