# 2 State of the Art

In this chapter a survey of pervasive healthcare applications is presented. In order to gain a greater understanding of which are the building blocks of an Internet of Things (IoT) system, a reference model is also presented.

## 2.1 Internet of Things

### 2.1.1 What is IoT?

Internet of Things (or IoT) is an emerging communication paradigm, often hailed as the driver of the Fourth Industrial Revolution [2].

The definition of this concept has evolved over time with the development of other technologies such as data analytics, embedded systems, etc. Nowadays it describes a strategy supported on the development of networks of smart devices that exchange and process information through Machine-to-Machine (M2M) communications, usually based on the Internet Protocol (IP). This technology enables ubiquitous systems to gather remarkable amounts of information regarding the surrounding environment, which can later be turned into insight through the usage of data analytics tools, like Machine Learning algorithms.

More specifically in the health care domain, this technology can provide many benefits as it enables remote and continuous health monitoring. It allows non-critical patients to be monitored from the comfort of their own houses rather than in hospitals or clinics, reducing the strain on scarce hospital resources such as doctors or beds. This is particularly beneficial to those who live in rural areas, with limited access to health care services. This allows elderly people and those with chronic diseases to have greater control over their own health, allowing them to life more independently. Moreover, with the automatization of medical procedures, these systems can make health care infrastructures more efficient and thus lower the costs of health care. Particularly, in the realm of clinical research, by analyzing the data collected by these ubiquitous systems, it may be possible to find new relationships between certain pathologies and different physiological signals, such as variations in body temperature or heart rate [3]. These correlations, commonly referred to as biomarkers, can then be used to guide treatment decisions, enabling novel predictive, prognostic, and diagnostic processes in health care.

## 2.2 A Reference Model for Pervasive Healthcare Applications

In order to develop an IoT system, it is crucial to design it based on a reference model. A reference model provides a general structure (or a "template") for designing systems, thus enabling the comprehension of these complex systems by breaking them down into simple and distinct functional layers, while also defining some common terminology used in its domain.

In 2014 the IoT World Forum (IoTWF) architectural committee published an IoT architectural reference model, composed by seven layers as shown in figure 2.1. This model provides a simple and clean functional view into the different components of an IoT system without restricting the scope or locality of its components. However, from a hardware perspective, in this work we will restrict our focus to the most common approach taken by investigators, using 3 different components:

- **Endpoint** or **edge** nodes (corresponding to Layer 1), which interact with the physical world, capturing data.
- **Gateway** devices (Layers 2-3), which connect to multiple **edge** nodes, filtering and aggregating the data generated by these, while relaying it to a central server;
- **Central** server (Layers 4-6), which is responsible for collecting, storing and analyzing the captured data in order to provide users with valuable insight;

While this model can be used to develop IoT systems for any industry (from agriculture to smart cities), in the context of the dissertation we will focus on pervasive healthcare applications and its enabling technologies.

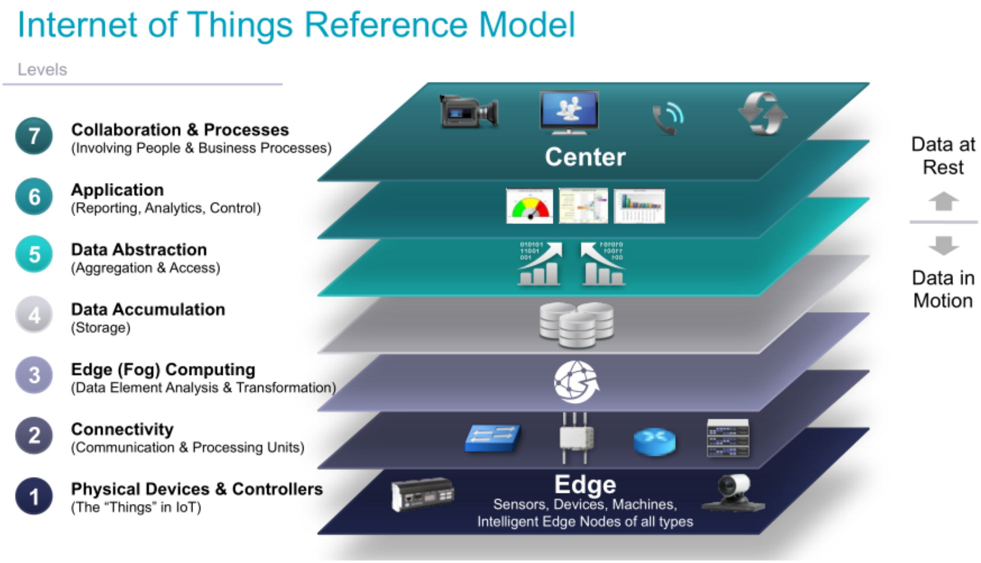

Internet of Things Reference Model
Levels
7 Collaboration & Processes
(Involving People & Business Processes)
6 Application
(Reporting, Analytics, Control)
5 Data Abstraction
(Aggregation & Access)
4 Data Accumulation
(Storage)
3 Edge (Fog) Computing
(Data Element Analysis & Transformation)
2 Connectivity
(Communication & Processing Units)
1 Physical Devices & Controllers
(The "Things" in IoT)
Center
Edge
Sensors, Devices, Machines,
Intelligent Edge Nodes of all types
Data at
Rest
Data in
Motion

Figure 2.1: IoT reference model published by IoTWF. Source: [4].

### 2.2.1 Layer 1: Physical Devices and Controllers

To-do: Review this section later.

The first layer of the model is the physical devices and controller layer. This layer houses the "things" in the Internet of Things: the endpoint devices composed of sensors and actuators that perceive and interact with the physical world. Through those interactions, the devices generate data, which is then sent across the network for analysis and storage.

When designing an IoT network, the first step should be to analyze the mobility and data requirements of the system [1]. **Mobility** describes the devices' ability to move and, if it is able to, how frequently it does so. **Data** requirements describe how much data is generated and transmitted by each device per unit of time, and how critical is it to the operation. Simpler health monitoring applications can include a single temperature or heart rate sensor while more complex applications can include pulse oximetry, electrocardiogram (ECG), respiration rate sensors, etc.

With these key requirements established, we can now discuss some other characteristics of the smart devices, like:

- **Power source**: This classification describes if the device has an internal energy supply powering the device or if it has continuous power delivery from an external source. If the device must be mobile, it will require a portable power source, a battery. Battery-powered devices are not bound to a single location, but the finite energy source constrains the device's energy consumption and lifetime, leading to limited memory, computation and connectivity capabilities. Shorter lifetimes also mean higher maintenance costs, as these devices will need to be replaced or upkeeped more often.

- **Transmission range**: This classification describes how far away the devices can communicate. In healthcare, these usually have short transmission ranges. For example, a fitness band that communicates with a smartphone will be at most located a few meters from it.

In order to properly design these devices, it is necessary to understand what problems currently reside within clinical environments. Today, hospitalized patients need to be wired to various measurement instruments when continuous biomonitoring is required. This confines the patients to their beds, restricting their mobility, and may also cause skin irritations and infections, aggravating their discomfort and deterioration of their health condition [5]. Moreover, the detachment of electrodes from the patient's body, provoked by patient's movements, is one of the main sources of false alarms. These require immediate attention from the hospital staff, contributing to their exhaustion and may ultimately result in the desensitization to the alarms, reducing their response time to real emergencies [6]. Wearable, wireless, and non-intrusive devices can to minimize these issues to a large extent, and are seen as one of the key components of IoT-based healthcare systems [7].

In recent years there has been remarkable progress on the development of wearable devices, driven by recent technological breakthroughs in the miniaturization of sensors and microfabrication processes [8]. From the literature, we can classify the sensors used in these devices in 3 distinct categories based on information that can be extracted from them, as shown in table 2.1:

- Monitoring the patient's biosignals, used for evaluating the patient's health condition.

- Monitoring the patient's activity or motion, used for detecting fall events, determining the patient's location and travelled distance, estimating the patient's body posture, etc.

- Monitoring the patient's environment, mainly used for assessing environmental hazards, *e.g.* gas leaks in a patient's home or an industrial workplace.

| **Type of monitoring** | **Type of sensors used** |
|---|---|
| Vital Signs Monitoring | Body Temperature, Heart Rate, Heart Rate Variability, Respiratory Rate, Galvanic Skin Response, Blood Pressure, Pulse Oximetry, ECG, Glucose Level Sensors |
| Activity Monitoring | Accelerometer, Gyroscope, Magnetometer |
| Environmental Monitoring | Air Temperature, Barometer, Humidity, Gas Sensors |

Table 2.1: List of sensors commonly used in pervasive healthcare applications, adapted from [9].

Note: Is it technically correct to designate these as "sensors" or should they be designated physiological / environmental signals instead?

### 2.2.2 Layer 2: Connectivity

The second layer of the model focuses on connectivity, on linking the different components of our system, ensuring reliable and timely data transmissions. This includes all communications within the system, which can be split into two categories: communications within the local network (*e.g.* between edge nodes and the gateway devices), and communications between the edge of the local network (*e.g.* gateway devices) and the central server.

#### Communication Protocols

Technologies are designed with certain use cases in mind. They drive their development and thus it is natural that each one has their own advantages and disadvantages, depending on their use. For instance, short range wireless protocols are limited by the transmission range, but long range protocols usually have a higher energy consumption, which may be unviable for networks with very constrained devices. Each protocol defines their own frame formats and communicates within certain frequency bands, some of which may require licenses. Using licensed frequency bands, provides better performance and ensures greater reliability since the network operator grants you exclusivity of frequency spectrum within a certain area.

From the literature, we can identify a set of key requirements for choosing the communication protocols:

- **Energy consumption**: Since the networks are often composed of energy constrained devices, the communication protocol should be lightweight and energy efficient in order to maximize the devices' lifetime.

- **Latency**: Certain applications deal with time critical events, for example the detection of health emergencies. In these cases, any delays in the communications can cause great detriment to the patient's well-being, making it crucial to minimize them.

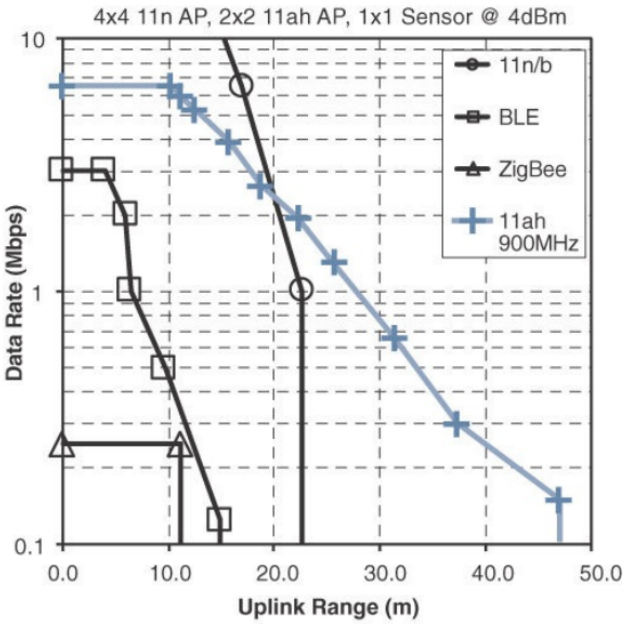
- **Reliability**: Depending on the critical nature of the data that is being communicated, the network stack may need to implement processes such as error-detection, retransmission or handshakes in the communications to ensure more robust transmissions (*e.g.* as implemented in TCP/IP based protocols). Generally, these features come at the cost of greater latency. Therefore, when choosing the communication protocol a balance must be found between reliability and latency.

- **Security**: Security is one of the most important requirements of any system, but this is especially true for healthcare applications. Due to the sensitive nature of the information, it is crucial to secure it from malicious actors. Communication protocols must implement security mechanisms, such as encryption or data integrity verifications, that ensure the transmissions are not compromised in transit, thus denying third parties the ability to snoop or tamper the transmissions. This issue is studied in depth in [10].

- **Throughput**: The communication protocol should ensure there is enough bandwidth to handle all communications within the designated transmission range. Even within similar technologies, this can vary wildly with the range as seen in figure 2.2.

- **Interoperability**: To ensure the interoperability of the system it is imperative to choose protocols that are widely accepted and supported in the application domain. This also contributes to the longevity of the system, as these will most likely remain supported for longer time periods.

- **Scalability**: These systems contain an enormous amount of devices, and each one must be uniquely identified. The communication protocol must ensure that it can address every device in the network.



Simulation Assumptions: 1% PER, 4dB NF,
32 Bytes, D-NLOS Fading, Indoor-to-Outdoor
PL Model. 900MHz has12dB propagation gain.

Sensor Antenna Gain: 11ah (-6.5dB)
and 11n (-4dB). AP antenna gain = 2dB.
* BT Long Range Adds 125 kbps and 500 kbps Modes

Figure 2.2: Throughput versus Transmission range for four WHAN to WLAN communications protocols. Source: [1]

Regarding the communications within the local network, these generally have short transmission ranges. The edge devices are usually energy-constrained devices, so it is crucial to choose an energy-efficient communication protocol. Within these, the most widely adopted protocol is Bluetooth Low Energy (BLE), a low-energy version of the classic Bluetooth protocol. ZigBee and Radio-frequency Identification (RFID) are also used in many systems, but are bested by BLE. Although there isn't a universal specification for RFID, the most widely used standard is the EPCglobal Gen2 RFID [11], which will be the standard discussed in this work. For the sake of simplicity, we will address it only as EPC/RFID.

Out of these three technologies, BLE offers greater throughput, better security, and nearly the lowest energy consumption [12]. Table 2.2 shows a comparison between the different protocols.

| | **Bluetooth Low Energy** | **EPC/RFID** | **ZigBee** |
|---|---|---|---|
| **Band of operation** | 2.4 GHz | LF, HF, UHF, EHF | 2.4 GHz |
| **Communication** | Bidirectional | Unidirectional (Bidirectional for Active tags) | Bidirectional |
| **Topology** | Point-to-Point, Piconet, Broadcast, Mesh | Point-to-Point | Mesh |
| **Range** | 100m | <10m, (100m for Active tags) | 20m |
| **Data rate** | 1Mb/s (up to 2Mb/s) | 40kb/s | 250kb/s |
| **Security Features** | AES-128, Secure pairing prior to key exchange | ✗ | AES-128 (Optional), Network key shared across network, Optional link key to secure application layer communications |

Table 2.2: Comparison between the more common communication protocols used within short range. Adapted from [7].

Regarding the communications between the gateway devices and the central server, most researchers try to make use of existing infrastructure in order to facilitate the deployment of new systems. This means, that the communications between the gateway devices and the central server are often done through generic IP-based protocols such as Wi-Fi and Ethernet.

To-do: Should a short comparison between long range protocols be discussed (e.g. NB-IoT)?

### Application Protocols

To-do: Review this section later.

So far we've discussed the underlying networking technologies that link the devices in our system. But according to the OSI Model, we can note that many of these technologies don't define the application layer: how the devices communicate with each other, how the data is formatted, if there's hierarchy within the network, etc. When considering networks composed of constrained devices, generic web-based protocols such as HTTP may be too heavy for IoT applications, which prompts the development of novel lightweight messaging protocols suited for these systems. The most popular application layer protocols in IoT systems are Message Queuing Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP). Table 2.3 shows a short comparison between these two.

| | **MQTT** | **CoAP** |
|---|---|---|
| **Transport protocol** | TCP/IP | UDP/IP |
| **Messaging pattern** | Publish-Subscribe | Request-Response |
| **Communication model** | Many-to-many | One-to-one |
| **Security** | SSL/TLS (Optional) | DTLS |
| **Strengths** | TCP and Quality of Service (QoS) options provide robust communications, easier to implement | Better for lossy networks, lower latency |
| **Weaknesses** | Higher overhead and energy consumption than CoAP | Not as reliable and is less supported than MQTT |

Table 2.3: Comparison between CoAP and MQTT protocols. Adapted from [1].

In [13] the authors compare these two protocols in greater length, along with the more commonly used web-based protocol HTTP. They analyze the latency in the communications (from the edge devices to remote servers) and the RAM usage in the devices for each protocol and for different data sources (respiration rate, oxygen saturation and heart rate signals) and found that CoAP presented the best overall performance. Nonetheless, all protocols had very low latencies (less than 1.5s) and low memory usage. The authors indicate that MQTT might be more suitable for a certain messaging pattern, designated "Publish/Subscribe" model.

In this model, the devices that want to send messages, also called "publishers", broadcast them within specific message topics. The devices that wish to receive these messages, also called "subscribers", can subscribe to topics to be notified when new messages are sent. CoAP uses a different messaging pattern, called "Request-Response". In this paradigm, a device sends a request to receive certain data and the second responds to this request. We can observe that former model is event-driven while the latter is query-based.

### 2.2.3 Layer 3: Edge (Fog) Computing

IoT systems can often have hundreds or even thousands of sensors generating data multiple times per second, 24 hours per day, which can demand an unsustainable amount of network and computing resources. Moreover, certain applications may be time critical, where delays in communication can be very detrimental. To minimize these effects, it is crucial to initiate data processing as close to the edge of the network as possible. This paradigm is usually referred to as "edge computing" (when the data processing occurs at the endpoint devices) or "fog computing" (when it happens at the edge of local network, *e.g* in "gateway" devices). The third layer of the model defines how the system prepares the data for storage and higher level processing for the next layers. However, the devices often have limited computing capabilities, so the data processing at this stage is generally very limited, mostly focused on "preprocessing" the data in real-time and handling more time critical events. More demanding and thorough analysis is left to the central server. The different processes applied at this stage usually are:

- **Filtering**: Assessing if the data should be processed at a higher level.
- **Formatting**: Reformatting data to ensure consistent formats for higher-level processing.
- **Cleaning**: Reducing data to minimize the impact of data on the network and higher level processing systems.
- **Evaluation**: Determining whether data represents a threshold or alert. This is especially relevant for applications that deal with time critical events as seen in the previous section.

### 2.2.4 Layer 4: Data Accumulation

To-do: Review this section later.

The data that is generated by the edge devices is propagated through the system, moving through each layer with each sensor reading. Up to this point, the model is event driven. However, most applications cannot make use of the data at the rate it is generated. In this layer, Data Accumulation, we define how the system captures the data and stores it, so it becomes usable for applications when needed, providing a transition from event to query-based processing. This includes any issues related with data storage: how the data should be stored (*e.g.* using a relational database, non-relational database, distributed file systems, data compression, etc.), what data should be stored and which should be kept for short-term use, etc.

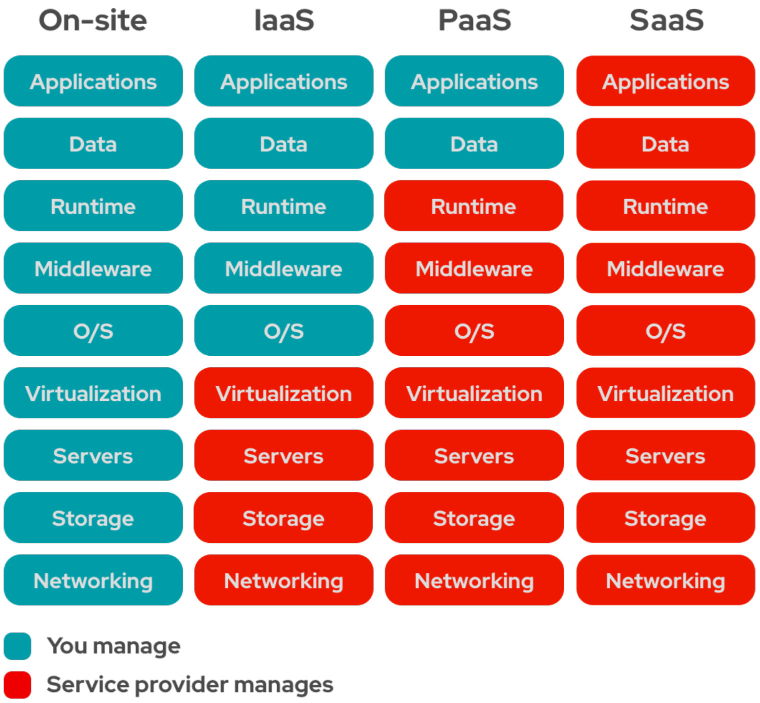
To-do: Should homomorphic encryption methods be discussed? Discuss what is Hadoop? What are the 5 V's of Big Data? Compare different database technologies? NoSQL technologies are better suited for unstructured data. Also, maybe mention that in the context of the project, we'll be using a local server?

In healthcare, big data analytics is seen as one of the most promising features of IoT applications. However, the massive data collection that is associated with ubiquitous systems causes many issues concerning the processing and storage of data. Cloud platforms are often seen as a solution to these problems [7]. This is made possible due to the elasticity in allocating, swiftly and inexpensively, computing and storage resources on-demand, adjusting itself to the needs of each application. We can find 3 distinct types of cloud services:

- **Infrastructure as Service (IaaS)**: Provides control over the remote machine (composed of virtual or dedicated hardware), operative system and middleware. This approach gives system designers the highest level of flexibility over the infrastructure, but requires more maintenance.

- **Platform as a Service (PaaS)**: Provides a simple framework for developing applications, where the service provider manages the underlying infrastructure issues such as software updates and hardware maintenance.

- **Software as a Service (SaaS)**: Provides the finished applications to be used by the end users, in this case health workers, that enable them to work. A simple example is a web-based email service, such as Gmail or Microsoft Outlook.



| On-site | IaaS | PaaS | SaaS |
|---|---|---|---|
| 🟦 Applications | 🟦 Applications | 🟦 Applications | 🟥 Applications |
| 🟦 Data | 🟦 Data | 🟦 Data | 🟥 Data |
| 🟦 Runtime | 🟦 Runtime | 🟥 Runtime | 🟥 Runtime |
| 🟦 Middleware | 🟦 Middleware | 🟥 Middleware | 🟥 Middleware |
| 🟦 O/S | 🟦 O/S | 🟥 O/S | 🟥 O/S |
| 🟦 Virtualization | 🟥 Virtualization | 🟥 Virtualization | 🟥 Virtualization |
| 🟦 Servers | 🟥 Servers | 🟥 Servers | 🟥 Servers |
| 🟦 Storage | 🟥 Storage | 🟥 Storage | 🟥 Storage |
| 🟦 Networking | 🟥 Networking | 🟥 Networking | 🟥 Networking |

🟦 You manage

🟥 Service provider manages

Figure 2.3: Differences between the cloud offerings and on-premise solutions. Source: [14]

Nonetheless, security and privacy remain as key concerns for the implementation of cloud-based solutions. The information must remain accessible to authorized parties such as health-care providers, but the patient's health data has to kept private. To solve this, there are two commonly adopted features in the literature: access control policies and data encryption. Access control policies define who can access the data, by authenticating them (validating the identity of the user) and by authorizing them (ensuring that the user has permissions to perform a given operation). Data encryption ensures that, even if the data is leaked, it is still unreadable to third parties, and therefore the sensitive information remains secure and private.

### 2.2.5 Layer 5: Data Abstraction

To-do: What other details should be added in this section? FHIR functionality will be developed in a later chapter

In the previous layer, Data Accumulation, we've defined how the system captures the information. In some cases, the collection of data may require the development of multiple concurrent storage solutions, each using different technologies, resulting in a very complex environment. The purpose of this layer is simplify how the applications access the data, to reconcile the different data stores and ensure the information is complete and consistent. This is generally accomplished with the development of Application Programming Interfaces (API).

An API is a computing interface that defines a set of rules that "explain how computers and applications communicate with one another", acting as an intermediary between different systems [15]. It defines what operations can be performed, how to request them, which are the accepted data types, etc. In this case, it decouples applications from the storage solutions, by encapsulating their functionality behind the interface. This ensures the modularity of the system as the applications become independent of whichever technologies are used in the data stores.

To-do: Discuss what is a RESTful API? Or leave it for the FHIR chapter? Also, review later this bridge between APIs and EHRs

Now, we need to understand what and how information is shared within the healthcare domain. As patients continuously move around the healthcare ecosystem, their health information must be available, discoverable and understandable to different entities (hospitals, laboratories, pharmacies, etc.). This prompts the digitization of medical files and the development of standards for exchanging these records instantly and securely to authorized parties [16], which are called Eletronic Health Records (EHRs). EHRs are the digital equivalent of a patient's paper-chart, they contain the patients' full medical history: previous diagnoses, treatment plans, test results, known allergies, among other details.

One of the most prominent standards for exchanging EHRs is Fast Healthcare Interoperability Resources (FHIR). FHIR is a standard developed by Health Level Seven International (HL7), which is a non-profit organization involved in the development of international healthcare informatics for over 20 years. FHIR builds upon previous data format standards like HL7 v2 and HL7 v3, and is becoming more and more widely adopted within the healthcare industry [17].

### 2.2.6 Layer 6: Application

To-do: Complete section.

The sixth layer is the application layer, where the system ingests the captured data and proceeds to analyze it. Users can then interact with the system through Graphical User Interfaces (UI), which provide different functionalities depending on the application. Some