

Design and Implementation of A Cloud Based Intelligent Surveillance System

LEI ZHOU

A thesis submitted to Auckland University of Technology in
partial fulfillment of the requirement for the degree of
Master of Computer and Information Science (MCIS)

School of Engineering, Computer and Mathematical Sciences

Abstract

Visual surveillance is widely used in different fields such as security, transportation, medicine, and logistics. However, these videos and images require appropriate storage and computing platforms. Cloud based surveillance systems are suitable for big data processing and storage.

This thesis will discuss how architecture and function modules of a cloud based system can be created, a cloud-based intelligent surveillance system (CISS) can be published on a private cloud using a Virtual Machine (VM). Our users are able to connect a camera at a desired location to CISS so that the system can capture videos and images, and then push notifications to a web page where the users can watch the videos via the cloud.

This thesis presents an in-depth demonstration of how the portable and efficient cloud based surveillance system (CISS) works. It makes full use of the cloud computing technology, and applies computer vision technology to visual surveillance. The system is able to quickly and accurately detect events from streaming videos, and apply the feature of push notifications to achieve event based alarming in real time. The new technology makes CISS more advanced than the conventional visual surveillance system.

The overall objective of this thesis is to implement four key functionalities, (a) streaming video input, (b) intelligent visual surveillance (IVS), (c) video transcoding and storage in real time, (d) message push and media streaming output.

Keywords: intelligent surveillance, computer vision, cloud storage, push notification.

Table of Contents

Abstract	I
List of Figures	IV
List of Tables.....	VI
Attestation of authorship	VII
Acknowledgements	VIII
Chapter 1 Introduction	1
1.1 Background and motivation	2
1.2 The research questions	5
1.3 Contributions.....	6
1.4 Objectives.....	6
1.5 Structure of this thesis.....	7
Chapter 2 Literature Review	9
2.1 Cloud computing.....	10
2.1.1 Cloud computing architecture.....	10
2.1.2 Service models	10
2.1.3 Deployment models	12
2.1.4 Characteristics of cloud computing.....	13
2.1.5 Commercial offerings of cloud computing	14
2.1.6 Advantages of cloud computing.....	15
2.1.7 Opportunities and hindrance for cloud computing.....	16
2.2 Video Surveillance as a Service	19
2.2.1 Definition of VSaaS.....	19
2.2.2 Function of VSaaS	19
2.2.3 VSaaS in security	20
2.2.4 Key benefits of VSaaS	22
2.3 Related work in surveillance	22
Chapter 3 Methodology.....	26
3.1 CISS requirements	27
3.2 Methodology for CISS	28
3.2.1 Research designing	28
3.2.2 Designing of system architecture	29
3.2.3 Determining of system operation process	30
3.2.4 Designing of subsystem	33
3.2.5 Designing of system modules	36
3.2.6 Setup experimental environment.....	42

3.3 Limitations of the research	43
Chapter 4 System Implementation	45
4.1 System architecture	46
4.2 Subsystem implementation	48
4.2.1 Surveillance system.....	48
4.2.2 Storage system	50
4.2.3 Web server.....	53
4.2.4 System database	54
4.3 System modules	56
4.3.1 IP camera based video streaming.....	56
4.3.2 Web based real-time video streaming	57
4.3.3 Image capture.....	59
4.3.4 Video recorder.....	61
4.3.5 Motion detection module	63
4.3.6 Face detection module	65
4.3.7 Camera sensor module	68
4.3.8 Event, alarm and push notification.....	69
Chapter 5 System Demonstration and Results	70
5.1 System testing plan	71
5.2 Experimental environment.....	71
5.3 CISS demonstration	74
5.4 Experimental results.....	78
5.5 Discussions and analysis	80
Chapter 6 Conclusion.....	82
6.1 Conclusion	83
6.2 Future work	84
References	85

List of Figures

Figure 1.1 System topology diagram	7
Figure 2.1 Cloud computing service models.....	11
Figure 2.2 SaaS provider to user.....	11
Figure 2.3 Hybrid cloud.....	12
Figure 2.4 Architecture of cloud based surveillance system.....	25
Figure 3.1 Structure of research design.....	28
Figure 3.2 B/S model of CISS.....	30
Figure 3.3 User authentication transmission.....	30
Figure 3.4 User operation flow chart.....	32
Figure 3.5 Setting of application pool.....	36
Figure 3.6 CISS user login process.....	37
Figure 3.7 Finite state machine for module: video recorder.....	39
Figure 3.8 Video player workflow.....	40
Figure 3.9 Event based alarm making module flow chart.....	42
Figure 3.10 System running on private cloud.....	43
Figure 3.11 Car plate recognition module.....	44
Figure 4.1 CISS topology diagram.....	46
Figure 4.2 Surveillance system flow diagram.....	48
Figure 4.3 The connection bridge: media connector.....	49
Figure 4.4 CISS iSCSI LUN.....	50
Figure 4.5 iSCSI target CHAP authentication.....	51
Figure 4.6 FTP VM network topology.....	51
Figure 4.7 Storage management system.....	52
Figure 4.8 CISS user interface software deploys on server-side.....	53
Figure 4.9 CISS database topology diagram.....	54
Figure 4.10 Web-based real-time streaming video.....	58
Figure 4.11 Image capture.....	59
Figure 4.12 Image capture process.....	60
Figure 4.13 Video recorder process.....	62
Figure 4.14 Motion detection module demo.....	63
Figure 4.15 Motion detection module flow diagram.....	64
Figure 4.16 Face detection module demo.....	65
Figure 4.17 Camera sensor module processes flow chart.....	68

Figure 5.1 RTSP: IP Cameras.....	71
Figure 5.2 Network facilities.....	72
Figure 5.3 Private cloud devices.....	73
Figure 5.4 CISS user authentication.....	74
Figure 5.5 CISS camera overview.....	75
Figure 5.6 CISS full page camera overview.....	75
Figure 5.7 CISS camera monitor with motion detection module.....	76
Figure 5.8 CISS camera monitor with face detection module.....	76
Figure 5.9 CISS camera sensor module	77
Figure 5.10 CISS alarm module.....	77
Figure 5.11 CISS event module.....	78
Figure 5.12 CISS event module comments part.....	78

List of Tables

Table 2.1 Advantages of public cloud vs. data center.....	13
Table 2.2 Characteristics of cloud computing.....	13
Table 2.3 Opportunities and hindrance for cloud computing.....	16
Table 3.1 Comparison of mainstream web application platform.....	29
Table 3.2 C# camera SDK supported codecs.....	34
Table 3.3 Comparison of web server.....	35
Table 4.1 user acc table in the database.....	54
Table 4.2 sensor table in the database.....	54
Table 4.3 events table in the database.....	55
Table 4.4 alarms table in the database.....	55
Table 5.1 QNAP TVS 682 specifications.....	73
Table 5.2 QNAP TS 253 PRO specifications.....	73
Table 5.3 CISS benchmark in local PC.....	79
Table 5.4 CISS benchmark in private cloud.....	80

Attestation of authorship

I hereby declare that this submission is my work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person except where explicitly defined in the acknowledgements, nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature  _____ Date 5 December 2016 _____

Acknowledgements

I would like to express my deepest gratitude to my thesis supervisor Dr Wei Qi Yan. This work will not be accomplished without his instructions and encouragement. Dr Yan always guides me to pursue the knowledge of real practice. I also hope to thank my secondary supervisor Dr Jian Yu for his invaluable advice. The two supervisors both help me a lot in the journey of this master's study. Finally, I would like to thank my peer friends: Mr Jia Lu, Mr Jun Shen, Mr Jia Wang and Mr Yun Su. Thanks for their support in this year of study indeed.

Chapter 1

Introduction

The objectives of the first chapter are to introduce the background and motivation of this thesis. In-depth analysis of the issues is included by the rationale underlying the research. A cloud-based visual surveillance system is proposed to be designed and implemented in this thesis. After referral the background, we clearly present and elaborate the research questions and contributions. Finally, the structure of this thesis will be presented.

1.1 Background and motivation

Security is important for every organization (Frank, 2011). The traditional security system is to build a “high wall” for secure protection. If there is an illegal invasion that cannot be timely detected, then it is unable to provide evidence and clues for tracking. One of the important components in traditional security system is Closed-Circuit Television (CCTV) (Norris, Moran, & Armstrong, 1998). CCTV as a part of the security system is used to protect our community.

Visual surveillance is an application of computer engineering, the functions of visual surveillance include camera control, video display and recording, data transmission. Traditional video surveillance refers to analog monitoring and digital monitoring, and its infrastructure includes a front-side camera, transmission cable, and video monitoring platform (Hossain, et al., 2012). Because of traditional video surveillance is lack of intelligent management, therefore it is referred to as “passive monitoring”, its primary functions are grouped in two categories, the first is real-time monitoring, the other is to replay surveillance video (Shibao, 2009).

The main function of real-time monitoring depends on security staff physically to watch the screen when unusual event happened, the staff can make decision to achieve surveillance alarming. As time goes, there is a big issue emerging, at very short notice amongst a large amount of surveillance videos it is extremely tough to find the evidence needed. Traditional video surveillance systems also are with other drawbacks, such as transmission and storage for a mass of surveillance videos which are extremely difficult and may waste huge social resources.

Nowadays, with a fast development of computer network, digital image processing, and data transmission technology, the conventional surveillance system (CCTV) cannot “swim” with the tide (BaoHong & Yan, 2015). However, lots of companies, warehouses, schools, hospitals (Soulsby, 2012), supermarkets, etc. (Li, 2011) still are using analog security system, such as CCTV and DVRs (Digital video records). With the exponential growth of massive surveillance videos, complex network environment and various brands of video-recording device cannot be compatible with each other, conventional surveillance systems are difficult to meet the needs of big data analysis, storage and

sharing. Therefore, a cloud based surveillance system (CISS) is required more than ever before.

Cloud computing has a positive impact on businesses. Many companies adopt it due to its profitability and scalability. In addition, its privacy and improved security are a key factor to business owners and investors (Yong, et al., 2012). Cloud helps SMEs to dispose of their costs in a significant way. Sharing and collaboration offered by cloud computing have made it easier and cheaper to work, especially in cross-industry. The most beneficial economic goal to users is the “pay-as-you-go” policy. The opportunities offered by cloud computing make trade decision easier, the technology also comes with more elasticity in employing and reserving resources.

The flexibility cuts on waste and reduces higher costs for running a server on “pay-as-you-go” as compared to buy (Kambatala, et al., 2014). As much hardware costs continue being declined, there are variable rates e.g. the costs of cloud computing and cloud storage are rapidly falling at a higher rate as compared to that of WAN. A cloud system can track the variations and relay the information to customers. Hence, it matches expenditure and resource utilization.

Cloud computing has reached another level whereby every business has its services (Armbrust, et al., 2009). Cloud computing as a service is expected to spend 32.8% of the whole worlds’ IT expenditure in the year of 2016. There are several integrated cloud systems in business (Devasena, 2014):

Cloud computing is convenient and easy to be used since one can work from anywhere for online transactions. Accountants use cloud computing to produce available monthly fee for their clients.

Cloud computing is a storage phenomenon that is growing popularity in surveillance. Video storage shows its challenges with the substantial infrastructure required because visual surveillance needs facilities of significant data storage that may be costly to any organizations. Moreover, any organizations have to release data captured over a period of time or replace the storage disks after numerous years. Also, once the storage disks are damaged or destroyed, the footages will be unavailable, and thus backup is absolutely

needed. With cloud computing, organizations can easily access their footages in anytime without worrying about managing the storage facilities (Song & Tian, 2015).

A cloud based visual surveillance system namely Video Surveillance as a Service (VSaaS) (Karimaa, 2011) is a cloud computing paradigm and has outperformed than other services of traditional video surveillance (Jiang, Sekar & Zhang, 2012). The VSaaS function is primarily driven by numerous key factors such as dynamic technology, cyber security, and remote access.

Visual surveillance offers increasing security for students, staff, and school property in an education enterprise. Educational institutions have diverse needs of integrated computing technology ranging from monitoring resources to manage the students and employees. Numerous scattered buildings are located within an institution and need surveillance with 24 hours a day, 7 days a week and 365 days a year. An institution may have multiple campuses located at a diversity of geographical locations and all need control from a center (Prati, et al., 1957).

Property companies are in direct need of high quality and real-time security systems. The environment where people interact raises a wild range of security concerns, while the property security staff has to utilize up-to-date surveillance systems so as to meet the security concerns (Behl & Behl, 2012). The VSaaS is the best service to be employed in such a surrounding where a system delivers a more secure scenario for the tenants despite the optimization and improvement of the property management are more effective (Lamy-Bergot, et al., 2009).

Cloud systems have data centers that make the system redundant and reliable. The traditional web systems do not have such centers, and an organization would require additional hardware that is to protect any failures to the maximize uptime (Yadav & Singh, 2016).

A shared infrastructure allows an organization to save energy costs since there is better hardware utilization. When an organization runs its data center, it will never utilize all the servers in the center that may waste energy.

Cloud based surveillance allows any organizations to benefit from upfront capital costs that offer them with an opportunity to invest their capital in other businesses since the system has saved benefits (Wo, et al., 2011).

Cloud based surveillance systems have shared resources like IT personnel and staff support that presents a lower cost than an organization having the whole team on board. Organizations can be in a position to reduce their IT expenses through adapting to a VSaaS.

Cloud computing systems allow infrastructures to be shared as servers with all hardware being fully harnessed. The platform also allows reduction of supporting workloads (Neal & Rahman, 2012).

1.2 The research questions

With emergence and gradual maturity of cloud computing, it provides a new way of methods for the construction of intelligent surveillance. Data analysis, search and resource sharing and other issues can be resolved through the cloud in sky. But at present, the research is not in deep, the actual applications are not common. To truly combine the idea and method of cloud computing together and apply them to the development of intelligent surveillance, it still has a long way to go.

The purposes of this thesis are to design and implementation a cloud based intelligent surveillance system. We take previous related work into account, it would seem that existing cloud based visual surveillance system still has many problems left for being resolved. Thus, this thesis focuses on creating a new cloud based intelligent surveillance system that has better function, performance and user experience. Hence, the research questions of this thesis are:

Question 1. *How to find sufficient cloud storage for a visual surveillance system?*

As visual surveillance requires enough storage space for videos, the surveillance system is running on cloud in sky, so we think the first question is how to find enough space to store surveillance visual data such as videos and images.

Question 2. *How to implement the push notification in a cloud based visual surveillance system?*

After examining research problems of cloud based systems, we think push notification is a pivotal feature of cloud based system. Hence, our surveillance system could push any events latest happened to the user's terminal in real time, in order to achieve alarm making simutanously.

Question 3. *How to integrate the necessary computer vision functions from Network Video Analytics (NVA) into this cloud based visual surveillance system?*

In order to make visual surveillance system more intelligent, we need update our web based system and integrate all components into the cloud based system. We set up the system with the functionalities like face recognition, motion detection, license plate recognition, etc.

1.3 Contributions of this thesis

The focus of this section is to discuss the contributions of this research project. In this thesis, a cloud based visual surveillance system is proposed and well developed. In this project, there are two contributions.

The first is a Cloud Based Intellgient Surveillance System (CISS) which is able to integrate the Computer Vision (CV) module. Users can specify Computer Vision module for every camera, such as motion detection, face detection and so on. When an event is detected, the system will push the corresponding messages to the users.

The second is that our users are able to view and manage CISS via cloud services, such as Google Chrome on smartphones or PCs. The users also are able to enter comments on an event for other users.

1.4 Objectives

Firstly, this thesis introduces cloud based visual surveillance system related to VSaaS. Cloud computing is an enhanced technology to run a business. Applications are runing on a multi-agent cloud system instead of on a LAN or PC. Cloud computing is set as

shared commodities that are scalable whose computing resources are scattered all over the world and available on a network when needed (Begum & Khan, 2011). VSaaS is a web hosted wireless security system that allows the users to remotely storage, manage, record, play and monitor surveillance videos, all these are possible on the cloud platform.

Secondly, this thesis presents a whole framework for cloud based visual surveillance system (web and database). The overall objective of this thesis is to develop a cloud based system which is to implement key four functions, (a) video streaming input, (b) intelligent visual surveillance (IVS), (c) video transcoding and storage in real time, (d) message push notification and streaming media output. Figure 1.1 shows the whole structure of cloud based intelligent surveillance system.

Finally, this thesis will demonstrate the whole system and present the experimental results of the comparison on the different environments; the experimental results will be discussed and analyzed at end of this thesis.

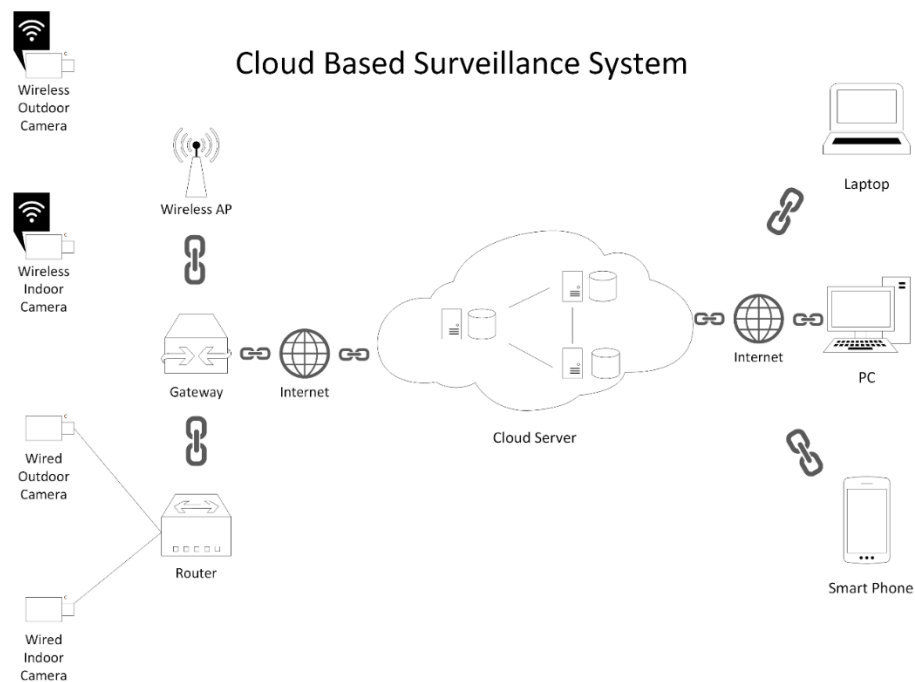


Figure 1.1 The topology diagram of the proposed system

1.5 Structure of this thesis

In Chapter 2, we will discuss the literatures related to two main categories in this thesis: cloud computing and VSaaS. Cloud computing models and survey of VSaaS are emphasized in this chapter.

In Chapter 3, we will explain the research requirements and methodology of this thesis. The potential solutions to the research question will be stated. Finally, the limitations of the research will be addressed.

In Chapter 4, we will present the system development and implementation within the scope of this thesis. The system architecture will be established, then we will implement the functionalities of each component in the CISS.

In Chapter 5, we will explain the experimental layout and setting, testing environment and implementation as described. The experimental results and the outcomes are compared with each running environment. The solutions to achieve research objectives will be tested, and the final experimental results will be demonstrated. At last, the conclusion of this research and future work will be presented in Chapter 6.

Chapter 2 Literature Review

The main objectives of this chapter are to explain and evaluate the Cloud Computing and Visual Surveillance as a Service. We present the evidence of an in-depth comprehension of the literature relevant. Firstly, we review the literatures on cloud computing that helps us decide research questions and the running environment of the CISS. At last, we determine the CISS architecture, functions and the research context with key issues raised through reviewing the literatures related to VSaaS.

2.1 Cloud computing

Cloud computing has emerged as a powerful and crucial force in system management, consumption, and information services. It excludes the provisioning mechanism at a level so that users can avail resources (Alamri, et al., 2015).

Cloud computing has become the major concept for high performance computer applications, and it saves the hassles of resource provision. Cloud computing started becoming popular in the year of 2007 and has expanded ever since going over the board ahead of grid computing regarding to the growth rate of web search interest. Because of its effective and powerful resource, it has drawn interests from researchers working with data and scientific computer applications (Agarwal & Prasad, 2012).

2.1.1 Cloud computing architecture

NIST (National Institute of Standards and Technology, US) defined the cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (Mell & Grance, 2011). Cloud computing refers to applications delivered as services over Internet and how the hardware and software in the data center are to provide these services (Luo, Jin, Song & Dong, 2011). In this case, the data equipment and software are the cloud, and the service is called Software as a Service (SaaS) (Qian, et al., 2010).

2.1.2 Service models

Cloud computing refers to the triple “as a service,” i.e. Application/Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). (Mell & Grance, 2011)

- Software as a Service (SaaS): Applications are availed through Internet for the consumption of the end users. The users cannot control the underlying cloud infrastructure, such as Operating System (OS), Hard Disk Drive (HDD), Network.
- Platform as a Service (PaaS): Platform, software kits, and tools are made available over Internet, such as programming language, services. The users also cannot

control the underlying cloud infrastructure, but they can figure out settings for the application-hosting environment and oversight the deployed application.

- Infrastructure as a Service (IaaS) / Hardware as a Service (HaaS) (Shawish & Salama, 2014): Tangible physical devices located at data centers can be accessed through the network via any terminals. Users are available to control OS, HDD, deployed applications, and limited control for networking, such as firewalls, access rules.

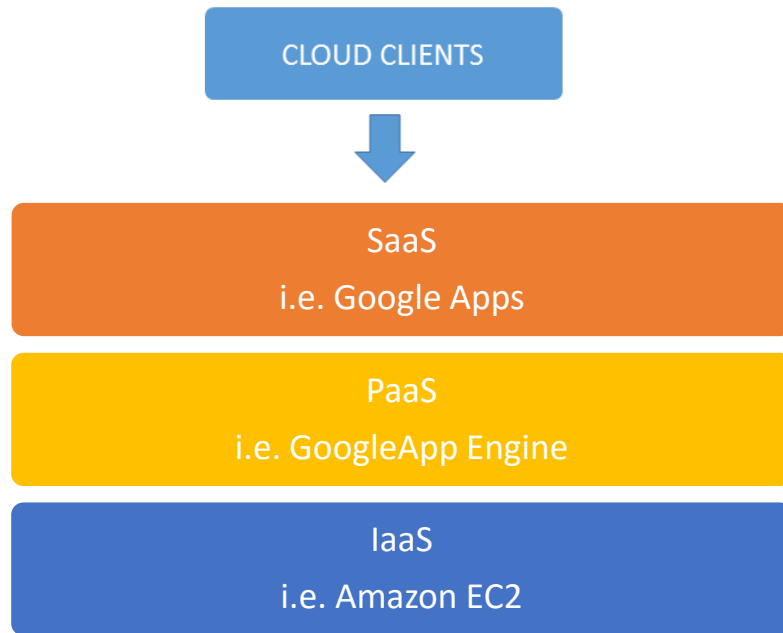


Figure 2.1 Cloud computing service models

Figure 2.1 points out the provider-user relationship. i.e. from the cloud provider through utility computing it goes to the SaaS provider, and from the SaaS provider through web applications it reaches to SaaS user in Figure 2.2.



Figure 2.2 SaaS provider to user

2.1.3 Deployment models

The cloud has four deployment models (Shawish & Salama, 2014):

- Public cloud is a cloud made accessible to public, and the service is related to utility computing, it is available from the third recipient service provider through the data highway which is affordable, such as Amazon, Microsoft Azure, Google Apps.
- Private cloud refers to internal data centers of a business or an organization that is not made available to the public cloud, e.g. Amazon Private Cloud.
- A community cloud is used and managed by a group of organizations with common interests. For example, group shared policy, security requirements, etc. The advantages of community cloud are that it is cheaper than that of private cloud though management of the cloud can be outsourced. The drawbacks of the community cloud are that its costs are higher than a public cloud, fixed bandwidth and data storage only shared amongst all community members.
- Hybrid cloud is a combination of the public and private clouds (Christian, et al., 2009). Figure 2.3 shows the interface between public cloud and private cloud.

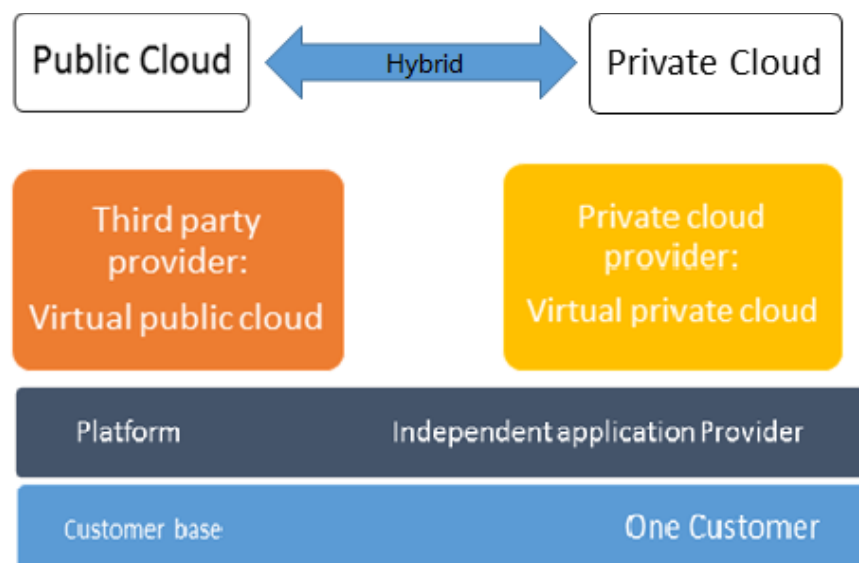


Figure 2.3 Hybrid cloud

The advantages of hybrid cloud are in reducing capital expense, easy to improve resource allocation, also supportive in cloud bursting, it offers controls for private cloud and rapid scalability using public cloud. But the disadvantages of hybrid cloud are security, privacy, and integrity (Goyal, 2014).

Public cloud is the best form of cloud due to its advantages shown in Table 2.1. We can make a private cloud to obtain multiple general clouds via hybrid cloud computing (Song, Tian & Zhou, 2014).

Table 2.1 Advantages of public cloud vs. data center

Advantage	Public Cloud	Conventional Data Center
Impression of unlimited computing resources on demand	Yes	No
Removal of an upfront devotion by cloud users	Yes	No
Economics of scale because of gigantic data centers	Yes	Usually not
Capability to compensate for the use of computing resources on a short-term basis	Yes	No
Simplifying undertaking and increase usage via resource virtualization	Yes	No

2.1.4 Characteristics of cloud computing

Due to exponential growth of cloud computing, the associated technologies utilize the availability of excess capacities, changes in management and storage catalyze the need for cloud computing. This is the reason why cloud computing has been emerged. There are five characteristics of cloud shown in Table 2.2.

Table 2.2 Characteristics of cloud computing

Characteristics	Descriptions
On-demand self-service	Computing resources are availed by a service provider to the customer without human interaction.
Broad network access	Cloud resources can be accessed from anywhere in anytime by any devices via Internet in this world. (Devasena, 2014)
Resource pooling	Cloud resources of a provider are assembled to provide

	services. Several customers share these pooled resources on demand.
Rapid elasticity	Cloud resources are unlimited and availed elastically by the client. The customer only pays for the total resources used.
Measured service	Cloud computing is an adaptive system such as it can balance loads and optimize the use of resources automatically. The user can monitor and control resource utilization and thus bill clarity (Armbrust, Fox & Griffith, 2010)

New aspects of cloud computing are,

- Elimination of cloud users due to the quick appearance of unlimited computing resources on request.
- Removal of up-front commitment by cloud consumers, therefore, allowing businesses starts small and expand hardware resources only when the need increases.
- Capability to compensate for the use of computing resources on a short-term basis as needed and release as required. The rewards protected by allowing machines and storage go where they are not needed.

Cloud based data centers are accessible (Vecchiola, Pandey and Buyya, 2010). An example of cloud is internal enterprise data center whose software is modified only with significant notice to administrators.

2.1.5 Commercial offerings of cloud computing

In cloud, there are three famous companies Amazon, Google, and Microsoft (Vecchiola, Chu, and Buyya 2009), they are renowned providers of commercial offerings of cloud. Amazon EC2 resembles the visible hardware and users can control nearly the entire software stack from the kernel upwards, it is hard for scalability and failover to occur since state management issues such as replication are application dependent.

Domain-specific platforms such as Google App Engine are targeted at regular web applications implementing an application structure of clear difference between computational tiers and storage tier or between the useful and useless tiers.

Microsoft Azure is between application frameworks (Microsoft Azure, 2016) i.e. App engine and virtual machines. MS Azure platform is a computing and service podium installed in Microsoft Data Center. Its programming paradigm undergoes two processes, web role, and worker role. A web part is Internet application that is reachable over HTTP and HTTPS breaking points, it is the forward end of any Microsoft Azure cloud situated application which is hosted in the ASP.NET and WCF advancements.

A worker role is a transforming entity representing the backend altering for Internet application. Azure service fabric is the structure of hooked physical computing nodes e.g. servers, high-speed network and switches. Computing and storage are also included in the Azure service fabric.

2.1.6 Advantages of cloud computing

Cloud computing has a positive impact on businesses (Devasena, 2014) as follows.

- A. It is flexible because cloud computing can meet business demands through the provision of different services.
- B. There are no upfront costs in infrastructure due to the “pay-as-you-go” policy. This allows the enterprise to reduce the cost of maintenance of infrastructure significantly, also decrease risks of infrastructure i.e. natural disasters, human-made disasters.
- C. Increased collaboration through staff is able to integrate and work on documents and common applications simultaneously. It also permits a follow up of other staffs, and records can be updated and received on time.
- D. Automatic software updates and server maintenance allow customers to use time and resources.
- E. Security is assured because the storage is developed based on cloud, therefore it’s ease in data access.

- F. Work can be accomplished from anywhere, workers receive work-life balance and better productivity. It is environmentally friendly because customers only use the required server space.

2.1.7 Opportunities and hindrance for cloud computing

The ten major opportunities for cloud computing are divided into three groups namely adoption, growth and policy. In deciding whether to employ cloud computing, one must evaluate the expected average and peak resource utilization, the potentials and delimitations of using physical equipment and the operating expenses that come with them in relation to the cloud computing environment are considered (Devasena, 2014). Table 2.3 presents the opportunities and hindrance relating to the adoption and growth of cloud computing.

Customers face a challenge in selecting their programs from one website to run in another due to a sever in the storage for cloud computing.

Another difficulty is in customer lock-in: it may be appealing to cloud providers, but the consumers are sensitive to increase its prices, authenticity problems and vendors going out of work.

Table 2.3 Opportunities and hindrance for cloud computing

Obstacles	Opportunities
Availability/business continuity setbacks	Employing multiple cloud-computing services
Data lock-in	Standardization of APIs, software compatibilities that enable hybrid cloud competition
Information audibility and confidentiality.	Using Firewall protection, data encryption, and VLANs
Problems of data transfer	Deploying Higher BW switches and FedExing
The unpredictability of the system performance	Using Gang Schedule VMs, improving VM support and employing Flash memory
Storage scalability	Developing scalable storage
Large distributed systems errors.	Using debugger, that uses distributive VMs
Quickly scalability	Using auto-scaling based on ML and snapshots in conservation
Sharing reputation fate	Using reputation-guards
Software Licensing limitations	Using open-source or pay-for-use licenses

The explanation is to systematize the API's such that SaaS developers could provide utility and data across several providers so that the collapse of organization won't affect all copies of customer data. However, several worries come up with this solution (Endicoot-Popvsky, 2014).

There are two possible bicker to solve these worries:

The Quality of Service (QoS) is equally important to price such as customers may not run after low-costs services.

Standardization enables a new usage model which enables hybrid cloud computing in which the general cloud is used to perform the added work that can't be carried out by the private cloud.

Cloud consumers experience security risks both from outside and inside of the cloud. The cloud users are responsible for the application while cloud providers are accountable for physical security and enforcing external firewall policies. Security between layers of the software stack is shared amongst the users and the operators.

Cloud computing offers external facing security in an easier and convenient manner, but the major challenge remains in interior-facing security. Therefore cloud providers must guard against theft or Denial of Service attacks by users (Davenport, et al., 2012).

Virtualization will be the primary security mechanism that protects against most attempts by attacking one another from the cloud infrastructure. But not all resources are virtualized, and not all virtualized environment is bug-free. Therefore a large Internet service needs to ensure that a single security hole does not compromise everything else. With protecting the cloud against the provider, the users will follow contracts rather than intelligent security engineering (Zhang & Chang, 2014).

Cloud users and providers will be required to consider the implications of placements and traffic at each level of the system if the need is to cut down costs. It can be overcome by shipping disks or whole computers, it will generate large delay intolerant point-point transfers (Franks, 2014)

It is possible for more than two Virtual Machines (VMs) to share the main memory assets of CPU in a cloud computing environments. However, the networks and disk I/O sharing continue to be a problem. This explains the variations in EC2's tier I/O performances. With improving the operating system and architecture to interrupt, virtualizing efficiently and I/O channels could overcome this challenges (Varghese, et al., 2014).

The features of cloud computing include infinite capacity on short-term demand usage, and no upfront cost involves the lack of clarity on the applications of persistent storage. The relational for this case is to come up with a storage system that offers more than just meeting the programmer expectations regarding durability but also includes the ability to manage data and excellent availability (Vecchiola, et al., 2010).

Cloud computing systems suffer from the problems of removing errors in the case of massive system distribution. The opportunity may overcome this on the reliance over Virtual Machines (VMs). The old version of SaaS was developed without the VMs, thereby capturing valuable information through their level of virtualization is possible (Gandomi & Haider, 2015).

The level of virtualization enables Google APP Engine to scale about load increase and decrease automatically, these scales apply to when the computer is idle. An auto-scaler that employs Machine Learning (ML) and snapshots conversion is applicable in this instance.

In cloud computing, a mistake made by one user affects the general of the reputation of all entities using the network; to solve this problem, reputation protection systems resembling to email services can be utilized (Hu, et al., 2014).

Since users buy software and pay for the additional maintenance costs, the current software licenses restrict the device that runs the application. These licensing models are not suitable for utility computing (Kaisler, et al., 2013).

2.2 Visual Surveillance as a Service

2.2.1 Definition of VSaaS

The term cloud based surveillance system is known as Video Surveillance as a Service (VSaaS) (Karimaa, 2011). Users are able to accessed from anywhere in any time. A user only needs an IP camera, Internet connection and VSaaS provider, then everything will be solved (Neal & Rahman, 2012).

The users access the videos distributed in the aspect that makes video tracking efficiently and the users do not require physical presence to the camera location. Again, on-site systems are necessary to process the footages.

On-site systems require substantial maintenance and complex infrastructure that always need managers to handle the technicalities (Hossain & Song, 2016). The VSaaS technology is very efficient in integrating the IP cameras at sites with the cloud hosting infrastructure (Tekeoglu & Tosun, 2015).

The real-time visual surveillance is streamed directly to a central management facility via the Internet as an aspect that enables the users' access in any time of their convenience (Gandomi & Haider, 2015). Important pieces of a video can be captured through the video streaming that is triggered by motion sensors.

VSaaS has five parts (On-demand self-service, Broad network access, Resource pooling, Rapid elasticity and measured service) that distinguish it from the traditional video surveillance system of DVR, NVR or video management software connected to the Internet and accessed remotely by users (Lamy-Bergot et al., 2009).

2.2.2 Function of VSaaS

In the cloud platform, a flexible stack of VM is provided, such as CPU cores, memory, storage space, network bandwidth. According to system usage requirements, users can select the proper configuration for VM, and it can avoid the waste of resources. VSaaS mainly supports following four functions:

The system allows access streaming media from the various cameras, such as IP camera, PTZ camera, fixed camera, USB camera, mobile camera, etc. (Hossain, et al., 2016), it also accepts multiple protocols (Kim, Nam, Kim & Cho, 2009) such as Real-time Transport Protocol (RTP), Real-Time Streaming Protocol (RTSP) (Chen, Shashidhar & Liu, 2012) and Real-time Transport Control Protocol (RTCP) (Liu, 2000).

The IVS system is from media streaming to conduct IVS using Computer Vision (CV) technology. The main function of IVS is automatically analyzing and extracting critical information from video resource. IVS technology uses significant cloud for image processing, filtering out extraneous information, providing vital information for staff (Shibao, 2009).

Every brand of camera has its own encoding format. According to user's requirement, the system can timely adjust the video resolution and bit-rate of network packet transmission, ensure that users receive a real-time video, at the same time the network bandwidth usage is reduced.

The system allows multiple clients (such as a mobile app, web page, etc.) to receive alarm messages in real time and streaming videos from anywhere in any time (Paul & Park, 2013).

2.2.3 VSaaS in security

The VSaaS gives users a better experience of control and security through the cloud (Wang, Liu & Fan, 2013). It offers simple, less maintenance and cheap method for system integrators to look for reliability and scalability in a security system (Zhang & Chang, 2014).

The differences between this modern technology and the traditional one in security are:

- Real-time video access from anywhere through any devices without network setup.
- Since VSaaS is hosted in the cloud providing better security access and storage capacity, unlike previous security systems that were hosted on-site (Limma & Tandayya, 2012).
- It offers great security, sharper image and found capabilities for big data analytics (Christian, et al., 2009).

- A cloud based system provides an on-demand deployment since one is required to plug in the on-premise appliance as well as auto-configuring the cameras. This is on the contrary to the traditional system that presents a long and complicated deployment process. A traditional system requires one install the OS software, router configuration, storage server setup, camera setup and supplication software (Rodriguez-Silva et al., 2012).
- The VSaaS allows an off-site support that is made possibly by on-site appliance while the traditional system requires a manually intensive process of support and maintenance with a massive on-site hardware and software in the company of configuration updates (Hassan et al., 2015).
- The VSaaS model helps its users acquire services with an extreme-low-upfront-capital expense. The system allows users to enjoy “pay as you use” monthly subscriptions based on the number of cameras an organization deployed as well as their retention period. On the other hand, the traditional system is associated with high-up-front expenses used for hardware purchasing and maintenance costs accompanied with variable support costs.
- VSaaS model offers a flexible combination integrating on-premise and cloud storage (Sharma, & Kumar, 2014). The users can have a live view of the cameras regardless of the video storage. The users can adjust resolution of the cameras without having to change the existing hardware as well. Numerous scales are associated with cloud based systems. However, the traditional systems have a rigid storage retention that limits the users with hardware capacity chosen during the time of purchase and system installation. Again, an additional hardware needs to be bought or replaced in the case the users need increase the retention period or the camera resolution (Rodriguez-Silva et al., 2012).
- Traditional systems don’t require bandwidth for on-site recording, but one would need it for remote view. On the other hand, bandwidth is required for remote viewing but it’s not necessary for on-site video storage buffering for VSaaS since the majority of the storage is streamed, a bandwidth is a requirement. Bandwidth allows Internet based streaming videos more attractive since it’s easily available and affordable (Neal & Rahman, 2012).
- Cloud based systems are free from cyber threats since they don’t have open ports or on-site firewalls like the traditional systems. The systems don’t require firewall installations like the traditional ones. On the other hand, Internet connected

traditional systems would require the use of remote accessibility that exposes them to possible cyber attacks due to the use of open ports and OS systems.

2.2.4 Key benefits of VSaaS

- Data is stored in cloud server which is allowed to be backup on a local hard drive.
- Users only require IP cameras for sending data through the cloud over Internet or Local Area Network (LAN).
- Without cloud storage, surveillance video footages cannot be destroyed.
- Users can receive emails in real time and updates from anywhere.
- Easy to expand on need and demand.
- Users are not required to purchase servers or store video onsite.
- A system is easy to be setup, used and maintained (Yan, 2016).
- It's an easy IP camera system with plug and play capacity.
- It offers more security and running time
- The system is 100% web based as well as applied Browser/Server (B/S) structure (Lage, Dolog & Leginus, 2014).

2.3 Related work in surveillance

There are at least hundred million cameras in the world that will generate real big data. A surveillance camera does not rest while working, truthfully recording everything (Nafisi & Azar, 2015). But most of the information is invalid, and effective information may only be distributed in a short period (Renkis, & Martin, 2013). Therefore, a large amount of video data stored to the database is not a minor pressure, and invalid data is a waste of resources (Lo, Wt, Chang, Ys, Sheu & Rk, 2014). This no doubt is a very big challenge to the traditional storage method (Beyer, J., Elhrouz, H., El & K., 2012,). The powerful storage capacity will reduce the cost of investment and management of data storage devices regarding to surveillance (Neal, D., Rahman & S. M., 2012).

Nowadays, surveillance systems are ubiquitously installed to store the huge amount of video footages. A surveillance system integrated Network Video Recorder (NVR) includes the engine of number plate recognition (Chen, Lin, Chieuh, Chang, & Tai, 2015). In the survey of this thesis, the use of secure digital (SD) card technology,

network attached storage (NAS), network video recorder (NVR), and cloud recording will be discussed (Engebretson & David, 2015).

Cloud computing is a networked technology that is entirely web based, which seems something quite different from surveillance of the primitive times (Stewart Garrett, 2012). However, as surveillance technology gradually becomes digitized and networked, visual surveillance market will continue to grow (Marko, 2012). In line with this trend, cloud computing and visual surveillance begin to be merged (Li, Zhang, T & Yu 2011). Cloud computing has advantages of resource optimization, reducing construction and maintenance costs, easy expansion and open sharing, etc. (Russell S, 2011).

Cloud computing allows operators to carry out a wide spectrum of applications more conveniently, but also enhance the efficiency (Dunkel, 2012). A computing resource minimization framework was proposed for cloud based surveillance video systems (Wu and Kao, 2014). Cloud computing brings new features to intelligent surveillance and becomes one driving force to promote the development of security technology.

Cloud surveillance is just at an early stage, and the future was immeasurable (Dunkel, 2011). Compressed Sensing (CS) and Dynamic Compressive Sensing (DCS) are used to prevent leaking any information, and the schemes effectively protect the security of the surveillance video across transmission processes (Qin, 2014).

The current surveillance equipment is mainly to be operated in front of the monitoring regions (Collins, et al., 2000), but the result is not effective and ideal. With the assistance of cloud computing technology, we can intelligently analyze in the back-end by launching the idle nodes in the network. Today, cloud computing technology can be very powerful in achieving intelligent visual analytics, comparing with simple analysis that has been unable to carry out effective monitoring of each detection point, which is playing an irreplaceable role in intelligence surveillance (Saeed, G. Juell-Skielse & E. Uppström, 2012).

Since intelligent analysis such as search and data mining of massive data is a series of complex calculations, and the resource requirements of various computing services are dynamic (Scull, 2012). It's a good way to use cloud computing to achieve the

convenient and on-demand access for sharing configurable computing resources (networks, servers, storage, applications, and services, etc.) (Rodriguez-Silva, D., A., Adkinson-Orellana & L., 2012).

In Figure 2.4, the architecture of visual surveillance system based on cloud is divided into four levels: acquisition layer, transport layer, support layer, and application layer (Wenzhe, Guoqing, Zhengjun & Xiaoxue, 2013). Cloud platform services are mainly located in the supporting layer and application layer of the surveillance system. The acquisition layer mainly completes the duty of acquisition and simple processing of visual information (S. Salleh, S. Teoh & C., 2012).

Transmission layer is mainly to complete the transmission, interaction, collection and other functions of visual information. The supporting layer mainly includes the basic resources, the fundamental software system, the infrastructural management, the distributed storage of video data, data mining and analytics (Chia-Feng & Shyan-Ming, 2012).

Basic resources include physical resources and virtual resources to provide video services in the way of cloud storage. The basic software system includes Web services, database services, application services, messaging services, database services and other content (Spillner, J, Muller, J, Schill & A, 2013). The basic management includes user management, resource management, task management, security management, deployment management, monitoring management, billing management and other functions (Xu, et al., 2016) (Xiong, et al., 2016).

The supporting layer is provided for a Platform as a Service (PaaS) and an Infrastructure as a Service (IaaS) (Yu-Sheng & Yue-Shan, 2012).

The application layer is to combine various related services based on a variety of different business users and constitute a complete application system that is able to meet the needs of users (Marier & Keven, 2012). In the cloud platform, the main service is the VSaaS which solves the problem of diverse terminals (Aleem, A., Sprott & C.R, 2013).

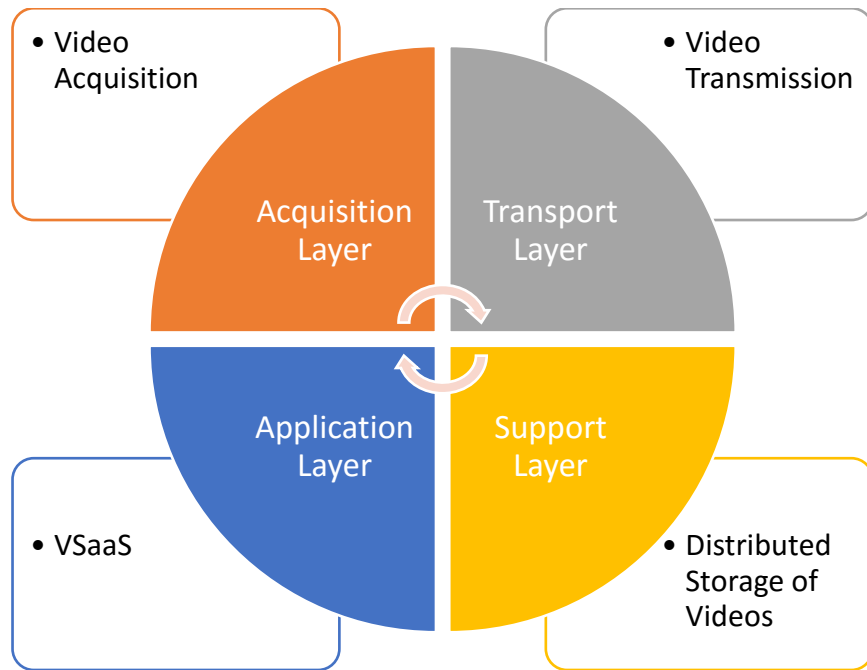


Figure 2.4 Architecture of cloud based surveillance system

Chapter 3

Methodology

In this chapter, we shall clearly present and elaborate the system requirements and the methodology for developing a cloud based intelligent surveillance system (CISS). Also, we introduce the details of designing the functionalities in each component of the system. With the confident and imaginative execution of research methods, the Boehm's Spiral software development model is applied to the CISS. Finally, we discuss the limitations of our research methodology in this thesis.

3.1 CISS requirements

The requirement of the CISS is necessary by using cloud computing for intelligent surveillance. Users can watch and get videos via cloud from anywhere in the world (Xiong, et al., 2014). By using CISS Network Video Analytics (NVA) module, when an event is detected, our users could get the push notifications that includes event entities in real time (Chen, et al., 2008). Users should be able to handle historical surveillance videos via their *ownCloud* APP.

The CISS requirements and detailed contents are defined as follows:

- User interfaces requirements
 - ✧ The web app should have user login page.
 - ✧ The web app should offer navigation bar to the user.
 - ✧ The web app should display video stream from camera in real time.
 - ✧ The web app should push event message in real time.
 - ✧ The web app should provide camera and sensor location information.
 - ✧ The web app should supply event log, and users can operate (view and delete) it in the system.
- Data requirements
 - ✧ Surveillance streaming should be in MJPEG format.
 - ✧ Surveillance event snapshot should be in JPEG format.
 - ✧ The snapshot should be stored in web APP snapshot folder.
 - ✧ Surveillance footage should be in MPEG-4 format.
 - ✧ Snapshot and footage URL should be stored in CISS database.
 - ✧ User authentication information should be stored in CISS user database.
- System capability requirements
 - ✧ The surveillance system should be able to transfer event data to the storage server via File Transfer Protocol (FTP).
 - ✧ The surveillance system should be able to display camera sensor location through Google Map API (Zhang, et al., 2010).
 - ✧ The storage system should be able to store surveillance footage and event data. The client can synchronize media data locally for offline view.
 - ✧ The surveillance camera should be able to stream videos via RTSP and transfer video data through FTP.
 - ✧ The system database should be able to store event log and data address.

3.2 Methodology for CISS

3.2.1 Research designing

To design and implement a cloud based intelligent surveillance system is the purpose of this thesis. As shown in Figure 3.1, there are six stages run through our research design.

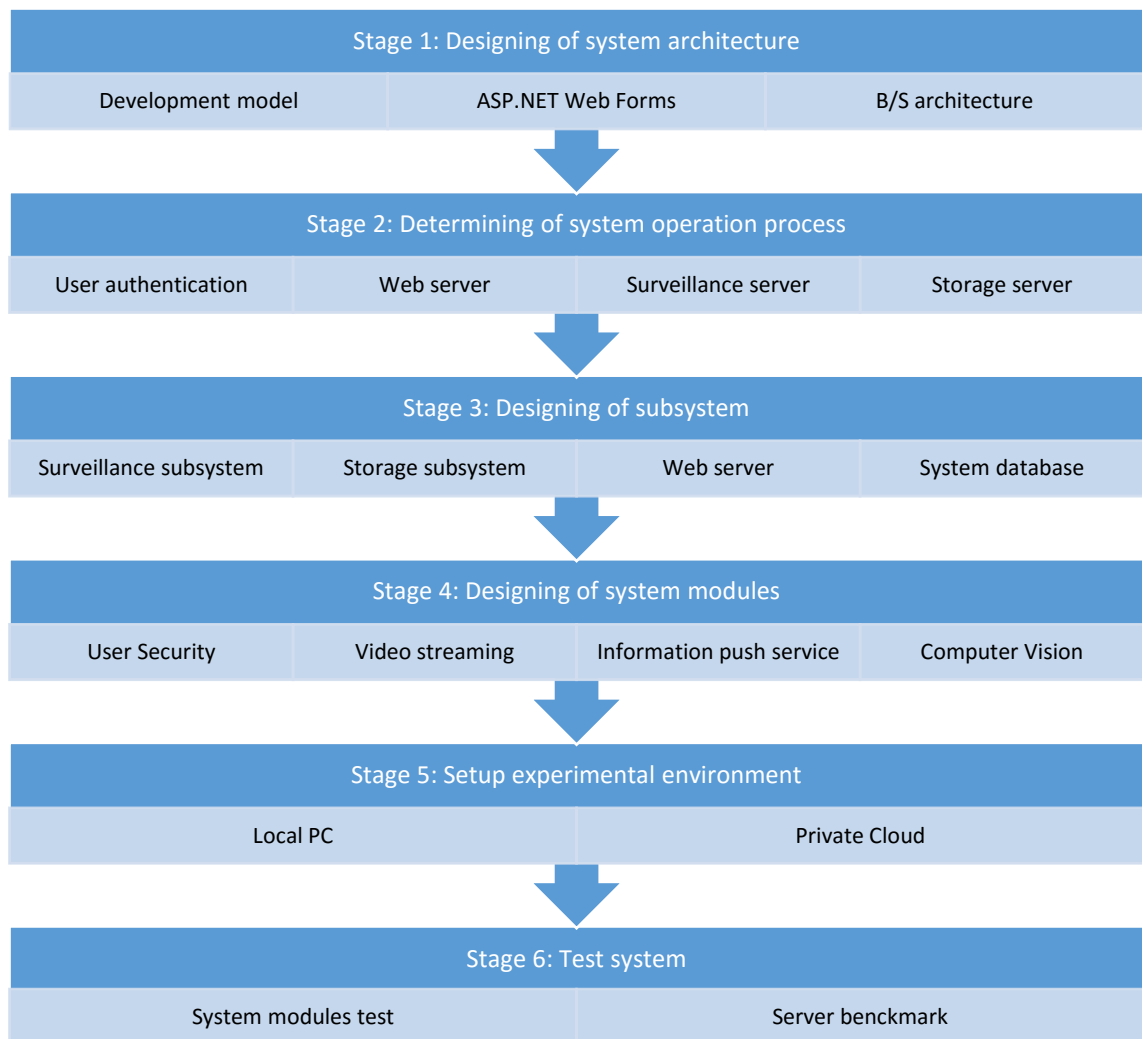


Figure 3.1 Structure of research design

The explanations of Figure 3.1 will cover through Section 3.2.1 to Section 3.3.6 and the experimental details will be explained in Chapter 5. We use Boehm's spiral methodology to implement the development process. This model is suitable for our project since it allows flexibility and does not require us to define the entire system from the beginning, because we do not have previous iterations to get a part of the development

process, this model relies on prototyping and client interactions. Our current system has been developed through cycles of this model (Boehm & Hansen, 2000).

3.2.2 Designing of system architecture

Our cloud based visual surveillance system consists of two parts. The client part uses a web browser, users should be able to control CISS via a web browser. In the server, Microsoft Hyper-V creates a Virtual Machine (VM). We deploy the runtime environment in Hyper-V based on Windows Server 2012 R2. Finally, we export the Windows server in a VM file to the private cloud.

Table 3.1 Comparison of mainstream web application platform

Performance compare	ASP.NET	LAMP	J2EE
Running speed	Fast	Very fast	Fast
Developing speed	Fast	Fast	Slow
Operating loss	High	Normal	Low
Maintenance	Easy	Easy	Hard
Running platform	Windows	Linux/UNIX/Windows	Most platform
Security	Very High	High	Very high
Construction cost	High	Very low	Very high

In this thesis, ASP.NET WebForm is applied to design the whole CISS. In Table 3.1, we conduct a comparison of three platforms. Before developing the surveillance system, we start considering the limitations and conditions. ASP.NET has the rapid-development features, and also it has excellent security. It is running on Windows platform, web application runtime environment based on .NET framework through Internet Information Service (IIS) provides the web service for users (Chen, Xu & Guo, 2013).

With the development of Internet and cloud technology, the traditional Client/Server (C/S) architecture is difficult to meet the feature of current Internet such as information sharing. B/S architecture known as Brower / Server structure is based on HTTP as the transmission protocol in use of maturity of WWW browser technology, users through the browser can use specific software to achieve the powerful features while reducing the pressure on the client.

The CISS system adopts the B/S architecture, a user accesses and controls the system through web browser in anytime from anywhere (Qi, & Yu, 2006).

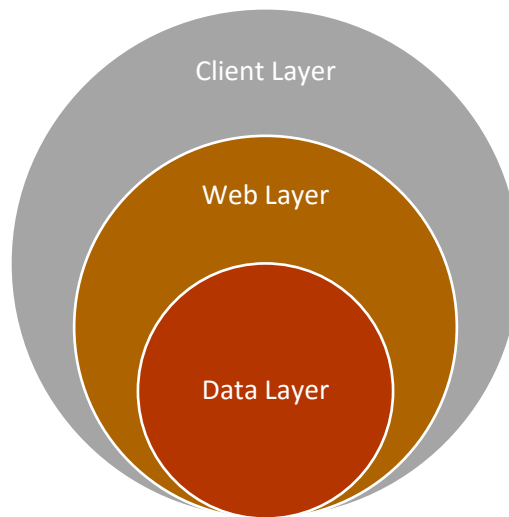


Figure 3.2 B/S model of CISS

Our system utilizes ASP.NET to deploy B/S model based on the three-tier architecture. In Figure 3.2, the model of CISS includes client layer, web layer and data layer (Yang, Deng & Wang, 2012). Client layer utilizes web browser to visit the website. Its responsibility is for human-computer interaction. Web layer is a system between client layer and data layer, across web server and application server to achieve the logical functionalities of the system. Data layer is the base tier of the system, it is used for data storage.

3.2.3 Determining of system operation process

ASP.NET supports three types of identity authentication (Bogard, Levendovszky & Charaf, 2006) Windows authentication, forms authentication, and passport authentication. Our system applies cookies as authentication mechanism which is based on forms authentication. A cookie is a small piece of text information, each time when a user visits the website as shown in Figure 3.3, cookies follow the user requests and web pages.



Figure 3.3 User authentication transmission

In login page, the users enter their username and password, and submit current page to the server, web applications read the certificate information (username and password) and compare it with a database record. When the users passed authentication system, ASP.NET will send a new cookie that contains efficient identity ticket. Function 3.1 shows the user authentication module, this cookie will include in any header of the request page.

Function 3.1 User authentication module
Operation: Verify user identity
Effect: Allow access or decline access
Utilize Microsoft Owin security module, set authentication mode as forms save in web.config file
<pre><authentication mode="Forms"> <forms name="AspxAuth" loginUrl="Login.aspx" /> </authentication></pre>

User operation processes

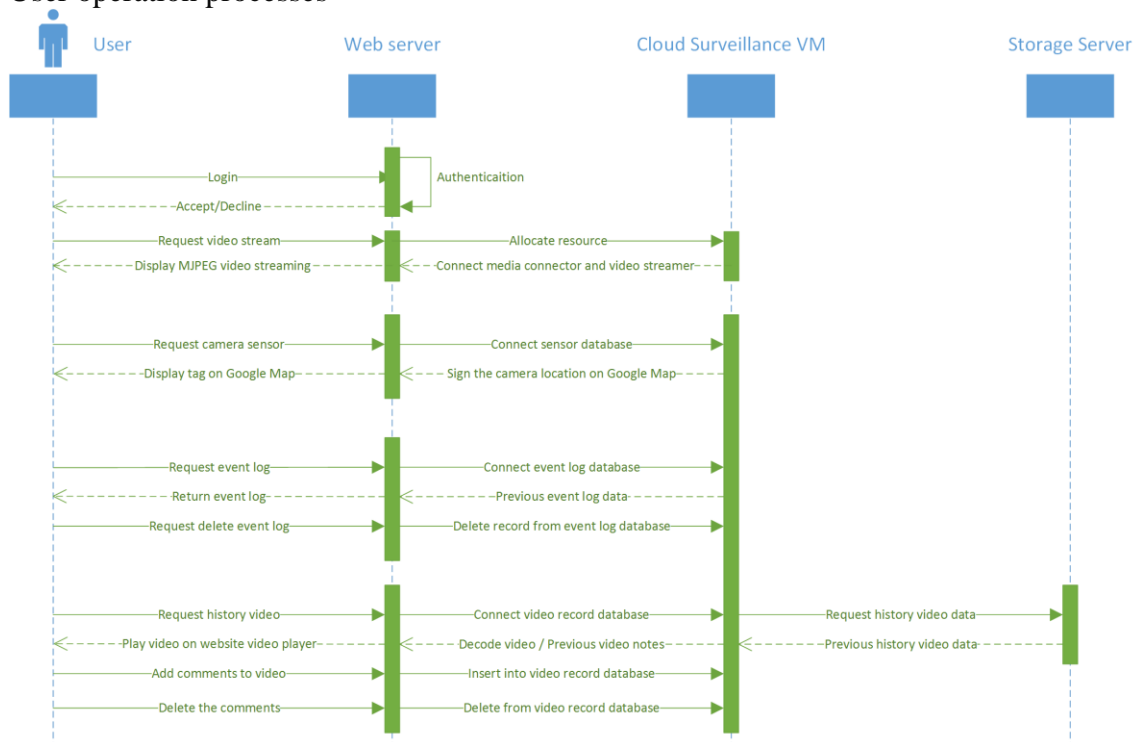


Figure 3.4 User operation flow chart

If the users pass through the step of user authentication, a new record would insert into the database, including access time, operation time and the time of ending visit. During each time, a user is operating the process while the recorder will update the database shown as Figure 3.4.

- Request the real-time video stream

When a user requires view historic video footages, there are two ways to view the videos. At first, the users could click the hyperlink specified to an event log. In the second way, the users use the video player for watching any videos they want.

- Set comments on the history video footage

The system allows users to set comments on the surveillance videos for other users to watch. The users are able to add multiple comments on each video footage.

3.2.4 Designing of subsystem

To increase the performance of cloud based surveillance system, we use QNAP TVS-682 private cloud as an access server. Our CISS adopts Microsoft Hyper-V to guarantee that the surveillance VM can efficiently be scheduled. The modules in a surveillance system will be discussed in Section 3.2.5.

We used C# based camera SDK to obtain streaming videos. This camera SDK provides outstanding compatibility in three aspects:

- It supported most IP cameras.
- The video streaming by C# camera SDK supports codecs shown in Table 3.2 and protocols support:
 - Session Initialization Protocol (SIP)
 - User Datagram Protocol (UDP)
 - Session Description Protocol (SDP)
 - Real-Time Streaming Protocol (RTSP)
 - Real-Time Transport Protocol (RTP)
 - Real-Time Control Protocol (RTCP)
 - H323 Protocol
- The system requirement for C# camera SDK in five aspects:
 - CPU requires 3GHz or faster processor
 - RAM should be more than 2 GB
 - Hard disk needs 1 GB or more free space
 - Operating system only supports Windows OS, and the version of Windows includes Windows Vista, 7, 8, 10 and Server 201X.
 - The software for development requires Microsoft Visual Studio 201X and running environment need.NET Framework 3.5/4.0/4.5.

Table 3.2 C# camera SDK supported codecs

Codec name	Application	Implementations
H.264	Recording, compression and distribution of video content	DivX, Xvid and Nero Digital
mp4v-es	Recording, compression and distribution of video content	Low-resolution surveillance IP camera
G.711	Voice recording	Multifarious VoIP software
G.726	Voice recording	Various VoIP software

Taking the security and maximum energy conservation into account, we choose QNAP TS-253 Pro private cloud as our storage server that provides AES 256-bit encryption and rating power consumption. The iSCSI known as IP-SAN (Storage area network) is a storage technology based on Internet Protocol (IP) and SCSI-3 protocol. The iSCSI uses TCP ports 3260 for the protocols itself. Utilizing Internet Small Computer System Interface (iSCSI) is easy to support Windows failover cluster and cluster shared volume. Setting iSCSI for Windows Server 2012 R2, the CISS is able to synchronize video data from the storage server. We create an FTP server VM and the OS still is Windows Server 2012 R2. In storage server, we build iSCSI LUN and utilize iSCSI initiator for connecting the target and setting up the FTP service for Windows Server. Both of cameras are supporting FTP, to configure the camera so as to link to our iSCSI target. In surveillance VM, it also connects the iSCSI target to achieve file synchronization.

There are two types of mainstream web servers, Internet Information Services (IIS) and APACHE (Woo, Joh, Alhazmi & Malaiya, 2011). In Table 3.3, we have compared two types of web servers. Finally, we choose IIS 8.0 as our web server. The reason is that our project applies .NET technology, and the Windows Server 2012 R2 already includes IIS 8.0. Because our system adopts MySQL database, we consider using the PHPMyAdmin to manage our database, the web server for running PHPMyAdmin is based on APACHE. To achieve the goal that users can freely access data from anywhere, we set up the ownCloud APP on APACHE server and also MySQL as the database.

Table 3.3 Comparison of web server

Features compare	IIS 8.0	APACHE 2.0
Reliability	High	High
Language support	ASP.NET/PHP (Fast CGI)	ASP.NET/JSP/PHP
Open source	No	Yes
Running platform	Windows	Windows/Unix/Linux
Security	High	High
Maintenance difficult	Medium	High
Performance @ ASP	High	High

Our CISS runtime environment is based on .NET 4.5, so the environment requires an installed .NET framework 4.5. In the first step, downloading Microsoft .NET Framework 4.5 is required; after the installation, how to set the pool configuration is shown as Figure 3.5.

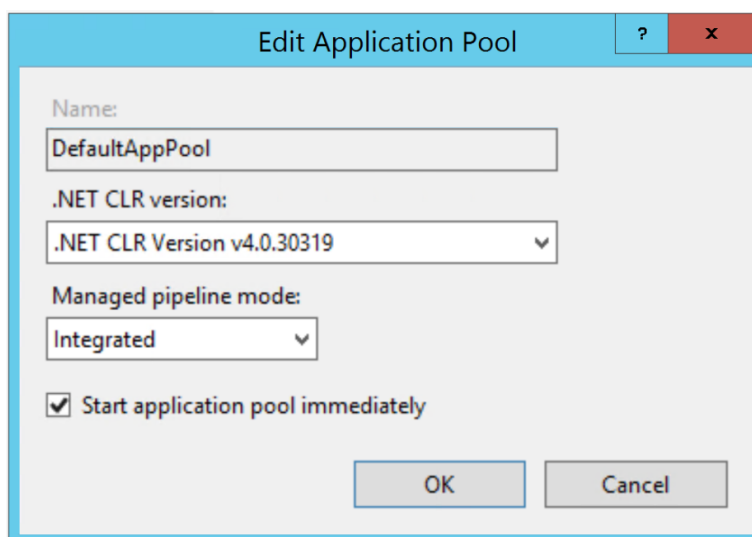


Figure 3.5 Setting of application pool

As an open source database, MySQL is one of the RDBMS (relational database management system), it usually applies to web APIs. We use MySQL as the database for our CISS due to free, safe and supported widely.

Function 3.3 Database connection
Operation: System connects MySQL database
Effect: Allow exchange data between website and database
Utilize MySQL Data Provider, set database connection information as forms to save in web.config file
<pre> <system.data> <DbProviderFactories> <remove invariant="MySql.Data.MySqlClient" /> <add name="MySQL Data Provider" invariant="MySql.Data.MySqlClient" description=".Net Framework Data Provider for MySQL" type="MySql.Data.MySqlClient.MySqlClientFactory, MySql.Data, Version=6.9.8.0, Culture=neutral, PublicKeyToken=c5687fc88969c44d" /> </DbProviderFactories> </system.data> </pre>

MySQL connector is an ADO.NET driver for MySQL database. Function 3.3 shows the connections to the CISS. ADO.NET provides data access services to the relational and non-relational system in .NET framework, each program header of the file requires introducing namespaces `MySql.Data` and `MySql.Data.MySqlClient` as a reference.

3.2.5 Designing of system modules

In order to achieve CISS, the key functions (such as video streaming, event recording, alarm making, etc.) of visual surveillance system are necessary, the application requirement will be given in Chapter 4. Hence the modules of our system are designed as below:

- User security module

We use `Microsoft.Owin.Security` in our system, as the user security module, the authentication method we used forms authentication. Figure 3.6 shows the user login process.

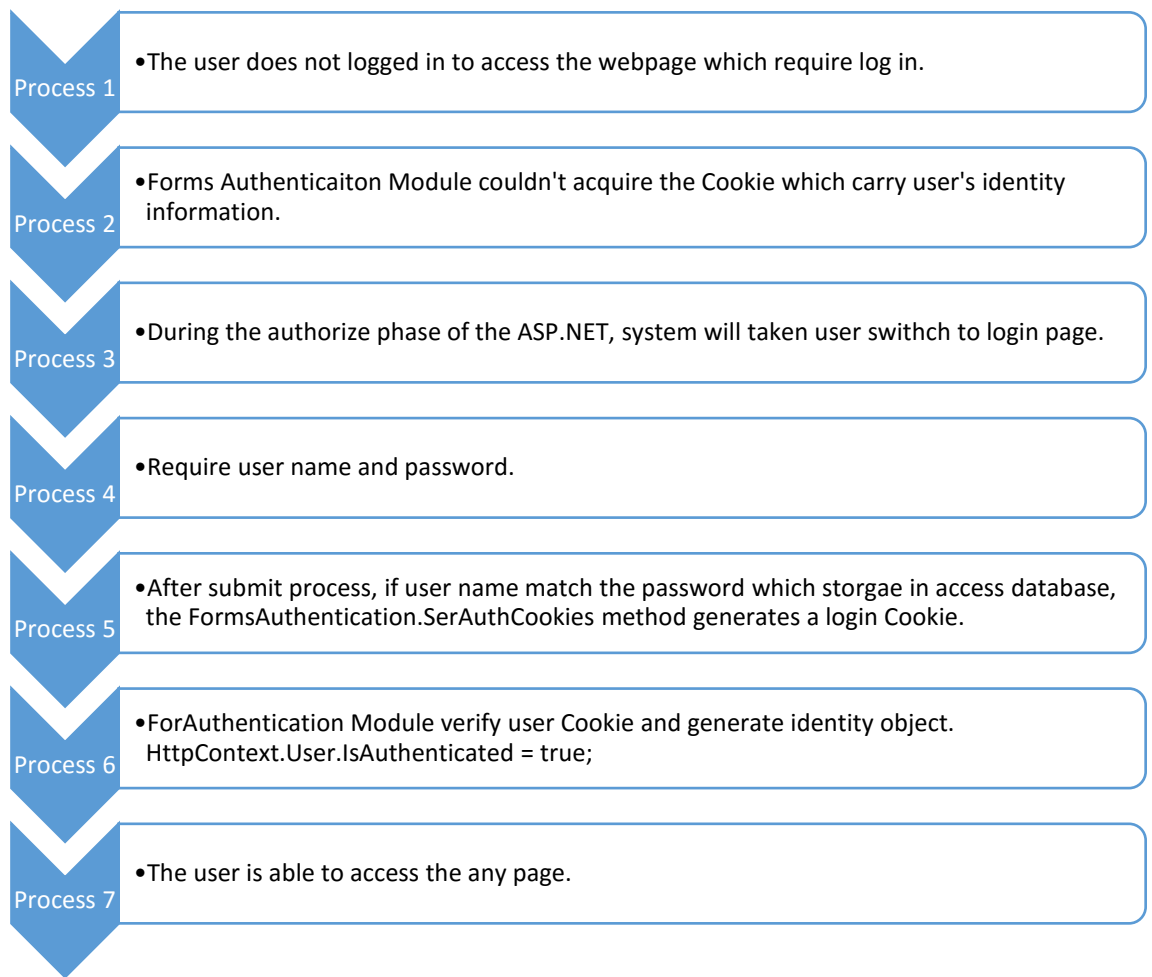


Figure 3.6 CISS user login process

- IP camera video stream

The class *MediaConnector* is integrated into the C# Camera SDK. It provides the interface connecting different *VideoHandler*, such as connecting *VideoChannel* (from the camera) and *MJPEGStreamer*. The purpose of class *MJPEGStreamer* is to establish the interface for real-time video streaming. It requires the class *OZConf_MJPEGStreamServer* as a parameter, and it includes two events: *ClientConnected* and *ClientDisconnected*. When the client camera is connected, the system invokes the methods *Start(·)* to starting video stream or *Stop(·)* to stop video stream. The event *ClientDisconnected* is as same as the *ClientConnected*.

- Web streaming

In C# Camera SDK, we utilize the class *OzConf_MJPEGStreamServer* to build a video streaming server, the first parameter of this class is its port number, which is used for providing video streaming via cloud. Function 3.4 shows the method for real-time video

streaming from a camera in a web browser. The backend method for real-time video streaming connect will be introduced in Section 4.3.2.

Function 3.4 Real-time camera video streaming in web browser
Operation: Connect VideoChannel(from camera) and MJPEGStreamer
Effect: Achieve real-time camera video streaming in web browser
Utilize MediaConnector, connect VideoChannel and MJPEGStreamer, invoking the MJPEG connection port set as an image link monitorpage.aspx file <pre><% @ Page Title="" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true" CodeFile="MonitorPage.aspx.cs" Inherits="MonitorPage" %> <img src="<%= this._streamer.ListenAddress %>" alt="image" /></pre>

- Image capture

As mentioned before, the class *MediaConnector* is able to connect different *VideoHandler*. We assert a field from class *Ozeki.Media.SnapshotHandler* that is invoked from the C# camera SDK. Then we utilize *MediaConnector* to connect *SnapshotHandler* and *VideoChannel*(camera). The module of computer vision will automatically detect the events from the *VideoChannel*, NVA module through event handler controls image capturing module.

- Video recorder

We create a MPEG-4 recorder from class *Ozeki.Media.MPEG4Recorder*, as same as image capturing, event driven is the key to trigger video recorder module. Figure 3.7 introduces the function: video recorder. This function is based on the module: Finite State Machine (FSM). In the FSM there are three events,

- Event detected

Computer Vision (CV) module detects the events from video streams.

- Event finished

CV module is unable to detect an event from current video channel.

- Video saving completed

Video recorder module saves the recorded footage.

Three statuses:

- Video recorder module stand-by

No new event has been detected from CV module

Video recorder module has not incoming signals

Video recording module stand-by until a new event has been detected.

- Video record

If CV module has detected a new event, the module will control the video recorder, module starts recording footage until event finished.

- Video save

When CV module releases control, the video recorder will automatically start saving the recorded footages.

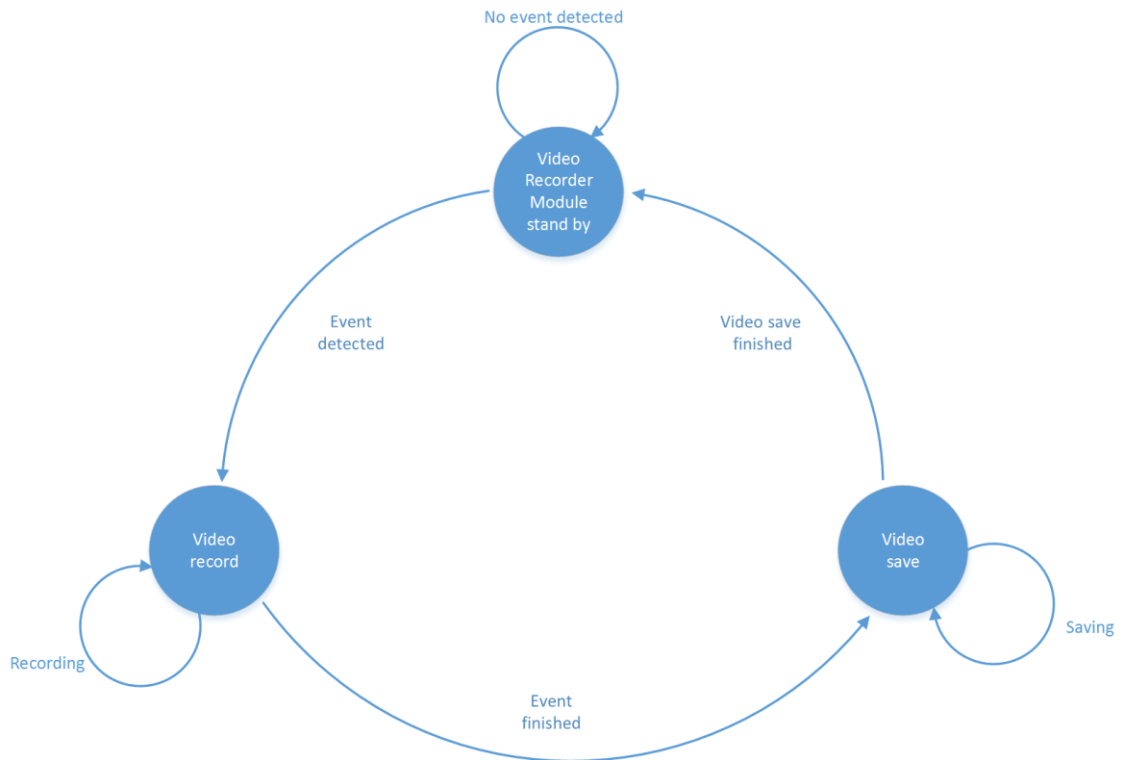


Figure 3.7 Finite state machine for module: video recorder

- Video player

Video player utilizes flash player plus XML playlist, because a flash player does not have permission to create or edit an XML file, so we create a method to update the XML file by ASP.NET. Finally, the flash player reads the XML list and loads the media files. The workflow is shown in Figure 3.8.

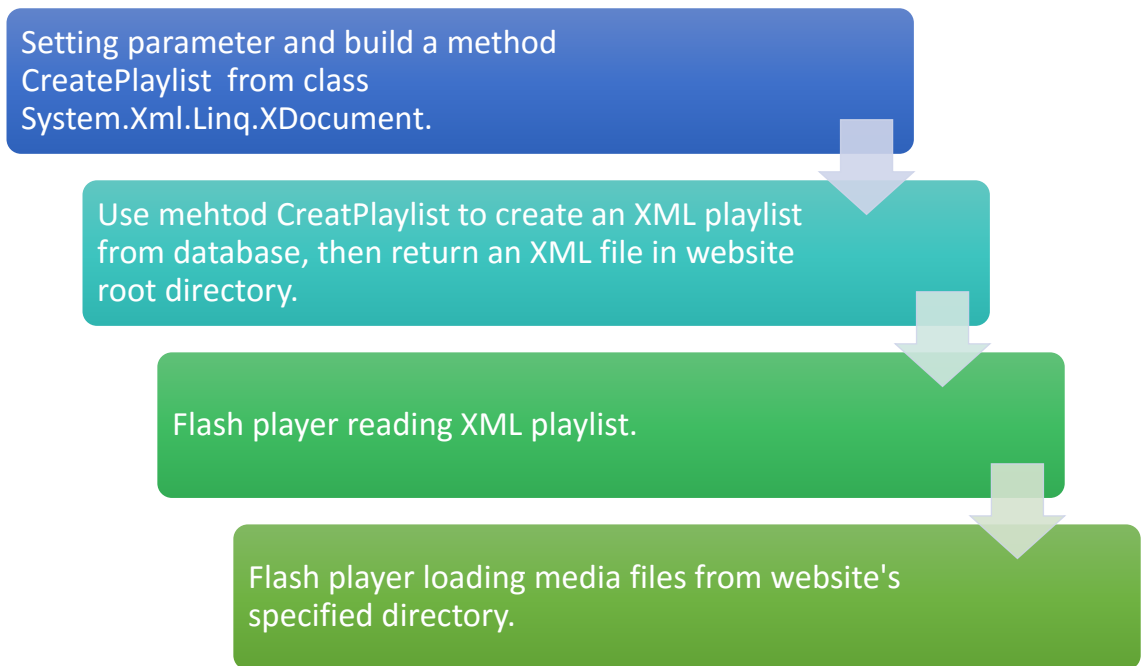


Figure 3.8 Video player workflow

- Motion detection

A motion detector is to detect moving objects. It is widely used in commercial and family applications (Chen, et al., 2015). The most common application of motion detector is the automatic door, and this sensor utilizes infrared ray to detect moving objects. But in our project, we choose computer vision (CV) module as the motion detector to substitute physical sensors.

OpenCV is the most popular computer vision platform which is based on C++ as the development platform. We maximize optimization of the performance for our CISS system, however, applying cross-platform is not the best solution. Hence we find a computer vision module that the platform is based on C#. Through the Internet, we seek the newest vision of NVA SDK to provide a perfect solution for CISS. The NVA ToolKit provides the method of motion detection that is able to connect with our camera.

- Face detection

Face detection is the effective function in visual surveillance (Yi, et al., 2012) (Anghelescu, Serbanescu & Ionita, 2013). The NVA Toolkit is the NVA.dll, it is possible to achieve face detection in real time. This Toolkit is working for the algorithms and tools that are crucial for the CV module. C# based camera SDK for media connector can connect video channel, image processor handler, and frame capture, then create an object from class

IFaceDetector (from NVA Toolkit), finally connect this object with the frame captured to conduct face detection for every ten frames.

- Event recorder, Alarm making and push service

Figure 3.9 shows the procedure of an event based alarm making module. The main function of this module is that when an event occurs, the system could automatically monitor an object in real time and record the event, send a message with event photos to the server. CV module controls the video recorder to generate an event log and store it into iSCSI. Finally, the system updates the CISS database including video player list.

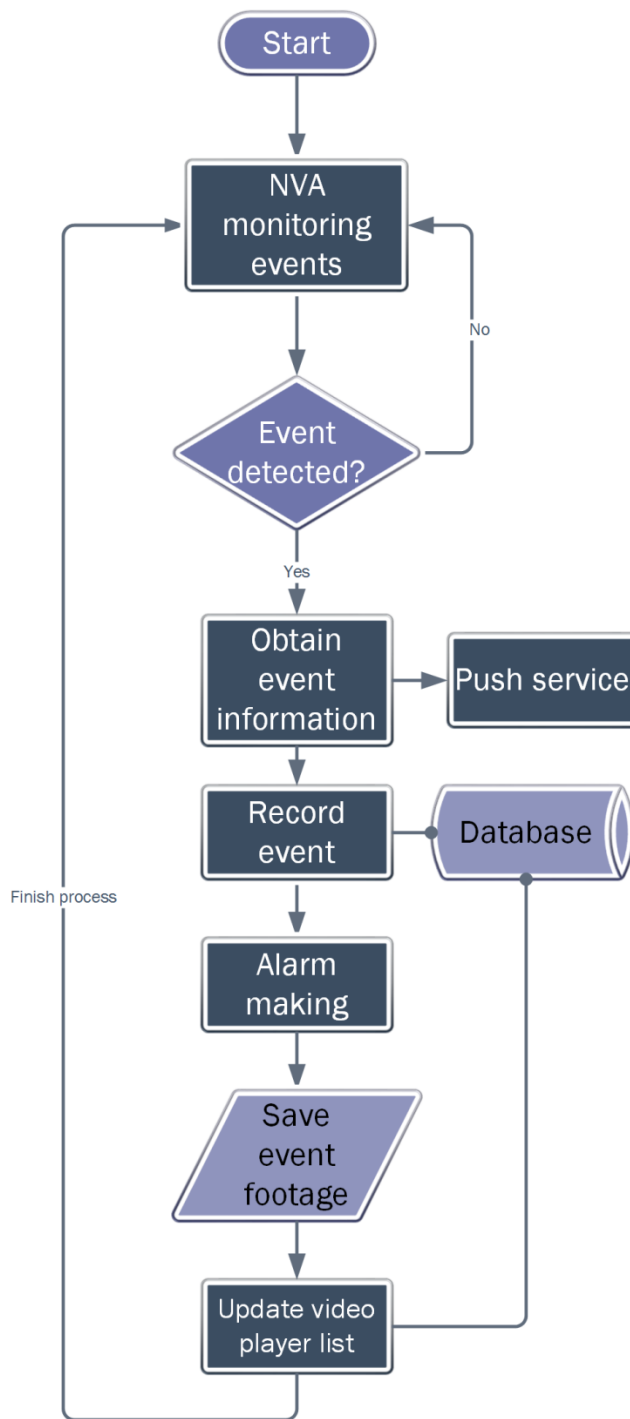


Figure 3.9 Event based alarm making module flow chart

3.2.6 Setup experimental environment

We have discussed cloud deployment models in Section 2.1.3. Private cloud provided high-level data security and service level agreement (SLA). Both of QNAP TS 253 Pro (storage server) and TVS 682 (surveillance server) are perfect as a private cloud solution. All these cloud servers allocate behind the firewall, even our storage server is only connected to the Intranet (no Internet connection).

There is a question, if a storage server does not even connect to the Internet, how to achieve freely access data anywhere? The answer is to use Camera + CISS + iSCSI LUN + FTP VM + ownCloud APP (Irene & Dhanalakshmi, 2013), the relationship between those subsystems is shown in Figure 3.10.

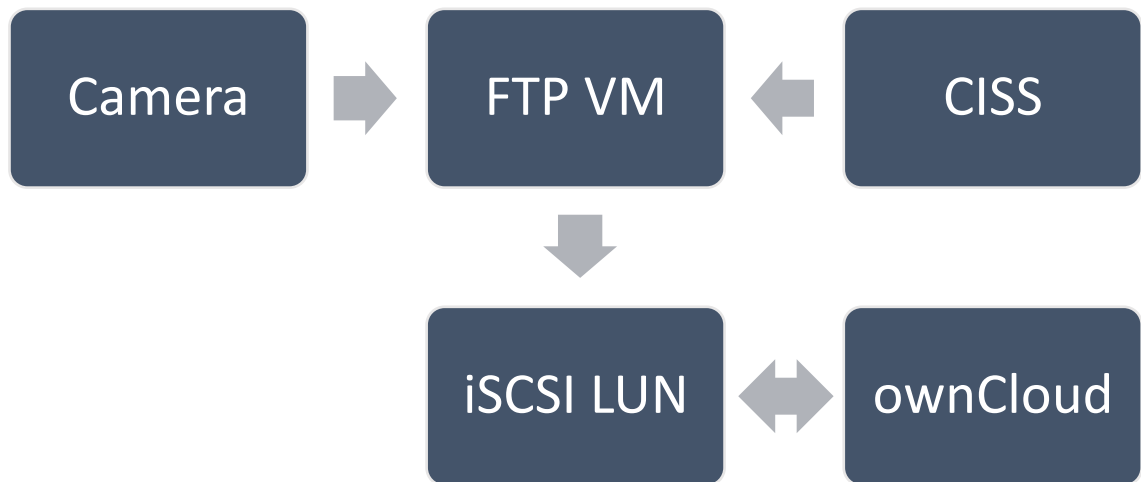


Figure 3.10 The system is running on a private cloud

Our public cloud supplier is Microsoft Azure. We deploy the user authentication database on the public cloud. Both of CISS and ownCloud APP user authentication operation need access the user table from the MS Azure MySQL database.

If a hacker attacks the cloud server, the server will be under pressure, in severe cases the server will be crashing. But for a decentralized authentication database and surveillance database, when a user authentication server turns down, the other subsystems could not be affected.

3.3 Limitations of the research

Our research is confronted with two limitations: first, there is not enough research environmental and equipment, the other is insufficient implementation time.

Our development was based on PHP programming language. Two reasons led us to transfer C# ASP.NET. The first is that IP camera only supports C++ or C#, while we expect CISS to use B/S architecture and run based on the web page, so we adopt C#. We

develop a cloud synchronization system using our familiar PHP framework *ownCloud* and FTP + iSCSI LUN + OwnCloud APP linkage to achieve data synchronization and push notification.

Due to environmental and equipment constraints, we could not shoot videos of vehicles on road, so license plate recognition module is not added. We use the license plate photo to align the camera and then test the license plate recognition module, the result is shown in Figure 3.11. Although the recognition accuracy was unsatisfactory, there was indeed a possibility of accurate recognition.



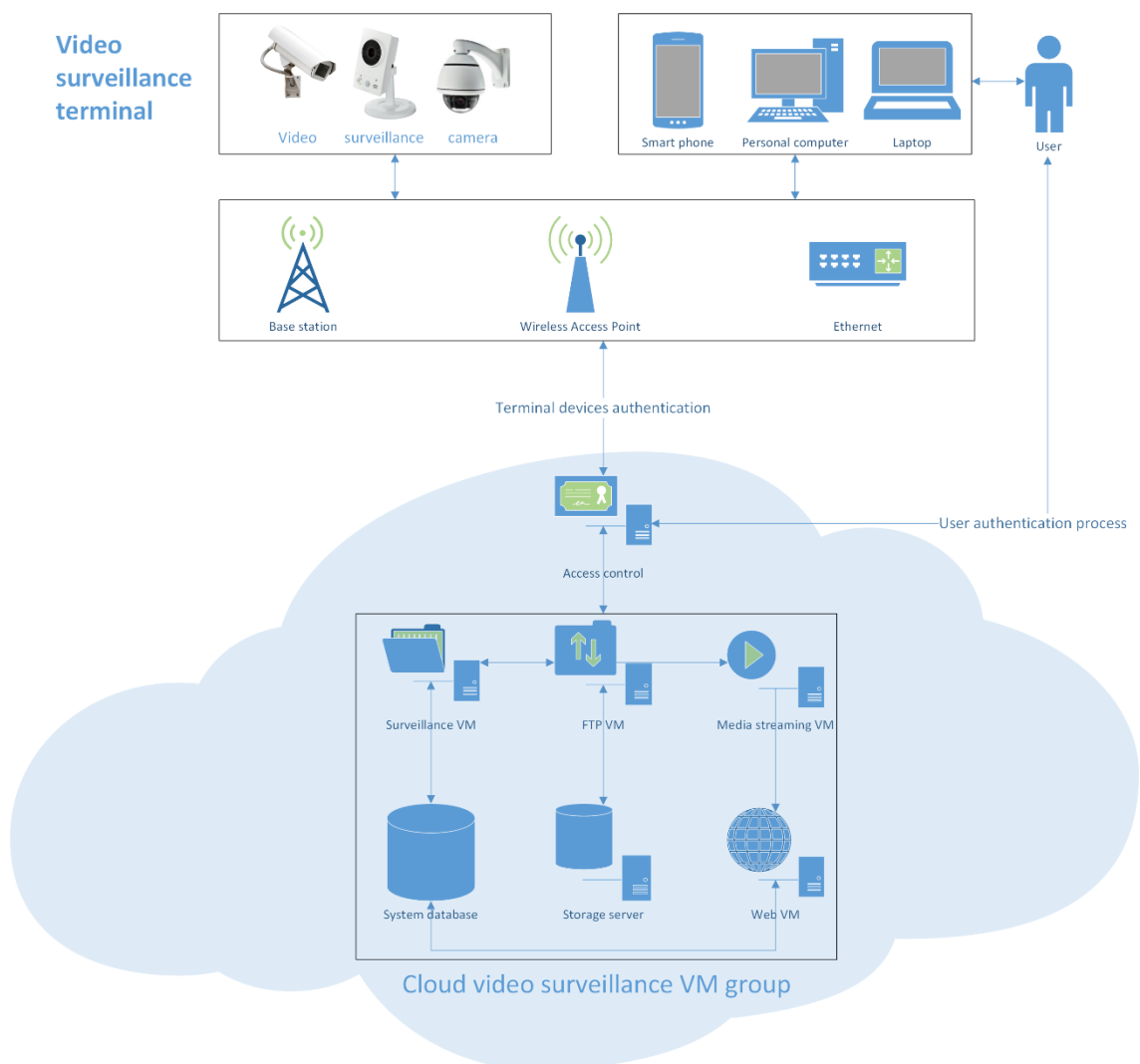
Figure 3.11 Car plate recognition module

Chapter 4 System Implementation

This chapter introduces a cloud based intelligent surveillance system (CISS) and details of the implementation. We demonstrate the functionalities in each component of this system. In this thesis, we enhance system architecture in Chapter 1, then we achieve each subsystem designed in Chapter 3. Finally, we implement and test the functions of each module in CISS.

4.1 System architecture

It has a very long time since analog signals were adopted in visual surveillance. However, times have changed, new video standard that defines the resolution such as HD, FHD, UHD, etc. became more and more popular. The current visual surveillance system faces challenges, such as transferring and storing for high definition surveillance video have resulted in issues of network bandwidth or storage capacity shortage, etc. The CISS is able to overcome the weaknesses of the traditional video surveillance model. The traditional video surveillance systems consist of two components: surveillance data center and video cameras.



Cloud video surveillance center

Figure 4.1 Topology diagram of CISS

In Figure 4.1, the CISS includes visual surveillance terminal, system security server, and CISS VM group. An IP camera as the visual surveillance device is used for collecting surveillance videos through networks (Base station, Wireless AP or Ethernet). Access server has been used for authenticating users and surveillance terminals, it is also used to record user's actions and camera status. The CISS VM group includes surveillance VM, FTP VM, Media streaming VM, Web VM, storage server and system database (user information database and surveillance database).

The surveillance VM is used to NVA (CV) module, and the CV module also controls the module of video recorder shown as Figure 3.9 when the CV module releases its control, the video recorder will automatically start saving the recorded footages. Finally, the videos through FTP protocol will be transferred to FTP VM and stored in the storage server.

Surveillance camera and surveillance VM transfer surveillance data through FTP, the receiver is a storage server, but the server would be less secure, if a hacker attacks FTP, the storage server may lead to system shutting down. At the worst case, the system maybe loses all data. On the FTP server, the first iSCSI service using iSCSI initiator is to discover and connect the target disk which is iSCSI LUN from the storage server, and the FTP site's root directory physical path is the target disk.

The storage server is responsible for supplying data storage space. The storage pool is divided into two parts including data volume and iSCSI LUN. We created a backup task for iSCSI storage every week. Media streaming server provides video streaming for the web server. The purpose of this streaming server is to reduce the web VM resource consumption and improve response speed. The web server (IIS) deployed on web VM and the necessary component of the surveillance system running in runtime environment .NET Framework 4.5 have been installed.

The CISS database is deployed on surveillance VM, it is mainly used to store camera records from CISS, and user authentication database is deployed in MS Azure. There are two aspects to be considered. The first is to take into account security of the system records, the other is to think over performance. The reason why the CISS database was deployed in surveillance VM is that it reduces the delay for data transmissions between the database and CISS.

4.2 Subsystem implementation

4.2.1 Surveillance system

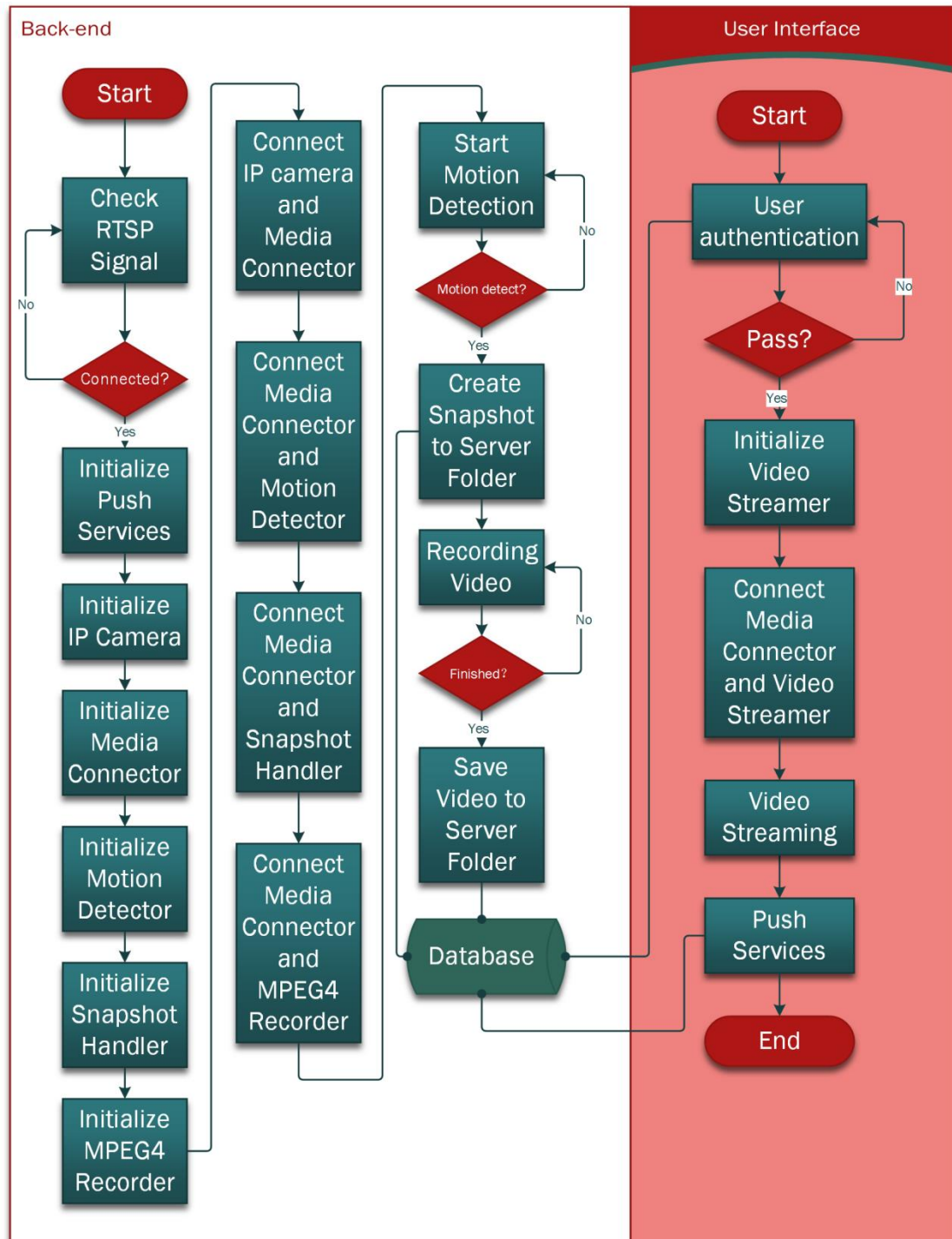


Figure 4.2 Surveillance system flow diagram

The surveillance system is one of the sub-systems in CISS which is mainly used to deal with the incoming video stream from the camera. Figure 4.2 is a flowchart of surveillance system. It has two parts, namely, foreground and back-end. The foreground of the

surveillance system is to provide a user interface, and the back-end is used to coordinate the operations between various modules of the CISS.

In Figure 4.2 we see the back-end part falls into three columns. The first column reveals initial system process for each module. Before the system is initialized, the camera must have passed a verification. It means the system can receive video signals from the IP camera. Once the module initialization is over, the module should be no longer initialized until the system will be rebooted. Because of the authentication, identification modules are independent. They are not included in any surveillance sub-systems.

The second column shows the connection process between system modules. In Figure 4.3 we see the media connector is the most important one, which provides the interface for a lot of system modules. After the connection module of IP camera connects with the media connector, the media connector provides visual interface for NVA, snapshot handler, video recorder and video streamer in real time.

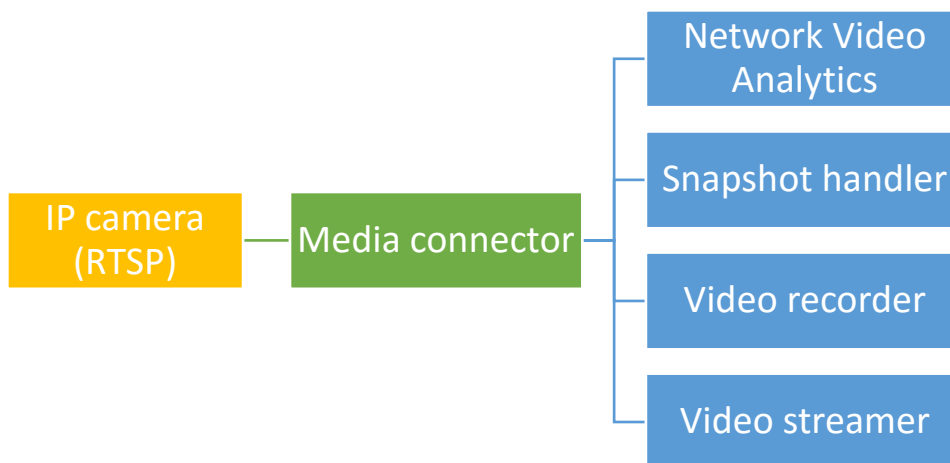


Figure 4.3 The connection bridge: media connector

The last column shows the NVA module interaction with the core module of surveillance system. The module includes snapshot handler, video recording and push notification service module. The NVA is based on event-driven to record events and control the core module to achieve the full functionalities of CISS.

4.2.2 Storage system

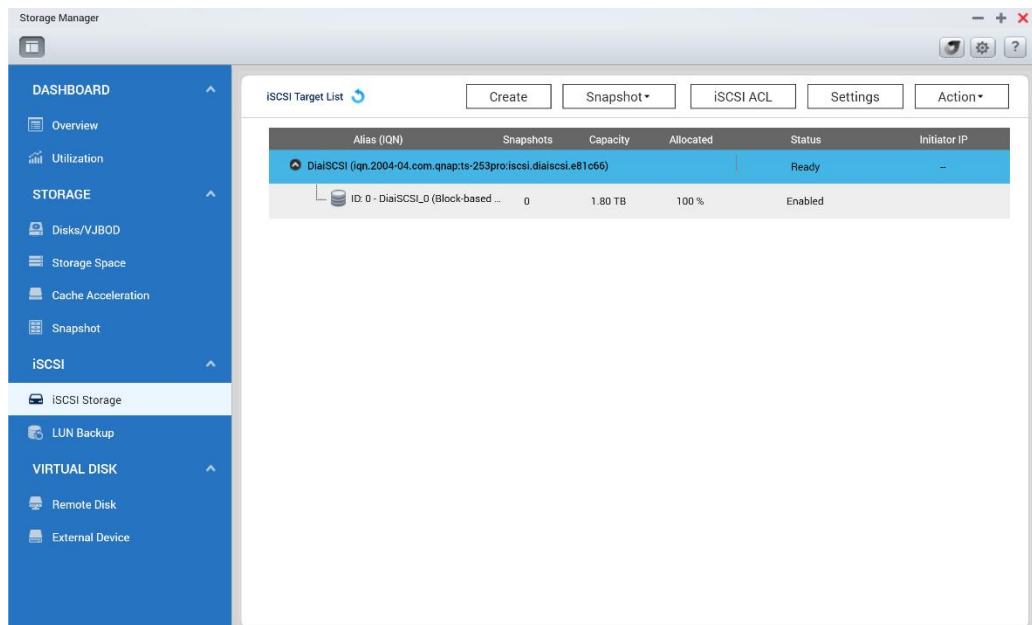


Figure 4.4 CISS iSCSI LUN

CISS storage system has three components, iSCSI LUN (storage), FTP virtual machine (VM) and storage management system. The implementation method of the storage system is explained as below.

The iSCSI is based on Internet Protocol (IP) and SCSI-3 protocol, iSCSI adopts TCP as the transport protocol, the port number is 3260. The hard disk is allocated at storage server QNAP TS-253 PRO. In Figure 4.4 we see that the capacity of iSCSI storage has 1.8 TB according to the requirements of data capacity, the iSCSI can achieve expansive capacity. As shown in Figure 4.5, the iSCSI target using CHAP authentication includes username and password, clustering access to our target from multiple initiators can be accepted. The user's IQN must be added to initiator connection list so as to ensure the data security. Otherwise, the connection for iSCSI target will be declined. The backup plan for iSCSI LUN has been employed, frequency of this plan is per week based, the backup files will be stored in both storage server and surveillance server.

Modify an iSCSI Target ✕

iSCSI Target
Initiators

iSCSI Target IQN: iqn.2004-04.com.qnap:ts-253pro:iscsi.diaiscsi.e81c66

Target Alias: DiaiSCSI

Use CHAP authentication Enable clustering access to the iSCSI target from multiple initiators

Username: Diadawnfly CRC/Checksum (optional) ▼

Password:

Re-enter Password:

Figure 4.5 iSCSI target CHAP authentication

FTP virtual machine is running on the storage server, the operation system uses Windows Server 2012 R2 standard edition, the VM storage capacity has 1TB, allocated four cores from Intel Celeron J1900, and 4 GB of memory. In Figure 4.6, there are two network adapters applied to FTP VM, VM connect public network (access to the Internet) via Switch 1. Switch 2 is connected to a private network, that means there is no Internet access, only four devices connect with this network includes storage server, FTP VM, Camera 1 and Camera 2. CISS uses FTP protocol through FTP VM to transfer surveillance data and store them in CISS iSCSI LUN. Users can manage system data via WebDAV, FTP or storage management system.

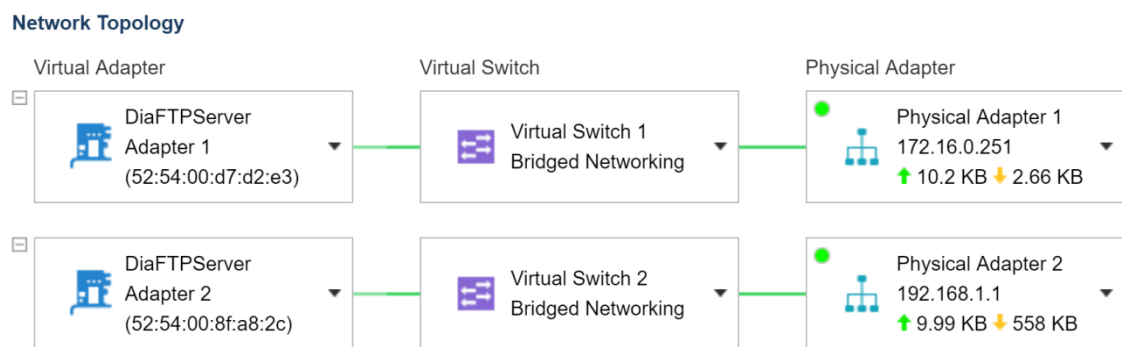


Figure 4.6 Topology of FTP VM network

The *ownCloud* is an open source StaaS application, we apply it to a storage as a Service (StaaS) (Martini & Choo, 2013), which is used to manage the system data. The *ownCloud* can be separated into two parts: server software and client software. Server software is running on the web server to provide StaaS and client software consists of a client application and web interface (Figure 4.7). In Figure 4.7 we see that the user interface could play videos using a web browser, the users also can download the files to local PC

as a backup or delete the files from iSCSI LUN. The system also allows user mapping drive to Windows system. The command line of drive connection is defined as:

```
net use Z: https://drive path /remote/WebDAV /user: username password
```

The Z means mapping to the computer Z drive. *Drive path* is the web domain name or IP address. *Username and password* are the logging user's name and password. Taken our storage mapping in this thesis as the example, the command should be:

```
net use Z: https://www.diasama.com:12356/remote.php/webdav /user: Diadawnfly aut
```

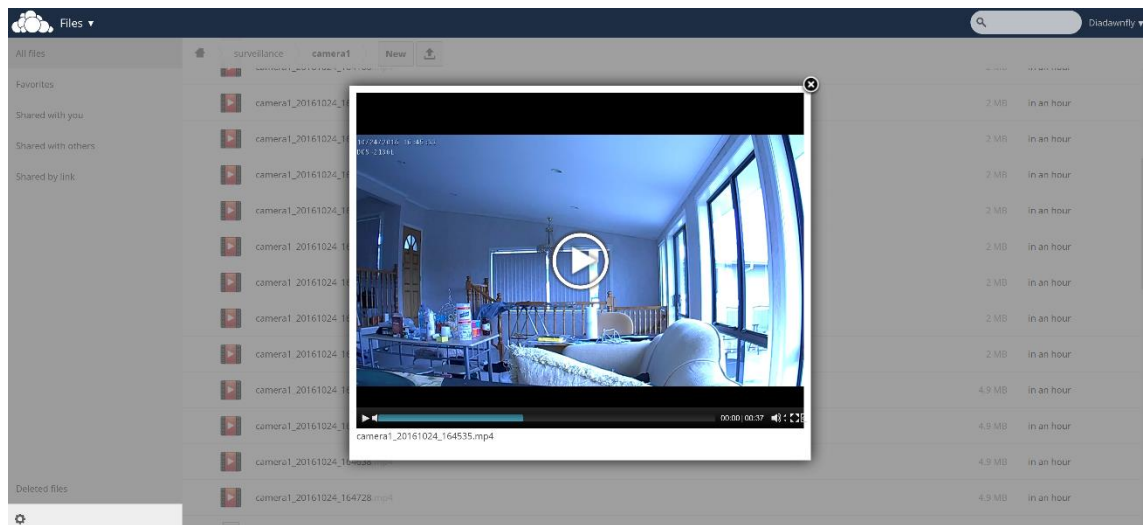


Figure 4.7 Storage management system (Storage as a Service)

Users are not limited to use StaaS to manage the storage system. There are other ways, such as Remote Desktop, WebDAV, FTP, or iSCSI LUNs. The original intention why we used ownCloud is that the user has a system with pretty rich experience of data management, thus he could easily achieve the goal of cloud storage. In the further research, we will develop an isolated storage management software for CISS.

4.2.3 Web server

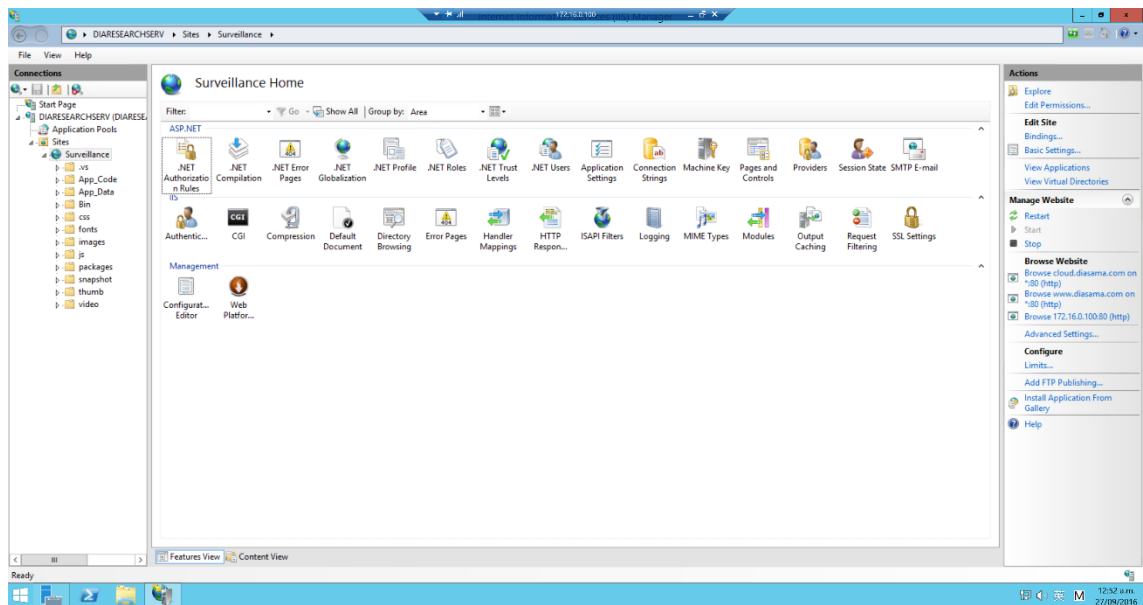


Figure 4.8 CISS user interface software deploys on server-side

The web server is used to provide services related to content. The server responds user requests and provides the client with the system data. Typically, users can utilize a web browser to access the website. However, the CISS is applied to Brower / Server structure. The user interface is based on cloud and HTTP protocol. The Internet Information Services (IIS) is one of the most famous web servers, it is deployed and run as CISS web host in web VM. The .NET framework also has been installed on web VM. Figure 4.8 shows the CISS web user interface deployed on server-side.

The user enters one of the two URLs to access CISS web interface. In web VM, the Apache web server also has been deployed for running ownCloud. We use the same domain name but different connection protocol and port, the connection protocol is HTTPS, and the port number is 12356.

4.2.4 System database

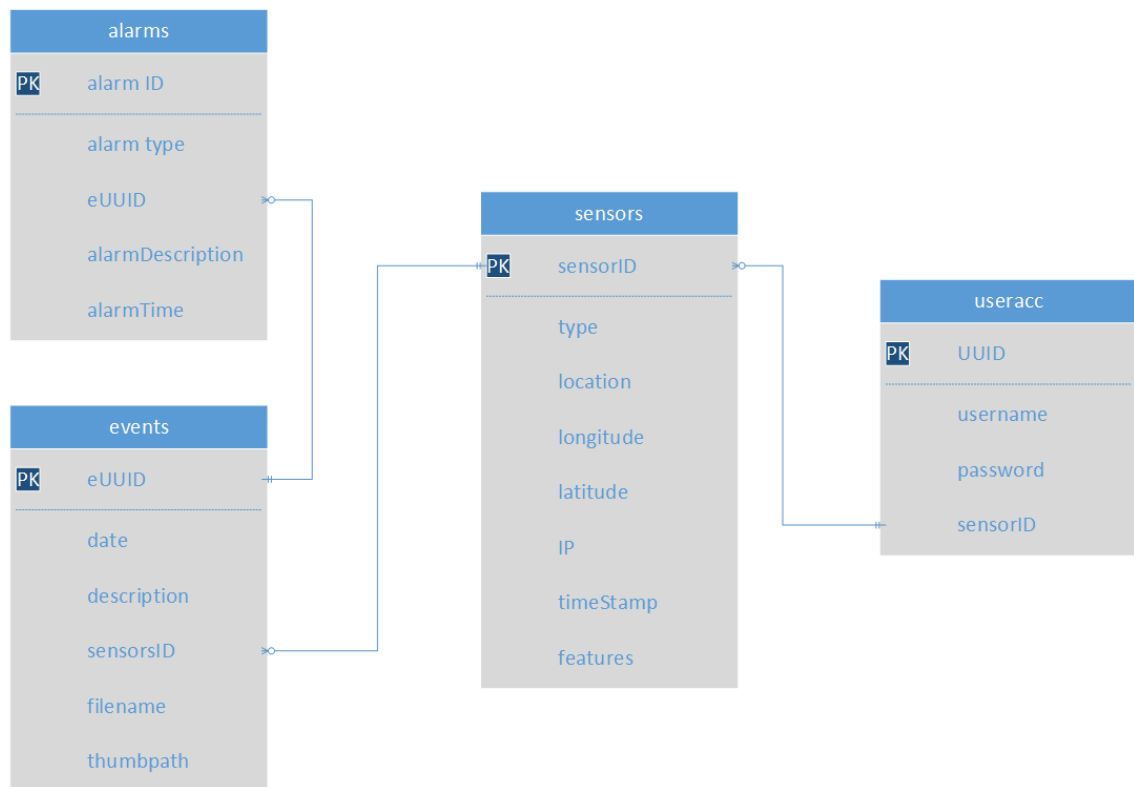


Figure 4.9 Diagram of CISS database

In this section, we will describe the database designed for CISS. Figure 4.9 shows the CISS database topology diagram. There are four tables designed for CISS. The tables include user acct, sensors, events, and alarms. The details are shown in Table 4.1, 4.2, 4.3 and 4.4.

Table 4.1 User account table in the database

Colum	Data type	Length	Auto INC	PK	NOT Null
UUID	INT	32	Yes	Yes	No
Username	VARCHAR	64	No	No	No
Password	VARCHAR	128	No	No	No
SensorID	INT	32	No	No	No

Table 4.2 Sensor table in the database

Colum	Data type	Length	Auto INC	PK	NOT Null
SensorID	INT	32	Yes	Yes	No
Type	ENUM	0	No	No	No
Location	VARCHAR	128	No	No	No
Height	FLOAT	128	No	No	Yes
Longitude	VARCHAR	8	No	No	No
Latitude	VARCHAR	8	No	No	No
IP	VARCHAR	15	No	No	No
TimeStamp	TIMESTAMP	0	No	No	No
Feature	TEXT	128	No	No	Yes

Table 4.3 Events table in the database

Colum	Data type	Length	Auto INC	PK	NOT Null
eUUID	VARCHAR	32	No	Yes	No
Date	VARCHAR	64	No	No	No
Description	VARCHAR	128	No	No	No
SensorID	INT	32	No	No	No
Filename	VARCHAR	128	No	No	No
Thumbpath	VARCHAR	128	No	No	No

Table 4.4 Alarms table in the database

Colum	Data type	Length	Auto INC	PK	NOT Null
AlarmID	INT	32	Yes	Yes	No
AlarmType	VARCHAR	32	No	No	No
eUUID	VARCHAR	32	No	No	No
AlarmDesc	TEXT	256	No	No	Yes
AlarmType	VARCHAR	32	No	No	Yes

In CISS, we installed MySQL Data Entity from ASP.NET NuGET, and the MySQL Data Provider is used for database connectivity. The syntax of MySQL database connection is defined as:

```
connStr = "server=database URL; user=database username; database=database name;
port=database port; password=database user password;";
```

Creating an object `connStr` is so useful to provide database connection information. The connection information includes *Database URL*, *username*, *database name*, *port* and *password*. Taken consideration of user authentication database in CISS, the format is shown as:

```
connStr = "server=au-cdbr-azure-southeast-a.cloudapp.net; user=b8b90ee0947dc4;
database=surveillance; port=3306; password=8ec7745a;";
```

Creating a new object `conn` from class `MySQLConnection` has been used to data manipulation, the syntax format is:

```
MySQLConnection conn = new MySqlConnection(connStr);
```

We use the command `conn.Open()` to start the MySQL database connection, then CISS executes the command `MySqlCommand cmd = new MySqlCommand(sql, conn)`. The `sql` is an object that includes data operation command (SELECT, DELETE, INSERT, UPDATE, CREATE, DROP etc.).

4.3 System modules

4.3.1 IP camera based video streaming

This module implements the real-time video streaming function, IP camera (private network) can be used to transfer data for surveillance virtual machine within public network. We utilize Routing and Remote Access Service (RRAS) which maps two cameras to the public network (Internet). Camera 1 occupies the port 554 and Camera 2 uses the port 555.

In the CISS, at the front of our program, we create an object camera from class `IPCameraFactory` to catch IP camera video. The syntax of connection with IP camera is defined as:

```
_camera = IPCameraFactory.GetCamera("rtsp://Camera URL:port",  
                                     "username","password");
```

It is used to create an object `_cameraX` for providing multiple IP camera video streams. The IP camera connection information includes *Camera URL*, *username*, *password*. Our connection method of two IP cameras is shown on Function 4.1.

Function 4.1 IP camera connection
Operation: Create an object <code>_camera</code> connect with IP cameras
Effect: An object can able to provide interface for real-time video streaming
<pre>public IIPCamera _camera1; public IIPCamera _camera2; protected void surveillance() { _camera1 = IPCameraFactory.GetCamera("rtsp://www.diasama.com:554 //live1.sdp","Diadawnfly", "aut"); _camera2 = IPCameraFactory.GetCamera("rtsp://www.diasama.com:555 //live3.sdp","Dia", "aut"); _camera1.Start(); _camera2.Start(); }</pre>

CISS has multiple CV modules. However, each CV module needs to capture video data from the IP camera. We use the class *MediaConnector* to coordinate the operation of these modules. The syntax establishes the media connector and CV module is defined as:

```
_connector.Connect(_cameraX.VideoChannel, _xDetector);
```

This function is used to create an object *_connectorX* so as to support specific IP camera. The module connection parameter includes one sender and one receiver. Function 4.2 shows the details of multiple media connector configuration.

Function 4.2 Media connector
Operation: Create objects <i>_connectorX</i> to connect one sender and one receiver
Effect: Achieve coordinates the operation between video sender and video receiver
<pre>public MediaConnector _connector1; public MediaConnector _connector2; protected void surveillance() { _connector1 = new MediaConnector(); _connector2 = new MediaConnector(); _connector1.Connect(_camera1.VideoChannel, _motionDetector); _connector2.Connect(_camera2.VideoChannel, _faceDetector); }</pre>

4.3.2 Web based real-time video streaming

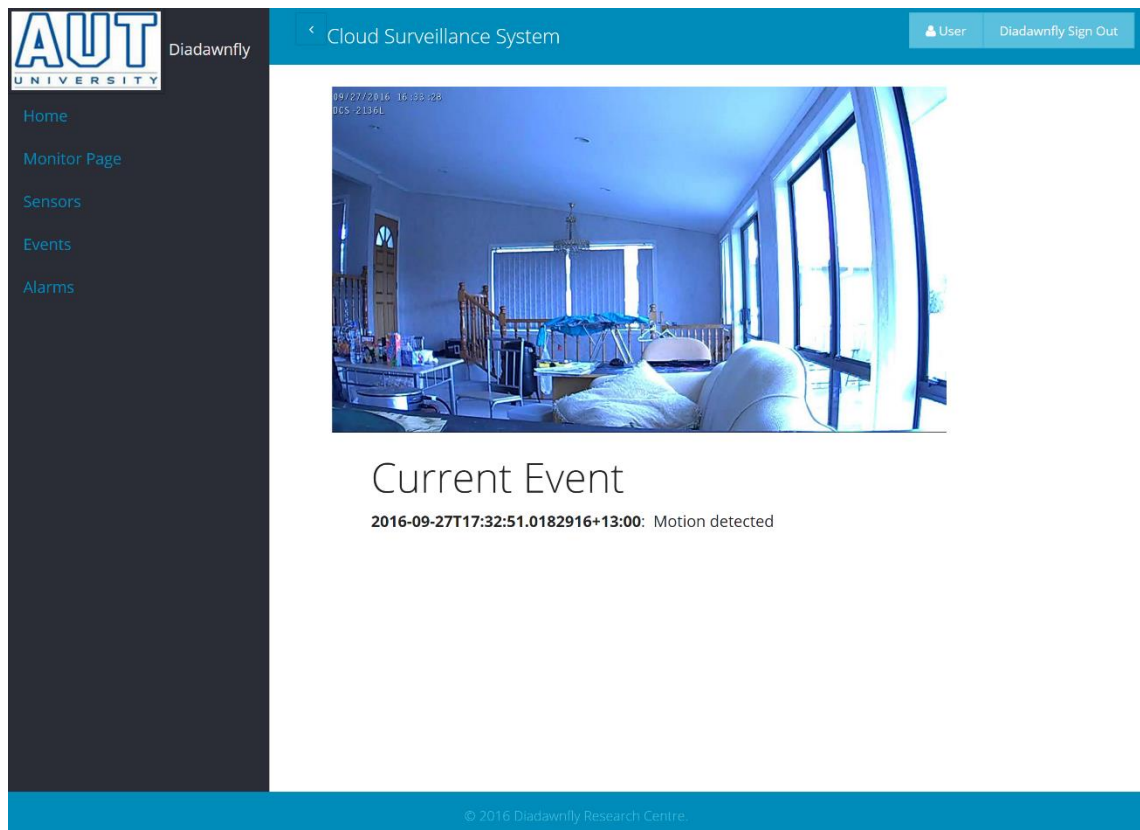


Figure 4.10 Web based real-time video streaming

CISS is based on B/S architecture. The system does not require our users to install any client software. Hence, the system must provide a web based real-time video streaming service. CISS has multiple CV modules. However, each CV module needs to capture the IP camera's video data.

Figure 4.10 demonstrates the module: real-time video streaming. The following discussion is related to its specific implementation. First, we create an object for class *MJPEGStreamer*, then configure *MJPEGStreamServer* (8 is streaming port, 25fps). We add event listener *ClientConnected* and *ClineDisconnected* for object *_streamer*. When a new user connects video streaming module, we use the function *streamer_ClientConnected* to control the user, such as starting video streaming or blocking user IP address. When a client disconnects from the stream server, the function *streamer_Disconnected* can automatically stop video streaming. The pseudocode is shown in Function 4.2.

Function 4.2 Web based real-time video streaming
Operation: Use media connector, connect IP camera (sender) and video streamer (receiver)
Effect: Achieve web based real-time video streaming from IP camera
<pre> public MJPEGStreamer _streamer; protected void surveillance() { _streamer = new MJPEGStreamer(new OzConf_MJPEGStreamServer(8, 25)); _streamer.ClientConnected += streamer_ClientConnected; _streamer.ClientDisconnected += streamer_ClientDisconnected; _connector.Connect(_camera1.VideoChannel, _streamer.VideoChannel); _streamer.Start(); } protected void streamer_ClientConnected(object sender, VoIPEventArgs<MJPEGStreamConnection> e) { e.Item.StartStreaming(); } protected void streamer_ClientDisconnected(object sender, VoIPEventArgs<MJPEGStreamConnection> e) { e.Item.StopStreaming(); } </pre>

4.3.3 Image capture



Figure 4.11 Image capture

The main task of image capturing module is to snap an image from the camera. In CISS, image capturing plays a pivotal role in image sending and receiving. In Figure 4.12, we see there are three procedures to achieve this function, the procedures are described as below:

Step 1: Establish the IP camera connection.

The first requirement of this module is to bind a sender. In this thesis, the video channel plays as a role of the sender. So the solution for image capture can utilize media connector to link them together.

Step 2: Create a snapshot from video channel.

We use the NVA module to start a snapshot, in Function 4.3 the TakeSnapshot(·). ToImage(·) method is able to get the actual image from the video channel.

Step 3: Save snapshot to the specified folder.

In image capturing module, the final step is to name the image and store the image to the destination folder. Each image should have a unique name, so we rename the photo so as to encapsulate the current date and time, and take convenience for users to check the time of the event occurred. Figure 4.11 shows the result.

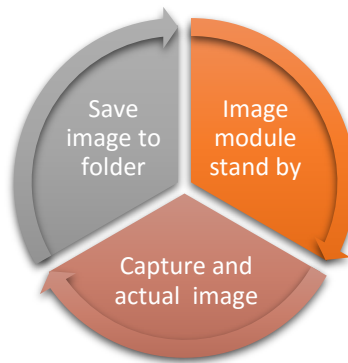
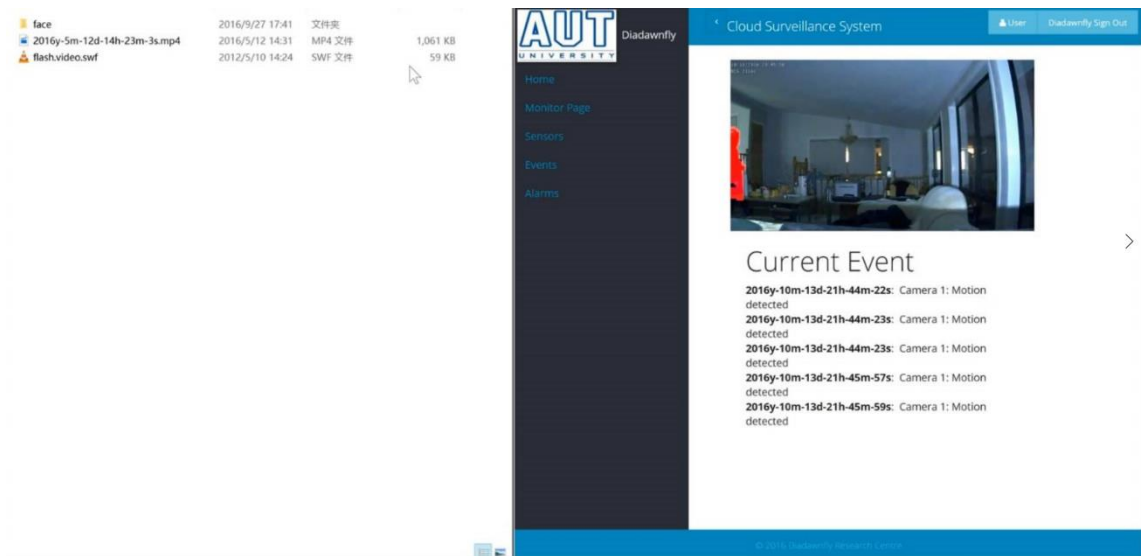


Figure 4.12 Image capture process

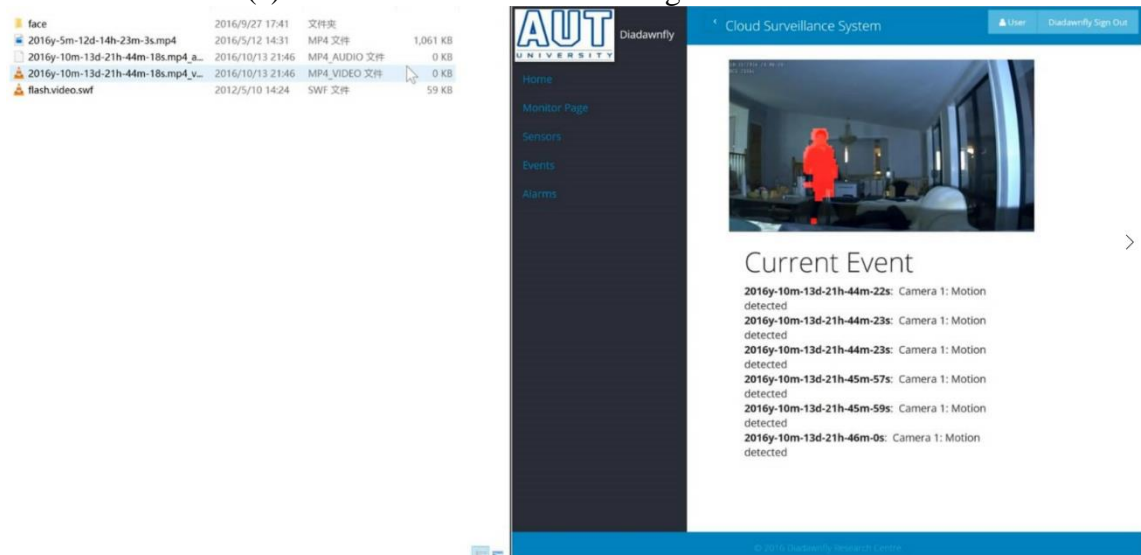
The specific methods for implementation of image capturing module are shown as function 4.3.

Function 4.3 Image capture
Operation: Capture an image from video stream
Effect: Create and save JPEG format snapshot
<pre> public SnapshotHandler _snapshotHandler; public string date = DateTime.Now.Year + "y-" + DateTime.Now.Month + "m-" + DateTime.Now.Day + "d-" + DateTime.Now.Hour + "h-" + DateTime.Now.Minute + "m-" + DateTime.Now.Second + "s"; protected void CreateSnapShot() { _snapshotHandler = new SnapshotHandler(); _connector.Connect(_camera.VideoChannel, _snapshotHandler); string path = @"~/snapshot"; string currentpath; currentpath = path + "/" + date + ".jpg"; var snapShotImage = _snapshotHandler.TakeSnapshot().ToImage(); snapShotImage.Save(Server.MapPath(currentpath), System.Drawing.Imaging.ImageFormat.Jpeg); } </pre>

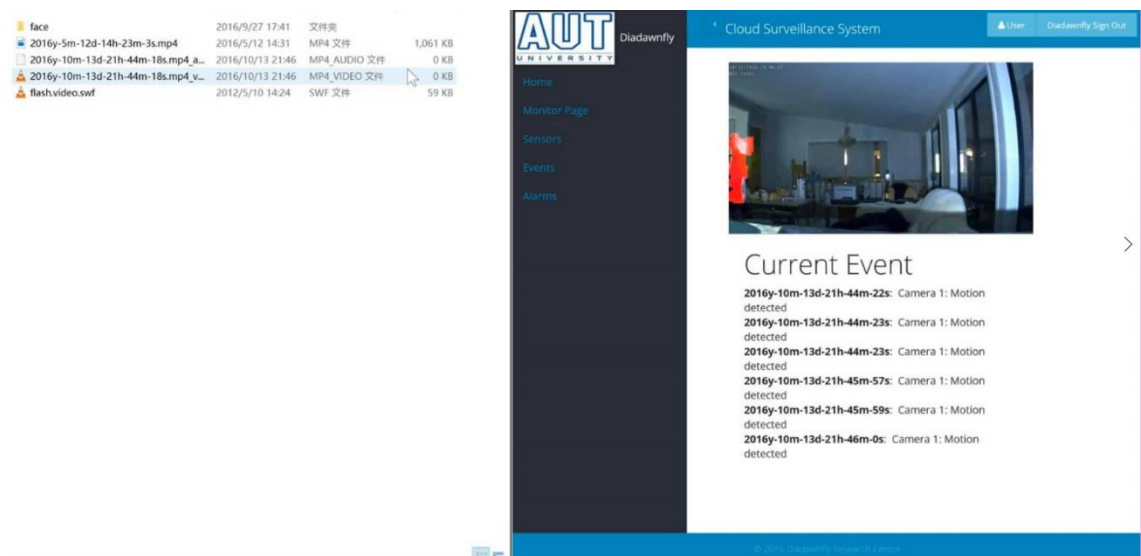
4.3.4 Video recorder



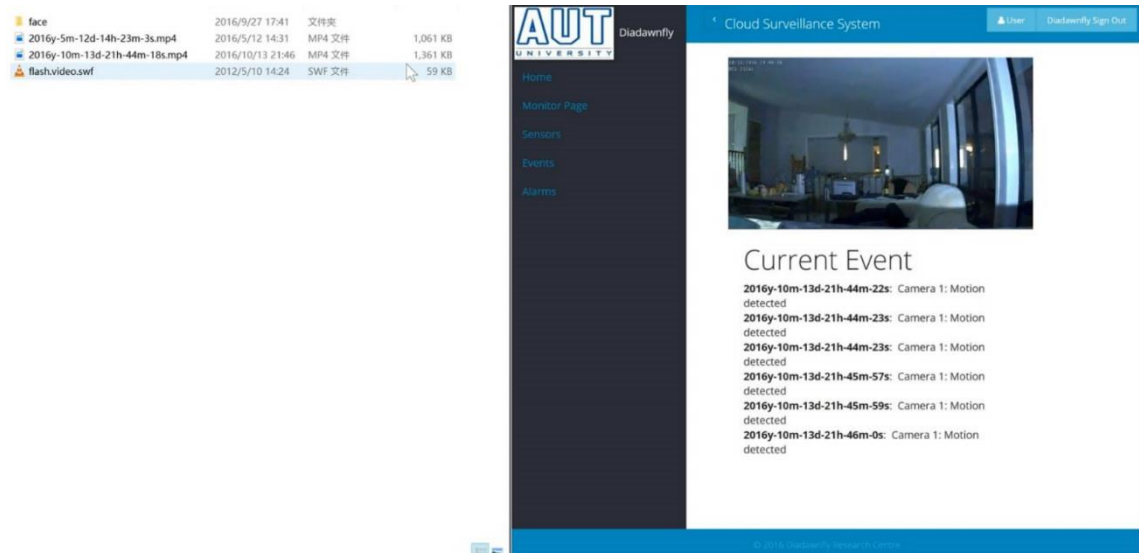
(a) Video recorder has connecting to video channel



(b) Video recorder start recording



(c) Video recorder keeps recording



(d) Video recorder save the footage and disconnect from video channel

Figure 4.13 Video recorder process

In Section 3.4.5, we mentioned that Finite State Machine (FSM) for video recorder module has three events and three statuses. In this section, our focus is on the implementation method. We use NVA event controller to manage video recorder module, once the specific sensor catches an event, the event controller calls the *StartVideoCapture(.)* method, then the video recorder starts recording the video from the video channels. The video files should include video and audio, through media connector, connect the video channel to MPEG-4 video recorder, and link the audio channel to MPEG-4 audio recorder.

Before an event is finished, we utilize *StopVideoCapture(.)* method, the event controller can be triggered with the inside parameter: multiplexing, the function *_connector.Disconnect* is used to stop recording video. Finally, the video has been stored in the destination folder. The full procedure of video recording is shown in Figure 4.13. The pseudocodes for video recorder module are presented in Function 4.4.

Function 4.4 Video Recorder
Operation: Capture video and audio from video channel
Effect: Generate a video file
<pre>public MPEG4Recorder _mpeg4Recorder; protected void StartVideoCapture () { string path = @"~/video"; string currentpath; currentpath = path + "/" + date + ".mp4";</pre>

```

_mpeg4Recorder = new MPEG4Recorder(Server.MapPath(currentpath));
_connector.Connect(_camera.VideoChannel, _mpeg4Recorder.VideoRecorder);
_connector.Connect(_camera.AudioChannel, _mpeg4Recorder.AudioRecorder);
}
protected void StopVideoCapture()
{
    _mpeg4Recorder.Multiplex();
    _connector.Disconnect(_camera.VideoChannel, _mpeg4Recorder.VideoRecorder);
    _connector.Disconnect(_camera.AudioChannel, _mpeg4Recorder.AudioRecorder);
    _mpeg4Recorder.Dispose();
}
}

```

4.3.5 Motion detection module

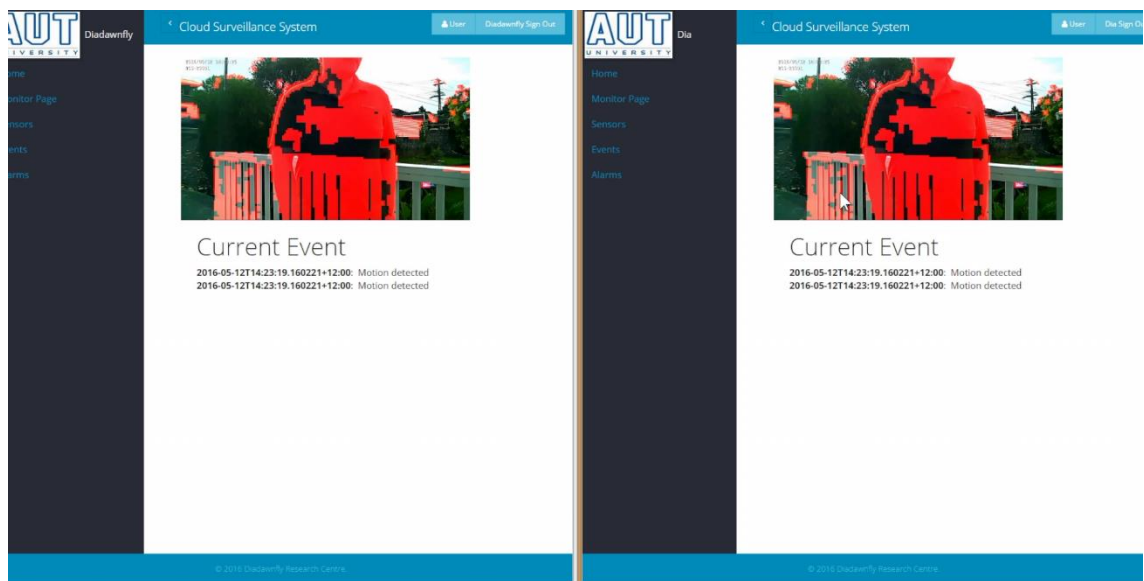


Figure 4.14 Motion detection module demo

Motion detection is one of the NVA modules in CISS. In Figure 4.14 we see a person is marked with a red rectangle, this is the result of video processing in real time by using the motion detection module. In this section, we discuss the implementation method. Motion detection module is a receiver's actions as well as the sender. The motion detector is at the middle of the module of camera video channel and the module of video streamer as shown in Figure 4.15, it receives video stream from a channel and sends the processed video to the video streamer.

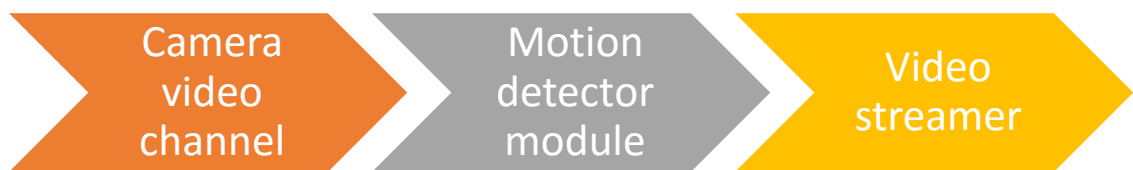


Figure 4.15 Motion detection module flow diagram

The details of working steps of motion detection module are shown as below,

- Create an object *motionDetector* from class *MotionDetector*.
- Initialize *motionDetector* at the beginning of the module.
- Connect camera video channel to *motionDetector* module.
- Determine the feature for the processed video from the motion detector, we apply highlight with red color to cover the moving objects.
- Add motion detection to event handler
- Connect *motionDetector* module to video streamer.
- Event handler monitoring events from motion detection module. If an event has been detected by the motion detector, the event handler will invoke and start additional modules.

The pseudocode of motion detection module is shown in Function 4.5.

Function 4.5 Motion detection module
Operation: Receive video from camera, after processed send to streamer
Effect: Achieve controls another module through motion detection event handler
<pre>public MotionDetector _motionDetector; protected void motionDetector() { _motionDetector = new MotionDetector(); _connector.Connect(_camera.VideoChannel, _motionDetector); _motionDetector.HighlightMotion = HighlightMotion.Highlight; _motionDetector.MotionColor = MotionColor.Red; _motionDetector.MotionDetection += _motionDetector_MotionDetection; _connector.Connect(_motionDetector, _streamer.VideoChannel); _motionDetector.Start(); } protected void _motionDetector_MotionDetection(object sender, MotionDetectionEvent e) { switch (e.Detection) { case true: PushServices(); CreateSnapShot(); StartVideoCapture(); break; case false: break; } }</pre>

4.3.6 Face detection module



Figure 4.16 Face detection module demo

Face detection is a kind of object detection in computer vision. In Figure 4.16, we see that a red box is marked on the region of a human face. The frame capturing module is able to take a picture per five frames, then send handler of the captured image for processing. Through setting the parameter of face detection, the face detection module can rapidly and exactly recognize the human face in the captured image.

The following steps introduce the whole processes in face detection module:

- Add an object *frameCapture* from class *FrameCapture* to face detection module. The module *frame capturing* works as the initial video receiver, after capturing an image, as a sender it provides the footage to image processor handler.
- Create and initialize an object from class *ImageProcessorHandler*. It is working as a receiver and sender at the same time.
- Use the object *faceDetector* from interface *IFaceDetector* as an image processor.
- Add face detector to image processor handler as CV processor.
- Set attributes for face detector:
 - DrawColor: Set the color of the square in Figure 4.16. The three parameters in DrawColor are Red, Green, and Blue (RGB), the number goes larger, the color goes darker.
 - DrawThickness: Set the square edge thickness, the number goes larger, the square edge goes thicker.
 - MinSize and MaxSize: It is mainly used to control the recognition size for the face.

- DetectionOccurred: The event handler for face detection module.
- Connect each sender and receiver, it connects the modules: video channel, frame capturing, image processor handler, and video streamer.
- Activate the module of frame capture and the module image process handler which is added to face detection processor. After the module is initialized, the face detection module starts listening to the events from the module face detector.

The pseudocode of face detection module is shown in Function 4.6.

Function 4.6 Face detection module
Operation: Receive video from camera, after processed send to streamer
Effect: Achieve controls another module through motion detection event handler
<pre> public FrameCapture _frameCapture; public ImageProcessorHandler _imageProcessorHandler; public IFaceDetector _faceDetector; protected void motionDetector() { _frameCapture = new FrameCapture(); _frameCapture.SetInterval(5); _faceDetector = ImageProcessorFactory.CreateFaceDetector(); _imageProcessorHandler = new ImageProcessorHandler(); _imageProcessorHandler.AddProcessor(_faceDetector); _faceDetector.ShowImage = true; _faceDetector.DrawColor = Color.FromArgb(255,0,0); _faceDetector.DrawThickness = Int32.Parse("2"); _faceDetector.MinSize = new Size(Int32.Parse("100"), Int32.Parse("100")); _faceDetector.MaxSize = new Size(Int32.Parse("500"), Int32.Parse("500")); _faceDetector.DetectionOccurred += _faceDetector_FaceDetection; _connector.Connect(_camera.VideoChannel, _frameCapture); _connector.Connect(_frameCapture, _imageProcessorHandler); _connector.Connect(_imageProcessorHandler, _streamer.VideoChannel); _frameCapture.Start(); _imageProcessorHandler.Start(); } protected void _faceDetector_FaceDetection(object sender, FaceDetectedEventArgs e) { foreach (var info in e.Info) { PushServices(); CreateSnapShot(); StartVideoCapture(); } } </pre>

4.3.7 Camera sensor module

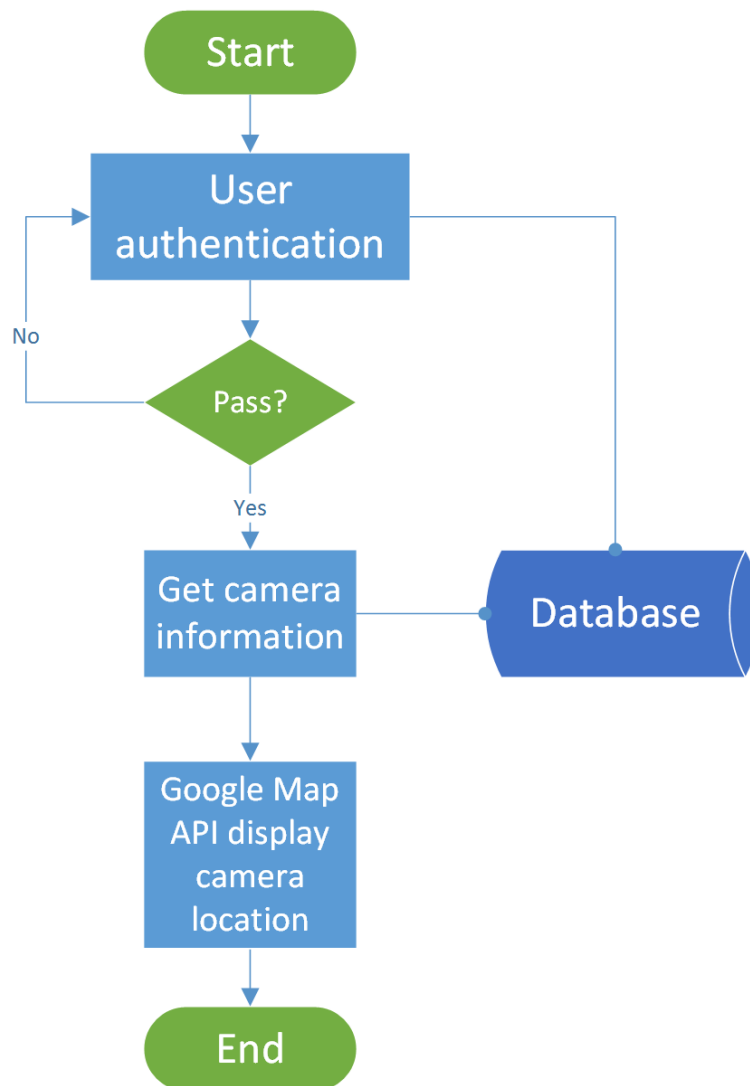


Figure 4.17 Camera sensor module processes flow chart

The main task of camera sensor module is to tag camera sensors on the Google Map (Chang, et al., 2009). Figure 4.17 shows the procedure of camera sensor module. If a user could pass user authentication, then the user is able to view the camera information. When the system receives a request from a user, the first step is to query information from CISS database, then carry the camera data back to the system, it will show the camera sensor list for the user. If the user clicks the select button, the sensor location information will be sent to Google Map API and invoke the Google Map GUI for displaying the camera location.

4.3.8 Event, alarm and push notification

Our event recorder is based on event-driven, both of motion detection and face detection provide the detecting method, if an event has been detected, the event recorder will insert the event detail into system database, at the same time it will invoke alarming and push notification service.

We use two methods to achieve alarm making. Because of .NET framework support, e-mailing is the main sending functionality. We assert class *System.Net.Mail* in our system, and create an object from the method *SendEmail*, then configure the parameters for SMTP. When an event happens, the system invokes the method *SMTP.send(·)* and sends an email to users. The email content includes snapshot, footage and detection information (i.e. motion detected, face detected). Moreover, we implement surveillance alarm making. We assert the class *System.Net.FtpWebRequest* in the CISS and create a method calling the *FtpService(·)*. Based on event-driven, our surveillance system will invoke this method, establish the connection with iSCSI drive, and upload the file to the storage server.

CISS maximizes the use of features in cloud computing, our system should have the push notification service. ASP.NET SignalR is a Toolkit for ASP.NET; it can achieve real-time web functionality to applications (Aguilar, 2014). We use the SignalR Hub which is one of the models for the communications between servers and clients, user client browser is able to receive the push message in real time (such as motion detected or face detected), the function is shown in Function 4.7.

Function 4.7 SignalR chat hub
Operation: Create the SignalR Hub on server side
Effect: All user can receive real-time push message from the client browser
Utilize ASP.NET SignalR, create a class <i>ChatHub</i> , it provides the real-time message push service for all client; the code saves in <i>App_Code\ChatHub.cs</i> <pre>public class ChatHub: Hub { public void Send(string name, string message) { // Call the broadcastMessage method to update clients. Clients.All.broadcastMessage(name, message); } }</pre>

Chapter 5 System Demonstration and Results

In this chapter, at first we introduce the plan of system testing and explain our experimental environment. Then we clearly demonstrate each component of the CISS. Finally, we draw conclusions and summarize findings through the research conducted. The results of the experiment will be demonstrated. In this thesis, in-depth discussions and evidence in relation to the significance of experimental results as well as their limitations will be stated.

5.1 System testing plan

In Chapter 5, we will introduce our system environment (hardware and software), then we will test the whole CISS system, we will check each of the modules mentioned in the previous chapter. The CISS testing is to ensure that the system can efficiently perform the user's requests. Moreover, the GUI snapshot of each module is provided to demonstrate availability of the whole CISS. For the experiments, we are going to test the system performance, and use Apache *JMeter* as a benchmark tool.

Apache *JMeter* is a powerful Java application for testing and measuring the performance of static and dynamic resources such as web server, database server, and FTP server (Halili, 2008). The test is used for CISS through setting up a thread group to simulate multi-user access. In the last part of chapter 5, the experimental results will be demonstrated and discussed.

5.2 Experimental environment

In this section we mainly introduce CISS testing environment which is divided into two parts i.e. hardware and software. The types of hardware are group into three classes, IP camera, network infrastructure and cloud server. The prototype CISS was developed in Microsoft Visual Studio 2015, the operating environment needs IIS, MySQL and. Net Framework 4.5. In the following sections, we will explain the details of parameters adopted by the CISS system.



Figure 5.1 RTSP: IP Cameras

In Figure 5.1, we see two RTSP based IP cameras, the left one is DCS-2136L and the right side is DCS-2330L.

D-Link DCS-2136L Cloud IP camera is able to provide HD videos (1280×720) up to 30fps, the wireless technology adopts the protocol 802.11ac. As the indoor camera, it supports night vision.

The maximum resolution of videos taken by camera D-Link DCS-2330L Cloud IP camera is 1280×800 and up to 30fps. The network technology uses 802.11n (300Mbps). As the outdoor camera, the DCS-2330L is applied for IP65 compliant weatherproof house.



Figure 5.2 Network facilities

Figure 5.2 shows the network facilities for CISS, the model of the router at the left hand is D-Link DSL-4320L, the right hand is D-Link DSL-2900AL.

DSL-4320L is a very high-performance router which totally has six external antennas. It supports most IEEE standards such as 802.11ac/n/g/b/a. It mainly works with the private cloud (QNAP TVS-682).

The DSL-2900AL also provides high-performance network environment which took use of dual channel, one 2.4GHz plus one 5GHz.



Figure 5.3 Private cloud devices

The private cloud devices QNAP TVS 682 (left) and TS-253(right) Pro are shown in Figure 5.3. The CISS system is running on TVS 682 and videos are stored in TS-253 Pro. We upgraded the CPU and RAM for TVS 682. The hardware configuration is shown in Table 5.1 and Table 5.2.

Table 5.1 QNAP TVS 682 specifications

QNAP TVS 682	
CPU	Intel Core i7 6700T @ 2.8GHz 4 Cores 8 Threads.
RAM	4x4GB DDR4 @ 2133MHz
Storage	4x256GB RAID 0 SSD and 4x4 8TB RAID 5 HDD
Network	4x1Gigabyte Ethernet port

Table 5.2 QNAP TS 253 PRO specifications

QNAP TS 253 Pro	
CPU	Intel Celeron J1900 @ 2.0GHz 4 Cores 4 Threads.
RAM	2x4GB DDR3L @ 2133MHz
Storage	2x4TB RAID 0 HDD (iSCSI LUN)
Network	2x1Gigabyte Ethernet port

In the previous chapter, we discussed that a network is divided into internal network and a public network, which is mainly used for explaining components of the DSL-2900AL, DSL-4320L links to the private cloud, provides Internet services, and the router has been mapped to the system ports. It is possible to access CISS via Internet.

5.3 CISS demonstration

In Section 5.2, we introduced the runtime environment of our system. In this section, the full demonstration of CISS will be shown. Also, each module will be tested to determine whether it is properly running. The CISS system includes user authentication, camera overview, camera monitor, camera sensor, event module and alarm making module.

When a user opens a browser, and enters CISS, the user requires to sign into the system shown as Figure 5.4. The user authentication database is allocated in Microsoft Azure Australian public cloud. When a user no longer needs the operating system, if the user clicks on the upper right corner, a sign will show that the user has logged out.

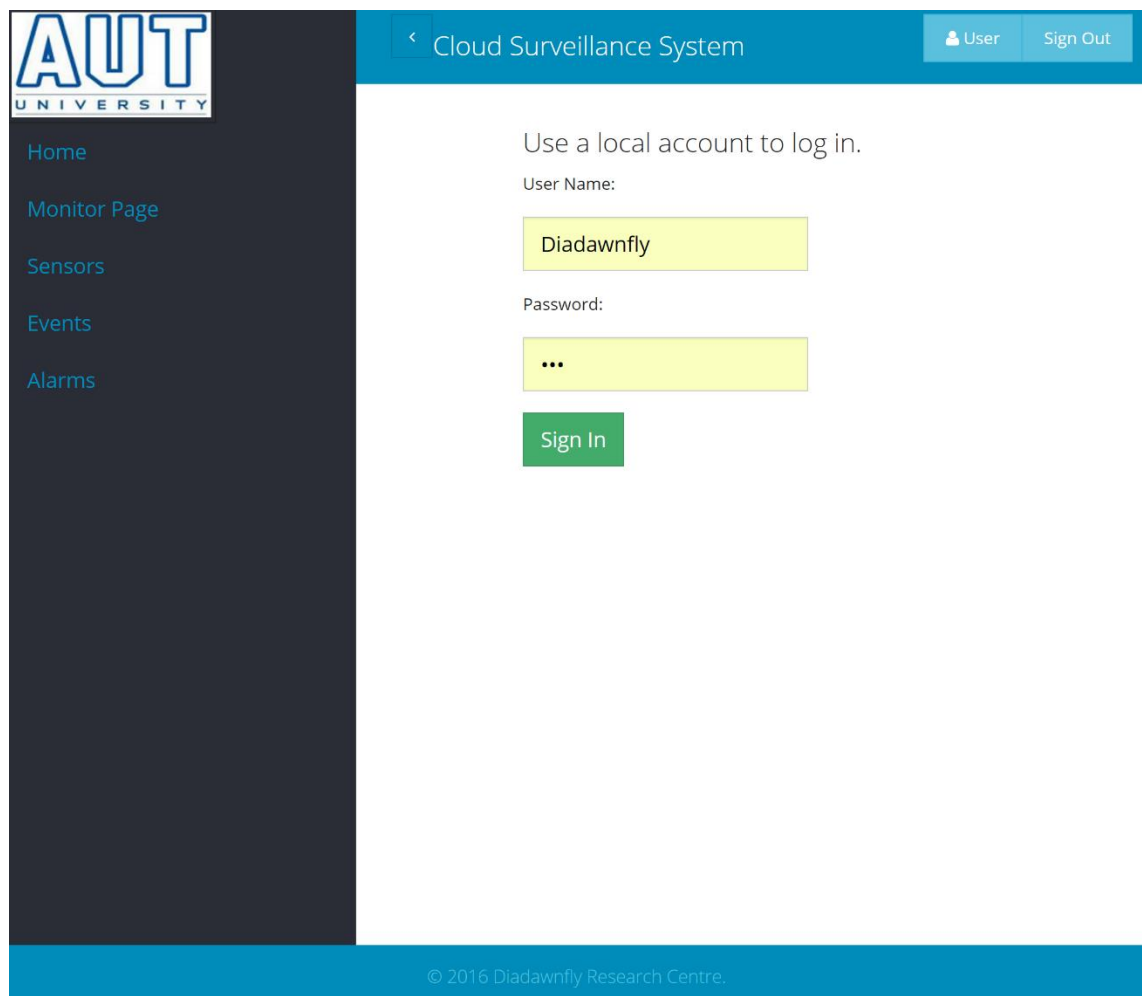


Figure 5.4 CISS user authentication

If a user passes the authentication, the browser will automatically jump to the camera overview page (Figure 5.5). The user can get all push messages in this SignalR hub, no matter which camera detects the event, the detection information will be pushed to the

SignalR hub in real time. If a user watches the large monitor, (s)he could click the top-left button to hide the navigation bar (Figure 5.6).

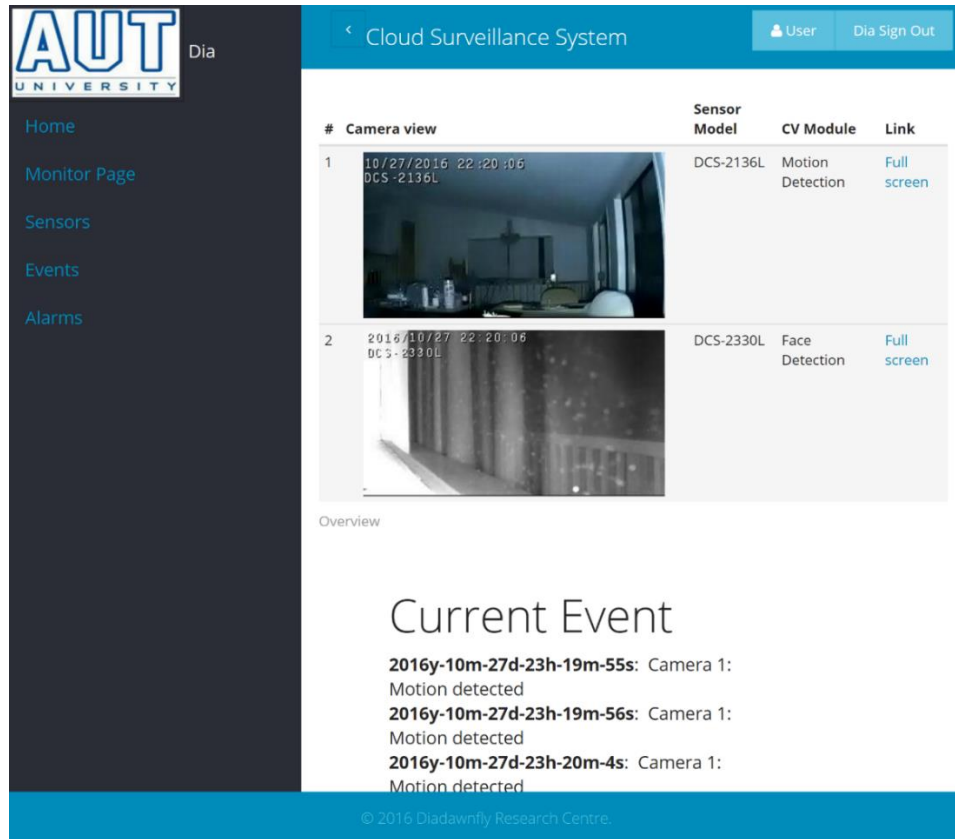


Figure 5.5 CISS camera overview

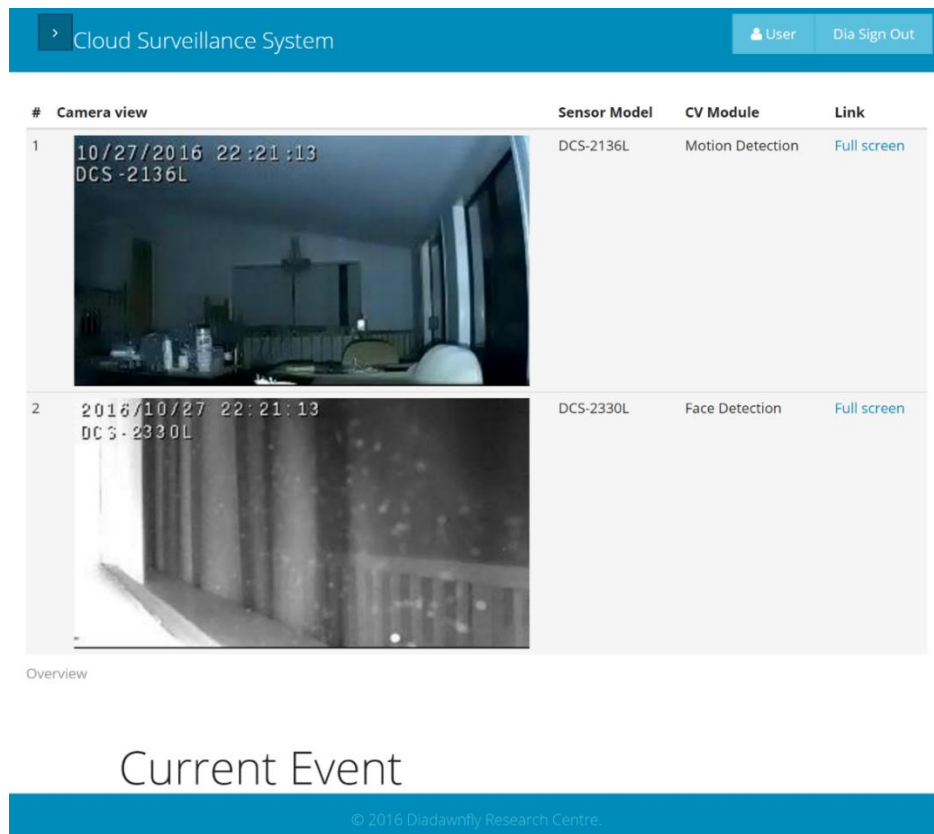


Figure 5.6 CISS full page camera overview

When a user clicks on the full-screen button on the right side of the table, the user will enter the specified state of camera monitoring, (s)he can see the real-time video monitoring screen after image processing. Figure 5.7 shows Camera 1 with motion detection module and Figure 5.8 shows Camera 2 with face detection module.

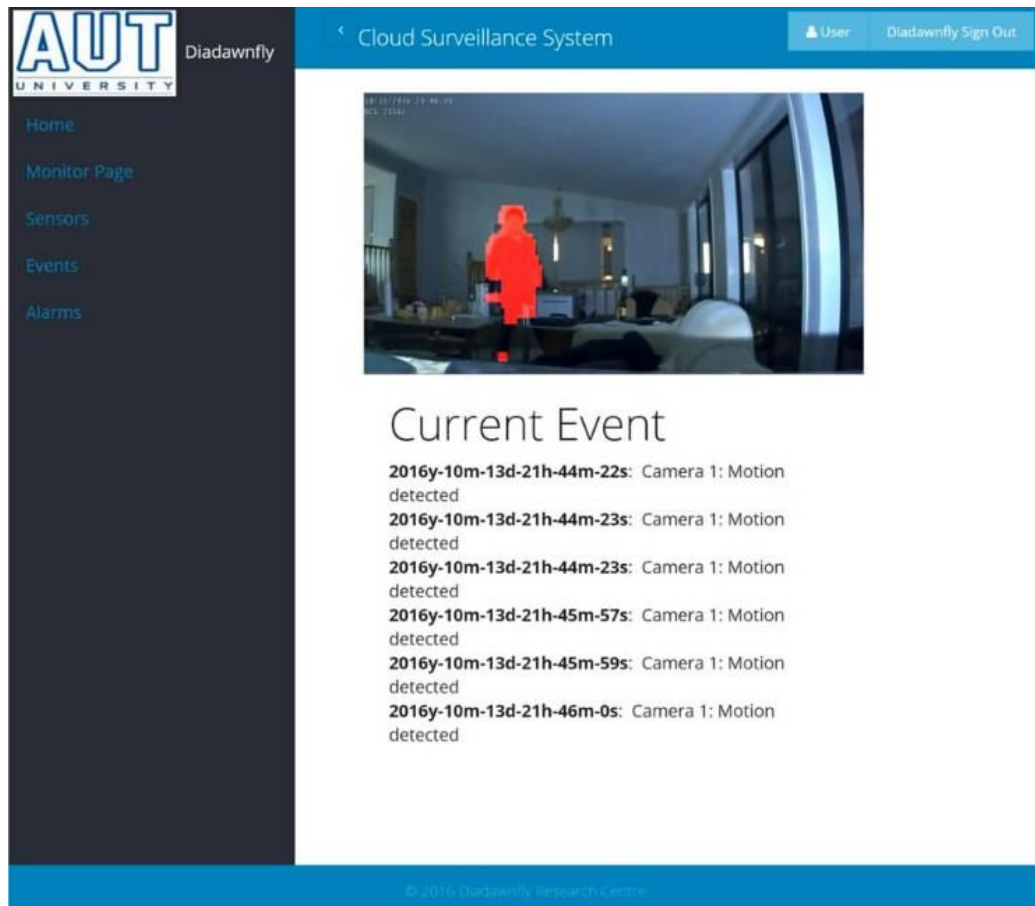


Figure 5.7 CISS camera monitor with motion detection module



Figure 5.8 CISS camera monitor with face detection module

When a user needs to view the camera location, the user can click the sensor button in the navigation bar to enter the camera sensor module. In Figure 5.9, if the user clicks the camera ID in front of the select button, the Google Map shows the user's action on the camera location.

Camera ID	Media stream	Location	Longitude	Latitude	Originated from	Media device properties	Delete
Select 1	Video	Auckland - Blockhouse Bay	174.7010	-36.9235	115.188.127.160	DCS-2330L	<input type="checkbox"/>
Select 2	Video	Auckland - Blockhouse Bay	174.7027	-36.9244	115.188.127.160	DCS-2136L	<input type="checkbox"/>

Delete selected

Figure 5.9 CISS camera sensor module

The user can access the alarm making module by clicking the alarm button on the left to view all the alarm information shown in Figure 5.10. The user can select the alarm records and delete the selected alarm entries in batch.

Alarm ID	Related Event ID	Alarm description	Originated from	Date Time	Delete
1155	#f639736-17e5-11e6-ad8b-c336c023	Motion detected	115.188.127.160	2016/5/12 14:23:19	<input type="checkbox"/>
1156	#f659e24-17e5-11e6-ad8b-c336c023	Motion detected	115.188.127.160	2016/5/12 14:23:19	<input type="checkbox"/>

Delete selected

Figure 5.10 CISS alarm module

When a user clicks the corresponding button for alarm description in the alarm making module. Then the user can view the corresponding events in the CISS event module. In Figure 5.11, the video footage related to the event will automatically be played by the flash video player. The system may have more view for the viewer. In Figure 5.12, the comments of a viewer on the event can be added.

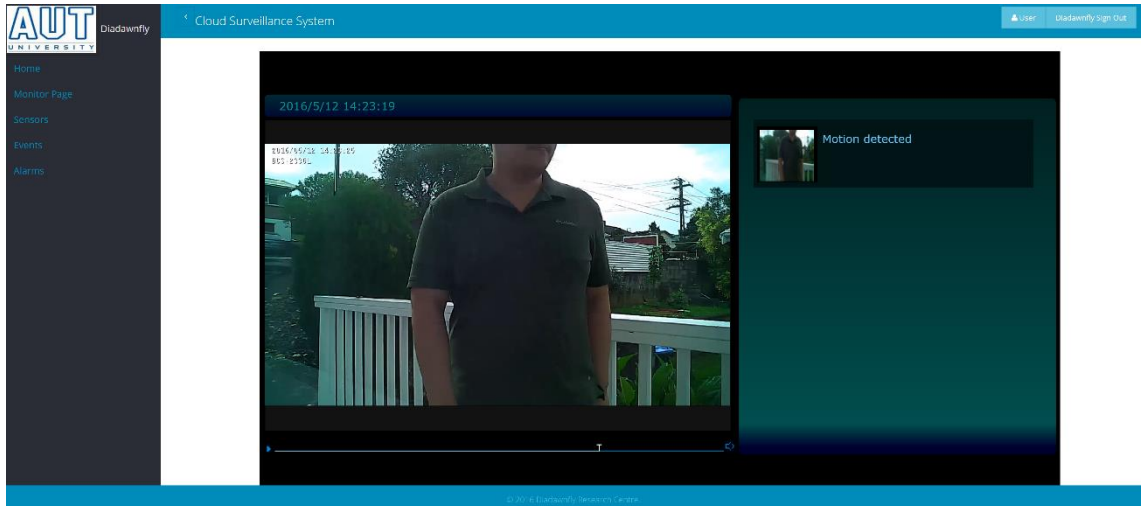


Figure 5.11 CISS event module

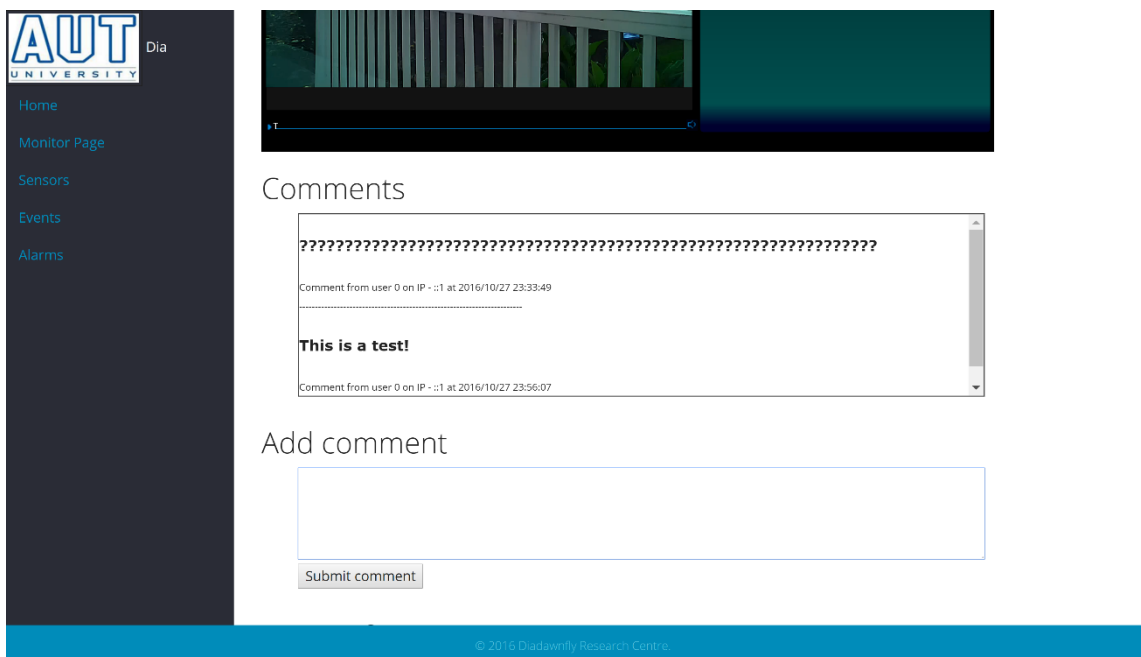


Figure 5.12 CISS event module comments part

5.4 Experimental results

In our experiment, each thread needs to access the system at the same time with six modules in the CISS system. Each test has a 10 seconds ramp-up period and needs to

cycle through five times (Loop count). The above parameters are for the experiments. Then we explain the experimental results related to the parameters. We use the first parameter as an example. Therefore, this parameter represents the total thread (user) access to X elements in X times. In our eight experiments, each thread was set to 5, 10, 20, 40, 80, 200, 500, and 1000. For an example, when the thread equals to 5, the number of the simple is 150 because of 5 threads with 5 loop counts accompanying with 6 elements. The *Min* stands for the minimum response time of the system in this experiment. The *Max* is the maximum response time. The average represents the average response time of the system in the experiment. *Total Error* means the total number of times the system has not responded or the network has lost packets when the error is occurred in the total connection. *Throughput* expresses the total time required, the unit KB/s is the throughput speed. The formula of Throughput and KB/s are,

$$Throughput = \frac{\text{Number of requests}}{\text{Total time to issues the requests}}$$

$$KB/s = \frac{\text{Average bytes}}{1024} \times Throughput$$

Our first test was conducted on the development machine. The development machine is called Dia Razer Profession Computer, CPU Intel i7-5930K 3.5GHz 6 Core 12 Thread, 32GB DDR4 2400MHz RAM, ASUS PCE-AC68 network card, NVMe SSD 512GB, NVIDIA GTX TITAN X 2xSLI (12GB GDDR5) graphic cards. As a top-of-the-line computer, the results are perfect. In Table 5.3, we see that the average access time is 1 second. The highest throughput rate is up to 19231KB/s = 153848Kbps.

Table 5.3 CISS benchmark in local PC

Experimental number	1	2	3	4	5	6	7	8	
Parameters	Thread (user) number	5	10	20	40	80	200	500	1000
	Ramp-UP Period (s)	10	10	10	10	10	10	10	10
	Loop Count	5	5	5	5	5	5	5	5
	Element	6	6	6	6	6	6	6	6
Results (Total)	Sample	150	300	600	1200	2400	6000	15000	30000
	Minimum(ms)	1	1	1	0	0	0	0	0
	Maximum(ms)	10	4	6	20	17	26	33	194
	Average(ms)	1	1	1	1	1	1	1	1
	Total Error	0	0	0	0	0	0	0	0
	Error Rate	0%	0%	0%	0%	0%	0%	0%	0%
	Throughput(s)	18.6	33.1	62.8	122.3	241.7	597.1	1476.8	2878.8
	KB/s	124.5	221.2	419.3	817.1	1614.4	3986.6	9865.4	19231

In Table 5.4, the testing client also is Dia Razer Professional PC, but the server we used private Cloud QNAP TVS 682. Due to the excellent configuration of the server, and advanced network equipment, the results of the experiment are also perfect.

Table 5.4 CISS benchmark in private cloud

Experimental number		1	2	3	4	5	6	7	8
Parameters	Thread (user) number	5	10	20	40	80	200	500	1000
	Ramp-UP Period (s)	10	10	10	10	10	10	10	10
	Loop Count	5	5	5	5	5	5	5	5
	Element	6	6	6	6	6	6	6	6
Results (Total)	Sample	150	300	600	1200	2400	6000	15000	30000
	Minimum(ms)	5	5	5	4	4	4	4	5
	Maximum(ms)	14	16	22	28	20	33	55	251
	Average(ms)	6	6	6	6	6	7	7	15
	Total Error	0	0	0	0	0	0	0	0
	Error Rate	0%	0%	0%	0%	0%	0%	0%	0%
	Throughput(s)	18.3	32.6	61.8	120.7	238	586.7	1456	2827.7
KB/s	122.2	217.6	413.2	806.5	1590.5	3920.7	9727.4	18854	

5.5 Discussions and analysis

The experimental results were demonstrated previously in this chapter. We have two groups of experiments. In the first experiment, we ran the system on local PC, and the testing PC is on the same hardware. In the second group of experiments, the system was running on our private cloud, the testing PC is our local computer. Because of limitations, there is no more system running environment for testing.

We compare the two groups of experimental results shown in Table 5.3 and Table 5.4, we found that the average response time of the system should be longer than that of the local tests, especially the results of the eight experiments, the local results are much better than the results of private cloud tests. At present, both of local and global testing results are quite perfect. We see that the maximum throughput rate is close to 150Mbps (1024 KB/s = 8Mbps), that means the wireless network protocol requirement of CISS should be 802.11n (>300Mbps) or 802.11ac, the 0 error rate shows the stability of the system. Through the experimental results we see that high-performance hardware devices for the system provide a more stable operating environment.

On the whole, we think the network bandwidth limit is challenging for CISS, insufficient network bandwidth results in high network delay for the system and

eventually lead to system errors. We have considered the network bandwidth of the CISS system. It is hard to use a program to control the system network bandwidth, so we decided to utilize the server configuration system policy to control maximum connections, and also to control CISS access of low quality video streaming from IP cameras. Overall, the network bandwidth requirement of CISS is relatively high.

In future, we will take the running system on the hybrid cloud into consideration, we will get more experimental results, after comparing them, we can better explain the operating environment of our system based on the performance.

Chapter 6

Conclusion

In this thesis, an in-depth articulation of the communication frameworks is detailed. We successfully designed, implemented and demonstrated a cloud based intelligent surveillance system. We clearly demonstrated the innovation in research outcomes. In the final stage, we conducted experiments to test the performance of our developed system when the system is running in different environments. In this chapter, we will summarize the entire research project and look ahead the future work.

6.1 Conclusion

This thesis presents an innovative cloud based intelligent surveillance system following the principles of software engineering. Based on the developed system, our users can connect to the terminals so as to access the surveillance monitoring and get the visual information captured. No matter where the users are, as long as they are connected to the Internet, they will not be limited by the capacity of the cloud based surveillance system. Once the system has been successfully implemented, the original research problems were resolved. Below is a summary of the findings after the procedures of the entire system were completed.

The cloud based intelligent surveillance system (CISS) is able to integrate with the Computer Vision (CV) module. In case there is need for facial recognition, then a face detection module for specified cameras will be added. When recording license plates, the car plate recognition module was activated, and the car plate recognition parameters were configured to record license plates in real time. Unlike most surveillance systems are only with motion detection, our innovation CISS has the common motion detection modules, and a CV module for each camera so that a single system can control multiple cameras, the CV module of each camera can be customized.

The cloud based intelligent surveillance system (CISS) can be monitored and remotely controlled. First of all, the system has a push notification feature and therefore a user can access the cloud based system from their phones, computer, and other network-enabled devices. However, the traditional client software is not Internet based. Moreover, the storage system uses iSCSI LUN + ownCloud solution to allow users to view the full logs of the video monitoring using network-enabled devices. Meanwhile, this solution also guarantees data security on the storage server since users can not directly access the system data. On the other hand, the solution can reduce the traffic jam related to the surveillance server. Users can obtain monitored visual data through web pages provided by *ownCloud* or the client login (including PC and mobile phones). If PC client software provided by *ownCloud* is used, visual surveillance data can be backed up remotely.

6.2 Future work

In future, system integration and stability will be enhanced. The SDK used by the system is not officially licensed, so ads from SDK vendors keep running. We hope to further integrate the new SDK in our work such as Emgu CV. At the same time, due to different conditions and time constraint, cloud client was not well developed. Finally, we used *ownCloud* as a solution for cloud storage.

We hope to improve the design, develop a cloud storage module and achieve perfect integration with CISS in our future study. In this thesis, we were only able to carry out tests on the private cloud and free public cloud platforms. However, the performance of the free public cloud platforms was very low with the testing results of the public cloud platforms being inferior to those of the private cloud platform.

Moreover, we hope to complete experiments on the hybrid cloud platform in the near future. The hybrid cloud platform as a test running environment is more suitable for our system. Because enterprises run their systems on the hybrid cloud platform, it can provide better security and connectivity for such a system.

Last but not the least, there is a certain security risk of running the CISS system on the public cloud platform because the provider of the platform is a tripartite enterprise rather than the enterprise itself. The problem with the private cloud platform is the feature closure. When the system needs to access multiple streaming videos at the same time the network is unable to provide sufficient bandwidth. We believe that running CISS on a hybrid cloud platform will make the system more efficient and the platform features will guarantee its security under good connectivity.

References

- Agarwal, D., & Prasad, S. K. (2012, May). Azure bench: Benchmarking the storage services of the Azure Cloud Platform. *In Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)*, pp. 1048-1057.
- Aguilar, J. M. (2014). SignalR programming in Microsoft ASP. NET. *Microsoft Press*.
- Alamri, A., Hossain, M. S., Almogren, A., Hassan, M. M., Alnafjan, K., Zakariah, M., & Alghamdi, A. (2015). QoS-adaptive service configuration framework for cloud-assisted video surveillance systems. *Multimedia Tools and Applications*, pp. 1-16.
- Aleem. A. & Sprott. C.R. (2013). Let me in the cloud: analysis of the benefit and risk assessment of cloud platform. *Journal of Financial Crime*, pp. 6-24.
- Anghelescu, P., Serbanescu, I., & Ionita, S. (2013, June). Surveillance system using IP camera and face-detection algorithm. *In Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1-6.
- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., & Zaharia, M. (2009). Above the clouds: A berkeley view of cloud computing.
- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, pp. 50-58.
- BaoHong.Y & Yan. W (2015). Investigation on application of video monitoring. *China building materials science and technology 2015*, Vol.1. pp.153-154.
- Begum, S., & Khan, M. K. (2011, July). Potential of cloud computing architecture. *In Information and Communication Technologies (ICICT)*, pp. 1-5.
- Behl, A., & Behl, K. (2012, October). An analysis of cloud computing security issues. *In Information and Communication Technologies (WICT)*, pp. 109-114.
- Beyer. J., Elhrouz. H.& El Seed. K. (2012). Streamlining test and evaluation with cloud computing. *Digital Avionics Systems Conference*, pp.9E3-1-9E3-6
- Boehm, B., & Hansen, W. J. (2000). SPECIAL REPORT CMU/SEI-2000-SR-008.
- Bogardi-Meszoly, A., Levendovszky, T., & Charaf, H. (2006). Performance Factors in ASP. NET Web Applications with Limited Queue Models. *In 2006 International Conference on Intelligent Engineering Systems*, pp. 253-257.
- Chang, A. Y., Parrales, M. E., Jimenez, J., Sobieszczyk, M. E., Hammer, S. M., Copenhaver, D. J., & Kulkarni, R. P. (2009). Combining Google Earth and GIS mapping technologies in a dengue surveillance system for developing countries. *International Journal of health geographics*, 8(1), p.1.

- Chen, L., Shashidhar, N., & Liu, Q. (2012, March). Scalable secure MJPG video streaming. *In Advanced Information Networking and Applications Workshops (WAINA)*, pp. 111-115.
- Chen, T. S., Lin, M. F., Chieuh, T. C., Chang, C. H., & Tai, W. H. (2015, September). An intelligent surveillance video analysis service in cloud environment. *In Security Technology (ICCST)*, pp. 1-6.
- Chen, W. T., Chen, P. Y., Lee, W. S., & Huang, C. F. (2008, May). Design and implementation of a real time video surveillance system with wireless sensor networks. *In Vehicular Technology Conference, 2008*, pp. 218-222.
- Chen, X., Xu, J. B., & Guo, W. Q. (2013, July). The research about video surveillance platform based on cloud computing. *In 2013 International Conference on Machine Learning and Cybernetics*, Vol. 2, pp. 979-983.
- Chen, Y. L., Chen, T. S., Yin, L. C., Huang, T. W., Wang, S. Y., & Chieuh, T. C. (2014, September). City Eyes: An Unified Computational Framework for Intelligent Video Surveillance in Cloud Environment. *In Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications (GreenCom), IEEE and Cyber, Physical and Social Computing (CPSCom)*, (pp. 324-327).
- Chia-Feng L., Shyan-Ming Y., Muh-Chyi L. & Ching-Tsorng T. (2012). A Framework for Scalable Cloud Video Recorder System in Surveillance Environment. *International Conference on Ubiquitous Intelligence and Computing*, pp.655-660.
- Christian, V., Xingchen, C. & Rajkumar, B. (2009). Aneka: A Software Platform for .NET-based Cloud Computing. *Manjrasoft Pty Ltd: Melbourne*.
- Collins, R. T., Lipton, A. J., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., ... & Wixson, L. (2000). A system for video surveillance and monitoring. *Technical Report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University*, pp. 1-6.
- Davenport, T. H., Barth, P. & Bean, R. (2012). How Big Data Is Different. *MIT Sloan Management Review*, 54(1), pp. 43-46.
- Devasena, C. L. (2014). Impact study of cloud computing on business development. *Operations Research and Applications: An International Journal (ORAJ)*, 1(1), 1-7.
- Dunkel, D (2011). Surveillance 'In the Cloud' is Only the Beginning. *SDM*, 41(3):68.
- Dunkel D. (2012). The "Wonderful World" of Cloud Surveillance. *SDM*, 42(6): 50.
- Engebretson, David J. (2015). Surveillance Video Storage. *Security Distributing & Marketing*.

- Franks, B. (2014). Taming The Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics.
- Frank H. (2011). Cloud Computing for syndromic surveillance. *Emerging Health Threats Journal*, 4(0):71-71.
- Gandomi, A. & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2): 137–144.
- Goyal, S. (2014). Public vs private vs hybrid vs community-cloud computing: A critical review. *International Journal of Computer Network and Information Security*, 6(3), p.20.
- Halili, E. H. (2008). Apache JMeter: A practical beginner's guide to automated testing and performance measurement for your websites. *Packt Publishing Ltd*.
- Hassan, M. M., Hossain, M. A., Abdullah-Al-Wadud, M., Al-Mudaihesh, T., Alyahya, S., & Alghamdi, A. (2015). A scalable and elastic cloud-assisted publish/subscribe model for IPTV video surveillance system. *Cluster Computing*, 18(4):1539-1548.
- Hossain, M. A. (2014). Framework for a cloud-based multimedia surveillance system. *International Journal of Distributed Sensor Networks*.
- Hossain, M. A. (2013, November). Analyzing the suitability of cloud-based multimedia surveillance systems. *In High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC)*, pp. 644-650.
- Hossain, M. A., & Song, B. (2016). Efficient Resource Management for Cloud-enabled Video Surveillance over Next Generation Network. *Mobile Networks and Applications*, pp.1-16.
- Hossain, M. S., Hassan, M. M., Al Qurishi, M., & Alghamdi, A. (2012, July). Resource allocation for service composition in cloud-based video surveillance platform. *In Multimedia and Expo Workshops (ICMEW)*, pp. 408-412.
- Hu, H., Wen, Y., Chua, T.-S. & Li, X. (2014). Toward Scalable Systems for Big Data Analytics: A Technology Tutorial. *IEEE*.
- Irene, D. S., & Dhanalakshmi, R. (2013, February). Video surveillance system and Content Sharing between PC and mobile using android. *In Information Communication and Embedded Systems (ICICES)*, pp. 485-490.
- Jiang, J., Sekar, V., & Zhang, H. (2012, December). Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. *In Proceedings of the*

- 8th international conference on Emerging networking experiments and technologies ACM*, pp. 97-108.
- J. Dale P. (2014). Introduction to Cloud Computing. *Journal of Electronic Resources*.
- Kaisler, S., Armour, F., Espinosa, J. A. & Money, W. (2013). Big Data: Issues and Challenges Moving Forward. *IEEE*.
- Kambatla, K., Kollias, G., Kuma, V. & Grama, A. (2014). Trends in big data analytics. *Journal of Parallel and Distributed Computing*, 74(7): 2561-2573.
- Karimaa, A. (2011). Video surveillance in the cloud: dependability analysis. *IARIA*. ISBN: 978-1-61208-149-6.
- Kim, S., Nam, Y., Kim, J., & Cho, W. D. (2009, December). ISS: intelligent surveillance system using autonomous multiple cameras. In *Ubiquitous Information Technologies & Applications, 2009. ICUT'09*, pp. 1-6.
- Lage, R., Dolog, P., & Leginus, M. (2014, July). The role of adaptive elements in web-based surveillance system user interfaces. In *International Conference on User Modeling, Adaptation, and Personalization*, pp. 350-362.
- Lamy-Bergot, C., Renan, E., Gadat, B. & Lavaux, D. (2009). Data supervision for adaptively transcoded Data supervision for adaptively transcoded. *Proceedings of the IEEE ITST'09*, pp. 415-419.
- Li. Q & Zhang. T & Yu. Y (2011). Using cloud computing to process intensive floating car data for urban traffic surveillance. *International Journal of Geographical Information Science*, 25(8): 1303-1322.
- Limna, T., & Tandayya, P. (2012, October). Design for a flexible video surveillance as a service. In *Image and Signal Processing (CISP)*, pp. 197-201.
- Limna, T. & Tandayya, P. (2016). A flexible and scalable component-based system architecture for video surveillance as a service, running on infrastructure as a service. *Multimedia Tools and Applications*, 75(4): 1765–1791.
- Liu, C. (2000). multimedia over ip: Rsvp, rtp, rtcp, rtsp. *Handbook of Communication Technologies: The Florida*.
- Lo. Wt, Chang. Ys & Sheu. Rk (2014). Implementation and Evaluation of Large-Scale Video Surveillance System Based on P2P Architecture and Cloud Computing. *International Journal of Distributed Sensor Networks*.
- Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference*, 2:1150-1157.

- Luo, J. Z., Jin, J. H., Song, A. B., & Dong, F. (2011). Cloud computing: architecture and key technologies. *Journal of China Institute of Communications*, 32(7): 3-21.
- Marko K. (2012). Cloud Storage Evolved. *InformationWeek*. 1331:30-33
- Marier & Keven (2012). Surveillance Managed Services Delivered via the Private Cloud. *Security*, 49(12): 48
- Martini, B., & Choo, K. K. R. (2013). Cloud storage forensics: ownCloud as a case study. *Digital Investigation*, 10(4): 287-299.
- Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.
- Nafisi. A. (2015). Surveillance States. *New York Times Book Review*.
- Neal. D. & Rahman. S. M. (2012). Video surveillance in the cloud-computing? *International Conference on Electrical and Computer Engineering*, pp.58-61.
- Norris, C., Moran, J., & Armstrong, G. (Eds.). (1998). Surveillance, closed circuit television and social control. pp. 255-257.
- Paul, A. K., & Park, J. S. (2013, May). Multiclass object recognition using smart phone and cloud computing for augmented reality and video surveillance applications. *In Informatics, electronics & vision (ICIEV)*, pp. 1-6.
- Peng-Jung W. & Yung-Cheng K. (2014). Computing resource minimization with content-aware workload estimation in cloud-based surveillance systems. *IEEE Conference Publications*.
- Prati, A., Vezzani, R., Fornaciari, M., & Cucchiara, R. (2013). Intelligent video surveillance as a service. *In Intelligent Multimedia Surveillance*, pp. 1-16.
- Qi, W. A. N. G., & Yu, Z. H. U. (2006). Design and Implementation of Digital Video Surveillance System Based on B/S Structure. *Computer Engineering*, 19, 089.
- Qian, L., Luo, Z., Du, Y., & Guo, L. (2009, December). Cloud computing: an overview. *In IEEE International Conference on Cloud Computing*, pp. 626-631.
- Renkis, Martin (2013). Bandwidth. Storage. Speed for cloud surveillance. *Security Systems News*. Vol.16(5). p.16.
- Rodríguez-Silva, D. A., Adkinson-Orellana, L., Gonz'lez-Castano, F. J., Armino-Franco, I., & Gonz'lez-Martinez, D. (2012, June). Video surveillance based on cloud storage. *In Cloud Computing (CLOUD)*, pp. 991-992.
- Russell S. (2011). Video's Cloudy Future. *Solutions for Enterprise Security Leaders*, Vol.48 (10). pp.34-36
- Saeed, G. Juell-Skielse & E. Uppström (2012). Cloud Resource Planning Adoption: Motives & Barriers. *Advances in Information Systems II. Aalborg*.

- Chen, L., Shashidhar, N., & Liu, Q. (2012, March). Real time streaming protocol (RTSP). Scully Patrick. 2012. Cloud storage. *Broadcast Engineering*, 54(11): 30-33.
- Sharma, C. M., & Kumar, H. (2014, March). Architectural framework for implementing visual surveillance as a service. *In Computing for Sustainable Global Development (INDIACom)*, pp. 296-301.
- Shawish, A., & Salama, M. (2014). Cloud computing: paradigms and technologies. *In Inter-cooperative Collective Intelligence: Techniques and Applications*, pp. 39-67.
- Shibao Z. (2009). Intelligent video surveillance technology and application. *Television Technology*, (1): 94-96.
- Shiwen Z., Yaping L. & Qin L. (2014). Secure and Efficient Video Surveillance in Cloud Computing. *International Conference on Mobile Ad Hoc and Sensor Systems*, pp.222-226.
- Song, B., Hassan, M. M., Tian, Y., Hossain, M. S., & Alamri, A. (2015). Remote display solution for video surveillance in multimedia cloud. *Multimedia Tools and Applications*, pp.1-22.
- Song, B., Tian, Y., & Zhou, B. (2014, August). Design and evaluation of remote video surveillance system on private cloud. *In Biometrics and Security Technologies (ISBAST)*, pp. 256-262.
- Soulsby D. (2012). Using Cloud Storage for NMR Data Distribution. *Journal of Chemical Education*, 89(8): 1007-1011.
- Spillner, J, Muller, J & Schill, A (2013). Creating optimal cloud storage systems. *Future Generation Computer Systems-The International Journal Of Grid Comput*, 29(4): 1062-1072.
- S. Salleh M., S. Teoh Y. & C. C. (2012). Cloud Systems: A Review of Literature and its Adoption. *The PACIS 2012 Proceedings*.
- Sunehra, D., & Bano, A. (2014, December). An intelligent surveillance with cloud storage for home security. *In 2014 Annual IEEE India Conference (INDICON)*, pp. 1-6.
- Tekeoglu, A., & Tosun, A. S. (2015, August). Investigating Security and Privacy of a Cloud-Based Wireless IP Camera: NetCam. *In 2015 24th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1-6.
- Rajaraman, V. (2014). Cloud computing. *Resonance*, 19(3), 242-258.
- Valera, M., Velastin, S. (2005). Intelligent distributed surveillance systems: a review. *Image Signal Process*, 152(2): 192–204.

- Varghese, B., Akgun, O., Miguel, I., Thai, L., & Barker, A. (2014, December). Cloud benchmarking for performance. *In Cloud Computing Technology and Science (CloudCom)*, pp. 535-540.
- Vecchiola, C., Chu, X., & Buyya, R. (2009). Aneka: a software platform for .NET-based cloud computing. *High Speed and Large Scale Scientific Computing*, (18): 267-295.
- Vecchiola, C., Pandey, S., & Buyya, R. (2009, December). High-performance cloud computing: A view of scientific applications. *In 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks*, pp. 4-16.
- Wang, Z., Liu, S., & Fan, Q. (2013, June). Cloud-based platform for embedded wireless video surveillance system. *In Computational and Information Sciences (ICCIS)*, pp. 1335-1338.
- Wenzhe J., Guoqing W., Zhengjun Z. & Xiaoxue Y. (2013). Dynamic Data Possession Checking for Secure Cloud Storage Service. *Journal of Networks*. pp 2713-2720.
- Wo, T., Hu, C., Li, J., & Huai, J. (2012, October). Netros: A virtual computing environment towards instant service of network software. *In Semantics, Knowledge and Grids (SKG)*, pp. 24-31.
- Woo, S. W., Joh, H., Alhazmi, O. H., & Malaiya, Y. K. (2011). Modeling vulnerability discovery process in Apache and IIS HTTP servers. *Computers & Security*, 30(1):50-62.
- Xiong, Y., Wan, S., She, J., Wu, M., He, Y., & Jiang, K. (2016). An energy-optimization-based method of task scheduling for a cloud video surveillance center. *Journal of Network and Computer Applications*, 59: 63-73.
- Xiong, Y., Wan, S. Y., He, Y., & Su, D. (2014). Design and Implementation of a Prototype Cloud Video Surveillance System. *JACIII*, 18(1): 40-47.
- Xu, Z., Mei, L., Liu, Y., Hu, C., & Chen, L. (2016). Semantic enhanced cloud environment for surveillance data management using video structural description. *Computing*, 98(1-2): 35-54.
- Yadav, D. K., & Singh, K. (2016). A combined approach of Kullback–Leibler divergence and background subtraction for moving object detection in thermal video. *Infrared Physics & Technology*, 76: 21-31.
- Yan, W. Q. (2016). Introduction to Intelligent Surveillance. *Springer*, pp. 139-141.
- Yang, S. J., Deng, X. Y., & Wang, M. Y. (2012, July). Construction of modern education technology web-based course platform based on .NET. *In Computer Science & Education (ICCSE)*, pp. 1702-1705.

- Yong, P. E. N. G., Wei, Z. H. A. O., Feng, X. I. E., DAI, Z. H., Yang, G. A. O., & CHEN, D. Q. (2012). Secure cloud storage based on cryptographic techniques. *The Journal of China Universities of Posts and Telecommunications*, 19: 182-189.
- Yi, S., Jing, X., Zhu, J., Zhu, J., & Cheng, H. (2012). The model of face recognition in video surveillance based on cloud computing. *In Advances in computer science and information engineering*, pp. 105-111.
- Yu-Sheng W., Yue-Shan C., Tong-Ying J. & Jing-Shyang Y. (2012). An Architecture for Video Surveillance Service Based on P2P and Cloud Computing. *International Conference on Ubiquitous Intelligence and Computing*, pp.661-666
- Zhang, C., & Chang, E. C. (2014, June). Processing of mixed-sensitivity video surveillance streams on hybrid clouds. *In 2014 IEEE 7th International Conference on Cloud Computing*, pp. 9-16.
- Zhang, H., Li, M., Chen, Z., Bao, Z., Huang, Q., & Cai, D. (2010, June). Land use information release system based on Google Maps API and XML. *In 2010 18th International Conference on Geoinformatics*, pp. 1-4.
- Zhao, Z. F., Cui, X. J., & Zhang, H. Q. (2012). Cloud storage technology in video surveillance. *In Advanced Materials Research*, 532: 1334-1338.