International Journal of Control and Automation
Vol.10, No.3 (2017), pp.181-192
http://dx.doi.org/10.14257/ijca.2017.10.3.15

A Service Platform for Real-Time Video Streaming Based on RESTful API in IoT Environments

Songai Xuan[1], Dong-Hwan Park[2] and DoHyeun Kim [1]

[1]Computer Engineering Department, Jeju National University, South Korea
[2]ETRI, 218 Gajeong-ro Yuseung-Gu, Daejeon, Republic of Korea
[1]xuansongai@foxmail.com, [2]dhpark@etri.re.kr, and [1]kimdh@jejunu.ac.kr

Abstract

Recently, real-time image service based on IP camera is more and more useful. It can be using in the security system, babysitting system or anywhere people want to monitoring legally. In this paper, we proposed a real-time image service platform based on RESTful API using IP camera in IoT environments. User can see the real-time image in the client part and choose to save videos in the local file system. And people also able to manage the saved videos in the client part. The communication between each part based on RESTful API, and the client part will connect to IP Camera by an IP.

***Keywords**: RESTful API, real-time image service, IP Camera*

1. Introduction

IP camera is a type of digital video camera commonly employed for surveillance, and can send and receive data via a computer network and the Internet. Video streaming is becoming rapidly popular among the internet users with the growth and expansion of broadband internet. A video streaming service has plenty of applications such as surveillance, coverage of entertainment events, video conferencing [1]. A video streaming service platforms offer functionalities and features such as remote camera pan/tilt control, recording of media, integration to charging mechanisms, web based access and securing streamed content. Also, A video streaming service platforms can be used for applications such as traffic control, tourist route information, video surveillance and disaster evacuation [2].

Recently, much research has been carried out on the subject of designing service platforms for multimedia streaming services The UMBS provides the related mechanisms for system manual setting, automatic configuration, and management to improve the whole procedures of setting and installation [3]. A video streaming service can be used with distributed peers on the unstable overlay networks, and it is available to provide streaming session which create network route as the best effort in that time [4]. Until now, a single service platform can adapt its functionality based on the application requirements given by each customer [2].

In this paper, we propose a real-time image service platform based on RESTful API using IP camera in IoT environments. This is a design of a generic live streaming platform which is capable of delivering subscription based services. There is a set of channels or live audio/video feeds captured from a number of IP cameras provided by the platform and the users can subscribe the channel they need.

To achieve our proposed platform, it’s necessary to overcome several difficulties including streaming systems, network environment, and functions adapted for suitable services by P2P technology. We decided to focus on the generating of stream session by peers allocated each environment and service continuous in unstable environment. The experimental results which shows the streaming service with best effort used hundreds of sessions.

This system consists of four part, the hardware part (IP Camera), the middleware part (receive video stream from IP Camera), the Image Service Platform part (supply the services users need and receive the video stream from the Middleware) and the Client part (show the real-time image and other functions).

In this paper, the related works are shown in Section II, the design of this system is shown in Section III, the simulation experimental are described in Section IV and the paper is concluded in Section V.

2. Related Works

Open API means everyone can use this API. API is Application Programming Interface. Open API is recent trend in service oriented web applications and companies offer their websites as services to the third-party developers. In the original indoor IoT system, we realize it by using WCF technique based on .Net framework platform which is provided by Microsoft.

SOAP (Simple Object Access Protocol) is a protocol specification for exchanging structured information in the implementation of web services in computer networks. It uses XML Information Set for its message format, and relies on application layer protocols, most often Hypertext Transfer Protocol (HTTP) or Simple Mail Transfer Protocol (SMTP), for message negotiation and transmission.

SOAP has three major characteristics. Firstly, it is extensibility (security and WS-routing are among the extensions under development); then it is neutrality (SOAP can operate over any protocol such as HTTP, SMTP, TCP, UDP, or JMS); the last one is independence (SOAP allows for any programming model). A SOAP message is an ordinary. Table 1 shows the XML document containing elements.

Table 1. XML Elements

Element	Description	Required
Envelope	Identities the XML document as a SOAP message	Yes
Header	Contains header information.	No
Body	Contains call, and response information.	Yes
Fault	Provides information about errors that occurred while processing the message.	No

REST (Representational State Transfer) is an architectural style, which is often used in the development of web services. REST is a popular building style for cloud-based APIs. A RESTful API means web services used REST architecture. REST architecture involves reading a designated web page that contains an XML file, which describes and includes the needed content. REST typically runs over HTTP (Hypertext Transfer Protocol) and is often used in mobile applications, social networking web sites, mashup tools and automated business processes.

REST use a limited number of operations (GET, POST, PUT and DELETE) to enhance the interactions between clients and services. And it is flexible because of assigning resources their own URIs (Universal Resource Identifiers). Table 2 shows how HTTP methods are typically used in a RESTful API.

Table 2. How HTTP Methods are Typically Used in a RESTful API

Uniform Resource Locator (URL)	Collection, such as http://api.example.com/resources/	Element, such as http://api.example.com/resources/item17
GET	List the URIs and perhaps other details of the collection's members.	Retrieve a representation of the addressed member of the collection, expressed in an appropriate Internet media type.
PUT	Replace the entire collection with another collection.	Replace the addressed member of the collection, or if it does not exist, create it.
POST	Create a new entry in the collection. The new entry's URI is assigned automatically and is usually returned by the operation.	Not generally used. Treat the addressed member as a collection in its own right and create a new entry in it
DELETE	Delete the entire collection.	Delete the addressed member of the collection.

SOAP and REST can't be compared directly, since SOAP is a protocol and REST is an architectural style. The main difference between SOAP and REST is the degree of coupling between client and server implementations. A SOAP client works like a custom desktop application, tightly coupled to the server. There's a rigid contract between client and server, and everything is expected to break if either side changes anything. You need constant updates following any change. Table 3 shows a more intuitive comparison between SOAP and REST.

Table 3. A More Intuitive Comparison between SOAP and REST

	SOAP	**REST**
Essence	SOAP is a protocol.	REST is an architectural style.
Full Name	SOAP stands for Simple Object Access Protocol.	Rest stands for REpresentational State Transfer.
Relationship	SOAP can't use REST because it is a protocol.	REST can use SOAP web services because it is a concept and can use any protocol like HTTP, SOAP.
The way of exposing business logic	SOAP uses services interfaces to expose the business logic.	REST uses URI to expose business logic.
Lava API	JAX-WS is the java API for SOAP web services.	JAX-RS is the java API for RESTful web services.
Standards	SOAP defines standards to be strictly followed.	REST does not define too much standards like SOAP.
Bandwidth and Resource	SOAP requires more bandwidth and resource than REST.	REST requires less bandwidth and resource than SOAP.
Security	SOAP defines its own security.	RESTful web services inherit security measures from the underlying transport.
Data Format	SOAP permits XML data format only.	REST permits different data format such as Plain text, HTML, XML, JSON *etc*.

A REST client is more like a browser. It's a generic client that knows how to use a protocol and standardized methods, and an application has to fit inside that. You don't violate the protocol standards by creating extra methods, you leverage on the standard methods and create the actions with them on your media type. If done right, there's less coupling, and changes can be dealt with more gracefully. A client is supposed to enter a REST service with zero knowledge of the API, except for the entry point and the media type. In SOAP, the client needs previous knowledge on everything he will be using, or it won't even begin the interaction. Additionally, a REST client can be extended by code-on-demand supplied by the server itself, the classical example being JavaScript code used to drive the interaction with another service on the client-side.

REST is more dynamic, no need for creating and updating UDDI (Universal Description, Discovery, and Integration). And REST is not restricted to XML format. REST web services can send plain text, JSON, and also XML.SOAP is more standardized.

The use of REST is often preferred over the more heavyweight SOAP (Simple Object Access Protocol) style because REST does not leverage as much bandwidth, which makes it a better fit for use over the Internet. The SOAP approach requires writing or using a provided server program (to serve data) and a client program (to request data).

3. Proposed Architecture for Real Time Video Streaming Based on IP Camera

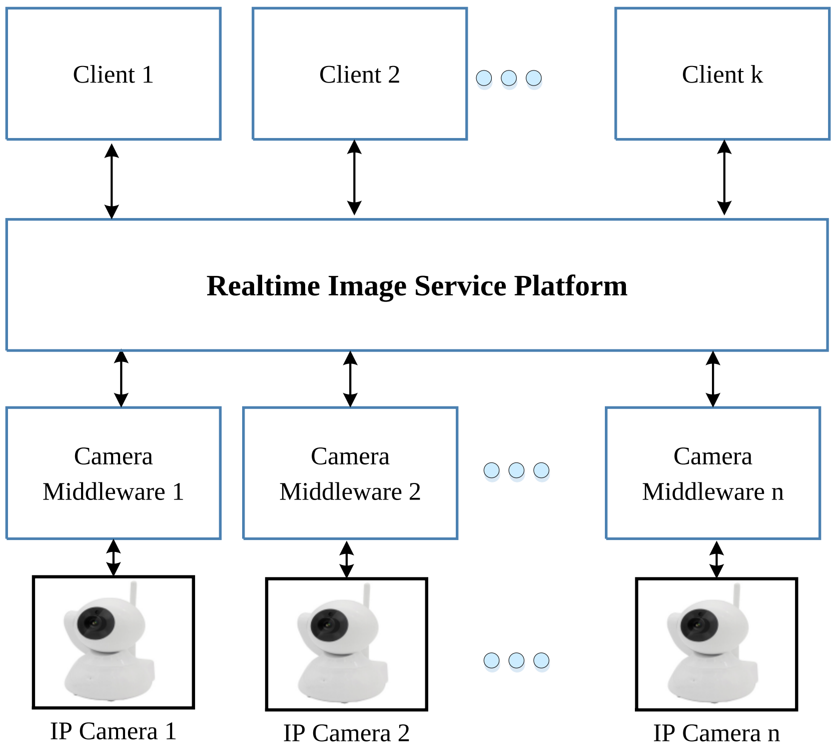


Figure 1. Conceptual Image Service Model

Figure 1 shows the conceptual model of this system. The system consists of four parts: Client, Image Service Platform, Middleware and IP Camera. When user ask to see the real-time video in the Client part, Client will send request to Image Service Platform, then the Image Service Platform will send request to the Middleware, and the Middleware will send request to the IP Camera, IP camera will return video stream to Image Service Platform via the Middleware and Image Service Platform will request the video stream to the Client. Finally, the user can see the real-time video. User can also choose to save video by click “start” and “stop” button. The videos will be saved in the file system of PC. After saving, user can search the saved videos by date, Image Service Platform will return the list of videos’ information to Client.

Figure 2 shows the system architecture we proposed. This system consists of four parts, including the Client part, the Image Service Platform part, the Middleware part and the hardware part. The Middleware part will receive the video stream from IP Camera and send to Image Service Platform. The Image Service Platform part will support provider service (send the video stream to the Client part), store service (user can choose to save video in the Client part) and search service (user can search the saved video by date in the Client part). The client part will show the real-time video and user can control to save videos or search saved videos.

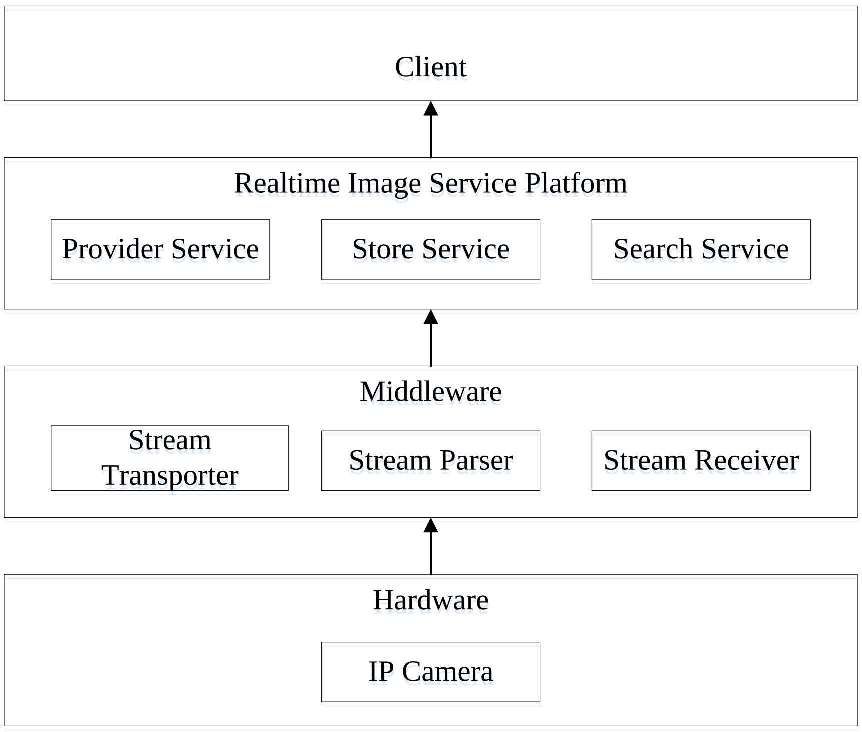


Figure 2. Proposed System Architecture

Figure 3 shows the detailed design of the Image Service Platform. Service State Viewer will show the states of the services (start or stop). Service Control will be able to control the state of the services (start or stop). And there are three interfaces for the three services. The services including provider service, store service and search service. In provider service, stream receiver will receive the video stream send from Middleware, stream transporter will send the video stream to the client. In store service, there is also a stream receiver to receive the video stream send from the Middleware, and stream memory will save the videos in the file system of PC. In search service, there are two functions, searching the saved videos by date in the file system and sending the matched videos' information list to the Client.

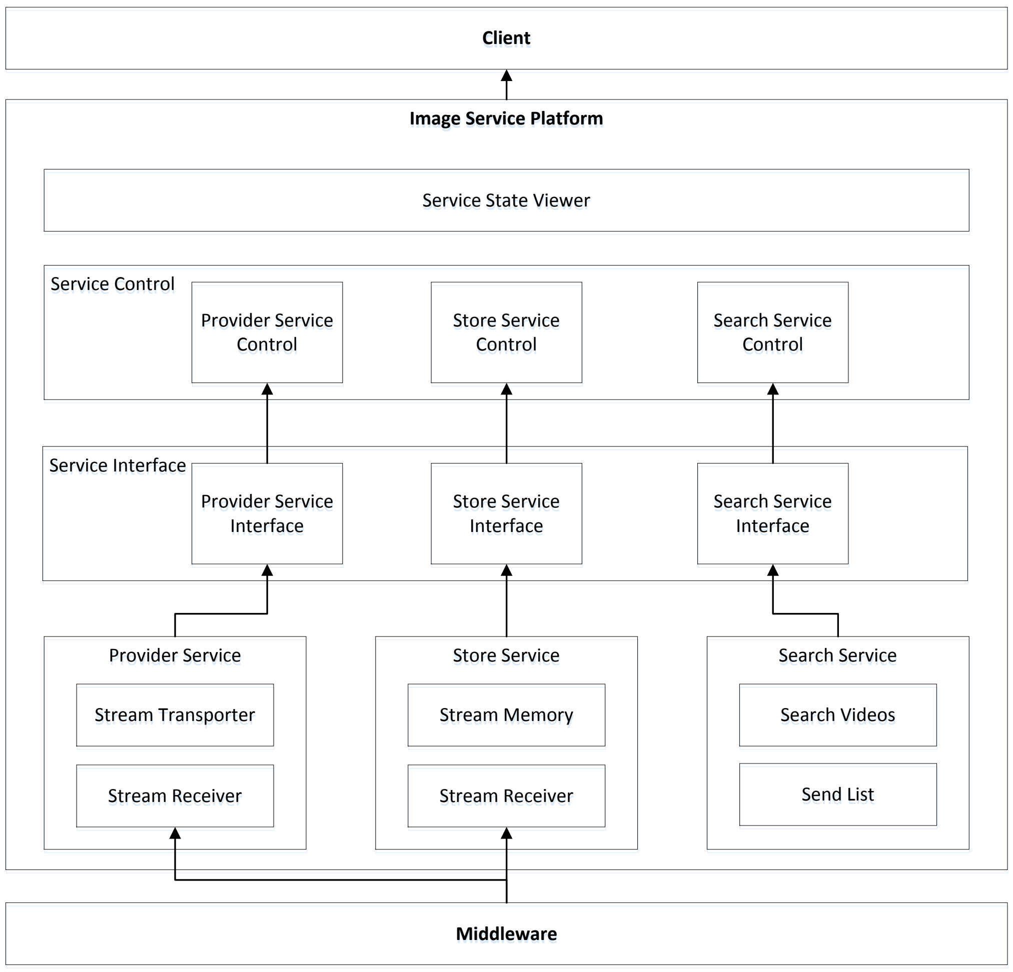


Figure 3. Configuration of Real-Time Image Service Platform

Figure 4 shows the sequence diagram of this system. When Client send "start" request to Image Service Platform, Image Service Platform will send start request to Middleware and Middleware will send request to IP Camera. After this, IP Camera will return video stream to Middleware, Middleware will return video stream to Image Service Platform and Image Service Platform will return video stream to Client. When Client send save video request to Image Service Platform, Image Service Platform will send save video request to Middleware and Middleware will send save video request to IP Camera. After this, IP Camera will return video stream to Middleware, Middleware will return video stream to Image Service Platform and Image Service Platform will save the videos in the file system. When Client send search video request to Image Service Platform, Image Service Platform will search the needed videos by date in the file system of PC and return the matched videos' information list to Client.

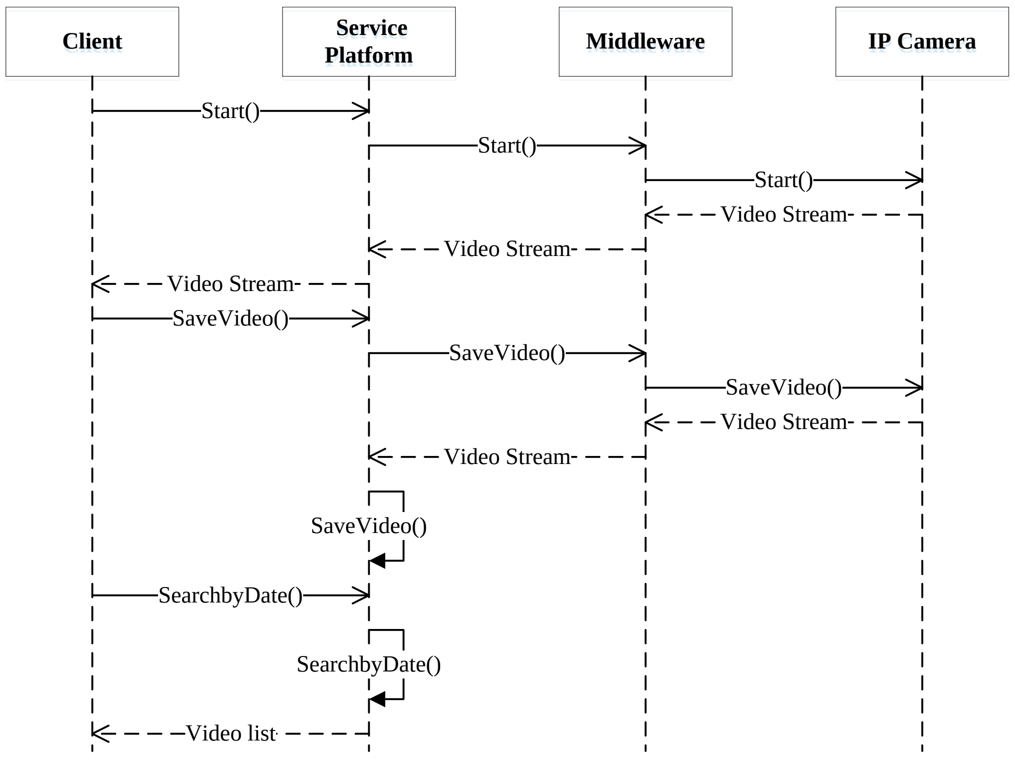


Figure 4. Sequence Diagram

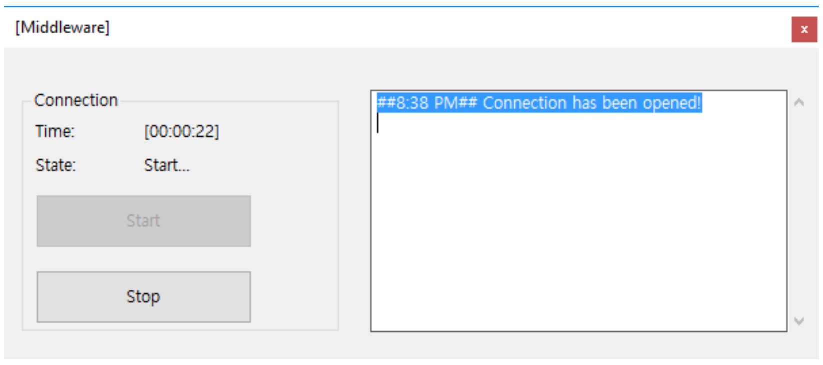
4. Implementation

To implement the design, we need to configure the environment first, Table 4 shows the needed software.

Table 4. Implementation Environment

Components	Version
Operating System	Windows 10 64 bit
Software	Visual Studio 2015
Library	AForge.NET
Hardware	IP Camera

The first part that we should run is Middleware, and the result is shown in Figure 5. We can control the connection through the "Start" button and "Stop" button. "Time" will show the connection time, and "State" will show the connection state. The textbox on the right will show the control message.

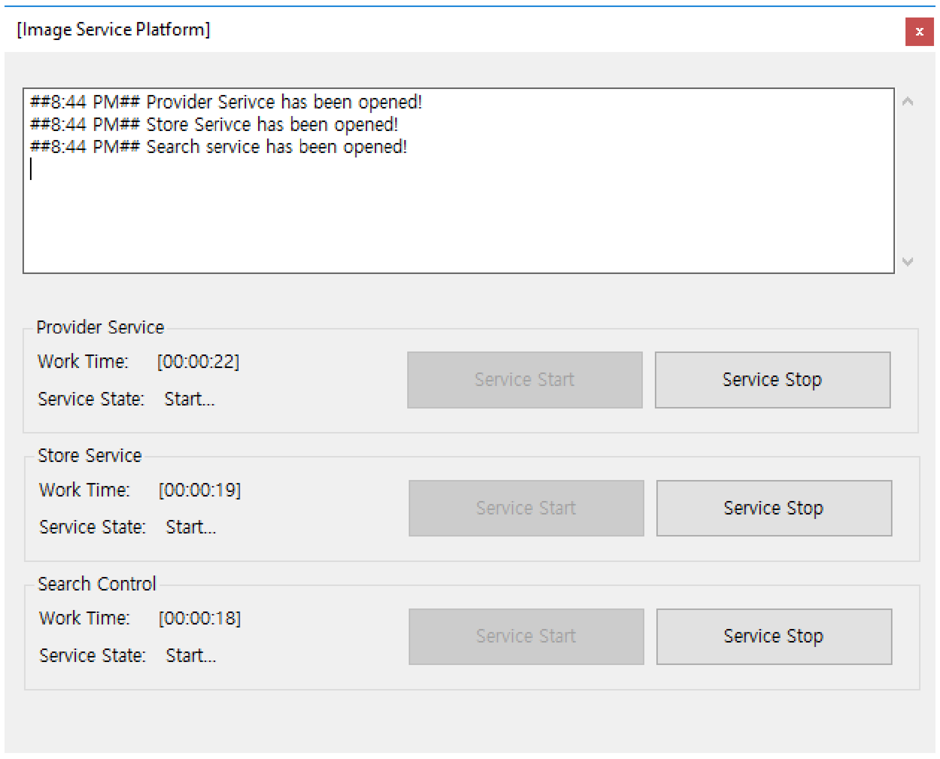


Time: [00:00:22]
State: Start...

```
##8:38 PM## Connection has been opened!
```

Figure 5. The Result of Middleware

After running the Middleware, we should run the Image Service Platform which is shown in Figure 6. The textbox on the top will show the control message and there are three group boxes for the three services. The “Work Time” will show the service’s start time, the “Service State” will show the service’s state (start or stop). “Service Start” buttons are for controlling start services and “Service Stop” buttons are for controlling stop services.



```
##8:44 PM## Provider Serivce has been opened!
##8:44 PM## Store Serivce has been opened!
##8:44 PM## Search service has been opened!
```

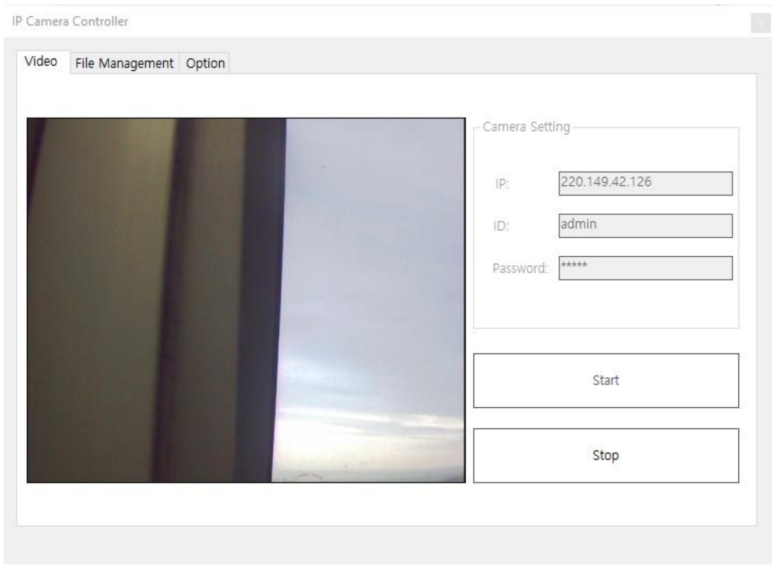
Provider Service
Work Time: [00:00:22]
Service State: Start...

Store Service
Work Time: [00:00:19]
Service State: Start...

Search Control
Work Time: [00:00:18]
Service State: Start...

Figure 6. The Result of Image Service Platform

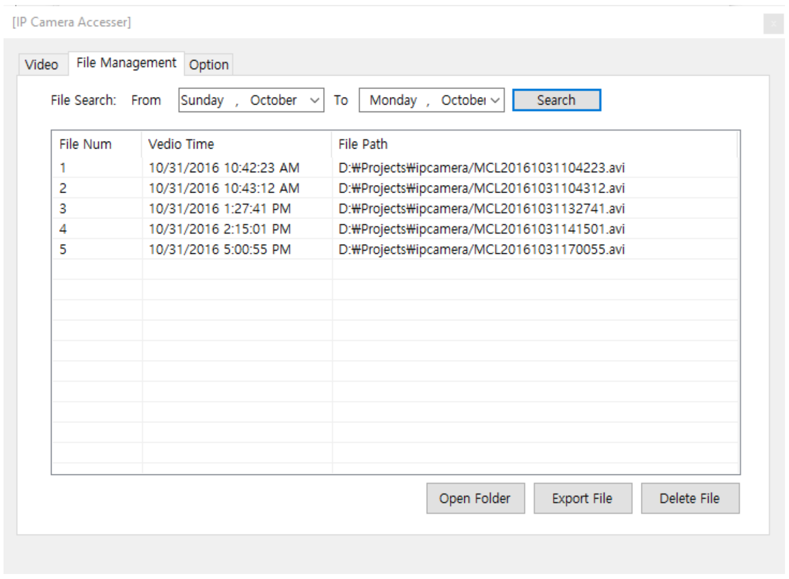
Finally, we can run the Client part and the main page is shown in Figure 7. We should input the IP, user ID and user password of the IP Camera firstly, then click the “Start” button, we will see the real-time image and start recording. We can click “Stop” button to start recording.



IP: 220.149.42.126
ID: admin
Password: *****

Figure 7. The Result of Client's Main Page

After saving the video, we can click "File Management" to switch to the Search video page which is shown in Figure 8. We can choose the date then click "Search" button, then the matched videos' list will be shown. Then we can manage the files, click "Open Folder" to show the file in folder, click "Export File" to export the needed file and click "Delete File" to delete the unneeded files.



File Search: From Sunday , October To Monday , Octobei

File Num	Vedio Time	File Path
1	10/31/2016 10:42:23 AM	D:₩Projects₩ipcamera/MCL20161031104223.avi
2	10/31/2016 10:43:12 AM	D:₩Projects₩ipcamera/MCL20161031104312.avi
3	10/31/2016 1:27:41 PM	D:₩Projects₩ipcamera/MCL20161031132741.avi
4	10/31/2016 2:15:01 PM	D:₩Projects₩ipcamera/MCL20161031141501.avi
5	10/31/2016 5:00:55 PM	D:₩Projects₩ipcamera/MCL20161031170055.avi

Figure 8. The Result of Client's Search Page

5. Conclusions

We have proposed a real-time image service platform based on RESTful API by using IP camera in this paper. Using this system, we can monitor somewhere and recording videos conveniently. And during this study, we have a deep understanding about how to use RESTful APIs. In future, we will try to extend this system by integrating it with IoT Smart Home system. With the help of such system, any user can live in a smart environment happy and conveniently.

Acknowledgment

This work was supported by the Convergence R&D program of MSIP/NST. [Convergence Research-14-2-ETRI, Development of Internet of Things (IoT)-based Urban Underground Utility Monitoring and Management System] and this work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No.B0126-16-1078,Creation of PEP based on automatic protocol behavior analysis and Resource management for hyper connected for IoT Services)), Correspond author : DoHyeun Kim

References

[1] A. Maria, K. Cobb, S. Colli, L. Greborio, R. Mercinelli, R.Walling, Survey of future broadband services. 2002, Ist for Broadband Europe. p. 5-7.

[2] Kelum Vithana; Shantha Fernando; Dileeka Dias,"A service platform for subscription-based live video streaming", 2009 International Conference on Industrial and Information Systems (ICIIS), pp 68 - 73, 2009

[3] Chia-Hsu Kuo, Huan-Ming Hsu, Shu-Chun Ho, Wen-Tin Lee," Universal Middleware Bridge System for IP Cam Networking", IEEE 2nd International Symposium on Next-Generation Electronics (ISNE), pp 25-26, 2013

[4] Takeshi TSUCHIYA, Hiroaki SAWANO, Hirokazu YOSHINAGA, Keiichi KOYANAGI," Streaming Management Platform for Distributed Camera Systems", 2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, pp 392-397, 2009

[5] Yang Luo, Hongbo Zhou, Qingni Shen, Anbang Ruan, Zhonghai Wu, RestPL: Towards a Request-Oriented Policy Language for Arbitrary RESTful APIs, 2016 IEEE International Conference on Web Services (ICWS), (2016).

[6] Chin-Yun Hsieh, Hong-An Hsieh, Yu Chin Cheng, A method for web application data migration based on RESTful API: A case study of ezScrum, 2016 International Conference on Applied System Innovation (ICASI), (2016).

[7] S M Sohan, Craig Anslow, Frank Maurer, SpyREST: Automated RESTful API Documentation Using an HTTP Proxy Server (N), 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE), (2016)

[8] Junying Gan, Xiaolin Wang, Yikui Zhai, A Real-Time Face Recognition System Based on IP Camera and Image Sets Algorithm, 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), (2015).

[9] Ali Tekeoglu, Ali Saman Tosun, Investigating Security and Privacy of a Cloud-Based Wireless IP Camera: NetCam, 2015 24th International Conference on Computer Communication and Networks (ICCCN), (2015).

[10] Chiung-Yao Fang, Chiao-Shan Lo, Su-Han Ho, Shih-Hsien Chuang, Sei-Wang Chen, A Vision-Based Infant Monitoring System Using PT IP Camera, 2016 International Symposium on Computer, Consumer and Control (IS3C), (2016).

Authors

Songai Xuan, is currently perusing M.S. in Department of Computer Engineering, Jeju National University, Republic of Korea. She received her B.S. degree in Computer Science from YanBian University, China in 2016.

Do-Hyeun Kim, He received the B.S. degree in electronics engineering from the Kyungpook National University, Korea, in 1988, and the M.S. and Ph.D. degrees in information telecommunication the Kyungpook National University, Korea, in 1990 and 2000, respectively. He joined the Agency of Defense Development (ADD), from Match 1990 to April 1995. Since 2004, he has been with the Jeju National University, Korea, where he is currently a Professor of Department of Computer Engineering. From 2008 to 2009, he has been at the Queensland University of Technology, Australia, as a visiting researcher. His research interests include sensor networks, M2M/IOT, energy optimization and prediction, intelligent service, and mobile computing.