Distributed Photo Printing System for Smart Phone Based on WeChat

Cheng Wang

Hainan College of Software Technology, Qionghai 571400, China;

naturavo@163.com

Keywords: Photo printing, JSP, socket, WeChat.

Abstract. Smart phone becomes a popular photographing tool, the demand of printing photos on smart phone will certainly has huge growth. WeChat is a mobile communication service developed by Tencent Corporation; it has millions of smart phone users in China. In this paper, we described a distributed photo printing system for smart phone based on WeChat. This photo printing system can bring more subscribers to companies which have public account; it means better branding and generates revenues for businesses.

Introduction

Nowadays, smart mobile phone is a very popular device. As smart phone camera resolution become increasingly higher, more and more people use mobile phone as their primary tool of taking pictures. As smart phone becomes a new tool of photography, the demand of printing photos on smart phone will certainly has huge growth.

WeChat is a mobile text and voice messaging communication service developed by Tencent in China. As of August 2014, WeChat has 438 million active users; with 70 million outside of China [1]. WeChat supports users to register as a public accounts, which enables them to push feeds to subscribers, interact with subscribers and provide them with service. By the end of 2014, number of WeChat public accounts had reached 8 million. Public accounts of organizations can apply for a verified official public accounts. In China, WeChat public accounts have become a common service or promotion platform for government, news media and companies. Specific public account subscribers use the platform for service like hospital pre-registration, visa renewal or credit card service. The development of the WeChat public account market is really heating up in China.

The Design of Photo Printing System

WeChat Public Account Working Principle.

Every public account has a QR code, WeChat user just scans the QR code to subscribe a public account. After subscribe a public account, WeChat user can receive from or send messages (text, image, sound etc.) to the public account.

WeChat has released a series of APIs and services to encourage all types of businesses and organizations to use the WeChat public account system or develop custom features for their accounts.

If a public account wants to develop custom features and interact with WeChat user automatically, it must provide a web server to interact with WeChat users. All user messages, WeChat will send to the web server via http protocol, its working flow as shown in Figure 1.

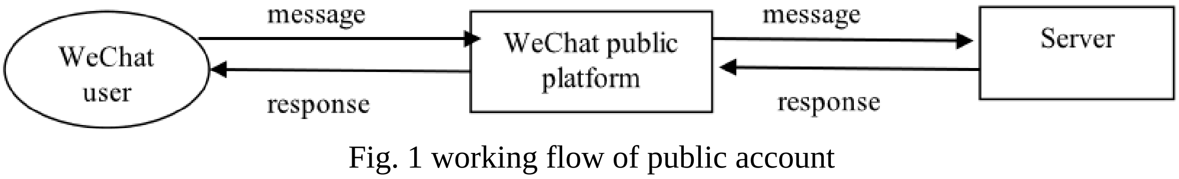

Fig. 1 working flow of public account

WeChat platform has seven type messages, text message, image message, voice message, video message, short video message, location message, and link message. These messages send to the web server in XML text format. For example, an image message has the following format [2]:

```
<xml>
 <ToUserName><![CDATA[toUser]]></ToUserName>
 <FromUserName><![CDATA[fromUser]]></FromUserName>
 <CreateTime>1348831860</CreateTime>
 <MsgType><![CDATA[image]]></MsgType>
 <PicUrl><![CDATA[this is a url]]></PicUrl>
 <MediaId><![CDATA[media_id]]></MediaId>
 <MsgId>1234567890123456</MsgId>
 </xml>
```

Web server can determine which user sends this message by From User Name field, and retrieves the image from PicUrl field.

Photo Printing System.

As described in section 2.1, we need a web server to interact with WeChat public account platform. The server parses and responses to the XML text send from WeChat public account platform by http protocol. As a distributed system, WeChat users can print his/her photos in any location that has a client. Figure 2 illustrates the photo printing system architecture.

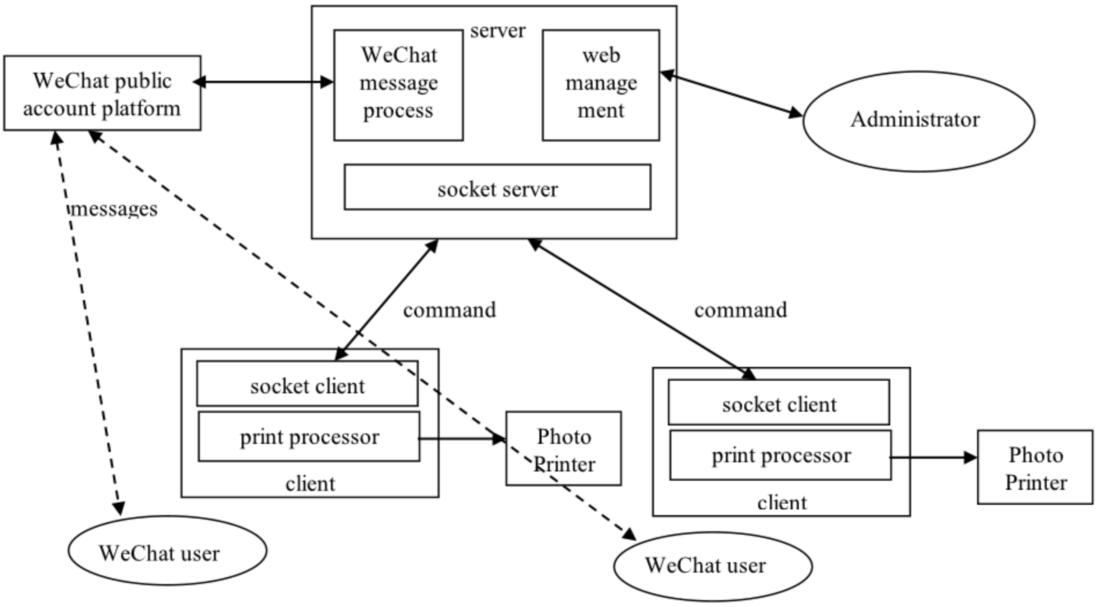

Fig. 2 Architecture of Photo Printing System

In server side, it has three modules. WeChat message process module parse the messages come from WeChat platform, it determines the response messages. As a photo printing system, we must response to WeChat text message and image message at least. How and when the system permit a WeChat user to print photos depends on business logic. In Client side, photo printer connected to a computer system. Client application receives and sends commands by socket, if the client received a print image command, it retrieves image from sever and prints it. For commercial and business logic, we need a web server to control WeChat user’s printing strategy. Web administrator can manage this website by JSP pages.

Implementation

WeChat message process.

Because WeChat public account platform connect third party by http protocol, it is easy to implements this module by JSP servlet.

Public account owner (organization or individual) must configure an url(Uniform Resource Locator) address when register public account, the address can be like this: *http://www.example.com/example*. Normally, there is one address for one account. We can add GET parameter in the url address as: *http://www.example.com/example?serviceid=1*. In the implementation of the servlet, it is easy to obtain the *serviced* parameter to distinguish different account.

Socket server and client.

If client side programming in C/C++, it is straightforward to use socket API to communicate with server, the Winsock programming model of client side in windows illustrated in figure 3 [3].

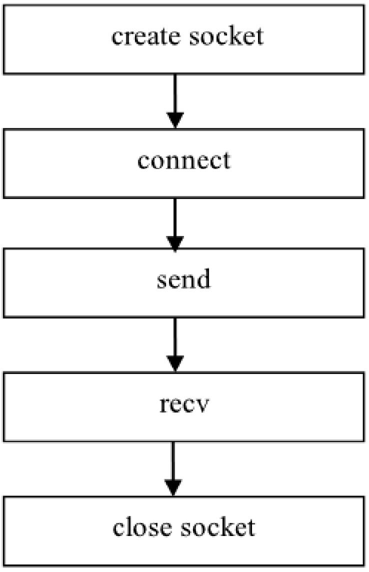

Fig.3 winsock client programming model

Web pages and JSP servlet can only response to http request passively. In our system, if a WeChat user send an image and request print, the server must sends message to client computer which locates at the user’s location and requests it prints the image. To communicate with client by socket, we can use java. nio package to implement a socket server [4, 5], this socket server is a java singleton thread object. Figure 4 illustrates its flow chart of its *run* method.

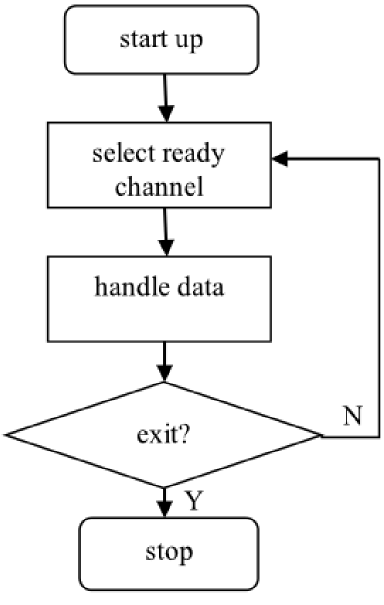

Fig. 4 Flow chart of socket server’s run method

In the implementation of WeChat message process, if the processor wants send a command to client, it can get the singleton socket server object and send command to client.

As a singleton object, socket server must start up when the web application start, so we must add a initializer which implements Servlet Context Listener interface, and register the initializer as a listener class in web.xml.

The initializer can be implemented as follow:

```
public class AppInitializer implements ServletContextListener {
private SocketServer server;
public void contextDestroyed (ServletContextEvent event) {
server.close();
}
public void contextInitialized (ServletContextEvent event) {
server =  SocketServer. getInstance ();
}
}
```

Web management.

Web management module is just a web site implemented in JSP. Any administrator can logs in and manage the site by browser. The management includes photo printer management, public account management, report management etc.

If data changes made by administrator need immediately send to client photo processor, in JSP code, it is easy to get the socket server singleton object and send command to client.

Summary

It is said that there are very a large number of companies on public WeChat account development or operation in China. WeChat public account development market is heating up in China. In this paper, we proposed a distributed photo printing System based on WeChat, this system can bring more subscribers to companies which have public account, it means better branding and generates revenues for businesses. The system development environment: client side use windows 7, IDE tool is Visual Studio 2010; server side use tomcat7.0, IDE tool is Eclipse Java EE IDE, Luna Service Release 2 (4.4.2).

References

[1]. Information on: http://en.wikipedia.org/wiki/WeChat

[2]. Tencent Corporation. WeChat Public Platform developer documentation. Information on: http://mp.weixin.qq.com/wiki/home/index.html

[3]. Anthony Jones. Network Programming for Microsoft Windows, MS.Press. 1999

[4]. Apache Software Foundation. Tomcat Documentation. Information on: http://tomcat. apache. org/ tomcat-7.0-doc/index.html

[5]. Jakob Jenkov. Java NIO Selector. Information on: http://tutorials.jenkov.com/ java- nio /selectors. html