Main architectural patterns of web applications and web services using the example of banking systems

A I Alchinov¹, O V Yalovenko² and M V Sygotina³

¹ Institute of Control Sciences of the Russian Academy of Sciences
² Irkutsk National Research Technical University, 83, Lermontov str., Irkutsk, 664074, Russia
³ Bratsk State University, Makarenko Str., 40, Bratsk, 665709, Russia

E-mail: alchinov46@mail.ru

Abstract. Today, web applications are firmly integrated into our daily lives. We not only use them as games and information resources, but also as means of orders and payments. Naturally, by conducting online payments, registrations and other similar actions, we transfer part of our personal data, as well as card data, phones, etc. Once in the hands of attackers, this data can cause irreparable harm. In this regard, the protection of web applications and services is seen as a very important aspect of modern network security. The content of this paper will be useful both to specialists in the field of information security and to ordinary users.

1. Introduction
Before analyzing the security of any web applications, we need to consider their features and differences [1, 2].

The application itself, for example, a client bank, as the most interesting for attackers, is a program that can be installed on a smartphone or other gadget so that at any time you can perform the necessary actions, such as:

• control the balance and the movement of funds;
• monitor expenditures;
• transfer funds;
• quickly find the nearest branches, ATMs and payment terminals.

Banking applications greatly simplify life for people, but at the same time pose a threat to bank customers. This is especially true in our time, when the number of cybercrime is growing every day and these crimes are mainly aimed at direct or indirect money making.

The most popular is remote banking (terminals, ATM, POS terminals, Internet and mobile banking). All these options of work with a bank contain web services or web applications, as well as the workplaces of operators [3-5].

2. Typology of banking systems
The following may be the users:

• client;
• employee;
• partner.
There are the following types of user interaction:
- customer access to services;
- employee access to automated systems;
- data publication by external users;
- data exchange between automated systems.

By the location of the system in the network infrastructure, all systems may be divided into:
- external (available via the network);
- internal.

This does not end with the typology of banking systems, but it should be borne in mind that each automated system has a web interface or uses certain web protocols.

Remote banking is divided into two main types:
- Internet and mobile banking;
- ATM and self-service kiosks.

It should be borne in mind that there are different types of clients and servers, hence, the client may be:
- thick;
- thin.

3. **Thin and thick clients**

A feature of a thick client is that the system processes most of the information on the user’s computer or on some other device. In case of a thick client, most of the information is stored on the user’s device as temporary files. Since most of the data is processed on the user’s computer, this mode is very demanding of the data channel.

The thin client performs all the actions on the server, the user only displays the received information. This mode of operation does not require large resources of both the system and the communication channel [6, 7].

In turn, the server is subdivided into:
- web application;
- web service.

![Sample web application and web service.](image)

**Figure 1.** Sample web application and web service.

The left image shows an example of a web application, the right image shows a web service.

4. **Web system components**

There are the following main components of a web application:
- Browser and plugins;
- Code (client, server);
- Web Server;
- Application server and framework;
• DBMS;
• Web service providers;
• Load Balancer;
• IDS/IPS;
• WAF;
• Firewall.

The web application architecture uses the following patterns:

**MVC:** Model – View – Controller

*MVC* – the partitioning scheme of the application data, the user interface and the operating logic into three separate components: model, view and controller in such a way that it is possible to modify each component independently.

**Examples of web platforms:**
• PHP: Zend Framework;
• Python: Django;
• JEE: Spring, Struts;
• .NET: ASP.NET MVC;
• Ruby: Rails.

**MVP:** Model – View – Presenter

*MVP* – a user interface design template that was developed to facilitate automatic modular testing and improve the division of responsibility in presentation logic (separation of logic from display). The Presenter element in this template takes over the proxy function (similar to the MVC controller) and is responsible for managing user interface events (for example, using the mouse) in the same way as in other templates the View usually does.

**MVVM:** Model – View – ViewModel

*MVVM* is used to separate the model and its view, which is necessary to change them separately from each other. For example, the developer specifies the logic for working with data, and the designer accordingly works with the user interface.

5. **Interactions in the web services architecture**

The web service architecture has two types of implementation, or in other words two styles of interaction:

1) REST

Example of Rest request:

```plaintext
PUT /services/payment/23 HTTP/1.1
Content-type: application/json
{"to":31, "amount":100.00}"
```

2) SOAP

The Soap query will look slightly different:

```plaintext
POST /url HTTP/1.1
Content-type: text/XML
<?XML version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns: ...> <soap:Body> <GetCustomerInfo xmlns="http://tempUri.org/"> <CustomerID>1</CustomerID> <OutputParam /> </GetCustomerInfo> </soap:Body> </soap:Envelope>
```
Table 1. Differences of application interaction types

| REST                                      | SOAP                            |
|-------------------------------------------|---------------------------------|
| all resources are available on a unique URI | family of standards and protocols |
| resource is key                           | operation is key                |

6. Differences between web applications and web services

The web application uses:
- HTTP(S)
- Internet and Intranet
- “Live Client”

The Web service uses:
- SOAP via arbitrary protocol (in practice – HTTP (S))
- Intranet – more often
- “Client-sutomation”

7. Conclusion

The problems with the security of web applications, in particular the protocols that are the basis of these applications, arose due to the fact that during the creation of the HTTP protocol no one thought that users would attack each other and do so quite actively.

HTTP was originally created to access and receive data from the provider:
- HTTP – stateless, the server is not interested in client status;
- Solution options (cookies, server sessions – analysis of headers, sessions in parameters, hidden parameters).

References

[1] Kondratiev V V, Karlina A I, Guseva E A, Konstantinova M V and Kleshnin A A 2018 Processing and Application of Ultra disperse Wastes of Silicon Production in Construction IOP Conference Series: Materials Science and Engineering 463(3) 032068
[2] Ershov V A, Kondratiev V V, Karlina A I, Kolosov A D and Sysoev I A 2018 Selection of control system parameters for production of nanostructures concentrates. Journal of Physics: Conference Series 1118(1) 012014
[3] Gura D A, Gura A Y, Volkova T A and Lipilin D A 2020 Recreational space as a factor of sustainable development of coastal geosystems of the Krasnodar region IOP Conference Series: Earth and Environmental Science 579(1) 012007 doi:10.1088/1755-1315/579/1/012007
[4] Suslov K, Gerasimov D and Solodusha S 2015 Smart grid: Algorithms for control of active-adaptive network components IEEE Eindhoven PowerTech 232462
[5] Voropai N, Ukolova E, Gerasimov D, Suslov K, Lombardi P and Komarnicki P A 2019 Study on Cost-Effectiveness of Energy Supply Based on the Energy Hub Concept Proceedings of IEEE PES Innovative Smart Grid Technologies Europe 8905736
[6] Balanovskiy A E, Karlina A I, Kolosov A D, Karlina Y I 2021 Study of the effect of nanomodifiers from silicon production wastes on morphological form of gray cast iron graphites. CIS Iron and Steel Review 21 64–69
[7] Sivtsov A V, Elkin K S, Pan’kov V A, Karlina A I 2021 Specific Features of the Electric Mode of the Technological Process of Smelting of Commercial Silicon. Metallurgist 64(9-10) 923–930