to the system functioning, it is important that it is always available and is scalable as the number of modules can grow quickly. As the module management process, the authentication system also lies entirely in the cloud.

As also previously mentioned, when a bank registers a module, it can be found in the cloud through mobile Azure services, or it can be an on premise service. In both cases it is necessary to access the data to perform the desired operations.

Considering now a scenario in which the application intends to access a cloud module, via a mobile Azure service.

Figure 14 depicts the access to a bank cloud module scenario, specifically a mobile service. Briefly, the bank account manager sends a to the application an operations request order, but the application instead of accessing directly the database and returning the results, invokes a line of business that is running on a server. The business line uses a WCF layer to expose their functionalities via SOAP for outdoor applications. More specifically, WFC uses a BasicHttpRelayBinding endpoint to expose their functionalities through the Service Bus.

In order to facilitate understanding of the architecture, we analyze all the steps processed from the order to the results obtainment (Salvatori, 2013).

1. The application calls the mobile service API. The same API contains methods that allow interaction with the data;
2. The API in turn sends a access control service request to acquire the necessary token to authenticate itself in the Service Bus. OAuth authentication protocol is used;
3. The access control service returns a security token;
4. The mobile service uses the API set to extract the wrap_access_token sent the access control service. According to the customer's request the API uses different functions. Each creates a SOAP envelope that invokes a WCF service. The header contains the access token in a format base 64. The body contains the outcome of each call. It is used the node-uuid module to generate a unique id for call security token. Node.js uses the HTTPS module to send the soap envelope to the Service Bus;
5. The Service Bus validates and removes the security token. Then forwards the request to one of the WCF service listeners;

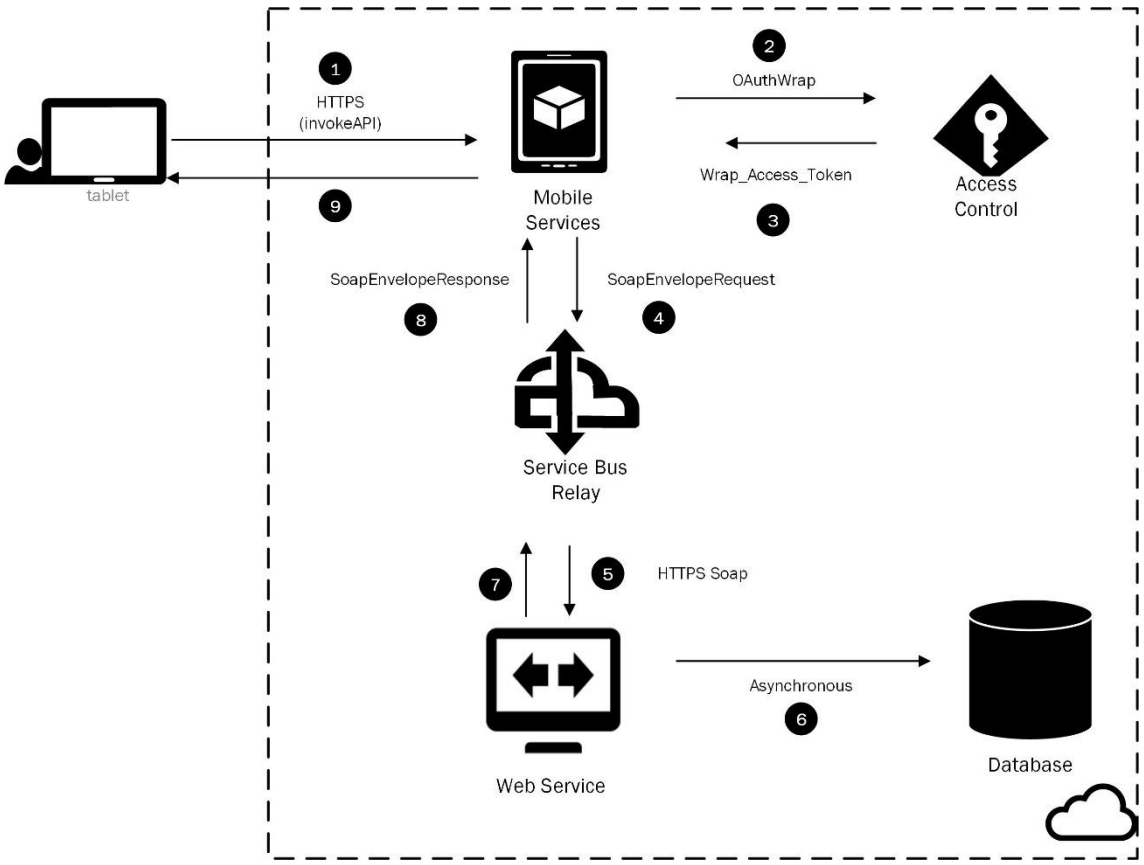

Figure 14. Access scenario to a module in the cloud

6. The WCF service uses asynchronous programming to access data kept in a local database;
7. In turn, the WCF service returns a response to the Service Bus;
8. The Service Bus forwards the message to the mobile service. The API uses the xml2js module to change the format of the SOAP response from XML to JSON. Then flushes the JSON object eliminating unnecessary arrays. Extracts the SOAP representation of data and creates a JSON response;
9. The mobile service returns JSON data to the client application;

This is the application access business line when trying to access a cloud module with a mobile service.

However, as already mentioned, there is also the case of bank registered modules not being in a mobile service but in on premise service. In such case, to be able to support this alternative the architecture has to be slightly different. Let us now analyze the scenario to access a module that is in another server on premise service.

Given the previously presented solution, the main difference to the depicted in Figure 15, it is that it does not use mobile services but only WCF. However this implies that Web services are implemented in data centers. These will later be accessed through the Service Bus Relay as the architecture tries to represent. Briefly, the service to which access is sought is exposed in the Service Bus Relay, when manager making the request, it is first sent to a proxy and there a request to the Service Bus Relay is made. In order to obtain the security token to access to data, the proxy sends a request to Access Control. Now, with the security token it's possible to access the Service Bus Relay where the service is exposed and, in turn, it accesses to the database in order to collect the required data by sending it back to the client.

Let's look in more detail the various steps taken:

1. In order to place an order, the client calls the proxy by sending him the address he wants to access, the method and the input parameters;
2. In the proxy, a request is sent to the access control service to acquire the necessary authentication token to the Service Bus Relay. The authentication protocol used is OAuth WRAP;
3. The access control service returns a security token;
4. Now with the security token, access is made to the Service Bus Relay which subsequently accesses the WCF;
5. Already with access to WCF, this in turn, accesses the SQL database that is on a local server;
6. The data is returned back to the Service Bus Relay;
7. The result of the request is sent back to the proxy;
8. The proxy returns the results to the manager so they can be processed and displayed properly.

Through the architectures shown above, we can see that regardless of the module type the banking entity registers if, it is possible through a hybrid solution to access these same modules. The Service Bus Relay plays an important role by enabling the external services display through the cloud and thus facilitates access to it. The mobile services component also showed a good disclosure, given its backend capacity. As with any system, it is necessary to understand its added value and its shortcomings, well as the advantages and disadvantages of the solution.

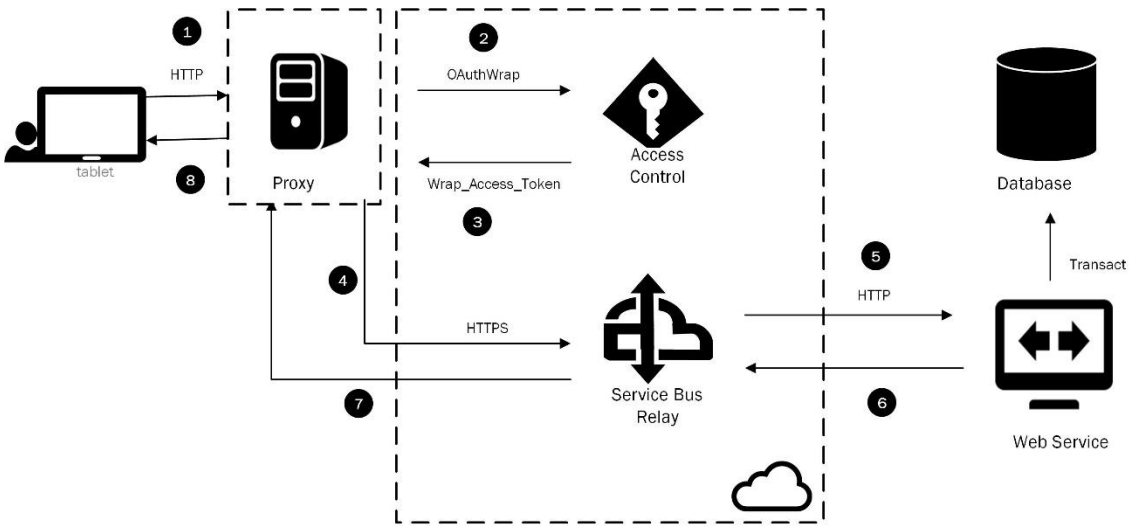

Figure 15. Scenario access to a local module

Advantages

One of the main advantages of this architecture is the fact that it presents clear and well defined business logic. This logic of three layers guarantees the possibility of cloud and local services integration. Through the service bus, it is possible to solve the communication problems between local applications and the outside world. This is advantageous because it allows it to be possible to make use of legacy services.

Another advantage is the multi-tenant capabilities. A multi-tenant application allows resources sharing, separating its users, in this specific case the banks. Thus, each institution sees the application in its own way.

The use of mobile services is an important advantage, as well. Mobile services give the architecture hybrid features since they allow the built of backends for any mobile operating system.

Disadvantages

One of the main disadvantages of the architecture is that the updating the data model is not automatic. The architecture was designed to support a single data model to several banks, where the data only are logically separated. However, if one of the banks needs to extend their line of business, it was not considered any form of automatic update process.

The fact that it's cloud based can also be considered a disadvantage. In addition to the utilization pricing issues, it also adds to the famous vendor lock-in problem (McKendrick J. , 2011). For more than a decade that worked has been done for the construction of solutions based on common standards and protocols that can be built, supported and replaced regardless of their supplier. However, this concept has regressed when it comes to cloud computing. By using a cloud solution based we are using protocols, standards and tools from a particular vendor, making a future migration a costly and difficult task.

CLOSING REMARKS

Conclusions

For some years now the banking sector has bet on electronic solutions in order to provide freedom to their customers so that they can carry out their banking transactions in more convenient way. However, the trend has been changing and has become necessary to develop these same mobile solutions, targeted at business managers. There are already many institutions that provide services for mobile devices, but these, for the most part, are intended solely for customers. Thus, the banking solutions development for commercial management teams has stagnanted, which leads to the continued use of monolithic and isolated systems, which most often than not lead to information silos. These systems eventually entail a distancing between banks and customers because they require the client's presence in a physical desk to perform a wide range of operations.

There are starting to emerge several solutions that claim to provide such services. Many of these are based on SaaS, which in addition to the inherent advantages are also a way for banks themselves to solve some of their problems.

Therefore, a SaaS-based solution was designed, that at the same time could be combined with the advantages of SOA so as to align technology with a structured and well-defined line of business. Some features considered essential for architecture of this kind, such as interoperability issues, scalability and platform usability were drawn. The interoperability issue, with the current banking systems consequent connection was considered crucial, because the disruption with it could in no way be considered. This architecture also accommodates the advantage of being designed to be used simultaneously by multiple banks. The presented architecture ensures that these requirements are considered. It was possible to combine the potential of the cloud and maintain a link with existing systems.

This architecture aims to be the key element for the banking development in emerging countries, bringing the services to the population and allowing growth, as it aids in decreasing the need for physical branches. Clearly there are still many steps to be taken, however it is considered that with TEA the most important step was made: the definition of a reference architecture for the systematization of access by mobile devices, banking systems, in SaaS environments integrated with on-premise solutions.

Future work

The work described throughout this article has still much to develop. In the case of an application with bank implications, it is clear that a rigorous implementation of the concepts presented here is necessary. Thus it is clear that it is necessary an entire security layer in communications within and outside of the architecture.

It will be necessary to ensure that data is not accessible from outside the application, ensuring total safety in the storage and handling of information. Bank's access control should also be monitored and alternative mechanisms for strong authentication should be implemented.

In addition to the security issues, clearly fundamental, there are many other factors that could make this project a strong channel of communication between banks and customers. Some of the challenges include improvement of the ability to access data offline. Often, especially in developing countries, access to the network cannot meet the needs, making that the communication is lost. Thus, it was desirable that it was possible an offline operations execution scenario. However, this scenario also presents competition problems in terms of changes made by banks. It would be necessary to determine the precedence of operations and how it could develop the rollback process to the previous situation.

Another interesting challenge to add is the ability to start a process on a tablet and then continue on another device. This challenge involves the distribution of encryption keys of the stored information.

There are many challenges that can sit on top of this proposed architecture. This is an area that despite all the advances that have suffered, it is always possible to take further and thereby reduce costs such as infrastructure physical counters. Many improvement proposals can be added in order to make a sustainable and possible system.

REFERENCES

Alecrim, E., 2013. Infowester. Retrieved August 2014, from, http://www.infowester.com/cloudcomputing.php

B-SaaS, 2012. Obtido em Dezembo de 2013, http://www.b-saas.com/

Buku, M. W. and Meredith, M. W., 2013. Safaricom and M-Pesa in Kenya: Financial Inclusion and Financial Integrity. Washington Journal of Law, Tecnhology & Arts, 8.

Camhi, J., 2013. Bank Tech. Retrieved December 2013, http://www.banktech.com/core-systems/temenos-acquires-us-saas-core-banking-provider-trinovus/d/d-id/1296255?

Cloud Computing for Financial Markets, 2011. CISCO. Retrieved from http://www.cisco.com/web/strategy/docs/finance/cloud_wp_c112D518876.pdf

Cloud Spectator, 2013. Cloud Server Performance: A Comparative Analysis of 5 Large Cloud IaaS Providers. Retrieved August 2014, from http://cloudspectator.com/2013/06/cloud-server-performance-a-comparative-analysis-of-5-large-cloud-iaas-providers-3/

Cpay360, 2013. Cpay360. (Reliable Software Solutions for Community Banks) Retrieved December 2013, from http://cpay360.com/markets/community-banks

CSC, 2013. Banking Software. Retrieved Dezembro 2013, from http://www.csc.com/banking/offerings/11090-banking_software

ETNA Software, 2011. SaaS & SOA in Retail Banking - Combination for Success. pp. 1-6. Retrieved from http://blog.etnasoft.com/wp-content/uploads/2011/03/ETNA_SaaS.pdf

International Telecommunications Unions (ITU), 2013. Key ICT indicators for developed and developing countries and the world (totals and penetration rates). Geneva: International Telecommunications Unions (ITU).

King, B., 2012. Bank 3.0: Why Banking Is No Longer Somewhere You Go But Something You Do (1ª ed.). Wiley.

King, B., 2013. Branch Today, Gone Tomorrow. Marshall Cavendish Business.

Kshetri, N. and Acharya, S., 2012. Mobile Payments in Emerging Markets. IEEE, pp. 9-13.

McKendrick, J., 2011. Cloud Computing's Vendor Lock-In Problem: Why the Industry is Taking a Step Backward. Forbes.

McKendrick, J., 2011. Cloud Computing's Vendor Lock-In Problem: Why the Industry is Taking a Step Backward. Forbes.

Mell, P. and Grance, T., 2011. The NIST Definition of Cloud Computing. National Institute of Standards and Tecnhology(Recommendations of the National Institute of Standards and Tecnhology), pp. 1-3.

mFoundry. (n.d.), 2013. Mobile Banking & Payments. Retrieved December 2013, from http://www.mfoundry.com/products-services/#mobile-banking

Microsoft Azure. (n.d.). Microsoft Azure. Retrieved from http://azure.microsoft.com/pt-pt/overview/what-is-azure/

Microsoft Azure. (n.d.). Microsoft Azure. (Base de Dados SQL). Retrieved from http://azure.microsoft.com/pt-pt/services/storage/

Microsoft Azure. (n.d.). 2014. Microsoft Azure. Retrieved August 2014, from http://azure.microsoft.com/pt-pt/campaigns/azure-vs-aws/

Microsoft Azure. (n.d.). 2014. Microsoft Azure. Retrieved September 2014, from http://azure.microsoft.com/en-us/documentation/articles/dotnet-develop-multitenant-applications/

Microsoft Azure. (n.d.). 2014. Microsoft Azure. (Serviços Móveis) Retrieved September 2014, from, http://azure.microsoft.com/pt-pt/documentation/services/mobile-services/

Microsoft Azure. (n.d.). 2014. Microsoft Azure. (Barramento de Serviços) Retrieved September 2014, from http://azure.microsoft.com/pt-pt/documentation/services/service-bus/

Microsoft Azure. (n.d.). 2014. Microsoft Azure. Retrieved September 2014, from http://azure.microsoft.com/pt-pt/services/active-directory/

Nasuni, 2013. White Paper: The State of Cloud Storage. pp. 1-16. Retrieved August 2014, from http://www6.nasuni.com/rs/nasuni/images/Nasuni-White-Paper-State-of-Cloud-Storage-2013.pdf

NCR APTRA. (). NCR APTRA Mobile Banking Gateway. Retrieved from http://www.ncr.com/products/banking/mobile-online/aptra-mobile-banking-gateway

Notícias ao Minuto, 2014. Aplicações Bancárias vão Superar Banca Online em cinco anos. Retrieved August 2014, from http://www.noticiasaominuto.com/tech/246870/aplicacoes-bancarias-vao-superar-banca-online-em-cinco-anos

Ribeirinho, V. and Silva, J., 2012. Financial Services - Angola Banking Survey. KPMG.

Salvatori, P., 2013. Microsoft Azure. Retrieved September 2014, from http://code.msdn.microsoft.com/windowsazure/How-to-integrate-a-Mobile-8780500c

Services, BTR., 2013. BTR Services. Retrieved December 2013, from http://www.btr-services.be/our-solutions-services/saas-banking-platform

Silva, J., Teixeira, C. and Pinto, J. S., 2015. Support architecture to desmaterialization of banking operations. 10th Iberian Conference on Information System and Technologies, Aveiro: IEEE. pp. 1-6.

Stevis, M., 2015. The Wall Street Journal. Retrieved February 11, 12, from http://www.wsj.com/articles/african economies-to-grow-4-5-on-average-in-2015-1432544482

Teixeira, C., 2011. Arquiteturas distribuídas. Universidade de Aveiro.