TCU open data services platform: crowdsourcing, civic cloud and civic applications

Monique Louise de Barros Monteiro

is an employee of the Federal Court of Accounts – Brazil. She has an MSc. and a BSc in Computer Science from the Federal University of the State of Pernambuco (UFPE). She holds 12 professional certifications: Big Data Certified Science Professional, Big Data Certified Professional, SOA Certified Governance Specialist, SOA Certified Architect, SOACP, PSM, SAFe Agilist, Oracle Master Java EE 5 Enterprise Architect, MTS, MSP, SCJP e IBM OOAD.

Marcelo Pacote

is an employee of the Federal Court of Accounts – Brazil. He has an MSc and a BSc in Computer Science from the Federal University of Brasilia (UnB). He has a specialization degree in Development of Mobile Applications, from the University of Araraquara (Uniara), and 17 professional certifications: PMP, CBDP, SOACP, SOACC, CSM, CSD, SCJD, SCJP, SCJA, SCEA (I), SCBCD, SCWCD, RUPF, IRIP, CTFL, ITILF and Oracle SQL Expert. He is the author of a book in the field of Information Technology.

Remis Balaniuk

is an employee of the Federal Court of Accounts – Brazil. He has a postdoc in Virtual Reality from Stanford University, U.S.A, and a PhD in Computer Science from the Grenoble Institute of Technology (*Institut National Polytechnique de Grenoble*), France, and an M.A. in Computer Science from the Federal University of the State of Rio Grande do Sul (UFRGS). He has a technology degree in Data Processing from the University of Brasilia (UnB). Currently, he is a professor and researcher at the Catholic University of Brasília.

SUMMARY

This article describes the development, availability and sustainability of a platform for open data exposure and information gathering through crowdsourcing, based on its use by civic applications. Among the results expected from the implementation of this platform, we can mention the opportunities of 1) centralization and availability of open data and 2) obtainment of contributions and perceptions of citizens. Such information may be very useful both for the definition of indicators for public services and policies and also to increase knowledge about citizens' wishes in different localities and population spheres, including the society's perception in relation to the services provided to them..

Keywords: civic applications, web services, crowdsourcing, civic cloud, open data.

1.   INTRODUCTION

A recurring theme in Public Administration in Brazil and in other nations is the sharing of open data for greater transparency of government actions.

For example, we currently have in Brazil the **Portal Brasileiro de Dados Abertos**[1] (Brazilian Open Data Portal), a tool made available by the government so that everyone can find and use data and public information. The portal includes data on budget execution,

cartographic information and service units, as well as different indicators and statistics.

Among the factors responsible for catalyzing initiatives such as the *Portal Brasileiro de Dados Abertos* (Brazilian Open Data Portal) is Law 12.527 - *Lei de Acesso à Informação Pública* (Law on Access to Public Information), sanctioned on November 18, 2011. In accordance with the guidelines established by this law, the general rule is to classify information of collective interest produced or guarded by the State as public, in order to guarantee the fundamental right of access to information. Thus, information should be classified as confidential in exceptional cases, when it is essential to the security of society and the State.

In its planning, the *Tribunal de Contas da União* – TCU (Federal Court of Accounts - TCU) includes actions to promote the disclosure of open data through its information vehicles. In this context, since the beginning of 2015 the TCU has internalized databases from various agencies. This includes data from the *Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira* – INEP (Anísio Teixeira National Institute of Educational Studies and Research– INEP), the *Cadastro Nacional de Estabelecimentos de Saúde – CNES*[2] (National Registry of Health Establishments – CNES), Ministry of Social and Agrarian Development (social assistance area), of the *Sistema Nacional de Emprego* – SINE (National Employment System - SINE), *Portal de Compras do Governo Federal* - SIASG/Comprasnet[3] (Federal Government Pro-

curement Portal - SIASG/Comprasnet), of the *Sistema Integrado de Administração Financeira do Governo Federal - SIAFI*[4] (Integrated System of Financial Administration of the Federal Government – SIAFI), of the *Cadastro Nacional de Empresas - CNE*[5] (National Business Register – CNE), among others.

Once the bases are internalized, the data received is made available to society, through consultation by end users or by software systems. The methodology adopted for the provision of the so-called open data is detailed below.

2.   METHODOLOGY

In spite of their usefulness, tools such as the Brazilian Open Data Portal traditionally have as a main characteristic the availability of information in a static and / or crude form (e.g. spreadsheets, files in XML, CSV or HTML format), with a large number of records (e.g. spreadsheets with tens of thousands of lines or files with hundreds of megabytes). However, raw government electronic data are often difficult for ordinary citizens to understand. Therefore, the intervention of people, groups and / or companies with the capacity and availability to develop applications and other technological elements that can translate open governmental data into products of interest and public or private utility is necessary.

Appropriation of much of the data set available by developers of civic technologies is difficult, as these databases, typically offered in raw format, are often of low information value and are available in the form of static files. This feature requires the developer to periodically transfer, understand, process, adapt and make them available in applications. Another common requirement is the need for remote and continuously available IT infrastructure, which implies hosting costs

(e.g. contracting cloud services) that developers have to pay for. Such difficulties limit the potential for open data usage by civic applications.

Aware of this scenario, the TCU adopted a model of performance in the open data ecosystem based on the need to provide civic technology developers with a service platform where their applications can remotely access processed and updated data in a more adequate and optimized way to be processed by client software systems. Another premise is that the platform offered allows applications to store the data they generate.

First, the data received from the databases cited in the introduction are persisted in a corporate relational database.

Once the data persistence step in the TCU storage infrastructure is complete, they are ready to be accessed by both end users and applications through web services. According to Erl et al. (2012), "from a general perspective, a service is a software program that makes its functionality available via a published technical interface, called a service contract". Considering that such software services may be implemented by different means, we will use in this article the name web service to refer to services available on the World Wide Web.

Each web service is responsible for grouping and exposing a set of related operations. Such operations can be accessed by externally developed applications to obtain data through standardized protocols according to World Wide Web standards. Such protocols include, for example, Hypertext Transfer Protocol (HTTP) (FIELDING et al., 1999).

Figure 1 shows the concept of web service and its operations at a conceptual level. We have two web services represented and each of them deals with a certain "theme", grouping query operations that will be executed on the persisted database. Each operation, in turn, supports the receipt of parameters that allow you

Figure 1:
High-level representation of two web services. One groups query operations on schools, and another, query operations on health establishments
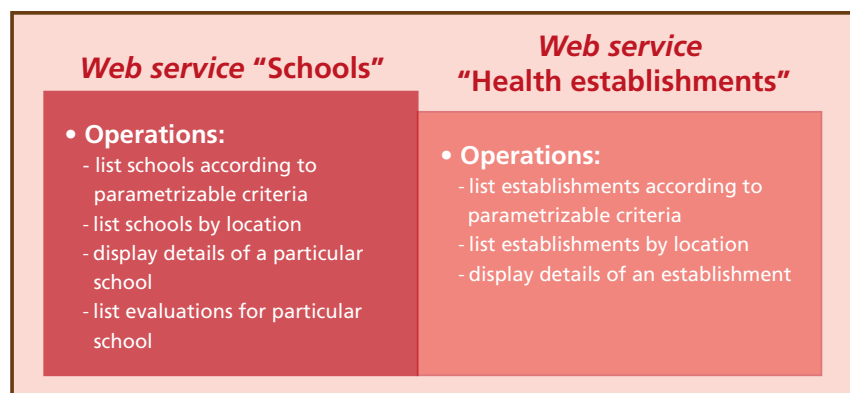
Source: Own elaboration

to filter dynamically the search criteria as well as the attributes and amount of data returned.

3.   DEVELOPMENT

The web services were initially designed as a support tool for the development of civic mobile applications, in the context of the *Projeto BEPiD - Programa Educacional Brasileiro de Desenvolvimento para IOS* (BEPiD Project - Brazilian Educational Development Program for IOS), through a partnership between TCU and *Universidade Católica de Brasília - UCB*[6] (Catholic University of Brasilia – UCB). This partnership involved the development of civic applications by students of that university, who used the information channels provided by the web services developed by TCU.

Web services packages were specified and developed for the following groups of information:

- **Health**: information on medical specialties, health establishments, professionals, medicines and specialized services, all within the scope of public services.

- **Social assistance**: information from referral centers and specialist reference centers for social assistance (CRAS and CREAS, respectively).

- **Jobs**: information regarding SINE posts.

- **Education**: education information, including basically registry data of educational establishments and their resources.

We will name the set of web services mentioned above "civic cloud". From here on, to refer specifically to the set of operations defined in the contract or interface provided by web services, we will use the acronym API, acronym for Application Programming Interface.

The API initially proposed and made available followed the requirements expected by the "*Mapa da Saúde*" (Health Map) and "*Nossa Escola*" (Our School) applications developed by UCB students.

Due mainly to the need of collecting feedback from citizens - a practice known worldwide as crowdsourcing - the applications were developed following a social network model, thus requiring a registry of users and of public service evaluations provided by them. The **metamodel for civic applications** emerged from this. It is an additional set of web services, which allows consultation, addition, change, and deletion of information related to users, applications, user groups, posts, hashtags, notifications, and so on.

In addition, in order for the same model to be adopted both for public health and for education, as well as for other areas of expertise of the State, it is possible

for users to submit assessments, in the form of **postings**, of generic entities denominated here as **objects**. Thus, a user can evaluate an object, which could be, for example, a school or a hospital, depending on the usage context determined by the application in use.

The masses of data generated by civic crowdsourcing will be invaluable sources of information, in particular for the design of indicators that will allow the discovery of knowledge about the functioning of the public administrative machine, the citizen's needs and perception. Crowdsourcing, when used as a practice of transparency, allows large amounts of data to make sense, and can generate new ideas for the development of projects for society.

This knowledge will serve in the future as input top TCU when planning control actions. Finally, the collection and storage of data in a format defined by the TCU itself facilitates future initiatives of statistical analysis and data mining, both predictive and prescriptive, ranging from simpler analyses to more complex applications based on machine learning.

As an example of a civic application, Figures 2 to 3 show screen shots of the Health Map application. The app has search features for health establishments within a radius of location, display of detailed data for each establishment (including, but not shown in captures, listing of medical specialties and quantitative of professionals by function) and also a functionality for the user to evaluate the quality of care provided.

4. TECHNOLOGICAL ASPECTS

The civic cloud was developed according to the REST style - Representational State Transfer (FIELDING, 2000). REST encompasses an architectural style for the construction of applications and APIs, and roughly, provides a mechanism in which, in the context of
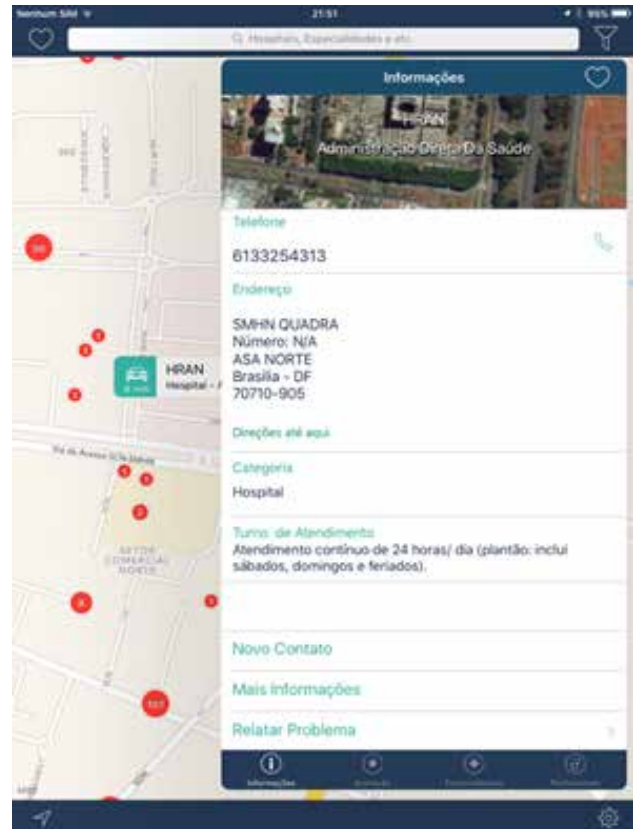
Figure 2:
Health Map - search for health establishment in radius, displayed on a map (satellite view)
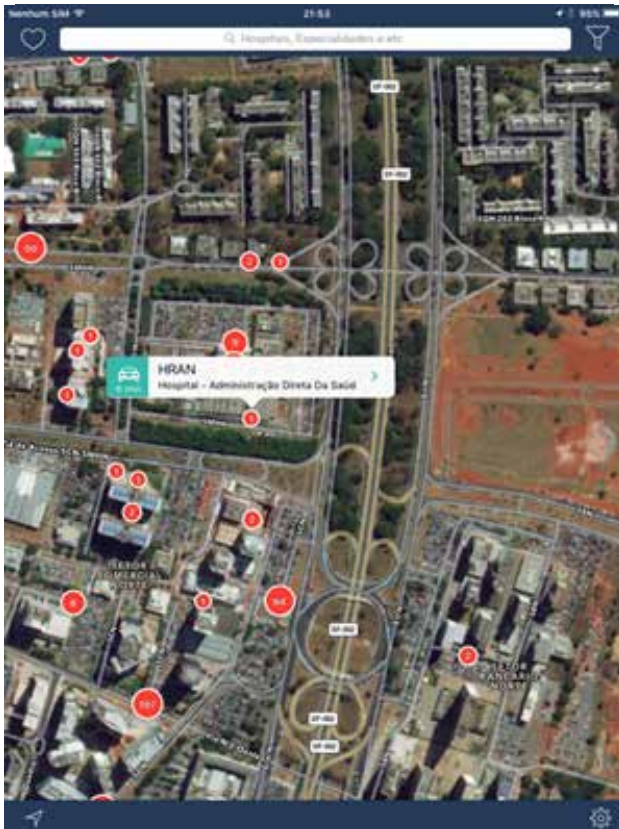
Source: Own elaboration

Figure 3:
Health Map - detail of an establishment

Source: Own elaboration

web services referenced here, each operation of a web service can be represented as follows:

> Method HTTP **(GET | POST | PUT | DELETE) + URI** (*Uniform Resource Identifier*)

Each of the above **methods** represents one of the possible methods or operations supported by the HTTP protocol, to be executed on a **resource** identified by a **URI**. Briefly, the methods have the following semantics:

- GET: Used to retrieve information (ex.: query operations);

- POST: Used to create new features (ex.: register new user);

- PUT: Used to update features (ex.: update an existing user's registration data);

- DELETE: Used to exclude features (ex.: delete a user).

Examples of calls to API operations that can be executed by applications:

- GET <http://contas.tcu.gov.br/nossaEscolaRS/escolas/123>: retrieves information from the school externally identified by the URL <http://contas.tcu.gov.br/nossaEscolaRS/escolas/123>.

It is considered in this example that "123" is the internal school identifier to the application (ex .: key in a database);

- DELETE <http://contas.tcu.gov.br/appCivicoRS/pessoas/456>: delete the user externally identified by the URI <http://contas.tcu.gov.br/appCivicoRS/pessoas/456>. It is considered in this example that "456" is the internal school identifier to the application (ex: key in a database);

- POST <http://contas.tcu.gov.br/appCivicoRS/pessoas> {"nome":"José da Silva", "email": "josedasilva@tcu.gov.br", "dataNascimento":"12/12/1950"}: register a new user, sending their information in the body of the message;

- PUT <http://contas.tcu.gov.br/appCivicoRS/pessoas/789> {"nome":"José da Silva", "email": "novoemail@tcu.gov.br", "dataNascimento":"12/12/1950"}:

The API is documented in the Swagger format. This format allows not only the visualization of the contract of the civic cloud (operations supported, formats of the URIs, format of the data trafficked, parameters for each operation) as well as the test of the functionalities. In addition, the documentation is generated dynamically, being updated with each change in the API contract.

Figure 4:
Example of web services listing (metamodel for civic applications)

AppCivicoRS Web Services referring to the common template for civic applications.			
Contact the developer TCU			
applications: Applications	Show/Hide	List Operations	Expand Operations
groups: Groups	Show/Hide	List Operations	Expand Operations
hashtags : Hashtags	Show/Hide	List Operations	Expand Operations
installations : Installations	Show/Hide	List Operations	Expand Operations
notifications: Notifications	Show/Hide	List Operations	Expand Operations
people: People	Show/Hide	List Operations	Expand Operations
posts: Posts	Show/Hide	List Operations	Expand Operations
object types: Object Types	Show/Hide	List Operations	Expand Operations
post types: Post Types	Show/Hide	List Operations	Expand Operations

Source: Own elaboration

Figures 4 to 6 illustrate the contracts of the meta-model API for civic applications and their visualization through the Swagger interface. The other APIs - health, education, social work, and employment - have similar documentation interfaces.

5.   CONCLUSION AND FUTURE PERSPECTIVES

The Health Map, Our School, My Medication, Mami and Vacin App applications are real examples of applications developed based on the API available. They are available in the App Store and Google Play stores.

In 2016, the TCU provided another opportunity to validate the model for civic applications described here: the Civic Applications Challenge[7]. It is a hackathon for external developers to register applications that make use of the web services available.

At the time this article was written, the contest was in progress with about ninety applications registered. Throughout the Challenge, we have maintained a direct open channel with the developers, through an institutional mailbox and a website in GitHub. Through this channel, we have received more than forty suggestions for adjustments and extensions to the operations provided by the API. Most of the suggestions were implemented and some are in the list of improvements to be implemented in the next version of the API.

Although TCU proposed and implemented the web services platform, it is open to contributions and partnerships. Its success will depend on the convergence of efforts and ideas in the construction of innovative and useful solutions to the citizen. If successful, it will bring results that can contribute to the improvement of public services and policies, providing new services and resources for society as a whole.

NOTES

1   Available on: <http://dados.gov.br>. Access on: 10 out. 2016.

2   Available on: <http://dados.gov.br/dataset/cnes>. Access on: 10 out. 2016.

3   Available on: <http://www.comprasgovernamentais.gov.br/acesso-aos-sistemas/comprasnet-siasg>. Access on: 10 out. 2016.

4   Available on: <http://www.tesouro.fazenda.gov.br/siafi>. Access on: 10 out. 2016.

5   Available on: <http://cne.smpe.gov.br/>. Access on: 10 out. 2016.

6   Available on: <http://www.bepiducb.com.br/index.html>. Access on: 10 out. 2016.

Figure 5:
Example of a web service operations listing

applications: Applications		Show/Hide	List Operations	Expand Operations
GET	/rest/aplicativos	Returns the set of registered applications		
POST	/rest/aplicativos	Registers a new application		
GET	/rest/aplicativos/pessoa/{codPessoa}	Returns the application set of an accountable agent		
GET	/rest/aplicativos/{codAplicativo}	Retrieves the data of a given application by code		
PUT	/rest/aplicativos/{codAplicativo}	Updates the data of an application already registered		
GET	/rest/aplicativos/{codAplicativo}/hashtags	Retrieves the registered hashtags of a given application		
GET	/rest/aplicativos/{codAplicativo}/tipos-perfil	Retrieves profile types created for a given application		
POST	/rest/aplicativos/{codAplicativo}/tipos-perfil	Registers a new profile type for a given application		
GET	/rest/aplicativos/{codAplicativo}/tipos-perfil/{codTipoPerfil}	Retrieves the data of a certain profile		
PUT	/rest/aplicativos/{codAplicativo}/tipos-perfil/{codTipoPerfil}	Updates the data of a profile type already registered		

Source: Own elaboration

Figure 6:
Example of detailing of the contract of an operation

applications: Applications	Show/Hide	List Operations	Expand Operations
GET /rest/aplicativos		Returns the set of registered applications	

Response Class (Status 200)
Model   Model Schema

Application {
    cod (long, optional),
    descricao (string, optional),
    links (Array[Link], optional),
    nome (string, optional),
}
Link {
    href (string, optional),
    rel (string, optional),
    templated (boolean, optional),
}

Response Content Type   application/json
Registers a new application

Source: Own elaboration

7   Available at: <http://portal.tcu.gov.br/desafio-aplicativos-civicos/>. Access on: 10 out. 2016..

6.   REFERENCES

BOOTH, D. et al. (Eds.). W3C. Web Services Architecture. February 2004. Section 1.4. Available at: <https://www.w3.org/TR/ws-arch/#whatis>. Access on: 10 Oct 2016.

BRASIL. Law No. 12,527, of November 18, 2011. Provides for procedures to be observed by the Federal Government, States, Federal District and Municipalities, in order to guarantee access to information, and provides other measures. Diário Oficial da União (Federal Official Bulletin), Brasília, DF, 18 Nov 2011. Extra edition.

ECMA. The JSON Data Interchange Format. Geneva, Oct 2013. Available at: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>. Access in: 10 Oct 2016.

ERL, T. et al. SOA with REST: Principles, Patterns & Constraints for Building Enterprise Solutions with REST. New Jersey: Prentice Hall, 2012. p. 24.

FIELDING, R. T. Architectural styles and the design of network-based software architectures. 2000. Dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Information and Computer Science. University of California, Irvine, 2000.

FIELDING, R. et al. Hypertext Transfer Protocol – HTTP/1.1. June 1999. Available at: <https://www.w3.org/Protocols/rfc2616/rfc2616.txt>. Access in: 10 Oct 2016.