

## Citizen-Centric and Multi-Curator Document Automation Platform: the Curator Perspective

***José António Menezes***

*INESC-ID, Instituto Superior Técnico, Universidade de Lisboa*

*Lisboa, Portugal*

*jose.menezes@tecnico.ulisboa.pt*

***Alberto Rodrigues da Silva***

*INESC-ID, Instituto Superior Técnico, Universidade de Lisboa*

*Lisboa, Portugal*

*alberto.silva@tecnico.ulisboa.pt*

***João de Sousa Saraiva***

*MDSS*

*Lisboa, Portugal*

*joao.saraiva@mdss.pt*

### Abstract

Document automation is an approach that supports the creation of electronic documents in a flexible and efficient way. These systems allow the definition and management of (document) templates, which are extended versions of common documents with particular elements called fields, merge fields, form objects, etc. This paper introduces and discusses qDocs, a citizen-centric and multi-curator document automation platform for managing dynamic electronic documents (e.g. id cards, forms, certificates) accessible to any citizen in a easy and secure way. qDocs provides a single point of access for citizens to create, use and manage their own documents. These documents are produced from templates curated by public or private organizations (named as curators) that also participate in this qDocs ecosystem. This paper discusses particularly how curators may define, design and configure their templates and then make them available to citizens, allowing access to their respective documents in a secure and flexible way.

**Keywords:** Document Automation, Document Template, Citizen-Centric, e-Government.

### 1. Introduction

Document automation is a popular approach that supports the creation of electronic documents in a flexible and efficient way. These systems allow the definition and management of (document) templates, which are extended versions of common documents with particular elements called fields, merge fields, form objects, etc. Document automation systems allow to create and assemble new documents from these previously defined templates by replacing form objects with data collected directly from end-users or from external data sources. Nowadays there are some document automation tools and platforms (e.g., ActiveDocs, SmartDocuments, HotDocs or xPressDocx), but these are mainly focused on document assembly for just one organization purpose and its more technical users, thus not accessible to end-users, in particular to citizens, when they intend to interact with such organizations in the scope of administrative or bureaucratic processes.

This paper introduces and discusses qDocs, a citizen-centric and multi-curator document automation platform, for managing dynamic electronic documents that are accessible to any citizen in a easy and secure way [16,22]. The main purpose of qDocs is to provide a single point of access for citizens to access and manage their own documents, like identification cards, certificates, reports, forms and questionnaires. These documents are provided by curators, which are private or public organizations that are connected to the qDocs platform. A curator creates its documents within the platform as document templates that then become available for citizens in their different roles. qDocs is a collaborative platform that facilitates and promotes the relationship between citizens and curators through processes that involve

the request, publication, access, delivery and sharing of electronic documents.

This paper focuses on the presentation and discussion of the qDocs' features available to the curators, which comprise the creation and authoring of document templates; the configuration of templates, including activation/deactivation of features such as share document, export data, strong authentication, and payment; or the management of general and specific roles to support different levels of responsibilities.

This paper is structured in 7 sections. Section 2 introduces the background and the related work. Section 3 overviews the qDocs platform. Section 4 presents and discusses the qDocs from the curator perspective. Section 5 shows with an illustrative example how qDocs platform works. Section 6 discusses the qDocs platform by contrasting its key aspects with other tools. Lastly, Section 7 presents the conclusion and identifies open issues.

## 2. Background

This research has been influenced by the relevance and impact of document automation platforms in the efficiency of businesses, but also on their lack of being citizen-centric. Figure 1 depicts the processes and artifacts commonly supported by document automation platforms. First (P1), a template manager is responsible for the definition and design of document templates, which are stored in a persistent data storage for document templates. Second (P2), an operator or end-user uses the document templates together with information obtained from end-user interaction or from data-sources to assemble or produce specific documents, which can also be stored in persistent data storage. Document automation (or document assembly) software tools intend to replace the manual filling of form-based documents with templates by allowing, for example, users answer software-driven interview questions. The information collected from end-users or from external data sources can be used with such templates to produce another set of concrete documents [11],[14],[18]. Document automation provides benefits for organizations such as improved efficiency in document production (i.e. documents take less time to be produced), reduced human errors (i.e., end-users only have to provide the necessary information rather than producing a document from scratch), and reduce costs (it takes less time to produce documents so costs are reduced) [1],[13],[20].

Document automation platforms are becoming popular in several areas, such as law [15], public administration [8],[20], e-business [11] or even the aerospace industry [4]. Document automation is particularly popular in business and legal services areas, since it is where a great number of repetitive documents (e.g. contracts, surveys, minutes) are produced [12], [23-24]. For example, "Do-It-Yourself (DIY) law" is becoming a reality and is improved due to these document automation tools, since it became easier for self-representing litigants to produce documents they needed [21]; for example, A2J Author is a platform designed for this purpose [15]. Regarding e-government, document automation is also changing the paradigm from "one-size-fits-all" to "one user, one document" [15], enabling the generation of personalized documents in domains with high variability, having a template that can be reusable and slightly changed by different organizations [20].

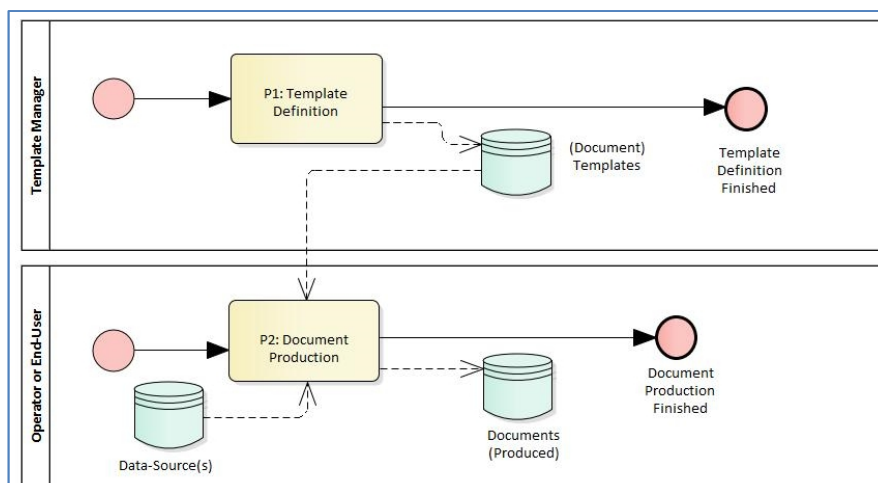


Fig.1. Document Automation Platforms' Main Processes (BPMN diagram)

Popular document automation tools are: ActiveDocs Opus (<https://www.activedocs.com/>), Smart Documents (<https://smartdocuments.eu/en/>), HotDocs (<https://www.hotdocs.com/>), A2J Author (<https://www.a2jauthor.org/>), xPressDox (<http://xpressdox.com/>), Templafy (<https://www.templafy.com/>) and SmartDocs – Acculynx (<http://www.acculynx.com>). These tools share features that support tasks like template definition, document generation and content management; they provide their services via Cloud, Server, APIs, or even Microsoft Office add-ins. Regarding template definition, some use MS-Word as template file format while others also use PDF. They support elements for template creation such as plain text, selection lists, text variables, repeating items and conditional blocks. Regarding document generation, some platforms may receive data from different data-sources as input. The generated documents are produced in multiple formats like PDF, HTML, XML, ODT or TIFF. They offer several integration possibilities and some have extra features such as digital signatures and customized workflow for documents.

ActiveDocs Opus is available via Server or Cloud infrastructures and supports advanced elements such as: links to external data-sources; production of tables, charts and graphs based on data received. It receives input data as XML from DBMSs and APIs, and the generated documents can have several formats such as PDF, DOC and HTML. It integrates with several DBMSs and APIs, Microsoft, SAP, Oracle, DMSs and legacy applications.

Smart Documents offers services that can be integrated with existing CRM or ERP systems. Generated documents receive input data as an XML file from DBMSs. It is deployable as a WS (Web Service) API. The platform has a data retrieval module to integrate with DBMSs and integrates with other WSs and APIs.

HotDocs is available via Server, Cloud or as mix in-between (Hybrid). The used formats for template creation are MS-Word file formats or PDF that works inside the platforms' template editor. It supports an element that is variable sharing between documents, which lets a text variable that was created in a document template to be reused in another document template. Generated documents can be MS-Word or PDF files. Besides the generated document, a file with the information provided to fill the document is created as an XML file so it can be used as input data for future use. The platform is deployable as a WS API, being able to integrate with DBMSs, WSs and APIs.

A2J Author (Access to Justice) is available via Cloud. After the template is created, it is uploaded to the Law Help Interactive (LHI) server, which provides internet-based document automation services powered by the HotDocs platform. All the data used to produce the document is given by an end-user and is sent to the LHI server as an XML file. The generated document is a PDF file. A2J Author is designed to be citizen-centric because its main purpose is to deliver greater access to justice for self-represented litigants by enabling non-technical authors from the courts, clerk's offices, legal services organizations, and law schools to create directly their own contracts from predefined templates [14].

xPressDox provides its services through three different channels: Desktop, Servers and Cloud APIs. xPressDox Desktop works as a MS-Word add-in to create and manage templates. It includes different versions (Supervisor, Author and Runner) that have different permissions regarding a template. Documents that are going to be produced receive input from DBMSs. The generated document is a MS-Word or PDF file. xPressDox is deployable as a REST, .NET or COM API. xPressDox Servers integrate with DBMSs, WSs and APIs.

Templafy enables its features via Microsoft Office add-ins. Templates are defined as MS-Word and MS-Powerpoint file formats. Templates are managed through an external web interface and generated documents are managed through an external web interface or through a content management system. Templafy also has a feature called BrandChecker that is used to check for non-compliant documents within a company, e.g., it detects and alerts if some document templates do not have (and should have) some set of predefined tags.

SmartDocs offers its services through a software platform that is embedded in a CRM platform (Acculynx). Documents can be uploaded and transformed into templates and they can be managed through the cloud. Digital signatures, legally binding contracts and signature notification are key features of the software. Digital signatures make it possible for a user to sign a document online on any device. Legally binding contracts are logs for each user's digital signature that include its name, email address, date, time and IP address.

### 3. qDocs Overview

qDocs is a citizen-centric and multi-curator document automation platform for managing dynamic electronic documents that are accessible through any device [16,22]. It is a collaborative platform that facilitates and promotes the relationship between citizens and curators through any process that involves the request, publication, access, delivery and sharing of electronic documents. Figure 2 presents the general architecture of the qDocs platform; which suggests the integration of four main applications: qDocs/Citizen, qDocs/Curator, qDocs/Admin, and qBox.

Regarding data storage, data stored in one or more curators is dynamically merged in documents generated in real time. The fact that data storage does not leave its respective curator's datacenters, gives curators full control over their information. Document materialization just happens in the citizen's device, increasing privacy and information security: the citizen manages and has full access over his own personal information.

The qDocs ecosystem involves the interaction of the qDocs platform itself, the citizens and the curators. Citizens interact through the qDocs/Citizen application. Curators create, define and configure templates through the qDocs/Curator app and then what delivers the data to the respective citizens is the qBox platform, which is the platform that integrates with all the curators' databases and applications (details of the qBox design are further discussed in [2]).

A citizen to get access to a document, the following tasks have to happen: (1) the citizen requests a document to the qDocs platform; (2) the qDocs platform sends a "request acknowledged" message to qBox; (3) qBox sends a key to qDocs; (4) qDocs sends that key with the document template to the citizen's device; (5) the citizen's device sends the key directly to qBox; (6) qBox sends the respective data to the citizen; and (7) the document is generated in the citizen's device; Finally and optionally (8) qBox may save the data from the citizen so he/she can later on access that document.

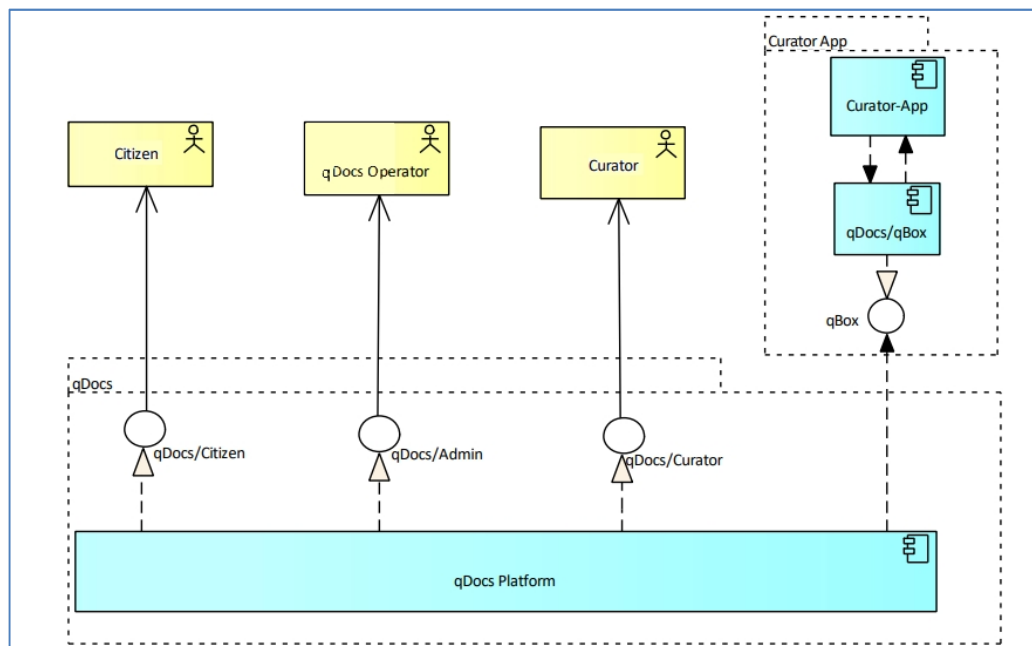


Fig.2. qDocs General Architecture (Archimate diagram)

### 4. qDocs/Curator Perspective

The qDocs/Curator application shall be used by public and private organizations (called "curators" in qDocs terminology). It allows the definition and design of document templates based on the orchestration of data services maintained by different curators, managed according a distributed, dynamic and secure manner. This section presents the key aspects of the qDocs/Curator application, namely, it describes how templates are defined, designed/authored and configured; and also discusses the template's workflow (details of the qDocs/Curator design are further discussed in [17]).

#### 4.1 General Aspects

Figure 3 suggests the main concepts managed by the qDocs/Curator application. A **document template** is represented by two classes: Document Template and Document Template Version. Document Template Version holds the information regarding a particular version such as date of creation and last update and content. Document Template stores the basic information of a document template such as full and short names and state and aggregates one or more versions. Document Template has a state attribute, which is an enumeration and represents its current state. A **data service** is provided by a curator through its respective qBox platform. It is composed by data service methods, which organize the retrieved data in tables. Data service methods are composed by data service method fields, which are the content of the table cells created by the data service method. **Form objects** serve as a link between document template versions and data service method fields. They are associated with document template versions and receive or retrieve data from a data service method field, depending if the data service method field is of input or output type, respectively. The data received or retrieved by the data service method field is then used by the document template. **Snippets** are reusable blocks of text may be used in document template edition; they can contain all the elements available in the qDocs template editor, such as plain text, form objects or even other snippets. Snippets are represented by two classes: Snippet and Snippet Version. Snippet Version holds the information regarding a particular version such as date of creation. Snippet stores the basic information of a snippet such as full and short names and aggregates the first class. Snippet versions may be associated to multiple template versions and vice-versa. **Document template feature configuration** allows to define and configure the active features for the respective document template like export or strong authentication.

In the scope of each curator, users can have general and specific roles. **Curator specific roles** are the ones that are particular regarding a certain curator; e.g. in the scope of an University, a specific role would be a student or a professor, therefore students may request documents that professors cannot and vice versa. On the other hand, **curator general roles** are roles defined at the qDocs level and are common to each curator; qDocs defines the following curator general roles: Curator Administrator, Curator Data Manager, Curator Templates Editor, Curator Templates Manager, Curator Auditor and Curator Documents Manager. Therefore, each curator general role has access to a specific set of features.

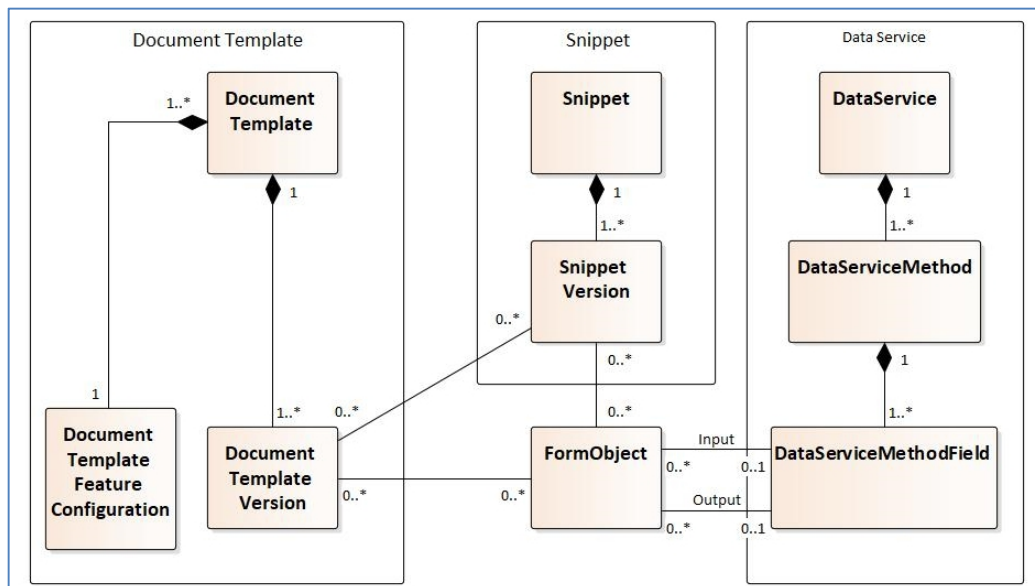


Fig.3. qDocs/Curator Domain Model (UML Class diagram)

The **Curator Administrator** is responsible for managing users at Curator-level and other Curator-level entities such as Document Groups, Curator Specific Roles and Form Object Groups. It is also responsible for managing users, namely associating users to the curator and managing a single user's general and specific roles. Regarding user association, it can delete or add Users. Regarding management of other Curator-level entities, the Administrator can create, edit or delete such entities.

The **Curator Data Manager** is responsible for management Data Services and Form Objects. The creation of a data service comprises the creation of an association between the qDocs/Curator application and the curator's respective qBox platform. He can also create, edit and delete Form Objects, which are data objects that store information retrieved by data services and then are used in template authoring.

The **Curator Templates Editor** is responsible for management document templates and their versioning. He can create, edit, change and delete a document template, including its metadata, i.e. all information regarding a template except for its content, like current template version, short name, full name and state. He can create a new version of a document template after editing an "active" document template and can search and browse through document templates' previous versions. He can also manage the workflow of document templates and configure features for templates (e.g. share document; export document – see section 4.3).

The **Curator Template Manager** is responsible for approve or reject document templates that are submitted for approval, therefore in the "Pending for Approval" state.

The **Curator Auditor** is responsible for analyzing Curator-level activity, i.e. can visualize dashboards and timeline graphics with the most relevant metrics at Curator level (these aspects are out of the scope of this paper and are further discussed in [19]).

The **Curator Documents Manager** is responsible for the approval or rejection of document creation requests made by users (i.e., by citizens with general or specific roles). When a citizen requests a document creation, the Curator Documents Manager consults the request, and then it can approve or reject it. If the request is approved, a new document is created and the citizen (who requested the document) shall have access to it. If the request is rejected, the citizen shall be notified of the involved problem.

## 4.2 Template Authoring

The Curator Templates Editor can create and edit templates directly in the qDocs/Curator app. This editor has similarities to other text editors such as MS-Word, since it has features like text alignment, font size, font color and image insertion.

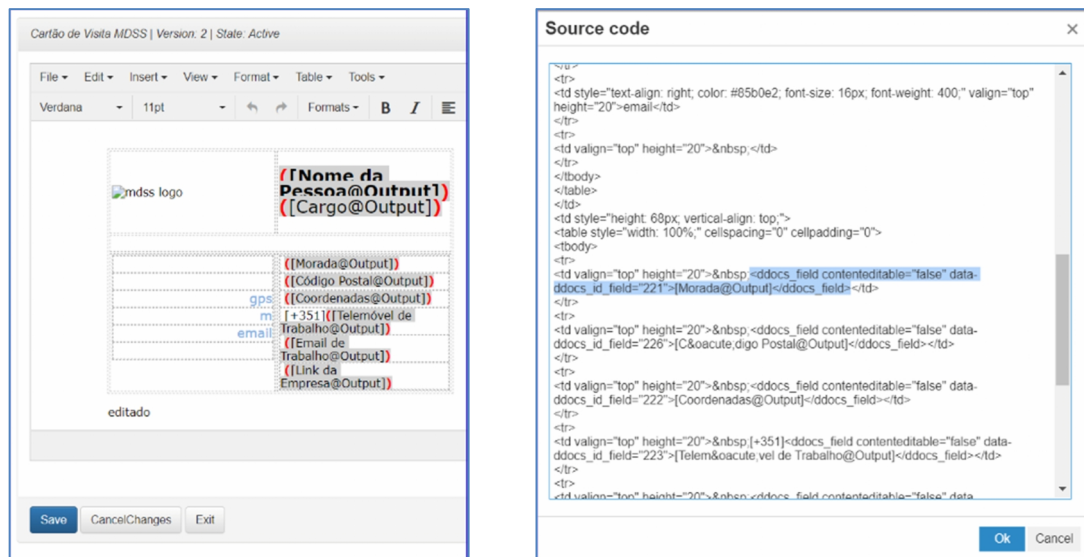


Fig.4. qDocs/Curator: Template (left) with the respective HTML source code (right)

The difference in the qDocs/Curator editor is that it writes everything in HTML and it has particular features such as: Form Objects insertion, i.e. data fields that are going to be filled by citizen input (if the Form Object is of type input) or by curator data directly from his curator's qBox (if the Form Object is of type output); If/Else blocks, i.e. reusable text blocks that only appear if certain conditions are met; and Snippets, i.e. blocks of text pre-defined by the Curator Templates Editor that can include previous features and can be added to multiple templates. These features are identified with a special HTML tag named "<code>< qdocs\_field ></code>". Figure 4 presents the production of a template along with its source code, with the qDocs tag highlighted.

### 4.3 Template Configuration

Besides template authoring in the qDocs/Curator editor, templates have metadata that can be viewed and configured by the Curator Templates Editor. This metadata is composed of the following elements: short name; full name; template version; creation date (i.e. the date of creation of the template's version); publication date (i.e. the date of activation of the template's version when the template's version becomes active); available features; document groups (i.e. life events, general document types and curator specific document types); user groups (i.e. curator specific roles); scope (i.e. if the template's version scope is defined as internal, the template can only be accessible to curator users; if defined as external the template is accessible by any citizen); and state (related to the template's workflow – see section 4.4).

A template can have multiple versions: a new template version is created whenever the content of that template is edited and saved. Curator Templates Editors can always have access to the template's previous versions.

The features that a template can have are the following, which are enabled or disabled by the Templates Editor:

- Document Share: the document can be shared with other citizens via hyperlink or access key;
- Document Data Export: the document's data can be exported to a PDF, CVS or JSON file format;
- Request Strong Authentication: for a citizen to perform a specific action to a document (like create, access, share), he has to go through an authentication process;
- Request Payment: for a citizen to perform a specific action to a document (like create, access, share), he may have to pay a specific value defined by the curator.

If a template has one or more user groups (i.e. curator specific roles) assigned to it, then only citizens that belong to those groups may create or access the respective document. If the document has no user groups assigned to it, then by default any citizen can create or access that document.

Document groups are used to categorize documents, which are further divided into the following classifiers: life events; document types; and curator specific document groups. These classifiers allow citizens to better filter and search for documents.

### 4.4 Template Workflow

A workflow comprises a set of states and events that describe the respective template's life cycle. Figure 5 presents the domain model regarding a template's workflow, showing the states that a document template may have. Figure 6 presents the general state machine diagram of such workflow.

Templates pass through several states and their sequence happens as follows: **created**, after a template is created; **pending for approval**, after the Curator Template Editor edits a template version he submits it for approval or if the template was rejected he can re-submit it for approval again; **rejected**, if the Curator Template Manager rejects it; **active**, if the Curator Template Manager approves a template or the Curator Template Editor activates a template after it is inactive or deprecated; **deprecated**, when a new document template version is created; and **inactive**, if the Curator Template Editor deactivates a template.

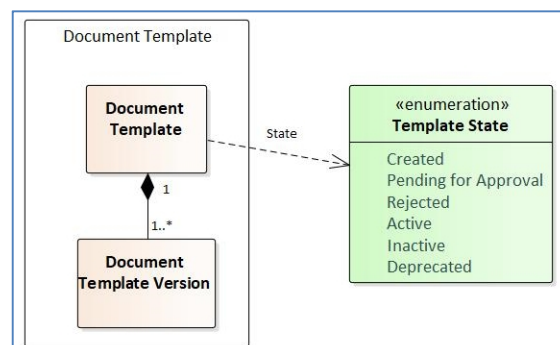


Fig.5. Document Template Workflow Domain Model (UML Class Diagram)

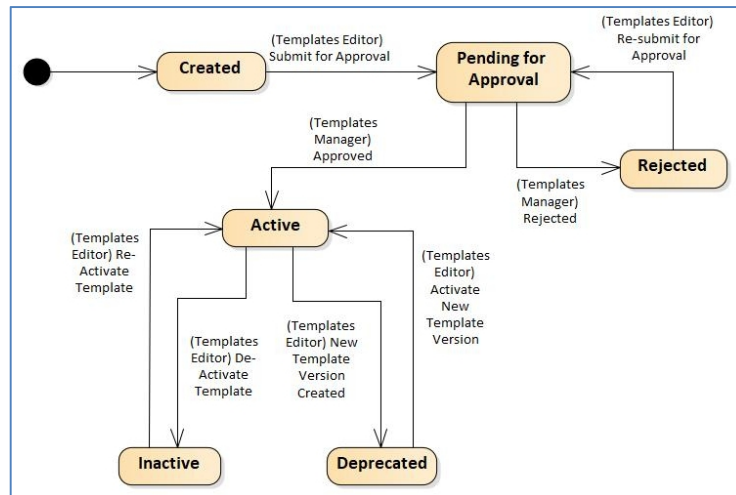


Fig.6. Document Template Workflow (UML State Machine Diagram)

## 5. Illustrative Example

Consider a public institute that produces certificates for citizens. Suppose it joined recently as a new curator to the qDocs platform and the first document that it intends to produce is a residence certificate. Figure 7 shows how the Template Editor of the qDocs/Curator application looks just before saving that template. The key difference between the presented editor and a regular HTML editor is the insertion of Form Objects into the template. In this example the included Form Objects are: ([Name@Input]); ([Address@Input]) and ([Date@Input]). These Form Objects are of input type so a document created with this template will need user input that will be stored in the Curator's database. If any of the Form Objects were of output type, it would mean that a created document would be filled with data from the Curator's database.

After the creation of the template, the Curator Templates Editor can access its metadata and change it according to his needs. Figure 8 shows the form to configure and define metadata for the "Residence Certificate" template.

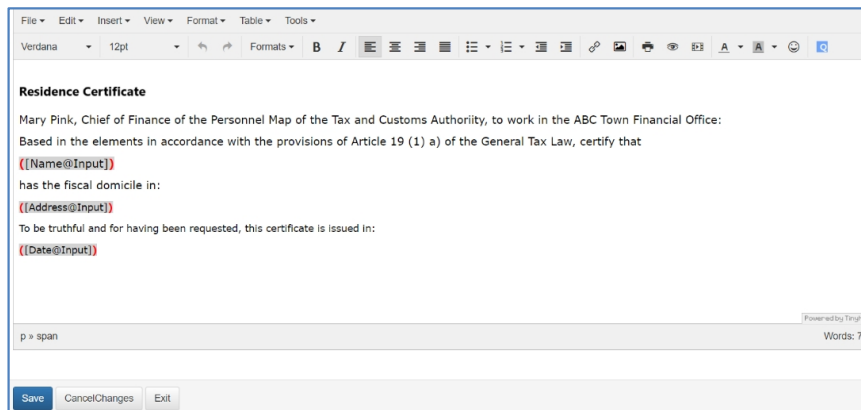


Fig.7. qDocs/Curator: Template Editor

From the properties presented on the left side of Figure 8, only "AllowedSharingTypes" and "AllowedFeatures" can be edited. In this example, the document template is already in version 2, allowing the Templates Editor to check the document template's history if he wants. The date in "CreationDate" and "PublicationDate" is the same because when the document template's content is changed, a new version is created, and it can become active in that moment. A citizen can share his document with other citizens, and they can access it either by using an access key or directly via hyperlink. Those options are only available if they are selected in the edit document template metadata menu, as shown in the Figure 8. As the "AllowedFeatures" of a document template are: (1) "Export", i.e. the citizen can export the document, e.g. as a PDF file; (2) "Strong Auth", i.e. to access the document, a Citizen must go through a Strong Authentication process; and (3) "Payment", i.e. to access the



document, a Citizen must pay the value that is defined by the Curator Templates Editor in the “PaymentValue” field.

On the right side of Figure 8 the properties inside the “DocumentGroups”, i.e. “LifeEvents”, “DocumentTypes” and “OrganizationGroups” are classifiers that may help citizens to search their documents. A “UserGroup” (mentioned before as a Curator Specific Role) can also be selected and if that is the case, only citizens with these roles can access that document. “UserGroups” are also set by the Curator Administrator, as stated before. Templates can be defined as an internal or external scope: if it is internal, only users associated with the Curator can access its documents, e.g. a student from university X cannot access documents from university Y; if it external, it becomes available for any citizen user.

**Fig.8.** qDocs/Curator: Template metadata configuration

**Fig.9.** qDocs/Citizen: Create new document (left), Consult Document (right)

After the document is created, its metadata set, and finally activated, Citizen users can create the document if they have the requirements for it, as explained before. Figure 9 (left) shows the menu presented to the Citizen user that wants to generate a residence certificate. This menu appears in the qDocs/Citizen application. According to what was defined (Figure 8), we can notice that the text boxes are filled with the citizen’s name, home address and the date, starting from the top of the figure, respectively. After filling the text boxes, the citizen saves the document and then it is created. When he wants to consult the created document, a similar form to Figure 9 (right) appears, and then the Citizen has the option to share and export the document, because those features were set previously at the respective template.

## 6. Discussion

After analyzing qDocs along with other document automation platforms, there are some features important for a citizen-centric multi-curator platform like qDocs but not so common in the other platforms that were mentioned in this paper.

Regarding document template definition, Word file formats (like doc, docx) are the most used. Using these formats gives flexibility to new users, because they can import templates that were created with popular tools like Word. qDocs does support HTML format and it provides its editor embedded, providing flexibility to every user, since he can create templates on any device. There is little information about supported elements (and extended elements). However, plain text, selection lists, repeating items, text variables and conditional blocks are common template elements in all platforms, qDocs not being an exception. HotDocs additionally has variable sharing between documents, which lets a text variable that was created in a document template to be reused in another document template. This is the same as qDocs' FormObjects and Snippets.

Concerning document generation, PDF is the most used output file format and Word file formats are common too. HotDocs and ActiveDocs Opus also support XML and qDocs supports JSON and CSV. All platforms can gather data directly from DBMSs. qDocs gathers data from multiple curators via their respective qBox's , which are qDocs' applications that integrate with the curators' databases. ActiveDocs Opus can also gather data from APIs. This data is read as an XML file for all platforms.

Regarding content management, most of the evaluated platforms support this by providing content management within the platform. Only Templafy needs the support of an external web interface and CMS because it offers its service via a Microsoft add-in. ActiveDocs Opus can manage its generated documents through an external DMS. Content management is done within the qDocs platform, although the support of external CMSs would provide more flexibility to both citizens and curators.

Regarding platform availability, most of the platforms offer their services via Server, i.e. they are installed in a dedicated server. Some also offer their service via Cloud API. HotDocs also supports a mix between Server and Cloud. Templafy only provides its services via a Microsoft Office add-in, which is also possible in the xPressDocx platform. qDocs and A2J Author only provide their services via Cloud, which is the most simple and easy way to provide services for a citizen-centric platform.

Concerning the digital signature aspect, only ActiveDocs Opus and SmartDocs - Acculynx support digital signatures; the support of digital signature is scheduled for future releases of the qDocs platform.

Although only one of the platforms was designed to be citizen-centric (A2J Author), besides qDocs, there are some aspects to highlight: the process of document generation adopted should be the same for both citizen-centric and not citizen-centric platforms, since it is a simple and user-friendly method. Regarding availability, cloud is the best and simplest option, because all services become available to users through a single point of access. Digital signatures should be supported to make these platforms more flexible. A feature that may be useful for a citizen-centric platform is a workflow trigger for a document, e.g. when a user creates a document and after it is generated, it could be automatically sent to another party connected to the platform (in qDocs case, the document would be sent to certain curators and citizens, for example). Furthermore, a citizen-centric platform, like qDocs, needs to interact with other entities to gather information. In the scope of qDocs, that is possible because of the qBox application. This would provide better integration with legacy applications, DBMSs and APIs. Regarding template definition, these platforms should be able to support a vast number of file formats, focusing on Word file formats since they are the most common. For template elements supported: plain text, selection lists, repeating items, text variables and conditional blocks are the ones that are most common between the all platforms and they should be adopted. Adding the variable sharing feature would be useful too, as stated before. Content management should be done directly in the platform, since its availability would be via an API in the cloud.

## 7. Conclusion

Document automation platforms have been useful in several business scenarios for different organizations [4-8]. However, most of these systems are particularly focused on document assembly for just one individual organization, not providing direct access to end-users (and in particular to citizens) when they intend to interact with these organizations in the scope of administrative or bureaucratic processes with documents like id cards, forms or certificates. The proposal discussed in this paper overcomes some of these limitations by discussing the architecture of the qDocs platform that allows managing electronic documents accessible in an easy and secure way to any citizen. From the curator perspective, qDocs enables curator users to define and design document templates that are then going to be used by citizens. These templates use form objects and snippets that connect to data service method fields which are indirectly (mediated by qBoxes) connected to curator's databases. This makes the information available in the document generated in the citizen's device secure.

Currently MDSS is conducting a proof-of-concept project, sponsored by the Portuguese Administrative Modernization Agency (AMA I.P.), with a limited number of public organizations, which would allow to validate the qDocs platform. For future work, we intend to apply and evaluate the qDocs in real world scenarios, in the scope of different types of curators, both public and private organizations.

The implementation of additional technical features shall be also considered in the near future, for example: security at communication-level using a Kerberos-based protocol; support further strong authentication services (like the Portuguese [autenticacao.gov.pt](http://autenticacao.gov.pt) for Portugal context); support payment gateways; support additional document-based features directly controlled by the citizen (e.g., features like document export, document send to, or data document export). Integration is another major concern, since qDocs is a cloud-based platform must support interoperability with external systems and that shall be request information from different sources, some of them Curators with legacy applications [9-10]. qDocs shall also be integrated with e-government managed external services and with qBoxes, which are curators' specific but standard way to integrate with data provided by curator' systems [2].

## Acknowledgements

Work partially supported by national funds under FCT projects UID/CEC/50021/2019 and 02/SAICT/2017/29360, and MDSS and AMA project qDocs-PoC/2018. Thanks to colleagues that participated in the design and development of qDocs, in particular Henrique Saias, Rui Craveiro, André Ribeiro, Rodrigo Santos, Daiane Oliveira and Daniel Caramujo, and also Rui Oliveira, Luis Vidigal and Miguel Jacinto for their contributions on the business focus.

## References

1. Betts, K. D., Jaep, K. R.: The Dawn of Fully Automated Contract Drafting: Machine Learning Breathes New Life Into a Decades-Old Promise. *Duke L. & Tech. Rev.* 15, 216, 2017.
2. Caramujo, D.: A Citizen-Centric and Multi-Curator Document Automation Platform: The qBox and Further Interoperability Aspects, Master Thesis. Instituto Superior Técnico, Universidade de Lisboa, 2019 (to appear).
3. Colineau, N., Paris, C., Linden, K. V.: Automatically generating citizen-focused brochures for public administration. In *Proceedings of the 12th Annual International Digital Government Research Conference*, ACM, 2011.
4. Eito-Brun, R., Amescua-Seco, A.: Automation of Quality Reports in the Aerospace Industry. In *IEEE Transactions on Professional Communication*, 61(2), 166–177, 2018.
5. Frank, J.: A2J author, legal aid organizations, and courts: Bridging the civil justice gap using document assembly. *Western New England Law Review*, 39(2), 251–261, 2017.
6. Glushko, R. J., McGrath, T.: Document engineering for e-business. *Proceedings of the ACM Symposium on Document engineering (DocEng'02)*, 2002.

7. Gordon, T. F.: A Theory Construction Approach to Legal Document Assembly. In *Pre-Proceedings of the Third International Conference on Logic, Informatics and Law*, 1989.
8. Guercio, G. F.: Revolutionary Technology. *Courts Today*, 15(4), 20–27, 2017.
9. He, W., Xu, L. D.: Integration of distributed enterprise applications: A survey. *IEEE Transactions on Industrial Informatics*, IEEE, 10(1), 35–42, 2014.
10. Iqbal, R., Shah, N., James, A., Cichowicz, T.: Integration, optimization and usability of enterprise applications. *Journal of Network and Computer Applications*, Elsevier, 36(6), 1480–1488, 2013.
11. Key, M. E.: The Universalization of Data Interaction Technology Research between Database and Spreadsheet of WORD Based on OLE. *DEStech Transactions on Environment, Energy and Earth Sciences (SEEIE 2016)*, 181–186, 2016.
12. Klemptner, R.: The case for court-based document assembly programs: A review of the New York state court system’s DIY forms. *Fordham Urban Law Journal*, Fordham, 41(4), 1189–1226, 2015.
13. Lankester, R.: *Implementing Document Automation: Benefits and Considerations for the Knowledge Professional*. *Legal Information Management*, Cambridge University Press, 18(2), 93–97, 2018.
14. Lauritsen, M.: *Knowing documents*. *Proceedings of the fourth international conference on Artificial Intelligence and Law*, 1993.
15. Loutas, N., Narducci, F., Ojo, A., Palmonari, M., Paris, C., Semeraro, G.: *PEGOV 2014: 2nd International Workshop on Personalization in eGovernment Services and Applications*. *CEUR Workshop Proceedings*, 1181, 2014.
16. MDSS: *qDocs , Citizen centric document technology, White Paper, Version 1.6*. 2019.
17. Menezes, J.: *A Citizen-Centric and Multi-Curator Document Automation Platform: The Curator Perspective*, Master Thesis. Instituto Superior Técnico, Universidade de Lisboa, 2019 (to appear).
18. Mountain, D. R.: Disrupting conventional law firm business models using document assembly. *International Journal of Law and Information Technology*, Oxford Academic, 15(2), 170–191, 2007.
19. Oliveira, D.: *dDocs/Dashboards: Data Analytics for a Citizen-Centric Platform*, Master Thesis. Instituto Superior Técnico, Universidade de Lisboa, 2016.
20. Penadés, M. C., Martí, P., Canós, J. H., Gómez, A.: Product Line-based customization of e-Government documents. In *PEGOV 2014: Personalization in e-Government Services, Data and Applications (Vol. 1181)*. *CEUR-WS*, 2014.
21. Petro Jr, Nerino J.: *Document Automation: Using Technology to Improve Your Practice*. *GPSolo*, 32, (5), 56–62, 2015.
22. Santos, R.: *dDocs – A Citizen-Centric Electronic Document and Record Management System*, Master Thesis. Instituto Superior Técnico, Universidade de Lisboa, 2016.
23. Smith, R., Paterson, A.: *Face to Face Legal Services and their Alternatives : Global Lessons from the Digital Revolution*. *White Report*, 2014.
24. Wong, M. W., Haapio, H., Deckers, S., Dhir, S.: Computational Contract Collaboration and Construction. *Proceedings of the 18th International Legal Informatics Symposium*, 505–512, 2015.