Framework Architecture Design for Emergency Response System

Marco Ruiz Herrera
Department of Computer Systems and Computation
Universitat Politècnica de València
Valencia, Spain *marruih@posgrado.upv.es*

Juan Sánchez Díaz
Department of Computer Systems and Computation
Universitat Politècnica de València
Valencia, Spain *jsanchez@dsic.upv.es*

Abstract

Emergency management is essential to mitigate the effects of unforeseen situations. However, this task is complex due to the large amount of information and complex procedures to be handled. To address these challenges, it is necessary to have tools that allow flexible responses to problems classified as knowledge-intensive procedures (KIP). In this sense, we propose the design of a framework for an Emergency Response System (ERS) based on Service Oriented Architecture (SOA) that integrates Adaptive Case Management (ACM) and Business Process Modelling (BPM). This framework is characterised by its interoperability with devices and collaborative systems, which allows the creation and association of content related to emergency management, thus improving usability. In addition, it is designed to be scalable, allowing the incorporation of new modular functionalities. Once the development of the framework has been completed, future lines of research will be opened for its validation and comparison with other ERS.

Keywords: Adaptive Case Management, Business Process Modelling, Knowledge-Intensive Procedures, Emergency Response System, Framework Architecture.

1. Introduction

The response procedures contained in the emergency plans are developed to respond to possible scenarios to which an organisation is exposed. The response procedures define the actions to be taken, the resources available, and how these resources are used to respond to emergencies. Due to the nature of the knowledge-intensive problems (KIP) related to emergency response, whose variables are unpredictable and unrepeatable, it is recommended that the design of technological tools that provide support during the response allow response procedures to be executable, scalable, adaptable, and flexible.

Likewise, during the response to an emergency, different actors from different agencies usually participate. Therefore, multidisciplinary coordination for emergency management represents a significant challenge [1].

The design of the architecture of a framework for an Emergency Response System (ERS) is proposed, which focuses on the interpretation and execution of response procedures that are represented by models integrated with Adaptive Case Management (ACM) and Business Process Modelling (BPM)[2]. With this, the personnel in charge of managing emergencies will have a tool capable of executing response procedures, providing critical information related to the incident, providing flexible responses when interacting with the environment, facilitating interoperability with internal and external systems, and favouring the incorporation of complementary modules.

The Service-Oriented Analysis and Design (SOAD) methodology [3] was used to design the framework architecture. We have considered the laws on civil protection in Mexico [4] and Spain [5], the contributions of previous works on ERS tools, and the opinion of Mexican stakeholders related to emergency management.

In this context, the content of this article is organised as follows: Section 2 describes related work that has contributed to the design of emergency management tools. Section 3 presents preliminary information necessary to address the topic. Section 4 describes the methodology used to analyse and design the device. Section 5 offers the proposed design of the framework architecture and the description of its elements. Finally, section 6 presents the conclusions and possible future work to be carried out.

2. Related Work

Several ERS proposals are in the literature, most based on traditional processes [6] and the most recent ones addressing case management as a promising alternative to the emergency domain. Turoff et al. proposed developing a framework to address first responders' communication and information to provide support in the decision-making process for command-and-control personnel [7]. The framework was named The Design of a Dynamic Emergency Response Management Information System (DERMIS), which can be used in the emergency management cycle's **Prevention**, **Response**, and **Recovery** phases [8].

Through the Federal Emergency Management Agency (FEMA), the United States government launched the National Incident Management System (NIMS) in March 2004. With NIMS, standardised incident management protocols and procedures were established for all responders to conduct and coordinate response actions [9]. This proposal is committed to standardising response messages of the various agencies involved [10].

Canós et al. presented the SAGA framework, an acronym for Self-protection Management Support System. SAGA provides a platform that offers tools to be used by the actors involved in the different stages of emergency management aligned with the Spanish civil protection regulatory framework [11]. The components that integrate the SAGA architecture are the emergency plan designer with the reuse of pieces of elements. It also has a Self-Protection digital library with a catalogue of emergency plans, fragments of reused response procedures, and a Wiki for exchanging experiences.

In 2015 Kushnareva et al. presented a proposal based on modelling objectives and possible scenarios of crisis management processes using MAP and state charts formalisms [12], [13]. Through these representations, they attempted to delineate the possible scenarios to which each response procedure could be presented. It tried to apply Business Process Model and Notation (BPMN) to the defined workflow.

The ERS consumes and analyses the information from the different intelligent devices transmitted through their communication channels. However, these devices can also receive messages from the ERS to interact. An example of this was presented by Asensio et al. when they proposed an intelligent signalling system implemented successfully in the Monrepos I tunnel in Aragon, Spain [14].

Shahrah and Al-Mashari presented the design of the architecture of a Case Management Framework (CMF) that has Adaptive Case Management (ACM) as its basis for the development of an ERS [15]. With the help of the IBM Case Manager platform, they developed a prototype and evaluated the components through simulations.

Although our proposal to design the framework architecture for developing an ERS has considered the contributions above, it should be noted that our offer is distinguished by integrating ACM and BPM as fundamental principles. Likewise, we highlight its capabilities when interconnecting with external systems and complementary modules.

3. Background

The term framework is commonly used in software systems development, whose purpose must be oriented to a particular field. Generally, when we use the term framework, we refer to a software structure integrated by customisable, independent, and interchangeable components for developing a final application. The main advantages of developing a framework for a final application are accelerating the development process, code reuse, adaptability to different scenarios in the same domain, and promoting good practices in using design patterns [16]. Frameworks contain key features that distinguish them from final applications, such as investment control, deterministic behaviour, extensibility, and non-modifiable framework code [17]. We aim to design a tool capable of improving through flexible responses, managing emergencies, and whose implementation is relatively easy. For this purpose, we chose the device to be a web application framework, which will have an immutable core but is extensible in terms of user-developed functionalities.

Currently, for the design of a distributed hypermedia system, particularly a web application framework that requires to provide high interoperability, the most recommended style standard in the software industry is the Service Oriented Architecture (SOA), where in turn, the Representational State Transfer (REST) architectural style is the one that is regularly used in a web 2.0 context [18].

4. Methodology

Features such as full lifecycle support, extensibility, content reuse, process awareness, and tool support must be considered when developing a framework for an ERS [11], where the design principles and concepts of traditional process and workflow-based systems are not appropriate [6]. An ERS should consider that the caseworker can act flexibly in the face of unrepeatable, changing, and stress-laden scenarios in decision-making [8].

Currently, there is no standard in the industry regarding the development of ACM and BPM platforms; for this reason, several proposals contribute in different ways to apply this methodology [19]. On the other hand, there are a variety of offers from suppliers that use BPM as a technological base. However, the problem is that the same solution only works for some cases and environments. Overall, the user interface should be simple, transparent, customisable, integrated, and scalable to use ERS and respond to emergencies efficiently.

We have considered covering the stages of *Start, Requirements, Design, Build, Verification,* and *Deployment* for the development of the framework. Getting *Started* allows us to define the purpose for which the system will be developed from the stakeholders' perspective. In the *Requirements* stage, stakeholders' needs are identified and classified into user, software, and development requirements. The *Design* stage specifies the software architecture and describes the software parts. The *Build* stage refers to the system's development and parameters' configuration. The *Verification* stage determines whether the developed system meets the needs of the stakeholders. Finally, *Deployment* refers to the transition of the system from its development phase to the operational step in the life cycle of a system [16]. To present the architecture of a framework for an ERS, this paper will address the stages of *Start*, *Requirements*, and *Design*.

During the **Response** stage of the emergency management cycle, the participation of actors from different agencies is expected, so the proper coordination of collaborative tasks is a significant challenge. It is essential to know the activities carried out by the other actors, the stages in which they are involved, the resources available to them, and their recommendations. Therefore, for the *Start* and *Requirements* stages, we interviewed stakeholders from Mexico, one of the most developed countries in civil protection [20].

We have contacted the Directorate of Risk Identification and Analysis of the Civil Protection Secretariat of the State of Chiapas, which is involved in the **Prevention** and **Recovery** stages. We have also contacted the Security Coordination of the Autonomous University of Chiapas, which is interested in the **Prevention** stage. Finally, we have the participation of both the Heroic Fire Department and the Emergency Call Centre 911 of Tuxtla Gutierrez, who participate in the **Response** stage. The following is a list of the functional requirements we gathered from stakeholders:

1) Represent through models the response procedures to be executed.
2) Allow users to store and retrieve in a repository the properties of the model and the activity log of the records of each execution performed.
3) Display through a navigation panel the hierarchical tree listing of the instantiated elements and the search of the details.
4) Allow the user to interact with the model in execution through a map where they can visualise the model image and, using colours, identify the elements available for execution and those that have already been instantiated.
5) Designer of customisable forms associated with the model tasks and their behaviour rules.
6) Allow to create and associating multimedia content and text for supporting documentation.
7) Access to the regulatory framework for emergency management.
8) Tracking of available material, human, and technological resources.
9) Facilitate collaboration with other information systems or devices in the organisation.
10) Support users in making decisions in times of crisis.

11) Management of users, roles, and permissions.
12) Add new modular functionalities to the tool.
13) Auditing activities during executions.
14) Interact with the tool through the web interface and mobile device application.

Once the Getting Started and Requirements stages have been completed, for the Framework Design stage, we have used the Service-Oriented Analysis and Design (SOAD) methodology, which allows us to define the scope of the analysis, identify the existing systems in the organisation and model the candidate services; this last step will enable us to group business logic into available services[3].

5. Design Architecture Proposed

Software architecture descriptions are commonly organised into viewpoints, among which we can find *Logic*, *Process*, *Development*, and *Physical*, also known as 4+1 model architecture views. Each viewpoint addresses system concerns and describes the notations, modelling, and analysis techniques used for architecture [21]. In the following, we present the proposed design of the framework architecture for the ERS, which has modular functionalities and uses Camunda Engine. In addition to Camunda, there are other workflow tools such as Activiti, Bonita, IBM Business Automation Workflow, Oracle Business Process Management, and Appian; however, in general, Camunda offers the following advantages over other tools: *reduced learning curve in terms of ease of use, *intuitive graphical interface, *compatibility with multiple database engines, greater flexibility and customisation through integration with other systems via REST API, *active user community, *wide range of resources available such as documentation, examples and plugins, *strong support for business, case and decision processes, *open source Community edition license with most of its features available. Fig. 1 illustrates the framework design for the *Development* view from a high abstraction approach.

5.1. Main Platform

It is the central component of the framework, where the essential functionalities are found, and the necessary tasks are coordinated. It can offer the user different environments to work in, so it can be used for training caseworkers, validating the behaviour of the models integrated by the BPMN, Case Management Model and Notation (CMMN), and Decision Model and Notation (DMN) specifications. This component comprises a *Management Core* module and five support modules, described below.

- **Management Core**

This module is an essential tool; its functionalities are the *Incident Alert, Procedures Execution, Activity Log,* and *Analytics ER*. The *Incident Alert* will notify the caseworker when an emergency occurs; this event can be initiated through various communication channels. Once a process has been initiated, the *Procedures Execution* functionality comes into operation, with which the response procedures represented by an integrated model can be loaded and executed. The Incident Alert will notify the caseworker when an emergency occurs; this event can be initiated through various communication channels. Once a process has been initiated, the Procedures Execution functionality comes into operation, with which the response procedures represented by models can be loaded and executed.

On the other hand, the *Activity Log* will collect a reliable and immutable record of each activity the caseworker executes. Business Intelligence techniques will be applied in the *Analytics ER* to develop a dashboard to facilitate the analysis of structured and unstructured data stored by the *Activity Log*. Subsequently, it will analyse the users' performance and the tool to audit compliance actions and identify improvements for the response models.

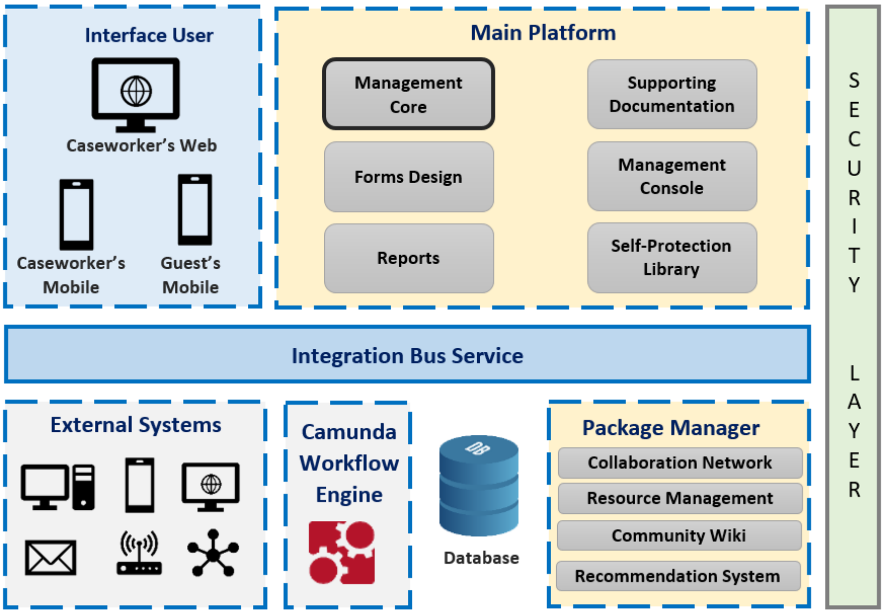

Fig. 1. Framework Architecture Design for an ERS.

- **Supporting Documentation**

During the execution, the caseworker will have a diverse amount and variability of data related to the emergency he is facing. The data may be structured with information related to the inventory of resources. Additionally, there are unstructured data such as text messages, images, or videos. The *Supporting Documentation* component must be able to process, retrieve, archive, associate, and present the data in an organised manner.

- **Form Design**

The user will have the option to design the forms and their elements. In addition, they will be able to add the rules of behaviour and validation of those elements according to the input data and define the output data. It is proposed to develop design components with properties that allow them to modify their visual characteristics and defined events through the drag-and-drop technique.

- **Management Console**

The administrator will have user and permissions management functionalities in the Management Console. During the definition of contracts in the *Integration Bus Service*, the errors to be managed are customised. In addition, it can handle the add-on modules that need to be installed or enabled from the *Package Manager* component.

- **Reports**

The users can design the report templates they will use with the same principle of the Forms Design module where they can add or remove components with drag-and-drop.

- **Self-Protection Library**

Response procedures are objects in traditional text formats or represented through modelling languages. Additionally, there is documentation that regulates the regulations in civil protection that is necessary to organise an agile consultation in times of stress.

5.2. Interface User

This component is critical for the user to have adequate resources when managing an emergency; therefore, its main characteristics for its development are clarity, conciseness, coherence, flexibility, and visual appeal [22]. Three different user interfaces are planned according to user profiles and needs. The caseworker web application is the interface where users can exploit all the tool's functionalities according to their access permissions. The mobile caseworker application is a pocket version, in which the user will have some functionalities available to be informed of the situation during the response and participate in managing the emergency remotely. The guest's mobile application is how the user outside the decision-making process can interact with the tool in activities such as notifying an emergency, consulting the organisation's response procedures, and providing support information about the environment in which they find themself.

During the execution of the response procedures, an interface called *Control Centre* is proposed, where the user will find different sections such as *TaskMap*, *Forms*, and *Assistant*. The *TaskMap* will show the model representing the response procedure and its progress in the workflow. Forms are the section through which the caseworker interacts with the user tasks. From the *Assistant*, the caseworker will have a work area to consult the information about the emergency coming from the *Supporting Documentation, Self-Protection Library, Analytics ER*, *Reports and External Systems* support modules. The user interface must be isolated from the rest of the tool to provide dynamism and flexibility and to take little effort to build the views through which the user will interact.

5.3. Integration Bus Service

The communication contracts of the published endpoints will allude to the functionalities of our agency that other environments will consume, i.e., outbound integrations and functionalities of different settings that our tool will consume. These definitions will identify the general data of the components, the type of integration to know if it is input or output, the management of the connections, the data model of the functionality, the format of the error messages, and the definition of the endpoints and methods.

5.4. Camunda Workflow Engine

This component provides the services to interpret and execute the integrated models in Extensible Markup Language (XML) format [2]. The Camunda BPM server is installed on-premises and contains the Workflow Engine product, which includes a java library called Process Engine and is responsible for executing the BPMN 2.0 specifications for processes, CMMN 1.1 for cases, and DMN 1.3 for decisions [23]. Although the engine provides services to be consumed via REST API [24], the definition of these services will be found in the *Integration Bus Service* component only as a binding reference to the *Procedures Execution* functionality found in the *Management Core* module.

5.5. Package Manager

This component contains the add-on modules catalogue and provides the necessary features to facilitate the installation or uninstallation of each. The add-on modules are extra functionalities that are coupled to the framework. The *Package Manager* executes and controls the installation dependencies of each module. Our proposal is an extensible framework to which new functionalities can be added. Now, we define the following ones:

- Collaboration Network: This add-on module collects data from social networks with the help of a bot that analyses content related to the organisation.
- Resource Management: This module allows the organisation to manage human, material, and technological resources. This will enable the caseworker to schedule tasks to supervise the fulfilment of objectives, review the status of material and technical resources, and estimate the cost of recovery based on the resources.
- Community Wiki: This forum aims to facilitate the exchange of experiences by the actors involved in the various emergency management tasks. The messages' content may include different formats such as text, image, video, audio, documents, etc.
- Recommendation System: With the information gathered by the *Activity Log*, this module will recommend to the caseworker actions to be taken based on the lessons learned from previous similar incidents.

5.6. Security Layer

The *Security Layer* component is critical in providing the protection demanded by an ERS tool, where the most vital asset is the data. It is essential to consider that the information handled in this domain is sensitive data, and any interruption or unauthorised access must be avoided to comply with the regulations of the Protection of Personal Data and Guarantee of Digital Rights of the Government of Spain [25].

6. Conclusion and Future work

The response procedures that are part of the emergency plans establish the tasks, resources, and actors participating in them. The above is to attend to an unpredictable environment produced by an emergency. Despite the importance of providing an adequate response to this type of event beyond the regulatory level, the response procedures are contained in text documents that are difficult to access during the **Response** stage. Therefore, it is essential to have an accessible and easy-to-use framework that provides adequate support to execute response procedures adapt to the environment.

Considering the ACM as a base approach to address knowledge-intensive problems, our proposal focuses on the definition of the architecture of a framework to develop an ERS that facilitates caseworkers to consult and execute the tasks of the response procedures to address emergencies. With this tool, we seek to benefit the personnel managing emergencies within organisations, response teams, and regulatory authorities.

Different stakeholders involved in the emergency management cycle have been interviewed to gather their experiences and needs and to consider their recommendations. With the above, the architecture of the proposed framework focuses on meeting four objectives: the execution of response procedures adaptable to changing environments, bilateral interoperability with other collaborative systems for emergency management, providing a flexible and intuitive user interface, and scalable growth with the incorporation of independent and complementary modules. We have decided to represent response procedures by integrating models based on CMMN, BPMN, and DMN specifications.

Through a catalogue of integrations based on REST API, the communication contracts with the endpoints to be exposed will be defined to make communication transparent with other external collaborative systems, with devices of the environment, and with complementary modules that will provide new functionalities to the ERS.

The architecture design of the proposed framework has great potential to improve the ERS; therefore, the development of a tool prototype is among the lines of future work. This will provide a platform used during all emergency management life cycle stages. During the **Prevention** stage, it can coordinate drills as a training tool for caseworkers and the validation and improvement of response procedures. When the device is sufficiently mature in the **Response** stage, it will support applying the appropriate tasks during emergency mitigation. Finally, during the **Recovery** stage, it will be available to facilitate jobs that will re-establish normal living conditions and rehabilitating service.

ACKNOWLEDGEMENTS

The work of Marco Ruiz was supported by the National Council for Science and Technology (CONACYT) of Mexico. The work of Juan Sánchez Díaz was supported by Spain's "Ministerio de Ciencia e Innovación" under grant MIGRATE (PID2019-105414RB-C31).

References

[1] G. Nestler and A. Jackman, "21st century emergency management," in *Thought Leadership White Paper*, 2014.

[2] M. Ruiz and J. Sánchez, "Improving emergency response through business process, case management, and decision models," *Proc. Int. ISCRAM Conf.*, vol. 2019-May, no. May, pp. 116–125, 2019.

[3] O. Zimmermann, P. Krogdahl, and C. Gee, "Elements of Service-Oriented Analysis and Design," *Design*, 2009.

[4] México. Gobierno de la República, "Ley General de Protección Civil," *D. Of. la Fed.*, pp. 1–39, 2012.

[5] España. Ministerio de la Presidencia, "Ley 17/2015, de 9 de julio, del Sistema Nacional de Protección Civil," *Boletín Of. del Estado*, vol. 164, no. 10 de julio de 2015, pp. 57409–57435, 2015, doi: BOE-A-2006-13554.

[6] M. Hofmann, H. Betke, and Sackmann Stefan, “Process-oriented disaster response management: a structured literature review,” *Bus. Process Manag. J.*, vol. 21, no. 5, pp. 966–987, 2015, doi: 10.1108/BPMJ-07-2014-0069.

[7] M. Turoff, M. Chumer, B. Van de Walle, and X. Yao, “The Design of a Dynamic Emergency Response Management Information System (DERMIS),” *J. Inf. Technol. Theory Appl.*, vol. 1, no. 1, pp. 1–35, 2004.

[8] F. Aligne, “Which information and decision support system for the crisis management,” in *Which information and decision support system for the crisis management*, 2009.

[9] FEMA, “National Incident Management System,” 2019. https://www.fema.gov/national-incident-management-system (accessed Nov. 25, 2020).

[10] M. E. Jennex, “Modeling emergency response systems,” *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, no. February 2007, 2007, doi: 10.1109/HICSS.2007.386.

[11] J. H. Canós, M. C. Penadés, A. Gómez, and M. R. S. Borges, “SAGA: An integrated architecture for managing advanced emergency plans,” *ISCRAM 2012 Conf. Proc. - 9th Int. Conf. Inf. Syst. Cris. Response Manag.*, no. April, pp. 1–10, 2012.

[12] E. Kushnareva, I. Rychkova, and B. Le Grand, “Modeling and animation of crisis management process with statecharts,” in *Lecture Notes in Business Information Processing*, 2015, doi: 10.1007/978-3-319-21915-8_10.

[13] E. Kushnareva, I. Rychkova, and B. Le Grand, “Modeling business processes for automated crisis management support: Lessons learned,” in *Proceedings - International Conference on Research Challenges in Information Science*, 2015, doi: 10.1109/RCIS.2015.7128900.

[14] Á. Asensio, T. Blanco, R. Blasco, Á. Marco, and R. Casas, “Managing emergency situations in the smart city: The smart signal,” *Sensors (Switzerland)*, vol. 15, no. 6, pp. 14370–14396, 2015, doi: 10.3390/s150614370.

[15] A. Y. Shahrah and M. A. Al-Mashari, “Modelling emergency response process using case management model and notation,” *IET Softw.*, vol. 11, no. 6, pp. 301–308, 2017, doi: 10.1049/it-sen.2016.0209.

[16] M. Chaudron, J. F. Groote, K. Van Hee, K. Hemerik, L. Somers, and T. Verhoeff, “Software Engineering Reference Framework,” *Computer science reports*, vol. 0439. pp. 1–12, 2004, Accessed: Nov. 10, 2020. [Online]. Available: https://research.tue.nl/files/1806631/200439.pdf.

[17] M. Imran, A. A. Alghamdi, and B. Ahmad, “Software Engineering: Architecture, Design and Frameworks,” *Int. J. Comput. Sci. Mob. Comput.*, vol. 5, no. 3, pp. 801–815, 2016.

[18] I. G. Gómez and S. O. Tortosa, “Arquitecturas orientadas a servicios,” *CEUR Workshop Proc.*, vol. 132, 2005.

[19] J. B. Hill, K. Chin, and R. Dunie, “Magic Quadrant for BPM - Platform - Based Case Management Frameworks Market Definition / Description,” no. March, pp. 1–11, 2015.

[20] J. Morán, “Panorama del Sistema Nacional De Protección Civil en México,” *Rev. El Col. San Luis*, vol. 7, no. 13, p. 156, 2017, doi: 10.21696/rcsl7132017717.

[21] P. Avgeriou and U. Zdun, “Architectural patterns revisited : A pattern language,” in *10th European Conference on Pattern Languages of Programs, EuroPLoP 2005 - Preliminary Conference Proceedings*, 2005.

[22] F. Calisir and F. Calisir, “The relation of interface usability characteristics, perceived usefulness, and perceived ease of use to end-user satisfaction with enterprise resource planning (ERP) systems,” *Comput. Human Behav.*, 2004, doi: 10.1016/j.chb.2003.10.004.

[23] Camunda Docs, “Introduction to BPM Platform,” 2019. https://docs.camunda.org/manual/7.12/introduction/ (accessed Nov. 10, 2020).

[24] Camunda API, “Rest API Reference,” 2019. https://docs.camunda.org/manual/7.12/reference/rest/ (accessed Nov. 10, 2020).

[25] España. Ministerio de la Presidencia, “Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.,” *Boletín Of. del Estado. Sec I*, pp. 119778–119857, 2018, doi: BOE-A-2012-5403.