Neuralyzer: A security pattern for the Right to be Forgotten in Big Data

JULIO MORENO, University of Castilla-La Mancha
EDUARDO B. FERNANDEZ, Florida Atlantic University
EDUARDO FERNANDEZ-MEDINA, University of Castilla-La Mancha
MANUEL A. SERRANO, University of Castilla-La Mancha

Abstract. Big Data environments are getting more popular for companies of every field. Big Data allows to obtain valuable information from huge amounts of data. Usually this kind of environments are very complex making them difficult to manage. Furthermore, in the last years there are different regulations about how to analyze data that can affect Big Data systems. One of them, is the right to be forgotten. Despite the opportunities derived from the use of this technology, if it does not comply with those regulations it can have severe consequences for the company. In this paper we propose a pattern to help in the implementation of the right to be forgotten in Big Data enviroments.

Categories and Subject Descriptors: D.2.11 **[Software Engineering]** Software Architectures–Patterns;

General Terms: Design

Additional Key Words and Phrases: Big Data, Security patterns, Right to be forgotten

ACM Reference Format:

Moreno, J., Fernandez, E. B., Fernandez-Medina, E., and Serrano, M.A., 2018. Neuralyzer: A security pattern for the Right to be Forgotten in Big Data, Procs. 25th Pattern Languages of Programs, PLoP'18, October 24-26, Portland, Oregon, USA. 7 pages

1. INTRODUCTION

Big Data is getting more important for companies from different sectors (Akoka, Comyn-Wattiau, & Laoufi, 2017). For all of them, data are essential to conduct their daily activities and to help senior management to achieve business objectives and, as a result, make better decisions based on the information extracted from such data (Mayer-Schönberger & Cukier, 2013). Big Data implies a change compared to traditional techniques in three different ways: the amount of data (volume), the rate of generation and transmission of data (velocity) and types of structured and unstructured data that can handle (variety) (Chen, Mao, & Liu, 2014). This set of characteristics is usually known as the basic three V's of Big Data. Different authors have added new V's to this original set like the value, it is important to highlight that if you do not obtain valuable information from your data, Big Data is meaningless. Furthermore, a Big Data system is usually a very complex ecosystem where different technologies work together like NoSQL databases, cloud computing resources, or software analytics. This heterogeneity makes its management very difficult to perform.

In Big Data systems, the context can significantly affect how the environment will be implemented. One of the things that you must take under consideration is the different regulations that can constrain the data that will be processed and the results that will be obtained. Lately, one regulation that is becoming trendy along with privacy laws in general (specifically with the General Data Protection Regulation (GDPR) driven by the European Union) is: the Right to be Forgotten (RTBF) (Villaronga, Kieseberg, & Li, 2018). The RTBF claims that any user can request that their data be deleted from all your systems. Of course, there are some cases where the data must remain in the system; for example, if the purpose of the data collected is still not reached and the user gave permission for the use of that data. Also there are some scenarios where the data must remain in the system due to legal reasons.

Authors' addresses: Julio Moreno (corresponding author), University of Castilla-La Mancha, Spain, email: Julio.Moreno@uclm.es; Eduardo B. Fernandez), Dept. of Computer and Electrical Eng. and Computer Science, Florida Atlantic University, 777 Glades Rd., Boca Raton, FL33431, USA; email: ed@cse.fau.edu; Eduardo Fernandez-Medina, University of Castilla-La Mancha, Spain, email: Eduardo.FdezMedina@uclm.es; Manuel A. Serrano, University of Castilla-La Mancha, Spain, email: Manuel.Serrano@uclm.es;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission. A preliminary version of this papers was presented in a writers' workshop at the 25th Pattern Languages of Programs, PLoP'18, October 24-26, Portland, Oregon, USA. Copyright 2018 is held by the author(s). ACM 978-1-4503-0107

The problem is that regulators tend to think about a computer like it was a human memory, and in a highly distributed scenario where the data is stored, and replicated, in different nodes from even different clusters and with different storage systems, it is very difficult to properly delete all the personal data from a specific user. There are some general techniques that try to solve this problem like anonymization or pseudo-anonymization, although none of them have been proven to be efficient enough to not affect the performance of the Big Data system. In this paper, we define the creation of a specific pattern to approach this problem. Our intended audience is the Chief Data Officer (CDO) of the company who is in charge of creating the strategy for managing the data and the information of the system, including its governance, control and development of different politics and controls to properly take advantage of the data. Therefore, the implementation decisions should be taken by considering the requirements and context of the Big Data ecosystem.

2. NEURALYZER

2.1 Intent

Describe how to manage the right to be forgotten in Big Data environments. Usually this right is asked by the data subject, which is the person who wants that the system "forgets" his data. However, there are some scenarios where this action should be done automatically by the system.

2.2 Context

Generally, Big Data ecosystems have a huge amount of personal and sensitive data stored in different databases and storage systems, supported by a variety of utility software.

2.3 Problem

The right to be forgotten, also known as the right to erasure, is a feature that suggests that it is necessary to find a way to effectively delete data from storage systems. Erasing an individual's data implies to find and delete all his data; possibly scattered in several places. This functionality gets even more complicated in Big Data contexts where not only personal data is stored, but also information inferred from it due to the use of analysis techniques like business intelligence or machine learning.

2.4 Forces

The solution is constrained by the following forces:

The gap between regulators and technology. There is a huge gap between the good intentions of the regulators and the complexity of real Big Data environments. It is necessary that our pattern meets both necessities.

Flexibility. Big Data ecosystems can be used in different scenarios from a hospital to a factory. Different kinds of contexts require different solutions.

Value. Probably, one of the main characteristics of a Big Data system is the value that can be obtained from this data. Due to the application of the RTBF, this characteristic can be compromised; erasing records may affect the obtained value from the analytics.

Access Control. Due to the importance of the operation, any erasure or anonymization must be performed by order of the CFO: who will manage all the requests made by the data subjects, and who is the only role authorized to perform the data changes.

2.5 Solution

Our solution focuses on an architecture pattern. Hence, it defines an architecture where the data subject demands the deletion of his data from all the databases of the ecosystem. It is important to highlight that the deletion of the data must be authorized by the data officer. Then, an entity called Neuralyzer will delete all the data related to the data subject. In order to do that, it will use different techniques, for example, data masking of the data. These decisions must be taken by considering the context of the system and how it can affect the performance of the analytics. Further details of the solution will be explained in the following subsections.

2.5.1 Structure

Figure 1 depicts the class model for this pattern. The *Subject* and the *Chief Data Officer* represent two interfaces where active entities are authenticated by the *Server* using an *Authenticator Service.* Once they are authenticated they have different objectives. On one hand, the subject makes a request to delete all his personal data from the system. On the other hand, the data officer receives from the server the notification and starts the process to comply with the RTBF. The *Neuralyzer* is the main class in this pattern, its main goal is to decide which technique should be used depending on the context and the data stored from the user. There are three main categories of techniques represented by three different sub-classes: *Anonymization, Pseudo-anonymization,* and *Deletion*. Once the *Storage* is properly modified to comply with the regulation the system should notify the subject that the erasure has been completed. All the operations performed on the data must be recorded in a log file using a *Security Logger/Auditor.*

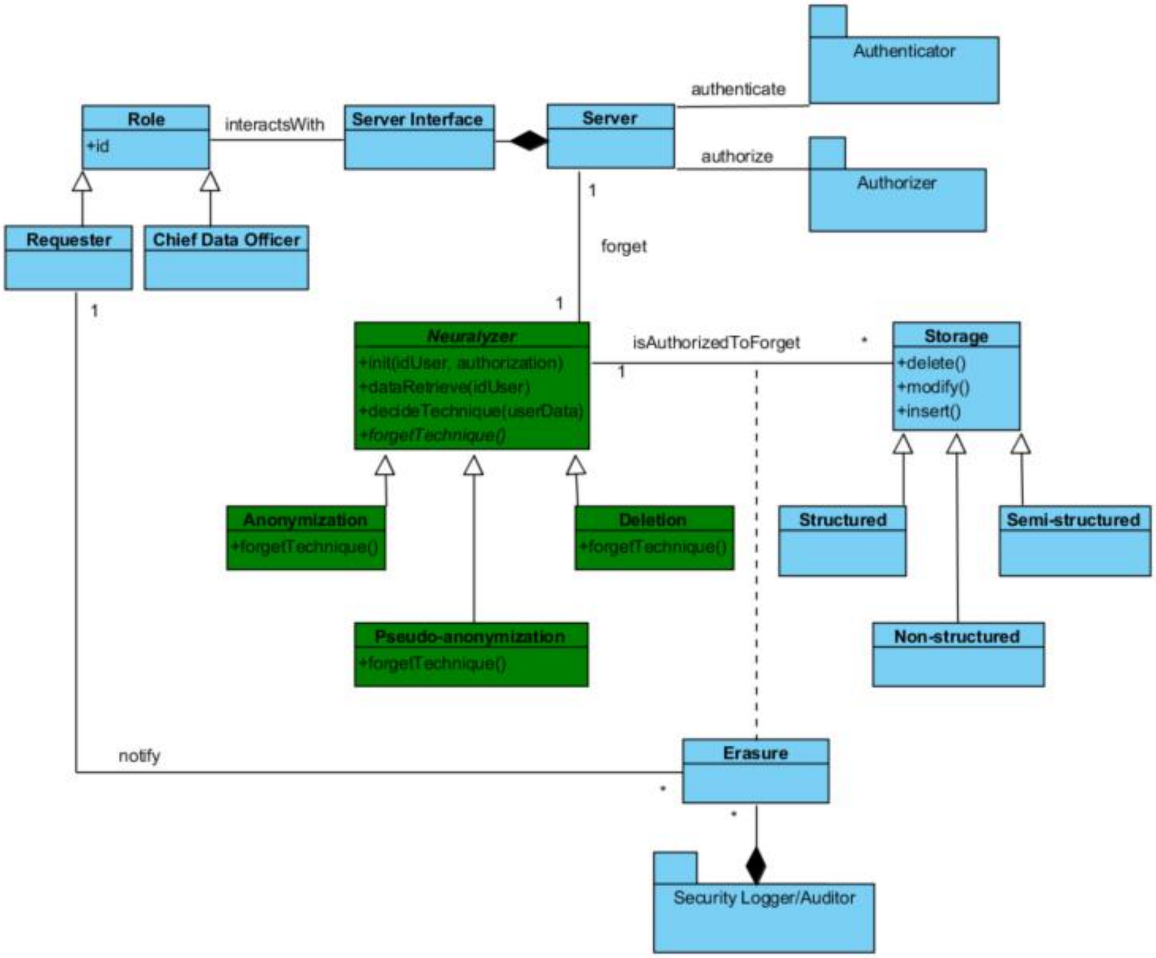

Fig. 1 Class model of the Neuralyzer security pattern

2.5.2 Dynamics

Figure 2 shows the use case “Forget a user”, which corresponds to the sequence:

1. Requester asks for authentication.
2. Requester receives proof of authentication.
3. Requester requests to be forgotten from the system.
4. Chief Data Officer receives the request from the server.
5. Chief Data Officer requests authentication.
6. Chief Data Officer receives proof of authentication.
7. Chief Data Officer initiates the Neuralyzer.
8. Neuralyzer queries the Storage systems to receive all the data from the requester.
9. Neuralyzer receives the data related to the requester.
10. Neuralyzer decides which technique will use depending on the data retrieved and the context of the system.
11. In this case, an anonymization technique is used to forget the data from the user.
12. The anonymization technique is used in the Storage systems.
13. The server notifies the involved users.

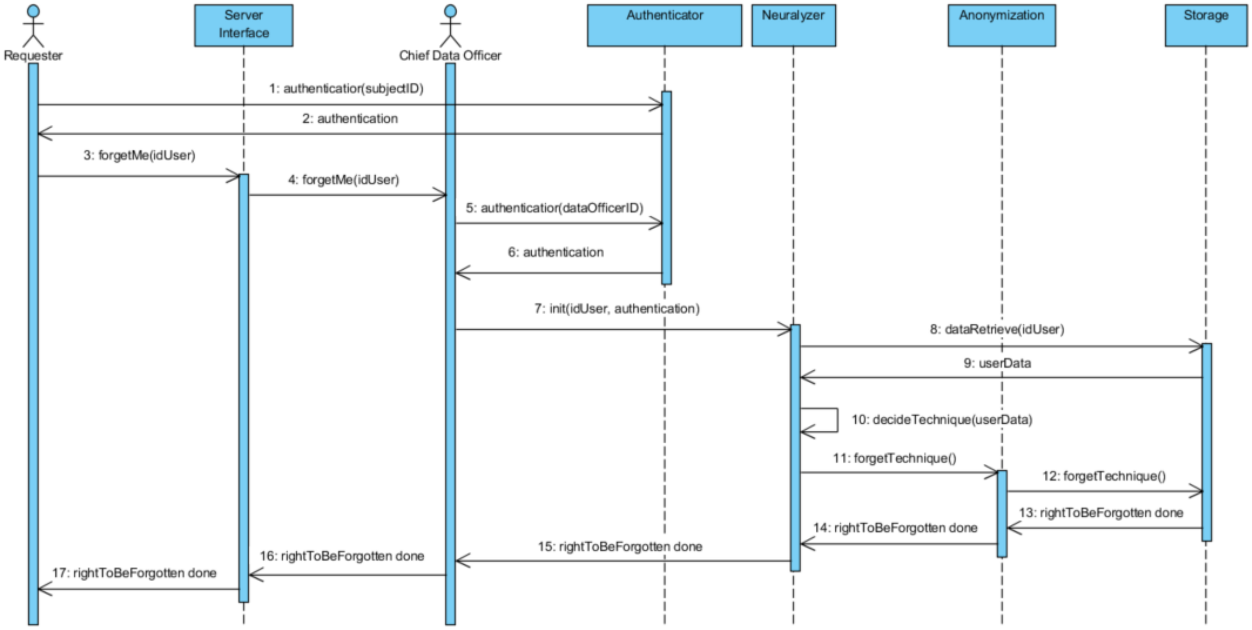
Requester
Server Interface
Chief Data Officer
Authenticator
Neuralyzer
Anonymization
Storage
1: authentication(subjectID)
2: authentication
3: forgetMe(idUser)
4: forgetMe(idUser)
5: authentication(dataOfficerID)
6: authentication
7: init(idUser, authentication)
8: dataRetrieve(idUser)
9: userData
10: decideTechnique(userData)
11: forgetTechnique()
12: forgetTechnique()
13: rightToBeForgotten done
14: rightToBeForgotten done
15: rightToBeForgotten done
16: rightToBeForgotten done
17: rightToBeForgotten done

Fig. 2 Sequence diagram for the use case “Forget a user”

2.6 Implementation

A possible implementation is shown in Figure 3. In this object diagram, we have a scenario with the two roles involved: the requester (who initiates the request to delete their data), and the Chief Data Officer (who is the person in charge to manage the deletion of the data. We have also added the role of data scientist, which is normally present in this type of system, to show that in this particular case it does not have any kind of right over the data. Furthermore, in this scenario, we have a Big Data system that has a storage system based on HDFS (Hadoop Distributed File System). Also, this is a scenario where there is no need to fully eliminates all the data; instead of doing that it is enough to apply anonymization (for example, k-anonymization) or pseudo-anonymization (for example, data masking) techniques on the data. This example tries to highlight that the pattern depends on the context and regulations that can affect the Big Data environment. It is also affected by the implementation features of the system, for example, the use of HDFS.

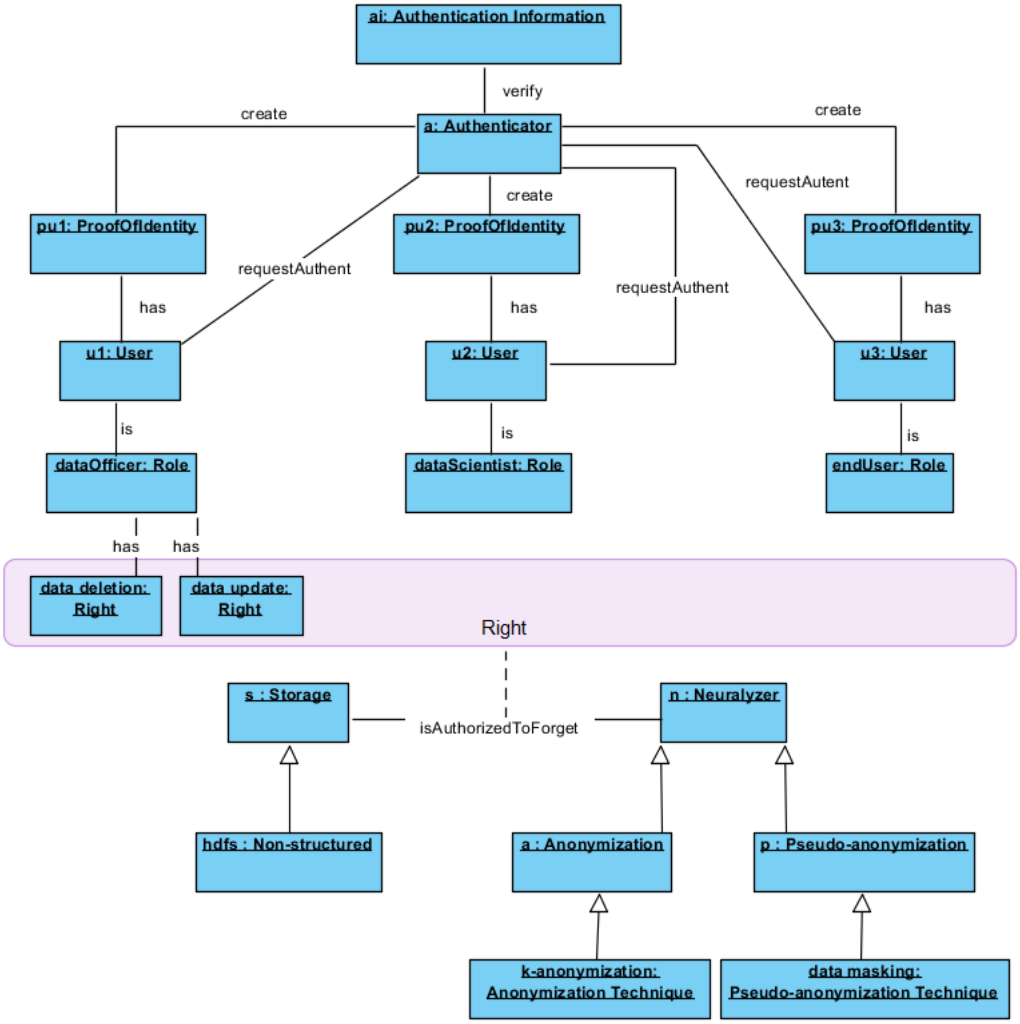

Fig 3. Example of use of the Neuralyzer pattern

2.7 Known uses

- In (Villaronga et al., 2018) the authors explain the main approaches to deal with the right to be forgotten in Artificial Intelligence environments. These solutions can be classified in three categories: anonymization, pseudo-anonymization, and deletion of the data. They also conclude that all these solutions have a relevant impact in the quality of the analysis. Furthermore, in (Mohammed J. Khan, 18AD) the author also highlights that the best way to comply with the right to be forgotten is the anonymization of the data. For that purpose, he shows some examples related to the GDPR regulation. We created an entity named “Neuralyzer” that generalizes the main kinds of solutions used to comply with the right to be forgotten.

- BankingHub (“GDPR deep dive—how to implement the ‘right to be forgotten,’” 2017). Explains a specific scenario where the right to be forgotten is applied in a bank context. In this case, they tag the different types of data stored and depending on that, they assign a retention period for the data; for example, the migration probability and the hobbies of the subject must be deleted immediately, while the account transactions must remain in the system 10 years after the account gets inactive. This is an example of how the data cannot be deleted by demand only, but also, because of the requirements of the context where the Big Data system performs its operations.

2.8 Consequences

Advantages derived from the use of this pattern include the following:

The gap between regulators and technology. The appropriate use of different techniques can reduce this gap. However, it is still important to improve training on these topics for regulators and IT people.

Flexibility. The pattern includes three ways to face the right to be forgotten problem. Hence, it can be adapted to different contexts.

Access Control. We can use RBAC to control actions on the databases and restrict access to only the CDO role.

However, the implementation of this pattern can have some liabilities:

Value. Each time any record of the data being analyzed is deleted, it is highly likely that the results will be affected and vary from the previous time.

2.9 Related patterns

Authenticator (Fernandez 2013). When an active entity like a user or system (subject) identifies itself to the system, how do we verify that the subject intending to access the system is who it says it is? Present some information that is recognized by the system as identifying this subject. After being recognized, the requestor is given some proof that it has been authenticated. This pattern is used to authenticate the Requester and the Chief Data Officer in the system.

Role-Based Access Control Pattern (Fernandez 2013). Describe how to assign rights based on the functions or tasks of users in an environment in which control of access to computing resources is required. The RBAC pattern is needed to establish the rights that the different roles have over the data.

Security Logger/Auditor (Fernandez 2013). How can we keep track of user’s actions in order to determine who did what and when? Log all security-sensitive actions performed by users and provide controlled access to records for Audit purposes. Deleting individual data is a very sensitive operation that must be registered in a log file.

3. CONCLUSIONS

This pattern is intended to assist in the implementation of the Right to be Forgotten in Big Data environments. Different regulations, such as the European Union's GDPR, include this right which may discourage companies from using such systems due to its difficulty to be implemented. This is a consequence of the fact that Big Data ecosystems are often heterogeneously configured, and their primary purpose is to collect and analyze data, it is especially difficult to comply with this regulation. This pattern considers the different storage systems that a Big Data system can have and also the different techniques that can be performed to comply with this limitation. Still, it is important to do more in-depth research on how these techniques can be implemented in different storage systems before this pattern is properly used in real-world scenarios.

ACKNOWLEDGEMENTS

This work was funded by the SEQUOIA project (Ministerio de Economía y Competitividad and the Fondo Europeo de Desarrollo Regional FEDER, TIN2015-63502-C3-1-R) y the project GENESIS (Consejería de Educación, Cultura y Deportes de la Dirección General de Universidades, Investigación e Innovación de la JCCM). We thank our shepherd Sumit Kalra for his valuable suggestions that significantly improved this paper.

REFERENCES

Akoka, J., Comyn-Wattiau, I., & Laoufi, N. (2017). Research on Big Data – A systematic mapping study. SI: New Modeling in Big Data, 54(Part 2), 105–115. https://doi.rg/10.1016/j.csi.2017.01.004

Bankinghub. GDPR deep dive—how to implement the 'right to be forgotten.' (2017, November 15). Retrieved June 5, 2018, from https://www.bankinghub.eu/banking/finance-risk/gdpr-deep-dive-implement-right-forgotten

Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. Mobile Networks and Applications, 19(2), 171–209.

Fernandez, E. B., Security patterns in practice: designing secure architectures using software patterns. John Wiley & Sons, 2013.

Fernandez, E.B., Monge, R., Carvajal, R., Encina, O., Hernandez, J., and Silva, P., "Patterns for Content-Dependent and Context-Enhanced Authorization". Proceedings of 19th European Conference on Pattern Languages of Programs, Irsee, Germany, July 2014.

Mayer-Schönberger, V., & Cukier, K. (2013). Big Data: A Revolution that Will Transform how We Live, Work, and Think. Houghton Mifflin Harcourt.

Mohammed J. Khan, C. (18AD). Big Data Deidentification, Reidentification and Anonymization. Retrieved from https://www.isaca.org/Journal/archives/2018/Volume-1/Pages/big-data-deidentification-reidentification-and-anonymization.aspx

Villaronga, E. F., Kieseberg, P., & Li, T. (2018). Humans forget, machines remember: Artificial intelligence and the Right to Be Forgotten. Computer Law and Security Review, 34(2), 304–313. https://doi.org/10.1016/j.clsr.2017.08.007