Conflict-free Access Rules for Sharing Smart Patient Health Records *

Matthew Banton[0000−0001−8170−3899], Juliana Bowles[0000−0002−5918−9114], Agastya Silvina[0000−0002−0012−9256], and Thais Webber[0000−0002−8091−6021]

School of Computer Science, University of St Andrews
St Andrews KY16 9SX, Scotland, UK
{mb471,jkfb,as362,tcwds}@st-andrews.ac.uk

Abstract. With an increasing trend in personalised healthcare provision across Europe, we need solutions to enable the secure transnational sharing of medical records, establishing granular access rights to personal patient data. Access rules can establish what should be accessible by whom for how long, and comply with collective regulatory frameworks, such as the European General Data Protection Regulation (GDPR). The challenge is to design and implement such systems integrating novel technologies like Blockchain and Data Lake to enhance security and access control. The blockchain module must deal with adequate policies and algorithms to guarantee that no data leaks occur when authorising data retrieval requests. The data lake module tackles the need for an efficient way to retrieve potential granular data from heterogeneous data sources. In this paper, we define a patient-centric authorisation approach, incorporating a structured format for composing access rules that enable secure data retrieval and automatic rules conflict checking.

Keywords: Healthcare systems · Patient Health Records · Blockchain · Data Lake · Access rules.

1 Introduction

Healthcare data systems have evolved from just systems for managing and organising health records to become trustworthy and secure platforms that deal with multiple sources data integration, transformation, and analytics [13]. Their ultimate purpose is to both support organisational decision making as well as medical professionals in clinical decisions, personalised treatments, overall services quality, and efficiency improvement [3].

Patient Health Records (PHR) contain crucial information to enable better clinical decisions such as the patient's medical history, past and ongoing treatments, prescribed medications, exams, and more recently, even data coming from home environment and health tracking technologies. PHR is an essential part in any healthcare data system, however adhering to different storage and access

* This research is funded by the EU H2020 project Serums (Securing Medical Data in Smart Patient-Centric Healthcare Systems), grant code 826278.

control policies for different jurisdictions and organisations, as well as the EU General Data Protection Regulation[1] (GDPR), makes creating a single healthcare platform difficult [21]. Data Lakes are an emergent technology that can aid with these challenges. It can manage the retrieval of diverse medical data, and place it in different repositories, enabling a myriad of strategies for data aggregation, through specialised database queries and processes [17]. However, Data Lakes do not support another crucial requirement of such platform, that being how one can securely provide access to legitimate healthcare providers, with decreased likelihood of data leaks or breaches [24].

This drives the proposal of fine-grained access control strategies within such systems to increase patients control over their own medical data while still establishing the same level of access control practised in healthcare organisations [24]. Recently, blockchain technology has emerged in the healthcare domain as a way to ensure data integrity and increase security and trust in verifiable data sharing transactions, preventing tampering as well as increasing the transparency in communications between patients and healthcare professionals [2,19,24].

The EU project Serums[2] [5,8,20,26] proposes the design of a rule-based authorisation mechanism, blending blockchain and data lake technologies, in a secure patient-centric data sharing platform. The project deals with modern challenges such as the size, complexity and variety of data format present in patient health records, which demand solutions that efficiently unifies these formats into an extensible and flexible standard, and ensures interoperability between data systems placed in different locations.

The Serums Smart Patient Health Record (SPHR) is the unified format proposed to integrate distributed sources of patient information registered in Europe [8,20]. The SPHR contains metadata, linking the patient medical history in a structured way in the data lake, built across authorised healthcare providers and approved health data sources. Based on the metadata, Serums provides an interface for users to create access rules. Thus, users can easily define who (professional or organisation) is allowed to access what (granular medical metadata), from whom (which patient), and when (rule expiration date) through the creation of collective and individual access rules. Conflicting rules may occur checking grantee, expiration date of the rules, overlapping metadata, and the action established by the rules (i.e., grant or deny access). A conflict-free state of the rules set for an individual can be reached using a strategy for conflict detection as well as assumptions to minimise and resolve these conflicts.

This paper presents the pathways in Serums that enable the integration of the scalable Serums data lake tied to a blockchain network to securely retrieve medical data, in a unified manner, and following established access rules for its users. We describe the access rules schema highlighting a structured way to define and validate them within the healthcare system.

The paper is structured as follows. Section 2 brings related work on blockchain for access control in healthcare data sharing systems. Section 3 describes the

[1] Information on GDPR can be found at `https://gdpr-info.eu/`

[2] For more information on Serums project please refer to `www.serums-h2020.org`

Serums platform with focus on the data sharing principles, authorisation mechanism and the pathway to secure SPHR retrieval in the system. Section 4 focus specifically on the access rules design, its structured format, and the subsequent logic-based formalisation. We demonstrate the access rules application through a patient journey, and the expected conflicts that may arise on real-time rules verification. Section 5 concludes this paper highlighting the paper contribution and future work towards enabling a rule-based multinational data sharing platform for healthcare provision in Europe.

2 Related Work

Blockchain allows the creation of transparent and secure user authorisation mechanisms since it can improve access control whilst recording a trail for auditing, especially in case of data breach investigations [19]. A recent survey [24] categorises the strategies to securely share confidential medical records and describes the characteristics of blockchain-based mechanisms employed in several healthcare platforms, so these records can be shared within and across multiple authorised healthcare providers.

We compare Serums to earlier contributions in the literature that specifically exploited the Hyperledger Fabric technology [1] to develop different authorisation mechanisms [2,14,16,18,25] and focus on the design of efficient permissioned blockchains for secure medical data sharing. They are similar to Serums since they also exploit the inherent secure-by-design feature of blockchain to provide tamper-proof logs for transactions over medical records. Moreover, they all construct a particular data retrieval infrastructure with underlying authorisation mechanisms to enable different functions to different user roles.

Serums highlights two essential aspects on access control strategy and patient-centric approach: (i) level of patient control over data and (ii) security measures applied to the access of confidential medical data. We selected two recent contributions in the literature [18,25] to trace a brief comparison with Serums design and their rule-based approach to define users access privileges.

Tanwar, Parekh and Evans (2020) [25] propose an architecture to authenticate and authorise users in a PHR sharing system. Patients register on a blockchain and control their own node, as well as who may access that node (using an algorithm), allowing them to grant and revoke access over the medical data to professionals. Similar to Serums, the architecture follows a patient-centric approach that allows patients to decide about access privileges. For instance, Serums also allows professionals to trigger requests to access the medical records of patients, and patients are responsible to agree or deny them. In both platforms, these requests are controlled and logged by their blockchain.

However, Serums prioritises that medical data is not stored on-chain. While the advantages of storing data on-chain are stated by the authors [25], there are also disadvantages, first being that data cannot be deleted from the chain (which may run into issues with legislation, especially the so called “Right to be forgotten” on GDPR). Secondly, blockchain blocks are typically not large enough

to store the variety of data needed (images such as X-Rays and video files such as Ultrasounds) [18]. Serums stores access rules on-chain, which determine who may access the patient data. This brings many benefits such as providing assurance of who, where and when data is accessed, but also allows users to request their data is deleted in line with their rights in the legislation.

Guo *et al.* [18] maintain the data off-chain, focusing on blockchain to verify the integrity of the data. Additionally, this solution uses customised Access Control Lists (ACLs) which define what users are allowed to do. When receiving a request for a patients PHR, the blockchain accesses the relevant ACL to determine if the user has the relevant permissions. Upon confirmation, the chain releases a single use URL directing the user to the data, as well as a hash of that data. The hash ensures that the data the blockchain is directing to is the same as the data that is eventually retrieved, ensuring integrity.

Serums allows creation of highly granular access rules to medical records by patients and organisations since it introduces the concept of flexible data tags. Similar to [18], Serums data lake component efficiently process data requests based on these tags securely linked to the original data sources. One of the challenges imposed by this feature is that conflicts between rules can arise such as defining different actions over same tags for a pair patient/grantee, when inserting a new rule; and after conflict detection, an action must be taken by the rule creator to resolve it.

Recently, Cui *et al.* [11] developed an example of conflict resolution for Software Defined Networking forwarding rules. They use a three-step process, finding related rules (i.e., any rule with the same source and destination addresses), finding any conflicts (i.e., when the action of two matching rules would be different), and then resolving the conflict. In this related work, conflicts are resolved based upon priority, which is based upon the network function that has generated the rule (security functions having higher priority), as well as the priority of function that generated it. For Serums, the key point is to provide an easy way to patients access their own medical data and update the access privileges given to professionals, especially when they are abroad and seeking to share medical records. Users can create customised access rules to allow professionals to access the medical information. Only that these conflicts may arise when different actions (allow/deny) are defined to the same grantee and set of tags. In this sense, Serums does not use priorities to process rules as [11], but instead requires that users themselves choose the valid rule to be stored in case of detected conflict.

3 Serums Data Sharing Platform Design

Serums platform allow patients to: (i) retrieve their own confidential medical records (i.e., SPHR) containing data from the diverse healthcare providers, as well as (ii) define data access rules to professionals and organisations. Serums should as well enable organisations enrolled in the platform to create and update the patients access rules for their own professionals, in such a way they comply

with the GDPR as well as with their policies and current legislation on medical data sharing for lawful data processing [21].

Serums architecture (Fig. 1) presents the Smart Health Centre System (SHCS) which comprises of a web-based front-end [5] to allow users to retrieve health records (SPHR) [9,10]; as well as a backend with integrated APIs to communicate with each internal module and with external data sources (e.g. hospitals, healthcare organisations, data systems). SPHR metadata (tags) are labels to medical data sources provided by each organisation in the process of their registering to Serums. An SPHR retrieval request triggers a Serums API call to the data lake, which checks the private and permissioned blockchain state [6,20,26] for access privileges (i.e., access rules in place) for the authenticated user.

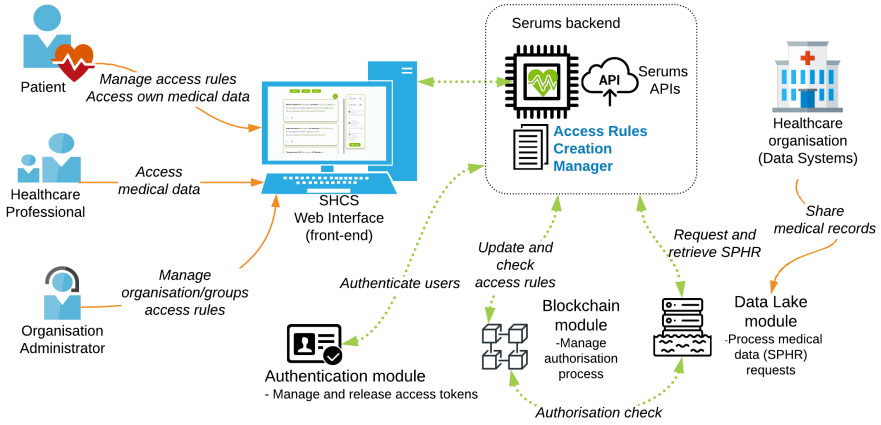

Fig. 1. Serums platform enables custom access rules creation and SPHR retrieval.

A user-friendly interface enables authorised users to easily create and update a set of access rules related to the patient's SPHR, which are secured through the blockchain [5]. Thus, the blockchain contribution to Serums backend is two-fold: first, related to data confidentiality and privacy, blockchain efficiently stores access rules defined by users allowing only authorised individuals to access patients records information; second, the ability to effectively track and audit users interactions within the system.

The customisation of access rules by an individual (patient or admin) assigns permissions to authorised users referring to selected SPHR entries (named data tags), within a specified timeframe. Every time a user attempts to access a patient record (SPHR) in the Serums data lake, the access privileges are checked by the blockchain, and the users can only access the granular SPHR data tags referred to in their own set of access rules. Rules defined by users operate with

an underlying logic-based approach that enable the automatic update of their access permissions over data tags and further conflict detection.

A conflict can be defined as whenever a new access rule, checked against the existent set of rules, would state privileges to the same user but in overlapping time frames, or when it contradicts another access rule in place to a user (i.e., denies it). In Serums, the verification of access rules conflicts follows an algorithmic solution (refer to Sec. 4) that ensures the storage of a conflict-free set of rules on the blockchain after any request of rule update by authorised users.

Blockchain always stores an initial set of rules for the users; for example, a user patient, as the data owner, has access to all tags available for them in the data lake to retrieve. Medical professionals will also have rules in place giving them access to patient data, according to local organisational policies.

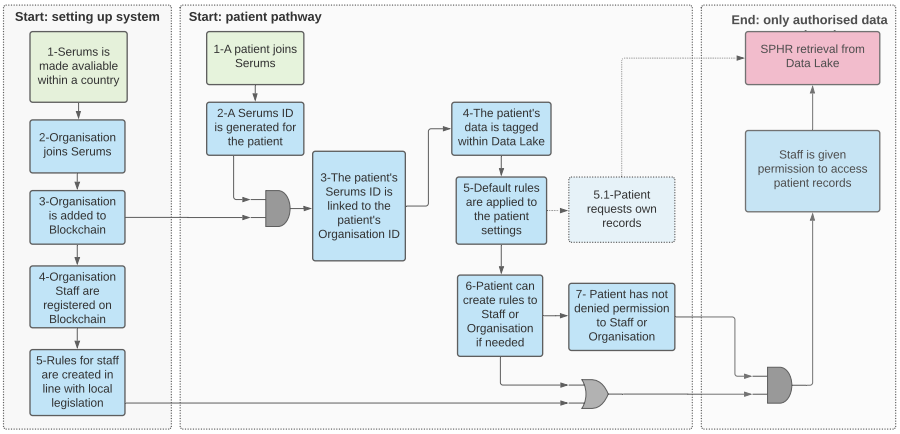

Fig. 2. Steps required for a user to gain access to patients records through Serums.

Fig. 2 outlines a diagram with the steps required for a professional to gain access to patient data. The flow in the diagram can be separated into the organisation pathway and the patient pathway to be enabled in the system. Setting up the system follows five basic sequential steps: from the point Serums must be offered within the European country the healthcare organisation operates, to the point organisation staff are registered on the blockchain. Then, standard access rules are defined in line with local legislation and organisational policies.

These initial steps (on Fig.2) only allow a medical professional to login into the system, but do not define any authorisation to access medical data, or even have the medical data available to upload. Many systems based on permissioned blockchains include a step that imports data from patients into the system [17,25] as part of the organisational setup, assuming the organisation is only going to be using its system for all data management. However, the focus of Serums platform

is to allow professionals to access patient data from other European healthcare organisations, and not to replace their data management system entirely.

The data importation is a part of the patient pathway (see Fig.2). First, when patients join Serums they automatically receive a unique Serums ID. This is separate to the username the patient decides as login detail when joining Serums. Second, the patients Serums ID is then linked to the organisation's Serums ID, the one the patient is joining at (setting up system pathway required). Then, the organisational access rules based on policies and legislation can be applied to the patient, and then patient data uploaded to Serums. This allows professionals (of that organisation) to access the patient data using Serums, as well as allowing the patient to access their own medical data available through the SPHR.

From this step forward, the patient can create new access rules as they see fit. For example, they may allow other organisation staff to access their data even if the default rules would not normally allow it, or they can prevent certain staff from accessing any medical records, should that be something they wish.

Assuming that a medical professional has been given permission to access the records (either from default organisational rules, or from a patient's custom rule) and the patient has not denied them access, then that medical professional can access the patient's data. From this point, it is straightforward to allow another organisation (or particular medical professional) to access their medical data, through the creation of a new custom rule using the professional or the organisation Serums ID.

4 Serums Access Rules Design

Serums users must first login successfully to the system to have access to specific functionalities (i.e., create or update an access rules to a professional, retrieve SPHR data, visualise data analytics, and other functionalities [20]), according to their roles in the system (e.g., patient, administrator, doctor, nurse, etc.).

A Serums user with the appropriate operational privilege (i.e., admin) can manipulate (create/read/update/delete) rules for users within the Serums front-end. Organisational rules created by administrators affect all patient records that pertain to an organisation, considering current legislation to specify the grantees, since patient registration on Serums. Also, patients can directly create access rules to authorised organisations and professionals in the Serums front-end. Access rules are defined with a given *validity* for its persistence in the system. The temporal duration of rules must be explicitly defined, i.e., each rule must specify a *time limit* from rule creation to an expiration date.

Access rules are defined over *tags* that categorise the medical data provided by authorised organisations to the Serums data lake. Serums itself, in this platform version, does not check the medical data provided for appropriate tagging, beyond the basic check performed to ensure data is in the correct format. Serums data lake retrieves all authorised pieces of information according to the rules defined over these data tags. Organisations can always add new tags, whenever they include new systems or applications in their healthcare settings.

The rules are stored in the blockchain right after their creation (or update) takes place, provided no conflict is detected with existent rules, i.e., new access rules are checked against existing ones. A conflict exists where similar rules (from the same grantee and set of tags) establish contrasting privileges (like granting or denying access, overlapping time frames, etc.). If a conflict is detected then the user will be notified and asked to take an action to choose which rule should be stored, or accept an amended rule to ensure there is no conflict. In a proof-of-concept platform version, we propose this format for access rules representation and for automatic conflict detection when creating (or updating) rules, thus users can take action to store only conflict-free rules in the blockchain.

4.1 Serums Access Rules Format

Let Act be a set of actions, Id_S denote a set of identifiers indexed by a sort in S where sorts correspond to granters and grantees, that is, S is a disjoint union where $S = S_G \uplus S_R$. Let T be a set of tags. Following we show examples of considered actions, sorts S_G and S_R, data tags, and rule creators.

Actions : allow, deny
Granters : patient, organisation
Grantees : nurse, doctor, consultant, organisation, department
Tags : consultation, treatment, test, device, medication, personal, chemotherapy, comorbodities, hospitalisation, symptoms
Rule creators : organisation administrator, patient

In particular, we assume that a granter sort can be *patient* or *organisation*, $S_G = \{p, o\}$. Similarly, sorts for grantees are $S_R = \{n, d, c, o\}$. We also note that in this context an organisation can be a hospital, general practice, clinic, etc. The organisation administrators can create the access rules commonly applied to staff in their local systems with patient consent.

Definition 1. *An* access rule *r is a tuple $r = (g, \alpha, R, D, \Gamma)$ where*

- *$g \in Id_G$ is a grantee,*
- *$\alpha \in Act$ is an action,*
- *$R \subseteq Id_R$ is a subset of granters where necessarily $g \notin R$,*
- *$D = (d_1, d_2) \subseteq \mathbb{N} \times \mathbb{N}$ is the time interval indicating when the rule is valid where necessarily $d_1 \leq d_2$, and*
- *$\Gamma \subseteq T$ is a subset of tags.*

We note that even though our implementation uses epoch times to represent dates, it suffices to think about these as natural numbers in this context. The time interval (d_1, d_1) or (d_1, inf) can be used to indicate that a rule is valid forever. In addition, implicit in a rule is the user creating it, so if $g \in Id_p$ then this is a rule created by a patient, and if $g \in Id_o$ then we have a rule

created by the organisation for all patients. One example of a possible rule is $r_1 = (p_1, allow, \{d_1, d_2\}, (t_1, t_2), \{treatment, medication\})$ where patient $p_1 \in Id_p$ allows doctors $d_1, d_2 \in Id_d$ to have access to all 'treatment' and 'medication' records that p_1 received in the time interval (t_1, t_2).

When rules are defined for the same grantee, their combined effect represent the complete access allowed (or denied) over the selected subset of tags.

Assume the complete set of rules to be given by $\mathcal{R}$. A set of rules R for grantee g is correct if and only if there are no rules in R that conflict with each other, that is, $\forall r_1, r_2 \in R, \neg(r_1 \perp r_2)$. Conflict can arise when different actions are placed, for instance simultaneously allowing and denying access over the same data tags and grantee for intersecting time periods. When rules are in conflict, the conflict is highlighted to the user on time of creation, with a request issued by the system for the user to choose which rule should be stored in the blockchain. Whenever possible, system can suggest a conflict-free amended rule.

To check rule consistency automatically and find the set of rules that should be used we can adopt a similar approach to others that have used Satisfiability Modulo Theories (SMT) solvers such as Z3 [22], as well as recommendations to resolve those conflicts [4]. Thus, to help move towards a more user-centric approach, we use a straightforward Z3 coding to identify potential conflicts in rules using our proposed format. We allow the user to select which rule should be applied in case of conflict, defining the next current conflict-free set of rules to be stored in the blockchain.

4.2 Access Rules Application Example

This section explores the access rules creation process within a use case description originated from a patient journey in real-world hospitals in Edinburgh (HE), Barcelona (HB) and Maastrich (HM). A patient journey example includes collection of personal information in several cross-country organisations such as their appointments in GP practices, interactions with professionals, scheduled treatments in hospitals, home care visits, prescribed medications, and the use of a smart device for toxicity data collection [23], just to name a few.

We divide the patient journey description into several points (P_i), and exemplify the creation of access rules and conflicts that can arise from their creation and update in a period of time.

> P1. *A hypothetical breast cancer patient will start chemotherapy at HE, in Scotland. A treatment plan and regimen has been established (this will be over several months with treatment in hospital every three weeks). The patient also has a comorbidity. As any cancer patient on chemotherapy, she might have a higher toxicity level as a result [15,23], but it is important to guarantee that the level does not go above 3. Toxicity levels range from 0 (no toxicity) to 5 (so high it causes death).*

From P1 we can generate a set of rules at organisation (hospital) level, which follows Scottish local regulation, where professionals and staff from the hospital

(HE) will be granted permission to access all data tags concerning the patient, for example. During her first visit to the HE, the patient is registered in the system and a Serums identity is created (refer to the patient pathway in Fig.2). After the patient enrolment, with patient consent, the organisation can create access rules linked to the patient and to the respective professionals.

In the UK, the principle of *implied consent* is one that operates in the process of patient referrals, for instance, from a General Practitioner (GP) to a Specialist within a hospital. This assumes the patient consent to the sharing of personal information, within the National Healthcare System (NHS), at the time the referral is made and for any subsequent treatment relating to the referral. Thus, the organisation can create the following access rule r_1 for the patient based on the local legislation and hospital policies once the patient is registered in Serums.

$$r_1 = (p_1, allow, \{d_1, d_2, n_1\}, (t_1, t_2), T)$$

In this rule example, p_1 is the Serums ID to refer to the patient, and $d_1, d_2 \in Id_d$ and $n_1 \in Id_n$ are doctors and a nurse working at HE; t_1 is the referral date and T denotes all tags. The creator of the rule (in this case, the admin) is explicitly stored on the blockchain component for auditing purposes, however this is not shown in the tuple to simplify the presentation.

The tags provided by the organisation (HE) to be shared as SPHR, for example, are in the set T ={consultation, treatment, test, medication, personal, chemotherapy, comorbidities, hospitalisation, symptoms, device}. As mentioned before, the organisation can also create a set of access rules based on Scottish legislation and compliant with GDPR at time of patient enrolment in Serums. Moreover, we can assume that the patient creates an additional rule r_2 that enables a further doctor $d_3 \in Id_d$ from a different healthcare organisation (her GP) to access the information about her chemotherapy treatment. Her GP is registered as a Serums user by that different organisation, also enrolled in Serums.

$$r_2 = (p1, allow, \{d_3\}, (t1, t2), \{chemotherapy\})$$

It is worth mentioning that Serums allows the creation of rules standing by the same grantee, tags, and grant action but with different (or extended) validity when checked against an existent rule. Once validity expires, the rule is not included in the information retrieval process since blockchain only returns authorised tags of valid rules.

P2. *Patient p_1 aims to give consent to sharing data in between treatment visits via the Cancer Data Gateway and the patient portal. Through a new access rule, she determines who in the medical team sees this information. The oncologist/nurse and her GP.*

In between treatments the patient is sharing symptoms information to both the doctors and nurse at HE and her GP.

$$r_3 = (p1, allow, \{d_1, d_2, n_1, d_3\}, (t1, t2), \{symptoms\})$$

P3. *Via a user-friendly web application with questionnaires provided by the hospital, e.g., the patient can provide information on symptoms daily during her treatment. Serious reported symptoms can be picked up by the clinical team and acted upon immediately.*

P4. *Combined health data can help clinicians adapt treatments better to the patient as an individual which results in controlled toxicity levels and improved health outcomes [23]. It uses data from several patients treated over the years with comparable characteristics.*

From P3, we exemplify that organisations can always provide new data tags to be linked in Serums, e.g., *symptoms*, to include data from this specific system, and from several other in-house applications. In addition, further rules have to be defined to guarantee that oncologists (d_1, d_2), nurse (n_1) and patient's GP (d_3), all have access to any additional important information, as mentioned in P4, where $R = \{d_1, d_2, n_1, d_3\}$.

$$r_4 = (p1, allow, R, (t_1, t_2), \{personal, comorbidities, hospitalisation\})$$

P5. *During the recovery at home between treatments there are signs that toxicity levels are high or that the condition of the patient is deteriorating.*

P6. *One of the members of the clinical team (oncologist, nurse or GP) notices in the system that there are irregularities in the patient's data [23] and phones the patient to intervene.*

P7. *During the phone call a decision is made for the GP/nurse to visit the patient at home and provide some additional medication to alleviate symptoms. Admission to hospital is not necessary. The patient improves. After a few weeks, patient comes to the HE to receive the next chemotherapy treatment.*

None of the points from P5 to P7 require the creation of new access rules. However, these can be steps of vital importance for the patient's improvement, considering the professionals clinical opinion, thus avoiding an unnecessary admission to the hospital. This would be difficult without the right people having access to the right information in a timely manner.

P8. *Patient p_1 has decided to visit her daughter that lives in Barcelona. As she is undergoing chemotherapy and to prevent potential problems, she gets in touch with an oncologist at a hospital in Barcelona (HB) so that he can evaluate her case.* In order to do so, the oncologist needs access to the information on her treatment. Consequently, p_1 creates a new access rule to allow the oncologist to access her information for two days, so he can evaluate the situation.

Thus, from P8, the patient would be creating the following rule with time validity (t3, t4) regarding the HB oncologist:

$$r_5 = (p1, allow, \{o_1\}, (t_3, t_4), \Gamma)$$

with $\Gamma = \{personal, comorbidities, hospitalisation, chemotherapy, medication\}$ and $o_1 \in Id_d$ the oncologist working at HB.

P9. *For unrelated reasons, the patient decides to cancel the trip and creates a new rule to deny the access to the doctor.*

The next rule r_6 is an example of a rule to comply with point P9 revoking access rights to the oncologist from HB. It should be noted that the patient could also update rule r_5 to deny access again, either approach will work, and would have the same end result.

$$r_6 = (p1, deny, \{o_1\}, (t_3, inf), \Gamma)$$

P10. *Let us now imagine that later the patient decides to move to Maastrich, in the Netherlands, and registers at the local hospital (HM).*

The hospital (HM) follows Dutch regulations that establish that only the doctor and nurse responsible for her case can have access to her Dutch records. Thus, this organisation creates rules concerning the local tags they have. In that case, the patient herself can decide if she wishes to share her previous Scottish medical history with additional staff and/or other EU organisations. Through Serums she can create these new rules and allow new clinical staff (not only the ones assigned to her case at HM) to access to her present and previous records.

From P10, we also emphasise how Serums treats new rules that operate in a similar manner to previous rules, i.e., having established the same action but over a different set of tags for a particular pair granter-grantee. For example, consider a patient (p_2) initially allowing a particular doctor (d_1) to access personal detail, chemotherapy, treatment, and tests information. A couple of months after, the patient gives access to the same doctor to personal details, chemotherapy, device information, and tests. It could just be a result of the patient acquiring a health tracking device, or doctor requesting further access, or it could only be the patient forgetting they already have given the doctor access to data, and then giving more (or less) than it is needed. The access rules (r_7, r_8) are as follows:

$$r_7 = (p2, allow, d_1, (t1, t2), \{personal, chemotherapy, treatment, test\})$$

$$r_8 = (p2, allow, d_1, (t1, t2), \{personal, chemotherapy, device, test\})$$

In this case, the system detects a potential conflict, and return an amended possible rule, with no conflict to be stored. The result indicates that the patient is only giving extra permissions to a doctor.

$$r'_8 = (p2, allow, d_1, (t1, t2), \{device\})$$

However, the patient will be notified on the current allowances to be sure that the rules contain the tags set she is willing to allow access to at that moment. Using Boolean algebra, we can see that this effectively mean the particular doctor has the following rule in place:

$$r_8'' = (p2, allow, d_1, (t1, t2), \{personal, chemotherapy, treatment, device, test\})$$

Serums can inform the patient that the doctor have access to the treatment information contained in the conflicting rule (r_7), which was not included in the patients new rule (r_8), and ask for additional confirmation that the amend (r_8'') is what the patient actually desires to share.

This use case illustrated the application of a straightforward format of access rules in different situations that can occur in a patient journey. The logic approach eases the integration of a user-friendly interface for users to define sets of conflict-free access rules to medical records.

5 Conclusion

The core of this work is to explore the requirements for access rules and to experiment on a structured format for representing and checking these rules. The advantage of having this format is to facilitate formal verification of the Serums blockchain-based authorisation mechanism. It enables us to tackle conflict resolution using SMT solvers and constraint solvers, as done in [4,7], for finding, respectively, the optimal treatment plan (in case of conflicts in medical recommendations for patients with multiple chronic conditions) and optimal medication combinations.

We have built a high-level model of data access authorisation. The proposed rules format can support individual (and collective) access rules definition in such a way users can easily define who is allowed to access what (through data tags), from whom (which patient), and when (time boundaries). Further definitions of conflict resolution will be done to take into consideration not only the overlapping tags, but also other important aspects of legislation by country and extended versions of the parameters in the rules. We proposed an initial concept of tags that can be formally expanded as we evaluate further use cases. The rule format also enables us to tackle and conform to important security issues such as access rights to medical data and governing policies.

In future work, we aim the integration of a user-friendly interface in natural language for defining rules, the validation and formal verification [12] of the structures built in the blockchain and data lake modules, as well as coding further real-world use cases.

References

1. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., et al.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the thirteenth EuroSys conference. pp. 1–15 (2018). https://doi.org/10.1145/3190508

2. Azaria, A., Ekblaw, A., Vieira, T., Lippman, A.: Medrec: Using blockchain for medical data access and permission management. In: 2016 2nd International Conference on Open and Big Data (OBD). pp. 25–30. No. 16337137 in OBD, IEEE, New York, NY, USA (2016). https://doi.org/10.1109/OBD.2016.11
3. Bardhan, I.R., Thouin, M.F.: Health information technology and its impact on the quality and cost of healthcare delivery. Decision Support Systems **55**(2), 438–449 (2013). https://doi.org/10.1016/j.dss.2012.10.003
4. Bowles, J., Caminati, M., Cha, S., Mendoza, J.: A framework for automated conflict detection and resolution in medical guidelines. Science of Computer Programming **182**, 42 – 63 (2019). https://doi.org/10.1016/j.scico.2019.07.002
5. Bowles, J., Mendoza-Santana, J., Webber, T.: Interacting with next-generation smart patient-centric healthcare systems. In: UMAP'20 Adjunct: Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization. pp. 192–193. ACM, New York, NY, USA (July 2020). https://doi.org/10.1145/3386392.3399561
6. Bowles, J., Webber, T., Blackledge, E., Vermeulen, A.: A blockchain-based healthcare platform for secure personalised data sharing. Studies in Health Technology and Informatics, Public Health and Informatics **281**, 208–212 (May 2021). https://doi.org/10.3233/SHTI210150
7. Bowles, J.K.F., Caminati, M.B.: Balancing prescriptions with constraint solvers. In: Liò, P., Zuliani, P. (eds.) Automated Reasoning for Systems Biology and Medicine, pp. 243–267. Springer Int. Pub. (2019). https://doi.org/10.1007/978-3-030-17297-8_9
8. Bowles, J.K.F., Mendoza-Santana, J., Vermeulen, A.F., Webber, T., Blackledge, E.: Integrating healthcare data for enhanced citizen-centred care and analytics. Studies in Health Tech. & Inf. **275**, 17–21 (2020). https://doi.org/10.3233/SHTI200686
9. Constantinides, A., Belk, M., Fidas, C., Pitsillides, A.: Design and development of the Serums patient-centric user authentication system. In: UMAP'20 Adjunct: Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization. pp. 201–203. ACM, New York, NY, USA (July 2020). https://doi.org/10.1145/3386392.3399564
10. Constantinides, A., Fidas, C., Belk, M., Pietron, A.M., Han, T., Pitsillides, A.: From hot-spots towards experience-spots: Leveraging on users' sociocultural experiences to enhance security in cued-recall graphical authentication. International Journal of Human-Computer Studies **149**, 102602 (2021). https://doi.org/10.1016/j.ijhcs.2021.102602
11. Cui, J., Zhou, S., Zhong, H., Xu, Y., Sha, K.: Transaction-based flow rule conflict detection and resolution in sdn. In: 2018 27th International Conference on Computer Communication and Networks (ICCCN). pp. 1–9 (2018). https://doi.org/10.1109/ICCCN.2018.8487415
12. David, A., Larsen, K.G., Legay, A., Mikučionis, M., Poulsen, D.B.: UPPAAL SMC tutorial. International Journal on Software Tools for Technology Transfer **17**(4), 397–415 (2015). https://doi.org/https://doi.org/10.1007/s10009-014-0361-y
13. Dhayne, H., Haque, R., Kilany, R., Taher, Y.: In search of big medical data integration solutions - a comprehensive survey. IEEE Access **7**, 91265–91290 (2019). https://doi.org/10.1109/ACCESS.2019.2927491
14. Dubovitskaya, A., Xu, Z., Ryu, S., Schumacher, M., Wang, F.: Secure and trustable electronic medical records sharing using blockchain. In: AMIA annual symposium proceedings. vol. 2017, p. 650. American Medical Informatics Association (2017), https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5977675/

15. Extermann, M., Boler, I., Reich, R.R., Lyman, G.H., Brown, R.H., DeFelice, J., Levine, R.M., Lubiner, E.T., Reyes, P., Schreiber III, F.J., et al.: Predicting the risk of chemotherapy toxicity in older patients: The chemotherapy risk assessment scale for high-age patients (crash) score. Cancer **118**(13), 3377–3386 (2012). https://doi.org/10.1002/cncr.26646
16. Fan, K., Wang, S., Ren, Y., Li, H., Yang, Y.: Medblock: Efficient and secure medical data sharing via blockchain. Journal of medical systems **42**(8), 1–11 (2018). https://doi.org/10.1007/s10916-018-0993-7
17. Gavrilov, G., Vlahu-Gjorgievska, E., Trajkovik, V.: Healthcare data warehouse system supporting cross-border interoperability. Health informatics journal **26**(2), 1321–1332 (2020). https://doi.org/10.1177/1460458219876793
18. Guo, H., Li, W., Nejad, M., Shen, C.C.: Access control for electronic health records with hybrid blockchain-edge architecture. In: 2019 IEEE International Conference on Blockchain (Blockchain). pp. 44–51. IEEE (2019). https://doi.org/10.1109/Blockchain.2019.00015
19. Hölbl, M., Kompara, M., Kamišalić, A., Nemec Zlatolas, L.: A systematic review of the use of blockchain in healthcare. Symmetry **10**(10), 470 (2018). https://doi.org/10.3390/sym10100470
20. Janjic, V., Bowles, J.K.F., Vermeulen, A.F., et al.: The serums tool-chain: Ensuring security and privacy of medical data in smart patient-centric healthcare systems. In: 2019 IEEE Int. Conf. on Big Data. pp. 2726–2735. IEEE, New York, NY, USA (December 2019). https://doi.org/10.1109/BigData47090.2019.9005600
21. Larrucea, X., Moffie, M., Asaf, S., Santamaria, I.: Towards a gdpr compliant way to secure european cross border healthcare industry 4.0. Computer Standards & Interfaces **69**, 103408 (2020). https://doi.org/10.1016/j.csi.2019.103408
22. de Moura, L., Bjørner, N.: Z3: An efficient SMT solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) Tools and Algorithms for the Construction and Analysis of Systems. pp. 337–340. Springer B.H. (2008). https://doi.org/10.1007/978-3-540-78800-3_24
23. Silvina, A., Bowles, J., Hall, P.: On predicting the outcomes of chemotherapy treatments in breast cancer. In: Conference on Artificial Intelligence in Medicine in Europe. pp. 180–190. Springer (2019). https://doi.org/https://doi.org/10.1007/978-3-030-21642-9_24
24. Sookhak, M., Jabbarpour, M.R., Safa, N.S., Yu, F.R.: Blockchain and smart contract for access control in healthcare: A survey, issues and challenges, and open issues. Journal of Network and Computer Applications **178**, 102950 (2021). https://doi.org/10.1016/j.jnca.2020.102950
25. Tanwar, S., Parekh, K., Evans, R.: Blockchain-based electronic healthcare record system for healthcare 4.0 applications. Journal of Information Security and Applications **50**, 102407 (2020). https://doi.org/10.1016/j.jisa.2019.102407
26. Webber, T., Mendoza-Santana, J., Vermeulen, A.F., Bowles, J.K.F.: Designing a patient-centric system for secure exchanges of medical data. In: Int. Conf. on Computational Science and Applications (ICCSA 2020). LNCS, vol. 12254, pp. 598–614. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58817-5_44