The Knowledge Engineering Review, Vol. 35, e6, 1 of 22.
doi:10.1017/S0269888920000089

Enacting policies in digital health: a case for smart legal contracts and distributed ledgers?*

ZORAN MILOSEVIC

Deontik Pty Ltd, 2/10 Buna Street, Brisbane, QLD 4032, Australia
e-mail: zoran@deontik.com

Abstract

This paper presents an approach for the enactment of policies in digital health based on our earlier work on the implementation of digital contracts in distributed systems. A formal policy model and an abstract policy language for the expression of healthcare policies are first proposed, leveraging the semantics of the ISO Reference Model for Open Distributed Processing enterprise language standard. Healthcare consent policies included in the HL7 Fast Health Interoperability Resource (FHIR®) standard are used to illustrate the modelling approach. Several distributed ledger and smart legal contract options were considered next as target platforms for implementation. Their benefits are highlighted along with considerations on their use reflecting business concerns of risk, trust and cost.

1 Introduction

Healthcare policies describe constraints associated with clinical and administrative activities in healthcare, as well as constraints on the collection, exchange and use of healthcare information for different purposes. This paper presents an approach to the problem of enactment of healthcare policies, covering their translation from natural language to a computable format and then using this format to integrate policy expressions across systems and stakeholders involved in digital health. This approach leverages our experience in analyzing healthcare policies and developing their computable expression in the context of national and international e-health interoperability frameworks, which follow the architecture guidelines of the Reference Model for Open Distributed Processing (RM-ODP) (Milosevic & Bond, 2016). We propose a *formal policy model* and an *abstract policy language* that makes use of RM-ODP concepts (Linington *et al.*, 2011), in particular the ISO ODP Enterprise Language (ODP-EL), (ISO ODP-EL, 2015) and earlier research related to the specification of digital contracts (Linington *et al.*, 2004; Andersen *et al.*, 2006).

We then investigate the applicability of some digital contract languages as *concrete policy language* candidates to implement the abstract policy language. Specifically, we focus on digital contract languages developed with distributed system principles in mind (Linington *et al.*, 2004; Berry & Milosevic, 2005), because they could be treated as *smart legal contracts* to be deployed on different distributed ledger technology (DLT) platforms. Smart legal contracts are a new area of inquiry (ISDA Linklaters, 2017) aimed at extending DLT-based *smart contracts* with legal constraints so that they can be legally enforceable with respect to the agreement between the parties involved. A smart contract is a 'computer code that, upon the occurrence of a specified condition or conditions, is capable of running automatically according to pre-specified functions; the code can be stored and processed on a distributed ledger and would write

* This is an extended version of the paper presented at the 3rd Symposium On Distributed Ledger Technology, Griffith University, Gold Coast, Australia (Milosevic 2018)

any resulting change into the distributed ledger' (The Smart Contract Alliances, 2019). There are some smart contract languages, such as Solidity (2019), that facilitate the expression of smart contracts. A concrete policy language can be used to express computable healthcare policies and these policy expressions can then be evaluated in a smart contract platform. The aim is to leverage underlying DLTs to provide an immutable audit trial of actions and to support policy monitoring and enforcement. This enforcement can either be on a discretionary basis, where humans audit and follow-up potential policy violation flags or notifications, or on a non-discretionary basis, for example, through controls on actions in a workflow engine. We use the abstract language to express consent policies of the HL7 Fast Health Interoperability Resource (FHIR®) standard (FHIR, 2019).

Our current focus is on permissioned DLTs, in particular Corda (Brown *et al.*, 2016)[1], as a way of implementing the formal policy model. We also provide general considerations for the use of DLTs and smart legal contracts reflecting business concerns of risk, trust and cost.

The following section provides motivation for this work, highlighting the increasing concerns of consumers and regulators regarding privacy and consent for digital health information, as special type of healthcare policies. Section 3 introduces several healthcare consent policies from the FHIR® consent resource used to test our approach. Section 4 presents key modelling concepts we use to formalize policy. Section 5 shows their use in modelling a privacy consent policy. Section 6 introduces our abstract policy language developed based on the modelling concepts from Section 4 and illustrates its use for several policy consent examples. Section 7 identifies candidate digital contract solutions to implement this language, investigates the use of Corda DLT and provides consideration of their use. Section 8 describes related work. Section 9 provides conclusions and outlines future research.

2 Motivation

This paper is motivated by the need to provide an increasing level of automation of healthcare policy monitoring, enforcement and integration with digital health platforms. This topic is traditionally addressed in the context of information security (Milosevic *et al.*, 1996) but is increasingly considered in higher-level policies that are associated with healthcare delivery, such as informed consent (Wikipedia, 2019; ResearchKit, 2019). Informed consent is a process for obtaining permission before conducting a healthcare intervention on a person or disclosing personal information. The latter is often referred to as privacy consent, that is, defining how individually identifiable health information is to be collected, accessed, used and disclosed.

Clear expression of policies is needed to increase trust among consumers and carers, where the question of access to information can be delegated based on various rules and regulations, including in acute and community care contexts. Policies are an important element in guiding the design and implementation of the next generation of automated systems, in particular AI, to ensure ethical principles are respected when designing, implementing and operating such systems.

Healthcare policies are described in a style similar to a legal contract, specifying obligations of providers or carers in the process of healthcare delivery to individuals, as well as their permissions, authorizations or prohibitions. Similarly, digital health policies define obligations, permissions, prohibitions, authorizations and other rules that apply to the actions and processes associated with the access and use of healthcare information. In general, policies reflect and apply the rules established by applicable professional, regulatory, legislative or organizational contexts, over interactions and processes by participants in some collaborative context (Figure 1).

Our approach to capturing legal semantics for the abstract policy language is by leveraging the formalism of the ISO ODP-EL (ISO ODP-EL, 2015) from the RM-ODP family of standards (Linington *et al.*, 2011), which makes use of formalisms from deontic logic and normative systems while providing explicit support for distributed and federated infrastructures. Further, we consider use of domain-specific

[1] Some references such as Lonsetteig (2017) do not consider Corda as a DLT technology but a shared ledger or even a business-to-business messaging protocol that is inspired by bitcoin. For the purpose of this paper, we do treat Corda as a DLT, as introduced in the Corda white paper (Brown *et al.* 2016).

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

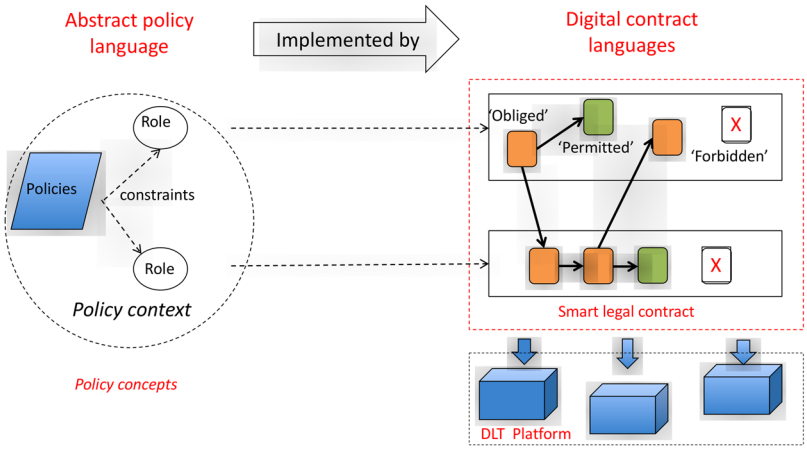

Figure 1 Mapping policy concepts to digital contracts

languages for digital contracts as concrete languages to realize the abstract policy language. This in turn motivates us to consider smart legal contracts and distributed ledgers as a way of enacting policies for certain class of digital health applications (Figure 1).

Some digital contract languages considered inherently support legal semantics, for example, Business Contract Language (BCL) (Linington *et al.*, 2004; Berry & Milosevic, 2005) and logic-based Formal Contract Language (FCL) (Governatori & Milosevic, 2005; Andersen *et al.*, 2006), while others can be extended with legal concepts, for example, Contract Specification Language (CSL) (Andersen *et al.*, 2006) and Digital Asset Modeling Language (DAML) (DAML, 2019). These languages can be mapped to different DLTs and are declarative, as opposed to smart contracts which are typically imperative languages (Governatori *et al.*, 2018). The aim of this paper is to explore these mapping options to enable some features of DLTs to be used in support of the enactment of digital health policies from both the business and technical perspectives.

We further note that some permissioned DLTs architectures, in particular Corda, can provide a suitable platform to directly implement the formal policy model proposed in this paper, while improving privacy, security and legal assurance of digital health applications.

3 Example: consent policies

An important healthcare policy relevant to many healthcare episodes is that of consent. Several definitions related to consent are listed below (FHIR, 2019). These introduce key policy concepts that are discussed in more in detail in the reminder of the paper.

Consent is *the record of a healthcare consumer's policy choices, which permits or denies identified recipient(s) or recipient role(s) to perform one or more actions within a given policy context, for specific purposes and periods of time*. Note the generic nature of this definition: actions can apply to both actions associated with information management and healthcare. Further, the policy choices are typically captured in an agreement referred to as a consent directive, defined as *the legal record of a healthcare consumer's agreement with a party responsible for enforcing the consumer's choices, which permits or denies identified actors or roles to perform actions affecting the consumer within a given context for specific purposes and periods of time*. Various consent-related options are typically specified in a Consent Form, defined as *Human readable consent content describing one or more actions impacting the grantor for which the grantee would be authorized or prohibited from performing*. It includes the terms, rules and conditions pertaining to the authorizations or restrictions, such as effective time, applicability or scope, purposes of use, obligations and prohibitions to which the grantee must comply. Once a Consent Form is 'executed' by means required by policy, such as verbal agreement, wet signature or electronic/digital signature, it becomes a *legally binding* consent directive.

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

There are different types of consent directives in (FHIR, 2019):

- Privacy Consent Directive: Agreement to collect, access, use or disclose (share) information.
- Medical Treatment Consent Directive: Consent to undergo a specific treatment or record of refusal to consent.
- Research Consent Directive: Consent to participate in a research protocol and its information sharing.
- Advance Care Directives: Instructions for potentially needed medical treatment, for example, do-not-resuscitate.

The consent directives above include a set of conditions that, when considered together, constitute elements of a *legally binding contract*. Thus, we suggest the use of contract language formalisms developed elsewhere to model this consent policy. Note several concepts common to the above definitions and typically used in many policy specifications, namely:

- Roles to which policies apply when participating in actions, for example, grantor and grantee.
- Policy constraints, for example, authorization, prohibition, permission and obligation.
- Policy context surrounding roles, actions and policy constraints, for example, purpose of the consent and the legal jurisdiction defining the conditions for legally binding status.

The following section provides a precise description of these policy concepts based on the modelling languages specified in the RM-ODP (ISO ODP-EL, 2015; Linington *et al.*, 2011).

4 Policy concept formalization

The following is a list of key policy modelling concepts from RM-ODP enterprise language (ISO, 2015), selected to illustrate their use in modelling consent policies.

4.1 Policy context

The central part in defining many healthcare policies is the specification of constraints on the actions of the parties who participate in interactions defined by some legislative, jurisdictional or regulative context. This is an important for providing legal clarity as to the situs of interactions, that is, applicable laws to apply to resources, data or interactions under question. For example, under various legal regimes, it is necessary to identify the location of an asset or contract to determine the applicable legal jurisdiction for various legal questions relating to it (ISDA Linklaters, 2017).

Policy context specifies rules that can identify the applicable jurisdictions or organizational policies, all of which govern interactions of the parties. There is thus a need to model wide range of healthcare policy contexts and for that purpose the precise semantics of the RM-ODP concept of *community* can be used to describe the organizational or social environment for the participants as introduced next.

4.1.1 Community concepts

A community defines how a set of participants should behave in order to achieve an objective. To make the rules reusable, a community is defined in terms of interactions between roles in the community and policy constraints that apply to the roles (Linington *et al.*, 2011). A *community role* can be played by a *party*, which models a natural person or legal entity. A role in a community can also be played by another community, making it possible to model hierarchical policy contexts. Note that one can think of community as an organizational pattern for interactions.

Communities are expressed in terms of a type, also known as a *community contract*, that expresses behaviour in terms of a set of participating community roles (Linington *et al.*, 2012). These roles are fulfilled in a particular community by specific *enterprise objects* which are constrained to the behaviour defined for their role. At any point in time, at most one enterprise object can fulfil a role, but an enterprise specification may include a number of roles of the same type, each fulfilled by distinct enterprise objects,

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

possibly with the constraint on the number of roles of that type that can occur, for example, maximum number of patients in a ward. Note that most enterprise objects display behaviour and such an object is referred to as an *active enterprise object*, the special kind of which is *party*, with legal responsibility and accountability, as introduced in Sections 4.2. and 4.3.

A community behaviour can be defined in terms of a process style of behaviour between parties involved. A *process* consists of a number of steps ordered in a sequential or concurrent manner, and each step can involve both an initiator and respondent in the interaction.

4.1.2 Domain and federation community types

One specific type of community of relevance for the definition of policy context is the so-called domain community (or simply a domain). A domain describes a set of objects with a characterizing relationship to some controlling object, for example, a management domain where managed objects of the domain are subject to a policy set defined by management domain object. Formally, *⟨X⟩ Domain* is a set of objects, each of which is related by a characterizing relationship ⟨X⟩ to a controlling object. We can use this concept to express legal or regulative domains for which a particular controlling object, for example, legal or regulatory authority, prescribes a set of policies that define legal or regulative constraints for individual members of that domain. Examples of such policies are obligations, prohibitions or permissions defined by the General Data Protection Regulation (GDPR) authorities and the controlling objects are the so-called Data Controllers. A Data Controller is the natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data in their respective domains (GDPR, 2019). Note that in enterprise terms, policies can be administered by the controlling object over the domain. Domains can be arranged hierarchically, through subdomains, which are subsets of a given domain.

RM-ODP uses the concept of domain to express more complex, cross-organizational interactions and the associated policy constraints to support federation. Formally, *⟨X⟩federation* is defined as a community of ⟨X⟩ domains, formed to meet a shared objective. The capability to express federation is critical for healthcare interoperability in view of the need to manage the combined actions of private and public stakeholders within health sector and across other sectors (Milosevic & Bond, 2016).

4.2 Deontic constraints

There are three fundamental types of policy constraints that reflect rules of any normative system, namely obligations, prohibitions and permissions. Their formal expression is the subject of deontic logic and these are often referred to as deontic constraints.

An *obligation* is a prescription that a particular behaviour is required. An obligation is fulfilled by the occurrence of the prescribed behaviour.

A *permission* is a prescription that a particular behaviour is allowed to occur. A permission is equivalent to there being no obligation for the behaviour not to occur.

A *prohibition* is a prescription that a particular behaviour must not occur. A prohibition is equivalent to there being an obligation for the behaviour not to occur.

The above definitions have been the subject of standard deontic logic, but their application in the realm of enterprise distributed computing requires explicit association with the agent to which these constraints apply. This is also needed to take into account an agent's goal-seeking behaviour, which may result in their willingness to violate the policies with the expected benefit of reward from doing so. This violation and the need to develop a broader view of enterprise behaviour was identified in (Linington, 1998). These developments resulted in a revised version of the ODP-EL standard that provides a pragmatic way of specifying deontic constraints for the purpose of building enterprise distributed systems, as described next.

The way that deontic constraints are associated with the agents (i.e. active enterprise objects in ODP speak) is through the concept of *deontic tokens*. These are enterprise objects which encapsulate deontic constraint assertions. The holding of the deontic tokens by active enterprise objects constrains their behaviour. This modelling approach provides a basis for manipulating deontic tokens, for example,

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

passing them between parties to model delegations, and activation or de-activation of policies that apply to the active enterprise objects in the context of their enterprise interactions. There are three types of deontic tokens that represent deontic constraints. These are called burden, representing an obligation, permit, representing permission and embargo, and representing prohibition.

In the case of a *burden*, an active enterprise object holding the burden must attempt to discharge it either directly by performing the specified behaviour or indirectly by engaging some other object to take possession of the burden and perform the specified behaviour. In the case of *permit*, an active enterprise object holding the permit is able to perform some specified piece of behaviour, while in the case of *embargo*, the object holding the embargo is inhibited from performing the behaviour.

In order to support the dynamics of deontic tokens, the concept of a *speech act* is further introduced. This is a special kind of action that is used to modify the set of tokens held by an active enterprise object. The name was chosen by analogy to the linguistic concept of speech act, which refers to something expressed by an individual that not only presents information but performs an action as well. Thus, a speech act changes the state of the world in terms of the association of deontic tokens with active enterprise objects, such as patient giving permit to a researcher to access their health record. Further, this modelling feature fits well with the nature of AI-enabled digital health applications. It allows the specification of actions that can be performed by both people and AI systems, as will be elaborated in Section 5.4.

In fact, deontic constraints and tokens provide foundations for expressing many types of policy constraints across enterprise objects in a system, including both human actors and automated agents, such as AI systems. ODP-EL provides added formalism to express traceability of obligations of parties, according to their broader responsibilities derived from ethical, social or legal norms. This formalism is referred to as accountability concepts, as described next.

4.3 Accountability concepts

The first concept from the set of ODP-EL accountability concepts is that of *party*, introduced informally earlier. Party is defined as an enterprise object which models a natural person or any other entity considered to have some of the rights, powers and duties of natural person, for example, company. ODP-EL introduces two other concepts which are useful to describe many forms of delegation in enterprise systems. *Principal* is defined as a party that has delegated something (e.g. authorization or provision of service) to another, and *Agent* is defined as an active enterprise object that has been delegated something (e.g. authorization, responsibility of provision of service) by, and acts for, a party (e.g. in exercising the authorization, carrying out responsibility).

Delegation is an action that assigns something (e.g. authorization, responsibility of provision of service) to another object. It is through this mechanism that deontic tokens can be passed across different active enterprise objects, with one example being a delegation from principal to agent, as mentioned above. Delegation is one action type in ODP-EL related to accountability, but there are several other action types to capture important business events in any organizational system, and reflect the dynamics of communication amongst parties, and broadly, active enterprise objects. These action types are listed next (ISO ODP-EL, 2015).

Commitment is defined as an action resulting in an obligation by one or more participants in the act to comply with a rule or perform a contract. This effectively means that they will be assigned a burden. Examples include commitments by clinicians to deliver safe, reliable and effective healthcare to patients.

Declaration is defined as an action by which an object makes facts known in its environment and establishes a new state of affairs in its environment. This can, for example, be performed by an AI system (or a party managing it), for example, informing the interested parties about the result of some analysis.

Evaluation is defined as an action that assesses the value of something. Value can be considered in terms of various variables such as importance, preference, usefulness. In digital health, variables can be performance parameters used, through research applications for example, to either express administrative performance or some accuracy or reliability measures. They can be used to assess the fairness of training

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

data or as part of mechanisms to measure the impact of AI algorithms as part of their explainability requirements.

Prescription is defined as an action that establishes a rule. Prescriptions provide a flexible and powerful mechanism for changing the system's business rules at runtime, enabling dynamic adaptation to respond to business changes and new needs. This ability is important in any digital health system, to establish the applicability of new policies reflecting new legislations for example, or after the adoption of recommendations from AI system components.

Authorization is defined as an action indicating that a particular behaviour shall not be prevented. Unlike a permission, an authorization is an empowerment. In terms of deontic tokens, the enterprise object that has performed authorization will issue a required permit and will itself undertake a burden describing its obligation to facilitate the behaviour. For example, the authorization for the consumer to challenge AI decisions is giving them permit to do so by the AI system (or its creator/manager) who has the burden to do so.

4.4 Implications for smart contracts: legal extensions

The community, deontic and accountability constraints provide the foundation for expressing many legal aspects that characterize various legal instruments, including organizational, regulative or legislative policies. Considering that current smart contract proposals are a form of a basic behaviour, specifying a sequence of events or state changes that reflect agreements between parties (Hyperledger, 2019; Solidity, 2019), deontic and accountability constraints can be superimposed on smart contracts to support legal expressions (Berry & Milosevic, 2005). We refer to such augmented smart contracts as *smart legal contracts*, as introduced in Section 1.

We note however that, although smart legal contracts provide an increasing level of support for legal expressions to be automated and enforced through computer code, the ultimate legal enforcement will require decisions by legal structures. For this reason, a link to the legal prose is likely to be required in foreseeable future, in particular for complex contracts, which is the approach also currently adopted by Corda DLT (Brown *et al.*, 2016).

5 Healthcare privacy consent example

This example illustrates the use of the ODP-EL concepts introduced in the previous section. We begin by specifying a privacy consent community which defines the key roles, enterprise interactions, policies and other entities involved in managing privacy consent for an individual.

5.1 Privacy consent community: stakeholders

Privacy consent is essentially a pattern applicable to many situations and can be formalized as a consent community template. When instantiated, the template can be parameterized with specific variables related to the situation, for example, the specific purpose for which consent was given. This community template specifies the following community role types (see Figure 2):

- *Grantor*, to be fulfilled by any individual giving a consent; an individual is a party that models a person giving consent under a set of permission rules, such as being of legal age. There may be a constraint on a number of role instances that satisfy this role type, because at most one enterprise object can fulfil a community role instance at a point in time.
- *Grantee*, which can be of either a clinician or researcher role type, expected to be fulfilled by professionals with the required credentials, namely:
  - *Clinician*, who has permission to access grantors individual health information for care purposes, covered by the patients consent for primary use, for example, covering access to all of the patient information in an emergency situation, with the associated constraints, such as certain time interval from the emergency event.

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

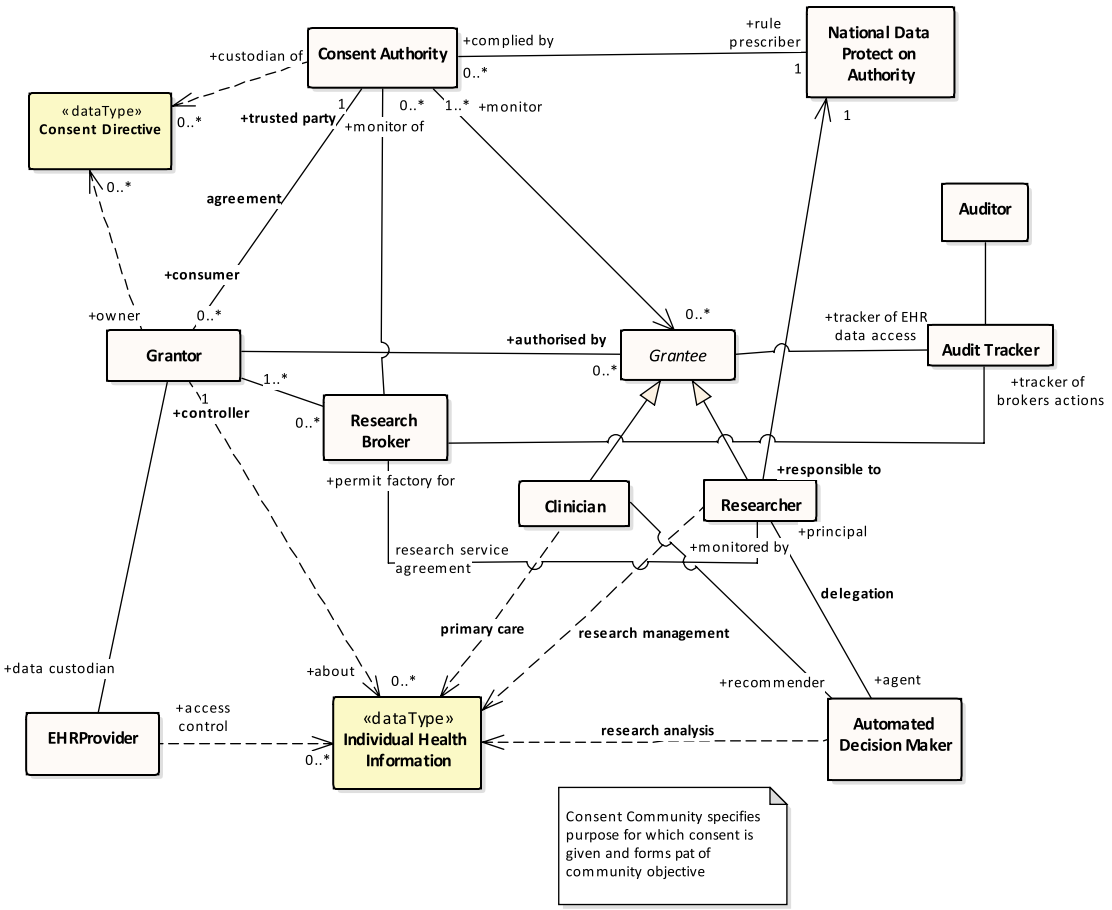

Figure 2 Privacy consent community example

  - *Researcher*, who has permission to access grantors de-identified health information for research purposes; researcher is obliged not to try to re-identify data about patient, as for example may be prescribed through the rules of a National Data Protection Authority; this is however hard to enforce unless all data in/out of their research are monitored, which is difficult to do when a researcher can access third-party data, so some monitoring mechanism may be required, such as tracking of actions of the researcher.
- *Consent Authority* as a trusted party responsible for storing individuals' consent agreements and overseeing the consent agreement rules; a trusted authority can be established by government and operated as a government entity to increase customers' trust in using their information for research purposes, as per their preferences; its function can also be to facilitate ethics approvals to govern the secondary use of data.
- *Research Broker*, a commercial entity authorized to search patient health information and consent data to identify patients suitable for research projects. The Broker might also be responsible for validating/verifying ethics approvals for research projects. The Broker is responsible to ensure that consumer preferences are enforced; thus it is accountable to the Consent Authority, and ultimately to the National Data Protection Authority, represented as a controlling object in this community (in fact, a Domain community, see Section 4).
- *National Data Protection Authority*, which is responsible for defining and enforcing data protection policies, as legislated.
- *Automated Decision-Maker*, performing analytics, recommendations and in some cases, active decision-making; this role guides and augments activities of clinicians, researchers and other stakeholders, such as population health experts; this role can be fulfilled by active enterprise objects, typically clinical decision support systems or AI systems.

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

- *Audit Tracker*, which logs events associated with actions of both clinicians and researchers, in support of building audit trails; this is to enable subsequent analysis of activities, including detecting breaches such as of clinicians accessing healthcare records outside of them providing care, or researchers accessing linked data provided by third parties; both of these are forbidden; in some cases breach detection can be implemented in real time, by a separate role of a *monitor* as we have used extensively in the context of business contracts monitoring (Linington *et al.*, 2004; Berry & Milosevic, 2005).
- *Auditor*, which provides analysis of event traces to support activities such as performance analysis or forensic investigations, to detect breaches and their consequences; this is needed because, in spite of the implemented preventative mechanisms, breaches can happen and these need to be detected, as part of applying subsequent enforcement measures; note that the *enforcer* role was not included in this community model for simplicity.
- *Electronic Health Record (EHR) provider*, who is custodian of individuals' personal health data in their EHR records.

Note that the privacy consent community should specify, as part of its objective, the *purpose* for which the grantor will give consent (see Figure 2), such as cancer research support. We believe that a well-defined ontology capturing different purpose for consent directive may support individuals in better transparency and trust in how their personal data are handled and enforced.

5.2 Privacy consent community: process scenario

There are many interactions and processes in this community, and we use the following scenario to illustrate key enterprise interactions and deontic and accountability concepts. The following are normal sequences of activities involved from the time an individual gives consent to the point their data are used by researchers to inform and contribute to the new medical treatments.

1. Grantor updates their consent directive at the Consent Authority to indicate that they are willing to disclose de-identified genomic data for cancer research, as part of their individually identifiable health data, but excluding data related to mental health. This requirement is in line with the recent GDPR directions to better support personalized and fine-grained controls over access to individual information.
2. Grantor permits matching on their de-identified health data by a Research Broker, allowing the Broker to discover accessible patient data for research matching purposes. Essentially, this is permission for the Research Broker to search the data and consent directives of patients and retrieve identifiers for patients whose consent matches the research requirements. It does not give them access to the health data, just search and retrieve identifiers.
3. Researcher, on some event for example, award of grant funding, would like to access health records of all patients who have consented to participate in cancer research. To this end, they contact the Research Broker stating that they are interested in conducting this research across all patients who have given consent for cancer research; this includes access to their medications, treatment and genomic information; the researcher also provides evidence of ethical and clinical approvals for the research.
4. Research Broker provides a de-identified list of eligible patients to the researcher and an authorization for them to access de-identified patient data from the EHR provider, but highlights the potential exclusion of medication data related to mental health treatment. The authorization requires the researcher to maintain an audit trail of all access to the data (i.e. obligation). Note that the list might be encrypted using the public key of the EHR provider, meaning that the researcher cannot access patient identifiers.
5. Researcher retrieves de-identified patient data from the EHR provider, as per the authorization supplied by the Research Broker. The EHR provider filters the data as required to comply with individual patient consent directives and lodges an audit record relating to the released data with the Auditor.

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

Note that the EHR provider will usually be prohibited by law from releasing patient data without consent, except when a clinician is providing care for the patient in an emergency setting. As such, the Broker cannot be responsible for retrieving/releasing the data unless the Broker is an agent for the EHR provider, appointed under agreed indemnity conditions. The EHR provider would still be subject to reasonable care provisions in their choice of a Broker.

6. Researcher accesses the EHR data to support their research, lodging an audit record for each access with the Audit Tracker. This includes access by an AI system in the role of Automated Decision-Maker, as an agent acting on behalf of the researcher using the authorization from the researcher (e.g. a research permit token).
7. Researcher publishes the result of the research and informs all relevant parties.
8. At a later point, there is a suspicion by a patient that some of their mental health data were used by a health insurer and patient then contacts the Consent Authority to lodge a complaint.
9. Consent Authority then engages Auditor who accesses audit trail produced by the audit tracking system to perform forensic investigation of patients data access by Grantees. Upon detection of specific violation, it notifies an enforcer role to apply penalty to either of the parties. Note that this role is not depicted in the diagram, and it can be fulfilled by a separate community, that is, enforcer community, with a number of constituent roles, including court. Further details about various roles in this community can be found in our earlier work (Dimitrakos *et al.*, 2003).

The above is a *normal* sequence of activities, in which each participant acts according to the expected behaviour but the privacy protection mechanisms need to be established to cover a range of possible breaches that can be envisioned at design time, that could include:

1. Broker incorrectly identifies the nature of the research, leading to inappropriate release of information.
2. EHR provider failed to apply access controls correctly, in spite of Research Broker correct implementation of consent preferences in the access token.
3. Broker does not adequately vet the research approvals, leading to inappropriate release of information for example to an insurance company.

5.3 Privacy consent community: deontic constrains

The privacy consent community defines a number of deontic constraints that apply to the active enterprise objects, including:

- Permission of the Grantor given to the Consent Authority to store consent agreements, for example, valid for a specified time period defined by the Grantor.
- Permission of the Grantor to the Broker to search patients data and if it satisfies researcher criteria include a link to this data in a data set for the researcher.
- Obligation on the Audit Tracker to log data access by the Grantee reliably and on-time and provide access to the audit trail by the Auditor; the tracker may also have an obligation to log actions of Research Broker which may be needed for forensic purpose.
- Authorization of the Grantor to the Grantee to access the Grantor's individual health information; this is realized through this chain of authorization:
  - Grantor issues permit to the Research Broker for searching their data to establish whether they satisfy research question criteria.
  - Research Broker issues a research permit to the researcher which includes a list of Grantors that provided consent to access their de-identified health data and whose data satisfy the research question.
  - EHR provider provides access permit to the researcher to access health records of specific patients, provided researcher satisfies KYC condition, that is, it has credentials requested by the EHR provider; the possession of such permit is an example of ODP conditional action that applies to the researcher.

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

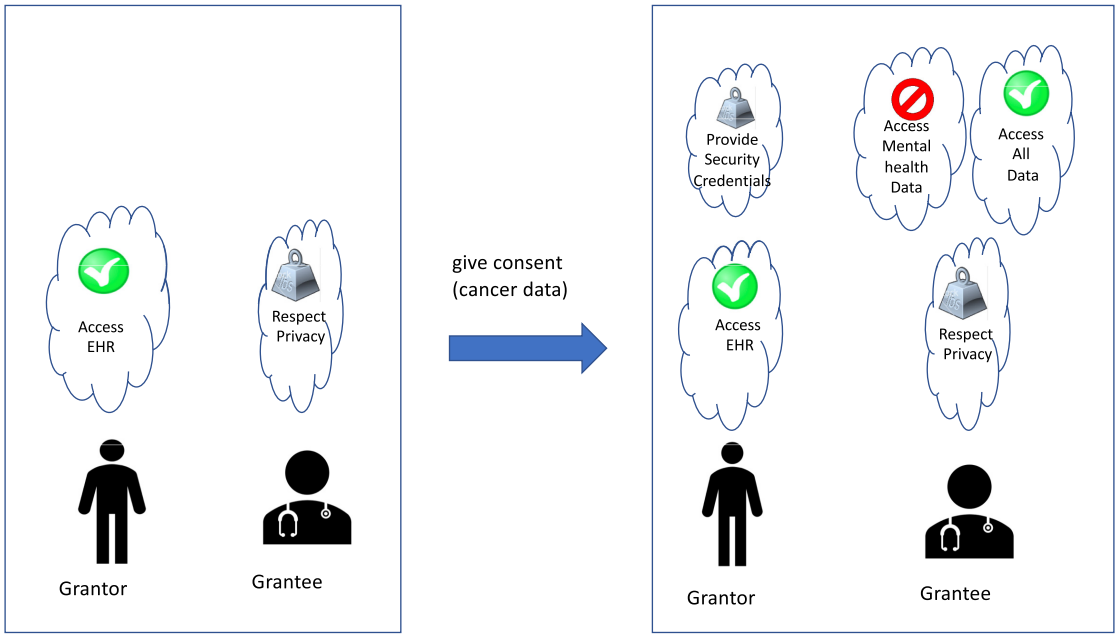

Figure 3 Deontic token manipulation – the dynamics of authorization

- Authorization of the researcher to the Automated Decision-Maker to perform necessary calculations on personal health data, after obtaining authorization by the Research Broker; this may be machine learning over a set of medications to predict its impact in terms of the percentage of patients with full recovery within certain time period that can be considered by Clinician, as an ultimate, legally accountable decision-maker.

5.4 Privacy consent community: accountability concepts

The machinery of deontic tokens can be exploited to define, implement, monitor and enforce various policy aspects associated with managing consent. As noted in Section 4.3, authorization is modelled using a combination of permit and burden deontic tokens. For example, authorization of the Grantor to the Broker involves both the permit being passed from the Grantor to the Broker to search its record but also places an obligation on the Grantor itself, through the corresponding burden, to ensure that access to its record is ultimately enabled.

This authorization action is also a speech act because it changes the deontic state of both the grantor and grantee. The effect of this speech act is that the existing grantor's permit to the Broker to search its healthcare data is passed on to the grantee, as shown in Figure 3. In this example, we assume that the consent directive gives permission to the researcher to access the Grantors health data but prohibits access to the Grantor's mental health data (if it exists). The use of speech acts and deontic tokens is a convenient means for describing the dynamics of deontic constraints and passing of tokens, including to the parties with ultimate legal responsibility.

Many data protection rules defined by a National Data Protection Authority set accountability and legal responsibility expectations for actions of researchers involved in using grantor's data. These data protection rules were established through *prescription* actions (see Section 4.3), performed by the National Data Protection Authority, which essentially establishes obligations and permissions for all the parties involved in accessing patient data.

Further, one needs to precisely specify the distinction between actions of autonomous decision-makers (e.g. Clinical Decision Systems and AI systems) and carers, because typically the latter can be legally

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

responsible in a particular jurisdiction[2]. This distinction is shown in the example (see Figure 2) whereby the researcher acting as a principal can be thought of as 'delegating' machine learning activities to an AI system as an agent in this relationship[3]. Note that researchers have their own responsibility to the National Data Protection Authority, to comply with general data protection principles.

6 Abstract policy language

This section outlines how policy concepts described in Section 4 are used to form part of an abstract policy language. The language use is illustrated with several consent policy examples.

The first element of our policy language is the concept of *policy context*. We use the ODP concept of community to model context, as introduced in Section 4.

The second element is further refinement of the *deontic constraints*, in terms of the behavioural modelling concepts listed next (Linington *et al.*, 2004):

- *policy activation trigger* that specifies triggering conditions which signify that normative policies are in force; these can be temporal events or other events, such as violation of other policies; this provides support for dynamic activation of policies triggered by various conditions such as timeouts that activate the policy in question (e.g. contrary to duty deontic logic constraint); noting certain accountability actions can also signify policy activations, such as prescriptions, delegations and declarations, as introduced earlier;
- a *community role*, to which modality and behavioural constraints apply (defined by the community context), thus defining deontic constraints for the role;
- *deontic modality* that applies to the party fulfilling a community role, for example, an obligation, permission or prohibition; a deontic modality can explicitly identify a target role in a modality expression;
- *event pattern*, specifying the expected behaviour of a party in terms of their actions and other occurrences such as expiration of deadlines or actions of other parties; detailed descriptions of different types of event patterns are beyond the scope of this paper and can be found in Linington *et al.* (2004) and Berry & Milosevic (2013);
- *target role*, which can be used to specify the applicability of modality constraints of the actions of the subject roles that affects other community roles;
- *violation* conditions specifying other policies that can be triggered in response to a violation of the primary deontic modality; this allows linking of the primary policies and those activated when violations occur.

Consequently, a high-level expression of a general policy constraint is of the following form: ⟨*communityContext*⟩ ⟨*policyActivation*⟩ ⟨*role*⟩ ⟨*modality*⟩ ⟨*eventpattern*⟩ ⟨*targetrole*⟩ ⟨*violation*⟩

6.1 *Example 1: privacy consent policy*

Privacy consent policy would thus look like:

⟨*ConsentContext*⟩ ⟨*consentCreation*⟩ ⟨*grantor*⟩ ⟨*permission*⟩ ⟨*accesPatientInfo*⟩ ⟨*grantee*⟩ ⟨*violation*⟩

In the above, accessPatentInfo specifies an event pattern, such as the period for which the consent was given and its purpose, for example, access to a specific IT or physical resource (not included in the policy expression for simplicity). This general consent statement can be instantiated for a specific consent policy instance. Consider a simple example for a personally controlled EHR:

[2] In some cases, torts laws or contractual mechanisms could come into play, complicating legal responsibility questions.

[3] There are many legal questions to be addressed regarding the responsibility of autonomous decision-makers, as demonstrated by the recent Google autonomous car case (The Conversation, 2019), as well as the specification of the responsibilities of the parties involved in the design and deployment of AI systems.

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

'A consumer Bob grants permission to an emergency clinician to access his EHR record, in case of emergency.'

'*⟨EDcare⟩ ⟨emergencyPresentation⟩ ⟨Bob⟩ ⟨permission⟩ ⟨accesEHRRecord⟩ ⟨accreditedEmergency Clinician⟩ ⟨⟩*

This policy is activated by *emergencyPresentation* event, selected from a set of possible triggering events that can be defined at an organizational, state or national level, possibly as part of a personal health record. The policy assumes the existence of patient identifier framework, for example, Individual Health Identifier in Australia, which would identify the patient Bob in this case.

6.2 Example 2: advance care directive

Another example is advance care directive policy that authorizes a substitute decision-maker, that is, a person permitted under the law to make decisions on behalf of someone who does not have capacity, namely:

⟨AdvancedCareContext⟩ ⟨AdvancedCareDirectiveCreation⟩ ⟨Grantor⟩ ⟨authorize⟩ ⟨Medical Decision⟩ ⟨SubstituteDecisionMaker⟩

⟨Legislation⟩ ⟨obligation⟩ ⟨Grantor⟩ ⟨actResponsibly⟩ ⟨violationCondition⟩

The example illustrates the use of a care policy and include authorization for the Grantee to make a medical decision for the Grantor. The example also illustrates one obligation that applies to the Grantor, as defined by the other community referred to as Legislation, as well as a number of violation conditions, that may either activate some other policies or generate alarms that may involve human decision-makers, with potential escalations actions.

6.3 Example 3: research consent directive

This example is applicable in cases where an individual wishes to give permission to a research organization to access their data, typically de-identified, for specific research purposes. This is one of the number of policies that was discussed in more in detail in Section 5.

⟨ReserachConsentContext⟩ ⟨ResearchConsentDirectiveCreation⟩ ⟨Patient⟩ ⟨authorize⟩ ⟨usePatient Data⟩ ⟨ResearchOrg⟩

⟨ResearchOrg⟩ ⟨obligation⟩ ⟨payForDataAccess⟩ ⟨Patient⟩ ⟨violationConditions⟩

For simplicity, we do not specify the purpose element of the policy expression but we note that audit trails can be used to infer the purpose based on the trace of events.

7 Concrete policy languages: smart legal contracts and DLT considerations

There are different options for enacting policies in digital health based on the formal models proposed. The preferred options will depend on several business and technical issues. Business concerns include *risks* associated with violation of agreements, such as with violation of consent conditions and associated *costs*, as well as *trust in others*, that is, the level of confidence that another party will act as expected. Technical concerns include security protections, scalability, technical infrastructure and interfaces between the existing and DLT-specific components. These concerns will determine the choice of monitoring and enforcement options (Berry & Milosevic, 2005).

This section presents some concrete policy language solution options with particular consideration for the role of smart legal contracts and distributed ledgers. We investigate the use of BCL and CSL as targets for the abstract policy language. These languages, when considered in the context of distributed ledger platforms, leverage the existing efforts to support secure, reliable and private interactions among parties. Note however that these domain-specific languages were developed prior to the emergence of DLTs and can be used independently of underlying DLT platforms.

We also investigate the applicability of the Corda enterprise distributed ledger as a way of implementing community modelling concepts resulting from initial considerations related to the consent community. This would leverage the strong privacy, security and immutability properties of the platform.

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

7.1 Business Contract Language (BCL)

BCL semantics (Linington *et al.*, 2004) is based on the formally defined concepts from the RM-ODP standard, particularly the ODP-EL concepts, introduced in Section 4 and thus it has a similar style of expression as abstract policy language, due to their common ODP underpinnings. BCL includes the concept of *community template*, serving as a context for the definition of roles, which specify expected behaviour of parties, including the applicable *deontic constraints*. BCL uses *event patterns* to specify triggering, behavioural and violation conditions for the policy language[4]. BCL supports the concept of *state*, defining the data of relevance for the contract and which may be required to evaluate policies. BCL back-end components are implemented in Java and use contemporary software to implement interfaces, including Web-based technologies.

BCL has been formally analyzed in terms of its deontic logic foundations and this led to the development of the FCL (Governatori & Milosevic, 2005). In FCL, a legal contract is expressed in terms of deontic modalities such as obligations, permissions and prohibitions, with additional support for the expression of and reasoning about violations of such deontic modalities and the subsequent actions that need to be taken to deal with violations. FCL has a well-developed mapping to BCL and it continues to serve as a logic-based counterpart to BCL, which can be used to detect inconsistencies in contracts such as contract gaps or duplicate statements.

7.1.1 Monitoring with BCL

In many digital health systems, it is valuable to use policy expressions to facilitate out-of-band real-time monitoring of activities of the parties against policy rules. This provides many benefits, such as faster reaction to important events that might signify occurrence of medical conditions requiring action or detecting potential breaches of policies. This is typically done by a trusted third party which can take the role of a monitor. Once the monitor detects a contract breach, it can invoke various discretionary or non-discretionary enforcement options. This can but does not have to involve smart contracts, and depending on trust one may use public or private ledgers.

BCL has been proven in many research projects to support real-time monitoring of business contract conditions (Linington *et al.*, 2004). It can thus be used to specify and implement monitoring and enforcement of healthcare policies. Consider for example the privacy consent community introduced earlier. The snippet of the BCL below shows how the consent for cancer research can be represented:

```
CommunityTemplate: CancerResearch
ActivationSpecification: IndividualConsentDirectiveSigned
Policy: PrivacyConsentResearch
Role: Individual
Modality: Permission
TargetRole: accreditedResearcher
Condition: On CancerResearchStart [NOT MentalData]accessEHRRecord
```

The above snippet uses the guard over the information content of EHRRecord data to ensure that access to mental health data from the patient personal health is not possible. Another option would be to specify a prohibition policy over the same data, with the same effect.

BCL constructs can be used to support legal extensions over relevant smart contract languages. Our current efforts in developing such extensions for Corda contracts are promising, due to the commonality in the semantics related to the concepts of party, state and interaction and the use of similar Java style of their expression.

7.1.2 Monitoring and enforcing with BCL

BCL is well suited to be linked with sophisticated choreography solutions to provide added functionality to support contract monitoring and various flavours of contract *enforcement*. In our earlier work, we

[4] Note that a similar event pattern language to that of the BCL event pattern language was successfully used in supporting real-time analytics solutions (Berry & Milosevic, 2013)

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

showed how it is possible to add BCL constraints to a choreography language and engine, called Finesse, supporting asynchronous implementation of processes without centralized control (Berry & Milosevic, 2005). This effectively means the ability to embed legal-style expressions within many distributed infrastructures. This research was performed prior to the emergence of DLT technologies, but much of the solution elements are applicable to distributed ledgers, owing to the distributed semantic underpinnings of BCL and Finesse.

For example, the execution of predefined sequence of activities introduced in the example earlier can be implemented using Finesse (as it can also be implemented using Corda, see Section 7.3). Finesse choreography engine can provide event-based communications and interactions between the community roles, and behaviour of these roles can be further translated into key Finesse constructs to ensure distributed execution of their interactions. One of the benefits of Finesse is that it can describe the behaviour of a distributed infrastructure, protocol or application, including DLT concepts. A good example for this is the Audit Tracker community role. This role can be realized as a Finesse role in this choreography solution, implementing the events associated with auditing tasks through capturing all visible behaviour of required participants, such as clinicians or researchers. We note that the distributed auditing functionality can also be implemented directly, using appropriate DLT platforms that provide immutable audit trails, such as for example possible with Corda (see Section 7.3).

7.2 Contract Specification Language (CSL)

Another candidate for mapping of the abstract policy language is CSL (Andersen *et al.*, 2006), a functional language for specifying contracts in terms of event composition, where events can be specified sequentially or concurrently, in similar manner as with BCL event patterns mentioned in Section 7.1. Formally, in CSL, a contract denotes a set of traces of events, each of which is an alternative way of concluding the contract successfully, in a style similar to communicating sequential processes denotational semantics (Andersen *et al.*, 2006). Further, its denotational semantics provided input into the development of operational semantics, where a contract is executed by interpreting events that arrive to the observer, and if matched, process them, followed by the similar analysis for the remaining events, which are regarded as remaining contractual commitments, or residual contract.

CSL does not have explicit language support for expressing deontic constraints, but can be extended, as per our analysis carried out with Deon, Digital, a Swiss company further developing CSL as a contract language agnostic of any DLT. We believe that it is of particular value that CSL provides DLT adapters for R3 Corda and IBM Hyperledger DLTs, so that it can be used as a smart contract language for those two DLTs (Deon Digital, 2019).

7.3 Corda distributed ledger option

The consent community model (see Section 5) introduced a number of peer-to-peer enterprise interactions between individual entities involved in different steps of the scenario. The first step, for example, involves a patient (in Grantor role) and Consent Authority, which manages consent directives of patients. One of the requirements in any instance of the consent community is to ensure privacy and non-repudiation of this interaction, so that only these two peers can share information about this particular transaction and have consistent view about it. In addition, the consent directive can be updated and removed from the consent directory, and again, it is only these two parties that can verify to this fact. Further, there is a requirement to record history of consent updates, thus producing an audit trail of these changes that may need to be accessed by a third-party Auditor role. Similarly, many of the other enterprise interactions in the scenario and the data involved have the requirements of privacy and consistent view on the shared data involved in their interactions. In addition, each of the pairwise interactions between parties forms a step in a larger process in the community that describes the overall scenario.

Much of this functionality can be supported using the concepts of Corda DLT (Brown *et al.*, 2016). Corda is a DLT initially developed for the needs of finance industry to facilitate collaboration between potentially distrusting parties while supporting them to maintain consensus about shared facts associated

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

with their interactions. Corda is a permissioned DLT, whereby parties join the Corda network, after their identity is checked (KYC principle), similar like the checks needed for an enterprise object to join a community role. Corda supports direct, secure (TLS encrypted), communication between parties on a peer-to-peer basis, as opposed to the gossip style of consensus adopted by Bitcon or Etherium. This allows communication on a 'need to know' basis, so while the patient need to communicate for example with the Consent Authority or clinician, they do not need to communicate with researcher or Auditor.

The consensus in Corda is achieved by both parties in the interaction verifying state updates, where state captures a shared fact of interest for the parties, for example, update of consent directive. This verification involves first the evaluation of a contract which specifies valid business transitions between states, after which both parties need to digitally sign the new state, making old state part of the chain of state history. This mechanism ensures consensus between the parties but also ensures non-repudiation, provenance and immutability of records and thus facilitates subsequent audit trail as required.

Finally, Corda supports definition of process style of behaviour through the concept of flow. In fact, the flow, contract and states represent elements for building a business logic or interactions between parties, and these are combined into CorDaps (Corda Distributed Applications).

In Corda network, each Corda Node runs Corda software plus CorDapps, which are used by parties that host Corda nodes.

These Corda components can provide good foundations for overlaying the features of ODP concept of community in general, especially when a community needs to support secure and private communications between parties while ensuring their consensus as a way of increasing trust within the community. This is indeed the case with the consent community and the following points illustrate how Corda could be used to implement elements of the consent community.

- Each of the parties in the community can deploy its own Corda *node*, forming together a Corda *network*, with a party representing a patient being able to access the network via appropriate CorDapps UI, perhaps offered by an agent who is part of Corda network.
- Corda *flow* can be used to specify distributed choreography that implements interactions between parties in each of the steps in our scenario and communicates through passing messages between the nodes.
- Corda *states* can be used to describe shared facts between the parties concerned about data involved in enterprise interactions. Examples are (a) consent directive between the grantor and Consent Authority, (b) EHR record entries (c) EHR individual attributes when the focus is on specific data to be access within EHR record and (d) states associated with deontic token life cycle. In the last case, the states associated with deontic tokens (i.e. deontic facts) capture shared views about mutual obligations, permissions or prohibitions between counterparts in the interactions, and their evolution in response to speech acts.
- The use of Corda *contract* can be used to verify the state transitions between key parties involved. This verification can apply to both basic behaviour and also deontic behaviour. Examples of the former are update of healthcare data of an individual, such as performed by the clinician. Examples of the latter are transitions from a state modelling an existing obligation to a state modelling the same discharged obligation, further characterized by a specified Corda time windrow.
- Corda *transactions*, implementing atomic updates of the contained states, can be used to ensure immutable audit trails, such as needed for the role of data tracker in the community.

The use of Corda platform effectively ensures embedding both of the monitoring and enforcement options, which was one of the options described in more in detail in Berry & Milosevic (2005).

7.4 Considerations for use

The use of smart legal contracts and DLTs can enhance the privacy, reliability and trustworthiness of digital health solutions, but their complexity and cost require a decision framework for managers, architects and developers to identify circumstances in which benefits outweigh costs. A good starting point is

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

National Institute of Standards and Technology (NIST) guidelines for determining whether a DLT might be valuable for a development initiative (Yaga *et al.*, 2019) as opposed to database solutions. Many of these guidelines would be useful for healthcare but with relaxed constraints related to trust. This is because in healthcare there are trusted institutions like hospitals or national bodies who can be relied on to verify the integrity of ledgers. This would mitigate the cost of the establishment and maintenance of trusted blockchain infrastructure, including some of the wastefulness of blockchain consensus mechanisms such as proof-of-work mining. We note that the recently established DeepMind verifiable Data Audit project (DeepMind, 2018) is aimed in further improving trust and they are developing their own ledger for data audit.

NIST guidelines are often used in the context of supporting immutable and tamper-proof storage of data. We believe however that there is value in additional guidelines for the use of smart contracts and support for enactment of certain processes in healthcare. The existing efforts to map traditional process languages onto smart contracts can be a useful starting point (Lòpez-Pintado *et al.*, 2018). This is relevant for those healthcare processes that can be standardized while supporting variability for personalization purposes. Early candidates are financial and supply chain processes, followed by referrals, discharges and care plans.

In doing so however, one needs to carefully consider the impact of non-digital transactions as they prevail in healthcare. In such cases, smart contracts are limited in their ability to control or monitor non-digital transactions. Rather, they can capture a trusted, immutable record of execution, if the transaction is subsequently recorded digitally. This applies to human-initiated actions, but even for a sensor reading, for example, the reading has already occurred and the ledger can only record the occurrence afterwards. A real-time monitoring system cannot wait for consensus before delivering an alert. In fact, there are many situations where a traditional smart contracts approach might not be necessary and can be replaced with process definitions.

In addition, many applications would benefit from guidelines for translating legal contracts into compliant processes (Milosevic *et al.*, 2006), because current process languages do not support expression of legal constraints over the activities in a process. Our earlier work provides many contributions in this respect (Berry & Milosevic, 2005).

8 Discussion and related work

This section presents an evaluation of our approach and considers our proposal in the context of early contributions related to the use of smart legal contracts and distributed ledgers in digital health. This evaluation also serves as a basis for future work presented in Section 9.

8.1 Policy formalisms for digital health

To the best of our knowledge, there are no specific policy formalisms for digital health. Our policy formalism relies on the general policy specification from RM-ODP standard recommendations, but Object Management Group (OMG)'s Semantics of Business Vocabulary and Rules (SBVR, 2017) standard can also be considered.

Our policy formalism has been developed over many years as part of the development and implementation of the Business Contracts Language (BCL). This language was evaluated and used in several proof-of-concept projects as extensively published in the past (Hanson & Milosevic, 2003; Milosevic *et al.*, 2004; Berry & Milosevic, 2005; Milosevic & Bond, 2016). The policy language was further evaluated through deontic formalism as also reported (Governatori & Milosevic, 2005). Similarly, the policy formalism was used as an element of the Australian e-health interoperability frameworks (Bond *et al.*, 2013; Milosevic & Bond, 2016).

Much of this formalism remains to be tested as part of specific digital health projects. We believe that the emerging interest in the specification of privacy, consent and ethics aspects of digital health, is likely to provide further fertile ground for its use, testing and enhancement.

This paper is mainly focused on one aspect of the enactment of digital policies, namely covering their translation from natural language to a computable digital format (Milosevic, 2018). The paper

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

also provides directions for integrating policy expressions across systems and stakeholders involved in digital health, shown through a privacy consent example and through more general mappings onto the components of Corda DLT.

The current issue in our approach is a lack of widely used software tools that can facilitate end-user entry of such policy expression and their integration with digital health systems, including the use of underlying DLT facilities. While the proposed formalism is independent of the underlying technology platform, our specific interest is in the context of DLTs and smart legal contracts. These technologies are in early stages of development, however, and lack widely adopted tools suitable for end-user policy definition and integration. The next two sections provide summary of current efforts in using DLTs and smart legal contracts in digital health.

8.2 Distributed ledgers in digital health

Primary use cases identified in early surveys are managing clinical trial records, regulatory compliance and managing medical/health records (IBM, 2016), and tracking and tracing pharmaceuticals, such as for the so-called 'cold chain break' (Fish and Barnard, 2018). This refers to the problem of vaccine degradation due to being stored in temperatures that are too hot, too cold or exposed to ultraviolet light. DLTs can provide increased trust through consensus, provenance and supply chain immutability across the parties involved, that is, manufacturers, public health authorities, central purchasers and auditing organizations. Three further use cases were identified in (Grieve, 2018). The first is legal assurance on audit trails to ensure that an audit trail has not been tampered with. Examples are compliance with GDPR removal-of-data requests, keeping records around infection control, and cases of sexual abuse or other criminal behaviour. The second use case is to support a legally established trust, where DLTs can be used to support the establishment of a legal agreement so that all parties involved are given confidence that true sharing of information would occur without the agreement giving advantage to any of the parties involved. The subsequent information sharing as specified in the agreement does not need to involve a DLT. The third use case is Clinical Credential Tracking.

There are an increasing number of open-source blockchain initiatives in healthcare (Center for Biomedical Blockchain Research, 2019). Some efforts propose the development of utility tokens for financial transactions between healthcare providers on the Web (MedicoHealth, 2019).

Dinh and Thai (2018) describe interesting work related to the opportunities arising from integration between blockchain and AI. Blockchain can be used to help in addressing AI explainability problems through tracing relevant steps in AI algorithms to help to understand decision-making processes. Such blockchain-based trails can assist in determining whether clinicians (and who specifically) or machines are at fault in case of accidents.

Further, DLTs augmented with finer-grain consent mechanism can support better sharing of patent information for research purposes, as presented in this paper. Moehrke (2018) proposes combining off-chain solutions, (i.e. FHIR servers which contain patient information), with a blockchain platform, which enforces legal rules of research consent directives between the patient and research organization.

There have been interesting experiments with the use of blockchain in the context of the Australian MyHealthRecord (MyHR) (Healthcare IT News, 2019). The experiments were aimed at allowing researchers to access medical information contained within MyHR, but uncertainty around current Australian legislation for the secondary use of data (Australian Government, Department of Health, 2018) suggests that these developments require a stable legal/regulatory framework before further investment.

8.3 Smart legal contracts in digital health

There are preliminary discussions from the finance industry (ISDA Linklaters, 2017) about the need to provide legal expression over the smart contracts, in particular, in support of derivative instruments, but there are no reported proposals yet. Early developments by Accord project on Ergo smart contracts language aim to help legal-tech developers quickly and safely write computable legal contracts (Ergo, 2019), but the language does not have deontic type of constraints over the actions of parties to support the elements of legal semantics such as provided by BCL and ODP-EL standard we adopt.

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

The related developments by Clause (2019) are focused on building services to automate business processes and contract management by connecting these two. For instance, they have built a tool that allows companies to receive payments automatically upon execution of a contract, eliminating the need to chase after payments (The National Law Review, 2019).

There is even more limited published work regarding the use of smart contracts in healthcare. One notable exception is the use of Etherium smart contracts to facilitate secure analysis and management of remote medical sensors, arising from the Internet Of Things and other patient remote monitoring systems as reported in Griggs *et al.* (2018). In this system, the sensors communicate with a smart device that calls smart contracts and writes records of all events on the blockchain. The solution supports patient monitoring and medical interventions by sending notifications to patients and medical professionals, while also maintaining a secure record of who has initiated these activities. The aim is to address security vulnerabilities associated with remote patient monitoring and automate the delivery of notifications to all involved parties in a manner compliant with Health Insurance Portability and Accountability Act (HIPPA, 2016). We note that the smart legal contracts could be used in activities such as procurement, where the automation of supply chain activities is likely to bring significant savings or in support of clinical trails, as discussed in Section 8.2.

There are several proposals for providing higher level business process expression of agreements between parties and mapping them onto specific smart contract solutions (Lòpez-Pintado *et al.*, 2018). These, however, do not consider adding legal constraints to process descriptions.

These are only preliminary suggestions and much remains to be developed to raise the level of abstraction to accommodate the expression of legal constraints over business processes and thus support the expression of smart legal contracts. It can be said that DLTs and smart contracts are not yet widely used in healthcare, despite a certain level of hype surrounding the technology. Perhaps, we will see some small changes in the short term and bigger changes in the longer term, as per Amara's law: 'We tend to overestimate the effect of a technology in the short run and underestimate the effect in the long run'.

9 Conclusion and future work

This paper provides our early thoughts about enactment of policies in digital health and use of distributed ledger and smart legal contracts technologies to support policy implementation over processes in healthcare. The availability of such technologies would facilitate better collaboration among patients and clinicians in a policy compliant manner, within the business constraints of cost, risk and trust. Our aim is to ultimately help empowering people in getting better control of their health data for their own care, while making it available to research community, to facilitate the development of true learning healthcare system.

Healthcare policy concerns are not only related to the distributed ledger topics but are also key concern for digital health interoperability and AI challenges, while ensuring their alignment with broader ethics principles. Our immediate goal is to engage clinical and health policy experts in detailed consultations about the form and usability of the policy language and leverage the latest design thinking practice to deliver rapid and stakeholder relevant policy language solutions.

We intend to develop a detailed mapping from our abstract policy language to BCL, CSL and recently published DAML language from Digital Assets (DAML, 2019), as well as to investigate positioning of smart contract concepts in the context of other platforms such as Etherium's Solidity smart contract language (Solidity, 2019) and Hyperledger Fabric smart contract language, often referred to as chaincode (Hyperledger, 2019). Further, we will investigate applicability of our earlier work on extending choreography with business contracts constraints in the distributed ledger context (Berry & Milosevic, 2005) and provide further experiments with Corda platform, following up on the analysis presented in this paper.

We plan to explore other uses of smart legal contracts as part of decentralized and federated health system, and the suitability of off-chain or hybrid solutions. For example, there might be value in specifying smart legal contracts purely for cross-organizational interactions (i.e. Business Process Model and Notation public processes) but leave the internal processes (i.e. Business Process Model and Notation private processes) performed off-chain. We also intend to look into specific deployment scenarios, where

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

smart legal contracts can be used to support reliable transfer of responsibility across providers within soft and hard time limits while ensuring provenance of data exchanged, for example, in referrals. We also plan to explore mapping of deontic and accountability concepts onto business rules used in the specification of shareable clinical pathways as part of a recent OMG initiative, exploiting the use of Business Process Model and Notation, Case Management Model and Notation and Decision Model and Notation in healthcare (BPM+, 2019).

In many situations, the terms of consent can change over time to reflect changing patient circumstances and we intend to investigate its impact on analytics, AI and privacy preserving approaches by considering the use of ODP-EL concept of policy. This is a general concept that describes any potential variation in a specification that is foreseen by the designer or other stakeholders at the design time, but the detail of which is determined subsequent to the original design, reflecting changing circumstances.

There is no standard reference architecture model for analyzing, designing and implementing DL and smart contracts. We plan to investigate the value of the RM-ODP architecture standards for this purpose (ISO, 2015; Linington *et al.*, 2011), as they provide a solid basis for describing and building widely distributed and federated systems systematically (Linington *et al.*, 2011).

Finally, we plan to apply our policy framework and related DLT solutions within FHIR® context, using relevant stable resources such as patient and observation while testing the emerging resources such as consent, contract, audit and provenance, using consent scenarios.

Acknowledgements

I would like to thank Dr Andrew Berry (Deontik) for his input to the earlier version of this paper and to Dr Florian Herzog (Deon, Digital) for fruitful discussions about the importance of legal semantics to smart contracts. I would also like to thank Grahame, Grieve (Health Intersections), for his insights regarding the FHIR® consent resource and for his outstanding technical contribution, vision and efforts in driving the HL7 FHIR® agenda.

References

Andersen, J., Elsborg, E. Henglein, F., Simonsen, J. & Stefansen, C. 2006. Compositional specification of commercial contracts. *International Journal on Software Tools for Technology Transfer* **8**, 485–516.

ResearchKit. 2019. https://www.researchandcare.org/researchkit/, (Accessed 4 Jan. 2020).

Australian Government, The Department of Health. 2018. *Implementing the Framework to guide the secondary use of My Health Record system data*, https://www.health.gov.au/internet/main/publishing.nsf/Content/eHealth-framework

Berry, A., Milosevic, Z. 2013. Real-time analytics for legacy data streams in health: monitoring health data quality. In *EDOC 2013.*

Berry, A. & Milosevic, Z. 2005. Extending choreography with business contract constraints. *International Journal of Cooperative Information Systems* **14** (02n03), 131–179.

Bond, A., Hacking, A., Milosevic, Z. & Zander, A. 2013. Specifying and building interoperable eHealth systems: ODP benefits and lessons learned. *Computer Standards and Interfaces* **35** (3), 313–328.

Center for Biomedical Blockchain Research. 2019. *Healthcare and Biomedical blockchains,* https://db.biomedicalblockchain.org, (Accessed 7 June 2019).

Clause, *Connected Contracting.* https://clause.io, (Accessed 27 Dec. 2019).

DeepMind, *Trust, confidence and Verifiable Data Audit*. 2018. https://deepmind.com/blog/trust-confidence-verifiable-data-audit/, (Accessed 7 June 2019).

Deon Digital, *The Deon Digital CSL Language Guide*, https://deondigital.com/docs/v0.39.0/, (Accessed 7 June 2019).

Digital Asset, *DAML Smart Contract Language, DAML,* https://daml.com, (Accessed 7 June 2019).

Dimitrakos, T., Djordjevic, I., Milosevic, Z., Josang, A. & Phillips, C. 2003. Contract performance assessment for secure and dynamic virtual collaborations, DAML. In *Proceedings of IEEE EDOC 2003 Conference.*

Dinh, T. N. & Thai T. 2018. AI and blockchain: a disruptive integration. *IEEE Computer* **51** (9), 48–53.

Ergo, Language Overview, https://docs.accordproject.org/docs/ergo, (Accessed 7 June 2019).

Etherium, *Solidity smart contracts,* https://www.ethereum.org

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

Fish, I. & Barnard, M. 2018. Saving money and lives with blockchain for coldchain breaks. *IBM Healthcare & Life Sciences Industries Blog, May 7.*

GDPR, *General Data Protection Regulation,* https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679ss, (Accessed 7 June 2019)

Governatori, G. & Milosevic, Z. 2005. Dealing with contract violations: formalism and domain specific language. In *IEEE EDOC 2005.*

Governatori, G., Idelberger, F., Milosevic, Z., Riveret, R., Sartor, G. & Xu, X. 2018. On legal contracts, imperative and declarative smart contracts, and blockchain systems. *AI and Law*, 1–33.

Grieve, G. *Blockchain in Heathcare – are standards needed?* 2018. http://www.healthintersections.com.au/?p=2778 (Accessed 7 June 2019).

Griggs, K. N., Ossipova, O., Kohlios, C. P., *et al.* 2018. Healthcare blockchain system using smart contracts for secure automated remote patient monitoring. *J. Med. Syst.* **42**, 130.

Lonsetteig, A. B. 2017. R3s Corda uncovered: it is not blockchain. *Global Trade Review*, https://www.gtreview.com/magazine/volume-15issue-3/r3s-corda-uncovered-not-blockchain/

Hanson, J. & Milosevic, Z. 2003. Conversation-oriented protocols for contract negotiations. In *IEEE EDOC 2003 Conference*.

Healthcare IT News (HITN), *Federal Government successfully trials blockchain for researcher access to Australian patient records.* https://www.healthcareit.com.au/article/federal-government-successfully-trials-blockchain-researcher-access-australian-patient, (Accessed 07 June 2019).

Brown, R. G. Carlyle, J., Grigg, I., Hearn, M. 2016. *Corda: An Introduction*, August 2016.

FHIR® Consent Resource, Release 4.0. 2019. https://www.hl7.org/fhir/consent.html, (Accessed 4 Jan. 2020).

HIPPA, *The Health Insurance Portability and Accountability Act of 1996,* https://www.hipaajournal.com/hipaa-compliance-checklist/, (Accessed 20 Dec. 2019).

Hyperledger, Architecture, Smart contracts, hyperledger.org, (Accessed 7 June 2019).

IBM Institute for Business Value. 2016. *Healthcare rallies for blockchains: Keeping patients at the center.*

ISDA Linklaters Whitepaper. 2017. *Smart Contracts and Distributed Ledger-A Legal Perspective, Aug. 2017,* https://www.isda.org/a/6EKDE/smart-contracts-and-distributed-ledger-a-legal-perspective.pdf (Accessed 7 June 2019).

ISO/IEC 15414. 2015. *Information technology: Open distributed processing, Reference model, Enterprise Language*, 3rd ed.

Linington, P., Milosevic, Z. Tanaka, Raymond, K. 1998. *Policies in communities: Extending the ODP enterprise viewpoint*. Proceedings Second International Enterprise Distributed Object Computing.

Linington, P., Milosevic, Z. Tanaka, A. & Vallecillo, A. 2011. *Building Enterprise Systems with ODP, An Introduction to Open Distributed Processing*. Chapman Hall/CRC Press.

Linington, P., Milosevic, Z., Cole, J., Gibson, S., Kulkarni, S. & Neal S. 2004. A unified behavioural model and a contract language for extended enterprise. *Data Knowledge Engineering* **51** (1), 5–29.

Linington, P., Miyazaki, H. & Vallecillo, A. 2012. Obligations and Delegation in the ODP Enterprise Language. *In The IEEE 16th International Enterprise Distributed Computing Workshops, EDOC 2012*.

Lòpez-Pintado, O., Garcia-Bañuelos, L., Dumas, M., Weber, I. & Ponomarev, A. 2018. *CATERPILLAR: A Business Process Execution Engine on the Ethereum Blockchain,* arXiv:1808.03517.

Milosevic, Z. & Bond, A. 2016. Digital health Interoperability frameworks: use of RM-ODP standards. In *IEEE EDOC SoE4EE Workshop.*

Milosevic, Z. & Bond, A. 2016. Services, processes and policies for digital health: FHIR® case study. In *IEEE EDOC SoE4EE Workshop*.

Milosevic, Z., Arnold, D. & O'Connor, L. 1996. Inter-enterprise contract architecture for open distributed systems: Security requirements. In *WETICE 1996.*

Milosevic, Z., Sadiq, S. & Orlowska, M. 2006. Translating business contract into compliant business processes. In *IEEE EDOC 2006 Conference.*

Milosevic, Z., Linington, P., Gibson, G., Kulkarni, S. & Cole, J. 2004. Inter-organisational collaborations supported by e-contracts. In *Building the E-Service Society Conference*, 413–429, Springer.

Milosevic, Z. 2018. Towards digitalisation of healthcare policies: case for smart legal contracts? In *Proceedings of the 3rd Symposium on Distributed Ledger Technology*, Griffith University, Nov. 2018.

MedicoHealth. 2019. https://medicohealth.io, (Accessed 4 Jan. 2020).

Moehrke, J. 2018. *Blockchain for Patient to Sell Their Data to Clinical Research,* Aug. 2018, https:healthcaresecprivacy.blogspot.com/2018/08/blockchain-for-patient-to-sell-their.html, (Accessed 07 June 2019).

The National Law Review. 2019. *Blockchain in Energy: Smart Legal Contracts on the Rise, July 2019,* https://www.natlawreview.com/article/blockchain-energy-smart-legal-contracts-rise, (Accessed 30 Dec. 2019).

Object Management Group (OMG), *Business Process Management for Healthcare (BPM+ Health),* https://www.bpm-plus.org, (Accessed 31 Dec. 2019).

Object Management Group (OMG). 2017. *Semantics of Business Vocabulary and Rules (SVBR*, https://www.omg.org/spec/SBVR/About-SBVR/, (Accessed 7 June 2019).

https://doi.org/10.1017/S0269888920000089 Published online by Cambridge University Press

Olsen, L. A, Aisner, D., McGinnis, J. M. (ed). 2007. Institute of medicine roundtable on evidence-based medicine. In *The Learning Healthcare System: Workshop Summary*. Washington (DC): National Academies Press (US). https://www.ncbi.nlm.nih.gov/books/NBK53494/ (Accessed 8 June 2019).

Smart Contract Alliance, *Lexicon,* https://digitalchamber.org/wp-content/uploads/2018/09/Lexicon.pdf, (Accessed 31 Dec. 2019).

Solidity, *The Solidity Contract-Oriented Programming Language.* https://github.com/ethereum/solidity, (Accessed June 2019).

The Conversation, *Google car crash: who is to blame when a driverless car has an accident?* http://theconversation.com/google-car-crash-whos-to-blame-when-a-driverless-car-has-an-accident-55664, (Accessed 20 Dec. 2019).

Yaga, D., Mell, P., Roby, N., Scarfone, K. 2018. *Blockchain Technology Overview*, NIST, Oct. 2018, https://doi.org/10.6028/NIST.IR.8202, (Accessed 7 June 2019).

Wikipedia Contributors. 2019. *Informed consent*, Wikipedia. https://en.wikipedia.org/wiki/Informed_consent, (Accessed 7 June 2019).

Zyskind, G. & Nathan, O. Decentralizing privacy: Using blockchain to protect personal data. In 2015 *IEEE Security and Privacy Workshops*.