Computable consent – from regulatory, legislative, and organizational policies to security policies

Zoran Milosevic[1][0000-0002-1364-7423] and Frank Pyefinch [1]

[1] Best Practice Software, Brisbane, Australia
{zoran.milosevic,frank.pyefinch}@bpsoftware.net

Abstract. Consumer-facing health applications are increasingly requiring flexible approaches for expressing consumer consent preferences for the use of their health data across multiple providers, and across cloud and on-premises systems. This and the recognition of the need for clear governance and legislative rules that specify enforceable policies over how consumer data is used by the nominated and other providers, including AI vendors, increasingly require machine readable, i.e. computable consent expressions. These expressions can be regarded as additional constraints over security policies, applicable to all stakeholders, while accommodating rules from regulatory and legislative policies. Support for both kind of policies contribute to improving consumer trust in the use of their data. This is applicable to both care delivery processes but also research projects, such as clinical trials. This paper proposes a computable consent framework and positions it in the context of the new developments within Health Level Seven (HL7®) Fast Health Interoperability Resources (FHIR®) standard. The proposal is based on the use of precise policy concepts from the ISO/ITU-T RM-ODP (Reference Model for Open Distributed Processing) standard. The aim is to provide general standards-based policy semantics guidance to interoperability/solution architects and implementers involved in digital health applications. The framework is driven by consent requirements, while leveraging broader policy input from medico-legal community.

Keywords: Consent, Policy, Interoperability, Health Level Seven (HL7®), Fast Health Interoperability Framework (FHIR®), RM-ODP, digital health.

1 Introduction

There are increasing number of initiatives aimed at engaging consumers in active participation in their healthcare, as part of the delivery of more effective, quality and evidence-based health care. One way of doing this is through new *digital health services* such as mobile applications or portals. They allow consumers pro-active participation in healthcare processes, spanning primary health care providers such as general practitioners and specialists, hospitals, and research institutions [12].

These services rely on the timely and effective access to consumer *health data* which can be shared in controlled way with relevant providers involved in delivery of health care or developing new health knowledge or solutions. Sharing of data is becoming

increasingly possible due to the growing adoption of the HL7 standard, Fast Healthcare Interoperability Resources (FHIR®) [1], which allows better sharing across organizational boundaries, supporting new *interoperability* solutions at scale.

It is natural that the individual health information, with its additional confidentiality and privacy constraints, requires that consumer privacy preferences are respected, including *consumer data rights*. This is needed to ensure consumers *trust* in how their data is used, for the benefits of their health care, but also in support of clinical research. The central element here is clear understanding of the *policies* surrounding consumers *consent*, which capture their preferences for what actions are allowed (or not) when accessing or sharing their data. These policies are guided by overarching legislative, regulative, corporate and security policies, for the use and sharing of health information, as for example documented in the Royal Australian College of General Practitioners (RACGP) guidelines [13]. This complex set of rules requires increasing automation in handling policies, including consent, which are currently predominantly paper based. This paper provides a proposal for expressing such policies in a machine interpretable, i.e. computable, manner, grounded in the latest approaches to the expression of policy semantics. We use the latest FHIR consent proposal, published in the Release 5, namely FHIR Consent Resource [9], as the focus for discussion.

1.1 Problem and contributions

This paper addressed the problem of expressing computable consent policies reflecting patient preferences, while adopting constraints by the enterprise and security policies. The enterprise policies cover legislative, regulative and corporate rules [2]. This problem is heightened in cloud-based environments used for building FHIR enabled applications across administrative boundaries. The paper provides two main contributions:

- semantic foundations for platform-independent models for consent related policies, supporting both enterprise and security policies
- positioning of the above models in the context of distributed architecture associated with FHIR APIs and its consent information models while accommodating broader policy support for governing data sharing/use across digital health ecosystem.

Next, we present related work in support of automated consent management. Section 2 describes current FHIR Consent resource specification. Section 3 presents the generic computable policy/consent framework. Section 4 discusses the positioning of the generic framework with FHIR. Section 5 discusses future work directions.

1.2 Related work

There are several research and standardization efforts relating to making certain aspects of consent automated and scalable.

One example are ‘dynamic consent’ approaches, which aim to facilitate more engaged and personalized communications between researchers and participants in a research study, through enabling participants to manage their consent preferences over

time. One such solution was recently used in genomic research in Australia [3], which developed a web-based application tool called CTRL (control). CTRL facilitates ongoing participant-led management of their involvement in research, by allowing participants to choose from granular consent options and change consent choices over time (including for future use of their data). Participants can indicate preferences for the kinds of results they would want returned, whether they receive alerts about further research their data is shared to, and their preferred methods of contact.

Another example is a scalable consent framework for electronic health records, developed by San Diego Health Connect, funded by ONC Leading Edge Acceleration Projects in Health IT (LEAP) program [4]. Their work focused on how to use FHIR-based application programming interfaces (APIs) to allow patients to electronically document and share their consent preferences to streamline availability of information relevant to their care. This proof-of-concept project proposed a scalable and decentralized architecture for managing and enforcing patient consents. The emphasis was in supporting relatively straightforward permit or deny type of policies regarding consent, but the growing complexity and sophistication of patients' control over their health data and their sharing across multiple providers requires more powerful computable consent framework. The solution components and available software however represent the most advanced contribution to the field, while also recognizing that further efforts are required across government and the private sector to build a scalable consent management policy and regulatory architecture.

Further, the HL7® standardization organization has recently published a Consent Management Service [5], leveraging contribution from the LEAP project but also from our earlier proposals [6]. This service is independent of any underlying digital health platform and was influenced by FHIR Consent resource developments [9].

There are also early efforts in better supporting consumers in primary health practice in expressing their consent preferences. One example is providing consent for various kind of communications to patients, such as for reminders of their appointments or clinical events, as is done with the Best Practice Premier on-premise product [16]. Another example is the profiling of the FHIR Consent resource (Version 4), for the My Script List (MySL) component of ePrescribing in Australia [11] to be discussed in section 2.3.

These initiatives, and the FHIR consent standardization (see next section), demonstrate different efforts in automating consent, but do not adopt an agreed modelling framework for expressing consent preferences as computable constraints on behaviour of parties involved in handling consent. This is particularly important when consent is considered in terms of interaction with other constraints that specify a broader set of accountability, responsibility and delegation policies, arising from legislative, regulative or security policies. This paper provides such a computable policy framework, leveraging stability and credibility of relevant parts of the ISO RM-ODP standards, in support of building systems in which parties' behaviour can be monitored and enforced by implemented systems.

2 Towards consent automation – FHIR approach

FHIR [1] provides specification of a number of modelling concepts for designing, deploying and operating digital health applications. The semantics of the modelling concepts is grounded in many years of HL7 standardization, while the adoption of the commonly used web technologies for building applications makes FHIR increasingly popular among the development community. Key interoperability features of FHIR are:

- common modelling language concepts referred to as FHIR resources, specified using UML, XML and JSON languages
- API style of application developments, relying on the modern web technologies, to support exchange of data and applications across the web
- controlled extension approaches, to reflect specific domain interests, e.g. different national requirements or application domains – known as FHIR profiles.

2.1 FHIR consent resource – basic policy and computable policy expressions

FHIR standard recognises the need to have a flexible specification of consent to reflect a wide range of preferences of consumers. FHIR defines consent as [9]

— *A record of a healthcare consumer's choices or choices made on their behalf by a third party, 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*.

This definition uses the general concept of an *action* performed by an agent, allowing to capture three type of uses of the Consent resource: a) privacy consent directive, being an agreement, restriction, or prohibition to collect, access, use or disclose (share) information, b) medical treatment consent directive, as consent to undergo a specific treatment (or refusal to it), and c) research consent directive, as an agreement to participate in research protocol and information sharing required. These agreements are provided by a healthcare consumer [grantor] or their personal representative, to an authorized entity [grantee] for an authorized or restricted actions with any limitations on purpose of use, and handling instructions to which the authorized entity must comply [9].

Simple consent form. In its simplest form, the Consent resource provides attributes to record the content and the metadata of a consent (either implicit consent as an event or an explicit consent document), enabling consent discovery by indexing, searching, and retrieval of consents based on this metadata. The key attributes are:

- Subject – reference to whom the consent applies (e.g. Patient, Practitioner)
- Grantor – reference to who is granting rights according to the policy and rules (e.g. Patient, RelatedPerson, Practitioner, CareTeam, etc)
- Grantee - reference to who is agreeing to the policy and rules (e.g. Organisation, Practitioner, RelatedPerson, CareTeam etc)
- DateTime - when consent was agreed to

- Manager – reference to a workflow consent manager (e.g. HealthService, Organisation, Patient, Practitioner etc.)
- Enforcer – reference to a consent enforcer
- Source - used to record the original consent document either in the form of a pointer to another resource or in the form of an attachment.

Note that the concepts of Patient, Practitioner, RelatedPerson and so on, are other FHIR resource concepts, capturing key properties of these information elements [1].

Support for computable consent. A more advanced usage of the Consent resource requires computable expression of privacy preference rules. These rules can be processed by a decision engine to decide whether the given consent permits a specific activity (e.g., sharing the patient information with a requester or enrolling the patient in a research project). There are two mechanisms for recording computable consent:

- the *provision* structure which provides a simple structure for specifying additional exception to the base policy rule (or default policy) which is about permitting or denying particular action; for example, access to patient Electronic Health Record (EHR) is generally not permitted (base rule), except when in emergency, and this hold for 7 days (exception with AND condition)
- the *policy* attribute which provides a more flexible mechanism via referencing a policy coded in a policy language of choice. FHIR does not prescribe a type of policy language to be used, with examples being XACML[7], ODRL[8] (see 3.2)

Note that each exception in the provision structure can further be refined in a hierarchical manner, but the approach does not provide ways of dealing with conflicts, such as when one exception conflicts with a higher-level exception, e.g. whether a more specific rule overrides a more general rule.

In terms of the consent enforcement options, this can be done using a mix of various access control enforcement methodologies (e.g. OAuth2.0, XACML). This enforcement includes the detailed elements of the privacy consent, such as the rules reflecting which organizational roles have access to what kind of resources (e.g. RBAC, ABAC).

2.2 Link with Smart on FHIR architecture pattern

We believe that the computable consent expressions, when available in the FHIR Consent resource, can be used to constrain the security policies for specifying and enforcing access to patient data on an EHR/FHIR server. For example, such policies can guide the use of OAuth2.0 security server to determine whether to issue a OAuth2 token for a client app. In fact, OAuth2 plays a central role in one approach to building FHIR applications, the so-called SMART (The Substitutable Medical Applications and Reusable Technologies) on FHIR. This is an open-source, standards-based API that leverages the OAuth 2.0 to ensure secure, universal access to EHRs [10].

The SMART on FHIR is intended to be used by developers of apps that need to access user identity information or other FHIR resources by requesting authorization from OAuth 2.0 compliant authorization servers. The apps can be used by clinicians, patients, and other parties, and it provides a reliable, secure authorization protocol for

a variety of app architectures, including apps that run on an end-user's device as well as apps that run on a secure server [10]. The SMART on FHIR process begins with a user starting an app requesting authorization from an EHR's authorization server, using *scope* parameters specifying the type of access, i.e. specific information about a patient, e.g. observations and read or write permission. If the authorization server permits this access, it returns an access token to the app, which allows the SMART app to call the FHIR server API, and access particular patient's record on the EHR's FHIR server according to the scope parameters. Smart launch also supports authorization for backend services, allowing their direct connection with an EHR when there is no user involved in the launch process, or when permissions are assigned to the client out-of-band.

Thus, the use of a computable consent policy expression to constraint scope of via consent management service with the SMART on FHIR allows linking enterprise policies from computable consent with the security enforcement approaches of SMART.

2.3 Analysis

There are recent efforts in FHIR Release 5 to improve FHIR Consent resource expressiveness to accommodate computable policies, through reference to computable policy expressions, i.e. provision structure and policy attribute, as mentioned above. New experience developed over initial deployment projects, e.g. LEAP project also suggests integrating their consent architecture with OAuth2. This, plus a broader set of policies surrounding consent, such as the expression of ownership and delegation, motivate us to apply a generic policy framework to consent, in the next section.

There were some initial attempts to use the FHIR Consent resource from an earlier FHIR version (Release 4), and specialized them for specific domain of use, specifically the Australian efforts for ePrescribing [10]. Here, My Script List (MySL) supports recording a) permissions from a patient to access their prescriptions to an Organization and b) for the MySL system to upload the patient's active prescriptions from the script exchange. This involved profiling of modelling elements such as identifier attributes for FHIR Patient resource, reflecting Australian elements such as Medicare, DVA (Department of Veteran Affairs), IHI (individual health identifier), telecom contact details, etc. It is to be noted that the earlier version of FHIR Consent resource did not support inclusion of computable consent expressions, and also used the Consent *scope* attribute to capture different type of consents, namely, the privacy, research or treatment consent, modelled using string datatype. This was certainly a modeling option available at the time, but this approach was not adopted in FHIR Release 5 Consent resource. This allows accommodating richer semantics needed for support of different type of workflows associated with different type of consent, including integration of better monitoring and enforcing of policies applicable to such workflows.

3 Computable policy framework

The FHIR Consent resource uses the terms of 'permit' and 'deny' which are constraints on the actions of the parties when they fill the role of grantee. In other words, they are

permissions or *prohibitions* for what the parties are allowed (or not) to do, including additional details such as for how long these conditions may be valid. It is thus possible to express consent in terms of the conditions specified in permissions and prohibitions as special type of policies.

For example, grantee's permissions are obtained through the grantor passing on their permissions, which in effect is the *authorization* for grantee, i.e. giving them ability for actions which otherwise they would not be able to do, i.e. giving them access to grantor's own health data. The authorization also places an *obligation* on the grantor, to ensure that access to the medical record is ultimately enabled, e.g. by passing security credentials to the grantee. Once the grantee has obtained permission, they would also need to satisfy other obligations, such as those arising from their medical duties (e.g. duty of care) and obligations to respect the grantor's privacy and confidentiality.

The concepts of permissions, prohibitions, obligations and authorization are regarded as fundamental types of policy constraints, each of which constrains actions of parties as they fulfill the roles to which these policies apply. Their formal expression is the subject of deontic logic [19] and these are often referred to as *deontic* constraints. They are prescribed by some combination of legislative, regulative of organizational authorities (i.e. policy context), each of which specifies rules of behaviour required to satisfy some objective, business, social or ethical.

Observe that these policy concepts are described in terms of actions, or composition of actions (behaviour), in a way that can be iteratively translated in machine executable statements, or computable expressions. This makes it possible to apply the semantics of the RM-ODP standards [2][14], which supports formal, and thus computable expressions of such policies, developed for the purpose of building technology independent and interoperable ecosystems. These policy concepts are defined next.

3.1 Modelling concepts for policy rules

The following is a list of several key policy modelling concepts, capturing the deontic and accountability constraints. Further details can be found in [6].

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.

Authorization is an action indicating that a particular behaviour shall not be prevented. Unlike a permission, an authorization is an empowerment.

Note that *prescription* is formally defined as an action that establishes a rule. Prescriptions provide a powerful mechanism for changing the system's business rules at runtime, enabling dynamic adaptation to respond to business changes and new needs.

The RM-ODP standard provides a pragmatic solution for translating these concepts into components that can be used in support of building enterprise distributed solutions. This is done through the concept of the *deontic token*, which has been developed to support explicit association of deontic constraints with the agent to which these

constraints apply [2][6][14]. These are enterprise objects which encapsulate deontic constraint assertions. The holding of the deontic tokens by parties constrains their behaviour. This is a powerful modelling approach because it provides a basis for manipulating deontic tokens, for example, passing them between parties to model delegations, and activation or de-activation of policies that apply to the parties. There are three types of deontic tokens, called burden, representing an obligation, permit, representing permission and embargo, 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 [13].

The deontic concepts above serve as primitives for expressing various type of accountability, such as the concepts of delegation, commitment and rights. Further, the organizational, regulatory or legal policies are defined within their corresponding contexts, which can be formally expressed by the RM-ODP concept of *community*. A community defines how a set of participants should behave in order to achieve an objective, through the interactions between roles and the policy constraints that apply to them. These participants (or enterprise objects in RM-ODP terms) fulfill *roles* in a community, and thus accept policy constraints that apply to the roles, as stated in the contract for community. At any point in time, at most one enterprise object can fulfil a community role. A community specification may include a number of role instances of the same type, each fulfilled by a distinct enterprise object, with the constraint on the number of roles of that type that can occur, e.g. maximum number of patients in a ward.

3.2 Policy language options

The modelling concepts above are used as a basis for designing an architecture in support of the specification and dynamic management of policies. The form of policy rule expressions that is embedded in each of the deontic tokens and other concepts, and which would need to be referenced by FHIR Consent resources, is not prescribed by the RM-ODP standard.

In our previous work [6] we have proposed a **generic policy language**, that is informed by the RM-ODP standard and with the following form:

⟨policyContext⟩⟨Activation⟩⟨role⟩⟨modality⟩⟨eventpattern⟩⟨targetrole⟩⟨violation⟩

<*policyContext*> denotes context of policy, such as legislative or organisational source of policies, for which the *community* can be used, as introduced above
<*Activation*> specifies trigger *events* for dynamic activation of normative policies; these can be temporal events such as timeouts, other events such as violation of other policies, or accountability actions, e.g. prescriptions or delegations;
<*role*> denotes a *community role*, to which deontic modality and behavioural constraints apply (defined by the community context);

<*modality*> denotes *deontic modality* that applies to the party fulfilling a community role, e.g. an obligation, permission or prohibition;
<*eventPattern*>specifies the expected behaviour of a party in terms of their actions and other occurrences such as timeouts;
<*targetRole*> denotes a community role that can be affected by the actions of the subject roles, and included as part of deontic modality;
<violation> condition which specifies other policies which can be triggered in response to a violation of the primary deontic modality.

So, privacy consent type or template for accessing consumer record can then be:

<ConsentContext> <consentActivation> <grantor> <permission> <accessConsumerRecord><grantee><violation>

accessConsumerReral specifies an event pattern, e.g. the start and end of an interval for which the consent was given and its purpose, for example, access to a specific IT resource. This general consent statement can be instantiated for a specific consent policy instance. Thus, the consent statement: 'A consumer John grants permission to an emergency clinician to access his EHR record, in case of emergency.'

<EDcare> <emergencySituation> <John> <permission> <accesEHRRecord> <accreditedEmergencyClinician <>

This policy is activated by emergencySituation event, selected from a set of possible triggering events that can be pre-defined by a clinical provider or jurisdiction. The policy assumes the existence of patient identifier framework, for example, Individual Health Identifier in Australia, which would identify the patient John in this case. Note that no violation condition is specified here.

There are several **specific policy languages** as targets for this generic language. They are selected to reflect event-condition-action pattern (suitable for real-time monitoring) while addressing deontic constraints semantics, as introduced next.

XACML (eXtensible Access Control Markup Language), is a security language for providing a declarative fine-grained, attribute-based access control policy language. Each policy is defined in terms of rules, the evaluation of which provides Boolean permit/deny decision to a particular action or resource. XACML adopts the IETF's architecture for policy management, with Policy Decision Point (PDP) evaluating policies against access requests provided by Policy Enforcement Points (PEP). XACML defines obligation as a directive from the PDP to the PEP on what must be carried out before or after an access is approved. XACML is suitable for expression of access control following the pessimistic style of enforcement but is not suitable for more flexible approaches to expressing optimistic enforcement options, where certain policy breaches are allowed to occur, once they are detected. Optimistic approaches allow for resolution through mediation mechanisms, such as negotiation. This means having a flexible way of dealing with violations of obligations, such as invoking other corrective policies.

Open Digital Rights Language (ODRL) to some extent address this limitation of XACML for consent management enforcement. In ODRL policies are used to represent

permitted and prohibited actions over a certain asset, across two predefined roles called ‘assignee’ and ‘assigner’. There is also support for obligations, through the concept of ‘duty’. Recent experience with ODRL however reports significant limitations in dealing with the dynamics of policies [15]. For example, there is a no mechanism that would support a patient’s revocation of their consent at any given time, there is semantic ambiguity in the concept of duty, and delegation approach using ‘transfer’ action leads to difficulty with the expression of delegation options in which grantor would allow delegating permission but still keeping it’s own permission [15].

Business Contracts Language (BCL) may best support the general language requirements [6], in part because it is grounded in the semantics of RM-ODP standard concepts, both for the behavioural and policy semantics, as presented in the previous section. As a result, BCL language would have similar structure as the general policy language above. 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. BCL back-end components are implemented in Java and use contemporary software to implement interfaces, including Web-based technologies. The language can be used to specify monitoring conditions for obligations and thus support the optimistic style of enforcement. This out-of-band real-time monitoring of activities of the parties against policy rules 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 in the role of a monitor. Once the monitor detects a breach, it can invoke discretionary or non-discretionary enforcement options.

Consider 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 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.

One disadvantage of BCL is that it was developed as a proof of concept, with many examples described in various publications, but there are no available open source implementations yet.

It is to be noted that policy language options introduced above are all declarative in style, which are suitable for the expressions of constraints. They are also independent of the details of widely used implementation languages that can be used to implement their functionality such as needed for event-based monitoring of policy expressions. It is expected that each deployment environment will dictate selection of the

implementation languages. Further, the FHIR based DevOps environment might require its own language options, reflecting in the FHIR based tooling available.

3.3 Example – privacy consent

Figure 1 depicts the key roles of Grantor and Grantee in a consent community, supported by several other roles needed for consent management [6], listed next.

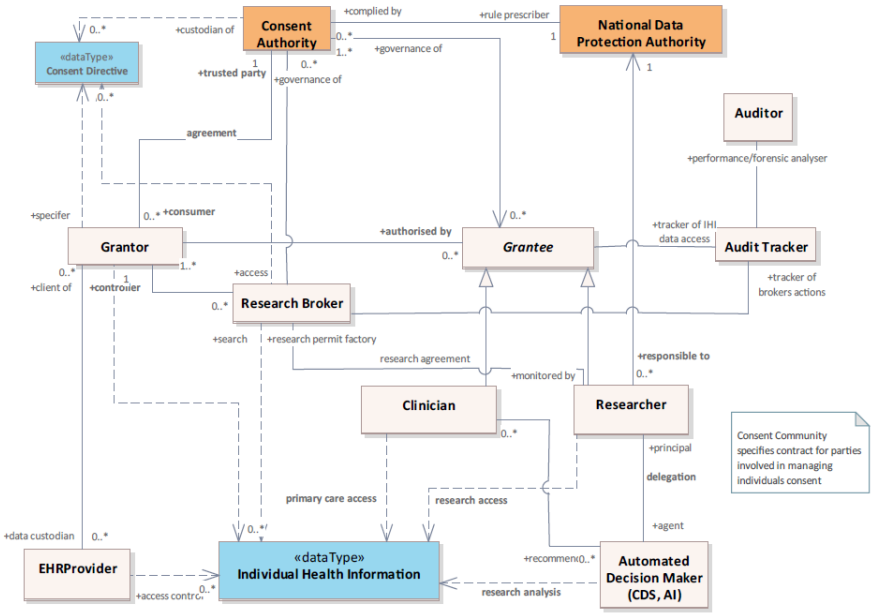

Figure 1: Consent management community

- Grantor, to be fulfilled by any individual giving consent, possibly respecting other constraints, such as being of legal age, having normal cognitive function etc.
- Grantee, to be fulfilled by professionals with the required credentials, such as Clinician, permitted to access Grantor's individual health information for care purposes, or Researcher, permitted to access Grantor's de-identified health data for research purpose, and with an obligation not to perform re-identification of patient data.
- Consent Authority, a trusted party responsible for storing individuals' consents and overseeing the consent agreement rules.
- Research Broker (Broker from here on), a legal entity authorized to search patient health and consent data to identify patients suitable for research study, e.g. cancer research. The Broker is responsible that patient preferences are enforced.
- National Data Protection Authority, responsible for defining and enforcing data protection policies.

- *Electronic Health Record (EHR) provider*, custodian of individuals' personal health data in their EHR records. They are usually prohibited by law from releasing patient data without consent, except when a clinician is providing emergency care.
- *Automated Decision-Maker*, performing analytics, recommendations and in some cases, active decision-making, augmenting activities of clinicians or researchers; this role can be fulfilled by clinical decision support systems or AI systems.
- *Audit Tracker*, logging actions of clinicians and researchers to generate audit trails, which can be used for subsequent activity analysis, e.g. by an Auditor;
- *Auditor*, providing analysis of event traces to support performance analysis or forensic investigations, such as detecting breaches of clinicians accessing healthcare records outside of them providing care.

The scenario below illustrates a normal sequence of *actions* from the time an individual gives consent until their data is used by researchers.

1. Grantor updates their consent directive at the Consent Authority allowing their de-identified genomic data to be used for cancer research, excluding mental health data
2. Grantor permits Broker matching on their de-identified data, needed to retrieve the identifiers of those patients whose consent matches the research study parameters. It does not give them access to the health data, just search and retrieve identifiers.
3. Researcher contacts the Broker stating their interest in conducting research across all patients who have given consent for cancer research; this includes access to their medications, treatment and genomic information.
4. Broker provides a de-identified list of eligible patients to the Researcher and gives an authorization for them to access de-identified patient data from the EHR provider, with the exclusion of medication data related to mental health treatment. The authorization requires the Researcher to maintain an audit trail of all data access.
5. Researcher retrieves de-identified patient data from the EHR provider. 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.
6. Researcher accesses the EHR data for their research, lodging an audit record for each access with the Audit Tracker; an AI system must also lodge access, as it acts on behalf of the researcher using their authorization (a research permit token).
7. Researcher publishes the result of the research and informs all relevant parties.
8. At a later point, a patient suspects that their mental health data were used by a health insurer and then contacts the Consent Authority to lodge a complaint.
9. Consent Authority engages Auditor who accesses audit trail to perform forensic investigation of patient's data access by Grantees. Upon detection of a violation, it notifies an enforcer to apply penalty to either party (not shown in this diagram).

The following are examples of *policies* for the community roles and their actions:

- 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 Broker which may be needed for forensic purpose.
- Authorization of the Grantor to the Researcher to access the Grantor's individual health information, as follows.
  - Grantor first authorizes (issues permit to) the Broker for searching their data to establish whether they satisfy research question criteria.
  - Broker then issues a research permit to the Researcher which includes a list of Grantors who provided consent to access their de-identified health data. Note that the Researcher might pass this permit to an AI system, delegating computations
  - EHR provider then allows access permit to the Researcher to access health records of specific patients, provided Researcher has credentials requested by the EHR provider; this can rely on the use of Smart On FHIR backend launch (see 2.2).

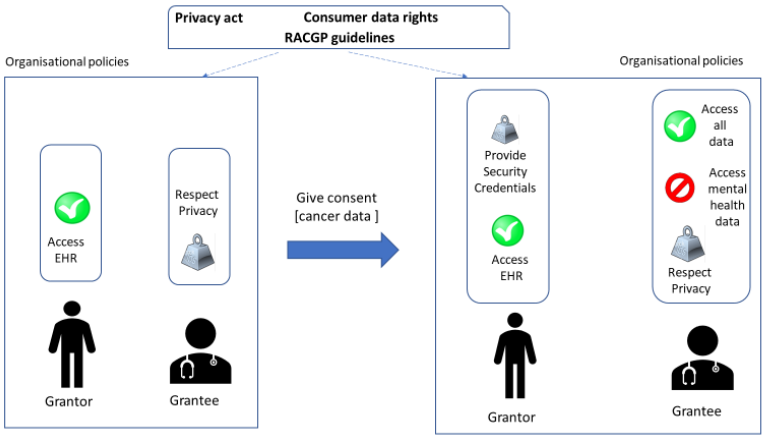

Figure 2: The dynamics of deontic tokens before and after giving consent

Authorization is modelled using a combination of permit and burden tokens. For example, authorization of the Grantor to the Broker above, involves the permit 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 (e.g. by providing security credentials). This authorization changes the deontic state of both the Grantor and Grantee, the effect of which is that the Grantor's permit to the Broker to search its healthcare data is passed on to the Researcher.

Figure 2 depicts how the consent action changes deontic states of Grantor and Grantee, in terms of different deontic tokens of the agents, before and after the action, while in compliance with legislation, regulatory and organizational polices. Note that data protection rules defined by a National Data Protection Authority set accountability and legal responsibility for researchers in using health data. These rules were established through *prescription* actions of the Authority, establishing obligations and permissions for all parties when accessing patient data in this community.

4 Positioning with FHIR ecosystem

The computable consent framework, consisting of all deontic and accountability modelling components from section 3.1, can be integrated with a FHIR application ecosystem, as shown in Figure 3. This includes the integration of FHIR resources with the burden and permit objects, which are associated with the actions of community roles. Some of these deontic constraints are result of the policies prescribed by regulators or other authorities, and others are dictated by the security policy mechanisms of the undelaying platforms, such as the access token of the Oauth2.0, which can be regarded as being a special kind of Permit.

The deontic tokens representing deontic constraints, can be accessed by or transferred with the data associated with processes in a consent community. The interpretation of policy language expression fragments that they carry (depicted as PLEs in the figure), can be executed by a policy engine (i.e. consent policy engine).

It is through these deontic token objects and policy language expressions that computable policy statements can be evaluated and enforced. In the FHIR application ecosystem, FHIR Consent can be modelled as a combination of consumer permits and providers burdens, which, when embedding a computable policy language of choice, such as XACML, ODRL or BCL, can be used as the target from the policy attribute specified in the FHIR consent resource (see section 2.1).

Similarly, a FHIR Contract resource stating rules for sharing data and services across partners, can be described in terms of the burdens associated with each party, reflecting the contract conditions, again described in a policy language of choice.

Recall that the policy framework above provides a solution in support of the dynamics of passing permits and burdens across parties in a system as well as creating new deontic tokens to constrain actions of the parties. This supports quite a general way of expressing accountability, ownership, creation/change of new policies, which surround consent to broader controlled data exchange. These token objects can also provide traceability to strong security mechanisms, such as for example when using Oauth2 authorization, of XACML and RBAC access control.

There are many other FHIR resources that use or are referenced by the FHIR consent. In our example, and in relation to a typical clinical trial research, a FHIR ResearchSubject resource can be used to model a party filling a grantor in the related research study (modelled as FHIR ResearchStudy resource). Here, an agreement between the EHRProvider, Broker, and any other third parties, such as the Broker or Automated Decision Maker community roles, can be specified using FHIR Contract (see Figure 3).

The figure also depicts a generic policy editor which can be used to create consent forms, and consent templates, and the FHIR Questionnaire and QuestionnaireResponse resources can be used for this purpose.

The FHIR Provenance resource, leveraging W3C provenance specification [18], can be used to manage the tracking of the changes to the Consent. Further, FHIR DocumentReference can be used as an attachment to show the stages of consent with additional or updated document(s) attached at each stage. The Contract resource can be used like a Document Reference where, as signatures are gathered or conditions applied, the Contract can be updated and attached to the Consent. In general, the Contract resource

represents a legally enforceable, formally recorded unilateral or bilateral directive i.e., a policy or agreement [1].

FHIR AuditEvent resource can be used to support the operations associated with AuditTracker and Auditor roles in the consent community.

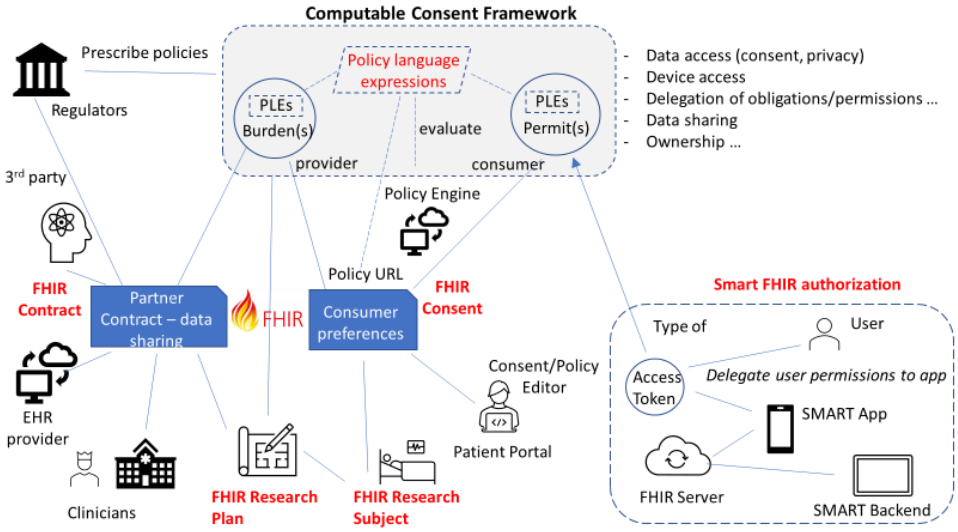

Figure 3 Computable consent framework integration within FHIR ecosystem

5 Conclusion and future work

This paper has proposed a computable policy framework that can be used in cases when relevant FHIR applications may require a domain language for expressing legislative, regulative and organizational policies - in a way that can be processed by machines, interpreted, and used to invoke security policy components, such as Oauth2 authorization or role base access control. The paper focuses on the variety of policies surrounding patient consent, both privacy and research consent, including those that are defined by relevant authorities, such as the RACGP's policies for managing health information and privacy in general practice [13].

Our future work will aim at implementing this computable policy framework in a FHIR application ecosystem such as Azure FHIR server [17], in primary health care context [12]. The first step is defining an overall architecture for consent management enforcement, making use of FHIR resources, followed by its implementation using FHIR based tools, patterns and implementation guides. The architecture could accommodate the components in Figure 3 but also additional component such as policy editors, consent forms and integration with SMART on FHIR launch. We also plan to give a better account of actions, known as *speech acts* in the RM-ODP enterprise language [14], needed for expressing delegation, authorization and commitment. The second step is to select a policy language of choice, that best reflects the policy semantics above and investigate its mapping into a suitable implementation language, used in the FHIR community, such as Java, C# or Python.

We also plan to discuss these issues with medico-legal practitioners to ensure that it is a legally verified approach, as well as ethics specialists to help inform building applications in which potential policy conflicts arise. Finally, we hope that this proposal may be of interest for future standardization of the FHIR Consent resource.

Acknowledgments

We would like to thank our colleagues from Best Practice Software, especially Daniel Kerridge, Gina Clement and Anthony Lee, for providing valuable input to this paper.

References

1. Health Level Seven (HL7®) Fast Health Interoperability Framework (FHIR®) https://build.fhir.org/index.html, last accessed 2022/08/08.
2. Linington, P., Milosevic, Z. Tanaka, A. & Vallecillo, A. Building Enterprise Systems with ODP, An Introduction to Open Distributed Processing. Chapman Hall/CRC Press., 2011.
3. Haas, M.A., Teare, H., Prictor, M. et al. 'CTRL': an online, Dynamic Consent and participant engagement platform working towards solving the complexities of consent in genomic research. European Journal of Human Genetics 29, 687–698 (2021)
4. Scalable Consent Framework for the Advancement of Interoperability with FHIR-based APIs, https://www.healthit.gov/topic/2019-leap-health-it-projects#Scalable, last accessed 2022/08/08
5. HL7 International, Services Functional Model: Consent Management Service, Release 1, Jan. 2021, HL7 STU Ballot.
6. Milosevic, Z., Enacting policies in digital health: A case for smart legal contracts and distributed ledgers? The Knowledge Engineering Rev, 35, Cambridge Univ. Press, Feb 2020
7. XACML, http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html
8. ODRL Information Model 2.2, W3C Rec., 15 Feb, 2018, last accessed 2022/08/08
9. FHIR Consent resource, https://build.fhir.org/consent.html, last accessed 2022/08/08.
10. SMART App Launch, http://hl7.org/fhir/smart-app-launch/, last accessed 2022/08/08.
11. ePrescribing API, StructureDefinition: MySLConsent https://fhir.medicationknowledge.com.au/dev/StructureDefinition-mysl-consent.html, accessed 2022/08/08.
12. Australian Institute of Health and Welfare, Primary health care, https://www.aihw.gov.au/reports-data/health-welfare-services/primary-health-care/overview, last accessed 2022/08/08
13. The Royal Australian College of General Practitioners. Privacy and managing health information in general practice., East Melbourne, Vic: RACGP, 2017.
14. ISO/IEC 15414. 2015. Information technology: Open distributed processing, Reference model, Enterprise Language, 3rd ed.
15. KEBEDE, M., SILENO, G and VAN ENGERSA, T, Critical reflection on ODRL, in AI Approaches to the Complexity of Legal Systems XI-XII: AICOL International Workshops 2018 and 2020, Revised Selected Papers
16. Best Practice, BP Premier, https://bpsoftware.net/bp-premier/, last accessed 2022/08/08.
17. FHIR Server for Azure, https://github.com/Microsoft/fhir-server, last accessed 2022/08/08.
18. PROV-DM: The PROV Data Model, W3C Rec. 30 April 2013, last accessed 2022/08/08
19. von Wright, G. H. (1951). "Deontic Logic". Mind. 60: 1–15.