International Journal of Medical Informatics (2006) **75**, 597—623

International Journal of Medical Informatics

www.intl.elsevierhealth.com/journals/ijmi

Modelling privilege management and access control

Bernd Blobel[a,*], Ragnar Nordberg[b], John Mike Davis[c], Peter Pharow[a]

[a] *Fraunhofer Institute for Integrated Circuits IIS, Health Telematics Project Group,*
Am Wolfsmantel 33, D-91058 Erlangen, Germany
[b] *Sahlgrens University Hospital, Gothenburg, Sweden*
[c] *US Department of Veteran Affairs, CISSP, SAIC, Veterans Health Administration, CA, USA*

Received 14 July 2005; received in revised form 19 July 2005; accepted 23 August 2005

KEYWORDS
EHR;
Model-driven architecture;
Components;
Constraint models;
Authorisation;
Privilege management;
Access control;
Security infrastructure

Summary
Objectives: For establishing trustworthiness in advanced architectures for future-proof health information systems being open, flexible, scaleable, portable, and semantically interoperable, security and privacy services needed must be designed as an inherent part of the architecture. Such architecture has to meet the paradigms of distribution, component orientation, formal modelling, separation of logical and technological aspects, etc.
Methods: In model-driven architectures components providing security and privacy services have to be specified using the same methodology of formal models with meta-languages as expression means, as deployed in computational, technical, or medical domains. The resulting approach must be based on the ISO Reference Model—Open Distributed Processing.
Results: Currently, standards developing organisation are defining emerging tasks and standards for semantic interoperability and trustworthy collaboration for advanced health information systems. Communication security issues have been specified and implemented, while application security challenges such as privilege management and access control are still under development. Therefore, a series of formal models have been developed by the authors covering, e.g. domains, service delegation, claims control, policies, roles, authorisations, and access control. The required models are introduced and interpreted in a generic way. The crucial concept of security policy and its relationship to the other concepts has been considered in detail.
Conclusion: Based on formal models, security services can be integrated into advanced systems architectures enabling semantic interoperability in the context of trustworthiness of communication and co-operation.
© 2005 Elsevier Ireland Ltd. All rights reserved.

[*] Corresponding author. Tel.: +49 9131 776 7350; fax: +49 9131 776 7399.
E-mail address: bernd.blobel@iis.fraunhofer.de (B. Blobel).
URL: http://www.iis.fraunhofer.de.

1386-5056/$ — see front matter © 2005 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.ijmedinf.2005.08.010

1. Introduction

Establishing an eHealth environment, organisational, legal, functional, social, ethical, and technical requirements must be met. In that context, security and safety are important challenges, which are influencing user acceptance and specifying the risks a healthcare establishment is faced with. Only professionals contributing to the patient's care, in relation to the information they are interested, shall be allowed to access extracts of personal medical records. Performing security analysis, risk assessment, policy specification, and Continuity of Business (COB) management, etc., narrative or semiformal methods are used. For analysing health information systems and describing their security requirements, many standards, methods, and tools are available; however, these are not supporting the implementation and the enforcement of the security services needed. Defining policies, requirements, procedures, and solutions in a narrative way, the realisation of principal interoperability provides an obstacle, which has to be overcome. The term *principal* is used for any living and non-living entities as thinkable actors comprising physical persons and legal entities, organisations, devices, systems, applications, related components, objects, etc., as introduced in the international standardisation. Regarding the basic principles of security and related models and solutions, see ref. [1]. The answer to meeting the aforementioned challenges is to embed security services into the architecture as one domain's aspect of the architectural components as being introduced in this paper. For that purpose, the advanced architecture and its embedded security services must be described formally as derived below. Services for secure communication between systems and components have been widely specified and implemented. Application security deals with the trustworthiness of the application's behaviour. Therefore, application-related security services have been put at the centre of an architectural approach to security.

2. Related work

In many real implementations, the classical lattice-based mandatory access control (MAC) model has been implemented, sometimes enhanced by an owner-based discretionary access control (DAC) model allowing the owner for privileging others up to the privilege for assigning respective privileges by themselves [2]. Such approaches could not be managed in healthcare establishments such as hospitals or health networks with hundreds or thousands of users.

Over the last decade, the role-based access control (RBAC) paradigm has been developed and stepwise enhanced as the way of managing authorisation and access control (e.g. [3]). Permission assignment is based on the role a principal is assuming during a work session. Early work of the authors about this has been validated and confirmed meanwhile (e.g. [4]). The policy is bound to the role and not to the principal, thereby forming rather stable (static) relationships. For mutually exclusive sets of roles, simple rules, or constraints have been defined. For meeting additional constraints, temporal RBAC and environmental RBAC have been used to manage more complex coarse-grained and rather simple security policies. Richer security policies can be handled by the generalised RBAC defining ordered groups of subject roles, object roles, and environmental roles. Another refinement considers the conditional assignment of permissions depending on context constraints. For considering context-related constraints (e.g. temporal, object-specific, location-related), also other models using first-order predicate logic have also been developed. To integrate organisational regulations including permissions, obligations, prohibitions, or even recommendations, an organisation-based access control model has been introduced [5].

The weakness of all currently available RBAC models is the definition of simplified policies without ways for implementing and controlling them. If some enforcement has been realised, it has been borrowed by the inclusion of assumptions about the underlying technology, then abstracting from application structure and functionality. In any case, security services and functionalities are defined completely different from the application functions, and are therefore not part of the application's behaviour. The only way of combining security services and application functions is the integrated enforcement of security. As a solution, security services have to be designed as an integrative part of the application system's architecture.

3. The challenge for an architectural approach

For providing efficient, high quality care under changed conditions such as demographic development, achievements in medicine and biotechnology, citizens' demands on health services, and finally decreasing insurance funds, new strategies in health policy and health systems have been established. As result, higher specialisations and closer collaborations have been introduced. This must be properly managed in the sense of managed care, shared care, or disease management. Such care plans require advanced communication and co-operation to comprehensively provide and share information and methodologies needed for shared care. Modern Electronic Health Records (EHRs) have to meet requirements enabling system behaviour such as life-long documentation, change resistance, context preservation, flexibility, scalability, interoperability at knowledge level and concept level, legal reliability, etc., as summarised in refs. [6,7]. They provide the core application of any health information system.

For keeping such highly complex systems manageable, the complexity must be reduced and the considerations have to be focused properly. This is done by modelling the reality: first encapsulating interesting parts as components to be considered, and second viewing them according to specific aspects the developer is interested in. Please be aware of the generic use of the term ''component'' here, and thereafter as a piece or ''a part of'' in any domain under any aspect. Following the paradigm introduced in the paper, the component may be constraint for specific purposes or under specific conditions. The language system used to describe structure and behaviour of the system model must be clearly specified regarding the terms used, the semantics behind them, and the rules applied. This is done using meta-languages such as UML, XML, etc.

In all domains and here especially in the domain of health as well as that of security, domain experts use their own language, which is normally non-formal and based on a specific ontology. For integrating all services and all views into the future-proof architecture, the different vocabularies used have to be harmonised or at least mapped properly.

Although *communication* security services are widely established in different domains and even while they are not health specific, it is a very challenging task to provide healthcare-specific *application* security services. This is caused by the aforementioned special sensitivity and the policy-dependency of application security services. This policy is determined by legal, ethical, organisational, functional, social, behavioural, and even personal aspects.

4. Architectural aspects of information systems

Architecture describes structure and function of a system on the basis of fundamental principles. It is the basis for defining components. Its appropriate specification assures future-proof solutions as given in Table 1.

4.1. Future-proof health information system architecture

Therefore, the framework for future-proof health information system architecture must be based on the Generic Component Model developed in the mid-1990s (e.g. [8]). Basis of that architecture are a Reference Information Model (RIM) such as the HL7 RIM [9] and agreed vocabularies such as SNOMED CT^{TM} [10] enabling interoperability. Referenced to them, domain-specific constraint models will be specified which represent domain-specific knowledge concepts, considering both structural and functional knowledge. Policies express rules or

Table 1 Fundamental Principles for future-proof systems

Requirement	Solution
Assuring openness and portability of solution by	Standards-based and model-based approach Separation of platform-independent and platform-specific modelling, i.e. separation of logical and technological view
Assuring flexibility and scalability of solution by	Distribution at Internet level Component orientation
Assuring semantic interoperability of solution by	Definition and installation of reference models and agreed vocabulary/terminology (by referring to standards)
Assuring usability and acceptability of solution by	Definition of domain models (concepts, contexts, knowledge) by using domain expertise (e.g. medical experts, management experts, finance experts, etc.) Definition and implementation of interoperability at service level (concepts, contexts, knowledge)

constraints. The corresponding components have to be established according to all views of the ISO Reference Model 10746-1—Open Distributed Processing (RM-ODP) [11], i.e. enterprise view, information view, computational view, engineering view, and technology view. A view focuses consideration on a single aspect abstracting from all others. The different domain concepts and their view representation is not the task of programmers but of domain experts. For that reason, they will use appropriate expression means such as specific graphical representation (e.g. UML diagrams) or sometimes even verbal templates expressed in XML.

The components can be aggregated to higher level of composition. Contrary to the ISO definition of primitives and composition, in the Generic Component Model at least four levels of composition/decomposition have been defined (Fig. 1).

The aggregation is performed according to content-related or process-related knowledge expressed by logics/algorithms/operations or rules/workflows/procedures/relationships. So, the aggregation of the building blocks ''constraint models'' is controlled by the aforementioned mechanisms or by the communicating or co-operating principal's behaviour. The specification is provided completely at meta-level. Different vocabularies as well as tooling environment and functionality are harmonised by meta-languages like XML Metadata Interchange (XMI) [12,13].

4.2. Principles for model-driven and component-oriented architectures

The crucial idea of the Generic Component Model approach is the walk through the matrix given in Fig. 1. Different enterprise models may be associated with information views, thereby connecting business aspects or alternatively security-related issues. Binding policies to business components, the aggregation allows any type of enterprise model and functionality and any policy ruling the behaviour of components and the resulting system. On the other hand, the resulting model can be implemented in many ways, mixing centralisation and decentralisation of components and services in an absolutely open fashion. The advantage of this approach compared with the normal way of discussion and definitions is the independence between logic and technology as, e.g. described in ref. [14].

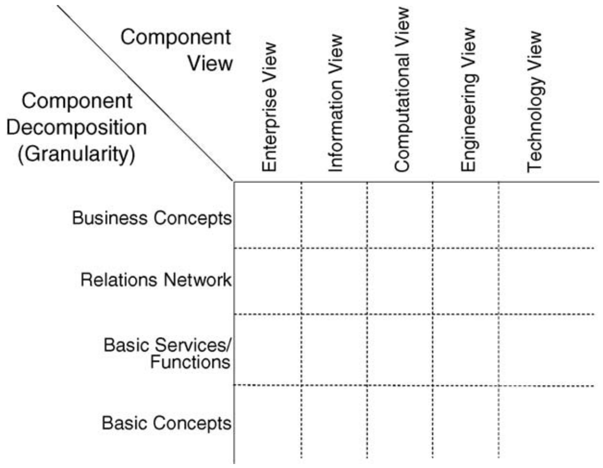


Fig. 1 Generic Component Model.

Information systems containing any patient-related information about observation results and procedures appearing in the context of care are highly sensitive due to the social and personal implications of such information. Security and privacy services to protect the patient's personal medical data are basic requirements, which have to be an integrated part of the architecture. Therefore, the specification and implementation of security services have to follow the above-mentioned paradigms of model-driven architectures [15] and RM-ODP. The paper introduces the basics of formal and generic modelling health-related security and privacy services to establish trustworthy health information systems.

4.2.1. Fundamental principles for future-proof systems

The definition of optimal component size is a challenge for an adequate architecture model. Fine-grained components increase flexibility and scalability of components, but aggravate standardisation. More complex components support standardisation, but decrease scalability and flexibility of solution. To support aforementioned principles, the optimal granularity is that of domain-specific basic concepts, i.e. knowledge concepts, which cannot be subdivided any further without loosing its meaningfulness. Such a component is also called ''atomic component''. Thomas Beale introduced the term ''Archetype'' for describing such components representing a knowledge concept [16].

In particular domains, for important use cases (scenarios) or contexts, the optimal component size may be above the knowledge-related basic concept. If, for example, certain legal basics or sector-specific regulations are ruling security-infrastructural services in a common way, such security infrastructure could be specified and implemented as one complex service. Nevertheless, in any case, attention must be paid to guarantee the re-usability of components in another context (e.g. for locally administering privileges, authorisations, and access control). In case of doubt as well as for providing future-proof solutions, the optimal granularity is that of the basic concept. For any component, the different views of the ISO RM-ODP must be specified [6,7,11].

The health telematics architectural framework bindingly defines and specifies the needed components for prioritised applications as well as next generation services to be introduced. Reference models as well as the vocabulary/terminology definitions are also part of the telematics architectural framework. Furthermore, the telematics architectural framework defines the methodology and tools for specifying further components.

4.2.2. Important security services for future-proof systems

Important basic services and therefore basic components are, for example, identification and authentication services including the infrastructure for establishing and managing such services, and also audit trail, services for assigning rights, duties, functions, roles, etc., as well as time stamping.

Regarding the aforementioned basic services, identifiers must be specified for every entity interacting with a system (what is called principal) as well as every transaction. For identifiers, standards such as the ISO OID [17] must be used. If unique identifiers are missing, or are not applicable for human beings as actors (e.g. for privacy reasons), standardised services such as the CORBA Person Identification Service may be applied [18]. More complex services are, for example, data entry and data retrieval or extracts, respectively.

It is very important to distinguish between services on one hand and mechanisms on the other. For implementing a service, different, sometimes concurrent, mechanisms may often be used.

If an architectural framework has not been defined in the described way, the properties of future-proof systems such as openness, scalability, flexibility, portability, interoperability, etc. cannot be achieved. Furthermore, this definition is the prerequisite for competition, because only the aforementioned principles enable diversity of competing implementations (products and services) while guaranteeing interoperability. Otherwise, dictatorship of companies up to monopoly would follow.

5. Security needs definition

For establishing an environment for safe, secure and reliable communication and co-operation, the security needs and requirements for systems and processes have to be analysed and defined. Even if a system analysis can also be performed using the demonstrated formal modelling approach, traditionally this is done in an informal narrative or in a semi-formal way.

The definition of security needs and requirements can be properly done using a layered model of security concepts, services, mechanisms, functions, and algorithms. A coarse approach for such a layered model is given in Fig. 2 [19]. The algorithm layer provides the essential examples for cryptographic algorithms used for encoding, hashing, or signing information. For details refer, for example, to refs. [1,20].

Alternatively, CEN ENV 13608 ''Health informatics—security for healthcare communication'' [21] provides a Security Concept & Terminology approach, which enables a checklist-type analysis and definition phase. Below, such narrative approach is shortly presented. In the main part of this chapter, however, component-based modelling is described, which is consistent with Chapter 9 of ref. [6].

CEN ENV 13608 renames the services of Fig. 2 to *global security needs*, thereby being restricted to availability, integrity, confidentiality, and accountability. At the next layer, *conceptual security needs* in terms of security objects, security subjects, and the security policy have been established. These *conceptual security needs* include object availability, subject availability, object integrity, subject integrity, transport confidentiality and clearance, sending, delivery as well as exchange accountability.

Each one of the four *global security needs* is refined into a set of *conceptual security needs*, and this refinement process helps to clarify the security needs in terms of a security policy, expressed in terms of security objects, security subjects and the constraints placed upon them as expressed in the security policy. Aspects covered by the security policy at this level include: the third level of the Security Concept & Terminology approach refines the conceptual security needs into *contextual security needs*. Those *contextual security needs* correspond to an even more detailed level of expression of security needs, taking into account the contextual aspects that might influence the security needs (e.g. the granularity level of the transported object, such as data item, message or communication channel).

At high-level, the next paragraphs compare functional requirements for security infrastructure services dealing with centralised authentication and authorisation versus those of distributed authentication and authorisation, also considering

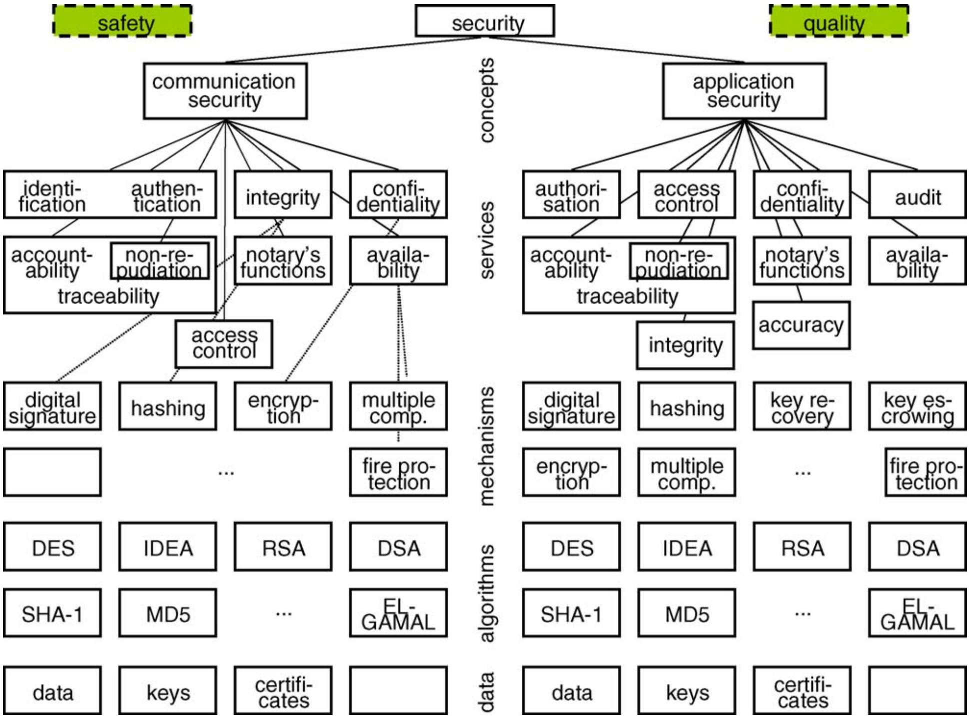


Fig. 2 Layered Security Model.

functional requirements for secure software as well as for security management infrastructure.

5.1. Centralised authorisation infrastructure functional requirements

High-level security requirements for centralised authentication and authorisation define that users must be authenticated once, logically centrally, with centralised authorisation and security management information bases supporting distributed access control decision function architectures. The uniform enterprise-wide fine-grained access control decisions must be maintainable without requiring time consuming and costly modification to individual application systems. The validation of permissions and roles must be validated in a centralised way across the enterprise.

Sign-on and access to network resources must be available from any workstation across co-operating security domains. Security policies must be capable of dynamic implementation, revocation or suspension of user permissions and accounts as determined by proper authority. People must be authenticated and authorised in a security domain supporting custom and personalised healthcare as well as individual privacy decisions (as permitted by policy). Based on flexible access decision rules, user authorisations to multiple systems must be managed, negotiated, and decided without the need to modify individual system account information.

5.2. Distributed authorisation infrastructure functional requirements

The distributed authentication and authorisation infrastructure must provide support for hierarchical security policies, implementation, control, and security management, replacing hard-coded policies with flexible network-based ones and providing ability to support multiple security policies. In that context, a common, reusable, distributed model for authentication/authorisation services, accommodating policies not in heritage/COTS system originally, has to be provided.

The process of user management (authorisation, roles, and authentication) must be centralised and automated by ensuring strong authentication for all users and systems and creating standard security authorisation infrastructures which support a variety of enterprise-wide end-to-end security applications.

End-to-end confidentiality for all communications containing patient-identifiable healthcare information must be provided with fewer-logon with a goal of achieving single sign-on to all applications.

5.3. Secure software infrastructure functional requirements

The authentication and authorisation infrastructure must support the development/application environment needs to reduce amount of security code maintained and developed as well as to simplify security interaction and modelling with application systems. For that reason, scalable security modules incorporating agile interfaces to external systems have to be provided.

To meet users' expectation, and to improve user acceptance, system certification must be speeded up. Furthermore, cost of application security, interfaces/interactions with external systems as well as the number of touch points to the security functions must be reduced. In the same context, security functions for assurance have to be isolated and the security behaviour must be made more visible.

Measures for meeting the requirements are to deal with changes by, e.g. replaceable renewable models and to improve control of versioning and configuration management.

5.4. Security management infrastructure functional requirements

The distributed authentication and authorisation infrastructure security management functions must improve enterprise management of security configuration by, for example, managing them centrally for all applications (update once and centrally). For achieving this goal, both the complexity of managing many systems/interfaces and administrative burden on limited resources must be reduced. In that context, the ability to apply changes has to be speeded up, rapid re-configuration to meet unexpected conditions/threats must be allowed, and provisioning of user security

Table 2 High-level security requirements

Requirement	Description	Security Service
• Enterprise-wide distributed authentication	• Strong authentication for all users and systems/persistent clinical sessions	• Identification and authentication
• Medical Sign-on/Single Sign-on	• Fewer logons with a goal of achieving single sign-on to all VHA applications	
• Healthcare Identifiers	• Enterprise-wide person identifiers	
• Enterprise-wide distributed authorisation	• Enterprise-wide hierarchical security policies, business oriented least privilege and need-to-know access, business partner access, centralised user profiles, policy based access control	• Access control and authorisation
• RBAC and role engineering	• Structural roles and functional roles	
• Break-glass access		
• Federated authorisations		
• Distributed/local access control		
• Health information system audit	• Centralised auditing, processing, and reporting	• Accountability
• Monitoring security function		
• Application audit		
Security management information base protection/integrity	• End-to-end confidentiality and integrity for security control, messaging and management data	• Confidentiality
• At rest	• Digitally signed security credentials	• Data integrity
• In transit		• Digital signature
• Centralised user and system security management	• Configuration management, provisioning, test, and operations	• Security management
	• Enterprise-wide security management information bases/operation centres	• Availability
• Software/hardware trust	• Secure software development life cycles	• Assurance
• Common secure software development environments	• Certification	
	• Standard interfaces to security function	

configuration must be simplified. Finally, the assurance of security function through certification has to be provided.

The aforementioned functional requirements must be met by appropriate security services fulfilling all the security needs derived from legal requirements, regulations, policies, business processes, and social behaviour as shown in Table 2.

Before the architecture integrated security services will be formally modelled, an architectural overview about health information system components and their relationship to security services derived from the requirements analysis will be presented in Fig. 3 based on the Veterans Health Administration (VHA) security architecture approach [22].

The figure demonstrates relationships between users and applications mediated by authentication components using different possible authenticators and infrastructure services (e.g. Public Key Infrastructure— PKI), by an authorisation component (e.g. Privilege Management Infrastructure—PMI) and a provisioning component (e.g. role engineering) [4].

6. Generic models

After analysing security needs and requirements in a structured but informal way, system design will now be performed according to the formal modelling approach introduced in Section 4.

Privilege management and authorisation may be assigned to individual actors or to groups of individual actors playing the same role. As said, actors interacting with system components are called principals, which can be a human user, a system, a device, an application, a component, or even an object. In order to obtain the above described structure and functionality, there are a number of models, mechanisms, processes, objects, etc. needed, which have to be looked upon in greater detail.

Regarding privilege management and access control management, two basic class types must be dealt with: Entities and Acts.

Entities can be specialised to principals, policies, documents, and roles. Specialisations of an Act relevant in the paper's context are, e.g. Policy Management, Principal Management,

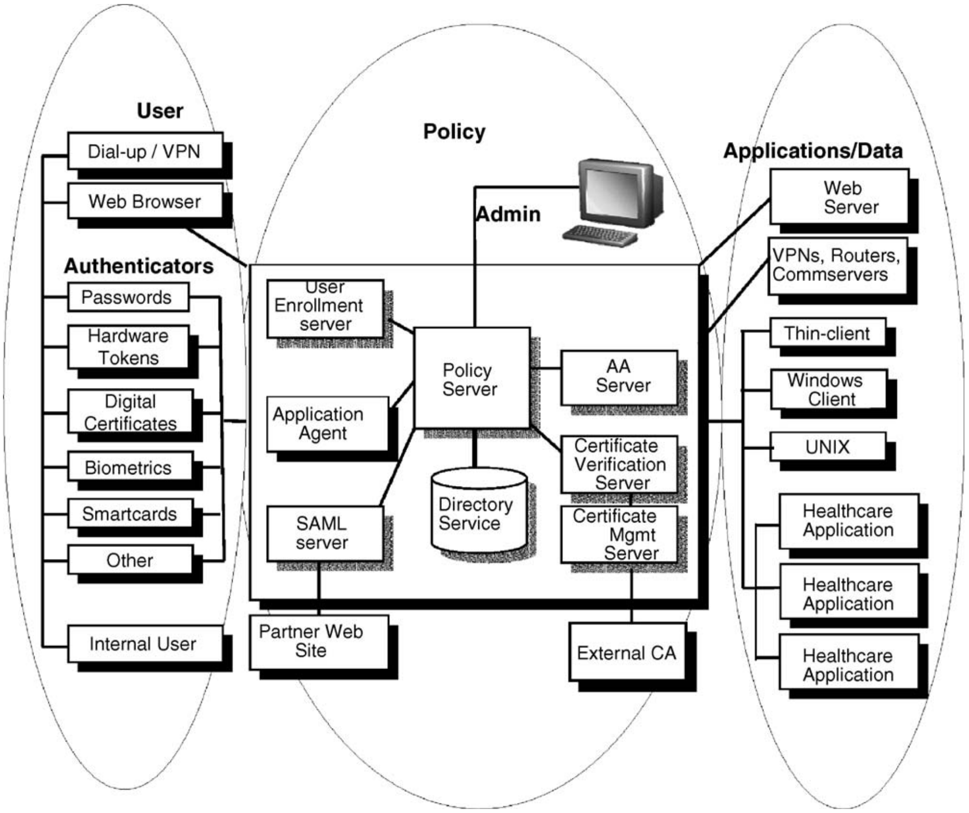


Fig. 3 VHA identity and access management.

Privilege Management, Authentication, Authorisation, Access Control Management, and Audit.

The acts mentioned are needed to enable the described security services. A series of static and dynamic models must be introduced to describe the entities and to define how the activities will be performed. Following models will be considered in more detail:

- Domain Model;
- Delegation Model;
- Control Model;
- Document Model;
- Policy Model;
- Role Model;
- Information Distance Model;
- Authorisation Model;
- Access Control Models.

Even if not explicitly mentioned in the corresponding sections, all those models follow the architectural principles expressed in the component paradigm and the Generic Component Model [6,7] introduced in Section 4.

For enabling future-proof EHR systems, all specifications must be kept open, platform-independent, portable, and scalable. Therefore, the models provided will be described at meta-model level and at the model level, keeping the instance level out of consideration. For expressing systems in such a way, specific languages and meta-languages are used such as UML and XML including a means for transfer from one vocabulary to another one.

Regarding UML, UML itself, UML profiles, and all of the different diagrams needed have been deployed. Regarding XML, several specifications within the XML standard set will be used.

All models being used establish a specific kind of constraint forming constraint models. This concerns any thinkable services or view on systems. A model is a simplified view at the reality according to special concepts. The language to be used for graphical models is UML and MOF [15]. The language for verbal models is the XML standard set.

It is expected that many documents will be expressed using XML. The structure for such a document is defined in a document type definition (DTD) or an XML schema instance. A privilege policy may act directly on the XML elements (e.g. by comparing attributes in an authorisation certificate to elements in the document).

6.1. Domain Model

An information domain consists of a set of data, users, and an information security policy linking both. Each defined information domain has a unique identification. Within an information domain, all information objects exist at the same level of sensitivity. Members of a domain may have different security attributes, such as read, write or execute permissions on information objects. Systems or networks of systems do not bound information domains. An information domain's objects may reside in multiple systems.

To keep (complex) information systems that support shared care manageable and operating, principal-related components of the system are grouped by common organisational, logical, functional, and technical properties into domains. Following Object Management Group's (OMG) definition, this could be done for common policies (policy domains), for common environments (environment domains), or common technology (technology domains) [23]. Any kind of interoperability internally to a domain is called an intra-domain communication and co-operation, whereas interoperability between domains is called an inter-domain communication and co-operation. For example, communication could be realised between departments of a hospital internally to the domain *hospital* (intra-domain communication), but externally to the domain of a special department (inter-domain communication). Regarding security requirements, security policy domains are of special interest. Real world systems are most likely composed of multi-domain information objects that cut across different information contexts.

A domain is characterised at least by a domain identifier, a domain name, a domain authority identifier, a domain authority name, and a domain qualifier (see Table 3). The data type definition provided resembles to the HL7 Version 3 Data Type Definition [24].

A policy describes the legal framework including rules and regulations, the organisational and administrative framework, functionalities, claims and objectives, the principals involved, agreements, rights, duties, and penalties defined as well

Table 3 Security policy domain attributes

Attribute	Type	Remarks
domain_identifier	SET <OID>	SET of ISO ObjectIdentifier
domain_name	BAG <EN>	Bag of EntityName
domain_authority_ID	OID	ISO ObjectIdentifier
domain_authority_name	ST	String
domain_qualifier	CS	CodedSimpleValue

as the technological solution implemented for collecting, recording, processing, and communicating data in information systems. For describing policies, methods such as policy templates or formal policy modelling might be deployed.

A domain is generic in nature and can be developed according to the component paradigm. A domain might consist of sub-domains (which will inherit and might specialise policies from the parent domain). The smallest-scale domains might be an individual workplace or a specific component within an information system. Domains can be extended into super-domains, by chaining a set of distinct domains and forming a common larger-scale domain for communication and co-operation.

This co-operation between domains requires the definition of a common set of policies that applies to all of the collaborating domains. It must be derived from all of the relevant domain-specific policies across all of those domains. These common policies are derived (negotiated) through a process known as policy bridging (see Fig. 4) [12]. The eventual agreed policies need to be documented and signed by all of the domain authorities. Ideally, this whole process will be capable of electronic representation and negotiation, to permit real-time electronic collaboration to take place within a (pre-agreed) permitted and regulated framework. The policy negotiation or verification would then take place at every service interaction.

Users may perceive a collection of objects from different information domains as a single object. This compound object is referred to as a multi-domain information object. Multi-domain information objects must be subject to a common security policy. The security policy must state the privileges that a user must have to view, print, create, delete, or transfer multi-domain information objects between information systems.

Middleware components can enable interoperability through direct invocation (middleware communication services) or chained invocation (including middleware application services). The latter is characterised by different models of delegation (see Section 6.7).

The general purpose of communication is the provision of services to a client requesting these services. Most of the services have to be provided by the functionality of the healthcare information system often combined with human users' interactions. Such application services are end-system services, limited to communication services, and do not provide additional application functions. Therefore, application security services are restricted to the communicating principals' domain.

Middleware concepts are increasingly used in the new(er) versions of healthcare information systems. In the corresponding model, not only are principals involved, but the middleware can also provide requested functionality and application security services. Such architecture can be represented by chains of different domains.

From the security point of view, a domain ensuring intra-domain communication according to its own policy is commonly considered to have need of protection only at its boundary to external domains with their specific policies (or even the policy-free Internet domain). This is done by firewalls, proxy

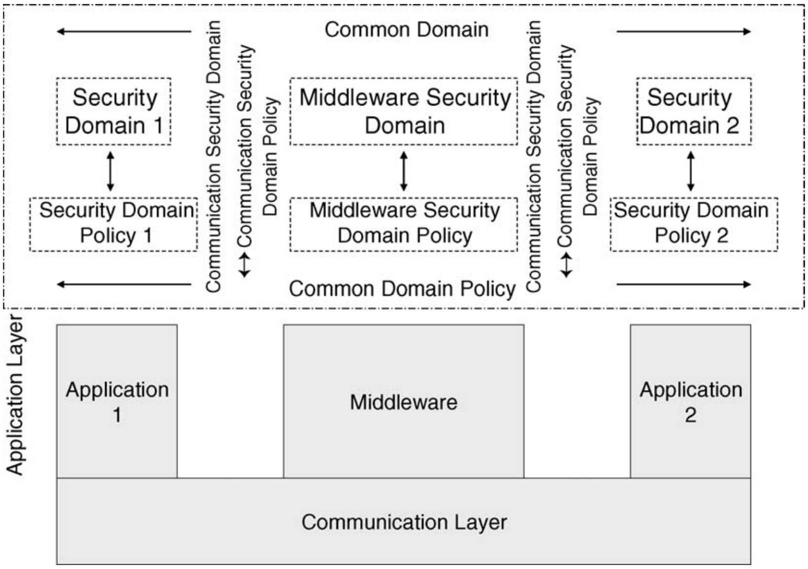

Common Domain
Security Domain 1
Security Domain Policy 1
Communication Security Domain
↔ Communication Security Domain Policy
Middleware Security Domain
Middleware Security Domain Policy
Communication Security Domain
↔ Communication Security Domain Policy
Security Domain 2
Security Domain Policy 2
Common Domain Policy
Application Layer
Application 1
Middleware
Application 2
Communication Layer

Fig. 4 Policy bridging.

servers, etc. Regarding the external environment, a domain is therefore often considered closed. The internal domain is mistakenly assumed to be secure, often neglecting internal threats and attacks which represent the majority of all security attacks.

Regarding the specific requirements and conditions of healthcare, the underlying security model must consider the whole spectre of security services and mechanisms, which can be accomplished by secure micro-domains.

6.2. Document Model

Processes, entity roles, etc. must be documented and signed expressing the particular relations between entities and processes. The combination of processes and relations leads to multiple signatures (e.g. in the case of delegation).

Advanced systems use the cryptographic message syntax to support multiple signatures on a document. Each signature is computed over the document content and optionally a set of signed attributes specific to the particular signature. These attributes may include timestamps, signature purpose, and other information.

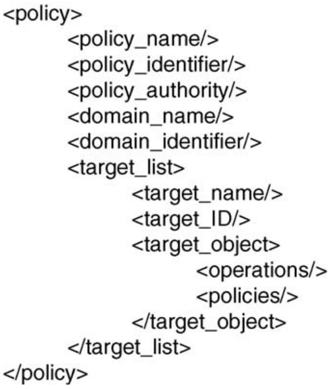
A Document Model also follows the component paradigm, enabling composition, and decomposition. This is restricted to its structural behaviour, not concerning functional behaviour, however. Contrary to an object which is implementing a service, a document has no operation, but must be processed by components, sometimes also including human users' activities.

6.3. Policy Model

A security policy is the complex of legal, ethical, social, organisational, psychological, functional, and technical implications for assuring trustworthiness of health information systems. A policy is the formulation of the concept of requirements and conditions for trustworthy creation, storage, processing, and use of sensitive information. Referring to the Generic Component Model and its relation to the RM-ODP, a policy is a set of constraints applied to components and their associations. A policy can be expressed

- verbally unstructured;
- structured using schemata or templates;
- formally modelled.

For interoperability reasons, a policy must be formulated and encoded in a way that enables its correct interpretation and practice. Therefore, policies have to be constrained regarding syntax, semantics, vocabulary, and operation of policy documents, also called policy statements or policy agreements (agreements between the partners involved).



```
<policy>
        <policy_name/>
        <policy_identifier/>
        <policy_authority/>
        <domain_name/>
        <domain_identifier/>
        <target_list>
                <target_name/>
                <target_ID/>
                <target_object>
                        <operations/>
                        <policies/>
                </target_object>
        </target_list>
</policy>
```

Fig. 5 Policy body template example.

One common way to express constraints is the specification of user-defined schemata such as XML schemata. This schema should be standardised for interoperability purposes mentioned above. Fig. 5 presents a simple XML instance for a security policy statement body.

To reliably refer to a specific policy, the policy instance must be uniquely named and identified via a unique policy ID. The same is true for all the policy components such as domains, targets, operations, and their policies, which have to be named and uniquely identified too. As any other component, also policy components can be composed or decomposed according to the Generic Component Model.

A policy is characterised by policy identifier, policy name, policy authority ID, policy authority name, domain identifier, domain name, target list, target identifier, target name, target object, operations allowed, and related policies. Additionally, other constraints such as validity time period or authority's signature may be attributed.

For readability reasons, domain-related attributes have been included in Table 4 even if policy inherits from domain. The provided data type definition resembles to the HL7 Version 3 Data Type Definition [24].

Nowadays, verbally unstructured policy descriptions or at best structured policy statements using schemata or templates are being used.

According to the Generic Component Model [6,7], the Policy class can be specialised (decomposed) as Basic Policy, Meta Policy and Composite Policy classes, furthermore, refining them up to the granularity of ''atomic components'' (see also Section 4.2.1) as shown in Fig. 6 below [15]. Constraints have to be expressed using the OMG's

Table 4 Security policy attributes (not completed)

Attribute	Type	Remarks
policy_identifier	SET <II>	Set of InstanceIdentifier
policy_name	CS	CodedSimpleValue
policy_authority_ID	OID	ISO ObjectIdentifier
policy_authority-name	ST	String
domain_identifier	SET <OID>	Set of ISO ObjectIdentifier
domain_name	EN	EntityName
target_list	LIST <INT>	List of INT
target_ID	SET <II>	Set of InstanceIdentifier
target_name	EN	EntityName
target_object	II	InstanceIdentifier
operation_code	CE	CodedWithEqivalents
policies	CD	ConceptDescription

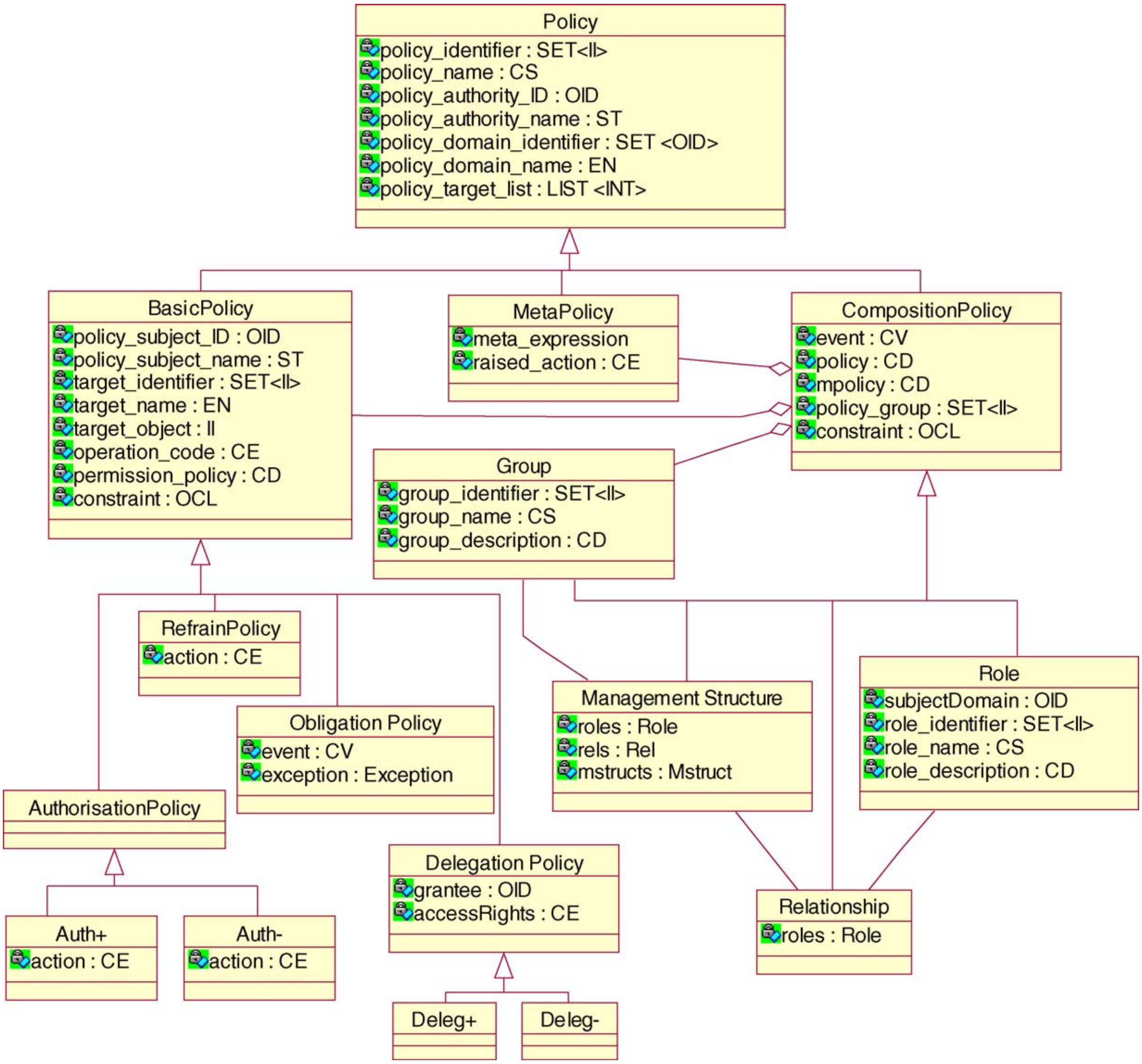


Fig. 6 Security policy base-class diagram.

Table 5 PONDER basic policy types [26]

Basic policy type	Purpose	Content
Authorisation policies	Define permitted actions	Subject (except in roles), target, action
Obligation policies	Event-triggered and define actions to be performed by manager agents	Subject (except in roles), action, event
Refrain policies	Define actions the subjects must refrain from performing	Subject (except in roles), action
Delegation policies	Define what authorisations can be delegated to whom	

Object Constraint Language (OCL), which is part of the UML standard [25].

The specialisations of the Composite Policy abstract class are interrelated in a complex way, which has been indicated in outlines for better readability as simple association. In the next two tables, the fine-grained policy specifications according to the PONDER model will be explained (Tables 5 and 6).

Beside the specific PONDER policy language used as an alternative to our OCL proposal, also other specific policy languages have been introduced. The most important example for such languages is XACML, an XML-based language for policy and access control definitions standardised by OASIS [27]. The weakness of the XACML approach is the fixation to the XML meta-language instead of following the OMG UML/OCL definitions. Nevertheless, a possible grammar transformation from one meta-language into another using XMI should be mentioned [13].

Another way for policy decomposition has been provided by the OMG's Security Services Specification distinguishing the following policies [23]:

- invocation access policy implementing access control policy for objects,
- invocation audit policy controlling event type and criteria for audit,
- secure invocation policy specifying security policies associated with security associations and message protection.

Regarding requirements for different object types, the following have been defined:

- invocation delegation policy,
- application access policy,
- application audit policy,
- non-repudiation policy.

Health information systems such as the EHR should at minimum have a Patient policy, an Enterprise policy, policies defined by laws and regulations, and one policy per structural role as well as one policy per Functional Role. Patient's consent is a special policy.

Every creation or modification of, or access to, an EHR component must be covered by one or more policies. The reference model of the EHR Extract includes a policy ID attribute within the Record Component class to permit references to such policies to be made at any level of granularity within the EHR hierarchy. The policies that apply specifically to an EHR may be included within the EHR Extract, eventually including any bridged policies.

Another way for expressing security policies is using a formal language of first-order logic. An interesting example for applying this grammar to health-related security policies has been given by Cohen [28].

6.4. Role Model

Roles may be assigned to any principal. Principals are the actors in healthcare. Therefore, roles are associated to actors and to acts. For managing relationships between the entities mediated by an activity, two different roles have to be defined: organisational roles at the entity's side and functional roles at the act's side.

Table 6 PONDER composite policy types [26]

Composite policy type	Purpose
Groups	Define a scope for related policies to which a set of constraints can apply
Roles	Define a group of policies (authorisation, obligation, and refrain policies) related to positions within an organisation
Relationships	Define a group of policies pertaining to the interactions between a set of roles

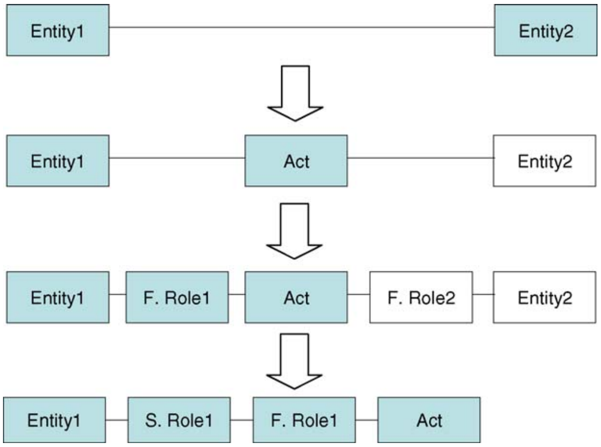


Fig. 7 Development of the functional Role Model.

6.4.1. Healthcare related roles

In general, two types of roles can be distinguished: rather static structural roles and highly dynamic functional roles. Structural roles reflect the structural aspects of relationships between entities, whereas functional roles reflect functional aspects of relationships between entities. These relationships are presented in Fig. 7 qualifying the associations by association classes. According to the HL7 approach and contrary to the UML specification, the association class is not bound to the association but has been represented by a class type in this figure, however.

Considering both structural roles and functional roles in the same context, structural roles provide the prerequisites/competences for entities to perform interactions (an Act) within their specific functional roles. Qualifications, skills, etc. are influencing both the assignment of the structural roles and the performance of activities according to their functional roles (Fig. 9). Possible examples for structural and functional roles of healthcare professionals are given in Table 7.

Table 7 Examples for structural and functional roles

Structural role examples	Functional role examples
Medical director	Caring doctor (responsible doctor)
Director of clinic	Member of diagnostic team
Head of the department	Member of therapeutic team
Senior physician	Consulting doctor
Resident physician	Admitting doctor
Physician	Family doctor
Medical assistant	Function-specific nurse
Trainee	
Head nurse	
Nurse	
Medical student	

6.4.2. Functional role model

Regarding the healthcare business process, functional roles can be defined in levels of authorisations and access rights in the following generic way reusing slightly changed definitions established in the Australian HealthConnect Project [29], cross-referenced against other works:

- subject of care (normally the patient),
- subject of care agent (parent, guardian, carer, or other legal representative),
- responsible (personal) healthcare professional (the healthcare professional with the closest relationship to the patient, often his GP),
- privileged healthcare professional
  - ○ nominated by the subject of care,
  - ○ nominated by the healthcare facility of care (there is a nomination by regulation, practice, etc.),
- healthcare professional (involved in providing direct care to the patient),
- health-related professional (indirectly involved in patient care, teaching, research, etc.),
- administrator (and any other parties supporting service provision to the patient).

This list fixes the set of functional roles applied to manage the creation, access, processing, and communication of health information. The formal model of functional roles is presented in Fig. 8.

Another way for grouping functional roles according to the relation to the information created, recorded, entered, processed, stored, and communicated could be: Composer, Committer, Certifier, Authoriser, Subject of information, Information provider.

Another approach for structuring functional roles related to information and its use complying with the European Data Protection Directive [30] and the related ISO CD 22857 ''Health Informatics—

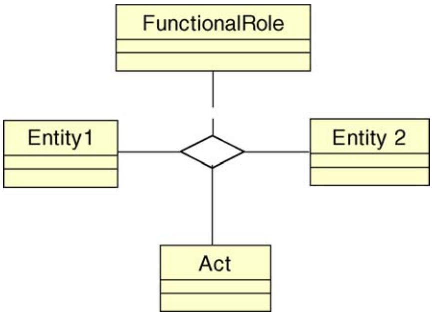


Fig. 8 Functional role model.

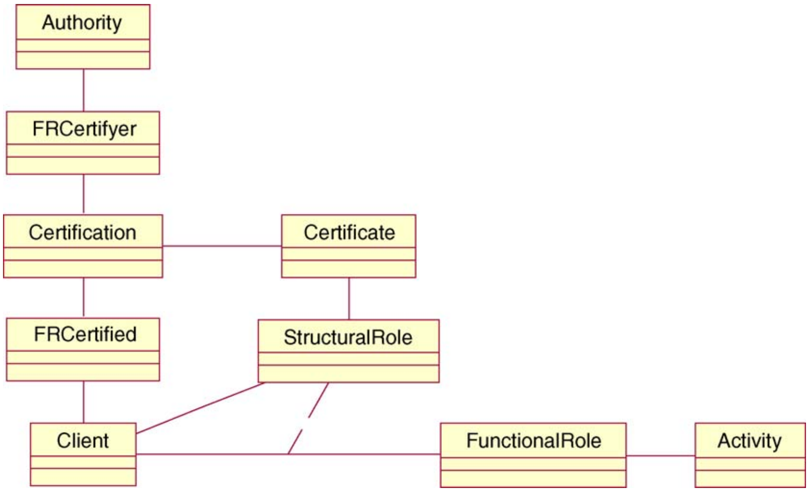


Fig. 9 Establishment of a structural role within an Act according to specific functional roles.

Guidelines on data protection to facilitate transborder flows of personal health information'' [31] has been introduced through the Information Distance Model [32].

6.4.3. Structural role model

Structural (or organisational) roles are also called static roles which place people in the organisational hierarchy as belonging to categories of healthcare personnel warranting differing levels of access control[1]. Organisational roles allow users to participate in the organisation's workflow (e.g. tasks) by job, title, or position but do not specify detailed permissions on specific information objects. Static roles allow a user to ''connect'' to a resource but do not grant authorisations. Some role group examples include: Physician, Pharmacist, Registered Nurse Supervisor, and Ward Clerk. Static roles may be found as non-critical certificate extensions entries to an X.509 certificate as specified in ASTM 2212-02 [34]. In ref. [12], the term ''role groups'' is used for organisational or structural roles. Because the RM-ODP-based approach of different levels of granularity, grouping in the sense of generalisation is applied to every component (also functional roles can be grouped to role groups). Therefore, the term ''structural role'' or ''organisational role'' defined in the ISO standards framework seems to be more appropriate.

Another critical aspect of terminologies used is the term permission to an Act. Harmonising with the HL7 RIM, ''organisational roles'' for HL7s ''roles'' and ''functional roles'' for HL7s ''participation'' to perform ''acts'' have been introduced. Resulting from a different approach to security services compared to the approach of architectural components used in this paper, permission with encapsulated operations and objects instead of the activity is used in NIST [35] and VA [22]. New permissions will lead to a new role. Our approach of policies bound to all basis classes to express conditions and rule on the one hand and permissions on the other hand is more promising and more consistent managing everything in the same generic way.

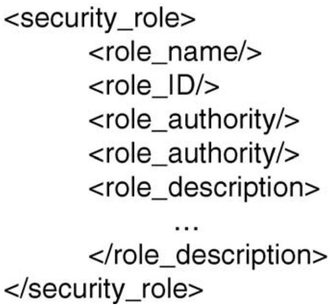
An Entity—Entity relationship may concern a contracting act resulting in a contract between entities playing specific functional roles (see below). The contract could define the structural role of being, e.g. a head physician. Another example may concern an Entity—Entity relationship for education resulting in a special qualification as well as a certificate certifying this qualification as a structural role.

These structural roles constraints another Entity—Entity relationship influencing the functional role played by the entities involved in an activity. The establishment of a structural role is provided within an act between entities according to specific act-related functional roles. Fig. 9 gives an example for the provision of a structural role certificate issued by a certification authority to a client to be certified. In this figure, the StructuralRole class has been drawn according to the UML specification for association classes. Following the HL7 presentation way of Fig. 7, the StructuralRole class (in HL7's RIM called Role) would be in line with the class types Entity, FuntionalRole (in HL7's RIM called Participation) and Act forming the four basis classes of the RIM

[1] See ASTM E1986—98 for a listing of healthcare personnel that warrant differing levels of access control [33].

Table 8 Role attributes

Attribute	Type	Remarks
role_identifier	SET <II>	Set of InstanceIndentifier
role_name	CS	CodedSimpleValue
authority_identifier_ID	OID	ISO ObjectIdentifier
authority_identifier_name	ST	String
role_description	CD	ConceptDescription



```
<security_role>
        <role_name/>
        <role_ID/>
        <role_authority/>
        <role_authority/>
        <role_description>
                ...
        </role_description>
</security_role>
```

Fig. 10 Role specification.

beside its relationship classes RoleRelationship and ActRelationship.

6.4.4. Generic role specification

Fig. 10 describes a role using XML.

Additionally, administration constraints may need to be enforced. For example, the separation of duties may be introduced as a widely used authorisation constraint.

Because the role concept is deployed in many different contextual relationships such as the professionals' administration, certification procedures, roster management, etc., the Role Model and its deployment has been defined as separate ISO TS 21298 ''functional and structural roles'' currently under development under the lead of the main author [36] (Table 8).

6.4.5. Role engineering

Role engineering names the process of defining and assigning roles. The role engineering process has been investigated in details by Neumann and Strembeck [4].

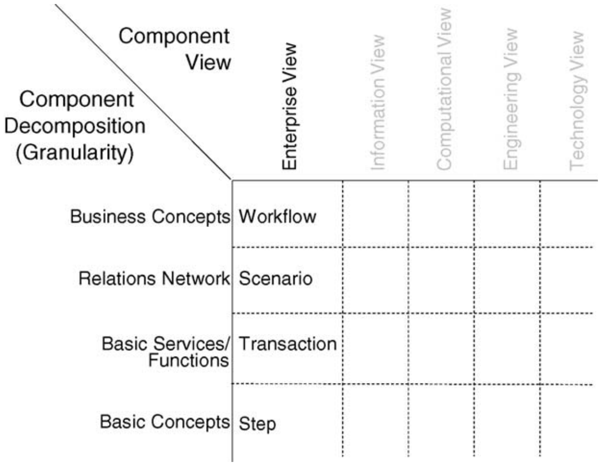


Fig. 12 Workflow analysis for deriving functional roles.

Fig. 11 gives a high level view on the role engineering process expressed in a state diagram.

For deriving appropriate functional roles, the workflow has to be considered, analysing its fine-grained structure. As the workflow and its scenarios are represented in the RM-ODP Enterprise View using the Generic Component Model (Fig. 1), this view has to be deployed as shown in Fig. 12.

6.5. Authorisation Model

Authorisation deals with granting privileges and assigning permissions. Following, the assignment

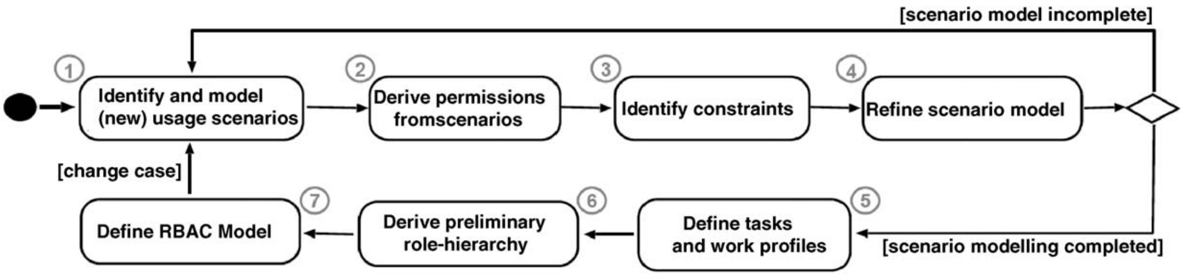

[scenario model incomplete]
1
Identify and model (new) usage scenarios
2
Derive permissions fromscenarios
3
Identify constraints
4
Refine scenario model
[change case]
7
Define RBAC Model
6
Derive preliminary role-hierarchy
5
Define tasks and work profiles
[scenario modelling completed]

Fig. 11 High level view on the role engineering process [4].

stuff as well as privilege management will be shortly discussed.

6.5.1. Role and privilege assignment

Roles provide a means to indirectly assign privileges to individuals. Individuals are issued role assignment certificates assigning one or more roles to them by role attributes. Privileges are assigned to a role, by role specification certificates, rather than to individuals. The indirect assignment enables the privileges assigned to a role to be updated without impacting the certificates that assign roles to individuals. Role assignment certificates may be attribute certificates or public-key certificates. Role specification certificates cannot be public-key certificates, but must be attribute certificates. If role specification certificates are not used, the assignment of privileges to a role may be done through other means (e.g. they may be locally configured at a privilege verifier).

The following scenarios are all possible:

- Any number of roles can be defined by any Attribute Authority (AA),
- The role itself and the members of a role can be defined and administered separately, by different AAs,
- Role membership, just as any other privilege, may be delegated,
- Roles and membership may be assigned any suitable lifetime.

If the role assignment certificate is an attribute certificate, the role attribute is contained in the attributes component of the attribute certificate. If the role assignment certificate is a public-key certificate, the role attribute is contained in the subjectDirectoryAttributes extension. In the latter case, any additional privileges contained in the public-key certificate are privileges that are directly assigned to the certificate subject. Thus, a privilege asserter may present a role assignment certificate to the privilege verifier demonstrating only that the privilege asserter has a particular role (e.g. ''manager'' or ''purchaser''). The privilege verifier may know a priori, or may have to discover by some other means, the privileges associated with the asserted role in order to accept/reject/modify a request. The role specification certificate can be used for this purpose.

A privilege verifier must have an understanding of the privileges specified for the role. The assignment of those privileges to the role may be made within the Privilege Management Infrastructure (PMI) by a role specification certificate or outside the PMI (e.g. locally configured). For role privileges asserted in a role specification certificate, mechanisms for linking that certificate with the relevant role assignment certificate for the privilege asserter are provided in this respective specification. The issuer of the role assignment certificate may be different from the issuer of the role specification certificate, and these certificates are administered (e.g. creation, expiration, revocation) entirely separately. The same certificate (attribute certificate or public-key certificate) can contain role assignment certificate as well as the assignment of other privileges directly to the same individual. However, a role specification certificate must be a separate certificate.

Note: The use of roles within an authorisation framework can increase the complexity of path processing, because such functionality essentially defines another delegation path which must be followed. The delegation path for the role assignment certificate may involve different AAs and may be independent of the AA that issued the role specification certificate.

The general privilege management model consists of three entities: the object, the privilege asserter, and the privilege verifier. For deciding access to resources, both the assignment of object security attributes and the location and deployment of the appropriate policy must be realised. Based on database access control models, Fig. 13 shows a general privilege management model [2].

It should be mentioned that there are three principle decisions made in the privilege management context:

- request authorised,
- request denied,
- request modified.

Credentialing, privileging, and authorisation are performed by connecting roles to policies.

A more detailed Authorisation Model is given in Fig. 14.

6.6. Control Model

The Control Model illustrates how control is exerted over access to a sensitive object operation. Access control is the process which determines if a Claimant's privileges permit it to access a service provided by a Target component. In this context, access is broader than acquiring some data. It might refer to any service offered by a Target component (e.g. deletion, computation, transfer). There are four components in the model: the Claimant, the Verifier, the Target, and the control policy (see Fig. 15). The Claimant has privilege attributes, contained in an attribute certificate. The Target has sensitivity attributes, which may be contained in a

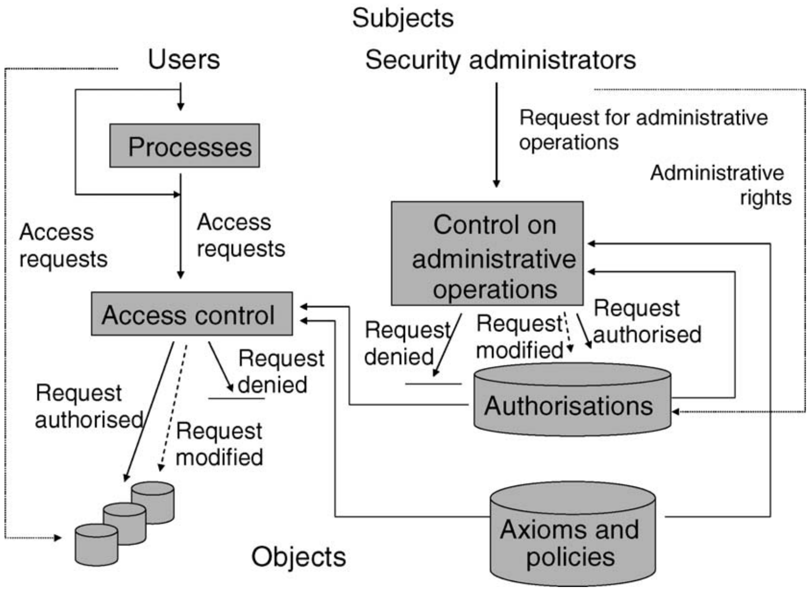


Fig. 13 Privilege Management and Access Control Model (after [2], changed).

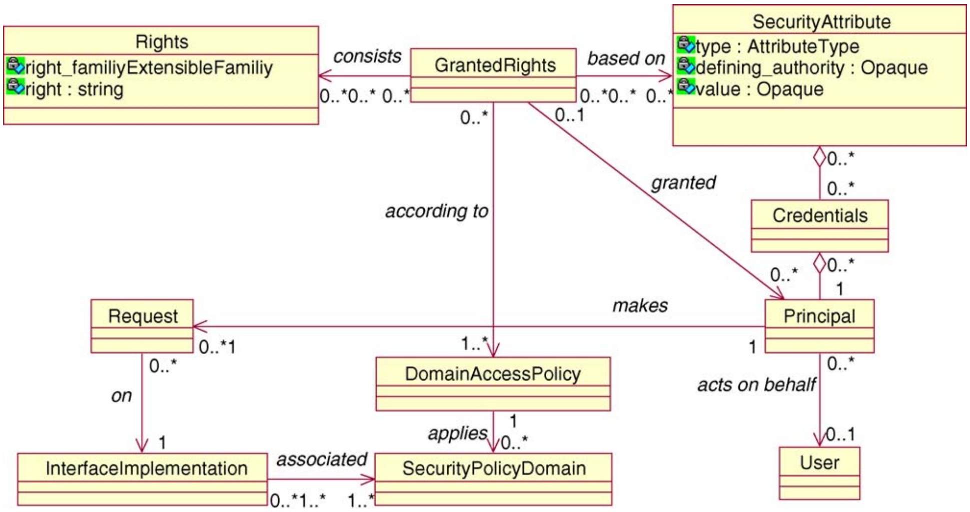


Fig. 14 CORBA Authorisation Model [37].

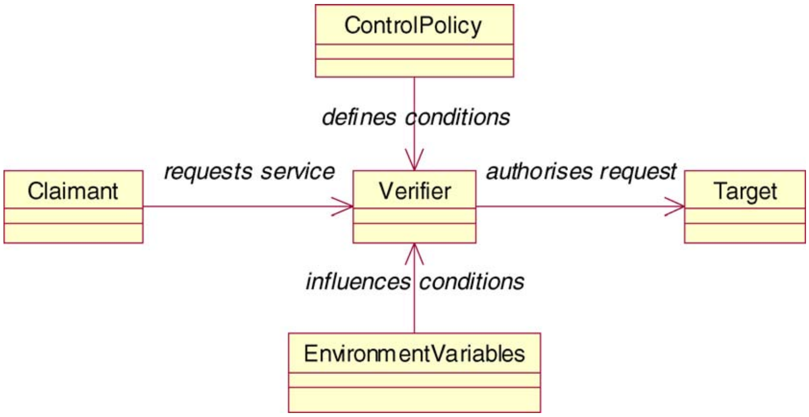


Fig. 15 Control Model.

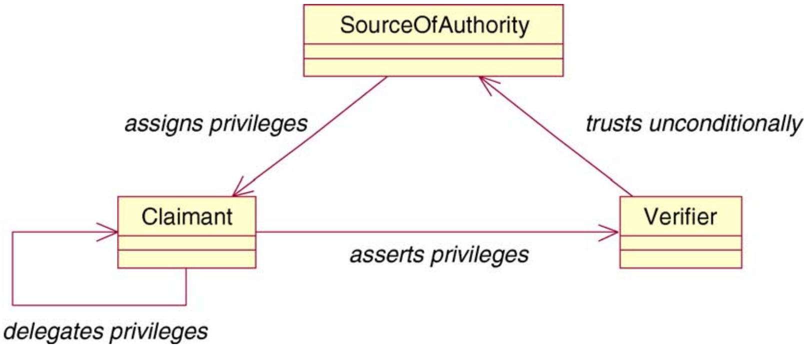


Fig. 16 Delegation Model.

security label, attribute certificate, or in a local database. The techniques described here enable the Verifier, who may be the owner of the Target or an independent authority, to control access to the Target by the Claimant in accordance with the control policy and optionally taking other environmental variables or components (Environmental-Variables in Fig. 15) into account (e.g. local time).

The Claimant's privileges are typically encapsulated in its attribute certificate. This may be presented to the Verifier in the service request (push strategy), or it can be distributed by some other means, such as via a Directory (pull strategy). The control policy must be protected for integrity and authenticity, for this purpose it may also be combined with the Claimant's privilege in its attribute certificate. Normally, however, it will be declared separately.

The Claimant may be an Entity identified by a public-key certificate, or an executable object identified by the digest.

The generality of this model makes the names of its parts appear somewhat abstract. However, with suitable instantiations, it can be applied to all thinkable real situations.

6.7. Delegation Model

In addition to the Control Model, there is a need for a delegation model. There are three components of the delegation model: the Verifier, the Source of Authority, and the Claimant (see Fig. 16).

The Verifier endows an Entity known as the Source of Authority with unlimited privilege. The Source of Authority is a special type of Attribute Authority. It delegates privilege to Claimants by issuing attribute certificates. The Claimant asserts its delegated privilege by demonstrating its identity. This can be done by proving its knowledge of a private key whose public counterpart is contained in a public-key certificate referenced by an attribute certificate which includes the claimed privilege. In the case of an executable object, it may alternatively be done by demonstrating that the digest is the same as the ''owner'' value of an attribute certificate which includes the claimed privilege.

Optionally, the Claimant may delegate its privilege to another Claimant. The Verifier must confirm that all entities in the delegation path possess sufficient privilege to access the Target requested by the direct Claimant.

The Source of Authority may also process a request from an entity to delegate its privilege by issuing an attribute certificate to another entity. However, this process is outside the scope of the aforementioned standard.

The Claimant and the Verifier may be entities in different security domains. In such cases, the Source of Authority may be located in the Verifier's domain, and a continuous section of the delegation path, which includes the direct Claimant, shall be in the other security domain.

The delegation path is distinct from the certificate validation path used to validate the public-key certificates of the entities involved in the delegation process. However, the quality of authenticity offered by the public-key certificate validation process must be commensurate with the sensitivity of the Target being protected.

Specifying interoperability between distributed objects or components, respectively, the Object Management Group has defined an alternative delegation model within its CORBA Security Services Specification [23] (Table 9). In an object system, a client calls on an object to perform an operation, but this object will often not complete the operation itself. So it will call on other objects to do so. This will usually result in a chain of calls on other objects.

In privilege delegation, the initiating principal's access control information (i.e. its security attributes) may be delegated to further objects in the chain to give the recipient the rights

Table 9 Delegation schemes (after OMG [23])

Intermediate performs	Target	Constraints
1. One method on one object		
2. Several methods on one object		
3. Any method on:	a. One object	
	b. Some object(s)	Target restrictions
	c. Any object	No target restrictions
	Using No privileges	
	A subset of the initiator's privileges	Simple delegation
	Both the initiator's and its own privileges	Composite delegation
	Received privileges and its own privileges	Combined or traced delegation, depending on whether privileges are combined or concatenated
	During some validity period	Part of time constraints
	For a specified number of invocations	Part of time constraints

to act on its behalf under specified circumstances.

Another authorisation scheme is reference restriction where the rights to use an object under specified circumstances are passed to the recipient as part of the object reference. Reference restriction is not included in the corresponding CORBA specification.

The following terms are used in describing delegation options within a chain of object invocations (call chain):

- *Initiator*—the first client in a call chain,
- *Final target*—the final recipient in a call chain,
- *Intermediate*—an object in a call chain that is neither the initiator nor the final target,
- *Immediate invoker*—an object or client from which an object receives a call.

Communication of health information is frequently connected with a supplier chain performing this activity (e.g. involvement of secretaries, clerks, service departments but also any other principals). This delegation model must be used for any such chaining of services.

6.8. Access Control Model

Access control is a process which occurs in different ways in both communication security services and application security services. If in the former case of logically and physically accessing a communicating and co-operating principal, access can be allowed or prohibited. In the latter case, the service requested and provided by the invoked component must be specified. Therefore, the outcome of the request may be allowing, denying, or changing the access requested. This scenario may be demonstrated for an EHR system, where a health professional might be allowed to access the EHR system, but not the EHR (components) of a specific patient he is not caring for. On the other hand, professionals caring for that patient may have different rights according to their functional role, possibly influenced by their structural role (e.g. nurse versus doctor).

The patient's right to specify who is allowed to access his or her information is expressed in the special policy of the patient's consent. It defines one rule set among others to be considered. The priority of competing rule sets (policies) is defined by other regulations or even legislation (policies).

Fig. 17 introduces the Access Control use case whereas Fig. 18 shows its refinement for the Resource Access Decision (RAD) process in relation to the specification provided by the OMG Healthcare Domain Task Force (HDTF) in its Resource Access Decision Service (RADS) [37].

Harmonising the role models specified before, and advanced access control models such as the proposed NIST Standard Role-Based Access Control [35], Fig. 19 presents an adapted Role-Based Access Control Schema.

Because policies define rules for association and use of any class, all model components are connected to policies expressing the constraints for those components.

Each model component is defined by the subcomponents:

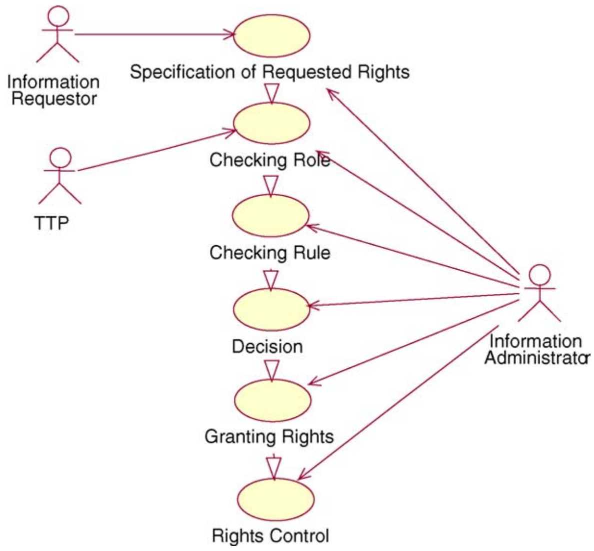


Fig. 17 Access control basic use case.

- a set of basic element sets;
- a set of RBAC relations involving those element sets (containing subsets of Cartesian products denoting valid assignments);
- a set of mapping functions that yield instances of members from one element set for a given instance from another element set.

Putting the aforementioned policies in the foreground, another class diagram has been derived by the OMG HDTF in its RADS (see Fig. 20). This figure is another view on the same issue, excellently demonstrating the openness and flexibility of the RM-ODP methodology. However, the model presents some of the constraint rules and attributes hidden in the coarse view of Fig. 19, settled in the refined NIST RBAC definitions.

Access control involves comparing specific information (privileges, roles, location, etc.) with the access rules associated with the requested data. Access is granted only if the validation information compares with, or dominates, system access rules. Access control is not distributed and each application manages the release of its own information.

6.8.1. Access control information

Access control information (ACI) comprises any information used for access control purposes, including contextual information (see [38]). Following, the types of ACI used to manage the PMI are described. ACI may be associated with the User, with a specific target (e.g. healthcare application), with a requested action, or with certain contextual information. For interoperability purposes, the format and syntax of the data supporting these ACI contexts may need further standardisation/specification. The following table (Table 10) presents more details about access control information.

6.8.2. Access control decision function

An access control decision function (ADF) is a specialised function that makes access control

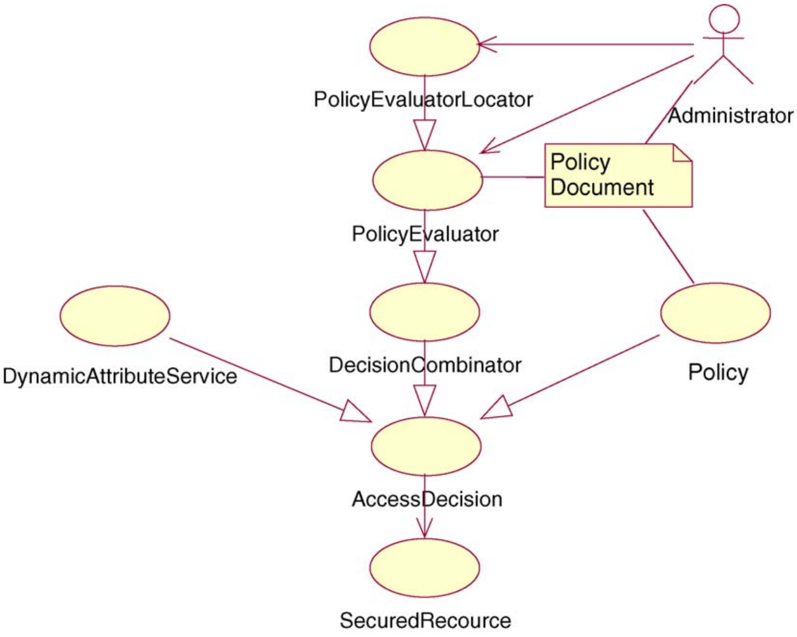


Fig. 18 Resource access decision use case.

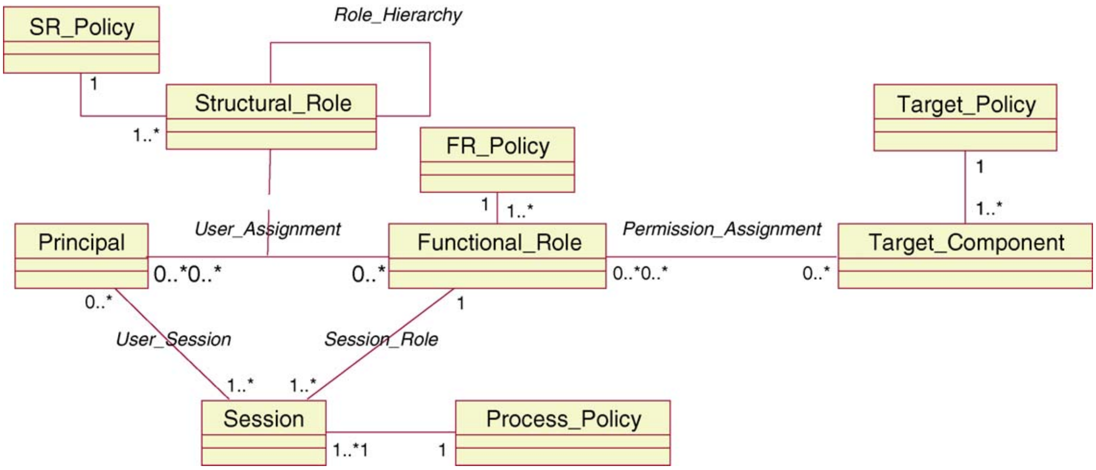


Fig. 19 Role-based access control schema.

decisions by applying access control policy rules to a requested action, ACI (of initiators, targets, actions, or that retained from prior actions), and the context in which the request is made (see [38]).

6.8.3. Access control decision information

Access control decision information (ADI) is the portion (possibly all) of the ACI made available to the ADF in making a particular access control decision (see [38]).

6.8.4. Access control enforcement function

An access control enforcement function (AEF) is a specialised function that is part of the access path between an initiator and a target on each access that enforces the decisions made by the ADF (see [38]).

6.9. Information Distance Model

Regarding the distance of persons to personal information, three person roles with growing distance to

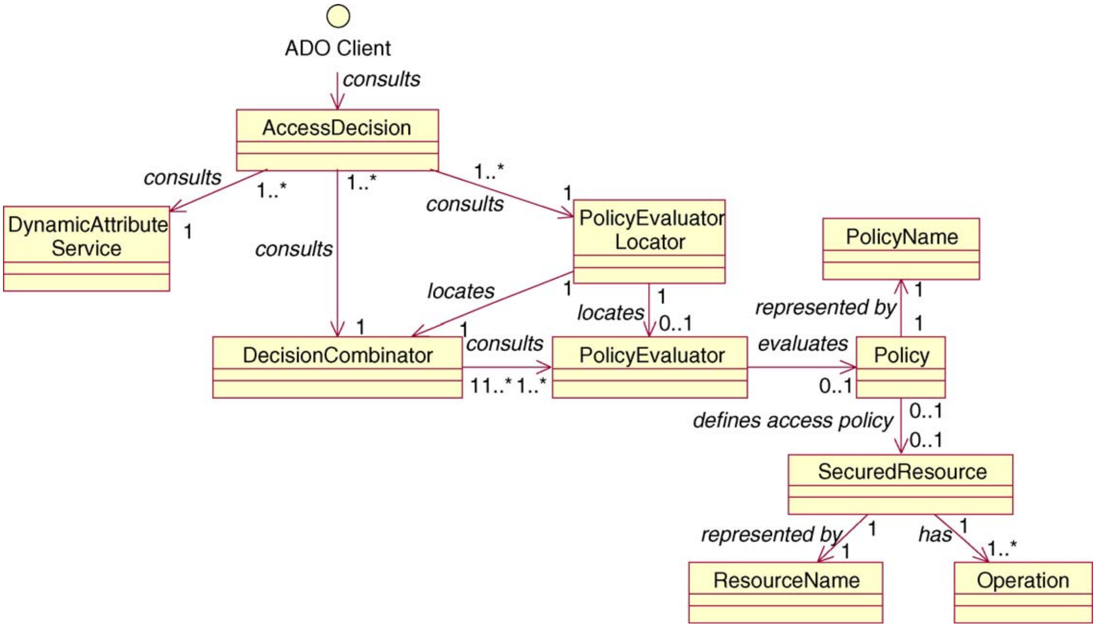


Fig. 20 Resource Access Decision Model [37].

Table 10 Access control information

User ACI	Target ACI	Action ACI	Contextual ACI
Individual access control identities	Target access control identities	ACI associated with operating zoning action (data ACI), for example	Time periods
Identifier of hierarchical group in which membership is asserted, for example, organisational position	Individual initiator access control identities and the actions on the target allowed or denied them	- Sensitivity markings	Route (an access may be granted only if the route being used to specific characteristics)
Identifier of functional group in which membership is asserted, for example, membership of a project or task group	Hierarchical group membership access control identities and the actions on the target allowed or denied them	- Integrity markings	Location (and access may be granted only two initiators as specific in-systems, workstations are terminals, or only two initiators any specific physical location)
Role that may be taken	Functional group membership access control identities and the actions on the target allowed or denied them	- Originator identity	System status (and access may be granted only for a particular ACI when the system has a particular status, for example during a disaster recovery)
Sensitivity markings to which access is allowed	Role access control identities and the actions on the target allowed or denied them	- Owner identity	Strength of authentication (an access may only be granted when authentication mechanisms of at least a given strength are use)
Integrity markings to which access is allowed	Authorities and the actions authorised for them	ACI associated with the action as a whole, for example	Other access currently active for this or other initiators
A target access control identity and the actions allowed on the target-that is a capability	Sensitivity markings	- Initiator ACI	
Security attributes of delegates	Integrity markings	- Permitted initiator and target pairs	
Location, for example, sign-on workstation		- Permitted targets	
		- Permitted initiators (users)	
		- Allowed class of operations (for example, read, write)	
		- Required integrity level	

the information can be specified:

- Originator of information (holder of data),
- Producer of information (interpreter of data),
- Administrator of information (user of information).

In a healthcare environment, the originator of information is normally the patient, and the producer of information is, for example, the doctor. An example of an information user is a pharmacist. Of cause, the Information Distance Model is also applicable to other principle instances aforementioned. Because of its coarse granularity structure, the model will normally be replaced or at least formally interpreted by other models discussed above.

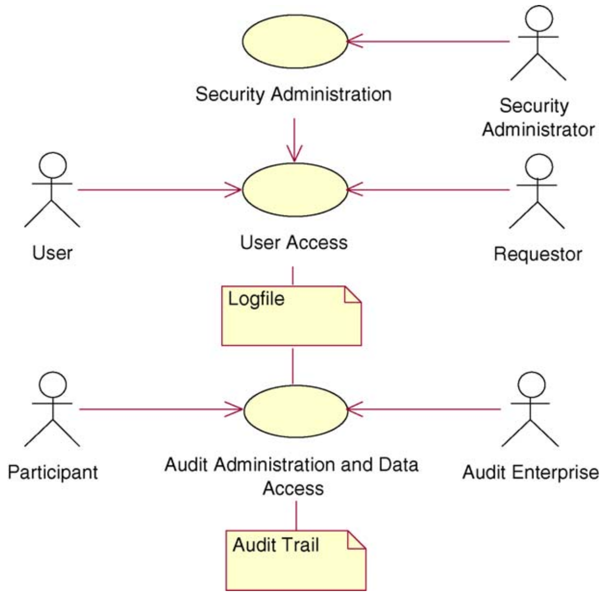


Fig. 21 Audit use case diagram.

Following the ''need to know'' principle, an increasing distance to information causes greater restrictions regarding privileges granted.

6.10. Audit Model

Extracts of personal medical records must only be accessed by professionals contributing to the patient's care related to the information they are interested in. Only professionals contributing to the patient's care in relation to the information they are interested in shall be allowed to access extracts of personal medical records, as stated already in Section 1. Therefore, personal permission lists should rule the use of personal health information. In cases of group-related access rights defined in the RBAC context, the authenticated professional's action undertaken in the record has to be logged [39]. Audit trail and audit procedure must be managed and communicated properly. Glen Marshall has developed a ''Security Audit and Access Accountability Message—XML Data Definitions for Healthcare Applications'' specification, which has been standardised at both HL7 and IETF level [40].

Fig. 21 demonstrates the rough use case showing the main actors in the audit process to derive the audit trail as the legally or at least regulatory defined outcome of this scenario. Because audit is a very complex approach going far beyond the scope of this paper, literature should be consulted for details.

7. Discussions

Health information systems (HIS) and especially the EHR as a core HIS application have to be built in a safe, secure, and reliable way, nevertheless following a future-proof (i.e. an open, scaleable, flexible, and portable) approach. For that reason, they have to follow advanced architectural paradigms. Security and privacy services that are applied must be an integral part of that architecture. Following the Generic Component Model with its abstraction paths according to views defined in the ISO Reference Model—Open Distributed Processing [9] and the components' composition/decomposition, security and privacy services have to be formally modelled at different levels of granularity using meta-languages such as UML and XML. Basic challenges to be met concern both policy modelling and role modelling. The different ways developed for dealing with definition and presentation of models needed have to be harmonised. The paper's outcome can be directly introduced in architectural models for, for example, EHR systems to analyse, design, implement, and maintain them towards future-proof health information systems as presented in refs. [6,7]. Thereby, the platform-independent models must be automatically transformed into platform-specific ones, which have to be eventually instantiated. Such an instance may be expressed currently using other representation forms such as ASN.1, see e.g. [41].

Currently, security solutions are at best based on a thread analysis followed by a risk assessment procedure for deriving requirements and solutions to establish security and privacy according to the policies defined legally, organisationally, etc. All specifications made are provided in narratives, at best following a formal procedure as described, e.g. in controls of ISO 27799 [42] or the Common Criteria [43] describing security objectives, requirements, functional and assuring components defining the protection profile, and finally the security targets.

For keeping security administration manageable, the individually assigned authorisation to accessing data and using functions has been turned from mandatory and hierarchical military-based or the discretional ownership-based access control concept towards the role-based access control management approach in the mid-nineties. As this approach has been improved in the late nineties in the security specialists' domain, security and privacy solutions according to the special requirements in the healthcare domain have been evolved separately as demonstrated in this paper.

Despite of the development that has taken place in Europe as well as in the international standardisation scene, the current approach of the VHA refers especially to a paper of Neumann and Strembeck from 2002 named ''A Scenario-driven Role Engineering Process for Functional RBAC Roles'' introducing functional and structural roles and a workflow-based role engineering process [4]. The Neumann and Strembeck article clearly confirms the long-term achievements resulting in our presented component-oriented, model-driven architectural approach for security services established and meanwhile even standardised at ISO under the responsibility of the first author. It does not add new aspects to our work but provides a clipping of some aspects.

The policy-based approach for dealing with security requirements in the healthcare domain seems to be the most promising way of dealing with the challenges mentioned. Additionally, it provides a framework for mapping the approaches published by different international and national teams.

8. Conclusions

Most of the available solutions for defining and enforcing security and privacy solutions suffer from the weakness of the separation of definition and enforcement regarding underlying paradigm and the process, i.e. structure, functionality, methodology, accountability, specification, and processing languages, etc.

The proposed paradigm has been developed over 10 years and demonstrated and evaluated in international projects. For the first time, security services have been directly embedded into the architectural components of health information systems using the same principles and the same process as for components' structure and functionality concerning any services. Already, since the beginning of the nineties, functional and structural roles have been defined and instantiated by the authors within several projects and initiatives. This status has been developed from the RBAC concept related to transactional steps of a workflow towards an architectural approach of security. For that purpose, a generic security model as well as a Generic Component Model for health information systems has been introduced in the mid-nineties, which has subsequently been embedded into a completely model-driven approach for analysing, specifying, implementing and maintaining health information systems. The proposed approach has been first implemented and evaluated within the European HARP project [44], and is now providing the basis for many ISO and CEN standards as well as for the architectural approach to the electronic health record (EHR) as the core application in healthcare.

The approach presented in this paper allows for the central management of users, privileges, rules, policies, separation of security management and secure application functions. It embeds security (policies as structural and functional constraints) into applications. Furthermore, it enables scalability of both security services and mechanisms on one hand and applications on the other. The approach separates AEF, ADI, ADC, etc., as defined in ref. [12]. By that way, it supports security-unaware environments.

What is already known on the topic

There are many solutions available for access control, and some work has been provided on privilege management. Some of the work has been dedicated to the healthcare domain.

Currently, requirements and solutions for security services and especially application security services have been specified using narrative text, platform-specific or domain-specific expression means, however.

What has been added by the paper

The paper integrates security services into a very advanced architectural framework using paradigms and methodologies in a model-driven architecture environment.

Being involved in research and development including standardisation for many years, the authors first introduced formal models to describe security requirements and solutions for advanced health information systems and especially the health systems' core application Electronic Health Record.

The presented results provide a framework for mapping the approaches published by different international and national teams as well as an approved basis for future-proof EHR systems and related applications.

Acknowledgement

The authors are especially in debt to Dr. Dipak Kalra and David Lloyd from CHIME, University College London (UK), Dr. Luuc Posthumus from AMC Amsterdam (The Netherlands), Bryan Manning from Goring-by-Sea (UK), Thomas Beale from Ocean Informatics Pty, Ltd. (Australia), as well as to the colleagues of ISO TC 215 and CEN TC 251 for their kind support.

References

[1] B. Blobel, F. Roger-France, A Systematic Approach for Analysis and Design of Secure Health Information Systems, Int. J. Med. Inform. 62 (3) (2001) 51—78, http://dx.doi.org/10.1016/S1386-5056(01)00147-2.

[2] S. Castano, M. Fugini, G. Martella, P. Samarati, Database Security, Addison—Wesley Publishing Company, Wokingham, 1995.

[3] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, C.E. Youman, Role-based access control models, IEEE Comput. 29 (2) (2001) 38—47 (February 1996) http://doi.ieeecomputersociety.org/10.1109/2.485845.

[4] G. Neumann, M. Strembeck, A Scenario driven role engineering process for functional RBAC roles, in: Proceedings of SACMAT'02, June 34, Monterey, CA, USA, 2002 (ACM 1581134967/02/0006) http://doi.acm.org/10.1145/507711.507717.

[5] A.A. El Kalam, R. El Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miège, C. Saurel, G. Trouessin, Organisation based access control, in: Policy '03, Proceedings of the FOURTHth IEEE International Workshop on Policies for Distributed Systems and Networks, 2003, pp. 120—134.

[6] B. Blobel, Analysis, Design and Implementation of Secure and Interoperable Distributed Health Information Systems, vol. 89, Series Studies in Health Technology and Informatics, IOS Press, Amsterdam, 2002.

[7] B. Blobel, P. Pharow, Implementing MDA-based distributed interoperable, flexible, scalable, portable and secure EHR systems, in: L. Bos, S. Laxminarayan, A. Marsh (Eds.), Medical and Care Compunetics 1, Series Studies in Health Technology and Informatics, vol. 103, IOS Press, Amsterdam, 2004, pp. 394—399.

[8] B. Blobel, Assessment of Middleware Concepts Using a Generic Component Model, in: Proceedings of the Conference Toward An Electronic Health Record Europe '97, London, 1997, pp. 221—228.

[9] Health Level Seven Inc., HL7 RIM, http://www.hl7.org.

[10] College of American Pathologists, SNOMED CT^{TM}, http://www.cap.org.

[11] ISO/IEC 10746 Information technology, Open Distributed Processing, Reference Model.

[12] B. Blobel, R. Nordberg, Privilege Management and Access Control in Shared Care IS and EHR, in: R. Baud, M. Fieschi, P. LeBeux, P. Ruch (Eds.), The New Navigators: From Professionals to Patients, Series Studies in Health Technology and Informatics, vol.95, IOS Press, Amsterdam, 2003, pp. 251—256.

[13] M. Jeckle, Entwurf von XML Sprachen, Java Spectrum 6 (2000) 56—60.

[14] SAIC: Role-Based Access Control (RBAC) Role Engineering Process, Version 3.0, May 2004.

[15] J. Siegel, Quick CORBA® 3, John Wiley & Sons, New York, Chichester, Weinheim, Brisbane, Singapore, Toronto, 2001.

[16] T. Beale, An interoperable knowledge methodology for future-proof information systems, Ocean Informatics Pty, Ltd. 2001, http://www.deepthought.com.au/it/archetypes/Output/front.html.

[17] ISO/IEC 8824 Information Technology—Open Systems Interconnection, Specification of Abstract Syntax Notation One (ASN.1).

[18] Object Management Group Inc., CORBA Person Identification Service Specification, Franingham 2001. http://www.omg.org.

[19] B. Blobel, Open information systems and data security in medicine, in: B. Barber, A. Treacher, K. Louwerse (Eds.), Towards Security in Medical Telematics, Series Studies in Health Technology and Informatics, vol. 27, IOS Press, Amsterdam, 1996, pp. 168—182.

[20] W. Stallings, Network and Internet Security. Principles and Practice, Prentice Hall, Hemel, Hempstead, 1995.

[21] CEN ENV 13608 Health informatics, Security for healthcare communication.

[22] VHA Security Architecture Framework, Enterprise Architecture 2001 Version, http://www.va.gov/OIT/EAM/EAservice.

[23] Object Management Group Inc., CORBA Security Services Specification, Franingham 2001, http://www.omg.org.

[24] Health Level Seven Inc., Data Types, http://www.hl7.org.

[25] Object Management Group Inc., Object Constraint Language, Franingham 2001, http://www.omg.org.

[26] N. Damianou, N. Dulay, E. Lupu, M. Sloman, Ponder—-A Language for Specifying Security and Management Policies for Distributed Systems. The Language Specification, Version 2.3, Imperial College Research Report DoC 2000/1, 20 October 2000, http://www.doc.ic.ac.uk/research/technicalreports/2000/DTR00-1.pdf.

[27] OASIS, Extensible Access Control Markup Language, http://www.oasis-open.org.

[28] B. Cohen, A formal model of healthcare security policy, London City University, 1996.

[29] Australian Government, Department of Health and Aging, The Australian HealthConnect Project, http://www.health.gov.au.

[30] Council of Europe, Directive 95/46/EC, On the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of such Data (OJ L281/31-50, 24 October 1995). Strasbourg1995, http://www.cdt.org/privacy/eudirective/EU_Directive_.html.

[31] ISO CD 22857 Health informatics, Guidelines on Data Protection to Facilitate trans-Border Flows of Personal Health Information.

[32] K. Yamamoto, K. Ishikawa, M. Miyaji, Y. Nakamura, S. Nishi, T. Sasaki, K. Tsuji, R. Watanabe, The Awareness of Security Issues among Hospitals in Japan, in: IMIA Conference: Caring for Health Information Safety, Security and Secrecy, Heemskerk, The Netherlands, November 13—16, 1993.

[33] ASTM E1986-98, Standard Guide for Information Access Privileges to Health Information.

[34] ASTM E2212-02, Standard Practice Healthcare Certificate Policy.

[35] D.F. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, R. Chandramouli, Proposed NIST Standard for Role-Based Access Control, ACM Trans. Inform. Syst. Security 4 (August (3)) (2001) 224—274.

[36] ISO TS 21298 Health informatics, Structural and functional roles.

[37] Object Management Group Inc., Object Constraint Language, Resource Access Decision Facility Specification, Version 1.0, Franingham, 2001, http://www.omg.org.

[38] ISO/IEC 10181-3 Information Technology, Open Systems Interconnection, Security Frameworks in Open Systems, Access Control.

[39] R.J. Anderson, Security in clinical information systems, Version 1.1. Computer Laboratory, University of Cambridge, 4 January 1996.

[40] Marshall G., Security Audit and Access Accountability Message—XML Data Definitions for Healthcare Applications. RFC 3881, http://www.ietf.org.

[41] ASTM E31.20 Privilege Management Infrastructure, Working Draft 0.6, November 2, 2001.

[42] ISO 27799 Health informatics Security management in health using ISO/IEC 17799.

[43] ISO/IEC 15408,1999 Common Criteria for Information Technology, Security Evaluation, Part 2: Security Functional Requirements, ISO, August 1999.

[44] B. Blobel, G. Stassinopoulos, P. Pharow, Model-based design and implementation of secure, interoperable EHR systems, in: M.A. Musen, C.P. Friedman, J.M. Teich (Eds.), AMIA 2003 Symposium Biomedical and Health Informatics: From Foundations to Applications, vol. 96—100, American Medical Informatics Association 2003 Proceedings, Bethesda, 2003.

Available online at www.sciencedirect.com

SCIENCE DIRECT®