BUSINESS-DRIVEN ENTERPRISE AUTHORIZATION

Moving towards a unified authorization architecture

Tom Beiler
Strat Hollis Business Systems Trust & Security, Zürich, Switzerland

Keywords: Authorization, Enterprise Security, System Specification, Modelling Concepts

Abstract: Information systems of large enterprises experience a shift from an application-centric architecture towards a focus on process orientation and web services. The information system is opened to business partners to allow for self-management and seamless cross-enterprise process integration. Aiming at higher flexibility and lower costs, this strategy also produces great new challenges the security and administrative support systems have to cope with. The security of the enterprise system has to keep up and scale with the new qualitive level of the overall system. In this context we propose an enterprise authorization system model which allows a unified treatment of the enterprise's authorization issues, and permits the native integration of authorization processes into the business system for greater synergy. The proposed model supports information system architects to avoid that authorization becomes a major obstacle for the new architecture strategy.

1 INTRODUCTION

As of today, large enterprises face a challenging paradigm shift within their enterprise information system architecture. No longer the enterprise system consists of a number of monolithic applications, individually interconnected and sharing only the database level. Instead, the system's business functionalities are generalized and parametrized to achieve a better synergy by shifting the component reuse one level up within the system architecture. In addition the enterprise information system is increasingly opened for business partners and customers, in order to integrate seamlessly their business processes, and to allow for inline self-management of the enterprise business partners. This strategic direction aims at a reduction in effort and costs of the business administration for both the enterprise and its business partners, while it also leads to a faster adoption of changed business objectives. In contrast to the situation today, applications will be light-weight, and concentrate on presentation and the assembly of existing web services providing the business functionalities. Applications are tailored easily to the particular needs of a multitude of user groups, and the development cycles are fast and efficient.

The downside is that the gain in flexibility also leads to a qualitative increase of complexity and dynamics of the security system. For the authorization system in particular this addresses the mapping of users to services and applications, i.e, to permissions within the enterprise information system. The number of combinations significantly increases, as well as the change frequency. For the authorization in the information system this fact imposes a heavy burden. More specifically, the authorization system is required

— to cope with the massively increased frequency of authorization change requests,

— to allow for self-management of users to streamline the administration processes,

— to seamlessly integrate with the business processes in order to minimize administrational efforts,

— to treat all system users with a unified approach unless business or security reasons dictate otherwise.

The typical situation of authorization in an enterprise today exhibits a mission-critical gap to these requirements inherited with the new paradigm. First, internal and external users are most often treated completely different throughout the whole information system. For each of both user groups, a separate, specific authorization system is employed. While for internal actors role-based access control has been subject to research and development for the

Beiler T. (2004).
BUSINESS-DRIVEN ENTERPRISE AUTHORIZATION - Moving towards a unified authorization architecture.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 67-72
DOI: 10.5220/0002643100670072

past decade, no similar, unifying effort has been undertaken for system-external actors.

Secondly, authorization is disconnected from the normal course of business processing. Authorization logic is, with its larger and more complex aspects, hard-wired within the application logic. Correct implementation of the authorization logic is left to the application developer. The authorization processes are often semi-manual causing a large administrative overhead. This inherently leads to difficulties with the much higher change frequencies and the expected self-management of users due to the lack of proper automation and integration into the business processes. This situation undermines any effort of an effective treatment of authorization, and may turn into a major obstacle for the web-services strategy.

In this article we propose an integrative model which enables the enterprise authorization system to be viewed as *one* system, with the add-on that it combines cleanly and reasonable with the business production of the enterprise information system. In the situation today enterprise authorization trivially consists of the sum of isolated system fragments covering particular aspects. Therefore we advocate a conceptual model which covers the future directions of authorization, as well as allowing a feasible migration from the current landscape towards a unified, centralized authorization system which is appropriate for a web-services architecture. Technically, the model is composed from three layers, including authorization policies, an authorization language which allows the formulation of the authorization logic and policies, and a virtual access matrix which encapsulated the different, system-specific access control systems within the enterprise information system's components.

The authorization system is connected to the business system by means of an event-based mechanism, similar to a model-view-controller mechanism where the business system is the model while the authorization system constitutes the view. This choice permits a clean separation from the business system in the specification of authorization policies.

We compare our approach with the Single Sign-On initiative which has established in the practice of enterprise information systems. Before Single Sign-On, authentication has been solved isolated for each application. As the application landscape grew complex, the management of the variety of authentication methods and means has no longer been feasible. Furthermore, with opening the enterprise for the web, the security requirements grew, and the need for stronger authentication means became urgent. With Single Sign-On, authentication is a centralized service of the information system's security infrastructure. We postulate that authorization experience a similar development from isolated, application specific solutions towards a centralized, unified, and mature authorization system.

The article is organized as follows. Section 2 provides a description of the proposed authorization system model in more detail with a black-box/white-box approach and the three system layers. With section 3, a conclusive discussion is given, while section 4 relates our approach to current research. This article is based on the authorization system reference model which serves as a conceptual foundation for related consulting cases at Strat Hollis.

2 SYSTEM MODEL

In our model, we define an enterprise authorization system as that particular subsystem of the enterprise business production system which decides, for all actors within the business system, on the actors' permissions within that system. The term 'permission' refers to any kind of system resource which is to be protected with acess control, being combined with the particular actions using or manipulating the referenced object. We explicitly state that the system also includes those parts of the authorization processes which are not implemented in terms of IT, like semi-manual or paper-based authorization processes. In contrast to the authorization system, we identify the access control system as that particular subsystem of the information system which enforces the authorizations produced by the authorization system during information system operation. We continue the system description with a blackbox approach characterizing the connected components in terms of consumers and producers, followed by a whitebox description of the system components. Figure 1 illustrates the model description.

Two types of producers, distinguished by their application times, contribute input to the authorization system. At development time, policy providers provide the system with the authorization policies to determine the decision behaviour of the authorization system according to their respective, domain-specific requirements. Typical authorization policy domains are the enterprise organisation and role model, enterprise security policies, and legal obligations. The union of these domain-specific authorization policies constitutes the authorization specification of the enterprise authorization system. Authorization policies are subject to change cycles and continous development.

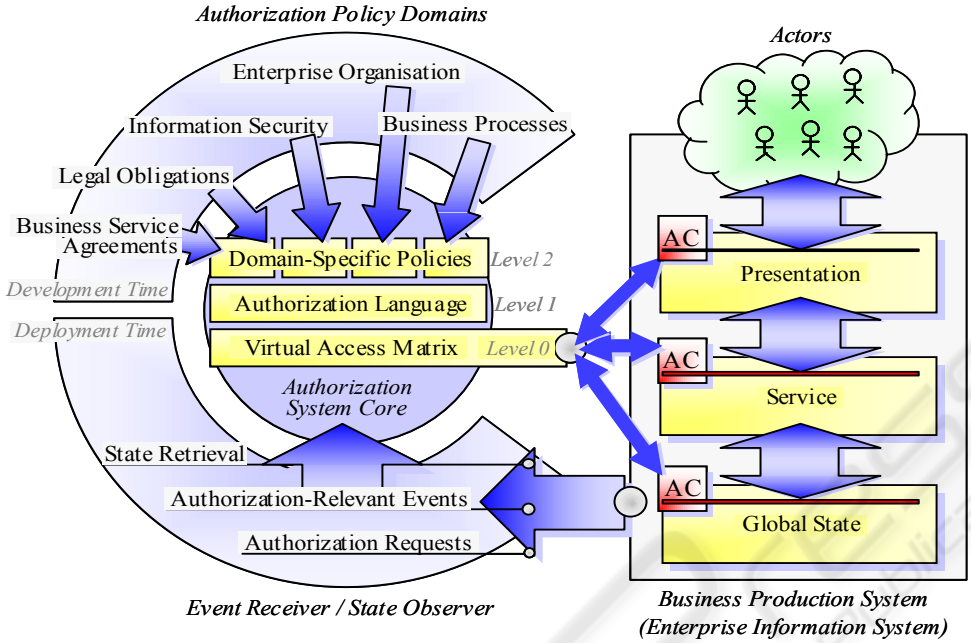

Figure 1: Overview of the Authorization System Model.

The second kind of input refers to the deployment time, i.e., the runtime of the productive authorization system. During the course of operation of the business production system the authorization system receives authorization requests, and subsequently reconfigures the authorization state accordingly. The authorization system is connected to the business production system by an event observer which allows to observe business processes as they are relevant for authorization. The mechanism allows to declare the events of the business system which are relevant for authorization, and follows the event/condition/action paradigm of statecharts: on occurrence of an authorization-relevant business event (the event), the configuration of the authorization state is recomputed (the action) in compliance with the authorization policies (the conditions). The business system ideally never stops its service, so the authorization system also never stops, and parallelizes both development and deployment time.

The most prominent consumer of the authorization state is the access control subsystem of the productive business system. The access control system retrieves the authorization state from the authorization system for a given access request, and additionally ensures that access control can not be circumvented. Also other subsystems may retrieve the authorization state, e.g., reporting, auditing and logging, user navigation for better convenience.

The construction of the authorization core system is layered into three levels following the abstraction sequence of common computer programs with runtime system and environment, programming language, and programs on top to solve particular problems respectively manipulate data. At level 0, the lowest level, resides the virtual access matrix. The access matrix provides consumers with the current authorization state, i.e., the current set of valid authorization decision results for any given access request. The virtual access matrix encapsulates the local, existing access control systems of the business production and is responsible for the timely propagation of the authorization state to the consuments.

Level 1 of the authorization core system provides a set of basic authorization policy primitives which are useful and common for the policy domains. In this sense, level 1 forms an all-purpose authorization language. The primitives of this language include grouping constructs, delegation, logical inclusion and exclusion (for example for separation of duties), conflict resolution strategies.

Level 2, the top layer, hosts the authorization policies from the authorization policy domains. The policies of respective domain form a compartment by their domain, though they are not strictly separated and may again build interaction and abstraction structures. This, however, depends on the specific circumstances of an enterprise.

2.1 Virtual Access Matrix

The virtual access matrix provides consumers with the authorization state, i.e., the permissions, for retrieval. Conceptually, the access matrix implements simply a look-up table for all system permissions. With an ideal access matrix, the current, correct permissions can be consumed at any place in the production system, and the retrieval is very fast. In reality a multitude of access control systems which are specifically bound to the information system components are found. These systems have to be consistently coordinated to allow an approximation of this idealized access matrix. To achieve this, the virtual access matrix utilizes a lazy evaluation mechanism with intelligent caching which transports the authorization state into the concerned access control system. In one variant this mechanism loops into the access requests and triggers a reconfiguration if necessary. The results are cached for reuse, taking into account the structure of the local access control storages. The other variant pushes a state change actively into the local stores. A heuristics of the change/request ratio is used for efficient determination of the reconfiguration time.

The maximal materialization time, i.e., the latency from an authorization-relevant until it is effective in the access control, is specified together with the event itself since the tolerated delay naturally depends on the event type.

The virtual access matrix allows not only retrieval of singular access requests but also provides query support for access sets. This feature is useful for reporting and audit, and for user navigation in congruence with the user's actual permissions.

2.2 Authorization Language

The authorization language constitutes the programming formalism with which the authorization domains express their authorization requirements and constraints (authorization policies). The language contains grouping constructs and operations as the basic collection structure, letting a group name representing its members. Grouping is the base for higher structures like hierarchical inclusion or other, more complex structures. Logical inclusion respectively implication, and exclusion allows the relating of groups. Groups can be dynamically specified by a predicate which is evaluated on specified events to determine group membership. For better convenience, also a single item can be treated as a group one-member, (singleton). The grouping mechanism allows the expression of role models, and of organisational units. More complex structures can also be built on grouping.

The authorization language supports delegation (grant and revoke). Delegation is the base mechanism for decentralizing the administration of permissions. Moreover, delegation is not an administrational convenience, but is a natural authorization construct. The trivial illustration is object ownership where the object owner determines the access rights for other actors, which in turn may delegate the right. Delegation can become complex whenever rights are passed on from actor to actor. In case of cascaded revocationn, the whole path has to be traced to render the authorization invalid. Delegation is a prerequisite for the automation of privileges management.

Basic conflict resolution strategies are required to handle conflicting authorizations as they arise from non-monotonous authorization policies. The nodes of the policy decision graph may also be, beside conflict resolution strategies, associated with a substitution or priority strategies. If as resolution is not possible or undesired, an exception is passed back.

Priorities may be set for overriding authorizations, negative or positive, of lower priority. A priority may be induced by an authoritative actor expressing an authorization exception: for example, a client advisor may exceptionally assign a certain service to a premium customer in an informal manner. Also policies are associated with priorities. For example, the security policy overrules positive authorizations from a role model.

The authorization language is equipped with process flow constructs to implement authorization processes, and which can also be interleaved with non-authorization workflows, e.g., for call back to the authorization requester. The process component allows a loop-in of authorization dialogues into the non-authorization processes, for example a disclaimer interaction. Workflow is a precondition for a smooth integration of authorization into non-authorization administration processes up to the level of sharing the user interface forms. The process aspects of authorization are not different from other processes besides that they concerned with authorization.

The authorization language also provides atomicity constructs to declare a set of primitives as one coherent unit with transactional semantics. The system must handle concurrent authorization requests and still has to preserve consistency.

In its essence, the authorization language is a logic programming system, such as Prolog is the classic representative, or such as the upcoming class

of business rule systems may implement. Logical programming languages are declarative, meaning that a request is formulated to the system without being concerned with the implementation of how the result is computed. To compute the result, the logic system evaluates a set of rules where the system selects and iterates the rules automatically based on a thorough mathematical-logical calculus. However, though being declarative at the frontend, the system still has to be fed with logical rules. This is the task of the authorization policy providers and domains.

2.3 Authorization Policy Domains

At Level 2, the enterprise authorization requirements of the business units are reflected in terms of authorization policies. In the following we give a heuristic base canon of policy domains. Refinement and extension depend on the specific circumstances and needs of the enterprise. We identify the following authorization logic domains:

— Enterprise organization structure. This domain includes the enterprise role model as the most prominent structure, and the decomposition of an enterprise into organizational units. Further structures may refer to projects, and to customer groups.

— Business processes and their definitions. From the definitions of business processes and workflows immediately result authorization specifications. Besides authorizing actors for the respective steps in the process sequence, this also covers more complex specifications as the four-eyes-principle and separation of duties.

— Legal obligations. This domain represents the enterprise guidelines on service usage and system access. For example, customer access to the information system via internet is not allowed from any arbitrary countries. Also upcoming privacy-related obligations reside in this domain.

— Business service agreements. This domain represents the obligations side of authorizations: If the enterprise promises its business partners the availability of certain services, possibly bound to a contract, the authorization must ensure the corresponding authorizations. The logic of this domain refers to the services and the product structure of the enterprise.

— Information security. This domain provides the authorization system with the enterprise security policies. These policies constrain the logic in a mandatory manner.

Note that the authorization logic domains are not separated strictly. Moreover there may again be a structure relating the policies, possibly building further layers.

2.4 An Illustrating Example

We illustrate the model with the following example. Assume that, in a bank, a customer may access a brokerage service. Usage of this service is, due to legal restrictions, allowed only for customers whose domicile is in a country included within a particular country list of the legal department. To determine a customer's permission for this service, the customer's domicile is required within a certain predicate of one of the legal policies. The predicate compares the individual domicile's country with the country list. To do this, the predicate retrieves this information from the CRM-system.

Whenever either a customer's domicile changes within the CRM-system or the country list changes, a recomputation of the permission value is required before the next access request. The permission may either not change, may change from positive to negative, or vice versa. The virtual access matrix determines the evaluation strategy for the change. Since in the case the change/access ratio is low, the reconfiguration is computed immediately and pushed into the access control systems of the web and application servers. In the second case where the country list changes it is of advantage to schedule the recomputation at a time of low load in the night. This is allowed because the maximal materialization delay is specified as 'within 24 hours'.

Before the first use of the brokerage service the customer is required to accept a disclaimer as an enabling precondition. This again is implemented by means of an active predicate which this time loops into the brokerage service call. An acceptance is stored within the authorization system, and the disclaimer process is not looped-in anymore. But nevertheless, if the disclaimer text itself changes, the process automatically starts again with the new disclaimer.

3 CONCLUSION

In this article we have presented an authorization system model which enables enterprise system architects to face the web services challenge from the authorization perspective. The model provides a conceptual reference for planning the future authorization system in order to shift this security system up to the level the web services paradigm demands. Though the proposed system yet is conceptual model, we believe that future products will be close to this model. To ensure this, the model combines recent research streams in the field of authorization, such as policy frameworks and the utilization of logic languages, with modern system

design and architecture. As of today, the system model can be implemented to certain degree with existing products such as IBM Tivoli Identity Manager, Access Manager, and Privacy Manager are suitable candidates. In that sense the proposed model provides guidance for a meaningful application of these products.

The given distinction of authorization and access control follows the typical responsibilities within an enterprise organization. We believe that the naive distinction in which any process at access time is the access control, and any other process changing permissions is authorization leads to confusion since access control products, e.g., Tivoli Access Manager, feature local portions authorization logic which clearly must be considered in the specification of the authorization policies, and should not be left to the pure access control domain.

A point of discussion is that logic programming is uncommon in enterprise development units. Nevertheless we believe that the new class of business-rule systems may break ground. The criticism that logic programs increase the system complexity is invalid since this complexity is only the reflection of the overall complexity induced by the web services paradigm.

4 RELATED WORK

Woo and Lam demand in 1993 that authorization be a system and language in its own respect, which is distributed, can handle conflicts, and is logic-oriented (Woo, Lam, 1993). They argue that authorization is an independent semantic concept, and criticise that authorization is mainly low-level and system-specifically addressed. Up to the authors knowledge, this issue has not been picked up in terms of a unified system model since. However, Varadharajan, Crall, and Pato (1998) give a practive-oriented approach for authorization in a full enterprise context.

Jajodia Samarati, Sapino, and Subrahmanian have proposed a framework in which different access control policies coexist (2001). This is a necessary prerequisite for complex authorization systems. The framework is specified in terms of logic programs. Bertino et al. (2001) also investigate authorization logics, based on the logic language Datalog from deductive databases. Ribeiro and Guedes (1999) provide an authorization language using the policy approach.

Karjoth and Schunter (2002) describe a privacy model in the enterprise context. This model integrates with our authorization model as a policy provider. Karjoth (2001) also gives an account on the Tivoli Access Manager which in our model would form part of the virtual access matrix, but which also contains authorization logic. (Sandhu et al., 1996) and (Sandhu, Ferraiolo, Kuhn, 2001) provide the original treatise on role-based access control models. Zhang, Ahn, and Chu extend the role-based model with delegation (2001).

REFERENCES

Bertino, E., Catania, B., Ferrari, E., Perlasca, P., 2001. A logical framework for reasoning about access control models, *Symposium on Access Control Methodologies and Techniques* (*SACMAT*).

Dai, J., Alves-Foss, J., 2000. Logic based authorization policy engineering, *available at: citeseer.nj.nec.com/596575.html*.

Jajodia, S., Samarati, P., Sapino, M.L., Subrahmanian, V.S., 2001. Flexible support for multiple access control policies, *ACM Transactions on Database Systems*, vol. 26, no 2.

Karjoth, G., 2001. The authorization model of Tivoli Policy Director, *Proceedings of the* 17th *Annual Computer Security Applications Conference* (*ACSAC*), New Orleans, Louisiana, pages 319–328.

Karjoth, G., Schunter, M., 2002. A privacy model for enterprises, *Proceedings of the 15th IEEE Computer Security Foundations Workshop*.

Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E., 1996. Role-based access control models, In *IEEE Computer,* vol. 29, no 2.

Sandhu, R., Ferraiolo, D., and Kuhn, R. 2001. The NIST-model for role-based access control: Towards a unified standard, *Proceedings of 5th ACM Workshop on Role-Based Access Control*, Berlin, Germany.

Varadharajan, V., Crall, C., Pato, J., 1998. Authorization in enterprise-wide distributed system: a practical approach, *14th Annual Computer Security Application Conference (ACSAC)*.

Woo, T.Y.C., Lam, S.S., 1993. Authorization in distributed systems: a new approach. *Journal of Computer Security*, vol. 2, no. 2–3, pages 107–136.

Zhang, L., Ahn, G.J., and Chu, B.T., 2001. A rule-based framework for role-based delegation, *6th ACM Symposium on Access Control Models and Technologies* (*SACMAT*), Chantilly, Virginia.

Ribeiro, C., Guedes, P. 1999. SPL: An access control language for security policies with complex constraints, Technical Report RT/0001/99, INESC.