Securing Collaborative Networks: Requirements of Supporting Secured Collaborative Processes

Paul de Vrieze and Lai Xu

Department of Computing and Informatics, Bournemouth University, Poole,
BH12 5BB Bournemouth, United Kingdom
{pdvrieze,lxu}@bournemouth.ac.uk

Abstract. The shift towards Industry 5.0 and Society 5.0 highlights the need for human-centric systems that balance automation with societal well-being. However, this must be achieved in a constantly evolving security landscape that emphasizes security by design. Furthermore, disruptive events are becoming more frequent, demanding systems with advanced transformative resilience and antifragility. Collaborative ecosystems that can improve and adapt throughout a sequence of attacks and disruptions are essential. Privacy regulations such as GDPR have also imposed strict requirements on the usage and security of personal data. To address these challenges, we investigate the security requirements for supporting collaborative processes across different systems and illustrate them using a specific collaborative process.

Keywords: Security: Collaborative Processes: Authorisation: Process Mediated Authorisation: Secured Collaborative Processes: GDPR.

1 Introduction

The emergence of Industry 5.0 and Society 5.0 has highlighted the importance of human-centric systems that balance automation with human and societal well-being. However, this must be achieved in a security landscape that is constantly evolving and increasingly requiring security by design. Additionally, disruptive events are becoming more frequent, requiring systems with advanced forms of transformative resilience and even antifragility. These challenges demand collaborative ecosystems that can improve and adapt throughout a sequence of attacks and disruptions.

Modern organizations operate in complex IT environments that include cloud systems, SaaS, and integration with suppliers and customers. Business process management (BPM) is often used to coordinate activities and automate repetitive tasks. With systems and organizations being increasingly interconnected, so has the threat landscape in terms of security breaches and data protection evolved. This means that ensuring the security of these systems has become paramount. A key element of the security is limiting access permissions to only what is necessary, when it is necessary.

In a collaborative context, managing authorisation of access to resources is important, but so is limiting that access to specific elements. For example, a subcontractor working on a part for a specific client may have access to that client's file, but not all client files. Similarly, privacy laws such as GDPR impose restrictions on

employee access to personal data unrelated to their tasks even within their own organization.

The General Data Protection Regulation (GDPR) [1] is a comprehensive EU regulation that governs the use of personal data within the EU or concerning EU citizens and has inspired similar laws in other jurisdictions. The GDPR's broad definition of personal data means that a vast amount of data is subject to its provisions. It requires businesses that process personal data to record the purposes of processing, the categories of data subjects and personal data, and the categories of recipients. Additionally, the GDPR mandates that controllers and processors ensure that any natural person with access to personal data only processes that data on the controller's instructions. These requirements underscore the importance of restricting data access to the specific process instance in which it is needed, which is provided by a BPM system in which the process definition also provides a process purpose and context for determining access needs and auditing information.

The traditional approach to authorising access to systems for use in activities involves providing indirect authorisation (see Figure 1). A user directory such as Microsoft Active Directory maintains a user database with users and roles, which are used in combination with further restrictions to restrict access to activities to only users with appropriate roles. However, the execution of the task may require access to further systems independent from the BPM system and not explicitly specified in the process model. Instead, the user is given permission to perform certain actions on certain systems through their roles, and only careful management ensures that the permissions provided are sufficient for the tasks. As the links between activities and permissions are mediated through the role concept, the permissions provided must be a superset of all permissions needed by any user in any of the activities with that role.

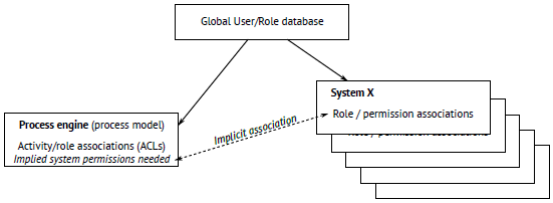

Figure 1 Traditional Authorisation of System Access

A significant amount of existing work in the authorisation area focuses on authorisation schemes such as Role Based Access Control (RBAC) [2] and more advanced schemes such as Attribute Based Access Control (ABAC) [3]. However, these works mainly concentrate on deciding and specifying the authorisation conditions without considering business process context and how to integrate these decisions into the authorised system's operation.

Previous works on task-based authorisation cover authorisation decisions that consider the process context [4–8], particularly to support concepts like separation and binding of duty. These works confirm that the need for authorised access to resources usually originates in a business process that initiates that access, even if it's sometimes a manual process that may or may not be described.

Existing works on authorisation schemes in a process-oriented context tend to solve the issue of dynamic permissions by adopting the approach of dynamic role assignment to the implementation. However, although dynamic role assignment reduces the temporal access to systems to only those times when access should be allowed, it limits access to resources in more specific ways. While the context enables access in this case, the access is not bound to the context in role-based systems.

In this paper we outline the requirements to allow for secure, privacy compliant collaborative processes that involve multiple systems and multiple parties. This is structured as follows: Section 2 provides a review of related research on the topic. Section 3 discusses federated authorisation schemes such as OAuth2, User Mediated Access (UMA), and Security Assertion Markup Language. In Section 4, we examine the application of these federated authorisation schemes in a process context. Section 5 outlines the requirements for process-mediated authorisation, and we use a real-world example to illustrate these requirements. Finally, in Section 6, we conclude this work.

2 Authorisation Models for Process Oriented Systems

There are several well-known models for organizing user authorisation to perform actions within a system, which can be applied in process-oriented contexts, though they are not limited to these contexts. In this discussion, we will explore several of these approaches, including both generic ones and specialized schemes for authorisation within process contexts. Leitner and Rinderle-Ma have provided a comprehensive review of authorisation and other security aspects in process-aware information systems [9]. In the following section, we will provide a brief overview of various systems, ranging from commonly used Role-Based Access Control (RBAC) to process-specific approaches for specifying access.

2.1 Role Based Access Control (RBAC)

Role-based access control [8, 10, 11] is a widely used and well-established approach to [12] authorisation. Its main goal is to simplify authorisation by assigning users to different roles and allowing authorisation to be based on roles rather than individual users. In this approach, each resource is linked to one or more role/permission combinations, which allows for different permissions to be granted to different roles. Because permissions are specified for each resource, it is easy to determine which roles have access to that resource.

In the context of authorisation for (distributed) processes, there are several drawbacks to the plain use of RBAC. Firstly, in practice, permissions are often designed at a very low resource granularity level, such as granting all customer service agents access to all customer data. This can lead to over-authorisation and increased security risks. Secondly, the evaluation of permissions is purely static, making it challenging to handle dynamic or temporary role assignments. Such assignments must be done outside the system, using administrative processes to update user roles or even dynamic roles and permissions set on resources. This approach can result in an unpredictable authorisation surface, complicating auditing of access and leading to potential security

vulnerabilities. Additionally, RBAC is designed for authorising people, not software that generally needs wide permissions as it acts upon instructions by uses or other software. Therefore, when traditional authorization is used to authorise software, this can result in blanket authorisations, violating the least privilege principle and presenting significant security and confidentiality risks in a collaborative context.

2.2 Attribute Based Access Control (ABAC)

To overcome the limitations of role-based access control, the Attribute Based Access Control (ABAC) [3, 13] approach offers a more flexible and comprehensive solution for specifying authorisation requirements. Unlike RBAC, which assigns access based on predefined roles, ABAC evaluates access requests based on a combination of attributes associated with the request. These attributes can be either static attributes associated with the user, including their role, or dynamic attributes that are context dependent.

ABAC is a well-established approach with both strengths and weaknesses [3, 14]. In the context of our specific problem, several properties are particularly relevant. First, ABAC enables temporal restrictions based on dynamic properties, without requiring role modifications. Second, attributes are used for specific resources, which can lead to high complexity and potential for inconsistency when specifying restrictions on a per-resource basis (without dynamic roles). Third, the approach is highly generic, but as a result, it lacks specific guidance and conceptual foundations. Finally, while ABAC provides more fine-grained conditions for specifying restrictions, it does not inherently separate the policy decision point from policy enforcement point or provide a clear separation between authorising and usage of authorisation, which are important aspects of policy management architecture.

2.3 Task based Access Control

In addition to generic access control approaches, there are also authorisation approaches specific to process contexts. Task-based access control (TBAC) [4–7] is one such approach that recognizes the importance of cross-process instance constraints, such as separation or binding of duty. TBAC is explicitly designed for multi-system environments and considers the usage of access control in a process context where permissible access is based on existing practices from fields such as accounting and auditing before the advent of computers.

Task-based authorisation recognizes the need for authorisations to be performed at the task level rather than just at the resource level. It also recognizes the concept of limited validity authorisations, as illustrated by paper-based signed documentation forms, as well as the possibility of delegations.

While TBAC provides a strong basis for providing cross-organizational authorization in processes, it is focused on deciding the authorizations needed and access to activities. First, the work by Thomas and Sandhu [4–7], among others, focuses on the specification and validity of processes using task-based authorisation constraints, rather than on the mechanism through which authorisation is effectuated. Second, it mainly focuses on access by human actors. Effectuating authorisation still relies on active management of permission states rather than a demand-based

evaluation of permissions. In general, the research has focused on other aspects of authorisation rather than the implementation.

Instead of replacing the work on TBAC, the proposed process-mediated authorisation approach focuses on embedding authorisation into modern standards-based information systems, rather than on the integrity of process models [5] or the specification of authorisation constraints [8, 15].

Lu et al.'s work on Task-Activity Based Access Control (TABAC) [16] presents an approach that is similar to process mediated authorisation. This approach divides tasks into separate activities that can be authorised individually and extends upon TBAC. The implementation is based on WS-BPEL [17] and uses a custom SOAP protocol requiring service support, which differs from our approach based on federated authorisation. The paper also explicitly addresses cross-organizational access control.

2.4 Authorisation in BPMN and eXtensible Access Control Markup Language

BPMN 2 [18] is the industry standard for graphical representation of process models. In this graphical notation, swim-lanes can be used to support the specification of roles. However, this approach is limited and does not provide resource access specifiers or constraints. The general assumption is that resources assigned to a task have appropriate permissions, but the details of assignment or what constitutes appropriate permissions are often beyond the scope of control flow specification.

The extensible access control markup language (XACML) [19] is a standard used to express access control policies in a centralized manner. While XACML is relevant in the specification of access control and authorization requirements it does not consider communication and enforcement, making it largely orthogonal to our work. One of its best practices is the separation of Policy Decision Point and Policy Enforcement Point, which is also applicable in process-mediated authorisation. In this context, the process engine, process model, and authorisation server provide access decision and resource services, which are used in enforcement. XACML's semantics for access decisions and its language to specify access control policies provide valuable background information for our work.

In short, authorisation models for process aware information systems focus heavily on access decisions, but often overlook access to additional systems beyond the primary system. These models excel at determining authorisation for activities and their assigned roles, but do not provide clear mechanisms for translating that authorisation into permissions for additional systems. To bridge this gap, we look at the requirements of process mediated authorisation in this work.

3 Federated Authorisation

In a collaborative network, which often operates in a distributed environment and involves multiple organizations, it is essential to utilize authentication approaches that enable each participating organization to authenticate its own employees with a single password. To achieve this, federated authentication and authorisation systems are recommended. The authorisations can be determined by the “home” authorization systems, based on a variety of factors, to be used by the relying systems. There are two

common protocols and approaches to federated authorization: OAuth 2 and SAML. We will discuss these below.

3.1 OAuth 2

OAuth 2 [12] is a popular authorisation approach widely used in consumer-facing applications on the Internet. Although designed with the end user in mind, the protocol is also suitable for corporate use. OAuth 2 provides a mechanism for granting limited access to a resource by enabling the end user to authorise another agent or system to act on their behalf. The protocol is highly adaptable and designed to integrate with other standards, including those within the OAuth family and beyond.

OAuth 2 explicitly includes a user agent notion to facilitate end-user access, and it is compatible with cookies in the HTTP protocol to carry authentication information without compromising the user's privacy. Building on this capability, the OpenID Connect [20] standard allows for authentication and access to basic user information in addition to the authorisation features provided by OAuth.

3.2 User Mediated Access (UMA)

OAuth 2 has a limitation in that it does not provide for permission grants that occur outside of a direct user interaction. This means that a user must initiate the authorisation request and be redirected to provide the authorisation. However, UMA [21, 22] solves this problem by extending OAuth 2 to support authorisation of access to resources by one user on behalf of another, even if the authorisation is temporally separated. For example, a patient could grant a health professional access to their electronic health records. In the context of a process engine, this could involve an authorisation activity where a manager approves an employee to book a specific flight. This capability is significant in process context.

UMA includes two related standards: UMA 2.0 Grant [21] for OAuth 2.0 authorisation, and UMA 2.0 Federated Authorisation [22] for the standardized interaction between resource services and authorisation services. UMA Grant is an extension to the OAuth Grant flow [12] and allows for the asynchronous authentication of users. However, unlike UMA, process-mediated authorisation does not require asynchronous authentication as authorisation is provided by the organization or process model.

3.3 Security Assertion Markup Language (SAML) 2.0

Security Assertion Markup Language (SAML) 2.0 [23] is an enterprise targeted language and protocol that can be used for both centralized authentication and authorisation in a federated manner. It enables users to access third-party systems using their corporate account details while the level of access is specified by the home organization (this is unlike OAuth where uses typically provide their own authorization to other systems). This decouples resource access rules from the resources themselves. SAML is widely used, and it is generally the basis for corporate SaaS login such as to Microsoft Office 365.

SAML 2 and OAuth 2 share some similarities. Both rely on authorisation tokens [24, 25], and SAML specifies its own token format which is transparent, while OAuth allows for both opaque and transparent tokens with arbitrary content. While OAuth can use SAML tokens, the most popular choice is JSON Web Tokens (JWT).

To summarize, federated authorisation, as provided by OAuth and SAML, offers the flexibility to separate authorisation provision from authorisation use, which is a crucial requirement for providing authorization in the context of collaborative processes. These protocols also support the capability of imposing per-token limits on authorisation. However, the downside of these approaches is that they assume a global user-driven authorisation model, even with UMA allowing for authorisation requests outside of direct control by the authoriser. For process-mediated authorisation, a different authorisation approach can be employed using tokens, similar to how UMA adapts OAuth.

4 Applications of Federated Authorisation in Process Context

Federated authorisation based on OAuth and SAML is widely used in process authorisation research. This section evaluates the applications of these protocols in relation to the contribution of this paper.

Chatterjee et al. [26] implemented OAuth 2 to develop healthcare information systems that integrate medical devices and applications in the Internet of Medical Things. The solution ensures secure collection and management of personal health data by encapsulating it into the services for sensitive data. While OAuth secures API access to the system, it does not provide within-process security as provided by process mediated authorisation.

In the context of secure access to resources in a process-based e-learning environment, Politze [27] used OAuth for user authorisation to access these systems and the underlying APIs. While this highlights the relevance of OAuth as an authorisation method for services, the permissions used in this study are derived solely from the user, not the process. The presented extension to the OAuth2 workflow is capable of authorising multiple attached services and thus combining existing services of a central IT service provider, while also allowing other services to run in a cooperative model with only one instance of the authorisation server [28].

To adapt to highly volatile market conditions, companies need to be able to quickly tailor their business processes to changing customer needs. Schäffer et al.[29] present a reference architecture for a process-driven web platform based on the BPMN standard and process engines, with OAuth2.0 chosen as the authorisation protocol. Similarly, in the design of an architecture for integrating production, plant, and enterprise systems, OAuth2.0/OpenID is selected as the authentication, authorisation, and identity management protocol[30]. While these works clearly demonstrate the relevance of OAuth, they do not address how user authentication to the task list and process engine translates to service, data, or resource access control, which is the focus of our work.

Suzic [31] discusses securing the integration of business processes over the cloud and compares OAuth2.0 and UMA from different authorisation aspects, such as client

authorisation, distributed access control, and evaluation of access requests. They describe a prototype implementation [32] that addresses confidentiality and privacy issues of API-based data and resource integrations for workflows over the cloud. The work establishes a common interoperability framework for defining and exchanging service models, policies, and requests in cross-domain collaborations, using OAuth2.0 and UMA for authorisation.

5 Requirements of Process Mediated Authorisation

To ensure privacy and security, it is essential to minimize data access. An effective authorisation system must be easy to adopt and operate, especially in existing environments. Tolone et al. [33] have outlined general requirements for access control in collaborative environments, which serve as a basis for our specific requirements for Process-Mediated Authorisation (PMA).

It is important to clarify that PMA does not pertain to access control to an activity (i.e., who can be assigned an activity). The language/approach used to specify authorisations is mostly out of scope, except that it should allow specifying limited scopes for such scopes to be used.

The specific requirements for PMA can be grouped into four categories. The first category (R1) deals with where authorisations should be specified when they are implied by a process/activity. The second category (R2, R3) focuses on the authorisations that can be provided and allows limits on duration and scope. The third category (R4, R5, R6) outlines how authorisation, once determined, can be used by the process engine in a transparent way, even indirectly. The fourth and final category (R7) pertains to general requirements. In this case, the system must be compatible with existing standards (OAuth, SAML, JWT, etc.), especially those involving services and systems used/invoked by the process.

When performing the activities required by a business process, the actor (service or human) must have the authorisations/permissions necessary to do so. Based on the requirements above, the following list of process-mediated authorisation-specific requirements is proposed:

- R1: Authorisation specification for activity access control is consistent with authorisation to supporting systems and is specified in a unified location (e.g., the process model).
- R2: Authorisations should be able to be limited to the runtime of the initiating activity instance. This means that any permissions should only be available after start and revoked upon completion.
- R3: It should be possible to specify dynamic authorisations that have a scope limited based on the activity context (e.g., only having access to the details of a specific customer).
- R4: The process engine should be able to acquire appropriate authorisation (tokens) that grant the required permissions needed to perform the activity.

- R5: It should be possible to use activity instance-derived authorisation even when services are invoked indirectly. In other words, any secondary authorisation tokens should remain associated with the activity instance.
- R6: For human tasks, the authorisation should be transparent to the user; the effect of limited authorisation can/would be visible, but no explicit user actions should be needed to use the authorisation that was implied by being assigned the task.
- R7: The system must be compatible with existing authorisation systems and authorised services, in particular for invoked automated and interactive services. In other words, it must support existing token formats such as SAML and JWT tokens and protocols OAuth/SAML. The impact on existing systems should be minimal, including the process engine, authorisation service, and worklist.

To illustrate the specific requirements in action, Figure 2 presents a simple process model where a customer calls a customer service representative to update their payment details. The representative needs to access the Customer Relationship Management (CRM) system using SAML-based authorisation (R7) to update the customer's financial information. However, to maintain data access minimization, the representative is only granted access to specific customers (R3) for the duration of the activity (R2), based on the principle of least privilege [34].

The process starts with the customer identification activity, where the representative is given access (R2) to search for users in the CRM system (R4, R6). However, the representative can only search for and review general details of the customer to identify them. Once the customer is identified, the process engine records their ID as the result of the activity instance, and the representative's access to search the CRM is revoked (R2). This ensures that the representative only has access to the necessary information for the task at hand, and not any more than that.

Once the customer is identified, the second activity is to ask a security question to verify the identity. This activity grants the representative access to security questions and other identity verification information, but only for this customer (access to information on other customers is not authorised - R3). The representative uses this information to verify the customer's identity, and upon confirmation, the activity results in a validated customer identity. Once the verification is complete, the representative no longer has access to the security verification details (R2).

The third step is a composite task that provides access to various parallel processes based on the specific user request. In this case, the customer wants to update/verify their payment information, so the representative initiates the appropriate activity/subprocess. The first sub-activity (Create financial update) allows the representative to access the financial details of the specific customer (R3 - for discussion with the customer) and to prepare an update request (but not to directly modify details). Once the update request is completed and sent to the CRM system, it can verify the card details using a delegated authorisation specific to this customer (R5, R3) - this permission is only valid during this activity (R2). To complete the update, the next activity for the representative (Confirm update) requires them to confirm the

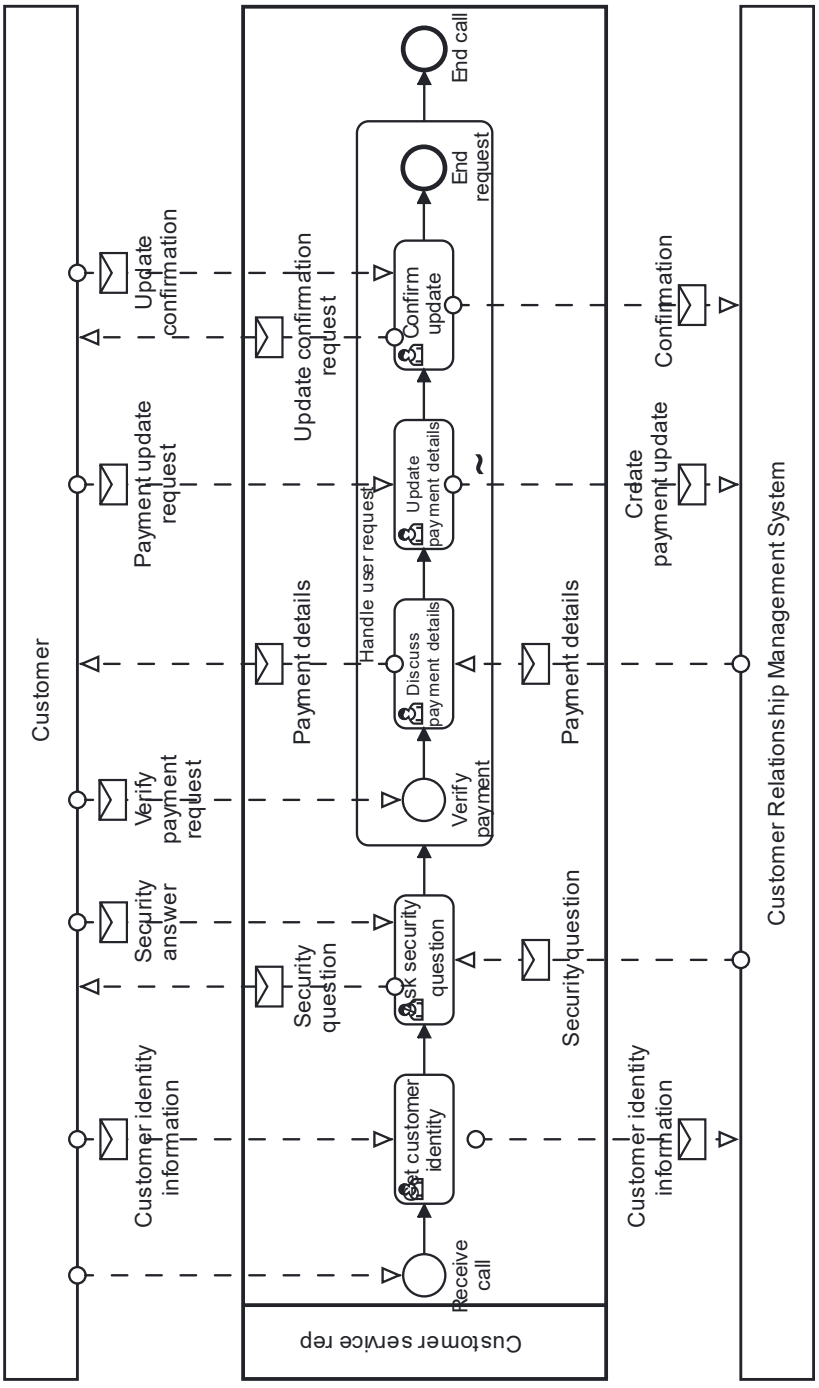

Figure 2 Example of Restricted Authorisation in Customer Service

update with the customer (with authorisation only to see the relevant details or revert

back to the previous step). Finally, the service activity applies the update to the customer record in the CRM.

The authorisations described above are contextual and have specific scopes that depend on the activity being performed. These scopes are linked to the activity context and the access control mechanisms used to assign actors to those activities. Therefore, it is advisable to specify the required authorisations only in the process model (as per R1) to ensure consistency between authorisation and the supporting systems, and to keep all authorisation information in one place.

6 Conclusions

Industry 5.0 and Society 5.0 call for human-centered systems that balance automation with human and societal well-being, but this balance must be achieved in a constantly evolving security landscape that emphasizes security by design. Collaborative ecosystems that can improve and adapt through attacks and disruptions are essential. In Sections 2 through 4, we examined various authorization models, including federated authorization, and their applications in process context with GDPR's data protection requirements in mind. In Section 5, we identified seven requirements for process-mediated authorization: time-limited (R2) and scoped (R3) permissions defined in a unified location (R1) that the process engine (R4) uses for transparent, limited authorization to access interactive (R6) and invoked services, transitive authorization (R5) between invoked services within the process systems, and adoption of existing authorization protocols like OAuth and SAML for PMA implementation (R7). This work lays the foundation for PMA implementation.

References

1. European Union (2016) Regulation 2016/679 (General Data Protection Regulation). Official Journal of the European Union 59:1–88
2. Ferraiolo D, Kuhn R (1992) Role-Based Access Control. In: 15th NIST-NCSC National Computer Security Conference. pp 554–563
3. Hu VC, Ferraiolo D, Kuhn R, et al (2014) Guide to attribute based access control (abac) definition and considerations. NIST Special Publication 800:162. https://doi.org/10.6028/NIST.SP.800-162
4. Thomas RK, Sandhu RS (1993) Towards a Task-based Paradigm for Flexible and Adaptable Access Control in Distributed Applications. In: Proceedings on the 1992-1993 Workshop on New Security Paradigms. ACM, New York, NY, USA, pp 138–142
5. Thomas RK, Sandhu RS (1994) Conceptual foundations for a model of task-based authorisations. In: Proceedings The Computer Security Foundations Workshop VII. IEEE Computer Society, Franconia, NH, pp 66–79
6. Thomas RK, Sandhu RS (1998) Task-based authorisation controls (TBAC): a family of models for active and enterprise-oriented authorisation management. In: Lin TY, Qian S (eds) Database Security XI: Status and Prospects. Springer US, Boston, MA, pp 166–181

7. Thomas R, Sandhu R, Das S (1999) Task-Based Authorisations. Ithaca, NY
8. Oh S, Park S (2003) Task–role-based access control model. Inf Syst 28:533–562. https://doi.org/https://doi.org/10.1016/S0306-4379(02)00029-7
9. Leitner M, Rinderle-Ma S (2014) A systematic review on security in Process-Aware Information Systems – Constitution, challenges, and future directions. Inf Softw Technol 56:273–293. https://doi.org/10.1016/j.infsof.2013.12.004
10. Thomas RK, Sandhu RS (1993) Towards a task-based paradigm for flexible and adaptable access control in distributed applications. In: Proceedings New Security Paradigms Workshop. pp 138–142
11. Thomas RK, Sandhu RS (1998) Task-based authorisation controls (TBAC): a family of models for active and enterprise-oriented authorisation management
12. Jones M, Hardt D (2012) The OAuth 2.0 Authorisation Framework: Bearer Token Usage [RFC 6750]
13. Jin X, Krishnan R, Sandhu R (2012) A unified attribute-based access control model covering DAC, MAC and RBAC. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer, Berlin, Heidelberg, pp 41–55
14. Sandhu R (2012) The authorisation leap from rights to attributes: Maturation or chaos? Proceedings of ACM Symposium on Access Control Models and Technologies, SACMAT 69–70. https://doi.org/10.1145/2295136.2295150
15. Schefer-Wenzl S, Strembeck M (2014) Model-driven specification and enforcement of RBAC break-glass policies for process-aware information systems. Inf Softw Technol 56:1289–1308. https://doi.org/10.1016/j.infsof.2014.04.010
16. Lu Y, Zhang L, Sun J (2009) Task-activity based access control for process collaboration environments. Comput Ind 60:403–415. https://doi.org/https://doi.org/10.1016/j.compind.2009.02.009
17. Jordan D, Evdemon J, Alves A, et al (2007) Web services business process execution language version 2.0 (OASIS standard). WS-BPEL. 1–264
18. OMG OMG (2011) Business Process Model and Notation (BPMN) Version 2.0. Business 50:170. https://doi.org/10.1007/s11576-008-0096-z
19. Parducci B, Lockhart H (2013) eXtensible Access Control Markup Language (XACML) Version 3.0
20. Sakimura N, Bradley J, Jones M, et al (2014) OpenID Connect Core 1.0
21. Maler E, Machulak M, Richer J, Hardjono T (2019) User-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorisation. In: Network Working Group. https://tools.ietf.org/html/draft-maler-oauth-umagrant-00
22. Maler E, Machulak M, Richer J, Hardjono T (2019) Federated Authorisation for User-Managed Access • UMA, 2.0. In: Network Working Group. https://datatracker.ietf.org/doc/html/draft-maler-oauth-umafedauthz-00. Accessed 7 Jul 2021
23. Lockhart H, Campbell B, Ragouzis N, et al (2005) SAML v2.0 Technical Overview

24. Rui He, Man Yuan, Jianping Hu, et al A novel service-oriented AAA architecture. In: 14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. IEEE, pp 2833–2837
25. Gommans L, Travostino F, Vollbrecht J, et al (2004) Token-based authorisation of Connection Oriented Network resources. In: GRIDNETS conference proceedings. System and Network Engineering (IVI, FNWI), Amsterdam
26. Chatterjee A, Prinz A (2022) Applying Spring Security Framework with KeyCloak-Based OAuth2 to Protect Microservice Architecture APIs: A Case Study. Sensors 22:1703. https://doi.org/10.3390/s22051703
27. Politze M (2019) A reference architecture and implementation enabling data protection in distributed learning and science processes. RWTH Aachen University
28. Politze M, Decker B Extending the OAuth2 Workflow to Audit Data Usage for Users and Service Providers in a Cooperative Scenario. In: 10. DFN-Forum Kommunikationstechnologien. Gesellschaft für Informatik eV.
29. Schäffer E, Schobert M, Reichenstein T, et al (2021) Reference Architecture and Agile Development Method for a Process-Driven Web Platform based on the BPMN-Standard and Process Engines. Procedia CIRP 103:146–151. https://doi.org/10.1016/j.procir.2021.10.023
30. Karadimas D, Panagiotou C, Gialelis J, et al (2021) Process based Machine Learning for Energy Optimization in Industrial Enterprises. In: 2021 10th Mediterranean Conference on Embedded Computing (MECO). IEEE, pp 1–4
31. Suzic B (2016) Securing integration of cloud services in cross-domain distributed environments. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing. ACM, New York, NY, USA, pp 398–405
32. Suzic B (2016) User-centered security management of API-based data integration workflows. In: NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium. IEEE, pp 1233–1238
33. Tolone W, Ahn G-J, Pai T, Hong S-P (2005) Access control in collaborative systems. ACM Comput Surv 37:29–41. https://doi.org/10.1145/1057977.1057979
34. Saltzer JH (1973) Protection and control of information sharing in multics. In: Proceedings of the 4th ACM Symposium on Operating Systems Principles, SOSP 1973. p 119