PLATTFORM INDUSTRIE 4.0

DISCUSSION PAPER

Access control for Industrie 4.0 components for application by manufacturers, operators and integrators

Imprint

Publisher
Federal Ministry for Economic Affairs
and Energy (BMWi)
Public Relations Division
11019 Berlin
www.bmwi.de

Editorial responsibility
Plattform Industrie 4.0
Bertolt-Brecht-Platz 3
10117 Berlin

Design
PRpetuum GmbH, Munich

Status
September 2019

Image credits
gilaxia – Getty Images (title)
Monty Rakusen – Getty Images (p. 4, 12)
Ralf Hiemisch – Getty Images (p. 8)
spainter_vfx – iStock (p. 15)
Fertnig – Getty Images (p. 16)
Erik Isakson – Getty Images (p. 23)
alengo – Getty Images (p. 31)
Caiaimage/Rafal Rodzoch – Getty Images (p. 38)
Vertigo3d – Getty Images (p. 40)

You can obtain this and other brochures from:
Federal Ministry for Economic Affairs
and Energy (BMWi)
Public Relations
Email: publikationen@bundesregierung.de
www.bmwi.de

Central ordering service:
Tel.: +49 30 182722721
Fax: +49 30 18102722721

This brochure is published as part of the public relations work of the Federal Ministry for Economic Affairs and Energy. It is distributed free of charge and is not intended for sale. The distribution of this brochure at campaign events or at information stands run by political parties is prohibited, and political party-related information or advertising shall not be inserted in, printed on, or affixed to this publication.

Content

Preamble

To fulfil the requirements and goals for cybersecurity, it is particularly important that manufacturers, operators and integrators ensure secure access control for Industrie 4.0 components. The access control which is necessary for Industrie 4.0 goes significantly beyond the previous boundaries of local systems. This is especially challenging for small businesses, so they should be offered help to develop a right understanding of access protection for equipment and systems by using modern access control features.

This discussion document contains not only technical considerations, but also up-to-date statements about secure cooperation between manufacturers, operators and integrators to ensure the secure use of these technologies in cyberspace. The following requirements are fixed elements in current national, European and international agreements on "security in cyberspace", so it is worth reminding ourselves of their content:

- Security in cyberspace plays a decisive role in every aspect of our lives. A large number of stakeholders have a shared responsibility to act in their respective roles to improve trust, security and stability in cyberspace.

- In this spirit, a peaceful cyberspace is necessary to ensure an open, secure, stable, accessible and integral environment for all aspects of our economic life, especially the secure use of information and communication technology (ICT) in Industrie 4.0.

- The threat of cybercrime requires greater efforts to improve the security of the products which are used, to enhance defences against criminal acts and to promote cooperation between all parties, both nationally and beyond national borders. The Budapest Convention on Cybercrime of 2001, in combination with the final report of the EU Commission of 2017 [1], is an important instrument in this respect. Its relevance to Industrie 4.0 must be noted.

- Trust, security and stability in cyberspace strengthen the security of digital processes, products and services. Cybersecurity standards can help infrastructures and organisations to improve cyber protection. Digital cooperation must strengthen the efforts of all stakeholders to develop their capacity, and it must also promote initiatives to increase the resilience of the digital infrastructure. The goals of Industrie 4.0 in this respect are:

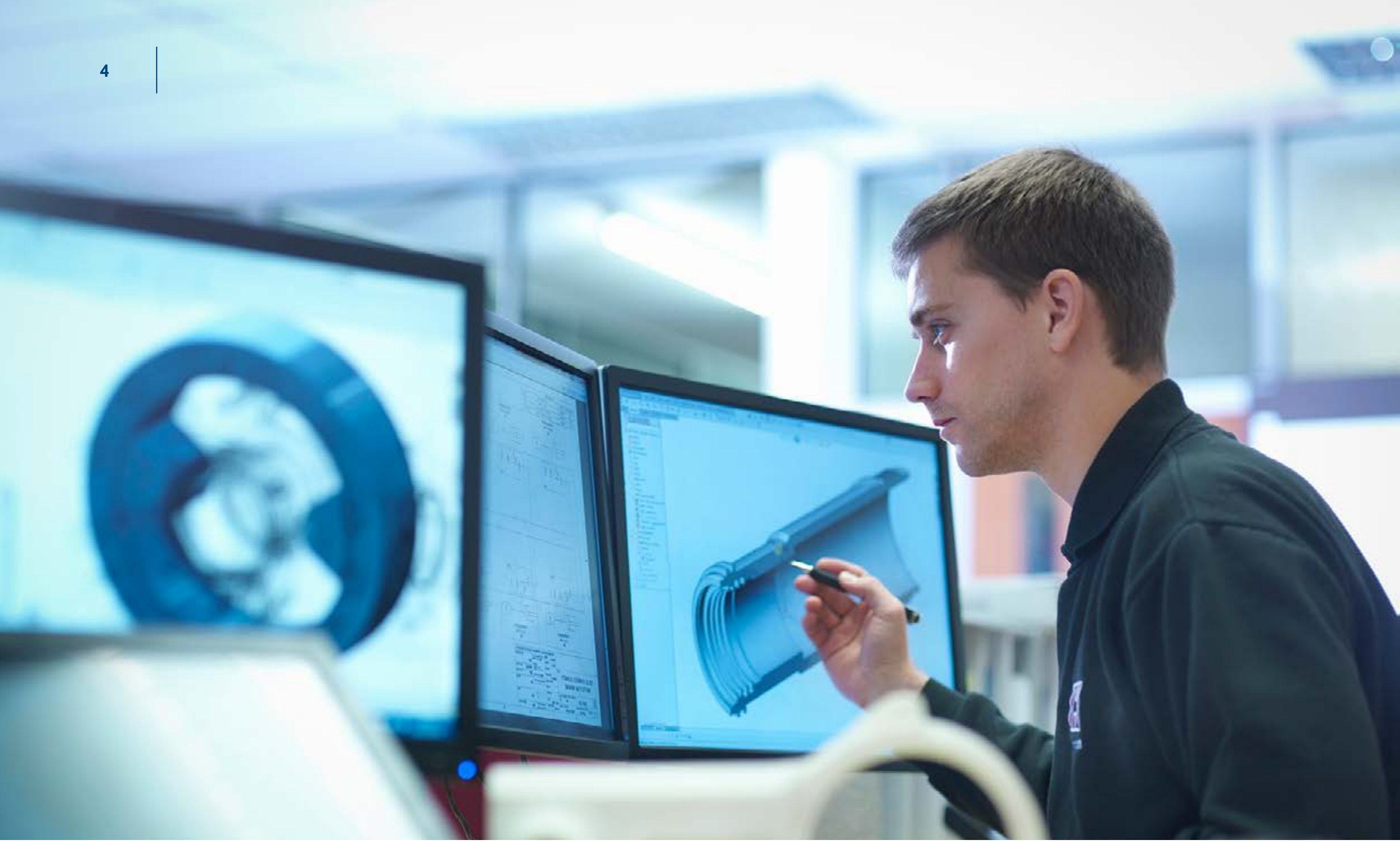
  - To prevent the theft of intellectual property by ICT, including business secrets or other confidential business information, which aims to gain competitive advantage for business companies or the commercial sector

  - To develop ways to prevent the spread of malignant ICT tools and practices which are designed to have harmful effects

  - To strengthen the security of digital processes, products and services throughout their whole life cycle and the total supply chain;

  - To promote the broad acceptance and implementation of international standards for responsible action and confidence-building measures in cyberspace.

This discussion document takes up the frequently discussed issue of the necessary and suitable access protection for Industrie 4.0 components and discusses known systems and new technologies.



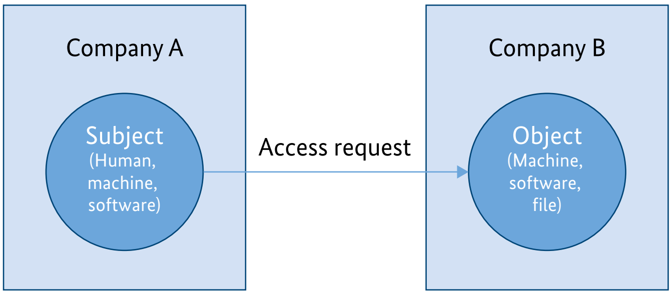
1 Introduction

Industrie 4.0 demands a high degree of interoperability in the use of networked systems which operate autonomously. The volume of digital data is growing, and the proportion of digitally stored data is also increasing continuously. As a result of this constant growth in the available digital data, there is an increasing interest in sharing such data between different stakeholders and working together to create or use the data ("collaboration") in order to improve factors such as the effectiveness and efficiency of production.

Access needs to be on an inter-company basis (Figure 1) and not only limited to the secure internal environment of an individual company as it was in the past. This creates a demand to provide appropriate access to the data (objects, cf. Figure 1), but only for the entitled stakeholders (subjects – cf. Figure 1).

The requirements for access control mean that an efficient exchange of information with other companies must be possible, but without risking any undesirable use of the data. Secure access control is therefore of central importance in secure cross-company communication. Easy application in networked systems must especially be ensured for small companies. The need for coordination between the stakeholders must be kept as simple and economical as possible.

Figure 1: Fundamental communication



Precise access control is needed, sometimes even down to the file level, to facilitate fast business transactions. At the same time, complex and sometimes contradictory rights must be avoided to minimise the risks that may arise from errors and fraud. In addition, the measures which are implemented must be correct and verifiable to satisfy the regulatory requirements.

In the light of these requirements, the access control measures which are currently established in industrial environments are not sufficient. This discussion document analyses the advantages and disadvantages of the various processes and the data modelling for a concept for efficient access control.

1.1 Access control for Industrie 4.0 components

This leads to the question: ”What must access control achieve in the context of Industrie 4.0?”

Access control has a general function of protecting trust, integrity and availability. It ensures that unauthorised parties cannot access the information, change information or compromise the system to make it unavailable or prevent it from working properly.

In future, Industrie 4.0 components will be assigned features such as their own electronic identity features, known as IDs. This applies to both hardware and software components. In other words, Industrie 4.0 components will become uniquely identifiable entities just like people.
For this reason, in the following text the concept of an **Industrie 4.0 entity** will be used as a comprehensive term for all communications partners.

For secure inter-company communication, for example in the course of the acquisition, commissioning or remote maintenance of Industrie 4.0 components, automatic access control processes take place between the Industrie 4.0 entities of different companies. Access control between Industrie 4.0 components is carried out in a horizontal networking process directly at the level of the entities. This includes both machine-machine communication and human-machine communication (M2M/H2M).

The increase in communication partners, especially partners from other security domains with their own specific administration, brings new requirements for access control. This includes access requests from unknown subjects in known domains, different users such as humans and machines from other domains, a constantly growing number of Industrie 4.0 entities which can be accessed and the need for granular control of access to objects, all of which is needed to ensure data exchange in accordance with the protection goals for “CIA” (confidentiality, integrity, availability).

The new requirements for Industrie 4.0 require a review of the user role access right models which were predominantly used in the past (RBAC, role-based access control). This concept was originally created in the 1990s for use between the known users of the machines. From the perspective of the access control system, the users, i.e. human beings, are Industrie 4.0 entities.

In future, the communication between Industrie 4.0 components will use their different administration shells [6] [11] [12]. Therefore the comments on access control apply to concepts for application in the administration shell.

1.2 Trust in inter-company relationships

However, trust for communication partners usually ends when the partners are outside the context of the individual company. To facilitate global relationships of trust, distributed access control systems are all the more dependent on **trustworthy** authentication of the communication partners.

For this reason, access is frequently via “authentication providers” (trusted third parties) which are trusted by each of the parties. This concept means that the subject characteristics can be credibly demonstrated (e.g. identity, subject attributes, integrity of the Industrie 4.0 entity).

If no authentication provider is used, the challenge is that the access information, e.g. the user/password, subject attributes, security tokens, certificate information etc., must be known to all communication partners in a semantically identical form.

But up to now, the decision about the allocation of object rights (authorisation) has not normally been outsourced to an external service such as an authentication provider which is outside the context of the individual company.

The subject of trust relationships in access control is treated in more detail in chapters 2 and 3.

1.3 Role-based access control

With regard to the role based access control system (RBAC [2] [3]) that is still often used, the simple approach of a rights structure based on user roles is generally emphasised. But this mainly applies to closed systems which do not require a permanent exchange of information with other closed systems. More about the details can be found in Chapter 2.2.

This RBAC method is known as a standard access method e.g. for Microsoft-based workplace systems (PCs). However, in a shop floor context, by contrast with the office area, users are often issued with standard roles or profiles and therefore also with standard passwords which are often generally known (user admin, password admin), e.g. to prevent any risk to the production facilities in case the machine user is unable to take any necessary action because of a lack of access data/permissions.

The user in Figure 2 represents a subject who is allocated a role and an access right to a component, machine or process, i.e. an object (cf. Figure 1). In the following text, these concepts are mainly used to explain access control. In the literature, the term resource is sometimes used instead of object.

The above granularity which is necessary for Industrie 4.0, and the flexibility of the access control are not reached by this process.

1.4 Attribute-based access control

In addition to the RBAC concept, Chapter 2 explains a process which is relatively new for industry, attribute based access control (ABAC [4][5]).

The purpose of this access control system is also to grant or deny the relevant access rights for an Industrie 4.0 component, i.e. an object, to an Industrie 4.0 entity, i.e. the requesting subject (Figure 3).

After successful authentication, an authorisation is issued ("the subject receives rights to the object"). After the positive authorisation, the subject can access the object with the appropriate rights ("the subject can carry out operations for the object in the framework of the assigned rights").

The significant difference from RBAC is that although the subject must have valid credentials for the authentication (= log-in data: proof or certificates of the right to credit, trust or authority), the subject does not need to be previously known to the object. This is possible because the object does not assign a role to the subject (i.e. no static user role tables are maintained) and the access does not need to be decided globally merely on the basis of the possession of a role issued by the company of origin. Instead, the access can also be made dependent on other subject,

Figure 2: RBAC – a flat role-based access rights structure

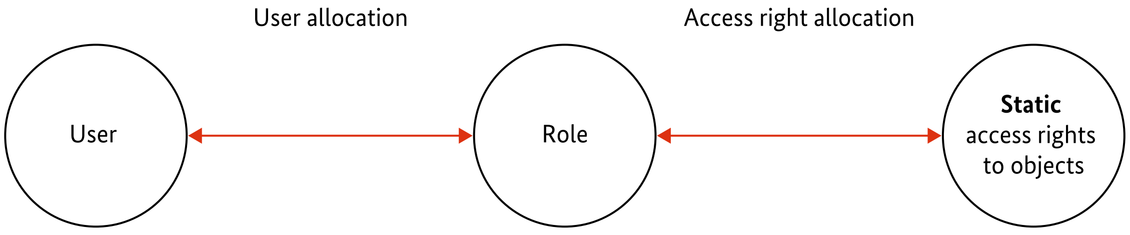
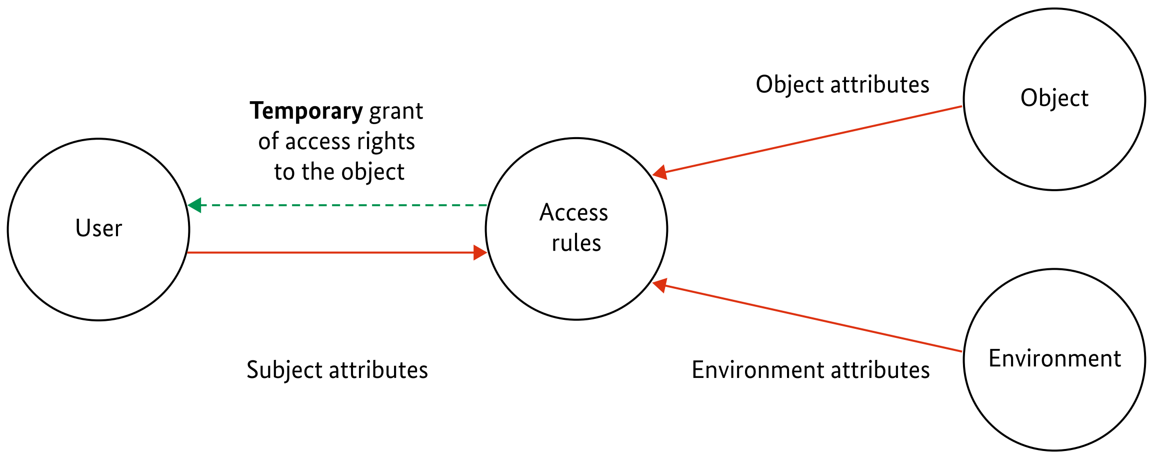


Figure 3: Fundamental ABAC concept



object and environment attributes. Examples of further-reaching subject attributes could be the qualification level or proof of security training. The object attributes include operating modes and machine statuses, whereas the environment attributes comprise factors such as the time. The details are explained in Chapter 2.

A rule may consist of an IF THEN formula. An example could be as follows: "Access to the system control is permitted if the subject belongs to the organisation of the control manufacturer and has a qualification for parameterisation of the control. At the same time, the machine must be in maintenance mode and the production must have halted according to the operator's shift plan." To this end, it must be possible to access the subject attributes (ID, qualifications), the object attributes (operating mode of the machine) and the environment attributes (time, shift plan). These attributes may be stored either inside or outside the administration shell.

Remark: The identity management of subjects and objects will not be treated in more detail in this document. The analysis in this document will assume that there are previously identified and identifiable "secure identities". Greater detail can be found in the documents of the subsidiary working group "Secure Identities". In addition, a number of standard identity technologies will be mentioned in the Appendix to Chapter 5.1.

1.5 Document structure

This document consists of the following chapters:

Chapter 1 presents the scope and purpose of the "Access control for Industrie 4.0 components for application by manufacturers, operators and integrators" and the methodology.

Chapter 2 explains the concepts of RBAC and ABAC in **Main section A**. This part is designed for conceptually interested readers.

Chapter 3 explains the practical implementation and applicability of ABAC and the mapping of roles and rights in **Main section B**. In addition, the UML diagrams relevant to security in the document "Administration shell in detail (VWSiD)" are explained and illustrated by reference to examples. And applicability in the environment of the Open Platform Communications Unified Architecture (UPC UA) and International Data Space (IDS) is presented. This part is designed for technically interested readers.

Chapter 4 provides a summary and an outlook.

Chapter 5 contains the Appendix with explanations of identity technologies, the glossary, historical information, the references and the editorial team.

2 Discussion of the concepts RBAC and ABAC

2.1 Requirements for the "security of the administration shell"

The basis for the considerations is a simple use case which describes the commissioning of three components on the premises of one operator. The components were supplied by different manufacturers. In a detailed analysis of the process steps "connect to the network", "taking possession by the operator" and "autonomous performance of a higher value process" in this use case, the relevant security requirements for the administration shell itself [6] and for the sub-models of the administration shell were derived.

Four fundamental requirements were identified:

- Identities and authentication:
  "It must be ensured that the right entities interact with each other. This applies both in a local communication context (within a machine or system) and in a more global context (inter-company cooperation). The unique identification of the communication partners (by authentication) is a fundamental requirement for interaction with a management shell. Without this identification, other security features (confidentiality, integrity etc.) cannot be guaranteed."

- Access control:
  "An administration shell of an I4.0 component can have different interaction partners. Access management is necessary to control the possibilities for interaction with the administration shell. The access administration consists of the administration of subjects and permission rules. An administration shell must include local basic access management, which should be replaced by a larger system or can be extended by adding external access management. A fundamental policy enforcement point (PEP) in conjunction with a policy decision point (PDP) is necessary in order to verify the permission rules, e.g. the access rules stored in a local access control repository. The access administration must be part of the administration shell life cycle management (and not a static configuration) and should make allowance for the transitions, e.g. from a manufacturer to an integrator (machine tool manufacturer) or from an integrator to an operator."

- Communication security:
  "Communication with the administration shell may contain confidential information. Similarly, a change in the data (integrity) in the communication between the administration shell and its communication partners can lead to serious and dangerous disruption in a machine or system. Therefore, suitable measures must be taken to guarantee the security of the communication. This must be achieved by using suitable security protocols."

- Logging of results:
  "The traceability of the interaction with the administration shell plays a decisive role in the detection of security incidents. This traceability is achieved by logging/recording results and by auditing. The administration shell must therefore provide methods to create uneditable records of access transactions and changes in the status of the management shell. It is also important to ensure that this information about the results can be centrally collected and evaluated."

The requirements for access control will now be discussed in greater detail.

2.2 Access control by RBAC and ABAC

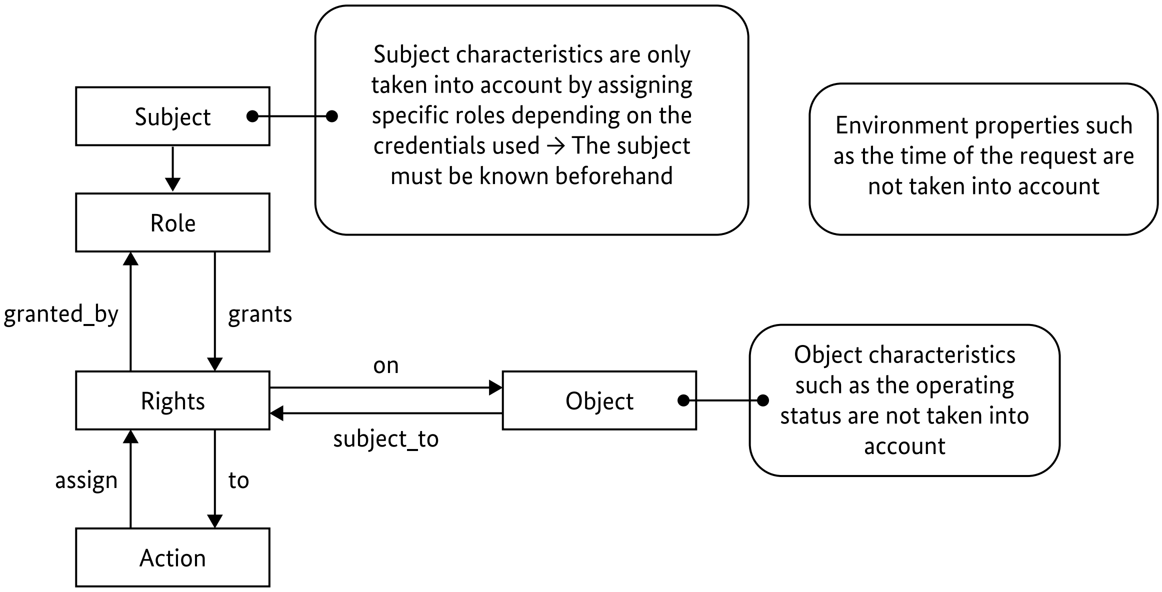
Both RBAC and the ABAC concept serve to grant subjects rights to specific actions, especially in digital processes, or to limit such rights.

The most important characteristics of RBAC and ABAC will now be discussed in the context of Industrie 4.0, and then a comparative analysis of RBAC and ABAC with their strengths and weaknesses will be presented.

RBAC

Role-based access control [3] is a method which has developed historically since the 1970s and seen continuous adjustments, e.g. to cope with web and cloud applications. The central concepts here are the role (a description of the special actions which a user may perform) and rights (the permissible or forbidden methods to carry out the action). Other concepts such as a group (a defined number of users) serve to simplify the administration.

Figure 4: RBAC model


Subject
Role
Rights
Action
Object
granted_by
grants
assign
to
on
subject_to
Subject characteristics are only taken into account by assigning specific roles depending on the credentials used → The subject must be known beforehand
Environment properties such as the time of the request are not taken into account
Object characteristics such as the operating status are not taken into account

The method shown in Figure 4 begins by creating a role which brings together the permissions to access an object. This role is subsequently assigned to one or more users (subjects) or a group of users. Other uses can then easily obtain the same role, or the rights can easily be withdrawn by removing the role assignment from an affected user. The object characteristics such as the operating status, or environment properties such as the time of the request, are not taken into account. This means that RBAC is a fairly rough method which can be used to administer certain areas easily.

The following four classes are defined in RBAC:

Action → Describes the possible actions, e.g. reading, writing

Resource → Describes the object.

Privilege/right → Assigns individual rights to a role, e.g. reading, writing, deleting

Role → Describes the role

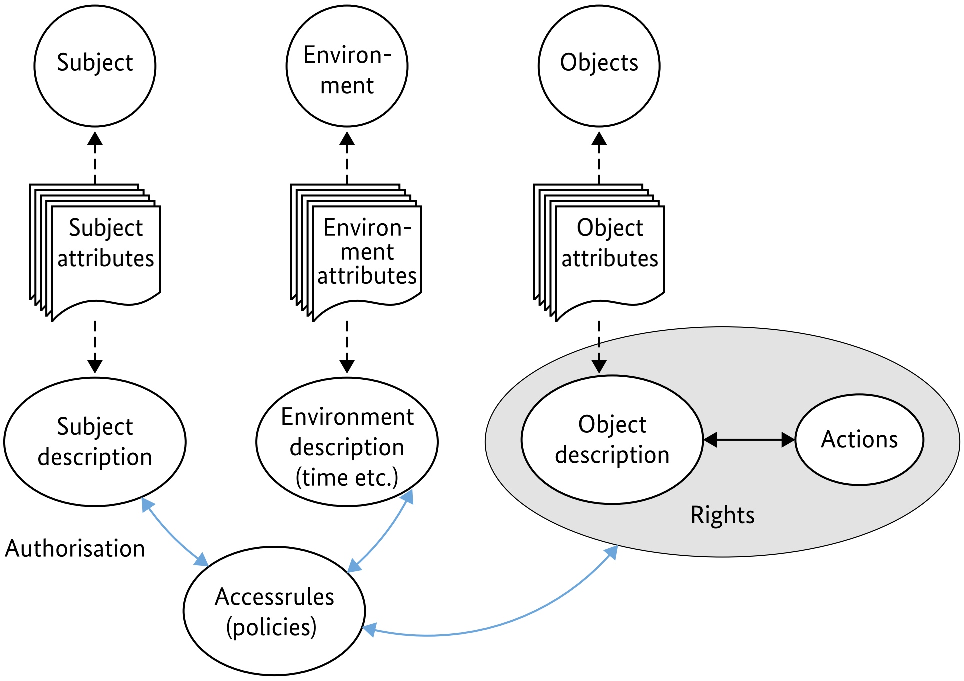
One known application is the assignment of rights in file systems, where files and folders can be created, edited or deleted depending on the role. RBAC is also used in the administration of cloud storage.

Therefore, with RBAC it is necessary to assign the appropriate rights to the subject before the subject can access the system. In companies this is usually initiated by a request. After the request has been cleared, the user is entered into the "RBAC system", a role is assigned and the corresponding rights to an object are granted for this role. This is generally carried out by an administrator or by an automated process. In ABAC, on the other hand, no explicit rights must be assigned to individual users before they can access an object.

ABAC

Attribute-based access control [5] is an extended method which not only takes the attributes of the subject into account, it also considers attributes of the objects and the required environment conditions when checking

Figure 5: Basic ABAC model



the access rights (cf. also https://csrc.nist.gov/projects/attribute-based-access-control). These attributes may be centrally stored, e.g. in a local area network in an open directory (cf. "administration shell"), or locally linked to the Industrie 4.0 entity. The allocation of subject attributes to a specific Industrie 4.0 entity is ensured by an ID management system (IDM) (identification, authentication). The correctness of the attributes at the time of issue can be verified by mathematical methods (hash, signature etc.). These methods can also determine and check the identity of the issuing body. Attributes with these values are known as meta-attributes. It is also possible to use "dynamic factors" as attributes for access control. Environment attributes such as time, place or security requirements can be used to adapt the access conditions for an Industrie 4.0 component so that they precisely match certain environment conditions.

No direct relationships are defined between the subject and the object as in the RBAC system, permissions are allocated on the basis of more extensive access rules (policies).

The rules can be used to achieve a very high granularity for the access control. And existing RBAC and other access control systems can be connected to ABAC.

Initially, more administration work is involved in implementing and setting up the ABAC system. But the later administration is then much easier because, by contrast with RBAC, no new rules need to be created when new subjects and objects are added unless the types of access change.

This is the significant advantage. Whereas in RBAC every subject must be known so that it can be assigned a specific role, which may even need to be newly defined, this is not necessary in the ABAC system because the rights are assigned to the subjects by means of rules on the basis of their existing attributes and the current status of the objects and the environment.

Especially with the distributed systems of Industrie 4.0, the possible subjects and conditions under which a specific access right is granted may not be known at the time when the system is set up. This means that it is only with ABAC that the necessary flexibility of the access control is achieved to support a dynamic group of subjects. By contrast with access control based on a defined role, an ABAC object can itself contribute the target attributes which will permit access (e.g. a specific subject certificate). Otherwise, external administration of the adaptation of roles is necessary, but here it can be eliminated. This considerably reduces the amount of administrative work.

Conflicting rights are easier to avoid with ABAC than with RBAC, and they can be more quickly solved. For example, a system which requires each step to be monitored by 2 people can easily be wrongly configured with RBAC and then accessed by unauthorised persons, and the error is then very difficult to detect and correct.

(Cf. also http://blog.identityautomation.com/rbac-vs-abac-access-control-models-iam-explained and https://pdfs.semanticscholar.org/7750/f0bffaff9c3bb66fa7b8dfef2f-46daf0525e.pdf)

2.3 Example scenario: remote maintenance

A service technician from company A needs to carry out remote maintenance on an Industrie 4.0 entity in company B. Access should only be possible if the Industrie 4.0 entity is in 'MaintenanceMode' and it takes place in the time between 15:00 and 17:00 hrs. In addition, access should only be permitted if the service technician has previously completed the self-assessment security check which is required for external access.

2.3.1 Implementation with RBAC

With RBAC-based access control, the service technician as the requesting subject is assigned one or more roles which should fulfil the following tasks:

- Role 'RemoteMaintenance': This contains the general permission for the assigned user to access the Industrie 4.0 entity and change specific parameters or properties.

- Role 'TimeAuthorisation': This is assigned to the user for the exact period for which the permission is to be granted. After that it must be removed – this can be done manually or by identity management. Alternatively, the role can be assigned a validity period if appropriate.

- Role ‘MaintenanceModeAuthorisation’: The role is assigned to the user as soon as an authorised party has set the machine status to ‘MaintenanceMode’. It is removed again when the status is changed.

- Role ‘SecurityCheck’: The role can be assigned to the user after presentation of the test certificate for the period of validity of the test certificate.

There is a direct dependence between the creation of the role and the awareness of the role in the Industrie 4.0 entity. A central permissions check within the Industrie 4.0 entity may not be able to check all roles and their associated authorisations, for example it may only be possible to check the role ‘RemoteMaintenance’ centrally, but special rules (e.g. a check for the assigned role ‘MaintenanceModeAuthorisation’) would possibly need to be checked by various individual components of the Industrie 4.0 entity, because only they know the role.

To permit a rather more dynamic check, annotations could be added to the special roles, for example to permit a time check by individual components via the note ‘15:00 – 17:00’ within the role ‘TimeAuthorisation’.

A later extension to incorporate unknown roles is difficult. The introduction of a role ‘SecurityCheckPlus’ would not be possible because none of the local components of the Industrie 4.0 entity would be equipped to check this role.

Conclusion: A purely RBAC-based permission check is possible for the example considered here, but it has many obstacles because dynamic environment checks can only be implemented with workarounds.

2.3.2 Implementation with ABAC

In ABAC, a rule (policy) is created which reflects the conditions of the check. The access control mechanism then checks whether the required conditions are fulfilled. In the example above, it would check whether the role ‘MaintenanceEngineer’ is set, the parameter ‘SecurityCheck’ has the value ‘true’, the mode of the machine has the value ‘MaintenanceMode’ and the current time is between 15 and 17 hrs.

To this end, the subject, object and environment attributes are provided to the access control mechanism:

- Subject attributes (provided by the requesting subject): role of the subject ‘SecurityCheck’ = ‘true’ / ‘false’

- Object attributes: mode of the machine

- Environment attributes: current time

If all conditions are fulfilled, the rule gives the value ‘true’ which leads to the instruction ‘allow access’ as the result of the check, and the access by the user can then be granted.

The rules can be centrally or globally formulated and distributed between the various Industrie 4.0 entities, depending on the policies of the company.

With ABAC it must therefore be ensured that the appropriate attributes for the rules are accessible and that they have previously been created in accordance with the policies of the company and passed on to the access control mechanism.

The check can be dynamically extended or replaced (e.g. it could also react to an additional user attribute ‘SecurityCheckPlus’). As a result of the ABAC approach, the Industrie 4.0 entity basically has an intrinsic ability to carry out any checks.

By contrast with the RBAC-based approach, the role as a self-stated subject attribute can be dynamically interpreted by the access control mechanism (e.g. specifically for the Industrie 4.0 entity). For example, the same attributes may lead to other or different permissions in an Industrie 4.0-capable 3D printer than the permissions which apply in an Industrie 4.0-capable power drill.

2.4 Comparison of RBAC and ABAC

An increase in the number of users (subjects), roles, objects and the necessary granularity of the rights leads to greater complexity in the rights structure and the amount of work involved in administering the permissions. This means that RBAC only scales within certain limits, and only within a company context. Implementation beyond the bounds of the company is more difficult because the identities and roles must be unambiguous to enable the roles and rights to be mapped.

In RBAC, the identity is represented as a role (e.g. accounting) and rights are assigned to the role (e.g. to the different systems in the accounting department). With different roles which cannot be represented together (e.g. accounting and the administrator of the accounting system), a user must be assigned several identities to prevent the misuse of rights. In addition, granular rights can only be achieved by using additional roles, which not only increases the number of roles – the sum of the rights of two roles may even provide undesirable combined rights. If this can be solved at all, it can only be done by extra roles (e.g. an accountant should be entitled to create records for all creditors and debtors; an accountant should be entitled to close outstanding payments of all debtors; however, an accountant should not be entitled to close the outstanding payments of debtors which he has created in the system – this example is extremely complicated to represent with roles alone).

RBAC can only allow or deny an access right as a whole. A reduction or extension of the rights in individual cases is not anticipated.

Special challenges also arise if roles must be defined which themselves have "sub-roles" creating a system of "nested roles". Understanding the assignment of rights of this type becomes confusing after a certain time, especially if several changes have been made, and in the long term a direct 1:1 allocation of permissions would be far more transparent.

RBAC is relatively static because roles are only adapted to identities when changes are made (e.g. a change of jobs). It is also static in the context. If the identity is proved (authentication process), the authorisation is granted. No other parameters are considered.

As was already mentioned, and in contrast to RBAC, ABAC does not require the provisioning of a user account for the subject. This means that ABAC is dynamic in several ways because authorisation does not necessarily require proof of the user's own identity because other attributes (e.g. the production area of the machine and the user's own department) can be used instead. This makes it easier, for example, to give visitors access rights for the duration of their visit without having to create a record for them in the identity management system. Eliminating the requirement for user account provisioning also enables access to be arranged between companies without any need to match the relevant identities.

The use of attributes in the access rules as a basis for authentication and authorisation permits a highly granular assignment of rights – and it can even accommodate different results in repeated authorisation cases. The accounting example from the section on RBAC can very easily be managed with ABAC. At first sight the rules may appear to be complicated, but they offer easier updating and verification than decentrally distributed roles and rights which are assigned identities in a different place. And to a certain extent, the rules can be used for automated intrusion detection by "user behaviour analytics" to recognise deviations from the normal behaviour of a user and thus to carry out intrusion prevention to avert threats before they even arise. Cf. also Chapter 5.3 "Access control RAdAC solutions".

2.4.1 Overview of the advantages and disadvantages of the access control concepts

	Attribute Based Access Control (ABAC)	Role Based Access Control (RBAC)
Access by unknown subjects with a valid authentication	++	--
Work involved for the administration of existing Industrie 4.0 entities when new Industrie 4.0 entities are integrated into the communication	++	--
Work involved in initial administration (definition of access rights)	-	+
Possible anonymity of the requesting subjects	Yes	No
Possible granularity of the access control	++	+-
++ Very good, + Good, +- Satisfactory, – Adequate, -- Poor		

2.4.2 Conclusion

In Industrie 4.0, far greater numbers of complex objects (identities) must be administered, and the networked structures can often no longer be represented on a top-down basis or as a tree structure. Innovative relationships between the parties form a veritable mycelium of communication channels. ABAC attributes may also include authentication features from an IDM (ID management system), so the boundaries between the Industrie 4.0 entities object/subject and identity must be rethought.

One of the greatest challenges for **RBAC** is the fact that the function of a person and the roles do not always correspond because RBAC does not map the company structure. All attempts to find a solution to this problem lead to unclear permission structures which cannot be sensibly administered and are sometimes uncontrollable. Therefore a new approach is needed which enables the interconnections between objects and subjects to be mapped.

This applies to the individual Industrie 4.0 entities (object, subject etc.), their organisational interaction in different domains (structure) and their mutual and retrospective relationships (object to subject, customer = supplier?)

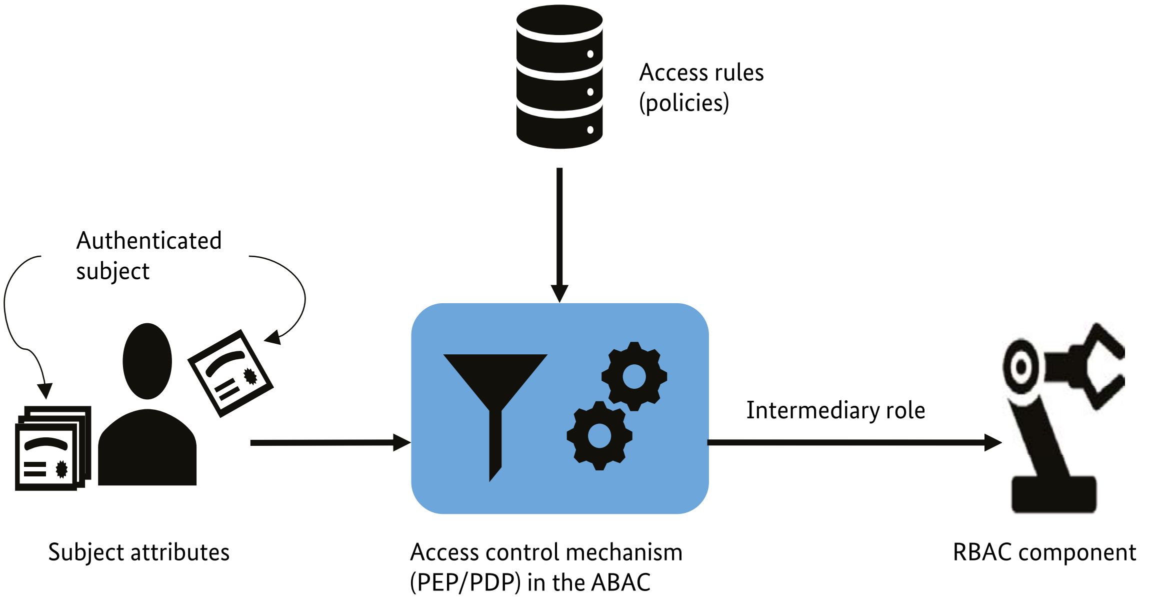
ABAC is a step in this direction because it facilitates the possible diversity of attributes and a much more detailed representation of the participating objects/subjects. And ABAC permits the use of these attributes across domain, company and national boundaries, even though it also requires extra work to standardise the intersecting elements in a non-local environment.

2.5 coexistence of RBAC and ABAC connection of RBAC components in an Industrie 4.0 context

As explained above, ABAC fulfils the requirements which arise from Industrie 4.0 better than RBAC. However, because of the networking of the Industrie 4.0 entity, different access control systems such as ABAC and RBAC must be able to interact.

With attribute-based access control, it is possible to integrate existing standards such as RBAC and proprietary protocols which are already used in the Industrie 4.0 infrastructure, and thus to enable overall access management to be created.

Figure 6: Scenario to integrate an RBAC component into an ABAC system



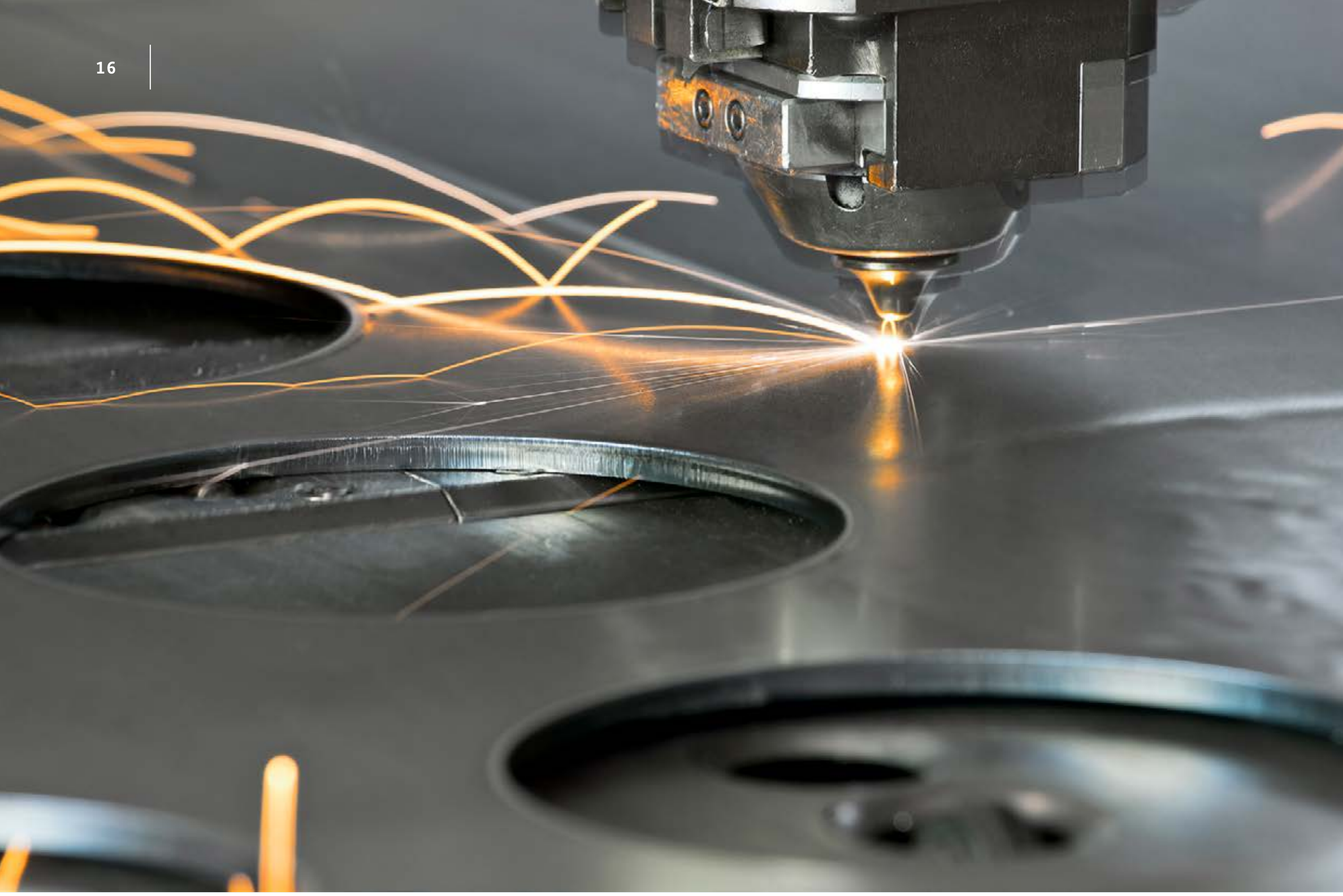
In Figure 6 the integration of an RBAC component into an ABAC system is shown. The assignment of the access rights is carried out on the RBAC component. The subject attributes of the requesting subject are transmitted to the access control mechanism in a sort of container, known as the credential, and the machine then initially checks whether the credential is compatible with the set of rules. Then the rules are checked against the subject, object and environment attributes, and if appropriate a corresponding intermediary role is presented to the RBAC component. Using this process, it is possible to extend an RBAC component by adding the flexibility or granularity of access control by the ABAC concept. A device which directly supports ABAC would transmit its manufacturer-specific set of rules to the PAP of the ABAC system. It would then no longer be necessary to transpose the subject attributes into an intermediary role. In addition, the specific configuration of the values is stored in the issued values specification. The evaluation specifications define which attributes should be used for evaluation.

To implement access control irrespective of manufacturers, it is necessary to introduce a consistent and descriptive language such as the language introduced in the publication "Administration shell in detail" (VWSiD) [9]. Here the data types, attributes, rules for access decisions and permissions are modelled.

The keywords used are described in a names directory for each device model or proprietary protocol. Each Industrie 4.0 entity that is described contains a reference to the applicable names directories, which can be stored in a local intermediate storage memory to limit and bring together the access to external resources.

By importing the applicable names directories, the access control mechanism can implement rules for different devices and can read and jointly process data drawn from different sources.

In the long term, the goal should be to create standardised name directories for different device categories to enable the plug-and-play integration of conforming devices from different manufacturers without extra configuration work, at least for the basic functions.



3 Practical implementation and applicability of ABAC

3.1 Process sequence of the ABAC mechanism

The consideration of the access management begins with the authenticated subject. This means believing the subject's statement, i.e. that the subject is who it claims to be. The subject can be any Industrie 4.0 entity (human, machine, product etc.). The subject has a number of attributes (attribute container – subject attributes). Several attribute containers may be provided by several signatories which conform with the newly defined "Industrie 4.0 class" and which supply attributes. The signature of the attribute container gives an assurance that the attributes come from a specific source and have remained unchanged (authentic) since they were issued.

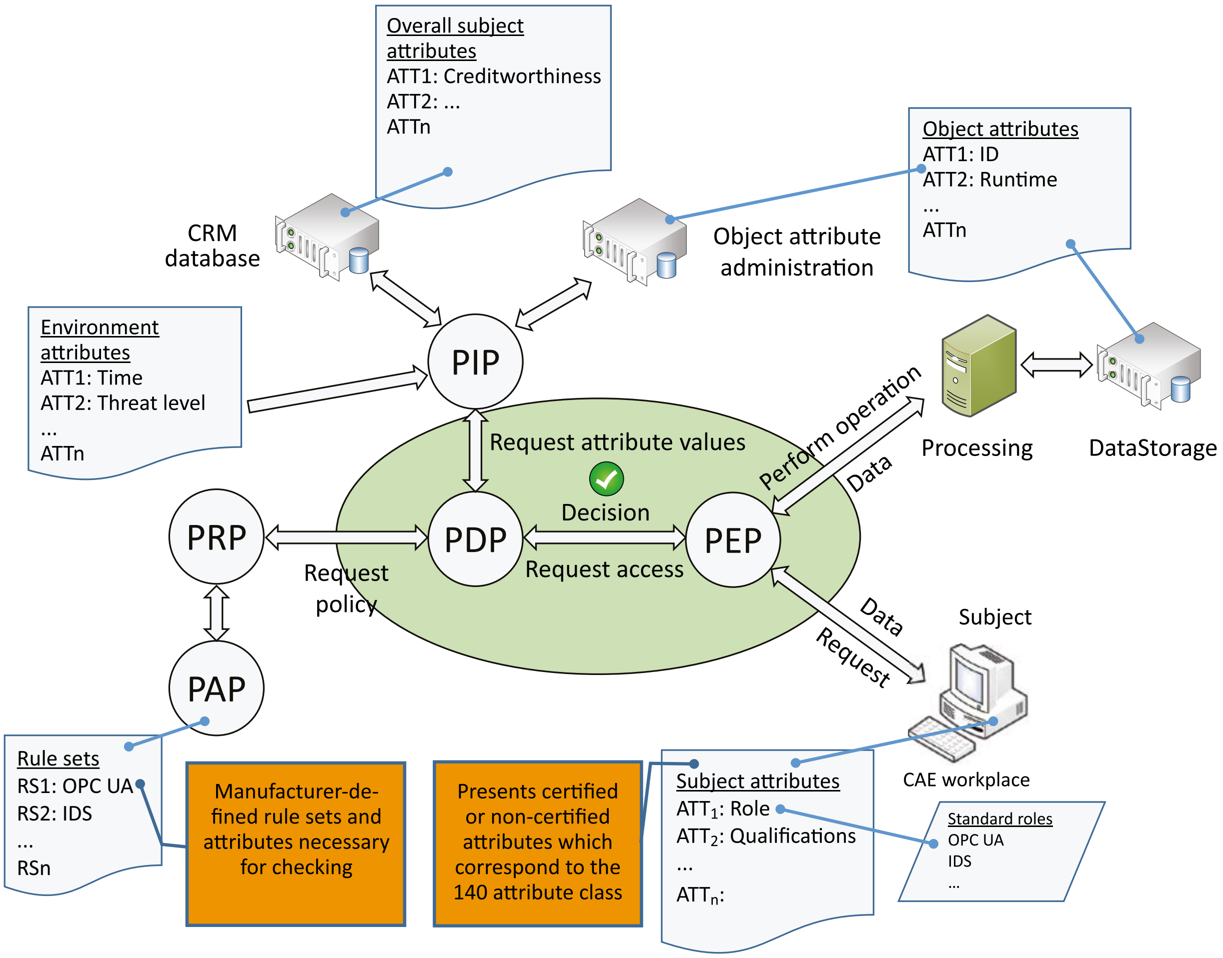
The attribute container may belong to a standard, such as OPC UA or IDS, or it may be proprietary. One of these attributes can be the role which the subject has in the subject's own organisation (e.g. 'service technician', 'maintenance engineer'; 'warehouse management system", "CNC milling centre 123" etc.). For human subjects, there may also be attributes such as their qualifications or experience.

Depending on the attribute sets supplied, it should be possible to select the access rules (policies) so that the interpretation of the different attribute containers is possible. Therefore, only the attributes which are present in the attribute container are checked.

The manufacturers of integrated components should be able to prescribe an attribute list which defines the minimum requirements to comply with the manufacturer's defined rules for access to its component. It must be noted that the rules of the integrator or operator can only add limitations to the access possibilities defined by the manufacturer.

In Figure 7 the process context in the ABAC concept is presented:

Figure 7: Process sequence in the ABAC concept by analogy with "Secure receipt of CAE data" [13]



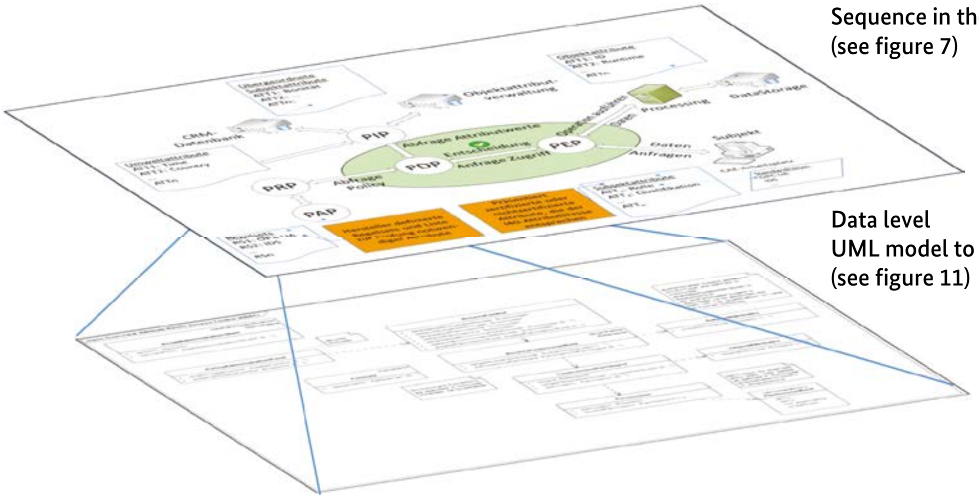
The authenticated subject (bottom right – CAE workplace in Figure 7) initiates the process with a request to the Policy Enforcement Point (PEP) for access to an object of the administration shell, here the CAE data of the manufacturer. The PEP interprets the subject attributes and passes the request on to the Policy Decision Point (PDP). The PDP demands the relevant access rule from the Policy Retrieval Point (PRP). The PRP obtains the policies from the Policy Administration Point (PAP), and on the basis of the properties of the subject it passes on the relevant rules for interpretation to the PDP. The rules or policies which can be implemented in an ABAC concept are limited to the extent defined by the computing language.

The PDP checks the rule provided by the PRP using the information from the Policy Information Point (PIP). Here, both the direct subject attributes and the extended subject attributes, such as the creditworthiness of the organisation to which the subject belongs, are included in the check. The PIP provides object attributes such as the status of the object and environment attributes such as the time to be used in the rule checks. The PDP notifies the PEP of the result of the rule check. The PEP carries out the decision in relation to the subject – by allowing or denying access.

3.2 Information model for ABAC in the administration shell

The processes of the administration shell can only use data if the administration shell knows of the existence and properties of such data. Therefore, the processes run on

Figure 8: Connection between the process sequence in Attribute-Based Access Control and the UML



Process level
Sequence in the ABAC process
(see figure 7)

Data level
UML model to describe the data types diagram
(see figure 11)

the basis of the data described in the meta-model of the administration shell. The meta-model has been specified as a UML diagram.

The meta-model of the administration shell also takes the requirements of ABAC into account from the outset (cf. Figure 8). For example, access rules can be described as part of the administration shell (cf. Figure 11).

The subject, object and environment attributes are described as sub-model elements. In the course of the further work, these elements must be implemented as data models.

A role is no longer defined in the conventional sense of access control by RBAC (cf. 1.3 and 2.2), it is merely one of several attributes of the requesting subject on the basis of which ABAC can check the access permissions. Therefore the role is assigned by the subject’s own organisation.

Access control by ABAC can be reduced to a static RBAC model if the only condition for access to an object is the possession of a certain role as a subject attribute.

The process for the interaction of PEP, PDP, PIP and PAP is not part of the UML modelling (cf. Figure 8). The process elements PEP, PDP, PIP, PAP etc. consist of code lines which access the defined data of the UML diagram. No UML modelling is needed for these processes because no specific data types are associated with them.

3.2.1 Mapping of security configuration features within the administration shell

The necessary security settings are mapped by using the data model of the administration shell. They are described in detail below.

3.2.1.1 General security settings/attributes

The general security settings/attributes include security settings for communication channels and settings for log data management and security monitoring. The modelling is carried out in the UML diagram via the class **Security** (cf. Figure 9).

Access to the data modelled in the class **Security** is implemented via the API of the entity **AssetAdministrationShell**.

The entity **Security** also contains a reference to the entity **AccessControlPolicyPoints**, which in turn includes the entities required to make and carry out decisions, i.e. the **PolicyAdministrationPoint**, **PolicyDecisionPoint**, **PolicyEnforcementPoint** and PolicyInformationPoint (cf. Figure 10).

Figure 9: Security settings/attributes of the administration shell

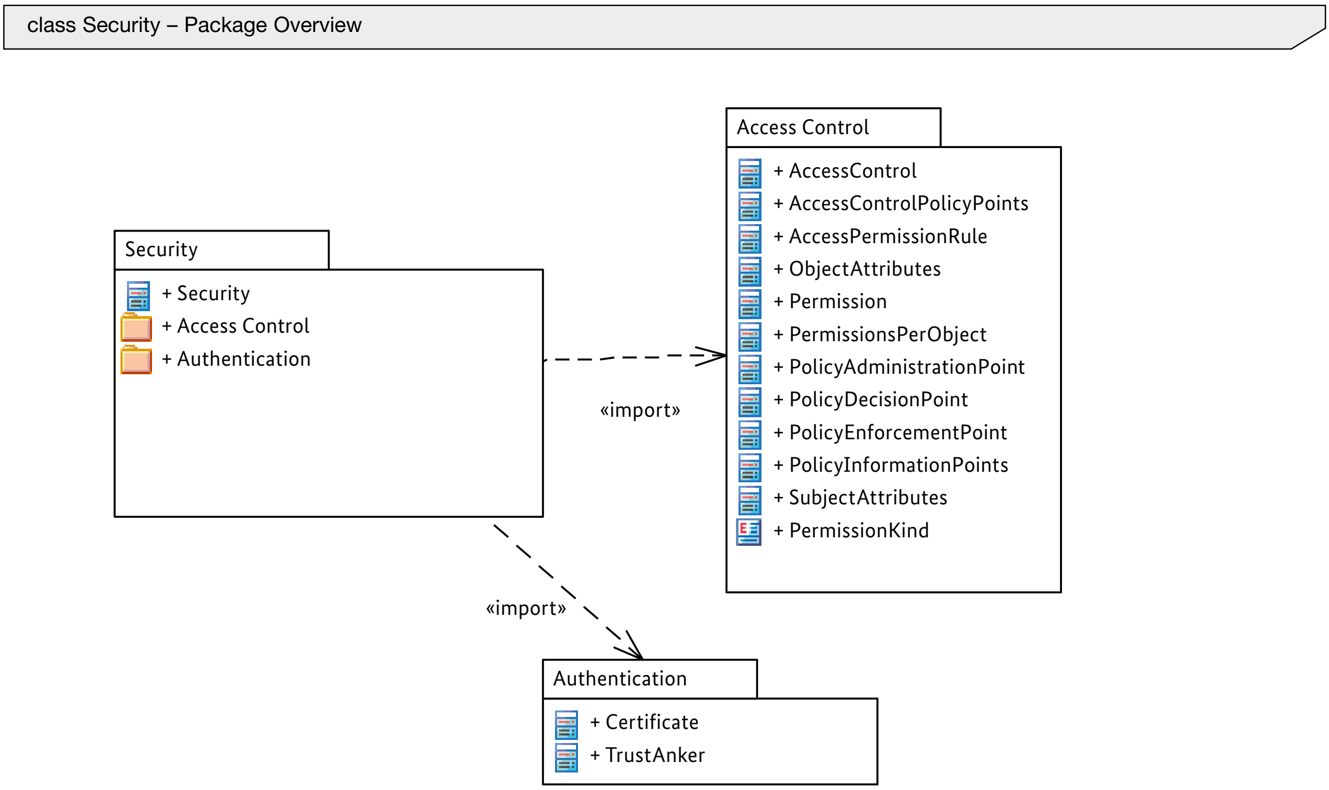
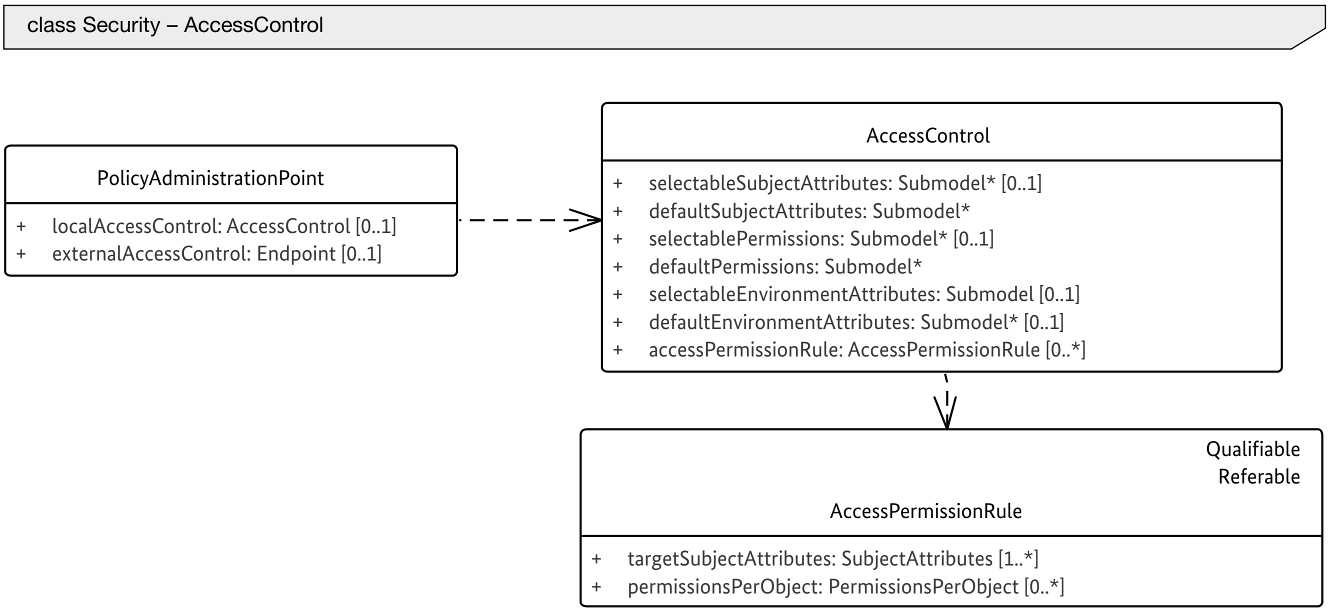


Figure 10: UML diagram for the class Security


class Security – AccessControl
PolicyAdministrationPoint
+ localAccessControl: AccessControl [0..1]
+ externalAccessControl: Endpoint [0..1]
AccessControl
+ selectableSubjectAttributes: Submodel* [0..1]
+ defaultSubjectAttributes: Submodel*
+ selectablePermissions: Submodel* [0..1]
+ defaultPermissions: Submodel*
+ selectableEnvironmentAttributes: Submodel [0..1]
+ defaultEnvironmentAttributes: Submodel* [0..1]
+ accessPermissionRule: AccessPermissionRule [0..*]
Qualifiable
Referable
AccessPermissionRule
+ targetSubjectAttributes: SubjectAttributes [1..*]
+ permissionsPerObject: PermissionsPerObject [0..*]

3.2.1.2 Settings and use of the ABAC access control mechanism

The ABAC-access control mechanism must be able to use the access rules, check the fulfilment of the access rules and allow or deny the access requested by the subject as appropriate.

It is optionally possible to configure and call up an access control mechanism either inside or outside the Industrie 4.0 entity. The selection of the relevant option will depend on factors such as the available computing power and memory size of the Industrie 4.0 entity.

3.2.1.3 Access rules and mapping of subject, object and environment attributes

In Figure 11 the relationships are presented in the context of the modelling of the access rules. The rules can either be stored internally, i.e. locally as part of the administration shell, or externally in the **PolicyAdministrationPoint**.

When there is an internal request, the **PolicyAdministrationPoint** refers to the sub-model AccessControl. The sub-model **AccessControl** contains among other things the list of access rules (**AccessPermissionRules**) for the ABAC access control mechanism.

The data used in the access rules (**AccessPermissionRule**) consists of 3 types which can each arise from different sources:

- **User-specific (or identity-specific) attribute**s
(subject attributes, Figure 7):
The subject attributes are additionally transmitted to the Industrie 4.0 entity with the identity which wants to access the Industrie 4.0 entity (object) and can be consulted in the rule-based validity check. A possible attribute which can also be consulted is the list of roles which are assigned to the identity. These attributes are then checked against the target values of the subject attributes. The target values are modelled in the sub-model **AccessPermissionRule** in the class **TargetSubjectAttributes** and refer to the class **SubjectAttributes**.

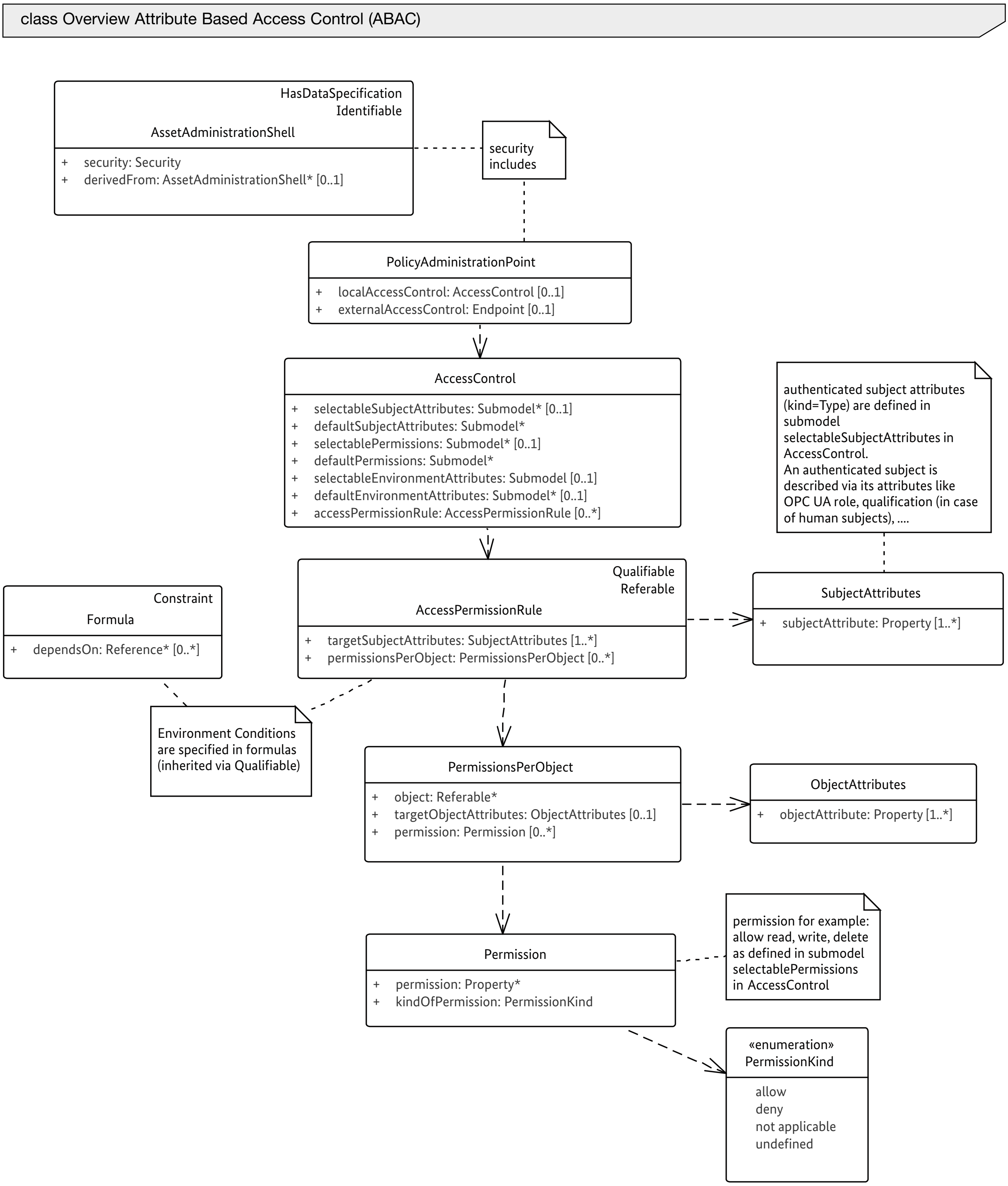
- **Industrie 4.0 entity attributes**
(object attributes, Figure 7):
The subject addresses the access request for a specific object to the access control mechanism and presents its subject attributes for checking. To check the access request by the subject, the request must refer to a referenceable element of the administration shell (**object** in **PermissionsPerObject**). The access rule may demand object attribute values (target values such as certain operating modes) in order to grant the subject access to the object. The target values are modelled in the sub-model **AccessPermissionRule** in the class **TargetObjectAttributes** and refer to the class **PermissionsPerObject**. The object attributes must be describable by references to elements of sub-models.

- **Environment condition attributes**
(Environment attributes, Figure 7):
At the time of the check, the environment attributes may need to be obtained externally in order to consult them when checking the access rules. The current environment variables can also be transferred to the Industrie 4.0 entity shortly beforehand and can then be queried internally (e.g. certain properties of the Industrie 4.0 entity which detect these attributes). Which environment condition attributes are available for the formulation of access rules is defined in **selectableEnvironmentAttributes** in the entity **AccessControl**. By means of **defaultEnvironmentAttributes**, properties can be stated which are provided as default by the administration shell or can be obtained from the administration shell. The **selectableEnvironmentAttributes** consist of a reference to a sub-model where the necessary environment information can be stored.

All attribute types (subject, object and environment attributes) can be used in the formulation of conditions (**formula** inherited via the abstract class **Qualifiable**). The condition must be fulfilled (true) so that the authenticated subject can be granted the desired access as defined in **PermissionsPerObject**.

The type of access involved is defined in **Permission**. The meta-model of the administration shell does not define which access types exist, therefore it refers to an appro-

Figure 11: UML data model of the rules for ABAC


class Overview Attribute Based Access Control (ABAC)
HasDataSpecification
Identifiable
AssetAdministrationShell
+ security: Security
+ derivedFrom: AssetAdministrationShell* [0..1]
security includes
PolicyAdministrationPoint
+ localAccessControl: AccessControl [0..1]
+ externalAccessControl: Endpoint [0..1]
AccessControl
+ selectableSubjectAttributes: Submodel* [0..1]
+ defaultSubjectAttributes: Submodel*
+ selectablePermissions: Submodel* [0..1]
+ defaultPermissions: Submodel*
+ selectableEnvironmentAttributes: Submodel [0..1]
+ defaultEnvironmentAttributes: Submodel* [0..1]
+ accessPermissionRule: AccessPermissionRule [0..*]
authenticated subject attributes (kind=Type) are defined in submodel selectableSubjectAttributes in AccessControl. An authenticated subject is described via its attributes like OPC UA role, qualification (in case of human subjects), ...
Constraint
Formula
+ dependsOn: Reference* [0..*]
Qualifiable
Referable
AccessPermissionRule
+ targetSubjectAttributes: SubjectAttributes [1..*]
+ permissionsPerObject: PermissionsPerObject [0..*]
SubjectAttributes
+ subjectAttribute: Property [1..*]
Environment Conditions are specified in formulas (inherited via Qualifiable)
PermissionsPerObject
+ object: Referable*
+ targetObjectAttributes: ObjectAttributes [0..1]
+ permission: Permission [0..*]
ObjectAttributes
+ objectAttribute: Property [1..*]
permission for example: allow read, write, delete as defined in submodel selectablePermissions in AccessControl
Permission
+ permission: Property*
+ kindOfPermission: PermissionKind
«enumeration»
PermissionKind
allow
deny
not applicable
undefined

priate sub-model element in the **selectablePermissions** sub-model, as defined in **Access Control**. Typical access types are “read”, “write” etc. In addition to the access types, the kind of access is also defined in the access rule (**kindOfPermission**): Here, a distinction is made between permitted access (**allow**), prohibited access (**deny**), undefined access (undefined) and non-applicable access (**not applicable**).

Whereas the access kinds ‘allow’ and ‘deny’ lead to a direct effect for the requesting subject, ‘undefined’ and ‘not applicable’ serve as internal error messages in an inconsistent set of rules. The effect of each message in relation to the requesting subject must be checked in individual cases.

3.2.1.4 Mapping of the subject attribute ‘Role’:

With ABAC, the role loses its conventional importance and is now just one of several subject attributes which are passed to the access control mechanism as a basis for the decision on access permission. The repository of target values which are checked against the submitted subject attributes can be found in the entity **TargetSubjectAttributes**.

3.2.1.5 Default permissions and default roles

The Industrie 4.0 entity must permit an initial log-in by the commissioner/operator/machine manufacturer using default permissions to configure the security settings.

At the same time, the compatibility of the access control system with existing access control concepts such as RBAC (cf. 2.5) and commonly used protocols such as OPC UA (cf. 3.3) must be guaranteed.

To accommodate such attributes, the attributes **defaultSubjectAttributes** and **defaultPermissions** from the entity **AccessControl** refer to the entity **Submodel** in which the relevant attributes can be stored.

3.2.1.6 Possibilities to combine interconnected objects of an Industrie 4.0 entity leading to an award of permissions

As described above, the role is not used as an integrating object for individual permissions. To prevent the decision-making rules for the ABAC access control mechanism from increasing dramatically (e.g. a decision on each individual property of the Industrie 4.0 entity), it is nevertheless necessary to have an entity to combine interrelated properties of an Industrie 4.0 entity (e.g. the list of interrelated properties of an engine inside the Industrie 4.0 entity, such as speed, maximum power consumption, oil pressure, temperature, etc.).

The combined summary is possible within the entity **AccessPermissionRule** by including a reference to a list of entities of the type **PermissionsPerObject** in the attribute **PermissionsPerObject** and deciding on the possible access to these entities by a subject in the course of the implementation of a rule in runtime.

3.2.2 Permission check by calling on the ABAC access control mechanism

Possible software implementations to perform a check on an access request will now be presented.

Note: The software implementations presented here are merely used as an illustration and are in no way binding.

The access control mechanism can be implemented either within the administration shell or outside it (cf. 3.2.1.2).

When a subject requests access to an Industrie 4.0 entity, the subject attributes are passed on to the access control mechanism and the check is then started.

If the access control mechanism is implemented within the administration shell, the entity ‘PolicyEnforcementPoint’ generates an entity of the class ‘PolicyDecisionPoint’ (in accordance with the PDP in Figure 7). The ‘PolicyEnforcementPoint’ passes the subject attributes on to the ‘PolicyDecisionPoint’. The ‘PolicyDecisionPoint’ then requests the

attributes contained in the entity AccessControl and generates individual entities of a class 'Rule' with the aid of the attributes in the entity AccessPermissionRule. The 'PolicyDecisionPoint' passes the subject attributes on to the 'Rule'. In addition, the 'PolicyDecisionPoint' compiles an entity 'PolicyInformationPoint' which combines all necessary object attributes and environment attributes in its implementation (cf. 3.2.1.3).

Then the software implementation of the class 'Rule' can request the data for the environment variables and object attributes from the entity 'PolicyInformationPoint' and then apply the actual rule, if appropriate with the aid of the optional entity **Formula**.

The implementation of the rules in the entity 'Rule' consists of several parts:

- Checking the subject attributes provided and their values against the attributes contained in **SubjectAttributes** and their target values for a successful check
- Use of an optional formula which carries out the check with the aid of the values of the object attributes and environment attributes.

The result of the check in 'Rule' is then passed on to the 'PolicyDecisionPoint' in the software implementation.

If the access rules have been successfully checked, the requested permissions can be granted. The mapping of permissions is implemented in the entity PermissionsPerObject which contains the objects and the type of access. The objects can be freely selected. For example, they could be sub-models or individual properties of the Industrie 4.0 entity in which the check is carried out.

Finally, the entity Permission contains the detailed permissions for the allocated element, such as read, write, execute in combination with allow, deny, undefined etc..

3.2.3 Example application of the UML model to different 'large' Industrie 4.0 entities and scenarios

Note: This chapter contains an interim summary of the first thoughts in the discussion, which will then be continued. Participation in the discussion is open to all interested parties[1].

To ensure that Industrie 4.0 access control can be deployed throughout, it must be possible to apply it to all Industrie 4.0 entities irrespective of their 'Size'. The 'Size' of an Industrie 4.0 entity is determined by its computing power and storage capacity. Against this background, two possible configurations of the administration shell will now be presented as examples. The first example is a temperature sensor, which represents the minimal implementation, and the other is a fully implemented administration shell.

3.2.3.1 Example application for a 'small' Industrie 4.0 entity

A small Industrie 4.0 entity has low computing power and small memory capacity, therefore it has an administration shell with a modest implementation depth. Only the most necessary functions are available, and individual software components are implemented externally and called up by the administration shell when needed.

Example: A thermometer provides a temperature which can be requested via the network for an Industrie 4.0/IOT scenario. An IP address can be dynamically obtained.

In this example, the ABAC check is reduced to an RBAC-based check because environment variables and object attributes are not consulted in the check.

- The communication is implemented via an HTTP-based interface. HTTPS (and the corresponding certificate administration) is not possible.
- Identities cannot be created/updated, and other roles in addition to the default roles cannot be assigned.
- Subject attributes added to the subject name cannot be evaluated.
- Sub-models cannot be added, changed or deleted, and the same applies to object attributes.
- Rules and formulae to check permissions cannot be added.
- The temperature range and the accuracy of the sensor can be set at the time of commissioning.

The UML modelling for this example is shown in Figure 12.

Solution: To start operating the device, the manufacturer can define defaultSubjectAttributes which correspond to the role of an administrator (to adjust the temperature range and accuracy) and a value reader (for requests for the temperature).

For the initial set-up, the software implementation of the device can react to a pre-installed fixed default identity (cf. 3.2.1.4). The default identities only need to be known within the software implementation. In access with HTTP, the log-in would be via this pre-installed fixed identity and (to ensure a certain level of basic security) via one different fixed password/PIN for each individual Industrie 4.0 entity (e.g.: User name: SENSORADMIN; Password: EQCXd8h0NCxGsoHChktAh0wBqSY298WA). After logging in, this identity would be entitled, for example, to change the temperature range and the resolution.

In addition, a second identity is needed which is only entitled to reading access to the device in order to supply the temperature to the overall system.

Here, again, a user name and a different fixed password are needed for each temperature sensor (e.g.: User name: SENSORREADER; Password: HChkxAh0wBqSY298WAFVeYA7IeEOSPBg).

The internal software implementation can then directly utilise the user name to assign the subject attribute a role with the values SENSORADMIN or SENSORREADER and provide this subject attribute to the very simple access control mechanism (Figure 13). The ABAC access control system is used because both the user name and the operating mode (initial configuration/operation) are distinguished as criteria to grant the relevant access.

[1] To participate, please contact the office of the Platform Industrie 4 (https://www.plattform-i40.de/)

In the case of this Industrie 4.0 entity it is not possible to encrypt the data streams, so for security reasons it advisable to adopt additional protective measures such as a gateway which shields insecure entities from exterior contact (i.e. via the Internet or other internal company networks).

Figure 12: Access control for a 'small' Industrie 4.0 entity

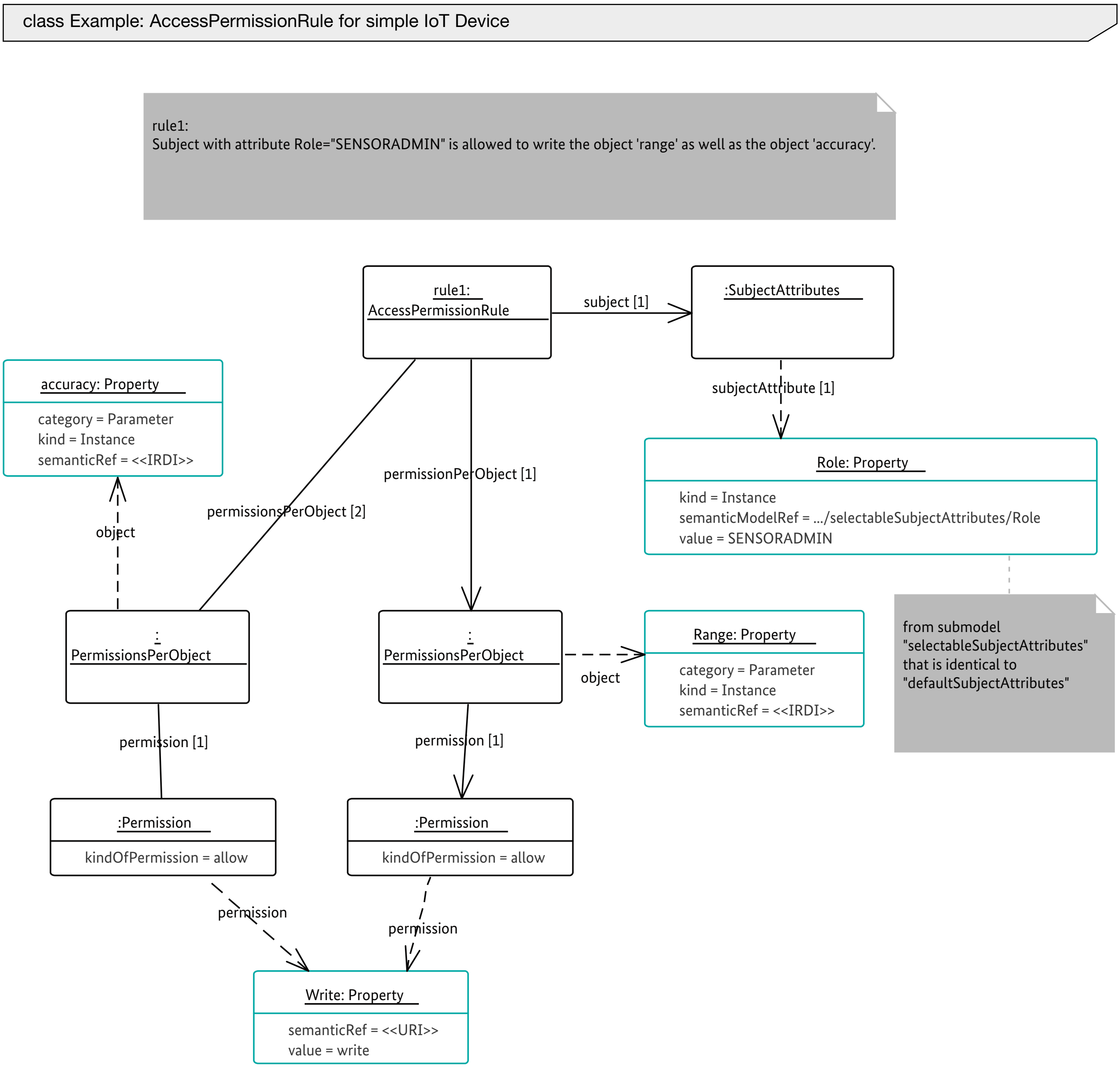
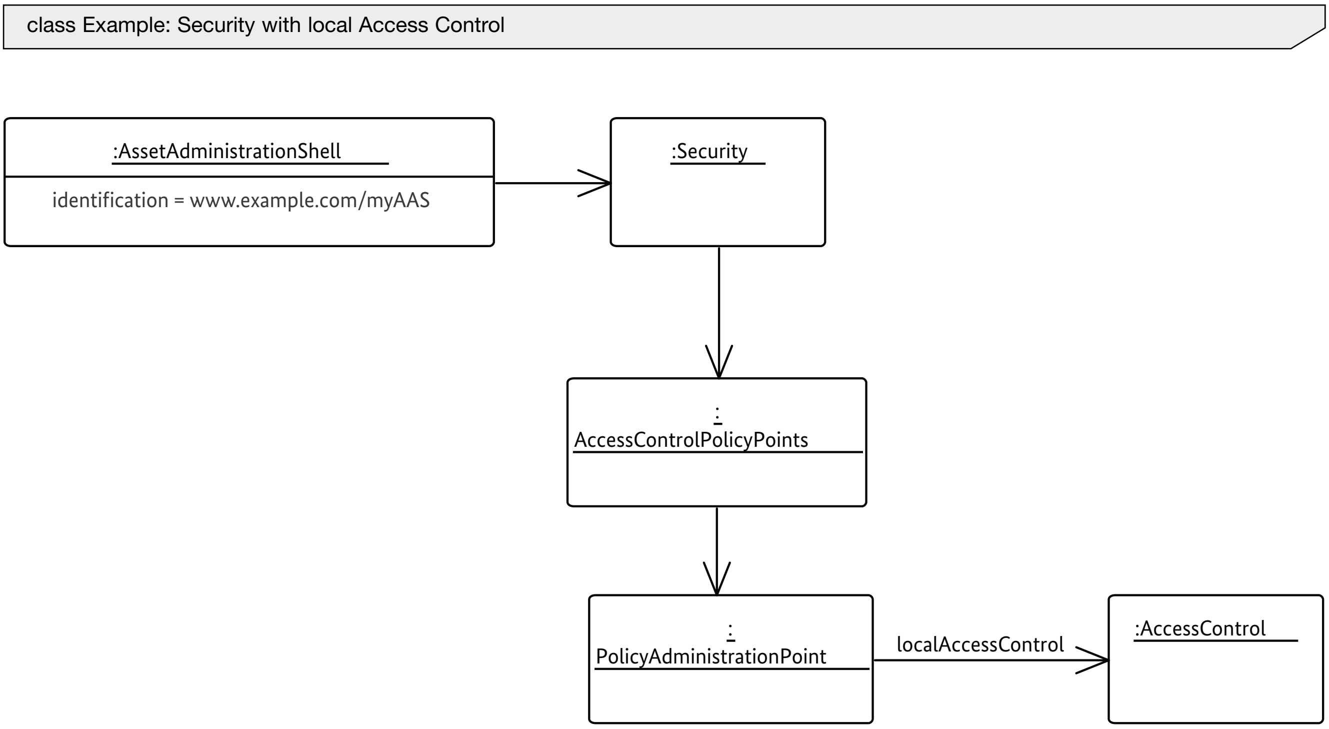


Figure 13: Security in of the administration shell for a ‘small’ Industrie 4.0 entity (with PolicyDecisionPoint and PolicyEnforcementPoint)


class Example: Security with local Access Control
:AssetAdministrationShell
identification = www.example.com/myAAS
:Security
:
AccessControlPolicyPoints
:
PolicyAdministrationPoint
localAccessControl
:AccessControl

The following entities are needed for commissioning and for operation:

- Commissioning: For the user SENSORADMIN and the derived subject attribute of the role with the value SENSORADMIN, the software implementation can run the rule which is stored. As shown in Figure 13, the check is carried out by merely comparing the role name with the target value of the role name in the entity **SubjectAttribute**s. An additional formula to check other conditions is not necessary. The entities **PermissionsPerObject** which are assigned to the entity **AccessPermissionRule** contain access to the **Properties** ‘TemperatureRange’ and ‘Resolution’, which are each stored with **read: allow; write: allow**.

- Operation: For the user SENSORREADER, in the course of the log-in process the software implementation assigns a role with the value SENSORREADER to the subject attribute which is derived from the user name. The check by the rule is similar to the example for commissioning, i.e. the role name is compared with the target value of the role name in the entity **SubjectAttributes**. The access control mechanism then creates access to the **Property** ‘Temperature’ which is stored with **read: allow; write: deny**.

In other words, the following important entities arise:

- **AssetAdministrationPoint** contains the access to **AccessControl**.

- **AccessControl** contains the attributes defaultSubjectAttributes (here referring to the sub-models for the two roles SENSORADMIN and SENSORREADER), defaultPermissions (here: access to settings properties), accessPermissionRuleSet (here consisting of a rule in **AccessPermissionRule**, which checks access by means of the property role contained in **SubjectAttributes** and its target value and checks access to the above properties by a comparison with the role name which is submitted as a subject attribute). When requested, the attributes selectableSubjects and selectablePermissions contain only the corresponding values from the attributes defaultSubjectAttributes and defaultPermissions, because other subjects and permissions cannot be updated in this Industrie 4.0 entity. The attribute environmentAttributes is not referenced in this case.

- Depending on the user name and the resulting role at the time of log-in, **PermissionsPerObject** refers to the properties 'TemperatureRange' and 'Resolution', or to 'Temperature' and the corresponding entity **Permission**.

- **Permission** contains the allocation to the permissions allow/deny for the two properties.

- **Formul**a is not present because no check is carried out on the basis of environment variables or object attributes.

Everything can be 'hard-coded' because the objects of the administration shell for this entity do not have any external contact, they only have the two HTTP-based interfaces and basic protocol functions for the HTTP protocol. There is no ongoing administration overhead for this Industrie 4.0 entity, the costs of production to provide the Industrie 4.0 entity remain low.

3.2.3.2 Example of the initial commissioning of a 'large' Industrie 4.0 entity

In a 'large' Industrie 4.0 entity there is sufficient computing power and memory capacity so that a completely implemented administration shell can be used.

In the example, it is assumed that a 'large' entity has not yet been set up, for example that no access rules have been stored (apart from one rule for initial commissioning).

The example also assumes that there are two parts of the initial commissioning:

1. Security commissioning: Secure network commissioning, definition of endpoints and creation of an initial identity, and provision or import of access rules.
2. Technical commissioning: Configuration of Industrie 4.0 entity properties and transfer of the entity into active production status.

The example deals with the first step of the initial commissioning.

In part 1, the initial (security) commissioning person must also be able to work on the entity without updated users, rules and roles, and with an initial (security) commissioning log-in this person must be able to perform important actions, such as setting up the network connections. Then, the person carrying out the initial commissioning must be able to generate an additional local user identity to test permissions with rules. The Industrie 4.0 entity is in the mode 'SecurityCommissioning' which is described by a corresponding property in the Industrie 4.0 entity.

Note: The administration of (initial commissioning) identities is currently not mapped in the administration shell. This means that the initial commissioning authentication is represented by means of the software implementation. This applies to the first two points in the following list of sub-processes.

Examples of what the process of initial commissioning can mean:

- The operating instructions for the entity explain how the initial log-in works. It could be without a user identity but with a log-in by means of a pre-set fixed and unique identity defined by the manufacturer, E.g.: "Enter the ID HChkxAh0wBqSY298WAFV in the log-in window". Then comes a dialogue which leads to the creation of a user with an assigned user identity and password who is then, for example, assigned the default role 'SECURITY_ADMIN'. The rules for checking requests for this role are also contained in the entity. After the new user identity has been created, the log-in via the fixed pre-set ID is automatically deactivated. The initial log-in function can then only be reactivated by a complete reset of the Industrie 4.0 entity.

- The initial security commissioning person now logs in for future activities with the newly assigned user identity and password. In this process, the person is assigned the role description SECURITY_ADMIN, and the rule check gives a successful check. The rule check here is carried out by comparing the submitted subject attributes with the target values for the role name in **AuthenticatedObject**.

- The security administrator now implements various network configurations, e.g.:
  - Installation of certificates to establish secure network connections
  - Assignment of an URL and/or a fixed IP address
  - Creation of endpoints, e.g. an external **AccessControl** endpoint, or an endpoint to obtain environment variables (e.g. time, outdoor temperatures)
  - Creation of endpoints for the connection of an identity management system to enable interaction with the identity management. The possibility of authentication via a central identity administration means that other users with other permissions can work on the entity (e.g. the next person could be a user in a technical commissioning role)
  - If appropriate, other entity properties can be created, including **PermissionPerObject** and its **Permission**
  - After the completion of these initial activities, the initial security commissioning person sets the mode of the Industrie 4.0 entity from 'SecurityCommissioning' to 'TechnicalCommissioning' or another interim mode. From this moment on, the standard authentication is carried out with the appropriate external subject administrations, and the Industrie 4.0 entity can be accessed from the production network.

3.2.3.3 Example: Checking the access rights for time-limited permission

The example of the service technician which was already described in Chapter 2.3 is used again here. For clarity, the wording of the scenario is repeated here.

A service technician from company A needs to carry out remote maintenance on an Industrie 4.0 entity in company B. Access should only be possible if the Industrie 4.0 entity is in 'MaintenanceMode' and it takes place in the time between 15:00 and 17:00 hrs. In addition, access should only be permitted if the service technician has previously completed the self-assessment security check which is required for external access.

Because several Industrie 4.0 entity properties are provided by the manufacturer or the commissioning person and the access rule for the validity check was therefore generated internally, the range of functions of the administration shell for the Industrie 4.0 entity must be greater than in the previous example (cf. 3.2.3.1). In this case, the implementation is not 'hard-coded', it is extendible or changeable at various points.

A 'large' Industrie 4.0 entity must therefore provide sufficient computing power and memory capacity so that a completely implemented administration shell can be used. The Industrie 4.0 entity which is considered here therefore permits the following actions:

- Configuration of secure communication
- Creation/administration of additional roles
- Generation of sub-models
- The option to set up new rules, and
- If appropriate, a connection to an external ABAC access control mechanism.

The differences compared with the previously described 'small' Industrie 4.0 entity are explained below.

- The **EnvironmentAttributes** of the entity **AccessControl** contain the reference to a **SubModel** which contains the property actualTime. To provide the data for this property from an external and trustworthy timer, Industrie 4.0 entity can access a configured endpoint from which it can request the information. This data could be obtained, for example, from an endpoint which is known in the class entity 'PolicyInformationPoint'.

Note 1:
Especially in time requests, the relevant time zones must be taken into account. In other words, the Industrie 4.0 entity needs a property 'TimeZone' to convert a UTC time obtained from the trustworthy external timer into local time, and then to carry out the corresponding check against the local time (e.g. check that it is 15:00 hrs. local time).

Note 2:
A general model for arbitrary endpoints within the Industrie 4.0 entity to obtain a variety of information is currently not included in the administration shell model.

- The attribute **targetObjectAttributes** in the entity **PermissionPerObjec**t contains the reference to a **SubModel** where the target values for the object attributes are stored. For the check of this part of the rule to return the value 'true', the attribute 'Mode' must be set to 'Maintenance'.

- **SubjectAttributes** contain the target values for the subject: **SubModelElement** 'Role' = 'RemoteMaintenance' and the **SubModelElement** 'securityCheck' = 'true', against which the values submitted by the subject attributes must be checked. If they are in agreement, this part of the rule gives the value 'true'.

- **Formula:** Contains a calculation formula which returns the value 'true' for the validity check if the time is between 15:00 and 17:00 hrs. and the attribute 'Mode' has the value 'Maintenance'. Otherwise the check returns the value 'false'. To access the time, the formula refers to the **SubModel** where the property actualTime is stored. To access the current value of the attribute 'Mode', it uses the attribute objectAttributes to refer to the corresponding **SubModel**.

- **PermissionsPerObject** contains the information about which object is involved, and it specifies what access rights (permission) for the object are assigned depending on the subject attributes.

In addition, the object attributes are defined. They correspond to a property which must be defined in one of the sub-models of the administration shell. These object attributes can be used in formulae in the access rule.

3.2.4 Connections between the existing ERP/ERM systems and access control

For historical reasons (see above), standard ERP software generally contains RBAC logic. In the past this was successfully used as a method to provide permissions. Newer ERP software is also equipped with ABAC, which is represented either by simple calculation rules in the standard permission checking process as an addition to RBAC or via an extra external/internal calculation logic which is able to take all attributes into account.

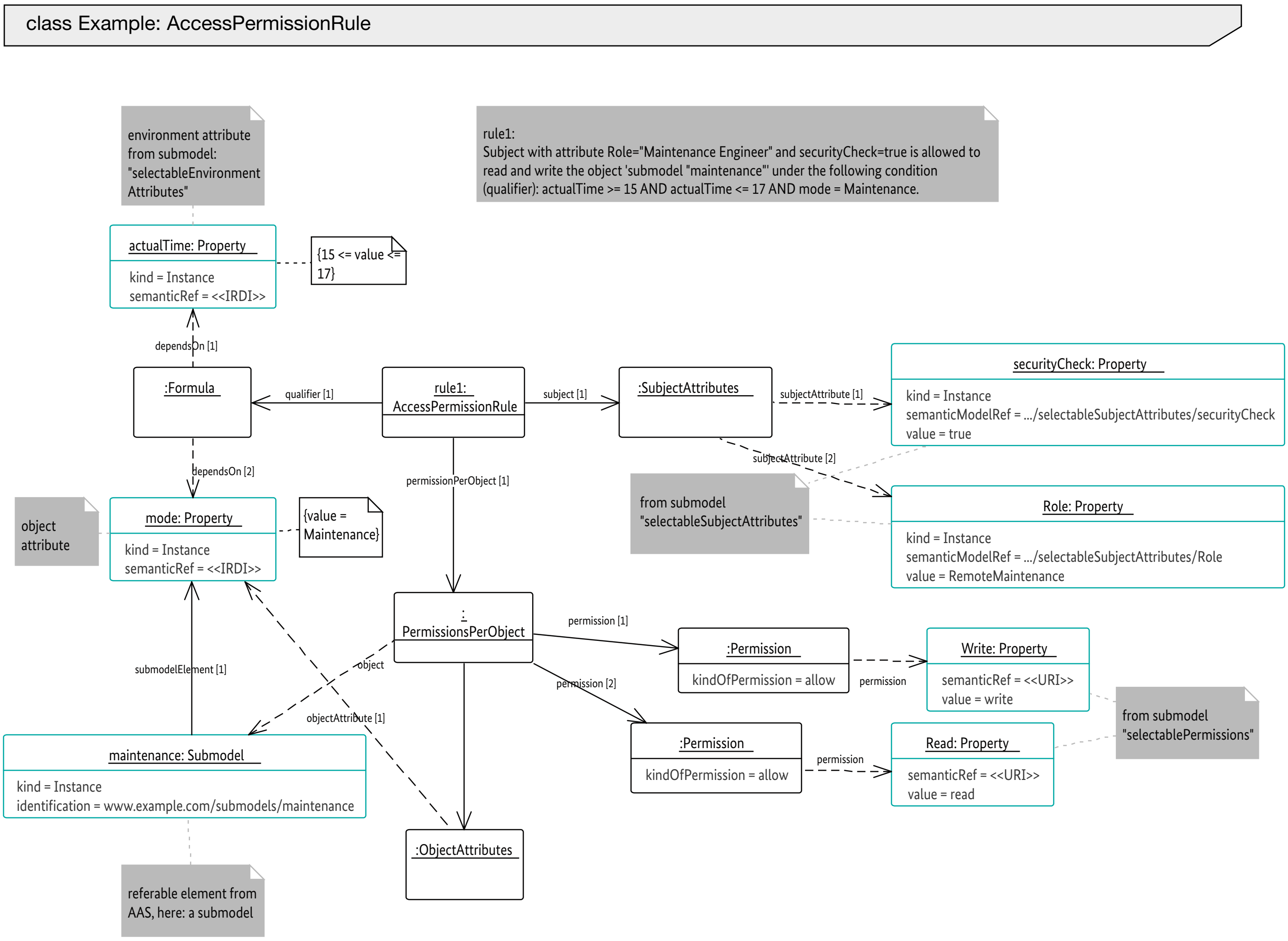
Example of a simple ABAC check in an ERP system: Under RBAC, a sales manager has permission to view order data. By means of a simple attribute-based rule, however, his permission is limited to the right to view the data of his own sales area. Here, in other words, a user-based attribute (subject attribute) is compared with attributes of the resource (object attributes).

Example: Connecting an Industrie 4.0 entity to an existing ERP system (e.g. for production planning purposes or for feedback from machines to the ERP system). It is possible for a production order to be (automatically) started/finished, irrespective of the relevant roles/rights system in the ERP system or the administration shell of the relevant Industrie 4.0 entity, because the relevant units (Industrie 4.0 component and ERP system) administer the rights of the identities on their own, and the communication interface used for this purpose is adapted for this purpose.

3.3 Usability of access management in OPC UA

The industry standard "OPC Unified Architecture" issued by the OPC Foundation has existed since 2006 and is constantly updated. On average, a new version is issued every three years. The most recent version is 1.04 which was published in 2018. Now, the standard consists of 14 parts and is supported by a number of "companion standards" which define industry-specific information models.

Figure 14: Endpoints for a time-limited permission



To summarise it roughly, the standard covers the following areas:

- Communication protocols
  - Between clients and servers, and
  - Between publishers and subscribers
  - Together with various transport possibilities
- A mapping and addressing model (address space, address range) for information and basic access to the information (services),
- Description possibilities for various information models (name spaces and types) which can coexist in the servers.

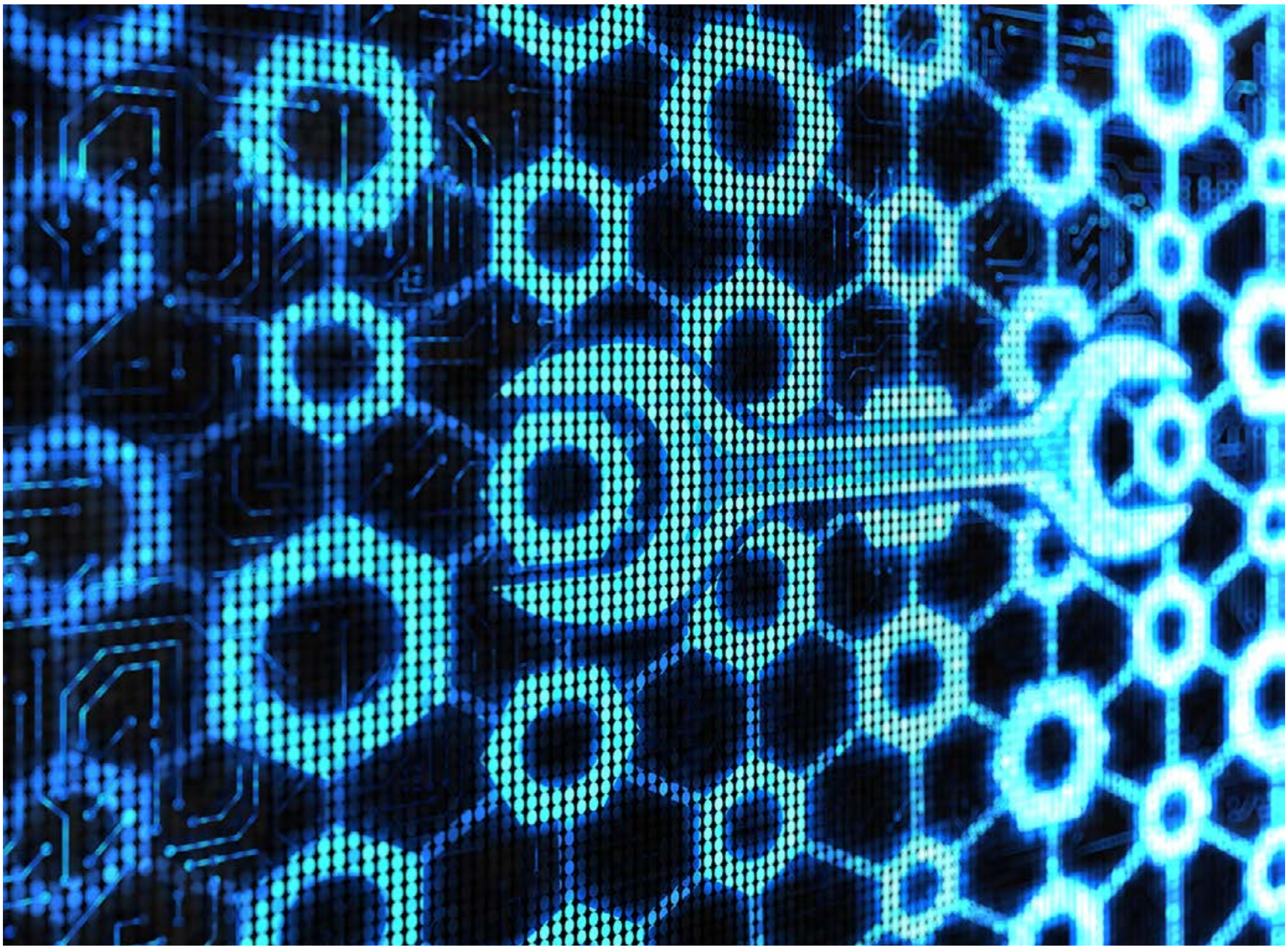
In the development of the standards, the principle of “security by design” was taken into account from the start. Therefore, from the outset they supported authentication and authorisation in access to information using OPC UA or in the address range of OPC UA servers.

Now there are many hardware and software products which are suitable for industry and which “speak” OPC UA. For example, hardware products with OPC UA servers include industry control systems which use OPC UA to make process information accessible. OPC UA clients are often found in panels which are used to visualise and operate processes. Master control units and even ERP systems have OPC UA interfaces to facilitate interaction with industrial processes.

The authentication and authorisation of access was covered in the standard from the outset. Even today, the OPC UA standard does not dictate whether to use RBAC or ABAC. From the very first version (1.00 in 2006), OPC UA servers had an address range which was able to show each client and user which access rights they have to individual objects in connection with their current session. Servers have the option to enforce individual access rights for different subjects to different objects.

The relatively recent version 1.04 contains a new description of how the servers can show which rules the access rights are based on which they enforce in relation to clients and users. The presentation of the origin of the access rights goes even further, because the standard itself describes methods to administer them with OPC UA. Because the presentation of the rules by means of OPC UA objects and their administration using standard access processes and OPC UA methods are defined, rights for the administration itself can also be explicitly described and differentiated. Enforcing access rights is still optional. Therefore it is also optional whether a server presents the access rights by means of OPC UA, or even makes administration possible. Enforcement, presentation and administration can still take place by other methods instead of the ways described by the OPC UA standard. The application or device manufacturer is allowed to decide which capabilities it provides with its product. A manufacturer can refer to part 7 of the standard to specify which profiles of the standard its product will support.

If we read parts 3 and 5 of the OPC UA standards in the version 1.04, which deal with access rights and their representation, our first impression could be that OPC UA exclusively supports role-based access control (RBAC). This is because the rules for access rights are represented on the basis of roles: for each object it is defined which roles grant which rights for the object. It is only on closer examination



that we recognise that the identity mapping rules defined in the standard (cf. “IdentityMappingRuleType”) can also be used to “connect” an attribute-based access control system to an OPC UA server. This is because the identity mapping rules define which roles are allocated to a current session (client session) according to which criteria. At the start of a client session, the identity mapping rules are used to define what set of rules should be used for the session.

The “roles” which are mapped in the information model of the OPC UA server should therefore always be regarded as “internal” roles. Externally defined roles and groups can be mapped to “internal” roles by means of the identity mapping rules. This is a good solution for manufacturers of products which are used in different contexts and have no standard which requires a quantity or set of roles, or for which several standards force different roles to be used. The manufacturers can define “internal” roles in their product in such a way that they can be most easily understood and applied in the use of the product. The user can then configure the mapping rules from to fit the real environment – and the decision on whether they are based on attributes or roles can be adapted to the individual needs of the business operation.

The rules permit the definition of complex criteria which cannot be fully represented in this document. They also include the rules for mapping attributes from access tokens such as the role or group. In addition, rules are possible which assign roles on the basis of specific user names or specific CA certificates or user certificates. It is even possible to create rules which assign roles on the basis of the use of certain applications (irrespective of their user) or on the basis of communication via specific endpoints of the server. The endpoints of a server can, for example, be connected to different network interfaces, so different rights can apply depending on “where” the access comes from.
If several rules are fulfilled for a session, the roles assigned by these rules are then treated as cumulative. Because the definition of presentation forms for access rules and their administration via OPC UA in version 1.04 of the standard is still relatively new, there are not yet many products which have this capability. Therefore the above considerations are largely theoretical. Now is a good time to transpose the requirements proposed in this document into real implementations of products with an OPC UA interface. The extent to which the identity mapping rules of OPC UA are actually suitable for connection with ABAC GMP will then need to be shown in practice.

Figure 15: Model of the allocation of identities to roles in OPC UA

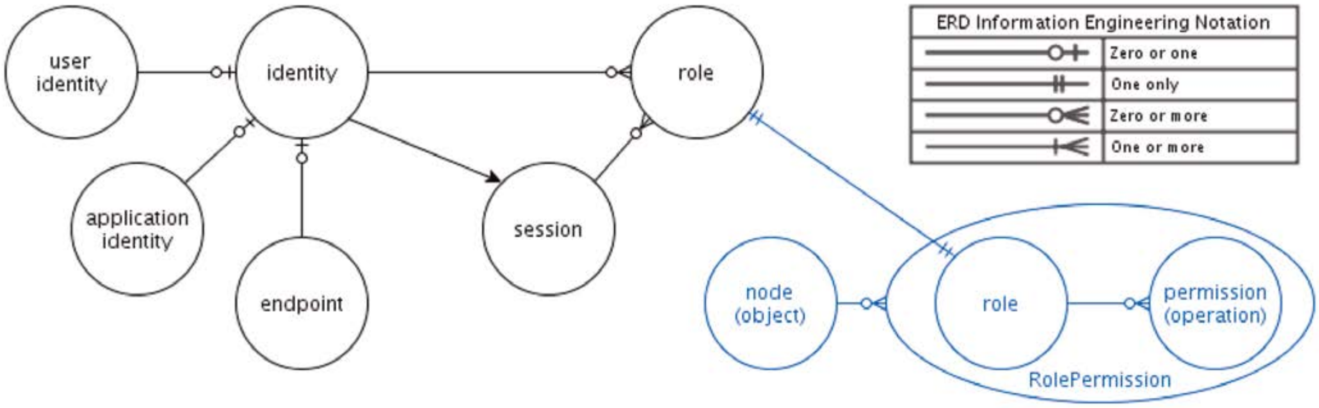
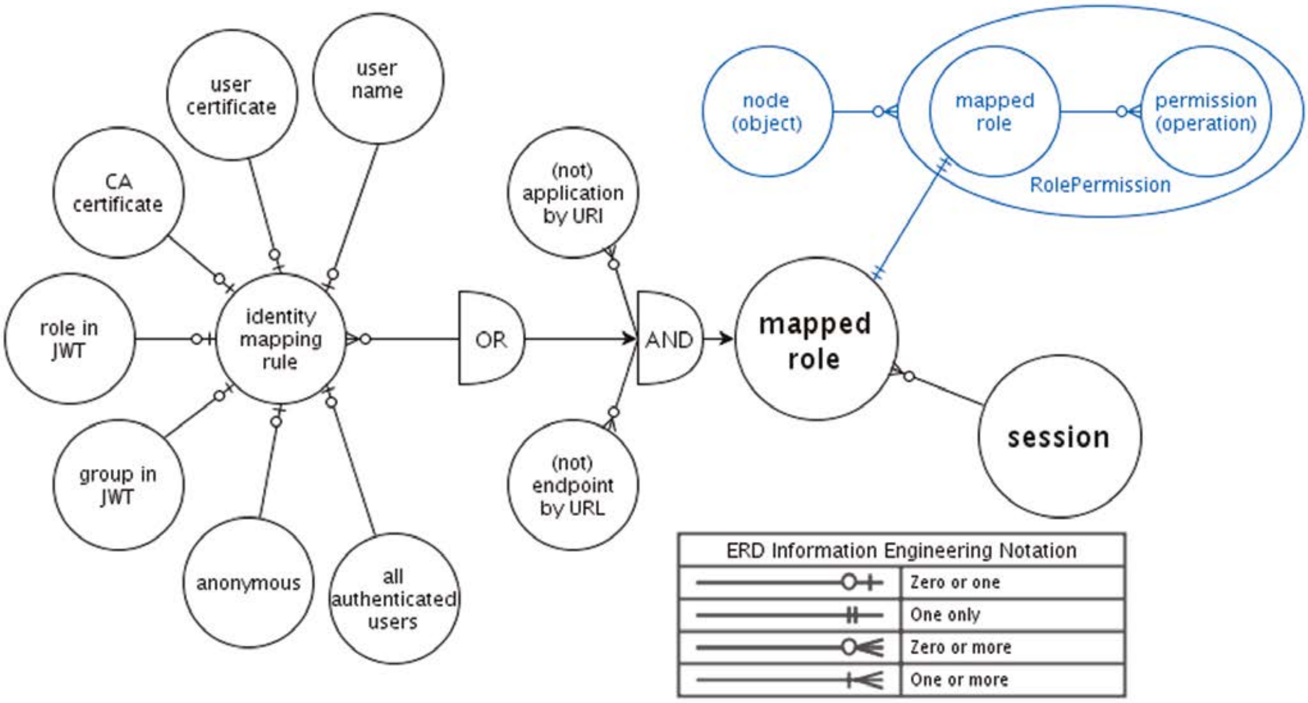


Figure 16: Identity mapping rules in OPC UA



3.4 The international data space as an application example for access control

The International Data Space (IDS) serves to create a shared data range for heterogeneous data from different application domains (e.g. industry, medicine, finance). The first applications have mainly arisen from industrial scenarios, such as the transmission of logistics data.

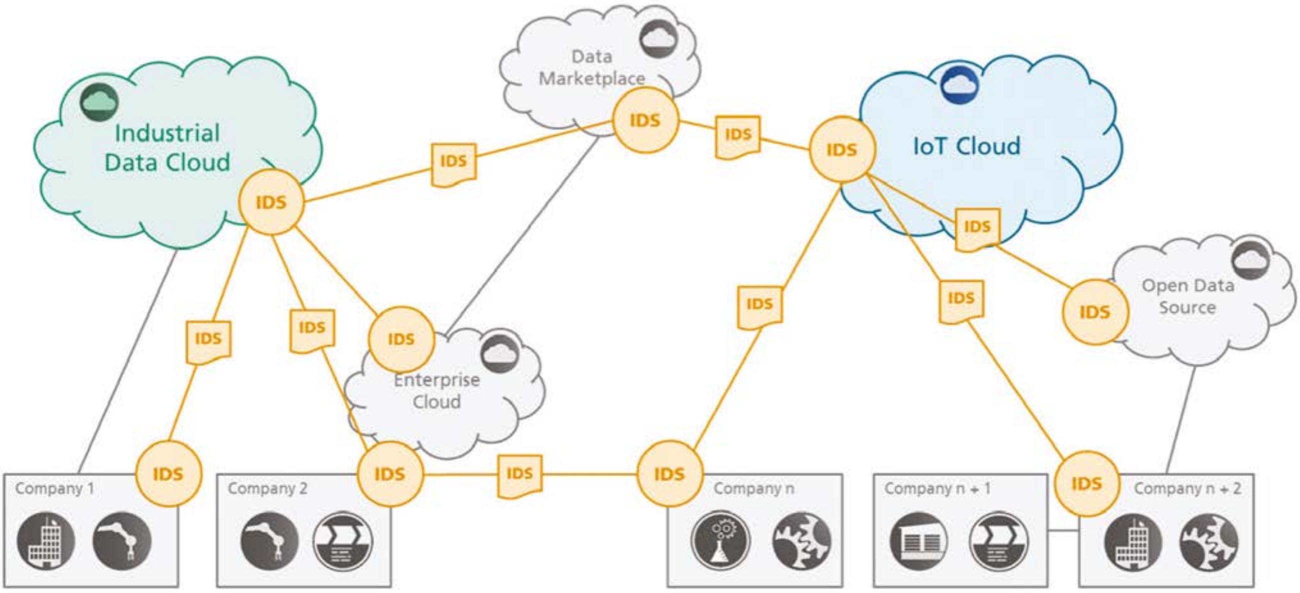
The IDS sees itself as a connecting link between existing infrastructures. For example, it offers transformation of data by means of IDS apps, description and retrievability of data sources, securing of communication, connection of data with conditions of use, definition of access control guidelines for data sources and enforcement of these conditions of use.

The central infrastructure component is the IDS connector, which can be regarded as an edge gateway. It defines the possibilities to access data outside the walls of the individual company. As a result, access control is of central importance. The relationships between data consumers and data providers are flexible and go beyond conventional bilateral relationships, so a flexible model is needed for this access control.

3.4.1 Attributes & identity in IDS

As an example, some of the attributes of organisations and IDS connectors will be mentioned here. An organisation must obtain certification in order to participate in IDS. The result of this process is the assignment to a certification level. The organisation must also be registered as a member company for IDS. The organisation operates one or more IDS connectors.

Figure 17: IDS architecture overview



The IDS connectors are also assigned attributes. For each data source, rules can now be defined to determine which attributes are necessary for access. For example, access to a data source with sensitive data can be restricted to connectors which have documented and certified audit logging components. This can be extended as required. For example, it can be specified that the software stack has been verified by remote attestation or that there is at least a specific framework for controlling the usage of data (see below).

Figure 19 maps the interaction between the connectors and infrastructure components which issue attributes and the award of access to data sources. Each connector has a unique identity which is verified by an X.509 certificate. But this identity is deliberately kept slim, so it only contains a Universally Unique Identifier (UUID). The actual attributes, which are usually transient, are stored in the Dynamic Attribute Provisioning Service (DAPS). After presenting the X.509 certificate (or proof of possession of the private key which belongs to the public key attested in the certificate), the requesting connector is granted a Dynamic Attribute Token (DAT) in which the relevant attributes are attested. This token can then be submitted in access operations to data sources. If the attributes match the stored access rules, access is granted.

This can be further differentiated, so that specific authorisation servers are called on for certain workflows.

3.4.2 Access control and usage control

The access control to data sources which are offered by an IDS connector is carried out on the basis of the attested attributes and determines which external party is entitled to access the relevant data endpoints. The “usage control” of data is an extension of the access control system. It refers to an extension of the control function beyond the time of actual access. This requires that the data source or data artefacts (e.g. a document) must be linked with technically enforceable conditions of use. This is done in the form of “usage control policies”. These policies can be enforced as long as the data remains in the target connector or is only passed on to compatible back-end systems.

The components mentioned above to implement access control and usage control can also be found in the IDS connector: with every request from the outside, an X.509 certificate is provided which uniquely proves the identity of the connector. This is done just once when the connection is established. This certificate bears the unique iden-

Figure 18: IDS architecture overview

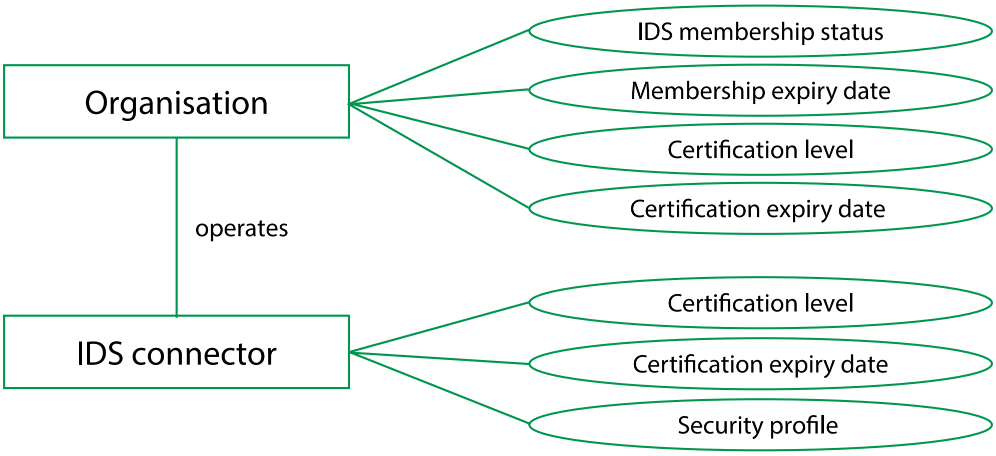
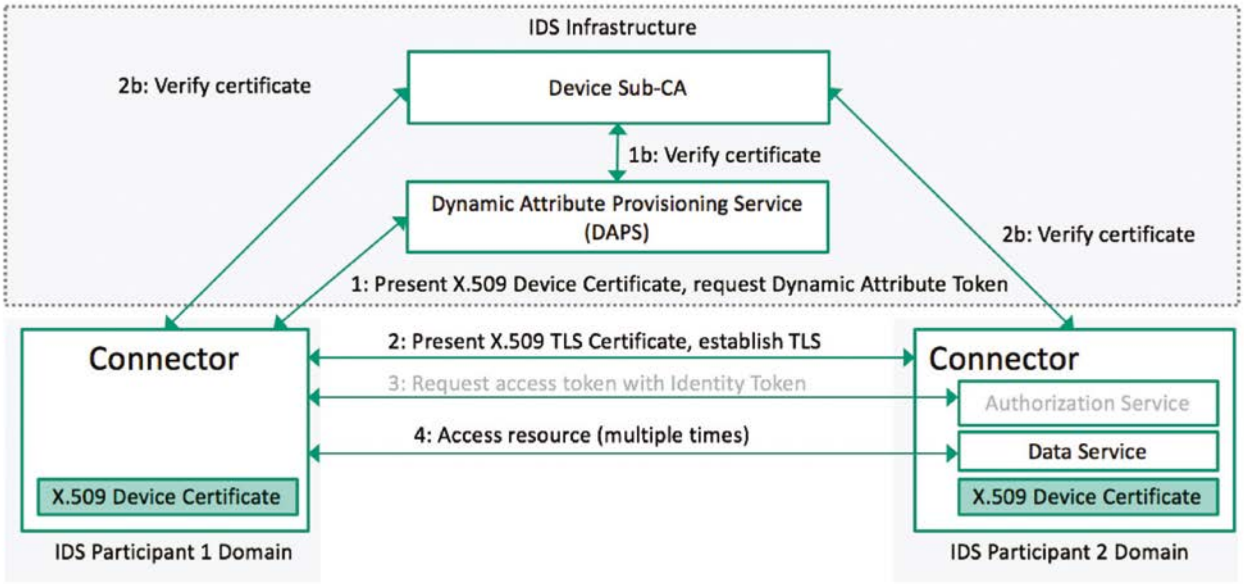


Figure 19: Sequence of attribute query and submission for data access



tifier of the requesting connector (the UUID). In addition, the dynamic attribute token, which was mentioned above, is presented and encapsulates the dynamic attributes of the connector (or the organisation which operates the connector).

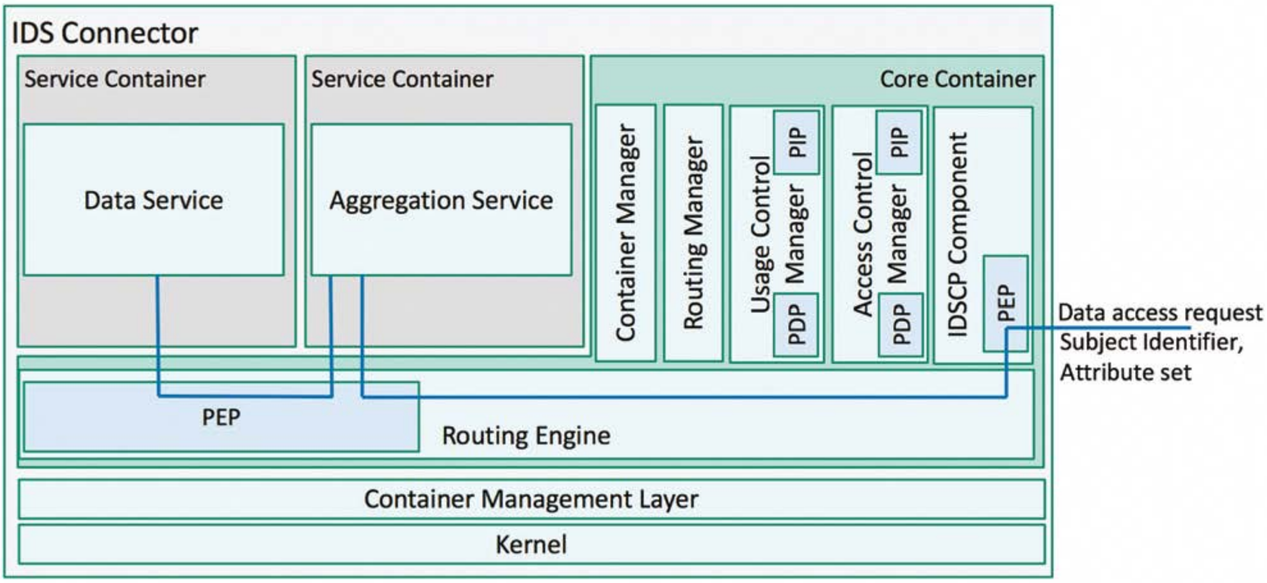
Whenever there is any data access from the outside, a Policy Enforcement Point (PEP) is activated. This PEP is contained in the "IDSCP component", which the IDS communication protocol offers for exterior use. The "Access Control Manager" controls the access by providing the associated PDP and PIP. Based on the stored set of rules and the attributes of the target connector, a decision is made on whether the access should be allowed. An extension is applied to implement the usage control. All permissible data flows between the data services are enabled by means of stored data routes. An integrated data flow control implements the data usage control.

Three fundamental data flows can be distinguished:

- Data flow between services within a connector (within a defined data route)
- Data flow into a data route (from outside the connector)
- Data flow out of a route (leaving the connector)

At every step, all data flows activate a PEP which checks the validity of the data flow based on stored policies. This allows the implementation of information flow control which can be integrated into a data usage control process. The necessary access control decisions for this PEP are mapped via a PDP which is stored in the "Usage Control Manager". But the specific storage location strongly depends on the usage control framework which is used. It is equally conceivable, for example, that the PDP and the corresponding components could be stored externally (with the associated implications for availability etc.). The other components such as PAP, PRP and PIP have been omitted for the sake of simplicity.

Figure 20: Data flow between connectors



3.4.3 Application of ABAC and integration into the administration shell

The access control operates based on the mechanisms mentioned above in section 3.1:

- The requesting connector submits an X.509 certificate. This provides a unique identification of the subject.

- The connector similarly submits a dynamic attribute token. This serves as a signed attribute container.

The connector provides the above architecture components to support ABAC directly. In any case, the PEP (or PEPs) are situated at the place of data access inside the connector. The other components such as PDP, PIP, PRP, PAP can be located inside or outside, depending on the implementation. The configuration "Trusted Connector" includes all components as encapsulated units.

Externally, the connector acts like an Industrie 4.0 entity and provides its own administration shell. Therefore, the concepts from "Administration shell in detail" can be mapped here:

The connector then contains an instance of the entity **AssetAdministrationShell** which has a reference to the entity **Security**. Here, as shown in Fehler! Verweisquelle konnte nicht gefunden werden., the appropriate information such as **AccessControlPoints** can be stored. As the concepts are identical, the concepts of the access control mechanism can also be mapped for access control and the provision of **AccessPermissionRules**.

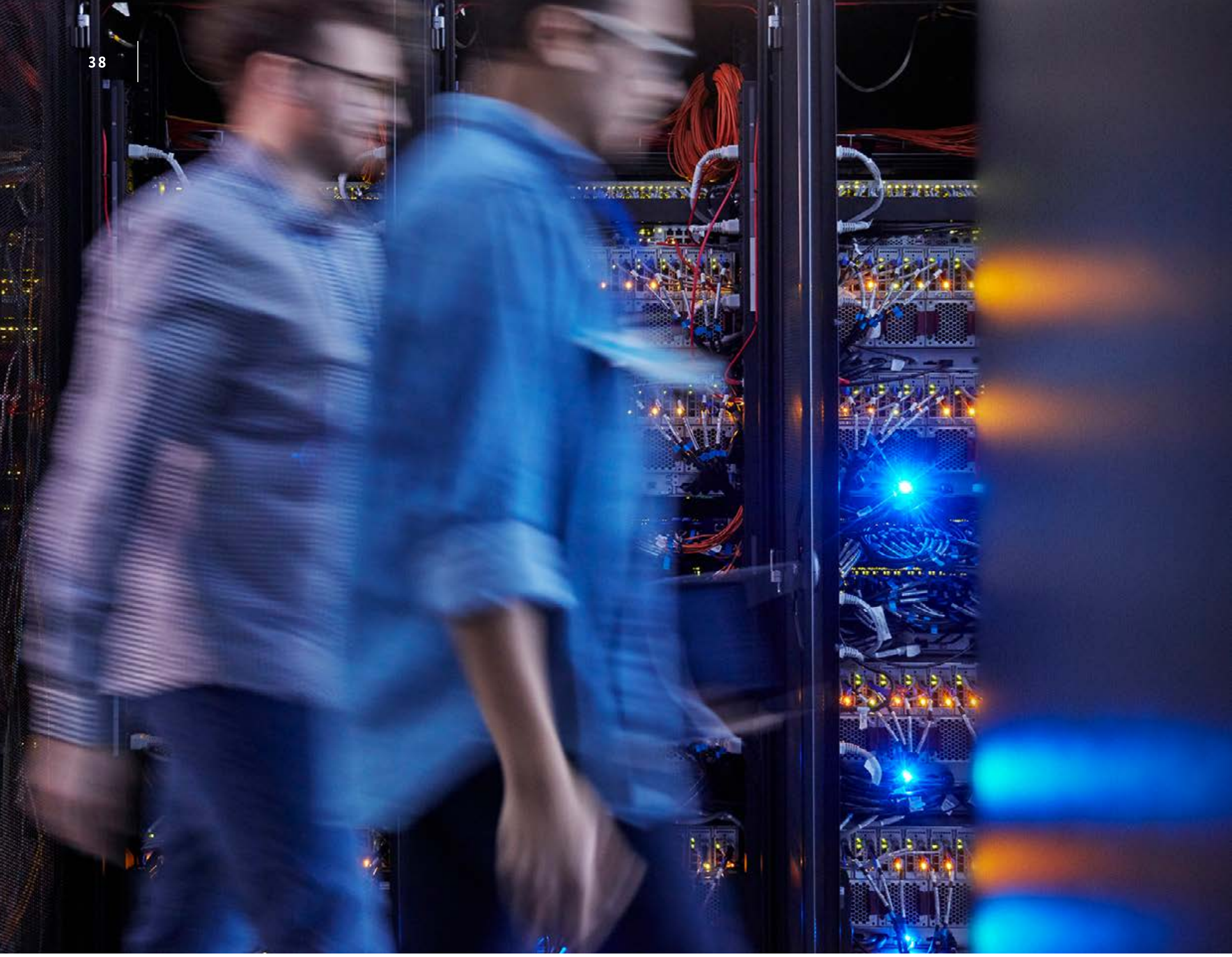
The attribute types mentioned can therefore be mapped as follows (3.2.1.3):

- **Subject attributes** are the attributes of the accessing connector against which the check is carried out.

- Industrie 4.0 **entity attributes** (stored in **objectAttributes**) can be mapped to the connector which makes the access decision.

- **Environment condition attributes** are mapped to the **selectableEnviromentAttributes**, which can be used in the formula to describe a condition.

The IDS connector acts like an Industrie 4.0 component and should be able to integrate smoothly into the corresponding concepts of the administration shell.



4 Summary and prospects

In access control by the Attribute Based Access Control method (ABAC), access is granted on the basis of a check of the subject, object and environment attributes. For this purpose, access rules are formulated, and their fulfilment is determined by comparing the target values and actual values of the attributes.

The rules or policies which can be implemented in an ABAC model are limited to the extent defined by the computing language. This flexibility allows the Industrie 4.0 entities to react with the greatest possible diversity to the great variety of objects, without the need to define individual relationships between every Industrie 4.0 entity and every object.

For example, an Industrie 4.0 entity is assigned a number of attributes when it begins its operation (e.g. the operator of a machine). An object (e.g. a machine) is assigned its object attributes when it is created (e.g. by the integrator of a machine). The values of this attribute can change in the course of its operation. One simple example is a change of the operating mode.

Environment objects may receive their attribute values either directly from other components (e.g. an integrated sensor, timer), other objects (e.g. an independent GPS receiver), from automated processes (e.g. microservices) or by alarm levels due to a threat situation (e.g. by a manufacturing execution system MES).

The manufacturer or operator of an object (e.g. a machine) prepares an access rule to define the set of permissible operations, e.g. service technicians with certain qualifications are granted access to the machine in the time from X (15) to Y (17) hrs. local time if the machine is in maintenance mode.

In addition to the flexibility of the logical access control model, attributes and their values can be changed throughout the life cycle of subjects, objects and attributes without the need to change each individual subject/object relationship. This offers dynamic access control because access decisions may change from one request to another if the attribute values change.

This capability makes it possible for object owners (e.g. operators) or subjects (e.g. service technicians) to apply access control policies to unknown subjects and an unlimited number of Industrie 4.0 entities when access requests are made.

If the operator integrates new Industrie 4.0 entities into its organisation, the access rules do not need to be changed for the existing Industrie 4.0 entities and their objects. It is sufficient to assign the new Industrie 4.0 entity the attributes which are necessary for access (e.g. a new administrator of the operator must be assigned the appropriate attributes).

This is one of the main advantages of using ABAC.

In the last few years, manufacturers have begun to implement functions similar to Attribute Based Access Control (ABAC) in their safety management and network operating system products without previously reaching a general agreement about a suitable set of ABAC functions. Due to the lack of a consensus about ABAC functions, these users are not reliably able to assess or use the advantages and challenges associated with ABAC.

Such an agreement on certain Industrie 4.0 ABAC functions and their configuration is the task for the next stage of integration into Industrie 4.0 components.

The prime focus of future documents for Industrie 4.0 is to achieve the goal of secure access control for the administration shell.

This organisational task will need to be described in one of the future documents on Industrie 4.0.

The requirements for the technical implementation to support Industrie 4.0 components, ranging from simple components to complex installations, will also be topics for the next documents.

This will also include standards which apply ABAC such as the eXtensible Access Control Markup Language (XACML) and ANSI/INCITS Next Generation Access Control (NGAC), which is based on the Policy Machine (PM) reference implementation.

XACML is widely accepted and implemented, and now it supports not only XML, but also interfaces such as JSON. The three sub-projects of NGAC, (1) the implementation requirements, protocols and API definitions; (2) the functional architecture (available as ANSI INCITS 499-2013) and (3) the generic operations & abstract data structures will be presented.

The access control mechanism has now developed from a concept to a reference. The appropriate implementations are subsidised as an open source project.

Therefore, the area of Industrie 4.0 now has efficient tools which are available for trustworthy access control and compatible with the goals and agreements for “Security in cyberspace” to protect the economy and civil society, as specified in the preamble, and which are technically and economically feasible.

5 Appendix

5.1 Identity technologies

There are a number of established identity technologies which use different strategies in an attempt to safeguard the secure creation and preservation of the integrity of previously uniquely determined subjects and their communication between identity providers and identity consumers.

Known frameworks must be examined to determine their suitability for secure deployment and secure attribute mapping to the access control system in the area of Industrie 4.0. The approaches include SAML 2.0, WS-Federation, LDAP, OAuth 2.0, OpenID Connect, Fido U2F, OpenOTP, TiQR, knowledge-based authentication (KBA), 2 factor authentication, frameworks with and without a public key infrastructure (PKI), public key enabling (PKE), global directory services (GDS) with a certification authority (CA), Common Access Card (CAC) and others.

In future this topic will be studied in greater detail by the subsidiary working group “Secure identities” within the working group “Security of networked systems” in Platform Industrie 4.0.

5.2 Glossary

IDM

Identity Management
Administrative system for individually distinguishable entities or identities

SAML

Security Assertion Markup Language
Open standard for the exchange of authentication and authorisation data between parties, especially between an identity provider (IDP) and a service provider (SP).

IDP

Identity Provider
SAML: Generates messages for the service provider (SP).

ID

Identity
A unique individually distinguishable entity which characterises an item or object, e.g. a natural person, an organisation (legal entity), a host name, a hardware or software component.

RBAC

Role Based Access Control
Role-based access control and monitoring for files or services. Was described in 1992 and approved in 2004 as the ANSI standard 359-2004.

ABAC

Attribute Based Access Control
RBAC systems have the general advantage that they can "bind" any identity "to a role" which can be used for high level access rights without changing it too often. ABAC is characterised by a deeper and more fine-grained permissions model which uses policies in combination with attribute values to enable a permission level which corresponds to the values presented. To achieve this, ABAC generally removes the identity and bases its decisions exclusively on these attribute values. In this way an RBAC policy, as a "Separation of duty (SOD) policy" – i.e. a system with a separation of powers – can significantly increase security and reduce risks, especially because "SOD" decisions are also made in real time.

X.509

Certificate format
In digital communication, X.509 certificates are used to authenticate and verify public keys. These certificates bind the public key to the identity of its owner in the framework of a public key infrastructure (PKI).

SP

Service provider
SAML: A system entity which receives and accepts authentication assurances in connection with a Single Sign-On profile (SSO) of the Security Assertion Markup Language (SAML).

RAdAC

Risk Adaptive Based Access Control
RAdAC is basically the process of receiving information about users, such as attributes, meta-data or other methods, in order to derive information and integrate the data into a risk engine algorithm.
This is important in highly classified systems. Here, ABAC works in conjunction with an "RAdAC-PDP", which thus becomes the de-facto access control mechanism.

Such ABAC solutions can enforce higher quality policies on the basis of classifications (e.g. by meta-tagging the data as "Secret", "Top Secret" and "Confidential"). RAdAC tools therefore normally have special risk algorithms.

OPC UA

Open Platform Communications Unified Architecture
The OPC Unified Architecture (UA), which was published in 2008, is a platform-independent, service-oriented architecture which integrates all functions of the individual OPC Classic specifications into an extendible framework.
This multi-layered approach achieves the following goals of the design specification:

- *Functional equivalence: all COM OPC Classic specifications are mapped to UA.*
- *Platform independence: ranging from an integrated micro-controller to a cloud-based infrastructure.*
- *Security: encryption, authentication and auditing.*
- *Extendible: possibility to add new functions without impairing the existing applications.*
- *Comprehensive information modelling: to define complex information.*

IDS

International Data Space
The goal of IDS is to ensure data sovereignty by providing an open, manufacturer-independent architecture for a peer-to-peer network which makes usage control of data from all areas possible.

PEP

Policy Enforcement Point
A point which intercepts the subject's request for access to an object and transmits a decision request to the PDP, receives the access decision (i.e. access to the object is allowed or denied), and then reacts to this decision.

PDP

Policy Decision Point
The unit which evaluates access requests on the basis of authorisation policies before it makes decisions on access.

PIP

Policy Information Point
The system unit which serves as the source of attribute values (i.e. an object, an environment, a subject).

PAP

Policy Administration Point
The point where the access permission policies are administered.

PRP

Policy Retrieval Point
The point where the access permission policies are stored, typically a database or the file system.

XACML

eXtensible Access Control Markup Language
XACML is an open XML-based standard language which was developed to express security policies and access rights to information for web services, digital rights management (DRM) and company security applications. XACML was ratified in February 2003 by the Organization for the Advancement of Structured Information Standards (OASIS) and was developed to standardise access control via XML, for example so that a member of staff can access several linked websites on the basis of a single log-in. XACML is sometimes called Extensible Access Control Markup *(XACL).*

MAC

Mandatory Access Control
Mandatory Access Control (MAC) is based on a collection of security policies which are restricted depending on the system classification, configuration and authentication. The administration and creation of MAC policies is carried out in a secure network and is limited to system administrators. MAC defines and guarantees central enforcement of confidential parameters relevant to security.

DAC

Discretionary Access Control
Discretionary Access Control is a type of security access control which allows or restricts access to objects by means of an access policy which is defined by the owner group and/or the persons associated with an object. DAC mechanism controls are defined by user identification on the basis of the log-in information provided during the authentication, such as the user name and password.

5.3 History of access control

Originally, **IBAC** (Identification Based Access Control) was used. But because users often also used different identities, IBAC was no longer practicable (1) in distributed systems and (2) for a rapidly growing number of users. Therefore it was quickly supplemented by methods such as "Owner" or "Groups", and this led to **MAC** (Mandatory Access Control) [7] and **DAC** (Discretionary Access Control) [8]

Mandatory Access Control (MAC) is a security strategy which provides a strict control of access rights. MAC policies define which resources a user can access in a file system and which resources the user cannot access. MAC systems are often used in government and military organisations. Each object in a file system is assigned a security level. Examples include: "confidential", "secret" and "top secret". Each user and each device in the system is assigned a comparable classification. Whenever a person or device wishes to access a certain resource, the operating system or security kernel checks the access rights and decides whether the access is allowed. Although mandatory access control is one of the most secure access systems, it requires very carefully planning and continuous review of the assigned rights – both for objects and for users.

The opposite of MAC is Discretionary Access Control (DAC). This system allows users to define their own policies and access rules. DAC mechanism controls are defined by user identification on the basis of the log-in information provided during the authentication, such as the user name and password. DACs are discretionary because the affected party (owner) is able to assign authenticated objects or access to information to other users. In other words, the owner determines the rights of access to objects.

ABAC systems are able to enforce both Discretionary Access Control (DAC) and Mandatory Access Control (MAC) models.

In addition, ABAC systems can facilitate risk-adaptive access control solutions (RAdAC) in which the risk values are expressed as variable attributes. These systems are similar to MAC systems in the functions offered.

In further documents issued by the Platform Industrie 4.0, access control for highly classified systems will also be considered.

ABAC as an access control model was the result of the work of the US Federal Chief Information Officers Council (Federal CIO Council) which published the Federal Identity, Credential, and Access Management (FICAM) Roadmap and Implementation Plan v1.0 in November 2009 to support government organisations in the development of their logical access control architectures. The aim was to evaluate attributes as a possible criterion to provide access within and between organisations in the whole of the government structure. In December 2011, the FICAM Roadmap and Implementation Plan v2.0 represented the next step in which ABAC was declared to be the recommended access control model to promote the exchange of information between different organisations. In 2014 the NIST published the document NIST 800-162 under the title "Guide to Attribute Based Access Control (ABAC) Definition and Considerations" [4], in which ABAC was described in greater detail. Parallel to this process and under the leadership of OASIS, the description language XACML [9] for access control systems was developed with specifications which first appeared in 2003. XACML supports RBAC and ABAC.

List of figures

5.5 Literature

[1] EU Commission: Final report of the seventh round of mutual evaluations on "The practical implementation and operation of the European policies on preventing and combating cybercrime", 2017.
https://data.consilium.europa.eu/doc/document/ST-12711-2017-INIT/de/pdf

[2] Ravi Sandhu, David Ferraiolo, Ravi Sandhu: NIST model for role-based access control: Towards a unified standard, 2000. https://www.researchgate.net/publication/220855076_NIST_model_for_role-based_access_control_Towards_a_unified_standard

[3] NIST: Role Based Access Control, 2016.
https://csrc.nist.gov/projects/role-based-access-control/role-engineering-and-rbac-standards

[4] NIST 800-162: Guide to Attribute Based Access Control (ABAC) Definition and Considerations, 2014.
https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.sp.800-162.pdf

[5] NIST: Attribute Based Access Control, 2016.
https://csrc.nist.gov/projects/abac/

[6] Platform Industrie 4.0: Security of the administration shell.
https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/security-der-verwaltungsschale.html

[7] TechTarget: Mandatory Access Control (MAC).
https://www.searchsecurity.de/definition/Mandatory-Access-Control-MAC

[8] Techopedia: Discretionary Access Control (DAC).
https://www.techopedia.com/definition/229/discretionary-access-control-dac

[9] OASIS: OASIS eXtensible Access Control Markup Language (XACML) Version 3.0, 2017.
https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

[10] OASIS: OASIS Security Services (SAML), 2012.
https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[11] Platform Industrie 4.0: The administration shell in detail – from the idea to the implementable concept. https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/verwaltungsschale-im-detail-pr%C3%A4sentation.html

[12] Platform Industrie 4.0: Details of the Administration shell, Part 1 – The exchange of information between partners in the value chain, Release 1.0.
https://www.plattform-i40.de/I40/Navigation/DE/In-der-Praxis/Online-Bibliothek/online-bibliothek.html

[13] Platform Industrie 4.0: Discussion document "Secure procurement of CAE data".

AUTHORS

Carsten Angeli, Kuka Deutschland GmbH | Dr Birgit Boss, Robert Bosch GmbH | Dr Andre Braunmandl, Federal Office for Information Security | Gerd Brost, Fraunhofer AISEC | Michael Claus, Schuler AG Göppingen | Olaf Dressel, Bundesdruckerei GmbH | Torsten Förder, Phoenix Contact Software GmbH | Kai Garrels, ABB STOTZ-KONTAKT GmbH | Dr Michael Hoffmeister, Festo AG & Co. KG | Dr Detlef Houdeau, Infineon Technologies AG | Dr Lutz Jänicke, Phoenix Contact GmbH & Co. KG | Michael Jochem, Robert Bosch GmbH | Sebastian Piecha, Huawei Technologies Deutschland GmbH | Thomas Walloschke (Chairman), Fujitsu Technology Solutions GmbH | Dr Michael Schmitt, SAP SE

This publication is a joint result of the working groups “Security of Networked Systems” and “Reference Architectures, Standards and Standardization” (Plattform Industrie 4.0).

www.plattform-i40.de