Pattern-based access control in a decentralised collaboration environment

Jeroen Werbrouck[1,2], Ruben Taelman[3], Ruben Verborgh[3], Pieter Pauwels[4], Jakob Beetz[2], and Erik Mannens[3]

[1] Department of Architecture and Urban Planning, Ghent University
[2] Department of Design Computation, RWTH Aachen University, Aachen, Germany
[3] Department of Electronics and Information Systems, Ghent University – imec
[4] Department of the Built Environment, TU Eindhoven

Abstract. As the building industry is rapidly catching up with digital advancements, and Web technologies grow in both maturity and security, a data- and Web-based construction practice comes within reach. In such an environment, private project information and open online data can be combined to allow cross-domain interoperability at data level, using Semantic Web technologies. As construction projects often feature complex and temporary networks of stakeholder firms and their employees, a property-based access control mechanism is necessary to enable a flexible and automated management of distributed building projects. In this article, we propose a method to facilitate such mechanism using existing Web technologies: RDF, SHACL, WebIDs, nanopublications and the Linked Data Platform. The proposed method will be illustrated with an extension of a custom nodeJS Solid server. The potential of the Solid ecosystem has been put forward earlier as a basis for a Linked Data-based Common Data Environment: its decentralised setup, connection of both RDF and non-RDF resources and fine-grained access control mechanisms are considered an apt foundation to manage distributed building data.

Keywords: Web Access Control · Decentralisation · Linked Building Data · Common Data Environment · Nanopublications

1 Introduction

When envisaging a decentralised management of construction projects throughout their life cycle, one of the main hurdles is to organise access to restricted data, considering the complex network of contractors, subcontractors, companies and employees. Each one of them has different roles and responsibilities, and may be actively involved in multiple construction projects at the same time. As indicated in [12], a decentralised environment for hosting building data requires either a Role-Based Access Control (RBAC) mechanism or a property-based one (also called Attribute-Based Access Control (ABAC)). To express more complex patterns, other contextual information might be taken into account, such as the content of the requested resource itself or assertions made by multiple parties.

In this case a Context-Based Access Control (CBAC) mechanism is used [7]. A recurring analogy to describe such context- or property-based approach in a general sense is that one might not be able to name firefighters or paramedics beforehand, but they can be given access if they are able to prove their function, e.g. with valid certificates. In context of the construction industry, this could be a (partial) delegation of access rights from contractors to subcontractors. Or, before getting access to a certain resource, a client should prove she is involved in the project as a 'leading architect' and at the same time demonstrate her membership of a recognised association of architects. Such flexibility is offered by existing Semantic Web technologies.

One of the main features of the Semantic Web is that it provides an enormous flexibility to express knowledge of varying complexity. Recent developments have added to the already existing technology stack the ability to validate such statements against a certain set of conditions, also called 'shapes'. Consequently, using a Property-Based Access Control, where Anyone can say Anything about Anything (the famous AAA paradigm of the Web), an assertion can be validated to contain specific information, and its author can be checked. The result of the validation determines access to specific data. Although this concept generally does not depend on any existing platform, a starting point for implementation could be the Solid platform [11,17,3], initiated by Tim Berners-Lee and actively developed by Inrupt, Inc.[5]. Solid provides an infrastructure where data and applications that use this data are separated, allowing the owners of the data to give and revoke access to it at any given time and ensuring maximal reusability of information. As indicated in [19], it has some properties that make it a promising candidate to provide the basis for a construction-oriented platform, i.e. a decentralised Common Data Environment (CDE) [6]. In this scenario, project data is stored in 'pods' (Personal Online Data storage) per stakeholder: semantic data is stored in graph files using RDF ontologies, other data (geometry, imagery etc.) can be file-based, according to the Linked Data Platform (LDP)[6] specifications. A 'pod' is connected to a WebID, i.e. a HTTP URI referring to a particular Agent (Person, Organisation etc.)[7].

In the search for technologies that support decentralised management of building data, there is no need to reinvent the wheel; a few well-chosen adjustments to the basic infrastructure could already answer many questions about how a decentralised CDE could look like. In other words, because the initial use case of Solid was to provide a decentralised alternative to social networks, its 'vanilla' server implementation is suited, but not yet optimised for use in the construction industry. For example, the emulation of role-based access patterns by setting up lists with WebIDs, corresponding to certain roles, is probably sufficient for many (smaller) construction projects. However, an extension might be necessary for managing larger project consortiums. Secondly, the performance of the directory-based approach of the LDP specification versus the fully data-

[5] `https://inrupt.com/`
[6] `https://www.w3.org/TR/ldp/`
[7] `https://dvcs.w3.org/hg/WebID/raw-file/tip/spec/identity-respec.html`

based approach of triple stores needs to be benchmarked. Thirdly, a distinction in pod organisation might be useful: other requirements can be present for stakeholder companies, their employees and 'project pods' containing basic information about the project that cannot be linked to one individual stakeholder (e.g. containing project planning or exchange requirements). This publication will only assess the first requirement.

In this paper, we propose a framework to allow property- and context-based access control in a decentralised system. This relies on existing web-standards and practices and is thus independent from existing platforms or ecosystems. However, since the Solid ecosystem already offers a quite extensive implementation of standards such as LDP, it will be used as an implementation use case. As the proposed method will rely on shape patterns, it will be called 'pattern-based' for the remainder of the text. After an overview of existing technologies upon which we rely is given in Section 2, Section 3 discusses the basic properties such a framework needs to incorporate. Section 4 then illustrates these properties with an implementation based on the node-solid-server[8] and an example request featuring the proposed method. A conclusion and future ambitions are the main topic of Section 5.

The paper forms part of the conSolid project, which aims to build an interoperable platform based on the Solid specifications, where specialised, 'tailormade' applications can use distributed building data throughout the building's life cycle; whether for design or simulation purposes, to manage inhabitant comfort preferences or link to historical datasets.

2 Related Work

2.1 Web-based building data

Over the last decade, the Architecture, Engineering, Construction and Operations (AECO) industry has been adopting digital techniques at an unseen rate; after a long digital dormancy, the sector is finally catching up with technological innovations that have long boosted productivity in virtually all other economic sectors. This new technological *hausse* is largely due to the emergence of Building Information Modelling (BIM), combined with rapid developments in the field of cloud services for management of building projects: advanced digital systems are set up to streamline project organisation and information exchange between stakeholders, guaranteeing a long-time preservation of data that may be of use in future phases of the building life cycle (BLC). Such environments are commonly referred to as Common Data Environments (CDEs).

Oftentimes, the providers of CDEs are the same companies that develop BIM authoring tools. This enables the setup of a highly integrated software suite and results in an optimised project management. However, it may also result in a vendor lock-in, making it more difficult to use software that is not included in the ecosystem. Since many building projects are only temporary collaborations

[8] `https://github.com/solid/node-solid-server`

between stakeholder offices, this scenario occurs quite frequently: every office has its own software stack, depending on their workflow, area of expertise and budget. To still be able to facilitate communication between software packages that internally use different (proprietary) file formats and data models, most BIM tools facilitate export and import of the Industry Foundation Classes (IFC) (ISO 16739), maintained by BuildingSMART International[9]. Closely related to this is the more recent OpenCDE[10] initiative, also covered by BuildingSMART, in which multiple CDE developing companies collaborate to establish an interoperable REST (Representational State Transfer) specification that will allow CDEs from different vendors to exchange project information more easily.

Current BIM authoring tools and CDEs have a strong focus on information exchange via documents [6], however, a data-based approach based on Semantic Web technologies is getting more and more attention. These technologies include, amongst others, the Resource Description Framework[11] (RDF), the Web Ontology Language[12] (OWL), the SPARQL query language[13] and the Shapes Constraint Language[14] (SHACL), all relying on the atomic building blocks of the world wide Web, namely URIs[15]. There is a growing consensus that the application of Semantic Web technologies can reduce information loss, improve interoperability and cross-domain collaboration, and enable automated checking of rules and regulations [1,14].

A considerable research community is currently involved with establishing and enhancing the foundations for a Web of building data. Probably the most well-known open source solution that is not fundamentally based on Semantic Web technologies is TNO's BIMserver [2]. IFC files form the main input for a BIMserver, a central repository from which data can be used by multiple microservices or 'BIM bots'[16]. In the realm of Linked Data, apart from ontologies for the built environment such as ifcOWL [13], the Building Topology Ontology (BOT) [16], the Building Product Ontology (BPO) [18] and the Ontology for Property Management (OPM) [15], projects such as the DRUMBEAT platform [5] and the LBDserver[17] focus on providing the infrastructure specifically for management of linked building data. As mentioned in Section 1, a more general approach is taken by the Solid initiative, which is completely independent from the AECO industry, but shows intriguing overlaps with ambitions for a web-based building collaboration environment.

[9] https://www.buildingsmart.org/
[10] https://github.com/buildingSMART/OpenCDE-API
[11] https://www.w3.org/TR/rdf11-primer/
[12] https://www.w3.org/TR/2012/REC-owl2-primer-20121211/
[13] https://www.w3.org/TR/sparql11-query/
[14] https://www.w3.org/TR/shacl/
[15] https://www.w3.org/wiki/URI
[16] https://www.youtube.com/watch?v=lyDSpTV6NQI
[17] https://github.com/JWerbrouck/lbdserver_project

2.2 The Solid Ecosystem

The advocates of Solid envisage a new way of using personal data in Web environments: by splitting up data and applications, people are put in charge of who has access to their personal data, and they can revoke these access rights at any given time. A use case only gaining relevance, given late concerns regarding the use of personal information by social network enterprises, and the recent legal response to these practices (e.g. the General Data Protection Regulation (GDPR) enacted by the EU). The core of Solid consists of a set of Web standards and practices, which can support a plethora of use cases in different domains. A combination of these standards with recent developments of modular vocabularies in the field of architecture and construction (Section 2.1), and networks of specialised services for specific tasks in the BLC, looks a promising strategy towards a data- and Web-driven construction project collaboration. A scenario can be sketched where multiple stakeholders of a construction project have their own Solid pods, where they manage their information about the project. The infrastructure offered by the Solid ecosystem then acts as a decentralised CDE, the semantic glue for interpreting this distributed information as a linked-data based digital twin, where typical BIM data produced by different stakeholders is connected to sensor data, geospatial data, historical data etc.

2.3 Access Control patterns in AECO

Multiple strategies currently co-exist on the Web to grant access to specific information, among which Discretionary Access Control Lists (DAC) and Role-Based Access Control (RBAC) are the most known ones. Where a DAC links a list of actors to a piece of information, RBAC allows, for example, groups of people with the same role to access the data. Both strategies are supported in Solid. However, as indicated in section 1, relationships within a building project often are more complex than this: a project involves a network of contractor and subcontractor companies, each one of them might have multiple roles and responsibilities, and access rights might differ internally between their respective employees as well. Some criteria are identified by [12], among which the possibility to express arbitrary access rules, such as:

- All employees of company X working in project Y;
- Inhabitants of the respective building;
- The facility manager of Project Z.

[12] mentions that for this kind of access rules, a Role- or Property-based Access Control mechanism is required. Since in the end, file content, company structure and other contextual information might also play a role, we extend this to Context-Based Access Control. Since one of the end goals of the conSolid project is to set up a useful ecosystem for construction professionals, establishing reusable patterns is one of the key priorities. Explicit references to certain individuals can already be expressed using the default ACL implementation in

Solid; implicit references form the main use case of the framework proposed in this paper. I.e. 'all employees of company X' may be expressed using ACL agent groups; 'all employees of the company that is responsible for architectural design of the project this resource belongs to' could be a reusable, property-based pattern that may be expressed using the method explained in Section 3. For such scenario to happen, at least two warrants may be needed: one stating that the employee works for company X (signed by company X or one of its 'full' delegates) and another one indicating that company X is indeed involved in Project Y (signed by one of the authorised project delegates). Although it would probably be possible to express these patterns in the default ACL implementation in Solid (e.g. by hardcoding the WebIDs of these people into 'agent-groups'), complex patterns that pose multiple requirements to visitors will be expressed and verified with more ease using a pattern-based approach.

3 Pattern-Based Access Control

This section describes the foundations of the pattern-based ACL framework. Namespaces and assigned prefixes that are used in the remainder of the paper are indicated in Listing 3.1. The framework is meant to extend beyond typical use cases of the AECO industry and be generally applicable, although regulating access to information within construction projects remains the primary target. In this paper, we define a new vocabulary for Pattern-based Access Control (PBAC). The ConSolid (CS) vocabulary, which will be used to indicate professional relationships in a construction project, is also part of the conSolid project, but has not been published at the time of writing.

```
# ontologies:
    @prefix pbac: <https://jwerbrouck.inrupt.net/public/PBAC/PBAC-ontology.ttl>
    @prefix cs: <http://www.consolid.org/ontology/cs#> .
    @prefix np: <http://www.nanopub.org/nschema#> .
    @prefix sh: <http://www.w3.org/ns/shacl#>.
    @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
    @prefix acl: <http://www.w3.org/ns/auth/acl#>.
# the document itself:
    @prefix : <#>.
# example webIDs:
    @prefix alice: <https://alice.engineers.com/profile/card#> .
    @prefix bob: <https://bob.architects.com/profile/card#> .
    @prefix proj: <https://exampleproject.consolid.org/profile/card#> .
```

Listing 3.1. Namespaces used throughout this publication

3.1 Trusting statements on the Web

By default, the Web is an open framework, where people can express anything they want, from the brightest theories to the purest nonsense. To take everything on the Web as truth would be very naive indeed. For a pattern-based access control to function properly, a mechanism to prove the statements thus needs to exist: how to know the assigned properties are valid? At the moment,

probably the closest we can get to verification of assertions is to use a system of provenance, similar to the system used in TLS certificates, ultimately leading to a trusted 'root authority'. Therefore, a provenance-sensitive way of expressing statements is necessary. Such provenance-based system is provided by the Nanopublications concept. Nanopublications [4,9] are 'a community-driven approach to representing structured data along with its provenance into a single publishable and citable entity'[18], and are mainly used in the field of bioinformatics and life sciences. As the name indicates, they represent very small assertions, along with provenance and metadata. A common format for expressing nanopublications is the TriG[19] format, an extension of the Turtle notation[20] to include multiple named graphs in one single document. A nanopublication thus essentially is a set of RDF quads, following a fixed template that consists of four tightly interconnected named graphs:

- A 'Head' graph, linking the different parts of the nanopublication to the nanopublication document itself.
- An 'Assertion' graph, containing the actual assertion.
- A 'Provenance' graph, indicating where a certain statement comes from. Depending on the case, this could be calculation input, an authority approving the statement etc.
- A 'Publication Info' graph, stating the metadata about the publication itself: authorship, publication date, signature etc.

To keep nanopublications immutable and verifyable, they can be integrated with a combination of digital signatures, using RSA key pairs, and 'Trusty URIs' [10]. The author of the nanopublication can digitally sign the document and then 'freeze' it using Trusty URIs: the content of the graph is hashed and embedded in the URI by which the document refers to itself[21]. A Java implementation to sign nanopublications and make them trusty at once is described in [8]. A local server implementation to sign these publications with one's WebID is discussed in Section 4.1.

Digitally signed, 'trusty' nanopublications play a major role in the proposed framework for pattern-based Access Control: if signed by a trusted authority, the assertion can be checked against a certain SHACL shape, which may or may not grant access to the requested resource, depending on the validation outcome. SHACL is a recent W3C standard designed to validate graphs against a given set of conditions. Unlike OWL, SHACL uses a closed-world paradigm, meaning that if something is not explicitly present, it is deemed false. This renders it an apt method for regulating access to online resources. To avoid confusion with other existing types of nanopublications, nanopublications which specifically relate properties to certain agents will be called 'nanocredentials' for the remainder of the paper.

[18] `http://nanopub.org/guidelines/working_draft/`
[19] `https://www.w3.org/TR/trig/`
[20] `https://www.w3.org/TR/turtle/`
[21] Note that this does not need to match a URL where the document can be found.

A final requirement is that the authors of the assertion expressed in the nanocredential are actually trusted. An assertion may match the SHACL shape that grants access to the requested resource, but if anyone can write down this assertion, the framework is not very powerful. Therefore, a server implementing the framework must know which authors can be trusted for expressions matching the SHACL shape. This trust may be established in a general way or within a limited scope. A general way would mean these people are 'blindly trusted' for their expressions; a limited scope of trust could be indicated by linking their WebID to specific statements (e.g. in the form of SHACL shapes refered to in a pattern-based access rule (Section 3.2). The development of a framework to strike a balance between these two extremes is outside the scope of this paper, but may base itself upon the PBAC vocabulary.

To summarise, three main components are identified for a minimal pattern-based Access Control framework:

- Trusty and digitally signed nanopublications for relating properties to the requesting agent ('nanocredentials');
- SHACL shapes indicating the requirements that are needed to get access to the resource;
- 'Trusted authorities' that can be expressed both in a very specific way and in a general sense.

In Section 3.2, these components will be combined into a method for expressing pattern-based access rules for distributed building data.

3.2 Method

The WAC[22] ontology forms the basis of the framework. If the requirements set by a pattern-based rule are met, ACL modes (Read, Write, Append, Control) will be allowed for a given visitor. Along with SHACL shapes and trusted authorities, a dynamic rule (`pbac:DynamicRule`) thus contains information about the ACL modes it grants. The connection to one or multiple (local or remote) SHACL shapes is established by the property `pbac:hasShape`. The difference between an 'inclusive' rule (a visitor needs to conform to only one of the mentioned shapes), or an 'exclusive' one (all shapes need to be met before the visitor is granted access), may be established by linking `pbac:hasShape` to a locally defined shape, which may combine different (possibly remote) shapes through various Boolean operators (`sh:and` or `sh:or`) (Example: Listing 4.2).

Section 3.1 mentions the necessity for trusted authorities to be indicated, either in a very specific way or generally. The most fine-grained way for indicating trusted authorities is to directly link shapes and trusted authorities. In this case, a local triple could be added to the document linking the shape to a `pbac:TrustedAuthority`. Another indication of a rule's trusted authorities is to embed them as a property of the `pbac:DynamicRule`, which can be done via

[22] `https://github.com/solid/web-access-control-spec`

`pbac:hasTrustedAuthority` (Example: Listing 4.2). In both scenarios, their authority is limited to the `pbac:DynamicRule` of interest, and a well-defined boundary is determined for trusting their statements. However, in certain cases (and for a more easy setup of rule frameworks) one might want to express authorities in a broader sense, for example within the scope of an entire directory, and then cascading down to its specific subdirectories and resources. For example, in the case of a distributed building project, we can imagine that there is a project pod containing the basic specifications of the project [19]. A stakeholder of the project may indicate at the top-level folder where their project data reside, that all nanopublications signed by the project pod WebID may be trusted. This authority then 'boils down' to the resources (graphs, geometry, imagery ...) stored in this folder.

When requesting access to a resource on a remote conSolid server, a client then informs the server about the nanocredentials it wishes to present. However, not all of them can be made public; and the same holds for shapes protecting a resource or the authorities that are trusted within a certain scope. Secondly, an agent may have lots of nanocredentials: to prevent a waste of precious bandwidth and server resources it is undesirable to present them all along with the request. Many use cases may therefore impose some negotiation steps [7]. Different strategies may be used here, depending on the level of trust between visitor and owner of the resource. An 'open strategy' is to refer to a public shape, a more controlled one could be a step-by-step release of requirements. Relating this to construction projects, such step-by-step approach may balance the need to keep (access) information internal to the project and the need to explain to stakeholders why they cannot access certain information, and who they should contact if this is to be changed. After a first requirement is met ('the visitor is stakeholder in the project ...'), the server could choose to 'release' the other shapes, thereby providing specific information about any other conditions that need to be fulfilled ('... with the task of performing a structural analysis'). As the SHACL specification includes the possibility to generate detailed validation reports, textual as well as machine-readable explanations may be sent to a stakeholder whose request just got rejected, which is one of the challenges mentioned in [7].

An elaborate discussion of possible negotiation strategies extends beyond the scope of this publication. Briefly, they may rely on the exposure of a visitor's nanocredentials via a restricted service (e.g. a SPARQL endpoint) and facilitate information exchange between the conSolid server hosting the resource of interest and the endpoint owned by the agent requesting access. Depending on the degree of publicness of the nanocredentials on the endpoint's side and the SHACL shapes on the conSolid server side, a series of HTTP requests may be performed back-and-forth before the final validation takes place. A schematic example of such negotiation is given in Figure 1.

During the validation step, SHACL shapes in each *relevant* rule are validated against the collection of present nanocredentials. A rule is relevant when it yields an ACL mode that has not been granted already (e.g. because the re-

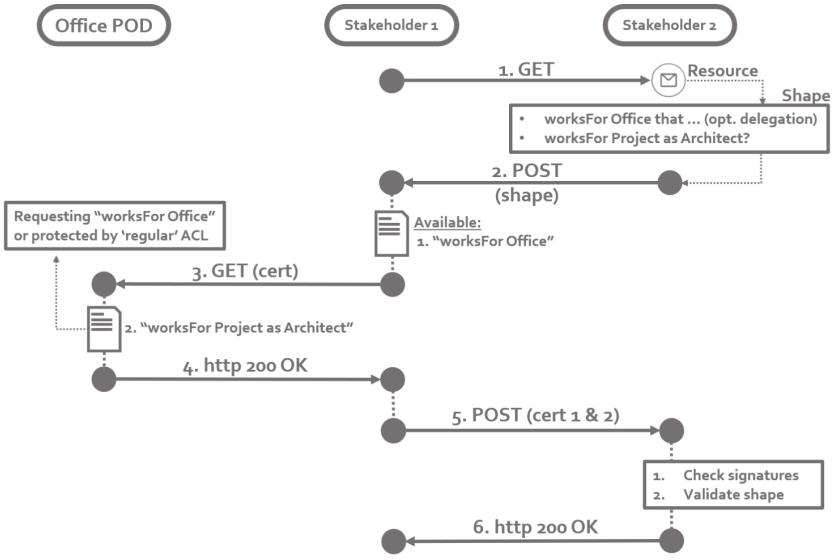

Fig. 1. Example negotiation: requesting multiple signed nanocredentials from different agents; the SHACL shapes protecting the resource are public.

quester is already mentioned explicitly in the ACL graph for the requested ACL mode). Since the identity of all possible visitors cannot be implemented directly in the shape (as this would totally undermine the purpose of the framework), the `sh:TargetNode` (i.e. the nodes against which the shape constraints are checked) of the SHACL shape relates to a dedicated resource, namely `pbac:visitor`. This implies that, at runtime, the `pbac:visitor` is changed to the WebID of the requesting agent. If the validation passes, the resulting ACL mode(s) are added to the array of allowed ACL modes, and the the visitor may proceed.

4 Implementation

4.1 ConSolid server

The framework outlined in Section 3.2 has been partly implemented in an adaptation of the node-solid-server. The resulting conSolid server is available on Github[23]. Up and running, the server can be tested using both browser requests and dedicated software such as Postman. Code for signing nanopublications using Solid credentials and 'freezing' them with trusty URIs is under development, a testing version is available at Github[24], relying on the code developed in [8]. To enable validation of the nanopublication, the corresponding public key needs to be present in the WebID graph of the signer. Section 4.2 illustrates the workflow defined in Section 3.2 using the abovementioned server implementation.

[23] `https://github.com/JWerbrouck/consolid-server`
[24] `https://github.com/JWerbrouck/validator-bot`

4.2 Example

As a summarising example, consider the following situation: the engineer of a construction project, Alice, needs information about the building's topology, which is mentioned in the project repository of Bob, the architect, his pod as 'topology.ttl'. Apart from the project engineer, project architects have access to this information, too (Listing 4.2). Alice sends a nanocredential, signed by the project pod (`proj:me`), along with her request, as a confirmation that she is involved in this project as a `cs:LeadingEngineer` (Listing 4.1). Before validation takes place, an inference engine could infer implicit statements, increasing the chances of success (e.g. a `cs:LeadingEngineer` is an `rdfs:subClassOf` a `cs:Engineer`, so if the Dynamic Rule allows instances of `cs:Engineer` to read the resource, instances of `cs:LeadingEngineer` should also be allowed access).

```
# asserted in the nanocredential
proj:me cs:hasLeadingEngineer alice:me .

# inferred triples (before validation)
proj:me cs:employs alice:me ;
       cs:hasEngineer alice:me .
alice:me a cs:LeadingEngineer, cs:Engineer ;
       cs:leadingEngineerOf proj:me ;
       cs:engineerOf proj:me ;
       cs:isEmployedBy proj:me .
```

Listing 4.1. Assertion graph in Nanocredential 1

The graph 'topology.ttl' is regulated by a dedicated ACL file, 'topology.ttl.acl'. Apart from some standard ACL access rules, the ACL defines a Dynamic Rule (Listing 4.2). Other rules with different requirements could be present in the ACL document as well. The `pbac:hasTrustedAuthority` declaration is implemented directly in the shape, although this could also be mentioned at a higher level or linked to the each of the individual shapes, as indicated in Section 3.2.

```
:ReadRule a pbac:DynamicRule;
    rdfs:comment "Allows project engineers and architects to READ the given resource.";
    pbac:hasShape :superShape_1;
    pbac:hasTrustedAuthority <https://consolidproject1.inrupt.net/profile/card#me>;
    acl:accessTo <topology.ttl>;
    acl:mode acl:Read.

:superShape_1 a sh:NodeShape ;
    sh:targetNode pbac:visitor;
    sh:or (
        <https://bob.architects.com/projects/BuildingProject1/EngineerShape.ttl>
        <https://shapes.consolidproject.be/shapes/ArchitectShape.ttl>
  ) .
```

Listing 4.2. Example ACL file enhanced with a PBAC rule

Only one SHACL shape needs to be valid in this example in order to complete the validation: both people conforming to the engineer shape and the architect shape are allowed to read the resource. Shapes needs to be dereferenceable and accessible for the validation engine, but can be located anywhere on the Web (e.g. in the agent's Pod, the project Pod or in a public repository, available

for reuse). Furthermore, if combined with a `sh:or` statement, they should be oriented towards the same targetNode, which might impose strict agreements on the shapes to use.

```
proj:EngineerShape
  a sh:NodeShape ;
  sh:targetNode pbac:visitor;
  sh:property [
           sh:path cs:isEngineerOf;
           sh:hasValue proj:me;
       ] .
```

Listing 4.3. Shape graph of the PBAC rule in Listing 4.2

For this example, the SHACL validation report does not contain any errors, so Alice can proceed her request to read the resources in the 'topology.ttl' graph, and use it within a linked building data web service, potentially in combination with other data that was retrieved in the same way, using a subset of her nanocredentials and those applying to her office. The full nanocredential example used in this publication can be found at `https://github.com/JWerbrouck/validator-bot/blob/master/example_np_consolid.trig`

4.3 Future work

In this section, an experimental implementation of the framework was applied to an adaptation of the node-solid-server. Based on this framework, an example was then discussed where one stakeholder of a certain building project proved her involvement and role in the project to another stakeholder, by sending one of her nanocredentials. After verification of the nanocredential, and validation against the shape mentioned in the PBAC rule 'protecting' the requested resource, read permissions were granted.

It may be clear that the approach taken in this paper only scratches the surface of this topic, and that additional work in several fields is required. In the short term, a clear approach for delegating access will be devised. Especially in construction, stakeholders will be mainly offices employing individuals. Although SHACL shapes might reflect this by integrating the delegation pattern directly, this would only solve the problem partly. Furthermore, use of web tokens may significantly improve performance: after a valid check, a token is issued allowing access for a limited timeframe, thereby omitting cumbersome checks and allowing to access real-time data with less latency. In the longer term, questions need to be answered about establishing chains of trustful assertions with more than 2 or 3 actors: if someone can only prove his right to access a resource by presenting a bundle of nanopublications, which only provide the right pattern if combined, some of those assertions owned by different actors and potentially not public, how does this network of servers negotiate about (delegated) view rights, how does one propagate through a network in which only a 'root trusted authority' is known but not the way how to reach this authority, and how can client and server privacy be balanced so that the amount of relevant information is proportional to the validation time and resources? Lastly, the question remains if shapes

should be based on nanocredential templates or the other way around; although a certain degree of reasoning is feasible, it is to be expected that shapes and nanocredential assertions should not differ too much in content.

5 Conclusion

In this paper, we illustrated a basic method for pattern-based access control scenarios. Although multiple technologies already exist to perform ID- or role-based access control, and for many use cases these technologies satisfy the requirements, pattern- (or context-)based access control has been identified as one of the more powerful ways of regulating access within complex, decentralised (building) projects. Based on existing standards such as RDF and SHACL, the Solid ecosystem and the nanopublication guidelines, a workflow was initiated in which not the identity of the client determines access to certain resources, but rather their properties as confirmed by trusted third parties. To a certain degree, such patterns can already be expressed using the default ACL schemes of Solid, with hard-coded user groups corresponding to certain roles. However, as the intention of this framework is to be extendible beyond AECO implementations, and to allow more complex patterns to be validated without hard-coding WebIDs, we believe this research can provide a basis for future work in this field.

6 Acknowledgements

This research is funded by the Research Foundation Flanders (FWO) in the form of a personal Strategic Basic (SB) Research grant (grant no. 1S99020N).

References

1. Beetz, J., Leeuwen, van, J., Vries, de, B.: An Ontology Web Language Notation of the Industry Foundation Classes. In: Scherer, R., Katranuschkov, P., Sconfke, S.E. (eds.) Proceedings of the 22nd CIB W78 Conference on Information Technology in Construction. pp. 193–198. Technische Universität Dresden (2005)
2. Beetz, J., van Berlo, L., de Laat, R., van den Helm, P.: Bimserver. org–an open source ifc model server. In: Proceedings of the CIP W78 conference. p. 8 (2010)
3. Buyle, R., Taelman, R., Mostaert, K., Joris, G., Mannens, E., Verborgh, R., Berners-Lee, T.: Streamlining governmental processes by putting citizens in control of their personal data. In: International Conference on Electronic Governance and Open Society: Challenges in Eurasia. pp. 346–359. Springer (2019)
4. Groth, P., Gibson, A., Velterop, J.: The anatomy of a nanopublication. Information Services & Use **30**(1-2), 51–56 (2010)
5. Hoang, N.V., Törmä, S.: Drumbeat platform—a web of building data implementation with backlinking. In: eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2016: Proceedings of the 11th European Conference on Product and Process Modelling (ECPPM 2016), Limassol, Cyprus, 7-9 September 2016. p. 155. CRC Press (2017)

6. ISO, B.: 19650-1: 2018. BSI Standards Publication Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM)-Information management using building information modelling (2018)
7. Kirrane, S., Mileo, A., Decker, S.: Access control and the resource description framework: A survey. Semantic Web **8**(2), 311–352 (2017)
8. Kuhn, T.: nanopub-java: A java library for nanopublications. arXiv preprint arXiv:1508.04977 (2015)
9. Kuhn, T., Barbano, P.E., Nagy, M.L., Krauthammer, M.: Broadening the scope of nanopublications. In: Extended Semantic Web Conference. pp. 487–501. Springer (2013)
10. Kuhn, T., Dumontier, M.: Trusty uris: Verifiable, immutable, and permanent digital artifacts for linked data. In: European semantic web conference. pp. 395–410. Springer (2014)
11. Mansour, E., Sambra, A.V., Hawke, S., Zereba, M., Capadisli, S., Ghanem, A., Aboulnaga, A., Berners-Lee, T.: A demonstration of the solid platform for social web applications. In: Proceedings of the 25th International Conference Companion on World Wide Web. pp. 223–226 (2016)
12. Oraskari, J., Törmä, S.: Access control for web of building data: Challenges and directions. In: eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2016: Proceedings of the 11th European Conference on Product and Process Modelling (ECPPM 2016), Limassol, Cyprus, 7-9 September 2016. p. 45. CRC Press (2016)
13. Pauwels, P., Terkaj, W.: EXPRESS to OWL for Construction Industry: Towards a Recommendable and Usable ifcOWL Ontology. Automation in Construction **63**, 100—-133 (2016). https://doi.org/10.1016/j.autcon.2015.12.003
14. Pauwels, P., Zhang, S., Lee, Y.C.: Semantic Web Technologies in AEC industry: A Literature Overview. Automation in Construction **73**, 145—-165 (2017). https://doi.org/10.1016/j.autcon.2016.10.003
15. Rasmussen, M., Lefrançois, M., Bonduel, M., Hviid, C.A., Karlshøj, J.: Opm: An ontology for describing properties that evolve over time. In: 6th Linked Data in Architecture and Construction Workshop (LDAC), CEUR Workshop Proceedings. vol. 2159, pp. 24–33 (2018), http://ceur-ws.org/Vol-2159/03paper.pdf
16. Rasmussen, M.H., Pauwels, P., Lefrançois, M., Schneider, G.F., Hviid, C.A., Karlshøj, J.: Recent changes in the Building Topology Ontology. In: 5th Linked Data in Architecture and Construction Workshop (LDAC). Dijon, France (2017), https://hal-emse.ccsd.cnrs.fr/emse-01638305
17. Sambra, A.V., Mansour, E., Hawke, S., Zereba, M., Greco, N., Ghanem, A., Zagidulin, D., Aboulnaga, A., Berners-Lee, T.: Solid: a platform for decentralized social applications based on linked data. Tech. rep., Technical report, MIT CSAIL & Qatar Computing Research Institute (2016)
18. Wagner, A., Rüppel, U.: BPO: the building product ontology for assembled products. In: 7th Linked Data in Architecture and Construction Workshop (LDAC), CEUR Workshop Proceedings. pp. 106–119 (2019), http://ceur-ws.org/Vol-2389/08paper.pdf
19. Werbrouck, J., Pauwels, P., Beetz, J., van Berlo, L.: Towards a decentralised common data environment using linked building data and the solid ecosystem. In: 36th CIB W78 2019 Conference. pp. 113–123 (2019), https://biblio.ugent.be/publication/8633673