

A Model-Theoretic Approach to Digital Identity

Luca Boldrin¹

¹InfoCert, Piazza da Porto 3, Padova, Italy

Abstract

Verifiable credentials, the mainstream model for digital identity today, rely largely on the Resource Description Framework (RDF). We argue that while RDF is oriented to ‘entities’, the world we are willing to represent is essentially based on ‘attributes’, creating a representation gap. We sketch a formal language and a set-theoretic semantics to illustrate this concept. We eventually show what implications this perspective might bring to our practical implementations of identity frameworks.

Keywords

Digital identity, set theoretic model, verifiable credentials, RDF

1. Introduction

We are used to thinking of a ‘digital identity’ as a set of attributes associated with an individual. Individuals can voluntarily disclose some of these attributes to third parties who need to know them to participate in some transactions. The tricky point is how to make these attributes “verifiable”, so that the relying party can conveniently be convinced of their truth. The core identity problem, which dates back to the pre-digital world, is *how to associate a human being with a set of attributes in a verifiable way*. The problem is even trickier in the digital realm. Verifiable Credentials (VC) data model, as defined in [1], has been widely adopted as a solution to this problem. It is at the core of many current activities in digital identity, including the European Union Digital Identity Wallet initiative [2].

In Section 2 we discuss the Verifiable Credentials data model to show that, at its core, it is entity-oriented. In Section 3 we argue that the real problem we face is not about representing entities, but about connecting attributes, and we introduce a taxonomy of attributes. Then, in Section 4 we introduce a simple language and a set-theoretic model to demonstrate this concept, and discuss how it can be extended, comparing with related formalizations. Eventually, in Section 5 we draw some practical conclusions. We would like to stress that, in the simplified perspective we are offering, many relevant aspects of real-world identity systems (including non-disclosure, unlinkability, trust frameworks, etc.) are wittingly neglected.

2. The Verifiable Credentials data model

The VC data model is based on the Resource Description Framework (RDF) [3], as summarized in Figure 1. According to the VC data model, “A verifiable credential contains claims about one or more subjects” and a claim is “an assertion made about a subject”. This information is included in the `credentialSubject` property: “The value of the `credentialSubject` property is a set of objects where each object MUST be the subject of one or more claims, which MUST be serialized inside the `credentialSubject` property. Each object MAY also contain an `id` property to identify the subject [...]”.

While the syntax of a claim is not fully specified, the examples clarify that it is essentially a couple “id” - “attribute” (like in the JSON-LD representation in Figure 1). The data model specifies “the `id` property expresses an identifier that others are expected to use when expressing statements about the specific thing identified by that identifier”. The data model claims that the “`id`” property is optional, which would

TDI 2025: 3rd International Workshop on Trends in Digital Identity, February 3, 2025, Bologna, Italy

✉ luca.boldrin@infocert.it (L. Boldrin)

🆔 0000-0001-8403-1506 (L. Boldrin)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

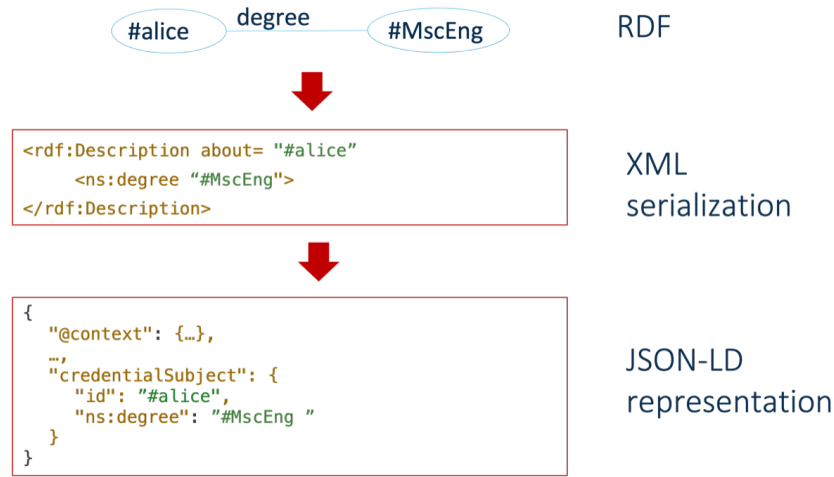


Figure 1: From RFD to JSON-LD representation

amount to having a credentialSubject including just the attribute. It is, however, legitimate to assume that in this case the attribute is predicated about the bearer of the credential, or the key holder.

Returning to our example, leveraging the formal semantics of RFD [4] we can deduce that the intended semantics of the credential is something like: the entity ‘alice’ is associated with the entity ‘MscEng’ via the relation ‘degree’. This approach is entity-centric. In the following, we take a different approach: we take an historical perspective on identity trying to identify what we really want to model.

3. The aspects of reality we want to model

When building a model, we necessarily only aim at capturing some specific aspects of the reality that are relevant to the problem we want to solve. In our case the problem can be stated as: *how can we associate a human being with some attributes in a verifiable way?* The answer is simple in case the attribute can be extracted by the physical presence of the person, e.g., ‘eye color’. It is harder when there is no specific relation between the physical presence and the attribute, e.g., ‘citizenship’.

Historically, the basic solution consisted in the possession of some object – e.g., having an armor likely qualified you as being a knight, owning a seal qualifies you as being a bishop – but possession is obviously a weak verifiable mean, since objects can be copied. In this case, the verifiability lies on the complexity and cost of replicating the object (a sort of proof-of-work). However, this approach suffers from the possibility that objects are stolen or voluntarily passed to someone else, and participants need to share some background knowledge on what a specific object is intended to represent. We may introduce ‘speaking objects’, i.e. objects which need not be interpreted, since they carry the information they want to convey: birth certificates (credited to have been introduced by Roman emperor Augustus as wooden diptych with waxed surfaces in 4 AD) provide such an example. Possession of the certificate proves that the bearer holds the attributes annotated in the certificate itself. This example provides an interesting insight, since we can split the problem into two parts: **1) association of the person with some item** (in this case, the the wooden tables); **2) association of the item with the attributes** (in this case, the birth information engraved in the tables).

We will refer to this intermediate item as the ‘confirmation mean’. In more recent times, this approach evolved using confirmation means for which the association to the person is harder to forge, typically the representation of a biometric feature of the person: a picture of the face can be associated to a person in a reasonably secure way, while the same picture can appear in an identity document, which binds together the picture with the name and additional attributes. In general, the verifiable binding leverages a confirmation mean the person is/have/knows. The same confirmation mean can be used to construct several bindings – e.g., an identity card binds the picture with name/family name, a laissez-passers, a

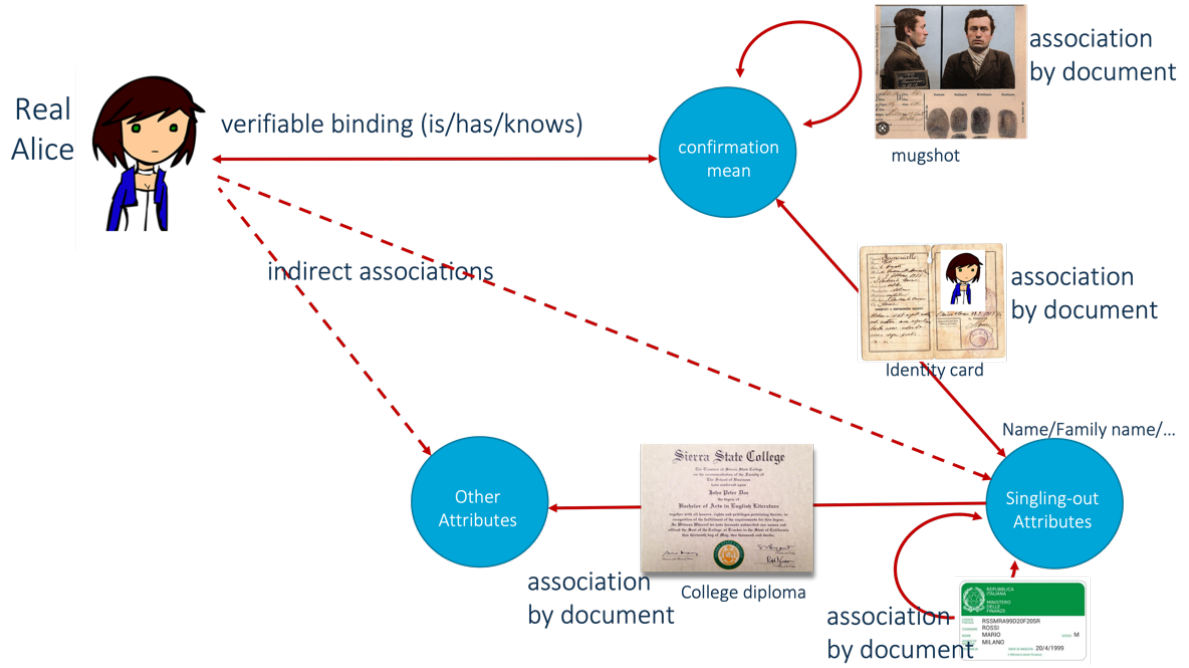


Figure 2: Indirect association to attributes via confirmation mean and documents

driving license, etc. As it happens, attributes have different ‘singling-out power’: a Social Security Number is likely to identify exactly one individual, while a degree only identifies a class of individuals. Singling-out attributes are regularly used for creating chains of bindings to associate a person to their attributes – e.g.: Alice \leftrightarrow Alice’s picture via ‘is’ relation; Alice’s picture \leftrightarrow Alice’s name via an identity card; Alice’s name \leftrightarrow Alice’s degree via a degree certificate (assuming, for the sake of simplicity, that Alice’s name is non-ambiguous). In general, we can assume that documents bind confirmation means to attributes, as well as attributes to attributes. Viceversa, it also makes sense to use more confirmation means to reach the same attribute, e.g., when both a picture and a fingerprint are used to independently establish the name of a person.

In the digital realm, we have the same model. However, we must adopt some **different form of confirmation means**, and a **different form of documents**:

- Typical **confirmation means** in this setting are public keys (possession of the related private key proves the association), user-ids (knowledge of the related password proves the association), biometric specimens (as far as matching with the real person is feasible).
- **Documents**, which in the physical realm gain their security mainly by engraving the two attributes on a physical substrate, may now assume a variety of forms. However, abstractly they consist of an assertion like $\langle attribute1, attribute2 \rangle_{vouched-for-by-T}$ where T is a trusted entity¹. Technically, the assertion may be made available as a signed file, as a record in a database, on a DLT, etc. Crucially, this assertion needs not be handed over by Alice (the subject).

If we think of confirmation means and singling-out attributes as specific types of attribute, and we think of documents as an implementation of the ‘is bound to’ relation, we eventually come to the simple diagram illustrated in Figure 3.

4. A set-theoretic model

Once we focus on the aspects of the reality that we want to model, it is straightforward to construct a formal model. The objective of the formalization is to prove the soundness of the language we want to

¹Note that $\langle a1, a2 \rangle_{vouched-for-by-T}$ is different from $\langle a2, a1 \rangle_{vouched-for-by-T}$.

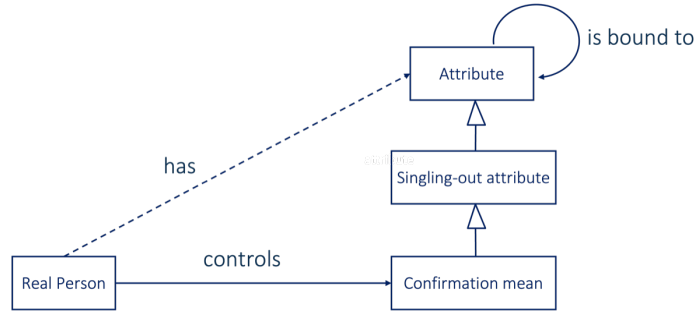


Figure 3: A basic diagram for attributes



Figure 4: Example of a derivation in the propositional calculus

use with respect to the underlying model – or at least to fix how identity sentences should be interpreted. The proposed model has some commonalities with formal approaches to access control like [5], [6], [7], [8], the main difference being that the mentioned approaches focus on the representation of access rights and policies, while we aim at capturing identifiers, attributes and their chaining.

An attribute is a couple $a = \langle tag, value \rangle$ which for syntactic sugar we will type as $a = tag : value$, where ‘tag’ belongs to a space of attribute names and ‘value’ belongs to the space of the respective values. Examples of attributes are $a_1 = name : John$; $a_2 = height : 178$; $a_3 = pub-key : 3f3dhc7css8b2323fe$.

The tag provides the semantics of the attribute, and it may help the verifier to decide whether to treat it as a confirmation mean or an identifier, how to interpret its format, unit, etc. As a matter of fact, there is need for a standardized ontology of tags to establish a shared semantics. A well-formed formula in our language L is a propositional composition of attributes, i.e.:

- an attribute $t_i : v_i$
- a propositional composition of formulas with $\wedge \vee \rightarrow \neg$

We are particularly interested in the subset of these propositional formulas which take the form $t_i : v_i \rightarrow t_j : v_j$. Such a formula represents a claim: $pub-key : 3f3dhc7css8b2323fe \rightarrow degree : MscEng$ is the claim binding attribute ‘ $pub-key : 3f3dhc7css8b2323fe$ ’ to the attribute ‘ $degree : MscEng$ ’. Its intended semantics is: *Whoever can prove to be associated with pub key ‘3f3dhc7css8b2323fe’ can also prove to be associated with the attribute ‘degree : MscEng’*. The inference rule is, as usual, by modus ponens. An inference looks like the one in Figure 4.

A model for this calculus would be: $M = (I, \sigma)$, where:

- $I = \{i_1, \dots, i_r\}$ – intended to represent a set of individuals
- $\sigma: Att \rightarrow (I)$ is a function that maps each atomic term of the language $t_i : v_i$ to an element of $\wp(I)$ (the power set of I)

We extend σ to the entire language $\sigma: L \rightarrow U$

- $\sigma(\neg A) = \sigma(A)^c$ (complement in I)
- $\sigma(A \wedge B) = \sigma(A) \cap \sigma(B)$
- $\sigma(A \vee B) = \sigma(A) \cup \sigma(B)$
- $\sigma(A \rightarrow B) = \sigma(A)^c \cup \sigma(B)$

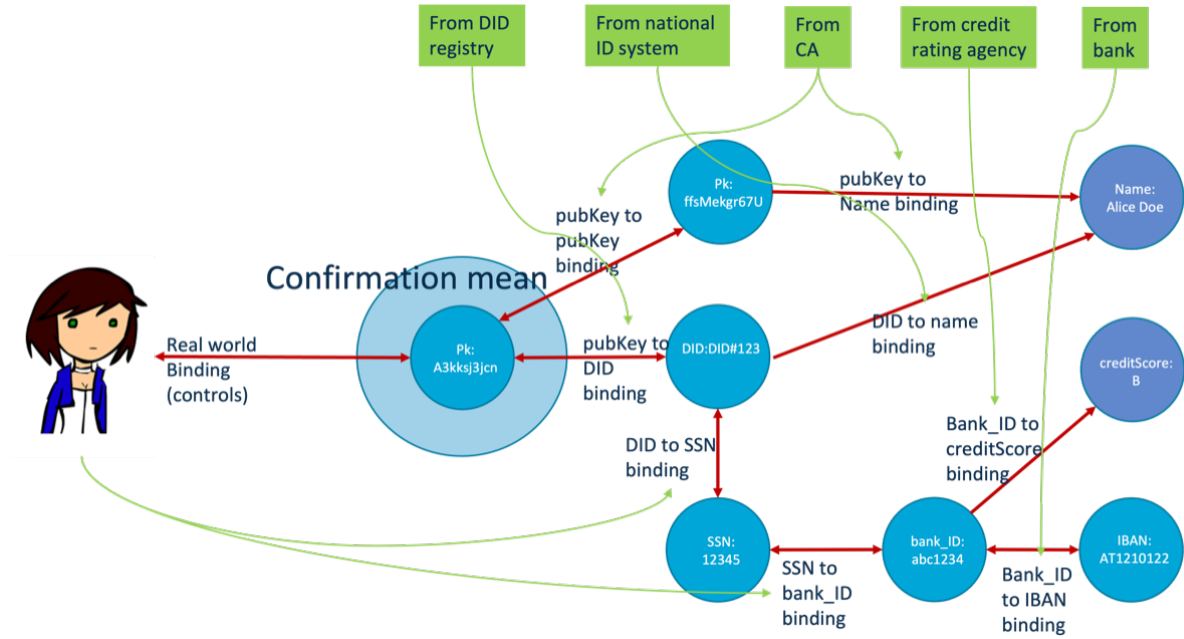


Figure 5: Example of a multi-step inference scenario

We define $M \models A$ iff $\sigma(A) = I$. Soundness and completeness are derived from propositional logic.

An interesting extension – which we do not explore here – would be construction of a more elaborate calculus with an epistemic flavor, which also accounts for trust relations. Well-formed formulas of this language would be something like:

- an attribute $t_i : v_i$
- a claim $c(a_1, a_2, a_3)$ – meaning: entity described by a_1 (likely, a singling out attribute) claims that whichever entity is associated to a_2 is also associated to a_3 . A possible example is $c(id : universityOfPadova, pubKey : 3f3dhc7css8b2323fe, degree : MscEng)$
- a trust relation $t(a_1, a_2)$ – meaning: entity described by a_1 (likely, a singling out attribute) trusts entity described by a_2 (likely, a singling out attribute). A possible example is $t(pubKey : 3f3dhc7css8b2323fe, id : universityOfPadova)$
- a propositional composition of formulas with $\wedge \vee \rightarrow \neg$

The calculus should be constructed in such a way that we can infer formulas like

$$t(luca, unipd) \wedge t(luca, CA1) \wedge c(CA1, marco, DID1) \wedge c(unipd, DID1, degreeMSc) \rightarrow c(luca, marco, degreeMSc)$$

5. Practical conclusions

Practically, to verify Alice’s attributes, a relying party needs to perform these steps:

- get one or more confirmation means (a picture from a scanner, a public key provided by Alice...)
- verify the binding between Alice and a confirmation mean(s) – (via: is/has/knows)
- get a set of bindings of which at least one starts from a confirmation mean (from any source)
- verify each binding using the respective validation information
- follow the chain of bindings starting from a confirmation mean to the desired attributes.

We insist on the fact that the bindings need not come from Alice. The source of the bindings is irrelevant, as long as they are verifiable. We also note that this approach also suits the case where

the verifier is interested in performing some inference about some subject outside of an interactive session. In this case, there is no need to get and verify the confirmation means, skipping the first two steps. An additional observation is that we only leverage ‘atomic credentials’, hence there is no need for selective disclosure. Further investigations would be needed to deal with many other important aspects (including unlinkability).

References

- [1] M. Sporny, D. Longley, D. Chadwick, I. Herman, Verifiable Credentials Data Model v2.0, W3C Candidate Recommendation Draft, W3C, 2025. URL: <https://www.w3.org/TR/vc-data-model-2.0/>.
- [2] European Commission, A digital ID and personal digital wallet for EU citizens, residents and businesses, 2025. URL: <https://ec.europa.eu/digital-building-blocks/sites/display/EUDIGITALIDENTITYWALLET/EU+Digital+Identity+Wallet+Home>.
- [3] R. Cyganiak, D. Wood, M. Lanthaler, RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation, W3C, 2014. URL: <https://www.w3.org/TR/rdf11-concepts/>.
- [4] P. Hayes, RDF Semantics, W3C Recommendation, W3C, 2004. URL: <https://www.w3.org/TR/rdf-mt/>.
- [5] M. Abadi, M. Burrows, B. Lampson, G. Plotkin, A calculus for access control in distributed systems, ACM Trans. Program. Lang. Syst. 15 (1993) 706–734. doi:10.1145/155183.155225.
- [6] M. Blaze, J. Feigenbaum, J. Lacy, Decentralized trust management, in: Proceedings 1996 IEEE Symposium on Security and Privacy, 1996, pp. 164–173. doi:10.1109/SECPRI.1996.502679.
- [7] Y. Gurevich, I. Neeman, DKAL: Distributed-Knowledge Authorization Language, in: 2008 21st IEEE Computer Security Foundations Symposium, 2008, pp. 149–162. doi:10.1109/CSF.2008.8.
- [8] M. Y. Becker, C. Fournet, A. D. Gordon, SecPAL: Design and semantics of a decentralized authorization language, Journal of Computer Security 18 (2010) 619–665. doi:10.3233/JCS-2009-0364.