Refining the Design Principles of Identity Relationship Management

Version: 2.0f
Document Date: 2017-05-26

Editors: Sal D'Agostino, Ian Glazer
Contributors: https://kantarainitiative.org/confluence/display/irm/Participant+roster

Status: WG Report

Abstract:

Relationships are not new but their representation has rarely been first class citizen in the realm of digital identity. In the past, relationships have been represented as attributes such as memberOf or implied in identifiers such as distinguished name. IRM addresses the current state of identity management and the need to promote relationships, in both representations and awareness, in order to provide identity management practitioners with more accurate, more manageable, and more deployable means of managing digital identities in our hyper-connected world. To that end, the IRM WG offers the following design principles of relationships.

Contents

1. INTRODUCTION

The practice of identity management does not take place in a vacuum; people and connected devices rarely, if ever, act completely on their own, devoid of any organizational, personal, and situational context. This has been known to identity management practitioners for many years, and this is why identity management progressed from managing individual digital identities to managing groups to managing roles. Each progression gave identity management practitioners great ability to manage larger populations of users. However, the industry's last progression to role management did not provide enough managerial leverage to adequately tackle such issues as self-sovereign identity, connected devices, and the Internet of Things. Furthermore, in order to more accurately portray the richness of the use cases facing the identity management industry, something more is needed beyond managing identities living in the vacuum of a directory.

In 2014, the identity management industry began to discuss the notion of Identity Relationship Management (IRM) and how relationships could provide the richness needed to represent our hyper-connected world and give administrative leverage to identity management professionals. Relationships are not a one-time thing; they are dynamic and driven by the context of the access control decision that needs action.

The Kantara Initiative formed the IRM Working Group which in turn produced its "Laws of Relationship Management" report in early 2015. Subsequently, the IRM WG examined the original design principles of IRM and sought out examples of IRM currently implemented. This document is the summation of that effort and is intended for people designing complex identity systems and interactions. This report does not offer prescriptive design patterns for large-scale relationship-oriented systems, but instead offers design principles for consideration and real-world use cases for identity professionals to study. In some cases these concepts overlap and in some cases they may be nested inside other concepts. This ambiguity is something to get comfortable with as we move forward and understand that they ambiguity can only be removed in context.

2. REFINED DESIGN PRINCIPLES OF RELATIONSHIPS

Relationships are not new but their representation has rarely been first class citizen in the realm of digital identity. In the past, relationships have been represented as attributes such as memberOf or implied in identifiers such as distinguished name. IRM addresses the current state of identity management and the need to promote relationships, in both representations and awareness, in order to provide identity management practitioners with more accurate, more manageable, and more deployable means of managing digital identities in our hyper-connected world. To that end, the IRM WG offers the following design principles of relationships.

It is important to note that in application these principles are not discrete. One cannot design a system that provides one principle without at least considering the other principles as well. That is not to say that each principle will be of equal important to a system that you design to meet your specific use case, but that as a designer you will at a minimum have to examine each principle in order to deliver a system that handles relationships well.

Furthermore, the systems you design, the principle you consider do not act in a vacuum. The context in which principles are applied has great sway over their practical application. In the previous version of the Design Principles of Relationships, Context was a 1st order Principle. However, upon further consideration, the IRM WG recognized the pervasive nature of context and attempted to reflect it in all of the following principles.

1.1. PROVABLE

The existence of a relationship between actors (be those individuals, groups, organizations, non-human entities, or any combination of these) carries meaning and provides large context. As such, systems handling relationships need a way to state authoritatively that a given relationship exists.

Care must be taken to ensure that the existence of a relationship is only provided to the right parties for the right purposes. For example, if a previously unknown actor asks a doctor’s office, “Is Alice a patient here?” and in doing so is asking for prove that a relationship exists between Alice and the doctor’s office, then the doctor’s office should vet the unknown actor before providing an answer. It may be that the unknown actor is a hospital system in another country where Alice is traveling and she requires medical assistance, and the actor’s request is appropriate and valid. It may be that the unknown actor is a gossip columnist and is looking for salacious morsels to report about Alice.

Keep in mind that in some cases, the actor asking for the existence of the relationship may be one of the parties in the relationship. For example, Bob could ask his employer for the existence of their relationship so Bob can present it to a bank to get a loan. Similarly, Eve may ask a credit bureau for proof of a relationship as her first step to correcting inaccurate information.

Responding to a request of relationship existence requires that the party making the request and the purpose of the request are appropriate as well as that members of the relationship have either explicitly or implicitly agreed that other parties can receive proof of relationship information.

There are several implications of the Provable design principle:

- Proof of Relationship information can be sensitive (e.g. the very fact a relationship exists between parties carries meaning) and thus care must be taken to not inadvertently distribute this information to the wrong or inappropriate parties.
- In the digital realm, a Proof of Relationship token may be needed. Consumers of such a token would require standard ways to validate that the relationship was still valid
- Generating Proof of Relationship information might require all parties in the relationship to interact in concert. For example, each member of the relationship has taken an action in order to allow the release of Proof of Relationship information.
- Concepts such as User-Managed Access Control and Consent Receipt may be a portion of what is needed to implement a Proof of Relationship service

1.2. CONSTRAINABLE

Although a relationship exists, parties involved may want to impose constraints on the relationship. These constraints may describe acceptable behaviors of the actors in the relationship, approved use of data by the parties, and the terms under which the relationship is terminated. And in this way, the "Constrainable" design principle feels familiar to our everyday lives in the analogue world.

But in that familiar is a bit of a trap. One cannot assume that all of the actors in a relationship are capable of:

- Asserting their desired constraints
- Acknowledging constraints
- Enforcing constraints
- Being held accountable for failure to uphold a constraint

For many of the principles there exists the aspect of context. Although an actor has put a constraint in place that constraint may not always be enabled or relevant. Based on context a constraint may be enabled or become relevant. In this way, the older design principle of "Contextual" because an aspect of this and other design principles. Contextual triggers turn on and off constraints based on the desires of the actors and potentially enforced by relationship managers or the actors themselves.

AN EXAMPLE OF CONSTRAINABLE

Consider a "smart" lightbulb. The owner may want to constrain what data the light bulb sends to its associated IoT platform, but the bulb does not provide such an affordance. In this case, the owner's only recourse (other than not entering into a relationship e.g. not using the lightbulb) is to look for something else to enforce her desired constraints such as the IoT platform to which the light bulb sends messages. Acting in this capacity the IoT platform takes on the role of a relationship manager - an actor which is aware of the context of a relationship and can act upon the relationships and parties in the relationship.

1.3. MUTABLE

Relationships, like most things in the digital identity world, change over time. Different parties enter and exit a relationship. Attributes of those parties change over time. And at the same time the properties of the relationship itself can change as well. Thus designs for systems that handle relationships must account for mutability.

Mutability introduces change and dynamic considerations for actors and attributes. Although not every relationship will change in the same way and although not every attribute for every actor will change, designers must at least explore what things can change, how often they will change, and what would be the impact if they did change. Furthermore, designers should consider mutability at three levels:

- The relationship as a whole including all of the actors, constraints, attributes and properties.
- The connections between parties and the associated attributes and properties of those connections
- The actors and their associated attributes

Some aspects of a relationship may actually be immutable. For example, a connected device may be immutably stamped that it was built by Company Q, but the connection between Alice and her light bulbs may only last as long as Alice owns her apartment.

If change is inevitable, a fair question to ask is, "What manages changes to relationships, actors, constraints, attributes, properties and context?" Although individual actors may manage their self-asserted attributes, the IRM Working Group felt the need for a "higher level" manager, one who could enforce mutability across an entire relationship graph and delegate authority as necessary. As with "Constrainable," the notion of a relationship manager appears.

1.4. REVOCABLE

Relationships end. This is true in the digital world as it is in the analog one. To "I am no longer in this relationship" may have a clear and distinct meaning to one party but not the other parties in the relationship. When discussing this design principle, the IRM Working Group thought of it as equivalent to terminating a relationship and it quickly realized implementing relationship revocation was not as simple as just disconnecting the parties in the relationship. Questions arose about who can revoke a relationship, how is that revocation enforced, how is the historical information about the relationship preserved, and what is the interplay of mutability and revocability.

How the revocation of a relationship works, what is required to revoke a relationship, and the process by which a party requests to revoke a relationship all differ based on context. Different industries and jurisdictions have their own interpretation of this design principle.

For example, what it means to revoke Bob's relationship with his smart light bulb is quite different from revoking Bob's relationship with the country of his birth.

And, it is important to note that not all relationships can end; irrevocable relationships exist. A light bulb is only built once and thus its relationship to its manufacturer is irrevocable. But surrounding the "manufactured by" relationship is a larger context. For example, the light bulb may have been built by Westinghouse which in turn was purchased by GE. The bulb's relationship with its manufacturer did not change and was not revoked but the relationship of the manufacturer to the larger world certainly did change.

Guidance for designing systems that handle the revocation of relationships includes:

- Consider legal and business requirements on the termination and revocation of a relationship.

AN EXAMPLE OF REVOCABLE

Consider the revocation requirement in the use of personal health information. There is a need to share information with a wide range of individuals, devices, location each of which issues their own shared authorizations. How might revocation and validation work in this complex environment? It must be conditional to deal with complexity, not a single binary status of an entity and/or its scopes. The breadth of tokens; PKI, OAuth, UMA, adaptive authentication and distributed ledgers are examples of revocation and can comprise aspects of relationship validation and as a result revocation conditions.

- Coordinate data retention requirements with relationship revocation. For legal reasons, an organization may need to retain, long after the relationship end, proof of relationship as well as materials used to form the relationship and data produced from the relationship.
- Design a process for a party to request to revoke a relationship. (Design a process for reinstating the relationship too.)
- Clarify how revoking a relationship is different from changing attributes or properties of the relationship or the parties in the relationship.
- Consider whether in the reader's use case revocation is actually adding a broad constraint to the relationship.
- Given jurisdictional or business requirements, design the system such that revoking a relationship does not impede providing proof a revoked relationship existed in the past and to allow third parties to validate that a revocation happened.

Given the influence of context on this design principle, the IRM Working Group did not delve into the specific mechanics of revocation. It is likely that the orchestration of business process, retention of records, etc are left to the notional "relationship manager" to sort out.

1.5. DELEGABLE

Relationships change. Relationships end. The actors in relationships can be replaced as well, so in some cases there is an actual transfer of the relationship. In order to represent and handle situations in which the actors in a relationship change, either permanently or temporarily, relationship-based systems need to accommodate the design principle of delegation. There are three areas of consideration for delegation and relationships:

- **Scope**: A party may choose to give another party all of its original capabilities and rights with regards to a relationship; in this case the scope of delegation is "full."
- **Permanence**: The original party may be able to put a time limit on the delegation, stating that the new party has delegated participation in the relationship for 60 days, 100 hundred years, or it may be a permanent delegation.
- **Constraint**: The original party may choose to impose no new constraints on the relationship meaning that the new party can do as they please in the relationship.

Notice the use of "may" in above list. The IRM WG found it difficult to assert that actors in relationships would always have the ability to delegate their participation in a relationship. Furthermore, if a party can delegate their participation it is unclear that the party can always delegate the entire relationship for an indefinite amount of time without constraints. Depending on the context (including the legal context in which the relationship exists) actors can delegate differently. In some cases, the Reader can foresee that the other parties in a relationship may have a say in whether an actor can delegate their participation. Sorting out who can delegate, how much, and for how long is likely the job of a context-aware relationship manager.

EXAMPLES DELEGABLE

Alice may choose to delegate her participation in a relationship to Bob completely with no time limit and no constraints. Going forward Bob is linked to all of the other actors Alice was in the relationship and is subject to all of the existing constraints that Alice was subject to. An interesting question to ask is, is Bob entitlement to all of the historical data generated by Alice in the context of the relationship?

Bob delegates his participation in a relationship to Eve for 6 months and also creates a constraint that Eve is only allowed to observe data flowing in the relationship but not allowed to create new data. Meanwhile, Bob also delegates his participation in the same relationship to Alice for 30 days, in which she is granted full rights, except she cannot access historical relationship data and she cannot further delegate participation. This example begins to highlight the challenge of the delegable design principle in determining what is actually delegated: the party's connection to the relationship or the party's

1.6. SCALABLE

Scalability is a must for identity relationship management. Originally, the IRM WG identified four axes of scalability: actors, attributes/properties, relationships, and administration, and these four variables of scalability still need to be solved for in order to have relationship management. But there is another crucial consideration for this design principle - every party in a relationship may be legion. IRM is not only meant for simply just single party to single party relationships, but also groups of actors in relationships with other groups of actors. As one member of the IRM WG stated, “this world is many to many on all sides of the equation.”

One way to think of the many to many nature of relationships is take a page from the Eames’ “Powers of Ten.” Observing a relationship graph at an actor-level, one would see each individual actors connected to one another. Zooming out, one would see the interconnected organizations with which the actors are associated. Zooming out again, one would see how the relationship graphs themselves link to other relationship graphs. Zooming in, one would see attributes and properties: of the connections between actors and of the actors themselves. The practice of zooming into and out from a relationship can help the reader then recognize some of the challenges related to other design principles. At a certain “scale,” delegation becomes an organizational policy while at a smaller scale an individual actor may be unable to delegate their portion of a relationship. At a certain scale, an actor may be allowed to change relationship properties but at another those properties are no longer mutable.

3. THE ONWARD JOURNEY

The IRM WG has explored principles and applications of relationships. In the course of its exploration, two things have become apparent. First, relationship systems often need some sort of manager to enforce policy and orchestrate actions between actors and the algorithms and tools to support this. Second, in order to operationalize relationship systems, a means of more efficiently describing relationships, their actors, attributes and properties is required. Both are potentially rich areas of further activity.

A Need for New Tools and Algorithms

The IRM Working Group also identified that there needs to be a different way to represent and to manage the identity relationships. These relationships are not a one-time thing; they are dynamic and driven by the context of the access control decision that needs action. We need a new set of access control algorithms which can deal with complexity and which embody the language of relationships put forward in this document. This is an initial attempt at providing a language to discuss the topic. In some cases these concepts overlap and in some cases they may be nested inside other concepts. This ambiguity is something to get comfortable with as we move forward and understand that they ambiguity can only be removed in context. And this is further evidence of the need to continue the work on the algorithms that define the management of identity relationships.

1.7. RELATIONSHIP MANAGER

As the IRM WG worked and re-worked these design principles, the group realized that it is not possible for the actors in a relationship to enforce all of the conditions of a relationship. For example, in the case of delegation, an actor who permanently delegates her participation in a relationship severs her ties to the relationship and thus is in no position to enforce anything about that relationship. But if she is not in a position to enforce the notional rules of a relationship then what is?

In order to facilitate the interactions of actors in relationships as well as to ensure that constraints and other conditions of relationships are consistently applied, what is needed is a relationship manager. This manager orchestrates interactions amongst the actors, blocks actions counter to the constraints of the relationships, manages relationship revocation, and enforces the rules of a relationship system. One can think about a relationship manager like a policy decision point for relationships: the relationship manager can “read” relationships, is aware of context, and makes decision as to whether actions related to the relationships are allowed.

Decisions regarding relationships are not centralized. In the world, we rely on a variety of 3rd parties to broker our interactions such as lawyers, the State, financial systems, and other people. In the real world, we do not need, nor would it be practicable, to go to a central office in order to conduct any relationship-based interaction. Similarly, in the digital world, a single centralized relationship manager is completely unworkable. When parties in a relationship need to interact, they have to use a “nearby” relationship manager without having to find a central, singular authority. But in order for this to work, digital relationship managers require a standardized format for representing relationships so that any relationship manager, if called upon, can work on any relationship.

1.8. RELATIONSHIP NOTATION LANGUAGE

The emerging requirement for relationship managers to operate on any relationship strongly implies a standardized method of representing relationships. Even before considering the problem of machine readable relationships, the Working Group quickly saw the need for more efficient representation of relationships; it found that describing relationships in full English sentences became cumbersome quickly.

Although the Working Group did not pursue potential representation formats in depth there was some discuss of enlisting set-builder notation. While the Group members were not necessarily keen to revisit their days in discrete mathematics class, they did acknowledge that set-builder notation might make it easier to talk about relationships. But something else is needed for computer-readable relationship representations. This is an open area of study and this report's editor suggests that such a format should lend itself well to both the RESTful web and graph databases.

4. CONCLUSIONS

Given the relevance of the work and the fertile ground for further effort, in order to best progress our findings collaboration across the other Kantara work must take place next. For example personal information with consent policy enforced by a relationship managed and controlled by the user for people and things spans multiple if not all of current Kantara efforts. Given this we conclude as a working group to take the effort in further exploring the relationship language across Kantara efforts. And as working group bring this to the Leadership Council to decide next steps.

5. REFERENCES

Kantara Initiative Identity Relationship Management Working Group. *The Design Principles of Relationship Management*. v1.0. Kantara Initiative, Inc., 25 Feb. 2015. Web. <https://kantarainitiative.org/file-downloads/kantara-irm-design-principles-of-relationship-final-report-v1/>.

6. REVISION HISTORY

May 2017 – Final Draft Version 2f

7. APPENDIX A: EVOLUTION OF THE DESIGN PRINCIPLES

WHAT'S DIFFERENT?

As the IRM Working group met and discussed relationships both in the abstract and in the real-world, its understanding of the design principles for IRM systems evolved. What follows is a summary of the major changes between the original version of the Principles of Identity Relationship Management and this report.

- **Acknowledgeable folds into Provable**: It became clear during the IRM WG's work that the original principle of Acknowledgeable was a special case of Provable. Acknowledgeable strongly hinted at the need for some form of token that could serve as evidence that actors were aware of their relationship and that need became the "Proof of Relationship" described in the section on Provable.
- **Transferable becomes Delegable**: When discussing transferability of relationships, the Working Group decided that these were examples of delegation.
- **Immutable becomes mutable**: Originally, the IRM WG presented the principle of Immutable but quickly realized that mutability as a whole was a larger, more important topic. Things are more likely to change than they are to stay the same. To reflect this, the WG changed the Immutable Principle into the Mutable Principle.
- **Contextual is everywhere**: Originally the Contextual Principle was a standalone principle. But, as the IRM WG realized, none of these Principles stand by themselves; their interrelations give the concept of identity relationship management its strength. Although the WG first tried to describe Contextual as a subset of Constrainable, it realized that was not accurate either. The WG settled on the idea that context is the substrate on which all of the principles float.
- **Actionable dissolves into the world of the relationship manager**: Originally, the WG described the Actionable Principle, in which conditions caused aspects of a relationship to become relevant or to be acted upon. However, that isn't an attribute of a relationship but instead the ability of a relationship manager: the ability to orchestrate actions on relationships and between actors within relationships.