Attribute Considerations & Designations In The Consent-informed Attribute Release (CAR) System

(aka “Attribute Taxonomy”)

Architects: Marlena Erdos & Rob Carter
Author: Marlena Erdos
Other contributors: Ken Klingenstein, Mary McKee

Table of Contents

Introduction: Attribute Designations & Meta Info

The CAR team has explored what categories of attributes and values we have in the CAR system (or really, in a higher education institution) so that we can plan and design for a configuration tool that allows the “super admin” to enter the needed “meta info” for attributes (attrs) as part of CAR deployment.

This meta info is crucial to:

- CAR’s displays (UIs) of attrs and values to end users and admins
- other CAR handling such as “always send/never display) (discussed below)

Before we go further in this document -- which is mostly about categories (or “designations”) of attributes and values -- let’s consider meta info.

Meta Info Overview

An example of “meta info” is a **display name** (sometimes called a “friendly name”) for an attribute. Another example is a **description** for the attribute (or a value). While these two examples of meta info are information that directly helps an end user make a “consent” decision, there are other types of meta-info that we need to store and handle. For example, CAR needs to know whether an attribute contains only one value at a time (“single-slotted”) or can have multiple values simultaneously (“multi-slotted”). This info is essential for UI handling.

We’ll go into detail about the meta info CAR needs to store after we discuss all of the attribute categories. You can find that meta info discussion here.

More About This Document

This document discusses the CAR team's findings, thoughts, and plans with respect to attribute categories, meta info, and our “super-admin” config tool. It started out as an internal CAR document, but now is serving the additional purpose of explanation to others. That’s not the easiest balancing act -- but we try :-).

The rest of the doc is organized by the categories we’ve found thus far. For each category, we note whether there is a **configuration tool implication** and whether there is a **UI implication**.

Note: In the discussion below, we use the term “attribute” and “information item **(II**)” interchangeably. An “II” is simply a generalized version of “attribute.” The CAR system can handle resources that aren’t attributes. In fact, CAR is agnostic about what it sorts of things it is holding “permit/deny” decisions on. It can as easily hold a decision for “view my photos of friends” as it can for “release my email address.

Our first category -- or designation -- relates to the display of values in an II.

Presentation Designation

Config Tool Relevant: Yes.
UI Implication: Yes.

The Presentation Designation (PRED) of an II governs whether the UI will show the raw value (or values) of an II to the user, or whether it will seek to retrieve[1] a display name and description for each value, showing these instead.

We need the super-admin to designate for each II whether its values are transparent, encoded or opaque. Those are the only choices.

We describe these choices here:

- **Transparent**: The raw value is immediately interpretable by an end user. Example: street address, phone number, etc.

- **Encoded**: The raw value is hard for the end user to understand “as is.” Consider a “marital status” attribute. “Single” may be coded as “1”, “Married” as “2” etc. The user needs to see a friendly name and/or description of the value to understand it. This means that we need the super-admin to supply a display name and description for each encoding. (And also provide internalizations of course.)

- **Opaque**: The raw value is (typically) meaningful to the RP, but not to the user. Values consisting of encoded binary data usually should be designated “opaque.” (Example: VGhlIHRpbWU…..) A value that is itself a multi-part item including (say) a digital signature or a timestamp also typically should be designated opaque.

[1] The UI calls a CAR API to seek a display name and description.

How CAR Uses The Presentation Designation

The UI subsystem seeks to retrieve a display name and description for the values **only if** the II has a PRED of “encoded.”

> Implementation note: Currently retrieval is by an **exact match** of the incoming raw value to the raw value that’s the key for the display name and description. Upshot: If the value has a varying component -- for example, a timestamp as part of the value -- the CAR implementation doesn’t have a way to retrieve a display name and description (because “exact match” necessarily fails).

The UI presents the matched display name and description instead of the encoded raw value. (We may allow the user to see the raw value via hover or others means.)

If there is no display name for the value, the UI will show the raw value.

For IIs which have PD’s of transparent or opaque, the UI shows the raw value it has in hand.

Single-Slotted vs Multi-Slotted IIs

The CAR UI needs to know if there's more than one "slot" in an II for values so that it can properly display the values to the user. For example, eduPersonAffliation has multiple slots (faculty, student, staff, etc).

We say "slot" (and “single-slotted” and “multi-slotted” instead of “multi-valued" because even an II with a single slot could have multiple values in that slot (e.g. marital status).

Note that to properly inform the user, CAR needs to have descriptions for all encoded values, whether or not there’s a single slot or multiple slots.

Note that “single-slotted” vs “multi-slotted” matters both for encoded values and transparent values. It doesn’t matter for opaque values XXX.

Dealing With Long Values

Some IIs have values that can be very long. This matters in particular for an II whose PRED is transparent or opaque. Presenting the entire long value to the user would take up a lot of screen real estate in the UI. And showing the full long value doesn’t actually help the user make a more informed policy decision.

Because of this, the CAR UI will show the beginning characters and final characters of the value with an indicator (likely ellipses "...") in-between.

Policy Designation

Config Tool Relevant: Yes.
UI Implication: Yes, for end users but not ARPSI/CARMA policy admins

In CAR, we allow admins to create policy (at the ARPSI and CARMA) about individual values of every attribute. We have more to say on this below.

But, for end users it doesn't always make sense for CAR to ask for a decision on every value.

For example, it doesn't make sense to force a policy decision for each value of an opaque II.

And it's not necessarily helpful to allow a separate policy decision about each line of a street address (where each line is a "value").

Because of this, we ask the super-admin to designate each II as either:

- Policy on Attribute Overall (PAO)
- Policy on Every Value (PEV)

This designation affects both how CAR will construct the UI for end users, and how it will store end user policies at the COPSU.

What Should Be PEV? Which Should Be PAO?

Here are guidelines for which IIs are PEV and which are PAO:

PEV:

- multi-slotted IIs where each value may have a different sensitivity (e.g. eduPersonAffiliation, isMemberOf)
- single-slotted IIs where the user may feel differently about release of some values than others (e.g. HIV status),

PAO:

- most single-slotted IIs (e.g. Primary_Department, official name, etc)
- IIs with opaque values
- IIs where the separate values truly make up one overall items (e.g. lines in a street address)

“All PEV, All The Time” For Policy Admins

As we mentioned, In CAR, we allow admins to create policy at the ARPSI and CARMA about individual values of every attribute. So, the PAO/PEV distinction doesn’t matter. All IIs are considered PEV; the admin can make distinct policy choices on every value of every II.

This may seem a little bit odd: How could an admin make distinct policy choices if the values of the II are “opaque” (as stated by the II’s Presentation Designation), and hence there’s no display name and no description for the values?

An example will clarify. Duke University’s Proxy Token attribute is multi-slotted (typically), where each value is itself a signed token, representing delegated rights from the user. (It’s akin to an OAUTH grant token.)

The signed token (remember, a **value** of the overall Duke Proxy Token) includes a service name and a timestamp, among other elements. So some elements of the token are stable and parseable by the policy admin (though others vary and are not parseable).

Here are two example values:

> 432143299**|** ActiveDirectory**|** Pv$bHgE3….
>
> 432121057**|** LDAP**|** **|**Pv$bHgE3….

The policy admin can use the regular expression matching capability of the ARPSI and CARMA policy languages to match on the service name portion of the value (e.g. LDAP).

In particular, the policy admin can knock out values (and hence privileges) that aren’t relevant for the RP (or RPs) designated in the policy.

Or course, the admin must know what these service names within the value are, and what they mean. The CAR system doesn’t have a way to provide a display name and description for anything but the Duke Proxy Token attribute overall.

More About Regular Expression Matching

The policy-matching engines at the ARPSI and CARMA match an incoming raw value in a decision query to a policy based on **regular expression** matching. That is, the policy can contain a regular expression. For example, a policy that said “red?*: permit” for a “favorite color” attribute would match incoming raw values “reddish_brown, “red_orange” etc.

Upshot: even if most of a raw value for an attribute varies -- say, a timestamp component -- if there's part of the value that's always present, the admin can write a policy rule using regular expressions to match the value.

"Always Send; Never Display" Designation (aka "AS/ND")

Config Tool Relevant: Yes.

UI Implication: Yes. These attributes (or values) should not be displayed to the end user in the intercept or self-service UIs. They should however be displayed to admins (though separately from "normal" ARPSI/CARMA policies).

Overview of AS/ND

There are two types of attributes for which the AS/ND designation is appropriate.:

"Deny" authorization attributes and values: For example, persona non-grata" attribute, or specific values of an II such as a "bad actor" group membership; They typically strongly figure into a "deny" authorization result at the RP.

Non-personal attributes: Attributes that are sent to an RP but which aren't specific to the user (e.g. the Duke "Buck id"), and which the institution doesn't want shown to the user.

The discussion below is about the "deny" case:

AS/ND Discussion/Decisions

- A user should never be able to make a choice about release of this type of attribute or value. If they could, they could get access to resources that they are not authorized for.
- The attribute/value **should never get displayed to the end user in any UI.** (Reason: We don't want to clue a nefarious person into how to subvert the system. Both the attribute name, and the attribute value could serve to inform the nefarious person.)
- The attribute/value possibly should be shown to a policy admin, but the "AS/ND" status should only be changeable by a super-admin (in a "config" UI as opposed to a "policy admin" UI.
- The "AS/ND" rule takes precedence over all other (normal) institutional policy rules with respect to AS/ND attributes and values.

Current decisions:

- Allow the super-admin to assign specific IIs and specific values to the AS/ND category in the config tool (and an associated API)
- Show AS/ND IIs and values to policy admins in other UIs[2]
- Implement in code not as a policy (because Rob Carter (implementation lead) says that's the best way to go).

Questions/Issues

- Do we only send the attr if it is asked for? Or "always send"? .

Use Case from Duke: "Never Hire" is an AS/ND value for them.

Note that Duke also has attributes/values that "de-authorize" a user to a particular RP but not all RPs. AS/ND could work for these situations but we need to decide on whether all RPs get the AS/ND items or only those requesting them.

"Special Category-GDPR" Designation

Config Tool Relevant: Yes. We'll have admin designate "Special Category-GDPR" or not for each II.

UI Implication: Yes. Ideally, we'd allow the end user to choose whether to display the values of these attributes (and their decisions) or not. We speculate that a single "boss button" (so to speak) makes more sense than hiding individual attrs. The Duke UI team will eventually do UI Testing on use/placement of "boss button."

Deployment Issue: We need to recommend that attribute bundles from the IdP and/or the home institution's SSO system are encrypted for confidentiality.

What are they? In EU Guidelines (GDPR), "special categories of personal data" include attributes that pertain to physical, physiological, genetic, mental, economic, cultural or social identity --- plus identifiers.

In the US, SSN is typically considered "sensitive."

The "final" version of the GDPR can be obtained via this website:
http://www.eugdpr.org/more-resources-1.html

"Persist Until Change" Handling

Config Tool Relevant: No.

UI Implication: Not different than normal "askme" handling.

[2] The code needs to ensure that the policy admin is not looking at their own policy!

Discussion Of “Persist Until Change”

The idea is that the end user may want to modify their policy decision about an attribute if the value of the attribute changes.

The traditional thinking on this topic would require a system to keep track existing values and poll for (or otherwise find out about) changes to to the values. This is expensive.

We have a different solution in CAR.

Designation of PEV for an II by the super-admin -- along with the user’s choice of “ask me” for “all other values” (in our self-service UI) -- gives the effect of “persist until change.” That is, if there’s a new value, the user will be asked about its release.

Given the current “intercept UI” implementation (at Duke), the user will have the opportunity to change their release decision for each of their current values of each II, not just the new one.

Non-Personal Attributes

> **Config Tool Relevant: Yes..**
> **UI Implication: Yes.**

An attr is “non-personal” when a its value is common across a large set of users and doesn’t provide any particular information about the user themself. For example, consider e.g Duke’s Buck Id. It is is essentially a contract id.

It’s possible that an institution does not want these attributes (and values) shown to an end user in the intercept UIs.

If that’s the case, the super-admin (or RH admin) can designate these attributes as “Always Send/Never Display.”

Attributes With Concatenated Values

> **Config Tool Relevant; Not at the moment.**
> **UI implication: Not at the moment**

Some institutions combine multiple “settings” into a single II value, where each setting might logically be its own value (or own II). Typically, the separate settings are concatenated for a single value.

Example: A “staff description” II that combines the staffers job type with their “on leave” status. E.g. “JL” to mean Junior faculty and on Leave.”

We (CAR) don’t have a way to allow for display names and descriptions that can fully handle this case (with its combinatoric combinations of values). However, an admin can use the standard description string for the value and II to help the user.”

Personal Attributes With Opaque Values

This type of II doesn’t need its own category. “Opaque” matters and we capture this in meta info via the Presentation Designation.

We note that “personal” matters not at all if the value is opaque.

Collections Of Similar Attributes

> **Config Tool Relevant: Not for version 1 but maybe a later version**
> **UI Implication: Yes**

In a typical higher education directory there are typically multiple variations on a number of attributes; These include a person’s name and what department they are in. For example, for “name” there is “sn,” “cn,” “Official name,” “display name,” etc.

It seems like it would be useful to group together “attributes that mean the same thing” in our self-service user interface in particular. And perhaps to allow an easy way in the UI to allow a single policy selection to apply to all the attributes in the collection.

We don’t have a means to do the yet but it’s something we want to consider for perhaps the next version of CAR.

Meta Info In Detail

Now that we’ve looked at the various categories of IIs, we can fully state what we need to hold as meta info about each II.

Meta Info about the II

- What is its Presentation Designation (opaque,encoded, transparent)
- What is its Policy Designation for end users (PAO or PEV)
- “raw” name/id of the II as defined by its Resource Holder (RH)

- Display string for II
- Description string for II
- What is the id of the Resource Holder
- Is the II “single-slotted” (e.g. marital status) or “multi-slotted” (e.g. eduPersonAffliation)

- Is the II “special” as per GDPR and other standards?

- Is the II ”AS/ND”? This means that we’ll “alway send/never display” all values of this II. **Note that the AS/ND designation supercedes the Presentation and Policy designations.** (Display name and description are in this case just for the super-admins themselves.)

For the IIs that are “encoded” (i.e. as defined by the Presentation Designation), we additionally need for each value:

  - the raw name of the value
  - the display name of the value
  - the description of the value
  - Is the value AS/ND?

Appendix A: Comparison with Attr Categories for Consent in the Shibboleth V3 IdP

The IdP v3 has the notion of a blacklist, a whitelist. and a regEx whitelist.

A blacklisted attribute is “always send; never displayed (AS/ND) ” but includes attributes with **opaque** values such as eduPersonTargetedId. In fact, that’s the only thing that’s discussed on the IdP page, i.e. attrs with opaque values. They don’t mention “matters for authorization” attributes such as persona non grata. (The IdP can’t do consent policy on values of attributes so there’s no means blacklisting for say a “bad actors” group.)

Seeing as we allow for both a friendly name and a description for each II, we believe that it’s not particularly useful to avoid displaying an II (or value) that might be confusing to a user without a description.

Of course, nothing would stop a super-admin from putting something like eduPersonTargetedId into the “AS/ND” category -- but that definitely isn’t the intent of the category.

Appendix B: Config/Deployment Tool Notes

In the future, we may deploy the CAR with a utility that contains policy rules that we think that a given set of Resource Holders (e.g. higher education IdPs) may typically want. There are some security implications to this however, if we wish the utility to use the CARMA API to instantiate the policies.

Appendix C: Policy Differences For Admins & End Users

In brief,

- Admins can make policy rules on all values of all attributes.

- Admins can use regular expressions in their designations of the values e.g. “red?*” to match “reddish_brown,” “red_orange,” etc.
- End users make release decisions (permit/deny/useAdvice/askMe) but are presented with making these decisions on a constrained policy consisting of:
    - a specific RP (e.g.HathiTrust)
    - a specific II
    - a specific value

- We don’t allow for regular expressions of values in COPSU policies.

- We also (separately) don’t allow for regular expression matching of a value to a display name and description. This means we can’t provide display names and descriptions for values that have a varying portion. (These attributes should be designated as “PRED=opaque” and “Policy=PAO.)

Comment: With respect to extending our display name facility to allow for regular expression matching, we have the “problem case” of a Resource Holder (in particular an IdP instrumented to defer to CAR for policy decisions) not knowing what attributes a given RP should get, and hence passing in all the user’s attributes and their values. CAR has to retrieve display names and descriptions for many of the attribute values passed in.

Allowing for regex matching for display names and descriptions for values could be very costly in terms of time for this case. (For example, at Duke, many users have hundred of attributes.)

Our specific concern is the delay that might occur in showing the consent UI to a user engaged an interaction with an RP.

Appendix D: Dealing With “isMemberOf’

Grouper has a display name for for each groupCAR/Informed Content service could import group names and associated display names from Grouper.

Issue: No one at Duke fills this in. (We can expect the same at other schools.)
Addressing this issue: Communication issue related to rollout; prepare community. Run import-from-grouper tool over and over. Tell people “you need to do this.”

[End]