i18n challenges in RCauth.eu

Considerations and solutions to universal i18n mapping

David Groep reporting work by Mischa Sallé

RCauth.eu & AARC

Nikhef PDP Advanced Computing Research

Nikhef

40th EUGridPMA Plenary Meeting

May 2017

RCauth.eu is operated by Nikhef as part of the
Dutch National e-Infrastructure for Research coordinated by SURF
for the benefit of the collective European Research and e-Infrastructures

SURF

CommonName – the big challenge

Requirements

- Contain a representation of the real name of the applicant as asserted by the IdP
  the opaque option is not very friendly to downstream services
- Must be unique and non-reassigned
- Allow – via the issuer – unique identification of the entity in the stated IdP

So we construct it out of 2 or 3 elements

1. Readable name of the applicant (max. 40 characters)
2. Unique Shortened Representation of the identifier provided by the IdP (16 characters)
3. *Optional:* ensured-uniqueness sequence number (max. 3 digits)

commonName – USR of the IdP identifier

Provides for issuer-assisted traceability of people. We pick and record the attribute used, preferring:

1. eduPersonUniqueID attribute (scoped) from the IdP (the ‘perfect’ attribute, available nowhere)
2. eduPersonPrincipalName (scoped) attribute from the IdP (a good attribute, OK 97% of the time)
3. eduPersonTargetedID constructed from IdP entityID and IdP-local (but targeted) opaque value

This is then pushed through the “Unique Shortened Representation”:

- first 16 characters of the base-64 encoded binary representation of the SHA-256 hash of the value, with any SOLIDUS (“/”) characters replaced by HYPHEN-MINUS (“-“) characters
- This mapping leaves 96 bits of entropy of the hash and a collision probability of 1 in 10^{28}

If the IdP gives	USR in CN RDN
40ea621a0a7355cf4fb1ca8d4f22a53d@nikhef.nl	uXmc85peL+35ONPO
davidg@nikhef.nl	Kydx8KT6xc1CHjD1
https://sso.nikhef.nl/sso/saml2/idp/metadata.php!02f7dfbb9605cf549e874bce55bfe0de030e9140	Wgt0ltSuF7BAA7FM

What does the CP/CPS say?

…

When the applicant name so constructed contains characters outside the set of PrintableString, these characters shall be minimally-casted to their closest PrintableString equivalent

or

– when impractical because no single-character mapping exists – shall be replaced by the upper-case character "X".

https://aarc-project.eu/wp-content/uploads/2017/04/AARC-JRA1.4I.pdf

commonName – readable name element

REFEDS R&S gives a subset of attributes that should be released: displayName, givenName + surname, commonName. We construct the readable name from (in order of preference)

1. the *displayName* attribute from the IdP
2. the *givenName* attribute, followed by a space, followed by the *sn* attribute from the IdP
3. the *commonName* (cn) attribute from the IdP

and then make it printable using *java.text.Normalizer.Form.NFD* and map the remainder to “X”

If IdP sends us this UTF-8	Representation in CN RDN
Jőzsi Bácsi	Jozsi Bacsi
Guðrún Ósvífursdóttir	GuXrun Osvifursdottir
Χρηστος Κανελλοπουλος	XXXXXXX XXXXXXXXXXXX
簡禎儀	XXX

but Νικόλας Λιαμπότης did not like that … and I understand …

- Current *java.text.Normalizer.Form.NFD* and ‘X-ing’ the rest particularly bad for Greeks, Bulgarians, Chinese, Russians, Georgians, Serbians

ICU - International Components for Unicode (icu-project.org) appears to be better, but:

- there are many options for transliteration
- some code points shared between different languages, that prefer different transliterations
- some code points are absent even in UTF-8 causing ambiguity

Baseline proposal for RCauth from now on:

UTF-8 $\xrightarrow{\text{ICU}}$ Latin-1 $\xrightarrow{\text{ICU}}$ ASCII $\xrightarrow{\text{regex}}$ IA5String (we need PrintableString + “@” and minus [:/=])

It’s all Greek to me!

ICU can do many things to Λιαμπότης

http://userguide.icu-project.org/transforms/general#TOC-Greek

- Greek-Latin → Liampótēs → Liampotes
- Greek-Latin/BGN → Liambótis → Liambotis
- Greek-Latin/UNGEGN → Liampóti̱s → Liampotis

and the official (passport) Greek ELOT-743 transliteration is “Liampotis”

But straightforward translation is not always good

Just Any-Latin fails for Slavonic unique “sh” sounds. E.g. for ‘Миша’

- with *Any-Latin* becomes ‘Miša’ which then translates into ‘Misa’ after the Latin-Ascii

but you want to see ‘Mischa’, so you need

- first *Russian-Latin/BGN*, making it ‘Misha’, which is slightly better, then do *Any-Latin* (1-to-1)
- but “*Russian-Latin/BGN+Serbian-Latin/BGN*” is different from the reverse …

First Any-Latin/BGN, then Any-Latin, to fix mapping to → š and the → s

- Բարեւ աշխարհ → Barev a**sh**kharh (with the /BGN, to ensure the “sh”)
- ישראל → ysr'l (taken care of without the /BGN, otherwise the ש never makes it)

And Unicode does not distinguish the *diaeresis* and the *umlaut*

- Mühlstraße ⟶ Muhlstrasse   is wrong, should have been ‘Muehlstrasse’
- reünie ⟶ reunie   is good, you definitely don’t want ‘reuenie’

As the so for stability, we keep Any-Latin here and treat all as a diaeresis

But straightforward translation is not always good

So the (for now) best combination seems to be the ordered transformation:

```
Transliterator.getInstance( "Russian-Latin/BGN;"+
          "Serbian-Latin/BGN;"+
          "Greek-Latin/UNGEGN;"+
          "[:Nonspacing Mark:] remove;"+
          "Any-Latin/BGN;“ +
          "Any-Latin;“ +
          "Latin-Ascii“
);
result.replaceAll("[^\\p{Lower}\\p{Upper}\\p{Digit} '()+,-.?@]",  "X");
```

← *ordering to retain “ш” → “sh”*

← *Fixes greek Λ adding a useless space*

← *Retain proper “sh” when coming from Armenian or Hebrew by /BGN first*

What will we get?

```
$ java -cp icu4j-59_1.jar:. transliterate2 [...]
 "Jőzsi Bácsi" "Guðrún Ósvífursdóttir" "Χρηστος Κανελλοπουλος"
 "簡禎儀" "毛泽东"
Input:   Jőzsi Bácsi
Output:  Jozsi Bacsi
Input:   Guðrún Ósvífursdóttir
Output:  Gudrun Osvifursdottir
Input:   Χρηστος Κανελλοπουλος
Output:  Christos Kanellopoulos
Input:   簡禎儀
Output:  jian zhen yi
Input:   毛泽东
Output:  mao ze dong
```

Organisation name – any better?

RCauth makes the *SubjectDN* O component based on

- schacHomeOrganisation attribute value
- organisationDisplayName from the SAML meta-data
- URI Entity ID: domain component (hostname or subdomain) of a URL, or the full URN

Each truncated after 63 characters (it's not needed for uniqueness, just human use)

- schacHomeOrganisation is fine, as per spec it's RFC1035
  some strange organisations will not be able to use it, but that's not an RCauth issue
- organisationDisplayName can be transliterated like the commonName
- URNs are printable string or castable, but do contain ":" – which we will make into an "X"
- URL may be or contain an IDN – here we propose to use punycode of this IDN from now on

xn--pxabb4d.gr (εδετ.gr) instead of (today) XXXX.gr, or the ICU 'edet.gr'

Planning

Deploy to RCauth.eu as soon as possible

- No or very minor change to CP/CPS needed (it's vague enough)
  for the "O" component, the same text as used for the CN will be added

- No users yet impacted, but we need to do this before the first Greek shows up …

Do you endorse this change to go into effect now?

Try yourself?
https://github.com/rcauth-eu/aarc-delegation-server/blob/master/delegation-server/src/main/java/org/delegserver/oauth2/generator/DNGenerator.java

Help? Ask Mischa Sallé at <msalle@nikhef.nl>

www.rcauth.eu/policy

SURF Nikhef

RCauth.eu is operated by Nikhef as part of the Dutch National e-Infrastructure for Research coordinated by SURF for the benefit of the collective European Research and e-Infrastructures

Thank you

Any Questions?

davidg@nikhef.nl

ca@rcauth.eu

AARC

https://aarc-project.eu

© GÉANT on behalf of the AARC project.
The work leading to these results has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 653965 (AARC).