IDNA 2003 & IDNA2008

Patrik Fältström
Head of Research and Development
Netnod
paf@netnod.se

What is IDN?

- Internationalized domain names (IDN) is a new feature in the Domain Name System
- It gives the ability for users to express domain names using non-latin script
- What is used in the DNS protocol is still latin script
  - IDN's are identified by the prefix "xn--"

Where do we see IDN?

- In a standard (IDNA)
  - What codepoints are allowed, and how?
- In Applications (IDNA)
  - Encode/decode to/from "xn--" form
- In policies that registries set up
  - What domain names can be registered?

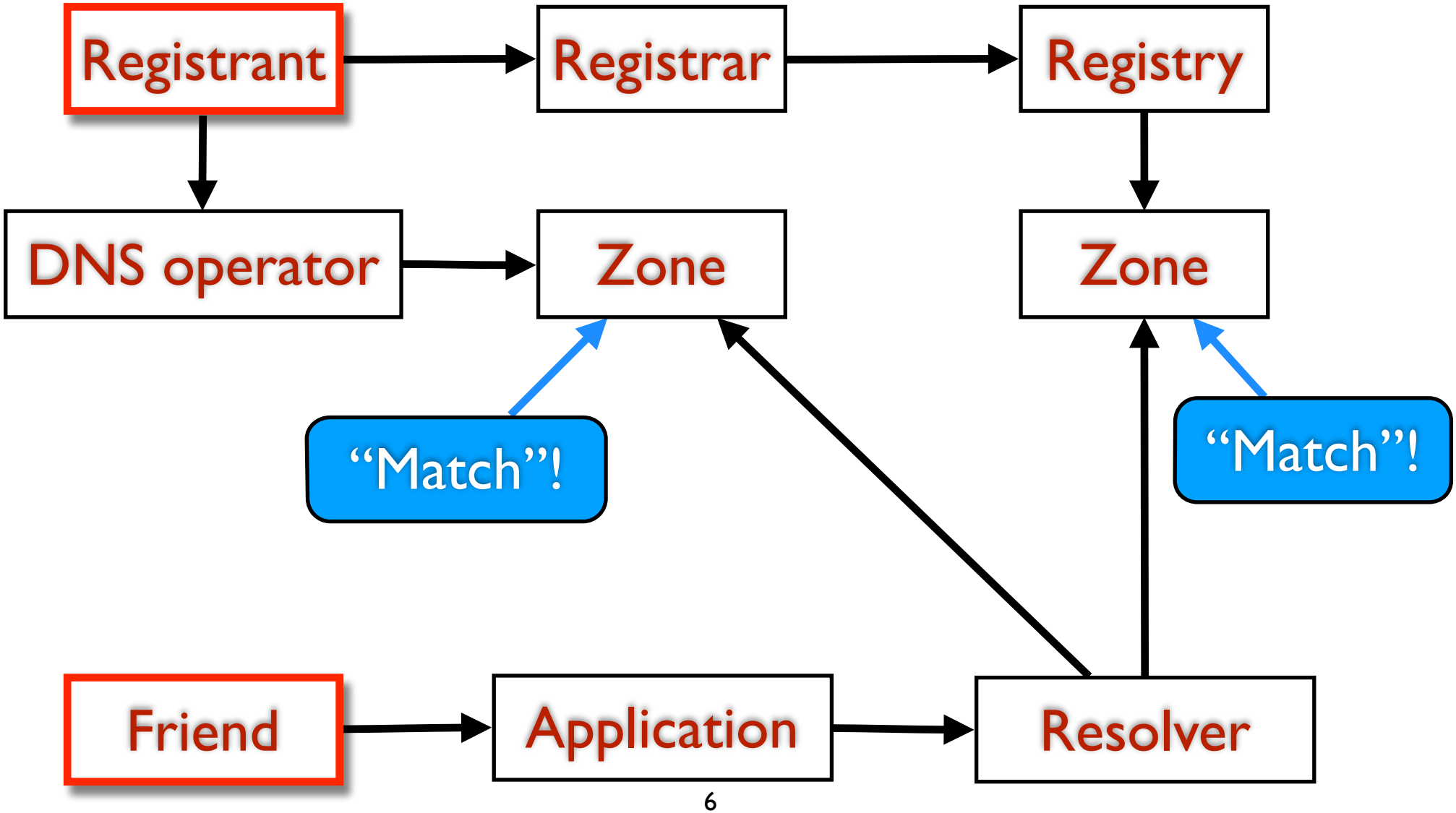
Why not “just unicode”?

- Unicode include tons of characters, many of them look the same
- Unicode is updated now and then, and we need to agree on what version to use
- Many protocols and applications can only handle latin script

More precisely

- I register a domain name:
  - fältström.se
- You type in a domain name:
  - FÄLTSTRÖM.SE
- Should you “find” what you want to find?
  - Solution, use: xn--fltstrm-5wa1o.se

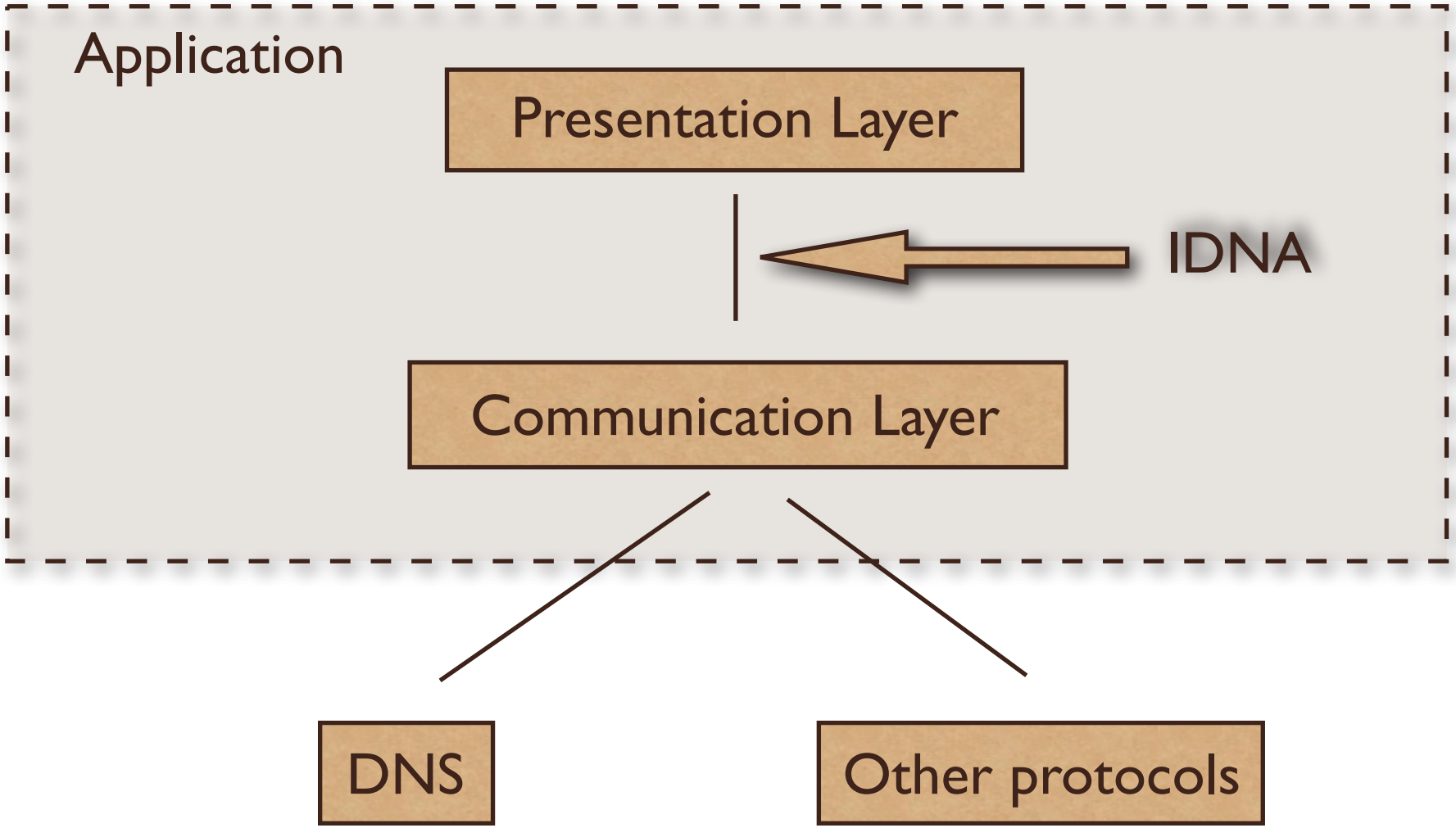
In even more detail



Conclusion

- For the implementation of IDN to work, what the registrant register and what other people use, must match when comparing the two domain names in various DNS servers in the world
- It is much easier to define what strings to send to the matching rule than change the matching rule in every DNS server

Where is this applied?



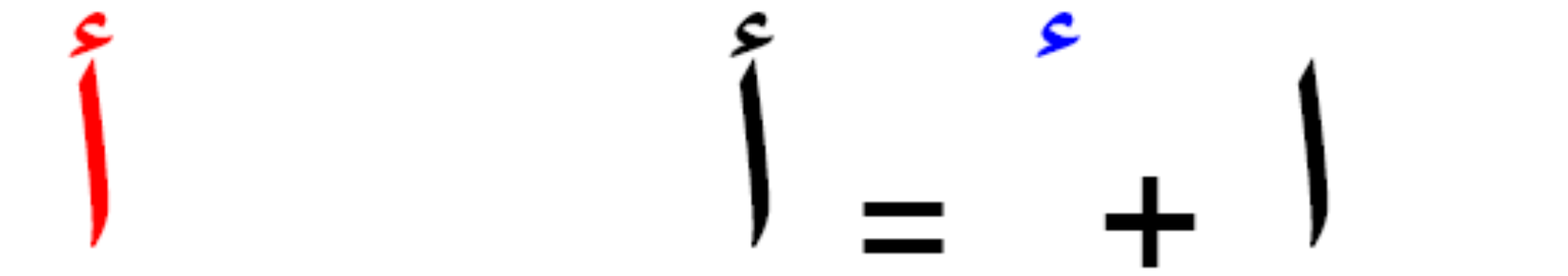
Spot The Difference…

≠

input[0] = U+0627
input[1] = U+0654
input[2] = U+066e
input[3] = U+06ec
input[4] = U+0627

input[0] = U+0623
input[1] = U+0646
input[2] = U+0627

They Look The Same To Us … But Not To A Computer



When 1 is not 1…

Arabic-Indic VS. Eastern Arabic-Indic digits

١٢٣٧٨٩٠

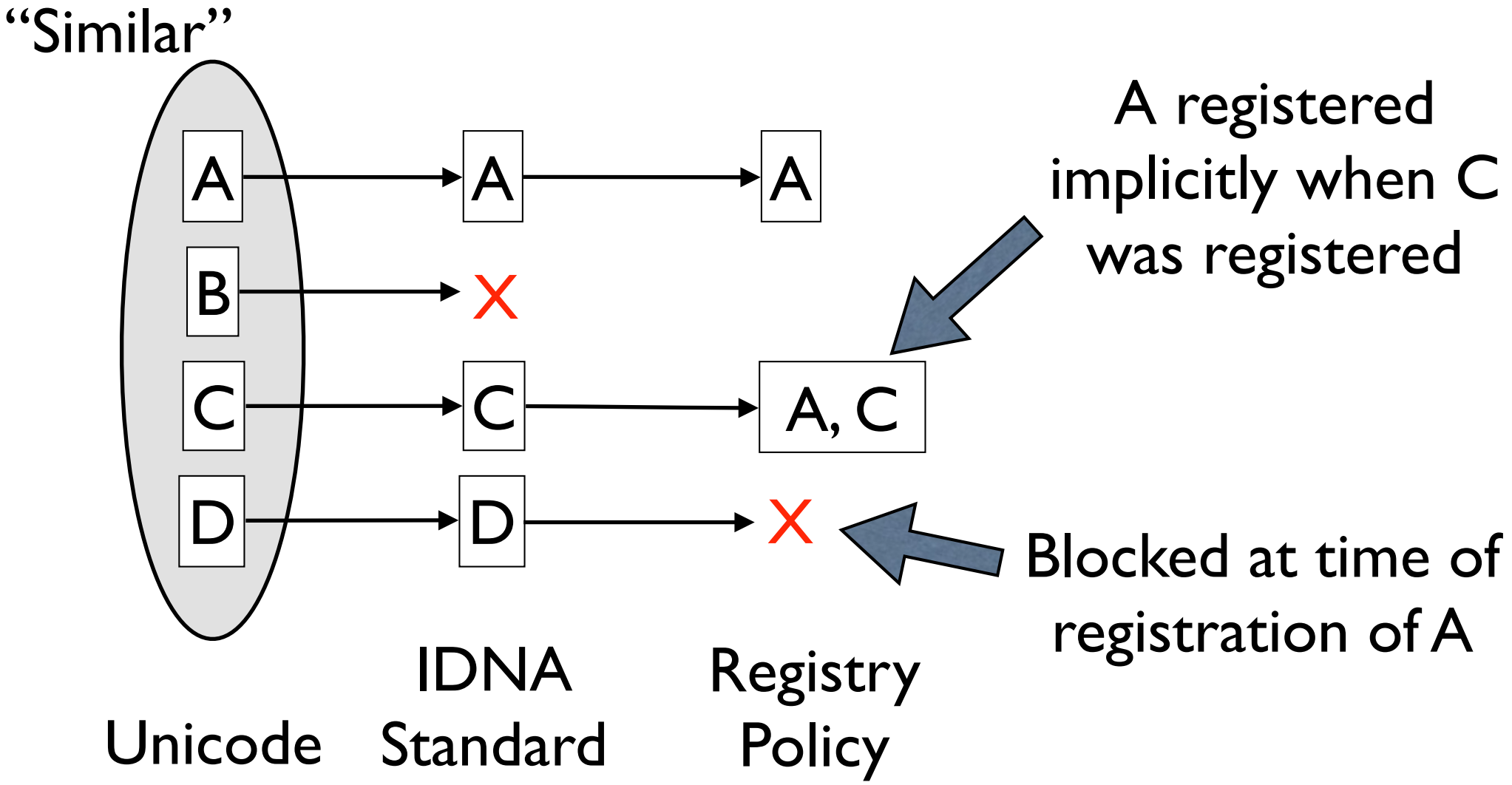
input[0] = U+06f1
input[1] = U+06f2
input[2] = U+06f3
input[3] = U+06f7
input[4] = U+06f8
input[5] = U+06f9
input[6] = U+06f0

≠

١٢٣٧٨٩٠

input[0] = U+0661
input[1] = U+0662
input[2] = U+0663
input[3] = U+0667
input[4] = U+0668
input[5] = U+0669
input[6] = U+0660

Variants, policies etc



The Arabic Language use only a small part of the Arabic Script table

Accepted characters for Arabic, Persian, Urdu, and Pashto

Conclusions

- We have to agree on what codepoints can be used in domain names, specifically when multiple variations for expressing "the same" character exist
- Agreements have to be global, across language groups, as scripts are shared
- We need rules both for TLDs themselves and for 2nd level domain registrations

In the beginning

3454 Preparation of Internationalized Strings ("stringprep"). P. Hoffman, M. Blanchet. December 2002. (Format: TXT=138684 bytes) (Status: PROPOSED STANDARD)

3490 Internationalizing Domain Names in Applications (IDNA). P. Faltstrom, P. Hoffman, A. Costello. March 2003. (Format: TXT=51943 bytes) (Status: PROPOSED STANDARD)

3491 Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN). P. Hoffman, M. Blanchet. March 2003. (Format: TXT=10316 bytes) (Status: PROPOSED STANDARD)

3492 Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA). A. Costello. March 2003. (Format: TXT=67439 bytes) (Status: PROPOSED STANDARD)

What was this - IDNA2003

- 3454 Specifies overall algorithm - stringprep
- 3490 Specifies IDN algorithm - IDNA
- 3491 Specifies Nameprep
- 3492 Specifies Punycode

stringprep

- With profiles, any Unicode based string can be converted to another Unicode string so that they can be compared
  - Include illegal codepoints
  - Include mapping table
  - Give ability to create profiles
- Used for IDN, LDAP and other protocols

idna

- Algorithm for how to convert a domain name with Unicode codepoints to ascii
  - How to use the stringprep profile and unicode
  - Includes specification on how to handle unallocated codepoints
  - “core” to IDN standard

nameprep

- Specific stringprep profile for unicode based domain names
- Convert a domain name with unicode codepoints to one of
  - Illegal domain name
  - Domain name with Unicode codepoints

punycode

- Converts a label with unicode codepoints to a domain name in ascii
- Example:
  - fältström
  - xn--fltstrm-5wa1o

What happened?

4690 Review and Recommendations for Internationalized Domain Names (IDNs). J. Klensin, P. Faltstrom, C. Karp, IAB. September 2006. (Format: TXT=100929 bytes) (Status: INFORMATIONAL)

In short...

- Explains the problems in the earlier standards
  - Bidirectional scripts
  - Non-spacing codepoints
- Explains the problems with scripts not yet created when IDNA was written
- Explains problem with versioning of Unicode
  - Old standard based on Unicode 3.2

Example

- If a label include a character that has right to left directionality, both first and last character of the string has to have right to left directionality
- Creates problem if for example the string ends with a codepoint with no directionality

```
U+05D9 HEBREW LETTER YOD (R)

U+05D9 HEBREW LETTER YOD (R)

U+05B4 HEBREW POINT HIRIQ (NSM)

U+05D5 HEBREW LETTER VAV (R)

U+05D5 HEBREW LETTER VAV (R)

U+05D0 HEBREW LETTER ALEF (R)

U+05B8 HEBREW POINT QAMATS (NSM)
```

- Note that last codepoint has no directionality (Non Spacing Mark)

New version, IDNA2008

- Also consists of a few documents
- Will not change punycode
- Backward compatible
- Does explicitly talk about what is possible "*to register*" in the DNS (old IDNA say what is possible "*to use*", and that include mappings)
- **ONLY** define what is possible to register in DNS (U-label / A-label)

New documents

- 5890 Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework J. Klensin
- 5891 Internationalized Domain Names in Applications (IDNA): Protocol J. Klensin
- 5892 The Unicode Code Points and Internationalized Domain Names for Applications (IDNA) P. Faltstrom
- 5893 Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA) H. Alvestrand, C. Karp
- 5894 Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale J. Klensin

RFC 5890

- Addresses the concerns in the IAB document RFC 4690
- Explain how the issues are resolved

RFC 5891

- Replaces the IDNA specification
- Core specification of new IDN standard

RFC 5892

- Defines **algorithm** to use to calculate whether a codepoint in Unicode is in one of the categories
  - PVALID (Protocol Valid)
  - CONTEXTO / CONTEXTJ
  - DISALLOWED
  - UNASSIGNED

RFC 5893

- Gives specifics for bidirectional scripts

But IDNA2003 had mappings

- Mappings are not part of IDNA2008
- Labels MUST be stable under NFC
- Codepoints in label MUST pass bidi requirements
- Codepoints MUST be ok according to algorithm specified in tables document (which might include contextual rules)
- We MIGHT separate documents on mapping, recommended behaviour for different applications etc

Implications for registry

- Should not require any change for a registry that do have a policy for what subset of IDNA2003 codepoints one can register
- Changes might be required cases:
  - ➡ Registry that allow **any** IDNA2003 codepoint
  - ➡ Registry that did allow registration of the few codepoints that indeed have changed, ß for example
  - ➡ Registry that define codepoints based on possible **input** to IDNA2003 (and not what is actually registered in DNS)

Why is this needed?

- IDNA standard **must** be independent of Unicode version
- IDNA standard **must** handle bidirectional scripts
- ...plus other things mentioned in RFC 4690

What has happened?

- Unicode has moved from 5.2 to 7.0
- RFC6452 - The Unicode Code Points and Internationalized Domain Names for Applications (IDNA) - Unicode 6.0

RFC6452

- Changes from Unicode 5.2 to 6.0
  - U+0CF1 KANNADA SIGN JIHVAMULIYA
  - U+0CF2 KANNADA SIGN UPADHMANIYA
  - U+19DA NEW TAI LUE THAM DIGIT ONE
- Conclusion: No update of IDNA2008 is needed

Unicode TR#46

- Client software, such as browsers and emailers, faces a difficult transition from the version of international domain names approved in 2003 (IDNA2003), to the revision approved in 2010 (IDNA2008). The specification in this document provides a mechanism that minimizes the impact of this transition for client software, allowing client software to access domains that are valid under either system.

- The specification provides two main features: One is a comprehensive mapping to support current user expectations for casing and other variants of domain names. Such a mapping is allowed by IDNA2008. The second is a compatibility mechanism that supports the existing domain names that were allowed under IDNA2003. This second feature is intended to improve client behavior during the transitional period.

Unicode TR#46

- Unicode Consortium TR#46 is incompatible with IDNA2008
- Includes good material for people that have implemented IDNA2003
- Transition must be made to IDNA2008

draft-klensin-idna-5892upd-unicode70-00.txt

- Codepoint in Unicode 7.0:
  - U+08A1 : ARABIC LETTER BEH WITH HAMZA ABOVE
- Existing allowed codepoints:
  - U+0628 : ARABIC LETTER BEH
  - U+0654 : ARABIC HAMZA ABOVE
- Should U+08A1 be disallowed?

Patrik Fältström
paf@netnod.se