Internationalized JIDs

Peter Saint-Andre

XMPP WG Interim Meeting

February 7, 2011

Unicode Recap (1)

- Every character is a "code point"
- Characters have properties, e.g.:
  - letter, number, symbol, etc.
  - uppercase vs. lowercase (etc.)
  - modifiers (e.g., accent marks)
  - left-to-right vs. right-to-left

Unicode Recap (2)

- We decide how to handle characters based on their properties
- A character can be *equivalent* to another character or a sequence of characters
- Things like Å and ç are "composite characters"

Unicode Recap (3)

- Two kinds of equivalence
- Canonical: "this character is the standard for that one" (e.g., Å ≡ Å or ç ≡ c + ¸)
- Compatible: "this character suffers with that one" (e.g., IV ≈ I + V or ſ ≈ s)

Unicode Recap (4)

- *Decomposition* analyzes a character into its component units
- Two kinds of decomposition: canonical and compatible
- Order matters (e.g., ώ̰̃ ≡ ω + ̔ + ̃ + ̰)

Unicode Recap (5)

- *Normalization* removes alternate representations of equivalent sequences so that we can convert the data into a form that can be compared for equivalence
- Normalization can involve both decomposition and recomposition, and both canonical and compatibility rules

Unicode Recap (6)

- NFD = canonical decomposition
- NFKD = canonical and compatibility decomposition
- NFC = canonical decomposition and recomposition
- NFKC = canonical and compatibility decomposition and recomposition

PRECIS (1)

- As we know, IDNA2008 moved away from stringprep for domain names
- Other technologies want to move as well (for Unicode agility and other reasons)
- PRECIS WG is working on a replacement for use by other stringprep customers
- XMPP WG to provide input to PRECIS

PRECIS (2)

- Stringprep provided:
  - Mappings (e.g., spaces, prohibited characters, case folding)
  - Normalization (typically NFKC)
  - Handling of right-to-left scripts
- PRECIS to provide similar "services"

PRECIS (3)

- Probably define string classes such as username, password, free-form identifer
- Enable sub-classing of string classes
- Define processing rules for each class based on Unicode properties
- Mapping rules probably included

XMPP Issues

- We have two stringprep profiles
  - Nodeprep for localparts
  - Resourceprep for resourceparts
- Many issues need to be decided....

Localpart vs. Resourcepart

- Continue to use treat localparts and resourceparts differently?
- Is localpart close enough to a "username" in most or all cases? (suitably sub-classed from PRECIS "username")
- Do we want resourcepart to be free-form?

Normalization Forms

- Is NFKC too Smart™ for its own good?
- Do we really need to recompose?
- Is NFD good enough for us?
  - Over the wire?
  - In memory (for comparison)?
  - In storage?

Mapping & Such

- Case folding for "username" but not "free-form string"?
- How do we handle *width* of East Asian characters?
- RTL – re-use bidi rule from RFC 5893?
- Define or recommend locale-specific rules?

Registration & User Interface

- Define "registrar-like" policies for XMPP servers (esp. IM servers)?
- Prohibit mixed scripts or encourage clients to treat them with caution? (XEP-0165)

JID Slots

- Account names
- Roster items
- Multi-User Chat / PubSub addresses
- vCards
- Privacy lists
- Other?

Enforcement & Error Handling

- Server only?
- Can we depend on the client?
- Component (e.g., MUC room)?
- All of the above?
- How do we handle errors?

Migration

- How are JIDs stored now?
- What data needs to be scrubbed?
- Do we need a "flag day"?
- How do we handle migration-related errors?

Making Progress...

- How will we work through these issues?
- What is the right division of labor between PRECIS WG, XMPP WG, and XSF?
- Need to coordinate with SASLprep
- How can we provide input to (or work within) the PRECIS WG?