Objectives

• Figure out how to proceed in the “post-IDNA2008” world

• Can we use Marc’s proposed framework?  
  <draft-blanchet-precis-framework-00>

• What is the simplest possible approach?
XMPP Addresses (JIDs)

- `<localpart@domainpart/resourcepart>`
- Localpart can be a username, a chatroom, a bot, etc.
- Domainpart is a server or component
- Resourcepart can be a device identifier, a nickname in a chatroom, etc.
JID Examples

• <münchen.de>
• <司馬 安平@jabbercn.org>
• <jiří@jabber.cz/domů>
• <jdev@conference.jabber.org/the ♔>
Domainpart

- Migrate to IDNA2008
- Support only U-labels on the wire?
- U-label comparison in IDNA2008 occurs without case-folding – is this OK?
- Translation between U-label (XMPP) and A-label (DNS) introduces processing load?
Localpart (I)

- Often a username, which might be my real name in my native language or script.
- All “namelike” characters should be valid (but do we really need symbols, stars, etc.?)
- XMPP server is "registrar" for accounts; do we need registration policies?
- Ideally, consistent with email, SASL, etc.
Localpart (2)

- Currently use a stringprep profile called Nodeprep
- Disallow: SP " & ' / : < > @
- Case mapping: B.1 + B.2 from stringprep
- Normalization: NFKC
- Bidi: essentially unspecified
Localpart (3)

- Disallow code points that are disallowed now
- Case mapping: as in IDNA2008, or specify case-folding methods?
- Normalization: NFC?
Localpart (4)

• Possible approaches:
  • Use whatever we develop for SASL to replace SASLprep ("simple username")
  • Harmonize localpart with EAI (but: disallowed chars are not the same)
  • Define separate identifier class?
Resourcepart (1)

• Essentially a free-form string (location, machine name, fanciful text such as a glyph or dingbat)

• All “namelike” characters should be valid

• Do we really need things like “the ♚”?

• Case sensitive (but is this a good idea?)
Resourcepart (2)

- Currently use a stringprep profile called Resourceprep
- Disallowed: non-ASCII spaces, controls, C.3-C.9 from stringprep
- Mapping: B.1 from stringprep
- Normalization: NFKC
- Bidi: essentially unspecified
Resourcepart (3)

- Restrict allowable code points?
- Case mapping: continue to ignore case?
- Normalization: NFC?
Possible approaches:

- Re-use whatever we define for localpart
- Define "free-form identifier class"?
- Use Net-Unicode (RFC 5198)?
- Say it's "just UTF-8"?