rfc3530bis i18n
(Unicode, why is it always Unicode?)

David Black, EMC

rfc3530bis-25: IESG Issues with i18n

- Stringprep specified, but not required
- Some comments that stringprep should be required
  - Infeasible due to inability to rewrite physical filesystems, operating system Unicode frameworks, etc.
- Other comments objected to complexity
  - Comments were not wrong ✪
- Reality: Stringprep not implemented for NFSv4.0

rfc3530bis i18n: New Approach

Base requirements on existing implementations:

- Multiple approaches to server Unicode normalization
  - Ignore normalization
  - Enforce a specific normalization form
  - Normalization-insensitive LOOKUPs
- Multiple approaches to UTF-8 checking
  - Reject strings that are not valid UTF-8
  - Accept all string even if they are not valid UTF-8.
    - Needed for legacy character set support.
  - UTF-8 remains the preferred string encoding
- All stringprep text removed

Working group ok with this approach

rfc3530bis i18n: WG list discussion

- Some clarifications suggested by David Black and incorporated in -31.
- Rewrote case folding text: new text is based on précis drafts
  - Issue arose around case folding for user and group names
    - Cannot forbid this - at least one server can be (mis?) configured to case fold
    - Current draft text uses “SHOULD NOT”, and explains:

```
The result SHOULD be a
canonically equivalent Unicode string [UNICODE].  Other sorts of
string modification, including case mapping or folding, SHOULD NOT
be done as such modifications may cause the server to merge
identities that are separate at the client.
```

Status of i18n in rfc3530bis-31

Reflects existing implementations

Ready for Working Group Last Call

David Black has discussed i18n approach with key Area Director (Pete Resnick, APP):

- Ok with overall approach ("running code")
- Will look at text during WG Last Call to find issues earlier than IESG Review