In response to the “Call for Participation: IAB workshop on Explicit Internet Naming Systems”, I would like to put forward a position paper addressing one of the challenges mentioned:

Unicode allows for both combining characters and composed characters when local language communities have different practices. When these do not have a single normalization, context is required to determine which to produce or assume in resolution. How can this context be maintained in Internet systems?

Unicode, even if restricted to characters used in modern-use languages, contains an unexpectedly large number of non-normalizable variations in code point (or sequence) that are displayed in an identical (or very nearly identical manner). If not carefully handled, they will undermine users' ability to uniquely identify and/or reliably distinguish different network identifiers.

In some cases, it may be feasible to summarily reject one of the variations (by excluding certain code points) without affecting the ability to create usable mnemonic identifiers for given script, writing-system or language. However, in other cases there is no a-priori way to select a preference. In some cases, such a preference does not exist even if the intended target community is known with some specificity while in other cases, a known context, limited to some community, would mark one of the encodings as preferred. A common situation is that users from different contexts have to access the same resource, but each community differs in their choice of preferred code points. (Chinese, traditional vs. simplified, or Arabic vs. Urdu for certain code points).

In addition to the issue of non-normalizable variants, there is a reasonably numerous set of code points in Unicode that, while being allowed in the protocol (e.g. PVALID in IDNA 2008) nevertheless may be unsuitable for identifiers or require special considerations to be taken into account before being able to be used safely.

Finally, there are many scripts in which arbitrary sequences of code points are not expected by either users or client systems (particularly display). In many of them, the logic of the writing system operates on the “syllable” level, and non-conforming syllables are to a degree uninterpretable, even if they can be expressed in Unicode. For safe and predictable network identifiers, such scripts require additional rules (beyond those contained in IDNA 2008 CONTENTO) that limit which code points may occur adjacent to other ones. The good news is that such rules do not need to fully model the syllable structure to be effective, but the required syllable structure may vary by language.

Due to the universality of Unicode, few individuals have a reasonably complete understanding of code points that may be considered “troublesome” for network identifiers for one of these reasons; however different communities, over time, have looked into local aspects of these issues, and there has been an ongoing multi-year project to identify issues relevant to the root zone. (The author is one of the participant in that activity).


That draft also discusses particular examples of affected code points and the
problems they present in some detail while providing suggestions on possible strategies for mitigating any of these issues. It is being extended to also cover problems arising from cross-script or mixed-script usage of Unicode.

One of the recommended mitigation strategies centers on a recommendation to define blocked variants. This strategy is relatively cheap to implement and increases the robustness of registered identifiers. However, it is not widely understood how to best support variants in registration policies, and even why not all types of variants are equal. A recent development (RFC 7940) resulted, for the first time, in a standardized, machine processable framework for representing variants (and other mitigation strategies, such as code point contexts), allowing the emergence of common tools to automate and guarantee that aspect of registration policies.

As the acceptance of IDNs increases, a better shared understanding of required policies to make them safer and usable is required, above and beyond simply being IDNA 2008 conformant.

In that context see also https://www.ietf.org/id/draft-freytag-lager-variant-rules-06.txt and https://tools.ietf.org/html/draft-klensin-idna-rfc5891bis-00

– Asmus Freytag