

# Locale Extensions

Ben Allen

**27 September, 2023**

# Three interrelated problems

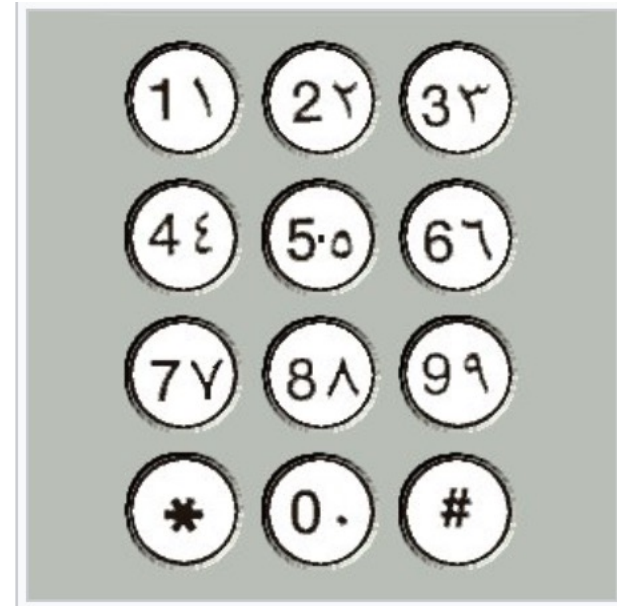
1. Oftentimes there are multiple numbering systems used in one locale with no easy way for users to indicate which they prefer
2. Often users will have content tailoring desires that differ from the defaults used for the locale the content they're viewing is in
3. Some users have combinations of preferences that differ from the defaults for their browser's locale and also the defaults for the content they're viewing



# The problem #1

Some regions have multiple commonly-used number systems

- `hi` defaults to "latn", even though many people requesting that locale might prefer "deva"
- Competing number systems are used in the United Arab Emirates, among other Middle Eastern/South Asian countries.



(telephone keypad with both Eastern Arabic and Latin numerals)



# The problem #2

Often users will have content tailoring desires that differ from the defaults used for the locale the content they're viewing is in.


A plurality of sites only offer content in English, and often in a regional variant of English with highly idiosyncratic defaults for hours of the day, temperature measurement, first day of week, etc. Users might want to view these things in a more globally common way.






# The problem #2

Elev 92 ft, 51.51 °N, 0.13 °W

London, England, United Kingdom

 66° CHARING CROSS STATION | CHANGE ▾

TODAY HOURLY 10-DAY

Time	Conditions	Temp.	Feels Like	Precip
3:00 pm	 Light Rain	66 °F	66 °F	<a href="#">74 %</a>
4:00 pm	 Showers	66 °F	66 °F	<a href="#">40 %</a>
5:00 pm	 Showers	65 °F	64 °F	<a href="#">40 %</a>

**BBC** Sign in Home

**WEATHER** Enter a city

San José + Add to your locations


Today

 29°  
18°

Tue 19th

 28°  
19°

Thundery showers

0900	1000	1100	1200	1300	1400	1500	1600
 26°	 27°	 28°	 28°	 27°	 27°	 26°	 24°



# The problem #3

Often users will have content tailoring desires that differ from the defaults used for the locale the content they're viewing is in.

3. Some users have combinations of preferences that differ from both the default for the locale specified in their OS and also the default for the content they're viewing
  - Someone who wants content tailored for `en-US`, except they want calendars to show Monday as the first day of the week.



# The problem within the problem

Fingerprinting opportunities abound!

- Users who list multiple languages in Accept-Language are likely making themselves immediately individually identifiable.
- Were we to provide a mechanism for clients to directly convey **all** their OS preferences to servers, this would also likely make any users with non-default settings **immediately** individually identifiable.



# Our goal:

1. Let users express their content tailoring preferences as fully as possible
2. While prioritizing tailorings that might seriously impact content legibility/intelligibility if ignored
  - (most notably: numbering system)
3. While leaving as small a fingerprinting surface as possible
4. Ideally, *smaller* than the surface offered by one item in Accept-Language.
5. All fingerprinting that happens must be detectable – no passive fingerprinting!





# Not our goal

1. Allowing users to express arbitrary preferences
  - This is a fingerprinting nightmare!
2. Making web applications that are as flexible as native applications
  - Not possible – the internet is a hostile place
3. Finding a way to avoid revealing any information at all
  - No revealed information -> no localization



# The elephant in the room

- This proposal is more directly related to other standards organizations (W3C)
- There is an possibility that the solutions we discover may require touching Intl
- This proposal could serve as a locus for discussing the approach to fingerprinting/privacy related problems related to 402 in general
- We are *only* asking for stage 1 – we’re exploring solutions, and may ultimately not attempt to advance to stage 2



# How close can we get?

How close can we get to complete localization without making users individually identifiable?

1. We could allow support for something like the ``-u-rg`` tag:
  - Allow users to express the concept that regardless of what locale the content they receive is, they would like that content tailored to match the first language in the `Accept-Language` header.
  - This doesn't reveal anything more than what's revealed in `Accept-Language`, but current privacy best practices say that "this is already revealed elsewhere" is not a valid defense for adding features that provide fingerprinting vectors



# How close can we get?

How close can we get to complete localization without making users individually identifiable?

1. We could *separate out individual components* of the preferences that could be expressed by ``-u-rg``, and send the individual components.
  - This is a **giant** gain for some very common use cases: people preferring content tailored as in ``en-US``, people preferring content tailored for global standards.
  - Enough people want those collections of preferences (representable as `"-u-fw-mon-hc-h23-mu-celsius"` and `"-u-fw-sun-hc-h12-mu-fahrenhe"`) that if people could ask for them, they'd be able to hide in the crowd.



# How close can we get?

How close can we get to complete localization without making users individually identifiable?

1. We could *separate out individual components* of the preferences that could be expressed by ``-u-rg``, and send the individual components.
  - Possibly even a gain for people with preferences that fall between the two (for example, people using the defaults for es-MX, which can be represented as ``-u-fw-sun-hc-h12-mu-celsius``), provided enough others use that set of preferences.
  - **User research is required.**



# Proposed solution #1: the complicated one

- For each locale, determine via **user research** what sets of **default preferences for other locales** might be safely expressed without unnecessary reductions in the size of each user's anonymity set
- For each locale, determine via **user research** what, if any, alternate preferences for that locale might be in common use (i.e. the 'hi-u-nu-deva' example)
- Why default preferences for locales?
  1. it's less difficult to measure than measuring bespoke preferences
  2. Likely the only preferences commonly used are either defaults in another locale or alternates for the current locale.



# Proposed solution #1: numbering system

- ``-u-nu`` is the highest priority extension to allow
- Consider *always* allowing users to select the numbering system designated as “native” for their locale?



# Proposed solution #1

During implementation:

- Implementers determine what combination of preferences are likely safe in each locale.
- These are included with each revision of the browser.

During use:

- Browser reads OS preferences from system
- Browser determines (pick your favorite algorithm) which of those preferences are expressible through the available preference strings
- Only those preferences are revealed to the server





# Settable preferences in current revision

In the current revision, these are the –u extension tags we're concerned with:

- ``ca``: calendar
- ``fw``: first day of week
- ``hc``: hour cycle
- ``mu``: temperature measurement unit
- ``nu``: numbering system



# Mechanisms:

1. `Client Hint` headers for each of the five tags.
  - If a server has to explicitly request each one, it becomes more clear when they're gathering irrelevant data for fingerprinting.
  - (makes the fingerprinting vector an active fingerprinting vector)
2. A JavaScript API that can be used to discover settings for each of these five tags
  - or rather, for each of the five tags **that are actually expressible using a safe -u extension string**
  - Others left undefined
  - Settings must be requested **individually** – as with `Client Hint`s, this prevents passive fingerprinting.



# **User research may show opportunities for simplification**



# Proposed solution #2: the simpler one

Only support a small number of agreed-upon tags, allow them in all locales

Locale Extension Name	Unicode Extension Key	Possible Values
"hourCycle"	`hc`	"h12", "h23", "auto"
"measurementUnit"	`mu`	"celsius", "fahrenheit", "auto"
"numberingSystem"	`nu`	"latn", "native", "auto"



# Thank you!

