1. PREAMBLE

This white paper is an attempt to place in a linguistic perspective the issues that had arisen during the Singapore meet. Two documents were circulated to the teams for deliberation on the working days:

1. A series of definitions to be ratified, suitably modified and eventually to be and agreed to
2. A set of questions having partly the definitions as a starting point and which would provide a direction to the though-processes pertinent to each script/language.

This white paper addresses the first issue and tried to arrive at a "unified approach" since unless consensus is arrived at, no progress can be made to meet the target dead-line. However implicitly the white paper also addresses the comments raised in the Questions. The white paper tries to see the issues in a broader perspective and attempts to encompass other writing systems to arrive at a holistic picture.

2. DEFINITIONS (& ALSO QUESTIONS)

The definitions circulated basically pertain roughly to three areas:
UNICODE, DNS & VARIANTS as the appendix shows.
This reflects the direction of the thought processes in developing multi-lingual IDN's starting off with the basic building Block: Unicode and going on to the DNS and finally handling the issues of Variants. Each of these will be treated in turn

2.1. UNICODE

4 definitions are pertinent to Unicode. Grosso modo, all teams seem to be agreed upon these definitions. In any case a character in Unicode is inviolable and once a code point is assigned to it, it cannot be modified or altered in any manner. Each of these definitions will be taken in turn and analyzed, since major issues do arise:

Abstract Character:	**A unit of information used for the organization, control, or representation of textual data. (Unicode Standard, section 3.4, D7)**
Assigned Code Point:	**A mapping from an Abstract Character to a particular Code Point in the code space. See Unicode Standard, section 2.4. Not to be confused with Valid Code Point.**
Code Point:	**A value in the Unicode code space. The meaning here is restricted to meaning D10 in the Unicode Standard, section 3.4.**
Language Character Repertoire:	**A set of Code Points identified by some identifier (such as a tag for identifying language as defined in RFC 5646). The definition of the Language Character Repertoire is ideally performed in a way appropriate to some community of language users, and might colloquially be understood as "the characters used to write a language". In most cases, all the Code Points in a Language Character**

	Repertoire will come from the same Script Table.
Script Table:	**A Script Table is a table of Unicode Code Points all having the same script property value. See Unicode Standard Annex #24.**

Each of these will be taken up in turn and commented

Abstract Character:

The term refers to a unit of information used for the organization, control, or representation of textual data. Basically on the analogy of a Phoneme or a Morpheme, the term is –emic in nature, representing an abstraction which is visualized in the mind of the user of a given language. One way of representing this abstraction would be in IPA, using the / / notation which identifies it as an –emic character.

EXAMPLE: a bilabial unvoiced stop [/p/ in IPA] would be an abstraction which in the native speaker's mind would be realised as a specific shape, associated with that glyph.

DISCUSSION

While the definition which refers to the Section 3.4, D7 of Unicode is correct and valid, attention is drawn to the *draft-ietf-appsawg-rfc3536bis-02,* which defines better the notion of "Abstract Character". An Abstract character is basically a "glyph" a term understood by both font designers and experts working with writing systems. The definition is reproduced below:

```
glyph
A glyph is an abstract form that represents one or more glyph
images. The term "glyph" is often a synonym for glyph image,
which is the actual, concrete image of a glyph representation
having been rasterized or otherwise imaged onto some display
surface. In displaying character data, one or more glyphs may be
selected to depict a particular character. These glyphs are
selected by a rendering engine during composition and layout
processing. <UNICODE>
```

The term ABSTRACT CHARACTER and Glyph seem to be co-terminous and for the purposes of clarity, the term "glyph" with its definition be also introduced to arrive at clarity.

Assigned Code Point:

This seems to be the logical next step. Once an "abstract Character" is defined as shown above a mapping of the glyph is carried out to a particular Code point in code space.

EXAMPLE: The bilabial unvoiced stop /p/ could be realised in a variety of scripts as a code point. Thus in the Arabic Codepage the assigned code point would be: 067E, Devanagari would assign 092A and all other scripts derived from the Brahmi family would have the same code point 0xxA with 0xx representing the requisite offset to the desired code-page. Thus Gujarati would be 0AAA, Bengali 09AA, Gurmukhi 0A2A, Tamil 0BAA and so on.

DISCUSSION

Although Uniqueness is maintained in the assigned code-point and Unicode itself tries to ensure that there is no duplication of a "glyph" which could be termed as allographs": Variants of the same Assigned Character do occur. (This discussion in fact is closely tied to the notion of valid code point (cf. infra).) This can be due to two reasons

a. The two assigned code-points are differentiated by a specific language register, which demands two separate assigned code-points. In the Devanagari code page for example the Abstract Character /æ/ (representing a full vowel in IPA) is assigned two code-points: 090D which caters to Hindi and 0972 which caters to Marathi and Konkani. Similar situations arise in the Arabic code-page where the vowel /i:/ is represented by quite a few assigned code-points: 064A (Arabic) , 06CC (Farsi, Urdu, Kashmiri).
   The discussion of language and script will not be taken up here; but as can be seen Unicode is not very clear in this regard
b. Unicode permits two ways of realising the same code-point. One as a single assigned code-point and the other as a combination of two assigned code-points. Once again Arabic and Devanagari (Bengali, Gurmukhi) are prime candidates. In Arabic the "harakat" "niqqud "or diacritic markers (such as Madda,fatta,damma,kasra) are provided separately as assigned code-points. At the same time, there are assigned code-points for a combination of a vowel+a harkat. To represent the glyph /ā/, Arabic assigns two code-points: 0622 as a unique assigned code-point and the combination of two assigned code-points 0627 + 0653. Devanagari exhibits a similar pattern of two "assigned code-points" for the same glyph. Thus 0958 is co-terminous with 0915+093C. Many other languages including the Latin set exhibit similar possibilities. Unicode resorts to the tried and tested technique of Normalisation where such "allographs" are normalised to a single assigned code-point. Within the browser.
   Normalisation thus is a fertile ground for variants (but more of this later)

Code Point and Valid Code-point:

A logical next step, these are "display representations of the assigned code-point within a given code-page. Since these are closely tied, they are taken together. These are defined as : *A value in the Unicode code space. The meaning here is restricted to meaning D10 in the Unicode Standard, section 3.4.: D10* ***"Code point: Any value in the Unicode codespace.• A code point is also known as a code position."*** The offshoot of this is often termed by Unicode as *the encoded character*

EXAMPLE: The table below tries to make this clear:

Abstract Character:	**/p/**
Assigned Code-point :	Arabic: 067E, Devanagari: 092A All other scripts derived from the Brahmi: The same code point 0xxA with 0xx representing the requisite offset to the desired code-page. Gujarati: 0AAA, Bengali 09AA, Gurmukhi 0A2A Tamil 0BAA
Code-point/Valid code-point : **Encoded Character**	Arabic: 067E : پ Devanagari: 092A : प Gujarati: 0AAA : પ Bengali 09AA : প Gurmukhi 0A2A: ਪ Tamil 0BAA: ப

Table 1

DISCUSSION: The discussion above is pertinent to the issues here, since the Assigned Code-point in turn becomes finally an encoded character after being given a valid code-point. The issues of Allographs and multiple representations of a given code-point are fertile ground for Variants, since Unicode has allowed the same bit of information to be represented in more than one manner. As mentioned above Normalisation becomes a key issue to avoid spoofing and phishing and browsers have to be compliant with Normalisation:

Thus ड़ ड़ look alike whereas the first is a single encoded character (0958) and the second is two assigned code-points realised as 0915+093C.

Language Character Repertoire:

This defined as under:

A set of Code Points identified by some identifier (such as a tag for identifying language as defined in RFC 5646). The definition of the Language Character Repertoire is ideally performed in a way appropriate to some community of language users, and might colloquially be understood as "the characters used to write a language". In most cases, all the Code Points in a Language Character Repertoire will come from the same Script Table.

EXAMPLE: No example is needed since the definition is self-evident.

DISCUSSION: Although the remark that follows deviates from the discussion at hand, it is important that the notion of Language tags become a reality. Quite a few Code-pages support more than one language. Thus code-page 600 for Arabic is a bank from which a considerable number of languages draw their code-points. Three official Indian languages Kashmiri, Sindhi, Urdu use this code-page. Similarly Devanagari supports 9 official Indian languages: Sanskrit, Hindi, Marathi, Maithili, Boro, Dogri, Nepali and Konkani (Sindhi is also written using Devanagari script). The absence of any language tag denies communities using these code-pages from registering their language. Thus a Marathi or a Konkani user would not be able to register a given IDN in case it is already adopted by say Hindi or Dogri. Opening up language tags would allow for a wider proliferation and would let a "hundred flowers bloom" 百花運動

A second important point to be noted is the caveat: *In most cases, all the Code Points in a Language Character Repertoire will come from the same Script Table.* A majority *of IDN's* allow for a mix of Latin and the Language. Thus the policy for Indian languages allows for Hyphen and Digits to be in the Latin Script. IDNs can be in the Indian script and no mixing of Latin and Indian script is allowed within a given domain. Thus bombayमुंबई.भारत is "illegal" whereas Bombay.मुंबई.भारतis valid.

However in the case of three Indian languages Dogri, Bro and to a certain extent Assamese, 02BC ʾ part of the Spacing Modifier letters is needed. 02BC is used in these languages either as a tone or a palatalisation marker. ICANN's policy of mixing of scripts needs to be modified to accommodate these languages. This will become more acute when South Africa which recognises [1]

Script Table:

[1] The English version of the South African constitution refers to the languages by the names in those languages: isiZulu, isiXhosa, Afrikaans, Sepedi (referring to Northern Sotho), Setswana, English, Sesotho (referring to Southern Sotho), Xitsonga, Siswati, Tshivenda and isiNdebele (referring to Southern Ndebele) http://en.wikipedia.org/wiki/Languages_of_South_Africa

This is defined as : *A Script Table is a table of Unicode Code Points all having the same script property value. See Unicode Standard Annex #24.*

EXAMPLE: No example is needed since the definition is self-evident.

DISCUSSION: Cf. the discussion above which is pertinent to this definition.

SUMMING-UP

The following points emerge from this discussion:

1. The term “glyph” be also recognised as coeval to Abstract Character.
2. Normalisation because of multiple assignment be admitted as a definition
3. Browser testing to be undertaken by each team to check whether such normalisation really does occur.
4. Language Tags be implemented.
5. In the case of Devanagari, 02BC be recognised as a valid code-point and be permitted since the same script table does not suffice for Boro, Dogri and Assamese.

2.2. DNS

In toto, there is no objection to the definitions proposed under DNS. Slight issues are present which are treated in the following table. Examples pertinent to Devanagari will be taken. This is also because modifying a DNS is a difficult if not impossible task.

ENTITY	DEFINTION	REMARKS & EXAMPLE
A-label:	**An ASCII-Compatible Encoding form of an IDNA-valid string. It must be a complete label: IDNA is defined for labels, not for parts of them and not for complete domain names. This means, by definition, that every A-label will begin with the IDNA ACE prefix, "xn--", followed by a string that is a valid output of the Punycode algorithm (RFC 3492) and hence a maximum of 59 ASCII characters in length. The prefix and string together must conform to all requirements for a label that can be stored in the DNS including conformance to the rules for LDH labels (See RFC 5390, Section RFC 2.3.1). If and only if a string meeting the above requirements can be decoded into a U-label is it an A-label. (RFC 5890)**	*Valid.* *Rider : the Government policy lays down that the string must be at least 3 characters long.*
Allocation:	**In a DNS context, the first step on the way to Delegation. A registry (the parent side) is managing a zone. The registry makes an administrative association between a string and some entity that requests the string, making the string a label inside the zone, and a candidate for delegation. Allocation does not affect the DNS itself at all.**	*भारत and its localised forms was proposed for 7 Indian languages in different scripts and has passed.*
Delegation:	**In a DNS context, the act of entering parent-side NS (nameserver) records in a zone, thereby creating a subordinate namespace with its own SOA (start of authority) record. See RFC 1034 for detailed discussion of how the DNS name space is broken up into zones.**	*भारत (along with its localised forms) allocated and delegated*
Fundamental Label:	**A U-label that consists only of Valid Code Points. In practice, this is the U-label requested to be registered.**	*The issue of Valid Code points is discussed below in 2.3.*
Fundamental TLD:	**The Fundamental Label form of a Variant TLD Set.**	*No objection to the theoretical construct, but the notion of variant needs to be refined*
IDNA Symmetry	**A-label/U-label transformation must be symmetric:**	*Debate whether Symmetry be*

Constraint:		*replaced by Transitivity*
U-label:	**An IDNA-valid string of Unicode Code Points, in Normalization Form C (NFC) and including at least one non-ASCII character, expressed in a standard Unicode Encoding Form (such as UTF-8). It is also subject to the constraints about permitted characters that are specified in Section 4.2 of RFC 5891 and the rules in the Sections 2 and 3 of RFC 5892, the Bidi constraints in RFC 5893 if it contains any character from scripts that are written right to left, and the IDNA Symmetry Constraint. (RFC 5890)**	*Accepted if the rider of Variants defined below is accepted.*

Table 2

2.3. VARIANTS

The issue of variant will be taken up without having recourse to a separate discussion on each definition provided. This is because of two major issues.

a. The notion of variant and the need to find an alternate term for the same.
b. The notion of preferred variant

In what follows an attempt will be made on the bases of linguistics to propose a "Unified theory" of variants which can accommodate not only the scripts under survey but also all other scripts to come under the opening-up of multilingual IDN's

2.3.1. PROPOSED ALTERNATIVE DEFINTION OF VARIANT

The starting point is to propose an umbrella term for "Variant" which can embrace all and every type of "variant" proposed by the different teams. Following the strategy deployed by Unicode the notion of "Variant" as an "Abstraction" be introduced. The prefix Archi (used in linguistics to define a global concept embracing all realizations for the concept) be used. The term Archivariant or Abstract Variant could be used. Since this "Abstract Variant" is fundamentally "-emic"[2] in nature, a term "Varianteme" could also be deployed to indicate the abstract nature of the concept.

2.3.2 TYPOLOGY

Once the Varianteme/Archivariant/Abstract Variant is accepted, each script/language uses the concept for realizing its own sets of "alternants" which are realizations of the abstract concept within the given language/script. These alternants could be of different kinds. A basic typology is given below.

The starting point of the typology is based on shapes exhibiting close identity (homographs at the display level) vs. Shapes representing Orthographic alternants (Crudely this could be termed as the Homophonic level since spelling reflects the way a spoken language is represented.)

2.3.2.1 Orthographic alternants

[2] The concepts –emic and –etic were introduced by Pike in 1954 and are still prevalent. Cf. http://www.sil.org/~headlandt/ee-intro.htm

Starting off with Orthographic alternants, these could be alternate spellings admitted in a given language. Encyclopaedia vs. Encyclopaedia in English(Greek base vs. simplified version of the same) . हिंदी हिन्दी to represent the language Hindi where the anuswar /ं/ alternates with /न्/ Spelling alternants are problematic since in a majority of languages, such variants are not regularized and hence it is very difficult to treat them as “alternants” of the same Archivariant. In languages such as Russian, unless the stress marker is placed the same word can lend itself to two different meanings: замок: "замок" - has two different meanings. If the accent falls on letter "a", it means a castle, if the accent falls on letter "o", it means a lock. Similar issues arise in the much quoted and discussed: β vs. ss as in fussball.de and fuβball.de. Spelling variants as has been practiced by the USA are best left alone. Both www.realise.com and www.realize.com are allowed and refer to two alternative web-pages[3]

2.3.2.2 Display alternants:

These are alternants which are at the display level. In the small point-size of the URL these look alike and can lead to spoofing and phishing. Since these are closely tied to the different scripts, the Archivariant gets “realized” in different manners. Without being totally accurate, a typology based on the writing system is attempted below:

SCRIPT TYPE	DEFINITION	POSSIBLE ALTERNANTS	EXAMPLAR SCRIPTS	EXAMPLE
Abjads Consonant Alphabets	Derived from the first 4 letters of the alphabet, abjads give priority to the consonants, especially the tri-consonantal root system of Semitic Vocalic modifiers are few such as the long vowels of which some can figure also as Consonants being distinguished from Vowels by their position. Other vocalic markers termed as harakat or niqqud are optional and not often used in writing.	1.Presence and absence of Harakats 2. Similar looking characters because of adoption of Arabic by a large number of languages and eventual identity of shapes . 3, Positional Variants	Semitic Scripts Arabic Hebrew	اردو vs. اُردو ي (064A) ى (0649) ى (06CC) كيل

[3] www.color.com and www.colour.com are however conflated to a single home-page

		4.Possibility of displaying the same glyph in two different ways : a single code point vs. a combination of two code points 5. Use of ZWNJ for Farsi[4]		کیل کیل اَ + ـَ آ کوهها - کوه kuhha - kuh Mountains -Mountain
Alphabets	Derived from the first two letters of the Greek alphabet, Alphabetic writing systems are a set of letters representing consonants and vowels which attempt to represent the spoken language. Since writing systems remain static whereas Spoken language evolves, there is usually a marked gap between the alphabet and the language it represents.	1. Many to one mapping 2 different ways of representing the same sound	English French Basically all Latin and Latin1 scripts Cyrillic Greek	ss, β both represent the sound /s/ in German German ü vs ue as in max-müller.de and max-mueller.de

[4] The plural markers added to all nouns in Persian take the stress and renders the specific plural. When the final consonant of the noun is orthographically connective, the plural suffix is usually joined directly, as کتابها although it is sometimes left separate as ها کتاب. (book-books) cf. http://www.ijar.lit.az/pdf/6/2010(4-52).pdf

		3.Unicode permits the same character to be generated in two ways		n+~ in Spanish and ñ
Abugidas Alphasyllabaries Akshara	Derived from Ethiopic, the term roughly means a writing system where the basic element is a syllable having a consonant and a vowel as its nucleus and followed but not necessarily by satellites such as Vowel Modifiers, nasalisers, Lengtheners, the all in a specific order defined by a Backus-Naur formalism which ensures a well-formed syllable	1.Syllabic clusters which look alike in the 10 point URL. The policy developed for Indic scripts does not allow for alternants between single consonants but rather alternants at the ligatural level where distinctions are difficult. 2. Normalisation of characters because the same glyph can be displayed in two different ways : a single code point vs. a combination of two code points 3. use of Zero width Joiners/Non-joiners a. Needed for generating out shapes not available in	1. Brahmi based writing systems such as Hindi, Malayalam, Dogri, Gurmukhi, Bengali ,Java, Bali, Thai etc 2.Ethiopic	द्र द्र द्र dga dna dra क़ क़=क+़ र+्++य=र्य Ra+halanta+zwj+ya

		Unicode (eyelash ra) b. Needed for root words ending in halanta to which a suffix is apposed as in Nepali		टेबल+्++कु=टेबल्कु Tebal+Halanta+zwnj+ku
Syllabaries	Distinct from Abugidas in the sense that syllabaries are a set of symbols where each symbol represents a combination of the consonant set with the vowel set of the language. The basic "barakhadi" or basic syllables of Indian scripts and the Tamil writing system are syllabary based to a certain extent.	1. Alternants where two shapes are practically alike 2. In the case of Japanese syllabaries, absence or presence of the voicing marker above the consonant	1. Tamil to a large extent with the possible exception of a couple of conjuncts such as "shri" 2. Japanese Kana: Hiragana, Katakana	Tamil ன �ன Short/Long e ஒ ஓ Short/Long o Japanese: カ ka ガ ga
Semantic writing systems	Originating as pictures to represent an object/concept in the real world (pictograms) , these writing systems became more and more stylized and became either ideograms or compound characters in which a "power sign" comprising the semantic element is joined to a phonetic element that hints at the pronunciation	The complexity of the strokes which can range from 1 stroke (Number 1) to 48 (3 dragons) creates a considerable amount of homographic alternants	Chinese Zhōngwén Japanese Nihongo	膀 膀 11 point resolution

Table 3

The table below sums up the issues involved in relation to each of the writing systems and also the notion of “preferred variant” (indicated by a “P”)

ISSUE	ABJAD	ALPHABETS	ALPHASYLLABARIES	SYLLABARIES	KANJI/IDEO-SEMANTIC
UNICODE ISSUES: 1. idna 2003 vs. 2008 Issues 2. Legacy inputting	Not all handled Harakats need normalization Laam+alif= laam-alif P NONE	OE Œ French sœur soeur P NONE	Not all handled Needs to be tested in browsers. Hence Normalisation is a must क़ क़=क+़ P Needed for generating characters that did not exist earlier in Unicode but are present in Unicode 5.2 onwards Chillus, अॅ/ae/. Normally ZWJ/ZWNJ was used	NONE FOR HIRAGANA & KATAKANA	NONE
ZWJ/ZWNJ ISSUES	Normally not used but could be used to create display variants. Used in Farsi for plural suffixes. Can be rule-driven کوه - کوهها	Not needed and can create spoofing issue	1. Used after a “ schwa Killer” /halanta/ to generate out characters not still available in Unicode: र+्++य=र्य Ra+halanta+zwj+ya 2. Used to combine a	NONE	-

			root word ending in a halanta to its suffix as in Nepali: टेबल+्++कु=टेबल्कु Tebal+Halanta+zwnj+ku		
TRUE HOMOGRAPHS	Positional variants especially In the case of “ye choti” 06CC كيل 064A كيل 0649 كيل	NONE	NONE	NONE	-
LOOK ALIKES BECAUSE OF SMALL POINT SIZE OF URL	ں ن Noon vs Nasalisation marker	l 1 vv w	1.Single syllable vs. Conjunct त - त्त t vs tta 2.Conjunct-Conjunct द्र द्र द्र dga dra dna	Voiceless vs. Voiced: カ ka vs. ガ ga	膀 VS 膀
ALTERNANTS WHICH CANNOT BE MAPPED TO THE SAME CODE-POINT SINCE THEY ARE NOT IN FREE VARIATION BUT IN OPPOSITION	NONE	German ss-ß Opposition in Büssen vs Büßen	NONE	NONE	NONE

CASE MARKING	NOT APPLICABLE	YES	NOT APPLICABLE	NOT APPLICABLE	NOT APPLICABLE
ORTHOGRAPHIC ALTERNANTS	-	Geo-linguistic variants -se -ze	गर्दन गरदन हिंदी हिन्दी	NONE	POSSIBLE
HANDLING OF SPACE	To ensure that two words which constitute a name do not join together in the Abjad Family normally a space is used to separate the two words e.g. Mango pickle (Urdu) آم اچار vs. آماچار This issue can be solved by using a hyphen instead of a ZWNJ which is not the case of Farsi plurals pointed out above	NOT APPLICABLE	NOT APPLICABLE	NOT APPLICABLE	NOT APPLICABLE
BROWSER ISSUES 1. THE BROWSER USES THE SYSTEM FONT TO DISPLAY THE URL. 2.THE RENDERING ENGINE USP10.DLL	Arial (Body CS) in IE Studies need to be undertaken on the	Times New Roman In IE Studies need to be undertaken on the	Mangal for Hindi in IE Studies have been undertaken on the	MingLiu Studies need to be undertaken	MS Mincho Studies need to be undertaken

/ICU/PANGO DETERMINES THE SHAPE OF COMPLEX SCRIPTS SUCH AS DEVANAGARI/ARABIC	rendering engine	rendering engine	rendering engine which creates ill-formed aksharas depending on the USP10.dll E.g. Ra+halanta+aa is illegal since a halanta cannot precede a Vowel and yet it is permissible with usp10.dll र्+आ ार्[5]		
3. THE BROWSER DOES NOT HANDLE IDNA 2008	Studies need to be undertaken	Studies need to be undertaken	Studies need to be undertaken	Studies need to be undertaken	Studies need to be undertaken

TABLE 4

The table below summarises the close interrelations between Unicode and Linguistic Issues No examples are given since the issues have been handled in Tables 3 and 4

[5] The case of विश्व vs. विश्व the former used in Konkani, Marathi, Sanskrit Maithili & Boro, whereas the latter is used in Hindi, Nepali, Dogri is interesting since the native browser of Windows will permit only the first and the second (artificially generated out through a ZWJ) is mapped to the same address in Chrome, Firefox and IE8
विश्व http://www.xn--y2bac9a9d.com/ विश्व http://www.xn--y2bac9a9d.com/

Writing system	UNICODE		LINGUISTIC ISSUES		
	Code-points (P) (H)	Legacy Inputting (P) (H)	Positional (P) (H)	Spell-variants at character level	Look-alikes
Abjad	YES	NO	YES	YES	YES
Alphabet	NO	NO	NO	YES	YES
Akshara	YES	YES	NO	NO	YES
Syllabaries	NO	NO	NO	YES	NO
Phonetic-Semantic	NO	NO	NO	NO	YES

TABLE 5

P-> Preferred Variant H-> Homograph

2.3.3. THE PROBLEM OF THE PREFERRED VARIANT

Two issues arise here:

a. The variant table
b. Activated vs. Non-activated variants

Adopting the notion of the archivariant provides a smooth solution to the above two problems

1. The Archivariant shall be the "abstract variant". Each script will realize this archivariant by means of alternants functional within the language (cf. Table 2 above). The notion of a preferred variant is determined solely by context. The following example will illustrate the issue:

The word कतर represents a country name Qatar. Given that the alternate form for ta /त/ is tta /त्त/; in the case of कतर the preferred variant /alternant would be ta /त/ and NOT tta /त्त/. On the other hand given kattar /कत्तर/ which means hairband or braid; the preferred variant/alternant would be tta /त्त/ and not ta /त/.

The notion of preferred variant/alternant is therefore as in the case of Phonology of a language, context-bound. One variant/alternant will be the preferred alternant and the remaining (up to 3) would be the non-preferred alternants. The structure is dynamic and given a pair or a triad or a "quatuor" of alternants, one will be the preferred one (the one first chosen) and the other(s) will be the non-preferred ones. It is now left to the registrar to permit the first "bidder" to keep both variants (fiscals to be decided) or permit one and block the other i.e. not activate it. [6] Under the policy developed by the Government of India, it is felt that in the case of TLD's/GTLD's/reserved names/sensitive names all the possible generated variants be activated and bundled or conflated to a single URL, whereas in other cases the policy to conflate/bundle is to be decided

Table 5 illustrates this

Case of mudrā मुद्रा Only for this word is द्र activated

ARCHIVARIANT	ALTERNANTS	PREFERRED "VARIANT"	NON-PREFERRED VARIANT TO BE ACTIVATED/RESERVED LEFT TO POLICY

[6] The case of color/colour mapping to the same webpage is an example where the client bought rights to both and "bundled" the two alternants in one.

AKSHARA: CLOSE LOOK-ALIKES	द्र, द्र, द्र dra,dga,dna	द्र dra	द्र,द्र dga, dna
AKSHARA			

Table 6

2. Preferred Variants are possible only in the case of IDNA where 2 forms are reduced to one which is the preferred variant. Thus in the case of Hindi क़ क़=क+़, the preferred variant is 0958 क़ and not 0915+93C क़ and the browser automatically reduces क़ (क+़,) to क़

2.3.4. FINAL DEFINITIONS

The above discussion allows for coining of the following definitions:

ARCHIVARIANT: The term refers to refers to a unit of information used for the organization, control, or representation of variants which dependent on the writing system can be orthographic, homographic or close look-alikes in nature, this to prevent ambiguity and/or misuse of an IDN in a given script/language. The archivariant is thus an omnibus term .

ASSIGNED VARIANT/ALTERNANT: The term refers to the graphemic realisation of an Archivariant within the writing system of a given script/language and which can be orthographic, homographic or a close look-alike within that particular system.

E.g. In the case of Devanagari used for Hindi Script/Language) three kinds of alternants/assigned variants can be identified: Pure Homographs, close look-alikes (cf. 2.3.2 and 2.3.3. supra)

PREFERRED VARIANT/ALTERNANT : The term refers to an assigned variant which is preferred among other coeval assigned variants within a given writing system. In a majority of writing systems the Preferred variant are pure homographs and the preferred variant shall be that where IDNA 2003-2008 et seq. rules that given a set of assigned variants, one variant shall be preferred to the other.

E.g. In Hindi the voiced flap ड़ 0921+093C and ड़ 095C are both reduced (normalised) to ड़ 095C, which would be deemed as a preferred variant/alternant to the exclusion of 0921+093C which will be the non-preferred variant/alternant

DYNAMIC VARIANT/ALTERNANT: The term refers to a set of assigned variants preferably within a writing system which are coeval in nature and whose distribution is governed by a given context within which and which alone one variant shall be deemed to be Preferred. The determination of such a variant shall depend on the first choice made by a user and which shall automatically exclude all other assigned variants belonging to that particular set.

TYPE	ALTERNANTS	PREFERRED "VARIANT"	NON-PREFERRED VARIANT TO BE ACTIVATED/RESERVED LEFT TO POLICY
Case of मुद्रा	द्र, द्र, द्र dra,dga,dna	द्र dra is chosen by the user	द्र,द्र dga, dna

NON-PREFERRED VARIANT: The term refers to a Variant/alternant which from a set of assigned variants is deemed as that which shall not be used. Non-preferred variants are of two kinds: Homographic assigned variants generated out by IDNA where a given assigned variant is deemed "non-preferable: and thereby reduced to its partner variant, and Contextual variants where within a given context, only one variant shall be deemed as preferred and all other variant(s) shall be deemed as Non-Preferred.

e.g.

TYPE	ALTERNANTS	PREFERRED "VARIANT"	NON-PREFERRED VARIANT TO BE ACTIVATED/RESERVED LEFT TO POLICY
HOMOGRAPHIC Case of पेड़	ड़ 0921+093C and ड़ 095C	ड़ 095C 0921+093C automatically reduced to 095C	ड़ 0921+093C
Case of मुद्रा	द्र, द्र, द्र dra,dga,dna	द्र dra is chosen by the user	द्र,द्र dga, dna

SUMMING-UP

1. "Variants" are a necessary evil to curb phishing, spoofing and scamming or even for that matter cyber-squatting.
2. The term Variant as defined and all the adjuncts of that term need to be reviewed.

   PROPOSALS

   a. The notion of variant be replaced by an abstract construct which needs to be named. Proposed name: Archivariant/Varianteme
   b. Variants be redefined as Assigned variant/Alternant to refer to entities which are functional within a given script/language.
   c. The Archivariant generates out depending on the script and its writing system structure a set of possible alternants defined by close identity or by reason of Unicode allowing more than one way of representation
   d. The notion of preferred/non-preferred variant be suitably emended
   e. Where a single alternant is valid and the other alternant(s) do not fit in the context i.e. are in exclusion, the "best fit" alternant be chosen and the others be either proposed or reserved as the case may be. This is the case of normalisation as proposed in IDNA 2008
   f. Where two or more alternants can apply the choice be left to the first user and the other either blocked or proposed for bundling.
   g. Browser behaviour be suitably studied.
   h. The RFC needs to be suitably emended to meet the requirement of other writing systems.

APPENDIX I

The appendix covers both the questions as well as the definitions and sorts them on three major themes: Unicode/DNS/Variants

TYPE	ENTITY	DEFINITION	REMARKS
DNS	**A-label:**	**An ASCII-Compatible Encoding form of an IDNA-valid string. It must be a complete label: IDNA is defined for labels, not for parts of them and not for complete domain names. This means, by definition, that every A-label will begin with the IDNA ACE prefix, "xn--", followed by a string that is a valid output of the Punycode algorithm (RFC 3492) and hence a maximum of 59 ASCII characters in length. The prefix and string together must conform to all requirements for a label that can be stored in the DNS including conformance to the rules for LDH labels (See RFC 5390, Section RFC 2.3.1). If and only if a string meeting the above requirements can be decoded into a U-label is it an A-label. (RFC 5890)**	*Available in the white paper*
DNS	**Allocation:**	**In a DNS context, the first step on the way to Delegation. A registry (the parent side) is managing a zone. The registry makes an administrative association between a string and some entity that requests the string, making the string a label inside the zone, and a candidate for delegation. Allocation does not affect the DNS itself at all.**	*Available in the white paper*
DNS	**Delegation:**	**In a DNS context, the act of entering parent-side NS (nameserver) records in a zone, thereby creating a subordinate namespace with its own SOA (start of authority) record. See RFC 1034 for detailed discussion of how the DNS name space is broken up into zones.**	*Available in the white paper*

DNS	**Fundamental Label:**	**A U-label that consists only of Valid Code Points. In practice, this is the U-label requested to be registered.**	*Available in the white paper*
DNS	**Fundamental TLD:**	**The Fundamental Label form of a Variant TLD Set.**	*Available in the white paper*
DNS	**IDNA Symmetry Constraint:**	**A-label/U-label transformation must be symmetric:**	*Available in the white paper*
DNS	**U-label:**	**An IDNA-valid string of Unicode Code Points, in Normalization Form C (NFC) and including at least one non-ASCII character, expressed in a standard Unicode Encoding Form (such as UTF-8). It is also subject to the constraints about permitted characters that are specified in Section 4.2 of RFC 5891 and the rules in the Sections 2 and 3 of RFC 5892, the Bidi constraints in RFC 5893 if it contains any character from scripts that are written right to left, and the IDNA Symmetry Constraint. (RFC 5890)**	*Available in the white paper*