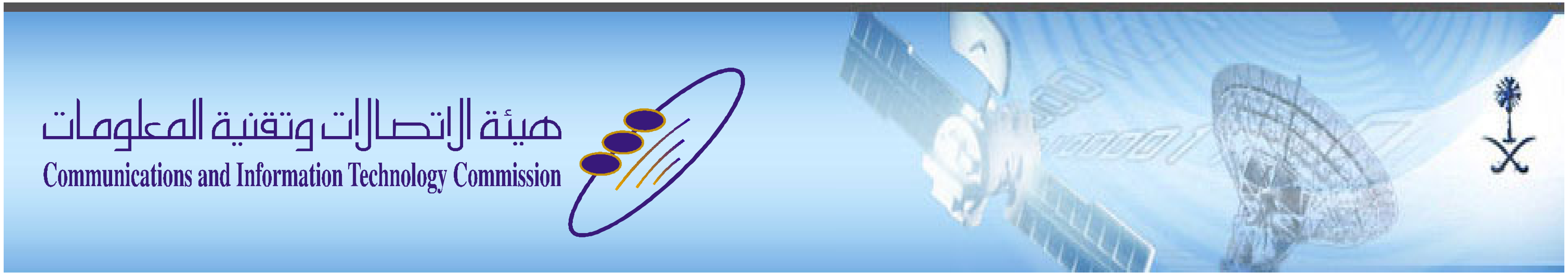


هيئة الاتصالات وتقنية المعلومات

Communications and Information Technology Commission

SaudiNIC's Proposed Solution

Registry-level Multilingual Arabic Script IDN Registration

MENOG 8, Khobar, May 14-18, 2011

Presented by:

AbdulRahman Al-Ghadir

Developed by SaudiNIC/CITC Staff:

AbdulAziz Al-Zoman

Raed Al-Fayez

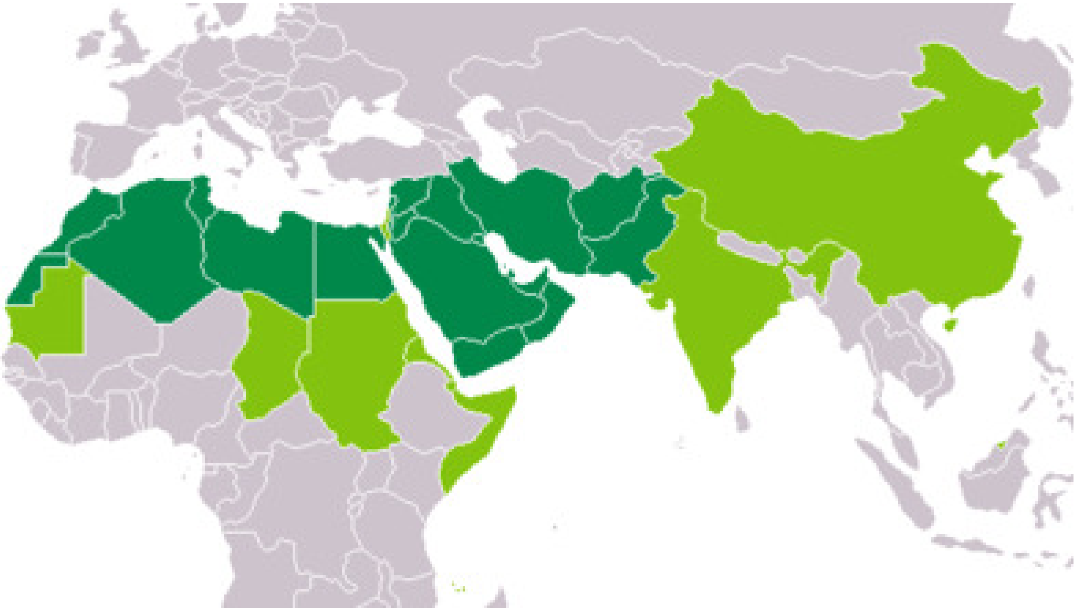
AbdulRahman Al-Ghadir

Content

- **Arabic Script Major Issues**
- **Confusing Similar Characters**
- **Proposed solution**
  - Characteristics
  - Language-level required tables
  - Language-level required tables
  - Language-level operation
- **Conclusion**

Arabic Script

- The 2nd most widely used alphabetic writing system in the world
- Used by many languages such as:
  - Persian, Urdu, Turkish, Kurdish, Pashto, Jawi, …

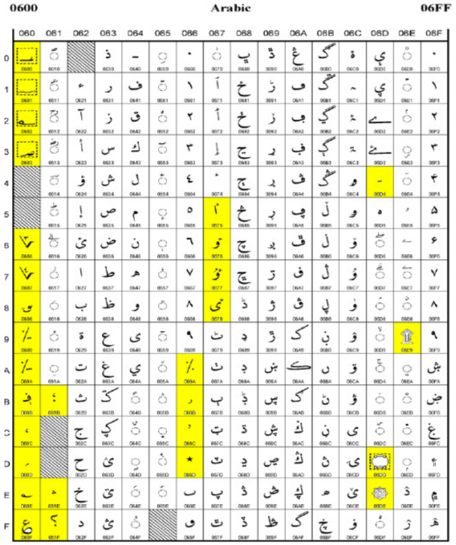


▪Source: http://en.wikipedia.org/wiki/Arabic_script

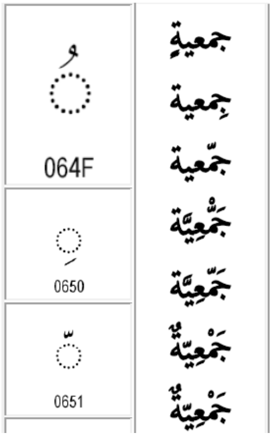
Arabic Script IDN - Major Issues

IDNA Table

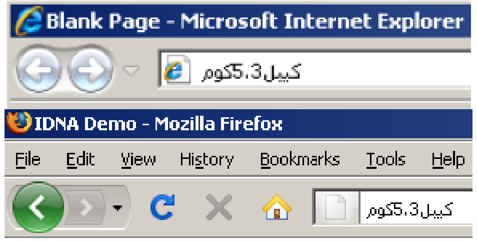
Pvalid, Disallowed, ContextO


0600
Arabic
06FF

Non-spacing Marks



bidirectional


Blank Page - Microsoft Internet Explorer

كيبل5.3كوم

IDNA Demo - Mozilla Firefox

كيبل5.3كوم

Digit

1.European digits	U+0030 .. U+0039	(0123456789)
2.Arabic-Indic digits	U+0660 .. U0669	(٠١٢٣٤٥٦٧٨٩)
3.Eastern Arabic-Indic digits	U+06F0 .. U+06F9	(۰۱۲۳۴۵۶۷۸۹)

ZWNJ/ZWJ

Examples not using ZWNJ	Examples not using ZWNJ
طبل	طبل

input[0] = U+0637
input[1] = U+0628
input[2] = U+0644
	input[0] = U+0637
	input[1] = U+200c
	input[2] = U+0628
	input[3] = U+0644

Combining Marks

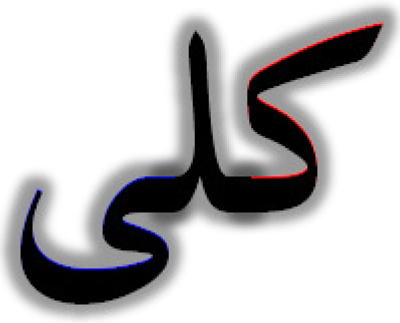
ى	+	◌ٔ	=	ىٔ ىٔـ ـىٔـ ـىٔ	is confusing with	ئ ئـ ـئـ ـئ
U+0649		U+0654		U+0649 U+0654		U+0626
Description: Alef Maksura + Hamza Above <> Yeh With Hamza Above						
Comments: This is a Unicode confusable!						

ی	+	◌ٔ	=	یٔ یٔـ ـیٔـ ـیٔ	is confusing with	ئ ئـ ـئـ ـئ
U+06cc		U+0654		U+06cc U+0654		U+0626
Description: Farsi Yeh + Hamza Above <> Yeh With Hamza Above						
Comments: This is Unicode confusable!						

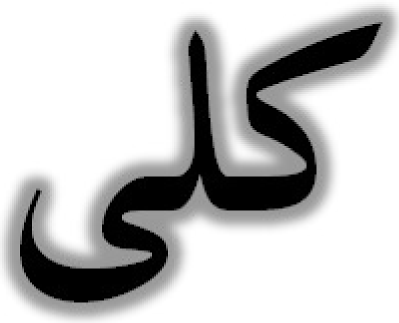
Arabic Script IDN - Major Issues

Confusing Similar Characters

- There are a number of groups of characters that have the same shapes (Homoglyph).
  - eg. Kaf, Heh, Yeh, Alef, … groups

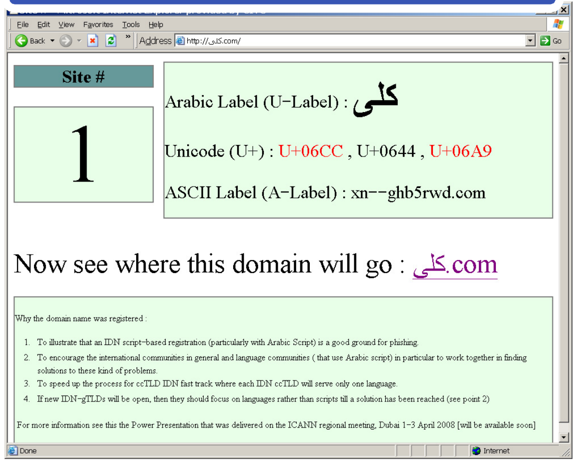


input[0] = U+06a9
input[1] = U+0644
input[2] = U+06cc



input[0] = U+0643
input[1] = U+0644
input[2] = U+0649

کلی.com



Site #	
1	Arabic Label (U-Label) : کلی Unicode (U+) : U+06CC , U+0644 , U+06A9 ASCII Label (A-Label) : xn--ghb5rwd.com

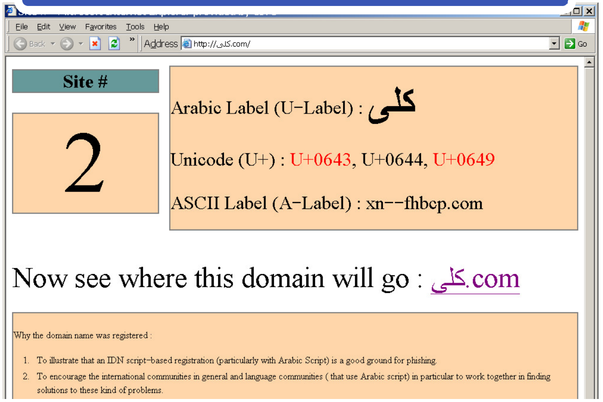
Now see where this domain will go : کلی.com

Why the domain name was registered :

1. To illustrate that an IDN script-based registration (particularly with Arabic Script) is a good ground for phishing.
2. To encourage the international communities in general and language communities (that use Arabic script) in particular to work together in finding solutions to these kind of problems.
3. To speed up the process for ccTLD IDN fast track where each IDN ccTLD will serve only one language.
4. If new IDN-gTLDs will be open, then they should focus on languages rather than scripts till a solution has been reached (see point 2)

For more information see this the Power Presentation that was delivered on the ICANN regional meeting, Dubai 1-3 April 2008 [will be available soon]

كلى.com



Site #	
2	Arabic Label (U-Label) : كلى Unicode (U+) : U+0643, U+0644, U+0649 ASCII Label (A-Label) : xn--fhbcp.com

Now see where this domain will go : كلى.com

Why the domain name was registered :

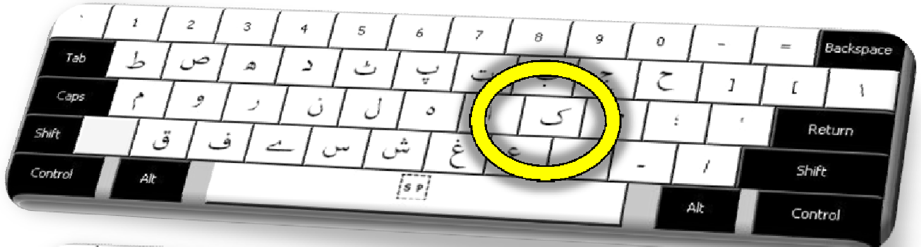
1. To illustrate that an IDN script-based registration (particularly with Arabic Script) is a good ground for phishing.
2. To encourage the international communities in general and language communities (that use Arabic script) in particular to work together in finding solutions to these kind of problems.

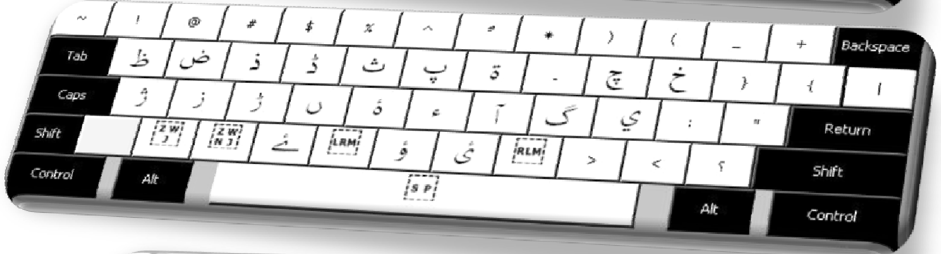
0600 **Arabic** **06FF**

	060	061	062	063	064	065	066	067	068	069	06A	06B	06C	06D	06E	06F
0	؀ 0600	◌ؐ 0610	▨	ذ 0630	ـ 0640	◌ِ 0650	٠ 0660	◌ٰ 0670	ڀ 0680	ڐ 0690	ڠ 06A0	ڰ 06B0	ۀ 06C0	ې 06D0	◌۠ 06E0	۰ 06F0
1	؁ 0601	◌ؑ 0611	ء 0621	ر 0631	ف 0641	◌ّ 0651	١ 0661	ٱ 0671	ځ 0681	ڑ 0691	ڡ 06A1	ڱ 06B1	ہ 06C1	ۑ 06D1	◌ۡ 06E1	۱ 06F1
2	؂ 0602	◌ؒ 0612	آ 0622	ز 0632	ق 0642	◌ْ 0652	٢ 0662	ٲ 0672	ڂ 0682	ڒ 0692	ڢ 06A2	ڲ 06B2	ۂ 06C2	ے 06D2	◌ۢ 06E2	۲ 06F2
3	؃ 0603	◌ؓ 0613	أ 0623	س 0633	ك 0643	◌ٓ 0653	٣ 0663	ٳ 0673	ڃ 0683	ړ 0693	ڣ 06A3	ڳ 06B3	ۃ 06C3	ۓ 06D3	◌ۣ 06E3	۳ 06F3
4	▨	◌ؔ 0614	ؤ 0624	ش 0634	ل 0644	◌ٔ 0654	٤ 0664	ٴ 0674	ڄ 0684	ڔ 0694	ڤ 06A4	ڴ 06B4	ۄ 06C4	۔ 06D4	◌ۤ 06E4	۴ 06F4
5	▨	◌ؕ 0615	إ 0625	ص 0635	م 0645	◌ٕ 0655	٥ 0665	ٵ 0675	څ 0685	ڕ 0695	ڥ 06A5	ڵ 06B5	ۅ 06C5	ە 06D5	ۥ 06E5	۵ 06F5
6	؆ 0606	◌ؖ 0616	ئ 0626	ض 0636	ن 0646	◌ٖ 0656	٦ 0666	ٶ 0676	چ 0686	ږ 0696	ڦ 06A6	ڶ 06B6	ۆ 06C6	◌ۖ 06D6	ۦ 06E6	۶ 06F6
7	؇ 0607	◌ؗ 0617	ا 0627	ط 0637	ه 0647	◌ٗ 0657	٧ 0667	ٷ 0677	ڇ 0687	ڗ 0697	ڧ 06A7	ڷ 06B7	ۇ 06C7	◌ۗ 06D7	◌ۧ 06E7	۷ 06F7
8	؈ 0608	◌ؘ 0618	ب 0628	ظ 0638	و 0648	◌٘ 0658	٨ 0668	ٸ 0678	ڈ 0688	ژ 0698	ڨ 06A8	ڸ 06B8	ۈ 06C8	◌ۘ 06D8	◌ۨ 06E8	۸ 06F8
9	؉ 0609	◌ؙ 0619	ة 0629	ع 0639	ى 0649	◌ٙ 0659	٩ 0669	ٹ 0679	ډ 0689	ڙ 0699	ک 06A9	ڹ 06B9	ۉ 06C9	◌ۙ 06D9	۩ 06E9	۹ 06F9
A	؊ 060A	◌ؚ 061A	ت 062A	غ 063A	ي 064A	◌ٚ 065A	٪ 066A	ٺ 067A	ڊ 068A	ښ 069A	ڪ 06AA	ں 06BA	ۊ 06CA	◌ۚ 06DA	◌۪ 06EA	ۺ 06FA
B	؋ 060B	؛ 061B	ث 062B	ػ 063B	◌ً 064B	◌ٛ 065B	٫ 066B	ٻ 067B	ڋ 068B	ڛ 069B	ګ 06AB	ڻ 06BB	ۋ 06CB	◌ۛ 06DB	◌۫ 06EB	ۻ 06FB
C	، 060C	▨	ج 062C	ؼ 063C	◌ٌ 064C	◌ٜ 065C	٬ 066C	ټ 067C	ڌ 068C	ڜ 069C	ڬ 06AC	ڼ 06BC	ی 06CC	◌ۜ 06DC	◌۬ 06EC	ۼ 06FC
D	؍ 060D	▨	ح 062D	ؽ 063D	◌ٍ 064D	◌ٝ 065D	٭ 066D	ٽ 067D	ڍ 068D	ڝ 069D	ڭ 06AD	ڽ 06BD	ۍ 06CD	۝ 06DD	◌ۭ 06ED	۽ 06FD
E	؎ 060E	؞ 061E	خ 062E	ؾ 063E	◌َ 064E	◌ٞ 065E	ٮ 066E	پ 067E	ڎ 068E	ڞ 069E	ڮ 06AE	ھ 06BE	ێ 06CE	۞ 06DE	ۮ 06EE	۾ 06FE
F	؏ 060F	؟ 061F	د 062F	ؿ 063F	◌ُ 064F	▨	ٯ 066F	ٿ 067F	ڏ 068F	ڟ 069F	گ 06AF	ڿ 06BF	ۏ 06CF	◌۟ 06DF	ۯ 06EF	ۿ 06FF

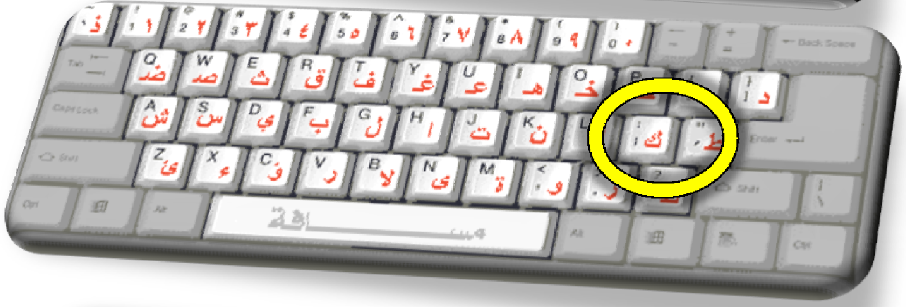
Confusing Similar Characters
Language-based Keyboards

06A9

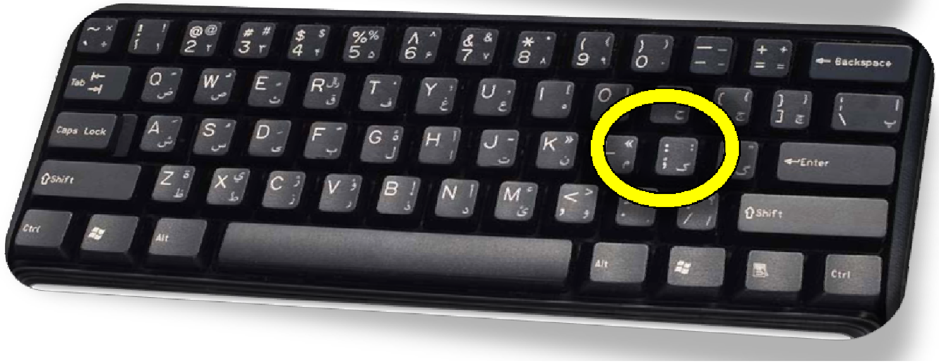




0643



06A9



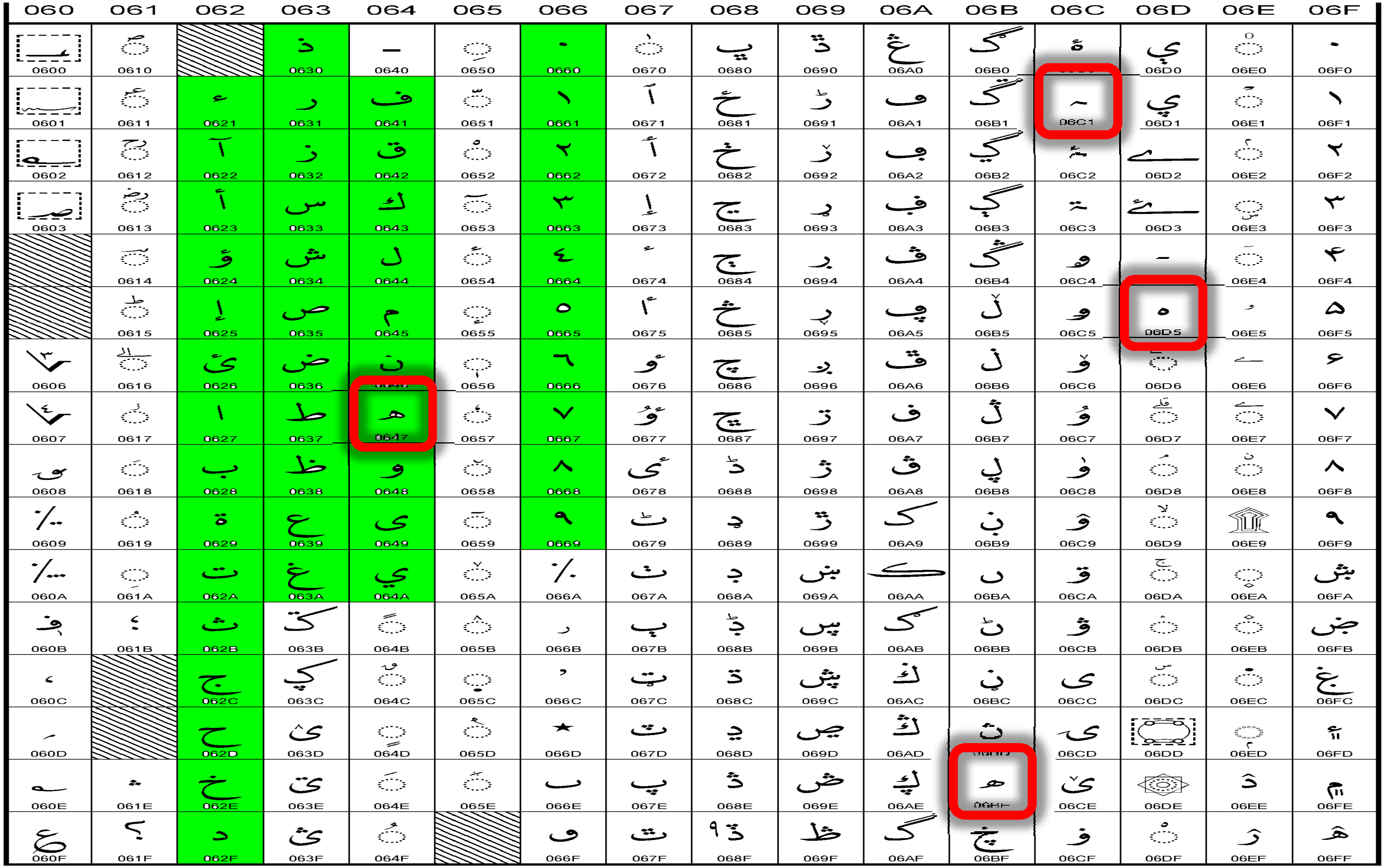
Confusing Similar Characters
Summary of Problems/Challenges

- Security issues (stability, trust,…) e.g. phishing
  - Some should be addressed at language level first
- Input devices (keyboards) are based on languages
- Not all Arabic-script languages are ready:
  - Not widely/commonly used
  - Language community are not ready
  - Hard to make decisions on behave of other language communities
  - Pressure to start with ready languages
- Many problems have been escalated from the protocol to be handled by the registry (e.g. variants, bundling ..etc)
- … and yet has to provide a simple and transparent registration services

Confusing Similar Characters

Example of Variant Characters

0600 Arabic 06FF



	060	061	062	063	064	065	066	067	068	069	06A	06B	06C	06D	06E	06F
0	؀ 0600	◌ؐ 0610	▨	ذ 0630	ـ 0640	◌ِ 0650	٠ 0660	◌ٰ 0670	ڀ 0680	ڐ 0690	ڠ 06A0	ڰ 06B0	ۀ 06C0	ې 06D0	◌۠ 06E0	۰ 06F0
1	؁ 0601	◌ؑ 0611	ء 0621	ر 0631	ف 0641	◌ّ 0651	١ 0661	ٱ 0671	ځ 0681	ڑ 0691	ڡ 06A1	ڱ 06B1	ہ 06C1	ۑ 06D1	◌ۡ 06E1	۱ 06F1
2	؂ 0602	◌ؒ 0612	آ 0622	ز 0632	ق 0642	◌ْ 0652	٢ 0662	ٲ 0672	ڂ 0682	ڒ 0692	ڢ 06A2	ڲ 06B2	ۂ 06C2	ے 06D2	◌ۢ 06E2	۲ 06F2
3	؃ 0603	◌ؓ 0613	أ 0623	س 0633	ك 0643	◌ٓ 0653	٣ 0663	ٳ 0673	ڃ 0683	ړ 0693	ڣ 06A3	ڳ 06B3	ۃ 06C3	ۓ 06D3	◌ۣ 06E3	۳ 06F3
4	▨	◌ؔ 0614	ؤ 0624	ش 0634	ل 0644	◌ٔ 0654	٤ 0664	ٴ 0674	ڄ 0684	ڔ 0694	ڤ 06A4	ڴ 06B4	ۄ 06C4	۔ 06D4	◌ۤ 06E4	۴ 06F4
5	▨	◌ؕ 0615	إ 0625	ص 0635	م 0645	◌ٕ 0655	٥ 0665	ٵ 0675	څ 0685	ڕ 0695	ڥ 06A5	ڵ 06B5	ۅ 06C5	ە 06D5	ۥ 06E5	۵ 06F5
6	؆ 0606	◌ؖ 0616	ئ 0626	ض 0636	ن 0646	◌ٖ 0656	٦ 0666	ٶ 0676	چ 0686	ږ 0696	ڦ 06A6	ڶ 06B6	ۆ 06C6	◌ۖ 06D6	ۦ 06E6	۶ 06F6
7	؇ 0607	◌ؗ 0617	ا 0627	ط 0637	ه 0647	◌ٗ 0657	٧ 0667	ٷ 0677	ڇ 0687	ڗ 0697	ڧ 06A7	ڷ 06B7	ۇ 06C7	◌ۗ 06D7	◌ۧ 06E7	۷ 06F7
8	؈ 0608	◌ؘ 0618	ب 0628	ظ 0638	و 0648	◌٘ 0658	٨ 0668	ٸ 0678	ڈ 0688	ژ 0698	ڨ 06A8	ڸ 06B8	ۈ 06C8	◌ۘ 06D8	◌ۨ 06E8	۸ 06F8
9	؉ 0609	◌ؙ 0619	ة 0629	ع 0639	ى 0649	◌ٙ 0659	٩ 0669	ٹ 0679	ډ 0689	ڙ 0699	ک 06A9	ڹ 06B9	ۉ 06C9	◌ۙ 06D9	۩ 06E9	۹ 06F9
A	؊ 060A	◌ؚ 061A	ت 062A	غ 063A	ي 064A	◌ٚ 065A	٪ 066A	ٺ 067A	ڊ 068A	ښ 069A	ڪ 06AA	ں 06BA	ۊ 06CA	◌ۚ 06DA	◌۪ 06EA	ۺ 06FA
B	؋ 060B	؛ 061B	ث 062B	ػ 063B	◌ً 064B	◌ٛ 065B	٫ 066B	ٻ 067B	ڋ 068B	ڛ 069B	ګ 06AB	ڻ 06BB	ۋ 06CB	◌ۛ 06DB	◌۫ 06EB	ۻ 06FB
C	، 060C	▨	ج 062C	ؼ 063C	◌ٌ 064C	◌ٜ 065C	٬ 066C	ټ 067C	ڌ 068C	ڜ 069C	ڬ 06AC	ڼ 06BC	ی 06CC	◌ۜ 06DC	◌۬ 06EC	ۼ 06FC
D	؍ 060D	▨	ح 062D	ؽ 063D	◌ٍ 064D	◌ٝ 065D	٭ 066D	ٽ 067D	ڍ 068D	ڝ 069D	ڭ 06AD	ڽ 06BD	ۍ 06CD	۝ 06DD	◌ۭ 06ED	۽ 06FD
E	؎ 060E	؞ 061E	خ 062E	ؾ 063E	◌َ 064E	◌ٞ 065E	ٮ 066E	پ 067E	ڎ 068E	ڞ 069E	ڮ 06AE	ھ 06BE	ێ 06CE	۞ 06DE	ۮ 06EE	۾ 06FE
F	؏ 060F	؟ 061F	د 062F	ؿ 063F	◌ُ 064F	▨	ٯ 066F	ٿ 067F	ڏ 068F	ڟ 069F	گ 06AF	ڿ 06BF	ۏ 06CF	◌۟ 06DF	ۯ 06EF	ۿ 06FF

Confusing Similar Characters
Valid Variant Strings

- Assume there are 4 variants to letter (هـ)

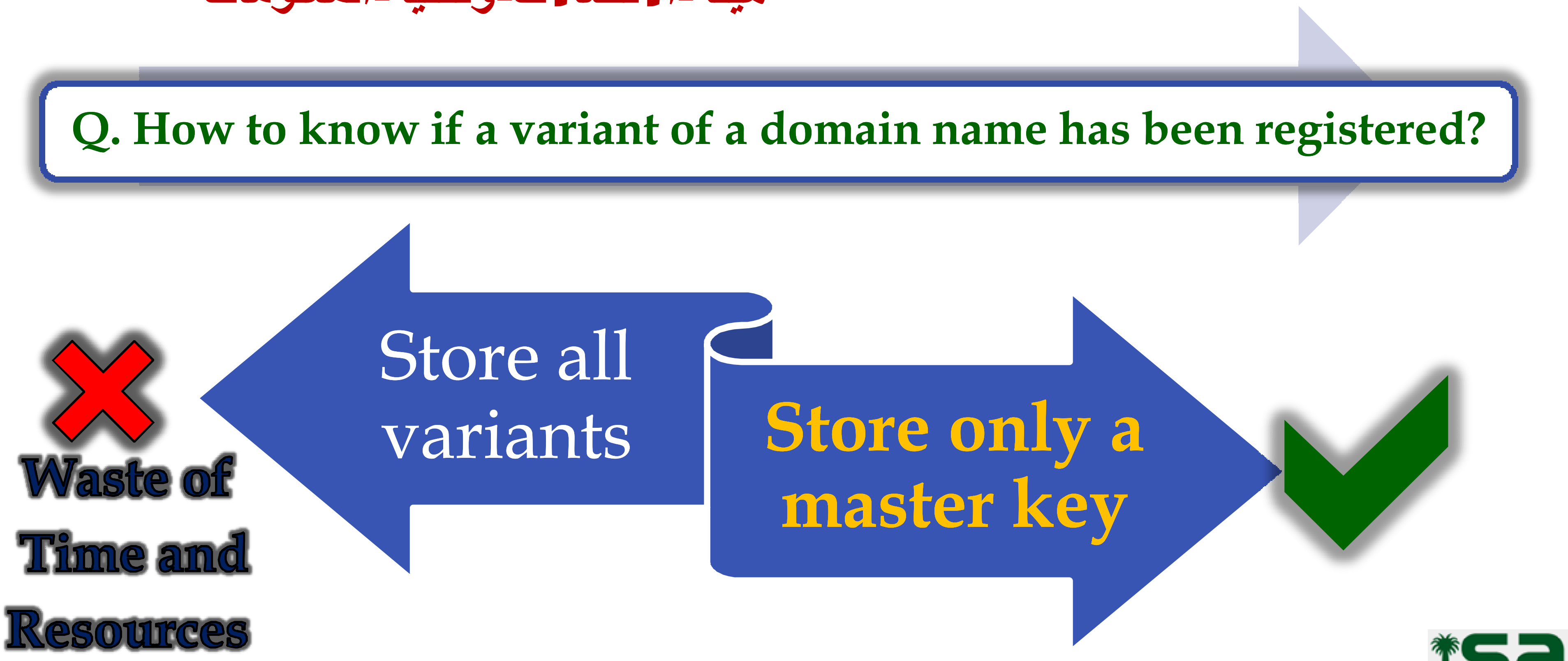
هدهد	هدھد	ہدهد	ەدهد
ھدهد	هدہد	هدەد	ھدھد
ھدہد	ھدەد	ہدھد	ہدہد
ہدەد	ەدھد	ەدہد	ەدەد

16 possible ways to write "هدهد"

Only 4 are confusingly similar (25%)

Confusing Similar Characters
Handling Domain Name Variants

- **It is expected that some domains will have a large number of variants, e.g.:**
  - There are **16,384** possible variants to write the domain “هيئة-الاتصالات-وتقنية-المعلومات”



Q. How to know if a variant of a domain name has been registered?

✗ Waste of Time and Resources

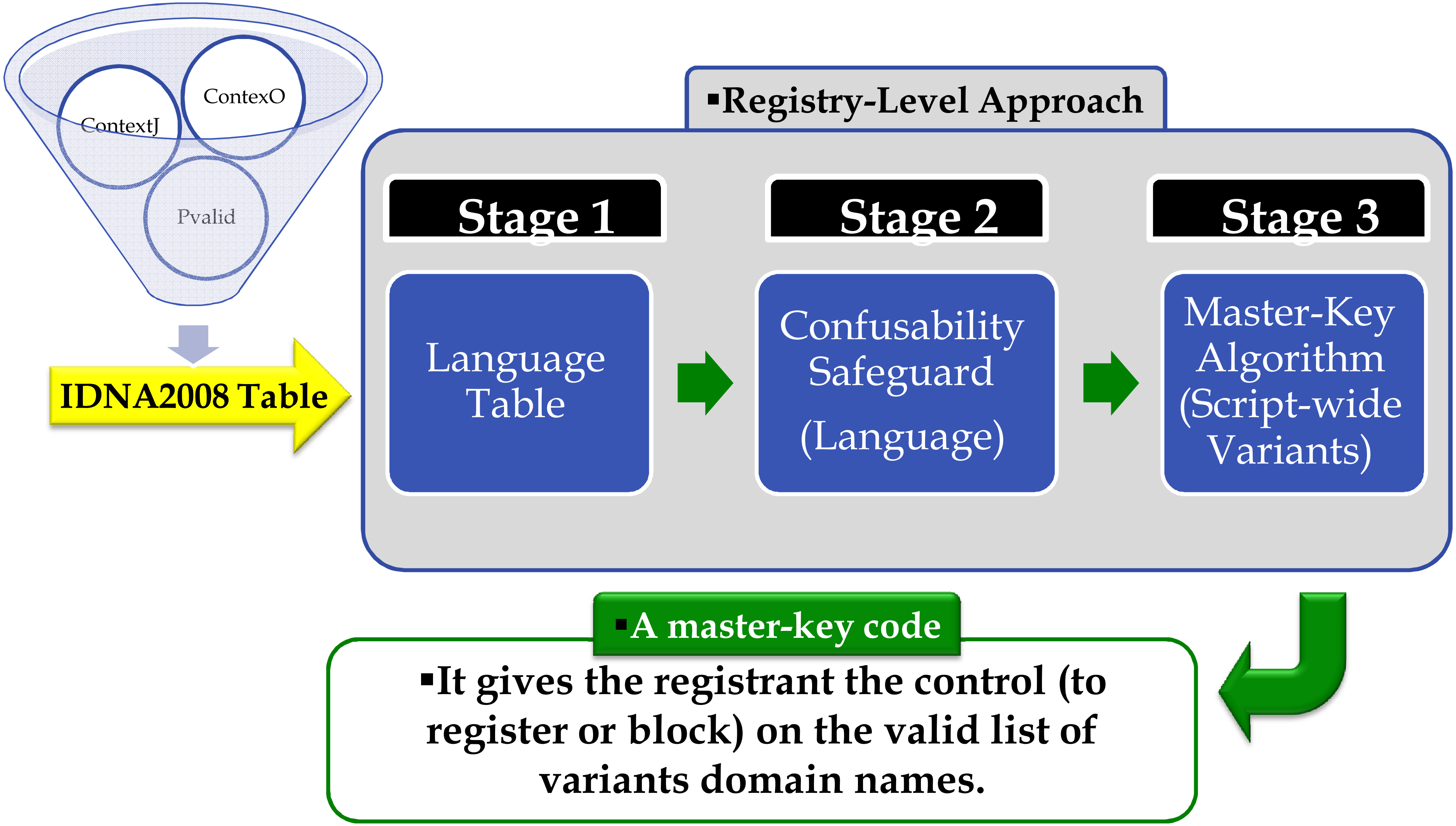
Store all variants

Store only a master key ✓

Characteristics of the Proposed Solution

- Work for both ccTLDs and gTLDs
- Easy and fast to be deploy by any registry
- Extendable to allow for adding new languages as they become ready
- Simple and Transparent for end users
  - Do not annoy/confuse end users with technical/special
  - Regular users should be able to register whatever they can type using available keyboards

Proposed Solution
General overview



IDNA2008 Table

▪Registry-Level Approach

Stage 1

Language Table

Stage 2

Confusability Safeguard

(Language)

Stage 3

Master-Key Algorithm (Script-wide Variants)

▪A master-key code

▪It gives the registrant the control (to register or block) on the valid list of variants domain names.

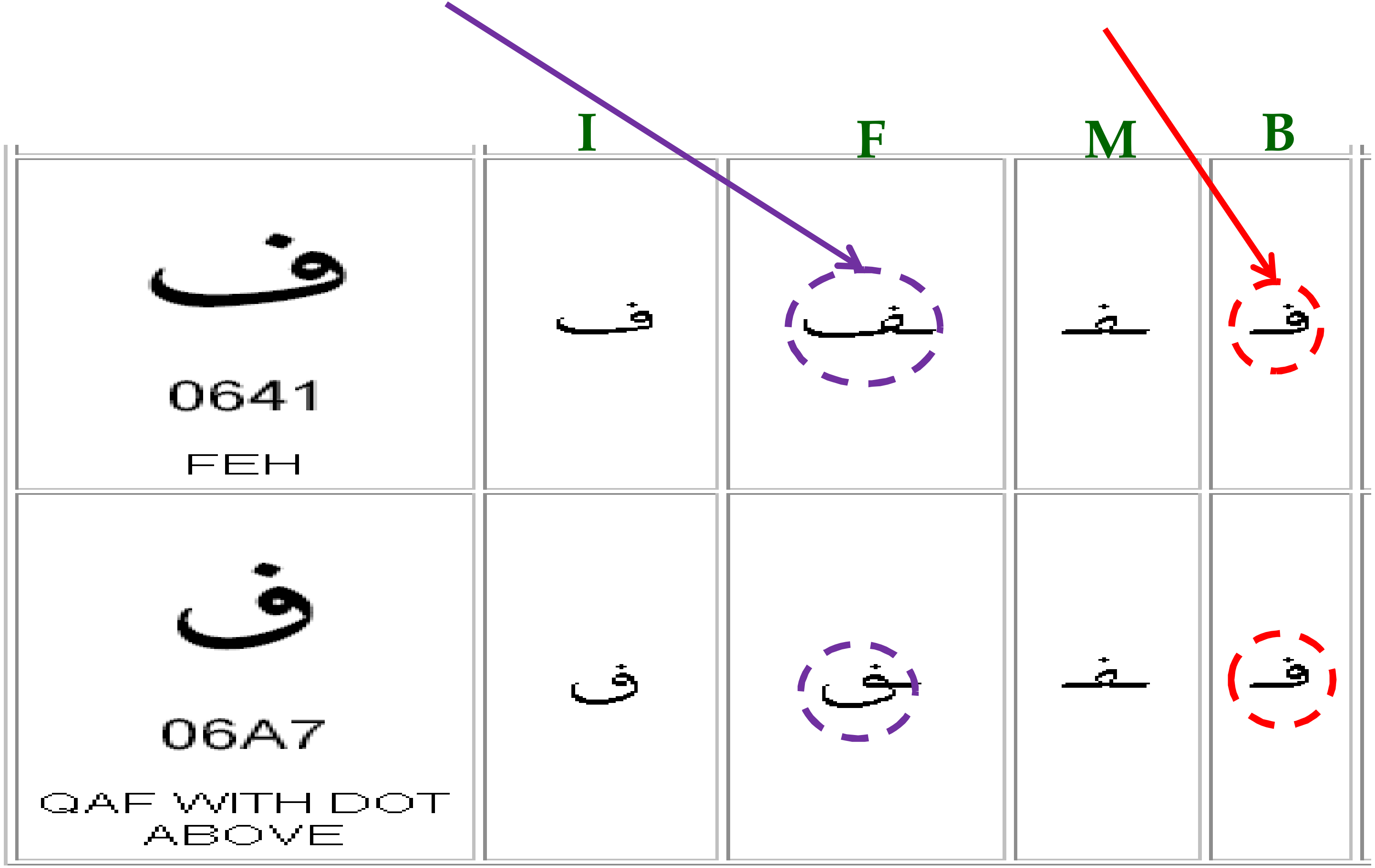
Language-level Required Tables

- **Language Table (LT)**
  - A set of code points (Base characters) to be used by a registry for registering IDN domains in the corresponding language.
  - LT can have Alphabetical, Numbers and Separators (Hyphens, Dots)
- **Variant Table (VT)**
  - A table that records all relations of the LT characters with other characters across the script.
  - Each relation is defined depending on its similarity either:
    - Exact similarity: refers to identically look between base character and another character (e.g. exact match/mirror image).
    - Typo similarity: refers to almost look between base character other character (e.g. typo/style match).
  - Consists of a list of records, each record contains:
    - Base character (from LT),
    - List of other characters (variants) with:
      - A set of positions of similarity [**B**eginning , **M**edial, **F**inal, **I**solated],
      - Relation type (**E**xact, **T**ypo)

Language-level Required Tables

Examples of Variants

- Typo Variants
- Exact Variants



	I	F	M	B
ف 0641 FEH	ﻑ	ﻒ	ﻔ	ﻓ
ڧ 06A7 QAF WITH DOT ABOVE	ڧ	ـڧ	ـڧـ	ڧـ

Language-level Required Tables

Building a Variant Table

Done for each base character in the Language Table (LT)

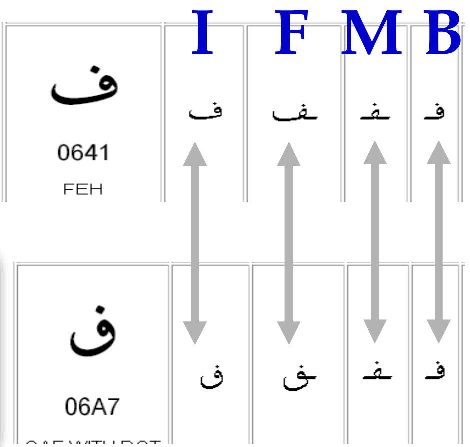
List all possible shapes for the basic character

Search for all its variants from the rest of the Arabic script

Then compare the basic character with its variants in all possible positions.

Find all similarity position(s).

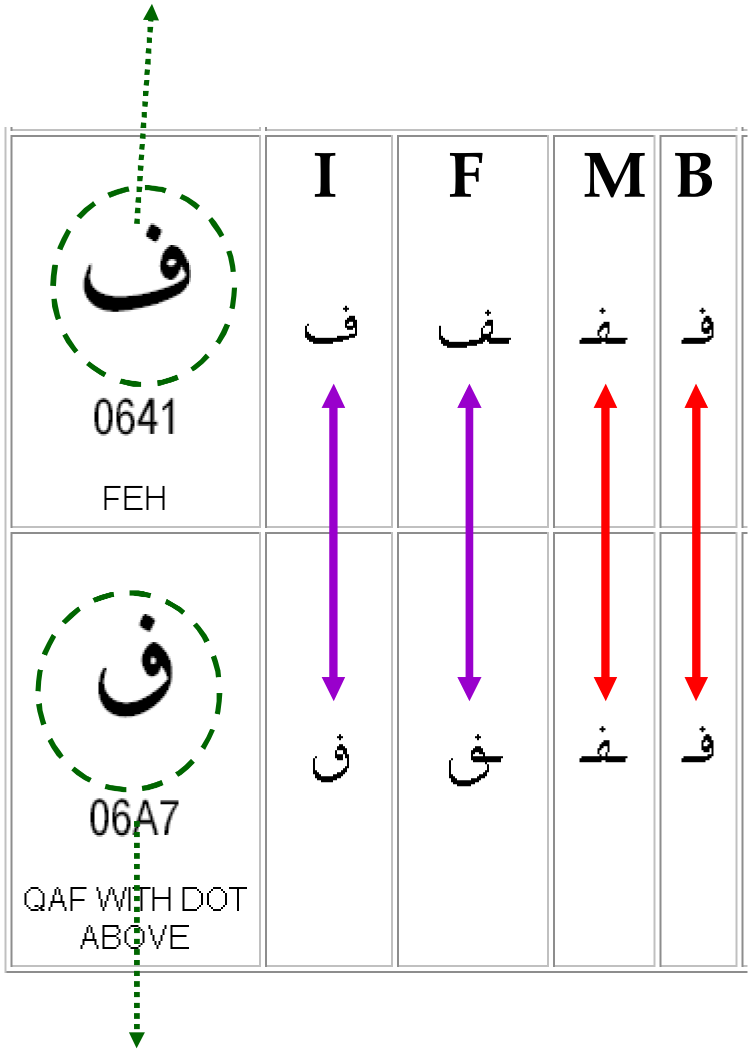
Record the similarity (type & position)



Language-level Required Tables

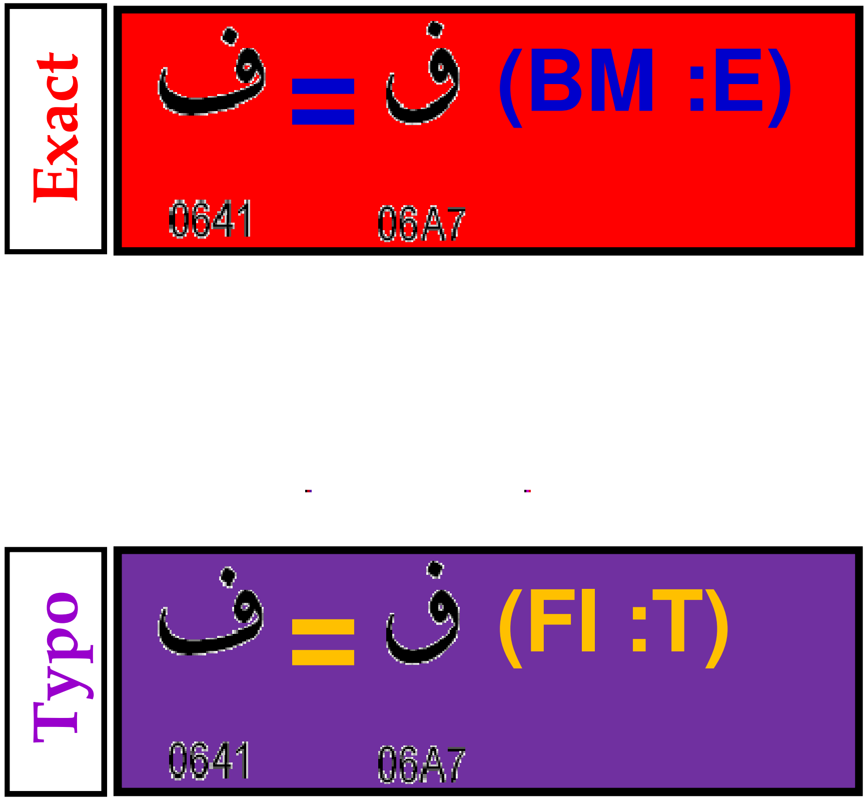
Example: Position of Similarity

A base character from LT



Compare

A variant character from script


Exact

ف = ڧ (BM :E)

0641 06A7

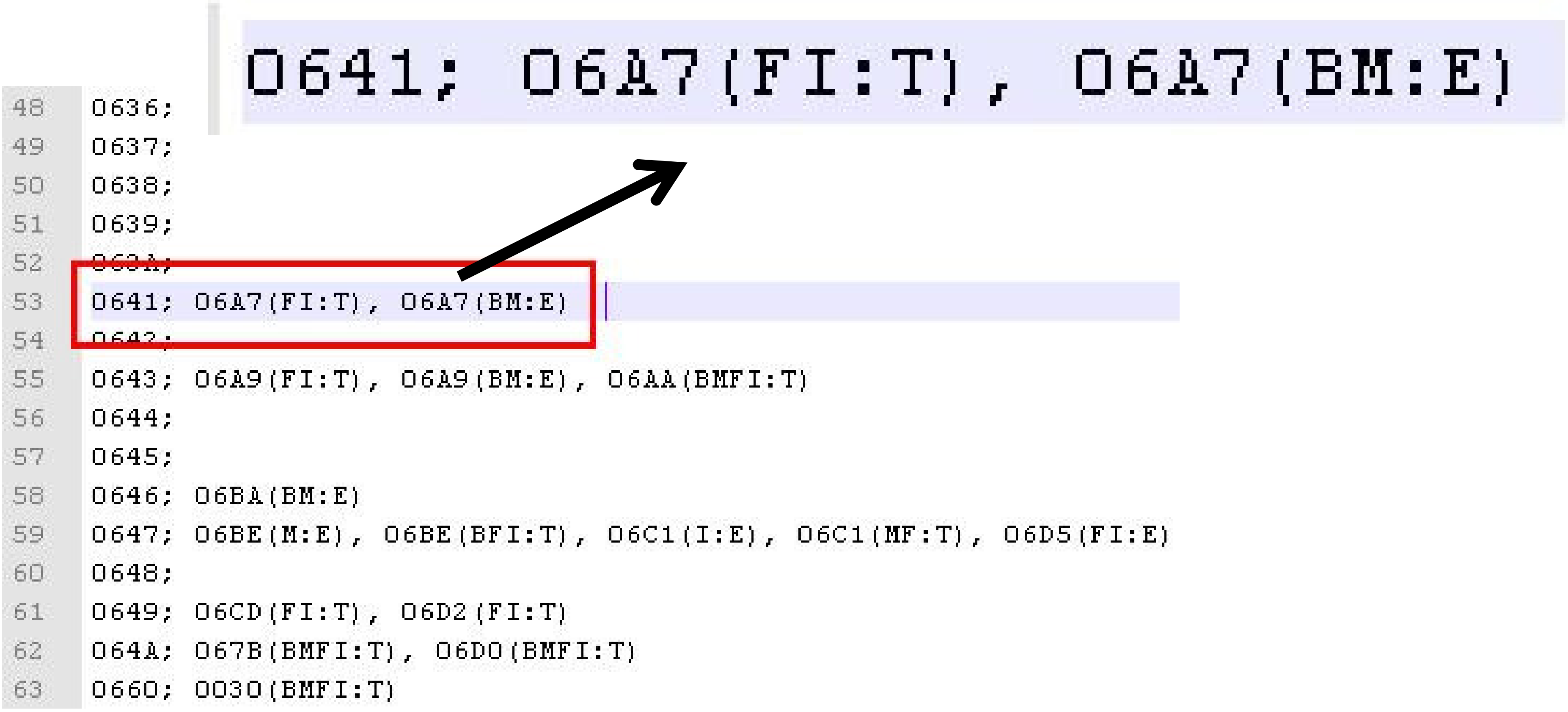
Typo

ف = ڧ (FI :T)

0641 06A7

Language-level Required Tables

Example: Variant Table



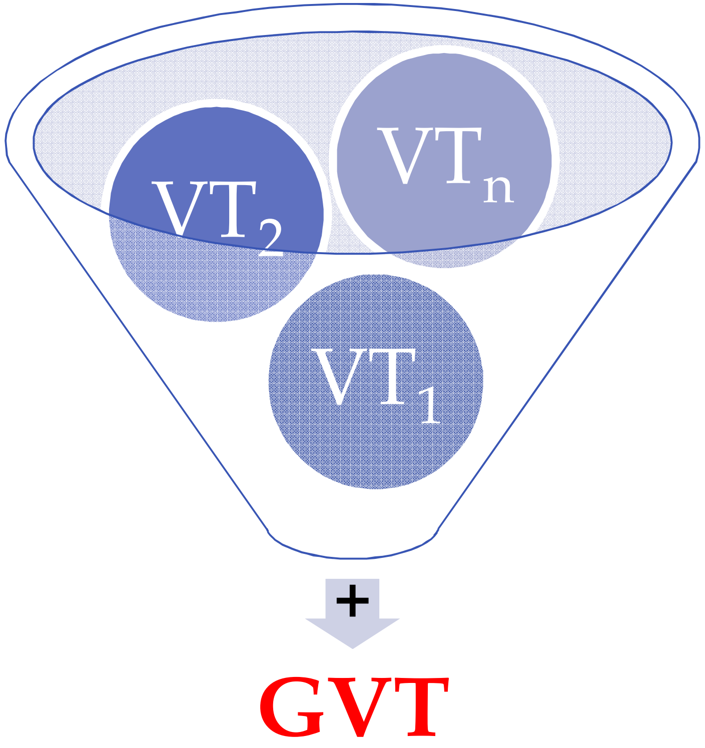
0641; 06A7(FI:T), 06A7(BM:E)

```
0636;
0637;
0638;
0639;
063A;
0641; 06A7(FI:T), 06A7(BM:E)
0642;
0643; 06A9(FI:T), 06A9(BM:E), 06AA(BMFI:T)
0644;
0645;
0646; 06BA(BM:E)
0647; 06BE(M:E), 06BE(BFI:T), 06C1(I:E), 06C1(MF:T), 06D5(FI:E)
0648;
0649; 06CD(FI:T), 06D2(FI:T)
064A; 067B(BMFI:T), 06D0(BMFI:T)
0660; 0030(BMFI:T)
```

Registry-level Required Tables

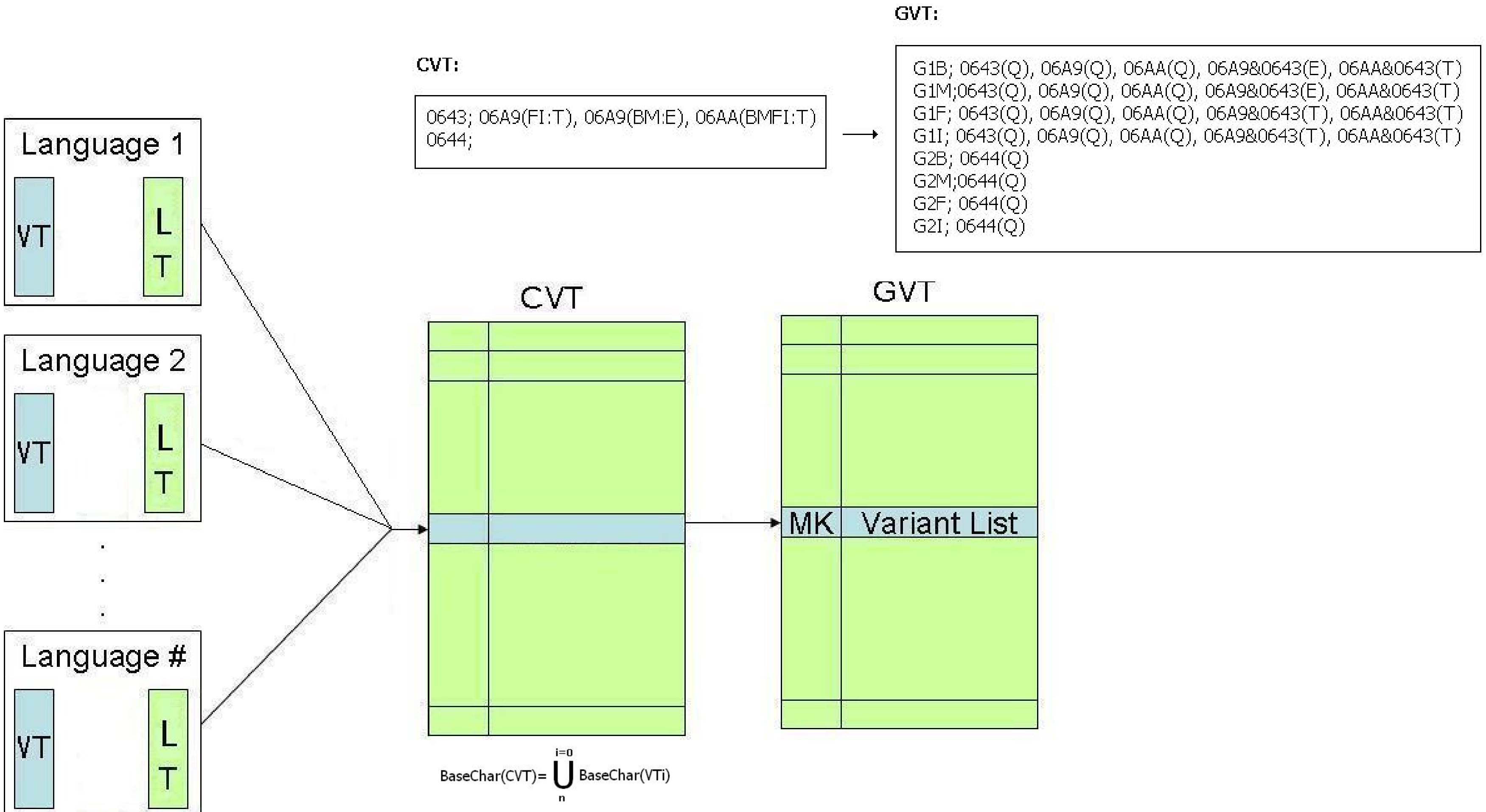
- **Language Tables (one for every supported language)**
  - Users can only register domains using base characters from only one language table.

- **Group Variant Table (GVT):**
  - Generated from variant tables..
    - It combines all VTs into one table that group all base characters with all relations across script.
    - Each variant list will be assigned to a unique group key (master key) that identify that group and will be used for generating the Master Key.



Registry-level Required Tables

Generating Group Variant Table



CVT:

0643; 06A9(FI:T), 06A9(BM:E), 06AA(BMFI:T)
0644;

GVT:

G1B; 0643(Q), 06A9(Q), 06AA(Q), 06A9&0643(E), 06AA&0643(T)
G1M;0643(Q), 06A9(Q), 06AA(Q), 06A9&0643(E), 06AA&0643(T)
G1F; 0643(Q), 06A9(Q), 06AA(Q), 06A9&0643(T), 06AA&0643(T)
G1I; 0643(Q), 06A9(Q), 06AA(Q), 06A9&0643(T), 06AA&0643(T)
G2B; 0644(Q)
G2M;0644(Q)
G2F; 0644(Q)
G2I; 0644(Q)

$$\text{BaseChar(CVT)}=\bigcup_{n}^{i=0}\text{BaseChar(VTi)}$$

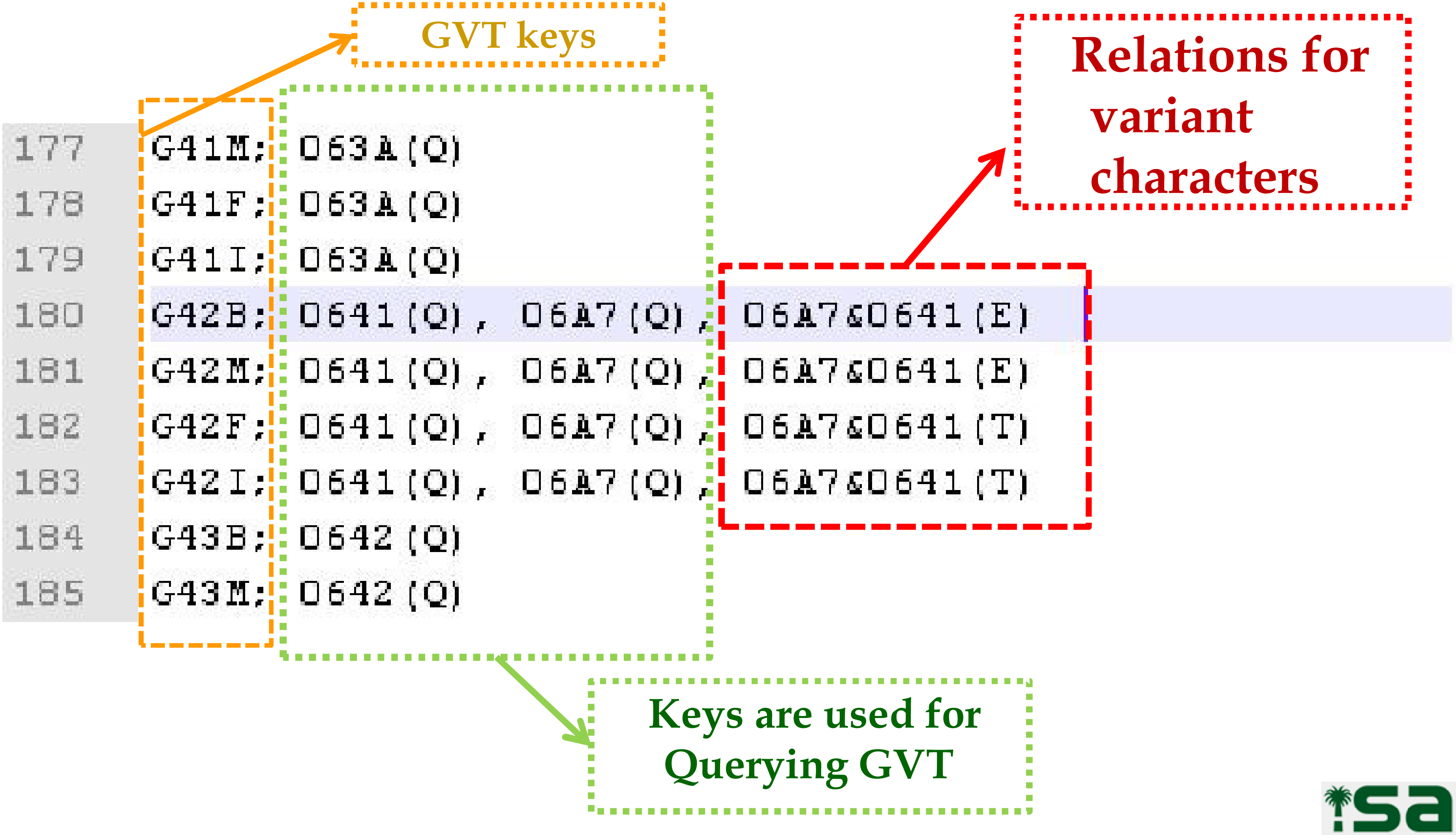
VT: Variant Table
GVT: Group Variant Table

LT : Language Table
MK: Master Key
CVT: Combained Variant Table

Registry-level Required Tables

Example: Group Variant Table

<KEY>; [<char_hex> ”(“<pos>”:Q)” (“,” | ε)]* [<char_hex1>”&”<char_hex2>”&” ”(“<pos>:<rel>”)” (“,” | ε)]*



```
G41M; 063A(Q)
G41F; 063A(Q)
G41I; 063A(Q)
G42B; 0641(Q), 06A7(Q), 06A7&0641(E)
G42M; 0641(Q), 06A7(Q), 06A7&0641(E)
G42F; 0641(Q), 06A7(Q), 06A7&0641(T)
G42I; 0641(Q), 06A7(Q), 06A7&0641(T)
G43B; 0642(Q)
G43M; 0642(Q)
```

GVT keys

Relations for variant characters

Keys are used for Querying GVT

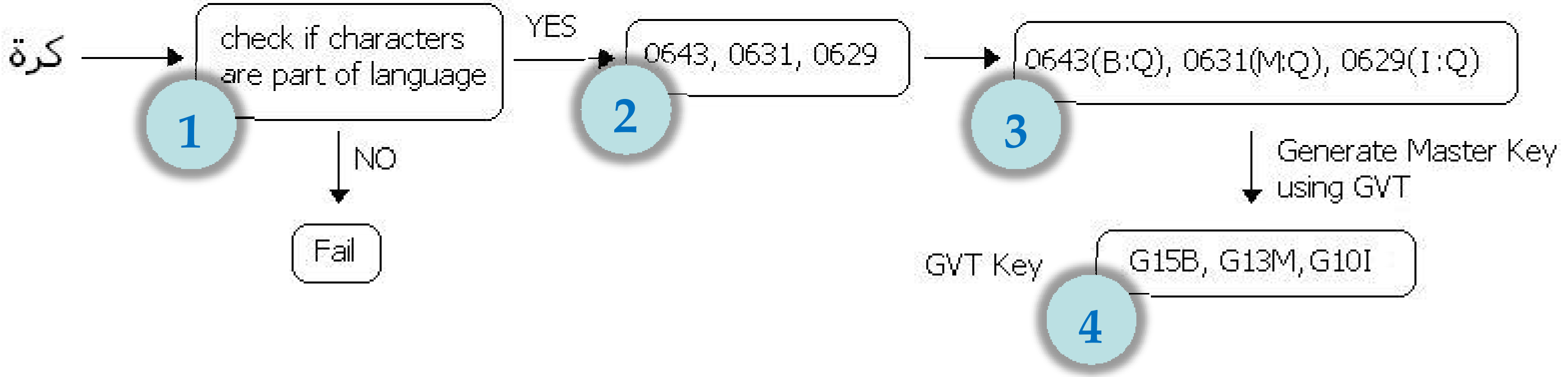
Registry-level Operation
Generating Master Key for a Label

1 Check if the input string follows certain language (using LT).

2 Generate UNICODE code for that input.

3 Identify the position for each character depending on language properties (UNICODE Standard).

4 Generate Master key by taking every code from (step 3) and do simple lookup in GVT.



كرة

1 check if characters are part of language

YES

NO

Fail

2 0643, 0631, 0629

3 0643(B:Q), 0631(M:Q), 0629(I:Q)

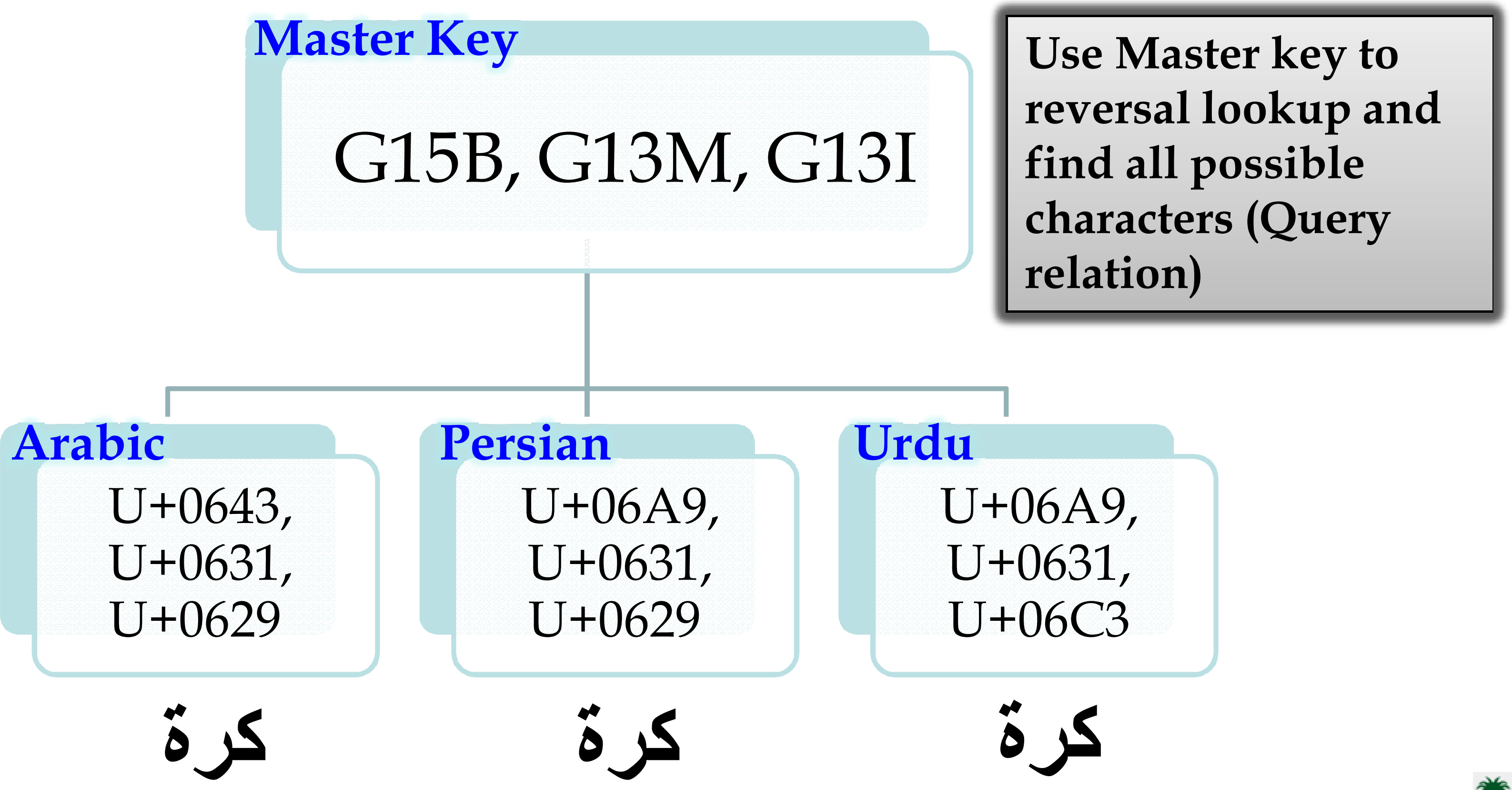
Generate Master Key using GVT

GVT Key

4 G15B, G13M, G10I

Registry-level Operation
Finding Exact Variants

- **Find all Exact strings using Master Key for activation purpose.**



Master Key

G15B, G13M, G13I

Use Master key to reversal lookup and find all possible characters (Query relation)

Arabic

U+0643, U+0631, U+0629

كرة

Persian

U+06A9, U+0631, U+0629

كرة

Urdu

U+06A9, U+0631, U+06C3

كرة

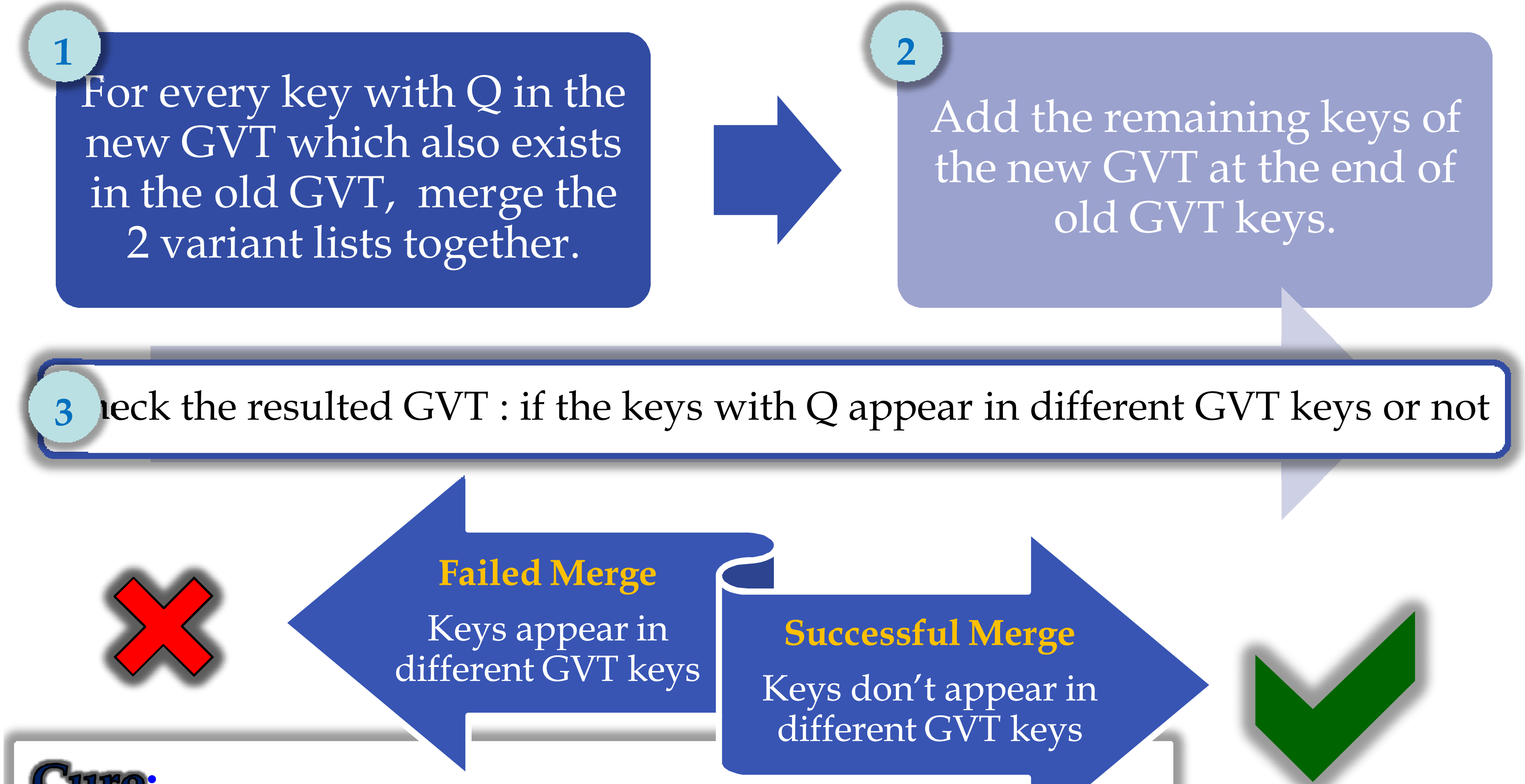
Registrant Interface

- **Lookup process (whois)**
  - Check domain syntax under any supported language using LTs.
  - Check if the same domain is available or not.
  - If it is found return the unavailable/whois-information; otherwise continue
  - Get the master- key for the domain (based on GVT)
  - Check if the master- key was registered before or not
  - If master- key is found return unavailable/whois-information; otherwise return domain is available

- **Registration process:**
  - Registrant should select one of supported languages and a domain (U-Label)
  - Registry should accept inputs based on the selected language table
  - If domain name can be registered (available based on Lookup process) then register the domain

- **Activation process (enable exact variants)**
  - Original Registrant can activate any exact variant from the registered domain's Master Key.
  - List possible Exact variants that can be typed using one of the LT without intermingling between them
  - Activate one/many of Exact variants (if not activated before)

Registry-level Operation
Adding New Languages



1 For every key with Q in the new GVT which also exists in the old GVT, merge the 2 variant lists together.

2 Add the remaining keys of the new GVT at the end of old GVT keys.

3 heck the resulted GVT : if the keys with Q appear in different GVT keys or not

Failed Merge

Keys appear in different GVT keys

Successful Merge

Keys don't appear in different GVT keys

Cure:

Regenerate old GVT using existing VTs including the new VTs.

Then regenerate all old Master keys using new GVT!

Conclusions

- **We tried to have a prototype that fulfill the concepts of script based registry that is:**
  - Optimized, Simple, Transparent, Automated, and addresses many local issues
- **Next steps:**
  - Automating the process of finding variant characters.
  - Variant TLDs should be delegated:
    - E.g. Arabic => كويت => U+0643 U+0648 U+064A U+062A

      Persian => کویت => U+06A9 U+0648 U+06CC U+062A

Thank you !

Thank you

شكرا

U+0634 U+06A9 U+0631 U+0627

شكرا

U+0634 U+0643 U+0631 U+0627

G35B G44M G32F G22I