



Proposed Solution for Writing Domain Names in Different Arabic Script Based Languages

ASIWG 3rd Meeting, Cairo Nov 8-10 2008

Dr. Abdulaziz H. Al-Zoman

Director of SaudiNIC - CITC

**Chairman of Steering Committee
Arabic Domain Name Pilot Project**

Raed I. Al-Fayez

SaudiNIC - CITC

**Chairman of Technical Committee
Arabic Domain Name Pilot Project**

Agenda

- Introduction
- Characteristics of A Desired Solution
- What we are trying to solve
- Requirements for Languages
- Requirements for Registries
- How to support new languages
- Conclusion

Introduction

About Arabic script



- The **2nd** most widely used alphabetic writing system in the world (used by more than **43 countries**)
 - more than **one billion potential users** could be concerned in using Arabic script domain names.
- Used by **many languages** such as: Arabic, Persian, Urdu, Turkish, Kurdish, Pashto, Swahili, ...
 - that may **add or change** characters to represent phonemes that do not appear in Arabic phonology.
 - A new character usually created based on **modifying the basic shape** of an existing Arabic character, for example, by adding more dots.
 - These additions have **meaning to the new language** but not to the original language.
 - Therefore, many characters would **easily be confused** with some other characters from other languages

Introduction

Unicode Arabic Block (5.1)

- Subtending marks
- Radix symbols
- Letterlike symbols
- Punctuation
- Currency Signs
- Poetic marks
- Honorifics
- Koranic annotation signs
- Based on ISO 8859-6
- Addition for early Persian and Azerbaijani
- Points from ISO 8859-6
- Combining Madah and Hamza
- Other combining marks
- Arabic-Indic digits
- Archaic letters
- Points
- Extended Arabic letters
- Extended Arabic letters for Parkari
- Eastern Arabic-Indic digits
- Signs for Sindhi
- Additions for Khowar
- Additions for Torwali
- Additions for Burushaski
- Additions for early Persian

298 Code points

Introduction

Arabic Script IDN Major Issues

- **Acceptable/disallowed characters**
 - IDNA200x table (Pvalid / Disallowed / ContextO)
 - Language tables
- **Non-spacing Marks**
 - Subtending Marks (U+0600 - U+0603)
 - Honorifics (U+0610 - U+0614)
 - Koranic annotation signs (U+0615 - U+061A, U+06D6 - U+06ED)
 - Points (U+064B - U+0652, U+0670)
 - Combining Maddaa and Hamza (U+0653 - U+0655)
 - Other combining Marks (U+0656 - U+065E)
- **Confusing similar characters (e.g. variant tables)**
- **World/label separators (space, ZWNJ, ZWJ, hyphen)**
- **Bidirectional**
- **They are addressed at different levels:**
 - IDNA protocol level
 - Registry level
 - Application level

Introduction

Registry Challenges!



- **Security issues** (stability, trust,...) e.g. phishing
 - They should be addresses at language level first
- **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

Introduction

Who is ready?



- Some language communities are **somehow ready** (alphabetical order):
 - Arabic
 - Jawi
 - Pashto
 - Persian
 - Urdu
 - ...

Introduction

Main problem: Security!



کلی

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

کلی

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

Site # - Microsoft Internet Explorer provided by CITC

File Edit View Favorites Tools Help

Address http://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

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 be open, then they should focus on languages that use Arabic script.
4. If new IDN-gTLDs will be open, then they should focus on languages that use Arabic script.

For more information see this the Power Presentation that was delivered on 11/11/2008

Done

Site # - Microsoft Internet Explorer provided by CITC

File Edit View Favorites Tools Help

Address http://کلی.com/

Site #

2

Arabic Label (U-Label) : کلی

Unicode (U+) : U+0643, U+0644, U+0649

ASCII Label (A-Label) : xn--fhbcp.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.

Now see where this domain will go : کلی.com

Characteristics of A Desired Solution



- Based on **standardized** (or agreeable) policies and procedures that are documented on RFC-like or Best-Practice documents
- **Extendable** to allow for adding new languages as they become ready
- **Easy and fast** to be deploy by any registry
- Work for both **ccTLDs** and **gTLDs**

Why we need this?



- **Optimized solution**
 - Only block domains that are **really** confusing to the end users
- **Simplify Registry operations**
 - Fast and accurate Whois service
 - Simple registration & activation services
- **Transparent to end users (**registrant** and **navigator**)**
 - Keep it simple & similar to what they used to
- **Automate expandability**
 - No need to meet and discuss the same issues again when a new language is ready!

Why we need this?

Example: Optimize the solution

- If some one want to register “هدهد”
 - Assuming there are 3 variants for the letter “ه”
 - There will be 16 possible ways to write it
 - Only 4 of them may confuse the end users (25%)
 - So why we block/bundle 75% not confusing domains?

هدهد	هدهد	هدهد	هدهد
هدهد	هدهد	هدهد	هدهد
هدهد	هدهد	هدهد	هدهد
هدهد	هدهد	هدهد	هدهد

Registration Form (demo)

نموذج التسجيل (تجريبي)

Invalid option (position)

Mixed of languages

Can be enabled

Choose a Language	<input type="text" value="عربي"/>	اختر اللغة
Type a domain name	<input type="text" value="هيئة"/>	اسم النطاق
TLD	<input type="text" value="مثال."/>	النطاق العلوي
<input type="button" value="Clear مسح"/> <input type="button" value="Search ابحث"/>		

Number of possible matches including requested domain: 64

Difficult

- (U+0629)ة (U+0626)ئ (U+064A)ي (U+0647)ه
- (U+0629)ة (U+0626)ئ (U+064A)ي (U+06BE)ه

For "هيئة" 16 out of 64 are confusing
(same results only 25% are really confusing)!!!

Why we need this?

Simplify the registry operation

- **How to know if a domain name have an already registered variant?**
 - Assuming we have define variants based on the inputs of current 3 ready languages;
 - There are **16,384** possible variants to write the domain “هيئة-الاتصالات-وتقنية-المعلومات”
 - Are we going to store/bundle all of them?
 - Dose the Whois function supposed to search in all registered domains & all possible variants for each one of them?
 - Not possible: Time & Resource consuming!!!!
 - Need alternative way to handle this issue (Master-Key)

Why we need this?

Transparent for the End users

- **Do not annoy/confuse end users with technical/special terms:**
 - Type & Exact Variants
 - Register & Bundle & Activate

- **Regular users should be able to register what he can type using his keyboard**
 - Advance users may seek for other options/solutions to fulfill their needs.

Why we need this?

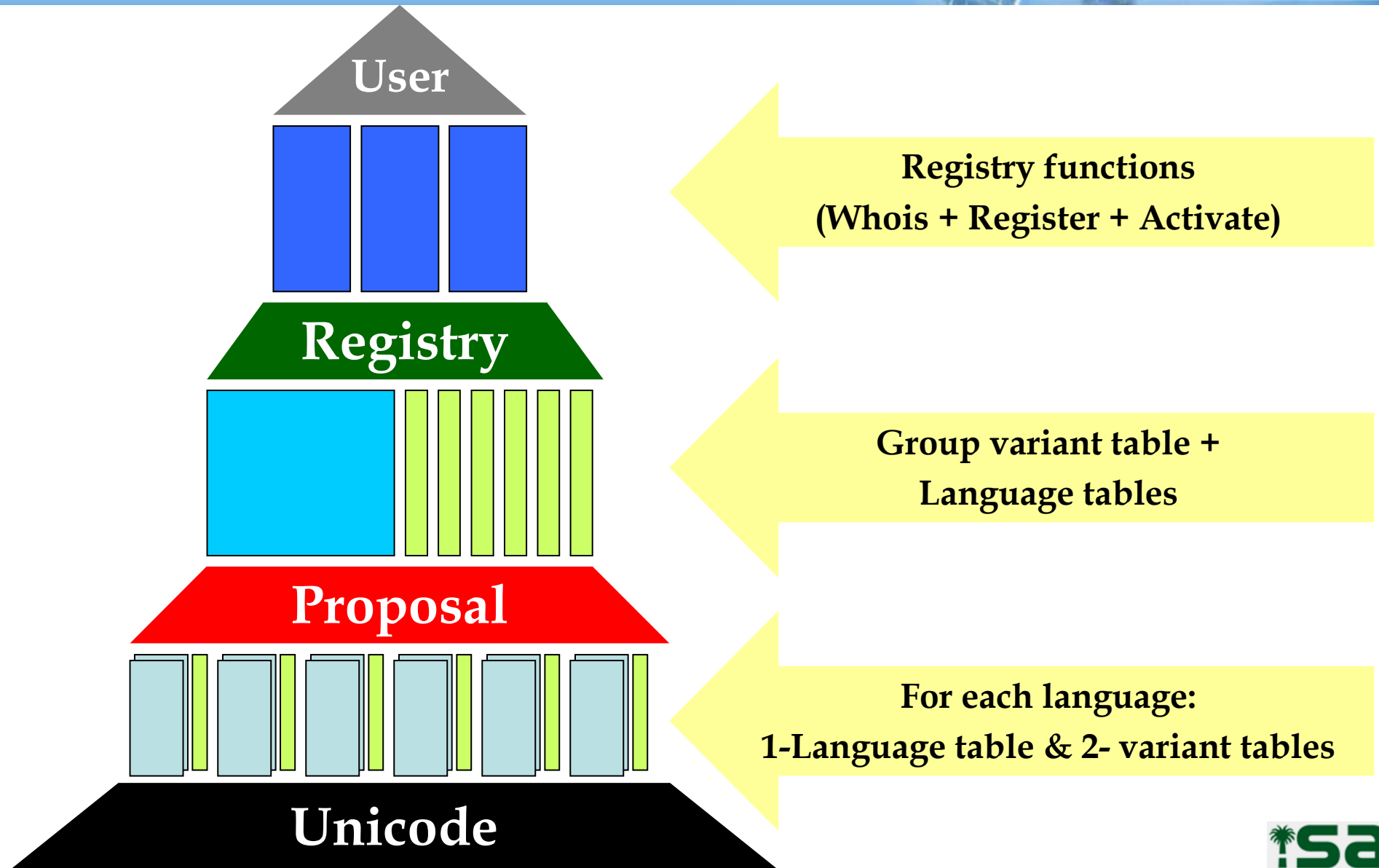
Automate expandability



- **Extendibility against**
 - Adding new languages
 - Updating existing language tables & variants lists
 - Handling Unicode updates (e.g. Unicode 5.2)

- **Define set of rules and algorithms that Registries can use to operate TLD using Arabic Script**
 - Rules for storing some structures for backward compatibility

General overview



Language Requirements

- **Language Table**
 - A simple code point set table that **includes ONLY needed** letters, digits or any other code points used in the language
- **Variant tables**
 - **Exact Variant Table (EVT)**
 - A comprehensive variant table that includes all codepoints (from the whole script) that are **confusingly similar** (exact match/mirror image)
 - **Typo Variant Table (TVT)**
 - A comprehensive variant table that includes all codepoints (from the whole script) that **may confuse** end users (e.g. **typo/style match**)
 - **Note: A variant table will consist of a list of records, each record contains the following information:**
 - Basic character codepoint,
 - List of other variant codepoints (from the rest of the script),
 - Each with position of similarity [**Initial, Medial, Final, Isolated**]

Language Requirements

Examples of variants



- Example **Exact** Variant match

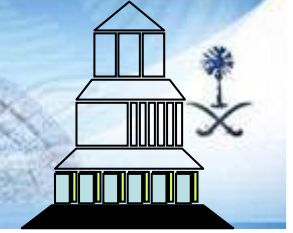
Representative shape in code chart	Possible shapes in context [Standalone, End, Middle, Beginning]			
<p>ف</p> <p>0641</p> <p>FEH</p>	ف	ف	ف	ف
<p>ف</p> <p>06A7</p> <p>QAF WITH DOT ABOVE</p>	ف	ف	ف	ف

- Example for **Typo** Variant match

Representative shape in code chart	Possible shapes in context [Standalone, End, Middle, Beginning]			
<p>ف</p> <p>0641</p> <p>FEH</p>	ف	ف	ف	ف
<p>ف</p> <p>06A7</p> <p>QAF WITH DOT ABOVE</p>	ف	ف	ف	ف

Language Requirements

How to build a variant table?













- **Steps : (done for each basic character)**
 1. List all possible shapes for the basic character
 2. Search for all its variants from the rest of the Arabic script
 3. Then compare the basic character with its variants in all possible positions.
 4. Generate the “similarity position” bits (see next example)
- **Do it in 2 rounds:**
 - 1st round for building Exact variant table (EVT)
 - 2nd round for building Type variant table (TVT)

Language Requirements

Example: Position of similarity



Codepoint from a language table

Representative shape in code chart	Possible shapes in context [Standalone, End, Middle, Beginning]			
 0641 FEH				
 06A7 QAF WITH DOT ABOVE				

Compare



Variant Codepoint from the script

1st Round (Exact)

Calculate similarity position

S	E	M	B
0	0	1	1

= 1+2 = 3


=

(3)



0641
06A7

2nd Round (Typo)

Calculate similarity position

S	E	M	B
1	1	0	0

= 8+4 = 12


=

(12)

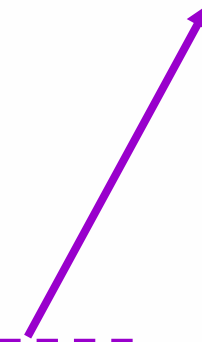
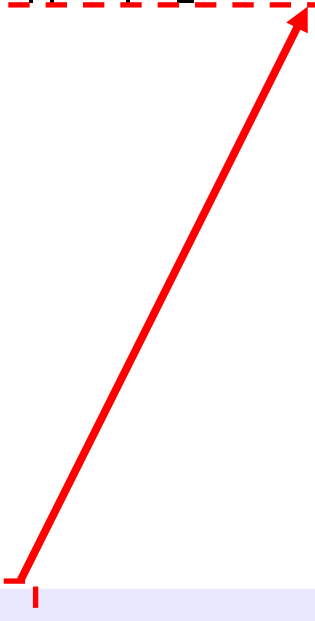
0641
06A7

Exact Variant Table

```
Arabic.txt
# 1.1 Letters
27 # 1.1 Letters
28 0621;
29 0622; U U J H ,
30 0623;
31 0624;
32 0625; 0641; 06
33 0626;
34 0627;
35 0628;
36 0629;
37 062A;
38 062B;
39 062C;
40 062D;
41 062E;
42 062F;
43 0630;
44 0631;
45 0632;
46 0633;
47 0634;
48 0635;
49 0636;
50 0637;
51 0638;
52 0639;
53 063A;
54 0641; 06A7 (3)
55 0642;
56 0643; 06A9 (3)
57 0644;
58 0645;
59 0646; 06BA (3)
60 0647; 06BE (2), 06C1 (8), 06D5 (12)
61 0648;
62 0649; 06CC (12)
63 064A; 06CC (3)
64
```

Typo Variant Table (TVT)

```
Arabic.txt
# 1.1 Letters
27 # 1.1 Letters
28 0621;
29 0622; 0671 (12)
30 0623; 0672 (12)
31 0624;
32 0625; 0673 (12)
33 0626; 06D3 (12)
34 0627;
35 0628;
36 0629; 06C3 (4)
37 062A; U U J H ,
38 062B;
39 062C; 0641; 06A7 (12)
40 062D;
41 062E;
42 062F;
43 0630; 0642 .
44 0631;
45 0632;
46 0633;
47 0634;
48 0635;
49 0636;
50 0637;
51 0638;
52 0639;
53 063A;
54 0641; 06A7 (12)
55 0642;
56 0643; 06A9 (12), 06AA (15)
57 0644;
58 0645;
59 0646;
60 0647; 06BE (13), 06C1 (6)
61 0648;
62 0649; 06CD (12), 06D2 (12)
63 064A; 067B (15), 06D0 (15)
64
65 # 1.2 Digits
```



Registry Requirements



- **Language Tables (one for each supported language)**
 - Users can only register domains using code-points from only one language table

- **Group Variant Table (GVT):**
 - To be generated from the variant tables.
 - It combines all variant tables into one table that group codepoints which are similar in a given position.

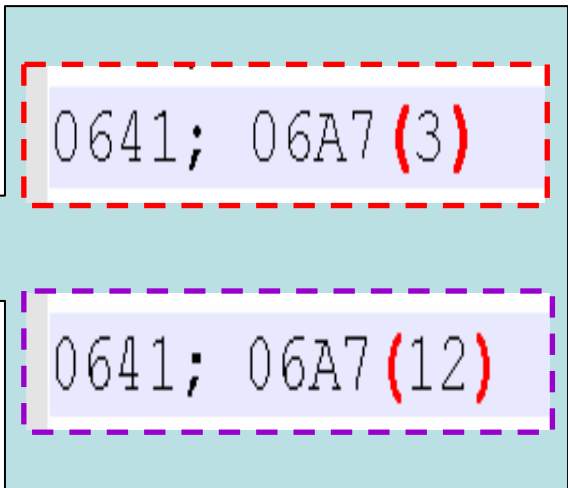
 - Each group will be assigned a group-key (master key) that uniquely identify the group and will be used to map all string variants to a unique string.

Registry Requirements

Example: Group variant table

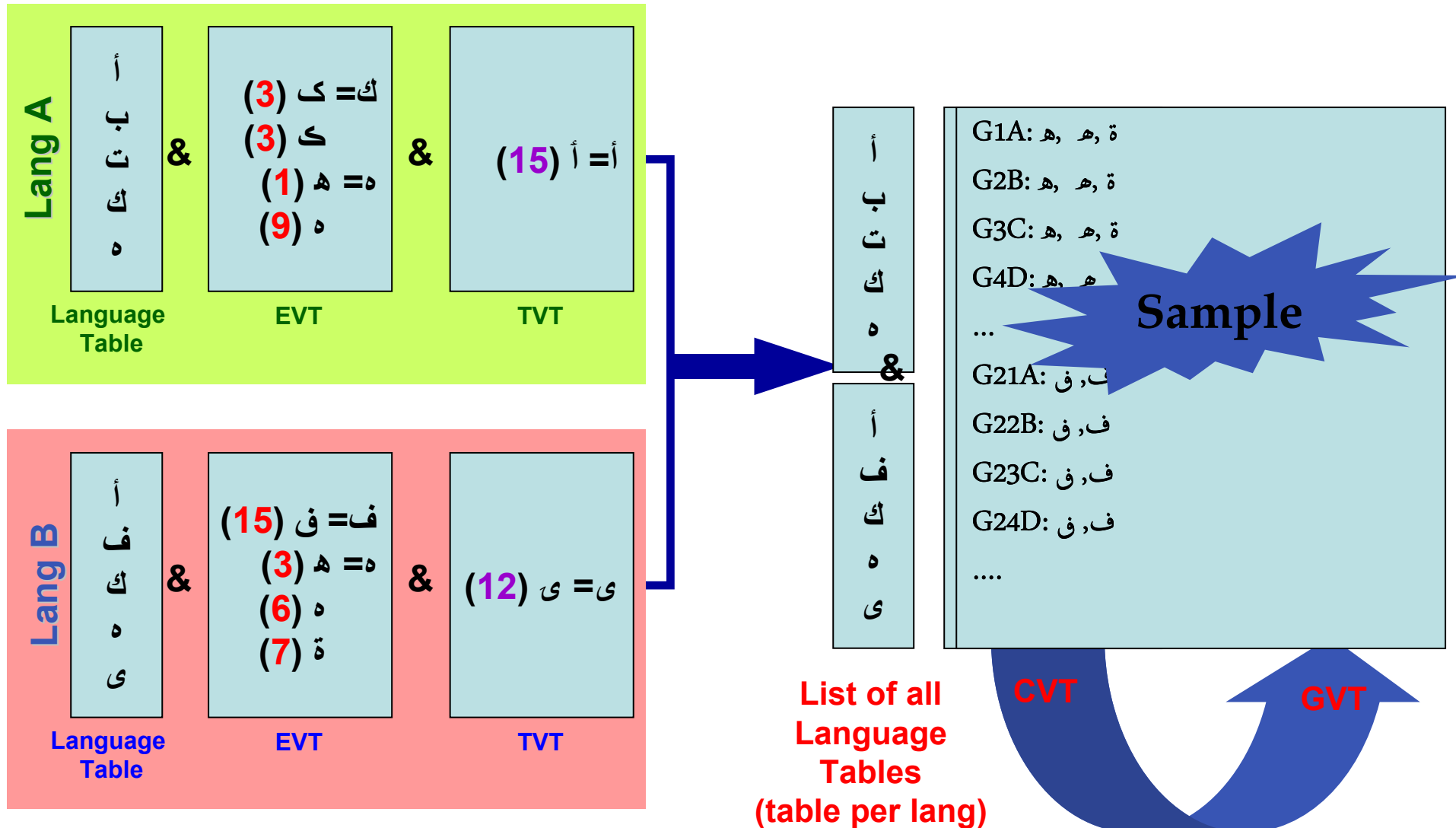


```
ASIWG.gvt
97 G25A; 0639(1:E)
98 G25B; 0639(2:E)
99 G25C; 0639(4:E)
100 G25D; 0639(8:E)
101 G26A; 063A(1:E)
102 G26B; 063A(2:E)
103 G26C; 063A(4:E)
104 G26D; 063A(8:E)
105 G27A; 06A7(1:E), 0641(1:E)
106 G27B; 06A7(2:E), 0641(2:E)
107 G27C; 0641(4:E), 06A7(4:T)
108 G27D; 0641(8:E), 06A7(8:T)
109 G28A; 0642(1:E)
110 G28B; 0642(2:E)
111 G28C; 0642(4:E)
112 G28D; 0642(8:E)
113 G29A; 0643(1:E)
114 G29B; 0643(2:E)
115 G29C; 0643(4:E)
116 G29D; 0643(8:E)
117 G30A; 0644(1:E)
118 G30B; 0644(2:E)
119 G30C; 0644(4:E)
120 G30D; 0644(8:E)
121 G31A; 0645(1:E)
122 G31B; 0645(2:E)
123 G31C; 0645(4:E)
124 G31D; 0645(8:E)
125 G32A; 06BA(1:E), 0646(1:E)
126 G32B; 06BA(2:E), 0646(2:E)
127 G32C; 0646(4:E)
128 G32D; 0646(8:E)
```



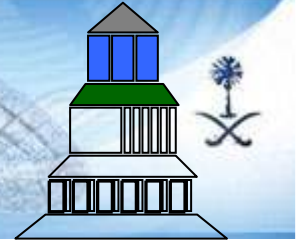
Registry Requirements

Example: Combining the languages work



Registry Requirements

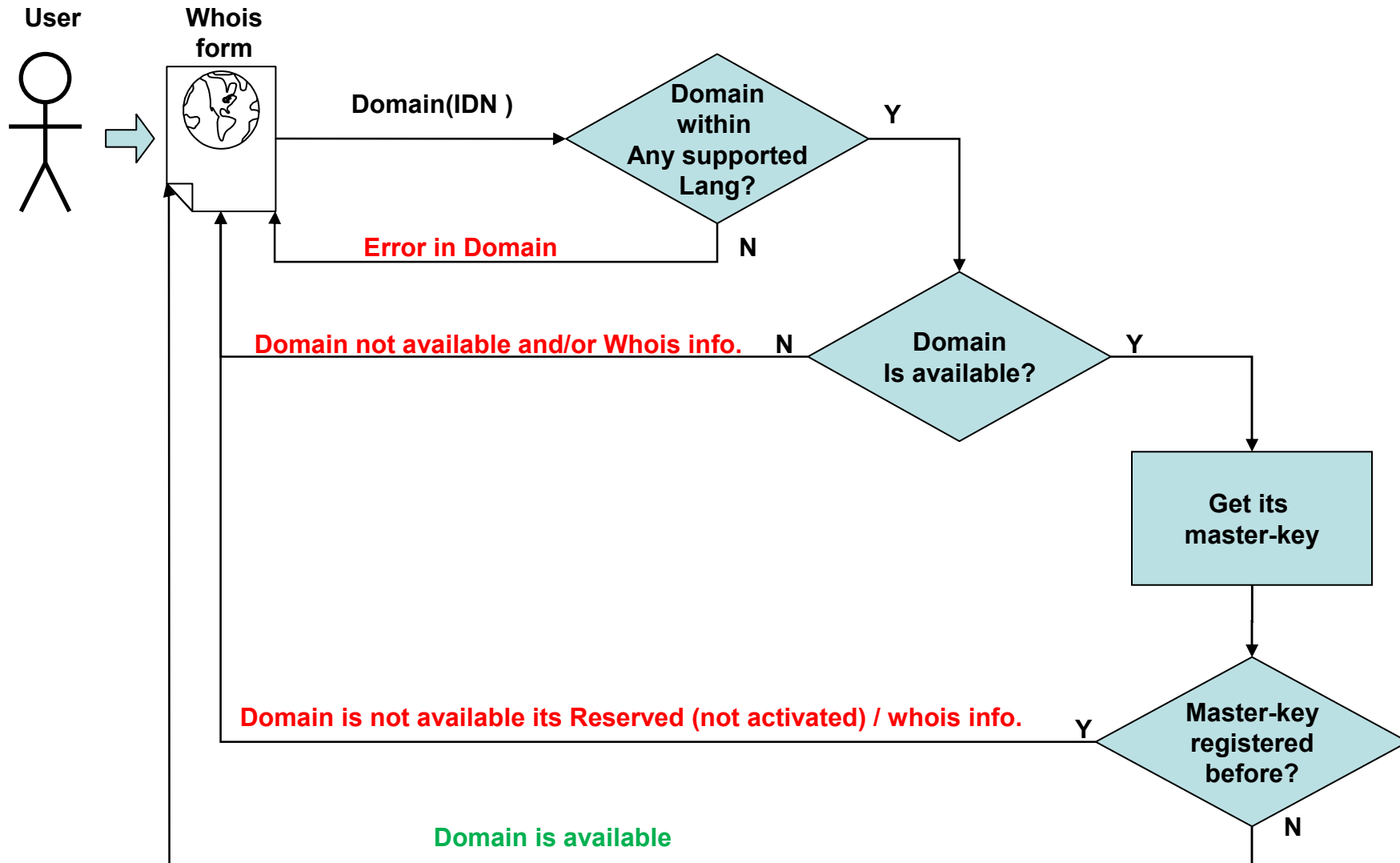
Registry-Registrant interface



- **Lookup process (whois)**
 - Check domain syntax
 - Check if the domain within any supported language
 - Check if the same domain is available or not
 - Check if the request domain exists
 - If it is found return the unavailable/whois-information; otherwise continue
 - Check if the domain is variant for a registered domain name (master-key)
 - Get the master- key for the string (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 these languages and type a domain (U-Label)
 - Registry should **accept inputs** based on the selected language table
 - If domain name is **register-able** (available based on Lookup process)
 - Register the domain name along with its original image
 - List of allowable exact variants (to be activated if needed)
- **Activation process (enable exact variants)**
 - Original registrant can **activate exact** variants from his registered domain
 - List possible **Exact** variants that can be typed using one of the supported languages without intermingling between languages tables and taking care of position similarity (suggestion)
 - Activate one/many of the previous variants (if not activated before)

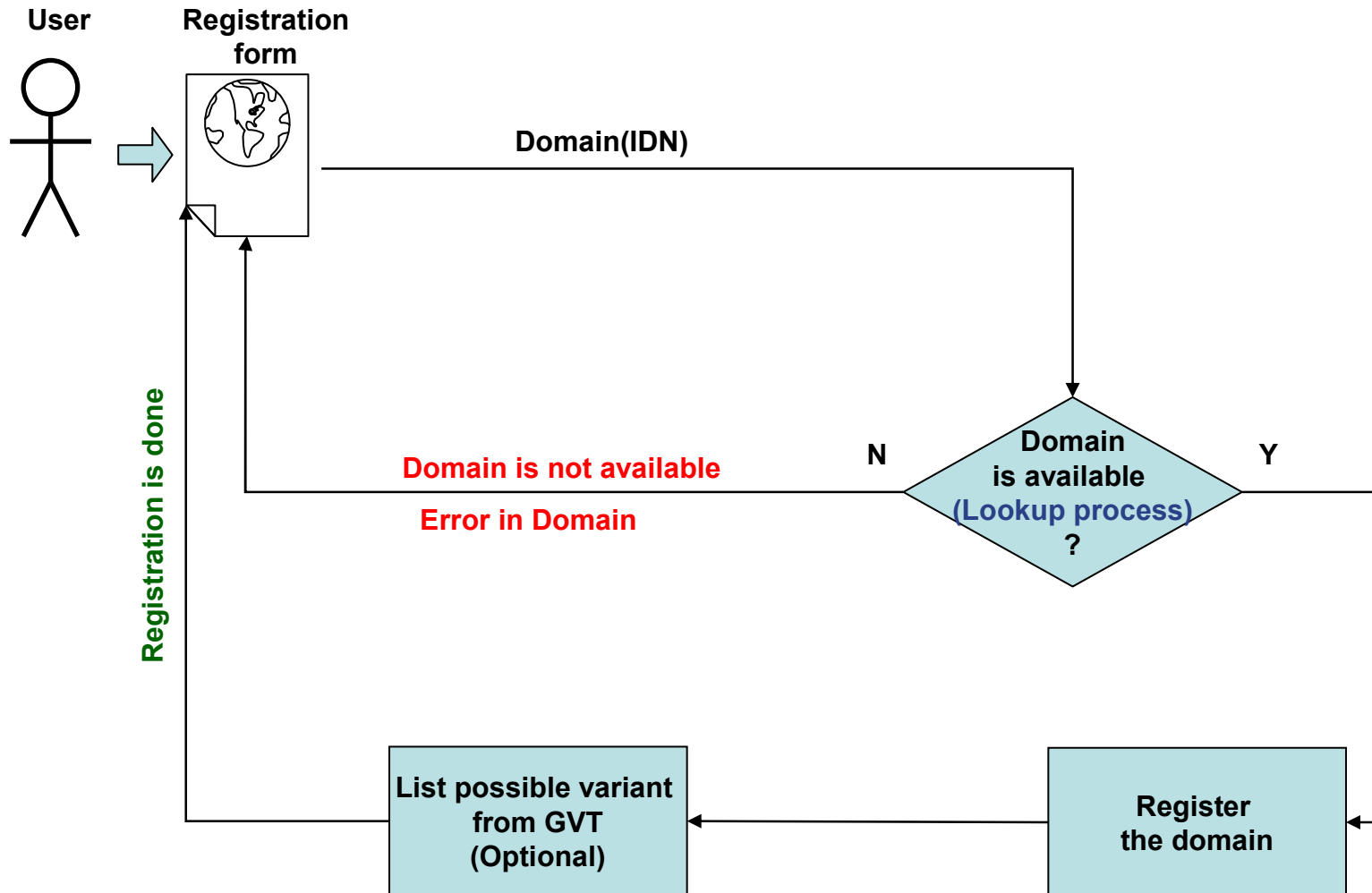
Registry Requirements

1. Lookup process (whois)



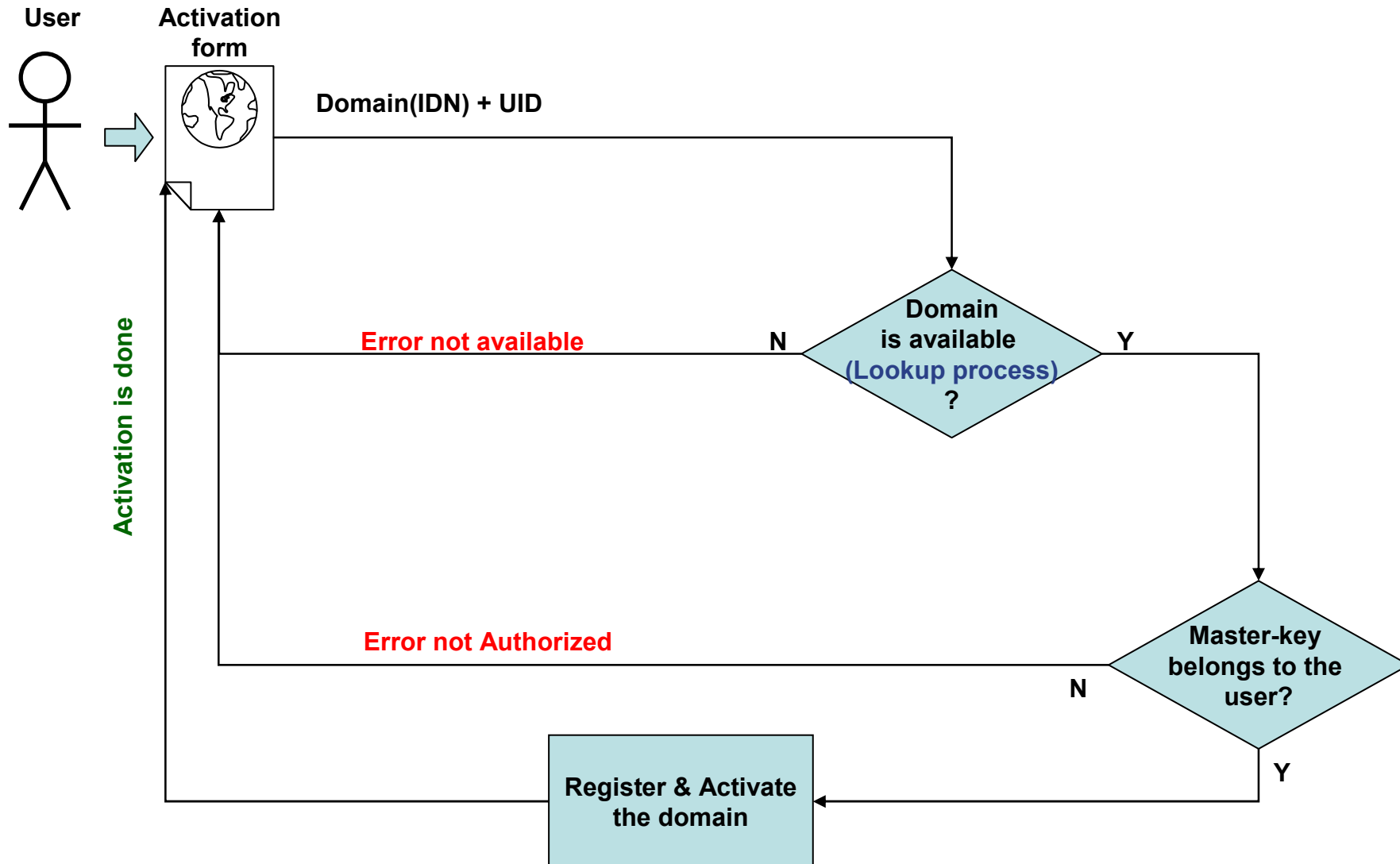
Registry Requirements

2. Registration process



Registry Requirements

3. Activation process



Steps to support a new language



- **Need process owners !!!!**
- **Language community should prepare**
 - Language Table
 - Exact Variant Table (EVT)
 - Typo Variant Table (TVT)
- **Support the new language:**
 - Add it to the **list** of supported languages
 - Add their **Language table**
 - Regenerate **GVT**
- **Communicate it with others (ICANN, Registries ..)**

Conclusion

- We tried to have a prototype that fulfill the concepts of script based registry that is:
 - Optimized, Simple, Transparent, Automated
- Next steps:
 - Prepare the Language tables & variant tables for all members.
 - We need more help to put the proposal in an RFC or best-practice document and communicate it with others

Acknowledgment



- **SaudiNIC**
 - Abdulaziz Al-Zoman
 - Anas Assiri
 - Mohammed Al-Hamed
 - Raed Al-Fayez
- **Outside consultant**
 - Mohamed Al-Abdulakareem (KSU)



Thank you

شكرا

شكرا

xn--mgbti28b

input[0] = U+0634
input[1] = U+06a9
input[2] = U+0631
input[3] = U+0627

xn--mgbti4d

input[0] = U+0634
input[1] = U+0643
input[2] = U+0631
input[3] = U+0627