

Introduction to Internationalized Domain Names (IDNs) and Universal Acceptance of Domain Names and Email Addresses (UA)

Sarmad Hussain
Sr. Director IDN and UA Programs

Budapest University
29 April 2021



ASCII Domain Name Label

www.cafe-123.com



②

Forming ASCII Labels Use LDH

- Letters [a..z]
- Digits [0..9]
- Hyphen (-)

Label length = 63

Other constraints (e.g. on hyphen)

①

Forming ASCII Labels Use only Letters

- Letters [a..z]

Label length = 63

Domain Name Mnemonics in ASCII

Using LDH

- Letters [a..z]
- Digits [0..9]
- Hyphen (-)

2

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(8	H	X	h	x
9	HT	EM)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

Top-Level Domain Name Mnemonics in ASCII

Using Letters only

- Letters [a..z]
- ~~Digits [0..9]~~
- ~~Hyphen (-)~~

①

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(8	H	X	h	x
9	HT	EM)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

Internationalized Domain Names



Syntax of IDN Labels

Valid U-Label: Unicode code points as constrained by the “LDH” scheme within IDNA 2008

2

Syntax of IDN Labels

Valid U-Label, further constrained by the “letter” principle for TLDs

1

IDN Mnemonics for All Actively Used Scripts

	060	061	062	063	064	065	066	067	068	069	06A	06B	06C	06D	06E	06F
0	ا	ب	پ	ذ	-	ر	ز	س	ش	ط	ظ	و	ح	خ	گ	ق
	0600	0610	0620	0630	0640	0650	0660	0670	0680	0690	06A0	06B0	06C0	06D0	06E0	06F0
1	ب	ب	ع	ر	ف	ا	ا	ا	خ	ر	ف	گ	ا	پ	ا	ا
	0601	0611	0621	0631	0641	0651	0661	0671	0681	0691	06A1	06B1	06C1	06D1	06E1	06F1
2	م	ب	آ	ز	ق	ا	ا	ا	خ	ر	ب	گ	ا	ا	ا	ا
	0602	0612	0622	0632	0642	0652	0662	0672	0682	0692	06A2	06B2	06C2	06D2	06E2	06F2
3	ص	ب	أ	س	ك	ا	ا	ا	ج	ر	ف	گ	ا	ا	ا	ا
	0603	0613	0623	0633	0643	0653	0663	0673	0683	0693	06A3	06B3	06C3	06D3	06E3	06F3
4	م	ب	ؤ	ش	ل	ا	ا	ا	ج	ر	ف	گ	و	-	ا	ا
	0604	0614	0624	0634	0644	0654	0664	0674	0684	0694	06A4	06B4	06C4	06D4	06E4	06F4
5	ا	ب	ا	ص	م	ا	ا	ا	خ	ر	پ	ل	و	ه	ر	ا
	0605	0615	0625	0635	0645	0655	0665	0675	0685	0695	06A5	06B5	06C5	06D5	06E5	06F5
6	ا	ب	ئ	ض	ن	ا	ا	ا	چ	ر	ق	ن	ؤ	ا	ا	ا
	0606	0616	0626	0636	0646	0656	0666	0676	0686	0696	06A6	06B6	06C6	06D6	06E6	06F6
7	ا	ب	ا	ط	ه	ا	ا	ا	چ	ر	ف	ل	ؤ	ا	ا	ا
	0607	0617	0627	0637	0647	0657	0667	0677	0687	0697	06A7	06B7	06C7	06D7	06E7	06F7
8	ب	ب	ب	ظ	و	ا	ا	ا	ڈ	ژ	ق	پ	ؤ	ا	ا	ا
	0608	0618	0628	0638	0648	0658	0668	0678	0688	0698	06A8	06B8	06C8	06D8	06E8	06F8

IDN Mnemonics for All Actively Used Scripts

060		061														
				120	121	122	123	124	125	126	127	128	129	12A	12B	
0			0	ሀ	ሐ	ሠ	ሰ	ቀ	ቐ	በ	ተ	ኀ	ነ	አ	ኰ	
	0600	0610		1200	1210	1220	1230	1240	1250	1260	1270	1280	1290	12A0	12B0	
1			1	ሁ	ሑ	ሡ	ሱ	ቁ	ቑ	ቡ	ቲ	ኁ	ነ	ኡ		
	0601	0611		1201	1211	1221	1231	1241	1251	1261	1271	1281	1291	12A1		
2			2	ሂ	ሒ	ሢ	ሲ	ቂ	ቊ	ቢ	ቲ	ኂ	ኑ	ኢ	ኰ	
	0602	0612		1202	1212	1222	1232	1242	1252	1262	1272	1282	1292	12A2	12B2	
3			3	ሃ	ሓ	ሣ	ሳ	ቃ	ቋ	ባ	ታ	ኃ	ና	አ	ኳ	
	0603	0613		1203	1213	1223	1233	1243	1253	1263	1273	1283	1293	12A3	12B3	
4			4	ሄ	ሔ	ሣ	ሴ	ቄ	ቋ	ቤ	ቲ	ኄ	ኑ	ኤ	ኴ	
	0604	0614		1204	1214	1224	1234	1244	1254	1264	1274	1284	1294	12A4	12B4	
5			5	ህ	ሐ	ሥ	ሰ	ቅ	ቐ	ብ	ቶ	ኀ	ን	እ	ኸ	
	0605	0615		1205	1215	1225	1235	1245	1255	1265	1275	1285	1295	12A5	12B5	
6			6	ሆ	ሐ	ሦ	ሶ	ቆ	ቐ	ቦ	ቶ	ኀ	ኖ	ኖ	አ	
	0606	0616		1206	1216	1226	1236	1246	1256	1266	1276	1286	1296	12A6		
7			7	ሀ	ሐ	ሠ	ሰ	ቀ		ባ	ታ	ኀ	ና	አ		
	0607	0617		1207	1217	1227	1237	1247		1267	1277	1287	1297	12A7		
8			8	ሀ	ሐ	ሠ	ሰ	ቀ		ባ	ታ	ኀ	ና	አ		
	0608	0618		1207	1217	1227	1237	1247		1267	1277	1287	1297	12A7		


IDN Mnemonics for All Actively Used Scripts

060		061		120		HEX	C	J	K	V	HEX	C	J	K	V	
0			0		4E50	乐					4E64	乏			乏	
	0600	0610		1200	J 4.4	G0-4056					乙 5.3	GK-6864			K2-213C	
1			1		4E51	丞	丞	丞			4E65	乏			乏	
	0601	0611		1201	J 4.5	GE-212F	T4-2227	J14-212C			乙 5.3	GK-6866			K2-213D	
2			2		4E52	兵	兵	兵	兵		4E66	书				
	0602	0612		1202	J 4.5	G0-4639	HB1-A5E2	T1-4724	K2-2136		乙 5.3	G0-4A69				
3			3		4E53	兵	兵	兵	兵		4E67	乚			乚	
	0603	0613		1203	J 4.5	G0-4552	HB1-A5E3	T1-4725	K2-2137		乙 5.4	GE-2134			K1-5D6A	
4			4		4E54	乔	乔				4E68	乚	乚	乚		
	0604	0614		1204	J 4.5	G0-4747	T3-2345				乙 5.5	G5-303F	T3-2264	J1-3037		
5			5		4E55	厝		厝			4E69	乚	乚	乚	乚	
	0605	0615		1205	J 4.6	GE-2130		J0-6949			乙 5.5	G0-5840	HB1-A5E4	T1-4726	J14-2130	K2-213E
6			6		4E56	乖	乖	乖	乖	乖	4E6A	乚	乚	乚		
	0606	0616		1206	J 4.7	G0-3954	HB1-A8C4	T1-4B65	J0-502A	K0-4E52	乙 5.5	G4-1201	H-9C57	T4-2228		
7			7		4E57	乘	乘	乘	乘	乘	4E6B	乚			乚	
	0607	0617		1207	J 4.8	GE-2131	T3-2B22	J0-3E68	K2-2138		乙 5.5	GK-6779			K0-4A61	
8			8		4E58	乘	乘	乘	乘	乘	4E6C	乚			乚	
	0608	0618		1208	J 4.9	G0-334B	HB1-A8BC	T1-537D	J0-502B	K0-632B	乙 5.5	GK-677C			K1-5B28	
					4E59	乙	乙	乙	乙	乙	4E6D	乚			乚	
					乙 5.0	G0-5252	HB1-A441	T1-4422	J0-3235	K0-6B60	乙 5.5	GK-682C			K0-544C	
					4E5A	乚	乚	乚	乚		4E6E	乚			乚	
					乙 5.0	GE-2132	H-C87B	T4-2124	J14-212F		乙 5.5	GK-6839			K2-213F	

IDN Mnemonics for All Actively Used Scripts


060 061		120 121		HEX C J				090 091 092 093 094 095 096 097											
0			0	U	ሀ	4E50	乐			0	ँ	ऐ	ठ	र	ी	ॐ	ऋ	०	
	0600	0610		1200	1210	4E50	乐				0900	0910	0920	0930	0940	0950	0960	0970	
1			1	ሀ	ሁ	4E51	丞	丞	丞	1	ँ	ऑ	ड	र	ु	ं	ऌ	·	
	0601	0611		1201	1211	4E51	丞	丞	丞		0901	0911	0921	0931	0941	0951	0961	0971	
2			2	ሀ	ሂ	4E52	兵	兵	兵	2	ं	ओ	ढ	ल	ॊ	॒	॑	अँ	
	0602	0612		1202	1212	4E52	兵	兵	兵		0902	0912	0922	0932	0942	0952	0962	0972	
3			3	ሀ	ሃ	4E53	兵	兵	兵	3	ः	ओ	ण	ळ	॑	े	॑	अ	
	0603	0613		1203	1213	4E53	兵	兵	兵		0903	0913	0923	0933	0943	0953	0963	0973	
4			4	ሀ	ሄ	4E54	乔	乔	乔	4	ऐ	औ	त	ळ	॑	ॅ	।	आ	
	0604	0614		1204	1214	4E54	乔	乔	乔		0904	0914	0924	0934	0944	0954	0964	0974	
5			5	ሀ	ህ	4E55	厖	厖	厖	5	अ	क	थ	व	ँ	ँ	॥	औ	
	0605	0615		1205	1215	4E55	厖	厖	厖		0905	0915	0925	0935	0945	0955	0965	0975	
6			6	ሀ	ሆ	4E56	乖	乖	乖	6	आ	ख	द	श	े	॒	०	अु	
	0606	0616		1206	1216	4E56	乖	乖	乖		0906	0916	0926	0936	0946	0956	0966	0976	
7			7	ሀ	ሇ	4E57	乘	乘	乘	7	इ	ग	घ	ष	े	॑	१	अु	
	0607	0617		1207	1217	4E57	乘	乘	乘		0907	0917	0927	0937	0947	0957	0967	0977	
8			8	ሀ	ለ	4E58	乘	乘	乘	8	ई	घ	न	स	ै	क	र	रु	
	0608	0618		1208	1218	4E58	乘	乘	乘		0908	0918	0928	0938	0948	0958	0968	0978	
				1209	1219	4E59	乙	乙	乙		9	उ	ड	न	ह	ॉ	ख	३	ज़
				1227	1237	4E5A	乙	乙	乙		0909	0919	0929	0939	0949	0959	0969	0979	
				1247	1257	4E5B	乙	乙	乙										
				1267	1277	4E5C	乙	乙	乙										

IDN Mnemonics for All Actively Used Scripts




ASCII
1 script
63 of 127
allowed

IDN Mnemonics for All Actively Used Scripts



Unicode 11.0
??? of 148 scripts
??? of 137,439
allowed



ASCII
1 script
63 of 127
allowed

epic
epic

epic (0065 0070 0069 0063)
epic (0435 0440 0456 0441)

“Same” or Different Domain Labels?

- ◉ Example of within-script variant labels (Arabic script)

شبكة (06C3 06A9 0628 0634)

شبكة (0629 06A9 0628 0634)

شبكة (0629 0643 0628 0634)

- ◉ Example of within-script variant labels (Simplified Chinese and Traditional Chinese)

名称 (540D 79F0)

名稱 (540D 7A31)

- ◉ Example of cross-script variant label (Latin script and Cyrillic script)

epic (0065 0070 0069 0063)

epic (0435 0440 0456 0441)

Root Zone Label Generation Rules (LGR) Procedure

Generation Panels

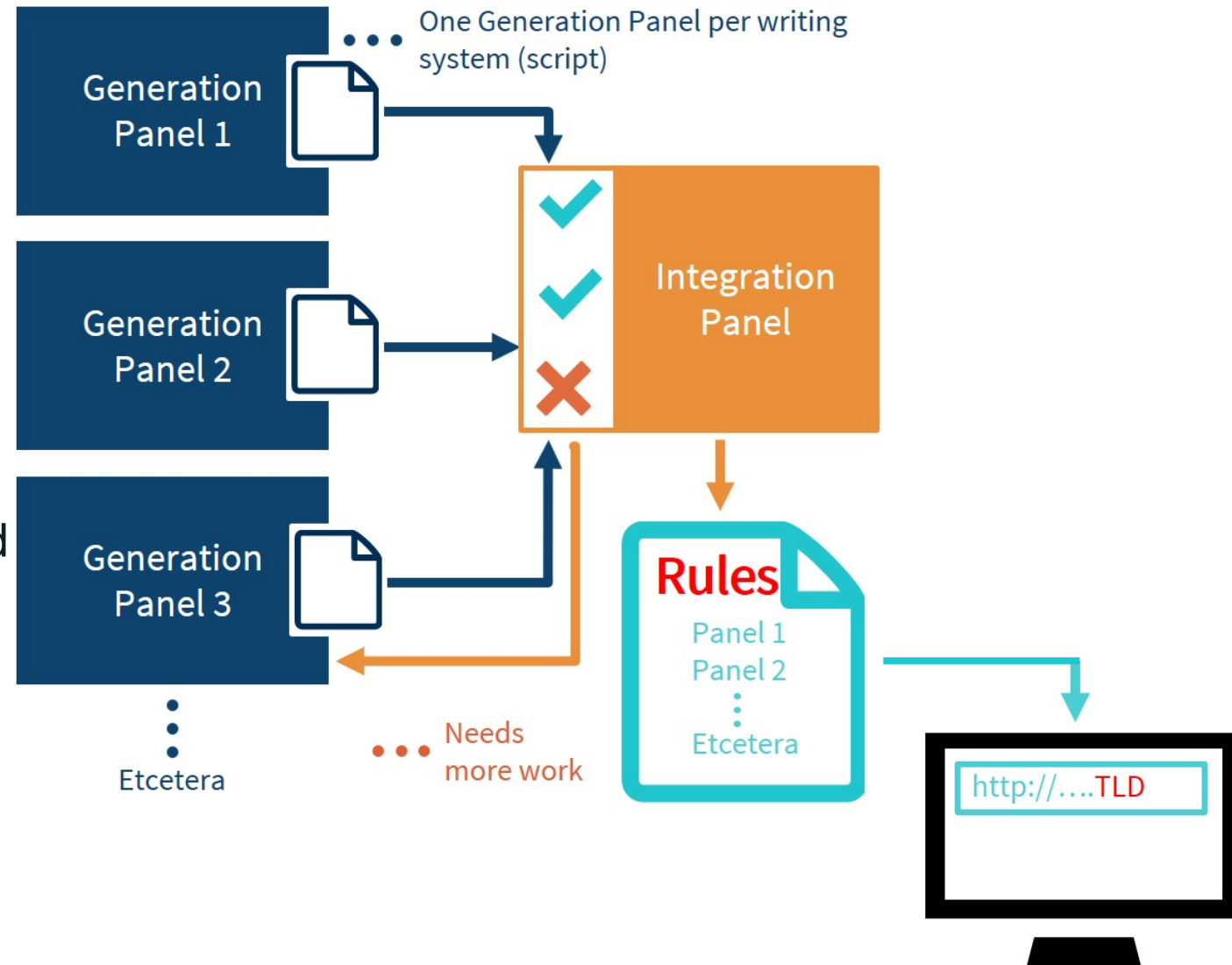
- Generate proposals for script specific LGRs, based on community expertise and linguistic, security and stability requirements

Integration Panel

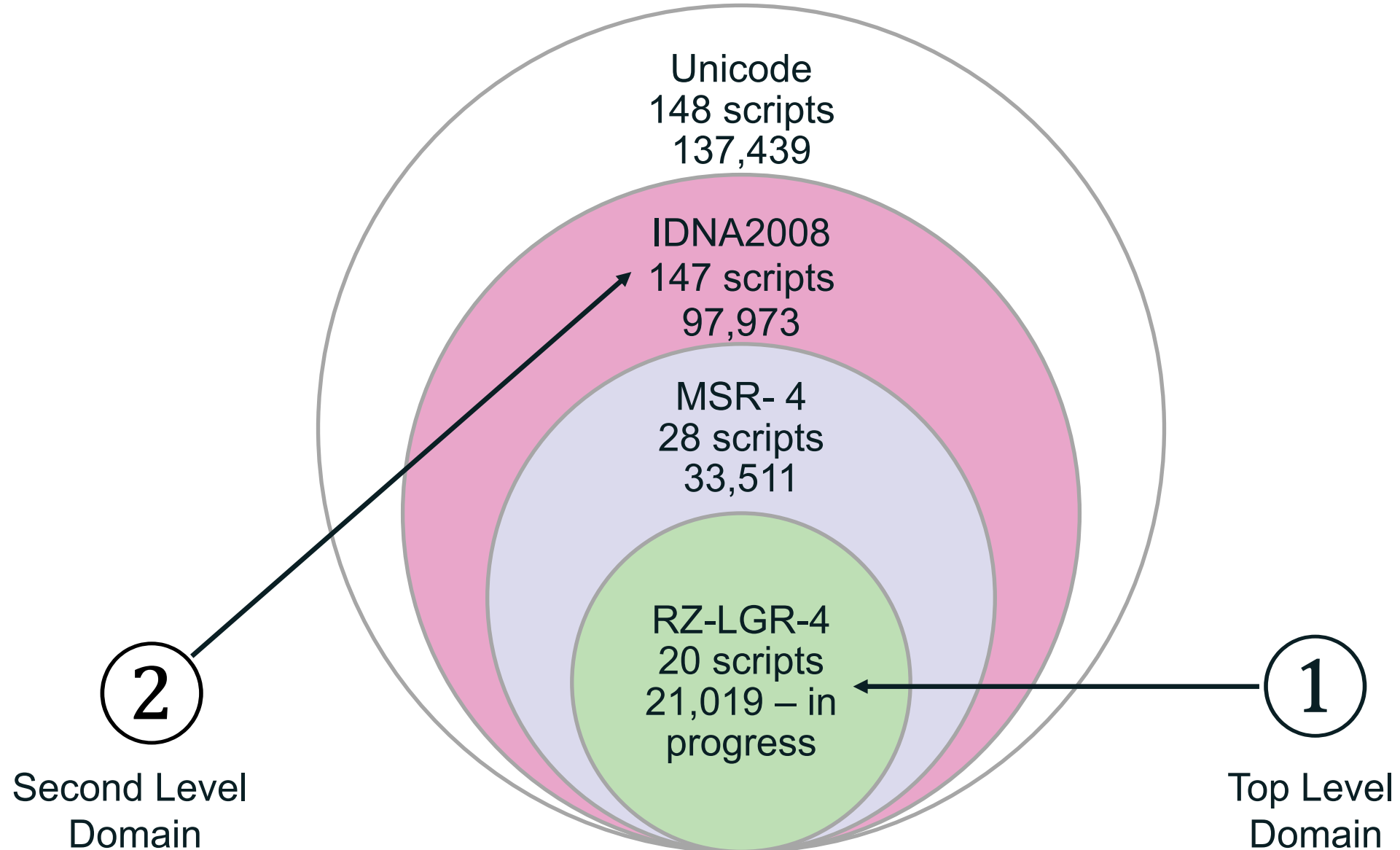
- Integrates them into common Root Zone LGR while minimizing the risk to Root Zone as a shared resource

Label Generation Rules (LGR)

- Which labels are permissible
- Which variant labels exist
- Are there any more constraints?

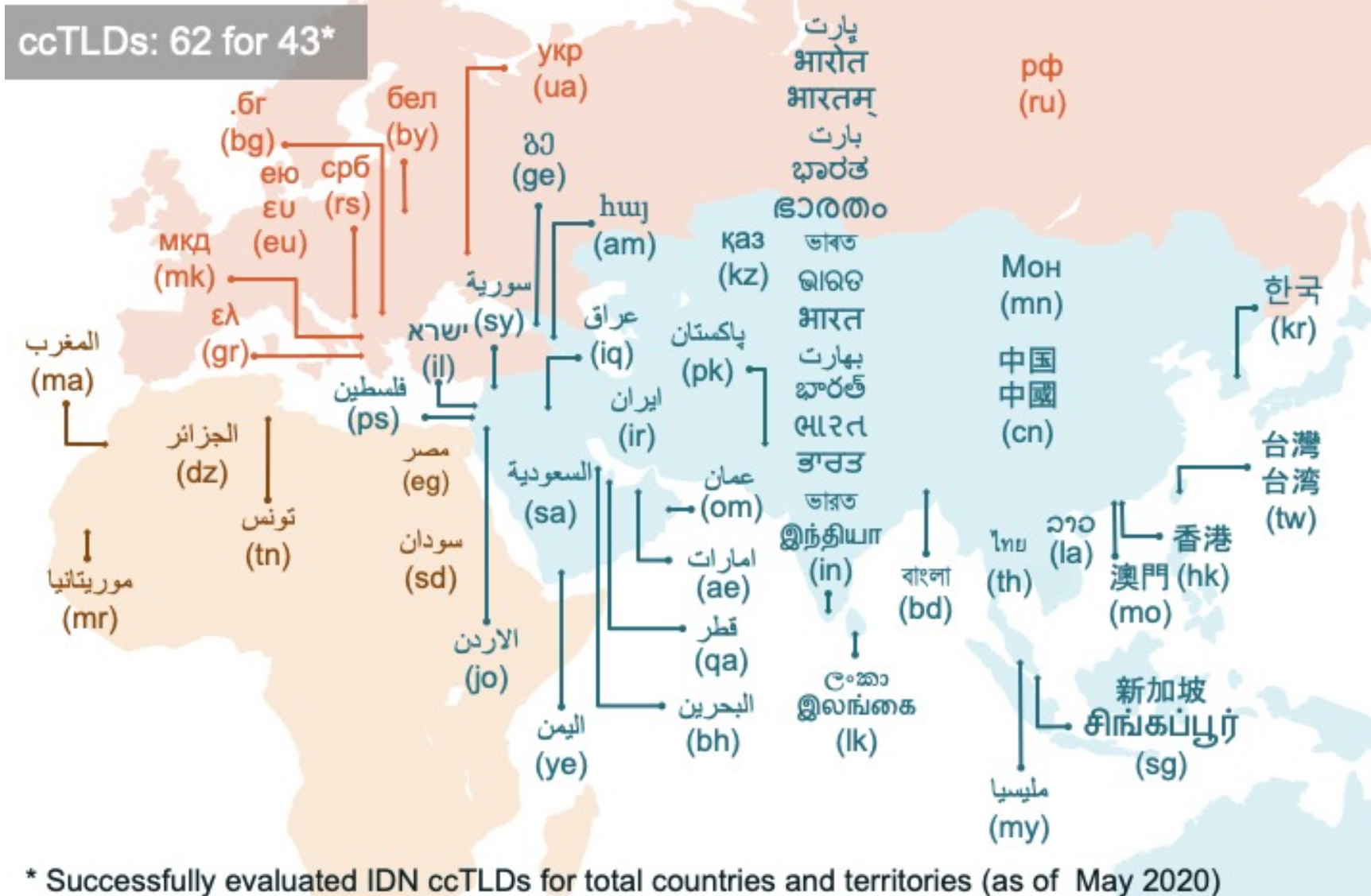


Scope of Code Point Repertoires for Domain Labels



	060	061	062	063	064	065	066	067	068	069	06A	06B	06C	06D	06E	06F	075	076	077	08A	08B	08C	08D	08E	08F
0	ا	ب	ي	ذ	-	و	و	و	پ	ڈ	غ	گ	ة	ي	و	و	ب	ب	ب	ب	ب	ب	ب	ب	ب
1	ا	ب	ء	ر	ف	و	ا	ا	خ	ز	ف	گ	ه	ي	و	ا	ب	ب	ب	ب	ب	ب	ب	ب	ب
2	ا	ب	آ	ز	ق	و	ا	ا	خ	ز	ب	گ	ا	و	و	ا	ب	ب	ب	ب	ب	ب	ب	ب	ب
3	ا	ب	أ	س	ك	و	ا	ا	ج	ر	ف	گ	ا	و	و	ا	ب	ب	ب	ب	ب	ب	ب	ب	ب
4	ا	ب	و	ش	ل	و	ا	ا	ج	ر	ف	گ	و	-	و	ا	ب	ب	ب	ب	ب	ب	ب	ب	ب
5	ا	ب	ا	ص	م	و	ا	ا	خ	ر	پ	ل	و	و	ر	ا	ب	ب	ب	ب	ب	ب	ب	ب	ب
6	ا	ب	ئ	ض	ن	و	ا	ا	ج	ر	ق	ل	و	و	-	ا	ب	ب	ب	ب	ب	ب	ب	ب	ب
7	ا	ب	ا	ط	ه	و	ا	ا	ج	ر	ف	ل	و	و	و	ا	ب	ب	ب	ب	ب	ب	ب	ب	ب
8	ا	ب	ب	ظ	و	و	ا	ا	ڈ	ر	ق	ل	و	و	و	ا	ب	ب	ب	ب	ب	ب	ب	ب	ب
9	ا	ب	ة	ع	ي	و	ا	ا	ڈ	ر	ك	ن	و	و	و	ا	ب	ب	ب	ب	ب	ب	ب	ب	ب
A	ا	ب	ت	غ	ي	و	ا	ا	ب	ب	ك	ن	و	و	و	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب
B	ا	ب	ث	ك	و	و	ا	ا	ب	ب	ك	ن	و	و	و	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب
C	ا	ALM	ج	ك	و	و	ا	ا	ب	ب	ك	ن	و	و	و	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب
D	ا		ح	ئ	و	و	ا	ا	ب	ب	ك	ن	و	و	و	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب
E	ا	ا	خ	ئ	و	و	ا	ا	ب	ب	ك	ن	و	و	و	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب
F	ا	ا	د	ئ	و	و	ا	ا	ب	ب	ك	ن	و	و	و	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب

IDN Country Code Top-Level Domains



Quiz

- ⦿ A Unicode character is categorized as a letter, digit, mark, punctuation or symbol. Which of these are generally allowed for Internationalized domain names (IDNs)?
 1. Only letters.
 2. Letters and digits.
 3. Letters, digits and marks.
 4. Letters, digits and symbols.
 5. All categories of characters.

Universal Acceptance of Domain Names and Email Addresses (UA)

Goal

All domain names and email addresses work in all software applications.

Impact

Promote consumer choice, improve competition, and provide broader access to end users.

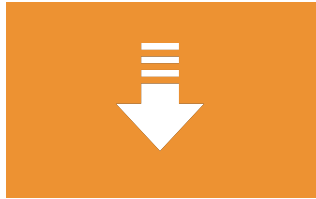
⦿ Domain Names:

- **Newer** top-level domain names: example.**sky**
- **Longer** top-level domain names: example.**international**
- **Internationalized** domain names: **คน.ไทย**

⦿ Internationalized email addresses (EAI):

- ASCII@ASCII (new and long TLD) ekrem@misal.**berlin**
- ASCII@IDN marc@**société**.org
- **Unicode@ASCII** **测试**@example.com
- **Unicode@IDN** **пример**@**тестовая-зона**.рф
- **Unicode@IDN**; right to left scripts **ای-میل**@**مثال**.موقع

1. Support All Domain Names



Accept



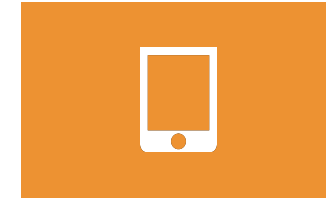
Validate



Process



Store



Display

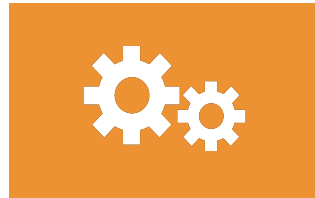
2. Support All Email Addresses



Accept



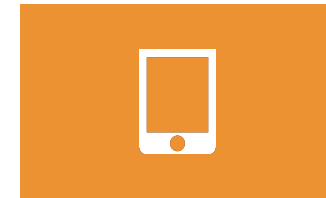
Validate



Process



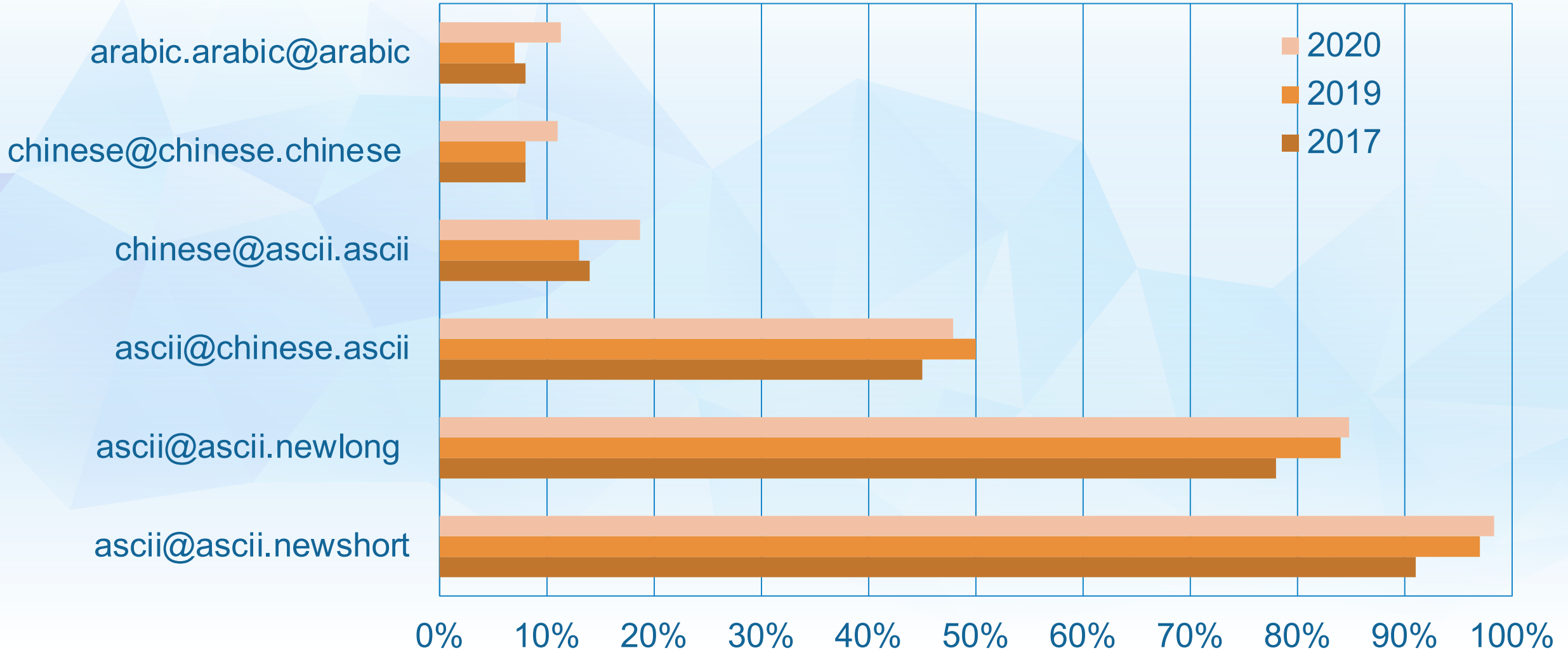
Store

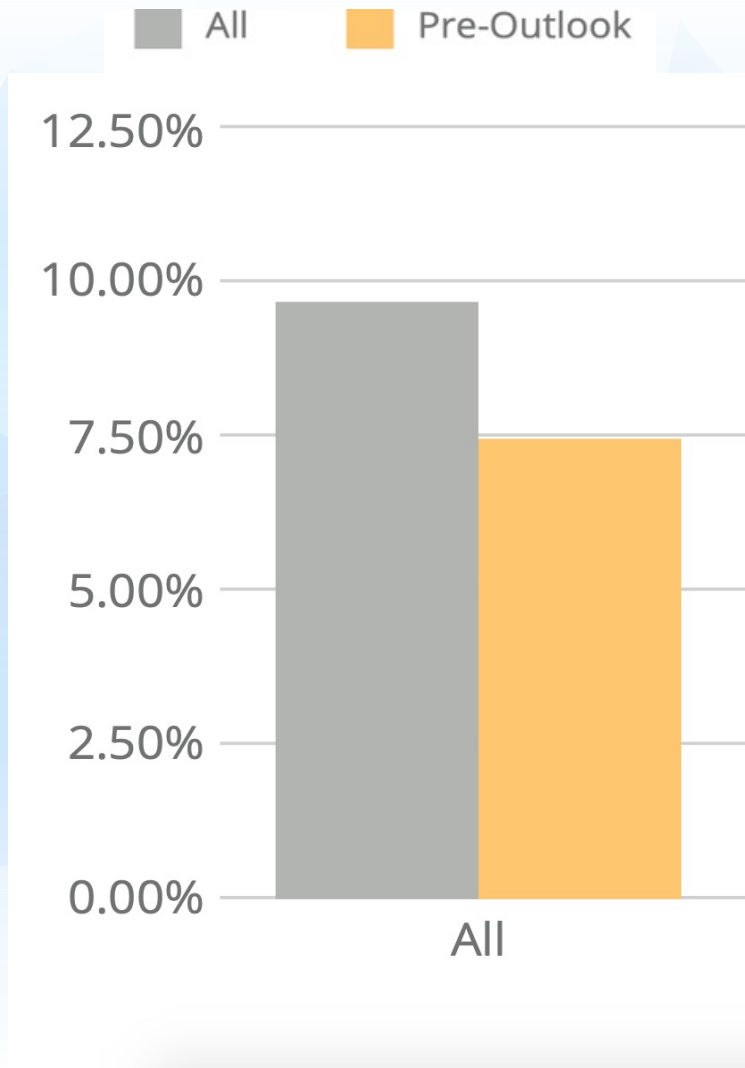


Display

Acceptance of Email Addresses by Websites Globally

For details, see [UASG027](#)





Only 9.7% of the domains sampled were EAI ready in 2019.

This is based on mail servers found through MX records in zones of all top-level domains (TLDs).

For details on methodology, see [UASG021D: EAI Readiness in TLDs](#)

Applications and Websites

- Wikipedia.org, ICANN.org, Amazon.com, custom websites globally
- PowerPoint, Google-Docs, Safari, Acrobat, custom apps

Social Media and Search Engines

- Chrome, Bing, Safari, Firefox, local (e.g., Chinese) browsers
- Facebook, Instagram, Twitter, Skype, WeChat, WhatsApp, Viber

Programming Languages and Frameworks

- JavaScript, Java, Swift, C#, PHP, Python
- Angular, Spring, .NET core, J2EE, WordPress, SAP, Oracle

Platforms, Operating Systems and System Tools

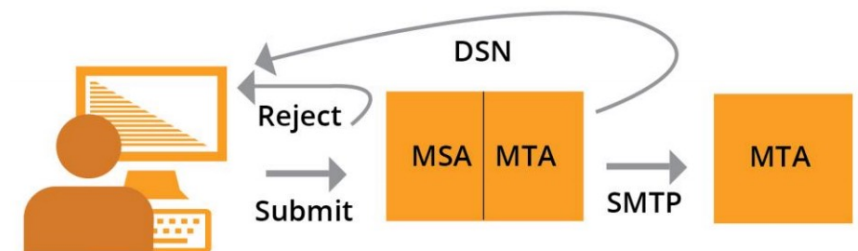
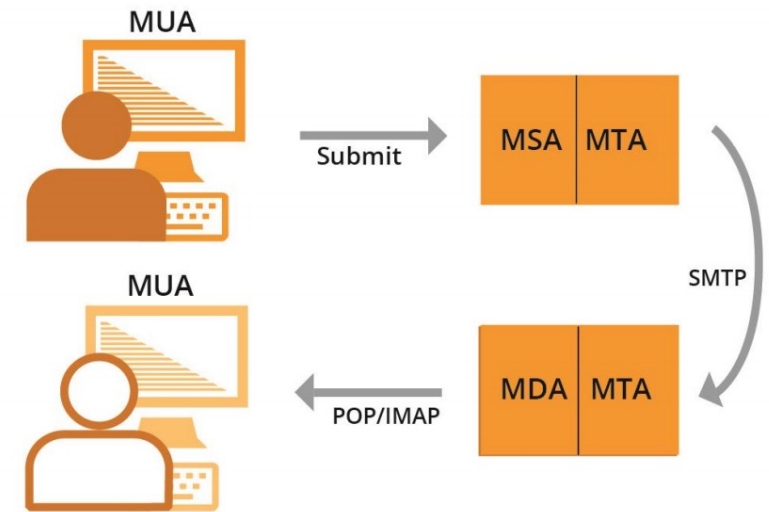
- iOS, Windows, Linux, Android, App Stores
- Active Directory, OpenLDAP, OpenSSL, Ping, Telnet

Standards and Best Practices

- IETF RFCs, W3C HTML, Unicode CLDR, WHATWG
- Industry-based standards (health, aviation, ...)

UA Readiness needs to be checked and fixed (as needed) for multiple frameworks, utilities, tools, and applications at multiple layers of technology

- All email agents must be configured to send and receive internationalized email addresses. See [EAI: A Technical Overview](#) for details.
 - **MUA** – Mail User Agent: A client program that a person uses to send, receive, and manage mail.
 - **MSA** – Mail Submission Agent: A server program that receives mail from a MUA and prepares it for transmission and delivery.
 - **MTA** – Mail Transmission Agent: A server program that sends and receives mail to and from other Internet hosts. An MTA may receive mail from an MSA and/or deliver mail to an MDA.
 - **MDA** – Mail Delivery Agent: A server program that handles incoming mail and typically stores it in a mailbox or folder.



Quiz

- ⦿ To enhance systems to be Universal Acceptance (UA) ready, which of the following categories of domain names and email addresses are relevant?
 1. ASCII domain names.
 2. Internationalized Domain Names (IDNs).
 3. Internationalized email addresses (EAI).
 4. All the above.
 5. Only 2 and 3.

Fundamentals for Internationalized Domain Names and Email Addresses

- ⦿ Unicode encodes glyphs into codepoints for different scripts of the world.
 - Codepoints shown in hex using the U+XXXX notation.
 - Unicode files typically in UTF8 format, using a variable number of bytes for a codepoint.
 - ASCII is used as is in Unicode: `e = ASCII 65 = U+0065`.
- ⦿ There are multiple ways to encode certain glyphs in Unicode:
 - `è = U+00E8`
 - `e + ` = è = U+0065 + U+0300`
- ⦿ Normalization ensures that the end representation is the same, even if users type differently.
 - IDN standards recommend using [Normalization Form C \(NFC\)](#).
 - Generates `U+00E8` for both input versions above.

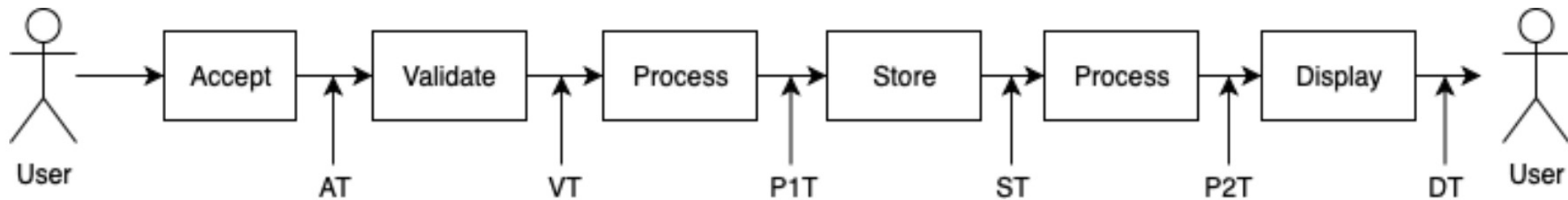
- ⦿ A domain name is an ordered set of labels or strings: www.example.co.uk.
 - The top-level domain (TLD) is the rightmost label: "uk"
 - Initially, TLDs were only two or three characters long (e.g., .ca, .com).
 - Now TLDs can be longer strings (e.g., .info, .google, .engineering).
 - TLDs delegated in the [root zone](#) can change over time, so a fixed list can get outdated.
- ⦿ Domain names can also be internationalized when one of the labels contains at least one non-ASCII character.
 - For example: www.exâmples.ca or [普遍接受-测试.世界.](#)
- ⦿ Use the latest IDN standard called IDNA2008 for IDNs.
 - Do not use libraries for the outdated IDNA2003 version.

- ⦿ There are two equivalent forms of IDN domain labels: U-label and A-label.
 - Human users use the IDN version called U-label (using UTF-8 format): [exâmp^le](#)
 - Applications or systems internally use an ASCII equivalent called A-label:
 1. Take user input and normalize and check against IDNA2008 to form IDN U-label.
 2. Convert U-label to punycode (using RFC3492).
 3. Add the “xn--” prefix is added to identify the ASCII string as an IDN A-label.
 - [exâmp^le](#) => [exmp^le-xta](#) => [xn--exmp^le-xta](#).
 - [普遍接受-测试](#) => [--f38am99bqvcd5liy1cxsg](#) => [xn----f38am99bqvcd5liy1cxsg](#).
- ⦿ Email address syntax: [mailboxName@domainName](#)
 - EAI has the mailboxName in Unicode (in UTF8 format).
 - The domainName can be ASCII or IDN.
 - For example: [k^evin@example.org](#) or [すし@快手.游戏](#).

- ⦿ Some applications are still verifying domain names incorrectly by using one of the outdated methods:
 - Check for a fixed length of TLD between 2-4 characters (TLD can be up to 63 characters).
 - Check from a fixed set of TLDs, e.g., using static list of strings.
 - Check for only ASCII characters.

- ⦿ Some applications do not cater to additional requirements for validating EAI:
 - Check mailbox name to be a valid string in UTF-8 format.
 - DomainName can be ASCII or IDN.

- ⦿ Based on [UASG026](#), the application components can be generalized to put emphasis on the processing of internationalized identifiers.
- ⦿ Each gate has its own set of requirements and processing.



- ⦿ AT: Accept test
- ⦿ VT: Validate test
- ⦿ P1T: Process test on the input
- ⦿ ST: Store test
- ⦿ P2T: Process test on the output
- ⦿ DT: Display test

- ⦿ Validating user input, or any input, is extremely useful for various reasons, some of which include: a better user experience, increased security, and avoiding irrelevant issues.
- ⦿ Validating domain names and email addresses is useful.
- ⦿ Some validation methods for domain names and email addresses:
 - Basic syntax checks: is the syntax of the string correct?
 - Does the domain name contain '.' ?
 - Does the email address contain '@' and a valid domain name part?
 - Functional checks: does the domain name or email address work?
 - Is the top-level domain (TLD) in use?
 - Is the whole domain name in use?
 - Is the email in use?

- ⦿ Validating syntax:
 - ASCII: RFC1035
 - Composed of letters, digits, and hyphen.
 - Max length is 255 octets with each label up to 63 octets.
 - IDN: IDNA2008 (RFCs 5890-5894)
 - Valid A-labels
 - Valid U-labels

- ⦿ Validating function:
 - Is the top-level domain (TLD) in use?
 - Verify against the list of TLDs.
 - Verify using a DNS request.
 - Is the whole domain name in use?
 - Verify using a DNS request.

- ⦿ After validation, a software would then use the domain name identifier as:
 - A domain name to be resolved in the DNS.
- ⦿ Therefore, to be UA compliant, the software has to use proper methods that support UA.
 - For example, passing a U-Label to the traditional functions or methods may not succeed, as it is not expecting a UTF8 domain name.

- ⦿ An email address is composed of: mailboxName@domainName
- ⦿ Validating syntax:
 - For domainName, see earlier discussion.
 - For mailboxName:
 - ASCII
 - UTF8 (for EAI)
- ⦿ Validating function:
 - Is the domain name set up to send and receive emails?
 - Is the mailbox name able to send and receive emails?

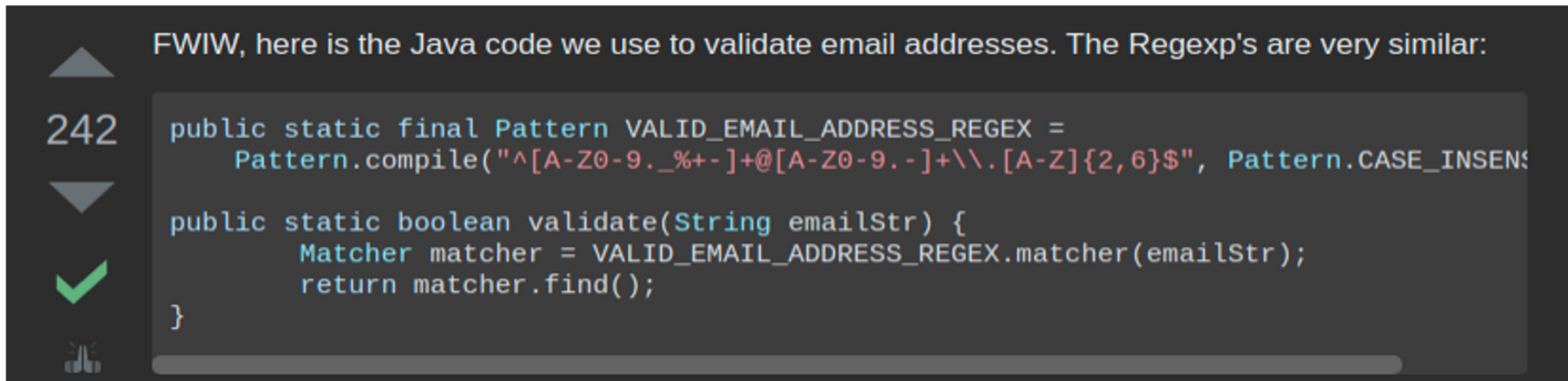
- ⦿ After validation, a software would then use the email identifier as:
 - An email-address based user id.
 - To send an email.
- ⦿ Therefore, to be UA compliant, the software must use proper methods that support UA.
 - For example, passing an UTF8 mailbox name email address to a mail sender may not succeed, as it is not expecting a UTF8 mailbox name in the email address.

- ◉ Basic: something@something
 - `^(.+)+@(.+)$`
- ◉ From owasp.org (security):
 - `[^[a-zA-Z0-9_+&*~]+(?:\.[a-zA-Z0-9_+&*~]+)*@(?:[a-zA-Z0-9-]+\.)+[a-zA-Z]{2,7}$`
 - Does not support EAI, i.e., mailbox name in UTF8 not allowed: `[a-zA-Z0-9_+&*~]`
 - Does not support ASCII TLD longer than 7 characters: `[a-zA-Z]{2,7}`
 - Does not support U-labels in IDN TLD: `[a-zA-Z]`
 - But OWASP is THE reference for security.
 - Therefore, you may end up fighting with your security team to use a UA-compatible Regex instead of the “standard” one from OWASP.

- ⦿ A comprehensive list of UA test cases is documented in [UASG004](#).
- ⦿ Developers are strongly encouraged to use these test cases in its unit and system testing.

Quiz

- A company built a website where international consumers can subscribe via their email. Since the subscription form is user input, developers validated the email address before trying to send the email.
 - Developers went to Stackoverflow and found a regular expression (regex) to perform the validation:



FWIW, here is the Java code we use to validate email addresses. The Regexp's are very similar:

```
public static final Pattern VALID_EMAIL_ADDRESS_REGEX =
    Pattern.compile("^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,6}$", Pattern.CASE_INSENSITIVE);

public static boolean validate(String emailStr) {
    Matcher matcher = VALID_EMAIL_ADDRESS_REGEX.matcher(emailStr);
    return matcher.find();
}
```

- The regex limits mailbox to letters A-Z, digits 0-9, and some symbols, the domain labels to letters, digits and hyphen, and the top-level domain to letters only with length 2-6.
- Would this regex work for the company's website? Why or why not?

Prog. Languages' UA Support

UASG018A

LANGUAGE	LIB NAME	COMPLIANCE (%)	Type
Javascript	Idna-Uts46	85.5	IDN
Javascript	Nodemailer	84.3	Mail
Javascript	Validator	94.2	Mail
Python3	Django_Auth	48.1	Mail
Python3	Email_Validator	86.3	Mail
Python3	Encodings_Idna	67.7	IDN
Python3	<u>Idna</u>	100	IDN
Python3	<u>Smtplib</u>	84.3	Mail
Rust	<u>Idna</u>	87.1	IDN
Rust	<u>Lettre</u>	7.8	Mail

LANGUAGE	LIB NAME	COMPLIANCE (%)	Type
C	Libcurl	84.3	Mail
C	Libidn2	95.2	IDN
C#	Mailkit	84.3	Mail
C#	Microsoft	83.9	IDN
Go	Mail	100	Mail
Go	<u>Idna</u>	79	IDN
Go	Smtplib	19.6	Mail
Java	Commons-Validator	85.5	Mail, IDN
Java	Guava	77.8	IDN
Java	ICU	93.5	IDN
Java	Jakartamail	82.4	Mail
Java	JRE	71	IDN

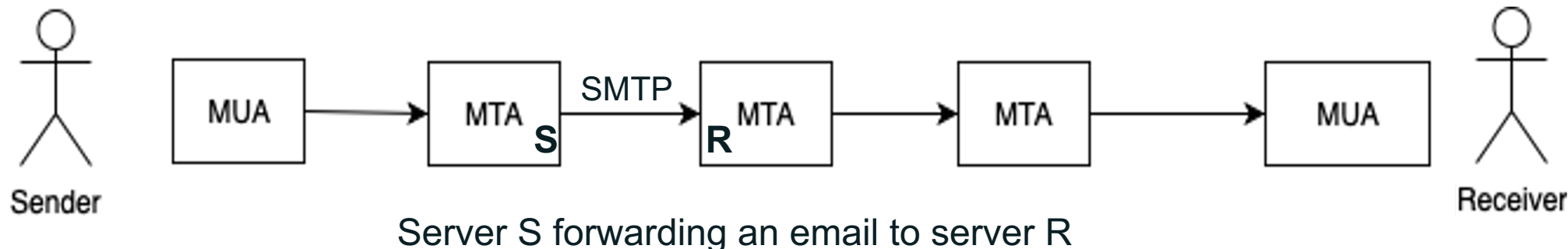
- ⦿ Be aware that UA identifiers may not be fully supported in software and libraries.
- ⦿ Use the right libraries and frameworks.
- ⦿ Adapt your code to properly support UA.
- ⦿ Do unit and system testing using UA test cases to ensure that your software is UA ready.

Email Address Internationalization (EAI)

- ⦿ SMTP
 - Is augmented to support EAI.
 - Has a signaling flag (SMTPUTF8) to specify support of EAI.
 - All SMTP servers in the path must support EAI to successfully deliver the email.

- ⦿ POP/IMAP
 - Are augmented to properly support EAI.
 - Have a signaling flag to specify support of EAI.

SMTPUTF8 Example



S: <connect>

R: 220 receive.net ESMTP

S: EHLO sender.org

R: 250-8BITMIME

R: 250-**SMTPUTF8**

R: 250 PIPELINING

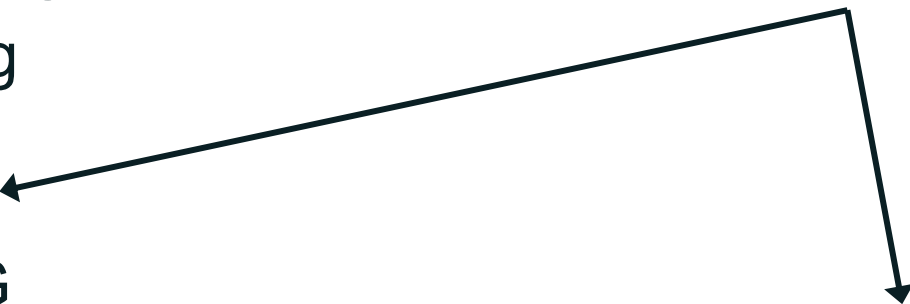
S: MAIL FROM:<猫王@普遍接受-测试.世界> **SMTPUTF8**

R: 250 Sender accepted

S:RCPT TO:<ray@receive.net>

R:250 Recipient accepted

Specific SMTPUTF8 Signaling
(EAI support)



S:DATA

R:354 Send your message

S:From: 猫王 <猫王@普遍接受-测试.世界>

S:To: ray@receive.net

S:Subject: 我们要吃午饭吗?

S:

S:How about lunch at 12:30?

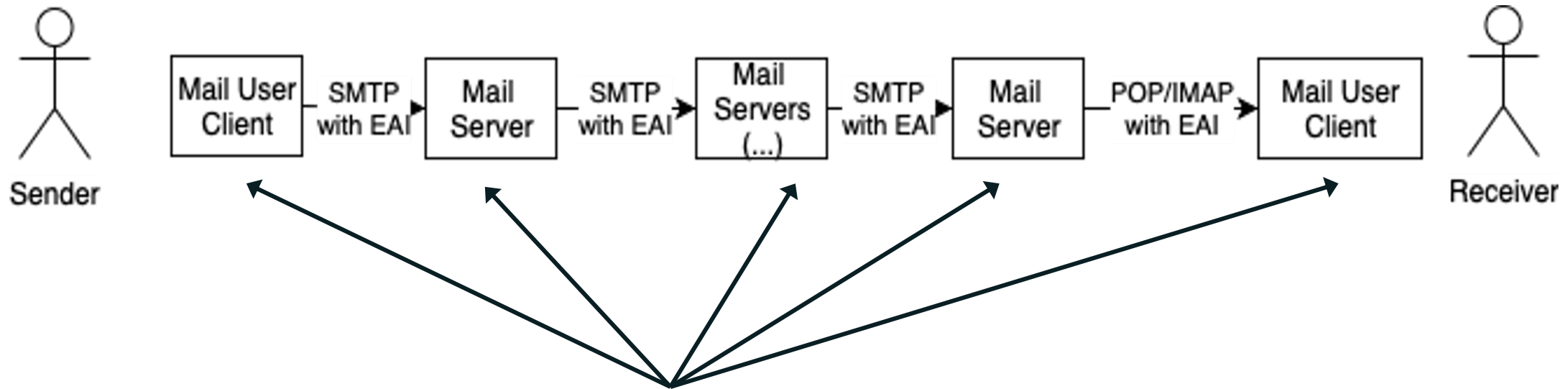
S:.

R:250 Message accepted 389dck343fg34

S:QUIT

R:221 Sayonara

} Email itself



To send and receive an email with EAI:

- All email parties involved in the delivery path have to be updated for EAI support.
- If a single SMTP server in the path does not support EAI, then the email is not delivered.

- What happens when one email (SMTP) server in the path does not support EAI?
 - The last server trying to send to the next hop:
 - Sends back to the sender user a report of unable to deliver.
 - Drops the email.
 - Similar to reports that a sender receives when an email address does not exist.

- ⦿ Case folding:
 - In ASCII, email users expect the equivalence of lowercase and uppercase.
 - Example: PETER@example.com and peter@example.com will be delivered to the same mailbox.
 - Typically for EAI, the case folding functionality is not implemented in most EAI-ready software.
- ⦿ SPAM:
 - EAI emails may be considered as spam by spam filtering software even when proper records are enabled.
- ⦿ Software/Services:
 - Not every server/client software and services support EAI.

EAI Support

Components of Email System:

- Mail User Agent
- Mail Submission Agent
- Mail Transfer Agent
- Mail Delivery Agent
- Mail Service Provider

L1 - EAI level 1 - sends to and receives from EAI addresses

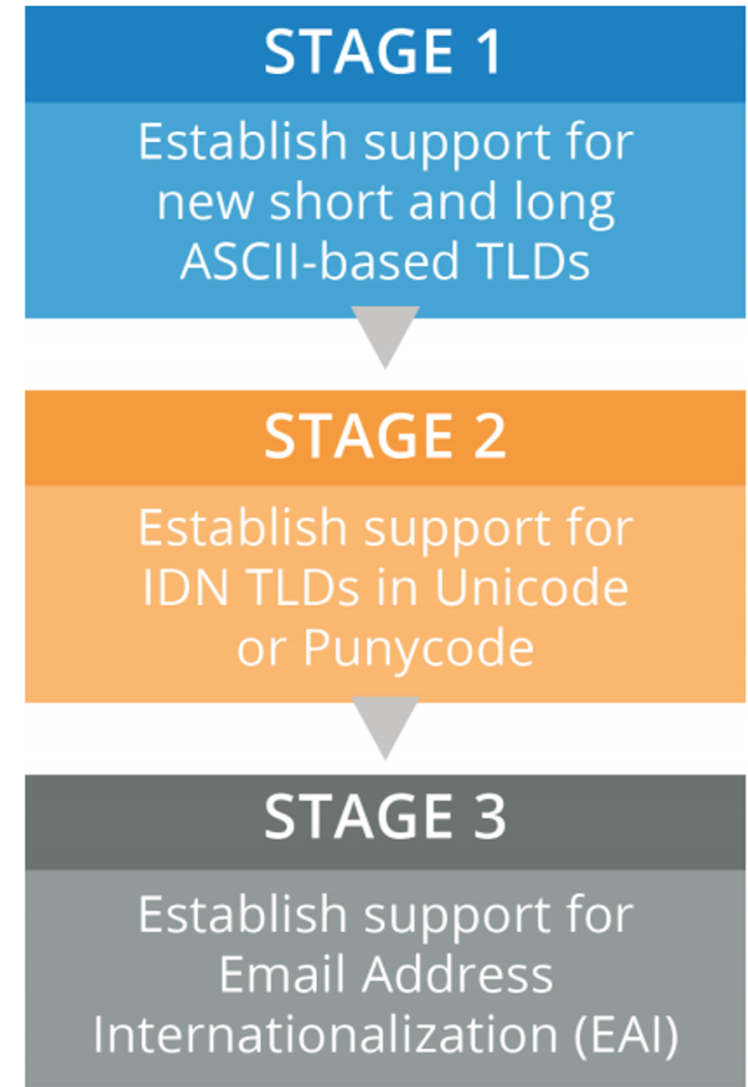
L2 - EAI level 2 - L1 plus provides local EAI addresses

[UASG030](#)

Name	MUA	MSA	MTA	MDA	MSP	Web mail
Coremail	Few	All L2	Most L2	Few	All L2	Most L2
MS Outlook.com	Most L1	Most L1	Most L1	None	None	Most L1
Yandex Mail	Few	None	None	Few	Part	Few
Roundcube	Most L2					
Apple Mail	Few					
Mozilla Thunderbird	Few					
MS Outlook	Most L1					
MS Exchange Server (hosted)		All L1	All L1			
Exim		Most L2	All L2			
Postfix		All L2	All L2			
Sendmail		Not tested	Not tested			
Fetchmail				Not tested		
Courier		All L2	All L2	All L2		
Gmail	All L1	All L1	All L1	Few		
XgenPlus		Not tested	Not tested	Not tested	All L2	Not tested

Are Your Software Applications UA Ready?

- ◉ Stage 1: Update services to support both new short and long ASCII TLDs.
- ◉ Stage 2: Update services to support non-ASCII Internationalized Domain Names (IDNs) in Unicode (U-label), and ASCII-based IDN representations in Punycode (A-label).
- ◉ Stage 3: Update infrastructure and services to support non-ASCII email addresses.
 - ◉ Note: all components must support Email Address Internationalization (EAI) before infrastructure is compliant.
- ◉ See details in [ICANN's Case Study](#)



- UASG and ICANN continue to undertake gap analysis, remediation, training and outreach:
 - Gap analysis – Social Media, Browsers, Programming Languages, EAI Tools, etc.
 - Remediation – engaging technology forums (e.g. Github) and bug reporting.
 - Training and outreach – through local initiatives and ambassadors.

We request the community to help address UA readiness and lead by example:

1. **Raise awareness** of the technical problems within the community.
2. **Upgrade and use UA ready systems** as a community to create the necessary demand, e.g. upgrade email servers, use email in local language.
3. **Advocate more broadly** to support UA in their systems (e.g. in e-govt. services; the private sector organizations, etc.).

Such activities may be undertaken in collaboration with UA Local Initiative and UA Ambassadors.

- See <https://uasg.tech> for a complete list of reports.
 - Universal Acceptance Quick Guide: [UASG005](#)
 - Introduction to Universal Acceptance: [UASG007](#)
 - Quick Guide to EAI: [UASG014](#)
 - EAI – A Technical Overview: [UASG012](#)
 - EAI – Evaluation of Major Email Software and Services: [UASG021B](#)
 - Universal Acceptance Readiness Framework: [UASG026](#)
 - Considerations for Naming Internationalized Email Mailboxes: [UASG028](#)
 - UA Readiness Report 2020: [UASG029](#)
 - Evaluation of EAI Support in Email Software and Services Report: [UASG030](#)
 - Frequently Asked Questions (FAQs): UA Readiness of Programming Languages and Email Tools: [UASG031](#)

- Please email info@uasg.tech or UAProgram@icann.org for further information.

Get Involved!

- ◉ Join [APAC EAI Implementers' Group](#) mailing list for technical support (by THNIC)

- ◉ For more information on UA, email info@uasg.tech or UAProgram@icann.org
- ◉ Access all UASG documents and presentations at: <https://uasg.tech>

- ◉ Access details of ongoing work from wiki pages: <https://community.icann.org/display/TUA>
- ◉ Register to participate or listen in the UA discussion list at: <https://uasg.tech/subscribe>
- ◉ Register to participate in UA working groups [here](#).

Engage with ICANN – Thank You and Questions



One World, One Internet

Visit us at icann.org

Email: sarmad.hussain@icann.org



[@icann](https://twitter.com/icann)



facebook.com/icannorg



youtube.com/icannnews



flickr.com/icann



linkedin/company/icann



slideshare/icannpresentations



soundcloud/icann