Universal Acceptance of Domain Names and Email Addresses (UA)

A Technical Overview

Sarmad Hussain and Champika Wijayatunga

AccessPlus Webinar Series on UA, Uganda
30 September 2021

Universal Acceptance of Domain Names and Email

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.

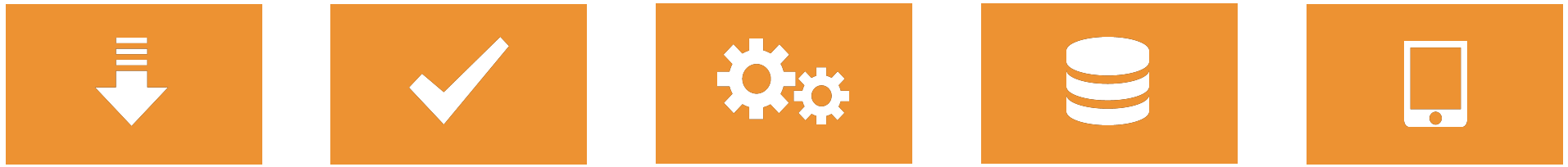
Categories Affected and UA Readiness

- **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.africa
  - 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

Scope of UA Readiness

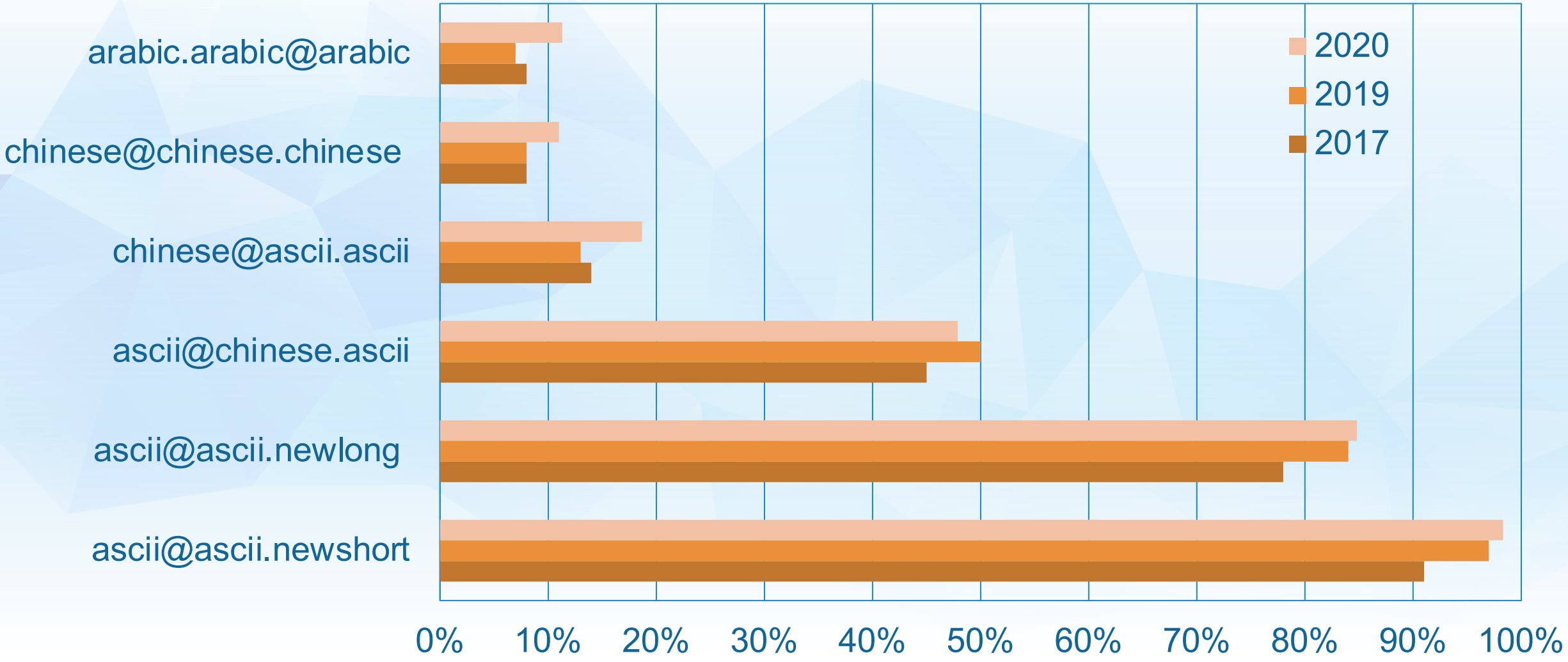
1. Support All Domain Names

2. Support All Email Addresses

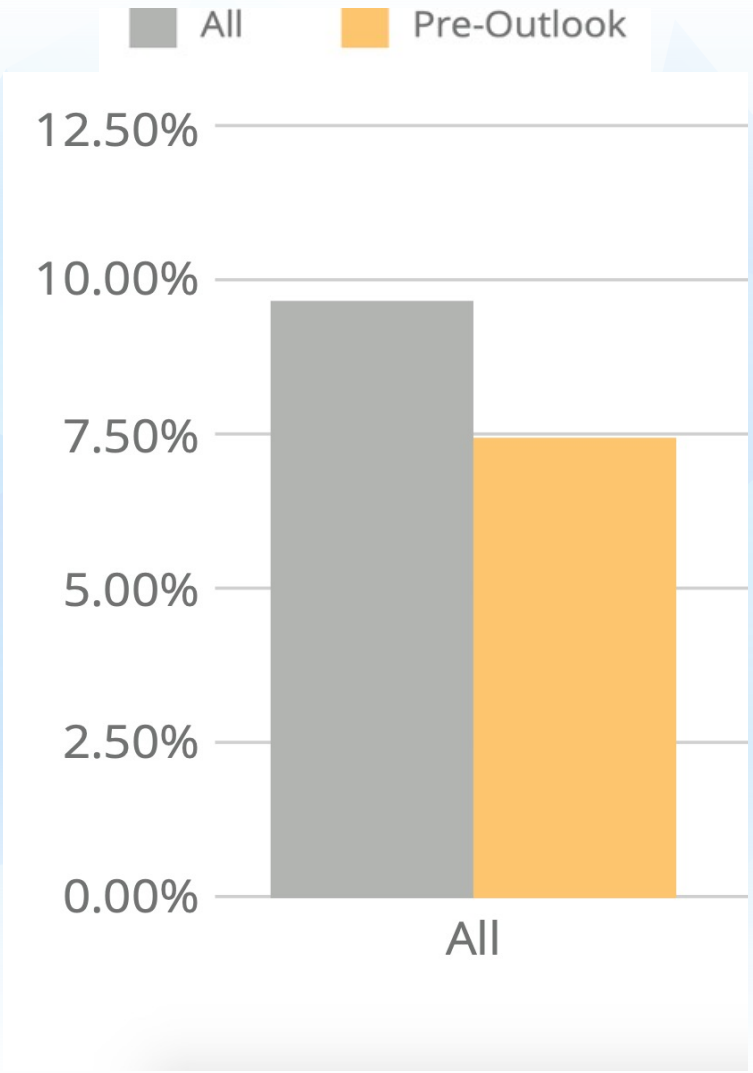


Acceptance of Email Addresses by Websites Globally

For details, see UASG027



Estimated Support of EAI in Email Systems Under All TLDs



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

Technology Stack for UA Consideration

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 Sytem Tools

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

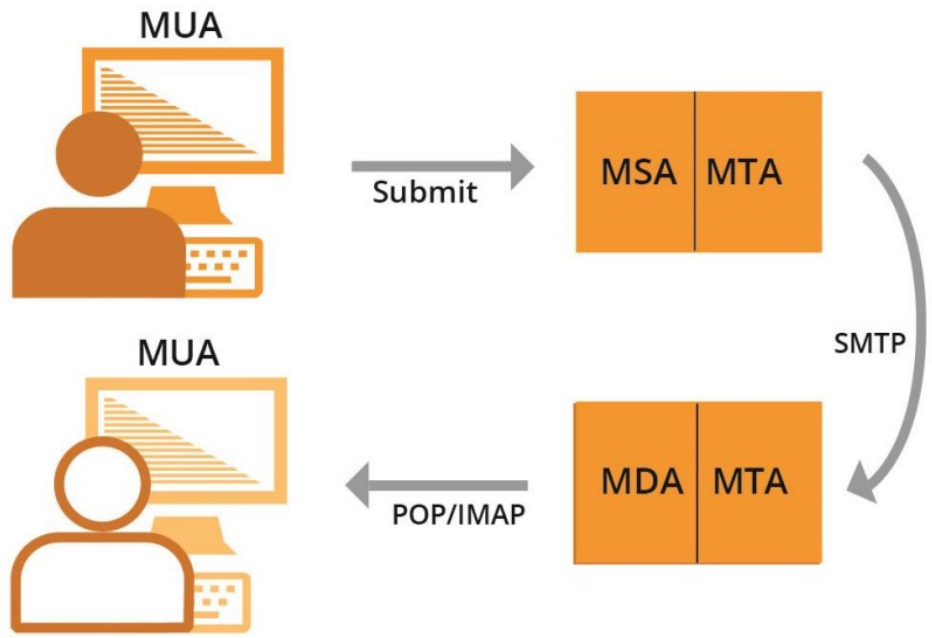
Standards and Best Practices

- 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

Email Systems and EAI Support

- 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 Transfer 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.



Fundamentals for Internationalized Domain Names and Email Addresses

Unicode Encoding

- 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.

Domain Names and Internationalized Domain Names (IDNs)

- 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.exâmple.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.

IDNs and EAI

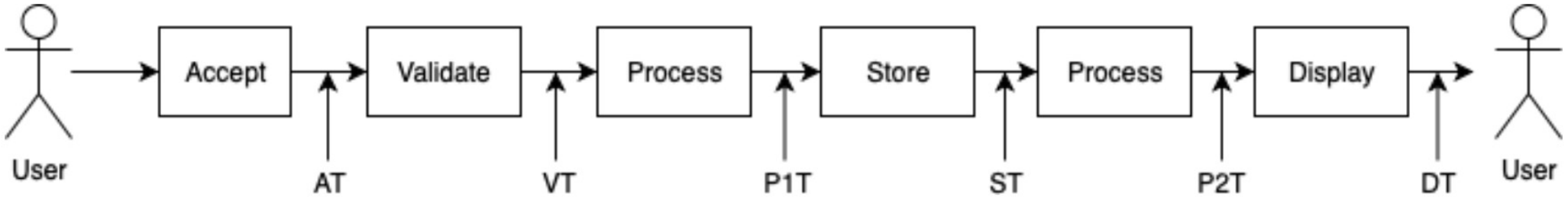
- There are two equivalent forms of IDN domain labels: U-label and A-label.
  - Human users use the IDN version called U-label (using UTF-8 format): exâmple
  - Applications or systems internally use an ASCII equivalent called A-label:
    1. Take user input and normalize and check against IDNA2008 to 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.
       - exâmple => exmple-xta => xn--exmple-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: kévin@example.org or すし@快手.游戏.

Universal Acceptance Issues

- 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.

UA Readiness Testing Framework

- Based on UASG026, the application components are generalized to put emphasis on the processing of internationalized identifiers. Testing data available in UASG004 (data file).
- 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

- 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 Domain Name

- 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.

Resolving Domain Name

- 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.

Validating Email Address

- 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?

Sending Email

- 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.

Email Regular Expressions (Regex)

- 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.

Sending Email

- 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.

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	Idna	100	IDN
Python3	Smtplib	84.3	Mail
Rust	Idna	87.1	IDN
Rust	Lettre	7.8	Mail
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	Idna	79	IDN
Go	Smtp	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

Summary

- 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.

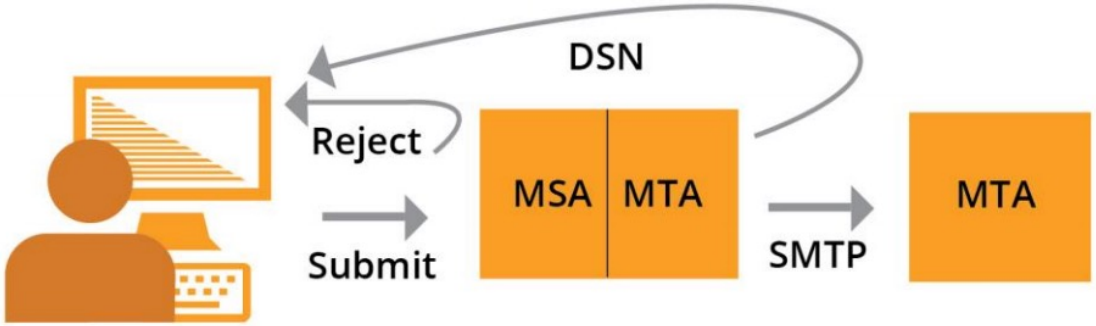
Email Address Internationalization (EAI)

Email Address Internationalization

- What is EAI
  - Having UTF8 support for:
    - mailbox name (before the @ sign)
    - domain name (after the @ sign)
- What is not EAI
  - Having UTF8 support in:
    - Subject line
    - Address comments
    - Message body
  - MIME provides all these in conventional mail
  - Use of any character set other than UTF-8

Levels of EAI Implementation

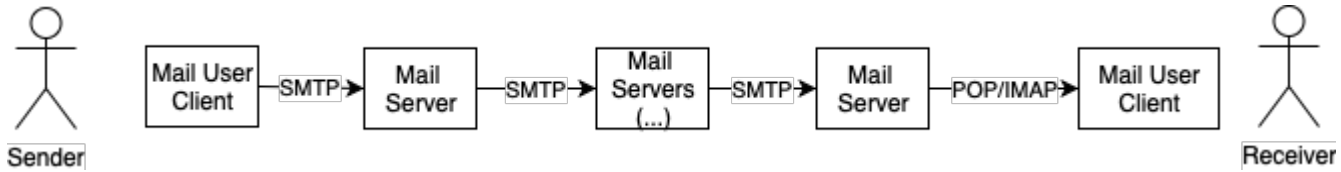
- No EAI support - only ASCII email addresses supported by the tools and services
- Level 1 - can exchange email with EAI addresses
  - Receive email from an EAI address
  - Send email to an EAI address
  - Cannot create mailbox and domain name in UTF8
- Level 2 - Level 1 + can create EAI addresses
  - Receive email from an EAI address
  - Send email to an EAI address
  - Create mailbox and domain name in UTF8



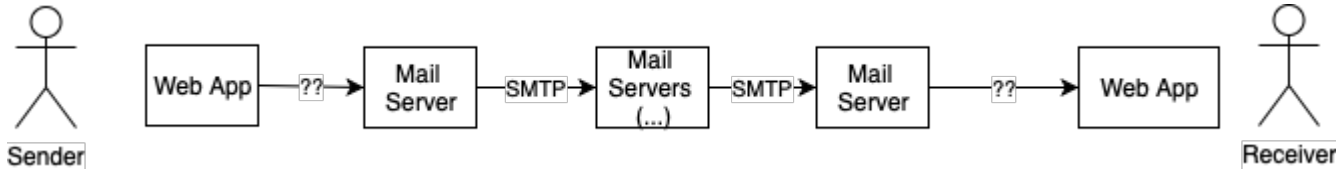
Email: How To Find the Destination Server

- When sending email to user@example.com, the method to find the destination email server is by querying the DNS for the MX records of the domain.
- For example, the MX records for example.com could be:
  - MX 10 server1.example.com
  - MX 10 server2.example.com
  - MX 20 server3.example.com
- The sender email server would then try connecting to either server1 or server2 since they have same priority (10). If none respond, it would then try server3 since it has a lower priority (20).
  - The higher number means lower priority.

Email Delivery Path



Using email software for both users.



Using web email for both users.

- Mix is also very common: Email software for one user, web email for other user.
- Mail server is the MTA; the source and destination servers are MSA and MDA, respectively.
- Mail User Client can be on desktop, laptop, or mobile.

Email Delivery Path Considerations

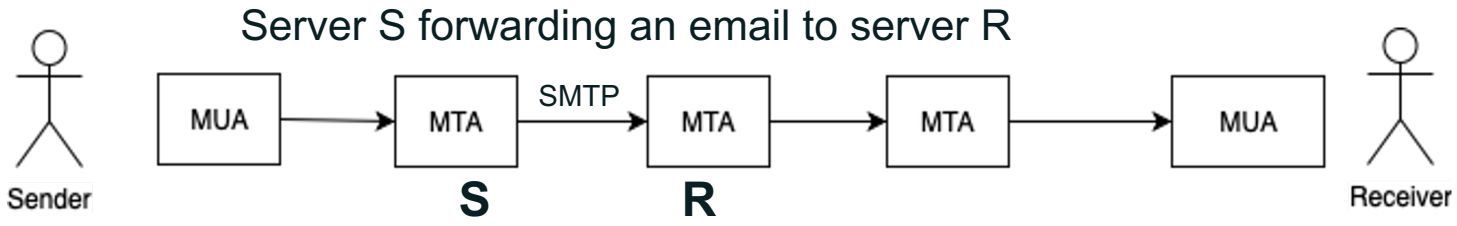
- Each user of an email communication chooses his own email environment/software/setup independently.
- The sender does not know the receiver email environment, meaning:
  - The sender does not know which protocols are used to deliver email.
  - The sender does not know if the receiver email supports some features.
- The delivery goes through a chain of email servers.
  - The number of email servers is unknown.
  - The actual chain of servers:
    - Is unknown at the beginning.
    - May change for any subsequent email sent.
  - The features supported by each email server is unknown to the path or from the sender.
  - Features are only discovered one hop at a time (i.e. the next hop).

Configuring for EAI

Email Protocol Changes for 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
  - Could “half support” EAI by providing a downgraded email version to the non-EAI conforming email software clients

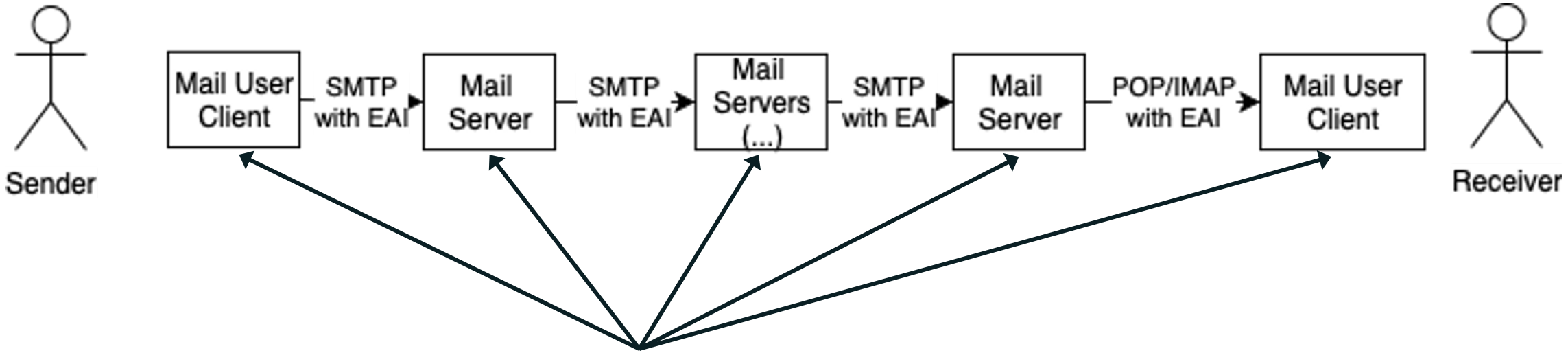
SMTPUTF8 Example



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
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

Specific SMTPUTF8 Signaling (EAI support)

Protocol Changes, Delivery Path Considerations



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

Protocol Changes, Delivery Path Considerations

- 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.

Additional Considerations

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

EAI Support by Email Tools and Services

Name	MUA	MSA	MTA	MDA	MSP	Webmail
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					
Apple iOS Mail 14.x	Most L2					
Mozilla Thunderbird	Few					
MS Outlook	Most L1					
MS Exchange Server (hosted)		All L1	All L1	Few		
Exim		Most L2	All L2			
Postfix		All L2	All L2			
Courier		All L2	All L2	All L2		
Gmail	All L1	All L1	All L1	Few		
XgenPlus		Most L2	Most L2	Most	All L2	Most L2
Sendmail 8.17 Alpha		Most L2	Most L2			
Halon		Most L2	Most L2			
Thunderbird 89 beta	Most L1					
Dovecot				None		

See detailed testing results in [UASG030A: EAI Software Test Results]

Considerations for Mailbox Names Using EAI

Considerations for Mailbox Names Using EAI

- UASG028 - Considerations for Naming Internationalized Email Mailboxes
- Supported Scripts
  - Know user expectations for writing systems for mailbox name and domain name portion
  - Understand complexities involved for additional scripts (e.g. security, confusion, etc.)
- Length of a Mailbox Name String
  - Know constraints of your system and user expectations
  - Consider same or a similar policy as for ASCII mailbox names
- Script Mixing
  - Allow limited script mixing only when clear user need based on local practice
  - Consider security and confusion due to script mixing for mailbox and domain name

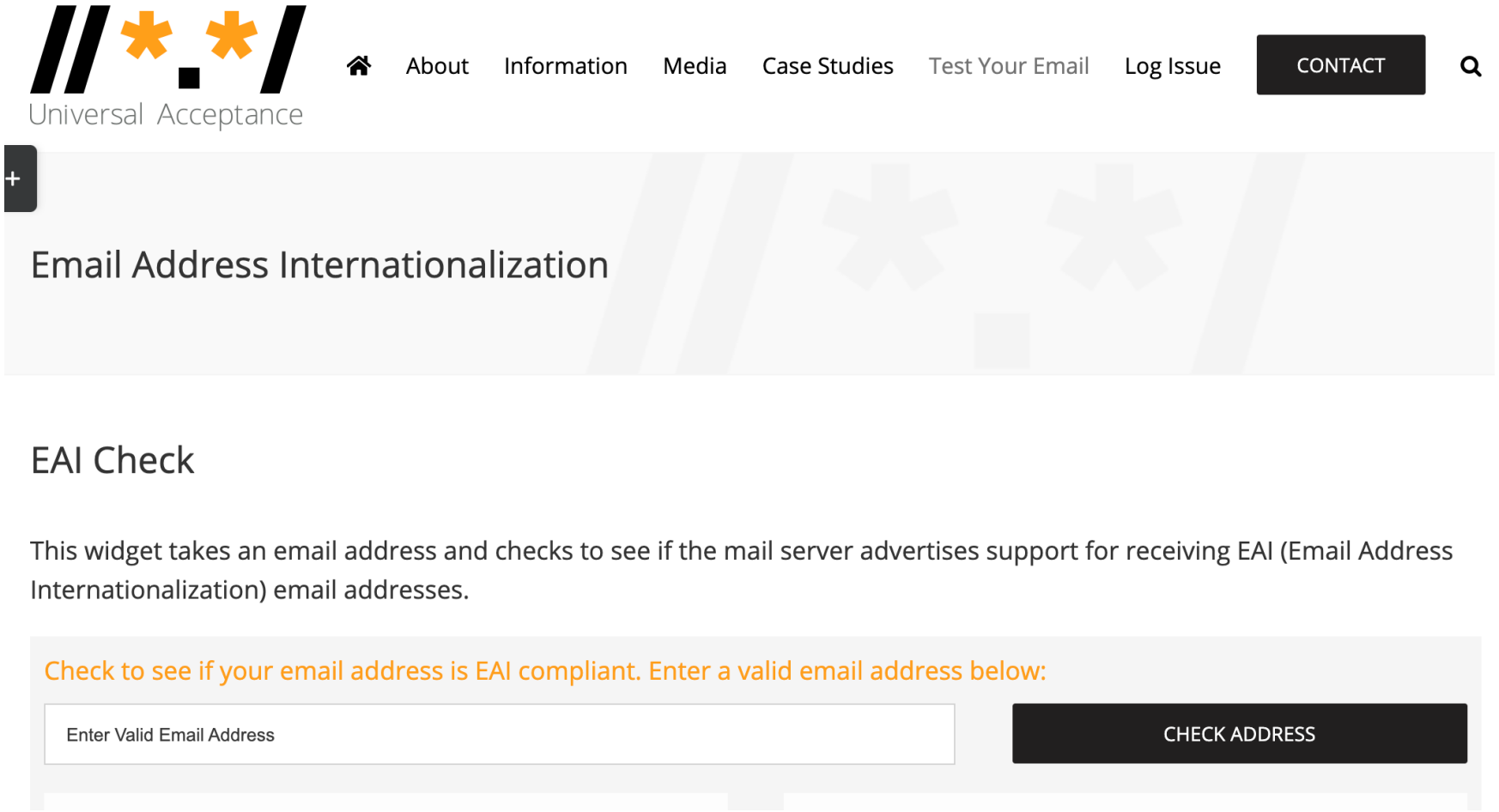
Considerations for Mailbox Names Using EAI

- Signs and Symbols
  - Avoid using signs and symbols, especially that do not exist on keyboard/input devices
  - If required for your market, the dot (.), underscore (_), hyphen (-) and plus sign (+) are commonly used
  - Review any additional signs (if needed) and ensure it does not cause a security issue
- Preventing Invalid and Unstably-Rendered Strings
  - Check if Reference IDN tables meet desired mailbox string and update as needed
  - Use a string validation tool (e.g. LGR Tool) to validate the mailbox strings
- Aliases and Display Names Consideration
  - Consider alias-creation option for the user interface during the mailbox name selection process. ASCII alias can be allowed with an EAI mailbox name
  - Optionally allow the user to add additional aliases at a later time

Are Your Software Applications UA Ready?

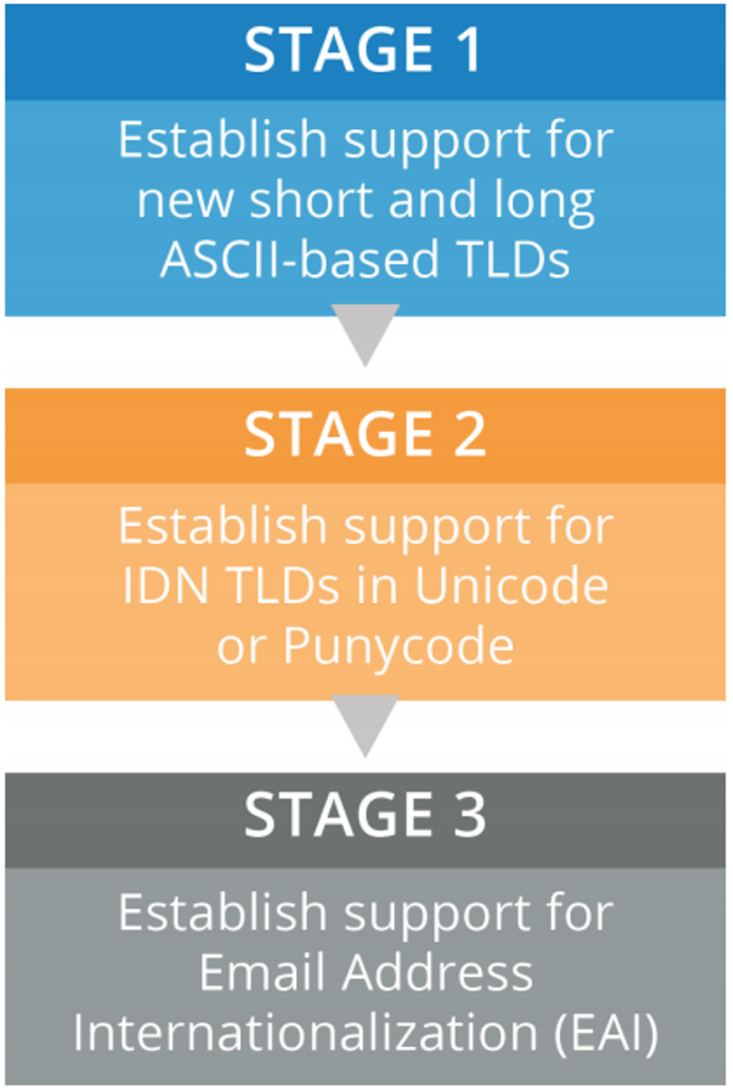
EAI Check

- Check if your email server supports Email Address Internationalization (EA):
  - https://uasg.tech/eai-check/



ICANN's Journey to UA Readiness - Model

- 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



Next Steps and Community Support

- 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.

Some Relevant Materials

- 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: UASG030A
  - 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!

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: UAProgram@icann.org

@icann

facebook.com/icannorg

youtube.com/icannnews

flickr.com/icann

linkedin/company/icann

slideshare/icannpresentations

soundcloud/icann