

# UA Technology WG and UA Email Address Internationalization WG Meeting

ICANN 67 / 9 March 2020



Universal Acceptance

# Overall Agenda

- \* UA Technology WG
  - \* Current progress
  - \* Overview of training on Programming in Java for UA Readiness
  - \* Overview of report on UA Readiness
  - \* Update on ICANN's progress on UA readiness
- \* UA EAI WG
  - \* Current progress
  - \* Best practice for email account name
  - \* Overview of training on EAI Implementation
- \* Q/A

# UA Technology WG

Satish Babu, Dessalegn Yehuala / ICANN67 / 9 March 2020



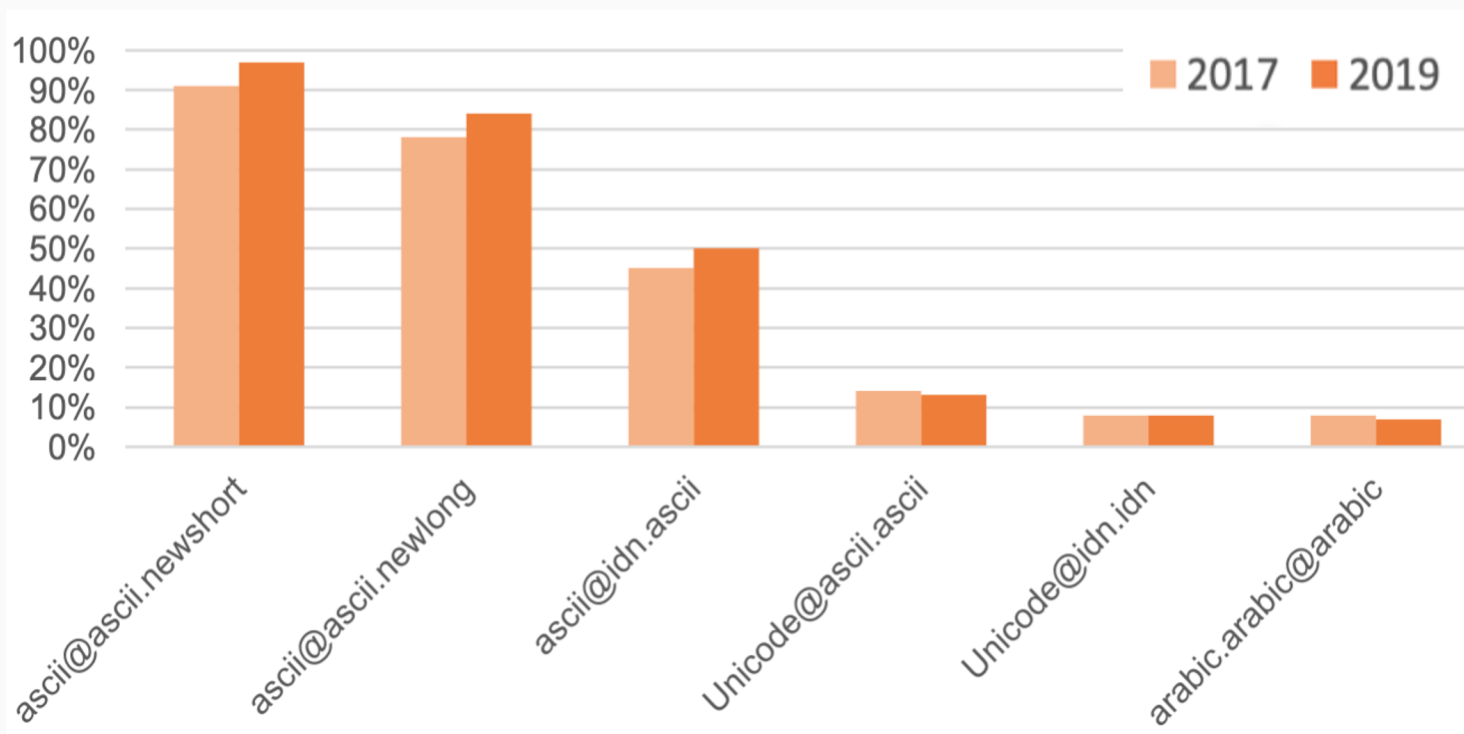
Universal Acceptance

# Agenda

- \* Summary of Charter: intended outcomes
- \* Overview of work
- \* Current progress
- \* Collaboration with UA working groups and the ICANN community
- \* Q/A



# Global Evaluation of Websites for Acceptance of E-mail Addresses



///532/

ascii@ascii.newshort test@test.exp	ascii@ascii.newlong test@test.example	ascii@idn.ascii test@普遍接受-测试.org
unicode@ascii.ascii 测试1@test.org	unicode@idn.idn 测试5@普遍接受-测试.世界	arabic.arabic@arabic (RTL) دون@رسيل.السعودية

# Technology WG: Intended Outcomes

- \* Develop the scope of UA readiness and its testing criteria.
- \* Evaluate and address technical standards and best practices.
- \* Evaluate and address the following for open source and proprietary applications on mobile and computer platforms:
  - \* **Application development environments:** Frameworks, programming language libraries, etc.
  - \* **Applications:** browsers, social media, content management systems, databases, legacy applications, etc.
- \* Develop technical documentation.
- \* Develop relevant technical training materials.

# Technology WG: Overview of Work

- \* Based on [FY20 Action Plan](#), the following activities are planned:

Define UA readiness in detail with other UA working groups.

Review and revise the test suite based on UA readiness definition, which can be used by developers.

Create a UA Readiness Maturity Model - a map for organization in terms of UA readiness.

Identify relevant standards and standards bodies; work towards updating these standards.

Review and remediate popular open source programming language libraries and utilities.

Develop technology training materials for UA readiness.

Develop communication materials to reach to UA technology WG stakeholders in collaboration with UA Comms WG.

# Technology WG: Current Progress

- \* UA Technology WG charter has been [published](#).
- \* [Statement of Work](#) to review the programming languages and their frameworks has been finalized.
  - \* Based on [earlier work](#) on Java, Rust and Python are summarized in [UASG018](#).
    - \* JavaScript, Angular, React, and Vue
    - \* Java and Kotlin on Android
    - \* Swift and Vapor
    - \* C#, VB and ASP on .NET core
    - \* Go and Gin
    - \* PHP
    - \* Mailers for the relevant platforms
- \* [Statement of Work](#) for the review of standards and best practices under development.



# Technology WG: Current Progress

- \* Training on Universal Acceptance for Java Developers being finalized after review.
  - \* Target audience:
    - \* Java developers, Software project managers, CTO
  - \* Objectives:
    - \* Understand base key concepts related to Internationalized Domain Names (IDNs) and email.
    - \* Understand issues with using plain Java for validating and using IDNs and email addresses.
    - \* Identify which libraries are appropriate for the use case.
    - \* Know how to use the libraries.
    - \* Develop best current practices for UA compliant applications.
  - \* Goal:
    - \* Develop UA-compliant Java applications.
  - \* ETA: Mid-April 2020

// [3.2]

# Collaboration with UA WGs and Community

- \* Jointly developing the scope of UA readiness with UA Measurement WG.
- \* Jointly developing communication for relevant stakeholders with UA Communications WG – provided input on the questionnaire for software developers.

# Q&A

Join UA Technology WG – sign up [here](#)

See details of UA Technology WG work at our [workspace](#)

For general information, email at [info@uasg.tech](mailto:info@uasg.tech)

More details about UASG at [www.uasg.tech](http://www.uasg.tech)

# UA Work Published Recently

- \* Recent studies completed and published along with translations:
  - \* Updated [Universal Acceptance Quick Guide](#) (UASG 005)
  - \* Updated [Introduction to Universal Acceptance](#) (UASG 007)
  - \* [UA Readiness of Command Line Networking Tools](#) (UASG 024)
  - \* [Global Evaluation of Websites for Acceptance of E-mail Addresses in 2019](#) (UASG 025)
  - \* [Email Address Internationalization: Evaluation of Major Email Software and Services](#) (UASG 021B)
  - \* [EAI Readiness in TLDs](#) (UASG 021D)

# UA Training for Java Developers

Marc Blanchet / ICANN 67 / 9 March 2020



# Current Status

- \* Goal:
  - \* Create training material
  - \* Targeted at Java developers
  - \* For proper handling of Universal Acceptance in their software
- \* 115 slides
- \* Code examples
- \* Common libraries are presented and discussed
- \* Best current practices

# Plan of the Tutorial

- \* Key fundamental concepts related to UA
  - \* Unicode
  - \* IDN
  - \* EAI
- \* Validating UA identifiers input
- \* Using UA identifiers
  - \* Resolving domain names
  - \* Sending email
- \* Best practices
  - \* Applies to Java, but most also apply to any programming language
- \* Conclusion
- \* References

# Libraries Discussed

- \* JRE-IDN
- \* Apache Commons Validator
- \* ICU
- \* Guava
- \* Java URL/URI
- \* HTTP Libraries
- \* JavaMail, Simple Java Mail, Jakarta Mail, Apache Commons,
- \* Spring framework
- \* Usage of Regex



# UA Readiness Conformance Scoping

Marc Blanchet / ICANN 67 / 9 March 2020



# Current Status

- \* Goal:
  - \* Define a framework
  - \* To help define boundaries for verifying UA readiness of software
  - \* To be used in future work of UASG
- \* 11 pages

# Structure of the Document

- \* Defines what UA means
- \* Defines two types of application:
  - \* Web-based
  - \* Native (which includes Mobile)
- \* Identifies the various components of each type of application
  - \* For the purpose of UA readiness verification. It is not a comprehensive list of possible components
- \* Defines steps of processing UA identifiers:
  - \* Accept, Validate, Process, Store, Process, Display
- \* Identify gates for testing UA in applications
- \* List test cases for each gate

# Web-Based Application Components

- \* Web-based application components:
  - \* Web browser
    - \* The browser is another application, however, it is an integral part of a web application delivery. Note that sometimes, web browsers appear as embedded into a native application.
  - \* Front-end
    - \* The front-end is the component that creates and enables code to be executed into the browser: so it is responsible for creating HTML and allowing HTML, CSS and Javascript code to be executed in the browser.
  - \* Back-end
    - \* The back-end is the component running on a remote server which typically executes more complex tasks as well as maintaining state and accessing a database.

///

# Web-Based Application Components

- \* Web-based application components (cont.):
  - \* Database
    - \* The database is the component storing information on a remote server. A database may be managed by a typical SQL database engine, or maybe just a file.
  - \* Filesystem
    - \* The filesystem is the place where files are stored on the Operating System.
  - \* External Service
    - \* Sometimes an application uses an external service to complement its core functionalities. A typical example of an external service is an authentication service provided by either internal to an enterprise or a cloud service provider.

# Native Application Components

- \* Native application components:
  - \* User Interface (UI)
    - \* The visual interface that interacts with the user. This interface is responsible for collecting user input such as domain names or email addresses.
  - \* Internal
    - \* The code processing the user's input and providing the functions of the application
  - \* Database
    - \* The database is the component storing application information. A database may be managed by a typical SQL database engine, or maybe just a file. The database may be local or remote.

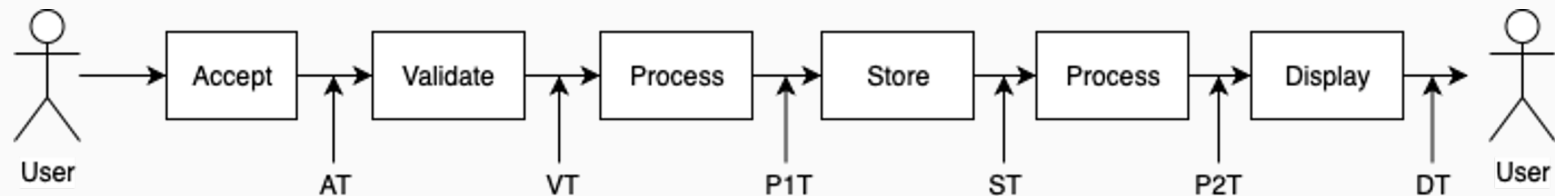
///\*\*/

# Native Application Components

- \* Native application components (cont.):
  - \* Filesystem
    - \* The filesystem is the place where files are stored on the Operating System
  - \* External Service
    - \* Sometimes an application uses an external service to complement its core functionalities. A typical example of an external service is an authentication service provided by either internal to an enterprise or a cloud service provider

# Gating Approach

- \* Gates where testing should be done
  - \* AT: Accept test
  - \* VT: Validation test
  - \* P1T: Processing test 1 (after validation)
  - \* ST: Store test (i.e. db and/or filesystem)
  - \* P2T: Processing test 2 (going to be displayed)
  - \* DT: Display test





# Q&A

# ICANN Universal Acceptance Readiness Update



Dean Edwards  
Sr. Product Manager

ICANN67  
9 March 2020

# Agenda

- Universal Acceptance Strategic Focus
- ICANN org Systems: UA Readiness Update
- Email Address Internationalization Update

# Strategic Focus

- Universal Acceptance is a strategic focus for ICANN org
- Ensure ICANN org's custom systems are Universal Acceptance ready
- ICANN org IT team includes Universal Acceptance readiness as part of our design guidelines
- ICANN org procurement process includes Universal Acceptance readiness provision
- Working with vendors of purchased solutions to confirm Universal Acceptance readiness, and educate where needed

# ICANN Systems Update

We have refined our approach to Universal Acceptance for ICANN org systems, making the UA readiness phases more measurable against specific targets, which include:

- Phase 1

  - Both short and long ASCII domain names

- Phase 2

  - Unicode / International Domain Names (IDNs)

  - Punycode (A-label) domain names

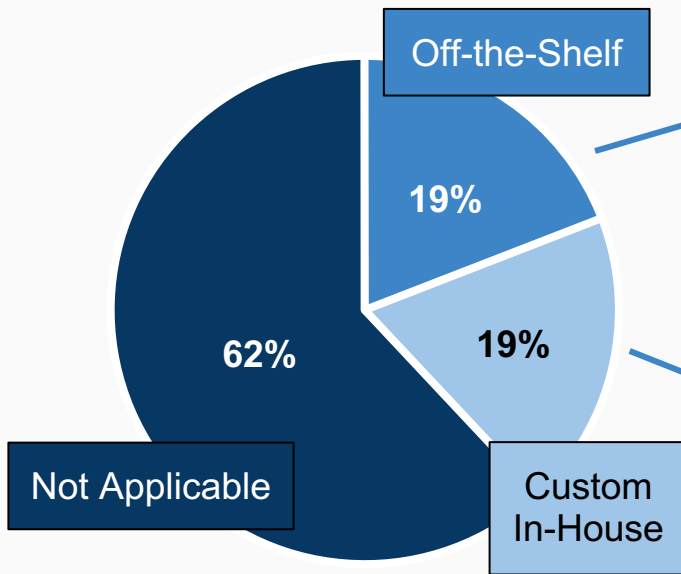
- Phase 3

  - Email Address Internationalization (EAI)

ICANN CIO/SVP Engineering meeting with major vendors

# ICANN Systems & Custom In-House Update

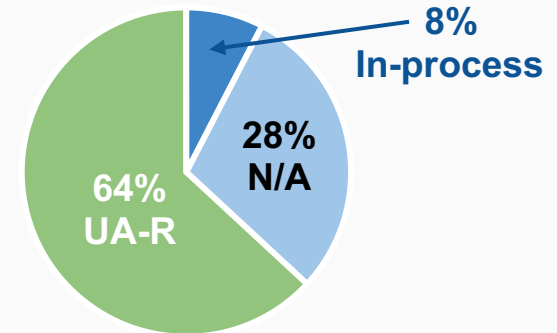
## ICANN Business-Directed Services



## Custom In-House Phase 1

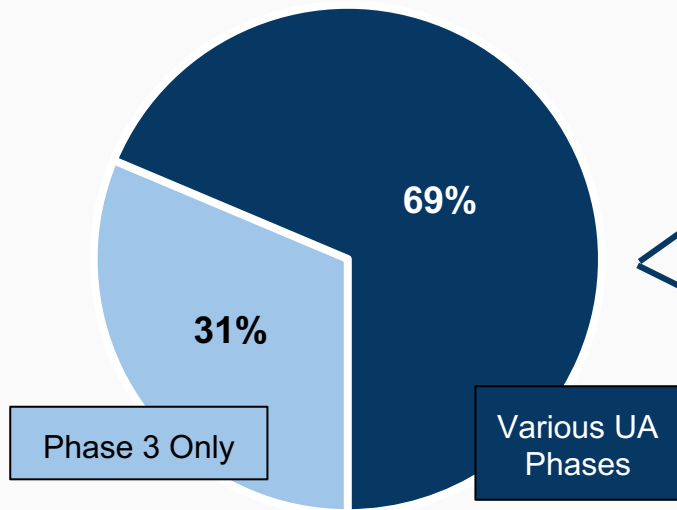


## Custom In-House Phase 2

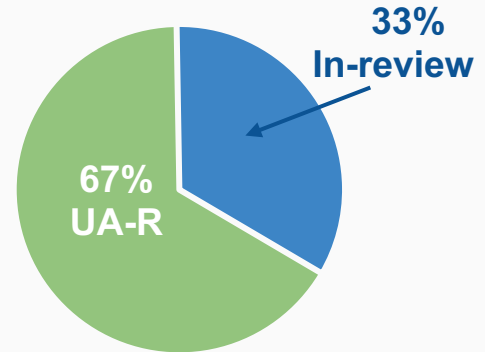


# ICANN Off-the-Shelf Systems Update

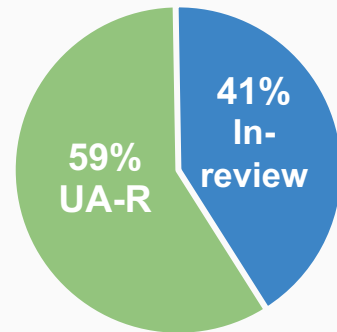
## UA Consideration for Off-the-Shelf Services



## Off-the-Shelf Phase 1



## Off-the-Shelf Phase 2



74%  
UA-R

# Email Address Internationalization

- ICANN org email system's EAI readiness is progressing

Outlook email client is compliant

Microsoft Exchange server support planned for Q4 FY20

Mail routers/gateway support planned for Q4 FY20

Security/spam filter and archive/backup - vendor Engineer actively working

ICANN org full EAI internal testing expected to begin in July 2020

- Salesforce platform expected to be compliant in Summer '20 release



# ICANN org Case Study

- **Updated ICANN org case study to be published shortly**



Creating Commonality to Increase the Inclusiveness and Diversity of the Internet

## Situation Overview

ICANN manages the highest level of the domain name system (DNS), called the Root Zone. The Root Zone matches each unique top-level domain name (TLD) with its unique Internet address (IP address). Without a seamless matching process, the Internet could work differently depending on a user's location.

Today, the Internet has expanded to include domain names in non-Latin based languages, and extensions that can be longer than the traditional two or three letters. Due to the rapidly changing domain name landscape, many Internet-enabled applications, devices and systems do not recognize or appropriately process

---

Universal Acceptance  
Case Study:

**Internet Corporation  
for Assigned Names  
and Numbers (ICANN)**

---

# Engage with ICANN – Thank You and Questions



One World, One Internet

Visit us at [icann.org](https://icann.org)



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



[linkedin/company/icann](https://linkedin/company/icann)



[facebook.com/icannorg](https://facebook.com/icannorg)



[slideshare/icannpresentations](https://slideshare/icannpresentations)



[youtube.com/icannnews](https://youtube.com/icannnews)



[soundcloud/icann](https://soundcloud/icann)



[flickr.com/icann](https://flickr.com/icann)



[instagram.com/icannorg](https://instagram.com/icannorg)

# UA Email Address Internationalization WG

Mark Svancarek, T. Santhosh / ICANN 67 / 9 March 2020



Universal Acceptance

# Agenda

- \* Summary of UA-EAI WG progress
- \* Future plan
- \* Discussion on best practice for mailbox account name

# Summary of UA-EAI WG Progress

Outcome	Status
1. Review the levels/phases of EAI support already developed and refine these as needed	Ecosystem of EAI, definitions, inventory and contacts are included in a Statement of Work currently being finalized by UASG
2. Develop an inventory of proprietary and open source tools and applications for EAI along with their contacts	
3. Develop an inventory of email service providers along with their contacts	
4. Develop technical requirements for implementing EAI and baseline metrics for measuring EAI support	Reviewed <a href="#">UA Conformance Scoping</a> by UA-Meas-WG
5. Undertake review of email technology and services to determine the degree of their support for EAI, in conjunction with measurements working group	Statements of Work was <a href="#">announced</a>
6. Review, update and develop technical documentation and make its translations available	Reviewed <a href="#">EAI Deployment Tutorial</a> developed by ICANN org

# Summary of UA-EAI WG Progress (cont.)

Outcome	Status
7. Review, update and develop technical best practices and standards for the security and usability of EAI (e.g. dealing with homographs, script mixing, variant characters, right-to-left scripts, etc.)	On-going ( <b><i>To be discussed at ICANN67</i></b> )
8. Prioritize and outreach to these tool and service providers' contacts to support EAI and maintain the status of the engagement with them to encourage them to support EAI in their tools and services (e.g. encourage one service provider to support EAI per year)	To be started after inventory is ready.
9. Develop training materials for EAI deployment for email administrators, email tools and application developers and web-developers (e.g. including advice for managing ASCII and non-ASCII usernames in one mailbox)	On-going ( <b><i>To be discussed at ICANN67</i></b> )

// 3.3 //

# Summary of UA-EAI WG Progress (cont.)

Outcome	Status
10. Develop and undertake training dissemination to promote EAI support and its deployment, by organizing webinars, training courses on ICANN learn, face-to-face training and other effective mechanisms	To be started after materials are finalized
11. Develop communication materials (blog, case studies, etc.) to acknowledge the support of EAI by various proprietary and open source email tools and services, in collaboration with the communications working group	Responded to <a href="#">Comms WG questionnaire</a> for developing messages.
12. Recognize email software and service providers who are EAI compliant	To be stated after measurement baseline is ready

///5.3/

# UA-EAI WG: Future Plan

1

## Build the capacity

Continue to develop and disseminate the training material to promote EAI support and its deployment by organizing webinars, training courses on ICANN Learn, face-to-face training and other effective mechanisms

2

## Communicate the success

In collaboration with Communication WG, develop blog, case studies, etc. to acknowledge the support of EAI by various proprietary and open source email tools and services

3

## Recognize the alliances

Acknowledge email software and service providers who are EAI compliant



Discussion:  
Best Practice for Mailbox Account



# 1. Definitions

1. **Mailbox Account:** The portion of an email address identifying the mailbox location under a domain name. Use interchangeably with other terminology such as “mailbox name”, “account name”, “username”, “UTF-8 part”, “local-part”
2. **Script and relevant terms:** refer to the Unicode Standard
3. **label validity, variant label:** refer to Label Generation Ruleset (LGR) as per [LGR Procedure](#)

///5.3/

## 2. Length of Mailbox Account

- \* Consider applying the same policy on length of an internationalized mailbox name as for ASCII mailbox name.

# 3. Script Used

- \* Script mixing should be prohibited.
- \* The exceptions to this are permissible for languages that conventionally require the commingled use of multiple scripts, for example, Japanese.

# 4. Rendering Stability

- \* Avoid security issues potentially caused by invalid labels and their variants.
- \* If you intend to offer mailbox account in one language, refer to the [Second Level reference LGRs](#). For example, if you want to offer mailbox names in German, then use the German language LGR.
- \* If you intend to offer mailbox names in one script (may cover multiple languages), refer to [Root Zone Label Generation Rules](#). For example, if you want to offer mailbox names in German, French, Swedish, then use the Latin script LGR.

# 4. Rendering Stability (Note)

\* Note:

- a. Reference Second Level LGRs for RTL script disallow a digit to be at the beginning of a label as per RFC5893
- b. Referring to Root Zone Label Generation Rules can prevent security issues, but it can be restrictive
- c. Using digits is not defined in RZ-LGR
- d. Some script do not yet have RZ-LGR available, for such case, the high-level principles in RFC6912 should be followed

///5.2/

# 5. Variant Labels

- \* After a mailbox name has been assigned, variants of that mailbox names should be identified and blocked by default (When the first label in the variant set is registered, the rest of the variant labels are blocked.)
- \* For some languages with sound reason, e.g. Arabic, Chinese, some variants may need to be activated based on usability motivation. For example, a simplified Chinese and a traditional Chinese label. Both are used interchangeably.

//5.2/

## 5. Variant Labels (cont.)

- \* Additional variants can be assigned to the same mailbox as aliases, but these could get very numerous (hundreds or thousands) and become a management challenge. As such assigning multiple aliases to the same mailbox may need to be considered with caution and be selected based on usability mentioned.



## 5. Variant Labels (cont.)

- \* All case folding permutations should be considered as the same mailbox. (ref. [https://www.w3.org/International/wiki/Case\\_folding](https://www.w3.org/International/wiki/Case_folding))
- \* When there are more than one set of digits being used in the same repertoire (set of possible codepoints used to create mailbox names), those digits should be mapped as variant codepoints. For example, a Hindi mailbox administrator may consider abc123 and abc१२३ as variant mailbox names.

// 5.2 //

# 6. Other Considerations

- \* EAI standards suggest mailbox names be represented in Unicode format. The mailbox account name beginning with “xn--” potentially causes confusion with Punycode representation, a different method of representing a Unicode string; therefore, such strings should be avoided.

# 7. Pending Discussions

- \* How to manage special character e.g. Signs, Symbols?
- \* Should Normalized Form be included in this document?  
(ref: <https://unicode.org/reports/tr15/> )
- \* Is it a good idea to associate the UTF-8 mailbox name to an ASCII mailbox? What are the options for such association(s), what are the benefits or disadvantages of each?

# Q&A

# EAI Training for System Administrators

Marc Blanchet / ICANN 67 / 9 March 2020



# Current Status

- \* Goal:
  - \* Create training material
  - \* Targeted at System Administrators
  - \* For enabling EAI on their systems
- \* 77 slides
- \* Configuration examples
- \* Fully functional demo setup
- \* Deployment considerations

# Considerations

- \* Show configuration examples of open-source software that could be installed by the SysAdmins
- \* Tried to introduce key concepts and considerations to be relevant and useful even if the Sysadmins are not using the software shown in the examples
- \* Show details of mail headers that are changed for EAI
- \* Considerations related to configuration and deployment are explained in detail

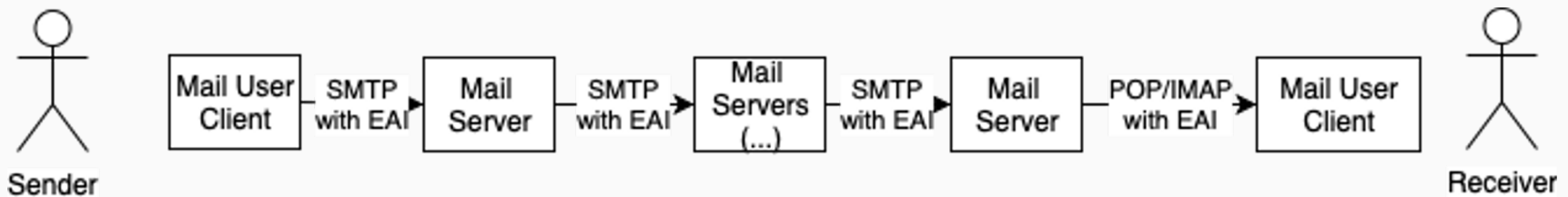
# Plan of the Tutorial

- \* Key fundamental concepts related to UA
  - \* Unicode
  - \* IDN
  - \* EAI
  - \* Email Systems
- \* EAI changes to protocols
- \* Deployment considerations
- \* Demo setup with configuration examples with Postfix, Courier, Mailmate and Gmail
- \* Conclusion
- \* References



# Key Considerations

- \* All servers in the mail delivery path must support EAI for the email to reach its destination



- \* Case folding in UTF8 EAI is not typically available
- \* If IDN is used in an email address, the servers must be configured to support both encoding (A-Labels and U-Labels)
- \* EAI may be considered as SPAM by SPAM filtering engines that were not updated to know that the local part maybe UTF8

# How Can You Be Involved

- \* Follow, share, like and engage with the UASG on social media, and use the UASG hashtag in relevant posts: #Internet4All
  - \* Twitter: [@UASGTech](https://twitter.com/UASGTech)
  - \* LinkedIn: <https://www.linkedin.com/company/uasgtech/>
  - \* Facebook: <https://www.facebook.com/uasgtech/>
- \* Sign up for UA working groups [here](#)
- \* Join the UA Discuss email alias: [ua-discuss@icann.org](mailto:ua-discuss@icann.org)
- \* Report a problem if you find an application or webpage that is not UA Ready: <https://uasg.tech/global-support-center/>
- \* For more information, visit <https://uasg.tech> or email [infor@uasg.tech](mailto:infor@uasg.tech)