L’Acceptation Universelle des Noms de Domaine et adresses mail: aperçu technique

Yaovi ATOHOUN, Yazid AKANHO et Malick ALASSANE

28-05-2021

Programme d’acceptation universelle

Vision

Tous les noms de domaine et adresses e-mail fonctionnent dans toutes les applications logicielles.

Impact

Promouvoir le choix des consommateurs, améliorer la concurrence et offrir un accès plus large aux utilisateurs finaux.

Catégories affectées et état de préparation de l'AU

- **Noms de domaine:**
  - **Nouveaux** noms de domaines de 1er niveau (TLD): example.sky
  - Noms de domaines de 1er niveau **Longs**: example.Africa
  - **Noms de Domaine Internationalisés** คน.ไทย

- **Adresses e-mail internationalisées (EAI):**
  - ASCII@ASCII (nouveaux et longs TLD) ekrem@misal.istanbul
  - ASCII@**IDN** marc@société.org
  - **Unicode**@ASCII 测试@example.com
  - **Unicode@IDN** пример@тестовая-зона.рф
  - **Unicode@IDN**; scripts de droite à gauche ای-میل@مثال.موقع

Quelques considérations pour la préparation à l'AU

L'état de préparation de l'AU doit être vérifié et corrigé (au besoin) à plusieurs niveaux (technique, opérationnel, …), utilitaires, outils et applications.

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

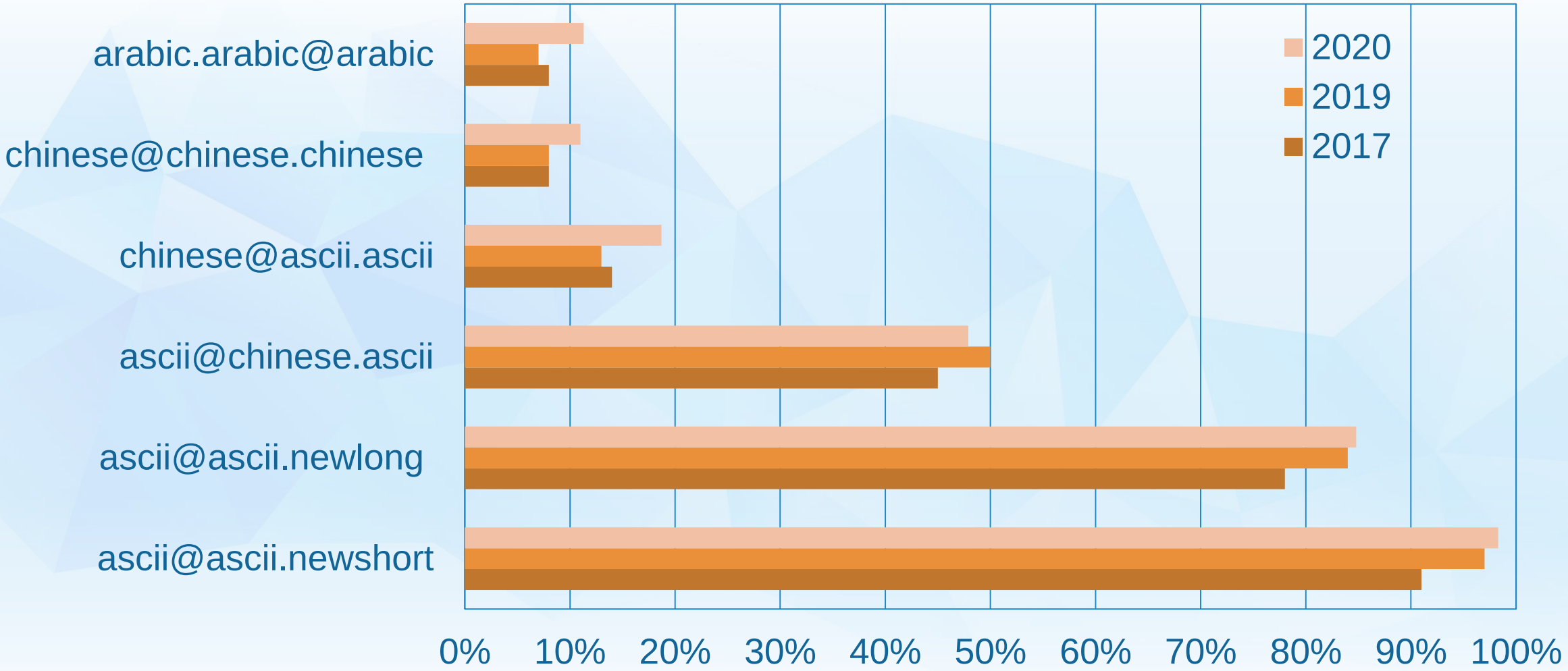
Platforms, Operating Systems and Sytem Tools

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

Standards and Best Practices

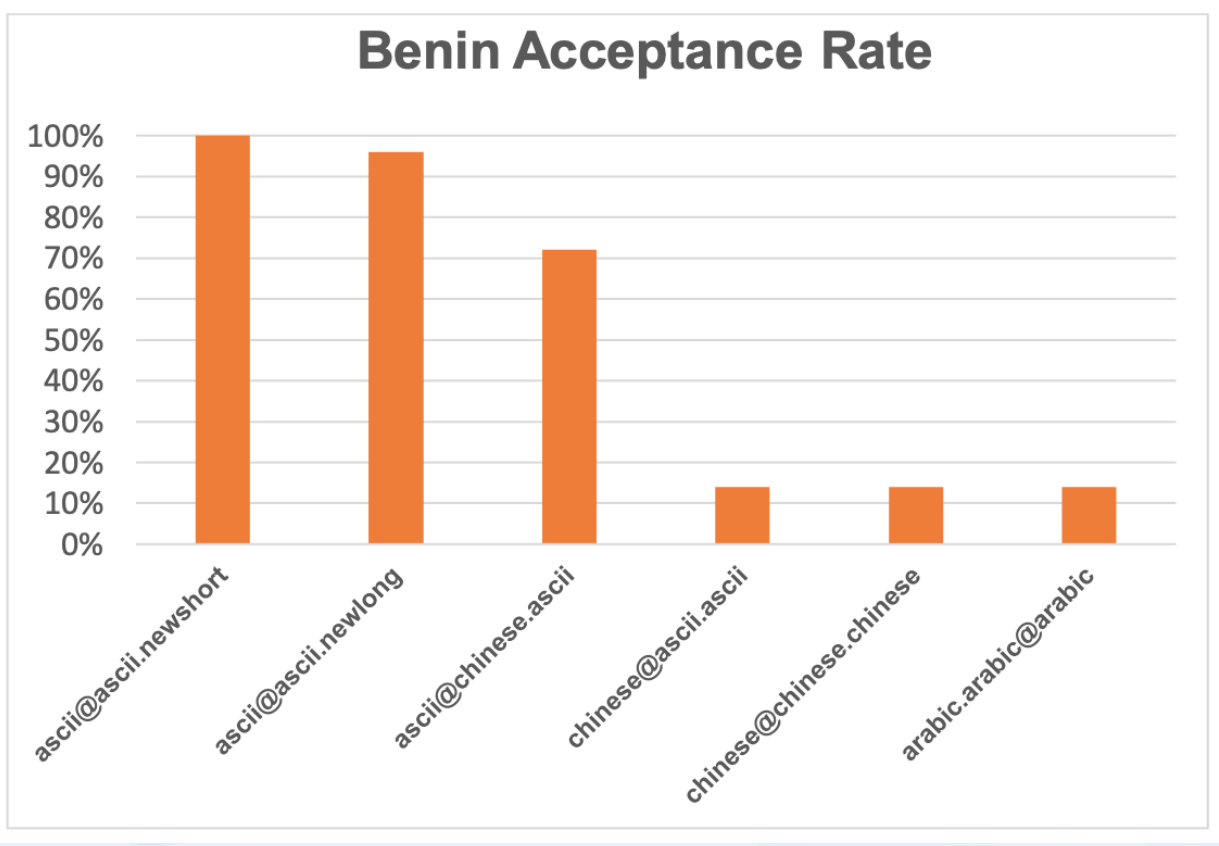
- IETF RFCs, W3C HTML, Unicode CLDR, WHATWG
- Industry-based standards (health, aviation, ...)

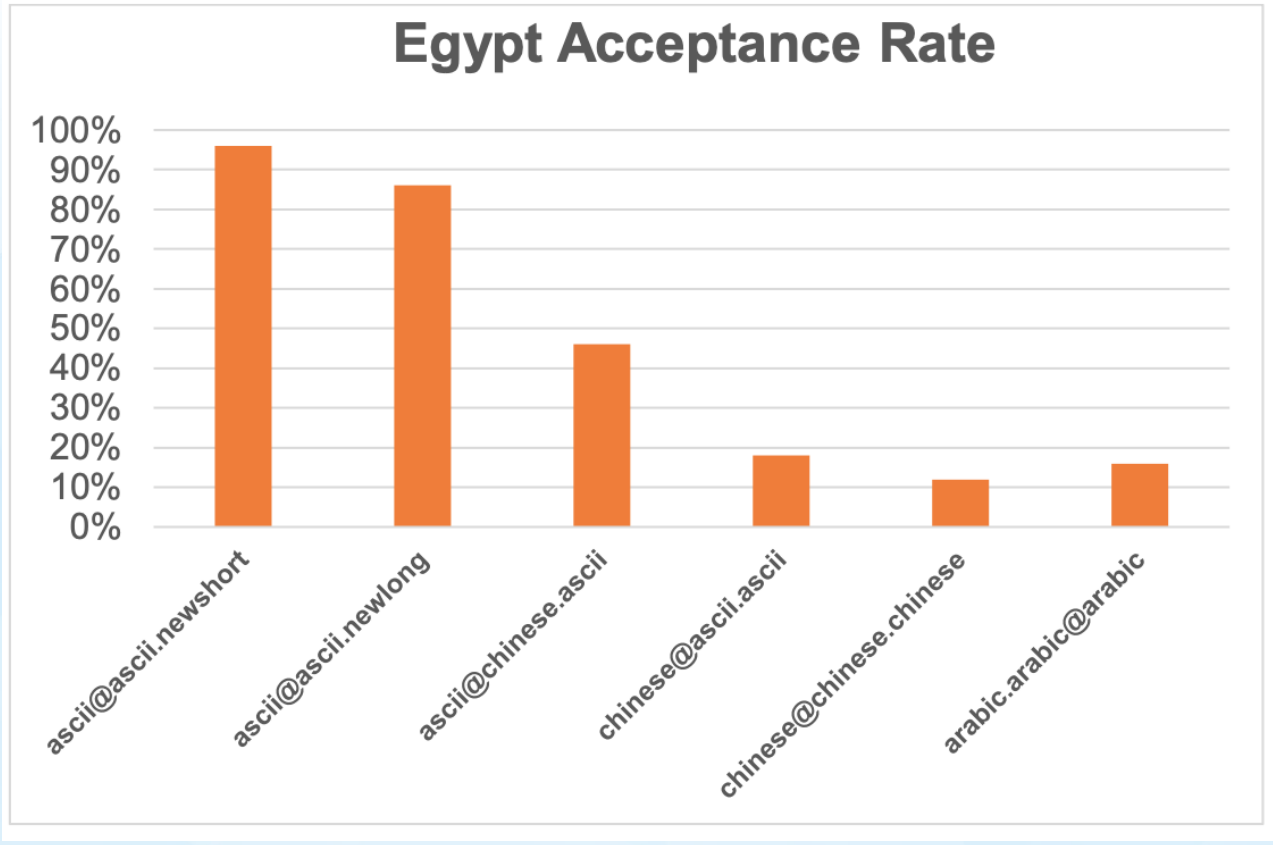
Acceptation d’adresses e-mail dans 1000 domaines



Évaluation des sites Web pour l’acceptation des adresses e-mail en 2019 - UASG025

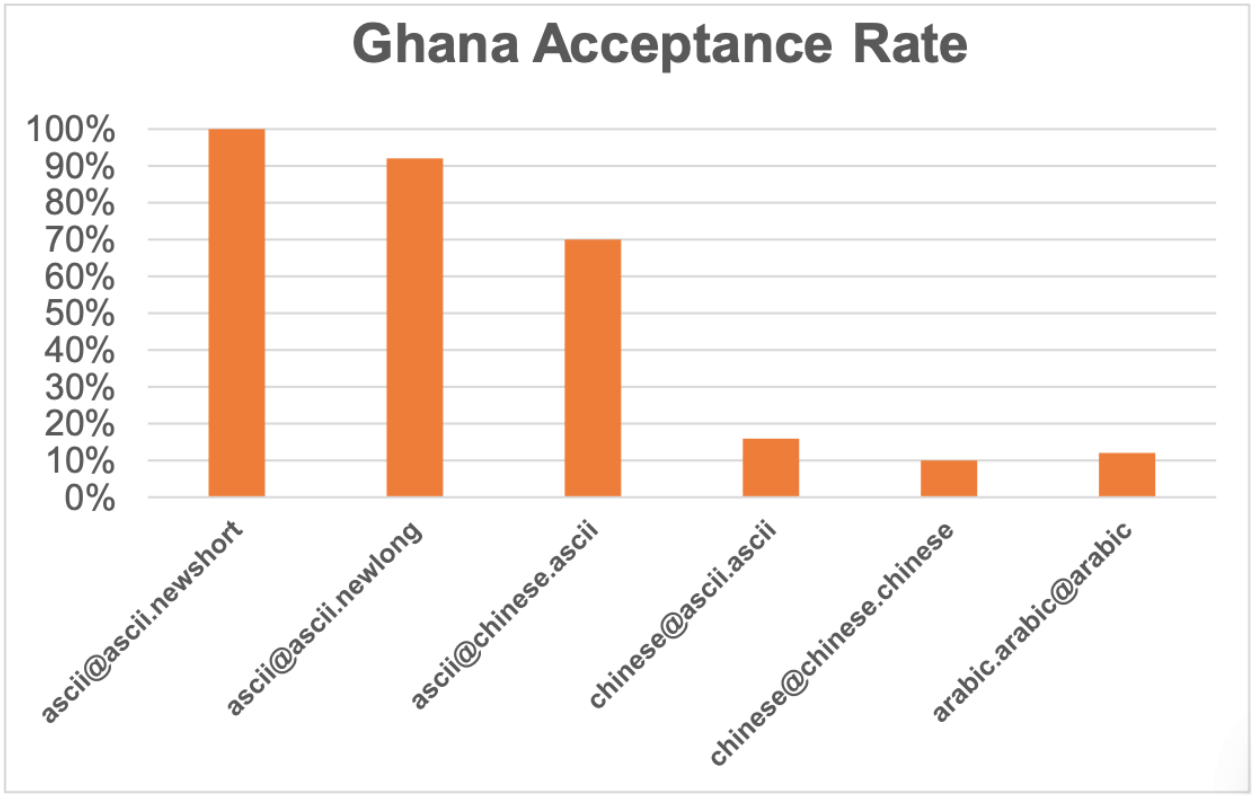
Acceptation d’adresses e-mail pour des sites Web africains en 2020

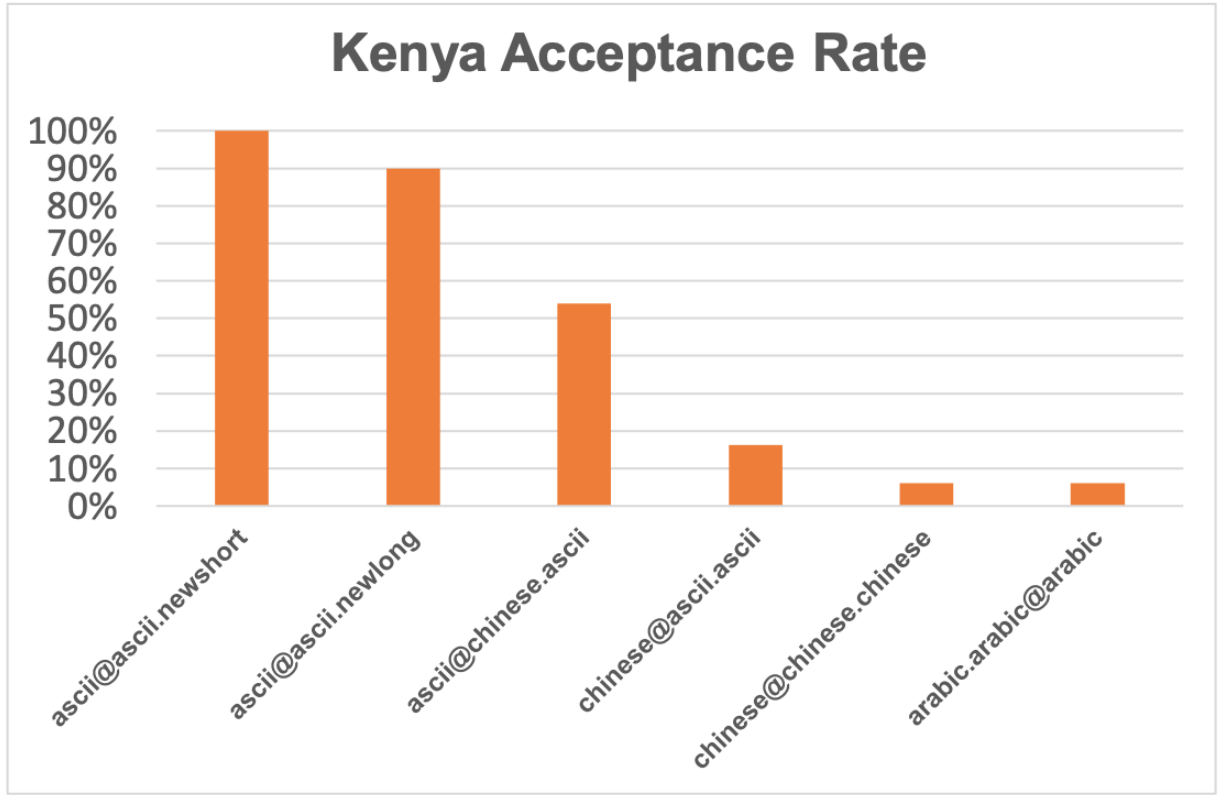




UASG027

Acceptation d’adresses e-mail pour des sites Web africains en 2020

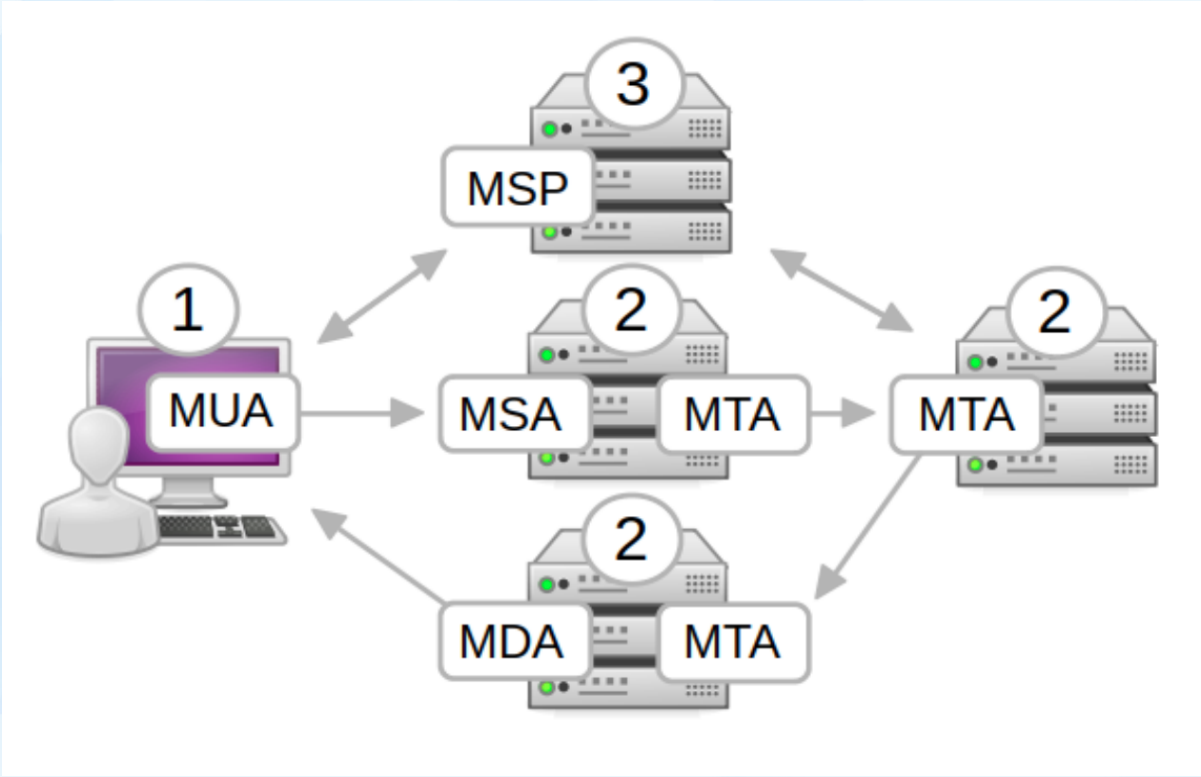




UASG027

Composants et services de courrier électronique pour l'AU

UASG021A et UASG021B: Évaluation des outils et services de messagerie électronique



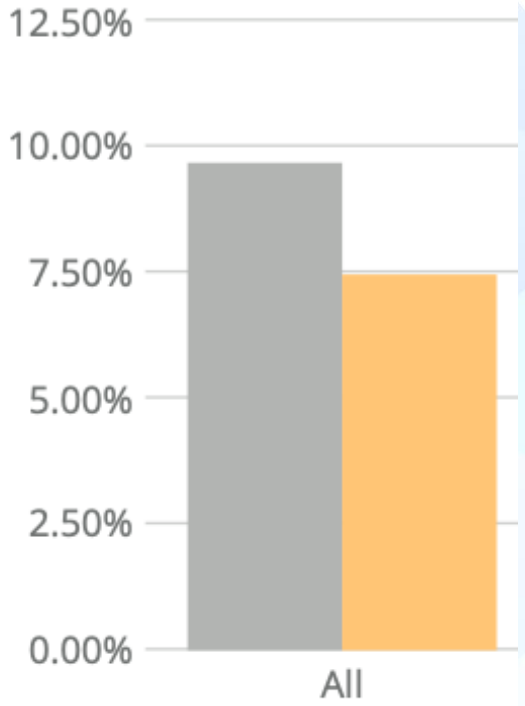
Webmail:
- Gmail
- Coremail
- Yandex
- …

Applications:
- Courier
- Dovecot
- Postfix
- Zimbra
- …

Autres:
- Filtres anti spams
- Calendriers
- …

Composants du système de messagerie électronique: Utilisateur de courier électronique, agent/interface de soumission, Agent de transfert, Agents de livraison, fournisseurs de service

Serveurs de messagerie prennant en charge l'EAI en 2019



Testez la compatibilité EAI POUR votre serveur de messagerie: https://uasg.tech/eai-check/

Préparation des domaines de premier niveau à l'EAI – UASG021D

Aperçu technique

Agenda

- Public cible et objectifs
- Concepts fondamentaux : Unicode, IDNs, EAI, UA
- Internationalisation des adresses de courrier électronique (EAI)
- Développement d’applications pour la prise en charge de l’AU en Java
- Bonnes pratiques pour le développement d’applications compatibles à l’Acceptation Universelle.
- Synthèse
- Références

Public cible et objectifs

- Public cible:
  - Administrateurs système
  - Développeurs
  - Managers IT
- Objectifs:
  - Comprendre les concepts clés liés à l’acceptation universelle (AU).
  - Comprendre l’internationalisation des adresses de courrier électronique (EAI).
  - Comprendre la programmation Java pour l’AU.

Concepts fondamentaux : Unicode, IDNs, EAI, UA

Plan

- Concepts fondamentaux (généralités):
  - Unicode, UTF-8, Normalisation
  - Noms de domaine, étiquettes (labels), domaines de premier niveau (TLDs), zones
  - Noms de domaine internationalisés (IDNs): Punycode, U-label, A-label.
  - Acceptation universelle (AU)
  - Agents de messagerie : MUA, MTA, etc.

Unicode

- Encodage des glyphes en points de code.
- Dans les spécifications, les points de code sont affichés en hex en notation U+XXXX.
- Les points de code sont généralement codés au format UTF-8 (Unicode Transformation Format, 8 bits).
  - Taille (nombre de bits) variable par point de code individuel.
  - L’ASCII est utilisé tel quel.
  - Gold est utilisé pour gérer les points de code Unicode sur le Web et dans les protocoles, etc.

Unicode

- Il existe plusieurs façons d’utiliser un glyphe:
  - “è” = U+00E8
  - “e`” = “è” = U+0065 U+0300
  - La normalisation est un processus permettant de fournir la même representation finale quel que soit le type d’utilisateur.
    - Par exemple, le processus de normalisation C (Normalization Form C - NFC) génèrera U+00E8 pour les deux entrées précédentes.

Noms de domaine internationalisés (IDN)

- Les noms de domaine internationalisés (IDN) permettent l'utilisation de caractères non ASCII pour toute étiquette d'un nom de domaine.
  - Toutes les étiquettes d'un nom de domaine ne peuvent pas être internationalisées.
- Exemple: exâmple.ca
- L'utilisateur utilise la version IDN, mais l'IDN est converti en ASCII pour la résolution DNS.
  - exâmple => exmple-xta => xn--exmple-xta
  - Le préfixe xn-- est ajouté pour identifier un IDN.

Noms de domaine internationalisés (IDN)

- Exemple de processus d'utilisation d'un IDN:
  - L'utilisateur entre dans un navigateur : http://exâmple.ca
  - Le navigateur applique la normalisation sur l'entrée de l'utilisateur.
  - Le navigateur convertit exâmple.ca dans une représentation compatible ASCII appelé Punycode [RFC3492], et ajoute « xn – juste avant.
    - exâmple.ca deviant donc : xn--exmple-xta.ca
  - Le navigateur interroge le DNS pour obtenir l'adresse IP de xn--exmple-xta.ca
  - Le navigateur établie ensuite une session HTTP avec le serveur dont il a reçu l'adresse IP.

Noms de domaine internationalisés (IDN)

- Le protocole de traitement des IDN est nommé IDN for Applications (IDNA).
  - IDNA2008 est le dernier actuellement utilisé.
  - IDNA2003 est une version plus ancienne et ne doit plus être utilisé.
- L’étiquette U est la représentation native Unicode d’un label IDN : exâmple
- L’étiquette A est la représentation Punycode d’une étiquette IDN : xn--exmple-xta

Public Suffix List (PSL)

- La Public Suffix List (PSL) est un moyen permettant aux développeurs web de savoir si des domaines donnés sont controlés par une même organisation ou pas. Cela peut être considéré comme si un nom de domaine autorise à d'autres de s'enregistrer en dessous?
  - https://publicsuffix.org/
- Certaines bibliothèques, frameworks et applications utilisent la PSL comme liste statique des TLD.
- Si elle est utilisée dans des applications:
  - Le développeur doit veiller à sa mise à jour (la PSL) dans son application.
  - un TLD peut ne pas être présent dans la PSL.
- Pour savoir si un nom est en fait un TLD, il est recommandé d'utiliser d'autres méthodes.
  - Consulter la liste à jour publiée par l'IANA: https://data.iana.org/TLD/tlds-alpha-by-domain.txt

Internationalisation des adresses de courrier électronique (EAI)

- Syntaxe de l’adresse électronique: partie-gauche@nom-de-domaine
- Le nom de domaine peut être internationalisé (IDN).
- La partie gauche (également connu sous le nom de compte) avec Unicode (UTF8) est EAI.
- Exemples:
  - kévin@example.org
  - すし@快手.游戏

Internationalisation des adresses de courrier électronique (EAI)

- Impacts:
  - Les en-têtes de courrier doivent être mis à jour pour prendre en charge l'EAI.
  - Les en-têtes (de messagerie) sont utilisés par le logiciel de messagerie pour obtenir plus d'informations sur la façon de livrer des e-mails.

- Comme tous les serveurs de messagerie ne supportent pas l'EAI, un protocole de négociation n'est utilisé pour envoyer EAI que lorsque le serveur cible le le supporte. Si non, il renvoie juste un message impossible de transmettre à l'expéditeur.

- L'option SMTPUTF8 est utilisée dans le protocole de transfert de courrier (SMTP: Simple Mail Transfer Protocol) pour informer qu'un serveur de messagerie donné supporte l'EAI.

Acceptation universelle (AU)

- L’AU est le concept permettant de gérer convenablement les identifiants internationalisés, ainsi que les nouveaux et les longs domaines de premier niveau.
  - Identifiants internationalisés: IDN et EAI

- L’AU, c’est également:
  - Les nouveaux et les longs domaines de premier niveau (new generic top-level domains – new gTLDs)
    - Au début, les TLD étaient de deux ou trois caractères (.ca, .com).
    - Depuis quelques années, les TLDs peuvent avoir plus de 3 caractères (.engineering, .google, .technology, .pizza).
  - Une étiquette de TLD peut avoir jusqu’à 63 octets.
  - Certaines applications continuent de vérifier que le TLD saisi par un utilisateur est sur 3 digits maximum.
  - D’autres utilisent regex pour autoriser un maximum de 6-7 caractères pour le TLD.

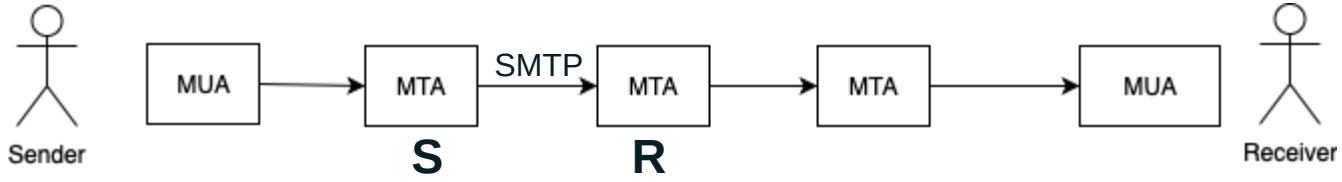
- L’AU, c’est aussi:
  - TLD ajoutés/supprimés:
    - Les TLD peuvent être ajoutés ou supprimés à tout moment.
    - Certaines applications vérifient l’existence d’un TLD sur la base d’une liste statique obsolète, ce qui est erroné.

Internationalisation des adresses de courrier électronique (EAI)

Modifications apportées par l’EAI

- SMTP
  - Est mis à jour pour prendre en charge l’EAI.
  - Dispose d’un drapeau (SMTPUTF8) pour notifier la compatibilité EAI.
  - Tous les serveurs SMTP sur le chemin doivent prendre en charge EAI pour livrer avec succès l’e-mail.
- POP/IMAP
  - Est mis à jour pour prendre en charge l’EAI.
  - Dispose d’un drapeau pour notifier la compatibilité EAI.

Exemple SMTPUTF8



Server S transférant un email au server R

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

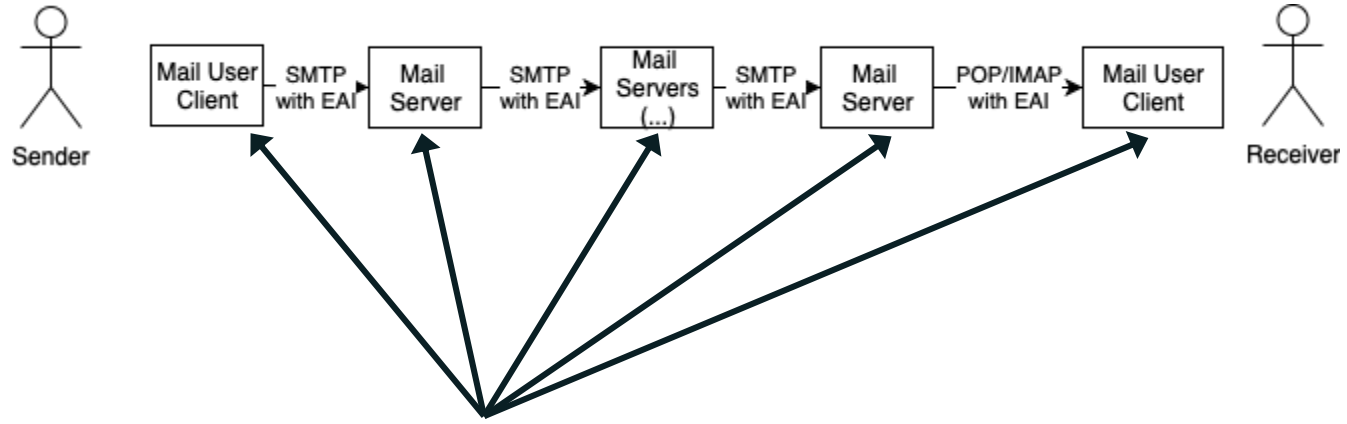
Notification SMTPUTF8 (support EAI)

Exemple SMTPUTF8

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

Le mail

Changements dans les protocols et considérations pour l’acheminement



Pour envoyer et recevoir un e-mail avec EAI:

- Toutes les parties impliquées dans le chemin de livraison doivent supporter EAI.
- Si un seul serveur SMTP dans le chemin ne prend pas en charge EAI, alors l’e-mail n’est pas livré.

Changements dans les protocols et considérations pour l’acheminement

- Que se passe-t-il lorsqu’un serveur e-mail (SMTP) dans le chemin ne supporte pas EAI?
  - Le dernier serveur essayant d’envoyer au suivant:
    - Renvoie à l’utilisateur expéditeur un rapport de non-livraison.
    - Jette le mail.
  - Rapport similaire à ceux qu’un expéditeur reçoit lorsqu’une adresse e-mail n’existe pas.

Considérations additionnelles

- Majuscule/minuscule:
  - En ASCII, les utilisateurs d'adresse mails ne font pas de différence dans l'utilisation des caractères majuscules et minuscule.
    - Exemple: un mail à PETER@example.com et peter@example.com sera livré à la même adresse.
  - Cette fonctionnalité n'est pas implémentée dans la plupart des applications compatibles EAI.

- SPAM:
  - Les e-mails EAI peuvent être considérés comme spam par certains logiciels de filtrage de spam, même lorsque les enregistrements SPF / DKIM appropriés sont disponibles.

- Applications et services:
  - Tous les logiciels et services serveur/client ne supporte pas encore EAI.

Support EAI

Composants du système de messagerie:

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

L1 - EAI Niveau 1 – envoie et reçoit des adresses EAI

L2 - EAI Niveau 2 - L1 et fournit également des adresses EAI locales.

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

Développer des applications compatibles à l’AU

Developement d’application compatible AU

- Eléments clés pour le support AU:
  - Validation de la saisie de l’utilisateur même lorsqu’il s’agit d’un identifiant internationalisé.
    - Par exemple, les adresses mails sont généralement utilisées comme source primaire d’authentification dans les applications.
  - Traitement des identifiants internationalisés dans l’ensemble de l’application (code).
  - Sauvegarde (base de données) des identifiants internationalisés.
  - Affichage des identifiants internationalisés aux utilisateurs.

- Voir les documents UASG007 et UASG026 pour plus de détails.

Validation de l’entrée de l’utilisateur

- Habituellement, les développeurs s’attendent à:
  - Des noms de domaine en ASCII.
  - Partie locale d’une adresse mail en ASCII également.
- La validation des entrées utilisateurs (nom de domaine ou adresse mail) a souvent été implémentée par de simples expressions régulières (regex).
- Difficile d’élaborer un regex efficient pour les IDN et EAI.
- Une bibliothèque IDNA2008 est le seul moyen actuel de valider les noms de domaine IDN.
- Donc:
  - N’utilisez pas de regex et mais plutôt une bibliothèque IDNA pour valider les noms de domaine.
  - Utilisez un regex simple (sans restriction pour la partie locale de l’adresse e-mail) et utilisez une bibliothèque IDNA pour valider le domaine.

Utilisation du DNS

- Habituellement, les librairies système pour les interrogations DNS tel que gethostbyname() s'attendant à des domaines en ASCII.

- Ces appels peuvent encore être utilisés si un pré-traitement est effectué:
  - Valider le nom de domaine.
  - Convertir les étiquettes U (U-label) en leur équivalent A (A-label).
  - Passer le résultat en entrée pour la fonction devant effectuer la requête DNS.

Librairies et Frameworks de haut niveau

- La plupart des OS, surtout mobiles, fournissent des frameworks assez fournis pour la gestion des noms de domaines et URLs.
  - Cependant:
    - Toutes les librairies et frameworks ne valident/traitent pas l'IDN ou l'EAI.
    - Du côté de Java, de nombreuses librairies et frameworks embarquent l'ancienne norme IDN : IDNA2003.

Quelques bonnes pratiques

- La validation de l'entrée EAI, IDN, UA est complexe.
- Ne vous fiez jamais à une liste statique de TLD pour effectuer des contrôles.
- Ne codez pas de syntaxe ou de longueur spécifique autre que ce qui est explicitement dans les normes. Par exemple, une étiquette de TLD peut avoir une taille jusque 63 octets en A-label ou format ASCII; ce qui inclue le préfixe 'xn--' pour l'IDN.
  - Effectuer un contrôle de longueur maximale de 6 ou 7 octets sur un TLD sur est donc une grosse erreur.
- Utilisez le type UTF8 pour traiter les noms de domaine et adresses mail.
- Faire la normalisation des entrées avant tout enregistrement, sauvegarde, comparaison et traitement des chaines de caractères de noms de domaines ou adresses mails.
- Au cours de l'enregistrement des identifiants en base de donnée, s'assurer que toute la chaine de traitement est effectivement compatible AU, y compris la base de données elle-même. Par exemple, la colonne recevant l'identifiant dans la table (base de données) doit être de type UTF8.

Quelques bonnes pratiques

- Adoptez une démarche consistant à faire des validations sommaires, ensuite effectuer des tests pour détecter les erreurs, plutôt que de faire plusieurs validations en un.
- Utiliser des librairies et frameworks compatibles AU (IDNA2008 pour ce qui concerne les noms de domaine).
- Faire des tests unitaires et système pour les identifiants AU.
- Adopter UASG004 comme tests préliminaires d’acceptance.
- Ne pas oublier la convertion des noms de domaine en A-label avant le passage aux différentes fonctions et ce, tout au long de l’application.

Validation et résolution de noms de domaine

- Convertir les U-labels en A-labels avant de continuer avec les méthodes standard.
- Par contre, faire recours aux U-labels pour l'affichage.
- La convertion entre A-labels et U-labels s'effectue de manière directe et sans perte. Il n'y a donc pas besoin de conserver les deux formats. Conservez les étiquettes A (A-labels) car elles sont davantage prises en charge dans le code et les dépendances.
- Toutefois, les étiquettes U sont utiles pour traitement (tri, comparaisons, recherches, …) car elles seront ordonées sur la base de leur valeur réelle.
  - c'est-à-dire la chaîne UTF8 au lieu de son encodage Punycode.
- Lorsque vous êtes sur le point d'afficher une chaîne, convertissez-là toujours en U-labels (format attendu par l'utilisateur).

Validation et résolution des addresses mails

- Normaliser les parties locales UTF-8 si une adresse mail est reçue en entrée.
- Utilisez toujours la version normalisée des parties locales lorsque vous comparez, triez ou recherchez.
- Pour la partie domaine, voir la section domaine.
- Effectuer la validation avec une librairie correcte avant de procéder à tout envoie.
- Si vous envoyez un e-mail à une adresse EAI, préparez-vous à une situation où:
  - L’e-mail pourrait être refusé par votre serveur de messagerie.
  - L’e-mail pourrait ne pas atteindre sa destination si l’un des serveurs de messagerie sur le chemin ne prend pas en charge EAI.

Quelques librairies et frameworks Java

- JRE-IDN (import java.net.IDN) implémente IDNA 2003.
- Apache Commons Validator (import org.apache.commons.validator) dispose d'une liste statique de TLD pour la validation; par conséquent, la bibliothèque est toujours obsolète dans une application.
- ICU (import com.ibm.icu.text.IDNA) implémente IDNA2008 comme un package sur la bibliothèque C, ce qui fait qu'il s'intègre bien à Java. C'est la meilleure bibliothèque pour l'instant.
- Guava (import com.google.common.net.InternetDomainName) utilise une copie de la liste statique PSL; par conséquent, la bibliothèque est toujours obsolète dans une application.
- Java URL/URI (import java.net.URL) ne valident pas les noms d'hôte. Il vaut mieux utiliser une autre bibliothèque pour la validation.

Compatibilité des langages de programmation

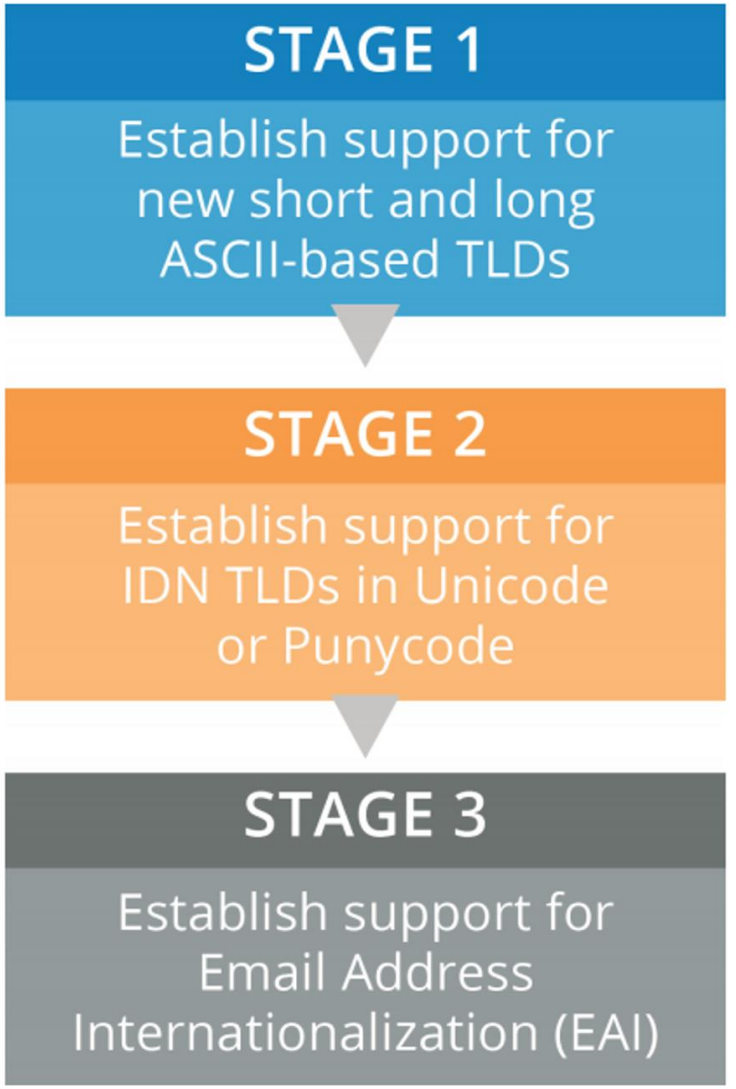
UASG018A

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

Vos applications logicielles sont-elles compatibles AU ?

Modèle d'ICANN pour le processus de préparation AU

- Etape 1: Mettre à jour les services pour prendre en charge les nouveaux TLD courts et longs ASCII.
- Etape 2: mettre à jour les services pour la prise en charge des noms de domaine (non-ASCII) internationalisés (IDN) en Unicode (U-label), et de la version ASCII des IDN, i.e la representation Punycode (A-label).
- Etape 3: Mettre à jour l'infrastructure et les services pour prendre en charge les adresses e-mail non ASCII.
  - Note: pour que l'infrastructure soit compatible, il faudrait que tous les composants supportent l'EAI.
- Plus de détails ici: ICANN's Case Study



Prochaines étapes et support à la communauté

- L'UASG et l'ICANN continuent de mener les analyses et identification de besoins, donner des formations et faire de la sensibilisation:
  - Analyse des besoins: médias sociaux, navigateurs, langages de programmation, outils EAI, etc..
  - Assainissement – engager des forums technologiques (p. ex. Github) et signaler les bogues.
  - Formation et sensibilisation – par le biais d'initiatives locales et des ambassadeurs.

Nous demandons à la communauté d'aider à répondre à la préparation de l'AU et de montrer l'exemple à travers:

1. **Sensibiliser les parties prenantes** sur les problèmes techniques.

2. **Mettre à niveau et utiliser des systèmes compatibles à l'UA** au sein de la communauté afin de susciter plus de demandes: mettre à jour les serveurs mail, utiliser les mails en langues locales.

3. **Faire un plaidoyer plus large pour la mise en conformité des systèmes** (services du gouvernement, secteur privé, etc.).

Collaboration avec l'Initiative locale de l'AU et les ambassadeurs de l'AU pour mener ces activités.

Parties prenantes utiles

- **Facilitateurs technologiques (Technology Enablers)** - Organisations produisant des normes et bonnes pratiques actuelles; fournisseurs de langages, librairies et frameworks de programmation.

- **Développeurs** - Organisations et particuliers développant et déployant des applications et services en ligne en utilisant les langages, librairies et frameworks de programmation.

- **Fournisseurs de logiciels et services mails**
  - **Fournisseurs de logiciels de messagerie** - Organisations et particuliers fournissant les applications, outils et services de courrier électronique.
  - **Fournisseurs de services de messagerie** - Organisations et particuliers fournissant des services de courrier électronique.

- **Administrateurs e-mail (et système)** - Organisations et particuliers déployant et administrant des logiciels et systèmes liés au courrier électronique.

- **Décideurs gouvernementaux** – demandent des produits et services compatibles AU en mettant à jour les normes à respecter et les processus d'acquisition.

L’AU à travers le monde

- **Ambassadeurs de l’AU**
  - Bénin
  - Chine
  - Egypte
  - Indie
  - Nigéria
  - Afrique du Sud
  - Turkie
- **Initiatives locales AU**
  - Chine
  - Inde
  - Thailande
  - Du Commonwealth of Independent States and Eastern Europe (CIS-EE):
    - Armenie, Biélorussie, Géorgie, Lettonie, Fédération de Russie, Serbie, Ukraine.

Quelques documents utiles

- Liste complète des rapports AU https://uasg.tech
  - 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

- Prière envoyer un mail à info@uasg.tech ou UAProgram@icann.org pour plus d'informations.

Merci