How to make your mail EAI compatible

ICANN 64 | Kobe | March 2019

Universal Acceptance

My new e-mail address

yés@nø.sp.am

A very short history of e-mail

In three acts

Internet mail, classic edition

From: Boris <boris@example.com>
To: Ines <ines@example.org>
Subject: Lunch cooperation

How about 1 PM at the cafe?

All text is ASCII

Internet mail, MIME edition

From: Борис <boris@example.com>
To: Iñes <ines@example.org>
Subject: Когда будет ланч?

How about 1 PM at the café?

Non-ASCII in most headers
Non-ASCII bodies

Internet mail, now with EAI

From: Борис <Борис@пример.com>
To: Iñes <iñes@example.org>
Subject: Когда будет ланч?

How about 1 PM at the café?

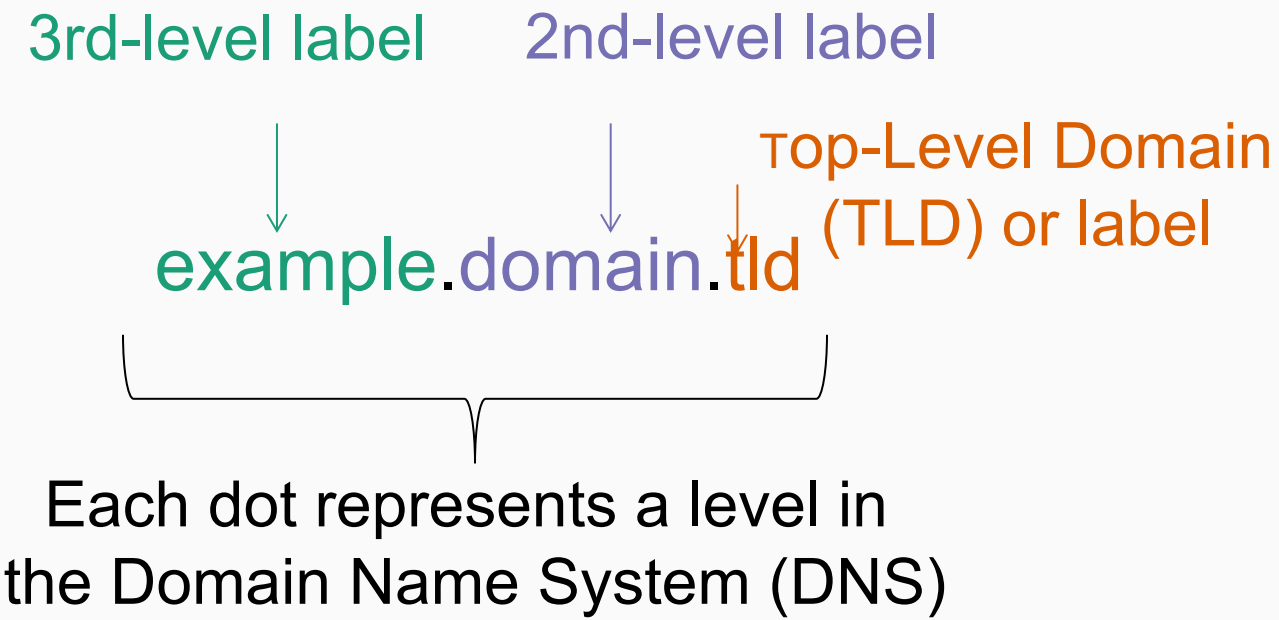
- UTF-8 everywhere
- In all visible headers and bodies

Goals for Today's Lecture

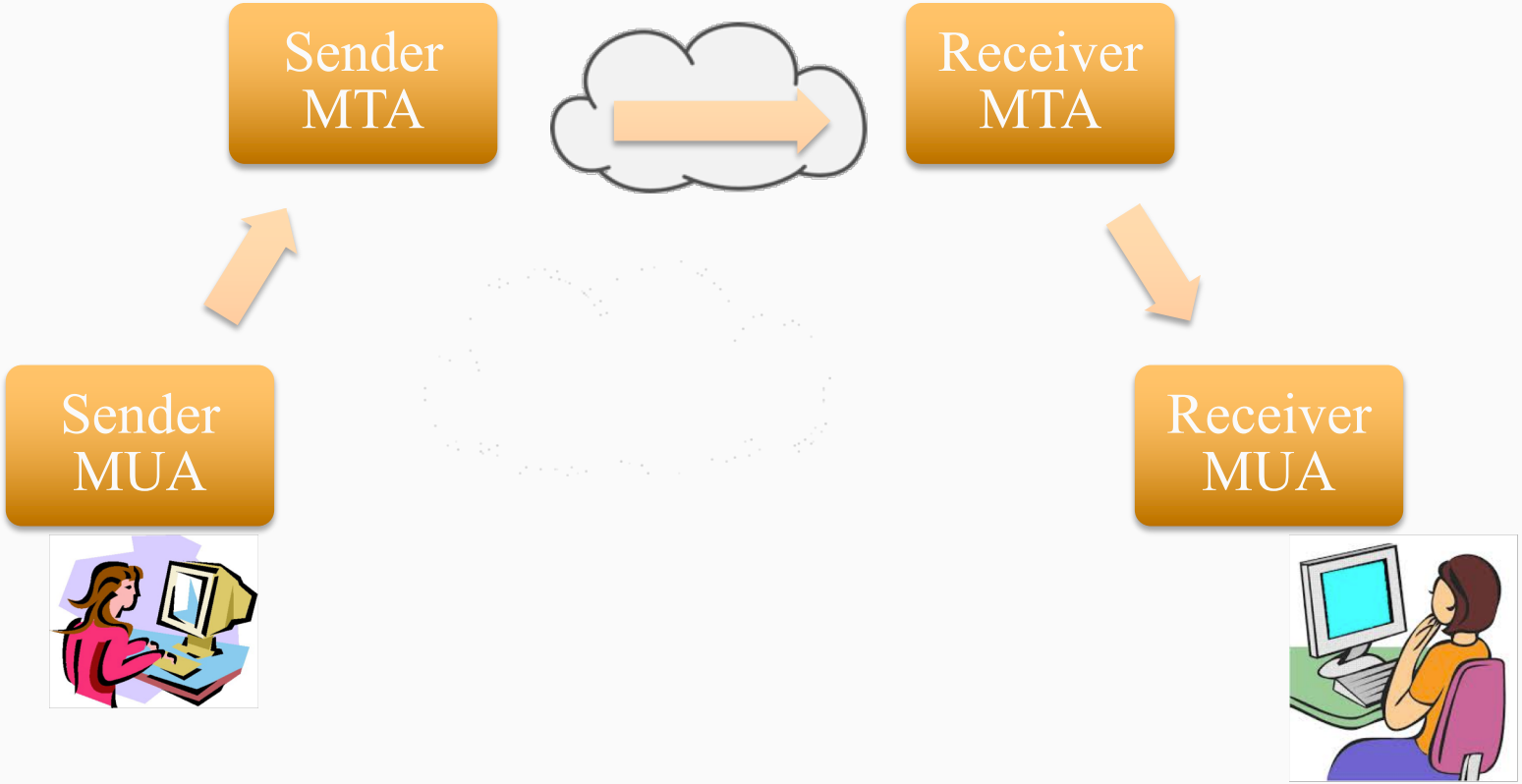
1. Understand the basics of Internet SMTP mail
2. Understand Unicode and Internationalized Domain Names (IDNs)
3. Understand what's needed for EAI mail

Building Blocks: **Domain Names**

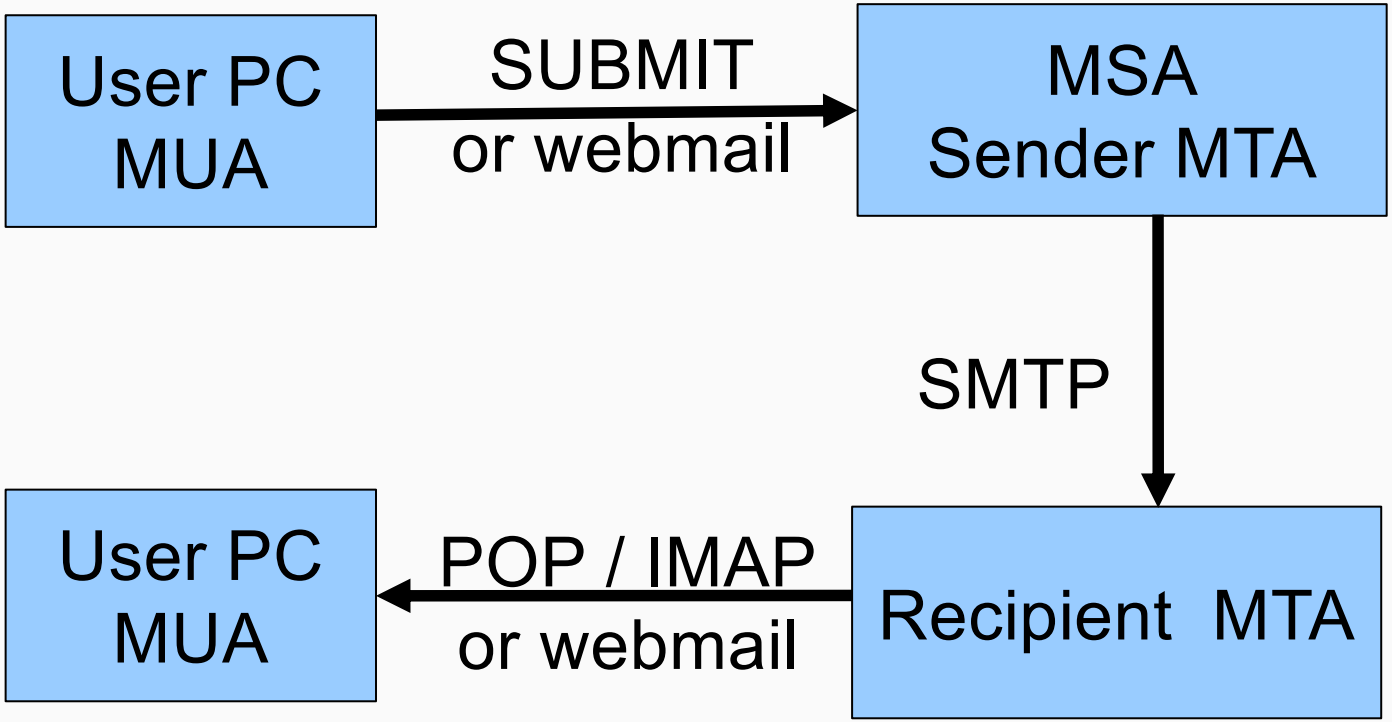
A domain name is dotted text strings used as a human-friendly technical identifier for computers on the Internet



Building blocks: **Internet Mail**



Building blocks: **SMTP**



Building blocks: **SMTP COMMANDS (1)**

```
R: 220 mail1.example.org ESMTP
S: EHLO mailout.example.com
R: 250-mail1.example.org
R: 250 8BITMIME
S: MAIL FROM:<boris@example.com>
R: 250 2.1.0 Sender ok.
S: RCPT TO:<ines@example.org>
R: 250 2.1.5 Recipient ok.
```

… to be continued ...

Building blocks: **SMTP COMMANDS (2)**

… continued from above …

```
S: DATA
R: 354 Send your message.
S: … message header and body …
S: .
R: 250 2.6.0 Accepted.
S: QUIT
R: 221 2.0.0 Good bye.
```

Building Blocks: **Character Sets and Scripts**

Languages are written using writing systems.

* Most writing systems use a single script, a set of graphic characters (glyphs).
* Some, e.g. Japanese use several scripts.

People can read scripts. But computers need numeric values that they can process. The mechanism for this is called an *encoding*.

Building Blocks: **ASCII and Unicode**

A character mapping associates characters with specific numbers. Many different mappings have been created over time for different purposes, two are now by far the most widely used: **ASCII** and **Unicode**.

ASCII: unaccented Latin letters, digits, punctuation

Unicode: everything else

Building Blocks: **ASCII and Unicode (cont.)**

ASCII

Domain names limited to the characters A-Z, the numbers 0-9, and hyphen “-“.

Unicode

Over 1 million characters, intended to represent every written language. Each Unicode character is assigned a number called a *code point*.

Unicode Code Points Examples

U+041A Cyrillic letter Ka	К
U+3069 Hiragana letter Do	ど
U+0636 Arabic letter Dad	ض
U+00E1 Small A with acute	á
U+0062 Small letter a	a
U+00B4 Acute accent	´

U+*xxxx* means the Unicode code point with hex value *xxxx*.

Building Blocks: **Unicode and UTF-8**

Unicode

Code points 0x0-0x7F are the same as ASCII. The highest code point is 0x10FFFF. Non-ASCII code points do not fit in a one 8-bit byte. UTF-32 stores each in a 32-bit word, convenient but bulky.

UTF-8

UTF-8 uses 1-4 bytes per Unicode code point. 0x0-0x7F are the same as ASCII.

Building Blocks – **Internationalized Domain Names and Email Addresses**

* Unicode enables domain names and email addresses to contain non-ASCII characters.
* Domain names with non-ASCII characters are *Internationalized Domain Names* (IDNs). An IDN can be all non-ASCII or a mix of ASCII and non-ASCII labels.
* Email addresses with non-ASCII characters are called Internationalized Email Addresses.

Building Blocks – **Internationalized Domain Names and Email Addresses**

* Non-ASCII labels use a new encoding in the DNS.
* Unicode labels are called U-labels. The ASCII-translated versions are A-labels, which start with `xn--`.
* For example, 普遍接受-测试.世界
  becomes `xn----f38am99bqvcd5liylcxsg.xn--rhqv96g`
* A-labels are not meaningful to human users, so display the U-label to them.

Email Address Internationalization: EAI

Email addresses contain two parts:

1. **Local part** (the part before the “@” character)
2. **Domain** (after the “@” character)

* Both parts may be Unicode.
* A Unicode domain is an IDN

Email Address Internationalization: EAI

Two levels of EAI support

* Level 1: handle other people's EAI addresses
  * ASCII addresses on your system correspond with EAI users
* Level 2: assign your own EAI addresses
  * EAI addresses correspond with EAI users and sometimes with ASCII users

Two levels of EAI support

* Level 1 is a lot easier
* Hard parts about Level 2:
  * Assigning good addresses
  * Matching addresses in incoming mail (later)
  * Kludges for ASCII compatibility

For MUA and MTA: Changes to SMTP

* New SMTP feature SMTPUTF8
* UTF-8 in addresses

```
R: 220 receive.net ESMTP
S: EHLO sender.org
R: 250-8BITMIME
R: 250 SMTPUTF8
S: MAIL FROM:<猫王@普遍接受-测试.世界> SMTPUTF8
R: 250 Sender accepted
```

Server Software (MTA - Mail Transport Agent)

* Servers advertise the SMTPUTF8 feature
* Clients check server for the SMTPUTF8 feature, use the SMTPUTF8 option when sending
* Don’t send EAI mail to servers that do not support it
  * Provide readable error reports when users try to do so
* Accept both U-label and A-label versions of domain names in e-mail addresses
* Do “fuzzy” matching in incoming addresses, variations such as upper/lower case or missing accents

POP & IMAP Servers

* Post Office Protocol (POP3) has UTF8 option to allow UTF-8 in usernames, passwords, and text strings.
* Internet Message Access Protocol (IMAP4) has UTF-8 option for UTF-8 in user names, passwords, folder names, and search strings.
* Both can optionally downgrade received messages for approximate versions for non-EAI clients (a poor second to upgrading MUAs to handle EAI)

POP & IMAP Servers

* Support is lagging
* At this point open source only Courier
* Gmail, Outlook provide IMAP for their users

Changes to Client Software (MUA)

* Handle Mailbox names in UTF-8
  * Also in address books, SUBMIT/POP/IMAP userid
  * UTF-8 passwords, too.
* Follow good practice for domain name validation
* Identify EAI messages when submitting to MSA/MTA
  * Be prepared for submission to fail with a non-EAI MSA
* Display headings and prompts in the user's language

Items for Email Service Providers to Consider

* Avoid addresses that can confuse users, offer Unicode mailbox names that conform to best practices
  * Unicode consortium and IETF provide guidance

* Avoid mailboxes with easily confused local parts
  * Don’t assign bob and bób and bøb

Items for Email Service Providers to Consider

* Do “fuzzy” matching on local parts of incoming mail
  * Allow variations such as upper/lower case, wrong accents, or variant characters
  * Handled locally in MTA, remote MTAs and users don’t do anything special
  * Fuzzy matching is not new, that’s why upper/lower case in addresses doesn’t matter

Items for Email Service Providers to Consider

* Offer ASCII mailbox aliases along with EAI mailbox names.
* Both names deliver to the same mailbox, so users can give addresses to both EAI and non-EAI correspondents.

Message downgrading

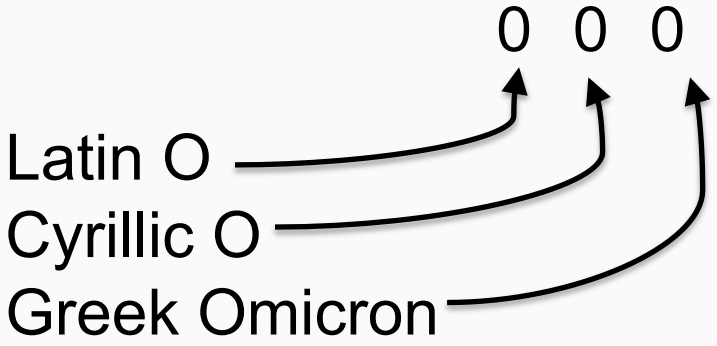
* You **can't** downgrade an EAI message to an ASCII message without losing information.
  * One cannot turn an EAI address into an ASCII address.
* In general, spend effort making software EAI-capable rather than trying to invent non-EAI workarounds.

Security challenges

- Homographs and near homographs
- Variants

Homographs

* They look the same but are not the same
* Also near-homographs like 1 l
* Forbid names in combined scripts



Variant characters

* Different appearance, same meaning
* Allow one in names, forbid the rest?
* Allow all, map to the same place?
* Something else?
* A decade long ICANN swamp

难以阅读的例子

難以閱讀的例子

Mail address challenges

- Longer, unexpected domain names

  `someone@home.sandvikcoromant`
- Several ways to write the same character
  - Is it á or ´+ a ?
- Punctuation possible in local parts
- Way too many emojis 😫 😩 🥑

Domain name challenges

- A-labels are usually unreadable

  `xn--onqrps50a3m1a8owtum7fb.xn--fiqs8s`

  or 难以阅读的例子.中国
- Tools to convert can help

Challenges during transition

EAI software can be tricky to debug fully. Some problems may only be apparent when using some scripts, e.g. LTR and RTL scripts.

- Ensuring reliable EAI mail
  - Send and receive test messages using different scripts
  - Exchange test messages with many *different* other EAI-capable mail systems

How to make your mail EAI compatible

ICANN 64| Kobe | March 2019

Universal Acceptance