Technical view on IDNA

Internationalized Domain Names in Applications

Patrik Fältström

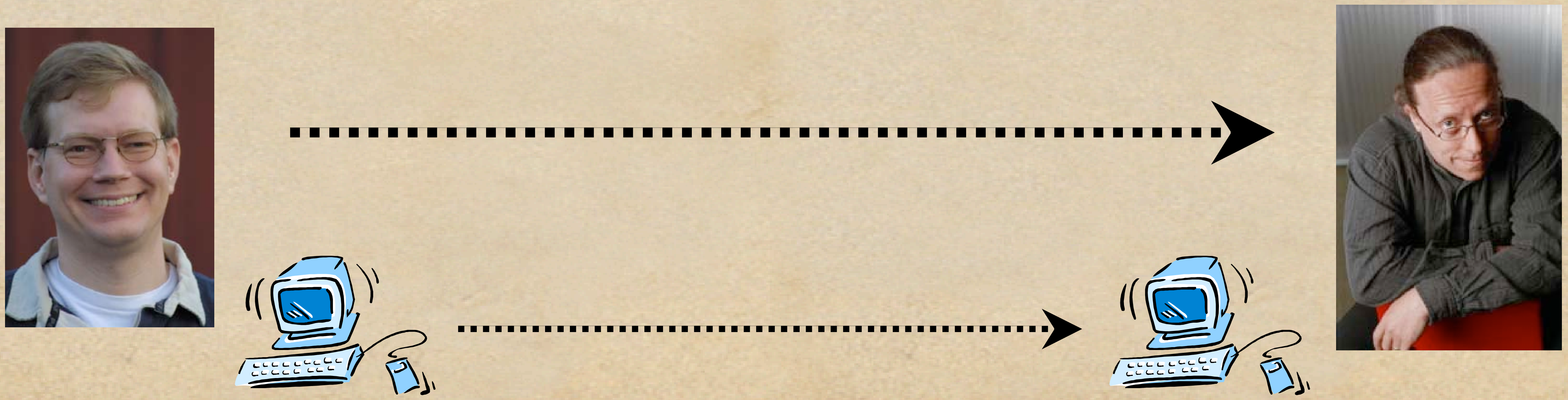
费思哲

Member ICANN Presidents Advisory Committee on IDN

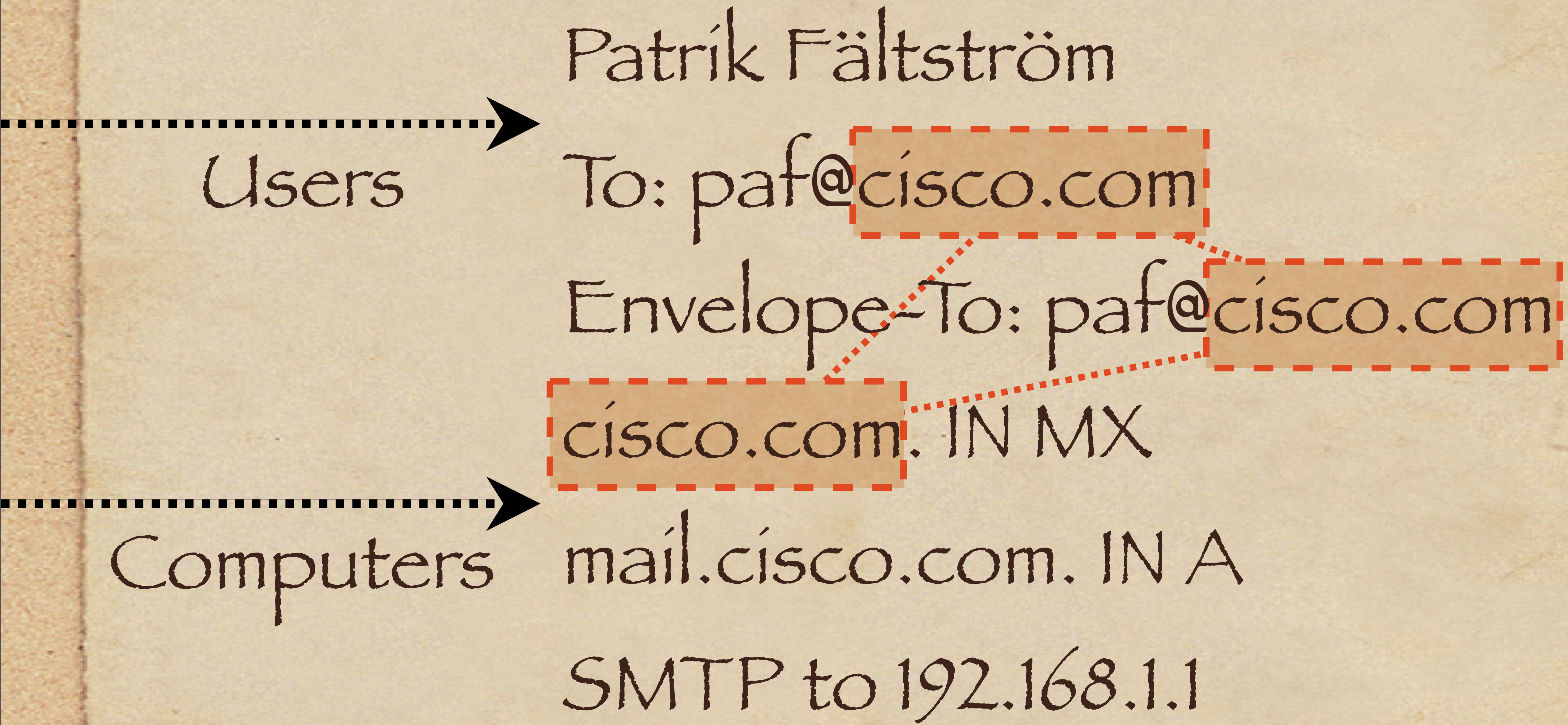
Senior Consulting Engineer, Cisco Systems

Communication

- When communicating, people have historically used “local” characters
- Communication was local, writing language was developed locally



Protocol stack



Patrik Fältström

Users

To: paf@cisco.com

Envelope-To: paf@cisco.com

cisco.com. IN MX

Computers

mail.cisco.com. IN A

SMTP to 192.168.1.1

Unicode in DNS?

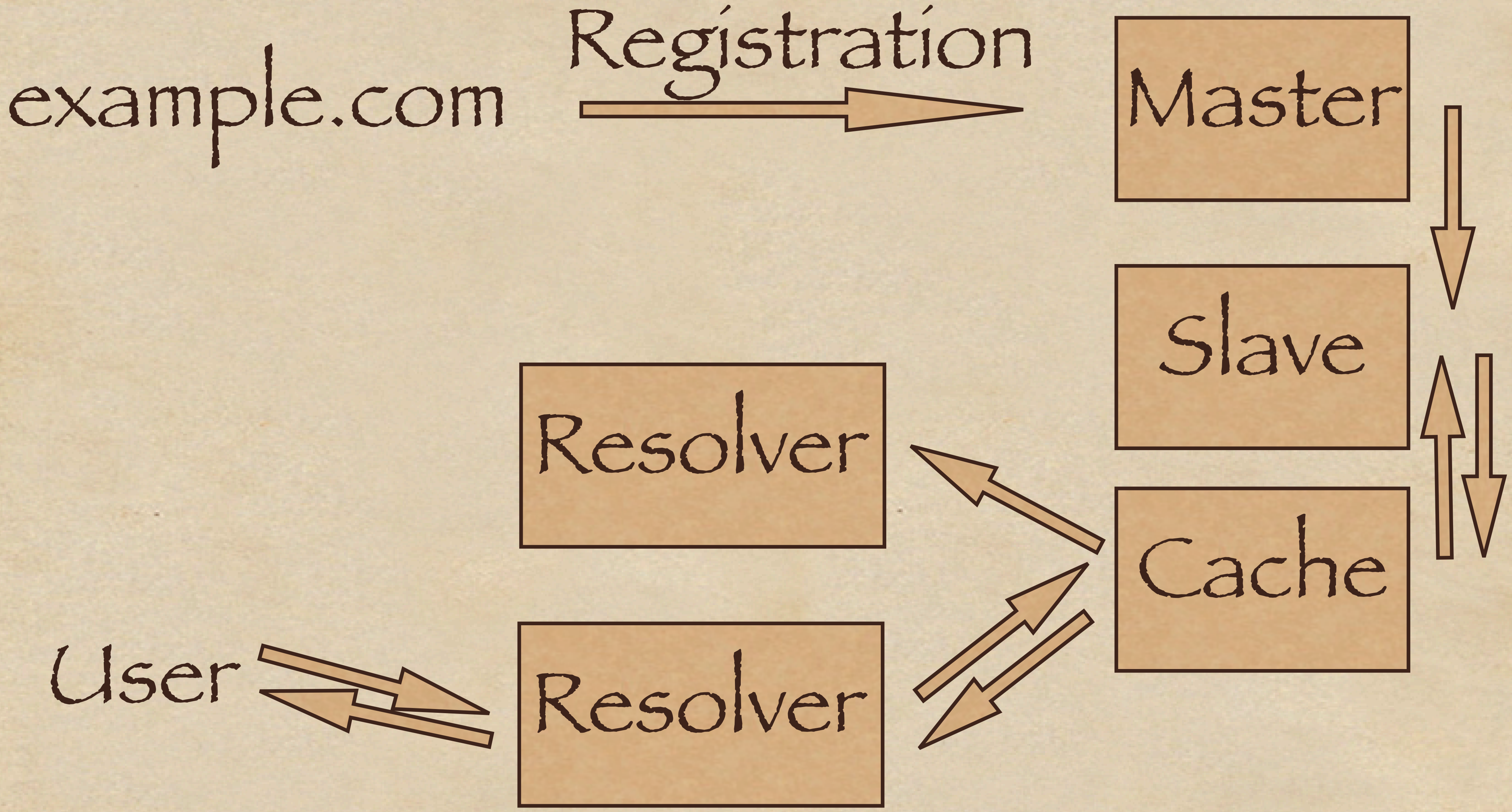
- Statement:
  - The DNS can transport any value of the octets in a DNS query
- Problem:
  - It is not decided what charset the octets are*

*Except for “case insensitive US-ASCII”

In more detail…

- Octets are saved in DNS at time of registration of a domain name
- Matching happens in the DNS server between the query and what's stored in the database (what's registered)
- DNS as a protocol doesn't include negotiation of context for queries

The storage problem



Protocol issues

- Old protocols can only handle a subset of US-ASCII (A-Z etc)
  - Remember local part of email addresses
- People want to use more characters when addressing resources (use Unicode)
- Two possible solutions:
  - Change protocols
  - “Encode” characters in US-ASCII

Before sending

1. Sender types domain name in application
2. If it is not Unicode already:

    Text is translated into Unicode

3. The Unicode string is encoded in US-ASCII

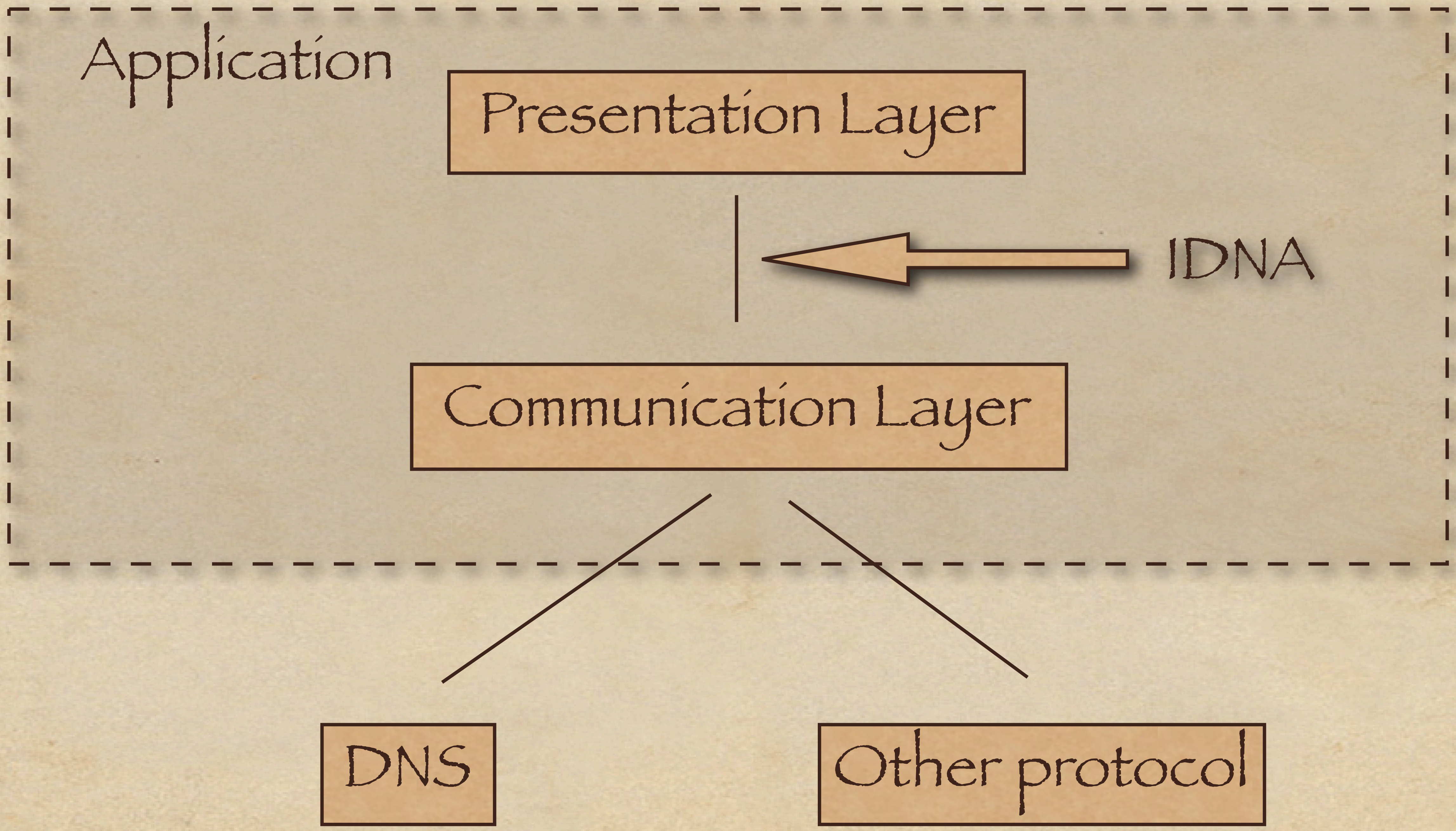
After receiving

1. Receiver decodes the US-ASCII string
2. If not Unicode can be used directly:

   Receiver translate text from Unicode to local charset

3. The domain name is presented to the receiver

Where is this applied?



IDNA in short...

1. Input from user

   Fa¨ltström

2. Apply Nameprep

   fältström

3. Apply Punycode

   xn--fltstrm-5wa1o

Implications

- Two different strings in Unicode might be “equal” according to the rules
- Two strings “looking” the same might be different Unicode strings and different strings according to the rules

Implications

- Example (same):
  - Fältström and fa¨ltström
  - xn--fltstrm-5wa1o
  - Today Faltstrom and faltstrom are equal
- IDNA does not change DNS rules

Implications

- Example (different):
  - CYRILLIC SMALL LETTER IE (U+0435) е
  - LATIN SMALL LETTER E (U+0065) e
- This is of course a font issue...
  - Both characters to the right in Lucida Grande Regular, 72 points

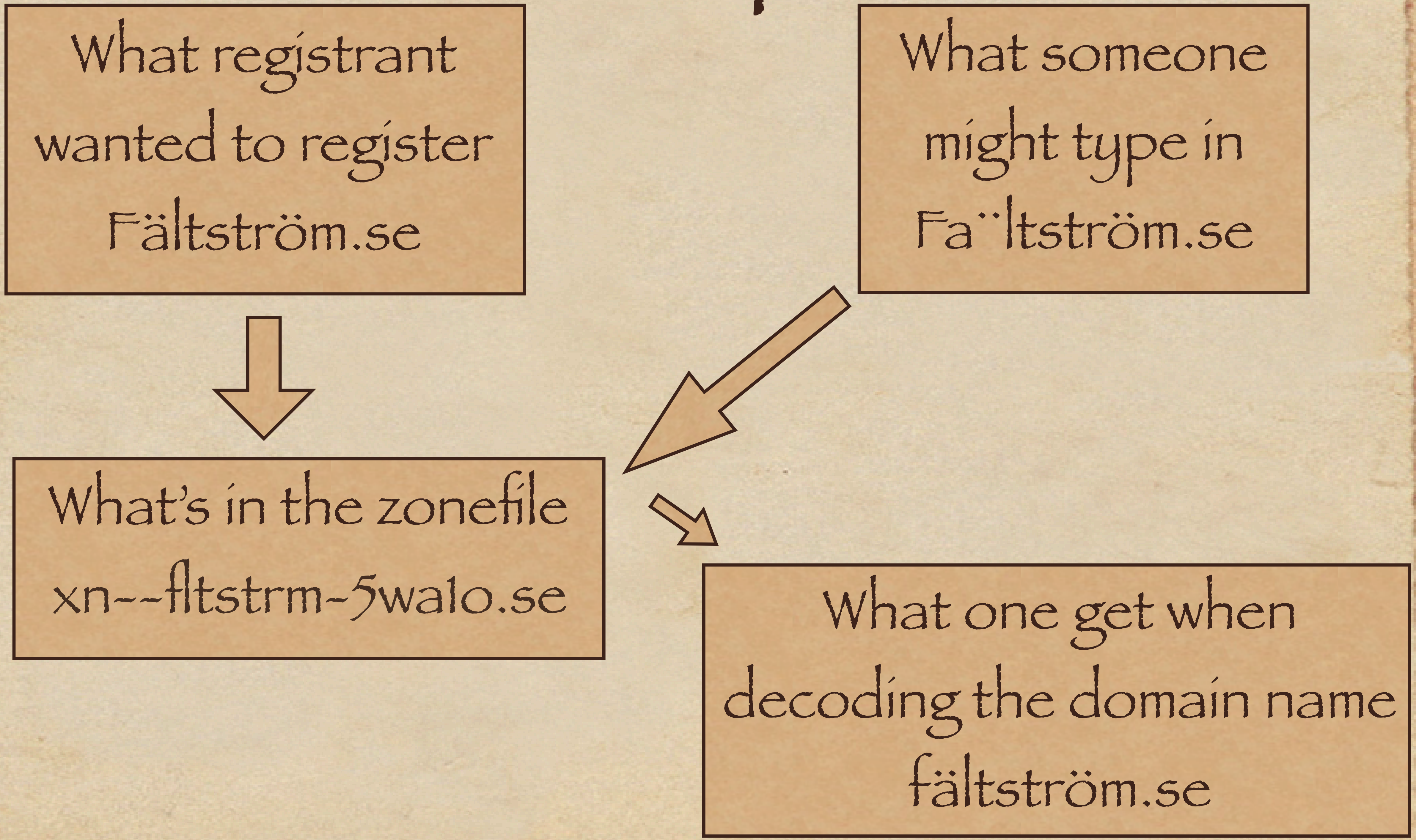
More implications

- What is “domain name” and what is in zone file are two different things
  - fältström.se

    xn--fltstrm-5wa1o.se
  - 费思哲.se

    xn--xwrt3x2r0b.se

Example



What registrant wanted to register Fältström.se

What someone might type in Fa¨ltström.se

What's in the zonefile xn--fltstrm-5wa1o.se

What one get when decoding the domain name fältström.se

RFC 4690 (IDN issues)

- Lists a number of issues with IDN's
- Outcome from some IAB discussions

Language specific matching

- Should **ö** match **ø**, or maybe **o**?
- Is variants (registration time “aliases”) as described in RFC 4290 a solution?

Multiple scripts

- Many scripts uses glyphs that look similar
  - Latin, Cyrillic, Greek
- Many languages can be expressed in multiple scripts
  - Asian languages in latin scripts

Normalizations

- Unicode contain several different models for representing characters
- Normalization compensate for this
- Normalization algorithms “have bugs”

URI's in printed form

- Many unicode strings might look the same but in reality they are different
- Similar to the problem mentioned earlier
    - Some glyphs are trademarks
    - Some fonts use curls even in latin that make them look similar to Thai

Bidirectional text

- Some text is right to left, and some left to right
- Should **1RtoL.2RtoL** be written as **LotR2.LotR1**?
    - What about **1LtoR.2RtoL**?
    - What about **http://1RtoL.2RtoL/**?

New version of Unicode

- The new version of Unicode (5.0) include some incompatible changes
- The changes are clearly mentioned
- Will applications and libraries know this?

What is happening?

- **draft-idnabis-issues-00.txt**
  - General issues with IDNA
- **draft-alvestrand-idna-bidi-00.txt**
  - Issues with bidirectional text
- **draft-faltstrom-idnabis-tables-00.txt**
  - What codepoints to include
- **http://www.ietf.org/html.charters/eai-charter.html**
  - EAI working group in IETF

Summary

- IDNA encodes Unicode characters in US-ASCII after normalisation so neither DNS, nor application level protocols have to understand Unicode
- Applications have to understand IDNA (and Unicode of course)
- Registries have to think more on what they do, and what their role is
- Email addresses (local part) might have a solution
- At coming IETF we will see discussions about IDN

Questions?

Patrik Fältström

Email: paf@cisco.com