Universal Acceptance Roadmap

for Domain Name Registry and Registrar Systems

Marc Blanchet & Michael Bauland

Tech Day ICANN 75
19 September 2022

Agenda

1. Overview
2. Analysis
3. The Gates
4. Test Cases
   Use of Labels
5. Example of one
   Registry Test Case
6. Example of one
   Registrar Test Case

Overview

- Study to help Domain Registries and Registrars make their systems support Universal Acceptance(UA)
  - Systems = registration systems (EPP, RDAP, Web, …), customer support, DNS zone generation, ...
  - Universal Acceptance = Σ IDN, EAI, long and new TLDs
- Report is currently in ICANN Public call for comments (closes october 17th)

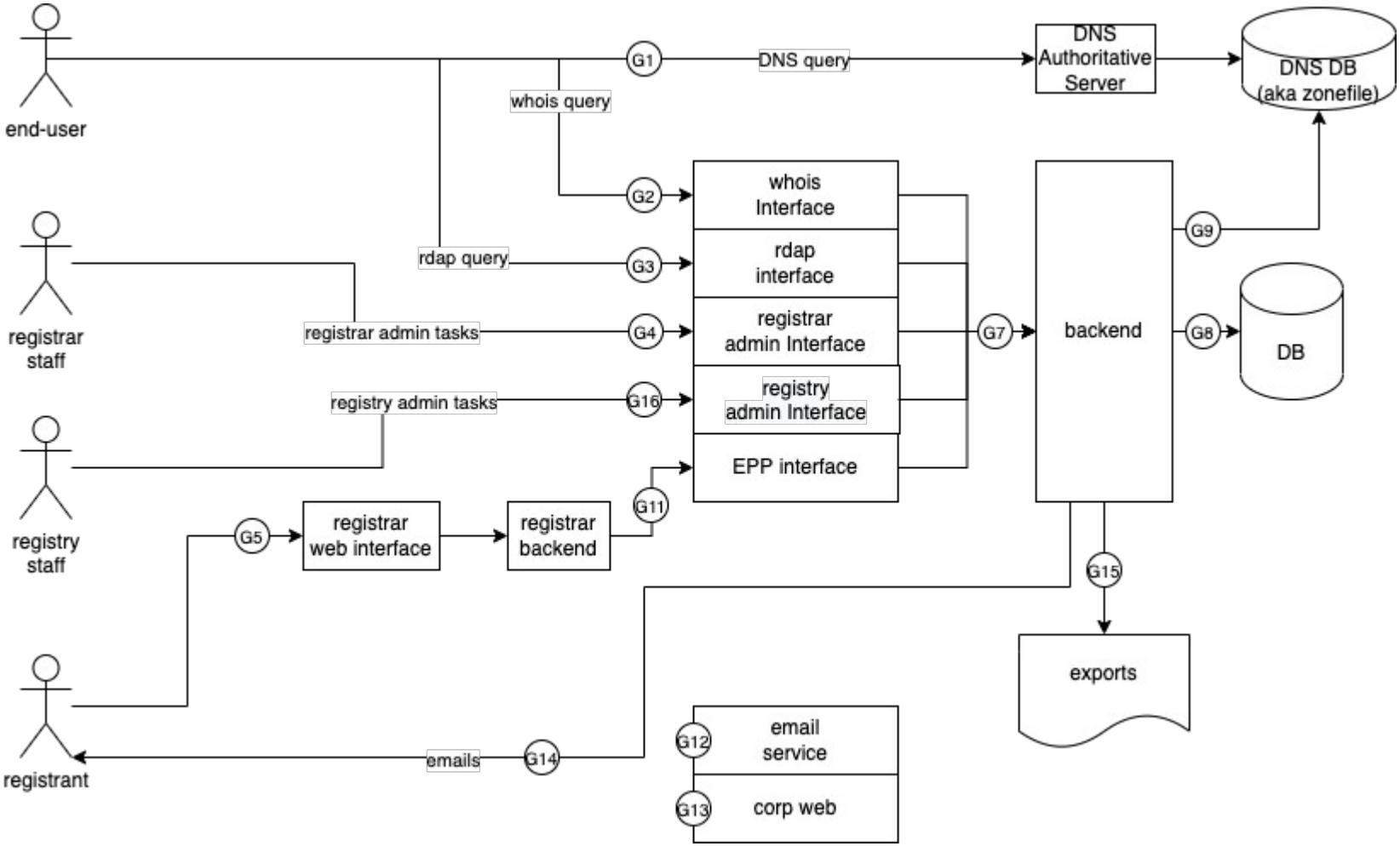
Study Methodology

- Uses the UASG026 UA Readiness Framework
  - Which is generic to any application
- Applies it to a generic model of Registry and Registrar systems
  - gTLD and ccTLD. Includes specifics of ICANN Contracted Parties Requirements
- Identifies gates within these systems where UA support needs to be verified
- Proposes test cases for this verification
- Analyses two registry systems and one registrar system as examples
  - Registry systems : Google Nomulus & KnippTANGO Registry Services
  - Registrar System : COREhub: GatewayNG
- Report targeted to registry and registrar operators, registry backend providers, developers and technical managers.

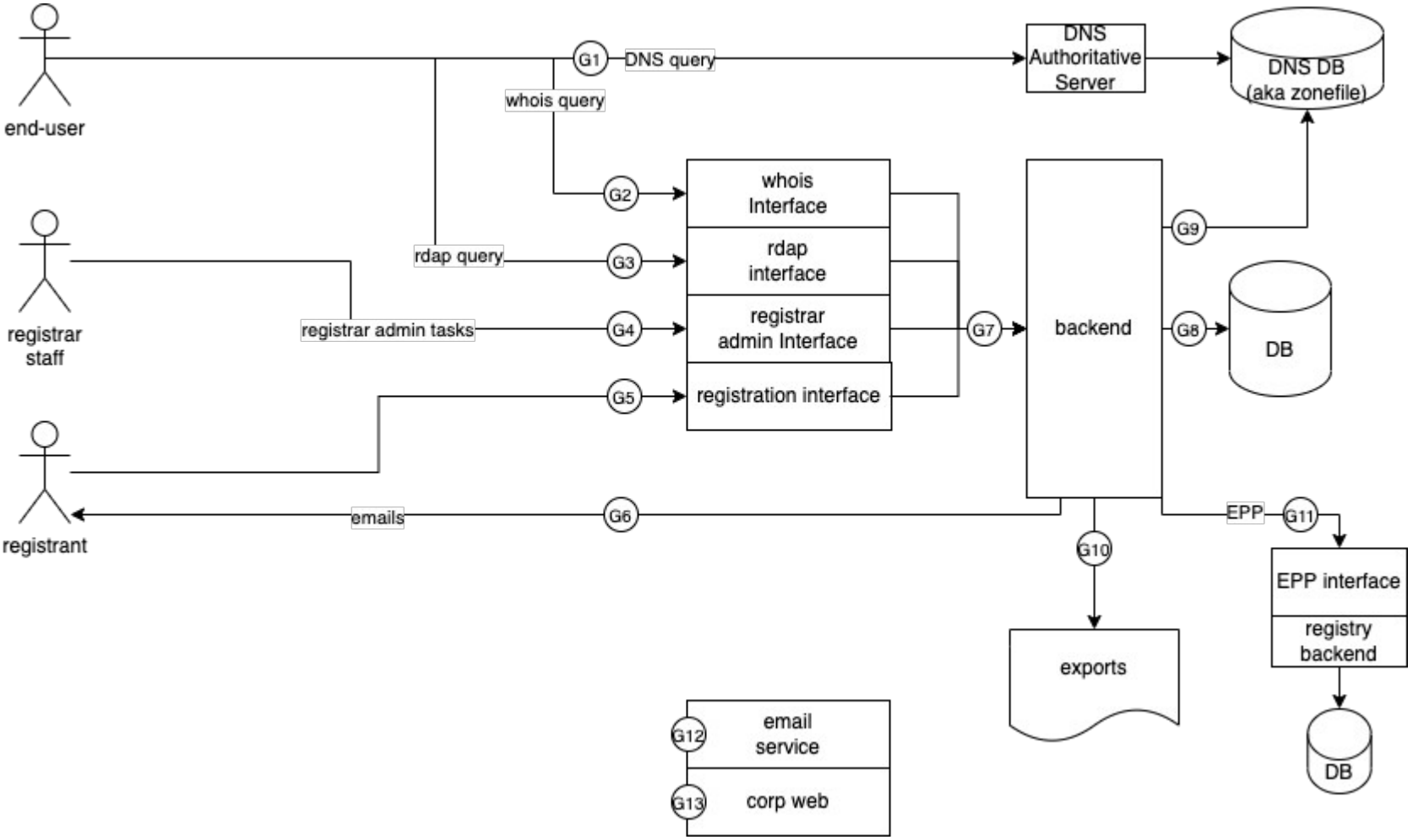
Architectures and Gates

- Registry High-Level Architecture
- Registrar High-Level Architecture
- For each identified gate, the expected behavior of the software is described.
- A set of test cases is provided

Registry High-Level Architecture

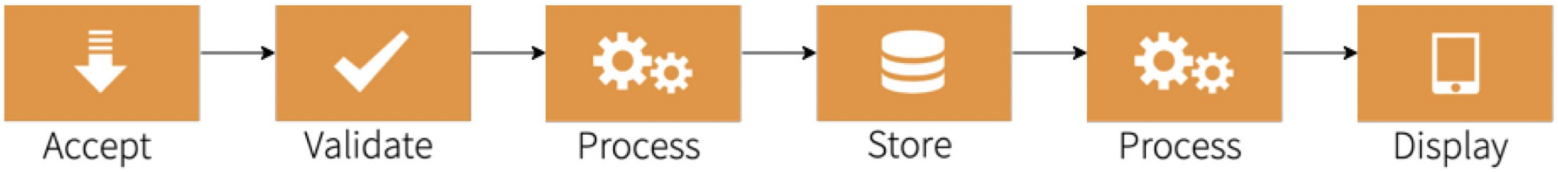


Registrar High-Level Architecture



Gates

- Gates are numbered, unique for both architectures
- Most are identical for both architectures, but some are different.
  - EPP usage is (obviously) different for registry and registrars : e.g. client vs server
- This is a generic architecture. Adapt accordingly to your own environment.
- Gates are identified based the UASG026 UA Readiness Framework model :



Gate Examples

- G10 and G15 identify exports to third-parties such as ICANN.
  - A list of relevant fields in these exports are identified with the expected format
- G7 identifies the backend. The report discusses important considerations about backend development and the fact that some language libraries and open-source software may or may not be UA compliant, therefore affecting the backend as a whole.

Other Considerations

- Protocols : report identifies key fields in EPP and Whois/RDAP protocols that should conform to UA.
- Generic considerations are provided about the processing of i18n elements such as :
  - string normalization
  - support of different scripts (directionality for example),
  - IDN handling (either UTF-8 or punycode)
  - ...

IDN Variants

- It is very important to note that IDN variants (different IDN labels that are considered equivalent for registration) have **NOT** been considered in this report.
- However, the impact of variants on these systems is pretty significant and therefore should be carefully thought of when starting the work.
- Some initial and minimal considerations for IDN variants are provided in the report.

Next Section and Next Steps

- Next section of this presentation discusses the tests made to two registry systems and one registrar system.
  - These are described in the appendices of the report
- The whole report (including appendices) is on ICANN Public comment. Comments are due by October 17th. Please read and provide comments.

Test Cases – Use of Labels

- Choice and Selection of Labels

Selection of Sample Test Cases - Labels

- Set up sample registry system serving .example and . テスト
- Sample non-existing test labels (obtained from the IANA Root Zone DB) used to build domain names and e-mail addresses, e.g.,
  - பரிசோ ≅ xn--hlcj6aya9esc7a (Tamil script)
  - 测试 ≅ xn--0zwm56d (Han script)
  - परीक्षा ≅ xn--11b5bs3a9aj6g (Devanagari script)
- Sample existing e-mail addresses (for checking receipt of e-mail)
  - michael@لكانبـازار ≅ michael@xn--igbi7fn.xn--mgbab2bd
  - grüün@knipp.de (EAI)

Registry Test Case Example

- Testing the TANGO Registry Services®

Registry Software Tested Features

- Registry Software
  - EPP
  - Control Panel (Web Interface)
  - DNS Name Server
  - Port 43 Whois
  - RDAP
  - Escrow Export

Registry Software Tested Features – Example

- Registry Software
  - EPP
    - Contact Update
  - Control Panel (Web Interface)
  - DNS Name Server
  - Port 43 Whois
  - RDAP
  - Escrow Export

EPP Contact Update - Request

Update a contact and set its e-mail address to δοκιμή@ テスト .பரிட்சை

Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
 <command>
  <update>
   <contact:update xmlns:contact=
    "urn:ietf:params:xml:ns:contact-1.0">
    <contact:id>C100033</contact:id>
    <contact:chg>
     <contact:email>δοκιμή@ テスト .பரிட்சை
 </contact:email>
    </contact:chg>
   </contact:update>
  </update>
  <clTRID>ABC-12345</clTRID>
 </command>
</epp>
```

Response (problem highlighted in red):

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
 <response>
  <result code="2306">
   <msg>Parameter value policy error</msg>
   <extValue>
    <value>
     <contact:email xmlns:contact=
"urn:ietf:params:xml:ns:contact-1.0">
      δοκιμή@ テスト .பரிட்சை
     </contact:email>
    </value>
    <reason>field value is disallowed by policy</reason>
   </extValue>
  </result>
  <trID>
   <clTRID>ABC-12345</clTRID>
   <svTRID>1651750771689-4065</svTRID>
  </trID>
 </response>
</epp>
```

EPP Contact Update - Request

Update a contact and set its e-mail address to 测试 @ 测试 . 测试

Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
 <command>
  <update>
   <contact:update xmlns:contact=
    "urn:ietf:params:xml:ns:contact-1.0">
    <contact:id>C100033</contact:id>
    <contact:chg>
     <contact:email> 测试 @ 测试 . 测试
</contact:email>
    </contact:chg>
   </contact:update>
  </update>
  <clTRID>ABC-12345</clTRID>
 </command>
</epp>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
 <response>
  <result code="1000">
   <msg>Command completed successfully</msg>
  </result>
  <trID>
   <clTRID>ABC-12345</clTRID>
   <svTRID>1651751124705-4068</svTRID>
  </trID>
 </response>
</epp>
```

EPP Contact Update – Analysis

- What went wrong?
  - Email address causing issues: δοκιμή@ テスト .பரிசை
  - Working email address: 测试 @ 测试 . 测试
- Both addresses uses non-ASCII characters both for the local part as well as for the domain part.
- It is also not the Tamil script as such. Another test showed that the following email address also works: δοκιμή@ テスト .பரி
- Debugging showed, the issue is with a 3rd party library (javax.mail), which simply marked the address as not valid.

EPP Contact Update – Solution

- How to fix the issue for TANGO?
  - It turned out the same e-mail address using the A-label version of the domain name was successfully validated by the javax.mail: δοκιμή@xn—zckzah.xn--hlcj6aya9esc7a
- Implementing a work-around:
  - validate the domain name part individually: テスト .பரிசை
  - if valid, convert domain name part to A-label: xn—zckzah.xn--hlcj6aya9esc7a
  - validate that e-mail address using the javax.mail library: δοκιμή@xn—zckzah.xn--hlcj6aya9esc7a
  - store the original e-mail address: δοκιμή@ テスト .பரிசை

Registrar Test Case Example

- Testing the CORE GatewayNG

Registrar Software Tested Features

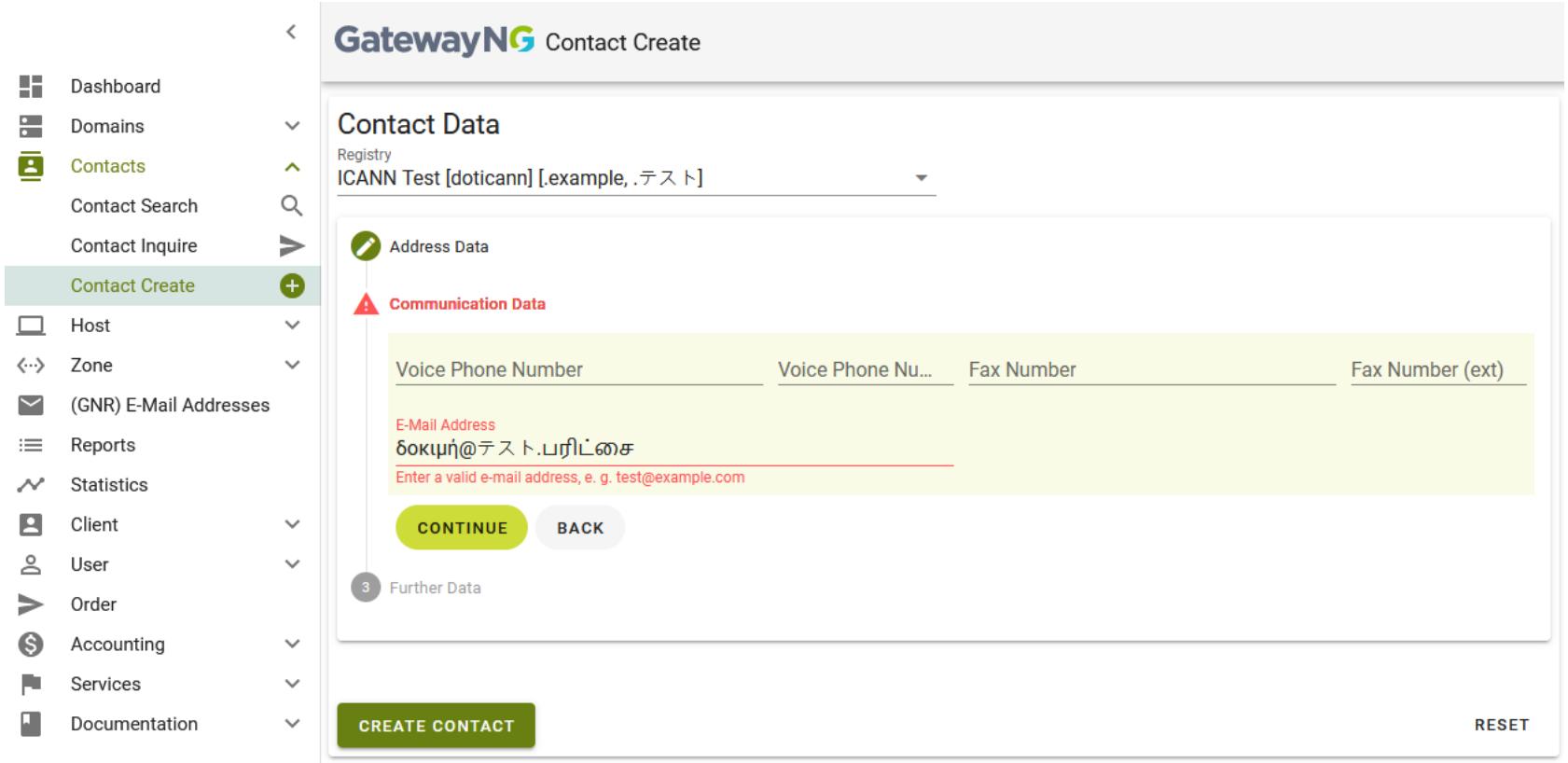
- Registrar Software
  - API (CORE Provisioning Protocol - Payload)
  - Control Panel (Web Interface)
  - DNS Name Server
  - Port 43 Whois
  - RDAP
  - Escrow Export
  - Email sending (WAP, Transfer Notifications)

Registrar Software Tested Features – Example

- Registrar Software
  - API (CORE Provisioning Protocol - Payload)
  - Control Panel (Web Interface)
    - Contact Create
  - DNS Name Server
  - Port 43 Whois
  - RDAP
  - Escrow Export
  - Email sending (WAP, Transfer Notifications)

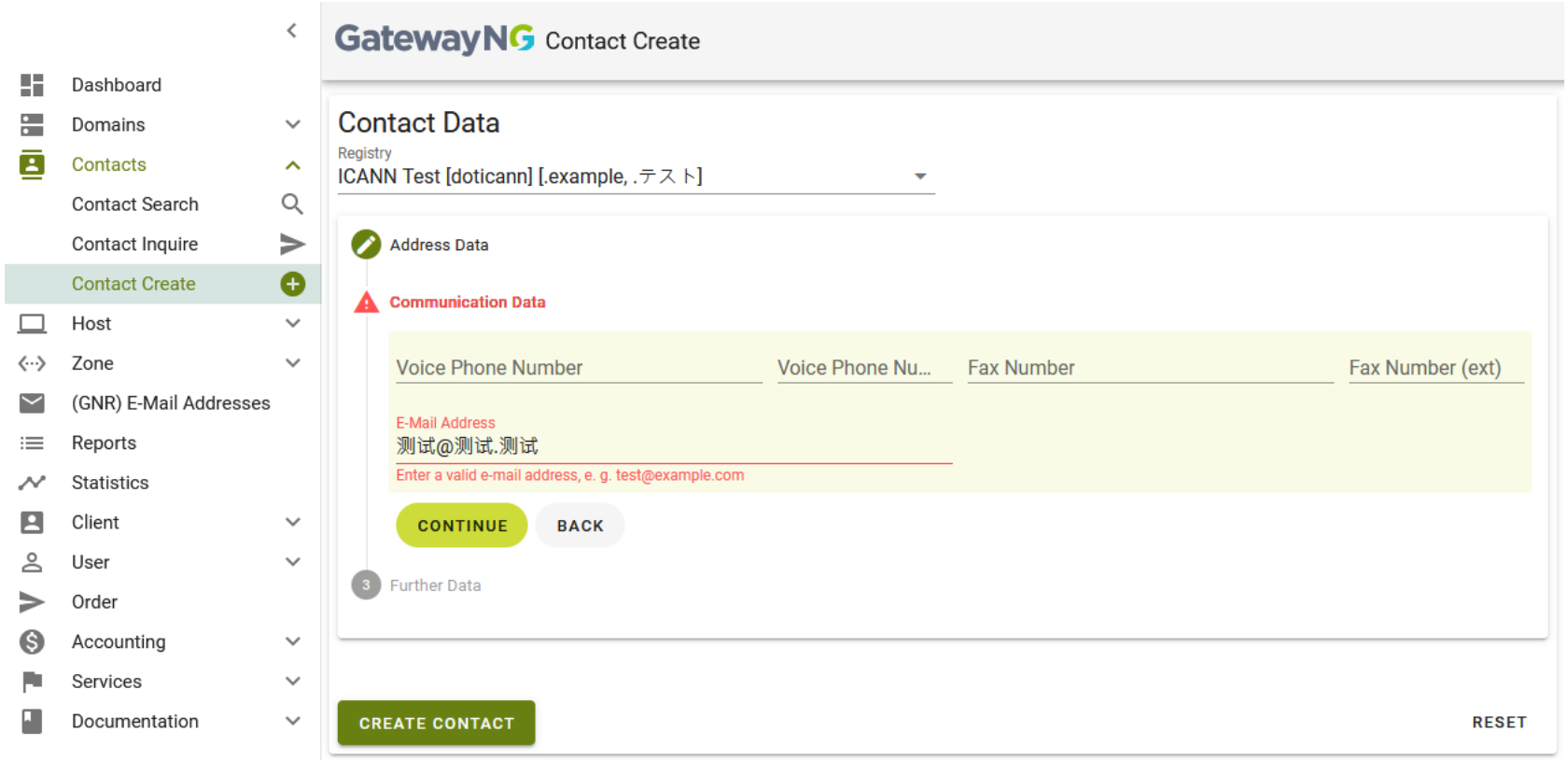
Web Interface Contact Create – Request

- Create a contact and with δοκιμή@ テスト .பரிட்சை



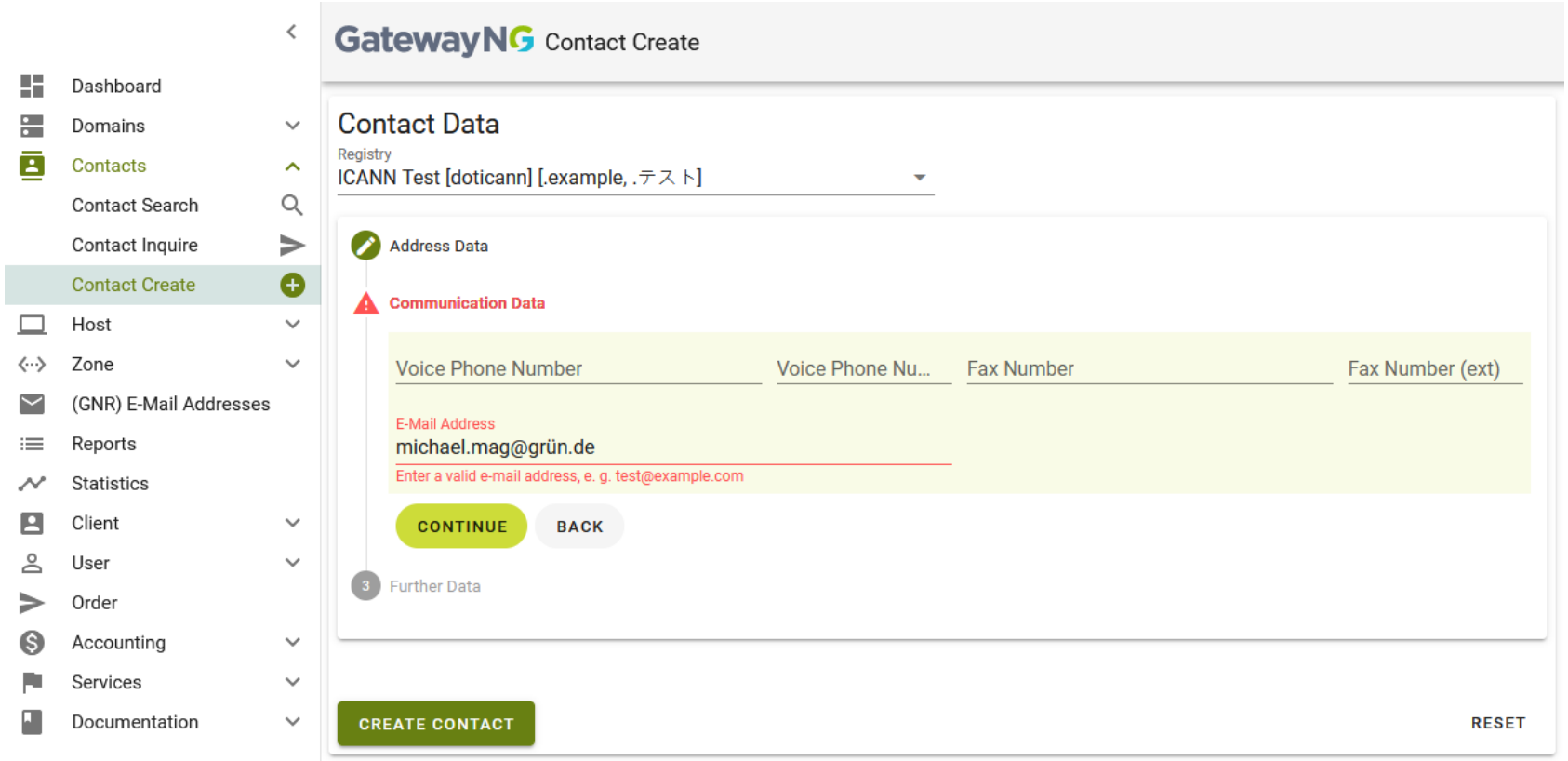
Web Interface Contact Create – Alternate Request

- Create a contact and with 测试 @ 测试 . 测试



Web Interface Contact Create – Alternate 2 Request

- Create a contact and with michael.mag@grün.de



Web Interface Contact Create – Analysis

- What went wrong?
  - Email addresses causing issues:
    - δοκιμή@ テスト .பரிசோ
    - 测试 @ 测试 . 测试
    - michael.mag@grün.de
- Presumably any non-ASCII character is rejected.
- Taking a look at the source code revealed:
  - Frontend is written using vue.js
  - Validation is carried out using vuelidate library, which uses a rather complex regular expression, but does not support any non-ASCII characters

Web Interface Contact Create – Solution

- How to fix the issue for GatewayNG?
  - It makes no sense to fix the regular expression and try to find one that takes all cases into account. Far too much work and error-prone.
- Let the backend do the work:
  - The backend Java code anyhow has to validate the email address (again).
  - Simplify the frontend javascript validation by just checking for very basic errors (i.e., not really an email address):
    <somestring>@<somestring>.<somestring>
    with <somestring> having no real restrictions.
  - The finer, more detailed validation is done afterwards in Java.

Engage with ICANN – Thank You and Questions

One World, One Internet

Visit us at **icann.org**


@icann


linkedin/company/icann


facebook.com/icannorg


slideshare/icannpresentations

youtube.com/icannnews


soundcloud/icann


flickr.com/icann


instagram.com/icannorg