Response to Question 25 (EPP)

1 Overview

The SRS of this proposed gTLD will use EPP for communication with registrars. The EPP interface is and in full compliance with the following RFCs and, where possible, entirely based on common standards:

- RFC 5730 - Extensible Provisioning Protocol (EPP)
- RFC 5731 - Extensible Provisioning Protocol (EPP) Domain Name Mapping
- RFC 5732 - Extensible Provisioning Protocol (EPP) Host Mapping
- RFC 5733 - Extensible Provisioning Protocol (EPP) Contact Mapping
- RFC 5734 - Extensible Provisioning Protocol (EPP) Transport over TCP
- RFC 5910 - Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP)
- RFC 3915 - Domain Registry Grace Period Mapping for the Extensible Provisioning Protocol (EPP)
- RFC 3735 - Guidelines for Extending the Extensible Provisioning Protocol (EPP)

Note that the objective is to base the EPP interface entirely on published RFCs and it is hence not planned to use any proprietary EPP commands. However, it is understood that some functionality required by ICANN cannot be implemented by means of EPP extensions specified in published RFCs. This includes support for:

- Trademark Clearing House: integration in the domain registration process (sunrise and claiming phase)
- IDN: exposure of language tags in the EPP interface
- IDN: selection of variants to be included in the DNS zone

Currently the following documents related to the topics listed above are available:

- draft-tan-epp-launchphase
- draft-obispo-epp-idn
- draft-kong-epp-variants-mapping

The registry backend operator participates in the standardization process and understands that the community is currently working on the respective documents. It is expected that specifications are published and implementable before the registry goes operational, which allows the registry operator to stick to its strategy of using only IETF RFC specified EPP extensions.

However, if such specifications are not available in a timely manner before the registry intends to go operational, draft specifications that reflect industry and community consensus will be considered instead in order to cover ICANNs functional requirements.

Via EPP, the following objects can be managed by registrars:

* domain objects
* host objects
* contact objects

The following commands are supported by the EPP interface:

* Session Management
** Login
** Logout
** Poll
** Hello
* Domain Commands
** Check domain
** Info domain
** Create domain
** Delete domain
** Renew domain
** Transfer domain
** Update domain (including “restore”)
* Host commands
** Check host
** Info host
** Create host
** Delete host
** Update host
* Contact commands
** Check contact
** Info contact
** Create contact
** Delete contact
** Update contact

According to the definitions in RFC 5730, the registry operator will apply for an EPP repository identifier with the IANA registry (http://www.iana.org/assignments/epp-repository-ids) as follows:

```
ID: immo, #x0069 #x006d #x006d #x006f
```

Registrant Contact: dotimmobilie GmbH <info@dotimmbilie.de>

The only language supported for message elements in EPP is English.

2 Session Management

The transport layer between EPP clients and the SRS EPP interface is protected using TLS with X.509 certificates. The registry will only use strong ciphers such as those required by the EPP RFC and listed below, but reserves the right to modify the list of ciphers depending on cryptographic developments.

- TLS 1.0 [RFC2246]: TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
- TLS 1.1 [RFC4346]: TLS_RSA_WITH_3DES_EDE_CBC_SHA
- TLS 1.2 [RFC5246]: TLS_RSA_WITH_AES_128_CBC_SHA

On top of the TLS based identity verification, login to the SRS's EPP interface is protected using two additional authentication factors, with one factor being the IP address of the client and the other factor being the clients' credentials. Failed login attempts are logged and reported. The administration of authorised IP address ranges can be performed by registrars via the Registrar's web interface or by contacting the helpdesk. Passwords can be changed via the EPP login command as described in RFC 5730, Section 2.9.1.1.

EPP sessions will be terminated by the server either after an idle timeout of 20 minutes or after the maximum session length of 24 hours. The registry operator reserves the right to restrict the number of concurrent EPP sessions per registrar (a limit of three sessions is currently defined but this may be amended depending on registry and TLD scaling requirements).

The EPP service also employs infrastructure elements and software measures to perform rate-limiting of EPP sessions (such measures may, for example, be required during landrush phases).

The SRS does not support EPP command pipelining.

3 Object Management

The registry supports the provisioning of contact, host and domain objects as defined in the respective RFCs and according to the lifecycle described in response to question 27.

Registration periods apply for domain objects only (in one year increments). The default initial registration and renewal period is 1 year. The client may choose another period of up to 10 years when issuing the respective request (total registration period of a domain must never exceed 10 years).

Since the registry uses grace periods, the grace period mapping of RFC 3915 is supported by the EPP interface. In particular, a restore command is issued as an extension to the update command, as described in this RFC. Furthermore, the restore report is also delivered to the registry via EPP.

The registry operator reserves the right to perform a garbage collection process on unlinked contact and unlinked external host objects. Internal hosts follow the lifecycle of their

superordinate domain and are not subject to garbage collection (for details refer to responses to question 27, 28).

For contact objects, only internationalized postalInfo elements are supported. All child elements as listed in the RFC are supported. Note that since the registry does not support contact transfers, contact authInfo is not used.

For the provisioning of DNSSEC trust chains, the EPP interface supports the extension described in RFC 5910 for accepting DS data (key data interface is not supported). Details on DNSSEC support are contained in response to question 43.

4 Domain Transfer

The domain transfer command has several subcommands. Note that a transfer can only be requested on domain objects but that the registry system will automatically transfer subordinate host objects when the superordinate domain is transferred. Contact objects are never transferred.

The set of transfer commands consists of the following subcommands: “request”, “approve”, “reject”, “cancel” and “query”. To request a domain transfer, the requestor sends a transfer request with a valid authInfo. The losing registrar is subsequently notified and can either reject or approve this request. In the event that the losing registrar doesn’t explicitly reject or approve the request, the registry will auto-approve the request after 5 calendar days. Before the transfer is approved, auto-approved or rejected, (i.e., the domain is in pendingTransfer state) the requestor may cancel it. A detailed description of the domain transfer lifecycle is contained in response to question 27 (Figure Q27-02).

AuthInfo is required for all domain objects. This information is necessary in order to authenticate a domain transfer process. The registry system requires that the authInfo is at least 8 characters long with a maximum length of 32 characters. Furthermore, at least one alphanumeric character (‘A’ to ‘Z’; both lower and uppercase letters), and at least either one numeric character (‘0’ – ‘9’) or one special character are required for each authInfo.

5 Status values

The domain object supports the following status values (as described in Section 5 of RFC 5731):

- **inactive** (to indicate that no hosts are associated with the object). This status value is set automatically by the server.
- **ok** (default status) set automatically by the server. This status value is never combined with any other status values.
- **pendingTransfer** is set by the server when the domain name is subject to a pending transfer
- **pendingDelete** is set by the server when the domain name is subject to deletion. Note that the registry also supports the RGP grace periods - redemption and pending delete as listed below.
- **serverHold/clientHold** set when the domain object should not appear in the zone.

- **serverUpdateProhibited/clientUpdateProhibited** set when the domain name cannot be updated due to server or client policy.
- **serverTransferProhibited/clientTransferProhibited** set when the domain name cannot be transferred due to server or client policy.
- **serverDeleteProhibited/clientDeleteProhibited** set when the domain name cannot be deleted due to server policy or client provisions.
- **serverRenewProhibited/clientRenewProhibited** set when the domain name is not eligible for renewal.

Each domain object will always have at least one associated status value.
Additionally, domain objects support the following status values related to the grace period mapping as per RFC 3915, Section 3.1:

- addPeriod
- autoRenewPeriod
- renewPeriod
- transferPeriod
- redemptionPeriod
- pendingRestore
- pendingDelete

The contact object supports the following status values (as described in Section 2.2 of RFC 5733):

- **linked** (when the object is used in at least one domain name object)
- **ok** (default status)
- **serverUpdateProhibited/clientUpdateProhibited** (for server or client policy reasons, modifications to the object are not allowed)
- **serverDeleteProhibited/clientDeleteProhibited** (for server or client policy reasons removal of the object from the registry is not allowed)

Each contact object will always have at least one associated status value.

The host object supports the following status values (as described in Section 2.3 of RFC 5732):

- **linked**: Set by the registry when a host is referenced by at least one domain
- **ok** (default status)
- **pendingTransfer**: Set on internal host objects when the superordinate domain is pending transfer.
- **serverDeleteProhibited/clientDeleteProhibited** (for server or client policy reasons removal of the object from the registry is not allowed)
- **serverUpdateProhibited/clientUpdateProhibited** (for server or client policy reasons, modifications to the object are not allowed)

Each host object will always have at least one associated status value.

6 EPP Server Implementation

The EPP server implementation is based on the Apache HTTP server, with the HTTP protocol handler replaced with a custom, Perl-based EPP handler. This allows for the reuse of Apache's session management, logging and resource allocation functionality. EPP systems based on this software have been deployed in production since 2004. The software has been continuously developed, in order to accommodate policy changes and scalability requirements.

The EPP software variant for the proposed TLD is already available and has already been deployed on prototype systems (with the exception of functionality where specifications are unclear at the time of this writing, i.e. Trademark Clearing House integration).

Since no proprietary extensions are planned, no EPP templates and no EPP extension schemas are provided in response to this question. Schemas and examples of the commands supported are included in the respective RFCs.

7 Resource Planning

The technical resources required for the operation of the EPP server (as part of the SRS) are described in response to questions 32 and 24. For EPP development and evaluation of related issues, the Registry Back-End Operator has a highly skilled research & development team of 5 persons, of which 3 people are intimately familiar with the details of the EPP protocol. They also monitor and contribute to the discussions within the IETF regarding future developments of EPP ("provreg" mailing list).

All technical staff are trained on the day-to-day operations of the EPP service. The helpdesk team is trained and experienced in troubleshooting EPP support problems with Registrars, and can escalate to the EPP experts or even core developers in case of more complex problems.