

Registry System Testing

RST-API and Test Specification Overview

Gustavo Lozano, Gavin Brown

14 December 2023



Agenda

1. Overview - Why are we doing this?
2. Objective for this meeting
3. RST workflow overview for RSP evaluation
4. RST-API overview
5. Anatomy of a test object
6. Key API endpoints
7. RST-API Specification
8. RST Test Specification components
9. Test Plans
10. Test Suites
11. Anatomy of a test case
12. Test plan example
13. Test suite example
14. Test case example
15. RST Test Specification
16. Version Control & Predictability

Overview - Why are we doing this?

From the [Final Report on the New gTLD Subsequent Procedures Policy Development Process](#):

- **Recommendation 39.1:**
 - *ICANN must develop a set of Registry System tests designed to demonstrate the technical capabilities of the registry operator.*

- **Recommendation 39.4:**
 - *Registry System Testing (RST) must be efficient.*

Meeting Objective

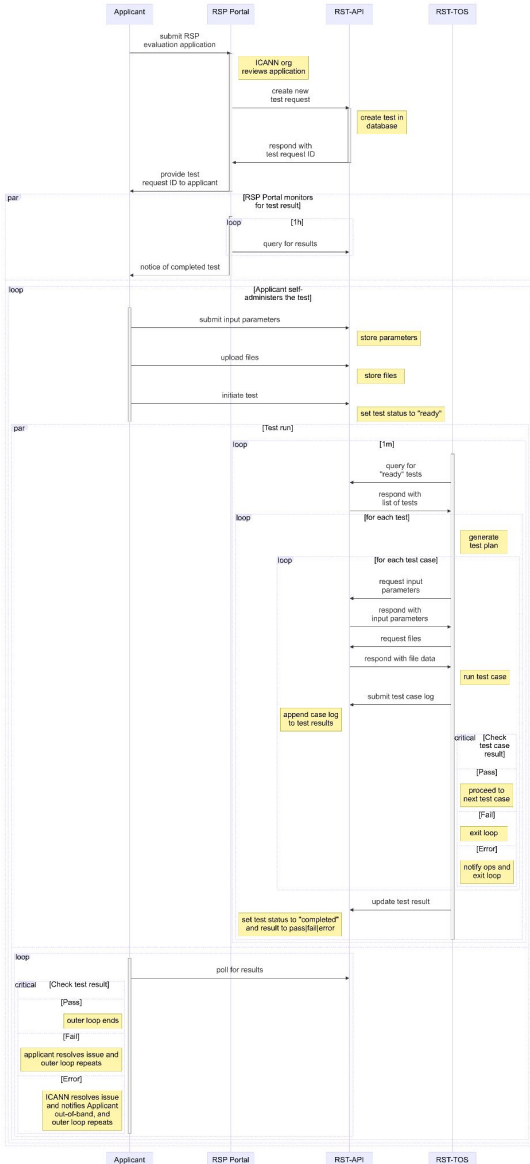
We want your feedback on:

- RST-API
 - The API is stable and unlikely to change much
 - Some object properties may be added/changed
- RST Test Specifications
 - There are hundreds of individual test cases; many are still in the early stages of definition
 - Each test case is tagged with its maturity level:
 - alpha - very rough outline, much more work needed
 - beta - complete but likely to require further changes
 - gamma - finalized and ready for review
 - You may want to focus just on those that are beta and gamma

RST-API General Overview

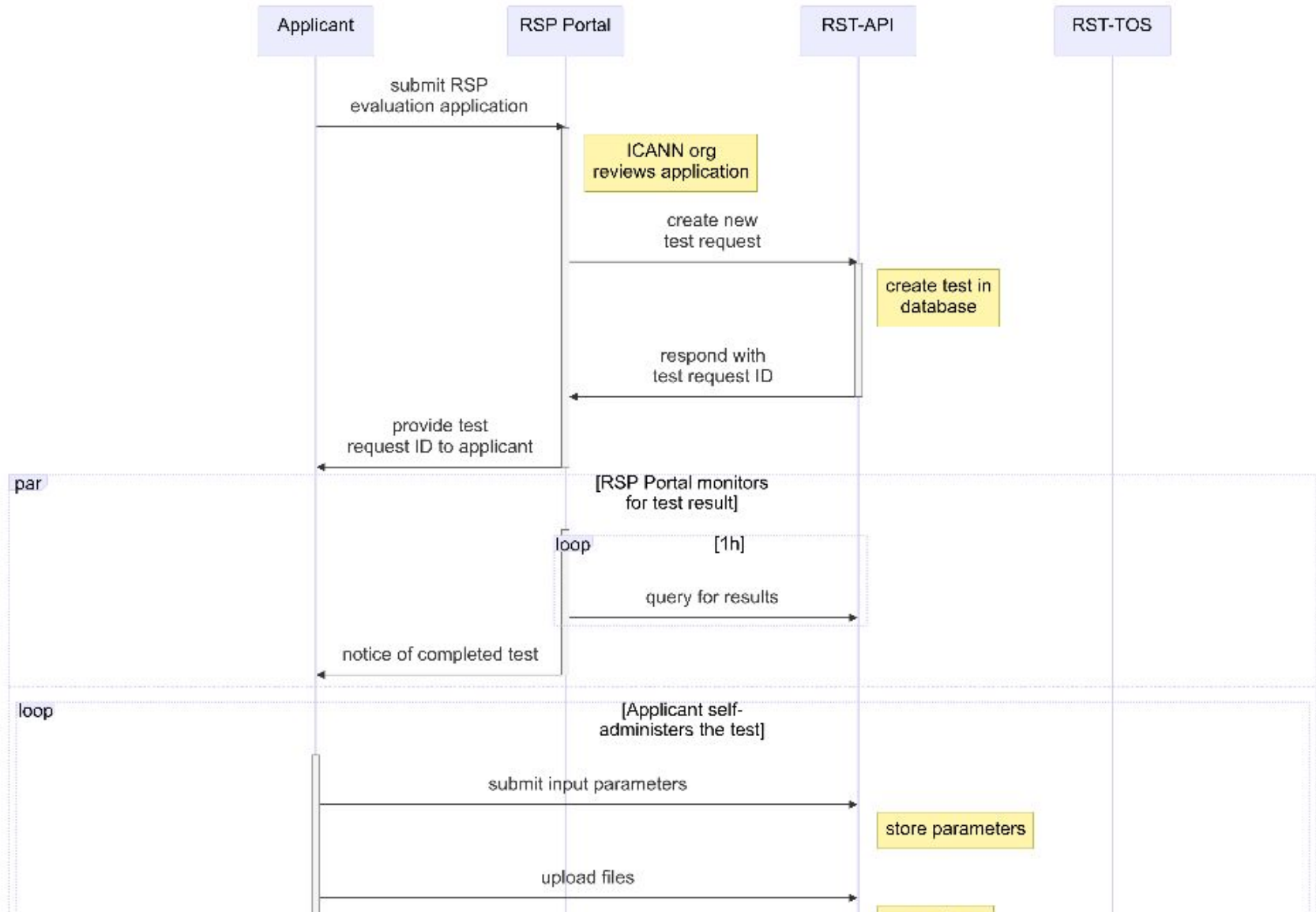
- RESTful API with JSON encoding
- OpenAPI v3 specification
- Client certificate authentication using TLSA records
- Production and Operational Test and Evaluation Environment (OT&E) environments

RST Workflow (RSP Evaluation)

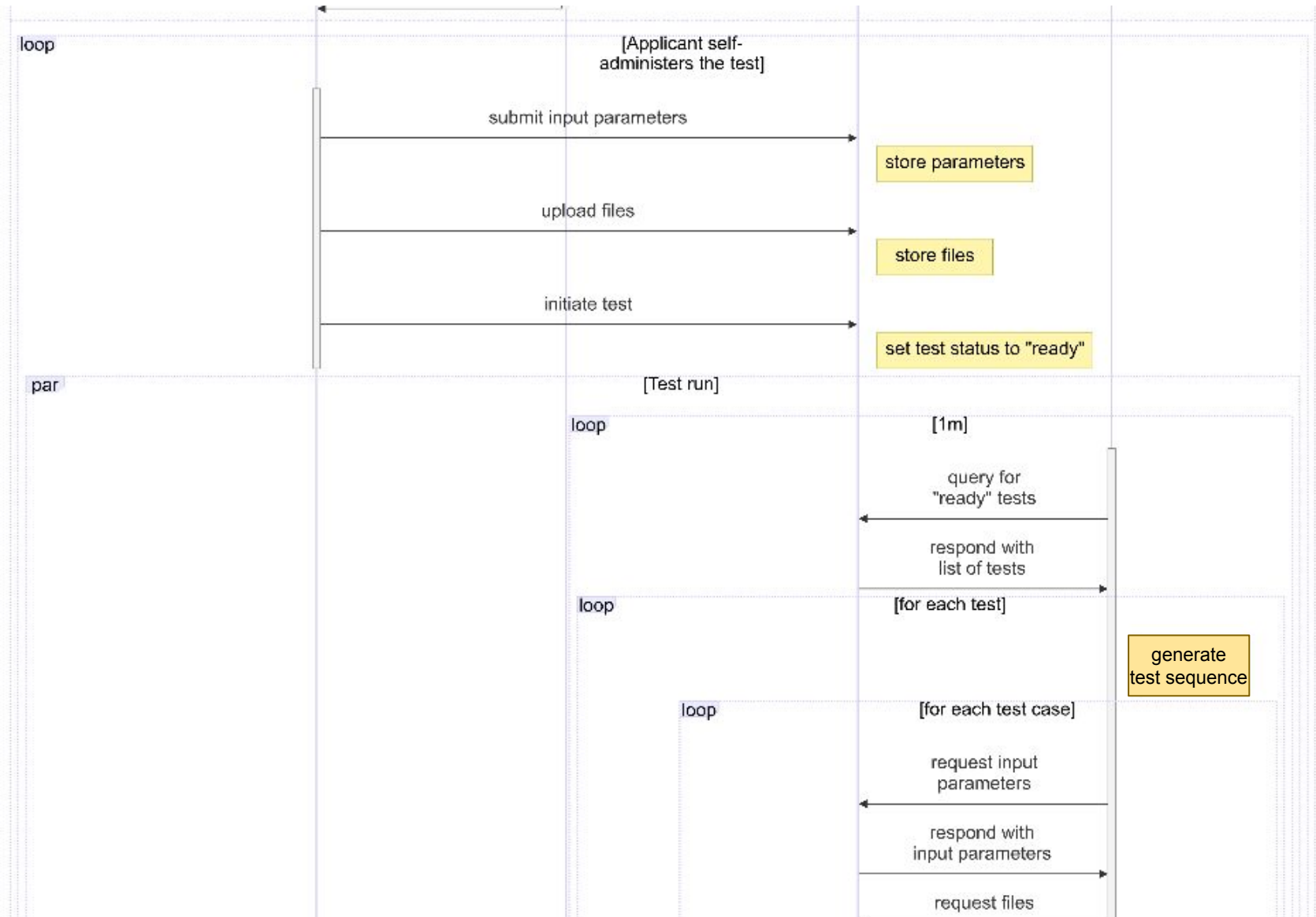


- An RST phase is incorporated into RSP evaluation after the technical screening.
- The RST phase will use the new RST v2.0 service being developed as part of the next round activities.
- Apart from the initial setup of the test, the system is self-service, via the new RST-API.
- All testing is automated and, with few exceptions, results will be made available quickly.
- There is a separate test plan for each RSP type: applicants seeking evaluation for multiple RSP types will undergo multiple tests.

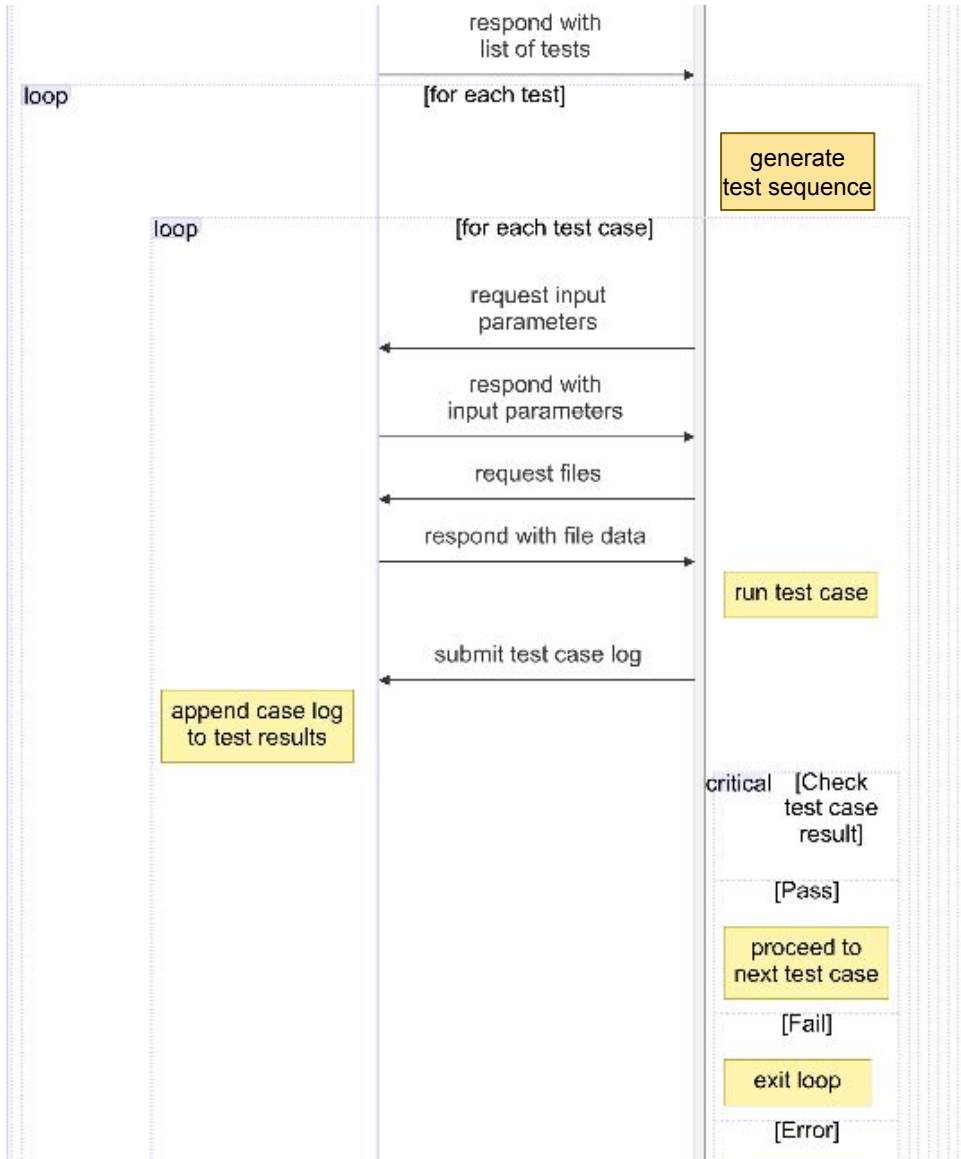
RST Workflow (RSP Evaluation) 1/4



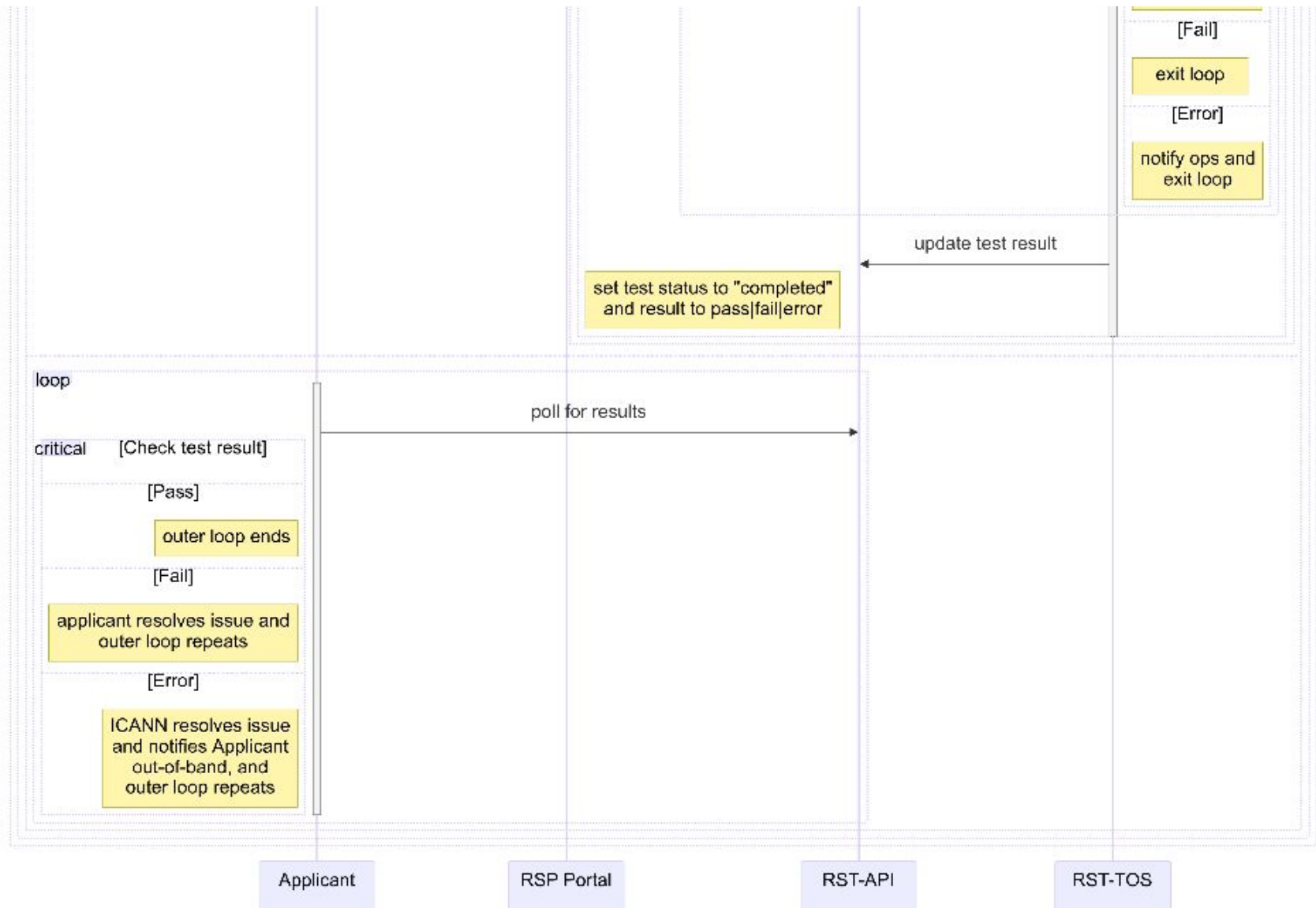
RST Workflow (RSP Evaluation) 2/4



RST Workflow (RSP Evaluation) 3/4



RST Workflow (RSP Evaluation) 4/4



Anatomy of a Test Object (1/4)

```
{
  "id": 987654321,
  "applicationID": "foo123",
  "tlds": [
    {
      "xn--8pvz8d": {
        "zh_CN": {
          "supportLevel": 3,
          "variantPolicy": "allocatable"
        }
      },
      "xn--8pvxs": {
        "zh_TW": {
          "supportLevel": 3,
          "variantPolicy": "allocatable"
        }
      }
    },
    {
    },
    {
    },
    {
      "example": {}
    }
  ]
  // ...
}
```

Anatomy of a Test Object (2/4)

```
{  
  // ...  
  
  "clientIDs": [  
    "tlsa.rsp.tech",  
    "tlsb.rsp.tech"  
  ],  
  "testPlan": "StandardPreDelegationTest",  
  "testPlanVersion": "v3.1.125",  
  "dueDate": "2024-11-29T14:50:55.977Z",  
  "dateRequested": "2023-11-29T14:50:55.978Z",  
  "dateUpdated": null,  
  "dateStarted": null,  
  "dateCompleted": null,  
  "status": "inputs-needed",  
  "result": null,  
  
  // ...  
}
```

Anatomy of a Test Object (3/4)

```
{
  // ...

  "inputs": {
    "example.name": "example.value"
  },
  "missingInputs": [
    "epp.kitchenSink",
    "rdap.coffeePotURL"
  ],
  "files": [
    {
      "name": "example_20231008_full_S1_R0.ryde",
      "type": "application/octet-stream",
      "uploaded": "2023-11-29T14:50:55.978Z",
      "href": "/v1/test/987654321/files/example_20231008_full_S1_R0.ryde"
    }
  ],
  // ...
}
```

Anatomy of a Test Object (4/4)

```
{
  // ...

  "results": [
    {
      "caseID": "epp-01",
      "result": "pass",
      "description": "successfully resolved epp.hostname for IPv4 and IPv6",
      "dateStarted": "2023-11-29T14:50:55.978Z",
      "dateCompleted": "2023-11-29T14:50:55.978Z",
      "log": [
        {
          "code": "MNEMONIC_ERROR_CODE_0123",
          "severity": "WARNING",
          "timestamp": "2023-11-29T14:50:55.978Z",
          "message": "Something informative just happened!"
        }
      ],
      "context": {
        "important.note": "the contents of this object are purely illustrative",
        "host.os": "Linux",
        "epp.hostname": "epp.rsp.tech"
      }
    }
  ]
}
```

Key Endpoints

- Submit input parameters

POST /v1/test/{id}/inputs

- Upload files

POST /v1/test/{id}/files

- Start a test

PATCH /v1/test/{id}

- Review test result

GET /v1/test/{id}

RST-API Specification

- OpenAPI specification v3.1.0: <https://spec.openapis.org/oas/v3.1.0>
- Swagger Codegen: <https://swagger.io/tools/swagger-codegen/>
- RST-API specification: <https://icann.github.io/rst-api-spec/>
- GitHub repository: <https://github.com/icann/rst-api-spec/>
- **Interested parties are welcome to report issues and submit pull requests.**

RST Test Specification Components

- The test specification has a hierarchical structure.
- Each *test plan*
 - has one or more *test suites*
 - each of which have one or more *test cases*.
- Test cases are run in order, and can have intercase dependencies.
- If a test case fails, the test stops and subsequent test cases aren't run.
- The self-service API allows test subjects to quickly re-run the test sequence once an issue has been fixed.

RST Test Specifications

- There are nine top-level *Test Plans*:
 - RSP evaluation tests:
 - Main RSP Evaluation Test
 - DNS RSP Evaluation Test
 - DNSSEC RSP Evaluation Test
 - SRS Gateway RSP Evaluation Test
 - IDN Test (if a Main RSP applicant wishes to support IDNs)
 - *BAU test plans*:
 - *Pre-Delegation Test*
 - *RSP Change Test*
 - *DNS RSP Change Test*
 - *IDN Test*
 - *SRS Gateway Test*

Test Suites

- **Main RSP:**
 - Registration Data Access Protocol (RDAP)
 - Extensible Provisioning Protocol (EPP)
 - Registry Data Escrow (RDE)
 - Minimum Rights Protection Mechanisms (RPMs)
- **IDN Test (optional)**
- **DNS RSP:**
 - Authoritative DNS Service
- **DNSSEC RSP:**
 - DNS Security Extensions (DNSSEC)
 - DNSSEC Operations
- **SRS Gateway RSP:**
 - SRS Gateway

Other Specification Elements

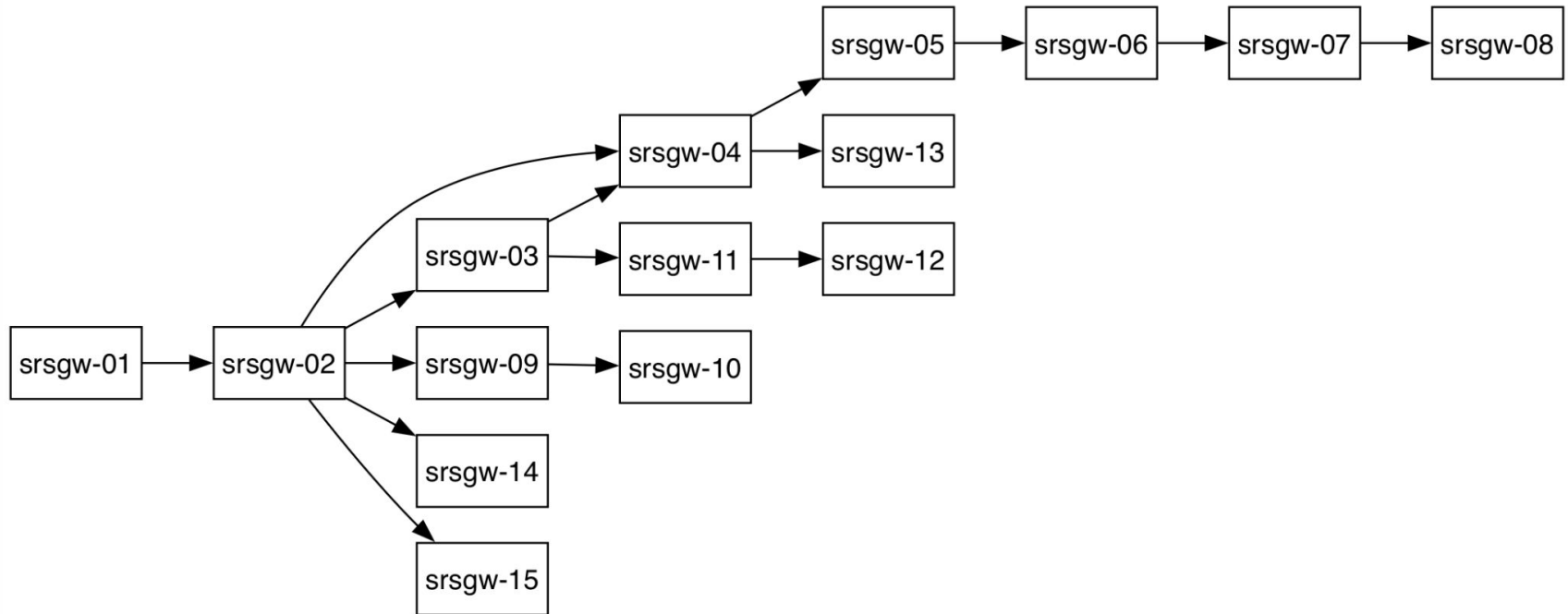
- **Input parameters:** Information required for a test to run must be provided by the test subject before the test starts.
 - Examples: EPP server hostname, credentials, RDAP base URL, RDE deposits
- **Resources:** Information provided by ICANN that may be useful to test subjects when setting up their systems.
 - Examples: Client certificates/CSRs, probe IP lists, etc
- **Error codes:** Error messages that a test case may emit.
- Input parameters, resources, and error codes are aggregated for each *test suite* and *test plan*.

Anatomy of a Test Case

A Test Case has:

- **1x Case ID** (e.g., epp-01)
- **1x Description:** Human readable description of what the test will do and pass/fail criteria.
- **0+ Input parameters**
- **1+ Error codes**
- **0+ Resources**
- **0+ Dependencies:** Other test cases that must run (and pass) before this test case can run. This allows a *test sequence* to be generated.

Test Suite Sequence Example



Test Plan Example (1/3)

3.6. Main RSP Evaluation Test



3.6.1. Description

The Main RSP is responsible for the creation and maintenance of domain name registrations in a Shared Registration System (SRS). This encompasses the lifecycle of a domain name registration using domain registrars and protocols such as the Extensible Provisioning Protocol (EPP) and adherence to policies regarding the use and transparency domain name registrations through reporting, the Registration Data Access Protocol (RDAP), and other mechanisms.

Pass/fail criteria

As with all other [test plans](#), for this test to pass, **all** the listed test cases **MUST** pass: if **any** fail, then the test as a whole will fail.

▼ 3.6.2. Plan ID

The following Test Plan ID may be used with the RST API:

```
MainRSPEvaluationTest
```

▼ 3.6.3. Test suites

This plan uses the following test suites:

1. [Registration Data Access Protocol \(RDAP\)](#)
2. [Extensible Provisioning Protocol \(EPP\)](#)
3. [Registry Data Escrow \(RDE\)](#)
4. [Minimum Rights Protection Mechanisms \(RPMs\)](#)

Test Plan Example (2/3)

▼ 3.6.4. Resources

The following resources may be required to prepare for this test plan:

- [epp.clientACL](#)
- [epp.clientCertificate](#)
- [epp.tlsCertificateStore](#)
- [rde.encryptionKey](#)

▼ 3.6.5. Errors

This test plan may produce the following errors:

- [EPP_CONNECT_ERROR \(ERROR\)](#)
- [EPP_GENERIC_COMMAND_ERROR \(ERROR\)](#)
- [EPP_LOGIN_ERROR \(ERROR\)](#)
- [EPP_SCHEMA_VALIDATION_ERROR \(ERROR\)](#)
- [EPP_SUNRISE_AVAILABILITY_CHECK_INVALID_RESULT \(ERROR\)](#)

▼ 3.6.6. Input parameters

This plan requires the following input parameters:

- [epp.clid \(string\)](#)
- [epp.hostModel \(string\)](#)
- [epp.hostName \(string\)](#)
- [epp.pwd \(string\)](#)
- [epp.registeredContacts \(array\)](#)
- [epp.registeredNames \(array\)](#)
- [epp.registeredNameservers \(array\)](#)
- [epp.restoreReportRequired \(boolean\)](#)
- [epp.transferClid \(string\)](#)
- [epp.transferPwd \(string\)](#)
- [general.gracePeriodsSupported \(boolean\)](#)
- [general.minimalPublicDataSet \(boolean\)](#)
- [general.sunriseModels \(string\)](#)
- [rdap.baseURLs \(object\)](#)
- [rdap.testDomains \(object\)](#)
- [rdap.testEntities \(object\)](#)
- [rdap.testNameservers \(object\)](#)
- [rde.depositFile \(file\)](#)
- [rde.publicKey \(file\)](#)
- [rde.signatureFile \(file\)](#)

Test Plan Example (3/3)

▼ 3.6.7. Required files

- [rde.depositFile](#)
- [rde.publicKey](#)
- [rde.signatureFile](#)

▼ 3.6.8. RST-API example

POST /test/987654/inputs HTTP/1.1
Content-Type: application/json
Content-Length: 1054

```
{
  "app.clid" : "clid-01",
  "app.hostModel" : "objects",
  "app.hostName" : "app.rsp.tech",
  "app.pwd" : "foo2bar",
  "app.registeredContacts" : [
    "abc123",
    "def321"
  ],
  "app.registeredNames" : [
    "example.example1",
    "example.example2"
  ],
  "app.registeredNameservers" : [
    "ns1.example.com",
    "ns2.example.org"
  ],
  "app.restoreReportRequired" : false,
  "app.transferClid" : "clid-02",
  "app.transferPwd" : "foo3bar",
  "general.gracePeriodsSupported" : true,
  "general.minimalPublicDataSet" : true,
  "general.sunriseModels" : "start-date",
  "rdap.baseURLs" : {
    "example" : "https://rdap.example.com/example/"
  },
  "rdap.testDomains" : {
    "example" : "example.example"
  },
  "rdap.testEntities" : {
    "example" : "9999"
  },
  "rdap.testNameservers" : {
    "example" : "ns1.example.com"
  },
  "rde.depositFile" : "example_20231004_FULL_S1_R0.ryde",
  "rde.publicKey" : "rsp-rde-signing-key.asc",
  "rde.signatureFile" : "example_20231004_FULL_S1_R0.sig"
}
```

Test Suite Example (1/2)

4.9. Minimum Rights Protection Mechanisms (RPMs)

4.9.1. Description

This test suite verifies an RSP's support for the minimum Rights Protection Mechanisms (RPMs), and the Launch Extension (RFC 8334) if supported.

Pass/fail criteria

As with all other [test suites](#), for this test suite to pass **all** the listed test cases **MUST** pass: if **any** fail, then the suite as a whole will fail.

▼ 4.9.2. Test cases

This suite uses the following test cases:

1. [minimumRPMs-01 - Sunrise domain <check> command test](#)
2. [minimumRPMs-02 - Claims <check> command test](#)
3. [minimumRPMs-03 - Trademark <check> command test](#)
4. [minimumRPMs-04 - Sunrise domain/launch application <create> command test](#)
5. [minimumRPMs-05 - Sunrise launch application <info> command test](#)
6. [minimumRPMs-06 - Trademark claims domain <create> command test](#)
7. [minimumRPMs-07 - LORDN submission test](#)

Test Suite Example (2/2)



▼ 4.9.3. Test plans

This suite is used by the following test plans:

1. [Main RSP Evaluation Test](#)

▼ 4.9.4. Resources

The following resources may be required to prepare for this test plan:

- *None specified.*

▼ 4.9.5. Errors

This test suite may produce the following errors:

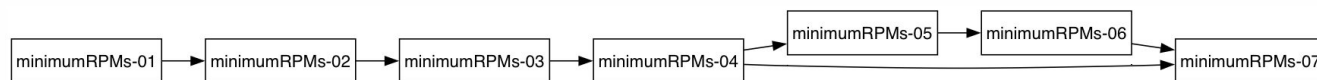
- [EPP_CONNECT_ERROR \(ERROR\)](#)
- [EPP_GENERIC_COMMAND_ERROR \(ERROR\)](#)
- [EPP_LOGIN_ERROR \(ERROR\)](#)
- [EPP_SCHEMA_VALIDATION_ERROR \(ERROR\)](#)
- [EPP_SUNRISE_AVAILABILITY_CHECK_INVALID_RESULT \(ERROR\)](#)

▼ 4.9.6. Input parameters

The test cases used by this suite require the following input parameters:

1. [epp.clid \(string\)](#)
2. [epp.hostModel \(string\)](#)
3. [epp.hostName \(string\)](#)
4. [epp.pwd \(string\)](#)
5. [general.minimalPublicDataSet \(boolean\)](#)
6. [general.sunriseModels \(string\)](#)

▼ 4.9.7. Sequence diagram



Test Case Example

6.114. minimumRPMs-01 - Sunrise domain <check> command test



6.114.1. Description

This test is used to confirm the conformance of the EPP server's implementation of the Availability Check Form, as described in Section 3.1.2 of RFC 8334.

For this test, it is assumed that the EPP server has been configured to use data (specifically the Sunrise List, as described in Section 5.4.2.1 of RFC 9361) from the TMCH test environment.

The client will connect to the EPP server using the provided credentials and will then perform a series of <check> commands, using the Launch extension to specify a value of `avail` for the `type` attribute of the <launch:check> element, and `sunrise` for the <launch:phase> element. It will then confirm that the server returns an appropriate response:

- The <domain:name> element for a syntactically valid domain which does not appear on the SURL **MUST** have an `avail` attribute of `0` or `false`.
- The <domain:name> element for a domain which is present on the SURL, and does not already exist, **MUST** have an `avail` attribute of `1` or `true`.

▼ 6.114.2. Errors

This test case may produce the following errors:

- [EPP_CONNECT_ERROR \(ERROR\)](#)
- [EPP_GENERIC_COMMAND_ERROR \(ERROR\)](#)
- [EPP_LOGIN_ERROR \(ERROR\)](#)
- [EPP_SCHEMA_VALIDATION_ERROR \(ERROR\)](#)
- [EPP_SUNRISE_AVAILABILITY_CHECK_INVALID_RESULT \(ERROR\)](#)

▼ 6.114.3. Input parameters

This test case requires the following input parameters:

- [epp.clid \(string\)](#)
- [epp.hostName \(string\)](#)
- [epp.pwd \(string\)](#)

RST Test Specifications

- Test Specifications:
<https://icann.github.io/rst-test-specs/rst-test-specs.html>
- GitHub repository: <https://github.com/icann/rst-test-specs>
- **Interested parties are welcome to report issues and submit pull requests!**

Version Control and Predictability

- API and test specifications are works-in-progress and subject to regular updates. Check the GitHub repositories to see changes and follow the repositories to be notified of updates.
- The API specification will be frozen before users have access to it and subsequent changes will be communicated ahead of deployment.
- The test specification will also be frozen in advance of the RSP Evaluation program. Each test request object will include the version number of the test plan that will be used. Test subjects can refer to the exact details of the test plan.

Engage with ICANN



Thank You and Questions

Visit us at icann.org

Email: email



[@icann](https://twitter.com/icann)



facebook.com/icannorg



youtube.com/icannnews



flickr.com/icann



linkedin/company/icann



soundcloud/icann



instagram.com/icannorg