

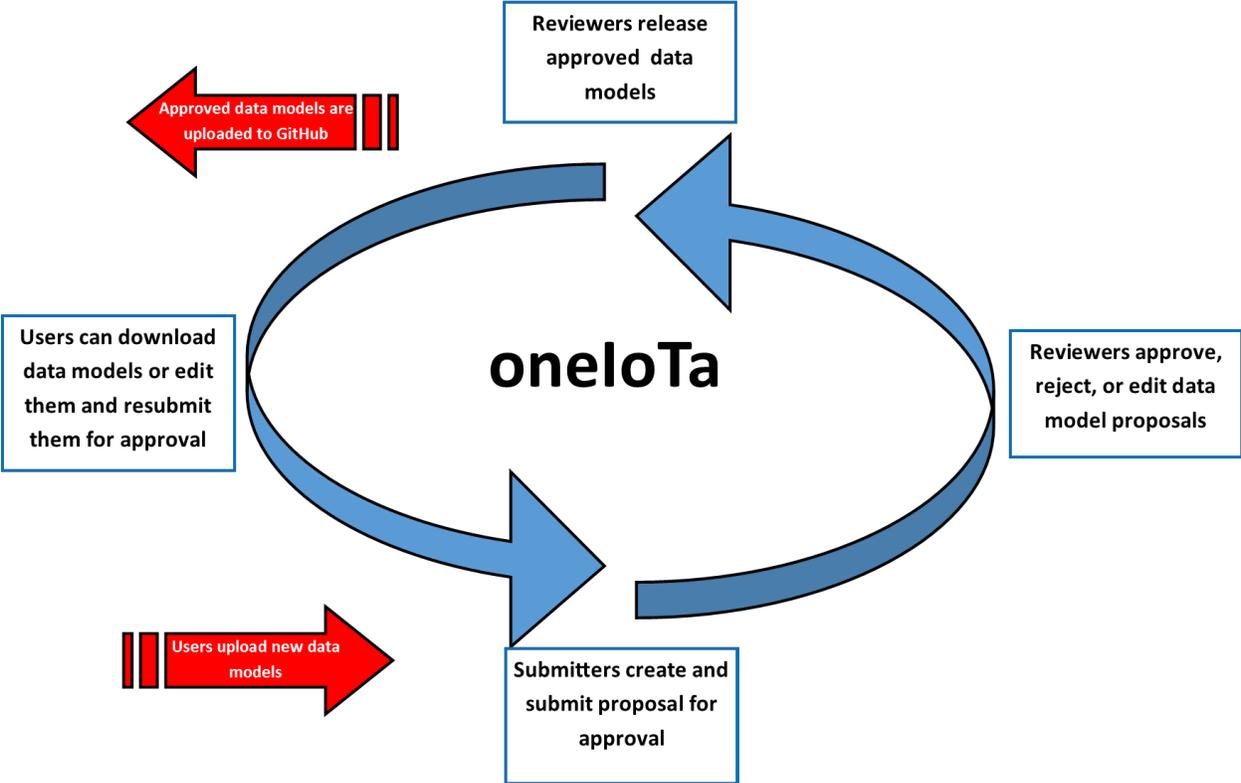
OPEN CONNECTIVITY FOUNDATION™

Open Connectivity Foundation oneloTa Tool

J. Clarke Stevens

Introduction

The Open Connectivity Foundation (OCF) was established with the goal of providing a common scalable standard for the Internet of Things with a certification tool that could ensure interoperability and an open-source implementation that could accelerate implementation time. In order to meet these objectives, a RESTful architecture was developed that would limit the system to just five (CRUDN) APIs and a constructive data model that would create complex devices and systems as collections of simpler devices and resources. It was also important to ensure interoperability with other ecosystems in order to expand the market and simplify the user experience.



Overview

oneloTa (one Internet of Things architecture) is essentially an Integrated Development Environment (IDE) that sits at the center of these requirements and delivers on this promise using OCF. It has integrated, syntax-colored, and validating text editors for the needed RAML (Restful API Modeling Language) interface definitions and the JSON (JavaScript Simple Object Notation) schemas that define each model. It supports a crowd-sourced process that allows anyone to submit model proposals, and a back-end approval process that allows multiple organizations to approve models for their particular ecosystems. Finally, it's backed by a git repository so anyone can get the models without ever using the tool if they choose to do so.

User Types

There are four types of users supported in the oneloTa tool. User rights are cumulative. The roles are listed here from most restrictive to least restrictive.

- **Viewer** – Viewers are allowed to browse the database of models and download them as they choose. Included files are discovered and linkable, so the device model tree can be easily traversed. The models list can be filtered by approval status and organization with the filter drop-down at the top right of the models screen
- **Submitter** – A submitter is allowed to create new model proposals and submit them to organizations for approval. A submitter must agree to the OCF contribution agreement.
- **Reviewer** – A reviewer has approval rights for a particular organization (e.g. OCF or UPnP). Once a proposal is submitted, a reviewer can change the status of a model from “pending” to “approved.” A reviewer can also tag a release. Typically, there will only be a handful of reviewers for a particular organization and they will determine their own process for reviewing models and tagging releases.
- **Admin** – An admin has the privileges of all other roles and is also able to grant any role to other users.

Process

All models have the potential to go through the process starting as a draft and ending as an approved model in a release. The process is described below.

Proposals

All models start as a proposal developed by a submitter. A submitter just clicks on the “Create Proposal” button to create a new model. A proposal can contain one or more models. The “Proposal Summary” is an optional description field that will eventually show up in the git repository if the proposal is approved. When a file name is added (all files must either have the file extension “.raml” or “.json”), the file is automatically created and an editing window is opened. The submitter can simply begin entering code, start with a default template, or paste in copied text. Files can also be batch-imported via an import window that allows for the selection of multiple files in a single import instance. If a file being added to or imported to a proposal already exists then oneloTa automatically creates a new version of that file.

Draft Status

Once created, the proposal is in “draft” status. Drafts are editable files that have not yet been submitted for approval by an organization. The editor automatically checks and highlights syntax and validates the file as you type. Errors are indicated above the edit window. All models must have a RAML interface and may include any number of JSON schema files. All the current OCF release 1.0.0 models are already entered into oneloTa, so they can be studied in order to understand the basic structure. Be sure to consult the organization for which the model is intended in order to understand naming conventions and other requirements.

Proposals can be shared so that groups of users can collaborate on models that are in draft mode. Just copy the listed link and share it with any intended collaborators. They must have the submitter role to edit or add files in the proposal. Only the proposal creator will be able to submit the proposal.

As models may evolve over time, new proposals can be created from existing proposals by clicking the pencil icon above the edit box of any approved model. It is up to the target organization to ensure backwards compatibility or make sure other requirements are met when new versions of a model are created. All versions of a model and the status of each are listed on the editor screen. Any version can be selected and viewed.

Submitting a Proposal

Once a proposal is deemed ready for submission, the submitter who created the proposal may submit it for approval. The submission must be targeted to a particular organization that can be selected from the drop-down on the editing screen. The proposal can also be discarded by clicking the “Discard” button. Once submitted, the proposal status will change to “pending” and will show up in the reviewer’s queue to be reviewed and potentially approved.

Approval Process

A reviewer will periodically review the pending models and determine if the proposal should be approved. If approved, no status indicator will show and the proposal will be pushed to the git repository. At that point, it can be pulled from the git repository or viewed in oneloTa, but it can no longer be edited. It will be tagged as part of the next release unless a new proposal is created and accepted before the release is tagged.

Releases

Releases are simply a noted point in time with an instructive label. The latest approved version of all accepted models at that point in time will be tagged with the label determined by the reviewer who tags the version. The tag will also be applied to the git repository at that point.

Git Repository

The oneloTa tool only commits to and tags a git repository master. oneloTa maintains a separate internal database for its functionality. The approving organization will determine

which repository it wants to use. In the case of OCF, the repository is open to all for pulling the models, but contributions back to the master must be done through oneloTa.

What's Next?

While oneloTa can do a lot to crowd-source the models required for the Internet of Things; this paper has only hinted at the interoperability features it supports. The is the subject of a separate paper. There are, however, a couple of features that are associated with oneloTa and the OCF data model approach. RAML and JSON files are machine readable and complete, and can therefore be directly used to create documentation and code. OCF data model specifications are currently created directly from the RAML and JSON files – not vice-versa. When a release is tagged, the models in the release are used to generate a file that can be used with mail merge in most word processing programs. In addition to documentation, the models can be used to generate code stubs in any programming language and on any platform (using Python scripts, for example). Generic user interfaces can also be generated. So, for example, a simple OCF RESTful interface core could be created on a Raspberry Pi. The code stubs for the CRUDN functions could then be automatically generated for any model in oneloTa. Another script could automatically generate a user interface with buttons for booleans, sliders for integers, etc. on an Android phone. A new prototype device could be created, implemented and fully functional in hardware in a matter of minutes.

The oneloTa tool is one step in the direction of a scalable and interoperable Internet of Things.