

THE ONEIOTA TOOL

J. CLARKE STEVENS

HOW SHOULD THE IOT WORK?

- Creation of new devices should scale at Internet speed
 - New interfaces should take minutes to develop, not months
- Creation of prototypes should be simple
 - A working prototype is better than a thousand pictures
- All ecosystems and devices should work together
 - The device maker shouldn't worry about being isolated by a technology choice

THE ONEIOTA TOOL AN IDE FOR IOT

- A crowd-sourced Integrated Development Environment (IDE) for the Internet of Things device models (oneloTa.org)
- RAML & JSON validated and syntax aware editors with shared editing
- Referenced files are clickable links
- Full versioning support

Indicates the proposal is in pending status awaiting approval by the organizations "Manager."

RAML files must reference at least one JSON scheme. Clicking this link will bring the corresponding file to the edit window

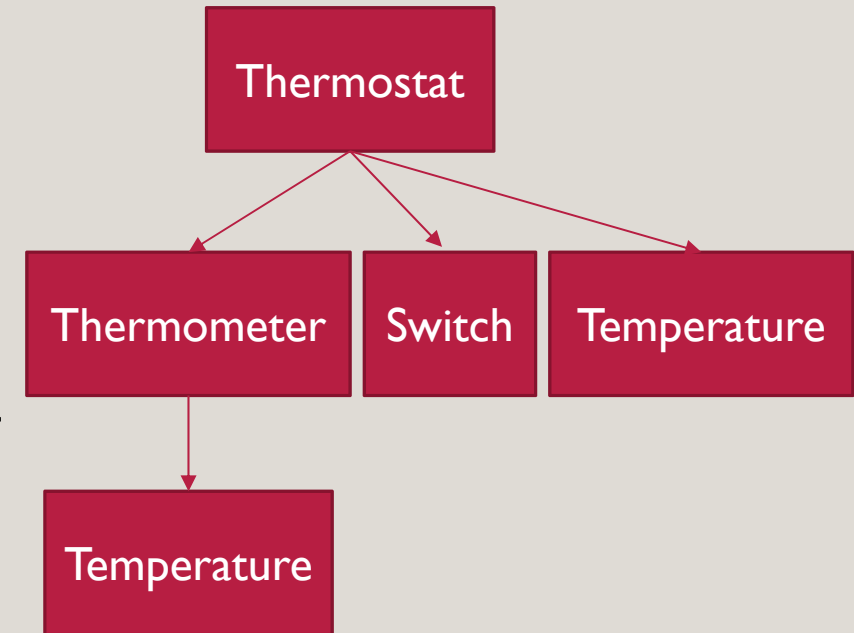
"Reviews" are able to Approve, Reject, or make edit to a file to make it compliant for their specific technology.

The screenshot displays the OneIoT tool interface. At the top, navigation links include 'All Models (63)', 'Proposals (1)', 'Releases (0)', 'Organizations (0)', and 'Users (17)'. The main content area is titled 'activityCount.raml'. On the left, a 'Versions' section shows a version from '20 Jan 2016' with a 'PENDING' status. Below this, a 'References' section contains a link to 'oic.r.sensor.activity.count.json'. A 'Proposal' section shows the organization 'UfNP' and a 'PENDING' status, with 'Approve' and 'Reject' buttons. A 'FILES' section lists various RAML files, including 'activityCount.raml'. On the right, the 'RAML' editor shows the following code:

```
1 #RAML 0.8
2 title: OICActivityCount
3 version: v1.0-20150727
4 schemas:
5   - Count: !include oic.r.sensor.activity.count.json
6   - traits:
7     - interface:
8       queryParameters:
9         if:
10           enum: ["oic.if.s","oic.if.a"]
11
12 /ActivityCountResURI:
13   description: |
14     This resource specifies an activity count.
15     The resource can be readwrite (oic.if.s interface) in
16     The resource can be readwrite (oic.if.a interface) in
17     The count property is an integer representing either
18
19   displayName: Activity Count
20   is: [ interface ]
21
22   GET:
23     description: |
24       Retrieves the current activity count.
25     responses:
26       200:
27         body:
28           application/json:
29             schema: Count
30             example: |
31               {
32                 "rt": "oic.r.sensor.activity.count",
33                 "id": "unique_example_id",
34                 "count": 2500
35               }
36
37   POST:
38     description: |
```

THE CONSTRUCTIVE DEVICE DATA MODEL (SCALES AT INTERNET SPEED)

- Choose a generic description strategy (e.g. RAML, JSON schemas)
- Start with physical properties (e.g. temperature, mass)
- All new devices are defined as collections of physical properties and previously defined devices (e.g. a thermostat is a collection of temperature, thermometer and switch)
- Abstract devices can also be defined (e.g. Clarke's house, upstairs bedrooms)



THE ONEIOTA TOOL PROCESS MANAGEMENT FOR APPROVAL

- Step-by-step process for creation of new proposed resources
- Integrated reuse of existing resources
- Interactive approval process for multiple organizations
- Git repo on the back end, so it fits into your regular development process

Indicates the proposal is in pending status awaiting approval by the organizations "Manager."

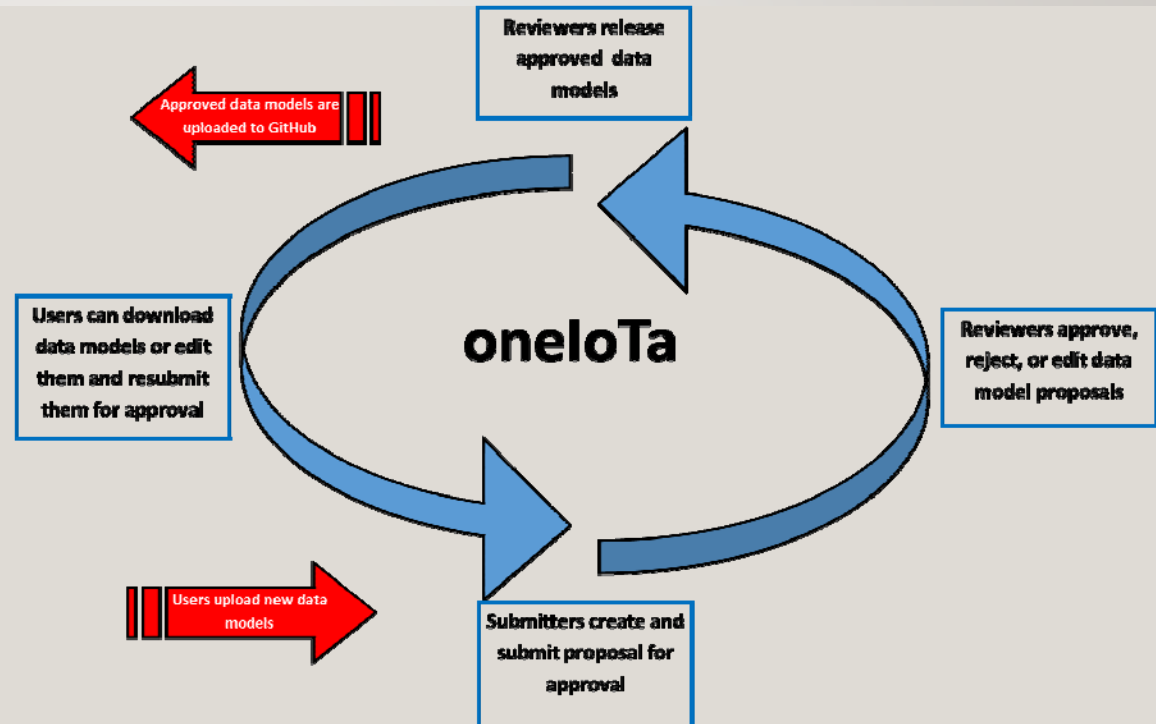
RAML files must reference at least one JSON schema. Clicking this link will bring the corresponding file to the edit window

"Reviews" are able to Approve, Reject, or make edit to a file to make it compliant for their specific technology.

The screenshot displays the ONEIOTA tool interface for managing proposals. The top navigation bar shows 'All Models (63)', 'Proposals (1)', 'Releases (0)', 'Organizations (0)', and 'Users (17)'. The main content area is titled 'activityCount.raml' and features a 'Versions' section with a '20 Jan 2016' version marked as 'PENDING'. Below this is a 'References' section with a link to 'oic.r.sensor.activity.count.json'. The 'Proposal' section shows 'ORGANIZATION: UFP' and 'INFO: OIC Draft: 01_SHT6', with 'Approve' and 'Reject' buttons. A 'FILES' list at the bottom includes 'activityCount.raml', 'atmosphericPressure.raml', 'audio.raml', 'autofocus.raml', 'automaticDocumentFeeder.raml', 'button.raml', and 'carbonDioxide.raml'. The right pane displays the RAML code for the resource, including a title 'OICActivityCount', version 'v1.0-20150727', and a schema for 'Count'.

PROPOSAL TO INTEGRATED MODEL FOR MULTIPLE ORGANIZATIONS

- Create a proposal
 - Start with existing models or a blank template (RAML & JSON schema)
- Submit the proposal to an organization
- The organization decides if it will be accepted (iterative process)



THE ONEIOTA TOOL INTEROPERABILITY IS BUILT-IN

- OCF as common data model facilitates interaction between different ecosystems
- Derived device data model syntax describes the relationship between data models of different ecosystems
- Everything referenced to the common data model is automatically interoperable with all other referenced models

Indicates the proposal is in pending status awaiting approval by the organizations "Manager."

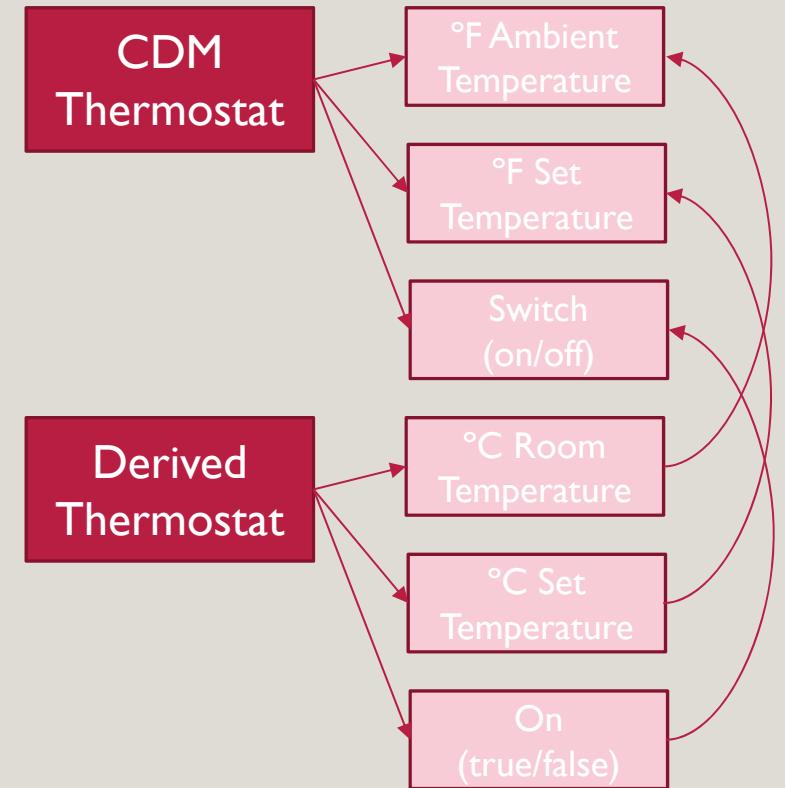
RAML files must reference at least one JSON schema. Clicking this link will bring the corresponding file to the edit window

"Reviews" are able to Approve, Reject, or make edit to a file to make it compliant for their specific technology.

The screenshot displays the ONEIOTA tool interface for a proposal titled 'activityCount.raml'. The top navigation bar shows 'All Models (63)', 'Proposals (1)', 'Releases (0)', 'Organizations (0)', and 'Users (17)'. The main content area is divided into several sections: 'Versions' (showing a version from '20 Jan 2016' with a 'PENDING' status), 'References' (listing a link to 'oic.r.sensor.activity.count.json'), 'Proposal' (showing the organization 'UfNP' and 'Approve' and 'Reject' buttons), and 'FILES' (listing several RAML files, including 'activityCount.raml'). On the right side, the RAML code is displayed, showing a schema for 'ActivityCount' with a 'Count' property. The interface is annotated with callouts: a yellow callout points to the 'PENDING' status, a blue callout points to the 'References' link, and a red callout points to the 'Approve' and 'Reject' buttons.

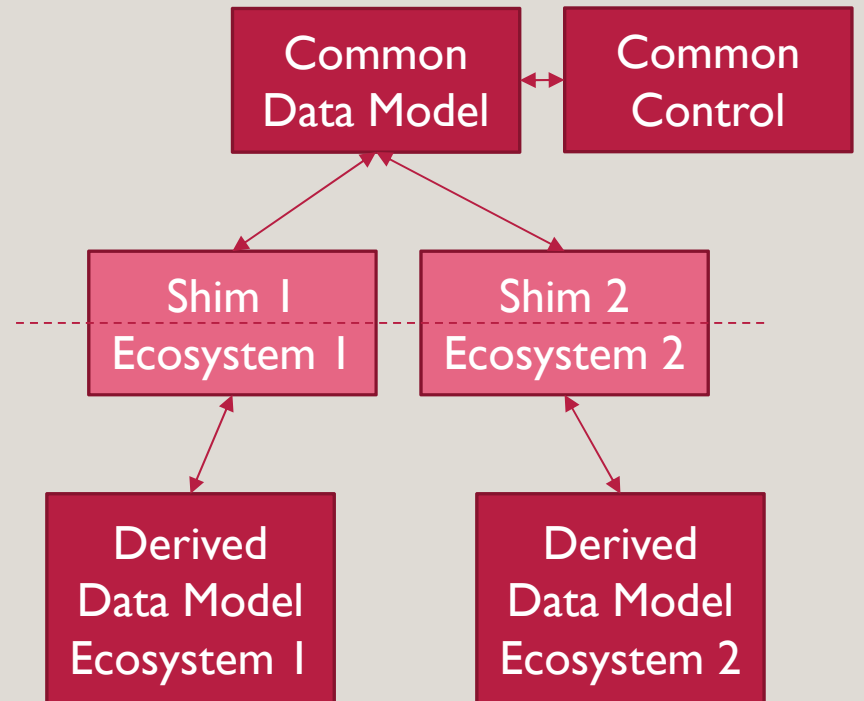
THE DERIVED DEVICE DATA MODEL (ALL ECOSYSTEMS WORK TOGETHER)

- ALL interoperable devices are defined exactly once in the common data model (CDM)
- Devices defined in other ecosystems (AllSeen, UPnP, etc.) are derived from devices in the common data model
- The definition of derived devices allows for differences in ecosystems (property names, variable types, range differences and conversions)



THE DERIVED DEVICE DATA MODEL (CONT.) (ALL ECOSYSTEMS WORK TOGETHER)

- In operation, a shim layer (code stubs automatically generated from the device data model) provides for conversion between ecosystems
- Since all ecosystems derive from the common data model, there are at most two conversions
- The conversion can happen in a gateway, in the cloud or in end devices



CALL TO ACTION

- Looking for developers who want to contribute new device data models
- Looking for advocates who want to develop code generators for specific target prototype platforms (e.g. Edison, Raspberry Pi, Arduino, etc.)
- Looking for advocates who want to develop code generators for specific target UI platforms (iOS, Android, Linux, etc.)
- Looking for developers who want to integrate data models for other platforms by defining derived models

Q&A



CONTACT INFORMATION

Clarke Stevens

jclarke.stevens@yahoo.com