Thing-to-Thing Research Group
Internet-Draft
Intended status: Experimental
Expires: September 3, 2018

K. Hartke
Universitaet Bremen TZI

March 2, 2018

Thing-to-Thing Data Hub
draft-hartke-t2trg-data-hub-01

Abstract

The Thing-to-Thing Data Hub is a RESTful, hypermedia-driven Web application that can be used in Thing-to-Thing communication to share data items such as thing descriptions, configurations, resource descriptions, or firmware updates at a central location.

Table of Contents

1. Introduction

In Thing-to-Thing communication, there is often a need to share data items of common interest at a central location. For example, the CoRE Resource Directory [I-D.ietf-core-resource-directory] aggregates descriptions of resources held on other servers, which enables things to easily discover these resources. Similarly, a W3C Web-of-Things Thing Description Repository [WOT] stores semantic metadata of things as well as functional descriptions of their interfaces, making this data available to Web dashboards, commissioning tools and other things.

As more and more thing-to-thing applications are implemented, it becomes increasingly important to be able to share not only resource and thing descriptions but also many other kinds of data, such as default configurations for new devices, service locations, firmware updates, or certificate revocation lists. The existing resource directories and thing description repositories are not a good fit for these kinds of data, as they're specialized to their use case and don't accept other kinds of data. And defining a new specialized application for each use case is not practical in the long term.

This document defines a simple "data hub" application, a RESTful Web application with a hypermedia API that is suitable for constrained environments and that generalizes the concept of a central repository for sharing any kinds of data. A data hub enables clients to share data items in any format and provides means for creating, reading, observing, updating, deleting and finding data items at a data hub server.

Data hubs are intended to be used primarily with CoAP [RFC7252].

Features:

- General

  The data hub generalizes the concept of a directory or repository to data items to any Internet media type. This means applications using the data hub aren't stuck forever with the same media types or limited to resource descriptions and thing descriptions.

- Searchable

  Clients can retrieve a subset of data items from a data hub based on item metadata.

- Observable

  Data items published to a data hub are exposed as resources. As such, they can be observed for changes [RFC7641]. This allows clients to stay informed of information that other clients update over time. As a result, the data hub functions similar to a CoAP Publish-Subscribe Broker [I-D.ietf-core-coap-pubsub], although this isn't its primary use case.

- Evolvable

  The key differentiator of the data hub compared to CoRE Resource Directory and CoAP Publish-Subscribe Broker is the evolvability -- the ability to respond effectively to the need for changes without negatively impacting existing and new clients. Data hubs enable fine-grained evolvability by driving all interactions by machine-readable hypermedia elements. Features can be added, changed or removed in a safe, backwards-compatible way simply by updating the data hub representation to expose appropriate links and forms.

1.1. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Data Model

The data model consists of two elements: the _data hub_ and a number of shared _data items_ (Figure 1).

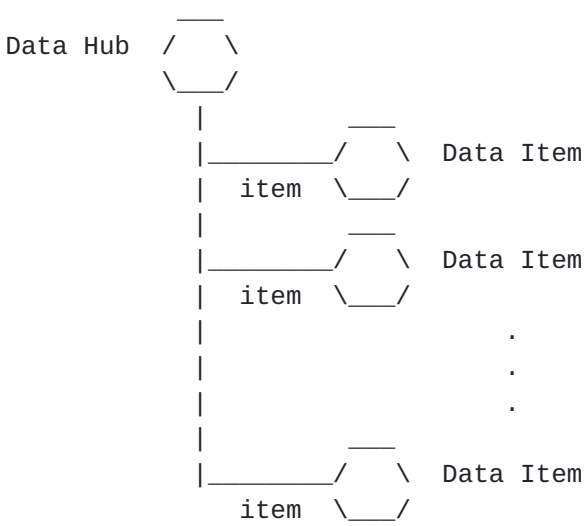

Figure 1: A Data Hub with a Number of Shared Data Items

Data Hub

A data hub resource is a collection of shared data items.

Data hub representations MUST be formatted in the "application/coral+cbor" or "text/coral" media type [I-D.hartke-t2trg-coral]. They primarily consist of links to the data items using the "item" link relation type [RFC6573]. To reduce the number of round-trips, they MAY also embed (complete or partial) representations of data items. Forms contained in the representation enable interactions with the hub and data items, as described in the following section. The representations MAY additionally contain other links and forms that are not described in this document, such as a link with the "alternate" link relation type that references an alternate representation of the data hub resource.

For a start, a data hub is defined to have a depth of only one level; i.e., all data item resources are organized directly under the top-level data hub resource. This could be extended to multiple levels in a future revision of this document.

Data Item

A data item is a member of the collection.

Data item representations MAY be formatted in any media type. However, a data hub instance MAY restrict the media types it accepts for publication. The form in the data hub representation for creating data items MUST list the acceptable media types in this case using form fields with name <urn:ietf:rfc:XXXX#accept>.

The representations of data items MAY link back to a data hub resource using the "collection" link relation type [RFC6573].

3. Interaction Model

The interaction model consists of eight possible interactions with a data hub: discovering and reading the data hub, and creating, reading, observing, updating, deleting, and finding shared data items in the data hub.

Discovering a Data Hub

For a start, this revision of the document assumes that clients are pre-configured with a link to a data hub.

Reading a Data Hub

A client can retrieve a representation of a data hub by following the pre-configured link. The representation of the data hub includes links to (and, optionally, representations of) the data items in the data hub. The data hub representation also includes forms for creating, updating, deleting, and finding data items.

Creating an Item

The representation of a data hub MAY contain a form with the <urn:ietf:rfc:XXXX#create> form relation type. Submitting this form with a representation in one of the acceptable media types creates a new data item in the data hub. The acceptable media types are indicated by <urn:ietf:rfc:XXXX#accept> form fields.

Data hubs implementing this specification MUST offer the POST method [RFC7252] in this form.

Reading an Item

A client can retrieve a representation of a data item by following a link with the <http://www.iana.org/assignments/relation/item> link relation type in the data hub representation.

Observing an Item

A client can observe a data item by following a link with the <http://www.iana.org/assignments/relation/item> link relation type in the data hub representation and observing the target resource as specified in RFC 7641 [RFC7641].

Updating an Item

For each data item in a data hub, the representation of the data hub MAY include a nested form with the <urn:ietf:rfc:XXXX#update> form relation type. Submitting this form updates the data item in the data hub to the submitted representation.

Data hubs implementing this specification MUST offer the PUT method [RFC7252] in this form.

Deleting an Item

For each data item in a data hub, the representation of the data hub MAY include a nested form with the <urn:ietf:rfc:XXXX#delete> form relation type. Submitting this form deletes the data item from the data hub.

Data hubs implementing this specification MUST offer the DELETE method [RFC7252] in this form.

Searching for Items

The representation of a data hub MAY contain a form with the <urn:ietf:rfc:XXXX#search> form relation type. This form can be used to find data items in the data hub. Submitting this form with a search query returns the subset of data items that match the query.

Data hubs implementing this specification MUST offer the FETCH method [RFC8132] in this form.

4. Security Considerations

TODO.

5. IANA Considerations

This document includes no request to IANA.

6. References

6.1. Normative References

[I-D.hartke-t2trg-coral]
Hartke, K., "The Constrained RESTful Application Language (CoRAL)", draft-hartke-t2trg-coral-04 (work in progress), October 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6573] Amundsen, M., "The Item and Collection Link Relations", RFC 6573, DOI 10.17487/RFC6573, April 2012, <https://www.rfc-editor.org/info/rfc6573>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/info/rfc7252>.

[RFC7641] Hartke, K., "Observing Resources in the Constrained Application Protocol (CoAP)", RFC 7641, DOI 10.17487/RFC7641, September 2015, <https://www.rfc-editor.org/info/rfc7641>.

[RFC8132] van der Stok, P., Bormann, C., and A. Sehgal, "PATCH and FETCH Methods for the Constrained Application Protocol (CoAP)", RFC 8132, DOI 10.17487/RFC8132, April 2017, <https://www.rfc-editor.org/info/rfc8132>.

6.2. Informative References

[I-D.ietf-core-coap-pubsub]
Koster, M., Keranen, A., and J. Jimenez, "Publish-Subscribe Broker for the Constrained Application Protocol (CoAP)", draft-ietf-core-coap-pubsub-03 (work in progress), February 2018.

[I-D.ietf-core-resource-directory]
Shelby, Z., Koster, M., Bormann, C., Stok, P., and C. Amsuess, "CoRE Resource Directory", draft-ietf-core-resource-directory-13 (work in progress), March 2018.

[RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690, DOI 10.17487/RFC6690, August 2012, <https://www.rfc-editor.org/info/rfc6690>.

[WOT] Kovatsch, M., Ed. and D. Peintner, Ed., "WoT Current Practices", February 2018, <http://w3c.github.io/wot/current-practices/wot-practices.html>.

Appendix A. Related Work

The data hub is an instance of the well-known collection pattern. As such, it might be used in places where a more specialized instance of the collection pattern is currently used, such as the CoAP Publish-Subscribe Broker [I-D.ietf-core-coap-pubsub] or the CoRE Resource Directory [I-D.ietf-core-resource-directory]. This section shows how these two applications might be implemented with a data hub (without trying to replicate all of their features in detail).

A.1. CoAP Publish-Subscribe

CoAP Publish-Subscribe [I-D.ietf-core-coap-pubsub] provides means for resource-constrained sensor and actuator nodes to publish and receive data without having to be available at the same time. The basic operation involves clients called "publishers" updating "topic" resources at a server called the "broker" and clients called "subscribers" observing these resources (Figure 2).

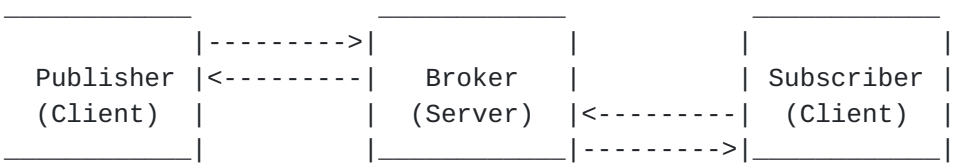

Figure 2: CoAP Publish-Subscribe

A broker might be implemented as a data hub by creating the topics as resources on the data hub server and linking to these from the data hub resource (Figure 3). Hypermedia controls in the data hub representation enable publishers to create, update, and delete topics as well as subscribers to read or observe these topics.

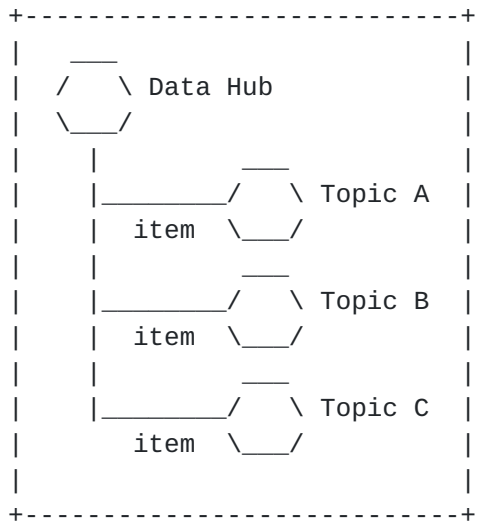

Figure 3: A Data Hub Acting as a Publish-Subscribe Broker

Interaction	Mapped to
DISCOVERY	Discovering a Data Hub / Reading a Data Hub / Searching for Items
CREATE	Creating an Item
PUBLISH	Updating an Item
SUBSCRIBE	Observing an Item
UNSUBSCRIBE	Observing an Item
READ	Reading an Item
REMOVE	Deleting an Item

Table 1: Mapping of Pub/Sub Interactions to Data Hub

A.2. CoRE Resource Directory

A CoRE Resource Directory [I-D.ietf-core-resource-directory] hosts descriptions of resources held on other servers, allowing lookups to be performed for those descriptions. The descriptions are encoded as links in CoRE Link Format [RFC6690] annotated with a variety of link attributes providing the type of and hints about the linked resources.

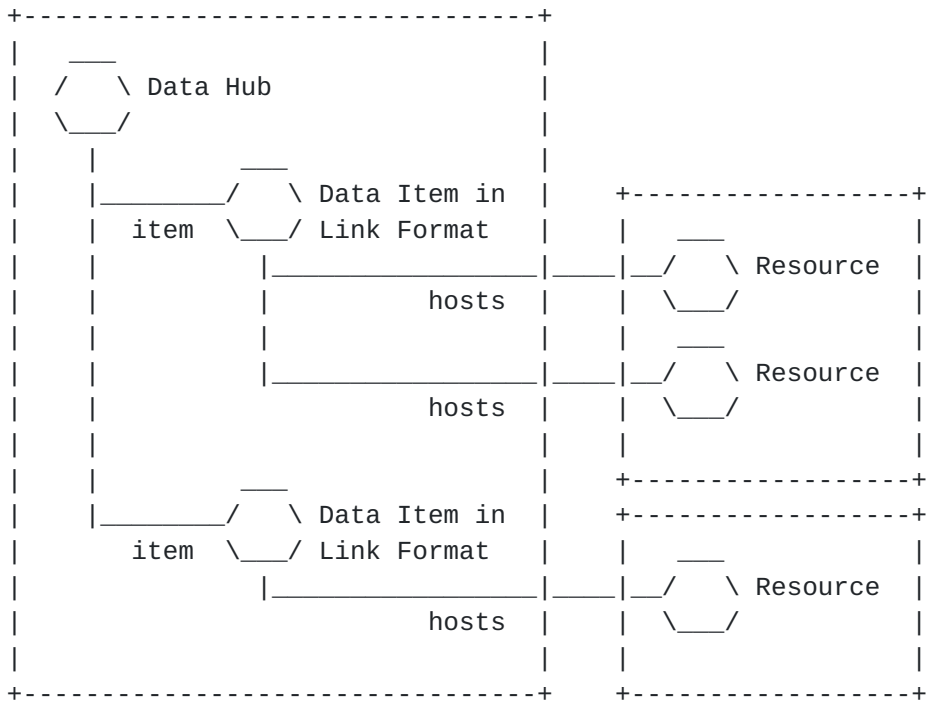

Figure 4: A Data Hub Storing Link Format Items

A data hub might be used to store these resource descriptions. Each resource description becomes a data item in the data hub (Figure 4). A specialized interface for querying the cumulative set of stored links might be provided separately.

Interaction	Mapped to
Discovery	Discovering a Data Hub
Registration	Creating an Item
Registration Update	-
Registration Removal	Deleting an Item
Read Endpoint Links	Reading an Item
Update Endpoint Links	Updating an Item

Table 2: Mapping of Resource Directory Interactions to Data Hub

Author's Address

Klaus Hartke
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

Phone: +49-421-218-63905
Email: hartke@tzi.org