

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 12 November 2025

P. M. Hallam-Baker  
ThresholdSecrets.com  
11 May 2025

JSDevice Device Description  
draft-hallambaker-jsdevice-00

Abstract

A JSON format for describing devices is described. Device descriptions provide identification information for the device and model number, images and video showing the device and its use, network configuration information and identifies associated consumable items, accessories and maintenance schedules.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 12 November 2025.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction . . . . .	3
---------------------------	---

1.1.	Onboarding . . . . .	4
1.1.1.	Preparation . . . . .	4
1.1.2.	Bootstrapping . . . . .	5
1.1.3.	Network Configuration . . . . .	7
1.2.	User Documentation . . . . .	9
1.3.	Operation . . . . .	11
2.	Definitions . . . . .	12
2.1.	Requirements Language . . . . .	12
2.2.	Defined Terms . . . . .	12
2.3.	Related Specifications . . . . .	12
2.4.	Implementation Status . . . . .	12
3.	Architecture . . . . .	12
3.1.	EARL . . . . .	13
3.2.	Updates . . . . .	13
4.	Data Binding . . . . .	13
4.1.	Data Type Notations . . . . .	13
4.2.	Common Data Types . . . . .	14
4.3.	Internationalization . . . . .	14
4.3.1.	Free-Form Text . . . . .	14
4.3.2.	URIs . . . . .	14
4.4.	Vendor Specific Extensions . . . . .	14
4.5.	Versioning . . . . .	15
4.5.1.	Version Format and Requirements . . . . .	15
4.5.2.	Current Version . . . . .	15
5.	Device Description . . . . .	15
5.1.	Common Properties . . . . .	15
5.1.1.	@type . . . . .	16
5.1.2.	uid . . . . .	16
5.1.3.	prodId . . . . .	16
5.1.4.	created . . . . .	16
5.1.5.	updated . . . . .	16
5.2.	Metadata . . . . .	16
5.2.1.	version . . . . .	16
5.2.2.	kind . . . . .	17
5.2.3.	language . . . . .	17
5.2.4.	localizations . . . . .	17
5.3.	General . . . . .	17
5.3.1.	deviceId . . . . .	17
5.3.2.	modelId . . . . .	18
5.3.3.	modelName . . . . .	18
5.3.4.	manufacturer . . . . .	18
5.3.5.	dateManufacture . . . . .	18
5.3.6.	endSupport . . . . .	18
5.3.7.	endLife . . . . .	18
5.4.	Physical . . . . .	18
5.4.1.	components . . . . .	18
5.5.	Media . . . . .	19
5.5.1.	images . . . . .	19

5.5.2. documentation . . . . .	20
5.6. Operations . . . . .	20
5.6.1. suppliers . . . . .	21
5.6.2. maintenance . . . . .	21
5.6.3. relatedItems . . . . .	21
5.7. Networking . . . . .	22
5.7.1. network . . . . .	22
5.7.2. cryptoKeys . . . . .	23
6. IANA Considerations . . . . .	24
7. Acknowledgements . . . . .	24
8. Normative References . . . . .	24
9. Informative References . . . . .	25
Author's Address . . . . .	25

## 1. Introduction

JSDevice is a JSON document format that provides a machine-readable description of a class of devices with the same characteristics or of a specific device, their use and associated consumable items, accessories and maintenance schedules. The format is intended as a counterpart to the JSContact [RFC9553] specification describing contact information for people and organizations.

The JSDevice data MAY be presented to the user by means of a QR code containing an Encrypted Authenticated Resource Locator (EARL) [draft-hallambaker-earl] printed on the device packaging, on the device itself or presented to the user on the device display. Alternatively the JSDevice description MAY be provided during the purchasing process.

A JSDevice description can present all the information necessary to configure and use the device including:

General information e.g. model identifier, device identifier, place of origin, data of manufacture, etc.

Physical properties The size and weight of the device.

Documentation Images and schematic diagrams depicting the device, manuals and tutorial videos.

Operations Required and advised maintenance schedule. Consumables, accessories and parts relevant to use of the machine.

Network Description of the networking services provided by and consumed by the device and the credentials used to authenticate the device to the network.

Description data MAY be contained within the JSDevice description itself or linked by means of a URI. If the URI is an EARL, the authentication context from the document context protecting it is preserved.

While the primary objective of the JSDevice specification is to enable seamless onboarding for network connected devices, JSDevice descriptions MAY be created for any type of device, including devices that have no networked functionality. This allows homeowners to compile a catalog of all their appliances supporting the specification, their consumable inks, filters, etc. and maintenance requirements.

### 1.1. Onboarding

The process of provisioning a device with the credentials and network configurations required for its initial function is known as 'onboarding'. The onboarding process is in turn divided into three phases:

**Preparation** The installer verifies that the device meets the intended purpose and that all the necessary support infrastructure is available.

**Bootstrapping** The device acquires an initial communication connection to the device administering the onboarding process.

**Configuration** The device acquires the credentials, network configurations and authorizations required to perform its function.

JSDevice is not an onboarding protocol, it is a means of specifying the bootstrapping and configuration protocols supported by the device and the network affordances required for the device to operate.

#### 1.1.1. Preparation

Preparation is an optional step in which the installer verifies that the device meets the intended purpose and that the necessary support infrastructure is available. This process would ideally take place before purchase.

To facilitate this use, a JSDevice description MAY describe a class of devices rather than a specific instance. Such a description does not contain a device unique identifier or device specific credentials.

```
{
  "@type": "Device"
  "uid": "NBEG-DJNO-DW3Q-RBKT-SXZ5-CAWH-MXUT"
  "prodId": "Configurator/1.0"
  "created": "2025-05-07T05:40:22Z"
  "updated":
  "version": "1.0"
  "kind": "model"
  "language": "en"
  "modelName": "Acme WebCam 4K"
  "manufacturer": "Acme Corp."
  "dateManufacture":
  "endSupport":
  "endLife":
  ...
}
```

### 1.1.2. Bootstrapping

Bootstrapping establishes an initial communication connection between the device being onboarded and the device administering the onboarding. A JSDevice description obtained from an EARL printed on the device itself, MAY specify one or more 'hailing channels' and contact credentials for establishing such a connection by means of a wired connection (e.g. Ethernet, USB-C), or a wireless connection such as WiFi or Near Field Communication.

Bootstrap mechanisms established in this fashion MAY make use of the Access Authenticator obtained from the EARL to authenticate the onboarding request, providing proof of physical possession of the device.

While the EARL mechanism is secure, retrieval of the ciphertext package referenced by the EARL depends on there being an external service available that can resolve the locator. A device MAY support delivery of the device description by the device itself using a hailing channel derived from the EARL by means of a cryptographic digest.

For example, a webcam has a QR code containing a JSDevice QR code printed on it:

```
jsdevice://example.com/el7i-i65v-c2ht-2zfe-pkeh-nocr-dd73
```

The corresponding JSDevice description describes the device itself. The document identifier is different to the one specified in the model description, the kind is device and there is a date of manufacture.

```
{
  "@type": "Device"
  "uid": "NDXT-KQ6W-FXFM-5D2A-4AXY-LTAM-OAFR"
  "prodId": "Configurator/1.0"
  "created": "2025-05-11T05:40:22Z"
  "updated": "2025-05-11T05:40:22Z"
  "version": "1.0"
  "kind": "device"
  "language": "en"
  "modelName": "Acme WebCam 4K"
  "manufacturer": "Acme Corp."
  "dateManufacture": "2025-05-11T05:40:22Z"
  "endSupport": "2030-05-11T05:40:22Z"
  "endLife": "2035-05-11T05:40:22Z"
  ...
}
```

The network property of the description provides a declarative specification of each of the network protocols supported by the device. Bootstrap protocols are identifier by the kind property value bootstrap.

```
{
  ...
  "network": {
    "boot1": {
      "kind": "bootstrap",
      "identifier": "mmm-connect",
      "ports": [80
    ],
    "keys": {
      "id": "udf:MCBG-ZSI4-XSWX-UNEE-GU6X-YTBZ-XP5" }}}
  ...
}
```

Devices MAY specify multiple bootstrap protocols or no bootstrap protocol leaving this to the user.

In this example, the Mathematical Mesh Device Connection protocol mmm-connect is used to establish the initial connection. This protocol cryptographically binds the device to a specific Mesh Account and provides the DNS address of the current provider servicing that account. Since the keys used to authenticate the device to the service are passed in-band during the connection protocol, it is sufficient to specify the fingerprint of the device profile through which the device keys are authenticated.

### 1.1.3. Network Configuration

Network configuration is the process of assigning network addresses, names and providing network credentials.

A core goal of the JSDevice description is to enable automation of the entire process of network configuration by offloading the task of configuring the network IP assignment, DNS, firewall, NAT and PKI to a local service.

While the network environment in which the device is to function MAY be of Byzantine complexity with multiple layers of firewalls, NAT and VPN, the device itself does not need to know anything more than the interface between itself and the rest of the network.

For example, regardless of the network in which Alice's Webcam is to be deployed, all the device needs to know is:

- \* The IP address of the device on the local network
- \* The DNS name by which the device is discovered
- \* The PKI credentials to be used to authenticate the device to clients.

In this example, the device employs a two stage configuration approach

In the first stage, the device obtains a temporary address on the local network via DHCP.

The device uses the IP address obtained in the first stage to connect to the 'anything' service and request that the necessary configurations be made to the network infrastructure and the corresponding IP, DNS and PKI data sent to the device

These protocols are specified as follows:

```
{
  ...
  "network": {
    "discl": {
      "kind": "discovery",
      "identifier": "dhcp",
      "endpoints": ["ip"]
    },
    "disc2": {
      "kind": "discovery",
      "identifier": "anything",
      "endpoints": ["ip",
        "dns",
        "webpki"]
    }
  }
  ...
}
```

At the end of the network configuration process, Alice and Bob to access the device as

<https://camera01.alice.example.com/>

The webcam supports viewing of the output via HTTP and MQQ and has a motorized mount that can be controlled through the proprietary acme\_cam\_control Web Service:



```
{
  ...
  "network": {
    "motor1": {
      "kind": "service",
      "identifier": "acme_cam_control",
      "ports": [443
      ],
      "endpoints": ["dns",
        "webpki"
      ],
      "permissions": ["admin"
      ]},
    "web1": {
      "kind": "service",
      "identifier": "https",
      "ports": [443
      ],
      "endpoints": ["dns",
        "webpki"
      ],
      "permissions": ["read"
      ]},
    "moq1": {
      "kind": "service",
      "identifier": "moq",
      "endpoints": ["dns",
        "webpki"
      ],
      "permissions": ["read"
      ]}}
  ...
}
```

## 1.2. User Documentation

JSDevice descriptions MAY include links to images of the physical device itself and schematics showing its use and to user documentation of various forms.

For example, the Webcam has photographs of the front and rear of a typical device and a schematic. The documentation includes a guide to unpacking it, a quick start guide and a user manual

```
{
  ...
  "images": [{
    "@type": "Media",
    "kind": "front",
    "uri": "https://acme.example.net/media/webcam4k.front.png",
    "mediaType": "image/png",
    "label": "Front View"},
    {
      "@type": "Media",
      "kind": "rear",
      "uri": "https://acme.example.net/media/webcam4k.rear.png",
      "mediaType": "image/png",
      "label": "Rear View"},
    {
      "@type": "Media",
      "kind": "schematic",
      "uri": "https://acme.example.net/media/webcam4k.schematic.svg",
      "mediaType": "image/svg",
      "label": "Rear View"}
  ]
  "documentation": [{
    "@type": "Media",
    "kind": "quickstart",
    "uri": "https://acme.example.net/media/webcam4k.quickstart.pdf",
    "mediaType": "application/pdf",
    "label": "Quick Start Guide"},
    {
      "@type": "Media",
      "kind": "unpacking",
      "uri": "https://acme.example.net/media/webcam4k.unboxing.pdf",
      "mediaType": "application/pdf",
      "label": "Rear View"},
    {
      "@type": "Media",
      "kind": "manual",
      "uri": "https://acme.example.net/media/webcam4k.manual.pdf",
      "mediaType": "application/pdf",
      "label": "Manual"}
  ]
  ...
}
```

### 1.3. Operation

JSDevice descriptions MAY include information relevant to the operation of the device itself including maintenance schedules, related parts, accessories and consumables, and suppliers of additional devices.

For example, the motorized mount requires that the camera mount be lubricated and checked every 24 months through a process described in the user manual.

```
{
  ...
  "suppliers": [{
    "@type": "Supplier",
    "uri": "https://store.acme.example.net/webcam4k",
    "mediaType": "text/html",
    "label": "Acme Store"}
  ]
  "maintenance": {
    "maint1": {
      "@type": "Maintenance",
      "uri": "https://acme.example.net/media/webcam4k.manual.pdf#lubrication",
      "label": "Lubricate mount",
      "recurring": true,
      "months": 24}}
  "relatedItems": {
    "part1": {
      "@type": "RelatedItem",
      "kind": "part",
      "uri": "https://acme.example.net/media/bearings/24U.jsdevice",
      "mediaType": "application/jsdevice",
      "label": "Mount bearing"},
    "accessory1": {
      "@type": "RelatedItem",
      "kind": "accessory",
      "uri": "https://acme.example.net/media/lenses/telephoto200.jsdevice",
      "mediaType": "application/jsdevice",
      "label": "Lens Kit"}}
  ...
}
```

## 2. Definitions

This section presents the related specifications and standard, the terms that are used as terms of art within the documents and the terms used as requirements language.

### 2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

### 2.2. Defined Terms

### 2.3. Related Specifications

JSContact: A JSON Representation of Contact Data [RFC9553]. Describes the format used for contact data interchange.

JSContact Extensions [draft-hallambaker-jscontact]. Extensions to the JSContact format to improve support for use of cryptographic credentials and allow for authenticated updates.

Encrypted Authenticated Resource Locator [draft-hallambaker-earl]. Describes a URI form that provides means of retrieving, decrypting and authenticating an associated ciphertext.

### 2.4. Implementation Status

Reference code under the MIT Open Source license has been developed to demonstrate all the features described in this document.

## 3. Architecture

The JSDevice format is limited to describing a device or class of devices. The means by which the device description is obtained by relying applications is outside the scope of this specification but MAY be relevant to the authenticity and currency of the description obtained.

Different delivery mechanisms present different authenticity and currency properties and thus the controls relevant to achieving these concerns differ by context.

A device description obtained by scanning a QR code on the device itself has a very high probability of being authentic and accurate with respect to the properties of the device scanned but a lower

probability of being current. A device description provided by a vendor during the acquisition process is more vulnerable to forgery but likely to reflect important updates if genuine.

It is anticipated that a JSDevice description will normally be delivered wrapped in an assertion structure which at minimum provides integrity controls. The decision to use an assertion format and its scope is left as an implementation decision.

### 3.1. EARL

The Encrypted Authenticated Resource Locator [draft-hallambaker-earl] is a resource location scheme that provides a basic cryptographic envelope that provides confidentiality and integrity controls.

The envelope payload and metadata MAY be signed providing an assertion structure and the keys used for signing MAY be supported by credentials issued by a Trusted Third Party.

### 3.2. Updates

Device descriptions MAY support updates via the mechanism presented in JSContact Extensions [draft-hallambaker-jscontact].

[Alternatively, the update mechanism could be specified in the protected header of the Assertion structure.]

## 4. Data Binding

The data binding conventions established in [RFC9553] are applied. For convenience, these are summarized here.

### 4.1. Data Type Notations

The underlying format for JSContact is JSON, so its data types also build on JSON values. The terms "object" and "array" as well as the four primitive types ("strings", "numbers", "booleans", and "null") are to be interpreted as described in Section 1 of [RFC8259]. All JSContact data MUST be valid according to the constraints given in I-JSON [RFC7493]. Unless otherwise noted, all member names in JSON objects and all string values are case-sensitive. Within the context of JSON objects, the term "key" is synonymous with "member name" as defined in Section 1 of [RFC8259].

## 4.2. Common Data Types

The common data type Resource specified in section 1.4.4 of [RFC9553] and the type CryptoKey specified in section 2.6.1 of [RFC9553] are used.

## 4.3. Internationalization

JSDevice aims to be used for devices in international use. Notably, text values such as product names, manuals, etc. are likely to cover a wide range of languages and cultures.

### 4.3.1. Free-Form Text

Properties having free-form text values MAY contain any valid sequence of Unicode characters encoded as a JSON string. Such values can contain unidirectional left-to-right and right-to-left text, as well as bidirectional text using Unicode Directional Formatting Characters as described in Section 2 of [TBS]. Implementations setting bidirectional text MUST make sure that each property value complies with the requirements of the Unicode Bidirectional Algorithm. Implementations MUST NOT assume that text values of adjacent properties are processed or displayed as a combined string; for example, the values of a given name component and a surname component may or may not be rendered together.

### 4.3.2. URIs

Several properties require their string value to be a URI as defined in [RFC3986]. Implementations MUST make sure to use proper percent-encoding for URIs that cannot be represented using unreserved URI characters. Section 3.1 of [RFC3987] defines how to convert Internationalized Resource Identifiers to URIs. JSContact makes no recommendation on how to display URIs, but the WHATWG URL Living Standard (see "Internationalization and special characters" (Section 4.8.3) of [TBS]) provides guidance for URLs found in the context of a web browser.

## 4.4. Vendor Specific Extensions

Vendors may extend properties and values for experimentation or to store contacts data that is only useful for a single service or application. Such extensions are not meant for interoperation. If, instead, interoperation is desired, vendors are strongly encouraged to define and register new properties, types, and values at IANA.

#### 4.5. Versioning

Every instance of a JSDevice description indicates which JSDevice version its IANA-registered properties and values are based on. The version is indicated both in the property within the description and in the version parameter of the JSDevice media type. All IANA-registered elements indicate the version at which they were introduced or obsoleted.

A JSDevice version consists of a major and minor version.

Differing major version values indicate substantial differences in JSDevice semantics and format. Implementations **MUST** be prepared for property definitions and other JSDevice elements that differ in a backwards-incompatible manner.

Differing minor version values indicate additions that enrich JSDevice data but do not introduce backwards-incompatible changes. Typically, these are new property enum values or properties with a narrow semantic scope. A new minor version **MUST NOT** require implementations to change their processing of JSDevice data. Changing the major version number resets the minor version number to zero.

##### 4.5.1. Version Format and Requirements

A version value starts with the numeric major version, followed by the FULL STOP character (U+002E), followed by the numeric minor version. Later versions are numerically higher than former versions, with the major version being more significant than the minor version. A version value is produced by the following ABNF:

```
jsversion = 1*DIGIT "." 1*DIGIT
```

Figure: The ABNF for JSDevice Version Values

##### 4.5.2. Current Version

This specification registers JSDevice version value "1.0".

#### 5. Device Description

##### 5.1. Common Properties

The following properties are common to JScontact, JSCalendar and JSDevice and used in the same fashion in each specification.

#### 5.1.1. @type

"@type": String (mandatory) This specifies the type that this object represents. The allowed value differs by object type and is defined in Sections 2.1, 2.2, and 2.3.

"@type": "Device"

#### 5.1.2. uid

"uid": String (mandatory) This is a globally unique identifier used to associate objects representing the same item. Updates to the document describing the same item MUST have the same UID.

"uid": "NDXT-KQ6W-FXFM-5D2A-4AXY-LTAM-OAFR"

#### 5.1.3. prodId

"prodId": String (optional) An identifier for the product that last updated the JSCalendar object. This should be set whenever the data in the object is modified (i.e., whenever the updated property is set).

"prodId": "Configurator/1.0"

#### 5.1.4. created

"created": UTCDateTime (optional) The date and time this object was initially created.

"created": "2025-05-11T05:40:22Z"

#### 5.1.5. updated

"updated": UTCDateTime (optional) The date and time this object was last modified.

"updated": "2025-05-11T05:40:22Z"

### 5.2. Metadata

Information related to the description itself

#### 5.2.1. version

"version": String (mandatory) The JSDevice version of this description. The value MUST be one of the IANA-registered JSDevice Version values for the version property.



```
"version": "1.0"
```

#### 5.2.2. kind

"kind": String (optional) The kind of the entity described

```
"kind": "device"
```

#### 5.2.3. language

"language": String (optional) The language tag, as defined in [RFC5646], that best describes the language used for text in the description, optionally including additional information such as the script. Note that values MAY be localized in the localizations property.

```
"language": "en"
```

#### 5.2.4. localizations

"localizations": String[JsDevice] (optional) The property values localized to languages other than the main language (Section 2.1.5) of the Card. Localizations provide language-specific alternatives for existing property values and SHOULD NOT add new properties. The keys in the localizations property value are language tags [RFC5646]; the values are of type PatchObject and localize the Card in that language tag. The paths in the PatchObject are relative to the Card that includes the localizations property. A patch MUST NOT target the localizations property.

The JsDevice object is defined in [TBS].

```
"localizations": {  
  "cy": {  
    "@type": "Device",  
    "modelName": "Acme GweGamera 4K"}}}
```

### 5.3. General

General information related to the device and model.

#### 5.3.1. deviceId

"deviceId": String (optional) A URI that uniquely identifies the device.

```
"deviceId": "NDXT-KQ6W-FXFM-5D2A-4AXY-LTAM-OAFR"
```

#### 5.3.2. modelId

"modelId": String (optional) A URI that uniquely identifies the device model.

"modelId": "NBEG-DJNO-DW3Q-RBKT-SXZ5-CAWH-MXUT"

#### 5.3.3. modelName

"modelName": String (optional) Human readable model name.

"modelName": "Acme WebCam 4K"

#### 5.3.4. manufacturer

"manufacturer": String (optional) Device manufacturer name.

"manufacturer": "Acme Corp."

#### 5.3.5. dateManufacture

"dateManufacture": UTCDateTime (optional) Date of manufacture.

"dateManufacture": "2025-05-11T05:40:22Z"

#### 5.3.6. endSupport

"endSupport": UTCDateTime (optional) Date at which support for the device is scheduled to end.

"endSupport": "2030-05-11T05:40:22Z"

#### 5.3.7. endLife

"endLife": UTCDateTime (optional) Date at which it is advised the device be taken out of service.

"endLife": "2035-05-11T05:40:22Z"

### 5.4. Physical

Information related to the physical properties of the device.

#### 5.4.1. components

"components": String[Component] (optional) The physical components making up the device and their dimensions.

A Component object has all the properties of the Resource data type, with the following additional definition:

"dimensions": String[Dimensions] (optional) The dimension specifications

A Dimensions object has the following properties.

"kind": String (optional) The type of dimensions specified, 'typical', 'maximum', 'shipping'

"weight": Number (optional) The weight of the component in kilograms

"width": Number (optional) The width of the component in meters

"depth": Number (optional) The depth of the component in meters

"height": Number (optional) The height of the component in meters

"temperatureMin": Number (optional) The minimum temperature for the device

"temperatureMax": Number (optional) The maximum temperature for the device

```
"components": {
  "main": {
    "@type": "Component",
    "dimensions": {
      "typical": {
        "kind": "typical",
        "weight": 0.05,
        "width": 0.1,
        "depth": 0.1,
        "height": 0.1}}}}}
```

## 5.5. Media

Media related to the device and its operation.

### 5.5.1. images

"images": Media[] (optional) Photographs and schematics of the device or model.

The Media object is defined in [TBS].

```
"images": [{
  "@type": "Media",
  "kind": "front",
  "uri": "https://acme.example.net/media/webcam4k.front.png",
  "mediaType": "image/png",
  "label": "Front View"},
{
  "@type": "Media",
  "kind": "rear",
  "uri": "https://acme.example.net/media/webcam4k.rear.png",
  "mediaType": "image/png",
  "label": "Rear View"},
{
  "@type": "Media",
  "kind": "schematic",
  "uri": "https://acme.example.net/media/webcam4k.schematic.svg",
  "mediaType": "image/svg",
  "label": "Rear View"}]
```

#### 5.5.2. documentation

"documentation": Media[] (optional) Manuals describing the device.

The Media object is defined in [TBS].

```
"documentation": [{
  "@type": "Media",
  "kind": "quickstart",
  "uri": "https://acme.example.net/media/webcam4k.quickstart.pdf",
  "mediaType": "application/pdf",
  "label": "Quick Start Guide"},
{
  "@type": "Media",
  "kind": "unpacking",
  "uri": "https://acme.example.net/media/webcam4k.unboxing.pdf",
  "mediaType": "application/pdf",
  "label": "Rear View"},
{
  "@type": "Media",
  "kind": "manual",
  "uri": "https://acme.example.net/media/webcam4k.manual.pdf",
  "mediaType": "application/pdf",
  "label": "Manual"}]
```

#### 5.6. Operations

Information related to operation of the device.

### 5.6.1. suppliers

"suppliers": Supplier[] (optional) Suppliers for the device and related accessories.

A Supplier object has all the properties of the Resource data type.

```
"suppliers": [{
  "@type": "Supplier",
  "uri": "https://store.acme.example.net/webcam4k",
  "mediaType": "text/html",
  "label": "Acme Store"}]
```

### 5.6.2. maintenance

"maintenance": String[Maintenance] (optional) List of maintenance events associated with the device.

A Maintenance object has all the properties of the Resource data type, with the following additional definitions:

"recurring": Boolean (optional) If true (default), the maintenance event recurs as specified by the days, months and years properties. If false, the maintenance event is a one-time operation occurring the specified interval after installation.

"days": Number (optional) Interval days.

"months": Number (optional) Interval months. If specified, the days property is ignored.

```
"maintenance": {
  "maint1": {
    "@type": "Maintenance",
    "uri": "https://acme.example.net/media/webcam4k.manual.pdf#lubrication",
    "label": "Lubricate mount",
    "recurring": true,
    "months": 24}}
```

### 5.6.3. relatedItems

"relatedItems": String[RelatedItem] (optional) Accessories, parts and consumables related to the device. For example, ink cartridges, parts likely to wear etc.

A RelatedItem object has all the properties of the Resource data type, with the following additional definitions:

"modelId": String[] (optional) A list of URIs specifying model identifiers. These MAY include related variations of the same item. For example, different sizes of the same color ink.

"suppliers": Supplier[] (optional) A list of URIs linking to suppliers for the consumable.

A Supplier object has all the properties of the Resource data type.

```
"relatedItems": {
  "part1": {
    "@type": "RelatedItem",
    "kind": "part",
    "uri": "https://acme.example.net/media/bearings/24U.jsdevice",
    "mediaType": "application/jsdevice",
    "label": "Mount bearing"},
  "accessory1": {
    "@type": "RelatedItem",
    "kind": "accessory",
    "uri": "https://acme.example.net/media/lenses/telephoto200.jsdevice",
    "mediaType": "application/jsdevice",
    "label": "Lens Kit"}}
```

## 5.7. Networking

Information related to device networking (if supported)

### 5.7.1. network

"network": String[Network] (optional) The network protocol connections supported.

A Network object has the following properties.

"kind": String (optional) The network connection kind

"address": String[] (optional) The Internet Protocol Address(es)

"identifier": String (optional) The IANA protocol identifier

"ports": Number[] (optional) The default IP ports.

"endpoints": String[] (optional) "keys": String[String] (optional)  
The identifiers of the set of device keys that MAY be used in combination with this protocol

"permissions": String[] (optional) Permission identifiers used to assign access rights to control of the device.

```
"network": {
  "boot1": {
    "kind": "bootstrap",
    "identifier": "mmm-connect",
    "ports": [80],
    "keys": {
      "id": "udf:MCBG-ZSI4-XSWX-UNEE-GU6X-YTBZ-XP5" }},
  "discl": {
    "kind": "discovery",
    "identifier": "dhcp",
    "endpoints": ["ip"]},
  "disc2": {
    "kind": "discovery",
    "identifier": "anything",
    "endpoints": ["ip",
      "dns",
      "webpki"]},
  "motor1": {
    "kind": "service",
    "identifier": "acme_cam_control",
    "ports": [443],
    "endpoints": ["dns",
      "webpki"],
    "permissions": ["admin"]},
  "web1": {
    "kind": "service",
    "identifier": "https",
    "ports": [443],
    "endpoints": ["dns",
      "webpki"],
    "permissions": ["read"]},
  "moq1": {
    "kind": "service",
    "identifier": "moq",
    "endpoints": ["dns",
      "webpki"],
    "permissions": ["read"]}}
```

#### 5.7.2. cryptoKeys

"cryptoKeys": String[CryptoKey] (optional) The cryptographic resources such as public keys and certificates associated with the device

The CryptoKey object is defined in JSContact and JSContact Cryptographic Extensions.

The CryptoKey object is defined in [TBS].

```
"cryptoKeys": {}
```

## 6. IANA Considerations

This document does not specify any actions for IANA (yet).

## 7. Acknowledgements

Michael Richardson, Alan DeKok,

Robert Stepanek

## 8. Normative References

[draft-hallambaker-earl]

Hallam-Baker, P., "Encrypted Authenticated Resource Locator", Work in Progress, Internet-Draft, draft-hallambaker-earl-00, 10 April 2025, <<https://datatracker.ietf.org/doc/html/draft-hallambaker-earl-00>>.

[draft-hallambaker-jscontact]

Hallam-Baker, P., "JSContact Cryptographic Key Extensions", Work in Progress, Internet-Draft, draft-hallambaker-jscontact-04, 10 May 2025, <<https://datatracker.ietf.org/doc/html/draft-hallambaker-jscontact-04>>.

[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/rfc/rfc2119>>.

[RFC7493] Bray, T., "The I-JSON Message Format", RFC 7493, DOI 10.17487/RFC7493, March 2015, <<https://www.rfc-editor.org/rfc/rfc7493>>.

[RFC8259] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/rfc/rfc8259>>.



[RFC9553] Stepanek, R. and M. Loffredo, "JSContact: A JSON Representation of Contact Data", RFC 9553, DOI 10.17487/RFC9553, May 2024, <<https://www.rfc-editor.org/rfc/rfc9553>>.

## 9. Informative References

[RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", RFC 3987, DOI 10.17487/RFC3987, January 2005, <<https://www.rfc-editor.org/rfc/rfc3987>>.

## Author's Address

Phillip Hallam-Baker  
ThresholdSecrets.com  
Email: [phill@hallambaker.com](mailto:phill@hallambaker.com)