Distributed Security Risks and Opportunities in the W3C Web of Things

Michael McCool (presenting) and Elena Reshetova

NDSS DISS 2018
February 18, 2018
San Diego

https://github.com/mmccool/ndss-wot-sec

Outline

Goals

W3C Web of Things

Risks and Opportunities

1. Local Links
2. Vulnerability Analysis
3. Endpoint Adaptation
4. Secure Discovery
5. Security Metadata

Summary and Conclusions

Goals

Why this paper?

- Necessary to perform security review of standards under development
- Paper lists a number of **problems** with the proposed W3C Web of Things standard under development that need to be addressed
  - Some issues are generalizable to other Web/IoT systems with metadata: DDS, OCF, IPSO, OneM2M, OpenAPI, etc.
- The paper does **not**, generally, propose **solutions**

Desired outcome:

Discussion, collaboration, and research to find solutions to these problems.

Web of Things

Working Group within W3C chartered December 2016

- Based on ongoing work in an Interest Group by the same name

Target date of December 2018 to deliver specifications for

- **Thing Description:** metadata for describing Things
- **Scripting API:** standardized mechanism to consume and expose Thing Descriptions and program the behavior of Things
- **Protocol Bindings:** mappings of WoT architecture to various concrete protocols: HTTP, CoAP, MQTT, etc.
  - Mapping from abstract Property/Event/Actions CRUDN and/or Pub/Sub

Web of Things: Resources and Links

W3C: World Wide Web Consortium: https://www.w3.org

Web of Things Interest Group: https://www.w3.org/WoT/IG/

- Charter: Leverage web standards and technology to enable IoT interoperation
- Web architecture: https://www.w3.org/standards/webarch/

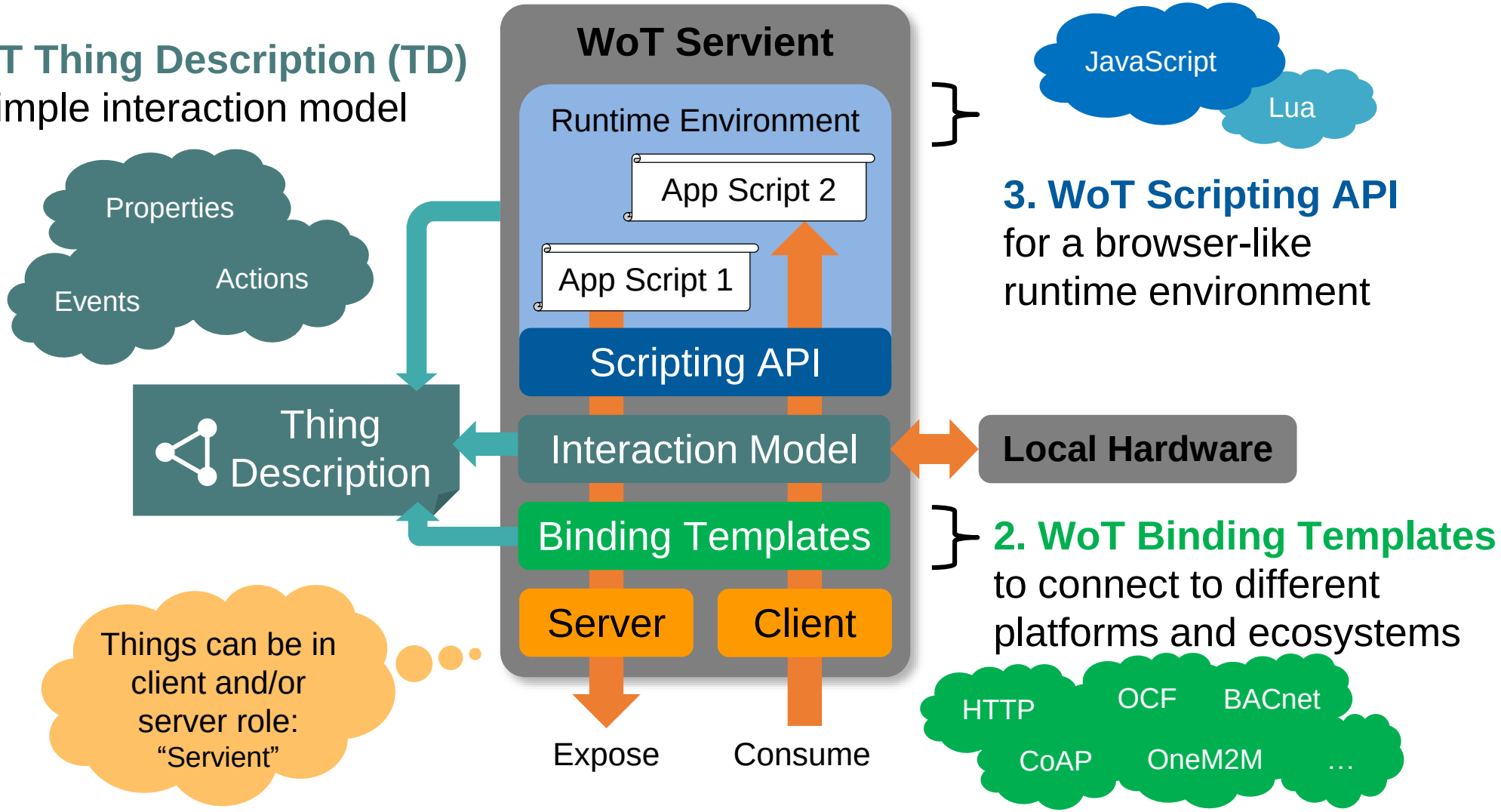
Web of Things Working Group to develop standard recommendations:

- Charter: https://www.w3.org/2016/09/wot-wg-charter.html
- Co-chairs: Matthias Kovatsch (Siemens), Kazuo Kajimoto (Panasonic), Michael McCool (Intel)
- White paper on WoT architecture: http://w3c.github.io/wot/charters/wot-white-paper-2016.html

WoT current practices:
http://w3c.github.io/wot/current-practices/wot-practices.html

WoT: Deliverables/Architecture

1. WoT Thing Description (TD)
with simple interaction model



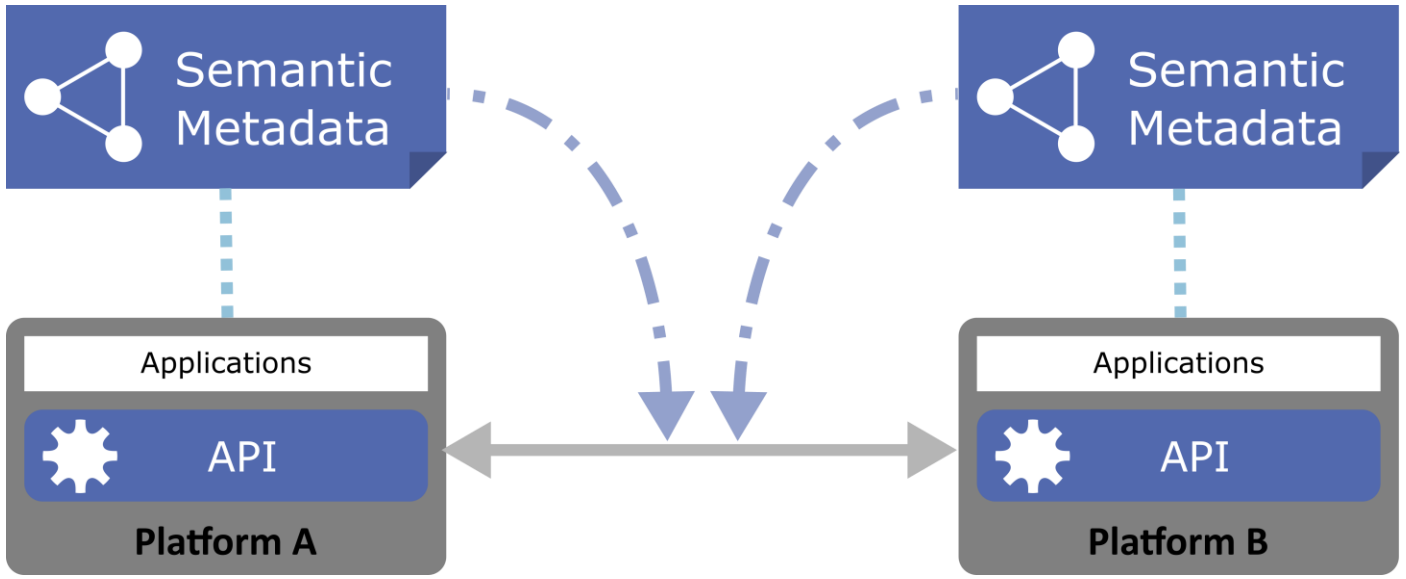
3. WoT Scripting API
for a browser-like runtime environment

2. WoT Binding Templates
to connect to different platforms and ecosystems

WoT Key Deliverable: Thing Description

Standardized Semantic Metadata

- Protocol-independent description of network APIs
- Communication and security requirements
- Data models and constraints
- Semantic annotation



Thing Description Example


```
{
  "@context": [
    "http://w3c.github.io/wot/w3c-wot-td-context.jsonld",
    { "domain": "http://example.org/actuator#" }
  ],
  "@type": "Thing",
  "name": "MyLEDThing",
  "base": "coap://myled.example.com:5683/",
  "security": {
    "cat": "token:jwt",
    "alg": "HS256",
    "as": "https://authority-issuing.example.org"
  },
  "interaction": [
    {
      "@type": ["Property", "domain:onOffStatus"],
      "name": "status",
      "outputData": {"valueType": {"type": "boolean"}},
      "writable": true,
      "links": [
        {
          "href": "pwn"
```

```
"interaction": [
  {
    "@type": ["Property", "domain:onOffStatus"],
    "name": "status",
    "outputData": {"valueType": {"type": "boolean"}},
    "writable": true,
    "links": [
      {
        "href": "pwr",
        "mediaType": "application/exi"
      },
      {
        "href": "http://mytemp.example.com:8080/status",
        "mediaType": "application/json"
      }
    ]
  },
  {
    "@type": ["Action", "domain:fadeIn"],
    "name": "fadeIn",
    "inputData": {
      "valueType": {"type": "integer"},
      "domain:unit": "domain:ms"
    },
    "links": [
      {
        "href": "in",
        "mediaType": "application/exi"
```

Property

Action

```
      "inputData": {
        "valueType": {"type": "integer"},
        "domain:unit": "domain:ms"
      },
      "links": [
        {
          "href": "out",
          "mediaType": "application/exi"
        },
        {
          "href": "http://mytemp.example.com:8080/out",
          "mediaType": "application/json"
        }
      ]
    },
    {
      "@type": ["Event", "domain:alert"],
      "name": "criticalCondition",
      "outputData": {"valueType": {"type": "string"}},
      "links": [
        {
          "href": "ev",
          "mediaType": "application/exi"
        }
      ]
    }
  ]
}
```

Event
(sources, sinks, …)

Problem 1: Local Links

Risks

- WoT is predicated on Web standards being useful for IoT
  - For example, being able to use web browsers as IoT device user interfaces
- However…
  - Web is oriented towards browsers and human-readable information
  - IoT is oriented towards machine-to-machine communications
  - Web browsers assume an active full internet connection
  - IoT devices may only have local network connectivity

One major pain point:

- Browser assumptions about certificate revocation checking under HTTPS
- Primarily affects use of HTTPS for “local” user interfaces

Problem 2: Vulnerability Analysis

Risks

- Pervasive metadata allows attacker to analyze a system in detail to find vulnerabilities and plan an attack

Opportunity

- Pervasive metadata allows a system owner to analyze a system in detail to find vulnerabilities and prevent attacks

➔ Metadata needs to be made available only to trusted users.

Note: Semantic annotation makes this more powerful, as we can also automate interferencing, and also get more information about the physical installation and use of the device.

Problem 3: Endpoint Adaptation

Risks

- Protocol conversion bridges are vulnerable to attack, as protocol conversion may require “unpacking” data in flight, making it available to interception

Opportunity

- A system wishing to talk to a WoT Thing can access the metadata for a thing and set up an end-to-end encrypted channel directly to that thing, bypassing intermediate payload translation steps

Problem 4: Semantic Discovery

Risks

- Semantic search is relatively expensive
  - Pathological semantic queries can be created that can consume an unreasonable amount of resources
- If semantic discovery services are “open”, then they will be subject to denial of service attacks

Opportunity

- Semantic discovery is a powerful capability we would like to make available to users

➔ **How to manage trust?**

Problem 5: Security Metadata

Opportunity

- A Thing Description can provide information that can be useful for enabling distributed and/or decentralized security mechanisms

→ How can we make validated and authenticated TDs available in a decentralized fashion?

→ What security mechanisms should we support and what information do they need?

Examples:

- Payments/deposits for services: Interledger address
- Web of Trust: References
- Access Control: Management Thing API
- Discovery: Distributed discovery services and Thing Directories
- Caching and proxies: Encrypted state and expected lifetime of cached data
- Firewalls: Network policy, including expected incoming *and* outgoing traffic

Summary

Main W3C WoT deliverable and differentiator:

- Universal metadata format (“Thing Description”) for IoT services (“Things”) and associated common Thing abstraction

Use of Web Standards for IoT has some specific issues:

- Local links, HTTPS, and certificates

Use of semantic metadata has specific risks and opportunities:

- Vulnerability analysis
- Endpoint adaptation vs. link-by-link translation
- Preventing denial-of-service attacks on semantic discovery services
- Securing metadata and providing it only to trusted entities

Follow-up Actions

- WoT Plugfest – Prague, March 24-25, 2018
- WoT Interest Group
- WoT Working Group
  - W3C Web of Things Security and Privacy Considerations
- Collaborations
- WoT Security Validation

Web of Things: Interest Group Members

To add to the paper…

- Expanded list of security metadata given in “Problem 5” slide on this presentation
- Reference to Michal Krol’s paper at NDSS DISS using cryptocurrency deposits to support offload markets; this is one possible solution to how to provide open access to discovery services.
- Discussion of “endpoint adaptation” issues wrt filtering proxies (eg what if proxies require access to headers etc. to do packet filtering?)