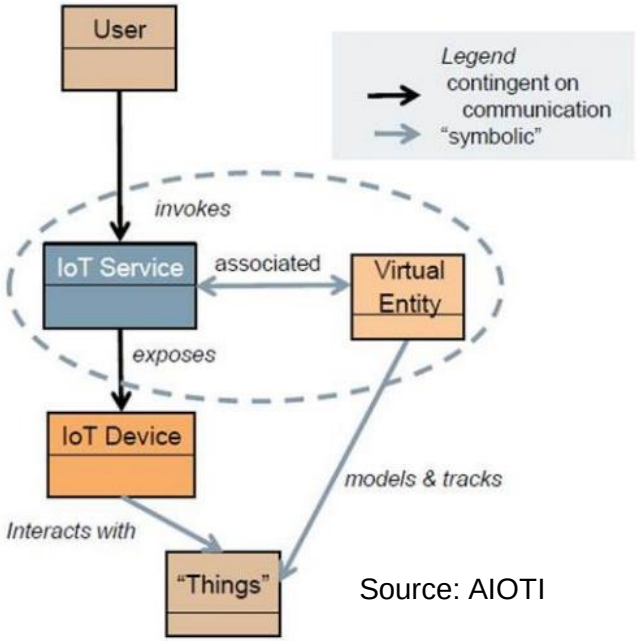
PBL5

- **Interoperability & semantics in IoT**
- **MQTT Sparkplug B/Protobuf**

Jarosław Domaszewicz

Instytut Telekomunikacji Politechniki Warszawskiej



Source: AIOTI

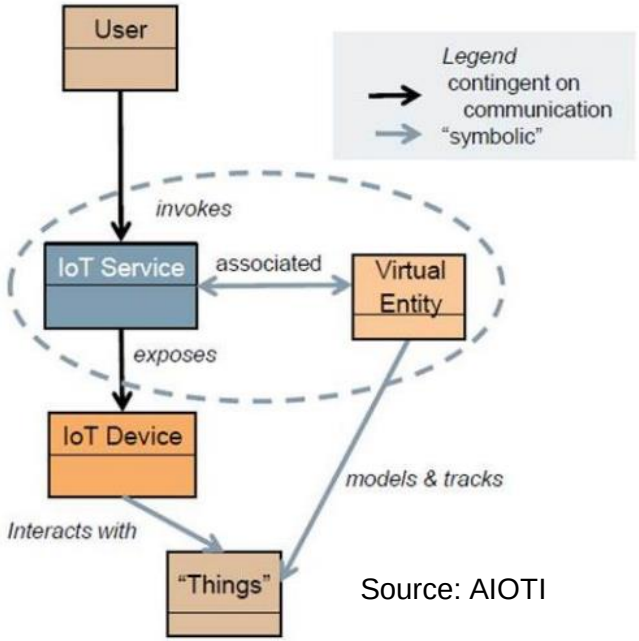
Objectives of this workshop

- Understand the concept of (application layer) interoperability.
  - syntactic vs. semantic interoperability
  - information model, data model, interaction model
- Understand the concept of semantics in IoT.
  - see some examples
- Get familiar with Sparkplug B.
  - an ”extension” of MQTT targeted at Industrial IoT, IIoT
  - shows what may need to be added to MQTT to strengthen interoperability
- Get familiar with Protocol Buffers (Protobuf).
  - used in Sparkplug B
  - a high-level message structure description and efficient, binary representation
  - in IoT, every byte counts

YOUR WORKSHOP ACTIVITIES

- Develop your own extension to MQTT for the smart home domain.
  - tools: your analytical skills
- Experiment with Protobuf in Node-RED.
  - tools: Node-RED, node-red-contrib-protobuf
- Experiment with Protobuf in Python.
  - tools: Python, protoc (a Protobuf compiler), google.protobuf package
- Use MQTT to send Protobuf-encoded messages with Python and receive them with Node-RED.
  - tools: Python, gogle.protobuf package, Eclipse Paho MQTT Python client library, Node-RED, mqtt_in, node-red-contrib-protobuf

Keyword of this workshop: interoperability



Source: AIOTI

Source:
https://en.wikipedia.org/wiki/Interoperability

interoperacyjność

INTEROPERABILITY

- Interoperability is a characteristic of a product or system to work with other products or systems.

- Interoperability implies exchanges between products from *different* vendors. **independently developed products**

- Interoperability based on *open standards*: different vendors can use the standards document to make products that implement the standard and are thus *interoperable by design*.
  - the vendors' products compete on the quality of their implementation, user interface, ease of use, performance, price, …
  - the customer can choose to switch to another competing product

Source:
https://en.wikipedia.org/wiki/Interoperability

SYNTACTIC AND SEMANTIC INTEROPERABILITY

interoperacyjność

- *Syntactic interoperability*: two or more systems use common data formats and communication protocols and are capable of communicating with each other.

  data formats and protocols

- *Semantic interoperability*: the ability to automatically interpret the information exchanged.
  - to achieve semantic interoperability, both sides must refer to a common information exchange reference model

    Information exchange reference model

Źródło:*Report from the Internet of Things (IoT) Semantic Interoperability (IOTSI) Workshop 2016,* RFC 8477, 2018

INFORMATION MODEL

- An information model defines an environment at the highest level of abstraction.
  - implementation details are hidden

Źródło:*Report from the Internet of Things (IoT) Semantic Interoperability (IOTSI) Workshop 2016,* RFC 8477, 2018

DATA MODEL

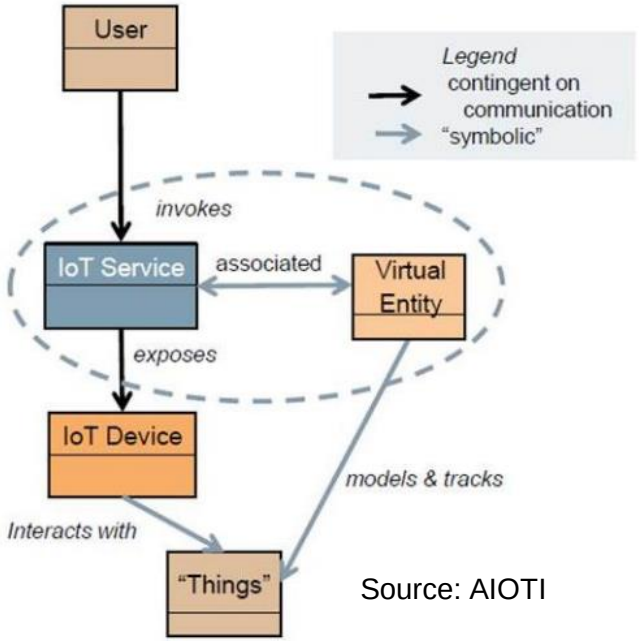
- A data model defines concrete data representations at a lower level of abstraction, including implementation- and protocol-specific details.

Źródło:*Report from the Internet of Things (IoT) Semantic Interoperability (IOTSI) Workshop 2016,* RFC 8477, 2018

INTERACTION MODEL

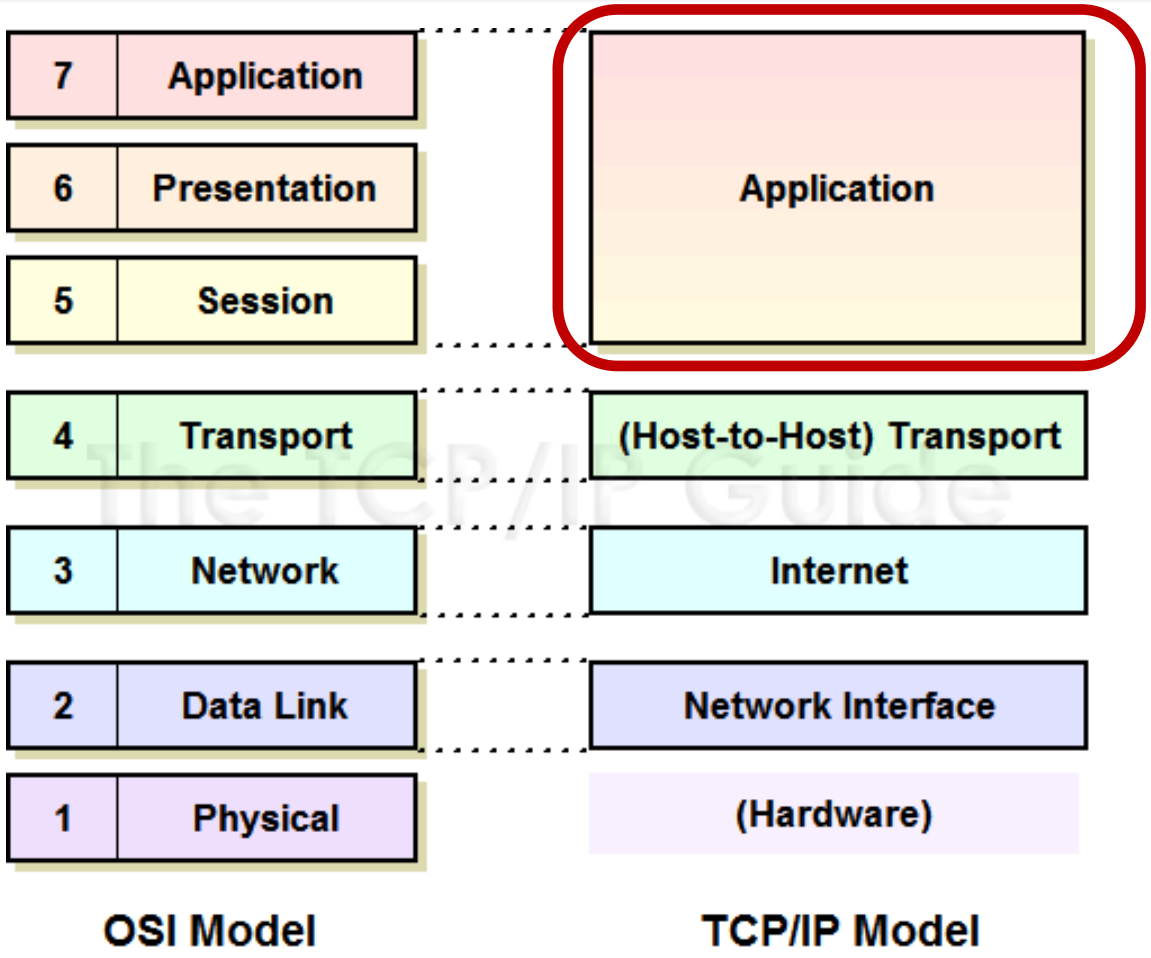
- An interaction model defines how data is accessed and retrieved from the endpoints, being, therefore, tied to the specific communication pattern.
  - e.g., REST methods, Publish/Subscribe operations, or RPC calls

Layer of this workshop: application layer



Source: AIOTI

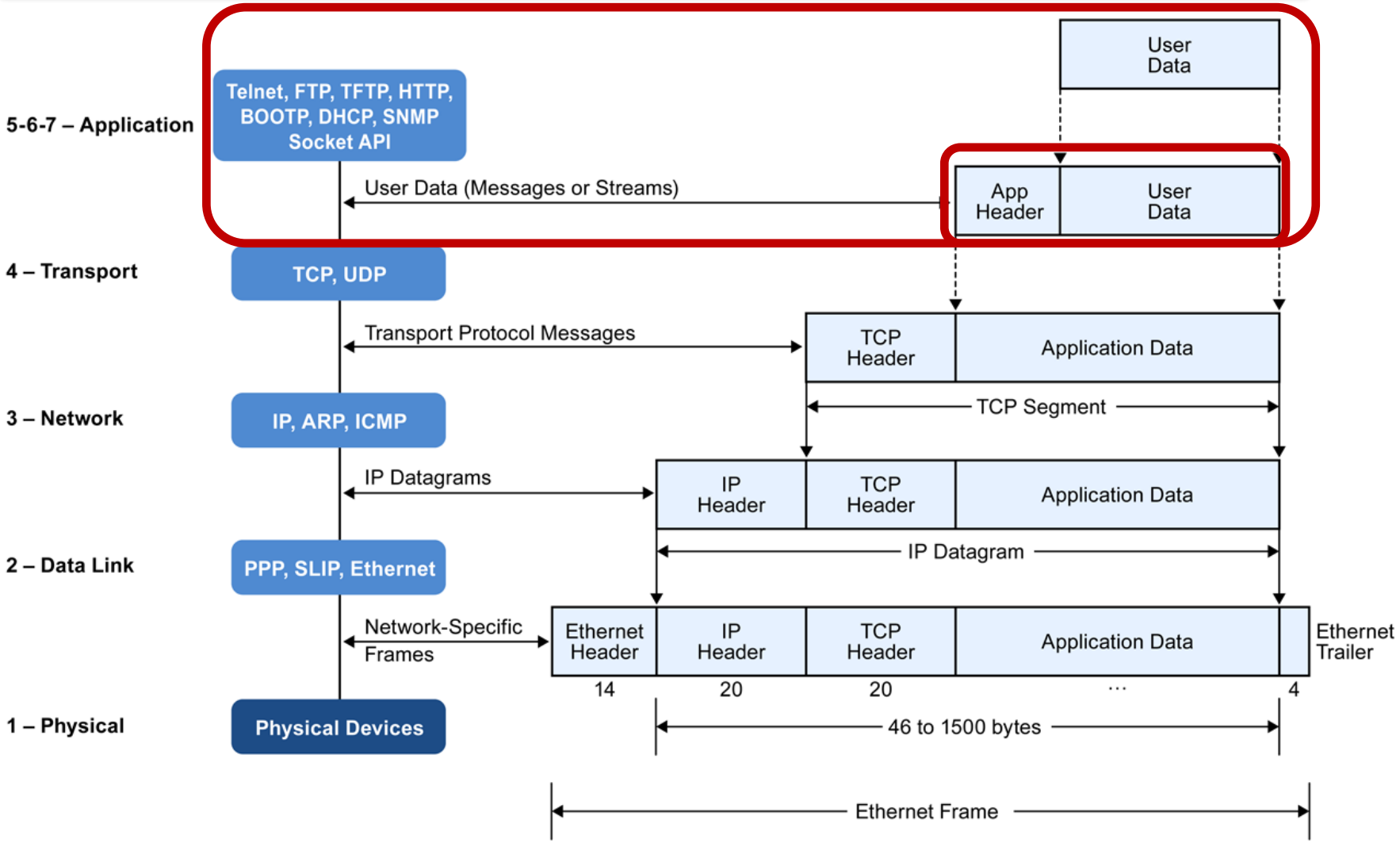
APPLICATION LAYER (1/3)



Źródło: http://www.tcpipguide.com

Innovations ? In the application layer!

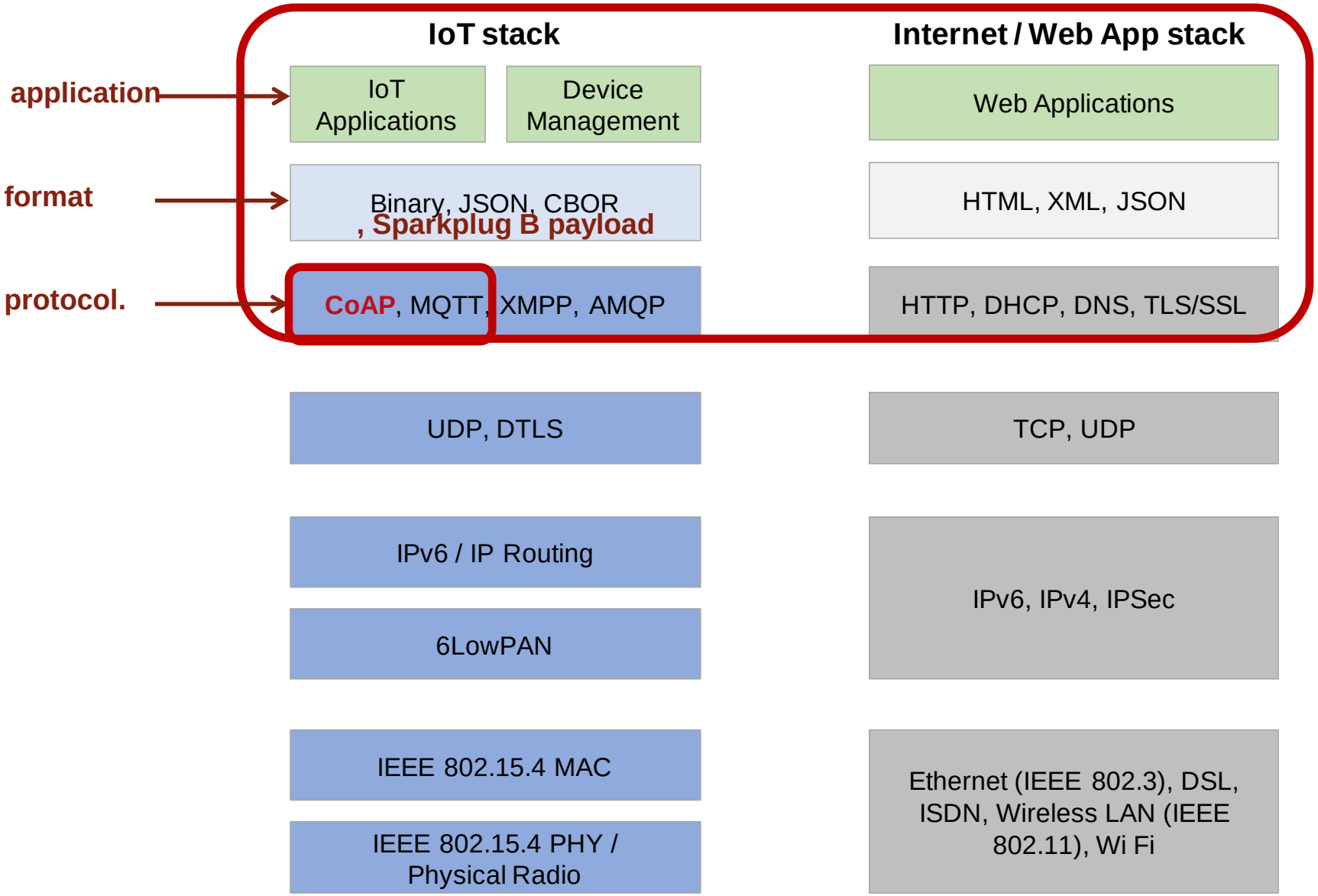
APPLICATION LAYER (2/3)



Source: https://www.micrium.com/iot/internet-protocols

Source:
Constrained Application Protocol (Web Protocol for IoT)
A. Chakrabarti, www.slideshare.net

APPLICATION LAYER (3/3)

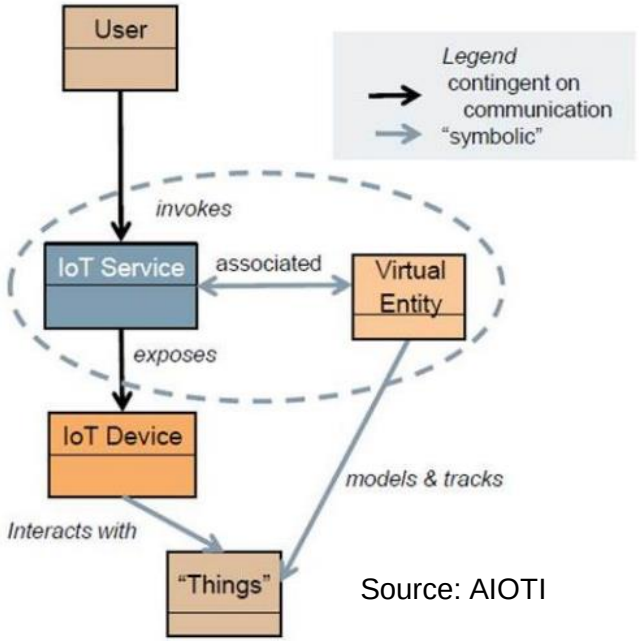


IoT APPLICATION LAYER COMPETITORS

- CoAP (Constrained Application Protocol)
  - developed by CoRE,Constrained RESTful Environments WG of IETF
  - an Internet (IETF) standard
  - runs on top of UDP
  - enables HTTP-like request/respons interactions in IoT: client/server, restful APIs

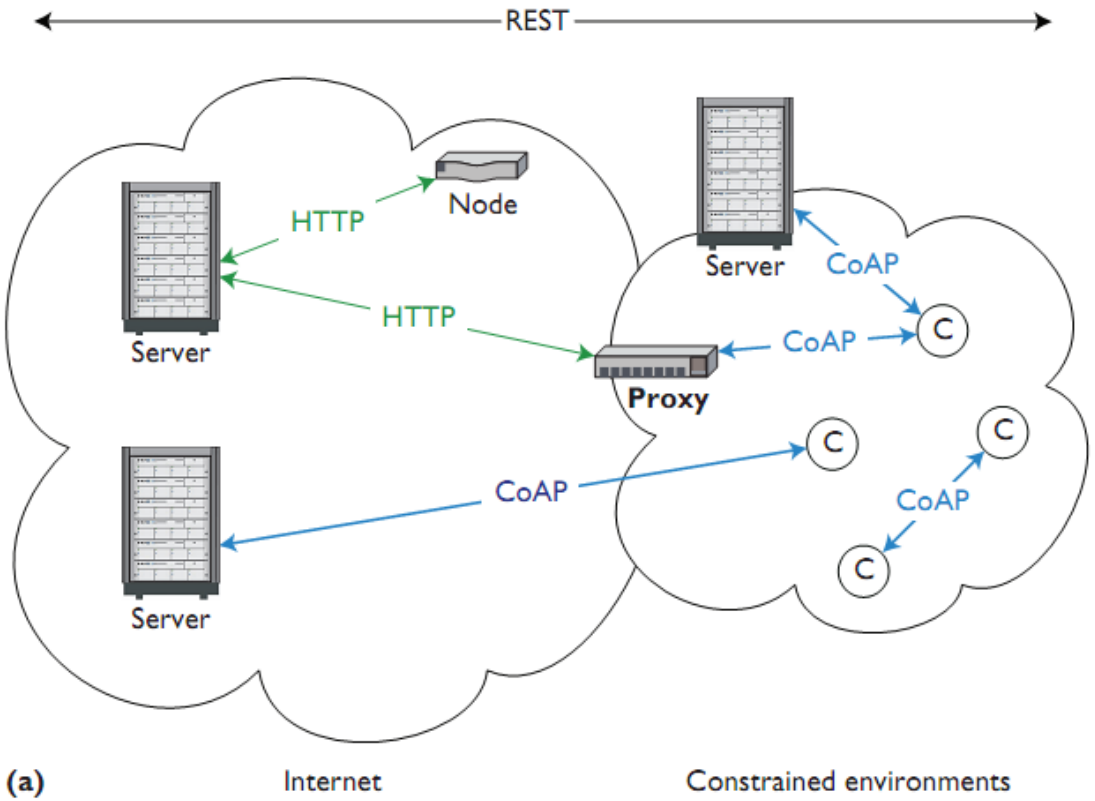
- MQTT (formerly Message Queuing Telemetry Transport, now MQTT)
  - developed by industry (IBM, Arcom)
  - supported by a major IBM product (MQ series)
  - now an OASIS standard and ISO standard
  - runs on top of TCP
  - based on the publish/subscribe interaction paradigm

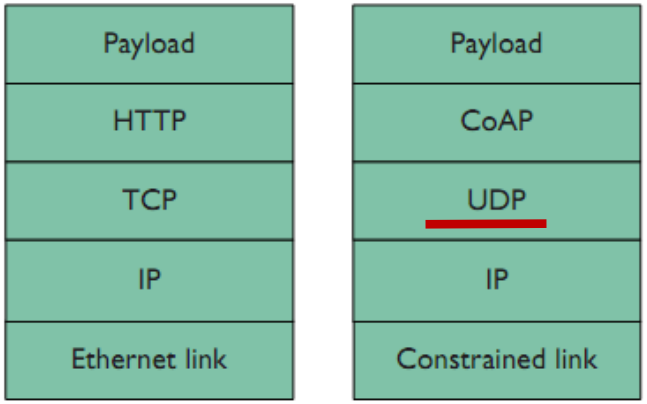
CoAP – support for interoperability



Source: AIOTI

CoAP SYSTEM ARCHITECTURE





Source: *CoAP: An Application Protocol for Billions of Tiny Internet Nodes*
C. Bormann, A. P. Castellani, Z. Shelby
IEEE INTERNET COMPUTING, 2012

CoAP: key RFCs

- [RFC7252] "The Constrained Application Protocol (CoAP)"
  - Z. Shelby, K. Hartke, C. Bormann, June 2014

  the main CoAP specification, 112 pages
- [RFC7641] "Observing Resources in CoAP"
  - K. Hartke, September 2015

  how to be up to date about the state of a resource without too many requests
- [RFC7959] "Blockwise Transfers in CoAP"
  - C. Bormann, Z. Shelby, August 2016

  how to transfer big resource representations
- [RFC6690] "Constrained RESTful Environments (CoRE) Link Format"
  - Z. Shelby, August 2012

  how to discover resources hosted by a server

CoAP MESSAGE FORMAT

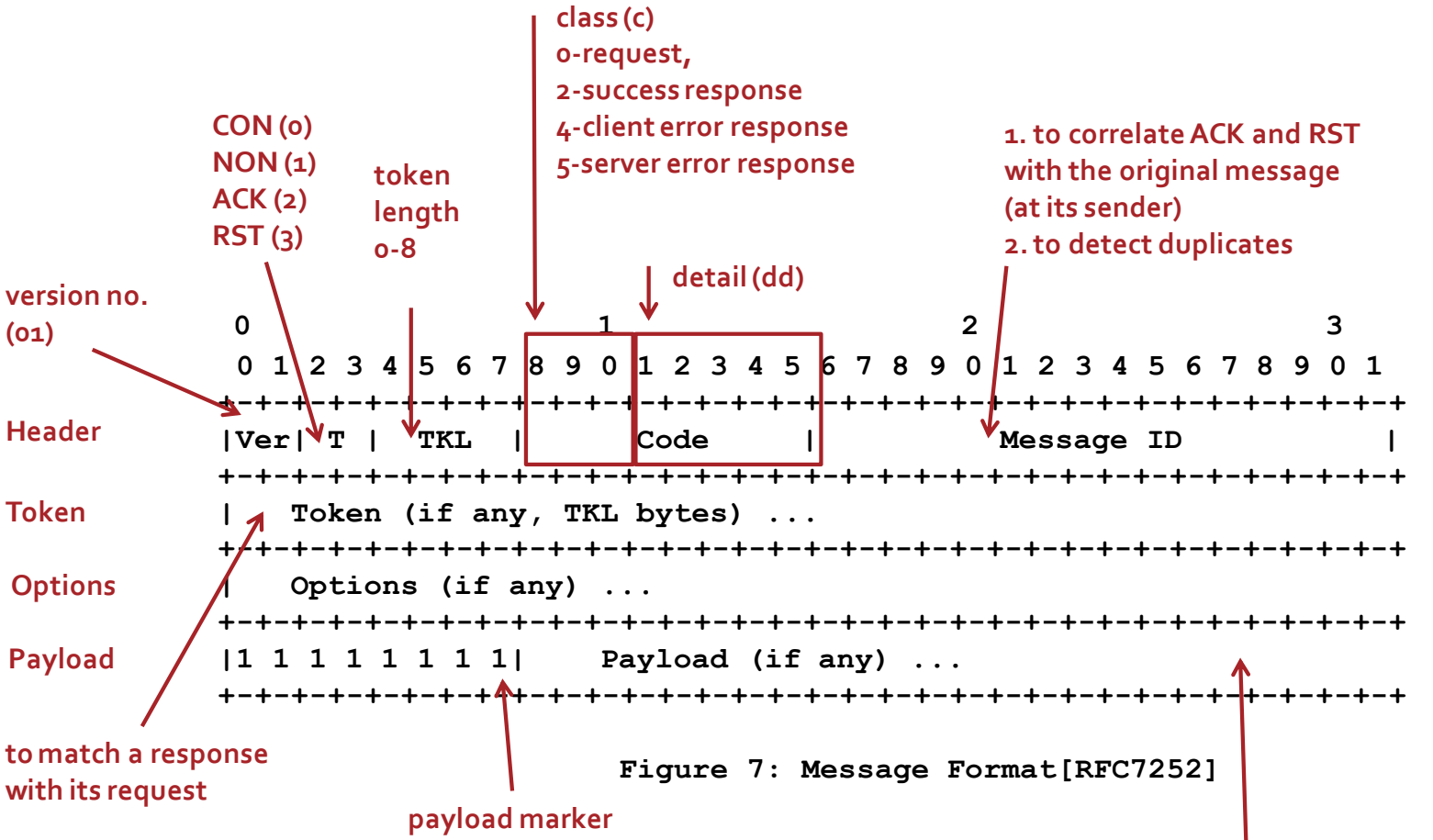


Figure 7: Message Format[RFC7252]

c.dd indicates a Request Method or a Response Code

0.00 Empty message
0.01 GET
0.02 POST
0.03 PUT
0.04 DELETE

2.dd success
4.dd client error
5.dd server error

CoAP OPTIONS

option = (option number, option value)

option number → No. … ← … of option value

No.	Name	Format	Length	Default
1	If-Match	opaque	0-8	(none)
3	Uri-Host	string	1-255	(see below)
4	ETag	opaque	1-8	(none)
5	If-None-Match	empty	0	(none)
7	Uri-Port	uint	0-2	(see below)
8	Location-Path	string	0-255	(none)
11	Uri-Path	string	0-255	(none)
12	Content-Format	uint	0-2	(none)
14	Max-Age	uint	0-4	60
15	Uri-Query	string	0-255	(none)
17	Accept	uint	0-2	(none)
20	Location-Query	string	0-255	(none)
35	Proxy-Uri	string	1-1034	(none)
39	Proxy-Scheme	string	1-255	(none)
60	Size1	uint	0-4	(none)

Table 4: Options [RFC7252]

SELECTED OPTIONS

- **`Content-Format`**
  - the representation format of the payload
- **`Etag`**
  - an entity-tag is intended for use as a resource-local identifier for a specific representation of a resource; generated by the server providing the resource; used for validation
- **`Max-Age`**
  - the maximum time a response may be cached before it is considered not fresh, default: 60s
- **`Accept`**
  - in a request, the client can indicate which content-format it prefers to receive

CONTENT FORMATS (CONTENT-FORMAT OPTION)

used for CoAP resource discovery

arbitrary binary data

Efficient XML Interchange (binary)

Media type	Encoding	ID	Reference
text/plain; charset=utf-8	-	0	[RFC2046] [RFC3676] [RFC5147]
application/link-format	-	40	[RFC6690]
application/xml	-	41	[RFC3023]
application/octet-stream	-	42	[RFC2045] [RFC2046]
application/exi	-	47	[REC-exi-20140211]
application/json	-	50	[RFC7159]

Table 9: CoAP Content-Formats [RFC7252]

Concise Binary Object Representation →

7.4. CoAP Content-Format

Media Type: application/cbor

Id: 60

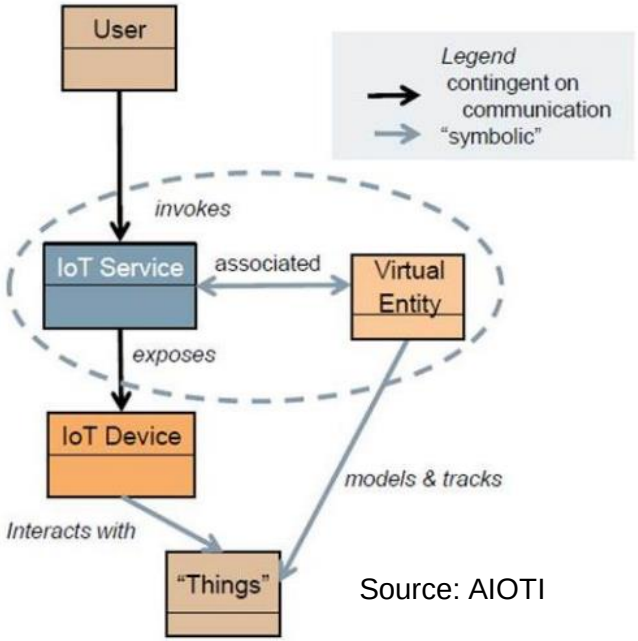
Source: *Concise Binary ObjectRepresentation (CBOR)* , [RFC7049]

So we know that the payload (a representation of a resource) is, say, in JSON. Is this enough?

CoRE Link Format

how to discover resources hosted by a server

[RFC6690] "Constrained RESTful Environments (CoRE) Link Format" Z. Shelby, August 2012



Source: AIOTI

WEB LINKING IN COAP: CORE RESOURCE DISCOVERY

- a default URI to discover resources hosted by a constrained server: **/.well-known/core**

- the resource at /.well-known/core is serialized using the **CoRE Link Format**
  - carried as payload (in HTTP this is a header)
  - assigned an Internet media type: **application/link-format**

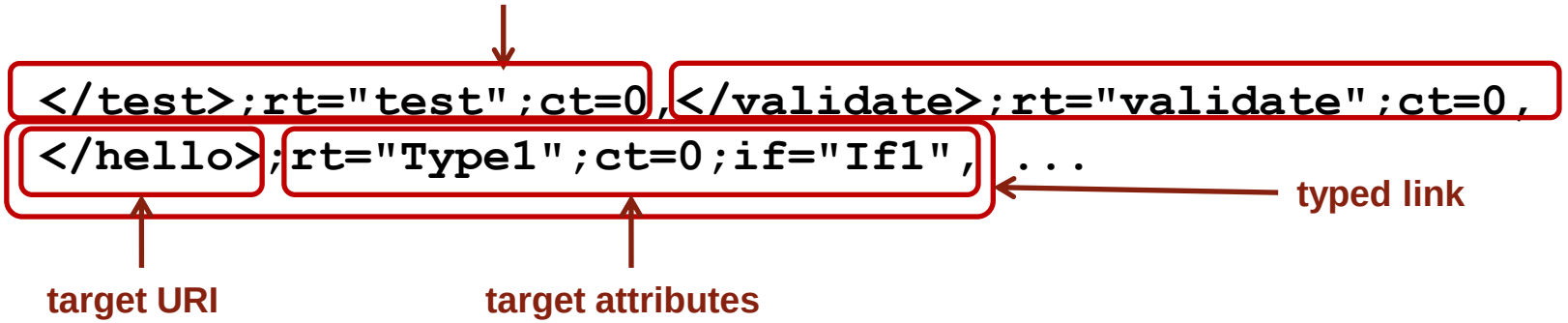
Media type	Encoding	ID	Reference
text/plain; charset=utf-8	-	0	[RFC2046] [RFC3676] [RFC5147]
application/link-format	-	40	[RFC6690]
application/xml	-	41	[RFC3023]
application/octet-stream	-	42	[RFC2045] [RFC2046]
application/exi	-	47	[REC-exi-20140211]
application/json	-	50	[RFC7159]

Table 9: CoAP Content-Formats [RFC7252]

TYPED LINK IN CORE

- in CoRE, a so-called typed link consists of
  - a target URI (the URI of a resource hosted by the constrained server)
  - target attributes (key/value pairs)

a part of the payload received from coap://coap.me:5683 in response to `GET /.well_known/core` (three typed links shown)

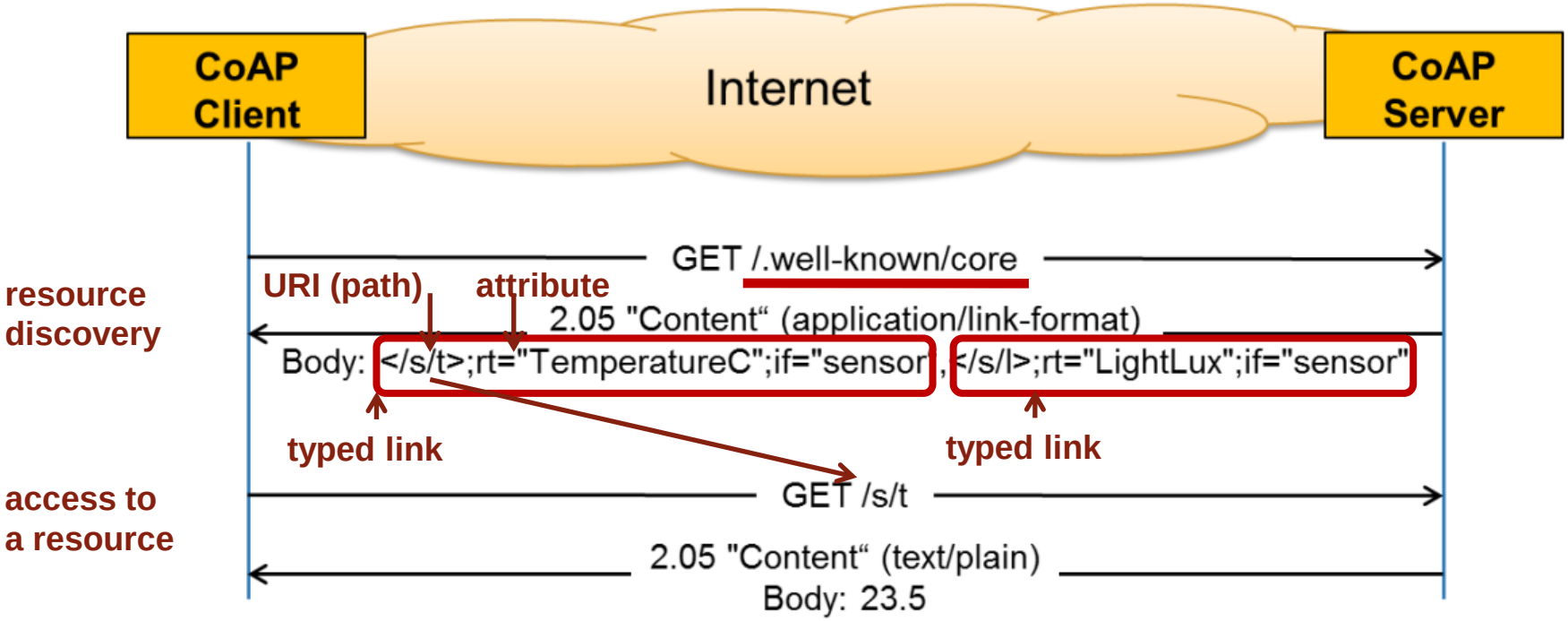


**the typed link says: „This server hosts a resource with the URI path `/hello`.
The resource is characterized by the following values of the attributes rt, ct, and if."**

TARGET ATTRIBUTES

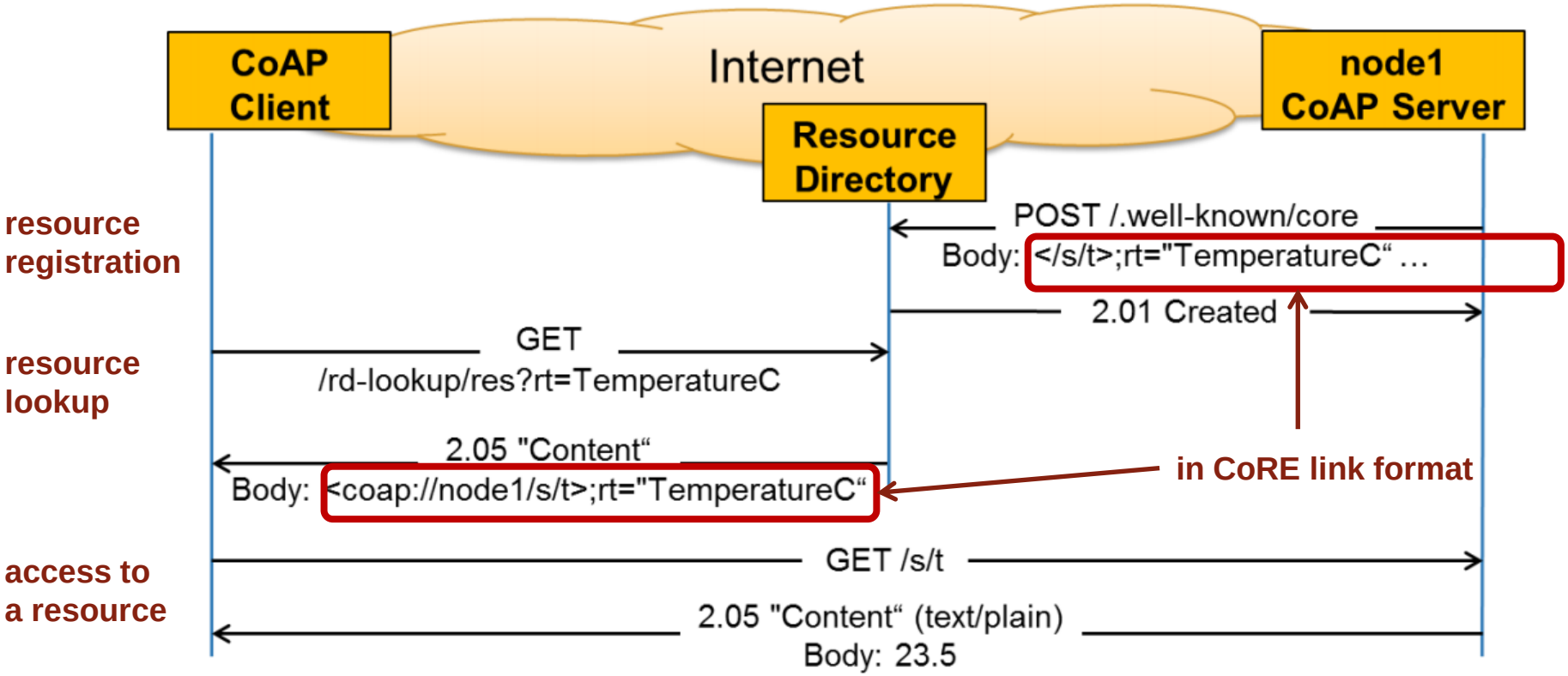
- **Resource Type**, **rt**: application-specific semantic type of a resource
  - example: outdoor-temperature
  - example: http://sweet.jpl.nasa.gov/2.0/phys.owl#Temperature
- **Interface Description**, **if**: describes the REST interface to interact with a resource
  - example: sensor
  - example: http://www.example.org/myapp.wadl#sensor
- **Maximum Size Estimate**, **sz**: an indication of the maximum size of the resource representation returned by performing a GET
- **Content type**, **ct**: a hint about the Content-Format this resource returns
- **Observable**, **obs**: a hint indicating that the resource is useful for observation (not a promise that the Observe Option can be used)

CoRE Resource Discovery: example



Source: *Flexible Unicast-Based Group Communication for CoAP-Enabled Devices* I.Ishaq et al., Sensors, 2014, 14

CoRE Resource Directory



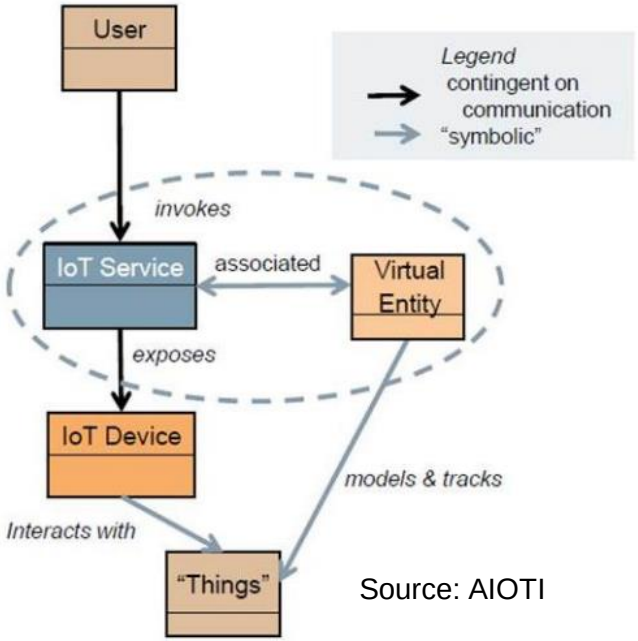
Source: *Flexible Unicast-Based Group Communication for CoAP-Enabled Devices* I.Ishaq et al., Sensors, 2014, 14

- See [draft-ietf-core-resource-directory-07] Z. Shelby et al. "CoRE Resource Directory", March 2016

Some questions on resource discovery

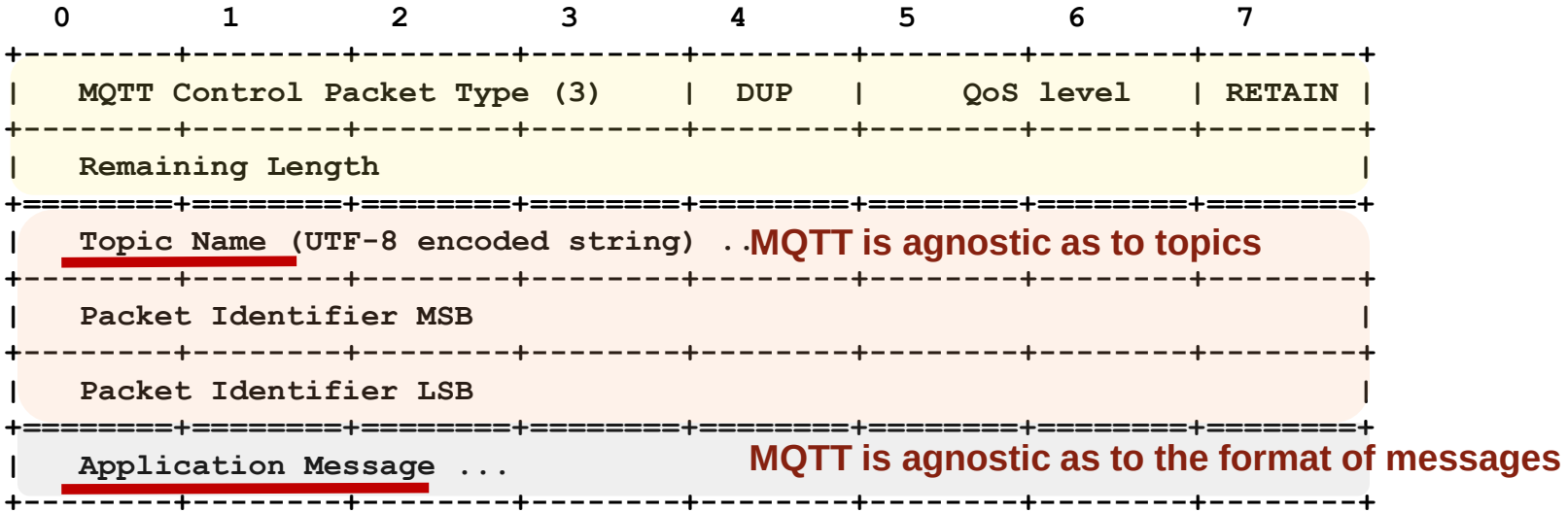
- how to identify resource types (rt=) and describe interfaces (if=)?
- how to ensure that the client "understands" these attributes?
- aside: what does it mean to "understand"?
  - recall semantic interoperability
- the resource discovery mechanism described so far may not be enough
- answer: **semantics**

MQTT – support for interoperability



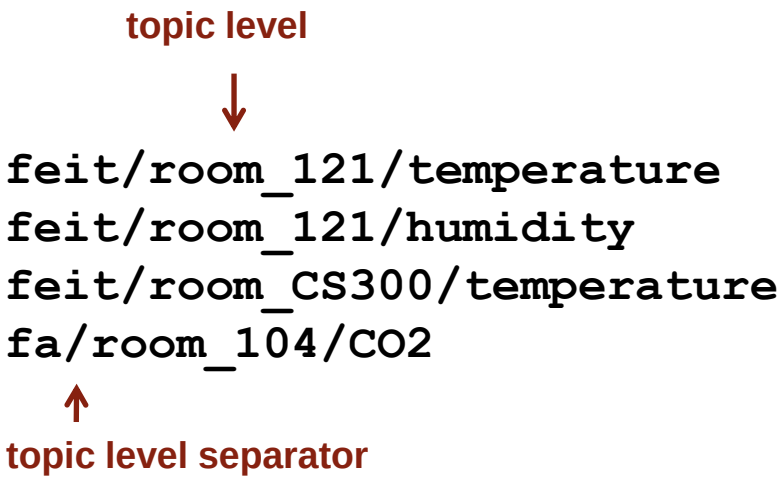
Source: AIOTI

PUBLISH


0 1 2 3 4 5 6 7
MQTT Control Packet Type (3)
DUP
QoS level
RETAIN
Remaining Length
Topic Name (UTF-8 encoded string) ..
MQTT is agnostic as to topics
Packet Identifier MSB
Packet Identifier LSB
Application Message ...
MQTT is agnostic as to the format of messages

TOPICS

- topics have a hierarchical structure:



- the broker accepts any topic, without earlier registration
  - MQTT is agnostic as to topics (just like with application messages)
- topics starting with ”$” are reserved (e.g., for monitoring purposes)

```
feit/room_121/temperature
feit/room_121/humidity
feit/room_CS300/temperature
fa/room_104/CO2
```

TOPIC FILTERS

- to subscribe to more than one topic at once, use **wildcards**
  - allowed only in topic filters (in SUBSCRIBE)

- **single-level** wildcard (**+**)
  - example: `feiti/+/temperature` (all temperature readings from FEITI)
  - example: `+/+/CO2` (all CO2 readings from entire WUT)

- **multi-level** wildcard (**#**)
  - example: `fa/#` (all sensor readings from FA)
  - must be the last character in the topic

TOPICS IN REAL LIFE (1/3)

Decoupling in the spirit of publish/subscribe.

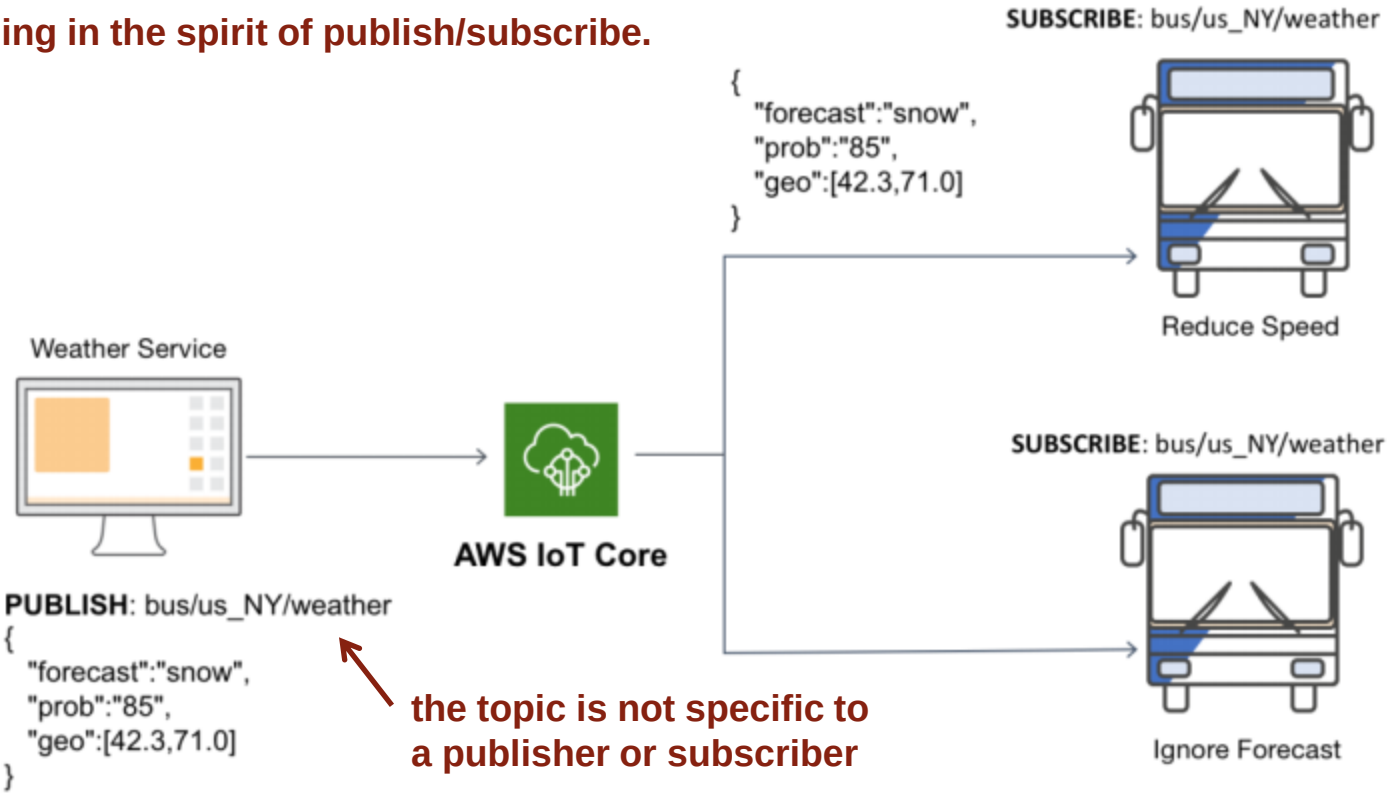


Figure 3: One-to-many messaging in broadcast communication

Source: Amazon Web Services, *Designing MQTT Topics for AWS IoT Core*, May 2019

TOPICS IN REAL LIFE (2/3)

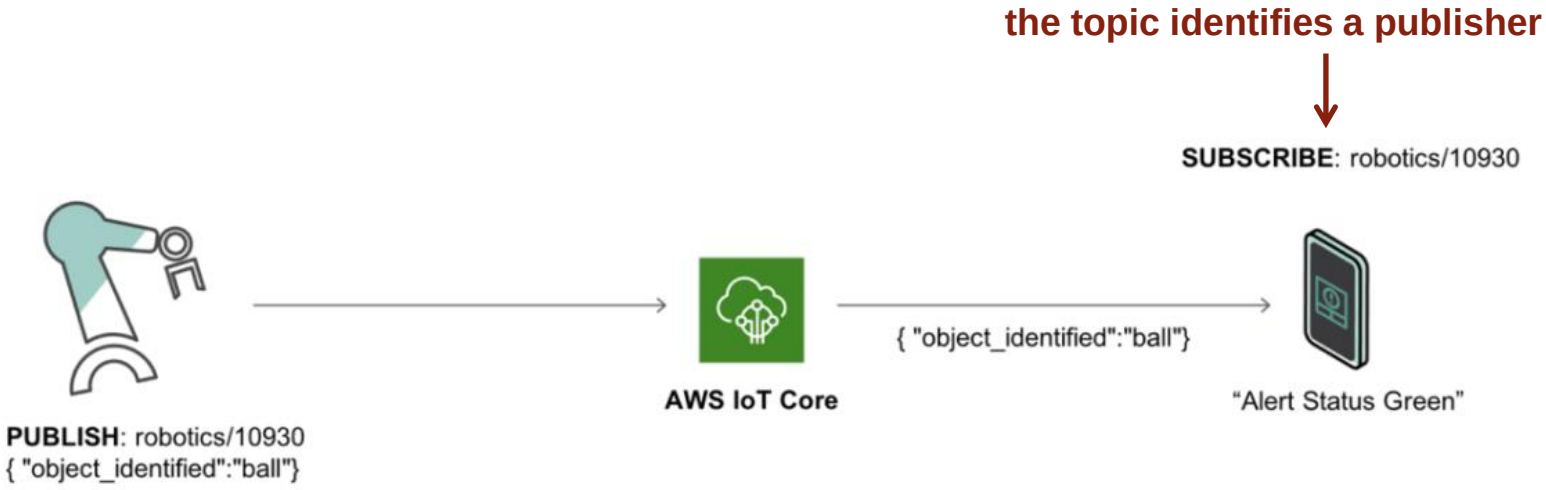


Figure 1: One to One Messaging in Point-to-point Communication

Not exactly in the spirit of publish/subscribe decoupling (the subscriber knows about the publisher).

Source: Amazon Web Services, *Designing MQTT Topics for AWS IoT Core*, May 2019

TOPICS IN REAL LIFE (3/3)

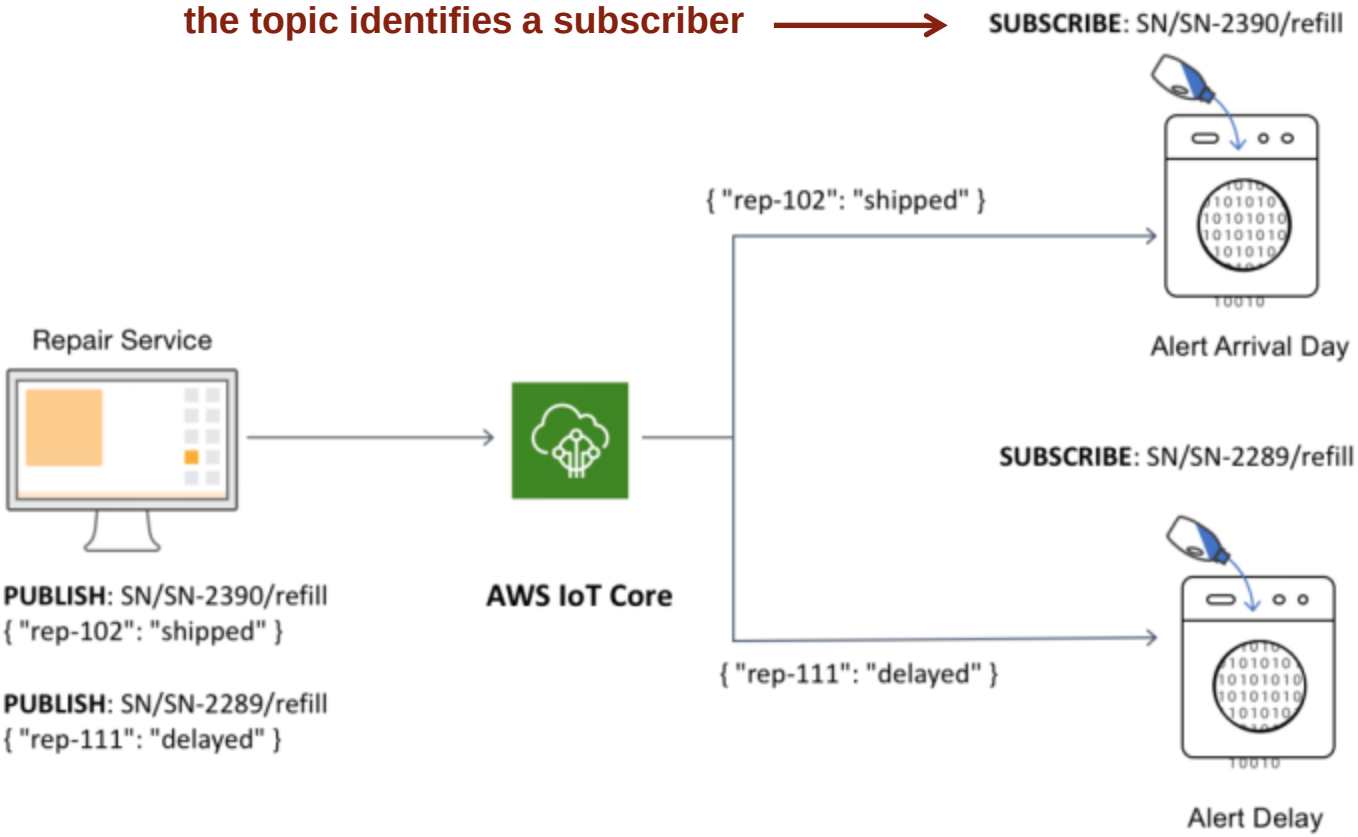
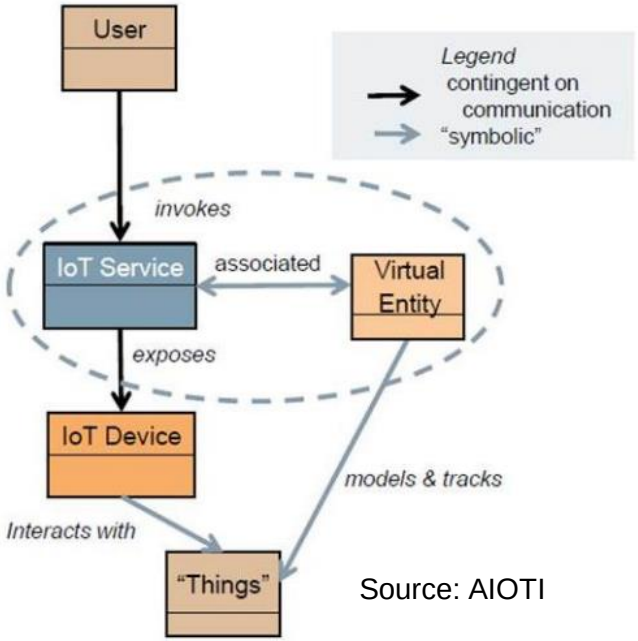


Figure 2: One-to-many messaging in point-to-point communication

Source: Amazon Web Services, *Designing MQTT Topics for AWS IoT Core*, May 2019

Not exactly in the spirit of publish/subscribe decoupling (the publisher knows about the subscribers).

Zadanie: uzupełnić MQTT tak, aby zapewnić interoperacyjność typu plug&play obiektów inteligentnego domu i aplikacji sterującej



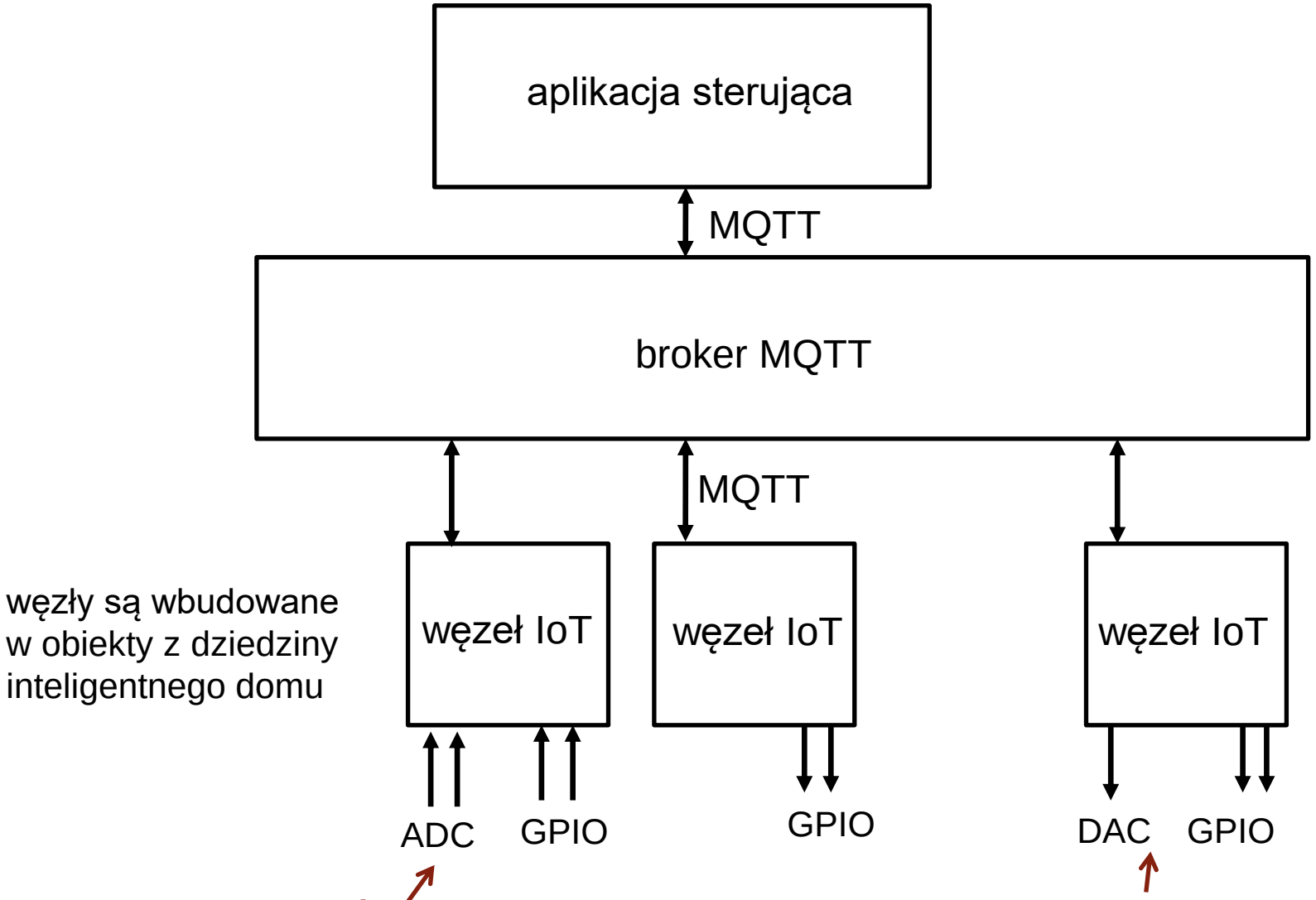
Source: AIOTI

CEL

- Wyspecyfikuj model informacyjny, model danych i model interakcji dla dziedziny inteligentnego domu, do wykorzystania z protokołem MQTT.

- Specyfikacja powinna uwzględniać
  - schemat tematów (*topics*)
  - format wiadomości (*payloads*)
  - schemat interakcji między elementami systemu

GENERYCZNA ARCHITEKTURA SYSTEMU



Założenia (1/3)

- Nie wiadomo z góry, jakie węzły będą dostępne i w jakich ilościach.
- Populacja węzłów zmienia się w trakcie pracy systemu
  - węzły (obiekty) mogą pojawiać się i znikać
- Węzły komunikują się tylko z aplikacją sterującą.
  - nie ma komunikacji bezpośredniej między węzłami

ZAŁOŻENIA (2/3)

- Aplikacja sterująca powinna mieć aktualną wiedzę o bieżącej populacji węzłów.
  - możliwości sensoryczno-wykonawcze dostępnych węzłów
  - ich lokalizacje logiczne
  - w jakich obiektach znajdują się dostępne węzły

- Aplikacja sterująca powinna mieć aktualną wiedzę o bieżącym stanie środowiska.
  - pełne wykorzystanie zasobów sensorycznych oferowanych przez węzły
  - aplikacja powinna wiedzieć, z którego węzła (obiektu) pochodzi odczyt z sensora

- Aplikacja sterująca powinna móc zmieniać stan środowiska.
  - wykorzystanie zasobów wykonawczych oferowanych przez węzły
  - aplikacja powinna użyć elementu wykonawczego z wybranego węzła (obiektu)

ZAŁOŻENIA (3/3)

- Specyfikacja powinna umożliwiać tworzenie węzłów i aplikacji sterujących przez niezależnych wytwórców, mających jedynie dostęp do specyfikacji.

- Nie wnikamy w algorytmy aplikacji sterującej.
  - zakładamy, że posiada algorytmy pozwalające jej wykorzystywać, nieznane z góry, zmieniające się zasoby sensoryczno-wykonawcze dostępnych aktualnie węzłów
  - takie algorytmy to „pole do popisu” dla wytwórców aplikacji sterujących

- Tam, gdzie to wygodne, należy wykorzystywać istniejące mechanizmy protokołu MQTT.

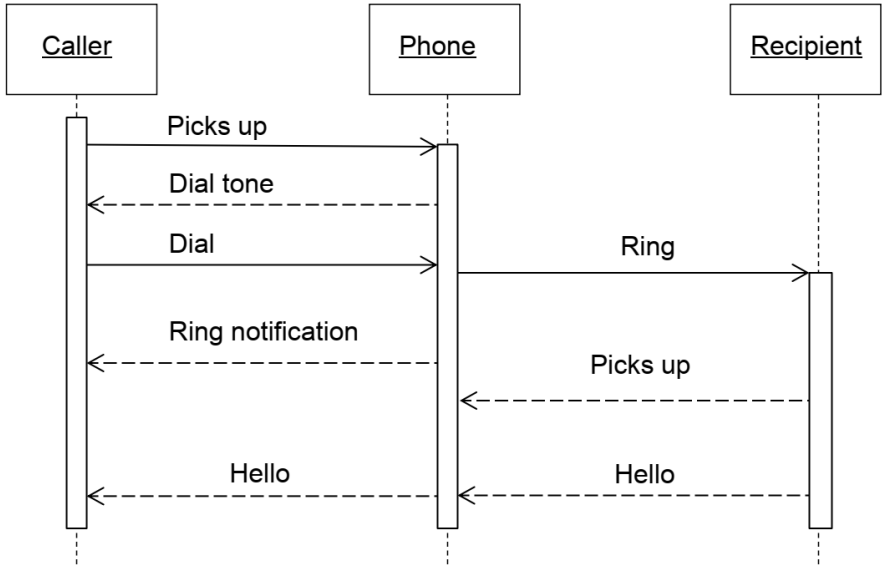
Model informacyjny

- Model informacyjny
  - lokalizacje logiczne (np. łazienka)
  - klasy obiektów (np. krzesło)
  - wielkości mierzone przez sensory węzłów wbudowanych w obiekty
  - akcje (*actuating actions*) do wykonania przez elementy wykonawcze węzłów wbudowanych w obiekty
  - rodzaje wiadomości wymieniane przez węzły i aplikację sterującą
  - treść dla różnych rodzajów wiadomości (opisana koncepcyjnie)

- Wynik:
  - listy pojęć, listy rodzajów, opis w języku naturalnym

Model interakcji

- Model interakcji
  - schematy tematów (*topics*)
  - kiedy wysyłane są wiadomości różnych rodzajów (”obowiązki publikacyjne” klientów MQTT)
  - ”obowiązki subskrypcyjne” klientów MQTT
  - typowe sekwencje wymiany wiadomości dla różnych warunków
- Wynik
  - nieformalny opis elementów modelu interakcji
  - diagramy sekwencji
    - np. UML *sequence diagrams*



Source: https://csis.pace.edu/~marchese/CS389/L9/Sequence%20Diagram%20Tutorial.pdf

Model danych

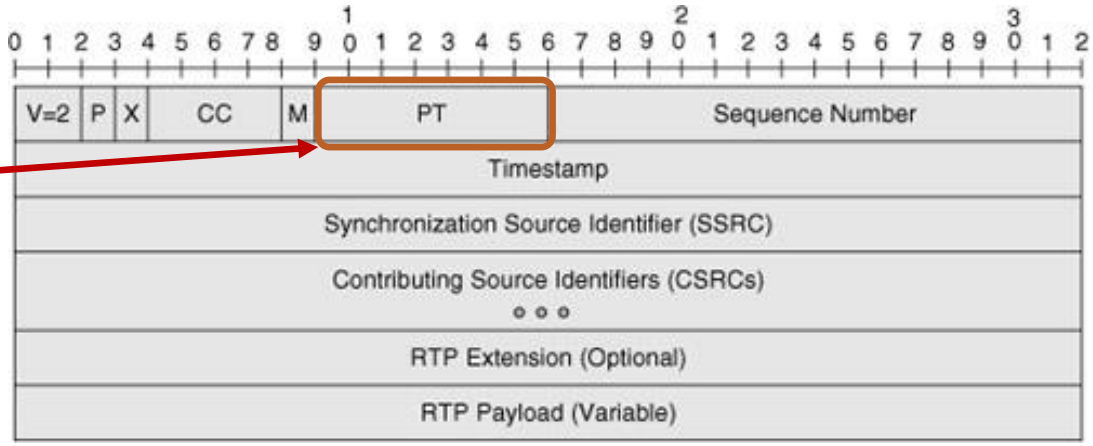
- Model danych
  - formaty wiadomości (MQTT payload): pola bajtów, pola bitowe
    - przykłady na następnych slajdach
  - kodowanie binarne dla pojęć z modelu informacyjnego
    - do użycia w poszczególnych polach wiadomości
    - dlaczego binarne? chcemy minimalizować ilość przesyłanych danych
- Wynik
  - diagram podobny do tych z poniższych przykładów
  - znaczenie sekwencji bitów w poszczególnych polach

FORMATY WIADOMOŚCI: PRZYKŁADY (1/3)

- RTP (Real-time Transport Protocol)
  - binarny protokół warstwy aplikacji dla VoIP

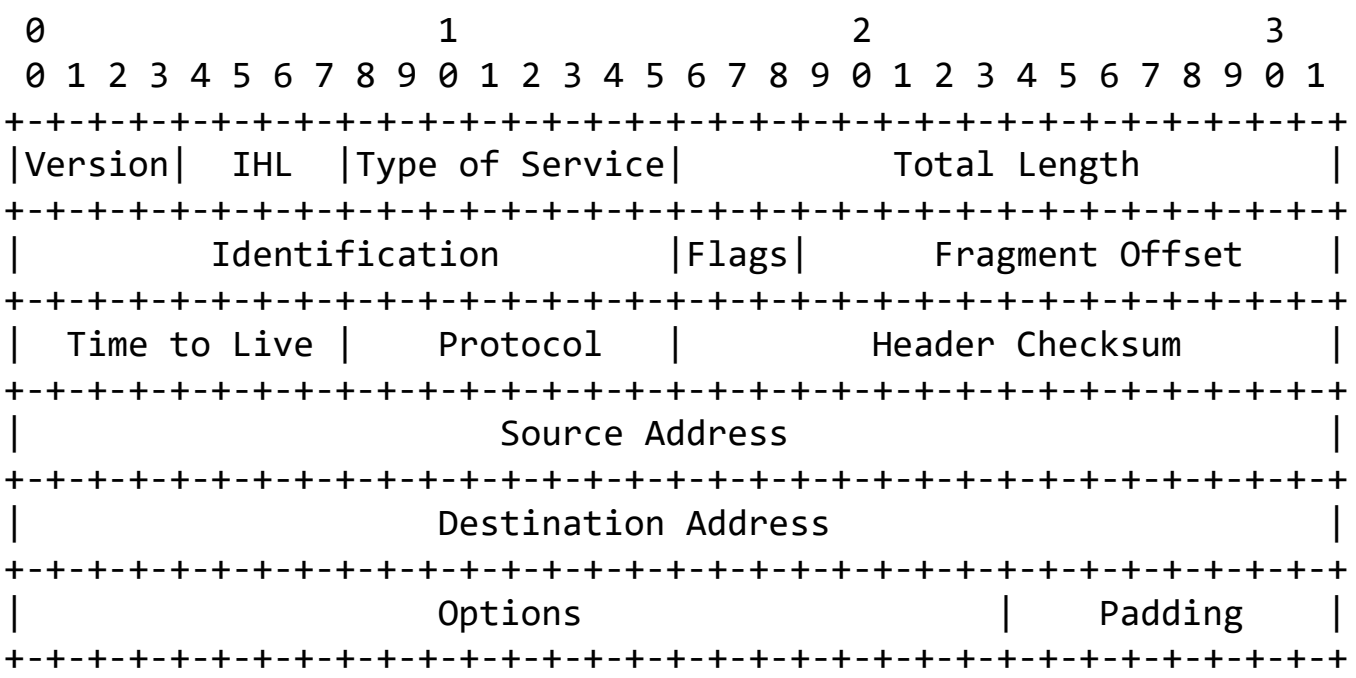
bit number

bit field
Payload Type (PT):
a 7-bit field that identifies
the format of the RTP payload



FORMATY WIADOMOŚCI: PRZYKŁADY (2/3)

- IPv4 (RFC 791)



Example Internet Datagram Header

Figure 4.

FORMATY WIADOMOŚCI: PRZYKŁADY (3/3)

- CoAP

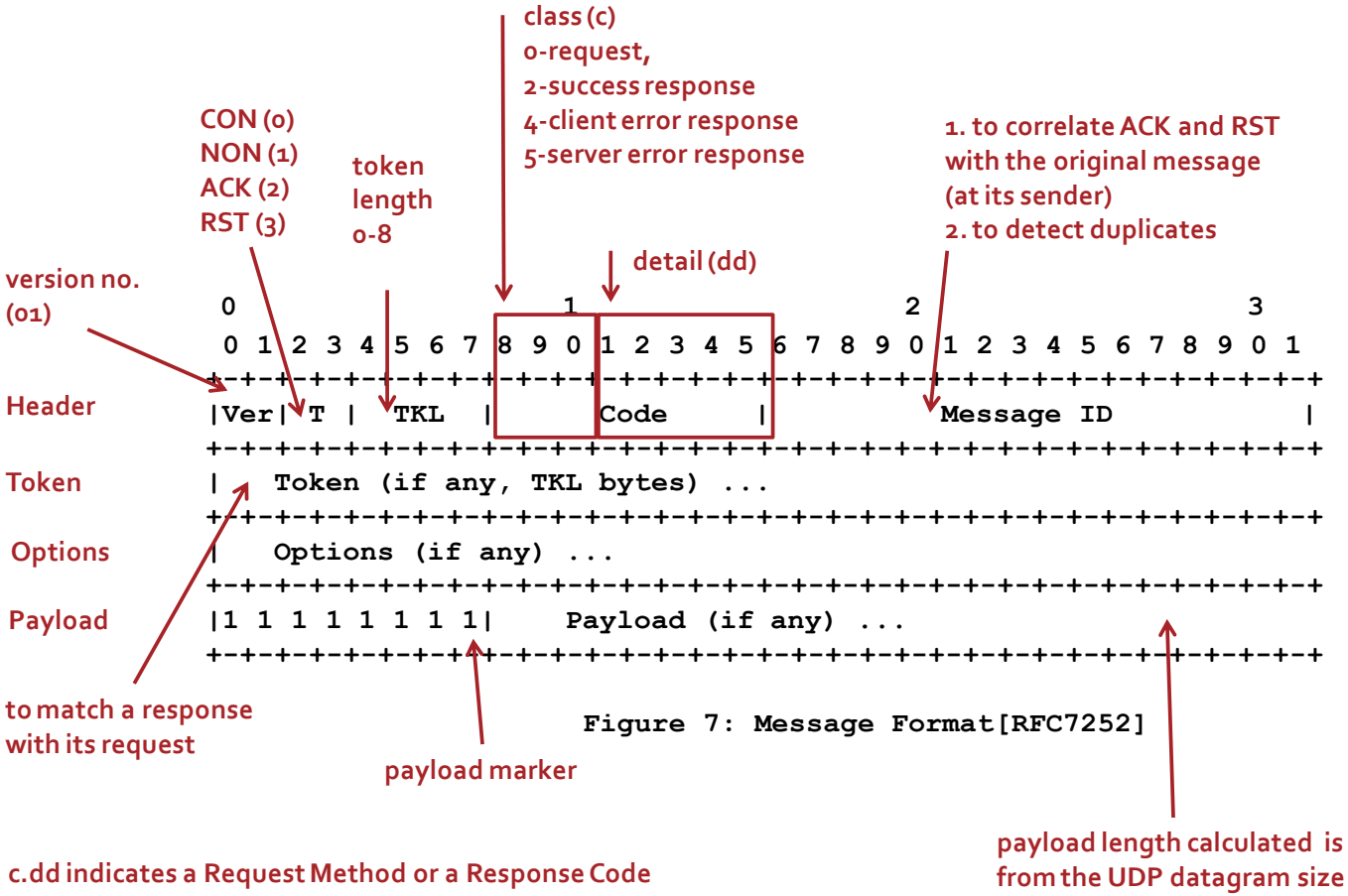


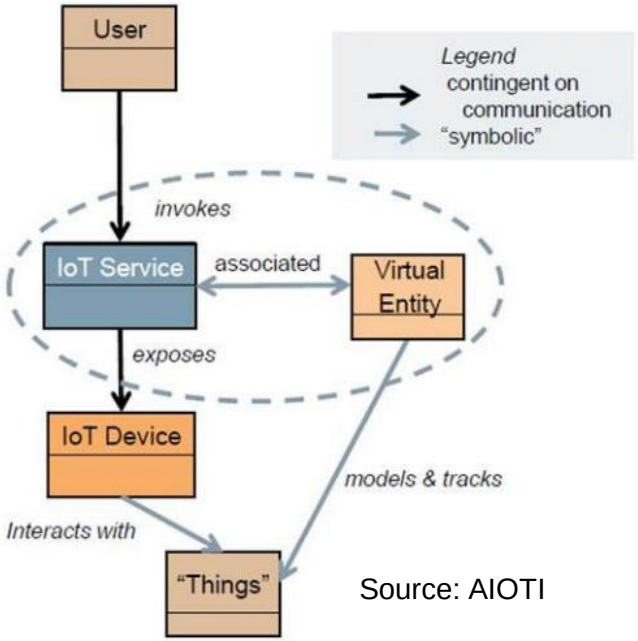
Figure 7: Message Format[RFC7252]

c.dd indicates a Request Method or a Response Code

0.00 Empty message
0.01 GET
0.02 POST
0.03 PUT
0.04 DELETE

2.dd success
4.dd client error
5.dd server error

Semantyczny opis zasobów IoT



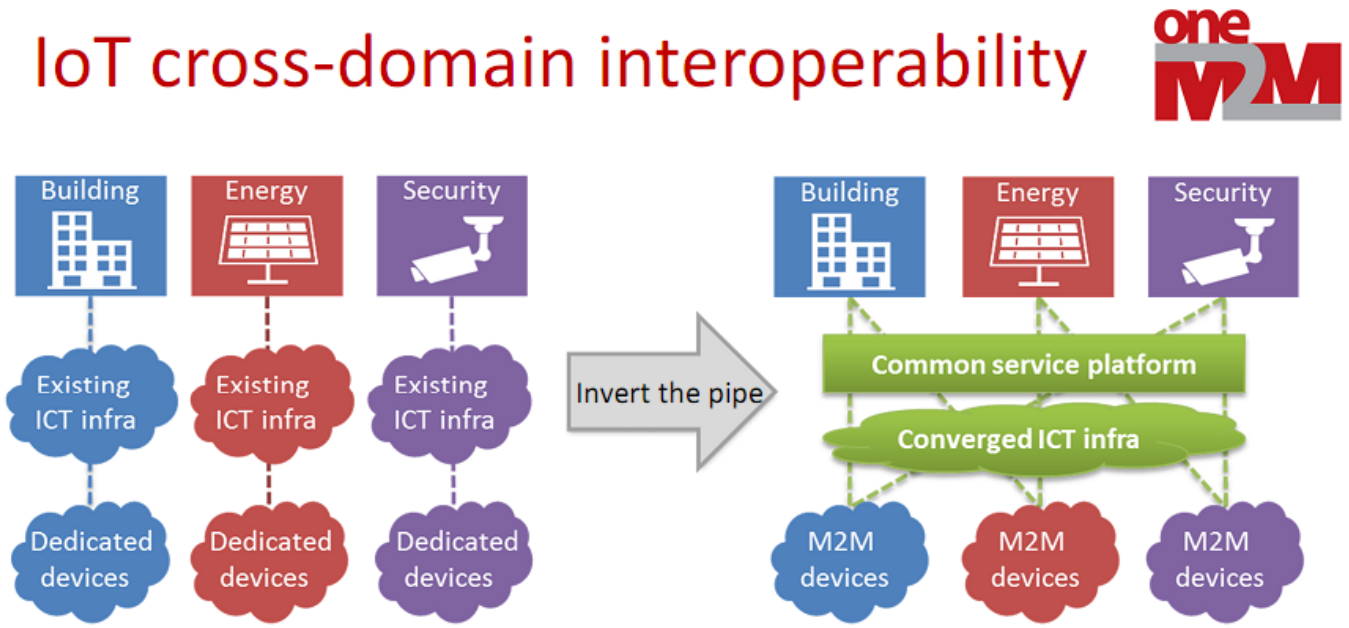
Source: AIOTI

Plan dalszego ciągu

- Wstęp: potrzeba opisu zasobów
- Stan bieżący: rozproszenie
- SenML
- ZigBee Cluster Library
- LWM2M, IPSO
- IoT-Lite

SEMANTYKA: CO TO DAJE?

Źródło: oneM2M



- Highly fragmented market with small vendor-specific applications.
- Reinventing the wheel: Same services developed again and again.
- Each silo contains its own technologies without interop.

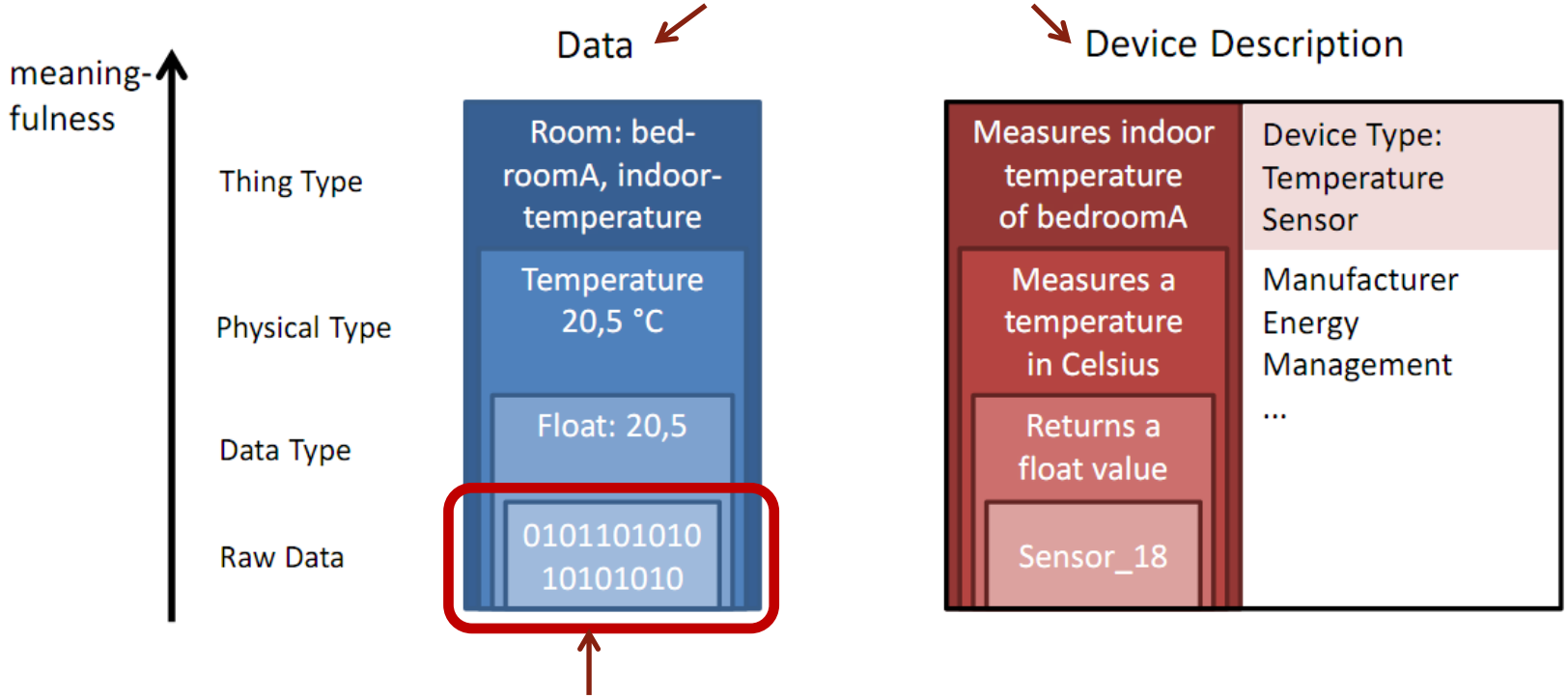
- End-to-end platform: common service capabilities layer.
- Interoperability at the level of communications and data.
- Seamless interaction between heterogeneous applications and devices.

Interoperacyjność między aplikacjami a węzłami pochodzącymi z różnych źródeł.
Reużywalność węzłów.
Żeby użyć – trzeba odkryć i zrozumieć jakie zasoby są oferowane.

SEMANTYKA: METADANE

Źródło::
Study of Abstraction and Semantics Enablements,
oneM2M, TR-0007-V2.11.1 , 2016

opis danych lub opis węzła – źródła danych (ten sam cel)



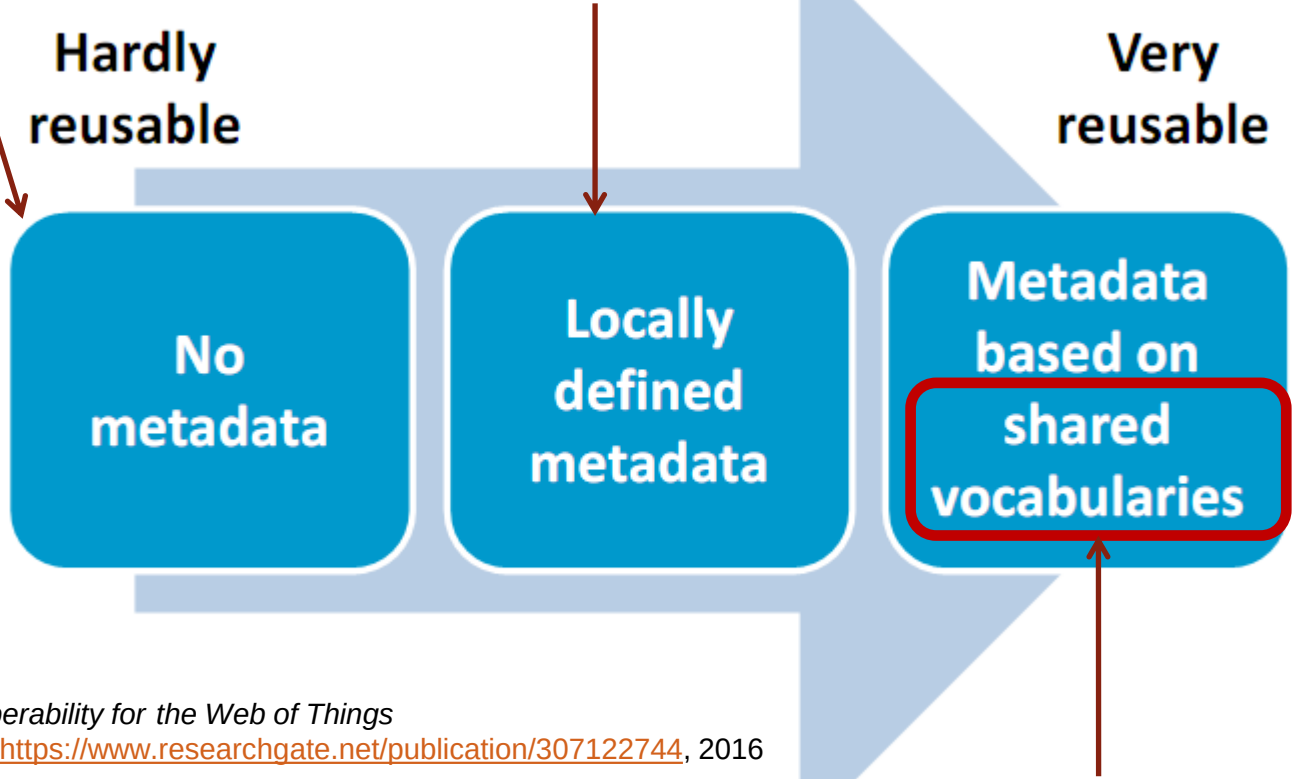
Jeśli pobieramy tylko „to", cała reszta musi być „zaszyta" w kodzie, a więc znana w fazie programowania (*design time*).
A gdzie odkrywanie w fazie *runtime*?
Co jeśli pojawiają się nowe sensory, już po uruchomieniu aplikacji?

PYTANIE: jak szeroko rozumiany powinien być język opisu zasobów?

SEMANTYKA: WSPÓLNE ROZUMIENIE METADANYCH

Wiedza o zasobach „zaszyta” w kodzie.

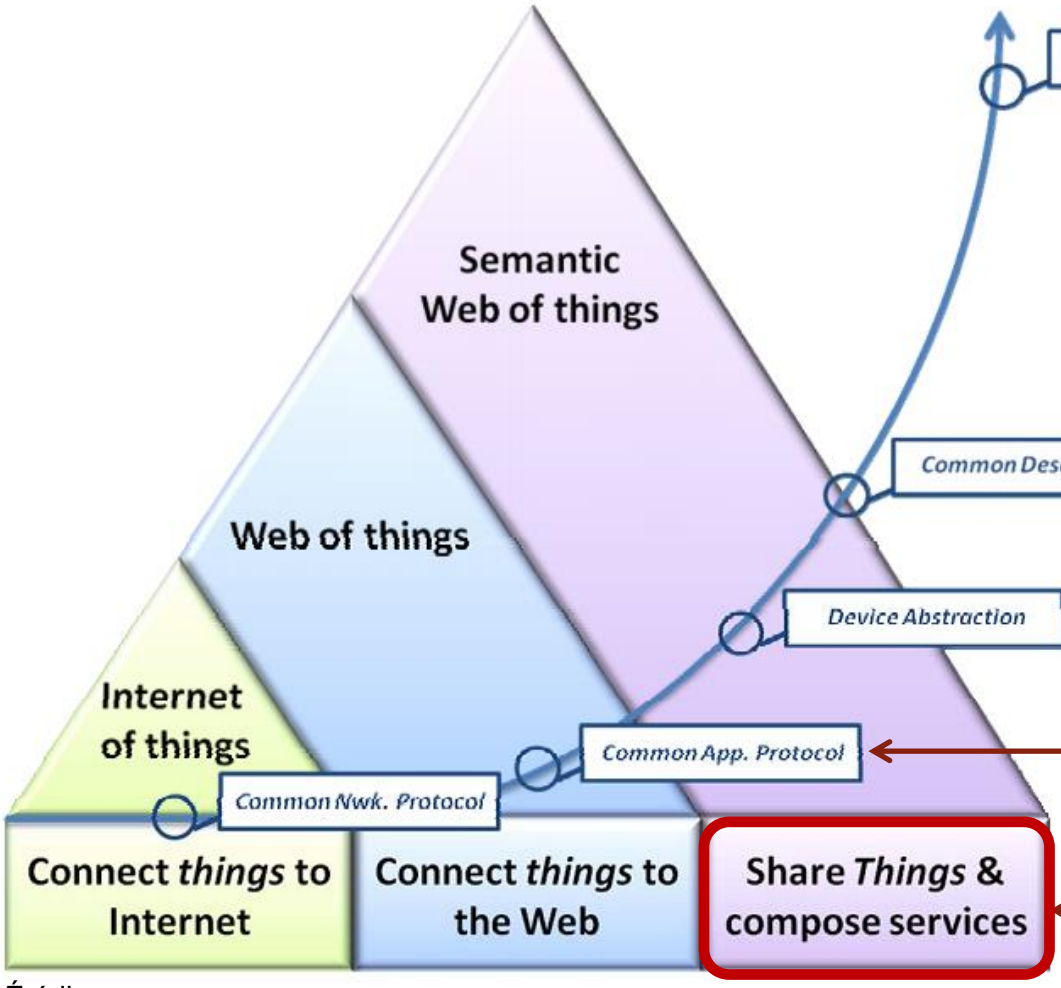
Język opisu zasobów rozumiany przez elementy (np. aplikacje, węzły), pochodzące z jednego źródła, np. jednej firmy (lokalne odkrywanie zasobów).



Źródło::
Semantic Intneroperability for the Web of Things
P. Murdock et al., https://www.researchgate.net/publication/307122744, 2016

Język opisu zasobów rozumiany przez elementy (np. aplikacje, węzły), pochodzące z różnych źródeł (globalne odkrywanie zasobów).

SEMANTYKA: POZIOMY INTEROPERACYJNOŚCI



wspólny język opisu zasobów

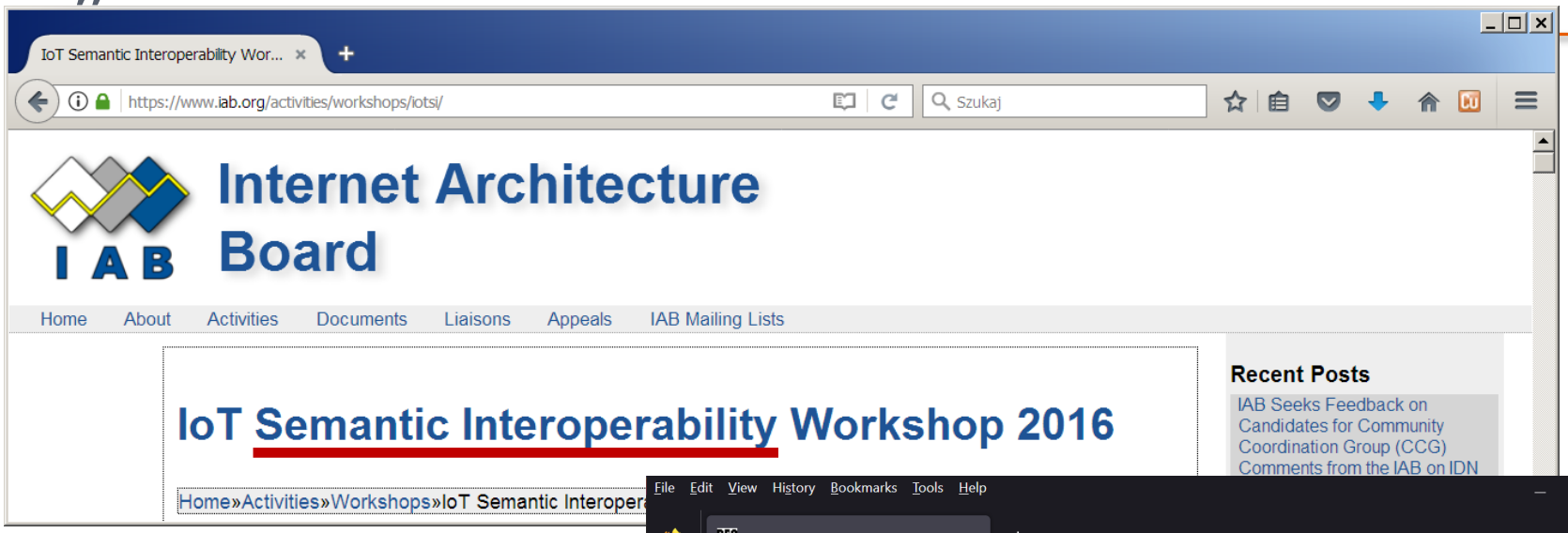
CoAP, …

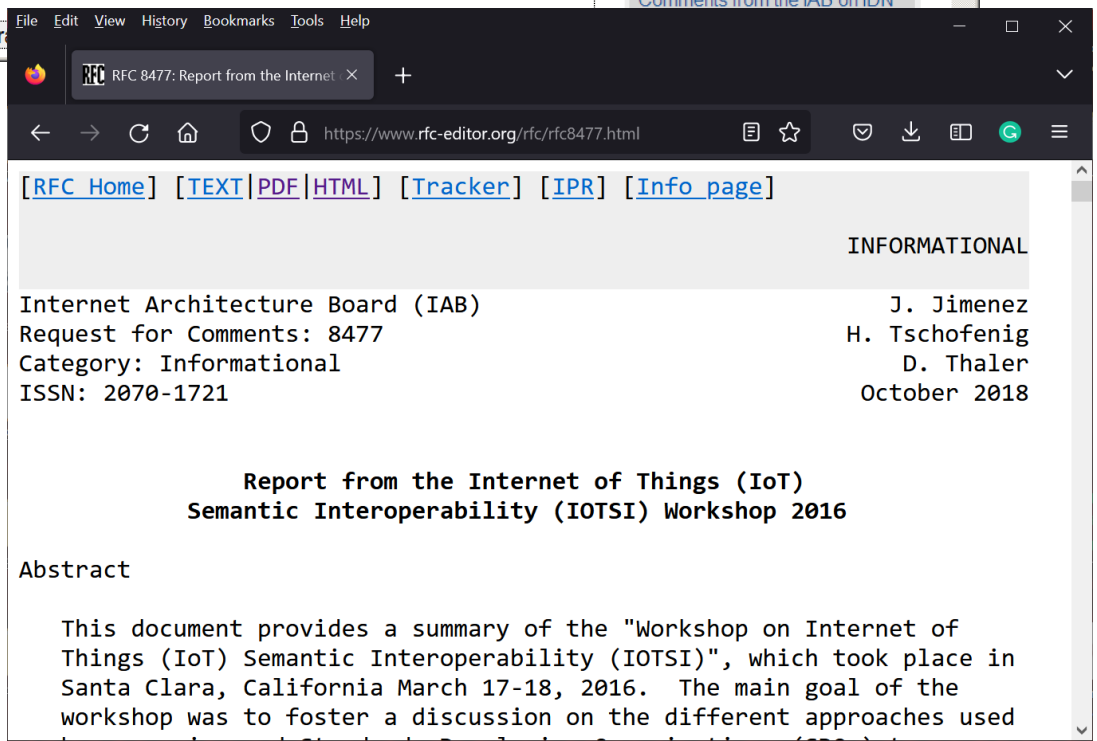
odkrywanie niezależnie udostępnianych zasobów i korzystanie z nich

Źródło::
Semantic Web of Things: an Analysis of the Application Semantics for the IoT Moving towards the IoT Convergence
A.J. Jara, Alex C. Olivieri, Y. Bocchi, M. Jung, W. Kastner, A. F. Skarmeta
Int. J. Web and Grid Services, Vol. 10, Nos. 2/3, 2014

Źródło:*Report from the Internet of Things (IoT) Semantic Interoperability (IOTSI) Workshop 2016,* RFC 8477, 2018

„INTEROPERABILITY AT THE APPLICATION LAYER”





Źródło:*Report from the Internet of Things (IoT) Semantic Interoperability (IOTSI) Workshop 2016,* RFC 8477, 2018

FROM IOTSI WORKSHOP REPORT:

”Increasing interoperability in the area of Internet of Things (IoT) has been a top priority for many standards organizations and particularly the lower layers of the Internet protocol stack have received a lot of attention.”

”Also at the application layer, such as with CoAP and HTTP, there is a trend in reusing RESTful design patterns. However, in data exchanged on top of these application layer protocols there is still a lot of fragmentation and the same degree of increase in interoperability has not been observed.”

Źródło:*Report from the Internet of Things (IoT) Semantic Interoperability (IOTSI) Workshop 2016,* RFC 8477, 2018

ROZPROSZENIE (ZE SPISU TREŚCI)

Appendix B: Accepted Position Papers

…

4. Robert Cragie, "The ZigBee Cluster Library over IP"
5. Dee Denteneer, Michael Verschoor, Teresa Zotti, "Fairhair: interoperable IoT services for major Building Automation and Lighting Control ecosystems"

…

12. Jaime Jimenez, Michael Koster, Hannes Tschofenig, "IPSO Smart Objects"

…

15. Achilleas Kemos, "Alliance for Internet of Things Innovation Semantic Interoperability Release 2.0, AIOTI WG03 - IoT Standardisation"
16. Ari Keraenen, Cullen Jennings, "SenML: simple building block for IoT semantic interoperability"

…

22. Marcello Lioy, "AllJoyn"
25. Open Geospatial Consortium, "OGC SensorThings API: Communicating "Where" in the Web of Things"

…

27. Joaquin Prado, "OMA Lightweight M2M Resource Model"

…

30. Jasper Roes, Laura Daniele, "Towards semantic interoperability in the IoT using the Smart Appliances REFerence ontology (SAREF) and its extensions„

…

35. J. Clarke Stevens, "Open Connectivity Foundation oneIoTa Tool"

…

37. Ravi Subramaniam, "Semantic Interoperability in Open Connectivity Foundation (OCF) - formerly Open Interconnect Consortium (OIC)""

…

41. Dave Thaler, Summary of AllSeen Alliance Work Relevant to Semantic Interoperability„

…

… a nie ma tu nawet wzmianki np. o ontologiach dla IoT, tworzonych przez środowisko R&D …

PROSTY MODEL DANYCH (NEST)

”locally defined metadata”

Źródło: https://nest.com

Źródło: https://developers.nest.com

Plik Edycja Widok Historia Zakładki Narzędzia Pomoc

API | Nest Developers

https://develop

Dictionary1 Dictionary2 Dictionary3 BGPW Stoa MEAG

Nest Developers GET STARTED


```
"thermostats": {
  "peyiJNo0IldT2YlIVtYaGQ": {
    "device_id": "peyiJNo0IldT2YlIVtYaGQ",
    "locale": "en-US",
    "software_version": "4.0",
    "structure_id": "VqFabWH21nwVyd4RWgJgN
    b292wa7hG_dUwo2i2SG7j3-BOLY0BA4sw",
    "name": "Hallway (upstairs)",
    "name_long": "Hallway Thermostat (upst
    airs)",
    "last_connection": "2016-10-31T23:59:5
    9.000Z",
    "is_online": true,
    "can_cool": true,
    "can_heat": true,
    "is_using_emergency_heat": true,
    "has_fan": true,
```


4 Eksplorator... [14LIV]A Survey... 3 Firefox Microsoft P

SenML (Sensor Measurement Lists)

Internet Engineering Task Force (IETF)
Request for Comments: 8428
Category: Standards Track
ISSN: 2070-1721

C. Jennings
Cisco
Z. Shelby
ARM
J. Arkko
A. Keranen
Ericsson
C. Bormann
Universitaet Bremen TZI

August 2018

Sensor Measurement Lists (SenML)

Abstract

This specification defines a format for representing simple sensor measurements and device parameters in Sensor Measurement Lists (SenML). Representations are defined in JavaScript Object Notation (JSON), Concise Binary Object Representation (CBOR), Extensible Markup Language (XML), and Efficient XML Interchange (EXI), which share the common SenML data model. A simple sensor, such as a temperature sensor, could use one of these media types in protocols such as HTTP or the Constrained Application Protocol (CoAP) to transport the measurements of the sensor or to be configured.

JSON, CBOR, XML, EXI: np. opcja `Content-Format` CoAPa; czyli ta opcja to za mało

SenML: records and packs

- The data is an array of SenML records containing attributes:
  - unique identifier for the sensor, the
  - time the measurement was made,
  - the unit the measurement is in, and
  - the current value of the sensor.
- SenML pack: an array of SenML records.

a single SenML record:

no timestamp: the measurement made roughly „now”

```
[
 {"n":"urn:dev:ow:10e2073a01080063","u":"Cel","v":23.1}
]
```

sensor identifier (name) tag

unit tag

value tag

Źródło: *Media Types for Sensor Measurement Lists (SenML),* draft-ietf-core-senml-07, Internet Draft, 2017

SenML: regular and base attributes

Źródło: *Media Types for Sensor Measurement Lists (SenML)*, draft-ietf-core-senml-07, Internet Draft, 2017

- Regular attributes
  - can be included in any SenML record
  - apply only to that record

timestamp tag (Unix epoch time)

```
[
{"n":"urn:dev:ow:10e2073a01080063","u":"Cel","t":1.276020076e+09,"v":23.5},
{"n":"urn:dev:ow:10e2073a01080063","u":"Cel","t":1.276020091e+09,"v":23.6}
]
```

same name in both records

- Base attributes
  - can only be included in the first SenML record
  - they apply to all records
  - all base attributes are optional

base name tag – prepend to name

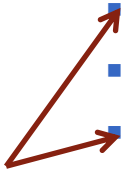
```
[
{"bn":"urn:dev:ow:10e2073a01080063","u":"Cel","t":1.276020076e+09,"v":23.5},
{"u":"Cel","t":1.276020091e+09,"v":23.6}
]
```

no ”n” tag – nothing to append to base name

SenML: regular and base attributes

Źródło: *Media Types for Sensor Measurement Lists (SenML)*, draft-ietf-core-senml-07, Internet Draft, 2017

- Regular
  - Name: the name of the sensor
    - when appended to the Base Name attribute, this must result in a globally unique identifier.
  - Unit: the unit for Value (from a registry of SenML unit symbols)
  - Value: the measurement value
    - "v" for float
    - "vb" for boolean
    - "vs" for string
    - "vd" for binary data.
  - Time: when Value was recorded
  - Update Time: the maximum time before this sensor will provide an updated Value. Can be used to detect failures.
- Base
  - Base Name: a string prepended to Name in a record (if any)
  - Base Time: a base time added to Time in a record (if any)
  - Base Unit: a base unit assumed for all records (unless a record contains Unit)
  - Base Value: a base value added to Value in a record



„kodowanie różnicowe"

SenML: content types and attributes

- Content types
  - application/senml+json
  - application/senml+cbor
  - application/senml+xml
  - application/senml+exi
- Attributes

Name	label	Type
Base Name	bn	String
Base Time	bt	Number
Base Unit	bu	String
Base Value	bv	Number
Base Sum	bs	Number
Version	bver	Number
Name	n	String
Unit	u	String
Value	v	Number
String Value	vs	String
Boolean Value	vb	Boolean
Data Value	vd	String
Value Sum	s	Number
Time	t	Number
Update Time	ut	Number
Link	l	String

Źródło: *Media Types for Sensor Measurement Lists (SenML),* draft-ietf-core-senml-07, Internet Draft, 2017

SenML: examples (1/3)

voltage and current (different sensor names), roughly now (no base time, no time)

```
[                                                    urn:dev:ow:10e2073a01080063:voltage
  {"bn":"urn:dev:ow:10e2073a01080063:","n":"voltage","u":"V","v":120.1},
  {"n":"current","u":"A","v":1.2}
]  urn:dev:ow:10e2073a01080063:current
```

voltage and current at Tue Jun 8 18:01:16.001 UTC 2010,
current at each second for the previous 5 seconds

```
[
      {"bn":"urn:dev:ow:10e2073a0108006:","bt":1.276020076001e+09,
       "bu":"A","n":"voltage","u":"V","v":120.1},
      {"n":"current","t":-5,"v":1.2},
      {"n":"current","t":-4,"v":1.3},
      {"n":"current","t":-3,"v":1.4},
      {"n":"current","t":-2,"v":1.5},
      {"n":"current","t":-1,"v":1.6},
      {"n":"current","v":1.7}
]
```

unit for this record is V (overrides base unit)

unit not specified, base unit (A) applies

base time – 1 second

time not specified, base time applies

Źródło: *Media Types for Sensor Measurement Lists (SenML),* draft-ietf-core-senml-07, Internet Draft, 2017

SENML: EXAMPLES (2/3)

relative humidity from a mobile device

```
[
     {"bn":"urn:dev:ow:10e2073a01080063","bt":1.320067464e+09,
      "bu":"%RH","v":20},
     {"u":"lon","v":24.30621},
     {"u":"lat","v":60.07965},
     {"t":60,"v":20.3},
     {"u":"lon","t":60,"v":24.30622},
     {"u":"lat","t":60,"v":60.07965},
     {"u":"%EL","t":150,"v":98}
]
```

relative humidity and location at base time

relative humidity and location, 60s later

remaining battery energy level 150s after base time

different value types

```
[
    {"bn":"urn:dev:ow:10e2073a01080063:","n":"temp","u":"Cel","v":23.1},
    {"n":"label","vs":"Machine Room"},
    {"n":"open","vb":false}
]
```

string

float

boolean

Źródło: *Media Types for Sensor Measurement Lists (SenML),* draft-ietf-core-senml-07, Internet Draft, 2017

SenML: examples (3/3)

setting actuators (the recipient should know that these are commands)

```
[
      {"bn":"urn:dev:ow:10e2073a01080063:"},
      {"n":"temp","u":"Cel","v":23.1},
      {"n":"heat","u":"/","v":1},
      {"n":"fan","u":"/","v":0}
 ]
```

set new thermostat setpoint

switch the heater on

switch the fan off

value of a switch

SenML unit symbols

full list is much longer:
https://www.iana.org/assignments/senml/senml.xhtml

. . .		
lat	degrees latitude (note 2)	float
lon	degrees longitude (note 2)	float
pH	pH value (acidity; logarithmic quantity)	float
dB	decibel (logarithmic quantity)	float
Bspl	bel (sound pressure level; logarithmic quantity)*	float
count	1 (counter value)	float
/	1 (Ratio e.g., value of a switch, note 1)	float
%	1 (Ratio e.g., value of a switch, note 1)*	float
%RH	Percentage (Relative Humidity)	float
%EL	Percentage (remaining battery energy level)	float
. . .		

Źródło: *Media Types for Sensor Measurement Lists (SenML)*, draft-ietf-core-senml-07, Internet Draft, 2017

ZigBee Cluster Library, ZCL

csa connectivity standards alliance

ZigBee® Alliance

Matter Application Cluster Specification

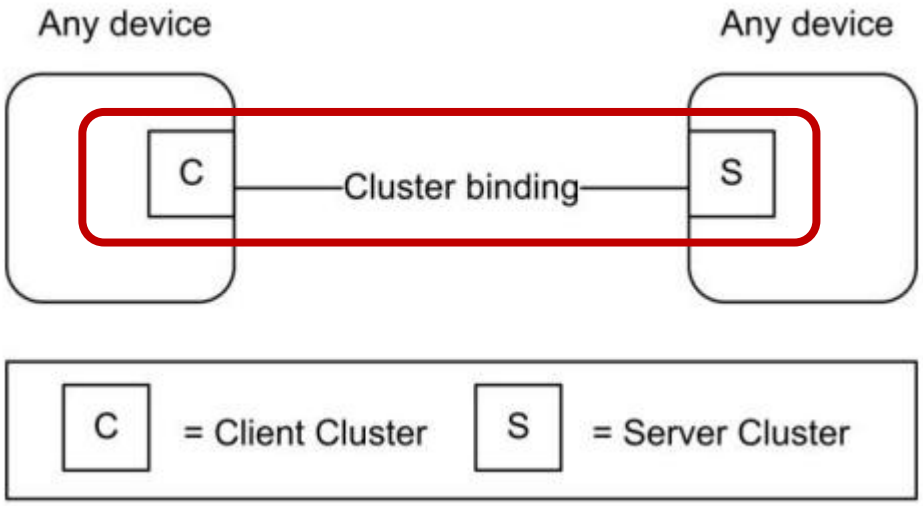
Version 1.0

Document: 22-22350-001_Matter-1.0-Application-Cluster-Specification.pdf

September 28, 2022

ZCL: CLUSTER

- CLUSTER a collection of related commands and attributes, which together define an interface to specific functionality
  - 16-bit cluster identifier
- CLIENT (OUTPUT) CLUSTER affects or manipulates the attributes
  - example: light switch
- SERVER (INPUT) CLUSTER stores the attributes
  - example: light
  - usually, most of the specification describes the server side



Źródło: *ZigBee Cluster Library Specification,* ZigBee Document: 075123 Revision 5, ZigBee Alliance, 2015

ZCL: CLUSTER ID, FUNCTIONAL DOMAINS

Functional Domain	Cluster ID Range
General	0x0000 – 0x00ff
Closures	0x0100 – 0x01ff
HVAC	0x0200 – 0x02ff
Lighting	0x0300 – 0x03ff
Measurement and sensing	0x0400 – 0x04ff
Security and safety	0x0500 – 0x05ff
Protocol interfaces & commercial building	0x0600 – 0x06ff
Energy	0x0700 – 0x07ff
Security credentials	0x0800 – 0x8ff
Telecom	0x0900 – 0x09ff

Cluster Identifier	Description
0x0000 – 0x7fff	Standard ZigBee cluster
0xfc00 – 0xffff	Manufacturer specific cluster within a standard ZigBee profile
all other values	Reserved

Źródło: *ZigBee Cluster Library Specification,* ZigBee Document: 075123 Revision 5, ZigBee Alliance, 2015

Źródło: *ZigBee Cluster Library Specification*, ZigBee Document: 075123 Revision 5, ZigBee Alliance, 2015

ZCL: FUNCTIONAL DOMAINS - GENERAL

Functional Domain	Cluster ID Range
General	0x0000 – 0x00ff
Closures	0x0100 – 0x01ff
HVAC	0x0200 – 0x02ff
Lighting	0x0300 – 0x03ff
Measurement and sensing	0x0400 – 0x04ff
Security and safety	0x0500 – 0x05ff
Protocol interfaces & commercial building	0x0600 – 0x06ff
Energy	0x0700 – 0x07ff
Security credentials	0x0800 – 0x8ff
Telecom	0x0900 – 0x09ff

clusters in the General Functional Domain (selection)

Cluster ID	Cluster Name	device configuration and installation Description
0x0000	Basic	Attributes for determining basic information about a device, setting user device information such as description of location, and enabling a device.
0x0001	Power configura-tion	Attributes for determining more detailed information about a device's power source(s), and for configuring under/over voltage alarms.
0x0002	Device temperature configuration	Attributes for determining information about a device's internal temperature, and for configuring under/over temperature alarms.
0x0003	Identify	Attributes and commands for putting a device into Identifica-tion mode (e.g., flashing a light)

Cluster ID	Cluster Name	on/off and level control Description
0x0006	On/off	Attributes and commands for switching devices between 'On' and 'Off' states.
0x0007	On/off switch configu-ration	Attributes and commands for configuring on/off switching devices
0x0008	Level control	Attributes and commands for controlling a characteristic of devices that can be set to a level between fully 'On' and fully 'Off'.

ZCL: Functional domains - Closures

Functional Domain	Cluster ID Range
General	0x0000 – 0x00ff
Closures	0x0100 – 0x01ff
HVAC	0x0200 – 0x02ff
Lighting	0x0300 – 0x03ff

Źródło: *ZigBee Cluster Library Specification*, ZigBee Document: 075123 Revision 5, ZigBee Alliance, 2015

clusters in the Closures Functional Domain

Cluster ID	Cluster Name	Description
0x0100	Shade Configuration	Attributes and commands for configuring a shade
0x0101	Door Lock	An interface to a generic way to secure a door
0x0102	Window Covering	Commands and attributes for controlling a window covering

ZCL: Cluster attributes and commands

- ATTRIBUTES: data items or states within a cluster
  - Each has and attribute ID (each cluster has its own attribute ID's).
  - They are a bit like resources.
  - They are maintained by the server.
  - Each has a type.
  - Some may be optional.

attribute ID within a cluster →

Attribute Identifier	Description
0x0000 – 0x4fff	Standard ZigBee attribute
0xf000 – 0xfffe	Global Attributes
all other values	Reserved

Źródło: *ZigBee Cluster Library Specification*, ZigBee Document: 075123 Revision 5, ZigBee Alliance, 2015

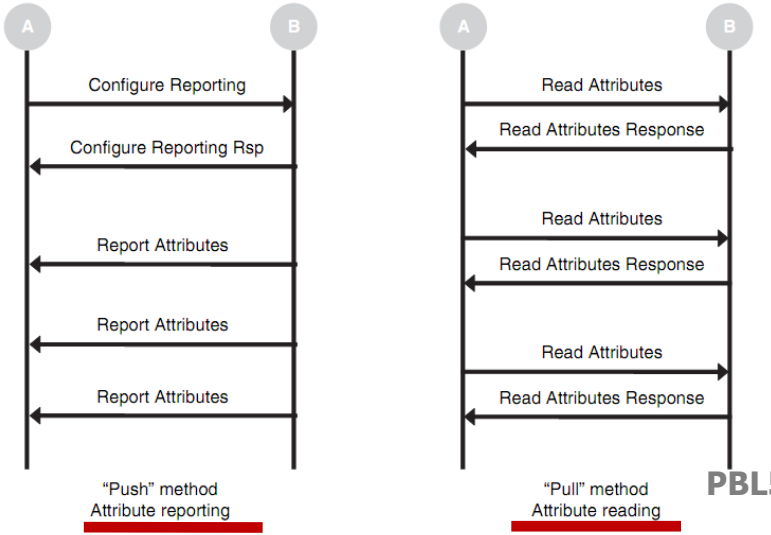
- COMMANDS: actions the cluster must perform
  - Each has a command ID.
  - Cross-cluster commands.
  - Cluster-specific commands.
  - Commands go both ways (client-to-server and server-to-client).

- A cluster is a bit like an object (in OOP), with methods (commands) and data (attributes).
  - but no inheritance

ZCL: Cross-cluster commands

Źródło: D. Gislason *ZigBee Wireless Networking*, Newnes, 2008

- In all clusters
- Used to handle attributes:
  - read
  - write
  - report
  - discover (esp. optional ones)
- Reporting (notifications)
  - value has changed
  - time-based

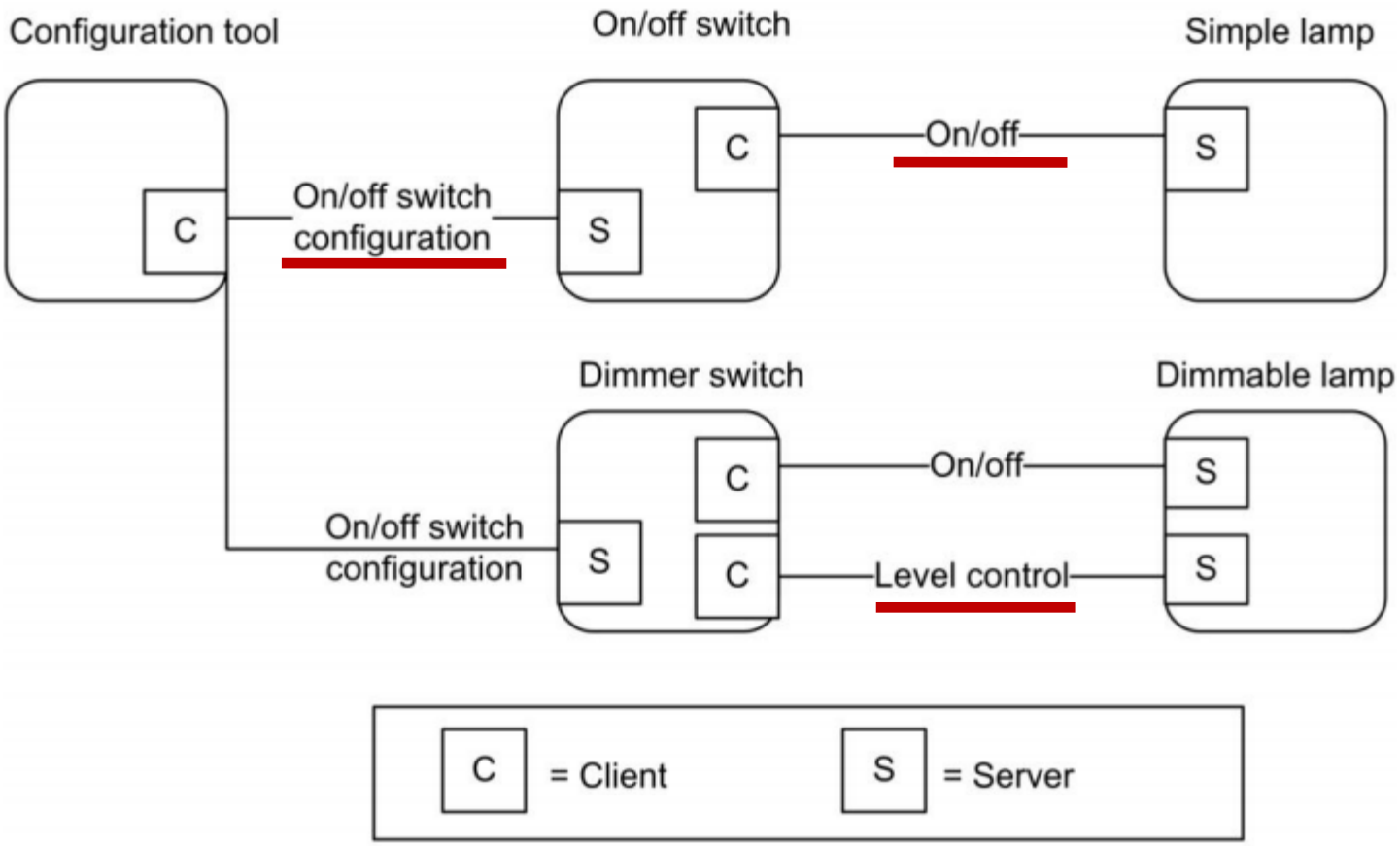


Źródło: D. Gislason *ZigBee Wireless Networking*, Newnes, 2008

cross-cluster command ID's

Cmd Id	Command Name	M/O	Description
0x00	Read attributes	O	Read one or more attributes
0x01	Read attributes response	M	Return value of one or more attributes
0x02	Write attributes	O	Write one or more attributes
0x03	Write attributes undivided	O	Write one or more attributes as a set
0x04	Write attributes response	M	Return success status of write attributes
0x05	Write attributes no response	M	Write one or more attributes, no response
0x06	Configure reporting	O	Configure attributes for reporting
0x07	Configure reporting response	M	Status of configure attributes
0x08	Read reporting configuration	O	Read current reporting configuration
0x09	Read reporting configuration response	M	Return current reporting configuration
0x0a	Report attributes	O	Attribute report, depends on configuration
0x0b	Default response	M	Unsupported command response
0x0c	Discover attributes	O	Determine supported attributes on remote node
0x0d	Discover attributes response	M	Results of discover attributes command
0x0e-0xff	Reserved		For future use by ZCL

ZCL: EXAMPLE

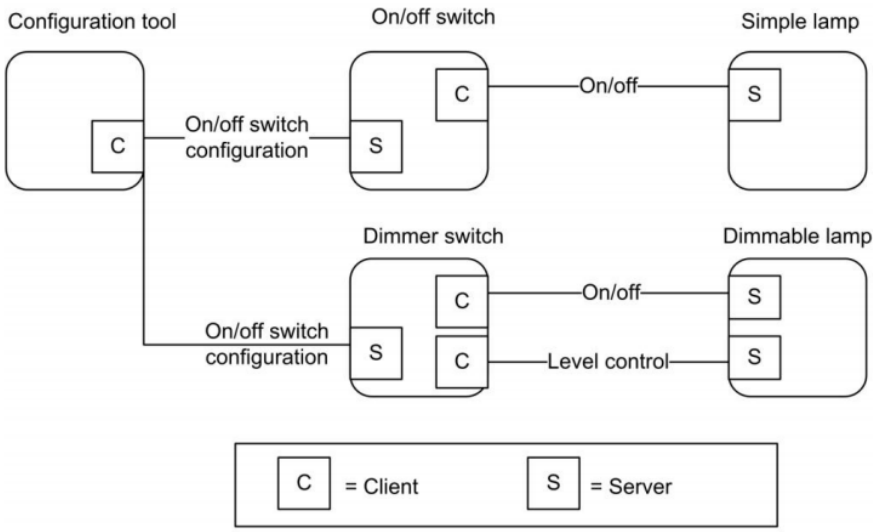


Note: these could be different devices (clusters are meant to be reusable).

Źródło: *ZigBee Cluster Library Specification,*
ZigBee Document: 075123 Revision 5,
ZigBee Alliance, 2015

ZCL: EXAMPLE

Źródło: *ZigBee Cluster Library Specification,*
ZigBee Document: 075123 Revision 5,
ZigBee Alliance, 2015



Cluster ID	Cluster Name	Description
0x0006	On/off	Attributes and commands for switching devices between 'On' and 'Off' states. need not be a lamp
0x0007	On/off switch configu-ration	Attributes and commands for configuring on/off switching devices
0x0008	Level control	Attributes and commands for controlling a characteristic of devices that can be set to a level between fully 'On' and fully 'Off'.

Cluster ID = 0x0006

ZCL: ON/OFF CLUSTER

"Attributes and commands for switching devices between 'On' and 'Off' states. "

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	*OnOff*	Boolean	0x00 – 0x01	Read only	0x00	M
0x4000	*GlobalScene-Control*	Boolean	TRUE or FALSE	Read only	TRUE	O
0x4001	*OnTime*	Unsigned 16-bit integer	0x0000 – 0xffff	Read/write	0x0000	O
0x4002	*OffWaitTime*	Unsigned 16-bit integer	0x0000 – 0xffff	Read/write	0x0000	O

attributes (server side only)

used with "On with timed off" command

keep Off state at least this long

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Off	M
0x01	On	M
0x02	Toggle	M
0x40	Off with effect	O
0x41	On with recall global scene	O
0x42	On with timed off	O

commands (server side only)

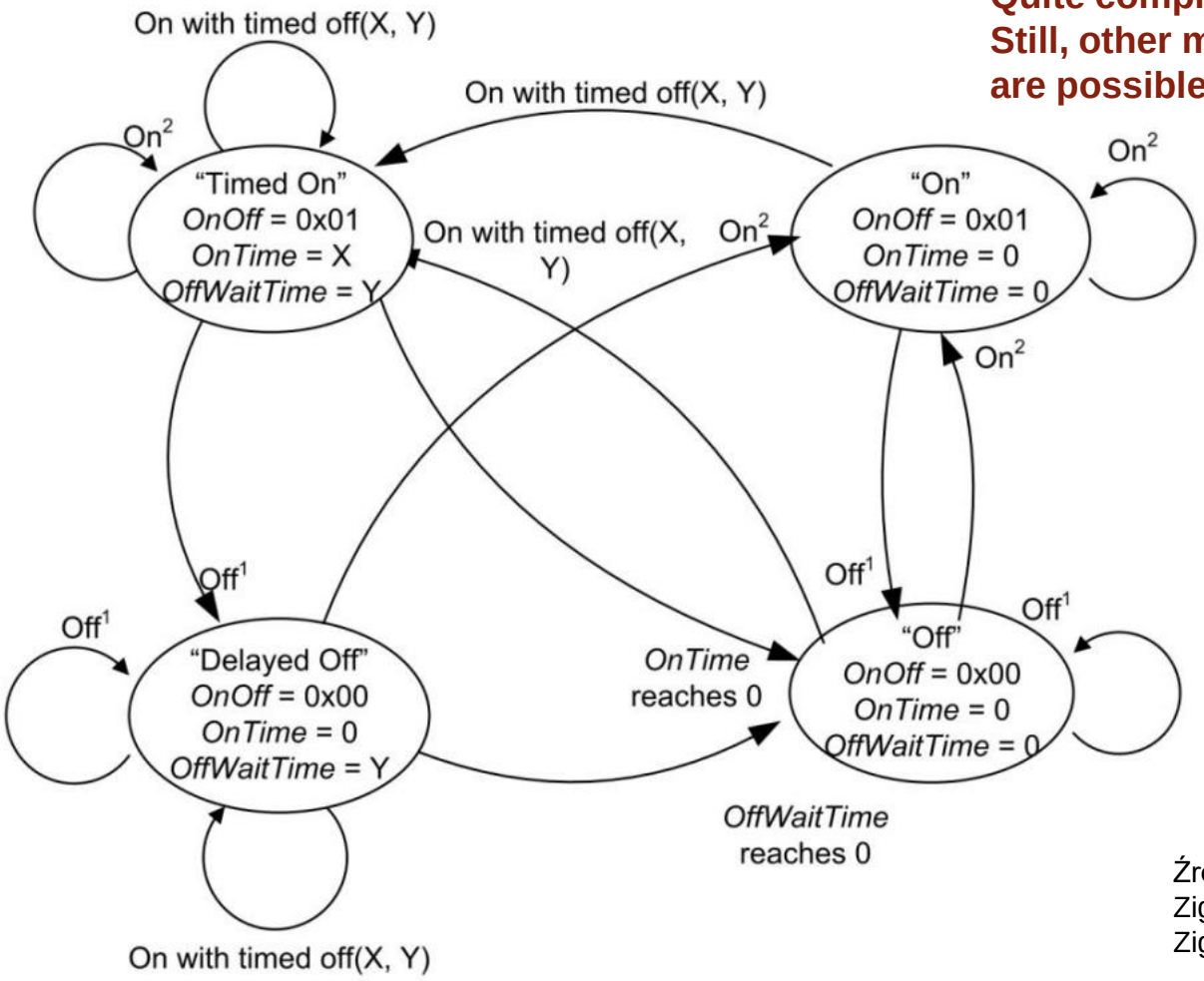
optional items make the cluster quite complex

e.g., „dying light" (fading effect)

Źródło: *ZigBee Cluster Library Specification*, ZigBee Document: 075123 Revision 5, ZigBee Alliance, 2015

ZCL: On/Off Cluster

Quite complex!
Still, other models of on/off functionality are possible.



Źródło: *ZigBee Cluster Library Specification,* ZigBee Document: 075123 Revision 5, ZigBee Alliance, 2015

Note 1: Any command which causes the *OnOff* attribute to be set to 0x00, e.g. Off, Toggle or Off with effect.
Note 2: Any command which causes the *OnOff* attribute to be set to 0x01, e.g. On, Toogle or On with recall global scene.

Cluster ID = 0x0402

”Attributes and commands for configuring the measurement of temperature, and reporting temperature measurements ”

ZCL: TEMPERATURE MEASUREMENT CLUSTER

Źródło: *ZigBee Cluster Library Specification,*
ZigBee Document: 075123 Revision 5,
ZigBee Alliance, 2015

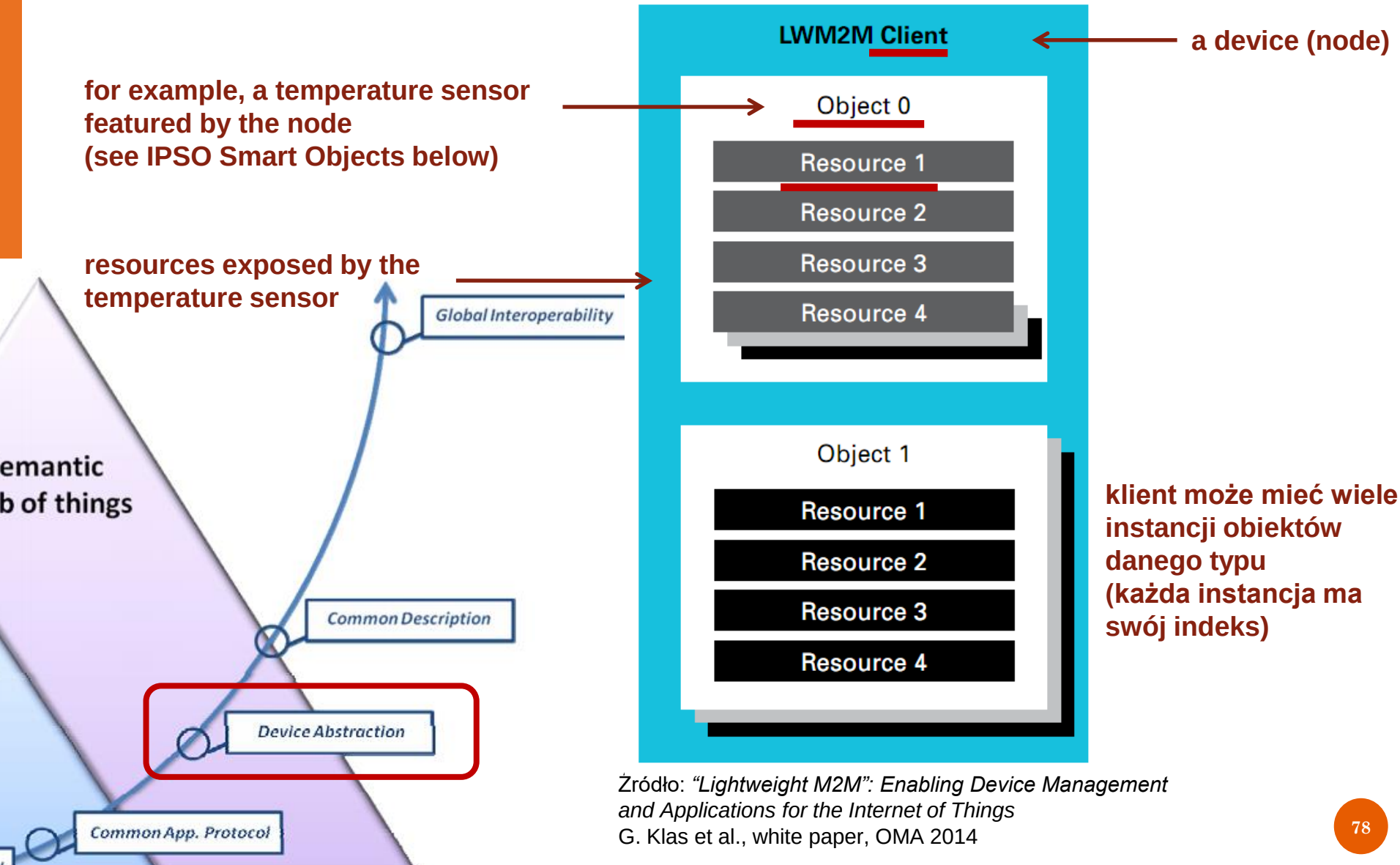
attributes (server side only), no commands

MeasuredValue = 100 x temperature in degrees Celsius, where -273.15°C <= temperature <= 327.67 ºC

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	*MeasuredValue*	Signed 16-bit integer	*MinMeasuredValue to MaxMeasuredValue*	Read only	0	M
0x0001	*MinMeasuredValue* min. possible	Signed 16-bit integer	0x954d – 0x7ffe	Read only	-	M
0x0002	*MaxMeasuredValue* max. possible	Signed 16-bit integer	0x954e – 0x7fff	Read only	-	M
0x0003	*Tolerance*	Unsigned 16-bit integer	0x0000 – 0x0800	Read only	-	O

”This cluster shall support attribute reporting …”

Model obiektu LWM2M



Źródło: *"Lightweight M2M": Enabling Device Management and Applications for the Internet of Things*
G. Klas et al., white paper, OMA 2014

IPSO Smart Objects (1/4)

Object Info:

IPSO używa modelu obiektu LWM2M

Object	Object ID	Object URN	Multiple Instances?	Description
IPSO Temperature	3303	urn:oma:lwm2m:ext:3303	Yes	Temperature sensor, example units = Cel

Źródło: *IPSO SmartObject Guideline. Smart Objects Starter Pack1.0*, IPSO Alliance, 2014

indeks instancji obiektu IPSO Temperature

Resources:

Resource Name	Resource ID	Access Type	Multiple Instances?	Mandatory	Type	Range or Enumeration	Units	Descriptions
Sensor Value	5700	R	No	Mandatory	Float			Last or Current Measured Value from the Sensor
Units	5701	R	No	Optional	String			Measurement Units Definition e.g. "Cel" for Temperature in Celsius.
Min Measured Value	5601	R	No	Optional	Float	Same as Measured Value	Same as Measured Value	The minimum value measured by the sensor since power ON or reset
Max Measured Value	5602	R	No	Optional	Float	Same as Measured Value	Same as Measured Value	The maximum value measured by the sensor since power ON or reset

URIs:
/3303/0/5700 → Sensor Value
/3303/0/5701 → Units
/3303/0/5601 → Min Measured Value
/3303/0/5602 → Max Measured Value

GET

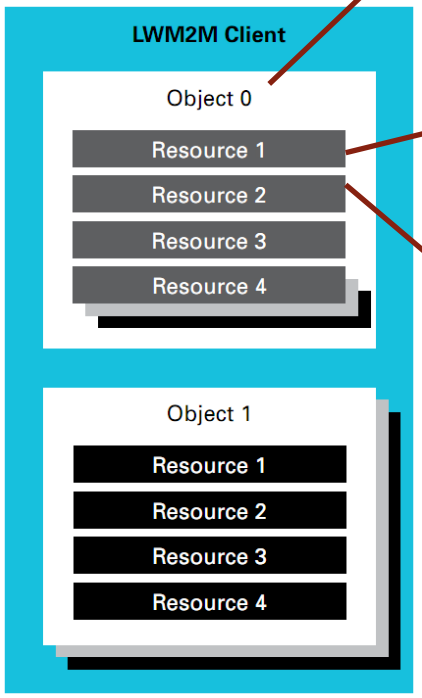
IPSO Smart Objects (2/4)

IPSO uses the OMA LWM2M object model

Object Info:

Object	Object ID	Object URN	Multiple Instances?	Description
IPSO Temperature	3303	urn:oma:lwm2m:ext:3303	Yes	Temperature sensor example units = Cel

Source: *IPSO SmartObject Guideline. Smart Objects Starter Pack1.0*, IPSO Alliance, 2014



Resources:

Resource Name	Resource ID	Access Type	Multiple Instances?	Mandatory	Type	Range or Enumeration	Units	Descriptions
Sensor Value	5700	R	No	Mandatory	Float			Last or Current Measured Value from the Sensor
Units	5701	R	No	Optional	String			Measurement Units Definition e.g. "Cel" for Temperature in Celsius.
Min Measured Value	5601	R	No	Optional	Float	Same as Measured Value	Same as Measured Value	The minimum value measured by the sensor since power ON or reset
Max Measured Value	5602	R	No	Optional	Float	Same as Measured Value	Same as Measured Value	The maximum value measured by the sensor since power ON or reset

Source: *"Lightweight M2M": Enabling Device Management and Applications f or the Internet of Things*
G. Klas et al., white paper, OMA 2014

IPSO Smart Objects (3/4)

Object Info:

IPSO używa modelu obiektu LWM2M

Object	Object ID	Object URN	Multiple Instances?	Description
IPSO Temperature	3303	urn:oma:lwm2m:ext:3303	Yes	Temperature sensor, example units = Cel

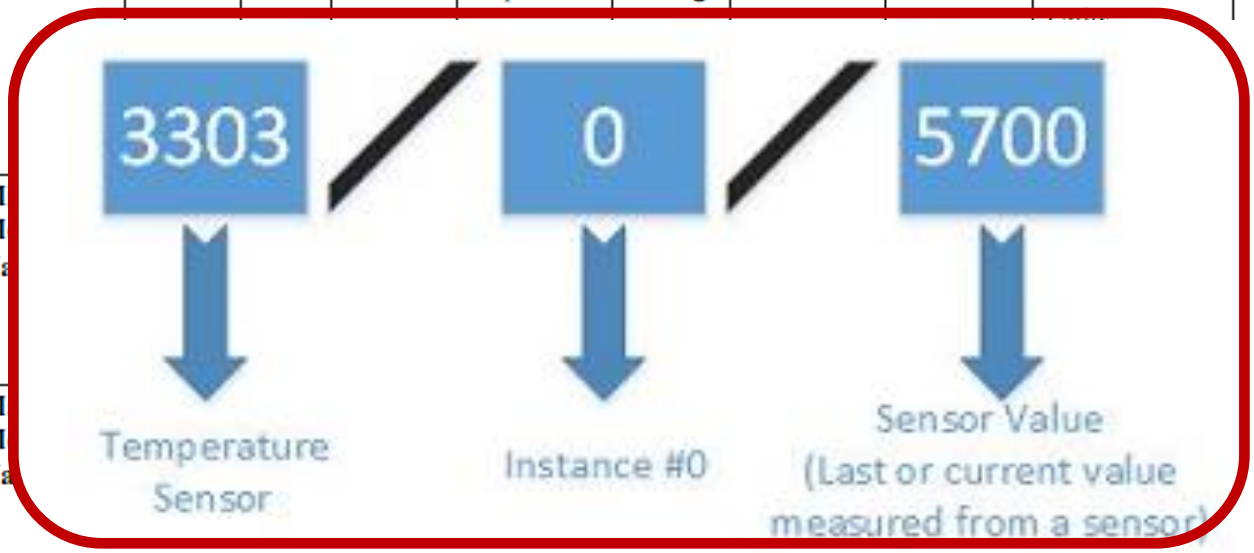
Źródło: *IPSO SmartObject Guideline. Smart Objects Starter Pack1.0*, IPSO Alliance, 2014

Resources:

Resource Name	Resource ID	Access Type	Multiple Instances?	Mandatory	Type	Range or Enumeration	Units	Descriptions
Sensor Value	5700	R	No	Mandatory	Float			Last or Current Measured Value from the Sensor
Units	5701	R	No	Optional	String			Measurement

URIs:
/3303/0/5700
/3303/0/5701
/3303/0/5601
/3303/0/5602

GET



Source: *IPSO Smart Objects,* J. Jimenez, M. Koster, H. Tschofenig, IOT Semantic Interoperability Workshop, 2016

IPSO Smart Objects (4/4)

IPSO Starter Pack.

Object	Object ID
Digital	3200
Digital Output	3201
Analogue Input	3202
Analogue Output	3203
Generic Sensor	3300
Illuminance Sensor	3301
Presence Sensor	3302
Temperature Sensor	3303
Humidity Sensor	3304
Power Measurement	3305
Actuation	3306
Set Point	3308
Load Control	3310
Light Control	3311
Power Control	3312
Accelerometer	3313
Magnetometer	3314
Barometer	3315

Source:
IPSO Smart Objects
J. Jimenez, M. Koster, H. Tschofenig
IOT Semantic Interoperability Workshop, 2016

ONTOLOGIE: IOT-LITE

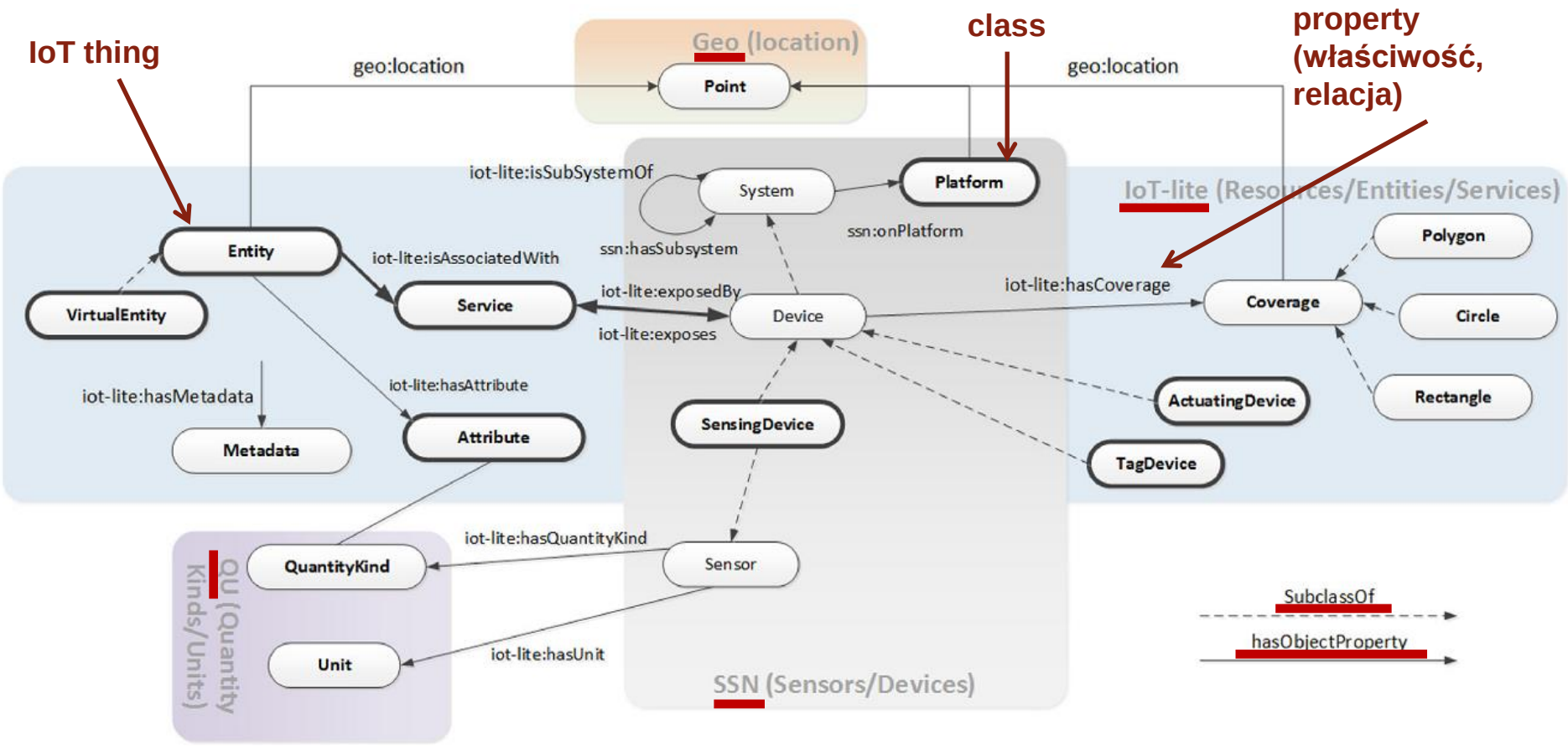


Fig. 1 An overview of the proposed semantic model, IoT-Lite

Source: *IoT-Lite: a lightweight semantic model for the internet of things and its use with dynamic semantics*
M. Bermudez-Edo *et al.,* Personal and Ubiquitous Computing, published online February 2017

IoT-Lite: reprezentacja sensora

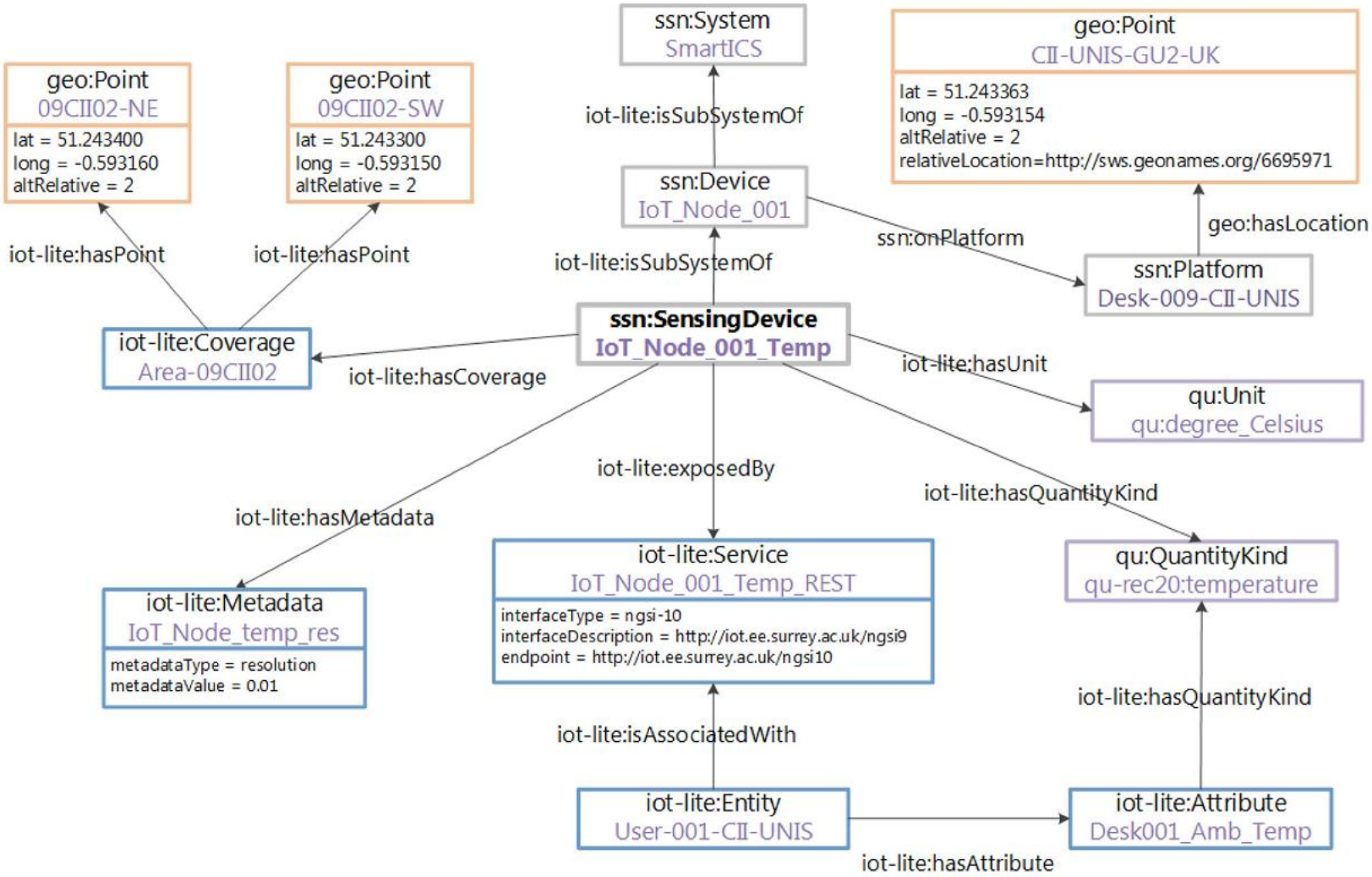


Fig. 3 An example of a sensor annotated with the proposed IoT-Lite ontology

Source: *IoT-Lite: a lightweight semantic model for the internet of things and its use with dynamic semantics*
M. Bermudez-Edo *et al.,* Personal and Ubiquitous Computing, published online February 2017

PBIS, 2022

IoT-Lite: reprezentacja (innego) sensora

This is RDF Turtle notation.

```
@prefix qu: <http://purl.org/NET/ssnx/qu/qu#> .
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix ssn: <http://www.w3.org/2005/Incubator/ssn/ssnx/ssn#> .
@prefix iot-lite:http://purl.oclc.org/NET/UNIS/iot-lite/iot-lite#

:temperatureSensorRoom13CII01 rdf:type owl:NamedIndividual ,
ssn:Sensor ;

iot-lite:type "SensorTelosB"^^xsd:string ;
iot-lite:id "telosB-001"^^xsd:string ;
geo:hasLocation :locationRoom13CII01 ;
iot-lite:exposedBy :ngsi10SensorRoom13CII01 ;
iot-lite:hasMetadata :resolution1024 ;
iot-lite:hasUnit qu:degree_Celsius ;
iot-lite:hasCoverage :areaRoom13CII01 ;
iot-lite:hasQuantityKind qu:temperature .
```

prefiksy (namespaces)

RDF triple (trójka RDF)

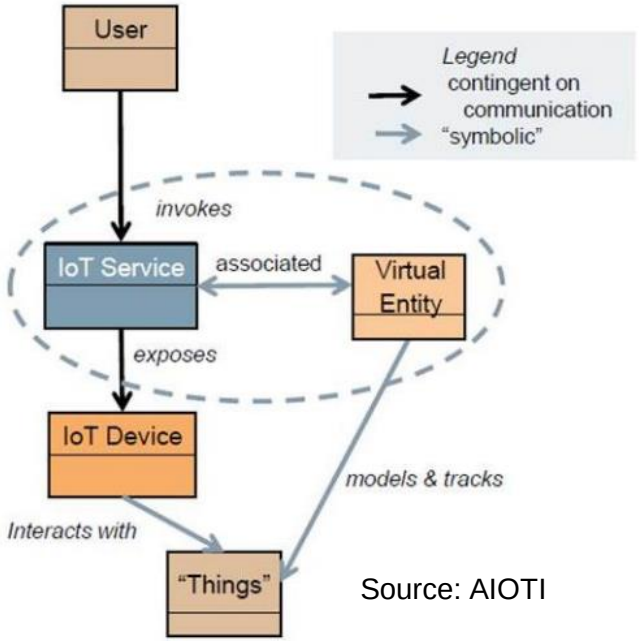
dopełnienie (object)

orzeczenie (predicate)

podmiot (subject)

Source: *IoT-Lite: a lightweight semantic model for the internet of things and its use with dynamic semantics*
M. Bermudez-Edo *et al.,* Personal and Ubiquitous Computing, published online February 2017

Dziękujemy za uwagę!



Source: AIOTI