





How to Make IoT Sensitive to Privacy? An Approach Based on ODRL and Illustrated With WoT TD

Zakaria Maamar¹ ^a, Amel Benna² ^b, Yang Xu³ ^c, Mohamed Adel Serhani⁴ ^d,
Minglin Li³, Huiru Huang³, Wassim Benadjel⁵ and Nacereddine Sitouah⁶

¹College of Computing and IT, University of Doha for Science and Technology, Doha, Qatar

²Department of Multimedia and Information Systems, CERIST, Algiers, Algeria

³School of Computer Science, Fudan University, Shanghai, China

⁴College of Computing and Informatics, University of Sharjah, Sharjah, U.A.E.

⁵Computer Science Faculty, USTHB, Algiers, Algeria

⁶Department of Electronics, Information and Bioengineering, Polytechnic University of Milan, Milan, Italy

Keywords: Internet-of-Things, ODRL, Privacy, WoT TD.

Abstract: Despite the multiple advantages of the Internet-of-Things (IoT), many users are still skeptical considering IoT as another disruptive information and communication technology that is “silently” invading their privacy with the risk of having their habits, preferences, and choices exposed publicly. To mitigate this silent invasion, this paper looks into innovative ways of making IoT sensitive to privacy by first, allowing users to explicitly express what is permitted, forbidden, and obliged over their personal data using the Open Digital Rights Language (ODRL), and second, adjusting things’ specifications like the Web-of-Things Thing Description (WoT TD), so that things would act according to these users’ permissions, prohibitions, and obligations. A system implementing and demonstrating the blend of ODRL with WoT TD is presented based on a case study capturing privacy concerns in a center for elderly people.


1 INTRODUCTION


Despite the tremendous advances in Information and Communication Technologies (ICTs), Privacy remains one of the persistent concerns that constitutes a major obstacle to the expansion and adoption of many ICTs (Silva et al., 2021). Indeed, poor handling of privacy could have severe consequences on both persons and businesses. To avoid similar situations and mitigate their consequences, many initiatives exist with focus, in this paper, on Open Digital Rights Language ODRL. ODRL is used to express that “*something is permitted, forbidden, or obliged, possibly limited by some constraints*” (W3C, 2018).


To exemplify ICTs that are revolutionizing the way contextualized services are provisioned to users, we consider the Internet-of-Things (IoT), (Gupta and


Quamara, 2020)). From autonomous farming equipment to driverless vehicles and wearable health monitors, IoT’s benefits and uses are endless. It is predicted that 41 billion IoT devices will exist by 2027 and that 70% of vehicles will be connected to the Internet by 2023¹. However, despite all the benefits and uses of IoT, privacy will slow down IoT’s “silent” invasion into people’s lives (Mohanta et al., 2021). Whether visible or invisible, things collect details about people’s (and even about other things) habits, preferences, etc. Although this collection could be subject to approvals, requesting approval for every single detail and continuously would become over time cumbersome and inefficient. To address privacy concern in an IoT ecosystem, we examine in this paper how to make things “aware” of their rights of and duties towards the data they get, and the actions they perform over these data. Could we blend ODRL with any appropriate thing description-language to achieve this awareness and hence, have a new generation of things

¹www.vxchnge.com/blog/iot-statistics.

^a  <https://orcid.org/0000-0003-4462-8337>

^b  <https://orcid.org/0000-0002-9076-5001>

^c  <https://orcid.org/0000-0002-0958-8547>

^d  <https://orcid.org/0000-0001-7001-3710>

sensitive to privacy? Being W3C *de facto* standard to describe things, we adopt the World-of-Things Thing Description (WoT TD) to concretize this blend.

WoT TD abstracts a thing, whether physical or virtual, into an entity that would logically reside in an ecosystem of things and would engage in interactions with these things. To blend ODRL with WoT TD, we proceed with (i) examining separately the model of each specification, (ii) identifying potential overlaps/correspondences between the 2 models' constructs, and finally, (iii) using these constructs to make WoT TD-based things sensitive to ODRL-based rights and duties. Section 2 is an overview of IoT, ODRL, and WoT TD. Section 3 is the core of the paper by presenting a case study and conceptualizing and operationalizing the blend of ODRL with WoT TD for an IoT sensitive to privacy. Section 4 concludes the paper and discusses future work.

2 BACKGROUND

This section presents IoT, ODRL, and WoT TD.

2.1 Overview of IoT

A good number of works on IoT exist, which in fact does not help define IoT from a unique perspective (e.g., (Abdmeziem et al., 2016), (Barnaghi and Sheth, 2016), and (Taivalsaari and Mikkonen, 2017)). For illustration, Abdmeziem et al. present IoT characteristics and enabling technologies (Abdmeziem et al., 2016). The former include distribution, interoperability, scalability, resource scarcity, and security. And, the latter include sensing, communication, and actuating, and are mapped onto a 3-layer IoT architecture referred to as perception, network, and application. In (Barnaghi and Sheth, 2016), Barnaghi and Sheth discuss IoT's requirements and challenges. Requirements are related to quality, latency, trust, availability, reliability, and continuity that should impact efficient access and use of IoT data and services. And, challenges result from today's IoT ecosystems that feature billions of dynamic things that make existing search, discovery, and access techniques inappropriate for IoT data and services. Finally, Qin et al. define IoT from a data perspective as *"In the context of the Internet, addressable and interconnected things, instead of humans, act as the main data producers, as well as the main data consumers. Computers will be able to learn and gain information and knowledge to solve real world problems directly with the data fed from things. As an ultimate goal, computers enabled by the Internet of Things technologies will be able to*

sense and react to the real world for humans" (Qin et al., 2016).

2.2 Overview of ODRL

ODRL is an example of Rights Expression Languages (REL) that provides a flexible and interoperable information model, vocabulary, and encoding mechanisms to represent statements about the potential uses of assets. An asset is an identifiable resource (or collection of resources) such as data/information, content/media, applications, and services. The ODRL information model is available at www.w3.org/TR/odrl-model and whose main constructs are:

- Policy could include one to many permission, prohibition, or duty rules. First, Permission allows an action over an asset if all constraints are satisfied and all duties are fulfilled. Second, Prohibition disallows an action over an asset if all constraints are satisfied. And, Duty is the obligation to exercise an action that over an asset or not.
- Party is an entity or collection of entities that could correspond to a person, group of persons, organisation, or agent. A party can fulfill different roles including assigner (issuer of the rule), and assignee (recipient of the rule),
- Constraint is used to refine the specification of an action and a party/asset collection or to declare conditions applicable to a rule.

In JSON-LD Listing 1, the assigner (line 6) in charge of an asset, *movie1* (line 5), refers to a policy labelled as *agreement* (line 3) consisting of one permission rule and one prohibition rule. The permission rule (line 4) that is concerned with *display* action (line 8) that an assignee, *smart TV* (line 7), will execute subject to satisfying the constraint of not enabling the permission rule for more than 4 hours (lines 10-13). The prohibition rule (line 14) that is concerned with *digitize* action (line 18) that is associated with the same assignee (line 17). Should the assignee execute this action, then the remedy would be to apply the action *anonymize* (line 20) to the asset that is *movie1* (line 21).

Listing 1: Excerpt of movie's ODRL specification.

```

1 {"@context": "http://www.w3.org/ns/odrl.jsonld"
2 ,
3 "uid": "http://example.com/policy:01",
4 "@type": "Agreement",
5 "permission": [{
6   "target": "http://example.com/asset:movie1",
7   "assigner": "http://example.com/Movie1Party:
   org",
8   "assignee": "http://example.com/smart-TV",
9   "action": "display",
10  "constraint": {
11    "type": "Prohibition",
12    "action": "enable",
13    "target": "http://example.com/asset:movie1",
14    "action": "digitize",
15    "remedy": {
16      "action": "anonymize",
17      "target": "http://example.com/asset:movie1",
18      "action": "digitize",
19      "target": "http://example.com/asset:movie1",
20      "action": "anonymize",
21      "target": "http://example.com/asset:movie1"
22    }
23  }
24 }

```

```

8  "action": "display",
9    "constraint": [{
10     "leftOperand": "meteredTime",
11     "operator": "iteq",
12     "rightOperand": { "@value": "P4H",
13                       "@type": "xsd:duration" },
14     "unit": "https://www.wikidata.org/wiki/Q25235" }],
15 "prohibition": [{
16   "target": "http://example.com/asset:movie1",
17   "assigner": "http://example.com/Movie1Party:org",
18   "assignee": "http://example.com/smart-TV",
19   "action": "digitize",
20   "remedy": [{
21     "action": "anonymize",
22     "target": "http://example.com/asset:movie1"
23   }]
24 }]
```

2.3 Overview of WoT TD

WoT TD is a central building block in the W3C WoT and can be considered as a Thing's entry point (W3C, 2020). It describes the metadata and interfaces of things, where a thing is an abstraction of a physical or virtual entity that provides interactions to and participates in the WoT. Thing Descriptions provide a set of interactions based on a limited vocabulary that makes it possible both to integrate diverse devices and to allow diverse applications to interoperate. Fig. 1 is an excerpt of WoT TD model that is built upon 4 Vocabulary parts having each a namespace: (i) *core TD Vocabulary* defines the interaction model, (ii) *Data Schema Vocabulary* describes a common subset of terms defined in a JSON Schema, (iii) *WoT Security Vocabulary* identifies the configuration of the security mechanisms, and (iv) *Hypermedia Controls Vocabulary* provides a representation for Web links and Web forms that a Thing exposes to potential end-users.

In Fig. 1, InteractionAffordance is how end-users and/or peers could interact with a Thing: *Properties* of type PropertyAffordance allows to sense and control parameters, *Actions* of type ActionAffordance allows to invoke physical processes, and *Events* of type EventAffordance allows to asynchronously push communications such as notifications, discrete events, and streams of values to receivers.

Listing 2 shows a WoT TD instance of a lamp Thing referred to as MyLampThing. Thanks to this description, we know that there exists one Property affordance with the title *status* (line 8), an Action affordance is specified to *toggle* (lines 12-13) the switch status, and an Event affordance known as *overheating* (line 15) that enables a mechanism for asynchronous messages to be sent by a Thing. The list-

ing also specifies a basic security scheme requiring a username and password for access (line 5).

Listing 2: Excerpt of lamp's WoT TD specification.

```

1  {"@context": "https://www.w3.org/2019/wot/td/v1",
2   "id": "urn:dev:ops:32473-WoTLamp-1234",
3   "title": "MyLampThing",
4   "securityDefinitions": {
5     "basic_sc": {"scheme": "basic", "in": "header"}
6   },
7   "security": ["basic_sc"],
8   "properties": {
9     "status": {..."forms": [{"href": "..."}]}
10  },
11  "actions": {
12    "toggle": {"forms": [{"href": "..."}]},
13  "events": {
14    "overheating": {...}
15  }
16 }
```

To the best of our knowledge, there are no research that specifically examine the blend of ODRL with thing descriptions/WoT TD.

3 ODRL-WOT TD APPROACH

After presenting a case study that sheds light on privacy concerns, the rest of this section conceptualizes and operationalizes ODRL/WoT TD blend. Different WoT TD and WoT TD specifications and screenshots are used for illustration purposes.

3.1 Case Study

To illustrate ODRL/WoT TD blend for an IoT sensitive to privacy, our case study targets a long-term care center for elderly people. On different occasions, these people get together in the living room to watch movies among other activities. First, we treat movie as an asset whose use needs to be regulated. And, we treat smart TV as a thing that would act upon movies through specific actions. The smart TV is synchronized with the elderly people's smartphones making them a second thing in the case study that would act upon movies sometimes through the smart TV.

When playing movies, the smart TV would record details like titles, ratings, and years of release. While these details seem insignificant for elderly people, third parties like drug companies could use them to develop targeted awareness campaigns that could make these people subscribe to some healthcare programs. To prevent the smart TV from collecting details for unauthorized uses and/or sharing details with unapproved third-parties, we resort to ODRL policies. Our objective is to ensure that the smart TV

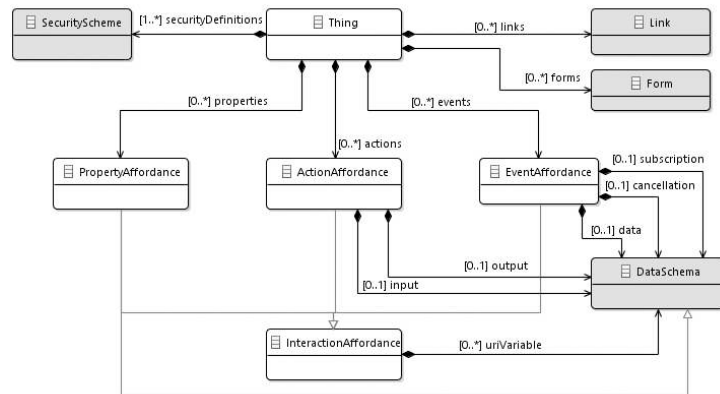


Figure 1: Excerpt of WoT TD information-model ((W3C, 2020)).

designated as assignee according to ODRL terminology, is aware of the permission, duty, and prohibition rules impacting movies and hence, sensitive to any privacy concerns. To this end, we specify the assignee/smart TV in WoT TD considering these permissions, duties, and prohibitions. This would make the smart TV’s actions compliant with ODRL policies by for instance, sharing a movie’s title because of a permission rule, enforcing a movie’s rating because of a duty rule, and scheduling a movie’s viewing hours because of a prohibition rule. The way we correspond asset-assignee:action to thing:action and identify and specify smart TV and smartphone is given next.

3.2 Approach Conceptualization

Fig. 2 conceptualizes the approach to blend ODRL with WoT TD by carrying out some operations at design-time (d) and the rest at run-time (r). During this blend, 2 software engineering best-practices are applied. First, ODRL specifications of assets and WoT TD specifications of things remain loosely coupled allowing their potential re-uses in other case studies without being restricted to specific ones. Second, this loosely coupling allows achieving the separation of concerns principle (Kambayashi and Ledgard, 2004).

Operations at Design Time. 2 stakeholders, asset owners and IoT engineers, are involved. The former use the *asset-definition* module to identify the necessary assets, e.g., movie, according to the case study’s needs and requirements and to specify these assets in ODRL. During the specification of assets (1^d), 2 ODRL constructs are deemed critical with respect to the policies linked to these assets: assignee and action. Upon receipt of the asset specifications from the *asset-definition* module, the *extractor* module (2^d) extracts the assignee and action constructs along with other associated details like policy and constraints and

makes them available to the IoT engineer. This one uses the *thing-definition* module to first, specify future things and their actions that would match assignees and actions (3^d), respectively, and second, identify additional things that would be deemed necessary with respect to the case study’s needs and requirements (4^d). In all cases, things will be specified in WoT TD. Since the additional things could act upon assets, the IoT engineer would inform the owners of these assets. The objective is to adjust these assets’ ODRL specifications, as they see fit (5^d). More details about developing thing specifications and adjusting asset specifications are given in Section 3.3. With respect to the elderly-center case study, smart TV would be associated with operation (3^d) and smartphone would be associated with operations (4^d) and (5^d).

Operations at Run Time. Upon submitting both asset specifications and thing specifications to the *deployer* module, this one makes things and assets reside in the cyber-physical ecosystem as either physical or digital entities so they act according to their respective specifications. In conjunction with this deployment, the *deployer* module informs the *monitoring* module about the things and assets so that it analyzes their behaviors and notifies the IoT engineers and asset owners, respectively, of any potential deviations from the prescribed thing and asset specifications. For instance, a thing acting as an assignee does not commit to a duty policy or violates a prohibition policy. More details about run-time operations are presented in Section 3.4.

3.3 Thing Identification and Specification

As stated earlier, there exist 2 ways to identify an IoT ecosystem’s future things, whether physical or digital, that could act upon assets. An ecosystem

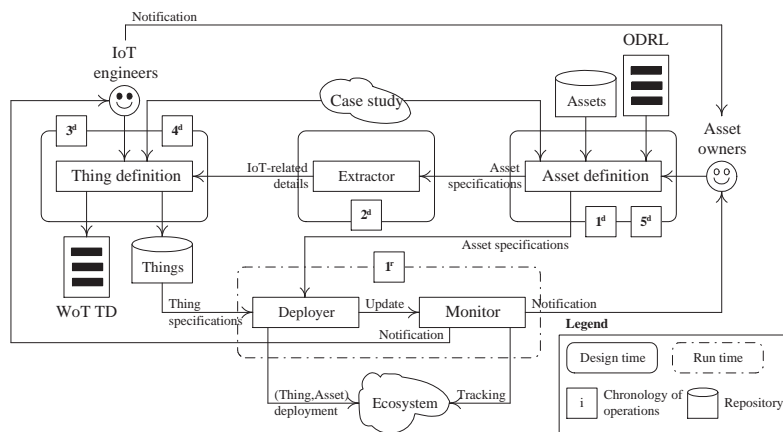


Figure 2: Representation of ODRL-WoT TD approach.

could be our elderly care-center. The first way taps into the assignee construct in all ODRL specifications where every assignee becomes an independent thing (Thing as per Fig. 1). The second way taps into the requirements of the under-development IoT applications of the case study like the elderly center. Indeed, the IoT engineer identifies new things not reported in the ODRL specifications, as he sees fit. Let us illustrate the first way of thing identification. Here the IoT engineer proceeds in 2 steps:

1. In the first step, the IoT engineer uses some details extracted from Listing 1 to convert the assignees, e.g., smart TV, into things and to put together a preliminary version of these things' WoT TD specifications like the one shown in Listing 3. The IoT engineer also refers to details from Listing 1 and how these things are used in daily life to insert some properties, e.g., *status* (line 7), and some actions, e.g., *switchON*, *switchOFF*, and *display*, (lines 23, 24, and 25-31, respectively), into their WoT TD specifications with focus on ActionAffordance as per Fig. 1. For those actions, e.g., *display*, already reported in Listing 1 and then, Listing 3, it is required that they are not part of any prohibition policies in the assets' ODRL specifications. Otherwise, the actions will be treated differently for instance, discarded at the IoT engineer's own discretion.
2. In the second step, the IoT engineer ensures that the actions in Listing 3 would comply with their existing constrained definitions in Listing 1, should these actions already exist in the ODRL specifications. For instance, the IoT engineer ensures the compliance of the *display* action with its associated constraints by including in Listing 3 first, an additional action, *suspend* (line 32), along with its necessary parameters, *displayValidityPeriod* and *movie-id* (lines 34-

35) to put displaying movies on hold, and second, additional properties, *possibleMovies* (line 21) for the list of possible movies that this TV can display and *dispTime* (lines 8-12) for the time elapsed since the permission rule to display a movie is activated.

Still in the first way of thing identification with focus here on the prohibited actions with remedy, e.g., *digitize* in Listing 1. Should the IoT engineer decide on retaining these actions in things' WoT TD specifications, e.g., *digitize* in Listing 3 (line 40), then the remedy actions must be included in these specifications, e.g., *anonymize* in Listing 3 (line 41). Finally, for those prohibited actions without remedy, they are automatically discarded from the analysis of assets' ODRL specifications. These actions cannot be executed.

Listing 3: Excerpt of smart TV's WoT TD specification.

```

1 {"@context": "https://www.w3.org/2019/wot/td/v1",
2  "id": "http://example.com/smart-tv",
3  "name": "SmartTV",
4  "securityDefinitions": {...},
5  "security": [...],
6  "properties": {
7    "status": {...}
8    "dispTime": {
9      "observable": true,
10     "description": "display time of movie by smart TV",
11     "type": "integer",
12     "minimum": 0...},
13   "movie_details": {
14     "title": "details",
15     "observable": true,
16     "type": "object",
17     "description": "details of the movie displayed",
18     "properties": {
19       "Date_of_release": {"type": "Date",

```

```

20     ...},
21     "Duration":{"type": "integer",...}...
22   },
23   "possibleMovies":{"..."},...},
24 "actions":{
25   "switchON":{"..."},
26   "switchOFF":{"..."},
27   "display":{
28     "description": "Display movie",
29     "input": {"displayTime": "integer", "
30       movie-id":"anyURI"},
31     "forms": [{
32       "href":"http://example.com/smart-tv/
33         dsp{?displayTime,movie-id}",
34       "http:methodName": "POST",
35       "contentType": "application/json"}]}
36   },
37   "suspend":{
38     "description": "Suspend movie",
39     "input":{"displayValidityPeriod":"240",
40       "movie-id":"http://example.com/asset:
41         movie1"},
42     "forms": [{
43       "href":"https://SMARTTV.example.com/
44         suspend{?displayValidityPeriod,
45         movie-id}",
46       "http:methodName": "POST",
47       "contentType":"application/json"},
48     ...},
49     "digitize":{"..."},
50     "anonymize":{"..."}...
51   }
52 }

```

We now discuss the second way of thing identification. Contrarily to smart TV that is specifically reported in movie’s ODRL specification, the IoT engineer decides on having extra things, e.g., remote control and smartphone, that would allow to satisfy the needs and requirements of the case study’s under-development IoT applications. To this end, the IoT engineer develops several WoT TD specifications from scratch like Listing 4 for smartphone. In this listing, many actions exist such as *controlTV* (line 21) that configures a smart TV’s main functionalities from the smartphone, *displayOnTV* (line 22) that switches displaying movies from the smartphone to a smart TV, and *digitize* (line 23) that produces a digital copy of the displayed movies on the smartphone.

Listing 4: Excerpt of smartphone’s WoT TD specification.

```

1 {"@context": "https://www.w3.org/2019/wot/td/v1",
2 "id": "http://example.com/Smartphone/001",
3 "name": "RemoteControl",
4 "securityDefinitions": {...},
5 "security": [...],
6 "properties":{
7   "status":{"..."},
8   "contactsList":{
9     "writable": true,

```

```

10     "observable": true,
11     "forms": [{
12       "href": "https://SmartPhone.example
13         .com/contactsList"}]},...},
14 "actions":{
15   "switchON":{"..."},
16   "manageContact":{
17     "forms": [{
18       "href": "https://SmartPhone.example
19         .com/contactList",
20       "http:methodName": "POST",
21       "contentType": "application/json"}]}
22   },
23   "downloadApp":{"..."},
24   "controlTV":{"..."},
25   "displayOnTV":{"..."},
26   "digitize":{"..."}...
27 }

```

After specifying the extra things, i.e., remote control and smartphone, the IoT engineer decides on allowing smartphones to act on movies by displaying their trailers, for example. To accommodate this display in compliance with the existing ODRL specification of movie (Listing 1), the owner adjusts this specification as per Listing 5 where the type is no longer agreement but policy² (line 3), smartphone is added as an assignee (line 7), and some properties are compacted such as target (line 4).

Listing 5: Excerpt of movie’s adjusted ODRL specification.

```

1 {"@context": "http://www.w3.org/ns/odrl.jsonld",
2 "uid": "http://example.com/policy:01",
3 "@type": "Policy",
4 "target": "http://example.com/asset:movie1",
5 "assigner": "http://example.com/Movie1Party:org",
6 "permission": [{
7   "assignee": "http://example.com/smart-TV",
8   "assignee": "http://example.com/smartphone/001",
9   "action": "display",
10  "constraint": [{
11    "leftOperand": "meteredTime",
12    "operator": "iteq",
13    "rightOperand": {"@value": "P4H", "@type": "xsd:duration"},
14    "unit": "https://www.wikidata.org/wiki/Q25235"}]}],
15 "prohibition": [{
16   "assignee": "http://example.com/smart-TV",
17   "action": "digitize",
18   "remedy": [{
19     "action": "anonymize"}]}]
20 }

```

²It permits to support a rule being related to multiple Assets, Parties, and Actions.

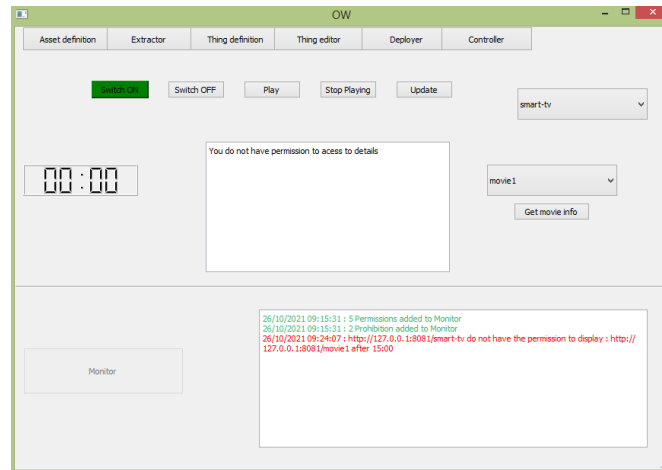


Figure 3: *OW*'s main interface along with some functionalities' outcomes like monitoring.

3.4 Approach Implementation

To put *ODRL-WoT* TD blend into action, we implemented a system, *OW*, in Python3 enhanced with sip, PyQt5, and PyQtWebEngine. Additional development tools and libraries also include Node.js, ThingWeb node WoT, Postman, json, os, Naked, and pathos. Fig. 3 shows *OW*'s main interface integrating 6 modules: *asset definition* that loads any ODRL editor³, *extractor*, *thing definition*, *thing editor* that loads Eclipse Edi{TD} or available at eclipse.github.io/edittdor, *deployer*, and *controller*.

We asked asset owners to define their assets by enacting the *asset-definition* module and then, filling out the necessary fields as they see fit. Next and in compliance with the first option of thing identification (Section 3.3), we invited an IoT engineer to enact first, the *extractor* module allowing to generate rules associated with the already-specified assets and second, the *thing-definition* module to generate the initial WoT TD specifications of future things like smart TV (Fig. 4-before). Should the IoT engineer wish to revise these specifications, he could do so by enacting the *thing-editor* module. A video summarizing the use of *OW* is available at youtu.be/SCN0EpgWAdE.

To deploy things, the *deployer* uses a functional virtual device known as *Servient*. It provides standardized access, communication, and control capabilities to physical devices. Fig. 4-after corresponds to a deployed smart TV acting both as a server offering services such as switch on and display to users, and as a client consuming these services. To deploy assets like movies, we also associated them with a *Servient* allowing things like smart TV to use these assets. To ensure thing compliance with ODRL poli-

cies, the *monitor* module is enacted as per Fig. 3. In this figure, the *monitor* reports the actions that a user, through the smart TV, attempted to execute over movies. Warnings due to rule violations are issued, if deemed necessary. Finally, the *monitor* also reports any updates of ODRL policies like added permissions or dropped prohibitions.

During the implementation, we had to tackle many challenges of different complexity levels. For instance, we had to consider the variety of tools adopted, the on-the-fly enrichment of assets, the simulation of real IoT devices and end-users during communication, the alteration of things' descriptions according to ODRL, and, finally, the real-time monitoring of things' operations according to ODRL again. In conjunction with these challenges, lessons learned include defining assets as things in order to ensure the privacy of their data and adding on-the-fly new properties to things like smart TV in order to check movies' lists before executing any action. These properties permitted to communicate to the smart TV what is permitted *versus* what is prohibited for each movie. Although ThingWeb node WoT required accesses to things' properties through JavaScript promises, coding with JavaScript promises was tricky for real-time applications. JavaScript code-execution is by default asynchronous, which is not in line with how we envisioned safeguarding the privacy of our things. These properties played a major role in making things aware of their current status. For instance, if a request is sent to smart TV to play *movie1*, then the *play* action will not start until the JavaScript promise containing *movie1*'s details is fulfilled and the *play_status* property contains the value "Playing". Consulting properties' values happened periodically throughout the smart TV's lifetime.

³E.g., www.w3.org/community/odrl/2019/10/29/odrl-editors-and-licenses.

```

1  WoT.produce({
2    "@context": "https://www.w3.org/2019/wot/td/v1",
3    ... 'id', 'title' ...
4    "properties": { /* Properties definition */ },
5    "actions": { /* Actions definition */ }
6  }).then(function (TVthing) {
7    --> code here : Initializing the properties values.
8    --> code here : Updating the different properties values.
9    TVthing.setActionHandler("switchON", (params, options)=>{ });
10   TVthing.setActionHandler("switchOFF", (params, options)=>{ });
11   TVthing.setPropertyWriteHandler("Movie_to_Display", (val) => { /* Selecting a movie to display */ });
12   TVthing.setActionHandler("display", (params, options)=> { /* Playing the movie selected */ });
13   TVthing.expose().then(function () { console.info(TVthing.getThingDescription().title + " ready"); });
14   }).catch(function (e) { console.log(e); });
15
16  WoT.produce({
17    "@context": "https://www.w3.org/2019/wot/td/v1",
18    ... 'id', 'title' ...
19    "properties": { /* Properties definition */
20      "possibleMovies": { },
21      "possibleMovies1": { },
22    },
23    "actions": { /* Actions definition */ }
24  }).then(function (TVthing) {
25    --> code here : Initializing the properties values.
26    TVthing.writeProperty("possibleMovies", [ "http://127.0.0.1:8081/movie1", "http://127.0.0.1:8081/movie2" ]) //Display
27    TVthing.writeProperty("possibleMovies1", [ "http://127.0.0.1:8081/movie1" ]) //Play ;
28    --> code here : Updating the different properti
29    Promise.all([possibleMovies]).then((values) => { });
30    Promise.all([possibleMovies1]).then((values) => { });
31    TVthing.setActionHandler("switchON", (params, options)=>{ });
32    TVthing.setActionHandler("switchOFF", (params, options)=>{ });
33    TVthing.setPropertyWriteHandler("Movie_to_Display", (val) => {
34      if (Object.values(test).includes(val)){ /* Selecting a movie to display */}else { /* Display not allowed handler */ } });
35    TVthing.setActionHandler("display", (params, options)=>{
36      if(movie!="" && Object.values(test1).includes(movie)){ /* Playing the movie selected */ } else {
37        TVthing.writeProperty("status","Oops ! you do not have the Permission to display this movie, Please contact your Administrator ... "); } });
38    TVthing.expose().then(function () { console.info(TVthing.getThingDescription().title + " ready"); });
39    }).catch(function (e) { console.log(e); });

```

Figure 4: Excerpt of smart-TV definition before/after adjustment.

4 CONCLUSION

This paper presented what we would refer to as a win-win partnership between ODRL and WoT TD. The aim is to address privacy concerns in the particular context of IoT. Users are more and more reluctant to embracing IoT due to a variety of concerns with focus on privacy in this paper. To ensure that future things “know” what they can and cannot do according to users’ preferences, captured in ODRL, we adjusted these things’ specifications, exemplified with WoT TD, according to these preferences that we define using ODRL-based permission, prohibition, and obligation rules. A system demonstrating the blend of ODRL with WoT TD has been developed using different tools and technologies, and tested in the context of a center for elderly people. Real things like smart TV and smartphone have been specified in a way they have become sensitive to privacy thanks to rules integrated into ODRL policies. In term of future work, we would to examine first, the appropriateness of exposing things as assets (a cord construct in ODRL) and hence, could be specified in ODRL as well, and second, the adoption of aspect-oriented programming to address cross-cutting concerns over things’ specifications.

REFERENCES

Abdmeziem, M., Tandjaoui, D., and Romdhani, I. (2016). Architecting the Internet of Things: State of the Art. In *Robots and Sensor Clouds*, chapter 3.

Barnaghi, P. and Sheth, A. (2016). On Searching the Internet of Things: Requirements and Challenges. *IEEE Intelligent Systems*, 31(6).

Gupta, B. and Quamara, M. (2020). An Overview of Internet of Things (IoT): Architectural Aspects, Challenges, and Protocols. *Concurrency and Computation: Practice and Experience*, 32(21).

Kambayashi, Y. and Ledgard, H. (2004). The Separation Principle: A Programming Paradigm. *IEEE Software*, 21(2).

Mohanta, B., Jena, D., Ramasubbareddy, S., Daneshmand, M., and Gandomi, A. (2021). Addressing Security and Privacy Issues of IoT Using Blockchain Technology. *IEEE Internet of Things Journal*, 8(2).

Qin, Y., Sheng, Q., Falkner, N., Dustdar, S., Wang, H., and Vasilakos, A. (2016). When Things Matter: A Data-Centric View of the Internet of Things. *Journal of Network and Computer Applications*, 64.

Silva, P., Monteiro, E., and Simões, P. (2021). Privacy in the Cloud: A Survey of Existing Solutions and Research Challenges. *IEEE Access*, 9.

Taivalasaari, A. and Mikkonen, T. (2017). A Roadmap to the Programmable World: Software Challenges in the IoT Era. *IEEE Software*, 34(1).

W3C (2018). ODRL Information Model 2.2. <https://www.w3.org/TR/2018/REC-odrl-model-20180215/>.

W3C (2020). Web of Things (WoT) Thing Description. www.w3.org/TR/wot-thing-description.