



Binding the Web of Things with LwM2M for a vehicular use case

Benjamin KLOTZ

klotz@eurecom.fr

Goals

- **Demonstration of vehicle sensors as Web Things**
 - Properties/actions/events from the car's sensor/actuator network
 - Potentially possible with every sensors/actuators
- **Demonstration of a protocol binding between WoT and**



Goals

■ Interest of WoT



- Integrate data from the vehicle with external knowledge bases
- Adds a semantic enrichment layer
- Can be linked with the work of the W3C Automotive WG^[1]

■ Interest of Leshan for LwM2M communication

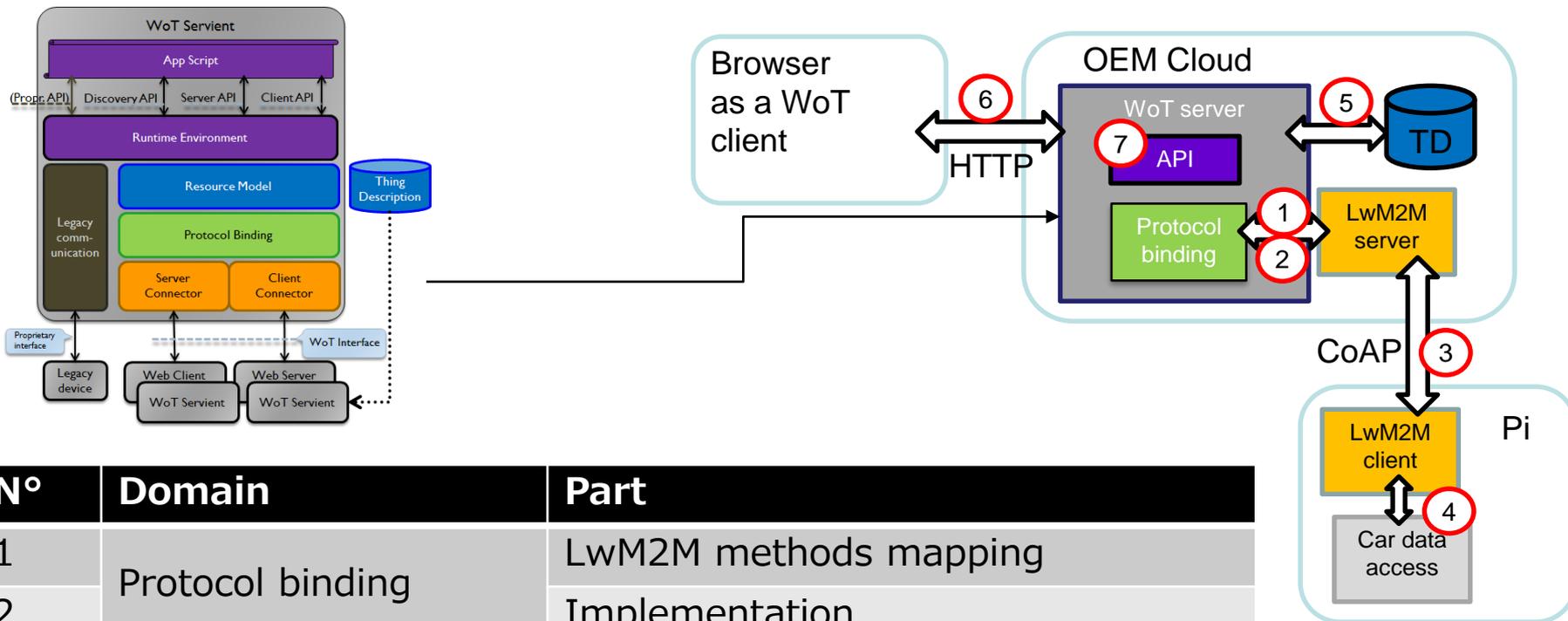


- LwM2M is fast deployable, with efficient device management and security workflow
- Leshan^[2] is open-source, java-based and adapted for the creation of lwM2M objects and instances on the client side.
- Leshan provides a user interface and an API

[1] <https://www.w3.org/auto/wg/>

[2] <http://www.eclipse.org/leshan/>

General architecture and parts



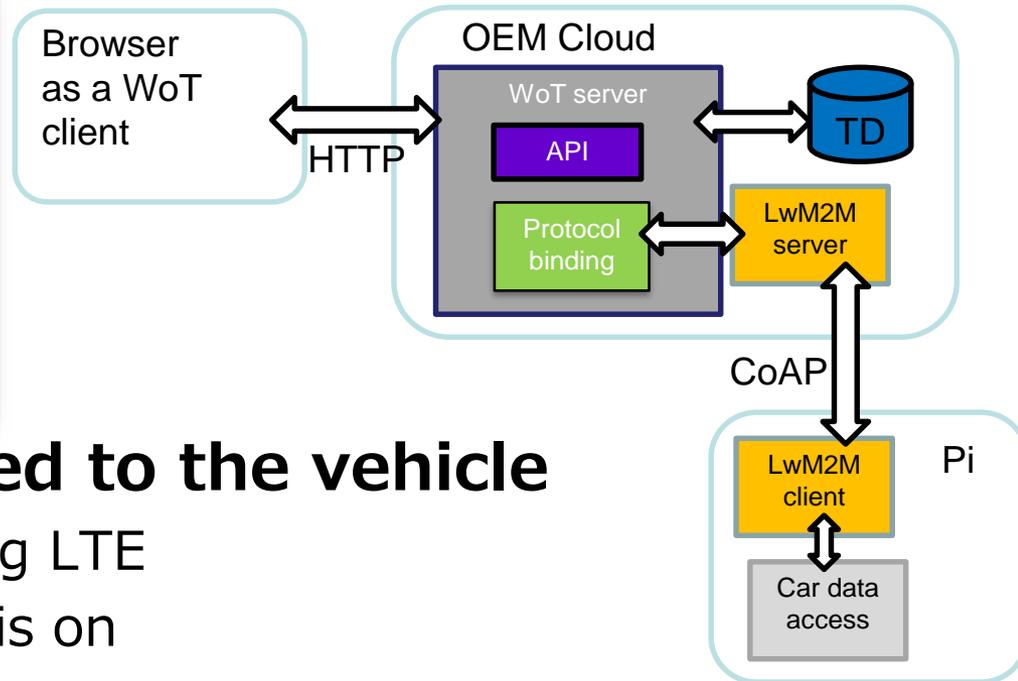
N°	Domain	Part
1	Protocol binding	LwM2M methods mapping
2		Implementation
3	Demo setup	Hardware and communication
4		Car data access
5	Thing description	Retrieval and parsing
6	WoT communication	data communication
7	API	WoT endpoint

Protocol binding: LwM2M

WoT		LwM2M		Input	Output
Interactions	Method	Method	if specific target		
Property	Read	Read		property	Value
	Write	Write		property, parameter	
Action	Invoke	Execute		action object	
	Update task	Write on instance	Action parameter	action instance, parameters	
	Cancel task	Write on instance	Action parameter	action instance, parameters	
Event	Subscribe	Observe		signal	Regular values
	Update subscription	Write on instance	Subscription parameter	subscription, parameter	
	Cancel subscription	Write on instance	Observe	subscription, parameter	



Demo setup



- **A Raspberry pi connected to the vehicle**
 - connected to the cloud using LTE
 - Accessible when the vehicle is on
- **Vehicle data is mapped with LwM2M objects in the Pi**
 - Private IDs remain hidden, but are accessible
 - The LwM2M server retrieves the result of the job without modification.

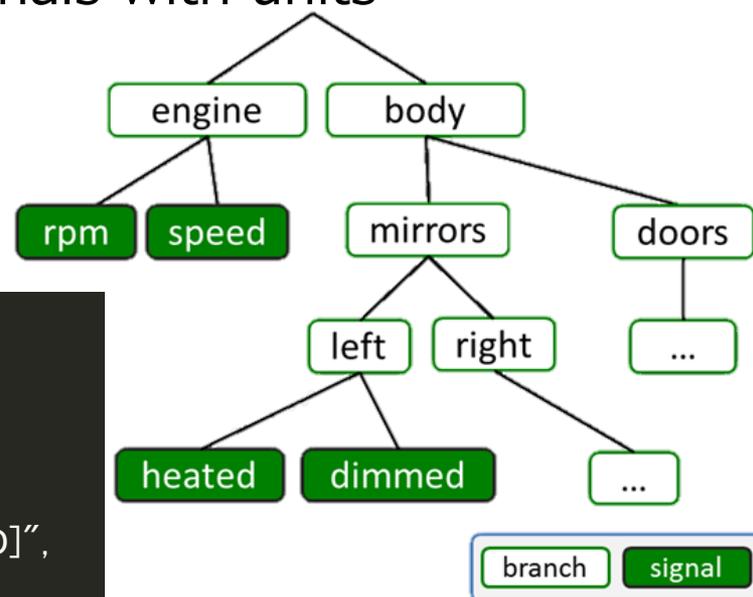
Thing description: context

Need: sensor/actuator ontology, car ontology

- <https://w3c.github.io/wot/w3c-wot-td-context.jsonld/>,
<https://w3c.github.io/wot/w3c-wot-common-context.jsonld/>
- <https://auto.schema.org/> for static information (name, brand...)
- Linked version of GENIVI's Vehicle Signal Specification (V. Charpenay)
 - Contains descriptions of car's signals with units

```
"@context": [ "https://w3c.github.io/wot/w3c-wot-td-context.jsonld/", {"wot": "https://w3c.github.io/wot/w3c-wot-common-context.jsonld/"}, {"auto": "https://schema.org/" }, {"vss": "vss2ttl/vss.ttl#" } ]
```

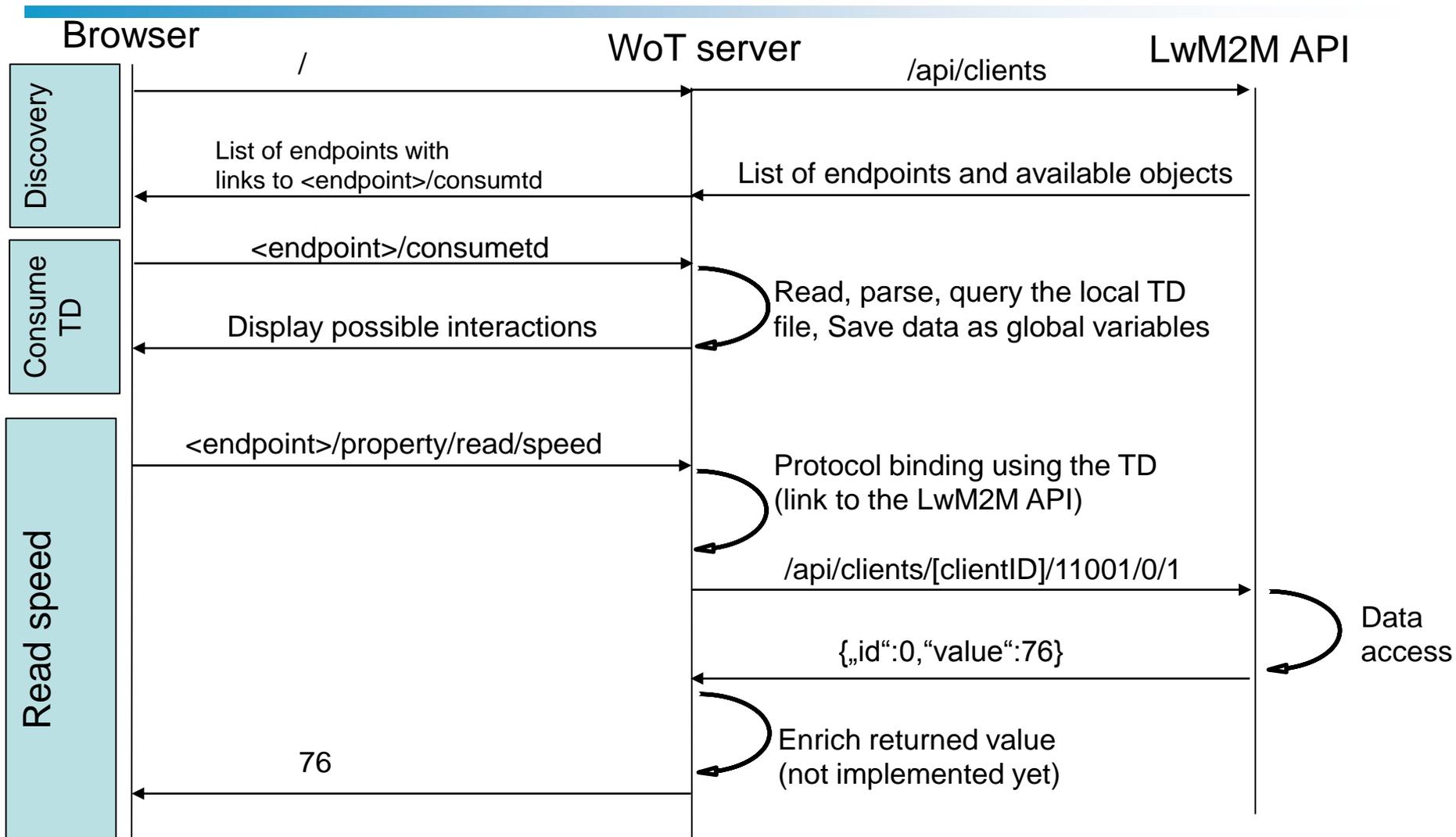
```
{ "@type": [ "Action", "vss:RearShadeSwitch" ],  
  "name": "openWindows",  
  "inputData": {  
    "valueType": { "type": "integer" },  
    "wot:unit": "wot:ms" },  
  "links": { "href": "http://localhost:8080/api/clients/[clientID]/[signalID]",  
    "mediaType": "application/json" } }
```



TD retrieval and parsing

- **The TD file is stored in the same repository as the WoT server**
 - File read and parsed
 - SPARQL query: find interactions and their parameters
 - Generate link to the interaction page
- **Endpoint/Read/<MethodId>:**
 - Start the function associated with the Read method

Thing description: sequence diagram



WoT communication

- **WoT server: python script with Flask**
 - A page for Thing discovery
 - A page to consume the TD
 - One web page per interaction
- **WoT client: browser**
- **Every exchange is based on the HTTP protocol**

Final interface

■ API

- Endpoint/consumetd consumes the TD
- Endpoint/read, Endpoint/write, Endpoint/subscribe, Endpoint/delete, Endpoint/invoke, Endpoint/cancel, Endpoint/update pages for one task
- Inputs in the URIs: Endpoint/write/methodID/Value

Conclusion and future work

■ Challenges faced:

- Provide a remote access to vehicle data adapted for web use cases
 - Is still limited by what the car data access
- Develop a protocol binding between WoT and LwM2M
 - Easy basic mapping of methods
 - Was made possible using the Leshan API
- Link the work of the W3C WoT IG and the W3C Automotive WG
 - Using the W3C Automotive WG/GENIVI's Vehicle Signal Spec

■ Future work

- Develop use cases with other sensors/actuators
- Do data integration from the vehicle and external knowledge bases
- Present the link between W3C WoT and auto to the latter

Thank you for your attention

Do you have questions ?

Contact:

Benjamin Klotz

klotz@eurecom.fr