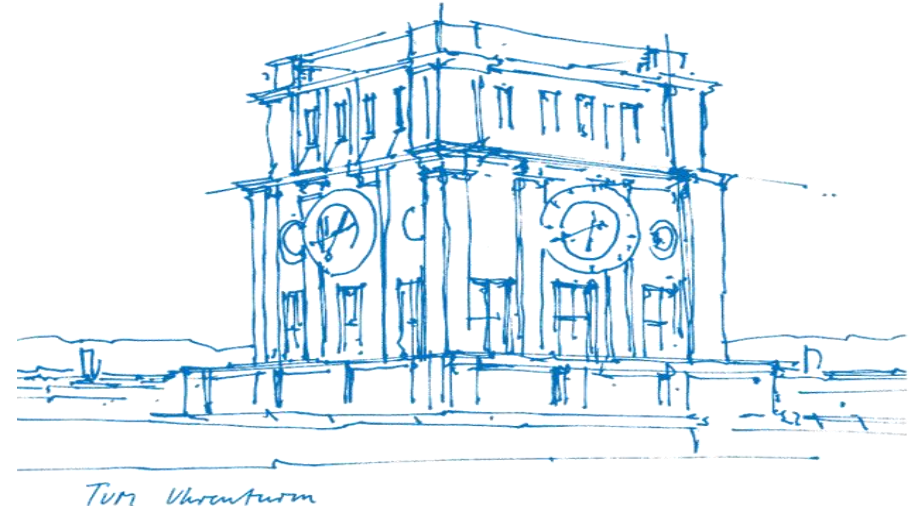


# Web of Things System Description for Representation of Mashups

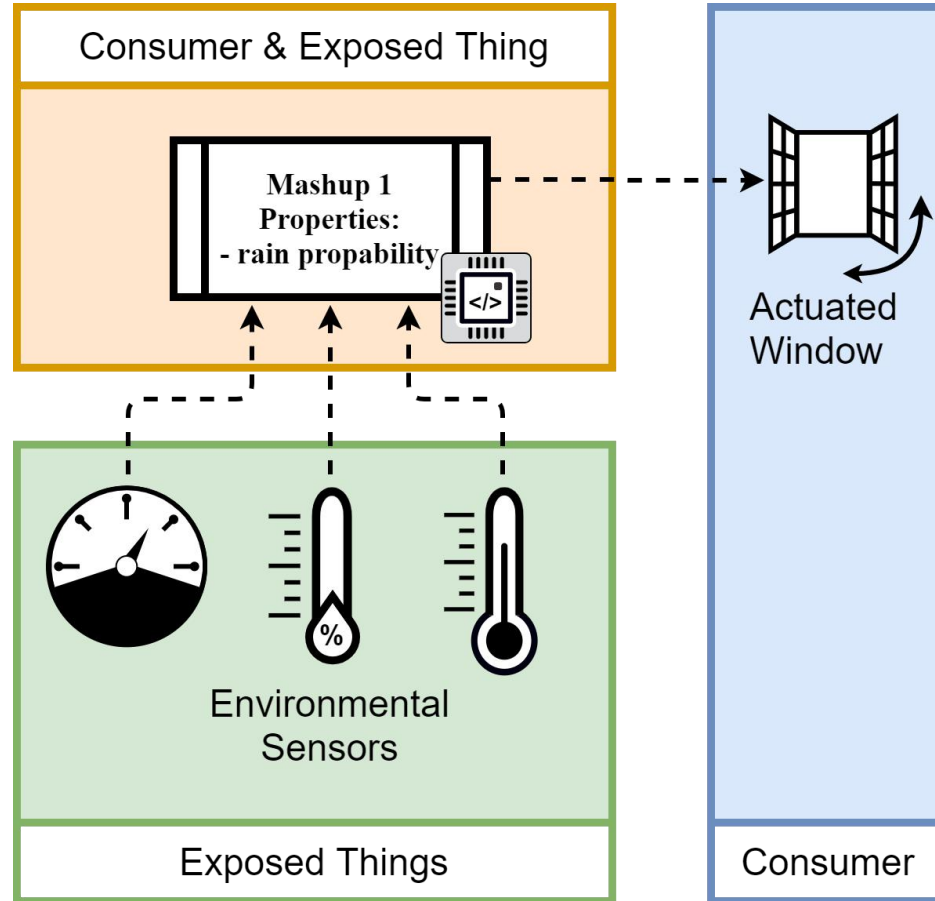
W3C TPAC 2020

Ege Korkan

October 21st, 2020



# Mashups



# How are Mashups done today?

Code with an arbitrary language, library/framework

```
var myTemperature;  
var myHumidity;  
request('http://myThing.com/properties/temperature', function (body1) {  
  myTemperature = body1;  
  request('http://myOtherThing.com/properties/humidity', function (body2) {  
    myHumidity = body2;  
    console.log( calculateRainProbability (myTemperature, myHumidity) );  
  });  
});
```

```
var myTemperature;  
var myHumidity;  
request('http://myThing.com/properties/temperature', function (body1) {  
    myTemperature = body1;  
    request('http://myOtherThing.com/properties/humidity', function (body2) {  
        myHumidity = body2;  
        console.log( calculateRainProbability (myTemperature, myHumidity) );  
    });  
});
```

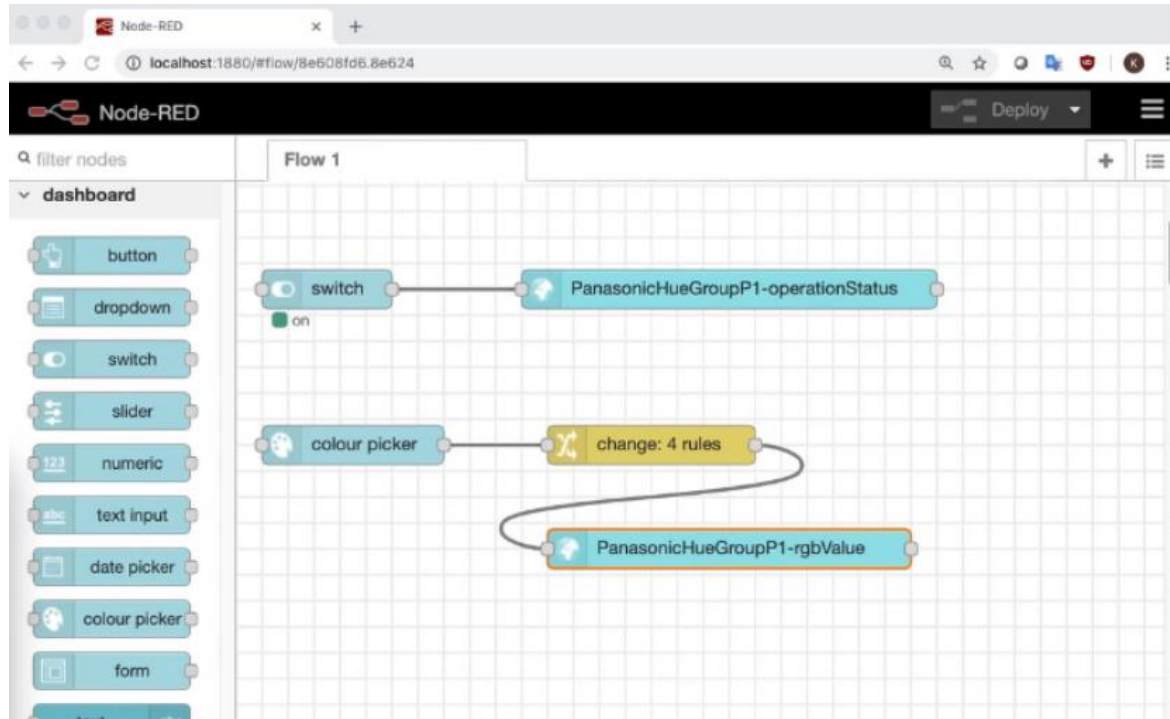
# How are Mashups done today?

Scripting API!

```
WoTHelpers.fetch("http://myThing.com").then(async (myTd) => {  
  WoTHelpers.fetch("http://myOtherThing.com").then(async (otherTd) => {  
    try {  
      let myThing = await WoT.consume(myTd);  
      let otherThing = await WoT.consume(otherTd);  
      var temperature = await myThing.readProperty("temperature");  
      var humidity = await otherThing.readProperty("humidity");  
      console.log( calculateRainProbability (myTemperature, myHumidity) );  
    } catch (err) { console.error("Script error:", err)}  
  })  
})
```

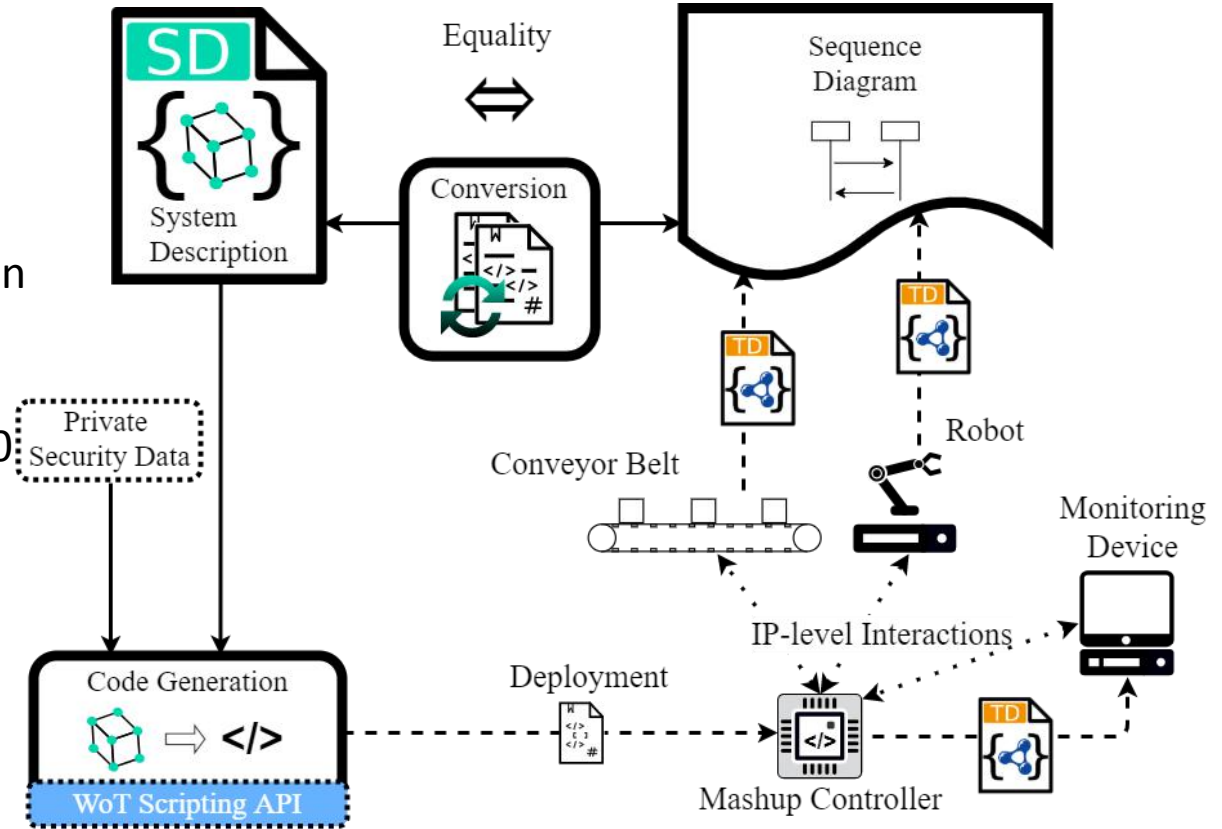
# How are Mashups done today?

Node-RED with node-gen

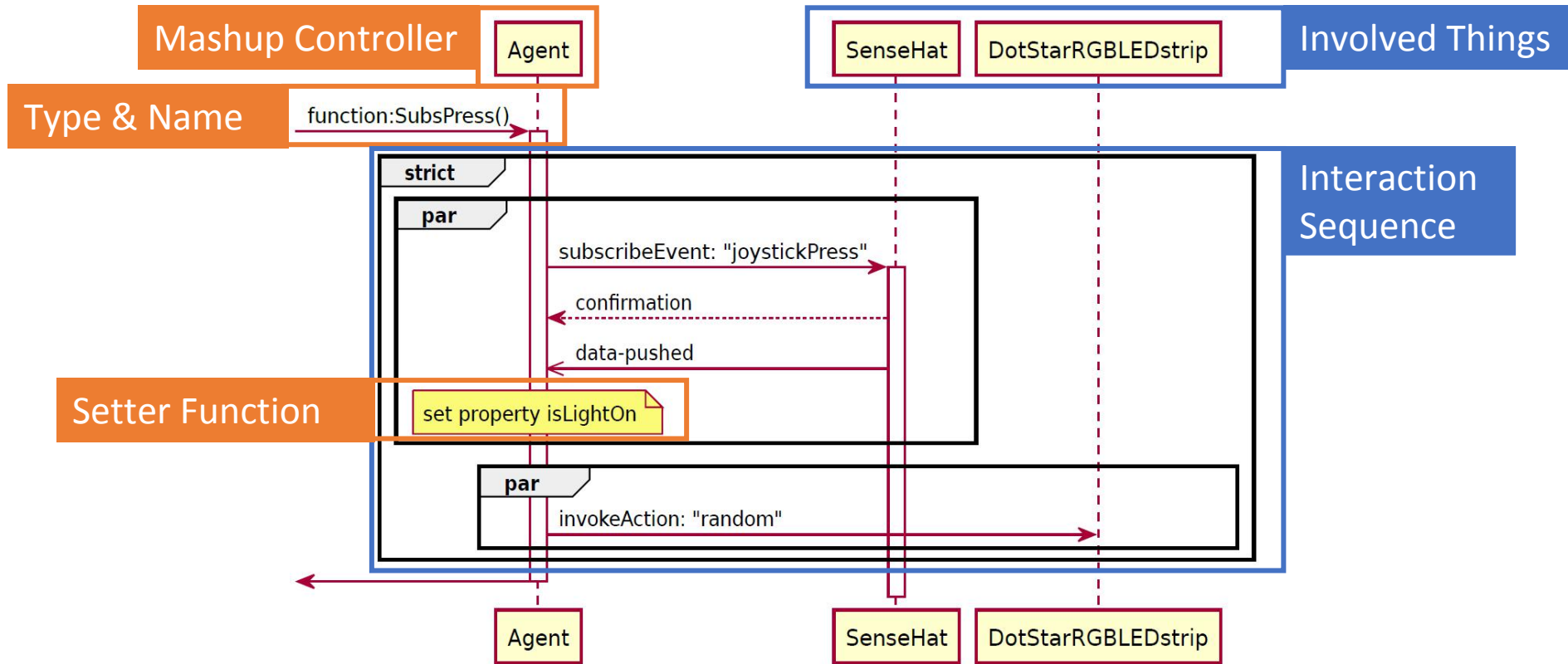


# Mashup Management Approach

- WoT Sequence Diagram
  - WoT System Description
  - Conversion
  - Automatic code generation
  - Interaction Affordances creation
- Published on COINS2020 Conference on September 2020



# WoT Sequence Diagram – Example





# WoT System Description – Structure




## Keywords

- *Things:*  
Things involved in the Mashup
- *Path:*  
Application logic
- *Functions & Variables:*  
Internal logic and data
- Keywords for their child elements



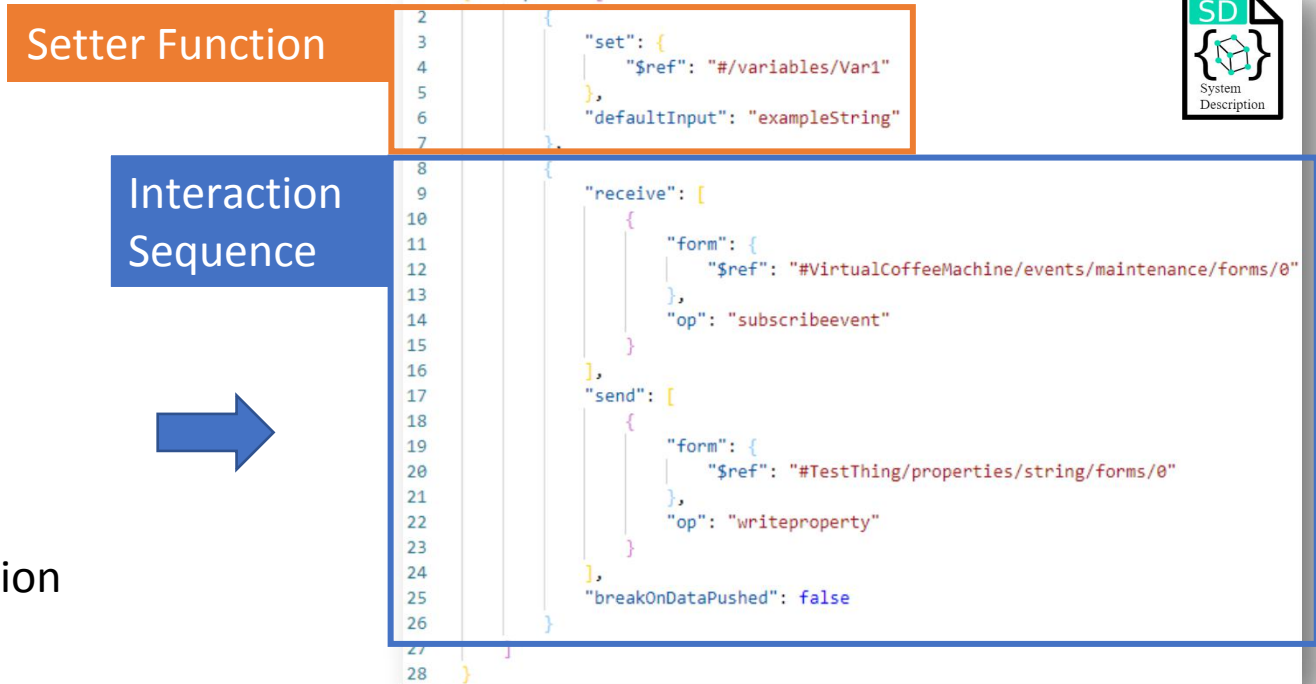
## Semantics

- JSON-LD context {  }
- SD ontology

# WoT System Description – Example



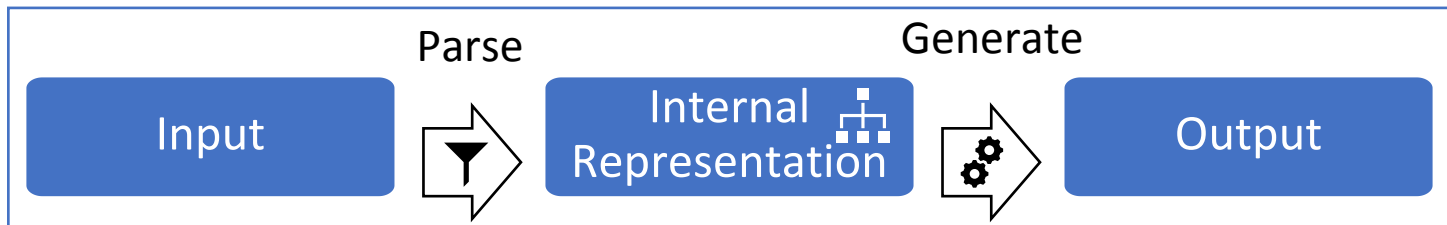
JSON Schema Definition



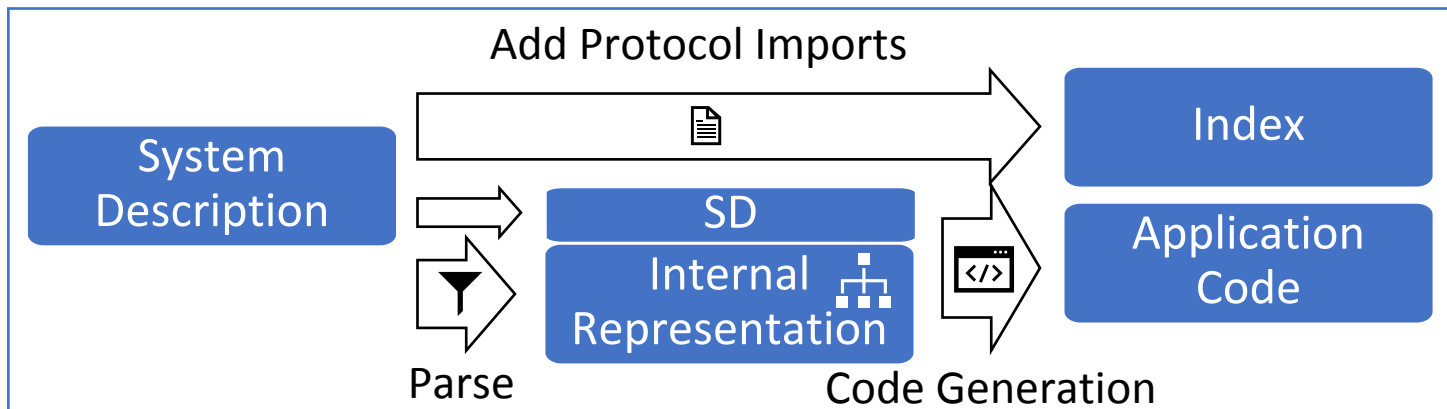
System Description

# Algorithms

## Conversion algorithms (System Description <-> Sequence Diagram)



## Code generation

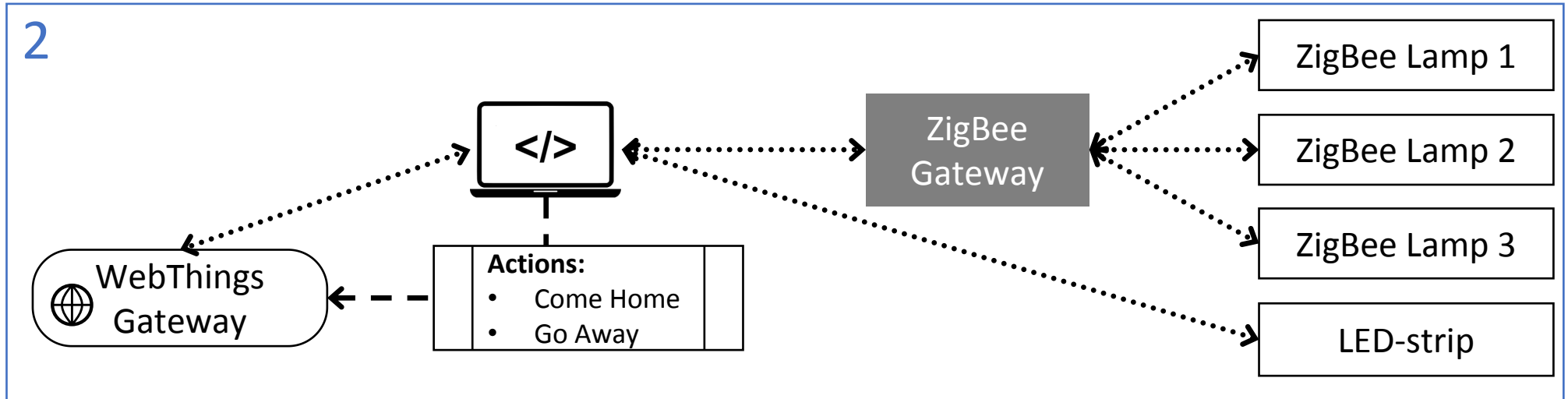
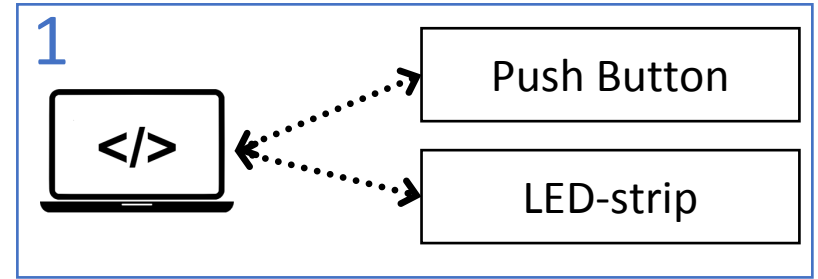


## Implementation



# Evaluation – Case Studies 1 & 2

- **Scenario 1**  
Simple, Turn LEDs for 10s on, if button is pushed.
- **Scenario 2**  
W3C 2<sup>nd</sup> WoT Workshop, Smart home scenario.  
Provide come home/go away (lights on/off).  
-> Provide, for example Smart Assistant Integration

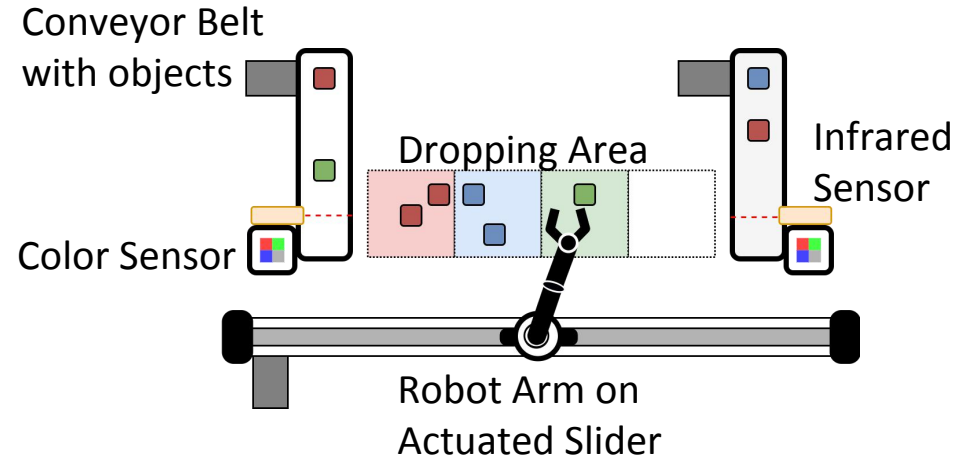


# Evaluation – Case Study 3

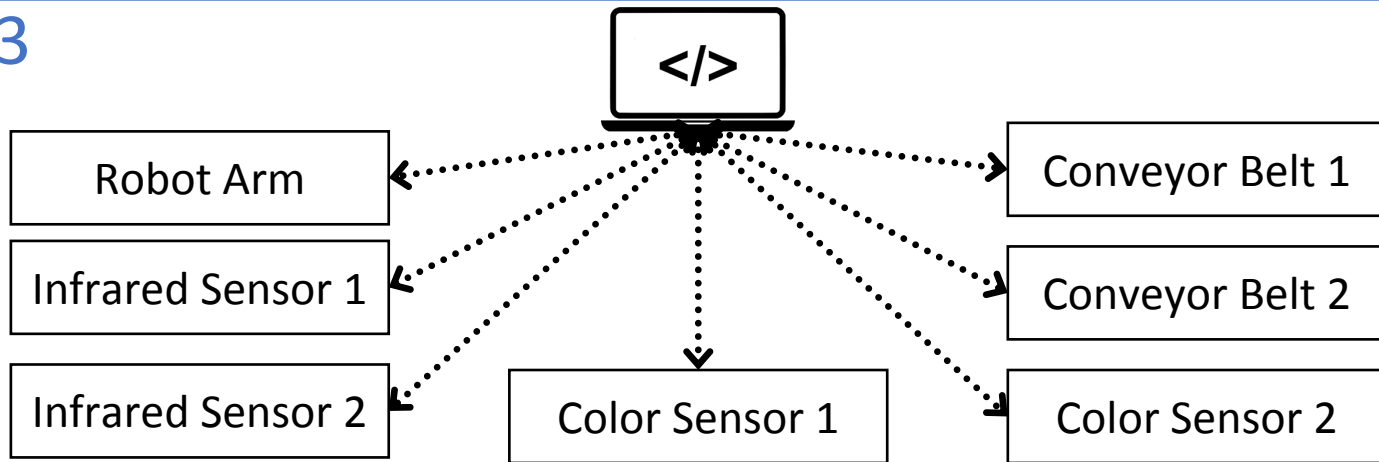
- **Scenario 3**

ESI industrial scenario testbench.

**Task:** Sort objects from conveyor belts.



3

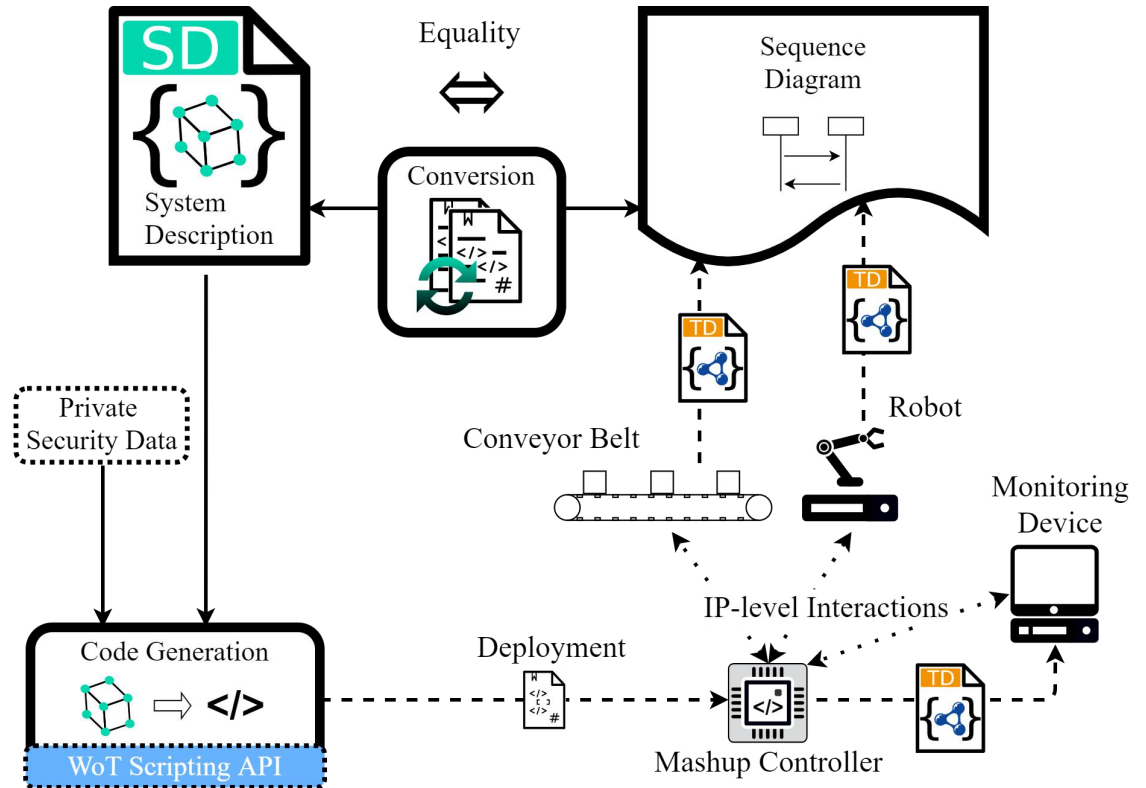


# Further Information

- Code generation implementation, examples, JSON-LD context file, grammar and more at <https://github.com/tum-esi/wot-system-description>
- Paper available at [https://www.ei.tum.de/fileadmin/tueifei/esi/2020-05-26\\_CRV\\_WoT-System-Description-COINS-compressed.pdf](https://www.ei.tum.de/fileadmin/tueifei/esi/2020-05-26_CRV_WoT-System-Description-COINS-compressed.pdf)
- Evaluation data (SDs, Sequence Diagrams, generated code) available as zip on request
- Video from the conference: <https://www.youtube.com/watch?v=yOfA24IbGj0>

# Conclusion

- WoT Sequence Diagram  
-> gives an insight
- WoT System Description  
-> open, machine-understandable
- Conversions combine advantages
- Code generation reduces manual effort
- Simple Interaction Affordance creation
- Evaluation with three case studies  
Improved WoT Mashup Management



# Contact

Ege Korkan  
Research Associate

Associate Professorship of Embedded Systems and Internet of Things  
Department of Electrical and Computer Engineering  
Technical University of Munich

E-Mail: [ege.korkan@tum.de](mailto:ege.korkan@tum.de)  
Phone: +49.89.289.23573

Website: <https://www.ei.tum.de/en/esi/staff/korkan/>  
Personal Website: <http://egekorkan.com/>  
Twitter: [@egekorkan](https://twitter.com/egekorkan)

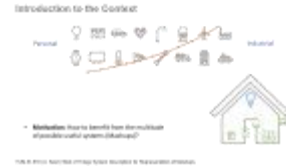
THANK YOU



# Questions?



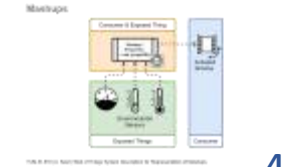
1



2



3



4



5



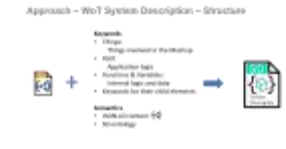
6



7



8



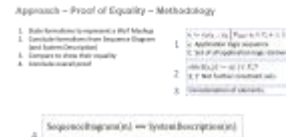
9



10



11



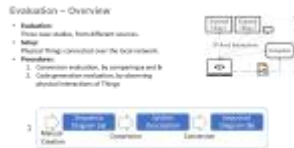
12



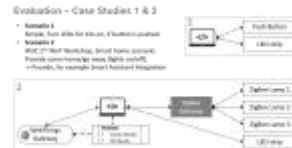
13



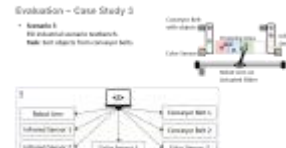
14



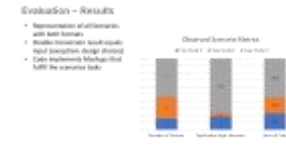
15



16



17



18



19

# References & Download

- 1)<https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/>
- 2)<https://www.w3.org/WoT/IG/wiki/images/4/40/Wot-large.png>
- 3)<https://plantuml.com>
- 4)<https://json-ld.org/images/json-ld-data.png>
- 5)<https://json-schema.org/assets/logo.svg>
- 6)H. Störrle, “Semantics of Interactions in UML 2.0,” in Proc. HCC., IEEE, 2003, pp. 129–136.
- 7)<https://github.com/npm/logos/blob/master/npm%20logo/npm-logo-black.png>
- 8)<https://www.typescriptlang.org/favicon.ico>
- 9)<https://nodejs.org/static/images/logos/nodejs-new-pantone-black.svg>
- 10)K.-H. Le, S. K. Datta, C. Bonnet, and F. Hamon, “WoT-AD: A Descriptive Language for Group of Things in Massive IoT,” in Proc. 5th WF-IoT. IEEE, 2019, pp. 257–262.
- 11)E. Korkan, H. B. Hassine, V. E. Schlott, S. Käbis, and S. Steinhorst, “WoTify: A platform to bring Web of Things to your devices,” arXiv preprint arXiv:1909.03296, 2019.
- 12)<https://wotify.org>
- 13)L. Sciallo, C. Aguzzi, M. Di Felice, and T. S. Cinotti, “WoT Store: Enabling Things and Applications Discovery for the W3C Web of Things,” in Proc. 16th CCNC. IEEE, 2019, pp. 1–8.
- 14)<https://iot.mozilla.org/gateway/>

## Download this Presentation:

- PowerPoint (PPTX): <https://1drv.ms/p/s!AmKo76mC4ZS5k78BO8PRReyfEe1qxcg?e=uhX6oQ>
- PDF: <https://1drv.ms/b/s!AmKo76mC4ZS5k78CIjDGQQISY2ANLg?e=5u2Cuj>

# Backup Slides – Overview

Backup Slides – MoT Overview



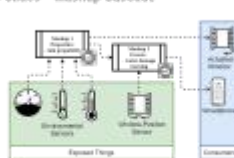
© 2005 Blackwell Publishing Ltd, *Journal of Internal Medicine* 258: 103–110

Backup Slides – Atomic Mashup



© 2005 Blackwell Publishing Ltd *Journal of Internal Medicine* 258: 259–267

Backup Slides – Mashup Cascade



100 *Journal of Health Politics, Policy and Law*

Backup Slides – Sequence Diagram – Elements



© 2005 Blackwell Publishing Ltd *Journal of Internal Medicine* 258: 105–112

Backup Slides – SD Keywords 1

[illegible]

(16) T. B. Edrington & R. A. Hunt: *Web-IT*: Rapid System/Procedure for Representative Effects.

Backup Slides – SD Keywords 2

name	Constructs the application of a meta-theory.
name	Constructs the application logic for a meta-theory if the conditions of a meta-theory are not known.
name	Constructs the application logic for a meta-theory if the conditions of a meta-theory are known.
name, $\alpha$ , $\beta$	Can be used to construct $\alpha$ and $\beta$ from two existing theories of base theories, which are constructed with the respective (floating) application.
test	Can be used to check whether $\alpha$ and $\beta$ are mutually modified conditions, which is known.
	Typically, determines whether a theory should be constructed independently or not.
type	Type of a theory, can derive the structure or instance.
type, $\alpha$ , $\beta$	Can optionally be used to determine where the value of $\alpha$ and $\beta$ , which is constructed of a floating property, is assigned. This assignment can be known, partially or only on request.
type, $\alpha$ , $\beta$	Can optionally be used to determine the content of an instance $\alpha$ of $\beta$ for one or several values.

© 2005-2010 R. Ratti Web-ITNego System-Description for Representative Attachment

### Backup Slides – SD Object Types

Type of item	Represent
abstract	abstract expression, called <i>Atomic Mapping</i>
map	a time delay before further rule execution
case	a conditional execution
loop	the repeating execution of another rule atom
get	getter function for Mapping variable, projection of default values
set	setter function for Mapping or projection
ref	reference to an action or a function

© 2004 Blackwell Publishing Ltd *Journal of Internal Medicine* 255: 299–307

Backup Slides – SD longer extract

20 T.M. El-Eidi, A. Hant, W.B. J.T. Negi / System Dynamics for Reproductive Health

Backup Slides – JSON Schema Example



100 978-0-13-035207-0 © 2010 Pearson Education, Inc. All rights reserved. This publication is protected by copyright. Any unauthorized distribution or reproduction of this work is illegal. This publication may be used for educational or promotional purposes. All other rights are reserved.

Backup Slides – Sequence Diagram Example 2



[8] T.M.B. & S.C. A. (2007) *A New System Description for Representation of*

Backup Slides – Algorithms Pseudo – Conversion

```

1 procedure PLOTDEGREE (v: VERTEX)
2   draw inputP(v) at Node(v);
3   for degree d in inputP(v) at Node(v) do
4     for e in degree d do
5       multiplicity ← MULTIMULTIPLY(e);
6   procedure GENERATEDEGREEDESCRIPTIONS
7     for degree d in inputP(v) at Node(v) do
8       for e in degree d do
9         if e is not null then
10           NodeDegree ← createNodeDegree(e);
11           AddNodeDegree ← createNodeDegree(e);
12           NodeDegree ← add NodeDegree to NodeDegreeList;

```

© 2001 Blackwell Science Ltd, *Journal of Internal Medicine* 250: 105–112

Backup Slides – Algorithms Pseudo – Code Generation

```

1: procedure GENERATEINDEX
2:   for protocol  $\ell$  from ExposeProtocol to SetImplementation do
3:     index  $\leftarrow$  IncludeWebAPI(protocol, linking(protocol))
4:     metadata  $\leftarrow$  parseSystemDescription(indexSD())
5:     role  $\leftarrow$  addVariables(indexBasis, addConstructors(indexSD()))
6:   procedure GENERATEINDEXFROMCONC
7:     while metadata.role  $\neq$  null do
8:       role  $\leftarrow$  generateRoleForIndexConc(indexSD(), role)
9:       if metadata.role  $\neq$  null then
10:         generateIndexCode(role, ConcIndex)
11:       metadata.role  $\leftarrow$  null

```

© 2005 Blackwell Publishing Ltd *Journal of Internal Medicine* 258: 105–112

Backup Slides – Evaluation – Videos

Case Study 2

© 2005 Blackwell Publishing Ltd, *Journal of Internal Medicine* 258: 105–112

Case Study 3

# Approach – WoT Sequence Diagram – Definition

- Subset of UML Sequence Diagrams standard
- PlantUML as implementation
- Grammar to specify the subset

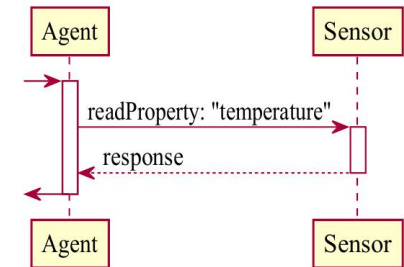
```
1  interactionReceive ::= getset?
2  |
3  |   interactionPre (
4  |       receiveSubs |
5  |       receiveInv |
6  |       receiveObs |
7  |       receiveRead
8  |   ) getset?;
9  interactionPre ::= "Agent" S '->' S
10 |
11 |   interactionTo S ':' S;
12 receiveRead ::= 'readProperty:' receiveMiddle
13 |
14 |   readResponse L deactTo L;
15 receiveMiddle ::= S interactionName L actTo L;
16 readResponse ::= interactionTo S '->' S S
17 |
18 |   "Agent" S ':' S 'response';
19 deactTo ::= 'deactivate' S interactionTo;
20 interactionTo ::= 'Ntitle';
21 <?TOKENS?>;
22 L ::= S? (#x000A | #x000D #x000A?)+ S?;
23 S ::= [#x0020#x0009]+;
24 Ntitle ::= [a-zA-Z] ([a-zA-Z0-9] | '-' | '_' )+;
```

Grammar



```
1  @startuml diagramName
2  [-> "Agent"
3  activate "Agent"
4  "Agent" -> "Sensor":readProperty: "temperature"
5  activate "Sensor"
6  "Sensor" --> "Agent":response
7  deactivate "Sensor"
8  [<- "Agent"
9  deactivate "Agent"
10 @enduml
```

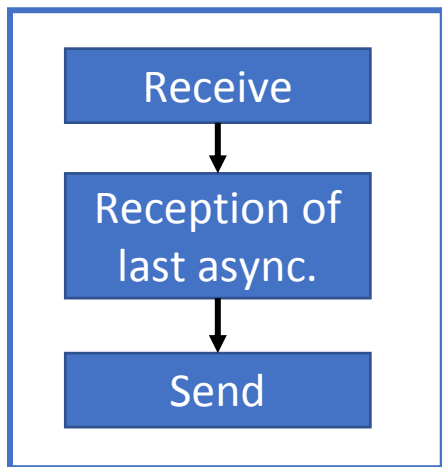
Notation Example



Presentation

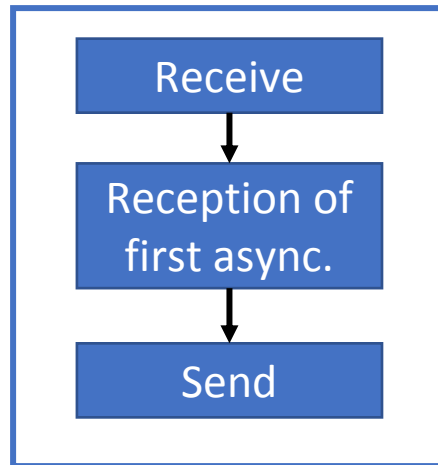
# Backup Slides – Atomic Mashup

Default



e.g., receive  
independent values

Break on  
data push



e.g., poll same physical  
value from different sensors

# Approach – Proof of Equality – Methodology

1. State formalisms to represent a WoT Mashup
2. Conclude formalisms from Sequence Diagram (and System Description)
3. Compare to show their equality
4. Conclude overall proof

1

$e := e_1 e_2 \dots e_n \mid \forall_{i \in n}: e_i \in C, n \geq 1$   
Application logic sequence  
C Set of all application logic elements

2

$\text{strict}(x, y) := xy \mid$   
 $X, Y \in X, Y$  Not further constraint sets

3

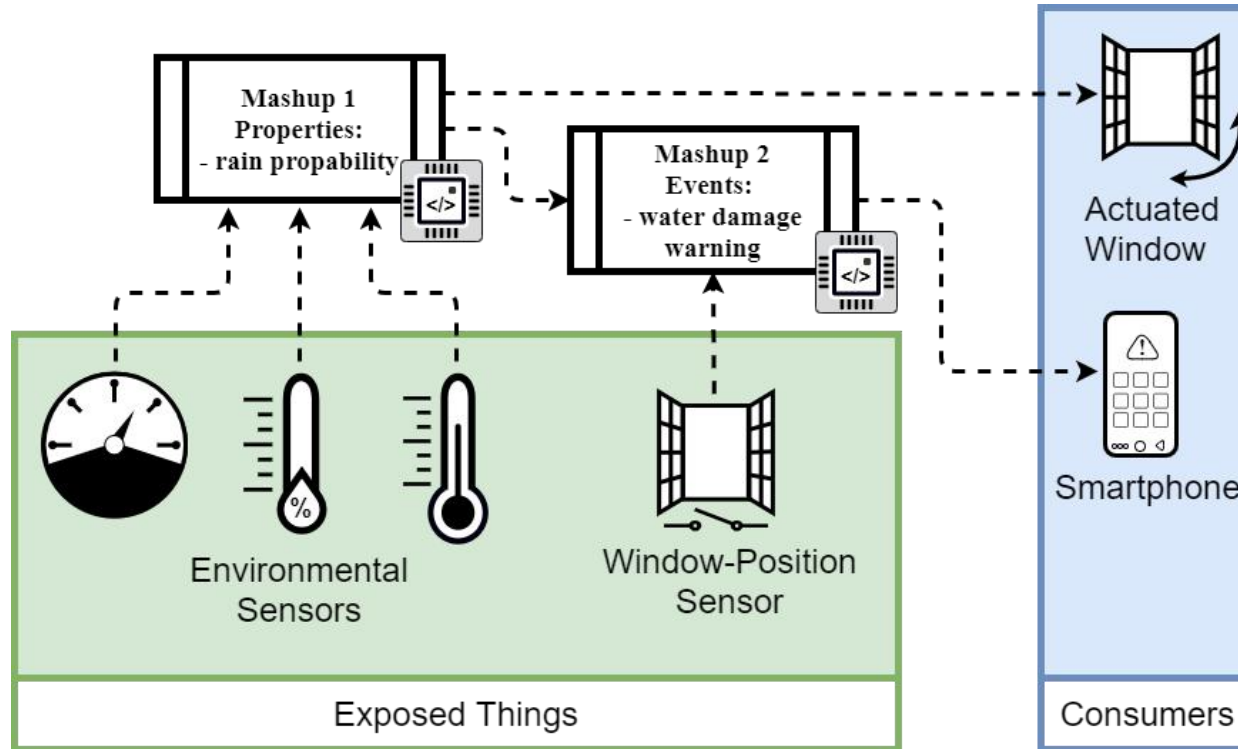
Concatenation of elements.

4

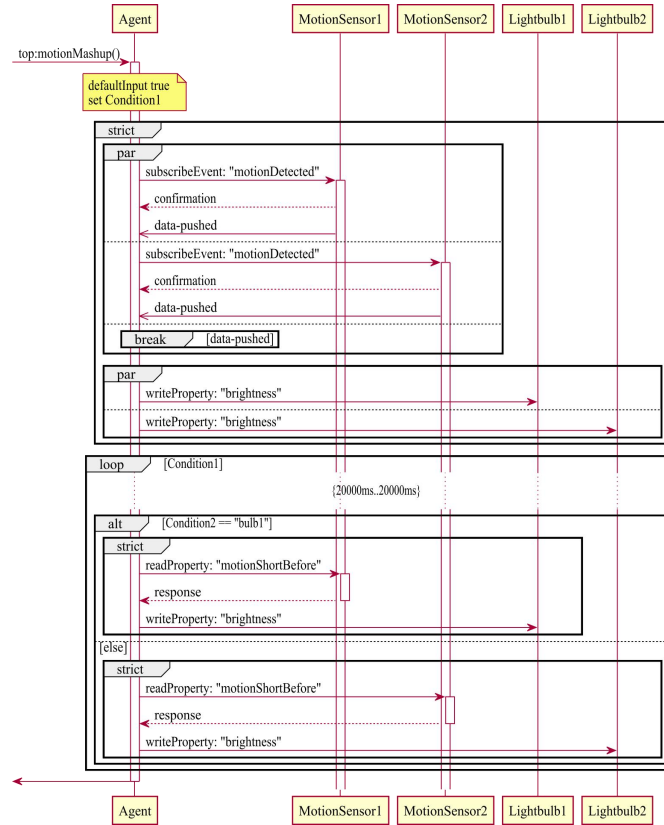
$\text{SequenceDiagram}(m) \iff \text{SystemDescription}(m)$

With  $m \in M$  the set of possible Mashups

# Backup Slides – Mashup Cascade



# Backup Slides – Sequence Diagram – Elements





# Backup Slides – SD Keywords 1

loop	Loop element of a path array.
defaultInput	Input to setter functions, loop count/period, case comparison or default value of variables.
\$ref	Reference to a variable, property, function or action. Is chosen because of the known meaning in JavaScript Object Notation (JSON) Schema.
wait	Pause execution command of a path array.
interact	Atomic Mashup element of a path array, contains <i>send</i> , <i>receive</i> and <i>breakOnDataPushed</i> .
send	Array of sending interactions, part of an Atomic Mashup.
receive	Array of receiving interactions, part of an Atomic Mashup.
breakOn-DataPushed	Defines when to execute sending interactions in an Atomic Mashup.
set	Setter function in a path array or as part of a receiving interaction.
get	Getter function in a path array or as part of another element which requires a variable as input, e.g., <i>send</i> , <i>loop</i> , <i>case</i> .
case	Conditional execution (if statement) parent keyword, element of a path array. Contains <i>if</i> , <i>else</i> (optionally) and <i>then</i> .

## Backup Slides – SD Keywords 2

if	Contains the condition of a <i>case</i> element.
then	Contains the application logic to be executed if the condition of a <i>case</i> is true.
else	Contains the application logic to be executed if the condition of a <i>case</i> is false.
oneOf, allOf, anyOf	Can be used as child elements of <i>if</i> and have to contain arrays of further conditions, which are computed with the respective Boolean operator.
not	Can be used as child element of <i>if</i> and has to contain another condition, which is inverted.
sync	Optionally determines whether a loop should be executed synchronously or asynchronously.
type	Type of a loop, can have the value <i>timed</i> or <i>logical</i> .
isUpdated-OnDemand	Can optionally be used to determine when the value of a path, which is child element of a Mashup's property, is computed. The computation can happen periodically or only on request.
defaultOutput	Can optionally be used to determine the output of an action if it has no other value to return.

# Backup Slides – SD Object Types

Type of item	Represents
interact	interaction sequence, called Atomic Mashup
wait	a time delay before further task execution
case	a conditional execution
loop	the repeating execution of another <i>path</i> array
get	getter function for Mashup variables, properties or default values
set	setter function for variables or properties
ref	reference to an action or a function

# Backup Slides – SD longer extract

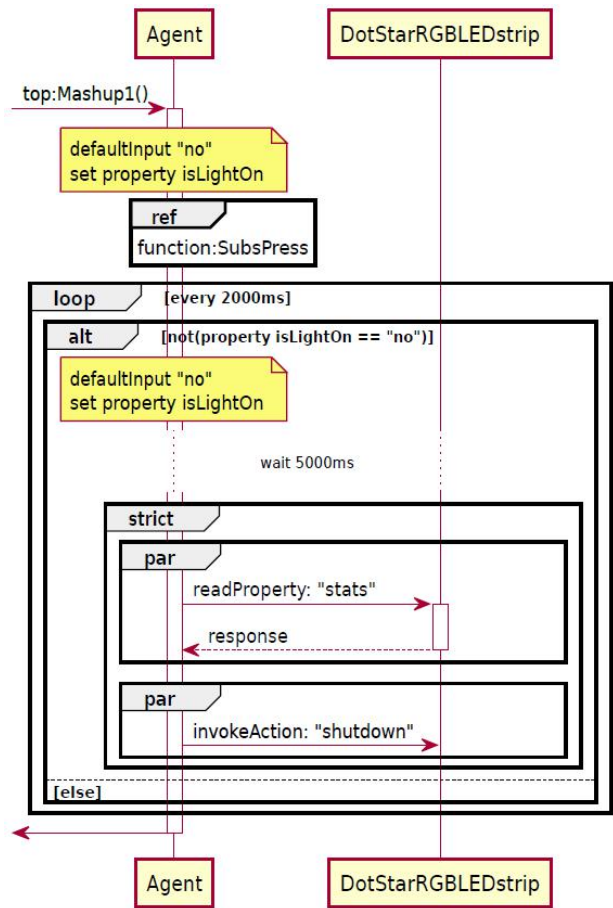
```
1 {
2 >   "@context": [ ...
7   ],
8   "id": "de:tum:ei:esi:MashDE:VirtualCoffeeMachineTestThing:",
9   "@type": "Thing",
10  "title": "mashupExample",
11  "things": {
12 >    "VirtualCoffeeMachine": { ...
29  },
30 >    "TestThing": { ...
46  }
47  },
48  "variables": {
49    "Var1": {}
50  },
51 >  "properties": { ...
59  },
60  "path": [
61    {
62      "set": {
63        "$ref": "#/variables/Var1"
64      },
65      "defaultInput": "exampleString"
66    },
67    {
68      "receive": [
69        {
70          "form": {
71            "$ref": "#VirtualCoffeeMachine/events/maintenance/forms/0"
72          },
73          "op": "subscribeevent"
74        }
75      ],
76      "send": [
77        {
78          "form": {
79            "$ref": "#TestThing/properties/string/forms/0"
80          },
81          "op": "writeproperty"
82        }
83      ],
84      "breakOnDataPushed": false
85    }
86  ]
87 }
```

# Backup Slides – JSON Schema Example

```
1 {
2   "title": "SD Validation Schema",
3   "description": "A schema to validate a system description",
4   "$schema": "https://json-schema.org/draft-07/schema#",
5   "$comment": "require SD context?!",
6   "definitions": {
7     "tdValidation": {
8       "$id": "#tdSchema",
9       "$ref": "https://raw.githubusercontent.com/thingweb/thingweb-playground/master/WebContent/td-schema.json",
10      "description": "A schema to validate that the system description is a valid thing description, this schema",
11    },
12    "variable": {
13      "type": "object",
14      "properties": {
15        "type": {
16          "type": "string",
17          "enum": [
18            "boolean",
19            "object",
20            "array",
21            "number",
22            "integer",
23            "string"
24          ]
25        },
26        "defaultInput": {
27          "description": "The initial value of the variable"
28        }
29      }
30    }
31  }
```



# Backup Slides – Sequence Diagram Example 2





# Backup Slides – Algorithms Pseudo – Conversion

```
1: procedure PARSESEQUENCEDIAGRAM
2:   clean inputPlantUmlNotation;
3:   for diagram  $\in$  inputPlantUmlNotation do
4:     for line  $\in$  diagram do
5:       mashupLogic  $\leftarrow$  uml_to_internal(line);
6: procedure GENERATESYSTEMDESCRIPTION
7:   SystemDescription  $\leftarrow$  generate_SD_Template();
8:   for logicArray  $\in$  mashupLogic do
9:     for element  $\in$  logicArray do
10:      if element[form] not in AddedForms then
11:        SystemDescription  $\leftarrow$  element[form]
12:        AddedForms  $\leftarrow$  element[form]
13:      SystemDescription  $\leftarrow$  add_SD_Path_Element(element)
```

# Backup Slides – Algorithms Pseudo – Code Generation

```
1: procedure GENERATEINDEX
2:   for protocol  $\in$  (confExposeProtocol  $\vee$  SdInteractionForms) do
3:     index  $\leftarrow$  include_WoT_API_protocol_binding(protocol);
4: mashupLogic  $\leftarrow$  parse_System_Description(inputSD);
5: code  $\leftarrow$  add_variables_handlers_and_classConstructor(inputSD);
6: procedure GENERATEMASHUPCODE
7:   while mashupLogic  $\neq$  null do
8:     code  $\leftarrow$  generate_code_for_logic(mashupLogic[0])
9:     if mashupLogic[0] has logicContent then
10:       generateMashupCode(logicContent)
11:     mashupLogic.removeIndex(0)
```



# Backup Slides – Evaluation – Videos

## Case Study 2

[https://1drv.ms/v/s!AmKo76mC4ZS5k7573\\_H69oUmbATyIA?e=VAEGDh](https://1drv.ms/v/s!AmKo76mC4ZS5k7573_H69oUmbATyIA?e=VAEGDh)

## Case Study 3

<https://1drv.ms/v/s!AmKo76mC4ZS5k758DjnnKp0ThrZr6A?e=4qIrMV>