# Rigorous Design and Deployment of IoT Applications

**Ajay Krishna** | Michel Le Pallec | Radu Mateescu | Ludovic Noirie | Gwen Salaün

FormaliSE 2019: International Conference on Formal Methods in Software Engineering

27 May 2019, Montreal, QC, Canada

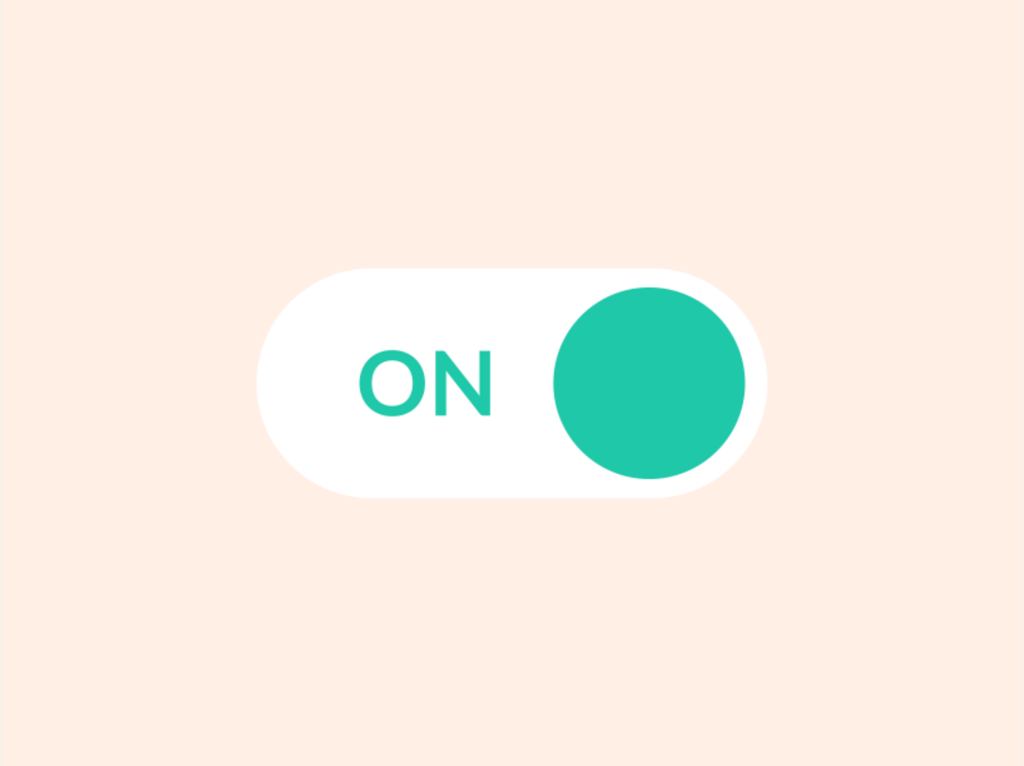
# Context

Internet of Things (IoT) applications are heterogenous and concurrent

Difficult for end-users to build correct applications by composing various objects

Formal modeling and verification can help users build correct applications

# Modelling behavior: Switch



# Modelling behavior: Philips Hue Lamp


Off


On


Change Color

# Interface-based Modeling

Off

On

Change Color

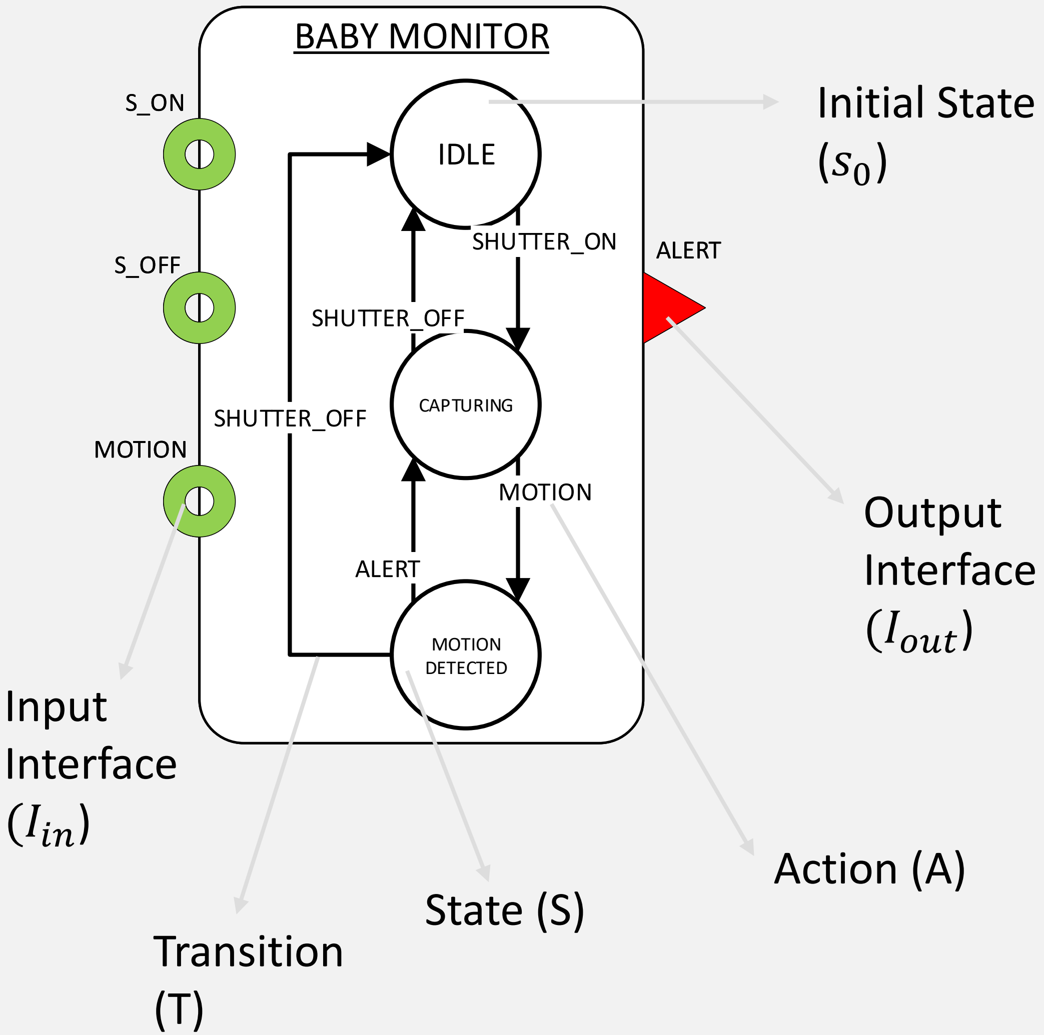
https://meethue.com/api/key/lights/1/state

```
{
"on": false
}
```

```
{
"on": true
}
```

```
{
"hue": 50000,
"bri": 200 }
```

# IoT Object Model




$$O = < I_{in}, I_{out}, LTS >$$

$$LTS = < S, A, T, s_0 >$$

$$A \subseteq I_{in} \cup I_{out} \cup \{ \tau \}$$

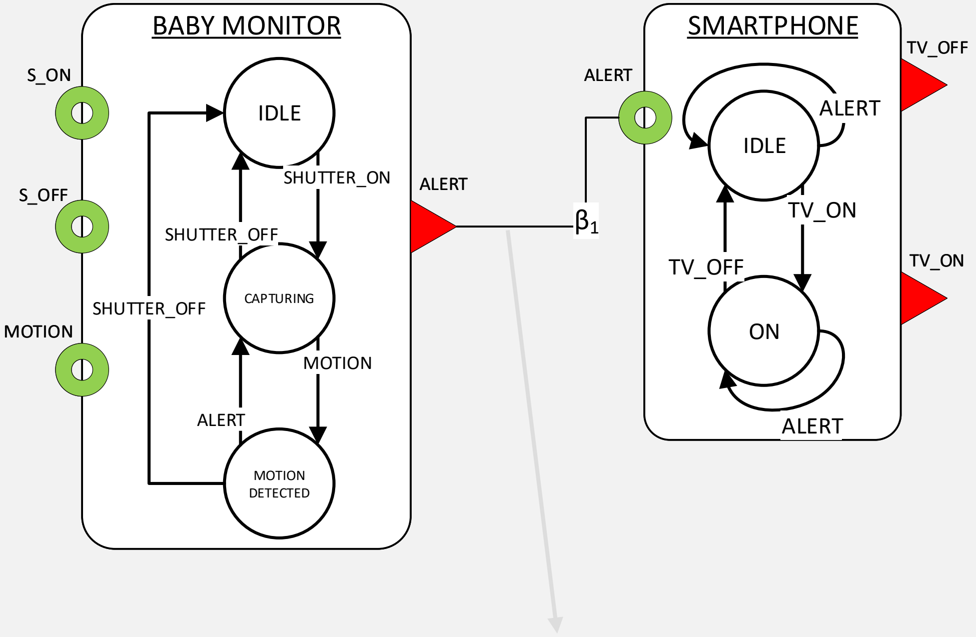
$$A \subseteq I_{in} \cup I_{out} \cup \{ \tau \} \;\; (s_1, a, s_2) \in T$$

$$s_0 \in S$$

$$I_{in} = \{ S\_ON, S\_OFF, MOTION \}$$

$$I_{out} = \{ ALERT \}$$

# Bindings




Binding
$(\beta)$

Binding $\beta = (i_{out}^{O_1}, i_{in}^{O_2})$

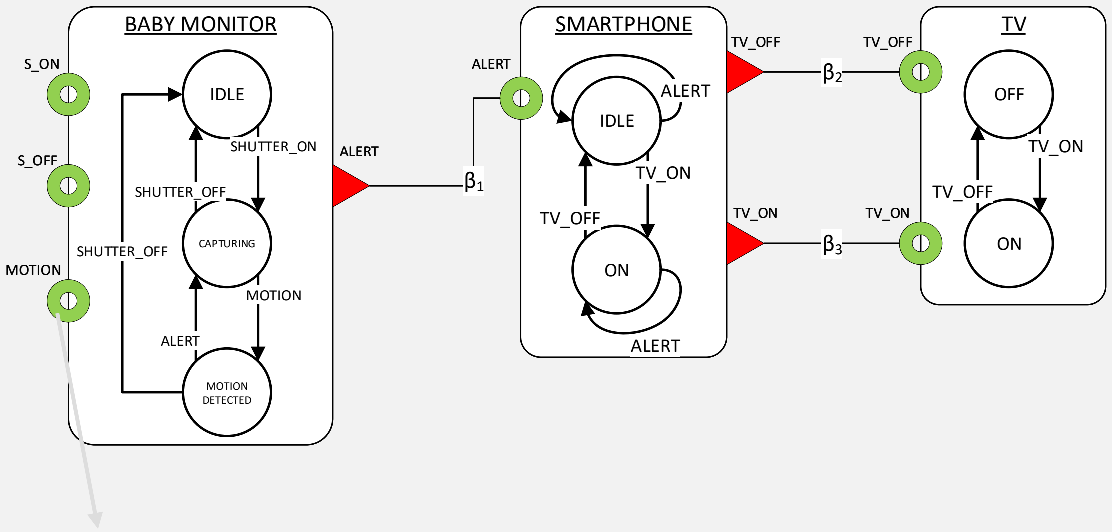
$\text{in}(\beta) = i_{in}^{O_2}$

$\text{out}(\beta) = i_{out}^{O_1}$

Notion of **Strong** and *Weak* Bindings

**Strong** : Functionally important
*Weak*: Optional (can provide additional service)

# Composition




Unbound interface ($I_{in}^{U}$)

$$C = < B\ , \Sigma, I_{in}^{U}, I_{out}^{U}, LTS, W\ >$$

$$B\ = \{\ \beta_1, \beta_2, \dots \beta_n\}$$

$$\Sigma = \{\ O_1, O_2, \dots O_n\}$$

$$I_{in}^{U}\ = \left(I_{in_1} \cup I_{in_2} \cup\ \ \dots \cup I_{in_n}\right) \setminus in(B)$$

$$I_{out}^{U}\ = \left(I_{out_1} \cup I_{out_2} \cup\ \ \dots \cup I_{out_n}\right) \setminus out(B)$$

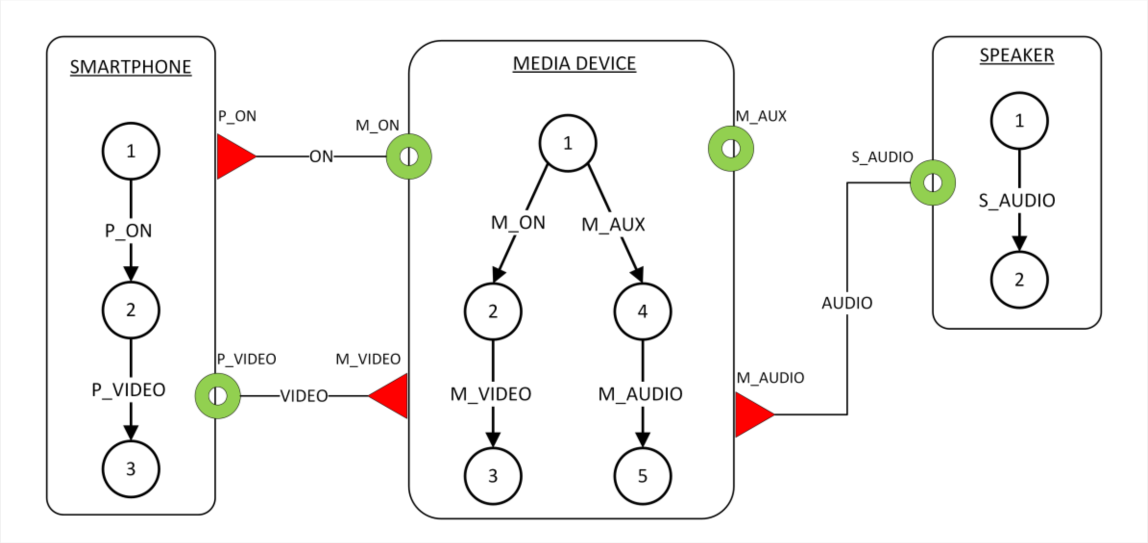
$$LTS\ = ren_{\rho B}(LTS_1) \otimes_B \dots \otimes_B ren_{\rho B}(LTS_n)$$

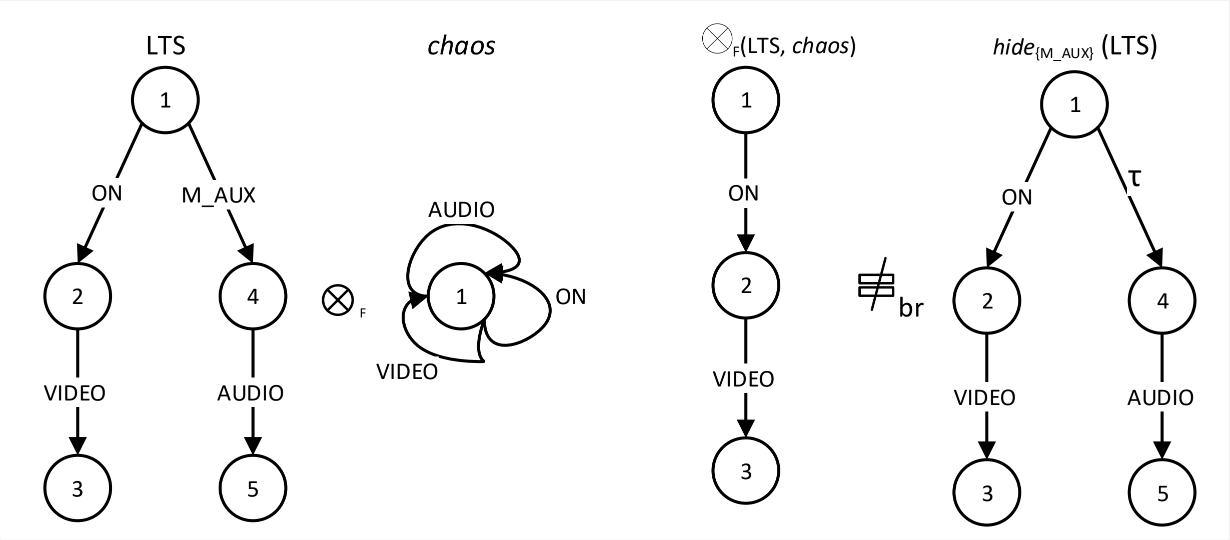
$W\ \subseteq B$    //weak bindings

# Compatibility: Intuition

A service composition is correct if all bindings can effectively be executed and if all reachable actions unbound in the composition do not prevent the bindings to be executed

# Compatibility: Example




$$hide_{(I_{in}^{U} \cup I_{out}^{U})(LTS)}(LTS) \equiv_{br} LTS \otimes_{A} chaos_{B}$$

# Deployment

Bindings describe dependencies among the objects

A composition can be viewed as a directed graph. Objects as nodes and bindings as edges

Dependencies can be identified by inverse topological sorting

Discard weak bindings in case of cyclic dependencies

# Deployment Plan

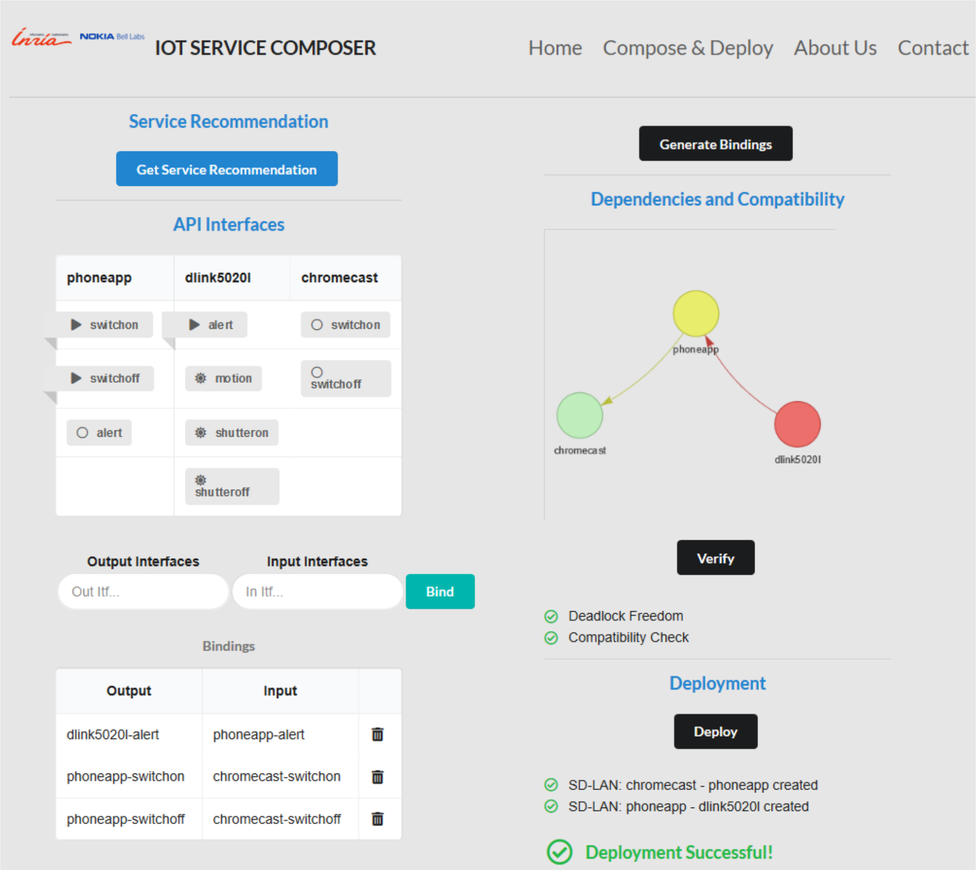
A deployment plan consists of sequence of steps involving 3 operations at network level (SDN)

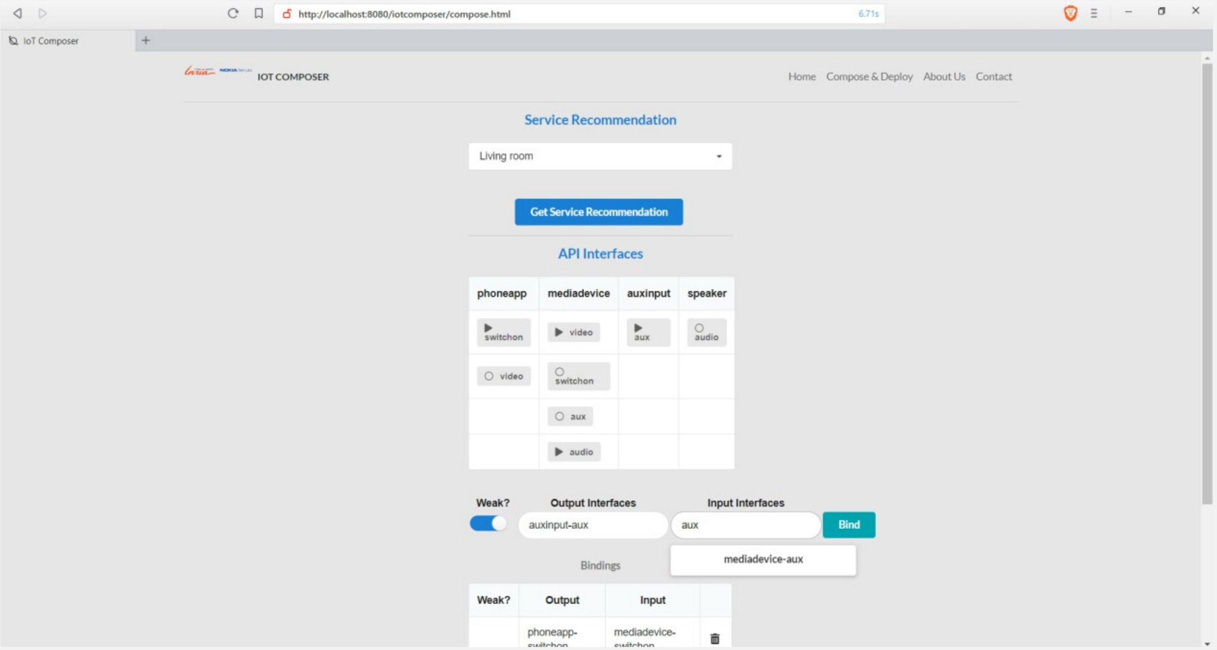
ADD – Provisioning of connected objects

BIND – Network configuration to allow communication between app interfaces

START – Enables app interfaces

# Tool Support: IoT Composer



# LNT Specification

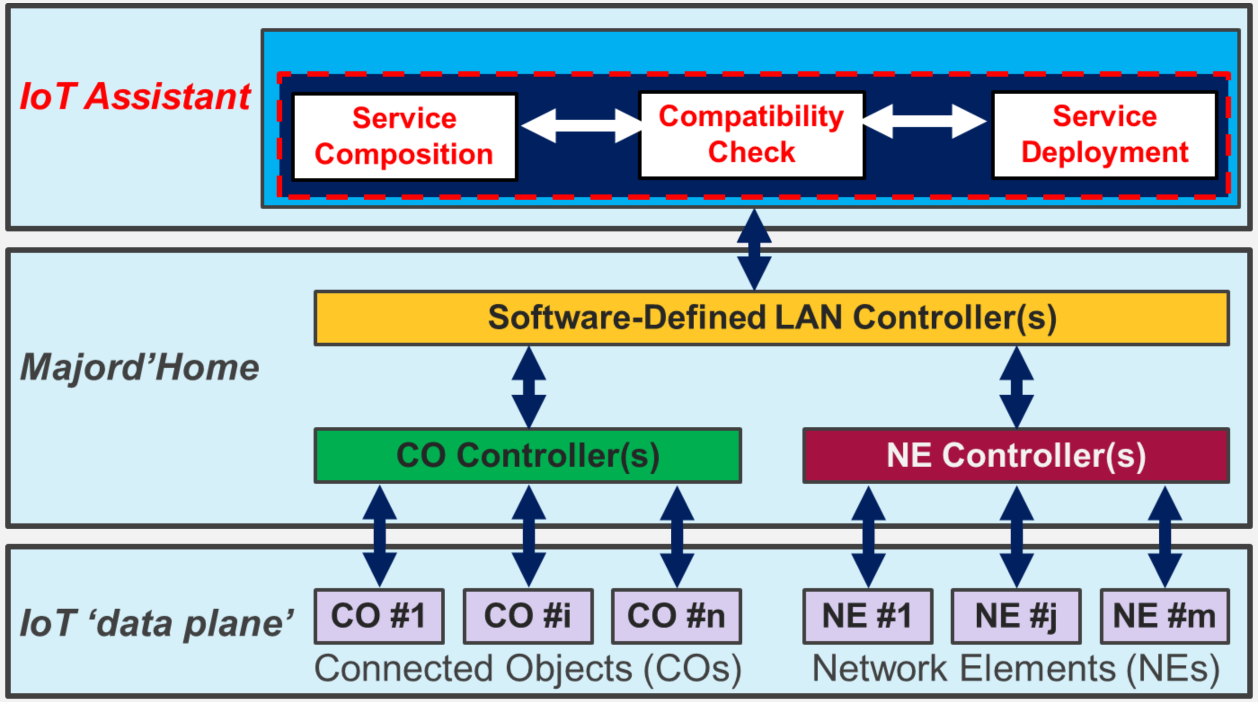
Object

```
module mediadevice is
 process mediadevice_idle[on,video,aux,audio:any] is
  select
   on; mediadevice_video[on,video,aux,audio]
  []
   aux; mediadevice_audio[on,video,aux,audio]
  end select
 end process
 process mediadevice_video[on,video,aux,audio:any] is
  video; mediadevice_exit[on,video,aux,audio]
 end process
 process mediadevice_audio[on,video,aux,audio:any] is
  audio; mediadevice_exit[on,video,aux,audio]
 end process
 process mediadevice_exit[on,video,aux,audio:any] is
  stop
 end process
end module
```

Composition

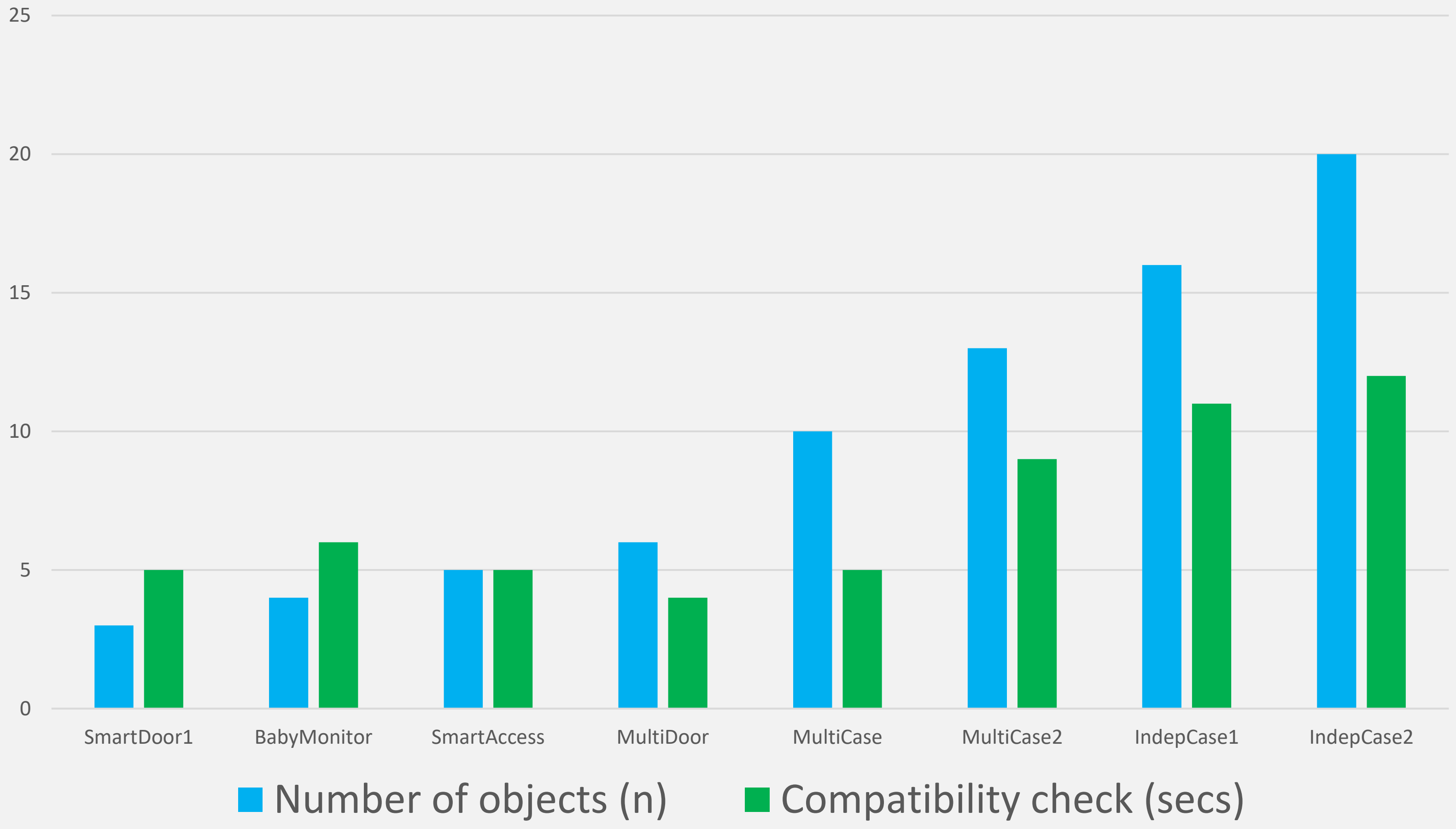
```
module prodall (phone,mediadevice,speaker) is
process prod [on,video,aux,audio:any] is
  par
    on,video -> phone_idle[on,video]
   ||
    on,video,audio -> mediadevice_idle[on,video,aux,audio]
   ||
    audio -> speaker_idle[audio]
  end par
end process
end module
```

# SDN-based Deployment



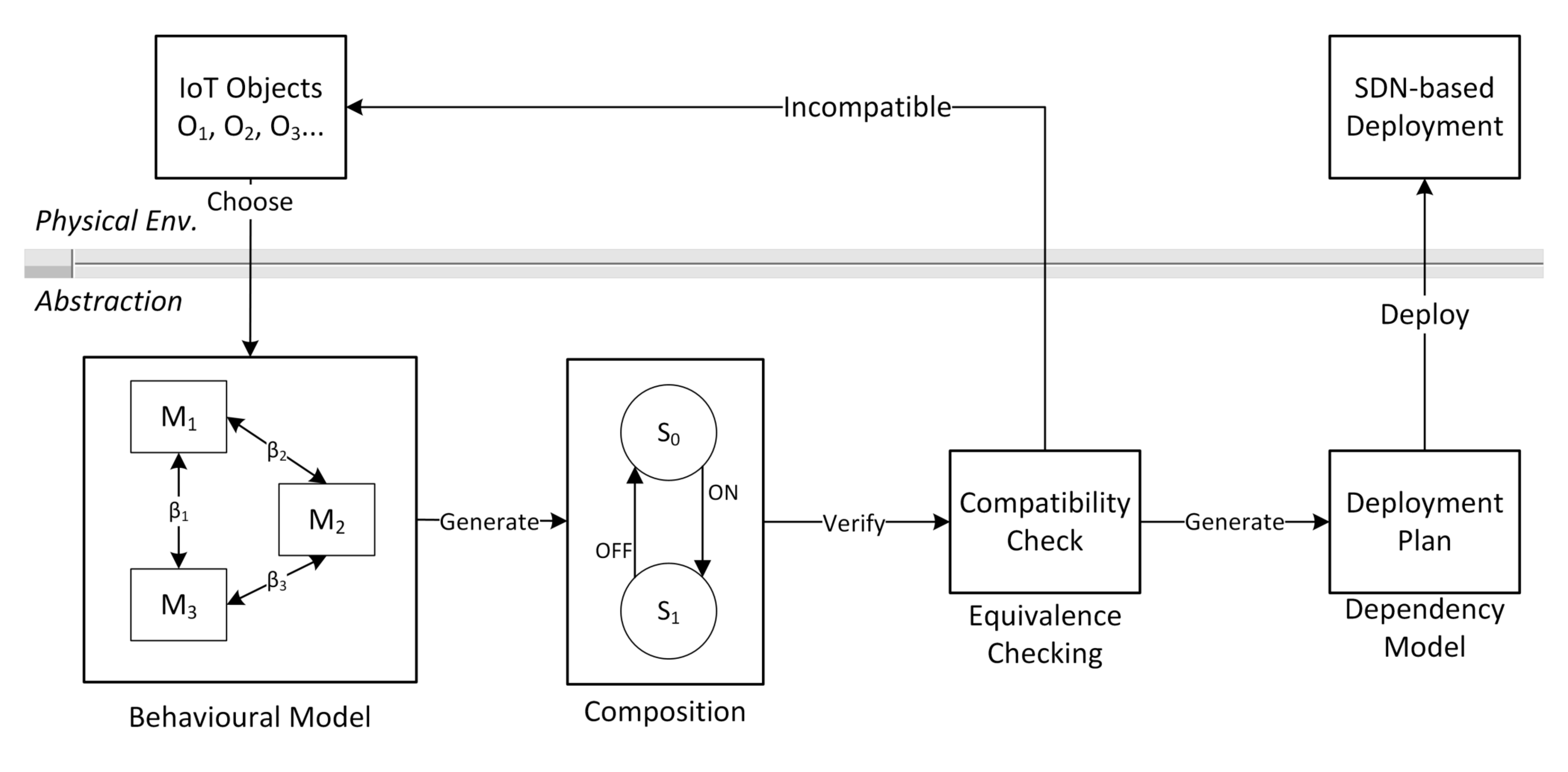

Objects can interact with different IoT services but preserves network isolation and discovery properties

# Experiments




| Use case | Objs | Bind | LTS States | LTS Trans. |
|---|---|---|---|---|
| Media | 3 | 3 | 5 | 4 |
| Thermo | 3 | 4 | 3 | 7 |
| SmartDoor1 | 3 | 4 | 8 | 13 |
| LightSense | 4 | 6 | 4 | 8 |
| SmartDoor2 | 4 | 6 | 6 | 7 |
| BabyMonitor | 4 | 6 | 13 | 17 |
| SmartAccess | 5 | 8 | 8 | 9 |
| MultiDoor | 6 | 6 | 6 | 7 |
| MultiCase | 10 | 12 | 36 | 154 |
| MultiCase2 | 13 | 15 | 176 | 892 |
| IndepCase1 | 16 | 18 | 876 | 5134 |
| IndepCase2 | 20 | 24 | 10501 | 79886 |

# Summary


IoT Objects
O1, O2, O3...
Incompatible
SDN-based Deployment
Choose
Physical Env.
Abstraction
Deploy
M1
β2
β1
M2
M3
β3
Generate
S0
ON
OFF
S1
Verify
Compatibility Check
Generate
Deployment Plan
Equivalence Checking
Dependency Model
Behavioural Model
Composition

# Concluding Remarks

Formal modelling and analysis contributes to correct composition and deployment

Proposals implemented as a tool for end-users

Future work on Thing Description, QoS and performance analysis of IoT services, and reconfiguration scenarios.