**How secure is your smart home?**

**Correctness and Security for Home Automation**

**Contributions**
- Architecture
- Security policies:
  - dependency policy
  - control policy
  - new item policy

**Goal:** verify whether interacting smart devices behave correctly

**Dependency Policy**
Controllers should send commands to dumb devices depending on sensor values.

**Complete specification:**
controller sends command $\iff$ \(<\text{sensor\_variable}=\text{state}> +$

**Partial specification:**
controller sends command $\iff$ \(<\text{sensor\_variable}=\text{state}> +$

**GARAGEDOOR\_CONTROLLER sends open\_garagedoor $\iff$**
(\(\neg\) \(\text{IS\_GARAGE\_OPEN}\))
\(\land\) ((\(\neg\) \(\text{IS\_CAR\_INSIDE\_GARAGE}\) \(\land\) \(\text{CAR\_DISTANCE} \leq "50m" \land\) \(\text{CAR\_SPEED} \neq 0\))
\(\lor\) (( \(\text{IS\_CAR\_INSIDE\_GARAGE}\) \(\land\) \(\text{IS\_CAR\_RUNNING}\))
\(\land\) ((\(\neg\) \(\text{IS\_OWNER\_INSIDE\_CAR}\))

**DISHWASHER\_CONTROLLER sends start\_dishwasher $\iff$**
(\(\neg\) \(\text{IS\_DISHWASHER\_ON}\)) \(\land\) ( \(\text{IS\_DOOR\_CLOSED}\) \(\land\) (\(\neg\) \(\text{IS\_CLEANED}\))
\(\land\) (\(\neg\) \(\text{IS\_EMPTY}\))

**New Item Policy**
- new sensor: no\_action
- new dumb\_device: no\_action
- new controller: verify\_dependency\_policy (this)
  \(\land\) verify\_control\_policy (this)

**Architecture**

**Controllers:**
- thermostat
- location-sensor
- door-state-sensor
- commander
- dumb devices: heater, air-conditioner, door

**Sensors:**
- thermometer
- location-sensor
- door-state-sensor

**Control Policy**
A controller \(k\) maintains a list of commands, \(C = \{ c_1, c_2, ..., c_m \}\) and a list of dumb devices, \(D = \{ d_1, d_2, ..., d_n \}\). Each \(d_i \in D\) maintains a list of actions, \(A_{p_j} = \{ a_{i1}, a_{i2}, ..., a_{i_p} \}\) it can execute.

\(k\) should not send:
- right commands to wrong dumb devices
- wrong commands to right dumb devices
- wrong commands to wrong dumb devices

\(\forall\) command\_dumbdevice sent by \(k\),
\(\exists\) \(c_i \in C\) | command\_dumbdevice = \(c_i\)
\(\land\) \(\exists\) \(d_j \in D\) | dumbdevice = \(d_j\)
\(\land\) command \(\in A_{p_j}\)