

# An Architecture Description Language for Dynamic Sensor-Based Applications

---



**Humberto Cervantes**

Universidad Autonoma Metropolitana-Iztapalapa, Mexico  
hcm@xanum.uam.mx

**Didier Donsez, Lionel Touseau**

University Grenoble-1, France  
Didier.Donsez@imag.fr, Lionel.Touseau@imag.fr



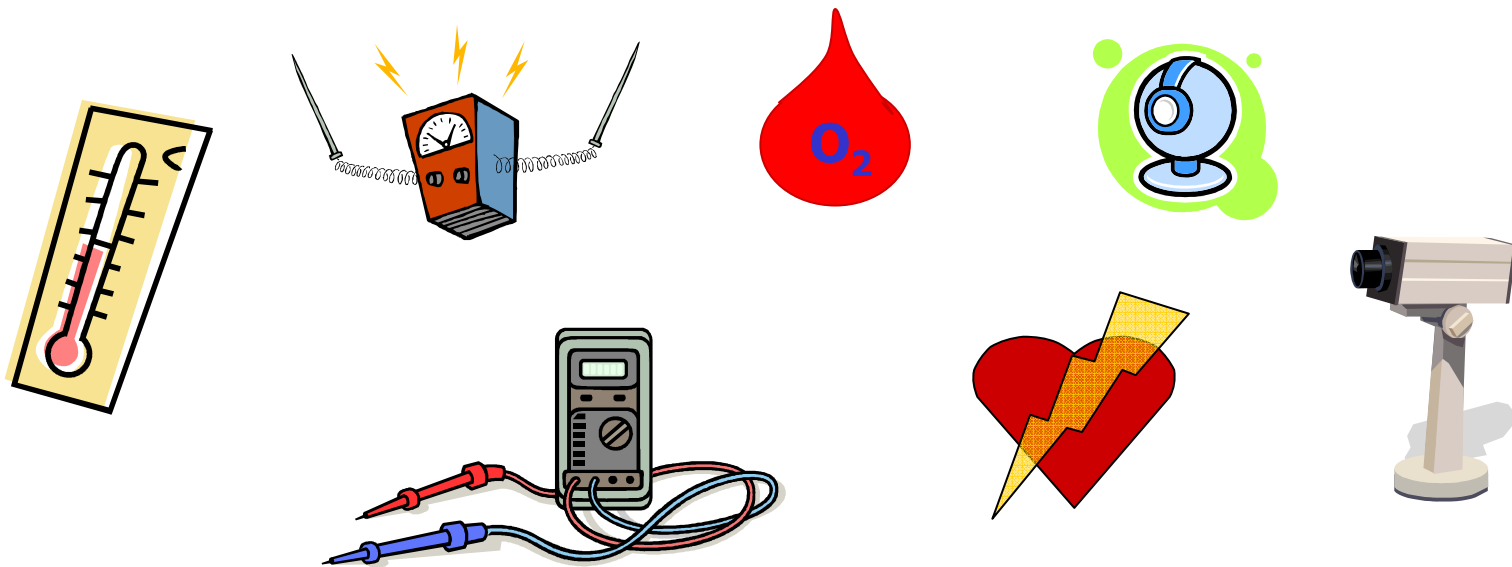
# Outline

---

- Machine-to-machine context
- Sensor-based applications
- A dynamic architecture description language
- WireAdmin binder implementation
- Examples
- Conclusion

# Context : machine-to-machine

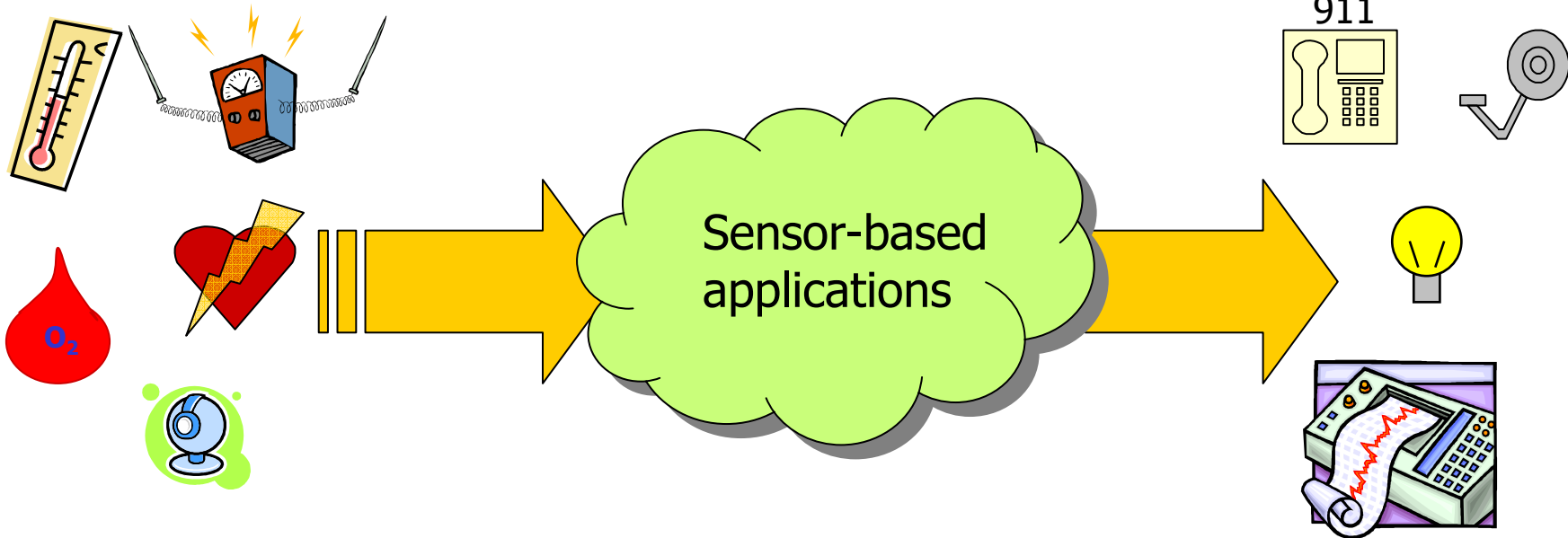
- New wave of e-business
  - ITU Internet report 2005, « The Internet of Things »
- Mainly sensor-based applications



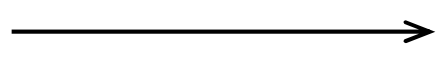
# Sensor-Based Applications

*Sensors*

*Actuators*



Producer



Consumer

Design pattern

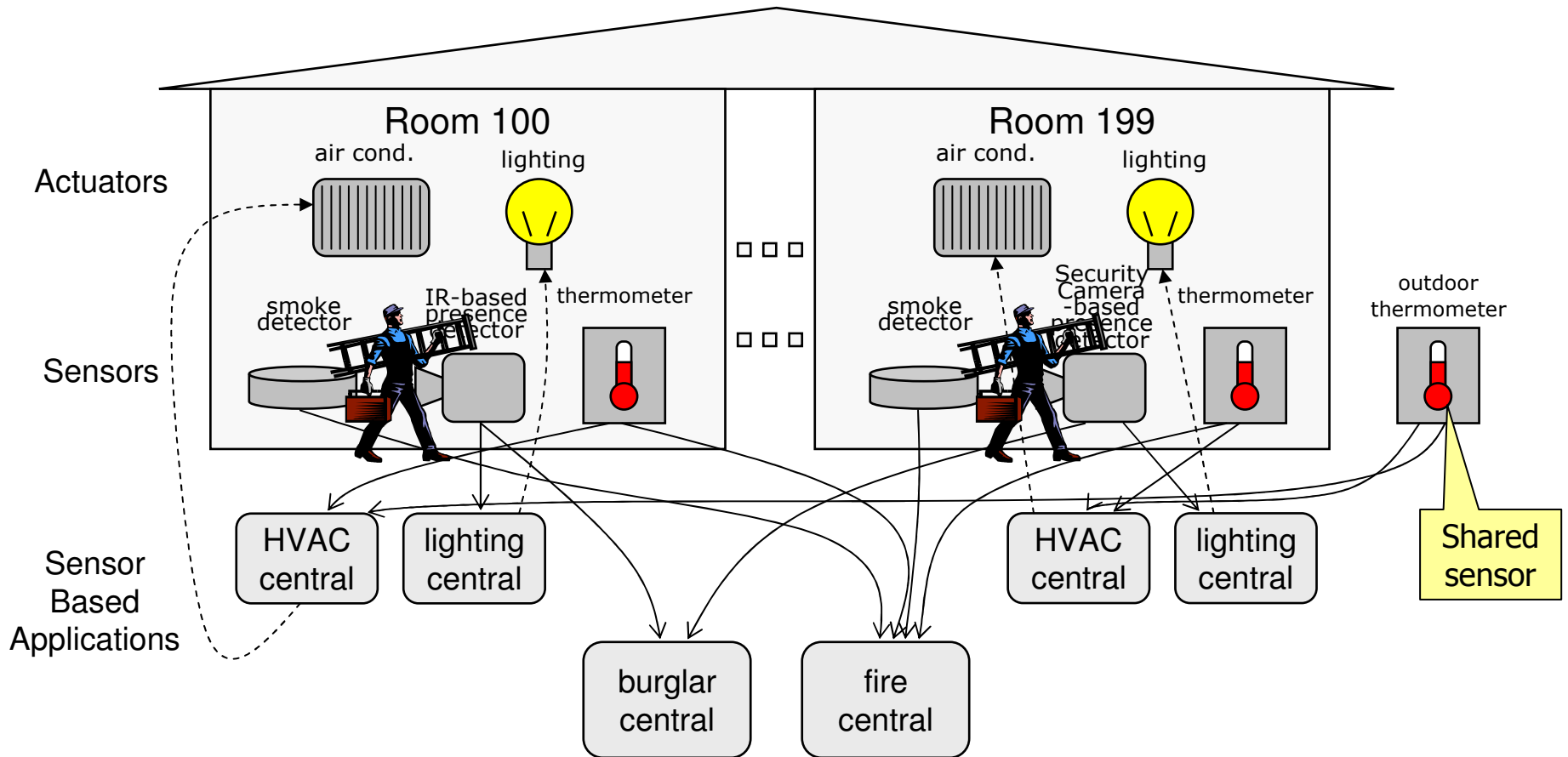
- Data processing chain

- Aggregation, filtering, correlation ( $P=U*I$ ), ...

# Example of a sensor-based application

## Small Office Home Office network

CCNC 2008, An Architecture Description Language for Dynamic Sensor-Based Applications



# Sensor-Based Applications Requirements

- More than 1000 rooms
- Complex hardware architectures
  - Large number of heterogeneous devices
  - → complex software architectures for SBA
- Taking into account evolution
  - Failures
  - Hardware or software maintenance
    - Addition / removal of sensors, actuators, applications, .
- Who can maintain these applications?
- Need for dynamic software adaptability
  - To support evolution



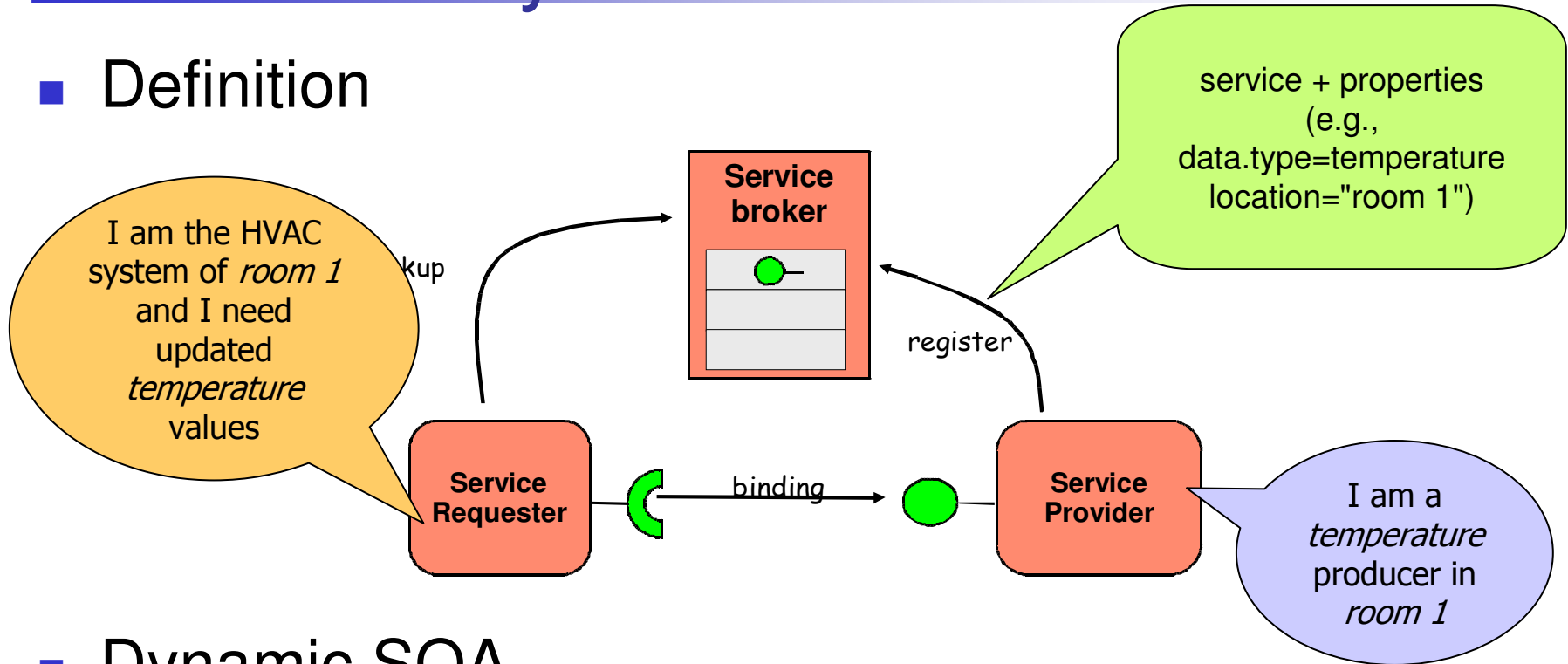
# Proposition

---

- Observations
  - Developers cannot foresee all components and the links between them
  - Complex applications are hardly maintainable
- Our approach
  - Enable loose coupling between producers and consumers through SOA
  - Describe **sets of possible bindings** between sensors and applications
    - Automate the bindings
    - Automate the application lifecycle
      - according to sensors and actuators availability

# SOA & Dynamic SOA

## ■ Definition

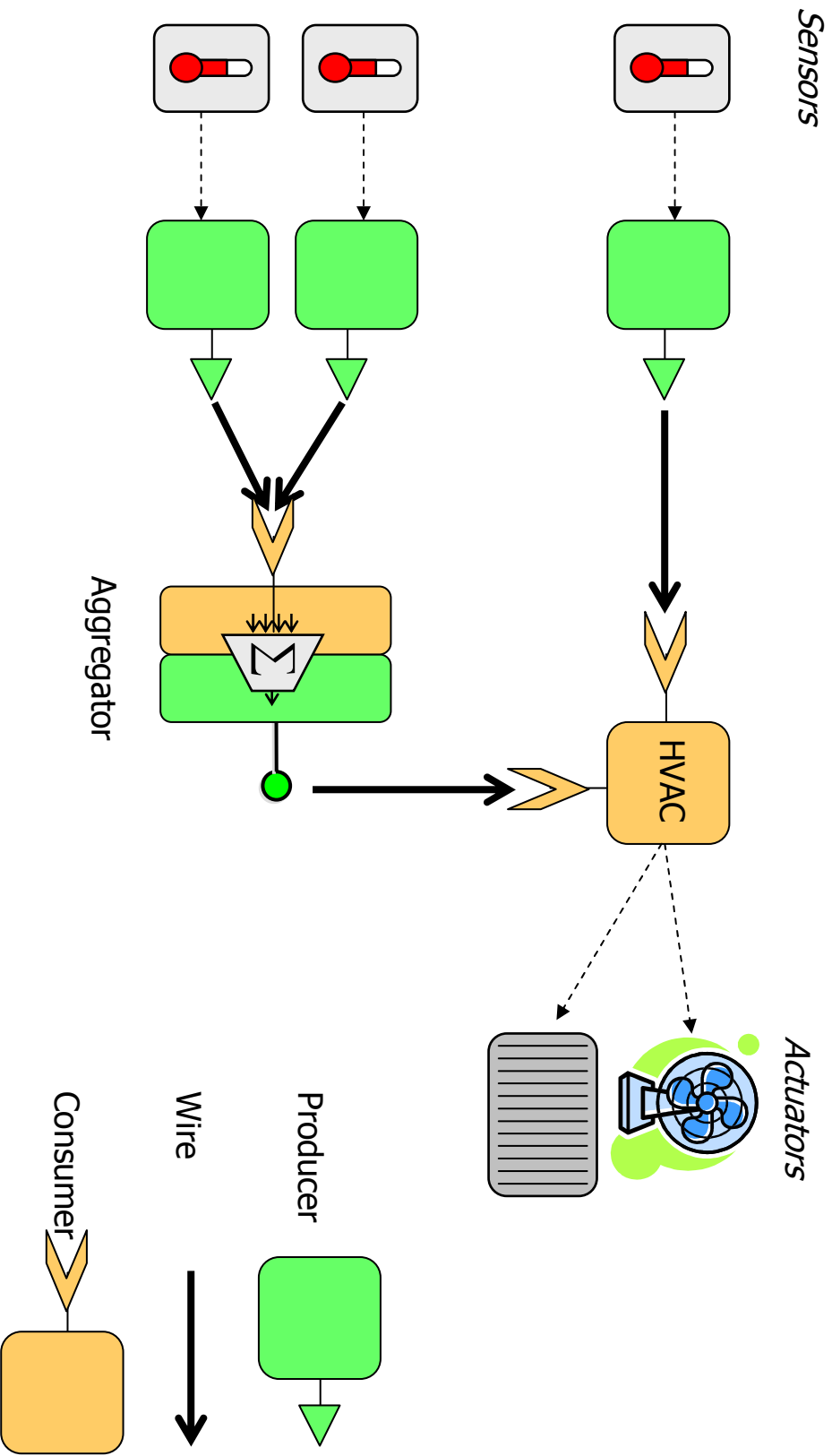


## ■ Dynamic SOA

- Components can be notified of service changes in the environment
- OSGi, UPnP, DPWS, ...

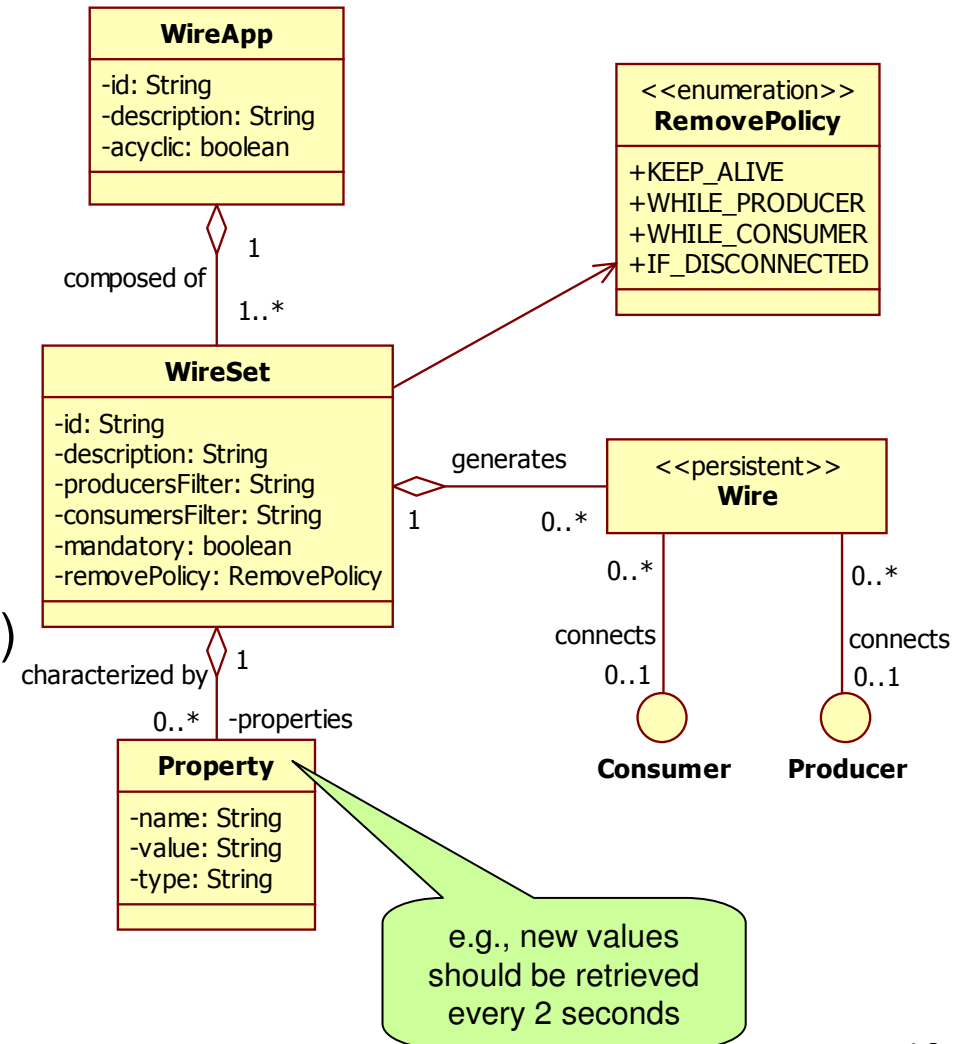


# Example: HVAC

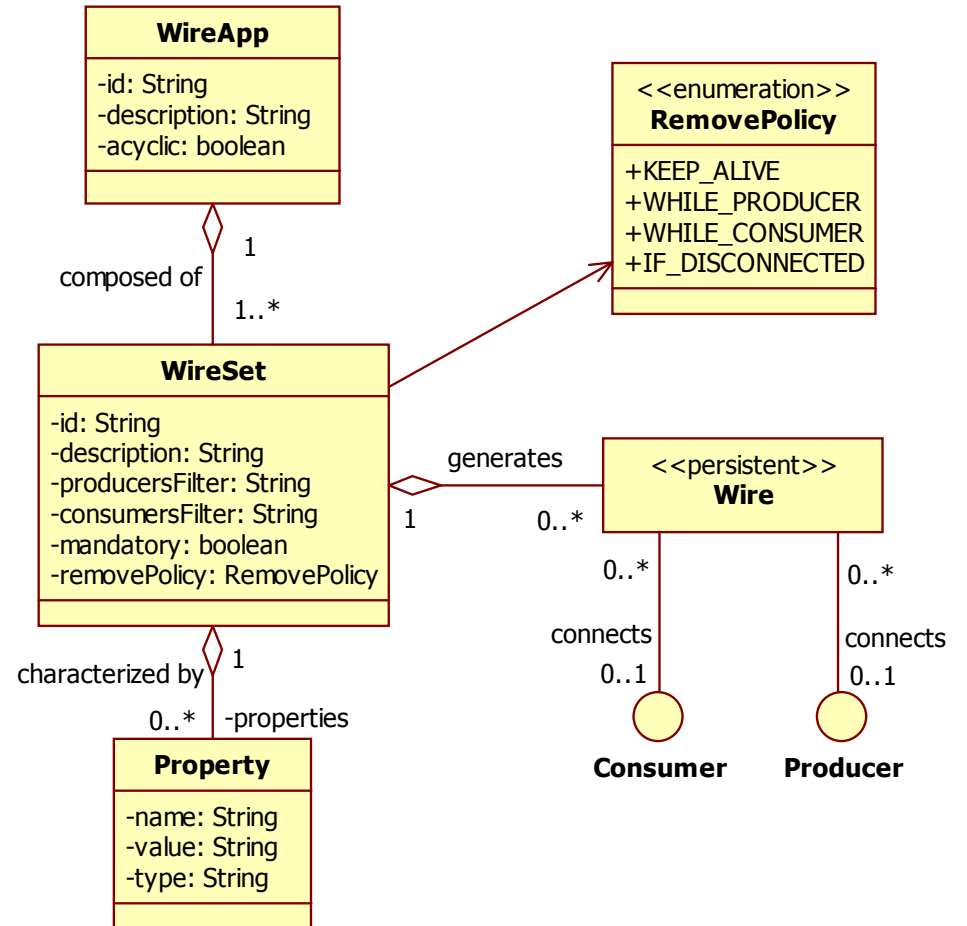


# WADL: an Architecture Description Language for Sensor Based Applications

- **WireApp**
  - A dynamic sensor-based application
- **WireSet**
  - Set of possible wires between producers and consumers according to types of data exchanged between them (+ constraints)
  - Define wire lifecycle
- **QoS Properties**
  - Parameters for the wire creation

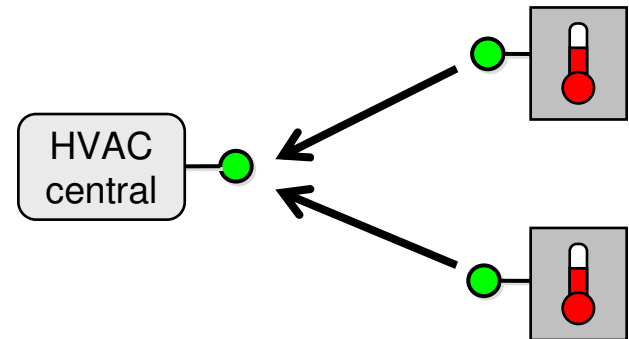
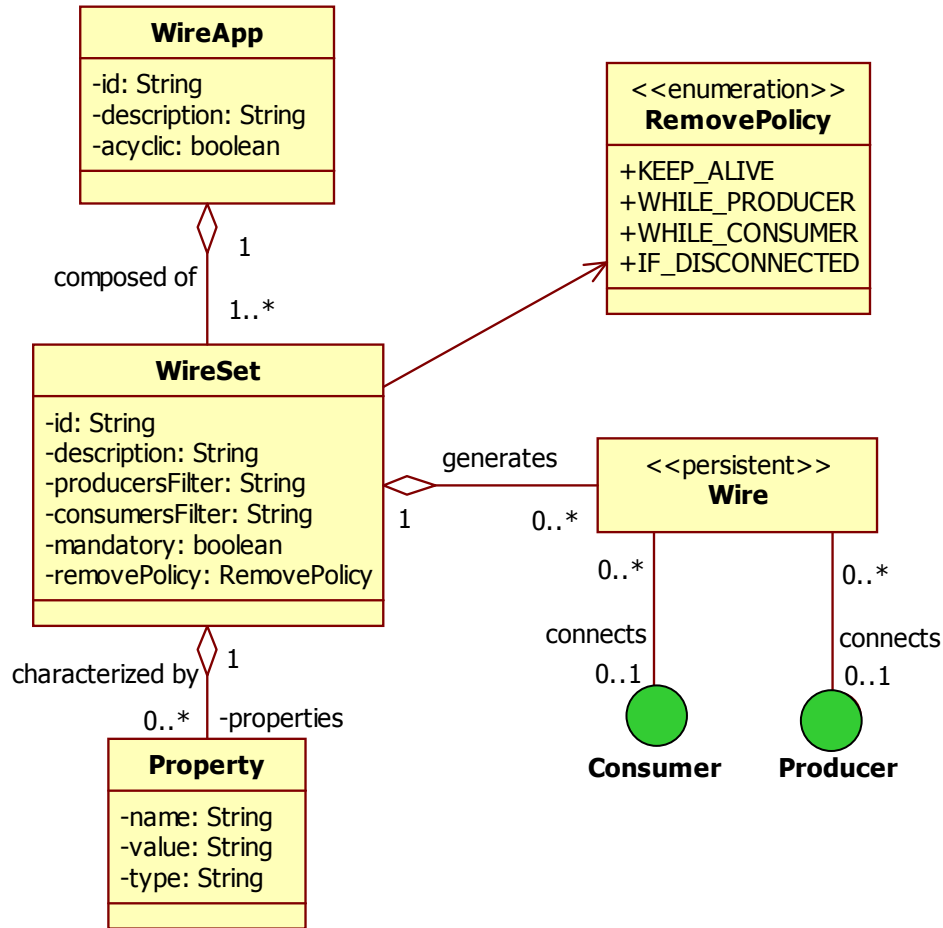


# WADL Wire Creation



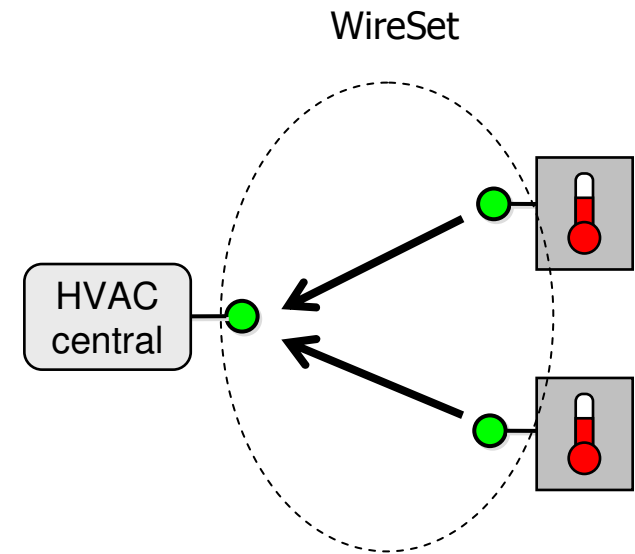
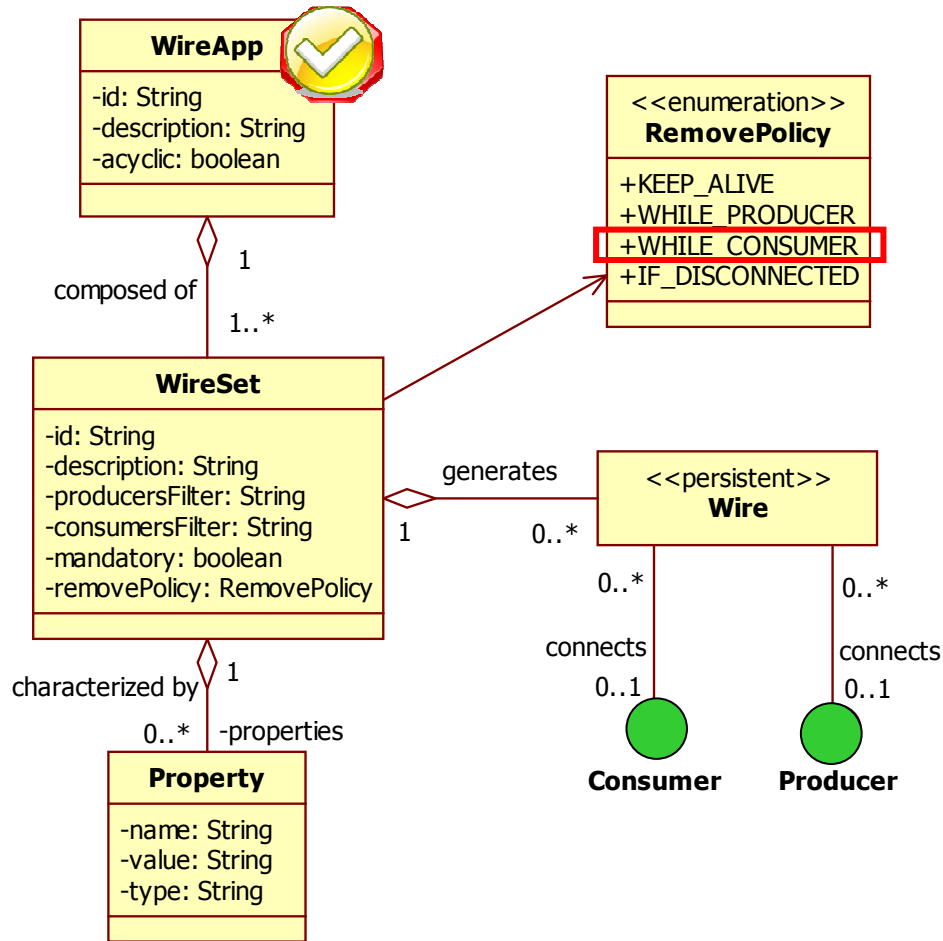
# WADL

## Wire Creation



# WADL

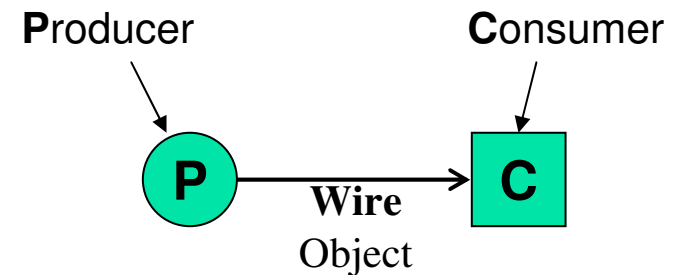
## WireApp Lifecycle



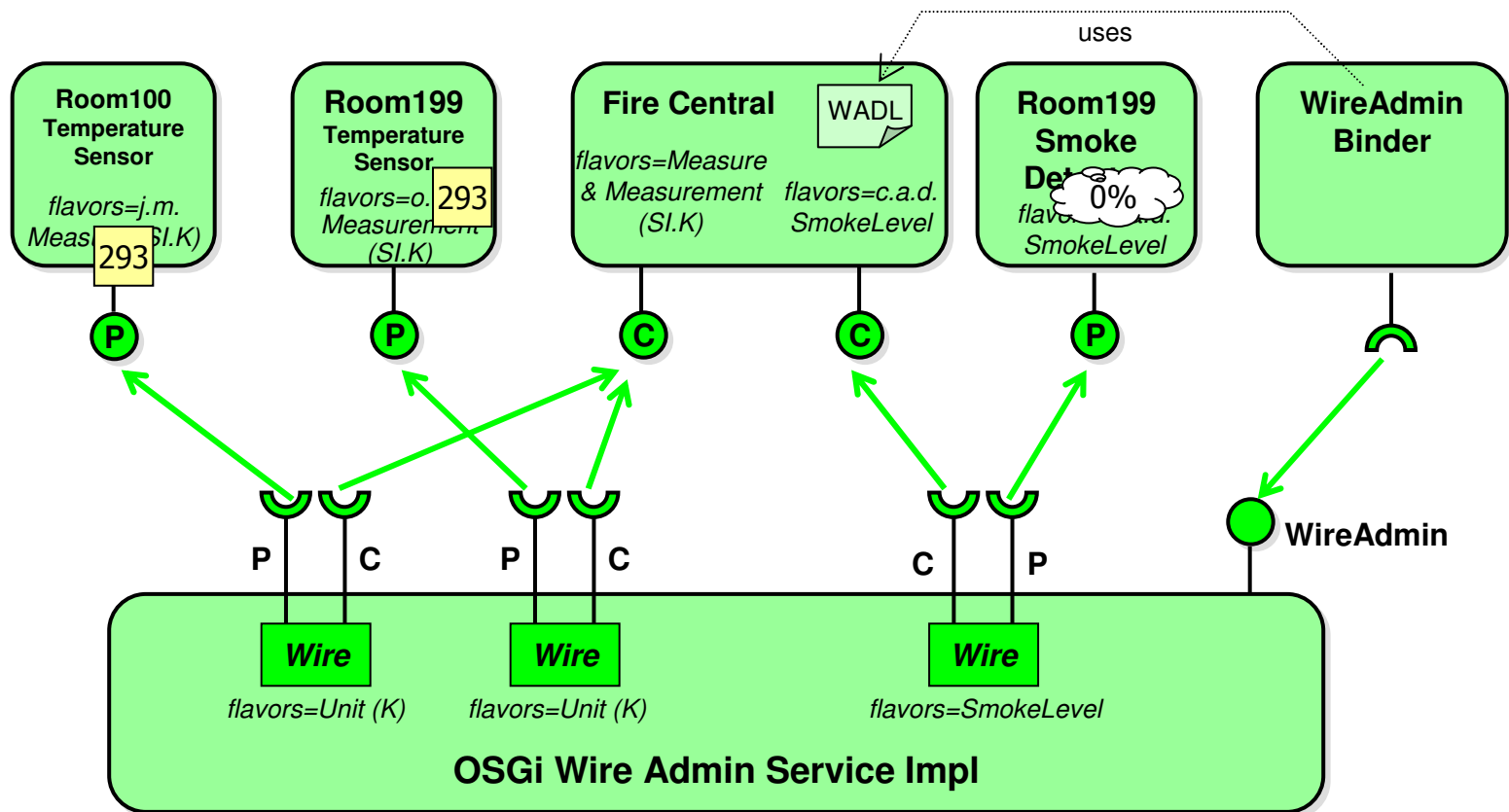
# WAB WireAdminBinder (WAB)

## Implementation for the OSGi WireAdmin service

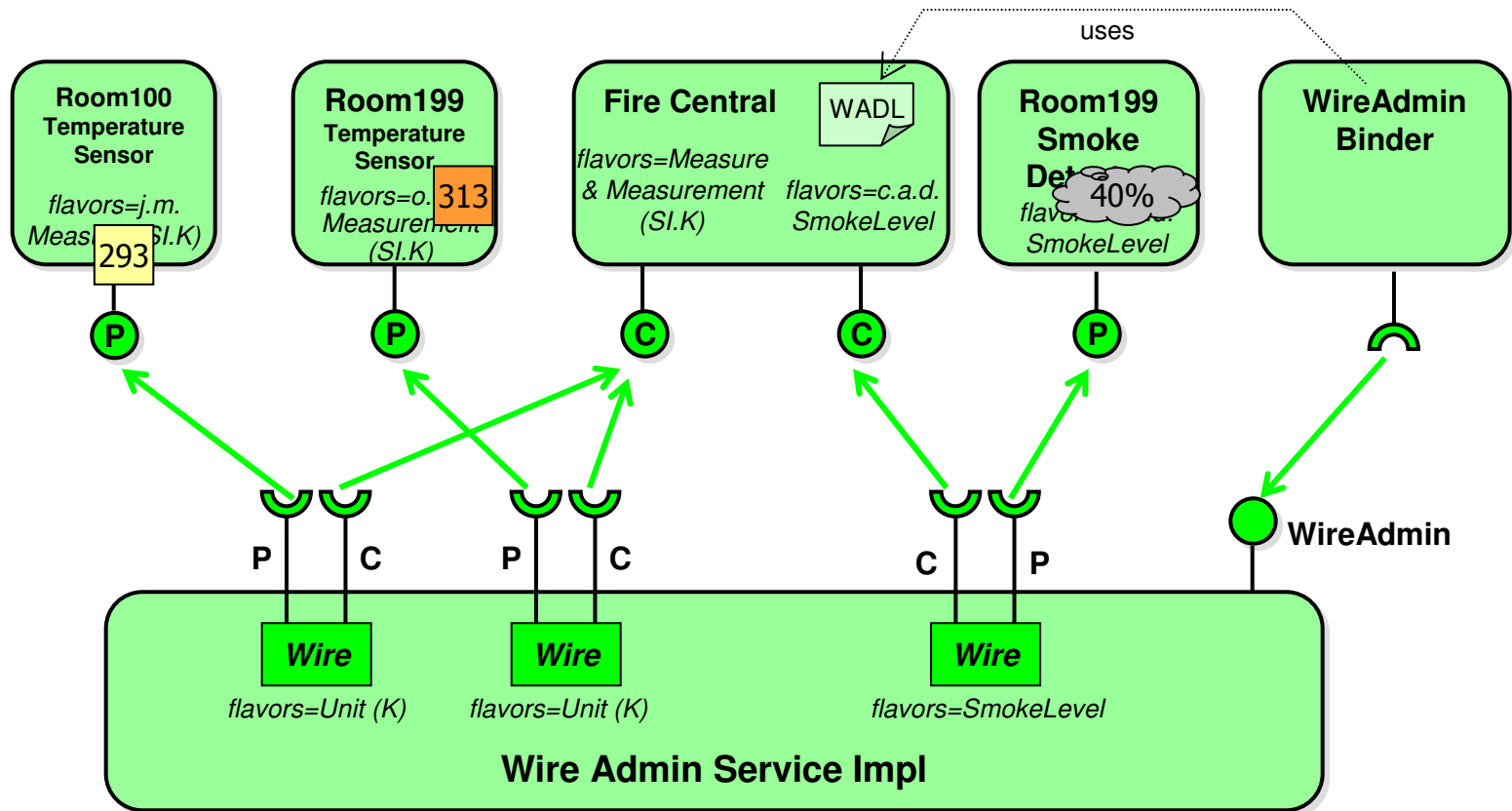
- WADL manager
  - creates and deletes wires through the OSGi WireAdmin service
- WireAdmin Service
  - Specification of a middleware implementing the producer/consumer design pattern for the OSGi framework
  - Manages wires lifecycles
    - Creation / deletion
  - Data values in the wire are
    - updated by producers
    - polled by consumer



# Example of an application wired by WireAdminBinder

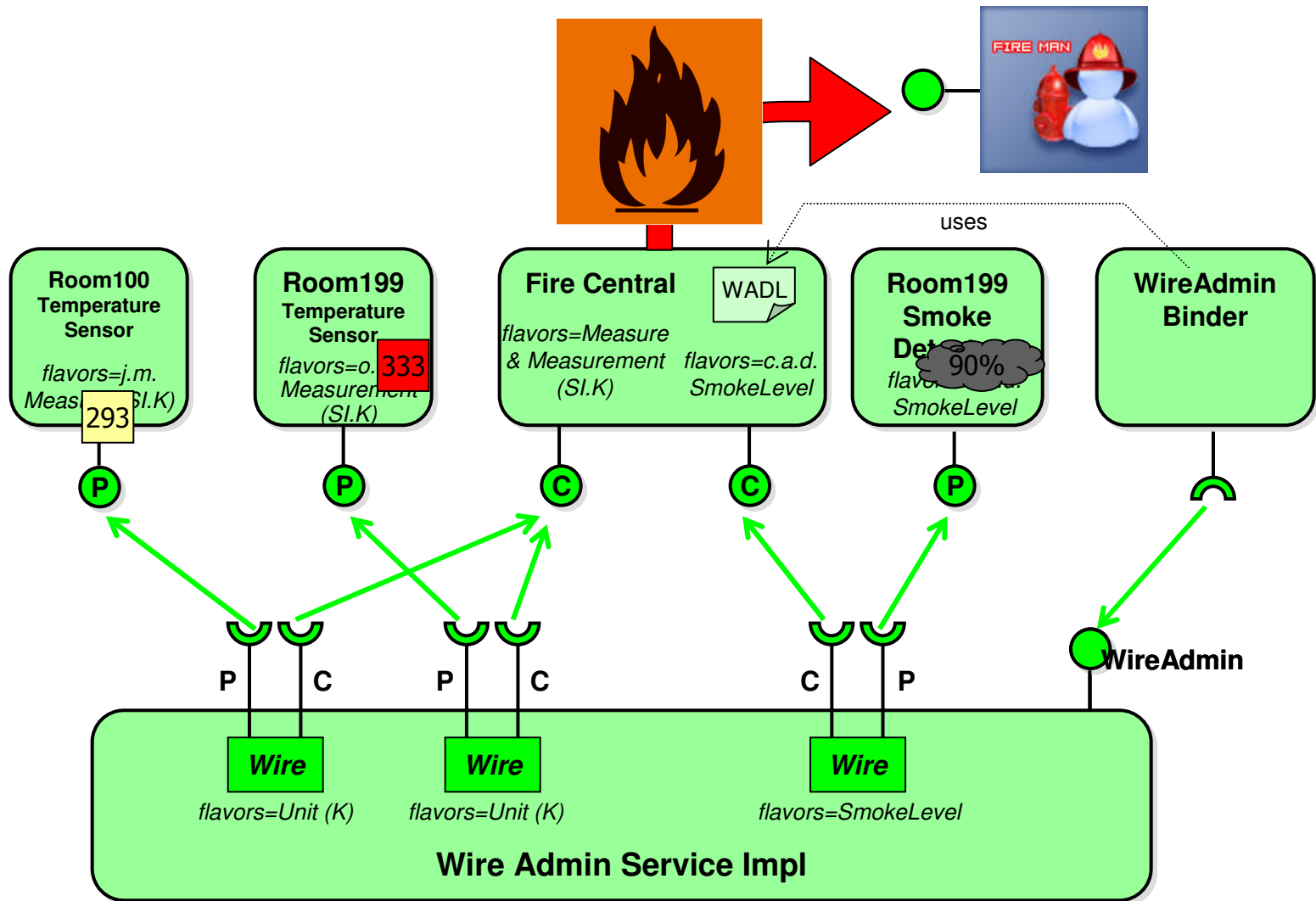


# Example of an application wired by WireAdminBinder



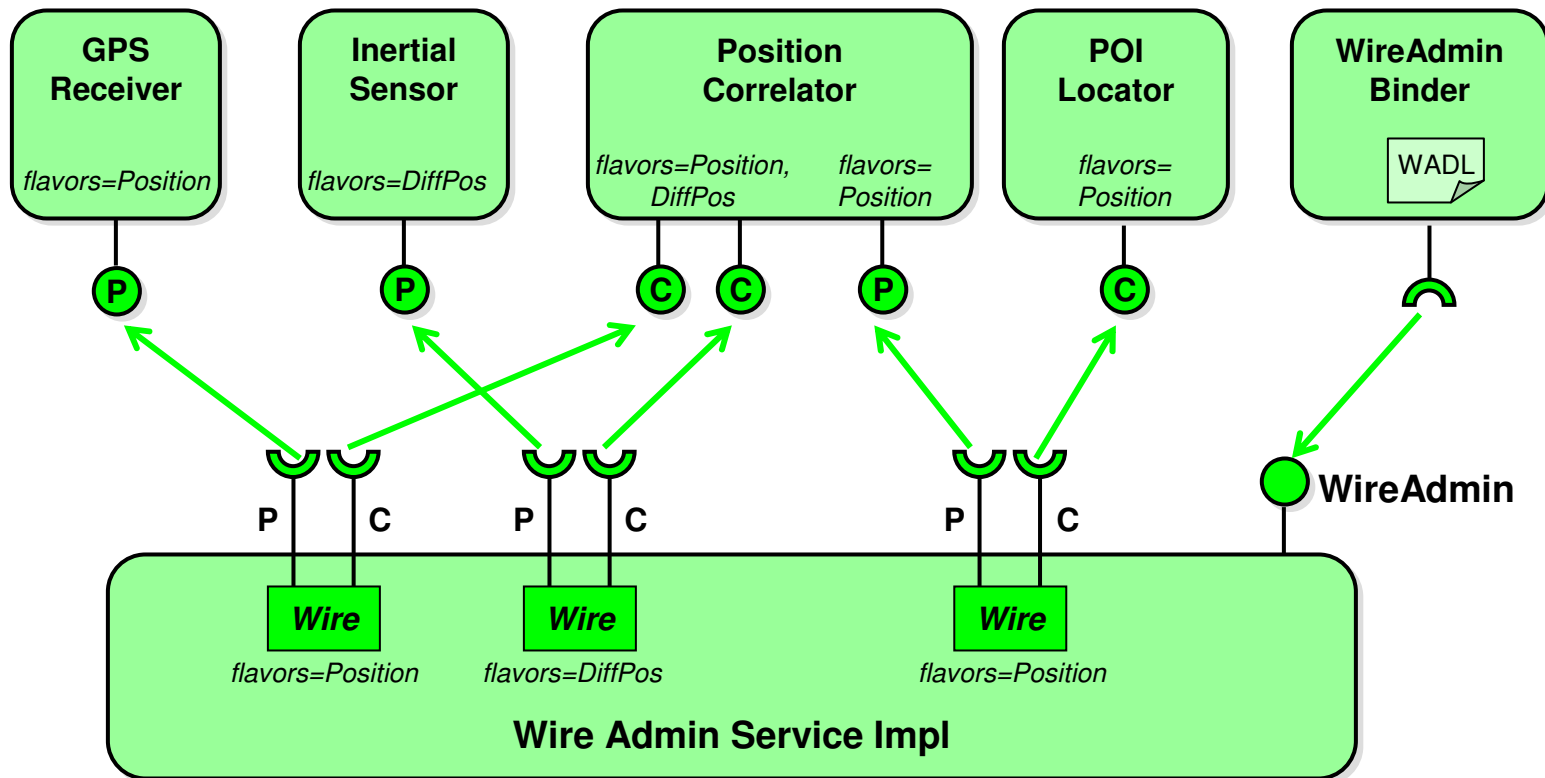


# Example of an application wired by WireAdminBinder



# Another example

- GPS-based navigation assistant



# Conclusion

---

- Ease the design of dynamic SBA with large number of sensors
  - Sensors can be added and removed at runtime
  - Sensors can be shared among several applications
  - Wire life-cycle control automation
  
- WireAdminBinder implementation for the OSGi WireAdmin
  - WAB : part of the Apache Felix project
    - <http://svn.apache.org/felix/sandbox/donsez/wab>
  - Validated in the PISE project (funded by the french DoI)
    - Domain : Energy distribution in factory plants



# Questions & Answers

---

