



# Modelling Things for IoT

Udo Hafermann  
Software AG

# The Internet of Things is Distributed and Heterogeneous

- We have "things" i.e. components of different kinds...
- ...distributed across many nodes of different kinds
- Don't want to specifically code for each kind of component or node
- Want to do it using modelling!

# HEADS: Heterogeneous And Distributed Services

- HEADS is an EU project
  - Partially funded by the European Commission under the Seventh (FP7 - 2007-2013) Framework Programme for Research and Technological Development
- Started October 2013, ends March 2017
- Industry and academic partners
- More details on GitHub and project website
  - <http://heads-project.eu/>
  - <https://github.com/HEADS-project>

# HEADS In a Nutshell

- Specifying behaviour
  - Through the HEADS modelling language
- Specifying configuration
  - Through the HEADS modelling language
- Transformation framework for generating code
  - Editor for live model
- Eclipse IDE support
  - Standalone or installed into Luna or later

# Specifying Behaviour (1)

- Code the component behaviour in the HEADS modelling language
  - Properties + logic + ports + messages
    - State machines
    - Event processing
    - (Native code if you must)

```
thing ArduinoSketch includes ArduinoApplication
{
    readonly property LED : UInt8 = 8

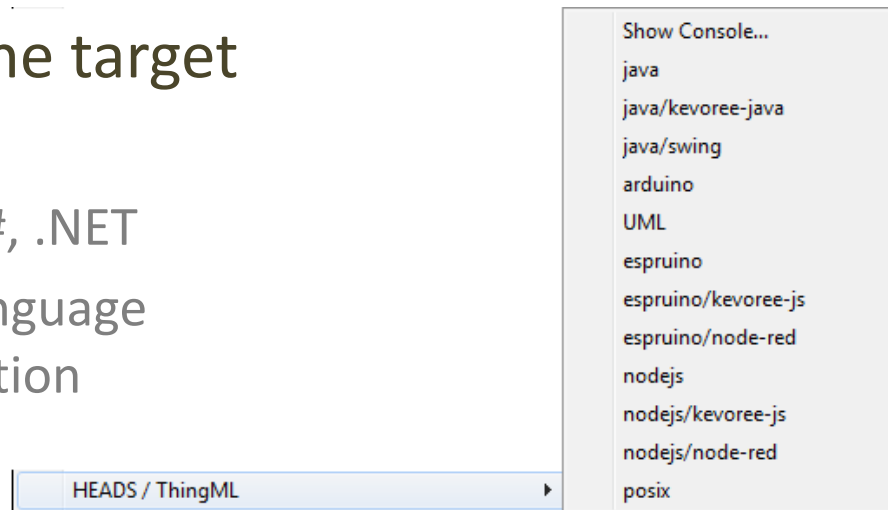
    statechart ArduinoTimerStateChart init OFF {
        on entry arduino!setOutput(LED) // init

        state OFF {
            on entry arduino!setDigitalHigh(LED)
            transition -> ON event arduino?100ms_interrupt
            //handeling of the arduino 100ms interrupt
        }
        state ON {
            on entry arduino!setDigitalLow(LED)
            transition -> OFF event arduino?100ms_interrupt
            //handeling of the arduino 100ms interrupt
        }
    }
}
```

# Specifying Behaviour (2)

- Using the editing support in Eclipse IDE
- Configuration
  - Instances and connectors
- Generate code for the target language
  - Java, JavaScript, C, C#, .NET
  - Support your next language using the transformation framework

```
import "../../lib/_Arduino.thingml"  
  
configuration ArduinoSketchCfg {  
  instance arduinoScheduler: ArduinoScheduler  
  instance application: ArduinoSketch  
  connector application.arduino => arduinoScheduler.arduino  
}
```



# Specifying Runtime (1)

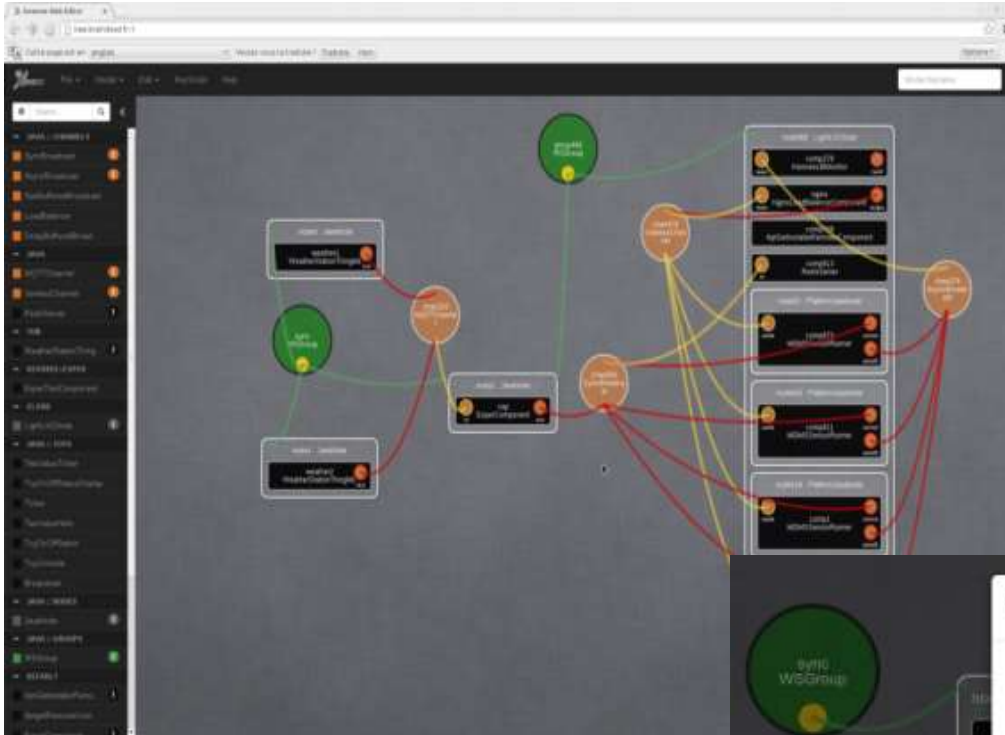
- Configuration script
  - Nodes, channels, groups
  - Generate code for deployment platforms
- Live runtime model editor

```
// create the two platform nodes
add javaNode : JavaNode
add jsNode : JavascriptNode

// provide a network way to connect to those nodes
network javaNode.ip.lo 127.0.0.1
network jsNode.ip.lo 127.0.0.1

// create a group to share the model between the nodes
add group : WSGroup
// set the master node for the group to be the jsNode
set group.master = 'jsNode'
// attach the nodes to the group
attach javaNode, jsNode group
```

# Specifying Runtime (2)



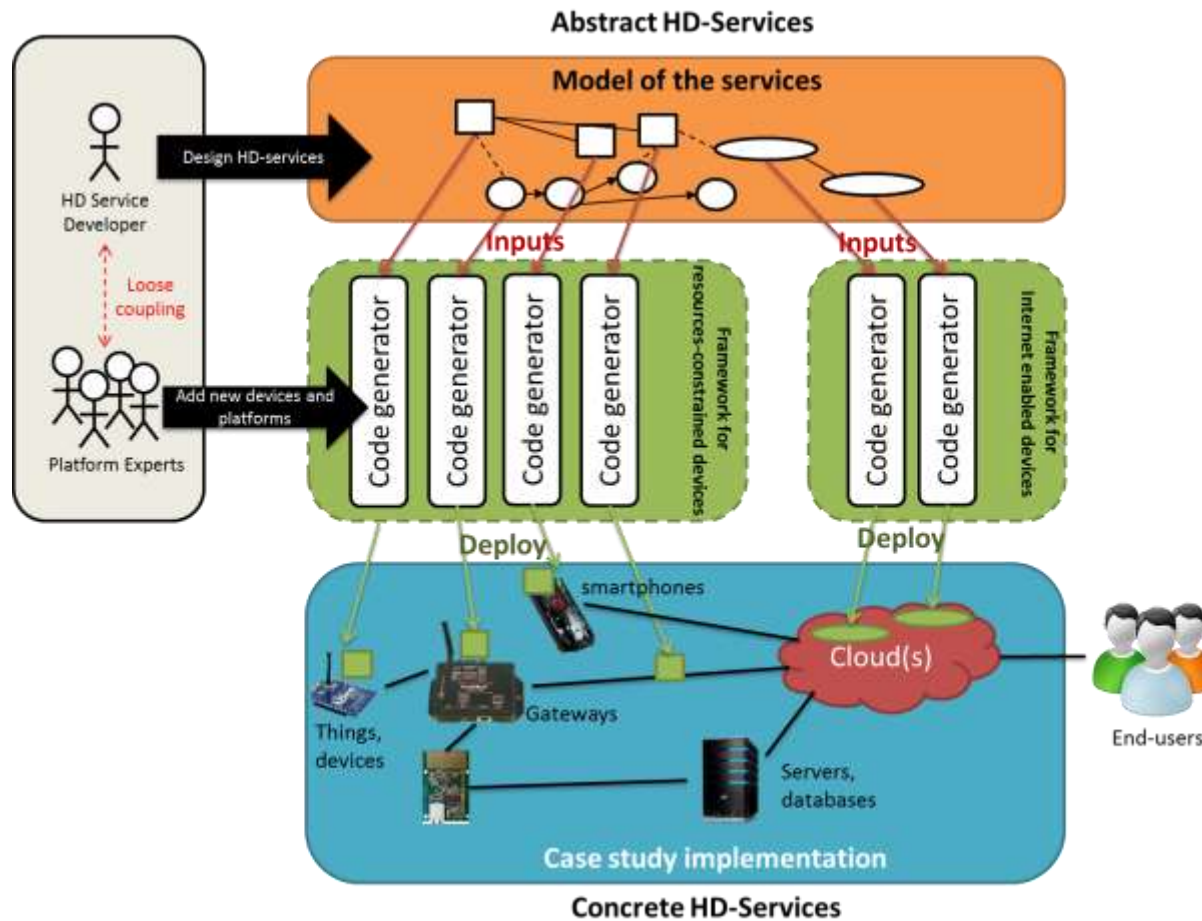
Component properties: myComp

Version	0.1.0
Name	myComp
Instance state	<input checked="" type="radio"/> Started <input type="radio"/> Stopped
greetMessage	Hello, world!

model@runtime editor



# Putting It All Together



# References

- <http://heads-project.eu/>
- <https://github.com/HEADS-project>
- <http://thingml.org/>
- <http://kevoree.org/>

# Thank You

## Questions?

# Consortium

