

# Device Abstraction Model Enables Multi-Protocol, Managed EVSEs



Infrastructure Working Council (IWC) - June 2019

Al Choperena

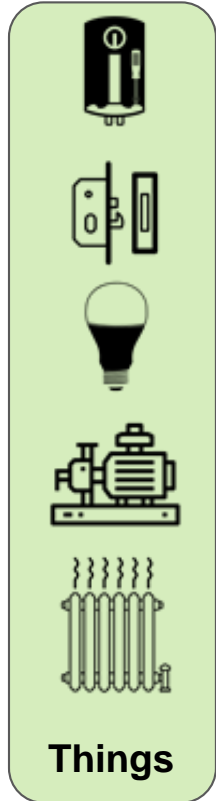
SMARTENIT

# About Smartenit

- Holistic End-to-End IoT Solutions Provider
  - Hardware and systems expertise to IoT enable **things**
  - Secure **IoT Platform** that connects disparate things
  - Sharp focus on **apps and solutions** with rapid deployment
- Rich background and track record in home automation and IoT with over 30 device SKUs and projects, delivered to small and large companies

	 	 		 	 <small>CALIFORNIA PLUG LOAD RESEARCH CENTER UNIVERSITY OF CALIFORNIA - BERKELEY</small> 
Gateway Platform	Energy Mgmt.	Energy Mgmt.	EV Charging	Technology	Academic

# We Excel At..



## IoT Protocol

Cellular  
WiFi  
ZigBee  
Thread  
BLE  
LoRa  
SigFox



**Consistent Device Model**  
**Device Management**  
Security  
Data Collection/Aggregation  
Data Analysis  
Rules Engine (Automation)

## Presentation & Exposure

Mobile App  
Web App  
Desktop App  
Cloud-Cloud Integrations  
Alexa / Google Assistant  
Utility Connectors  
Diag./Inst.Tools



# IoT Device Connectivity

# Device Model Demystified

- Simplistically, an object can be modelled and handled with two items:
  - Object **Attributes / Properties**, characteristics or values that distinguish it
  - Object **Methods / Commands** that change those characteristics or values
- Examples:

## Weather Station



### Attributes:

- Wind Speed
- Wind Direction
- Temperature
- Humidity

### Methods:

- Read Attributes
- Stop Reporting
- Start Reporting
- Read battery state

## Water Pump



### Attributes:

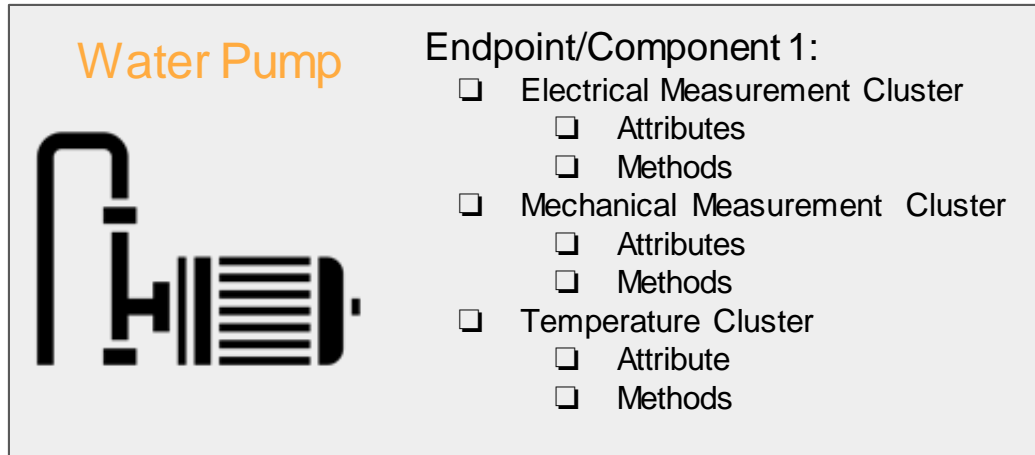
- Speed
- AC Current
- On/Off State
- Power Factor

### Methods:

- Set speed
- Turn ON
- Turn OFF
- Read AC Current

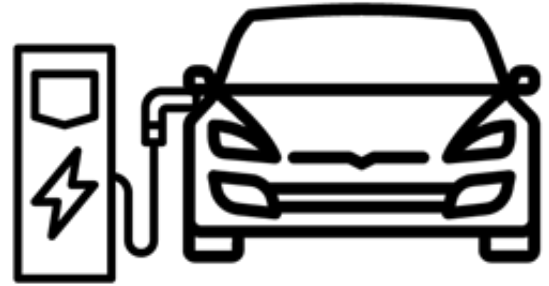
## Device Model Demystified (2)

- For convenience, Attributes and Methods are further grouped within devices:
  - **Clusters / Processors** group attributes and methods
  - **Endpoints / Components** group clusters



# Why Abstract The Managed EVSE?

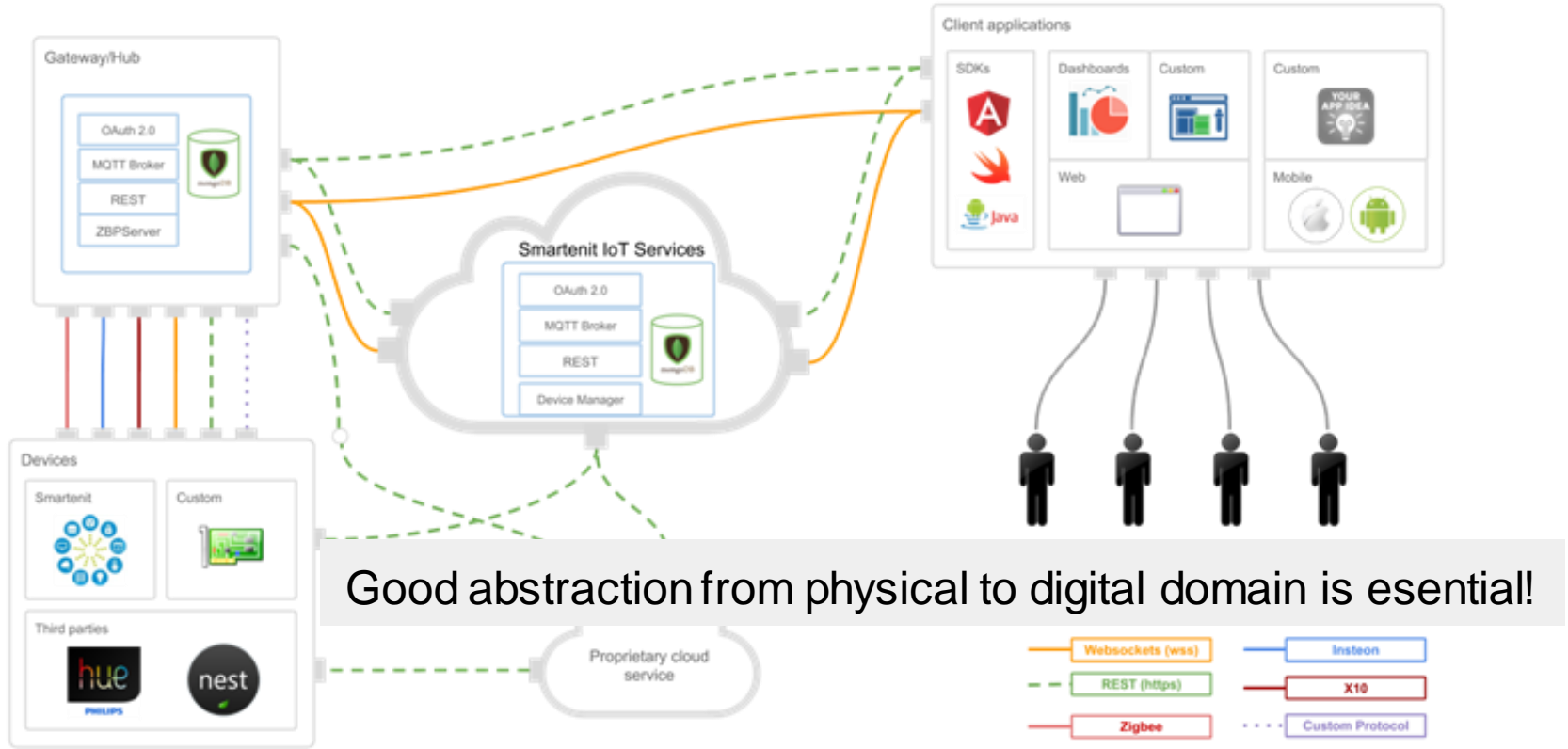
- A **connected** EVSE is not necessarily a **managed** EVSE
- A given communication protocol (WiFi, Bluetooth, Zigbee, Thread, Cellular) may only define the physical communications layer and not the command/control messages
- A common device view is a prerequisite to standardization and interoperability



## EVSE Parts:

- Contactor
- Electrical meter
- Controller
- Safety circuits
- Pilot
- Communication radio(s)

# IoT Framework and Device Models



Good abstraction from physical to digital domain is essential!

# The EVSE Management Cluster

\* Current effort at Zigbee SE and 3.0 technical groups

## Attributes

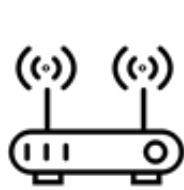
- EvseStatus
- EvseType
- EvseFault
- EnableTime
- MinChargeCurrent
- MaxChargeCurrent
- ChargingVoltage
- ChargingCurrent
- ChargingPower
- LineFrequency
- ChargingSessionID
- ChargingSessionDuration
- ChargingSessionEnergy

## Methods

- EnableEVSE
- DisableEVSE
- StartDiagnostics
- StopDiagnostics
- ReadEvent
- ReadEventSession
- ClearAllEvents
- ReadAttributes
- WriteAttributes
- NotifyEvent
- ReadEventResponse
- ReadAttributeResponse
- WriteAttributeResponse



# A Managed EVSE Architecture



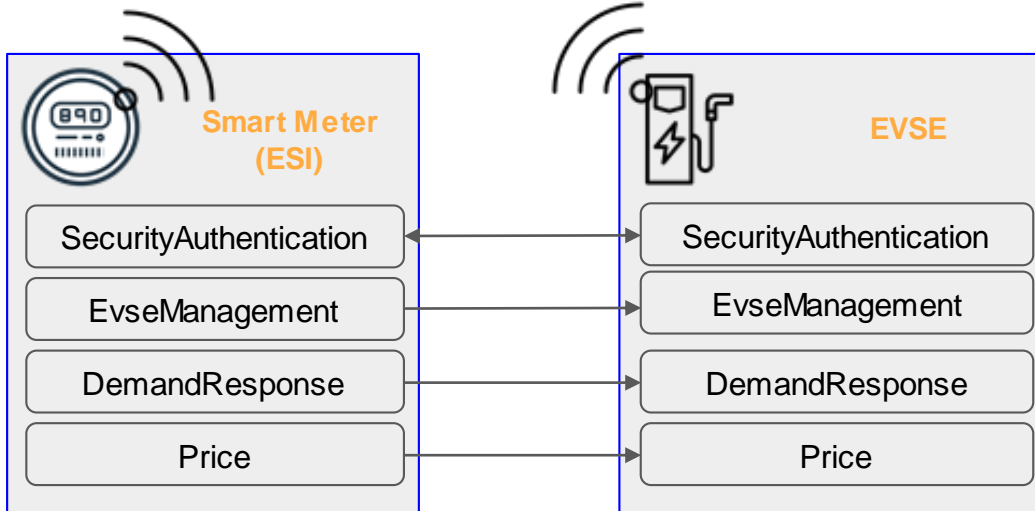
Gateway



Cloud/Backend



Apps and Dashboards



Several possible variations (Zigbee shown)

- WiFi EVSE through router
- Thread EVSE through edge router
- Cellular EVSE direct
- LoRaWan through gateway and network controller

# Benefit: A Protocol Independent API

```
https://api.smartnit.io/v2/devices/{{device_id}}/comps/1/procs/EVSEServer/methods/StartCharge
```

```
Response: {  
  "message": "Device method executed",  
  "data": {}  
}
```

```
https://api.smartnit.io/v2/devices/{{device_id}}/comps/1/procs/EVSEServer/attributes/ChargerSessionSummation
```

```
Response: {  
  "message": "Attribute response",  
  "apiCache": false,  
  "data": {  
    "timestamp": 1559862633,  
    "value": 1670 //kWh  
  }  
}
```



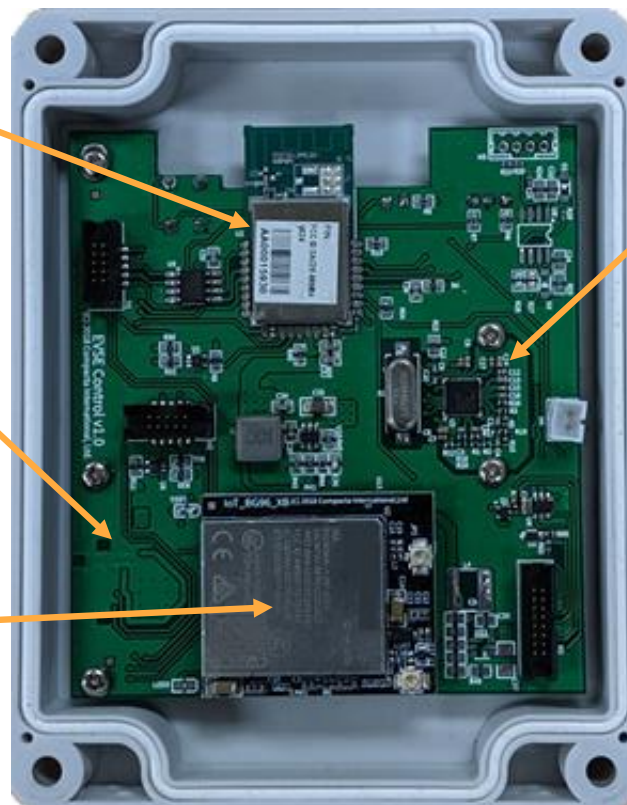
# Multi-Protocol Managed EVSE Implementation

Zigbee/Thread  
SOIC

Energy  
metering and  
ground-fault  
detect

WiFi and  
Bluetooth  
radios  
(on opposite side)

Optional Radio  
LoRa, Cellular,  
AMI specific..  
(Shown w/ LTE-M  
/NBloT)



# Typical Interactions For End User

- Set electric rates from utility bill or utility API/Portal
- Commission EVSE and Set EV profile
- Set automation rules
- Manage device (start charge, stop charge, see consumption, change schedules)
- See/monitor historical consumption and cost



# Managed EVSE Benefits

## To end users:

- Multiple EVs (load management with limited electric service capacity)
- Utility incentives
- Optimum time of use / lowest operating cost
- Alerts when not plugged in

## To utility/service providers:

- Load Management
- Demand response and Price/TOU shedding
- Improved asset utilization
- Choice of connectivity (Zigbee, Thread, WiFi, Cellular, existing AMI net, etc.)

# Conclusions And Parting Thoughts

- The impact of the EV on the grid is widely recognized. EVs also speed up the shift from a fossils economy to a renewables one
- A **managed** EVSE is really another tool for EV and grid management
- There are many IoT / connectivity protocols in place. Choice should be based on needs - no dominant standard should be expected
- A common device view is a prerequisite to standardization and interoperability at the application layer
- A managed EVSE offers multiple benefits to utilities, service providers and end users alike

# About The Author

Al is an electrical engineer with expertise in control systems and telecommunications. Following a successful career in medical devices, he turned his energy as an entrepreneur to extend the promise of the “Internet of Things”.

He founded Smartenit with the idea of eliminating “islands of automation” with the premise that everything should work harmoniously together, for the benefit of mankind.

1 Technology Drive, D405

Irvine, CA 92618

eMail: [alchoper@smartenit.com](mailto:alchoper@smartenit.com)

Web: <https://smartenit.com>

LinkedIn: <https://linkedin.com/company/smartenit>