



Universal Development Kit for Creating and Deploying Smart Home/Building Applications

An example based on Smart Energy Profile 2.0 (SEP 2.0) deployment use case

Oleg Logvinov – STMicroelectronics
(oleg.logvinov@st.com)

Frédéric Vaute – Oracle
(frederic.vaute@oracle.com)



ST: Where you find us

2



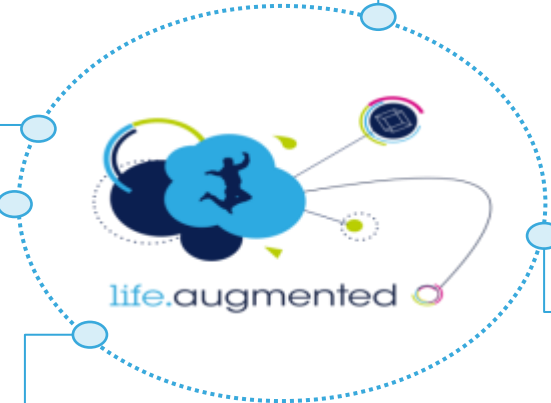
Our MEMS & Sensors
are augmenting
the consumer experience



Our digital consumer products
are powering the augmented
digital lifestyle



Our automotive products
are making driving safer,
greener and more
entertaining

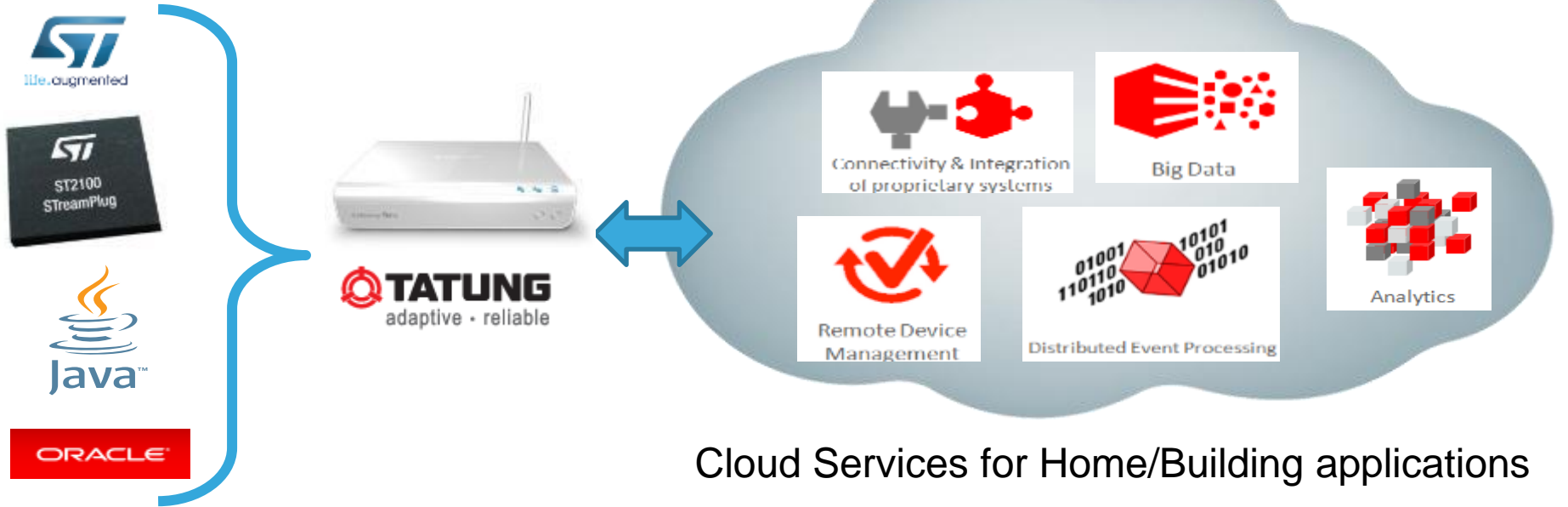


Our Microcontrollers
are everywhere
making everything smarter
and more secure



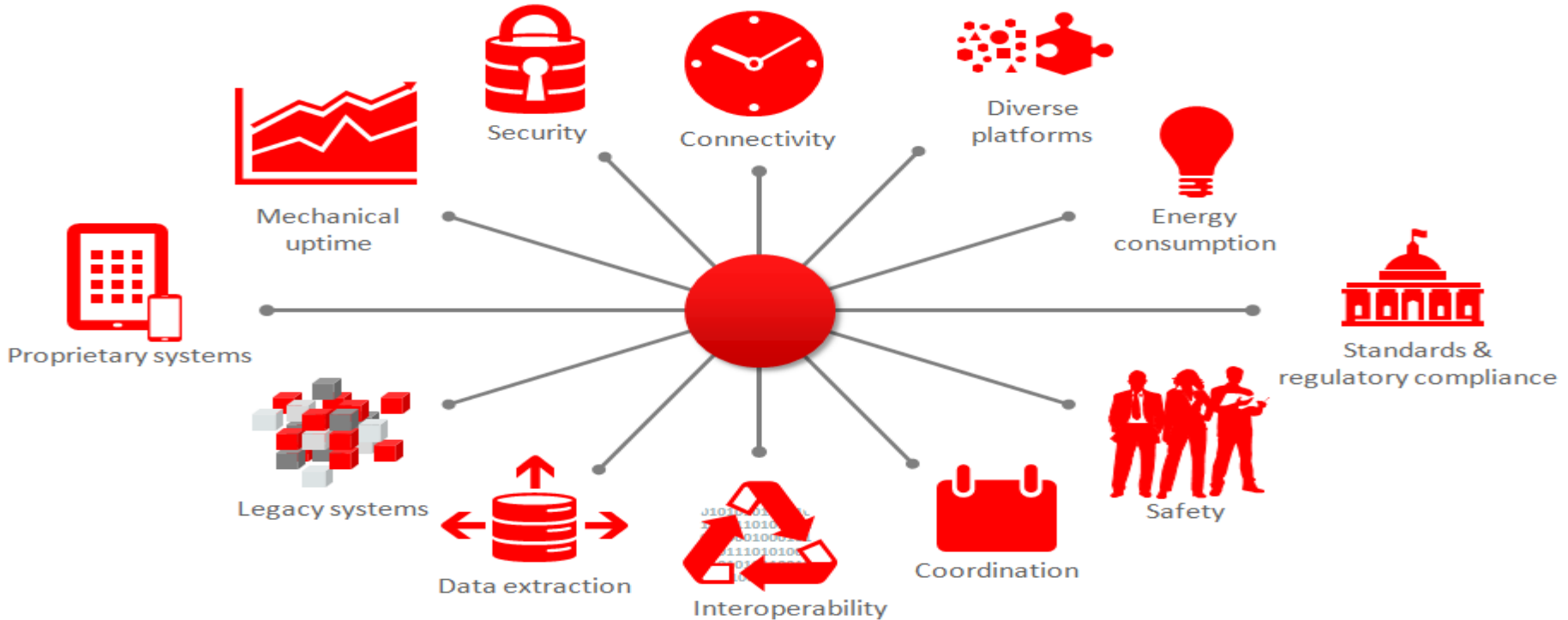
Our smart power products
are making more of our energy resources

Universal Dev. Kit for smart Home/Building



Cloud Services for Home/Building applications

Challenges for The Internet of Things



Java embedded platform overview

5



ENABLING smart devices in Industrial Internet



- HW/OS independence
- Local DB, web-enabled, event aware
- Optimized for embedded
- Rich graphics support
- Communications ready

Java Embedded Platform



- Standards-based + modern language
- Strong tool chain
- Reusable code
- Access to native system resources
- Large worldwide developer pool

Java Language & Developers

APPLICATIONS



- Write Once, Run Anywhere
- High performance
- Dynamically optimized
- Consistent runtime environments

Java Applications

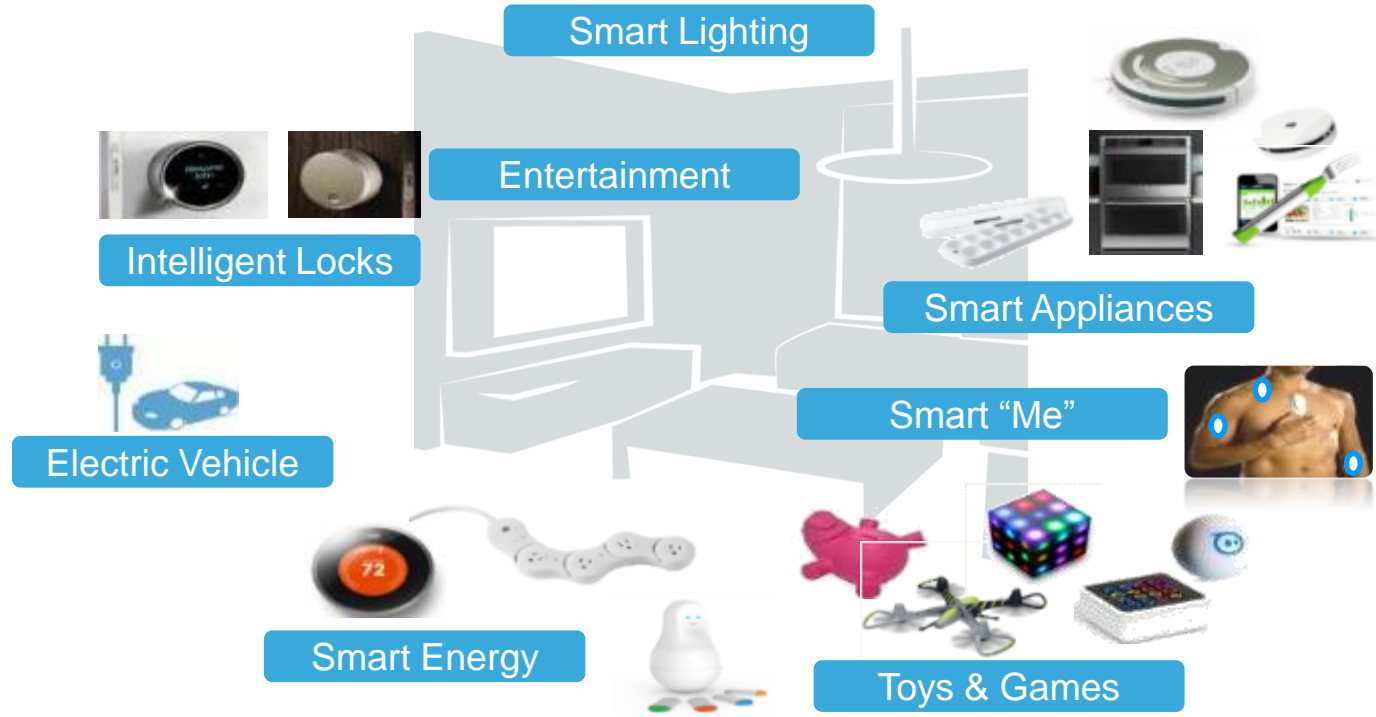


- Proven security model
- Strong cryptographic support
- Sandbox security model

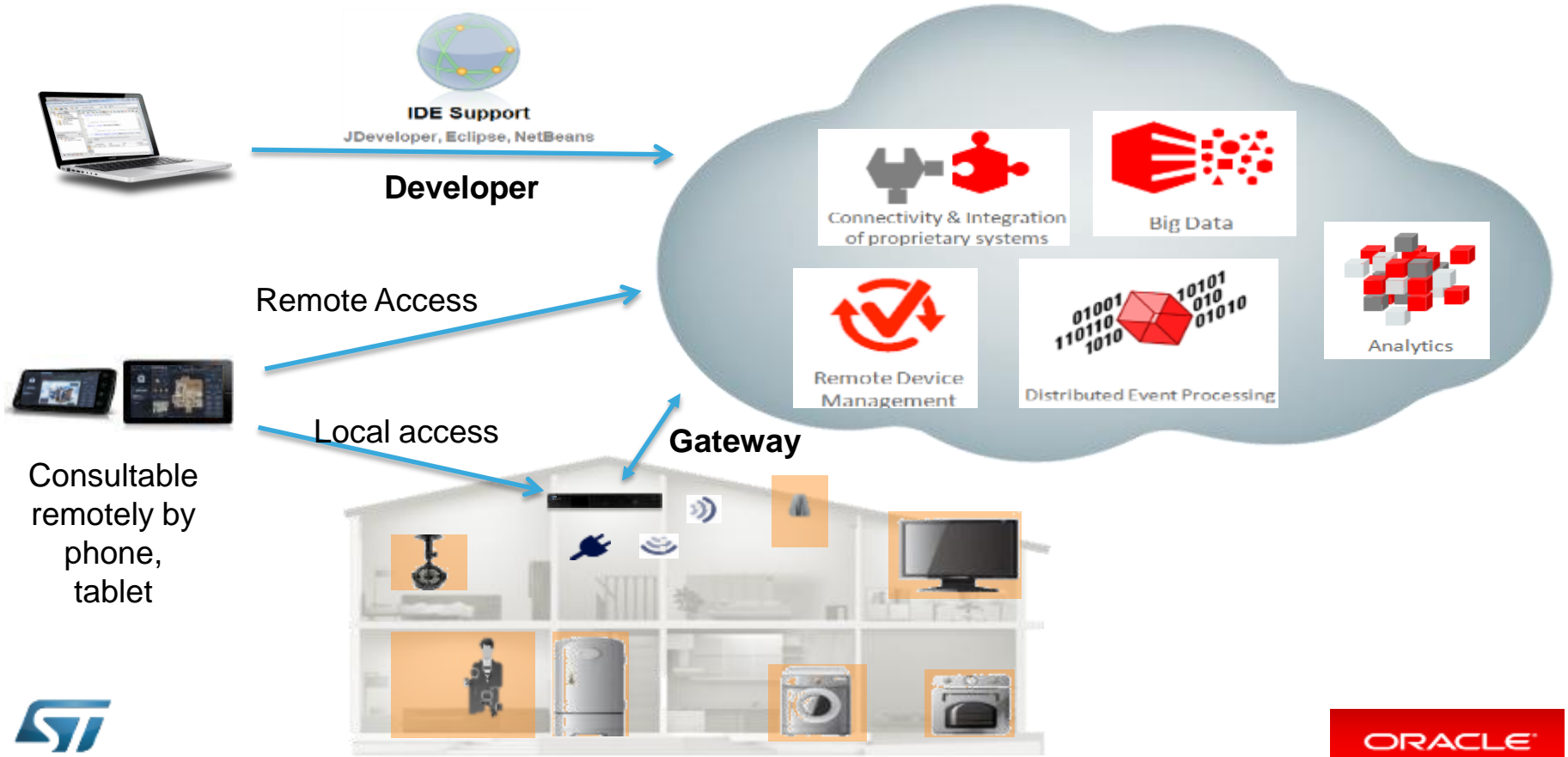
Java Security

Embracing the Smart Home

- Sensors, intelligence and connectivity being added to many devices in the home
- Innovative nature of the products allows new companies to challenge established leaders
- ST present with many of the leaders in the first wave of augmented things in the home



Smart Home End2End Architecture



Java SE 8: a platform for embedded

Themes

- Bringing Java 8 language to the embedded
- Dedicated and optimized embedded software runtime
- Bring enterprise features to the embedded

Key Features

- Modern, flexible, standards-based software runtime
- Value-add new & enhanced features for embedded
- Improved configurability and optimized footprint
 - Target devices as low as at **16/32* MB RAM, 12/30* MB Flash/ROM**

Target Markets

- Mid to to high-embedded covering wide range of use cases/markets
- Intelligent smart embedded and gateways and embedded HMI

Smart Home GW Platform

9

GatewayOne by Tatumg

ARM 926EJ-S@333MHz

- 360 DMIPS; 200 when running HPAV
- Linux
- Oracle Java SE Embedded 8 CP1
- OSGI
- HomePlug AV/GP
- Wi-Fi 802.11n
- BT Smart Ready
- Ethernet
- USB 2.0
- Optional Zwave and ZigBee

STreamPlug: HomePlug® Smart SoC



ORACLE

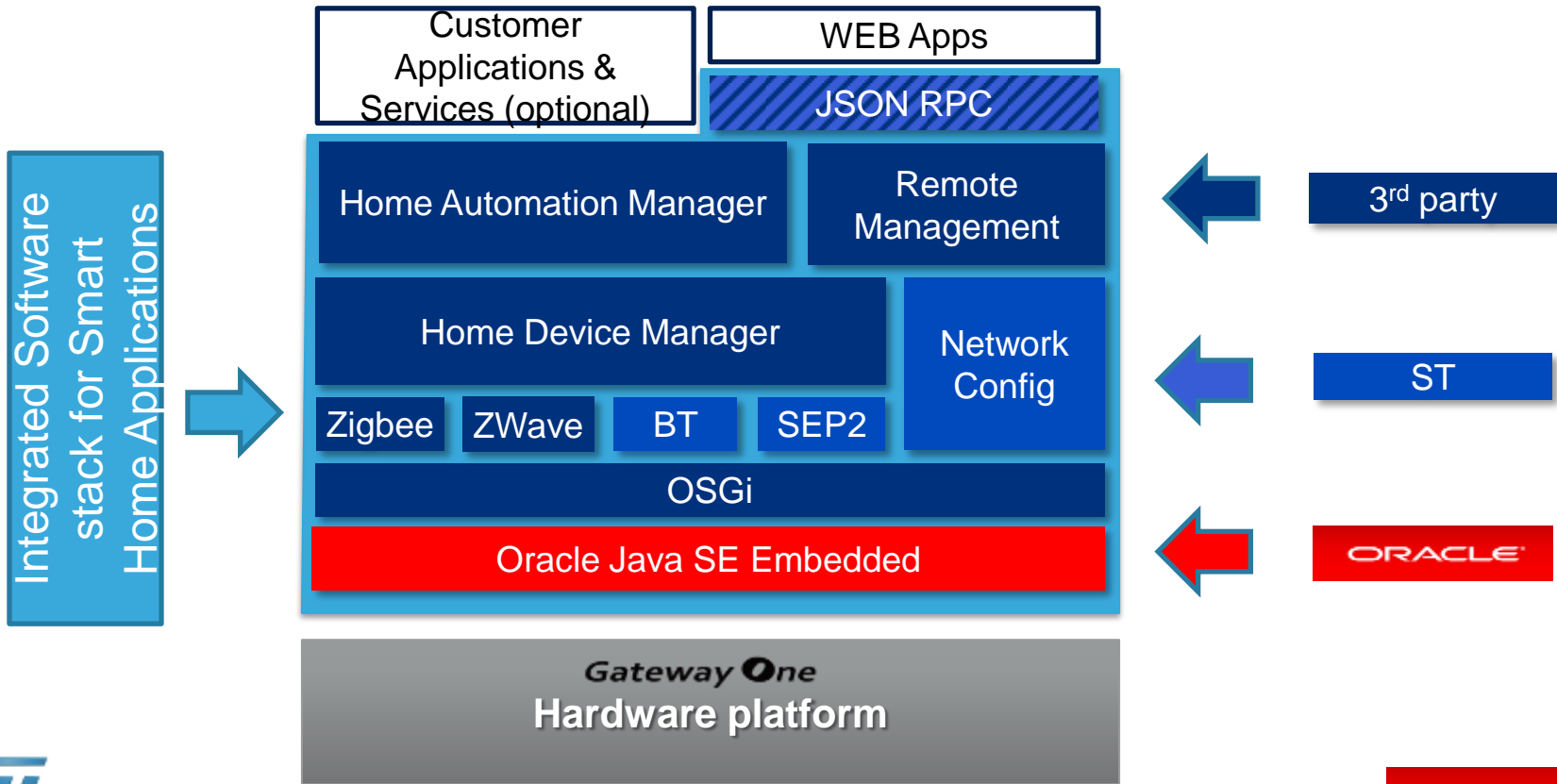


Press release: <http://www.st.com/web/en/press/p3478>

Product information: <http://www.tatumg.com/Site/Detail/10664>

ORACLE

Smart Home Gateway Stack



Application layer interaction

Service POV



CPE

Remote Gateway Management



Secure channel



Cloud

JSON-RPC/Websockets

Developer POV

1

GUI extensions



- JAVAscript commands
- Graphical Interface

2

JSON RPC bundle



- Browser callable methods
- Allows exporting data to cloud

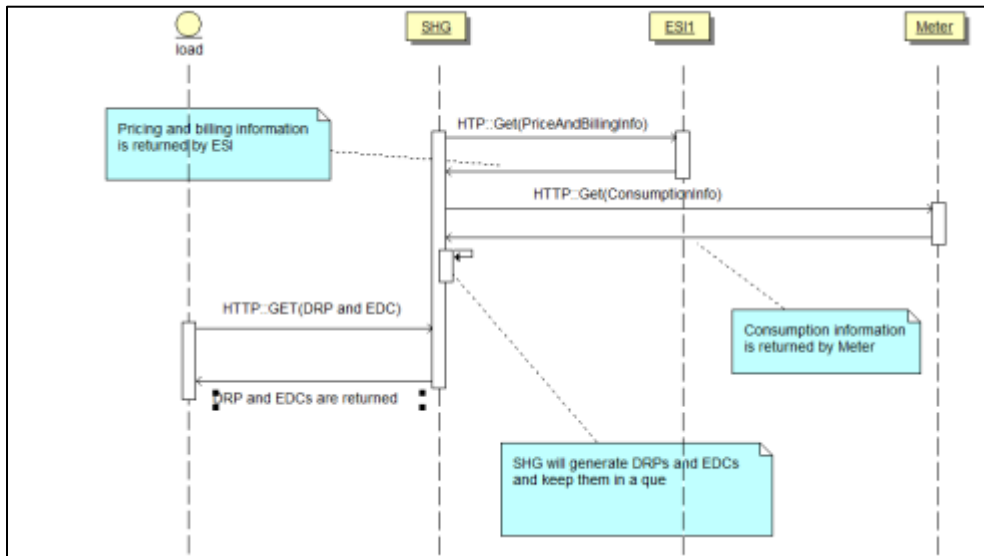
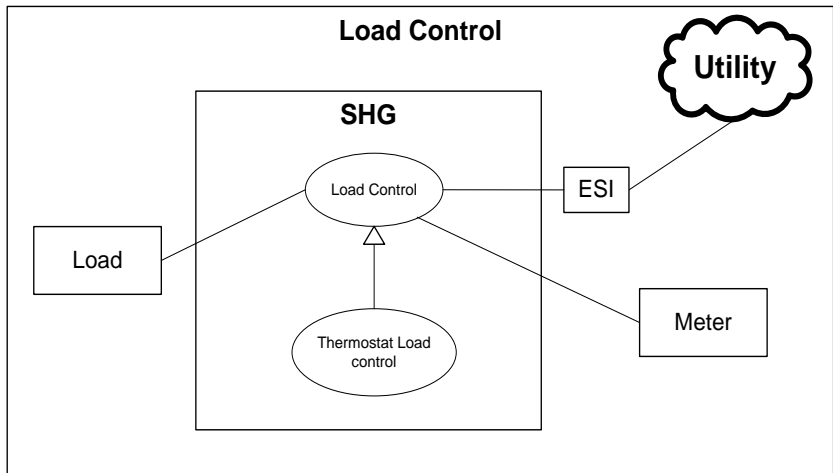
3

Service bundle

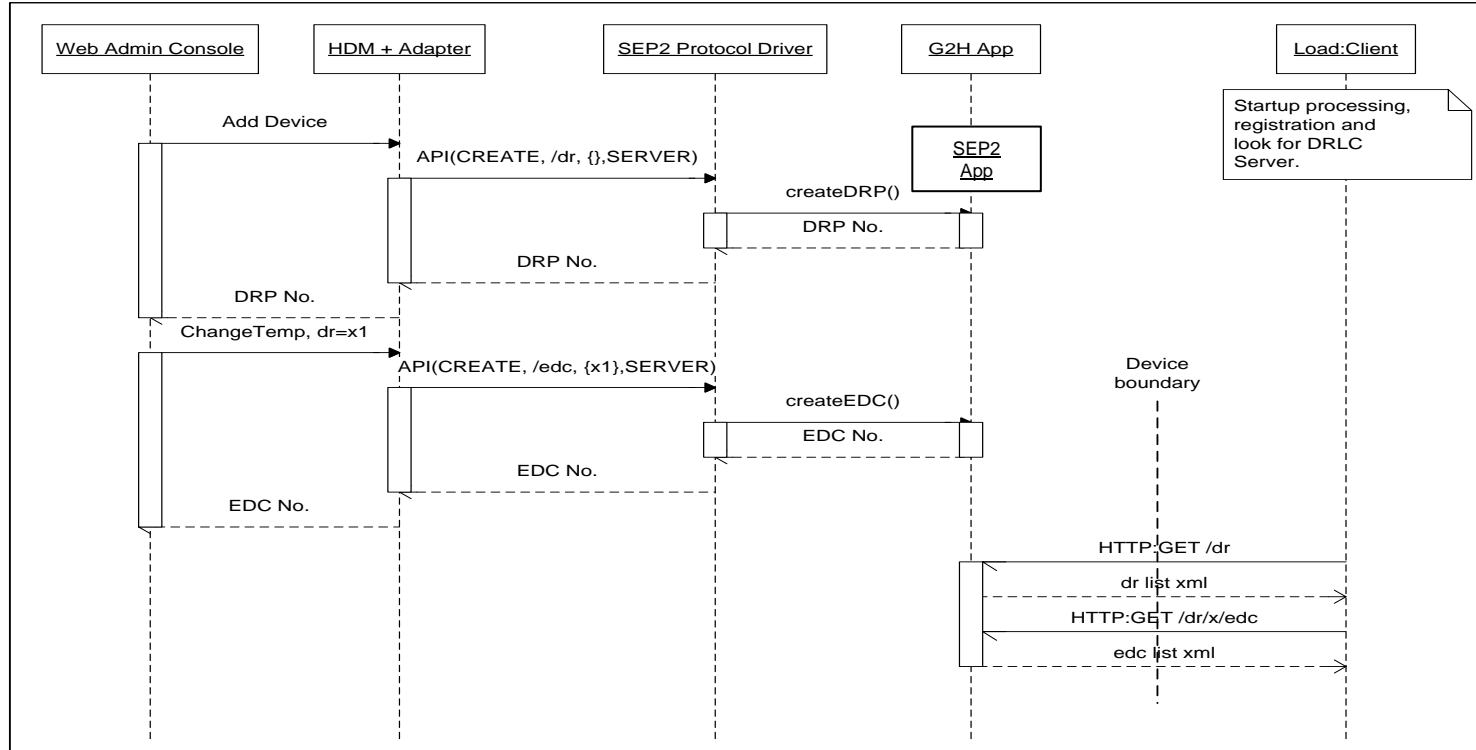


- JAVA code implements functionality
- Interface HW/SW on platform

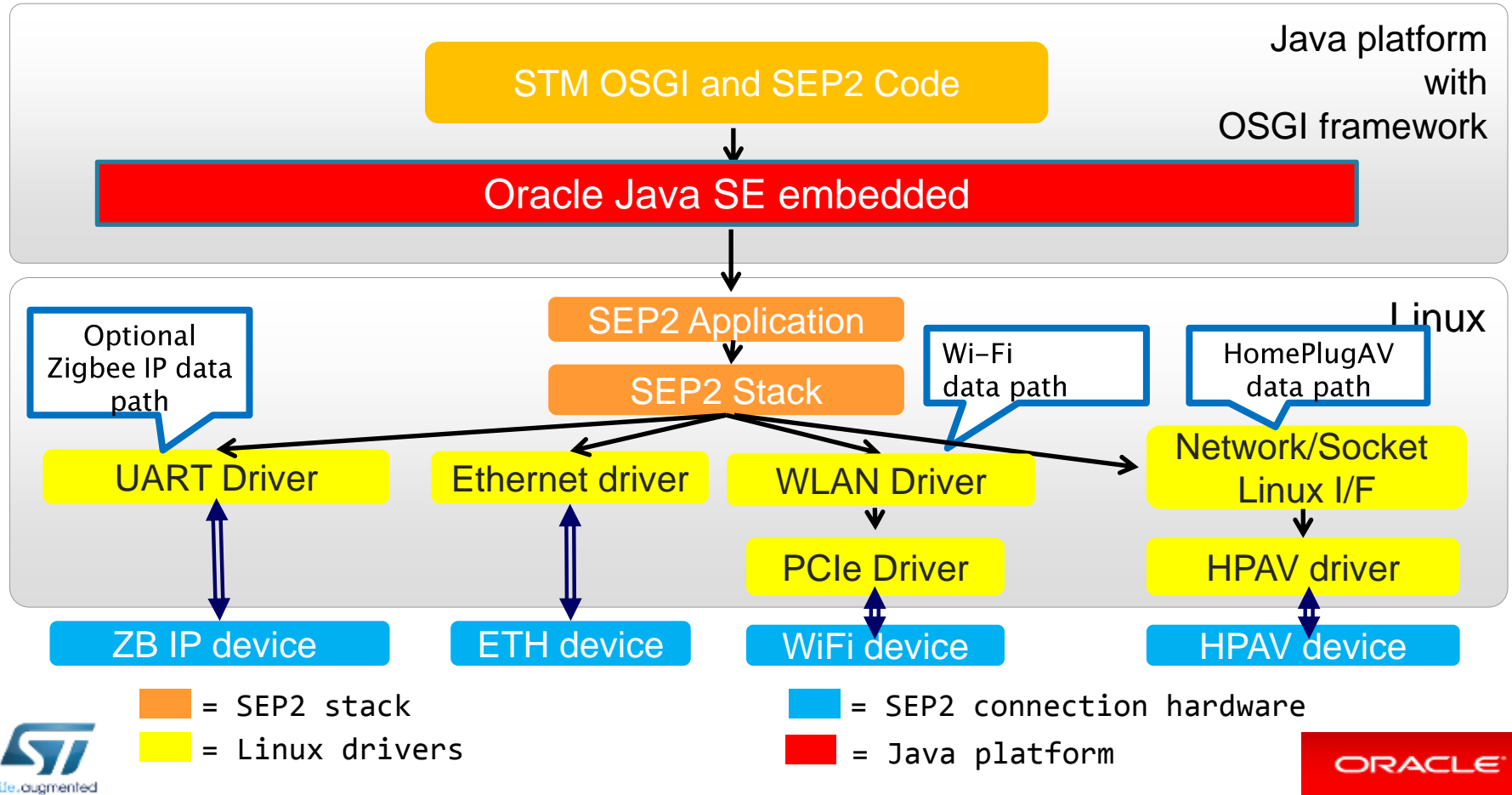
Example of a Thermostat controlling the temperature (1)



Example of a Thermostat controlling the temperature (2)



SEP2.0 SW ARCHITECTURE



SEP2 demo description



SHG SEP2 Server Supported Devices

Thermostat

Simple Load

Interior Light

Smart Home Device Table

- SEP2 Server

- GUI Server side set controlled devices
- Uses JSON-RPC commands to interact with HDM abstraction layer
- Register new resources and control them

- SEP2 Client Devices

- Emulates the presence of SEP2 appliances
- Usually it is run on a PC with Tomcat
- Emulated devices are controlled by the SEP2 Server



SEP2 Client Devices Current Monitored Devices

Thermostat Utility

Load Pool Pump

Interior Light



SEP2 resources in OSGI (Prosyst) console

Web Admin Console Home Devices



total devices: 13 online devices: 13 offline devices: 0

Name	UID	Type	Status	Error	Protocol Adapter	Hardware Vendor	Hardware Version	Actions
SmartHomeGateway_ZWave_USB	hdm:ZWave:23191754/1/020201/202198	Z-Wave Local Controller	Online	---	ZWave	---	Z-Wave 2.7B	
+ TELEGESIS_DONGLE	hdm:ZigBee:0006f0002804722	Controller	Online	---	ZigBee	---	---	
+ ZMOVE	hdm:ZigBee:00124b000124934b	End Device	Online	---	ZigBee	CLECODE	0	
+ ZW_MULTIT_SENSOR	hdm:ZWave:23191754/44/042001/202031707200848586	2	Online	---	ZWave	---	1.34	
Zone1 Thermo	SEP2Thermo1	---	Online	---	SEP2 Adapter	STMicroelectronics	1.0.0	
+ hdm:ZigBee:0006f0002319aa	hdm:ZigBee:0006f0002319aa	Router	Online	---	ZigBee	4-mob	0	

Linked Devices: 1..8



SHG SEP2 Server Supported Devices

Thermostat

Thermostat ID [SEP2Thermo1]

Thermostat Name [Zone1Thermo]

[Get New Thermostat](#)

Thermostat ID [2..4]

Clear Request [25]

[Get Thermostat](#)

Smart Home Device Table

Device ID	Device Name
0	SEP2Thermo1

Simple Load

Simple Load ID [2..4]

Simple Load Name [2..4]

[Add Simple Load](#)

Simple Load ID [2..4]

Setup Cycle [25]

[Get Simple Load](#)

Inverter Light

Inverter Light ID [2..4]

Inverter Light Name [2..4]

[Add Inverter Light](#)

Inverter Light ID [2..4]

Duty Cycle [25]

[Get Inverter Light](#)

- Registered resources are seen as devices in the Prosyst console and listed as SEP2 Adapters

JSON RPC Methods to control/access SEP2 devices

- SEP2 devices in the network could be controlled or accessed through HTTP/IP protocol from any other device using JSON-RPC methods described in the Prosyst framework
- On top of Prosyst JSON-RPC methods, new methods are defined to access SEP2 devices, described in the following:
 - Sep2Json/addSEP2Device
 - This JSON RPC can be used to add new SEP2 device.
 - Sep2Json/removeSEP2Device
 - This JSON RPC can be used to remove a SEP2 device.
 - Sep2Json/getDeviceCount
 - This JSON RPC can be used to get the number of SEP2 devices connected to the gateway.
- Some standard JSON-RPC methods can be used to do things like modify attributes/values, access device objects:
 - HDAccess/getDeviceClassObjects
 - HDAccess/SetDCOProperty
 - HDAccess/getHomeDevices

Application layer interaction

Service POV



CPE

Remote Gateway Management



Secure channel



Cloud

JSON-RPC/Websockets

Developer POV

1

GUI extensions

GUI

- JAVAscript commands
- Graphical Interface

2

JSON RPC bundle

RPC extensions

- Browser callable methods
- Allows exporting data to cloud

3

Service bundle

Service bundle

- JAVA code implements functionality
- Interface HW/SW on platform

HTML/JAVA page JSON/RPC 1/2

20

- Initial scanning of the available displayed resources

```
$(document).ready( function() {
  var tableDevices = $("#tableDevices tbody");
  tableDevices.empty();
  var x = sendRequest("HDAccess/getHomeDevices", null);
  if (isError(x) || !x.result || !x.result.length) {
    tableDevices.html("<tr><td>No devices</td><td></td><td></td><td></td></tr>");
  } else {
    var devices = x.result;
    var htmlToAppend = [];
    //TEST
    document.write(obj2str(devices[0], true));
    for (var i in devices) {
      htmlToAppend[0] = '<tr><td>' + (devices[i].Name && devices[i].Name.length ? devices[i].Name : devices[i].UID) + '</td>';
      htmlToAppend[1] = '<td>' + devices[i].UID + '</td>';
      htmlToAppend[2] = '<td><div id="' + devices[i].UID + '">' + getStatusText(devices[i].Status) + '</div></td>';
      htmlToAppend[3] = '<td>' + getDCTest(devices[i].UID, devices[i].DeviceClasses) + '</td>';
      htmlToAppend[4] = '<td>' + getDCOTest(devices[i].UID, devices[i].DeviceClasses) + '</td>';
      htmlToAppend[5] = '</tr>';
      tableDevices.append(htmlToAppend.join(''));
      htmlToAppend = [];
    }
  }
}
```

- The JSON/RPC function call...

```
var requestID = 1;

function sendRequest(remoteMethod, params) {
  var id = requestID++;
  var response = null;
  jQuery.ajax({
    async      : false,
    contentType: 'application/json',
    type       : 'post',
    processData: false,
    dataType   : 'json',
    url        : '/remote/json-rpc',
    cache      : false,
    data       : '{"jsonrpc": "2.0", "method": "' + remoteMethod + '", "params": ' + params + ', "id": "' + id + '"}',
    success    : function(data) {
      response = data;
    },
    error: function(req, stat, err){
      alert('Error\nreq: ' + req + '\n stat: ' + stat + '\n err: ' + err);
    }
  });
  return response;
}
```

- Insertion of a new device in the setup...

```
84 <div class="row">
85   <div class="col-xs-6 col-md-4">
86     <a href="#" class="thumbnail">
87       
88       <h4 align="center" > Thermostat </h4>
89       <br>
90       <p>Thermostat ID <input class="gray" type="text" id="thermostatIdentifier" value="th-id" /> </p>
91       <p>Thermostat Name <input class="gray" type="text" align="right" id="thermostatDescription" value="th-name" /> </p>
92       <input type="button" value="Add New Thermostat" onclick="addThermostat()" />
93       <br>
94       <br>
95       <p>Thermostat ID <input class="gray" type="text" id="uid_thermostat" value="th-id" /> </p>
96       <p>Cool-Setpoint <input class="gray" type="text" id="coolSetPoint" value="25" /> </p>
97       <input type="button" value="Set Temperature" onclick="thermostatSetCoolSetpoint()" />
98     </div>
99   </div>
100 <div class="col-xs-6 col-md-4">
101   <a href="#" class="thumbnail">
102     
```

- ... and the related JSON/RPC request.

```
function addThermostat() {
  var id = document.getElementById("thermostatIdentifier").value;
  var description = document.getElementById("thermostatDescription").value;
  addSEP2Device("Thermostat", id, description);
}
```

```
function addSEP2Device(devType, devID, devDescription) {
  var x = sendRequest("Sep2Json/addSEP2Device", '[' + devType + ', ' + devID + ', ' + devDescription + ']' );
  isError(x);
  addRowToTable(devID, devDescription)
```

Network transactions

Filter: **ip.addr eq 192.168.2.104** Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
78	1.800377	192.168.2.1	192.168.2.104	HTTP	668	POST /remote/json-rpc HTTP/1.1 (application/json)
79	1.801158	192.168.2.104	192.168.2.1	TCP	66	http-alt > 49934 [ACK] Seq=1 Ack=601 Win=1994 Len=0 TSval=7874720 TSecr=4060143
82	1.952542	192.168.2.104	192.168.2.1	TCP	341	[TCP segment of a reassembled PDU]
83	1.952566	192.168.2.1	192.168.2.104	TCP	66	49934 > http-alt [ACK] Seq=601 Ack=276 Win=2641 Len=0 TSval=4060181 TSecr=7874735
84	1.953566	192.168.2.104	192.168.2.1	TCP	69	[TCP segment of a reassembled PDU]
85	1.953579	192.168.2.1	192.168.2.104	TCP	66	49934 > http-alt [ACK] Seq=601 Ack=279 Win=2641 Len=0 TSval=4060182 TSecr=7874735
86	1.954264	192.168.2.104	192.168.2.1	TCP	78	[TCP segment of a reassembled PDU]
87	1.954276	192.168.2.1	192.168.2.104	TCP	66	49934 > http-alt [ACK] Seq=601 Ack=291 Win=2641 Len=0 TSval=4060182 TSecr=7874735
88	1.955819	192.168.2.104	192.168.2.1	TCP	70	[TCP segment of a reassembled PDU]
89	1.955831	192.168.2.1	192.168.2.104	TCP	66	49934 > http-alt [ACK] Seq=601 Ack=295 Win=2641 Len=0 TSval=4060182 TSecr=7874735
90	1.956428	192.168.2.104	192.168.2.1	TCP	104	[TCP segment of a reassembled PDU]
91	1.956441	192.168.2.1	192.168.2.104	TCP	66	49934 > http-alt [ACK] Seq=601 Ack=333 Win=2641 Len=0 TSval=4060182 TSecr=7874735
92	1.957098	192.168.2.104	192.168.2.1	HTTP	71	HTTP/1.1 200 OK (application/json)
93	1.957116	192.168.2.1	192.168.2.104	TCP	66	49934 > http-alt [ACK] Seq=601 Ack=338 Win=2641 Len=0 TSval=4060182 TSecr=7874735
114	2.613293	192.168.2.1	192.168.2.104	HTTP	419	GET /ServerApp/DisplayData HTTP/1.1
115	2.613373	192.168.2.1	192.168.2.104	HTTP	421	GET /ServerApp/DisplayDCycle HTTP/1.1
116	2.613452	192.168.2.1	192.168.2.104	HTTP	420	GET /ServerApp/DisplayLight HTTP/1.1

```
Accept-Language: it-IT,it;q=0.8,en-US;q=0.5,en;q=0.3\r\n
Accept-Encoding: gzip, deflate\r\n
Content-Type: application/json; charset=UTF-8\r\n
X-Requested-With: XMLHttpRequest\r\n
Referer: http://192.168.2.104:8080/sepgui\r\n
Content-Length: 106\r\n
Connection: keep-alive\r\n
Pragma: no-cache\r\n
Cache-Control: no-cache\r\n
\r\n
[Full request URI: http://192.168.2.104:8080/remote/json-rpc]
JavaScript Object Notation: application/json
```

```
Object
  Member Key: "jsonrpc"
    String value: 2.0
  Member Key: "method"
    String value: Sep2.Json/addSEP2Device
  Member Key: "params"
    Array
      String value: Thermostat
      String value: th-id
      String value: th-name
  Member Key: "id"
```

192.168.2.1



192.168.2.104



1

2

Application layer interaction

Service POV



CPE

Remote Gateway Management



Secure channel

JSON-RPC/Websockets



Cloud

Developer POV

1

GUI extensions



- JAVAscript commands
- Graphical Interface

2

JSON RPC bundle



- Browser callable methods
- Allows exporting data to cloud

3

Service bundle



- JAVA code implements functionality
- Interface HW/SW on platform

- Declarations for JSON RPC call registration...

```
public Object addingService(ServiceReference reference) {
    Object service = bc.getService(reference);
    if (service instanceof HomeDeviceAdmin) {
        application = new Application((HomeDeviceAdmin)service, bc);
        Commands commands = new Commands(application);
        regCommands = bc.registerService(PluggableCommands.class.getName(), commands, null);

        application.devicesAdmin = (HomeDeviceAdmin)service;
        Hashtable prop = new Hashtable(5);
        prop.put(Constants.SERVICE_PID, "Sep2Json");
        prop.put(RemoteServiceConstants.SERVICE_EXPORTED_INTERFACES, new String[] { "" });
        prop.put(RemoteServiceConstants.CONFIG_EXPORTED_METHODS, new String[] { "Sep2Json/addSEP2Device", "Sep2Json/removeSEP2Device", "Sep2Json/getDeviceCount" });
        regCommands = bc.registerService(SepCommands.class.getName(), application, prop);

        return service;
    }
    return null;
}
```

- ...and the addSEP2Device definition

```
public void addSEP2Device(String deviceType, String id, String description) {
    System.out.println("addSEP2Device(): deviceType: " + deviceType + ", id: " + id + ", description: " + description);
    ProtocolAdapterInfo adapter = devicesAdmin.getProtocolAdapterInfo(SEP2_ADAPTER);
    if (adapter == null) {
        throw new RuntimeException("Missing adapter: " + SEP2_ADAPTER);
    }

    if(0==deviceType.compareTo("InteriorLight")) {
        System.out.println("adding InteriorLight");
        addDevice(adapter, id, description, new String[] {InteriorLight.class.getName()});
    }else if(0==deviceType.compareTo("Thermostat")) {
        System.out.println("adding Thermostat");
        addDevice(adapter, id, description, new String[] {TemperatureActuator.class.getName()});
    }else if(0==deviceType.compareTo("SimpleLoad")) {
        System.out.println("adding SimpleLoad");
        addDevice(adapter, id, description, new String[] {SimpleLoad.class.getName()});
    }else {
        System.out.println("unknown deviceType");
    }
}
```


Application layer interaction

Service POV



CPE

Remote Gateway Management



Secure channel



Cloud

JSON-RPC/Websockets

Developer POV

1

GUI extensions



- JAVAscript commands
- Graphical Interface

2

JSON RPC bundle



- Browser callable methods
- Allows exporting data to cloud

3

Service bundle



- JAVA code implements functionality
- Interface HW/SW on platform

- Using the devices requires standard HDM APIs that are available at
 - http://dz.prosyst.com/pdoc/mBS_SH_SDK_7.3.0/modules/hdm/jsonrpc/devices.html

Home Device Access

Methods for accessing and controlling devices.

Method	Return Value	Parameters	Parameter Description
HDAccess/getHomeDevices	HomeDevice[]. Null can be returned if no devices are found against filter. Error will be returned if the filter is incorrect.	filter: string	LDAP filter for the needed devices. If it is null or empty string all devices are returned. The filter keys are case insensitive and can be: uid, name, vendor, version, type, device classes, status, adapter, parent uid, additional device property or device class object property (combination of device class name, dot and device class property).
HDAccess/getHomeDevices	HomeDevice[]. Null can be returned if no devices are found against filter for the specified protocol adapter. Error will be returned if the adapterName is incorrect or filter is incorrect.	adapterName: string filter: string	Name of protocol adapter LDAP filter for the needed devices. If it is null or empty string all devices of specified protocol adapter are returned. The filter keys are case insensitive and can be: uid, name, vendor, version, type, device classes, status, adapter, parent uid, additional device property or device class object property (combination of device class name, dot and device class property).
HDAccess/getHomeDevice	HomeDevice. Error if UID is invalid.	uid: string	ID of the device.
HDAccess/getDeviceClassObject	DeviceClassObject. Error if UID or device class for this UID are invalid.	uid: string deviceClass: string doMetadataFlag: boolean	ID of the device. The device class name. If false then metadata for device class objects' properties and operations will not be included. Otherwise it is included.
HDAccess/getDeviceClassObjects	DeviceClassObject[]. Error if UID is invalid. Null can be return if the device doesn't have device class objects.	uid: string doMetadataFlag: boolean	ID of the device. If false then metadata for device class objects' properties and operations will not be included. Otherwise it is included.
HDAccess/getDCOPProperty	Supported JSON Types. Error if UID, device class or property name are invalid. Error if property can not be read for some reason as network error.	uid: string deviceClass: string propertyName: string	ID of the device. The device class name. The name of the property to be gotten.
HDAccess/getDCOPProperties	Supported JSON Types []. Error if UID or device class are invalid. If some of properties can not be read the related element in array will be null.	uid: string deviceClass: string propertyNames: string[]	ID of the device. The device class name. The name of the properties to be gotten.
HDAccess/setDCOPProperty	Error if UID, device class, property name or property value are invalid. Error if property can not be set because of some reason as network error.	uid: string deviceClass: string propertyName: string propertyValue: Supported JSON Types	ID of the device. The device class name. The name of the property to be set. The value that needs to be set to the specified property. String can be used in value to represent number and boolean.
HDAccess/setDCOPProperties	object. Key is property name and value is boolean. If property is set, value is true else value is false. Error if UID, device class, property names or property values are invalid.	uid: string deviceClass: string properties: object	ID of the device. The device class name. A set of (key, value) pairs. The key is the name of the property to be set and the value is the value of the property. The key must be a string and value must be Supported JSON Types. String can be used in values to represent number and boolean.

Terminal

Web Admin Console - Home Devices

localhost:8080/

Back Forward

Most Visited Getting Started

Web Admin Console Home Device

Bundles Configuration

total devices: 0 online devices: 0

Devices Zones

Name	UID

1. Server Side Setup:
Install all dependent bundles using kitman source file
To install all packages:
kitman.i "SEP2 HDM Adapter"
2. To run deno:
Open browser and goto url: "http://localhost:8080/segui"
Server GUI will be up and running and we can now use to add/remove devices and make basic property changes.
3. Client

goto

```

mridup@mridup: ~/recordings
(10790420 of which were video data and 1013604 audio data)

Cleaning up cache...
Done!!!
Goodbye!
mridup@mridup:~/recordings$ rm out.ogv
mridup@mridup:~/recordings$ recordmydesktop
Initial recording window is set to:
X:0 Y:0 Width:1280 Height:1024
Adjusted recording window is set to:
X:0 Y:0 Width:1280 Height:1024
Your window manager appears to be Compiz

Detected compositing window manager.
Reverting to full screen capture at every frame.
To disable this check run with --no-wm-check
(though that is not advised, since it will probably produce faulty results).

Initializing...
Buffer size adjusted to 4096 from 4096 frames.
Opened PCM device default
Recording on device default is set to:
1 channels at 22050Hz
Capturing!
X Error: BadAccess (attempt to access private resource denied)
Bad Access on XGrabKey.
Shortcut already assigned.
X Error: BadAccess (attempt to access private resource denied)
Bad Access on XGrabKey.
Shortcut already assigned.
X Error: BadAccess (attempt to access private resource denied)
Bad Access on XGrabKey.
Shortcut already assigned.
X Error: BadAccess (attempt to access private resource denied)
Bad Access on XGrabKey.
Shortcut already assigned.

```

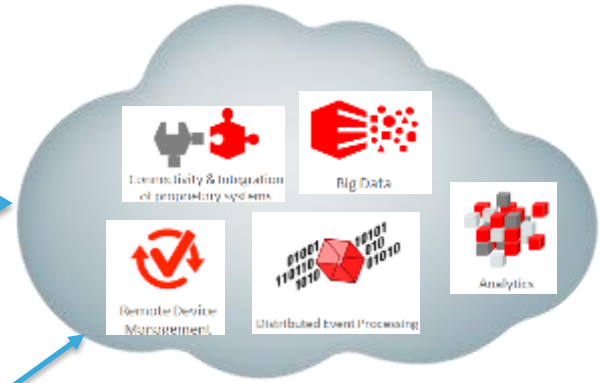
A product example



KLIKA·TECH

Remote Access

Local access



A product example



TATUNG

ORACLE



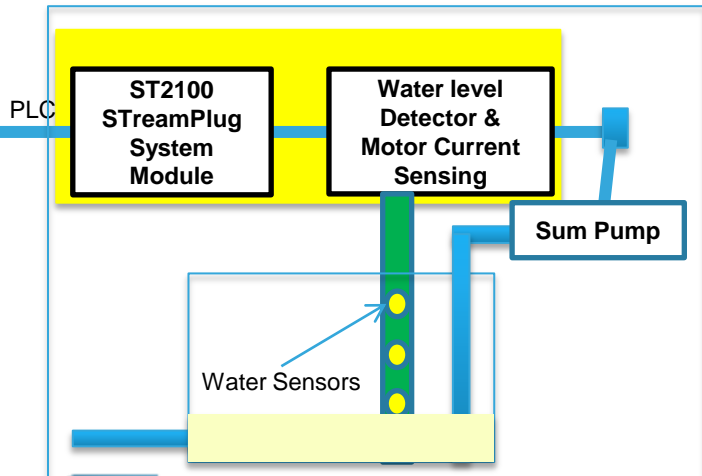
KLIKA·TECH

arKados



ProSyst

GRID2HOME
SMART ENERGY SOLUTIONS



- The American Society of Home Inspectors (ASHI) estimates that over 60% of US single family homes experienced basement water intrusion problems at some point.
- According to the American Housing Survey, in 2011, 14 million homeowners suffered from damage due to water, freezing and mold, resulting in \$11 billion in property loss claims, as reported by the Insurance Information Institute.
- Main functions:
 - Alarms and alerts via email
 - Monitoring and control of the sump pump via an app
 - Sensor hub for HVAC and Water Heater temperature, Moisture, and other sensors
 - Detection of anomalies in the operation of the equipment and maintenance alerts

ORACLE

- ST and its partners have developed a comprehensive solution portfolio for Smart Home and Energy gateways
 - This presentation provided an overview of available HW/SW technologies
- ST software solution is based Java and OSGi
 - Java + OSGi benefits of modularity and easy software reuse
 - ST integration of hardware devices in a complete solution
- → Programmers can focus on applications development

WEDNESDAY, OCT 01, 2014

High Security for the Internet of Things with Java and a Secure Element

Fred Vaute, Master Principal Sales Consultant, Oracle

Thierry Bousquet, Application engineer at Secure Microcontroller Division, ST Microelectronics

Anne-Laure SIXOU, Smartgrid Marketing Manager, STMicroelectronics

16:30 - 17:30

Hilton - Golden Gate 4/5

CON2225

- **ST**
 - ST online: www.st.com
- **Tatung**
 - Gateway one is available for purchase
 - Product information
 - <http://www.tatung.com/Site/Detail/10664>
- **Java SE embedded**
 - Oracle Java SE Embedded 8u6 is available for download
 - Eight reasons to love Java 8
 - <http://www.oracle.com/us/technologies/java/oracle-voice/index.html>
- **Speaker contacts**
 - Oleg Logvinov – STMicroelectronics (oleg.logvinov@st.com)
 - Frédéric Vaute – Oracle (frederic.vaute@oracle.com)



Thank You