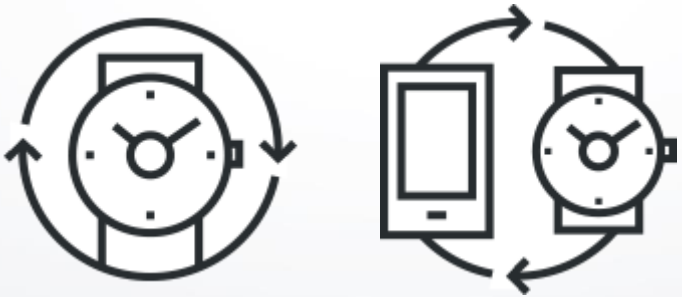
Wearable Sensors in Health Applications

Kamil Grondys

Samsung Electronics, Poland

Agenda

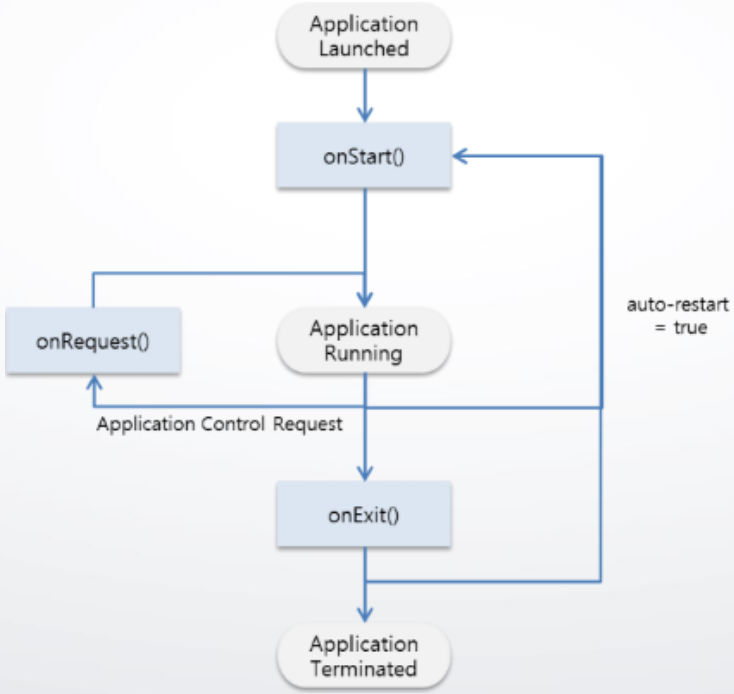


1. **Application Types**
2. Web vs Native Sensor API
3. Service Application
4. NPAPI & NPRuntime
5. Gear S2

Agenda

1. Application Types
2. **Web vs Native Sensor API**
3. Service Application
4. NPAPI & NPRuntime
5. Gear S2

Agenda



1. Application Types
2. Web vs Native Sensor API
3. **Service Application**
4. NPAPI & NPRuntime
5. Gear S2

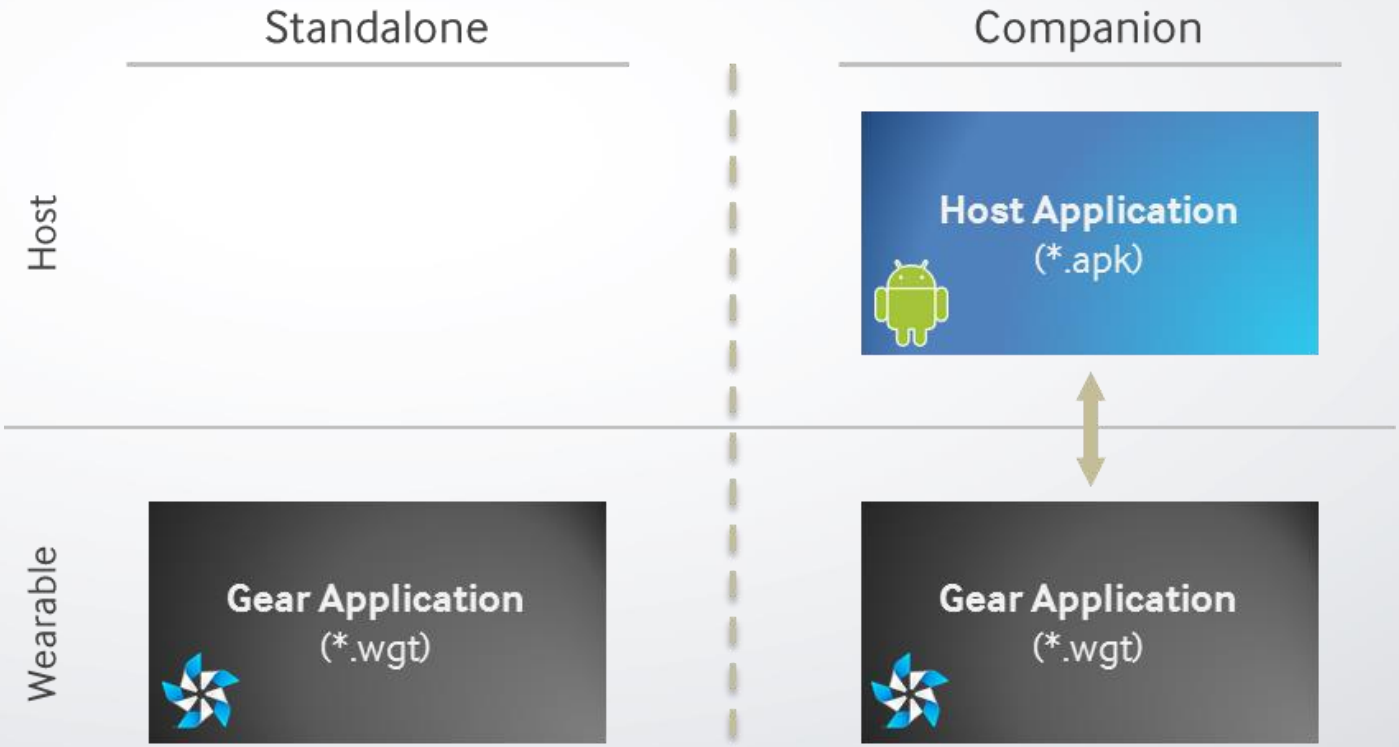
Agenda

1. Application Types
2. Web vs Native Sensor API
3. Service Application
4. **NPAPI & NPRuntime**
5. Gear S2

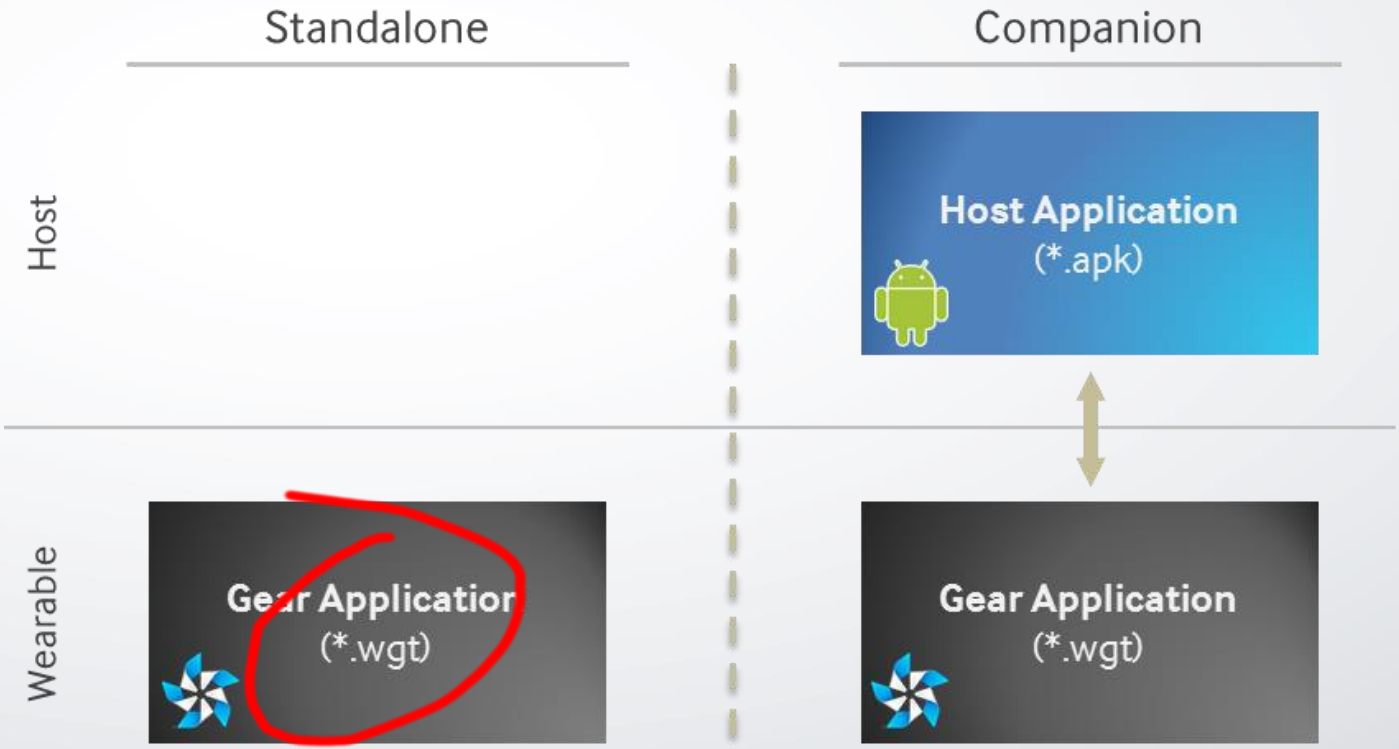
Agenda

1. Application Types
2. Web vs Native Sensor API
3. Service Application
4. NPAPI & NPRuntime
5. **Gear S2**

Application Types



Application Types



Sensor API

1. Magnetic
2. Ambient Light
3. Barometer
4. Proximity
5. Ultraviolet
6. Accelerometer
7. Gyroscope
8. Pedometer
9. Heart Rate Monitor

Sensor API

1. Magnetic
2. Ambient Light
3. Barometer
4. Proximity
5. Ultraviolet
6. Accelerometer
7. Gyroscope
8. Pedometer
9. Heart Rate Monitor

Sensor API

1. Magnetic
2. Ambient Light
3. Barometer
4. Proximity
5. Ultraviolet
6. Accelerometer
7. Gyroscope
8. Pedometer
9. Heart Rate Monitor

Sensor API

1. Magnetic
2. Ambient Light
3. Barometer
4. Proximity
5. Ultraviolet
6. Accelerometer
7. Gyroscope
8. Pedometer
9. Heart Rate Monitor

Sensor API

How to check which sensors are available on the device:

sdb shell -d

cat /etc/config/model-config.xml

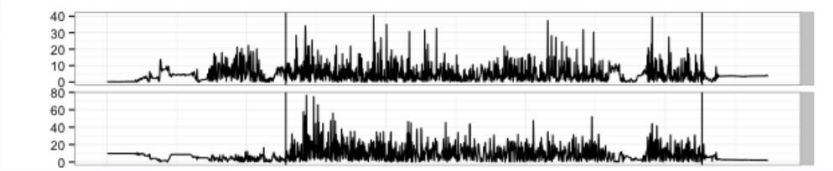
```
...
<key name="tizen.org/feature/sensor.accelerometer" type="bool">true</key>
<key name="tizen.org/feature/sensor.barometer" type="bool">false</key>
<key name="tizen.org/feature/sensor.gyroscope" type="bool">true</key>
<key name="tizen.org/feature/sensor.magnetometer" type="bool">false</key>
<key name="tizen.org/feature/sensor.photometer" type="bool">false</key>
<key name="tizen.org/feature/sensor.proximity" type="bool">false</key>
<key name="tizen.org/feature/sensor.tiltmeter" type="bool">false</key>
<key name="developer.samsung.com/tizen/feature/heart_rate_monitor" type="bool">true</key>
<key name="developer.samsung.com/tizen/feature/pedometer" type="bool">true</key>
```

Sensor API: Example apps

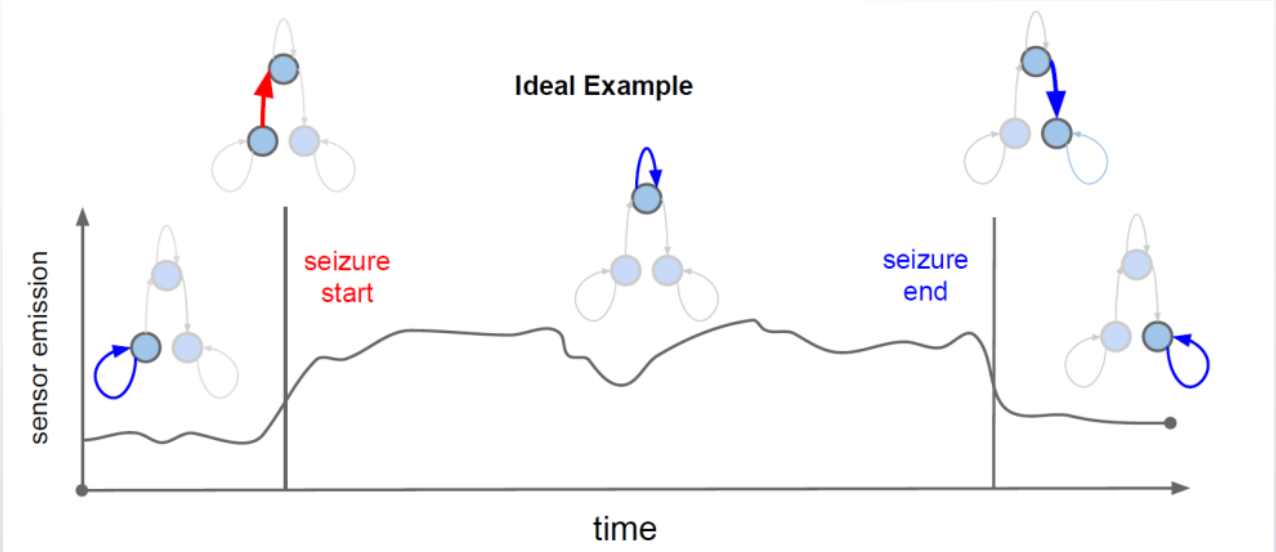
1. Fitness measurness (running, walking, etc.)
2. Navigation (distance)
3. Calories intake/outtake

Sensor API: example for epilepsy management

- Data from the accelerometer



- Epilepsy episode detection



Sensor API: Tizen API

```
enum SensorType {
  "LIGHT",
  "MAGNETIC",
  "PRESSURE",
  "PROXIMITY",
  "ULTRAVIOLET",
  "HRM_RAW"
};
```

1. MAGNETIC – Magnetic
2. LIGHT – Ambient Light
3. PRESSURE – Barometer
4. PROXIMITY – Proximity
5. ULTRAVIOLET – Ultraviolet

…

9. HRM_RAW – Heart Rate

Sensor API: HRM_RAW example

```
var HRMSensor hrm = tizen.sensorservice.getDefaultSensor("HRM_RAW");

hrm.start(function(){

  //on success
  hrm.getHRMRawSensorData(function(hrmSensorData){
    //callback to be invoke when sensor data has been read
    //do something with a data from the sensor
    //...
  });

});
```

Sensor API: HRM_RAW example

```
var HRMSensor hrm = tizen.sensorservice.getDefaultSensor("HRM_RAW");

hrm.start(function(){

  //on success
  hrm.getHRMRawSensorData(function(hrmSensorData){
    //callback to be invoke when sensor data has been read
    //do something with a data from the sensor
    //...
  });

});
```

Sensor API: HRM_RAW example

```
var HRMSensor hrm = tizen.sensorservice.getDefaultSensor("HRM_RAW");

hrm.start(function(){

  //on success
  hrm.getHRMRawSensorData(function(hrmSensorData){
    //callback to be invoke when sensor data has been read
    //do something with a data from the sensor
    //...
  });

});
```

Sensor API: Code example for Gear 2


```
var hrmSensor = webapis.motion.getDefaultSensor("HRM");
hrmSensor.start("HRM", function(){
  //on success
  hrmSensor.getHRMRawSensorData(function(hrmSensorData){
    //sensor data success callback to be invoked when sensor data has been read
    //do something with a data from the sensor
    //...
  });
});
```

Sensor API: W3C

6. Accelerometer

```
function handleMotionEvent(event)
{
  var x = event.accelerationIncludingGravity.x;
  // or event.acceleration to receive data excluding gravity
  var y = event.accelerationIncludingGravity.y;
  var z = event.accelerationIncludingGravity.z;
  // Do something awesome.
}

window.addEventListener("devicemotion", handleMotionEvent, true);
```

Sensor API: W3C

7. Gyroscope

```
function handleDeviceOrientationEvent(event)
{
  var alpha = event.alpha;
  var beta = event.beta;
  var gamma = event.gamma;
  // Do something awesome.
}

window.addEventListener("deviceorientation", handleDeviceOrientationEvent, true);
```

Sensor API: Pedometer Web vs Native

- Web

```
enum PedometerStepStatus {

 "NOT_MOVING",
 "WALKING",
 "RUNNING"

};
```

- Native

```
typedef enum
{

ACTIVITY_STATIONARY,      /**< Stationary */
ACTIVITY_WALK,            /**< Walking */
ACTIVITY_RUN,             /**< Running */
ACTIVITY_IN_VEHICLE,      /**< In a moving vehicle */

} activity_type_e;
```

Sensor API: Native advantages

- Possible to get data all the time

```
typedef enum
{

SENSOR_OPTION_DEFAULT,          /**< Does not receive data when the LCD is off
*/
SENSOR_OPTION_IN_SCREEN_OFF,   /**< Receives data when the LCD is off */
SENSOR_OPTION_IN_POWERSAVE_MODE,  /**< Receives data in the PS mode */
SENSOR_OPTION_ALWAYS_ON,                /**< Always receives data */

} sensor_option_e;
```

Sensor API: Native advantages (set interval)

- Possible to get data all the time

```
typedef enum
{

SENSOR_OPTION_DEFAULT,        /**< Does not receive data when the LCD is off
*/
SENSOR_OPTION_IN_SCREEN_OFF,  /**< Receives data when the LCD is off */
SENSOR_OPTION_IN_POWERSAVE_MODE, /**< Receives data in the PS mode */
SENSOR_OPTION_ALWAYS_ON,              /**< Always receives data */

} sensor_option_e;
```

- Possible to get even more accurate data

```
int sensor_listener_set_interval(sensor_listener_h listener,
    unsigned int interval_ms);
```

Sensor API: Privileges

Privilege: http://tizen.org/privilege/healthinfo
Privilege: http://tizen.org/privilege/medicalinfo

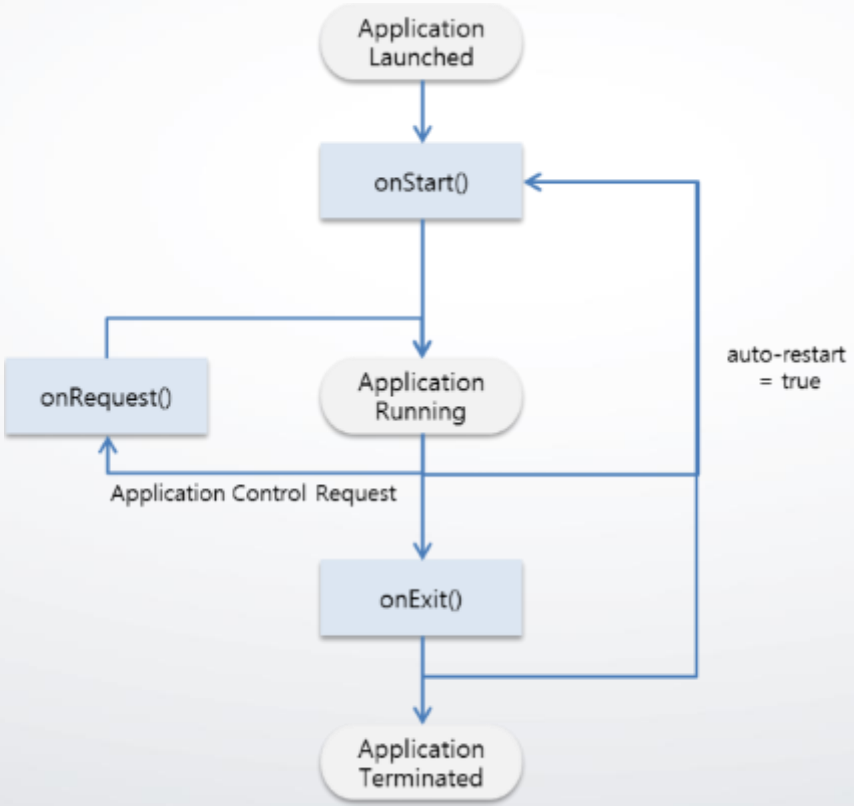
Web vs Native Sensor API: Summary

1. JavaScript for Web Sensor API
2. tizen.sensorservice since Tizen 2.3
3. webapis.motion for older devices

4. c/c++ for Native Sensor API
5. Additional options (interval, when to collect data)
6. Higher frequency rate

7. Don’t forget to add privs

Web Service Application



1. Runs in the background
2. node.js is used as runtime
3. Partner Level cert. is required
4. Gear 2 not supported

Web Service Application: when to use

1. Data processing without blocking UI
2. Can be launched during a device boot
3. Can communicate with UI application

https://developer.tizen.org/development/tutorials/web-application/tizen-features/service-application

Web Service Application: features

```
<widget>
    <tizen:service id="[App_ID]" auto-restart="true" on-boot="true">
       <tizen:content src="[Start_JS_File]"/>
       <tizen:name>[App_Name]</tizen:name>
       <tizen:icon src="[App_Icon]"/>
       <tizen:description>[Description]</tizen:description>
    </tizen:service>
</widget>
```

Web Service Application: features

```
<widget>
    <tizen:service id="[App_ID]" auto-restart="true" on-boot="true">
       <tizen:content src="[Start_JS_File]"/>
       <tizen:name>[App_Name]</tizen:name>
       <tizen:icon src="[App_Icon]"/>
       <tizen:description>[Description]</tizen:description>
    </tizen:service>
</widget>
```

Web Service Application: example

```
module.exports.onStart = function() {
    console.log("service start");
    ...
}

module.exports.onRequest = function() {
    console.log("service received appControl request");
    ...
}

module.exports.onExit = function() {
    console.log("service terminate");
    ...
}
```

Web Service Application: Summary

1. JavaScript service application
2. Can be launched automatically
3. For Partners only
4. Supported since Gear S

NPAPI & NPRuntime

1. Plugins for web
2. Available on web since Netscape Navigator web browser
3. c/c++ available for JavaScript
4. Partner Level cert. required

NPPlugin: how the plugins works

- HTML5 code:

```
<object id="plugin" type="application/x-plugin-example"></objec
t>
```

- JavaScript code:

```
var plugin = document.getElementById('plugin')

plugin.someMethod();
console.log(plugin.someProperty);
```

NPPlugin: why?

1. Data from a sensor can be obtained with higher freq. rate
2. c/c++ can be used for data processing
3. AI and gesture recognition alghorithms can be used
4. Custom codecs can be implemented for Web Apps

NPPlugin & NPRuntime: Summary

1. Call c++ code from JavaScript
2. Tizen Native API avaialble
3. Partner level cert. required
4. Available in Gear 2, Gear S

Gear S2

1. Tizen **Wearable Native API** available
2. Tizen Wearable Web API avaialable
3. Tizen 2.3.1

Summary

1. Health applications often collects data from sensors
2. Sometimes that apps need to work in the background
3. Sometimes an application need to work all the time
4. Wearables can collect and proccess data
5. You can use both HTML5/JavaScript and C/C++

Resources

1. developer.tizen.org
2. developer.samsung.com


KamilGrondys


KamilGrondys

k.grondys@samsung.com