The Vorto project proposal
EclipseCon 2014
# Bosch Key Figures 2013

**Bosch Group**
- 46.4 billion EUR in sales
- 281,400 associates
- 225 manufacturing sites

### Bosch Group Segments

<table>
<thead>
<tr>
<th>Segment</th>
<th>Sales Share</th>
<th>Associates</th>
<th>Manufacturing Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Automotive Technology</strong></td>
<td>66%</td>
<td>174,400</td>
<td>140</td>
</tr>
<tr>
<td><strong>Industrial Technology</strong></td>
<td>15%</td>
<td>33,500</td>
<td>33</td>
</tr>
<tr>
<td><strong>Energy and Building Technology</strong></td>
<td>9%</td>
<td>73,500</td>
<td>52</td>
</tr>
<tr>
<td><strong>Consumer Goods</strong></td>
<td>10%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Regions

<table>
<thead>
<tr>
<th>Region</th>
<th>Sales Share</th>
<th>Associates</th>
<th>Manufacturing Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Europe</strong></td>
<td>55%</td>
<td>174,400</td>
<td>140</td>
</tr>
<tr>
<td><strong>Americas</strong></td>
<td>21%</td>
<td>33,500</td>
<td>33</td>
</tr>
<tr>
<td><strong>Asia Pacific</strong></td>
<td>24%</td>
<td>73,500</td>
<td>52</td>
</tr>
</tbody>
</table>

2 Including other regions
The Vorto project proposal

Bosch Software and Systems House

Overview Bosch Software Innovations

Industry Focus
- Mobility, Energy, Industry, City & Building

Bosch IoT Suite
- BPM, BRM, M2M

Professional Services
- Planning, Implementation, Training and Operation of Solutions, based on our Software Suite

Locations:
- Berlin, Immenstaad, Stuttgart, Chicago, Palo Alto, Singapur, and Shanghai

Employees:
- ~ 500 worldwide
Bosch Software Innovations: The Systems and Software House for Internet of Things Solutions

Bosch IoT Suite

**Smart Energy**
- Smart Metering
- VPP (Virtual Power Plant)

**Smart Mobility**
- Fleet Management
- eMobility Solutions

**Connected Industry**
- Service Portal

**Smart Home**
- Intelligent Solutions
Bosch IoT Platform Strategy

➤ Overall goal:
   - Business Success for Bosch Group in IoT with Products and Services in the Verticals Mobility, Industry, Energy, Building
   - Interconnections and Interoperability of Devices/Services (therefore Bosch should “back on the right (platform) horse”)

➤ Assumption:
   - 2-5 major IoTS platforms (in next 5-7 years)
   - At least one of them will be Open Source
   - Bosch not able to develop one of these proprietary platforms alone and customers/partners would not accept it
   - No risk/dependency on proprietary 3rd party platform

➤ Conclusion
   - Open Platform strategy with OSS
There are four scenarios that shall be addressed by the Vorto approach:

1. **Consumers** want to use a large variety of devices in their ecosystem and don't want to be limited to using devices of one specific vendor.

   - **Flexibility, Ease of use**

2. **Vendors of IoT devices** want to increase the number of ecosystems where their devices can be integrated.

   - **Increase sales**

3. **Vendors of IoT platforms** want to integrate as much as devices as possible into their ecosystem without major effort.

   - **Increase sales, become standard**

4. **Application developers** want to support a broad range of devices without a need to develop vendor specific code.

   - **Increase sales, reduction of complexity**
The Vorto project proposal

Scenario 3 and 4: Platform vendor / Application developer

- Bosch M2M
- ThingWorx
- Qivicon
- Device 1
- Device 2
- Device 3
- Device 4
- Device 5
The Vorto project proposal

An information model is an abstract representation of the functionalities, properties, and status of a real object.
An **information model** is an abstract representation of the functionalities, properties, and status of a real object.
The Vorto project proposal

Idea: Creation of a flexible meta information model and code generators for generating specific representations

- Meta Model
- Qivicon
- Bosch M2M
- Qivicon Driver
- Bosch Code Generator
- ThingWorx Code Generator
- ThingWorx
- Bosch Driver
- Bosch
- Qivicon Code Generator
- Device i
- Instance of
- Information Model of Device i
- generates
- uses
The goal of the Vorto project is to enable a global standardization.
The Vorto project proposal

Information models are standardized by means of a flexible and sustainable **meta information model**

**Meta Information Model**

Specifies the structure of derived information models

Eclipse EMF based Meta Information Model

Instance of

Info model

Display

Info model

Camera
The Vorto project proposal

Information models that conform to the meta information model can be authored using the **IoT Tool Set**

Eclipse Platform based **IoT Tool Set**
The Vorto project proposal

The initial version of the IoT Tool Set is realized as an Eclipse plugin and provides a textual DSL editor

- The grammar of the DSL corresponds to the meta information model
- Auto completion and syntax highlighting allow for comfortably creating information models

- A graphical environment for creating information models shall be added
- The Tool Set shall allow for connecting to the information model repository

Part of the initial contribution

In scope of the project
The Vorto project proposal

The IoT Tool Set allows for creating information model based implementations using **Code Generator** plugins.

**IoT Tool Set**

The code generator extension point of the IoT Tool Set allows developers to create additional (domain specific) code generators.
The Vorto project proposal

The central **Information Model Repository** allows for managing and providing existing information models.

When version 1.0 of the repository is released it shall be **hosted by Eclipse** to promote a worldwide standardization of information models.
The Vorto project proposal

The components of the Vorto project in combination at a glance

- Governance
- Repository
- Import
- IoT Tool Set
  - OSGi DAL Code Generator
  - Code Generator Extension Points
  - OSGi
  - oneM2M
  - HGi
  - ...

OSGi
oneM2M
HGi
...
Use Case: A device vendor creates a new smoke detector and an openHab community member wants to integrate it

Vendor A creates an information model for his new Z-Wave smoke detector using the IoT Toolset.

After Vendor A has finished his work he publishes his new information model into the repository.

openHab Community member uses an openHab code generator in combination with the IoT Tool set to create the related items.

openHab community member would like to include it into his/her environment.

(openHab community member adds specific Z-Wave information to the generated items.)

Bosch Software Innovations

EclipseCon | INST / TST | 29/10/2014 | © Bosch Software Innovations GmbH 2014. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
The Vorto project proposal

A UI Mock-Up for the repository and code generator integration.
The Vorto project proposal

A UI Mock-Up for the repository and code generator integration.
The Vorto project proposal

Technical Aspects:

- Is based on Ecore (part of the Eclipse Modeling Framework).
- Attributes and Capabilities which are described in the information meta model are realized as Ecore compliant Java interfaces and implementation classes.
- The meta information model contains also technical validation functionalities for meta model instances.

The Meta Information Model

- The DSL editor is based on the Eclipse XText framework.
- The grammar of the DSL is based on the meta information model.
- The code assistance, and the syntax highlighting information is an additional part of the DSL grammar.
- The editor provides a textual standard formatting of information models which allows for comfortably comparing different (versions of) information models.
- Allows for serializing the meta information model instances as user readable DSL text file and machine readable XMI file.
- The Tool Set provides an code generation extension point consisting of an interface Java class and an extension schema.

The IoT Tool Set
The Vorto project proposal

Technical Aspects:

- Must implement the „information model generator“ interface which is part of the code generator extension point
- Code generators are based on the information meta model
- Input for the code generators are specific instances of the information meta model
- Can be based for example on Eclipse XPand or JET...
- Code generators can be integrated into the tool set by registering the code generator as an extension of the given extension point.

The Code Generators

- TBD

The Repository
Contact Details
Olaf Weinmann
Olaf.Weinmann@bosch-si.com

Bosch ConnectedWorld Blog
www.blog.bosch-si.com