



SensIDL

Towards a generic framework for implementing sensor communication interfaces

Christoph Rathfelder

Emre Taspolatoglu



Gefördert durch:

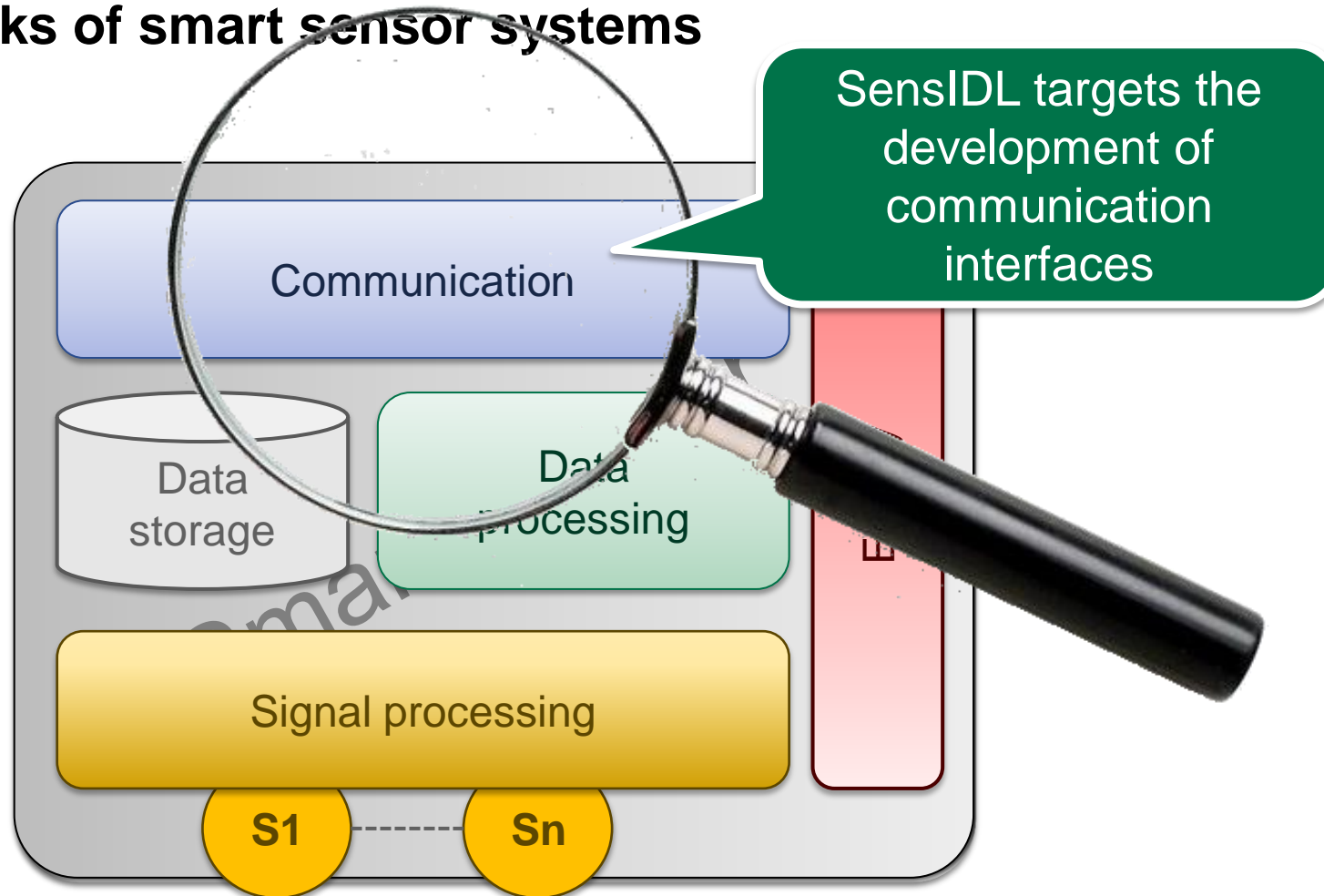


aufgrund eines Beschlusses
des Deutschen Bundestages

IGF-Vorhaben: 18363 N

Smart System and CPS Reference model

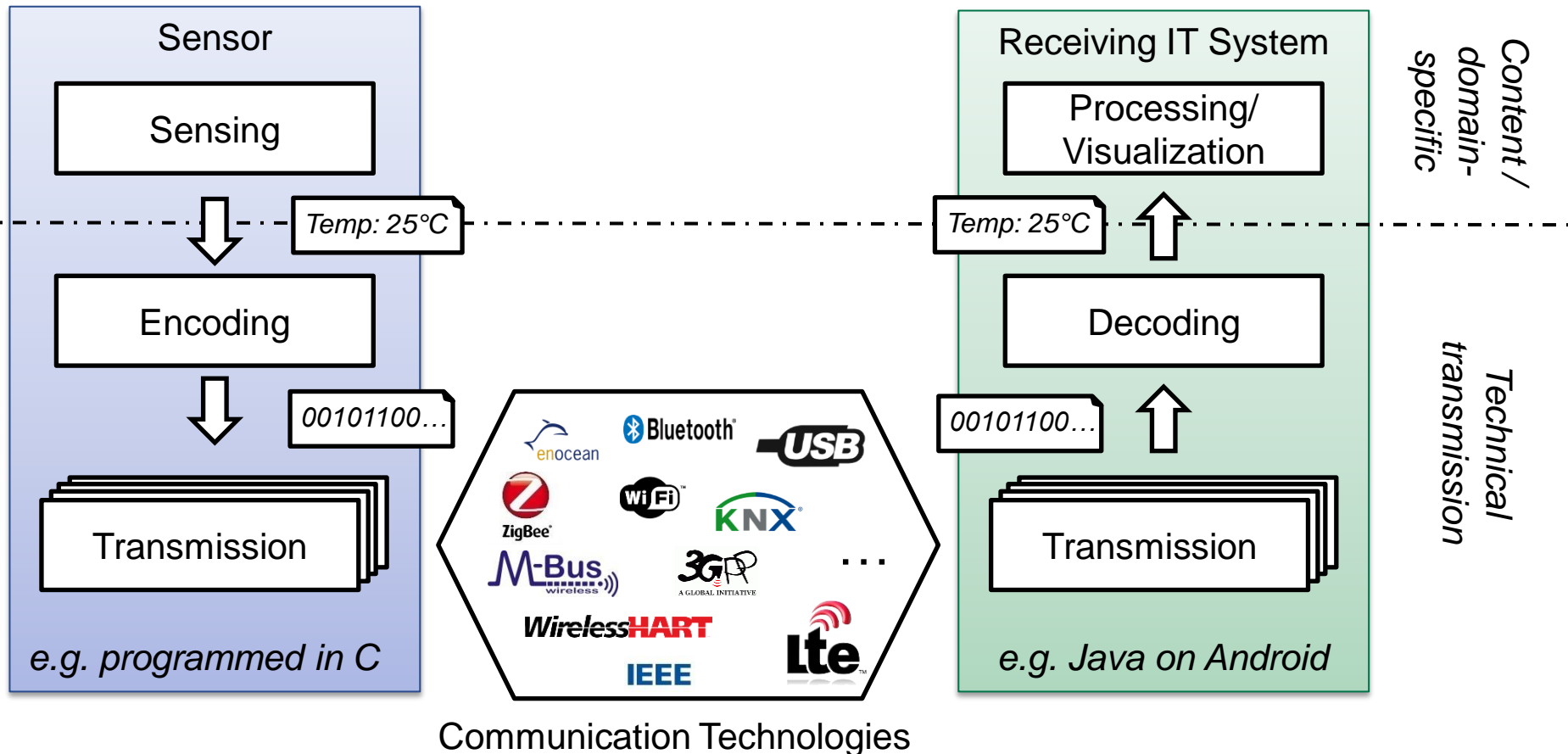
Building blocks of smart sensor systems



- A software toolbox for sensor developer
- Simplifying the implementation of communication interfaces



The long Way from Sensors to IT Systems



Objectives of SensIDL

Support for both developer roles

- Sensor and embedded developer
- Data processing within the receiving IT system



Simplification and automation of development steps

- Tool support
- Generation of code
- Documentation with additional value

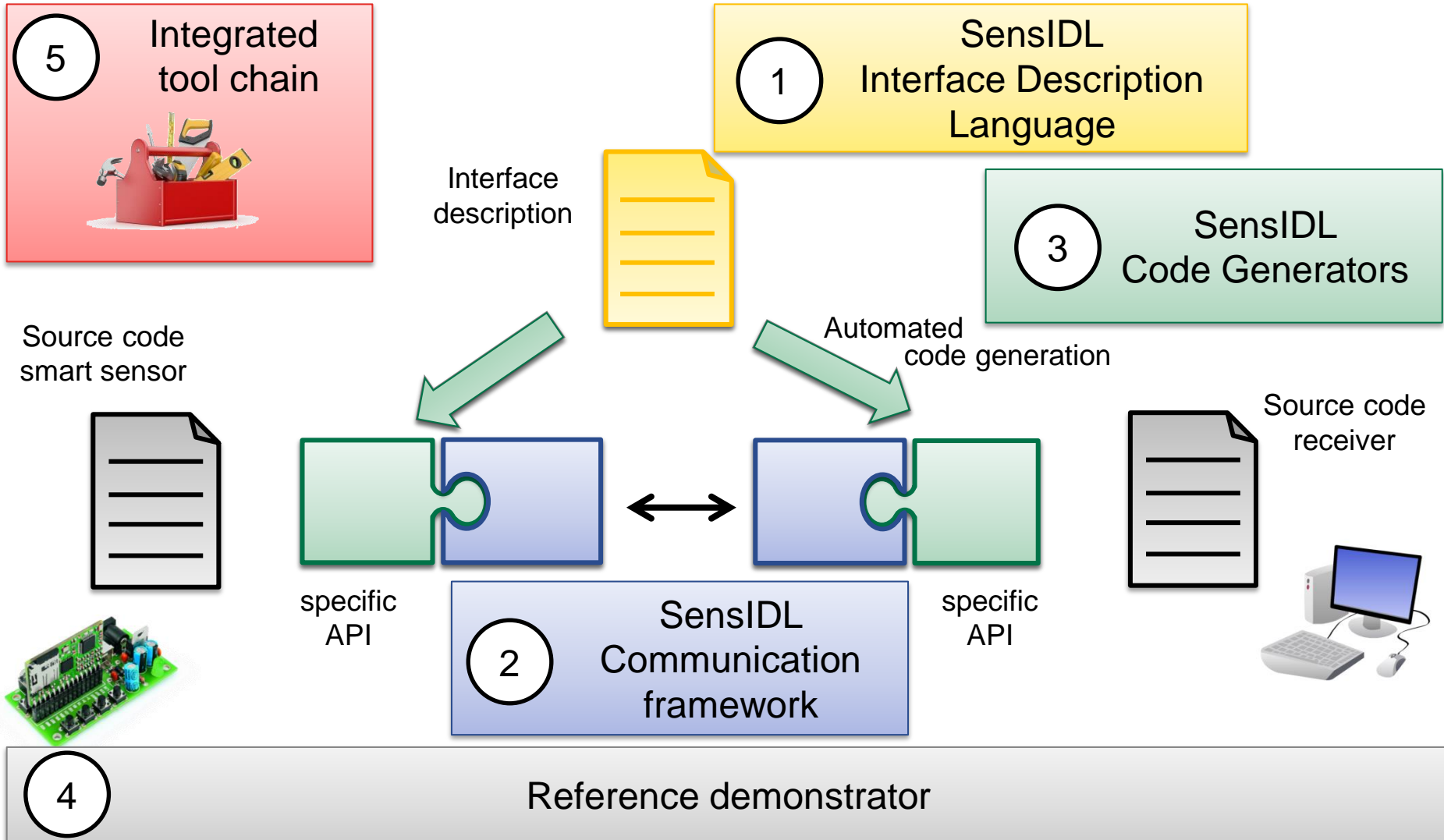


Increase of efficiency and quality

- Automation of recurring tasks
- Focusing on application-specific details



Expected SensIDL Results



Central Question to be Answered

Interface description language

- Which information needs to be modeled?
- Abstraction of implementation- and platform-specific details

Reference demonstrator

- Being representative for different application domains (e.g. Smart Home and Industrie / Production)
- Basis to derive code generators

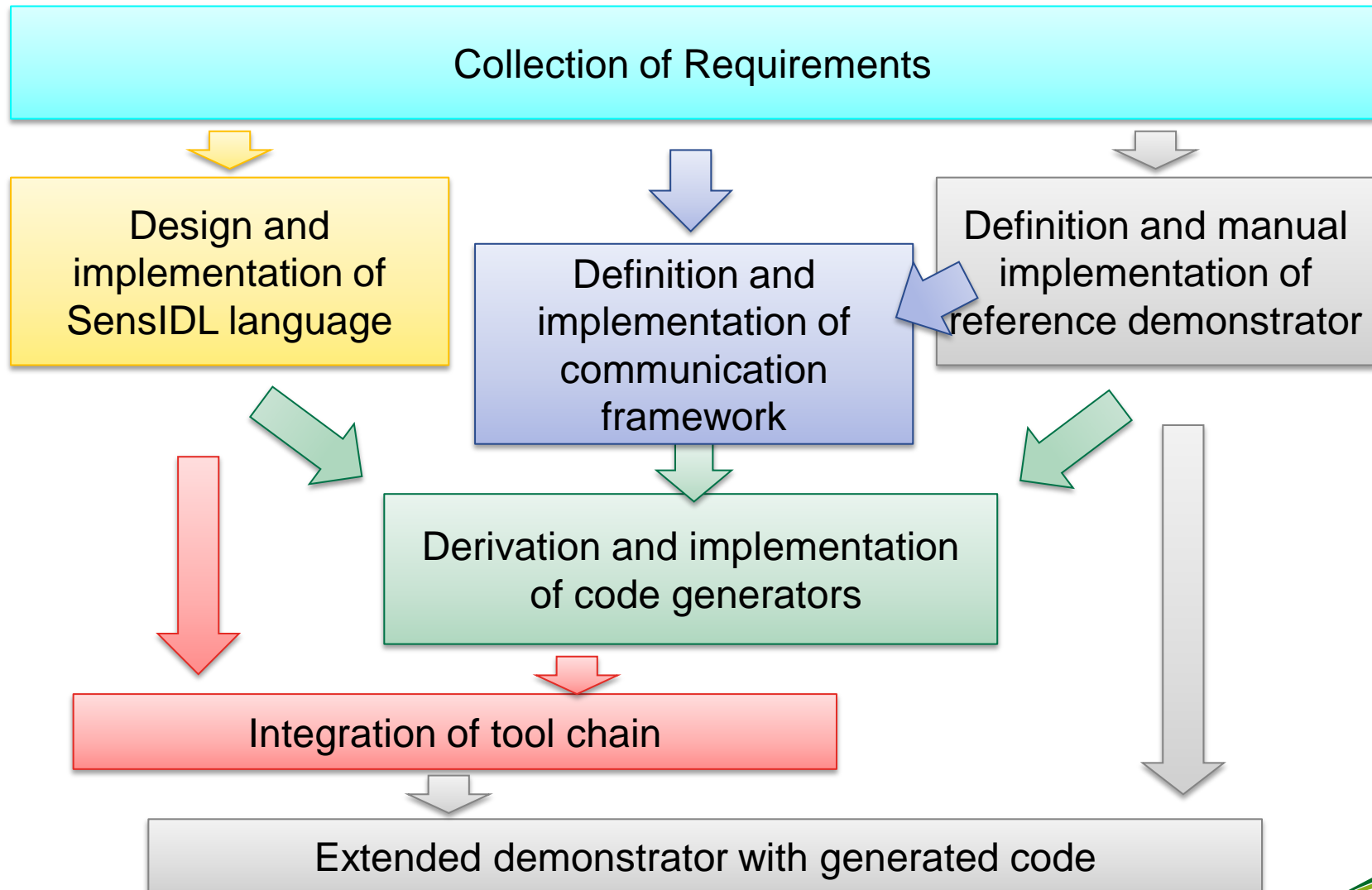
Communication

- How to transmit data in a generic and efficient way?

Code generators and automation

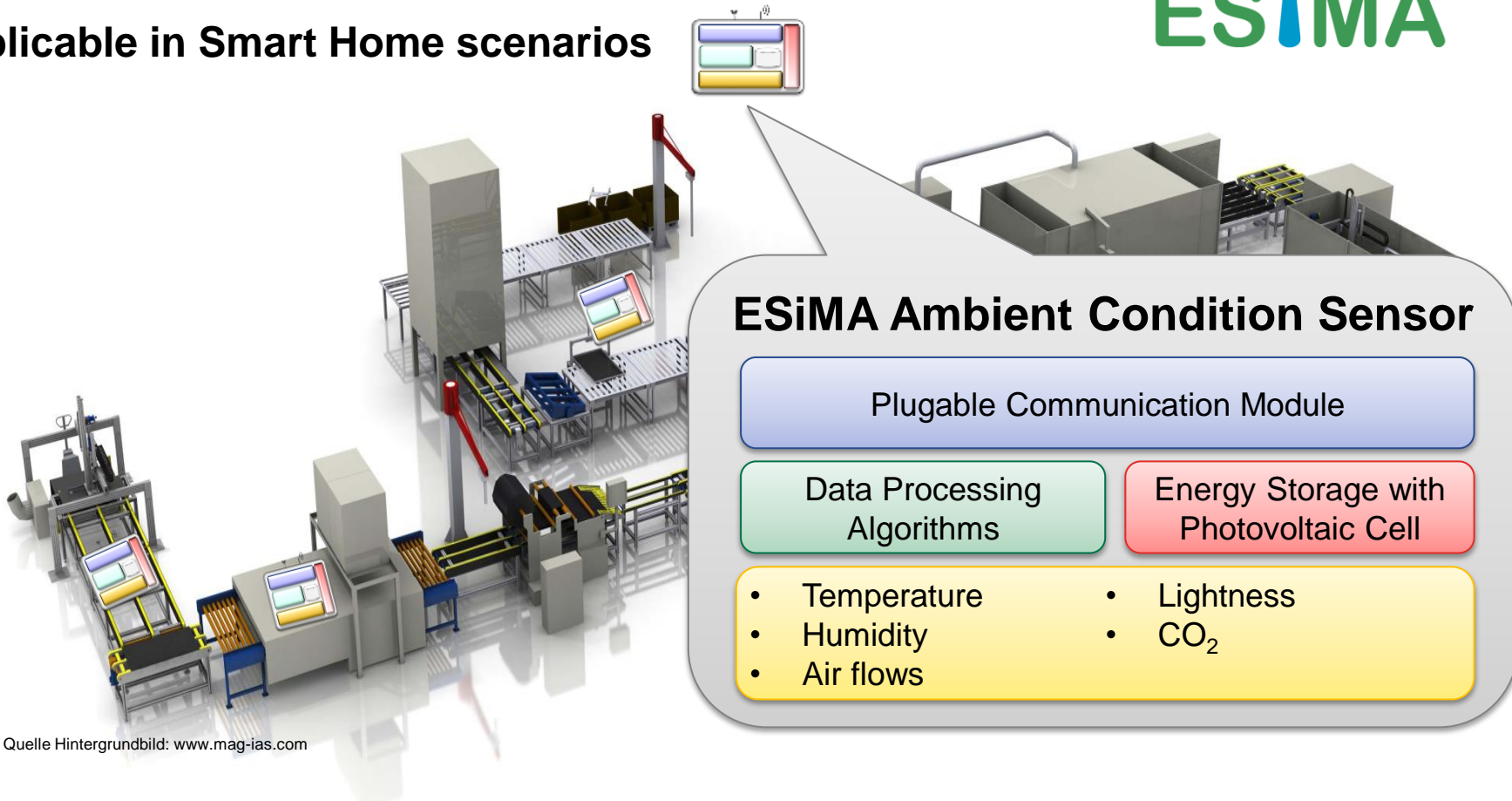
- Identification of recurring code fragments?
- Identification of recurring development tasks?

Project Roadmap



Initial Reference Demonstrator Based on an Industrial Project

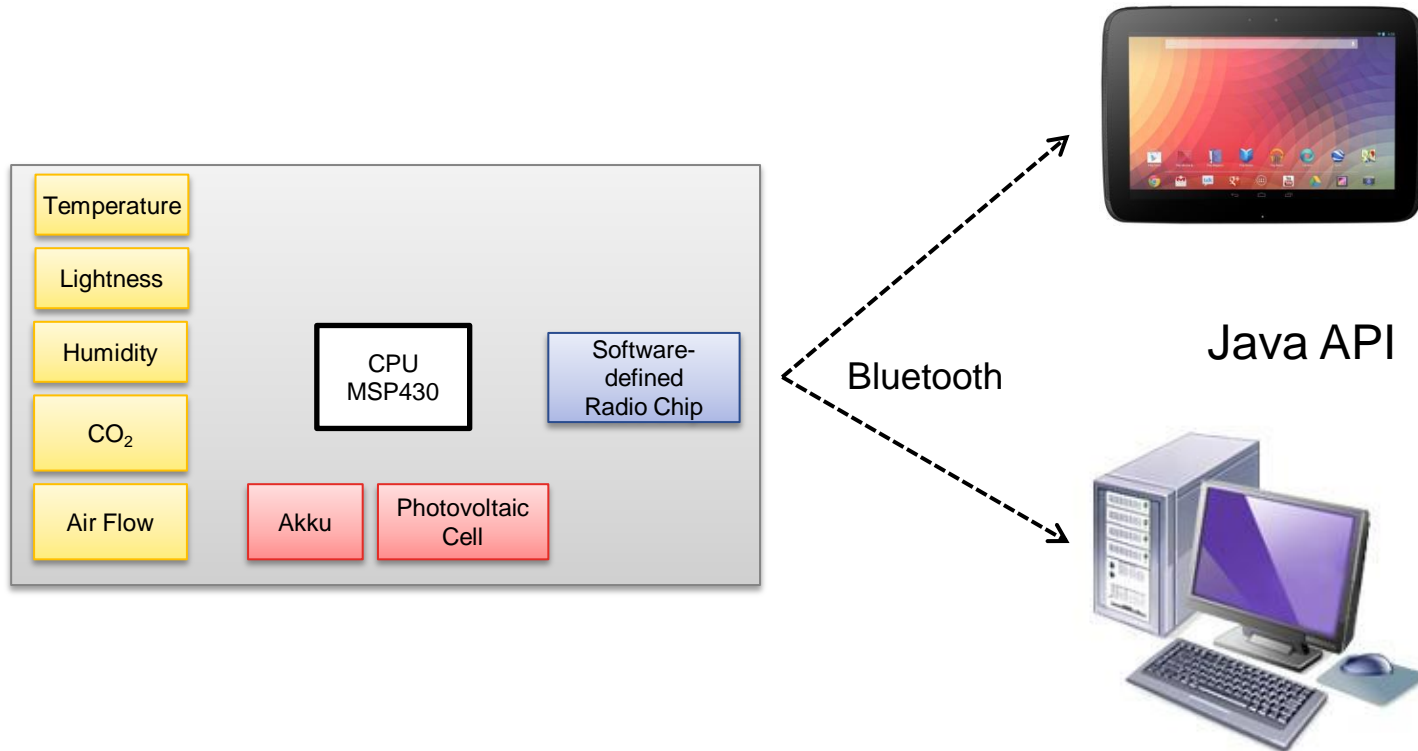
- Measuring energy flows and ambient conditions
- Extendable hardware basis
- Applicable in Smart Home scenarios



Quelle Hintergrundbild: www.mag-ias.com

Demonstrator – Step 1

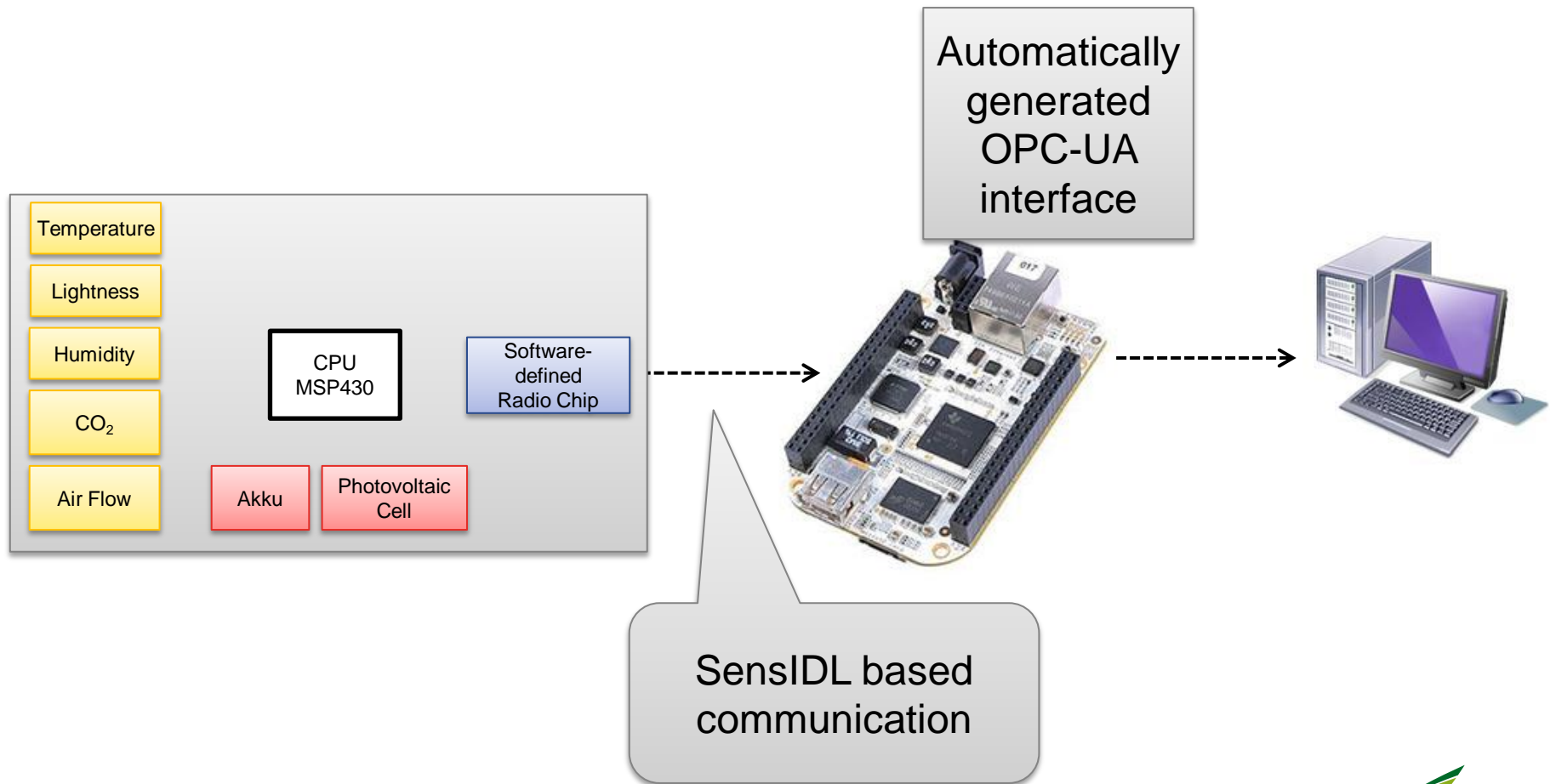
Direct Data Access



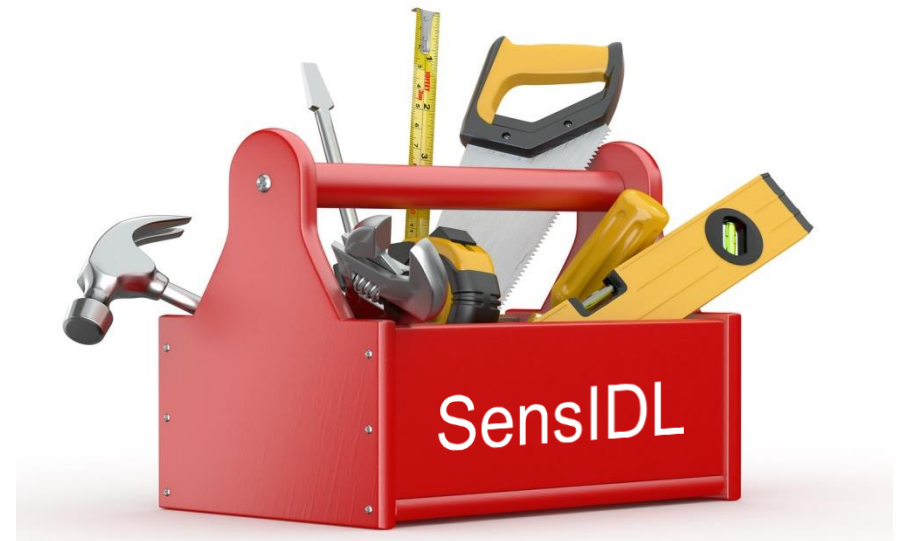
Demonstrator – Step 2

Gateway Generation

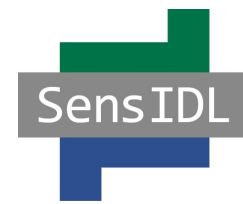
Generation of Technology Adaptors



SensIDL Toolchain



Used Technologies within the SensIDL Development Tool



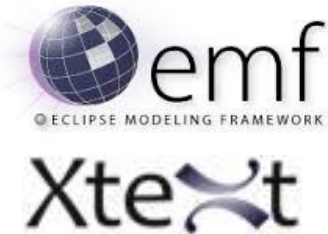
SensIDL Tool

- Eclipse-based plugins
- Integrated tool chain



SensIDL Language

- Model / language for describing sensor data
- Textual editor based on Xtext
 - Eventually additional graphical editor



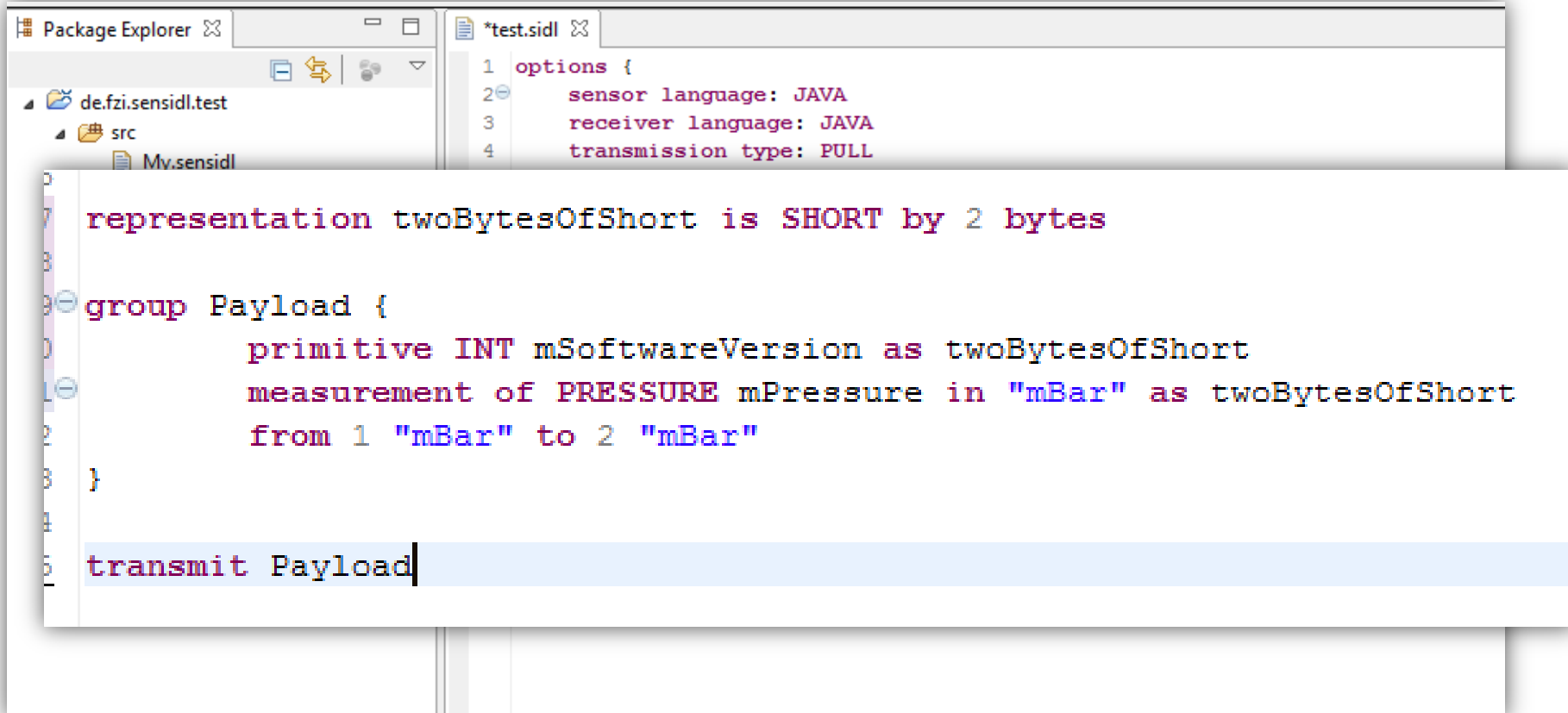
SensIDL Code Generators

- Code templates based on Xtend
- Automated code generation



Feasability Study

Textuel Editor

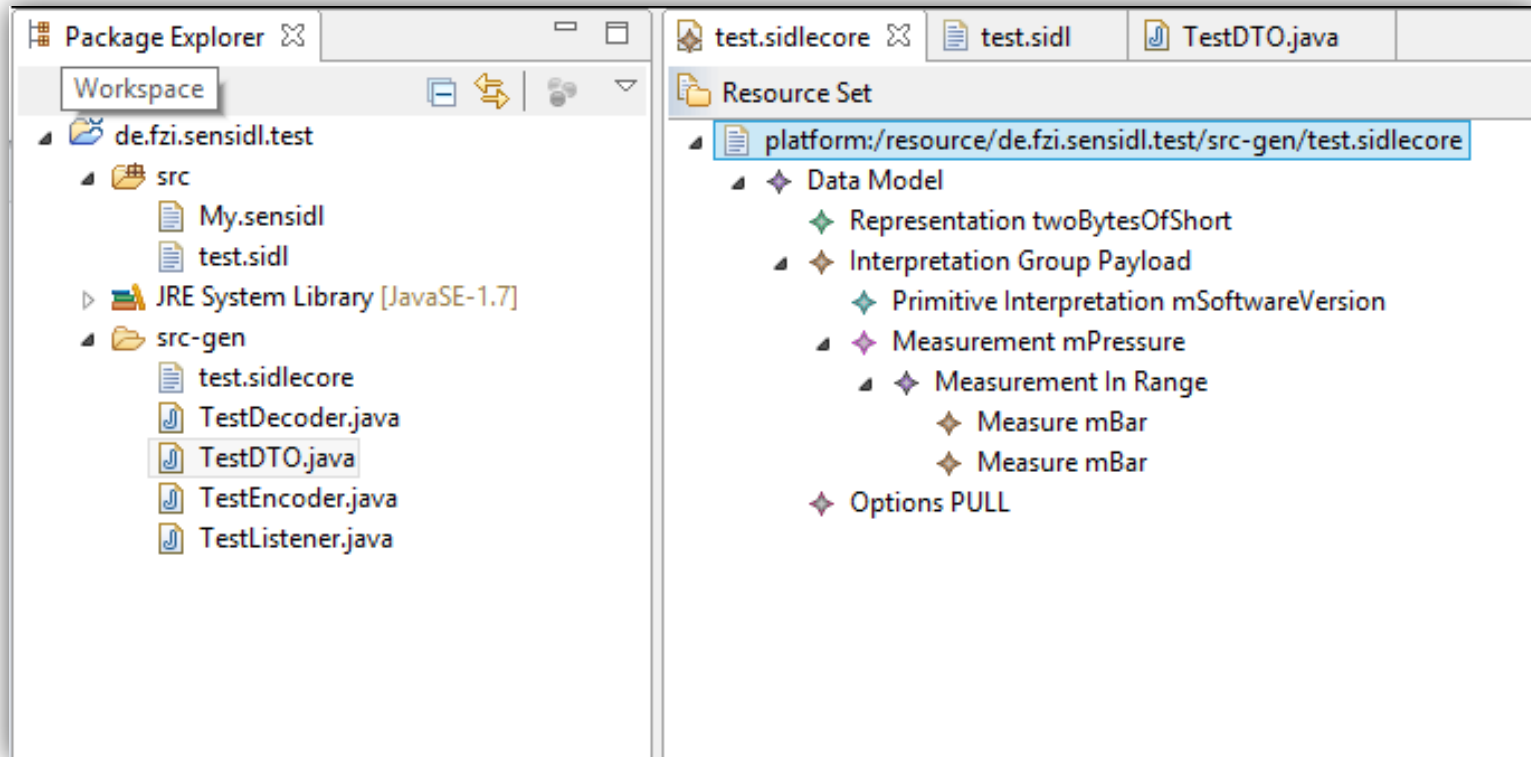


```
Package Explorer  [X]  [ ]  [ ]
└─ de.fzi.sensidl.test
   └─ src
      └─ My.sensidl

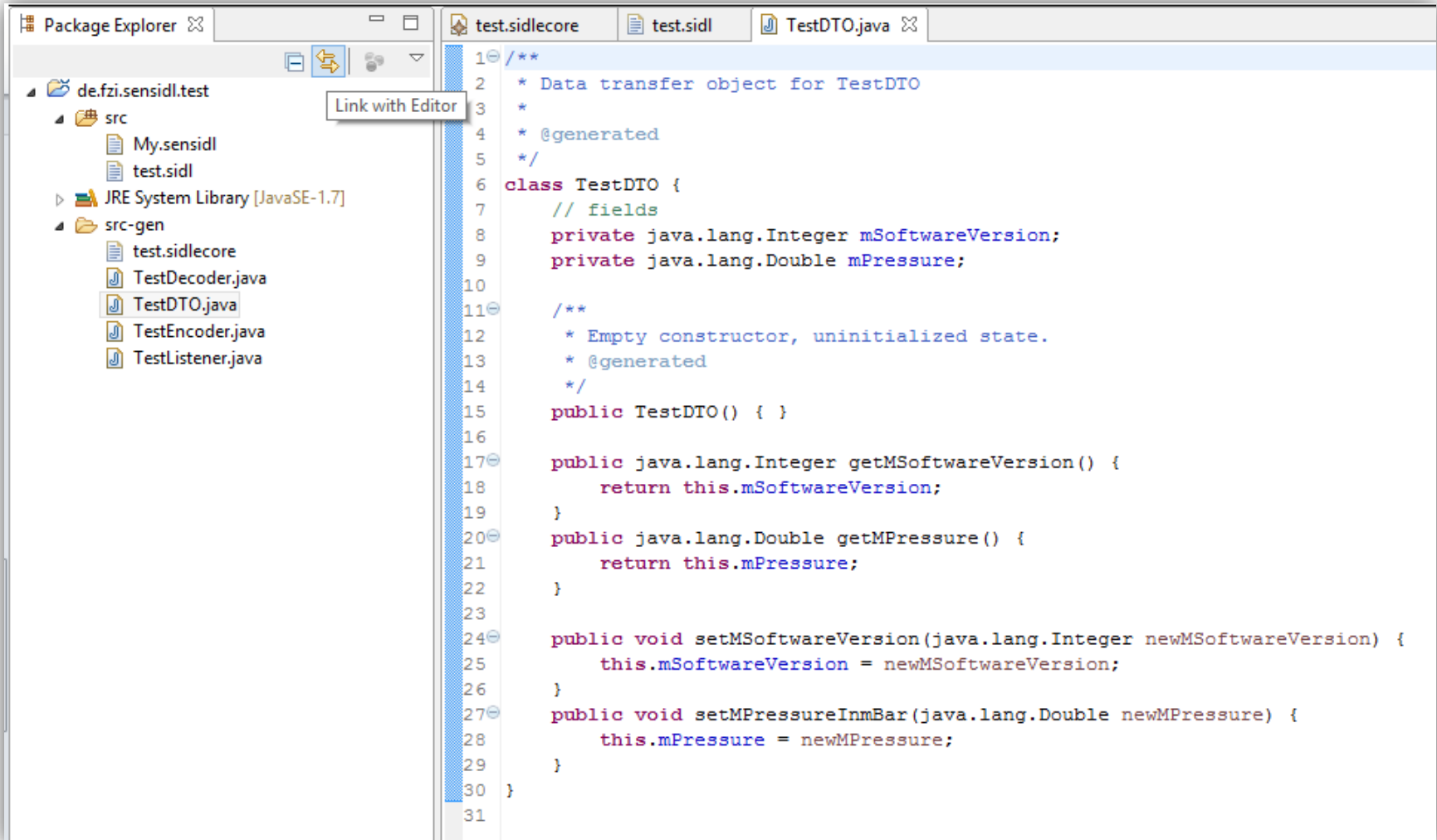
*test.sidl [X]
1 options {
2     sensor language: JAVA
3     receiver language: JAVA
4     transmission type: PULL
5
6
7 representation twoBytesOfShort is SHORT by 2 bytes
8
9 group Payload {
10     primitive INT mSoftwareVersion as twoBytesOfShort
11     measurement of PRESSURE mPressure in "mBar" as twoBytesOfShort
12     from 1 "mBar" to 2 "mBar"
13 }
14
15 transmit Payload
```

Feasibility Study

The EMF Model in the Background



Feasibility Study Code Generation



```
1 /**
2  * Data transfer object for TestDTO
3  *
4  * @generated
5  */
6 class TestDTO {
7     // fields
8     private java.lang.Integer mSoftwareVersion;
9     private java.lang.Double mPressure;
10
11     /**
12      * Empty constructor, uninitialized state.
13      * @generated
14      */
15     public TestDTO() { }
16
17     public java.lang.Integer getMSoftwareVersion() {
18         return this.mSoftwareVersion;
19     }
20     public java.lang.Double getMPressure() {
21         return this.mPressure;
22     }
23
24     public void setMSoftwareVersion(java.lang.Integer newMSoftwareVersion) {
25         this.mSoftwareVersion = newMSoftwareVersion;
26     }
27     public void setMPressureInmBar(java.lang.Double newMPressure) {
28         this.mPressure = newMPressure;
29     }
30 }
31
```


Who we are

Collaborating Research Institutes



- Software engineering
- Model-driven software development



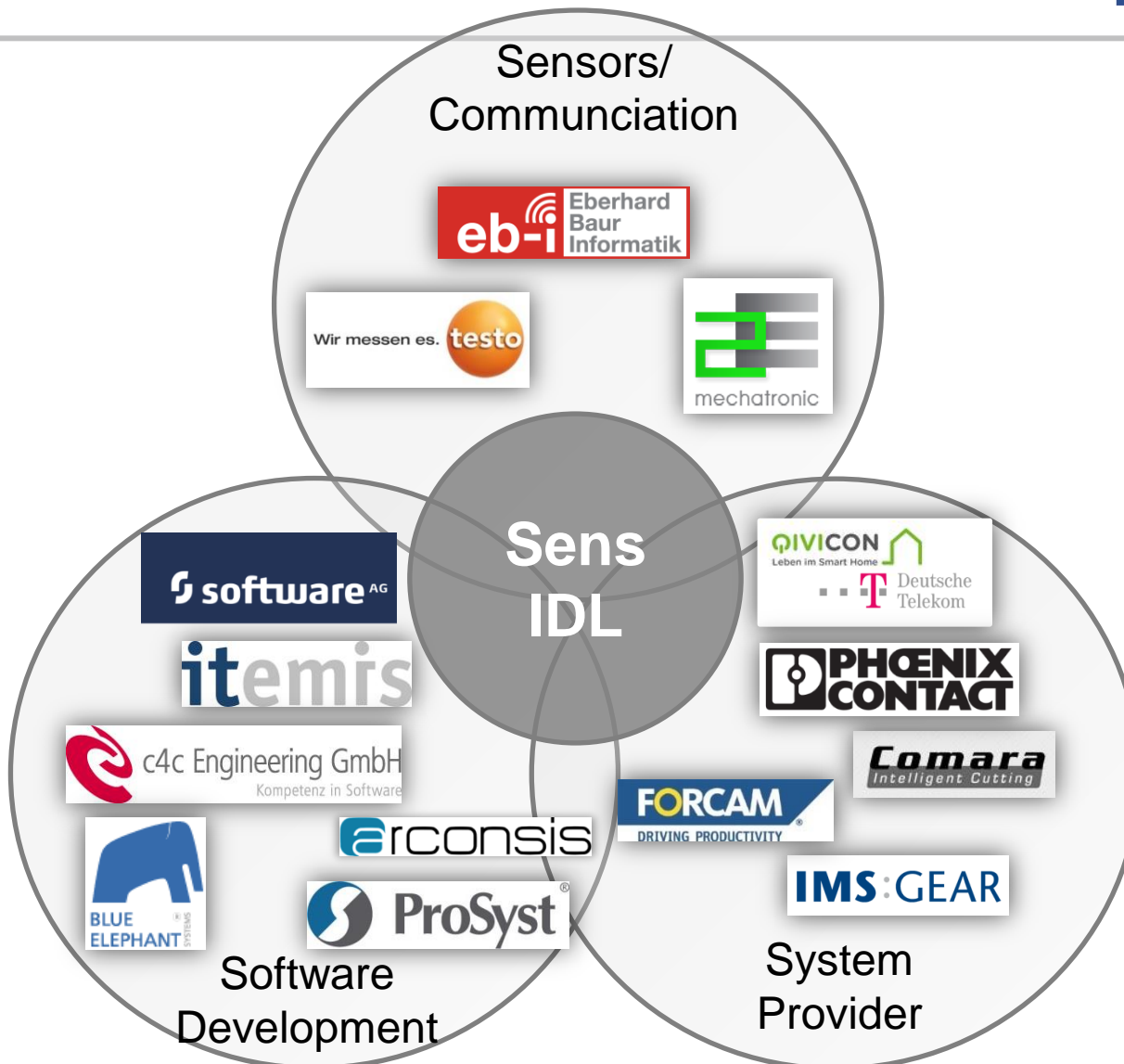
- Embedded Software
- Energy efficient sensor systems
- Wireless communication



Karlsruhe

Villingen-Schwenningen

Industrial Accompanying Committee



How we look like



SensIDL Tool

- Supporting the implementation of communication interfaces
- Based on model-driven rechniques

Current Status

- Collection of requirements
- Implementation recently started

Outlook

- Initial funding until end of 2016
- Further represantative demonstrators
- Extending the community

Questions?



<http://www.sensidl.de>



Dr.-Ing.
Christoph Rathfelder
R&D Sensors & Systems

HSG-IMIT
Wilhelm-Schickard-Str. 10
78052 Villingen-Schwenningen

christoph.rathfelder@hsg-imit.de
+49 7721 943-161

