



Fraunhofer Institute for Integrated  
Systems and Device Technology IISB



Stefan Waldhör – 2025-03-20

---

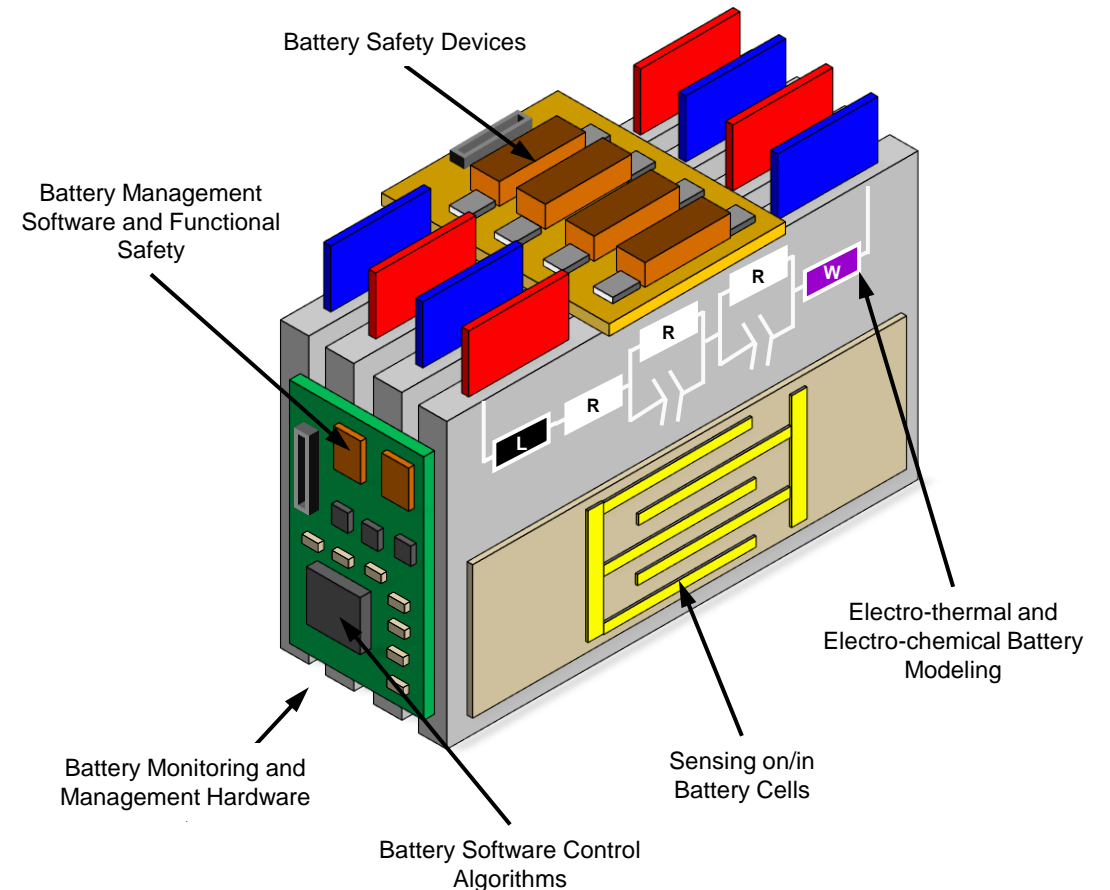
# Unleashing the Potential of Battery Data: Transforming Measurement Data into Strategic Insights for Battery and Battery System Optimization

# Introduction – Fraunhofer IISB

## Group Battery Systems: Competences and Services

### Research and Development Activities

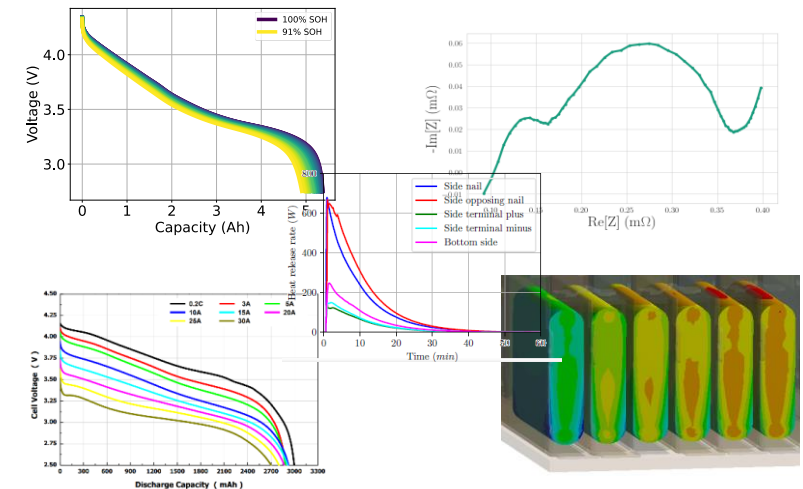
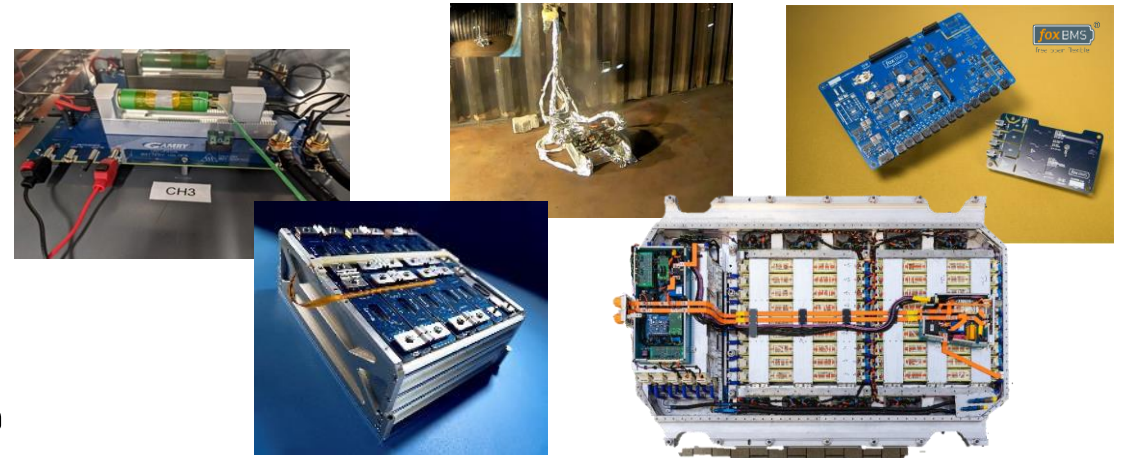
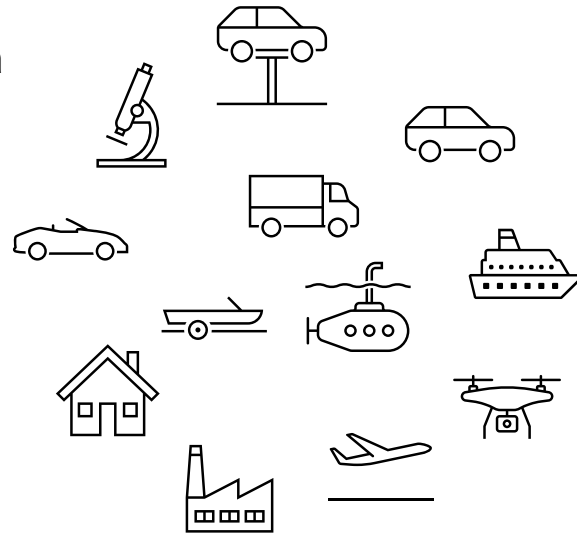
- **Design and Prototyping of custom battery systems** from requirements engineering up to the mechatronic integration
- **Battery Management Systems** (BMS Hardware and Software) with **high availability** concepts (**foxBMS®**)
- **Passive and active safety concepts** for high functional safety integrity levels in multiple domains (ASIL-D / SIL-3 / DAL-C)
- **Battery Junction Boxes** (BJB) and **Power Distribution Units** (PDU)
- **Thermal management** for non-propagation in case of **thermal runaway** and temperature homogenization concepts for **ultra-fast charging** (>3C-rate)
- **Modelling and simulation methods** for battery system design and predictive control of battery-based applications



# Collecting Battery Data

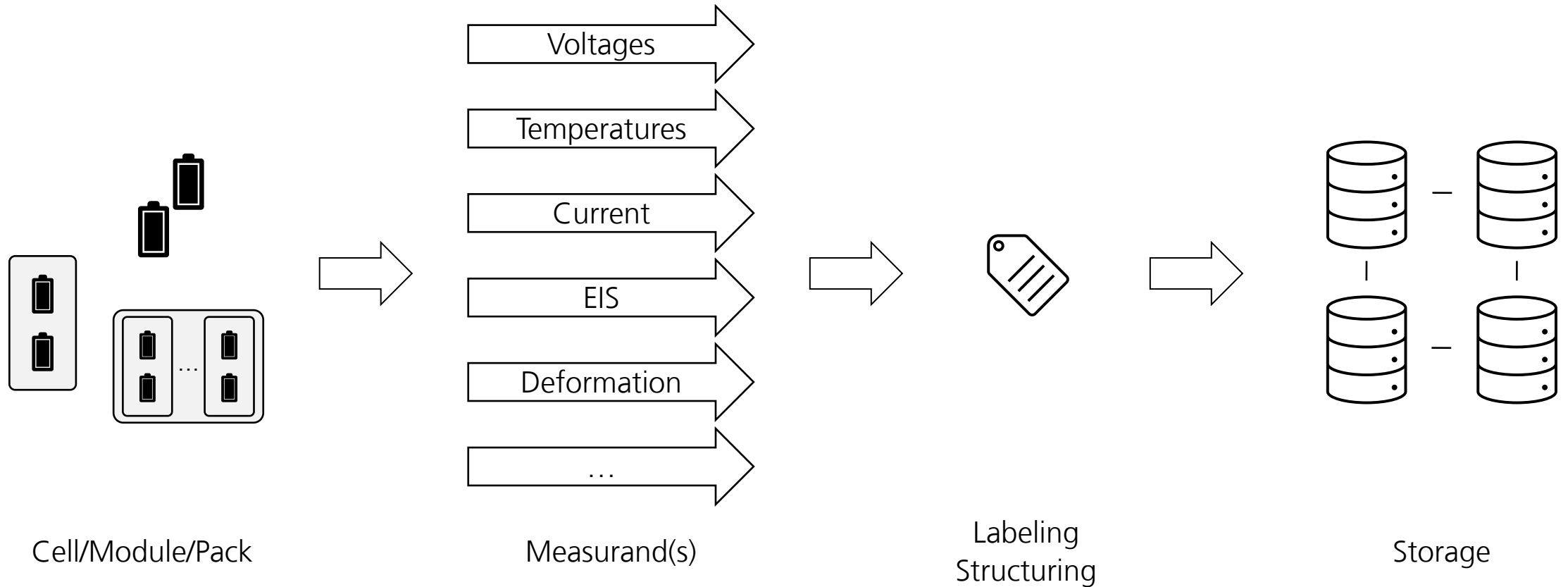
## Where and why?

- Battery data is collected under different conditions aspects and for different purposes
  - **Laboratory:** characterization, modelling during design phase
  - **Application:** Validation, service, lower data rate, product iteration/improvement



# Collecting Battery Data

## What?



# Exemplary Battery Usage Data

## foxBMS® Battery Data

- The BMS communicates e.g., via CAN with the higher-level control unit
- Transmitted battery data at least contains cell voltages, temperatures and system current
- CAN messages
  - contain a timestamp and a portion of the system data in hexadecimal representation
  - Use a semi-structured format
- Time intervals between different CAN message might be irregular

Timestamp (s)	ID	DLC	Data
1212239.036196	253	8	00 07 FF C0 06 00 2F FF
1212239.035950	252	8	00 07 FF C0 02 00 10 01
1212239.036442	254	8	00 07 FF BF FE 00 0F FF
1212239.036934	256	8	00 08 01 40 06 00 30 01
1212239.037182	257	8	00 08 00 C0 06 00 2F FF
1212239.037382	35D	6	01 0D 00 05 FF 55
1212239.037604	35E	6	02 0C 00 05 FF 01
1212239.037828	35F	6	03 00 00 05 FF 70
1212239.038055	528	6	07 01 FF FF E8 60
1212239.038497	526	6	05 0F FF FF FC 5F
1212239.040365	527	6	06 0B FF FF 26 6D
1212239.040977	223	8	80 1B FF 80 00 00 17 15
1212239.051044	240	8	00 08 00 40 02 00 10 01
1212239.051290	241	8	00 08 00 C0 02 00 30 01
1212239.051780	242	8	00 08 00 C0 02 00 50 00
1212239.052270	244	8	00 07 FF BF FE 00 10 00
1212239.052761	246	8	00 08 00 C0 0A 00 10 02
1212239.053007	247	8	00 08 01 C0 0A 00 50 03
1212239.053253	248	8	00 08 01 40 0A 00 30 02
1212239.053363	130	5	00 F0 FC FF FF

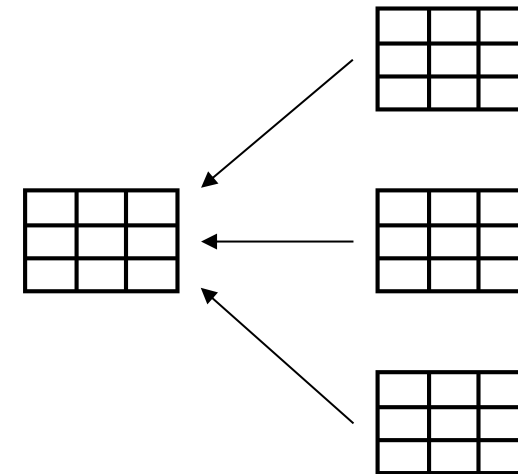
# Exemplary Battery Usage Data

## foxBMS® Battery Data

- Irrelevant CAN messages are filtered out.
- CAN messages are decoded and converted into JSON format.
- Decoded CAN messages are grouped together with respect to their CAN ID → Grouping creates multiple regular time series
- Synchronize measurement data

## Decoded CAN Messages

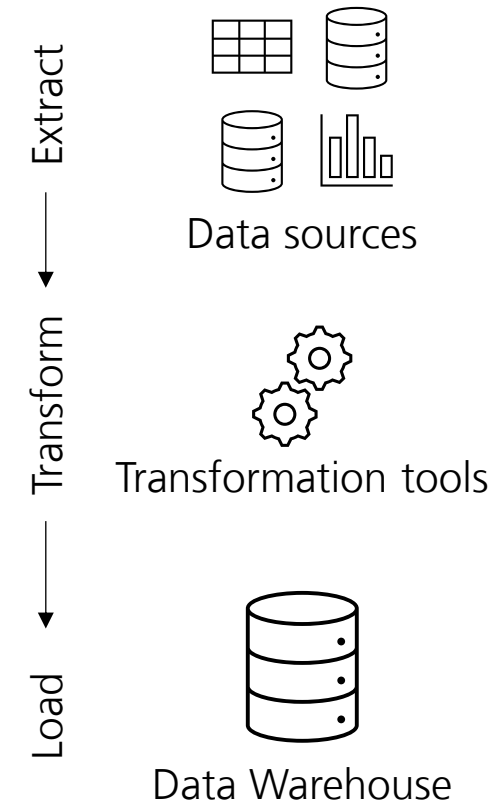
```
{"Timestamp": 1212239.057110, "860_CurrenSensor_SIG_Current_mA": -2309}  
{"Timestamp": 1212239.077085, "860_CurrenSensor_SIG_Current_mA": -2301}  
{"Timestamp": 1212239.097439, "860_CurrenSensor_SIG_Current_mA": -2327}  
{"Timestamp": 1212239.117353, "860_CurrenSensor_SIG_Current_mA": -2328}  
{"Timestamp": 1212239.137263, "860_CurrenSensor_SIG_Current_mA": -2352}  
{"Timestamp": 1212239.157239, "860_CurrenSensor_SIG_Current_mA": -2354}
```



# Battery Data

## Extract, Transform, Load (ETL)

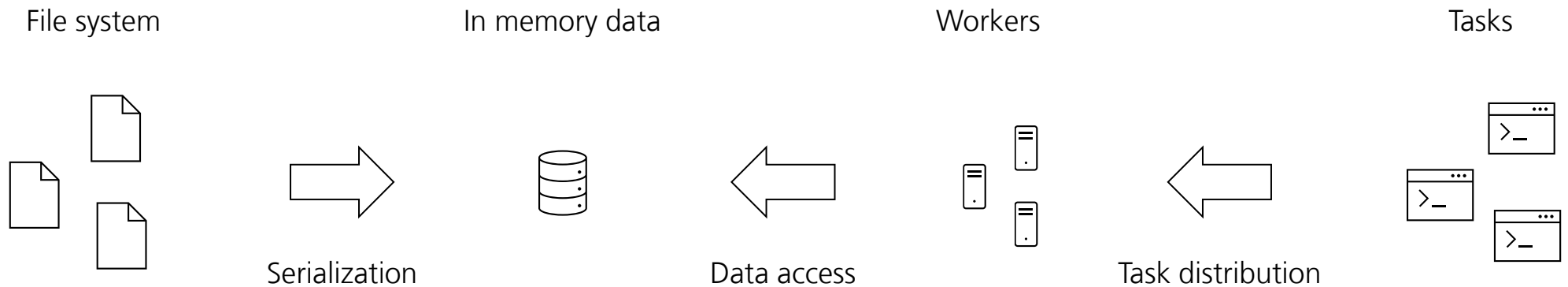
- ETL is a data engineering concept to automatically **preprocess data from various sources** and **upload this data** (e.g., to a database or a big data framework for later analysis)
- ETL engine for battery data of different sources will be part of foxBMS 2
  - support data sources providing data from CAN logs, MQTT or structured file formats (e.g., csv, parquet)
  - upload data to various databases, file formats and big data frameworks will be supported



# Battery Data

## Infrastructure Requirements

- Efficiently store large amount of structured data
- In memory processing should be possible
- Interfaces to different programming, data and visualization frameworks needed





# Battery Data

## Infrastructure Requirements

- Battery data analysis: most data queries effect many values of few columns and therefore a **column-oriented data structure** (data locality) is beneficial with respect to performance.
- Tools should
  - offer comprehensive support for **different programming languages**
  - provide support for different **databases and analysis tool** to enable the analysis of a **huge amount of data**

## Apache® Hadoop®

- Open-source framework providing a cluster-based file system (HDFS) and a cluster-based batch processing engine (Map Reduce).
- Process and analyze structure, semi-structured and unstructured data
- HDFS inspired the concepts of modern data lakes, a storage for structure, semi-structured and unstructured data
- Map Reduce inspired Apache Spark which tries to accelerate any data processing by prioritizing in memory operations



# Battery Data

## Infrastructure Requirements

- Battery data analysis: most data queries effect many values of few columns and therefore a **column-oriented data structure** (data locality) is beneficial with respect to performance.
- Tools should
  - offer comprehensive support for **different programming languages**
  - provide support for different **databases and analysis tool** to enable the analysis of a **huge amount of data**

## Apache® Arrow™

- Open-source language-agnostic software framework for developing data analytics applications that process columnar data
- Support for multiple
  - file formats (e.g., csv, parquet)
  - databases (e.g., InfluxDB®, DuckDB™) via Arrow Flight and,
  - big data frameworks (e.g., Apache® Spark™)



# Battery Data

## Infrastructure Requirements

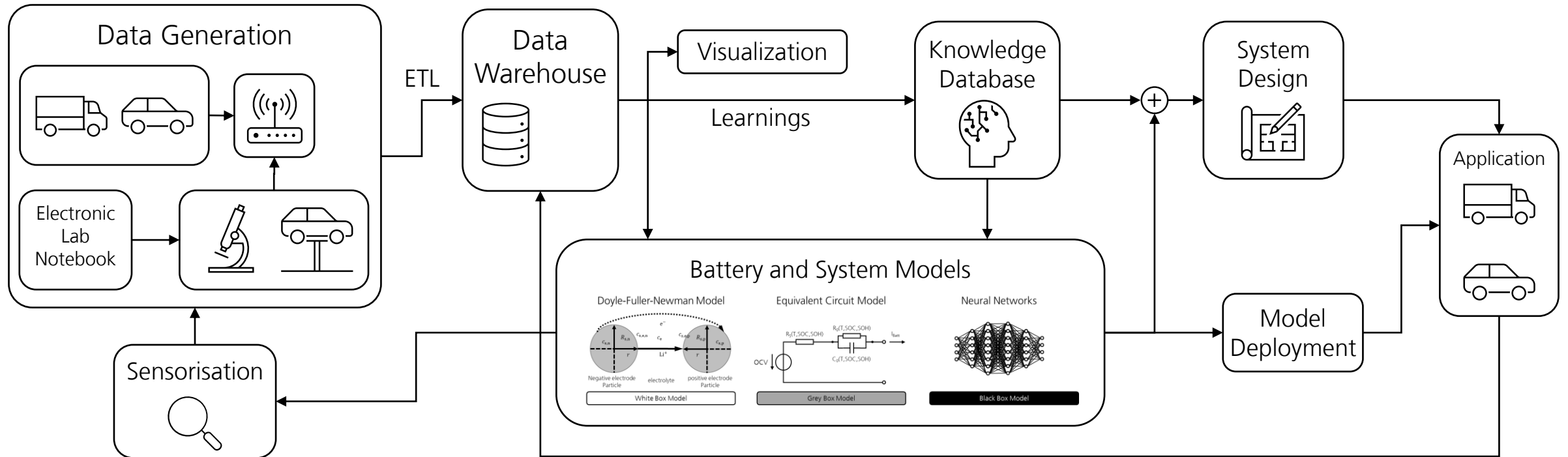
- Battery data analysis: most data queries effect many values of few columns and therefore a **column-oriented data structure** (data locality) is beneficial with respect to performance.
- Tools should
  - offer comprehensive support for **different programming languages**
  - provide support for different **databases and analysis tool** to enable the analysis of a **huge amount of data**

## Apache® Spark™

- Open-source multi-language engine for executing data engineering, data science, and machine learning on single-node machines or clusters.
- Reads files from the underlying HDFS and creates an in-memory data structure
- A master node distributes tasks between the worker nodes



# Data and Insights: What do we know now and where to go?



# Contact

---

**Stefan Waldhör, M.Sc.**  
**Battery Systems**  
**[stefan.waldhoer@iisb.fraunhofer.de](mailto:stefan.waldhoer@iisb.fraunhofer.de)**

[www.iisb.fraunhofer.de](http://www.iisb.fraunhofer.de)

