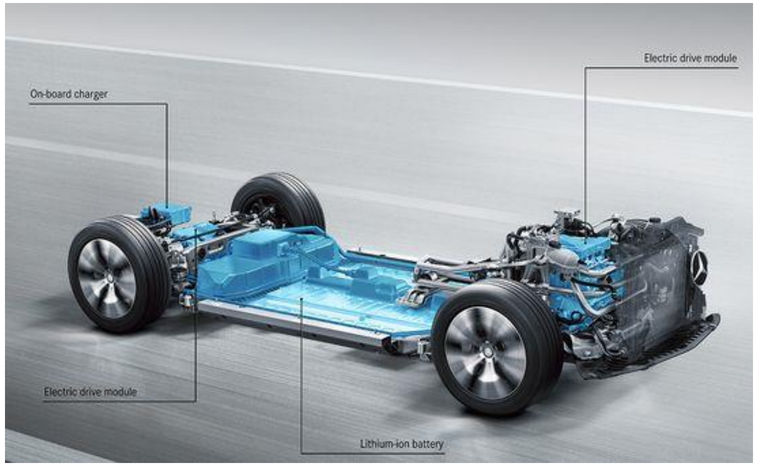
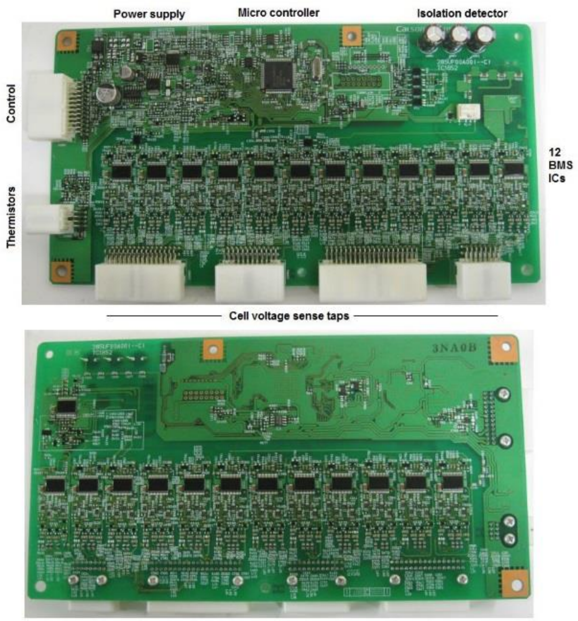
From the vehicle to the BMS: systems engineering approach





ADAMA BA

18/09/2018

Table of contents

Table of contents

Introduction: why systems engineering?

Electric vehicles are a ***disruptive innovation*** ➔ they can't be developped the same way as ICE vehicles

Electric vehicles are very complex systems, made of very complex subsystems

Other innovations are shaping the automotive industry alongside electrification – self driving and connected cars – which make the vehicle even more complex

A new approach is required to develop these vehicles, which is systems engineering

Definition of systems engineering (INCOSE*)

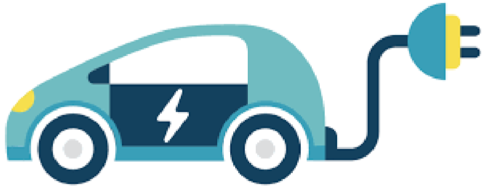
- "Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem: Operations, Performance, Test, Manufacturing, Cost & Schedule, Training & Support, Disposal."

*International Council on Systems Engineering

Different applications, different requirements, different lifetimes

Stationary energy storage
10 to 15 years

Medical devices
5 to 20+ years



BEV & PHEV
8 to 10 years

Consumer electronics
2 to 5 years

Industrial automation
2 to 5 years

Spacecraft
15+ years

NO "ONE SIZE FITS ALL" BMS !!!

Automotive application

Vehicle requirements are driven by (not exhaustive):

- Mechanical power (Vmax, acceleration, braking behaviour…)
- Range
- Charge time
- Durability
- Safety
- Internal standards for assembly, service…

BMS requirements will support requirements through requirements on (not exhaustive):

- Power estimation
- Battery state estimations
- Cell voltage monitoring
- Temperature monitoring
- Power derating
- Redundant measurement channels

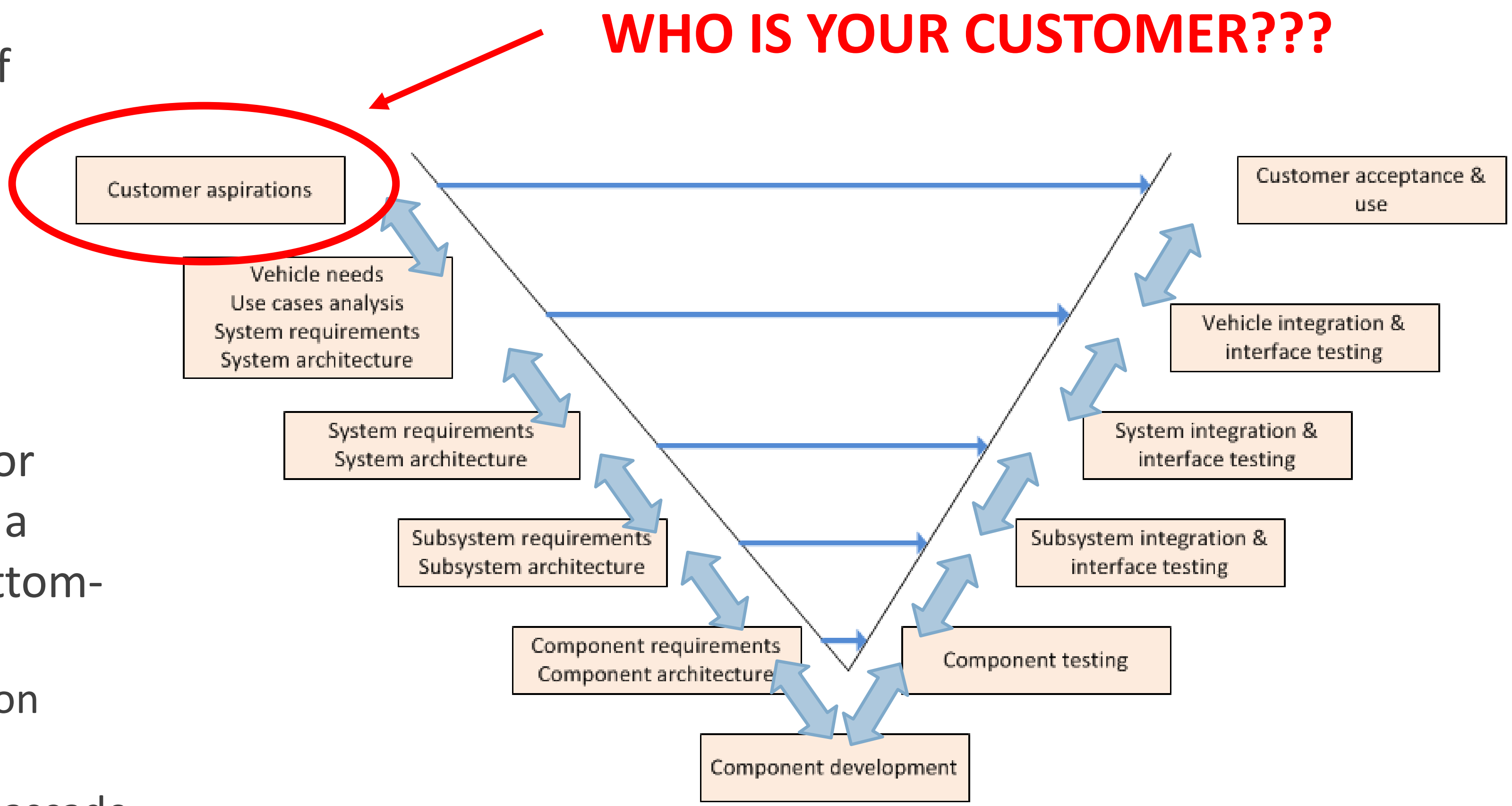
Table of contents

BMS Design

A **proper** systems engineering approach can significantly reduce the cost of development and validation of any complex systems.

This requires a rigorous follow-up of

- System definition
- Subsystems decomposition
- Interfaces definition
- Requirements definition



All these steps shall be performed for every layer of the system, following a global top-down approach, with bottom-up activities only used to:

- Refine the system definition based on subsystems constraints
- Arbitrate subsystem requirements cascade

System definition

The very first step in system design, it consists basically in giving the « raison d'être » of a system, i.e. what does it do, what it is acting on, and for what purpose

There is no *standard definition* for a BMS, due to the numerous applications of lithium-ion batteries ➔ scope may vary, depending on the project

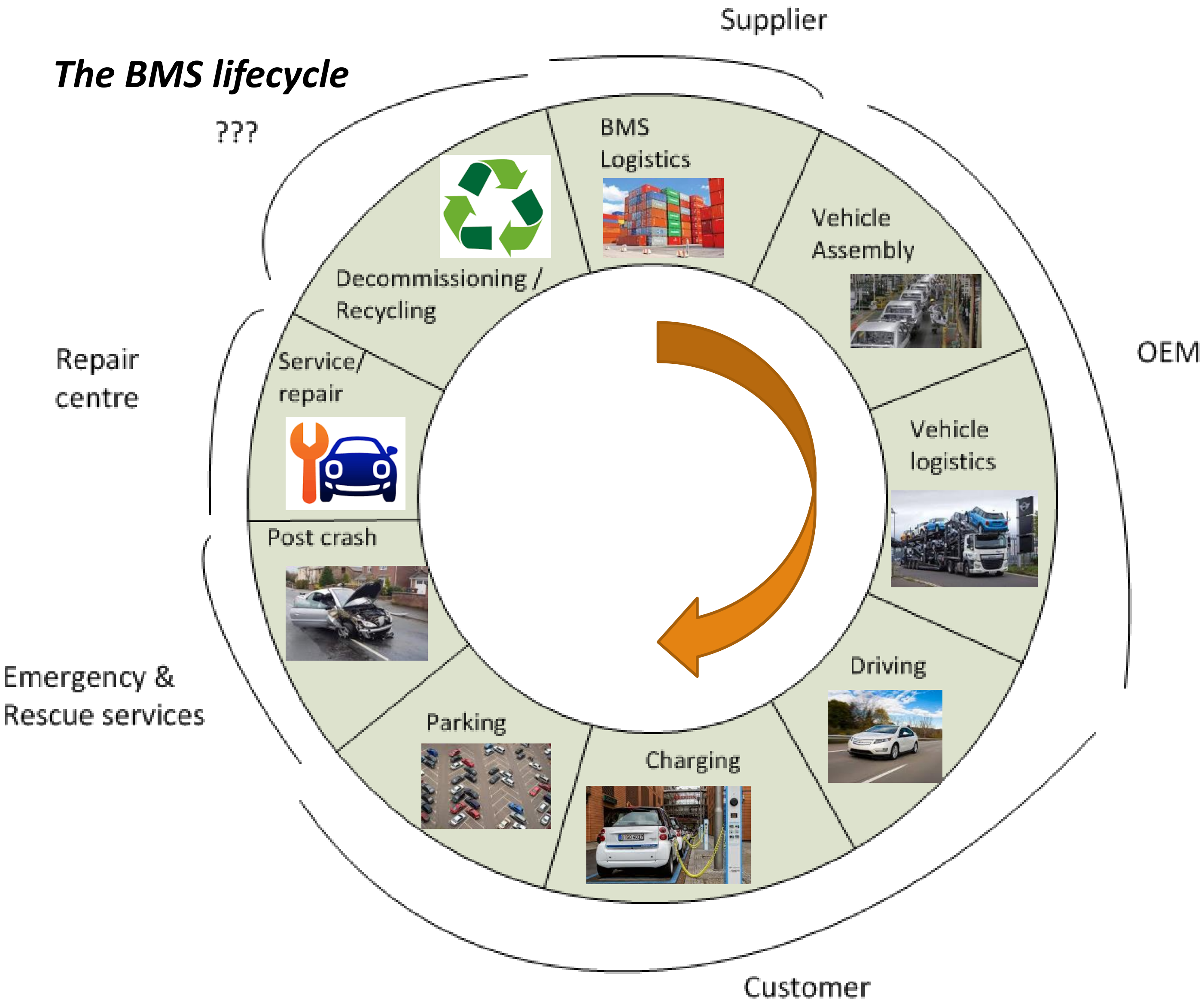
However, a tacitly accepted definition of a BMS is:

- A battery management system is an electronic regulator whose function is to keep the battery within its operative range during charge and discharge by monitoring parameters such as temperature, voltage, and current

A more accurate definition of a BMS shall be based on

- Battery system requirements
- Battery lifecycle
- List of use cases

System definition



Many different end users, but a single responsible for the requirements from vehicle assembly to decommissioning: the OEM

Usually, end of life requirements are poorly specified by OEM. They could yet decrease the cost of recycling if implemented properly (ex: quick screening algorithm of surviving cells in a battery pack)

The impact of external events in BMS development

EXTREMETECH

Search Extremetech

Computing Phones Cars Gaming Science Extreme Deep

HOME › EXTREME › NISSAN IS RECYCLING OLD ELECTRIC VEHICLE BATTERIES TO POWER STREET LIGHTS

Nissan Is Recycling Old Electric Vehicle Batteries to Power Street Lights

By Ryan Whitwam on March 26, 2018 at 1:01 pm | 11 Comments

REUTERS World

ENVIRONMENT

China puts responsibility for battery recycling on makers of electric vehicles

Reuters Staff 2 MIN READ

SHANGHAI (Reuters) - China will make manufacturers of electric vehicles (EV) responsible for setting up facilities to collect and recycle spent batteries, as part of its efforts to tackle mounting waste in the sector, say new rules published on Monday.

BUSINESS INSIDER UK MORE

Chinese carmaker BYD close to completing battery recycling plant

Reuters
Mar. 21, 2018, 11:49 AM 7

Menu Search Bloomberg Businessweek

June 28, 2018, 8:04 AM GMT+1

Electric-Car Batteries Find Second Life

- Even after coming off the road they can be reused and then recycled for raw materials

By David Stringer and Jie Ma

IET sites Welcome Account

E&T ENGINEERING AND TECHNOLOGY Menu

Japan's first electric car battery recycling plant to sell old batteries at half price

By Jack Loughran

Published Tuesday, March 27, 2018

Japan's first plant specialising in the reuse and recycling of lithium-ion batteries from electric vehicles (EVs) is set to open amid growing demand for electric cars.

Subsystem decomposition

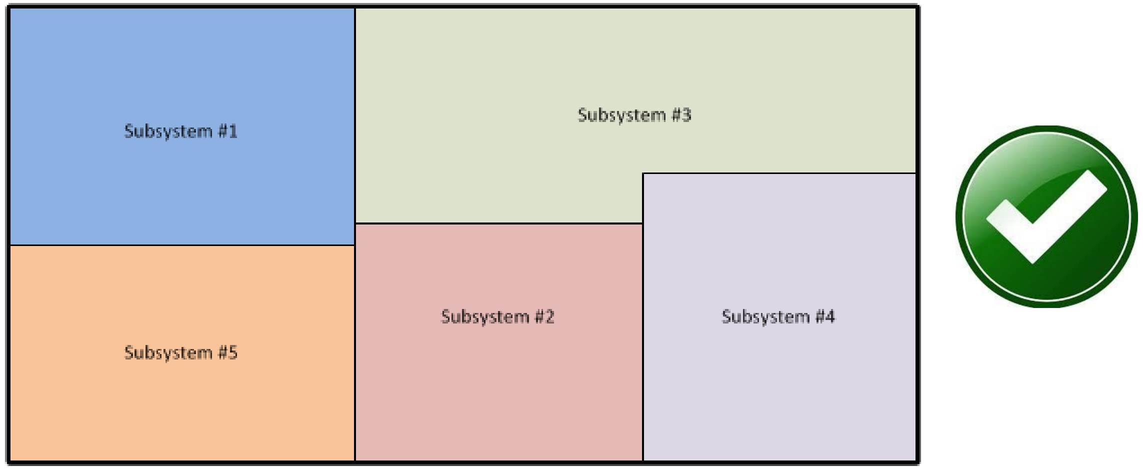
A decomposition of the system into different subsystems makes sense if the system is is **too complex**

Such decomposition shall follow some rules:

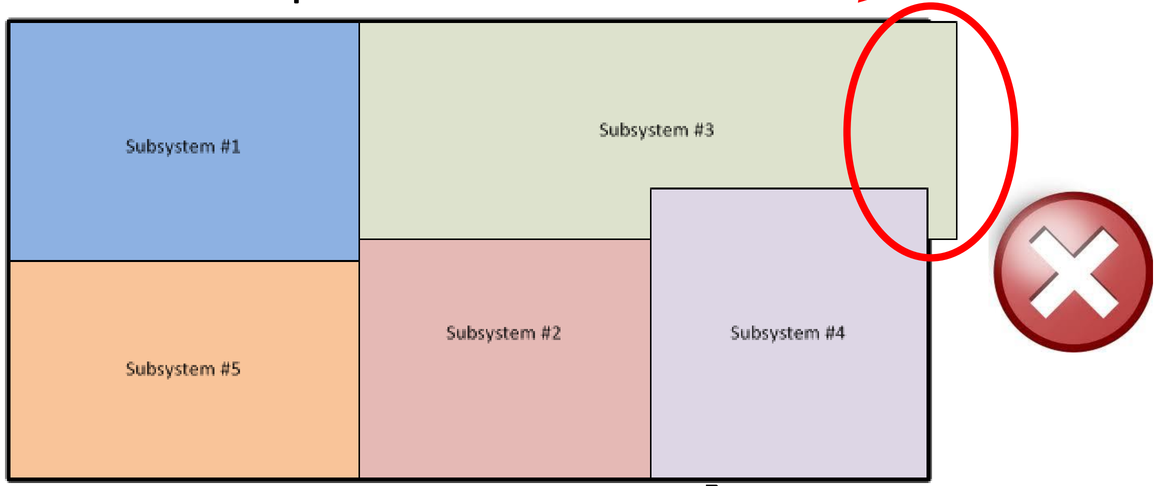
- The scope of all subsystems remains within the scope of the system
- The system is fully covered by the subsystems
- Every subsystem has its function(s) identified, that contribute to the function fulfilled by the system
- A single function is not hosted by two or more subsystems. However, a single function may be split between different subsystems

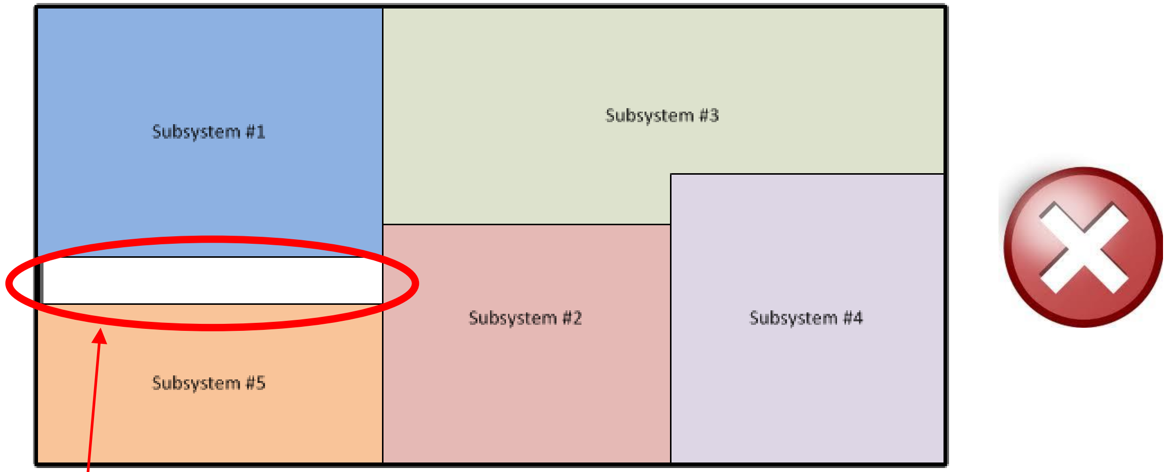
Subsystem decomposition

The most common mistakes in system decomposition



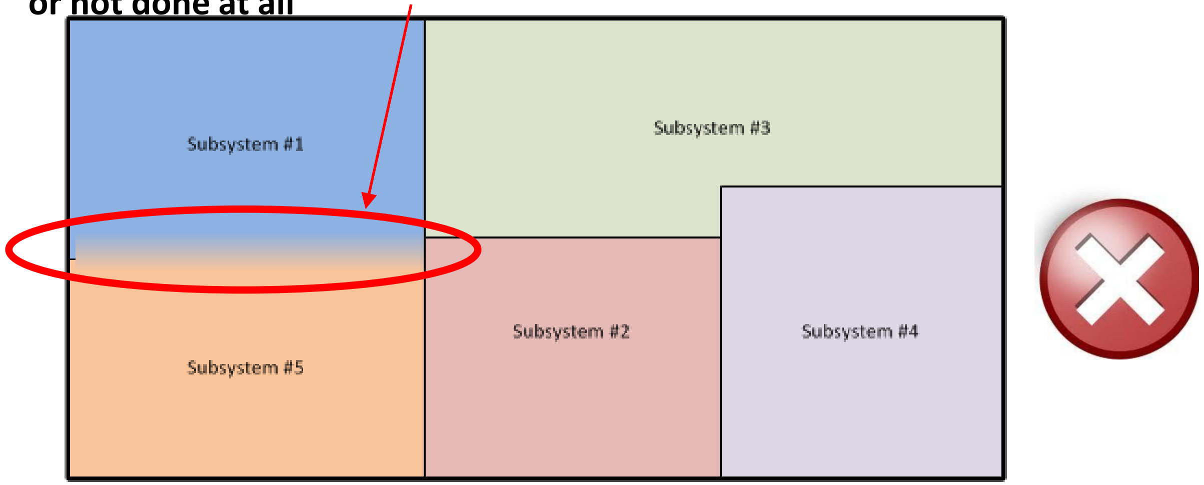
Subsystem scope beyond system scope





No responsible for an area within system scope

Unclear boundaries between two subsystems ➔ work done redundantly or not done at all

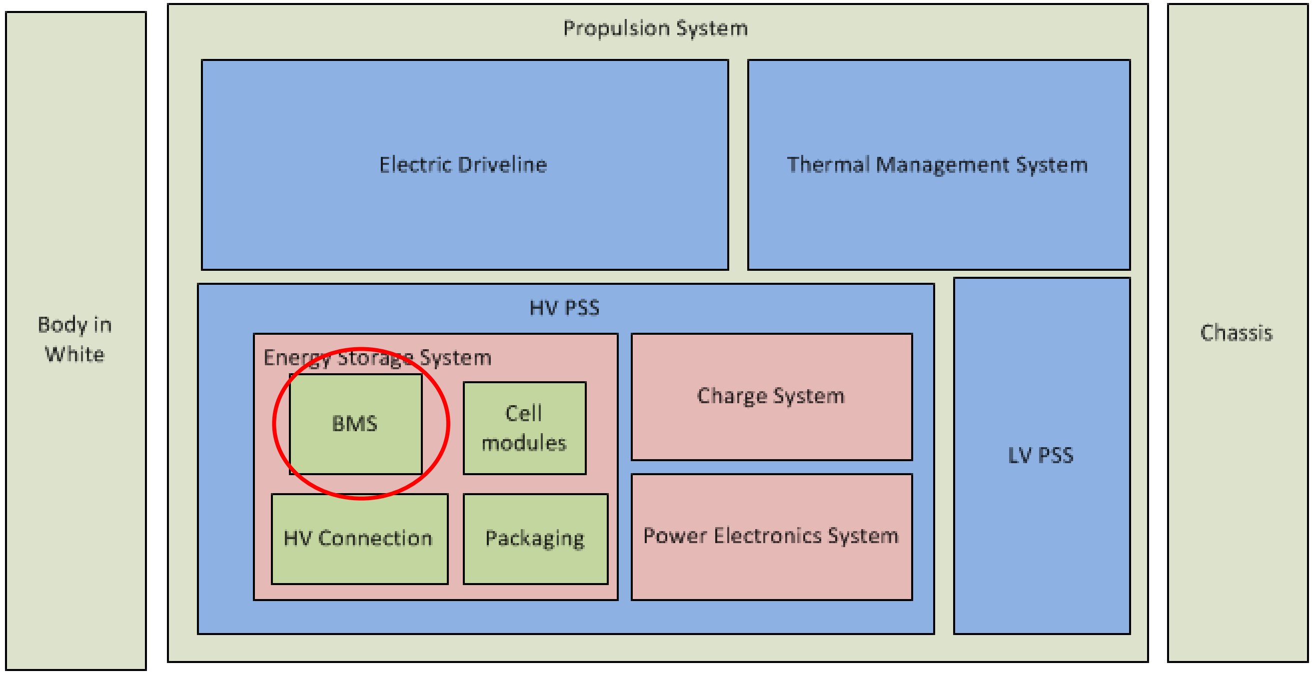


Subsystem decomposition

- A BMS is a system, but also a subsystem, which is part of an energy storage system
- The energy storage system itself is a subsystem, if compared to the high voltage power supply system (HVPSS)
- HVPSS is itself another subsystem, part of the propulsion subsystem

To avoid risks discussed in previous slide, none of these subsystems shall be defined **without support from their system owner**

Example of a decomposition



There are many system decomposition possible, but only one should be agreed at the beginning of development!!!

Interfaces definition

Interfaces describe how the different subsystems and/or components are connected to each other

Interfaces can be of different kind:

- Electrical (defined by electrical power, voltage, current…)
- Mechanical ((number of mounting points, fastening torque, stiffness, waterproofness, package integration …)
- Thermal (thermal power, flow rate…)
- Information (messages exchanged on CAN, LIN, Flexray…)

A best practice is to separate internal interfaces (those within the scope of study), and external interfaces (those which are going to items outside the system)

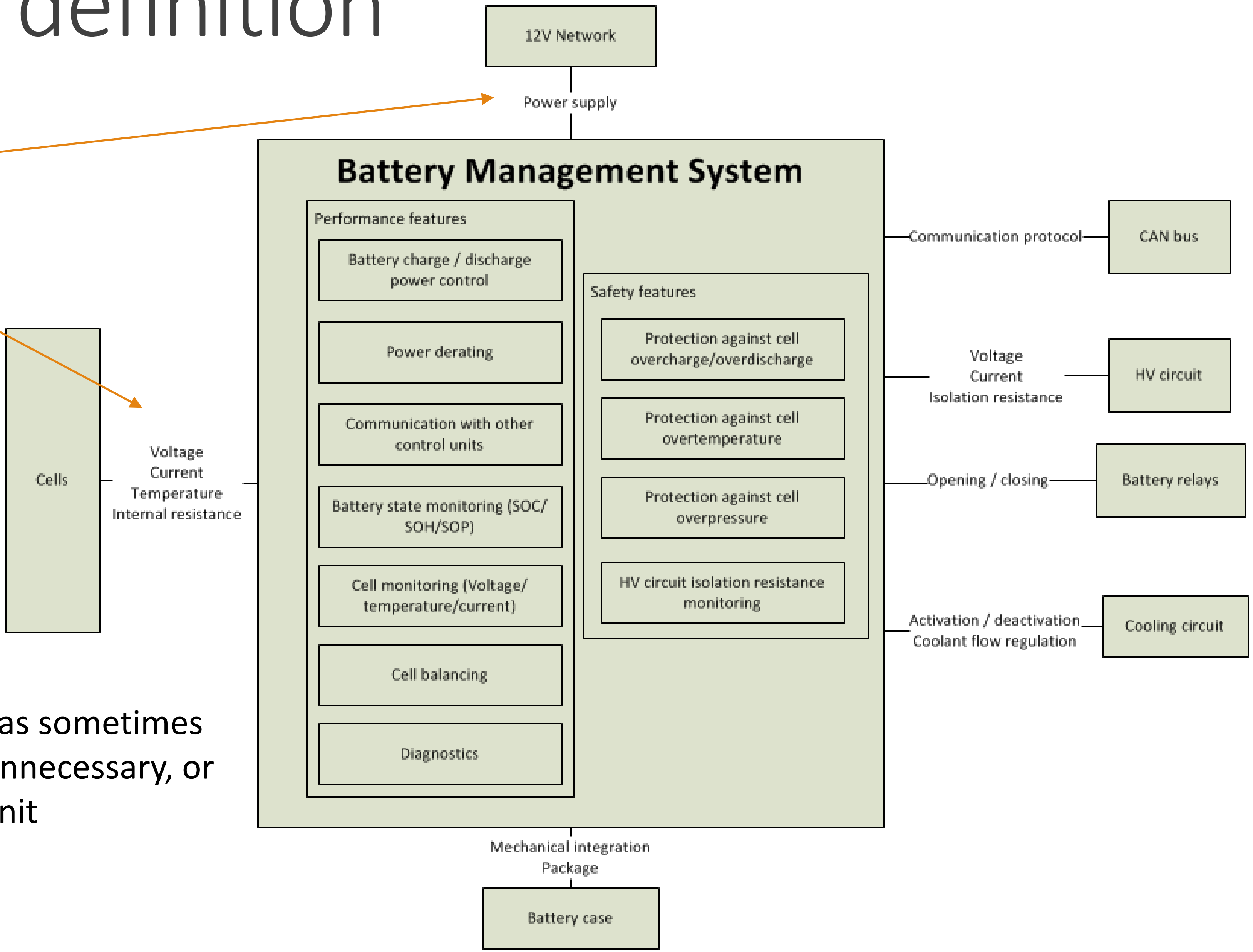
Interfaces well defined :

➔ better cross-compatibility of the BMS ➔ reduced time and cost of development
➔ open the competition to different BMS supplier

Interfaces definition

BMS Interfaces example

Interfaces

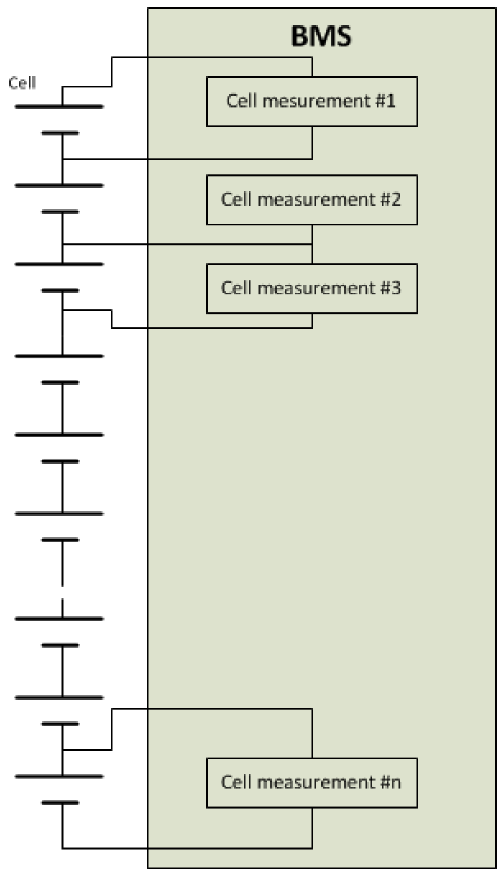


This is just an example, as sometimes cooling circuit may be unnecessary, or controlled by another unit

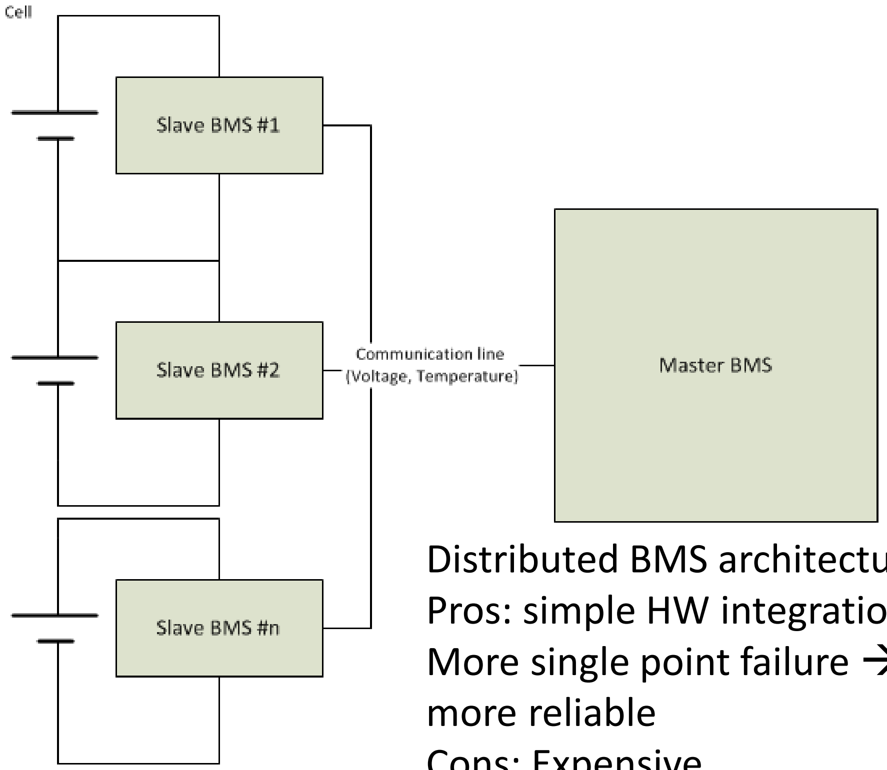
Interfaces definition

Interfaces can lead to rethink about the architecture, especially if there is a way to reduce them

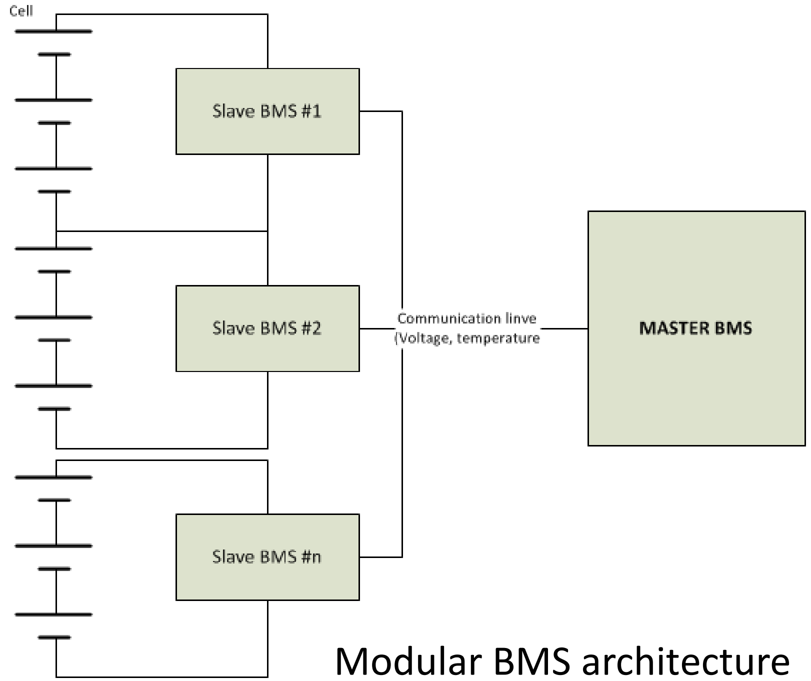
Example:



Centralized BMS architecture
Pros: cost effective
Cons: Wiring challenges



Distributed BMS architecture
Pros: simple HW integration
More single point failure → more reliable
Cons: Expensive



Modular BMS architecture
A mix of centralized and distributed architectures

Requirements definition

Requirements are tasks or conditions that should be met to achieve a function.

Contrary to a function, a requirement is defined by engineering measurables, targets, and verification methods

In an ideal world, BMS requirements should all stem from parent requirements written from either higher level systems (e.g. the energy storage system), or equivalent level systems → no bottom-up requirement

A **good requirement** describes something that is

- Necessary
- Verifiable (with acceptance criteria, by design review or test)
- Attainable (must be feasible and fit within budget, schedule, and other constraints such as weight…)
- Clear (no ambiguity)

In case of doubt about a requirement:

What worth thing could possibly happen if the requirement is not respected?

If no relevant answer

Requirements definition

Functions decomposition performed at the beginning of the study help defining the requirements, as any subsystem requirement is supposed to support a system function

- Example

Function 1	**Keep the battery within temperature operating range**
Function 1-1	Monitor cell temperature
Requirement 1-1-1	The system shall measure the temperature of the cells within a range from -40 to 60
Requirement 1-1-2	The measured cell temperatures shall be digitized
Requirement 1-1-3	The accuracy of the digitalization shall be 16bits
Requirement 1-1-4	The sampling rate of the measurement shall be 1Hz
Function 1-2	Reduce battery available power when Temperature $\geq T_1$
Function 1-3	Dissipate thermal losses through the cooling system

Requirements definition

A **best practice** for BMS requirements is to sort requirements by priority

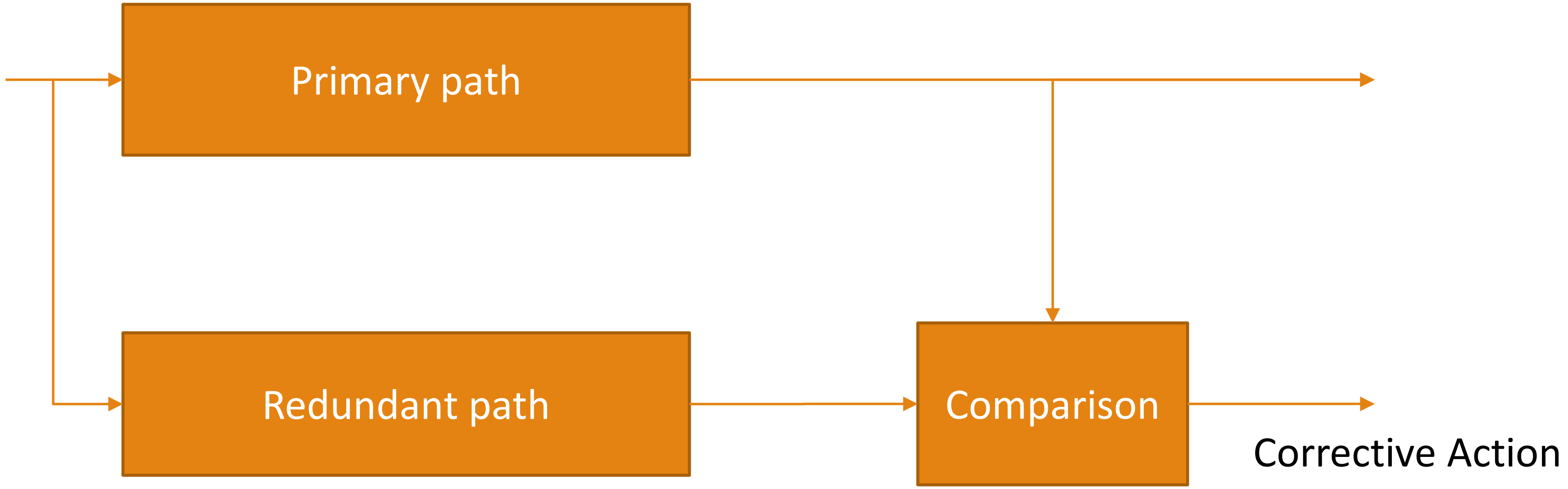
Example:

- 1 – Safety (from FMEA or ASIL decomposition) & regulations requirements
- 2 – Performance requirements
- 3 – Cost requirements (e.g. manufacturing, service, compliance with internal standards…)

Sometimes, a requirement can be met through different implementations. In that case, select the best requirement based on criteria such as cost, safety, reliability, customer usage, service…

Requirements definition

Impact of safety requirement on system architecture: example of an ISO26262 requirement



ASIL C and D safety goals generally require a redundant path to achieve a function, which significantly impact an architecture

Requirements definition

Impact of ISO2626 on system design

	Hardware qualification	Software qualification
ASIL D	Maintainability highly recommended Deductive analysis highly recommended Single point fault metric ≥ 99% Latent fault metric ≥ 90% Random HW failure target < 10^{-8} h^{-1} Statistical test highly recommended	Independent parallel redundancy highly recommended Prototype generation highly recommended Formal verification Fault injection test highly recommended Resource usage test highly recommended Error detecting code highly recommended
ASIL C	Maintainability highly recommended Deductive analysis highly recommended Single point fault metric ≥ 97% Latent fault metric ≥ 80% Random HW failure target < 10^{-7} h^{-1} Statistical test recommended	Independent parallel redundancy recommended Prototype generation recommended Formal verification Fault injection test highly recommended Resource usage test recommended Error detecting code recommended
ASIL B	Maintainability recommended Deductive analysis recommended Single point fault metric ≥ 90% Latent fault metric ≥ 60% Random HW failure target < 10^{-7} h^{-1} No statistical test required	No redundancy required No prototype generation required No formal verification Fault injection test recommended Resource usage test recommended Error detecting code recommended

Requirements definition

The most common mistakes in requirements definition

- Wrong assumptions because of a lack of information
- Poor implementation (e.g. requirements describing HOW a function shall be implemented instead of WHY it is needed)
- Unverifiable requirements (ex: terms such as minimize, maximize, sufficient, user-friendly…)
- Missing requirements
- Over-design (ex: nice to have requirements)
- Requirements not driven from top to bottom

BMS design considerations

Battery chemistry
- Energy vs power
- Format
- Safety characteristics

Cell architecture
- Series vs parallel

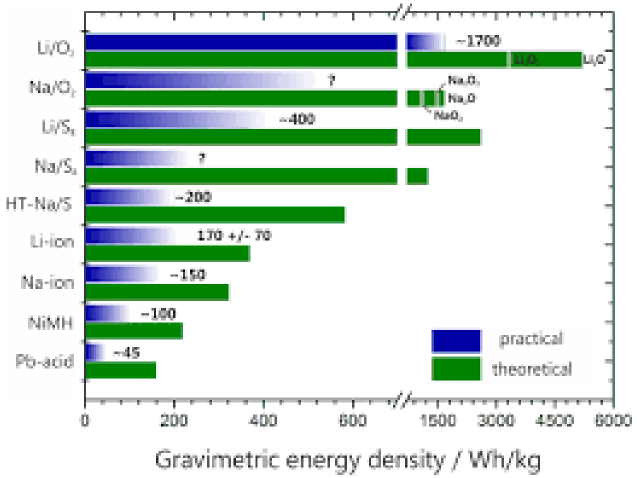
BMS architecture
- Communication architecture
- Centralized vs distributed vs modular

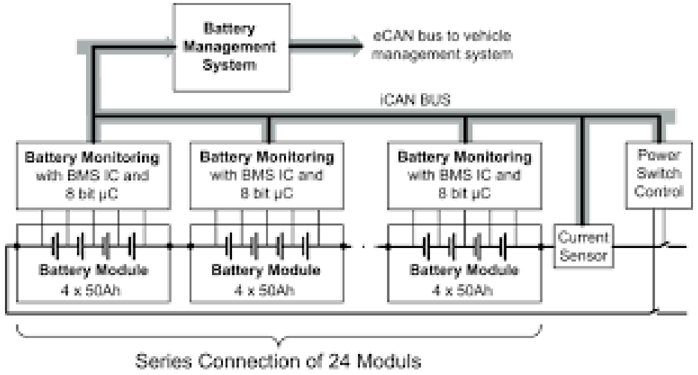
Fault robustness

Fault mitigation technique

Cost vs complexity vs performance

Cross compatibility



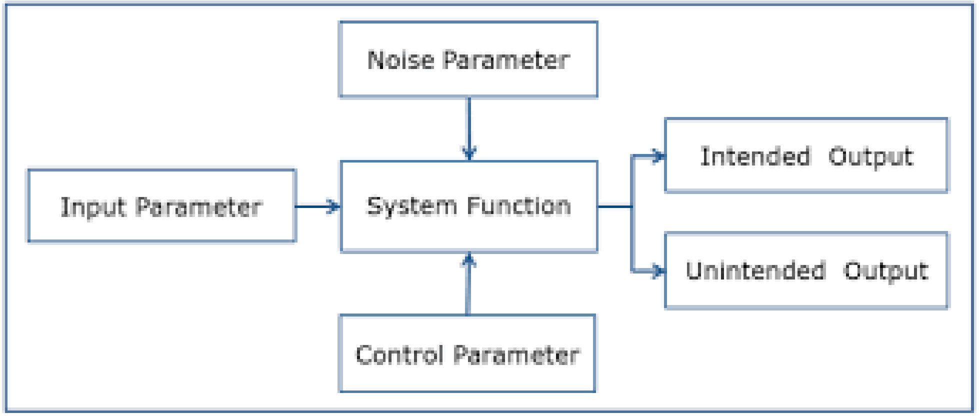


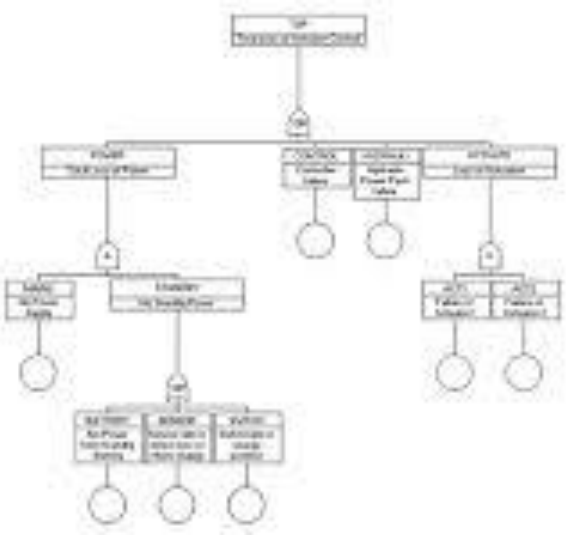
Design verifications

Analysis
- Parameter diagram
- Failure Mode and Effects Analysis (FMEA)
- Fault Tree analysis
- CAD analysis
- Electrical worst case scenario analysis

Test
- Electrical performance
- Mechanical performance
- Environmental performance
- Safety

Inspection
- Parts
- Processes





Lessons learned

Any system developed and **documented accordingly** is a base for future development of similar technologies

Lessons learned and properly captured can accelerate a system design, depending on how the system is carried over. It helps also to avoid design mistakes

However, lessons learned can sometimes constrain a design, as well as existing standards:

- Some functions and/or requirements may be relevant for one system, and irrelevant for another one
- Risk of requirements driven from a bottom-up approach, rather than a top-down one

Table of contents

How future BEV will shape future BMS ?

More battery recycling ➔ more individual cell screening ➔ more distributed architecture

Smart batteries (batteries with an interface to display SOC and/or SOH)

New communication interface to include additional external loads

Level 5 autonomous cars → more redundancies to be expected

Systems engineering could actually become a state-of-the-art !!

External regulations or standards becoming de facto states-of-the-art

Table of contents

Definition

System: a group of components forming an integrated whole, delineated by spatial and/or temporal boundaries, surrounded and influenced by its environment, described by its structure and purpose and expressed in its functioning

Subsystem: A subsystem is a system itself, and a limited component of a larger system. In general, a subsystem function defines a part of the function of a system

Function: a task, action or process that a system shall perform

Requirement: tasks or conditions that should be met to achieve a function. Contrary to a function, a requirement is defined by engineering measurables, targets, and verification methods

Functional requirement: any requirement which specify what the system should do

Non functional requirement: any requirement which specify how the system should behave, and what are the limits on its functionality

List of battery hazards

Battery system hazards			BMS hazards
Electrochemical hazards	Electric hazards	Mechanical hazards	Battery cooling fault
Emission of toxic or flamable gas	Short circuits	Crash	Battery heating fault
Emission of toxic or flamable liquid	Loss of electrical continuity	Liquid ingress	Communication fault
Electrolyte leakage	Impedance increase	Operator fault	Short circuit
Cell fire		Road bump / road debris	Overcharge due to BMS fault
Thermal runaway			Overtemperature due to BMS fault
			Overdischarge due to BMS fault