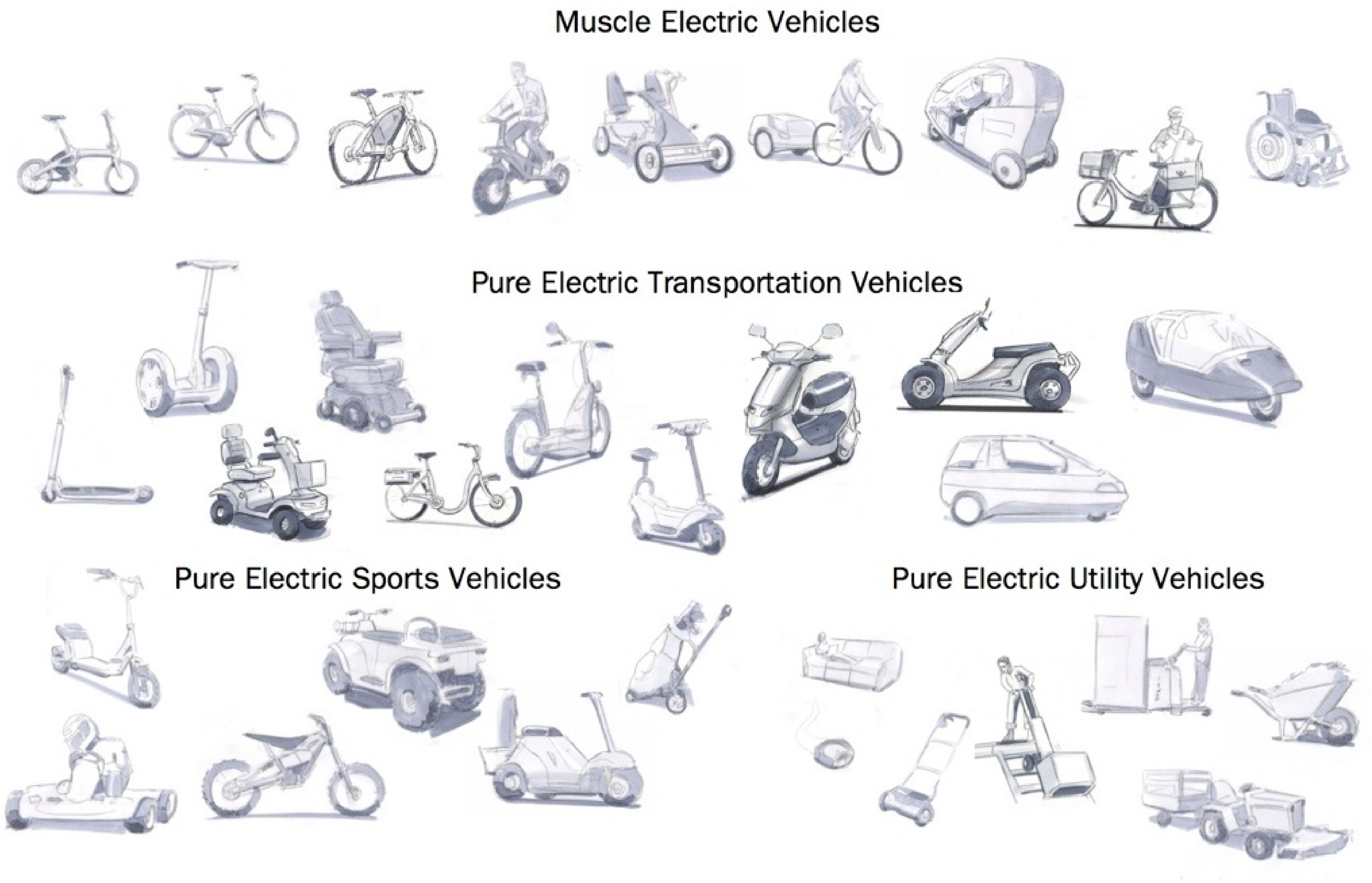
A Model-based Certification Framework for the EnergyBus Standard

Alexander Graf-Brill, Saarland University
Holger Hermanns, Saarland University
Hubert Garavel, Inria Grenoble

Light Electric Vehicles

LEV Product Groups and Markets


Muscle Electric Vehicles
Pure Electric Transportation Vehicles
Pure Electric Sports Vehicles
Pure Electric Utility Vehicles

Light Electric Vehicles

- rapidly growing market
- big OEMs and suppliers

BOSCH

Panasonic

- fleet administrators

DB

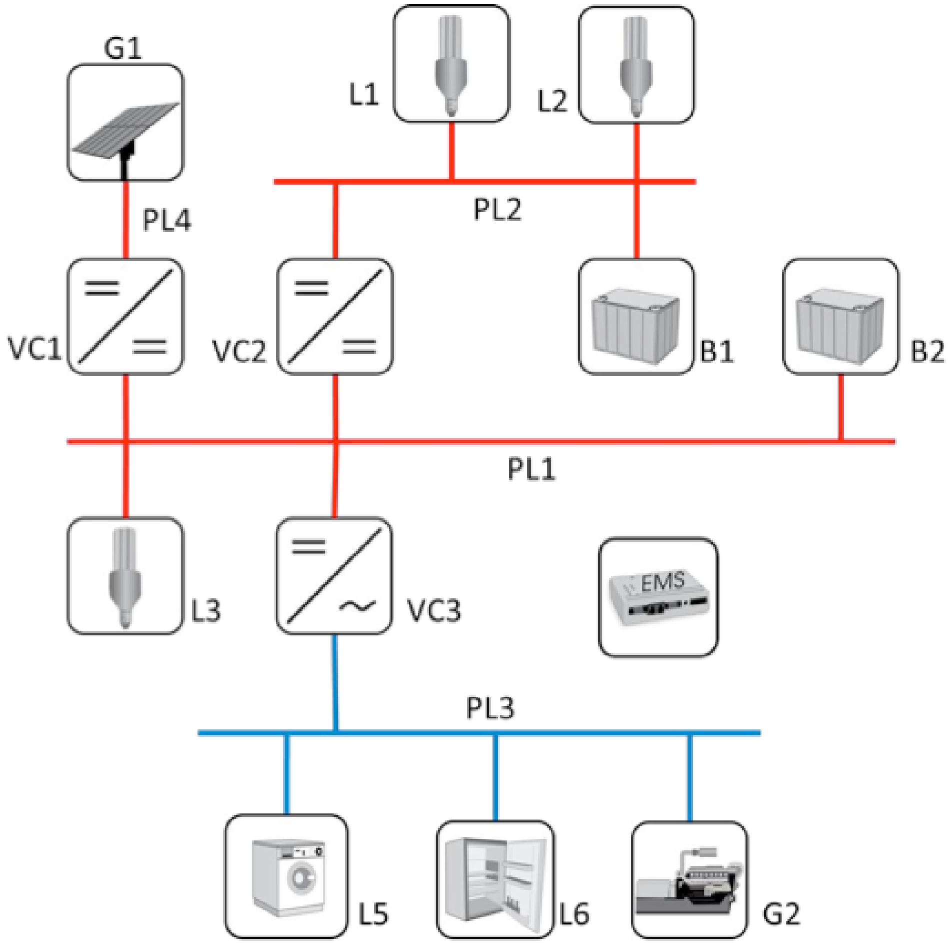
- bike vendors entering this market

Light Electric Vehicles

- need for Energy Management Systems
- proprietary solutions
  - incompatible connectors and platforms
  - no exchange of components

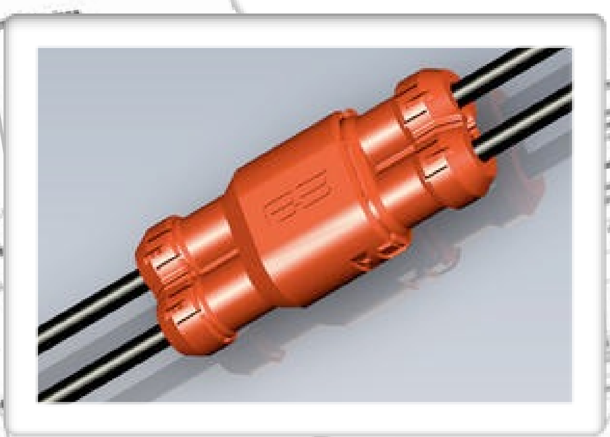
- safety critical application
  - distribution of electrical power
  - battery explosion

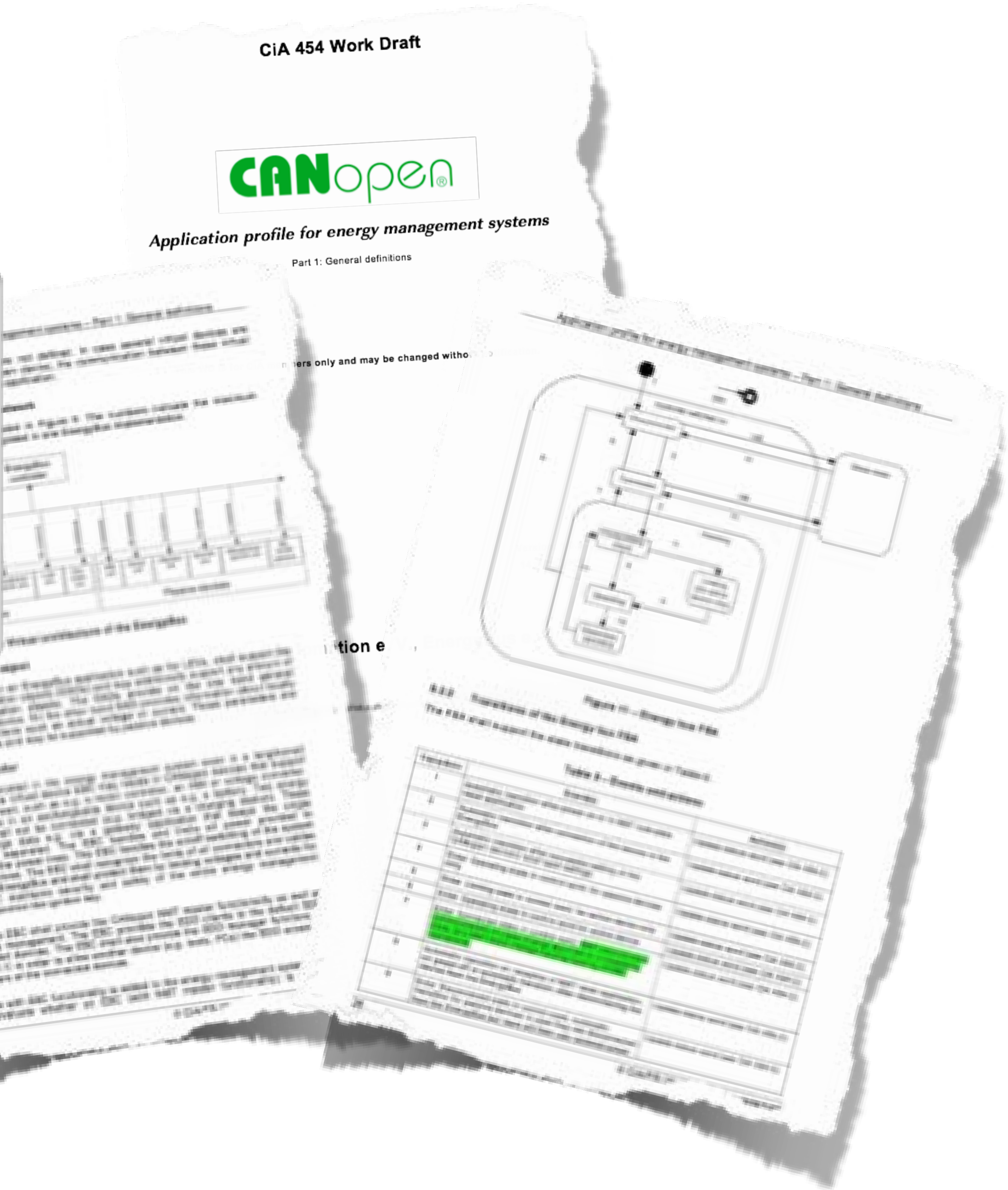
Stationary Applications



EnergyBus e.V.

- a universal standard for Energy Management Systems
- implementation of official certification procedure





CiA 454 Work Draft

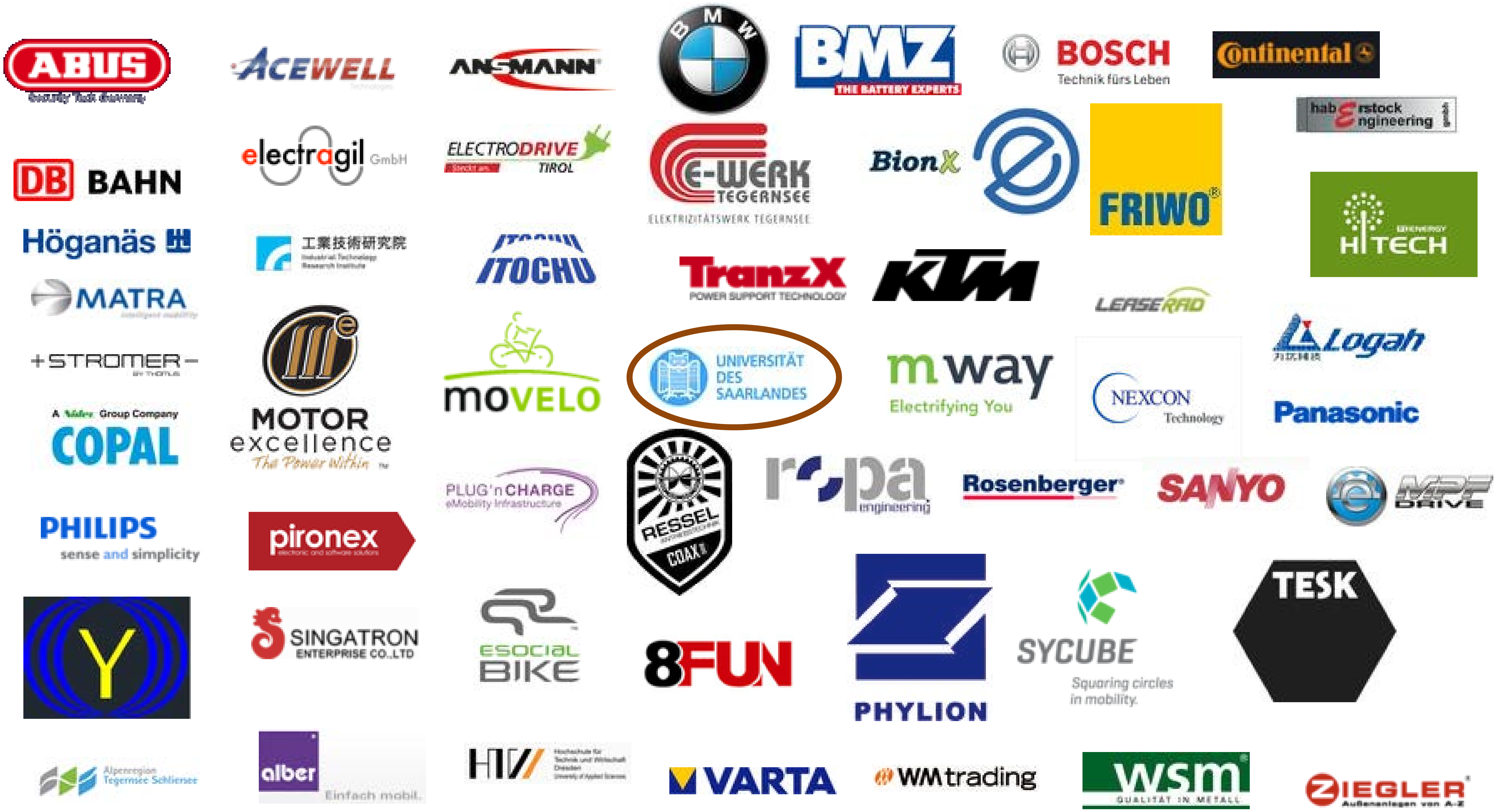
CANopen

Application profile for energy management systems

Part 1: General definitions

EnergyBus e.V.

EnergyBus



Outline

- EnergyBus documentation
  - overview of CAN / CANopen
  - EnergyBus extension

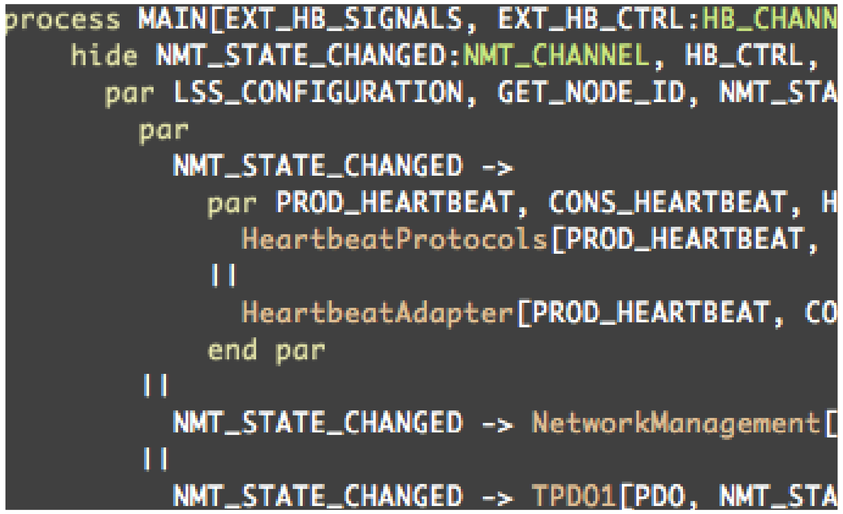
CiA 454 Work Draft

CANopen

Application profile for energy management systems

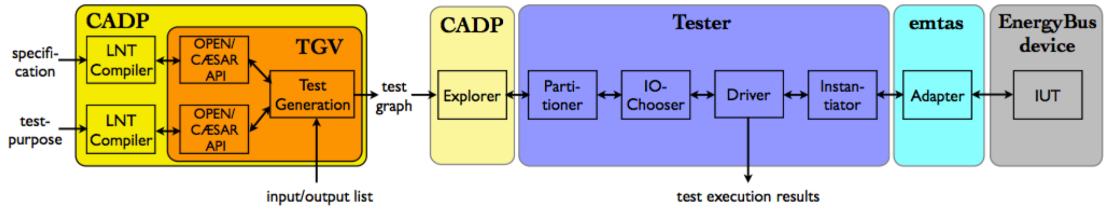
Part 1: General definitions

- Formal specification
  - modeling techniques
  - specification issues



```
process MAIN[EXT_HB_SIGNALS, EXT_HB_CTRL:HB_CHANN
    hide NMT_STATE_CHANGED:NMT_CHANNEL, HB_CTRL,
      par LSS_CONFIGURATION, GET_NODE_ID, NMT_STA
        par
          NMT_STATE_CHANGED ->
            par PROD_HEARTBEAT, CONS_HEARTBEAT, H
              HeartbeatProtocols[PROD_HEARTBEAT,
            ||
              HeartbeatAdapter[PROD_HEARTBEAT, CO
            end par
        ||
          NMT_STATE_CHANGED -> NetworkManagement[
        ||
          NMT_STATE_CHANGED -> TPDO1[PDO, NMT_STA
```

- Certification framework
  - Model-based testing
  - tool setup
  - abstraction techniques
  - testing results



EnergyBus Documentation

CiA 454 Work Draft

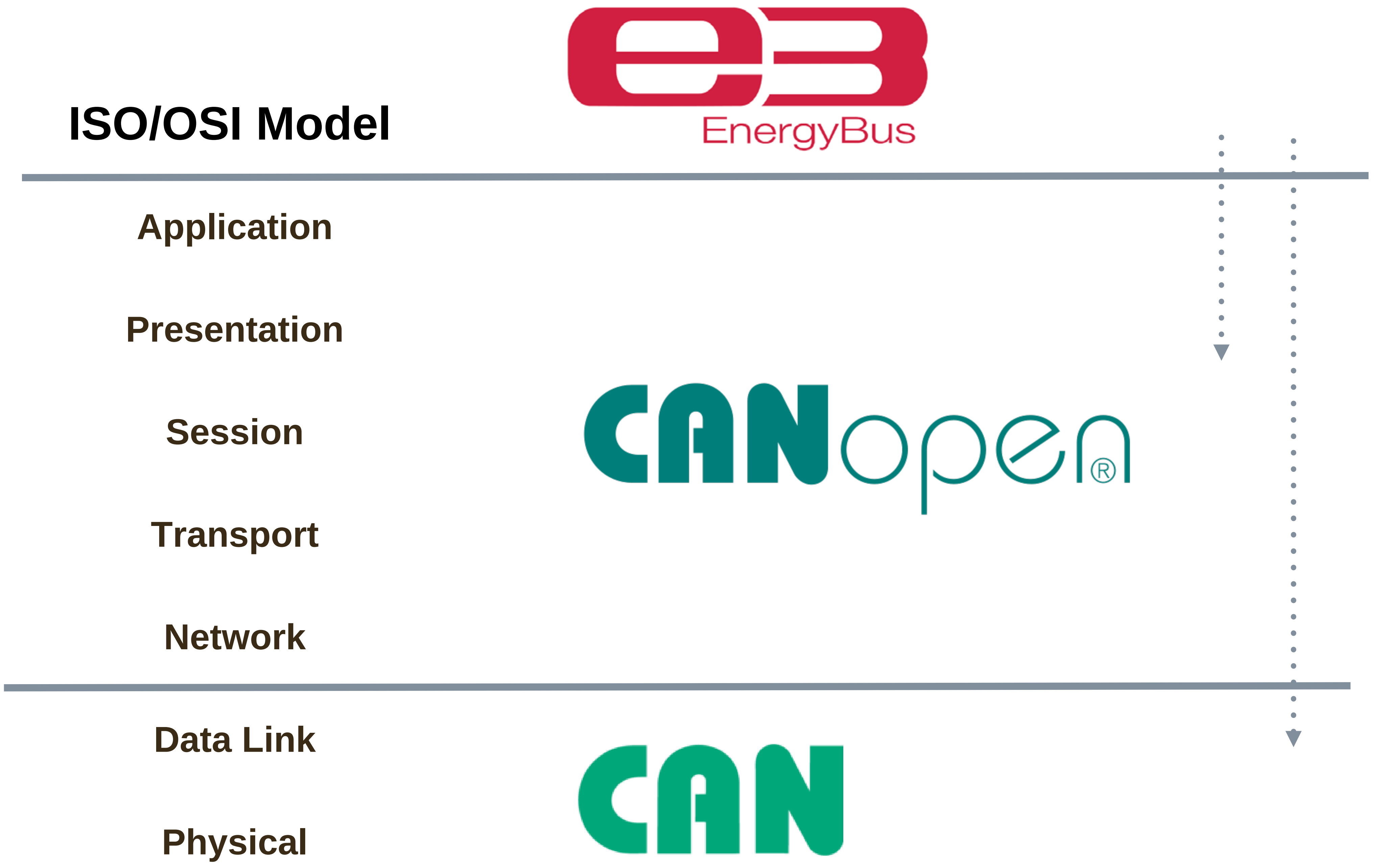
CANopen

Application profile for energy management systems

Part 1: General definitions

bers only and may be changed without

Protocol Stack



ISO/OSI Model

Application

Presentation

Session

Transport

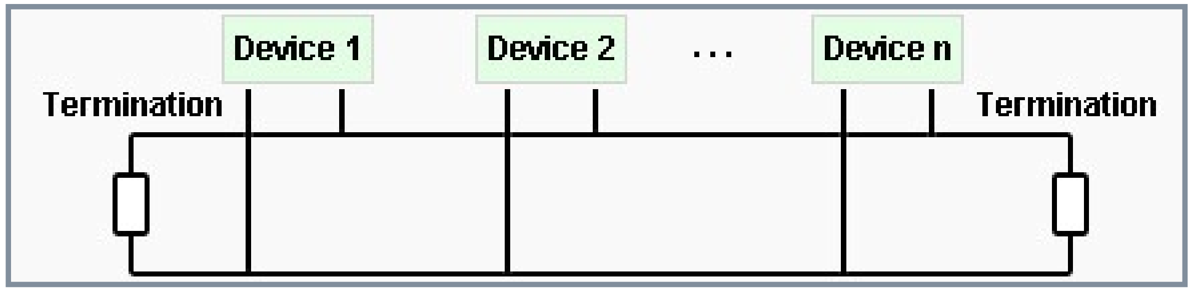
Network

Data Link

Physical

CAN / CANopen

- Bus arbitration: CSMA/BA
- Network topology



- CAN/CANopen frame

function code	node ID	RTR	data length	Data
4 bits	7 bits	1 bit	4 bits	0-8 bytes

CANopen

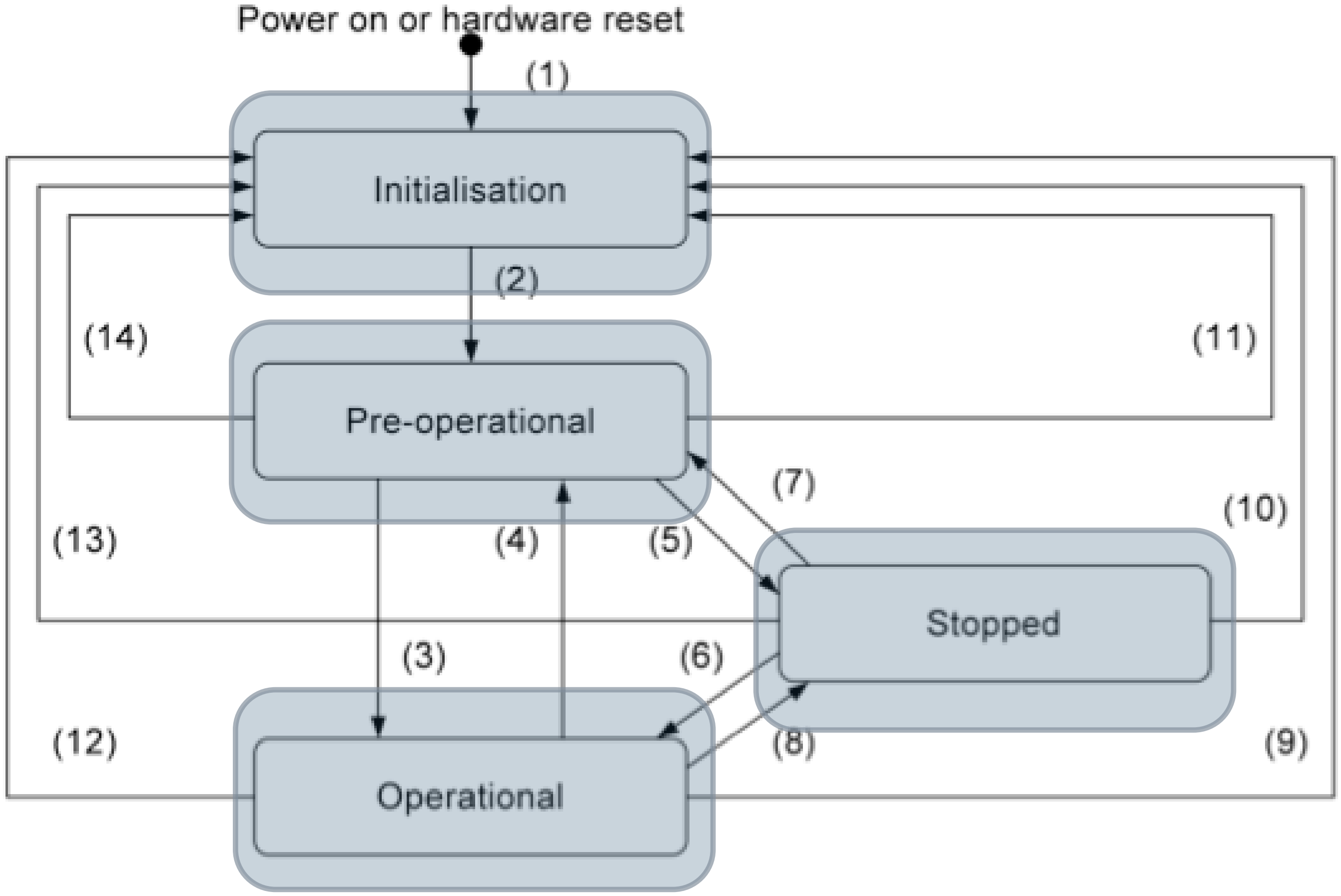
- defines various services
  - NMT
  - SDO
  - PDO
  - LSS
  - Node control
  - SYNC
  - EMCY
  - TIME

CANopen Services

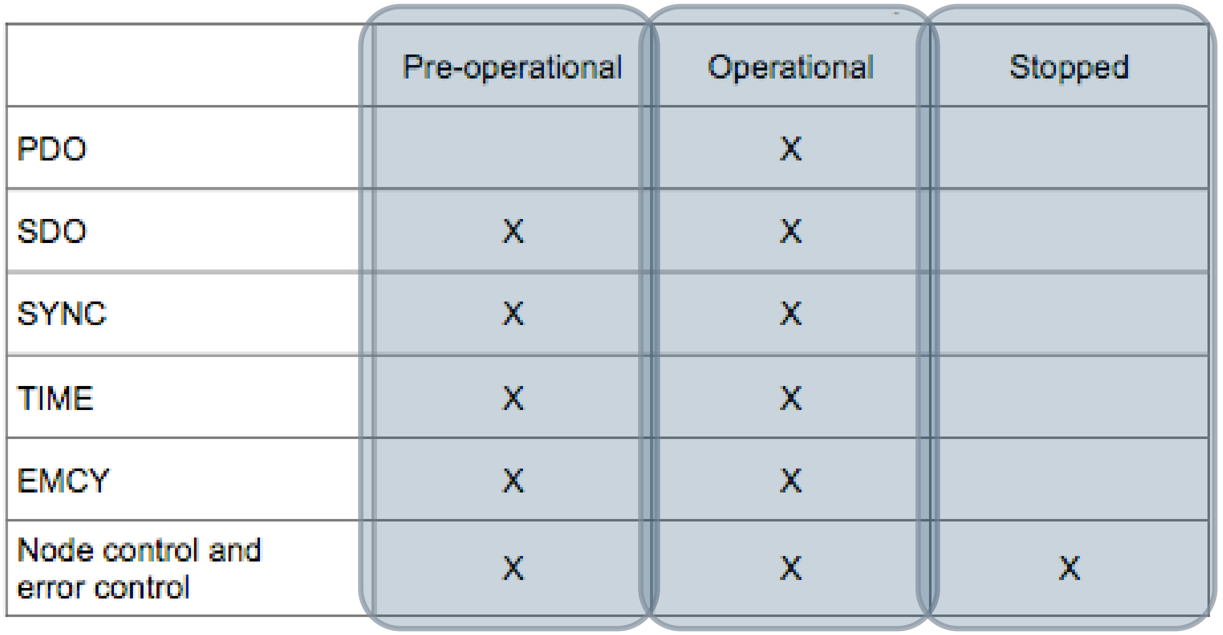
Network Management (NMT)

- master/slave protocol
- operational state

NMT Automaton



Communication capability



	Pre-operational	Operational	Stopped
PDO		X	
SDO	X	X	
SYNC	X	X	
TIME	X	X	
EMCY	X	X	
Node control and error control	X	X	X

CANopen Services

Service Data Object (SDO) communication:

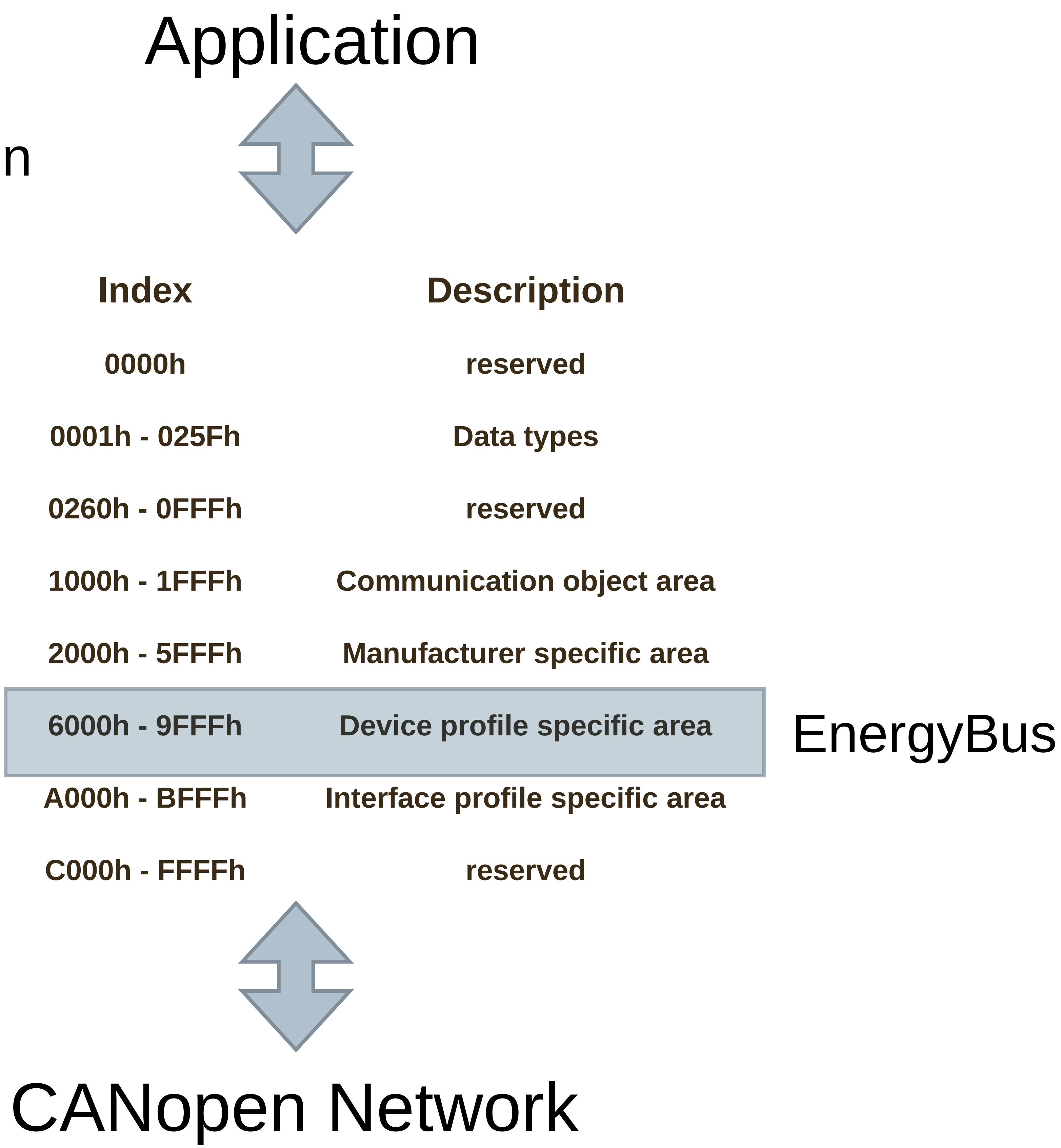
- binary communication
- server/client protocol
- configuration
- segmentation, acknowledgements

Process Data Object (PDO) communication:

- broadcast communication
- producer/consumer protocol
- dynamic data exchange, notifications
- single frame (max. 8 data bytes)

Object Directory (OD)

- 16-bit main index
- 8-bit sub-index
- service configuration
- data exchange



Application

Index	Description
0000h	reserved
0001h - 025Fh	Data types
0260h - 0FFFh	reserved
1000h - 1FFFh	Communication object area
2000h - 5FFFh	Manufacturer specific area
6000h - 9FFFh	Device profile specific area
A000h - BFFFh	Interface profile specific area
C000h - FFFFh	reserved

EnergyBus

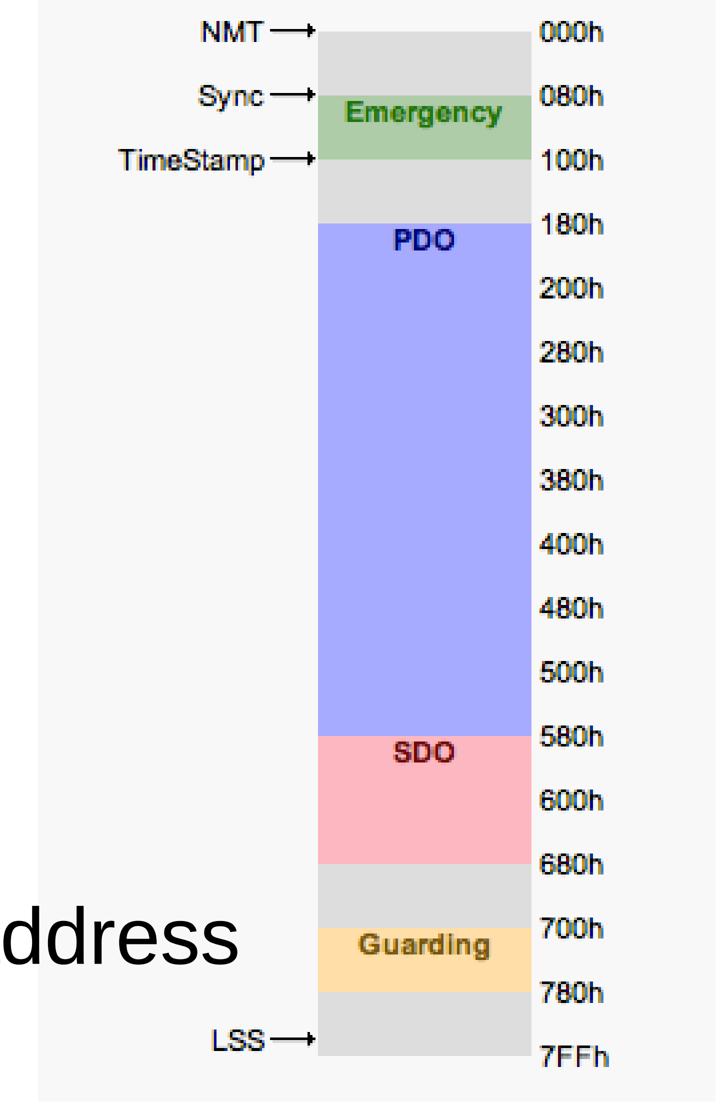
CANopen Network

CANopen Services

Layer Setting Service (LSS):

- master/slave protocol
- independent of NMT state
- detection of connected/unconfigured devices
- configuration of node-ID
- identification by device-specific 8-byte LSS address

highest priority

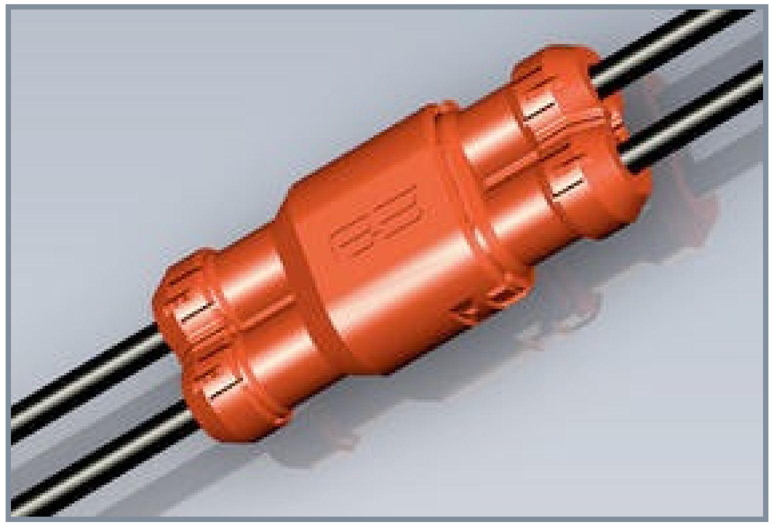


lowest priority

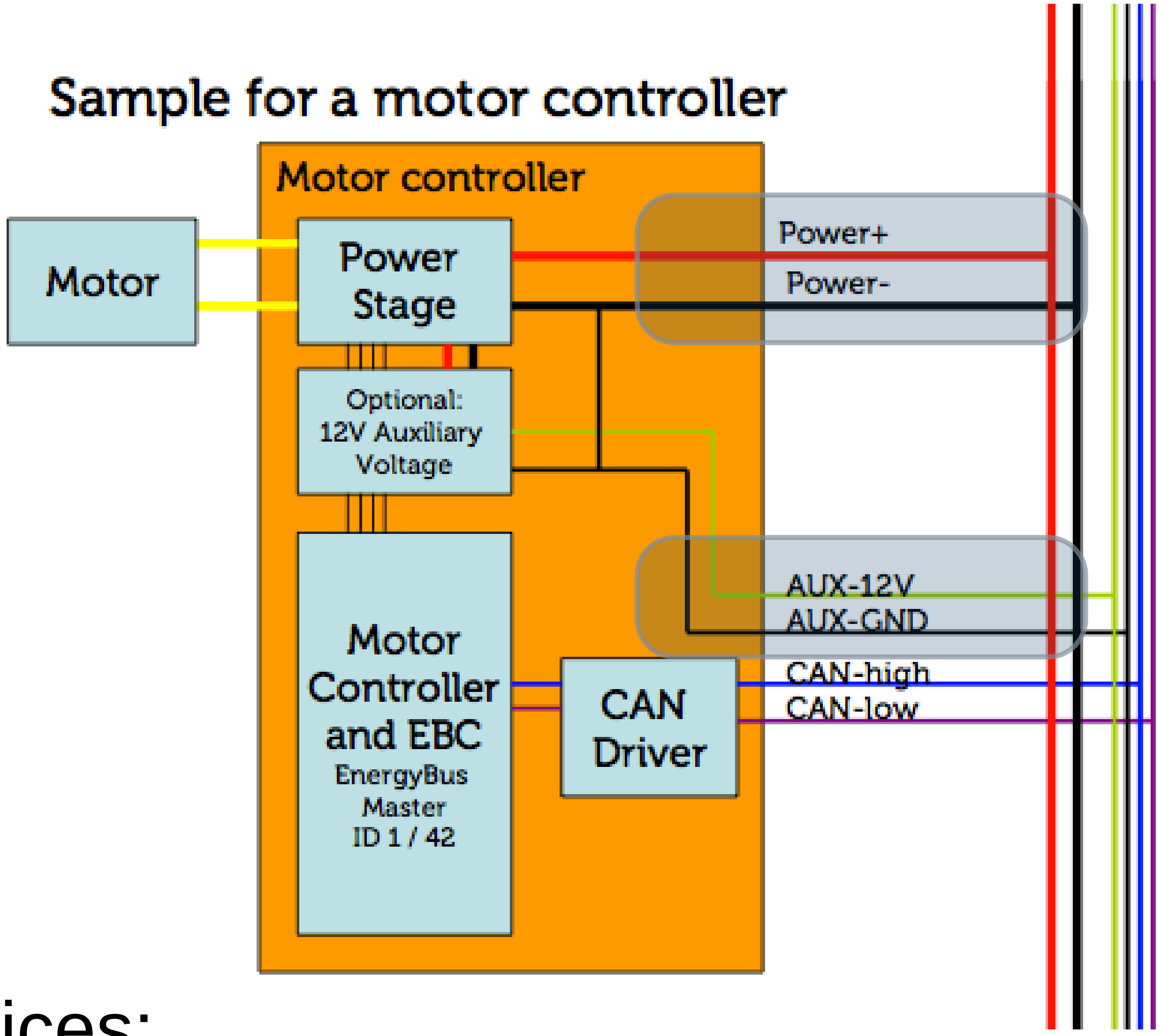
EnergyBus

Hardware aspects

- family of connectors
- additional power lines (MAIN and AUX)



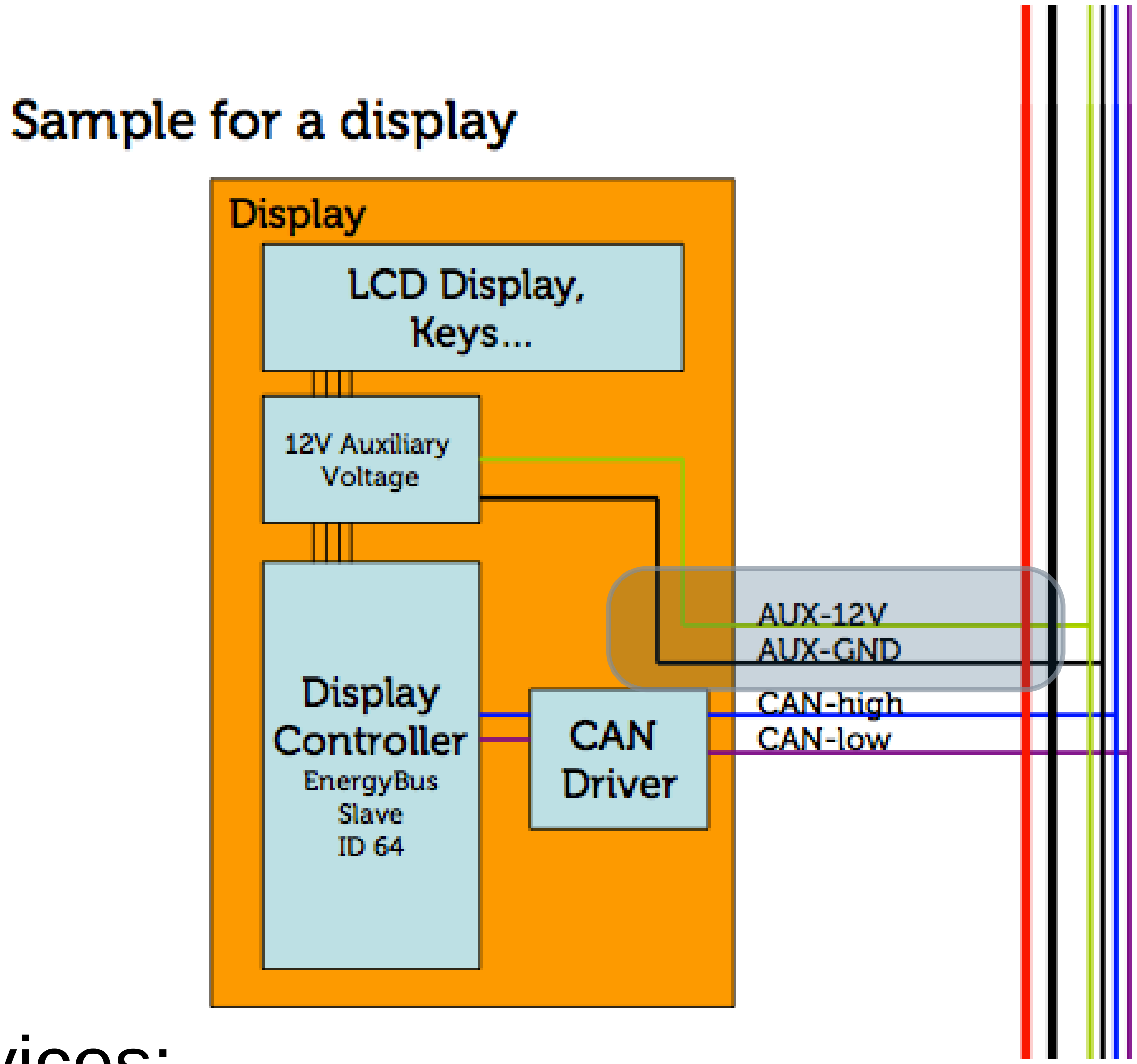
Active Devices



further active devices:

- voltage converter
- battery pack
- load monitoring unit
- generator unit
- …

Passive Devices



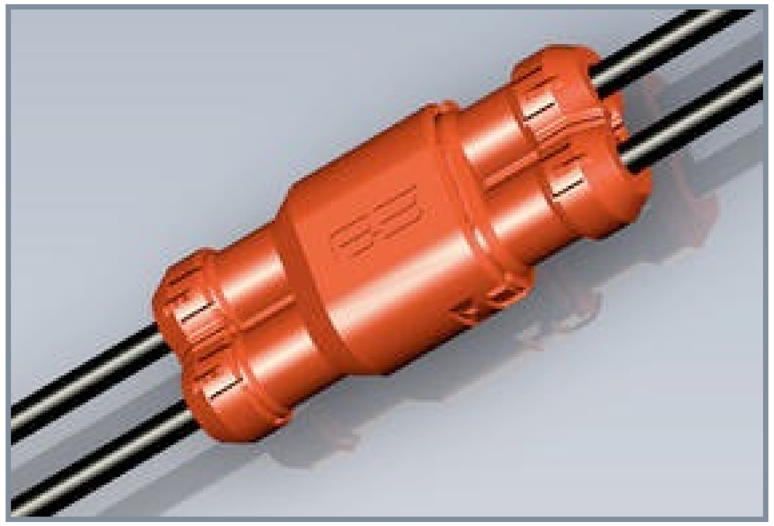
further passive devices:

- sensor unit
- gateway unit
- security unit
- manufacture specific unit
- …

EnergyBus

Hardware aspects

- family of connectors
- additional power lines (MAIN and AUX)



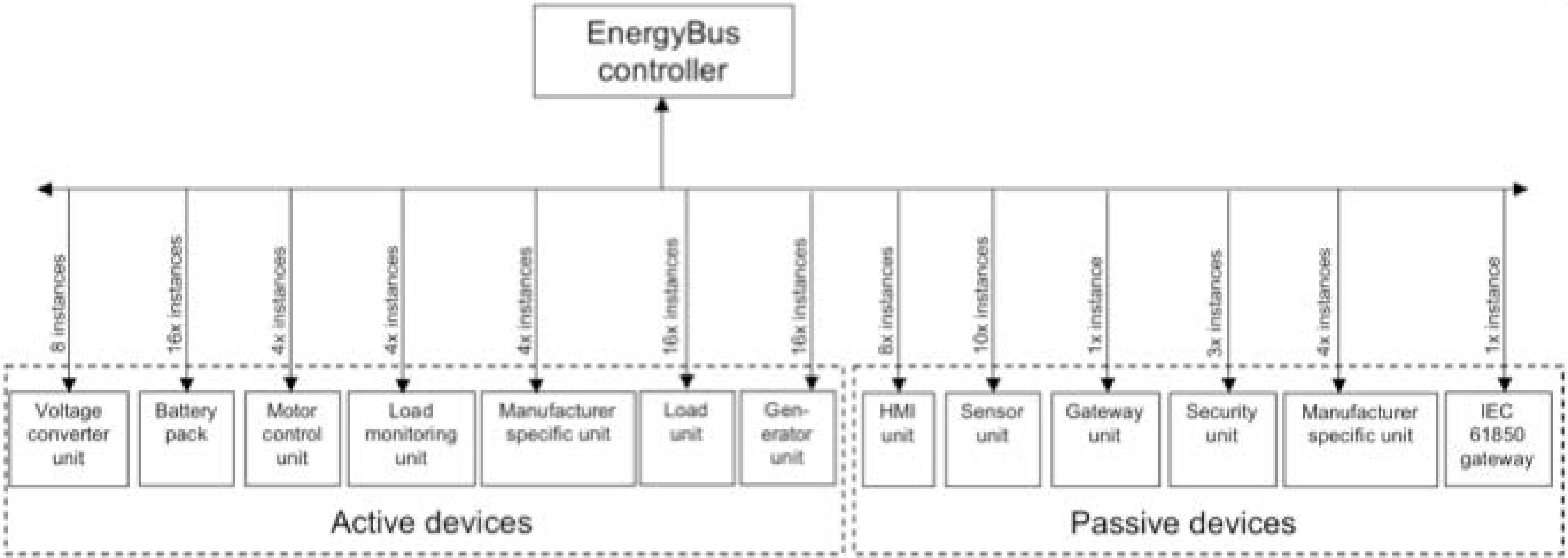
Software aspects

- protocols
- Object Directory definitions
- specific EnergyBus applications

EnergyBus Network

EnergyBus controller (EBC)

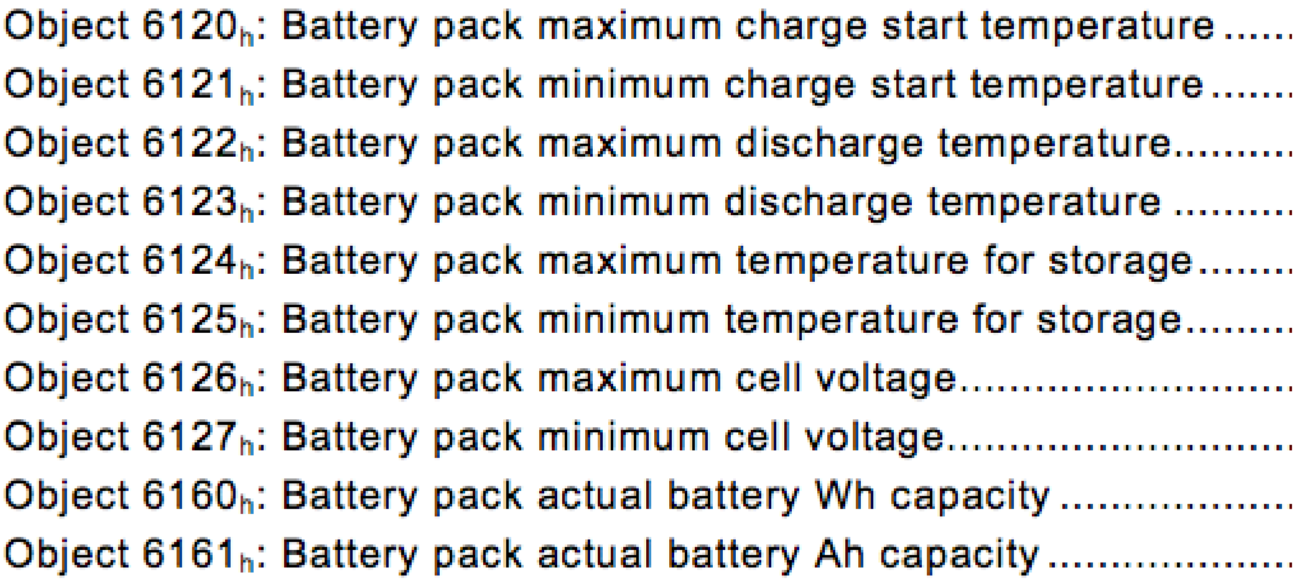
- distribution of energy
- ensures safety of the network
- monitors and controls other EMS devices
- acts as NMT and LSS Master
- maintains SDO connections to all devices
- only one activated EBC in the network
- fixed node-ID 01_h



EnergyBus Virtual Devices

- extend the Object Directory

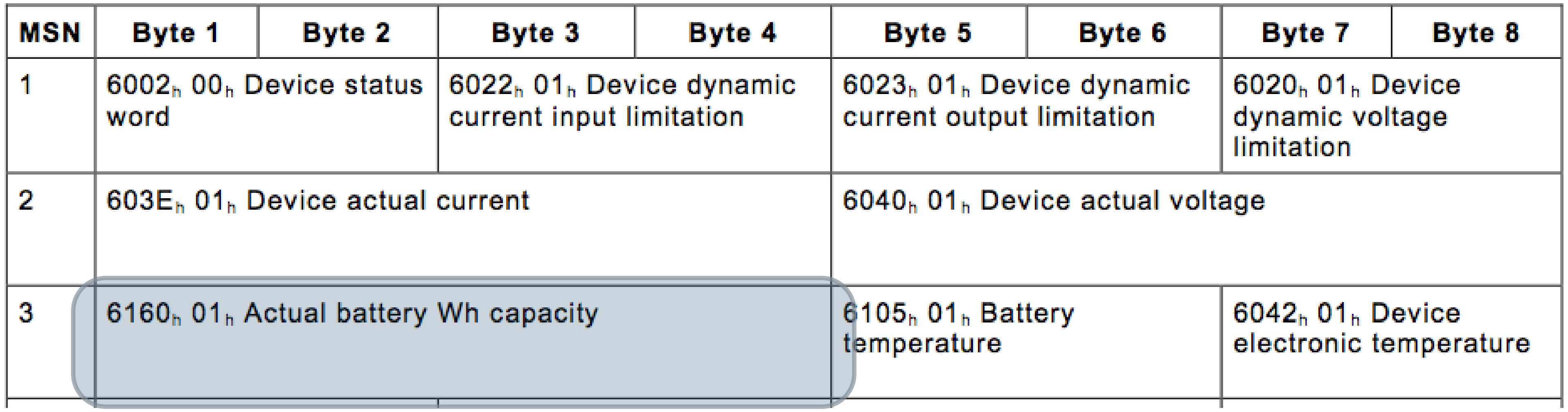
Index	Description
0000h	reserved
0001h - 025Fh	Data types
0260h - 0FFFh	reserved
1000h - 1FFFh	Communication object area
2000h - 5FFFh	Manufacturer specific area
6000h - 9FFFh	Device profile specific area
A000h - BFFFh	Interface profile specific area
C000h - FFFFh	reserved



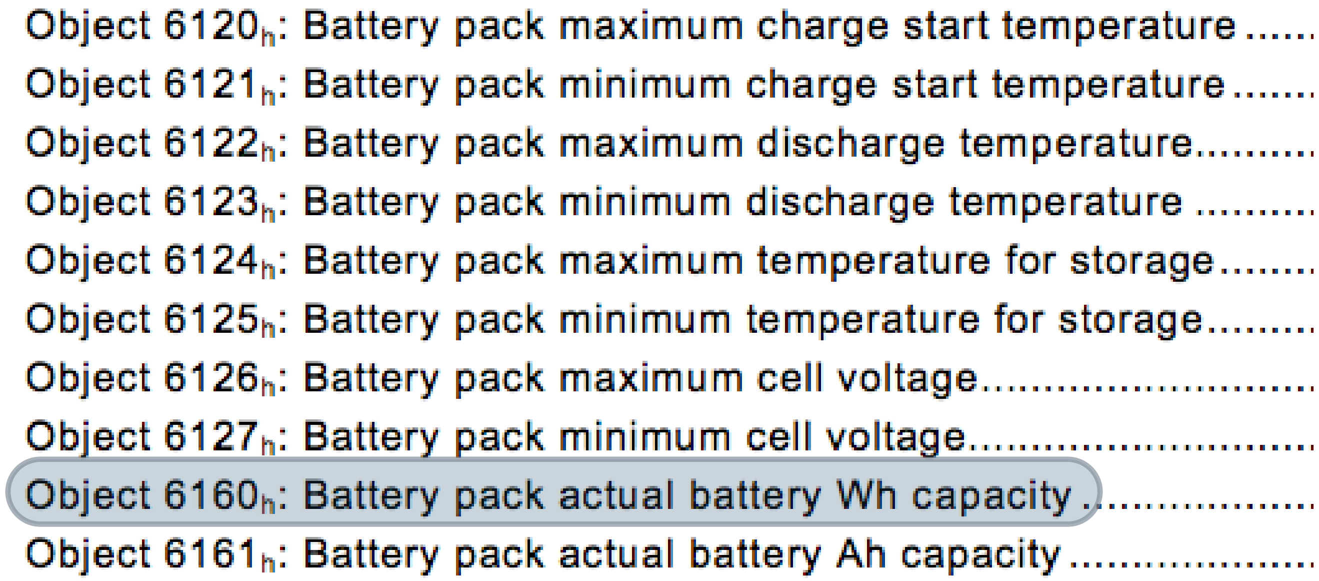
Object 6120_h: Battery pack maximum charge start temperature ......
Object 6121_h: Battery pack minimum charge start temperature ....
Object 6122_h: Battery pack maximum discharge temperature.........
Object 6123_h: Battery pack minimum discharge temperature ....
Object 6124_h: Battery pack maximum temperature for storage........
Object 6125_h: Battery pack minimum temperature for storage.........
Object 6126_h: Battery pack maximum cell voltage..........................
Object 6127_h: Battery pack minimum cell voltage...........................
Object 6160_h: Battery pack actual battery Wh capacity ....
Object 6161_h: Battery pack actual battery Ah capacity ....

EnergyBus Virtual Devices

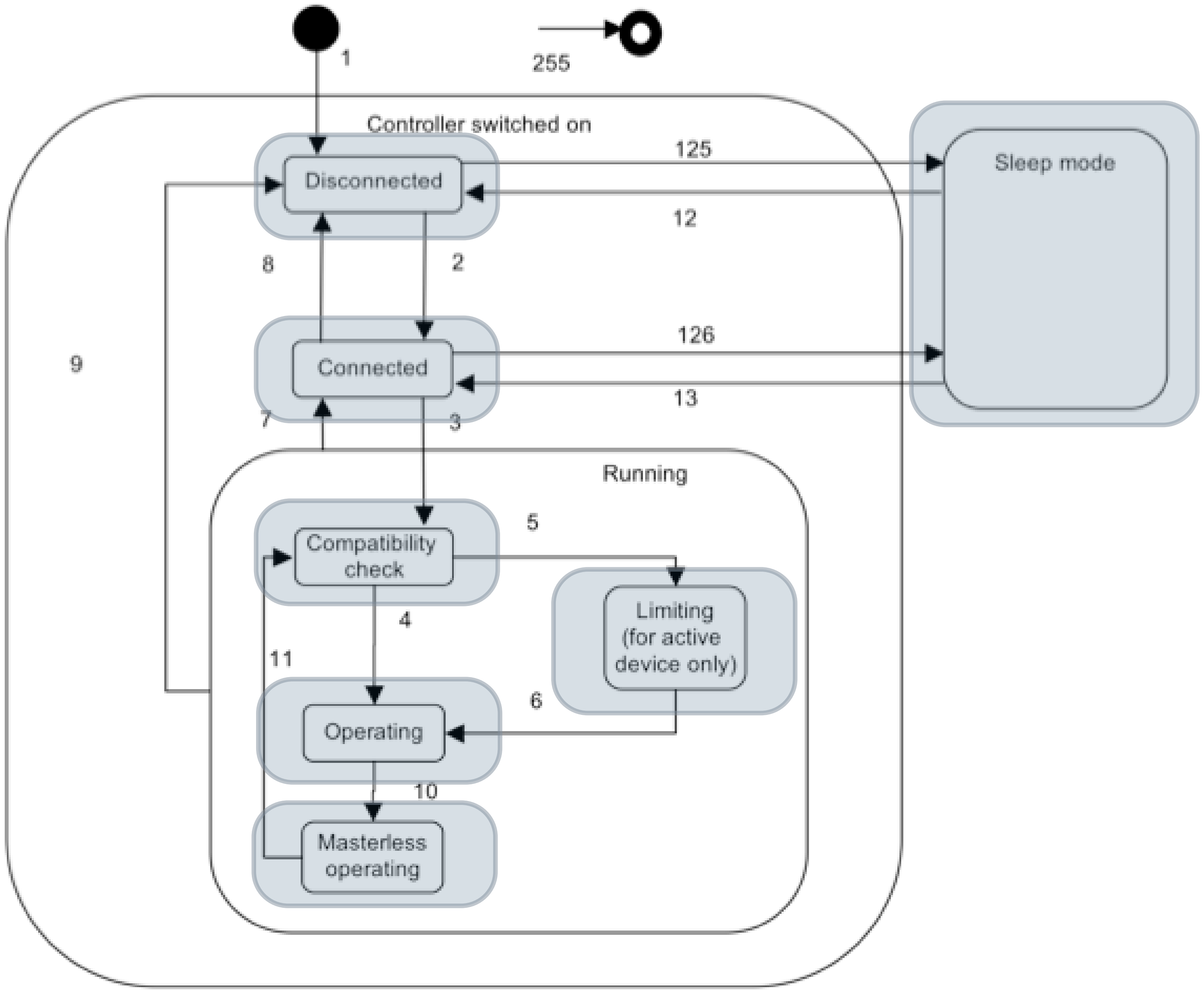
- extend the Object Directory
- predefined set of PDO messages



MSN	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1	6002$_h$ 00$_h$ Device status word		6022$_h$ 01$_h$ Device dynamic current input limitation		6023$_h$ 01$_h$ Device dynamic current output limitation		6020$_h$ 01$_h$ Device dynamic voltage limitation	
2	603E$_h$ 01$_h$ Device actual current				6040$_h$ 01$_h$ Device actual voltage			
3	6160$_h$ 01$_h$ Actual battery Wh capacity				6105$_h$ 01$_h$ Battery temperature		6042$_h$ 01$_h$ Device electronic temperature	



Energy Management Automaton



Disconnected:

- no power consumption
- no CAN communication

Connected:

- low power supply (CAN)
- layer setting service

Compatibility check:

- awaits EBC check

Limiting:

- adjustment of electrical parameters

Operating:

- device specific application is running

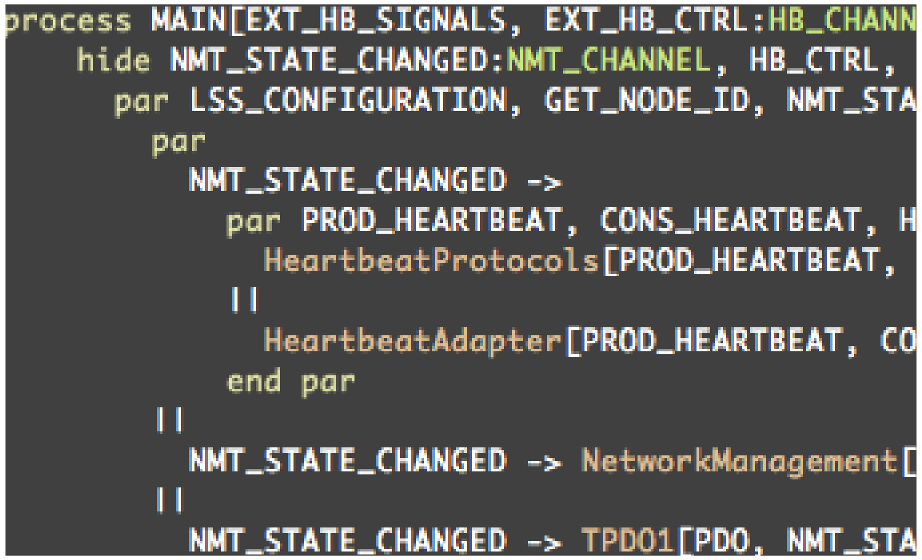
Masterless operating:

- optional, running without Master
- mandatory for stationary EMS

Sleep mode:

- energy saving mode
- Sleep Mode automaton & service

Formal Specification



```
process MAIN[EXT_HB_SIGNALS, EXT_HB_CTRL:HB_CHANN
    hide NMT_STATE_CHANGED:NMT_CHANNEL, HB_CTRL,
      par LSS_CONFIGURATION, GET_NODE_ID, NMT_STA
        par
          NMT_STATE_CHANGED ->
            par PROD_HEARTBEAT, CONS_HEARTBEAT, H
              HeartbeatProtocols[PROD_HEARTBEAT,
            ||
              HeartbeatAdapter[PROD_HEARTBEAT, CO
            end par
        ||
          NMT_STATE_CHANGED -> NetworkManagement[
        ||
          NMT_STATE_CHANGED -> TPDO1[PDO, NMT_STA
```

Formal Specification

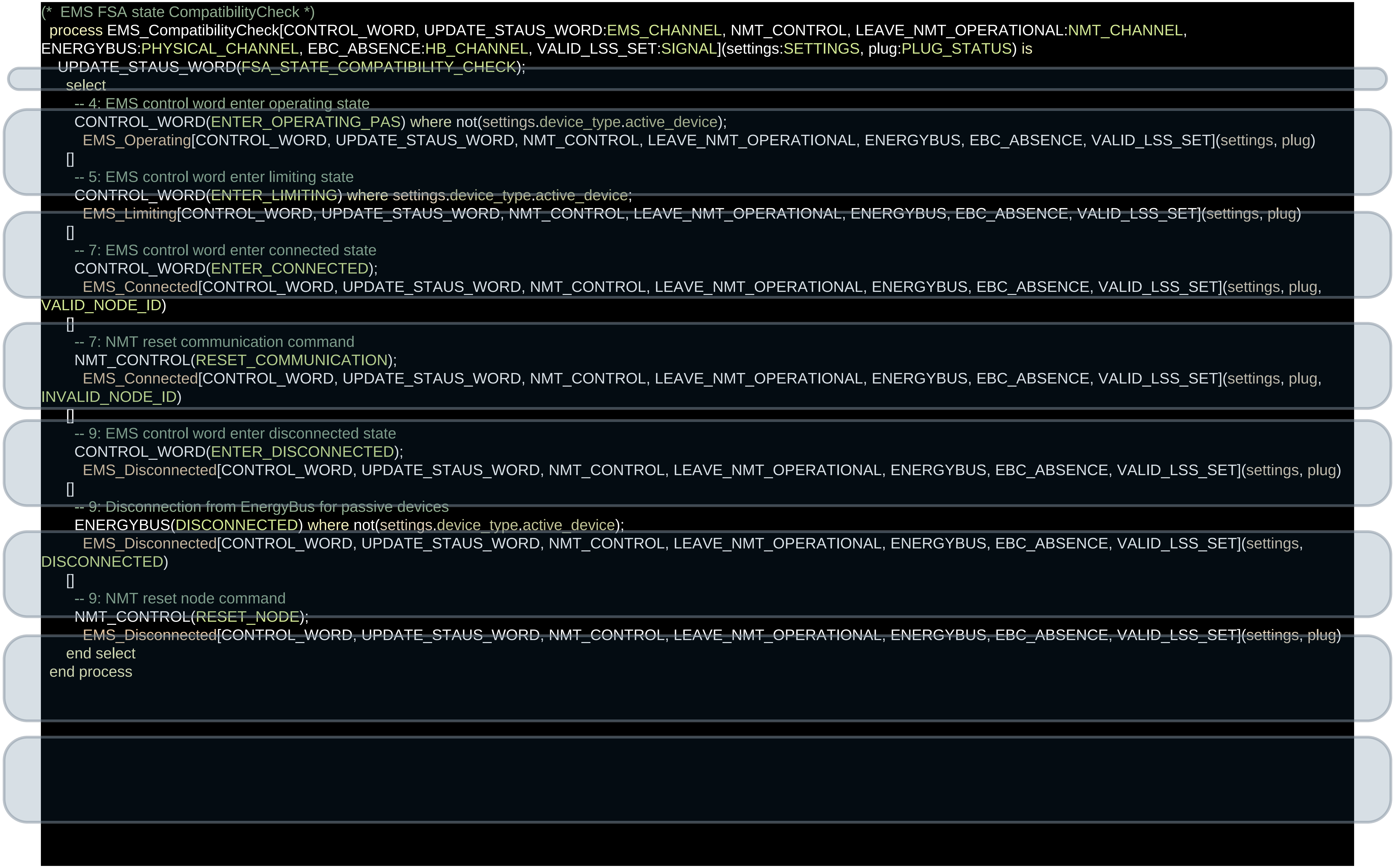
Informal documentation

- CANopen CiA 301, 302 series, 305
- EnergyBus CiA 454 series - 14 documents
- textual description
- sequence diagrams
- automata

Formal language: LNT

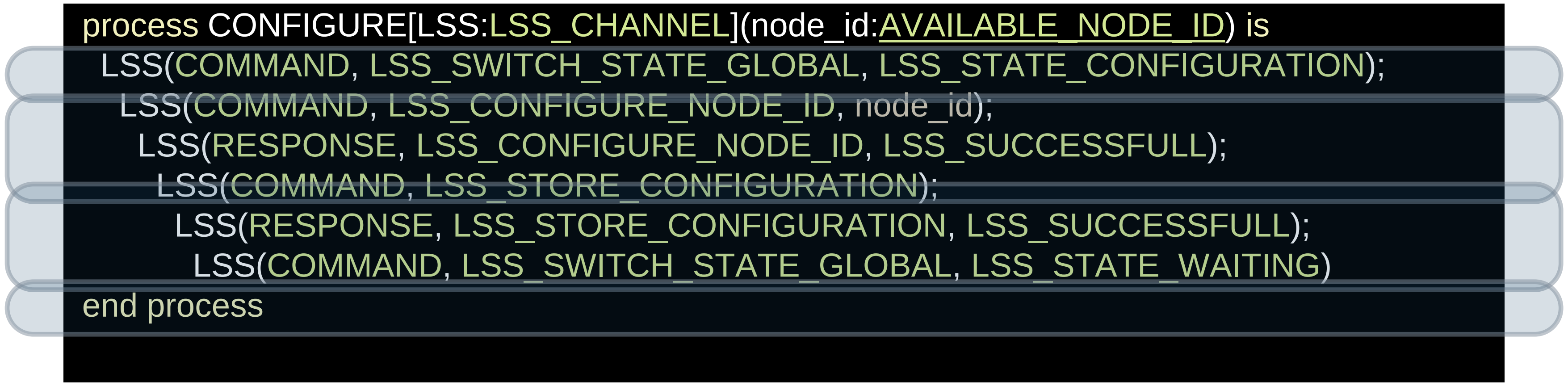
- descendent of LOTOS & E-LOTOS
- modern combination of process algebra, functional and imperative languages
- LTS semantics / SOS rules
- supported by CADP http://cadp.inria.fr

Formalization of Automata

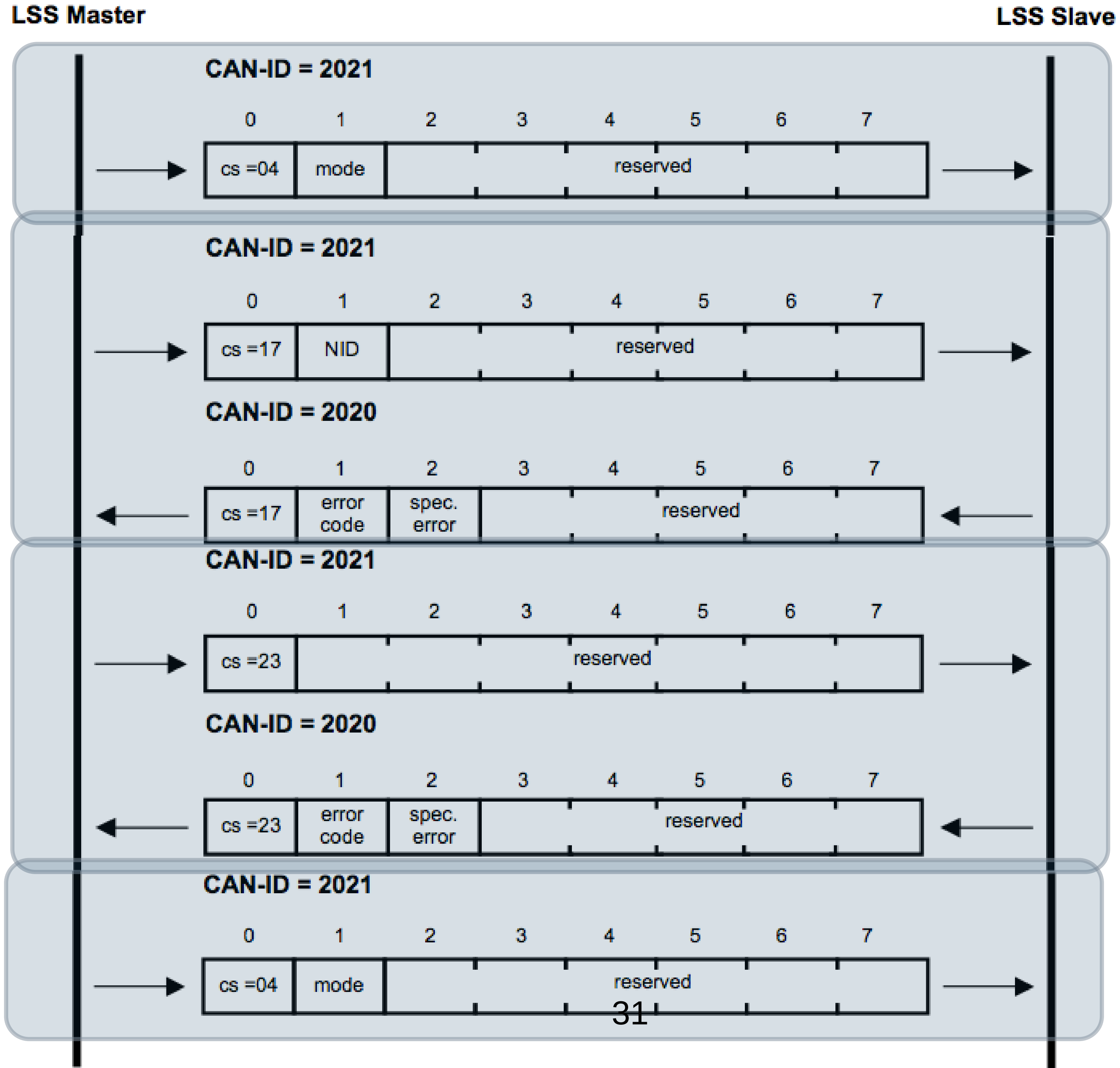


```
(*  EMS FSA state CompatibilityCheck *)
  process EMS_CompatibilityCheck[CONTROL_WORD, UPDATE_STAUS_WORD:EMS_CHANNEL, NMT_CONTROL, LEAVE_NMT_OPERATIONAL:NMT_CHANNEL,
ENERGYBUS:PHYSICAL_CHANNEL, EBC_ABSENCE:HB_CHANNEL, VALID_LSS_SET:SIGNAL](settings:SETTINGS, plug:PLUG_STATUS) is
    UPDATE_STAUS_WORD(FSA_STATE_COMPATIBILITY_CHECK);
      select
        -- 4: EMS control word enter operating state
        CONTROL_WORD(ENTER_OPERATING_PAS) where not(settings.device_type.active_device);
          EMS_Operating[CONTROL_WORD, UPDATE_STAUS_WORD, NMT_CONTROL, LEAVE_NMT_OPERATIONAL, ENERGYBUS, EBC_ABSENCE, VALID_LSS_SET](settings, plug)
      []
        -- 5: EMS control word enter limiting state
        CONTROL_WORD(ENTER_LIMITING) where settings.device_type.active_device;
          EMS_Limiting[CONTROL_WORD, UPDATE_STAUS_WORD, NMT_CONTROL, LEAVE_NMT_OPERATIONAL, ENERGYBUS, EBC_ABSENCE, VALID_LSS_SET](settings, plug)
      []
        -- 7: EMS control word enter connected state
        CONTROL_WORD(ENTER_CONNECTED);
          EMS_Connected[CONTROL_WORD, UPDATE_STAUS_WORD, NMT_CONTROL, LEAVE_NMT_OPERATIONAL, ENERGYBUS, EBC_ABSENCE, VALID_LSS_SET](settings, plug,
VALID_NODE_ID)
      []
        -- 7: NMT reset communication command
        NMT_CONTROL(RESET_COMMUNICATION);
          EMS_Connected[CONTROL_WORD, UPDATE_STAUS_WORD, NMT_CONTROL, LEAVE_NMT_OPERATIONAL, ENERGYBUS, EBC_ABSENCE, VALID_LSS_SET](settings, plug,
INVALID_NODE_ID)
      []
        -- 9: EMS control word enter disconnected state
        CONTROL_WORD(ENTER_DISCONNECTED);
          EMS_Disconnected[CONTROL_WORD, UPDATE_STAUS_WORD, NMT_CONTROL, LEAVE_NMT_OPERATIONAL, ENERGYBUS, EBC_ABSENCE, VALID_LSS_SET](settings, plug)
      []
        -- 9: Disconnection from EnergyBus for passive devices
        ENERGYBUS(DISCONNECTED) where not(settings.device_type.active_device);
          EMS_Disconnected[CONTROL_WORD, UPDATE_STAUS_WORD, NMT_CONTROL, LEAVE_NMT_OPERATIONAL, ENERGYBUS, EBC_ABSENCE, VALID_LSS_SET](settings,
DISCONNECTED)
      []
        -- 9: NMT reset node command
        NMT_CONTROL(RESET_NODE);
          EMS_Disconnected[CONTROL_WORD, UPDATE_STAUS_WORD, NMT_CONTROL, LEAVE_NMT_OPERATIONAL, ENERGYBUS, EBC_ABSENCE, VALID_LSS_SET](settings, plug)
      end select
  end process
```

Formalization of Sequence Diagrams



```
process CONFIGURE[LSS:LSS_CHANNEL](node_id:AVAILABLE_NODE_ID) is
 LSS(COMMAND, LSS_SWITCH_STATE_GLOBAL, LSS_STATE_CONFIGURATION);
  LSS(COMMAND, LSS_CONFIGURE_NODE_ID, node_id);
   LSS(RESPONSE, LSS_CONFIGURE_NODE_ID, LSS_SUCCESSFULL);
    LSS(COMMAND, LSS_STORE_CONFIGURATION);
     LSS(RESPONSE, LSS_STORE_CONFIGURATION, LSS_SUCCESSFULL);
      LSS(COMMAND, LSS_SWITCH_STATE_GLOBAL, LSS_STATE_WAITING)
end process
```



0	1	2	3	4	5	6	7
cs =04	mode	reserved					

0	1	2	3	4	5	6	7
cs =17	NID	reserved					

0	1	2	3	4	5	6	7
cs =17	error code	spec. error	reserved				

0	1	2	3	4	5	6	7
cs =23	reserved						

0	1	2	3	4	5	6	7
cs =23	error code	spec. error	reserved				

0	1	2	3	4	5	6	7
cs =04	mode	reserved					

Specification Effort

Component	Documentation (pages)	LNT code (lines)
NMT	8	260
Heartbeat	6	200
EMCY/Error	4	145
LSS	62	360
EMS	3	440
PDO	45	60
SDO	25	30
OD/Variables	(300)	70

Time spent: 6 months

- including progression of the EnergyBus specification
- including model abstractions

Specification Issues Detected

Issue 1:

ambiguities in the Node-ID configuration in LSS v2

➡ when is a node-ID temporarily / persistently stored?

Issue 2:

insufficient specification of Sleep Mode

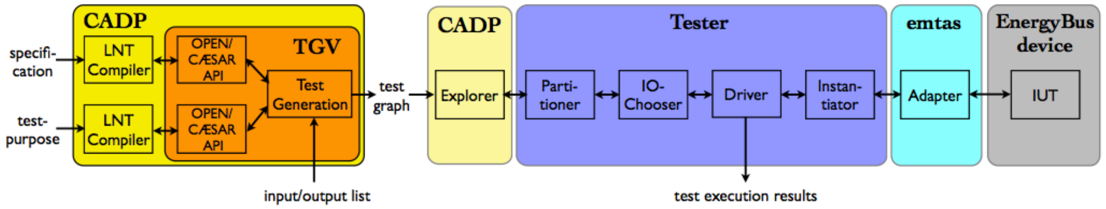
➡ interferences with NMT competence

➡ unclear definition of involved messages / events

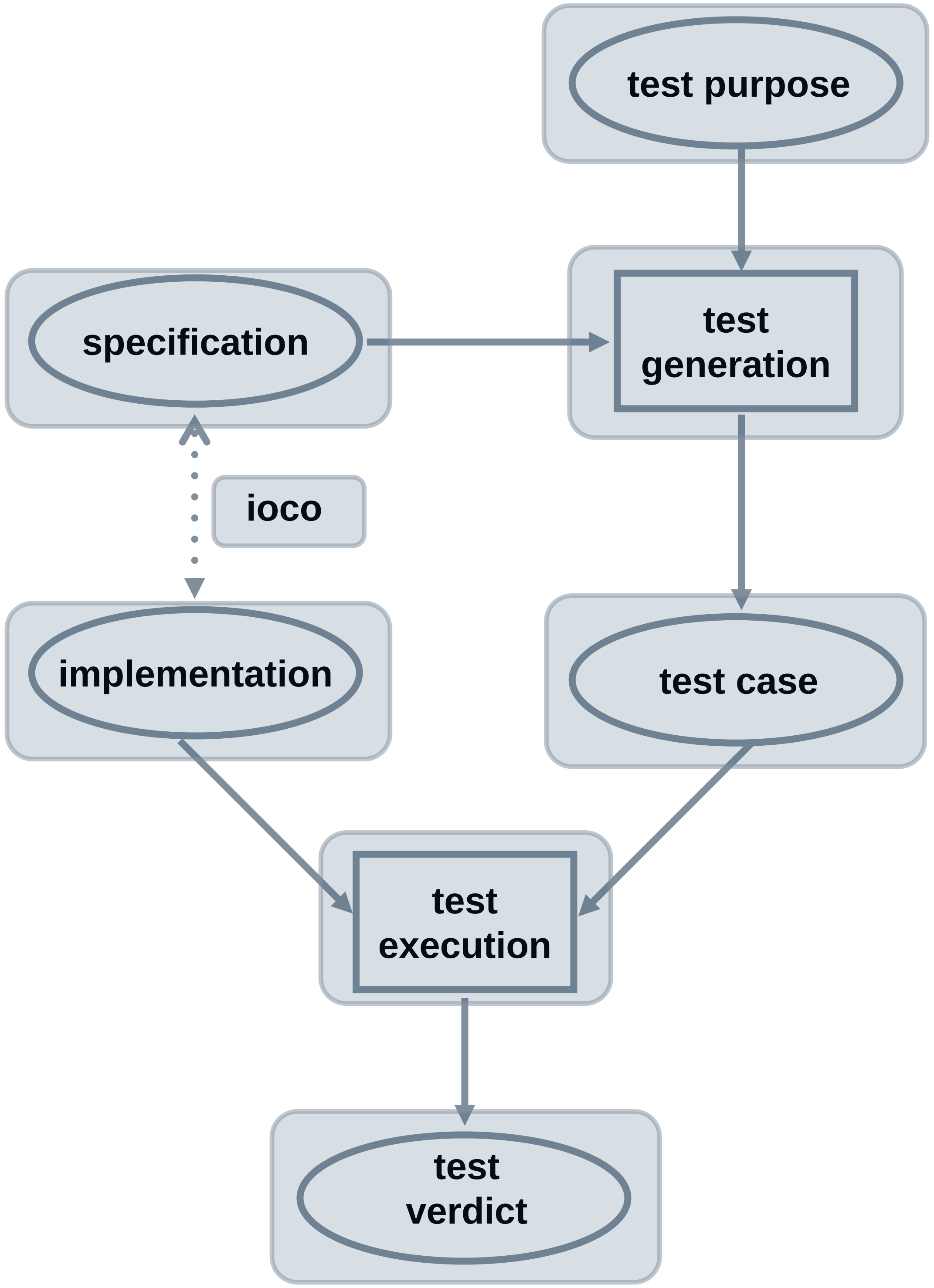
Issue 3:

confusing and non-consistent naming in new documents

Certification Framework

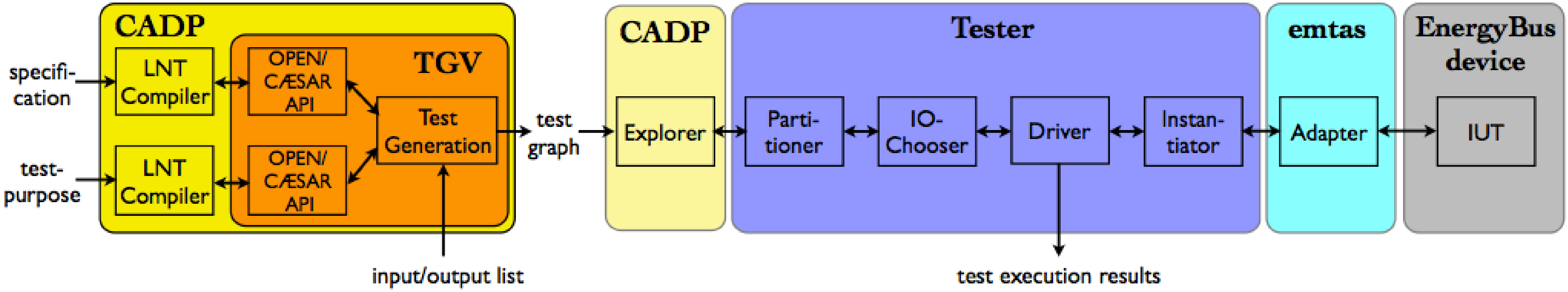


Model-based Testing



- **implementation:**
  - Implementation Under Test (IUT) is real object
  - assume existence of formal model
- **specification:**
  - formal model of correct behavior
- **input-output conformance relation:**
  - defines condition for correct implementation
- **test purpose:**
  - formal model of relevant application scenario
- **test generation:**
  - automatic derivation of test cases
- **test cases:**
  - experiment description as formal model
- **test execution:**
  - execute experiment
- **test verdict:**
  - **pass**, **fail**, **inconclusive**

Tool Setup



Offline:

- input: specification, test purpose as .lnt
- input/output list for turning LTS into IOLTS
- test graph generation by TGV [INRIA Rennes]
- output as .bcg

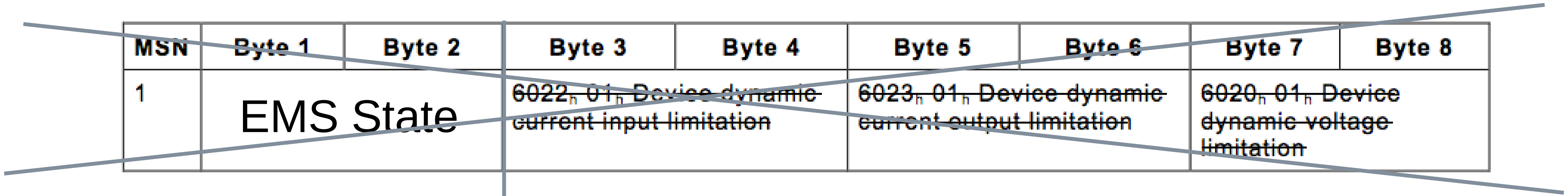
Online:

- on-the-fly derivation of virtual test case
- provides input/receives output via C-Library from **emtas**
- output: test verdict, run log as .txt

Fighting State-Space Explosion

combined approaches:

- on-the-fly algorithms of the OPEN/CÆSAR API
- data abstraction
  - qualitative: two-valued information
  - relative: three-valued information
- functional abstraction
  - focus on important data
  - focus on needed protocol parts

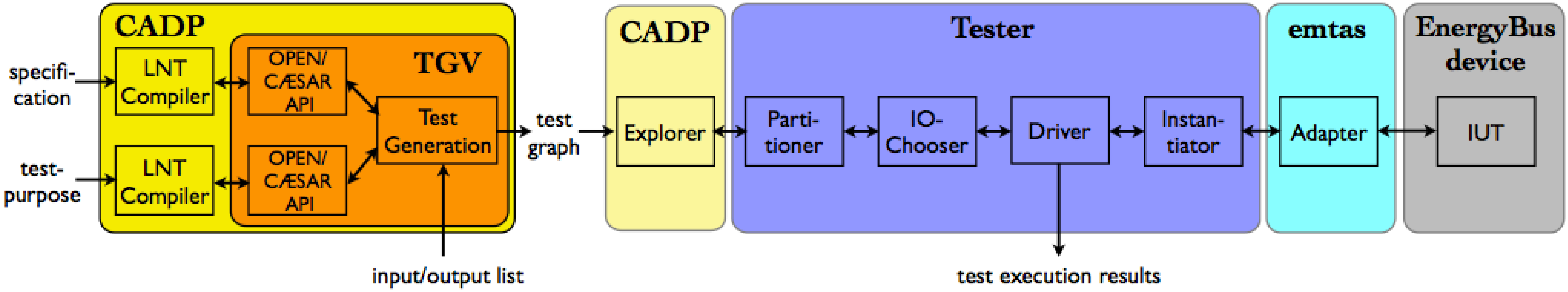


MSN	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1	EMS State		~~6022$_h$, 01$_h$ Device dynamic current input limitation~~		~~6023$_h$, 01$_h$ Device dynamic current output limitation~~		~~6020$_h$, 01$_h$ Device dynamic voltage limitation~~	

LIMITATION / NO_LIMITATION

DECREASE_LIMIT / NO_CHANGE / INCREASE_LIMIT

Applied Tests



Test purposes:

- different initializations of unconfigured devices
- boot-up procedure and on-line PDO transmission of configured devices
- simple scenarios
- 80 lines LNT vs. test graph 600 states / 1100 transitions

IUT:

- sample C applications
- based on the emtas CANopen C-Library
- running on a Linux workstation
- plugged to Tester via CAN connection

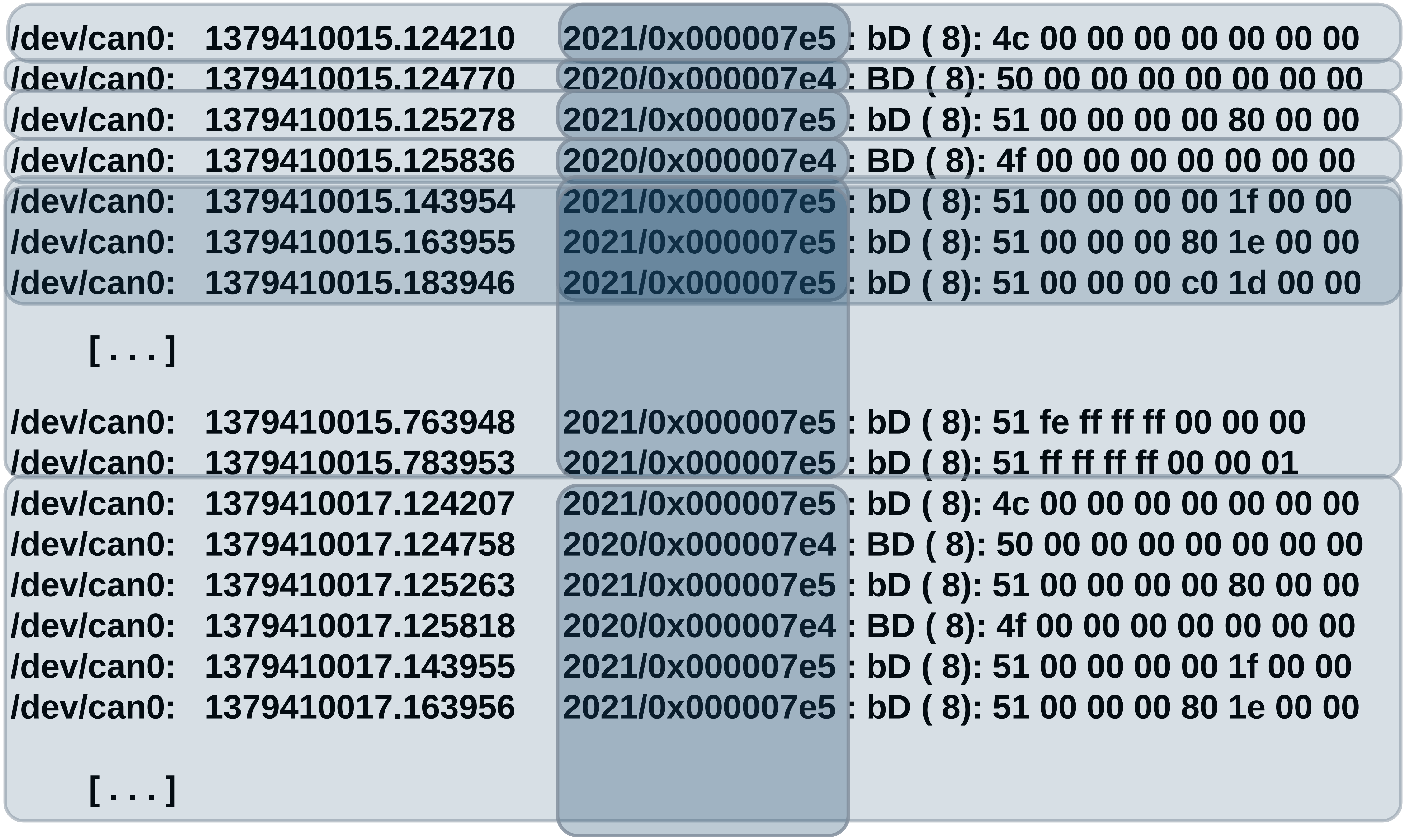
Issue #1 Found in the CANopen Layer

Missing data part of SYNC message

```
/dev/can0:   1371475223.952977    1794/0x00000702 : bD ( 1): 05
/dev/can0:   1371475224.052980    1794/0x00000702 : bD ( 1): 05
/dev/can0:   1371475224.153015    1794/0x00000702 : bD ( 1): 7f
/dev/can0:   1371475224.198949     128/0x00000080 : BD ( 0):
/dev/can0:   1371475224.199208    1793/0x00000701 : BD ( 1): 7f
/dev/can0:   1371475224.252981    1794/0x00000702 : bD ( 1): 7f
/dev/can0:   1371475224.352985    1794/0x00000702 : bD ( 1): 7f
/dev/can0:   1371475224.452983    1794/0x00000702 : bD ( 1): 7f
```

Issue #2 Found in the CANopen Layer

Livelock in LSS Fastscan service

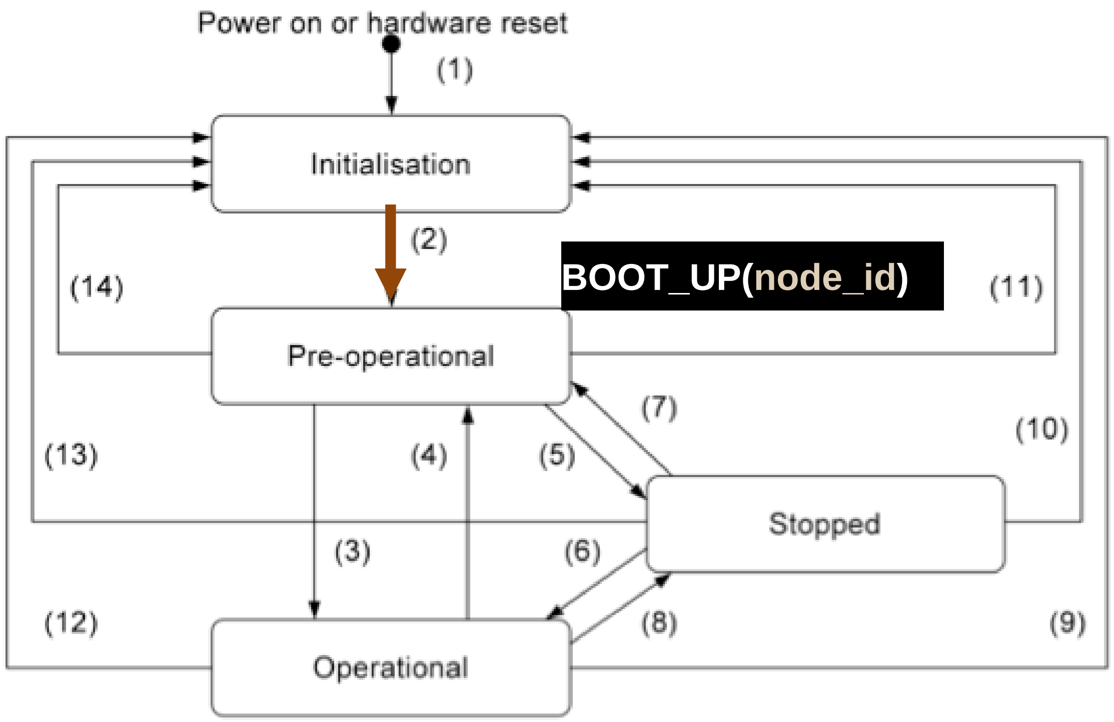


```
/dev/can0:   1379410015.124210    2021/0x000007e5 : bD ( 8): 4c 00 00 00 00 00 00 00
/dev/can0:   1379410015.124770    2020/0x000007e4 : BD ( 8): 50 00 00 00 00 00 00 00
/dev/can0:   1379410015.125278    2021/0x000007e5 : bD ( 8): 51 00 00 00 00 80 00 00
/dev/can0:   1379410015.125836    2020/0x000007e4 : BD ( 8): 4f 00 00 00 00 00 00 00
/dev/can0:   1379410015.143954    2021/0x000007e5 : bD ( 8): 51 00 00 00 00 1f 00 00
/dev/can0:   1379410015.163955    2021/0x000007e5 : bD ( 8): 51 00 00 00 80 1e 00 00
/dev/can0:   1379410015.183946    2021/0x000007e5 : bD ( 8): 51 00 00 00 c0 1d 00 00

        [ . . . ]

/dev/can0:   1379410015.763948    2021/0x000007e5 : bD ( 8): 51 fe ff ff ff 00 00 00
/dev/can0:   1379410015.783953    2021/0x000007e5 : bD ( 8): 51 ff ff ff ff 00 00 01
/dev/can0:   1379410017.124207    2021/0x000007e5 : bD ( 8): 4c 00 00 00 00 00 00 00
/dev/can0:   1379410017.124758    2020/0x000007e4 : BD ( 8): 50 00 00 00 00 00 00 00
/dev/can0:   1379410017.125263    2021/0x000007e5 : bD ( 8): 51 00 00 00 00 80 00 00
/dev/can0:   1379410017.125818    2020/0x000007e4 : BD ( 8): 4f 00 00 00 00 00 00 00
/dev/can0:   1379410017.143955    2021/0x000007e5 : bD ( 8): 51 00 00 00 00 1f 00 00
/dev/can0:   1379410017.163956    2021/0x000007e5 : bD ( 8): 51 00 00 00 80 1e 00 00

        [ . . . ]
```

Issue #3 Found in the CANopen Layer

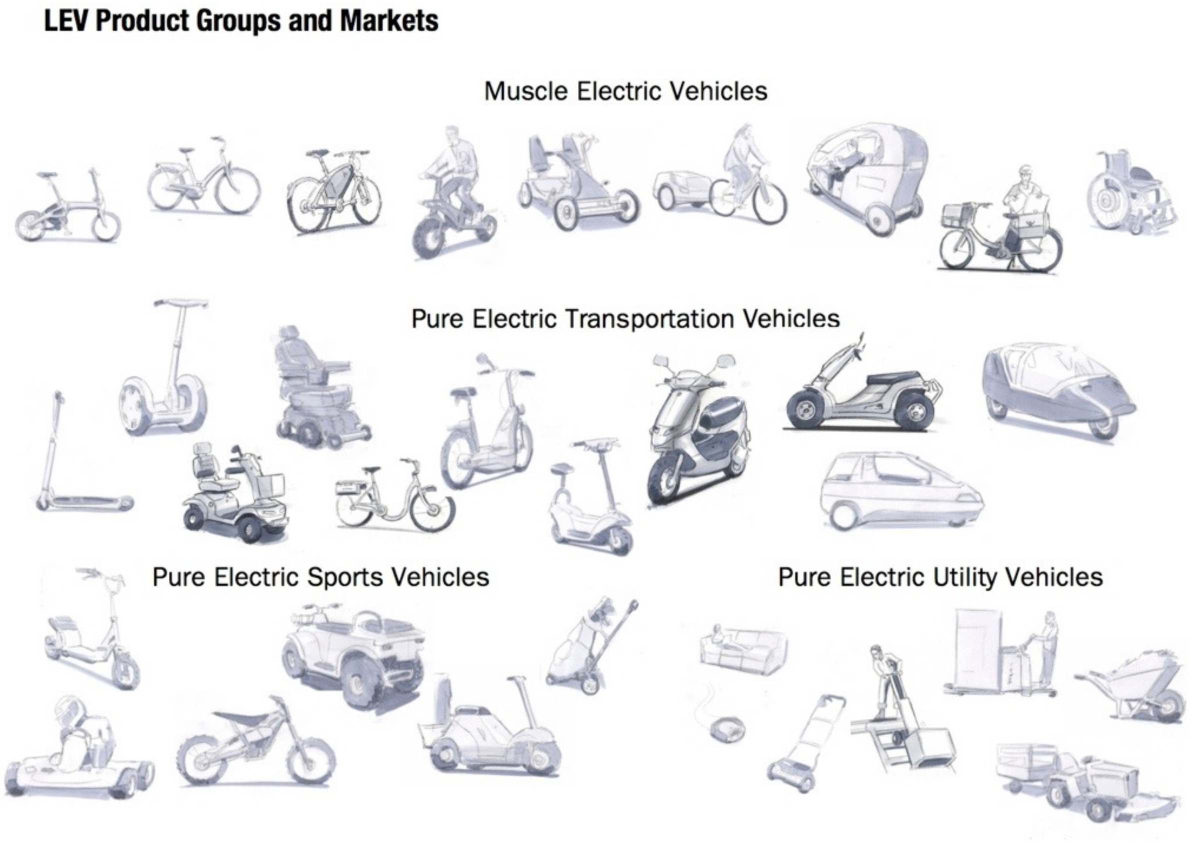
Missing state change during LSS configuration

```
process CONFIGURE[LSS:LSS_CHANNEL](node_id:AVAILABLE_NODE_ID) is
  LSS(COMMAND, LSS_SWITCH_STATE_GLOBAL, LSS_STATE_CONFIGURATION);
    LSS(COMMAND, LSS_CONFIGURE_NODE_ID, node_id);
      LSS(RESPONSE, LSS_CONFIGURE_NODE_ID, LSS_SUCCESSFULL);
        LSS(COMMAND, LSS_STORE_CONFIGURATION);
          LSS(RESPONSE, LSS_STORE_CONFIGURATION, LSS_SUCCESSFULL);
            LSS(COMMAND, LSS_SWITCH_STATE_GLOBAL, LSS_STATE_WAITING)
end process
```



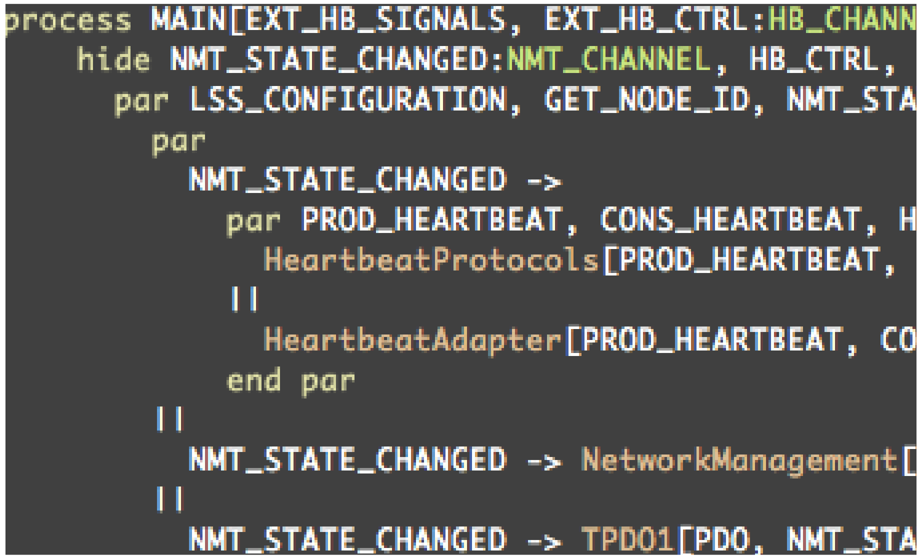
Conclusion

Energy Management Systems are increasingly relevant


LEV Product Groups and Markets
Muscle Electric Vehicles
Pure Electric Transportation Vehicles
Pure Electric Sports Vehicles
Pure Electric Utility Vehicles

Conclusion

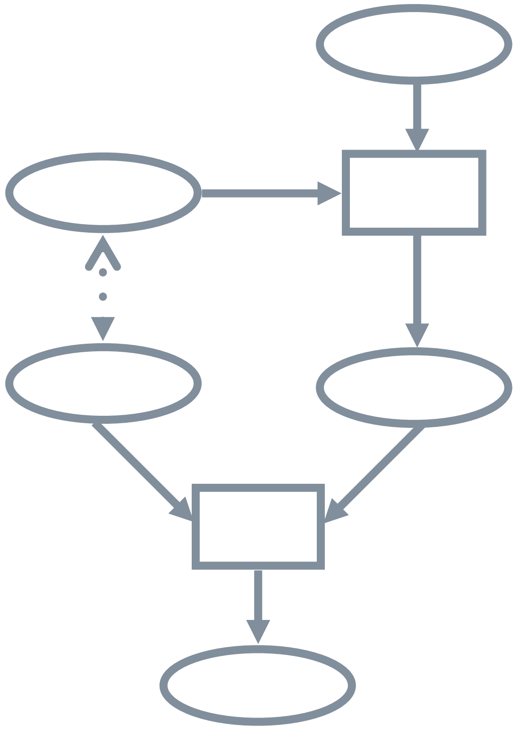
- Formal specification:
  - a sound basis for verification techniques
  - EnergyBus documentation issues found

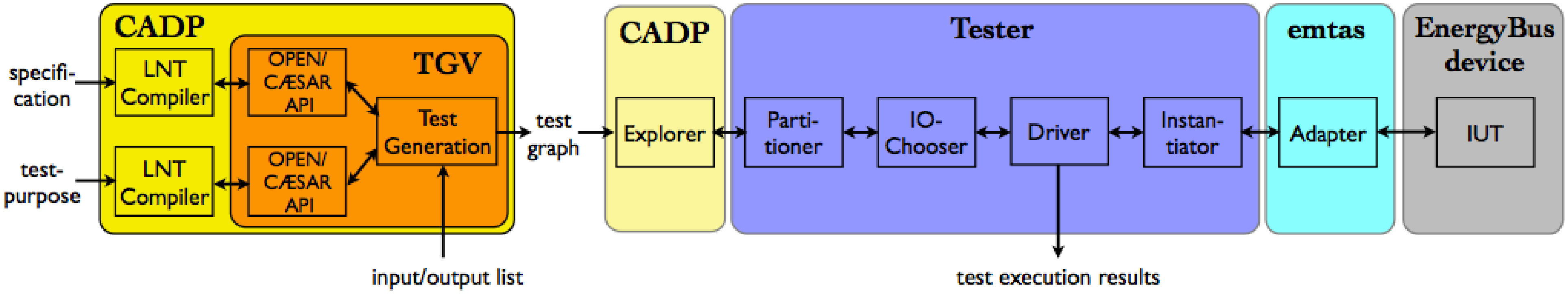


```
process MAIN[EXT_HB_SIGNALS, EXT_HB_CTRL:HB_CHANN
    hide NMT_STATE_CHANGED:NMT_CHANNEL, HB_CTRL,
      par LSS_CONFIGURATION, GET_NODE_ID, NMT_STA
        par
          NMT_STATE_CHANGED ->
            par PROD_HEARTBEAT, CONS_HEARTBEAT, H
              HeartbeatProtocols[PROD_HEARTBEAT,
            ||
              HeartbeatAdapter[PROD_HEARTBEAT, CO
            end par
        ||
          NMT_STATE_CHANGED -> NetworkManagement[
        ||
          NMT_STATE_CHANGED -> TPDO1[PDO, NMT_STA
```

Conclusion

- A certification framework for the EnergyBus:
  - Model-based testing
  - Tool setup
  - Three issues detected in the CANopen layer





Future Work

- Extending the formal model
  - **Charging Protocol**
  - power-related model components
  - virtual devices
- Further approaches against state-space explosion
  - abstract from CANopen layer
  - compositional model reductions based on bisimulations
  - TGV successor supporting online test case generation [INRIA]
  - **motest:** online model-based tester [Saarland University]
- Applying further verification techniques