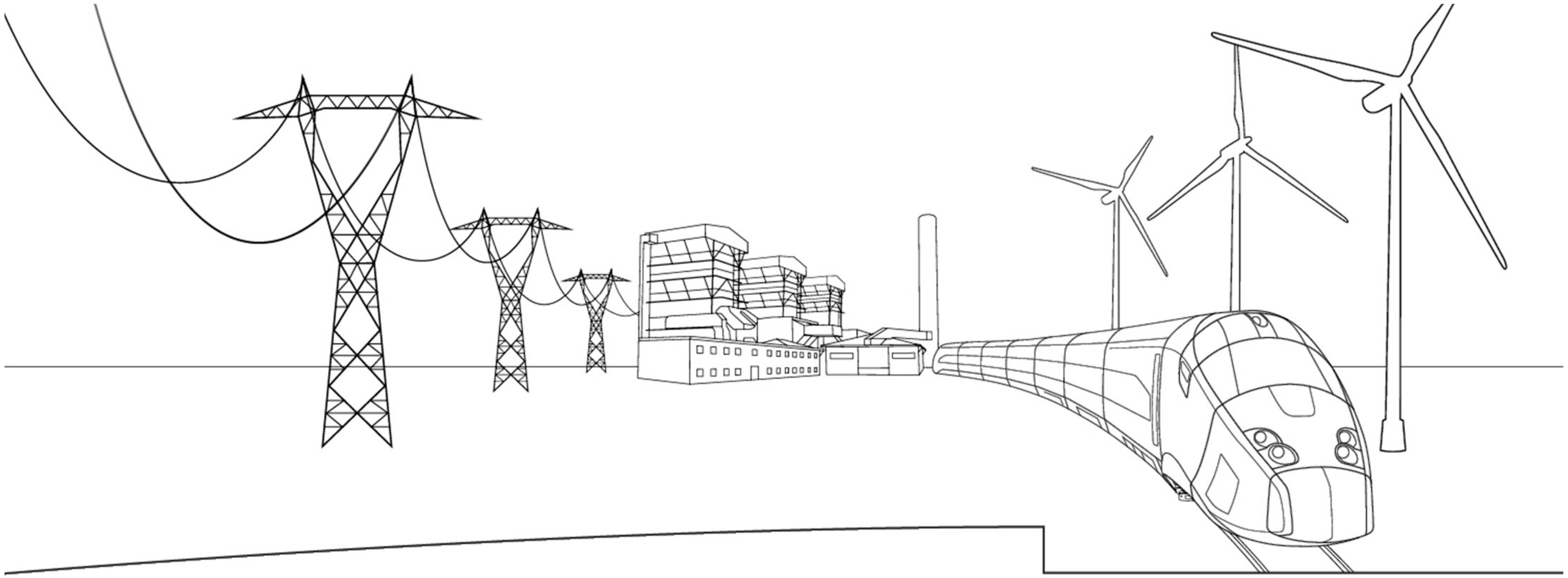


Open-ETCS API

N. Boverie

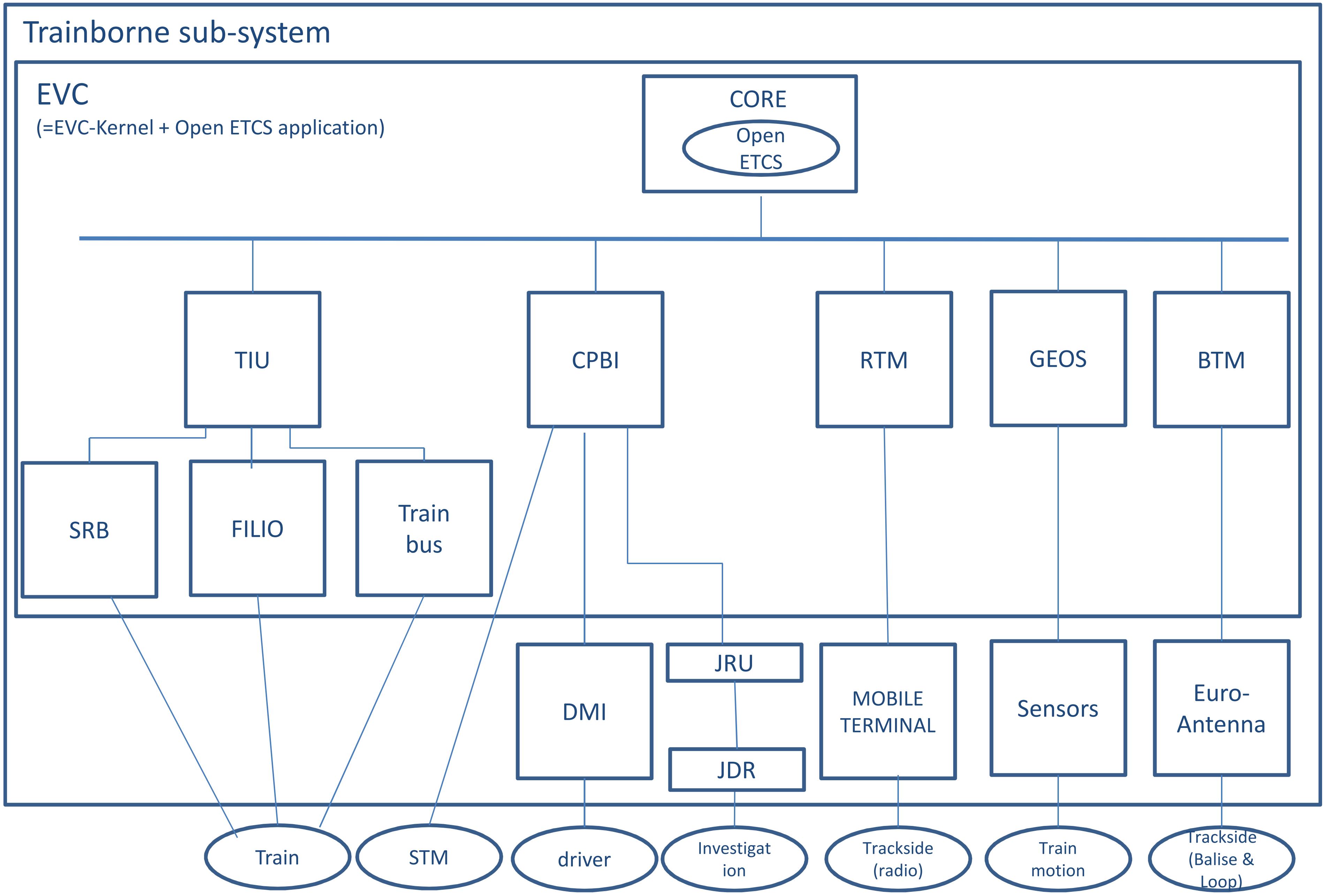
TIS Charleroi

22/10/2013

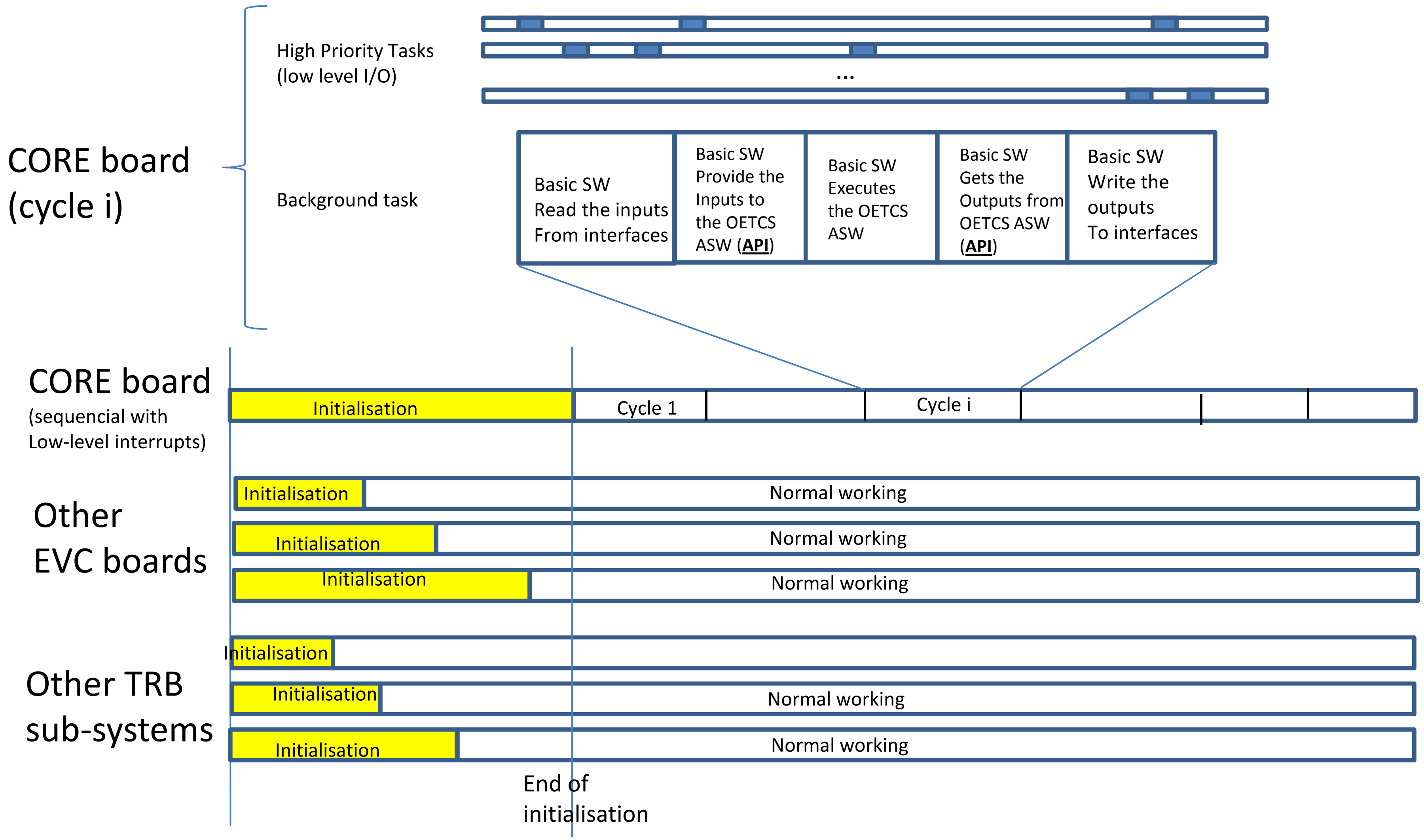
Agenda

1. Overall architecture and principles
2. CORE board SW architecture (API)
3. Functionnal interfaces list
4. API services (some examples):
    - TIME
    - MMU (Movement Measurement Unit)
    - Euro Balise
    - Euro Radio
    - DMI/JRU (control messages only)

Overall architecture & principles



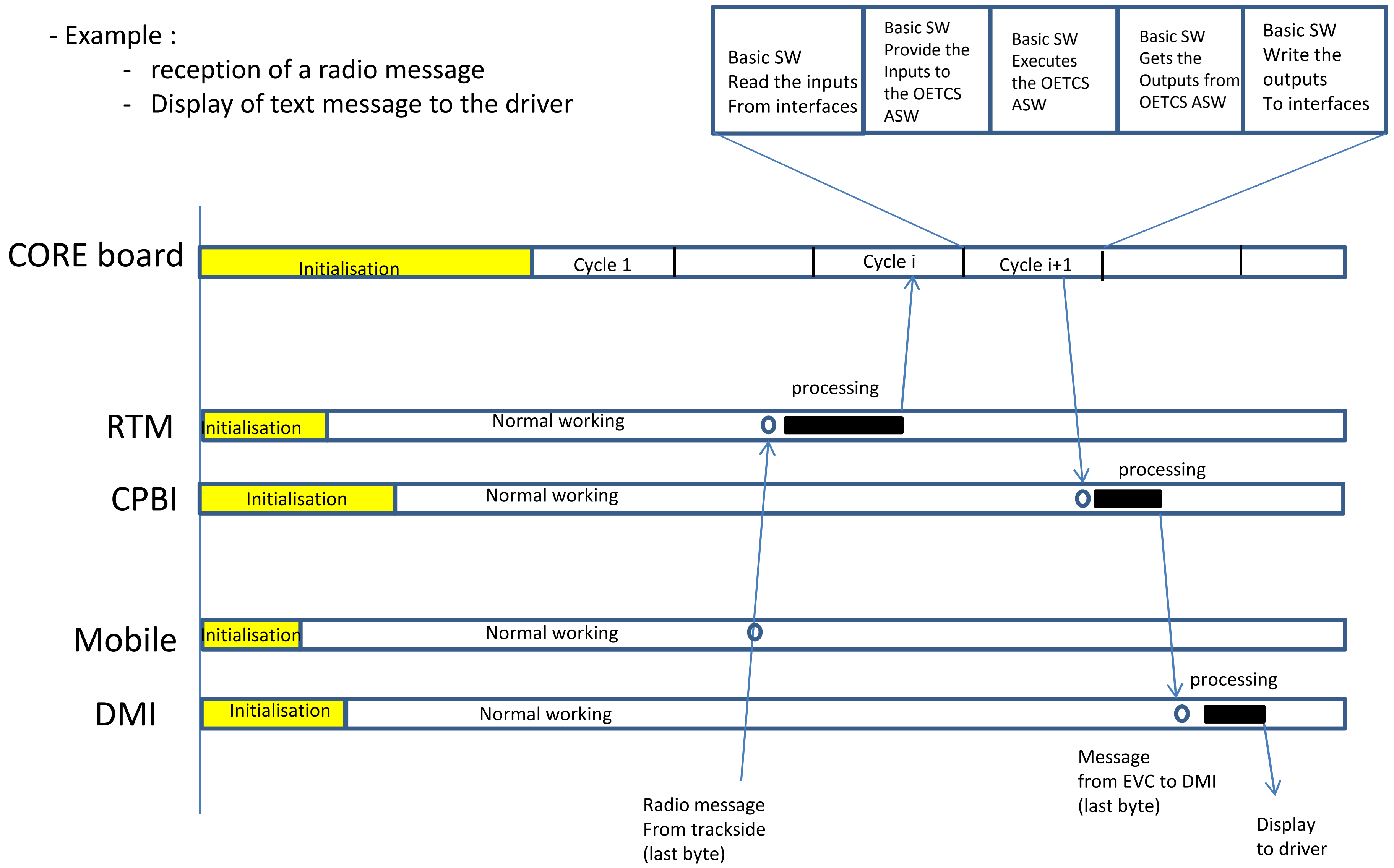
Overall architecture & principles



* The duration of the CORE board cycle does not have to be constant (typically 300ms on Alstom EVC; but can be higher in peaks). The application shall manage variable cycle duration.
* The maximum duration of the CORE cycle (and therefore the maximum duration of the application too) shall be compliant to Unisig Subset-41 performance constraints.

Overall architecture & principles

- Example :
  - reception of a radio message
  - Display of text message to the driver


Basic SW Read the inputs From interfaces
Basic SW Provide the Inputs to the OETCS ASW
Basic SW Executes the OETCS ASW
Basic SW Gets the Outputs from OETCS ASW
Basic SW Write the outputs To interfaces
CORE board
Initialisation
Cycle 1
Cycle i
Cycle i+1
processing
RTM
Initialisation
Normal working
processing
CPBI
Initialisation
Normal working
Mobile
Initialisation
Normal working
processing
DMI
Initialisation
Normal working
Message from EVC to DMI (last byte)
Radio message From trackside (last byte)
Display to driver

Overall architecture & principles

* The application must take into account that :

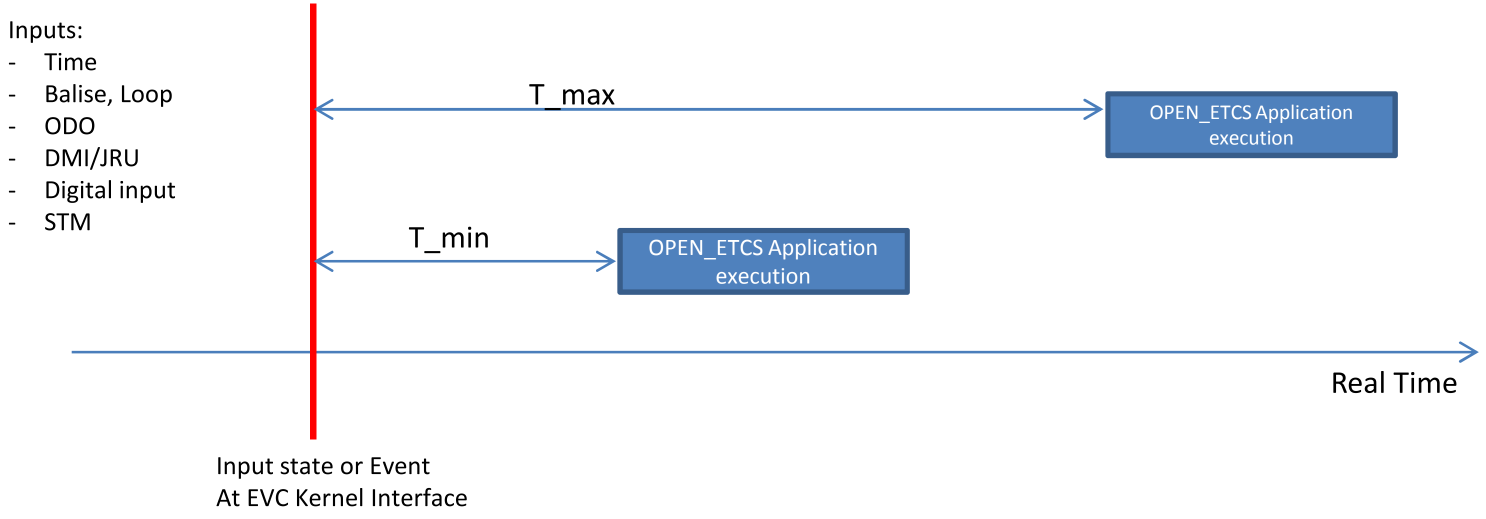
- the various peripheral boards/sub-systems in charge of the input/output treatment are not synchronised and that their performance can vary in time;
- e.g filtering of analog data such as GEOS sensors, train digital inputs, …
- e.g normal propagation durations on buses for which the performance can vary.

* Therefore,

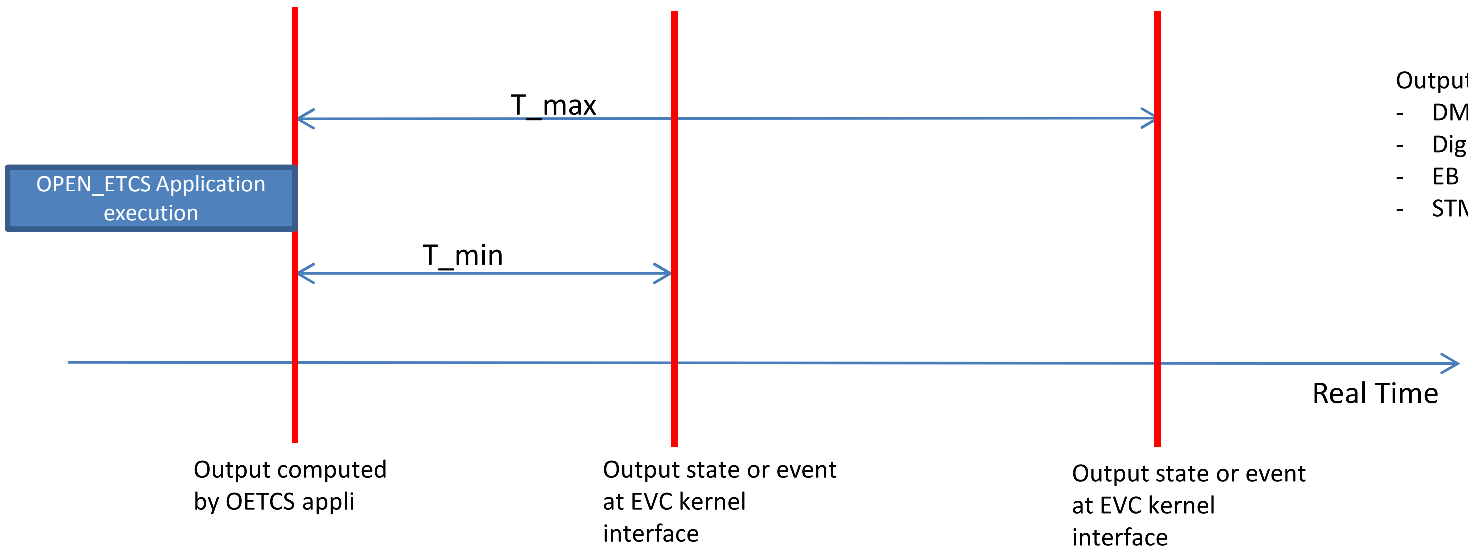
- The input data present at the interface of the EVC Kernel are provided with a variable delay to the application;
- The application output data are provided to the EVC Kernel interface with a variable delay.

For each input and for each output data, a T_min (minimum delay of EVC Kernel input/output treatment) and a T_max (maximum delay of EVC Kernel input/output treatment) will be defined.

=> TIME STAMPING of the data will allow accurate treatment by the application.

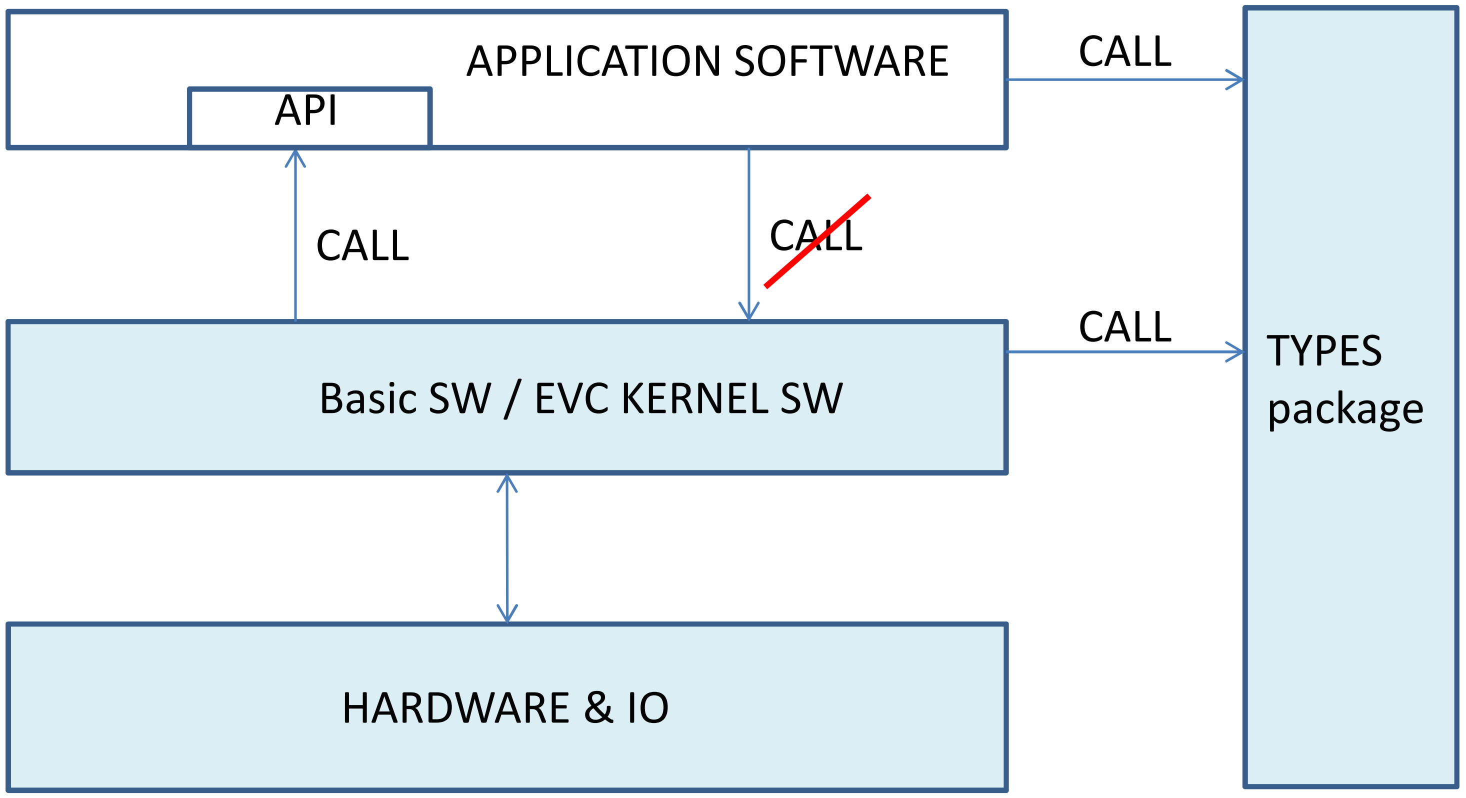


Overall architecture & principles


T_max
OPEN_ETCS Application execution
T_min
Real Time
Output computed by OETCS appli
Output state or event at EVC kernel interface
Output state or event at EVC kernel interface

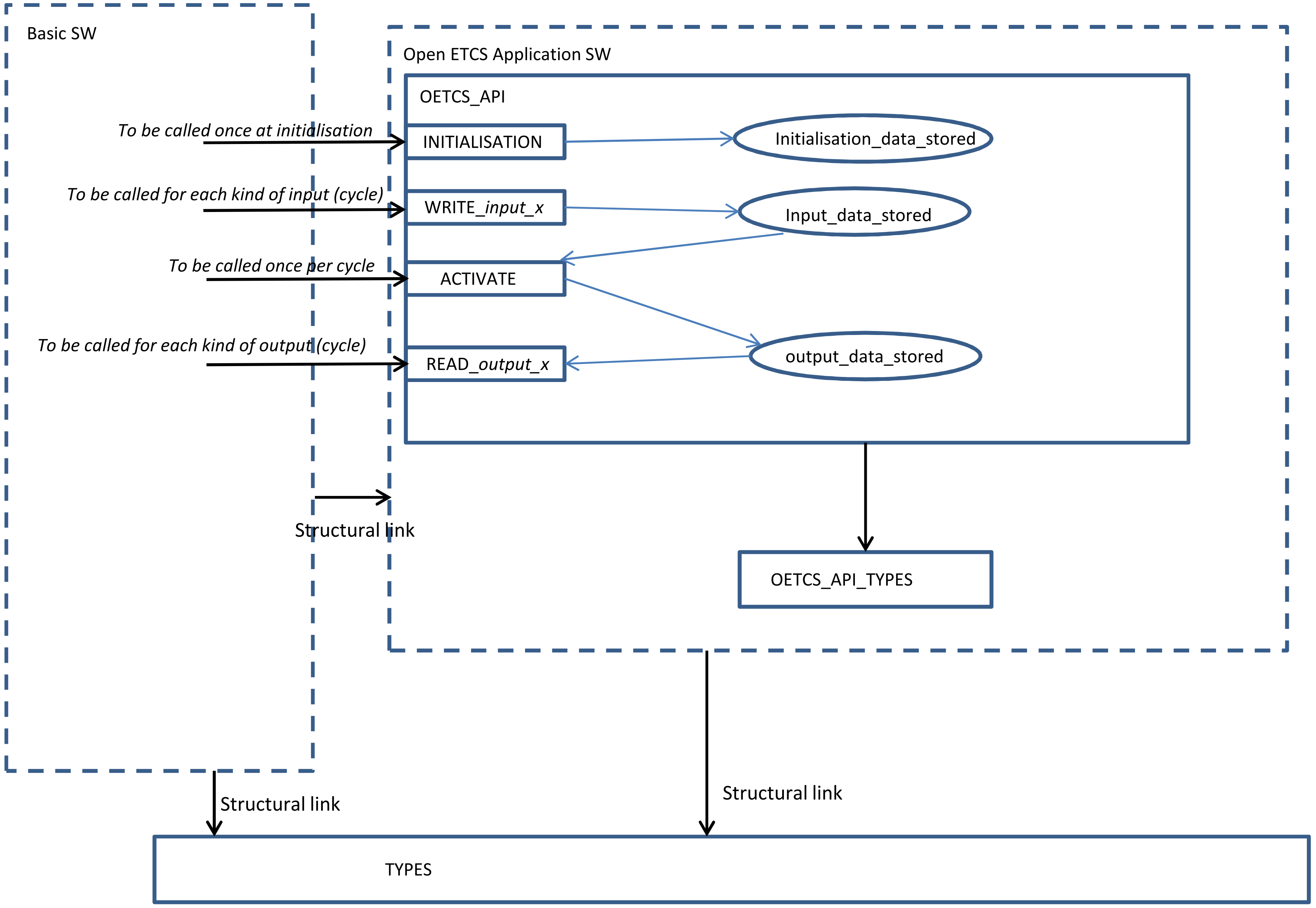
Outputs:
- DMI/JRU
- Digital output
- EB
- STM

CORE board SW architecture (API)

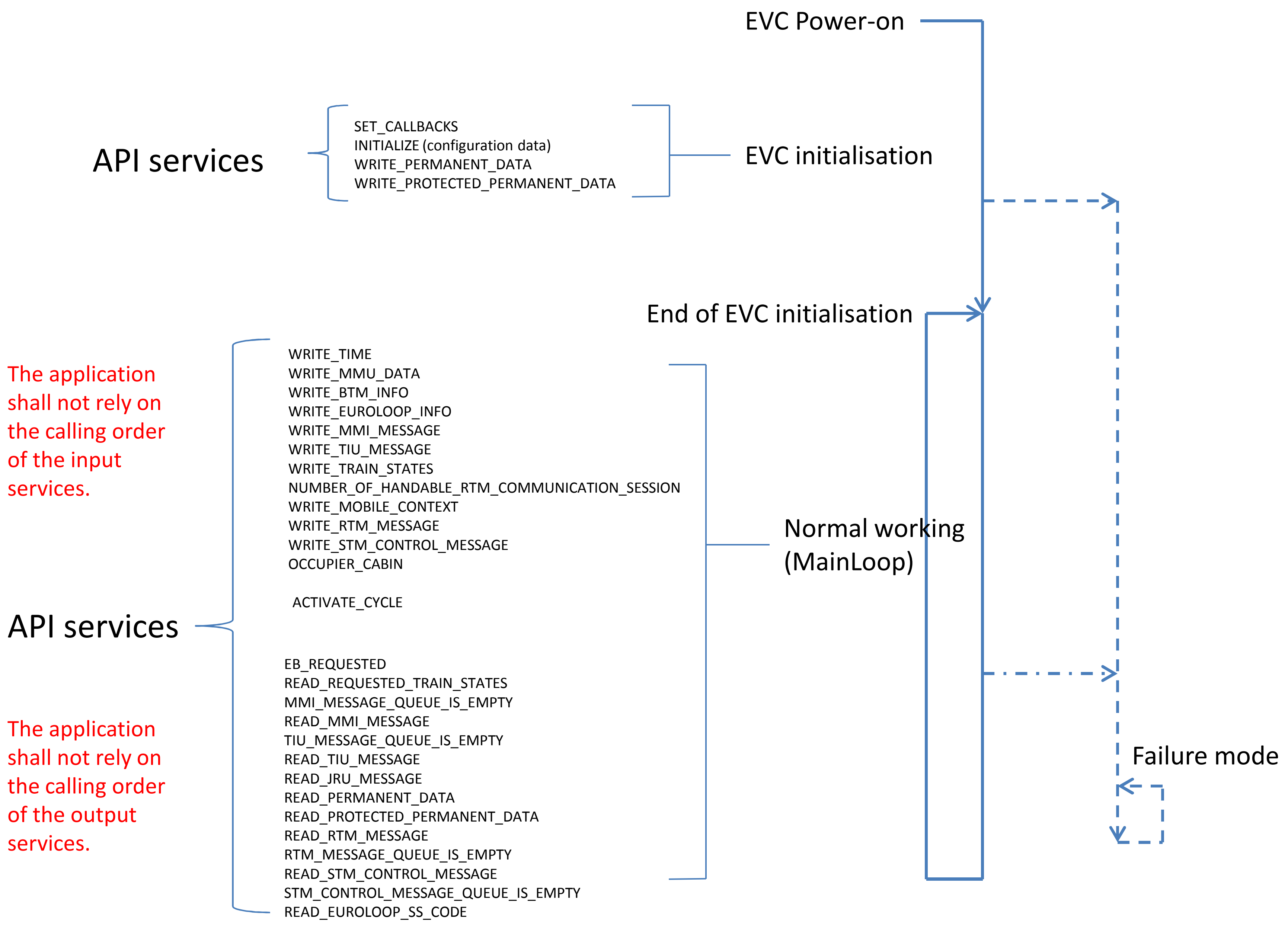


The application SW has minimum dependancies with other SW components : ONLY the TYPES package may be called by the application SW.
This SW Architecture allows easy portability of the Application Software on several platforms (EVC, PC, others, ..).

CORE board SW architecture (API)



CORE board SW architecture (normal calling sequence)



API services

SET_CALLBACKS
INITIALIZE (configuration data)
WRITE_PERMANENT_DATA
WRITE_PROTECTED_PERMANENT_DATA

API services

The application shall not rely on the calling order of the input services.

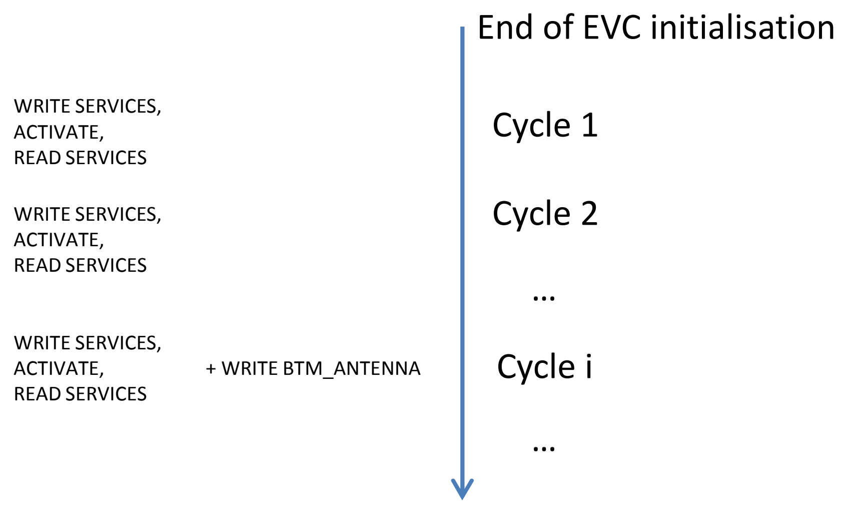
WRITE_TIME
WRITE_MMU_DATA
WRITE_BTM_INFO
WRITE_EUROLOOP_INFO
WRITE_MMI_MESSAGE
WRITE_TIU_MESSAGE
WRITE_TRAIN_STATES
NUMBER_OF_HANDABLE_RTM_COMMUNICATION_SESSION
WRITE_MOBILE_CONTEXT
WRITE_RTM_MESSAGE
WRITE_STM_CONTROL_MESSAGE
OCCUPIER_CABIN

ACTIVATE_CYCLE

The application shall not rely on the calling order of the output services.

EB_REQUESTED
READ_REQUESTED_TRAIN_STATES
MMI_MESSAGE_QUEUE_IS_EMPTY
READ_MMI_MESSAGE
TIU_MESSAGE_QUEUE_IS_EMPTY
READ_TIU_MESSAGE
READ_JRU_MESSAGE
READ_PERMANENT_DATA
READ_PROTECTED_PERMANENT_DATA
READ_RTM_MESSAGE
RTM_MESSAGE_QUEUE_IS_EMPTY
READ_STM_CONTROL_MESSAGE
STM_CONTROL_MESSAGE_QUEUE_IS_EMPTY
READ_EUROLOOP_SS_CODE

CORE board SW architecture (Service called “on event”)


End of EVC initialisation
WRITE SERVICES, ACTIVATE, READ SERVICES
Cycle 1
WRITE SERVICES, ACTIVATE, READ SERVICES
Cycle 2
…
WRITE SERVICES, ACTIVATE, READ SERVICES
+ WRITE BTM_ANTENNA
Cycle i
…

Functionnal interfaces list (1/4)

API data		IN	OUT
EVC-Kernel time		X	
MMU (odometry)	MMU data (Position, Speed, acceleration, Direction, Motion, time-stamp)	X	
Eurobalise	Balise message (without coding strategy)	X	
	Antenna to be used and modulation		X
	Metal mass info		X
Euroloop	euroloop data message	X	
	Spread Spectrum code for euroloop		X
Radio	Radio data message received	X	
	Radio emergency data message	X	
	Radio/network control message received	X	
	Radio data message to be sent		X
	Radio/network control message to be sent		X
	number of handable radio sessions	X	
	train is in a radio hole		X
	mobile context (state & network_id)	X	

Functionnal interfaces list (2/4)

API data		IN	OUT
DMI	DMI data or control received	X	
	DMI data or control to be sent		X
	DMI driver selected language		X
JRU	JRU data or JRU control received	X	
	JRU data or JRU control to be sent		X
STM	STM info (NID_STM, STM state)	X	
	STM data or STM control received	X	
	STM data or STM control to be sent		X

Functionnal interfaces list (3/4)

API data		IN	OUT
Train interface inputs (filtered & not-filtered)	EMERGENCY_BRAKE	X	
	SERVICE_BRAKE	X	
	TRACTION_CUT_OFF	X	
	ISOLATION	X	
	SLEEPING	X	
	TILTING	X	
	DRIVER_DIRECTION	X	
	DESK	X	
	INTEGRITY	X	
	DRIVER_EMERGENCY	X	
	DRIVER_VIGILANCE	X	
	VIGILANCE_SYSTEM	X	
	COLD_MOVEMENT	X	
Train interface outputs	SB_COMMAND		X
	EB_COMMAND		X
	CUT_OFF_TRACTION		X
	DISABLE_DRIVER_VIGILANCE		X
EB request			X
EVC isolation is requested	ASW ask to shutdown EVC		

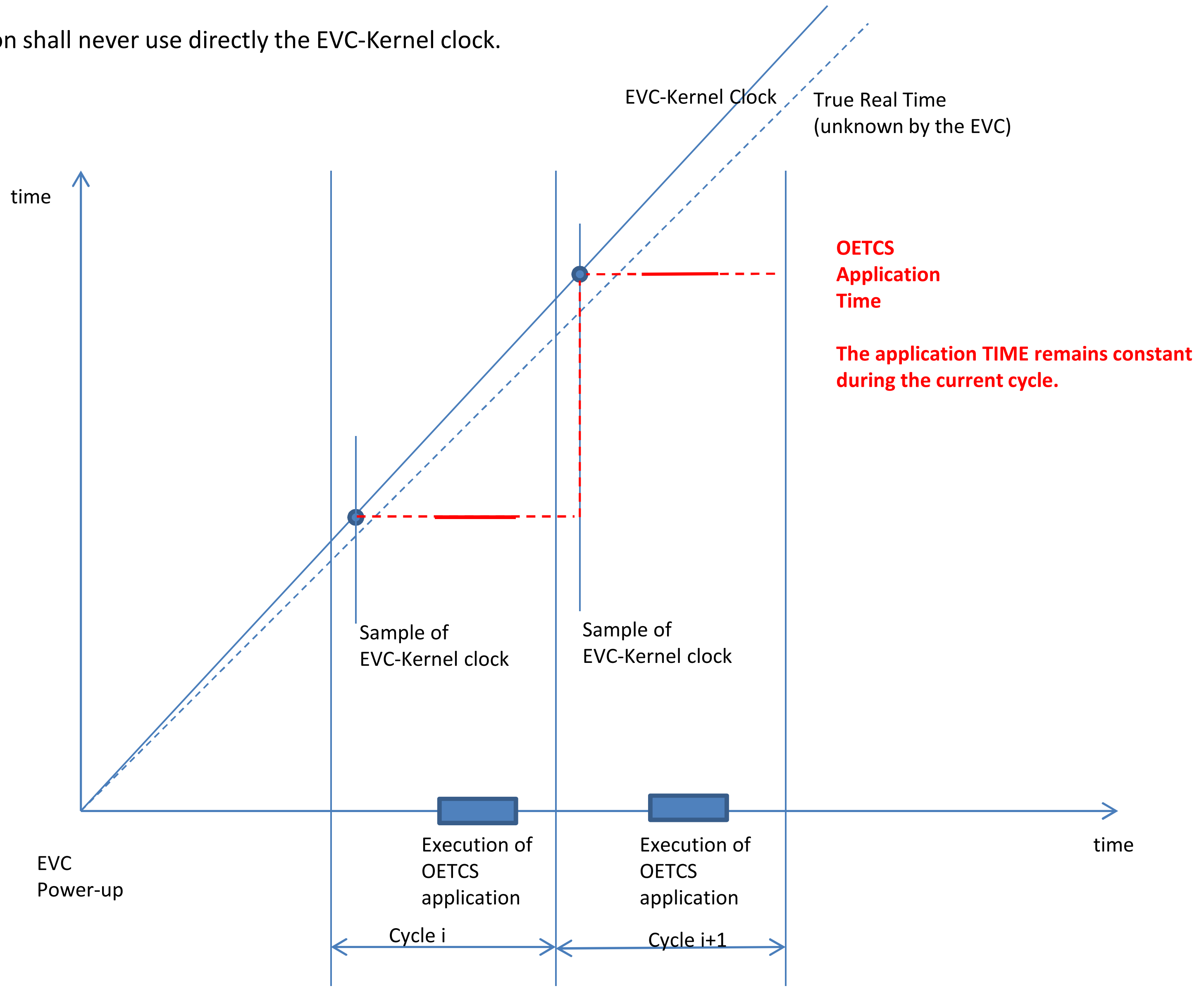
Functionnal interfaces list (4/4)

API data		IN	OUT
Configuration data		X	
ETCS id		X	
permanent data		X	X
permanent data network id		X	
protected permanent data		X	X

API : TIME

* The time provided to the application is equal to 0 at power-up of the EVC, then increases at each cycle (unit = 0,01s), until it reaches its maximum value (EVC-Kernel limitation = 24 hours)

* The Application shall never use directly the EVC-Kernel clock.

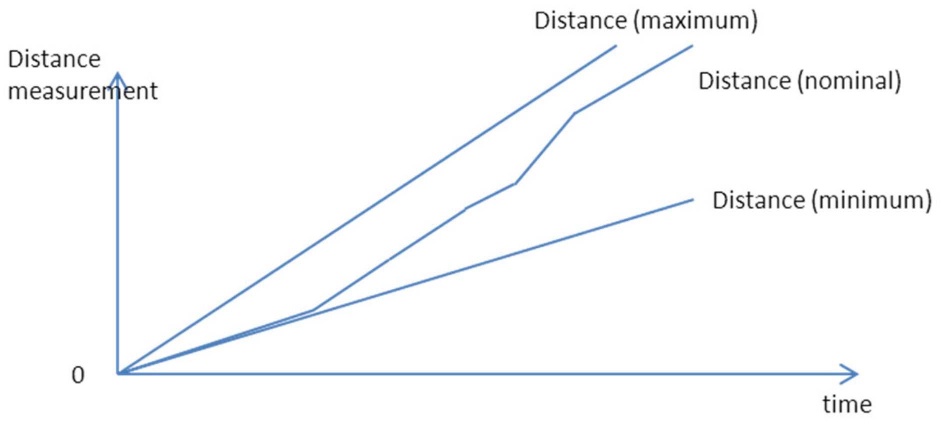


API : Movement Measurement Unit (MMU) (1/3)

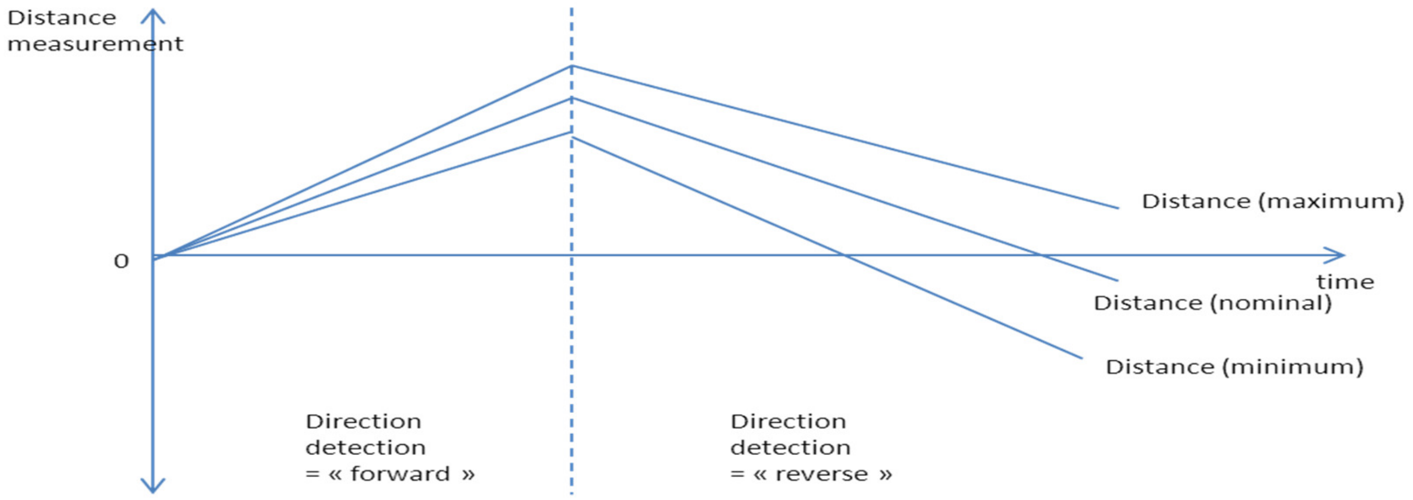
Name	Type	Direction	Description
COORDINATE	OPEN_ETCS_API_TYPES.MMU_COORDINATE_T	in	The current position of the train, bounded with the Upper and Lower values. The position given is related to the train (does not change with the selected cabin). For motions where cab A cross a point of the track before cab B, the location increases into the positive. For motions where cab B cross a point of the track before cab A, the location decreases into the negative. [m]
SPEED	OPEN_ETCS_API_TYPES.BOUNDED_SPEED_T	in	The current speed of the train, bounded by the Upper and Lower values. The speed is always positive, ranging from 0 to 600 km/h [m/s]
ACCELERATION	OPEN_ETCS_API_TYPES.ACCELERATION_VALUE_T	in	Acceleration of the train [m/s^2]
MOTION_STATE	OPEN_ETCS_API_TYPES.MOTION_STATE_T	in	Indicate when the train is moving or not.
MOTION_DIRECTION	OPEN_ETCS_API_TYPES.MOTION_DIRECTION_T	in	Indicates the direction of the train
TIME_STAMP	OPEN_ETCS_API_TYPES.CLOCK_T	in	The time at which the measure was valid

API : Movement Measurement Unit (MMU) (2/3)

The position of the train is always 0 at the power-up of the EVC and it corresponds to the head of the train. The nominal distance will always be in the range of the minimum distance and the maximum distance. But it can be closer to one or the other boundary. (same principle for the speed measurement).

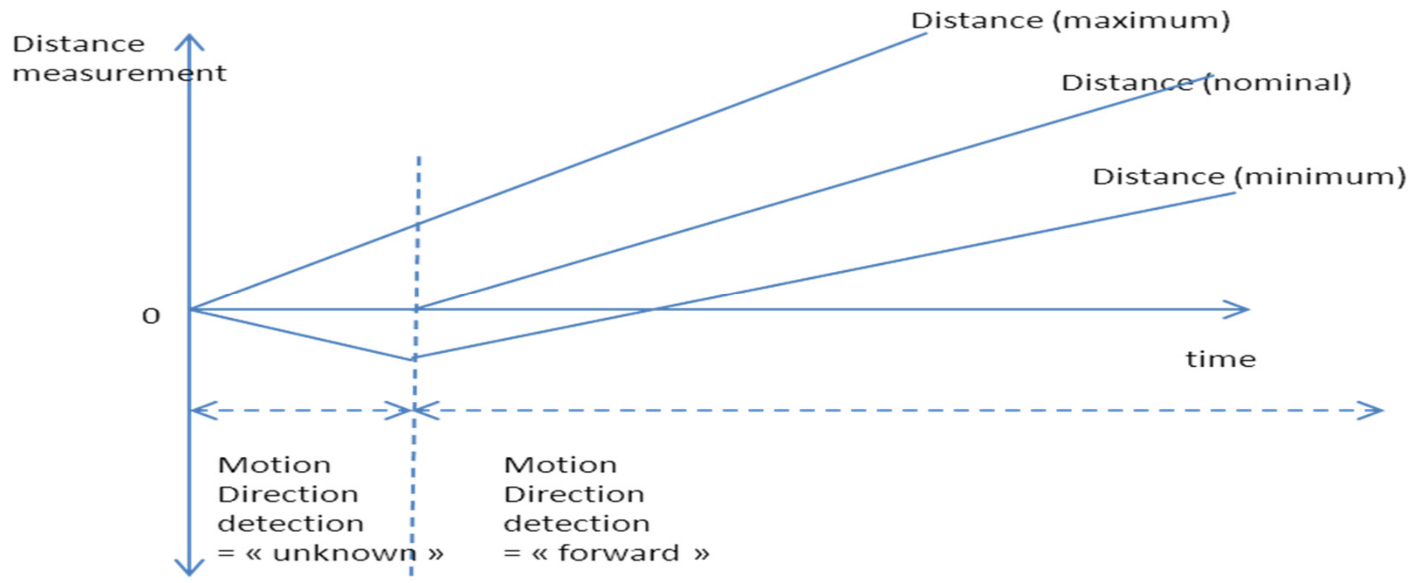


* The error range |Distance_max – Distance_min| will always increase during the time. Here-after is an example when the train is going first in forward and then after in reverse :



API : Movement Measurement Unit (MMU) (3/3)

* When the train starts moving, the Motion Direction may be set to “unknown” due to very low speed of the train (under ~1 km/h).
In such case, the distance measurement data may vary as following (e.g) :


Distance measurement
Distance (maximum)
Distance (nominal)
Distance (minimum)
0
time
Motion Direction detection = « unknown »
Motion Direction detection = « forward »

API : Euro Balise (1/2)

Name	Type	Direction	Description
THE_BTM_INFO	OPEN_ETCS_API_TYPES.BTM_INFO_T	in	The BTM message as received from the track : - balise telegram (useful data; no coding strategy) - center of balise position (nominal, min, max) - center of balise position inaccuracy - center of balise time-stamp - Id of antenna used for reception

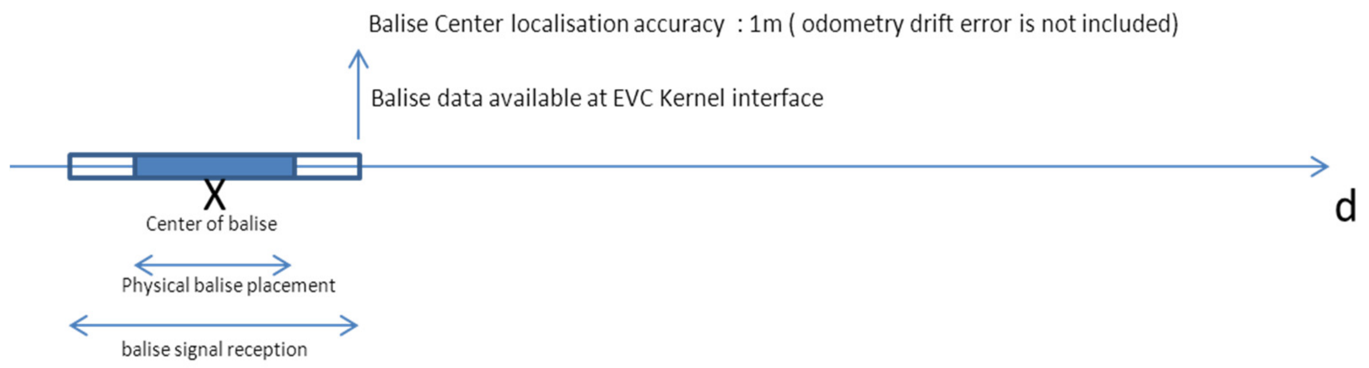
* Before the activation of the applicative software, the EVC kernel software shall call this service once per balise crossed (since the previous cycle), respecting the chronology (older balises first), with the last information received from each balise.

* This service WRITE_BTM_INFO could be called multiple times in one cycle if multiple balises have been crossed during this cycle, so the OPEN_ETCS application should bufferize the received data and achieve processing only at activation of the application.

API : Euro Balise (2/2)

* When crossing a balise ;

- invalid telegrams will be discarded (e.g wrong transmission of telegram in the airgap)
- in case of telegram change , only the last received telegram will be kept for the application. So, when crossing a balise, only one message shall be provided to the application.
- the balise telegram will be ready for the EVC Kernel only when the train antenna has reached the “end of the balise signal reception” point. As long as the train is stopped over the balise, the telegram will not be provided to the application.


Balise Center localisation accuracy : 1m (odometry drift error is not included)
Balise data available at EVC Kernel interface
X
Center of balise
Physical balise placement
balise signal reception
d

* The position of the center of the balise will be provided in the same coordinate system as the MMU distance calculation. So those data may be compared.

* The timestamp of the center of the balise will be provided in the same referential as the Write TIME service. So those data may be compared.

API : Euro Radio (1/2)

* Service READ_RTM_MESSAGE will provide to the EVC-Kernel, and to the radio sub-system, the data messages and the radio/network control messages to be sent :

- NETWORK_REQUEST : order to register the mobiles to a network. In such case, the network-id is provided by the application.
- CONNECTION_REQUEST : order to connect. In such case, the NID_RADIO (phone number) is provided by the application
- DISCONNECTION_REQUEST : order to disconnect
- RESET_CONNECTION : order to disconnect and then reconnect
- INFINITE_CONNECTION_RETRIES : order to try to reconnect infinitely after a connection loss or a reset connection
- DATA : euroradio data message from trainborne **(<u>only the useful telegram</u>**). In such case, the application provides the radio data message which is a bit stream (maximum length = 8*500 bits

API : Euro Radio (2/2)

** Service NUMBER_OF_HANDABLE_RTM_COMMUNICATION_SESSION* will provide to the application the amount of communication sessions that are possible simultaneously:

- This value is dynamic according to the hardware equipment status but once it reaches 0 it never increases anymore until the next EVC power-up.

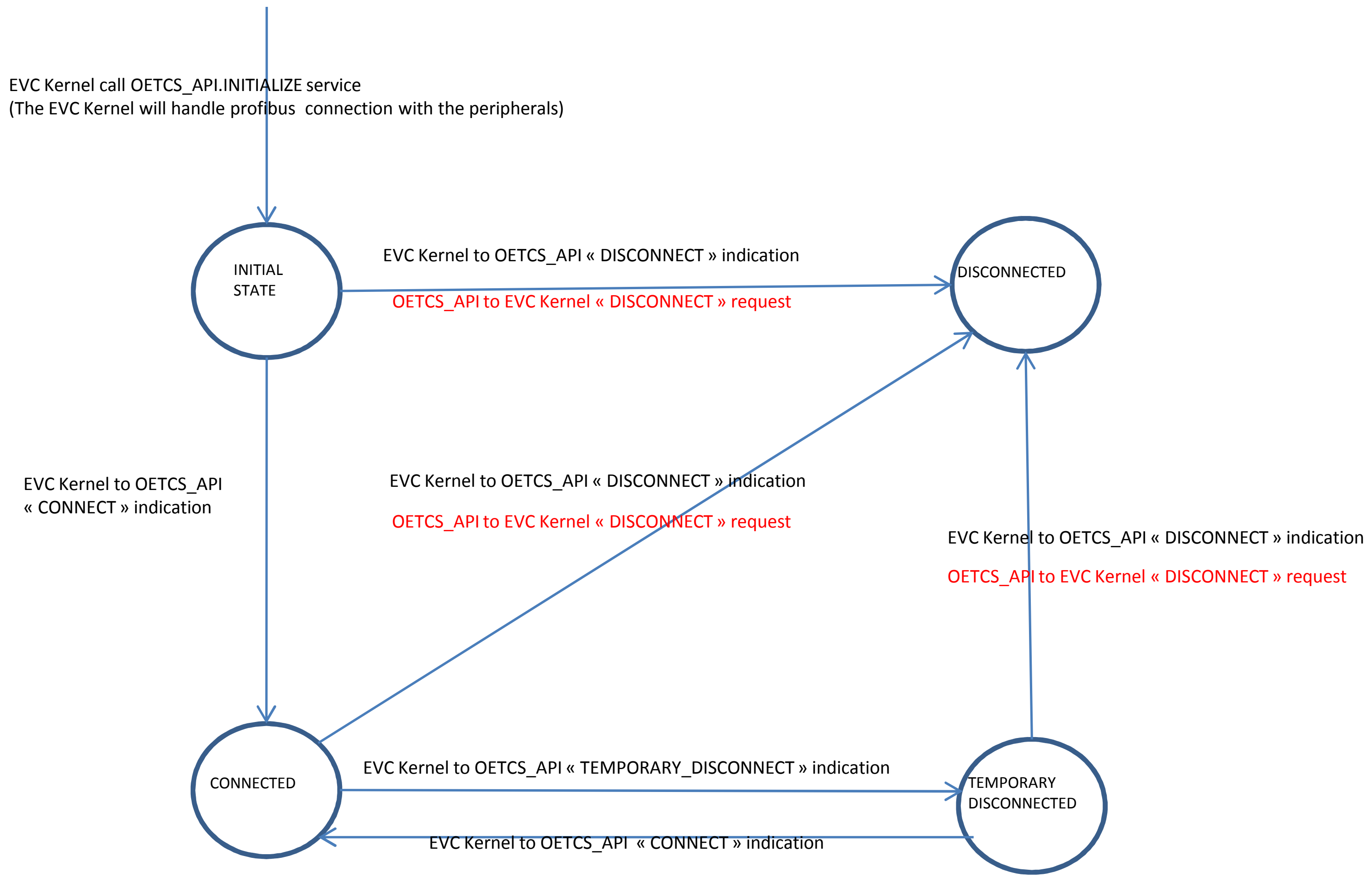
* Service WRITE_MOBILE_CONTEXT will provide to the application the status of each mobile :

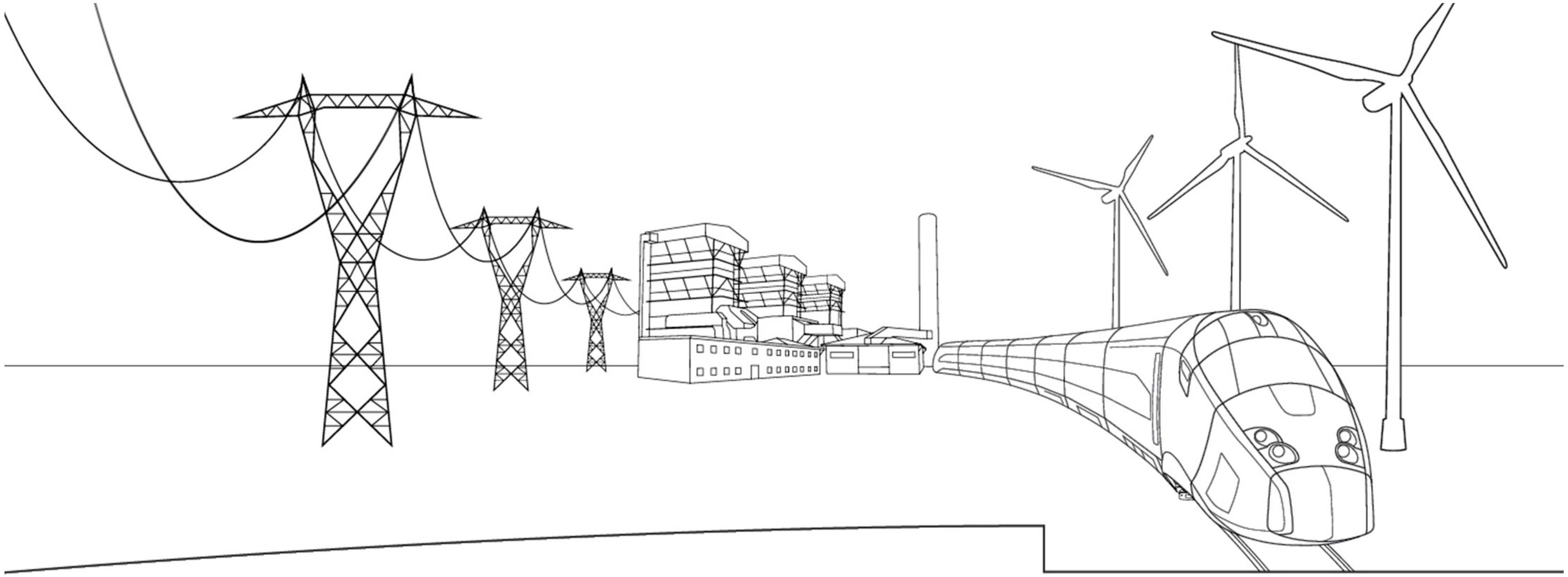
- FAILED : mobile is out of order
- NETWORK_REQUEST : mobile with a network-request in progress (NOT_REGISTERED)
- NETWORK_CONFIRM : mobile has received a network-confirm and no communication is running (REGISTERED)
- BUSY : mobile with a communication running (IN COMMUNICATION).

* Service WRITE_RTM_MESSAGE will provide to the application the received data message and the received network/radio control messages :

- CONNECTION_CONFIRMATION : provided at every connection or reconnection
- CONNECTION_LOST : provided when the connection is lost (if it was previously established)
- CONNECTION_FAILURE : provided when it has not been possible to (re)establish the connection after 3 attempts (if re-connection retries is not infinite). In such case the origin of the failure is provided by the EVC-Kernel to the application (see further in this section)
- CONNECTION_NOT_RE_ESTABLISHED : provided when it has not been possible to re-establish the connection after 3 attempts (if re-connection retries is infinite) In such case the origin of the failure is provided by the EVC-Kernel to the application (see further in this section)
- DATA : euroradio data message from trackside (<u>only the useful telegram</u>). In such case, the EVC-Kernel provides the radio data message which is a bit stream (maximum length = 8*500 bits)

API : DMI & JRU – control data


EVC Kernel call OETCS_API.INITIALIZE service
(The EVC Kernel will handle profibus connection with the peripherals)
INITIAL STATE
EVC Kernel to OETCS_API « DISCONNECT » indication
OETCS_API to EVC Kernel « DISCONNECT » request
DISCONNECTED
EVC Kernel to OETCS_API « CONNECT » indication
EVC Kernel to OETCS_API « DISCONNECT » indication
OETCS_API to EVC Kernel « DISCONNECT » request
EVC Kernel to OETCS_API « DISCONNECT » indication
OETCS_API to EVC Kernel « DISCONNECT » request
CONNECTED
EVC Kernel to OETCS_API « TEMPORARY_DISCONNECT » indication
TEMPORARY DISCONNECTED
EVC Kernel to OETCS_API « CONNECT » indication



ALSTOM

Shaping the future