ITEA 2

INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

DB Mobility Networks Logistics

openETCS

Architecture Merging Workshop

supported by:

Federal Ministry of Education and Research

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE

INVESTISSEMENTS D'AVENIR

Région de Bruxelles-Capitale

GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA E INNOVACIÓN

openETCS@ITEA2 Project

J. Gärtner (LEA/Railergy) on behalf of DB and NS

Munich, Nov 3, 2014

Purpose of these slides

openETCS has made some choices on processes and tools

- In the context of WP3/architecture, these slides are intended as a reminder for the rationale behind these decisions

openETCS has taken some architectural decisions

- These slides should help the team to align along these decisions

As we are preparing the workshop with Alstom, the slides also intend to analyze the proposals (openETCS + Alstom) in order to help align them

Objectives for WS preparation

In order to start design work in 2014, we need to align the architectures elaborated by DB/NS and Alstom during the workshop

This means that

- We must speak as one team
- The structure of the architecture document must be consistent in itself and compatible with the Alstom API
- The graphical description of the architecture must be consistent with the document
- We need a clear plan for the next steps

Reminder: openETCS tool chain

openETCS toolset (built around Papyrus)

SCADE Suite with KCG C- code generator

sysML for architecture design

SCADE Suite for software design

→ there is some possible overlap between sysML and SCADE as both use block diagrams as a structuring element

What is a model?

A description which follows well- defined principles and techniques

Different techniques for different purposes

Within one technique, potentially different views for different objectives

There are

- Nonformal models (like a “pen & paper” description)
- Semiformal models (like a SysML model)
- Formal models (like a SCADE model)

For the purpose of Systems and Software Engineering, it is best practice to combine different modeling techniques for the various objectives

Why layers?

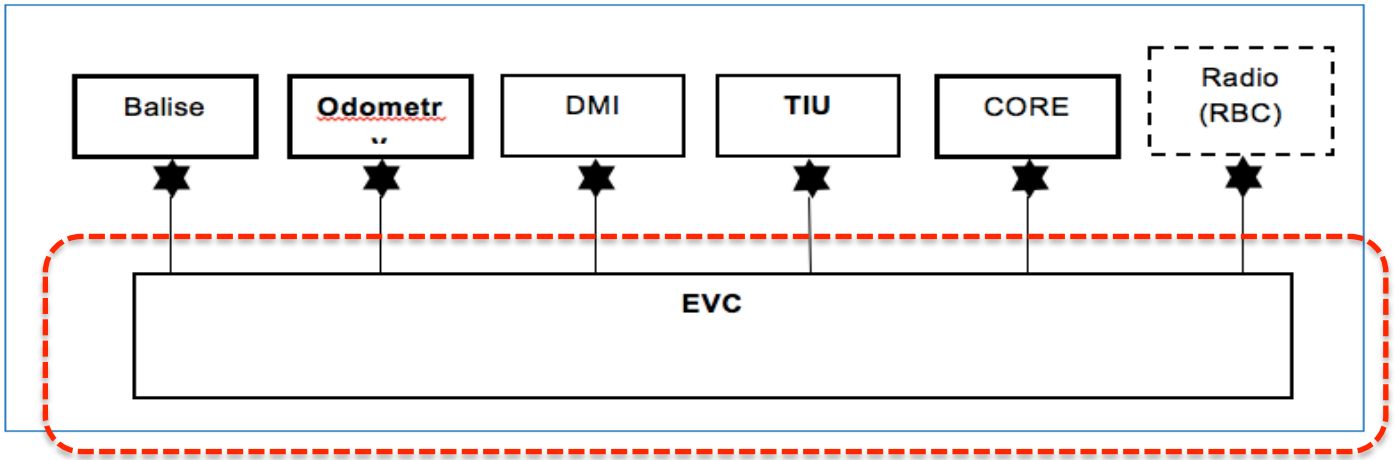
Abstraction, modularity, information hiding, independence are good SW engineering practices which are enabled by a layered approach.

The basic idea is that we can change each layer independently of the others

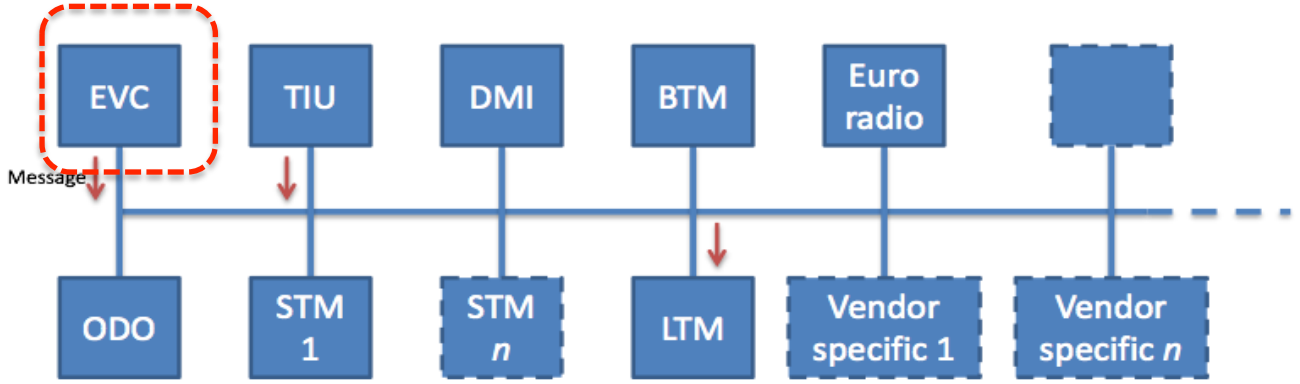
Changes on a layer that impact another layer are easily identifiable

→ **only the interface and system model layers are relevant for discussion compatibility with Alstom API**

High level HW architecture

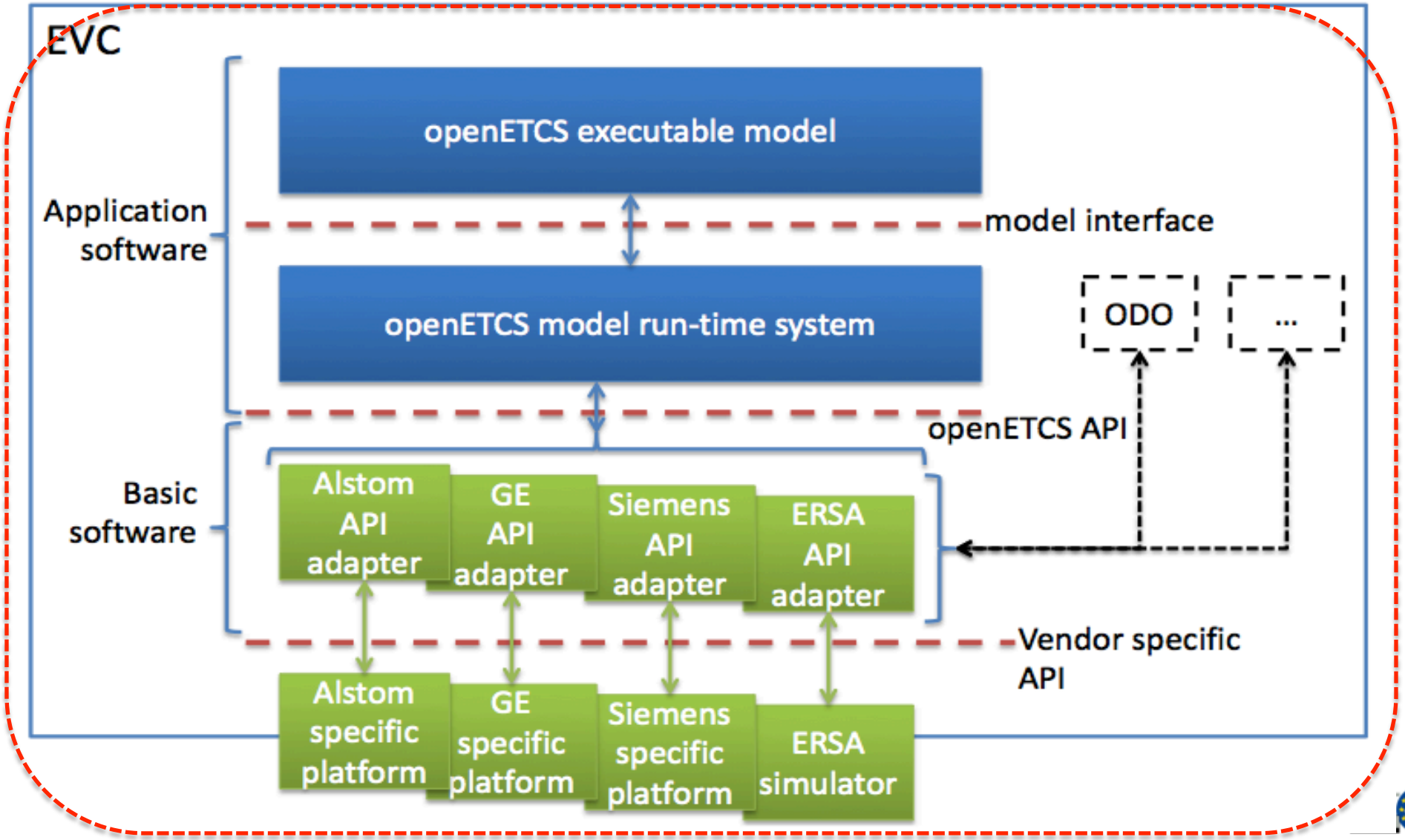


Alstom

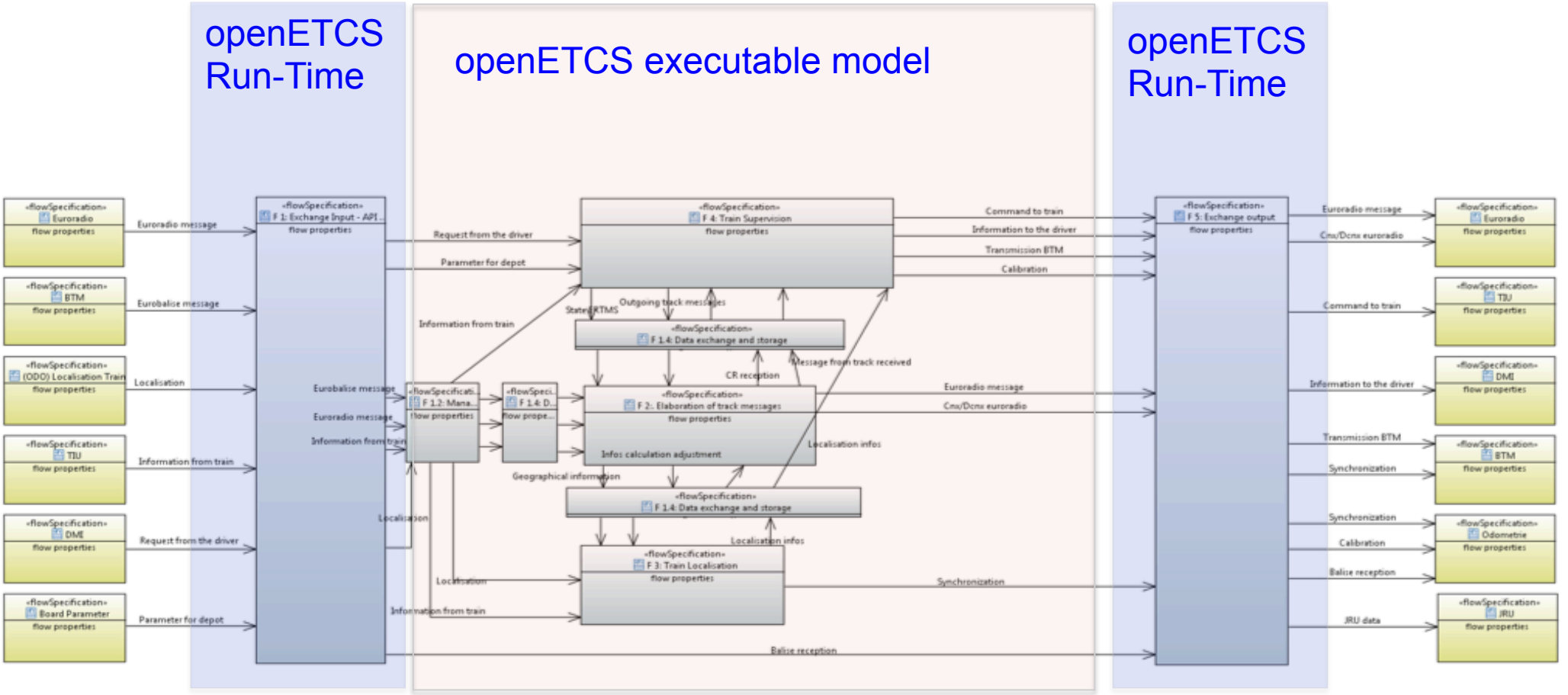


openETCS

openETCS architecture (layer view)



Open ETCS SysML view (prelim) top hierarchical level



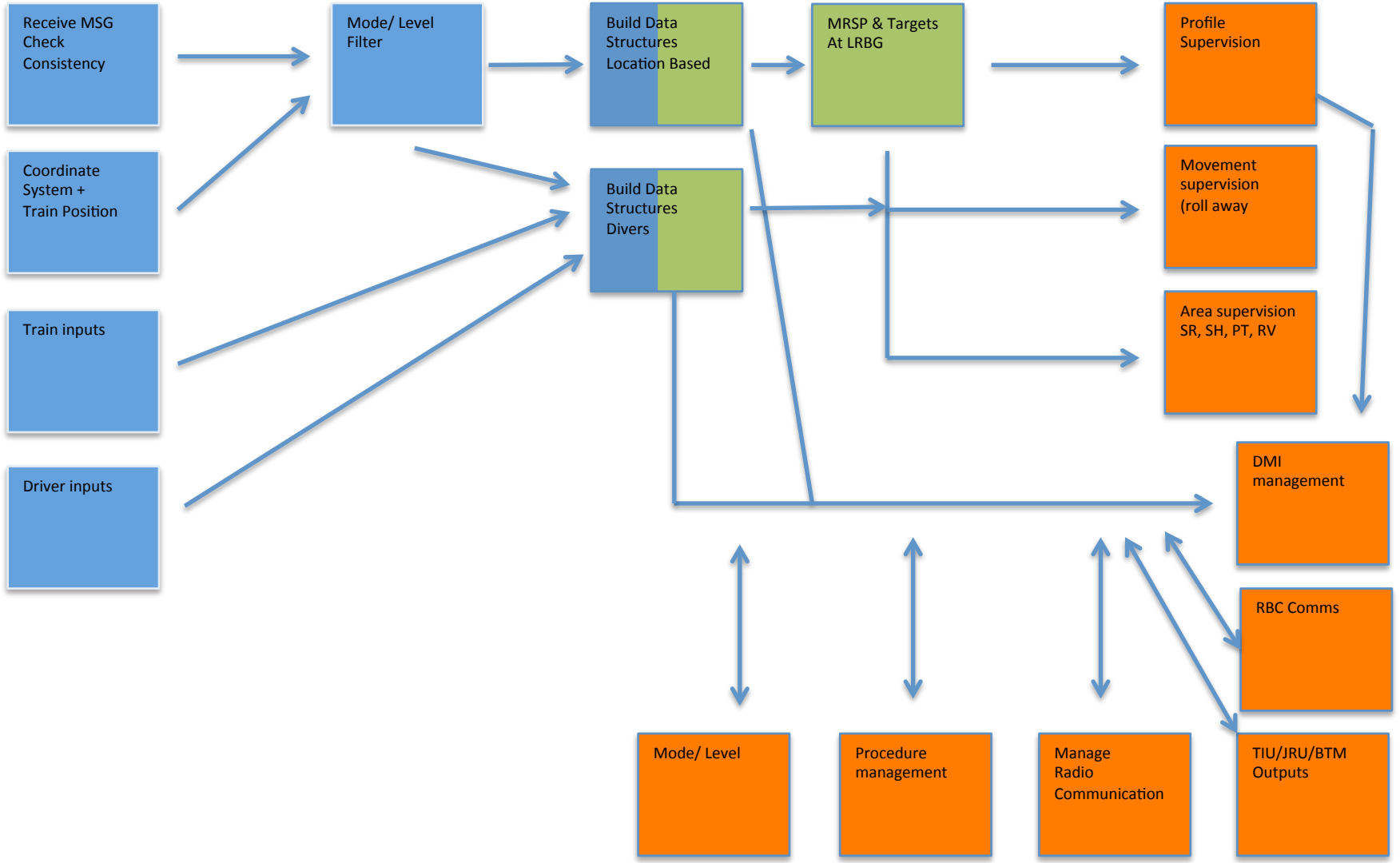
*this is an example view. We used an intermediate SysML model as illustration

Abstract functionality

- **As an abstract machine, openETCS is a typical embedded system.**
- **In order to be as efficient as possible, the three basic steps need to be implemented separately, in order to minimize the interfaces in between**
- **The key performance factor is probably an efficient implementation of the persistent database**

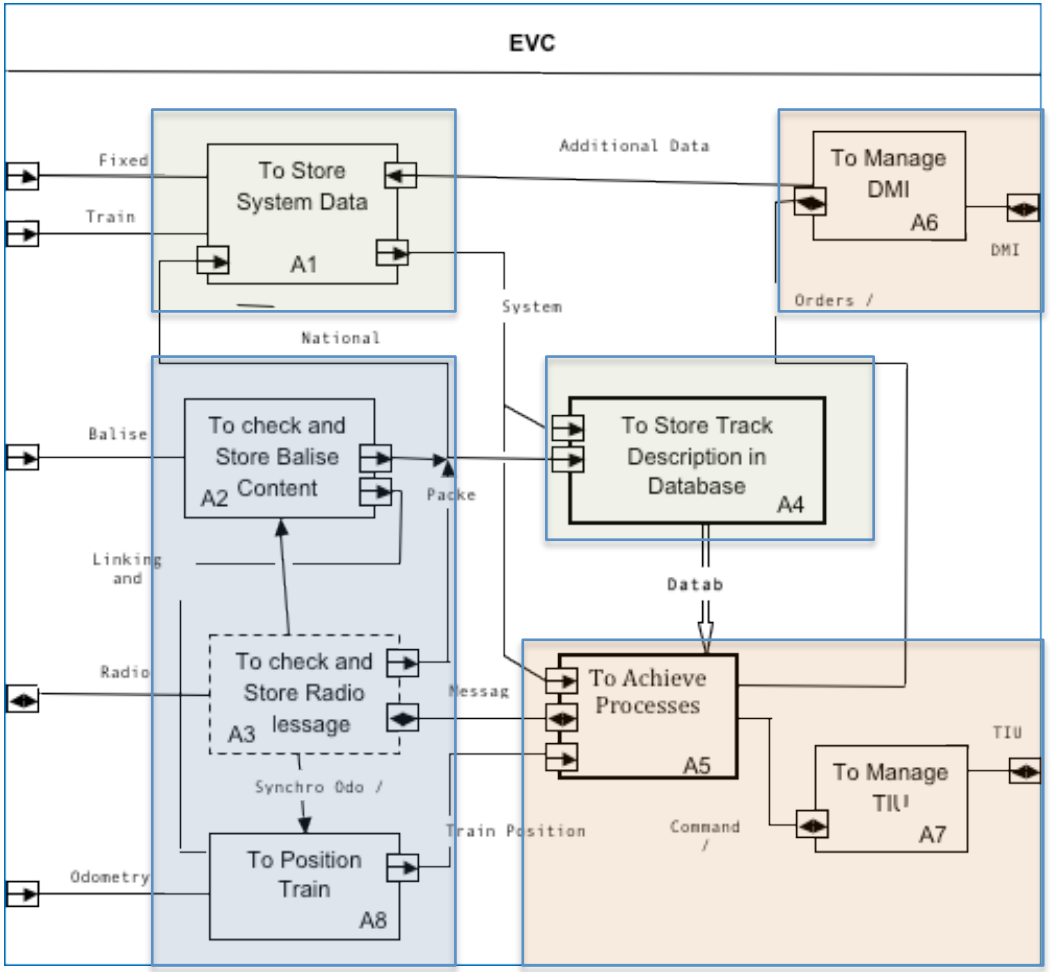
persistence is limited to time the system is on power

openETCS executable model functional view (flat)



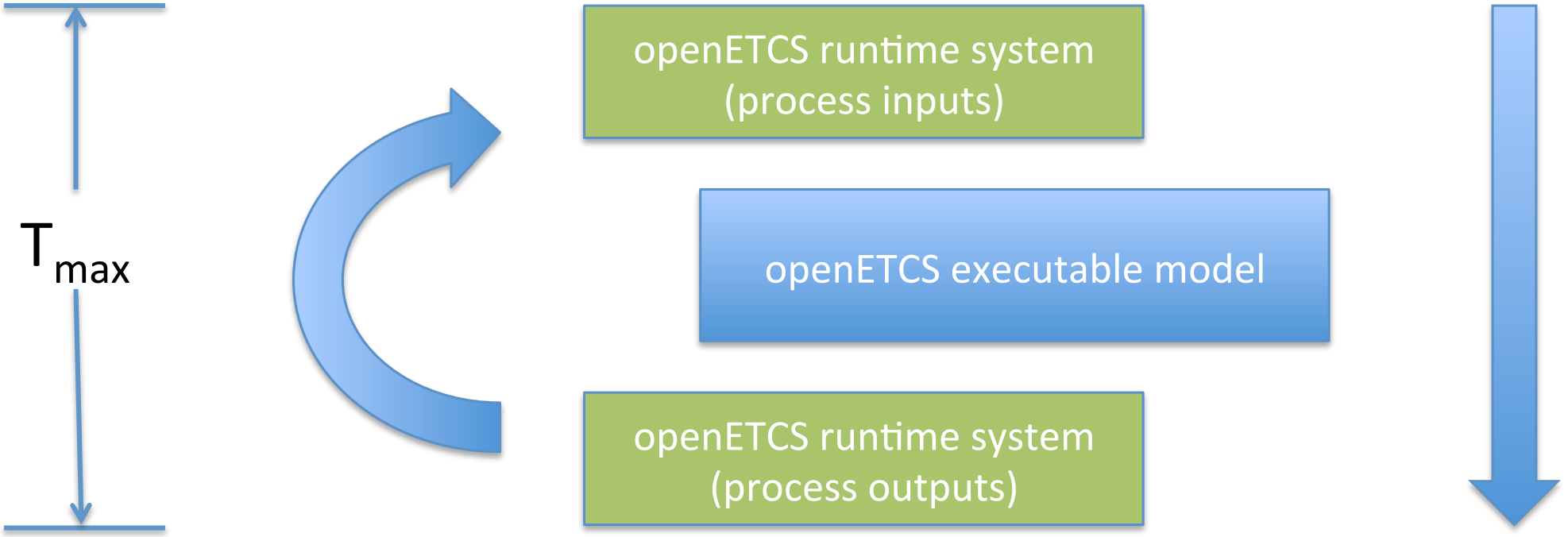
*Schematic view, no claim of completeness

Alstom high- level architecture (mapped to abstract architecture)

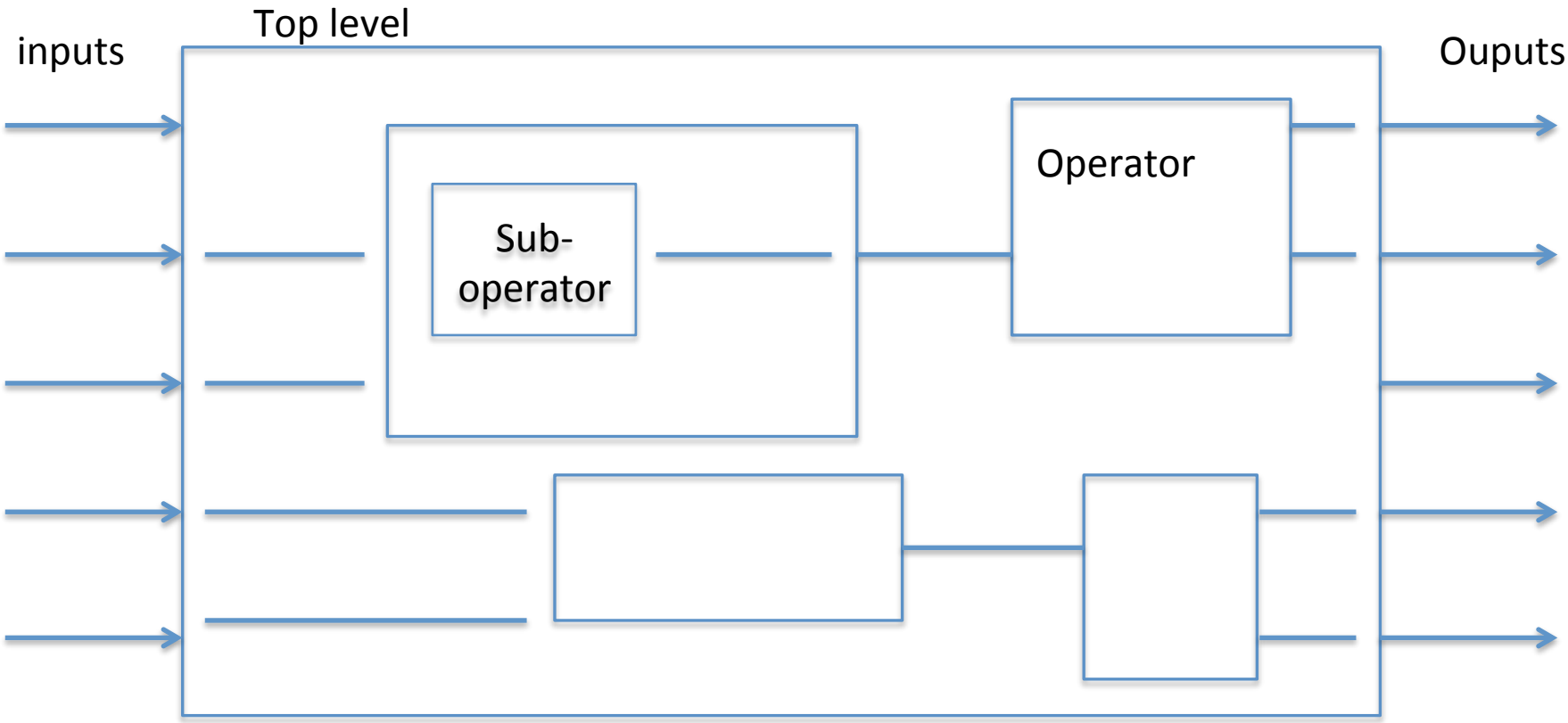


This is the top level of a hierarchical model

Cyclic execution model

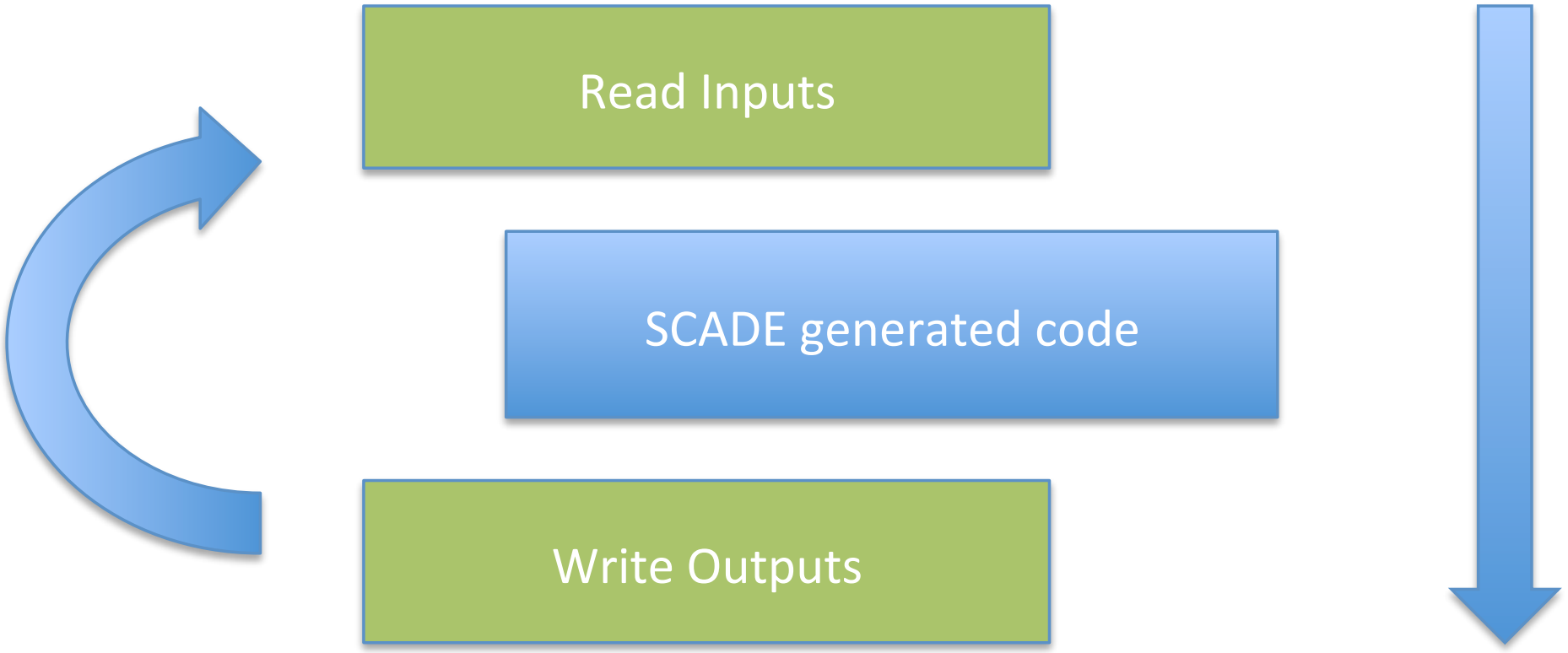


Typical SCADE model architecture

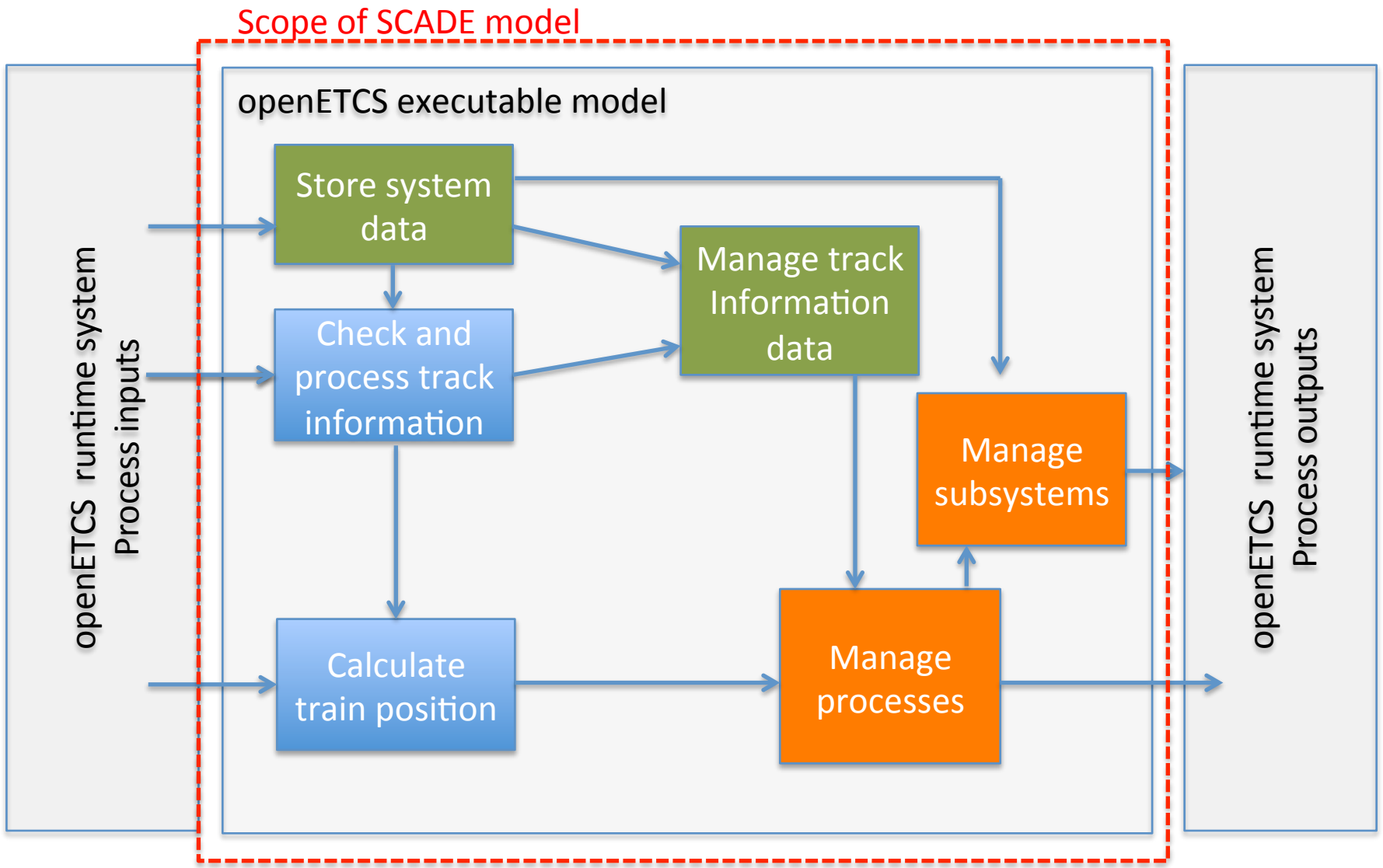


Hierarchical network of operators
For simulation and code generation, any operator can be the root
The top level is a normal operator as well

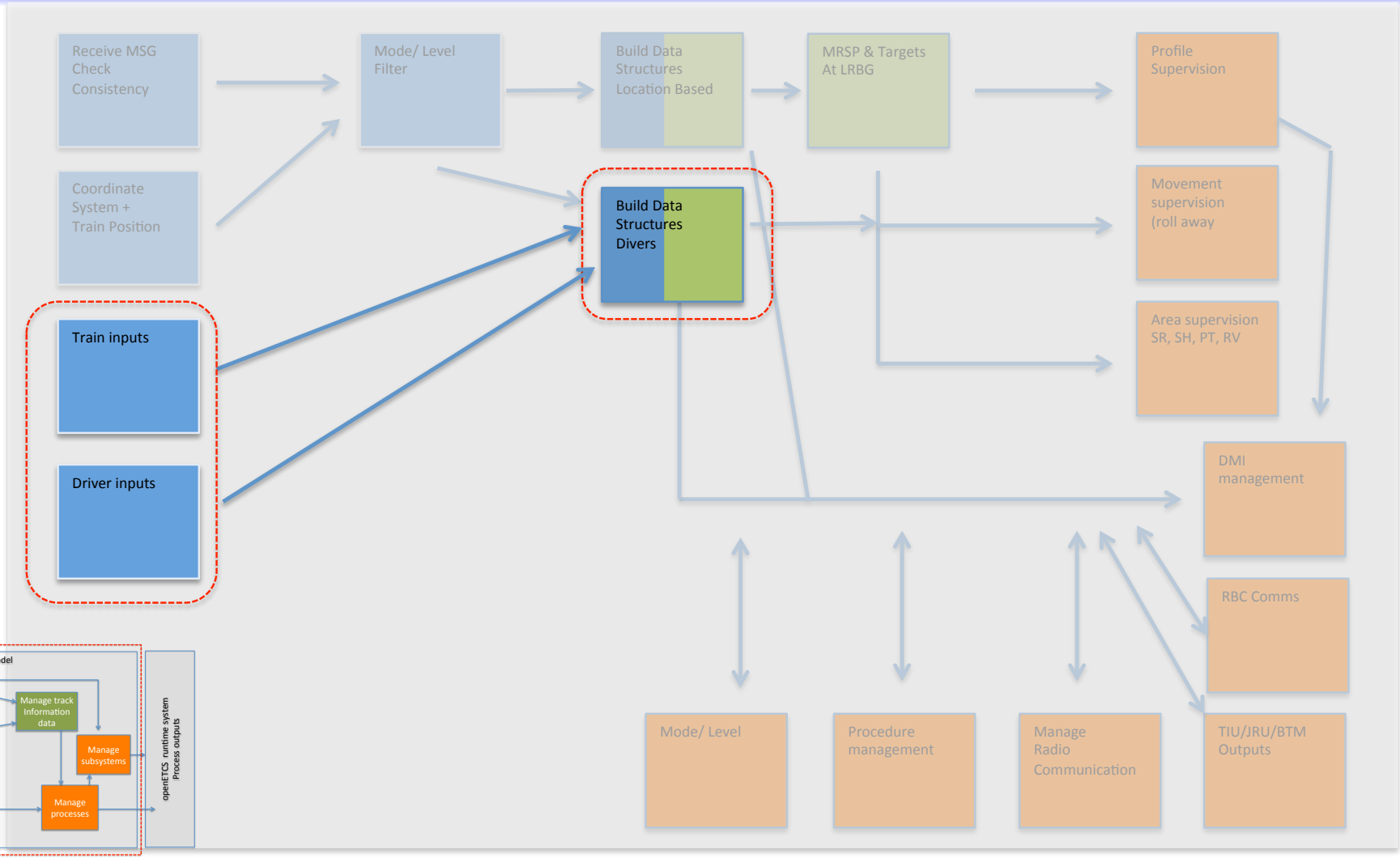
Standard SCADE execution model



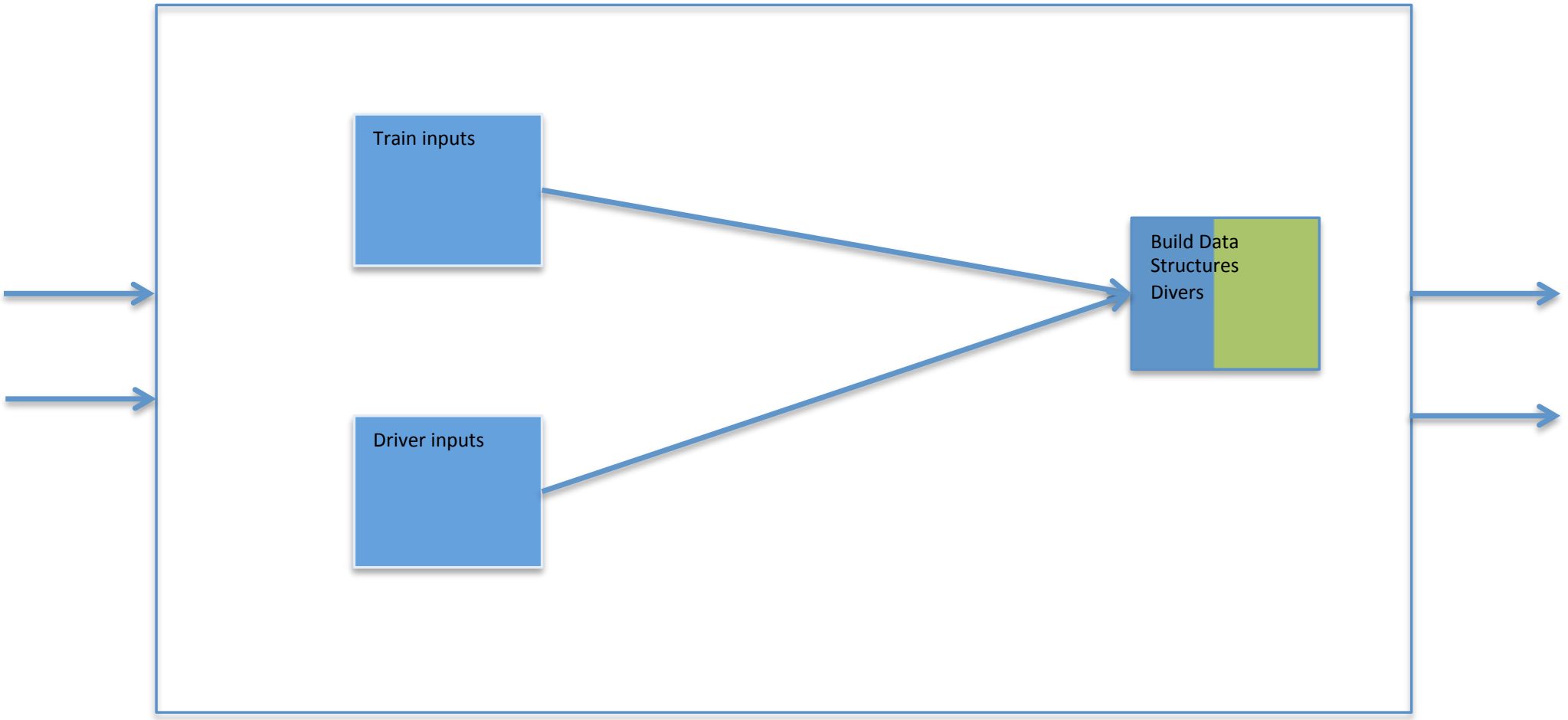
Top- level openETCS architecture (draft)



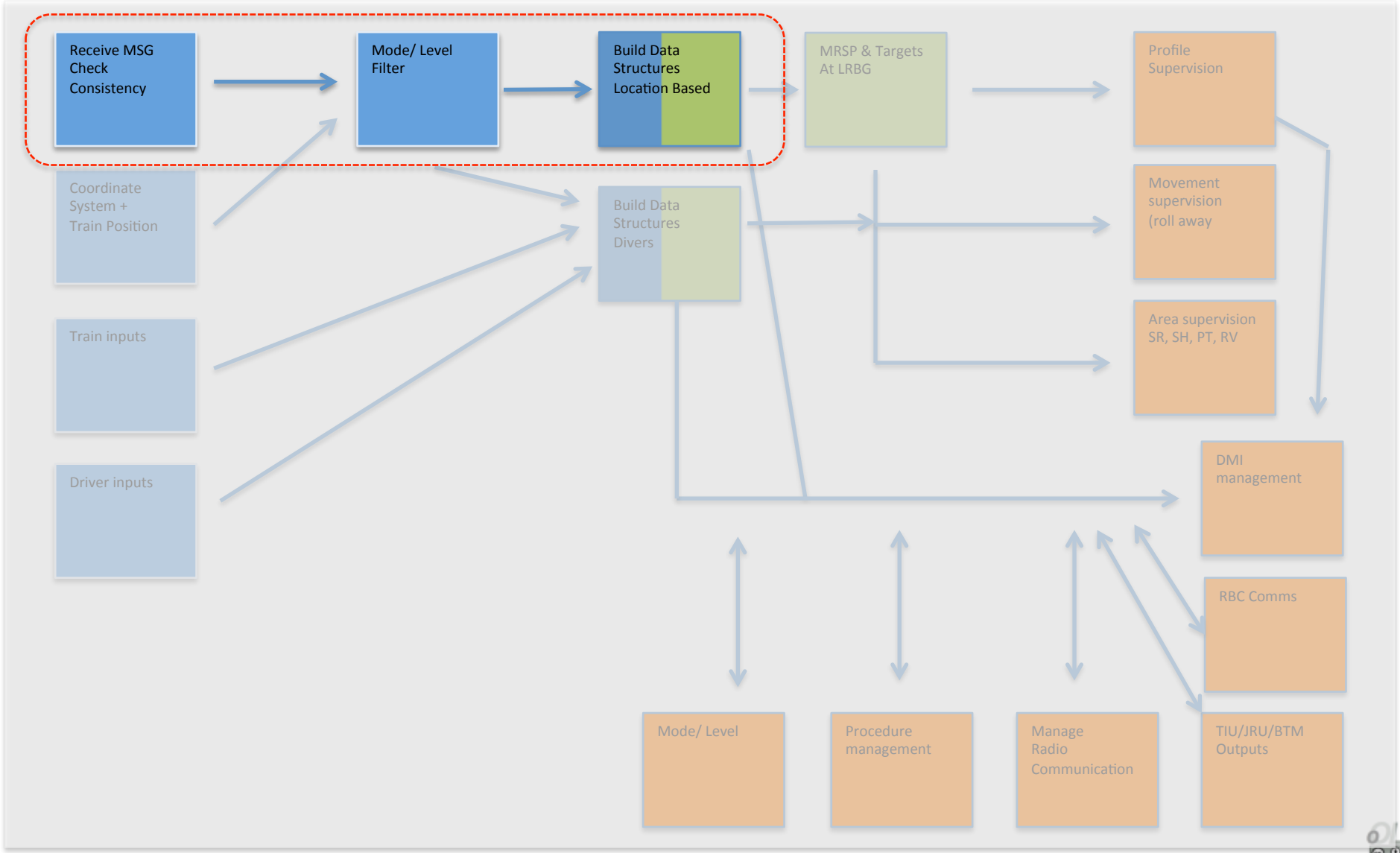
Store system data



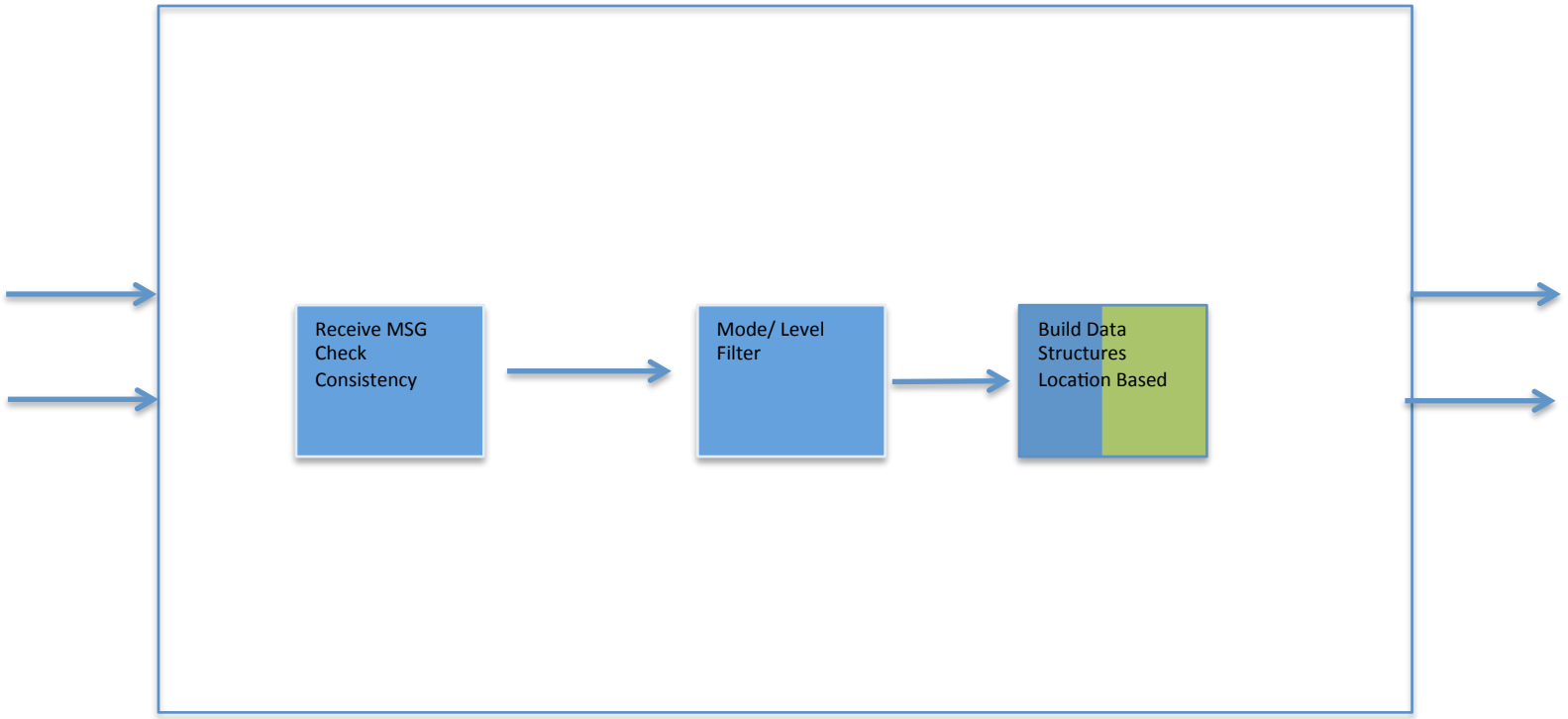
Store system data



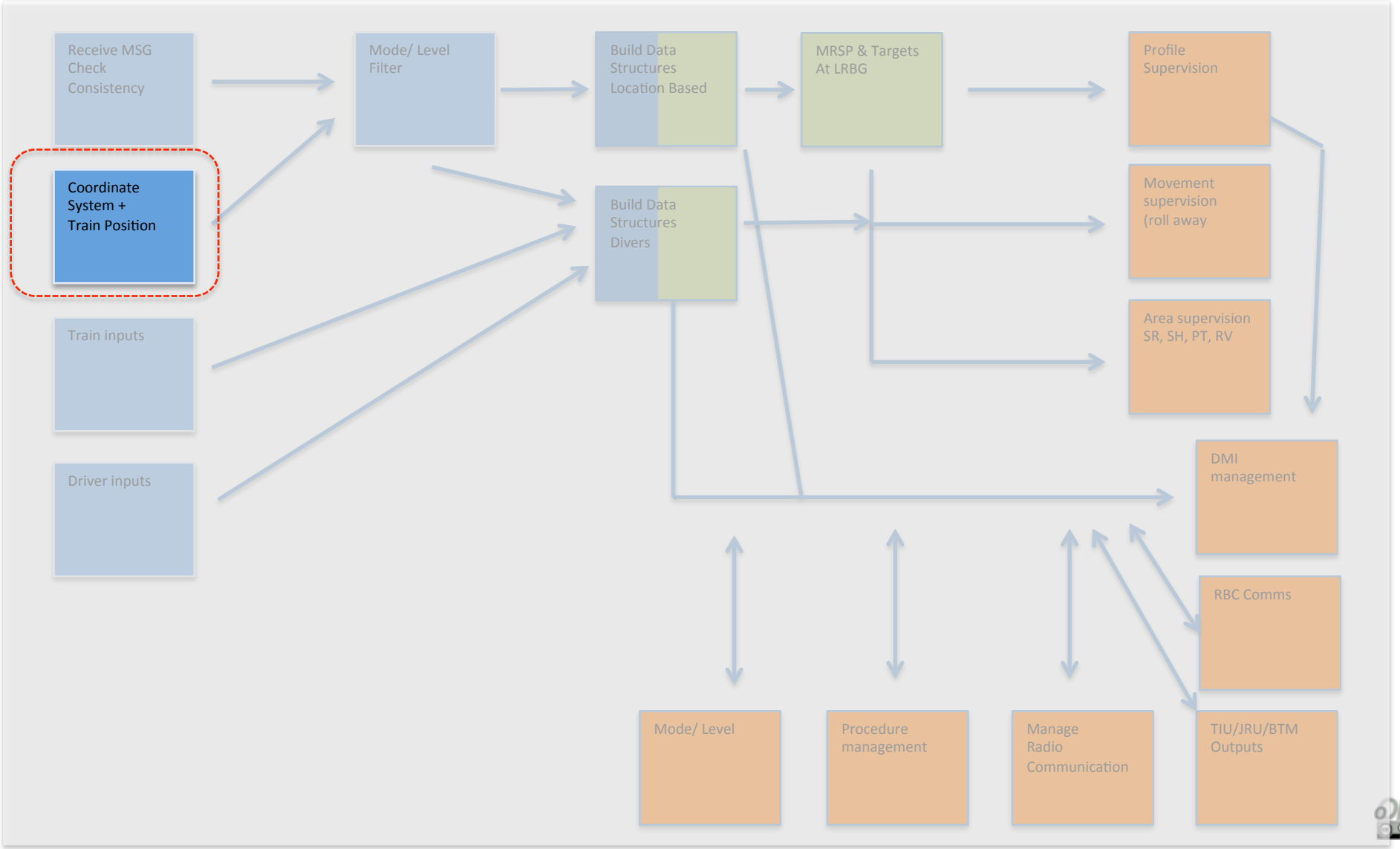
Check and process track information



Check and process track information

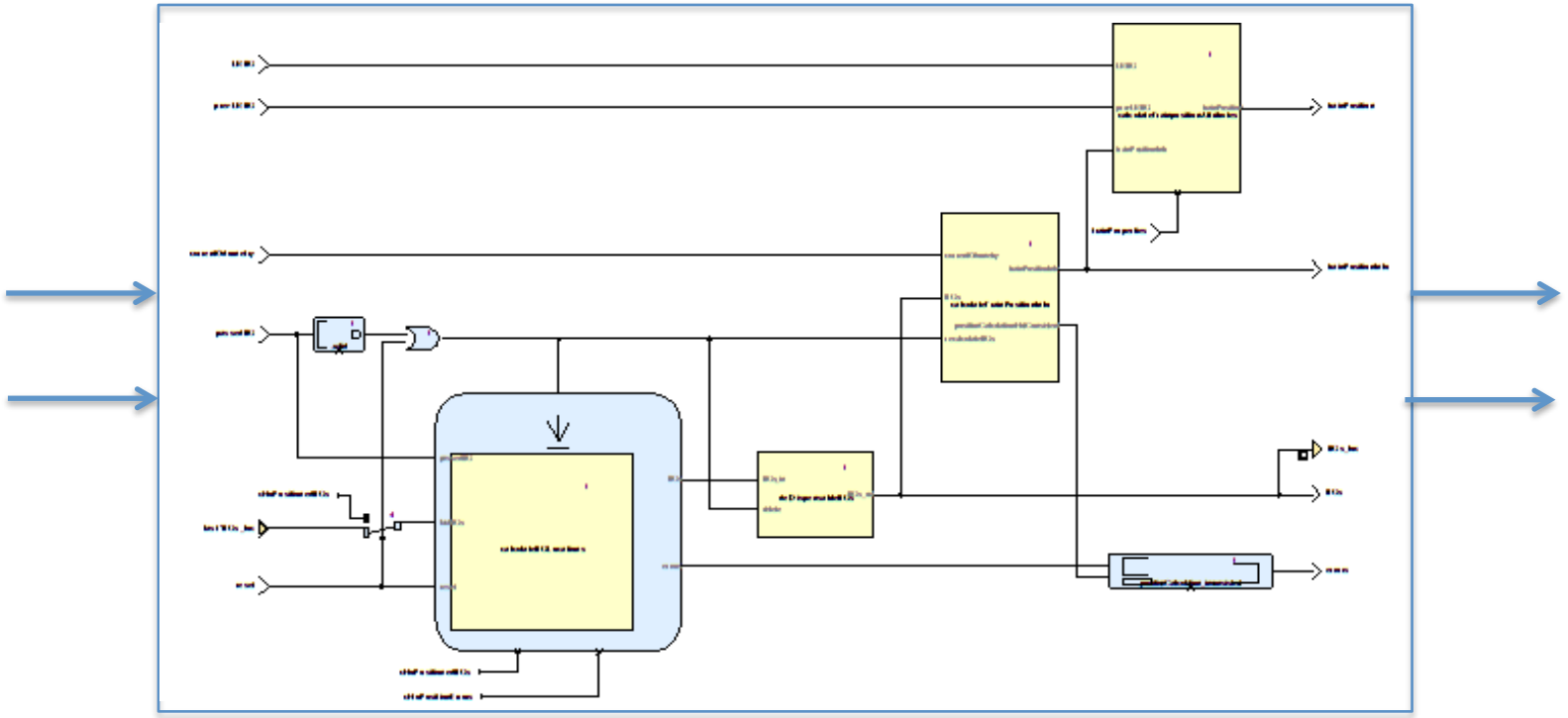


Calculate train position

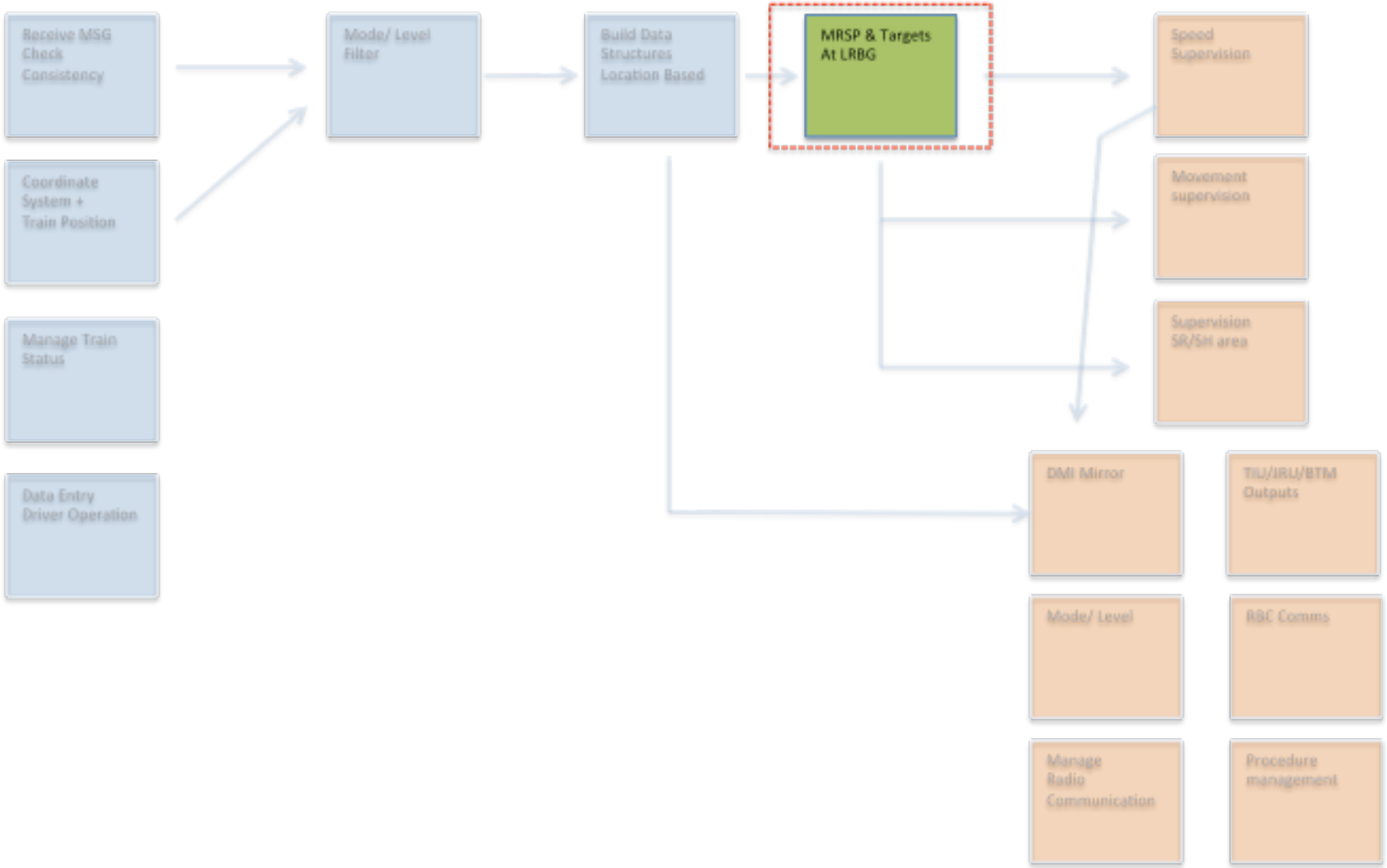


Calculate train position (Block view)

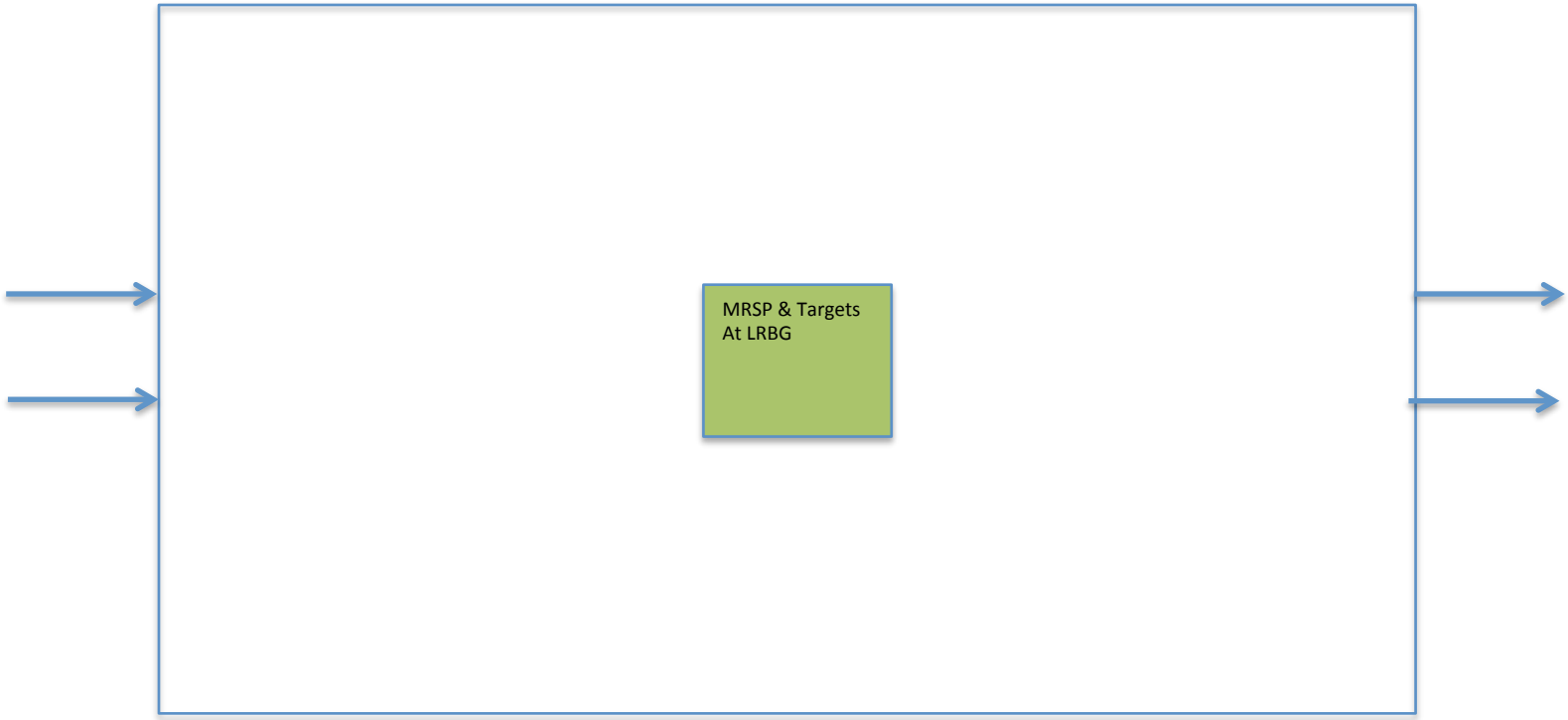
Calculate train position (SCADE view)



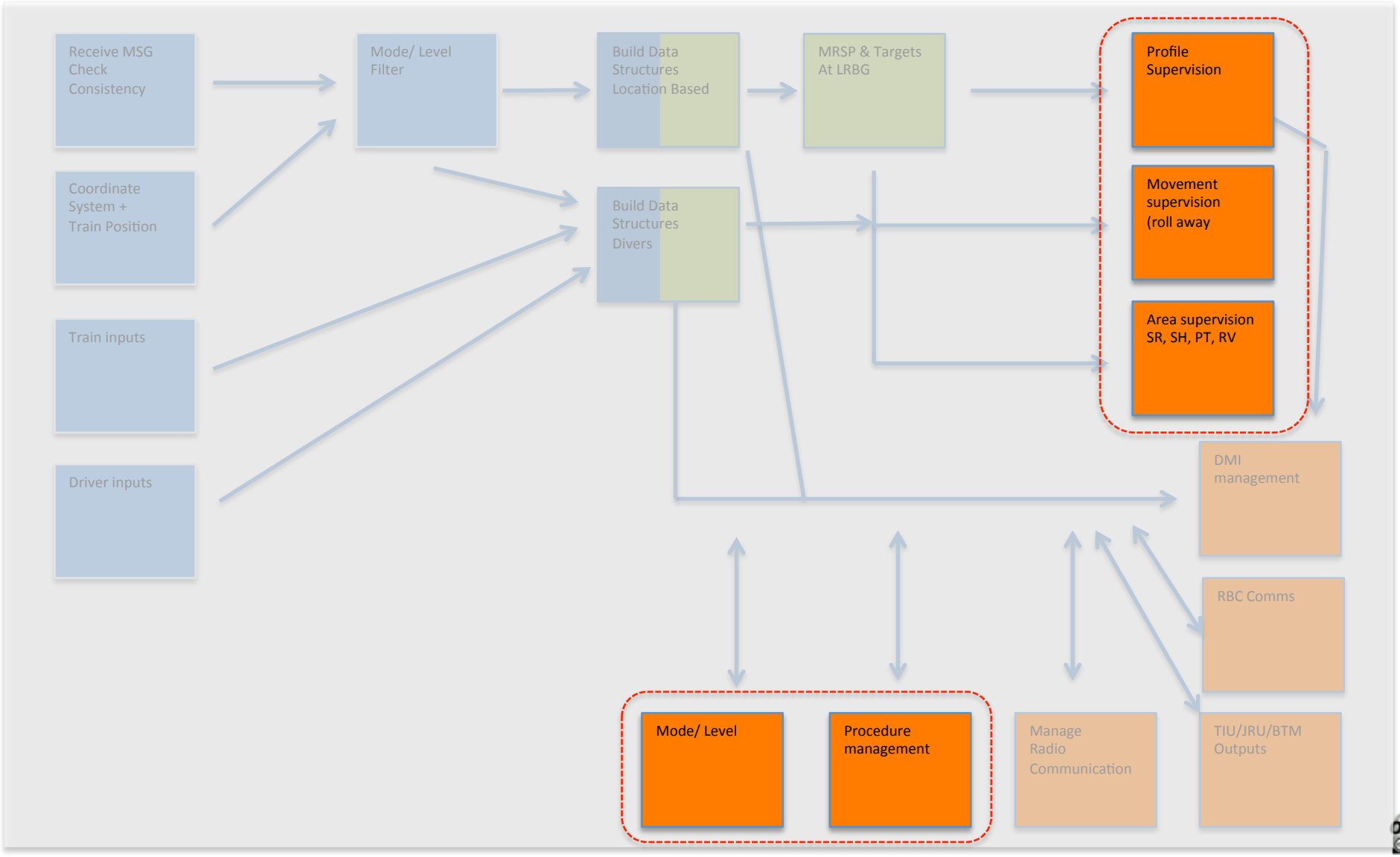
Manage track information data



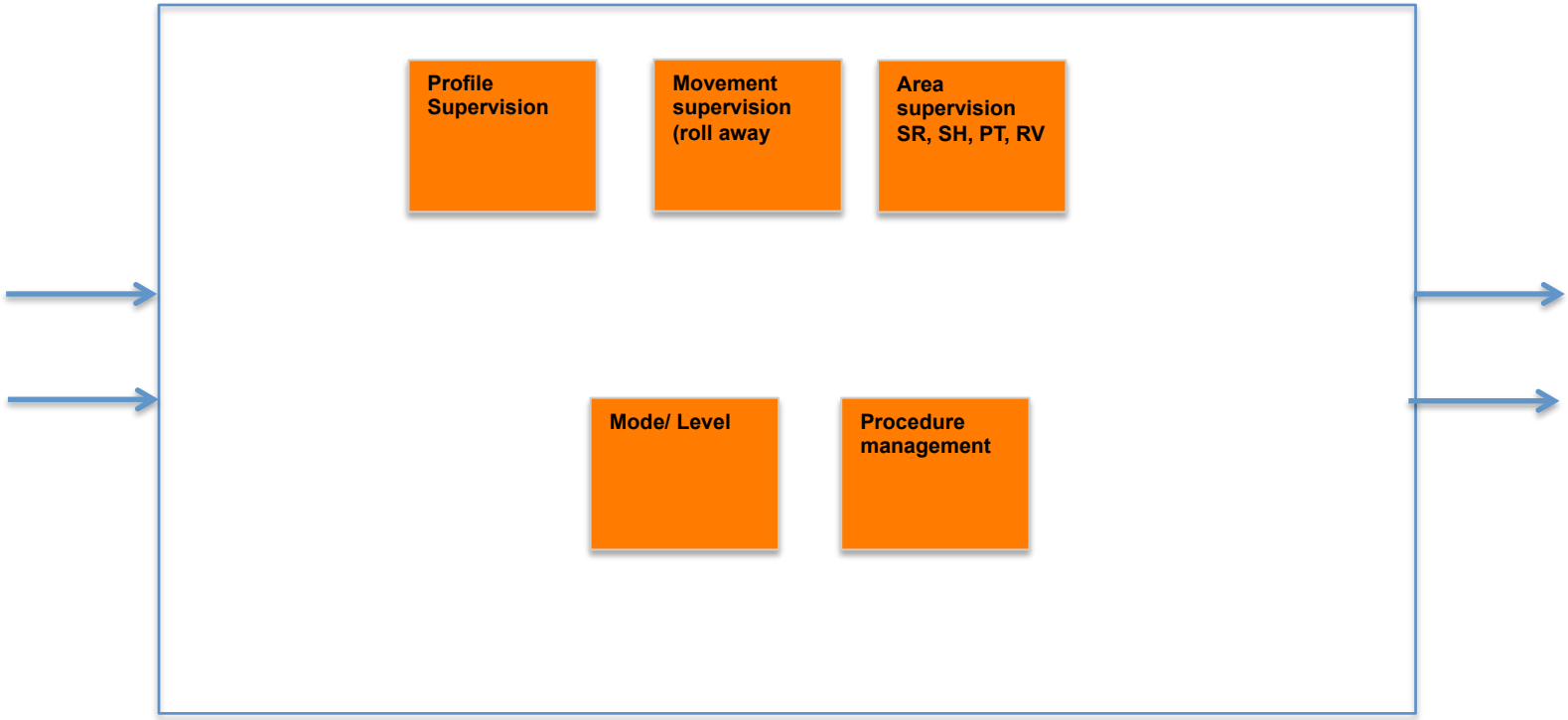
Manage track information data



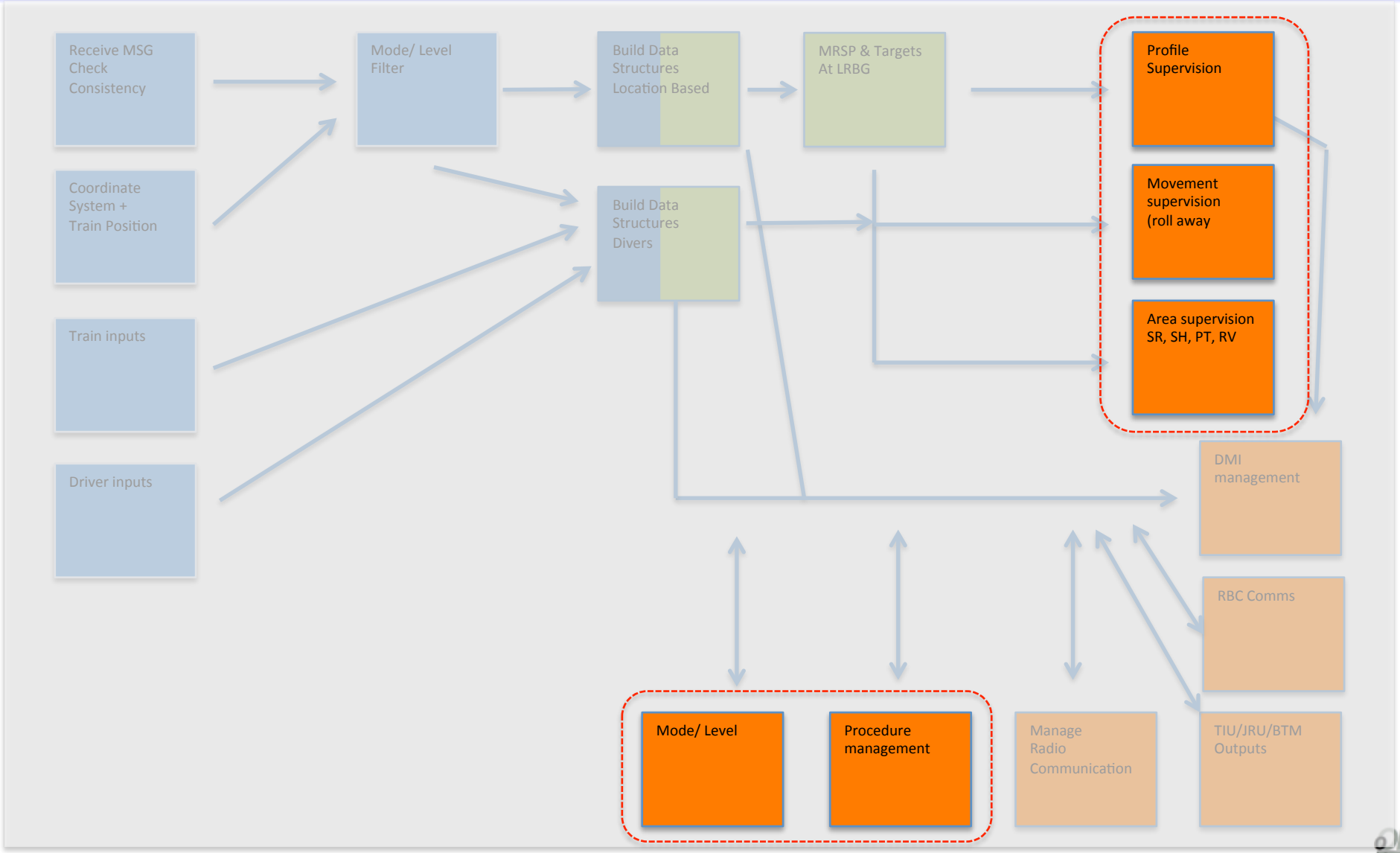
Manage processes



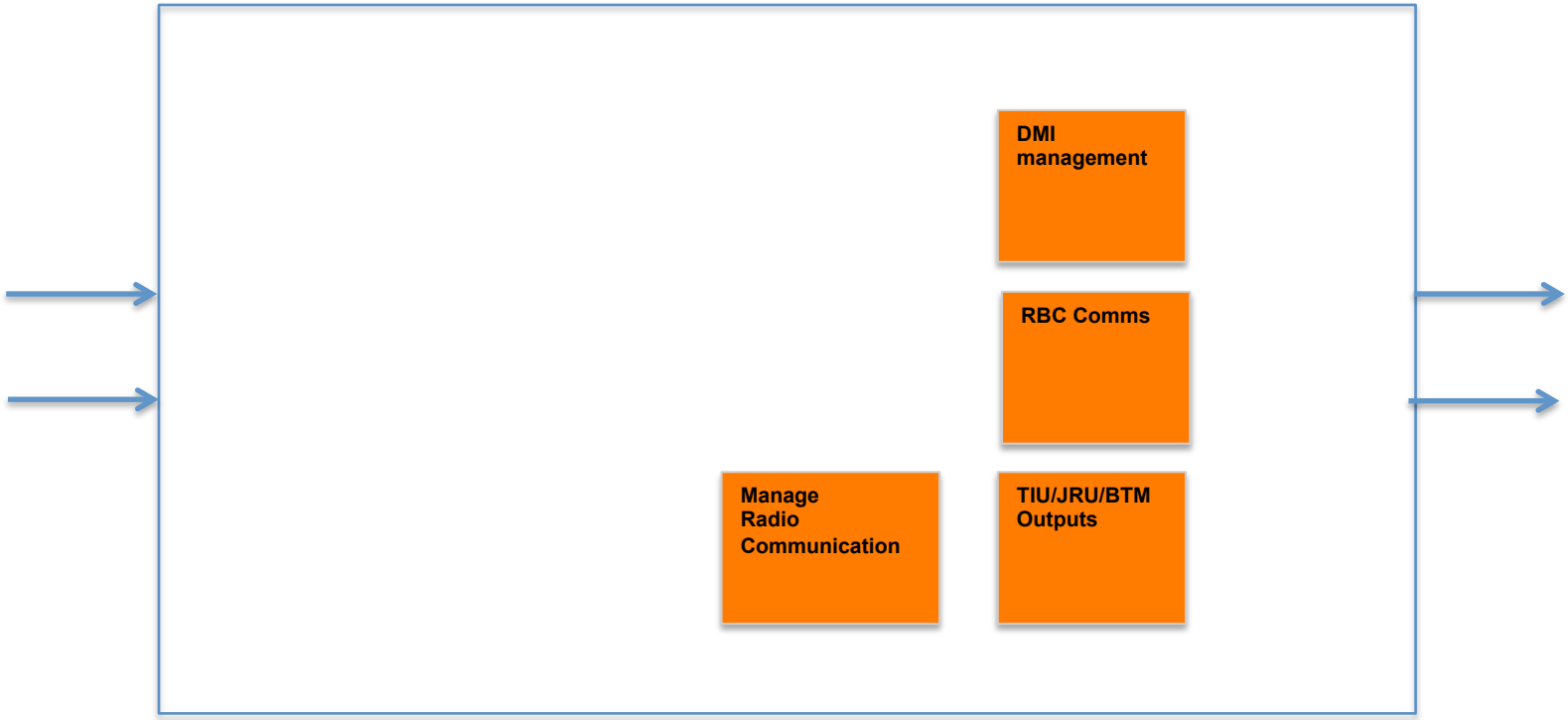
Manage processes



Manage subsystems



Manage subsystems



Data flows and inputs/outputs

From the perspective of an operator, we talk about its *inputs and outputs*

From the higher perspective of a model, we talk about data flows

For our architecture, we can work *top down* identifying the main data dependencies between our functional blocks

We can also work *bottom up* identifying the in- and outputs of each operator during SRS analysis, and later use this information to complete the data flows of the next higher level (and so forth)

After the main dependencies are identified, it is usually more efficient to analyze each operator separately and integrate the flows on the next higher level

The work on this part of analysis can equally be done using SysML or SCADE, but we need to ensure a safe updating/ syncing process.

We need to make sure to mirror all modeling work in the ADD document.

Scope of SysML vs. SCADE

SysML: Interface description, relations with other systems, high- level architecture

SCADE: Functional model

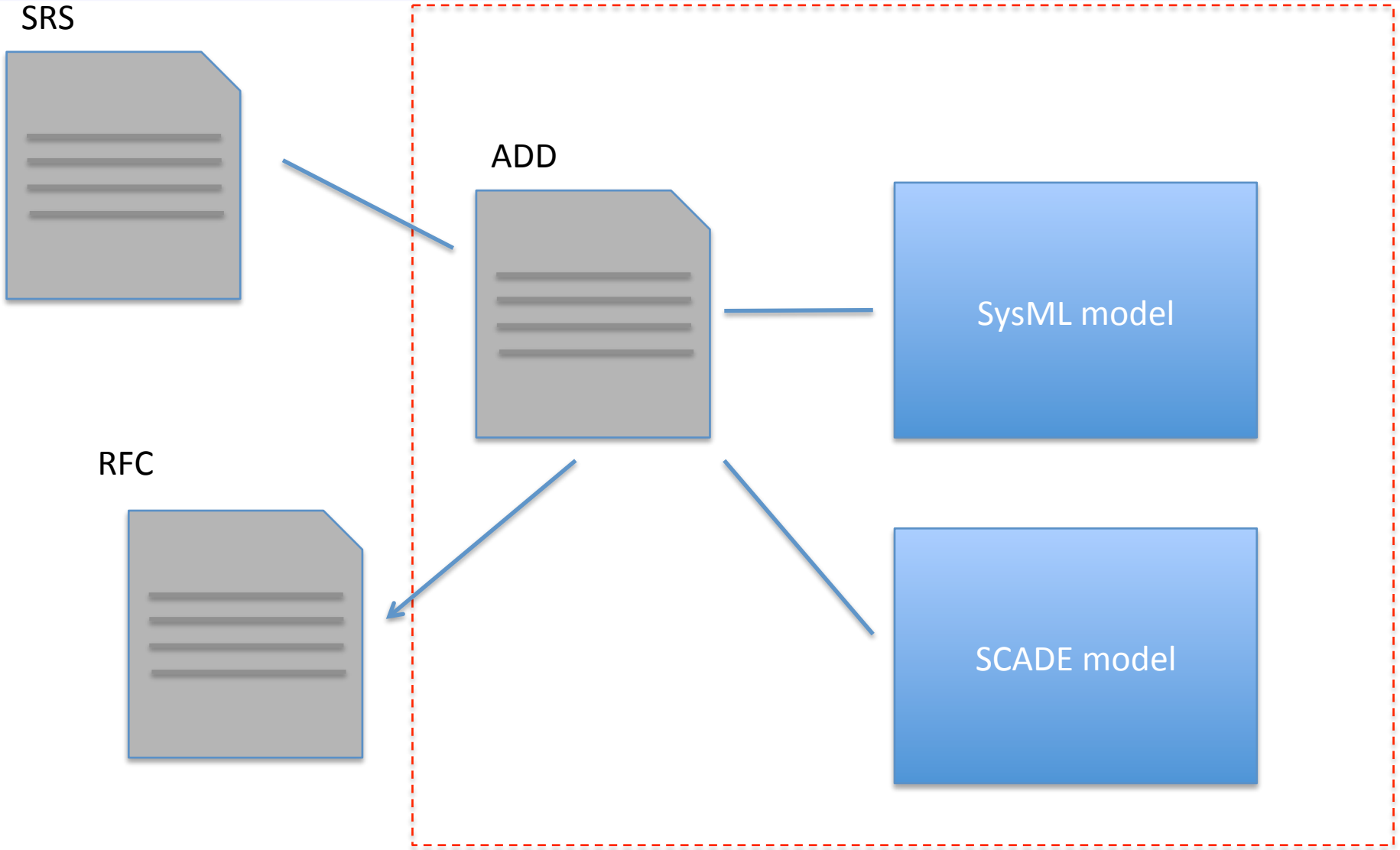
Typically, the SCADE model will encompass the entire openETCS executable model. There should be no need for any handwritten code, unless for some operations on bit level (which are typically part of the runtime system anyway)

The SysML model will extend from the system view down to the main functions (for example “Mode/ Level”)

The SCADE model will include the “openETCS executable model” as it’s top level operator

→ **The two highest SCADE levels will overlap with the SysML model**

Elements of architecture description



Typical scope of SCADE

In our experience, 90-100% of the openETCS executable model can be implemented in SCADE

- Typical exceptions are bitwise operations, which in our architecture are part of the basic software
- Hand-written code could be integrated in a SCADE model for such cases

Data flow (with proof for completeness and consistency)

Control flow (with proof for determinism and completeness)

Algorithms (SCADE has been invented for numerical algorithms at first)

Timing and scheduling: Determinism with multiple clocks inside the same model

Alignment strategy and process proposal

- Objectives for this workshop
  - Align high- level architecture
    - *(Levels 1, 2, and 3)*
    - *Define interfaces*
  - Align processes and tools

- Establish procedures for further design work
  - Interfaces
  - Responsibilities
  - Decision process(es)

- Identify issues
  - Resources
  - Skills
  - Risks

Step 1: Establish work flow for this workshop

- Agenda ok?
- Document structure agreed?
- Presentation/ discussion rules agreed?
- Decision process for these 5 days
  - Presentation
  - Discussion
  - Consensus
    - *Direct consensus*
    - *Mediated consensus*
    - *Interpretation/ decision by content leader*

Step 2: Establish work flow for following phase

- Based on the workshop results, we must
- Finalize the architecture
- Transition to the design phase

- Specific sessions are foreseen in the agenda in order to prepare this work