API for a ERTMS/ETCS onboard model

Niklas Schaffrath
Principal Key Expert
Software Architecture

Siemens Rail Automation

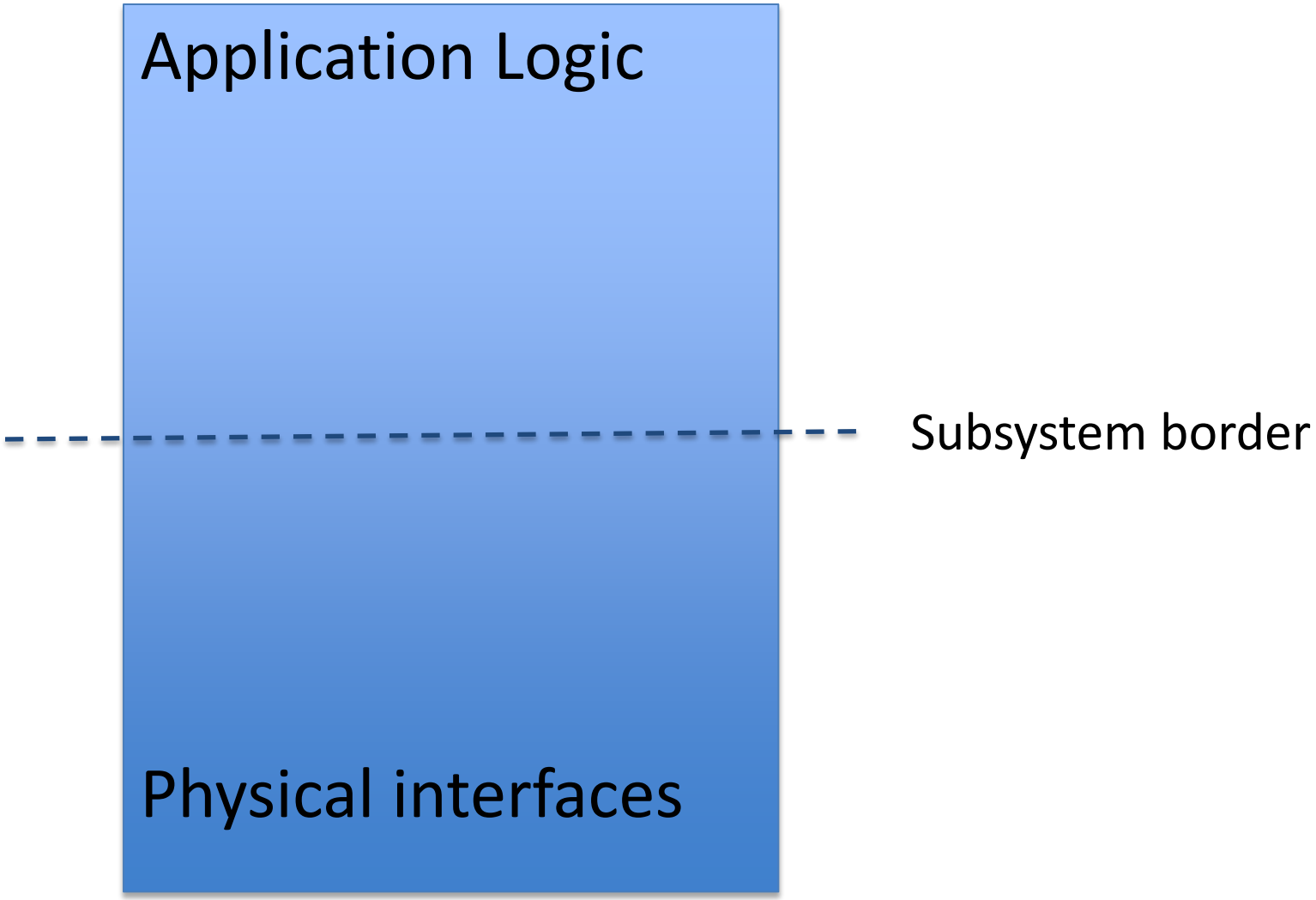
Operating system

- provides runtime environment
- provides communication means
- provides infrastructure functions

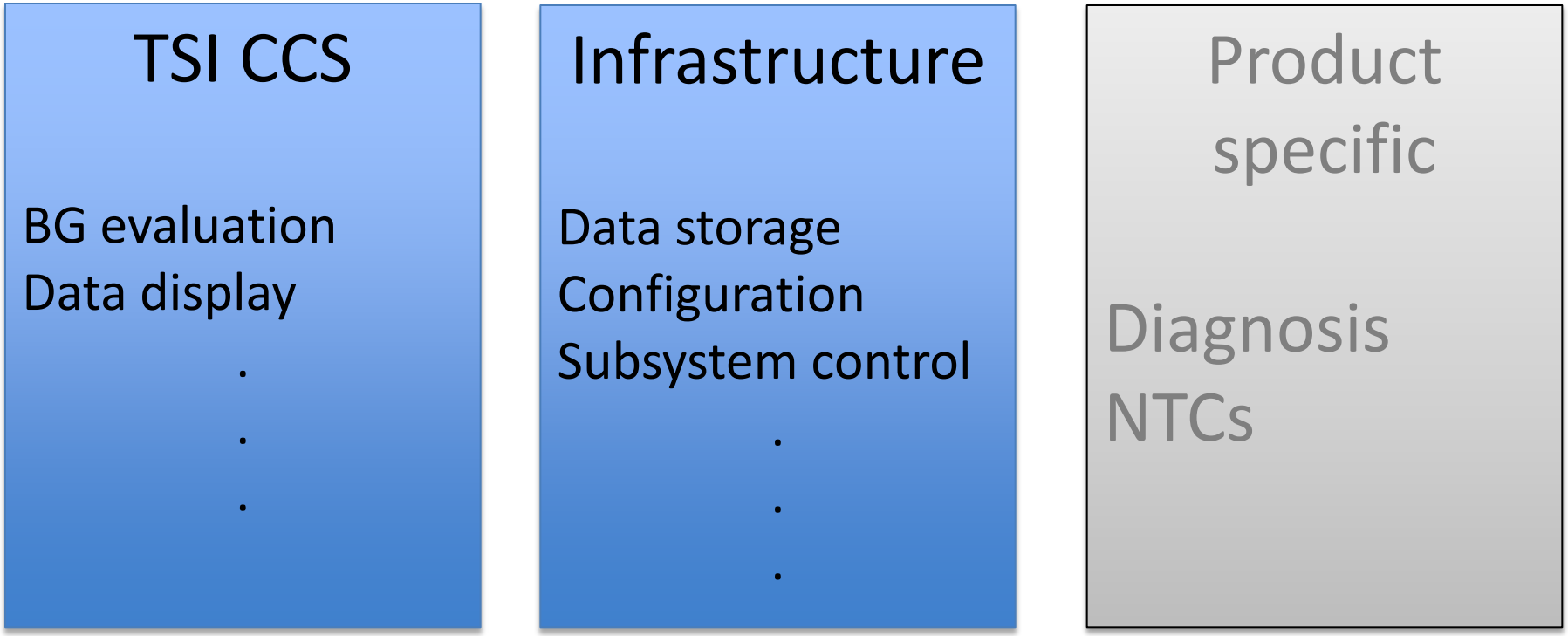
Functional Subsystems

- Only the ETCS onboard at its external interfaces is defined
- Idea of a white box architecture with subsystem
- Interfaces between kernel and subsystem are not defined
- split of responsibility between kernel and subsystem ?

Structure of the ERTMS/ETCS functions



Functional Interfaces



IF based on functional split

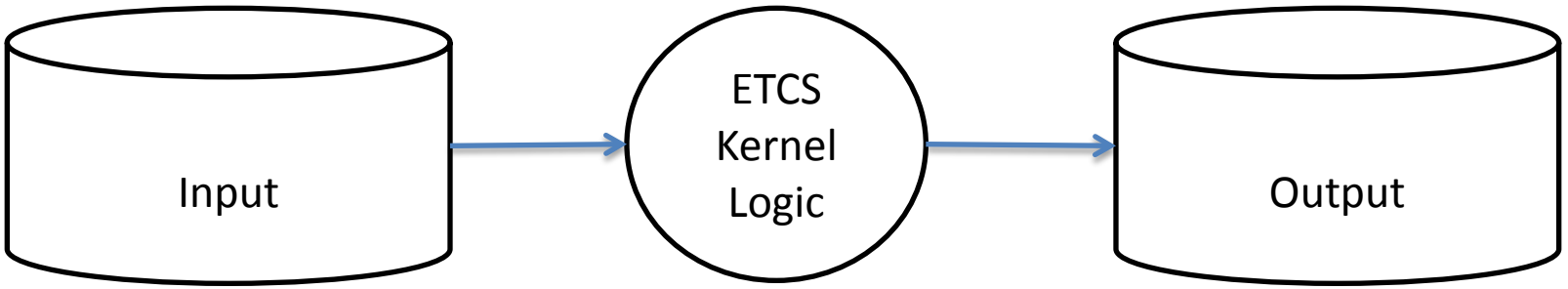
Generic IF (e.g. system image)

Not considered

Application Logic

Best practice:

- cyclic execution model
- Communication with timed data streams



Cycle time

$t_{min} \leq$ reaction_time / 2

ETCS: reaction_time ≈ 1s at external interfaces

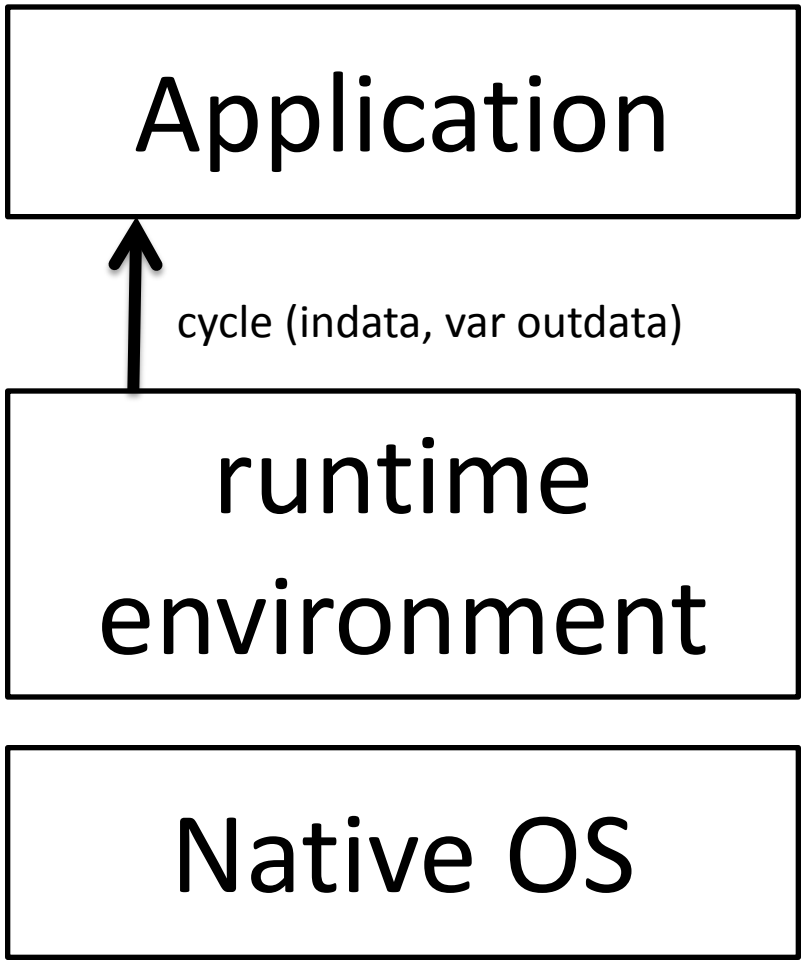
With 200ms for input/output operations each

-> $t_{min} \leq$ 300ms -> 200ms (1s = 5*t_{min})

Maximum Balise Frequency 8/s

-> with t_{min} = 100ms only one Balise/cycle has to be evaluated

The runtime system



Connects to subsystem
- from/to data stream
- **platform depended**

Cross cutting concerns

- Error reaction

  The state “system failure” is usually heterogeneous represented in the application and the platform

- Debug diagnosis

  Data output for testing and debugging

  (≠ product feature self diagnosis)

Summary

Runtime framework

- cycle (indata, var outdata)
- (init)
- abort
- Debug

Input / Output data structures based on subsystem definition & functional analysis