SOFTWARE ARCHITECTURE GATHERING DIGITAL

# Embedded System Architecture


Wolfgang Reimesch

Reimesch IT-Training und Beratung GmbH
www.reimesch.eu

Peter Hruschka

Atlantic Systems Guild
www.arc42.de
www.req42.de

USING THE PROVEN

arc42 TEMPLATE …

… TO DESIGN & DOCUMENT AN EMBEDDED HARDWARE/SOFTWARE SYSTEM

# WHAT WE INTEND TO SHOW

# Wolfgang Reimesch

30+ years developing
embedded systems

21 years CEO of
Reimesch Kommunikationsysteme
(developing HW/SW communication solutions)

Now: Certified Trainer for ISAQB CPSA seminars
Consultant for HW/SW architectures

Contact: wr@reimesch.eu


# Peter Hruschka

Principal of the
Atlantic Systems Guild
(www.systemsguild.com)

Founding Member of ISAQB and IREB
Trainer, Consultant, Author

Co-Founder of arc42
Co-Founder of req42

Contact: peter@systemsguild.com


arc42:

- A simple "container" structure to capture the essence of architecture
  - based on (only 3) Architectural Views
- · Support for "structured laziness"
  - Stop meta-discussions on documentation structure
  - Start filling in the content
- Many different formats and languages
- Open Source since 2007
  - www.arc42.org

This template cannot only be used for software architectures, but can also **capture system architectures (with HW, SW, electronics, mechanics, ...)**

1 Intro & Goals
2 Constraints
3 Context
4 Solution Strategy
5 Building Block View
6 Runtime View
7 Deployment View
8 Concepts
9 Design Decisions
10 Quality Scenarios
11 Risks/Technical Debts
12 Glossary

1.1 REQUIREMENTS OVERVIEW

1.2 QUALITY GOALS

1.3 STAKEHOLDER

This document describes the Traffic Pursuit Unit, short TPU, which is a **speed measuring device** equipped with video recording facilities that is installed within a police car. It is used to **measure and record the speed profile of a car** driving in front of the police car, so that speed limit **violations can be proven and legal action can be taken** based on the documentation and video recordings produced by the system.

…

The MeasuringUnit of this system shall be able to run autonomously and be **marketed as a low cost variant** of TPU without video proof.

….

arc42

1 Intro & Goals
2 Constraints
3 Context
4 Solution Strategy
5 Building Block View
6 Runtime View
7 Deployment View
8 Concepts
9 Design Decisions
10 Quality Scenarios
11 Risks/Technical Debts
12 Glossary

1.1 REQUIREMENTS OVERVIEW

1.2 QUALITY GOALS

1.3 STAKEHOLDER

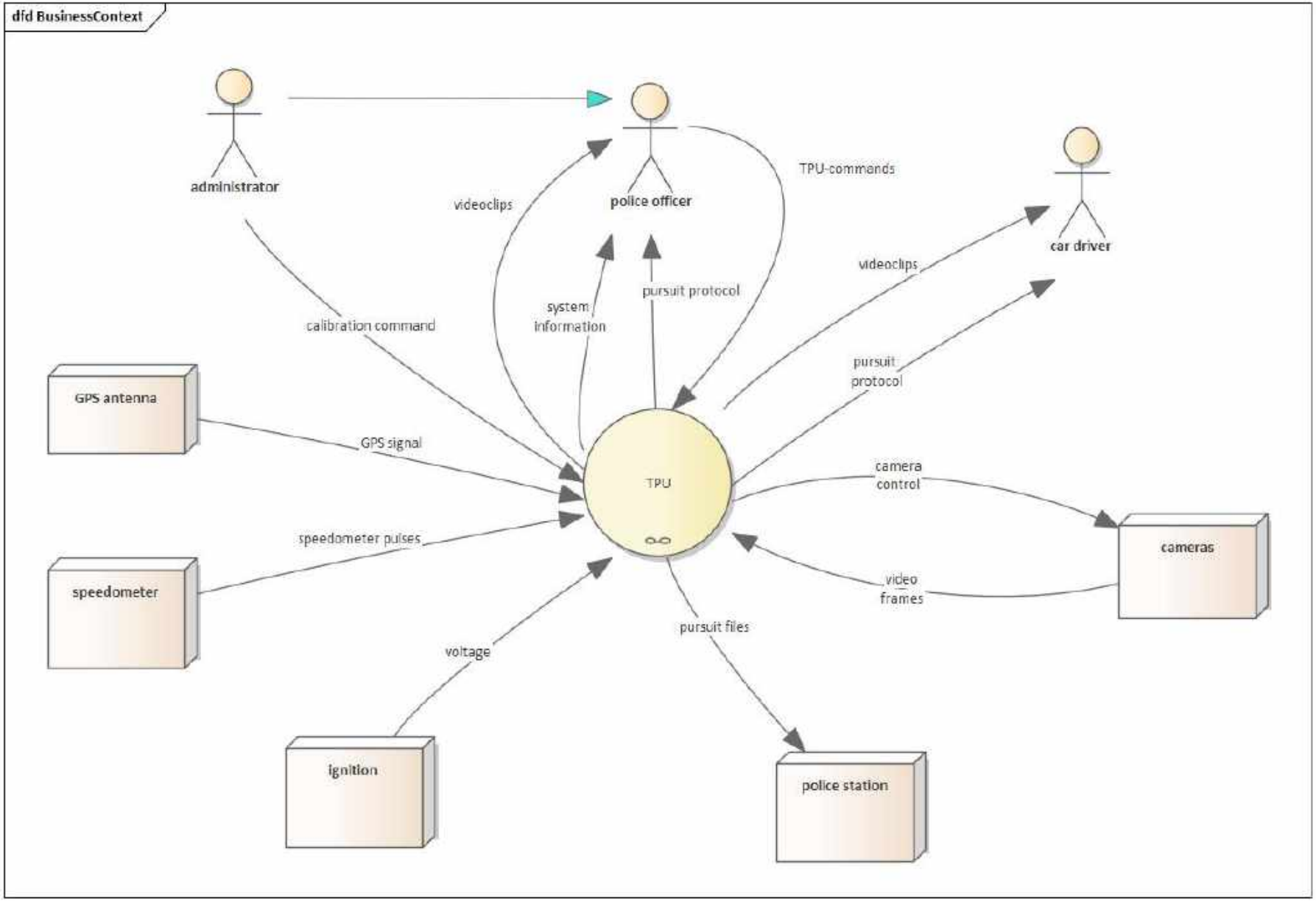
| | | |
|---|---|---|
| **1** | **Accuracy** | **All measurements and calculations shall be correct and precise within the specified deviation range.** |
| 2 | Robust-ness | The system shall work reliable under all specified environment and operating conditions. |
| 3 | Ease of use | Ease of use by the police officers, especially when pursuing another car |

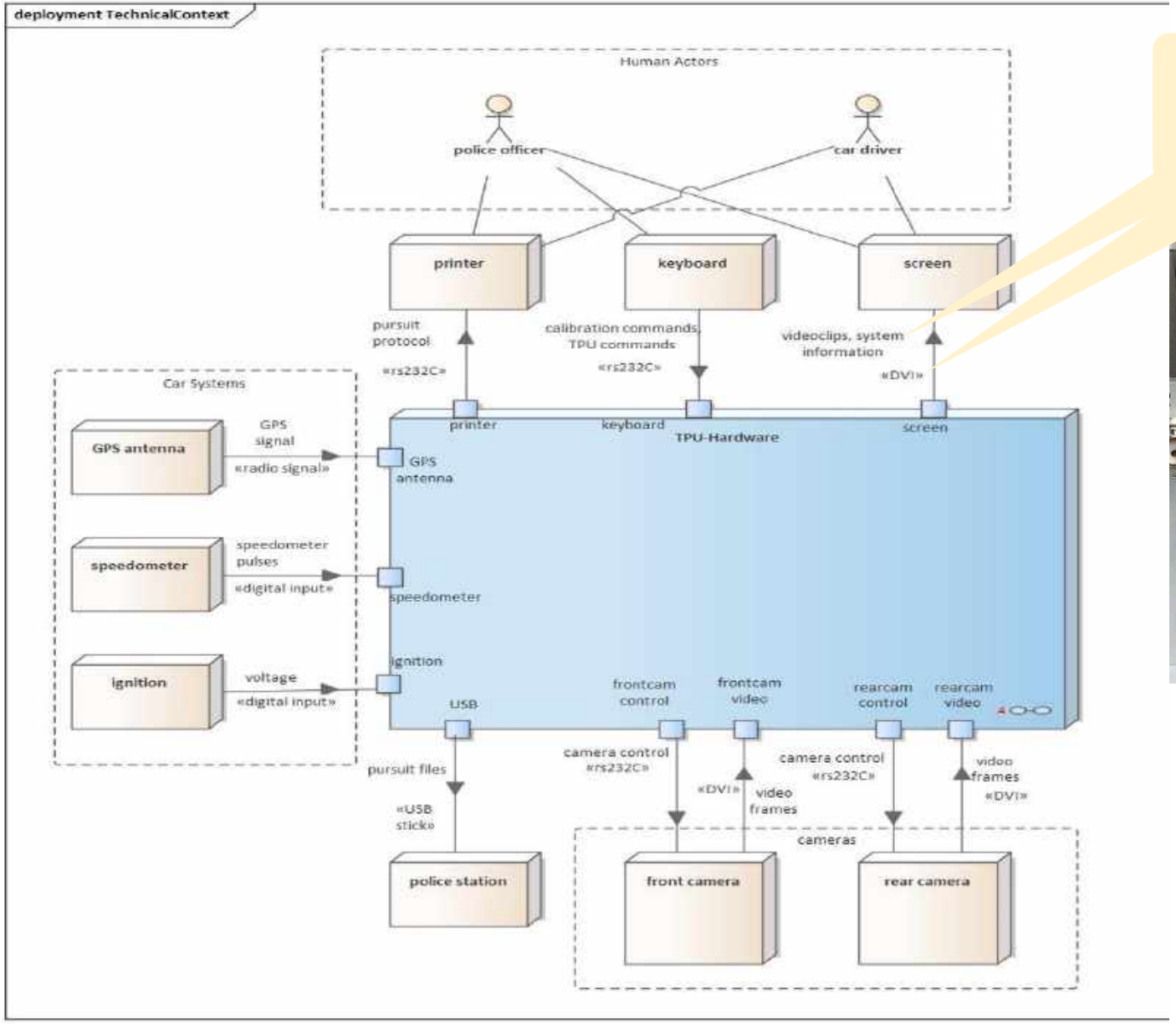
For more precision of system qualities we use quality trees and scenarios

1 Intro & Goals
2 Constraints
3.1 Business Context
3.2 Technical Context
4 Solution Strategy
5 Building Block View
6 Runtime View
7 Deployment View
8 Concepts
9 Design Decisions
10 Quality Scenarios
11 Risks/Technical Debts
12 Glossary

Shows the system inside its context,
i.e. all neighbors and all external interfaces
(the **inputs and outputs**)

Shows the system with all
**physical interfaces** to the neighbors
(and a mapping to the inputs
and outputs)


dfd BusinessContext
administrator
police officer
car driver
videoclips
TPU-commands
videoclips
system information
pursuit protocol
calibration command
pursuit protocol
GPS antenna
GPS signal
TPU
camera control
cameras
speedometer pulses
speedometer
video frames
pursuit files
voltage
ignition
police station




The output mapped to the channel

arc42

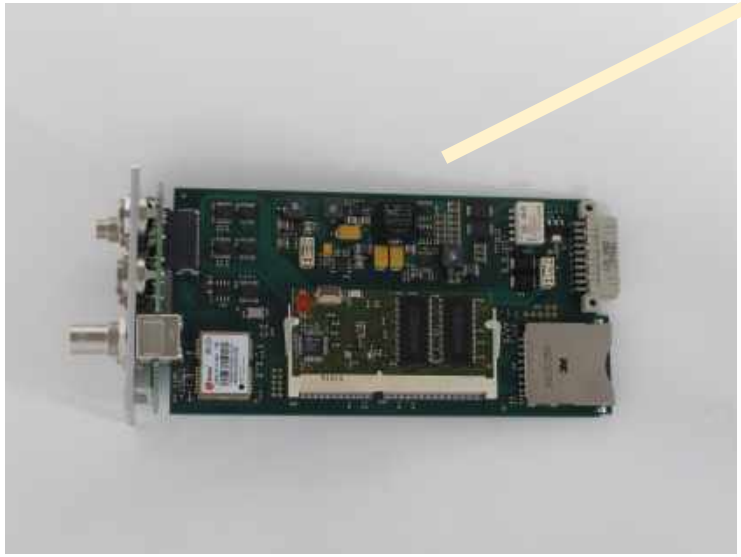
1 Intro & Goals
2 Constraints
3 Context
4 Solution Strategy
5 Building Block View
6 Runtime View
**7 Deployment View**
8 Concepts
9 Design Decisions
10 Quality Scenarios
11 Risks/Technical Debts
12 Glossary

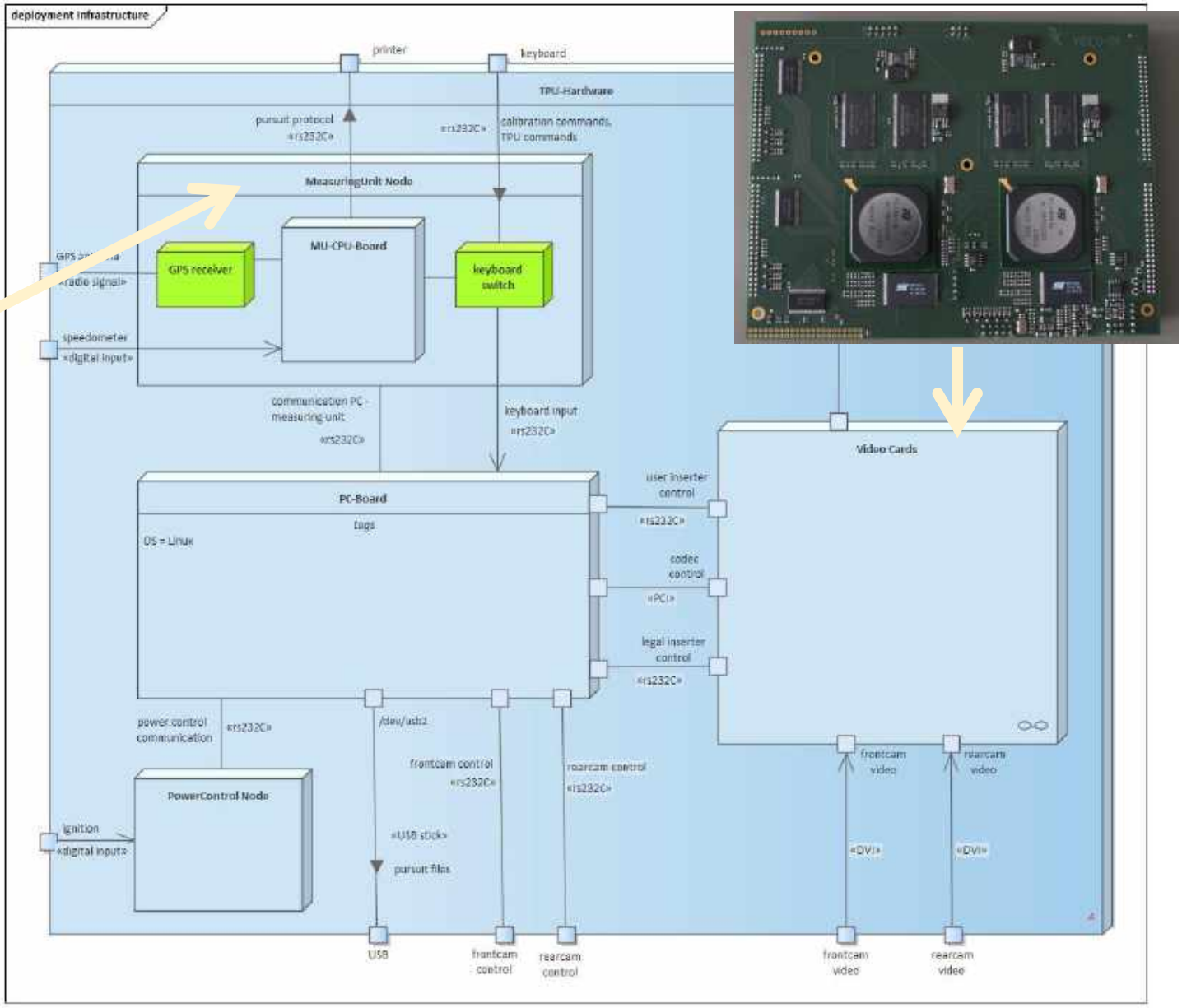
In embedded systems the top level system decomposition is very often hardware-driven.

You often already know which functionality requires specific processors or boards or chips

# The inside of this hardware …

# ... shown as Deployment Diagram





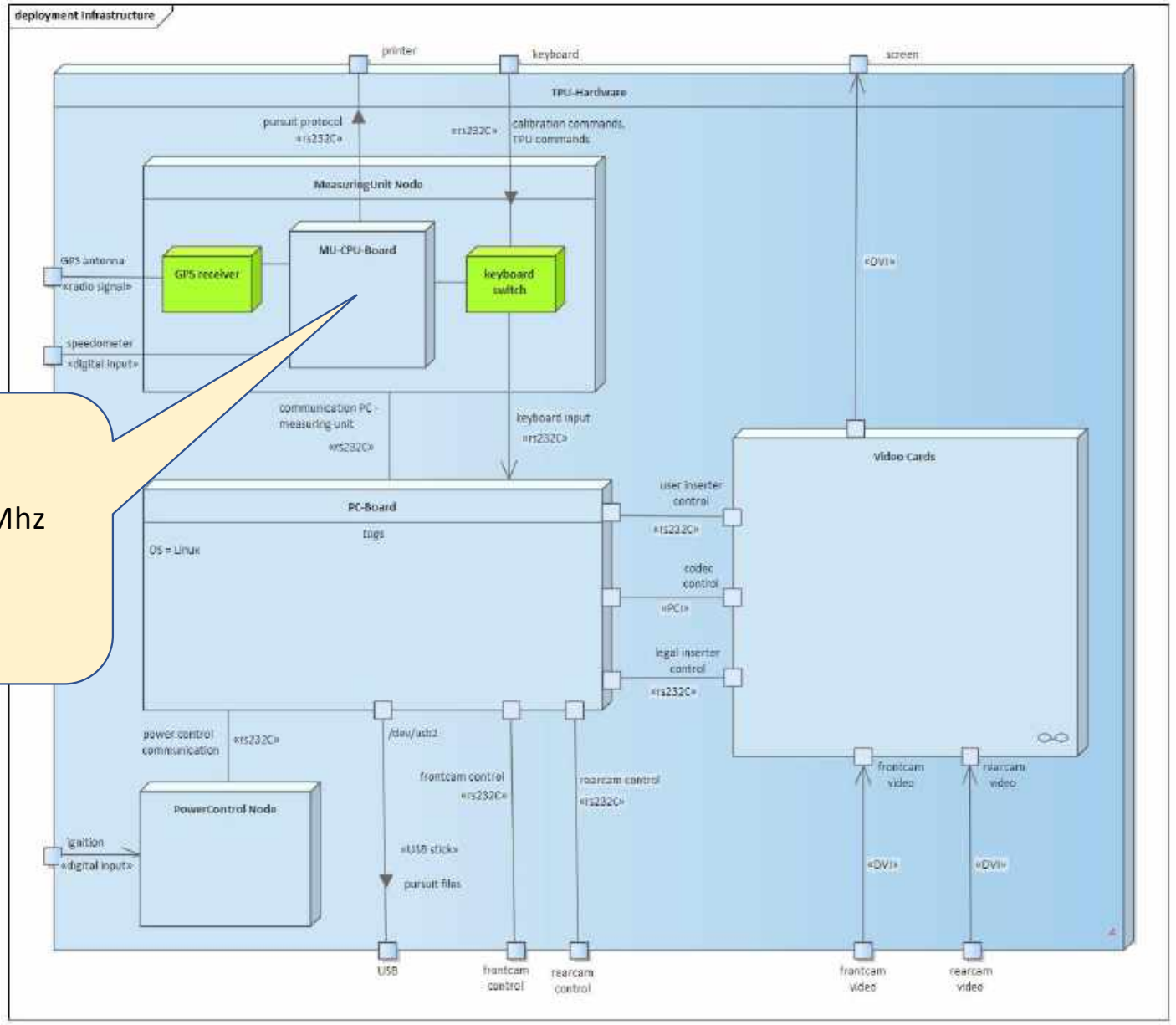

**Its purpose (1):**
Document the infrastructure
& the connecting channels

# ... shown as Deployment Diagram

**MU-CPU-Board**

CPU: ARM processor 200 MIPS at 180 Mhz

multiple RS232 lines with speeds up to 115Kbaud




**Its purpose (2):**
**Define capabilities of nodes and channels**

arc42

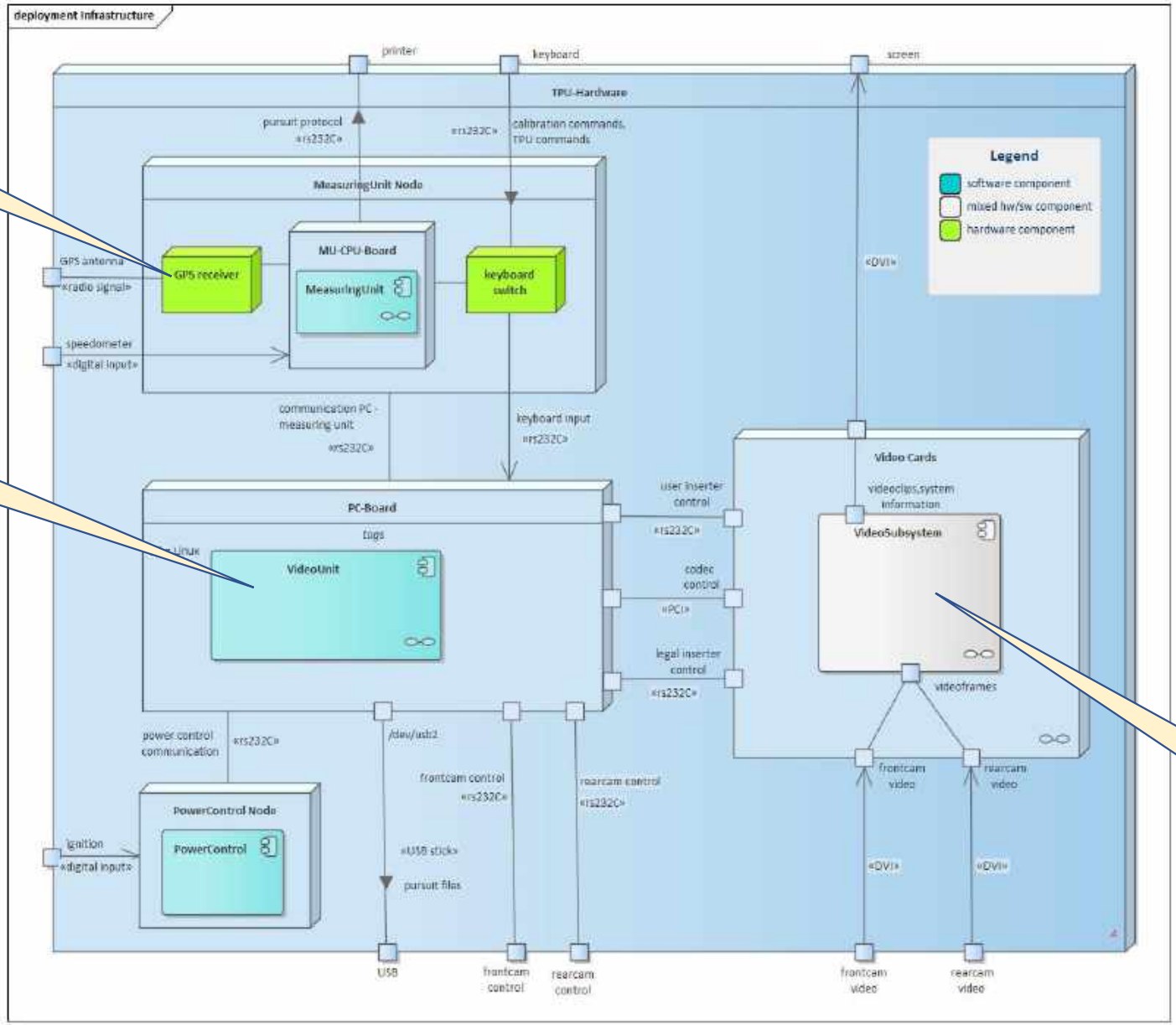


# The infrastructure determines the top-level decomposition of the functionality *)

*) Note that we talk about "functional" **building blocks**: It is not yet decided which functionality is implemented in hardware and which part is software.

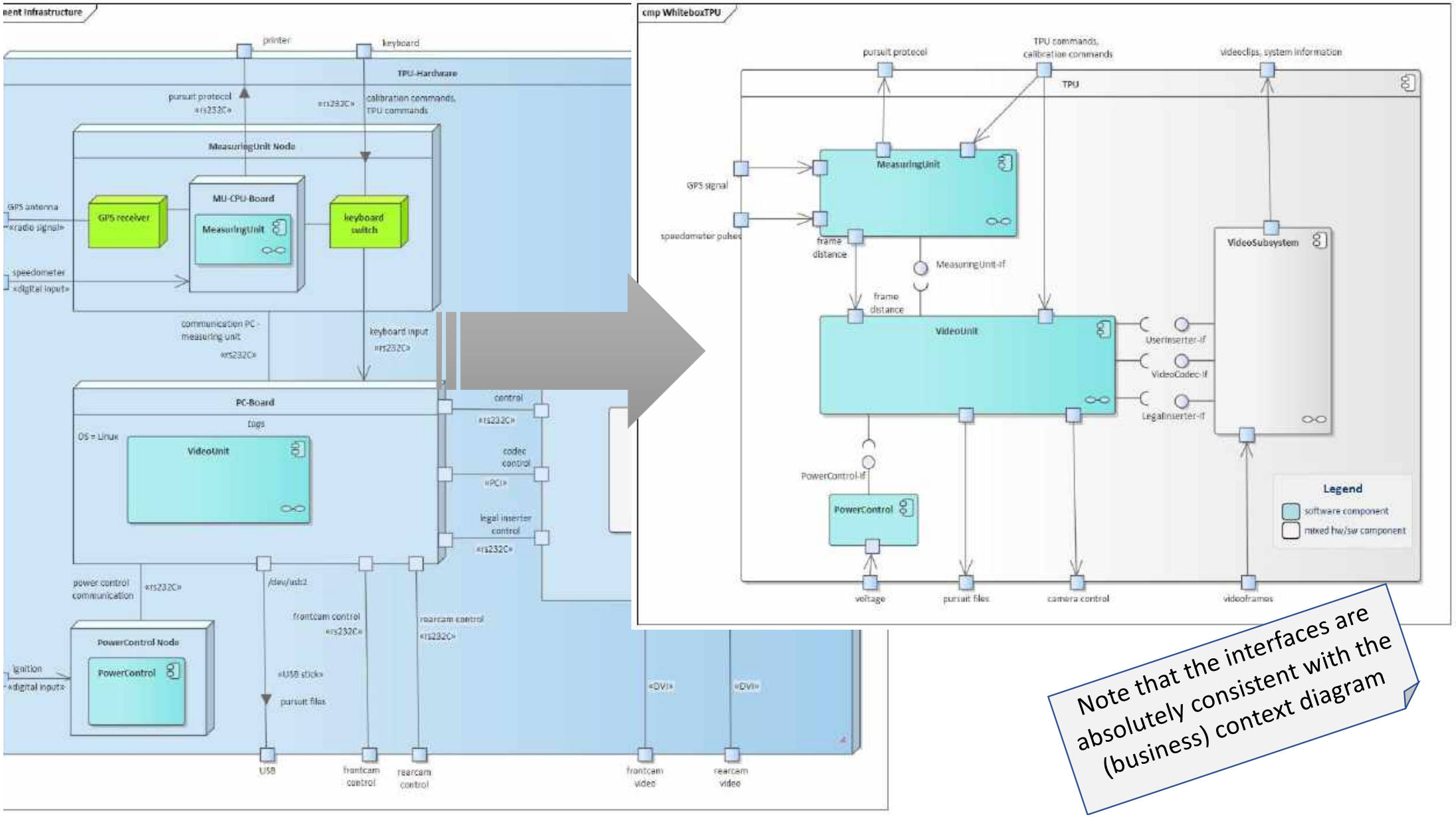
Pure HW function

Pure SW functions



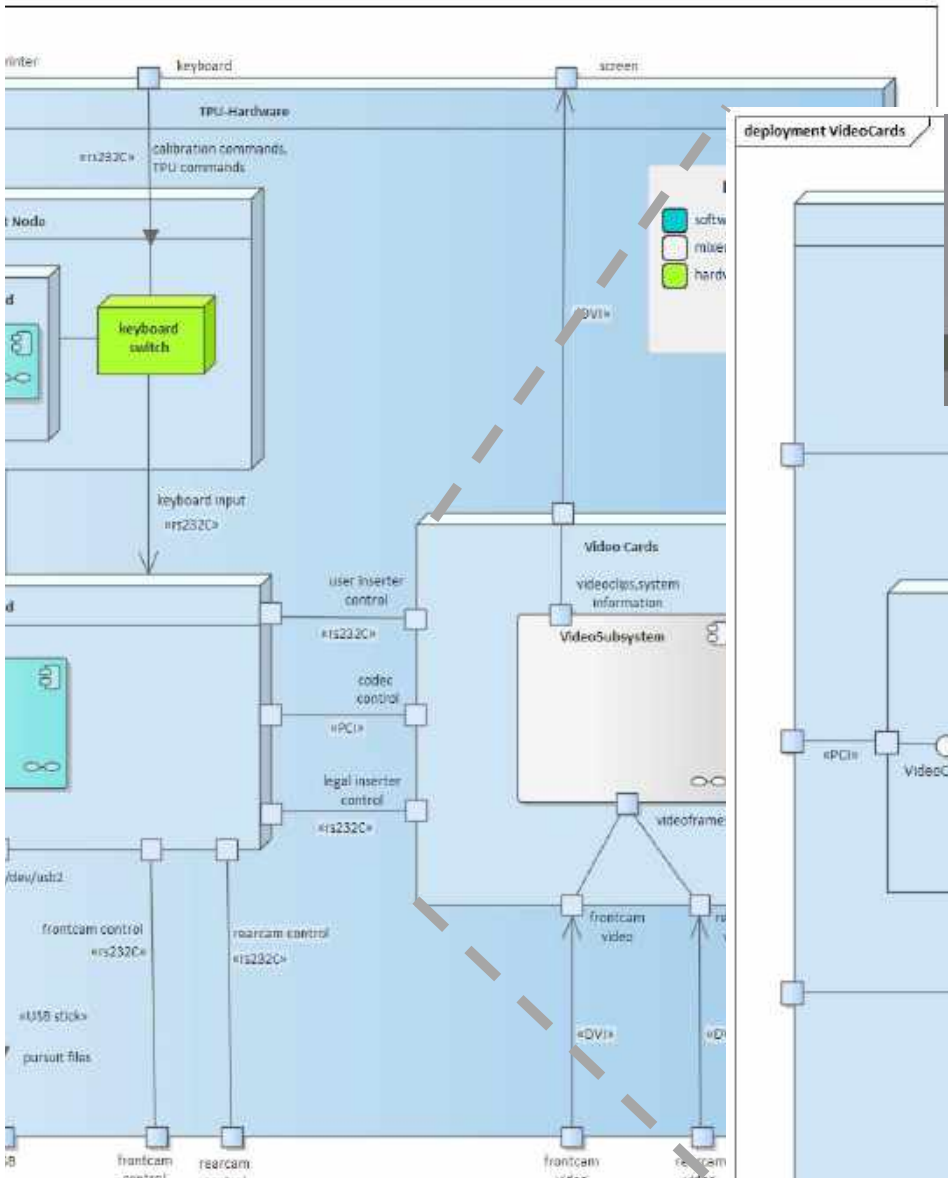

**Deployment Diagram Purpose (3):**
Map the functionality (i.e. the building blocks) onto the nodes

Still mixed technology (HW+ SW)



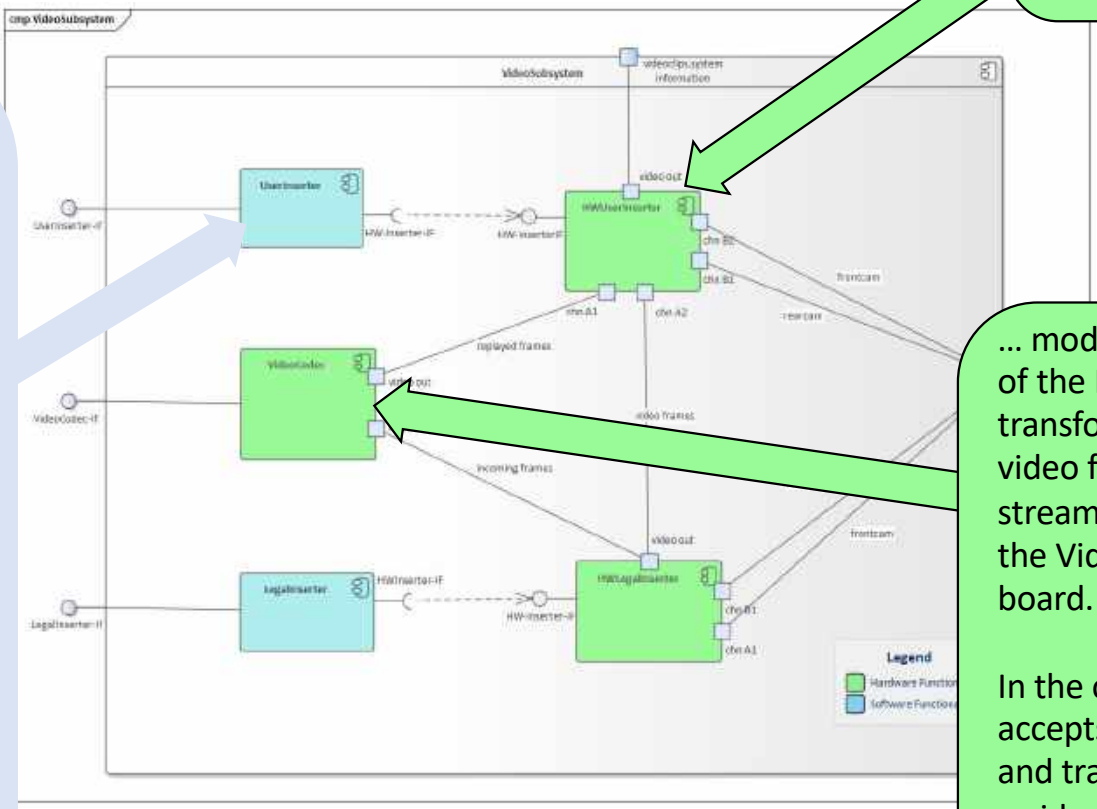

Note that the interfaces are absolutely consistent with the (business) context diagram

# Let us zoom in a little bit




# Extracting the Building Block View Only (for the VideoSubsystem) and Specifying the Responsibility of Each Block

… augments the video stream by inserting text and renders it for the display. This text is only shown to the police officer and is not stored in the video file.
Depending on the current operating mode, this component displays graphical elements for the management user interface or it shows current state information.
The latter contains all relevant current information for the police officer. This information is displayed in a big font for good readability. This data contains: date and time, current speed, available satellites, free storage info for the hard disk and the USB-stick, information of active copy operations to the USB-stick. During a pursuit, additionally the time and the length of the pursuit and the current maximum violation is displayed.




… receives videoframes from various sources and augments them by inserting text according to the controls from the UserInserter.

… models the functionality of the HW Codec. It transforms the incoming video frames into a digital stream that is captured by the VideoUnit on the PC-board.

In the opposite direction it accepts digital streams and transforms them into a video stream to be displayed.

1 Intro & Goals
2 Constraints
3 Context
4 Solution Strategy
5 Building Block View
6 Runtime View
7 Deployment View
8 Concepts
9 Design Decisions
10 Quality Scenarios
11 Risks/Technical Debts
**12 Glossary**

| Term | Definition |
|---|---|
| Frame Interval | the smallest period for data sampled by the TPU. It corresponds to the duration of one video frame, which is 40 milliseconds for a frequency of 25 videoframes per second |
| Distance Per Frame | the distance (in millimeters) travelled during one frame |
| Distance Per Second | the distance (in millimeters) travelled during one second |
| … | |
| Pursuit Step | the measured and calculated data for one second within a pursuit. It consists of pursuit duration, distance, current speed, current maximum violation, … |
| … | |
| Video Frame | one picture within a video stream. Usually, we have 25 frames per second. |

Don’t forget to define your **key domain terms**

arc42

1 Intro & Goals
2 Constraints
3 Context
4 Solution Strategy
5 Building Block View
**6 Runtime View**
7 Deployment View
8 Concepts
9 Design Decisions
10 Quality Scenarios
11 Risks/Technical Debts
12 Glossary

Purpose: Use scenarios to …

1. find missing building blocks
2. extend, modify existing building blocks
3. prove that everything works fine ☺
4. **communicate your architecture**

**There are many different notations to develop & communicate scenarios:**

Examples: Plain Text
Simple Activity Diagrams, Sequence Diagrams,
Communication Diagram, StateCharts,
Complex Activity Diagrams showing async coop.

**Pick the notation that supports your design considerations best or that is most convincing for your stakeholders**

# 6.1 Scenario: Measurement Propagation (Speedometer Pulses, Distance, …)

e.g: plain text

Receive speedometer pulses

For every *frame interval* (40ms)

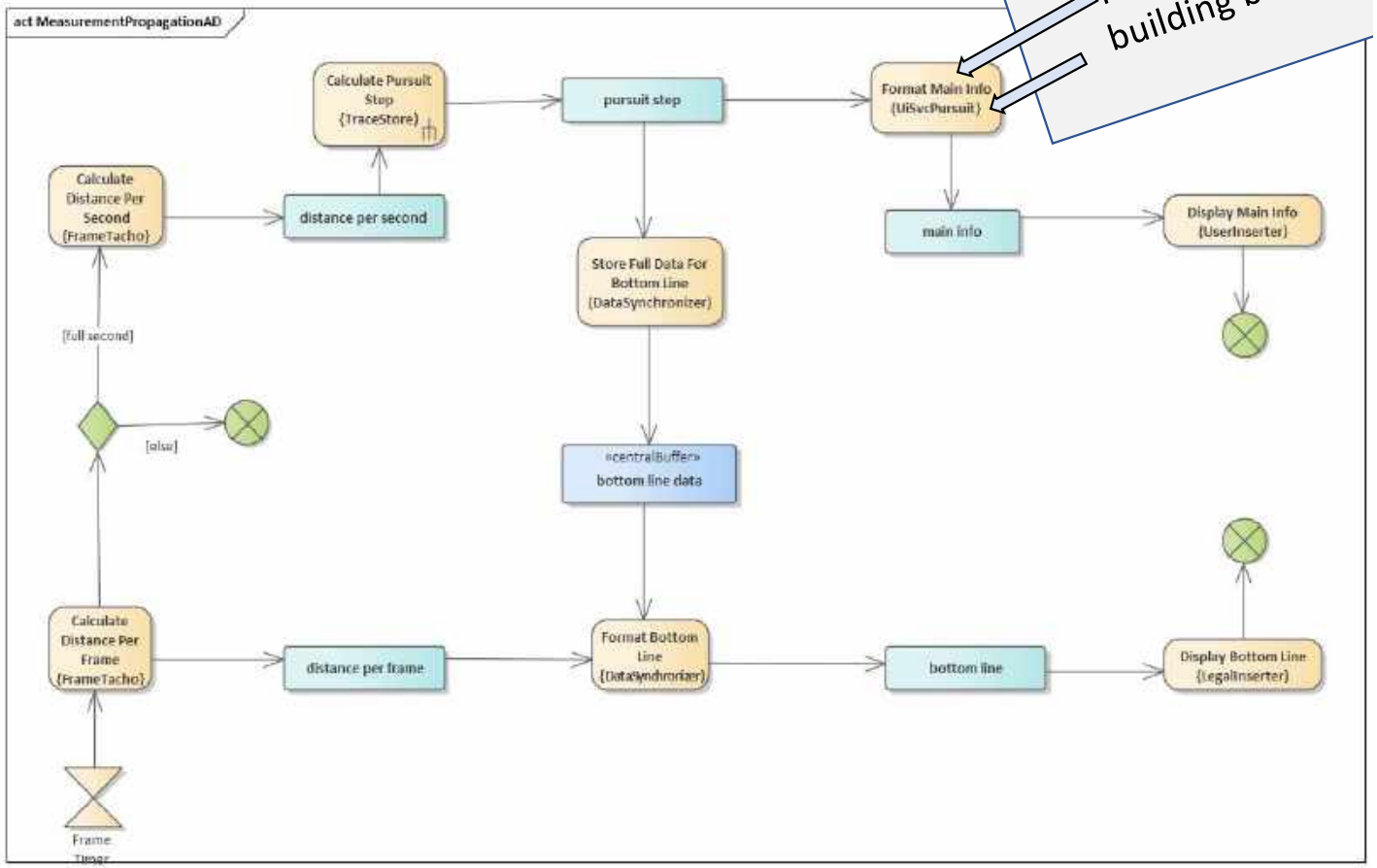
- calculate *distance per frame*
- display *bottom line* in the **LegalInserter**

Every full second

- Calculate *distance per second* and all other data for the *pursuit step*
- Update display on **UserInserter** and **LegalInserter**

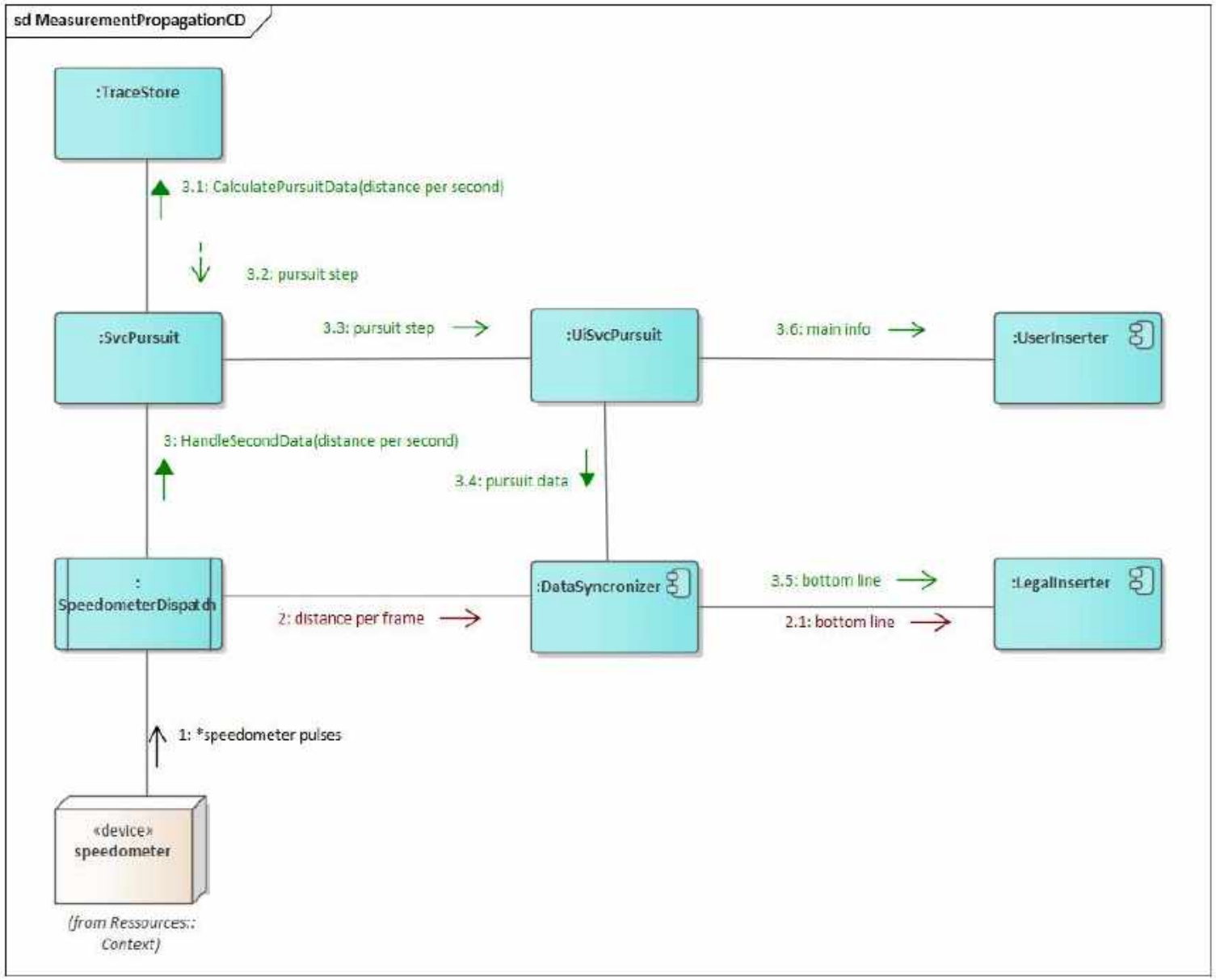
Note the consistent use of *defined terms* and **building block names**

… or as simple UML activity diagram




Note the alignment of functions with building blocks

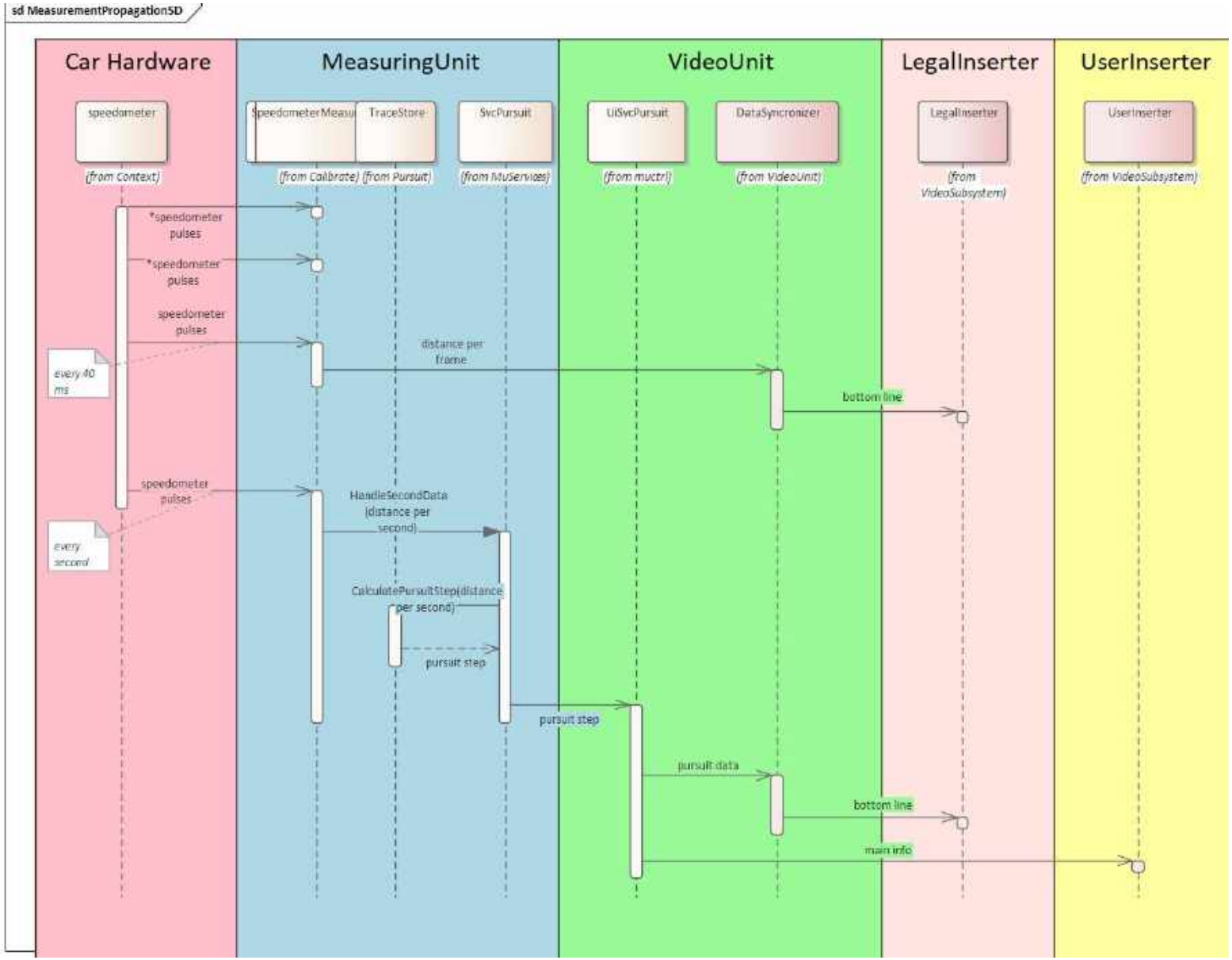
… or as UML Communication Diagram




Note the color code:

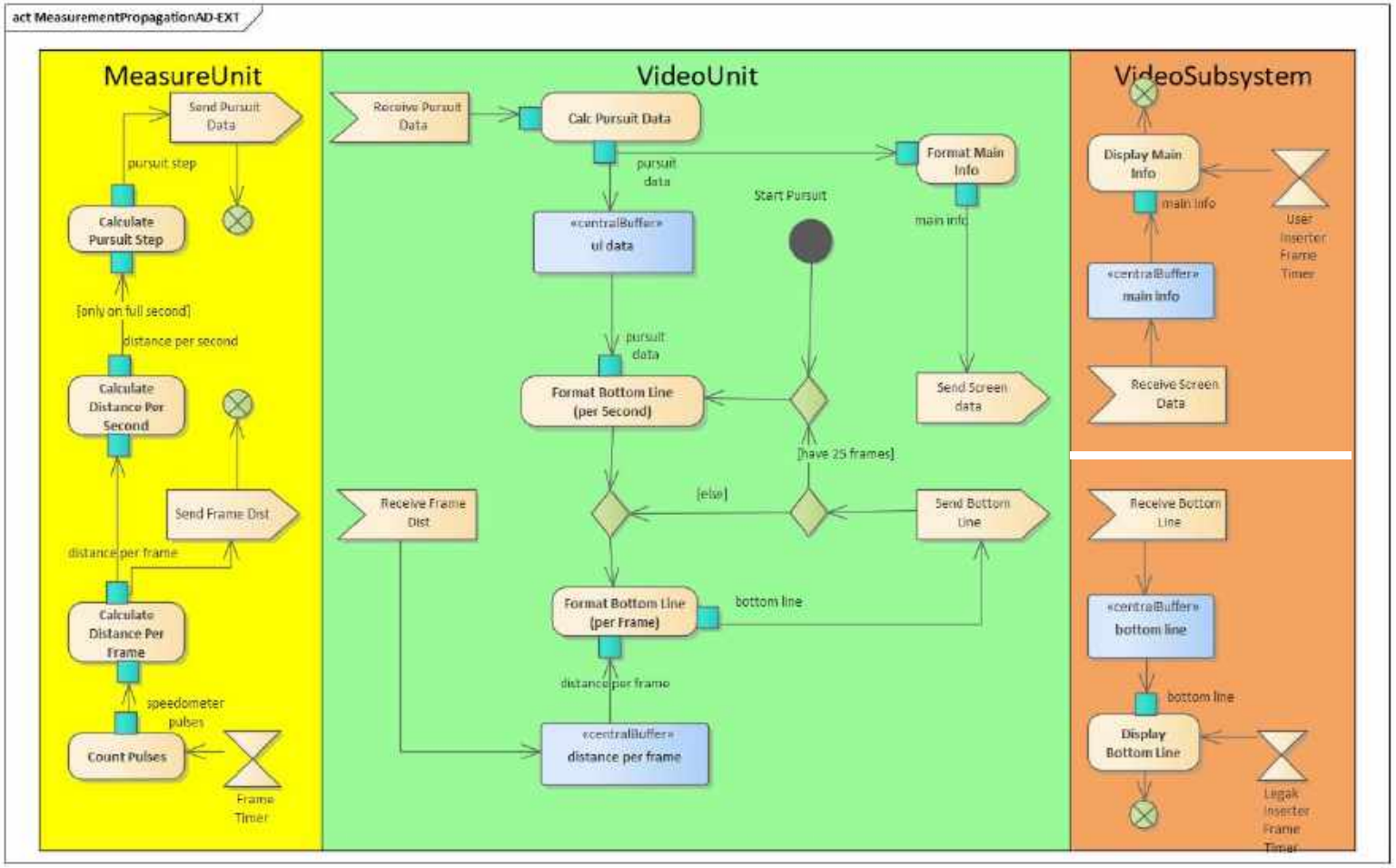
Black message: trigger
Red messages: every 40 ms
Green messages: every second

… or as UML Sequence Diagram overlaid with Hardware Nodes



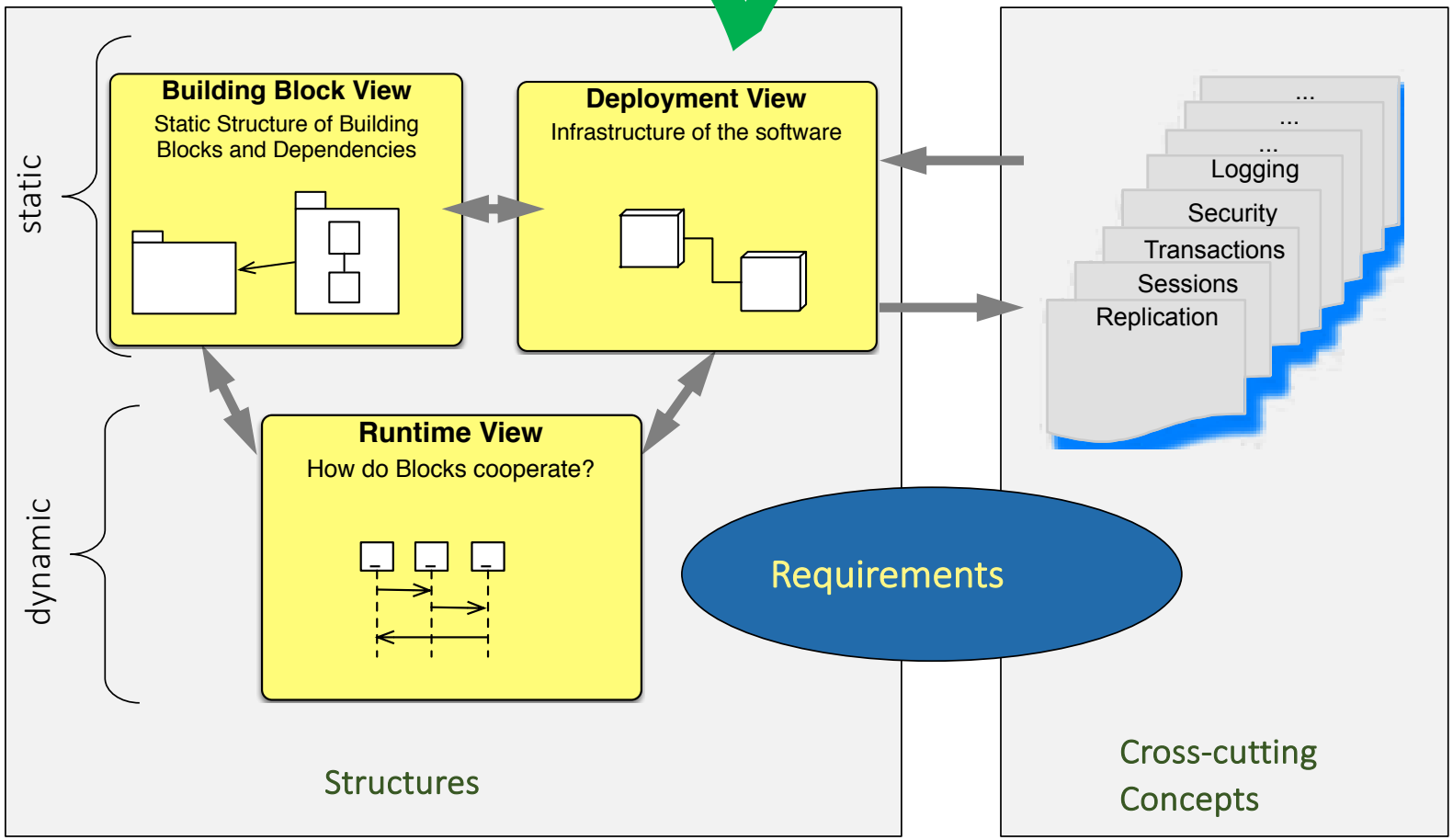
… or as UML Activity Diagram with Level-1 BuildingBlock-Swimlanes

concentrating on async communication between hardware nodes



Documenting Architecture...
... with different views

... and crosscutting concepts



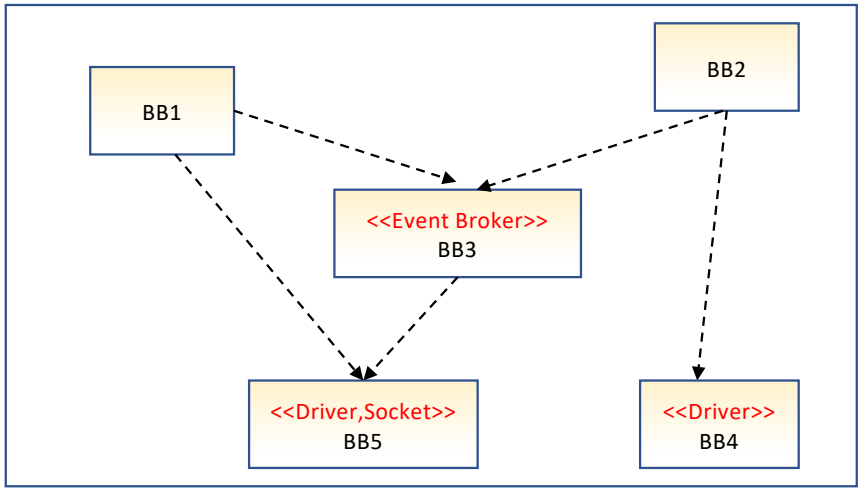
arc42

1 Intro & Goals
2 Constraints
3 Context
4 Solution Strategy
5 Building Block View
6 Runtime View
7 Deployment View
**8 Concepts**
9 Design Decisions
10 Quality Scenarios
11 Risks/Technical Debts
12 Glossary

1. **Explain cross-cutting concepts once** (in chapter 8)
   - with concrete technology stacks
   - with examples
   - with rules for implementation

8.x Drivers
.....
..... .....
8.y EventBroker
.....
..... .....
8.z Socket Communication
.....
..... .....

2. **Use concepts** by linking them to building blocks,
   e.g. via stereotypes



# e.g. Event Handling

In a system with several components generating a variety of events, that should be handled by appropriate event handlers, **we face the following problems:**

**1.** In a robust and flexible architecture, components that generate events should be kept free of any knowledge about the handling of these events (separation of concern). So they are not able to deliver the events they produce to the right destinations. An appropriate event propagation concept is needed.

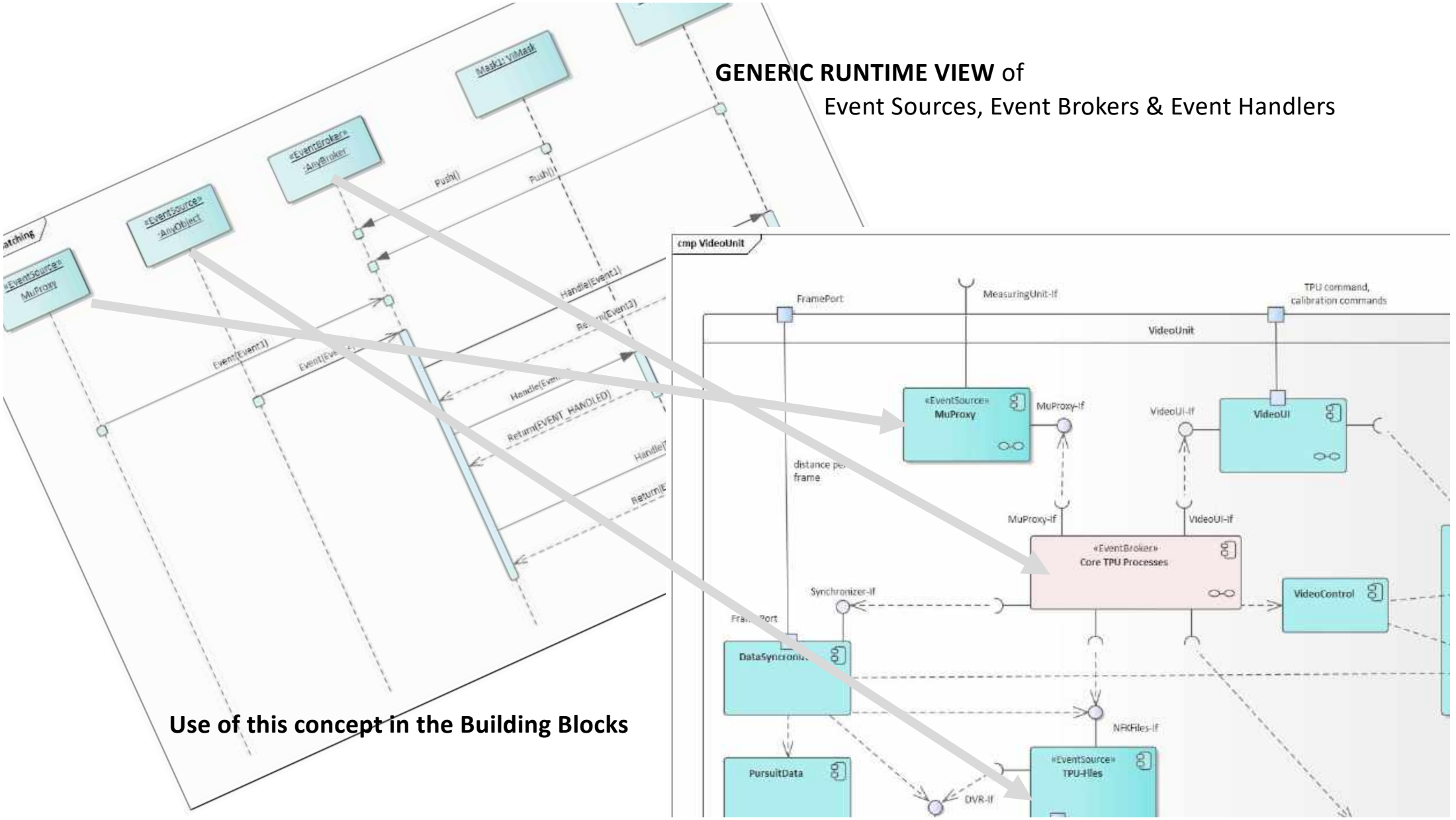
**2.** An event handler in general does not want to be prepared to handle a newly arriving event while just handling one event. Especially if the handling of the current event may result in a change of state leading to a different handling of successive events.

**SOLUTION:**

Within the TPU many different events are generated partly as a result of measurements on a periodic time base and partly asynchronously like user input or special system events (e.g. disk space warning). Different Building blocks act as **events sources.** The events are **managed by an event broker in a central event queue for sequential delivery on one single thread.**

On that thread the events are delivered to **event handlers**, which are **organized in a stack**. The first queued event is offered to the topmost event handler on the stack. The handler can decide to consume it, or return the same or a different event for delivery to the event handler located on the previous stack position.

**GENERIC RUNTIME VIEW** of
Event Sources, Event Brokers & Event Handlers




**Use of this concept in the Building Blocks**




# One more thing: Design Decisions

**Efficient Calculation**

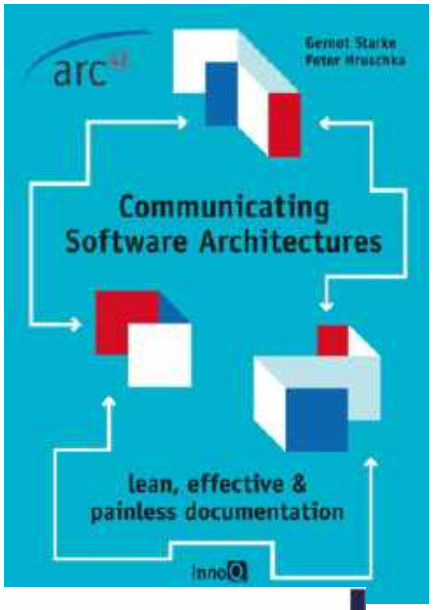
In order to perform calculations as quickly as possible and to save CPU power, all calculations are performed with integers (fixed point). Float calculations are only used if unavoidable (e.g. root extraction for Pythagorean theorem).

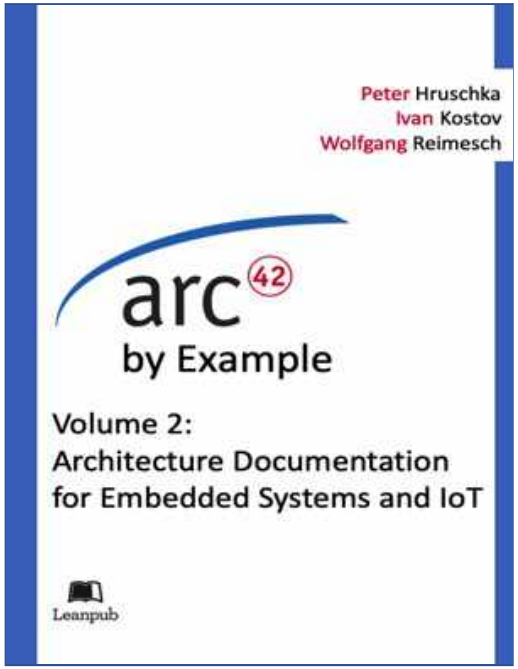
**Buffering of Video Information**

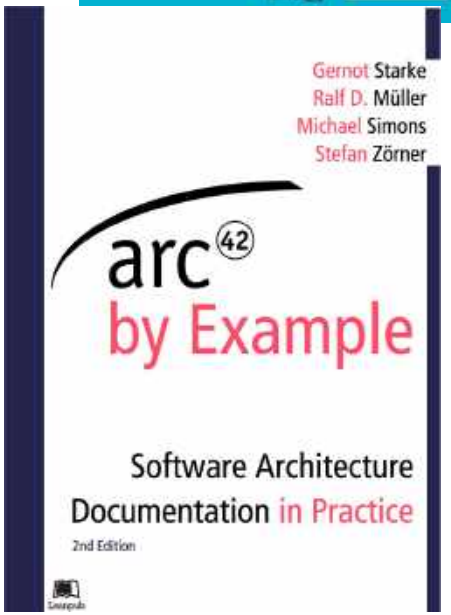
The propagation of measurement data from the MeasuringUnit via the VideoUnit to the VideoSubsystem has to pass via serial lines and may suffer variable delays during the processing within the different nodes. We decided to achieve a fixed delay, such that the data rendered on the screen exactly represents the real time behavior shifted by a fixed time delay. This is achieved by keeping a buffer of video information to be displayed in the LegalInserter. The display of the first piece of information is delayed until the buffer has accumulated a reserve of 5 data items. This precaution assures that every video frame can be supplied with current data at the right time. The system has a constant delay of 200 milliseconds (5 frames * 40 ms time per frame), so the exact data corresponding to video frame number n can be found in the data inserted into frame number n + .

# More Info

- G. Starke, P. Hruschka:
  Communicating Software Architecture –
  lean, effective & painless documentation (Leanpub)
- www.arc42.org
  - The home of the arc42 template
- faq.arc42.org
  - 132 frequently asked questions about arc42
- arc42 by Example (Leanpub)
  - 6 real world examples of
    architecture documentation
- arc42 by Example Volume 2 (Leanpub)
  - 2 real world examples from embedded systems



YES, YOU CAN …

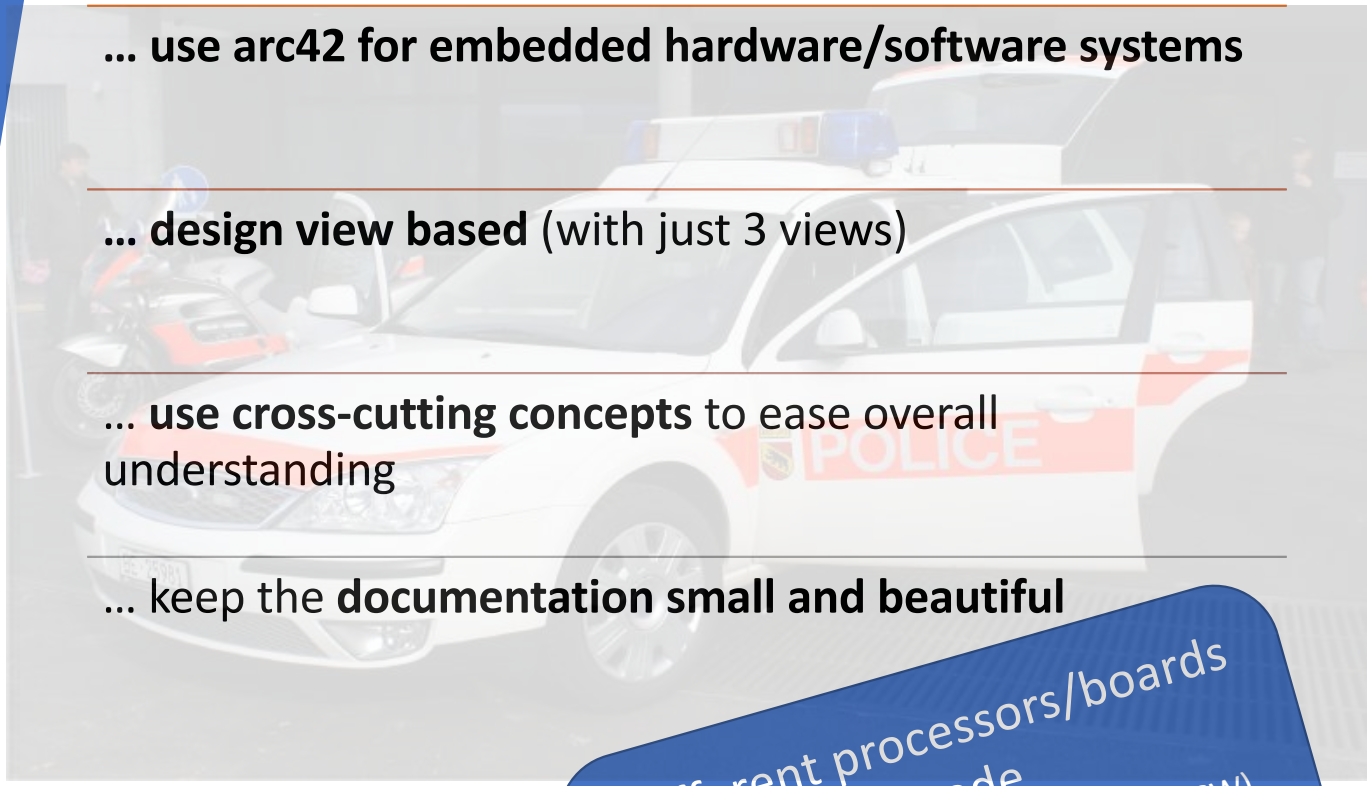
Thank you

… **use arc42 for embedded hardware/software systems**

… **design view based** (with just 3 views)

… **use cross-cutting concepts** to ease overall understanding

… keep the **documentation small and beautiful**




7 different processors/boards
200 KLOC C++ code
60 person months (20 HW, 40 SW)

**Just 42 pages of docu ☺**