

ASAM MDF - New Features in Release V4.2.0

Tobias Langner (ETAS)

tobias.langner@etas.com

9. Oktober 2020
München



Agenda

1	Introduction
2	Column-oriented storage
3	Event signals
4	Signal reduction changes
5	

Introduction

Introduction

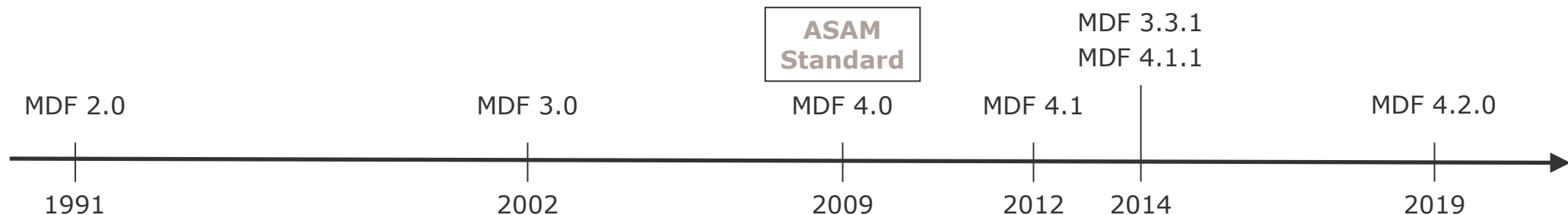
MDF – Measurement Data Format

- Binary file format to store measured or calculated data for post-measurement processing and long-term conservation.
- Common sources of the data to be stored are sensors, ECUs or bus monitoring systems.
- With MDF a high performance can be achieved for both writing and reading signal data.
- In addition to the plain measurement data, MDF also contains descriptive and customizable meta data within the same file.

Introduction

History

- ▶ **1990**: MDF designed for use in the automotive industry
- ▶ **1991** until today: MDF versions 2.x and 3.x have successfully been used over many years and evolved to a de facto standard
- ▶ **2009**: release of ASAM Common MDF 4.0.0 as result of a major update of the format and standardization by ASAM e.V.
- ▶ **2012**: release of ASAM Common MDF 4.1.0 including three new associated standards
 - ▶ most important new features: compression of data, bus logging
- ▶ **2019**: release of ASAM Common MDF 4.2.0
 - ▶ including new way to store data for enhanced read performance



Introduction

Key Concepts of MDF

- ▶ Compact binary format organized in loosely coupled blocks
- ▶ Measurement data stored in records according to sampling rate
- ▶ Record layout and general signal description given by channels
- ▶ Supports multiple and non-periodic sample rates
- ▶ Synchronization via master channel concept
- ▶ Special data types and meta information used in automotive area
- ▶ Data received (e.g. from ECU) can be stored "as is"
- ▶ Conversion rules for calculation of physical values from stored raw values
- ▶ Extension of meta information by XML or "attachments"
(embedding or linking of other files)

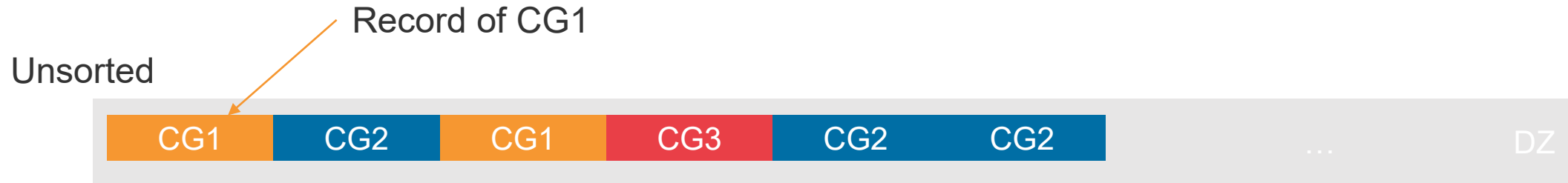
Backward Compatibility

- ▶ **ASAM MDF 4.2.0 is an extension of ASAM MDF 4.0.0**
 - ▶ Backward compatibility: every valid MDF 4.0/4.1/4.11 file is also a valid MDF 4.2 files
 - ▶ Forward compatibility: old tools that only support MDF 4.0/4.1/4.11 should be able to read MDF 4.2 files while ignoring the new features.
 - ▶ Signal meta data for signals stored in column oriented storage is accessible in this case, but data isn't.

Column-oriented storage

Column-oriented storage

status quo - unsorted / sorted in MDF 4.x



Sorted



- Unsorted: well suited for fast writing
- Sorted: good for reading, acceptable for writing in most cases.

Column-oriented storage

sorted vs. column oriented

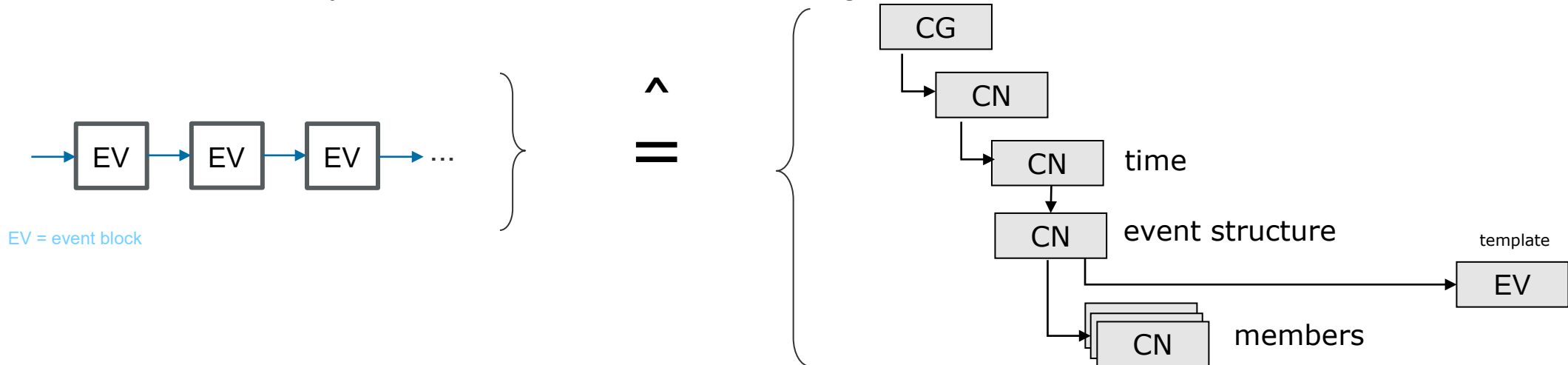


- Column oriented option allows the distribution of signals acquired at the same point in time across different storage locations.
- This can be used to
 - Optimize for fast access
 - Append data to a group without re-writing the file

Event signals

Event Signals

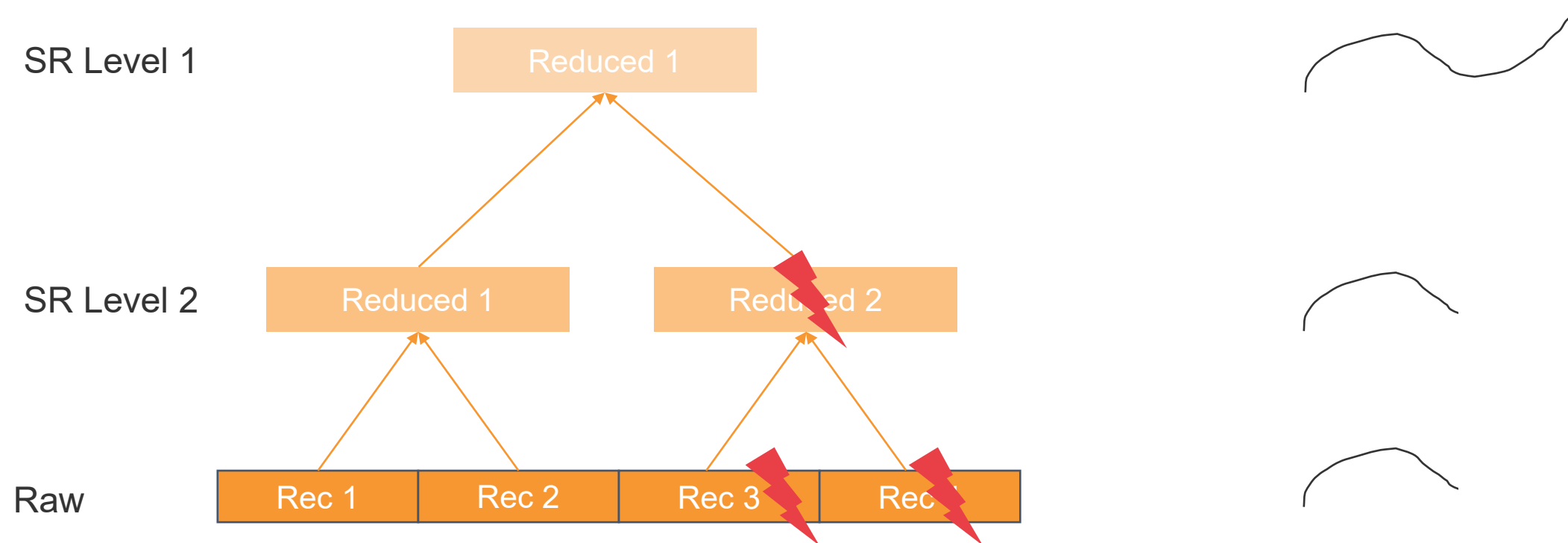
- ▶ Previously: events stored as linked list
 - ▶ OK for small number of events, but slow in case of large number (> 1000)
 - ▶ MDF 4.2 offers an alternative way of storing events in channels ("event signals")
 - ▶ store events of same type in a structure and use a "template" event
 - ▶ channels are a proven mechanism to handle millions of samples
 - ▶ now open for events as well
- => lose a little bit of flexibility for the benefit of more efficient reading



Dominant invalidation bit in sample reduction

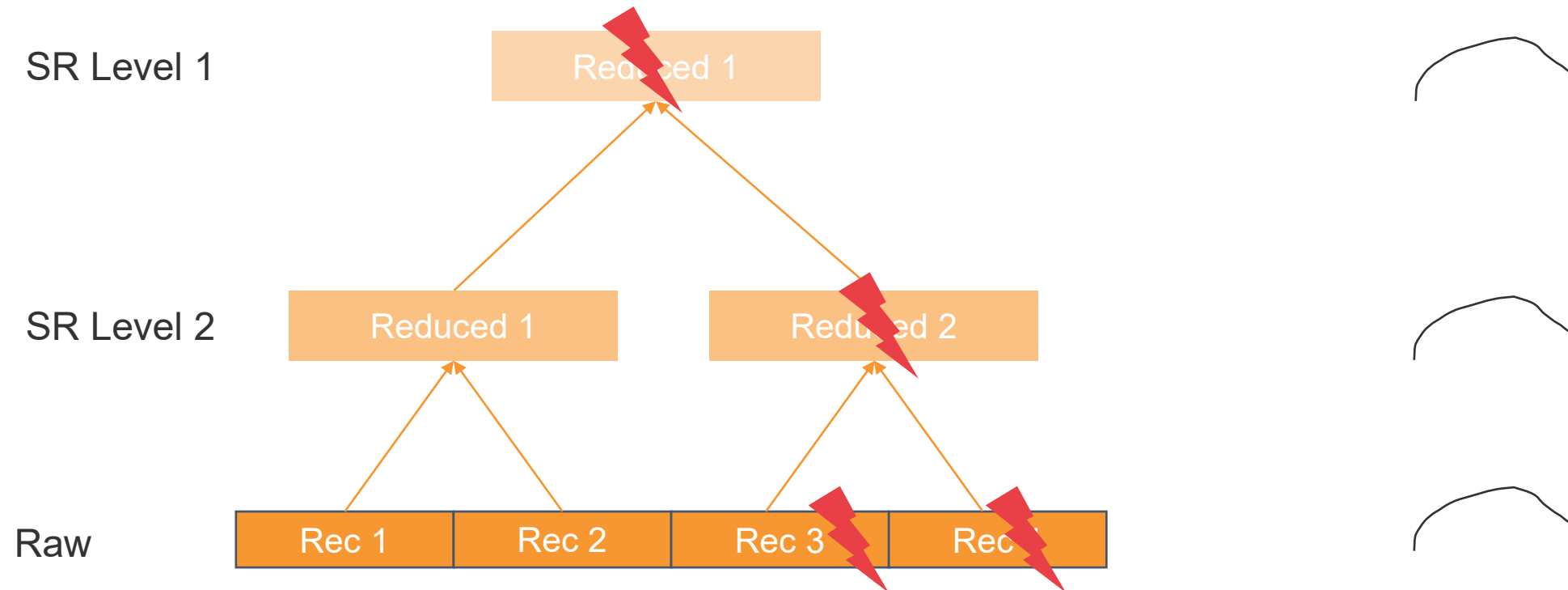
Dominant invalidation bit in sample reduction

old



Dominant invalidation bit in sample reduction

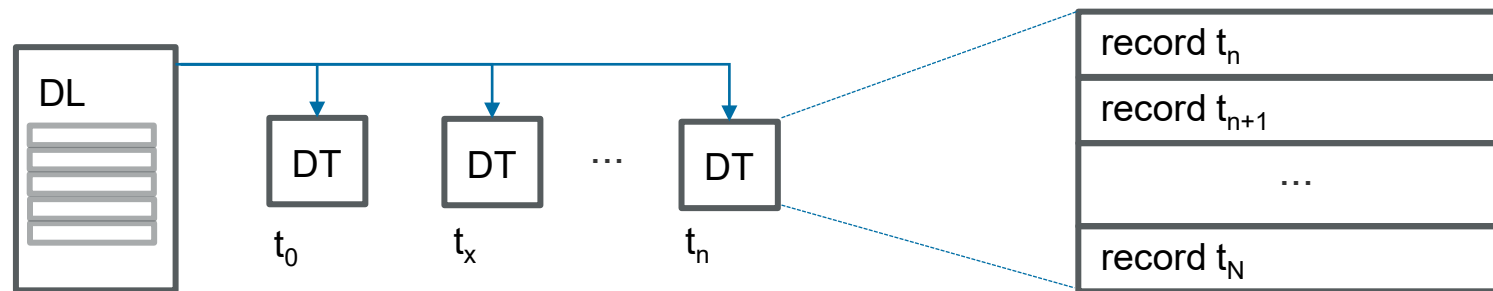
With dominant invalidation bit



Minor additions

Minor additions

- ▶ Store time stamps for single parts of a distributed data block
 - ▶ faster seek of a time stamp by determining relevant partial data block
- => avoids unnecessary reads of partial blocks (esp. for compressed data)



Minor additions

▶ New Data Types

▶ Complex Number

- ▶ enables standard way to store complex number (real and imaginary part)

▶ IEEE754 half precision float (2 bytes)

- ▶ market requirement for compact storage of float values
- ▶ retains MDFs capability to write data without transformation (write performance)

▶ New Conversion Rule

▶ Bitfield Text Table

- ▶ mapping of special conversion rule used in FIBEX / AUTOSAR

Thank you for your attention

Tobias Langner (ETAS GmbH)
tobias.langner@etas.com