RADAR STREAMING FORMAT FOR 1722

May 27, 2025

Abhijit Deb

Automotive Ethernet, NXP Semiconductors, Netherlands
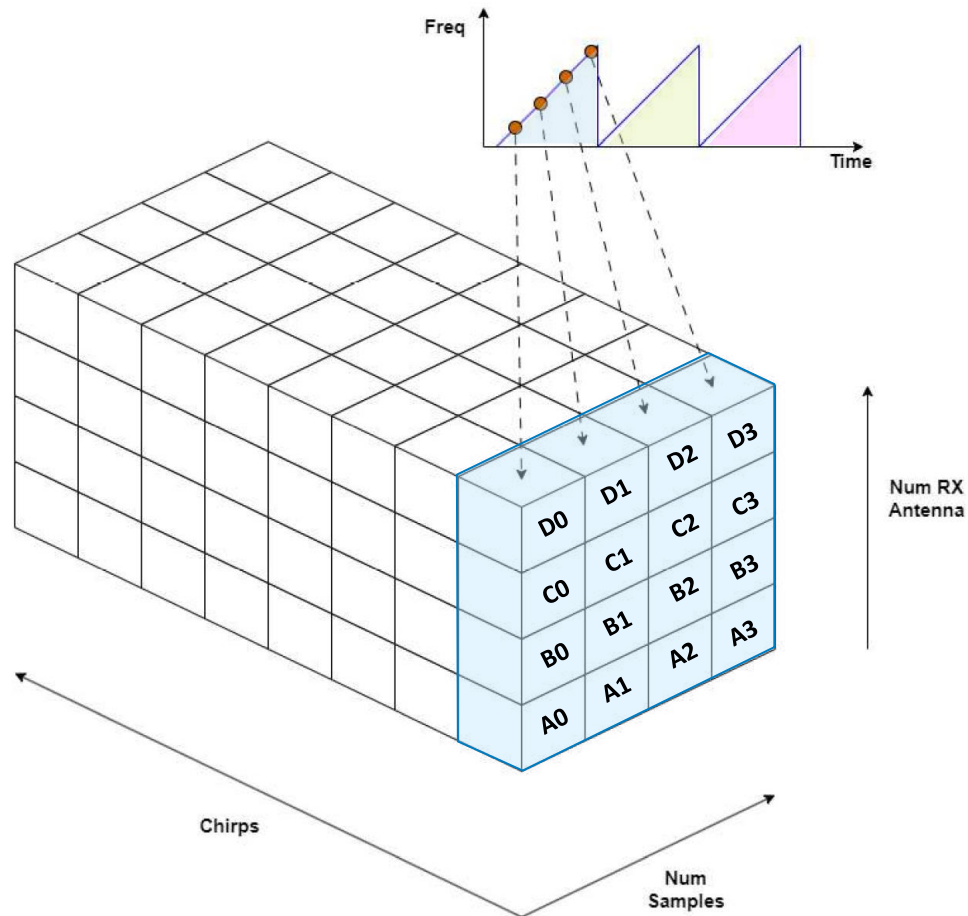
Adriaan Niess

Cross-Domain Computing Solutions, Bosch, Germany

BOSCH

NXP

OUTLINE

- Introduction
  - Sensor data organization
  - Chirp Data
  - Sensor Data Segmentation

- GISF ACF Message Format

- Proposed Radar Format
  - Renaming / re-purposing GISF fields

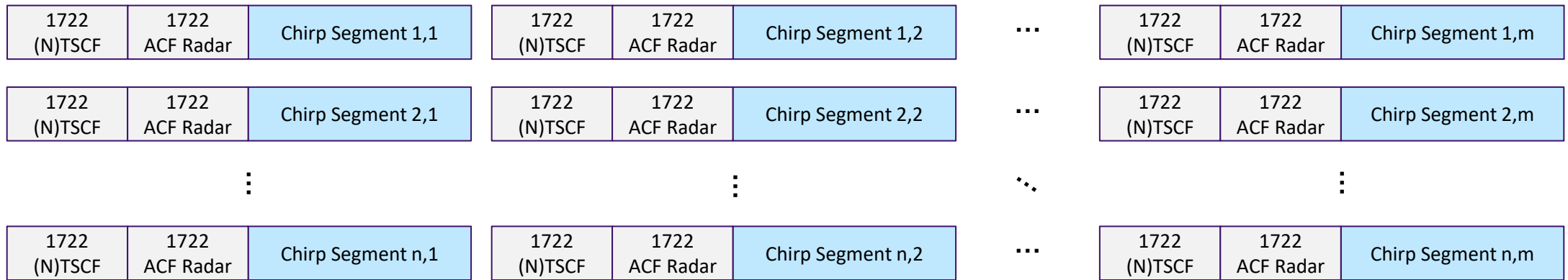- Conclusion

SENSOR DATA ORGANIZATION

- Current generation radar sensors perform local data processing and transmit an object-based data to an ADAS control unit

- Radar sensors need to support Ethernet → Data Streaming

- Radar data (before/after 1st FFT) interpreted as a cube
  - Number of Chirps
  - Number of samples (per chirp)
  - Number of RX antennas

- Analogy to image frames
  - Radar Cube → Image Frame
  - Radar Chirp → Image Line

- New ACF type proposal for radar encapsulation in 1722c
  - Based on the existing GISF format

CHIRP DATA

Chirp
(Row major order)

or

Chirp
(Column major order)

Chirp
(embedded in 1D space)

optional
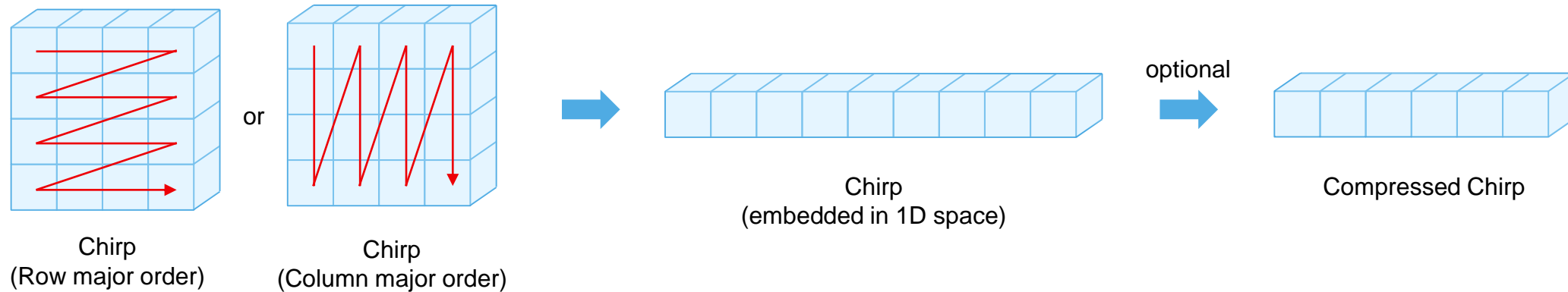
Compressed Chirp

- To reuse the existing GISF format, chirps must be encoded as 1D data
- This way chirps can be treated the same way as the lines of an image
- Chirps might (optionally) be transmitted in a compressed form

SENSOR DATA SEGMENTATION

1722 (N)TSCF	1722 ACF Radar	Chirp Segment 1,1

1722 (N)TSCF	1722 ACF Radar	Chirp Segment 1,2

$\cdots$

1722 (N)TSCF	1722 ACF Radar	Chirp Segment 1,m

1722 (N)TSCF	1722 ACF Radar	Chirp Segment 2,1

1722 (N)TSCF	1722 ACF Radar	Chirp Segment 2,2

$\cdots$

1722 (N)TSCF	1722 ACF Radar	Chirp Segment 2,m

1722 (N)TSCF	1722 ACF Radar	Chirp Segment n,1

1722 (N)TSCF	1722 ACF Radar	Chirp Segment n,2

$\cdots$

1722 (N)TSCF	1722 ACF Radar	Chirp Segment n,m

- Chirps are expected to be bigger than 1.5KB and need to be split into m segments
- A radar cube consisting of n chirps therefore would be split up into a total of $(n \times m)$ segments
- Counters for n, m will make reuse of GISF's line_number and i_seq_number fields
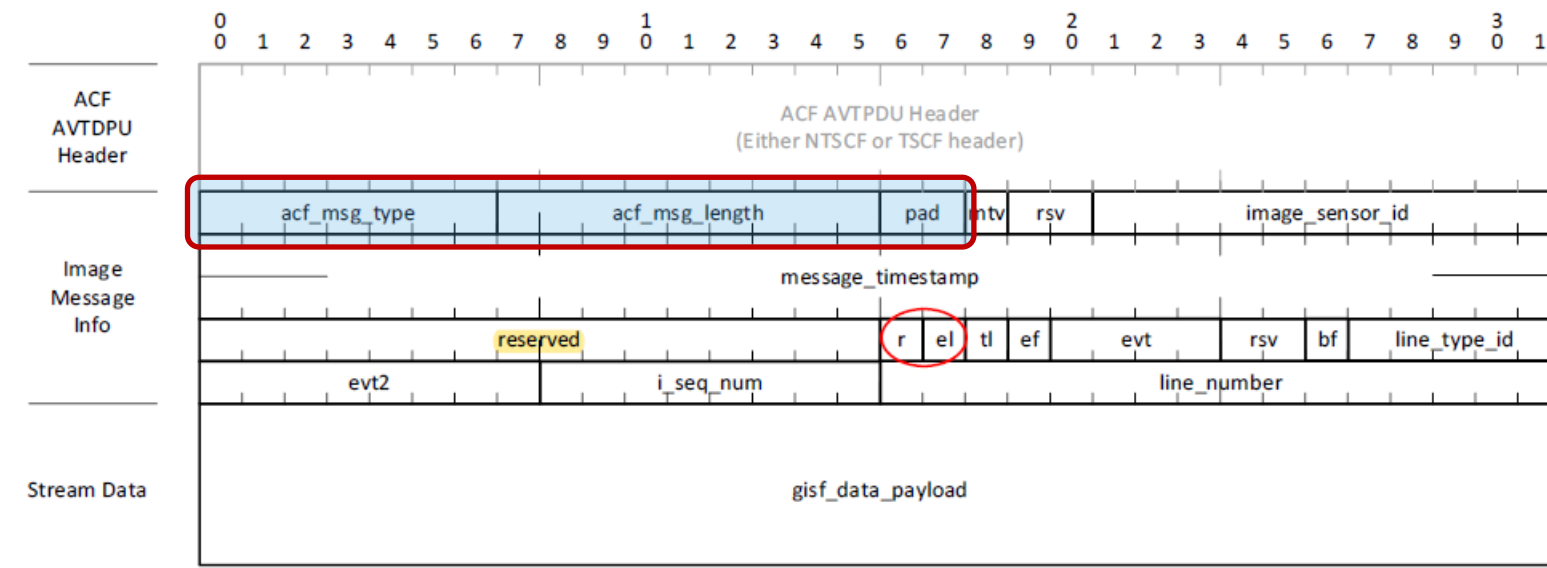
BOSCH

RADAR FORMAT

Figure 112 —GISF ACF message

- **acf_msg_type**
  - Type of message contained in the payload
  - 7-bit: use one of the reserved value → *0x13 to 0x20 or, 0x24 to 0x75*

- **acf_msg_length**
  - 8-bit: number of quadlets of data contained in the message (including the quadlet containing the acf_msg_length field and all subsequent quadlets including the last quadlet of acf_msg_payload)

- **pad**
  - 2-bit: number of padding bytes in the last quadlet of payload (to make it an integer multiple of quadlets)
  - Shall be 0 except for the last packet of a chirp
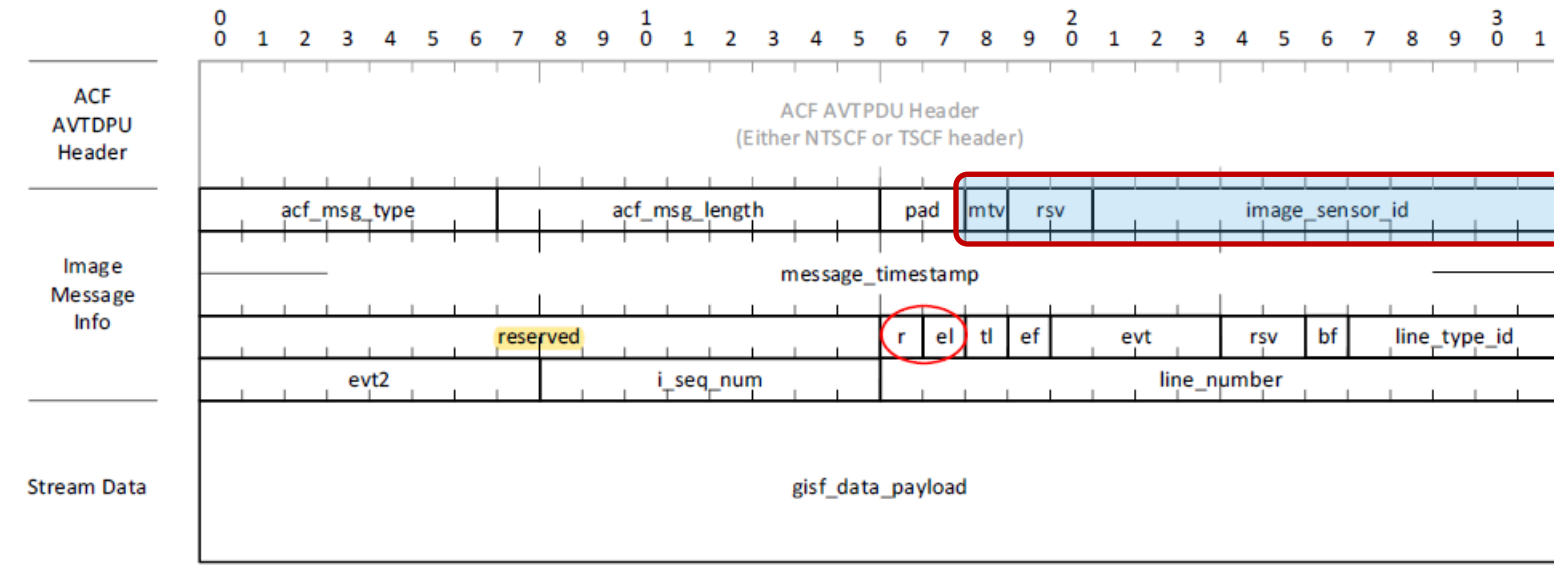
RADAR FORMAT

Figure 112 —GISF ACF message

- **mtv**
  - 1-bit: message_timestamp valid indicates whether the 8-octet message_timestamp field contains valid data
  - '1' means that the message contains valid timestamp

- **rsv**
  - 2-bit: reserved

- **radar_sensor_id (~~image_sensor_id~~)**
  - 11-bit: customer defined ID of RADAR sensor (application dependent, not defined by standard)
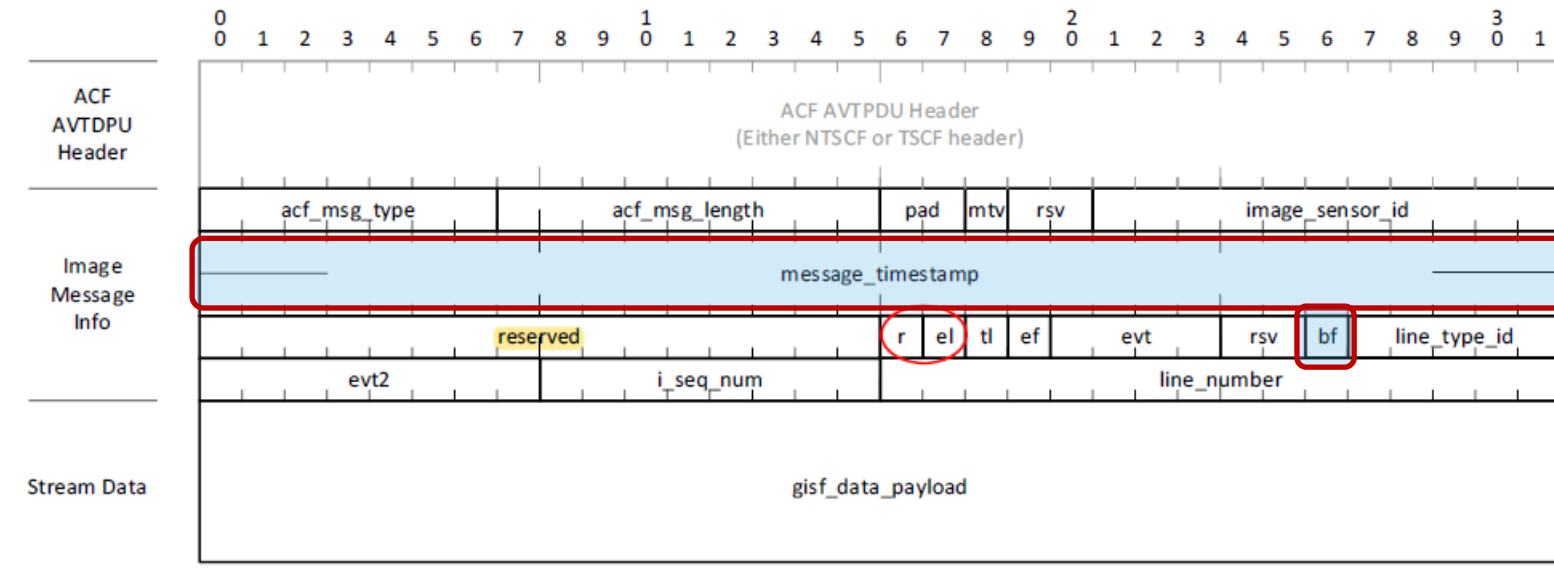
RADAR FORMAT

Figure 112 —GISF ACF message

- **message_timestamp**
  - acquisition time of the first sample of *either the chirp or the cube → received chirp*
  - 64-bit: GPTP time in nanoseconds

- **tc (bf)**
  - timestamp per chirp or per cube
  - '0': timestamp per cube
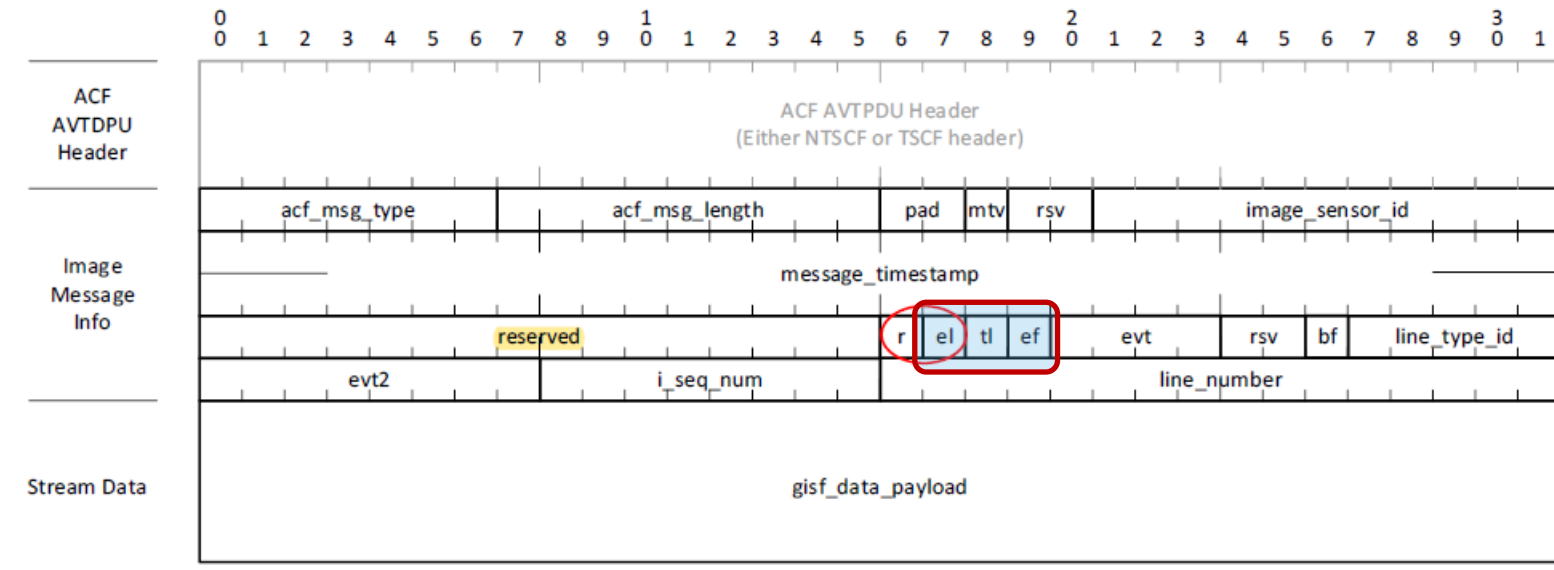 - '1': timestamp per chirp

RADAR FORMAT

Figure 112 — GISF ACF message

- **bb (el)**
  - 1-bit: begin of cube identifier
  - '1' → indicates the first AVTPDU (first segment) of a radar cube

- **ec (tl)**
  - 1-bit: end of CHIRP identifier
  - '1' → Last AVTPDU of the CHIRP → last segment, if CHIRP is segmented over several Ethernet frames

- **eb (ef)**
  - 1-bit: end of cube identifier
  - '1' → indicates the end of the CHIRP sequence → Last AVTPDU of the cube, when segmented over several Ethernet frames
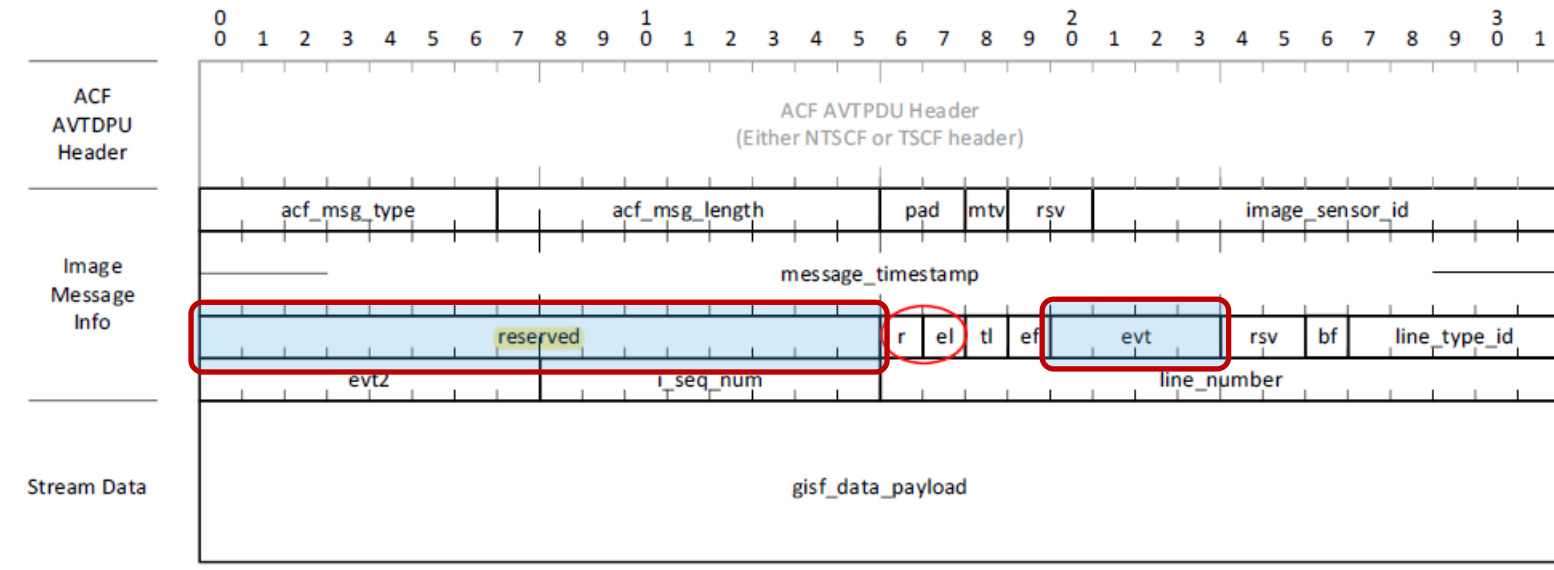
RADAR FORMAT

Figure 112 —GISF ACF message

- **seg_offset (~~reserved~~)**
  - 16-bit: segment offset within the CHIRP in bytes
  - Maximum → 64kB-1

- **evt**
  - 4-bit: indicates sensor event / CHIRP profile number (pointer in the Table that gives detailed parameter of the CHIRP)
  - Events → reconfiguration, restart, failure since previous packet sent, etc.
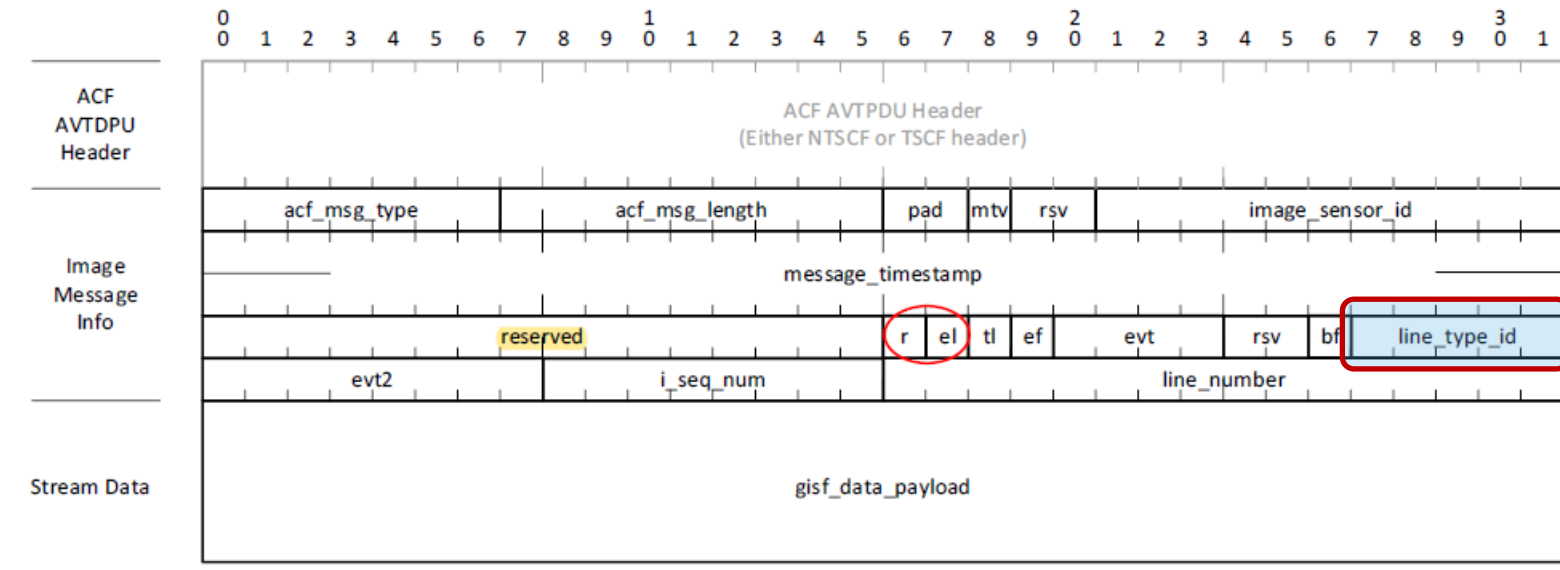
RADAR FORMAT

Figure 112 — GISF ACF message

- **data_type_id_1 (~~line_type_id~~)**
  - 5-bit: describes the chirp profile ID of the configuration table
    - The same configuration table is created in both the sensor node and in the ECU during the initial configuration phase
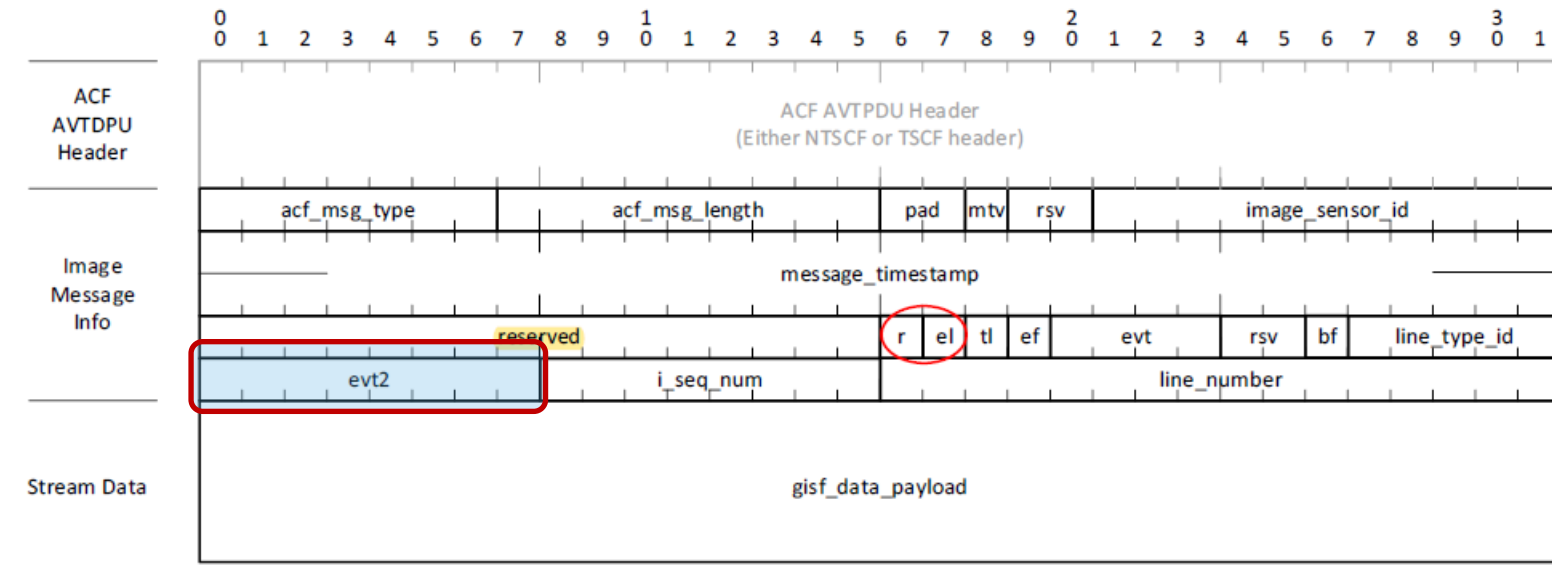
RADAR FORMAT

Figure 112 —GISF ACF message

- **data_type_id_2 (~~evt2~~)**

    – 8-bit: defines the payload data type using enumeration
      - ADC data   → 12 values (8/10/12/14/16/32-bit; real + complex)
      - FFT data   → 6 values (8/10/12/14/16/32-bit; complex)
      - FFT compressed   → 8 values (vendor dependent)
      - Interleaved/non-interleaved → 2
      - Total nr of channels   →
      - Reserved
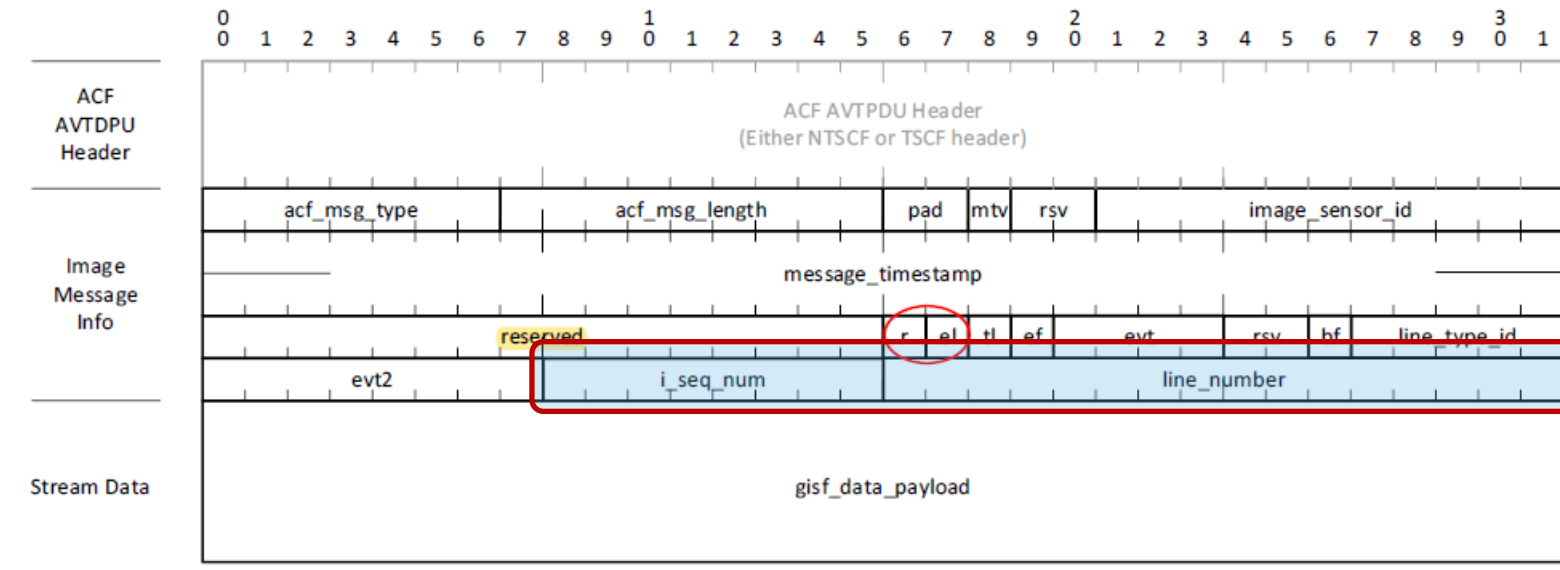
    – Enumeration table - tbd

RADAR FORMAT

Figure 112 —GISF ACF message

- **r_seq_num (i_seq_num)**
  - 8-bit: sequence number for segmented packets within a CHIRP
  - Starts with 0 for the first segment of each CHIRP and increments with every additional segment within the CHIRP

- **chirp_num (line_number)**
  - 16-bit: CHIRP number ➔ more than enough (more than 4k CHIRPS are not foreseen ➔ 12bits are ok)

CONCLUSION AND NEXT STEPS

- Radar format proposed
  - Based on GISF ACF format
  - Encapsulate the necessary radar information

- Review and feedback welcome

- Next steps
  - Feedback from working group participants
  - Alignment and consolidation

BOSCH

SECURE CONNECTIONS
FOR A SMARTER WORLD