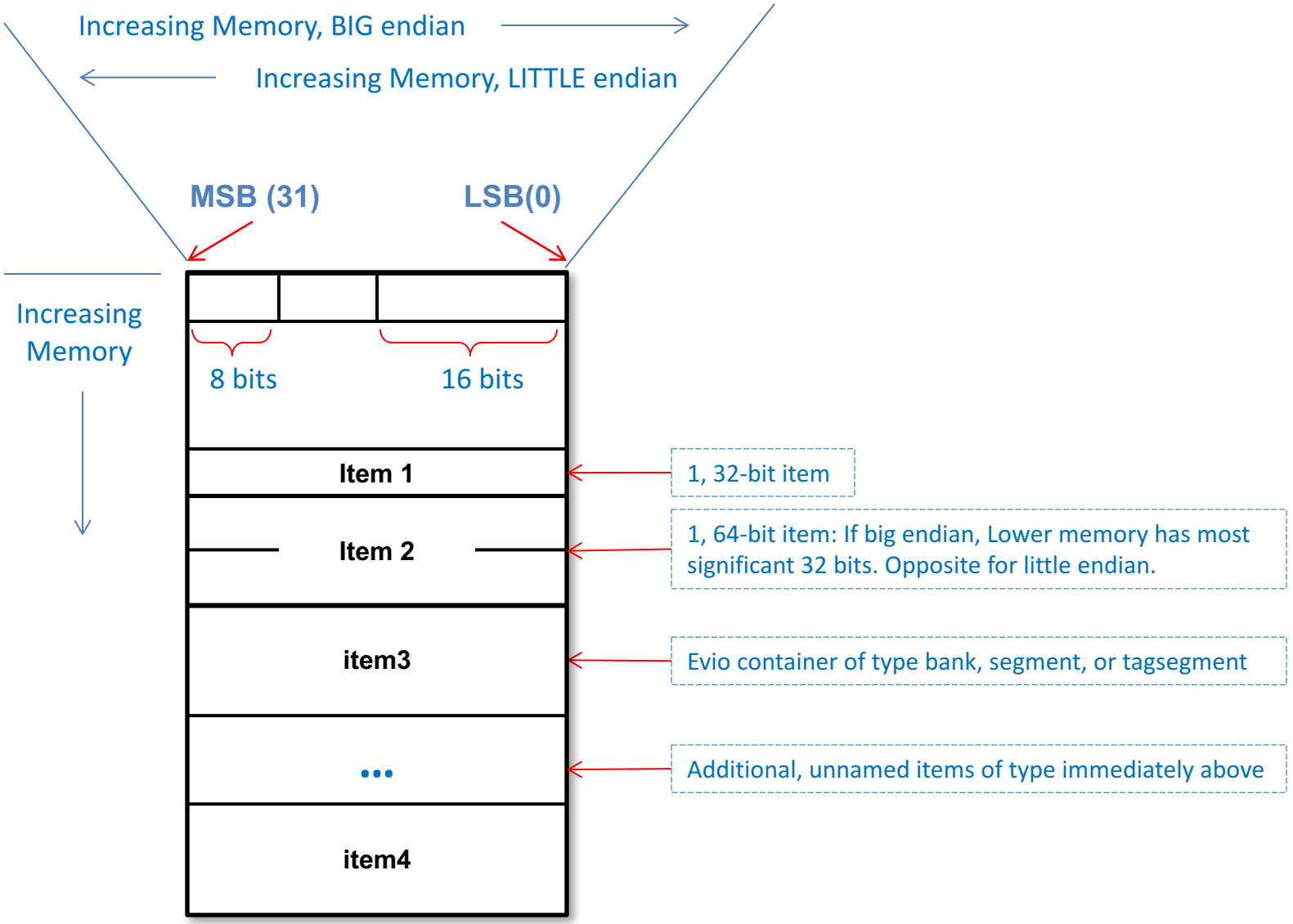
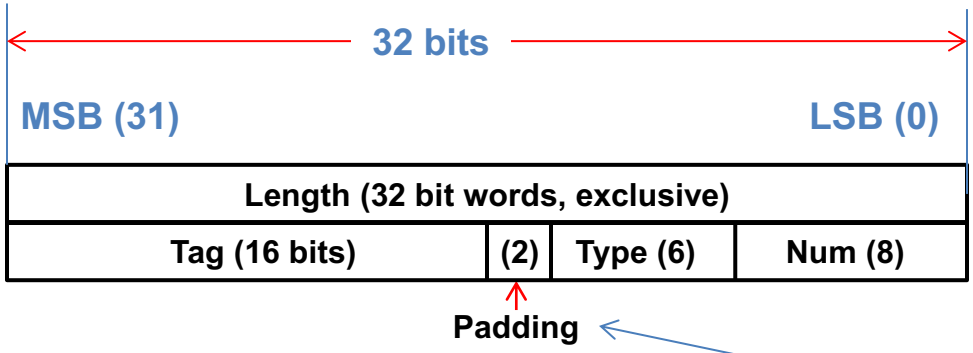
CODA Online Data Formats

Key to Reading Data Layouts



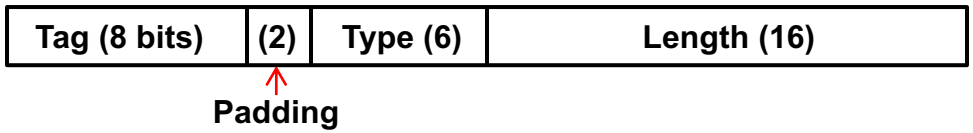
Evio Header Formats

Bank :



Number of unused bytes at end of following data if not a multiple of 32 bits.
For shorts, it is 0 or 2.
For chars (not strings), it is 0, 1, 2, or 3

Segment :



Tag Segment :

Evio Content Type Codes

Content Type	Primitive Data Type	Content Type	Primitive Data Type
0x0	32 bit unknown (not swapped)	0x21	Hollerit (Composite data internal)
0x1	32 bit unsigned int	0x22	N value (32 bit int, Composite data internal)
0x2	32 bit float	0x23	n value (16 bit int, Composite data internal)
0x3	8 bit char* (string)	0x24	m value (8 bit int, Composite data internal)
0x4	16 bit signed int		
0x5	16 bit unsigned int		
0x6	8 bit signed int		
0x7	8 bit unsigned int		
0x8	64 bit double		
0x9	64 bit signed int		
0xa	64 bit unsigned int		
0xb	32 bit signed int		
0xc	Tag Segment		
0xd	Segment		
0xe	Bank		
0xf	Composite		
0x10	Bank		
0x20	Segment		

Block Header (evio format versions 1-3)

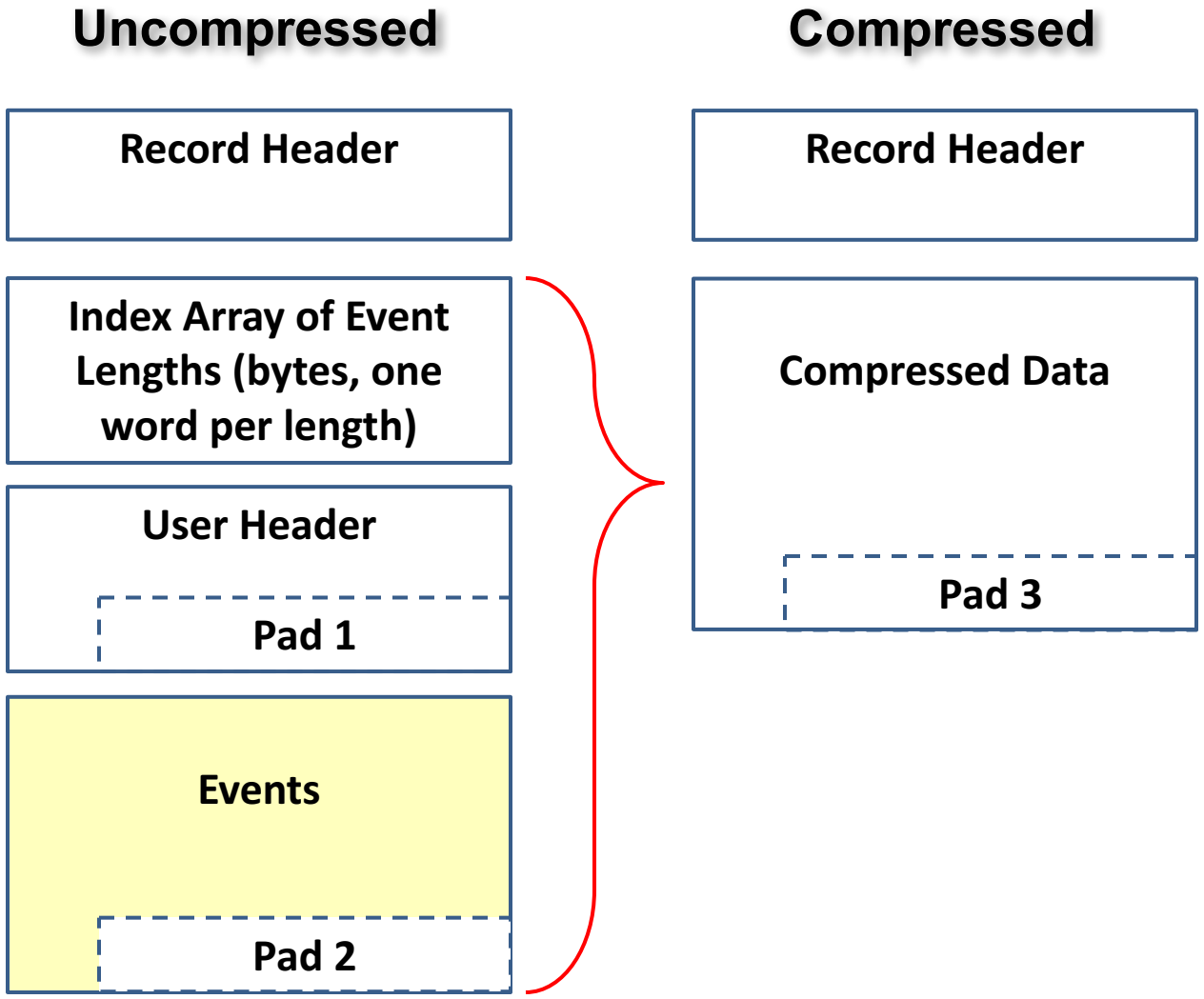
1	Block Length	Length of block in 32-bit words, inclusive
2	Block Number	Record id starting at 0
3	Header Length	Length of block header in 32-bit words (8)
4	Start	Offset in words to first event header in block relative to start of block
5	End	Number of valid words in block (header + data). Same as block length except for the last block.
6	Version	Evio format version
7	Reserved	Reserved
8	Magic Number	Number for endianness tracking (0xc0da0100)

Block Header (evio format version 4)

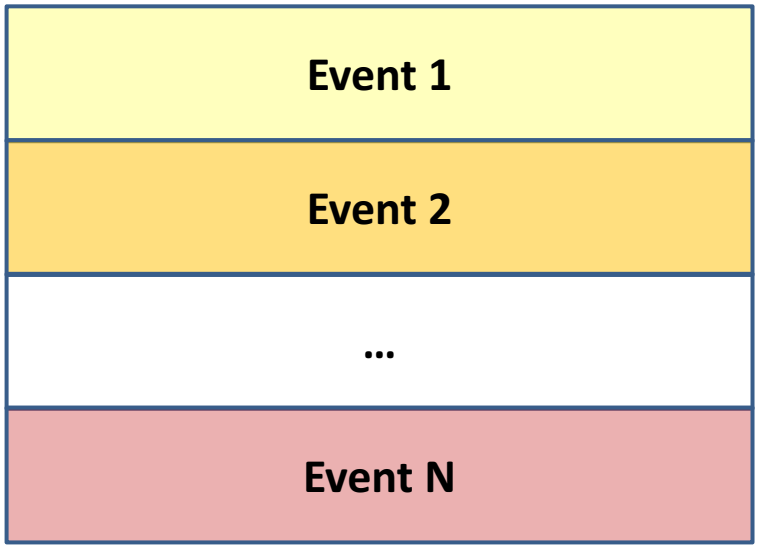
1	Block Length		Length of block in 32-bit words, inclusive
2	Block Number		Order of block in network transfer (record id) starting at 1. From ROC: -1 if payload banks not being built.
3	Header Length		Length of block header in 32-bit words (8)
4	Event Count		Number of evio events (payload banks) in block, not including dictionary.
5	Reserved 1		If content is being built (eg ROC Raw type), = source CODA id, else reserved
6	Bit Info	Version	Version: lowest 8 bits (0-7). Bit Info: Bit 8 = has dictionary, Bit 9 = is last block, Bits 10-13 = payload bank type (ROC Raw = 0, Physics = 1, PartialPhysics = 2, Disentangled = 3, User = 4, Control = 5, Other = 15). Bit 14 = has "first event" (in every split file) is first USER type event in this block NOTE: User events from ROC are typed as ROC Raw (EB handles this).
7	Reserved 2		Reserved
8	Magic Number		Number for endianness tracking (0xc0da0100)

HIPO/EVIO FORMAT VERSION 6

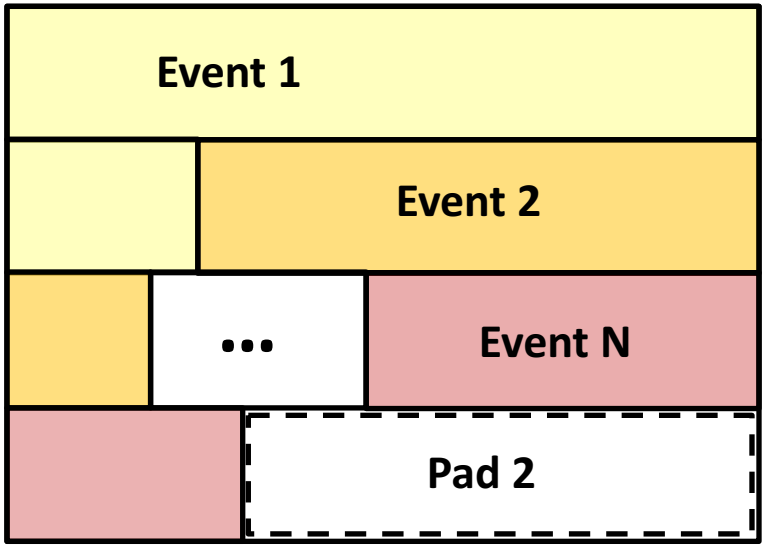
Record


Uncompressed
Compressed
Record Header
Index Array of Event Lengths (bytes, one word per length)
User Header
Pad 1
Events
Pad 2
Record Header
Compressed Data
Pad 3

Evio Events



HIPO Events



File Trailer

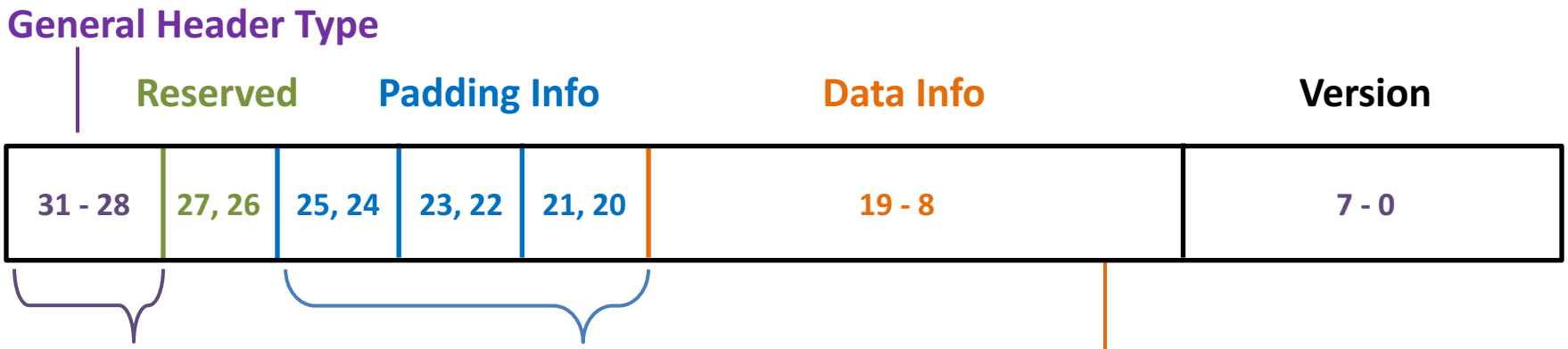
Record Header

Optional Uncompressed Array:
record length in bytes,
followed by its event count
(2 words / record)
(all records except this trailer)

Record Header

1	Record Length		Length of record in 32-bit words, inclusive
2	Record Number		Record id from 1. From ROC: -1 if payload banks not being built
3	Header Length		Length of this header in 32-bit words (always 14)
4	Event Index Count		Number of events contained (Evio: not including dictionary). Same as index array length in 32-bit words if array exists.
5	Index Array Length		Length of index array in bytes. Each array word is an event length in bytes.
6	Bit Info	Version	Evio format version in low 8 bits. Bit Info in high 24 bits
7	User Header Length		Optional user header length in bytes
8	Magic Number		Number for endianness tracking (0xc0da0100)
9	Uncompressed Data Length		Length of uncompressed record, without this header, in bytes
10	Type	Compressed Data Length	Compression type in high 4 bits (0=none, 1 = LZ4, 2 = LZ4 Best, 3 = GZIP). Length of compressed data in 32-bit words (low 28 bits).
11 12	User Register 1		User defined long word (64 bits)
13 14	User Register 2		User defined long word (64 bits)

File/Record Headers, Bit Info / Version Word



Value	Header Type
0	Evio record
1	Evio file
2	Evio extended file
3	Evio file trailer
4	HIPO record
5	HIPO file
6	HIPO extended file
7	HIPO file trailer

Bits	Padding
25-24	Pad 3
23-22	Pad 2
21-20	Pad 1

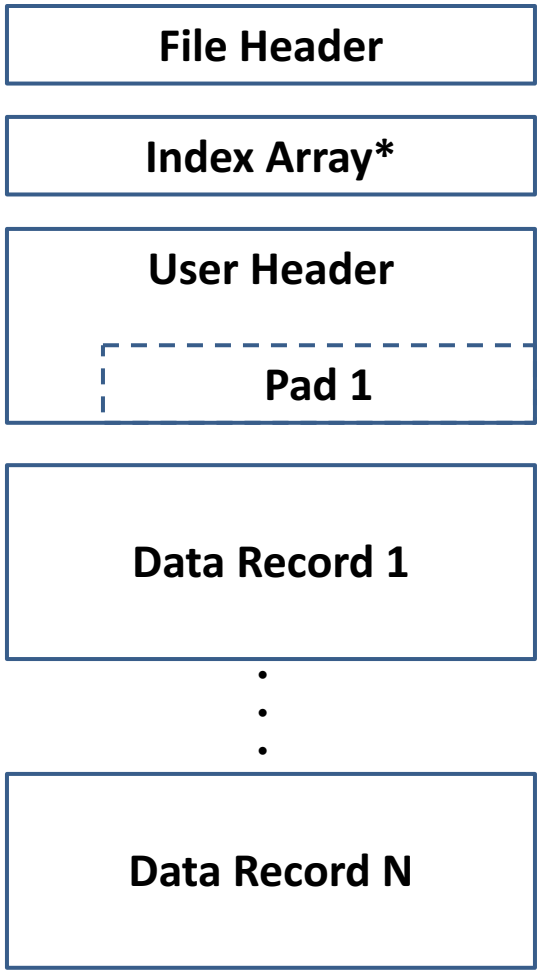
Bit uses depend on header type

Data Info Bits for Headers

BIT (from 0)	FILE HEADER (if bit on)
8	Dictionary exists
9	Has “first event” (in every split file)
10	File trailer with index array exists
11-19	Reserved

BIT (from 0)	RECORD HEADER (if bit on)
8	Dictionary exists
9	Is last record in stream or file
10-13	Data content type for CODA online only: ROC Raw = 0, Physics = 1, Partial Physics = 2, Disentangled = 3, User = 4, Control = 5, Other = 15
14-19	Reserved

File



* Same format as file trailer index:
1 word of record length in bytes,
followed by 1word of event count

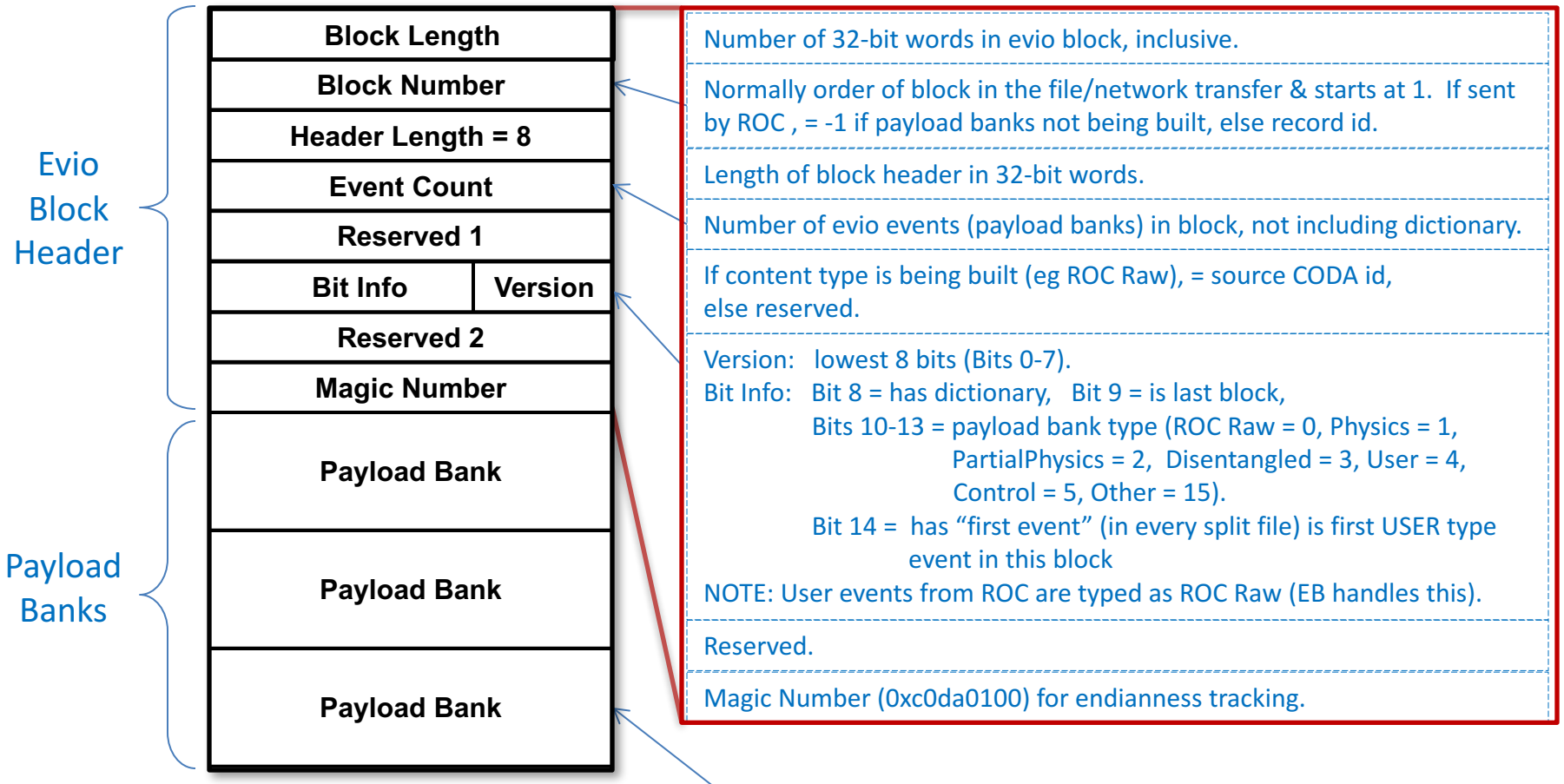
File Header

1	ID		Identification word. For Evio = 0x4556494F (EVIO in ascii). For HIPO = 0x43455248 (CERH in ascii).
2	File Number		If file being split, the split number (starting at 1)
3	Header Length		Length of this header in 32-bit words (always 14)
4	Record Count		Number of records contained. Same as index array length in 32-bit words if array exists.
5	Index Array Length		Length of index array in bytes
6	Bit Info	Version	Evio format version in low 8 bits. Bit Info in high 24 bits
7	User Header Length		Optional user header length in bytes
8	Magic Number		Number for endianness tracking (0xc0da0100)
9 10	User Register		64 bit register available for user
11 12	Trailer Position		Number of bytes from beginning of file to beginning of trailer (ending general record header). Value of 0 means either no trailer exists or its position is unavailable
13	User Integer 1		Integer available for user
14	User Integer 2		Integer available for user

EXTENDED File Header (Differences)

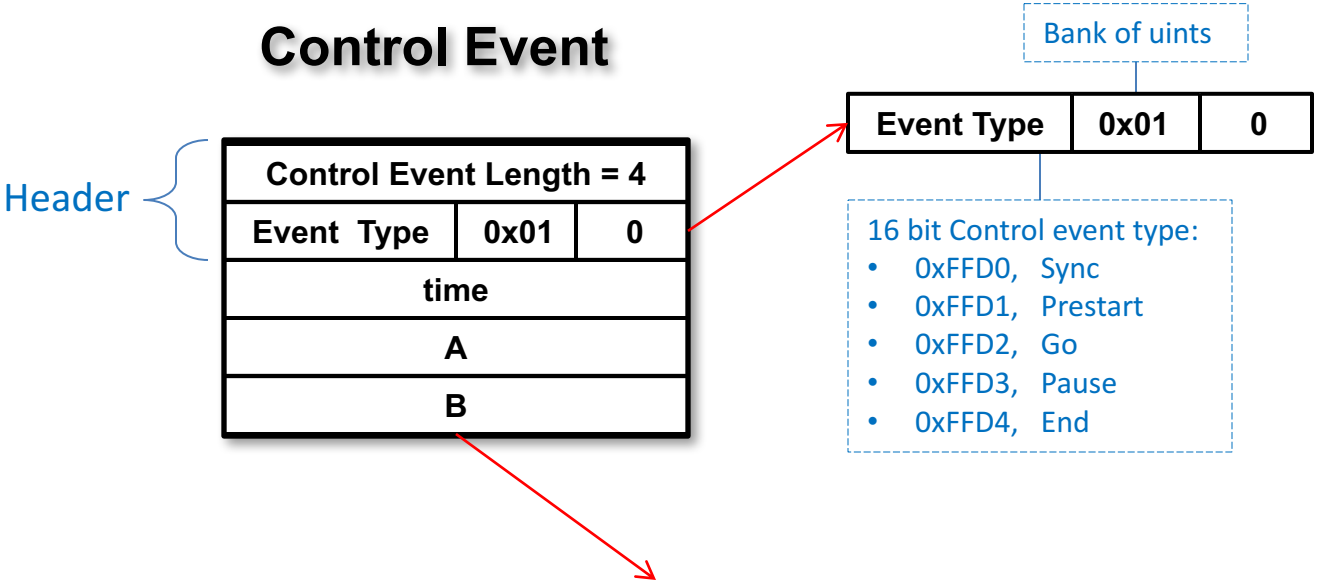
3	Header Length	Length of this header in 32-bit words GREATER THAN 14
15 +	User Integers 3+	Additional integers available for user beyond the regular general file header.

Network Transfer (Evio Output) Format



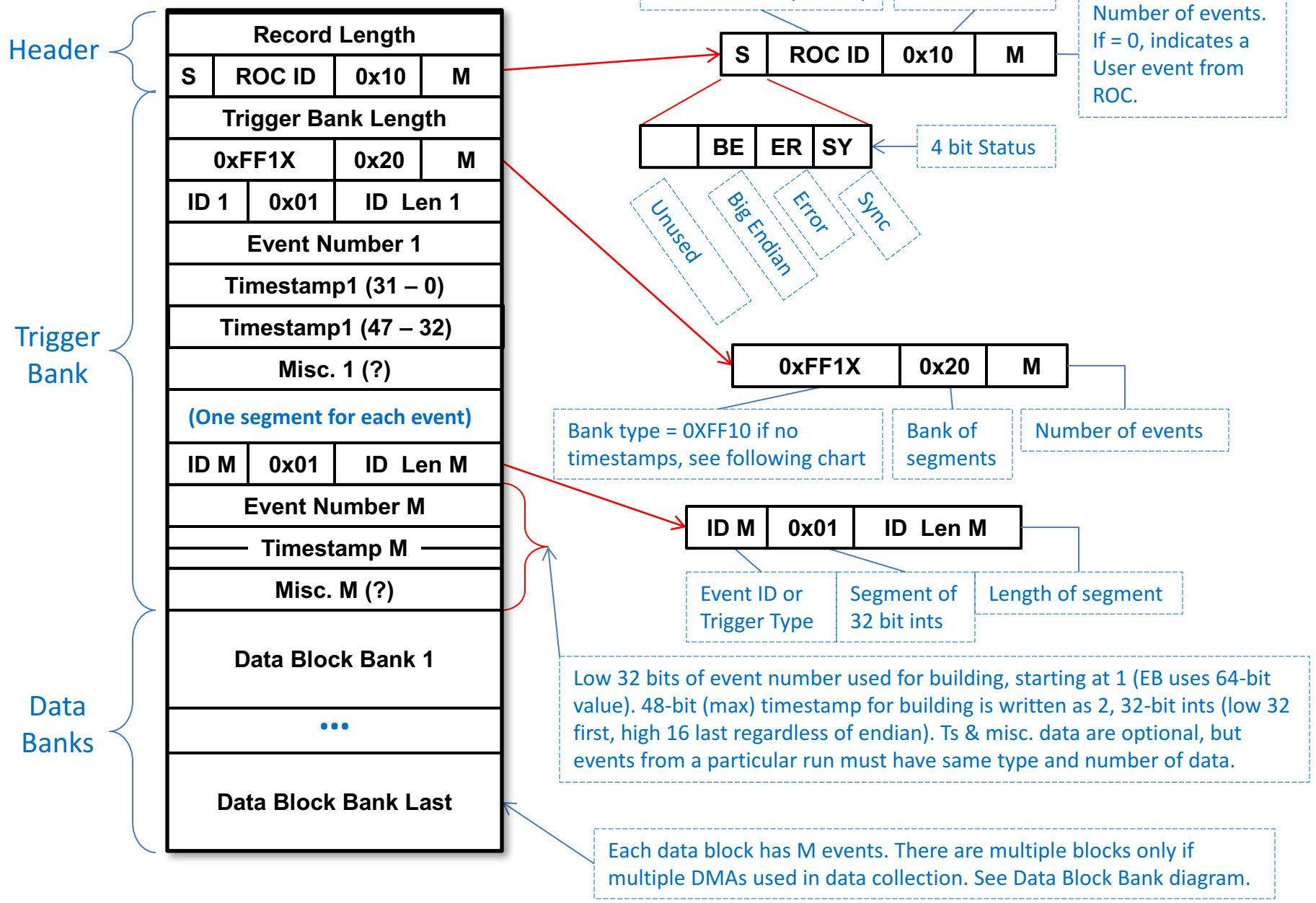
Format used when sending all types of online CODA data over the network. They are in standard evio buffer/file output format with block headers.

Control Event



Event Type	A	B
Sync	# events since last sync	# events in run
Prestart	run number	run type
Go	(reserved)	# events in run so far
Pause	(reserved)	# events in run so far
End	(reserved)	# events in run so far

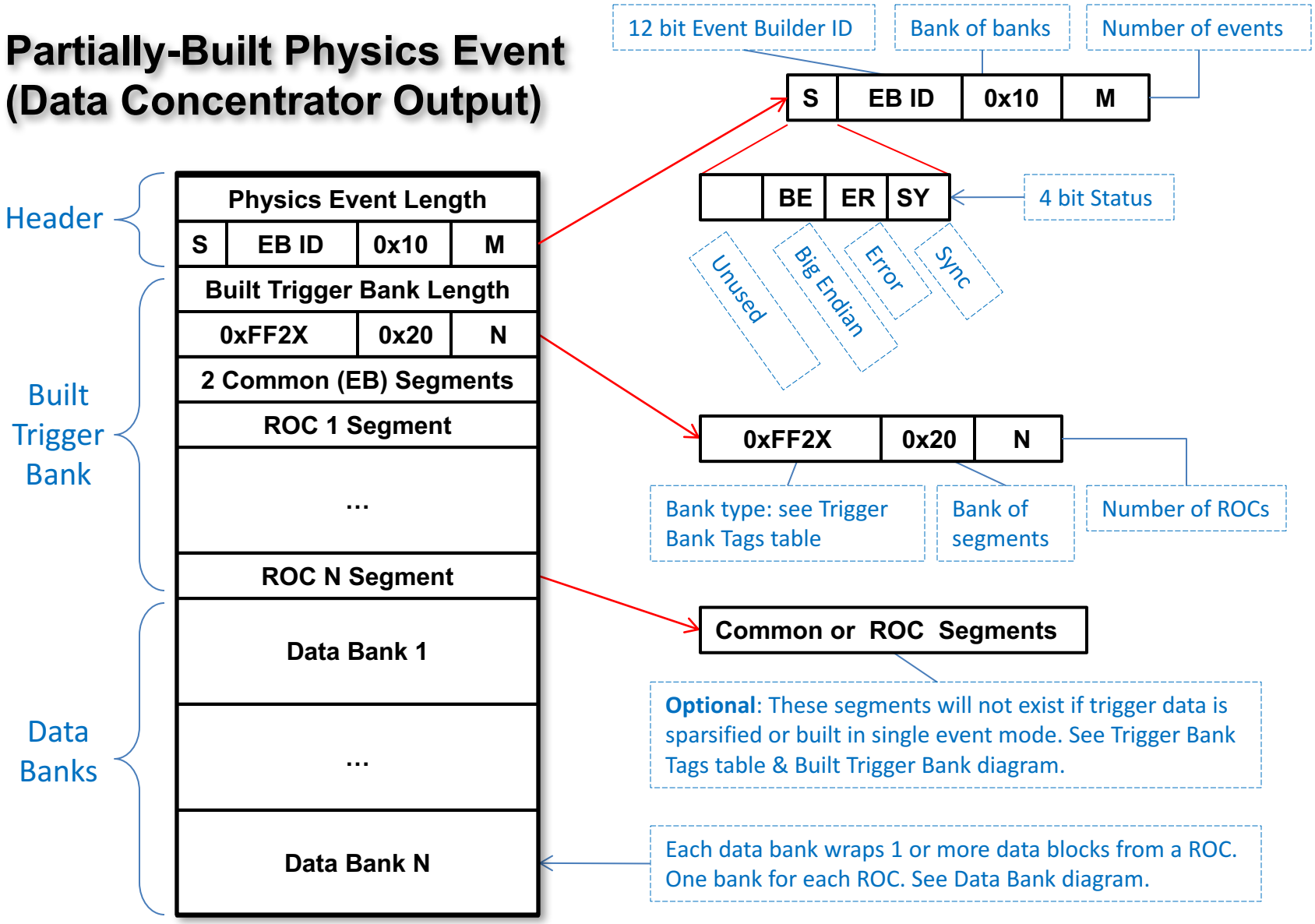
ROC Raw Data Record



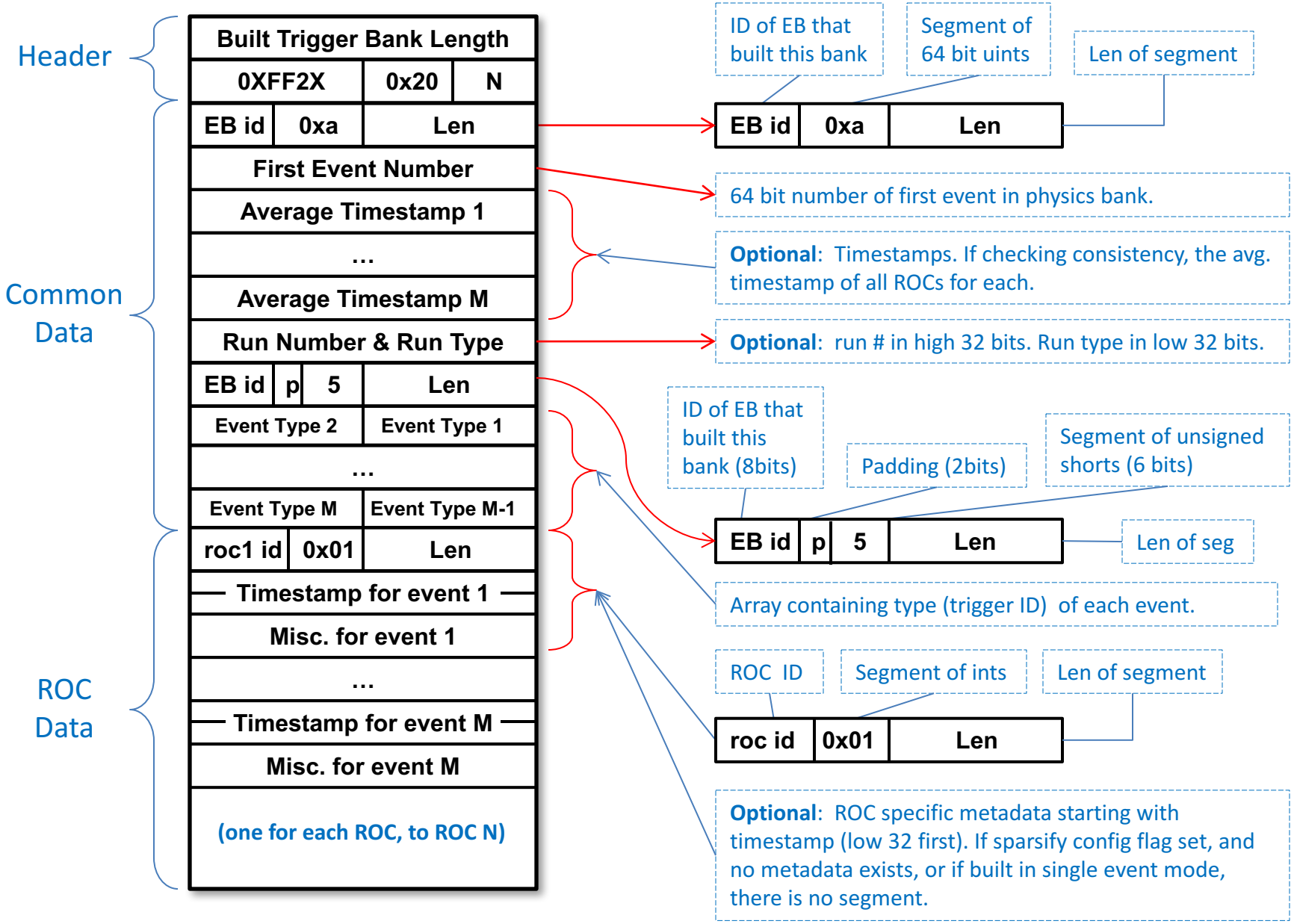
TRIGGER BANK TAGS

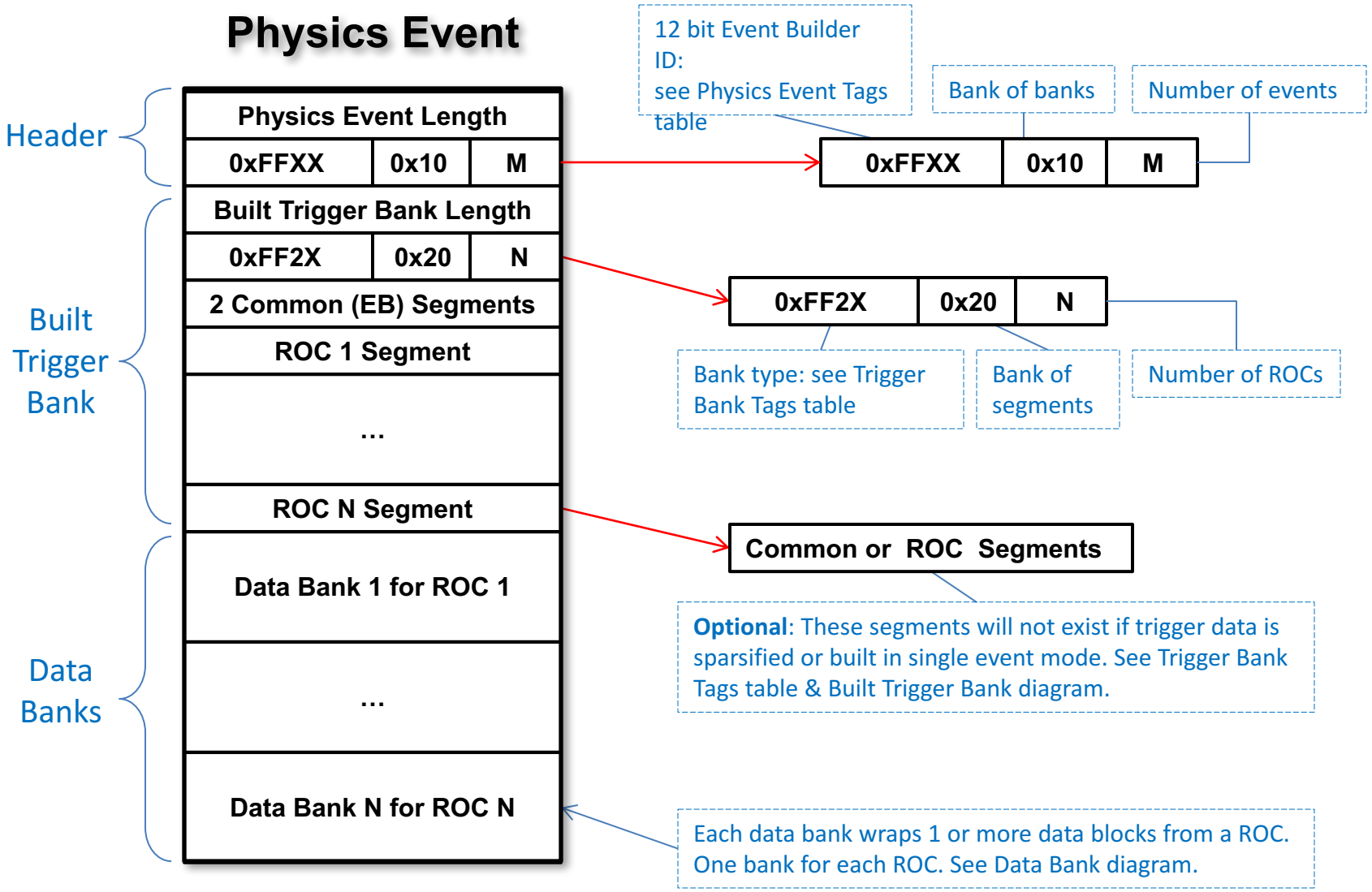
Tag Value	Purpose
0xFF10	Raw trigger, no timestamps
0xFF11	Raw trigger, w/ timestamps
0xFF20	Built trigger, **no** timestamps, **no** run # & run type, includes run specific data
0xFF21	Built trigger, w/ timestamps, but **no** run # & run type, includes run specific data
0xFF22	Built trigger w/ run # & run type, but **no** timestamps, includes run specific data
0xFF23	Built trigger with timestamps and run # & run type, includes run specific data
0xFF24	Built trigger, **no** timestamps, **no** run # & run type, **no** run specific data
0xFF25	Built trigger, w/ timestamps, but **no** run # & run type, **no** run specific data
0xFF26	Built trigger w/ run # & run type, but **no** timestamps, **no** run specific data
0xFF27	Built trigger with timestamps and run # & run type, **no** run specific data

Partially-Built Physics Event (Data Concentrator Output)



Physics Event's Built Trigger Bank




Physics Event
Header
Built Trigger Bank
Data Banks
Physics Event Length
0xFFXX
0x10
M
Built Trigger Bank Length
0xFF2X
0x20
N
2 Common (EB) Segments
ROC 1 Segment
...
ROC N Segment
Data Bank 1 for ROC 1
...
Data Bank N for ROC N
12 bit Event Builder ID: see Physics Event Tags table
Bank of banks
Number of events
0xFFXX
0x10
M
0xFF2X
0x20
N
Bank type: see Trigger Bank Tags table
Bank of segments
Number of ROCs
Common or ROC Segments
Optional: These segments will not exist if trigger data is sparsified or built in single event mode. See Trigger Bank Tags table & Built Trigger Bank diagram.
Each data bank wraps 1 or more data blocks from a ROC. One bank for each ROC. See Data Bank diagram.

CODA RESERVED BANK TAGS

Tag Value Range	Purpose
0xFF00 - 0xFFFF	Complete range of reserved values
0xFFE0 - 0xFFFF	Undetermined
0xFFD0 - 0xFFDF	Control events
0xFF90 - 0xFFCF	Undetermined
0xFF50 - 0xFF8F	Physics events
0xFF10 - 0xFF4F	Trigger banks
0xFF00 - 0xFF0F	Undetermined

CONTROL EVENT TAGS

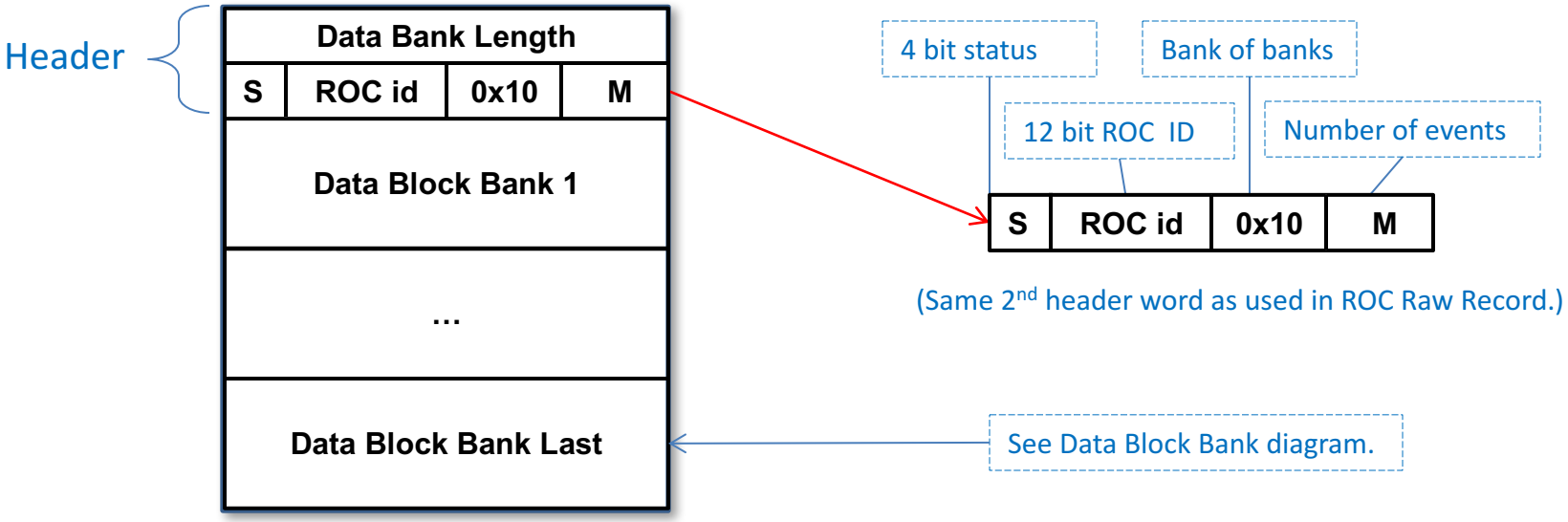
Tag Value	Control Event
0xFFD0	Sync
0xFFD1	Prestart
0xFFD2	Go
0xFFD3	Pause
0xFFD4	End

PHYSICS EVENT TAGS

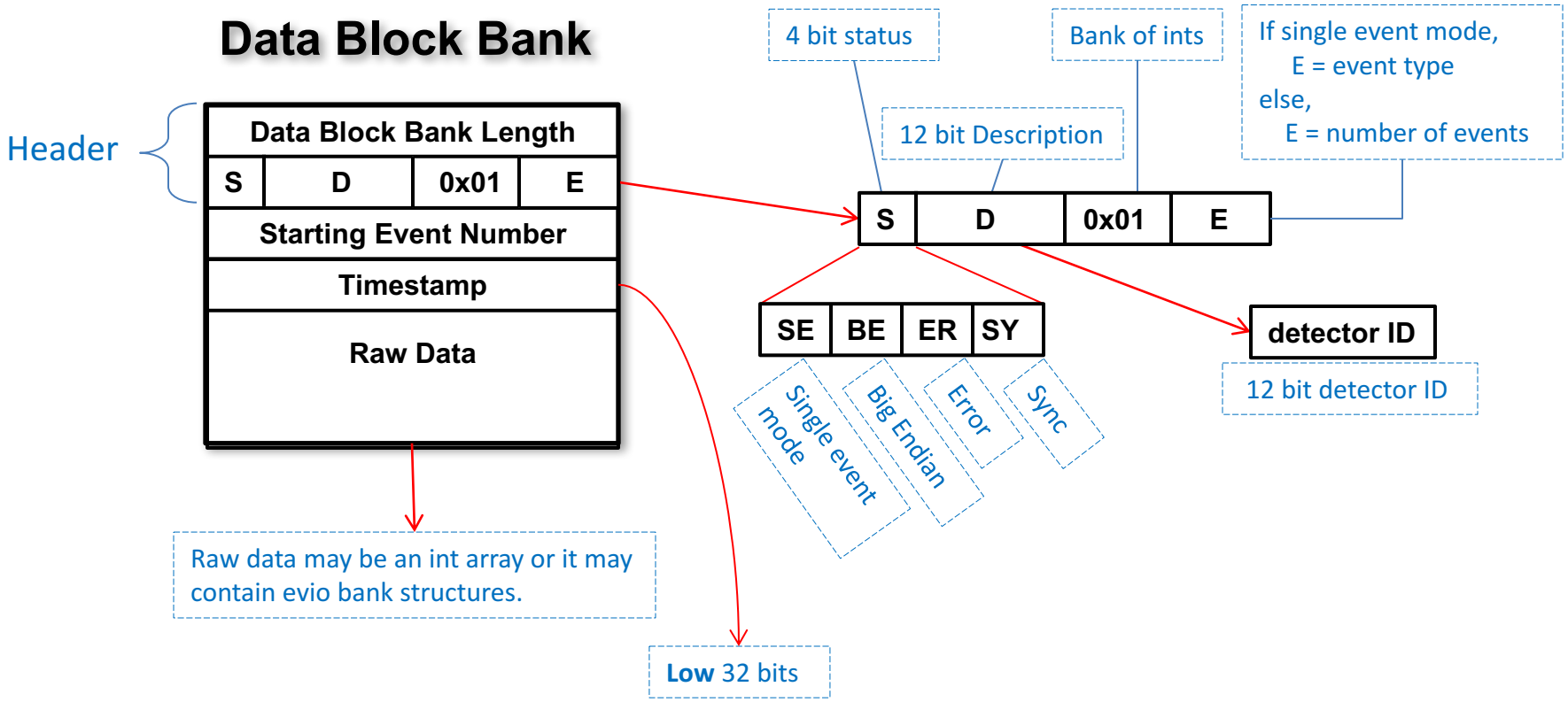
Tag Value	Event Made by
0xFF50	PEB
0xFF58	PEB with sync set
0xFF70	SEB
0xFF78	SEB with sync set

4th bit set indicates that the last event in the entangled block is a sync event

Physics Event's Data Bank

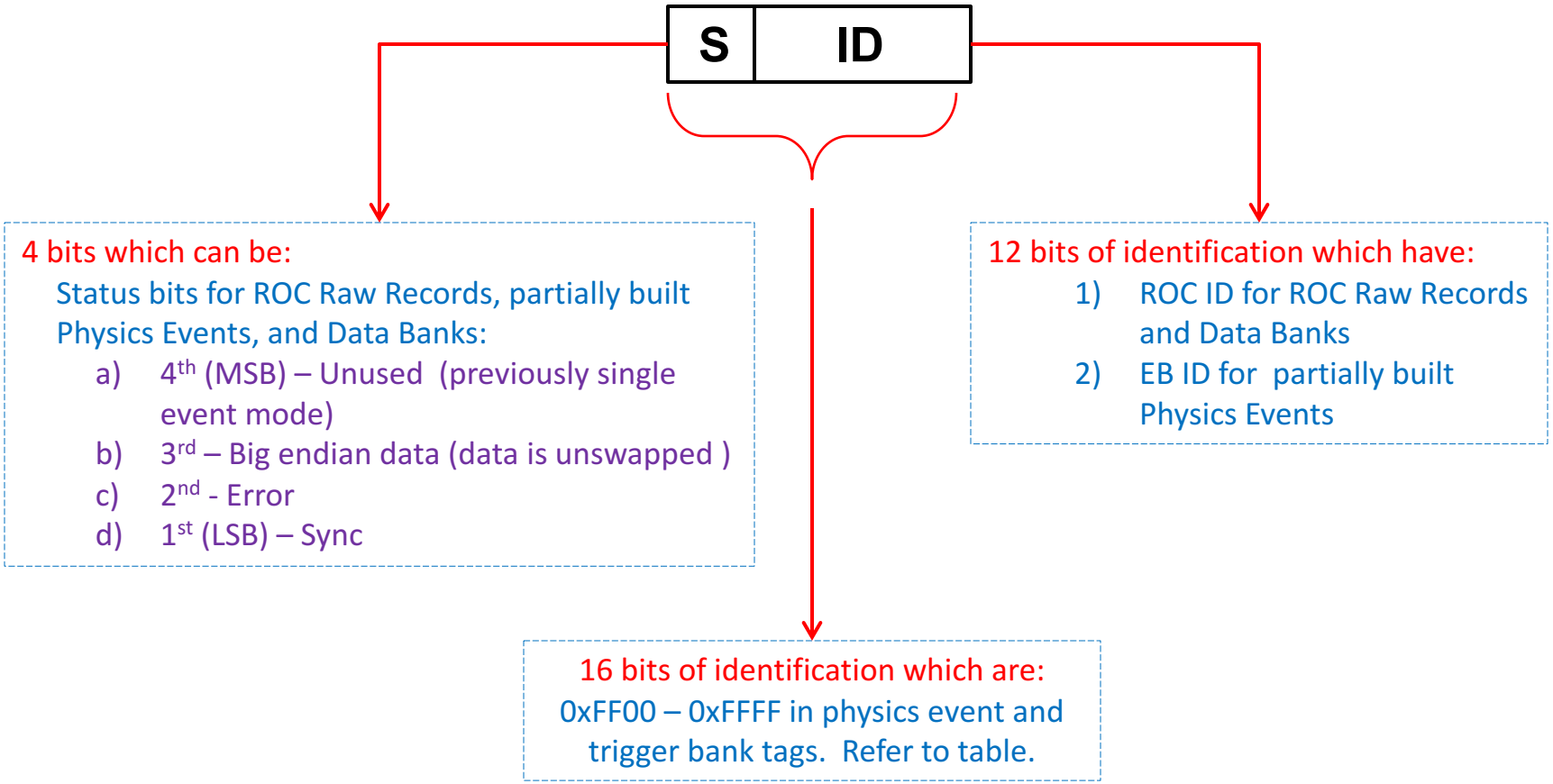


Data blocks from a single ROC are wrapped in this data bank. There should be at least one data block and there may be more if more than one DMA is used in acquiring data for this ROC. If more than one block, each contains a fragment for every one of the M events and from unique modules. In addition, the last block may have data associated only with the last event (such as scalar data).

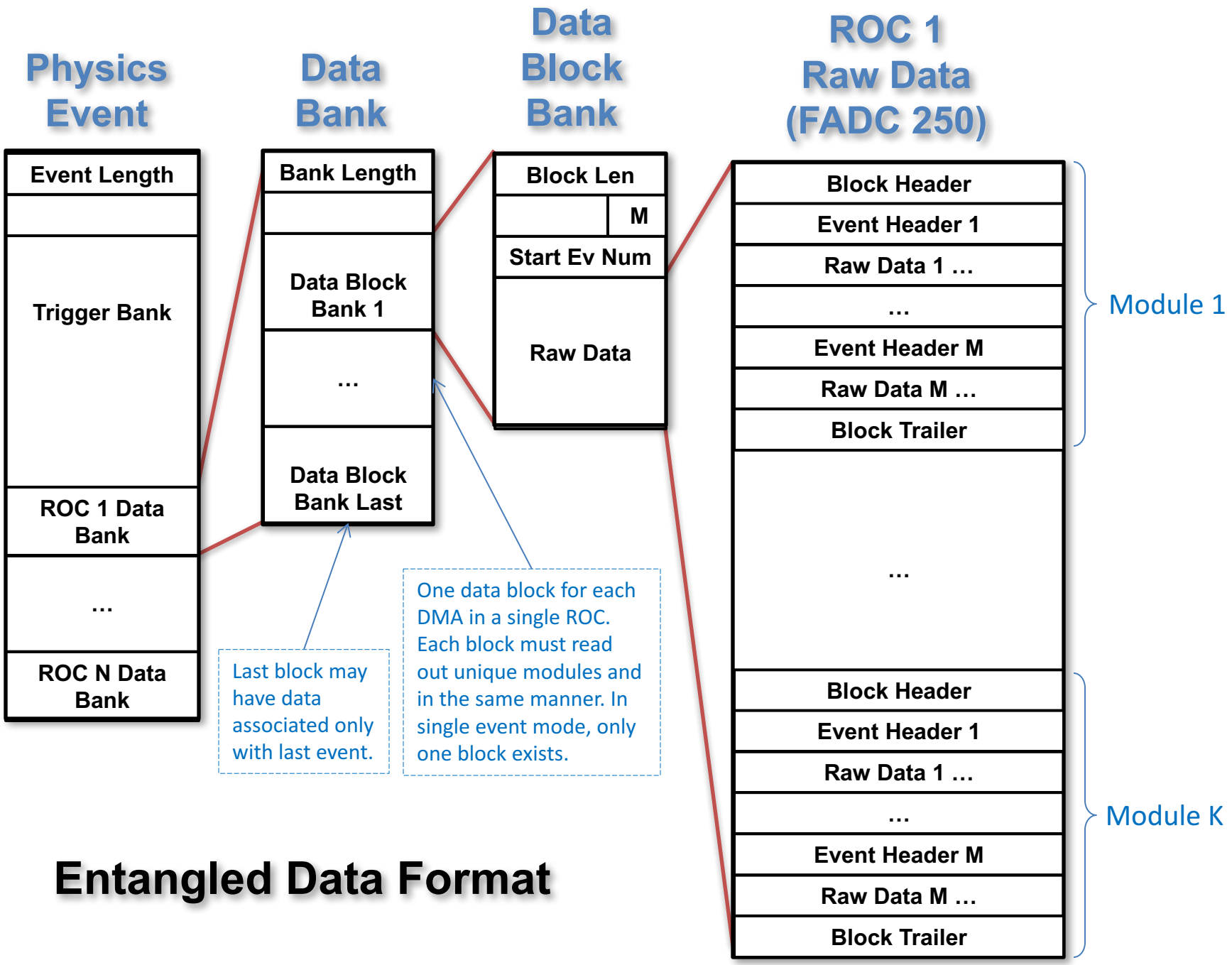


Contains raw data from a single ROC containing one or more events. If this block is the last in a data bank, and there are multiple events, and E = 1, then this data is associated only with the last event (e.g. scalar readout).

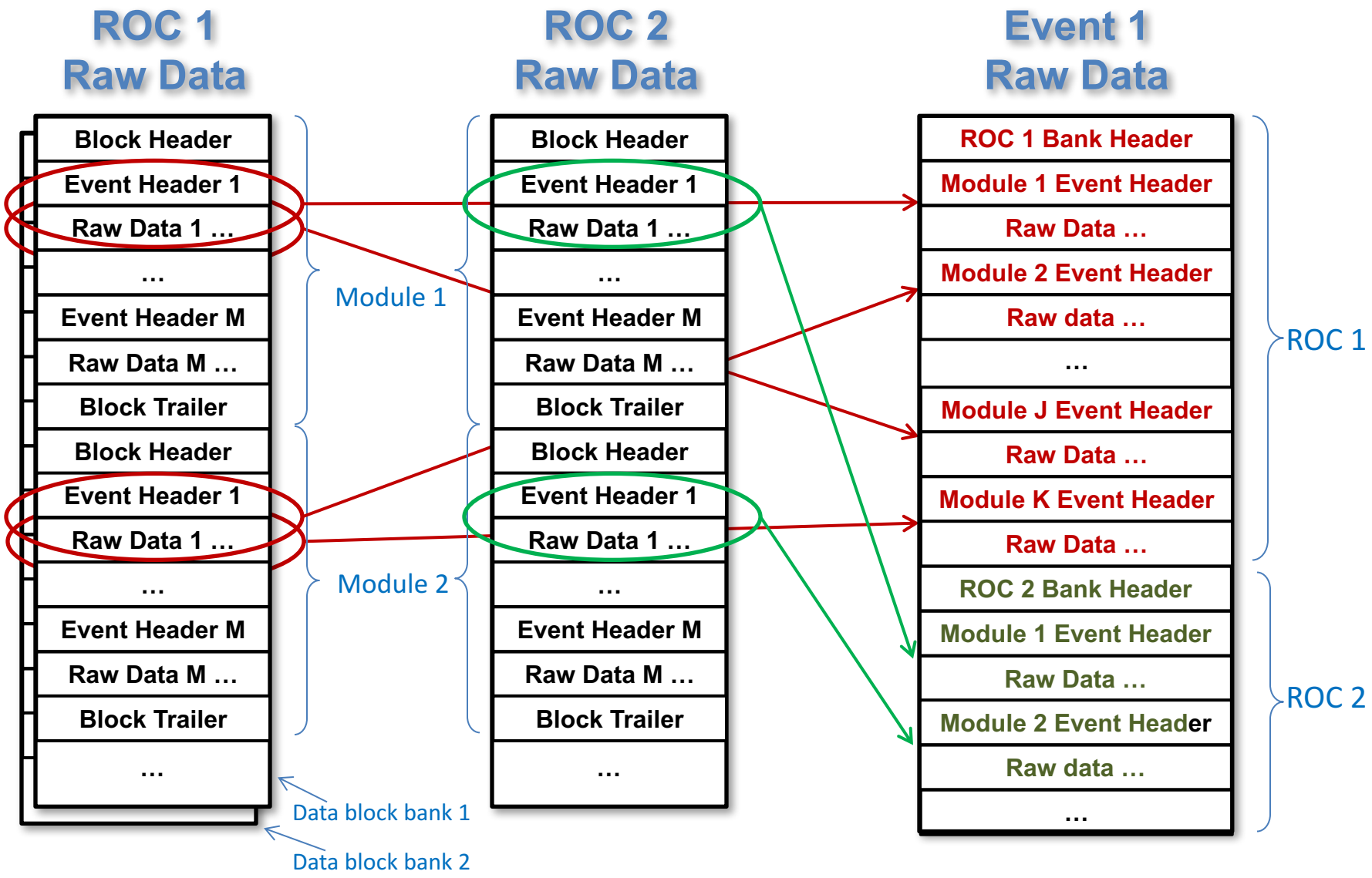
16-bit EVIO CODA-Format Tag



Disentangling Built Physics Event

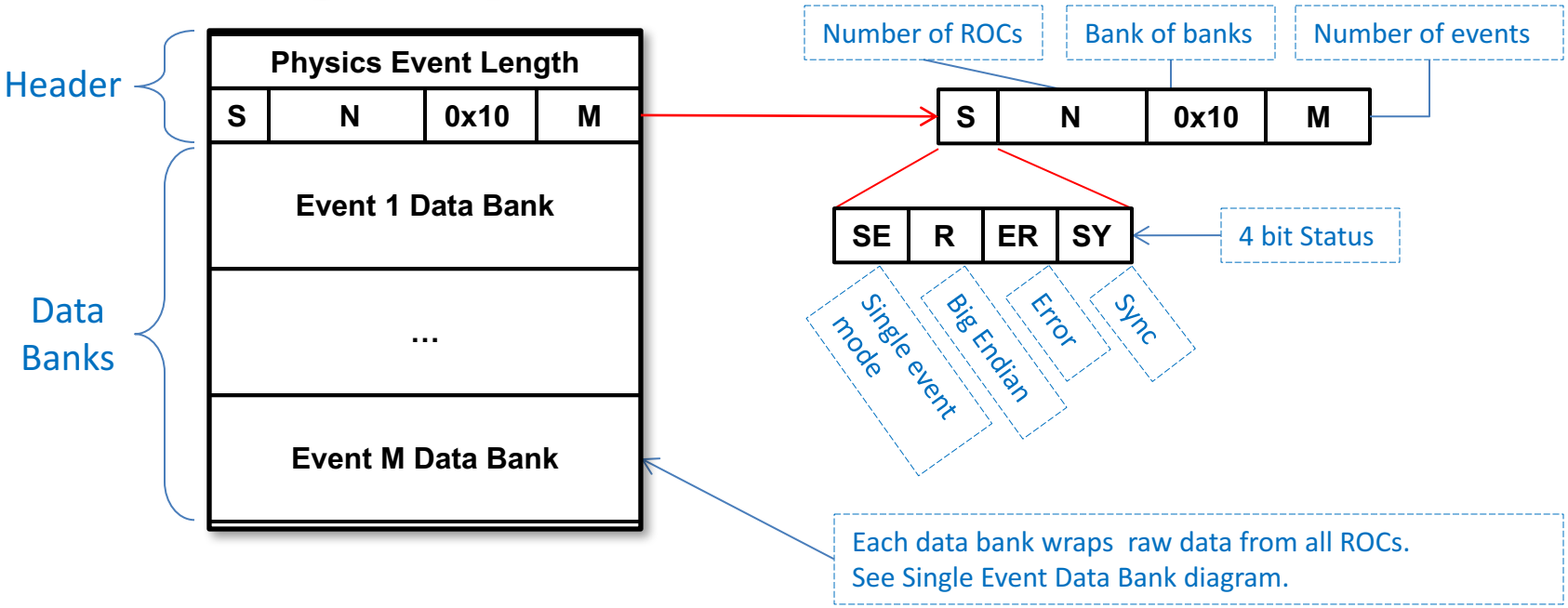


Entangled Data Format

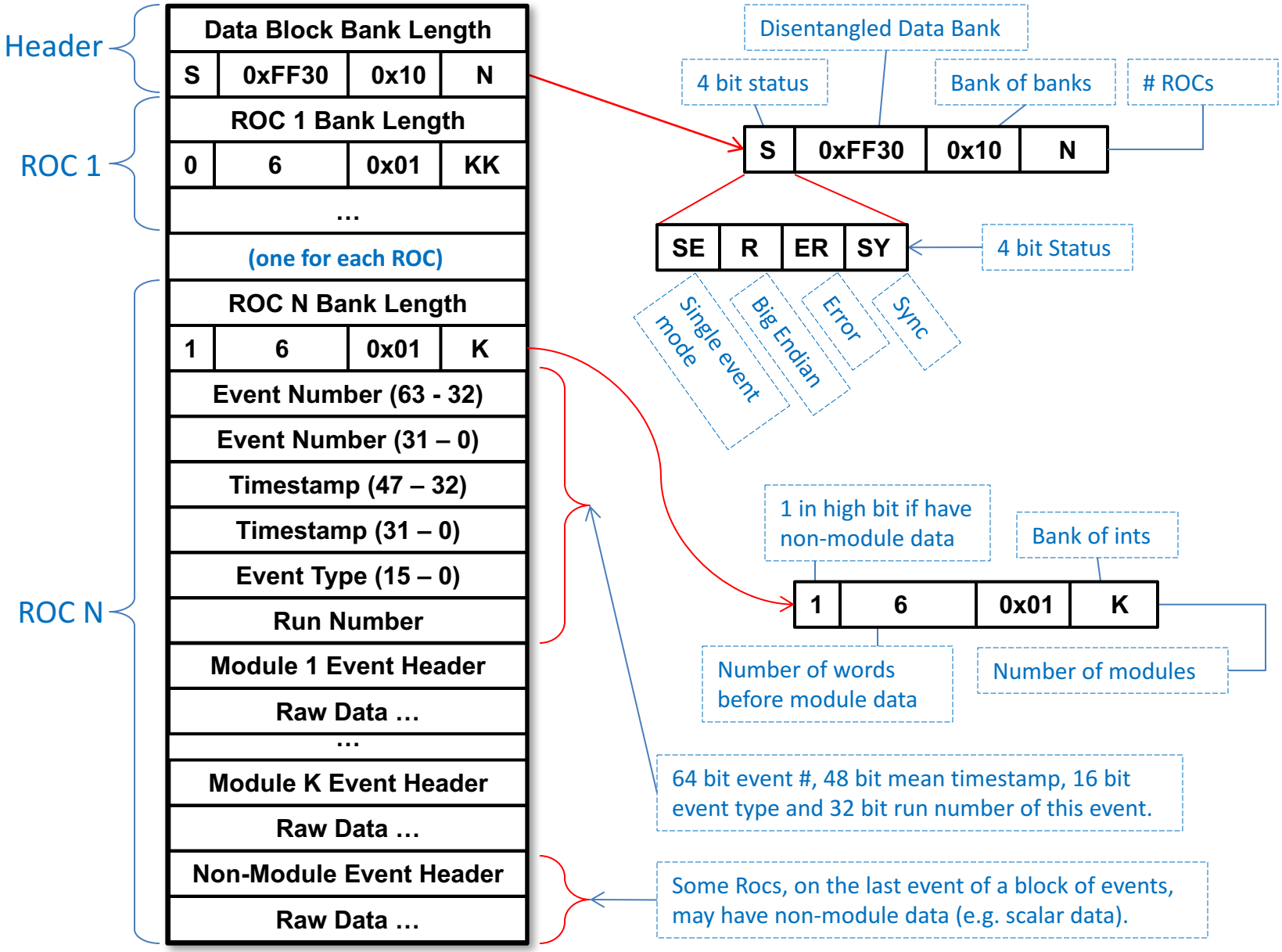


Entangled To Disentangled FADC 250 Raw Data

Disentangled Physics Event



Single Event (Disentangled) Data Bank



FADC 250

Data Type Values	
0 – block header	7 – pulse integral
1 – block trailer	8 – pulse time
2 – event header	9 – streaming raw data
3 – trigger time	10 – 12 user defined
4 – window raw data	13 – event trailer (debug only)
5 – window sum	14 – data not valid (empty module)
6 – pulse raw data	15 – filler (non-data) word

Block Header Word Format

Bits	Value	Comment
31	1	This is a type defining word
30 – 27	0	Data type = block header
26 – 22	Slot ID	Set by VME64 backplane
21 – 14	Event #	Number of events in block
13 – 12	Module Type	0=FADC250, etc.
11 – 0	Event block #	Used to align block when building events

General Data Word Format

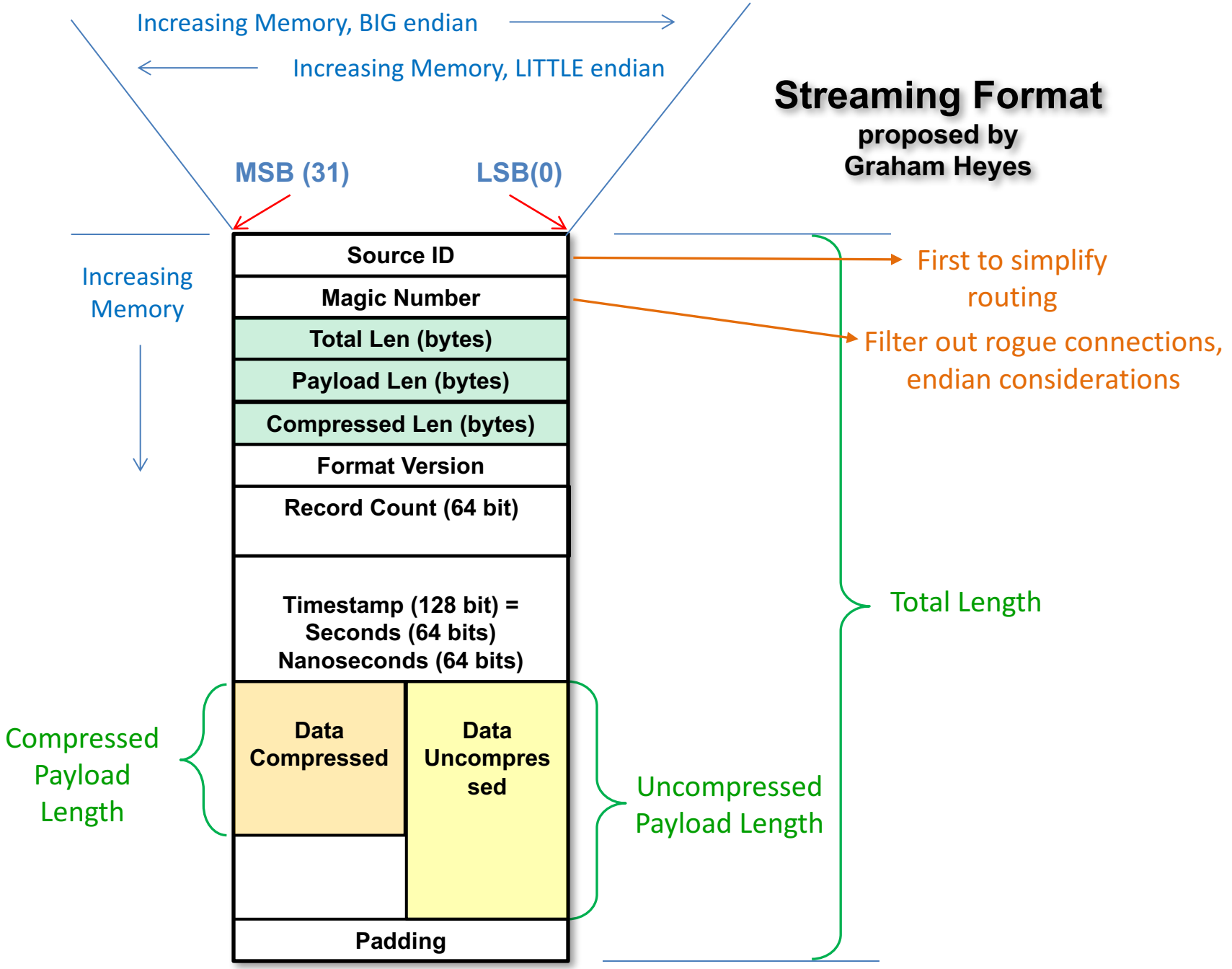
31st bit	Bits	Usage
1	30 - 27	4-bit data type (see chart)
1	26 - 0	Data type dependent data payload
0	3 0 - 0	Data payload using last defined data type

Block Trailer Word Format

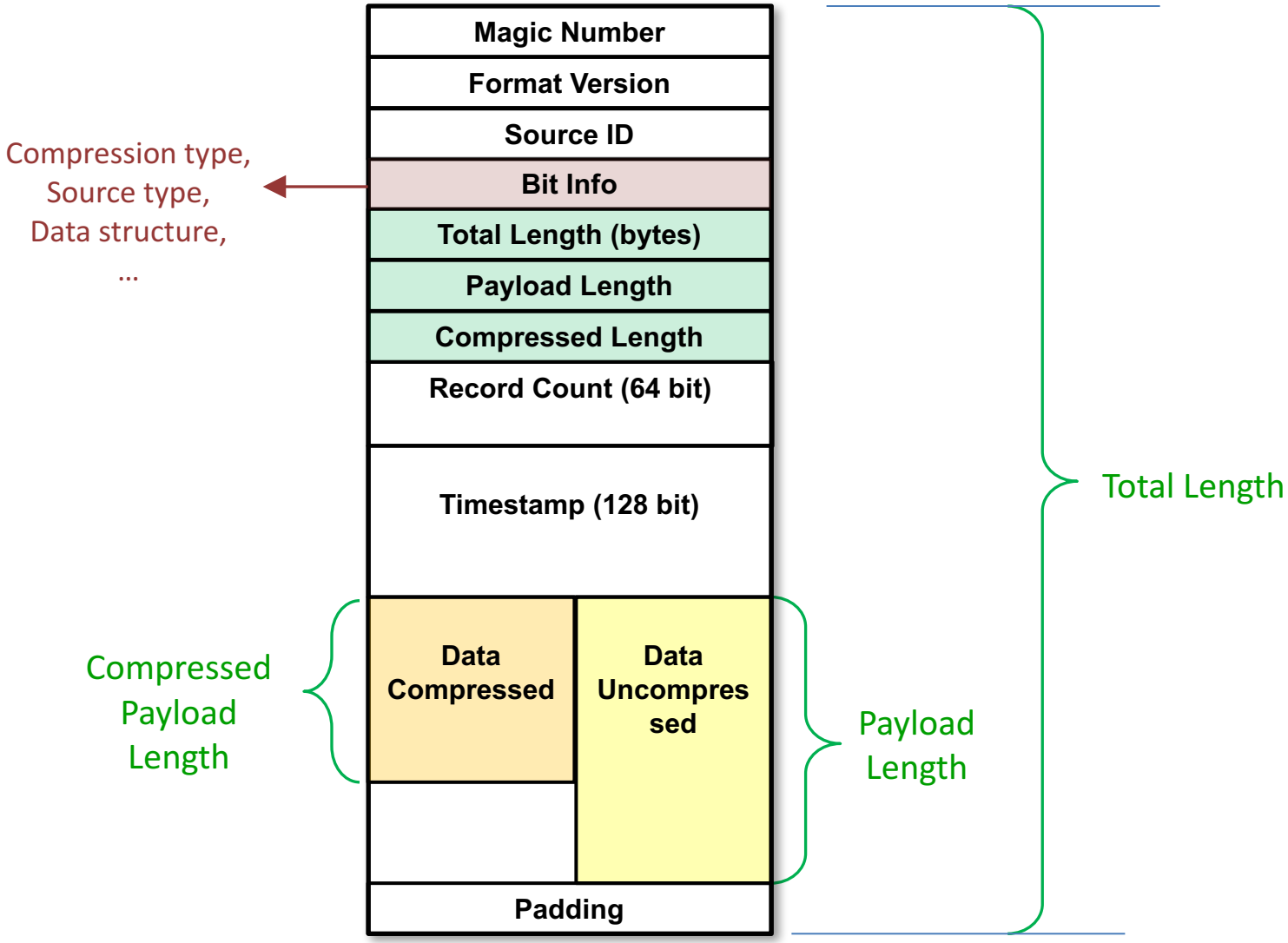
Bits	Value	Comment
31	1	This is a type defining word
30 – 27	1	Data type = block trailer
26 – 22	Slot ID	Set by VME64 backplane
21 – 0	Total # of words in block of events	Number of 32 bit words in block

Event Header Word Format

Bits	Value	Comment
31	1	This is a type defining word
30 – 27	2	Data type = event header
26 – 22	Slot ID	Set by VME64 backplane
21 – 20	Module type	0=FADC250, etc.
19 – 0	Trigger number	ADC processing chip #


Increasing Memory, BIG endian
Increasing Memory, LITTLE endian
Streaming Format
proposed by
Graham Heyes
MSB (31)
LSB(0)
Increasing
Memory
Source ID
Magic Number
Total Len (bytes)
Payload Len (bytes)
Compressed Len (bytes)
Format Version
Record Count (64 bit)
Timestamp (128 bit) =
Seconds (64 bits)
Nanoseconds (64 bits)
Data
Compressed
Data
Uncompres
sed
Padding
First to simplify
routing
Filter out rogue connections,
endian considerations
Total Length
Compressed
Payload
Length
Uncompressed
Payload Length

Streaming Format



Questions:

- Do we pick a fixed endian for simplicity? (and skip the magic #)
- Pick an endian just for the header?
- What if data / record have mixed endian values?
- Could we always ensure all data is 1 particular endian?

- Merge fields like format version and compression type that may not require 32 bits each?

- Record count to ensure sequential records made obsolete by timestamp?
- Send time window size? so we know if data is missing.
- Is time slice window fixed?

- Don’t allow fields that require the writer to go back and change it after writing data?