

RIBFDAQ データフォーマット  
2008年9月2日  
Version 1.1

馬場 秀忠

# コンセプト

- ▶ 以前のもものと完全互換はしない
- ▶ ある程度抽象化したルールを作っておく
  - ▶ 柔軟性をもたせる
  - ▶ 冗長性をもたせない
- ▶ Event FragmentでもEvent Assemblyでも同じルール
  - ▶ 同じルーチンで解析できる
- ▶ シンプルに

# ルール



- ▶ Reserve : 予約Bit (2bit=4)
- ▶ Layer : Block中の階層の深さ (2bit=4)
- ▶ Class ID : こういった種類のブロックか
  - ▶ Class ID毎にフォーマットが違う
- ▶ Block Size : Blockのサイズ(Short word単位)
  - ▶ Short word 22 Bit = 4 M
  - ▶ 1 Blockの最大サイズは8 MB
- ▶ Address : 送信元のID (32bit)
  - ▶ IP Addressと同じにしてよい
  - ▶ 同時にEvent Fragment IDに対応

# 1 Block

R	Ly	Class ID	Size
Address			

-	0	0	Block Size
192.168.0.10			
-	1	1	Event Size
192.168.0.10			
Event Number			
-	1	1	Segment Size
192.168.0.10			
Segment ID			
Data			
⋮			

# Slow系も同様に

-	0	0	Block Size
192.168.0.10			
-	1	3	Event Size
192.168.0.10			
Event Number			
-	2	4	Segment Size
192.168.0.10			
Segment ID			
Data			
-	1	10	Status Size
192.168.0.10			
Status ID			
Status Data			
⋮			

# 階層構造

## ▶ Block header

### ▶ Event data

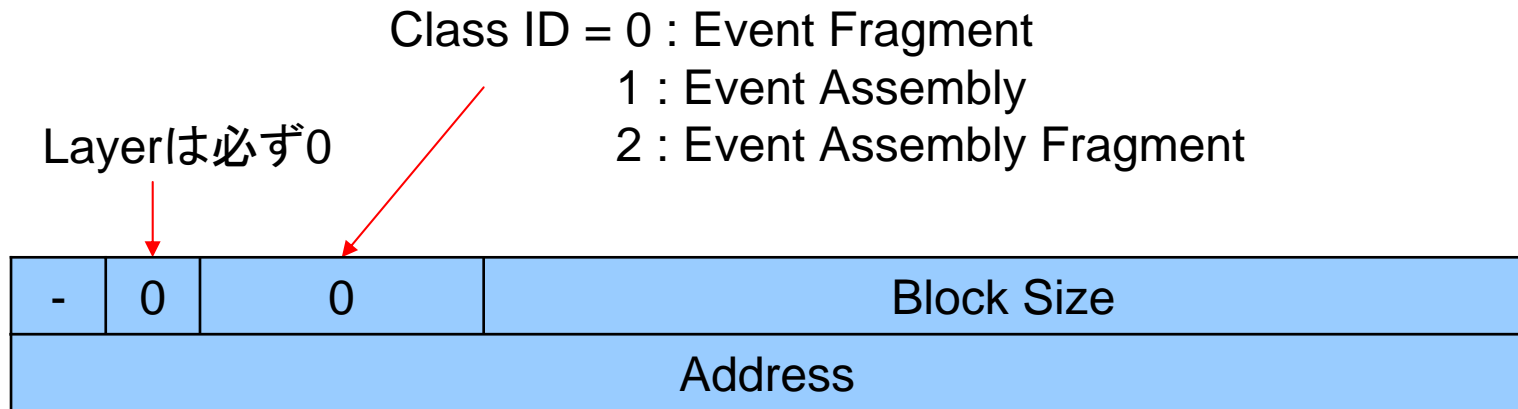
- ▶ Segment data
- ▶ Analyzed segment data
- ▶ Scaler data
- ▶ Status data

### ▶ Scaler data

### ▶ Status data

### ▶ Comment data

# Block data header



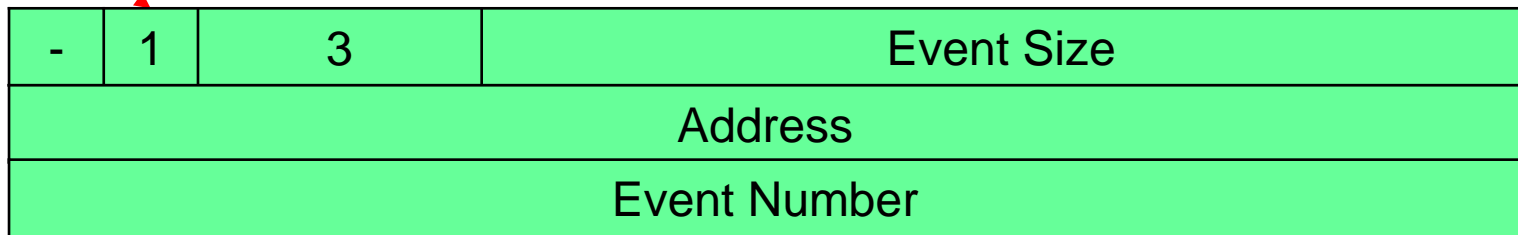
## ▶ 不定長BlockのTCP転送

- ▶ `recv(sock, header, 4, MSG_WAITALL);`
- ▶ `blksize = (header & 0x003ffff) * 2;`
- ▶ `recv(sock, buf, blksize, MSG_WAITALL);`

# Event data header

Layerは必ず1

Class ID = 3



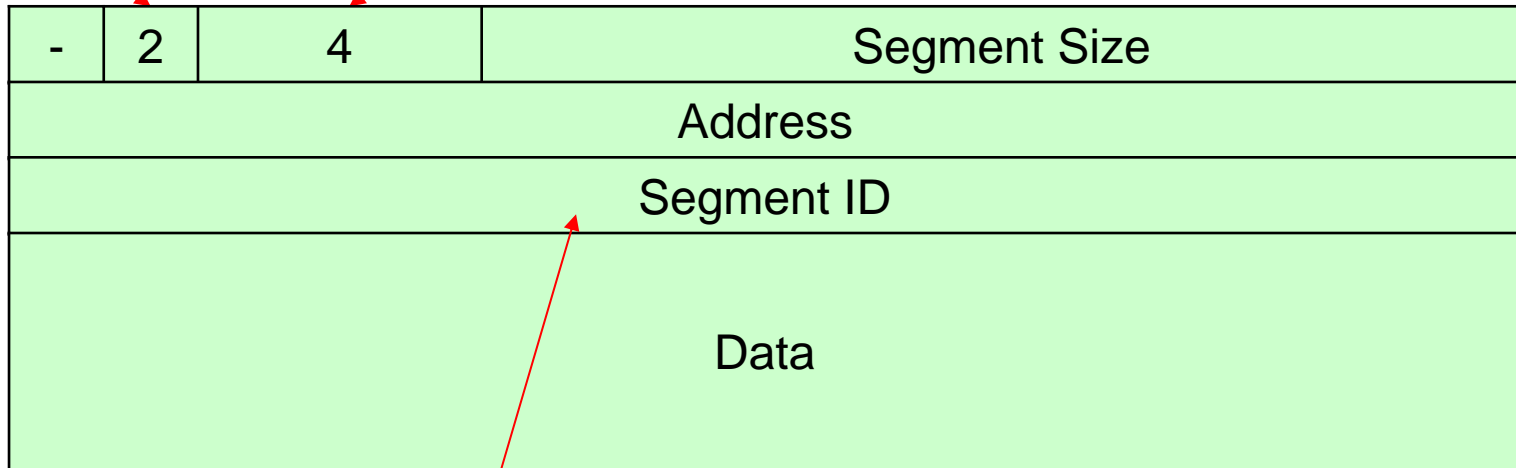
32Bit Event numberあれば  
10kcps triggerでも5日間もつ



# Segment data

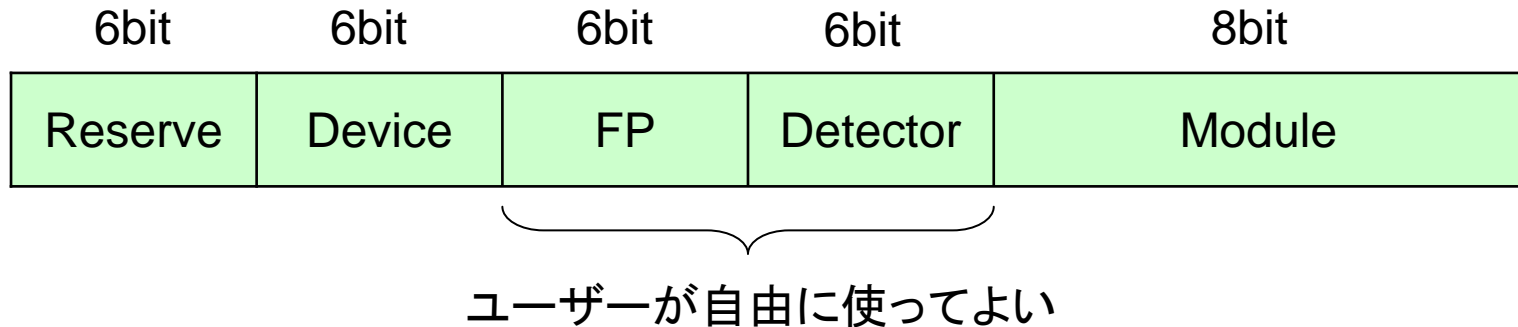
Layerは必ず2

Class ID = 4



Segment IDで何のデータか判断  
決まりきっているものは固定  
(次ページ参照)

# Segment IDのルール

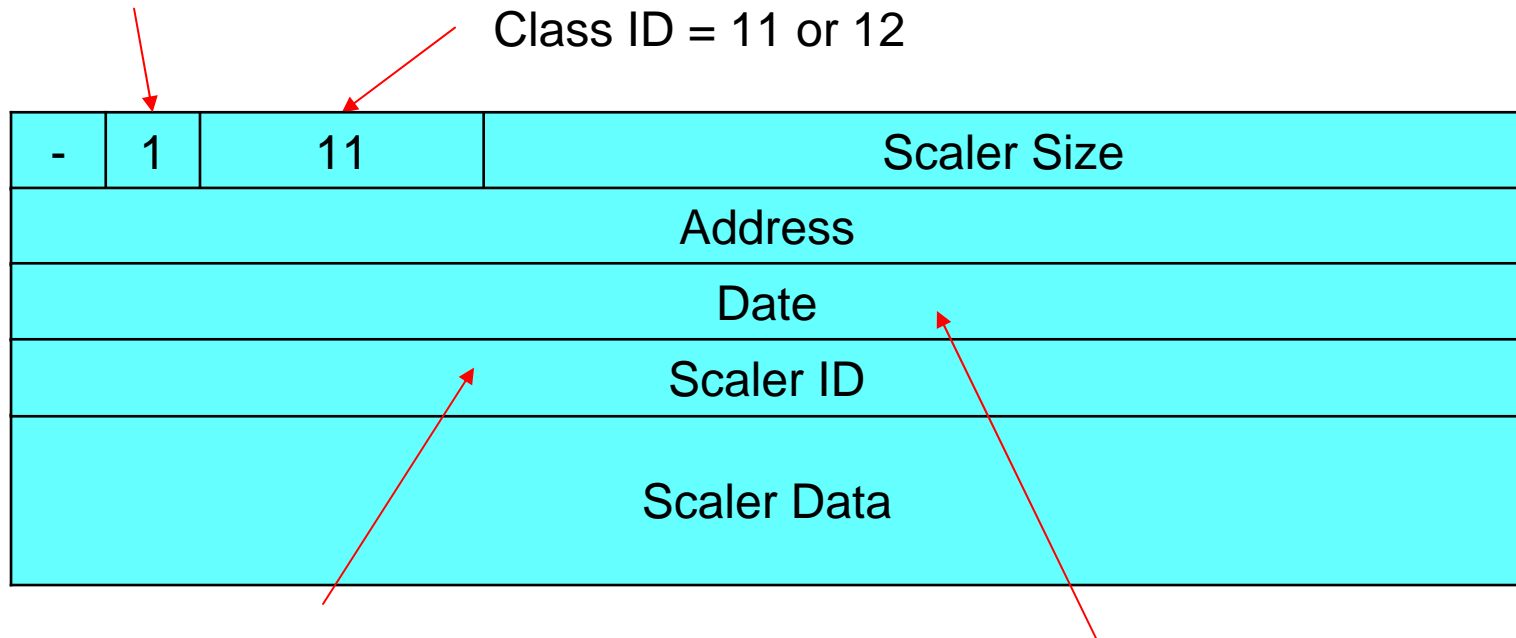


- ▶ Moduleはモジュール固有のID
  - ▶ 解析ルーチンではこのIDを使ってデコード
- ▶ Deviceは何のデバイスかを示している
  - ▶ 0=BigRIPS, 1=ZDS, 10=DALI, 11=GRAPE ...
- ▶ FP (Focal Plane)、Detectorの12bitはこれに従わず、ユーザーが自由に決めてよい
  - ▶ BigRIPS, ZDSの場合はFP 1~11=F1~F11、Detector 0=PPAC

# Scaler data

Layerは1か2  
Eventに同期してScalerをとる場合は2にして  
Event dataの下に入れる

Class ID 11 はラン中にクリアされない  
Class ID 12 はバッファ転送毎にクリアされる



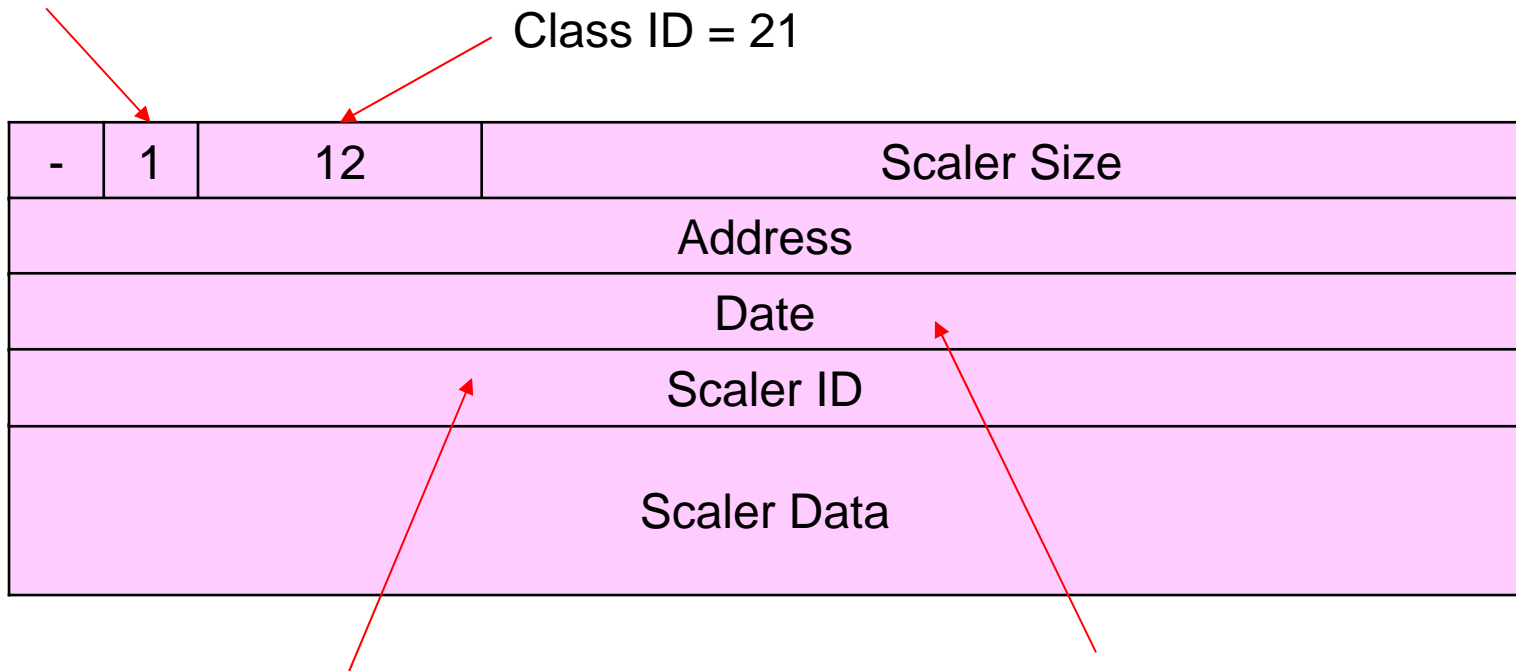
Scaler IDで何のデータか判断  
固定できるものは固定  
たとえば1 = BigRIPS Beam Line

Date=Event numberで時刻をタグ  
Layer 1の場合、Block転送時に  
Storeされるので、Event dataと  
同期はされない

# Status data

Layerは1か2

Eventに同期してStatusをとる場合は2にして  
Event dataの下に入れる



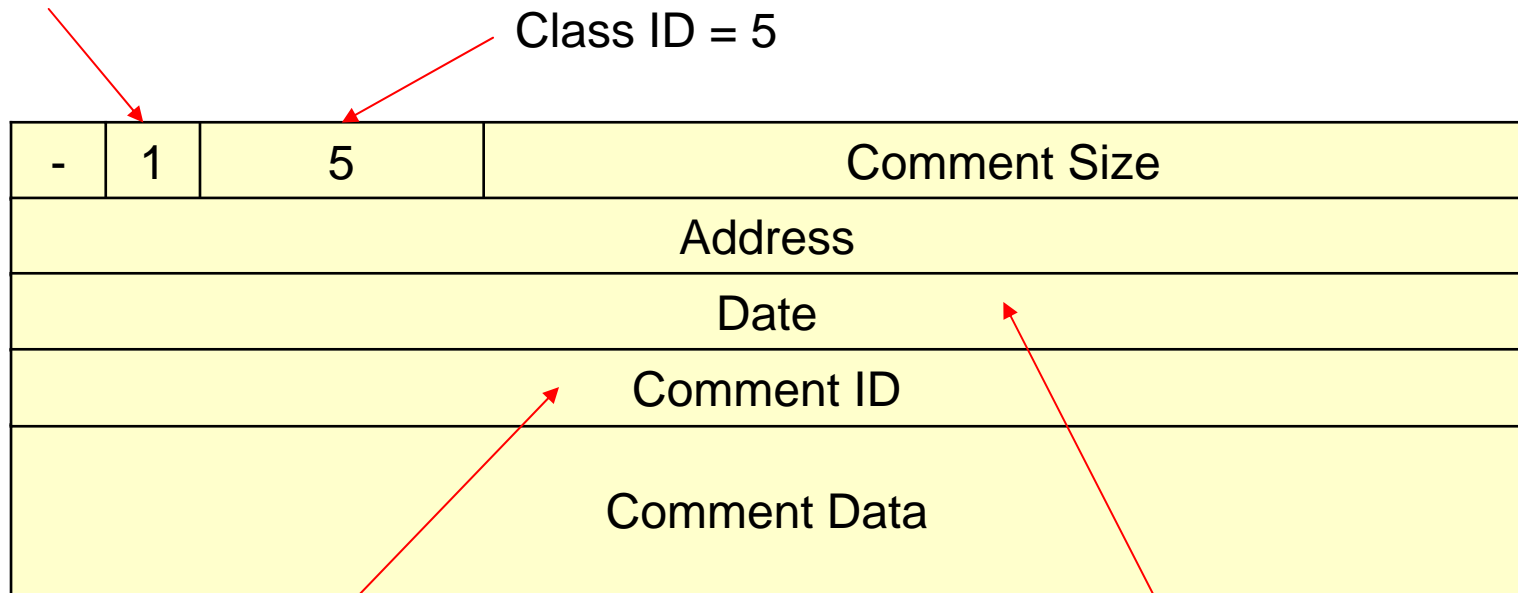
Status IDで何のデータか判断  
固定できるものは固定  
たとえば1 = F3 PPAC H.V.

Date=Event numberで時刻をタグ  
Layer 1の場合、Block転送時に  
Storeされるので、Event dataと  
同期はされない

# Comment data

Layerは1か2

Eventに同期してCommentをとる場合は2にして  
Event dataの下に入れる



Comment IDで何のデータか判断

0 = Header

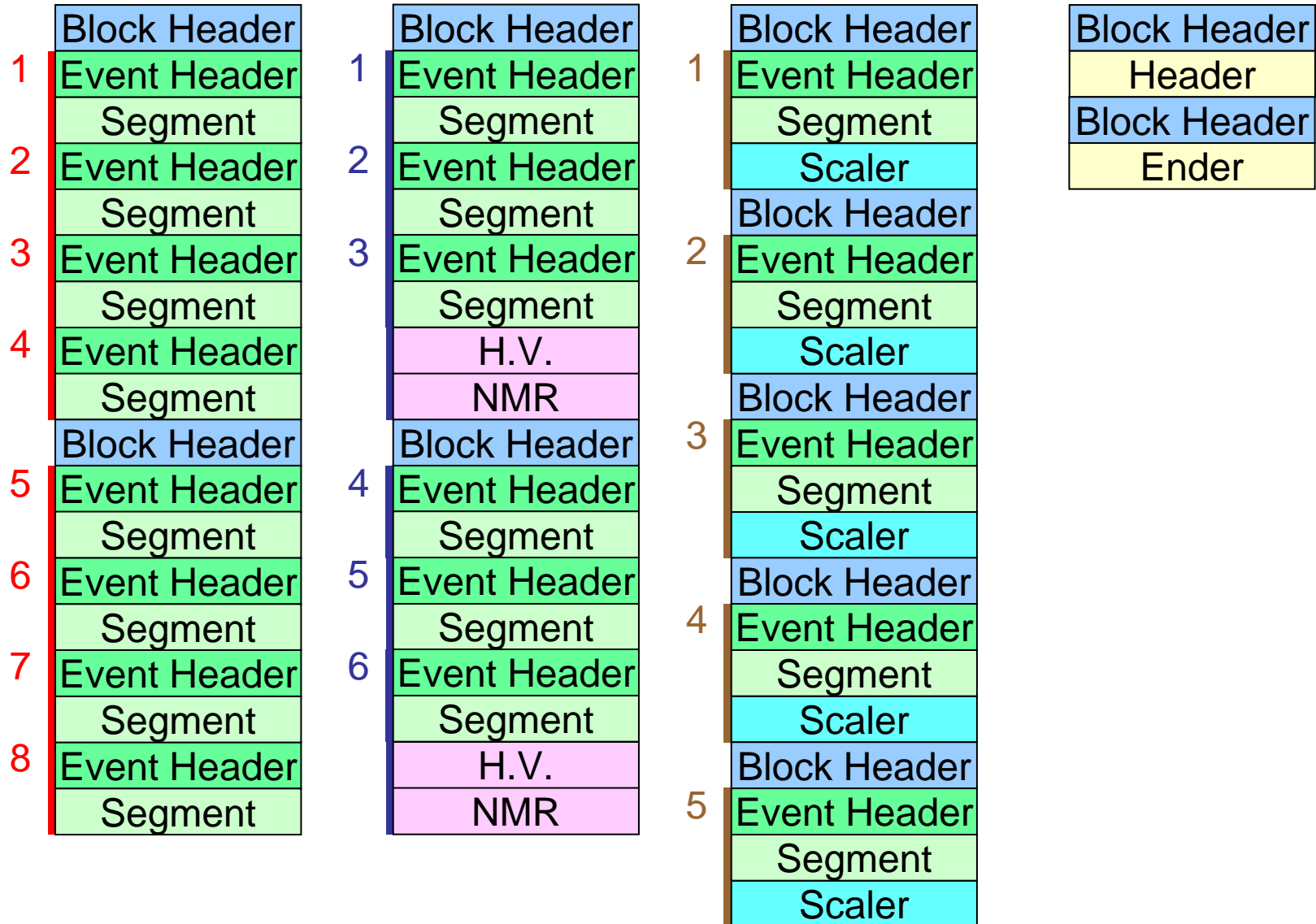
1 = Ender

2 = Warning

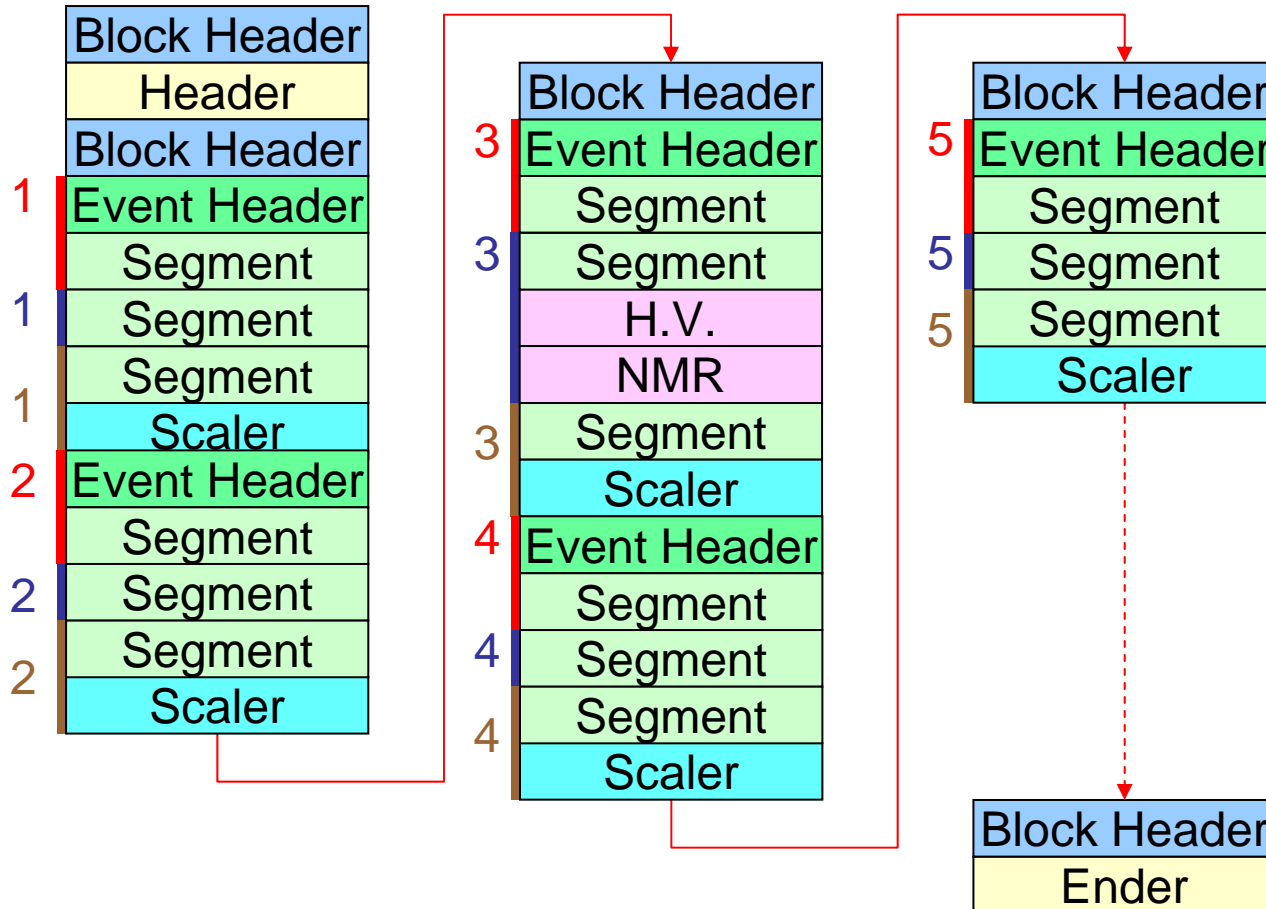
時系列をタグ

Layer 1の場合、Block転送時に  
Storeされるので、Event dataと  
同期はされない

# Event fragment



# Event assembly



# 最後に

- ▶ フォーマットの名称はRDF改めRIDF形式
  - ▶ RIBF Data Format
  - ▶ 拡張子は .ridf