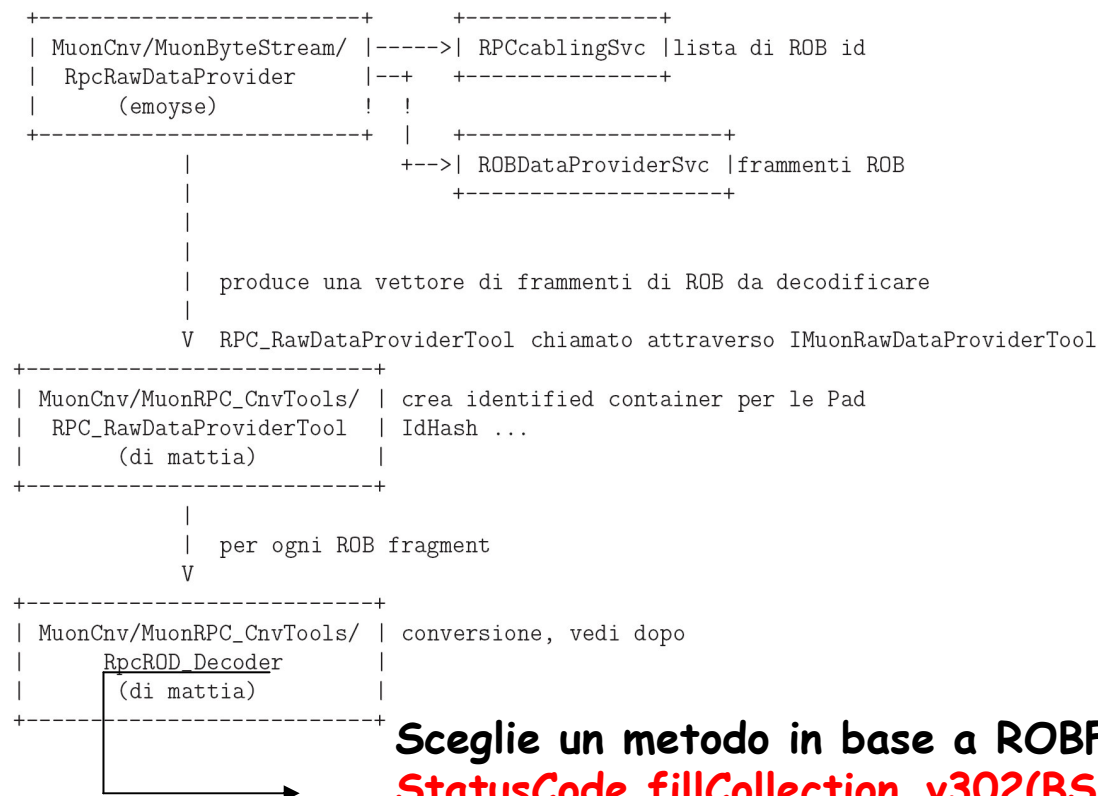


Esercizio: Controllare la struttura del buffer di dati e inserire un contatore per le eventuali failures

versione usata AtlasTier0 nightly 14.2.20.8

Conversione da bytestream (BS) a RDO (Raw Data Objects)



Sceglie un metodo in base a `ROBFragment.version`

```
StatusCode fillCollection_v302(BS data, const uint32_t
data_size, RpcPad& v, const uint32_t& sourceId ) const;
```

2 nuovi metodi inseriti in RpcROD_Decoder:

- StatusCode **checkdataformat**(std::vector<uint16_t>*) const; → Called from

⇒ Checks header, subheader, footer....:

fillCollection_v302

CM Footer

PAD Header

PAD Subheader

PAD Footer

CM Subheader

RX Header

PAD Prefooter

RX Footer

CM Header

SL Header

SL Footer

⇒ Count number of failures

⇒ Returns SUCCESS if no fail found, FAILURE otherwise

- void **printcheckformat**(); → Called from **finalize**

⇒ Print final statistics

Tested on the following cosmic data files:

`/castor/cern.ch/grid/atlas/DAQ/2008/82075/physics_HLT_Cosmics_MU2/
daq.ATLAS.0082075.physics.HLT_Cosmics_MU2.LB0000.SFO-2._0367.data`

Output di **printcheckformat**:

Number of processed events 722

==== number of **CM Footer** failures0

==== number of **PAD Header** failures0

==== number of **PAD Subheader** failures0

==== number of **PAD Footer** failures0

==== number of **CM Subheader** failures0

==== number of **RX Header** failures0

==== number of **PAD Prefooter** failures0

==== number of **RX Footer** failures0

==== number of **CM Header** failures0

==== number of **SL Header** failures0

==== number of **SL Footer** failures0

Utile esercizio per impratichirsi e cominciare a maneggiare il codice
 Necessario contattare gli esperti per capire esattamente
cosa c'e' di attualmente implementato per la gestione degli errori e
quali sono le vie "standard" per inserire ulteriori checks