Advanced Virgo DQ Model

Virgo detchar group

Virgo note VIR-0261A-15

→ Target = search groups

Rule #1: Keep it simple

Searches only need 2 inputs:

1/ when to run

2/ what to discard

Rule #2: Optimize

Limit the use of generic DQ input

→ be search-specific

State flags (1/0)

State flags = DQ flags monitoring the Virgo experiment

– high-level detector status (science, locked...)
– sub-system status (Detector Monitoring System)
– environment (Band-limited RMS)
– processes (h(t) reconstruction, frame generation/transfer...)
– hardware injections

How are they used?

1/ For detchar noise investigations
2/ To define valid time segments for analyses
3/ To define a veto recipe

Input for online analyses

→ A relevant selection of state flags is provided in a 16-bit state vector (@1Hz)
→ An appropriate bit mask is to be applied by online searches to define valid time segments

Input for offline analyses (transient & continuous)

→ All the state flags will be uploaded in DQSEGDB
→ A relevant selection of state flags is to be downloaded from DQSEGB and combined to define valid time segments

Old-fashioned language: valid segments = (science – CAT1 flags – injections)

Veto philosophy

A veto is defined by 3 ingredients

1/ **a veto recipe**: set of conditions (solely based on aux. data). The veto is **ON** if all conditions are met, **OFF** otherwise

2/ **a veto procedure**: when a veto is ON, a search trigger is rejected if a set of conditions using the trigger parameters is checked

3/ **a veto validity period**: a veto recipe/procedure is only valid for a limited time.

In general, a veto is no longer a pre-defined list of time segments. It should be seen as another rejection cut in an analysis pipeline (no associated dead-time!)

Veto procedure → mainly search group's responsibility

Veto recipe → "cooked" for every search
Statistic tools (UPV+EXCAVATor) over the search background triggers

Veto safety to be checked systematically!

Veto implementation

Online searches: *"we provide the best we can"*

→ one veto stream/pipeline (`V1:MBTA_VETO`, `V1:CWB_VETO`...) sampled @100Hz taking 3 values: 1 = the veto is ON, 0 = the veto is OFF, -1 = UNKNOWN

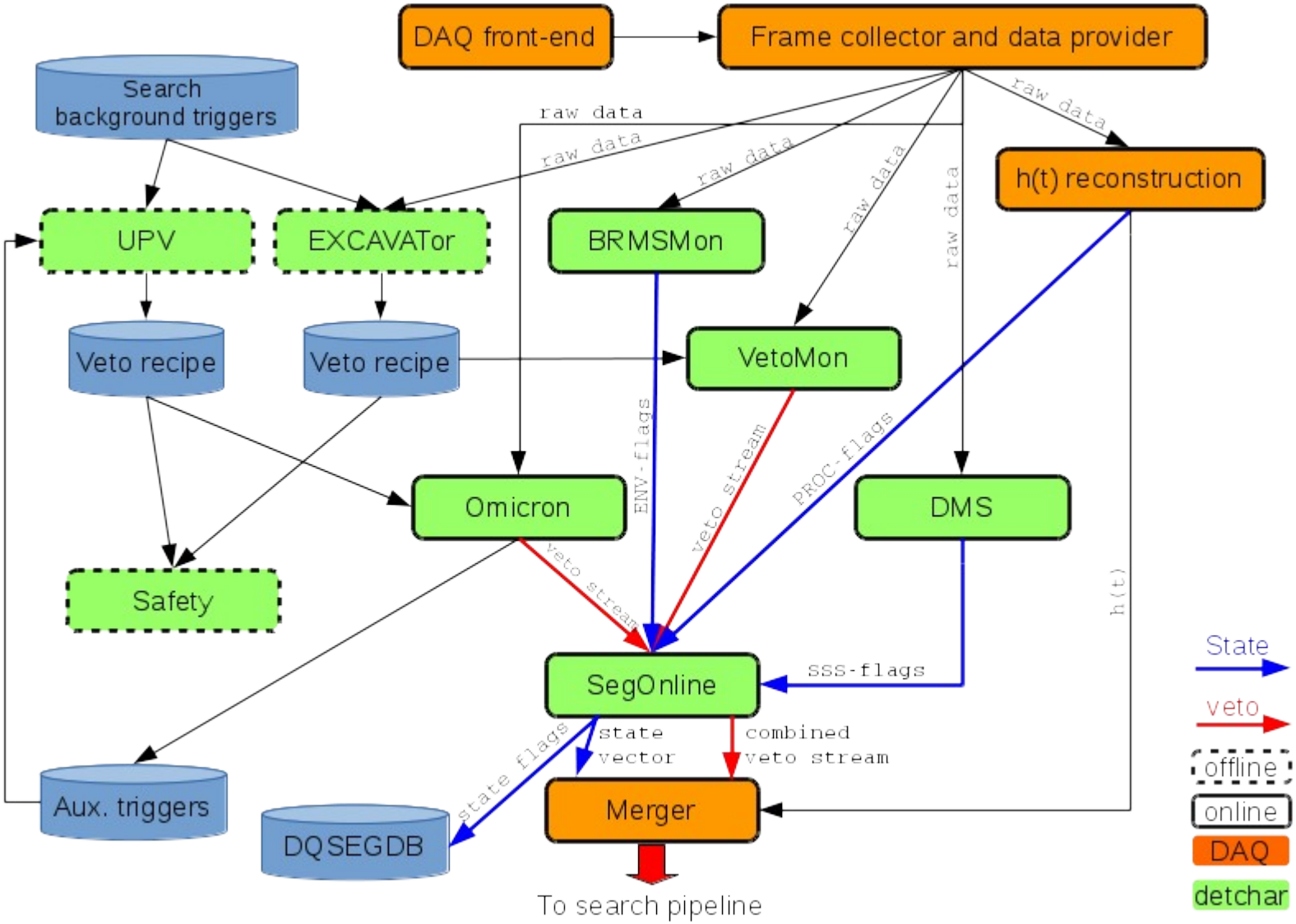
→ the veto stream is a combination of many veto recipes

→ the veto recipes are tuned offline using the most recent search triggers over the last 2 or 3 days.

→ the veto recipes are applied to online raw data and triggers (omicron) → veto streams

→ all the veto streams are combined into one: `V1:[PIPELINE]_VETO`

Veto implementation



Veto implementation

Offline searches: *“vetoes must be optimal”*

→ develop search-specific veto procedures

→ tune and apply the veto recipe on the same data set.

→ tailor ad-hoc vetoes based on the experience of the run

→ define veto validity periods based on the noise stationarity

→ tune the vetoes for every LIGO-Virgo searches. Search group's involvement is mandatory

Spectral noise rejection

→ Topic still in discussion in the Virgo detchar group

→ Lines are tracked by NoEMi and stored in a database (LineDB)

→ Lines are identified individually

→ Search pipelines must query LineDB to discard false candidates