AANet and JPP: A Status Report

Robert Bormuth
Leiden University and Nikhef
August 25, 2015

Concepts

JPP

- Software foundation for KM3NeT
- DAQ, Trigger algorithms and man more …
- Optimized for speed and performance
- Raw data and triggered data read-in

AANet

- Light wrapper for analysis purposes
- 5 easy to use classes: Evt, Det, Hit, Trk, EventFile
- mixed in some handy short-cuts (foreach …)

Python

AANet allows for Python steering files

- JPP hides a lot of its features in hard to read nested C++
- JPP needs setup for a lot of stuff analysis people want to happen automatically (e.g. applying time calibration, setting up the detector, . . .)

Usability

Documentation:

- Doxygen
- Example scripts

In the case of AANet this is enough for a work in, JPP examples hard to understand (more comments needed)

Detector handling

Idea

Both should share the same read-in and time calibration algorithms to reduce testing and error potential

↕

Status

Disjointed read-in, AANet reinvents the wheel