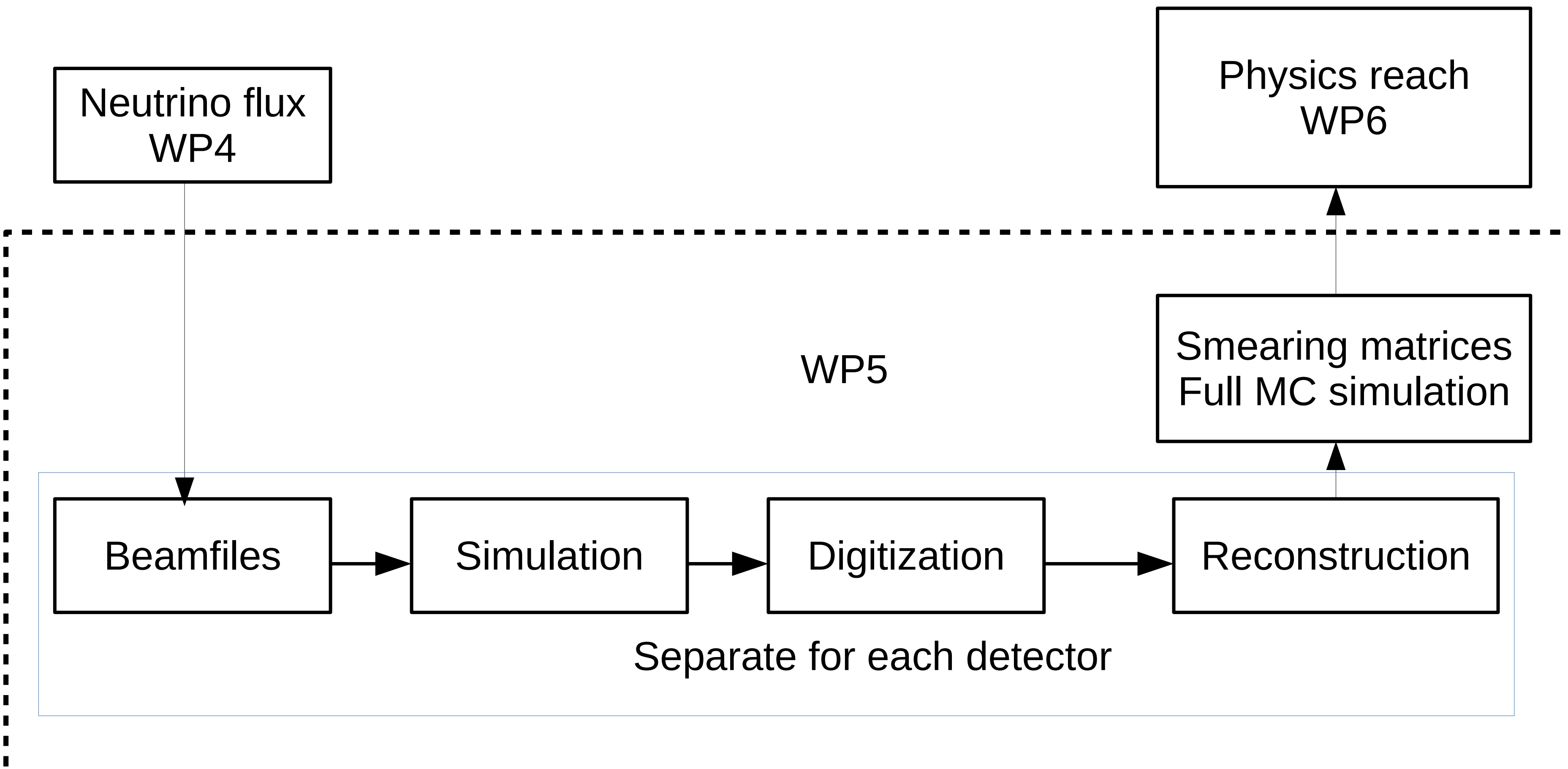
General design and status of the WP5 software and further plans

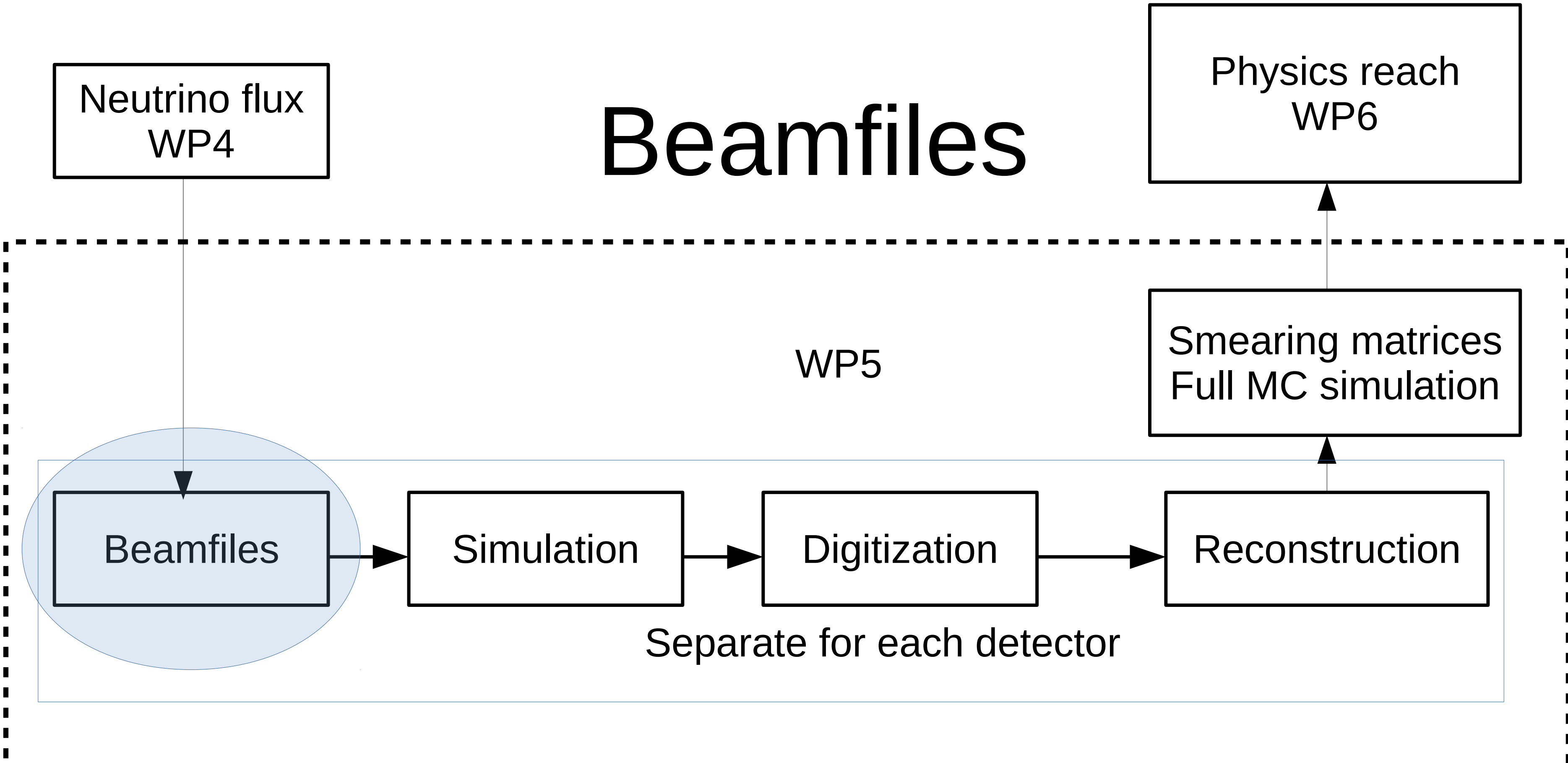
Budimir Kliček
IRB, Zagreb

ESSnuSB collaboration meeting
5-9 November 2018
Strasbourg

Simulation software

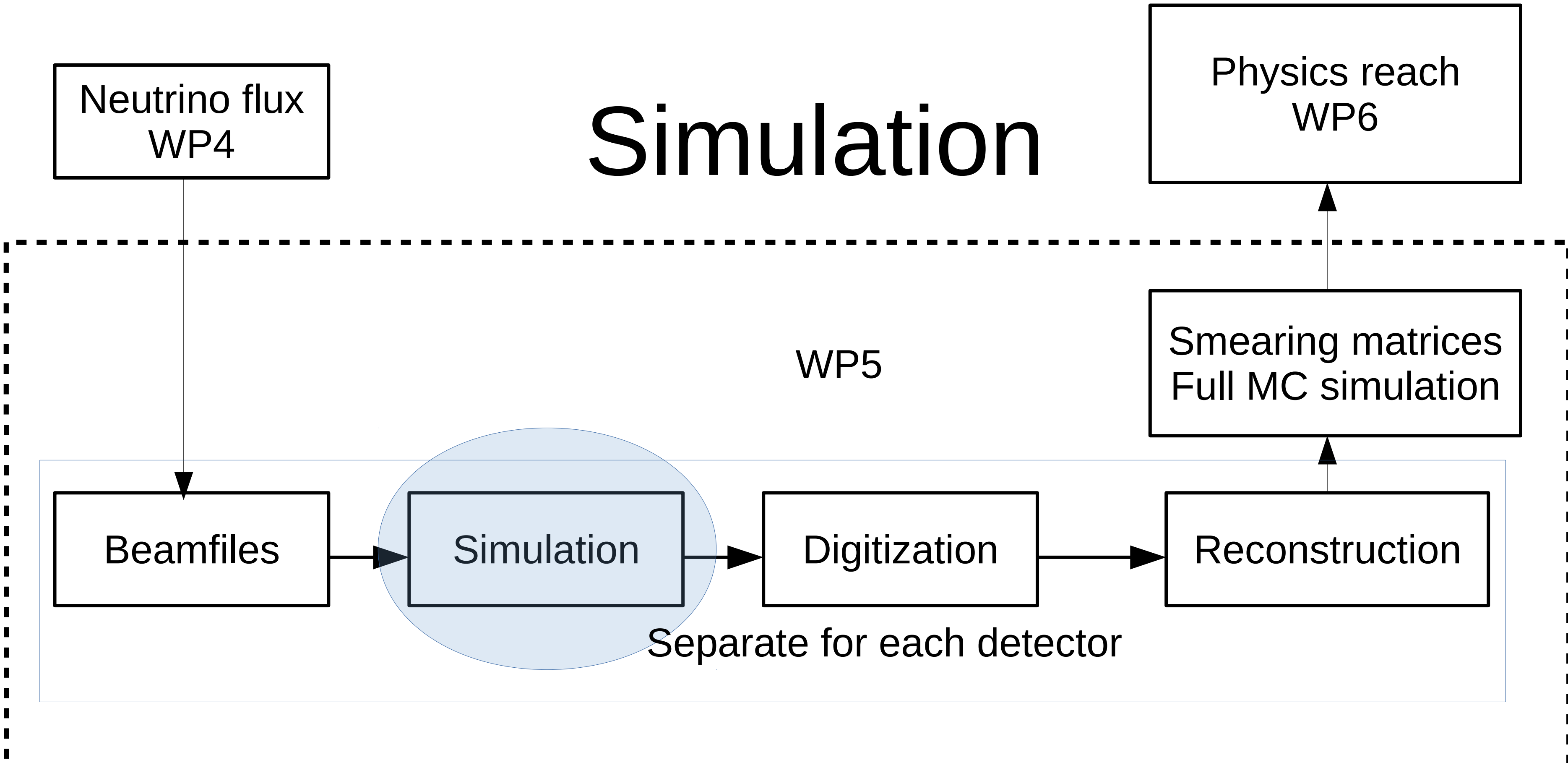


Beamfiles



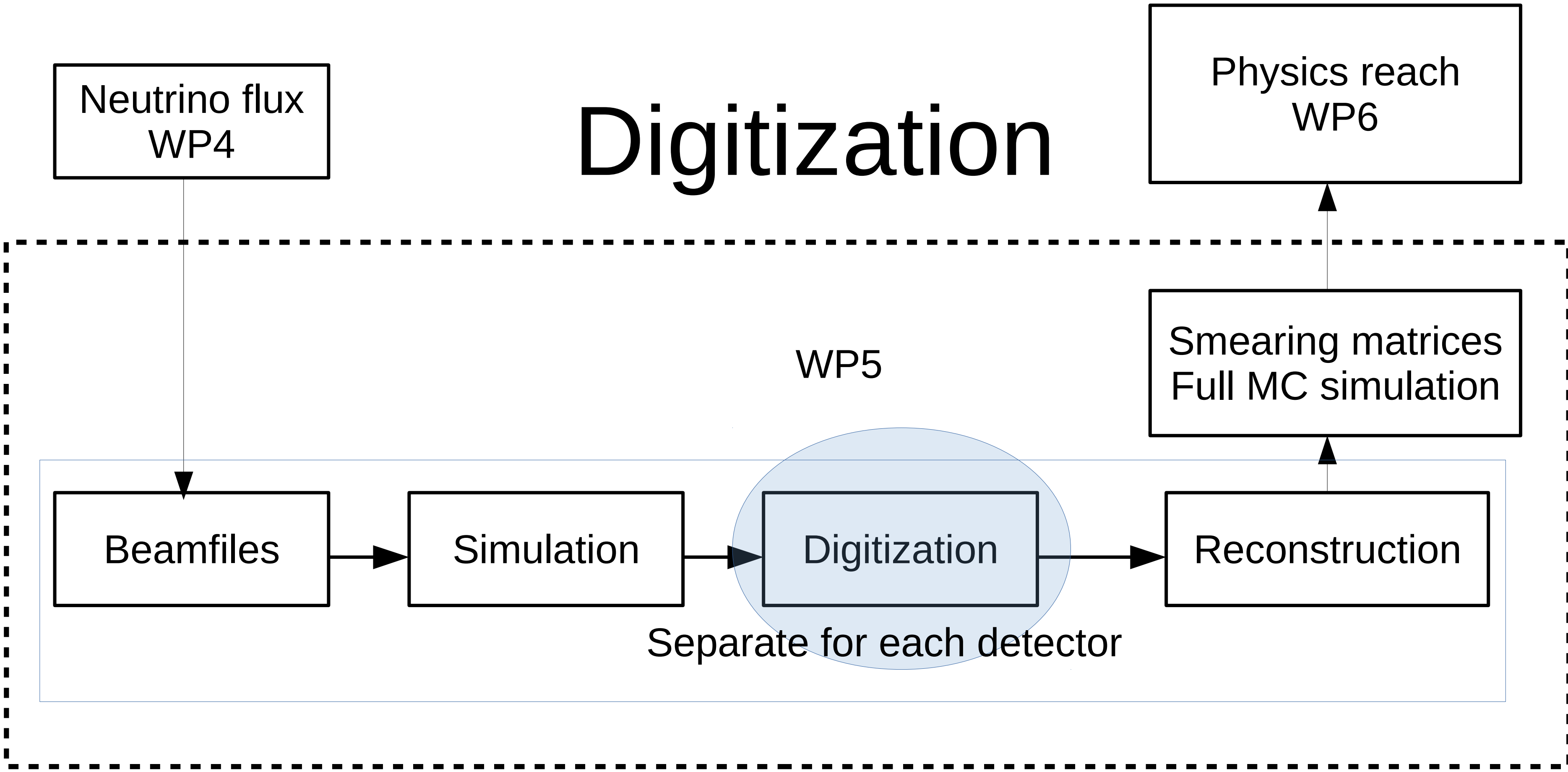
- simulation of neutrino interaction vertices
  - vertex position and time – geometry of detector required
  - incoming particles – neutrino + nucleus or electron
  - outgoing particles – charged lepton for CC interactions, neutrino for NC interactions + haronic system
- primary goal here is to simulate ESS beam neutrinos, but one can also add e.g. atmospheric neutrino interactions

Simulation



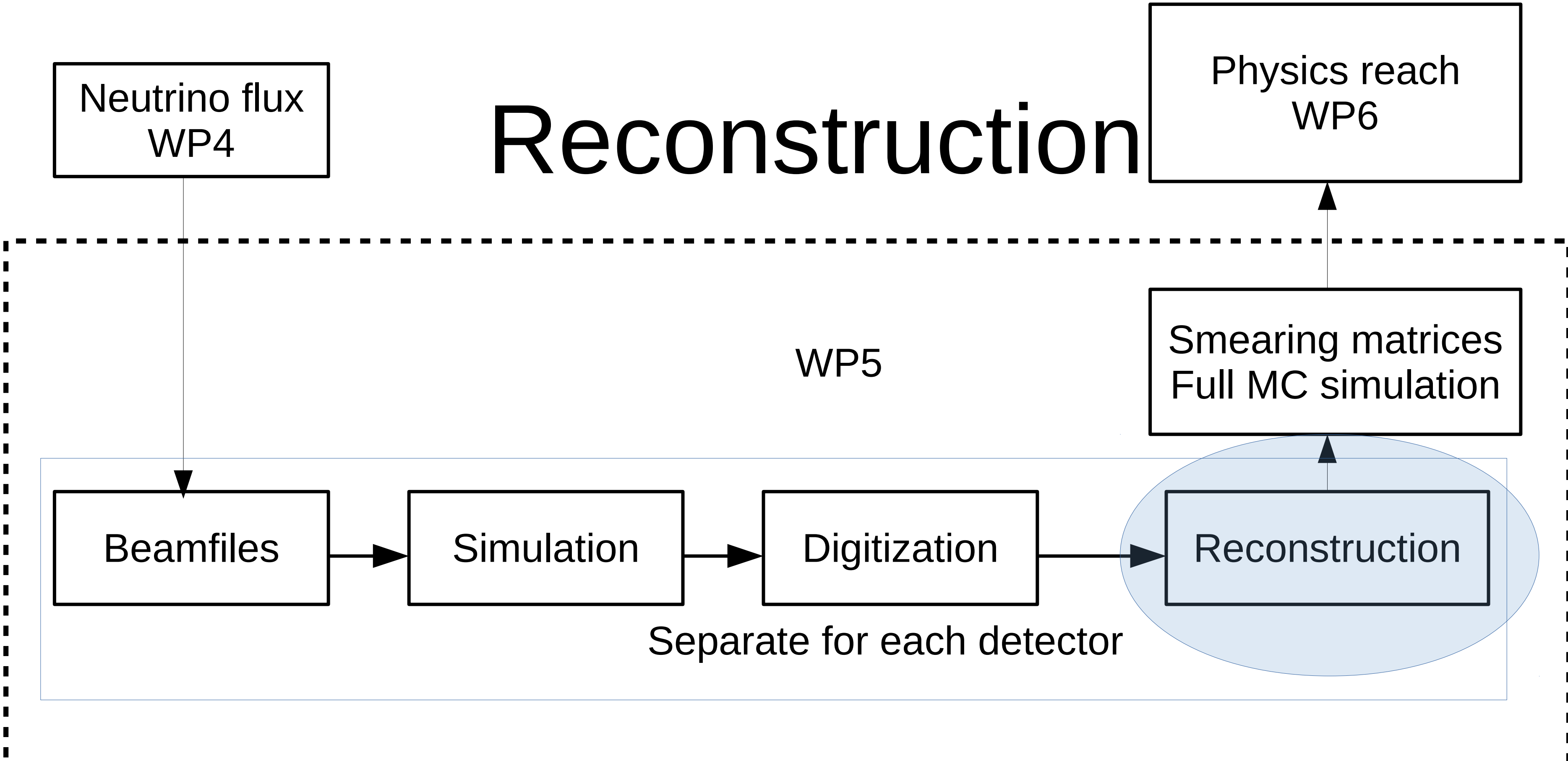
- propagation of primary outgoing particles
- calculation of *hit* positions
  - *a hit* is a point in space and time in which a particle enters and exits the sensitive volume of a sensor (e.g. a photocatode in a PMT)
  - hits used to simulate the response of a sensor in the next step

Digitization



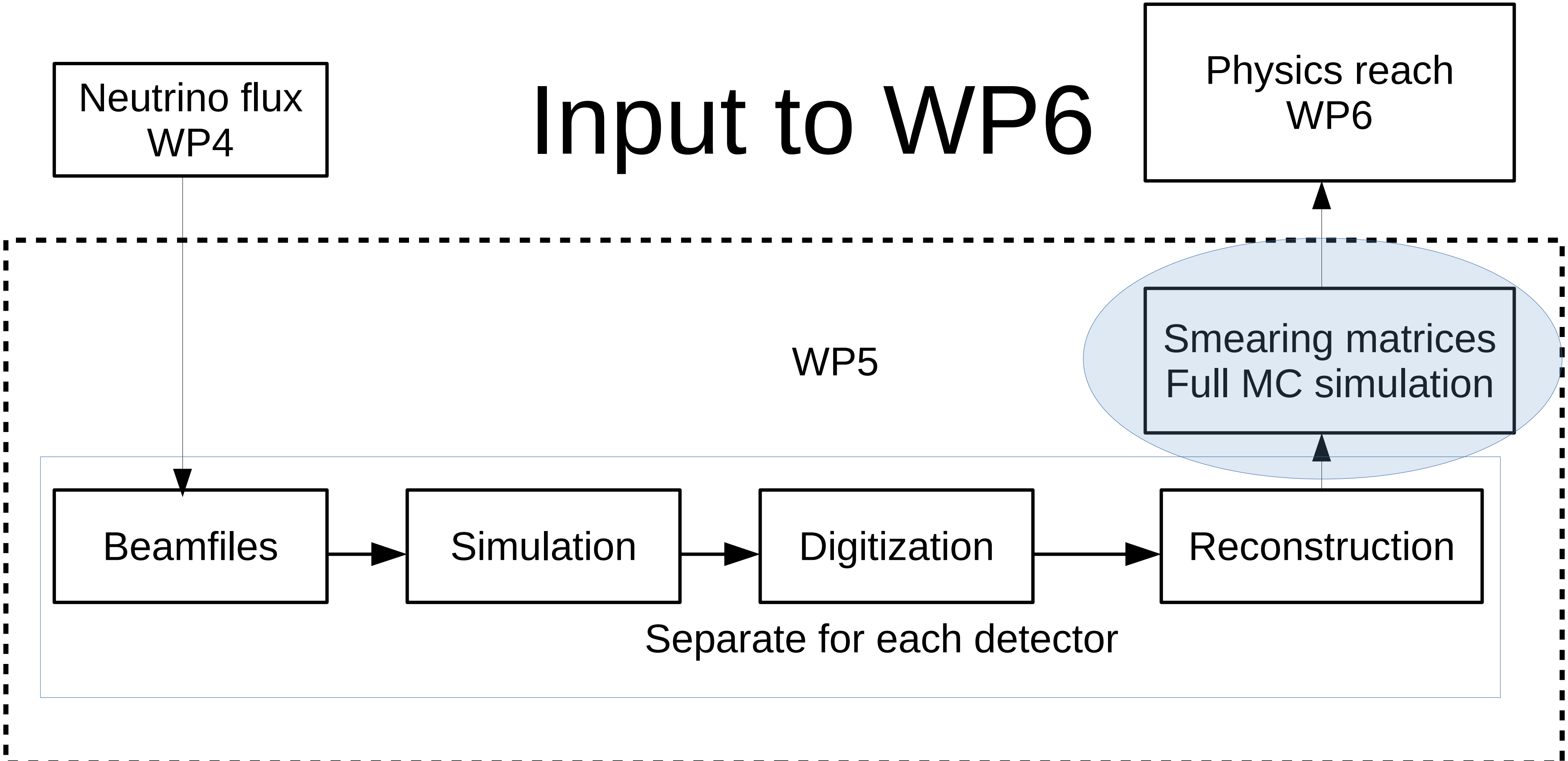
- *digit* – sensor readout (e.g. number of photoelectrons recorded by a PMT + the PMT id)
  - is defined both in MC and real data
- digitization is using hits to calculate digits
- after digitization we have simulated events in the exacty the same format as we'll have when recording the real data

Reconstruction



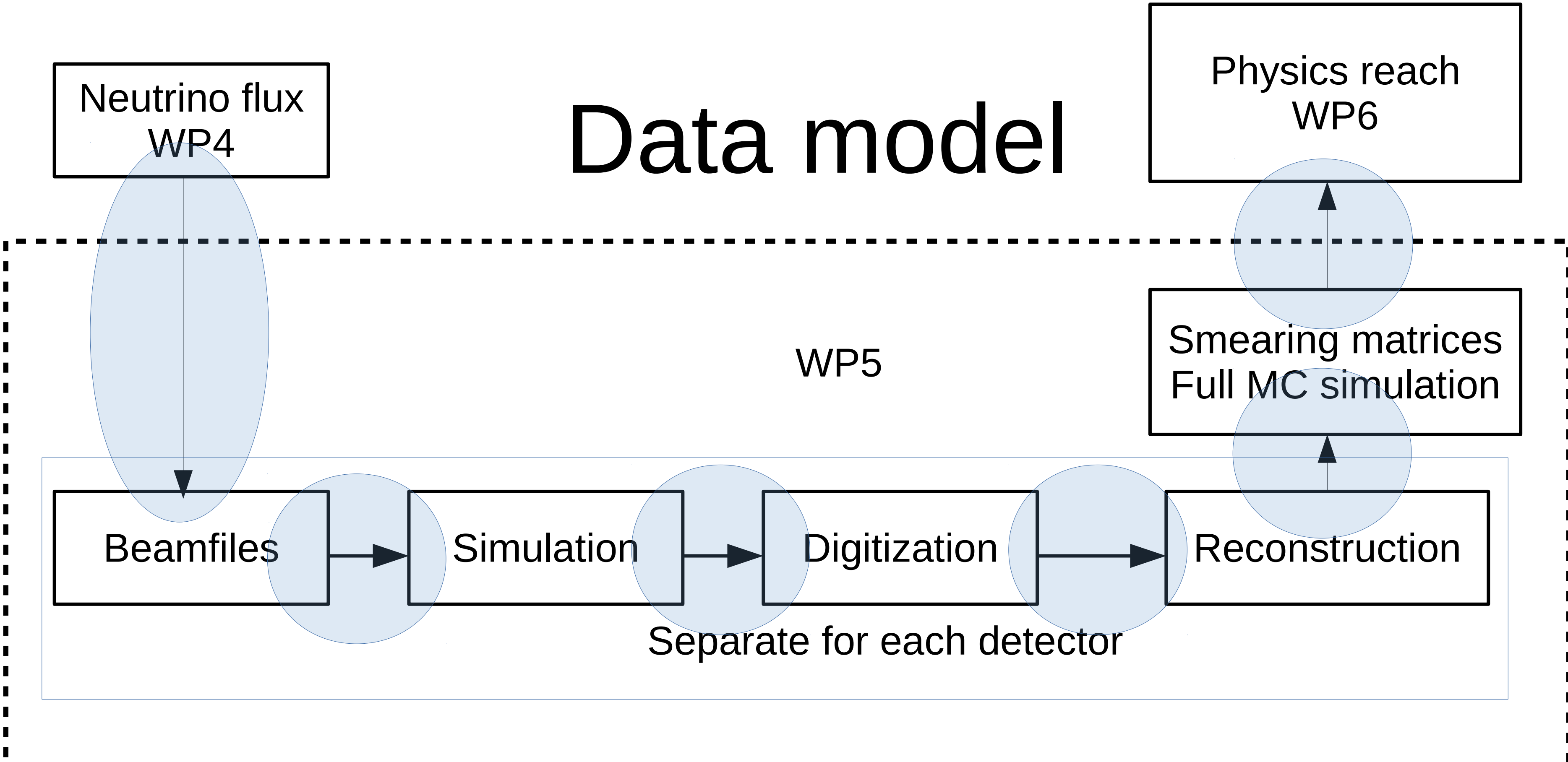
- using digits (and only digits) to reconstruct higer level objects
  - most importantly, reconstructing the particle tracks
  - e.g. for a water cherenkov detector – reconstructing rings, reconstructing tracks, particle ID …
- this can be done both on MC and real data

Input to WP6



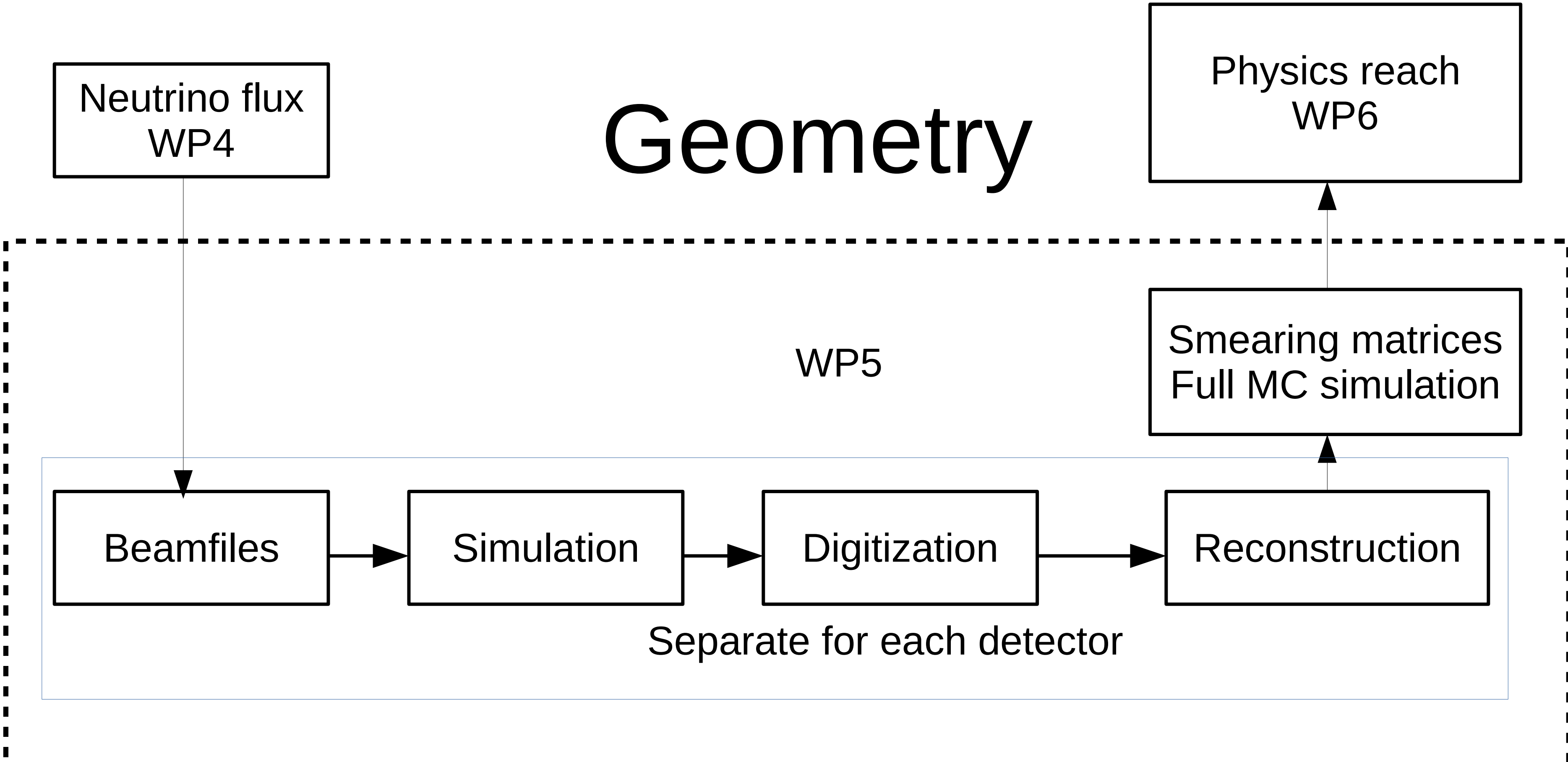
- We should have a clear understanding what exactly does WP6 need

Data model



- The format of the data
  - we will use *ROOT* objects to hold data and it's I/O system for persistence
  - **but we need to define the format ourselves** but later it must be fixed

Geometry



- Physical description of the detector
  - geometrical position of all the parts
  - physical properties of all the parts
    - materials, densities, masses, ...
- Each part of the chain needs to know about the detector geometry
  - we'll use the *TGeometry* class from ROOT framework, which can be used in all stages of the chain
- We will start with a simple water volume

Software stack

- C++11 ~~(or C++14, if agreed)~~
- FairRoot – framework to put everything together (see talk by Konstanin)
- ROOT v6 – data model, data persistency, geometry definition
- ROOTVMC – for particle propagation (using Geant3/Geant4)
  - Included in FairRoot
- Genie – beamfiles (see talk by Marco Roda)
  - GiBUU as an option down the road
- CMAKE – the build system
  - Included in FairRoot
- DoxyGen – documentation
- SVN – software repository
  - actually we use GitHub which is both SVN and git compatiblee

Current plan

- Set up a (public) SVN repository - DONE
- Define ESSnuSB coordinate systems - DONE
  - should be agreed by all working groups
- Define detector geometry using ROOT geometry package
  - for the far detector – start with the a simple water volume
  - for the near detector – need to discuss, I guess we can start with a small Cherenkov
- Define part of ESSnuSB MC data model
  - MC vertices, MC particles
- Genie
  - write flux driver and attach the detector geometry
  - integrate with FairRoot
- Define more of ESSnuSB MC data model
  - MC hits
- Geant propagation
  - should be easy using FairRoot
- **First milestone** – having hits at the water volume boundary using our software
  - ~~hopefully to be done by the Strasbourg meeting~~
- Debug this part of software and compare it to Memphys
- Go on to more realistic geometry, neutrino flux, PMT digitization, ...

ESSnuSB GitHub repository

- There is a ESSnuSB GitHub repostory at https://github.com/ESSnuSB
- It can be accessed both via git and SVN
- To use it (apart from reading) you need a GitHub account
  - please create a GitHub account using e-mail adress that you have on ESSnuSB mailing list
- So far it’s empty :)

IRB cloud

- We have 200 GB of storage at the IRB cloud
  - regularly backed up
    - and I keep a mirror on my computer, for additional safety
  - the capacity can be easily increased if needed
- Can be accessed via the link

  https://mojoblak.irb.hr/s/u1Y5MGydwQGujDX
  - password is the same as for the VM
  - read only (for security)
    - if you want to write be able to write, send me a mail (budimir.klicek@irb.hr)

Dedicated virtual machine

- A dedicated virtual machine has been set up
  - Linux Mint 18.1 LTS (supported until April 2021)
- included software:
    - ROOT 6.10-02
    - Genie 2.12.6
    - MEMPHYS
      - simulation
      - visualisation
      - see: https://www.youtube.com/watch?v=S1ExE0vbASM
- It's a good starting point to join the work
- Can be found on the IRB cloud
- You can run it out-of-the box using VirtualBox
  - or any other virtualization software, if you tweak the VM a little

WP5 software workshop

- Needs to be organized ASAP
- All people actively working in WP5
  - we need as many people as possible to agree on standards and conventions
- Five full days
  - I think this is the bare minimum
- **Main goals,** which we need to start serious work:
  - to produce the skeleton of the code
  - to define the data model
- I propose December 2018 or January 2019
  - but, unfortunately, can’t confirm which of these at this time..

WP4 + WP5 software workshop

- From discussion with Eric B.
  - it would be extremely useful to have a joint workshop between WP4 and WP5
  - we would integrate the neutrino beam simulation and detector simulation
- Could be done in March or April 2019?
  - depends on when the postdocs will be hired

Do we need this heavy framework?

- The short answer is **YES** :)
- This software is a complex tool for understanding physics of our detector, and we can not escape from this complexity. Therefore we must organize and and be very pedantic – we must have a framework and well defined conventions
  - and it should be as easy to use as possible – we want to remove the complexity from the people doing the analysis
- If we want to do any serious optimizing, we need to have a fully automatic system to generate MC
  - we don’t want to do any (serious) human work to reproduce MC after changing some parameter
- So, in my opinion, **the goal is to have a push-the-button system**: you set the parameters, push the button and get full MC, from neutrino flux to physics reach

What about MEMPHYS code?

- MEMPHYS is an existing code to simulate the big Čerenkov detector
- The idea is to make a simulation using modern tools and use MEMPHYS as a reference because
  - we should use the most up to date tools available (GEANT4, FairRoot, C++11, ...)
  - we want to expand and integrate the FD simulation with the rest of the ESSnuSB
  - software has a lifetime – a time after which is more expensive to maintain it that to make new one

Conclusions

- The work is going slowly, but it is progressing
  - in my opinion, it is better to go slow than to go in a wrong direction (slower is faster :)
- We definitely need to have a week-long meeting with all people involved to kickstart the creation of the ESSnuSB software framework for WP5
- Next year we will have pretty event displays to show at conferences :)

The end