**Working Group 1 - Objectives & Principles Minutes**

**22 October 2020**

The objective of the meeting was to discuss the results of the POC and determine the way forward. The following discussion and decisions were made.

* Discussions on the main take outs from the POC:
  + It is relatively easy to wrap a GUI on the solution.
  + Parallel computing is key but practically how to do it needs to be investigated, particularly how to enable both multi-core and clustering. It was decided that both was needed as the solution had to allow for quick desktop use and larger corporates with large policy books.
  + A rules engine/class structure with a hierarchy which allows for product rules to be stored and modified is necessary.
  + The solution should be modular so that components can be replaced/changed or worked on without impacting the others.
* The following modules where decided on:
  + Web GUI
  + Rules/class structure
  + Calculation engine
  + Input module to convert data into the required format for the calculation engine and rules/class structure.
  + Output model to write results into the desired format to be used by data ware housing solutions.
* It was decided there would need to be a decision on the way that each module communicates with each other.
* Each module could be written in a different language but it preferable to keep the number of languages to a minimum.
* The preference is Julia especially for the calc engine and rules/class structure. The reasons Julia was chosen are:
  + It is easy to understand and learn.
  + It is a language build for computations, so it is "fit for purpose".
  + It is as fast as compiled languages like (C++)
  + The only disadvantage is that it is not as widely used as other languages.
* Other languages considered where Python (including PySpark) and C++.
* It was decided that Julia would be the language subject to more investigation on its parallel computing capability.
* The following next steps where agreed to:
  + **Investigate Julia parallel computing and further training.**
  + **Start look at the design of the Input Module, calculation engine and rules/class structure.**