**Introduction**

The purpose of this document is to highlight key components of Rule Engine and how they are connected. Scope of this document is only to cover components briefly, separate ribbit posts are created to capture detailed information on each components.

**Component Diagram**

TBD: GUI makes use of Availability / Allocation REST services as shown in diagram, however details around cache update, what-if-analysis, etc. will be documented later.

**Component: Preprocessor (REST Service)**

Preprocessor is really a starting point for Rules Engine. It helps in preparing all data required for components: Availability and Allocation, and puts them into cache. Broadly, below are the data-structures created by Preprocessor.

1. Security Details ( TE id, TOL, TOS, etc.)
2. Rules-Security Details (Securities corresponding to each Security)
3. Fund Details (Fund, fund-level restriction, etc.)

Detailed information about Preprocessor component is available in ribbit:

Stats on Data-Structures:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sl. No** | **Data Structure Name** | **No. of Fields**  **(approx.)** | **No. of Rows**  **(approx.)** | **Data size**  **in MB (approx.)** | **Time to build**  **in sec (approx.)** | **Remarks** |
| 1 | Security Details | 100 | 4000 |  |  | Rows & Fields(columns) are shown for representation purpose only.  Data structure may be built in a form of structured objects  (returned as JSON) |
| 2 | Rules-Securities | 100 | 4000 |  |  | Rows & Fields(columns) are shown for representation purpose only.  Data structure may be built in a form of structured objects  (returned as JSON) |
| 3 | Fund-Details | 100 |  |  |  | Rows & Fields(columns) are shown for representation purpose only.  Data structure may be built in a form of structured objects  (returned as JSON) |

**Component: Availability (REST Service)**

Availability component is built using Akka toolkit and gets invoked by Autosys via. Java Client after Preprocessor completes building all data structures. Availability component requests Data-Structure-2 (Rules-Securities) from Preprocessor. Since data structures are already built and cached, Preprocessor will just return cached data without the need of re-computation.

Detailed information about Availability component is available in ribbit:

**Component: Allocation (REST Service)**

Allocation component is again built using Akka toolkit and gets invoked by Autosys via Java Client after Availability completes is calculation and data is persisted into Availability table. Allocation REST service makes use of Availability API and Data-Structure-3 (Fund-Details) to compute Allocation.

Detailed information about Allocation component is available in ribbit:

**Component: File Generator (Java Standalone)**

File Generator component gets invovked by Autosys after Allocation is successfully computed.

Detailed information about File Generator component is available in ribbit: