GTS

General Trading System Demo Architecture Draft

Communication Architecture

Legend:

CIL - Communication Interface Layer

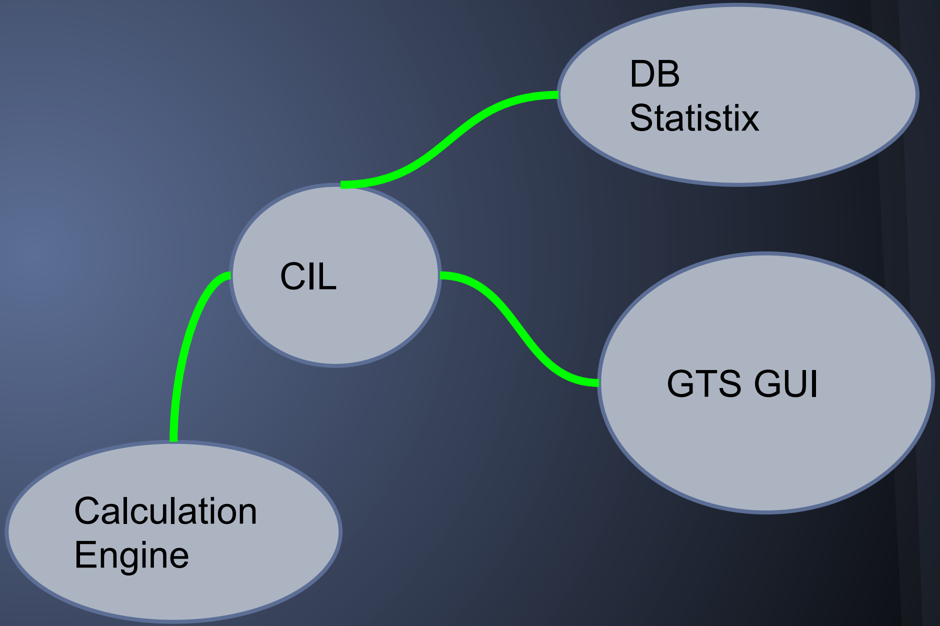
Calculation Engine - various evaluations on historical data, prepares risk factors, risk factor values, prepares samples, margins, backtesting, stress-testing …

Calculation Engine has no DB -

all data is read/stored on DB external aka **Statistix**

Communication is provided by ZeroMQ broker on CIL

Data is being transported using messages formatted by Google Protocol Buffers.



ZeroMQ as the messaging layer

The transport layer can be made via INPROC, IPC, MULTICAST, TCP. I’ve chosen TCP which provides unicast transport using TCP, the reason for the decision was because there is no need to cross the machine border. The multicast is out of choice because it behaves difficult when scaling up (while crossing many switches over network).

The Infrastructure pattern - QUEUE, FORWARDER and STREAMER.

The Message Pattern is based on Request/Reply bidirectional, load balanced and state based or publish/subscribe to publish to multiple recipients at once, upstream/downstream - data arranged in a pipeline.

Data model

General entities - via Espresso - automatic generation.

Manually modelled entities - compile via protoc

Development Environment

Actually this point is not implemented yet, I would suggest to use CMakeLists.txt for automatic build and linkage. The CmakeLists.txt would define what database model will be used to compile with the module.

Data access to Statistix

Write access

- insert, update and delete store requests
- always the timestamp should be used
- best performance would be reached if store requests are sorted per entity and type I/U/D, otherwise there is awaited very poor responsivity
- Data cache (as unordered boost::unordered map), not yet implemented