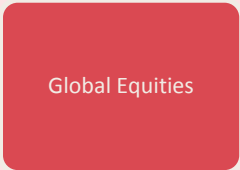# Problem Statement

Build an intraday, near *real-time,* Daily Profit/Loss 'PnL' aggregation module as a standalone component for Global Equities Desk.

Allows for easier access to multi-geographic, multi-sector portfolio aggregation with flexible handler to support real time changes to market values.
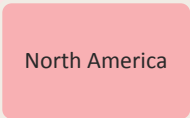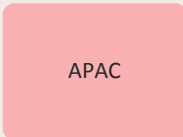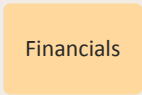
Context & Dataset(s)
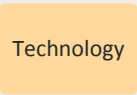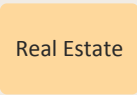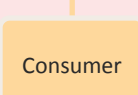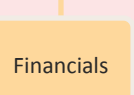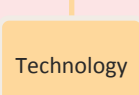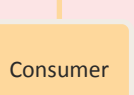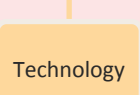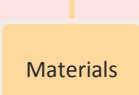
Hackathon Objectives

Objective 1 – Ingest 'Bootstrap' holdings data from End of the Previous Day　　　　(50 points)

Objective 2 – Ingest real-time changes to stock prices 'intra-day' by processing mini-batch of files as they appear in a folder　　　　　　　　　　　　　　　　　(150 points)

Objective 3 - *Calculate* and *Show* 'real-time' PnL by *the handle* through an interactive command or visual interface, without restarting the file processing. Examples, but not limited to:

　　　　　　　　　　　　　　　　(150 points) – Calc / Command line

　　　　　　　　　　　　　　　　(150 points) – Visual and/or Time-sliced

showPnL /global

showPnL /global/nam

showPnL /global/apac

showPnL /global/nam/fin

showPnL /global/name/fin:BAC

(Note: Candidate can come up with alternate command conventions (like parameter prefixes like –h, -s etc.,) as long as they are intuitive and not constricting)

Utilities, Helper Classes & Solution Assumptions

1. **File Handler Helper** – A file handler class will be provided that parses bootstrap and any 'real-time' files into a structured value object(s). It is candidate's responsibility to leverage this code in any way helpful to incorporate into his/her main module(s). Candidate can build a better one his/her own for additional credits.

2. **A graphing utility** – A graphing utility code will be provided to show any time-series data visually. It is candidate's responsibility to leverage this code in any way helpful to incorporate into his/her module(s). Candidate can build a better one on his/her own for additional credits. Example: This utility shows a line chart with X-axis (time in 1 hour increments), Y-Axis (Price, PnL, etc.,).

3. **FX conversion utility** – A static FX conversion utility code will be provided to convert prices or any derived values from one currency to another currency. It is candidate's responsibility to leverage this code in any way helpful to incorporate into his/her module(s). No additional credits for building his/her own as this is a simple static conversion utility.

Solution Assumptions:

a)   Stock holding amounts do not change from the previous day. If Holding amount is provided in the file, it can be ignored or processed, no changes to holding amounts

b)   Prices may not change at all during the day for a given set of stocks

c)   Prices will not be negative or zero

d)   No new stocks (or holdings) would be added or removed to the portfolio during the day

e)   Regions and Sectors are static

f)   A new file or multiple files might be dropped into a 'landing folder' and should be immediately processed and PnL should be reflected

g)   All times quoted are in EST (Single time-zone throughout the data sets)

h)   Files are provided in 1 hour increments and snapshotted respectively (no other intermediary time periods will be posted) – Only following could be posted (PREV_DAY, 7AM, 8AM, 9AM, 10AM, 11AM, 12PM, 1PM, 2PM, 3PM, 4PM, 5PM, 6PM, 7PM)

i)   Files will be sequenced for ease of use but candidate can manually drop files in a sequence if needed (No penalties for manually dropping the files in the drop zone) during the Demo(s)

j)   No persistent data store is expected, module needs to be able to restart from "Start of the Day" if required by terminating all the appropriate daemons and restarting in necessary sequence

Solution Presentation & Judging

Correctness	Extendibility	Modularity	Presentation of Solution
Exception Handling	Elegant use of Data Structures	Ease of Use *(e.g., multi-currency results)*	Performance