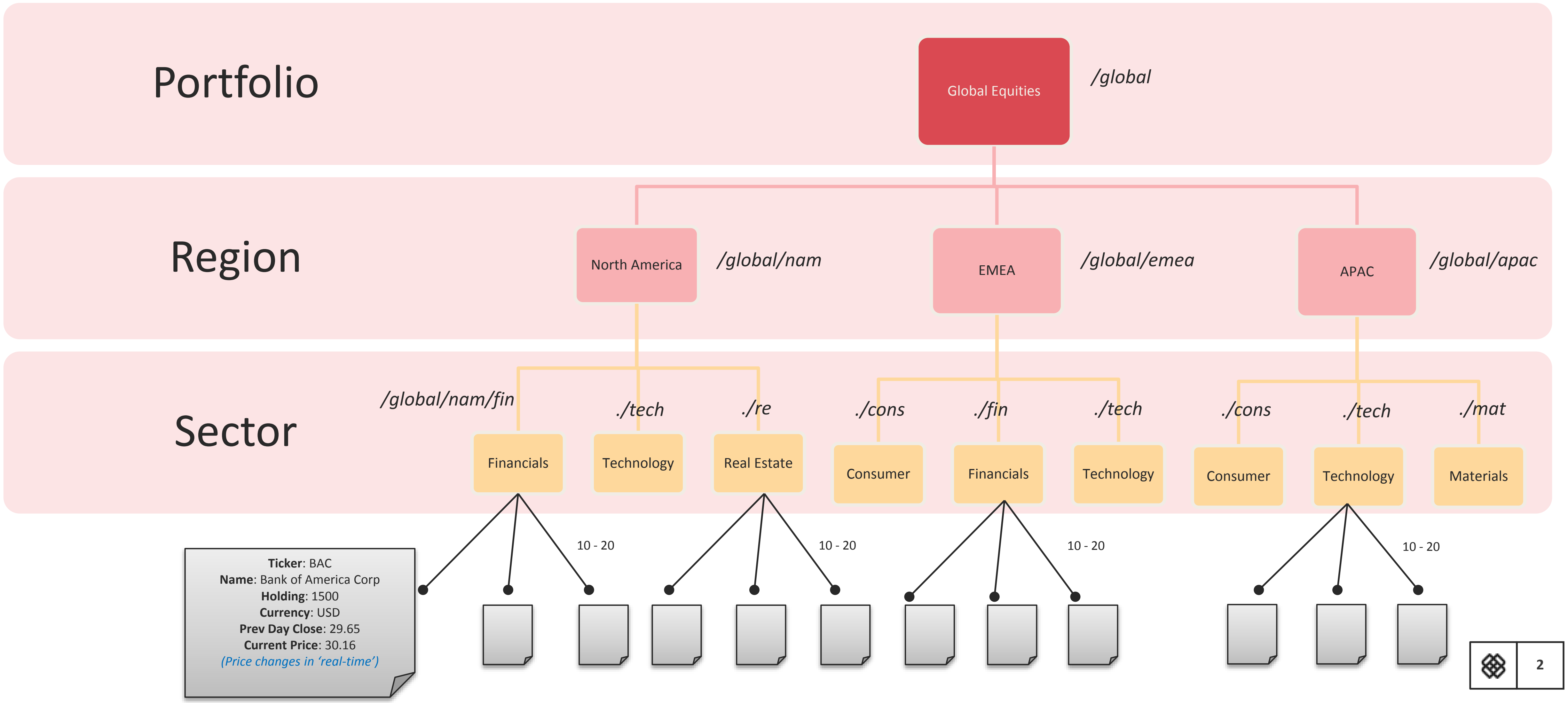
Problem Statement

Calculating daily Profit & Loss ('PnL') of portfolio of stocks is a key part amongst many processes that are typical for our clients.

We want you to build an intra-day Profit/Loss 'PnL' calculation and aggregation (group of stocks) module as a standalone component for a hypothetical Global Equities (Stocks) Desk.

The component should provide summarization of profit (or loss) across multiple groupings of stocks when market values (stock prices) change.

Context & Dataset(s)



Ticker: BAC
Name: Bank of America Corp
Holding: 1500
Currency: USD
Prev Day Close: 29.65
Current Price: 30.16
(Price changes in 'real-time')

Hackathon Objectives

Objective 1 – Ingest 'Bootstrap' holdings data from End of the Previous Day

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

Objective 3 - _Calculate_ and _Show_ 'real-time' PnL by _the handle_ through an interactive command or visual interface, without restarting the file processing that might happen in the background.

Example of tests, but not limited to:

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 and Helper Classes

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) This component needs to be a Core Java application/solution that handles simultaneous (non-blocking) loading of files based on availability and showing visually through interactive commands latest Profit/Loss of the Portfolio at various regions

b) 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

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

d) Prices will not be negative or zero

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

f) Regions and Sectors are static

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

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

i) 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)

j) 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)

k) 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

l) No need to handle changes to files and re-loading of same timestamp of prices

m) Participants can use any open source or 3rd party packages as long as the components are pre-built Portfolio and Profit/Loss calculators

Basic Calculations

1. The system receives a 'bootstrap' snapshot of the portfolio from the previous day's close of business and uses that as a basis for all PnL calculations. Throughout the 'day' (defined as multiple files) the system receives intra-day price updates and is be expected to process this information in real-time.

    1. *PnL in a currency = **PrevDay** <u>Value of Portfolio</u> – **Current** <u>Value of Portfolio</u>*
    2. ***Value of portfolio** = Sum of all **stock values***
    3. ***Stock value** = # of stocks of a particular company * stock price in a single currency*

<u>All the necessary utilities/helper code/data files/instructions/sample excel for review are available for participants at:</u>

https://github.com/skandula76/PS_NY_Fall2018_CoreJavaHack

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