

Building A Quantitative System

A Do-It-Yourself Guide

Parijat Garg, CFA

**To a man with a spreadsheet,
every problem looks like a vlookup candidate**

~~Mark Twain~~

Don't

**Beware of entrance to a quarrel, but being in,
Bear 't that thy opposed may beware of thee.**

Shakespeare

Applications

- Power Screening
- Technical Research
- Trading Strategies
- Fundamental Research
- Risk Assessment
- Reporting

Considerations

- Data Architecture
- Technology Stack
- Sourcing / Cleaning
- Maintenance / Pipelines
- Identifiers
- Accessibility / APIs
- Documentation

Data Architecture

- Consider purpose of the exercise, while maintaining some flexibility of shift in requirements over time
- Identify *types* of data that will be required
- Identify the most common use cases in terms of consumption

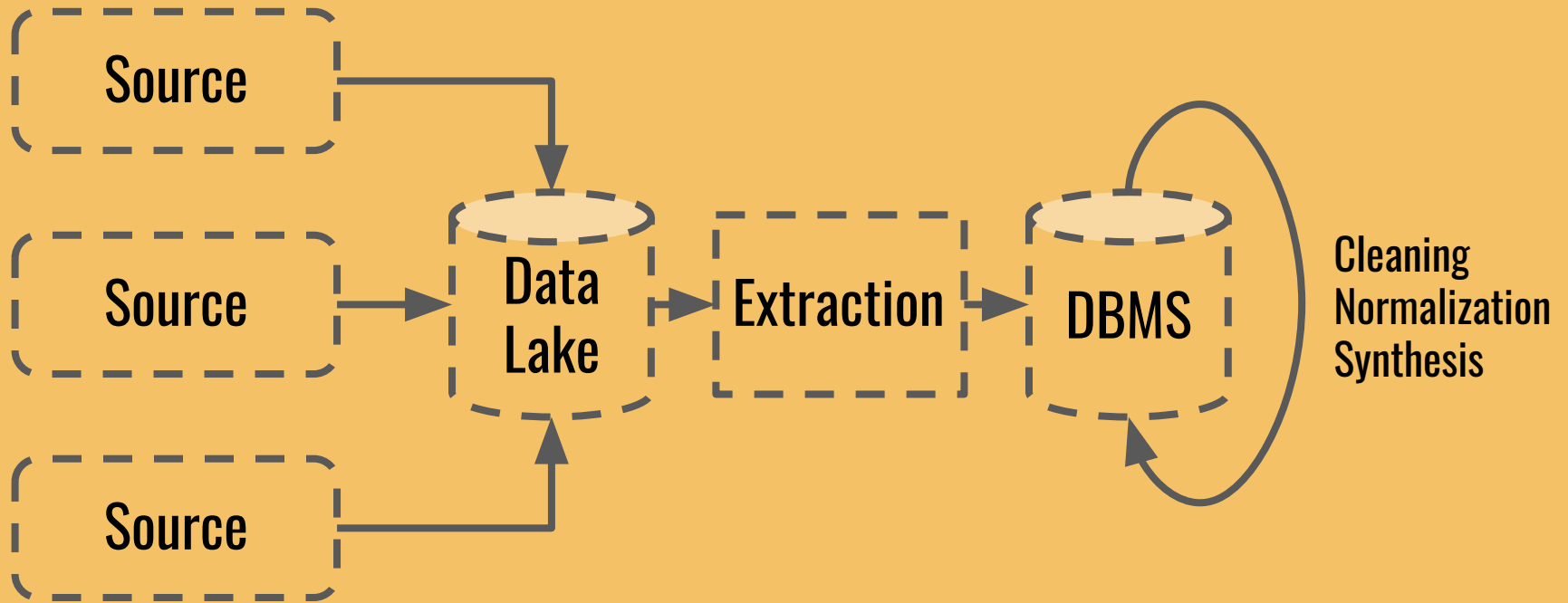
Technology Stack - Considerations

- In-house capability
- Regulations (data security rules)
- Budget
- Nature of data and analytics
- Nature of end-use

Simple Technology Stack - Example

- AWS S3 - Data Lake
- MySQL
- Python + Pandas & family
- Flask + Nginx - for web interfaces or Excel API

Data Acquisition Pipeline



Maintenance / Pipelines

- Automated periodic data ingestion/updation
- Occasional/event-based updation
- Ability to re-generate data (with corrections)
- Keeping the pipelines up and effective
- Data integrity, identifier drift and other issues
- Source format drift

Identifiers

- Identifiers are a massive challenge in finance
- Exchange tickers, ISIN/CUSIP/SEDOL, Bloomberg, Reuters, and others
- Security vs. Entity
- Change management
- Ideal: Internal identifier for all data, managed by concordances

Accessibility

- Abstract data layer away into a library
- Modularize any analysis to allow reuse
- Ensure standard formulas be written only once
- Keep presentation layer separate from analysis

All standard rules of software development

Documentation

- System evolves quickly and ends up with lots of moving parts and capabilities
- Try to develop documentation as early as you can
 - Will help prevent accidental re-implementation of features
 - Speed up implementation of actual analyses
- Applies not only to code but to deployment architecture - there will tonnes of scheduled scripts, etc.