Evolution of Financial Risk Management

Harnessing software and data engineering technology

Dr Sioned Baker
IET 19 May 2026

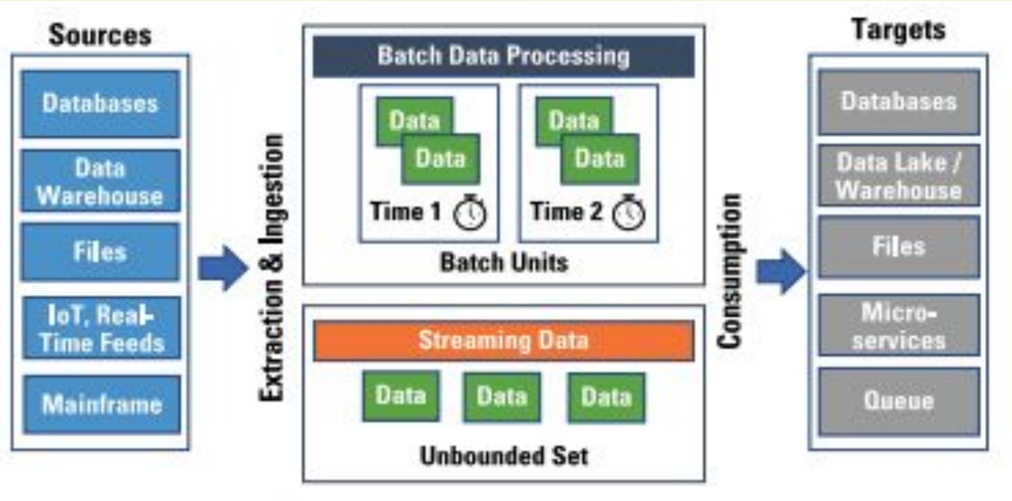
Getty Images (@gettyimages)

Sioned Baker
Data engineer @ ki insurance

Previously worked

- Thomson Reuters
- Application Networks
- Scott Logic
- ADC Metrica

Data Engineering



- Building data-pipelines:

Extract-> Cleanse-> Transform -> Serve

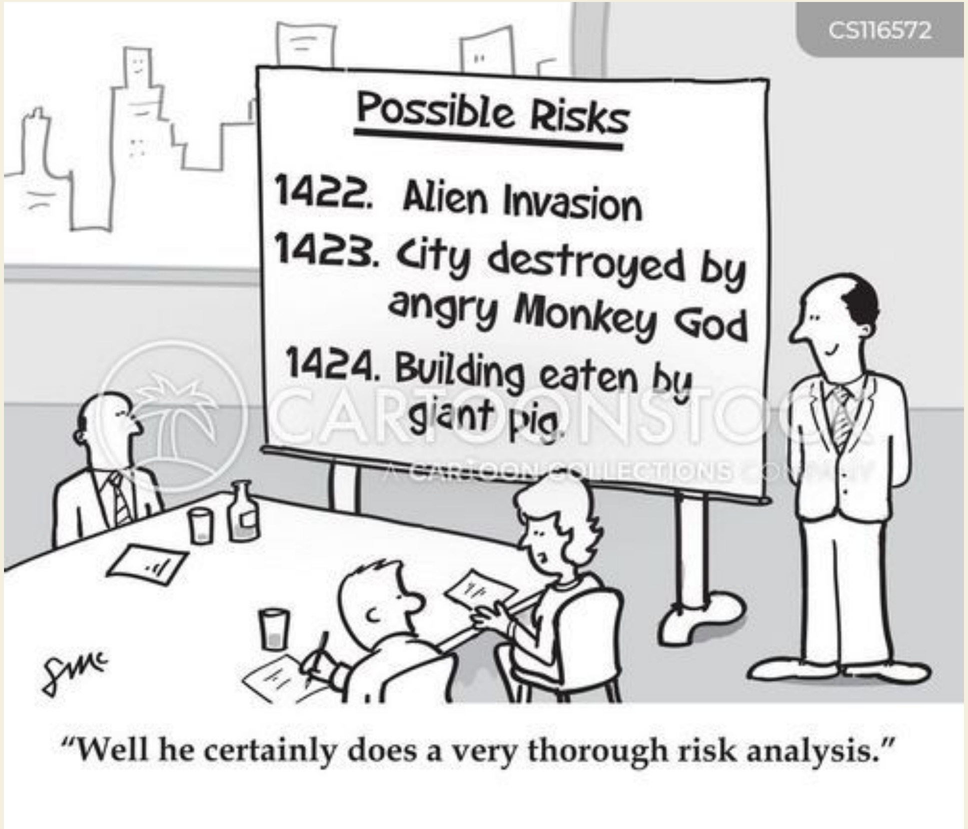
- Maintaining metadata
- Security & Governance
- Storing data
- Compute & Process data

Some of the technologies:

Cloud platforms:

What is Risk?

It combines the probability of an adverse event occurring with the severity of its consequences.



cartoonstock.com

What is financial risk management?

Practice of protecting economic value by managing exposure

- *Market Risk*
- *Credit Risk*
- *Liquidity Risk*
- *Operational Risk*

How is it managed?

Banking sector – Basel Accords

Insurance – focus on solvency and ability to pay claims

Investment management – diversification and optimization

Insurance

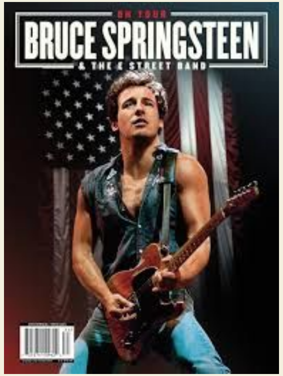
…is a financial product and risk management tool that protects individuals and businesses from financial losses.

LLOYD'S

…acts as a hub where syndicates (underwriters) come together to pool capital and cover complex, high-risk or unique risks

- Marketplace structure
- Specialized Coverage
- Syndicates underwrite policies
- Global Reach

Niche in speciality insurance



Risk Modelling Challenges

- Small number of claims with high value
- Low volume, Low frequency =>Limited data representation
- Long tail – claims can come years later

Predict claim probability by
- Severity
- Frequency

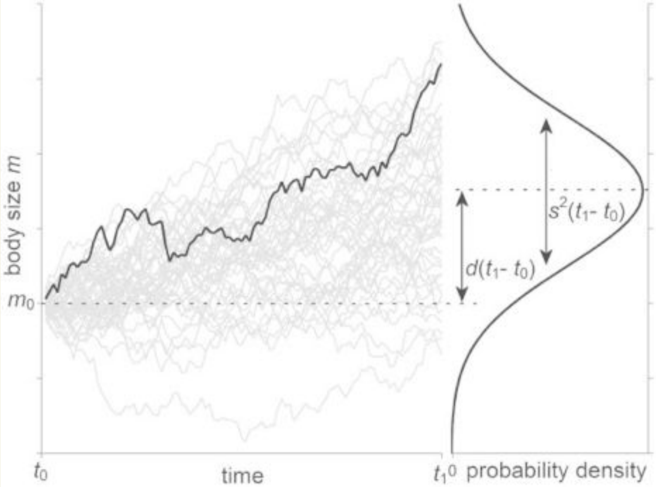
Data → Model → Expected loss cost

Modelling Market risk

FTSE 100

S(t)=S(t-1) exp(u)
=> u(t)= ln[S(t)/S(t-1)]
=> **Periodic Daily Return** (PDR)

PDR is random variable of time



Modelled as a generalised Wiener process with drift

Asset Classes

- Equities (Stocks)
- Fixed Income (Bonds)
- Cash Interest rate
- Commodities
- FX (Foreign Exchange)

Derivatives

contracts deriving value from underlying asset

- Futures
- Forwards
- Options
- Swaps
- Structured Products

numerically intensive calculation

Pricing Models
Calculate present value (PV) of the asset

Examples

- Equity options
- Bond future options
- IR Swaption
- Commodity future
- FX Swap
- Barrier & exotic options

Calculate Risk

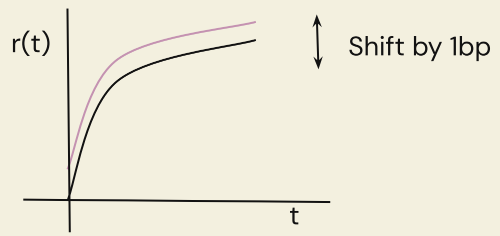
- Perturb the underlying risk factor
- Reprice the asset
- Calculate the delta in PV

Taylor series expansion

Risk factors, e.g. Commodity price, interest rate, fx rate, volatility, etc.

$$PV = f(x_1, x_2, \ldots\ldots)$$

Underlying market data, e.g. Yield curve



Greeks:

Delta Δ: rate of change value wrt underlying price

Gamma Γ: rate of change in delta wrt underlying price

Vega: sensitivity to volatility

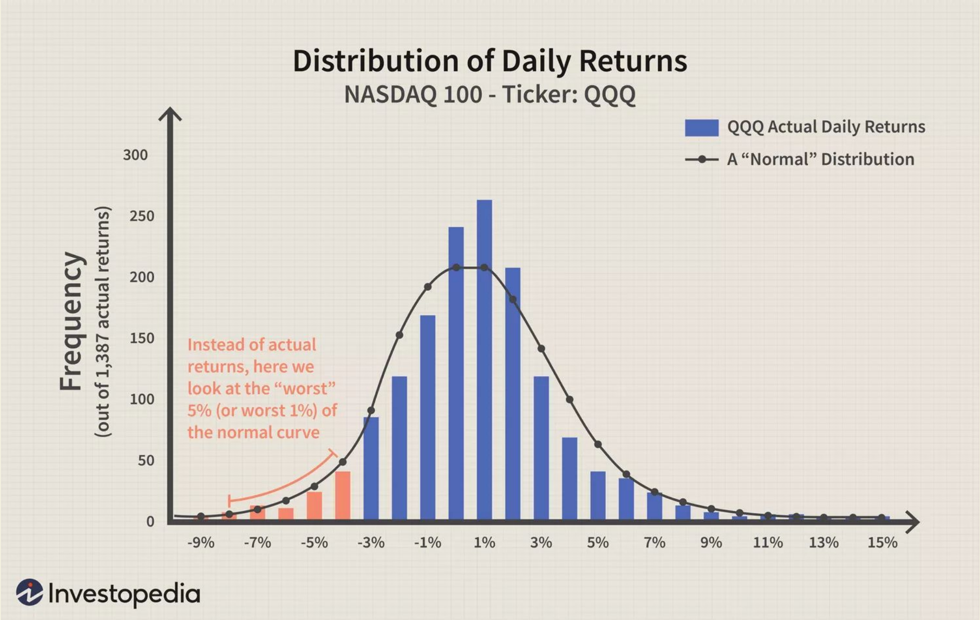
Theta, Θ: sensitivity to time

1st order approximation

$$\Delta = \frac{PV(x_1 + \Delta x_1, \ldots) - PV(x_1, x_2, \ldots)}{\Delta x_1}$$

Market risk example : Value at Risk (VaR)

What is VaR?
The maximum loss expected over a given time horizon at a given confidence level



Example: Given a $1,000,000 portfolio, a **99% 1-day VaR of $7,000** means:

There's 99% chance the portfolio will not lose more than $7,000 over 1 day

⚠️Note: It does not mean you can't lose more than $7,000. It simply means that, based on historical data and model assumptions, losses worse than that should only happen about 1% of the time.

Historical VaR: Full lookback of historical data ->Millions of rows for one VaR run

Montecarlo: 10k similuations per position per risk factor per day using large covariance matrices (e.g. 500x500)

Scenario	Methodology	Complexity	Time
Full MonteCarlo	Reprices every instrument in every scenario	Very high	Hours
Historical simulation	Historical data	Medium	Minutes - hours
Delta-gamma approximation	Taylor expansion	Low	Seconds-Minutes

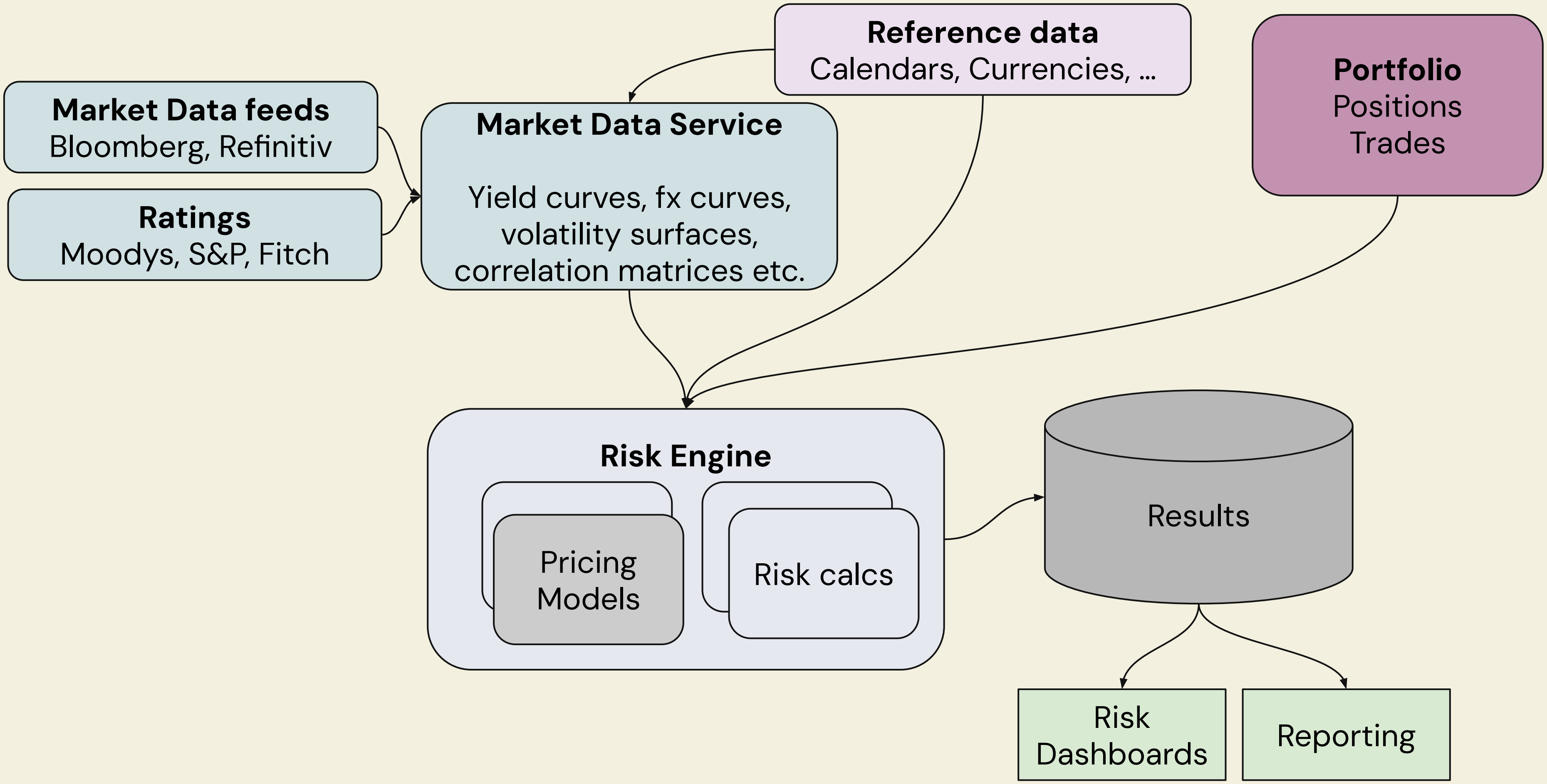
How have financial risk management systems evolved?

End of day
overnight batch processing and T+1 reporting, where risk managers used reports showing yesterday's exposures

Intraday
hourly updates and incremental processing in a batch-oriented architecture

Real time
Event streaming and machine learning models

- detect patterns humans can't see,
- provide actionable insights



Reference data
Calendars, Currencies, …
Portfolio
Positions
Trades
Market Data feeds
Bloomberg, Refinitiv
Ratings
Moodys, S&P, Fitch
Market Data Service
Yield curves, fx curves, volatility surfaces, correlation matrices etc.
Risk Engine
Pricing Models
Risk calcs
Results
Risk Dashboards
Reporting

Why so difficult to move from daily to intra-day?

Challenge 1: Data integrations

- Market Data dependencies
- Complex data integrations

Challenge 3: Reporting

- Storage and access of data
- Visualisation & insight tools

Challenge 2: Compute

- Expensive calculations
- Scalability & Resilience

Challenge 1: Data integrations

What is it?

- combines messaging queues and log storage
  => versatile tool for real-time data flows
- *defacto technology for streaming and event sourcing*

Distributed: Massive scalability & throughput - process million events per second across clustered servers

High Durability: Persist to disk, replicate across cluster, prevent data loss

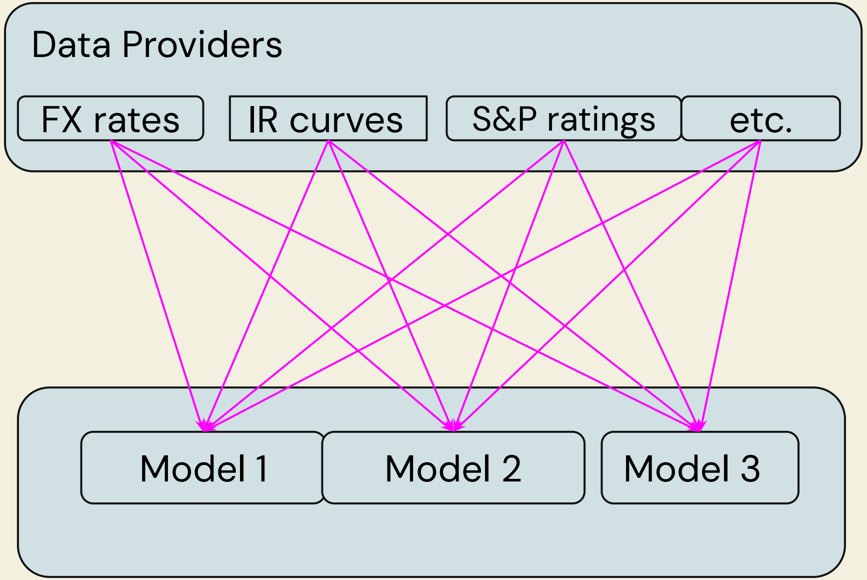
Replayability: consumers “rewind” and replay events

Decoupled: separates producers and consumers, asynchronous communication

Data integration at scale

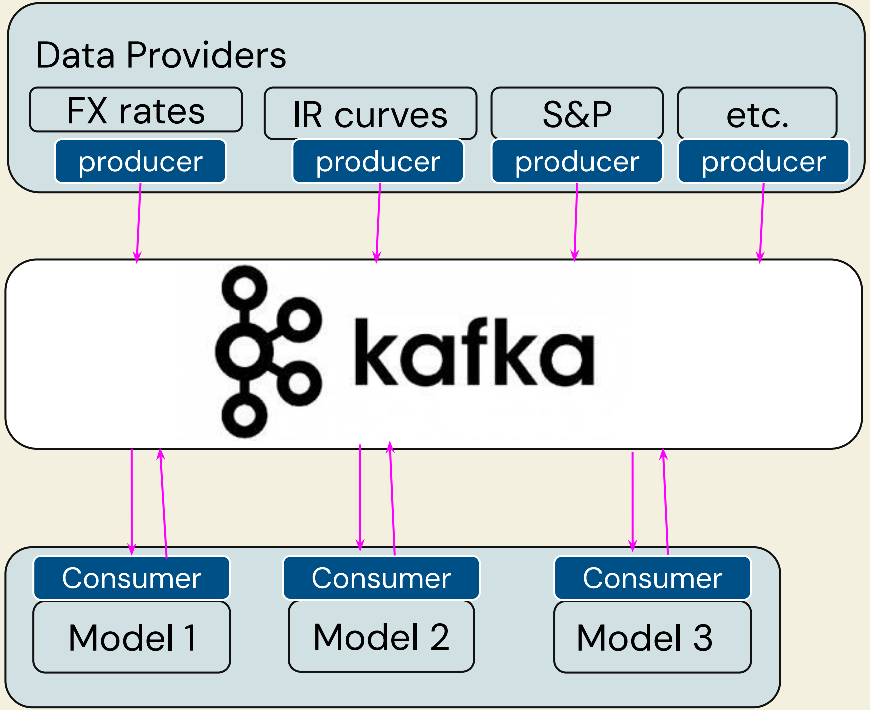
BEFORE

Point to point integrations
->O(N^2) integrations



AFTER

LinkedIn 2011



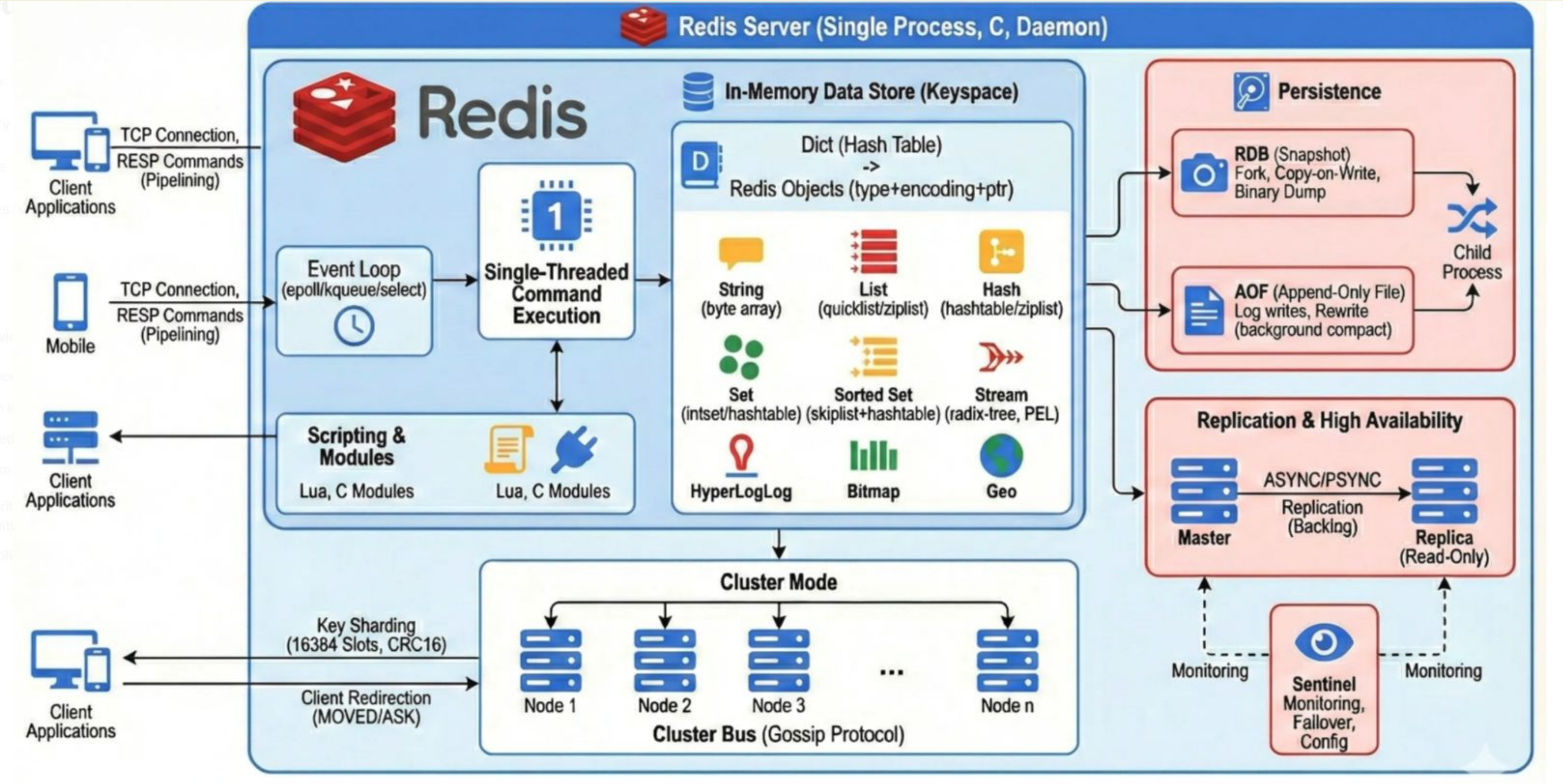
Challenge 1:
Data integration – Caching

Redis

A lightning-fast open-source and in-memory Data Structure Store

A NoSQL key-value database, cache, and message broker

Source: https://ashutoshkumars1ngh.medium.com/



Challenge 2: Compute

Pre 2005
On prem systems
Excel/Spreadsheets

2005 – 2008
Grid computing
Trade books expanded
Models increasingly complex
Infrastructure fragile

2008
- Banks couldn't aggregate exposure to Lehman
- VaR models failed
- Regulators not get timely data

2010 – Basel III
Banks must be able to aggregate risk exposures within hours NOT days

Scalability

Vertical Scaling
- More CPU
- More memory
- Faster process

Horizontal Scaling
- Add more servers

Google (2004) : Map Reduce

Perform parallel and distributed processing on huge dataset

Map:
- Splits input data
- Processes in parallel
- Output key-value pairs

Reduce:
- Aggregate results into final output

hadoop HDFS

open-source framework processes big data efficiently on commodity server cluster

But
- Slow disk I/O between stages, (data written to HDFS)
- No support for real-time or iterative tasks
- Verbose Java code with complex job chaining

fast, open-source, distributed analytics engine for large-scale data processing.

2014

- **In-memory:** Minimizes disk usage, store intermediate results in RAM
- **High-level API:** Python, Scala, SQL, Java
- **Resilient Distributed Datasets (RDDs):** Fault tolerant

Performance 5 bn calculations 10k batch jobs	**Traditional Grid Computing** (at least 4 hours)	**Apache Spark Cluster** (e.g. less than 15 mins)
Data I/O	Each task independently pulls market data from external DB and writes back results	**In memory :** All static portfolio and market data broadcast and cached across workers
Scheduler	High overhead, queue latency	Low overhead, DAG scheduler routes tasks to data threads
Inter-task data sharing	Not possible	Native: maps dependencies directly across the cluster

Distributed systems
- Network failures
- Service failures

How to design for maximum resilience, fault tolerance, and high availability?

- open-source container orchestration platform designed to automate deploying, scaling and managing containerized applications.
- It provides a framework to run distributed systems resiliently.
- Declarative approach: users define the desired state of their applications and infrastructure, and Kubernetes continuously works to maintain that state.

You name them and when they get sick, you nurse them back to health. Now servers are treated like cattle. You number them and when they get sick, you shoot them.
Bill Baker, Microsoft

Key to creating a highly available system with
- reduced failures
- smaller blast radius
- faster disaster recovery

Challenge 3: Reporting

Columnar databases

Designed for analytical workloads and reporting big data applications

Relational databases E.g. Oracle, SQL Server

- Row based, not possible to access individual columns
- Compression limited because different data types
- Contention of multiple threads access same data page

Columnar Databases E.g. BigQuery, Aws Redshift

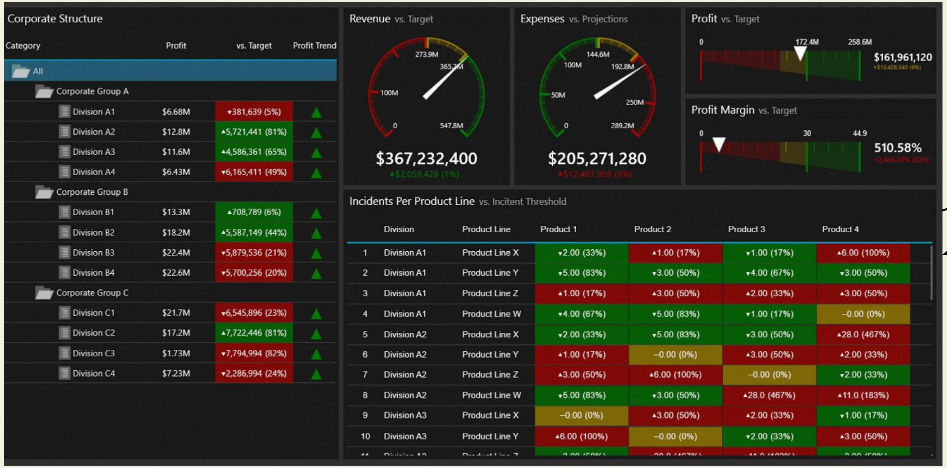
- Retrieve only the specific columns of data needed for a query => faster query performance.
- apply compression methods to decrease the storage capacity needed and enhance its efficiency

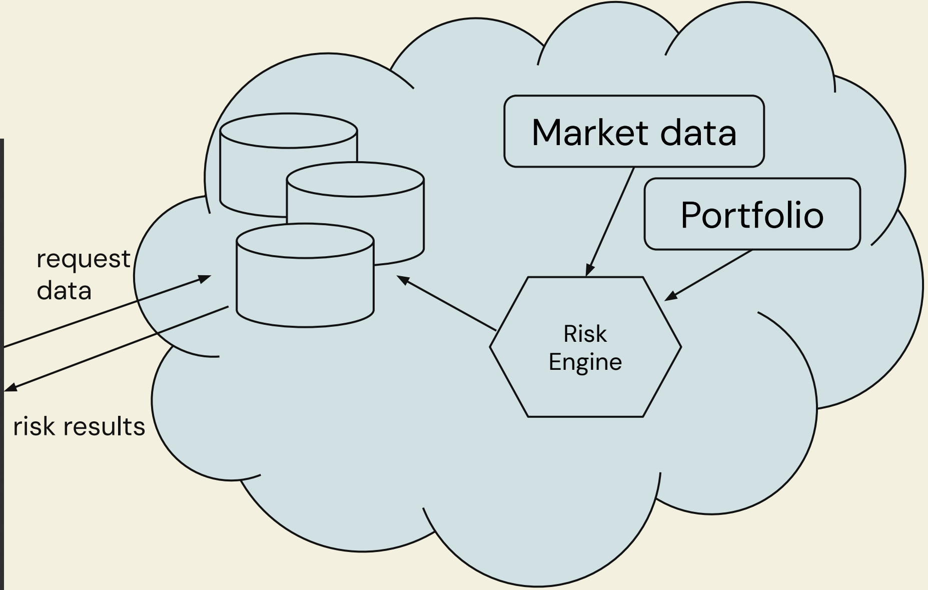
Trade Id	Ccy	Sim1	Sim2	...
t1	USD	140034	146742	
t2	EUR	32535	31967	
t3	EUR	-1265	-2931	
...				

Data stored:
t1,t2,t3... USD,EUR,EUR...140034,32535,-1265,...

Compression Techniques:

- **Dictionary encoding**
- **Run-length encoding**: 1000 consecutive EUR can be stored as EUR1000
- **Bit packing:** E.g. boolean value can be stored as single bit not byte
- **Delta encoding:** Useful for sequential data like dates





Delivering interactive, real-time risk reporting requires **massively parallel processing** and **column-oriented** database architectures to handle high-velocity analytical **READ** queries

2019 Fundamental Review of the Trading Book (FRTB)

- Capital requirements are calculated using Expected Shortfall (ES)
- VaR backtesting

=> Forced real-time adoption

Present

- Real-time reporting
- Intraday VaR
- Live Stress Testing
- Instant WhatIf scenarios

Future

- Streaming architecture
- Machine learning risk factor models
- LLM assisted tooling

Risk Management
Identify, Measure, Mitigate

- Credit Risk
- Market Risk
- Operational Risk
- Liquidity Risk

Data Engineering
Ingest, Transform, Serve

- Data pipelines
- Real time Streaming
- Caching
- Databases & Storage

Cloud Platforms
Provision, Scale, Observe

- AWS/GCP/Azure
- Autoscaling
- Distributed systems
- Kubernetes

FinTech
Automate, Innovate, Drive

- Real-time Pricing
- Reg Tech
- Insuretech
- AI adoption

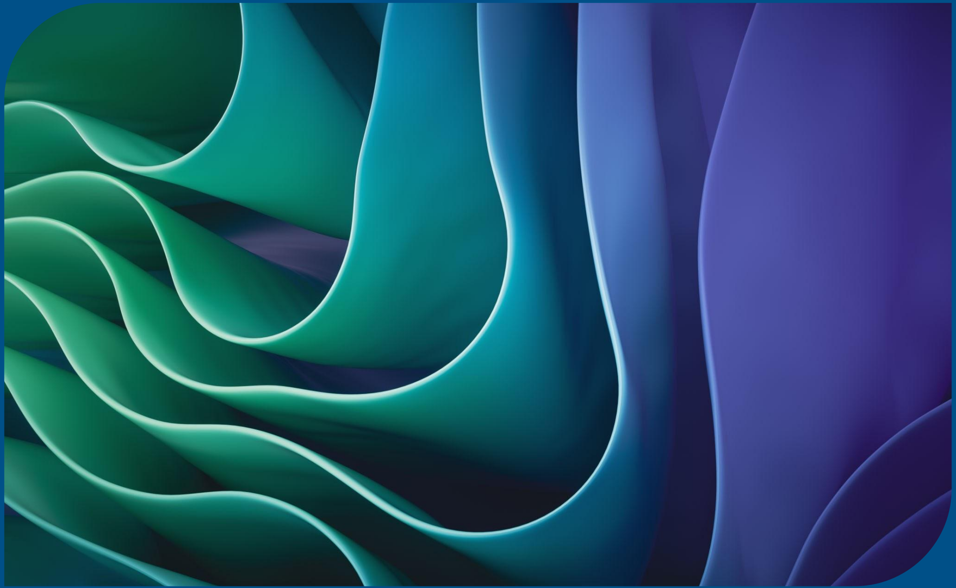
Thank you

Questions ?

Contact details:

www.linkedin.com/in/sionedbaker

Extra



FinTech - **financial technology** to improve, automate, and innovate the delivery of financial service by efficiently managing operations through specialized software, algorithms and engineering technology

1866 – 1967
Infrastructure

1967 – 2008
Digitisation

2008 – 2020
Democratisation

2020 onwards
Scale & AI

1.0
First transatlantic cable(1866)
Fedwire funds transfer service(1918)
First credit cards in 1950s

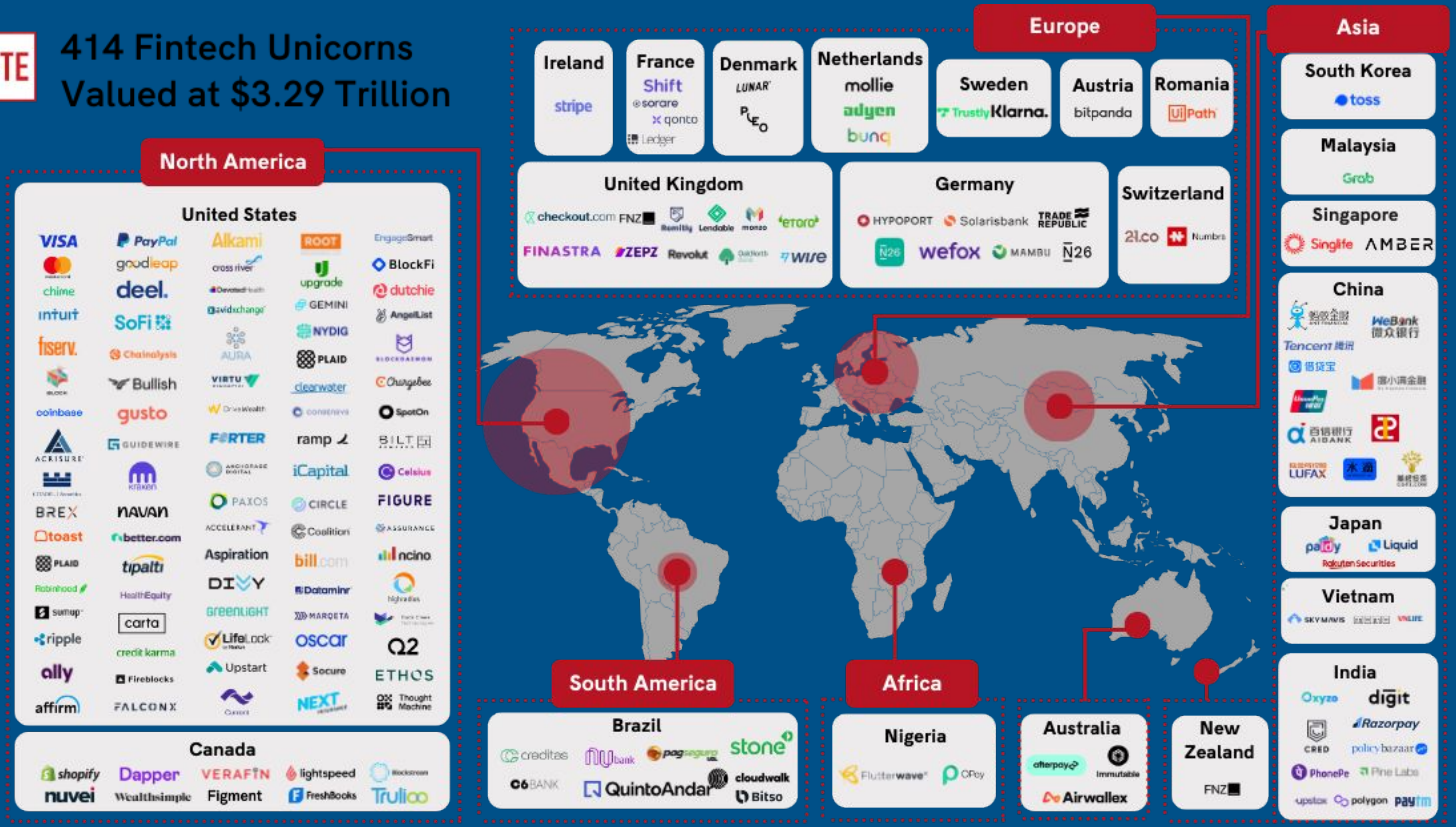
2.0
First Barclays ATM (1967)
NASDAQ (1971),
SWIFT communications (1973)
Online banking and PayPal in the 1990s

3.0
Challenger neobanks
Bitcoin (2009),
P2P lending
"Open Banking" frameworks.

4.0
Accelerated by the pandemic
Massive digital platforms,
Heavy AI integration

CFTE

414 Fintech Unicorns Valued at $3.29 Trillion

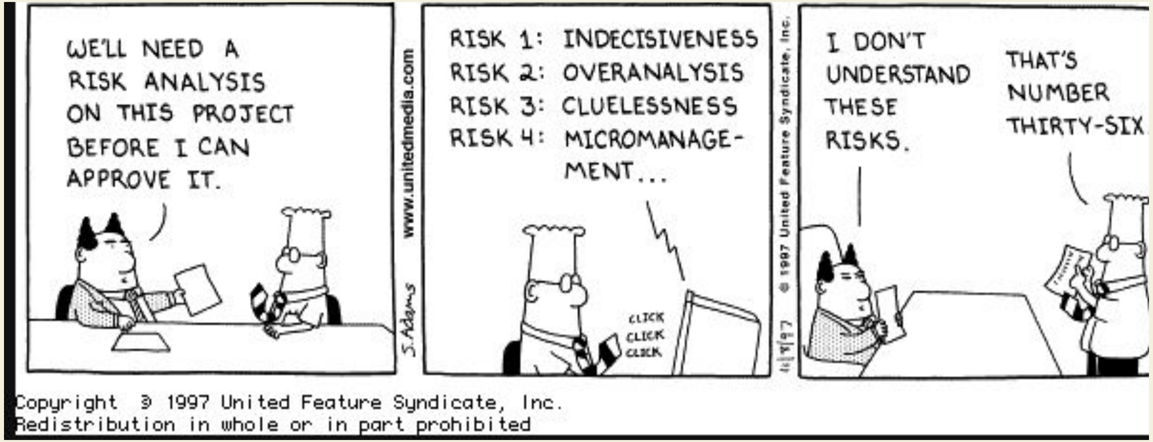


Source: https://courses.cfte.education/ranking-of-largest-fintech-companies/

- *Faster operations*
- *Democratisation*
- *Automation*
- *Innovation*

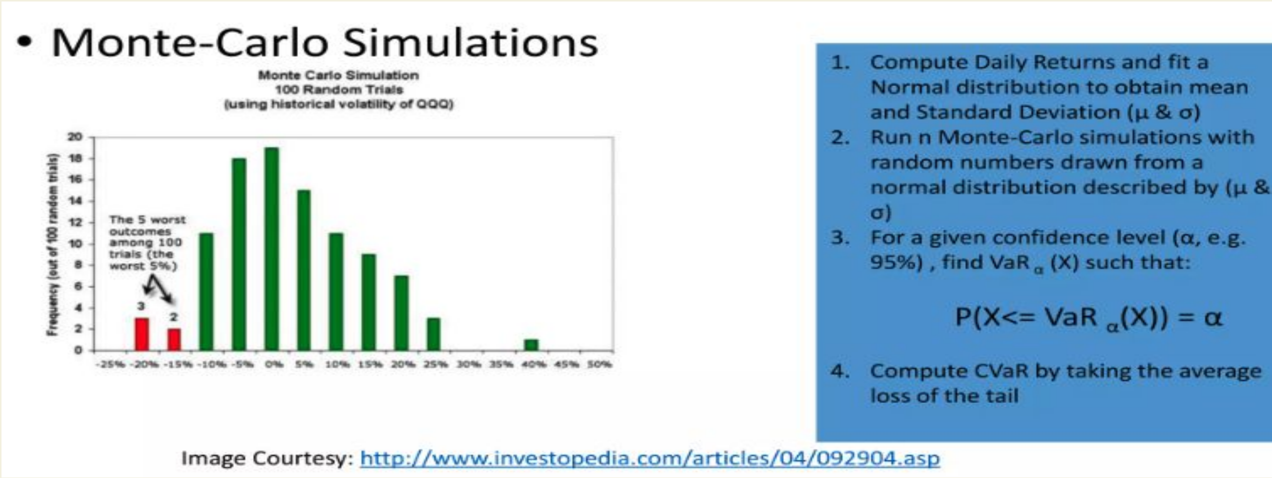
But what about the risks?

As financial technology evolves, risk management systems also need to evolve and adapt

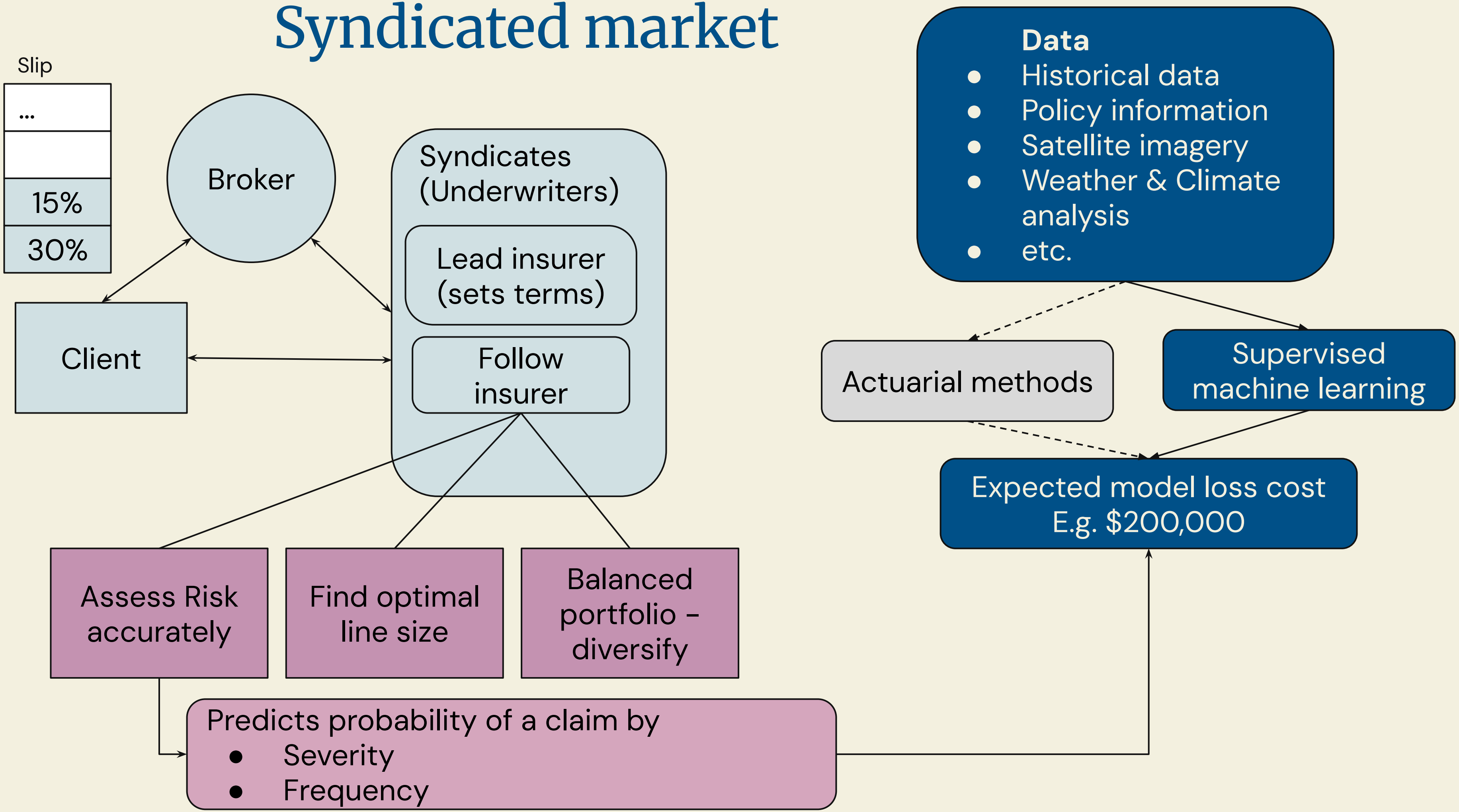

WE'LL NEED A RISK ANALYSIS ON THIS PROJECT BEFORE I CAN APPROVE IT.
www.unitedmedia.com
S. Adams
RISK 1: INDECISIVENESS
RISK 2: OVERANALYSIS
RISK 3: CLUELESSNESS
RISK 4: MICROMANAGE-MENT...
CLICK CLICK CLICK
© 1997 United Feature Syndicate, Inc.
I DON'T UNDERSTAND THESE RISKS.
THAT'S NUMBER THIRTY-SIX.
Copyright © 1997 United Feature Syndicate, Inc.
Redistribution in whole or in part prohibited

Historical Method
looks at one's prior returns history over past year and orders them from worst losses to greatest gains

Montecarlo Method
simulate projected returns over hundreds or thousands of possible iterations



Syndicated market



Data Engineering Issues

1 Data Volume

2 Data lineage & Consistency
E.g. position booked at 4pm, price snapped at 3:58pm -> incorrect PV

3 Data completeness
Trades booked across multiple system
Amendments, cancellations, duplicates

4 Missing Market data and proxies
Missing price - interpolatation
New instrument - map to proxy
Corporate action (stock split) - adjust historical prices

5 Correlation Problem
500 Montecarlo risk factors -> 125000 covariance matrix
Correlations non-stationary, everything correlates to 1 in crisis

6 Reproducibility & Auditability

- Point in time snapshorts
- Version control on parameters
- Immutable audit trail
- Ability to re-run historical VaR wih yesterday's model and yesterday's data

7 Backtesting
VaR must be backtested daily,
Compare yesterday's VaR against actual P&L

If actual loss > VaR on more than 3 days/year
=> Regulatory capital addon
=> Models need recalibrating

Market Risk example

UK manufacturer needs to buy steel, alumunium and copper from suppliers in middle east. Exposed to

- *Commodity price risk*
- *FX risk*

Hedging strategies

1. Commodity future
2. OTC forward contract (FX & Commodity)
3. Options

Manufacturer

Buy a call option on 500 tonnes of copper that pays out if price rise above the strike, but keep upside if prices fall

Strike: $9000/t,

Premium: $200/t

If copper $11000/t : option pays out $2000/tand limits loss

If copper $7000/t : option expires, firm buys cheap at spot price

Bank

If copper $13000/t, the bank owes $4000/t !!!

It has **uncapped downside**, so the bank must immediately **delta hedge**

E.g. if delta = 0.4, bank will buy 0.4x500=200t of copper futures

So, if copper rises $1000,

- Option liability rises $400/t
- but futures gains $400/t

-> Net P&L=0

delta =measures how much the option price move per $1 move in underlying (copper price)

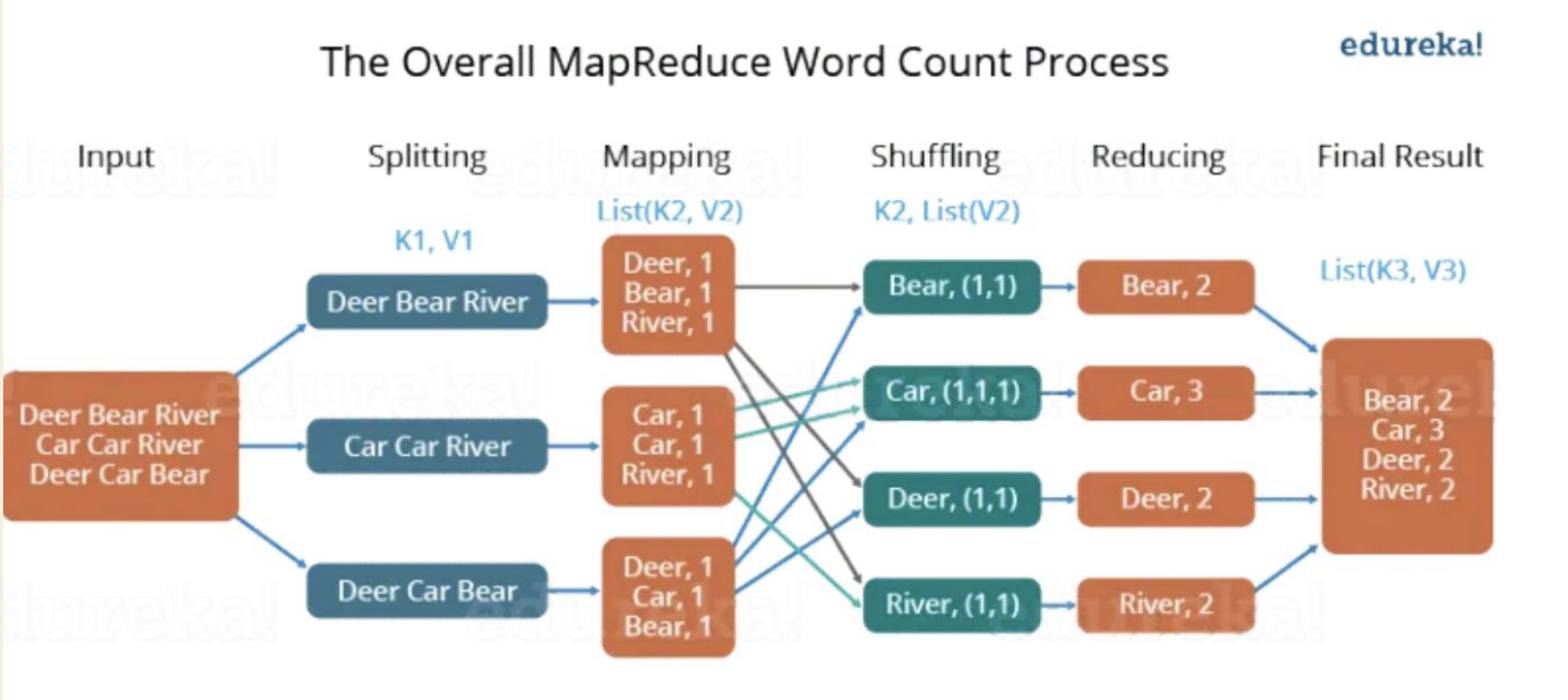
However delta is NOT fixed – it changes as the price moves – so the bank is continuously rebalancing E.g.

Copper price	Delta	Strategy
$8500	0.4	Hold 200 futures
$9000	0.55	Buy more, hold 275 futures
$9800	0.72	Buy more hold 360 futures
$7500	0.18	Sell, hold just 90 futures

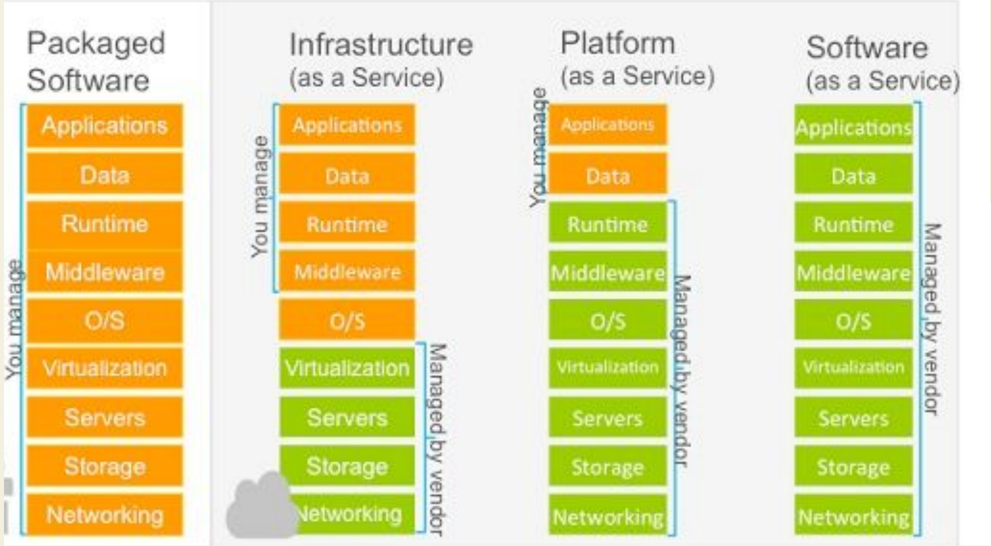
In practice bank doesn’t hedge each option but runs commodity options book, one of many books it must manage.

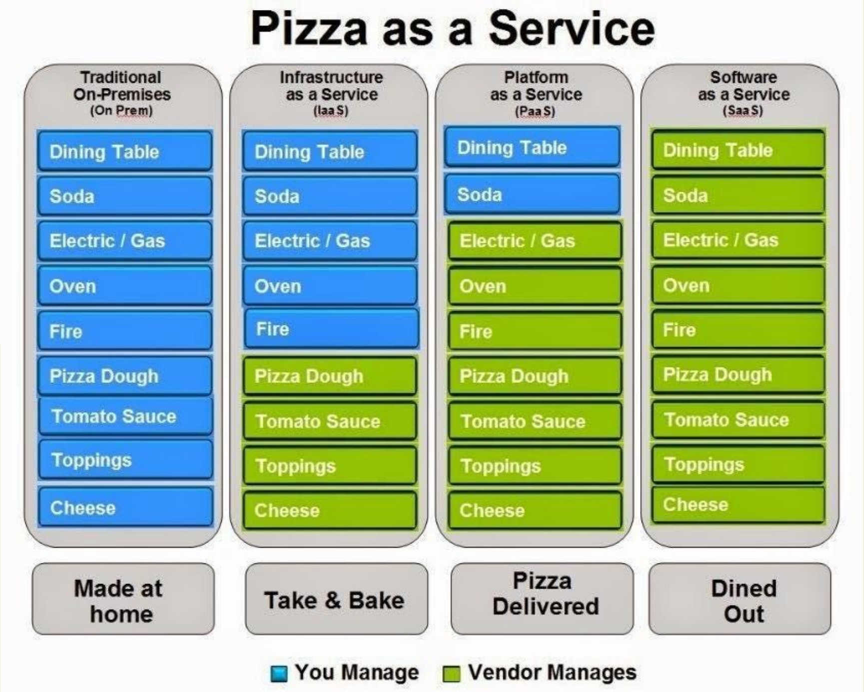
Example of MapReduce

sourceE: https://medium.com/edureka/mapreduce-tutorial-3d9535ddbe7c



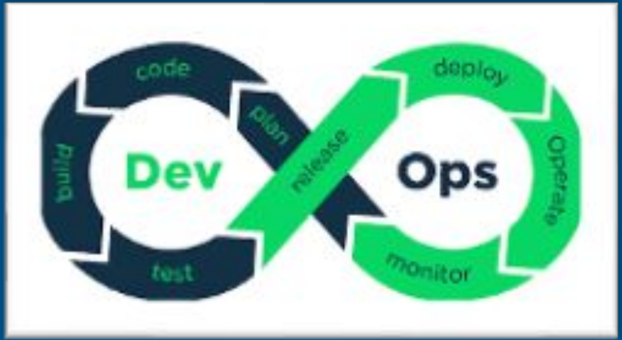
Cloud Service Models

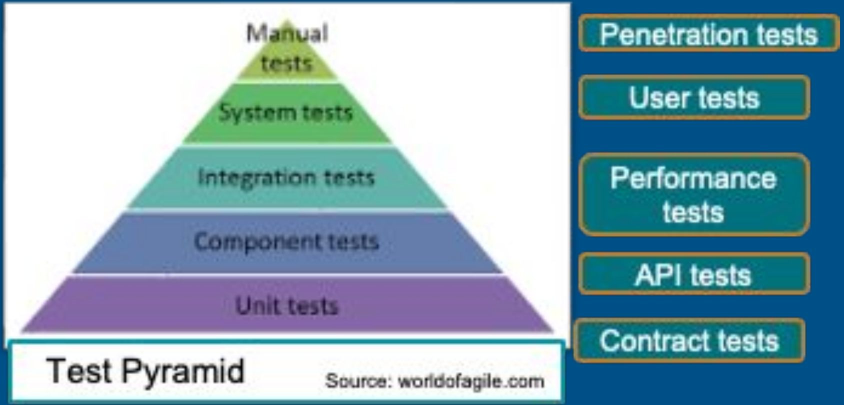




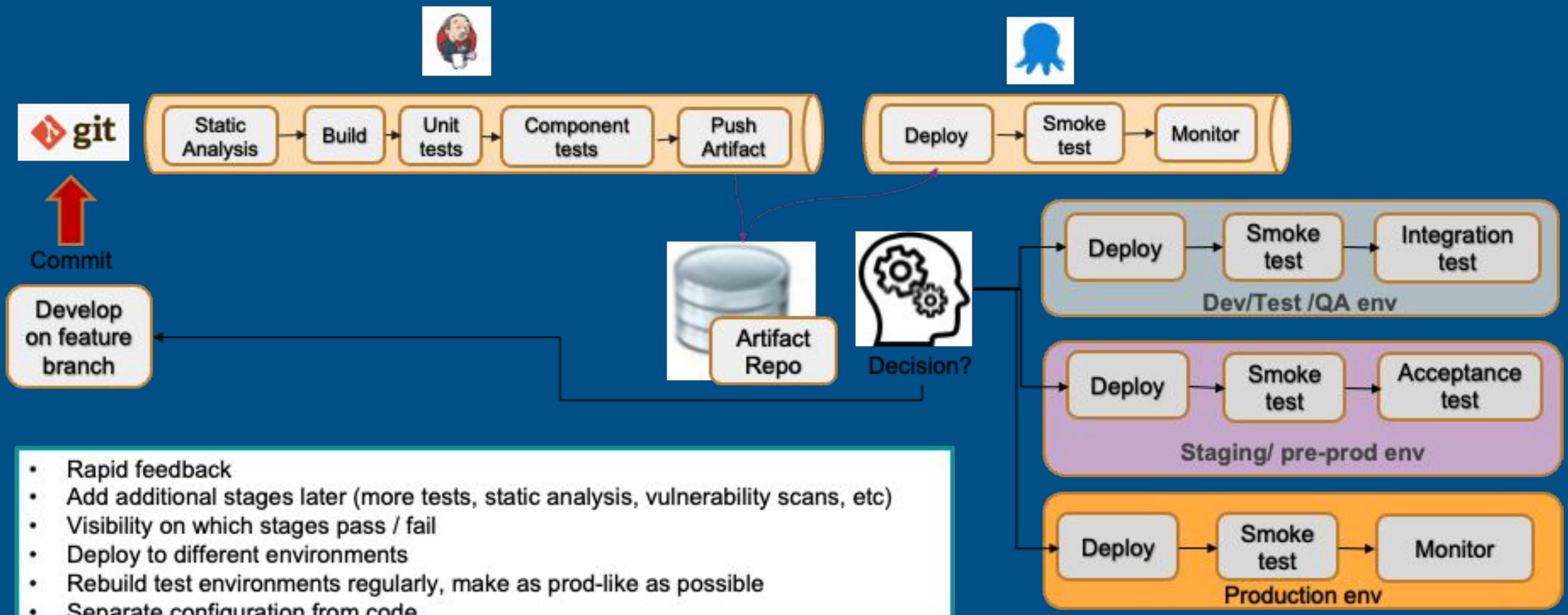
Improve Software & Data delivery

- Reduce and eliminate manual steps
- Make test environments as prod like as possible
- Increase visibility and awareness
- Feedback loop
- Make incremental change to prove repeatability and stability
- Focused testing





CICD – Continuous Integration Continuous Delivery



- Rapid feedback
- Add additional stages later (more tests, static analysis, vulnerability scans, etc)
- Visibility on which stages pass / fail
- Deploy to different environments
- Rebuild test environments regularly, make as prod-like as possible
- Separate configuration from code

	With Redis (High Performance)	**Without Redis (Standard)**
Data Access Speed	Sub-millisecond < 1ms	Milliseconds to seconds
Use cases	Caching, real-time dashboards, high-concurrency user sessions	Only reads and logging
Data Storage	In-Memory (RAM)	Disk-based (HDD/SSD)
Complexity	High (Manages cache eviction, consistency)	Low (Simpler architecture)
Scalability	High horizontal scaling (Redis Cluster)	Vertical scaling (bigger database server)

Future Data Engineering Challenges

- Data delivery & timeliness
- Data accuracy & quality
- Data processing reliability
- Data reporting explainability

Kafka