AQUAQ Analytics

Interfacing kdb+ with Refinitiv Tick History in the Cloud with Google BigQuery

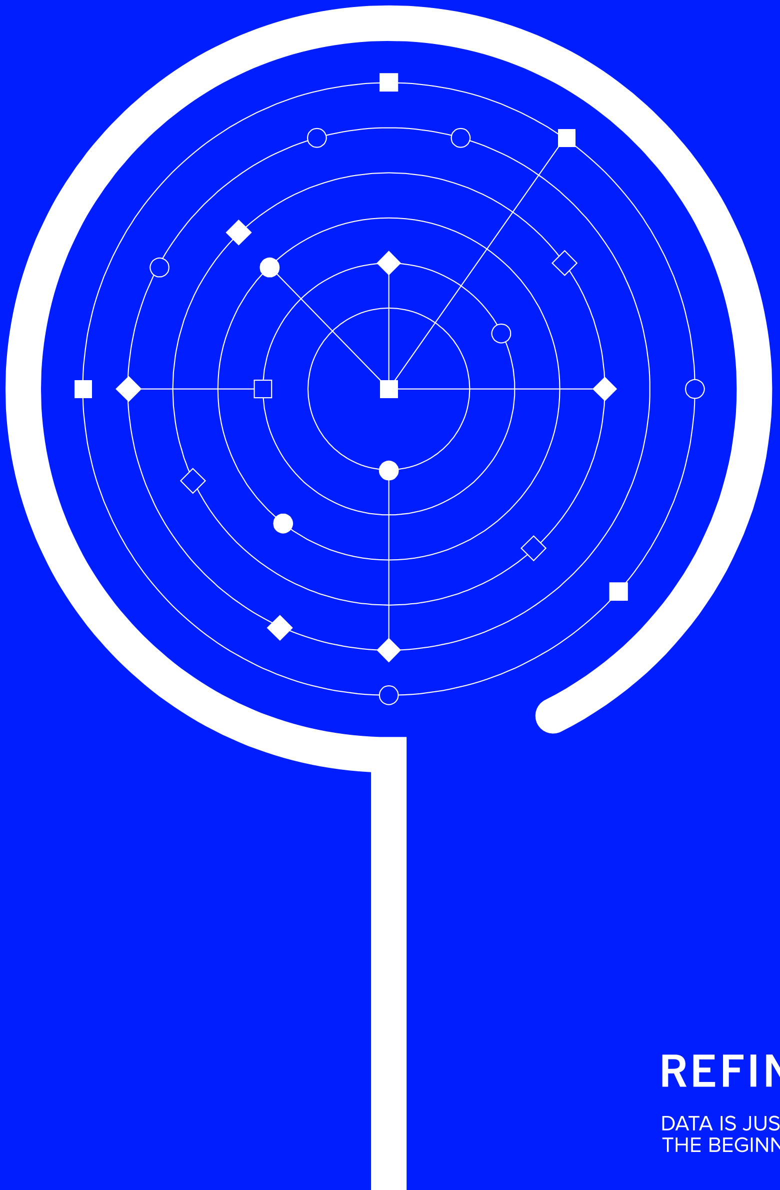
REFINITIV™
DATA IS JUST
THE BEGINNING

AQUAQ Analytics

Refinitiv Tick History in the Cloud with Google BigQuery

Query in the cloud – analyze in the cloud

April 2021



REFINITIV™

DATA IS JUST
THE BEGINNING

For every $1 spent on financial market data a further $8 is spent on processing, storage, transforming that data before it can be analysed

Refinitiv Report, 2019

Refinitiv Tick History Compute in the Cloud

Refinitiv has changed the paradigm: bringing the compute to the data rather than the data to the compute **< utility model >**

You can now query and analyze **decades** or more of **Venue by Day** data directly within **Refinitiv BigQuery** with **zero** spin up time without the cost or delay of a hard media cut - the available historic depth profoundly outperforms the rolling 30 days in traditional Venue by Day offerings and there is simply no need to download the content or wait for it to be delivered on storage devices – **query in the cloud – analyze in the cloud**.

Powerful analytical tools with full SQL Windowing

Fully Encrypted data at rest & in transit

Artificial intelligence and machine learning ready

Open source via GCP or API with Refinitiv RDP & Datascope

Serverless Design – use when needed

Cut the cord to the datacenter – query in the cloud

Refinitiv Tick History Compute in the Cloud with Big Query

Tick History Compute in the Cloud – Features

Tick compute in the Cloud provides you with the full set of historical Tick Data to be accessed with best-in-class performance query tools in seconds

*6PB+ / 24 Years

Tick History Archive with 500 Global venues & party contributors complete with Level 1 and 2 data with 24 years of trading from 1996

*4TB added

Data added daily, stored in RIC format

Microsecond power

2 Microsecond aggregation to the entire 24year dataset using SQL Windowing (0.000002 seconds)

Interface independent

Run any interface to query, compute and review / extract output (Python, R, SQL etc)

*Cloud independent

Use any cloud service to query & extract output (GCP, AWS, AZURE, Snowflake)

Export

Export to visualization tools – DataStudio, Neo4J & file formats XLS, Tableau, JSON, flat files etc.

Zero Latency design

Run queries & see results in the cloud with minimal latent hops

Time series partitioning

Time series partitioning with in–memory analytics for super fast returns with SQL Windowing

Schedule, Add, Merge

Schedule, Save, share, add your own data – merge from several sources – all in the cloud (no need to cut down data)

*Multiple Queries

Run multiple queries on the same set of data concurrently with no impact to performance (no job queuing)

API

Full suite of API's to query or port data into any engine / endpoint in common file formats

High Performance

Run queries over millions of rows of data in seconds (using GCP Big Query)

Serverless design

Storage maintained by Refinitiv (with Google Cloud Platform – Big Query)

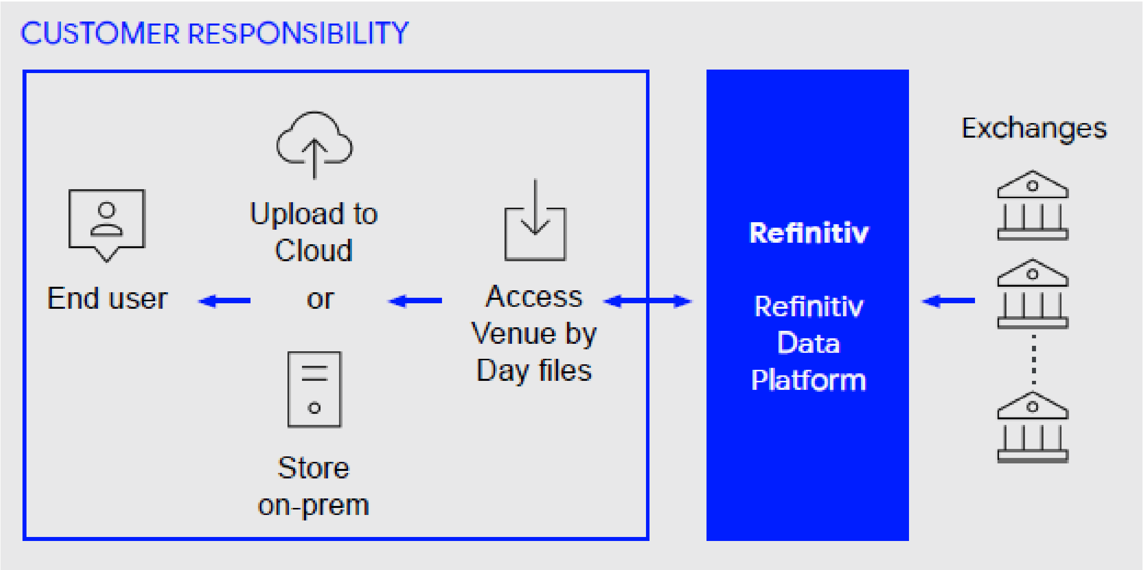
Access

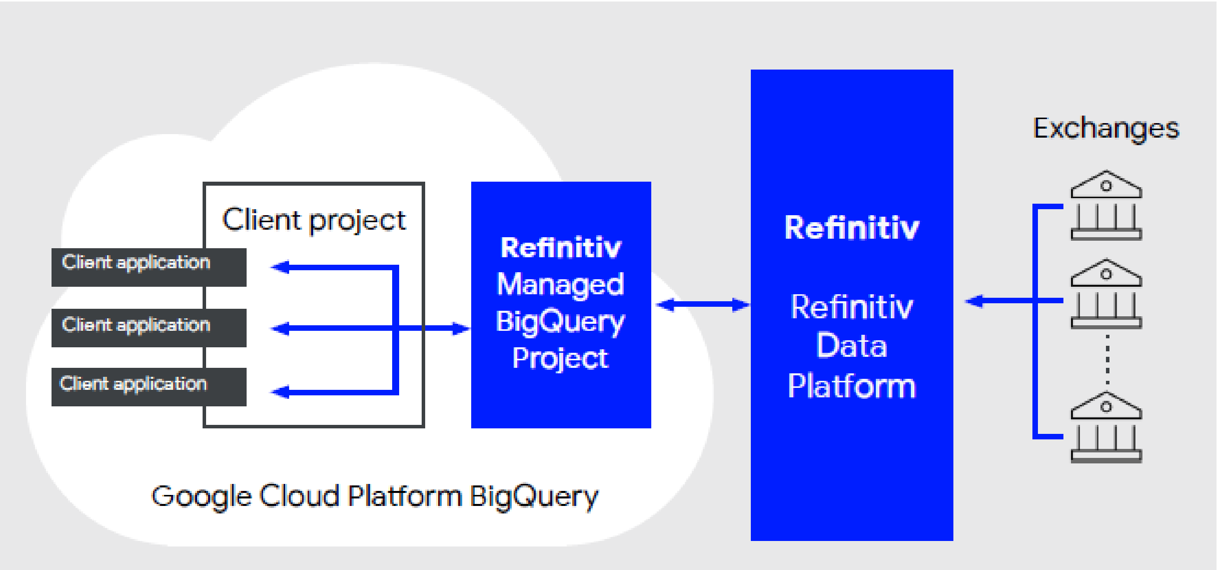
Multiple access for several uses/nodes with no loss of performance – create instances in cloud

Encryption

Full end to end encryption on all data at rest and in transit

Extract, Transform, Load – Bringing Analytics to Data not Data to Analytics





The ETL (Extract, Transform and Load) process is the customer's responsibility

Query Ready Data where the capability is

REFINITIV™
DATA IS JUST THE BEGINNING

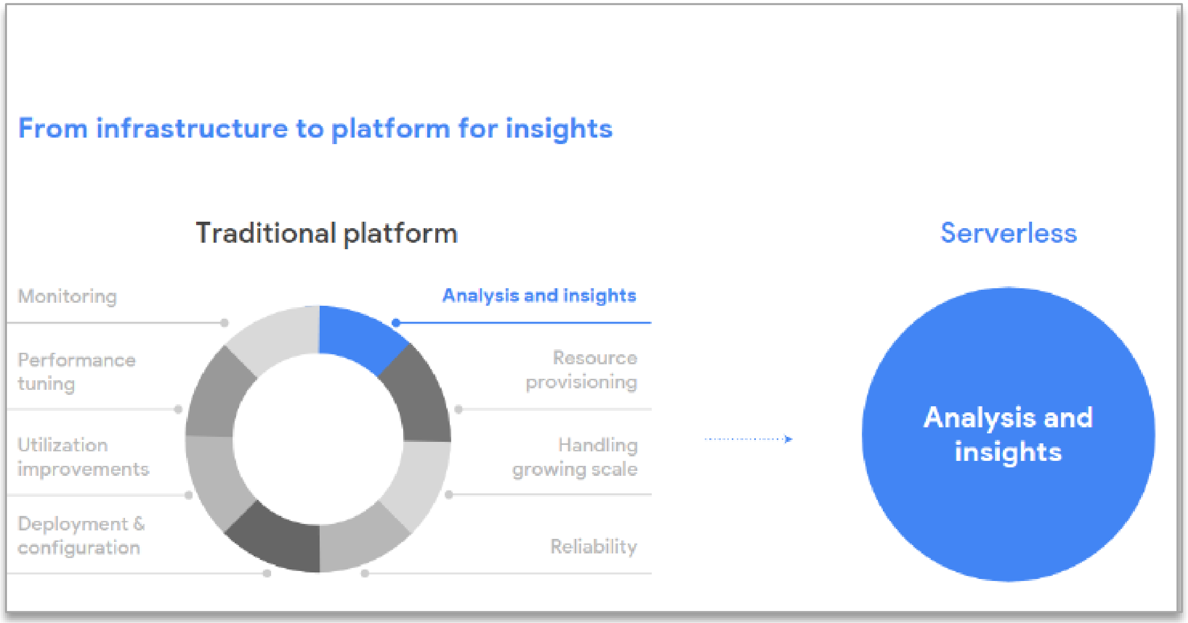
Tick ‘Compute’ in the Cloud – Performance & Price Results

Algorithm / Test	Description	Result	Est Compute Cost*
VWAP	VWAP = (Cumulative (Price * Volume)) / (Cumulative Volume) on NYSE, **one day** of tick by tick data (microsecond updates) - **165 million rows** computated in under	**1.7 seconds**	**$0.00072**
OHLC (bar)	Highest price, Lowest price, Open and Close interrogated all of the NYSE exchange for **15 days** at **75 second** intervals - **4 million+** rows of data computated	**27 seconds**	**$0.02**
Latency / Performance (VWAP / TWAP)	Desktop in USA query Tick in EU – query **150 million** rows of Tick – run two algorithm concurrently on same set of data – display results on iPhone in US over LTE	**2.7 seconds to iPhone displaying 50,000+ rows of data**	**$0.0014**
Local Usage	Two algorithms running concurrently, 3 year date range on **155 million rows** in the EU porting **5000 rows** of results sent to local Microsoft Excel instance in the US	**5 seconds**	**$0.00062**
Intraday Summaries	Intraday summaries on **LSE** at **10 second** intervals – **single day** of data – (Trades & Bid/Ask Quotes (difference)) + Volume – **10 GB** of data analyzed and ordered in sequence **1.8 million** records produced	**23 seconds**	**$0.02**
	Intraday summaries on **LSE** at **10 second** intervals – **one month** of data – (Trades & Bid/Ask Quotes (difference)) + Volume – **255 GB** of data analyzed and ordered in sequence **4.75 million** records produced	**59 seconds**	**$0.05**
	Intraday summaries on **LSE** at **10 second** intervals – **one year** of data – (Trades & Bid/Ask Quotes (difference)) + Volume – **2 TB** of data analyzed and ordered in sequence **475 million** records produced	**2 minutes 07 seconds**	**$10.00**

* charged once in given 24-hour window regardless of the amount of queries run on same data range

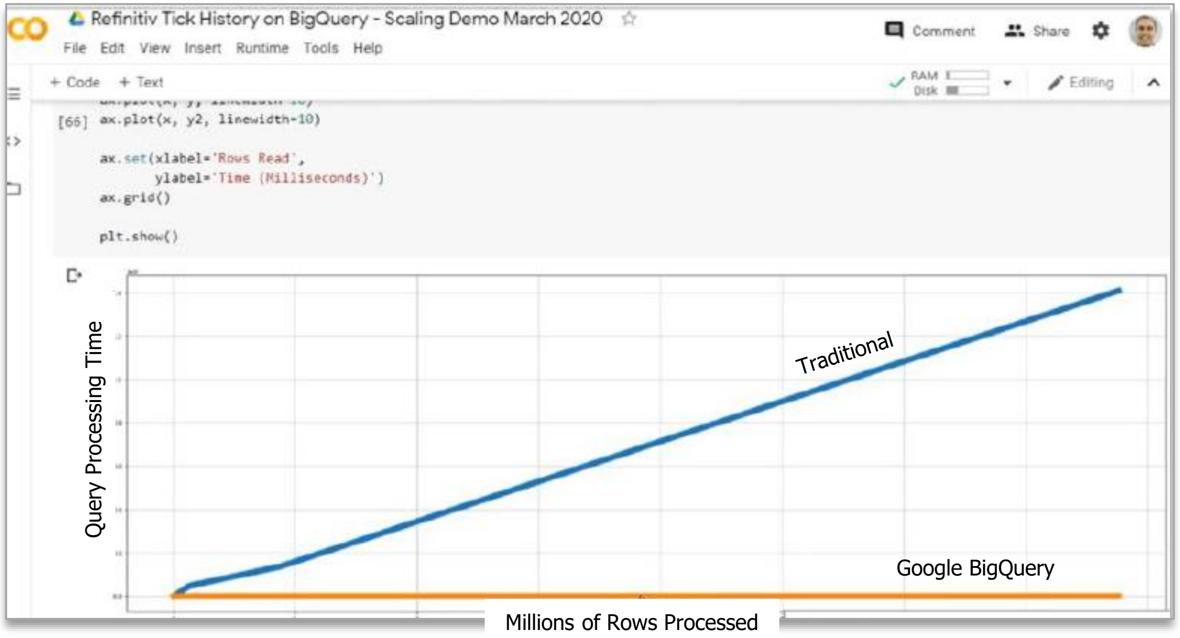
Tick History with the Compute Power of Google BigQuery

Spend More Time on Analysis & Insight and Less on Data Management Using Serverless Data Analytics



From infrastructure to platform for insights

Introduce Sublinear Scaling and Compute Power to your Queries of Tick History data



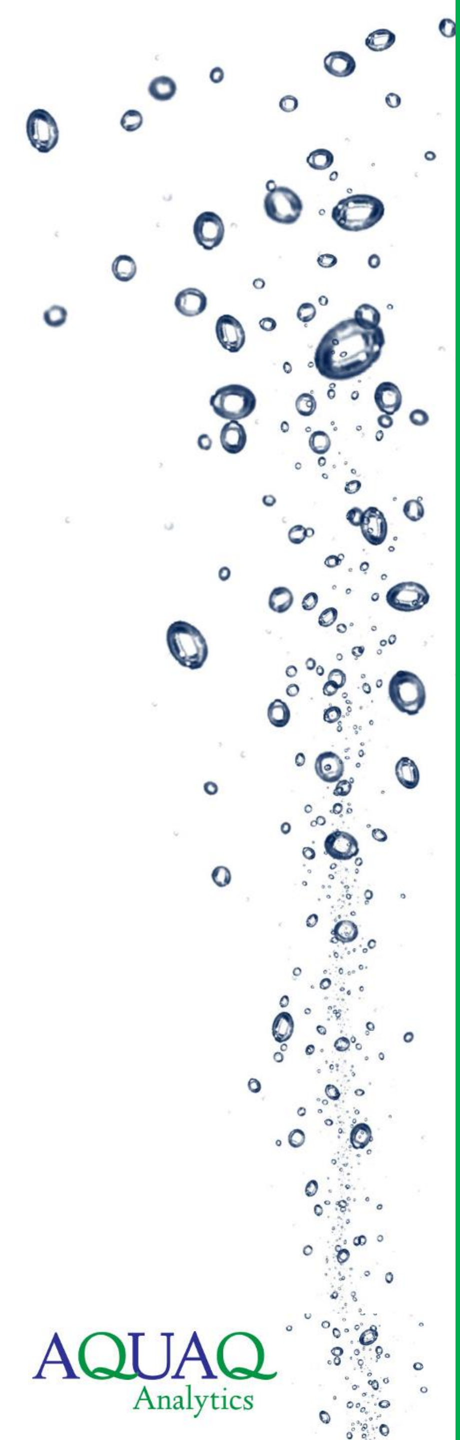
Refinitiv Tick History on BigQuery - Scaling Demo March 2020

```
[66] ax.plot(x, y2, linewidth=10)

     ax.set(xlabel='Rows Read',
            ylabel='Time (Milliseconds)')
     ax.grid()

     plt.show()
```

REFINITIV™
DATA IS JUST THE BEGINNING



Interfacing BigQuery with kdb+

Query in q – analyze “wherever”

Why?

Why not store Everything in kdb+ on-premise?

- Internal storage costs are high
- Clients delete/downsample data they would rather keep

Why not store everything in a cloud data warehouse?

- Lose flexibility and performance of kdb+
- Lacks real-time kdb+ capabilities
- Potentially good for older historical data

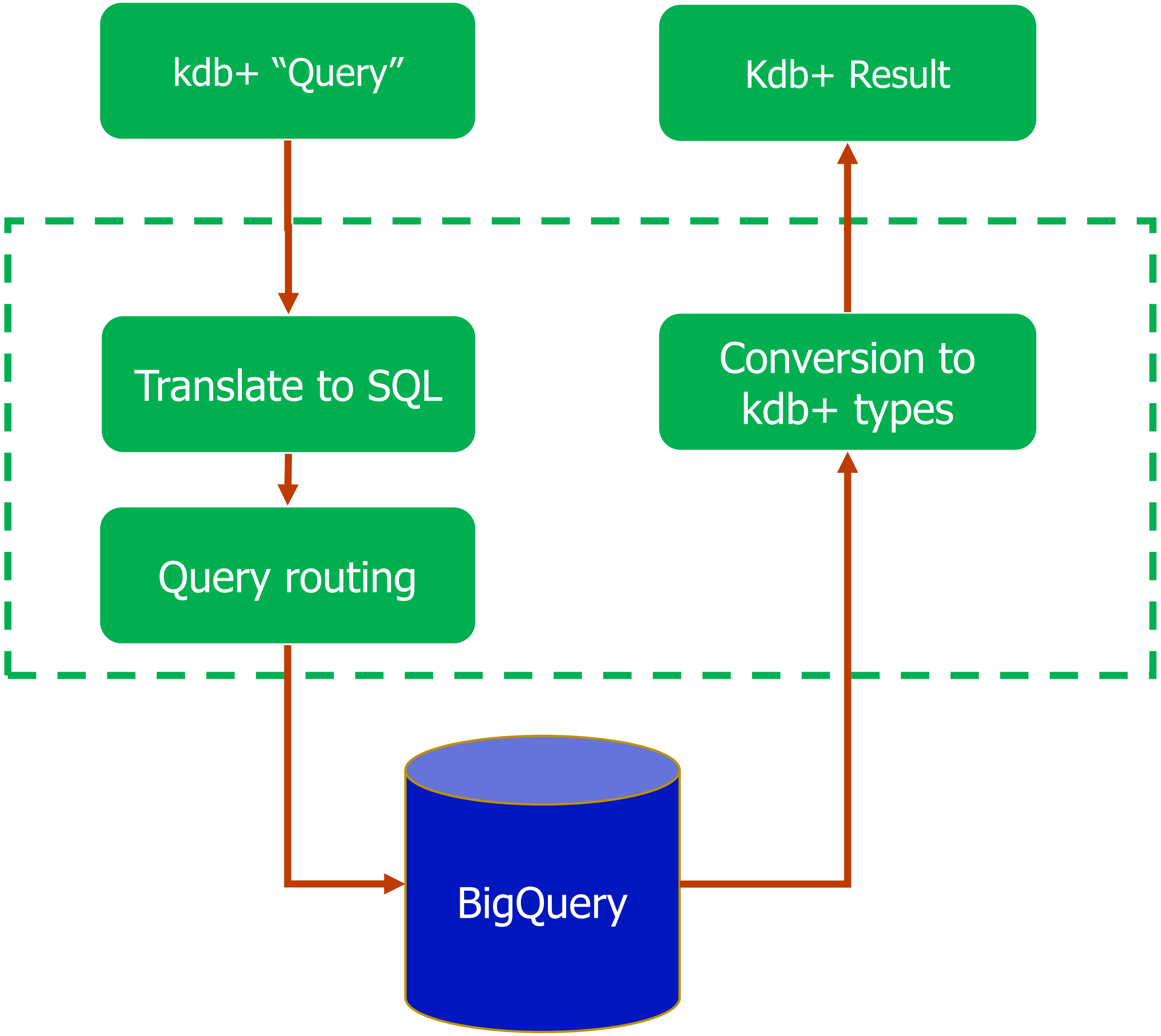
Why use a cloud warehouse rather than kdb+ in cloud?

- Less configuration e.g. servers, storage, network groups, docker images, scaling, licences, global replication…
- Presumed cheaper cost

Why BigQuery over other cloud data warehouses?

- GCP/BigQuery gaining traction in Financial services
- Refinitiv data offering in BigQuery
- Pricing model seems attractive (low storage cost)
- Easy to set up with limited configuration options

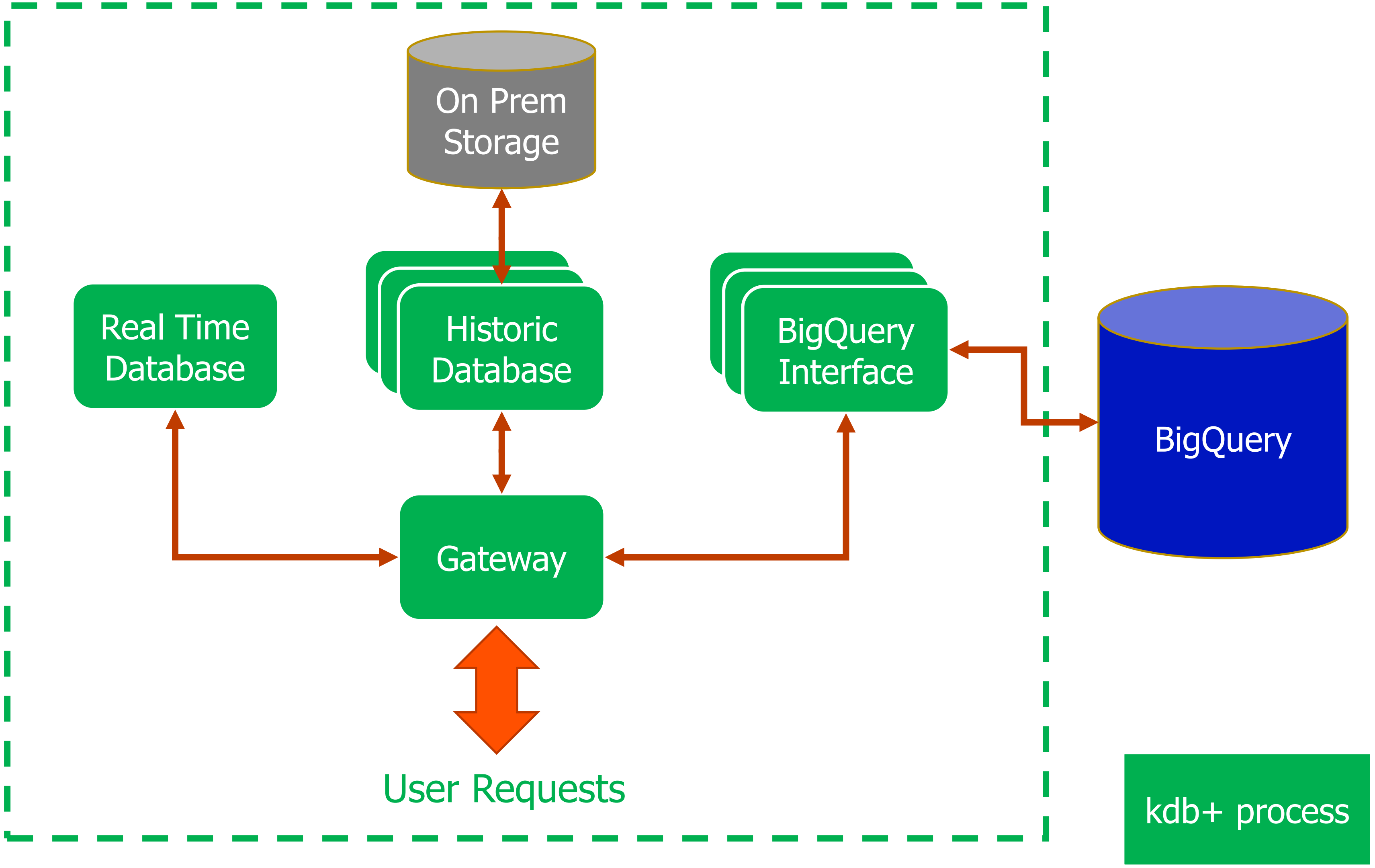
The Interface



BigQuery Interface

- Uses python API to send requests to BigQuery and marshal responses
- Can performs additional processing and analytics at the data

Example Architecture



Gateway

- Receives user requests, applies querying routing and load balancing logic
- Post processing before returning to the gateway
- Joins results in custom ways

What?

Built on TorQ

- AquaQ open source, kdb+ framework (TorQ excluding API available on Github)
- Extension of a tried and tested architecture

OR **Run Standalone**

- Interface available to run standalone and integrate into existing frameworks

OR **Wrap with TorQ**

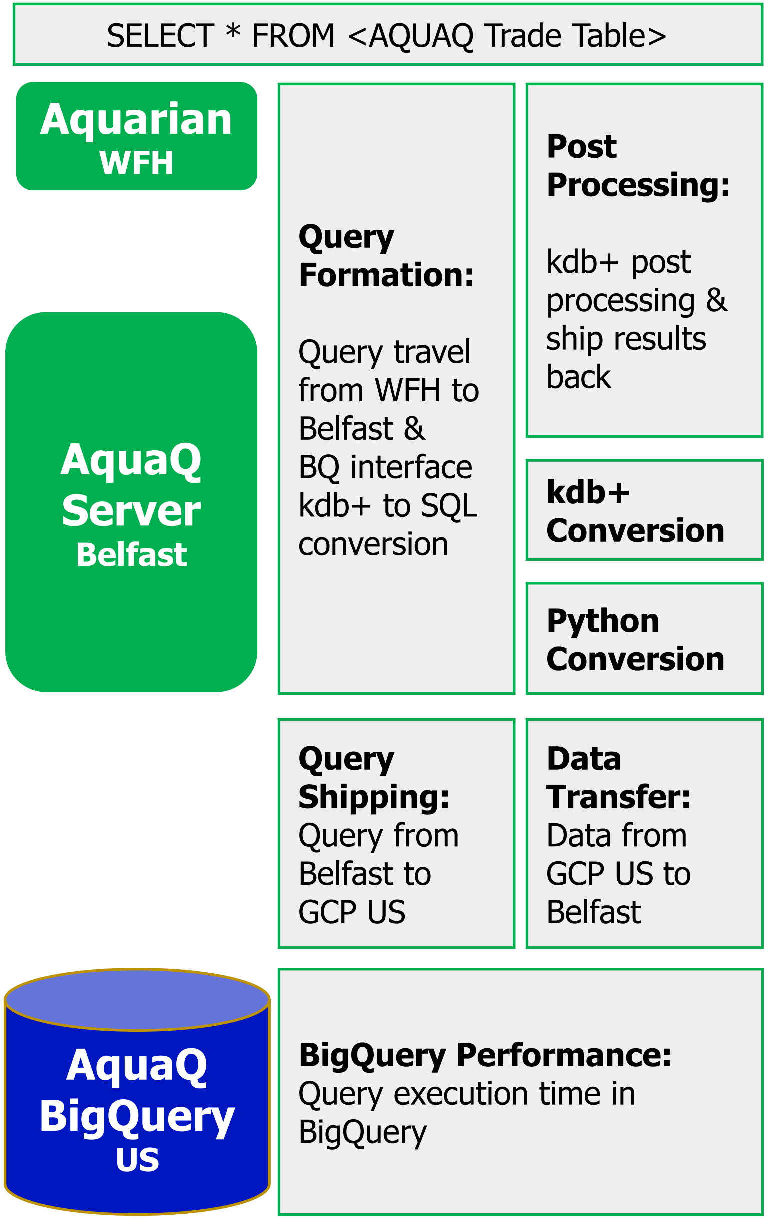
- TorQ can be wrapped around existing setups

Generic Data Access API

- API provides common interface to BigQuery and kdb+
- A dictionary (map) input is translated into SQL or the parse query format used by kdb+
- Not all dictionary keys are required or support by both sides

Key	Description	Example
tablename	Table to query	`trade
timecolumn	Temporal column to filter on	`exchangetime
starttime	Start time of data (date timestamp)	2020.11.01D09:00
endtime	End time of data (date or timestamp)	2020.11.03D16:00
columns	Columns to return	`time`ticker`price`size
filters	Additional (not time) filters	`exch`size!((=;`LSE);(>;100))
aggregations	Aggregations (defined by us)	`max`min`first`last!4#`price
grouping	Grouping of data	`ticker`exch
timebar	Time bucketing (grouping)	(5;`minute;`time)
ordering	Order of return result set	(`asc`price;`desc`size)

Performance: Query Journey [AquaQ GCP]



SELECT * FROM <AQUAQ Trade Table>

Aquarian WFH

AquaQ Server Belfast

AquaQ BigQuery US

Query Formation:

Query travel from WFH to Belfast & BQ interface kdb+ to SQL conversion

Post Processing:

kdb+ post processing & ship results back

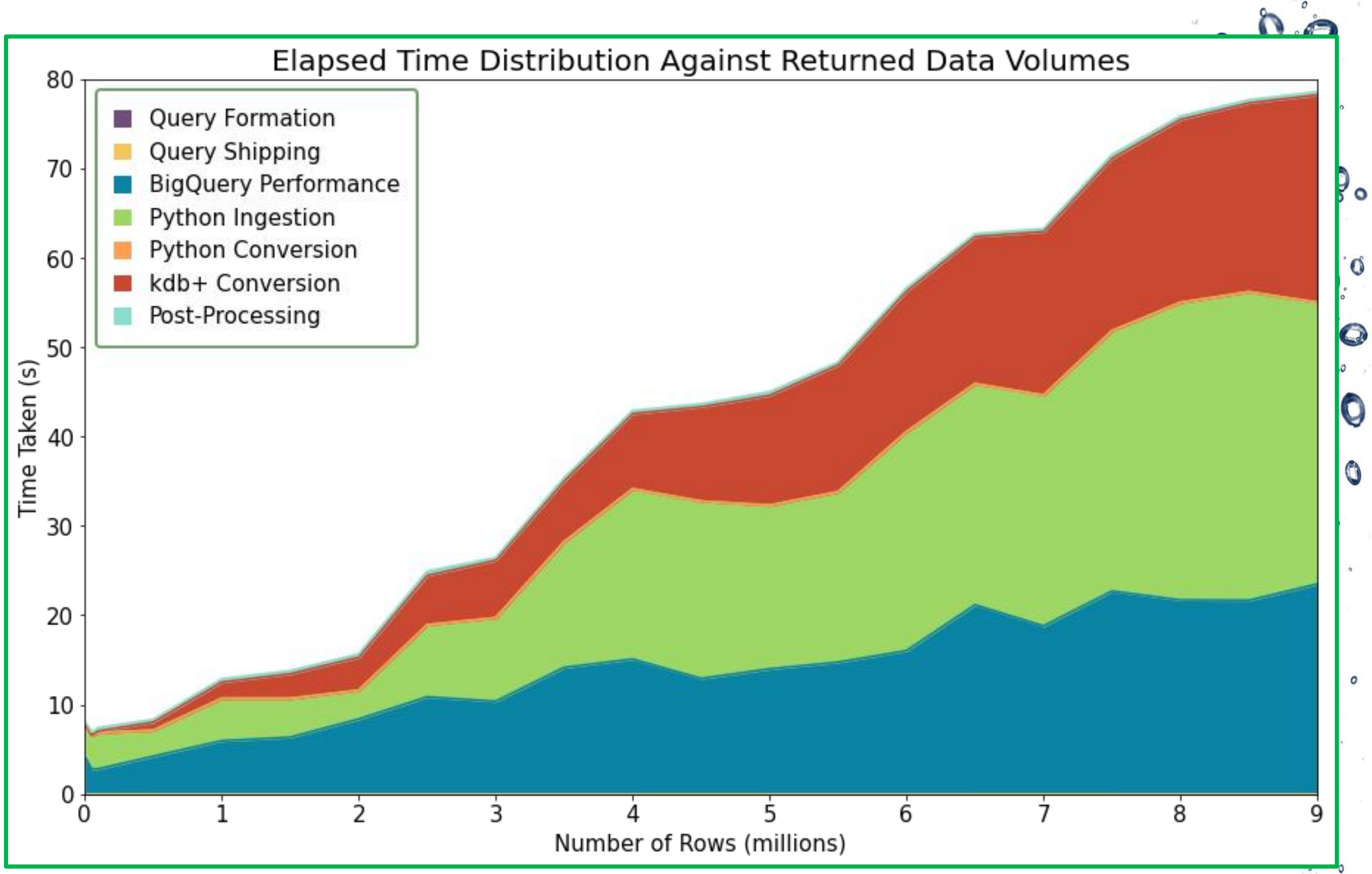
kdb+ Conversion

Python Conversion

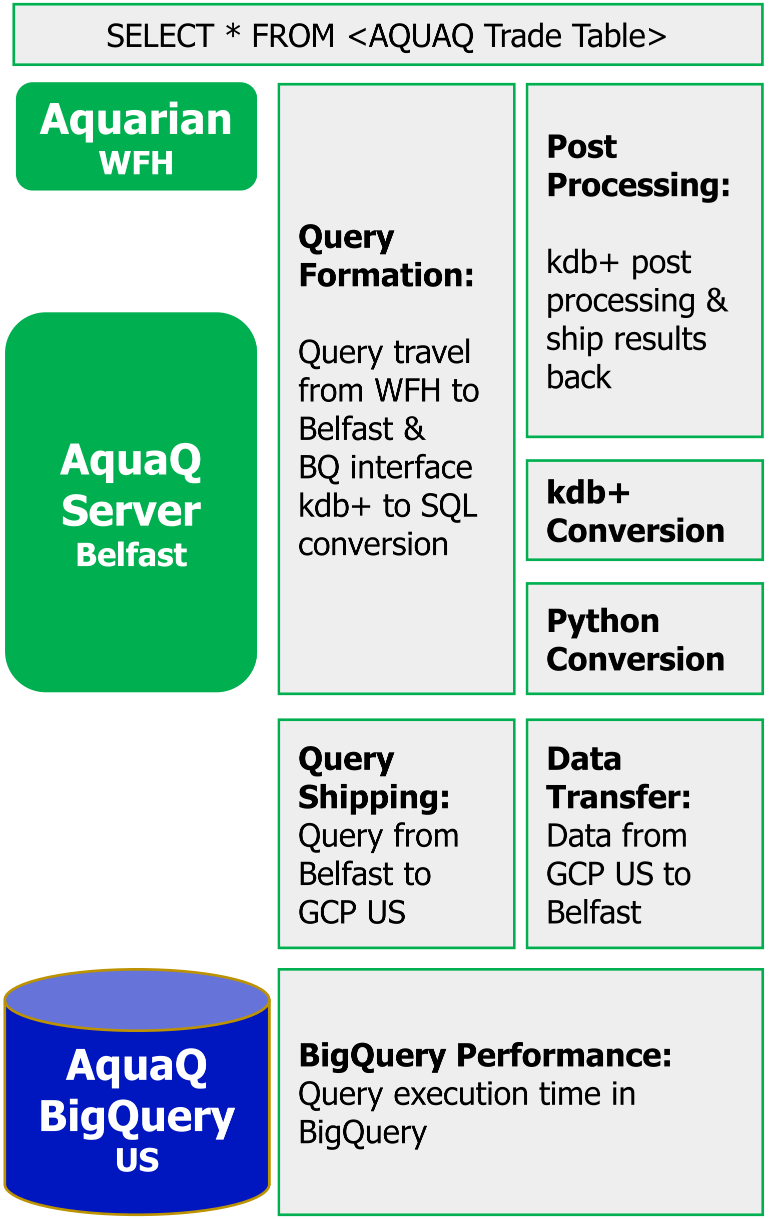
Query Shipping: Query from Belfast to GCP US

Data Transfer: Data from GCP US to Belfast

BigQuery Performance: Query execution time in BigQuery



Performance: Journey Proportions [AquaQ GCP]



SELECT * FROM <AQUAQ Trade Table>

Aquarian
WFH

AquaQ Server
Belfast

AquaQ BigQuery
US

Query Formation:

Query travel from WFH to Belfast & BQ interface kdb+ to SQL conversion

Post Processing:

kdb+ post processing & ship results back

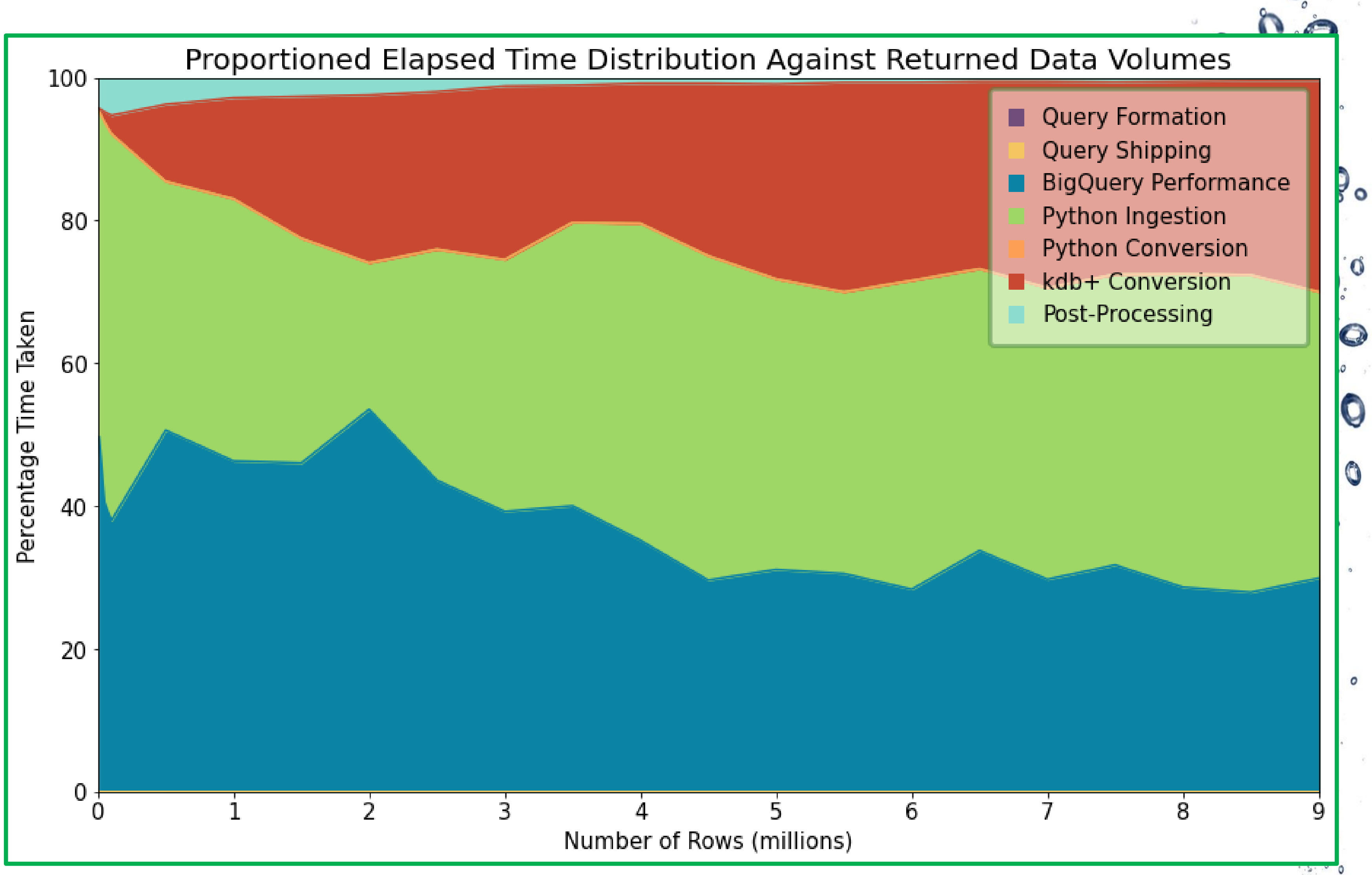
kdb+ Conversion

Python Conversion

Query Shipping: Query from Belfast to GCP US

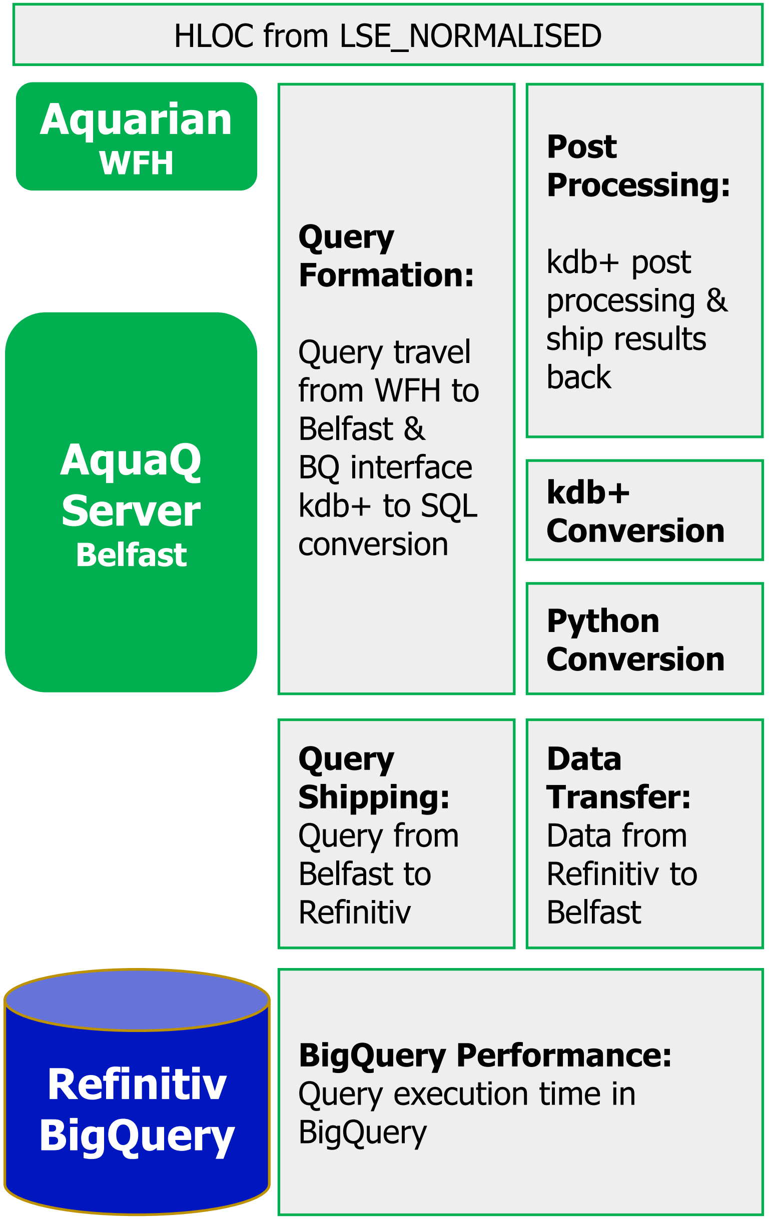
Data Transfer: Data from GCP US to Belfast

BigQuery Performance: Query execution time in BigQuery



AQUAQ
Analytics

Performance: Query Journey [Interface to Refinitiv]



HLOC from LSE_NORMALISED

Aquarian
WFH

AquaQ Server
Belfast

Refinitiv BigQuery

Query Formation:

Query travel from WFH to Belfast & BQ interface kdb+ to SQL conversion

Post Processing:

kdb+ post processing & ship results back

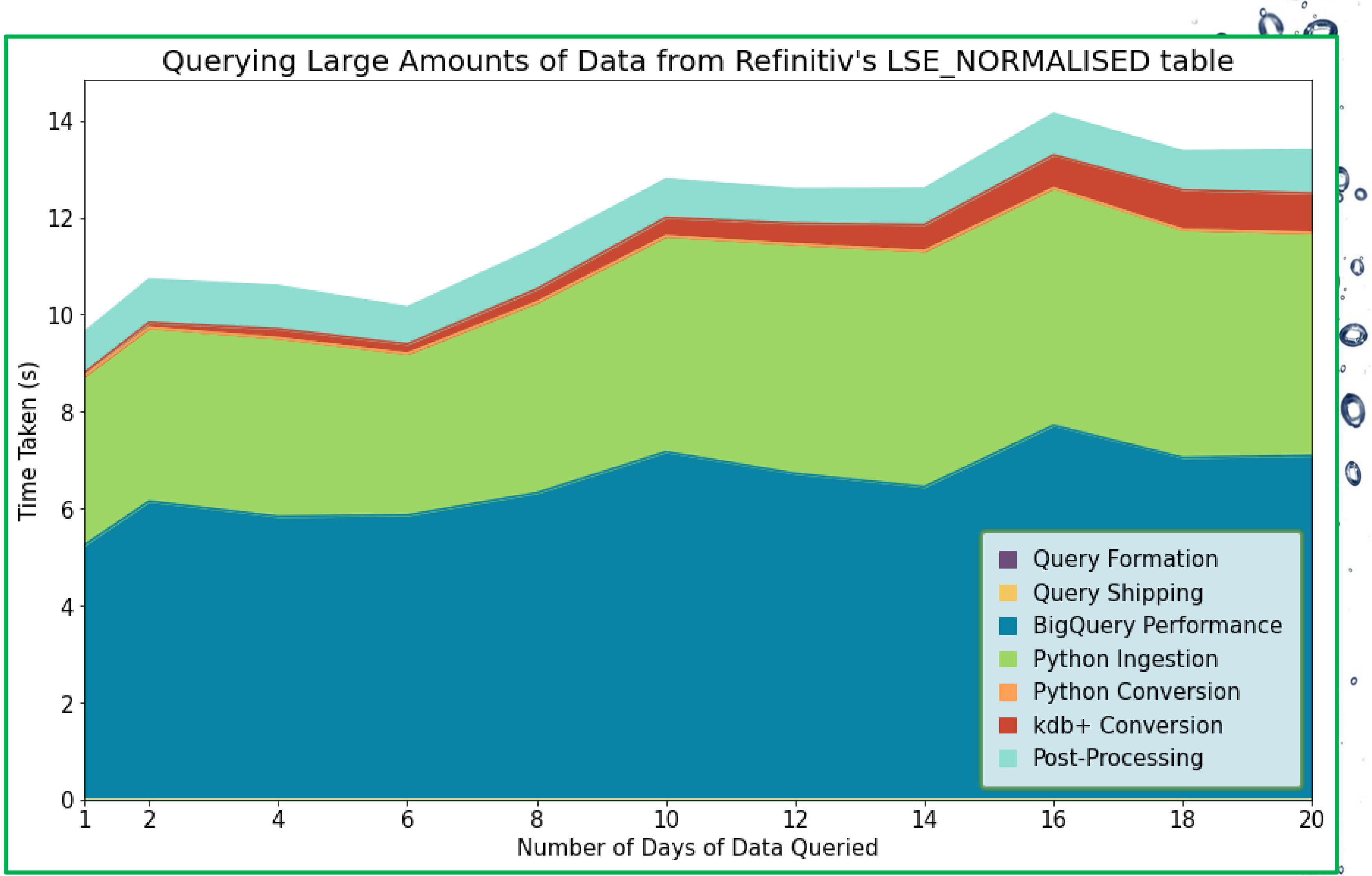
kdb+ Conversion

Python Conversion

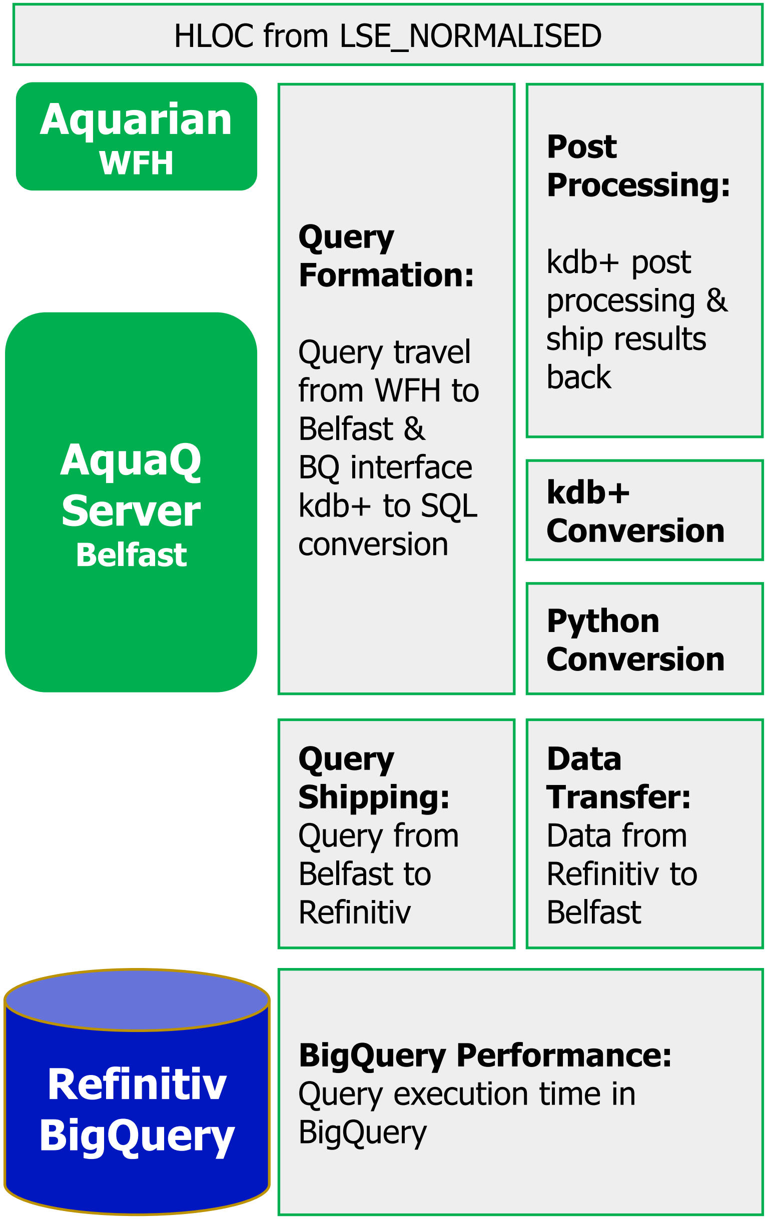
Query Shipping: Query from Belfast to Refinitiv

Data Transfer: Data from Refinitiv to Belfast

BigQuery Performance: Query execution time in BigQuery



Performance: Journey Proportions [Interface to Refinitiv]



HLOC from LSE_NORMALISED

Aquarian **WFH**

AquaQ Server **Belfast**

Refinitiv BigQuery

Query Formation:

Query travel from WFH to Belfast & BQ interface kdb+ to SQL conversion

Post Processing:

kdb+ post processing & ship results back

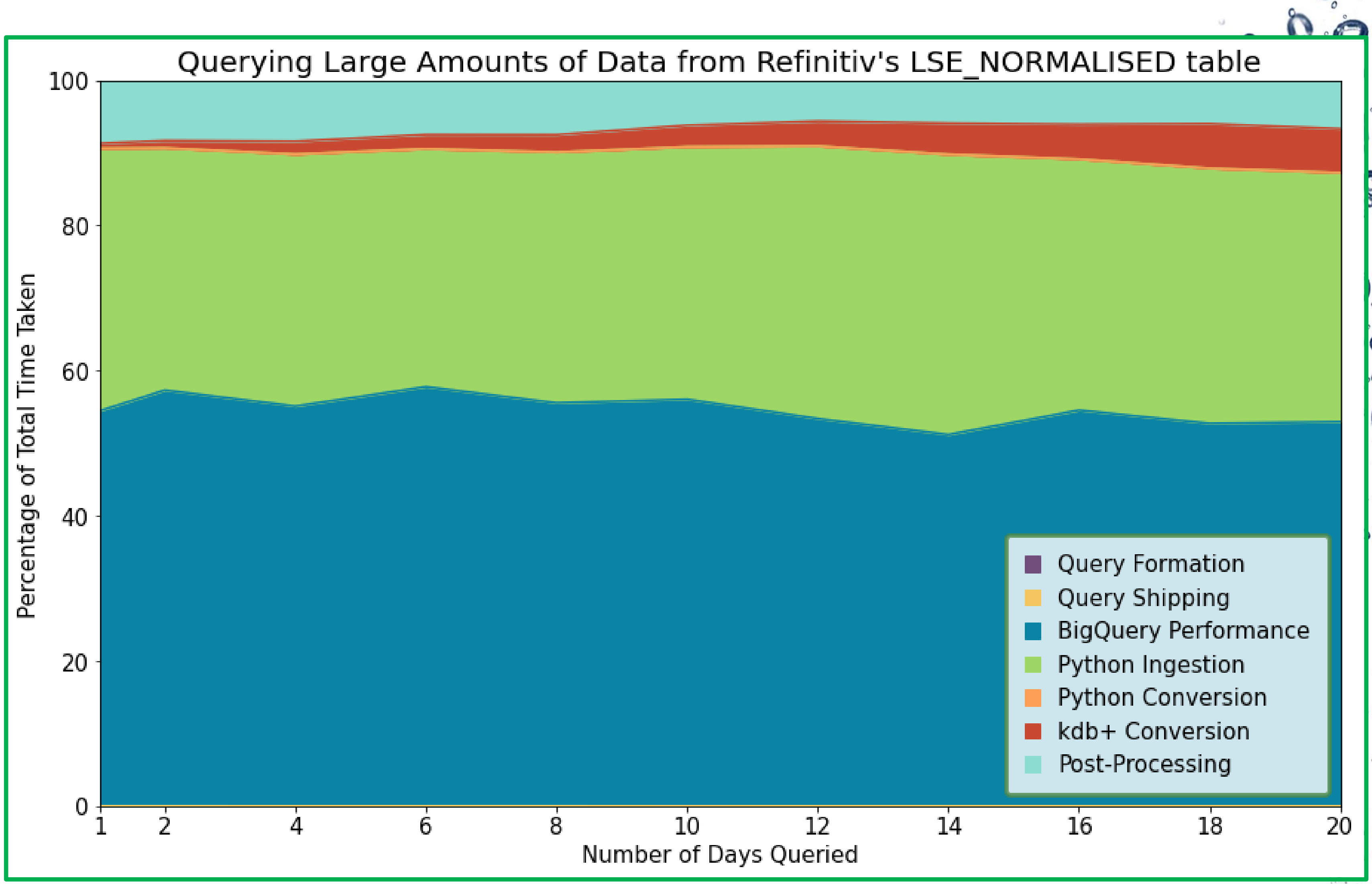
kdb+ Conversion

Python Conversion

Query Shipping: Query from Belfast to Refinitiv

Data Transfer: Data from Refinitiv to Belfast

BigQuery Performance: Query execution time in BigQuery


Querying Large Amounts of Data from Refinitiv's LSE_NORMALISED table
Percentage of Total Time Taken
Number of Days Queried
Query Formation
Query Shipping
BigQuery Performance
Python Ingestion
Python Conversion
kdb+ Conversion
Post-Processing