**Chapter 1 – Selecting Appropriate Storage Technologies**

**Ingestion Modes**

* Application
  + generated by apps. Pushed back to end services
  + e.g. log data, shipping addresses, click stream
  + Ingested by Compute, Kubernetes, App Engine
  + Can also be written to cloud monitoring, Cloud SQL or Cloud Datastore
* Streaming
  + Small messages transmitted continuously
  + E.g. sensor data, IoT (Internet of Things)
  + Event time = time data point generated
  + Process time = time data point entered pipeline
  + Pub/Sub
* Batch
  + Bulk ingested
  + E.g. csv files
  + If batch load volume is large use Cloud Transfer Service & Transfer Appliance

**Storage Considerations**

* Data Access Patterns
  + e.g. transaction data that is filtered, store in databases
* Access Controls
  + e.g. restrict access to tables/views in database
  + Cloud IAM for storage buckets
* Time to store
  + Temp data – HDD/SSD on Compute Engine VMs
  + Nearline/Coldline/Archive class on storage buckets

**Process and Analyse**

* Data Transformations
  + data cleansing – erroneous data correction
  + Cloud Dataflow (both batch and stream)
* Data Analysis
  + Extract useful info from the data
  + Cloud Dataflow, Cloud DataProc, BigQuery and ML Engine
* Explore and Visualise
  + Cloud Datalab – based on Jupyter Notebook, comes preinstalled with widely used python modules
  + Data Studio – tabular and charts with a drag and drop GUI

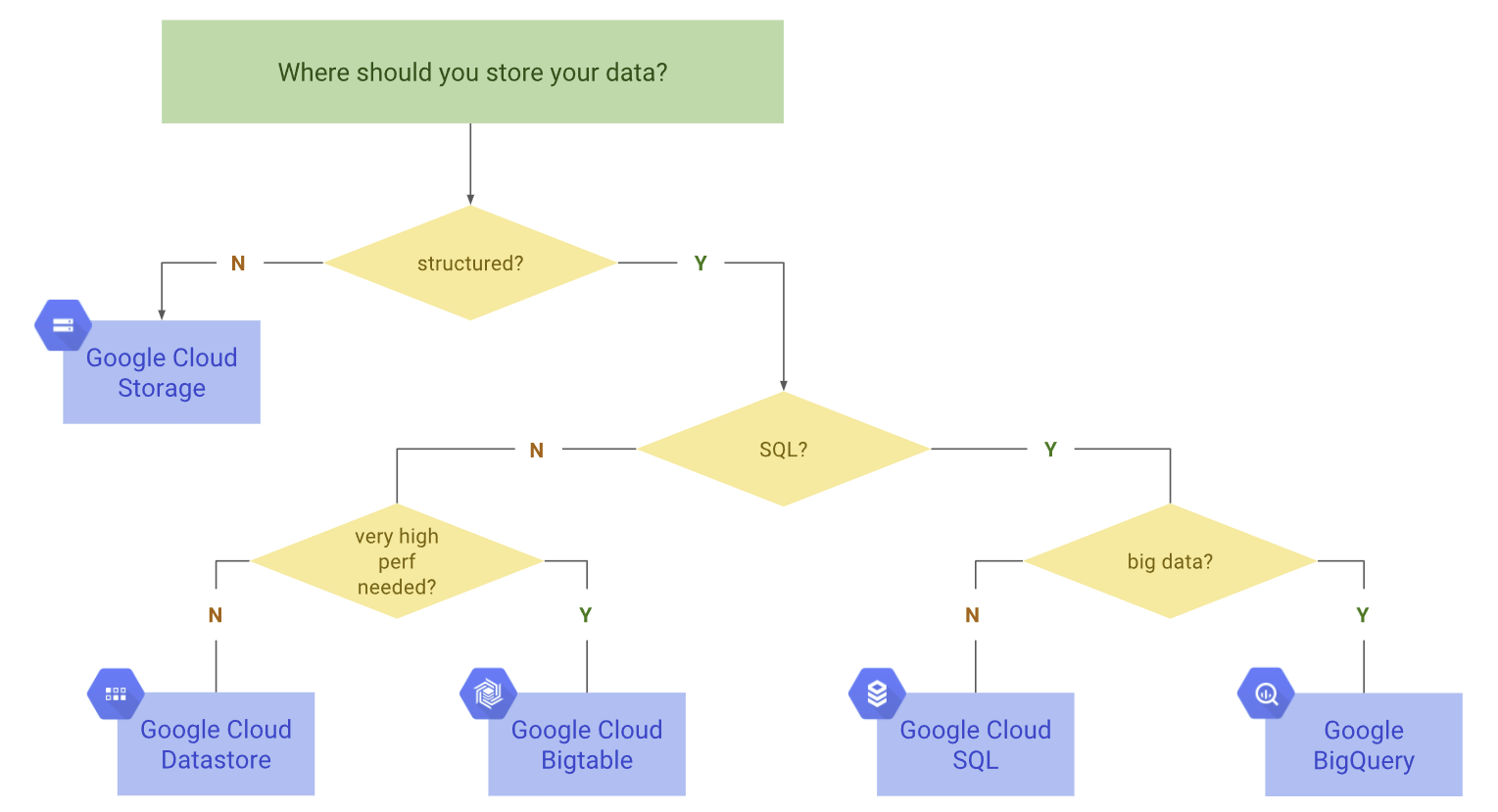
**Technical Aspects of Data**

* Volume
  + Cloud Storage – max item size 5GB, no limit read/write
  + BigTable – 8TB HDD or 2.5TB SSD per node, 1000 tables
  + BigQuery – no table limits, up to 4000 partitions per table
  + Compute Engine – 64TB persistent disks
  + My SQL – 1st gen 500GB
  + MySQL, PostgreSQL, SLQ Server – 2nd gen 30TB
* Velocity
  + Rate data is sent and processed by application
* Variation
  + How much does the data structure vary?
  + NOSQL database (e.g. MongoDB)
    - when data varies a lot
    - stored key:value pair
* Data Access Pattern – data can be accessed in different ways. 4 considerations:
  + How much data is retrieved in read operation?
  + How much data is written in insert operation?
  + How often is data written?
  + How often is data read?

**Data Structure**

* Structured
  + Fixed set of attributes, tables where cols = attribute, rows = records
  + Cloud SQL/Spanner – row-orientated storage (transactional)
  + BigQuery – column-orientated storage (analytics)
* Semi-Structured
  + Attributes can vary
  + Stored either as documents or wide-column
  + Document retrieved using indexes
  + Wide-column retrieved using Row Key access
* Unstructured
  + No defined schema or data model
  + E.g. text files, mages, videos, BLOBs (Binary Large Objects)

**Storage Decision Tree**



**Schema Design Considerations**

* Relational Database design
  + OLTP
    - Online Transaction Processing
    - Transactional
    - Normalised
  + OLAP
    - Online Analytical Processing
    - Dimensional models
* NoSQL Database
  + Key-value
    - Associative arrays or dictionaries
    - Keys lock up values
    - Cloud Memorystore – based on Redis
  + Document
    - Complex data structures, used as values
    - Document group data that is read together
    - Cloud Datastore
  + Wide Column
    - High volumes, low-latency writes
    - More write than read
    - Limited query range
    - Single key lookup
    - BigTable – Hbase/Hadoop
  + Graph Database
    - Model entities and relationships as nodes and links
    - i.e. social network
    - GCP has no managed graph database