Building a Google Cloud Data Platform

A blueprint for success with BigQuery

April 2020

servian_

Table of Contents

Table of Contents

1. Overview

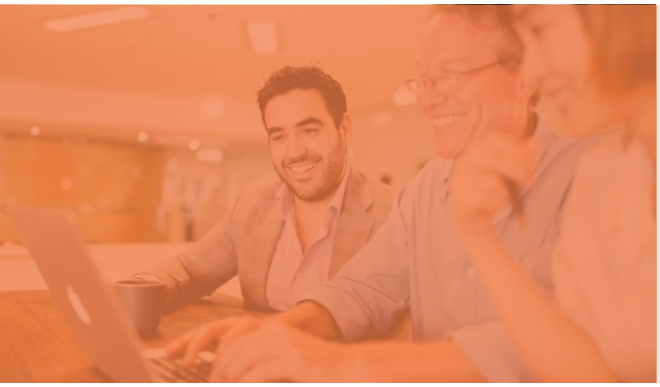
1.1 Introduction

Servian designs, delivers and manages innovative data and analytics, AI/machine learning, digital, customer engagement and cloud solutions that help clients sustain competitive advantage.

It has a diverse enterprise and corporate customer base that includes over 200 clients across government, finance, telecommunications, utility, insurance, construction, airline and retail sectors.

Founded in 2008 and headquartered in Sydney, Servian has offices across Australia and New Zealand, as well as London and Bangalore. It has over 500 consultants, making it the largest pure play IT consultancy in Australia.

Servian is platform agnostic and can implement technology solutions across any data, digital and cloud environment (including Google, Amazon and Microsoft).



1.2 Objective

The objective of this document is to outline Servian's approach on how to build and operate a successful cloud based data platform on Google Cloud using BigQuery at the core of the architecture.

1.3 Approach

Most IT and data projects fail. Data lakes and data harbours typically become data swamps, but our approach is unapologetically different. With a heritage in data, we understand what works.

Working on cloud-based platforms for a number of years, we have learnt what needs to be transferred, upgraded and changed from traditional approaches. For example, those who have tried to set up secure Hadoop for fInancial services in the cloud will understand that it is difficult and that there are more productive options.

Cloud infrastructure has changed the ability to access scale for compute and storage, which has also enabled streaming and near real-time use cases. This has enabled faster time to more accurate insights, which we know from our clients has never been more paramount.

Cloud does significantly change data and security risks. If implemented appropriately though, cloud-based solutions can, in many ways, be implemented with tighter controls than the on-premises predecessors.

2. Cloud Data Platforms - Key Concepts

Terminology

- Data warehouse - the collective home in an organisation for analytical data and activities
- Data mart - a collection of tables usually arranged along business lines or analytical domains
- Data lake - a collection of data sources which are linked through federated access mechanisms
- OLTP - Online Transaction Processing
- OLAP - Online Analytical Processing

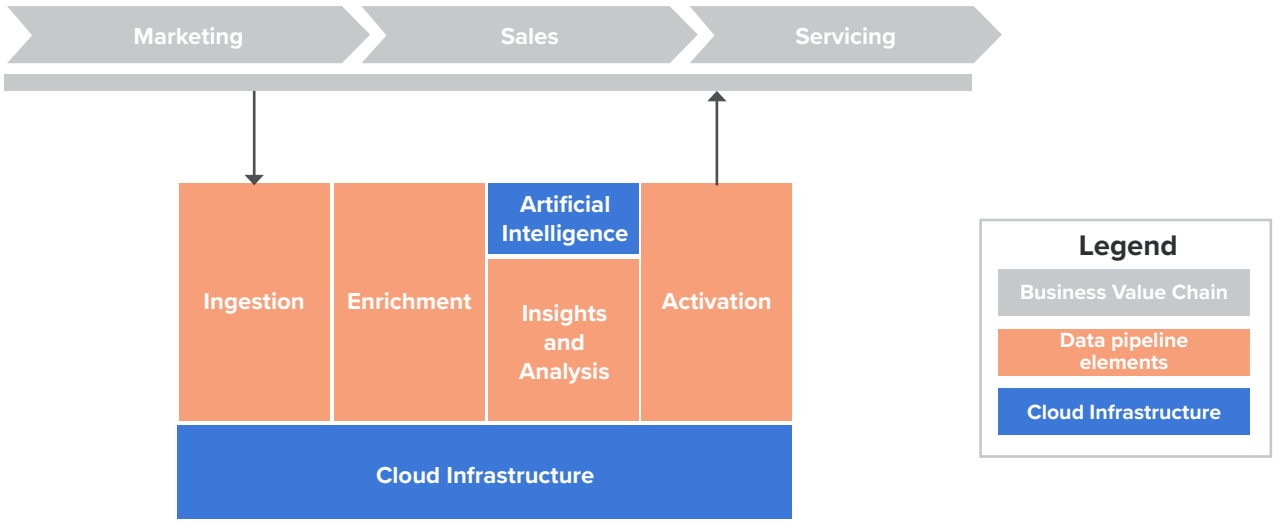
2.1 Platform objectives

There are a number of drivers towards cloud-based data platforms. The most common include:

- Reduced time to insight
- Reduced cost
- Support for streaming / near real-time data sources
- Support for machine learning
- Ability to scale storage or compute to handle greater data volumes from an increasing range of data sources
- Ability to tap into technical capabilities beyond an organisation's own
- Ability to refresh solutions as hardware/software end-of-life approaches

To make the most of the opportunity presented by cloud's cost, scale and capability advantages, cloud requires a shift not just in technology, but also, in the processes and people that support and utilise the platform.

2.2 Conceptual architecture



Our conceptual architecture is based on:

- Broad based ingestion from across the business value chain
- Activating outcomes into business processes in the value chain
- Separating data processing to maximise consistency and agility to support multiple use cases
- Understanding that different user groups will access different parts of the platform
- Understanding that not all data is equal
- Leveraging cloud infrastructure with Infrastructure as Code

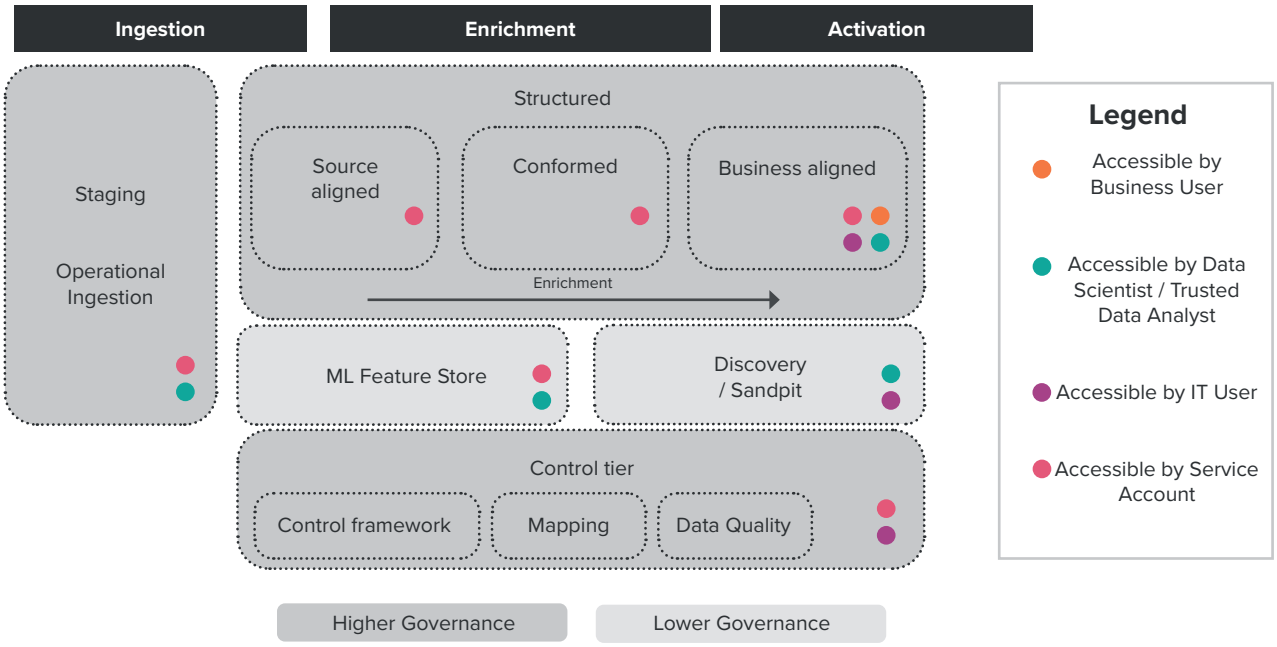
Our approach sits in the middle of the extreme approaches to building platforms. We do not advocate for either the loosely governed data lake – which becomes a data swamp – or the tightly governed, model-everything approach of a constricted, traditional, data warehouse.

2.3 Platform user considerations

From our experience with vendors, data platforms are truly successful when their outputs are used.

To ultimately unlock value in a cloud data platform, Servian looks to understand this through its advisory and data strategy work by starting with a view of the users which we map to different areas of a data platform. We then apply detailed analysis to look at applying fine grain permissions for the various groups across the organisation.

The diagram which follows outlines how we aim to balance secure, dependable processing with appropriate accessibility to help various users unlock value from a cloud data platform that contains shared data.



Note that the diagram above seeks to represent how coarse grain roles are mapped to the different areas of a cloud data platform.

Further user details include:

- Business users - business insights analysts, dashboard users, report users
- IT users -technical business analysts, operations, data engineers, ML engineers
- Service accounts - technology processes which may include the accounts used by ETL/ELT data processing, access via APIs

2.4 Information management

2.4.1 Data classification

Servian is of the strong view that all data is not equal. It is not all liquid gold.

Within a cloud data platform we recommend at least three tiers for classifying data use such as:

- Gold - governed to support processes such as financial reporting
- Bronze - low cost, typically just staged source data
- Discovery - sandpit areas which can be used to experiment and explore data to quickly find value

In addition to general data use, data in many organisations is classified to include the following privacy concerns:

- PII - Personally Identifiable Information
- Confidential information - e.g. health data
- National Privacy Principles, GDPR and other overseas requirements

2.4.2 Data governance

Data platforms by their nature have data from multiple stakeholders and, as a shared service, operate and provide more value when a well-constructed set of rules are used to enable the teams which develop and operate the platform. These rules help the operators act on behalf of the various stakeholders.

This can include looking at how to address concerns which include data security, regulation and compliance, visibility and control.

Servian's data governance principles are:

1. Consistency within a domain is important. Consistency across domains is helpful but not mandatory.
2. It's more important to know the accuracy of the data than it is for the data to be accurate.
3. All data assets should have an expectation of a timeline to be refreshed, for example, three years for a report, five years for a table and eight years for a domain.
4. Establish cadence for prioritising data programs and measuring benefits.
5. No one is omniscient enough to know the outcome beforehand. Businesses need to be able to prototype, creating new data to understand the outcome they desire.
6. Ingestion of source should always be in its entirety (i.e. all columns / full width) and to its lowest grain.
7. There is no such thing as shared ownership of data. All datasets belong to a business owner.
8. Business users may use reporting/visualisation tools as a prototyping approach, but these should be handed over to technology teams for production support when support is required.

2.4.3 Data quality

What gets measured, gets done.

Measuring data quality is vital for data which is to be trusted for financial decision-making and other key business processes.

This typically falls into two categories:

- Technical verification - row counts of the various steps within key data pipelines
- Business verification of key data entities - this will include counts and/or calculation of key business entities and their most important attributes. For example:
- Number of customers by postcode
- Transaction value by product
- Account value by customer grouping

These measures are typically calculated at each of the three key points in a data pipeline: at source, as it is staged into a cloud data platform and at the point of consumption.

When measuring data quality, various data quality measurement dimensions should be considered. These are accuracy, integrity, consistency, completeness, validity, accessibility, and timeliness.

Another important component of data quality is understanding the lineage of data. Data lineage becomes a key maintenance activity as platforms become more and more complex. Without well-documented lineage, the ability to deliver change in the platform slows as impact assessments, development and testing times grow or become at risk as people who are Subject Matter Experts move on to new roles.

Third-party tools such as data transformation tools including Talend, Informatica and IBM can provide lineage within their tooling environment.

In many organisations, the most important aspects of data lineage are detailed interface contracts for data that comes in and out of a data platform.

Data lineage can also be derived from sources such as source code and scheduler configuration. A runtime view can be developed from logging sources or job batch control if effective frameworks are utilised.

2.4.3 Data risks and privacy

Many organisations have legitimate concerns about data risk and data privacy.

Moving to a cloud-based environment does impact the controls environment as existing access models are changed when moving from infrastructure that has physical access to infrastructure that has public cloud access.

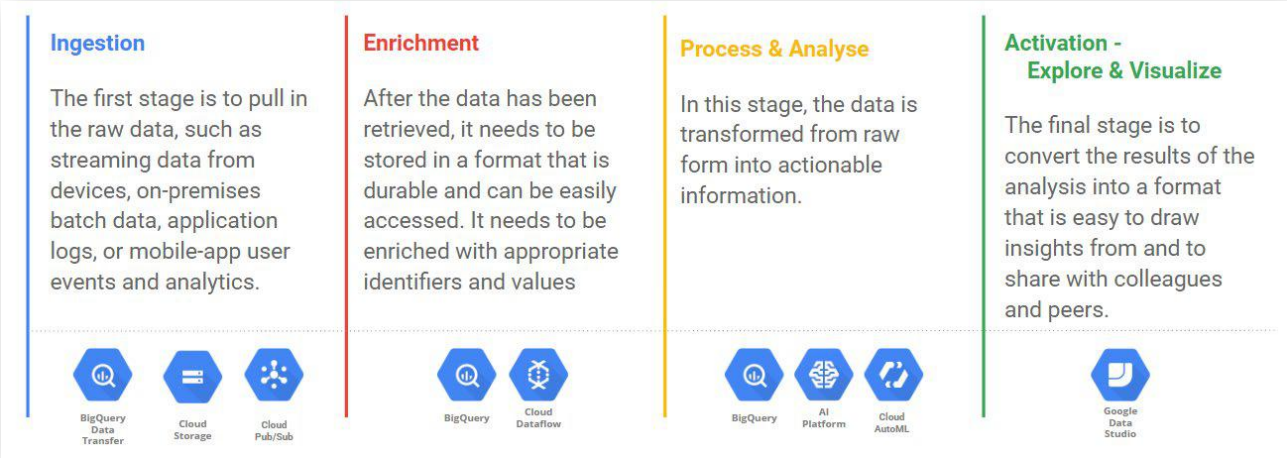
However, building a data platform using cloud infrastructure is also an opportunity to reset how security and data risk concerns are addressed. This is due to the ability to configure the full infrastructure lifecycle in code and create immutable infrastructure with consistent and centralised logging. These cloud capabilities are an opportunity to move to a more proactive stance on risk, as security and data risk concerns can be addressed by being incorporated into automated build and deploy processes. Tooling can scan the environment for policy exceptions in a continuous, proactive manner.

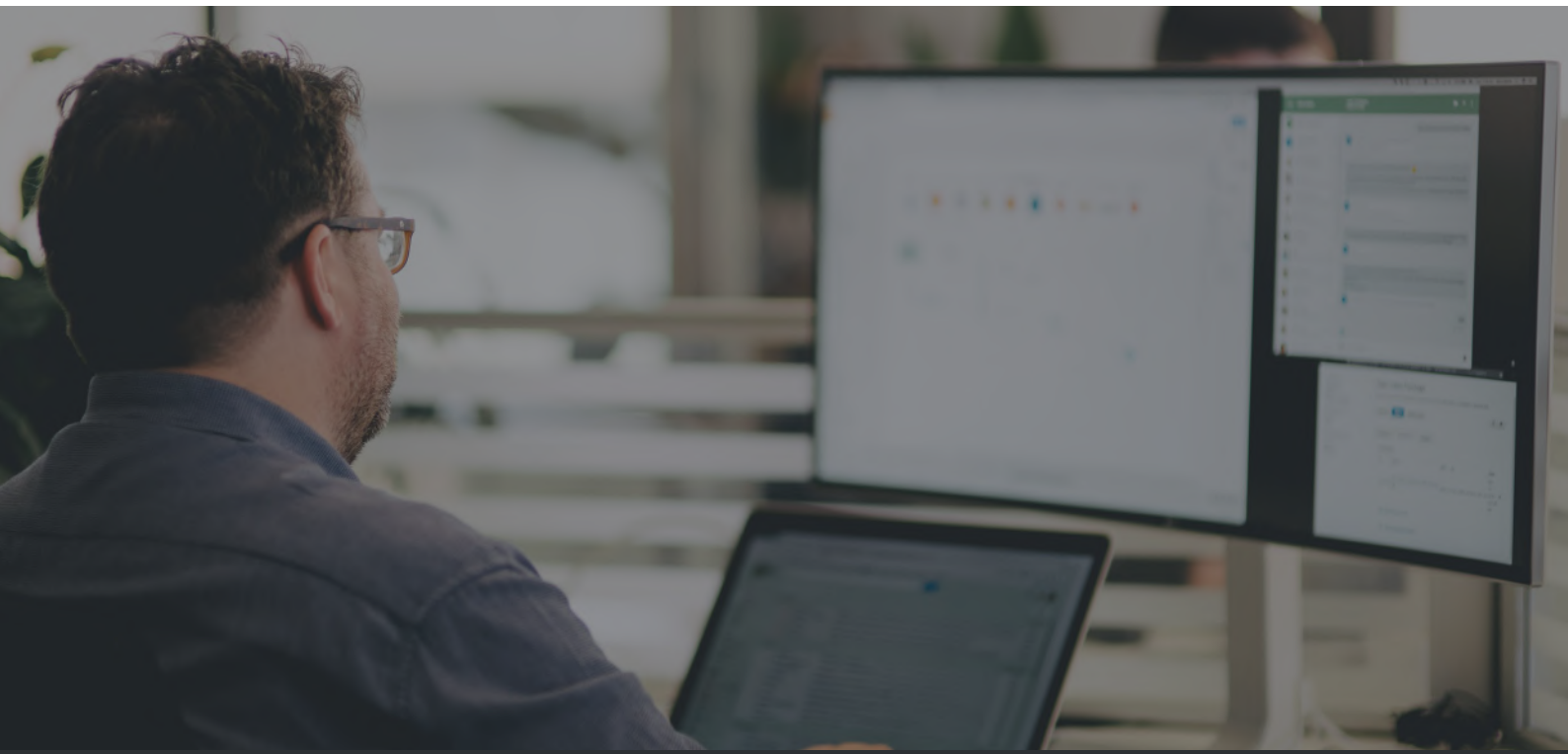
Cloud data platforms are also able to access security encryption/hashing/tokenisation/obfuscation services with an on-demand model, via APIs which any reasonable developer can leverage. This lowers the barrier to entry for creating safe data capabilities, which can integrate broader sets of data to unlock more value propositions from integrated and shared datasets.

3. Google Cloud as a Data Platform

3.1 Google Cloud's data capabilities

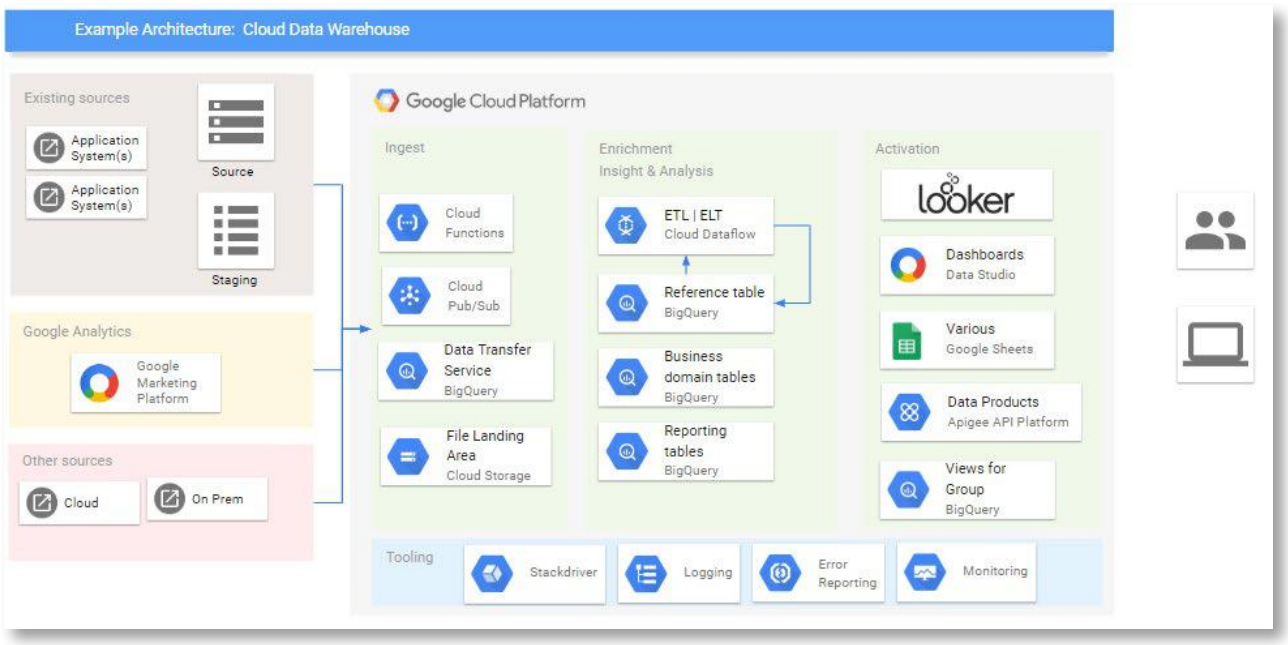
Google Cloud provides a comprehensive set of tools that deliver capability across the full spectrum of needs for any organisation. Many of the tools are serverless and remove many overheads in regards to setup and maintenance, enabling fast time to insight. In other words, Google Cloud abstract a lot of the operational complexity of their data and analytics tools, making it easier and faster for BigQuery practitioners to solve business problems. In addition, Servian views Google Cloud as a developer-friendly platform, which helps maintain high levels of productivity across teams.





3.2 Solution architecture - Google Cloud data capabilities

As the diagram below shows, multiple tools can be brought together to provide a data platform solution on Google Cloud. BigQuery, with its ability to enable analysis of billions of rows of data at massive scale, is at the centre of the architecture. That said, other tools are needed to build an end-to-end data platform solution.



3.3 BigQuery

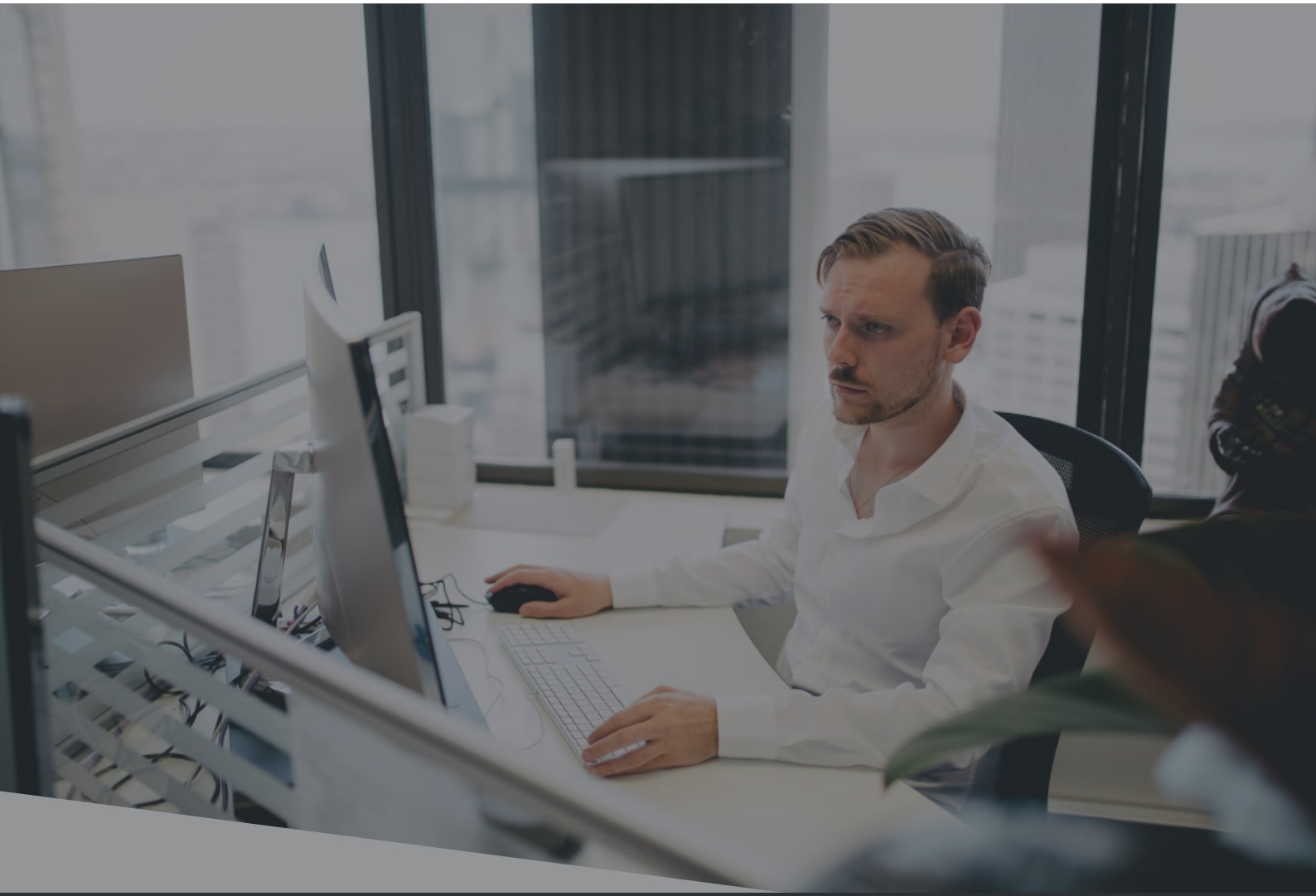
BigQuery is Google Cloud's enterprise data warehouse, which delivers incredibly fast and scalable SQL based data capabilities on Google's infrastructure. Based on the Dremel technology, BigQuery has the ability to process billions of rows in seconds while delivering this capability at a fraction of the cost of high-end dedicated hardware.

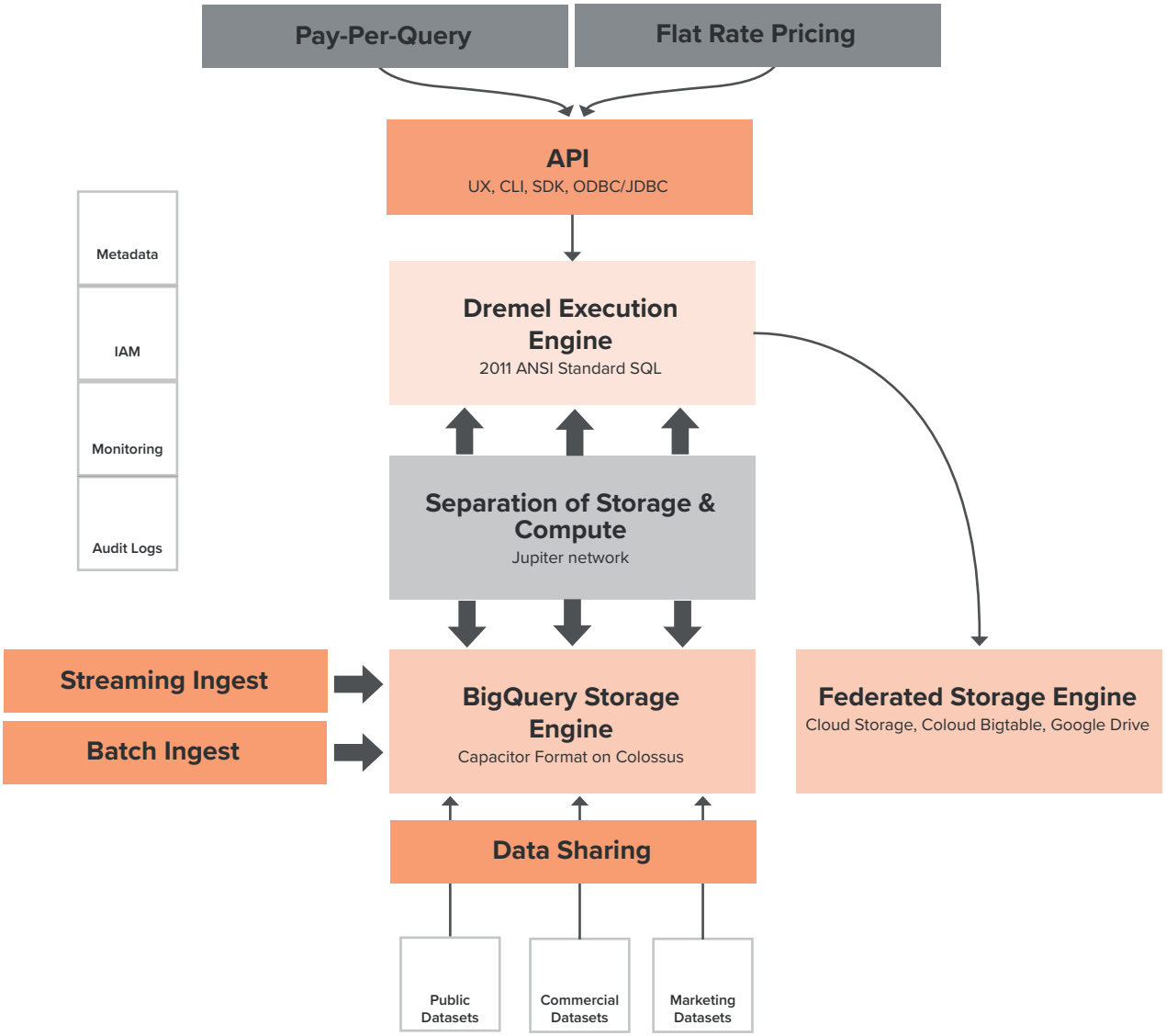
3.3.1 Overview

BigQuery's secret sauce is made up of four main components:

1. **Capacitor**, which is its proprietary columnar storage format.
2. **Colossus** is the distributed file system, and the next generation of GFS.
3. **Dremel** is the core distributed query engine that runs on Google's custom hardware.
4. **Jupiter** is the Google petabit network that it all sits on top of, allowing vast amounts of data to be read/shuffled/moved at incredibly fast speeds.

These four components are what enable BigQuery to have a complete separation of storage and compute architecture, which has been at its core since inception. When a query is received by the BigQuery service, it first reads the data from disk (Colossus) over the petabit network. From there, all data is loaded into memory (RAM) using container based technology running on Google's Borg system. All BigQuery queries run in memory. It is worth pointing out however, that spill to disks can occur in rare circumstances.





BigQuery through the lens of a practitioner (source: http://tiny.cc/3xphmz)

BigQuery's success also partly comes from exposing all its power through standard SQL as an interface, which makes it accessible to millions of IT developers who have SQL skills. Although, these SQL skills need to adapt to the columnar structured way in which tables are organised to get the most out of BigQuery. In addition, although BigQuery is ANSI 2011 SQL compliant, there are some features and functions that are unique to BigQuery. As such, this requires some level of upskilling by teams.

BigQuery has a mature and rich Rest API that is exposed to provide aid in the setup of tables, datasets and ingestion functionality. Several client SDKs in different languages exist, making it easier for developers to interact with the API in their software.

It is these underlying components in BigQuery that enable a shift in the cost, speed and scale in capability when compared to traditional data warehouse hardware and software combinations. The following table highlights the contrasts between the old world and BigQuery:

THE OLD WORLD	THE NEW WORLD
Hardware constrained	Scalable storage and compute, capped only by budget
Batch oriented data processing in hours	Batch processing in minutes. Ability to support streaming ingest and therefore continuous processing for near real-time insights
Additional backup and recovery process, software and configuration	Built-in seven day history of changes, with point-in-time snapshot recall
Archiving processes with tape	Sliding time window approach - Long-term storage - Partition expiration
Software and hardware security patches, maintenance, index creation	Fully-managed service
Large technical upgrade costs	Features added continuously
Relational based table structures, i.e. Inmon, Kimball, 4NF	Denormalised, columnar optimised table structures using nested/repeated fields are best practice

3.3.2 Data modelling and structuring BigQuery

Due to its underlying OLAP architecture, BigQuery prefers wide, denormalised tables with nested and repeated data. Servian recommends to denormalise whenever possible when using BigQuery, contrasting the classic way in which relational data stores are typically designed. However, that doesn't mean BigQuery can't handle normalised data and joins. It absolutely can. It just performs better on denormalised data.

Storage savings from normalising data are insignificant when compared to the ability to aid in the execution of parallel jobs by using an optimal columnar design based on denormalised data structures. Denormalising the data structures involves adding extra columns to tables rather than splitting them. We also recommend separating BigQuery datasets (sets of tables) inline with conceptual structure outlined in this paper. **Note:** the project which a query is initiated from drives the billing for the same query. So some forethought into the way datasets and projects are organised should be undertaken to consider billing requirements.

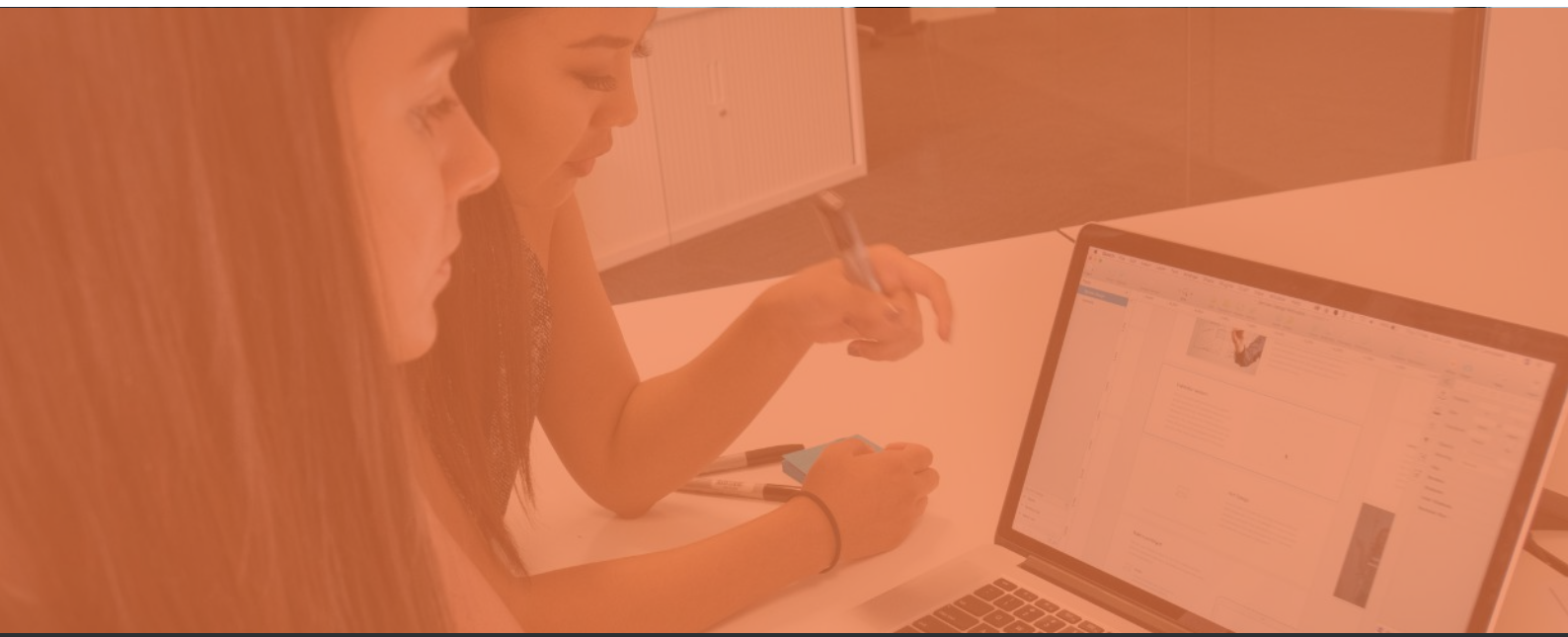
3.3.3 Ingestion

There are two ways to load data into BigQuery: batch and streaming. In most cases, batch loading is sufficient. Batch (or "bulk") loading into BigQuery is free of charge. Batch loading into BigQuery is almost always done via GCS, which acts like the glue holding everything together on Google Cloud. This is the easiest and most cost-effective way to get data into BigQuery and it is relatively quick. It is worth noting that a batch load from GCS is in fact running a federated SQL query under the hood in BigQuery to fetch and load the data. A common pattern that Servian employs when working with BigQuery in the enterprise for batch loading is to use Cloud Functions as the trigger when a file to be batch loaded arrives in GCS. There are two ways we recommend to load from GCS this way:

1. Configure and deploy a Cloud Function to trigger on file events in GCS. If the file size is always expected to be small (i.e. not a lot of data) and not expected to grow, the Cloud Function itself can call BigQuery's load API. This is a common pattern to use for micro-batching with small data volumes. The maximum nine minute execution time of Cloud Functions should not be an issue in this case. The pattern is: `GCS -> Cloud Function -> BigQuery.`

2. Configure and deploy a Cloud Function to trigger on file events in GCS. If the file size is large or expected to grow, then using the above approach is not recommended. Because load jobs are asynchronous in BigQuery, you need to poll and wait for the job to succeed or fail. In the case of a failure, you will need to propagate this, or handle it with retry logic. If the load job takes longer than nine minutes (the maximum execution time of a Cloud Function), then you will have no way to poll for the load job status. Instead of using a Cloud Function to directly trigger the load job, use a Cloud Dataflow template instead. This has no maximum timeout, and can alert on failed jobs. The trade off is that by using Cloud Dataflow this will be more expensive. The pattern is: `GCS -> Cloud Function -> Cloud Dataflow -> BigQuery.`

The second way to load data into BigQuery is by using its streaming API. This is not free, and charges are applied by Google. If users need their data in near real-time, then using the streaming API is a good way of meeting this requirement. However, it should only be used if and only if users absolutely need near real-time analysis. Under the hood, when BigQuery's streaming API is used, it uses Bigtable as a staging layer for the data before workers process the staged data and bring it into BigQuery. The most common (and Servian recommended) pattern for streaming to BigQuery is to use `Cloud Pub/Sub -> Cloud Dataflow -> BigQuery`. This is a cost effective and scalable way to ingest streaming data into BigQuery.



3.3.4 Enrichment, processing & analysis

ETL and ELT are common patterns that are applied in any data platform. When working with BigQuery and the use case dictates ETL, then the recommended pattern is to use Cloud Dataflow. Cloud Dataflow can enrich the data on the way into BigQuery using side-inputs from various other data sources e.g. GCS, CloudSQL, Spanner etc. When using Cloud Dataflow for ETL for BigQuery, it also provides the added benefit of being able to write unit tests around the pipeline code. These tests can be embedded in the automated CI/CD process. However, the tradeoff of using Cloud Dataflow for ETL is that of cost and performance.

Another pattern is to use ELT instead of ETL. This pattern is based on the premise that the data should be loaded like-for-like into BigQuery. Once the data is loaded into BigQuery, then it is transformed/cleansed/enriched directly using SQL. The benefit of this approach is that it is more performant and cost-effective. However, using SQL is harder to test and control. Servian currently recommends evaluating a tool like "dbt" when using an ELT pattern with BigQuery, which dramatically increases code reuse and maintainability when doing your transforms.

3.3.5 Performance and cost optimisation

Contrary to popular belief, BigQuery is not limitless. Users are bound by the number of "slots" that they can consume concurrently across their project(s). A BigQuery slot is a unit of computational capacity required to execute SQL queries. BigQuery automatically calculates how many slots are required by each query, depending on query size and complexity.

BigQuery has two pricing models:

1. **On-demand:** this is the default model. This is the most flexible option. It is based on the amount of data processed by each query users run. Currently, each project can consume up to 2,000 slots concurrently. However, BigQuery may decide to burst through this at any time if the demand on the multi-tenant cluster allows for it.

2. **Flat-rate:** this pricing option is best for users who desire cost predictability. Flat-rate users purchase dedicated resources for query processing and are not charged for individual queries. Flat-rate only applies to compute/analysis. Users still have to pay for storage.

Deciding between on-demand and flat-rate for BigQuery will depend on the organisation's use cases and query access patterns. As a general rule of thumb, Servian recommends the following:

1. Begin with on-demand on all projects.
2. Continuously monitor slot usage and cost.
3. When monthly compute/analysis cost exceeds $15K (AUD), then begin to look at flat-rate.
4. If flat-rate is appropriate, then keep adding flex-slots until it gets fast enough or too expensive, then convert those flex slots to monthly/annual commitments to save money
5. Use flex-slots to cover one-off, or bursty workloads

When using a tool like BigQuery, users can quickly become lazy and costs can creep up. It's important to educate your teams/users on how BigQuery charges for queries and to reinforce best practices to keep costs down, such as:

1. Avoid using `` `SELECT *..` `` queries, which will charge for a full table scan.
2. Keep SQL queries lean and performant. BigQuery is not limitless. Data skews, too many joins, ORDER BY etc. will hurt performance and queries will fail
3. Use partitioning where possible.
4. If using partitioning, add clustering in addition for maximum cost savings and performance gain.
5. Use custom quotas to control costs when using on-demand. Also, use max-bytes-billed to control individual query costs.
6. For partitioned tables, enforce users specify the partitioned column in their WHERE clause by setting the require-partition-filter to true. This reduces cost and speeds up query time.

3.4 Data transformation - ELT / ETL

3.4.1 Cloud Dataflow

Cloud Dataflow is Google Cloud's managed service for scalable streaming and batch processing. Cloud Dataflow can be thought of as fully-managed Apache Beam pipelines on Google Cloud. Cloud Dataflow should not be seen as an alternative to BigQuery, but rather a complementing tool that aids in the building of data platforms on Google Cloud.

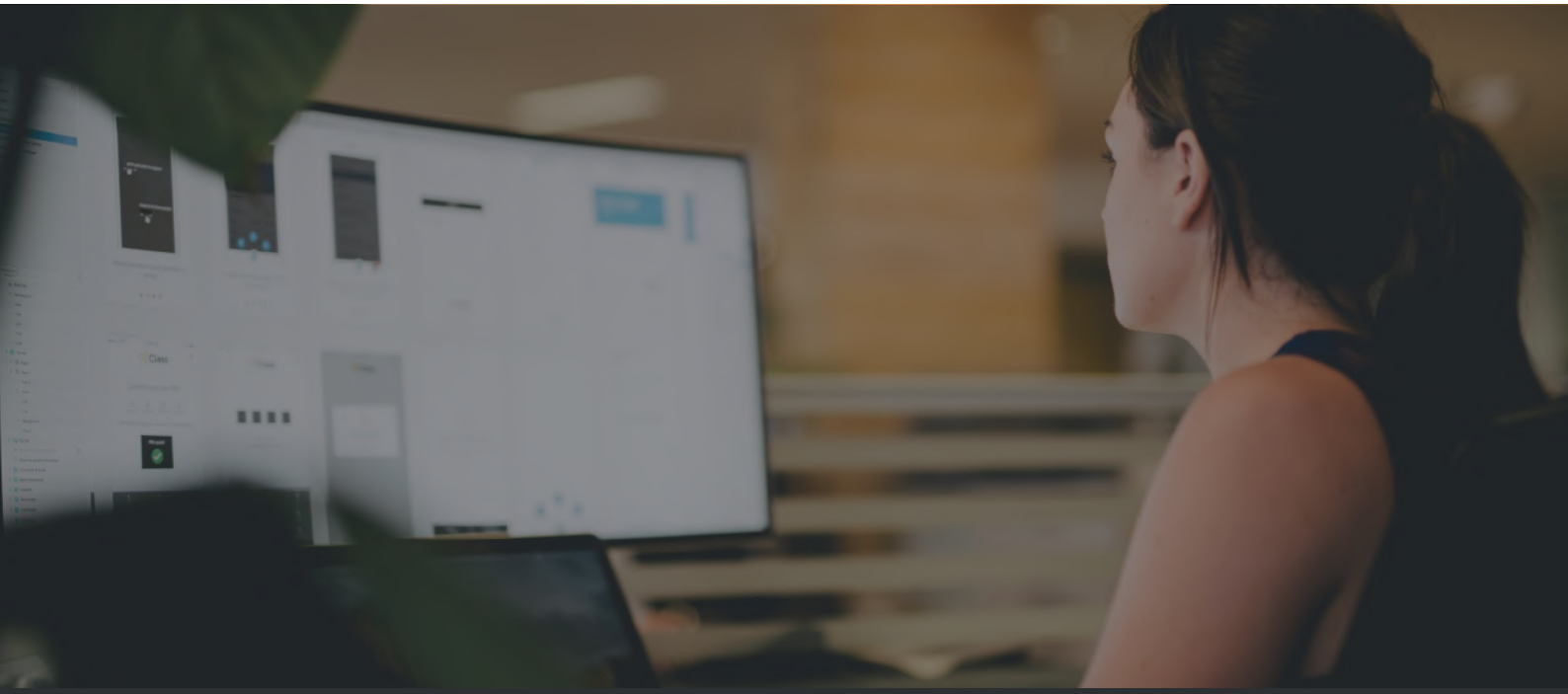
The performance and scalability of Beam based processing has truly impressed us as we work on data platforms for our large clients. The standard patterns libraries and documentation are a great place to start to look at how to implement many typical ELT/ETL processes in Apache Beam. See the Appendix for link.

Servian strongly recommends using the Java SDK/libraries for Apache Beam over Python, or other language implementations for that matter. This is primarily due to the maturity, feature parity, and performance of the Java SDK/libraries being superior to the other options. We have come across many clients who have tried to use the Python SDK/libraries, but encounter many problems and bugs that simply don't exist in the Java implementation.

3.4.2 Dataprep

Dataprep is a powerful ELT tool provided by Trifacta, which runs on Google Cloud as a fully-managed service. It can help to explore, clean and prep data very quickly and provides a mature UI for creating pipelines without the need to write code. Under the hood, Dataprep generates Cloud Dataflow pipelines that run and process user's data.

However, Servian highly recommends that users be aware that this is not a Google Cloud native tool, and as such, customers must accept Trifacat's own T&Cs before using it. This also means that you need to give Trifacta elevated permissions on your Google Cloud project(s) for it to work. For companies operating outside of the United States, be aware of the terms and conditions of this service and that the locality of processing is in the United States. Due to these points, it is imperative that enterprises engage their security, compliance and privacy teams before allowing teams to use this tool.



3.4.3 Other data manipulation tools on Google Cloud

There are a number of third-party data tools that can service and/or ingest into Google Cloud. These include options available from Google in addition to tools available on the marketplace and tools which can run elsewhere and interact with Google Cloud. Some of the Google tools include their recent acquisition of Alooma and also the CDAP based fully-managed offering of Data Fusion. More traditional ETL tools such as Informatica, IBM Infosphere, Talend can be easily integrated with Google Cloud tooling. ELT tools such as Trifacta, Datameer, 5Tran, Striim, Matillion are also worth considering if you are building a cloud data platform on Google Cloud, especially if your enterprise already has existing licensing and skills.

We recommend you also consider some of the following criteria when evaluating these tools:

- Cost based on a realistic usage profile across dev, test and production environments. For example, Data Fusion runs on a GKE cluster (24/7) and creates ephemeral Dataproc clusters.
- Connectors to cloud services - Salesforce, AWS S3, Hubspot, Marketo, etc.
- Connectors to existing data sources - MS SQL Server, Oracle, Redshift etc.
- Available as a marketplace service.
- Do you need the power of tools which have scale out processing? These can have implications for cost and spin up time
    - Dataproc/Hadoop - Alooma
    - Dataflow - Stitch
- Source control support.
- Support for hooks which can be leveraged by a CI/CD process.

3.4.4 Scheduling and orchestration

Apache Airflow spun out of AirBnB as an open source tool to schedule and monitor workflow schedules which are time or dependency triggered. Google offers Apache Airflow as a fully-managed service called Cloud Composer. However, Servian has worked with clients where Cloud Composer has presented several challenges in the enterprise. As an alternative, we have seen clients decide to use plain vanilla Apache Airflow in a GKE cluster, but this brings a lot more operations and maintenance to manage. Another option that we've seen success with, which is a lot simpler and cheaper, is to use Cloud Scheduler with Cloud Build for basic data and pipeline orchestration. However, although this is a much simpler pattern, it is quite limited when it comes to catering for the requirements of the enterprise.

3.3.5 Data risk

Google Cloud has various mechanisms to address data risk.

The most important is Cloud IAM which controls Identity and Access management. A well-defined access model, which plugs into robust access lifecycle controls, when linked to Cloud IAM.

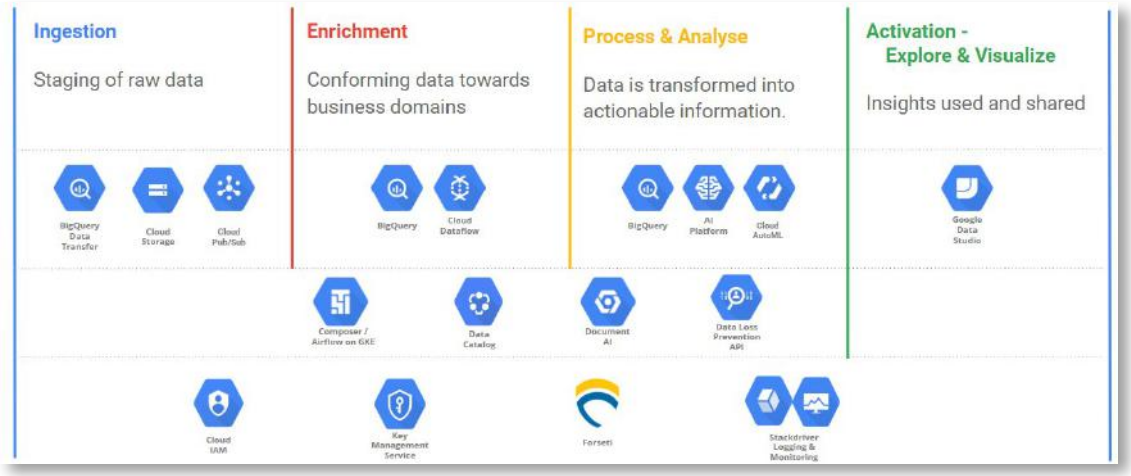
There are various dataset and project sharing techniques for addressing separation of control concerns which can balance the need for accessibility with the need for appropriate access. BigQuery itself can be the centre store for Audit logs and other security and risk data in the organisation. Alternately, this information can be exported if an organisation has a different SIEM or security analysis platform within which it consolidates data. Whichever platform, the immutability of audit logs is an obvious requirement.

The Data Loss Prevention API and Document AI tools from Google Cloud can be used to scan and then mask/obfuscate or encrypt sensitive data. These tools can be used to create a pre-processing

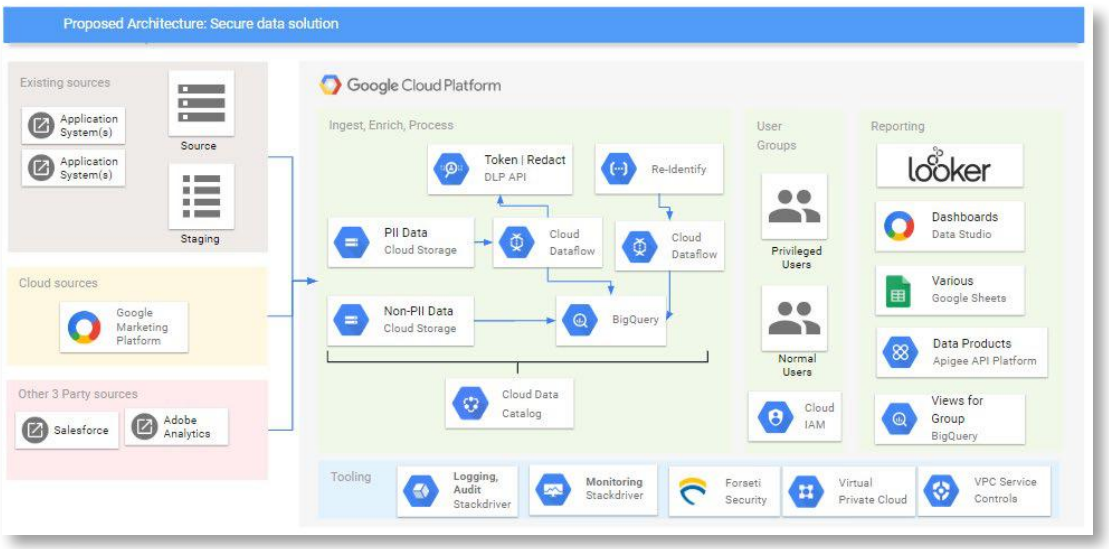
staging area of a data platform to triage sensitive, confidential or private data. The data is then able to be joined within normal data pipeline processing, but also accessed if specifically required. An outline is provided in the diagrams which follow.

Tools like Hashicorp Vault or Cloud KMS can be used to manage a token or public key cryptography lifecycle to also support the ability to manage the secrets at the center of these privacy or confidentiality datasets.

Another key concern for many organisations is limiting the exposure to the Privacy Act by ensuring that data is processed and resides within the country. This can be achieved by using Infrastructure as Code with tools such as Terraform to explicitly deploy locally. Implementing the open source security scanning tool, Forseti, which plugs into Cloud Security Command Center, can then be used as a way of scanning and ensuring compliance to policies which require localised data residency.



The diagram above shows the conceptual extension of the security tooling to augment data pipeline processing tools in Google Cloud



The diagram above shows a privacy/confidentiality data processing pipeline for de-sensitising data.

Leveraging BigQuery, and setting up an in-country instance, is also an opportunity to aggregate non-PII data, such as Google Analytics / GA360 data with your other data in a controlled environment. Potentially creating value from linking GA360 data with enterprise or other non-PII data to improve attribution and propensity models that can be used to drive more effective marketing and sales processes.

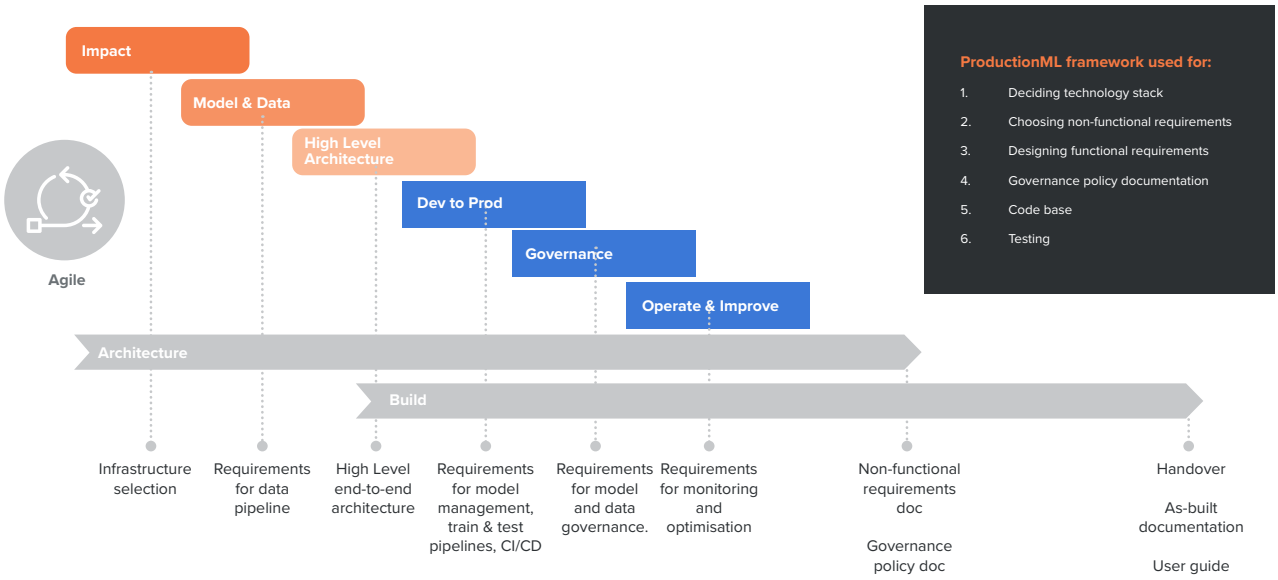
3.5 Machine Learning & AI

3.5.1 Google ML & AI tooling with used with applied data science

At Servian, we don't look at data science as a name to describe bashing away at the keyboard applying the latest ML libraries to a data or business problem. We believe in a hypothesis-driven, time-boxed approach to align data science to business outcomes with a focused search for value. Working with Google Cloud, we have observed over the last few years how the Google Cloud tools are helping to shorten the time to generate AI/ML insights, while at the same time growing the ability to consume more and more data.

This started with Cloud DataLab, the first Jupyter hosted notebook service amongst the cloud providers. This has since expanded to AI Platform & AI Platform Notebooks, AI Hub the various AutoML based APIs including Vision AI, Video AI, Speech to Text, Recommendations AI, Natural Language, AutoML Tables and the ability develop custom models as extensions of Google's models with Cloud AutoML.

The Servian approach to moving AI/ML processes into a production-ready state is outlined in the following diagram and description:



The above process is typically developed and deployed with a couple of different patterns using Google Cloud tools.

The patterns include:

- Loading historical data
  - `GCS -> BigQuery`
- Stream processing using expressive data science libraries
  - `Cloud Functions -> PubSub -> Dataflow -> BigQuery`
- Model training
  - AI Platform Training service
  - AI Platform Notebooks accessing BigQuery/GCS
  - AutoML Tables accessing BigQuery
  - Kubeflow defining whole model pipelines accessing BigQuery
- Predictions
  - AI Platform Predictions service
  - Exposed by Cloud Functions or App Engine hosted services
    - Accessing Cloud Firestore / Cloud SQL (PostgresQL)

3.5.2 Kubernetes for ML payloads

Prior to cloud capabilities influencing machine learning, a lot of machine learning was undertaken on single machines, perhaps with vertically-scaled hardware to support ML software solutions powered by solutions such as Matlab, R, and SAS.

Then came Hadoop, which typically required a full engineering team to support the platform. To do it well was beyond what many companies could muster.

Distributed and cloud-friendly technologies have dropped the barriers to entry for running machine learning at scale. However, it does require the usage of frameworks designed for distributed machine learning training and prediction such as Spark or Tensorflow. Alternatively, frameworks such as Scikit-learn on Python can be packaged up and applied in parallel through data parallelism applied in a co-ordinated pipeline.

Enter Kubernetes which, when combined with Google Cloud Storage and Apache Airflow/Cloud Composer, provides a highly effective alternative to leverage cloud computing resources to scale out and replace Hadoop running Apache Spark.

Kubernetes and GKE can also host Kubeflow, which is a Google supported open source project to help deploy Machine Learning workflows in a simple, scalable and portable manner.

Running Kubernetes and GKE well still requires specific expertise, but far less than Hadoop or many other solutions. It provides an opportunity to standardise how compute and storage is used, but the ability to rapidly deploy and re-deploy containers gives an amazing amount of flexibility which is well suited to fast evolving data science and data analysis workloads which need to leverage compute and storage at scale.

3.6 Data accessibility, reporting and visualisation

A key constraint of many traditional data warehouses was the restriction of access to reporting tools or the underlying data. With cloud solutions like BigQuery providing scale and tools such as Data Studio and Google Sheets enabling direct access to BigQuery with low cost to leverage, the traditional constraints have been removed.

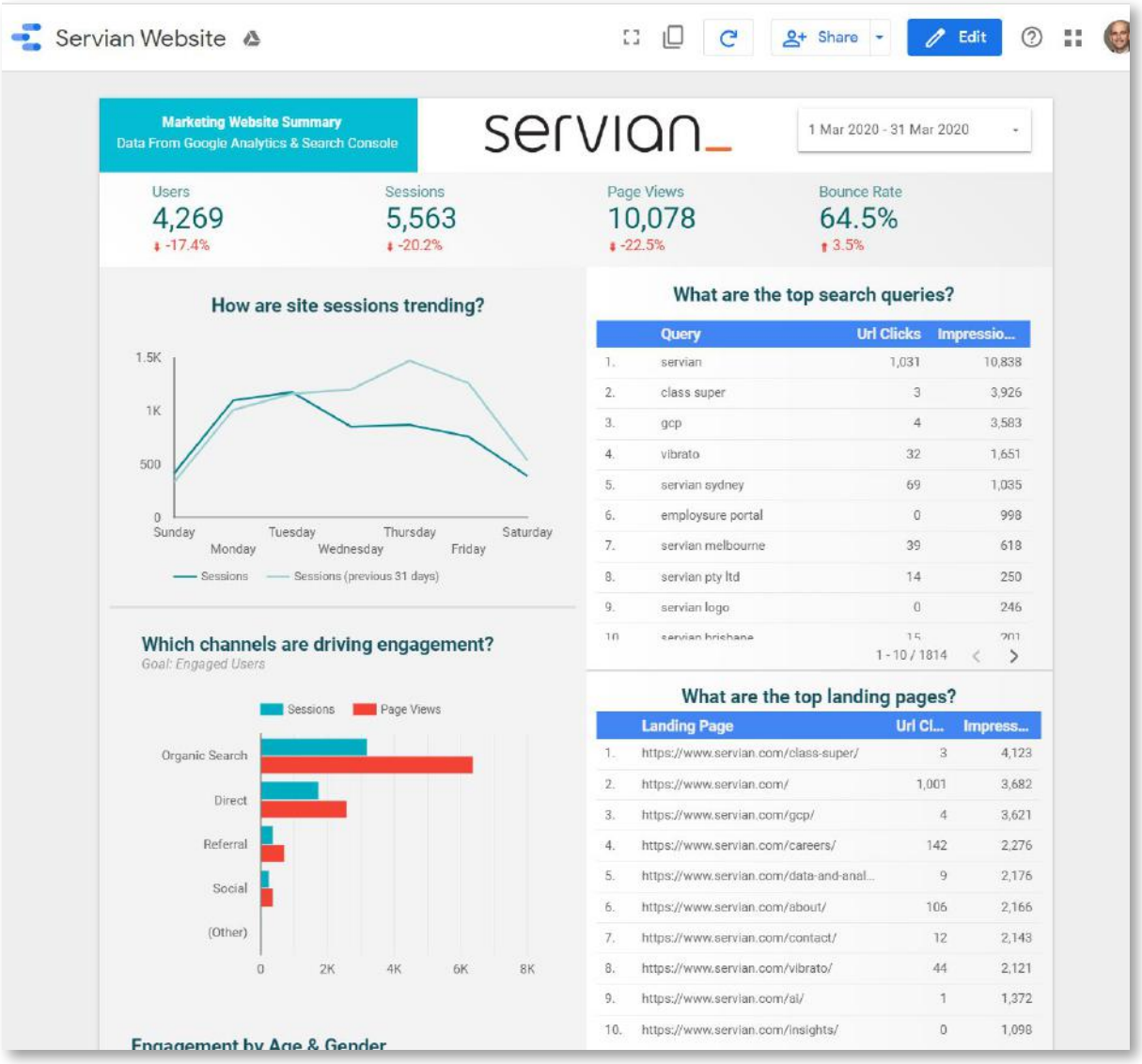
A well-designed access model which we have covered elsewhere in this document therefore should enable all staff in an organisation to have access, to help inform them of data that helps them make informed business decisions.

3.6.1 Data accessibility tools

Google Cloud provides a range of tools, all hosted and available to be accessed via a browser. These tools can all access data which resides in BigQuery.

Tool Class	Google Cloud tool	Other comparative tools	User groups
Basic Dashboarding	Data Studio Google Sheets		All Staff
Advanced Dashboarding & Reporting	Looker	Tableau	Executive users Business Insight Analysts
Data Science development	AI Platform Notebook	Juypter Notebooks	Data Scientists
Data Engineering Development	BigQuery tabs in Google Cloud Console	MS Visual Code/ IntelliJ Talend Informatica IBM	Data Engineer Technical Business Analysts

3.6.2 Data Studio



The above screenshot comes from Servian's usage of Google Cloud Data Studio looking at GA360 data. The impacts of COVID19 are evident with the change in website sessions late in the week. Our Class Super use case also seems to be picking up traffic as people are more interested in the superannuation related aspects of the Australian government response, so our SEO is being effective, but not necessarily with the audience we are after.

Data Studio is a great way for distributing key data points across an organisation.

It can leverage views in BigQuery (including BI Engine) which are cached to provide appropriate information. These views can have enterprise access controls used to restrict access.

However, Data Studio is not a replacement for advanced dashboarding or reporting tools as its component palette and data capabilities do have limitations.

4. Building a Cloud Data Blueprint

While many organisations consider a solution architecture for a cloud data platform, we recommend considering a blueprint approach, which includes looking at the program's delivery structures, operational structures, and people/process/technology factors through different phases of developing a platform.

4.1 Principles of constructing a blueprint

The following are principles to consider when building a blueprint for an organisation:

- What are the key drivers for establishing the new platform?
- Who are the primary stakeholders? Data platforms are typically always shared by their stakeholders.
- What are the security and risk requirements for the platform and for operating in the cloud?
- Have all the sources and target systems been detailed?
- What is the definition of 'complete'?
- Have the enterprises key data entities and related sources of truth been identified?

4.2 People, process and technology considerations

Each cloud provider has their own data capabilities and commercial model. How does an organisation evolve its operating model to be able to benefit from this variation, while also delivering a level of coherence and consistency across their service offerings? There are a number of concerns to consider when designing a program. These include:

1. People - are the skills and structures ready at the right time?
    - What are the skills needed to operate in a cloud data platform?
    - What training and training approaches work for your staff?
    - What documentation is needed to empower staff?
    - Do you have the right mix between technical business analysts, data engineers and data scientists?
    - Is your program delivery structure set up correctly for the phase you are in?
        - Functional teams to deliver initial components
        - Cross-functional to deliver scale and end-to-end business outcomes
        - Platform teams to work ahead, laying the tracks for other squads

2. Process - how are technology and data-related processes able to leverage cloud based solutions?
    - What are the paths to production for 'Gold' (highly conformed,high quality data)?
    - How will data discovery and data science be supported with the flexibility required?
    - What will the path to production look like for data discovery and data science features which show benefit?
    - How much automation will be used, for deployment, for testing, for different levels of change impact? How will service transition work with a high level of automation?

3. Technology - what technology is going to be used to provide the end to end solution?
    - Do you have a preference for alignment with existing skills? If so, which?
    - Do you have a preference for a GUI tools based approach or coding based approach?
    - What are your procurement and cost preferences for licensing and consumption?

4.3 Foundation cloud capabilities

All cloud capabilities require adequate base foundations to be put in place.

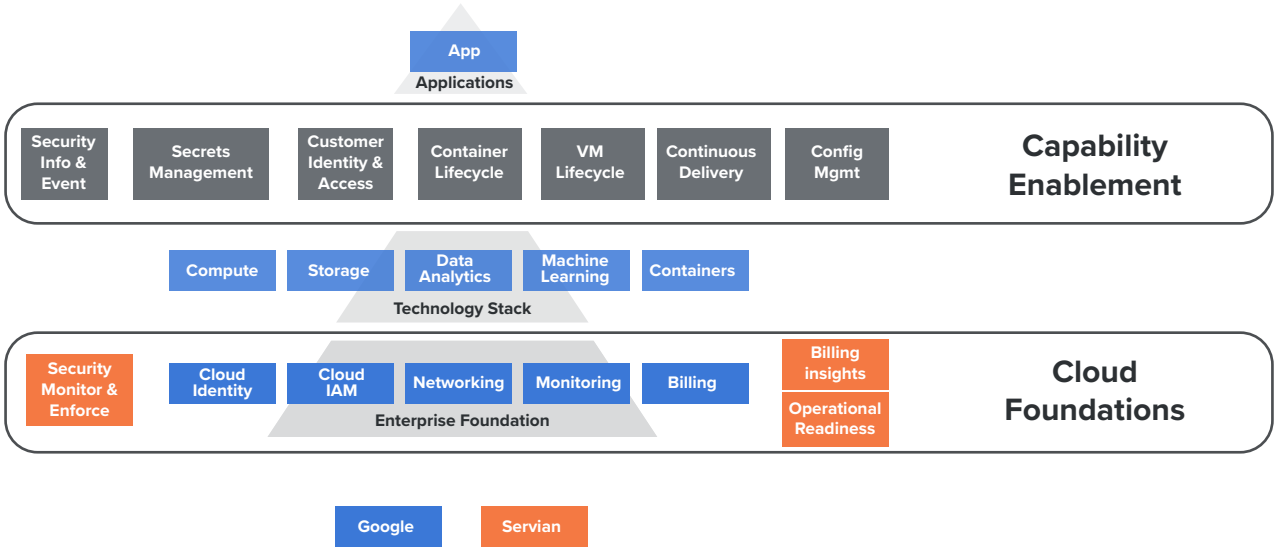
The fundamental building blocks are identity and access management, setting up networks, monitoring and managing the cost base.

Cloud foundations should be established prior to building a cloud data platform.

Servian's Cloud Blueprint highlights that all the same disciplines are required in a cloud world, however, there are differences in the application of each discipline in public cloud, in addition to the tooling and other differences across each public cloud.

Data-related capabilities that are also needed for a high maturity level of operating include:

- Proactive SIEM capabilities
- Secret management to handle database username/password obfuscation patterns and the life cycles for token and public key cryptography
- Leveraging containers with a secure, immutable life cycle to support a variety of continually improving and updating data and analytical libraries and capabilities, e.g. Python, R, Spark
- Continuous delivery capabilities to deploy, rollback code and configuration
- Configuration management to deploy and manage the life cycle of configuration changes across distributed systems



5. Building a Roadmap

5.1 Estimates

Building estimates for a data platform is the same as any other engineering undertaking.

Criteria such as experience, skill level, capacity, priority, and scope all need to be factored in.

We have found the work by Renertsen and Leffingwell (which is the basis for Scaled Agile) has stood the test of time. Some key points to approach estimation include:

- Leverage multiple point of views for each item being estimated
- Focus on estimating the complexity of tasks and consider using story points as a currency for this
- Allocate the estimates to the project tasks/user stories and track their accuracy
- Re-factor the duration and effort estimates against the complexity estimates over time as you get feedback
- Complexity estimates should also be refined as programs progress. If frameworks are established, the time to ingest and stage data on the platform should be reduced
- Ensure an appropriate level of contingency is included

5.2 Sequencing

Sequencing a program of work is an art form in balancing priorities, requirements and dependencies.

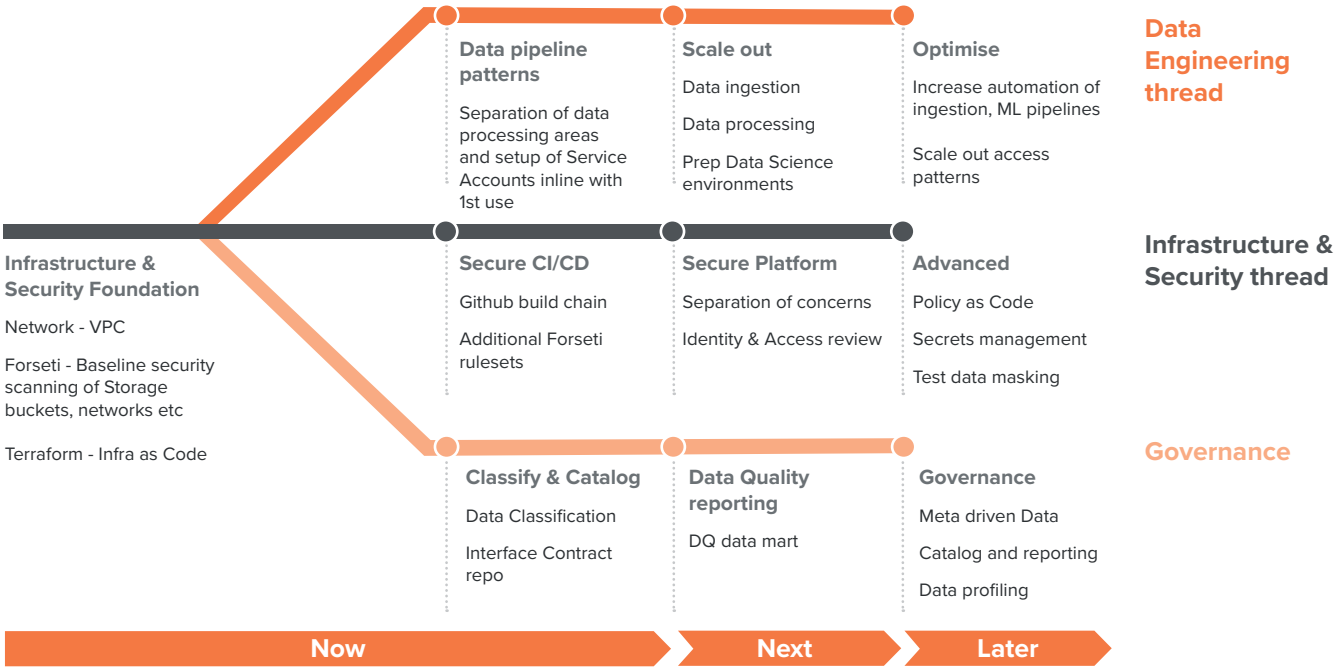
We recommend considering:

- Having a very clear understanding of technical skill and people dependencies
- Having a very clear understanding of critical path activities
- Considering using Weighted Shortest Job First as a default approach for sequencing
- Undertake technical spike activity to reduce implementation risks that are conducted ahead of a main program of work
- Build a steel thread, end-to-end through the solution before scaling out. For example, build a batch pipeline and a streaming pipeline, then automate and scale out.
- Ensure that data ingestion and data sources have provided adequate information, and test data to build out ingestion steps as a priority within the program

Many of these suggestions fit neatly with disciplined Scaled Agile approaches already mentioned.

5.3 Example roadmap

The following is an example roadmap based on the sequencing that Servian has observed across multiple industries. It grows an organisation's information and data maturity while also increasing the pace of delivery and lowering the time to insights.



6. Appendix

6.1 Australian data risk and security references

APRA

- Information Paper - outsourcing involving shared computing services (inc cloud)
- Prudential Standard - Outsourcing CPS231
- CPG 235 Managing Data Risk
- Outsourcing involving cloud computing

- Australian Privacy
  - Australian Privacy Principles (APP)
  - Personal Information for direct marketing (APP 7)
    - Cross-Border disclosure of Personal (APP 8)
    - Adoption, use and disclosure of government related identifiers (APP 9)
  - Privacy Act 1988
  - Spam Act 2003
  - Do Not Call Register Act 2006
- Consumer Data - emerging industry data standards - Banking | Energy | Telco
  - ACCC Consumer Data Right - https://www.accc.gov.au/focus-areas/consumer-data-right-cdr-0
  - https://consumerdatastandards.org.au/
- OAIC - Office of the Australian Information Commissioner
  - Guide to Data Analytics and the Australian Privacy Principles
  - De-identification Decision-Making Framework
  - De-identification and the Privacy Act
  - Telecommunications
  - Privacy & record keeping in Telecommunications
  - Healthcare identifiers
- TSSR - Telecommunications Sector Security Reforms
  - Telecommunications Sector Security Reforms Administrative Guidelines
  - TSRR Revised Draft Legislation
- ISO27001 and related ISO2700x standards
  - https://www.iso.org/isoiec-27001-information-security.html

6.2 Google Cloud security and compliance references

Security

- Encryption at rest - https://cloud.google.com/security/encryption-at-rest/default-encryption/
- Encryption in transit - https://cloud.google.com/security/encryption-in-transit/

Compliance

- Compliance references - https://cloud.google.com/security/compliance/
- Australian Privacy Principles - https://cloud.google.com/security/compliance/australian-privacy-principles/
- Financial Services, APRA - https://cloud.google.com/security/compliance/apra/
- Google Cloud Control mapping to APRA CPG 234 and CPG235 controls are available on request
- Government IRAP - https://cloud.google.com/security/compliance/irap/

6.3 Recommended references

Google

- BigQuery for data warehouse practitioners
  - https://cloud.google.com/solutions/bigquery-data-warehouse
- BigQuery - controlling access to views
  - https://cloud.google.com/bigquery/docs/view-access-controls
- BigQuery best practices - denormalize
  - https://cloud.google.com/bigquery/docs/best-practices-performance-input#denormalize_data_whenever_possible
- Dataflow - deploying a pipeline
  - https://cloud.google.com/dataflow/docs/guides/deploying-a-pipeline#streaming-engine
- Apache Beam built in I/O transforms
  - https://beam.apache.org/documentation/io/built-in/
- Dataflow patterns
  - https://cloud.google.com/blog/products/gcp/guide-to-common-cloud-dataflow-use-case-patterns-part-1
  - https://cloud.google.com/blog/products/gcp/guide-to-common-cloud-dataflow-use-case-patterns-part-2

7. About Servian

Mission

Servian is focused on enabling its clients to build competitive advantage and maximise ROI on their technology investment by:

- Providing thought leadership and innovation to deliver practical and best practice advice and solutions that address many of the key business challenges faced by its clients
- Allocating experienced, quality consultants who work collaboratively with its clients to meet outcomes and exceed business expectations
- Transforming its clients information landscapes with new technology-driven interactions that help them build trust both internally and externally
- Driving a cloud-first mindset that builds collaboration across its clients' entire ecosystems to help them build best-in-class products and services

Servian is committed to providing excellence, with quality of delivery being a key metric in achieving the desired customer satisfaction and delivering efficiencies, cost-effectiveness and performance enhancement over the life of the client relationship.

History

Servian is a professional services organisation which provides IT advisory, consulting and managed services to clients seeking to use their data better to drive business performance. Founded in 2008, Servian has grown to become the largest pure play participant in the Australian information management consulting market with over 500 consultants across ten main offices.

Servian has a diverse enterprise and corporate customer base that includes over 200 clients across government, finance and banking, telco, utility, insurance, construction, airline and retail sectors. The organisation is highly referenceable among clients, known for both our technology thought leadership and the quality of our delivery. It operates within the $12B Australian IT services industry.

Servian is platform agnostic and can implement technology solutions across any data, digital and cloud environment (including Google, Amazon and Microsoft).

Use cloud and data to gain competitive advantage. Get in touch today.

We are experienced in delivering solutions across many industries such as banking, retail, telecommunications, insurance and utilities. Our clients include many of Australia's leading Tier 1 companies as our valued customers.

sydney
Level 46, 264 George Street
Sydney NSW 2000
t +61 2 9376 0700

melbourne
Level 20, Tower 5, 727 Collins Street
Docklands VIC 3008
t +61 3 9081 3700

brisbane
Level 3, 200 Mary Street
Brisbane QLD 4000
t +61 7 3193 3200

adelaide
Level 1, 5 Peel Street
Adelaide SA 5000
t +61 414 458 763

canberra
Suite 2, 6 Napier Close
Deakin ACT 2600
t +61 457 345 536

hobart
Level 2, 162 Macquarie St
Hobart, TAS 7000
t +61 402 658 878

auckland
Level 22, Crombie Lockwood Tower,
191 Queen Street,
Auckland NZ 1010
t +64 9 918 0580

wellington
Level 1, 139 The Terrace
Wellington 6011
t +64 4 499 6988

london
Uncommon, 34-37 Liverpool Street,
London, EC2M 7PP
t +44 (0)20 8092 5231

bengaluru
Level 2, Plot 23, 8th Main Road
Jayanagar 3rd Block
Bengaluru, India 560 011
t +91 80 4370 4670

servian.com