






















Fundamentals of Data engineering Chapter 4

☑ Favorite	<input type="checkbox"/>
☑ Archived	<input type="checkbox"/>
☑ Fleeting	<input type="checkbox"/>
↗ Area/Resource	 <u>Data engineering</u>
↗ Project	
▼ Type	
📅 Review Date	
📎 Image	
🔗 URL	
🕒 Created	@March 13, 2023 12:07 PM
🕒 Updated	@March 13, 2023 2:38 PM
🔍 Root Area	https://www.notion.so/59a9ea296b924e64af4b69632f2dc92f
🔍 Project Area	
Σ Updated (short)	03/13/2023
↗ Pulls	
🔍 Resource Pulls	
🔍 Project Archived	
Σ URL Base	
Σ 🔍 Recipe Divider	   RECIPE BOOK PROPERTIES   
☰ 🔍 Recipe Tags	
Σ 📖 Book Divider	   BOOK TRACKER PROPERTIES   
☰ 📖 Author	

  Date Started	
  Date Finished	
  Book Status	
  Rating	



This chapter is concerned with choosing technologies across the data engineering lifecycle. It is easy to get caught up in chasing bleeding-edge technology while losing sight of it's core purpose which is designing robust and reliable systems to carry data through the full lifecycle and serve it according to the needs of end users.

Architecture first design



We should never design our architecture based on tools but based on the value added to the end user. The Architecture is the what, why and when whereas the tools are the how.

The key considerations when choosing tools are:

- Team size and capabilities
- Speed to market
- Interoperability
- Cost optimization and business value
- Today versus the future: immutable versus transitory technologies.
- Location
- Build versus buy
- Monolith versus modular
- Serverless versus servers

- Optimization, performance the benchmark wars
- The undercurrents of the data engineering lifecycle

Team size and capabilities



The team size and their capabilities is important in determining the complexity of the tools you adopt to produce the architecture you designed.

Small teams



Small teams should put off as much technical complexity and focus on providing value by using managed solutions and SaaS. Your limited bandwidth should be dedicated on adding value through their tools not managing them.

Large teams



Large teams with more in dept and diverse skill sets can benefit from more complex tools as they have more bandwidth to handle the effects of managing them.

Speed to Market



Speed to market always wins in technology. Choosing the right technologies that help you deliver features and data faster while maintaining high quality standards and security is crucial. It also means working in a tight feedback loop of launching, learning, iterating and making improvements. Deliver value early and often. Choose tools that help you move quickly, reliably, safely and securely.

Interoperability



Interoperability describes how various technologies or systems connect, exchange information, and interact.



How easily do tools interact with the rest of your tools. There is often a spectrum of difficulty ranging from seamless to time-intensive. Do you need a lot of effort to integrate technologies? You should always be aware of how simple it is to connect your various technologies across the data engineering lifecycle.



You should design for modularity and giving yourself the ability to easily swap out technologies as new practices and alternatives evolve.

Cost optimization and business value



Your organization expects a positive ROI from your data projects, so you must understand the basic costs you can control.

Total Cost of Ownership (TCO)



TCO is the total estimated cost of an initiative, including the direct and indirect costs of products and services utilized.

Direct costs



Direct costs are directly attributed to an initiative. Such as the salaries of the workers involved or the bills for the services consumed.

Indirect costs



Indirect costs or overhead are independent of the initiative and must be paid regardless of where they're attributed.



How something is purchased impacts the way costs are accounted for. Expenses fall into two big groups: Operating expenses, and capital expenses.

Capital expenses, CAPEX



Capital expenses: Require an upfront investment. Payment is required today. Before the cloud existed most expenses were capex for data processing as you would typically purchase hardware and software up front.

Operational expenses, OPEX



Operational expenses: Gradual expenses that are spread out over time. Whereas capex is long-term focused, opex is short term. Opex can be pay as you go and offers a lot of flexibility. Opex is closer to a direct cost making it easier to attribute to a data project.



With the advent of the cloud, data platform services allow engineers to pay on a consumption-based model. Opex allows for far greater ability for engineering teams to choose their software and hardware. Given the flexibility engineers should take an opex-first approach centered on the cloud and flexible pay-as-you-go tech.

Total Opportunity Cost of Ownership (TOCO)



Total opportunity cost of ownership is the cost of lost opportunities that we incur in choosing a technology, architecture or process.

- If you choose a tool A you are forgoing the benefits of using tool B
 - You're committed to tool A and everything it entails
 - What happens if tool A becomes obsolete?

FinOps



The goal of FinOps is to fully operationalize financial accountability and business value by applying the DevOps like practices of monitoring and dynamically adjusting systems.

If it seems that FinOps is about saving money, then think again. FinOps is about making money. Cloud spend can drive more revenue, signal customer base growth, enable more product and feature release velocity, or even help shut down a data center.

Today versus the future: Immutable versus transitory technologies



You should focus on the present and near future when choosing technologies but also in a way that allows you to handle future unknowns and evolution. Ask yourself where are you today and where would you like to be in the future? These answers should inform decisions about your architecture. This is done by **understanding what is likely to change and what tends to stay the same.**

Immutable technologies



Immutable technologies are components, languages or paradigms that have stood the test of time. Such as S3 object storage. These technologies benefit from the Lindy effect which states that the longer a technology has been established the longer it will be used.

Transitory technologies



Transitory technologies are those that come and go. They typically come with a lot of hype, followed by meteoric growth and a slow descent into obscurity.



Identify immutable technologies from transitory ones every two years. Once you identify immutable technologies use them as your base and build transitory tools around the immutables. You should also consider how easy it is to transition from a chosen technology.

Location

- Premises
- Cloud
- MultiCloud
- Hybrid

Premises



Most startups are born in the cloud, and on premises is still the default for established companies. When migrating to the cloud it is important that companies have an understanding of the opex first approach of cloud pricing.

Cloud



Users are able to dynamically scale resources that were inconceivable with on-premises servers. This dynamic scaling makes cloud models extremely appealing to agile teams.

- IaaS
- PaaS
- SaaS

IaaS



Infrastructure as a service are products such as VMs and virtual disks that are essentially rented slices of hardware. Slowly we have seen a shift towards platform as a service (PaaS)

PaaS



PaaS includes IaaS products but adds more sophisticated managed services to support applications. They allow engineers to ignore the operational details of managing individual machines and deploying frameworks across distributed systems. Providing turnkey access to complex, autoscaling systems with minimal operational overhead.

- DataBricks

SaaS



SaaS offerings move one step up the ladder of abstraction. They provide a fully functional enterprise software platform with little operational management.

- Salesforce
- Microsoft 265



Enterprises that migrate to the cloud often make major deployment errors by not appropriately adapting their practices to the cloud pricing model.

Cloud economics



Cloud services are similar to financial derivatives. Cloud providers use virtualization to sell slices of hardware, but also sell these pieces with varying technical characteristics and risks attached. There are massive opportunities for optimization and scaling by understanding cloud pricing.

Cloud providers on risk



Cloud vendors deal in risk. They offer services that have certain technical characteristics that you anticipate you will need. For example archive storage. The risk is that while storage is cheap, if you ever need the data you will pay a high cost to retrieve it.



Rather than charging for CPU cores, or other features, cloud providers charge based on characteristics such as durability, reliability, longevity and predictability. Compute platforms offer workloads that are ephemeral or can be interrupted where capacity is needed elsewhere.

Curse of familiarity



Many products are intentionally designed to look like something familiar to facilitate ease of user and accelerate adoption. But any new technology product has subtleties and wrinkles that users must learn to identify, accommodate and optimize.

Increase business value



We should aim to increase business value by leveraging the dynamic value of the cloud.

Data gravity



The concept that cloud providers will lock you in through high egress fees, and other fees that make it expensive to swap providers.

Location Part 2

Hybrid cloud



As companies may believe they have achieved operational excellence in some areas of their own prem solutions they may only want to migrate certain workloads to the cloud.

Multicloud



Deploying workloads to multiple public clouds. Data intensive applications may want to use multi cloud solutions where network latency, and bandwidth limitations hamper performance.

Disadvantages



The complexity of managing the overheads relating to a multi cloud solutions usually makes it this a bad solution. A new generation of cloud of clouds tools aims to reduce this complexity by offering services across clouds, seamlessly replicating data between clouds or managing workloads across clouds from a single work pane. Snowflake makes use of this, it is an evolving trend worth paying attention to.

Advice



Have an escape plan. Preparing to move away from your current solution will make you more agile.

Managing your own hardware



Very large companies may have specific needs and the economies of scale at which managing your own hardware and network connections make sense. This usually requires massive workloads.

- Cloudflare
- Dropbox
- Netflix and their custom CDN