# Fundamentals of Data Engineering - Reis & Housley

## Part III. Security, Privacy, and the Future of Data Engineering

### Chapter 11 – The Future of Data Engineering

* Warp speed changes in the field have created a significant knowledge gap for existing DE’s, people interested in moving into a career in DE, technology managers, + executives who want to better understand how DE fits into their companies
* It certainly feels like the field of DE (and, really, all things data) is changing daily
* Sifting through the noise and finding the signal of what’s *unlikely* to changeis very challenging
* The order of operations + names of best practices + technologies might change, but the primary stages of the **DE lifecycle** will likely remain intact for many years to come
* But be keenly aware that tech continues to change at an exhausting pace, since several years ago, DE didn’t even exist as a field or job title

#### The Data Engineering Lifecycle Isn’t Going Away

* While data science has received the bulk of the attention in recent years, DE is rapidly maturing into a distinct + visible field
* It’s one of the fastest-growing careers in tech, with no signs of losing momentum (lol)
* As companies realize they first need to build a data foundation before moving to “sexier” things like AI and ML, DE will continue growing in popularity + importance
* This **progress centers around the DE lifecycle**
* Some question whether increasingly simple tools + practices will lead to the disappearance of DE’s
* But as organizations leverage data in new ways, new foundations, systems, + workflows will be needed to address these needs.
* **DE’s sit at the center of designing, architecting, building, + maintaining these systems**
* **If tooling becomes easier to use, DE’s move up the value chain to focus on higher-level work**

#### The Decline of Complexity and Rise of Easy-to-Use Data Tools

* **Simplified, easy-to-use tools continue to lower the barrier to entry for DE**
* This is a great thing, especially given the shortage of DE’s (2022-2023)
* The **trend toward simplicity will continue**
* **DE isn’t dependent on a particular technology or data size, + it’s also not just for large companies**
* In the 2000s, deploying “big data” technologies required a large team + deep pockets
* **Ascendance of SaaS-managed services has largely removed the complexity of understanding the guts of various “big data” systems**, so DE is now something that *all* companies can do
* Big data is a victim of its extraordinary success
* Ex: Google BigQuery, a descendant of GFS and MapReduce, can query PB of data
* Once reserved for internal use at Google, this insanely powerful technology is now available to anybody w/ a GCP account
* **Users simply pay for the data they store + query rather than having to build a massive infrastructure stack**
* Snowflake, Amazon EMR, + many other hyper-scalable cloud data solutions compete in the space + offer similar capabilities
* **The cloud is responsible for a significant shift in the usage of OSS tools**
* Even in the early 2010s, using OSS typically entailed downloading the code + configuring it yourself
* **Nowadays, many OSS data tools are available as managed cloud services that compete directly w/ proprietary services**
* Linux is available preconfigured + installed on server instances on all major clouds
* Serverless platforms like AWS Lambda and Google Cloud Functions allow you to deploy event-driven applications in minutes, using mainstream languages such as Python, Java, + Go running atop Linux behind the scenes
* DE’s wishing to use Apache Airflow can adopt Google’s Cloud Composer or AWS’s managed Airflow service
* Managed K8’s allows us to build highly scalable microservice architectures.
* And so on
* This fundamentally changes the conversation around OSS code
* **In many cases, managed OSS is just as easy to use as its proprietary service competitors**
* **Companies w/ highly specialized needs can also deploy managed OSS, then move to self-managed OSS later if they need to customize the underlying code**
* Another significant trend is the **growth in popularity of off-the-shelf data connectors** (as of 2022-2023, popular ones include Fivetran and Airbyte)
* DE’s have traditionally spent a lot of time + resources on building + maintaining plumbing to connect to external data sources
* **The new generation of managed connectors is highly compelling, even for highly technical engineers, as they begin to recognize the value of recapturing time + mental bandwidth for other projects**
* **API connectors will be an outsourced problem so that DE’s can focus on the unique issues that drive their businesses**
* **The intersection of red-hot competition in the data-tooling space w/ a growing number of DE’s means data tools will continue decreasing in complexity while adding even more functionality and features**
* This simplification will only grow the practice of DE, as more + more companies find opportunities to discover value in data

#### The Cloud-Scale Data Operating System and Improved Interoperability

* Let’s briefly review some of the inner workings of (single-device) OS’s, then tie this back to data and the cloud
* Whether utilizing a smartphone, a laptop, an application server, or a smart thermostat, **these devices rely on an OS to provide essential services and orchestrate tasks + processes**
* Ex: You can see roughly 300 processes running on a MacBook Pro that you’re typing on
* Among other things, see services such as WindowServer (responsible for providing windows in a graphical interface) and CoreAudio (tasked w/ providing low-level audio capabilities)
* When running an application on this machine, it doesn’t *directly* access sound and graphics hardware
* Instead, it **sends commands to OS services** to draw windows and play sound
* **These commands are issued to standard APIs, where a specification tells software developers how to communicate with OS services**
* **The OS *orchestrates* a boot process to provide these services, starting each service in the correct order based on dependencies among them + also *maintains* services by monitoring them + restarting them in the correct order in case of a failure**
* Now let’s return to **data in the cloud**
* The **simplified data services** mentioned throughout this book (e.g., Google Cloud BigQuery, Azure Blob Storage, Snowflake, + AWS Lambda) **resemble OS services, *but at a much larger scale, running across many machines rather than a single server***
* **Now that these simplified services are available, the *next* frontier of evolution for this notion of a cloud data OS will happen at a *higher* level of abstraction**
* Benn Stancil called for the emergence of **standardized data APIs for building data pipelines + data applications** (<https://benn.substack.com/p/the-data-os>)
* **One could predict that DE will gradually coalesce around a handful of data interoperability standards**
* **Object storage in the cloud will grow in importance as a batch interface layer between various data services**
* New generation file formats (such as Parquet and Avro) are already taking over for the purposes of cloud data interchange, significantly improving on the dreadful interoperability of CSV and the poor performance of raw JSON
* **Another critical ingredient of a data API ecosystem is a metadata catalog that describes schemas and data hierarchies**
* Currently, this role is largely filled by the legacy Hive Metastore
* *Expect that new entrants will emerge to take its place*
* **Metadata will play a crucial role in data interoperability, both across applications + systems and across clouds + networks, driving automation + simplification**
* We will also see **significant improvements in the *scaffolding* that *manages* cloud data services**
* Apache Airflow has emerged as the first truly cloud-oriented data orchestration platform, but we are on the cusp of significant enhancement
* **Airflow will grow in capabilities, building on its massive mindshare, + new entrants such as Dagster and Prefect will compete by rebuilding orchestration architecture from the ground up**
* This **next generation of data orchestration platforms will feature enhanced data integration and data awareness**
* **Orchestration platforms will integrate w/ data cataloging + lineage, becoming significantly more data-aware in the process**
* In addition, **orchestration platforms will build IaC capabilities (similar to Terraform) and code deployment features (like GitHub Actions and Jenkins)**
* This **will allow DE’s to code a pipeline + then pass it to the orchestration platform to *automatically* build, test, deploy, + monitor**
* **DE’s will be able to write infrastructure specifications *directly* into their pipelines**
* **Missing infrastructure and services (e.g., Snowflake databases, Databricks clusters, and Amazon Kinesis streams) will be deployed the first time the pipeline runs**
* We **will also see significant enhancements in the domain of live data (e.g., streaming pipelines and databases capable of ingesting + querying streaming data)**
* In the past, building a streaming DAG was an extremely complex process w/ a high ongoing operational burden (see Chapter 8)
* Tools like Apache Pulsar point the way toward a future in which streaming DAGs can be deployed w/ complex transformations using relatively simple code
* We have already seen the emergence of managed stream processors (such as Amazon Kinesis Data Analytics and Google Cloud Dataflow), but **we will see a new generation of orchestration tools for managing these services, stitching them together, + monitoring them**
* ***What does this enhanced abstraction mean for DE’s?***
* As already argued above, **the role of the DE won’t go away, but it will evolve significantly**
* By comparison, more sophisticated mobile OS’s + frameworks have not eliminated mobile app developers
* Instead, mobile app developers can now **focus on building better-quality, more sophisticated applications**
* We expect similar developments for DE as the **cloud-scale data OS paradigm increases interoperability + simplicity across various applications + systems**

#### “Enterprise-y” Data Engineering

* **The increasing simplification of data tools and the emergence + documentation of best practices means DE will become more “enterprisey.”**
* “Three Data Engineering Experts Share Their Thoughts on Where Data Is Headed” (2021)
* <https://betterprogramming.pub/3-data-engineering-experts-share-their-thoughts-on-where-data-is-headed-a9582aa2e7b2?gi=74ef1eba3c20>
* The term “enterprise”, for some, conjures Kafkaesque nightmares of faceless committees, endless red tape, + waterfall-managed development projects w/ constantly slipping schedules + ballooning budgets
* In short, some will read “enterprise” + imagine a soulless place where innovation goes to die
* Fortunately, we’re referring to some of the *good* things larger companies do w/ data (management, operations, governance, + other “boring” stuff)
* We’re presently living through the golden age of “enterprise-y” data management tools
* **Technologies + practices once reserved for giant organizations are trickling downstream**
* **The once-difficult parts of big data and streaming data have now largely been abstracted away, w/ the focus shifting to ease of use, interoperability, + other refinements**
* This **allows DE’s working on new tooling to find opportunities in the abstractions of data management, DataOps, + all the other undercurrents of DE**

#### Titles and Responsibilities Will Morph

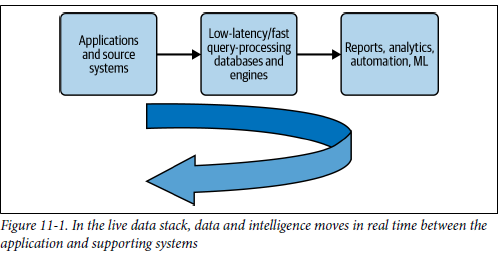
* While the DE lifecycle isn’t going anywhere anytime soon, **the boundaries between SWE, DE, data science, and MLE are increasingly fuzzy**
* Many data scientists are transformed into DE through an organic process, as they are tasked with doing “data science” but are lacking the tools to do their jobs, so they take on the job of designing and building systems to serve the DE lifecycle.
* As simplicity moves up the stack, data scientists will spend a smaller slice of their time gathering and munging data
* *But this trend will extend beyond data scientists*
* **Simplification also means DE’s will spend less time on low-level tasks in the DE lifecycle (managing servers, configuration, etc.), and “enterprise-y” DE will become more prevalent**
* **As data becomes more tightly embedded in every business’s processes, new roles will emerge in the realm of data and algorithms**
* One possibility is a role that sits *between* MLE and DE
* **As ML toolsets become easier to use + managed cloud ML services grow in capabilities, ML is shifting away from ad hoc exploration + model development to become an operational discipline**
* This new ML-focused engineer who straddles this divide will **know algorithms, ML techniques, model optimization, model monitoring, + data monitoring**
* However, their ***primary* role will be to create or utilize the systems that automatically train models, monitor performance, + operationalize the full ML process for model types that are well understood**
* They **will also monitor data pipelines + quality, overlapping into the current realm of DE**
* **MLE’s will become more *specialized* to work on model types that are closer to research and less well understood**
* Another area in which titles may morph is at the **intersection of SWE and DE**
* **Data applications (which blend traditional software applications w/ analytics) will drive this trend**
* **SWE’s will need to have a much deeper understanding of DE**
* They will develop expertise in things like streaming, data pipelines, data modeling, + data quality
* We will move beyond the “throw it over the wall” approach that is now pervasive
* **DE will be integrated into AppDev teams, + SWE’s will acquire DE skills**
* **The boundaries that exist between application backend systems and DE tools will be lowered as well, w/ deep integration through streaming and event-driven architectures**

#### Moving Beyond to Modern Data Stack to the Live Data Stack

* We’ll be frank: **the modern data stack (MDS) isn’t so modern**
* Applaud the **MDS for bringing a great selection of powerful data tools to the masses, lowering prices, + empowering data analysts to take control of their data stack**
* **The rise of ELT, cloud DW’s, + the abstraction of SaaS data pipelines certainly changed the game for many companies, opening up new powers for BI, analytics, + data science**
* Having said that, **the MDS is basically a repackaging of old DW practices using modern cloud and SaaS technologies**
* And **b/c the MDS is built around the cloud DW paradigm, it has some serious limitations when compared to the potential of next-generation *real-time* data applications**
* From some POV’s, the **world is moving beyond the use of DW-based internal-facing analytics and data science toward powering entire businesses and applications *in real time* w/ next-generation real-time databases**
* *What’s driving this evolution?*
* **In many cases, analytics (BI + operational analytics) will be replaced by automation**
* Presently, most dashboards and reports answer questions concerning *what* and *when*
* Ask yourself, “If I’m asking a *what* or *when* question, what action do I take next?”
* If the action is repetitive, it is a candidate for automation
* **Why look at a report to determine whether to take action when you can instead automate the action based on events as they occur?**
* And **it goes much further than this**
* Why does using a product like TikTok, Uber, Google, or DoorDash feel like magic?
* While it seems to you like a click of a button to watch a short video, order a ride or a meal, or find a search result, a lot is happening under the hood
* **These products are examples of true real-time data applications, delivering the actions you need at the click of a button while performing extremely sophisticated data processing and ML behind the scenes with miniscule latency**
* Presently, this level of sophistication is locked away behind custom-built technologies at large tech companies, but **this sophistication and power are becoming democratized, similar to the way the MDS brought cloud-scale DW’s + pipelines to the masses**
* **The data world will soon go “live”**

##### The Live Data Stack

* This democratization of real-time technologies will lead us to the successor to the MDS, in which the **live data stackwill soon be accessible and pervasive**
* This will **fuse real-time analytics + ML into applications by using streaming technologies, covering the full data lifecycle from application source systems to data processing to ML, and back**



* **Just as the MDS took advantage of the cloud and brought on-prem DW + pipeline technologies to the masses, the** **live data stack takes real-time data application technologies used at elite tech companies + makes them available to companies of all sizes as easy-to-use cloud-based offerings**
* This **will open up a new world of possibilities for creating even better user experiences + business value**

##### Streaming Pipelines and Real-Time Analytical Databases

* **The MDS limits itself to *batch* techniques that treat data as *bounded***
* In contrast, **real-time data applications treat data as an *unbounded*, continuous stream**
* **Streaming pipelines + real-time analytical databases are the 2 core technologies that will facilitate the move from the MDS to the live data stack**
* While these technologies have been around for some time, **rapidly maturing managed cloud services will see them be deployed much more widely**
* **Streaming technologies will continue to see extreme growth for the foreseeable future**
* This **will happen in conjunction w/ a clearer focus on the *business utility* of streaming data**
* Up to 2022-2023, streaming systems have frequently been treated like an expensive novelty or a dumb pipe for getting data from A to B
* **In the future, streaming will radically transform organizational technology + business processes, and data architects + DE’s will take the lead in these fundamental changes.**
* **Real-time analytical databases enable both fast ingestion + sub-second queries on streaming data**
* This **streaming** **data can be *enriched* or combined w/ historical datasets**
* **When combined w/ a streaming pipeline + automation (or dashboard that is capable of real-time analytics), a whole new level of possibilities opens up**
* **No longer are you constrained by slow-running ELT processes, 15-minute updates, or other slow-moving parts, + data moves in a continuous flow**
* **As streaming ingestion becomes more prevalent, batch ingestion might be less + less common**
* Why create **a batch bottleneck** at the head of your data pipeline?
* We **may eventually look at batch ingestion the same way we now look at dial-up modems**
* In conjunction w/ the rise of streams, expect a back-to-the-future moment for data transformations
* **May shift away from ELT (in database transformations) to something that looks more like ETL**
* Can provisionally refer to this as **stream, transform, and load (STL)**
* **In a streaming context, extraction is an ongoing, continuous process**
* **Of course, batch transformations won’t entirely go away**
* **Batch will still be very useful for model training, quarterly reporting, + more**
* **But streaming transformation will become the norm**
* **While DW’s + data lakes are great for housing large amounts of data + performing *ad hoc* queries, they are NOT so well optimized for low-latency data ingestion or queries on rapidly moving data**
* The **live data stack will be powered by OLAP databases that are *purpose-built* for streaming**
* Today, databases like Druid, ClickHouse, Rockset, + Firebolt are leading the way in powering the backend of the next generation of data applications
* **Expect that streaming technologies will continue to evolve rapidly + that new technologies will proliferate**
* **Another area some think is ripe for disruption is data modeling, where there hasn’t been serious innovation since the early 2000s**
* **Traditional batch-oriented data modeling techniques** (Chapter 8) **aren’t suited for streaming data**
* **New data-modeling techniques will occur *not* w/in the DW but in the systems that generate the data**
* **Expect data modeling will involve some notion of an upstream definitions layer (including semantics, metrics, lineage, + data definitions** (see Chapter 9)**), beginning where data is generated in the application**
* **Modeling will also happen at *every* stage as data flows + evolves through the full lifecycle**

##### The Fusion of Data with Applications

* We expect the next revolution will be the **fusion of the application and data layers**
* **Right now, applications sit in one area, and the MDS sits in another**
* To make matters worse, **data is created w/ no regard for how it will be used for analytics**
* Consequently, **lots of duct tape is needed to make systems talk with one another**
* This patchwork, siloed setup is awkward and ungainly
* **Soon, application stacks will be data stacks, and vice versa**
* **Applications will integrate real-time automation + decision making, powered by the streaming pipelines + ML**
* The **DE lifecycle won’t necessarily change, but the time between stages of the lifecycle will drastically shorten**
* A lot of innovation will occur in new technologies + practices that will improve the experience of engineering the live data stack
* **Pay attention to emerging database technologies designed to address the mix of OLTP and OLAP use cases, + also feature stores may also play a similar role for ML use cases**

##### The Tight Feedback Between Applications and ML

* Another area to be excited about is **the fusion of applications and ML**
* **Today, applications and ML are disjointed systems, like applications and analytics**
* SWE’s do their thing over here, data scientists and MLE’s do their thing over there
* **ML is well-suited for scenarios where data is generated at such a high rate + volume that humans cannot feasibly process it by hand**
* **As data sizes and velocity grow, this applies to every scenario**
* **High volumes of fast-moving data, coupled w/ sophisticated workflows + actions, are candidates for ML**
* **As data feedback loops become shorter, expect most applications to integrate ML**
* **As data moves more quickly, the feedback loop between applications and ML will tighten**
* **The applications in the live data stack are intelligent + able to adapt in real time to changes in the data, which creates a cycle of ever-smarter applications and increasing business value**

##### Dark Matter Data and the Rise of...Spreadsheets?!

* Something that’s widely ignored in today’s data world, especially by DE’s is that **the most widely used data platform is the humble spreadsheet**
* Spreadsheets are the dark matter of the data world
* **A good deal of data analytics runs in spreadsheets + never makes its way into the sophisticated data systems** describe in this book
* In many organizations, spreadsheets handle financial reporting, supply-chain analytics, + even CRM
* **At heart, a spreadsheet is an interactive data application that supports complex analytics**
* Unlike purely code-based tools such as Python pandas, spreadsheets are accessible to a whole spectrum of users, ranging from those who just know how to open files + look at reports to power users who can script sophisticated procedural data processing
* **So far, BI tools have failed to bring comparable interactivity to databases**
* Users who interact with the UI are typically limited to slicing + dicing data w/in certain guardrails, not general-purpose programmable analytics
* Some predict that a **new class of tools will emerge that combines the interactive analytics capabilities of a spreadsheet w/ the backend power of cloud OLAP systems**
* Indeed, some candidates are already in the running
* The ultimate winner in this product category may continue to use spreadsheet paradigms, or may define entirely new interface idioms for interacting with data

#### Conclusion

* We traversed good architecture, the stages of the DE lifecycle, + security best practices
* We’ve discussed strategies for choosing technologies at a time when the field continues to change at an extraordinary pace, as well as wild speculation about the near + intermediate future
* Perhaps the trend toward real-time data will stall once again, w/ most companies continuing to focus on basic batch processing
* Surely, other trends exist that we have completely failed to identify
* **The evolution of tech involves complex interactions of technology + culture, and both are unpredictable**
* DE is a vast topic, so continue exploration on your own
* As you discover interesting topics + ideas, continue the conversation as part of a community
* **Identify domain experts who can help you to uncover the strengths + pitfalls of trendy technologies + practices**
* **Read extensively from the latest books, blog posts, and papers**
* **Participate in meetups and listen to talks**
* **Ask questions and share your own expertise**
* **Keep an eye on vendor announcements to stay abreast of the latest developments, *taking all claims with a healthy grain of salt***
* Through this process, you can choose tech with confidence
* Next, you will need to **adopt technology + develop expertise, perhaps as an individual contributor, perhaps w/in your team as a lead, perhaps across an entire technology organization**
* As you do this, **don’t lose sight of the larger goals of DE**
* **Focus on the DE lifecycle, on serving customers (internal and external), on your business, on serving, + on your larger goals**
* **Tech trends are defined not only by those who create the underlying tech but also by those who adopt it and put it to good use**
* Successful tool *use* is as critical as tool *creation*
* **Find opportunities to apply real-time technology that will improve the UX, create value, + define entirely new types of applications**
* **It is this kind of practical application that will materialize the live data stackas a new industry standard (or perhaps some other new technology trend**)