

unravel™

GUIDE TO DATA OPS

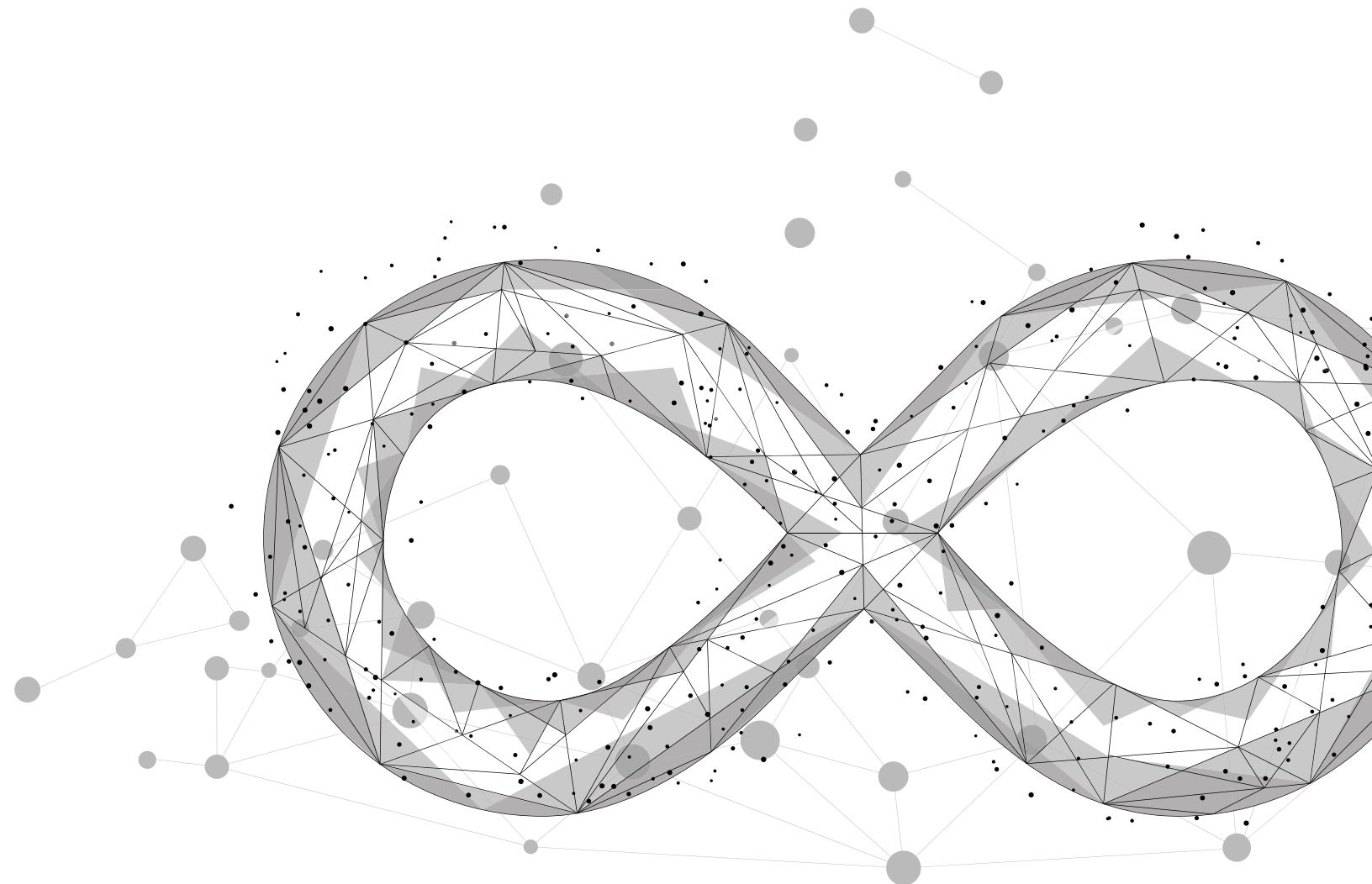




TABLE OF CONTENTS

Foreword	4
What is DataOps?	6
The Ten Steps of the DataOps Lifecycle	7
End-to-End Observability	8
Real-Time Collaboration	8
Bringing It All Together	9
Why Modern Data Stacks Require DataOps	10
Key DataOps Statistics	12
Key DataOps Findings	14
DBS Bank Automates the Verify and Deploy Stages of DataOps	16
MasterCard Optimizes its Modern Data Stack	18
84.51° and Kroger Revamp Analytics	20
Conclusion	22



FOREWORD

When we started Unravel Data several years ago, we knew there was something important emerging. The ever-increasing flood of data was becoming a topic of universal interest – in business, in government, in academia, and in society as a whole. Those who could put this flood of data to productive use – who could help achieve beneficial results, in business and in the world – would be able to contribute greatly.



And we knew that an important part of the puzzle – how to actually observe, process, manage, and optimize rivers of new data, in pursuit of valuable results – was missing. We set out to help solve that problem, and thus, Unravel Data was born.

What we didn't know was what to call the new practice that was being created by the very people we wanted to serve. These people are now called data teams. They are data professionals who have turned their attention from research – always a fruitful area, especially when it comes to data – to development, and onward from there to actual delivery of useful results.

“Operationalization” is the somewhat clumsy word for that important process. We may have even said “data operationalization,” or “data operations,” a few times, as we were standing up our new company.

We noticed the parallels between the problems found in operationalizing the work of data teams, and the emerging practice of DevOps. A new term, DataOps, was born.

The rest is history – but it's history that all of us are still creating today. DataOps can now be found in the Gartner Hype Cycle, in articles, books, podcasts, and elsewhere. More importantly, it's increasingly being used by data teams to describe themselves and their roles. Have a look around relevant groups on LinkedIn, and the CVs of the people who contribute, and you'll see what I mean.

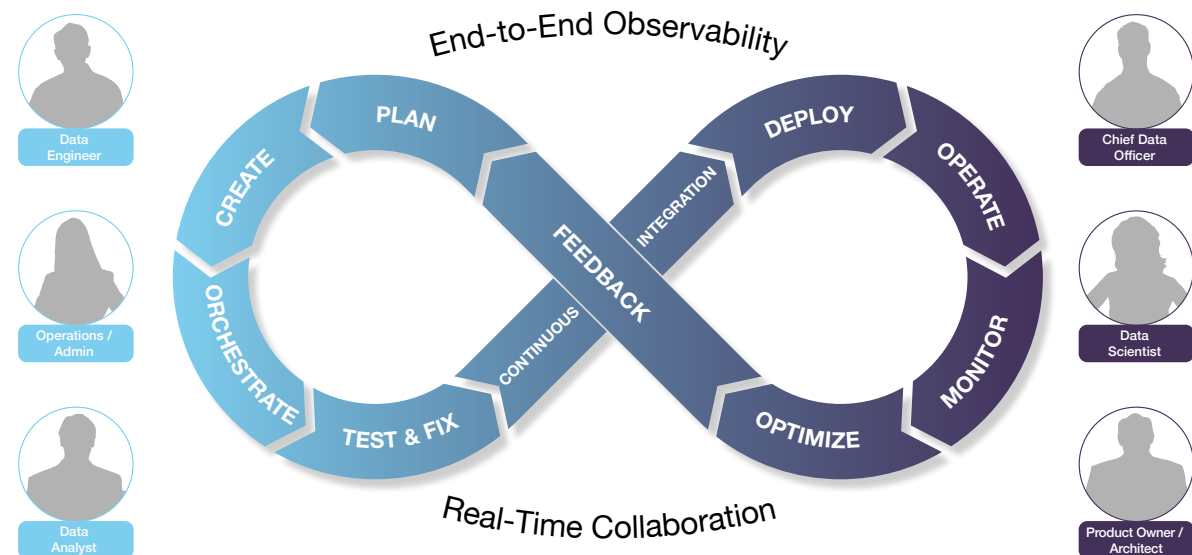
So Unravel Data was a DataOps company from the beginning, and since before the term even emerged. It's important to us to contribute to the discussion around DataOps, and how this emerging discipline can contribute to making this a better world. So we have produced this white paper, and offer it for your consideration. We look forward to the many interesting discussions that will result.

KUNAL AGARWAL
CEO, Unravel Data

WHAT IS DATAOPS?

DataOps is the use of agile development practices to create, deliver, and optimize data products, quickly and cost-effectively. DataOps is practiced by modern data teams, including data engineers, architects, analysts, scientists and operations.

The data products which power today's companies range from advanced analytics, data pipelines, and machine learning models to embedded AI solutions. Using a DataOps methodology allows companies to move fast in extracting value out of data.




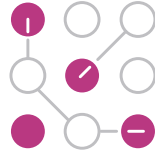
“The term DataOps and related concepts are at early stages of awareness and adoption, so many working definitions exist today.”

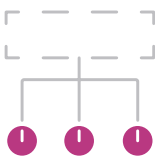
SARA QUOMA
IBM


The Ten Steps of the DataOps Lifecycle

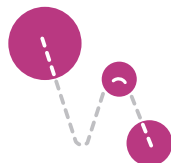
The DataOps lifecycle shown below takes data teams on a journey from raw data to insights. Where possible, DataOps stages are automated to accelerate time to value. The steps below show the full lifecycle of a data-driven application:

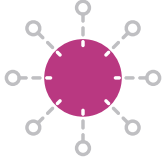
- 

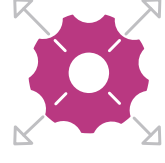
1 PLAN. Define how a business problem can be solved using data analytics. Identify the needed sources of data and the processing and analytics steps that will be required to solve the problem. Then select the right technologies, along with the delivery platform, and specify budget and performance requirements.
- 


2 CREATE. Create the data pipelines and application code that will ingest, transform, and analyze the data. Based on the desired outcome, data applications are written using SQL, Scala, Python, R, or Java, among others.
- 


3 ORCHESTRATE. Connect stages needed to work together to produce the desired result. Schedule code execution, with reference to when the results are needed; when cost-effective processing is most available; and when related jobs (inputs and outputs, or steps in a pipeline) are running.
- 


4 TEST & FIX. Simulate the process of running the code against the data sources in a sandbox environment. Identify and remove any bottlenecks in data pipelines. Verify results for correctness, quality, performance and efficiency.
- 

5 CONTINUOUS INTEGRATION. Verify that the revised code meets established criteria to be promoted into production. Integrate the latest, tested and verified code and data sources incrementally, to speed improvements and reduce risk.
- 

6 DEPLOY. Select the best scheduling window for job execution based on SLAs and budget. Verify that the changes are an improvement; if not, roll them back and revise.
- 

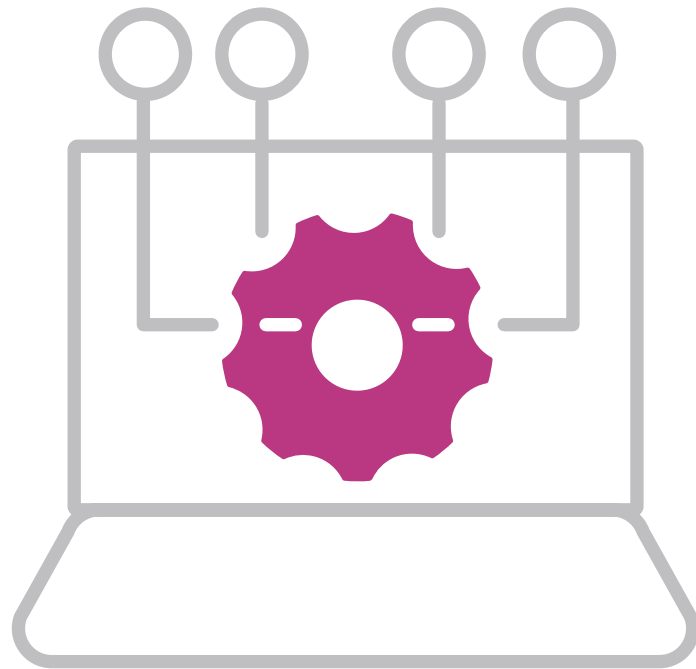
7 OPERATE. Code runs against data, solving the business problem, and stakeholder feedback is solicited. Detect and fix deviations in performance to ensure that SLAs are met.
- 

8 MONITOR. Observe the full stack, including data pipelines and code execution, end-to-end. Data operators and engineers use tools to observe the progress of code running against data in a busy environment, solving problems as they arise.
- 

9 OPTIMIZE. Constantly improve the performance, quality, cost, and business outcomes of data applications and pipelines. Team members work together to optimize the application's resource usage and improve its performance and effectiveness.
- 

10 FEEDBACK. The team gathers feedback from all stakeholders – the data team itself, app users, and line of business owners. The team compares results to business success criteria and delivers input to the Plan phase.

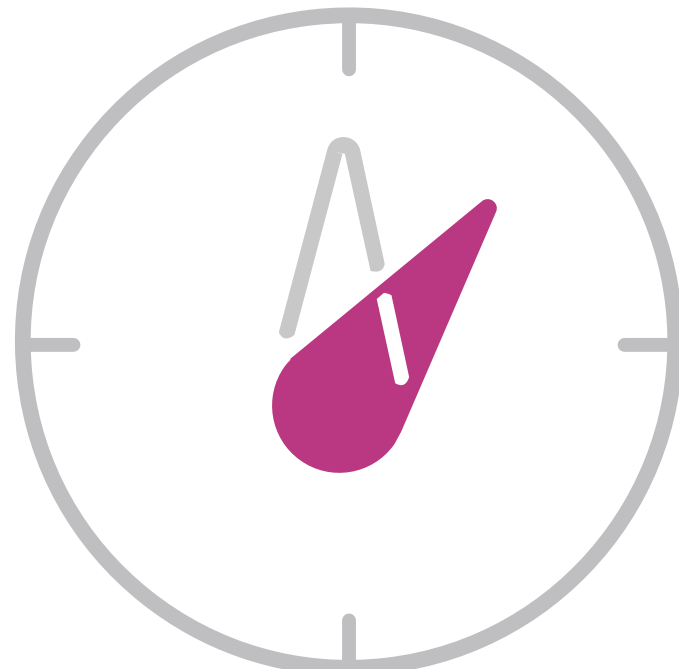
There are two overarching characteristics of DataOps that apply to every stage in the DataOps lifecycle: *end-to-end observability* and *real-time collaboration*.



End-to-End Observability

End-to-end observability is key to delivering high-quality data products, on time and under budget. You need to be able to measure key KPIs about your data-driven applications, the data sets they process, and the resources they consume. Key metrics include application / pipeline latency, SLA score, error rate, result correctness, cost of run, resource usage, data quality, and data usage.

You need this visibility horizontally – across every stage and service of the data pipeline – and vertically, to see whether it is the application code, service, container, data set, infrastructure, or another layer that is experiencing problems. End-to-end observability provides a single, trusted “source of truth” for data teams and data product users to collaborate around.



Real-Time Collaboration

Real-time collaboration is crucial to agile techniques; dividing work into short sprints, for instance, provides a work rhythm across teams. The DataOps lifecycle helps teams identify where in the loop they’re working, and to reach out to other stages as needed to solve problems – both in the moment, and for the long term.

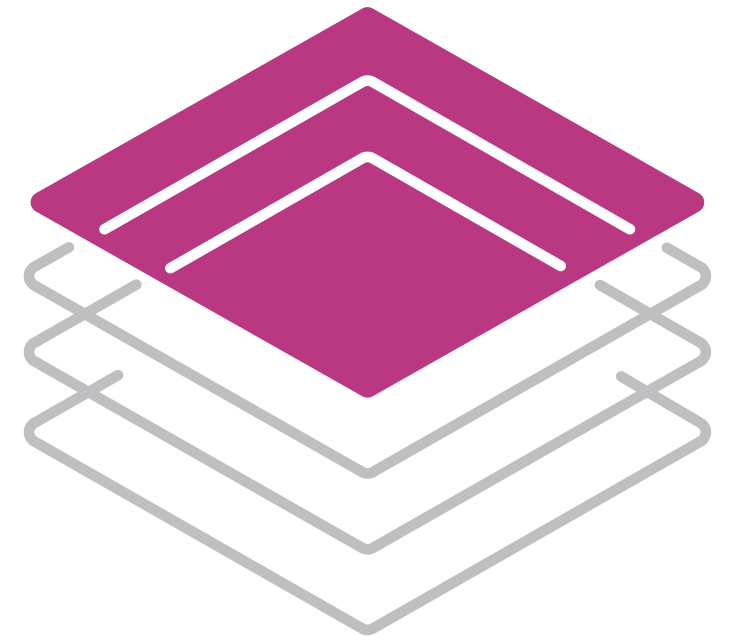
Real-time collaboration requires open discussion of results as they occur. The observability platform provides a single source of truth that grounds every discussion in shared facts. Only through real-time collaboration can a relatively small team have an outsized impact on the daily and long-term delivery of high-quality data products.

Bringing It All Together

Information from our DataOps platform, Unravel Data, is used throughout the DataOps lifecycle. Unravel is always used in the Deploy, Operate, Monitor, and Optimize phases – the right half of the diagram. And some are now integrating Unravel into their software development process for data-driven apps – the Create, Orchestrate, Test & Fix, and Continuous Integration phases. This establishes Unravel on the left half of the diagram as well. And information from Unravel is used in the Feedback and Plan stages, especially when cloud migration is underway.

It’s useful to look at each step in the DataOps lifecycle separately, but in practice, each step continually affects the others. Planners gather information between active planning phases; participants in each step are always working on the underlying, usually software-defined, infrastructure that makes everything work faster and more efficiently.

Through the use of a DataOps approach to their work, and careful attention to each step in the DataOps lifecycle, data teams can improve their productivity and the quality of the results they deliver to the organization. As the ability to deliver predictable and reliable business value from data assets increases, the business as a whole will be able to make more and better use of data in decision-making, product development, and service delivery. Advanced technologies, such as AI and machine learning, can be implemented faster and with better results, leading to competitive differentiation and, in many cases, industry leadership.



WHY MODERN DATA STACKS REQUIRE DATAOPS

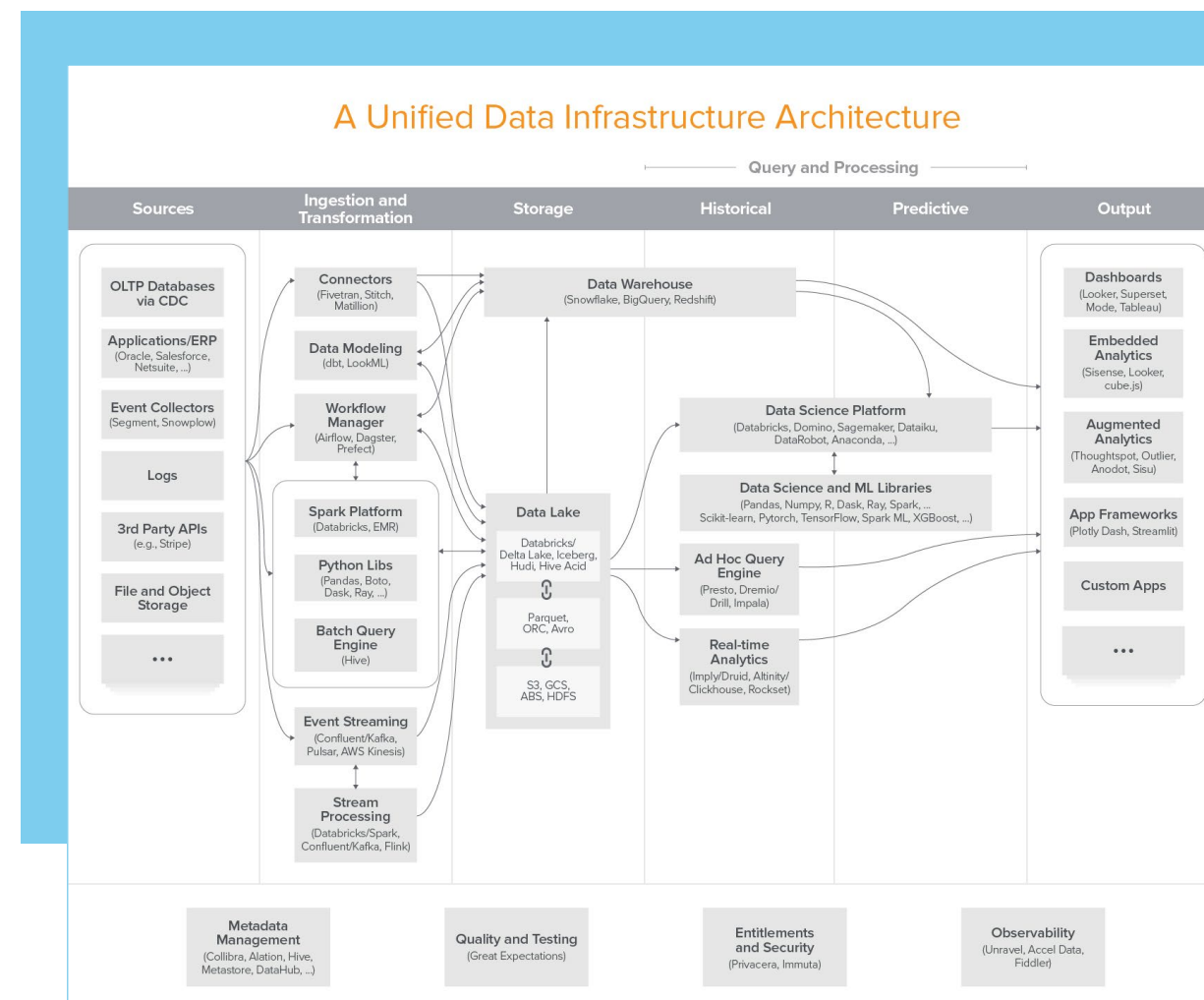
As the importance of data to organizations increases, the modern data stack continues to grow in both power and complexity. Every substantial organization processes core data for business operations through relational databases; extract, transform, and load (ETL) processes; and one or more data warehouses, for reconciliation, reporting, and research.

To take advantage of the flood of new data resources flooding in, organizations have added “big data” and streaming data, using technologies such as Spark, Presto, Kafka, Databricks, Snowflake, Hadoop, Hive, and similar tools. A whole additional set of tools is used for developing and deploying machine learning models and AI-powered applications. All of this adds to the size of the data stack.

While a picture is worth a thousand words, in this case, a single image substantially answers the question posed in the header above: Why does a modern data stack require DataOps? The answer is; because it’s very, very complex. The diagram on the right, from Andreessen Horowitz, is a reference architecture for the modern data stack. It’s also actually a simplified representation of the core data stack used by most substantial organizations.



The diagram is a simplified representation of the core data stack used by most substantial organizations.



Data infrastructure architecture with representative tools (courtesy, Andreessen Horowitz)

Though this is a simplified representation, it includes roughly two dozen different kinds of data sources, processing technologies, storage repositories, analytics approaches, and more. Its very complexity – and the additional complexity that occurs in the real world – makes the need for a unified, DataOps approach for your applications extremely clear.

Most organizations have a core data stack that is at least this complex, and many other data repositories and processing steps besides. This doesn’t even include

the widespread use of software as a service (SaaS) apps, which cutting-edge companies manage almost as carefully as if they owned the hardware, software, and data repositories themselves.

Unravel Data note: Unravel is shown under Observability in the architecture diagram. Unravel works across the entire stack. Unravel monitors the data sources on the left of the diagram and the major technology types across it, from Airflow to Spark to Databricks to BigQuery, Snowflake, and Redshift.

KEY DATAOPS STATISTICS

Last year Unravel Data had a customer event, Untold, which attracted hundreds of attendees. We surveyed them and gathered the following statistics as audience responses. While not a scientifically designed survey, the results indicate rough trends in the world of DataOps.

Key fact: Nearly half of our event attendees have ten or more technologies in production.

What percentage of your data pipelines have SLAs?

<25%	8.3%
25-50%	29.2%
>50%	41.7%
All of them	20.8%

What is usually the biggest reason for cost overruns?

Bad applications taking too many resources	75.0%
Oversized containers	20.0%
Small files	5.0%

Do you know who is the most expensive user / app on your system?

Yes	65.0%
No	25.0%
No, but would be great to know	10.0%

How often do you have rogue users/apps on your cluster?

All the time!

42.9%

Once a day

19.0%

Once a week

38.1%

One-offs:

- More than **90%** of attendees find automation helpful in preventing poor-quality apps from being promoted into production.
- More than **80%** have a quality gate when promoting apps from Dev to QA, and from QA to production.
- More than half have a well-defined DataOps/SDLC (software development life cycle) process, and nearly a third have a partially-defined process. Only about one-eighth have neither.
- About one-quarter have operations people/sysadmins troubleshooting their data pipelines; another quarter put the responsibility onto the developers or data engineers who create the apps. Nearly half make a game-time decision, depending on the type of issue, or use an “all hands on deck” approach with everyone helping.
- More than two-thirds are finding their data costs to be running over budget.
- Roughly **70%** are in the process of migrating to cloud – though only a quarter have gotten to the point of actually moving apps and data, then optimizing and scaling the result.
- Half find that automating problem identification, root cause analysis, and resolution saves them 1-5 hours per issue; the other half save from 6-10 hours or more.
- Somewhat more than half find their clusters to be on the over-provisioned side.
- Fewer than a third are following a DataOps practice in their organizations.

KEY DATAOPS FINDINGS

KEY FINDINGS FROM THE CDO **BATTLESCARS SERIES**, HOSTED BY UNRAVEL DATA CDO AND VP ENGINEERING, **SANDEEP UTTAMCHANDANI**: YOU NEED TO INCORPORATE A MULTITUDE OF DIFFERENT FACTORS, SUCH AS **COMPLIANCE, COST, ROOT CAUSE ANALYSIS, TUNING**, AND SO ON, SO THAT **DATAOPS IS SEAMLESS** AND YOU CAN AVOID SURPRISES.

IT IS IMPORTANT TO STRIKE THE **RIGHT BALANCE** BETWEEN **GOVERNANCE** AND **TIME TO MARKET**. WHEN YOU HAVE TO **MOVE FAST**, GOVERNANCE ALWAYS **SLOWS YOU DOWN**. AND GOVERNANCE DOESN'T JUST REFER TO **REQUIRED REGULATIONS** OR **COMPLIANCE**. IT'S ALSO JUST **GOOD DATA HYGIENE**, MAINTAINING THE **CATALOG**, AND MAINTAINING THE **GLOSSARIES**.

BUILDING YOUR **COMPANY'S PLATFORM** AND **SERVICES** AS A PRODUCT CAN BE **EXTREMELY BENEFICIAL** AND **PAY YOU BACK** AFTER SOME TIME. YOU NEED TIME FOR THE **INVESTMENT** TO RETURN, BUT ONCE YOU GET TO THAT STAGE, YOU'LL **GET YOUR ROI**.

DBS BANK AUTOMATES THE VERIFY AND DEPLOY STAGES OF DATAOPS

Financial services (finserv) companies are under constant pressure to improve their operational efficiency and, not surprisingly, their financial results. This leads them to become early adopters of information technology, often blazing a trail for others – in finserv, and elsewhere – to follow.

DBS Bank is the number one bank in Southeast Asia, and was named Best Bank in the World for 2020 by Global Finance. The bank is also a leader in digital transformation in the financial sector. Much of their data is stored in MariaDB and Hadoop, and processed through Spark.

DBS Bank uses Alluxio heavily for managing their workloads in production. Alluxio was used by DBS Bank as a supporting element in designing their own architecture.

DBS Bank has faced, and resolved, many of the challenges described in this case study. They have also taken a proactive approach to a classic DataOps problem: “bad” or “rogue” jobs that wreak havoc in production.

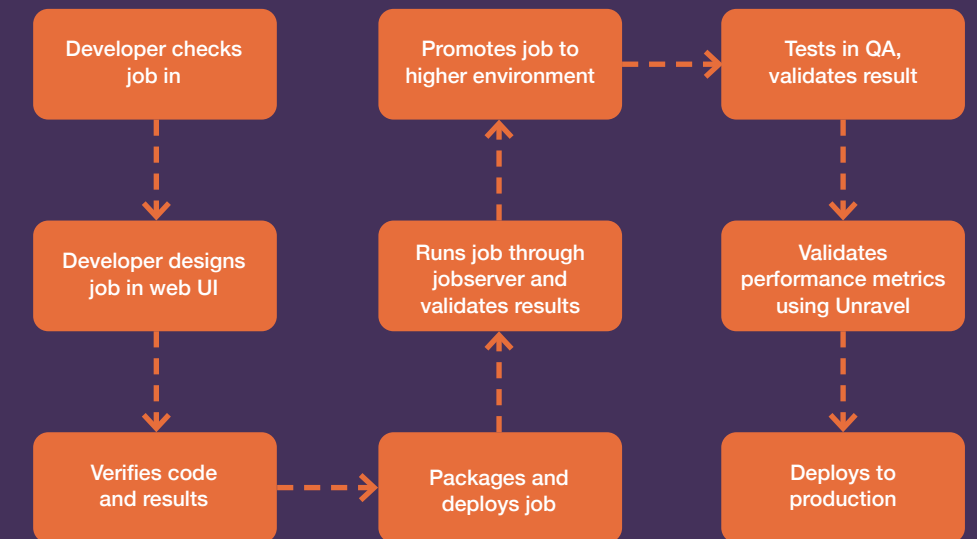
Many of these jobs are run on code that’s largely well-written and that passes standard testing routines. However, under load, the code can suddenly bog down, taking an inordinate amount of time and resources. Such jobs are costly in two ways: they use expensive resources out of proportion to their results, and they crowd out other jobs, for processing time and for operations attention.

“

Building our platform and our services as a product has been extremely beneficial... You’re gonna get your ROI.

MATTEO PELATI
HEAD OF TECHNOLOGY, DBS BANK

PROCESS FLOW



DBS Bank workflow for code acceptance using Unravel Data

DBS Bank came up with a novel approach: they stress-test new code before they put it into production. Many jobs must therefore be revised before they can be approved. However, this approach has several benefits:

- Developers can run the stress tests themselves. Fixes are made quickly and efficiently, before operators even see the code in production.
- Operations has many fewer problems to deal with. Problems are solved from the “bottom up” – before code goes into production – and from the “top down,” by optimization after going into production. The two approaches have a pincer effect, so only high-quality software remains in production.
- Developers get better at creating efficient software. They learn from both the stress test results and from the infamous calls in the middle of the night, when a job fails to stay within its resource envelope or crashes. They learn from both kinds of experience and steadily produce better code, making everyone’s life easier.

DBS has made Unravel Data part of their software development life cycle (SDLC). They have integrated their web UI for software acceptance with Unravel, and in user acceptance testing (UAT), then again in QA, they use Unravel to characterize the job – compute resource usage, storage usage, etc. Only jobs that meet stated criteria are promoted.

The DataOps team benefits in an additional way. They can use the knowledge gained from UAT and QA to understand a job better. Particularly demanding jobs can be run in an isolated environment, during quiet times, or only when the code author is available to help troubleshoot any problems. More jobs complete on time, and everyone is happier.

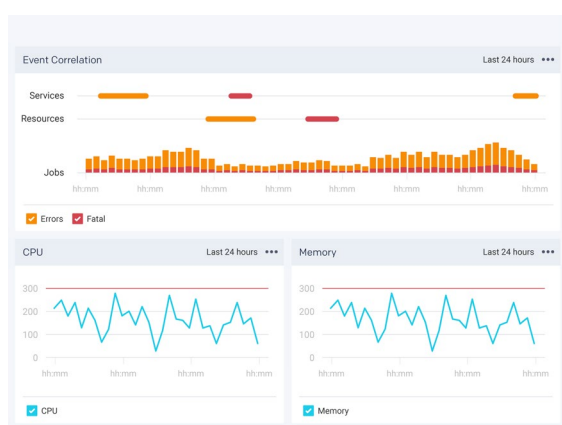
MASTERCARD OPTIMIZES ITS MODERN DATA STACK

Credit card processors Mastercard have more need to optimize their data clusters than most. Initially, technology was used to help credit card companies process transactions – and look for fraud, the misuse of stolen cards and credentials – on a batch basis, overnight. It was possible for fraudsters to rack up several transactions before a card could be cancelled, causing great stress and expense for cardholders and credit card issuers alike.

Now, fraud checks are performed “on the swipe” – in a fraction of a second, while the cardholder is at the point of sale (POS) terminal. This requires massive data processing capability. And the more data that can be processed in that fraction of a second, the greater the ability of the processing company to reject bad transactions and accept good ones.

Cloudera provides a reference architecture for secure data transactions, with credit cards serving as the very definition of the payment card industry (PCI) security standards used to protect personal financial information. Data is secured using LDAP and Kerberos for authorization and authentication, and a certificate authority for verification.

- Users were frustrated; sometimes jobs would work, other times not
- Constant high CPU/Memory allocation
- Simple queue structure replaced by business unit-based queuing with predetermined resources, enforced with entitlements
- Small files spinning up 50k mappers for a single query were identified and redesigned; admission control is now enforced
- Large batch job schedules were revised to spread load



Unravel helps to maximize cluster utilization and reduce contention

In addition, Mastercard runs a wide range of business reporting and related applications from their stack. In mid-2019, Mastercard found itself facing a fairly typical set of problems common to those who practice DataOps at scale. These problems included:

- Significant spending on additional nodes for more storage and compute capacity
- Overallocation of memory and CPU, outstripping actual usage
- Many man-hours spent troubleshooting failed jobs, slow jobs, cluster issues
- DataOps engineering team overloaded with tickets, making support hard to get

Users were frustrated. A given job might work one time, then not work the next time, leaving recipients waiting. The issues were often due to resource contention, which is by nature variable, so DataOps often could not reproduce, nor solve the problems.

Name	Created By	Last Edited By	Created	Last Run	# Run	Actions
Long running Hive query	admin	admin	09/22/2020 08:55:34	09/22/2020 09:55:53	5	[edit] [trash] [refresh] [dropdown]
Long running impala query	admin	admin	09/22/2020 07:07:00	09/22/2020 08:54:23	2	[edit] [trash] [refresh] [dropdown]
Long running workflow	admin	admin	09/21/2020 18:05:24	09/22/2020 06:56:14	13	[edit] [trash] [refresh] [dropdown]
Rogue impala application (HDFS READ/WRITE)	admin	admin	09/21/2020 17:50:26	09/22/2020 07:30:27	2	[edit] [trash] [refresh] [dropdown]
Resource contention in queue (allocated memory)	admin	admin	09/21/2020 17:41:59	09/21/2020 17:46:46	1	[edit] [trash] [refresh] [dropdown]

Unravel Data offered a range of techniques for resolving these, and other issues:

- A “heat map” showing heavy and light workload processing by day and time of day. This made it easy to move less time-critical tasks to quieter periods, reducing resource contention and crashes.
- TopX reports showing the top resource-using jobs by duration, disk I/O, CPU consumption, memory usage, and other metrics. DataOps was able to target the most intensive jobs for improvement and rescheduling, reducing total resource usage and contention.
- AutoActions allow Mastercard to trigger alerts, and more complex responses, against heavy resource use, contention, long-running jobs, etc. Operators can intervene before job completion is threatened. Users can even delay their own troublesome job before it threatens other workloads.
- AI-powered recommendations. Unravel proactively recommends changes to right-size configurations, adjust resource consumption in real time, and more. Both long-term and short-term changes result, making the entire stack more flexible.

The results have improved operations at Mastercard, at every level:

- The company’s DataOps culture is strengthened as users inside and outside IT see improved results. Communication has improved, and trust has grown.
- The existing infrastructure can process more jobs, faster, more reliably, and with less strain.
- New jobs can be added within existing hardware, software, and cloud estates and budgets. New budgetary commitments are only requested when needed, avoiding overspending.
- Reduced tension in DataOps and elsewhere in the organization. *“I am sleeping a lot easier than I did a year ago,”* says Bob Jackson of Mastercard. This improvement is reflected throughout the IT organization, helping the team spend less time and energy on emergencies, and more on creativity, infrastructure improvements, and forward planning.

These changes were critical to Mastercard’s ability to process ground-breaking numbers of transactions during the Covid-19 pandemic, as card transaction numbers skyrocketed. And they leave Mastercard with the ability to be much more responsive going forward. *“There’s no doubt that Unravel played a huge part in what our people were able to do this year,”* says Bob Jackson.

“I am sleeping a lot easier than I did a year ago.”

BOB JACKSON
DIRECTOR – BIG DATA ENGINEERING, MASTERCARD

84.51° AND KROGER REVAMP ANALYTICS

84.51° is the marketing services arm of retail giant Kroger, providing real-time insights and strategy guidance to the parent company and to more than 1,400 leading consumer packaged goods (CPG) clients. Born as legendary analytics firm dunnhumby, 84.51° uses predictive analytics, AI, and machine learning to convert extensive transaction data into actionable insights, such as state-of-the-art consumer loyalty programs.



We can invest the dollars that we're saving elsewhere, which makes everybody in our organization super happy.

JEFF LAMBERT
VP OF DATA SOLUTIONS

In some ways, 84.51° is the modern-day embodiment of the spirit of DataOps. dunnhumby were true pioneers in the art and craft of applying data science to consumer behavior. And 84.51° is so valuable that Kroger keeps the company independent, so it can best serve a very wide range of stakeholders.

Powering their analytics excellence, 84.51° runs a complex technology stack, from Oracle Exadata and SAS, to Hadoop, Spark, and Impala on-premises, and Databricks on Microsoft Azure in the cloud.

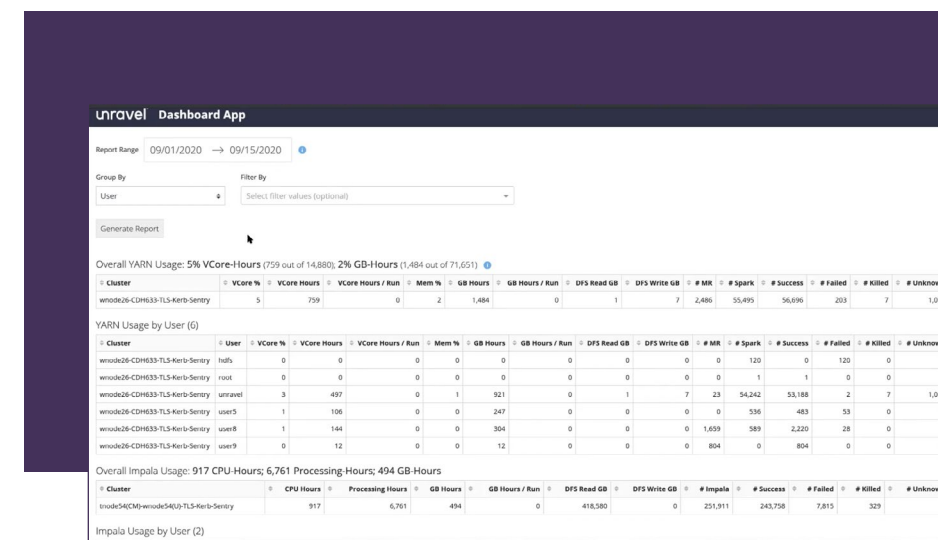
Managing this complex technology stack caused many issues – issues familiar to anyone involved in DataOps. Resource allocation, processing delays, and failed jobs meant customers and consumers went unserved, at great cost in decision-making efficiency, consumer spending, and customer satisfaction. Small files from rogue jobs clogged storage and monopolized processing resources.

- **Cluster efficiency:** Memory-constrained clusters
 - Team may over-allocate Spark memory, clogging up cluster
 - Queue contention causing apps to hang
 - Proactive alerting (AutoActions) for runaway queries, rogue jobs and resource contention
 - Make our hadoop clusters efficient and our users good citizens
 - Queues at peaks get busy so we need to focus on tune-ability – cluster and app wide
 - Visibility into usage analysis and capacity planning – ex: showback report by LOB, job, cluster
 - Visibility into tables that are accessed frequently
- **Small files:** Check the Small File reports periodically for anomalies – scheduled to run every day
- **Adoption:** App team enablement – identify resource-intensive and tune-able apps
 - Jupyter Notebook, **Executive Dashboard**, Sherlock and Pipeline Monitoring apps

Marketing services company 84.51° faced classic DataOps problems

84.51° used Unravel Data to solve these problems. According to Jeff Lambert, VP of Data Solutions, “I want to be able to select a date range, then have seven different group-bys, and then see the total amount for that timeframe, what’s happening on that cluster.”

Lambert continues: “Then I want to be able to see user-level detail for what they’re consuming for compute in the form of vCores, memory allocations, and read-rights. I’m basically looking for the heavy hitters that are out there on the cluster, that are really putting it to the test, and understand how these things translate to impacts,” in the cloud and on-premises.



A new executive dashboard helped 84.51° solve issues at runtime

Unravel Data met the needs of 84.51°’s needs on a wide range of fronts:

- **New executive dashboard.** A new executive dashboard to summarize cluster status and performance at a glance, making incipient problems visible and providing reassurance. Unravel created this dashboard for the company, and has since offered it to all customers.
- **Memory and CPU allocation.** Allocations were formerly guesstimates. Unravel allows 84.51° to rightsize allocations per job, freeing up large amounts of memory and CPU resources, and dramatically reducing costs.
- **Real-time alerting.** Unravel AutoActions let operators know when jobs hang, cost or resource overages occur, and so on, so action can be taken to fix them immediately. And Unravel then provides the information needed to diagnose and fix the problem(s).
- **Small files.** Some jobs generate large numbers of small files that consume surprisingly large amounts of system resources. 84.51° used Unravel to find these files and fix underlying problems, freeing up resources for all users.
- **Operational costs.** Unravel provides direct information on costs and cost reduction. 84.51° is able to find the most expensive jobs, optimize them, and measure the cost reduction directly.
- **Chargeback / cost allocation.** 84.51° is able to assign costs per job, per user, per department, per customer, and by a number of other criteria, as costs are incurred – not as one lump sum, weeks later.
- **Cloud migration.** Unravel also provides solid cost and performance projections for cloud migration, helping allocate workloads flexibly between the on-premises estate and the cloud, reducing costs and improving performance.
- **Customer effectiveness.** Unravel helps 84.51° support its CPG customers with more incisive and cost-effective analytics, with cost allocation to help track effectiveness.
- **Jupyter notebooks (resource usage).** Users were launching jobs via Jupyter notebooks, then not closing the notebooks, leaving the jobs running. Unravel helps clean this up.
- **Jupyter notebooks (integration).** Unravel enhanced its Jupyter notebook integration to provide improved control of jobs through this technology, for greater manageability.

The parent company, the IT organization, and customers are all freed to innovate. According to Lambert: “We can invest the dollars that we’re saving” – in productivity gains and efficiency – “elsewhere, which makes everybody in our organization super happy.”

CONCLUSION

DataOps has emerged as a critical discipline for effective use of big data and streaming data, in the cloud and on-premises. Companies are solving critical operational problems and achieving competitive advantage through the use of DataOps.

Unravel Data customers are more easily able to work all steps in the DataOps cycle, because they can see and work with their data-driven applications holistically and effectively. If you're interested in assessing Unravel for your own data-driven applications, you can [try out Unravel for free](#) or [contact us](#).

