

onix

Automated Code Transformation From Legacy ETL to Cloud-Native Technologies

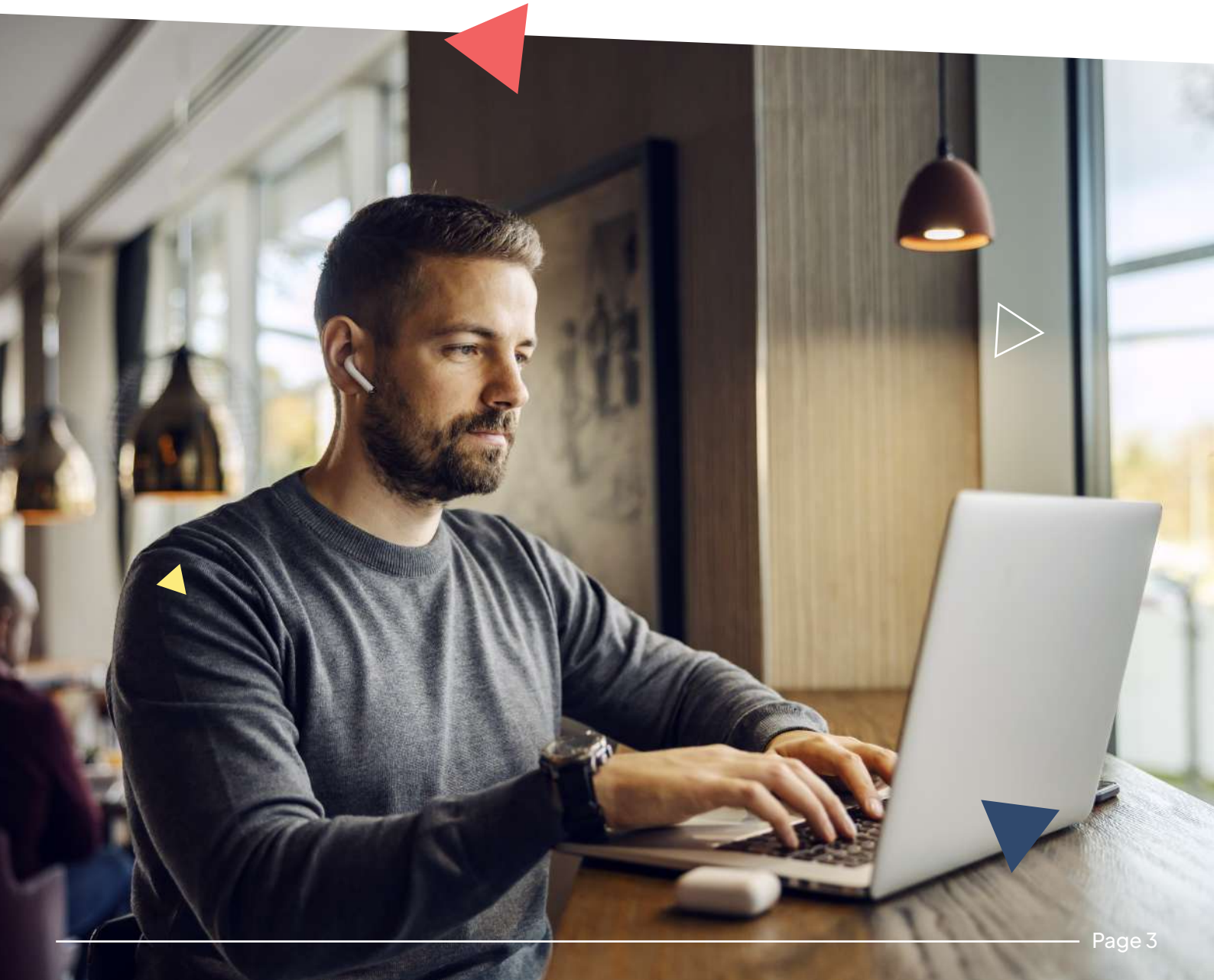
Table of Contents

- Introduction 03
- Complexities in legacy ETL tools 04
  - Time 04
  - Cost 04
  - Scalability 04
  - Process Limitations 05
  - Maintenance 05
- How to overcome ETL related pitfalls? 06
- How can businesses move their ETL systems to the Cloud? 07
- Why is automation a must-have for efficient ETL migration? 08
- Why Raven? 09
- Enterprise benefits that Raven unlocks 10
- ETL conversion challenges solved by Raven 11
- Onix's automated ETL code conversion technology 12
- Working of Raven: The Transformer 13
  - Step 1: Extraction 13
  - Step 2: Canonical Model Creation 13
  - Step 3: Code Generation 14
- Raven ETL: high-level logical view 15
- Success stories delivered by Raven 16
- About Onix 18

Introduction

The rapid growth of data across businesses is fueling technological innovation. The last few years have seen more modernization in the space of data management than the previous twenty years combined. The cloud is one of today's hottest and most dynamic technologies since traditional on-premise systems are insufficient to handle the everincreasing high operating costs, fixed storage capacity, and restricted infrastructure. The introduction of cloud platforms has transformed traditional business models and made new things possible in business. In order to leverage futuristic technologies and analytical capabilities, businesses are rapidly modernizing their legacy data warehouses. ETL providers are pushing for disruptive innovations; database vendors are creating attractive opportunities for quicker analytics; and cloud vendors are delivering standardization cases to client C-suites. There appears to be activity in the data management sector everywhere you turn. But, with legacy data warehouse migration, a major question arises: What do we do with our legacy ETL system?

Before we think about what we should do with our legacy ETL, we first need to understand the complexities and limitations of the legacy ETL tools.



Complexities in legacy ETL tools

Legacy ETL tools are mainly instructions written as codes on how to move data from a source to a target. ETLs extract data from varied data sources, put it into the transformation phase, and load it into the data warehouses or data lake. Further in the legacy data warehouses or data lakes, there are many more ETLs tools that carry out the same function internally. As data sources, ETL tools, and data warehouses/data lakes are in the same on-premises environment, their dependency on the network infrastructure is significantly less, contributing to the better performance of the ETL tools, and they don't become problematic until the data warehouses expand in size.

Legacy ETL tools also lack the agility to convert data at the speed required for loading into a data warehouse or data lake in today's significantly rising pace, volume, and multi-source data. Apart from their inability to keep up with the speed of modern big data, legacy ETL solutions have other flaws.



ETL processing generally involves the use of a large number of external tools for extraction and ingestion. A team of skilled data engineers usually needs several months to build up such a process and integrate the tools, which causes bottlenecks from the very beginning. Even the maintenance process requires additional time. When an organization has tens of thousands of ETL jobs for moving data or applications, timelines become very important.



ETL processing generally involves the use of a large number of external tools for extraction and ingestion. A team of skilled data engineers usually needs several months to build up such a process and integrate the tools, which causes bottlenecks from the very beginning. Even the maintenance process requires additional time. When an organization has tens of thousands of ETL jobs for moving data or applications, timelines become very important.



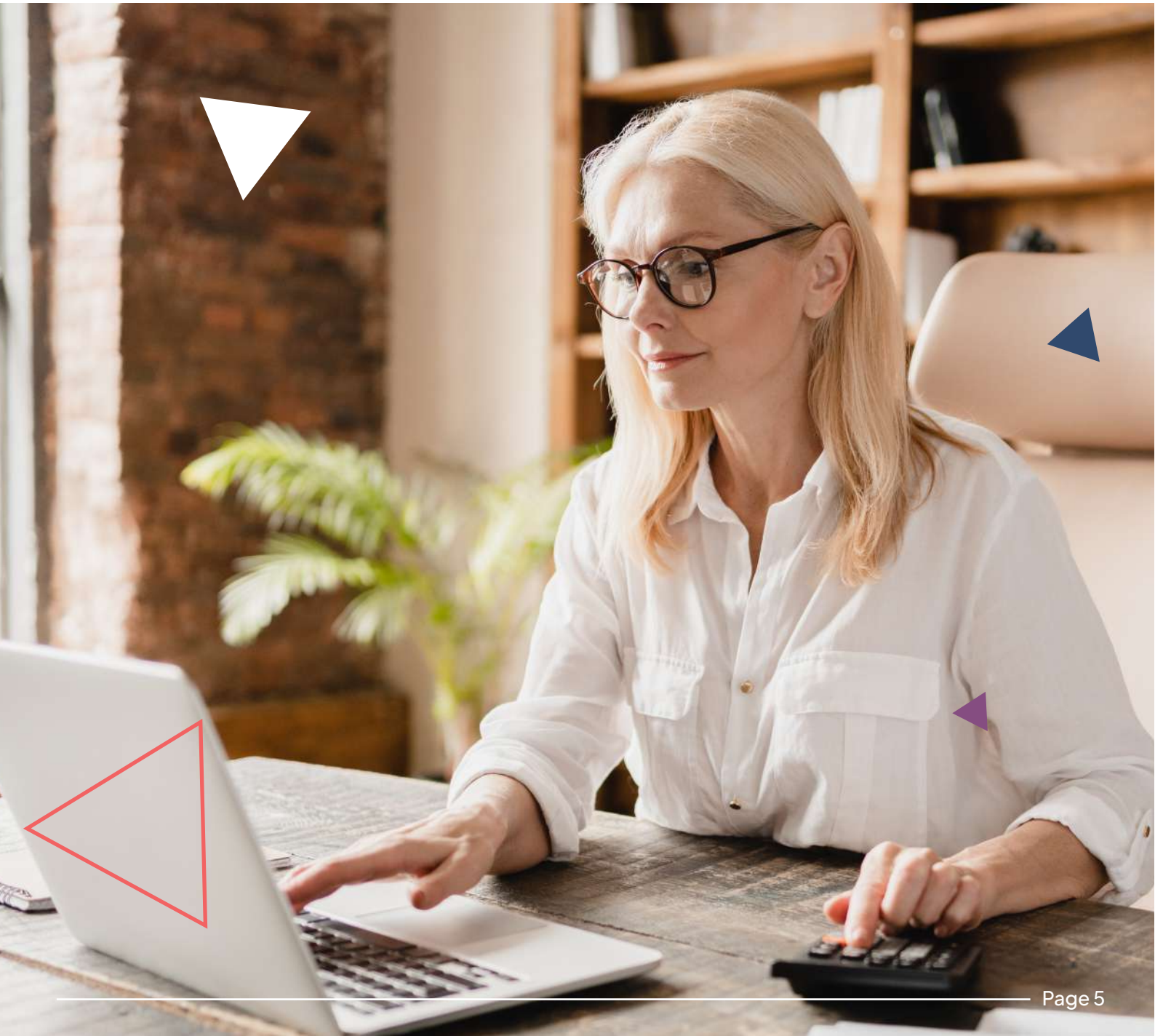
As we all know, legacy ETLs were never designed with cloud platforms in mind. This clearly means that the tools are incapable of unlimited scalability, which is an important pillar of cloud platforms.

Process Limitations

Initially, ETLs were developed for periodic batch processing. Even today, it hardly supports continuous and automated data streaming. A limited number of processes are allowed for users in legacy ETL systems.

Maintenance

Legacy ETL tools require a dedicated manager for maintenance, updates, and renewals. These time consuming and operational processes drain resources and bandwidth.



How to overcome ETL related pitfalls?

Traditional ETL has become a burden for data-driven enterprises that demand continuous data integration for near-real-time business insights due to the current data boom. This is why many businesses are supplementing, if not replacing, conventional extract, transform, and load procedures with ELT or a cloud based modern data pipeline. For legacy ETL operations, managing this madness became almost impossible and contributed to the rise of a replacement, ELT.

Data integration occurs between the source and target systems in ELT, without prior business logic-driven transformations. ELT merely reorders the typical integration processes, with the transformation occurring at the end. The massive business shift to cloud-based software services, along with ELT and data pipelines, has the potential to significantly enhance and streamline data processing for enterprises. Businesses that continue to use batch processing can begin incorporating new continuous processing approaches without disrupting their existing systems. Rather than an expensive rip-and-replace, the adoption might be incremental, beginning with specific types of data or business sectors.

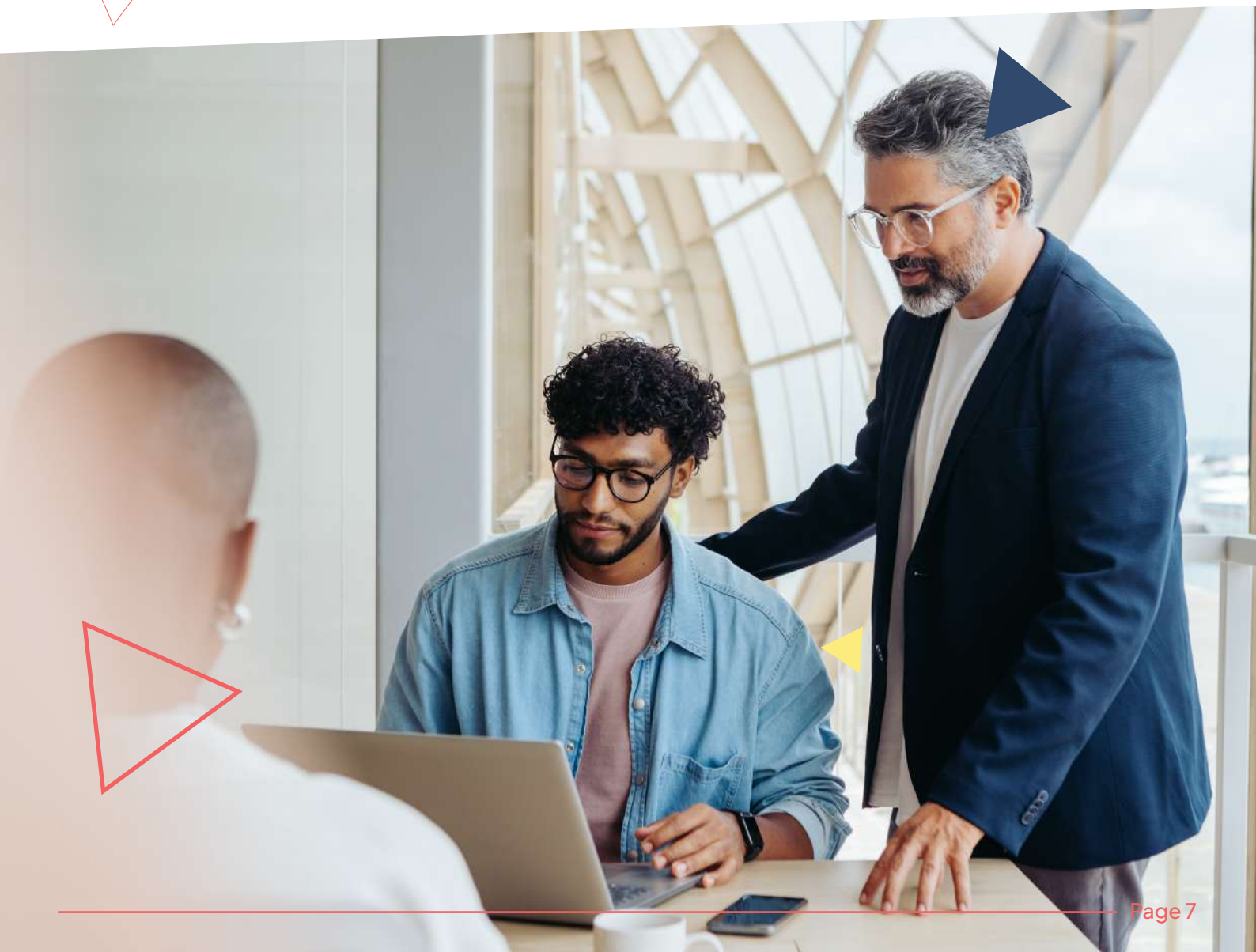


How can businesses move their ETL systems to the cloud?

Modernizing legacy ETL systems and making them compatible for the cloud has become a strategic need for enterprises dealing with petabytes of unstructured data that is rapidly growing from various sources, as well as high data ownership and operating costs. Reuse, or repointing, is one of the solutions that allows you to use existing legacy ETL tools after a few updates to the codes. This method is also considered as ETL migration, but all the major complexities and limitations of legacy ETL tools remain untouched.

Another method that is considered as the most effective way of handling legacy ETL is ETL modernization, or rewrite, or decommissioning. This method of ETL modernization requires the alteration and modification of the existing ETL codes to make them compatible with the cloud platforms.

The core principle of this solution is to separate data flow, which is data ingestion and export, from data processing, which occurs in the same ETL pipelines. Then, put your data movement and processing tasks in the hands of cloud-native methods and tools like Google Dataflow. This implies that the entire task code will be rewritten in a cloud-native language.



Why is automation a must-have for efficient ETL migration?

As we know, 'rewrite' or 'decommission' addresses significant performance, cost, lookup, and data transfer issues and is the right approach towards ETL modernization; it is nearly impossible to rewrite ETL codes manually. Many organizations try to convert ETL codes manually, ignoring all the complexities and challenges, and end up derailing the migration project or having a significant delay. Also, manual ETL conversion is a very tedious and laborious process. So, today, one of the most effective methods of ETL modernization is automation. It has been proven to significantly reduce the cost, time, and complexity of ETL modernization.

In order to get started with automating the ETL conversion, first Eagle technology could be used to access the legacy ETL environment and provide a detailed analysis of complexities and risks, as well as a complete inventory listing for jobs and their components. These details will then be used by Eagle to provide a precise modernization plan with an estimated conversion timeline.

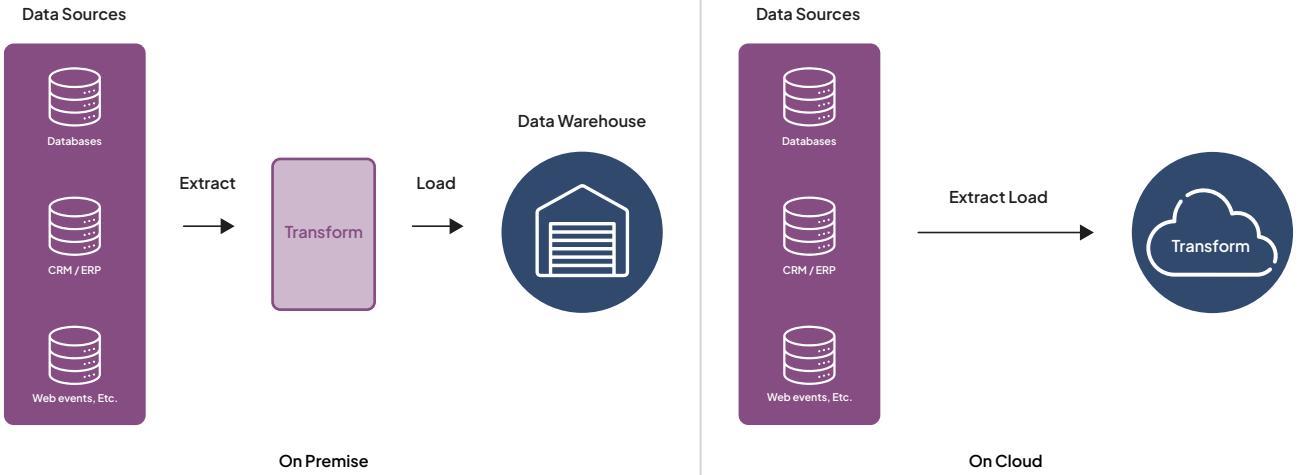
Once the assessment is complete and we have a modernization roadmap from Eagle, Onix's Raven can be brought into action. Raven automates the process of ETL modernization, thus reducing the complexity of the process and delivering it at a lower cost and in less time.



Why Raven?

Raven, an automated code and ETL transformation technology, simplifies the code conversion process in the cloud migration journey. This automated code converter significantly reduces the time taken for code transformation and ensures that a system-wide standard for the converted tool exists. Raven transforms ETL-based custom expressions and non-SQL ETL workflow components.

Rewriting ETL Codes with Raven: ETL to ELT





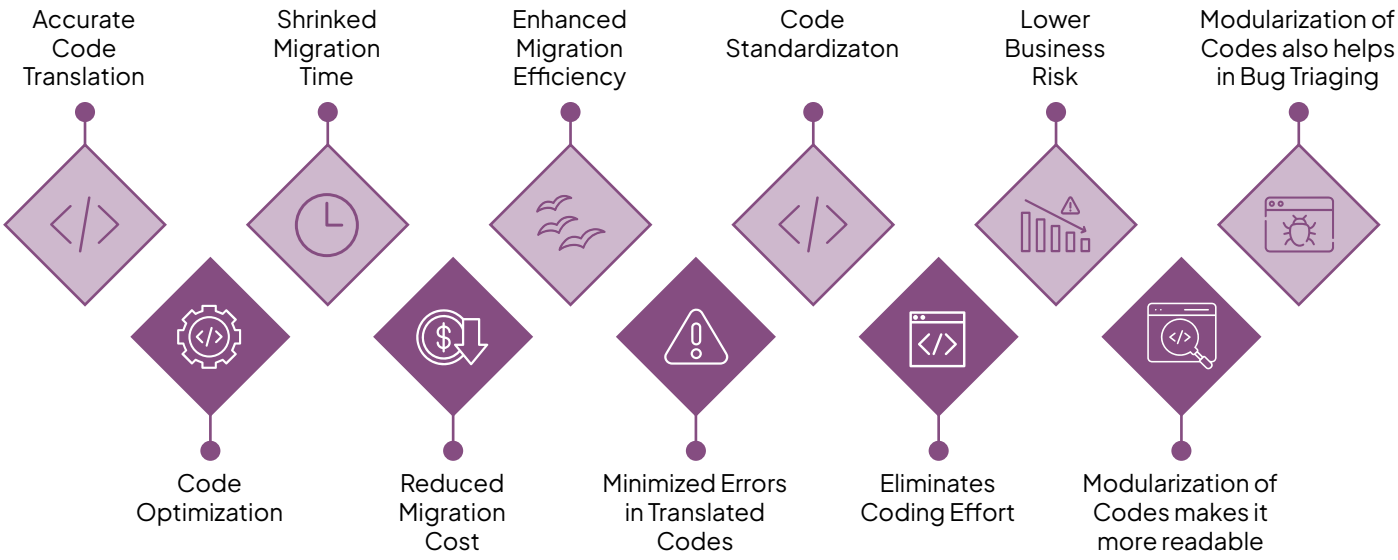
Raven swiftly and reliably transforms ETL into ELT-based scripts that are compatible with the chosen cloud platform. Raven makes sure that the extraction and loading processes are retained on-premises by decoupling them, while only the transformation phase is kept on the cloud platform, where the data is re-coded. The conversion of ETL codes to ELT codes eliminates network bandwidth needs as well as potential performance difficulties. Raven's automated methods minimize the reliance on technical expertise from both the source and destination systems, as well as the costs involved with the migration's code conversion process.

Raven employs a canonical model to execute one-to-one pipelines while converting legacy code to target platform code. This canonical model is produced by reading and evaluating the legacy data's metadata and is entirely independent of the source platform. This model is subsequently converted into system code for implementation.

Enterprise benefits that Raven unlocks

Raven eases the laborious job of code conversion and brings enormous business benefits to the organization. Datametica's automated ETL migration solution, Raven ETL, leverages its automation capabilities to deliver optimal conversion of legacy ETL to modern cloud-based ETL/ELT tools. Here are some of the business benefits that Raven delivers:

Enterprise benefits that Raven unlocks



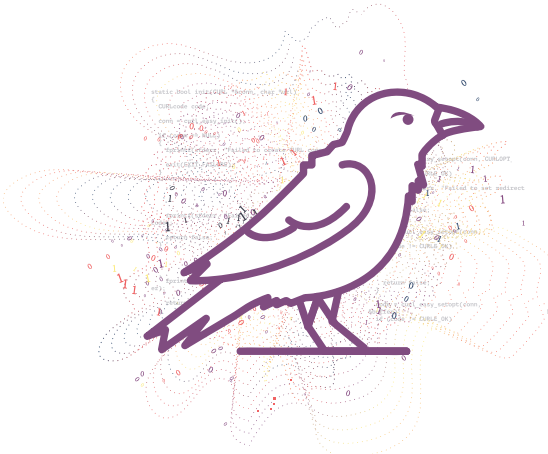


ETL conversion challenges solved by Raven

As mentioned above, ETL conversion enhances the performance and also reduces the cost incurred, but doing it manually needs infinite time and is almost impossible.

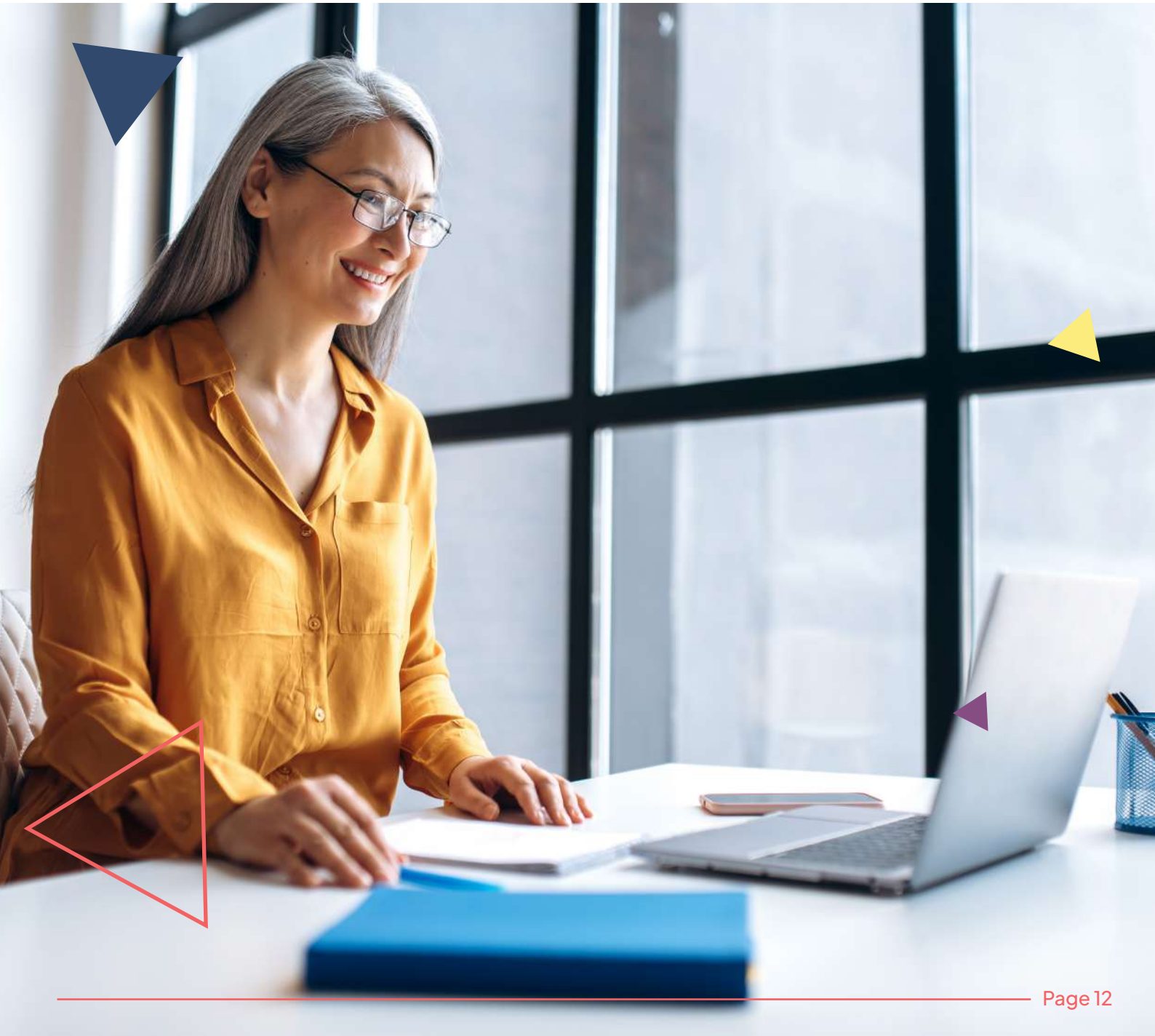
So, what are the complexities that can be skipped using Datametica's automated code converter, Raven? These are the complexities that are associated with ETL code translation and make this process very complex and impossible.

- Logic understanding and extraction of ETL pipelines
- All ETL tools involve different programming stages
- Similar and redundant stages/pipelines/business logic
- Rewriting the pipeline/stages while the underlying data sources are also being migrated
- Complex embedded SQL within pipeline/stages
- Embedded SQL needs to be re-written, as the underlying query engines are changing
- Non-SQL elements written in statically typed languages in ETL workflows hinder quick development
- Redundancies are introduced when heterogeneous data sources move to a homogeneous data source
- It is hard to analyze and understand the data lineage and complete data flow pipeline in the system
- Custom expressions written in the ETL engine need to be translated to a target query
- Performing optimizations like predicate pushdown, projection pruning, join shuffling, etc. takes time and skill
- Optimization at the orchestration layer, like merging/removing redundant jobs, requires knowledge of the complete pipeline
- Merging complex custom queries and a job's logic is a time consuming and error prone process
- Common processing branches involved in a complex job require special attention



Onix's automated ETL code conversion technology

Raven ETL is a tool that turns ETL-based business logic into ELT-based scripting. Raven ETL takes input in the form of exported ETL pipelines from ETL tools. It parses the pipelines and extracts the business logic from them. This business logic is represented by Raven as a relational tree. Which is a common relational model for representing data processing jobs. After this relational model is built, it is further optimized and then converted to SQL. All of the processing required for the data pipeline is encapsulated in generated SQL. These SQLs are also tailored for the target system in order to completely use the target query processing engine. Raven also uses emulators to translate functions, expressions, and programming constructs generated using ETL tools to SQL-based processes.



Working of Raven: The Transformer

If we talk about how Raven performs this translation, it basically follows a three-step method. We can also consider these steps as phases of conversion.

Step 1: Extraction

Raven ETL takes workflow metadata. Legacy ETL tools can export these XMLs. Raven ETL extracts all metadata information for the job or workflow once it is fed into the system. Because the extraction process is unique to legacy ETL technologies, Raven ETL provides a distinct extraction layer for each tool.

Step 2: Canonical Model Creation

In this step, the information gathered is transformed into a canonical model. This step is independent of legacy ETL tools. It generates relational frameworks for embedded custom SQLs in jobs and business logic, which are then integrated to complete the canonical model development for the job. This step is also in-charge of the orchestration layer (for example, Sequencers in DataStage and Workflows in Informatica).

In order to build an optimum query execution plan, optimization techniques such as filter pushdown, projection pruning, etc. are used after model creation. This model is then put through another optimization layer, which recognizes common branches in the ETL job and optimizes the query plan to run common logic just once in order to reduce processing costs. This phase concludes by mapping all entities engaged in the job to entities in the target system.

The Canonical Model Helps in Performing the Following Operations:

- Possible to analyze lineage-aware data models
- Helps in the identification of the common branches of data mapping
- Enables SQL Optimization, Expression Optimization, and Workflow Optimization
- Enables adding support for new target systems in significantly less time



Step 3: Code Generation

The canonical model developed in the previous step serves as an input for this stage. Raven ETL generates code for a variety of target environments, including orchestrators and data warehouses. Users, for example, can migrate orchestration logic to Airflow, Control-M, etc., and conversions to BigQuery, Snowflake, and so on.

Code Generation can be further divided into the following steps:

1. Target System-Specific Optimization

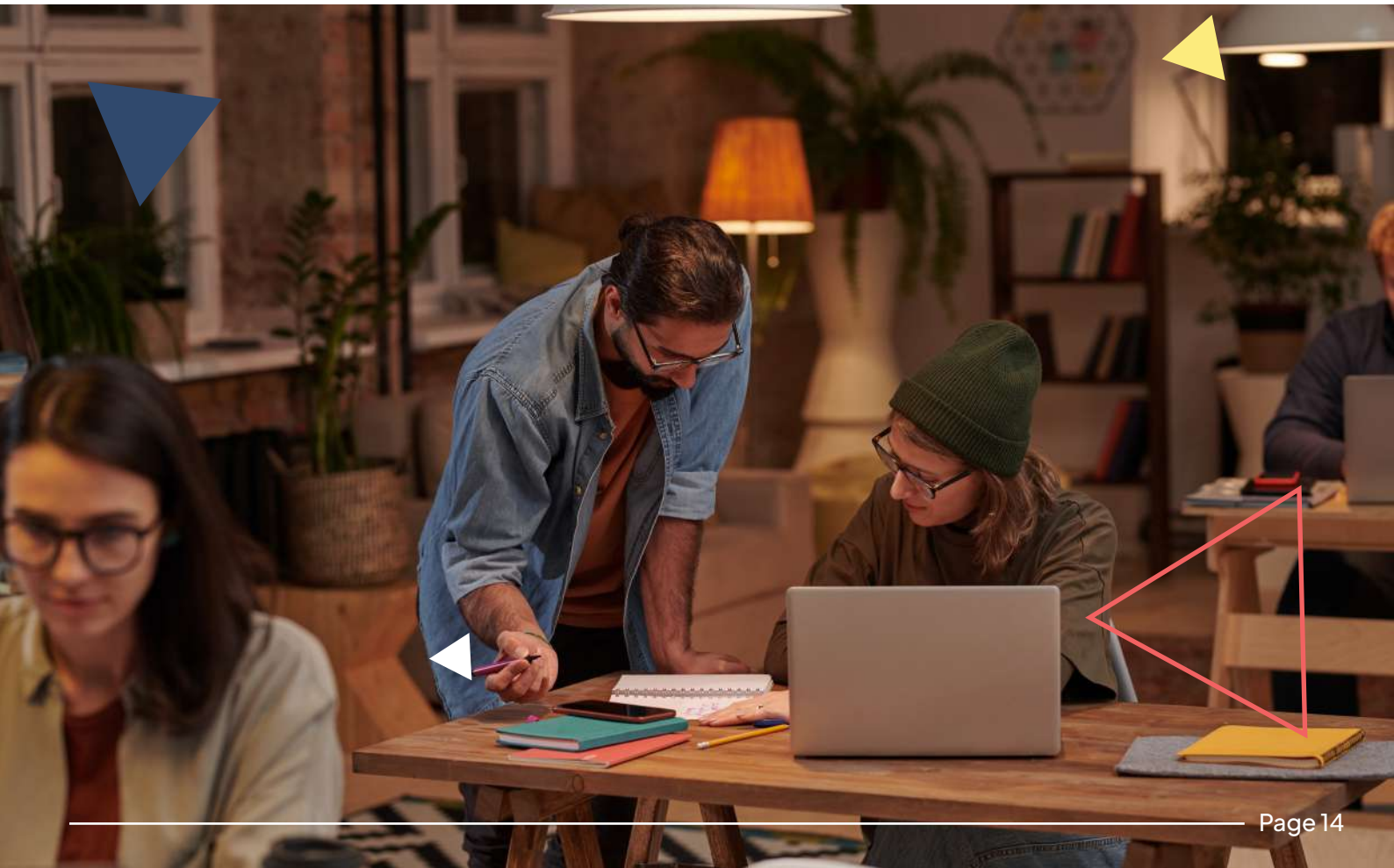
Code generators are target-specific and offer an additional level of optimization. Because this optimizer is target-conscious, it utilizes all the features of the target system.

2. Maintainable vs. Optimized Code Generation

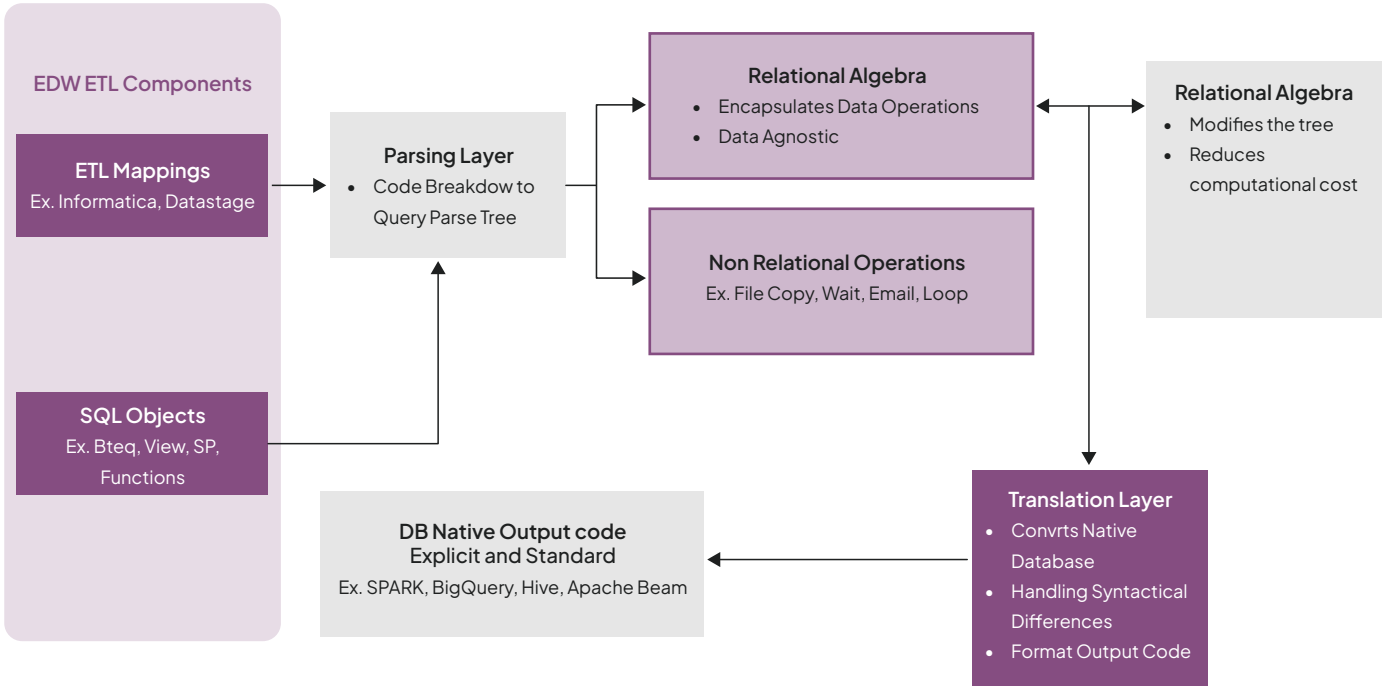
For a given relational model, Code Generator can generate either minor modular scripts that are easier to test and validate or a single monolithic script that performs better. The code generator can generate code for either of the mentioned variations or an intermediate route that is both modular and performant based on a configurable score given by the user.

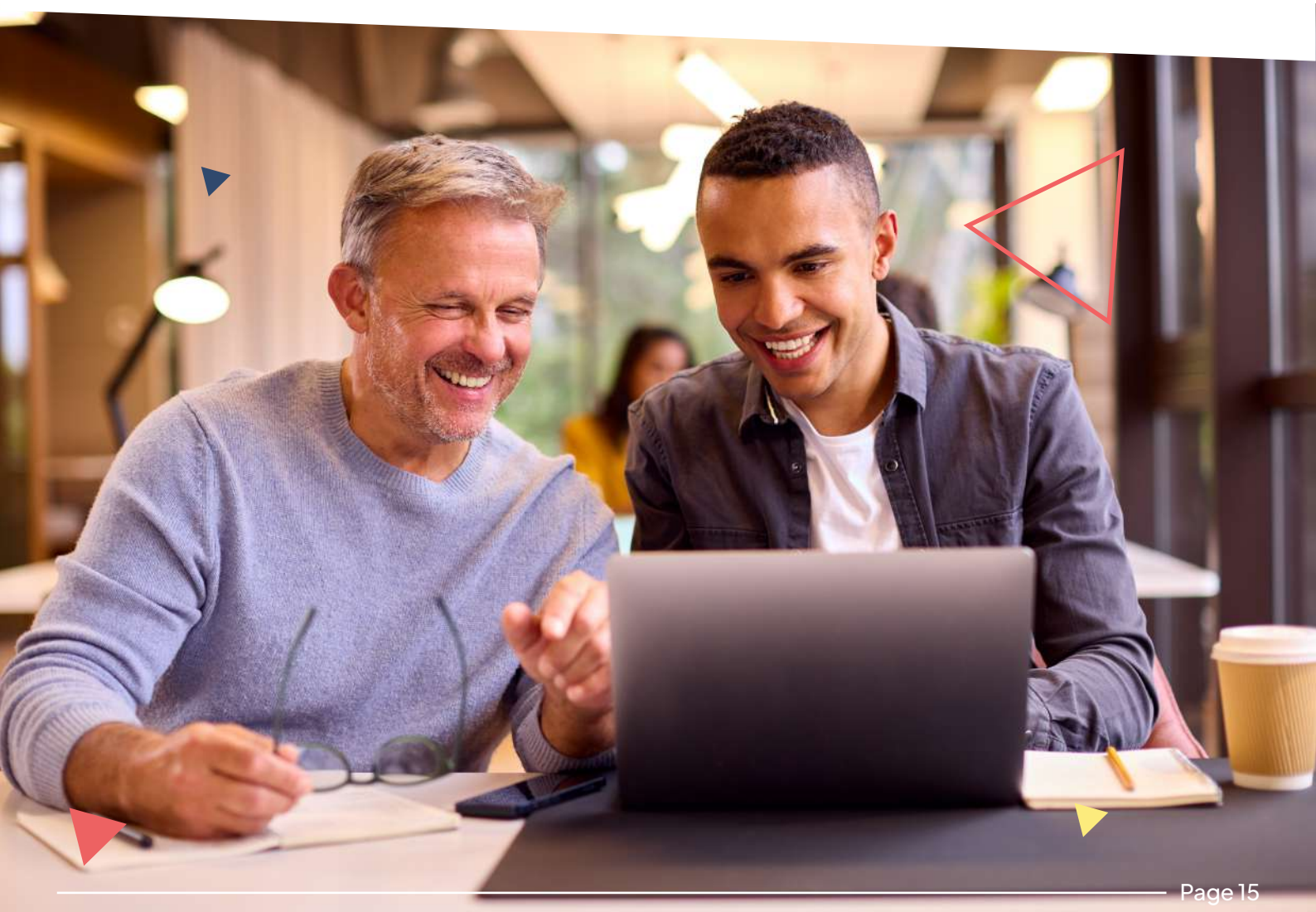
3. Artifact Writer

Artifacts are formed at this stage based on the target environment. This stage generates appealing scripts using a default configuration that can be further customized to the user's preferences.



Raven ETL: high-level logical view





Success stories delivered by Raven

Onix has deployed its automated code conversion technology, Raven, in many of its projects, one of which is An American Multinational Financial Services Company.

The Objective

An American multinational financial service provider was looking for a technology partner that could take care of its code conversion in its cloud migration process in a cost-effective and rapid manner. Onix's Raven, an automated code transformation product, helped the client convert their current system code and data objects to BigQuery as part of their migration to GCP.

Challenges

The client wanted to leverage GCP capabilities, so they initiated the cloud migration process. They were facing difficulties in converting the current complex codes of Teradata and Hadoop.

Solution

Datametica determined the code details and deployed its automated code (SQL, script) and ETL conversion technology – Raven to convert all the Teradata and Hadoop objects (scripts).

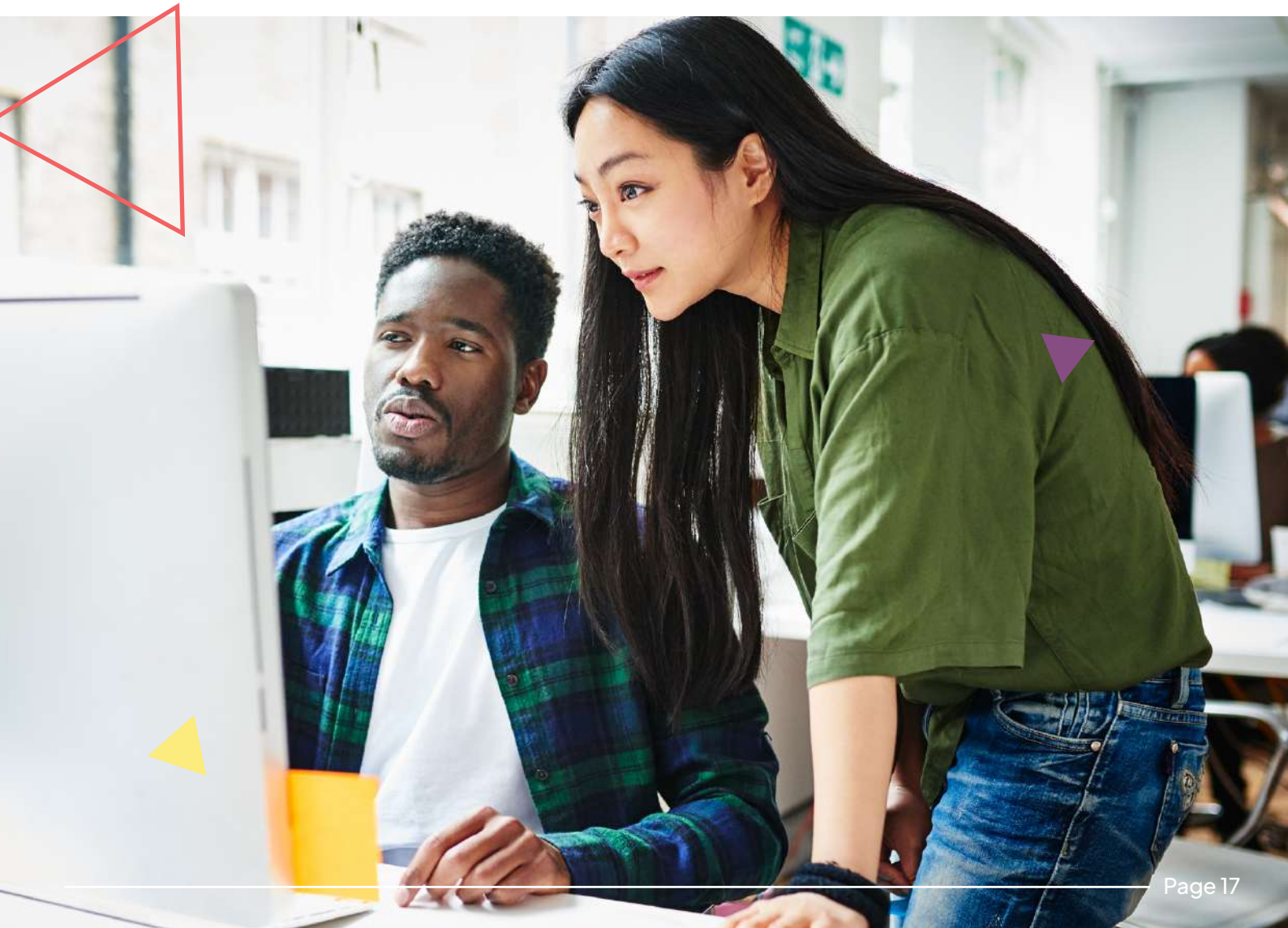
- The automated code conversion process by Raven translated the source artifacts to target without any change to source logic
- 100% Conversion of Teradata tables, view DDLs, UDFs, Macros, stored procedures, Informatica workflows, ETL codes, BTEQs, and Shell Scripts to GCP native using Raven
- 100% conversion of Hadoop objects, scripts, HDFS, etc.
- Support non-ANSI compliant constructs, function and clauses
- SQL, expressions, and workflow optimizations
- Smart emulations for incomplete SQL statement and clauses

Benefits Delivered

- 50% Reduction in total cost and enhanced performance
- 35% Faster migration to GCP
- Successful translation of complex Teradata, Hadoop, and related codes
- GCS bucket was created and shared with the client for data transfer and conversion
- The unit tested all the converted codes on the empty data structures
- High quality delivery in production

Onix has extensive experience in making the cloud migration journey of their clients seamless. You can check out the case studies of the projects that Onix has successfully delivered, tackling all the complexities of the legacy system and fulfilling the desired expectations of the clients.

You can also get in touch with Onix's team to get more details about Raven through code conversion guides and Raven product demos. Raven is also available on Google Cloud Marketplace.



About Onix

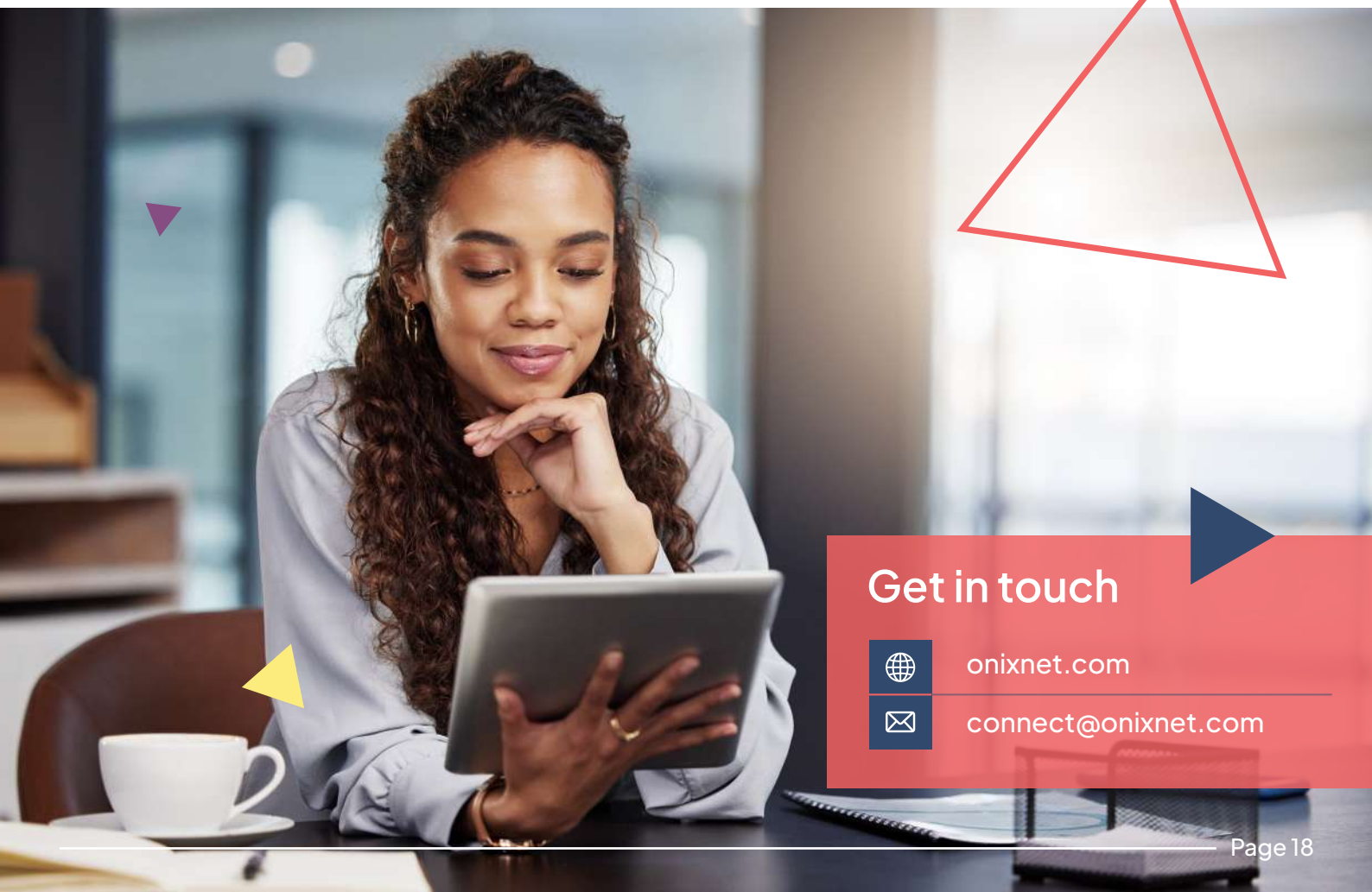
Running a business in a digital world demands the ability to evolve. At Onix, we know this firsthand. Since we launched in 1992, we've had to evolve and adapt to changing technology developments and customer needs.

In fact, we had the foresight to see the potential in cloud very early on. We were one of the very first companies to place a bet on Google Cloud and established our partnership in 2002. Our partnership expanded over the years to cover the entire Google Cloud product portfolio, serving some of the world's leading companies.

In 2022, Onix partnered with Tailwind Capital to accelerate the next phase of our company's growth and to build a leading Google Cloud partner globally. In October 2023, Onix made the strategic acquisition of Datametica, an IP-powered consulting firm specializing in data migration, modernization and analytics. The addition of the Datametica Birds data migration product suite has accelerated Onix's hyper-growth powered by data and generative AI solutions.

Today, we have thousands of employees working globally to elevate our customers to new levels of automation, collaboration, efficiency and success. Our traditional hardworking roots and forward-thinking vision have resulted in Onix being named a Google Cloud Partner of the Year 14 times.

As we embark on this next phase of evolution as a company, the common thread remains: a laser focus on solving our customers' greatest challenges with technology and providing the highest quality service.



Get in touch

onixnet.com

connect@onixnet.com