advanced

People. Partnership. Solutions.

Modernization Platform as a Service (ModPaaS)

Google Cloud

A solution overview for Google Cloud Platform

ModPaaS: Collaborative modernization platform

ModPaaS is a purpose-built, customizable cloud solution that allows for highly collaborative customer participation throughout the journey of a legacy application modernization project. By providing you with access to our ModPaaS solution, you can move through the modernization journey at your own pace in a completely self-service manner, with assistance from our modernization specialists, or in a fully managed approach (similar to a traditional project engagement).

ModPaaS is available for customers looking to analyze mainframe legacy code, understand interrelationships, complexity, dependencies and mine business rules. These are essential activities in order to successfully plan for any complex modernization project. Having access to ModPaaS allows you to:

> Perform a comprehensive assessment of your existing legacy applications, including asset count and asset type categorization, missing and unreferenced components, program flows, impact and code path analysis, reporting and documentation, and more.

> Categorize objects and applications to determine disposition options (e.g. rehost, automatically refactor, reengineer, replace, etc.), and drive ongoing modernization initiatives with different levels of assistance.

> Trace and isolate COBOL-based business rules which can be reused for reengineering initiatives.

> Develop a strategic modernization plan which includes recommendations and estimates that best suit your individual technical and business needs.

> Select and extract code to be modernized. As an example, you can select code specific to the UI layer or data access objects to aid microservices development efforts. Post-assessment options are described later in this document.

FACT SHEET

ModPaaS: Solution Overview for Google Cloud Platform

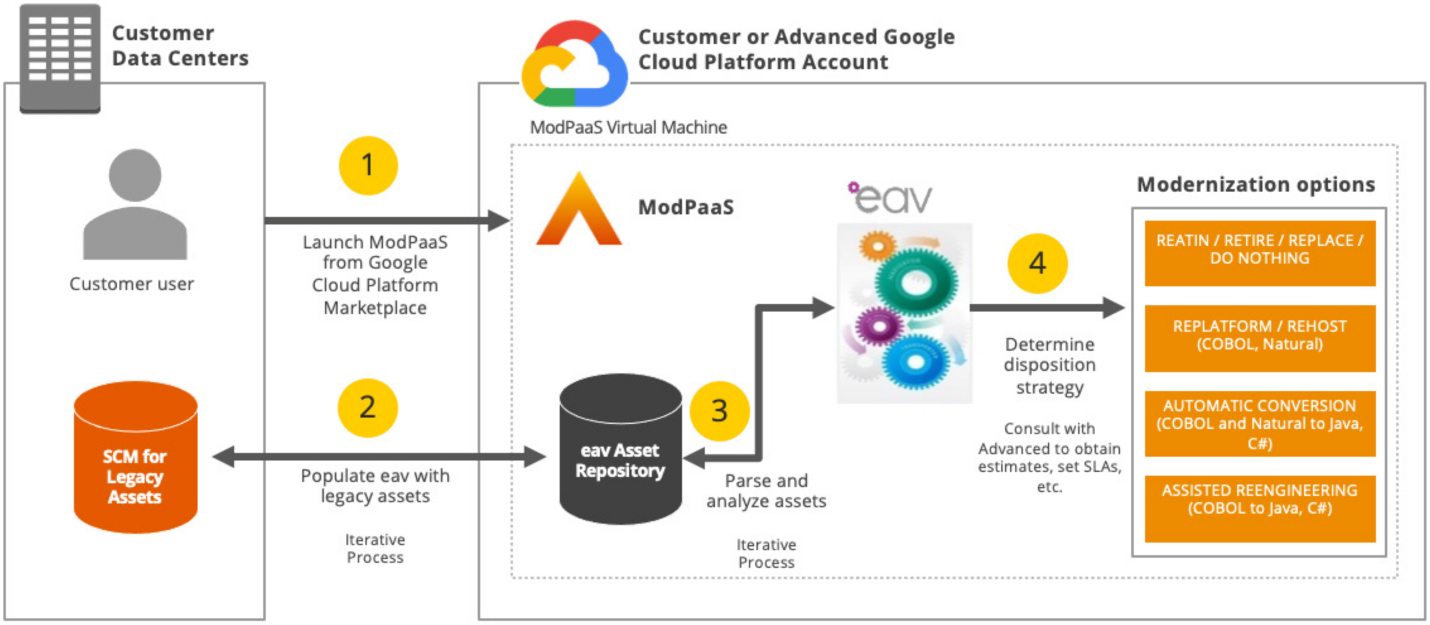

Customer
Data Centers
Customer user
SCM for Legacy Assets
1
Launch ModPaaS from Google Cloud Platform Marketplace
2
Populate eav with legacy assets
Iterative Process
Customer or Advanced Google Cloud Platform Account
ModPaaS Virtual Machine
ModPaaS
eav Asset Repository
3
Parse and analyze assets
Iterative Process
eav
4
Determine disposition strategy
Consult with Advanced to obtain estimates, set SLAs, etc.
Modernization options
REATIN / RETIRE / REPLACE / DO NOTHING
REPLATFORM / REHOST (COBOL, Natural)
AUTOMATIC CONVERSION (COBOL and Natural to Java, C#)
ASSISTED REENGINEERING (COBOL to Java, C#)

Figure 1: Advanced ModPaaS assessment process overview

Using ModPaaS for an application assessment

At the heart of ModPaaS is the Advanced Enterprise Application Viewer (eav), an automated assessment solution that has been used to process millions of lines of legacy code. The following image represents the ModPaaS assessment process, and the numbers in green represent each step in the process and correspond to the related text that follows.

1. Within your Google Cloud Platform account, launch a Windows-based ModPaaS instance.

2. Populate the eav asset repository with legacy source code and data definitions.

3. Automatically parse assets, produce asset counts, categorize types, perform impact analysis and code path analysis, generate program flows, dependencies, and create numerous reports and documentation. Steps 2 and 3 are an iterative process, performed until all assets reported as being missing are resolved.

4. Collaboratively analyze results and further categorize assets into specific application groups based on possible disposition strategies. Discuss modernization options and associated migration tasks, service levels and pricing, define statements of work (SOW), Advanced.

ModPaaS: Solution Overview for Google Cloud Platform

	Programs			Lines of Code (LOC)		
	Assessed Count	**Referenced Count**	**Scope Reduction Based on Count**	**Assessed Lines**	**Referenced Lines**	**Scope Reduction Based on Lines**
1	**10,897**	**6,919**	**37%**	**2,099,751**	**1,461,965**	**30%**
2	**60,727**	**36,332**	**44%**	**14,777,426**	**11,574,369**	**22%**
3	**20,424**	**10,913**	**47%**	**5,641,564**	**3,909,054**	**31%**
4	**18,824**	**9,989**	**47%**	**5,662,567**	**2,985,486**	**47%**
5	**43,922**	**10,140**	**54%**	**8,995,039**	**3,642,442**	**60%**
6	**6,560**	**2,911**	**56%**	**1,507,337**	**1,231,898**	**18%**
7	**8,876**	**3,921**	**56%**	**2,271,508**	**976,180**	**57%**
8	**18,772**	**7,367**	**61%**	**6,221,896**	**3,799,189**	**39%**
9	**17,487**	**6,540**	**63%**	**4,517,160**	**2,295,331**	**49%**
10	**12,157**	**4,481**	**63%**	**1,511,899**	**752,142**	**50%**
11	**6,118**	**2,229**	**64%**	**2,668,985**	**753,356**	**72%**
12	**27,844**	**9,377**	**66%**	**3,516,482**	**2,475,551**	**30%**
13	**55,631**	**18,662**	**66%**	**7,177,265**	**2,810,004**	**61%**
14	**62,446**	**20,640**	**67%**	**22,166,838**	**10,415,518**	**53%**
15	**20,170**	**6,127**	**70%**	**3,189,892**	**1,543,033**	**52%**
16	**48,685**	**14,055**	**71%**	**6,637,431**	**3,904,086**	**41%**
17	**70,106**	**18,379**	**74%**	**13,124,693**	**4,155,391**	**68%**
18	**76,607**	**17,811**	**77%**	**27,307,461**	**6,438,850**	**76%**
19	**39,116**	**9,028**	**77%**	**7,937,964**	**2,642,381**	**67%**
20	**39,116**	**28,301**	**78%**	**44,310,210**	**9,594,969**	**78%**

ModPaaS capabilities and benefits

Using ModPaaS can result in significant business benefits, including a simplified, lower cost modernization project by dramatically reducing the scope of existing applications, and refining the modernization disposition strategy.

The main benefits of using ModPaaS for an assessment include:

> **Complete visibility into the legacy system** – Assets loaded into ModPaaS are automatically parsed, counted, and categorized by asset type. Reports and documentation are automatically generated, and program interactions are created to represent the overall program flow, individual paragraph flow, and inter-relationships between the programs, data, and other component types.

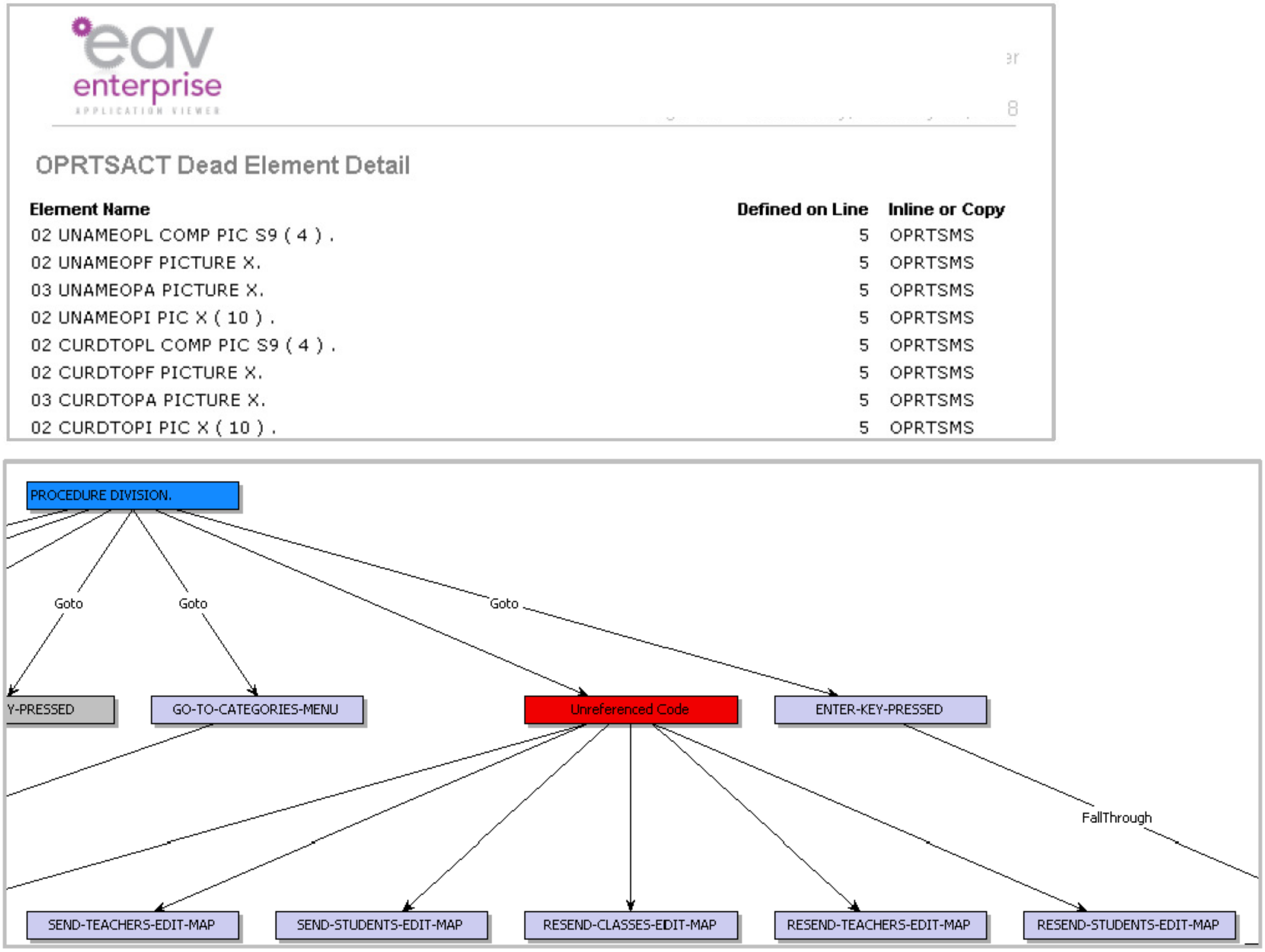
> **Reduced application footprint** – Results can be interactively analyzed to resolve assets being reported as missing, unused, or unreferenced. This is a critical task as it reduces the cost, complexity, and risk of the future modernization project. The following table lists actual scope reduction results from 20 past projects, ranging from 40 to 70 percent.

> **Global reports and documentation** – Each time an asset is parsed, eav automatically updates its documentation repository, generating multiple reports which can be viewed/downloaded as PDFs or exported into a variety of formats (e.g. CSV, Excel, XML and DIF) for further manipulation. Key reports include:

ModPaaS: Solution Overview for Google Cloud Platform

- *Object Summary*: a summary of object type count and line count within object type. The totals are displayed at the bottom of the report. The Statement Count and Comment Count columns are only visible for component types that eav creates metrics for.
- *Repository Detai*l: an alphabetical list of every object in the repository. Total objects listed are displayed at the end of the repository section and the total number of objects listed in the report appears on the last page.
- *Complexity Detail:* the maximum, minimum and average cyclomatic complexity values for every object in each repository. A cyclomatic breakdown appears at the end of the report.

- **Code clean-up** – ModPaaS assists with the clean-up of COBOL code to make future maintenance of the code more efficient. Cleaning is provided at three levels:
  - *Source code file level:* Identify entry points for both online (transaction code) and batch (job) source code. Using these entry points, eav automatically identifies programs and copybooks which are potentially not used so may be deleted from the code base.
  - *Paragraph/Section level:* Identify unreachable paragraphs and sections within the code and remove them.
  - *Line of code level:* Identify and remove unreachable executable lines in the code and unused variables in the code and copybooks.



OPRTSACT Dead Element Detail

Element Name	Defined on Line	Inline or Copy
02 UNAMEOPL COMP PIC S9 (4) .	5	OPRTSMS
02 UNAMEOPF PICTURE X.	5	OPRTSMS
03 UNAMEOPA PICTURE X.	5	OPRTSMS
02 UNAMEOPI PIC X (10) .	5	OPRTSMS
02 CURDTOPL COMP PIC S9 (4) .	5	OPRTSMS
02 CURDTOPF PICTURE X.	5	OPRTSMS
03 CURDTOPA PICTURE X.	5	OPRTSMS
02 CURDTOPI PIC X (10) .	5	OPRTSMS

Figure 2: ModPaaS eav Dead code analysis

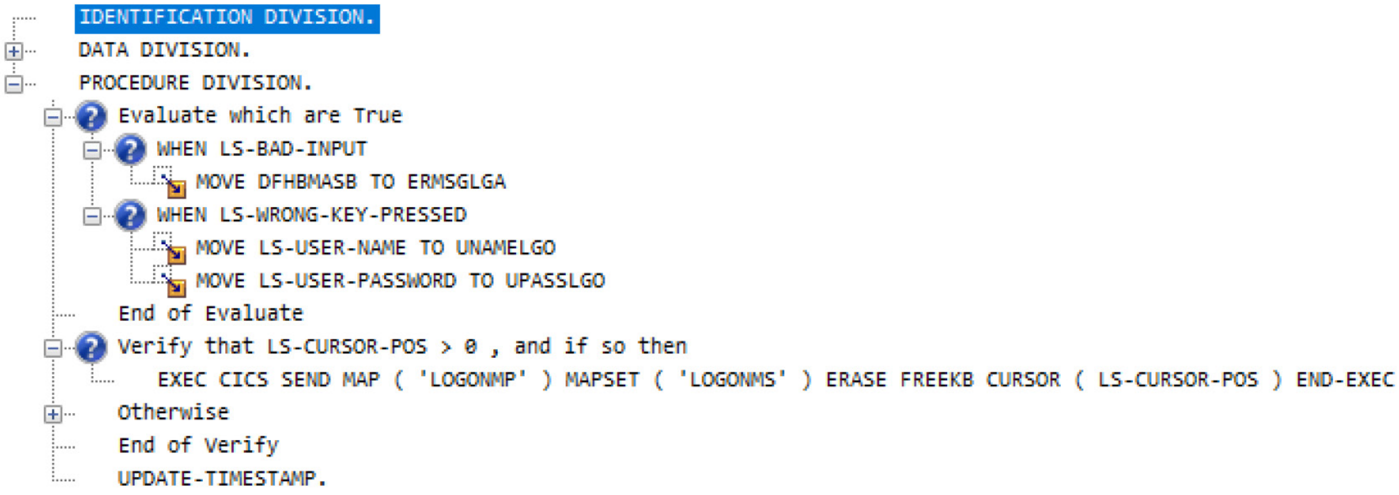
ModPaaS: Solution Overview for Google Cloud Platform

- **Business Rules Mining** – eav assists with reengineering initiatives such as the creation of microservices functionality by allowing you to mine mainframe COBOL business rules from within application logic. Select rules can be extracted from your code and saved as COBOL – then automatically refactored to the target language (e.g. Java, C#) and further optimized and refactored as required. Starting from the original business rules ensures that you at least understand the base usage requirements from the code before building out any new functionality.

As shown below, a tree diagram is presented for each application program or object, allowing you to collapse or expand each decision point to isolate, review, or extract the business rules. Gaining an understanding of inter and intraprocess flows is important in the extraction of business rules and in developing independent, single-function components. This level of understanding is critical to optimization of cloud-native microservice approaches, where the extraction of distinct business rules and transformation to new languages and platforms is required.

- **Application categorization** – ModPaaS bridges and processes the top down results of a business assessment with the bottom up results of a technical assessment, providing the ability to confirm, invalidate or refine your target disposition strategies. Categories can be created for a single application, multiple applications, or even an identified sub-domain following a service modelling exercise. Entry points into a category are defined (e.g. screens, transactions, batch jobs, etc.) and the results dynamically generated to show associated dependencies and interactions between objects in the category. From these results you can determine how tightly coupled specific objects in a given category are, what the complexity values of specific objects within a category are, or what the interaction points between multiple categories are.

Figure 3: ModPaaS business rules mining



```
IDENTIFICATION DIVISION.
DATA DIVISION.
PROCEDURE DIVISION.
    Evaluate which are True
        WHEN LS-BAD-INPUT
            MOVE DFHBMASB TO ERMSGLGA
        WHEN LS-WRONG-KEY-PRESSED
            MOVE LS-USER-NAME TO UNAMELGO
            MOVE LS-USER-PASSWORD TO UPASSLGO
    End of Evaluate
    Verify that LS-CURSOR-POS > 0 , and if so then
        EXEC CICS SEND MAP ( 'LOGONMP' ) MAPSET ( 'LOGONMS' ) ERASE FREEKB CURSOR ( LS-CURSOR-POS ) END-EXEC
    Otherwise
    End of Verify
    UPDATE-TIMESTAMP.
```

ModPaaS: Solution Overview for Google Cloud Platform

Post assessment modernization options

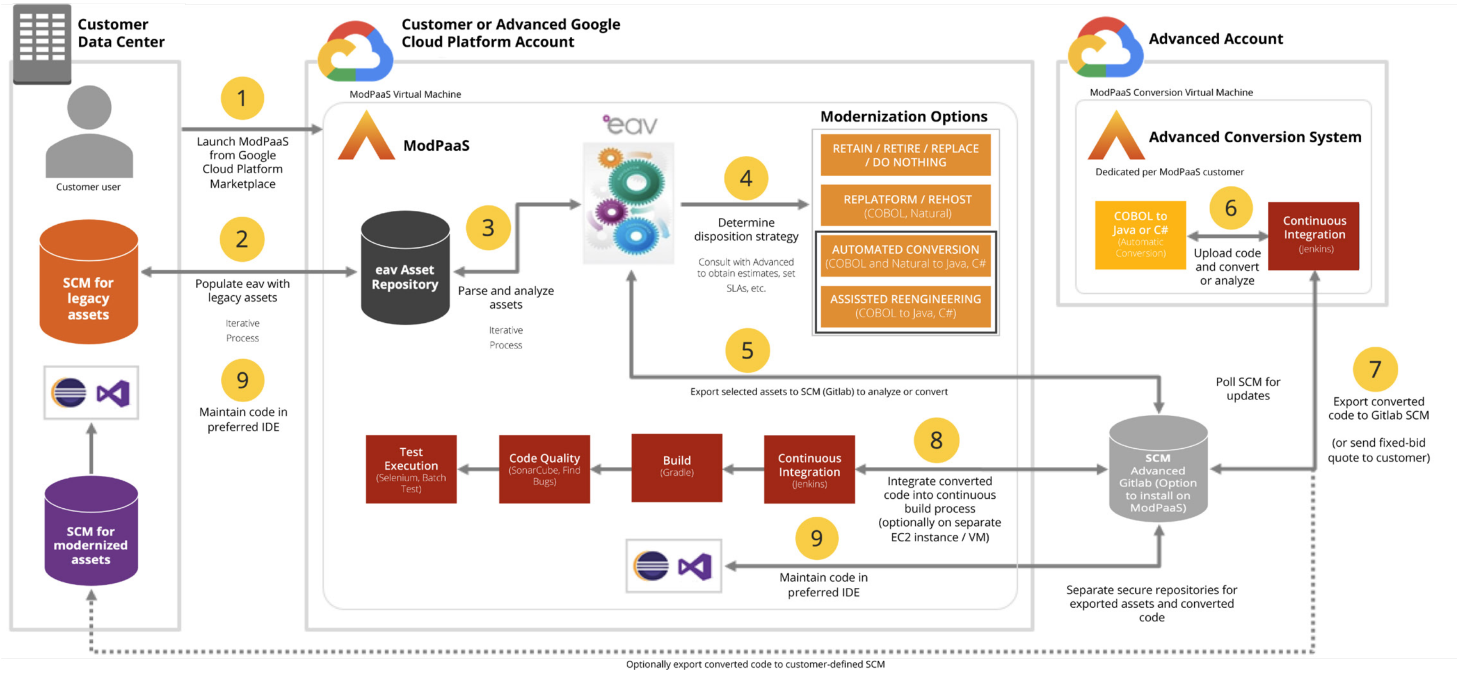
We support multiple modernization options post-assessment which map to the different disposition options and categories identified, including:

- Rehosting COBOL and Natural applications to cloud-based platforms.
- Automatically refactoring COBOL and Natural-based applications to Java or C#, and modernizing legacy data stores to standard or cloud-specific relational databases. Different levels of application refactoring and optimization can be applied to achieve the desired level of elasticity and availability.
- Reengineering selected business functions or application groups to support a cloud-native architecture such as microservices. This option can be driven from ModPaaS with different levels of support from Advanced.
- Determining specific functionality to be retired or replaced with a packaged application. Similar to reengineering options, this can be driven from ModPaaS with different levels of support from Advanced.

You can obtain estimates from Advanced for each different modernization option and the related service tasks – and can decide on the required level of engagement for both parties.

The following image represents the end-to-end modernization approach with ModPaaS. Steps 1 through 4 are specific to using ModPaaS during an assessment and are described previously. Steps 5 through 9 relate to post-assessment modernization activities, where identified COBOL source code is extracted, analyzed further to determine fixed-bid conversion pricing, and automatically refactored to Java or C#.

Figure 4: ModPaaS end-to-end modernization process overview


Customer Data Center
Customer user
SCM for legacy assets
SCM for modernized assets
1
Launch ModPaaS from Google Cloud Platform Marketplace
2
Populate eav with legacy assets
Iterative Process
9
Maintain code in preferred IDE
Customer or Advanced Google Cloud Platform Account
ModPaaS Virtual Machine
ModPaaS
eav Asset Repository
3
Parse and analyze assets
Iterative Process
eav
4
Determine disposition strategy
Consult with Advanced to obtain estimates, set SLAs, etc.
Modernization Options
RETAIN / RETIRE / REPLACE / DO NOTHING
REPLATFORM / REHOST (COBOL, Natural)
AUTOMATED CONVERSION (COBOL and Natural to Java, C#
ASSISSTED REENGINEERING (COBOL to Java, C#)
5
Export selected assets to SCM (Gitlab) to analyze or convert
Test Execution (Selenium, Batch Test)
Code Quality (SonarCube, Find Bugs)
Build (Gradle)
Continuous Integration (Jenkins)
8
Integrate converted code into continuous build process (optionally on separate EC2 instance / VM)
9
Maintain code in preferred IDE
Advanced Account
ModPaaS Conversion Virtual Machine
Advanced Conversion System
Dedicated per ModPaaS customer
COBOL to Java or C# (Automatic Conversion)
6
Upload code and convert or analyze
Continuous Integration (Jenkins)
Poll SCM for updates
7
Export converted code to Gitlab SCM
(or send fixed-bid quote to customer)
SCM Advanced Gitlab (Option to install on ModPaaS)
Separate secure repositories for exported assets and converted code
Optionally export converted code to customer-defined SCM

ModPaaS: Solution Overview for Google Cloud Platform

Steps for refactoring of identified COBOL source code or business rules to Java or C#:

5. Identify select assets to be analyzed and refactored, and export them from ModPaaS into an AdvancedcGitLab SCM repository.

6. Extract assets from GitLab into a ModPaaS conversion system running in an Advanced Google Cloud Platform account. Analyze code to determine refactoring effort and delivery options with fixed-bid pricing. When ready to proceed, execute the code refactoring process, returning code to a level of completeness based on the selected delivery option, ranging from clean compile through to refactoring with functional equivalence. The next release of the ModPaaS solution will support a completely automated and dynamic conversion delivering code with default customization options. Slated to be available later in 2018, this version will not require services or support.

7. Export and populate the refactored code into a target GitLab SCM repository, ready for continued modernization.

8. Optionally create a continuous integration workflow for the refactored code – either on separate system, or existing ModPaaS system.

9. Manage refactored code in preferred development environment and continue modernization efforts.

The following sections provide a brief overview of the different post-assessment modernization options we support.

Automated refactoring

COBOL-based source conversion

One of our core offerings is the automated refactoring of COBOL to Java or C#. We use our proven refactored solution which supports specific asset types such as CICS and IMS COBOL, batch COBOL, and associated assets such as BMS, MFS, JCL, in addition to non-mainframe COBOL dialects. Providing 100% automation is the aim of every automated refactoring project we deliver - this applies to the repeatability of the refactoring process once specific rules and configuration options have been iteratively applied.

We offer different code refactoring service levels based on who is leading versus supporting the refactoring effort, and the required level of automation. A Service Level Agreement is created for each refactored project. All code to be refactored is populated directly from eav into a GitLab repository, and the refactoring process triggered via continuous integration. refactored code is returned to a level of completeness that is mutually agreed upon, ranging from:

> **Base level automated refactoring:** No services or support, delivers refactored code with default customization options.

> **Refactoring with clean compile:** Requires a level of service and support which includes customization and is integrated with your team over a fixed period of time.

> **Refactoring with functional equivalence:** Similar to a traditional project approach offered today. Automatically refactoring COBOL and Natural-based applications to Java or C#, and modernizing legacy data stores to standard or cloud-specific relational databases. Different levels of application refactoring and optimization can be applied to achieve the desired level of elasticity and availability.

A future roadmap item is to provide the ability to customize rules and refactor options via a web-based interface into our refactored tooling, allowing you to lead more of customized refactored effort.

Natural-based source conversion

Another core offering is the automated refactoring of Natural to Java, C# or COBOL using eav. In addition to supporting the refactor of Natural source code, the automated refactoring solution also addresses the replacement of the utilities that are typically part of existing mainframe environments, including Entire Operations, Entire System Server, EntireX, Natural Construct, Natural Security, and Com-plete.

ModPaaS: Solution Overview for Google Cloud Platform

Additional-based source conversion

There is often more than COBOL and Natural applications to convert. You'll most likely have other components such as Easytrieve, SAS, Assembler, PL/I, FOCUS, REXX, etc., which are tightly integrated with the selected COBOL or Natural-based applications you want to convert. This is a perfect example where the collaborative nature of ModPaaS is leveraged. We have additional transformation solutions for other asset types such as IDMS ADS/O, in addition to several conversion solutions/tools for converting assets such as Assembler, CA Gen, CA-Telon, CA-Easytrieve, PL/I and VA Gen to Java or C#.

Optimization

Applications can be further refactored and optimized further during an automatic refactoring project in order to achieve the desired levels of elasticity and availability. Legacy workloads are stateful by nature, and will therefore require modifications to be able to take advantage of the stateless nature of a cloud environment. During the assessment we will work with application SMEs to identify and discuss areas where state may be stored on the application tier. Possible recommendations may include leveraging a distributed caching layer to support state, and moving any local file access to a scalable shared file system.

Reengineer

Your ultimate goal might be to distill your existing monolithic workloads down into a set of loosely coupled microservices. This is an extremely complex task considering you're likely dealing with millions of lines of tightly integrated code. While there's no magic bullet, eav is perfectly suited to aid in the definition and creation of the microservices. It's obviously important that you first scope and model the required microservices using proven top-down analysis driven approaches and techniques such as Event Storming and Domain-Driven Design (DDD). These workshop-style sessions are led by the business team during the Assessment phase, and should incorporate user experience.

Eav is used to analyze and validate the top-down results, and support the creation of new microservices functionality. With eav, you can:

- Define application or sub-domain categories with entry points you define, and from each entry point automatically generate top-down relationships highlighting the dependencies between elements (i.e. called programs, copybooks, etc.) required to support the scoped microservice function. Once aligned with the bounded context results of a Domain-Driven Design, the required element scope can be further refined for the specific microservice function.
- Automatically generate Java or C# code for any functionality deemed feasible to be reused. You can also aid any new microservice development efforts by converting selected code such as the UI layer and data access objects.
- Trace and isolate COBOL-based business rules, then save as COBOL ready for refactoring to Java or C#. Automatically refactoring COBOL and Natural-based applications to Java or C#, and modernizing legacy data stores to standard or cloud-specific relational databases. Different levels of application refactoring and optimization can be applied to achieve the desired level of elasticity and availability.

Rehost (also known as Replatform)

We support the rehosting of Software AG Adabas/ Natural in addition to standard online and batch COBOL-based workloads. Our ATP solution rehosts Adabas/Natural based workloads. Adabas is converted to a relational database such as Db2, SQL Server or Oracle, while the ATP solution enables the Natural code to be seamlessly moved. ATP interprets and executes each Natural command, and the applications can execute against the new relational database without changes to the Natural syntax. We also provide a standard COBOL rehosting offering backed by resources with extremely deep experience, having been responsible for delivering some of the largest rehosting projects worldwide over the past 20+ years.

ModPaaS: Solution Overview for Google Cloud Platform

Replace

Using eav you can identify and isolate application groups targeted for replacement with a packaged application (COTS) – and in turn determine the net-effect caused by the dependency on other applications. Our Data Migration and Data Archiving solutions are used to move to COTS solutions to retain, access, and leverage historical mainframe data.

Ongoing development of refactored code using ModPaaS

Refactored Java or C# code is fully maintainable, following object oriented concepts and paradigms such as encapsulation, abstraction, modularization and loose coupling. Code is populated into a target GitLab SCM repository, and can then be optionally integrated with a continuous integration workflow. We can support you in building a default continuous integration workflow with ModPaaS based on our experience with solutions we use internally, e.g. Jenkins, Gradle, SonarQube, Selenium, etc. This can obviously be customized depending on your own preferences. The workflow is completely independent of any Advanced specific solutions once the code is in the target repository. The focus becomes on configuring the workflow to extract code based on a specific trigger (e.g. SCM update), then automatically running the code through select solutions to support features such as:

Code quality measurement

We can supply an initial set of SonarQube rules with some limitations related to the original application. For example, with automated refactoring, it's not possible to reduce the cognitive complexity of a program. Depending on the selected code conversion service level, we can also improve the quality of the refactored code by incorporating your suggested changes, even from code coming as-is from the COBOL program. In addition to analyzing and measuring code quality against defined SonarQube rules, we also recommend using an additional open source static code analyzer solution such as FindBugs to detect possible bugs in Java programs.

Automated test execution

We use Selenium as part of the continuous build cycle to support the execution of created test cases. Since Selenium test cases need to be created after the application is running, the tests won't trigger on the initial translation. Once created though they can be added to the build cycle, and Jenkins will trigger the build using a solution such as Gradle. The Gradle build project packaging will include specific JUnit (for Java) or NUnit (for C#) test cases to support batch job testing and "headless" screen testing, in addition to the Selenium test cases to be triggered. This is a process that our development teams utilize internally today.

Leverage a target IDE

Refactored code can be exported from the target SCM and managed in your preferred IDE, e.g. Eclipse, Microsoft Visual Studio, IntelliJ IDEA, etc. You're responsible for installing and configuring third party solutions such as an IDE – and if applicable, for supplying required licenses.

ModPaaS future roadmap

> Offer customizable modernization options for target refactored source code and build environment, e.g. input into COBOL to Java or C# code refactoring options, selection of a target SCM (other than the default GitLab), continuous integration and build solution (other than Jenkins and Gradle, respectively).

> Expanded language support for Business Rules Mining, e.g. support for multiple COBOL dialects and support for features such as Micro Focus COBOL screen definitions.

> Add non-COBOL language convertors to Automated Refactoring capability.

> Provide plugins for the select target development environment (e.g. Eclipse, MS Visual Studio) in order to map modernized assets back to the eav repository.

ModPaaS: Solution Overview for Google Cloud Platform

ModPaaS accessibility

ModPaaS is available from the Google Cloud Platform Marketplace. You're responsible for managing and limiting access to the ModPaaS environment using company and/or Google Cloud Platform recommended security best practices, such as implementing Active Directory Domain Services and granting the least amount of access with least privileges, etc. Once within the ModPaaS environment, only users with specific authorization can gain access to eav (and the eav repository) and view specific source code.

More information

w modernsystems.oneadvanced.com **e** hello@oneadvanced.com

UK +44 0333 230 1884

Ditton Park, Riding Court Road Datchet,
Slough, Berkshire, SL3 9LL

US +1 855-905-4040

3200 Windy Hill Road, Suite 230 West,
Atlanta, GA 30339