G4 - Automated Code Conversion Tool

White Paper / FAQs

Date: April 17th, 2023
The intention of this document is to provide a high-level overview of Google’s G4. An automated code conversion tool used for mainframe modernization. This document is intended to provide an introduction to our refactoring solution, the benefits this will bring to customers, detail how we engage with customers to deliver this solution to meet their objectives and answer frequently asked questions. Below are the main description points we will cover in this document.

**Table of Contents**

- **What is G4?** 2
- **What are the benefits?** 3
- **What challenges does this solve?** 4
- **Industry leading technology from Google** 5
- **Google’s approach** 5
- **What infrastructure changes are required?** 6
- **What infrastructure is recommended?** 7
- **G4 Support Stack for z/OS** 8
- **G4 Key Features for iSeries** 9
- **Recommended Architecture** 10
What is G4?

G4 is Google’s modernization solution for customers looking to refactor their Mainframe or Midrange applications. Developed by Cornerstone Technology and acquired in 2020, G4 automates the code conversion process by providing tools to transform legacy program languages like COBOL, PL/1 or RPG into modern Java.

The G4 Platform is both programming language- and platform-independent and can handle the most common legacy languages. Customers taking advantage of this approach to legacy modernization have seen significant business agility improvements and cost savings.

We recognize the fact that our customers all have a unique combination of languages, database systems, transaction monitors, and schedulers. This solution takes into consideration all of these different unique characteristics.
What are the benefits?

Converting your legacy application to run on a cloud-native environment eliminates all dependencies on legacy technologies and skills, while creating cultural change for the organization to move into a more modern approach.

G4 allows for more innovation in the following areas:

- Cloud-ready solutions that can be run and be managed in containerized environments
- Access to a larger pool of Java developers and modern tooling
- Access to leverage Google Cloud’s reliability, scalability, and elasticity
- Ability to granularly maintain, deploy, and scale services
- Large run cost savings compared to Mainframe
- The agility customers need to create a new product or neutralize a competitive threat in the market
- An accelerated approach to unlocking mainframe data compared to a rewrite
What challenges does this solution solve?

G4 is targeted to solve for the following challenges customers face today:

- Lack of mainframe skills in the customer’s organization and in the market
- Lack of SMEs available within the customer’s organization
- Inability to meet market demands in terms of business agility and time to market
- Cloud readiness and transitioning to a modern language, tooling, and skillset
- No funding to spend on a rewrite
- High cost of running on Mainframe
- Wanting to leverage services available in Google Cloud
Industry leading technology from Google

ISG, a leading analyst in the industry, named Google’s G4 as a leader in Mainframe modernization technology in 2021 and 2022. This year 2023, Google continues to lead the industry in competitive strength of Mainframe modernization solutions.

Google’s approach

As the first step in a modernization journey, Google recommends starting with an assessment where we will conduct a deep-dive analysis of the application environment—including the applications, their dependencies, the languages, and associated databases. With that knowledge, we can then derive the recommended modernization solution. The important thing to keep in mind is that there is certainly no one-size-fits-all solution for every environment.
The information gathered during the engagement will provide a blueprint for the journey from legacy to Google Cloud, and provide any supporting financial and business cases for project approval. The assessment engagement requires access to customer-owned source code to analyze the interdependencies and understand technical debt. After completing the assessment, customers can get a proof of concept (POC) to prove the capabilities. Detailed below is a typical engagement process we follow for a G4 approach with customers.

What infrastructure changes are required?

G4 doesn’t require any infrastructure changes on the mainframe. If the end solution requires connectivity back to the mainframe, then it requires TCP/IP connectivity to the mainframe from Google Cloud. Cloud connectivity options should be evaluated and selected based on customer requirements.
What infrastructure is recommended?

To run the converted applications on Google Cloud, we recommend Google Kubernetes Engine (GKE), GCS, and Filestore. Since the converted application is containerized, depending on the application requirements, you can execute the app in a serverless infrastructure such as Cloud Run.

### Recommended architecture for production

*could change based off of customers requirements/choices

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GKE Regional Cluster</td>
<td>Hosts highly available modernized application after conversion</td>
</tr>
<tr>
<td>Filestore Enterprise</td>
<td>Shares files between batch jobs running on GKE</td>
</tr>
<tr>
<td>Google Cloud Storage</td>
<td>Archives data, such as files and backups</td>
</tr>
<tr>
<td>Apigee or API Gateway</td>
<td>Consumes/exposes services</td>
</tr>
<tr>
<td>Cloud SQL / Spanner / AlloyDB</td>
<td>Database of choice</td>
</tr>
</tbody>
</table>

### Recommended architecture for POC

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMs (w/ persistent disk)</td>
<td>Runs docker images and stores output/input files</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Data purposes</td>
</tr>
</tbody>
</table>
G4 Support Stack for z/OS

The following diagram details individual legacy technology components and the recommended corresponding mapping of the target state technology.

G4 key features for Mainframe

- **Scheduler Integration**: Ability to invoke jobs in GKE from external schedulers like CA-7, CTL-M etc including restart, DD override, return codes and symbols.
- **REST Enabled API’s**: CICS Commerce Programs, COBOL callable routines can be REST enabled.
- **Angular Frontend**: BMS and IMS MFS are converted to Angular front end.
- **Containerized Workloads**: Programs are packaged as containers and deployed in GKE.
- **VSAM, DB2 & IMS DB, GDGs & QSAM**: VSAM, DB2 and IMS DB are supported in RDBMS format. Handle GDGs and QSAM.
- **COBOL, JCL, PL/1, CICS, IMS DC**: Support for mainstream z/OS software.
- **IBM Utilities support**: Support for IBM Utilities like SORT, IEEGENER, IDCAMS etc.

*IMS DC currently in development*
G4 Key Features for iSeries

The following diagram details individual legacy technology components and the recommended corresponding mapping of the target state technology.

G4 key features for iSeries.

- **Scheduler Integration**: Ability to invoke jobs in GKE from external schedulers like CA-7, CTL-M etc.
- **REST Enabled APIs**: RPG and CL callable routines, can be REST enabled.
- **Angular Frontend**: DSPF files are converted to Angular frontend.
- **Containerized Workloads**: Programs are packaged as containers and deployed in GKE.
- **Data area, Data Queue, DDS, PRTF**: Support for Series constructs.
- **RPG, CL, Free Format, MQ**: Support for mainstream iSeries software.
- **Utility to generate DDL**: Utility to generate DDL.
Recommended Architecture

Google’s approach to code conversion is to ensure that the converted code is dockerized and run in a container orchestration platform. This makes the converted code portable. The recommended target platform for the converted code is GKE due to the fact that batch jobs will create files and these files might need to be shared across multiple containers and require Google Cloud’s Filestore or an equivalent NFS file system. As depicted in the diagram below, the target GKE environment provides high availability, scalability, and recoverability. The diagram also demonstrates how to integrate with the mainframe, including scheduler.
Thank you

For questions, examples, references:

Please contact mainframe@google.com

Google Cloud Consulting