cormac
HEALTH DATA ANALYTICS

Redesign Legacy Mainframe System to AWS Cloud using Microservices – Prototype for Redesigning Complex Medicare Final Action Claim Logic

Onsite Mainframe systems have always been an unavoidable cost for any business.HHS/CMS maintains complex systems that are operating on IBM Mainframes today. These Mainframe systems are running the complex Medicare Payments and Data Systems that require large-scale infrastructure, and trained personnel to support and maintain the system with higher cost of ownership.

CORMAC has been the prime contractor for HHS/CMS maintaining the Medicare Claims systems in IBM Mainframe including MQA, NCH Suite of Systems, MEDPAR,NMUD and MSIS for the past 7+ years. We have in-depth expertise with these complex mainframe systems.

We developed a prototype in our Innovation Lab to redesign one of the functions of MEDPAR: the Final Action Claim. We used microservices and data API's using Open source in the AWS Cloud. These microservices can be used as plug-and-play code based architecture so that they can be put together in varying orders to achieve the desired functionalities. This approach will minimize the complexity of the code resulting in decreased maintenance requirements.

Microservices Based Architecture on Cloud

The Advantages of this architecture are easier maintenance and lower cost to build and support, as well as faster performance ,better redundancy,and cutting edge security.

CONTACTS

Dominic Raj
President
703.793.0931 ext. 103
703.568.1292 (Cell)
dominic.raj@cormac-corp.com

COMPANY DATA

NAICS Codes: 541511, 541512, 541513, 541519, 541611, 541690, 541990, 518210, 541614, 541618, 519130
Certifications: WOSB, SDB
Contracting Vehicles: GSA IT 70, SPARC, STARS II
EIN #: 13-4205991
DUNS #: 122154771
CAGE Code: 38BU8

LOCATIONS

Virginia: 2201 Cooperative Way, Suite 600 Herndon, VA 20171 • 703.793.0931
Maryland: 5950 Symphony Woods Road, Suite 500 Columbia, MD 21044 • 443.864.5880

Salient Features and Benefits:

• Reusable microservice architecture that builds the foundation for a generic Plug and Play architecture for other components.

• Migrating file storage from physical tape drives to S3 bucket technology will speed up data storage and retrieval, eliminating workflow bottlenecks and resource contention.

• Final Action Rules are configured dynamically in a JSON file which enables the user to add runtime conditions.This will result no or fewer code changes with reduced maintenance cost.

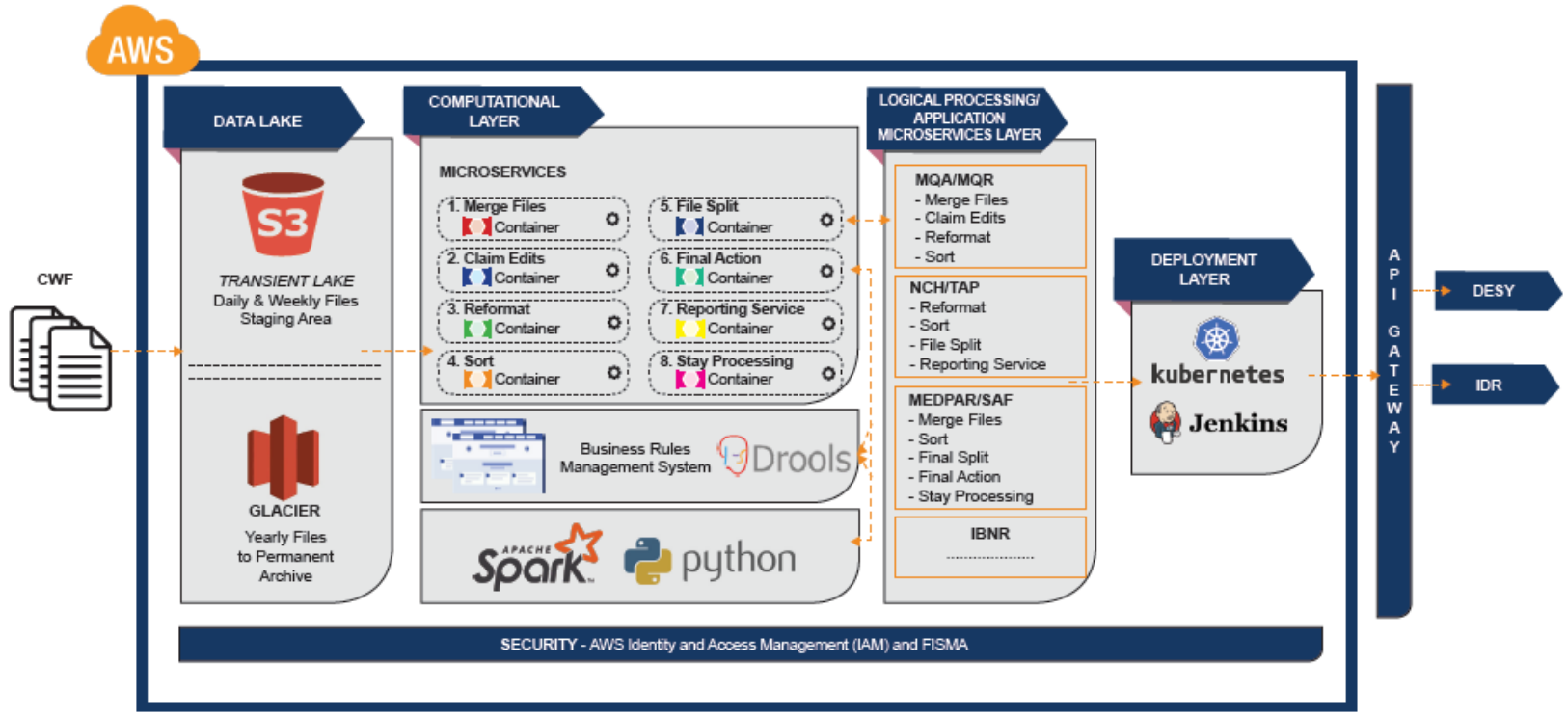
• API Gateway that allows easier accessibility of data to other sources like DESY and IDR.

• Containerized Applications that are DevSecOps ready and allows fast deployment with ability to respond to change and need rapidly.

Proposed Architectural Framework

CORMAC's proposed Architectural Framework is a MicroServices based architecture using a modern data framework including Data Lake for processing files, and integrated with DevOps tools for faster integration and deployment.

Conceptual Architecture



cormac
HEALTH DATA ANALYTICS

CMS Challenges	CORMAC Solution
Legacy systems business logic is written in COBOL, with no centralized inventory. Changes from upstream systems have to be evaluated and manually implemented across multiple systems.	New Data Lake - Microservice architecture creates **centralized management of business rules from within business rule engines**. Changes from upstream systems will be implemented in only **in business rule engines**.
Data stored in the business logic of legacy systems use custom COBOL code with no centralized inventory for users to understand the custom codes. Maintaining changes is costly as a result of upstream systems such as Common Working Files (CWF) and/or Shared Systems. Changes to the claim record cause system-wide impact because the programs and the structure (COBOL Copy books) have a shared dependency.	The new architecture removes **business data (**HIPCS codes, DRG codes etc.) as hardcoded values. Data will be maintained in separate data stores with periodic changes from upstream systems managed in a centralized location with no systems impact. With data lakes, the micro services independent of claim data structure (data dictionary). **Future changes to the data directory may be implemented with minimal effort and without system-wide changes.**
Redundant copies of data exist across data repositories with no function to track changes between the two over the data lifecycle.	**The proposed approach is to remove redundant data** by removing intermediate quarterly files once the annual files are created. **All the other versions could be derived from this final annual dataset.**
Legacy systems produce data extracts in variable block format or fixed block format. It relies heavily on COBOL and mainframe processing. These datasets are not easily accessible with modern Business Intelligence(BI) tools.	Claims from the CWF will be transformed from variable block to a standard fixed block format (length will be the maximum allowed in claim structure). All of the datasets generated thereafter will be only in fixed block format which can be accessed easily with any modern BI tool.
NCH datasets are stored in virtual tapes and are not readily available for analytics.	Data will be stored in AWS S3 buckets, available with near-real-time experience for analytics using In-memory computing (SPARK/EMR).
Active data access is inefficient due to lack of archival strategy.	**Updated archiving methods offer reduced storage costs without reducing the processing efficiency.** Infrequently used data will be stored in Amazon Glacier. which can be brought up to the active S3 zone as needed. This reduces archival access time from days to hours (manual fetch of archived mainframe data vs. Amazon Glacier/S3)