Leveraging Data Integration using APIs for Enhanced Data Analytics in Papua New Guinea

Steward Kalan, Rajendran Bhojan

School of Mathematics & Computer Science, Papua New Guinea University of Technology, PMB 411 Lae, Papua New Guinea.

*Correspondence Author email: rajendran.bhojan@pnguot.ac.pg

Abstract: Data is the "New Oil". In a fast-evolving technological world where the survival of organizations mostly depends upon technology, data is the catalyst that is driving business optimization, growth and success. Most successful organizations are data driven. They capitalize on the magical spell of data analytics. Data analytics plays a pivotal role in enforcing well informed decisions in strategic marketing and planning, new product development and other shifts that drive business growth and development across various sectors. Unfortunately, in Papua New Guinea (PNG), due to the splintered nature of diverse data systems and the incapability of interoperability poses significant barriers to effectively capitalize on the strengths of data analytics. Due to the involvement of traditional manual processes and the existence of data and systems in silos, the integration of data and advance data analytics remains a major challenge. This research explores the integration capability of Application Programming Interfaces (APIs) and the potential of leveraging it to address these challenges. It involves a mixed methods approach, consolidating qualitative interviews with data science and analytics professionals and quantitative analysis of case studies from both local and international organizations. It assesses the current state of data integration and analytics in PNG, taking into consideration the limitations of traditional and manual processes of data exchange methods. Significant findings have shown that API-driven data integration presents advantages for growth and efficiency. The study does not only reveal and promotes the rich advantages of API, instead it acknowledges and uncover the challenges that are to be attended for a successful implementation. It proposes a practical framework that is workable in PNG context for API adoption in data integration, which includes a customize architectural design for API Integration. The significance of securing APIs and the transmission of data safely between different systems is echoed in the proposed architectural design. The research exposes the transformative potential of APIs in the space of data integration and analytics in PNG. By embracing API-based data integration, diverse organizations can unlock more reliable and timely insights, ultimately supporting data-driven development and policy-making in the country that are based upon reliable data.

Keywords: Data, Analytics, API, Integration, Interoperability, Silos

Presenter or Main Author Biography: Steward Kalan is currently a Master's of Philosophy student in School of Mathematics and Computer Science, PNG University of Technology. He is Data Analyst by profession. He has been working in the industry as Data Analyst for four (4) years. For the four (4) years of industry experience, he has been working with National Superannuation Fund (NASFUND) and Bank South Pacific (BSP) or BSP Financial Group. He is currently serving as a Data Analyst with BSP, attached with the Central Data Analytics Business Unit. Mr. Kalan holds a Bachelor's Degree in Computer Science from the Papua New Guinea University of Technology. His main research work focuses on Leveraging Data Integration using APIs for Enhanced Data Analytics in Papua New Guinea.

1. INTRODUCTION

In today's data-driven world, the ability to access, analyze, and interpret data in real time has become vital for informed decision-making and strategic planning. Across the globe, organizations are increasingly adopting

modern data integration techniques to streamline operations and uncover valuable insights. One such method is through the use of Application Programming Interfaces (APIs), which allow seamless communication and data exchange between different systems, databases, and applications. This research focuses on how leveraging of data integration through APIs can significantly enhance data analytics capabilities in Papua New Guinea (PNG) - a nation that is gradually embracing digital transformation.

Data analytics in Papua New Guinea is often challenged by data silos, inconsistent data formats, and manual data handling processes, just like any other developing countries and some developed countries as well. Many government departments, businesses, and institutions still rely on legacy systems that lack interoperability. As a result, decision-makers face delays and limitations in accessing real-time, accurate data or most times, decisions are made or passed without data to justify their resolutions to a certain problem (decisions are not data driven). APIs provide a viable solution to these issues by enabling various systems to connect and share data in a standardized and automated manner (Stylos, J., 2009). When effectively implemented, APIs can bridge the gap between isolated datasets and enable integrated platforms for better data visibility, consistency, and reliability.

Globally, industries such as finance, healthcare, education, and logistics have already adopted API-driven integrations to optimize their operations and provide better services. These integrations support functions such as real-time data syncing, automated reporting, predictive analytics, and cross-platform access to key information. Papua New Guinea stands to benefit greatly from adopting similar technologies, particularly in areas like public service delivery, health information systems, financial reporting, and national monitoring and planning. The growing use of cloud-based services and mobile technologies in PNG further underscores the importance of API-enabled platforms for scalable and sustainable data integration solutions.

However, the adoption of APIs in PNG is still in its infancy, hindered by several factors including limited technical expertise, inadequate infrastructure, lack of data governance frameworks, and concerns over cybersecurity. These challenges necessitate a strategic approach to building capacity, enhancing infrastructure, and developing policies that support secure and scalable API deployment (Zibran, M., 2008). Additionally, collaboration between the public and private sectors can play a pivotal role in advancing data integration efforts and promoting data sharing across key sectors.

This paper presents the potential of API driven data integration as a transformative tool for enhancing data analytics in Papua New Guinea. By examining case studies, current practices, and technical frameworks, the study aims to highlight both the opportunities and challenges associated with API adoption. It will also provide actionable recommendations for stakeholders including policymakers, ICT professionals, and business leaders on how to implement and manage API ecosystems effectively to foster data-driven decision-making and national development.

In conclusion, as Papua New Guinea continues its journey towards digital transformation, embracing modern data integration techniques such as APIs is essential. Doing so will not only enhance analytics capabilities but also empower institutions with timely, accurate, and comprehensive data to address the nation's unique development challenges and opportunities.

2. DATA

In a fast-evolving technological world, data is the new oil. It is raw, unprocessed facts and figures that are collected from various sources for analysis, storage and processing. Have you ever wondered how your phone knows which ads to show you at lunch time? Or how the supermarket emails you with specials on your favorite products? It is not a coincidence. It is the work of data being put to use, encouraging you to buy more. Data is a sleeping giant, the maximization of its capabilities is like stepping into a portal of business improvement, success and growth in both private and governmental entities.

Data comes in different forms. Some of these forms include: numbers, texts, images and sensory readings but are not limited to these. Data is pivotal in this age rapid technology involvement in daily operations and strategy development as it drives decision making, supports innovation and enables insights through processes like:

- Data Mining
- Machine Learning

- Data Analytics

By nature, data can be classified into two (2) main categories, data can be either quantitative (numerical) or qualitative (descriptive). From these categories, data can be presented in two (2) types:

1. Structured
   Structured Data is organized in predefined formats, often stored in databases or spreadsheets, making it easily searchable and analyzable.

2. Unstructured
   Unstructured Data are such data as emails, social media posts and videos, does not have a specific format. This type of data requires sophisticated tools and techniques to analyze and derive insights for decision making. The proper collection, management and analysis of data can lead to powerful insights, helping organizations to optimize operations, predict trends and improve decision making. This paper discusses about both qualitative and quantitative data category that of structured data type.

3. DATA INTEGRATION

Data Integration is the process of combining or harmonizing data from different sources into a unified and coherent format that can be easily accessible for various analytical, operational and decision-making purposes.

In a growing digital landscape were data and the analytics capability are being prioritized by organizations for various business requirements, organizations typically cannot function without gathering data from different sources like databases, apps, spreadsheets, cloud services and others. Furthermore, the storage of data in different locations and formats contributes to the variation in the levels of data quality, which leads to data silos and inconsistencies.

The data integration process aims to overcome these challenges by bringing together data from disparate sources, transforming it into a consistent structure and making it accessible for analysis and decision making. There are different types of data integration in existence. The type of data integration chosen to be used differs as it is based on the needs and requirements of the stake holder. Here are some types of data integration:

1. Data Consolidation
   This includes the ETL (Extract, Transform, Load) Process, where data is extracted, transformed and loaded or merged into a single centralized database or data warehouse from various sources.

2. Data Federation
   Instead of physically moving data, this approach is done by the creation of virtual database that enables users to access or query multiple data sources as one. This approach works in databases when real time access is needed.

3. Data Propagation
   This approach involves the replication or migration of data between databases in real-time or near real-time to ensure all the system in a distributed system have the latest data available.

4. Master Data Management
   It gives priority in ensuring consistency in key data across an entity. This approach is all about maintaining single source of truth for essential business data to avoid discrepancies.

5. API Based Data Integration

This approach is common on cloud-based services or applications where APIs are used to allow different systems or applications to communicate and exchange data seamlessly with a secured connection.

6. Big Data Integration
   For large scale data integration that handles vast amount of unstructured or semi structured data.

Listed above are just few of the many approaches that are used for data integration. This paper will discuss about the API Based Data Integration and its benefits in the space of integration.

3.1 Benefits of Data Integration

Data Integration is not just a concept just for discussion, instead it is a concept that have a lot of benefits in the data world. Here are some of the most pivotal and key benefits of data integration:

1. Improved Data Quality
2. Enhanced Decision Making
3. Streamlined Operations

4. APPLICATION PROGRAMMING INTERFACE (API)

An Application Programming Interface (API) is a set of protocols, tools and definitions that allow different software application to communicate with each other. APIs define method and data structure that developers use to interact with the software component, be it an operating system, library or a web service.

- Application: The Software that uses or provides the API
- Programming: The use of code to interact with the API
- Interface: Point of interaction where the software can request services or data

4.1 Types of API

Different types of APIs are used for different functionalities. Listed here are four (4) basic types of APIs:

1. Web APIs
   The most common, allowing web applications to interact via HTTP Requests (e.g. REST and SOAP) further elaborate these two.

2. Operating System APIs
   Allow interaction with system level operations (e.g. Windows API, POSIX)

3. Library APIs
   Provide functionality from libraries (e.g. graphics, machine learning)

4. Database APIs
   Enable interactions with databases (e.g. SQL queries through drivers)

4.2 How APIs Work

APIs expose specific functionality of an application via endpoints (URLs) or functions. When a developer sends a request, the API processes it, performs the requested operation (e.g. retrieving data) and then returns a response in a structured format (often JSON or XML). APIs are essential in modern web development as they allow

developers to connect different services or applications. For example, using an API, a frontend application can communicate with a backend server to fetch or send data without directly accessing the database.

4.3 Key Benefits of API

What makes APIs the best approach or concept to be used in different functionalities? Here are some of the key benefits of using APIs:

- Modularity
  Different parts of an application can evolve independently. APIs allow developers to reuse existing services and functionality rather than building everything from scratch

- Interoperability
  APIs enable communication between different technologies and platforms.

- Scalability
  APIs can be reused, making it easier to expend of modify software. APIs allow applications to grow by adding or modifying features without disturbing the existing code.

- Third Party Integrations
  APIs enable integration with third-party services like payment gateways, social media platforms and cloud services.

- Data Sharing
  APIs allow different applications to share data seamlessly (e.g. fetching data from a weather API)

This paper writes about WEB APIs by maximizing on the benefits that APIs possess. In a technology and data driven society where different institutions use different systems with data kept in silos, the benefits of using APIs does stand out. The benefits of using APIs provides efficiency and that is why the usage of APIs in Data Integration is instrumental in PNG's data ecosystem.

5. API BASED DATA INTEGRATION

APIs are widely used for Data Integration across different systems, applications and platforms. Data integration involves combining data from various sources into a unified view and APIs play a crucial role in facilitating this process. Here's how APIs are typically used in data integration.

5.1 Common API Types for Data Integration

Here are some common types of APIs are typically used in data integration:

1. RESTFUL APIs
   Representational State Transfer (REST) is the type of API that is most commonly used for Data Integration. RESTFUL APIs uses HTTP Methods (GET, POST, PUT, DELETE) to perform CRUD (CREATE, READ, UPDATE, DELETE) functionalities. It is popular in Cloud Services, Web Applications and Micro Services and data is exchanged in JSON or XML format.

2. SOAP APIs
   Using XML Messaging, Simple Object Access Protocol (SOAP) APIs are well structured and are secured and robust than REST APIs.

3. GraphQL APIs
A query language for APIs that uses a more specific approach which require clients to specify the exact data they need.

4. Database APIs
Ensures access to database through standard queries. It allows CRUD Operations directly with tables in a database.

5. Streaming APIs
Handle continuous data flow, such as real time integration from IoT devices, social media streams of financial markets.

6. File-Based APIs
Allows transferring of files between systems (e.g. FTP APIs, SFTP APIs) and is mostly used for base data integration tasks where large datasets are uploaded and downloading periodically.

7. Platform-Specific APIs
This is a Cloud API (Google Cloud, AWS, AZURE) that is often used for integrating cloud resources and services.

This paper presents RESTFUL APIs as a most suitable API for integration of data in PNG's data ecosystem (Costa, P. 2014).

5.2 Key Roles of APIs in Data Integration

Here are some roles that APIs play in Data Integration:

- Data Synchronization – ensuring data consistency
- Data Aggregation – Pull data from different sources and unify them
- Data Migration – moving of data from one system to another
- Automation and Orchestration – automations of data workflows
- Real-Time Integration – enables real-time data transfer and integration

6. DATA ANALYTICS

Data analytics is essential for businesses and organizations in today's data-driven world (Sarker, I. H., 2021). It helps derive actionable insights from vast amounts of data and provides a competitive edge through better decision-making, efficiency, and customer insights. Depending on the type of analysis used, descriptive, diagnostic, predictive, or prescriptive, organizations can understand past trends, investigate causes, forecast future events, and prescribe the best actions. As businesses continue to generate more data, the role of data analytics will only grow in importance, enabling more intelligent, data-driven strategies across industries.

Data Analytics refers to systematic computational analysis of data. It involves transforming raw data into useful information that supports decision making. Data analytics enables businesses to uncover trends, measure performance and identify actionable insights that leads to improve outcomes (Sarker, I. H., 2021). It encompasses various techniques and methodologies from statistical analysis to machine learning.

Data Analytics involves examining raw data to extract meaningful insights, patterns and trends. The process technically includes the following key steps:

a. Data Collection
b. Data Cleaning

c. Data Exploration
d. Data Modelling
e. Data Visualization
f. Interpretation and Reporting

6.1 Different Types of Data Analytics

There are different types of analytics. Depending on the business requirements, specific type of analytics is used to attain the specific results to meet business demands. Here are the four (4) main types of data analytics:

a. Descriptive Analytics
b. Diagnostic Analytics
c. Predictive Analytics
d. Prescriptive Analytics

6.2 Key Benefits of Data Analytics

Given the nature of how institutions are operating these days, data analytics is a demanding and growing field. Most successful organizations are data drive or data centric as they prioritize the analyzation of data. Here are few key benefits of data analytics:

1. Improve Decision Making
2. Increase Efficiency
3. Enhance Customer Insights
4. Risk Management
5. Innovation and Product Development
6. Cost Reduction
7. Competitive Advantage
8. Personalization
9. Enabling business to offer personalized products, services and experiences.

7. DATABASES

In this approach, the API will be used to transfer structured data in a database from one system to another. Hence, there will two (2) databases:

1. Source Database
2. Destination Database

The conceptual and logical design of the two (2) databases will be identical. There will be two RESTFUL APIs in each of the databases for enabling communication and data transfer or integration.

8. RESTFUL API DATA INTEGRATION PROCESS

Firstly, the integration of data in this paper involves two (2) databases (databases of two different systems) as mentioned above, source database and destination database. The RESTFUL APIs in this paper are developed using Python Programming for simulation purposes. The security measures in this RESTFUL APIs approach using Python Programming as following:

- AES-256 Encryption protects the data during the process of transferring [9].

- JWT Authentication make sures that only the authorized requests are the processed in the APIs and nothing foreign.
- Initialization Vector (IV) ensures encryption remains secure even if the same data is sent multiple times.
- Base64 encoding ensures data is safely transmitted over HTTP.
- Token Expiration within an hour prevents long term unauthorized access.

This part of the paper provides a step-by-step explanation of how data is transferred from one system to another or integrated.

8.1 Destination System - Requesting Authentication Token

According to the automation schedule that is set on the destination system or database, the process kicks from that side. The Destination System requests for an Authentication Token. It is a request from Destination System that is sent to the Source System.

```python
# Function to get JWT token
def get_jwt_token():
    try:
        response = requests.get(TOKEN_URL, verify=False)  # verify=False for self-signed certs in local dev
        if response.status_code == 200:
            return response.json().get("token")
        else:
            log_transfer("FAILED", "Token retrieval failed.")
            return None
    except requests.exceptions.RequestException as e:
        log_transfer("FAILED", f"Token request error: {e}")
        return None
```

```python
# Dummy JWT token endpoint (placeholder for real auth)
@app.route("/token", methods=["GET"])
def get_token():
    return jsonify({"token": "dummy-token"})
```

Fig. 1 Destination and Source System Requesting and Generating JWT Token

Explanation:

- Destination System sends a request to System A.
- Source System generates JWT Token that is valid for an hour.
- Source System returns the token to Destination System

Example:

An example of a response from the Source System sending the generated JWT Token

```
{
  "token": "eyJhbGciOiJIUzI1NiIsIn..."
}
```

8.2 Destination System – Requests Encrypted Data from Source Database

Through the JWT Token that was generated, the two systems can now communicate and exchange data or integrate it.

```python
# Function to request data with retries
def fetch_encrypted_data_with_retry():
    token = get_jwt_token()
    if not token:
        return None
    headers = {"Authorization": f"Bearer {token}"}
    for attempt in range(5):
        try:
            response = requests.get(API_URL, headers=headers, timeout=10, verify=False)  # verify=False for local dev
            if response.status_code == 200:
                return response.json().get("encrypted_data")
            else:
                log_transfer("FAILED", f"Error {response.status_code}, Retrying...")
        except requests.exceptions.RequestException as e:
            log_transfer("FAILED", f"Request error: {e}, retrying in {2 ** attempt} seconds...")
            time.sleep(2 ** attempt)
    return None
```

Fig. 2 Destination System requests Encrypted Data from Source System

Explanation:

- Destination System uses the JWT token in the request header to access the data.
- Destination System sends a data request to Source System via the APIs
- Source System verifies the JWT Token and fetches data from the database

8.3 Source System Encrypts Data with AES-256

Data is a very crucial asset that needs to be protect and secured from theft and data bridges. Hence, encryption and decryption are very important as part of data integration.

```python
AES_KEY = os.getenv("AES_KEY").encode()  # 32 bytes for AES-256
AES_IV = os.getenv("AES_IV").encode()    # 16 bytes IV
```

Fig. 3 Source System Data Encryption

Explanation:

- Fetches the data from its database
- Convert it to JSON format.
- Encrypts the data using AES-256 in CBC mode.
- Encodes the encrypted data in Base64 for safe transmission.
- Sends the encrypted data back to Destination System.

Example:

Encrypted response from Source System

```
{
 "encrypted_data": "7gFgL8D3... (Base64-encoded AES data)"
}
```

8.4 Destination System decrypts Data

The Source System sends the encrypted data to the Destination System. The Destination System will decrypt the data and inserts the data into the database.

```python
# Function to decrypt AES-256 data
def decrypt_data(encrypted_data):
    encrypted_bytes = base64.b64decode(encrypted_data)
    cipher = AES.new(AES_KEY, AES.MODE_CBC, AES_IV)
    decrypted_bytes = unpad(cipher.decrypt(encrypted_bytes), AES.block_size)
    return json.loads(decrypted_bytes.decode("utf-8"))
```

Fig. 4 Destination System Data Decryption Python Code

Explanation:

Destination System:

- Extracts the encrypted data from the API response
- Decodes the Base64 string to get the raw encrypted bytes
- Extracts the Initialization Vector from the first 16 bytes
- Decrypt the data using the same AES-256 Secret Key
- Removes padding to restore the original JSON Data
- The JSON Data is parsed and inserted into the database

Example:

Here is an example showing decrypted data after processing.

```
[
  {"id": 1, "name": "Steve", "email": "steve@decrypt.com"},
  {"id": 2, "name": "Nathan", "email": "nathan@decrypt.com"}
]
```

9. CONCLUSION

Therefore, leveraging API driven data integration ensures a sustainable and secure pathway to drive and enhance analytics capabilities, increase institutional efficiency and supporting evidence-based policy and decision making across PNG's public and private sectors. The simulation using Python Programming as presented above demonstrated how secure, automated data integration can be achieved through RESTful APIs. Even the secure structure that is used in the API assures that API has the reliable and safe approach to transfer data between two systems or integrate data. To conclude, the usage of API in data integration seem to be an ideal approach to integrate data for enhanced data analytics in PNG.

REFERENCES

A. M. Abdullah, "Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data," Cryptography and Network Security, 2017.

B. Costa, P. F. Pires, F. C. Delicato, P. Merson, "Evaluating a Representational State Transfer (REST) Architecture: What is the Impact of REST in My Architecture?" IEEE/IFIP Confer- ence on Software Architecture, pp. 105–114, 2014.

D. Suciu, "Semi-structured data and XML," Book: Information Organization and Databases: Foundations of Data Organizations, pp. 9 – 30, 2001.

F. Bulthoff, M. Maleshkova, M, "RESTful or RESTless – Current State of Today's Top Web APIs in The Semantic Web," ESWC 2014 Statelite Events, pp. 64 – 74, 2014.

G. M. P. Lazetic, "Native XML Databases vs. Relational Databases in Dealing with XML Documents," Kragujevac J. Math. vol. 30, pp. 181 – 199, 2007.

K. Rabah, "Theory and Implementation of Data encryption Standard: A Review," Information Technology Journal, 4(4), pp. 307 - 325, 2005, doi: 10.3923/itj.2005.307.325

M. Maleshkova, C. Pedrinaci, J. Domingue, “Investigating Web APIs on the World Wide Web,” Eighth IEEE Euro- pean Conference on Web Services, pp. 107–114, 2010.

M. N. Alenezi, H. N. Alabdulrazzaq, Q. Mohammad, “Symmetric Encryption Algorithms: Review and Evaluation study,” International Journal of Communication Networks and Information Security (IJCNIS), 12(2), pp. 256 – 272, 2020.

N, Aleisa, “A Comparison of the 3DES and AES Encryption Standards,” International Journal of Security and Its Applications, 9(7), pp. 241-246, 2015, doi: 10.14257/ijsia.2015.9.7.21

O. P. Verma, R. Agarwal, D. Dafouti, S. Tyagi, “Peformance Analysis Of Data Encryption Algorithms,” IEEE Delhi Technological University India, 2011.

R. Battle, E. Benson, “Bridging the semantic Web and Web 2.0 with Representational State Transfer (REST),” Web Semant. Sci. Serv. Agents World Wide Web, 6(1), pp. 61 – 69, 2008.

Sarker, I. H. (2021). Data Science and Analytics: An Overview from Data-Driven Smart Computing, Decision-Making and Applications Perspective. SN Computer Science, 2(5), 377. https://doi.org/10.1007/s42979-021-00765-8

Stylos, J. (2009). Making APIs more usable with improved API design, documentation and tools (Doctoral dissertation). Carnegie Mellon University. Retrieved from http:// www.cs.cmu.edu/NatProg/papers/

Zibran, M. (2008). What makes APIs difficult to use? International Journal of Computer Science and Network Security, 8(4), 255–261.