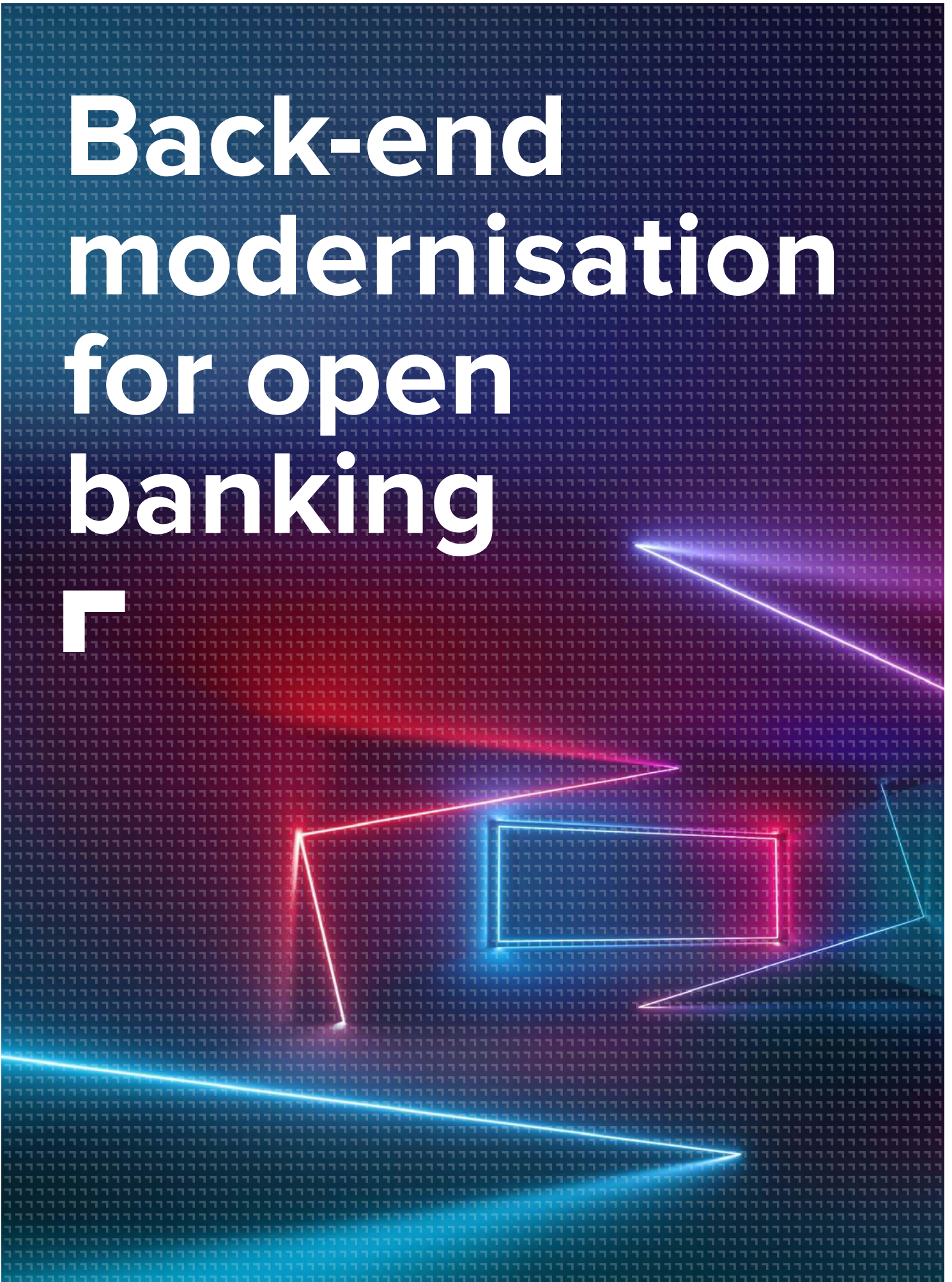


Back-end modernisation for open banking



Introduction



For decades, mainframes have been the cornerstones of IT infrastructure in many businesses, with banks at the forefront. However, the demand for increased flexibility to cope with ever-changing business processes and the need to reduce recurrent costs are forcing companies to move towards open platforms.

When mainframe computers first appeared on the market, they quickly gained popularity, especially in the 1960s. The factors behind this rise in popularity were their security and considerable processing capacity, which allowed thousands of transactions (from cash withdrawal orders to bank transfers) to be sent simultaneously.

However, technological advances are giving way to other platforms that offer major advantages over mainframe systems. One of the biggest problems associated with these traditional environments is the dependence on one provider, which involves high maintenance costs.

Today, many banks still run legacy applications that were written in the 1980s and 90s, some relating to such critical areas as credit card payment processing.

At the same time, there is also an increasing shortage of mainframe specialists, systems technicians and experts in COBOL (the language used by mainframes), since no generational renewal has taken place as older experts have retired. The training possibilities are gradually reducing, and organisations are looking towards more modern alternatives.

This shortage of mainframe specialists also contributes to rising costs, since mainframes require highly skilled staff. And it is not only a matter of costs, application maintenance and security are also being compromised at a time when GFT estimates MIPS consumption is growing at an annual rate exceeding 10%.

The challenge of the banking industry



Banks are not exempt from this problem, especially given the highly competitive market in which they operate. The factors that affect banks, such as low interest rates, the entry of new players that are taking over tasks traditionally performed by banks, and customer attrition, are forcing entities to innovate and move towards service-based models.

Just as banks have upgraded their entire range of front-end services to personalise and improve the customer experience, they are now starting to move away from monolithic back-end systems. After all, it becomes increasingly difficult to align these legacy systems with the new business context where continuous changes require real-time responses.

The modernisation of such systems is a strategic move that is part and parcel of the digital transformation banks must undergo to adapt to the market, become more agile and accelerate innovation. Abandoning a mainframe environment involves making a shift towards standardisation through an approach aimed at reducing complexity and thereby increasing cost efficiency.

According to a 2017 GFT survey¹ of 285 banking professionals on the level of digitalisation within the financial sector, 62% of all banks surveyed said they had started to adopt the “bank as a platform” (BaaP) vision, a finding that gives a very clear picture of the degree of industry awareness of the need to transition to open banking.

This is the only way to make the paradigm shift in banking from a financial product-centric approach to customer-centric solutions. In fact, customer satisfaction, rather than product creation, is where banks’ true economic value lies. Customers are increasingly demanding two-way, multi-channel, multi-device and, above all, more personalised contact.

Do monolithic mainframe environments have the capacity to cope with this new scenario and to allow banks to offer third-party products or even use the platforms of other banks via APIs? The answer to that question is no. And, for that very reason, there is a need for infrastructure and application changes.

¹ Banking expert survey 2017, GFT, October 2017. Survey results report available at > www.gft.com/BankingSurvey2017



Modernisation options

7

The immediate benefits of transitioning from a monolithic to an open environment range from greater simplicity and flexibility when integrating different technologies, to greater cost efficiency in terms of licensing, maintenance and support.

The transition from mainframe infrastructure to an open environment can be handled in different ways. Where possible, the choice should be based on advice from experts who have already overseen such projects, since the process is complex and requires a team with specialist knowledge of both the source and destination platforms.

Three options stand out from the rest. Their advantages and disadvantages must be carefully considered, since successful implementation depends on the bank having correctly aligned its objectives with its IT development plans:

7 **Rewriting**

This option consists of rewriting the applications that run on the mainframe in a new programming language that can run in open environments.

This is the best modernisation option, but it is also the most expensive and involves the most risks. It suits ad-hoc updates of specific features. However, when attempts have been made to undertake this process in full IT modernisation projects, it has failed to produce the expected results producing major deadline and cost deviations.

The COBOL language used to develop mainframe applications is underpinned by historical knowledge that unfortunately is rarely well documented. For this reason, rewriting such applications in a new language is a highly complex process in which it is relatively easy to omit features that are hidden between the different layers of code. An application rewriting project can thus turn into a nightmare for IT departments by significantly delaying scheduled production dates and affecting the business itself. Furthermore, although mainframe applications are compiled as individual units, they are made up of hundreds of programs that work together.

7 **Emulation**

This type of solution is suitable for companies whose sole medium-term objective is to reduce infrastructure costs. There are two emulation-based options.

The first is re-platforming, in which both the base software and the application code are retained and only the infrastructure on which the applications run is changed. This option offers clear and immediate cost savings, but its main disadvantages include incompatibility with some third-party software and the lack of options for future development.

The second option is re-hosting, which involves making changes in both hardware infrastructure and software. This option usually means retaining COBOL as the programming language in the target environment, although it may require changes to the database management system. It is usually accompanied by the use of productivity tools not generally found in the mainframe. The current trend is to move towards a private cloud environment.

However, these modernisation options do not allow companies to make changes in their business model. Companies remain confined in a proprietary environment, as with a mainframe system, in which they depend on the roadmap and commercial decisions of the provider of the emulation option they have chosen. Sooner or later these solutions are exposed as half measures.

7 **Transformation**

This approach consists of upgrading the applications within a context of automated code conversion and data migration processes, without emulating the mainframe's base software and with the possibility of carrying out native development from the moment it is production ready. In essence, mainframe environments are abandoned in favour of standard hardware with more open operating systems such as Linux and Unix and a programming language such as Java, which has replaced COBOL as the standard business language.

With the transformation option, one of the major benefits of an open environment is that competition between providers is so high that customers are much less dependent on them. Such options involve cloud computing models and a complete move away from mainframe systems in which companies are at the mercy of the sole provider's decisions concerning updates and licensing.

The transformation option therefore allows companies to change their entire environment from one provider to another in the event of a change in licensing policy, for example. With this formula, no software affects the other applications, which means that the provider migration can be done in a controlled manner at any time with easier integration of different technologies.

Shaping
the future
of digital
business

Retail
banking

Back-end
modernisation
for open
banking

Point
of view



GFT's stance

7

GFT's experience in the banking sector shows that the transformation option generally provides the greatest benefits and lowest risks for companies.

There are certainly critics of this option who base their arguments on the challenges of understanding and maintaining procedural programming in Java. However, the fact that the amount of mainframe code updated in the last three years rarely exceeds 20% of the total amount of system code is a good indication that transformation is the best modernisation option.

Naturally, projects of this magnitude should be approached on a case-by-case basis, because each company's needs are unique. Having decided on the transformation process, the next step is to select an application to perform a proof of concept (PoC). It is important to consider applications that are stable, medium-sized and unaffected by major projects.

The aim of this PoC is to define the architecture to be adopted by creating a detailed mapping document that maps out the legacy technologies that will be migrated to the new platform; to design and implement platform integration and coexistence (online and batch interfaces, data synchronisation in near real time, etc.); to define the test strategy and the use of automated testing tools; and, the final, important step, to measure and adjust the performance of the transformed applications.

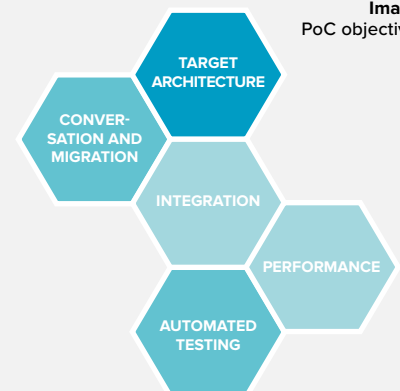


Image:
PoC objectives

The one-size-fits-all approach should be dismissed to avoid the error of thinking that the same transformation strategy can work for every bank.

Furthermore, GFT experts conclude that a big bang transformation approach is usually preferable to an incremental approach. The argument that it is more sensible to transform the bank's core business applications in stages to make the process more manageable and controllable does not always apply.

Often, particularly in small and medium-sized mainframe environments, the costs and, more importantly, the risks associated with having two different platforms at the same time and the integration required to align them with the business are much greater than those associated with a big bang approach, which involves drastically changing the entire platform from one day to the next.

Conclusion

7

Open banking is no longer just another option

Mainframe and application transformation is the first step, and it's an important one because it offers a range of benefits. However, it doesn't end there, and a rethink of the core banking strategy is essential. Many steps are therefore required in the modernisation process, starting with the transformation itself, which should then be followed by an evolutionary process.

In fact, new needs have already begun to emerge for which the ideal scenario is an open environment, given that the integration process is technologically much more straightforward. This is the case with third-party core banking systems and the ability to interact with APIs.

However, before embarking on this evolutionary process, it is advisable to assess whether the previous stage, in which GFT advocates transformation, has been

implemented successfully. To achieve this, it is necessary to analyse whether the initial objectives have been achieved in terms of renewal, cost, more efficient production processes and standardisation of the IT infrastructure among others.

In this context, adopting an open distributed environment means speeding up the pace of innovation, which will undoubtedly put the company in a better position to tackle future challenges. One such challenge is time-to-market, which has accelerated so much that apps must now be able to provide the required information in real time.

Banks have an additional challenge associated with the real-time imperative: being proactive. To this day, banks and savings banks anchored in outdated models still regard text messages as an innovative response. Proactivity towards

customers, fuelled by the evolution of front-end services, is a fundamental feature that can be further strengthened when it is underpinned by an open technology platform. A customer's ability to submit a request or carry out a banking transaction through a mobile banking app depends on the performance of the back-end system.

Transitioning to open banking is no longer a choice; it is a question of survival. This is even more true in the short term, since banking products will no longer be marketed exclusively by traditional banks, and the need for speed and flexibility will make all the difference.

MAKING THE TRANSITION TO OPEN BANKING

In terms of stability, there are no significant differences between mainframe and non-mainframe environments.



Mainframe environments and their current operations are not true cloud environments, so they do not offer the benefits of a cloud platform.

Mainframe costs are significantly higher than those of non-mainframe environments.



The use of productivity tools enhances the agility and innovation of non-mainframe platforms. In addition, strong competition between providers reduces the dependency experienced by customers in mainframe environments.

Mainframe experts tend to be older and are expected to retire from companies in the next 10 years. It will be increasingly difficult to attract and retain mainframe talent.



Abandoning mainframe platforms offers additional potential in terms of increasing standardisation and reducing the complexity of app development and maintenance.

Success story



80% savings

GFT has extensive experience in the banking industry helping companies successfully transition to open banking systems. One financial institution whose mission-critical applications were running in a mainframe environment, with COBOL as the programming language, CICS as a teleprocessing monitor and a DB2 database, undertook a two-year transformation project. This involved not only modifying its hardware infrastructure and database, but also migrating to the Linux operating system and replacing COBOL with Java.

The benefits of this major project were revealed following the preliminary assessment carried out with the company and were mainly associated with infrastructure costs:

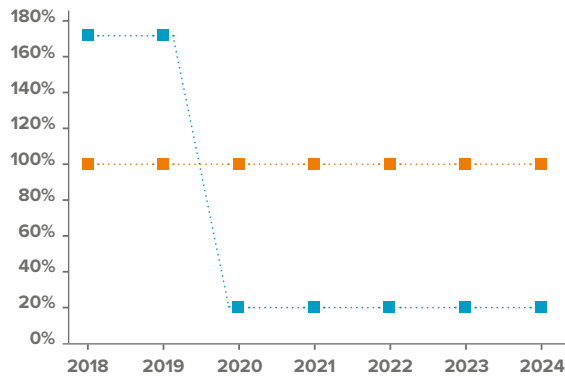


Chart 01:
An 80% annual reduction in recurrent costs

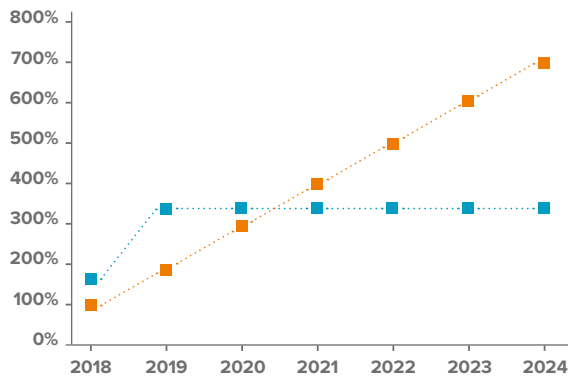


Chart 02:
Return on investment in 21 months

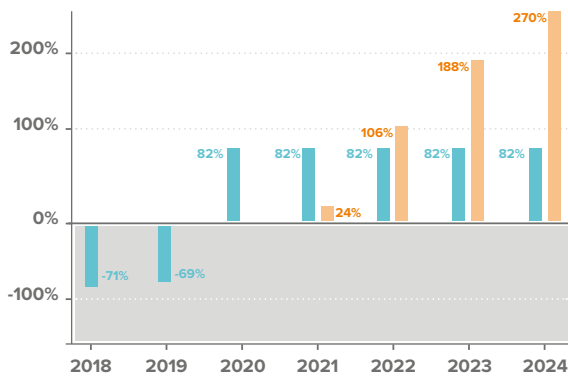


Chart 03:
270% estimated cost savings in five years with respect to present recurrent costs

Following the transformation process, the company will also be able to respond more quickly and proactively to its customers' demands by anticipating many of their requirements. In addition, the company will experience improvements in application development and maintenance costs (usually around 20%), although this aspect varies depending on the characteristics of each system.



About GFT



World-class financial services specialist in technology and innovation

As an experienced technology partner, GFT Technologies SE (GFT) is committed to driving the digital transformation of the financial services industry. Drawing on our extensive knowledge of the sector, we advise the world's leading financial institutions and develop bespoke IT solutions – from banking applications and trading systems to the implementation and support of complete platforms, and the modernisation of core banking systems.

Our global innovation team also develops new business models, focusing on topics such as blockchain, cloud engineering, artificial intelligence and the internet of things across all sectors. Together with our clients, we aim to shape the financial world of the future.

Founded in 1987, the company is now represented with a global team of around 5,000 employees in Europe and North and South America.

-  blog.gft.com
-  twitter.com/gft_en
-  [linkedin.com/company/gft-group](https://www.linkedin.com/company/gft-group)
-  [facebook.com/GFTGroup](https://www.facebook.com/GFTGroup)
-  www.gft.com

Our specialists



Enric Pérez



Technical program manager and senior IT architect



Enric Pérez is technical program manager and senior IT architect at GFT, with over 25 years of IT experience covering IT development, architecture and infrastructure. His work focuses on mainframe consumption optimisation and downsizing, mainframe IT architecture, core business evolution and IT operations automation.

Prior to joining GFT in 2010, Enric worked for almost two decades at Deutsche Bank Spain where he held various positions, including his last appointment as director of architecture and infrastructure.

Enric studied Mathematics at the University of Barcelona.

Fabrizio Di Peppo



Delivery executive manager



Fabrizio Di Peppo is delivery executive manager at GFT, overseeing large technology deployments for an international banking group. He has extensive experience in the deployment of IT systems in the financial and industrial sectors, with a particular focus on application modernisation and migration from legacy systems to open platforms and over 20 years of experience in rehosting projects.

Since joining GFT in 2009, Fabrizio has also been involved in various EU international IT projects. Prior to his employment at GFT, Fabrizio served as project manager for Datel SpA for over two decades.

Fabrizio holds a degree in Computer Programming and Accounting from ITC Sommeiller in Turin.

This report is supplied in good faith, based on information made available to GFT at the date of submission. It contains confidential information that must not be disclosed to third parties. Please note that GFT does not warrant the correctness or completion of the information contained. The client is solely responsible for the usage of the information and any decisions based upon it.