Flexibility Meets Scale: Cyberbank Next-Gen Core

Reaches 10,000 TPS on the Cloud

Microsoft and Galileo:
Delivering Next-Gen Digital Banking
A technology capabilities overview - including performance testing - for scaling one of the top five largest banks in the world, running on the Microsoft Azure Cloud and the Galileo Cyberbank Platform.*

*This scalability report was initially conducted in October 2021 by Technisys in conjunction with Microsoft. Subsequently, Technisys was acquired by SoFi in 2022 and soon thereafter was rebranded under its affiliate's name, Galileo Financial Technologies.

CONTENTS

AUTHORS

Ricardo Zuasti (Galileo)
Lalo Steinmann (Microsoft)

Michel Jacobs (Galileo)
Diana Ermini (Galileo)

Ezequiel Kobylnik (Galileo)
Esteban Vaschetto (Galileo)

Federico Pinella (Galileo)
Ariel Hernandez (Galileo)
José Maria Balmaceda (Galileo)

Leandro Gallo (Microsoft)
Carolina Estrada (Microsoft)
Iván Contreras (Microsoft)

Chapter 1. Business Value

CHAPTER 1. BUSINESS VALUE
Financial Services Today

Due to current market conditions, many consumers were forced to adopt digital practices to shop, to bank, to learn and to be entertained. While the world was increasingly going digital, last year accelerated the trend. Consumer behaviors, habits and preferences have changed, and continue to evolve. To grow and thrive in the digital-first age requires that banks are agile enough to create and deliver tailored financial products to customers, dynamically, at point of need. However, many existing banking systems are not designed to support a complex layer of customization and then deliver tailored offerings across multiple channels at exceptional speed.

Banks and fintechs are effectively and rapidly meeting the needs of multiple generations – particularly millennials who use their mobile devices as the predominant mode of interaction. The rising popularity in digital banking and fintech services have opened opportunities for financial institutions across the industry. To meet modern-day consumer demands, leading financial institutions are looking to implement next-gen digital banking platforms to bring innovative financial products and services to life and deliver meaningful value – creating a dynamic playing field throughout the Americas.

Cyberbank is a next-gen, cloud-native digital banking platform from Galileo (formerly Technisys*) empowers large-scale banks to create and deliver tailored financial products and services that meet the needs of customers at any point of interaction, at point of need – all at the speed of commerce. Cyberbank's unique architecture offers powerful structural flexibility – meaning that product definitions can be arranged and rearranged dynamically to meet customer needs when it matters most – whether to one or to hundreds of millions of customers. Empowering banks to meet evolving customer behaviors in new ways, while embracing agility and seamless connection.

The Power to Differentiate with Microsoft and Galileo

In this report, we highlight how Microsoft and Galileo are well positioned to support one of the top five largest banks in the world; to be referenced as "The Bank" from this point forward. In this overview, we will cover the key elements needed to support The Bank in its mission to elevate the customer experience in today's fast-paced, digitalfirst environment; highlight the key elements to look for in a next-gen digital banking platform; articulate the unique design of a proven deployment architecture; and showcase performance benchmarking results

*This scalability report was initially conducted in October 2021 by Technisys in conjunction with Microsoft. Subsequently, Technisys was acquired by SoFi in 2022 and soon thereafter was rebranded under its affiliate's name, Galileo Financial Technologies.

The Four Essentials of a Cloud-native, Next-Gen Digital Banking Platform

1. Access the power of structural flexibility.
Change any business rule, process, product definition, or customer experience -- all while minimizing friction and risk with every update. Fully automate deployment and environment management processes and allow for new versions to be released in minutes, including automated tests, and security and code tests.

2. Redefine customer-centricity.
Deliver custom offerings that become integral to customer lifestyles in new and profound ways, whether to large target segments or down to a segment of one. Add a delightful experience layer to your existing systems and support an entire product lifecycle to: dynamically manufacture new products, distribute products via multiple interaction points (e.g., banking portal, partner portal, eMarketplace), deliver tailored products with high accuracy, and package financial products to support the best, online self-service experience possible.

3. Use enterprise-grade performance and scalability.
No matter whether your bank supports hundreds - or millions - of customers, a cloud-based, next-gen digital banking platform must be designed to flex with your business. Look for the highest levels of security, reliability, performance and resilience. A microservices-oriented architecture will drive ultimate efficiency and scalability - and enable any system change or product upgrade in one module or service to not affect the entire platform, reducing costs and saving time.

4. Access an end-to-end digital backbone.
Use leading-edge technology, like Docker, Kubernetes, Java, SpringBoot, Azure DevOps, Helm, Terraform, and more to manage the entire creation process of a new service or product. Due to the powerful, end-to-end digital backbone of the Cyberbank platform, leading industry analyst, Forrester, ranks it highest in the criteria of architecture and digital banking foundation in The Forrester Wave™: Digital Banking Processing Platforms (Retail Banking), Q3 2020 report.

Novel Products + Real-time Delivery = Seamless Experience Azure & Cyberbank. Together.

Together, Microsoft and Galileo can support The Bank's success in the digital journey. Through our integrated solution, joint support, unified management and interoperable application development capabilities, we can build, deploy, and manage a security-focused hybrid environment - designed to meet today's business needs and adapt to future changes.

While Galileo provides an end-to-end digital backbone, Microsoft Cloud for Financial Services brings additional capabilities with multilayered security and comprehensive compliance coverage to support differentiated customer experiences, improve employee collaboration and productivity, manage risk, and modernize core systems. Microsoft Cloud for Financial Services, with Galileo, supports a wide range of key, industry-specific certifications to further accelerate deployment, performance and impact of all tailored solutions brought to market.

This comprehensive report will not only explore the power of a robust deployment architecture, but also will reveal benchmarking results that support The Bank's 150M+ customers - especially at peak performance levels.

Chapter 2. Deployment Architecture

Galileo's Digital Banking Architecture

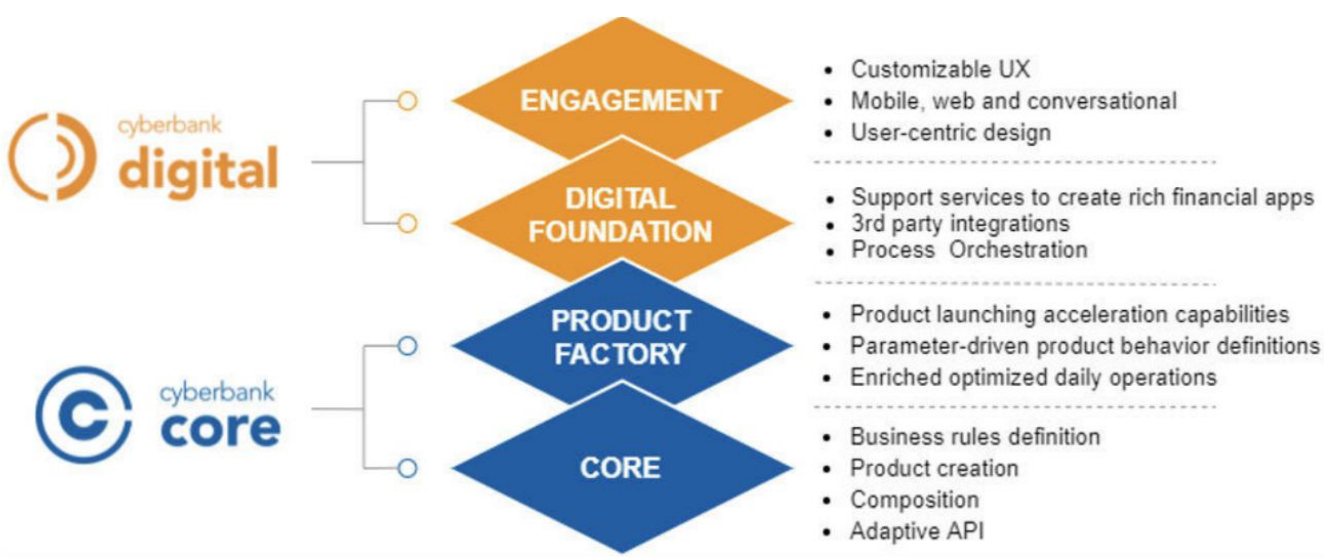
Cyberbank is based on a cloud-native metadata architecture and designed to be cloud-agnostic. As a result, it can be deployed in hybrid solutions such as cloud-on-premise, various cloud-to-cloud providers, or via full on-premise providers.

The Galileo solution is divided into two main layers: 1) the product factory (upper layer), and, 2) the core (lower layer). Each of these layers is built to fulfill a specific purpose. Let's start from a bottom-up perspective.

The core layer is where the high-performance execution machine and the stack of pre-built atomic services reside. This is the first entry point to full flexibility given our API-centric approach. This is possible through a set of business-oriented tools built by banking experts for business experts provided by Galileo. Also, supported by Cyberbank Core is the Product Factory - a no-code, parameter-driven layer that is easy-to-use and empowers the platform to build innovative financial products in a matter of minutes. This layer provides full transactional support and ready-to-use features to operate a retail and/or commercial bank.

There are three key concepts associated with Cyberbank Core that are important to call out: services, workflows, and layouts. A service is a set of logical steps (i.e., instructions) that allow for modeling business rules within the platform. Thus providing the ability to define the behavior of an operation in a declarative and user-friendly way. A workflow can be defined as a group of states, transitions, screens, and API calls defined in a BPM context that gives the flexibility to generate and customize functionalities. A layout is a point of interaction with Cyberbank users.

Each layout is made of logical structures and data binding definitions, that are created using a no-code tool provided by Galileo. Through this point of interaction, a user can create settings, perform banking operations, and add new data into the system of record.



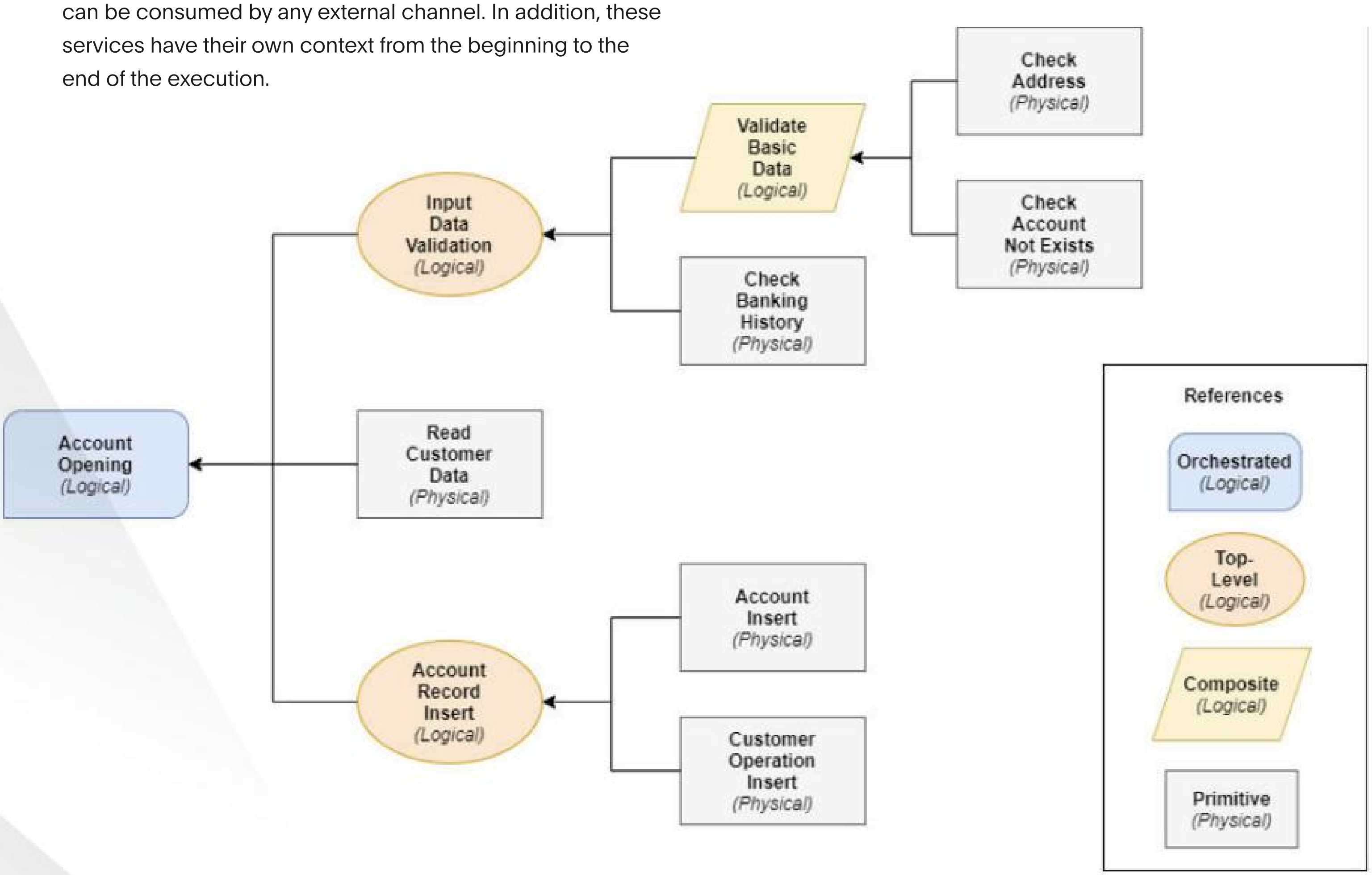
Let's take a deeper dive into the basics and cover the service types. From a high-level perspective, Cyberbank Core has two main types of services: Logical and Physical Services.

A Physical Service is a java class file that is one instruction in a Logical Service. The most common names for physical services are Primitive or Basic services. These services are the only ones where coding is required. Since Cyberbank Core pursues a no-code approach, the available primitives are as generic as possible so they can be reused across multiple logical services. Cyberbank provides an out-of-the-box library with more than 2.5K pre-built primitive services.

A Logical Service is made of metadata that contains parameters, instruction calls, and other commands that Cyberbank will interpret. There are three different subtypes of logical services: Composite, Top-Level, and Orchestrated services.

- A Composite is a conjunction of Physical Services (primitive) and Logical Services (other composites). This kind of service is only available for consumption within the platform and cannot be exposed directly through the Core API.
- The structure of a Top-Level Service is similar to a composite, but its behavior is different. Unlike Composite Services, the Top-Level Services are available through the Core API and can be consumed by any external channel. In addition, these services have their own context from the beginning to the end of the execution.
- Lastly, an Orchestrated Service is the highest level of service type. This type is made of any sub-type of service except another Orchestrated Service. Its main purpose is to provide a unique and centralized execution context that ignores the context of multiple Top-Levels in a service definition. This gives the service the ability to reuse the context at any level of nesting.

By making use of these powerful concepts, Galileo has hired banking experts from multiple disciplines to model and create a rich and unique framework library. This library provides a variety of out-of-the-box business cases that any financial institution would need to offer digital banking capabilities cost-effectively, and that can be delivered at the speed of commerce. It is important to highlight that, based on the previous definitions, any service can be tailored - or newly created - to a customer's needs by The Bank's banking experts without the need for coding skills. An example of how services can be defined to model a business case is shown in the diagram below. The diagram illustrates a simplified definition of an account opening business case and clarifies the responsibility of each type of service.



Cyberbank Core on Microsoft Azure

Cyberbank Core is designed to operate within financial institutions of all sizes. From small fintechs with only a thousand customers and a few products, to banks with hundreds of millions of customers and a wide variety of product and service offerings. In this technical overview, we highlight the recommended architecture and sizing needed to support The Bank's market growth, and show why Microsoft Azure is the most suitable infrastructure and cloud provider to deploy Cyberbank.

About Microsoft Azure

Azure is a complete cloud platform that can host any existing application and streamlines new application development. It can even enhance on-premise applications. Azure integrates the cloud services that you need to develop, test, deploy, and manage your applications, all while taking advantage of the efficiencies of cloud computing.

By hosting your applications in Azure, you can start small and easily scale your application as your customer demands grow. The platform offers the reliability that's needed for high-availability applications, including failover between different regions. The Azure portal not only lets you easily manage all of your Azure services, but also manage your services programmatically by using service-specific APIs and templates.

Azure services are helping financial services institutions create the next generation of banking and transactional systems. Payments are fundamental to banking and wholesale payment transactions. The rise of fintech firms, the shift in consumer habits and demands, the new data economy, and emerging technologies in financial services are forcing changes. Realtime payments (RTP) are not just about speed. Banks can achieve agility with RTP and enable new and open business models. Modern and global cloud services are enabling organizations to innovate and create solutions that process complex transactions across multiple parties faster than ever and gain insight from the transactions.

Deploying Cyberbank

Cyberbank is designed as a cloud-native solution deployed via container technology and leveraging key elements of the cloud that enable scalability, high availability, disaster recovery and observability. The container manager chosen by Galileo to handle the platform components is Kubernetes. Cyberbank supports out-of-the-box multiple Kubernetes implementations, including Azure Kubernetes Services (AKS).

Cyberbank Core back-end components are developed entirely in Java. In terms of its database support, it is validated and tested on Oracle 11G and 12c (with 19c coming over), Microsoft SQL Server 2017 and 2019, and Microsoft Azure SQL Database (RTM) 12. In order to deploy Cyberbank Core on Microsoft Azure, the following Azure components are needed:

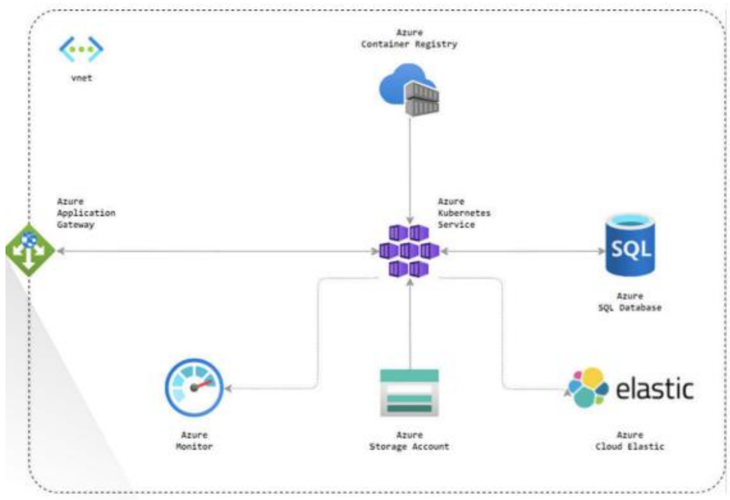
- Azure Storage Account or Azure Data Disk: used to store metadata files and plug-and-play primitive services files.
- Azure Kubernetes Service: used to deploy the solution docker images.
- Azure SQL Server: used to deploy Azure SQL Databases where a customer's business data is stored. (Applicable when traffic peaks require less than 3000 TPS)
- Azure SQL Database: used to store a customer's business data. (Applicable when traffic peaks require less than 3000 TPS)
- Azure Virtual Machine: used to deploy Microsoft SQL Server to store a customer's business data. (Applicable when traffic peaks require more than 3000 TPS) • Azure Virtual Machine or Elastic Cloud Service: used to deploy the platform monitoring stack - by default, ELK stack.

There are two possible deployment models based on the described components: 1) the standard scalable deployment (with HA support) and 2) what is called a high availability - high throughput deployment. For the first model, a PaaS (Platform as a Service) schema is applied and the platform data sources are managed as Azure SQL Services. The second model is certified with Microsoft Azure specifically for customers with incredibly high transaction volumes. The main difference between them is the database deployment format. The second deployment model requires a higher throughput than what is available on a PaaS; it is for this reason, an IaaS (Infrastructure as a Service) schema is applied. Requiring Azure VMs with instances of Microsoft SQL Server 2019 installed on Windows Server 2019.

Standard Scalable Deployment

The standard scalable deployment is the default type of deployment. The used topology is applicable for a variety of customers, from those small ones with 5,000 customers and barely 100 TPS traffic peaks, to large ones with 1,000,000 customers and peaks of 3,000 TPS.

This scenario stands out for its simplicity and low maintenance. The bulk of the solution is managed by cloud services such as Azure SQL Database and Cloud Elastic. Having the data managed by Azure SQL Service, provides easy-to-use migration tools, minimal downtime, scalability based on compute usage, and many other advantages.



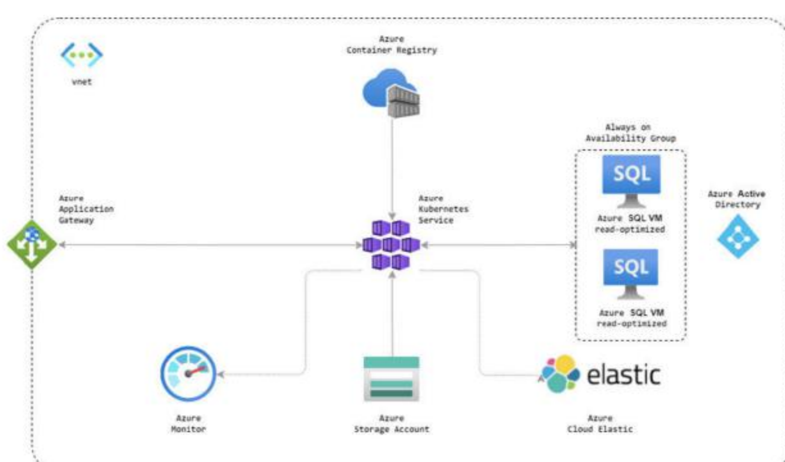
High-Availability / High-Throughput Deployment

High-availability / high-throughput deployment is an exceptional type of deployment. The used topology is applicable for customers with traffic peaks that exceed 3,000 TPS or have huge data volumes bigger than 5,000,000 customers and accounts.

This scenario stands out for its reliability, high performance, and tuning capacity. The most significant difference with a standard deployment consists in how the databases are deployed.

In this scenario, the PaaS databases are replaced with provisioned VM's running Windows Server 2019/2017 with SQL Server 2019/2017. This enables a set of server properties that provides fine-tuning and more governance of the resource itself.

A disadvantage with this deployment is that it requires a higher maintenance ratio and experienced people to handle any issue with the server than a standard deployment.



Chapter 3.
Performance Benchmarking Results

To test the platform, we needed to determine the number of transactions per second the system would receive from the outside world. We based this on a variety of factors - the number of active users, their behavioral patterns, and the products and services offered by The Bank. We also referenced empirical data (i.e., industry experience from Microsoft and Galileo, and data from other known services that deal with interactive, real-time financial transactions).

Empirical data from real life scenarios yields that approximately 0.002% of the active customer base executes concurrent requests at the exact same time in real-time interactive scenarios (where the system total response time remains at an average of 300ms). With The Bank's 150M+ total customer base, peak load is expected to be approximately 9.000 transactions per second; assuming three transactions per customer per second (which is within the upper boundary of a real life scenario).

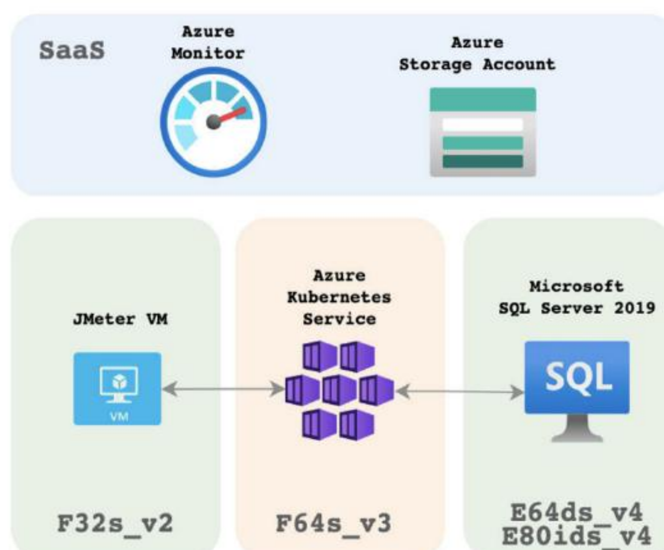
Financial services customers need to run a mixture of workloads, including both online transaction processing (OLTP) and batch jobs. OLTP performance must keep pace with rapid and unpredictable growth, supporting fast response times to provide interactive performance with a large number of concurrent users and ad-hoc transactions.

Batch performance must be sufficient to process significant numbers of records during available batch processing windows.

Because of this, and to test the performance and the ratio of transactions per second, a full benchmarking of the platform was conducted. Different subtypes of performance tests were performed for this benchmark. However, the primary aim was on scalability tests rather than stress tests. For load simulation, Galileo chose Apache JMeter, an open-source software piece developed entirely on Java, designed to perform workload tests and simulate functional behavior that allows performance measuring on a wide range of application types.

Architecture Schema

The conceptual architecture used for this stage with all the components and the performed workflow is illustrated below. As shown in the diagram, for this test we use three types of cloud services: Saas, PaaS, and IaaS.



High-Availability / High-Throughput Deployment

High-availability / High-throughput deployment is an exceptional type of deployment. The used topology is applicable for customers with traffic peaks that exceed 3,000 TPS or have huge data volumes bigger than 5,000,000 customers and accounts.

Test Plan

To test Cyberbank Core in a more realistic scenario, we examined the most commonly used services by customers:

- Retrieving customer consolidated position
- Retrieving an account balance (both savings and checking)
- Performing credits on both savings and checking accounts

These services are executed with a different probability according to how likely customers will use that functionality. In other words, every customer may want to see the consolidated position of all her/his products, but not all would, necessarily, want to make a transfer.

Benchmark Results

The objective of the online workload testing was to determine the performance of Cyberbank Core for an online transaction processing mix of three business-critical transactions. The workload simulated a load of 3,000 concurrent threads for a period of five minutes (including a ramp-up of 20 seconds) executing the chosen workload mix of three transactions (two of them inquiry-based and the last one non-inquiry). Results collected during the benchmark are shown in the following table, which includes aggregated information of more than 3,134,399 business transactions achieved in the complete test execution period.

Parameter	Value
Maximum number of concurrent threads	3000
Test duration	300 seconds
Ramp-Up Time	20 seconds
Total number of business transactions passed	3,134,399
Percent of failed transactions	0%
Average business transactions processed per second (TPS)	10500
Weighted average response time	250 ms

CPU and memory utilization were both carefully monitored using Azure Monitor to determine the impact of the workload on the configured system. CPU and memory consumption would need to be lower than 80% to be considered healthy

System Resource	Average CPU Utilization	Average Memory Utilization
Cyberbank Core Nodes	79%	9%
Database	85%	22%
Load Generator VM	12%	7%

Results obtained in this phase are considered to be baseline; looking to improve the load or hardware.

Provisioned Resources

High-availability / High-throughput deployment is an exceptional type of deployment. The used topology is applicable for customers with traffic peaks that exceed 3,000 TPS or have huge data volumes bigger than 5,000,000 customers and accounts.

Component	Active Directory VM	Cyberbank Core - AKS Nodes	JMeter	Read SQL Server VM	Write SQL Server VM
Size/Family	Standard_D2s v3	Standard_F64s _v2	Standard_F32s _v2	Standard_E80i ds_v4	Standard_E64 ds_v4
vCPU [#]	2	64	32	80	64
RAM [GB]	8	128	64	512	504
Disk/s [#]	1	N/A	1	3	3
Disk/s Size/s [GB]	128	N/A	64	128, 8192 (data), 2048 (log)	128, 8192 (data), 2048 (log)
Disk/s Performance Tier	P10	N/A	P10	P10, P60 (data), P40 (log)	P10, P60 (data), P40 (log)
Disk/s SKU/s	Premium SSD (locally-redundant storage)	N/A	Premium SSD (locally-redundant storage)	Premium SSD (locally-redundant storage)	Premium SSD (locally-redundant storage)
Accelerated Networking	YES	NO	YES	YES	YES

Scaling the Platform

Part of an application's performance is based on how it scales to provide more throughput while maintaining the same behavior in terms of reliability and responsiveness. Another point to consider is how the solution will respond to an increase in workload when many more consumers submit requests. Lastly, application performance can be impacted by an increase in data volume - this is directly related to how the model and product are built - and will determine the scalability of the application.

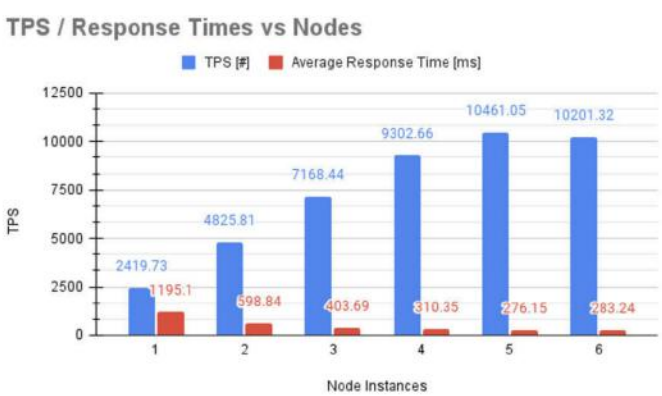
In order to gather information about these capabilities on Cyberbank, each use case was prepared and performed with Cyberbank on Microsoft Azure.

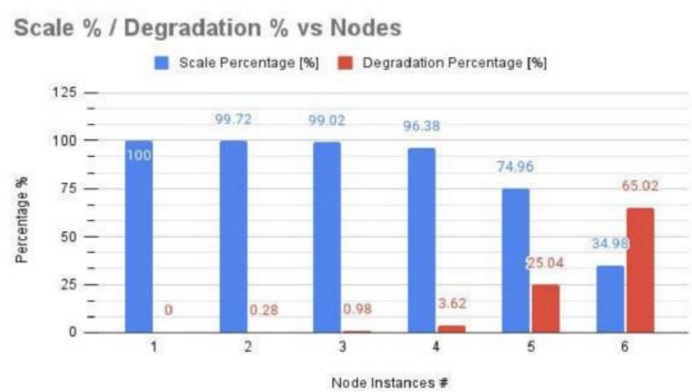
Application Scaling

The first analysis is focused on the scale out test. In this analysis, the platform was tested for a load of 3,000 concurrent threads with a database setup of Standard_E80ids_v4 VM size, which has 80vCPU and 512GiB of RAM available. The database disk was configured with a P60 disk tier for data, a P40 for logs and a P10 for the operating system.

In this effort, the nodes were added incrementally to test how much the application could scale while adding more instances

Con-current Threads [#]	Cyber-bank Nodes [#]	Total Samples Processed [#]	Average Response Time [ms]	Expected TPS [#]	Obtained TPS [#]	Reached Percent age [%]	Degra-dation Percent age [%]	DB C PU [%]
3,000	1	729,712	1195.1	2419.73	2419.73	100	0	17.03
3,000	2	1,455,692	598.84	4839.46	4825.81	99.72	0.28	34.67
3,000	3	2,157,234	403.69	7238.71	7168.44	99.02	0.98	54.65
3,000	4	2,804,902	310.35	9651.62	9302.66	96.38	3.62	74.9
3,000	5	3,149,645	276.15	13953.99	10461.05	74.96	25.04	85.20
3,000	6	3,070,710	283.24	15691.57	10201.32	34.98	65.02	88.21





As illustrated in the table and graphs above, the application scales almost linearly. The degradation percentage is minimal and trivial. An important point to make here is that when the application nodes are more than 5, as shown in the data, the database reaches critical vCPU peaks affecting performance and degrading the processing capacity of the application. Consequently, it is safe to assume that having more resources in the database will enable the application to expand scalability and support more TPSs.

Database Scaling

At this point the scale out is still present, but a different approach was taken. In this case, the database was clustered providing one read-replica and a primary instance for writing processes. In order to do this, the team divided the load into two different types of application nodes: one for read-transactions, and another for write-transactions. In this particular situation, the platform was also tested for a load of 3,000 concurrent threads with a database setup of Standard_E80ids_v4 VM size with 80vCPU and 512GiB of RAM for read-transactions and a database setup of Standard_E64ds_v4 VM size with 64vCPU and 504GiB of RAM for write-transactions. Both database disks were configured with a P60 disk tier for data, a P40 for logs and a P10 for the operating system.

This experience included the incremental increase of nodes to test the scalability of the application, while adding more instances with reads and writes split on different nodes.

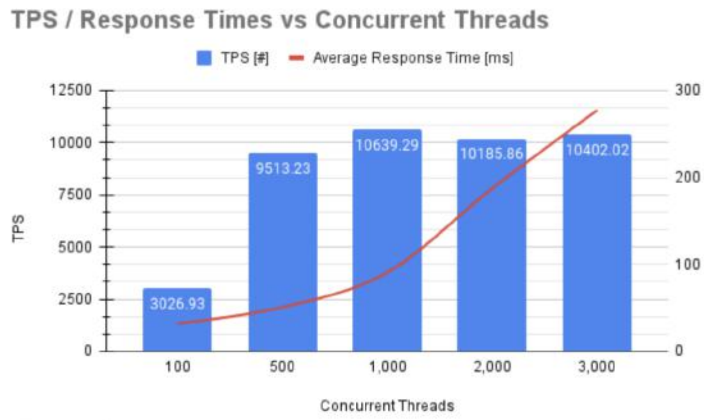
Concurrent Threads [#]	Cyberbank Read Instances [#]	Cyberbank Write Instances [#]	Total Samples Processed [#]	Average Response Time [ms]	Expected TPS [#]	Scale Percentage [%]	Degradation Percentage [%]	DB Read CPU [%]	DB Write CPU [%]
3,000	1	1	901,420	967	2,985.2	100	0	19.5	6
3,000	2	2	1,807,206	482	5,983.7	100	0	39	10
3,000	3	3	2,678,259	325	8,869.6	98.8	1.8	62.1	13
3,000	4	4	3,631,868	233	12,381.4	93	7.17	84	17
3,000	5	5	3,352,859	248	11,108.7	74.3	25.69	79	21
3,000	6	6	3,893,967	223	13,234.5	73	28.74	85	27

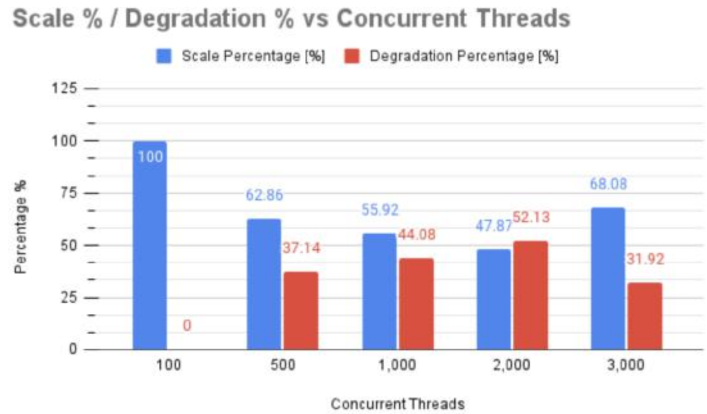
Workload Test

In this part, the platform was tested for different types of workloads with the goal of understanding how much load a group of nodes can handle (five in this test) and determine the "breakdown" point of the solution. In this case, a database setup of Standard_E80ids_v4 VM size, which has 80vCPU and 512GiB of RAM available, was selected. The database disk was configured with a P60 disk tier for data, a P40 for logs and a P10 for the operating system.

This experience included increasing the workload in various stages to test how much load could be processed without breaking and/or degrading.

Concurrent Threads [#]	Cyberbank Nodes [#]	Total Samples Processed [#]	Average Response Time [ms]	Expected TPS [#]	Obtained TPS [#]	Reached Percentage [%]	Degradation Percentage [%]	DB C PU [%]
100	5	907,822	31.84	3,026.93	3,026.93	100	0	22
500	5	2,854,482	50.71	15,134.65	9,513.23	62.86	37.14	77
1,000	5	3,195,361	90.70	19,026.46	10,639.29	55.92	44.08	85
2,000	5	3,098,253	187.28	21,278.58	10,185.86	47.87	52.13	85
3,000	5	3,134,399	277.12	15,278.79	10,402.02	68.08	31.92	87





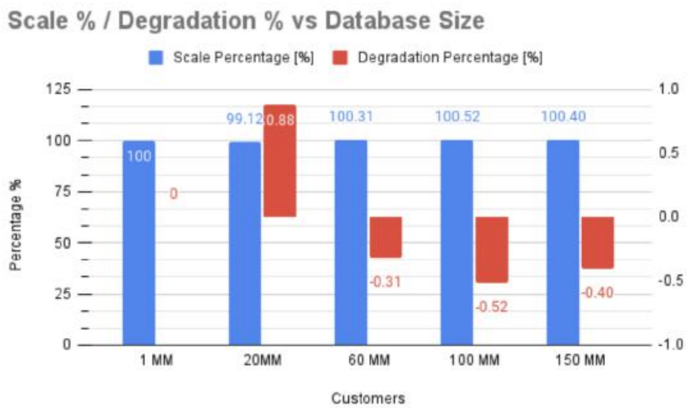
As illustrated in the table and graphs above, the application behaves as expected, increasing the number of TPSs on the first load increases and degrades performance when the load is more than 500 concurrent threads per instance node.

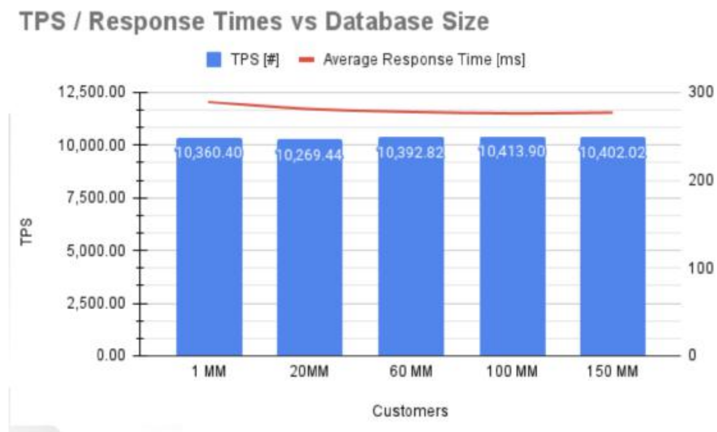
Application Behavior on Huge Data Volumes

In this performance assessment, the platform was tested for various workloads with the goal of understanding how much of a load a group of nodes can handle (five in this test) and determining the "breakdown" point of the solution. For this case, a database setup of Standard_E80ids_v4 VM size, which has 80vCPU and 512GiB of RAM available, was selected. The database disk was configured with a P60 disk tier for data, a P40 for logs and a P10 for the operating system.

This experience included increasing the workload in different stages to test how much load could be processed without breaking and/or degrading performance

Accounts [#]	Customers [#]	Concurrent Threads [#]	Cyberbank Instances [#]	Total Samples Processed [#]	Average Response Time [ms]	TPS [#]	Degradation Percentage [%]	DB CPU [%]
2,000,000	1,000,000	3,000	5	3,310,036	289	10,360.4	0	85
40,000,000	20,000,000	3,000	5	3,094,706	281.01	10,269.44	0.88	85
120,000,000	60,000,000	3,000	5	3,128,540	277.85	10,392.82	-0.31	87
200,000,000	100,000,000	3,000	5	3,005,803	276.24	10,413.9	3.71	89
300,000,000	150,000,000	3,000	5	3,134,399	277.12	10,402.02	-0.40	92





As illustrated in the table and graphs above, the application behaves as expected, increasing the number of TPSs on the first load increases and degrades performance when the load is more than 500 concurrent threads per instance node.

Conclusion

Faced with incredible challenges and promising new opportunities, banks need flexible and agile solutions that solve their unique problems, while providing a high-performance infrastructure that scales as business needs dictate. As demonstrated by the multiple tests highlighted in this paper, Galileo Cyberbank Core, running on Microsoft Azure, is a highly scalable core banking solution that not only can manage a high volume of data and many concurrent threads at the same time, but also provide high response times. While Cyberbank is designed to handle the workload, Microsoft Azure is engineered to reduce cost and risk in the enterprise, providing top-to-bottom integration of the solution stack and ease of deployment to deliver maximum potential with an easy-to-use portal to manage it. With Azure and Galileo, fintechs and banks of all sizes are able to flex and scale in the digital age.

Lessons Learned

Carrying out tests of this magnitude is certainly not a simple task. There are a multitude of variables to keep in mind (e.g., defining resources, timeframes, data population, infrastructure metrics, application behavior, data gathering, and more) in order to capture valuable insights.

One of the first, “What should I do here?” moments for these tests came up very early on; especially as it related to resource allocation. “How much vCPU?” “What about RAM?” “How much will it cost?” “Do you need any storage service?” “How many application nodes?” “How much time is needed?” These are some of the questions that most would ask in order to determine the cost of a solution from both a budget and time perspective. While other technical concerns were addressed - “Is this really feasible?” “How much test data will I need?” “How much time do I have to do this?” “Is the customer’s traffic profile information available?” - and so forth.

Pains

From the perspective of Galileo, three particular situations were difficult to determine: the test data generation, timeframe, and the amount of network traffic that Cyberbank generates.

Data Generation

Let's talk about data generation. At first glance, it sounds easy right? You just need to know which tables to populate, create a process to generate random data (probably between some "functional" ranges), and load it into your datasource. But not everything is as it seems. Stored Procedures(SP) and Functions(Func) perform well on small data volumes - 10,000 records, 500,000, even up to 1,000,000. However, everything changes when that data is multiplied by 10x, 50x, or even 150x - producing close to 150,000,000 records in a single table (and don't forget the related data points per record…). Trying to generate that amount of information with an SP or a Func is an overwhelming task for any database engine, the required time is measured in uninterrupted days or even months, something not feasible currently. The team had to pursue a completely different approach, a procedure called "Bulk Inserts." According to Microsoft, a Bulk Insert (BK) is a process that imports a data file into a database table or view in a user-specified format in SQL Server. This became the magical solution, but there was still a missing part. The team needed to create those files in order to import them. For that task, a Java process (such as Python, Javascript, Bash… etc…) was created based on the initial parameterized sequence number for IDs which generates a .csv file with the test data. The lesson here? The larger the dataset, the more difficult and expensive the process became.

Time Frame

Establishing the time frame was not a trivial issue. The initial estimate was 4 months to collect the information, deploy and configure the infrastructure, perform the tests, improve the code, perform database tunings, re-test, and document the process. However, we were tasked with completing all of the tasks within a month and a half. This was a real challenge to the team, but they managed to get through it by creating a joint task force across several Galileo departments and the Microsoft team; recruiting experts, where needed, to help with the data analysis. After all, "Two heads are better than one."

Network Traffic

Last, but certainly not least, the network traffic was a challenging part during the testing. Reaching a massive number of Transactions Per Seconds (TPS) is one of the goals of performance tests, it shows that your application is processing correctly and is performing as efficiently as possible. Having more TPS also means the infrastructure is having to handle a lot of data transfers. Once Cyberbank reached numbers close to 3,000 TPS, the network bandwidth between the application and the database was above 3.5Gbps on average - this was the "magical" number, a maximum quantity of transactions obtained (3,000) with PaaS (Azure SQL Database). After trying endless setups from infrastructure improvements of business critical tiers, hyperscale to O.S. tuning on the workload VM and AKS nodes, the application was not able to show any performance improvement. As a result, the team turned to an IaaS type of deployment on the database engine (Microsoft SQL Server) that enabled the traffic volumes we were looking to achieve. Once tested with the VM powered database, the network bandwidth was close to 12Gbps in border cases where the TPS hit 13.2K TPSs.

Testing - Pros & Cons

Each of the previously described situations highlight the complexity of simulating real-life scenarios on performance tests. A seemingly simple task like loading data can sometimes be the most difficult part of testing. Something as trivial as a parameter on an O.S. can be a significant bottleneck in TCP port assignments and negatively impact results - making it challenging for the team to see an inch of improvement despite weeks of effort.

Performance tests should not be taken lightly as they often require unwavering commitment, analysis, re-testing, trial and error, troubleshooting and much more. Regardless, numerical data and analysis are highly effective in helping banks make important business decisions related to the future of the business, even if the testing process demands steadfast patience and persistence.

*This scalability report was initially conducted in October 2021 by Technisys in conjunction with Microsoft. Subsequently, Technisys was acquired by SoFi in 2022 and soon thereafter was rebranded under its affiliate's name, Galileo Financial Technologies.