



INSTITUTE *of*
TECHNOLOGY
CARLOW

Institiúid Teicneolaíochta Cheatharlach

CLOUD AND SYSTEM INTEGRATION SOLUTIONS

RESEARCH

CONLETH KENNEDY
OCTOBER 5TH, 2019



STUDENT NUMBER: C00172853

LECTURE: GREG DOYLE

Contents

- 1 Table of Figures 2
- 2 Abstract..... 3
- 3 Introduction 4
- 4 Business Problem(s)..... 5
 - 4.1 Mainframe reliance..... 5
 - 4.2 Client Integration..... 5
 - 4.3 Data Accessibility 6
- 5 Possible Solution Ideas 7
 - 5.1 Buy Over Build 7
 - 5.2 Point to Point Integration..... 8
 - 5.3 Middleware 9
 - 5.4 Message Orientated Middleware 10
 - 5.5 Service Orientated Architecture 11
 - 5.6 Cloud Computing Platforms 12
- 6 Possible Technologies 13
 - 6.1 Azure 13
 - 6.2 AWS..... 14
 - 6.3 AWS Lambda 14
 - 6.4 Kafka..... 15
 - 6.5 Kinesis 16
 - 6.6 IBM WebSphere MQ..... 17
 - 6.7 IBM Integration Bus (IIB) 18
 - 6.8 MuleSoft..... 19
 - 6.9 QlikView 19
- 7 Testing..... 20
 - 7.2 Junit..... 20
 - 7.3 N-Unit..... 20
- 8 Conclusion 21
- 9 References/ Bibliography 22

1 Table of Figures

- Figure 1. Point To Point 8
- Figure 2. Middleware architecture..... 9
- Figure 3. Message Oriented Middleware 10
- Figure 4. Service Oriented Architecture 11
- Figure 5. Cloud Computing 12
- Figure 6. Azure 13
- Figure 7. AWS..... 14
- Figure 8. Kafka..... 15
- Figure 9. AWS Kinesis 16
- Figure 10. MQ Architecture 17
- Figure 11. IIB architecture..... 18
- Figure 12. Mulesoft architecture 19

2 Abstract

Integration has always been one of the most complex challenges in the IT industry over the last 40 years, as the digital age is upon us and computing has become more affordable for the average person the number of devices and data has grown exponentially as expected. The necessity of these applications and devices to communicate efficiently is important. We have never been in such a connected world as we have today with the access to high-speed broadband and the cost of cheap storage the accessibility of data and information to the public in comparison to 10 years ago is something to behold.

The goal of this project is to explore the different means and methods of integration that are being used more commonly in the market today, examine these practices in live software ecosystems and evaluate how they run and interact with other systems and compare the overall implementation.

3 Introduction

At Unum over 36 million lives are protected worldwide. The company has a focus of helping people through difficult times by supplying income protection, cancer, vision and dental insurance etc. Unum use a business to business model which for the most part only target clients with over 200+ employees. Typically the customer is billed at payroll for the coverage. Income protection is extremely important in the USA, two thirds of people live from paycheck to paycheck without income protection and could face extreme poverty due to the lengthy process of unemployment. Unum accommodates short and longer terms of unemployment due to illness and injury etc. This makes these payments much faster than government services.

In recent years within the insurance market in the USA technology seems to be the driving factor on the evolution of the insurance market. As part of Unum's digital business approach, Human Resource Connect (HR Connect) is one of the most innovative solutions the company has produced. The current version of the solution has a unique integration with Workday which has given the company an advantageous lead on competitors. This integration allows an individual to submit a leave of absence into the workday system which will then connect with Unum's API's to submit a claim in tandem with the leave of absence with no effort from the client. Workday itself has around 2000 large customers and 135 of them being listed in the Fortune 500 companies. This has allowed Unum to partner with some very prestigious clients from a business and technological aspect and hopefully created long term relationships that can generate large amounts of revenue.

The key reason as to why Unum was able to capture this new business was due to integration with a large cloud solution. From this success story Unum has taken an API first approach to connecting with their customers. Five years ago, a customer was happy to fill out paper forms in order to make a claim for a leave of absence but in today's digital world, paper is not thought to be the first method of interacting with a large multinational company. Unum have adhered to this need and moved focus to the theme of integration and cloud.

4 Business Problem(s)

The three main challenges Unum face today and are determined to solve is

1. Reliance, cost and support of legacy systems and data storage
2. Integration with clients
3. Accessibility to data

4.1 Mainframe reliance

Unum is a company that was founded in 1848 and over that long history of business the company has went through numerous mergers and acquisition of many small to medium insurance providers. Due to this they have acquired a 20% control of the US insurance market for disability insurance which has generated a lot of revenue but also complicated their internal processes and systems due to laws and regulations over many states and countries. The laws have led to complex business logic to be integrated into legacy systems and due to market demand, it has led to many products to sit on top of these systems. As - Menge, F. (2007). States "First, it is nearly impossible to develop one huge application which performs all business functions of a typical enterprise because there are far too many requirements."

These systems were implemented in the late 1970's and were a huge investment for the company due to the cost of memory and storage at that time. These systems are the backbone of the company and have served the company well over the years but as they become older and older developers retired, upskilled or left. With a lack of skills in these technologies with recent software graduates they have left Unum in a difficult situation.

These systems continue to work today and are efficient at what they do but has a large price tag associated for support and testing. In recent years the company has adopted some of the newer cloud systems such as Salesforce, Fineos and Microsoft Dynamics with the intention of managing some of the business processes better and retiring older systems. Although there is still a need for mainframe to administer the policy information. This eventually needs to be changed without impacting the company's growth goals in an ever-evolving market.

4.2 Client Integration

Now to address the second issue, customer integration has also been a challenge for the company over the years. This typically involved the client gathering employee information into large excel sheets which were transferred over to Unum systems and mapped into mainframe by homegrown mapping software on a monthly or bi-weekly basis. This process involved a lot of specialized coding for specific keywords and was often prone to failure by misspellings, alternative wording etc. Also, another issue being that it was not on demand for customers to make updates and requests when they see fit and typically didn't follow industry standards of how integration should be implemented. The client typically was notified by email or phone call that the structure of the document did not match the systems criteria and would have to be changed. This was not a good customer experience for the client and a poor turnaround time in comparison to what could be done with some of today's software.

4.3 Data Accessibility

Due to the many acquisitions as specified above this has meant that Unum have also inherited a multitude of data stores and business workflows. Over the years due to cost, planning, requirement changes and several other reasons data is now spread across the organization and no one fully understands the entire architecture alone due to the complexity. Along with this many of the integrations to these data sources we're not reusable. Analytics and Data Science are now important pieces to the puzzle today to find where a business can improve, and customers now expect more accessibility to their data.

5 Possible Solution Ideas

5.1 Buy Over Build

In recent years in Unum the option of buy over build has been a talking point when faced with a challenge or idea. Previously Unum had been known for being a Microsoft software house (still some truth to this today) and typically looked inward for solutions from their IT team opposed to external to the marketplace. The disadvantage of the build approach was that the design, build and test phase were undertaken by the company. This costed the company the time to build it while also taking on support of the solution after it had been delivered.

The buy over build approach allowed the company to look externally and do some research on the current tools in the market to see how they performed compared to competitor's along with the opportunity to have the infrastructure in the cloud. Whilst the buy approach seems attractive it does come with its drawbacks; some solutions can be quite rigid and not allow much customization along with this, updates of the software could be pushed resulting in bugs which are out of your control. Also, one other disadvantage was tying the organization into an expensive long-term contract with other companies while introducing specific training for the employees in the buyer company to learn the new software from a usability and IT perspective. The procurement process through legal red tape can also be an issue.

5.2 Point to Point Integration

Point to point or synchronous integration is the simplest form of integration between two systems. This is where one system makes a direct connection to send or receive information from another. These integrations are typically easy to implement using protocols such as HTTP requests. They allow for low short-term risk and do not require a specialist skill set and there is typically a greater degree of control over the API by the application development team. With that there are certain downsides to this method of integration. Over time as the organization grows there is a high chance more and more applications require more information from other systems as requirements grow. In times of outage of these systems it can cost the company a lot of revenue and costly fees in support and audit. Along with this there is lower reusability of API's across the company also and different implementation methods due to language or technology will drive up costs over time.

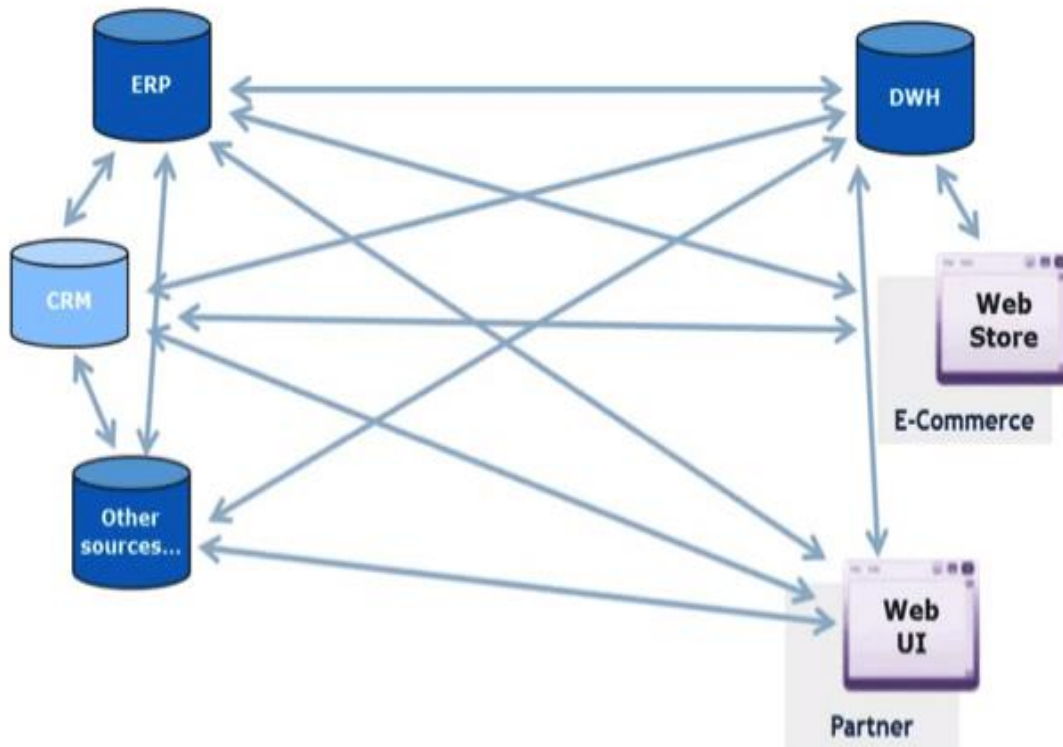


Figure 1. Point to Point

5.3 Middleware

The idea of middleware and ESB's (Enterprise Service Bus) is still a relatively new concept in the software development world as - Menge, F. (2007). states "A typical scenario is that an enterprise runs hundreds or thousands of applications, which could be custom built, acquired from a third party or parts of legacy systems". This follows the exact problem Unum faces today as the applications and systems within the company are required to communicate but the problem is most of them are different platforms, data format or language. These systems should not be closely coupled to minimize complexity and dependence on each other whilst keeping down cost and costume code. Along with this many connections between multiple systems without one source of logging could be a nightmare to debug and trace the source of failures. The tricky part is the external piece as one size does not fit all in the software world "Enterprise Application Integration (EAI), becomes even more interesting if applications of external business partners are to be integrated." It's important that any connection made to an external source allows an asynchronous connection this means that we can receive messages and data and not require a response within the transaction this allows the company to take the incoming requests and process them as they see fit and return a response later.

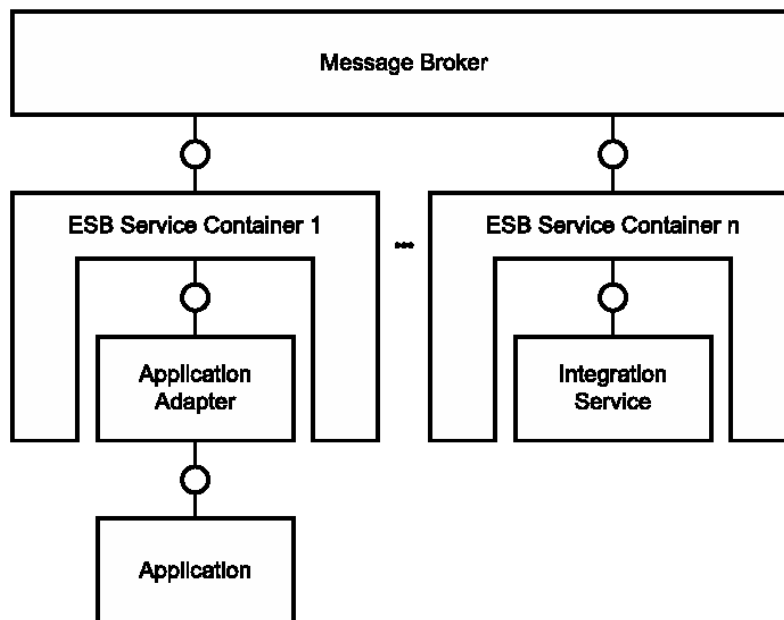


Figure 2. Middleware architecture

5.4 Message Orientated Middleware

Message orientated middleware is the idea of a system used specifically as a routing service for messages within an enterprise to decouple and allow systems to easily communicate. This means that software teams can simply add a message to a location and not worry about whether the receiver of the message is currently able to receive it. This system typically allows for a few different methods of how messages can be submitted to a receiver. The one drawback of this is that it only can perform delivery and routing tasks and not transformation. This means that an enterprise should set a standard of what format the messages should be and the development teams should ensure that the message is transformed into that format before committing to a location.

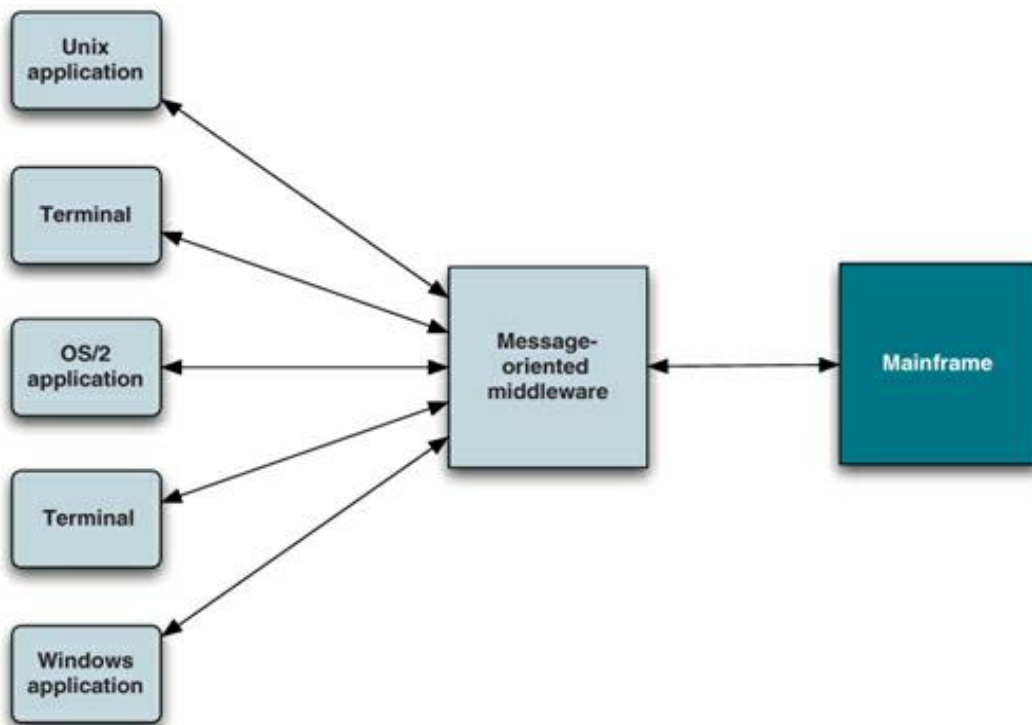


Figure 3. Message Oriented Middleware

5.5 Service Oriented Architecture

The Service Oriented Architecture (SOA) is the concept of structuring a company's API's based on the services they provide. The idea of this is that the organization builds up a portfolio of web services typical in the REST (Representational State Transfer) or SOAP (Simple Object Access Protocol) format this allows the services to be reusable for other projects or to be remodeled if the needs of the company change over time with little effort and speed.

It's important these services follow one standardized data format so that systems can communicate simply between legacy and newer systems, typically a data format such as XML (Extensible Markup Language) or JSON (JavaScript Object Notation) is chosen. Due to SOA being introduced it allows for great scalability and allows developers to distribute applications to multiple locations/departments within an organization and multiple businesses on the internet.

Typically, these services are released in a server farm in a load balanced environment this should allow services to handle a high amount of network traffic with ease along with gateway software to allow for security, authorization and routing of requests. The key principle of this model is that the services are reusable, reconfigurable and scalable this allows the company to reuse existing services where possible that they know are reliable and hence a better finished project with better versatility and greater speed to market.

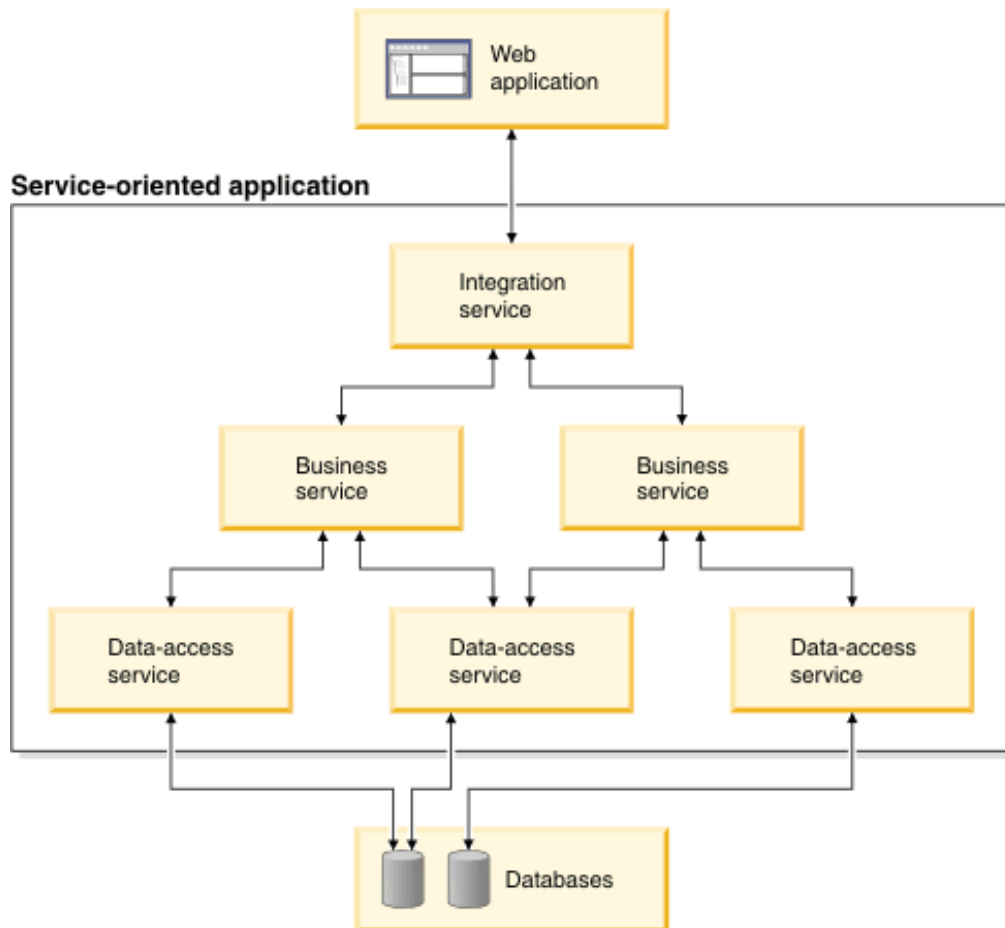


Figure 4. Service Oriented Architecture

5.6 Cloud Computing Platforms

In recent years the idea of cloud computing has been a hot topic in meeting rooms and web conferences but what does cloud computing provide? The idea of Cloud computing is to provide one of three services over the internet these services are known as SaaS (Software as a service), PaaS (Platform as a service), IaaS (Infrastructure as a service) or all of them in tandem and the customer is typically charged by usage.

This allows smaller companies who are starting off to not require a large amount of capital to invest in expensive hardware and software before they can release a finished product instead, they can move their code into the cloud and utilise some of the tools and configurations provided by companies such as Amazon or Microsoft and scale as the software's user base grows.

The advantage of this is that you can focus on the important parts of the software requirements and infrastructure support, versioning, security and load balancing is handled by someone else. This frees up your developers to focus on solving the business problems. Along with this it enables your business to dabble in concepts such as machine learning and analytics that you have not had the resources for as these services are provided within the platforms.

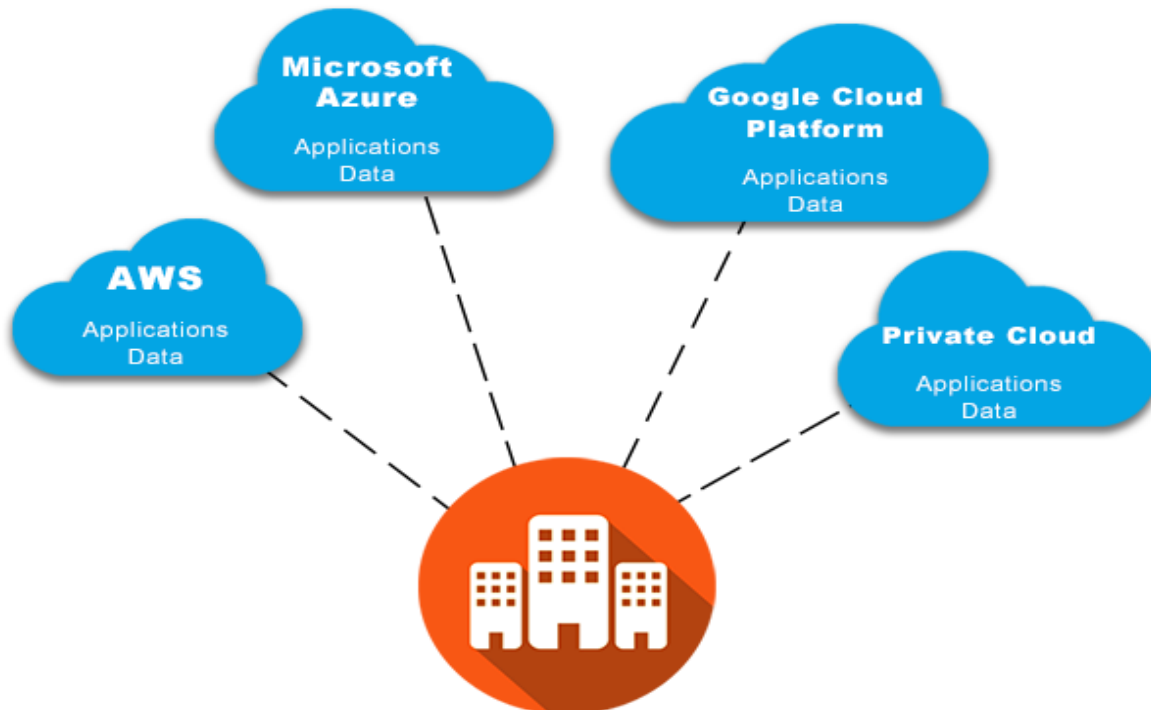


Figure 5. Cloud Computing

6 Possible Technologies

Unum as stated above is primarily a .net hub although it has taken a liking to a more open source approach in recent years. This allows the project to look towards two languages/tool chains being C# and Java. Both languages have similar support and capabilities with one major stand out difference of Java having the advantage of being cross platform due to the Java Runtime Environment (JRE) but otherwise both languages are the two most widely used languages in Unum. This should be considered when choosing other components of the project to consider that each language has the necessary capabilities to communicate with each other.

6.1 Azure

Azure is Microsoft's cloud computing platform that allows developers to build, test, deploy and process code in the cloud while also providing services such as virtual machines for Windows and Linux and data storage. Azure supports a wide variety of Microsoft specific products such as Team Foundation Server (TFS), Xamarin, Windows and PowerShell which are all products that Unum currently use today along with the bonus of support for languages such as Python, Java, php and Node JS. The platform also allows for a range of different storage such as relation style databases, NoSQL and unstructured BLOB (binary large object).

The platform also provides a messaging queue service for async communication called Microsoft Azure Service Bus which will be of use and caters towards the SOA model. One of the unique aspects of Azures functionality is Azure functions which consists of a serverless tool which allows a user to execute a piece of code on the cloud platform without the worry of resource management this means the customer does not need to worry about scalability and pays by the usage. The service also supports a wide variety of languages and handles security along with giving performance metrics on your code.

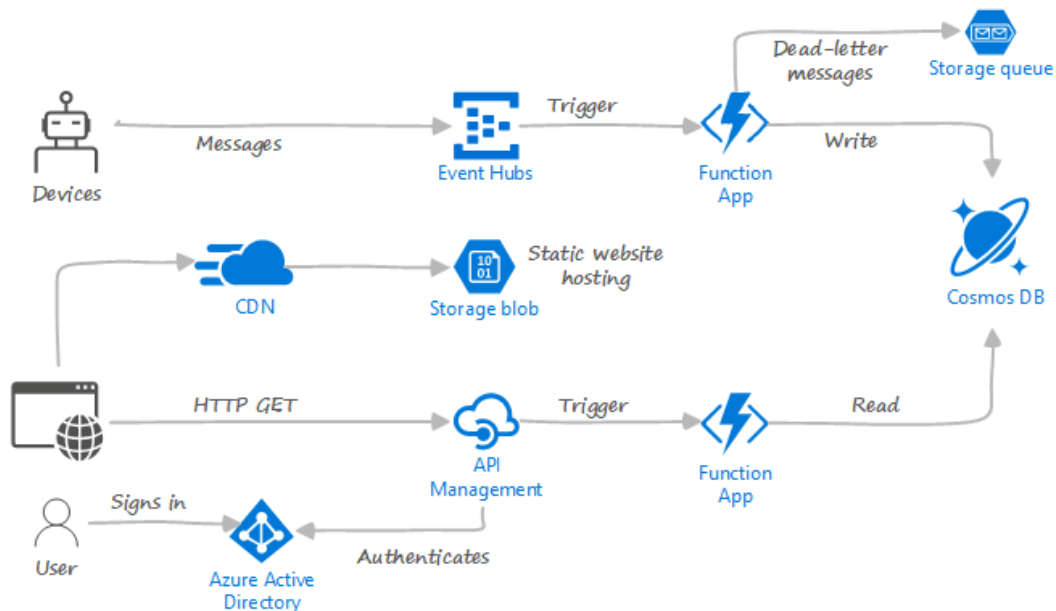


Figure 6. Azure

6.2 AWS

AWS is another cloud computing platform which provides much of the same services as Azure, but the one main difference is the maturity of the platform. The platform has many more custom-built tools such as S3 for storage which supports multiple different no SQL databases such as Dynamo DB, Aurora and relational databases such as MySQL which can work well with AWS Athena for fast scalable serverless querying. EC2 is their computing platform which allows a user for a secure, scalable way of obtaining servers for a project, the platform allows you to pick how many servers you need and what OS at the touch of a button. The platform has data streaming software such as Apache Kafka and AWS Kinesis. Elastic Bean Stalk is also a useful tool for deployment of web applications.

6.3 AWS Lambda

AWS Lambda is a serverless function within the AWS platform, lambda allows a user to run a software function in the cloud based on events that are passed to a lambda function. These events typically come through AWS gateway. The main selling point of this method of code execution is that everything is handled by AWS irrespective of the language and platform, AWS has configured lambda to include all the necessary runtime environments and packages along with this the service will scale automatically and you will only pay by usage.

One of the useful features of lambda is the layers that you can configure and use with your lambda functions this allows you to configure additional libraries, runtimes and dependencies as part of your function. These layers can be shared with an organization or the entire AWS community if the user chooses meaning that there is a community of useful layers that can be reused as the platform grows.

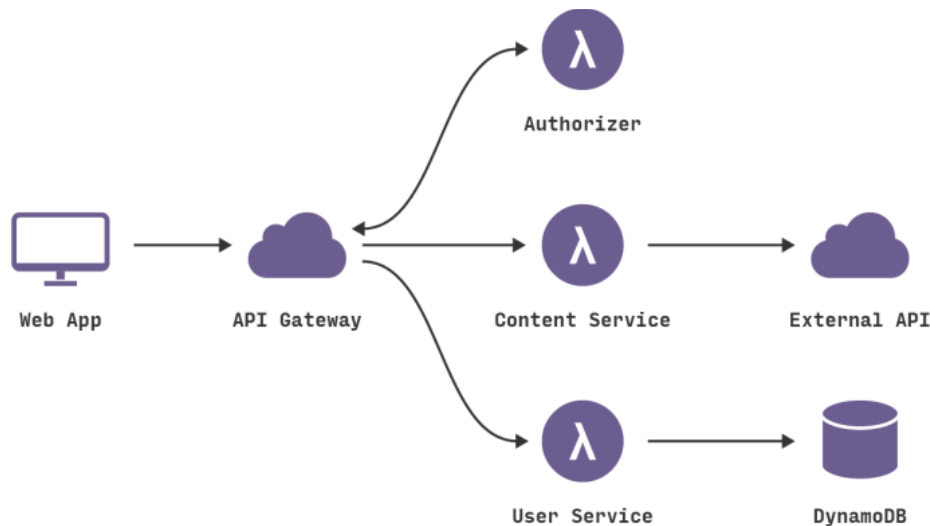


Figure 7. AWS

6.4 Kafka

Apache Kafka is an open source streaming, messaging and data transformation software made by LinkedIn. The software uses the publish and subscribe model for data transfer. Kafka listens for incoming messages from publishers for some period of time, publishers send a piece of data called a topic. A topic can be consumed by many different consumers all with different purposes. Data is typically processed in real time but can be configured to run in batch also.

The data is typically divided into partitions and the user decides the order of the data. The main issue of Kafka is getting data into a Kafka cluster, these are usually fed from connectors such as HDFS (Hadoop file system) or a relational database and the second issue is processing. How the software scales are by using multiple instances of a Kafka cluster. Kafka is also very reliable for data delivery as it will store the message within the system in the case where a consumer is not available and can pick up where it last left off. Lastly the system can also have stream processors, stream processors can do transformation on data in order to create new topics.

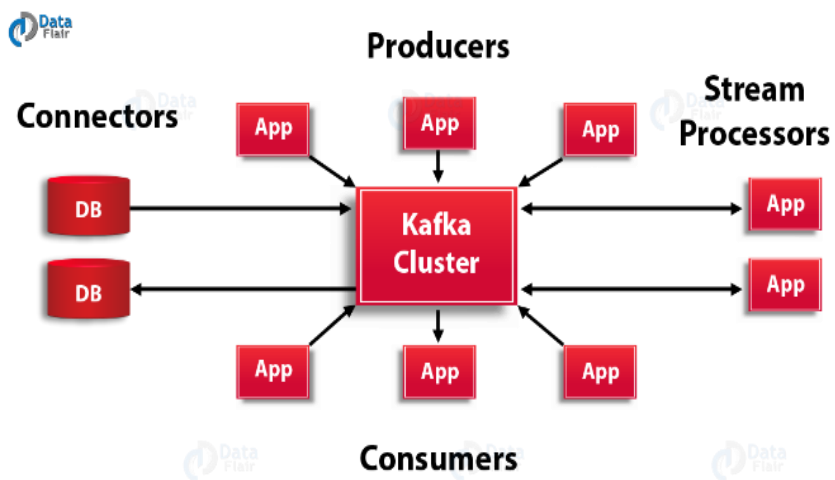


Figure 8. Kafka

6.5 Kinesis

Amazon Kinesis is a data streaming tool that can connect to hundreds of thousands of data sources with low latency. The platform allows for real-time streaming of data of many different formats such as video, images, audio and data such as log files. The ability to stream in real time can be a huge advantage as you can make decisions almost instantly instead of waiting on large batches to run. The application automatically handles your infrastructure as it runs and therefore is able to take in such volumes of data without any effect on performance. Typically, users will use Kinesis to run analytics or machine learning within the AWS platform to make clever business decisions or to just give a health check on a system or how sales might be performing on a real time basis. The tool can be complemented with either EC2 or Lambda to carry out processing tasks.

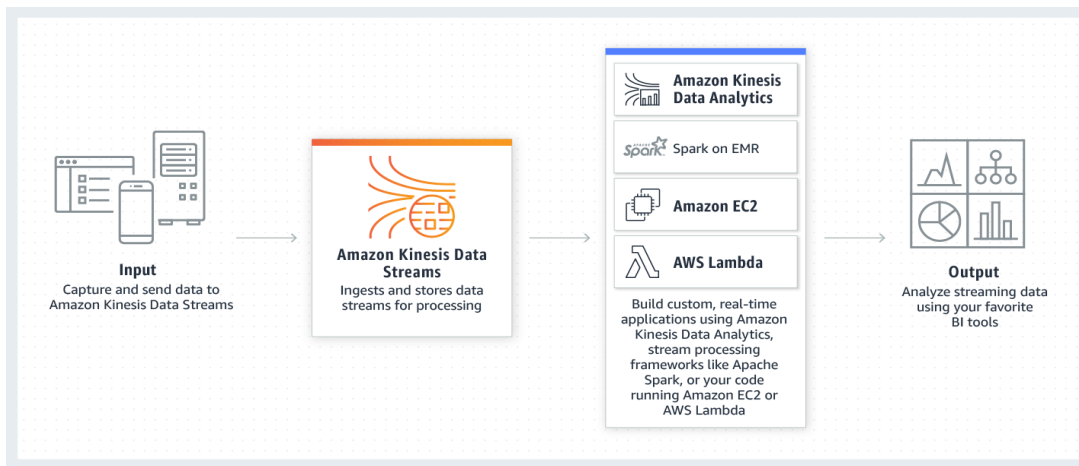


Figure 9. AWS Kinesis

6. 6 IBM WebSphere MQ

Unum's current tool for messaging and middleware is IBM MQ. It is the most used messaging system in the industry today with it having a hold on 70% of the market. This software product follows the MOM(Message Orientated Middleware) model of messaging. Some of the main advantages of using MQ is that it is cross platform with the use of channels this is extremely useful in a large enterprise. The software is queue based where you define a sender and receiver queue on the queue manager.

This system works async and takes care of all storage logs and communication. The reason why this software is so popular is its ability to integrate well with mainframe for transfer of data. The sender and receiver applications just need to know the name of the queues and agree on the message format. The software supplies its own API for use with the 6 functions being

- Connect to Queue manager
- Open queue
- Put or get messages
- Close Queue
- Commit rollback
- Disconnect

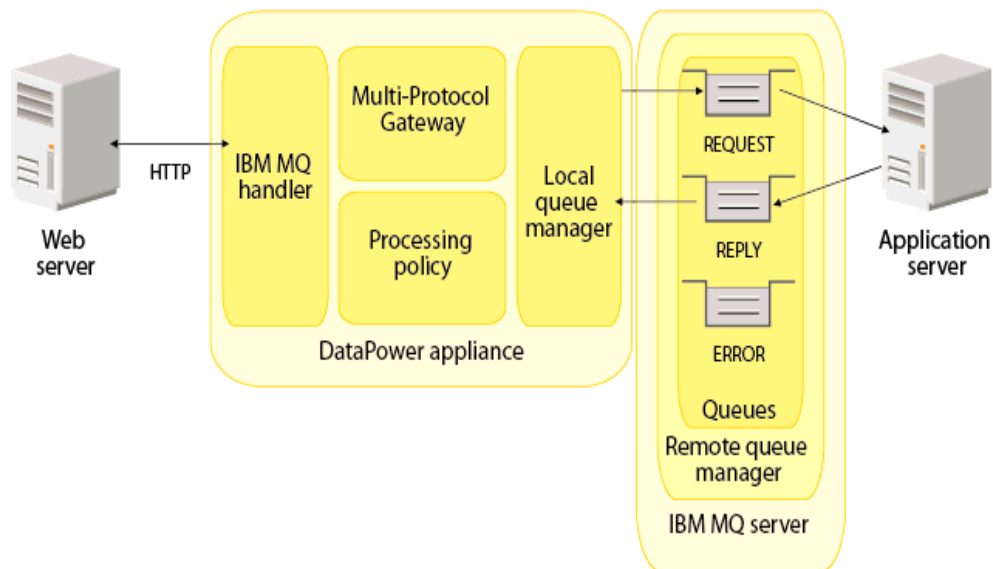


Figure 10. MQ Architecture

6.7 IBM Integration Bus (IIB)

IBM Integration bus is an enterprise series bus used for routing, message enrichment and transformation processes between systems regardless of message format or protocol used. The product is typically complemented with IBM MQ as IIB'S strong message transformation support works well with preparing the data for another system.

Typically, the most common use case within Unum is to get data from the mainframe into MQ where IIB picks it up and converts the message to a friendlier format such as xml or json. IIB also supports a range of inputs such as SOAP, REST, HTTP, File and TCP/IP. Along with this many different data formats such as C, Cobol, XML, EDI and HIPAA which are all commonly used within the company.

The tool supports a range of different methods to transform data using a graphical map interface, a language known as ESQL (Extended Structured Query Language), Java and XSL (Extensible Stylesheet Language). The tool also comes with a range of connectors for tools such as Kafka, Salesforce, WTX (WebSphere Transformation Extender) and .NET. The tool also can quickly create rest API's to mainframe by proxy using swagger documents. Many of Unum's API's are currently created in this tool with all ready predefined enterprise logging system. This tool does require a license.

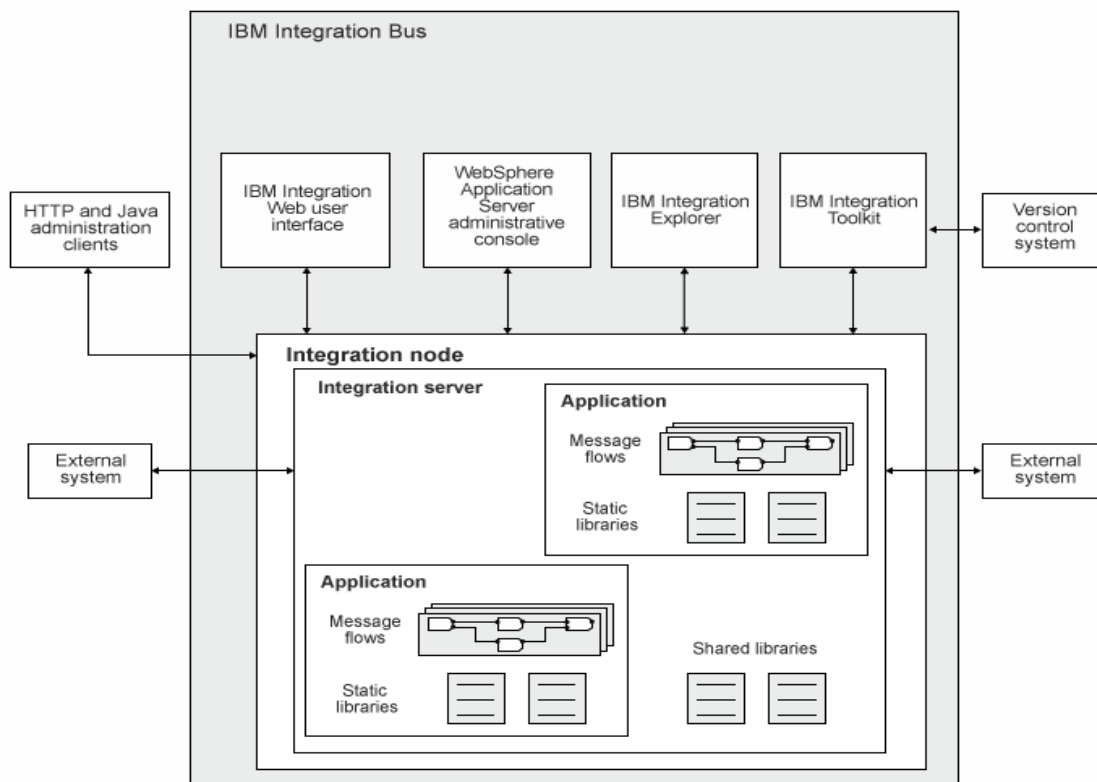


Figure 11. IIB architecture

6.8 MuleSoft

MuleSoft is a lightweight ESB based on the java language. The central component of Mule is a service container for so called Universal Message Objects (UMOs). The key selling point of MuleSoft is the number of connectors the software has in the case, that there is a custom connection required MuleSoft provide a connector SDK to develop it. This means Mule enables easy integration with any type of system. The data formats are like that of IIB supporting XML, Json etc. MuleSoft allows a component-based architecture which enables significant component reuse.

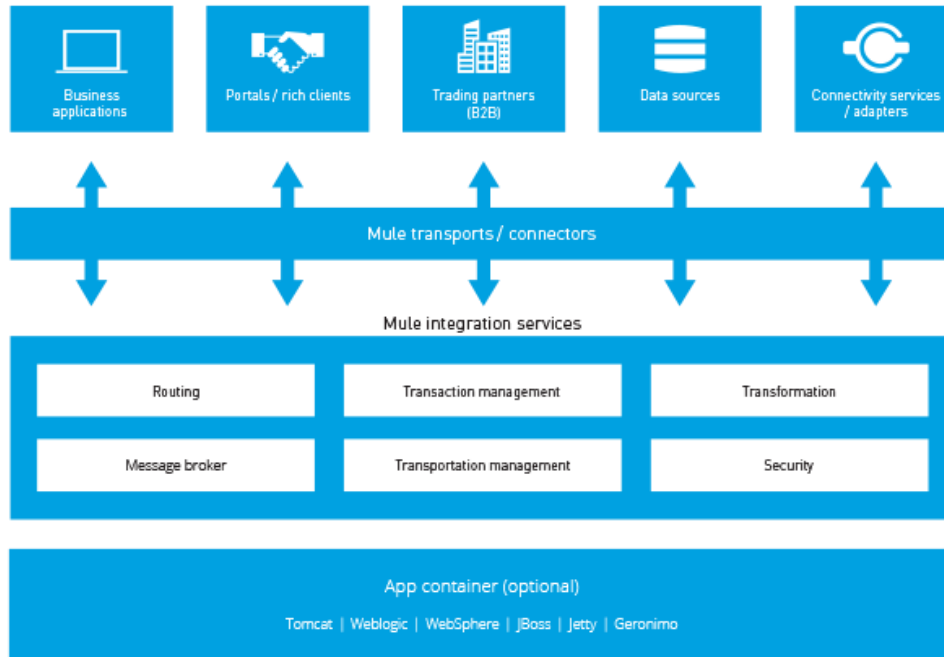


Figure 12. Mulesoft architecture

6.9 QlikView

QlikView is a business intelligence and analytics tool that loads all data in memory in order to allow for quick and easy data manipulation. The tool provides multiple different methods to consume data such as ODBC (Open Database Connectivity) connector which allows QlikView to connect to a wide variety of databases such as Teradata, Oracle, MySQL and DB2. The tool also has the ability to have connections to NoSQL databases such as MongoDB. Lastly the tool can consume REST and SOAP API's. Once the data has been loaded into the application the tool can create charts and trend graphs or simply display the data in a list. Again, Unum owns a license to this piece of Software.

7 Testing

7.2 Junit

J-unit is an open source testing framework for the Java programming language. Junit allows a developer to test individual pieces of code by creating test methods called unit tests. Unit tests include simple tests of a method of passing in an input and asserting the expected output if at any point where an unhandled exception or assertion failure occur the test will either be marked as failed or skipped. The goal of when writing a test for an application is to achieve 100% branch coverage of a class while also testing the functionality with realistic scenarios. This gives the developer the ability to run a series of checks at the click of a button and get a health check on the code and see if new code has had a negative effect on existing code.

7.3 N-Unit

N-Unit is the C# version of J-unit. It enables a developer to do the same as J-unit along with providing such functionality such as test suites which let you organize your tests based on a category. Also, can use annotations to run pre and post conditions like J-unit.

8 Conclusion

As time has progressed and the requirement for better conductivity between systems it seems as though point to point integration in large enterprises seem to be unfeasible and unmaintainable in the long term. As leading technology companies have built middleware solutions for integrations across a large network and cloud providers have built real time connectors to enable companies to build and scale their solutions. It appears Unum have no choice but to jump on the bandwagon in order to enable them to compete with rivals as these technologies will inhibit greater speed to market and the automatic ability to scale at the click of a button or involuntary. Having on premises infrastructure may give the company more control and just an initial cost of implementation and maintenance costs of the servers but standing up infrastructure takes time and human resource while losing out on new toolsets and perks of being on a cloud platform.

9 References/ Bibliography

- [1] Takend.com. 2019. *ESB vs P2P: Why It's Finally Time to Ditch P2P*. [ONLINE] Available at: <https://www.talend.com/resources/esb-vs-p2p/>. [Accessed 01 November 2019].
- [2] Falko-Menge. 2007. *rapidminer Enterprise Service Bus*. [ONLINE] Available at: <https://www.semanticscholar.org/paper/Enterprise-Service-Bus-Menge/75393c46ab62d8c25d13f79d68ef42e232474b53> . [Accessed 01 November 2019].
- [3] Geoffrey C. Fox, Vatche Ishakian, Vinod Muthusamy, Aleksander Slominski. 2019. *Status of Serverless Computing and Function-as-a-Service(FaaS) in Industry and Research*. [ONLINE] Available at: <https://arxiv.org/abs/1708.08028> [Accessed 01 November 2019].
- [4] Softwareengineeringdaily. 2019. *softwareengineeringdaily*. [ONLINE] Available at: https://i0.wp.com/softwareengineeringdaily.com/wp-content/uploads/2016/08/serverless_webapp.png?resize=730%2C389&ssl=1 . [Accessed 01 November 2019].
- [5] Educba.com. 2019. *Java vs C#*. [ONLINE] Available at: <https://www.educba.com/java-vs-c-sharp/>. [Accessed 01 November 2019].
- [6] Microsoft. 2019. *Serverless*. [ONLINE] Available at: <https://docs.microsoft.com/en-us/azure/architecture/serverless/code>. [Accessed 01 November 2019].
- [7] Microsoft. 2019. *Azure tools*. [ONLINE] Available at: <https://azure.microsoft.com/en-us/tools/>. [Accessed 01 November 2019].
- [8] data-flair.com. 2019. *Kafka Architecture and Its Fundamental Concepts*. [ONLINE] Available at: <https://data-flair.training/blogs/kafka-architecture/>. [Accessed 01 November 2019].
- [9] Bruce Snyder, Dejan Bosanac, and Rob Davies. 2011. *Active MQ in Action*. [ONLINE] Available at: <https://livebook.manning.com/book/activemq-in-action/chapter-2/25>. [Accessed 01 November 2019].
- [10] IBM. 2019. *Service-oriented architecture (SOA)*. [ONLINE] Available at: https://www.ibm.com/support/knowledgecenter/en/SSMQ79_9.5.1/com.ibm.egl.pg.doc/topics/pegl_serv_overview.html. [Accessed 01 November 2019].
- [12] MuleSoft. 2019. *What is mule ESB*. [ONLINE] Available at: <https://www.mulesoft.com/resources/esb/what-mule-esb>. [Accessed 01 November 2019].
- [13] IBM. 2019. *MQ Basic architecture*. [ONLINE] Available at: https://www.ibm.com/support/knowledgecenter/en/SS9H2Y_7.5.0/com.ibm.dp.doc/mq_basicarchitecture.html [Accessed 01 November 2019].
- [14] IBM. 2019. *IBM Integration Bus technical overview*. [ONLINE] Available at: https://www.ibm.com/support/knowledgecenter/en/SSMKHH_9.0.0/com.ibm.ertools.mft.doc/ab20551.htm. [Accessed 01 November 2019].

[15] Multicloud. 2019. *Muti-Cloud*. [ONLINE] Available at: <https://avinetworks.com/glossary/multi-cloud/> . [Accessed 01 November 2019].

[16] Amazon. 2019. *AWS Kinesis*. [ONLINE] Available at: <https://aws.amazon.com/kinesis/> . [Accessed 01 November 2019].

[17] Junit. 2019. *Junit*. [ONLINE] Available at: <https://junit.org/junit5/> . [Accessed 01 November 2019].

[18] Nunit. 2019. *Muti-Cloud*. [ONLINE] Available at: <https://nunit.org/> . [Accessed 01 November 2019].