Whitepaper

aleios
part of Theodo Group

A Cloud Approach to Airline Recovery

Building for Disruption with Serverless

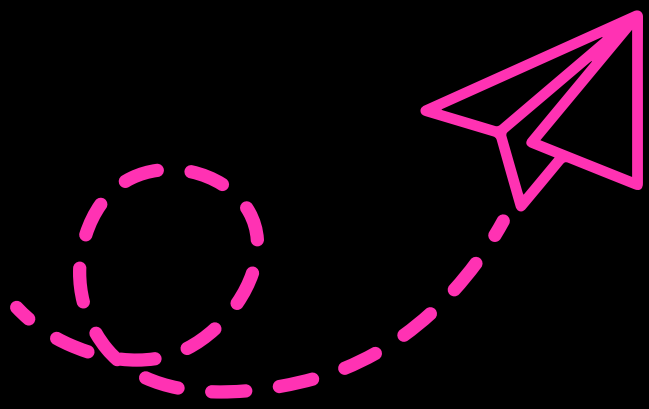


Table of Contents

Executive Summary

The airline industry is still recovering from the COVID-19 pandemic, with passenger preferences forever changed and parallel industry trends accelerated by half a decade. How airlines build back from this disruption will decide the winners and the losers in an ever-changed travel landscape.

To successfully react to these changes and build for the new trends of sustainability, personalization, and modern payment mechanisms, airlines need to build a foundation that enables them to experiment rapidly and scale.

Delivering the customer experience needed in an increasingly competitive landscape can't be achieved with the technical debt found in legacy systems after industry-wide underinvestment in digital. Modernizing to adopt a modern cloud (i.e. Serverless) will provide the foundation needed, but the strategy to migrate the workload will be the critical decider of success. Moving away from a "big bang" all-or-nothing migration approach and adopting a progressive migration strategy is essential.

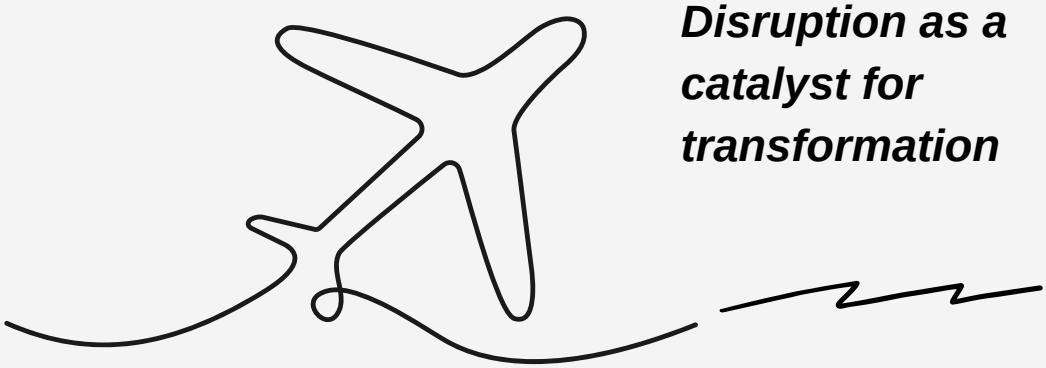
The Minimum Viable Migration (MVM) framework provides the approach, tools, and techniques to move to the modern cloud in a series of iterative vertical slices shipped to production. The emergence of Microfrontends and Edge Compute further support achieving an MVM approach.

To catch up with competitors it is essential to prioritize modernizing the areas of the system with the most significant experimentation potential and greatest impact on the customer experience. For airlines this typically involves prioritising the e-commerce function. Going more granular specific areas of the booking flow and payment mechanism can be modernized and released first.

2023 Airline Challenges

Disruption as a catalyst for transformation

2023 Airline Challenges



Disruption as a catalyst for transformation

The immediate impact of the COVID-19 on the aviation industry was self evident. In an instant, everyone across the globe had to change, and the full recovery from this is still years away. The long term impact on passenger preferences and expectations are far more subtle and nuanced than short term ones. How airlines build back to account for this change will determine their ability to scale and expand in the post-pandemic world.

Passenger intention is blurring

The key long term changes relate heavily to shifts in passenger behaviour. Passengers now expect flexibility, require increased safety procedures, and are blurring the lines of travel intention. It's no longer the case that a trip is simply business or leisure, but a combination of intentions that can change throughout the trip itself.

Leisure will dominate the industry's recovery as businesses have learnt to operate remotely. This requires a change to the current go-to-market approach and has a huge impact on the economic model for long haul carriers who rely on the blend of economy and business travellers to achieve critical seat load and profitability.

The pandemic disruption has also accelerated existing trends with the move to a fully digital customer experience leaping forward 5 years. Passengers were forced to adopt digital vaccination certificates and contactless boarding procedures. This has not only increased app adoption rates but has normalised sharing identification data and accelerated the use of a digital device as an end to end travel tool.

Reacting to the call

How airlines react to this call for change will determine the winners and the losers in the competition for a smaller pool of passengers with access to a broader range of distribution channels. Some airlines have made radical changes while others have just scraped by through the pandemic. Interestingly, those airlines that felt the most pressure through the pandemic (e.g. those not relying on large levels of government aid) will have transformed to find efficiencies. These airlines will be ahead in adapting for the long term, and are positioning themselves to be the winners over the next decade.

Some airlines have made radical changes while others have just scraped by

Transformation needs to be focused for the changed environment and not merely cutting costs to survive. Intentional investment in digital is key to react to the long term industry trends and needs to be the focus of the limited budgets available to airlines to succeed.

Sustainability

Personalization

Modern Payments

Wider Trends

Outside of the pandemic, several pervasive trends continue in parallel. Sustainability continues to move to top of mind for passengers and governments alike. Passengers are travelling more intentionally, reducing the impact of their trips and trying to maximise what can be done in a singular trip. In addition there is a move to more local travel. How airlines cater for this adapted demand in route planning, fleet composition and digital innovations (e.g. personalized suggestions) will determine the level of impact on their relevance in the market.

Customers continue to demand greater personalization and higher quality customer experiences, driven by wider cross-industry changes and disruption from large scale modern online travel aggregation sites.

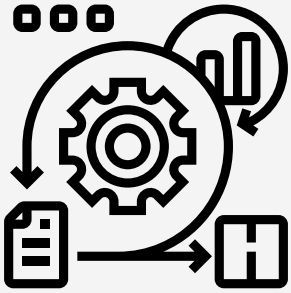
Finally the payment strategy of airlines has become more strategic than ever. There is a crunch point coming for the industry as payments charges continue to cannibalize over half the net profit of the industry. To realise the full value of this airlines need to build a payment strategy as part of customer experience while in parallel continuing to optimize wider dynamic pricing tooling.

Half Payment charges add up to more than half the net profit of the airline industry.

The Need to Modernize

Acting at a pivotal moment

The Need to Modernize



Legacy systems are preventing the change needed to keep up with third-party aggregators

There is a clear need to modernize the systems that underpin the airline industry. The industry has underinvested in technology compared to other industries for a prolonged period and the impact of this technical debt is becoming increasingly constraining. Airlines need to modernize their systems from the ground up to create a foundation to address the emerging trends in passenger behaviour and wider macro-economic trends.

IT is not a Cost Centre
Airlines need to shift their mindset of IT as a cost centre and see the opportunity to enable revenue generation. In a world of aggregation sites and third party search airlines will need to add value to maintain their direct customer relationship. This requires novel features delivered with stability at scale. The current technology landscape of most airlines can't achieve this.

More channels, less passengers
In the past, airlines have managed to establish positive relationships with comparison/search aggregators such as Skyscanner and Google Flights as these were only a source of referral, with the booking is still owned by the airline. However, aggregation has evolved over the past years with the emergence of large-scale online travel agencies encroaching on airlines' customer acquisition.

The emergence of a new trend of end-to-end online large scale travel agencies such as Booking.com has already disrupted the hotel industry. Booking.com has now added direct flight booking, joining Expedia, Kiwi, Trip and others. This disrupts the ownership of the booking - impacting the customer relationship, experimentation, ancillary upsell ownership and payment mechanisms. This disruption is also happening at a time where passenger numbers are still yet to recover to pre-pandemic levels

One of the reasons customers are using these platforms is the ease of use and quality of the experience. Booking.com provides a consistent experience across airlines and supports a wide range of payment options (including Apple and Google Pay) for frictionless checkout.

The Need to Build a Customer Centred Platform

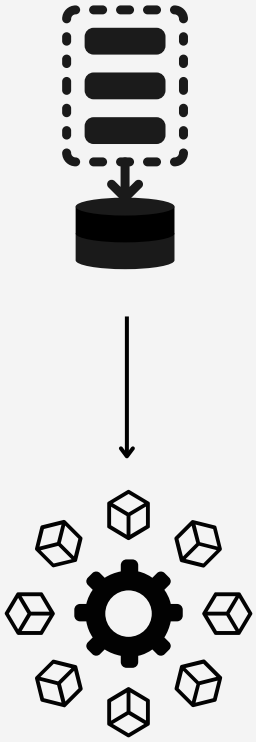
Airlines need to learn from the lessons of traditional retail brands in the battle of the aggregators. Retail brands have had their value eroded by the move to large scale marketplace platforms like Amazon and shopping aggregation search tools like Google. This move is removing the direct customer relationship from the brands, and consumers are starting to doubt the value behind the logo as products are in presented undifferentiated girds of results and related generic items.

Traditional brands such as Nike have moved to a platform strategy to bring together customers, immersive content, third-party integrations and personalization. For instance, their Nike Running Club provides coaching, personalized playlists and even integration with Headspace for wellness content. All of this adds value to direct purchasing and allows Nike to maintain direct relationships with customers and ownership of the purchasing flow.

Retail brands have achieved this platform strategy via investment in technology and digital. The Nike Running Club supports a Mobile Apple, Apple Watch experience and seamless integration with third-party content providers, all while building up the data needed to provide a truly personalized experience. For airlines to do the same, they need to have the ability to build innovative new experiences at speed and scale. This needs a modern technology stack and a Cloud strategy to build, scale, and gather the data needed for advanced personalization.

Lower Total Cost of Ownership, Higher Velocity

The above calls to modernize require higher velocity, i.e. a reduction in time-to-value. This requires the ability to have small independent product-led teams building business value generating features. To support these teams, a technical foundation that abstracts complexity and shifts responsibility for hosting, securing, and scaling away from the airline (i.e. Serverless) is needed. At the same time, there is a need to decouple monolithic systems to enable independent release cycles and eliminate the slow change approval processes that hamper innovation and experimentation.



The wider macroeconomic environment means that airlines must perform at high scale and lower cost and with the ability to have cost linked to demand. The answer is a technological solution (e.g. Serverless) that enables higher developer velocity while reducing the cost to build, maintain, and scale applications (Total Cost of Ownership).

To achieve this the right technology stack and Cloud Strategy need to be implemented. Typically, this includes decomposing the legacy monolithic applications and the layers separating them from legacy third-party.

E-Commerce First Modernization

A Value Driven Transformation

E-Commerce First Modernization

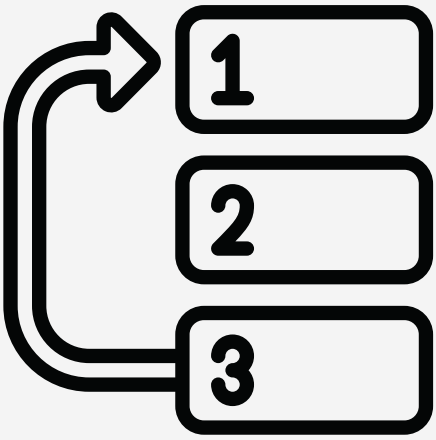
Start with the area of maximum experimentation potential for the customer experience

A working definition of e-commerce is the systems and services that enable regulatory-compliant booking creation and ancillary purchasing. As the primary interface with passengers, it should prioritised in the modernization strategy to sooner realize the benefits of improved conversion, ancillary revenue and customer relationship strengthening.

This is not to say that other operational systems should not be modernized, but typically that should be the later phase of a modernization effort due to the innovation potential contained in e-commerce.

It is typically not a simple task to split out the e-commerce functionality due to the intertwined nature of many monolithic legacy airline platforms. It can be hard to draw boxes around different parts of the system but this is needed in any case to build the target decoupled architecture.

Tools and to approach this decomposition are discussed in the "Minimum Viable Migration" section.



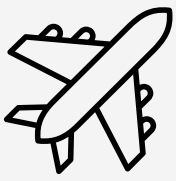
To go a step further, e-commerce itself can be broken down into a finer granularity of functionality. Likely, it is the core booking funnel that has the highest impact on conversion and ancillary upsell. This is often followed by the opportunity to modernize the payments engine.

Modernizing the payments engine enables a wider range of lower-margin options to be added. In addition a decision-making logic that proposes the optimal payment choices to the customer based on margin and conversion can be added.

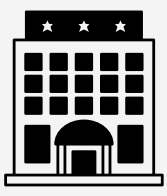
Alongside this traditional view of e-commerce, it's essential to note the increased transactional opportunities e-commerce offers. Unlike in brick-and-mortar retail or online retail, the passenger journey can be a set of micro-transactions adding up to an aggregate trip. Frictionless transactions allow airlines to capitalize on the revenue opportunities before, during boarding, in-terminal, in-air, and between flights, especially as recent trends have shown that passengers are more likely to change their travel preferences and intentions on the go.

Booking

Terminal



In Air
Outbound



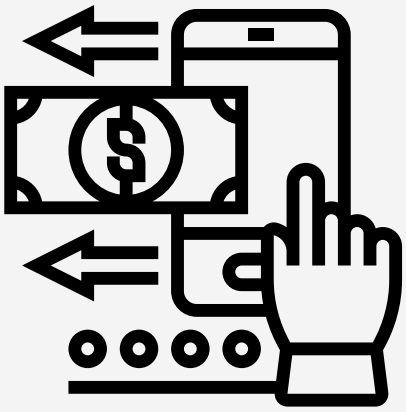
Between

In Air
Return

Payment Modernization

Lower cost with lower friction

Payment Modernization



Flexibility, personalization and smart decision making is key to optimize conversion and cost

The payment ecosystem is changing considerably, with a growing number of options available and trusted by customers. However, credit cards continue to dominate bookings and ancillary purchasing across the industry. While credit cards are well established and trusted by both the buyer and seller, their transaction costs are high. Being able to leverage alternative mechanisms will provide considerable cost savings to airlines across the customer journey. From crypto wallets, to buy-now-pay-later partners (e.g. Klarna) to country specific mechanisms (e.g. iDEAL) the number of options continue to grow.

Credit cards also have a high impact on the derived customer experience. Manually typing card numbers (for a card a customer might not even carry anymore) and Strong Customer Authentication requirements can create high friction. This issue is particularly pronounced in mid-trip ancillary purchasing - an area many airlines are looking to grow as passenger numbers recover. To this end, it's important to note that, unlike traditional retail, airlines support many transaction opportunities across a single trip. Therefore it's key to create low friction future transactions.

Initial Booking | Ancillary Purchasing *At any time* | Flight Change | In Flight Purchases | Claims / Refunds

Optimizing conversion rates across the customer journey requires airlines to control the payment, simple integration with a wide variety of locale-specific providers, personalized suggestions based on fee, and conversion optimisation.

The first step is to abstract any underlying third-party PSS (Passenger Service System) from the payment capability. Typically these PSS systems, often based on a legacy monolithic architecture, lack the support for the variety and control over payment options needed.

The second step is to build a Payment Federation layer that abstracts the complexities of each underlying payment provider into a common interface. Typically this comprises of an independent payment microservice with Event-Driven integration with the rest of the services. The key design of this system is that it can easily add new providers, all while providing frontend teams with a homogenous interface that is abstracted away from the underlying payment choice.

Finally, a Payment Selector Engine is needed to decide which payment option to use. This complexity can range from being as simple as using location-based rules logic to more complex developments such as using only a payment provider supporting the locale of the customer. Other possibilities include using a complex AI-driven approach to predict the conversion likelihood of particular users and display payment options based on their propensity to purchase and associated fees.

When it comes to changing user behaviour, e.g. using credit cards, the alternative needs to either be easier (e.g. in the case of Apple Pay / Google Pay) or provide additional value. The lowest fee options are not always significantly simpler for customers. Therefore incentives need to be created. This can be as simple as varied pricing through to novel loyalty systems and built-in wallets. These later options can enable innovative NFC-based in-flight purchasing and rewards that motivate customers to completely enclose end-to-end trip experiences.

A Note on Regulation

PCI/DSS regulation is key when it comes to payments. Even if external third parties handle all card data, airlines typically transact a volume that still necessitates PCI/DSS auditing.

Adding a wide range of payment options can significantly increase auditing scope, adding cost to both development and external auditors. Leveraging a technology stack (e.g. Serverless) that abstracts the maximum amount of security responsibility is essential to continue rapidly adding payment options.

Serverless

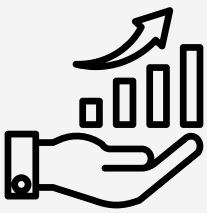
The modern cloud strategy for airlines

Modern Cloud: Serverless

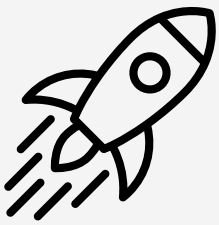
Serverless is the modern way to build applications at high scale and lower cost.

Serverless, by nature, is highly scalable and stable. This means that applications built on Serverless have these properties by default, even when built rapidly. By eliminating the need to manage infrastructure development teams can focus on feature development and spend less time on maintenance tasks. In addition, Serverless services have a pay-per-use model, meaning the costs is directly liked to usage. This reduces over-provisioning and enables an optimal cost profile.

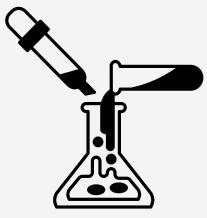
Lower Cost



Higher Scale



Faster Development



Higher Experimentation

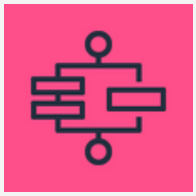
Serverless architectures leverage the best of the cloud by using the truly cloud-native services of a particular cloud provider. On AWS (Amazon Web Services), AWS Lambda is seen as the "poster child" for Serverless, enabling a pay-per-use compute model in which developers send code to be run by AWS in response to events. Serverless is not constrained to just Compute though, there are Serverless services to compose the entire application, enabling a the benefits of Serverless across the entire stack.

Example AWS Serverless Services

Lambda
(Compute)

Dynamo DB
(Data)

EventBridge
(Event Bus)



Step Function
(Orchestration)

While some airlines have lift-and-shifted (e.g. moved their applications unchanged) into Cloud hosting environments, only a subset have managed to leverage the full potential of the Cloud in Serverless. The latter have demonstrated massive reductions in TCO, increased developer throughput, and increased stability.

60%
Reduction in Operational Costs

71%
Reduction in Unplanned Outages

33%
Increase in Developer Productivity

IDC 2018 Study [1]

This shift in responsibility to the Cloud provider is a form of outsourcing that is complementary to the operating model of airlines, allowing low-cost and optimal asset utilization. In addition, Serverless development can handle the spiky traffic patterns and the global distribution of customers airlines face.

[1] Generating Value Through IT Agility and Business Scalability with AWS Serverless Platform - IDC 2018

Why Modernizations Fail

Avoiding the Big Bang

Why Modernizations Fail

Airlines need to modernize their technical foundations to react to emerging trends accelerated by the pandemic. A digital approach that can bring together the business and technical teams will enable the organization to experiment more and build faster, all while being more scalable and stable. This will allow airlines to address the most pressing trend, the need to personalize.

Modernizations typically involve an end-to-end refactor of the systems involved. The goal is to build a system that uses the best of the cloud, leverages a modern technology stack, and is constructed of independent systems that are separated from the constraints of legacy third-party services. To successfully decompose the monolithic structure to complete such a complex refactor migration, there must be a complete reconceptualization of the design of the domain and underlying systems. This is not be purely a technological change but rather a digital transformation of the sociotechnical organization.

Digital Transformation of the Sociotechnical Organization

Avoiding the "Big Bang"

Modernization projects that take a "big bang," all-or-nothing release strategy can sound more straightforward on paper - starting what feels like a greenfield project and rebuilding all the features of the existing system. The problem is this is a huge task. The organization has to wait for a complete rebuild of everything before any assumptions (technical and user experience based) are validated and until there is any return on investment. The blocker to releasing earlier is feature parity paralysis, needing all the features/functionalities of the existing system before any one modernized feature can go live.

Feature Parity Paralysis Example

	Existing	New
Booking	✓	✓
Payment	✓	✓
Email Notifications	✓	✗
User Accounts	✓	✓
Further Features	✓	✓

Release Blocked

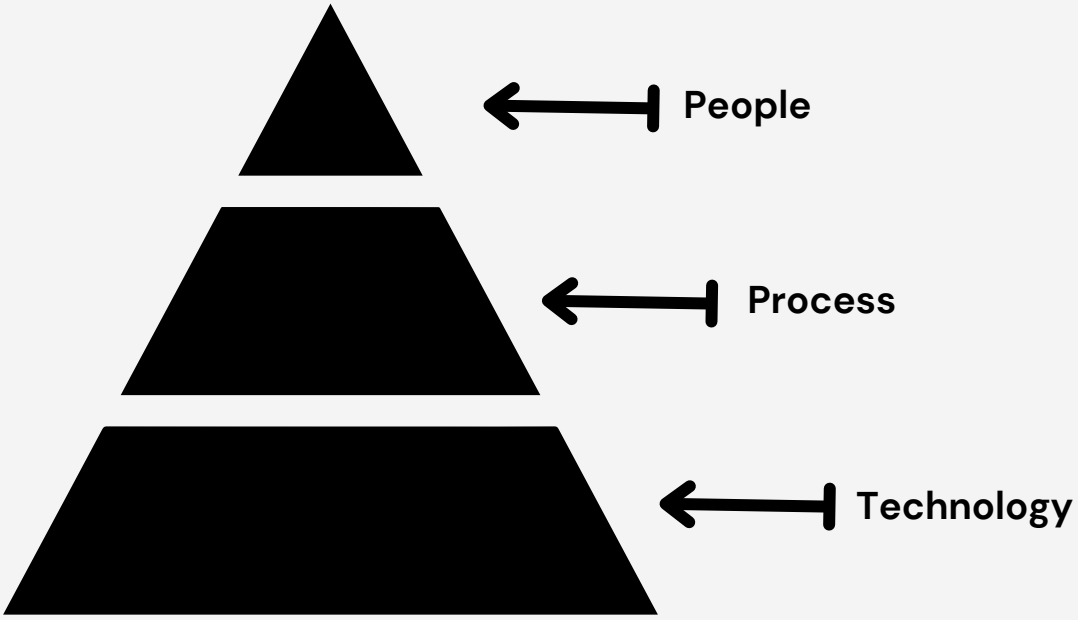
When held hostage to feature parity paralysis, organizations attempt the "big bang" approach. Due to the lack of feedback loop, validation of assumptions, and value to customers, this approach often fails. These big bang projects get started only to get stopped later due to a lack of business value during the modernization or an overall lack of focus as a result of the absence of continuous production releases. It's also impossible to bring the organizational changes needed if both systems are run as separate teams, as it becomes a never-ending race for the new system to catch up with the legacy.

Instead we need a way to circumvent feature parity paralysis and rebuild iteratively, pulling risk forward and testing assumptions as early as possible.

The People Problem

Any wide-ranging modernization project requires the buy-in of the people involved. Many organizations attempting a "big bang" approach silo the development of the new system into separate programs. This creates big 3 problems.

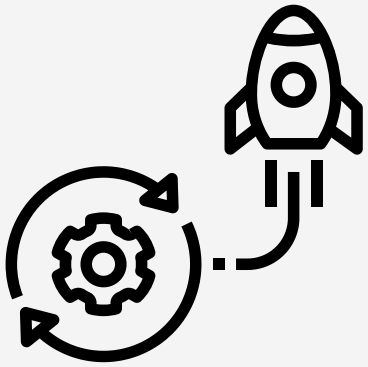
Firstly, the team building this new system, brought in for their expertise in the new technology, differs from the ones who built the existing system. As a result, they lack the institutional knowledge needed to work effectively. Secondly, some people are "left behind" to work on the legacy system and are not brought along on the journey. They don't get the opportunity to learn the new technology stack and will often perceive the new program as a threat. Finally, combining product and technology in the new team topology is critical for a successful migration. This key element of the project is only possible in the big bang approach once the new system is in production, which is too late.



Minimum Viable Migrations

A different approach

Minimum Viable Migrations



MVMs provide a progressive path to modernization, pulling risk and value forward

Companies have been adopting an Agile approach to software development for over a decade. The evident success of Agile means that waterfall methodologies have largely been left to the wayside and any new product will likely follow an Agile approach. However, when it comes to new migration projects, technology and business teams have reverted to the waterfall mindset of a "big bang" all-or-nothing approach.

The Minimum Viable Migration (MVM) Framework provides an alternative method - applying the successful practices of Agile and the Minimum Viable Product (MVP) to migrations of all types. This "stepping stone" approach massively reduces risk, provider value to customers faster, supports innovation, and brings a culture of continuous release.

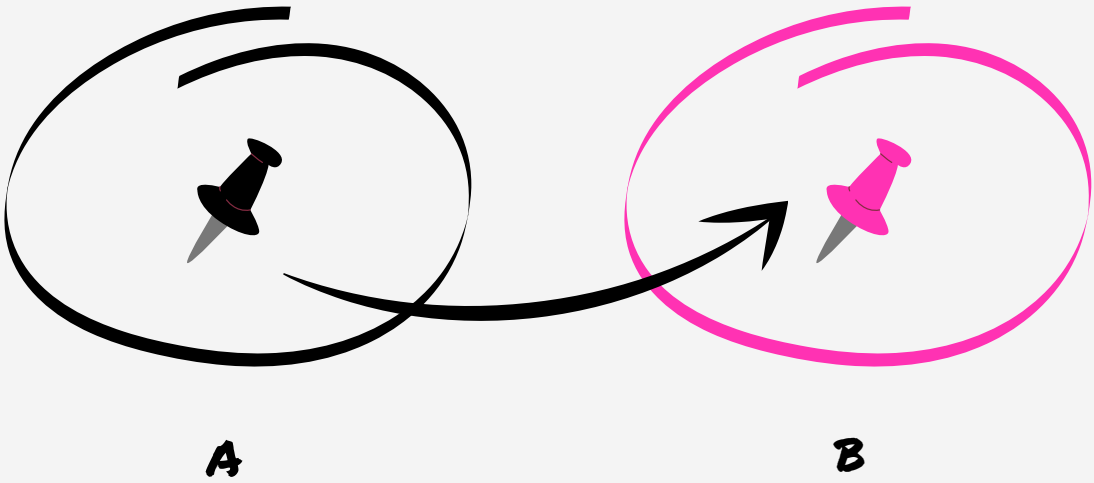
This framework can be applied to many and varied types of migrations - but has seen significant success in modernizing to state-of-the-art Cloud-Native architectures, namely "Serverless." Regardless of the kind of modernization, on-premise to Serverless or a continuation from a prior lift-and-shift to Serverless, the Event-Driven nature of modern Serverless architectures makes it an optimal migration target for the MVM approach.

Application Modernization is the process of moving an application to a more modern technology, be that a programming language, a framework, a paradigm or an infrastructure. More specifically, Cloud Migrations involve moving digital business assets and operations to a cloud provider.

Many airlines have now undergone Cloud Migration, with their applications "lift-and-shifted" to basic cloud containerization/virtualization technologies. To fully reduce the Total Cost of Ownership and realise the full agility the cloud provides, Modern Cloud Applications need to take advantage of the underlying cloud provider (e.g. AWS). Cloud Modernization is the process of moving digital business assets and operations to be fully Cloud-Native (i.e. Serverless), taking full advantage of the underlying Cloud Provider.

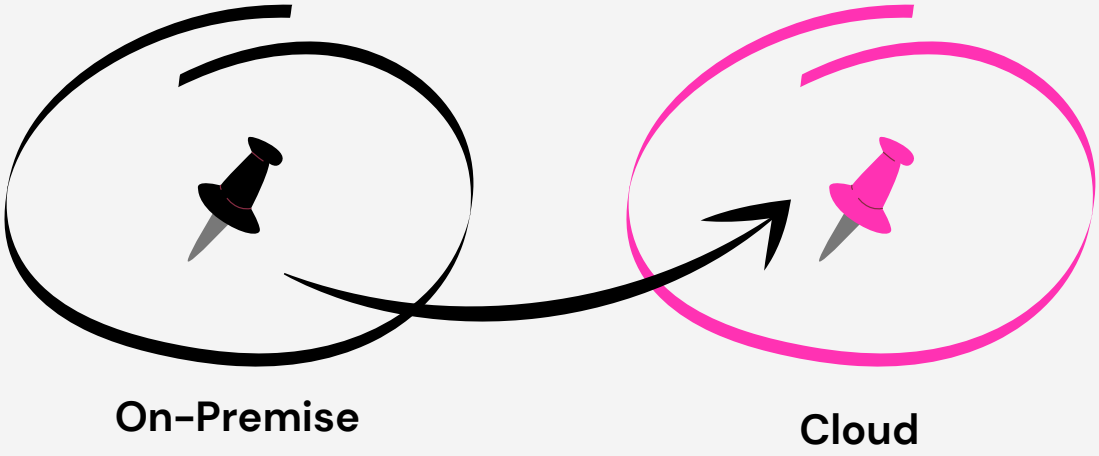
A migration of any type is simply a change in domain, from an "as-is" domain to some target "to-be" domain. A "big bang" approach to migration sees just one step between the two domains, with a "flick the switch" moment to change from the system as-is to the system to-be.

Generic Migration

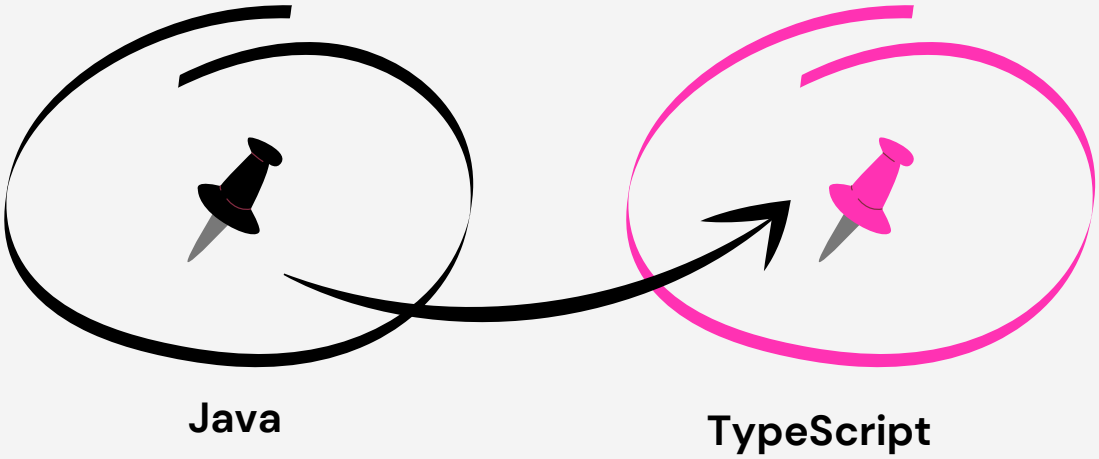


There are many different types of migration, with different "as-is" and "to-be" domains. The migration strategy is the key determination of the success of a migration - regardless of the destination.

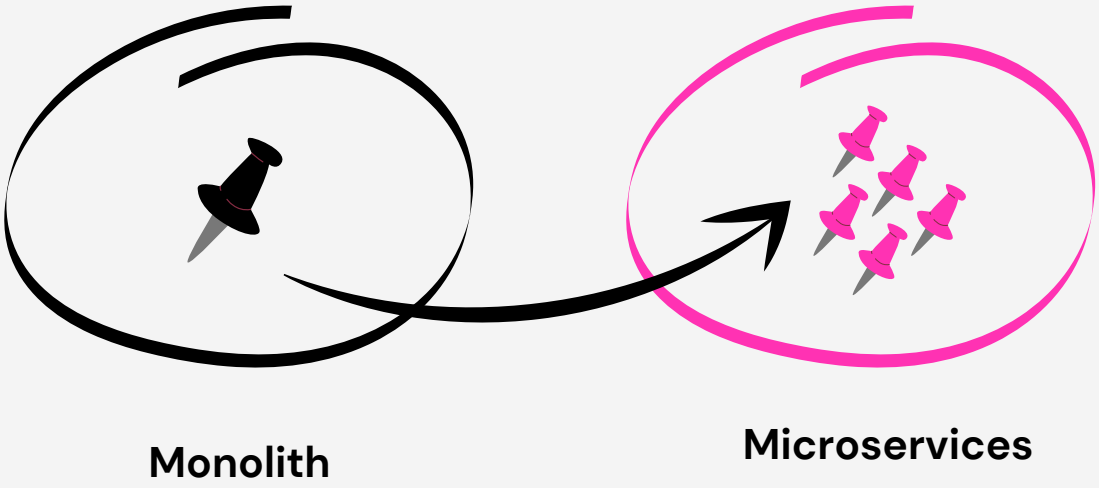
Traditional Cloud Migration

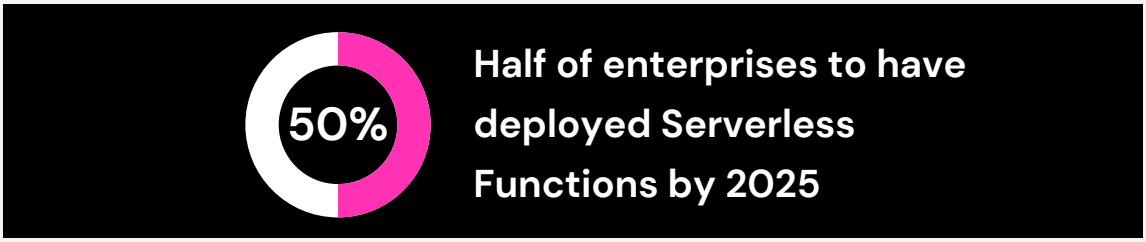


Language Migration



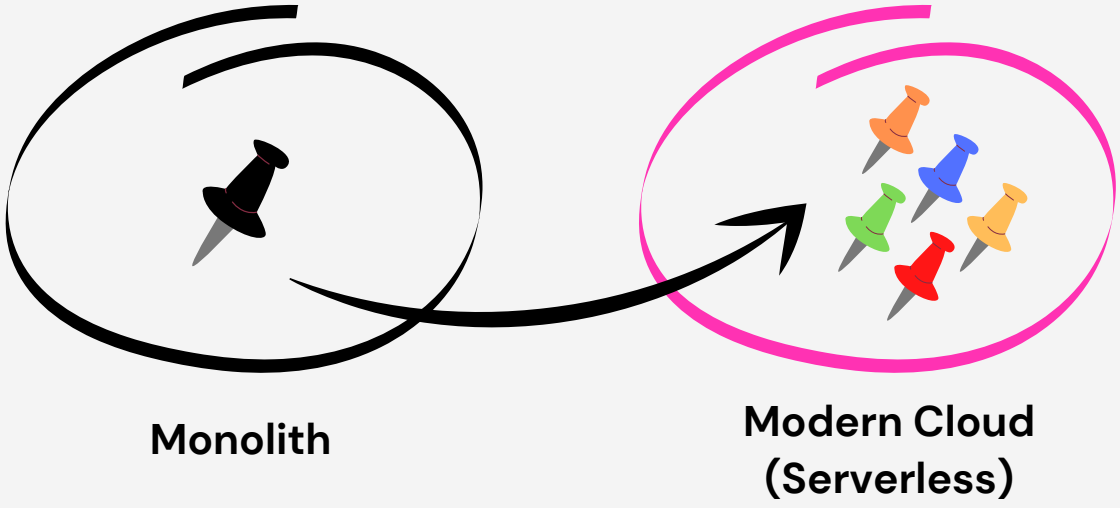
Structural Migration





As discussed, Serverless is emerging as the future of the Cloud — a range of services that allow you to build and run applications without having to think about servers. 50% of global enterprises are predicted to have deployed Serverless Functions by 2025 [2]. The value proposition of Serverless for airlines to recover and build back from the pandemic makes it the optimal target for modernization across the industry.

Cloud Modernization

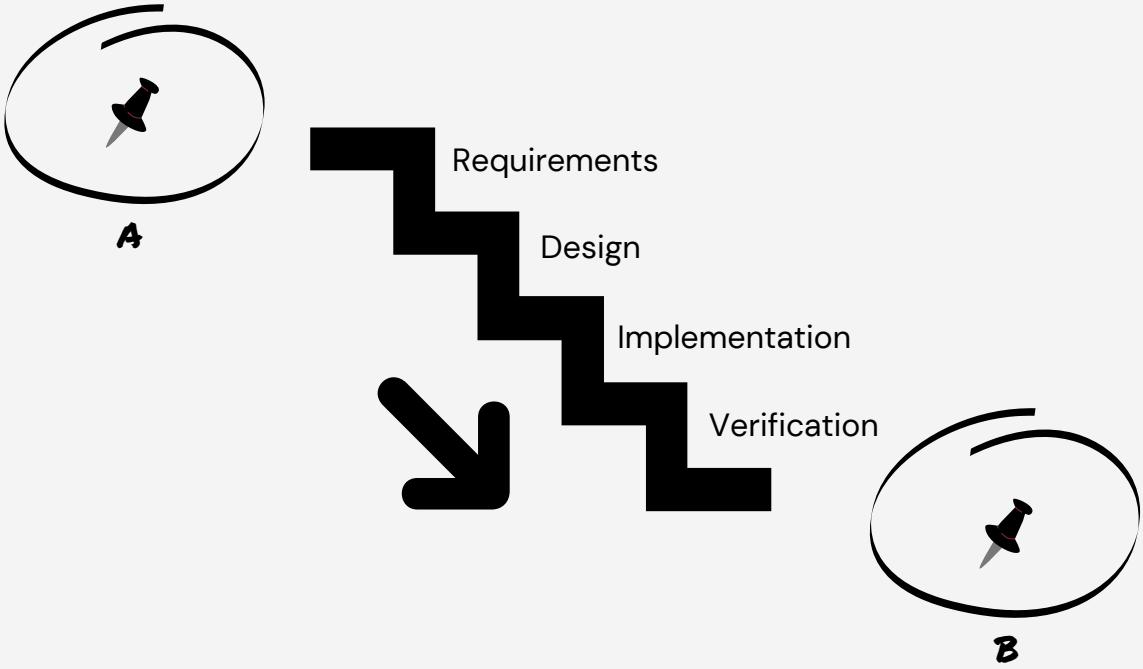


Airlines are looking to migrate their applications to Serverless through Cloud modernization Migrations. For the majority, this involves mastering a new set of technologies along with refactoring and restructuring their application to ensure they realize the full potential of the cloud.

Migration to Modern Cloud is not as simple as the initial lift-and-shift migrations many airlines have taken to the Cloud. Significant refactoring and rebuilding is required. The complex nature of this task means an Agile approach is needed to avoid pitfalls of the "big bang" all-or-nothing mindset.

[2] https://www.gartner.com/smarterwithgartner/the-cios-guide-to-serverless-computing

Avoid Waterfall Cloud Modernization Migrations

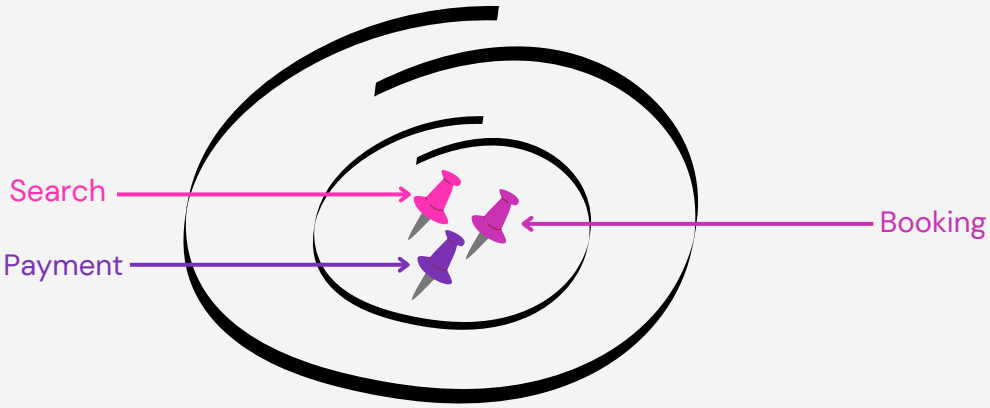


Traditional waterfall migration projects (e.g., a lift-and-shift from on-premise to cloud) typically view the transition as a single step from a "system-as-is" to a "system-to-be". This is only appropriate for more straightforward migrations, where less refactoring and rebuilding is needed to move between domains.

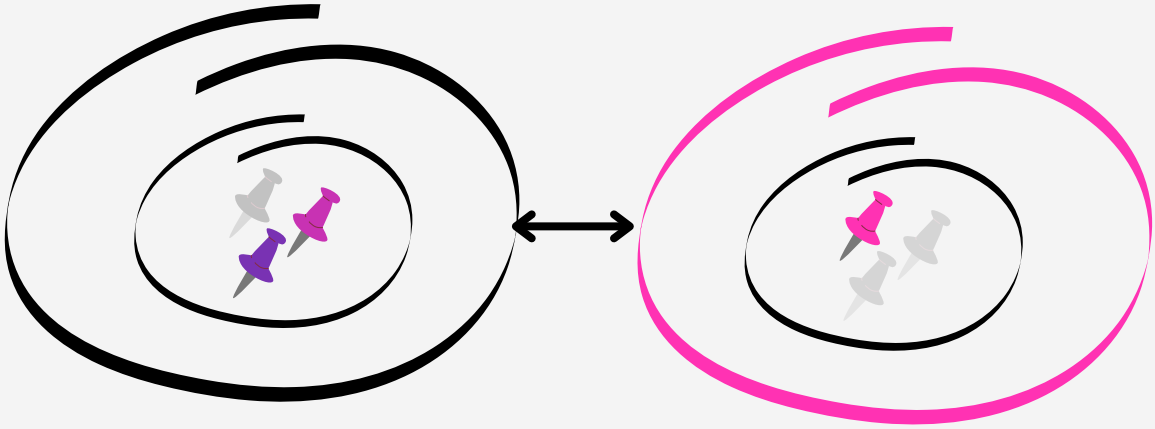
For Cloud Modernizations, when the aim is to realize the value of the cloud, there is too large a gap for a single-step "big bang" approach. A more progressive method is the best way to ensure risk is reduced, assumptions are tested, user value is delivered faster, and a culture of continuous releases is facilitated. This progressive approach is not a single step but rather a stepping-stone path of small, individually migratable chunks.

To bring agility to such migrations, there needs to be a set of progressively evolving points between the system "as-is" and the system "to be." To do this, the application(s) need to be broken into composite features and migrated as subsets of the overall feature. In other words, we need to break the "feature parity paralysis".

The Application Composite Feature Sets



Two Systems in Parallel

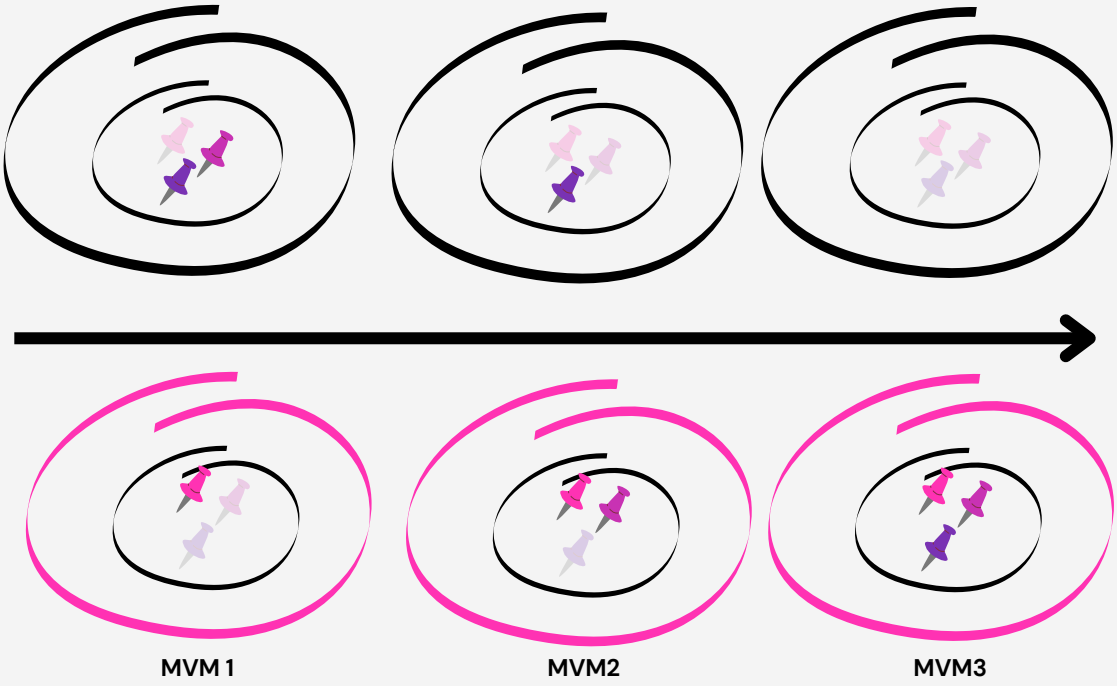


As illustrated above, we can see 2 systems running at the same time. For instance, the existing system (A) is serving 2 out of 3 features and the new system (B) is serving 1 out of the 3 features. This results in the holistic user experience being served by the union of the 2 systems.

Due to this separation, we can go live with the new system (B) in the new domain (i.e. Serverless), while continuing to run our legacy system (A) in the old domain (e.g. an on-premise monolith). A feature like "Booking" can be implemented in the new system while leaving "Check-In" and "Payment" in the legacy system. In this apporach, we can break down the entire system into a series of independently migratable stepping stones until the legacy system has been completely replaced. This is the concept of Minimum Viable Migrations.

A Minimum Viable Migration (MVM) allows you to migrate a portion of your application from one domain/paradigm to another with a subset features to be used by some or all customers. The framework is a step-by-step guide to adopting a progressive approach to migration and modernization. Its design identifies candidate MVMs that can then be prioritised and injected into the existing customer experience, protecting the new system's design from the old constraints.

A Minimum Viable Migration (MVM) creates a usable migrated version of an application to a new domain/paradigm with a subset features to be used by some or all customers.

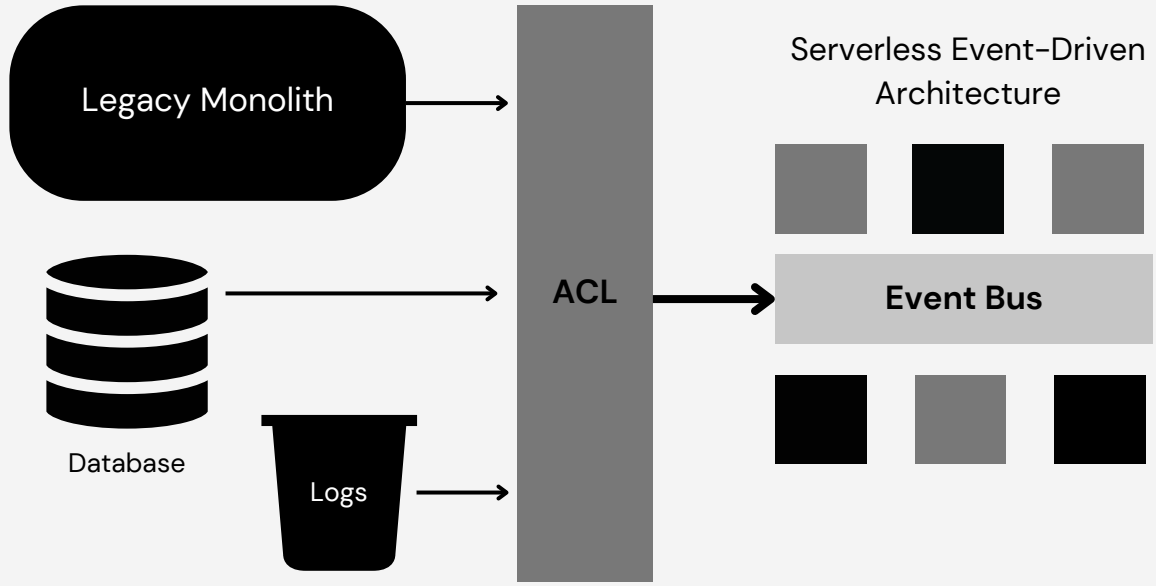


For an MVM approach to succeed there needs to be a communication mechanism between the legacy and new systems. This mechanism must be asynchronous in nature and isolate the new system from the technical debt and constraints of the legacy system. Typically the most effective mechanism is an Event Bus. This is channel through which events can be exchanged. Conveniently state-of-the-art modern cloud (i.e. Serverless) application are by their nature event-driven. This means that the decomposed set of services in the new system are already reacting to a predefined set of events. How these new events are designed and how the legacy system is able to trigger them is a key design decision to ensure success in an MVM approach.

Event Design and Eventification
Success with Serverless at scale requires an embrace of Event-Driven Architecture. The best way to design this new Serverless architecture is via Domain-Driven Design and EventStorming workshops. Through these workshops, an Event Schema for the new system can be designed with an accurate representation of the underlying business domain. This event schema becomes the contract between the services in the new architecture.

For the new system to react appropriately to changes in the legacy system it's critical to trigger events from the old to the new - "Eventification". To achieve this, various techniques are available, ranging from manual code triggers to database change data capture and log tapping. All of these will result in events tainted by the design/constraints of the legacy system.

An "Anti-Corruption Layer" (ACL) is needed to transform, aggregate, or split events from the legacy system to match the clean event schema of the new world. This ACL should be built within the domain of the new technology and act as a clear interface separating the new system and its teams from the complexity of the legacy world.



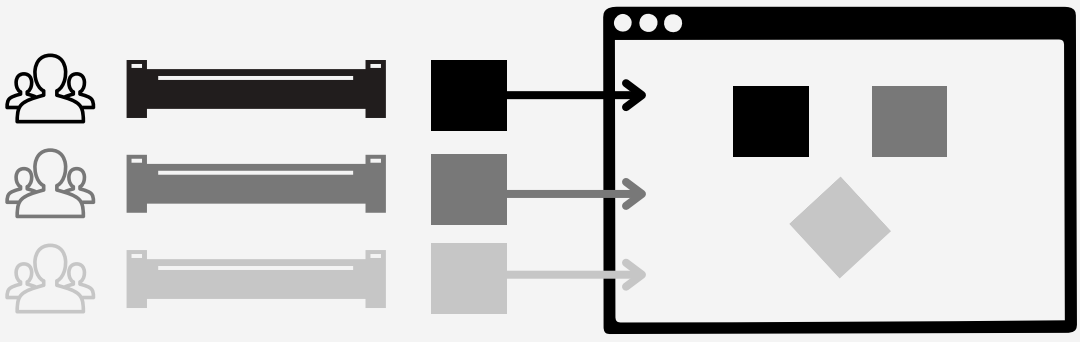
Injection Approaches

When distributing the customer experience across multiple systems, there needs to be a mechanism to inject the new experience into the old while maintaining the ability to switch back and forth dynamically. There are several possible injection strategies, the combination of which will be used on an MVM by MVM basis.

Backend-only systems are the simplest to inject into as it can be handled via forwarded requests or a reverse proxy sitting in-front of the two systems. When it comes to the areas of highest experimentation potential for the customer experience a more vertical slice approach to migration is needed. Taking vertical slices of the end-to-end frontend to backend experience and distributing them between multiple systems requires more advanced injection techniques.

Microfrontends and Edge Routing

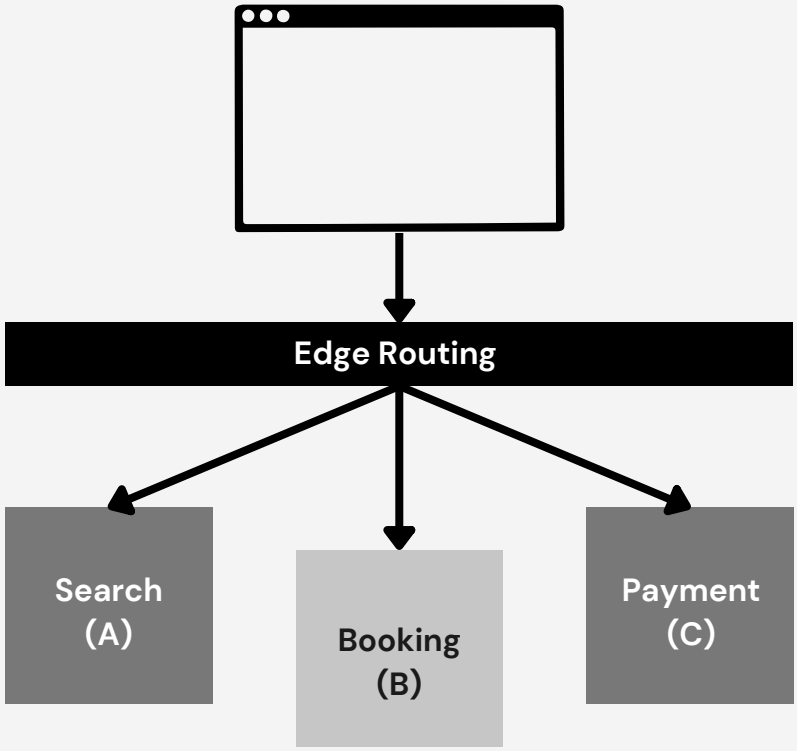
The new microfrontend approach to applications brings the concept of microservices to the frontend. In this way a large application is split up into independent modules that can be built and released completely independently. This is becoming a common pattern for modern airlines as it allows for the independence of product-lead teams at scale to achieve high deployment frequency and independent experimentation.



For instance, Search can be built and shipped independently to Payment, and so on. Not only do microfrontends provide a useful target for airline e-commerce, they are also an injection mechanism for an MVM modernization themselves. The type of microfrontend strategy adopted during and after the modernization has a large impact on the customer experience and granularity of MVMs possible.

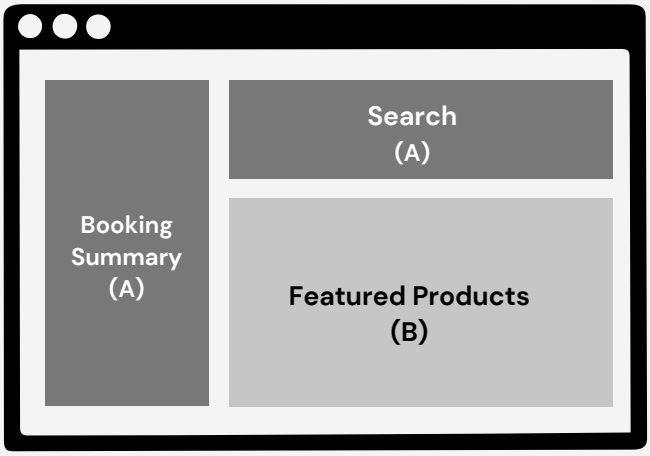
Horizontal vs Vertical Approaches

There are two distinct types of Microfrontend strategy, vertical and horizontal. In a vertical approach, the microfrontends are entire pages, and as such users switch between microfrontends page by page. This is the simplest type to inject into a legacy system as this can be handled via Edge redirection. In this approach an Edge compute service operating in the Content Distribution Layer (CDN) is able to detect if a given page should be served by the legacy system (A) or the new system (B).



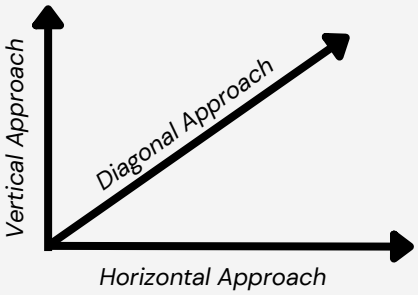
As there is a need to have product-lead teams building and deploying independent this approach as a target state only works when the subdomains of the site are split clearly page by page. This is not typically the case for airlines and e-commerce as different functions are mixed together to provide a holistic customer experience. For instance a search bar may be presented above marketing content on the landing page. This might then lead to a booking flow interspersed with ancillary upsell and accompanied by a cross journey basket. This is not only an issue for the end state system but also for the modernization strategy. Forcing page level migrations increases the number of areas of the system that have to go-live together, increasing the size of MVMs and reducing the opportunity to pull risk forward, deliver value faster and create a culture of continuous delivery.

The alternative microfrontend strategy is the horizontal one. In such an approach individual components, fragments of pages, are built and deployed independantly before finally being pulled together on a page to build the aggregate user experience.



A vertical microfrontend approach not only brings flexibility to the end state system, but allows for a much finer granularity of MVMs. Such an approach requires a greater investment in building a microfrontend orchestration platform, and requires optimization to achieve good performance and state management protection.

A Diagonal Strategy



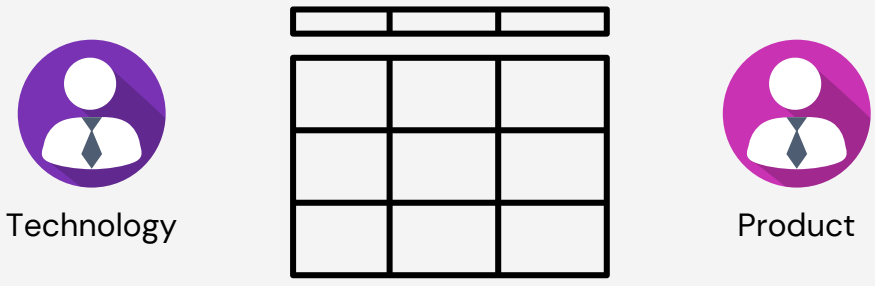
In practice a combination of the 2 is needed, with the precise injection strategy determined on an MVM by MVM basis. In such an approach pages such as "Search" can be shipped initially as entire vertical microfrontends while the "Booking Flow" components are broken down and require the surrounding orchestration to be shipped to production.

Investment in a Microfrontend Orchestrator is not only useful for the modernization and final state team independence. It provides a foundation for large scale multi-variant A/B testing that enables greater experimentation as well as a route to Personalization.

MVM Priority Matrix

A key step to ensure the correct phasing of modernization, resolving of dependencies, complementary injection strategies and overall modernization governance is clear visibility over the MVMs needed to deliver the aggregate customer experience. This requires an analysis of the aggregate experience and validation against existing code, analytics data and QA suites / manuals. From this the technical team can identify the candidate MVMs, the service they relate to, injection strategy and any scope limitations needed.

As a rule of thumb an MVM should take no more than 8 weeks. This ensures a deployment culture can be built and planning can accurately provide visibility. For some feature sets is a natural constraint, for others it can be a challenge. Payment for instant can be hard to break down in to individual MVMs shippable in such a timeframe. In these cases applying scope limitations against each MVM is needed. For instance, with Payments a typical limitation can be on locale/market. In this way individual payment providers can be added until the federation layer supports all methods globally.



The living embodiment of this set of MVMs, their domain, scope limitation and injection strategy is captured in the MVM Priority Matrix. This matrix is in priority order and is a living document that tracks what has gone live, anticipated dates for the next candidate MVMs and is crucially co-owned by the technology and product organisations.

Team Topology and Governance

While a wider topic outside of the scope of this whitepaper, the correct structure of teams and governance practices during and after the modernization are key for technical and organisational success.

About **aleios**

aleios is the Cloud-Native Serverless speciality of the Theodo Group, working with customers to build state-of-the-art platforms that truly take full advantage of the Cloud. Building highly scalable systems we connect people, liberate data and create innovation through technology.

Theodo Group is an ecosystem of technology experts launched from that support organisations through digital enablement by bringing together all the skills needed under a common methodology, culture and commercial model.

Theodo Group has > 800 digital specialists across London, Paris and New York - with specialist verticals in the Group focused on Cloud, Web, Mobile, UI/UX, Data Science, FinTech, Aviation and Health Tech.

Authors

Author: **Ben Ellerby**
CTO, aleios

Editing: **Elle Futernick**
Chief of Staff, aleios

https://www.aleios.com
+44 204 5716 968

V1