

Building a Composable Future

The evolution of digital transformation and why a composable approach is crucial for your business technology strategy.



Contents

Part 1 – How did we get here?	4
Part 2 – Composability as a philosophy	8
2.1 Composability in Development	10
2.2 Composability in Design	21
2.3 Composability in Strategy	26
Part 3 – Where to next?	33

Introduction

Breaking down the evolution of digital and why **a composable philosophy** should impact the way you think, plan, design and build your business technology.

The digital landscape is in a constant state of evolution. Since Inlight's inception in 2008, the purpose and function of the web, mobile apps and other digital assets has shifted dramatically. The job for digital managers and service providers has never been more complex, making it an important time to look back and reflect on where we have come from (both individually and collectively), unpack the challenges and obstacles faced in recent times, and take the opportunity to explore what the future of digital might look like.

The objective of this document serves a dual purpose:

1

As Inlight marks its 15th anniversary, it is an opportune moment to consolidate our learnings into a tangible resource for future endeavours.

In a services-based industry, finding the time for introspection is never easy, and our commitment first and foremost will always be to our team and clients. But now, as much as ever, it's crucial that we take stock of where we have come from so that we might understand where we are heading.

2

Articulate a guiding philosophy for constructing digital products that maximises the chance of long-term success for our clients and partners.

The digital landscape is ever expanding and non-linear, resulting in countless ways to approach the same problems. While there is no universal solution, our experience over the past 15 years has led us to a recurring theme: the utility of 'composable thinking.'

This paper aims to explore the present-day challenges facing digital leaders and to demonstrate how composable thinking can offer more effective and sustainable solutions.

PART 1

How did we get here?

An exploration of the **digital journey**: from the early days of static, inflexible systems to the dynamic, adaptable architectures of the present.

Here, we understand the shifts that have driven businesses to seek more agile, responsive solutions and the role of composability in meeting modern demands.



The trajectory of the World Wide Web over the last 30 years has been nothing short of transformative. Originating as Web 1.0 — a static information hub — the internet has undergone **continual metamorphosis**, evolving into today's multifaceted digital landscape.

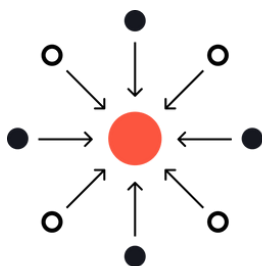
What began as a rudimentary platform for content dissemination has grown into an intricate, functional medium that enriches our personal and professional lives in countless ways. Consider, for instance, the leap from Web 1.0 to Web 2.0 around 2004. The latter ushered in an era of collaboration and user-generated content. This shift wasn't merely incremental; it represented a seismic change in how users interact with the digital ecosystem.

From the advent of social media to the proliferation of mobile, e-commerce and cloud computing, the web has expanded its utility exponentially. Businesses, once content with online 'brochureware,' now rely heavily on a broad array of digital tools to remain competitive in a global market.

Why is this crucial to understand?

The internet and digital ecosystems have become exponentially more complex. As we advance towards Web 3.0—a semantic, intelligent web—some of the tools and processes we once leaned on to streamline our digital endeavours have become antiquated and bloated, limiting the pace and potential of what can be achieved. This mounting complexity demands a rethink, steering us towards breaking digital products into more manageable, efficient components.

As the utility and function of the web exploded over the last two decades, we witnessed the surge of platforms adept at consolidating diverse needs into singular 'monolith' solutions. This encompassed tools like Content Management Systems (CMS), E-commerce platforms, Customer Relationship Management (CRM) systems, and Enterprise Resource Planning (ERP) solutions. The monolithic architecture wasn't merely a platform preference but also a dominant design pattern for building complex digital products.



WEB 1.0



WEB 2.0



WEB 3.0

This approach of aggregating various functions into one unified system had distinct advantages at the time:

- Ease of development and deployment.
- Speed and efficiency within small teams.
- Easier to test and debug.
- Simplicity of maintenance.

Though monolithic systems initially provided many benefits, they soon showed significant limitations as their scale increased. Features that originally streamlined operations turned into sources of inefficiency and rigidity. Multi-faceted products became expensive and hard to use and often failed to deliver their promised value. As time went on and systems continued to evolve, more and more problems arose with these types of systems, including:

- Difficulties and cost in scaling.
- Tight coupling of functionality meant a failure in one component often lead to entire system failures.
- Minor changes had wide-reaching implications that slowed development and evolution cycles.
- Innovation and progress were slowed.
- Platform lock-in, which lead to to high-risk dependencies and loss of organisational autonomy.

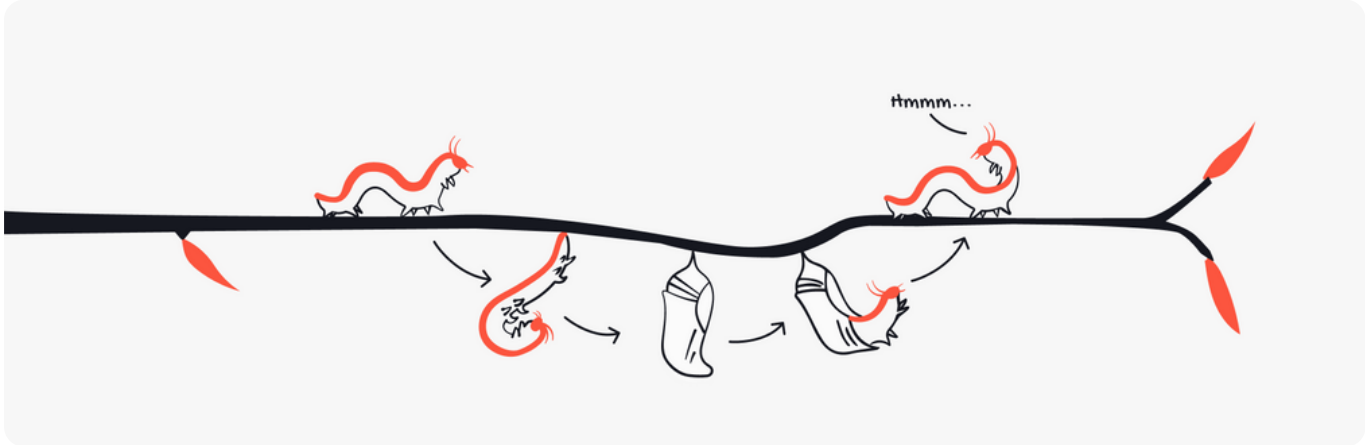
One poignant analogy is that of a large ship; the bigger it gets, the harder it is to change course. This holds true for both digital systems and organisational structures. Operational complexity grows, and increasingly, more resources are funnelled into mere sustenance rather than value creation.



In light of these challenges, the concept of 'Digital Transformation' gained traction in the enterprise space. Broadly, Digital Transformation is the process by which companies employ a new or upgraded technology across their business to drive fundamental change.

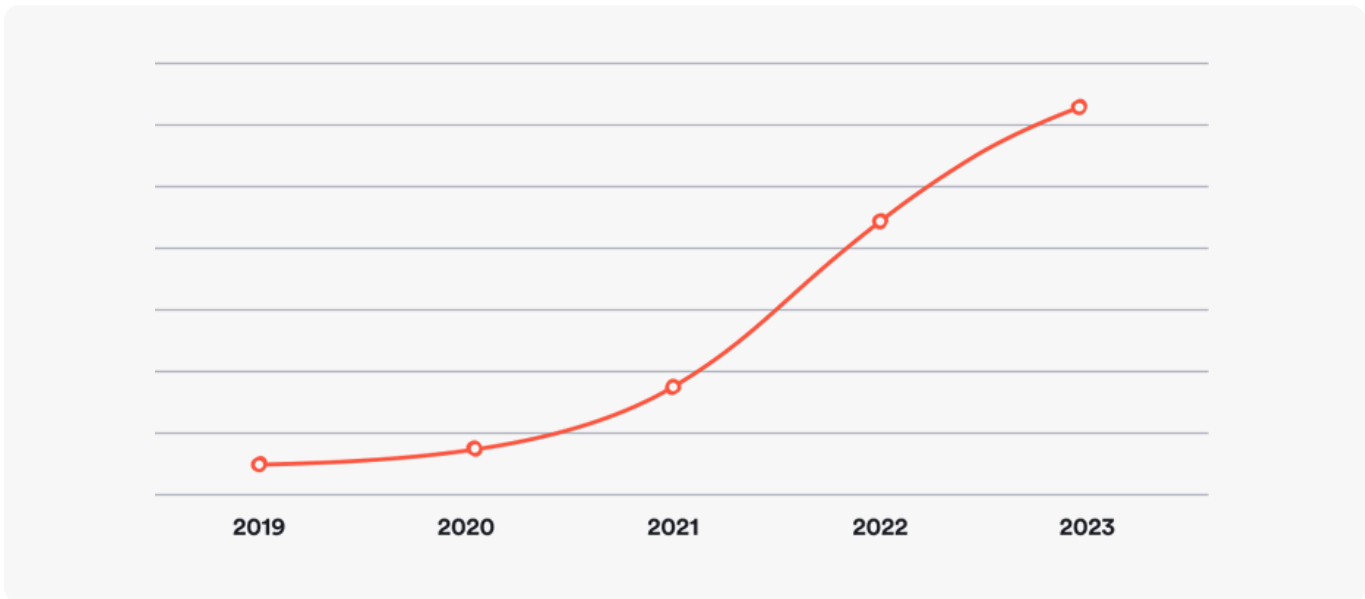
Due to the size and complexity of these technologies, these initiatives often turned into sprawling, multi-year projects that seldom delivered on their promises. A report by Boston Consulting Group *'The Success Formula for Australian Companies to Digitally Transform' (2021)* substantiates this point: only 33% of Australian companies succeed in their digital transformations, while 47% create some value, and a concerning 20% generate limited value. These figures underscore the daunting complexities and pitfalls associated with large-scale transformation projects. Among the challenges:

- **Shifting team composition:** The long timelines inherently involved team member turnover, leading to a loss of accumulated knowledge.
- **Rapid technological changes:** What was cutting-edge at the project's inception often became obsolete or sub-standard by its conclusion.
- **Resource allocation:** The focus on transformation often left existing systems in a state of neglect, introducing risks to customer satisfaction and platform security.



Many years ago, Inlight gave a presentation entitled **'Where are all the butterflies?'** which discussed the challenges digital transformations face when companies enter a cocoon for several years, hoping to emerge as a beautiful butterfly. Sadly, what often happens is that their customers are neglected for many years, and the caterpillar never transforms into the butterfly that was promised.

So, what's a solution to these compounding complexities? The principle is surprisingly simple: decomposition. Whether we're talking about teams, platforms, products, or strategies, breaking systems into modular, autonomous components can yield greater agility, speed, and specialisation. This paradigm is known as **'composability'** and its application extends beyond product development to every facet of our work.



Search interest for “Composable Architecture” is increasing over time (Google Trends)

PART 2

Composability as a philosophy

Delving into the core principles of composability, this section uncovers its relevance not just in technology, but as a holistic approach that impacts business strategies, team dynamics, and overall organisational design. We explore **why this mindset is crucial for success** in today's ever-evolving digital landscape.



Composability as insurance against change

Anyone who has worked in the digital industry over the past decade has likely heard the term ‘composable architecture’. The simplest way to describe this term is the idea of building software from small, independent components that can be combined to form a complete system. Typically, these components are connected via a common interface language or API (Application Programming Interface) and can be seamlessly integrated to build more complex solutions.

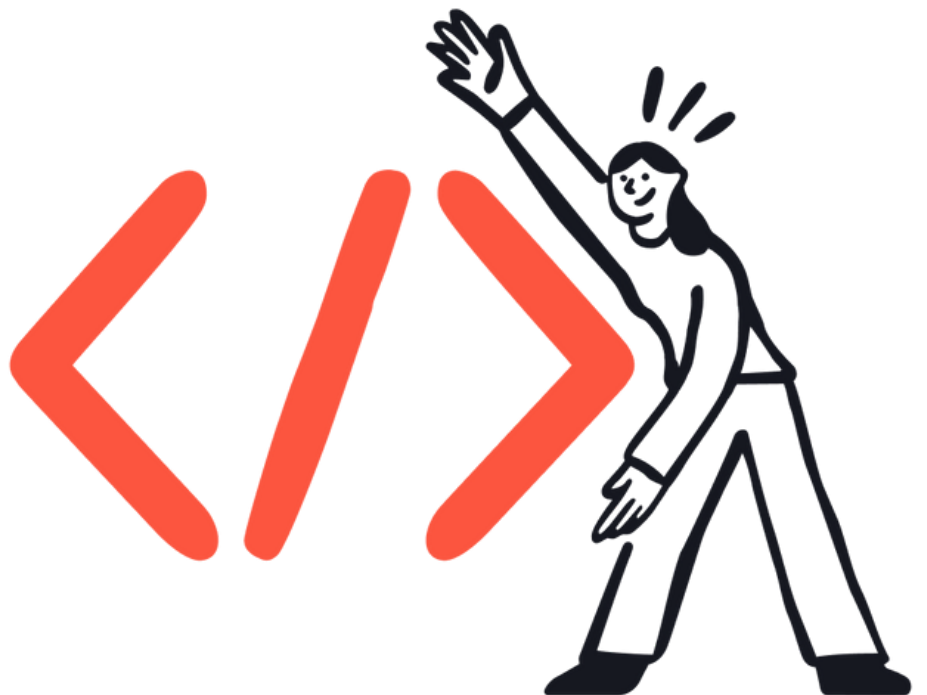
A physical analogy that relates perfectly to this idea is LEGO. Whilst each LEGO brick provides a unique form or function, the universal connection pattern allows a collection of components to be built into complex machines or structures.

Whilst this pattern is well known in the software industry, the principle can be applied to any complex system. Later, we’ll explore how this philosophy can be leveraged across various parts of a business and their digital products. In each case, we’ll look at some of the benefits and considerations.

Broadly speaking, composability can be seen as insurance against change. It provides a way of **building agility and resilience** into the foundation of any system.

PART 2.1

Composability in Development



Introducing MACH

When planning your next digital product, understanding certain development patterns aligned with composability is key. One such pattern is the MACH architecture, popularised by the MACH Alliance.

The MACH Alliance is a not-for-profit advocacy group whose members include software vendors, systems integrators, agencies, and individual experts, who advocate for open and best-of-breed enterprise technology ecosystems.

MACH stands for four core principles in product development: Microservices, API-first, Cloud-native, and Headless. These principles steer from traditional monolithic design towards a more modular and composable approach. Whilst these principles do not mandate an ‘all-in’ approach, used together they provide the best chance of long-term success for your products.

MACH principles:

Microservices	Distinct units of business functionality developed, deployed, and managed independently.
API-first	All functionality is available through APIs.
Cloud-native	Software that utilises the cloud for more than just storage; for example elastic scaling and automatic updates.
Headless	The presentation layer is decoupled from back-end logic, providing flexibility in how your content is delivered to users.

Benefits of implementing a MACH-based architecture:

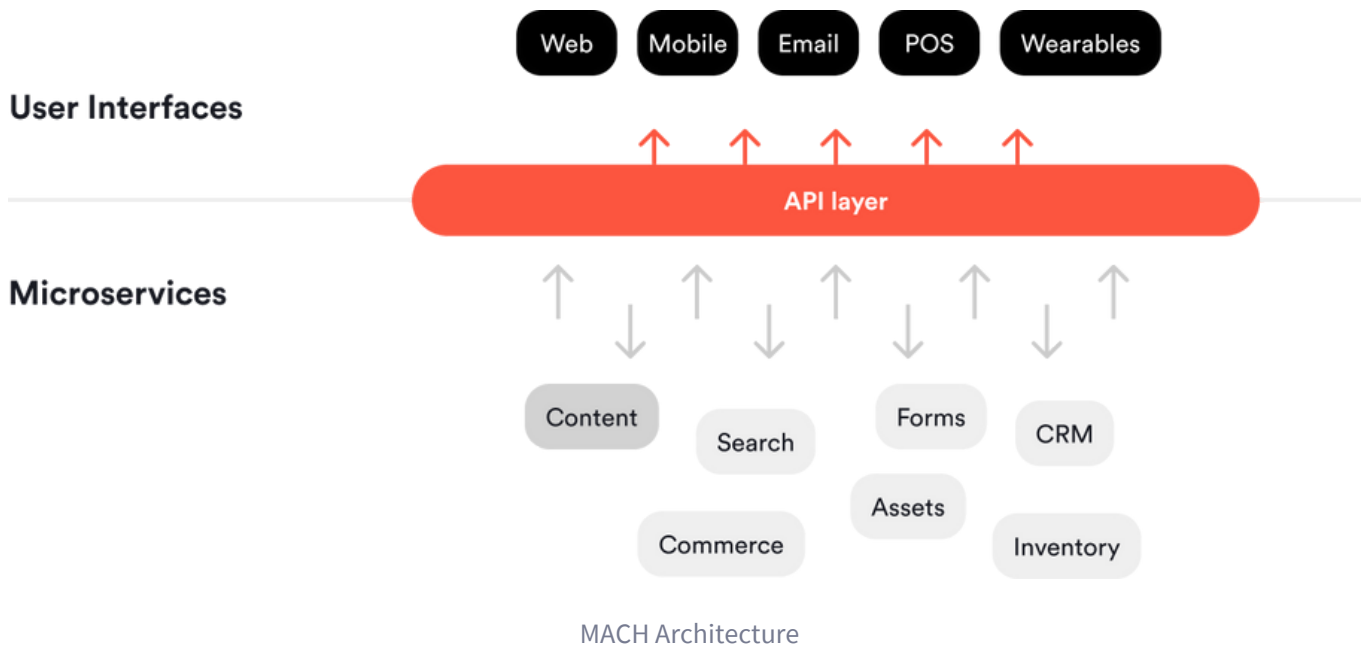
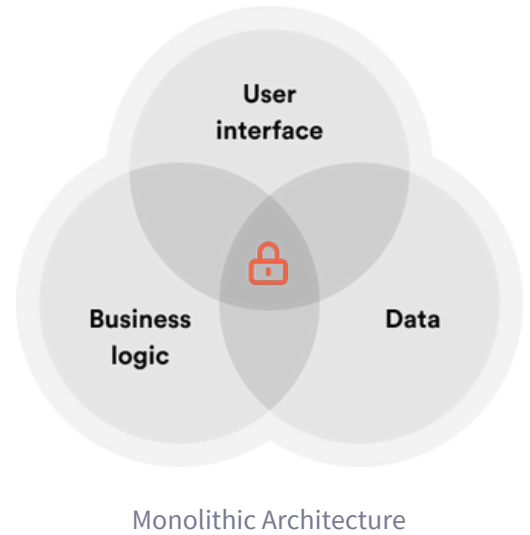
Customer centric	The decoupling of front-end and back-end allows for customer-focused experiences.
Best in breed technology	MACH provides freedom from being locked into any specific technology stack.
Extensability	The architecture supports easy add-ons of features or new integrated solutions over time.
Scalability and performance	Slow monolith solutions are replaced with infinitely scalable, high-performance cloud-based services.
Enhanced security	Service decoupling limits the fallout of any single component's failure.
Cost-efficiency	Reduced licensing costs and efficient cloud hosting equate to both immediate and long-term savings.
Evergreen	Cloud-native architecture supports automatic, immediate updates.
Future-proof	Intrinsically compatible with future growth and scalability through the freedom to swap out, re-build, and improve components over time.

To put MACH architecture into a relatable context, consider the rise of headless Content Management Systems (CMS).

Traditional CMSs tightly couple the content and business logic layer (back-end), which limits the ability to serve the same content to multiple digital products or assets. It can also make integrating content or functionality from other sources outside the chosen CMS challenging.

A shift to MACH means the CMS is headless (no inherent presentation layer). This means content is served to the front-end via an API, the CMS itself is often cloud native and/or fully managed, and additional microservices can be used to provide specific functions to the website or app.

A headless architecture also offers the flexibility to serve content to a single or multiple experiences, and leverage supporting services where required.



To illustrate the practical advantages of adopting MACH principles, particularly a headless CMS, let's look at some real-world applications.

Industry leaders including Adidas, Airbnb, and The New York Times have migrated to a headless CMS architecture. Similarly, closer to home, JB HiFi, Atlassian, Kmart, Officeworks, and Scentre Group (Westfield), along with our own clients Powershop, Nando's, Vicinity Centres, and Transport Accident Commission (TAC), have all experienced significant improvements after migrating to a headless CMS architecture.

Global software pioneer, Atlassian, utilises a headless CMS to centralise content for a myriad of external support services: everything from help articles and technical documentation, to in-product support and resources for prospective customers. The impact on speed to market has been transformative. Before migrating to a headless CMS, it took Atlassian two weeks to publish content; now, that timeframe has been slashed to just 5 minutes (contentful.com/case-studies/atlassian/).

What are some leading headless CMS platforms?

If this approach sounds appealing, you might wonder which headless CMS platforms you should consider. Following are some that we use regularly with great effect when building products for our clients.

Each platform has its strengths and weaknesses, and the choice will ultimately depend on your organisation's unique requirements. Therefore, it's crucial to conduct a thorough evaluation to determine the best fit for your business's individual needs and workflows.



Introducing JAMstack

Unlike MACH, which is a comprehensive strategy for both the front-end and back-end, JAMstack is predominantly focused on the front-end or client-side/user interface.

The term is an acronym for JavaScript, APIs, and Markup. It employs pre-rendered static pages served through a Content Delivery Network (CDN), with JavaScript handling the dynamic aspects via APIs.

The benefits of JAMstack overlap significantly with those of MACH:

Security	With a simplified hosting infrastructure and pre-rendered, read-only pages, JAMstack provides robust security measures against attacks.
Scale	Caching and CDN support make it exceptionally good at managing high traffic and avoiding back-end bottlenecks.
Performance	Light-weight API requests and client-side processing mean quicker page load times, improving the user experience, conversion rates, and SEO rankings.
Maintainability	CDN's reliability and the static nature of the pages result in minimal maintenance.
Portability	JAMstack's architecture is remarkably versatile, making it easy to migrate or upgrade with little hassle.
Developer experience	JAMstack allows for a wide variety of tools and techniques that do not lock developers into proprietary technologies or frameworks.
Evergreen	Cloud-native architecture supports automatic, immediate updates.
Future-proof	Intrinsically compatible with future growth and scalability through the freedom to swap out and improve components over time.

Another important but more intangible benefit is the **ease of attracting and retaining top-tier developers.**

As the demand for software developers continues to grow, aligning your architecture with sought-after technologies can be a sound strategic move for long-term success.





Mini case study

One local success story praising the benefits of composable architecture is **ANZ Banking Group**.

ANZ has embarked on an evolutionary journey, rearchitecting its "monolithic" applications to be composed of microservices and API-based services. The shift has led to improved stability, reliability, and speed of new feature delivery.

Key takeaways include:

Improved stability and availability	Before, when a part of their monolithic system went down, it took the entire application. Now, issues can be isolated to specific components, ensuring that other functionalities remain operational.
Increased cadence in feature delivery	The shift allows multiple teams to work on multiple services in parallel, leading to quicker deployments of new features.
Scalability and learning from user behaviour	With real-time data coming in, ANZ is continually refining how services are broken down to optimise performance and customer experience.
Future potential	ANZ foresees additional advantages once more microservices are migrated to cloud-based platforms.

“Now that we’ve moved to a microservices architecture, we’ve got the ability to have multiple squads working on multiple services in parallel. We’re really seeing a big uptake in our feature delivery by moving to this new architecture.”

Peter Tsatsaronis

Product Area Lead for Digital Channels in Institutional Technology,
ANZ (IT News, 2023)

Composable user interfaces (UI)

The principle of composability also extends to the user interface, revolutionised in part by Meta's front-end JavaScript library, **React.JS**.

React enables the construction of interfaces from reusable components. This modular approach ensures a more cohesive user experience, accelerates development, and improves performance. In addition to React.JS for web apps, its sibling, React Native, offers similar benefits for mobile app development. Popular alternatives such as Angular, Vue.js, and Ember.js offer similar frameworks depending on your team's preferences and experience.

Frameworks:

React	A JavaScript library for building user interfaces, particularly known for its component-based architecture and virtual Document Object Model (DOM) and often used for web applications.
React Native	An extension of React that allows you to build native mobile applications for iOS and Android using JavaScript and React.
Angular JS	A TypeScript-based open-source framework for building web applications, developed by Google. Features a rich set of tools and a powerful ecosystem.
Vue.js	A progressive JavaScript framework for building user interfaces. It is incrementally adoptable and designed to be simple yet powerful.
Ember.js	An opinionated JavaScript framework for building ambitious web applications. Provides a set of conventions and best practices out of the box.

In addition to these frameworks, newer tools such as Tailwind UI are making user interface (UI) development even more efficient and composable by provisioning ready-made, expertly designed components provided in React and Vue and powered by headless UI.

Considerations for composable development

Like any other technology or approach, composable architecture is not always the answer to every problem.

There are several considerations or costs for choosing these paths, and there are things to be aware of before embarking on a composable re-design:

Key considerations:

Change management

Moving to a composable architecture requires a significant shift in mindset in how to develop, maintain and engage with your digital assets. It can take time to upskill your development team and other stakeholders that interact with these digital assets.

Complexity

While composability can significantly reduce complexity in individual systems and functions, it can lead to a more complex digital ecosystem overall. Choosing and integrating best-in-class technologies that meet your specific needs also means understanding, maintaining and evolving more systems and technologies across your entire digital portfolio.

Implementation

Individual monolith platforms often provide out-of-the-box, ready-to-use capability that requires minimal setup or configuration. Implementing a custom set of integrated solutions can sometimes increase the initial effort before the benefits of composability are realised.

Teaming and capability

Implementing a composable architecture may mean understanding a more diverse set of development languages and solutions. This may require a shift in internal teaming to ensure team members have adequate skills to support these individual components, which may prove prohibitive for smaller organisations with limited resources.

Future considerations – adaptability through composability

While the shift towards a composable architecture is significant on its own, it becomes crucial when you consider the ever-evolving technology landscape.

Future considerations:

Web 3.0

Composability is core to the principles of Web 3.0; a user-centric, decentralised internet. Open standards and interoperability are key, allowing organisations to easily integrate emerging technologies like blockchain.

AI

Generative AI and large language models are breaking new ground in AI. A composable approach allows easy integration of these tools, enabling organisations to keep pace with AI advancements without major overhauls.

Internet of Things (IoT)

The sheer scale of IoT demands a composable architecture. It's not just about connecting devices but doing so in a way that's efficient and scalable, enabled through robust APIs and event-based architecture.

Extended reality

As AR (Augmented Reality), VR (Virtual Reality), and MR (Mixed Reality) technologies mature, a composable architecture can adapt readily, allowing organisations to more easily experiment with new platforms and delivery mechanisms.

By adopting a composable architecture, organisations don't just prepare for the current digital landscape but create a resilient, adaptable foundation for whatever the future holds.

PART 2.2

Composability in Design



In the realm of software development, the term 'composability' often conjures images of modular code, microservices, and APIs. However, composability isn't a concept restricted to developers; it also plays a pivotal role in design.

While development focuses on building systems from reusable, interchangeable components, design aims to create user experiences that are intuitive, adaptable, and scalable.

When both design and development embrace a composable approach, the result is a harmonious blend of form and function. The synergy between the two ensures that not only is the system's architecture robust and flexible, but its user interfaces are also fluid and user-centric. This section will explore how applying composable principles to design can elevate the entire lifecycle of your digital products.



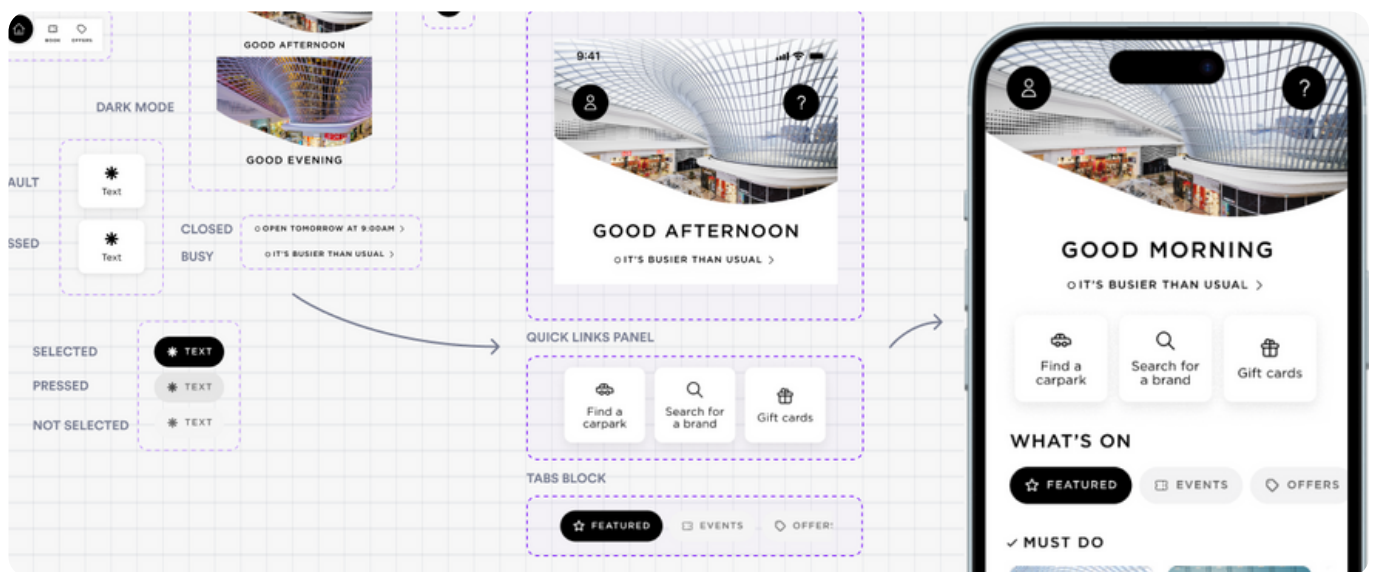
Modular design systems

The idea behind modular design systems is akin to using building blocks.

Let's re-use the LEGO analogy here to explain. Imagine your design elements as LEGO pieces—each one distinct yet designed to fit seamlessly with the others. This approach enables you to construct complex layouts and pages from a set of simple, reusable components. The real value here is efficiency and consistency. When a change is made to one "block," it cascades across all instances where it's used, ensuring a consistent user experience and accelerating design iterations.

This modular approach allows for easier, more cost-effective iterations. It also allows larger design teams to work from a consistent, centralised design system whilst focussing on their specific area of responsibility or functionality. Online design tools like Figma make this collaboration even easier, with real-time syncing of changes that mean updates to a component flow through immediately to other parts of the design system.

By using a modular, component-based design system, you can easily scale your digital products. Need to add a new feature or redesign a specific user journey? Simply rearrange, add, or remove the necessary components without affecting the entire ecosystem.



A modular design system in action

Atomic design

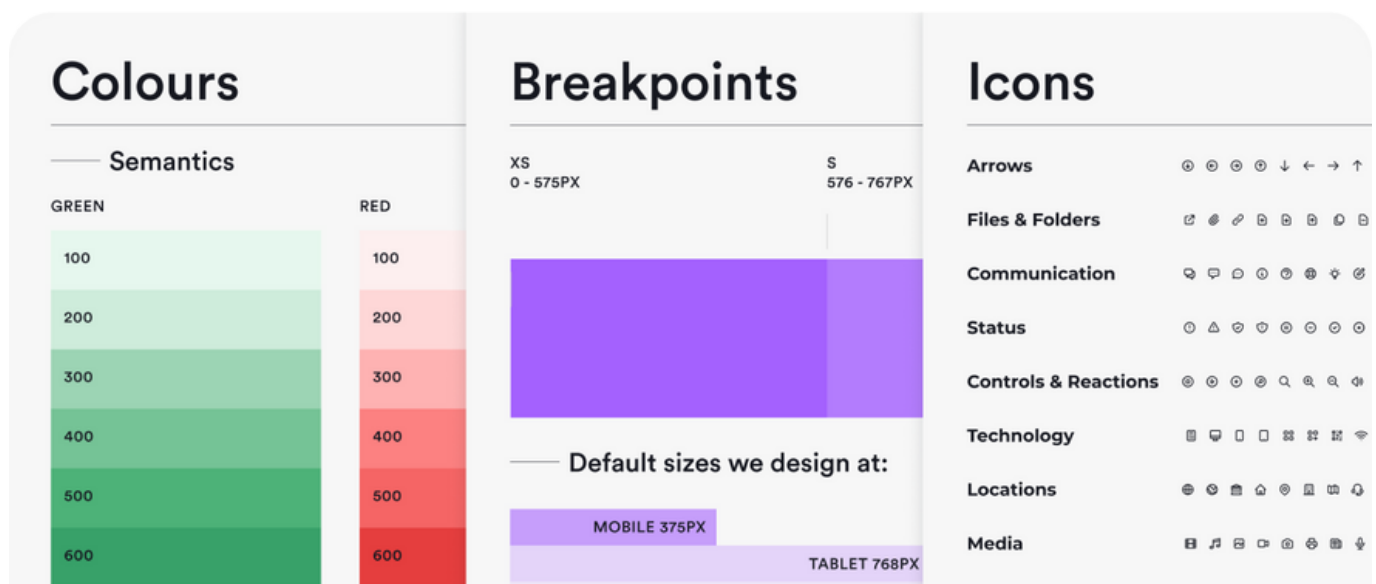
Atomic design is about breaking the building blocks into their most basic forms and working up from there.

You start with fundamental elements like buttons, text, and icons, which combine to form molecules (cards, lists), organisms (headers, footers), and ultimately whole templates and pages. This granular focus ensures you have all the building blocks you need to construct any interface.

Aligned with atomic design, design ‘tokens’ act like variables in programming. Rather than specifying a button should be a particular blue “#0024FF” for example, you'd set it to "Primary Colour."

If that primary colour changes, it can be updated in one place, and the change cascades throughout the design system. This simple yet powerful concept ensures that design elements can adapt quickly to different themes or branding changes.

A composable design philosophy dovetails beautifully with our commitment to a user-centric approach to product development. The flexibility of component swapping enables the ability to adapt swiftly to user feedback, conduct effective A/B testing, and continually optimise the user experience over time.



Figma atomic design set up

Considerations of composable design



Collaboration is key

While composability in design focuses on how interface elements work together, its full potential is unlocked when there's a seamless collaboration between designers, developers, and other stakeholders. When considering the "design-to-development" handoff, it's not merely about passing a static design file to a developer. It's an ongoing conversation, facilitated by collaborative tools like Figma that ensures the design's integrity is maintained during the development phase.

Maintaining a close link between design and UI components also ensures consistency in how components can be organised and utilised within a design concept and end solution.



Future-proofing through composability

As with development, the modular nature of composable design is a built-in insurance policy for the future. When your design components are modular, updates become a breeze. If a design pattern needs to change—be it due to user feedback, updated branding guidelines, or accessibility standards—you only need to update it in one place.

Future-proofing is not just about looking forward but also leveraging what you've built before. A composable design enables you to reuse components across different projects or streams of work. This saves time and resources and ensures a consistent user experience across various touchpoints.

PART 2.3

Composability in Strategy

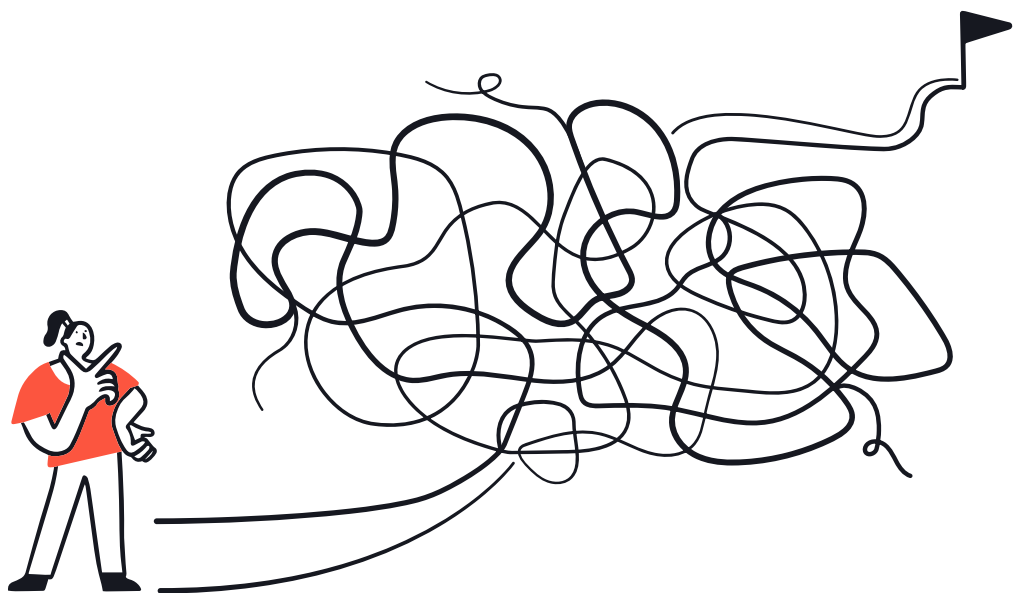


One of the consistent things we hear from our clients and organisations is that they have a well-established long-term technology strategy but a lot of uncertainty on how to practically execute that strategy and start making progress towards their long-term goals.

These strategies often involve a series of interrelated dependencies and significant investments that present significant risk and uncertainty. Combined, these factors can lead to slow decision-making and hesitancy to invest in change, which is counter to competitive progress in a fast-moving environment.

There are **three key ideas** that we advocate for when looking at and supporting an organisation's technology strategy. They all align with composable thinking, and encourage agility and evolution over rigidity and perfection:

- 1 **Evolution over transformation**
- 2 **Product over projects**
- 3 **MVP over end-state vision**



1 Evolution over transformation

The term ‘digital transformation’ often evokes a significant degree of hesitation and consideration within our team. Whilst the process of transformation is often necessary and can be effective, the approach and expectations of these transformations are often misaligned with reality and the level of uncertainty in which the transformation is taking place.

Reasons transformation projects fail to deliver:

Goals are too lofty

It is important to set long-term goals and a desired end state. However, digital transformations often involve a single step from where you are today to where you want to be tomorrow. This can lead to huge shifts in technology, systems, teams and processes; which are extremely difficult to estimate in terms of time to implement and cost to deliver.

Timeline is too long

It is not uncommon to embark on a digital transformation that is roadmapped over several years. The digital ecosystem, as well as customer needs and expectations, are shifting far too quickly to anticipate where you need to be several years in advance, and it is too long to keep your customers waiting on a stale product or experience whilst you put all your investment into the future state.

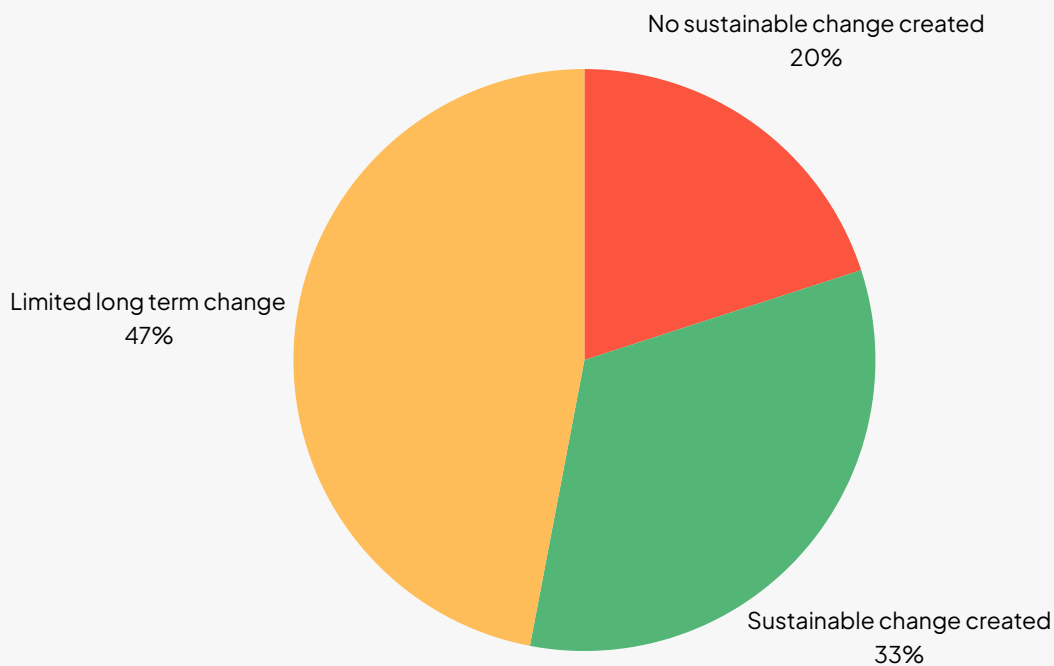
Projects are too big

The delivery of a transformation project often involves the development and implementation of entirely new systems and solutions. These projects require large teams and come with extensive risk due to their scale and complexity.

Where possible, we recommend breaking your roadmap down and taking an iterative approach to evolving your digital products and supporting systems. A composable architecture makes this a lot easier as you can focus on one system or function at a time.

If you are moving from a monolithic solution to a composable architecture, you may require a transformation process to get there - however, it should be **the last transformation you will ever need.**

Only 1/3 of Australian Digital Transformation projects deliver sustainable change, Boston Consulting Group analysis shows:



Corporate Australia's digital transformation track record (BCG Analysis, 2021)

Source: Boston Consulting Group

2

Product over projects

At Inlight, one of our guiding philosophies is simple but powerful: treat digital assets as evolving products, not one-off projects. This shift is more than semantic; it's a fundamental change in how we approach digital strategy. By embracing a product-oriented mindset, we foster continuous improvement, streamlined budgeting, effective teaming, and proactive maintenance.

The traditional project-based approach treats digital assets as isolated initiatives with a defined beginning and end. Once completed, they're handed off, and the team moves on. However, this approach is limiting in an age where the digital landscape is ever-evolving. Enter the product mindset, where assets are nurtured, refined, and evolved over time.

This mindset recognises that even after launch, there's always room for new features and enhancements based on user feedback, technological innovations, and market shifts.

But why products?

Longevity and continuous improvement

Unlike projects that have a finish line, products are designed for the long haul. Continuous monitoring, A/B testing, and iterative improvements ensure the product stays relevant and efficient.

Adaptive budgeting

A product approach embraces a fluid budgeting method. Instead of a lump-sum allocation, resources are adjusted based on evolving needs, ensuring money is invested in features and adjustments that offer the most value.

End-to-end ownership

With products, teams take ownership for the entire lifecycle. There's a deeper investment in the product's success, leading to more holistic thinking and integrated solutions.

In essence, moving from a project to a product mindset is not just a shift in operational strategy; it's a cultural change. It demands a departure from rigid, short-term goals to embrace a vision of continuous growth and evolution, all underpinned by the robust framework of composability.

3

MVP over end state vision

The last idea that we strongly advocate for our clients and partners is to avoid the temptation to aim for the end-state vision in your first release to customers and instead target a minimum viable product (MVP).

An MVP isn't about launching an incomplete or inferior product, but rather smartly launching a product that meets core needs, enabling rapid market entry and real-time learning. There is often considerable excitement generated by visualising a final product that your customers would love, alongside a natural hesitancy to release anything less than perfect into the world. However, customers are usually more understanding than they're given credit for, and they have a greater appreciation for seeing continuous improvement and feedback being taken on board than experiencing a product that is exceptional on the first release but never improves.

Whilst this strategy itself aligns with a composable philosophy, it is made significantly easier through the implementation of a composable architecture.

Rather than committing to an end-to-end monolith solution that has tightly integrated functions, a composable architecture allows you to build and launch a minimum viable product that is easily extensible and scalable over time. These two principles go hand-in-hand.

Planning and designing an MVP and subsequent product roadmap requires a sound understanding of your customer's needs and a way to prioritise value and impact against the effort. At Inlight, we utilise Productboard to capture internal and external requirements, map these requirements to audience needs, drivers and objectives, and prioritise a release cycle to maximise value. This tool also allows us to communicate a roadmap and timeline with stakeholders to validate our assumptions and gather feedback.

PRODUCT	STATUS	Owner	User impact score	Effort	INITIATIVES Address critical painpoints	PEOPLE	DRIVERS Functionality
Zlack							
Communicate with team							
Edit messages		[Avatar]	6	8	🟡	[Avatar]	■ ■ ■ ■ ■
Unfurl links		[Avatar]	0	5	🟡	[Avatar]	■ ■ ■ ■ ■
Share your screen		[Avatar]	46	23	🔴	[Avatar]	■ ■ ■ ■ ■
Rich text formatting		[Avatar]	11	32	🟡	[Avatar]	■ ■ ■ ■ ■
Upload a file		[Avatar]	10	7	🟡	[Avatar]	■ ■ ■ ■ ■

Productboard Roadmapping Tool

Composability for your business & teams

In the digital age, staying ahead isn't just about technological dexterity; it also demands organisational nimbleness.

This adaptability is encapsulated in the concept of "composable business." As highlighted in the Gartner keynote "*The Future of Business is Composable*" (2020), composable business is capturing attention well beyond IT departments; it's resonating across the C-suite and boardrooms.

At its essence, composable business encourages organisations to shift from static, fixed structures to dynamic, modular ones. Drawing a parallel to our earlier discussion on technology, just as monolithic software systems are broken down into discrete, flexible components, composable business urges a reimagining of the entire organisation as a collection of interchangeable blocks. These units can be quickly repositioned in response to both internal and external pressures, ensuring adaptability in real-time.

It's essential to recognise that composable business represents a foundational shift in business perspective. This architectural change ensures organisations are designed for resilience, not just efficiency. As the Gartner keynote emphasises, enterprises are moving towards continuous adaptability, with a significant number of corporate directors accelerating digital strategies in response to ongoing disruptions.

At Inlight, we see the significant potential of this mindset. Our team acts as an agile pillar amidst larger, often slower-moving entities. We offer the agility to adapt, innovate, and deliver, assisting larger organisations to navigate unpredictable landscapes.

In this evolving digital world, challenges are a given. However, with a composable mindset, they become opportunities for growth and innovation. As we forge ahead, we're inspired by the vision of a future where flexibility and continuous innovation are the norms.



PART 3

Where to next?

Exploring the actionable steps for embracing a composable future, this section provides insights on recognising the need for change, choosing the right technologies, and beginning the transformation journey. It also sheds light on how Inlight can be a pivotal partner in this evolution.



How should you move forward?



As we navigate the complexities of the digital era, the need for foresight and strategic planning becomes more pronounced. It's not merely about reacting to present challenges but preparing for the ones that lie ahead.

Ensuring the sustainability of our digital endeavours demands that we not only identify the signs indicating change but also chart a course that guides us towards progressive innovation. Let's delve into the crucial considerations for embracing a composable future.

Recognising the need for change is the first step in any digital evolution. Are you often delayed by rigid legacy systems that can't keep up with modern needs? Do your teams grapple with cumbersome processes that hamper agility? If technological constraints and inefficiencies are holding you back, it's time to consider a new direction. Even if you're deeply invested in legacy monoliths, you might be surprised that many are evolving to offer more modular, composable solutions. A migration rather than a complete overhaul might be a viable option.

When it comes to technologies that enable a more composable, agile approach, the options are abundant. From CMSs like DatoCMS, Contentful, and Sanity to search capabilities via Algolia and cloud solutions like AWS and Vercel, the modern tech stack offers a plethora of options. The key is to build a robust strategy that aligns with your specific needs and challenges, and that's where a partner like Inlight can help.

Switching to a composable approach should be the last transformation you'll ever need. Yes, it involves a significant change in mindset and approach, but the good news is that you can start small. Identify your most pressing needs, and begin by modularising those elements of your business. This evolution can be broken down into stages, making it less daunting and more manageable to get started. Mapping out your requirements will help you draft a plan you can execute step-by-step.

How can Inlight help?

Navigating the intricacies of composable systems can be challenging. **This is where Inlight shines, offering unparalleled guidance and adding substantial value.**

Our experience and proven track record in implementing composable architectures and building composable products means we're well-positioned to guide your journey, whether it involves a complete shift towards composability or modular enhancements to your existing systems.

Moreover, if you or your team would benefit from a deeper understanding of composability and its implications, Inlight offers tailored learning sessions to help build understanding. Through these sessions, we can share insights, strategies, and practical knowledge, providing a foundation for you to harness the power of composability in your business.

In our rapidly evolving digital landscape, being able to adapt and pivot is paramount. Embracing composable approaches can be your key to enduring flexibility, resilience, and responsiveness.



Ready to explore how composability can impact your organisation's future?

Contact us for a conversation or to schedule a learning session tailored to your needs.

inlight.com.au
hello@inlight.com.au
+61 3 9416 4665

inlight

inlight.com.au