mxa consulting

Why you should consider "composable" architectures to modernise your ICT

Introduction for Tech Leaders

August 2024

01 Executive Summary

Most ICT systems were built to be a stable, low cost “monolith”, which are **extremely difficult and expensive** to maintain and change.

There are two main alternatives to monoliths; microservice architectures and composable architectures.

- A **microservice architecture** is an approach where ICT systems are built as a collection **of small, loosely coupled**, **independently deployable** services.
- A **composable architecture** is a modular approach where the ICT environment is built from interchangeable, plug-and-play components, including internally developed services (such as microservices) and vendor SaaS products. that promotes flexibility, and strong business-ICT collaboration.

Microservices solve many monolithic problems in theory but **require high technology maturity** for success.

Composable architectures, on the other hand, **provide an “easy road”** for all needs where “near enough” is good enough (in which case a full product SaaS will do), **as well as the ability to build** microservices in use-cases that justify the increased investment. It’s the **best of both worlds**.

Composable architectures becoming more common; made possible by modern improvements including increasingly sophisticated Software-as-a-Service offerings and integration platforms.

There’s no free lunch though; to succeed with a composable architecture, organisations need a proactive operating model, an architecture that promotes flexibility, and strong business-ICT collaboration.

02 Monolith ICT systems were not built to be easy to maintain and change

An ICT "monolith" is a software system built as a single chunk. They are often efficient and low cost to keep exactly as they are. They exist primarily because, before the arrival of cloud Software-as-a-Service (SaaS), integration between services was extremely difficult and expensive, and vendors had no other way to package, install, upgrade, and license to their thousands of users.

Unfortunately, the world around us is constantly changing in a way that is very difficult for us to avoid. Business needs, security threats, changing vendor support, the evolving talent market and more are all factors that impact our software that we can't avoid.

Monoliths were not built to be easily updated or changed to keep up with these factors. Monoliths are also very expensive to customise, with many costs coming later.

For example, one MXA client paid $7M for a large ERP solution, before spending an additional $40M to customise it. Then, when the ERP vendor had a major upgrade, the client had to spend another $40M to implement the upgrade with their changes.

The final challenge with monoliths comes from testing. The complexity means that few people understand all of the internal workings. This means that any change, no matter how small, needs a full end-to-end-test regime to ensure confidence that the change hasn't broken something in another functional area. This is time-consuming and costly.

03 There are three standard approaches to dealing with the challenges of monoliths

The standard approaches to dealing with the challenges of monoliths are:

1 Reactively fund remediation initiatives when challenges become too big to tolerate.

2 Transition to microservices by breaking down the monolith into small chunks

3 Transition to a composable model, by progressively replacing with fit-for-purpose SaaS

Technology laggards have dealt with the monolith problem by reactively funding **remediation initiatives** without dealing with the root cause. This is like fixing a leaky boat by putting your finger over the hole; it works fine provided the holes aren't too big and there aren't many.

Unfortunately, the longer they avoid addressing the underlying issues, the more expensive and risky maintenance and improvement activities become.

On the other hand, those looking to leapfrog to be a technology leader will **transition to microservices**, which means carving up the monolith into autonomous bite-sized pieces. Unfortunately, this is extremely difficult, owing to the significant increase in capability and dramatically different operating model required to be successful.

The final approach is to **transition to a composable** model. This is similar to transitioning to microservices, but with a preference towards utilising vendor SaaS solutions. Build efforts are restricted to specific use-cases with a very strong business case. As such, the challenge with this approach is that it needs a very different operating model; one that allows careful selection of SaaS as well as an architectural design that minimises vendor lock-in.

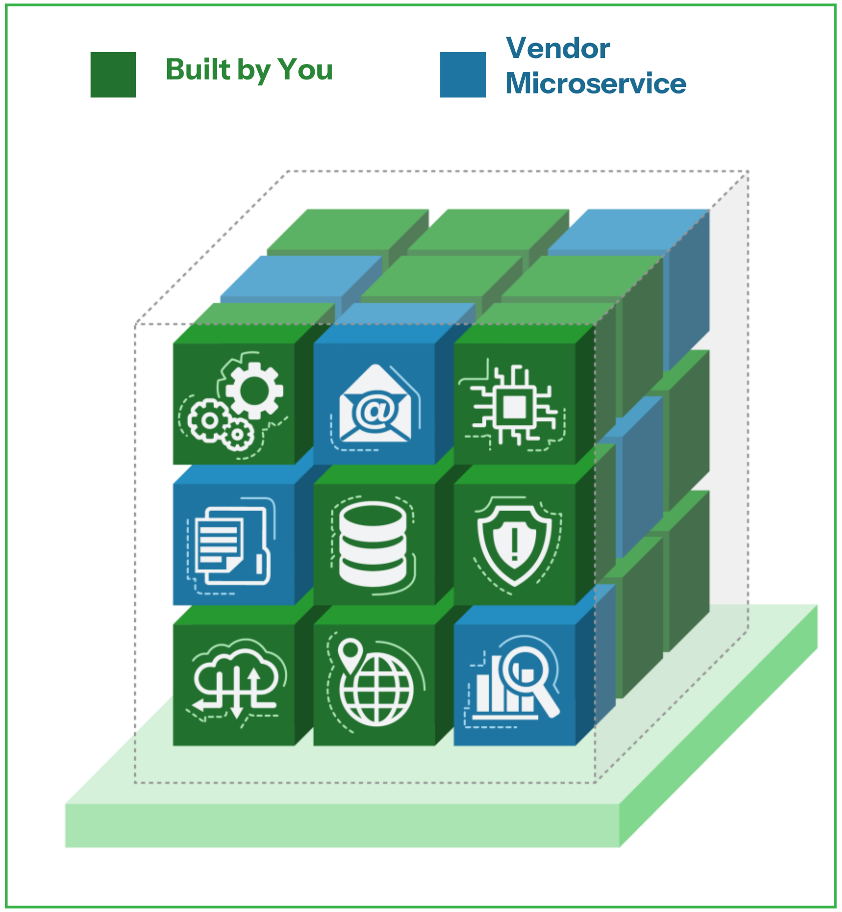
04 Microservices are advocated by digital leaders as a way to increase agility...

A "**microservice architecture**" is a design approach where applications are built as a collection **of small, loosely coupled**, **independently deployable** services, each responsible for a **specific piece of business functionality** and **communicating over standard protocols**.

Microservices were designed for high-performance, hyper-scale companies with ample supply of top-tier engineers. They have rarely been adopted effectively by organisations outside of big-tech and other class-leading organisations.

What are the Benefits of Microservice Architectures?

- Enabling faster iterations and quicker time-to-market through the independent development, deployment and updating of services.
- Improved overall system resilience by isolating failures, preventing a single point of failure from affecting the entire ecosystem.
- Optimised resource usage and system performance by independently scaling services based on specific needs.
- Enhanced maintainability though the management of smaller, modular codebases.
- Increased efficiency by aligning teams around specific services, fostering ownership, accountability, and streamlined processes.



mxa consulting

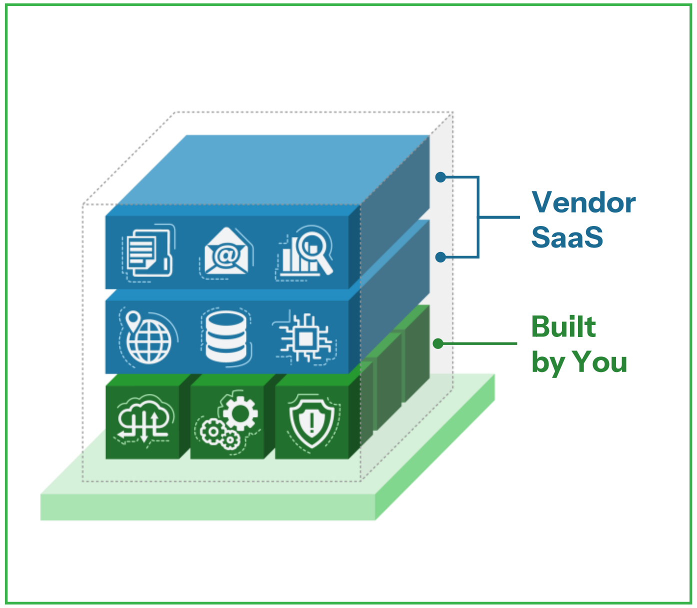
… but they're hard to implement well

Why do organisations fail to gain the benefits when transitioning from "monolith" to microservices?

- **The cultural shift is large:** Transitioning to microservices requires a cultural change towards agile practices, automation, and team autonomy, which can be difficult for organisations accustomed to monolithic and top-down approaches.
- **The end-state is complex:** Managing many microservices instead of a single monolith is a big step-change in difficulty that many organisations don't have the needed talent or sophisticated technical practices to execute.
- **The end-state can be expensive:** When not managed well, it is common for development staffing, cloud consumption and hidden license costs to ballon, blowing the business case and creating friction between ICT, business, and finance areas.

mxa consulting

05 For many organisations, composability offers a compelling alternative



Composable Architectures can be a pragmatic solution to being able to change without the difficulty of microservices

A **composable architecture** is a modular approach to system design where the ICT environment is built from interchangeable, plug-and-play components, including internally developed services and externally sourced SaaS products. This enables easy integration, scalability, and adaptability.

Composable architectures differ from microservices mainly by emphasising assembling vendor SaaS products, rather than building your own solutions. Doing this correctly can allow organisations to gain much of the value of a custom solution at a fraction of the cost.

What are the Benefits of Composable Architectures?

- Reduced costs for maintenance or large remediation projects, as SaaS vendors will cover most of this.
- Less complexity, as there are "few pieces of the puzzle" to manage.
- Best of both worlds; ability to incorporate built applications (e.g., microservice, monolith) where it makes sense.
- Can be more secure in practice; since large SaaS vendors often have superior talent, and greater funds for security management.
- Lower requirements for recruiting and retaining expensive engineering talent. Access to a wider pool of digital talent (such as digital marketers, CRM experts, and customer support) who can operate your solutions.
- Less management time spent on technology build projects, providing more time to focus on business priorities.

06 Composable isn't a new concept, but it's only recently become viable and feasible

The rise of composable architectures are driven by four factors

Driver	
Increasingly sophisticated SaaS products	There has been an explosion of **high-quality SaaS products** providing **ready-to-use solutions** that can be configured to meet the bulk of business needs, can be **easily integrated**, and can **scale as an organisation grows**[1]. Moreover, these products come with robust **security features and compliance certifications**, making it easy for internal teams to meet regulatory and risk requirements[2].
Advances in Integration Management	Many modern SaaS providers design their products with an **API-first approach**, ensuring **seamless integration and easier orchestration of services**[1,2]. In addition, there has been a **rise in Integration Platforms** (e.g., Mulesoft, Workato) that are designed to enable organisations to easily connect and disconnect systems and workflows[3].
Frequent Technology Build Failures	**Building technology is difficult**, **often very expensive**, and it can be a distraction from your core mission. A study conducted by Boston Consulting Group, interviewing over 400 C-suite executives found that more than 30% of technology development projects experienced cost and schedule overruns, and **more than 50% delivered less than satisfactory benefits**[4].
Accelerating need for business agility	Business areas expecting **faster responsiveness and change from ICT functions**[5]. This includes being able to rapidly bring new offerings to market or implement new policies. Composable architectures provide a relatively **low-cost-and-effort way of responding**, by configuring existing SaaS products in the environment, or quickly replacing them if they cannot meet evolving needs[6,7].

07

There's no free lunch; composable architectures require a different operating model

Success requires a proactive operating model, a strategy and architecture to mitigate vendor lock-in, and increased business-ICT collaboration

Strong, proactive, operating model

ICT must be resourced to be responsive to business needs by proactively understand the SaaS market and engaging with business areas. Strengthened processes are needed to govern solution choices and changes.

Vendors must be managed carefully, since critical business capabilities will be served by vendor solutions. Vendor management resources must have the skill and the teeth to deal with vendors who are not acting in the organisations best interest.

Strategy and Architecture that maximises agility

Over time, procured software solutions will no longer be the right match for your organisations needs.

In addition, vendors will implement lock-in strategies to make it challenging to change when their solutions are no longer fit-for-purpose.

To ensure agility, architecture governance and design mechanisms must be aligned to the intent of composability, and technical enablers are required (e.g., modern integration and data platforms).

Increased business-ICT collaboration and empowerment

By making heavy use of SaaS, business and ICT must accept that they are gaining most of the benefit of a custom build at a fraction of the cost.

This acceptance means compromises will need to be made. For example, some low-importance business requirements will not be met, and some processes will need to be redesigned to fit the software

In addition, ICT will need to establish clear guardrails and standard patterns to empower business areas to leverage SaaS low-code/no-code capabilities.

08 Implementing a Composable Architecture requires initiatives to establish the new operating model

01 Build the Case for Change

1. **Conduct and ICT audit**, including all existing systems, roadmaps, data and contracts.
2. **Develop a business case** and **recruit sponsors/champions** from business and ICT areas.
   - Leverage the insights from the ICT audit to define cost-saving initiatives that will help pay for the transition.
   - Map out key constraints, such as business seasonality, contracts, difficult to remove systems etc.
3. **Communicate the case for change and benefits** clearly; anticipate and manage resistance from areas that will lose influence (e.g., ICT build) and those who maintain the status quo (e.g., governance for policy and process constraints).

02 Establish Firm Foundations

1. Uplift **Business and ICT strategies** to support the new direction.
2. Develop an **implementation plan.**
3. Design your **new operating model**, diverting resources from technology build to procurement, vendor management, architecture, security, integration, and building tighter business ICT processes.
4. Design your future state and interim **Business and ICT architecture**.
5. Depending on your business case, align with finance on a **funding model** that shifts some emphasis from capital funded projects to predictable operational costs (primarily SaaS licenses).
6. **Upskill** people on the new composable architecture approach to ensure everyone is aligned and capable of executing the plan.

03 Continue to Grow Needed Capabilities

1. **Resource and operationalise your new critical capabilities**; procurement, vendor management, architecture
2. Develop a **vendor strategy** and procurement policy and guardrails
3. **Procure and implement key ICT enablers**, such as an integration platform and data infrastructure (e.g., data lakehouse), if your scale and use cases justify

mxa consulting

09 References

1 MuleSoft. *Opportunity on Demand: The Rise of the Composable Enterprise* [Whitepaper].

2 Natis, Y., Santoro, J., Liversidge, J., White, S., Petri, G., Vincent, P., & Thomas, A. (2023, January 17). *Predicts 2023: Composable Applications Accelerate Business Innovation*. Gartner.

3 Boomi. (2023). *From Chaos to Order: Connecting a Fragmented Digital Architecture: A Roadmap for Reducing Digital Application Sprawl Through Modern Integration* [eBook].

4 Palumbo, S., Rehberg, B., & Li, H. (2024, April 30). *Software Projects Don't Have to Be Late, Costly, and Irrelevant*. Boston Consulting Group. Retrieved from https://www.bcg.com/publications/2024/software-projects-dont-have-to-be-late-costly-and-irrelevant.

5 Snow, R. (2022, May 6). *What Government CIOs Need to Know About Composability*. Gartner. Retrieved from https://www.gartner.com/en/articles/what-government-cios-need-to-know-about-composability.

6 Jerenz, A. (2022, September 8). *Winning in Digital Banking*. McKinsey & Company. Retrieved from https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/tech-forward/winning-in-digital-banking.

7 Gaughan, D., Natis, Y., Alvarez, G., & O'Neill, M. (2020). *Future of Applications: Delivering the Composable Enterprise*. Gartner.

Web illustrations by https://storyset.com/

10 Authors

Dr Rishni Ratnam
CEO
rishni@mxa.com.au

Barry Saunders
Partner
barry@mxa.com.au

A/Prof Peter Grant
Distinguished Executive
peter@mxa.com.au

mxa consulting

National Office
Level 35
100 Barangaroo Avenue
Sydney NSW 2000

hello@mxa.com.au