

Images haven't loaded yet. Please exit printing, wait for images to load, and try to print again.

Innovation Enablers For Enterprises



Enabled Solutions [Follow](#)

Aug 20, 2018 · 19 min read



Is this what your IT system looks like to Silicon Valley Startups?

Startups build software to solve their customers' problems rather than using bought IT to streamline their internal processes.

In this long blog post, we cover these main topics:

- Win with software, or be eaten by it
- The power of customers = speed & agility ever more crucial
- Contrasting Enterprises vs. Startups in viewing IT
- How did the divide happen?
- Gartner predicted “death of ERP”
- Monoliths vs. the new software architecture
- How top-performing businesses innovate with *Microservices*

. . .

Being Eaten by Software

“If you went to bed last night as an industrial company, you’re going to wake up today as a software and analytics company.” This is a statement from GE’s then CEO back in 2014. The underlying message of his statement is that unless you invest in software you will not remain competitive and so you may never wake up.

Yet until recently software and IT in general, had never proved to be source of competitive advantage. For instance, you buy a mainframe to better manage your business, so does your competitor. You buy a laptop for all of your employees, so does your competitor. You put in the leading ERP system for your industry, so do all your competitors.

All this has dramatically changed with the arrival of the digital natives like Airbnb, Uber and Netflix who are gaining an advantage by configuring and investing in their own software, unique to them and accessible by no one else, and they are disrupting markets.

How has this happened? According to venture capital firm [a16z](#), who coined the phrase “software is eating the world”:

“Six decades into the computer revolution, four decades since the invention of the microprocessor, and two decades into the rise of the modern Internet, all of the technology required to transform industries through software finally works and can be widely delivered at global scale.”

Unfortunately, existing companies are not the ones who are winning with software, but rather, they are the ones being eaten by it.

Ultimately, this article will explore how existing businesses can change their ways to successfully leverage IT and enhance their competitive advantage. But to do this correctly, we need to shift our attention from IT, towards the sole reason for a business to exist... “To increase shareholder returns!”

Sorry, I jest, it should have been “the Customer!”—how can we satisfy customers better and faster, by offering products or services that truly take care of their unmet jobs.

The Power is now with your Customers

The arrival of smart devices like the iPhone has made computing a very mobile, accessible and personal experience.

The internet provides consumers with a plethora of choice, meaning geography and distribution costs are no longer the barriers they once were.

"If you make customers unhappy in the physical world, they might each tell six friends. If you make customers unhappy on the Internet, they can each tell 6,000 friends."—Jeff Bezos

Customers of today are highly empowered, motivated and self-determinant. Armed with knowledge and computing power at their fingertips, they are quick to seek out the best/cheapest/most advanced etc. option.

As we've discussed in another blog article, customer experience is the brand goal.

Extending from that point, employees should be considered 'internal' customers. As connoisseurs of software, they are no longer happy to use cumbersome and limiting software provided by the IT division.

With millennials making up a significant portion of the workforce, it is also important to note how they might be shaping the workplace. A report by PwC discussed the specific expectations millennials have about technology at work, e.g. employer's provision of state-of-the art technology, and being able to use workplace technology alongside their own.

Speed and Agility is Key

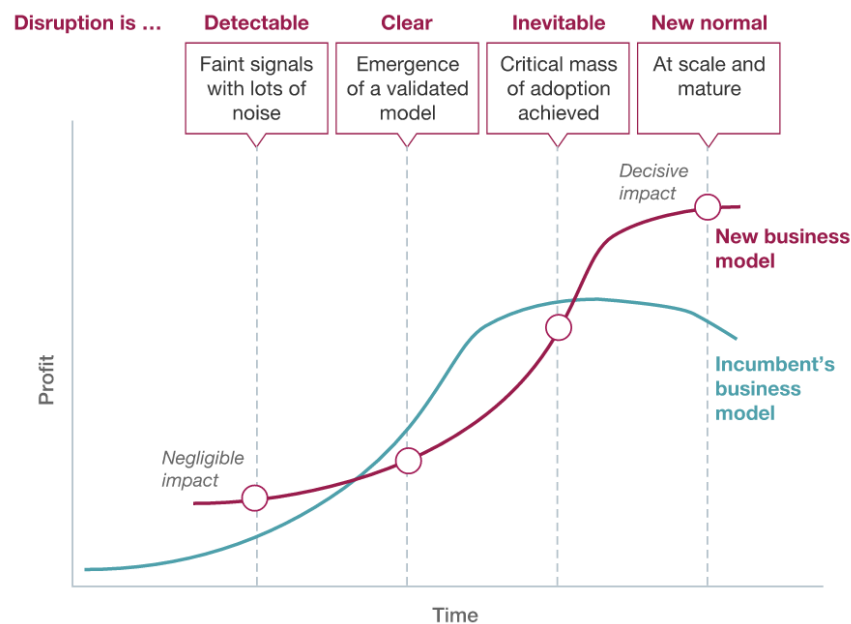
If all customers have this power over your brand, it is no longer about being able to predict the future 12–18 months in advance, it is about being able to quickly adapt to the changing expectations of your customers.

Those businesses who can tweak and adapt their business models quickly have a better opportunity to grab market share and maintain a competitive edge.

The latest survey on the state of corporate innovation notes that high-performing companies were 3x more likely to strive for first-mover advantage, i.e. they understood the importance of speed. Today, this is more important than ever, as leveraging the internet can enable speed of scale, while significantly reducing costs. Think about how much easier it is to set up a news website than it was to start a newspaper.

Failure to recognise the importance of speed, adaptability and agility will have a highly detrimental impact on a business' success. For example, a business may be coasting along just fine and seemingly within a couple of weeks, their business model has been severely challenged, and possibly even disrupted.

Think taxis and Uber, travel agents and comparison sites, DVD rentals and Netflix etc. You can read more about business model disruption in our article on digital transformation.



So what are businesses doing in the face of changing customer demands and tech advancements? Fundamentally, there are two contrasting groups in terms of how they use digital technology. For the sake of this article we will label them as follows:

- Enterprises: existing incumbents in their industry, who are mostly non-tech companies

- Startups: digital natives trying to displace the incumbents

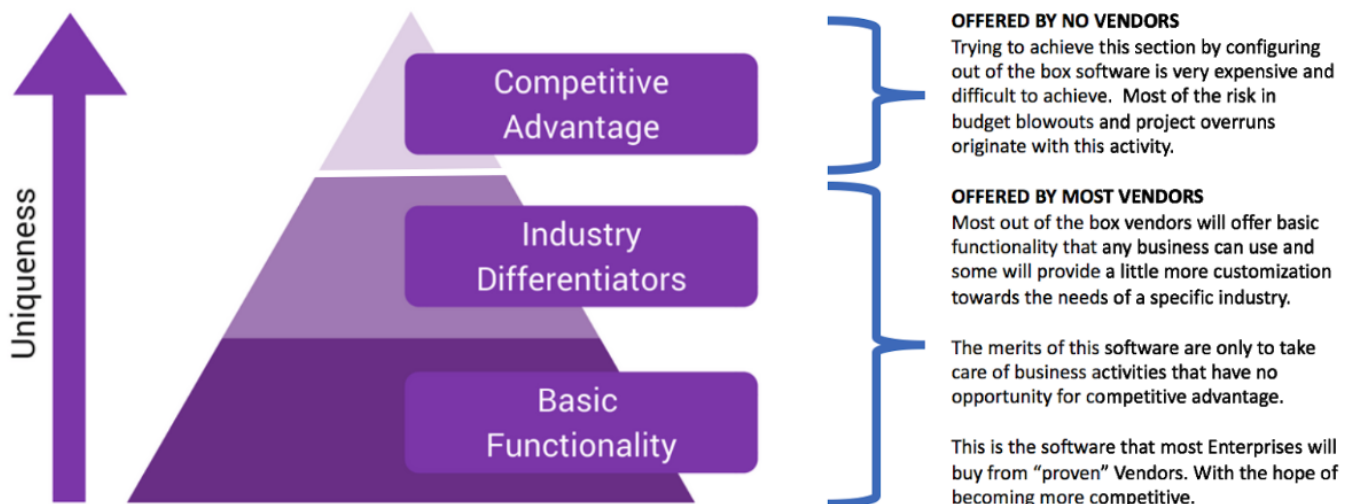
Enterprises

Corporate Enterprises have all-encompassing IT systems, most often purchased through 'proven' vendors.

These systems are employed to solve internal problems and optimise according to industry standards and perceived best practice.

As shown in the graph below, those purchasing any of the available off-the-shelf Enterprise grade software will only receive the basic functionality required by their industry.

DIFFICULTY IN GAINING ADVANTAGE FROM OUT-OF-THE-BOX SOFTWARE



Thus this methodology raises two key issues:

- Using the industry standard makes you standard in the industry
- Following what is labeled 'best practices' may only mean watering down your competitive advantage

This means enterprises:

- Cannot create a point of difference for their customers
- Struggle to keep up with shifts in customer preferences and behaviours

Best practices, or more accurately understood as common practices here, will only get enterprises so far. Then to try and gain a competitive advantage or address a change in customer preference, enterprises may work hard to customise restrictive out-of-the-box software.

Unfortunately, this generally leads to added complexity, time, and cost to IT budgets, making competitive advantage an elusive goal. This is where most of the unpredictability and risks of enterprise system implementations lie, yet it is the most commonly accepted practice.

Startups

In contrast to Enterprises, successful Startups don't use IT to support their business. Rather, they focus on customer problems that can be tackled through any means they have, which is usually software.

They orientate around a small set of problems and, starting with a clean slate, they optimise to solve them really well.

Unencumbered by the concept of industry best practice, Startups are equipped to invent business models that have never been seen before which can disrupt existing business models.

Again using examples of well-known successful startups: Netflix invented the on-demand movie streaming model, Uber invented the on-demand ride hailing model, Amazon invented the on-demand ebook publishing model.

Let's contrast these two approaches:

- Enterprises solve their own problems with IT
- Startups solve customers' problems with software

Which is more powerful?

"Take your eyes off the problem (customers' problems) and soon a competitor will be solving it better."

How did this divide happen?

The enterprise software journey

A few decades ago, software started as 'single purpose' often aligned with a single business function, e.g. finance.

Over time, other software arrived to benefit more business functions. These software applications were separate and poorly connected, which meant passing data between them was hard. In the beginning, they were separated because connecting the physical computers was hard. Applications and the data they created would be passed on physical media like cards and discs.

Software was expensive to buy and maintain. Updates required major work and carried risk to the systems' availability.

These factors made dealing with a single vendor appealing, which in turn motivated single software vendors to address multiple business functions.

This desire for integration turned functionally discrete software applications into monoliths intended to do everything for the enterprise.

The epitome of the enterprise monolith is the Enterprise Resource Planning (ERP) system.

Monolith of the ERP kind

ERPs were meant to realise efficiencies by sharing information across business functions and organising this information into a central place for decision making.

ERPs have instead become the 'ball and chain' of Enterprise:

- They are expensive, consuming time and budgets
- They are too rigid to adapt, and thus too slow to gain advantage
- Often no one person fully understands the system, yet this is almost a requirement
- They are not created on open standards, thus talented staff pools are restrictive
- They can be complex so users are motivated to find other ways of getting business done outside these systems (often sticking with more manually intensive and error prone paper methods)

- They still remain orientated to business functions, rather than being customer-oriented. (Read more on the [Jobs to be done](#) framework, a customer-oriented approach to product development including software)

Customisation is very hard

Monolith software by nature assumes they have all things covered in one application. Moreover, ERP providers may not be motivated to allow other software to work with it, as it is “cutting their lunch.”

Enterprises must rely on a limited supply of domain experts: you either need to get in line and pay through the nose to make a change, or employ and maintain dedicated people.

Note that monoliths are not exclusively ERPs. Some website content management systems (CMS) and customer relationship systems (CRM) fit the Monolith description and are equally as burdensome.

In 2000, the Gartner Research Group predicted that by 2018, the concept of using [ERPs](#) and monoliths would be “dead.” They proposed a change in business strategy to enable more open and collaborative processes, rather than the procurement of a single monolith product to optimise internal enterprise performance.

Yet today many companies are just embarking on a 2–3 year ERP journey. They are buying the unwritten mantra of ERP vendors “providing all things to all people,” which renders the monolith “ill-suited to a future that demands focus and [external connectivity](#).”

Failure risk is high

From a project management point of view, large scale projects such as ERP implementation carry a larger risk than smaller ones. As obvious as it sounds, ERP projects are a “[potential minefield](#)” because there are just more moving parts that need simultaneous attention.

Check out this list of [ERP disasters](#).

In a world where speed is one of the key defenses against disruption for businesses, being bogged down by legacy systems presents a big concern. How can we responsibly bring a stop to this behaviour?

“We will be competitive in 24 months when our ERP project is rolled out,” said a leading player in Australian aged care.

New software

In line with Gartner's predictions of a more open and collaborative business strategy, there has been a visible shift to the newer cloud architecture—pioneered by innovative tech companies.

The cloud allows the “on-demand delivery of compute power, database storage, applications, and other IT resources” via the internet rather than your on-premises hard drive.

For instance, back then Microsoft Office was an application that you installed and accessed locally on your computer. Now you can have the web-only version of Microsoft Office applications via Office 365, just like Google's web-based G Suite products.

But, simply shifting a monolith to the cloud only means same ball and chain on a different computer—all the shortfalls of a monolith can remain.

Mega software companies like Amazon, Google, Facebook and Netflix realised this problem. Their monoliths became so large and so distributed that they started facing unprecedented scale issues:

- Their hardware and software requirements became too big to understand holistically by any one team
- The release of any one small feature required the release of the entire system, thus presenting a risk to their systems as a whole.
- Their employee and customer base became too distributed to manage centrally
- The arrival of mobile devices only exacerbated this dilemma, with multiple types of devices, models and operating systems to manage



New paradigms

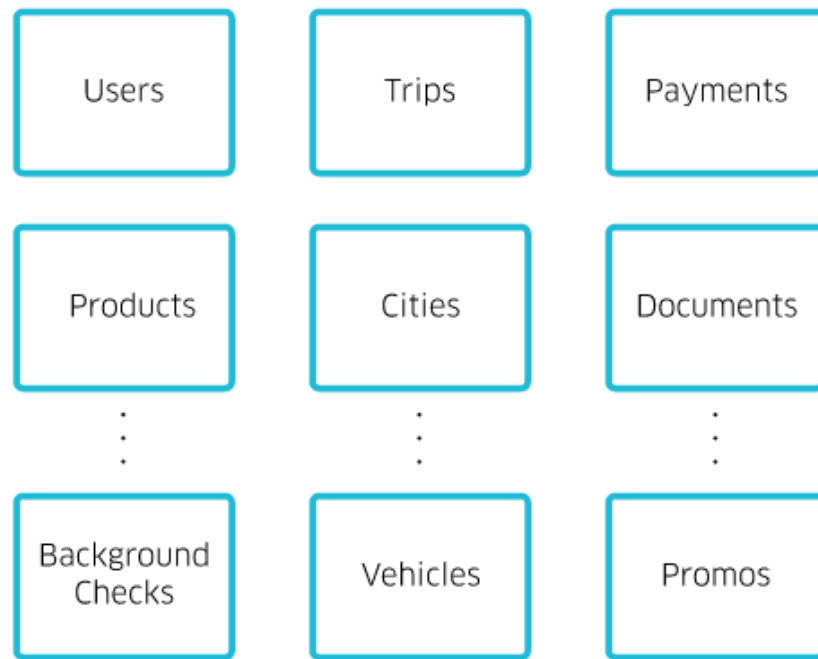
As cloud computing means having no “real” computers, it creates computing power which is elastic and controlled by automated software, thereby increasing or decreasing with demand.

This has led innovative companies to abandon massive teams building software monoliths, in favour of having smaller teams focused on building smaller, limited purpose software elements called Microservices.

Microservices started as an architectural style whereby a specific, well-encapsulated domain area (or business capability) is developed as a suite of small services.

The idea is the microservice dedicated to that capability, can become the best it could be. Independently deployable, each of these microservices then communicate to each other via well-defined interfaces called Application Programming Interfaces (APIs).

Take a look at the Uber app's architecture consisting of these building blocks or microservices.



Source: Uber

This doesn't mean a solution is cobbled together

An important distinction to make: a microservices architecture is not cobbling different software components together.

For instance, it is not the case of using a piece of software over here, then exporting some data into a spreadsheet or CSV only to import it into another piece of software.

In the scenario above, there is no agreement between the two independent softwares to maintain any consistency, thus if one on either side makes a small change, even as small as renaming the column name in your spreadsheet, the whole process will break.

Whereas with the microservices architecture, these small components are independent and loosely coupled through an agreement to keep a consistent entry and exit point to their software via a documented API.

Thus, microservices talk to each other via these APIs to ensure the business operation is seamless and real time.

APIs and their consistency is one of the crucial mechanisms to escape monoliths, as they essentially are the “contract” between microservices.

Again using Uber as an example, one microservice that they have is for fetching up-to-date currency and exchange data. If the app was “cobbled” together, Uber wouldn’t be able to serve riders simultaneously paying in nearly 60 currencies around the world.

Monoliths vs. microservices

Let’s bring it back and clarify the difference between the old and new paradigms—or monoliths and microservices.

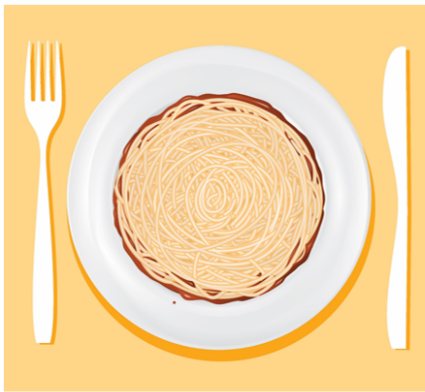
Using a food analogy, we have monolith software as spaghetti. It is an integrated meal with two main parts, the noodle and the sauce.

Although the dish is somewhat customisable at its construction with different pasta shapes or a different sauce, once constructed, it is impractical to swap one of the components out for another as they are too tightly integrated. For instance, once you discover that your dinner guest is gluten intolerant, you cannot adapt quickly to their needs.

A transitional state called Service Oriented Architecture (SOA) is an improvement over a single Monolith. The services here still have to be carefully shaped to work with the overall solution. Think more of swapping one slice of cake out for another of the same size and shape.

Unfortunately, SOA is still dominated by vendor lock-in, meaning you could only use cake slices from the one cake store.

On the other hand, microservices could be considered a box of donuts. No trouble with gluten intolerance here as functionally, one could be swapped out for a wheat-free chocolate brownie, or even a piece of fruit. The key is: they have no dependence on each other, they are self-contained pieces of software tackling a very discrete function.



With monolithic, tightly coupled applications, all changes must be pushed at once, making continuous deployment impossible.



Traditional SOA allows you to make changes to individual pieces. But each piece must be carefully altered to fit into the overall design.



With a microservices architecture, developers create, maintain and improve new services independently, linking info through a shared data API.

Kanban Solutions @kanbansolutions kanbansolutions.com

Source: DZone

Another often used analogy is the shipping container. This technology revolutionised the shipping of goods. The shape of the goods no longer mattered, as long as they could be placed in a container, they could be moved through a set of standardised interfaces, or APIs in our analogy.

Also just like a shipping container, it does not matter what the microservice contains, or even what computer language it was developed with. This containerisation has even become a class of technology which wraps a microservice, making it deployable to any computer environment. We also wrote about container as a technology in another post.

“Saasification”

Along with the rising popularity of cloud and microservices, don't forget about SaaS (Software as a Service). SaaS is a subscription-based, delivered-over-the-Internet software model.

It is possible today to find very large SaaS offerings that could almost be considered to be monoliths. For instance, the Salesforce CRM is an enterprise-grade SaaS product that comes close to being a monolith, yet within its own suite of applications, it does implement strong APIs providing some microservice benefits.

Additionally, there are many micro-SaaS applications that are almost microservices in themselves and have very strong APIs that can be used by other applications.

For instance, besides their own custom services that make up their app, Uber has used a bunch of 3rd party SaaS applications for different business capabilities:

- Google Maps for location and navigation services
- Twilio for SMS notifications
- Braintree for payment processing
- SendGrid for automating email campaigns

What are big software vendors doing to combat this shift?

Note: we used the word “combat” rather than “adapt” or even “embrace.” Microservices are seen as a threat by many incumbent software vendors than a better way of benefiting customers.

Any ERP software vendor who started 10 years ago may still be trapped trying to transition in methods and technology. With customers locked in, they may not even be motivated or able to transition to the postmodern ERP world.

This postmodern ERP world presents complexity and a high failure rate for those who “assume the vendors peddling cloud will take care of it. When—invariably—they don’t.”

Why?

Because many vendors themselves cannot do anything to combat the shift. They are trapped between needing to move to the new and getting caught in a sunk cost fallacy, i.e. they don’t want to abandon their existing software asset. This is very similar to the simplified story of Blockbuster not wanting to give up their DVD stores, then ultimately going out of business because of the faster-moving Netflix.

Guilty of putting self-interest ahead of their customers, these software vendors must raise their game or risk being shunned by their enterprise clients who increasingly demand value in less than two years.

And don't be fooled by many who claim to provide so-called cloud ERP—these are often released by the same vendors and built by the same software architects of their legacy monoliths.

How can someone say they help their client with “digital transformation” when all they do is repackage an aging ERP system? It's merely the same ball and chain on a different computer.

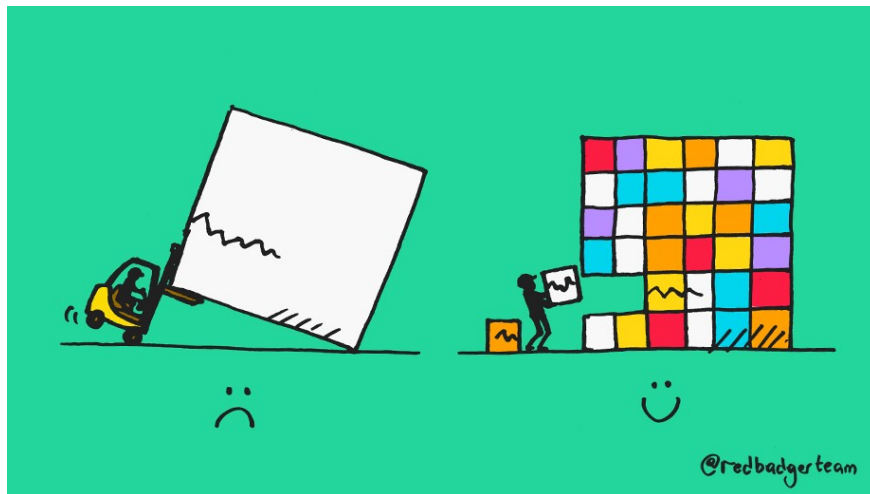
Benefits of microservices

Meanwhile, the innovative companies who have understood these shifts, are already reaping benefits from microservices in a number of ways.

Efficiency

Microservices are reusable and interchangeable, which makes replacement with new pieces much easier than fixing the whole monolith. It provides an evolutionary architecture that can grow and adapt to the changing business landscape.

For instance, if your business intelligence dashboard is no longer meeting your needs, you can swap it out for another more suitable option without breaking the whole system.



Source: Red Badger

Consider this excerpt from Terry Pratchett's book, [The Fifth Elephant](#):

“This, milord, is my family's axe. We have owned it for almost nine hundred years, see. Of course, sometimes it needed a new blade. And sometimes it has required a new handle, new designs on the metalwork, a little refreshing of the ornamentation . . . but is this not the nine hundred-

year-old axe of my family? And because it has changed gently over time, it is still a pretty good axe, y'know. Pretty good."

The axe is modular: blade, handle, ornamentation. As one of these wears out, it can be swapped for a new or better option. Today, the wooden handle could be replaced with a more impact absorbing fiberglass option. Eventually for the blade, you might go for a hardened titanium core with a carbide-tungsten edge, sandwiched between two layers of carbon fibre.

Enterprises can realise cost savings and become far more resilient by leveraging the underlying modular nature of microservices.

Contrast this with the traditional IT governance environment where enterprises want to standardise as much as possible to maximise resource sharing. Thus, they tend to build a list of the most complex possible use cases and try to buy a single tool to account for all scenarios. Think building a "what if" shopping list.

For example, a team might only need a simple data store without sophisticated query functionality. But they still have to settle with an industrial strength database server, which presents unnecessary complexity, inefficiency and much larger licencing fees.

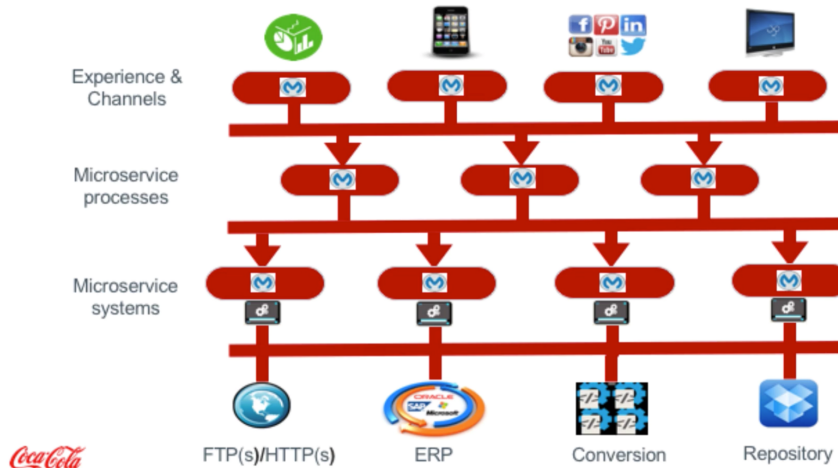
Microservices are also designed for risk mitigation/management:

- Less likely for an application to have a single point of failure because functionality is dispersed across multiple services
- Hence, applications can perform better and have less downtime

For businesses with lots of branches or subsidiaries, microservices architecture can help reduce time and costs, while still making sure it works seamlessly.

For example, multinational conglomerate Coca-Cola moved away from on-premise software to microservices, which has helped them create great customer experience.

Microservice architecture framework



If interested, you can watch this webinar where Coca-Cola's tech people talked about this [microservices implementation](#) journey.

Incremental improvements

Any microservice-oriented software enjoys the ability to incrementally enhance with superior agility. How? As mentioned above, microservices are independently replaceable, thus upgrades to the modular components can be done in a very agile manner. Therefore:

- Custom software projects are easier, shorter, faster to realise, and simpler to understand
- With continuous improvements, customer retention and engagement can also increase as you can rapidly adapt to their needs or because of their behaviours.
- A less obvious impact on human resources is you might attract and retain more talented engineers. As microservices enables you to incrementally adopt and test new technologies, which is what good engineers want to do, and when they cannot, they will leave. This is in fact what is happening with many monolith vendors, with good talent leaking away, they leave behind those stuck in the old paradigms.

Empowering innovations

With microservices, enterprises can now foster new business models and be adaptive to the changing needs of customers and employees.

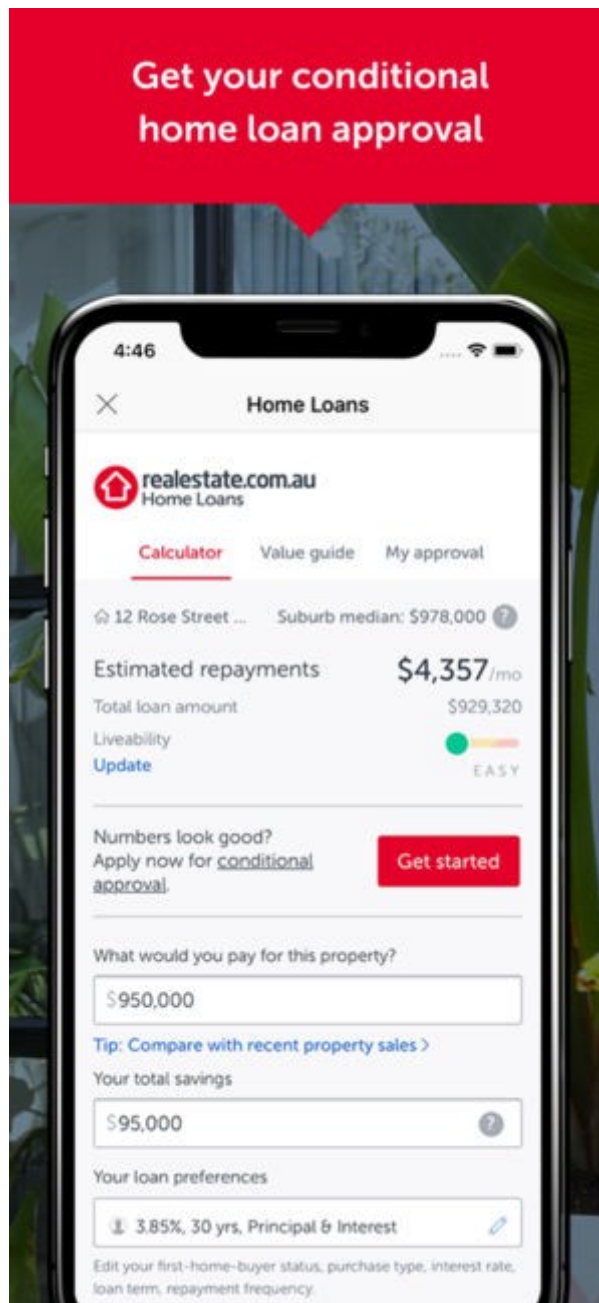
Enterprises can be more like Startups with high velocity deployment. They can launch a new service, a new business line or even a new startup to fend off other startups and incumbents.

A McKinsey article discusses how fostering a culture of experimentation is vital to keep large enterprises going in today's fast moving world. Microservices is the enabler of experimentation.

For example, fast moving realestate.com.au has started the shift from monolithic to microservices architecture since 2014. They have had gradual replacement of key services such as the “sold” section and the buyer's section to better serve customers' needs.

Moreover, the nimble infrastructure has allowed REA group (operator of realestate.com.au) to release a world-first innovation in partnership with NAB. The idea was to “bring property search and finance together in a single platform.” This kind of integration would have been problematic if both sides had a cumbersome ERP to navigate.

Instead, “the functionality is mostly enabled using microservices that run on AWS infrastructure.”



Users searching on realestate.com.au now see the home loan add-on

Take risks and disrupt without causing chaos

Startups naturally have a higher propensity for risk taking as they have less to lose than their enterprise counterparts. However, high-performing corporates are twice as likely to report being risk-seeking compared to low performers.

So smart disruptors and innovators are always looking for ways to solve their customers' problems faster and better. To wrap this up, let's re-

examine how they are taking advantage of new technology and view enterprise IT differently.

Choose your investment wisely

The adage is “every company is now a technology company,” yet “sometimes the way firms manage IT spend is the problem.”

Despite fear of disruption, most enterprise innovation budget goes to continuous improvements of existing processes and products, i.e. the small stuff.

So putting in an industry-standard ERP system is not exactly going to help you disrupt the market as it is mostly about efficiency. The irony is, ERP system is not “the small stuff” in size.

Meanwhile, high-performing companies tend to invest more in empowering and disruptive innovation. Cloud and microservices architecture are enablers of this, as well as the efficiency innovations that most enterprises focus on.

Leaders are already moving fast into the new software paradigm: small stuff in size, yet big in impact. Industrial giant GE (mentioned at the start of this article) transformed itself into a digital organisation, stayed on top of progress in cloud and microservices architecture, and ranked 18th in BCG’s annual list of the most innovative companies.

Why have trade-offs?

High performing companies don’t do trade-offs between speed and performance.

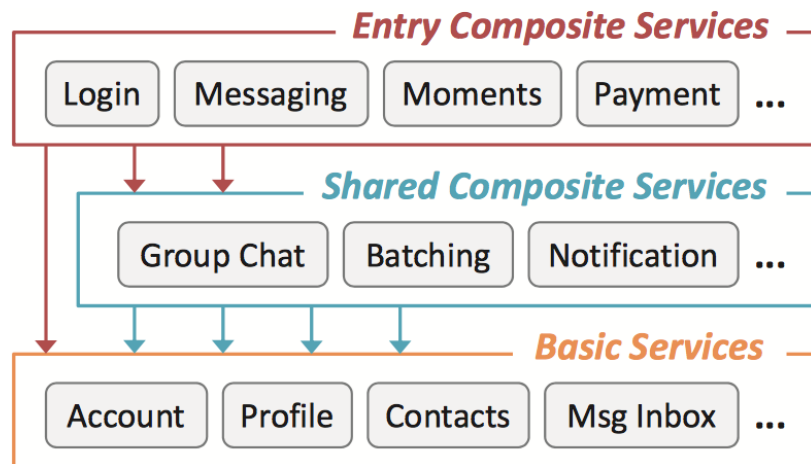
Toyota didn’t win by making cars faster, they won by making high quality cars faster. In a sense, making software shares some similarities to manufacturing, as lean methodologies can be applied.

If interested, you can listen to the rest of the debate in this podcast by venture firm a16z.

While companies are spending all their efforts, time, and money trying to roll out and customise their ERP, the competitors are focusing on new offerings that meet customer expectations. In fact, 60% of companies take a year or longer to create new products. Within that time, new startups or competitors could crop up.

Let's look at China's equivalent of Facebook—WeChat—which runs on 2K+ microservices. They are constantly innovating, with impressive growth and a portfolio of services. WeChat users basically manage all the necessary aspects of their lives through the app: from messaging, to payments, shopping, and hailing taxis, etc.

To have such scale without compromising on reliability and agility, their microservices architecture plays a crucial role.



WeChat microservices architecture

Make something unique

Tying this back to competitive advantage, if companies are all using the industry standard ERP or any enterprise-grade system, they have a point of parity and rarely a point of difference.

Since we've talked about ERP a lot throughout this article, it is worth reiterating that ERP is not a piece of software. It is an activity or objective (hence the name). Thus, even without a single monolithic solution, businesses still want to conduct ERP activities.

As mentioned above, Gartner predicted the death of the ERP concept and introduced ERP II, not as another monolith, but the same activity now with a different strategy, which is basically a "loosely coupled and federated microservices ERP environment."

This environment gives enterprises the chance to make something powerful by combining best of breed SaaS products with their own custom microservices, all configured to their own unique business model and process, while still remaining agile enough to adjust according to customers' demands.

Now, take that Silicon Valley startups!

. . .

Written by:

Grant Hull, CEO and Director at Enabled

Apart from guiding the business, Grant is also a popular presenter and speaker in the digital space. He has delivered innovation courses at several Australian universities and masterclasses to executive groups. Follow Grant on Twitter and LinkedIn

. . .

Originally published on Enabled's blog.

