

DIGITAL TRANSFORMATION IN BANKING AND FINANCE:

The How and Why

CONTENTS:

1. NEW AGE – NEW POSSIBILITIES

1.1. Digital Transformation in Banking and Financial Services.....	4
1.2. Banks and Fintech Startups: Why Invest?.....	6
1.3. Banking-as-a-Service: Earn, Evolve, Embrace	7

2. WHAT YOU NEED TO DO BEFORE BUILDING A FINTECH SOLUTION

2.1. Study the target country	8
2.2. Find your team.....	10
2.3. Include customer development in your process	12
2.4. Find the development/marketing balance	14
2.5. Choose your technology	14

3. WHAT TO CONSIDER WHEN DEVELOPING A FINANCIAL SOLUTION

3.1. Find Your Audience.....	16
3.2. Build an MVP.....	17
3.3. Define the Architecture	19
3.4. Figure Out Third-Party Integrations/Bank APIs	21
3.5. Log Your Codes	22
3.4. Test Your Sandbox.....	23
3.5. Work On Your UI/UX.....	24
3.6. Find Your Flow	25
3.7. Make It Fun	26
3.8. Conclude	29

We are all busy living our lives, and we rarely notice change, especially change that comes gradually. And then, suddenly, your favorite Berlin brunch cafe goes cashless. In 2018, for the first time in German history, cards outweighed cash, accounting for €208 billion in transactions.

The country hasn't reached the cashless levels of neighbouring Holland, Sweden, or Finland, but if the Germans have made this step towards a cashless future, imagine the giant leaps the rest of the world is making.

Retailers like Honda, for example, have introduced mobile pay options that allow car owners to pay for gas using their own car's dashboard. Under pressure from mobile banking apps and money transfer services

like PayPal, bank branches disappear, and mobile banks like N26 take their place.

African countries like Liberia, Kenya or Rwanda are gaining momentum in the use of mobile banking, as their banking infrastructure remains extremely underdeveloped. African emigrants send money to their loved ones back home via mobile banking applications, and African locals use their phones to pay for electricity, water, and cable television bills. Obviously, such broad markets open up endless possibilities for modern fintech.

Besides, fintech has another big advantage over traditional banking: *technology adapts a lot faster and more precisely to the needs of the modern world.*

NEW AGE - NEW POSSIBILITIES

The digital age has long since conquered the world, and the younger generation is propagating the new lifestyle. Millennials are the ones building the future as they gloriously master contactless payments, use online banking, and all kinds of digital financing services. In an effort to *avoid old-school bureaucracy*, they build their lives in a way that no generation before them did. They mature, they start their own businesses — and change the world as they go.

DIGITAL TRANSFORMATION IN BANKING AND FINANCIAL SERVICES

Too many financial services providers waited for the millennials to grow up and become part of the real, adult world. Basically, they waited for the younger generations to adapt to the way they were catering their clients' needs. However, it turned out that the truly right way was the complete opposite — banks needed to adapt to the new millennial ways in order to stay afloat in the new era of the financial industry.

For instance, take the mortgage market. Traditional lenders who don't accept online technology and for some reason won't become more customer-centric are losing their market share. According to a report compiled by Accenture, *by 2020 they could lose up to 35% of their market share to lenders who embrace the new business models*. Thus, taking the younger generations into consideration is not only necessary in terms of customer development, but also for growing business and increasing profit.

According to the [Deloitte 2019 Millennial Survey](#), the gig economy appeals to four in five millennials and Gen Zs, which means that they would rather freelance and do contract work. This choice is mainly dictated by an aspiration to earn more money — which is exactly why the millennial generation chooses online financial services to:

- have full access and full control of their money, anywhere, anytime;
- enjoy financial freedom;
- ▲ not be confined to places or people, financially or otherwise.

On top of this, millennials behave in ways different from earlier-generation customers. Of course, consumers of all ages are susceptible to advertising and dislike poor service or faulty products. But millennials and Gen Zs tend to start and cut off relationships with businesses for far more personal reasons such as the company's impact on society, or their stand on ecological and/or ethical issues.

However, let's not forget about the Baby Boomers — by far the wealthiest generation right now, again, according to [Deloitte](#). No wonder they outspend younger generations. In attempts to cater to this target audience, startups are finding ways to provide services in estate planning or fraud prevention, for example. In addition, retirees are using their newfound freedom for travelling and wellness, which is also a part of fintech. They might be not as advanced in their technical skills, but they have their own insights and needs that need to be addressed. And online financial services are much easier for them to handle than going to an actual bank and waiting in line.

“What was surprising to us was how interested the older generation is in our proposition as well. Seventy-year-olds, and so on. They wanted to get a mortgage at Molo, primarily because the process is so simple. They might ask for some help, but they want to do this, they want to be in control of their mortgage process, and not be dictated by somebody in a bank, a broker, etc”

Amitabh Ghatak,
Chief Product Officer at Molo Finance

Knowing all these facets of the target audience, be it millennials or their parents, will be vital to building any fintech product. Using the customer development approach — through interviews, A/B testing, etc. — you can find out your users' behavior, habits, and the decision-making process you should use to alter your product according to their needs. We will go into more detail on this further below.

BANKS AND FINTECH STARTUPS: WHY INVEST?

If you take a closer look at the changes in the financial sector, you'll see that banks are not only transforming; they are also investing money into fintech startups. Why?

- ① It's a great investment and development opportunity.
- ② It's a stepping-stone for representatives of traditional institutions who become board members of the most innovative businesses.
- ③ This way, traditional and financially stable organizations are familiar with how new financial technologies work, which can only be good for their growth.

globally there are

48 FINTECH UNICORNS

valued, in aggregate, at

\$ 187 BILLION

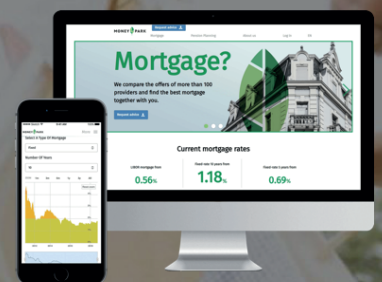
© CB insights



An example we're particularly fond of is the acquisition of our client MoneyPark by Helvetia, a large insurance company. This acquisition allowed them both to combine the newest technologies of the former and the connections and possibilities in insurance of the latter. Today, they offer a unique set of services as an independent mortgage broker to the most demanding customers.



The first independent provider of personalized financial advice on mortgage and insurance products in Switzerland.



BANKING-AS-A-SERVICE: EARN, EVOLVE, EMBRACE

Providers of financial services quickly realized the value of these new technologies. It's not only startups that offer online services; more and more companies have started to add banking services into their existing apps and products.

In Q1, new product lines primarily consisting of BaaS programs contributed over 420,000 new active accounts, all from product lines which largely didn't even exist just two years ago,"

*says Steve Streit,
Founder & CEO at Green Dot*

Aside from all of the profit companies can get from doing business in both the online and the brick-and-mortar markets, technologies and transformation are the best ways to scale a business and embrace more business opportunities. That's why Banking-as-a-Service is such a big hit right now.

WHAT YOU NEED TO DO BEFORE BUILDING A FINTECH SOLUTION

Building a successful fintech solution isn't just about being good at technology. It's about team efficiency, tools, planning, and many other factors. Some of these factors include:

STUDY THE TARGET COUNTRY

One way or another, new fintech products will have to comply with the existing rules and legislation of the target country. After all, governments regulate all financial bodies and services except cryptocurrencies. You cannot predict all the challenges you might encounter, but learning about domestic regulations will minimize your losses. For instance, some countries have launched a *regulatory sandbox* in their financial markets. This means they have produced a framework to provide small-scale testing of innovative technologies in a controlled environment.

A sandbox is a structured and controlled environment that allows governments to formulate regulations at a pace fast enough to leave room for companies to develop new technology. Commonly, regulatory sandboxes are used in areas such as customer databases, identity-verification procedures, payment systems, the tracking of physical and digital assets, and transaction recording.

Also, there's another factor that has less to do with technology, but with convenience. Startups should take into consideration the country's *time zone and traditional local working hours* so they can organize transactions and customer support that will work for the target audience.

FIND YOUR TEAM

No matter how good your idea is, it's not worth anything without a great team that can bring the idea to life. And you probably won't be able to do this on your own. You need to find a team that will:

- understand you;
- understand the market;
- ▲ understand the product.

To develop a great fintech product, you don't just need great developers, but great developers that understand finance. Thus, pay attention to the experiences of the people you hire, and make sure they understand financial jargon. Experienced specialists are less likely to make mistakes that can have huge consequences in an area such as finance.

Building a successful product is possible with an outsourced team, just as it is with an in-house team. But how do you decide which one to choose?

- **IN-HOUSE TEAMS** are usually associated with startups. In this case, their competencies should cover all the major and project-specific issues that might arise. In-house teams have only one product to focus on and only one priority. For products that require long-term development, this is the key factor. On the other hand, the priorities of an in-house team can be changed, even within a sprint if necessary, and problems can be fixed more quickly. An in-house team is especially valuable if your product is your unique selling proposition.
- If the technology you're developing is only a medium for delivering a service — like online-stores or mortgages — you might consider **OUTSOURCING**. This can be a more practical choice if your company is actually non-technical and you need development services. You can outsource until your venture generates revenue and even during scaling.

Reasons to outsource

Focus on core

Improve quality

Increase efficiency

Foster innovation

Conserve capital

Reduce costs

Saving of valuable time

Last but not least, get to know your potential candidates on a more personal level to see how they interact with the other team members – whether they're in-house or outsource. No matter how good they are, if they can't trust others (or can't get others to trust them), working with them will be hard. A tightly-knit team will care more about the product and help you with support and maintenance after the launch, because it's their product too.

INCLUDE CUSTOMER DEVELOPMENT IN YOUR PROCESS.

Ever heard of Everpix or Google Wave? The first one was supposed to be an app that organizes photographs on your phone. Google Wave was planned to be a communication platform. However, neither was successful, since users couldn't figure out what the app was or how to use it. They just didn't see value in the products. What was missed here — despite the utmost professionalism of the creators — was the product/market fit. In this case, it didn't matter whether the product was created by two fellow developers or a technology giants such as Google, or how much money was invested. What could have saved the developers from failure was proper customer development.

Customer development is defined as a process of obtaining customer insights that can be used to create, review, and optimize the ideas in product development.

The most important aspect of custdev is that it helps you find customers and create a product and messaging specifically for them.

With Everpix and Google Wave, it seems that the developers just came together and started working, keeping their process a secret. And this was exactly where

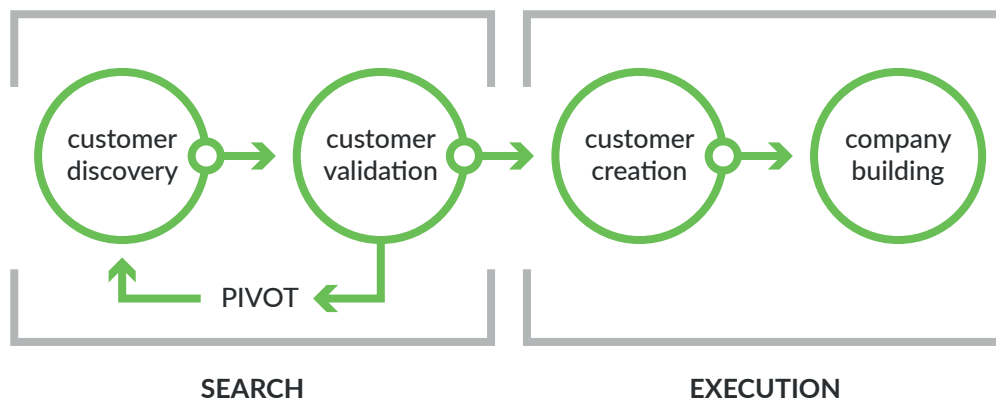
they went wrong — *they should have gone outside the building*. Not getting any feedback from the real world prevented them from realizing whether they were on the right track. And when they did get feedback — most probably after a hyped launch — it turned out that only a small percentage of users had stuck around and the product hadn't made its target sales.

The most important aspect of customer development is that it *helps a company find customers and create a product specifically for them*. Customer development can be introduced at any stage of creation, as its strategies don't require huge resources or interfere with other processes.

Custdev basically consists of five major elements:

- ① **CUSTOMER DISCOVERY** allows you to find out whether your audience accepts your product and helps you test your hypothesis.
- ② **CUSTOMER VALIDATION** is the phase when you test the data you gathered during the customer discovery phase in a real environment. It allows you to make sure you've understood your customers correctly, and that you're developing proper corporate and product positioning.
- ③ **CUSTOMER CREATION** is when you reach out to a larger base of customers. Now that you've discovered and validated them, you have a better understanding of who the people using your product are and where to look for them.
- ④ **COMPANY-BUILDING** is the final stage. This is when your startup turns into a full-scale company. The focus shifts from research and discovery to long-term planning and mission development.
- ⑤ **CUSTOMER DEVELOPMENT TOOLS** help you extract the most value from your customer development efforts. When you're finally out of the

building, you can use customer interviews, experiments, A/B testing, and applications/technical support to gather insights from customers.



This section brings us back to the first part of this abstract — the target country. Customer development will help you learn more about the people who are supposed to use your product and provide country-specific insights.

FIND THE DEVELOPMENT/MARKETING BALANCE

The most important thing here is to get feedback with every single new release. No matter if it's good or bad, feedback is the key. The truth is, people don't expect a technologically perfect product that will solve all their problems. Which means, in the early stages, your product should be just good enough. Spending too much on unreachable technological goals and underfunding marketing will lead to zero ROI. A perfect product isn't worth much if people aren't aware of its existence and features.

CHOOSE YOUR TECHNOLOGY

Fintech isn't unique in its complexity per se. However, it has unique challenges that make it more exciting to work with. There are state-level regulations, integrations with different services and institutions, bank API connections, etc. to deal with. On top of this, fintech companies require a high level of trust from

the end users, so they'll be allowed to handle people's finances, mortgages, investments and such. These, in turn, require:

- the highest level of security;
- functionality;
- ▲ correspondence with requirements.

What we learn from this is that the more unique and complex the technology is, the harder it is to crack, and the higher it's valued. Without properly working and trustworthy software, any financial venture will die down and lose worth. Financial technology should last in order to be trusted.

To create a technology robust enough to withstand the stress of worldwide financial perturbations, but flexible enough to follow all the new changes and customers' needs, we chose to work with the Django framework within Python. Even years later, we're still discovering Python's hidden possibilities. We're not saying this language is the ultimate cure, but we definitely know its perks when it comes to financial technology — and we use them for the benefit of our clients.

- The Python/Django stack lets you build an MVP pretty quickly, which means it helps you find your product/market fit faster. Thus, it gets you to market a lot quicker, because you can avoid wasting your time on developing small things like authorization or user management tools from scratch; instead, you simply take them from Python libraries.
- Python is the language of mathematicians and economists, and fintech obviously can't exist without these two groups who most of the time use Python to calculate their algorithms and formulas.

- ▲ Python has simple syntax and open libraries.
- ◆ Since it continuously develops, finding Python specialists won't become a problem anytime soon.

WHAT YOU NEED TO CONSIDER WHEN DEVELOPING A FINANCIAL SOLUTION

Let's start from the very beginning, as good planning is halfway to success. The better you lay out your working process, clear away any possible misunderstandings, set goals and make sure everyone understands and shares them, the smoother the process will be. So, don't underestimate the pre-development phase.

FIND YOUR AUDIENCE

First of all, **DEFINE YOUR END USER(S)**. It can be individuals or companies, or both. Thus, define their roles and create an image of your customer. Once you do this, you will be able to answer the following questions:

- ✓ Who are we doing this for?
- ✓ How do users interact with each other?

As soon as you find out who will use your product, you'll be able to see how they will use it. Create a customer journey map, highlight the stages of the customers' use of the product, and predict possible integrations with third-party services. Based on the user flow, you'll see your users' pain points and figure out how to heal them.

If you're working with remote teams, organize the work with your international co-workers. Consider working hours, weekends and holidays.

Customer journey map helps designers identify customer emotions

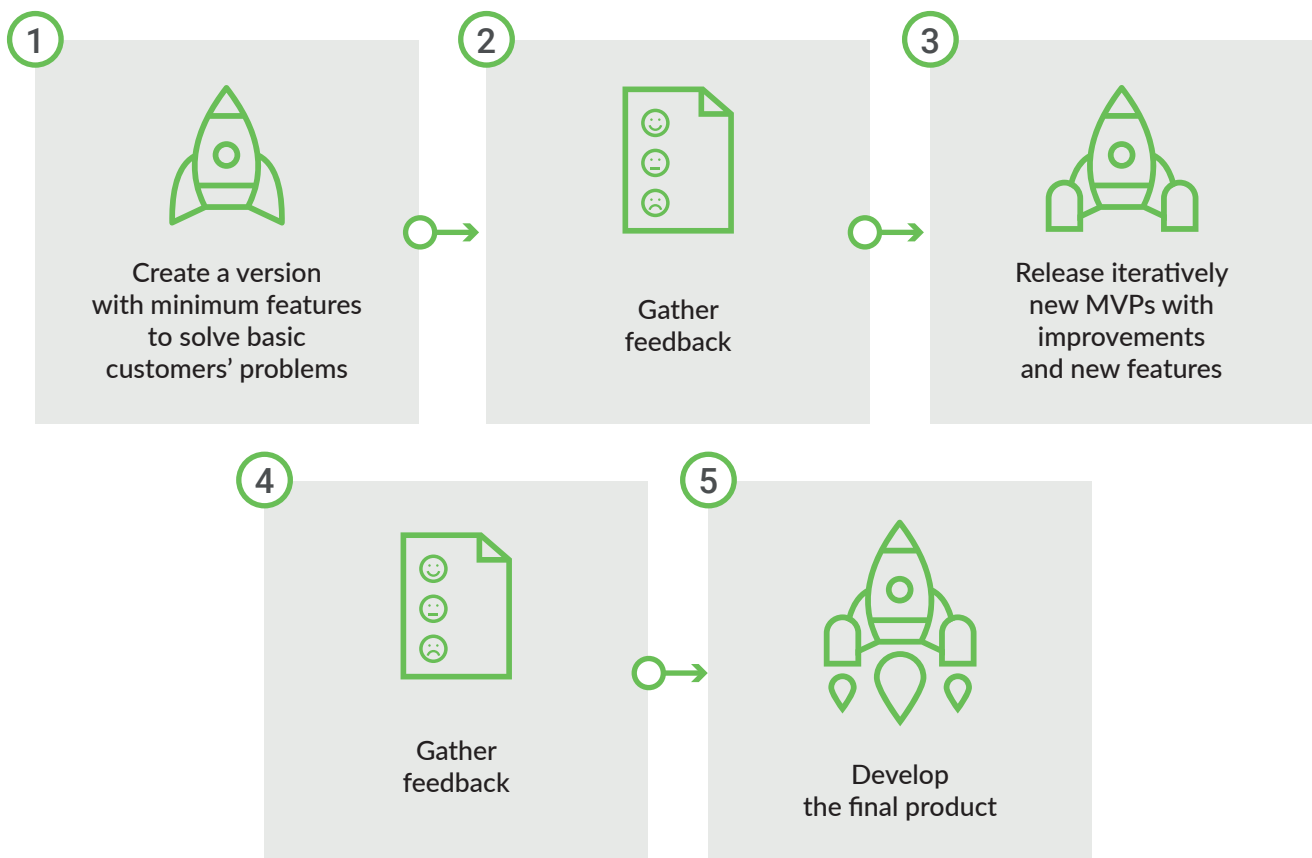


BUILD AN MVP

We've already talked about customer development and its importance to the development process. In the early stages of development, you can gather insights from customers using an MVP — a minimum viable product. Well, to be fair, it's not a product yet, it's just an early version with the smallest set of features that will fulfill your customers' needs. An MVP is created using minimal resources simply to validate economic viability.

If you see your MVP still needs a lot of work, you can pivot back to the customer discovery phase and start all over again without the need for a large budget. Thus, by creating a simpler (and cheaper) version of the product, you can save a lot of money and resources in the future. But if the MVP is well received, you can finish your product and release a full version.

Release New MVPs Iteratively



Choosing a programming language for a startup isn't easy, considering the number of options out there. Each option serves different purposes and offers different performance. As for Python, it's simple and allows you to build an MVP quite quickly. It's especially important for a young company, as it allows the firm to:

- easily implement new features;
- attract investments;
- ▲ test hypotheses.

The important thing is that Python increases time-to-market. But, in addition to its MVP-related perks, Python also offers a number of other advantages.

By the way, as for our MVP creation process, we don't build complex microservices at first. We write code in a way that is easy to replace when necessary. When needed, we change it to a microservice architecture after testing the MVP.

DEFINE THE ARCHITECTURE

Architecture is a tricky thing. You can't reuse the same approach that you've already used on another product. Building an architecture for every new product is a unique experience, and it may sound like it doesn't get easier over time. But, in our experience, the more projects you do and the broader your expertise, the easier it gets to find the best solutions for each and every individual case. It also gets easier to mitigate risks.

Be yourself; everyone else is already taken.

Oscar Wilde

This applies not only to human beings, but to businesses as well. *Don't try to build your architecture around someone else's cool technology.* Rather, think about what makes this technology so good, and what the best solution is for your business.

Before you start working on your architecture, *make sure your tech team fully understands your business objectives* and your strategy. Everyone has to be on the same page and know exactly where the project is going. The thing is, architecture might seem minor in the beginning, but it should be able to change and expand over time. This means you have to develop good chemistry with your team and keep them updated on any changes that your business might undergo, so that architecture and development flow always remain aligned with your business vision.

When building a tech stack, you should *not only consider its possibilities, but also its limitations*. And make the best of it. One of the limitations you may encounter is that not everything you might need to make it work is accessible free of charge.

For instance, for one project we needed to use the MongoDB, which has a free database. But as we dug deeper, we realized that their new licensing doesn't allow free usage if the product you're developing provides MongoDB as a service. For cases like this, you should use the enterprise edition. Things like this don't affect the development itself, but *unexpected expenses can affect your business*.

Another thing to consider at the very beginning is where to keep the product, and *which platform to use*. Starting with Amazon right away isn't the best solution if you don't have any clients yet, i.e. if your business isn't generating revenue yet. Even if your website or product isn't open to the public, you'll still pay for AWS, sometimes up to thousands of dollars.

Usually, when you're just starting out, you don't need an extra-powerful database. [You're not Google](#), so you don't have that capacity yet. A small database will suffice in the beginning, and when your business grows and you feel like it's time to expand, you can move to something more powerful (like Amazon

AWS). When you actually have paying customers, shelling out for Amazon services makes a lot more sense.

Working with Amazon, we've gathered a couple of insights that might affect your choice of database:

- With Amazon's vendor lock-in, Amazon makes it hard to move to other providers. Amazon provides cloud services that other providers don't have, and if you decide to move to a different provider, you'll have to rewrite the code for each service.
- You can't predict Amazon's monthly costs, as you pay for time of usage. So, if one month you have 10 customers, you don't pay much; but when you have thousands of customers, Amazon's fee adds up to a substantial amount.
- ◆ Using Amazon, you don't have to worry *about backups*. *It's easier on the infrastructure, but harder on the budget, as it can get quite expensive.*

FIGURE OUT THIRD-PARTY INTEGRATIONS/BANK APIS

Integrations are a vital part of any fintech product. Most integrated *third-party services* are banks that provide user information such as IDs, customer emails, credit records and so on to platforms upon request. Based on the data, every user gets a score that serves as a basis for investment suggestions. Besides, a secure bank integration lets you integrate exchange rates into your service. Usually, there's a fixed fee per request — from \$1 up to \$10. The trick is to create a flow that allows for as few requests as possible to avoid extra expenses.

One way to *reduce the number of paid requests* is to have users finish the registration process, i.e., go through all registration phases only if they really want to follow through with the service. (Like when they're getting a mortgage estimate, for instance.)

By the way, if we take online mortgages, it often happens that many people are checking out the same house, which means that there are a lot of identical requests. With a well-built integration, you can send only one request instead of several. Also, you can build a custom integration logic and have the backend “decide” where to send requests. For instance, you can work with several services that help you check a user's credit score, and depending on the situation, you can send your request to the one that's most appropriate (cheaper/faster/etc.) at the moment.

When it comes to *bank APIs*, make sure you've tested the payment system before releasing the product. So that developers can avoid using their own credit cards, *banks provide test data*. This enables developers to check the various outcomes of using payment systems. What happens when a card has expired, or is stolen or lost? What happens when a user logs in invalid data? There are many scenarios that can happen, and you need to test as many of them as possible to avoid big and costly flow changes in future.

LOG YOUR CODES

It's only logical to talk about code logging at this point, since with third-party integration it is vital to have proper code error logging. On top of that, logging helps you see:

- the reasons behind errors;
- how you can prevent them in the future.

Once, we encountered a problem with a product: the format of a third-party API changed, and transactions weren't going through anymore. For a long time, it wasn't clear what the actual problem was, because errors weren't shown appropriately. Inspired by this incident, we created an error-logging tool that would log both all of our enquiries to the third party and their responses to us. After that, every error that arose was shown in the log list. Now we use [Sentry](#) for code logging.

TEST YOUR SANDBOX

Financial information is a highly sensitive matter. When choosing a third party to integrate with, check whether they offer a sandbox option that will allow you to develop your code without any risk of accidentally compromising user information. Experiment as much as you need. In a bank integration, a sandbox will also help you avoid unwanted charges for using their information.

A sandbox environment is essentially a copy of the third-party function that's connected to your code, but without any risk to customer data. You don't have to be afraid of any extra moves when working in a sandbox. When you're done coding, you replace the sandbox URL with a real one, and you're ready to go.

Note that not all sandbox functions are the same. For instance, *Experian, which is used to check credit scores, clears its data every 24 hours.* On one hand, it's good for security reasons — sensitive information is less likely to get into wrong hands. But on the other hand, you have to be quick, as in the morning the sandbox will be cleared of anything you were working on the day before.

If you're offering SaaS, you will need to build your own sandbox. This one doesn't work through integrations, but the logic is similar.

WORK ON YOUR UI/UX

In the age of social media and gadgets, it's vital for any digital product to have a design that catches the eye of a user, yet doesn't offend it. Since fintech deals with numbers and formulas — which already seem like rocket science to most people — fintech applications should be as simple as possible.

COPY AND MICROCOPY. Copy and microcopy includes all the text in your product. To users, this is the most complicated part of the product. Your job is to make every text clear and straightforward for everyone, no matter their financial and/or educational background. However, some financial terms may still need clarification. So think of a simple way to explain difficult terms without overburdening the design. Here's how we did it for [Moneypark](#):

The screenshot shows a web interface for a mortgage calculator. At the top, there's a small text input field for 'mortgage amount and the location of the property.' and a blue button labeled 'Calculate your personalized rates'. Below this, the main content is divided into two columns. The left column is titled 'How much rent do you pay per month?' and features a slider with a green handle. The slider has a label 'CHF 1'500' at the left end. Below the slider, it says 'You can afford: CHF 300'000' in green, followed by 'Based on CHF 60'000 own funds'. There's a 'Learn more' link with a magnifying glass icon. At the bottom of this column is a blue button labeled 'Request advice' with a person icon. The right column is titled 'Our calculators' and lists seven options: 'Mortgage calculator', 'LTV calculator', 'Purchase price calculator', 'Buy or rent calculator', 'Affordability calculator', 'Refinancing calculator', and 'Amortization calculator'. Each option has an information icon (i). The 'Affordability calculator' is highlighted with a green background and contains the text 'Learn whether your own funds are sufficient for buying a property.' At the bottom right of the interface, there's a Windows activation notice in Russian: 'Активация Windows. Чтобы активировать Windows, перейдите в раздел "Параметры".'

Source: moneypark.ch

To ensure all your team members understand the terms they're working with, compile a glossary.

FIND YOUR FLOW

As we know, most financial services are complex and use large amounts of information. Which is why it's important to keep fintech development as simple and transparent as possible. Try to keep it smooth and not overwhelm your users with tasks. Make the structure of your application clear.

For instance, you can divide the registration process into two phases, like our client Molo Finance did.

GET A QUICK DECISION IN PRINCIPLE

A Decision in Principle indicates how much you might be able to borrow, pending a full credit check and valuation of the property. We only perform a soft credit check at this stage, and the application should take no longer than a few minutes.

Which are you applying as?

☒ JUST ME ☐ JOINT APPLICANTS ☐ LIMITED COMPANY

What type of loan are you looking for?

☒ PURCHASE ☐ REMORTGAGE ☐ REMORTGAGE & BORROW MORE

What is your annual income before tax?

€ 30,000

How much is the property worth?

€ 250,000

What is the expected monthly rent?

€ 1,200

How much do you want to borrow? (Maximum: £200,000)

€ 150,000

Your loan-to-value (LTV) is 60%

[Chat](#)

YOUR LOAN: £150,000

Your loan-to-value (LTV) is 60%

We show you products that best match your needs based on the options selected above. You don't need to choose a product yet. Our rates are based on LTVs of up to 65%, 75% and 80%. All our loans are interest-only.

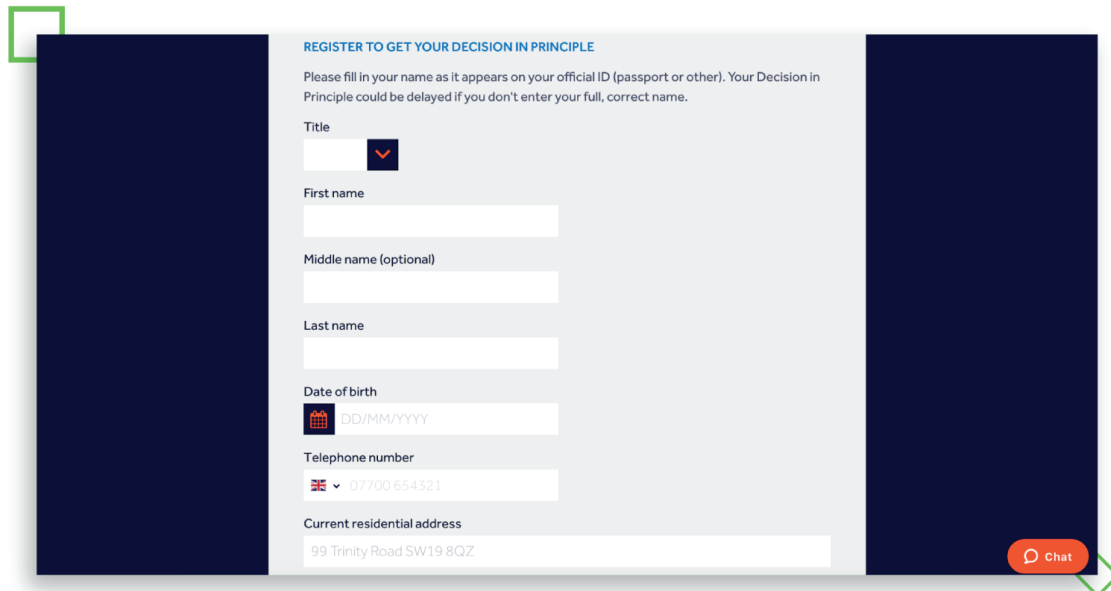
2 year fixed rate - 65%	1.80%
Product fee: £450	Show Example
Initial monthly payments: £225.00	
3 year fixed rate - 65%	2.50%
Product fee: £450	Show Example
Initial monthly payments: £312.50	
5 year fixed rate - 65%	2.57%
Product fee: £450	Show Example
Initial monthly payments: £321.25	
2 year tracker rate - 65%	1.80%
Product fee: £450	Show Example
Initial monthly payments: £225.00	

* This indicates how much you might be able to borrow.

[CONTINUE](#)

[Chat](#)

Source: molofinance.com



REGISTER TO GET YOUR DECISION IN PRINCIPLE

Please fill in your name as it appears on your official ID (passport or other). Your Decision in Principle could be delayed if you don't enter your full, correct name.

Title

First name

Middle name (optional)

Last name

Date of birth

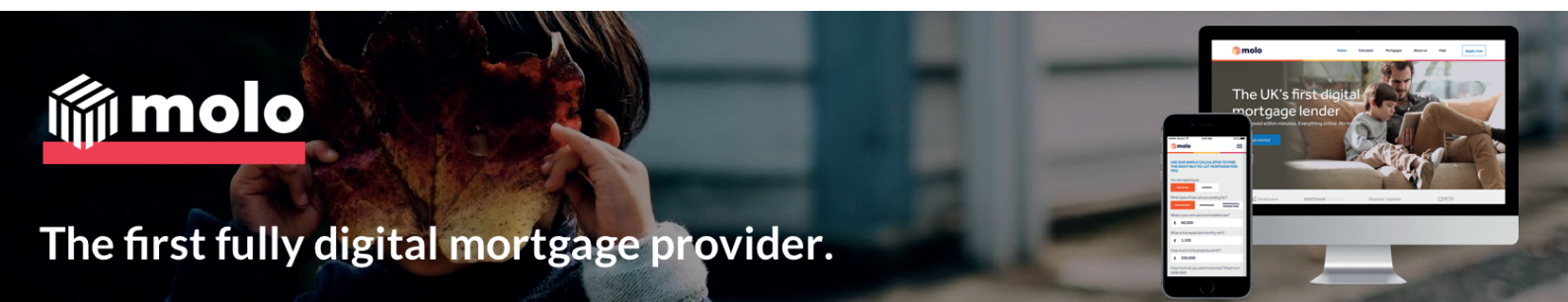
Telephone number

Current residential address

Chat

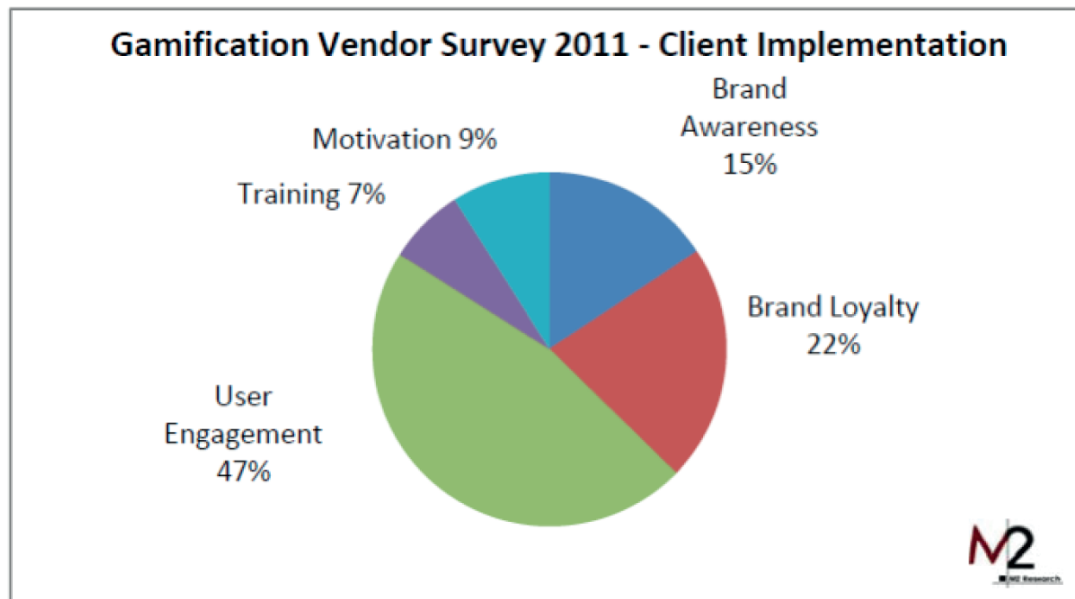
Source: molofinance.com

In the first phase, only ask for general information that is necessary to provide basic services and offer the customer a draft. If they like what they see, in the next phase they can fill in more personal information to get a more precise result, altered according to their details.



MAKE IT FUN

Unlike education and healthcare, fintech is still sceptical about gamification. However, the gamification market seems to be a promising area that's expected to grow to an astonishing \$19 billion by 2023.



Source: M2 Research

Gamification is a way to engage audiences by applying gameplay principles and game design elements in a non-game environment. Game mechanics can solve different problems for users — like reduce workplace stress and boost productivity — and hereby help businesses reach their business goals.

What gaming elements can be introduced into banking?

- ① **CONTESTS AND REWARDS.** Invite your users to participate in a referral program. Offer rewards for both existing users and invitees, like Airbnb does.
- ② **POINTS FOR SHOPPING.** Offer your customers points for shopping with retailers you partner with. Users can earn points they will be able to exchange for gift cards or other presents. Think of a grand prize for the most active players – a car, a trip, etc.
- ③ **SIMPLICITY AND FUN.** Remember that the game process should be simple and fun, and not require too many actions (such as going places and filling out forms). If the gaming is too complex, your customers will soon lose interest.

- ④ **SOCIAL MEDIA.** Let your customers share their accomplishments or any features they like with their friends and followers on social media. This way, you're not only engaging your customers, but also getting them to spread the word about your product or service.
- ⑤ **EDUCATION.** Use gamification to educate your customers. Get them watch videos on certain topics and offer them points for doing it. With the help of these videos, you can raise awareness about financial matters and help your customers make smarter decisions about their money.
- ⑥ **NEW SERVICES.** Introduce your new services through gameplay. For instance, complicated financial topics can be made more understandable if you show your customers short videos in which the main character uses your product or service to solve a problem similar to the one your user has.

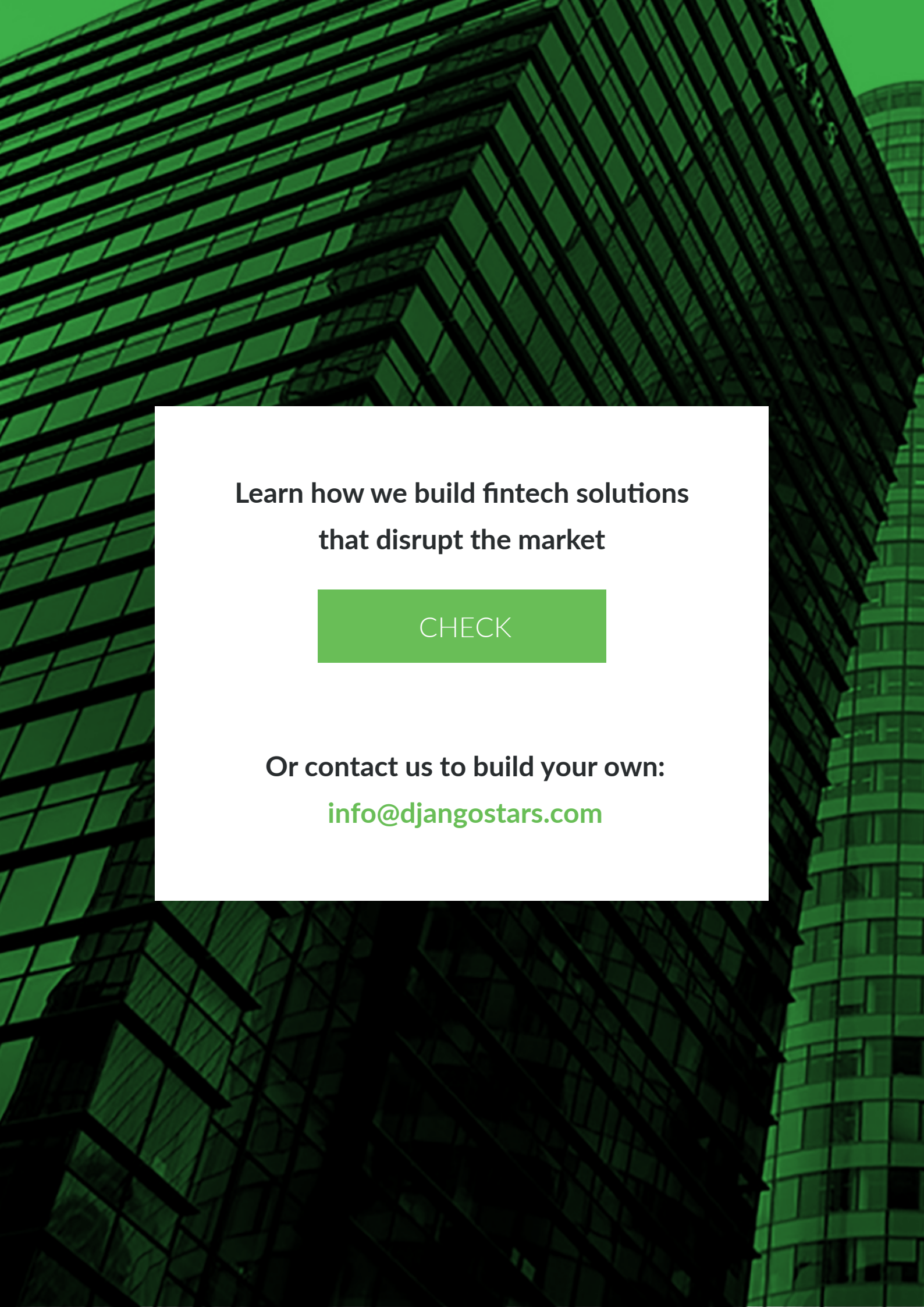
CONCLUSION

Every day, the world moves from traditional practices to digital alternatives, faster and faster. Some undergo this transformation because of their eagerness to bring the future closer, while some do it out of necessity and to keep up with the visionaries. But *the process can't be stopped*.

The world is changing faster than it used to, and even if it's physically impossible, it feels as if the Earth has started spinning at a faster pace. Apparently, the only way to keep up is to move everything to digital. After all, digital can *adapt better and faster to the new lifestyles* – like those of digital nomads, travelling business people, remote development teams

and many others. And even though different people have different needs, one thing we all have in common is that *we want our money to be safe, and we want to have sole control over it*.

As mentioned at the very beginning of this book, digital transformations in banking and finance have already started in different corners of the world. More and more people will recognize the convenience and security of digital banking services, and the number of users of financial technologies will grow. Which, in turn, means that fintech has to gear up, expand, and grow as well to offer everyone the services and the products they want and need.



Learn how we build fintech solutions
that disrupt the market

CHECK

Or contact us to build your own:
info@djangostars.com