Patterns

CellPress
OPEN ACCESS

Opinion

# How to Tell When a Digital Technology Is *Not* Ready for You

Jon Crowcroft[1,*]
[1]Department of Computer Science & Technology, University of Cambridge, Cambridge, UK
*Correspondence: jon.crowcroft@cl.cam.ac.uk
https://doi.org/10.1016/j.patter.2020.100001

The stages of digital technology readiness are viewed through the lens of three contemporary and widely discussed examples, namely distributed ledger technology, machine learning, and the internet of things. I use these examples to clarify when there is really just an old technology being re-branded, when there is something genuinely new and useful, and whether there may be over-claiming.

This Opinion piece is about judging when a digital technology is ready for you, or rather, when it is not and how to make that decision based on your organizational skill and knowledge levels. I will cover the murky worlds of distributed ledger technology, artificial intelligence, and internet of things, evaluating the technology as it stands, and offering a framework for making a judgement call in each case. For each of the three "solutions," I will explore how they live up to the hype, where their strengths and weaknesses lie, and which technical questions you can ask proponents to identify whether it is the right option for you. (We are really distilling the difference between technology readiness levels and hype, as per Gartner's excellent analysis.)[1]

As a computer scientist, I have worked on internet technologies since 1981, nearly their inception, and my experience has shown me how slow technology adoption can be. While the early internet provided a rather poor user experience, the functionality of e-mail and file transfer and remote system access could be enhanced to provide primitive versions of what later became commonplace. Examples of funds transfer (you could buy pizza on the net and have it delivered back in 1980) and sending sophisticated orders (you could send a circuit design to a company, and they would ship you back the product), presages internet banking and Amazon.

Many large organizations in industry and government didn't notice this functionality for a surprisingly long time. In 1992, the research nature of the network was changed. The US government stopped subsidizing the operations of research networks, and several commercial internet service providers (ISPs) started up. In the same year, the first web browser and web servers appeared. Still, nothing much changed except in research-active organizations, until a significant fraction of citizens started to use the internet, which took off with home broadband (and streamed media) and later (early 2000s) with cellular data services and smart phones.

It was only around 2010 that this change was reflected by the publishing behaviors of large companies and government agencies as information (e.g., public data, contact details, goods and services) began to be published and online mechanisms for citizen interactions (e.g., by direct democracy, opinion polling) began to appear. The last large transformation, now nearly 15 years old, was the advent of cloud computing, allowing low-cost processing, which still took nearly a decade to become common place.

There are two running themes that I will revisit in the rest of this Opinion. First, technologies are often around a long time before they are noticed. Second, the form in which they find recognition and adoption is sometimes quite different from their origins. Some people actually think that the World Wide Web *is* the internet. (Some people even think Facebook is the internet.) The slow burn and multi-faceted aspect to new technologies can be confusing, and risk aversion is an entirely reasonable position in this uncertain environment, particularly in sectors that hold sensitive data or have limited ability to invest in new technologies.

I will now explore three "buzzword" technologies and provide recommendations for evaluating their usefulness and practicality in your case.

## Distributed Ledger Technologies

Distributed ledger technologies (DLTs) come in numerous guises, though the blockchain is the most common example. It is important to not mix up cryptocurrency and blockchain. Cryptocurrencies, such as Bitcoin, are built on the blockchain, but you can have a blockchain without a cryptocurrency, and you can build cryptocurrencies that aren't built on a blockchain. The blockchain is a distributed replicated chain of blocks of data constructed and stored as a form of distributed ledger (record) of transactions that have occurred in the world. It makes use of various data structures and consensus algorithms to make it hard to tamper with the record of transactions without being detected. It's key to remember that this isn't a blue-sky technology—it has evolved out of digital currency and payment systems (e.g., PayPal) and digital ledgers for accounting that predate blockchain by decades.

Proponents will often advocate for adoption based on three characteristics: decentralization, immutability, and smart contracts.

Decentralization means there is no single point of trust, as each participant in the network (whether public or private) has an up-to-date copy of the transaction history, providing resilience to faults caused by errors or by adversaries. In the simplest case, so long as a majority of copies of data are the same, we can tolerate a minority being altered or deleted. This, again, is not revolutionary. Replicated databases have existed (e.g.,

in booking systems and banking) for decades, and replication is not unique to DLTs.

The consequence of replication is that a government body or organization may have its system distributed over many other organizations' computers, and it is not yet clear what the consequences of this will be. One notable present-day difficulty is that each transaction, due to both the processes of block creation and of replication, is heavy, placing a burden on transaction speed and latency. For example, the bitcoin network sustains approximately seven transactions per second, where single (replicated) transactional database systems for online sales can support tens of thousands of transactions committed per second.

The second claim, immutability and persistence, is grounded in the challenge of changing a transaction record on a DLT network. An adversary needs to change each of the replicated transaction ledgers and do so without raising suspicion. Critically, what this constitutes is tamper evidence, not immutability. Any change to the ledger gives a warning to each network participant, along with a mechanism to restore (or ignore) the change. This is done in transactional databases (used in audit trailing in financial sector) and has worked for decades at extreme performance.

Finally, a blockchain can implement smart contracts, i.e., the ability to trigger small programs when executing updates to the ledger. Computers are programmable, so this isn't a surprise. The programs on the ledger can be as general as desired. How can we determine if smart contracts are meaningful, in the sense that parties on each side of a contract have a common understanding of what it means, how it can go wrong, and what remediation is available if it does? We can't yet determine this. E-commerce featured smart contracts back in 1981, and developing countries used SMS (text messages) to build similar services in the 1990s.

The following questions may help you determine whether a distributed ledger is fit for your purpose:

- What transaction rate is supported (read/write)? How does that compare with a transactional database product? How does it compare with your current rate?
- What is the latency-per-transaction commit? How does that compare with older technologies? Will it match your users' expectations?
- Does your business need smart contracts, or are third-party arbiters integral to your processes?

For reference and comparison, Oracle Database servers can handle millions of transactions per second on a single system.[2] Visa worldwide credit card servers can cope with up to 50,000 per second, while a fast ledger system today might handle merely hundreds per second, in total, as reported in Vermeulen.[3]

## Machine Learning and Artificial Intelligence

Though fuzzy in reporting, I will maintain the following distinction between machine learning (ML) and artificial intelligence (AI): ML is often better statistics with bigger, faster computing and storage, and its use is encouraged. ML and statistics are already used in business and government to drive decision making. In AI, researchers seek to model and even reproduce human intelligence using machines.

Here I will focus on AI, which as a field is about 40 years old. Progress has been slow, partly because we have little idea what human intelligence consists of and partly because machines are simple compared with animal brains. Early on, some headway was made in simple robotics, vision, and natural language understanding. These advances were not seen as a success in AI, but decades later the core techniques have re-emerged as a massive success for problems in pattern recognition, production optimization, and human-computer interaction through speech.

But what of the latest hype? For today's AI deployers, interpretability is critical. The phrase "black box" is often used, because it can be difficult to determine exactly which inputs and processes result in which outputs. In a corporate or government scenario, this means a result can be hard to explain, making accountability opaque and auditing a challenge. In order to de-risk scenarios where the output of an algorithm is incorrect, we must operationally restrict our choice of ML (or AI) to systems that are explainable. It is also important to consider who is the audience for the explanation. A medical diagnosis system should explain its output to a human doctor, who may translate that explanation for a lay patient, for example.

The following questions are useful when thinking about what an AI might be useful for:[4]

- What's the interpretability model? Can results be reproduced?
- How can we believe/trust the outputs are the right ones?
- How were the training data debiased, reducing bias in the output?
- How can we be sure the system isn't making the same mistakes humans made before?
- How do we have confidence the system will go on giving us useful decision support?

## Internet of Things and Smart X

Here, there is another key distinction: the internet of things (IoT) is a broad term referring to the connection of sensors and actuators to a communications infrastructure, which may or may not lead to devices that embed these sensors and actuators being accessible from the internet. A smart system typically refers to a set of such devices, coordinated through some software system (perhaps in the cloud or in an app on your smart phone) creating a more sophisticated service, for example, constituting a smart home where heating and lighting are adaptive and optimized for power consumption and personal preferences.

Most IoT systems are silos, and the word "internet" is being severely mis-used. If you look at sensors such as CCTV cameras, smart meters, fitness monitors, or even home heating and security systems (some with actuators), these systems live in separate worlds. You cannot connect them together. This may be for good reasons, not just privacy—safety is paramount (car brakes, defibrillators, etc.). However, it may also be for business lock-in reasons, which have no place in today's world.

Data ownership is often central to discussions about IoT networks. A network of devices that can accurately give your location at any time of day, your eating habits, your energy consumption, your bank details, and your biometric data points has great potential when held by either corporations or governments. Legislation like the recent GDPR in Europe

may help in this space by mandating the erasure of expired data, collection of only data that are relevant to the service, and the reduction of operational data collection.

For example, in the case of a coffee maker, a continuous log of production can be replaced by simple statistics that act as a predictor for when maintenance may be required. This data minimization offers benefits to data holders as well as producers, e.g., cheaper network and cloud operational costs and lower energy use as well as (likely) easier legislation compliance.

What fitness-for-purpose questions should you ask about IoT?

- Where are the product liability statements? When stuff breaks, who pays?
- What are the published APIs for me to integrate with other IoT products, so if I want to create a smart home, I don't have to buy everything from one company or depend on their cloud service? This is even more important if I want to create a smart city or smart country.
- What are the software update/support plans 6–10 years from now for any current product? Physical infrastructure has to last decades or longer, while electronic infrastructure has a far shorter lifespan.

## Generic Technology-Readiness Lessons

I'd like to conclude with some general rules of thumb about how to determine when a technology is ready for you: Don't listen to academics—they recommend things massively too early. Don't listen to industry—they are often massively too late. Especially don't listen to consultants—they are frequently massively too expensive.

So who do you listen to? All of the above, with a pinch of salt. Some organizations offer coordinated advice—for example, professional bodies (the Institutution of Engineering and Technology) as well as national academies of science or engineering (e.g., the Royal Society and Royal Academy of Engineering in the UK). Often, they combine all of the resources of the communities above in a timely and useful way. (The author was involved in a recent Royal Society report on privacy-enhancing technologies, which may serve as a useful example.)[5]

Regardless, it's down to you to do your due diligence and make your own decisions. Good luck!

## ACKNOWLEDGMENTS

Thanks are due to colleagues in the Microsoft Cloud Computing Research Center for helpful suggestions, and to the editor.

## REFERENCES

1. Gartner. Gartner Hype Cycle. https://www.gartner.com/en/research/methodologies/gartner-hype-cycle.

2. Hood, D. (2018). Scaling SQL to millions of transactions per second with a single database. Oracle's TimesTen Talk. https://blogs.oracle.com/timesten/scaling-sql-to-millions-of-transactions-per-second-with-a-single-database.

3. Vermeulen, J. (2017). Bitcoin and Ethereum vs Visa and PayPal – Transactions per second. MyBroadband. https://mybroadband.co.za/news/banking/206742-bitcoin-and-ethereum-vs-visa-and-paypal-transactions-per-second.html.

4. Singh, J., Walden, I., Crowcroft, J., and Bacon, J. (2016). Responsibility & Machine Learning: Part of a Process. SSRN. https://doi.org/10.2139/ssrn.2860048.

5. The Royal Society (2019). Protecting privacy in practice: The current use, development and limits of Privacy Enhancing Technologies in data analysis. https://royalsociety.org/topics-policy/projects/privacy-enhancing-technologies/.

## About the Authors

**Jon Crowcroft** is the Marconi Professor of networked systems in the computer laboratory of the University of Cambridge. Prior to that he was professor of networked systems at University College London in the computer science department. He has supervised over 45 PhD students and over 150 masters students. He is a fellow of the Royal Society, a fellow of the Association for Computing Machinery, a fellow of the British Computer Society, a fellow of the Institution for Electrical Engineers, and a fellow of the Royal Academy of Engineering as well as a fellow of the IEEE. Jon's research interests include communications, multimedia, and social systems, especially internet related.