TODA

A Brief Introduction

TODA provides the digital equivalent of a piece of paper. Emails, PDFs, and other documents sometimes seem like pieces of paper, but they aren't. They're more like spoken words. Speaking and writing are very different: given a spoken IOU, you have something of value; given a written IOU, that value is transferable.

Suppose Alice emails Bob a promise of payment, and he accepts it, because he trusts her to honour it. Bob forwards it to Charlie, who must not only trust Alice will honour it but also trust Bob to only forward it once. This chain is only as strong as its weakest link. When it gets to Zelda she must not only trust everyone who came before her, but also believe that the next recipient will share that trust. Email transfers fail in the same way as spoken transfers: they collapse under the weight of cumulative trust.

Written things, on the other hand, can be transferred. Alice hands it to Bob, who hands it to Charlie, and so on. What does Zelda need to believe when it reaches her? Only two things: that Alice signed this specific piece of paper, and that she will honour anything she signed.

This quality of transferability is important. Without it, Alice and Bob need a trusted party to manage the IOU. This might be a mutual friend, a bank, a federation of organizations holding each other accountable, or even a large group of untrusted entities relying on a consensus mechanism to keep them honest. Alice needs them to perform the transfer to Bob, and then Bob needs them to transfer to Charlie, and so on.

This isn't at all the same as a paper-based transfer, where Bob possesses the written IOU, which is its own intrinsic source of truth. Instead the digital IOU is possessed by the trusted party -- in fact they must possess it, because Bob has no way to prove he hasn't transferred it. Information-based things require an extrinsic source of truth, so they can only be possessed by trusted parties.

For awhile our motto was "Nearly as good as paper". It's actually quite a high bar. Paper binds information in a special way. The information on a piece of paper can always be copied. This is the nature of information. But the paper itself can not be cloned: given one sheet of paper you can not magically create from it a second identical sheet.

This gives physical things like paper a wonderful efficiency that digital things don't currently have. The third party that has possession of the digital IOU needs to do work as part of each transfer. They need to be compensated for the work they are doing, which ultimately extracts value from every transfer. Paper doesn't have this problem: because it is self-validating, Bob can possess the written IOU. To transfer it to Charlie requires both of them, but no one else. No value is lost to a third party during the transfer.

Note that these qualities don't stem from paper being harder to forge than email. It's a difference of kind, not degree. Email is just information. Paper is something fundamentally different: a non-cloneable binding structure for information. In fact the security guarantees of paper are actually rather easily subvertable: signatures can be forged, the information it contains can be modified, there's no way to tell when it was written, and there's no canonical way to refer to a given piece of paper (only to the information on it, which can be copied).

Helpfully, there are actually good solutions to most of those problems in the digital world. Public key cryptography gives us unforgeable signatures. Signing the hash of a message makes any change to that message immediately detectable, so content can never be modified. A timestamp service can provide a signal which, when incorporated in a signed message, proves that message was created after a certain time (like holding up a newspaper in a picture). A timestamp service could also incorporate information into its service, proving the message was created before a certain time (like placing a classified ad in a newspaper).

TODA incorporates all of these solutions, providing some improvements over physical paper. These benefits of modern cryptography don't directly solve the problem of making digital things transferable though. To do that we must introduce a binding structure for information, in the same way that paper provides a binding structure for information. That information can always be copied, but the binding structure can not be cloned.

So Bob can possess a TODA-based IOU in a particularly strong sense: not only does he have the digital keys required to transfer it, but he also does all of the work of the transfer himself, with no additional validation or third party input required. The only one who has information about that asset is Bob. He becomes the source of truth for his own digital things.

Some technicalities

A TODA file is the digital equivalent of a piece of paper, which means it needs a way to bind information. The file's binding structure is called its proof of provenance, or POP. It connects the file's unique identifier, called its file id, with a sequence of cryptographic hashes, known as cycle roots, that are shared among users.

Despite its bubbly name the POP is a complicated cryptographic data structure, and this section just skims its surface. For an accessible work on the subject see the TODA Primer. The TODA POP doc more fully plumbs its depths, describing the concrete data layouts and providing proofs of its properties. Some technicalities follow, so if cryptographic hashes unsettle your constitution then please do skip ahead.

We'd like to ensure a file has one owner at a time. This is known as preventing double spending. If Bob sent Alice's email IOU to both Charlie and Dave that would be an example of double spending. Bob could do that because email does not solve double spend, and solving double spend is necessary to have transferrable value.

The main expense of running a decentralized ledger comes from solving double spend. Alice can try to send the same thing to both Bob and Charlie. Only one of those transactions can go in the ledger, but which one? In a centralized system a single entity makes that decision, but in a decentralized system the deciding power needs to be spread out fairly. Deciding who gets to make that decision is the basis of things like proof of work.

In TODA double spend is solved differently. The POP structure ensures double spend cannot occur by allowing only a single entry per cycle, per owner, per file. That's a mouthful, but it means the file has an unbroken chain of custody, starting from its initial creation all the way through to its current owner. This conclusively proves its provenance, and solves the double spend problem to boot.

Doing this requires a using fancy data structure, called a Merkle trie, cousin to the more well known Merkle tree. One can think of a Merkle tree as a list with a couple of extra properties: it has a unique fingerprint, called its "Merkle root", and for each item in the list there is a short cryptographic proof of membership. Given the fingerprint of such a list, just a few short hashes suffice to prove something is in it.

A Merkle trie is very similar. Instead of a list it contains a dictionary of keys and their associated values. Each such dictionary has a unique fingerprint, and each dictionary entry has a short proof of membership. It offers one additional guarantee: a given key has at most one value. In other words, Alice can send a short proof to Bob connecting a key to a value in a trie, and Bob knows that key has that unique value in that trie.

One might suspect that this uniqueness property of Merkle tries would come in handy while trying to build a unique binding structure, and one would be correct. Doubly correct, as it happens, because we're going to use one Merkle trie for associating a file id with an owner, and then take that file trie's fingerprint as the value for a second Merkle trie that has owners as its keys. This is a cycle trie, and its fingerprint is called a cycle root.

Each of those cycle tries incorporate all of the activity that occurred during its construction. Everyone works to build the cycle trie structure, arriving at the shared cycle root together, but following a different path to get there. The Merkle trie guarantees an owner has a single file trie in that cycle, and that a file has a single entry in that file trie. This gives us exactly the quality we said we needed: a single entry per cycle, per owner, per file.

This Merkle trie structure also provides composition. For example, while it would be possible to handle files individually, it would be unwieldy to make a separate transaction for every file in a transfer. The file trie allows an owner to manage millions of files at once, entirely locally, while still yielding very short proofs for individual files thanks to the Merkle trie guarantee. Likewise, millions of owners can contribute file trie roots to the cycle trie with only a trivial amount of information shared among them.

The cycle roots incorporate everything that occurred during that cycle, providing a notion of time. The idea of space is suggested by the owner (usually called an address), who represents a point in the space of possible addresses. Using these and a few other appropriately chosen values we can construct a file's initial dataset, called its kernel, and have a guarantee that the file id generated from this kernel is globally unique.

Another way of saying this is that given two identical file ids, they must also have identical first entries in their POPs. In fact, the same principle can be used to prove that every single POP entry is the same. A given file id has a unique proof of provenance, and must therefore have a unique owner at any given time. That guarantee relies only on the properties of the data structure itself, not any validation or approval from an external party.

This allows TODA files to be transferable without an external authority. Given a POP, Bob knows who owns that file. If Bob owns it, he can transfer it to Charlie with no centralized state management, no trusted party maintaining its integrity, and no validation from previous holders, because the file's POP conclusively proves that no double spend has occurred. The efficiency implications of this can not be overstated.

This approach fully decentralizes the transfer of digital things. In fact, the only thing left to further decentralize is the notion of a single canonical sequence or system, and the singular consensus process responsible for maintaining it. This is an ongoing area of research, but a few points bear mentioning. It's important, because every consensus process comes with a cost, and different use cases have different cost tolerances. We need digital things that can be used in many different places, instead of places that constrain their digital things.

In TODA the unit of consensus is called a ring. Rings can be very large (millions of users) or very small (a single user). Rings are flexible in their setup, with the ability to employ a variety of different consensus mechanisms for constructing cycle tries. They can limit participation to a closed set of nodes, or open it up to anyone who presents a proof of work, stake, or what have you. Rings support each other, pooling their security.

The proof of provenance data structure described above can be extended to incorporate cross-ring file transfers, giving the flexibility to use files in the widest array of use cases while still maintaining their global uniqueness across the space of all rings.

This is what TODA provides: totally unique digital things. And uniqueness turns out to be exactly what digital things need to be meaningfully incorporated into our lives. We each become the source of truth for our own digital possessions.

The early years

TODA began in early 2016, when Toufi Saliba called Dann Toliver at four in the morning with a wild idea. They were experiencing cost, throughput, and latency issues while scaling applications with blockchain components, like a PKI for end-to-end encrypted email. The idea was to give each file a unique number, and use a Merkle tree and a fixed number of validators to ensure ownership was limited to a single node in each block of time, using just the computational power of the devices themselves. They worked on it as a science experiment for months, trying to get traction on the problem, until the pieces finally started to fit together.

In the summer of 2016 Lila Tretikov and Todd Gebhart came onboard and helped guide the early strategic steps. Later that year Hassan Khan joined forces, forming TODAQ, the first venture on TODA. Adam Gravitis took the CTO role at TODAQ in spring 2017, managing the engineering team's work on the reference implementation of the protocol. The researchers, implementors, executives, and partners who have joined along the way would fill more than this article. We're incredibly grateful to everyone who has contributed to the birth of TODA. It wouldn't be what it is today without you.

The first use case we focused on was supplementing cash in cash-primary regional economies in a non-extractive way. Doing this would improve countless lives by enabling efficient delivery of financial services. Regional products like M-Pesa and bKash help prove this hypothesis. A globally available system would have the potential to help billions of people, and a system without profit extraction could offer even greater benefits.

Doing this turns out to be rather difficult. In fact it's impossible without something like TODA. In a world where digital things are just information, a third party must always manage that information. They must be compensated for that work. That compensation extracts value from regional economies. Deliver $100 in aide and move it around via credit cards, and in just a few years over $90 of it has been extracted from that regional economy. With the current implementation of blockchains like Ethereum and Bitcoin there can even more extraction. This is not a small problem.

It's hard even with TODA. Building the infrastructure to maintain a locally operated, globally interoperable TODA installation can be done, with relatively minor expenditures. Even better would be relying solely on people's mobile devices, an active area of research.

Having options like those available at all is due to having this hard case as our primary target. This shaped the protocol, forced us to hack away at inefficiencies and to focus maniacally on places of value leakage. We embraced fragility, turning all the robustness and resiliency knobs down. This gave us access to the hardest use cases, those most sensitive to extra economic weight like micropayments, at the core protocol level. Adding robustness for use cases that need it is easy in comparison: you can easily have as much robustness as you are willing to pay for.

It also forced us to prioritize asymptotic computational complexity over constant factor optimizations. How adding more nodes impacts a single transaction, for example, is central to the protocol. How much work a single transaction requires in isolation is secondary. Indeed, there are a great many optimizations we could bring to TODA, but they add complexity and need to be weighed carefully. Asymptotics limit usage scaling. Complexity limits feature scaling. Flat foundations are easier to build on.

We made a number of other choices in those early days that were counterintuitive or contrary to market trends. We got a lot of pushback for it. In some cases even we weren't sure they were right, but in hindsight it's clear they laid the groundwork for TODA being what is today.

We decided early on that we didn't want the TODA Protocol to be a source of revenue for us, or for anyone. It was clear that the economics of blockchains, which are necessary to allow them to spread trusted state management over many untrusted nodes, also preclude their use in cost-sensitive use cases, and trend toward volatility, extraction, and consolidation. The deep integration of tokens causes them to behave more like products than protocols. Important products, that provide an important service, but TODA needed to take a different course to achieve our goals. So we worked to remove the internal currency from the protocol, and focused the revenue model on partnering to build products and services on top of TODA while leaving the protocol pure.

Internal protocol currencies cover a multitude of sins. Any time there's an incentive misalignment, or extra work needs to be done, or you need to keep someone honest, you can throw economics at it to sort it out. It's the duct tape of decentralized protocol design. If the protocol doesn't understand a currency then those patches have to be torn out, and all those areas ground down and restructured. It was a lot of work, and it wasn't clear it was even possible.

When we finished, though, what we were left with was something small and simple and clean. A protocol that describes how to create a globally unique digital thing, how to efficiently transfer the ownership of that thing, and little else.

By extracting the base currency we'd forced efficiencies, removed a variety of economic weaknesses, and made it a protocol instead of a product. Removing the internal currency means all things created on TODA are treated as equals. It means the protocol works for any kind of asset. Anything you can print on paper, we used to say. And more, as it turned out.

Another big decision came in balancing privacy and compliance. This is actually quite a bit easier in a cash-style system than a stateful, managed model. Cash already has a decent story around privacy and anonymity, but regulatory compliance is difficult because it's hard to prove where it came from. TODA's proof of provenance changes this dynamic, though, by allowing files to have additional metadata attached during each transfer. This metadata could contain identifying material, or proof of limited attestations (like "I am legally allowed to drive" or "this is a legitimate business account"). Voluntarily adding this to the file's POP would allow entities like financial institutions, governments or other large organizations to fulfill their compliance requirements.

In order to preserve the ability for this particular file to be used in those use cases, then, one ought to ensure that its POP contains all the required material. Otherwise it will be difficult to use in those situations, reducing the utility of the file. Thus we render unto governments their due for assets they manage, while keeping impedance low for things like stickers, songs, and micropayment assets.

Over time we came to identify the qualities physical things had that digital things lacked as transferability, agency, possession, and permanence. Transferability means when the owner transfers it they don't need to notify a third party. No one else has to do any work, no one else needs to be compensated. We refer to this ability to be transferred losslessly as value preservation. Agency means you can do all the things you can usually do with a physical item: give it away, sell it, rent it, lend it, and so on. Possession means the source of truth of the ownership is in your hands, and decisions made in some corporate headquarters can't take it away from you. And permanence means if you take care of it well there's a chance you can pass it down to your kids. Those qualities imbue every file in TODA, providing an important part of the foundation for restoring ownership and control of identity, assets, and data to every individual human.

The present

The simplest things often create the widest array of opportunity and application. Today TODA is being put into use across verticals like supply chain applications, real estate, smart city services, retail, education, entertainment, digital media, AI, healthcare, government, finance and insurance.

Frankly, this barely scratches the surface of what TODA can do. Having digital things that work like paper things unlocks our ability to manage our own health records, credit scores, legal documents and more. It reduces the trust burden on organizations and individuals by allowing their claims to be validated, lowering the barrier of entry to markets and financial inclusion. It's changing the Internet and e-commerce through efficient micropayments and secure two-way swaps, redecentralizing the web.

There's a common vision shared among the builders of these products, markets, and systems. Whether from a corporate, technical, or academic background, everyone involved is working to maximize the utility and value created in the world by TODA. Together, we can make life a little more ideal for everyone.

The future

TODA's trajectory involves continually finding ways to expand the boundary of the places where TODA files can be used. Adding functionality and increasing efficiency are the primary ways of doing this.

Adding new functionality allows accessing use cases with more sophisticated requirements. This is mostly done by building functionality at higher levels, through additional protocols built on TODA or inside applications that use it.

Increasing the efficiency and decentralization of TODA allows accessing use cases that are highly sensitive to factors like cost and latency. When applied to rings, for instance, this means breaking down the idea of having one ring to rule the whole space of digital things. That doesn't mean there won't be a single, globally acknowledged ring that everything is lifted into, but this needs to be de facto, not de jure. The use cases of the future demand it.

One of the advantages that emerges from that kind of radical decentralization is integration with other decentralized technology, like ledgers for managing complicated state transitions. An obvious next step is to teach those ledgers to manage TODA files, which can move into ledgers, out of ledgers, and flow between ledgers. Assets currently trapped in ledgers can be released, allowing them to be used in ways that are currently inaccessible, like efficient micropayments.

Another long term advantage is opening the space of possible use cases to support high latency connections, including local rings occasionally syncing into more well connected rings and ultimately culminating in full offline mode and true peer-to-peer transfers, whether here at home or far away.

TODA files are digital things that have the qualities of physical things. Transferability. Permanence. Agency. Possession. Qualities that physical things have always had. Qualities that digital things have today, thanks to TODA.