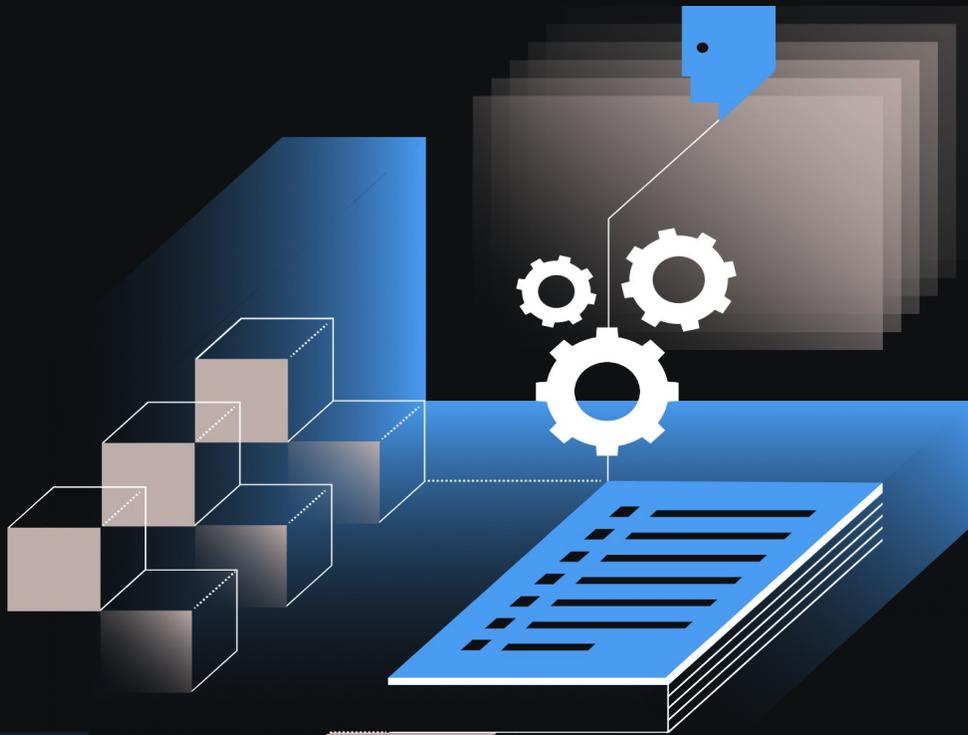


# Use smart contracts without any blockchain experience



# Table of Contents

4	<b>Executive Summary</b>
5	<b>Introduction &amp; Background</b>
6	• Blockchain Developers and Industry Growth
9	• Executing Smart Contracts: a Mammoth Headache
11	<b>Blockmason Link</b>
12	• Why Link?
12	• Link for Developers <ul style="list-style-type: none"><li>○ Unburden Yourself (of Boring Dev Tasks)</li><li>○ Treat Your DApps as Apps (and Vice-Versa!)</li></ul>
14	• Link for End-Users
15	• Mechanics
15	• Features <ul style="list-style-type: none"><li>○ Documentation &amp; Sample Code</li><li>○ Insights, Monitoring, &amp; Data Visualization</li><li>○ User Directory Integration</li><li>○ Automated Ethereum Account Generation + Maintenance</li></ul>
16	• Benefits <ul style="list-style-type: none"><li>○ Convenience &amp; Ease of Use</li><li>○ Managed High-Availability Infrastructure</li><li>○ Enterprise-Grade Security</li><li>○ Portability</li><li>○ Blockchain-Enabled Mobile Applications</li></ul>

17	• Use Cases
	◦ Traditional Application Developers
	◦ Hybrid Deployment
	◦ ICOs and Product Legitimacy
	◦ Private Side Chains for Enterprises
18	• The Next Step in Managed Infrastructure
20	• Technical Overview
	◦ Web Service
	◦ Access Control
	◦ Analytics
	◦ API Reference Documentation
	◦ Client Software Developer Kits (SDKs)
	◦ Synchronized Versioning
	◦ Logs
21	• Security
23	<b>Token Features, Usage + Economics</b>
25	<b>Revenue + Business Models</b>
25	• The Path to Revenue
25	• Premium Management Services
26	<b>Development Roadmap</b>
28	<b>Use of Funds</b>
29	<b>Meet the Blockmason Team</b>
31	<b>Disclaimer</b>
32	<b>Glossary</b>

# Executive Summary

## Blockmason Link is a new application designed to generate simple web APIs from smart contracts written on programmatic blockchains.

While Link will initially launch on the Ethereum network — and we will use Ethereum as an example throughout the whitepaper — Link has the potential to support smart contracts on every major programmatic blockchain. The goal of Link is to remove the barriers developers and users face in interacting with blockchain-powered applications. It accomplishes this objective by abstracting away the many confusing, esoteric processes currently necessary for a developer to interact with a smart contract or a user to interact with a decentralized application, including:

- Learning Solidity or other programming languages
- Purchasing Ether (ETH)
- Hosting an Ethereum node or related infrastructure
- Running MetaMask or similar wallet software
- Creating an Ethereum address

The process of creating a “Link app” from a smart contract is quite simple: any developer may register a smart contract with Link by providing its address and ABI, from which Link generates a set of standard API endpoints. Link also generates synchronized, dynamic API reference documentation, as well as interactive analytical tools to help developers understand how their smart contract is used, both on and off Link. Finally, Link also generates an API key for use by the developer, as well as an ETH account that Link will use to perform requests on the developer’s or client’s behalf. Link has three primary, and vital, value propositions:

- 1. Create a traditional, web-based application from a smart contract or decentralized app**
- 2. Allow traditional developers to empower their web-based applications with smart contract infrastructure**
- 3. Enable users unfamiliar with blockchain to utilize blockchain-integrated applications in a seamless, recognizable web-based system**

Link is a tool of adoption and growth for blockchain-based DApps. It opens up blockchain programming to developers without blockchain experience, and opens up blockchain-powered applications to users unfamiliar with the arcane intricacies of accessing networks like Ethereum.

Furthermore, Link is a tool applicable to every blockchain enterprise. By managing the necessary, but tedious mechanisms of blockchain interaction, Link frees developers to pursue more creative and innovative projects. It also opens every blockchain project to the general public, making decentralized applications as easy to use as any other application with which the population is already familiar. With Link, it is entirely possible to imagine a near future in which every member of our society utilizes a blockchain-powered application on a daily basis without even knowing it.

Consistent with the theme of removing barriers to blockchain adoption, Blockmason is working hard to make the Link Software-as-a-Service (SaaS) offering simple and familiar for all users. Developers, startups and entrepreneurs will have the option of staking their account with Link's token, or using traditional payment methods like credit cards.

# Introduction & Background

## Blockmason is excited to announce the creation of our second blockchain industry revolutionizing product: Blockmason Link!

At its core, Link removes the barriers to interact with smart contracts, allowing developers to instantly turn smart contracts on any programmatic blockchain into user-friendly applications. We'll use Ethereum smart contracts as the example throughout this whitepaper, but Link is main-chain agnostic and is architected in a manner that is adaptable to the leading smart contract supporting main-chains without a native Web API layer.

Link solves some of the most persistent and pervasive problems plaguing the blockchain community: ease-of-use and accessibility. While Bitcoin and Ethereum have grown in the public consciousness, blockchain still primarily functions as a vehicle for cryptocurrency trading and other financial applications.

At Blockmason, we are true blockchain believers — it is only a matter of time until blockchain [spurs the healthcare revolution](#), [empowers economic freedom in developing countries](#), and [stabilizes our fragile mass payment systems](#). However, it is impossible to realize this beautiful crypto-future until the completion of essential components of blockchain infrastructure.

Blockchain has a high barrier to entry. From a development perspective, it requires learning Solidity, cryptography APIs, and the nuances of esoteric decentralized systems. From a user perspective, the hesitation is even simpler: decentralized applications are too damn tough to use! In a world that increasingly values convenience, it's difficult to convince people to increase their DApp usage when usage necessitates so many prerequisites, including creating and learning to use an Ethereum wallet, MetaMask, safely storing your private keys, and so on. Just try explaining to someone the long-term benefits of decentralized contracts when even those in charge of regulating new technologies have [no idea how they work](#). All the public thinks is: why use a “smart contract,” when dumb contracts work just fine? Why use [Lndr](#), when I could use Venmo?

Which brings us to one of Blockmason's core beliefs: for an emerging technology to succeed, it must value the user. For too long, the blockchain community has ignored functionality, in favor of niche technical development, finance, or profit — hindering mass adoption. This is not to say we don't love nerdy, precise technical advancement (we do!), but rather that we believe a smooth, exciting user experience is a necessary part of technological growth.

Fully realizing the potential of blockchain technology requires the mass adoption of blockchain technology. Mass adoption requires an enhanced user experience, which itself requires technological innovation. Conversely, any technical breakthrough, regardless of its importance, is squandered if it does not solve the user experience. Greater access to technology leads to greater technological advancement. Just think of us as blockchain's “Typhoid Mary.”

**Blockmason Link — the missing block in blockchain's ever growing tower. An innovation that sparks innovation.**

# Blockchain Developers and Industry Growth

Blockchain is on the verge of a breakthrough. It is the web in the nineties — the public is obsessed with it but has no idea how to use it. This enthusiastic, curious tech community means that now is the perfect time for Link, an application that enables developers to make possible the mass adoption of blockchain technology.

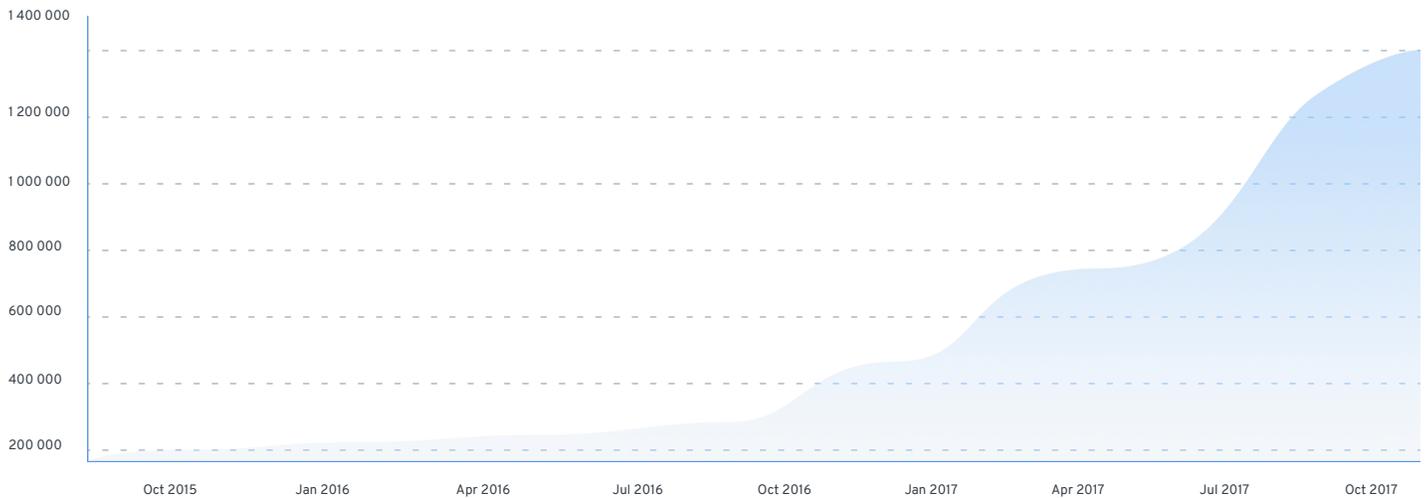
Consider some basic information about the growth of the Ethereum ecosystem in particular.

In 2017 the average number of Ethereum transactions per day increased more than

# 600%

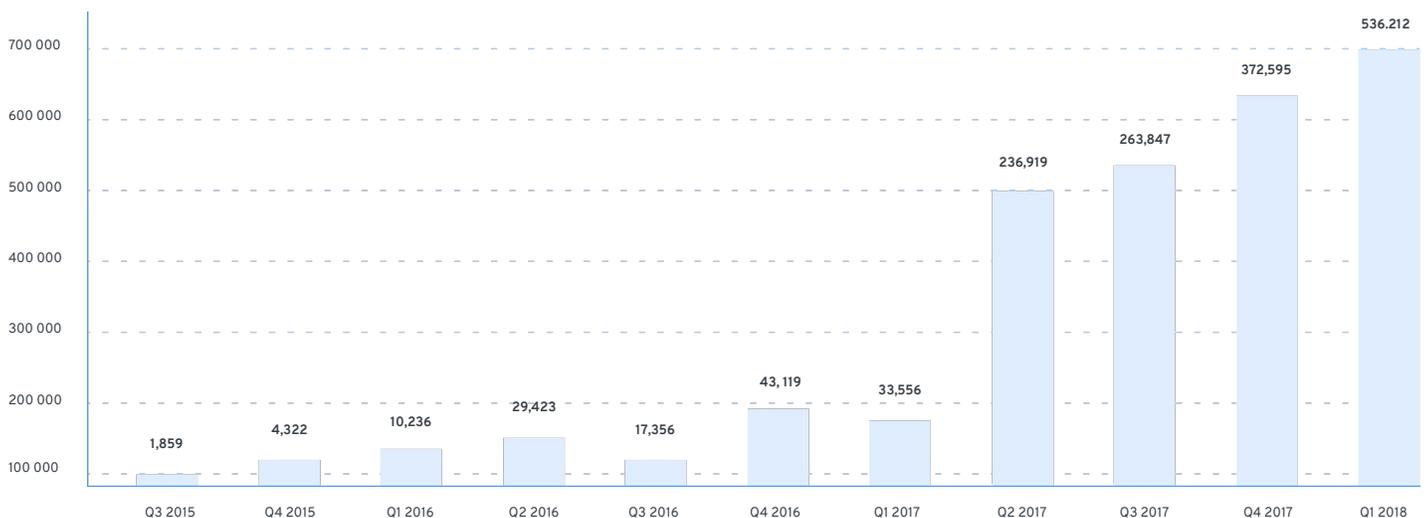
by the end of the fourth quarter

## Ethereum network transactions per day



Even more relevant to the necessity of Blockmason Link is the creation of new Ethereum smart contracts.

## New Ethereum smart contracts by quarter



Despite the increase in the cost of ETH by early 2018, and therefore a temporary increase in the cost of creating smart contracts, 2018 has already seen a record number of smart contracts created, demonstrating significant developer interest. However, further research indicates that 94% of smart contracts were called fewer than 10 times — indicating some structural impediment to further use and development. Notwithstanding these difficulties, an increasing percentage of developers have reported already using or plan to use blockchain in their development:

## Blockchain Development – Familiarity with Blockchain

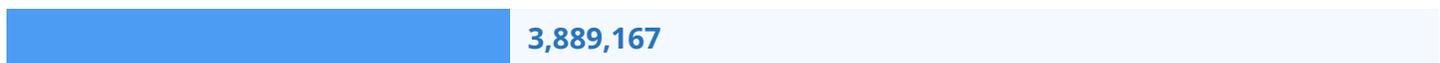
Know and plan to use in my development



Not familiar at all



Know and use now in my development



Am familiar with the concept but have no plans for them

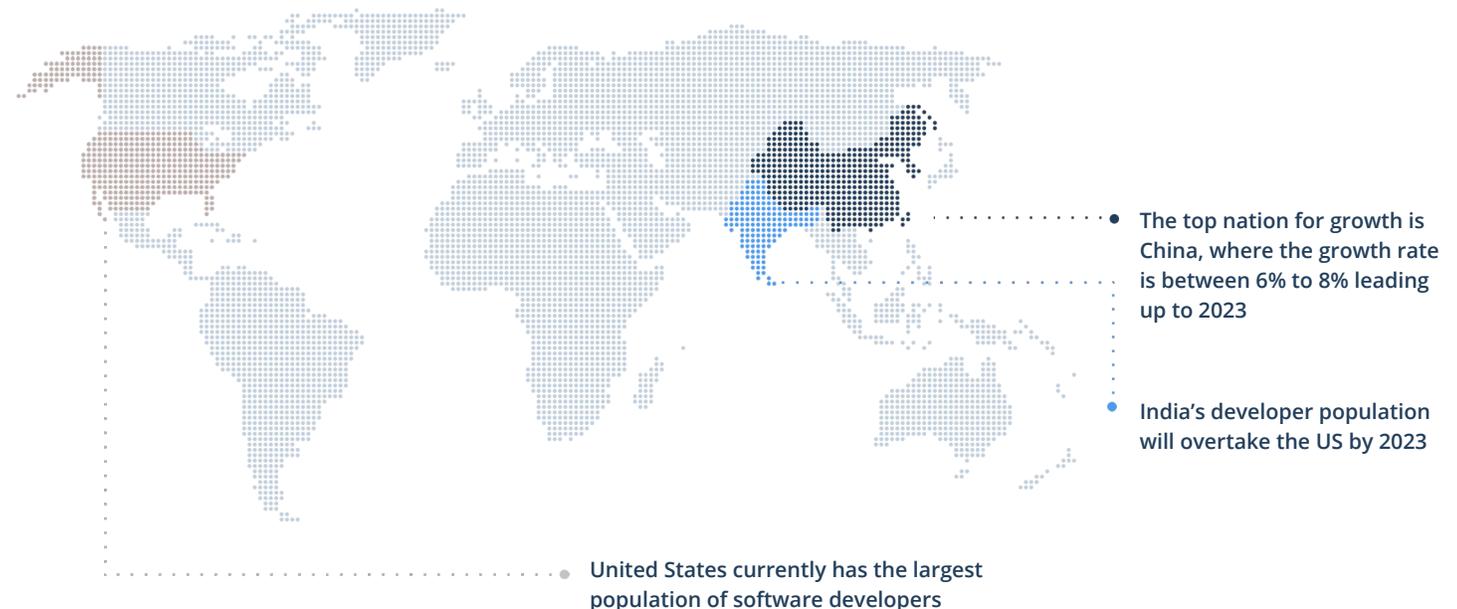


Have heard about this but am not familiar with the concept



In addition, the global developer population, a primary target of Link, continues to grow at unprecedented rates. It is projected that by 2023, the global developer population will reach almost 28 million:

## Global development population



2018 : 23 Million developers      2023 : 27.7 Million developers

At the same time, Ethereum continues to grow in the public consciousness. Ethereum-related Google searches recently reached their highest levels since March, when ETH's price had reached \$800.

Beyond programmatic and societal growth, Ethereum has already seen a record number of ICOs in 2018. Though the year is not yet finished, more than 1,000 ICOs have occurred raising over \$6.2 billion USD — beating 2017's total for the entire year.

As we detail in our Use Case section, ICOs comprise one of the spaces to which Link is most valuable, potentially allowing new blockchain companies to bolster their token offering by using Link to generate working apps prior to token sale, a feat all too rare in today's ultra-fast, "Wild West" ICO world.

# Executing Smart Contracts: a Mammoth Headache

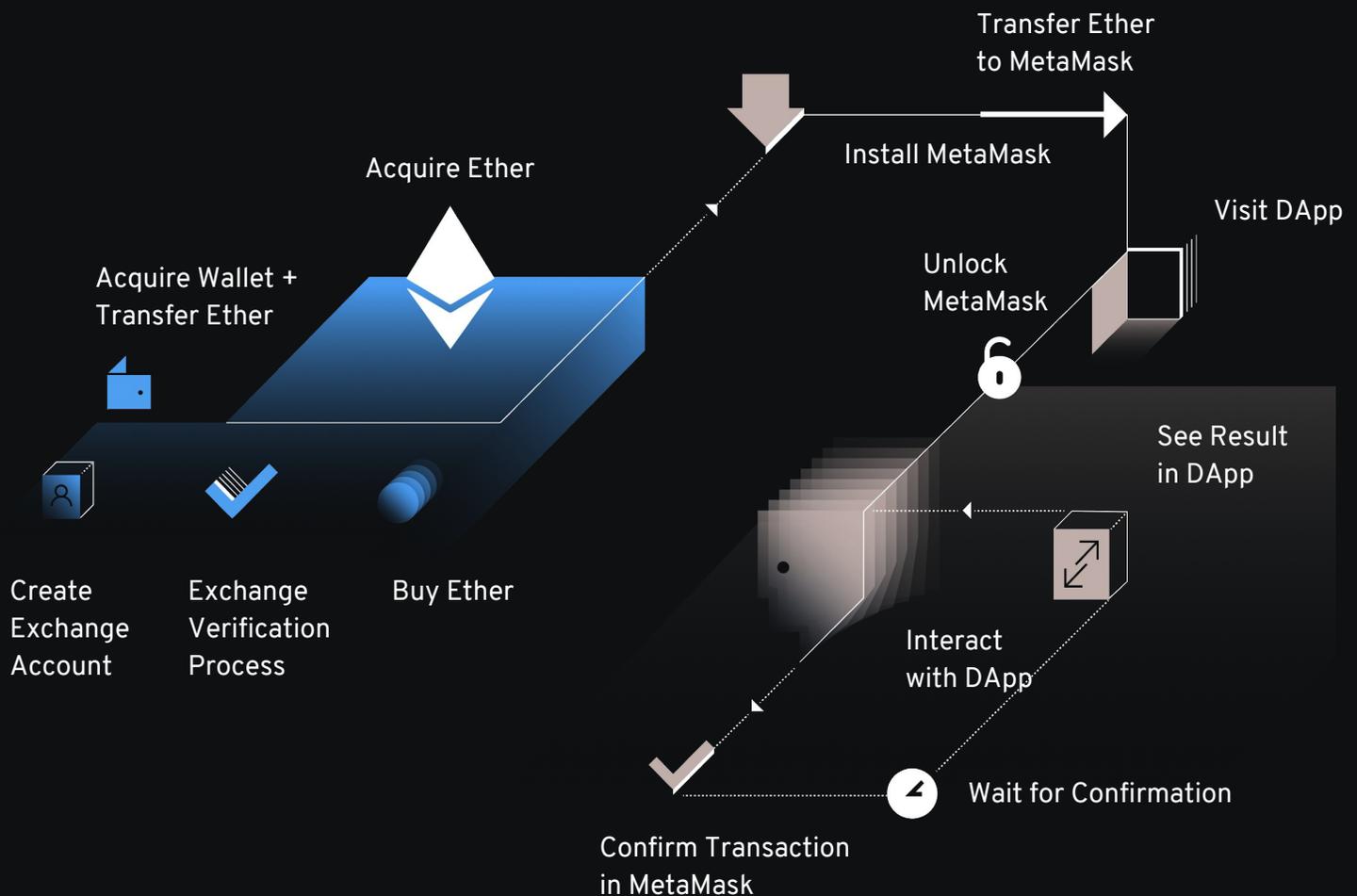
In order to understand how Link will revolutionize smart contract usage and development, it is important to understand just how frustrating it currently is to interact with smart contracts.

Blockchain programming is no joke. While Solidity is relatively straightforward, it is an oasis in a vast desert. It's small, uncomplicated, and effective. However, it is impossible for a developer to use anything they have built in that oasis without hauling across the vast desert and back to civilization.

Relatively few developers outside of blockchain have a working understanding of Solidity or how to build applications that can interact with Ethereum-based decentralized applications or smart contracts. Even fewer prospective users understand how to utilize and access those applications.

This is completely understandable — quite simply, decentralized applications have a huge barrier to entry.

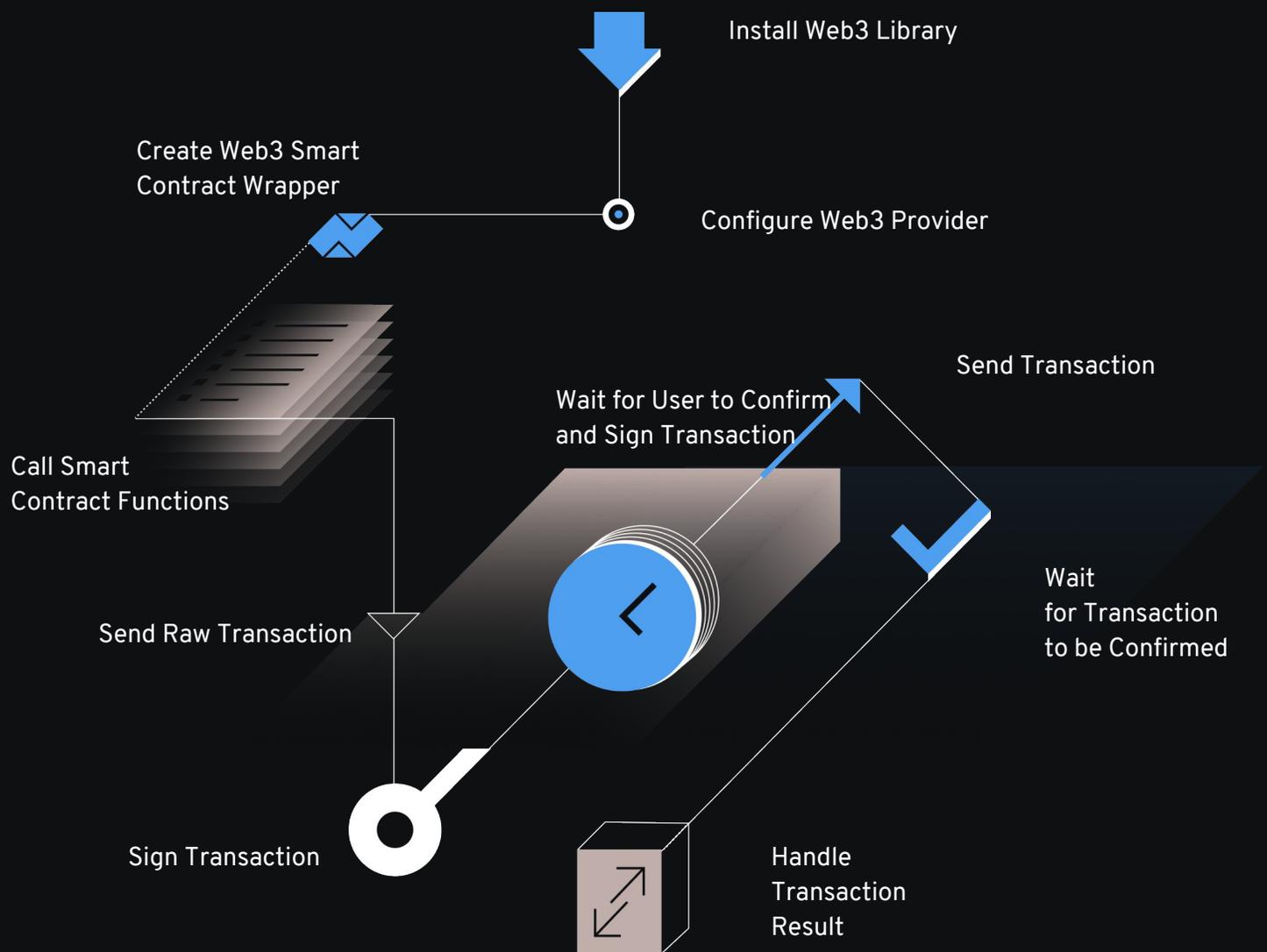
Just check out everything needed to interact with a DApp from the perspective of an end-user:



Before a user can even think about jumping into the complete mess of DApp usage, they must first acquire an Ethereum address, which is a process complicated enough there are many dozens of blog posts dedicated to doing it correctly. The user must also acquire ETH and transfer it to that address, which often requires first making an account at a large exchange (Coinbase, Binance, etc.), submitting personal data for identity verification, linking a bank account, receiving approval, and purchasing ETH, all of which can take several days or weeks.

Once the user finally has a wallet and some ETH, they still need to boot up Google Chrome — or install if they don't already use it — download MetaMask and link their wallet before they can even think about playing around with a decentralized application. Of course, this isn't even accounting for all the complicated mechanics of sending transactions, calculating sufficient gas, and waiting around to query the network again to see if the transaction was a success.

We can all agree this is far from the "click of a button" experience everyone is used to in the app store. It's a scary, confusing, and frustrating process. And that's not even taking into account the difficulties of using Web3 to build a typical DApp that interacts with a smart contract:

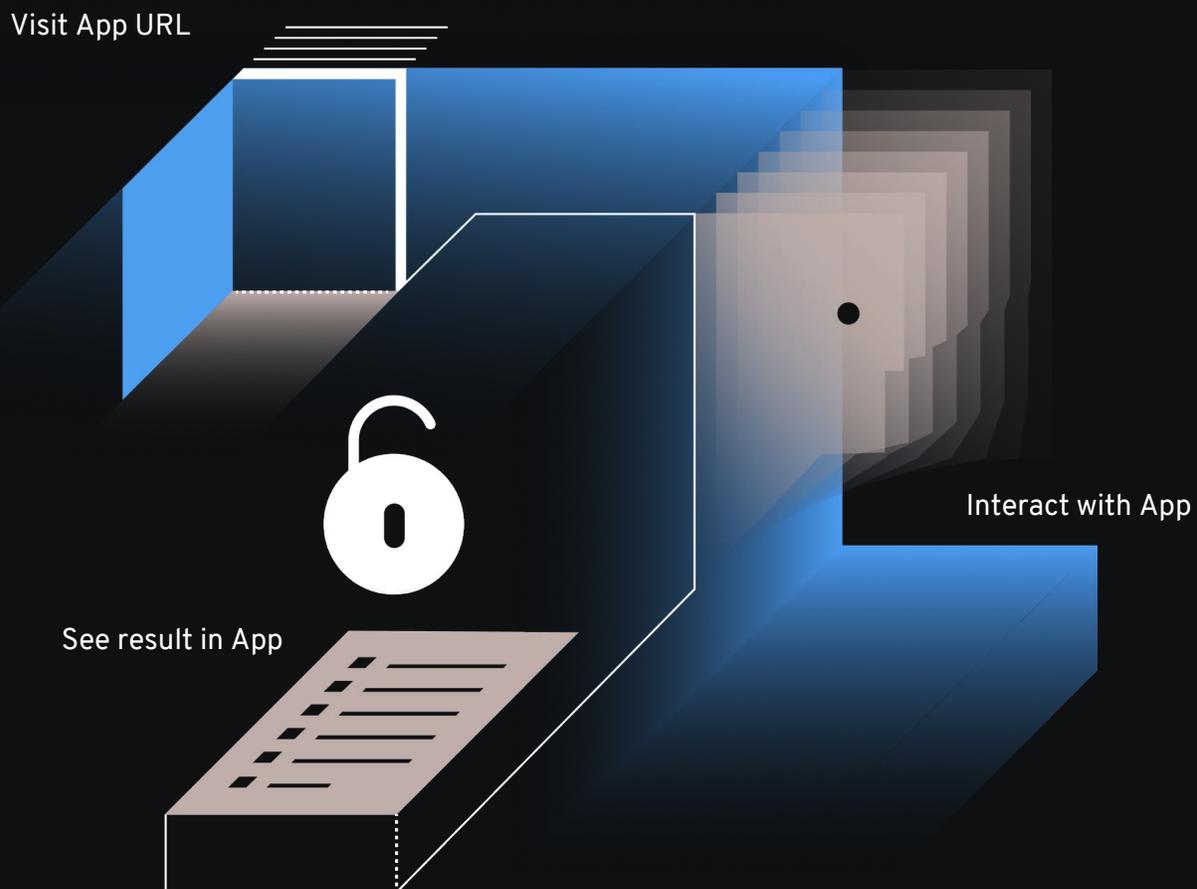


Not to mention writing the actual smart contract itself if the developer isn't utilizing a contract that is open source. But, have no fear! Link is here to transform these nigh-impossible processes into something vastly easier to use.

# Blockmason Link

Blockmason Link is simplification. It is elegance. Before, you were running with ankle weights. Allow Link to remove them, so that you may run free! But what does Link actually do?

Well, remember that nasty graphic above with all the arrows and long pathways? Let us present you with something a whole lot simpler:



Notice everything that using Link does not require: an Ethereum address, deposited ETH, MetaMask, a Coinbase account, and a ton of other annoying steps. Link facilitates a smooth user experience by handling everything that traditional application users don't care about, including Ethereum wallet creation, gas calculation, and communicating directly with the Ethereum Network.

**Link creates a classic, conventional web API interface for any smart contract, making the benefits of blockchain programming available to developers without any blockchain experience.** Even more exciting, Link enables hybrid applications that utilize the power of the Ethereum network for certain tasks, while employing alternative networks or cloud services for others, including the user interface. Additionally, Link opens an entirely new population of prospective users and clients to the growing world of decentralized applications, finally making it possible for blockchain technology to expand beyond the pool of early adopters.

## So what do we mean when we say “Turn your DApps into Apps?”

Link doesn't make a special kind of app; it makes apps less special. Link makes existing DApps and smart contracts more recognizable from the user perspective, allowing prospective consumers and customers to use Ethereum-based decentralized applications without even having to think about blockchain.

### Why Link?

Link constitutes necessary infrastructure in the developing blockchain ecosystem. It is more than a single application; it is every possible application. Link simplifies and streamlines the process of interacting with smart contracts for everyone involved in blockchain — developers, users, tech enthusiasts, cryptocurrency traders — and makes it easier for those outside of blockchain to join us in the tech revolution.

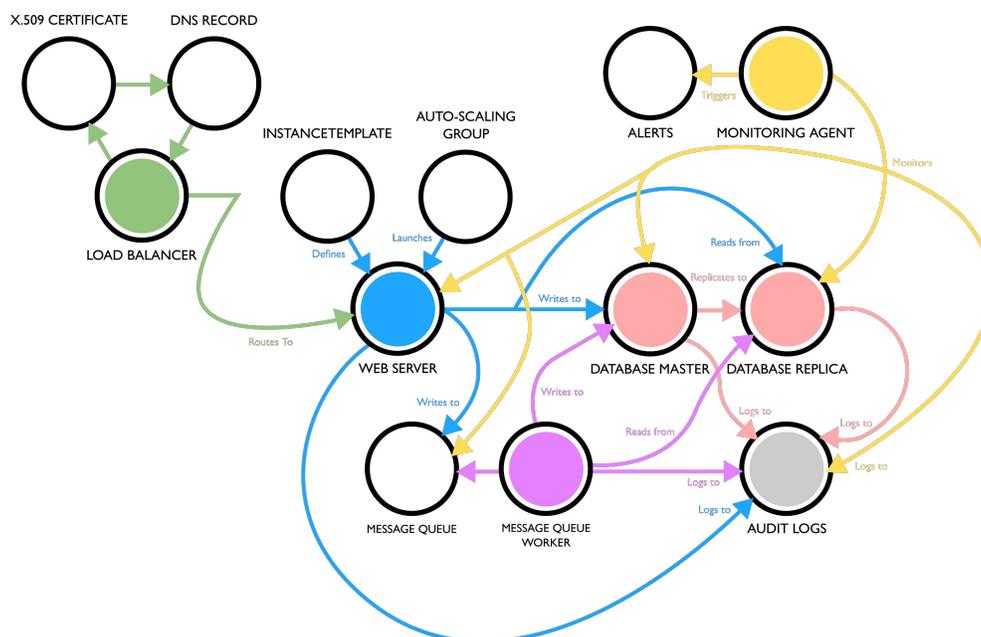
### Link for Developers

#### Unburden Yourself (of Boring Dev Tasks)

Imagine if Atlas was no longer tasked with holding up the entire world. With that weight off his shoulders, he could be a star athlete, the leader of a country, or maybe even a blockchain dev!

That's not exactly what we're promising, but without the burden of connecting their smart contracts to the world outside the blockchain, developers will be free to focus on more satisfying creative endeavors. We know that every developer has limited energy. When they must spend their time focusing on tedious, technical aspects of development, it reduces the mental power that may be spent elsewhere. Perhaps that means taking much longer to complete a project; perhaps that means never fully realizing a project at all.

To understand what goes into building and maintaining different types of applications, below is a simplified graphic that represents the infrastructure of a typical database-backed web application. Each node represents an aspect of the application that the developer must build and maintain:



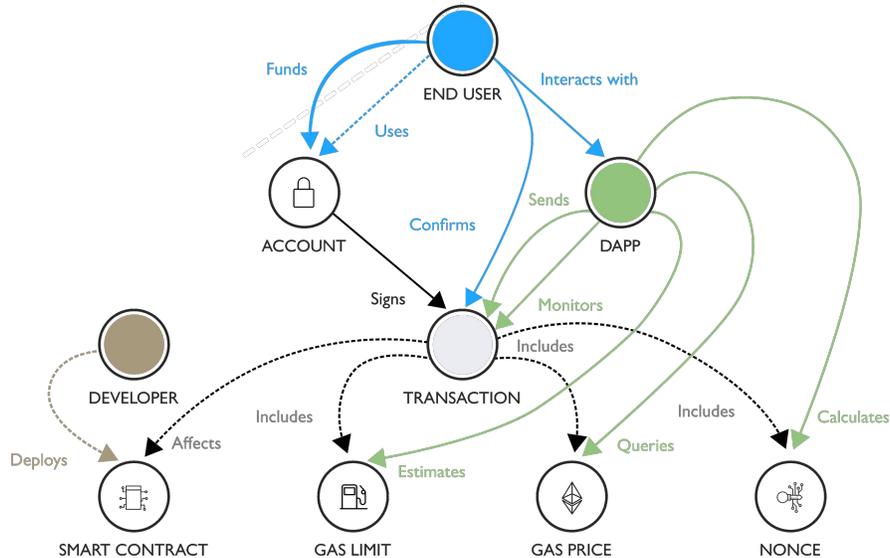
Link creates a classic, conventional web

# API

interface for any smart contract, making the benefits of blockchain programming available to developers without any blockchain experience.

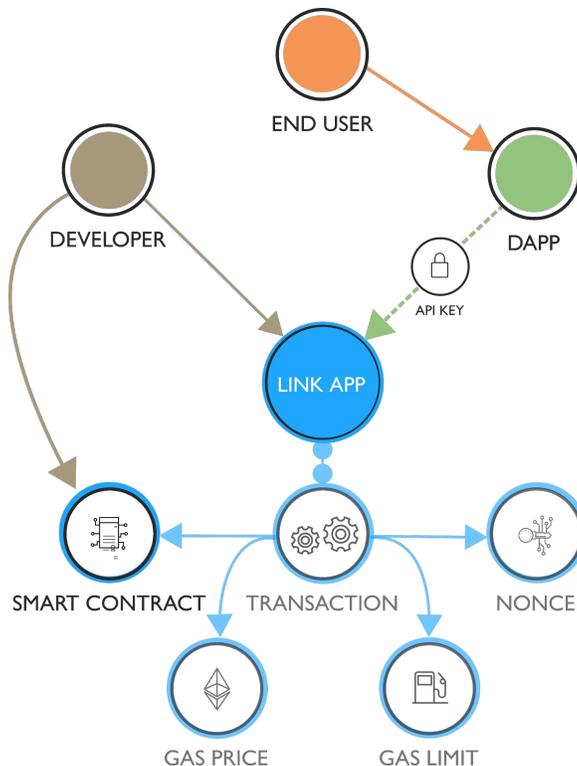
While it is entirely possible, and even common, to operate a web application with a less sophisticated setup than that described above, such a deployment is not without significant trade-offs in reliability, security, scalability, and performance.

In many ways, creating and maintaining a traditional decentralized application is much simpler. It moves much of the burden of interacting with the application onto the end user, which in turn makes the application less accessible:



After writing the smart contract, the developer has minimal responsibilities, but their application may have limited functionality or usability. Of course, this diagram does not include any of the logistical challenges of hosting an Ethereum node or using a third-party infrastructure service, as described previously.

Next, let's consider the steps involved in building and maintaining a web-based, blockchain-backed application using Blockmason Link:



As you can see, Link eliminates many of the steps required of the stakeholders involved with the building, operation, and use of an app. Once deployed, a developer can connect a smart contract to Link and use the resulting API endpoints to build an application that is simple to access for the end-user, backed by the blockchain without hosting an Ethereum node, and that interacts with traditional web browsers without the complicated infrastructure necessary to run a regular web-based application.

The goal is to maximize the usability and power of the developer's application while minimizing the amount of logistical effort necessary to deploy and maintain that application. This is one of Link's main value propositions: Let Link shoulder the burden, so that you may spend your energy on what truly matters to you.

For developers, Link is like the introduction of agriculture. Once the general population had their needs met, new, more specialized jobs and skills emerged. Only then, could human society grow and flourish.

### — Treat Your DApps as Apps (and Vice-Versa!)

With Link, blockchain development is accessible to anyone with traditional programming experience. Anyone who can use a web API can consume a smart contract. Link generates all documentation, providing for developers all references, data types, endpoints, and demos. Additionally, Link can connect multiple smart contracts without the developer ever having to interact directly with the blockchain, leading to the possible creation of elite, collaborative, hybrid blockchain microservices. Any developer working on a traditional application can charge their product with the power of the blockchain, without even knowing how blockchain technology works.

Of course, the reverse is also true. While Link opens blockchain programming to those with more traditional development experience, it also opens the brilliant work of blockchain developers to the general public. It creates a simple, easy way for users to interact with blockchain applications through the web. Once confusing and complex, decentralized applications will now be as easy to use as anything else in the app store. Additionally, because Link allows traditional developers to interact with smart contracts through a registered portal, it offers blockchain developers the potential to monetize their creations by charging license fees for the use of the smart contract.

Extrapolating from this premise, Link enables the creation of a massive database of open source blockchain apps powered by smart contracts registered on Link. Such a database would empower developers with unprecedented discoverability, allowing them to pick and choose which pieces of a new application they build themselves, and which parts are built upon Link-enabled smart contracts.

## o Link for End-Users

For users, Link is all about accessibility. In some circles blockchain is a dirty word — people don't know what it is, and they don't want to know. With Link, it will be possible to create apps powered by the blockchain without users having any idea. Additionally, because Link automatically generates Ethereum addresses for all apps, users won't even necessarily need to have an Ethereum account or own any ETH to access the network. Sometimes, you know you're doing your job well when no one knows you're doing your job.

## ◦ Mechanics

In essence, Link takes care of all the annoying steps between a user interacting with an application and that application connecting with the Ethereum network. Everything necessary to execute a smart contract under normal circumstances still occurs — a transaction is created, signed, and sent — only now, all those interactions are handled in the background by Link, away from the user.

If you're a developer looking to register a smart contract with Link to create an app, the process is quite simple. Register a smart contract by providing its address and ABI to Link. Then, Link generates a set of endpoints from the ABI. The Link app developer can then adjust these endpoints to their liking by renaming the path or parameters, and Link will handle mapping those to the appropriate contract function and parameters. Link also generates an API key for your use, and an Ethereum account that Link will use to perform requests for you.

## ◦ Features

Beyond its basic functionality — creating easy-to-use web APIs from difficult-to-access smart contracts — Link also hosts a variety of other features useful to both developers and users alike.

### — Documentation & Sample Code

Link generates usage documentation for all registered smart contracts, including sample code, to guide developers in how to use the Link-generated app. This documentation is synched to the app's configuration and hosted by Link, so developers do not need to manage or update their documentation on their own.

### — Insights, Monitoring, & Data Visualization

Link provides visualizations, logs, and insights into how registered applications are being used, with tooling that makes all of the data filterable, sortable, and searchable. With Link, it is possible to understand exactly how your app is being used, and even see how the associated smart contract is being used outside of Link for comparison. Developers can sign up to receive notifications if there are ever any issues with their app, including if one of the associated accounts is running low on ETH to cover transaction fees, or if error rates on the smart contract have increased.

### — User Directory Integration

Typical DApps and smart contracts rely on Ethereum's native authentication, using an Ethereum account's public/private key-pair.

Link uses a delegated authentication model, so transactions on a smart contract are signed using an Ethereum account within Link, which a developer can connect to one or more identities within the Link app. These identities can be authenticated in any number of ways. Link supports a variety of authentication methods for individual Link apps, so users and developers can seamlessly integrate access with their own identity provider (e.g: Active Directory, Okta, SAML, OAuth), or let Link handle authentication and identity management for them.

### — Automated Ethereum Account Generation + Maintenance

Recall that when using a typical web app, a user will authenticate with a trusted third party. For example, clicking "Login with Google," "Sign In with Facebook," or entering a username and password which is stored by the app service provider.

In a typical Link-based DApp, it is exactly the same from the end-user's perspective. They will authenticate with the DApp, thus proving their identity and accessing the app. However, in order to communicate with a programmatic blockchain-based network like Ethereum, this type of authentication isn't quite enough. A DApp may have to communicate with a smart contract on a user-by-user basis, which in turn means that each user would need their own Ethereum account.

Link automatically generates these Ethereum accounts and manages them for DApp developers. Each account is trusted by the DApp and permitted to interact with the Ethereum network on a specific user's behalf. The DApp does not have to provide any information about the users or identities it manages to Link. Nor does the DApp have to supply any of the user's private authentication details. The user connects to the DApp, the DApp connects to the Ethereum account, and together with Link they connect to the Ethereum network.

Of course, the DApp does need to verify that the corresponding Ethereum account for a particular user has sufficient ETH to cover the transaction fees for any transactions the DApp intends to perform. The Link developer can transfer ETH to these managed accounts directly or allow Blockmason to manage these accounts on their behalf, abstracting away transaction costs from end users.

It is important to note that Ethereum accounts generated within Link are managed exclusively by Blockmason — the private keys are never available to the Link developer. This is significant for:

- **Security:** Link app developers cannot impersonate their users on the Ethereum network.
- **Reliability:** Guarantees transactions are processed in order, reducing spurious failures due to transactions potentially occurring in parallel.
- **Cost:** Reduced potential for failed transactions means less wasted gas.

In automating the Ethereum account generation and maintenance processes, Link eliminates many of the financial and technical hurdles of blockchain usage, while retaining the decentralized computation and persistent storage features of blockchain technology.

## ○ Benefits

### — Convenient & Easy to Use

With Blockmason Link, building a DApp becomes no different than building any other app. Now, there is no need for developers to concern themselves with Web3, running an Ethereum node of any kind, requiring the end-user to load an Ethereum account with gas money, RLP encoding, wallets, or any specialized libraries. Just get an auth token and use your smart contract like you would any other web API. Any software capable of using a web API can now use your Link app, with no special libraries, tools, or blockchain knowledge required. It's really that simple. Additionally, because Blockmason Link automatically generates new Ethereum addresses, it is possible for users completely unfamiliar with blockchain technology to operate the app without any additional effort on their part. From a user perspective, the published decentralized application will be no different from any other typical application, even though it's really blockchain-powered...and that's just cool.

### — Managed High-Availability Infrastructure

With Blockmason Link, you don't need any other infrastructure in order for people to start using a decentralized application. Developers must only deploy their smart contract, create their app in Blockmason Link, and they're done. No servers to maintain or infrastructure to monitor.

We handle it all. In this way, Blockmason Link builds on the important work of companies like Infura and other managed service providers, allowing developers and the blockchain community to focus on building exciting products and services without worrying about the difficulties of accessing and utilizing the Ethereum network. Focus on creating amazing products, not reinventing the blockchain wheel.

### — Enterprise-Grade Security

At Blockmason, we take security very seriously and make every effort to secure Link apps and their data. In addition to a responsible disclosure policy and a generous security bounty program, Blockmason Link will, as a regular feature of ongoing maintenance and support, undergo quarterly penetration testing, use automated continuous security monitoring of our infrastructure and products, and follow security-conscious development practices that include threat modeling, integrated static analysis, and mandatory peer review. Blockmason maintains a presence at some of the most prominent security conferences in the United States and remains highly engaged in the security community.

Additionally, if a developer's app is targeting medium-large businesses and enterprise customers, a common barrier to entry with these customers is compliance with a number of security certifications, including SOC 2, ISO 27001, HIPAA, FIPS, and others. Pursuing these certifications and maintaining ongoing compliance is an inherent feature of ongoing maintenance and support of Link. Certifications can then be applied to applications registered with Blockmason Link. Compliance with these certifications can be burdensome and expensive, but with Blockmason Link, developers can let us handle compliance for their app.

### — Portability

Blockmason Link is just a layer on top of a smart contract, so developers or their users can elect to use the smart contract directly at any time.

Any data a developer has loaded into the Blockmason Link layer can be exported at any time. a developer may delete all of his or her data from Blockmason Link, and have any funds remaining in the app's ETH accounts transferred out.

### — Blockchain-Enabled Mobile Applications

For the first time, Blockmason Link enables the uncomplicated creation of walletless, blockchain-enabled mobile apps. Because Link generates a simple web API, any developer can create a mobile application that seamlessly interacts and integrates with the blockchain. Additionally, Link potentially disrupts [app store censorship](#) of third-party applications that send and receive cryptocurrency.

## o Use Cases

As Link is applicable to any smart contract or DApp currently being developed, its use cases are near infinite. Any outward-facing DApp that interacts with Ethereum or other programmatic blockchains can (and should) use Link.

With that in mind, here are a few general use cases:

### — Traditional Application Developers

Many traditional application developers are interested in harnessing the power of the blockchain but don't have the resources or technical knowledge to do so.

Instead of facing the challenges of hosting their own nodes or the hassles of using Infura, developers may now connect to the blockchain using Link's web APIs, creating a blockchain-powered application in minutes. For further comparison to existing infrastructure services, including Infura, see "The Next Step in Managed Infrastructure" below. Link also features multi-chain support, for DApp developers that desire connectivity on many chains without the hassle of supporting each chain in turn. Developers may rapidly launch an application on all platforms with nodes hosted by Blockmason, without having to code in anything other than JavaScript. It's the easiest possible way to develop a decentralized application.

### — Hybrid Deployment

With Link, it is finally possible to run web services alongside Ethereum-based blockchain services. While this has near limitless applications, a simple and familiar example would be a hybrid-blockchain authentication or authorization service. By delegating certain tasks to the Ethereum network, developers can utilize blockchain only for those aspects of the product which benefit from added decentralization, all while having a UI that is completely indistinguishable from other non-blockchain applications.

### — ICOs and Product Legitimacy

With so many ICOs launching each month, it can be difficult for new projects to differentiate themselves. Additionally, prospective buyers are rightly skeptical of ICOs not attached to a working product prior to their token sale. By connecting a smart contract to Link, ICO project teams can generate an easy-to-use application that demonstrates their product's legitimacy to all potential buyers and allows users to test a product before they buy the token. Link will host all relevant nodes and provide the necessary web APIs, allowing the product to take center stage.

### — Private Side Chains for Enterprises

While we have primarily focused on Ethereum-based smart contracts — as that is the most active programmatic blockchain — Link supports smart contracts developed in any blockchain. That makes Link particularly useful for anyone looking to establish private side chains, a trend that has caught on among large companies like IBM. However, even private blockchains still have a high barrier to entry for the average employee of such a company. With Link, companies like IBM can generate simple apps that issue credentials to registered employees, allowing anyone associated with the company to interact with privately-run smart contracts that deal with everything from supply chains to employee management.

## o The Next Step in Managed Infrastructure

The Ethereum network, as it currently exists, is entirely dependent on several innovative and vital managed-infrastructure providers, most notably Infura. Infura makes it possible to interact with the Ethereum network without installing and maintaining costly Ethereum infrastructure, in addition to improving the ability of the network to scale at times of high demand. In essence, Infura is a managed Ethereum client node, with no direct awareness of smart contracts.

Link improves on this premise by providing a way for developers and users to interface with smart contracts from outside the Ethereum infrastructure layer. To appreciate these nuances, it is important to understand the differences in how various infrastructure options or services operate.

- **Client Applications** operate on the application layer, interacting with a smart contract as if it were a native RESTful web service;
- **Blockmason Link** operates on the web service layer, providing a RESTful interface for interacting with smart contracts;
- **Infura operates** operates on the infrastructure layer, providing a thin wrapper around direct interaction with the Ethereum network, with no special treatment for smart contracts;
- **Ethereum** operates on the foundational persistence and computation layer.

Now, let's examine a chart comparing the different aspects and features of various infrastructure for client application interface and end-user Ethereum network access.

	Self-hosted Ethereum Node	Infura	Blockmason Link
Does the user need a browser extension or hardware wallet?	✓	✓	✗
Does the user need to buy Ether (ETH)?	✓	✓	✗
Does the user have to pay transaction or gas fees?	✓	✓	✗
Does the user need to explicitly approve every state change?	✓	✓	✗
Does the DApp need a specialized library?	✓	✓	✗
Does the DApp need a blockchain-specific authentication scheme?	✓	✓	✗
Does the developer need to manage any extra infrastructure?	✓	✗	✗

As the table indicates, Link provides a vital service that builds upon existing IaaS architecture by simplifying the method of interfacing with Ethereum client applications. Link is, in essence, a smart contract gateway. What Infura does for the Ethereum network in general — increasing accessibility and usability through delegated management — Link does for smart contracts specifically. Link is the natural evolution of these essential services, opening up programmatic blockchains and their smart contracts to the general public for the very first time.

Link may be utilized by a wide variety of projects in any field and is applicable for everything from blockchain voting to supply chain verification. Link is not a single killer app; it is every killer app, a building block without which the Ethereum network, and blockchain in general, will soon be unimaginable.

## ◦ Technical Overview

Link itself runs as a web service deployed on Blockmason's cloud infrastructure. A developer may link a smart contract by sending the contract's address and ABI to Link. Link uses the contract's address and ABI to generate a Link app. After creating a Link app, its creator has access to each of these features:

### — Web service

HTTP endpoints for each function in the linked smart contract, and an endpoint for querying its events.

For example, a constant function called `getPizza` would be available via GET

<https://apps.blockmason.link/example-contract/getPizza>.

### — Access Control

Developers may define how clients are allowed to authenticate to the application. The most straightforward authentication strategy is to generate an API key. This API key has an Ethereum address that can be used to send ETH to cover transaction fees incurred by that API key. More advanced authentication strategies may involve federation to external identity providers via OAuth 2.0 or SAML.

Regardless of the authentication strategy, each identity used to interact with the app possesses an Ethereum address that can receive ETH to cover transaction fees.

### — Analytics

Link offers a variety of analytical tools for developers to inform their understanding of their app and how users are employing their smart contract. These tools include interactive charts that are searchable, filterable, sortable, and pivotable. Possible queries include request volume, gas cost, and error rate (total, per function call, per account). Using the ELK stack behind the scenes to create powerful visualization and search engines, Link provides analytic tools on par with products like Domo or Kibana, all within the Link user interface.

### — API Reference Documentation

HTTP endpoints for each function in the linked smart contract, and an endpoint for querying its events.

Conforms to the Open API Specification, a common standard for web API documentation that is both human-readable and machine-readable.

<https://swagger.io/resources/open-api/>

### — Client Software Developer Kits (SDKs)

SDKs — including screen builder, editor, compiler, and linker — for various programming languages that can be used by client apps to integrate with the Link app. Each SDK comes complete with documentation, example code, and package hosting.

- **Constant functions** are mapped to endpoints using the HTTP GET method.
- **Non-constant functions** are mapped to endpoints using the HTTP POST method.

- **Payable functions are ignored.** For example, functions involving the transfer of ETH (i.e: “payable” functions) are out of scope.
- **The HTTP resource path for the endpoint** is mapped to the function signature. For example, in a function adding a topping to a pizza order: `/addTopping(uint256)`.

After creating a Link app, its creator has access to each of these features. Additional features are available for managing and monitoring the app:

- **The inputs of a function** are mapped to query parameters (for GET requests), or to JSON key-value pairs in the request body (for POST requests). For example, a non-constant `addTopping(string name, uint256 count)` function would be mapped to `POST /addTopping` and expect a JSON request body that looks something like `{"name": "pepperoni", "count": 100}`.
- **Numeric inputs** may be provided as hex strings (e.g: `“0x1F2E”`) or as regular numbers (e.g: `123`).

### — Synchronized Versioning

The names of all endpoints and parameters can be renamed, and all documentation and SDKs will be kept in sync. When making changes, Link will automatically adjust the version of each client SDK according to semantic versioning (<https://semver.org/>) and will maintain backwards-compatible endpoints wherever possible to ensure compatibility with clients that may be using older versions.

### — Logs

A developer may also request logs on the application’s web service and audit them for administrative activity. For example, a developer may audit functions such as renaming endpoints and parameters or adding and removing accounts. These logs are interactive, searchable, filterable and sortable.

## ○ Security

Blockmason is fully committed to the protection of Link’s clients and users, as well as their data. We have taken the following steps to ensure our application is reliable and secure:

- Writing code according to best practices, including extensive peer review;
- Extensive and verifiable test coverage of application code, integration of static analysis tools into the continuous integration and deployment toolchain, and fully automated and auditable deployment infrastructure;
- Professional, independent auditing of smart contracts interacting with Ethereum and Link;
- Offering ‘bug bounties’ to encourage white-hat hackers to identify and eradicate coding errors;
- Undergoing quarterly penetration tests and other independent security audits as part of our general security hygiene.

Additionally, while some developers or users may be concerned about the risks of offloading certain transactions to third-party applications, Link poses no more risk of data exposure than using Ethereum itself. Every transaction is still logged on the Ethereum layer; Link only retains a temporary copy while waiting to deliver the results of a call or transaction.

Finally, even if a developer is using an account managed by Link, only a limited amount of funds to cover transaction costs are ever stored in that account. At any time, a developer may choose to connect a personal Ethereum account instead of using a Link-managed account. Link poses no risk to any connected accounts.

# Token Features, Usage + Economics

Blockmason has elected to use a traditional Software-as-a-Service (SaaS) model for Link, which is inline with the theme of a simple, familiar offering for our primary target: developers, especially those in small-to-medium businesses and enterprise settings. While the pricing model for Link is still in development, we intend to charge monthly subscription fees, API credit costs, or a combination of both. Developers and other Link customers can choose to pay these fees in one of two ways: using the Link utility token, or using a traditional payment method like a credit card.

**The Link utility token is the required "fuel" that provides API transaction capacity in Link.** If a Link user elects to pay their monthly subscription and usage fees with the Link utility token, a discount will be applied as Blockmason does not incur payment processing fees and other costs associated with credit card processing.

If the user instead pays with a credit card, Blockmason will use the revenue to acquire Link utility tokens on the user's behalf and apply them to their fees. Initially, these will be drawn from Blockmason's reserve inventory pool. After the reserve is depleted, Blockmason will acquire Link utility tokens on a pro-rata basis from all blockchains where the token is available at that time.

Regardless of how a Link user pays for their usage and subscription fees, the resulting tokens will be locked and removed from the circulating supply until 30% of the tokens remain in circulation. Therefore, all Link utility tokens used to access or pay for the Link platform will be locked up at this initial stage, prior to reaching the 70% token supply lockup target.

When the circulating token supply reaches the 70% lockup target, the functionality of the tokens will change to a membership model in the form of "staking." The token will no longer be consumed when using Link services, providing Link utility token holders with Link platform access as long as they maintain their stake.

Blockmason will conduct these supply removals on an ongoing basis, with regular reporting and independent audits. We are confident that this will provide transparency and visibility into how the Link utility token operates and allow any observer the chance to verify Link user growth and adoption metrics.

As Link is designed to support a variety of programmatic blockchains, the Link utility token will be minted with this in mind. Initially, Link will support the use of smart contracts on the Ethereum network. As such, 100% of the total Link utility token supply will be minted on the Ethereum network as ERC-20 compliant tokens. Link utility token support for alternate blockchains will be added in the future, in line with Link's development roadmap. When this occurs, Blockmason will act to reduce the ERC-20 token supply via lockup to ensure the total Link utility token supply across all blockchains remains constant.

Key details regarding the tokenomics of the Link utility token are provided in the table on the next page.

<b>Total Link utility token supply</b>	5,000,000,000
<b>Initial IEO price per token</b>	USD \$0.01
<b>% of total Link utility token supply intended for Ethereum blockchain</b>	35 %
<b>% of total Link utility token supply reserved for alternative blockchains</b>	35 %
<b>Marketing, bounty &amp; airdrop allocation</b>	10 %
<b>Advisor allocation</b>	5 %
<b>Team allocation</b>	15 %
<b>Circulation lockup target</b>	Remove + lock 70 % of total circulating supply
<b>Circulation lockup frequency</b>	Monthly

In lieu of a traditional ICO, Blockmason has elected to fund the development of Link via an IEO (initial exchange offering) with our trusted exchange partners. This "early adopter" access to the Link utility token allows developers, startups, entrepreneurs and other potential Link customers the opportunity to pre-purchase their API transaction capacity.

An IEO run by trusted, professional exchanges also avoids the many potential difficulties that come along with a traditional ICO, like scammers and other nefarious actors. It also provides an automatic post-IEO exchange listing and immediate legitimacy for the Link project thanks to the exchange partners' rigorous due diligence.

Blockmason firmly maintains that the Link utility token is a product-use token. It is used to provide Link API transaction capacity and to cover monthly access fees. However, given the continued scrutiny and lack of clarity regarding regulatory issues, the Link IEO will exclude citizens, permanent residents and passport holders of the United States, China and India.

# Revenue + Business Models

To ensure the immediate, tangible benefit to the blockchain community upon Link's release, Blockmason is committed to providing the same quality of robust support and maintenance that we have provided since our first products, the Credit Protocol and Lndr, were launched a year ago. Link will support, if a developer so chooses, the ability to sell licenses to use a DApp in which Link will issue the paying user or client developer a credential to access the application. This credential can be restricted at the developer's discretion, including limits on credential length, or transaction capacity.

Another use case Link will support is the formation of a large open source database of smart contracts with basic functions, thus enabling for the first time hybrid blockchain microservices, allowing developers to weave a complex tapestry of applications with each component precisely tuned to its unique function. Additionally, such a database can allow developers to save time and energy by building on the work of other programmers, picking and choosing which basic smart contracts to employ and which to build from scratch, along with a community rating system in which other developers can provide feedback on the functionality of a particular application or smart contract. Other considerations for the Link SaaS model include:

- A "freemium" Link offering, with the base tier including one free DApp and one free, rate-limited client per user;
- Pricing tiers structured around:
  - Number of users (i.e. end-user credentials for mobile, desktop users);
  - Number of linked smart contracts;
  - Number of API requests being made;
  - Multi-user accounts (i.e. organizations or teams).

In addition, fees may be charged for providing priority customer support and 'hands-free' transaction fee handling for developers.

Blockmason is interested in growing Link by building a community of developers and clients enthusiastic about the future of blockchain technology. We are interested in long-term partnerships as well as a strong and social Link community. We believe that a traditional SaaS model will aid us in achieving these goals.

## — Premium Management Services

If any of the aspects of account management seem overwhelming, Blockmason can manage a developer's application. Services may include ensuring all Ethereum addresses in an account hold sufficient ETH for gas, handling user onboarding for the application, and managing payments for client application usage.

# Development Roadmap

This is the current edition of the development roadmap and is subject to change:

## — Q4 2018 — Link “Demo” Release — Fully-functional Product Demo

A completely functional end-to-end demonstration experience of linking a smart contract and using Link to execute functions on it.

## — Q4 2018 — Link “Series A” Fundraising Round

In lieu of an initial coin offering, Blockmason has elected to fund the development of Link via a traditional equity-based “Series A” financing round. This private sale will be made available to close partners and associates who believe in Link as a product with lasting business potential.

## — Q1 2019 — Link “Developers” Release — Individual Developers Can Apply for Access

Individual developers who want to use their own smart contracts with Link can apply for access and begin using Link. Features including self-registration, polished and finished front-end user interface for managing Link apps, self-funding of Link accounts and more.

## — Q2-Q3 2019 — Programmable Blockchain Support Expansion

Link will expand to include support for programmable blockchains outside of Ethereum.

## — Q3-Q4 2019 — Link “SMB” Release — Ready for Small to Medium-Sized Business Clients

Ready for small to medium-sized business clients who want to build or port some of their back-end infrastructure over to Link. Features including support for multiple users, access control, Link app usage analytics, deeper customization and more.

## — Q1-Q2 2020 — Link “Enterprise” Release — Ready for Enterprise-Scale Clients

Ready for enterprise-scale clients who are ready to use Link across hundreds or thousands of apps, users and clients. Features including single-sign-on, directory integration, auto-provisioning of accounts, highly detailed usage analytics, sophisticated filtering, sorting and search of apps, users and clients, individual access control and more.

## — Future — Custom-Requested and Private Blockchain Support

Based on client feedback and requests, Link will expand to support private programmable blockchains.

## — Future — Credential Vending Support

Link makes managing and monetizing smart contract applications easier than ever. If a developer so chooses, Link may sell licenses to use an app. Link will issue the paying user or client developer a credential to access the application. This credential can be restricted at the developer’s discretion, including limits on credential length, or transaction capacity.

### — Future — Smart Contracts Database + Link App Store

By registering so many smart contracts in one location, Link will make possible the formation of a massive open-source database of smart contracts with basic functions. For the first time, hybrid blockchain microservices will be possible, allowing developers to weave a complex tapestry of applications with each component precisely tuned to its unique function.

### — Future — Professional Standards Compliance

Security certification compliance is one of the most important — and difficult — aspects of marketing applications to larger businesses and enterprise customers. Many such businesses require compliance with any number of standards including SOC 2, ISO 27001, HIPAA, and FIPS. Blockmason intends to pursue these certifications for Link, and may then manage compliance the applications registered with Link, saving developers significant time, energy, and money.

# Use of Funds

The majority of the funds raised to support the launch and growth of Link will be committed to the development of the platform. As Link becomes more feature complete and we begin to add developers and other customers, our marketing, operations, growth and support departments will expand to ensure that we can maintain a high level of service.

Here is an estimated breakdown of how we plan to dedicate the funds we raise to build, launch and grow Link:

## — Link Platform Development (40-50%)

The majority of funds raised will be used to develop Link per the milestones laid out in the Development Roadmap section. Necessary investments include new hires and salaries for our development team, hardware and software infrastructure and more.

## — Marketing and New Customer Acquisition (20-30%)

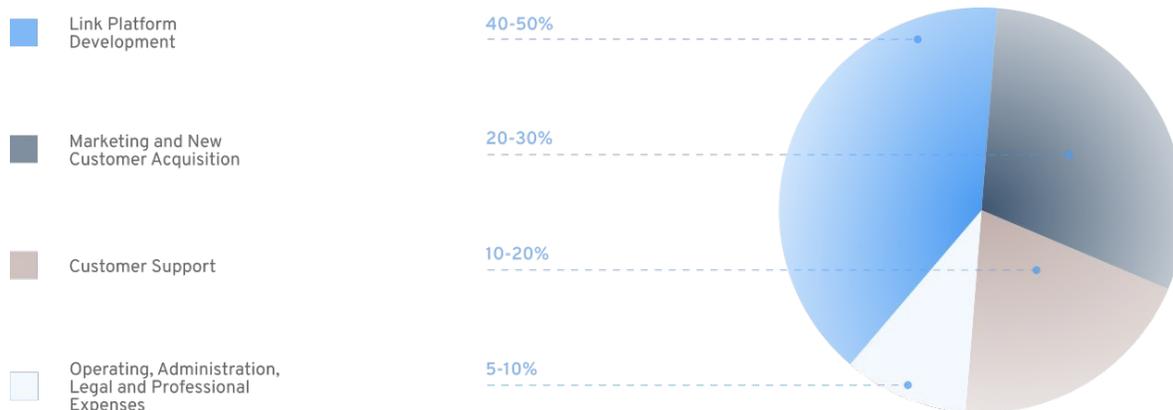
New, paying customers are the lifeblood of any business. Our marketing plan includes digital advertising, email marketing, social media marketing, event sponsorships, hosting development hackathons and more. This may also include new hires or the outsourcing of work to specialized agencies.

## — Customer Support (10-20%)

As we continue to add new freemium and paying customers who have mission-critical application needs, our need to provide immediate, high-level support will rise. Our customer support expenses include new hires to provide 24-hour assistance, hardware, software and ticket response systems, documentation, maintenance and other emergency services.

## — Operating, Administration, Legal and Professional Expenses (5-10%)

Finally, a portion of our funding will be used to cover traditional operating expenses. This can include salaries, office space, computer hardware and software, legal fees, accounting fees and other miscellaneous costs.



# Meet the Blockmason Team



## Michael Chin CEO

After growing up in Seattle, Michael graduated magna cum laude from Washington University in St. Louis with degrees in finance and accounting. He began his career at J.P. Morgan's principal mezzanine credit investment group in New York, then in JPM's Debt Capital Markets and Syndicated Leveraged Finance group in Hong Kong. Afterward, Michael joined Intermediate Capital Group's investment team, a FTSE 250 company with EUR 20 billion in assets under management, and subsequently served as a Managing Director at Constance Capital specializing in structured financing deals. He then co-founded BlockMason, which created the BlockMason Credit Protocol, a system for permanently recording ledgers of debts and credits between parties using the Ethereum blockchain. Michael is an investor in technology companies and properties, and currently splits his time between Vancouver and Hong Kong.



## Devin Canterbury CTO

Devin is a distinguished generalist with a deep background in applied cryptography. He is a steward of software craftsmanship and professionalism, applying sustainable and scalable practices to building libraries, applications, and teams. He has served as CTO for transportation services startup, Roadify; founder of Amused Pony Games (developer of Axes of Evil, the luxury zero-player RPG); Principal Software Engineer at Silent Circle (purveyor of encrypted communications); and Principal Software Engineer at Outreach.io (a scalable communications platform for sales teams). Devin hails from north Louisiana and graduated with a B.S. in Computer Science from the University of Louisiana at Monroe, before moving to Seattle in 2007. Now, Devin is building the next generation of blockchain technology with Blockmason, laying the foundation for a new era of massively distributed applications.



## Erik MacKinnon CMO

Having launched and successfully exited his first startup during the original 'dot com boom', Erik is an accomplished entrepreneur with a passion for building technology companies that disrupt established markets. As Chief Marketing Officer, Erik is excited to introduce Blockmason's innovative base-layer blockchain technologies to the world. Before joining Blockmason, Erik served as paid media and digital strategy consultant to SAP, the world's premier enterprise software company. Prior to SAP, Erik was Chief Executive Officer of Gravytrain, a boutique content marketing and digital advertising agency he founded in 2012.



## Ken Cornutt Director of Finance

A Seattle native currently living in Los Angeles, Ken has a long history working in Chief Financial Officer roles with growing startups. a Certified Public Accountant and member of the AICPA, Ken served as Senior Auditor with Deloitte & Touche before stepping into the startup arena. Outside of work, Ken is a competitive chess enthusiast.



## **William Galebach** *Software Engineer*

Will is a full-stack software engineer with a background in web and mobile apps and dapps, with a focus on user experience and scalability. He previously worked as a Senior Software Engineer at Farmers Insurance and as an intelligence officer in the Marines. Will is passionate about making blockchain technology accessible for both developers and consumers.



## **Harish Raisinghani** *Business Development Manager*

Harish is an all-round generalist having worked as a systems engineer, software developer, project manager, sales support and general manager for the cleantech startup Pulse Energy, which was acquired by Yardi Systems. He is passionate about blockchain technology and has experience investing in cryptocurrencies and building smart contracts and trading bots. Harish grew up and resides in Metro Vancouver and is an Engineering Physics graduate from the University of British Columbia. At Blockmason, Harish manages business development and operations in North America.



## **Clayton Savage** *Digital Marketing Specialist*

Clayton Savage has joined Blockmason as a digital marketing specialist. He has a lifelong passion for business, technology and human behavior. Before joining Blockmason, Clayton worked as a digital marketing specialist for the world's leading knowledge management solution provider. Clayton completed a four-year Business Management degree at the University of British Columbia. While at UBC he competed in national business competitions, lead as a Marketing Director for the Management Students Association and acted as the Sector Head of Technology and Telecommunication for the UBC Portfolio Management Group. He shares a strong belief in craftsmanship and sweat equity that goes into achieving the vision of Blockmason's technology.

# Disclaimer

This White Paper presents a general informational overview of Blockmason’s current intentions with respect to Link, as of the date of this document. Prospective Link users should rely on their own evaluation of Link, rather than on this White Paper or any other statements, written or verbal, about Link; Our products are offered “AS IS” — purchasers make their own judgments based on the beta versions and documentation for the products. We make no warranties, and we expressly disclaim all warranties, express or implied, including the warranties of merchantability, fitness for a particular purpose, and non-infringement. We cannot guarantee against service interruptions or make warranties about the Ethereum network or other programmatic blockchains. Purchasers are responsible for consulting their own country’s laws and regulations. Purchases of our Link product should be based on each purchaser’s intended use and development of the Link product as well as each purchaser’s evaluation of Link based on its beta version and open source code.

Title and ownership of Link and all related documentation and support structure resides in Blockmason. Blockmason reserves the right to adjust the amount of network capacity, and the number of transactions allocated to each Link user, using reasonable efforts to maintain efficient network operation and avoid congestion.

# Glossary

- **ABI**

Abbreviation for “Application Binary Interface,” an Ethereum concept for defining how a smart contract can be used

- **API**

Abbreviation for “Application Programming Interface,” a generic term for how one unit of software interacts with another

- **HTTP Endpoint / REST API Endpoint**

A technical term referring to an HTTP method and resource path. a common way for web servers to expose functionality for other software (including web browsers) to execute

- **ETH**

The native cryptocurrency of the Ethereum blockchain

- **Gas**

A common term in Ethereum to describe transaction fees

- **ICO**

Abbreviation for “Initial Coin Offering,” a common funding mechanism for blockchain-related projects whereby a new blockchain-based token is sold to the general public, often to serve as a license or usage right for a blockchain-based product, service or network

- **Gas**

The name of the protocol typically used by DApps and other software clients for interacting with Ethereum network.

- **Metamask**

The name of a popular extension for Google Chrome that provides a software wallet for ETH and other tokens operating on the Ethereum network; a common requirement for many decentralized applications

- **Microservices**

An architecture pattern for server software in which, instead of one monolithic server handling all logic, logic is distributed across many loosely coupled services, each handling a small domain of responsibility

- **RESTful Web Service**

A common design pattern for server software specifying how HTTP methods and resource paths map to actions and resources

- **Solidity**

A popular programming language for writing Ethereum smart contracts

- **Smart Contract**

A concept native to the Ethereum blockchain, meaning a unit of software deployed to and which runs on the Ethereum network

- **Web3**

The name of Ethereum’s official library for software written in JavaScript to use when interacting with the Ethereum network

- **DApp**

Commonly referred to as “decentralized application,” the term describes any user-facing app that interacts with the Ethereum network

# Thanks for your attention!

