



## 1 INTRODUCTION

---

Creating applications to realize a project or business idea has long been too difficult and expensive. Over the last decade we have seen successful tools built to simplify website creation and e-commerce that relieve the need for extensive coding proficiency. However, the same can't be said in the domain of microservice architectures or decentralized applications where a lack of tools exist to simplify the process.

The blockchain has begun to constitute the potential to decentralize our world. At an accelerating pace individuals and companies are attempting to reinvent industries by creating decentralized applications (DApps) executed through protocols such as the Ethereum Virtual Machine (EVM). As of December 2017, two years after the launch of Ethereum, there are over 840 decentralized applications. This curated list merely represents a fraction of the ecosystem. We have also seen a sharp increase in funding to projects building decentralized applications through Initial Coin Offerings (ICOs) and traditional venture capitals that amounts to billions of USD.

Yet with decentralization comes an added layer of complexity, and alas with greater complexity development costs inflate. The EVM is remarkable at facilitating consensus across the blockchain and executing unstoppable code, however it does not solve all our problems. As a result, many projects falter or are severely limited by running in a distributed environment. The roadblock becomes a bottle neck for the plethora of great ideas attempting to fulfil their intended visions.

Despite the added layer of complexity, some off-chain projects (e.g. Golem, iEx.ec, Swarm, IPFS, CloudWith.me) still overcome early obstacles and offer the potential to disrupt old conventions. Yet to build a healthy ecosystem we need more individuals and companies to experiment, fail cheaply, and succeed. Blockie has for the past few years been dedicated to creating tools which are aimed at making software and service development, deployment, and management accessible for more visionaries. With the potential that blockchain and smart contracts permit, we see an even more important role for these tools to help close the gap between who is a producer and who is a consumer on the internet.

As a community, we need better tools to code smart contracts and connect these pieces of code to off-chain services without neglecting decentralization whilst reducing their inherent limitations. We need a simple and intuitive process to create decentralized applications that can leverage a private cloud of powerful analytics software that can store and secure user data in specific regions of the world to keep it secure and GDPR/HIPAA compliant. We need tools to build Decentralized Autonomous Organizations that can leverage artificial intelligence and big data for decision making, or simply reach out to APIs in the Web 2.0 to order a pizza.

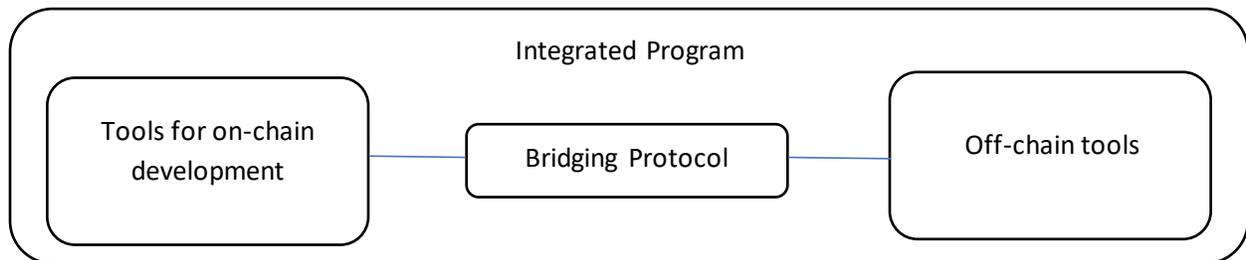
Mailchimp made email marketing easy, Squarespace made website creation easy, Blockie will make DApp creation easy. We are here to empower more visionaries to "build their thing" and create the decentralized future.

## 2 AN OVERARCHING VIEW OF BLOCKIE TECHNOLOGIES

---

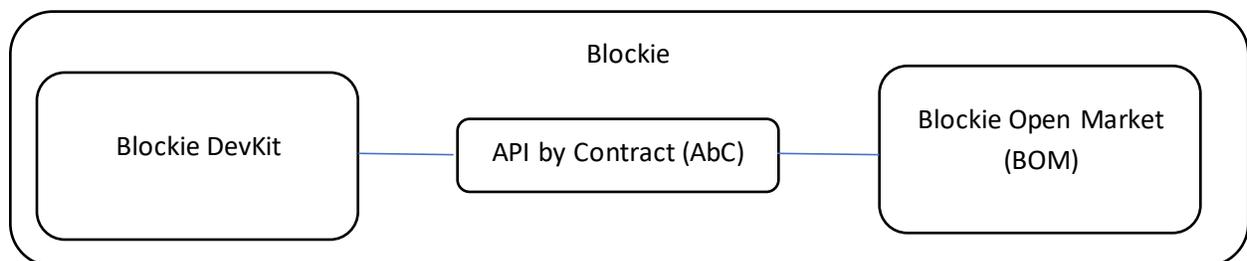
Blockie takes an overarching and holistic perspective to solve how we can create DApps that are as powerful and versatile as microservices, GDPR/HIPAA compliant, and developed by non-coders. Blockie must hence build or integrate open-source tools that have the capabilities to:

1. Create and deploy smart contracts in an approachable manner (on-chain tools)
2. Connect smart contracts with Web 2.0 to remove their inherent limitations, whilst remaining genuine to the concept of decentralization (bridging protocol)
3. Manage these interactions off-chain and aid in developing useful services on Web 2.0 (off-chain tools)
4. Execute steps 1-3 in one multifaceted program that can eventually be used by non-coders



Moving across this spectrum must be made easy if we are to truly decentralize and disrupt many industries in a realistic and meaningful way. The on-chain tools need to support the bridging protocol that connects you to the off-chain tools, and vice versa. The entire process should be manageable through one program that integrates all the necessary components. Blockie has solutions that will cover the base necessities to create a working integrated program, yet will look to partner with providers that specialize in certain areas (e.g. smart contract/DAO template providers) and can add value to the process to the entirety. Our aim is not to necessarily compete on all fronts, rather to ensure that a holistic and integrated solution remains in focus. The steps between envisioning a DApp idea to having a finished product should be a straightforward process guided by a single program that combines the best of all tools to reach the intended outcome in the quickest manner possible.

The basic default components of our integrated program built by Blockie solves the core problems. The on-chain tool called "Blockie DevKit" allows you to create, manage, and deploy smart contracts. The bridging protocol "API by Contract" makes them as powerful and versatile as microservices whilst GDPR/HIPAA compliant. Finally, our off-chain tools comprised of several layers bundled into the "Blockie Open Market" will enable you to develop, manage, sell and buy services that make your off-chain DApp management a fluid process.



The solution will not be all-encompassing at the start, yet overtime the interactions will flow together. Eventually non-coders will be able to interact from all fronts of the application. In areas that are temporarily too technically challenging or unique, they will be able to seek assistance in the Blockie Open Market.

### 3 THE BLOCKIE DEVKIT

---

A tool to help code smart contracts and deploy them. More information including an alpha release coming in December.

### 4 API BY CONTRACT - ABC

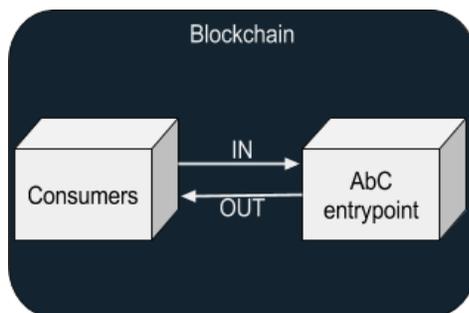
---

Blockie product “API by Contract (AbC)” handles the communication between smart contracts and the cloud. It serves as a common API entry point for consumers (smart contracts), meaning it is possible to build powerful decentralized services. It will allow smart contracts to connect to Artificial Intelligence, Big Data, traditional e-commerce, servers for computer storage or computing power, and anything else that is found in the Web 2.0. This is extremely powerful for Decentralized Autonomous Organizations (DAOs) that will need to interact with the Web 2.0 for both decision-making and operational purposes.

Yet, if we turn to cloud computing, it does beg the question: are we not re-centralizing our efforts again? That depends on whether those cloud servers can be taken down by any single authority. If they can't, then we have a decentralized smart contract running a decentralized cloud computing service. There are four main points to all of this:

1. The smart contract must be able to launch the services it depends on autonomously.
2. The smart contract must be able to communicate with the services after they are launched.
3. The services which the smart contract launches in the cloud must be resilient to takedown.
4. Preferably, these services should not require their creators to be extremely skilled coders.

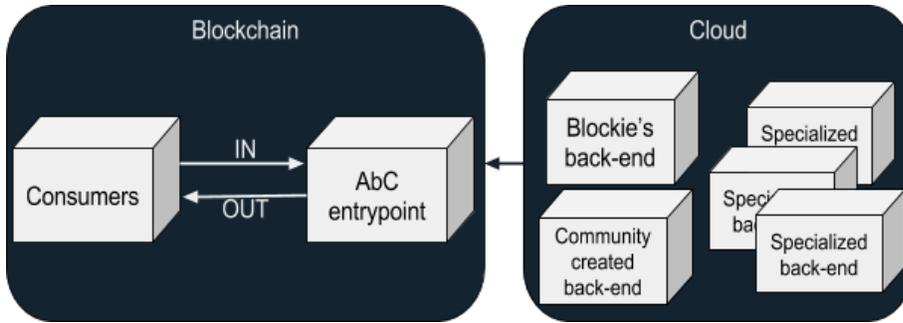
For the smart contract to be able to launch and communicate with a cloud service, there must be some cloud-based processes that handle this action for the contract. This is not a problem per se. The big challenge comes when decentralizing it so that no one can stop this from happening.



In practice AbC is defined by a protocol that involves components running and cooperating both on current blockchain technology and in the cloud. One must run a form of AbC both on the blockchain and in the cloud so that these two counterparts speak the same language. The first AbC core component resides on the blockchain embedded in a set of smart contracts. We call this component the “AbC entry point” as these smart contracts manage the consumer-facing part of the AbC protocol. By consumer, we refer to any Ethereum address, smart contract, or DAO (a collection of smart contracts) that requires AbC services.

The AbC entry point routes requests from other smart contracts to their respective back-ends by registering new tasks. The AbC smart contracts tag the tasks with a specialized category and are optionally encrypted to map to a back-end processor as pending work. In technical terms, this is an event. AbC writes this data to the blockchain or logs it in the transaction receipt. In case of encrypted messages, the event is recorded in the function of a public key signature connected to a specific back-end service required for fulfilling the request.

The second core component of AbC is attached to back-ends which define the way services will be implemented. They react to task requests by connecting with AbC entry point, after agreeing which one will handle the next job within that particular network of nodes, considering their domain of expertise. That is, a back-end specialized in distributed storage will not be able to fulfil a payment request on its own. The back-ends will consider which category the tasks pertain to. After that, the chosen back-end will complete the given task and return the information back to AbC, together with the task identifier and signed by a back-end signature which can be verified.



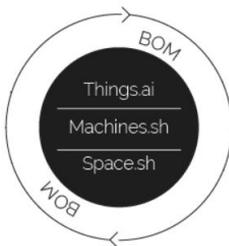
AbC back-end services can be composed of one or more nodes. They are initially provided by Blockie as a standard gateway but can be replaced by any participant of the system with their own services ensuring openness and customization of the system. Because of these “detachable” back-ends, a DAO or a single smart contract will be able to customize the endpoints in a transparent and seamless way. Developing back-ends is an advanced topic and will be further explained in Blockie’s technical yellow paper.

The evolution of the architecture is aimed towards making it resilient to takedowns. Should any cloud provider decide to turn off our cloud machines, the smart contract can just respawn them somewhere else thanks to AbC responsiveness and awareness of the participant nodes. If one cloud-based process is taken down, there must be an incentivized network which will make sure the smart contract can always spawn its services on some other cloud provider. That is the beauty of AbC; it is built to persist.

## 5 THE BLOCKIE OPEN MARKET

Blockie set out on a mission long time ago to enable anyone who wants to take part in software creation, to produce content and deploy it in a user friendly and reliable way. Blockie believes that individuals, communities and organizations should have access to tools and services which enable them to create software, either new or derivative. They should be given a chance to solve problems regardless of their coding proficiency while running their creations as decentralized services. We need to start closing the gap between consumers and producers of services on the internet. BOM will close this gap. It is a web portal where users come to develop (decentralized) services, taking advantage of a marketplace for publishing original content or reuse existing solutions to create potentially unlimited variations of services and products.

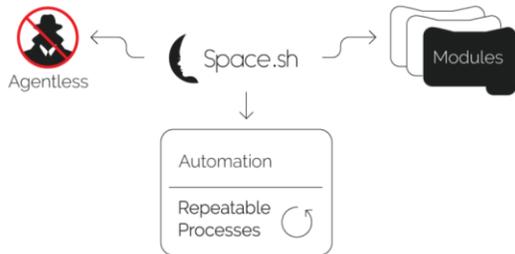
By leveraging the Ethereum network, users will be paid for software creations or pay others for their coding help using microtransactions. Furthermore, contributors can be rewarded for other tasks performed in the portal which increases the overall usability. By enabling users to contribute unique pieces of software to BOM, or take coding tasks to create specific software, BOM resembles a job market. But it is conceptually different. It is designed to break up big problems into smaller more comprehensible chunks.



In order to achieve this, BOM’s core is comprised of different layers. Each layer relates to a different set of expertise in the software development process — from infrastructural to application level. These layers are called Space, Machines and Things. This way all users, regardless of their software developing skills, can either contribute or seek solutions for specific tasks. The Space layer represents the foundational functionalities, Machines the creation and management of infrastructure and Things the composing of functionality in a block-on-block “lego-like” approach.

## 5.1 SPACE.SH

Space.sh is the foundation. The first layer in the stack is an advanced, free of charge, open-source, modular software tool that runs practically anywhere and requires very minimal setups to do so. It can be used for developing any type of application, extensions or automation recipes. The main aspect of Space is that anyone can build upon it.

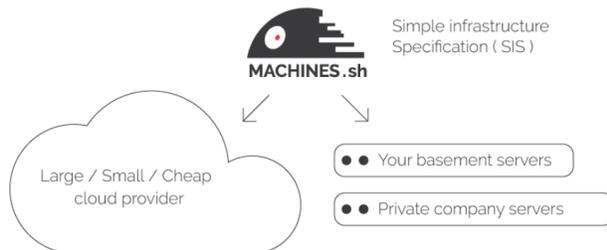


The most common use so far has been for creating multi-platform, portable server applications and performing general automation, in particular provisioning and administrative tasks on servers without the need of pre-installed software, agents or any other complex setup. Other use cases involve remote controlling computer resources, even in the most intricate environments that require dynamic re-routing, passing through firewalls and multiple layers of authentication.

Space was first made publicly available in November 2016 and currently offers more than 20 public modules that extend core functionalities. It can be found at <https://space.sh/>.

## 5.2 MACHINES.SH

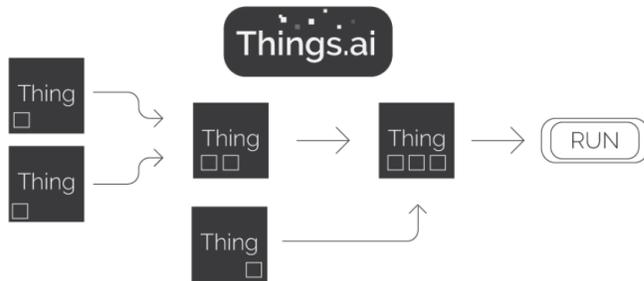
Machines.sh is the second layer in the stack, and is comprised of technical specifications for virtual resources. It handles provisioning of virtual machines, network interfaces and storage in a provider-agnostic way. Commonly referred to as dev-ops, Machines is used for the creation and management of server infrastructure, regardless of the host provider, in a very accessible way by using a simple descriptive language.



## 5.3 THINGS.AI

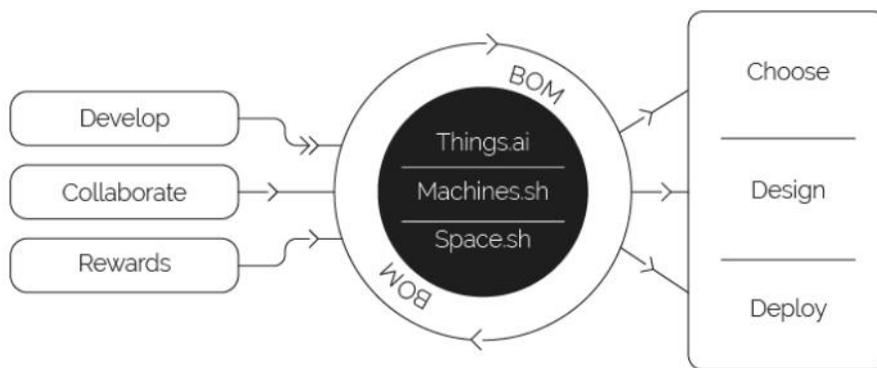
Things.ai is the third layer of the BOM structure. Things contains platform-independent software services and microservices designed in such a way that it is possible to stack them together into new, more competent, pieces of software. A new Thing element can then be reused in some other Thing composition, which can be returned back to the pool and so on in a potentially endless cycle of customization, improvements and ever growing competence.

The “magic” happens when users start to combine other users’ Things into their own Things. We envision a rapidly growing pool of Things that will accommodate ever more complex and fulfilling services for an audience that should swell in numbers over the years — hence fulfilling Blockie’s vision about enabling non-coders to create software. Things are based on containerized reusable application compositions, taking advantage of containerization provided by Docker and Kubernetes to deploy working services.



## 5.4 THE BLOCKIE OPEN MARKET WEB PORTAL

The Blockie Open Market (BOM) web portal acts as a hub for connecting peers at a common marketplace. In each discrete layer of BOM, there will be many building blocks to pick and choose from when building your own BOM project. If you are missing a specific functionality or you need assistance in any layer or about any specific building block you can get help from your peers in the BOM community. Pick and choose, mix and match functionality blocks.



BOM creations in conjunction with API by Contract (AbC) will enable decentralized applications to take advantage of existing cloud-based software, including all open source software such as Tensorflow A.I. for machine learning and intelligence, Apache Spark for data processing, Storm for real-time processing and Hadoop for distributed storage and computing. This enables intercommunication between blockchain-based applications and highly efficient open source software, offering these new creations the ability to expand their environment and execution boundaries. For example, smart contracts can use AbC in conjunction with Machines to launch a cloud service built with BOM, without any human interactions.

BOM enables participants to share work on each layer by offering or asking for help and get paid for their work with Blockie tokens. As an example, non-coders will be able to outsource one or more challenges in different layers of expertise, in order to build their own BOM project.

Idealistically, the goal of BOM is to become an expanding community portal, filled with working parts and components that can work on their own or be used as a principal part of a new creation in an easy and intuitive way. Users are then able to choose from, reuse and combine all the various Things into endless possibilities. The need for empowerment, to let people have the choice of building or customizing applications in a world that is so involved with technology is something Blockie deeply cares about.