Blockchain Application Builder for Ethereum dApps

No Coding Required

White Paper

1. dApp Introduction

The emergence of dApps

A new model for building successful and massively scalable applications is emerging

WHY YOU SHOULD EMBRACE DAPPS

A **dApp** is an open source application that allows every individual that contributes to its development, design, and publishing to partake in Token contributions. A decentralized application does not require a leader or central node that gives restrictions.

Definition of a dApp

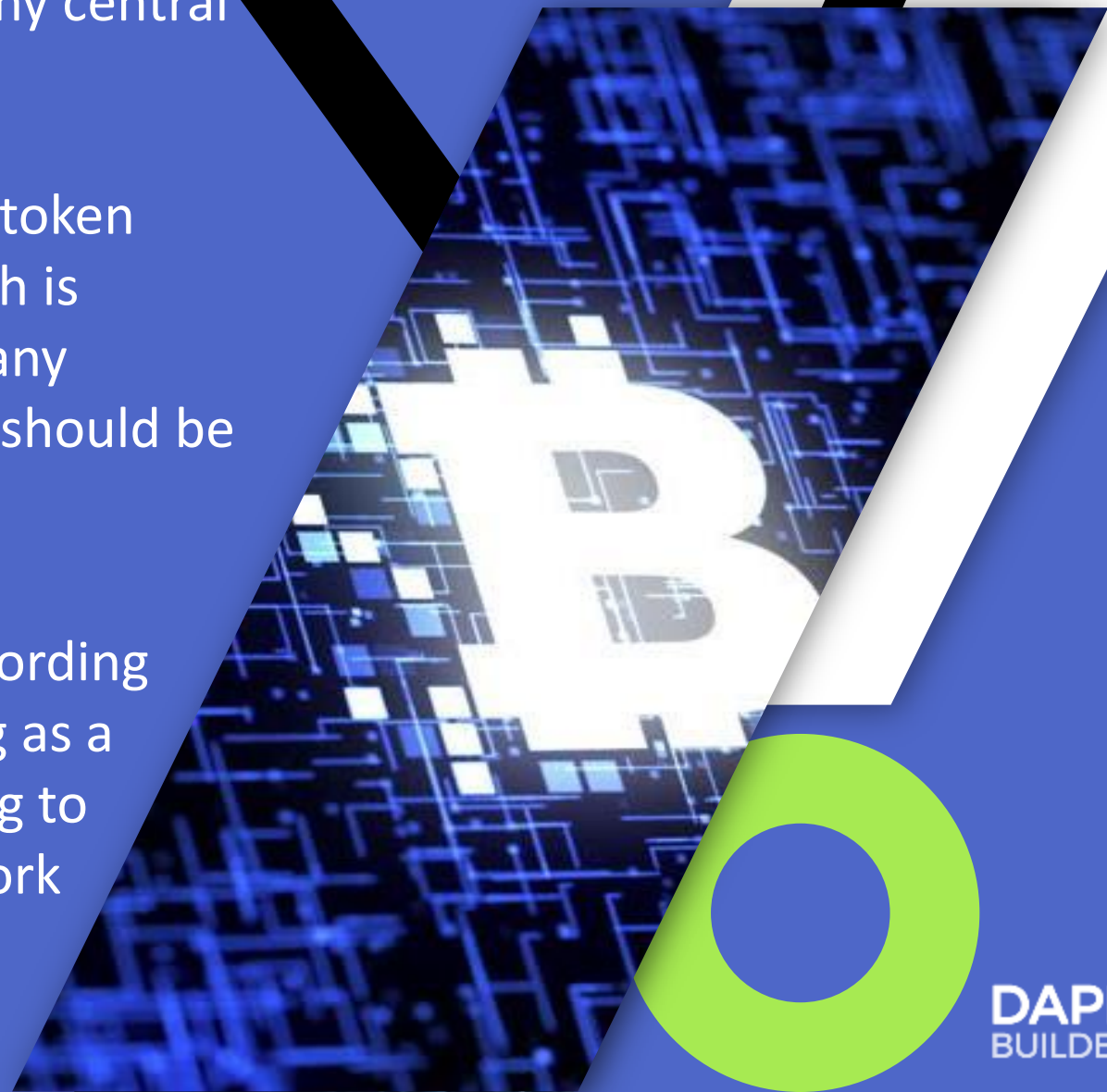
Any dApp must adhere to the following criteria:

1. The application must be completely open-source, it must operate autonomously, and with no entity controlling the majority of its tokens. The application may adapt its protocol in response to proposed improvements and market feedback but all changes must be decided by consensus of its users.

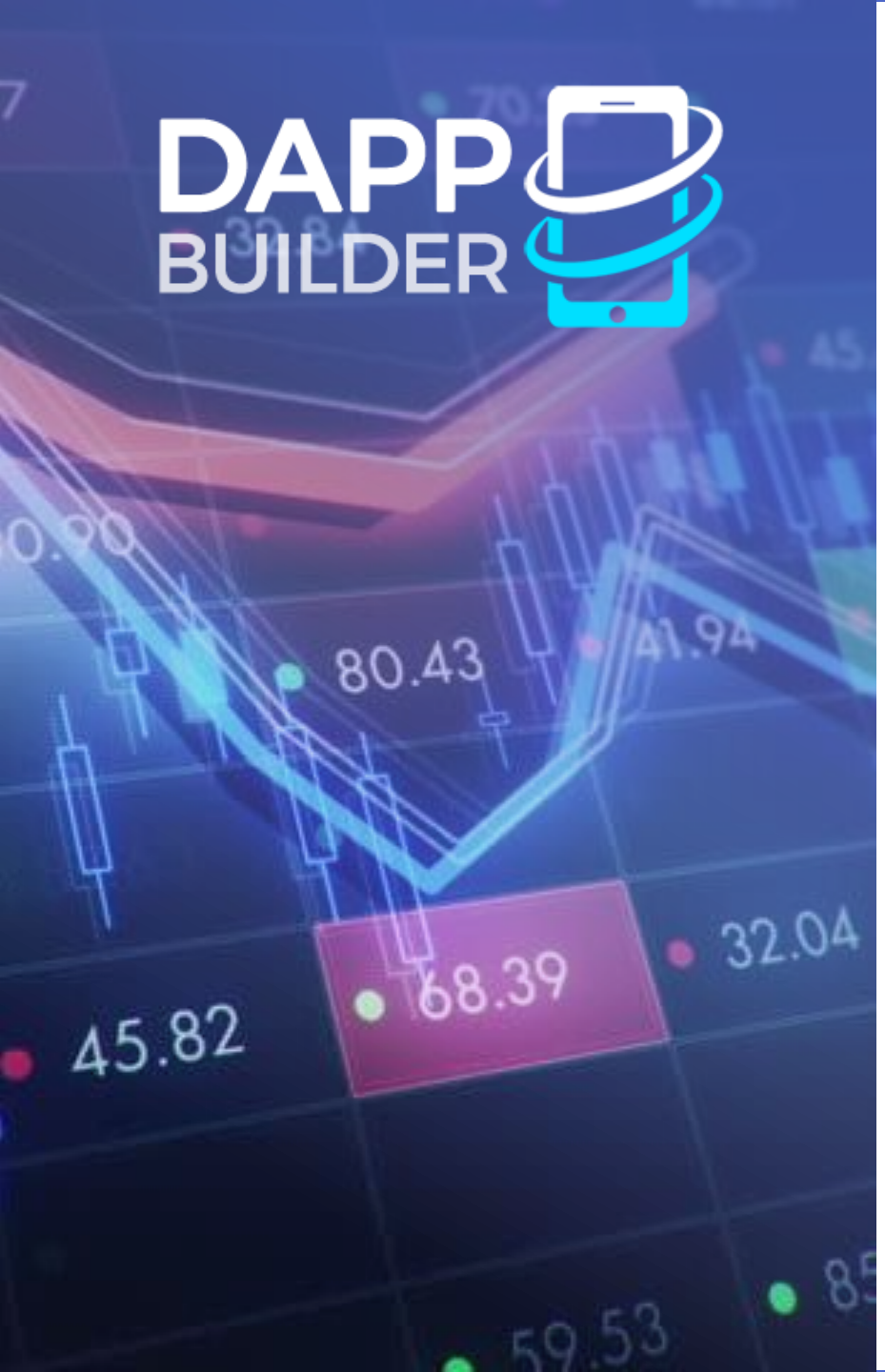
2. The application's data and records of operation must be cryptographically stored in a public, decentralized **blockchain** in order to avoid any central points of failure.

3. The application must use a cryptographic token (bitcoin or a token native to its system) which is necessary for access to the application and any contribution of value from miners / farmers should be rewarded with the application's tokens.

4. The application must generate tokens according to a standard cryptographic algorithm acting as a proof of the value the nodes are contributing to the application (Bitcoin uses the Proof of Work Algorithm).



DAPP BUILDER



Bitcoin as a Dapp

Bitcoin has been effective in solving the problems that arise from a trust-less and scalable electronic cash system by using a peer-to-peer, distributed ledger, the Bitcoin blockchain. In addition to being a peer-to-peer electronic cash system however, Bitcoin is also an application that users can interact with through computer software. But most importantly, Bitcoin is a decentralized application.

Please see below why:

1. All Bitcoin software applications are open-source, no entity (government, company, or organization) controls Bitcoin and all records related to the use of Bitcoin are open and public.

2. Bitcoin generates its tokens, the bitcoins, with a predetermined algorithm that cannot be changed, and those tokens are necessary for Bitcoin to function. Bitcoin miners are rewarded with bitcoins for their contributions in securing the Bitcoin network.

3. All changes to Bitcoin must be approved by a majority consensus of its users through the proof-of-work mechanism

2. Classification of **Dapps**

There are several characteristics according to which decentralized applications can be classified. For the purposes of this paper, we will classify Dapps based on whether they have their own block chain or they use the block chain of another Dapp. Based on this criterion, there are three types of Dapps.

01

Type I

decentralized applications have their own block chain. Bitcoin is the most famous example of a type I decentralized application but Litecoin and other “alt-coins” are of the same type.

02

Type II

decentralized applications use the block chain of a type I decentralized application. Type II decentralized applications are protocols and have tokens that are necessary for their function. The Omni Protocol is an example of a type II decentralized application.

03

Type III

decentralized applications use the protocol of a type II decentralized application. Type III decentralized applications are protocols and have tokens that are necessary for their function. For example the SAFE Network that uses the Omni Protocol to issue ‘safecoins’ that can be used to acquire distributed file storage is an example of a type III decentralized application.

3. Executive **Summary**

- **dApp Builder (DAP) is a SaaS platform for businesses** to create their own Ethereum dApps. It is our vision that anyone will be able to create a secure, flexible and legally binding dApp based on smart contracts by using DAP Platform, no coding required.
- **dApp Builder (DAP platform)** is about to enable mainstream business blockchain applications to benefit from deploying their own branded crypto-backed token economies, without a need to support their own publicly-tradeable ERC-20 tokens.
- With **DAP it is easy to create and edit dApps** with no coding skills, similar to how Wordpress and Wix allow for the creation of websites easily.
- **DAP platform is end-to-end, No Coding** skills will be needed as businesses/users can easily move into the blockchain world; project owners will be able to easily activate “Smart token” by just selecting a few parameters in the UI interface like Smart-token activated and constant reserve ratio %.
- **Users/businesses can easily create decentralized applications.** These decentralized applications will be able to launch branded tokens powered by DAP Smart Token.
- In this case the **dApp Builder DAP token** will become a Network token for the tokens created on the platform.

- **Building dApps on the DAP Platform** also enables businesses to benefit from network effects across the participating companies. The **DAP Platform connects businesses with the blockchain technology.** It is not a trivial task to integrate applications and blockchain for businesses. Currently the **blockchain networks** have issues like slow speed and scalability that mostly defines the traditional business operations.

- This white paper describes a **utility network token for DAP**, implementation of Ethereum selected functionality on desktop and mobile (iOS and Android) platforms.

- The DAP Platform aims to enable DApps distribution through its marketplace, while providing a well established mobile **development and hosting platform for Ethereum dApp developers**. At the heart of it, Ethereum public blockchain will be used for every single moment. We are leveraging blockchain technology to build real world solutions that demands the use of a token.

- DAP doesn't depend on the price of **Bitcoin or Ethereum cryptocurrencies**. The value of DAP token is solely dependent on the number of nodes in our network and the demand we are creating on our platform — functional applications and use cases for users and businesses.

- **The dApp Builder token (DAP)** is a utility token that fuels the DAP platform. This includes dApp Builder and a Marketplace of dApps built on top of the DAP platform. The marketplace will allow anyone to create and distribute their own Ethereum dApps.

DAPP BUILDER

Industries already making Headways with Ethereum Blockchain Apps:

Many tasks that could otherwise call for the use of multiple intermediaries have been automated using Ethereum smart contracts across various industries, as highlighted in the following use cases:

1. Financial Sector and Prediction Markets:

- sluggish systems continue to bedevil the banking industry, with basic transactions taking hours or days to be confirmed, especially in global interbank transfers that have often relied on the SWIFT system, among others. Ethereum-based blockchain solutions like *Branche, ICONOMI,* and *Augur,* allow for faster processes with higher returns for investors, as well as prediction tools for events in the real world, translating to profits if predictions turn out correct.

2. Real Estate:

- the idea here is to use smart contracts to neutralize the conflicts involving payments, mortgage contracts, liens, and privacy concerns that strain relationships between lenders and borrowers – *Rex, Chainy* and *Trust Stamp* are good examples here. This way, users will also do away with hidden fees which characterize other platforms in the course of listing and searching for properties.

3. Music and Entertainment:

- with Ethereum based technologies, the media and entertainment industry is overcoming copyright challenges, payment and sharing of revenue difficulties, as well as tracking of online sales and downloads. Platforms to consider here include *Ujor, Peertracks*

4. Background

4.1 dApp Builder (DAP)

DAP aims to bring **Ethereum DApps** distribution through its marketplace on the mobile platform, while providing well-established development and hosting for **DApp developers**.

DAP will be using the DAP existing mobile app building and hosting platform to reach its current **45M** users on mobile and **2.5M** businesses and developers that monetize their apps on DAP

As DAP expands it will reach millions of businesses worldwide iBuildApp is a separate entity than **dApp Builder** and aims to connect **50,000,000 DAPs** users on mobile with decentralized applications that run on the Ethereum Network and powered by DAP
dApp Builder company was formed in Singapore

Team Behind dApp Builder?

dApp Builder is founded by **iBuildApp Mobile App Builder** and its executives and investors

Since launching in **2011**, BuildApp Mobile Builder received over **50M** downloads of mobile apps created by over **2,500,000** businesses and developers (app makers). There have been over **3,000,000** apps built and over **150,000** apps published on app stores and monthly visitors on mobile exceed **2.5** million.

iBuildApp has been named one of the **5 Best Mobile App-Building Programs** by Complex.com and featured on CNet, VentureBeat, Wired, ZDnet, (was mentioned in Techcrunch), TMCNet, PCMagazine, Huffington Post UK, Ubergizmo, Adweek, Folio Magazine, Enterprise Apps Today, Content Review.

4.2 dApp Builder **ecosystem**

1. dApp builder

- Predefined customizable smart contracts with Solidity source code
- A web interface to build mobile apps accessing these smart contracts.
- Javascript WEBAPIs and UI/UX framework, adopted for Android and iOS

2. dApp marketplace

- Listing of customized dApps
- dApps customers ratings and reviews
- Instruments to advertise and promote dApps
- Integration of dApps with 3rd party services – payment gateways, oracles, etc.

3. Desktop and Mobile access to dApps

- Adaptive to Desktop, iPhone and Android
- Work with dApps through WEB APIs talking to dApp Builder node
- dApp Builder node talking to Ethereum blockchain

3. The **Problem**

3.1. Marketplace problem

Launching a decentralized application on mobile can be difficult, various barriers to entry can prevent many great concepts from coming to life. The DAP Platform exists to make it so that any business can create their own dApps.

The DAP Market will enable anyone to distribute and monetize their decentralized applications on mobile and desktop.

The dApps on the DAP Market exist as decentralized autonomous entities. The participants will have no intermediaries — users, developers, and business are all one and the same. By providing the essential functionality needed to create and **monetize Ethereum dApps**, the DAP Market essentially removes all barriers to entry.

All developers will have the following baseline functionalities:

- ✓ Posting and listings
- ✓ Search and filtering
- ✓ Ranking and reputation
- ✓ Payments

How to Create a dApp in 3 Steps:

Step 1. Select a smart contract

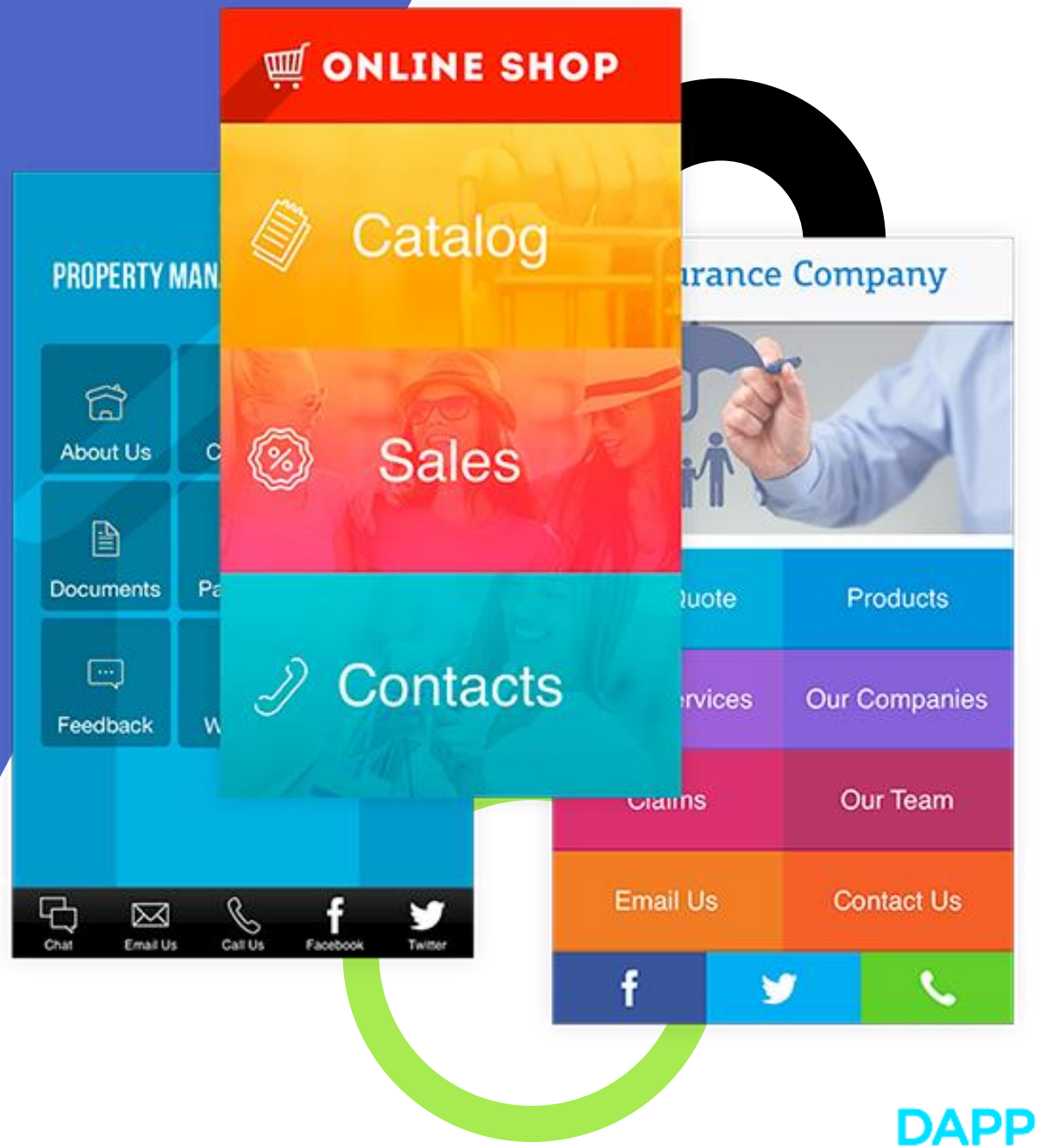
100+ smart contracts

Step 2. Edit your smart contract (use code or no coding required)

Easy to edit: input your data, select your logic

Step 3. Deploy on dApp Builder hosting, get mobile app and web browser client

Publish on iTunes or Google Play app stores and web browser



DAPP BUILDER

Use Cases:

Individuals

- ✓ Freelancing (online services)
- ✓ Lending an apartment (with a smart lock)
- ✓ Escrow for deals with real estate
- ✓ Car leasing
- ✓ Cryptocurrency as collateral

Large enterprises

- ✓ Delivery of containers
- ✓ Internal deals
- ✓ Long-team contracts with delayed or step-by-step payment
- ✓ Chained contracts (derivatives)
- ✓ Multilateral contracts

Blockchain Community

- ✓ TGE (ICO) contracts
- ✓ Creating custom ERC20 tokens
- ✓ Affiliate system
- ✓ E-commerce
- ✓ Online services

Small business

- ✓ Internal trade
- ✓ International trade
- ✓ Automatic royalty payments
- ✓ Ordering services from agencies
- ✓ Offline services

3.2 Ethereum dApp Development on Smart Contracts Problems

01 Programming experience with Solidity required

Smart Contracts must be developed in Solidity, an experience that very few people have.

02 It's difficult and Complicated

It requires special knowledge and its easy to mix up required functions and properties in **Smart Contracts** that will make it hard to understand.

Security and **dApps** exploits is also an issue

03 it's not easy to update Smart Contracts

To customize and maintain **smart contracts** you need a programmer

Marketplace problem

Launching a decentralized application can be difficult, various barriers to entry can prevent many great concepts from coming to life.

dApp Builder exists to make it so that anyone can start an **Ethereum dApp**.

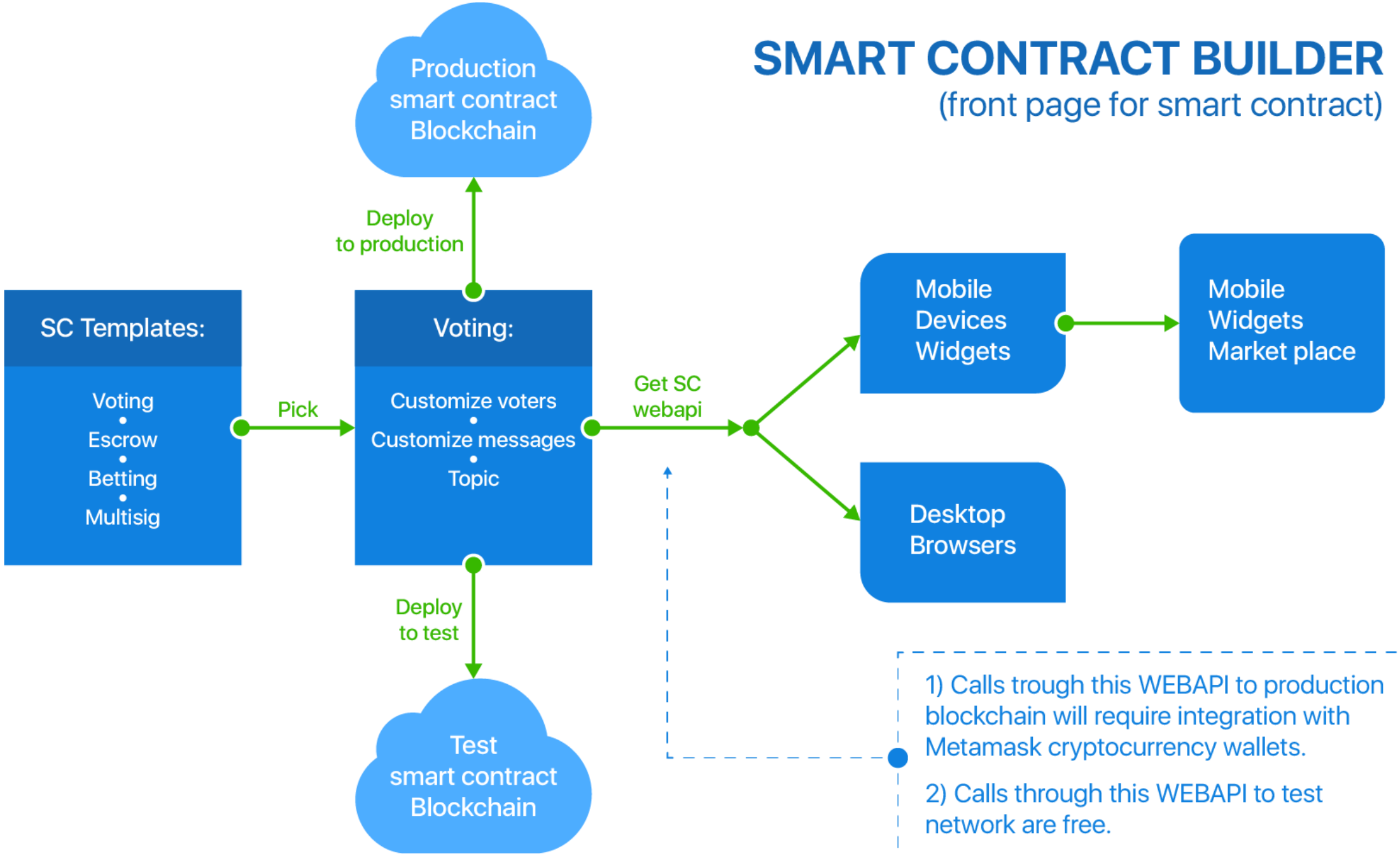
dApp Marketplace will enable anyone to create, customize, and deploy their decentralized applications.

4. Solution

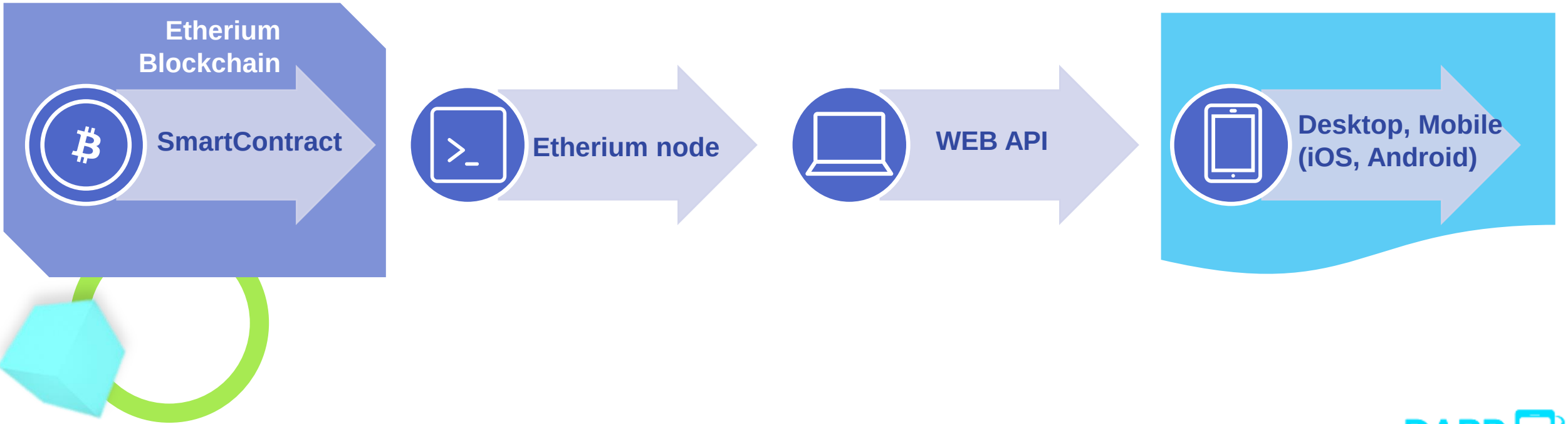
dApp Builder aims at instantly bringing the blockchain-based smart contracts to ***mobile apps and web browser***

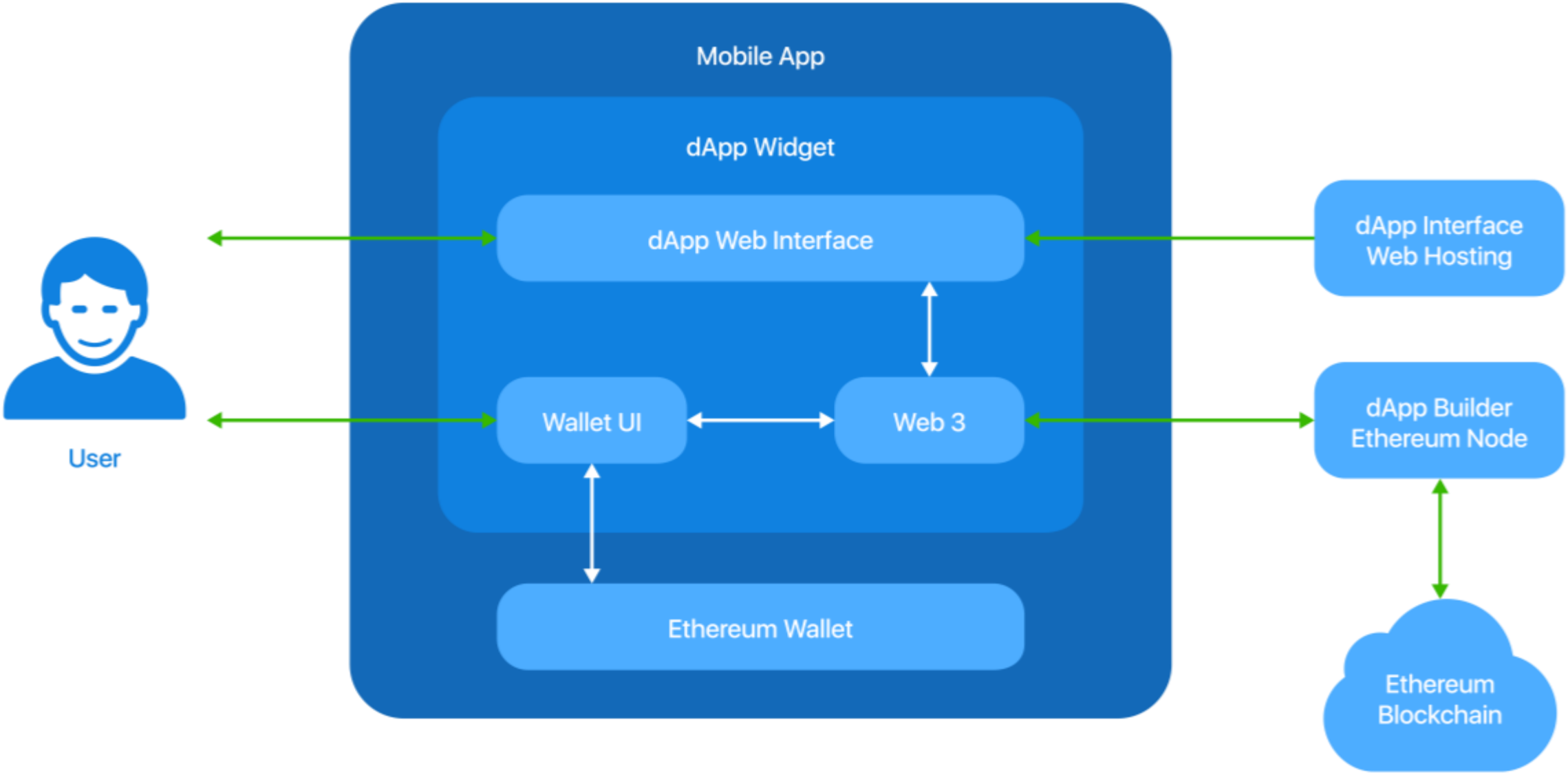
SMART CONTRACT BUILDER

(front page for smart contract)



Examples of how **dApps** will work on desktop and mobile:





Smart Contract Builder for dApps on Desktop and Mobile

iBuildApp Smart Contract Builder is a web portal that allow users to easily create and edit smart contracts with no coding skills, similar to how Wordpress and Wix allow for the easy creation of websites

1

Select a prebuilt Smart Contract template (voting, Escrow, multisig wallet)

2

Easily customize without any programming

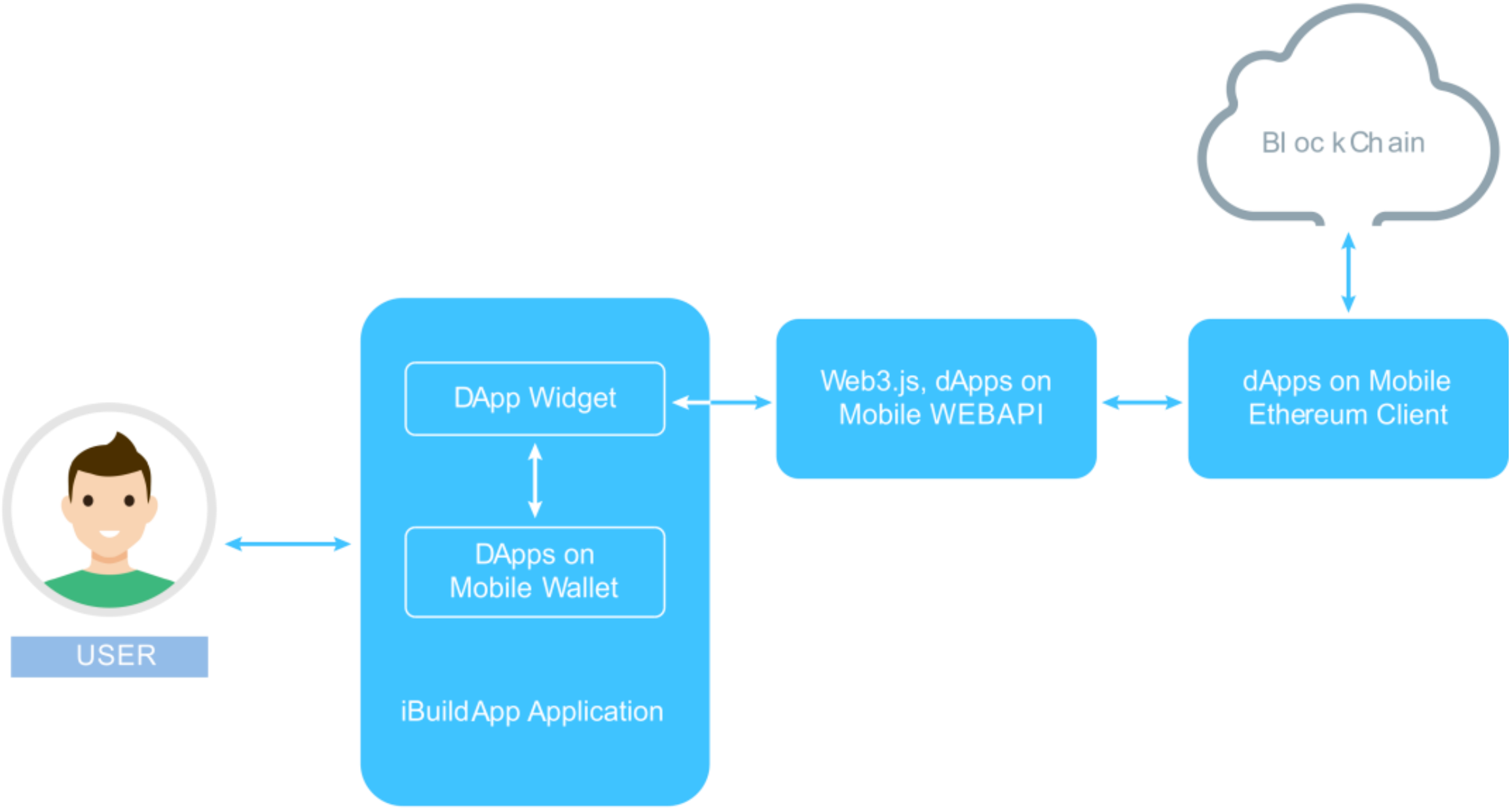
3

Deploy it to Ethereum blockchain and get it running on Ethereum instantly

TYPES OF
SMART CONTRACTS

- ✓ *Supply chain management Legally contracts.*
- ✓ *Voting and pool contracts.*
- ✓ *Notary services*
- ✓ *escrow contracts.*
- ✓ *Family trusts.*
- ✓ *Money management*
- ✓ *Corporate stock options*
- ✓ *virtual communities*
- ✓ *ICO*
- ✓ *project management*
- ✓ *M of N escrows and vaults*
- ✓ *Real estate transactions*

DAPP BUILDER



Solution

A **dApp Market will be implemented** allowing anyone to create and sell their own dApps, smart contracts and UI templates to developers and users all over the world.

The dApp Marketplace is a decentralized solution on Ethereum. iBuildApp Market will allow developers to create, publish and monetize their own **dApps** (JavaScript programming skills only necessary) while regular users can interact with these applications on desktop and mobile.

DAPP BUILDER

dApp Builder platform will connect dApps to desktop browsers and mobile apps:

dApp Widget -
the widget works as a web widget but also allows webpages that opens up in it to make calls to the WEB API. It has a front-end of the decentralized application, and besides the front-end has JS code with WEBAPI calls. Basically, this widget is a mobile light-client to connect to blockchain.

dApp Mobile Wallet - is an Ethereum-based wallet stored on the user's device; it is used for storage of DAP tokens which are required for the execution of transactions within the dApps Builder platform.

Network WEBAPI -
API, that receives calls from the front-end of the decentralized application that runs in the dApps Widget. The front-end and back-end of the application are connected by an Ethereum Client.

Network Ethereum Client -
is used to connect smart contracts in blockchain.

DAPP BUILDER

dApp Marketplace:

01

Smart Contracts

smart contracts of the decentralized applications, describe logic

02

License Contracts

define the terms of placement and royalty to users for using dApps defined in these License Contracts.

03

Smart Contract Marketplace

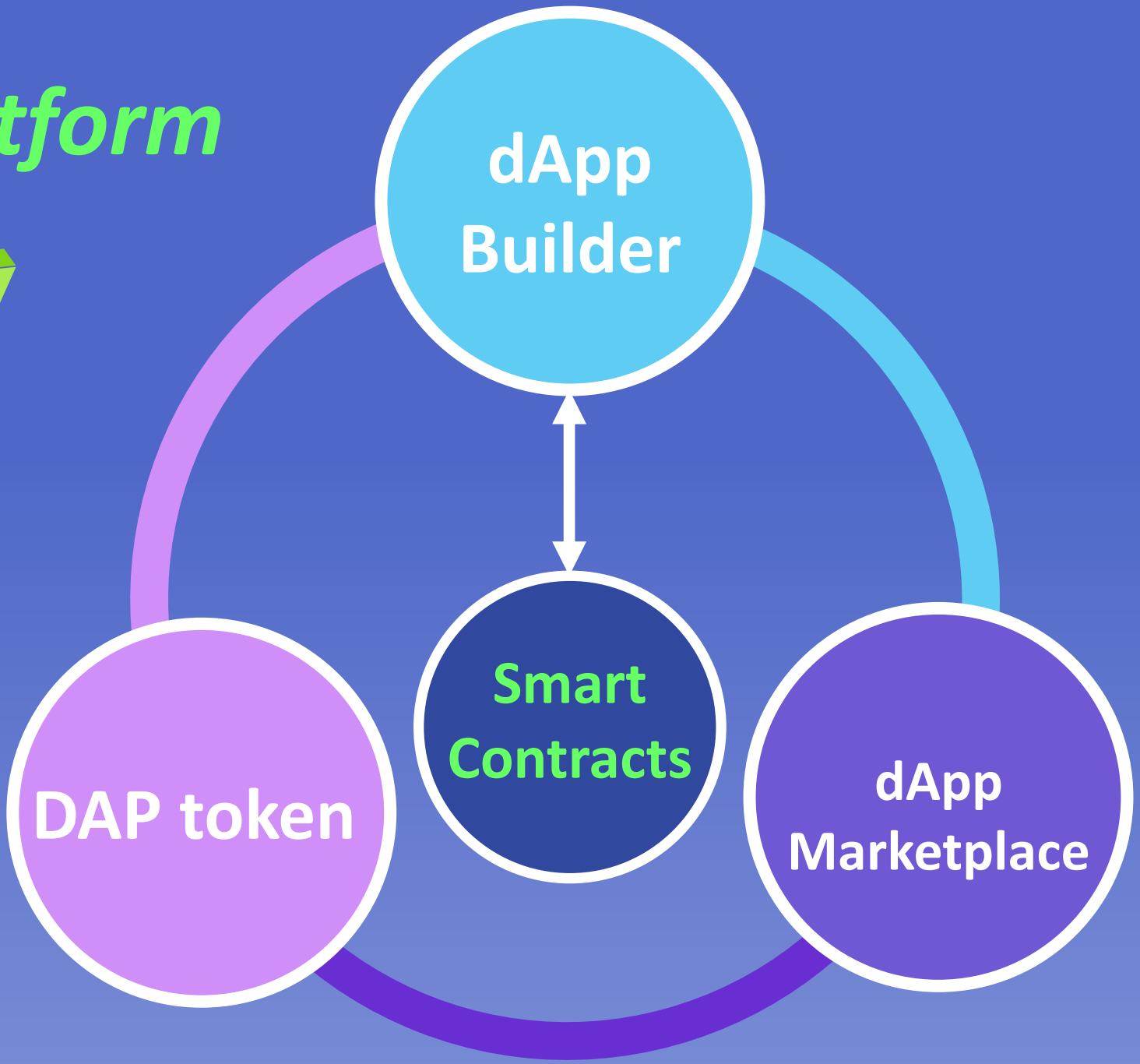
define terms for the dApp widgets to be added/used in mobile apps and to be used in other 3rd party apps (license)

04

DAP Token

used to conduct transactions

DAP Platform



DAPP BUILDER

5. dApp Builder Technical Description

The purpose of **dApp Builder** is to bring the smart contract technology to the business users who do not necessarily have the technical know-how needed to create a smart contract. You can think of **dApp Builder** as an online Microsoft FrontPage for smart contracts.

1. User Identity

For the purposes of **dApp Builder** we will need to have a strong user identity. This is especially necessary in a corporate/enterprise environment.

We are **not going** to develop our own login/password system but instead we will integrate with such strong identity providers like:

- ***Google.com (OAUTH)***
- ***Civic.com (JWT Token)***

The user will login into our system using one of these identity providers.

2. Once the user has logged into our system, she can do one of the two things:

2.1 Customize a mobile UI/UX to one of the standard dApps developed by **dApps Builder**. The description of standard dApps is available here. In this case we provide to the users something like Microsoft Frontpage for smart contracts.

2.2 Provide us with an address of user smart contract already published in Ethereum blockchain. In this case we will:

- Inspect the address of this smart contract pulling out Contract Metadata and Application binary Interface.
- Automatically create WEBAPI methods under https://dapps.ibuildapp.com accessing the methods of user smart contract

Let user create and publish mobile widgets and templates a-la https://www.ibuildapp.com/ accessing the user's smart contract through our WEBAPI (underneath this WEBAPI will be using Ethereum Web3 package.). These widgets and templates will be published on IBuildapp Marketplace in a special category of smart contracts with sub-categories

6. The process of **dApp creation**

I. User Identity

Please see description of user identity above

II. Choosing the type of smart contract

The user chooses one of the available "**dApp Builder"** smart contracts.
The following smart contracts are available:

1. Voting

1.1 in dApp Builder the owner customizes:

- the list of candidates to vote for, giving them:
  - *a name/description*
  - *a picture (optionally)*
- whether or not the voting is "blind"

1.2 The mobile widget shows:

1. a list of configured candidates with pictures and click on a candidate image/name to record his/her vote;
2. The vote of the mobile user gets recorded on Ethereum blockchain through a WEBAPI call to a method of "**dApp Builder**" voting smart contract;
3. Also shown are:
   - *If the voting is "blind" - how many votes are given for which candidate*
   - *if the voting is not "blind" - how many votes are given for which candidate and by whom*

2. Betting/Escrow

2.1 in dApp Builder the owner customizes:

- a list of options for betting, giving them:
  - a name/description
  - a picture (optionally)
- the maximum and minimum amount of bid;
- the time after which the bids are no longer accepted
- the Ethereum address of the "oracle" (could be his/her own address)
- A small amount of Ether that "bet oracle" gets for his fair choosing of the winning bet

2.2 The mobile widget shows:

2.2.1 For those who are not "bet oracle":

- an option to let mobile user to bet a certain amount of ether on each bid option (provided that the time has not expired);
- the bet of the mobile user gets recorded on the blockchain through a WEBAPI call to a method of "dApp Builder" Betting smart contract. The amount of bet is transferred to "dApp Builder" Betting smart contract
  - An option to see the bid and (potentially) re-call unless it is too late
  - An option to see who won the bet (one it is decided)

2.2.2 For the "bet oracle":

- before the end of bidding the widget just shows a message saying "The bidding is not yet over till [date]"
- after the end of bidding the widget shows the list of bids and let the "bet oracle" choose the winning bet.

- The choice of winning bet is recorded by a WEBAPI call to "dApp Builder" smart contract. At this point the money accumulated in the smart contract get re-distributed to the users who made the right bet (minus the small pre-configured amount that goes to the Ethereum address of the "bet oracle" for his fair choosing of the winning bet/bid)

3. Multisig Wallet

3.1 the owner

- ✓ configures the amount of ether to send;
- ✓ configures the address to send the ether to;
- ✓ funds the smart contract with the necessary amount of ether;
- ✓ Configures 1 or more Ether addresses that need to approve the sending of Ether.

3.2 whoever installed the widget:

✓ if the mobile device has the Ether address of the destination then we just show if the ether was already sent or we are still waiting for the signatures (we can list who are awaiting the signatures from). Or we show that all the signatures were collected at such and such time and the ether was transferred;
✓ if the mobile device has the Ether address of the one of the approvers of the transaction then we show them the button or approve (unless they have already done so);
✓ if the mobile device has none of these addresses then we show an ad for **dApps IBuilder** (or whatever the name of the new company).

III. Customization and deploy of smart contracts

By using a smart contract specific web form, the user can pick customizable values in the contract, for example, she may choose the list of candidates for Voting. When the user presses "Submit" button, we initialize a user record in a pre-deployed **dApp Builder** smart contract in Ethereum **blockchain**. The user pays for Gas required by this transaction, the gas can be paid through integration with **MetaMask**.

It is the responsibility of the user at all times to retain control of the Ethereum identity used for this initializing transaction. Only this identity will be allowed to make changes in the smart contract user record.

DAPP BUILDER

7. DAP utility token

7.1 The Purpose of Having a Token

- **DAP** will create utility tokens that provide access to **DAP platform-related services** in a decentralized, token-based ecosystem. These tokens represent a unit of account for the network.
- The bigger the network grows, the **more usage of the tokens** — and because the number of tokens are limited (no inflation in the final issuance, although they will be released over time). As the size of the network and transaction volumes within it grows, this will create demand for the tokens.
- A proprietary, uniform token like the **dApp Builder token (DAP)** can minimize traditional transaction costs as well as increase speed to set up settlement for all participants in the ecosystem.
- It also allows the DAP platform to help users to **monetize their content** (pictures, videos, blog posts, themes and widgets) and effectively creates what is really a public **utility for transactions among the network participants.**
- The value of DAP is completely dependent on the nodes within our participant network and the need that we are driving in developing connections with functional applications and connections with users, businesses and developers. We are just leveraging the blockchain technology to create real world solutions that demand the use of DAP token.

7.2 Token Economics

DAP TOKEN

- DAP, and its token distribution, will be created around **smart contracts built on Ethereum**. The number of DAP token issued will depend on the amount of contribution received by DAP smart contract.
- **Tokens issued by dApp Builder (DAP)** will be used for the access, creation, and deployment of dApps on smart contracts.
- **DAP Tokens** are the key to creating your own dApp, these tokens are used to manage the fees to use the DAP Platform, contract library, smart contract testing, monitoring, and **management of the entire smart contract process**.
- The DAP token serves as a method of validating the user's interactions with DAP Platform and will allow users to buy, execute, or barter for other smart contracts in the platform. In the future, **DAP token will serve as access for smart contracts on multiple blockchains, integrating these disparate blockchains into one unified view within the DAP platform**.
- There are two types of tokens on the platform. **DAP Platform customers** will need to buy the DAP tokens in order to create their own dApps and issue their own tokens. These customer tokens are called **custom tokens**. The custom tokens cannot be placed on any exchange and will be used in the internal transactions of their dApps.
- **The DAP token will be issued as a smart token**, or in other words a programmable token. I t will be using the Bancor protocol that holds one or more other tokens, in its reserve. **And DAP token can manage this reserve in a way that it sets a price and sells itself in exchange for the reserve currency or vice versa.** It would buy its own units and payout in the reserve currency in exchange for that. So users **can buy the smart token from the smart token itself/ smart contract** and they can sell the smart token to the smart token itself through the smart contract. And whenever DAP token is purchased the price goes up and whenever it's sold the price goes down.

DAPP BUILDER

Custom Tokens

- **The DAP Platform will create custom (utility) tokens** for the businesses building dApp applications on top of DAP Platform to tokenize their interactions with the user base. By using tokens in dApps, the company's users can earn and spend tokens that have redeemable external value, increasing the appeal of the application to new and existing users.
- The custom tokens can easily be exchanged with **DAP tokens (super tokens)** based on the supply/demand ratio. When **dApp developers (businesses) create their dApps** they need to set out a monetary policy for their **custom (utility) token** in the smart contract.
- The custom tokens, **that belong to their corresponding dApps**, can be exchanged in a secondary market and their liquidity will be obtained through DAP token.
- When businesses sign up their users in dApps, they **will be provided with an embedded wallet**.

Constant Reserve Ratio and DAP Token Supply

- **The Constant Reserve Ratio is set by DAP Platform as the creator of DAP token**, and is used in the price calculation, along with the smart token's current supply and reserve balance.
- This calculation ensures that a constant ratio is kept between the reserve token balance and the DAP token's market cap, **which is its supply times its price**. When DAP tokens are purchased the payment for the purchase is added to the reserve balance, and based on the calculated price, new DAP tokens are issued to the buyer. Additionally, when DAP tokens are destroyed, they are removed from the supply, and based on the current price, **reserve tokens are transferred to the liquidator.**
- DAP tokens hold one or more other tokens in reserve, i.e. **decentralized baskets** (all tokens powered by DAP Platform can be of help in reserve)
- **Example:** If we have a Constant Reserve Ratio 2% ratio and the price goes by 100%, the DAP token supply goes up by 1.5% automatically (nobody should have the right to manually issue tokens). **The DAP token supply shrinks as well if the price goes down.**

DAPP BUILDER

Network Effect

- What **dApp Builder** will do with DAP, is that it is going to issue DAP tokens that will be used as a network token. The DAP token, essentially will be creating the network effect, and creating the demand for the token
- If the DAP token goes up in value, it affects all the tokens that hold it in its baskets.
- With the DAP token as a smart token, we essentially created a “basket” of currencies (custom tokens created for the dApps by businesses).
- A currency basket is a portfolio of selected currencies with different weightings. A currency basket is commonly used to minimize the risk of currency fluctuations.

DAPP BUILDER

7.3 Token Sale

The token PRE Sale will begin in May 3

We will provide the details on how you can participate before the token distribution beginning.
Contributors willing to support the development of the DAP can send ether to the address we will provide before the contribution event.

The DAP utility network token ('DAP') will be distributed to participants in 3 contribution events.

Hard cap:
30,000 ETH + 72 hours

Market CAP:
$80M USD

Total amount of DAP tokens:
2,000,000,000

ERC-20 Token :
Yes

Price:
$0.04

Public Pre-Sale:
50% discount

Timeline:

Pre-Sale –
May 3

Main Sale –
May 31

dApp Builder will be managing the contract execution.
DAP token will be distributed among the contributors within 15 days after the end of the Token Sale.

Team:

Alexander Patrakeev

System Admin

Ekaterina Petrova

Android Developer

Vadim Smirnov

iOS Developer

Vyacheslav Suskov

Community Manager

DAPP BUILDER

Rafael Soultanov

CEO, dApps on mobile
iBuildApp Founder
45M downloads

Carlos Rodezno

Customer Support Manager

Alexander Bykov

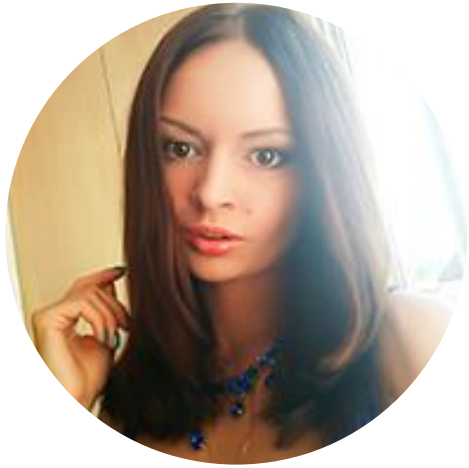
smart contract developer

Constantine Filin

CTO, 1 exit before at 140M valuation at Intermedia where Constantine was CTO

Yaroslav Timantsev

Community Manager



Anna Bodrova

Web Developer

Sergey Lobanov

Architect

Alexander Plekhanov

smart contract developer

Advisors:

Vlad Pigin

Microsoft Venture Fund

Eyal Hertzog, Israel

Bancor, Founder

Nick Mitushin

ICO Road Show

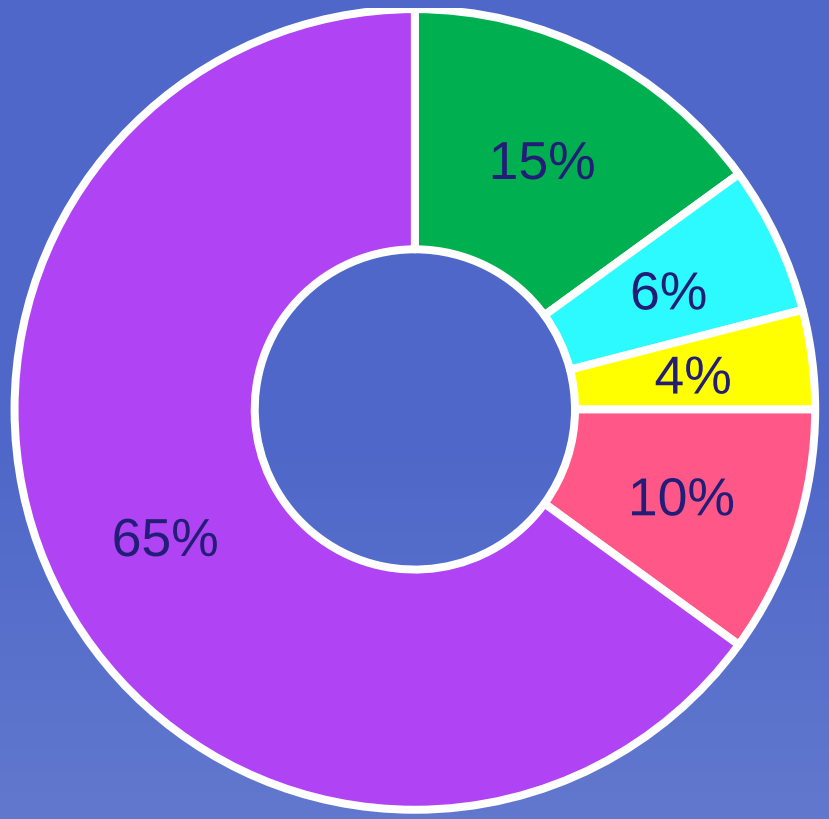
Gary Baiton

Senior Advisor at Crypto Law Group, Advisor at ICOBox , Angel Investor, Crypto Investor, ICO Expert, AI Evangelist

Andrey Verbitsky

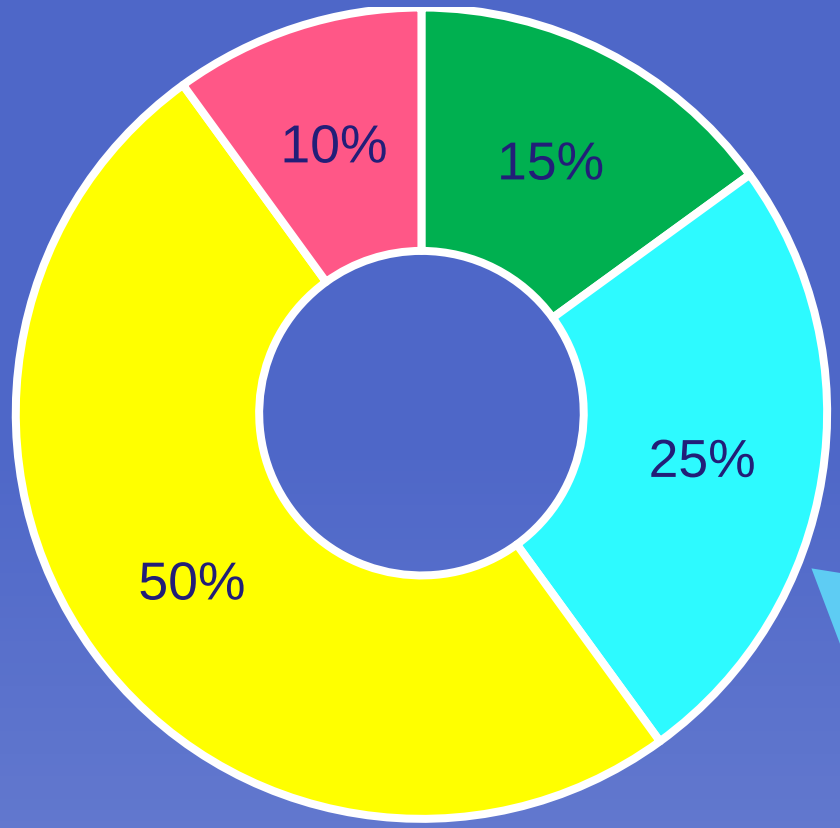
Token Economics Expert

Token Sale Structure (TBD):



- 15% team
- 6% advisers
- 4% marketing & legal
- 10% reserve
- 65% crowdsale

Funds Distribution (TBD):



- 10-20% R&D
- 20-30% community & partnership
- 40-60% marketing
- 10% sales

Some of **iBuildApp Customers** on Mobile:

DAPP BUILDER

Sample **Mobile Apps**
Built by using iBuildApp mobile app builder:

School Of Success

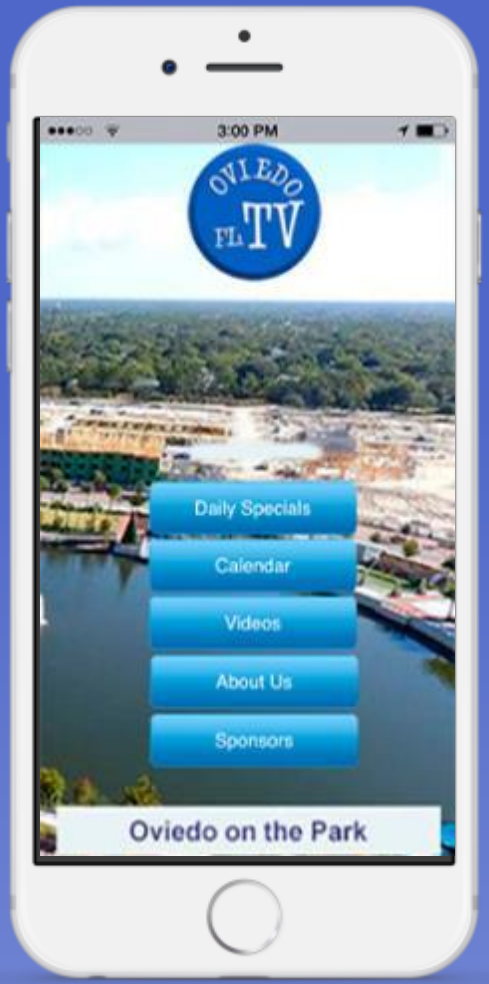
T-Town

HR Reference App

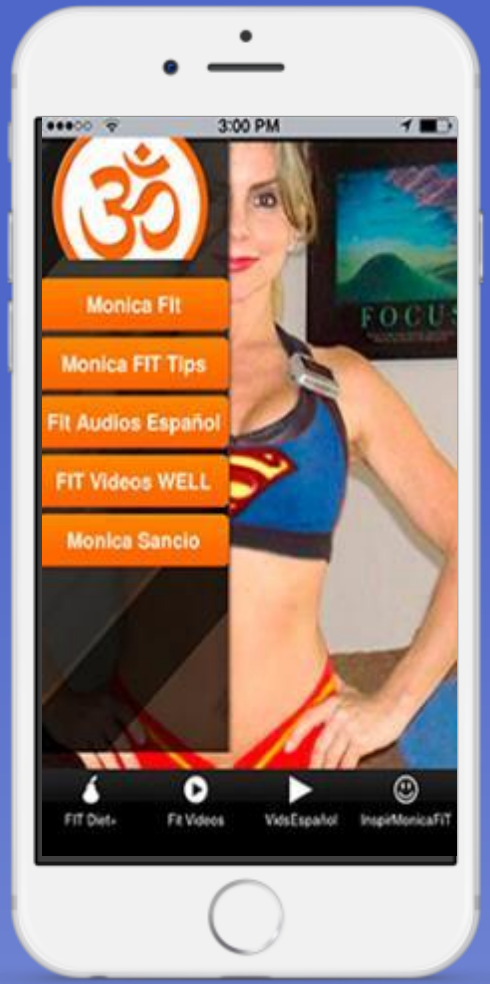
X61 Radio

DAPP BUILDER

Winner's Supermarkets

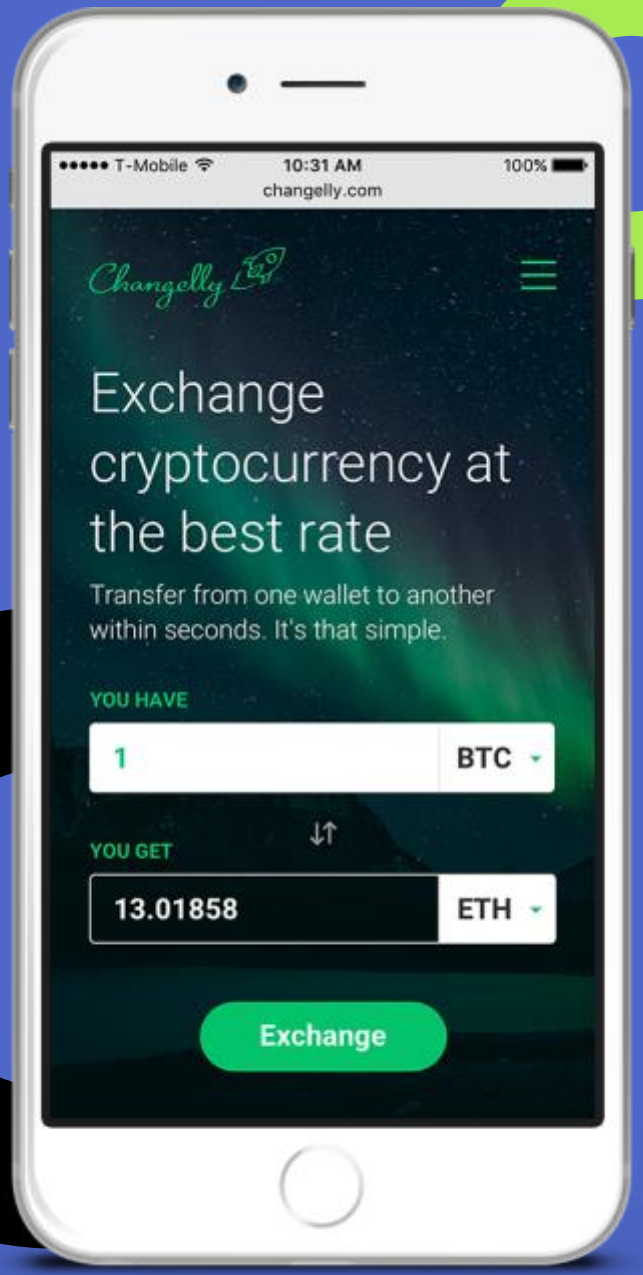


Oviedo TV



Monica FIT

Industrial Engineer



Changelly Exchange **integration**

DAPP BUILDER

Try Test dApp on Android Now!

ibuildapp.com/dapps.php