

# Learn how to create your own Dapp with Angular 9 - Part I.



Eli Elad Elrom

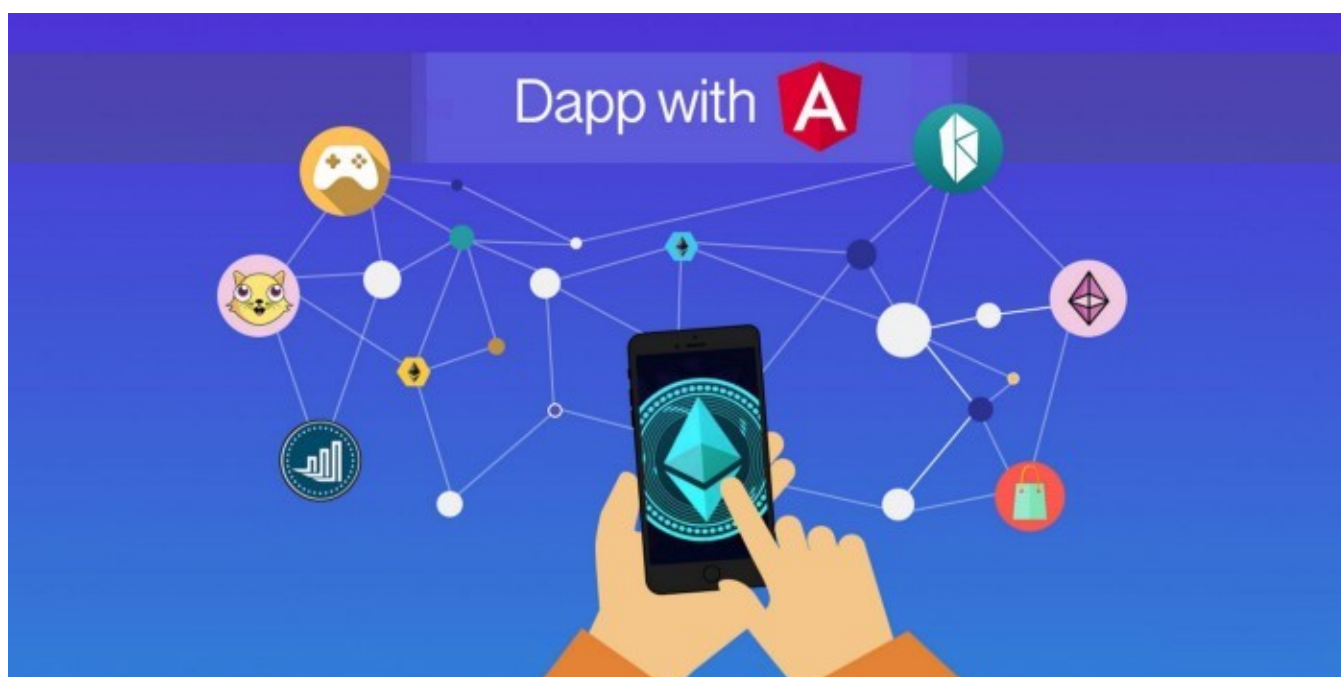
Follow

Jan 1, 2020 · 10 min read ★

In this six-part article, we will cover how to create a Dapp with Angular. We will cover how to create a decentralized application with the help of Angular so that users can interact with your smart contract using a friendly and intuitive user interface (UI). I broke down the process into a six-part article:

We will be covering the following topics:

- Part I — **Introduction** — Developing the dapp, including its benefits and classification. Using Angular, Angular architecture, benefits.
- Part II — **Creating an Angular Dapp** — prerequisites, and creating an Angular skeleton app
- Part III — **Styling Angular** — including custom components
- Part IV — **Creating a smart contract** with Truffle
- Part V — **Integrating** the smart contract with the dapp
- Part VI — **Linking** and connecting your dapp to the Ethereum network



You may have developed a smart contract, and sure developers and more savvy users can interact with a smart contract via the command-line interface and tools, but developing a front-end application that is able to interact with a blockchain is essential for all end-users. You do this by creating a decentralized application (dapp).

Let's get started.

## What Is a Dapp?

A decentralized application (shortened as DApp, dapp, Dapp, dApp, or DApp and pronounced as “dee-app”) is a web application that is able to interact with a smart contract. Dapps run on the blockchain and utilize the distributed ledger. The Ethereum blockchain is currently the most popular platform to run dapps; however, other distributed ledger technologies (DLTs) you have seen also provide the ability to create dapps such as NEO, EOS, and Hyperledger, ICON, Cardano, and Hashgraph (Hedera).

***“Everything that can be decentralized will be decentralized.”***

*— David Johnston, CEO of the DApp Fund*

If you have ever developed a standard desktop, web, or mobile application, you will

find that dapps are similar but also very different. A dapp is built using the same tools and languages you use to build any other app, but for an app to be categorized as a dapp, it needs to meet the following criteria:

— **Open source:** Its code is published as open source and should not be governed by one entity (centralized). Keep in mind that the application may adapt its own protocol in response to proposed improvements and market feedback; however, the consensus of its users drives all changes.

— **Decentralized:** Dapps utilize a blockchain or a P2P network.

— **Incentive:** Dapps use digital assets for funding.

— **Algorithm/protocol:** Dapps often generate tokens and include a consensus mechanism such as PoW, PoS, or even their own.

These criteria ensure dapps don't have downtimes like other apps you download from marketplaces such as iTunes or Google Play; dapps also give control to a community instead of one entity. These criteria can be significant. For instance, Apple and Google often reject apps for not meeting their arbitrary or monetary-based policies. These policies do not always make sense and are not always in the best interest of the end-user; they often are there to block usage of a competitor or for monetary gain. A good example is Google Bans Crypto App Metamask From Play Store.

Dapps that are based on open source code implemented on decentralized blockchains and funded by tokens generated using a specific consensus mechanism are believed by many to be the future of all businesses. Only time will tell.

Additionally, open-source software is an advantage because it allows users to view the source code and potentially contribute. Decentralizing using a blockchain harnesses the advantages of blockchain as DLT and

serves as a replacement to the traditional one-server database.

Finally, adding records/transactions to ledgers is usually done utilizing tokens and the consensus mechanism of a token is also an agreement between all the users of the dapp.

## Dapp Classification

In addition to the previous criteria, dapps can be categorized. The classification is based on the infrastructure the dapp is utilizing and can be broken down to these three categories:

— **Dedicated blockchain dapps:** These are dapps that use a dedicated blockchain directly; examples are bitcoin, Ethereum, EOS, and NEO.

— **Dapps relying on another blockchain:** For instance, the Omni Layer Protocol (formerly called Mastercoin) is a digital currency and communications protocol that is built on top of the bitcoin blockchain.

— **Dapps relying on another protocol that built on top of another blockchain:** These dapps use a protocol that is built on top of another blockchain. An example is the safe network using the Omni Layer Protocol.

A good example to help understand the classification concept is the USDT (Tether) token. This token was issued twice based on two blockchains: bitcoin and Ethereum. In this case, there are two types of USDT. The original, which is based on bitcoin, is done by using the Omni Layer Protocol to generate the token, and the Ethereum-based USDT is compatible with the Ethereum's ERC20 standard. Take a look at Figure 2.

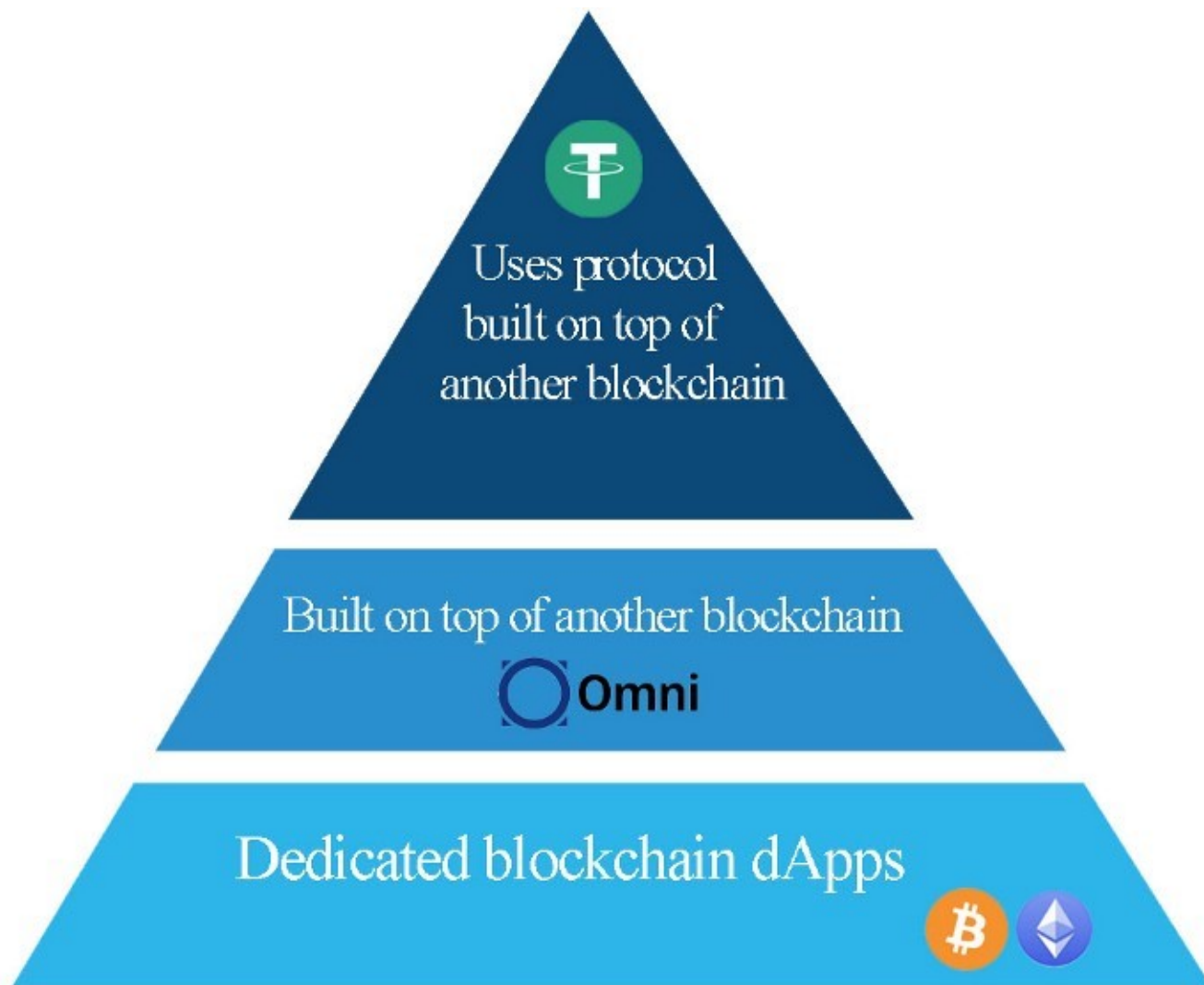


Figure 2. USDT (Tether) token

## Dapp Projects

Most dapps are built directly on top of the Ethereum blockchain or use a blockchain for their tokens. However, there are some dapps that even build their own dedicated blockchain. Take a look at Figure 2, for a sample of different dapps and their classifications.

Dapp	Description	Classification	Token	Blockchain
Ethlance	Marketplace for job postings and hiring freelancers. 0 percent fees.	Uses Ethereum directly	No token	Ethereum blockchain
Golem	Global market for idle computer power.	Token based on Ethereum	GNT	Ethereum blockchain
The SAFE Network	Data storage and communications network.	Implementation relying on another protocol (Omni protocol) that is built on top of another blockchain (bitcoin)	SFE	Bitcoin

Table 3. Example of Dapps and Classifications

In addition to the information in Figure 3, there are many resources to find more dapps; these two web sites provide a list of dapps that you can check: <https://dapps.ethercasts.com/> and <https://coinsutra.com>.

## How Do You Create Your Own Dapp?

The success of bitcoin and blockchain has brought an explosion of dapps. Developers and business owners have created a basic process to follow for developing dapps. You don't need to follow this exactly, and it may change by the time of writing; however, many of the published dapps out there have followed this process. The process consists of these five steps:

1. Write a white paper.
2. Launch an initial coin offering (ICO).
3. Develop the dapp.
4. Launch your dapp.
5. Market your dapp.

Let's review these steps.

## Write a White Paper

The white paper is similar to a company's business plan aimed at investors. However, it targets more than just investors; it's the technical blueprint.

The white paper is the technical document as well as the business plan and should explain the problem being solved and the concept, features, and technical aspects of the dapp.

Just like in a business plan, it's a good idea to include your unique selling proposition (USP), road map, members' résumés, capabilities, and history to help establish credibility.

---

**Note** *The unique selling proposition (USP) is the problem your dapp is aiming to solve.*

---

Once the white paper is published, it is good to get feedback from peers and the community in the early stages and prior to development. Social media, forms, and publications are often used to promote dapps and help create credibility.

## Launch an Initial Coin Offering

Once the white paper is published, the next step is to launch an ICO and sell coins or tokens to fund and support your dapp. The coin should have a reason for existence, rather than be the same as another coin/token out there, so you should explain how and why your dapp needs its own token or coin.

You also need to decide on the type of classification for your dapp, which will determine whether you will need any or all of the following:

1. issue token
2. set usage fees.
3. have a dedicated blockchain.

4. have a mining mechanism
5. set the allocation of fees
6. rewards investors
7. allocate fees to pay for different departments of your business: support, development, marketing, and business.

## Develop the Dapp

Development should be open source, and GitHub is usually used for repos for the development effort. On every release, it's a good idea to let investors and others know of a release to build users and a developer community around your project. Many dapps have tried to get funds and delivered no usable products; set yourself apart and avoid potential problems with regulators.

## Launch Your Dapp

Launch your dapp and include your release notes, documentation, road map, and maintenance plan. It's crucial to meet the promised launch date.

## Market Your Dapp

The last step is marketing. In addition to traditional marketing, dapps often hire or work with promoters during early phases or after release to get the word out. Another unique marketing aspect for a dapp is to get the coin/token listed on exchanges. This is the final stamp of recognition. Some exchanges have a voting system put in place to select the next coin/token to be listed. Some exchanges have been abusing this process and charging hefty fees to list a token or coin. For instance, a utility token listing on Binance exchange can cost from \$0.5 million to \$3 million.

Many early investors including dapp owners have been able to “cash out” if a token is listed on major exchanges as its price often goes up high because of the listing; however, it has become more and more difficult for a dapp to be listed, and it needs to provide real value. Fraudsters are often exposed, and coins/tokens get de-listed as quickly as they are listed.



## Why Angular?

With dapps, just like with any traditional app, you can write your application natively to the device you publish your app to (in the supported language of your device such as Xcode for iOS); however, it has been proven that using a framework can speed up development. For instance, if you want to utilize the same code and deploy your application on multiple devices with different screen sizes, that can become a challenge for a small team. Angular helps you build cross-platform modern applications for web, mobile, and desktop at the same time. The Angular CLI and Component Dev Kit (CDK) can help accelerate the development of apps.

Using Angular can be beneficial because of the following factors:

- Large community support
- Enterprise architecture and scaling
- Cross-platform support
- Documentation

Angular is a structural framework and enables you to create front-end client-side applications. The pieces are loosely coupled and structured in a modular fashion, resulting in less code to write, added flexibility, easier-to-read code, and quicker development time.

Angular allows the developer to put together a toolset for building a framework that will fit your exact application's needs. You can use HTML as your template language and extend HTML's syntax so the application's components can be read easily. Other than HTML, the coding is done with TypeScript, which turns JavaScript into an object-oriented programming language and gives you an enterprise-level environment.

Additionally, Angular is well structured and built to be fully accessible, in accordance with accessible rich Internet applications (ARIAs), so your app or site can be built correctly for people with disabilities.

Angular also gets along well with other JavaScript libraries so you can

install libraries such as the Ethereum JavaScript API web3.js with npm manager. Lastly, Angular's features can be easily modified or replaced to fit your exact needs.

**Note** *The word angular means having multiple angles or measured by an angle. Angular is a structural framework and enables you to create front-end client-side applications for the Web, mobile, and desktop. It is an open-source, a front-end framework for dynamic app development.*

Angular's most significant features are data binding and dependency injection. These can help decrease code. Also, Angular has been around for years; it's on its 9th release candidate.

**Note** *Dependency injection is a design pattern technique. As the name suggests, it means using one object as a dependency to another object by injecting the code.*

Angular 2 was a complete rewrite of AngularJS and offered a major change; however, there are no major differences between Angular 2 up to version 9RC. The latest release version of Angular at the time of writing is Angular CLI: 9.0.0-rc.7, and in this version, a few features were added such as the following:

— **Dependencies:** The dependencies were upgraded, and support for Typescript 3.1, RxJS 6.3, and Node 10 was added.

— **Bundle budget:** You can set a warning for the size of the application to ensure you don't exceed the limit (the default is 2 MB).

— **Angular CLI:** By running the CLI wizard, you can add components such as routing and decide on the format of the CSS.

— **Component Dev Kit (CDK)** of Angular Material: Add new features such as out-of-the-box virtual scrolling, drag, and drop, and “mat-form-field”

support for native select fields.

Angular 8 features are mostly to improve performance. It will include an improved view engine called Angular Ivy, improved upload of JavaScript for modern browsers that support ES2015+, support for web workers to use hardware for heavy lifting, support for TypeScript, a benchmarks tool, and more.


Blockchain <sup>20</sup> Dapps <sup>21</sup> Angular <sup>22</sup> Development <sup>23</sup> ICO *framework that can help expedite development. You can use other frameworks such as React (<https://reactjs.org>) and achieve similar benefits. This decision is really a matter of personal taste and your team's skill set. You could easily convert this project to a React project mostly by copying your project's files over to the React project.*

## Let's recap!

Get the Medium app

In this article, we took a deep dive into what a dapp is and looked at dapp

1. If you're interested in starting your own dapp project

: writing a white paper, launching an ICO, developing the dapp, launching it, and marketing your dapp.

We looked at why to use Angular. In the next article, we will create an Angular dapp, first ensuring the prerequisites were installed and then installing the setting Angular.

## Where to go from here

To learn more about what's possible with Blockchain as well as develop your own project check [The Blockchain Developer](#).