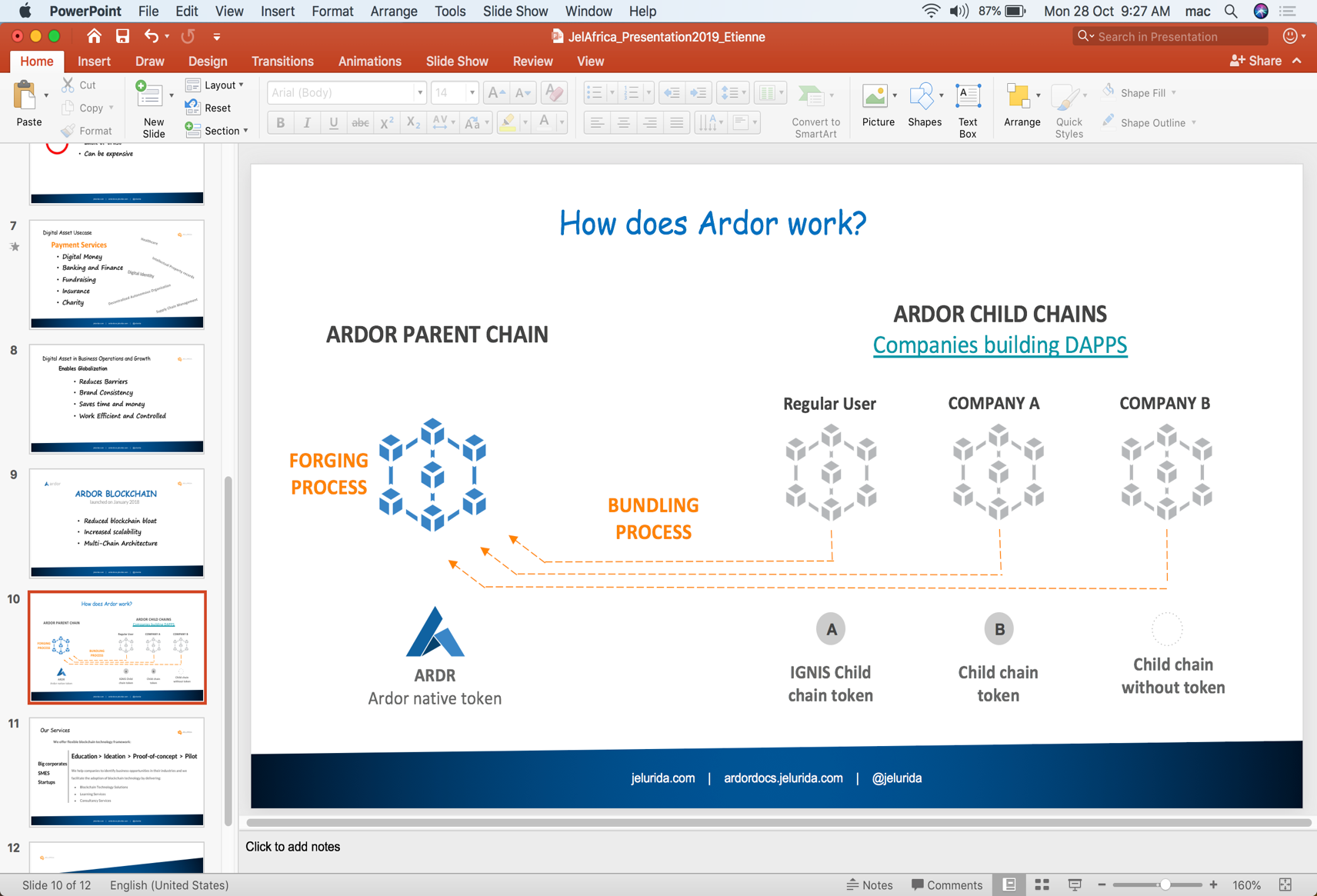
**module 3: ardor blockchain basice and its architecture**

1. **Ardor Blockchain Structure.**
2. ***Parent-Child Chain Architecture: Blockchain as a Service.***

Ardor platform is a *Blockchain as a Service* solution which can be used as Enterprise and Peer-to-Peer network. Ardor Blockchain is a multi-chain platform which evolved from the time-tested and energy efficient NXT blockchain.

NXT software is designed to be a universal framework for PoS blockchains. This is seen in its simple and flexible software architecture which makes it easy for adoption and can be extended to fit any public or private use case. NXT is the first implementation of the PoS protocol. This makes it energy efficient, and it can run on a minimal hardware requirement of 1GHz processor speed, 1GB RAM and 16GB ROM.

Ardor achieves both scalability and decentralization through its parent-child chain architecture. Also, Ardor is a PoS network and this makes it energy efficient, and it can run on a minimal hardware requirement of 1GHz processor speed, 1GB RAM and 16GB ROM.



**How dose Ardor Blockchain Work:**

Ardor multi-chain structure is comprised of the parent chain and the child chain. The parent chain has the ARDOR token as its native currency and this is the chain which secures the network. The transactions on this chain has an ID of 1. The ardor token is considered for its security. Other chains are known as the Child Chains.

The first child chain is the Ignis chain which have a native token called IGNIS. Its Transaction ID is 2. The Ignis chain is the chain where regular uses transact and build their Decentralized Applications on. It is seens as the token for Utility and general-purpose interaction on the network. There are other chains with their native tokens such as the Bitswift, AEUR and MPG tokens. Also, there can be chains without tokens and there can be private chains. The network child chains can be permission, not permission controlled.

**Benefits of Multi-chain Blockchain Structures**

* Scalability, Decentralized and Secure
* It can be confidential or permission
* Native Currency integration.
* It uses Archive nodes which reduces blockchain bloat.

**Further Reading:**

[Jelurida](https://www.jelurida.com/).

[Ardor Tutorials](https://ardordocs.jelurida.com/Basic_tutorials).

[NXT Documentation](https://www.jelurida.com/nxt).

[Ardor Whitepaper](https://www.jelurida.com/sites/default/files/JeluridaWhitepaper.pdf).

1. **Accounts, Transactions and fees.**
2. ***Economic Incentives:***

Blockchain Accounts can be described as identities on the network. A blockchain account or wallet is a digital wallet that allows one to manage his activities on the blockchain. Unlike the physical wallets, they are digital. Like the physical wallets, they can help you check your balance, sign transactions, read messages etc. there are several types of wallets. They include:

1. Hard wallet,
2. Desktop wallet/Desktop Client.
3. Online Wallets

Ardor wallet is integrated with the Ardor Client which makes the computer a node. It has both a Desktop Client and a Mobile Client. The most exciting feature of this is that you can forge both from your phone or laptop.

As one interacts with the Blockchain through a wallet/client, certain set of actions require fees to be paid to the network. This set of actions commit changes to the database. Some include, uploading of data to the blockchain, sending transaction, sending messages, setting Aliases etc. There some other actions which just read data from the database and this is carried out of fee. Some of these actions includes verifying transactions, verifying uploaded data, reading messages etc.

Writing these commits into the chain depends on the Bundlers. A Bundlers is like a type of forger who verifies your actions and transactions on a child chain and receives the payments for verifying and effecting changes on that child chain. If you run a private child chain you will have to be your own Bundler. The great benefit of this system is that it you can Bundle your transaction at zero cost.

After all the work done been done by the Bundlers, the bundled data block is forwarded to the Forger who submits the bundle block to the parent chain. The Ardor blockchain makes use of the NXT forging protocol.

**Further Reading:**

[Ardor Tutorials](https://ardordocs.jelurida.com/Basic_tutorials)

[Ardor fee Structure](https://www.jelurida.com/sites/default/files/ArdorFees.pdf)

[Forging and Account Leasing](https://ardordocs.jelurida.com/Forging_and_leasing).

[How to Create an Account on the Ardor Platform](https://ardordocs.jelurida.com/How_to_create_an_account_on_the_Ardor_Platform).

Video: [Ardor Bundlers](https://youtu.be/dpmEQxU4dR4)

Video: [Ardor Demo and Tour](https://youtu.be/6Dtn5oLuVTQ)

1. ***Brief overview of Ardor Features:***

Blockchain can be described as just a time stamp database on a computer which can communicate with other computers running the same clients. We can take the blockchain further by having inbuilt features. These features add more usability to the blockchain and increases the system’s scalability. Ardor Blockchain has lots of inbuilt features which includes Account Leasing, Monitoring and properties, Asset Control, Exchange, and Properties, Aliases, Bundling, Chain Control, Coin Shuffling and Exchange, Forging, Data Cloud Messaging, Transaction Voucher, Voting Systems etc. For more technical readings about the blockchain features, one can access it below.

**Further Reading:**

[Ardor Features.](https://ardordocs.jelurida.com/Features)

1. **How to setup an Ardor Node Setup**
   * 1. ***Download and Installation (for both Mainnet and Testnet)***

Before running an Ardor client, the basic requirement is to have at least JDK8 running on your system. It is most preferred to run JDK11. One of the most secured ways to use blockchain clients is to run a full network client. To download Ardor Blockchain, you should ensure you are on the original Jelurida site or you can clone the software from Jelurida’s Bitbucket. In case there is a specific version of the Ardor Client you are in search of, Jelurida’s Bitbucket is the right place to look at.

There is a new release of Ardor Client for Smart Phone. Your mobile phone needs a minimum requirement of being an Android 8 system. The mobile client can also be downloaded from the link provided below.

**Further Reading:**

[Getting Started with Ardor Node](https://ardordocs.jelurida.com/Getting_started).

[Download Ardor from Jelurida](https://www.jelurida.com/ardor/downloads).

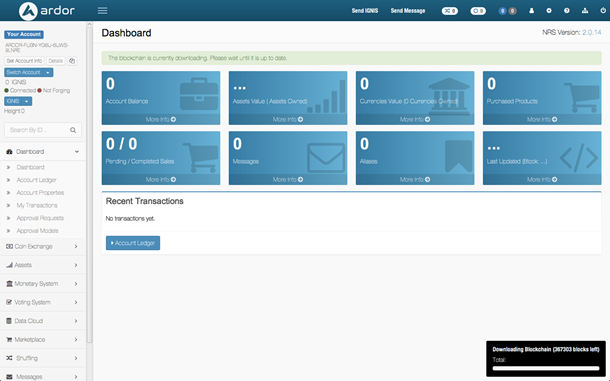
[Ardor Source Code on Bitbucket](https://bitbucket.org/Jelurida/ardor/src).

[How to download the Ardor Client](https://ardordocs.jelurida.com/How_to_download_the_ardor_client_wallet)

* + 1. ***Ardor Account and Account Funding.***

When you start the client, you can login into your already existing account or you can create a new one. If it is your first time of using the client, you will have to create a new account and make sure you save your passphrase. The passphrase is the only key of your account that can allow you to commit changes (like fund transfer) on the blockchain.

Firstly, if you start your client and it doesn’t have support for a User Interface, you will have to open the Ardor Client on your browser using at the address: [http://localhost:27876/index.html#](http://localhost:27876/index.html) for mainnet and [http://localhost:26876/index.html#](http://localhost:26876/index.html) for testnet environment. Click on “*Don’t have an account? Click here to create one*”. Ensure use save your *Passphrase* and your *Account Address*. Re-enter your Passphrase and login to the account.



Once the Client starts running, it starts downloading the database block by block, but for a faster download, you can download a zip of the blockchain database from the Jelurida website for [mainnet](https://www.jelurida.com/Ardor-nxt_db.zip) or for [testnet](https://www.jelurida.com/Ardor-nxt_test_db.zip).

Funding your account can be through transferring real coins to the mainnet account address or you can go to the [Ardor Faucet](https://www.ardor.world/en/faucet_ardor/) to get Ardor and/or Ignis test coin (i.e. if you configured client to run in the testnet).

**NOTE: -**

DO NOT USE YOUR **MAINNET** PASSPHRASE ON

YOUR **TESTNET** ACCOUNT.

**Further Reading:**

[How to setup Ardor account.](https://ardordocs.jelurida.com/How_to_create_an_account_on_the_Ardor_Platform)

[Download from Jelurida](https://www.jelurida.com/ardor/downloads).

1. **Introduction to the Turning Complete Lightweight Contract**

As described in Module 2, Nick Szabo coined the world Smart Contract in his paper “[Smart Contracts: Building Blocks for Digital Markets](https://pdfs.semanticscholar.org/9b6c/d3fe0bf5455dd44ea31422d015b003b5568f.pdf)” and he noted that,

*“A smart contract is a set of promises specified in digital form including*

*protocols with which the parties perform on these promises.”*

Smart Contracts are business logic or programmable agreement which are codified are executable by computers. For instance, you rented an apartment which is on the blockchain and you are to pay with cryptocurrency. The owner will have to give you a digital entry key which is time bound (and this depends on your payment plan). The smart contract handling this operation has some functions it can perform automatically. These include,

* Giving you back your fund if the owner didn’t send you the digital entry key.
* The owner cannot stop you from entry before the rent is due.
* The system only releases the key when you have paid the rent.
* Rent must be complete in order to have access.
* The owner is sure to paid.

The system works on the If-Then premise and is witnessed by the people on that blockchain network, so you can expect a faultless delivery.

You can use smart contracts for all sort of situations that range from financial derivatives to insurance premiums, breach contracts, property law, credit enforcement, financial services, legal processes and crowdfunding agreements.

**The Lightweight Smart Contract**

This is a Smart Contract framework developed by the [Jelurida](https://www.jelurida.com/) core team. The contract is written in plain Java programming language. This type of contract is for developing a layer of automation on top of the existing Ardor protocol. This type of smart contract is referred to as the stateless contract. Ethereum smart contract are stateful smart contracts.

Stateful contracts save contract modifications by itself but stateless contract submitted all state modifications required to be changed by the contract as normal transactions. Lightweight contracts are Turning Complete.

Lightweight contract class files are deployed as cloud data onto the blockchain network and then a contract reference transaction is submitted from the account running the contract. A Contract Manager utility has been developed to help developers on managing both deploying the cloud data and submitting the contract reference transaction. One of the major advantages of this is that if a contract has bud you will not have to upgrade the contract but just replace it with another.

**Triggers**

When these contracts are written and deployed on the blockchain, they can be called or triggered in four distinctive ways as mention below.

1. By Transaction of Messaging the contract account.
2. By Block height trigger without a trigger transaction.
3. By Voucher where a digitally signed voucher is used.
4. By using APIs to pass there required parameters.

**Further Reading:**

Jelurida: [Lightweight Contracts](https://ardordocs.jelurida.com/Lightweight_Contracts).

[Smart Contract: The Blockchain Technology that will replace Lawyers.](https://blockgeeks.com/guides/smart-contracts/)

* + 1. ***Development Environment Setup (IntelliJ, JDK8, Ardor Client)***

Setting up the environment includes downloading the minimum necessary tools, setting them up and also, installing the blockchain client as described the Ardor Account section above. Basically, you are required to have a PC which can be a simple as Raspberry Pi but for Contract development, you will need something high.

The two primary tools include:

1. Java (JDK 8 or 11) and
2. A text Editor or IDE

It is most recommended that you make use of IntelliJ IDE for Ardor Development. The community version of IntelliJ is suitable for Ardor Developments.

**Further Reading:**

[Lightweight Contracts](https://ardordocs.jelurida.com/Lightweight_Contracts).

Download IntelliJ from [JetBrains](https://www.jetbrains.com/idea/download/) Website

Download Java from [Oracle](https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html) Website.

* + 1. ***Contract Configurations***

The contract configuration can generally be categorized into two parts:

1. Configuring the IntelliJ and
2. Configuring the Contract Manager.

The configuring your IntelliJ is setting up the development environment which includes loading the Contract Manager Plugin and setting up the IntelliJ Project Structure settings and then rebuilding the application.

For the Contract Manager configuration, you will have to create a file named [*nxt.properties*](https://ardordocs.jelurida.com/Faq#Node_configuration_using_the_nxt.properties) in the ***conf*** file. where you will both configure the Contract Runner and the Contract manager.

**What is a Contract Runner?**

*“The contract runner is an add-on, a server-side extension, which every node*

*can choose if to run or not. When it’s running, it loads the contracts*

*stored in the blockchain and execute them based on various conditions.”*

- By Lior Yaffe, MD at Jelurida

**What is a Contract manager?**

This is a plugin that enable Ardor Developers to handle and automate running of the Ardor node on a Computer and also the deployment of the Lightweight Contracts by that node.

**Contract Runner Parameters**: This configures the operation of the contract runner and private contract parameters which cannot be stored on the Blockchain, for example account passphrases. This is defined in the contracts.json configuration file of the node running the addon. These parameters are shared by all contracts deployed on this specific contract runner.

**Contract Setup Parameters:** This configures the specific contract parameters for the particular account running the contract. These parameters are stored on the Blockchain as the value of the contract reference transaction.

**Contract Runtime Parameters:** These parameters are for contracts triggered by a trigger transaction or a voucher, the transaction itself provides parameters to the contract as an attached message, these parameters are specific to this invocation of the contract.

**Below is a general set up parameter for the nxt.properties**

**#ContractRunner**

nxt.addOns=nxt.addons.ContractRunner

addon.contractRunner.secretPhrase=<…passphrase…>

addon.contractRunner.feeRateNQTPerFXT.IGNIS=100000000

**#ContractManager**

contract.manager.serverAddress=localhost

contract.manager.userHttps=false

contract.manager.secretPhrase=<…passphrase…>

*Note:* The <…passphrase…> should be replaced with the passphrase of the contract account.

**Further Reading:**

[Lightweight Contract FAQ](https://medium.com/@lyaffe/lightweight-contracts-faq-30273120da9a).

[Lightweight Contract Design](https://medium.com/@lyaffe/lightweight-contracts-design-f37fa5211bae).

[Lightweight Contract: The Next Step](https://medium.com/coinmonks/lightweight-contracts-next-steps-9a88a60e5fec).

* + 1. ***Contract Development and Deployment***

Assuming all configuration settings have been made and the next step is to develop the codes for a particular smart contract. The contract can compose of one or more Java class files. More about contract development can be found at the [Lightweight Contract](https://ardordocs.jelurida.com/Lightweight_Contracts) section on the Jelurida website.

You need to start running the node and make sure that all the blocks have been synced in the network. The contract can then be deployed to the network and triggered as required by the creator.

**Benefits of Lightweight Contracts**

* Backward compatibility with new system upgrade.
* Contracts query the blockchain for data.
* Avoids costly and risky data migrations during contract upgrade and bud fixing.
* The contracts are stateless so they don’t store data by it is self.
* The same contracts can be running in parallel with different setting.
* Contracts can easily be integrated with Java library.

**Further Reading:**

You can take the [Ardor Bootcamp](https://www.udemy.com/course/ardor-bootcamp/) from Udemy and/or learn more about [Blockchain Programing: Smart Contract with Ardor](https://www.udemy.com/course/blockchain-contract-developer-course-for-beginners/).

1. **Forging and bundling with a Raspberry Pi.**
   * 1. ***Selecting the necessary Hardware.***

Forging and/or Bundling can be done direct on a PC but it is not sustainable since it is not advisable for one to run his PC for long period of time (days and weeks). It is necessary to get a fast and light performance CPU like a Raspberry Pi 3 model B. Selecting the necessary hardware for the Forging node depends on how the owner wants the performance improved. The minimum requirement for a fast sync node includes:

* Raspberry Pi3 model B
* 16GB MicroSD Card (Ultra SanDisk, Type 10)
* Raspberry LCD Screen (3.5” or 5”) or External Display or E-Ink Screen.
* USB cable (class B).
* External Keyboard and Mouse.
* Power source (Solar Panel or Power Bank of 30,000mAh and a charger for it).
  + 1. ***Setting up the Raspberry Pi and its accessories***

The Ardor blockchain client is downloaded and installed on the Raspbian OS almost as in the case of the Linux OS. You can also set up a screen view just to keep track of how much progress is been made.

Setting up the SD Card will require you to install Raspbian Stretch OS with [Etcher](https://www.balena.io/etcher/) and then you can enable SSH to be able to login to the Raspberry Pi. You will have to install both JDK and the Ardor node on the Raspberry Pi. If you have an external screen, you can actually use the Raspberry HDMI but if not, you will have to clone a Screen Library from GitHub form the link in the reference section below.

You will have to listen using your browser on your localhost at port [http://localhost:27876/index.html#](http://localhost:27876/index.html). Set up your account, fund the account with not less than 1000Ardor, and then start the node forging process to secure the Ardor Network.

* + 1. ***Wi-Fi Configuration Settings.***

This starts with you having to create a file called wpa\_supplicant.conf in the boot partition of your Raspberry Pi. Copy the code below in to the file and save.

country=XX  
ctrl\_interface=/var/run/wpa\_supplicant GROUP=netdev  
update\_config=1  
network={  
 ssid="YOUR\_SSID"  
 psk="YOUR\_PASSWORD"  
}

You will have to change the Country Code, Wi-Fi SSID and the Wi-Fi password. The [PuTTY](https://www.putty.org/) app can also be used for gaining remote access from your PC to the Raspberry Pi.

**Further Reading:**

[Headless Solar Powered Ardor Blockchain Node with e-ink information screen](https://medium.com/@mrv_89306/headless-solar-powered-ardor-blockchain-node-with-e-ink-information-screen-8456a7e98f0f).

LCD-Show: [Git Clone](https://github.com/goodtft/LCD-show.git).

[A Solar Powered Ardor Blockchain Node](https://medium.com/@mrv_89306/solar-powered-ardor-blockchain-node-aa680a976c50).

[How to set up LCD Display for the Raspberry Pi](https://www.youtube.com/watch?v=Fj3wq98pd20)

[How to connect a 3G modem to the Raspberry PI](https://www.youtube.com/watch?v=C92Wygv5p0w) (if your Wi-Fi is not stable or steadily you can use modem).

[Connect Raspberry Pi to a 3G Network](https://lawrencematthew.wordpress.com/2013/08/07/connect-raspberry-pi-to-a-3g-network-automatically-during-its-boot/).

1. **Decentralized Application Developments**
   * 1. ***Brief discussion on the Ardor API documentations.***

Developing decentralized application on Ardor can be very easy for Ardor has more than 250 [APIs](https://ardordocs.jelurida.com/API). This makes it stable for developers to quickly and cost-effectively deploy dApps and can be integrated into your business processes. You can read more from the link below

**Further Reading:**

[Ardor Features](https://ardordocs.jelurida.com/Features/en)

1. **Ardor Tools and Applications**
   * 1. ***Ardor Blockchain Explorers.***

Some Ardor Explorers are:

* [Ardor Tool](https://ardor.tools/).
* [Bitswift](https://bitswift.network/)
* [Ardor Platform](https://ardorplatform.org/)
* [Ardor FAQ](https://ardordocs.jelurida.com/Faq)
  + 1. ***Ardor Forging Pools.***

Some for the active forging pool include:

* [ArdorPool](https://www.ardorpool.org/)
* [Bitswift Forging Pool](https://bitswift.wiki/index.php/LZLZ_Forging_Pool)
  + 1. ***Ardor online Communities***
* [Slack Community](https://ardornxt.community/)
* [Ardor Forum](https://ardorforum.org/)
* [LinkedIn](https://www.linkedin.com/company/ardor-platform/)
* [Telegram](https://t.me/ardorplatform)
* [Reddit](https://www.reddit.com/r/Ardor/)
* Twitter: [@Jelurida](https://twitter.com/Jelurida), [@JeluridaAfrica](https://twitter.com/JeluridaAfrica), [@Ardorplatform](https://twitter.com/ArdorPlatform), [@IgnisGuide](https://twitter.com/IGNISguide)

1. **Advanced Topic**

* [Resource Library](https://www.jelurida.com/resources)
* [Advanced Guide](https://ardordocs.jelurida.com/Advanced_guides)

1. **Brief discussion on how to actively get involved on the Ardor Project**

This section brings us to the most frequent questions we ask must times, how to become involved in the Blockchain space. There are several steps to follow and I will list them here.

1. Define the type of blockchain service you want to offer in the space.
2. Pick a Blockchain project, (in this case, it is the Ardor Blockchain).
3. Learn about an industry and how to apply Blockchain to it.
4. Get acquainted to some tools.
5. Probably learn a programming language and build web projects.
6. Join the Blockchain project community (help is often faster from there)
7. Prepare for Blockchain job interviews and then apply.

There are some of these steps that can be done simultaneously. The blockchain space has different types of job opening. You can make some research to find out positions which better fits you.