install scala sbt rpm:packageBin create temporary folder: mkdir /tmp/techgen install crated in (2) rpm: sudo rpm -ivh /path/to/rpm run in background: /usr/bin/bd-app & see logs in /tmp/techgen

# Brief description of the TechGen blockchain

The TechGen blockchain project consists of two parts:

## Blockchain

1. The block explorer namely the TechGen blockchain itself, is a chain of cryptographically linked blocks. For the development of the blockchain were chosen time-tested solutions of the classic blockchains. However, widespread blockchains have a number of issues that, on case of absence of solutions shall lead to the degradation of the blockchain base in the future. At the moment, solutions have appeared, but their implementation for a working blockchain is not a trivial task and requires significant resources. Much more than creating a new blockchain with modern solutions. These solutions provided a number of advantages over traditional well-known blockchains. Let's list these benefits shortly:
2. There is no need to do pointless work during block production or “mining”.
3. The ability to flexibly configure and control the number of manufacturers depending on the network configuration. All these allow to reduce the latency of the network, thereby making it more resistant to attacks on consensus.
4. The ability to create a system of rights. The system of rights already allows you to design simple business logic at the stage of creating a network. The vast majority of blockchains do not have a system of rights and are limited to allowing the production of blocks by a node based on the characteristics of the node itself.
5. Account identification capability. This is a crucial point from a legal side. Recently, one of the important aspects of blockchain operation has become user deanonymization. The KYC mechanism is increasingly used, and more and more government regulators express a negative attitude towards anonymous actions on the blockchain. There are also enough cases in business when you need to identify the user. It might seem that the solution is obvious to allow only those users we know to use the blockchain. However, this approach is too rigid and some potential users may simply refuse to use such a blockchain [platform]. Therefore, deanonymization should be gradual.
6. TechGen blockchain offers a different scheme. Important activities such as block production will require user identification. Since, it is possible that the owner of the node will make a profit for the production of blocks. And users who just need to move funds between accounts can remain anonymous until they want more rights in the system. In the future, this aspect can be regulated.
7. The TechGen blockchain uses technical solutions that reduce the size of the code, therefore the speed of changes. For example, one of the most famous blockchains takes over 100,000 lines of code. And that's just the core. Naturally, maintaining and developing such a large amount of code requires huge resources. The technical solutions that are used in TechGen several times reduce the code base, and therefore the resource required for support, which is extremely important for business.
8. Possibilities for developing smart contracts. The functional paradigm is more appropriate and largely important more secure for writing smart contracts. This conclusion is based on experience using existing smart contract implementations. Unfortunately, now smart contracts and the languages ​​in which they are written are becoming more complicated. Which leads to the fact that programmers are busier with the problems of describing an action in technical terms, i.e. they think about how to write, rather than what to write. This approach takes technology out of business, which has a significant impact on the number of successful applications [produced] that actually work to solve todays’ problems. The functional paradigm is focused precisely on the description of the program’s logic itself, which is its clear advantage. At the moment, blockchains with smart contracts alongside with functional paradigm have just begun to appear. Since the TechGen blockchain uses a functional paradigm, the development of smart contracts of the described type will be logical and seamless, and the contracts themselves will become more compact, understandable and focused on writing the business logic itself, and not on how to describe this business logic technically. This approach was chosen to initially make a base for a great development potential in the TechGen blockchain.
9. Today, there are already developed a practical solutions based on the functional paradigm. Moreover, such blockchains have common libraries and already use a functional paradigm in writing smart contracts, which confirms the practical value of the chosen approach. The TechGen blockchain use libraries that have been tested by other blockchains in practice.
10. Optimization of block production on demand. We often see empty blocks to appear in PoA consensus blockchains. This is caused by a frequent generation of blocks, which does not depend on the filling of transactions. Thus, we fill the database with empty records. And we all know what sizes’ of bases the modern blockchains have. TechGen can generate blocks depending on blockchain occupancy. This will allow you to avoid empty blocks and adjust the optimal ratio of transaction execution speed to block filling.

The TechGen blockchain has the technical ability to work with multiple assets.

Let's consider the solutions that achieve these advantages in more detail:

## Consensus

The TechGen blockchain is powered by Proof Of Authority consensus. This approach offers several advantages over more common consensus. Let's compare. Proof Of Work is a well-researched and stable consensus. However, it uses computing power in vain and sometimes requires a significant amount of time to produce blocks. Proof Of Stake - ownership of shares. Although it does not force the miner to do useless work, it has not been studied enough; has a number of technical problems and is difficult to implement and test. The most appropriate is Proof Of Authority. It doesn't do unnecessary work, is easier to implement than Proof Of Stake, is not so vulnerable and provides flexibility in the appointment of the manufacturer, if the blockchain is private, the speed is significantly higher. The main disadvantage of this consensus is weak decentralization. However, based on the example of widespread blockchains, we know that real decentralization is not what users are told. Typically, there are a number of large miners in control of the network. In the TechGen blockchain, decentralization can be controlled by the number of nodes. And it will not differ much from the real situation with well-known blockchains.

## Setting up manufacturers

The number of producers can be configured using node transactions according to the corresponding rights. This allows you to quickly change the configuration in order to achieve the minimum network latency with the maximum number of nodes. A node with the authority to add or remove a producer from the consensus submits a transaction. As soon as a transaction hits the blockchain, other manufacturers will start updating their list of nodes. And based on this list, new blocks will be filtered.

## System of rights

The system of rights assumes that the BC starts working when the supernode is created. Node that has a complete list of rights. Further, the owner of the node implements the mechanism for delegating rights. It is assumed that the minimum set of rights required for the operation of the node is as follows:

1. The right to produce blocks;

2. The right to delegate a block production;

3. The right to delegate of the right to delegate of the production of blocks.

Of course, this is just a basic list based on which the node can start working. Unfortunately, classic blockchains do not have a system of rights. Therefore, the transition of already existing BC to the system of rights is extremely resource-intensive. Recently the rights systems have started to appear in blockchain units. Such a system is a significant advantage and a quantum leap forward compared to other blockchains. In the TechGen blockchain, the development of a rights system is simple, because the technology is designed to be extensible. The system of rights allows in some cases to implement simple business logic for clients at the stage of finalizing the BC. Let's give an example of the implementation of the system of rights. For example, you can realise the right to manage funds in your account. Thus, without the need to access your account password, external control can be entered. Such a case in classic blockchains requires writing smart contracts at best, or even writing bytecode. Naturally, when the client is an ordinary user, and not a programmer, he cannot use such logic. The system of rights allows you to significantly reduce the entry threshold for some business cases and increase the attractiveness of BC.

## Identification

As we wrote earlier, in the modern world, identification is an important legal aspect, and can also serve as a reason for refusing to perform any actions in accordance with the business logic embedded in the blockchain. A ban to use the BC by unidentified users is not an option. Therefore, such a decision is fraught with massive user refusal to use BC and a high entry threshold (one of the well-known blockchains with a similar consensus has such a problem and it strongly affects mass distribution). In TechGen, it is necessary to relieve the user of problems when getting to know the blockchain. In other words, on the one hand, it is necessary to leave the possibility of simple registration and at the same time provide the possibility of identification at will, if the user requires specific actions. As an example - the production of blocks. TechGen solves this problem in the following way. If the user wants to send funds, he just needs to generate a public and private key. This is a one-click action on a web service. Thus, we left the entry threshold as simple as possible. If the user wants to become a block producer, then he needs to go through identification. One of the mechanisms is the purchase in the identifier system. Paying for an identifier will avoid bulk generation of identifiers unless otherwise restricted. Hence, we get a flexible mechanism, which, on the one hand, relieves the BC of complications when entering the user, and on the other hand, allows you to identify the user if necessary. Unfortunately, at the moment there are no blockchains with the same flexible system.

## Technology

As we wrote earlier, in developing TechGen were used modern technologies, which made it possible to significantly reduce the code base, increase security and simplify development. The codebase of most blockchains is a fork of existing ones. And the existing ones were written in languages that were created mainly in the last century.

Today we have the opportunity to write in languages that were specially created to solve such problems. Our choice fell on the Scala language. Let's describe the advantages:

1. It was designed specifically to solve scalability problems, to be simple and therefore has the appropriate tools out of the box.

2. New languages have a significant drawback! Lack of a large number of libraries that have been written for older languages over the years. But there is no such problem in Scala. The point is that Scala is backward compatible with the Java Virtual Machine. And we all know that Java is a simple industrial language that has been at the top of the list for many years and is mainly used in business and banking. This way Java and Scala libraries are compatible. We can use the Scala library from Java and vice versa.

3. Scala language combines several paradigms: object-oriented and functional. Nowadays, developers are increasingly looking towards the functional paradigm. And for a reason. After all, this approach allows you to avoid a lot of errors in multithreaded programming.

4. Significant reduction in the code base. If we compare Scala with C or Java, then solving the same problem in Scala requires writing code 2 or even 5 times less.

5. Scala is statically typed. This means that many errors can be detected at compilation step.

6. Experience in solving routine tasks of working with data structures in Java was taken into account when designing Scala. Therefore, many tasks that were solved in Java by writing routine code in Scala are solved in one line. Ready-made solutions for everyday problems mean that a programmer does not have to constantly write almost the same code, in which mistakes can be made even due to trivial inattention.

7. The problem of multiple inheritance is solved. In Scala, you can flexibly combine functionality for classes, while in Java, in some cases, this functionality had to be duplicated, which led to an increase in the code base and the potential to make mistakes when fixing such functionality. These are not all the advantages of the chosen technology. However, all of these benefits boil down to the three most important business concerns: reduced time to create a product, reduced cost, and increased productivity. Also, experts agree that the functional paradigm is obviously a suitable solution for blockchain problems.

## Performance

The concept of productivity includes primarily two factors. Let us briefly outline them in relation to their implementation in the TechGen blockchain:

1. The number of blocks produced per unit of time - in other words, the rate of block production. Since we are dealing with a consensus that does not depend on the manufacturer's work, then theoretically the speed is limited only by the power of the manufacturer's machine. And, since technically even a relatively weak and not modern computer is capable of producing more blocks than the current level of development requires, then the speed can be considered an unlimited technology of the blockchain itself. Which is a significant advantage of the chosen consensus.
2. The volume of a transaction in a block is technically unlimited in the implementation of the TechGen blockchain.

It should also be understood that both of these factors, although practically not limited in the implementation of the blockchain itself, have limitations in other places. Namely: the first and especially the second factor depends significantly on the bandwidth of the network connection, its quality and latency. And, if we put an infinite number of transactions in the block, then we can clog the network, and as a result, allow an attack on the consensus. Same relates with blocks. If we generate a huge number of blocks in the network, then other network participants will not have time to synchronize and thus the network will diverge. Therefore, for security purposes, the blockchain has the ability to flexibly configure these parameters. As a rule, these parameters are selected relatively to the current development of the Internet at the time of testing the network. From the moment of its launch, the parameters are already assigned as constant and in the future can only be changed in the fork. However, it is clear that the constant production of blocks can lead to empty blocks. In this case, the TechGen blockchain has the ability to build optimization. If someone decides to make money on an empty block, the network will reject it. Such an implementation will optimize the space occupied by blocks in the TechGen blockchain. It is also worth keeping in mind that the final parameters are selected depending on the goals and testing of the network. For example, in some networks, the time is counted by the number of blocks. Therefore, TechGen blockchain is not limited in consensus by the speed of block production, unlike Proof Of Work blockchains. All other parameters are configured as a result of testing and adjusting the network.

## Efficiency

As we wrote earlier, by network efficiency, we mean the ratio of transaction, execution speed and block occupancy. We can generate blocks a thousand times per second or more. This already depends on the performance of the computing complex on which the node is located. However, if the blockchain serves a large number of transactions. Blocks are generated 10 times per second. And their occupancy is no more than 50%. And yet the blocks are small enough to be transmitted over the network. Then, in this case, it is logical to build in a mechanism for dynamic or static (based on blockchain rights) regulation of the filling of blocks and the speed of their generation. This allows you to quickly adapt blockchain to changing business needs. For example, speed is more important for payment transactions, and for signing and storing information.

## Architectural support for the implementation of new assets

Creating and testing smart contracts is a very time consuming task. However, we know how important it is to be able to support business project tokens. Modern blockchains strive to support coins without smart contracts in the early stages of development. Which allows you to do swaps (moving tokens from one blockchain to another) of coins. And this is very important for the popularization of the young blockchain [project]. TechGen is architecturally capable of implementing such a mechanism.