BDF - MODULE 4

Multichain Blockchain

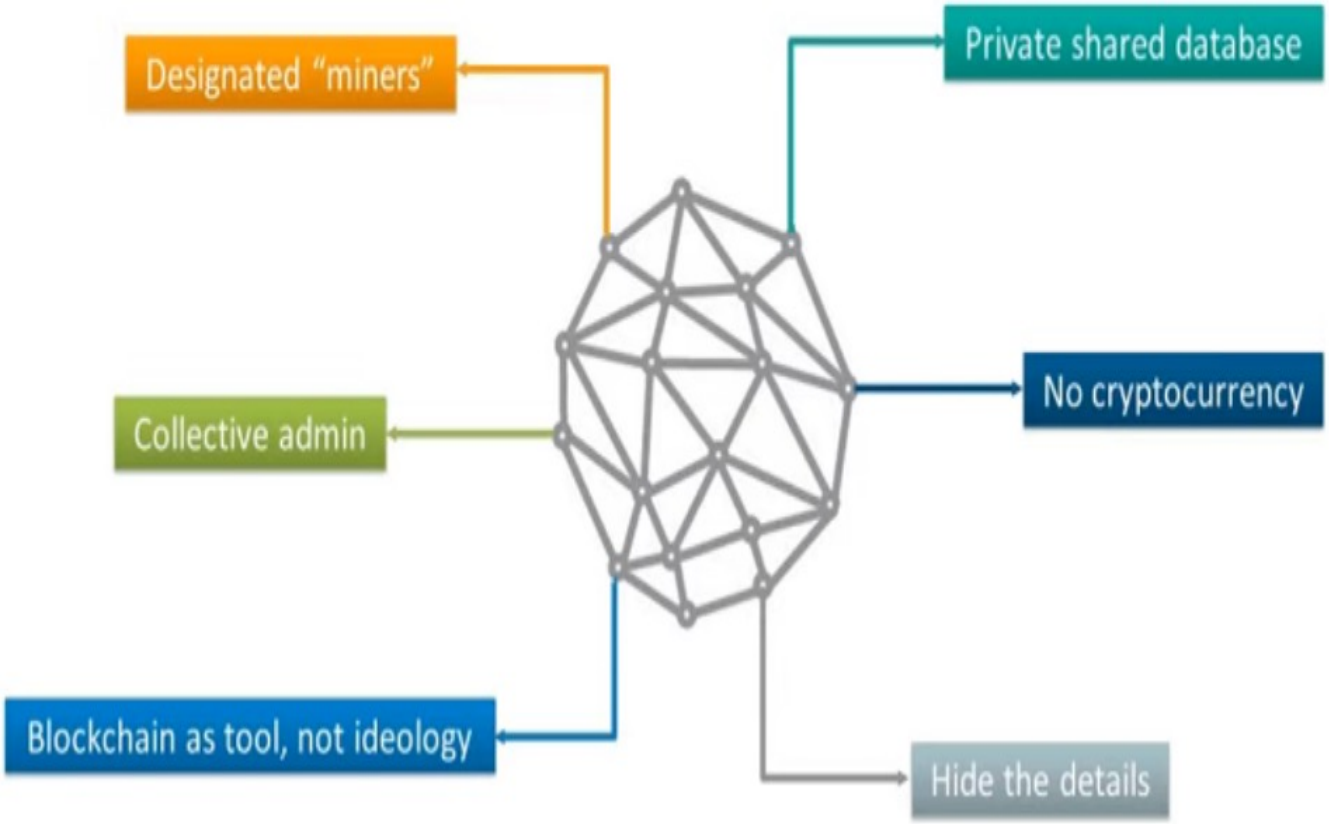
What Is MultiChain?

About Multichain Blockchain

- Lightweight
- Bitcoin fork (soft fork - started in 2014)
- No Smart contract (limitation)
- No native/default currency/tokens (customize)
- No PoW
- Concept of Assets, Streams, Mining
- No support for Docker container/emulator (limitation)

Blockchains For Enterprise

Following parameters should be kept in mind while implementing Blockchain in Enterprise domain:



Introducing MultiChain

01 MultiChain is an off-the-shelf platform for the creation and deployment of private Blockchains, either within or between organizations

02 Derived from Bitcoin core software, hence it is compatible with Bitcoin ecosystem

03 MultiChain supports Windows, Linux and Mac servers

04 Provides a simple API and command line interface that makes it easy to maintain & deploy

MultiChain Helping Enterprise In Blockchain

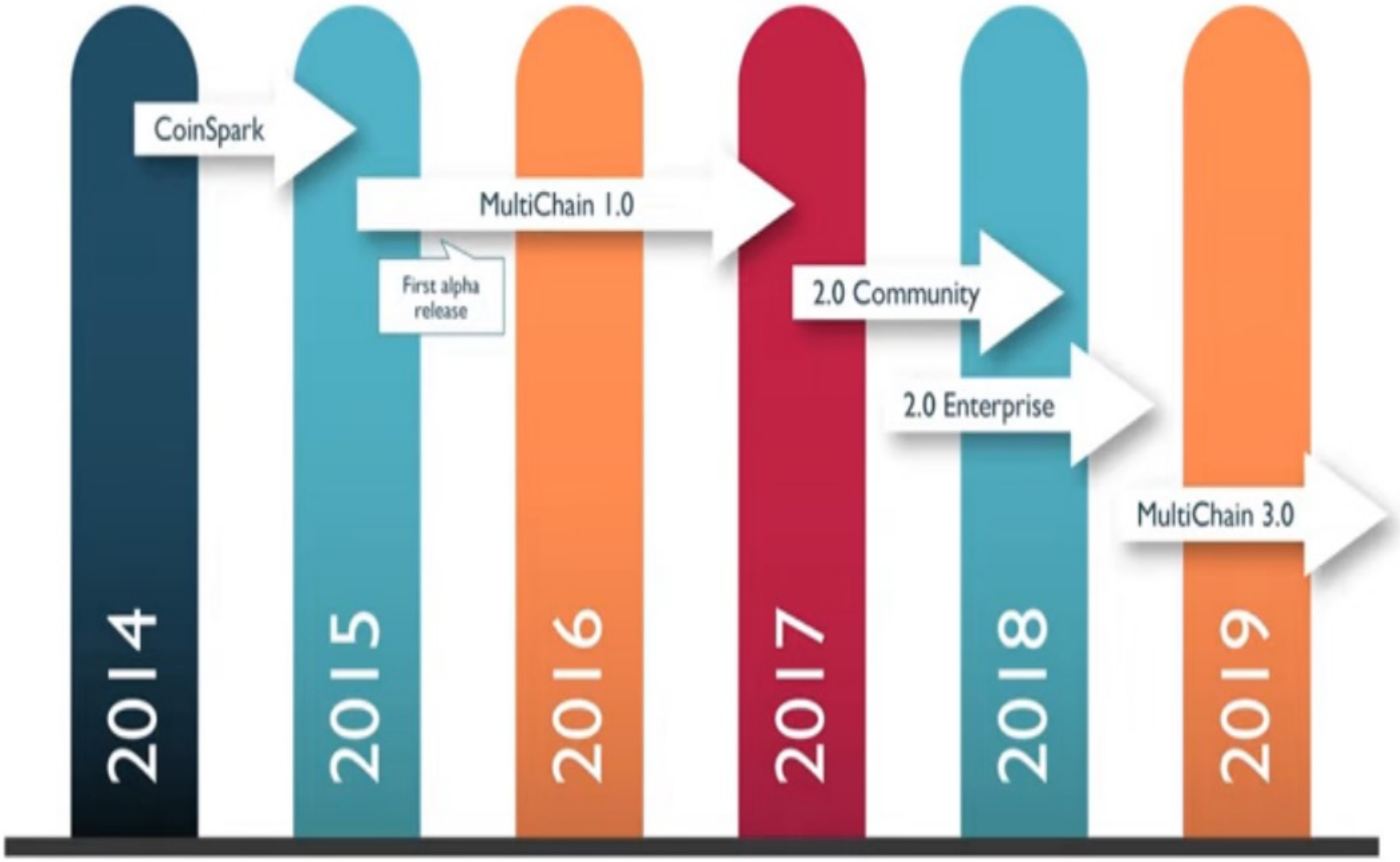
Managed permissions
Dynamically control who can connect, send and receive transactions, create assets, streams and blocks.
Each Blockchain is as open or as closed as you need

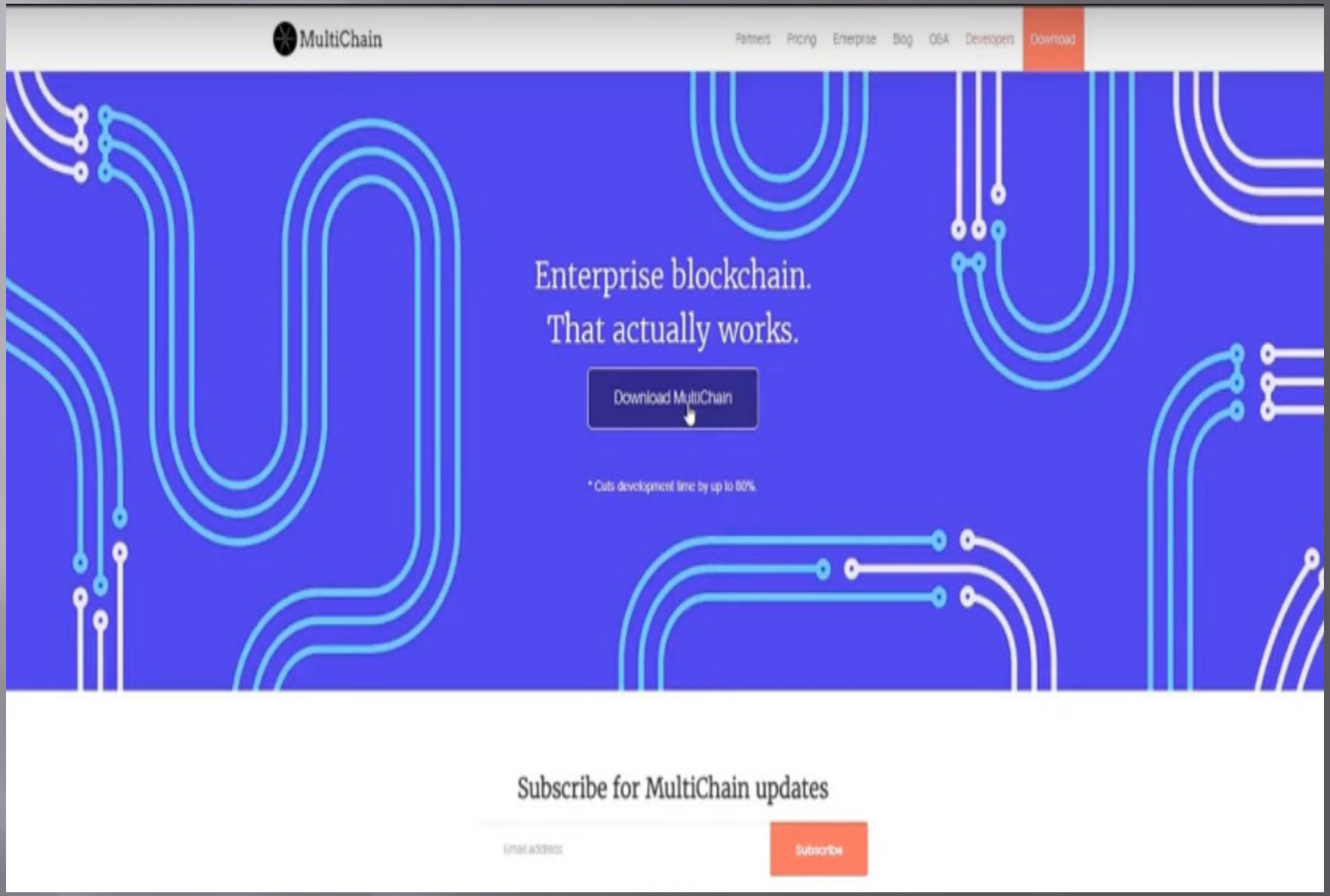
Rapid deployment
Just two simple steps to create a new Blockchain, and three to connect to an existing one. Deploy unlimited Blockchains per server for cross-chain applications

Unlimited assets
Issue millions of assets on a Blockchain, all tracked and verified at the network level.
Perform safe multi-asset and multi-party atomic exchange transactions

Data streams
Create multiple key-value, time series or identity databases on a Blockchain. Ideal for data sharing, timestamping and encrypted archiving

MultiChain – Development Timeline




MultiChain
Partners
Pricing
Enterprise
Blog
Q&A
Developers
Download
Enterprise blockchain.
That actually works.
Download MultiChain
* Cuts development time by up to 80%.
Subscribe for MultiChain updates
Email address
Subscribe

MultiChain Download Developers Q&A Blog Enterprise Pricing Partners Search

Get started by downloading the open source MultiChain 2.0 Community, or get a free trial of MultiChain 2.0 Enterprise with high-end blockchain features. MultiChain Enterprise is fully compatible with all Community API commands and parameters, and you can mix Community and Enterprise nodes on a single network. See below for a comparison:

Feature	Benefit	MultiChain 2.0 Community GPLv3	MultiChain 2.0 Enterprise Commercial
Smart Filters	Programmability	✓	✓
Permissions Management	Ease of Use	✓	✓
Unlimited Assets	Ease of Use	✓	✓
Data Streams	Data Management	✓	✓
Seamless Off-Chain Data	Ease of Use	✓	✓
Private Key Flexibility	Security	✓	✓
Stream Read Restrictions	Confidentiality	✗	✓
End-to-End Encryption	Confidentiality	✗	✓
Real-Time Data Feeds	Data Management	✗	✓
Selective Stream Indexing	Scalability	✗	✓
Selective Data Retrieval	Scalability	✗	✓
Off-Chain Data Purging	Compliance	✗	✓
		Community Download	Enterprise Download

Bitcoin To Private Blockchain

Many aspects of MultiChain's design are aimed at enabling smooth transitions between private Blockchains and the bitcoin Blockchain in either direction.

Based on a fork of Bitcoin Core. Code changes are localized, enabling future bitcoin enhancements to be merged in

Its multicurrency and messaging features work very similarly to the CoinSpark protocol for enhancing bitcoin transactions

It uses bitcoin's protocol, transaction and Blockchain architecture, with changes only to the handshaking process

It can act as a node on the regular bitcoin network via a simple protocol setting in the per Blockchain configuration file

Its interface is fully compatible with that of Bitcoin Core, with all additional functionality provided by new commands

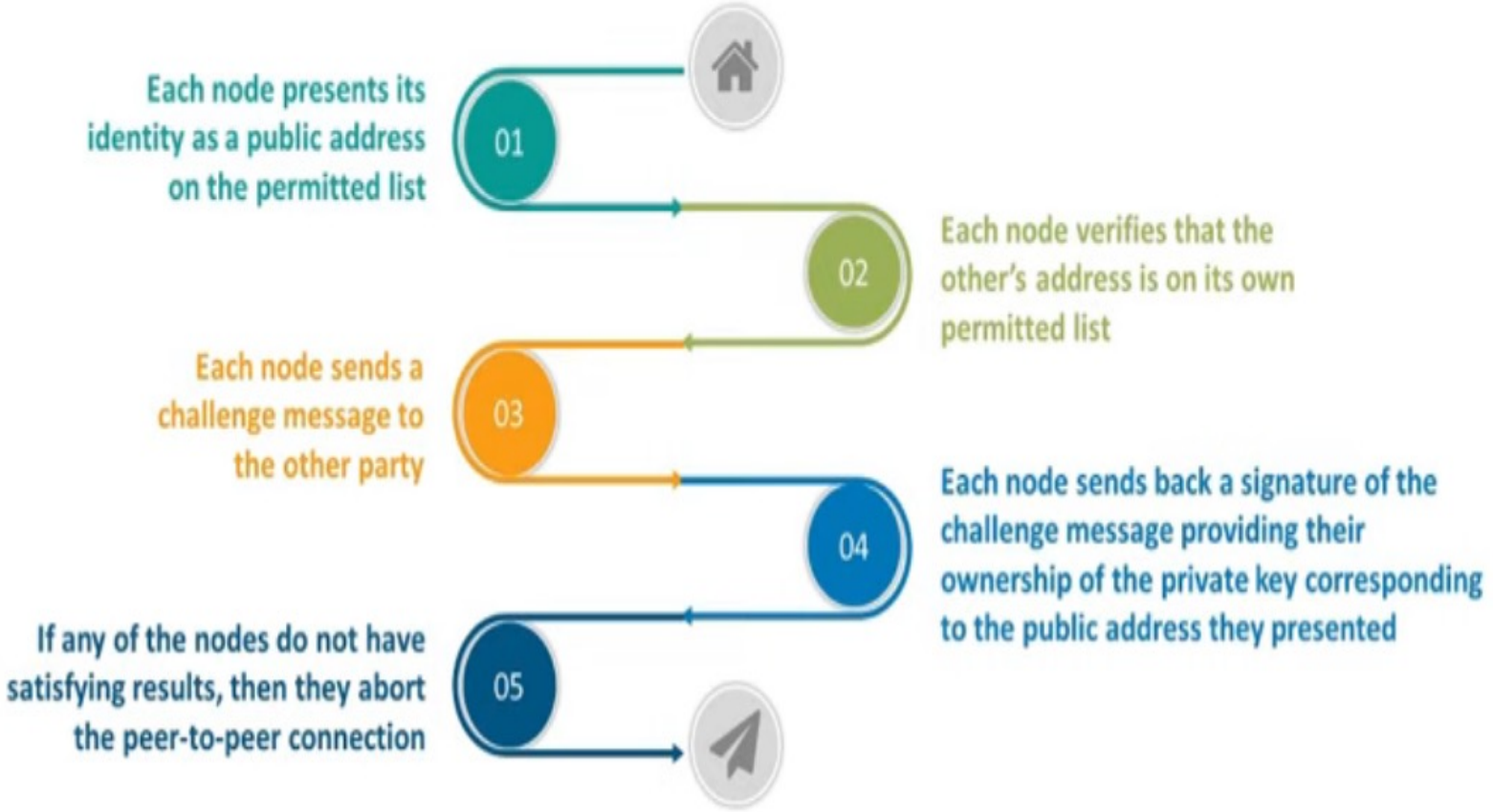
Aim Of MultiChain

The main aim of MultiChain is:

1. To ensure that visibility of Blockchain's activity must be chosen through the chosen participants
2. To control over which transaction are permitted
3. To enable more secure mining with Proof-of-Work and its associated cost
4. The Blockchain system only stores transactions related to participants

The Hand-Shaking Process

MultiChain uses private key cryptography property to restrict Blockchain access to a list of permitted users, by expanding the "handshaking" process that occurs when two Blockchain nodes connect.



Use-Cases Of MultiChain

Lightweight financial systems

Examples: Crowdfunding, gift cards, loyalty points and local currencies especially in cases where assets are redeemable in more than one place

Provenance tracking

Example: Tracking the origin and movement across a supply chain, such as luxury goods, pharmaceuticals, cosmetics and electronics. And also documentation such as bills of lading or letters of credit

Inter-organizational record keeping

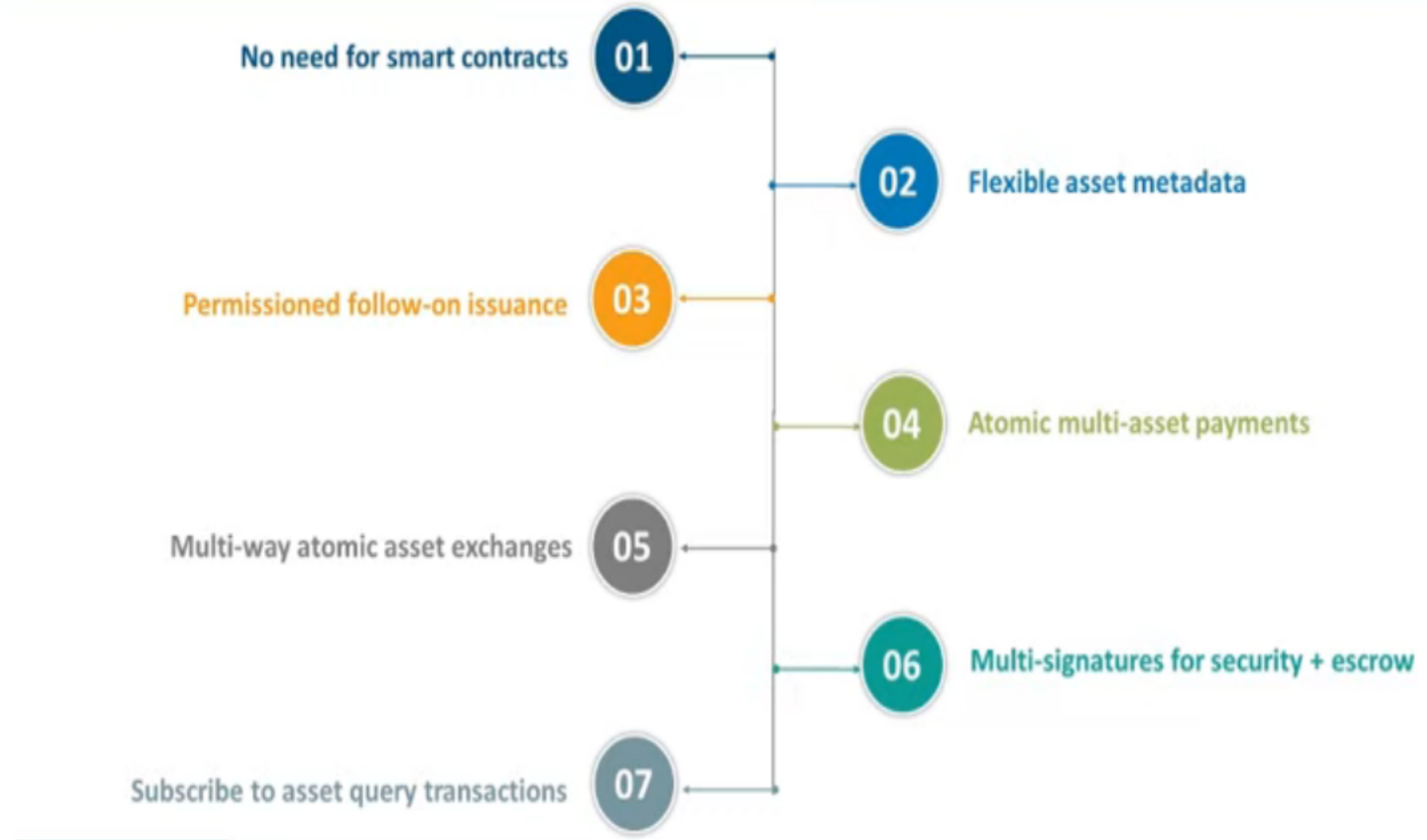
Example: Audit trail of critical communications between two or more organizations, say in the healthcare or legal sectors

MultiChain Permissions

MultiChain provides following permissions:

- Connect to network
- Send and receive transactions
- Write to a stream
- Issue assets
- Create streams
- Add blocks to chain
- Change permissions by consensus

MultiChain Assets



MultiChain Streams

Provide a natural abstraction for Blockchain use cases which focus on general data retrieval, timestamping and archiving

Any number of streams can be created & each stream acts as an independent append-only collection of items

Nodes choose which streams to index

Streams can be used to implement three different types of databases on a chain:

A key-value database or document store, in the style of NoSQL

A time series database, which focuses on the ordering of entries

An identity-driven database where entries are classified according to their author

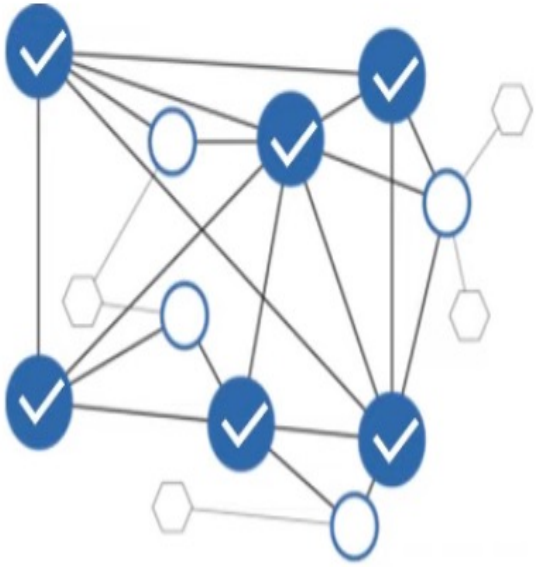
Consensus Model

Distributed consensus between identified block validators. There is one validator per block, working in a round-robin type of fashion

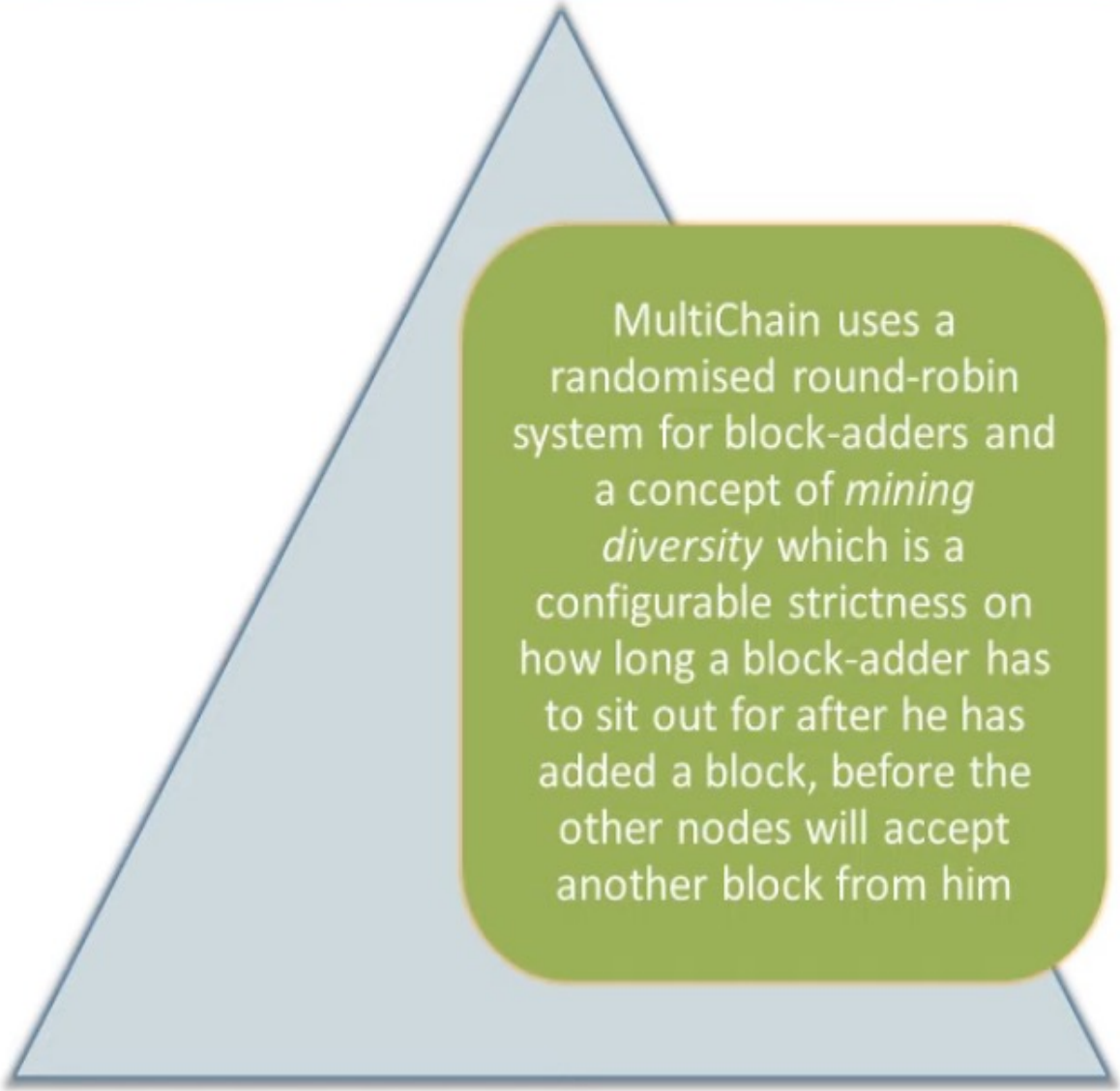
Every block digitally signed by creator. Only permissioned parties can “mine”

Mining diversity for distributed consensus

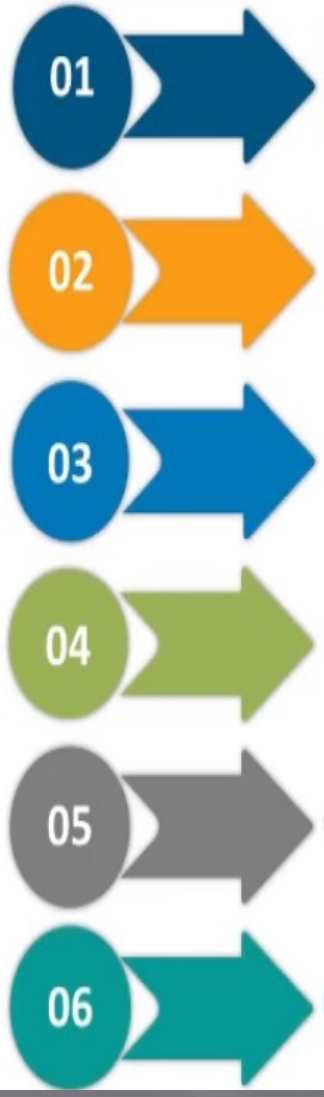
No proof-of-work or cryptocurrency



Mining In MultiChain Technology

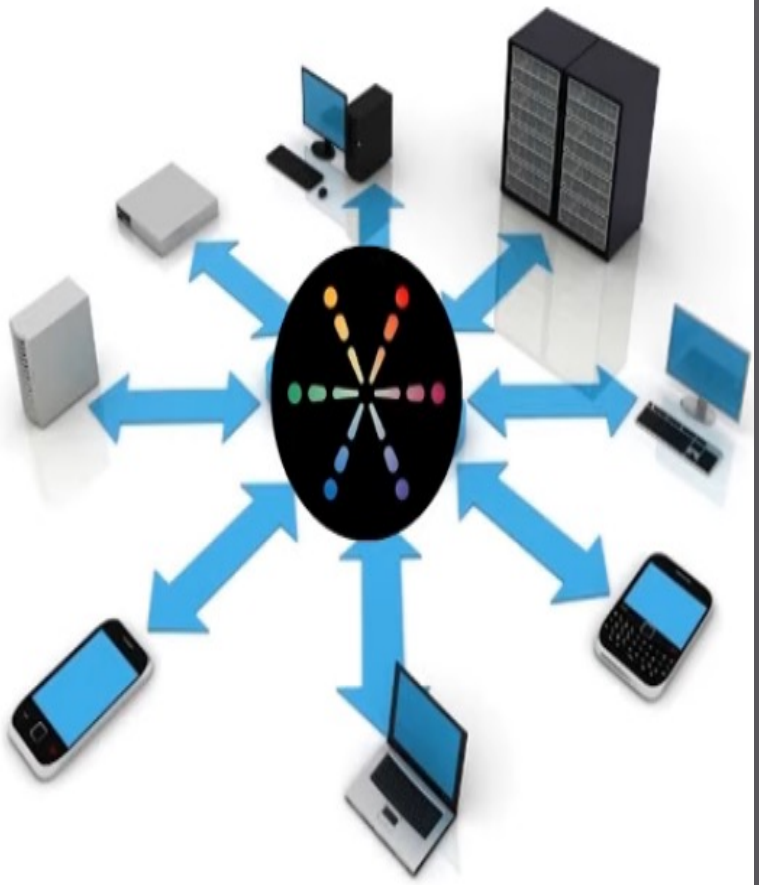


MultiChain Flexibility



Deployment Options

- Environment agnostic
  - Self-hosted in datacenter
  - Public or private cloud
  - Accessed as a service
- Nodes added simply and quickly
- API cleanly separates app from node
- Shared administration model
  - Smooth governance transitions

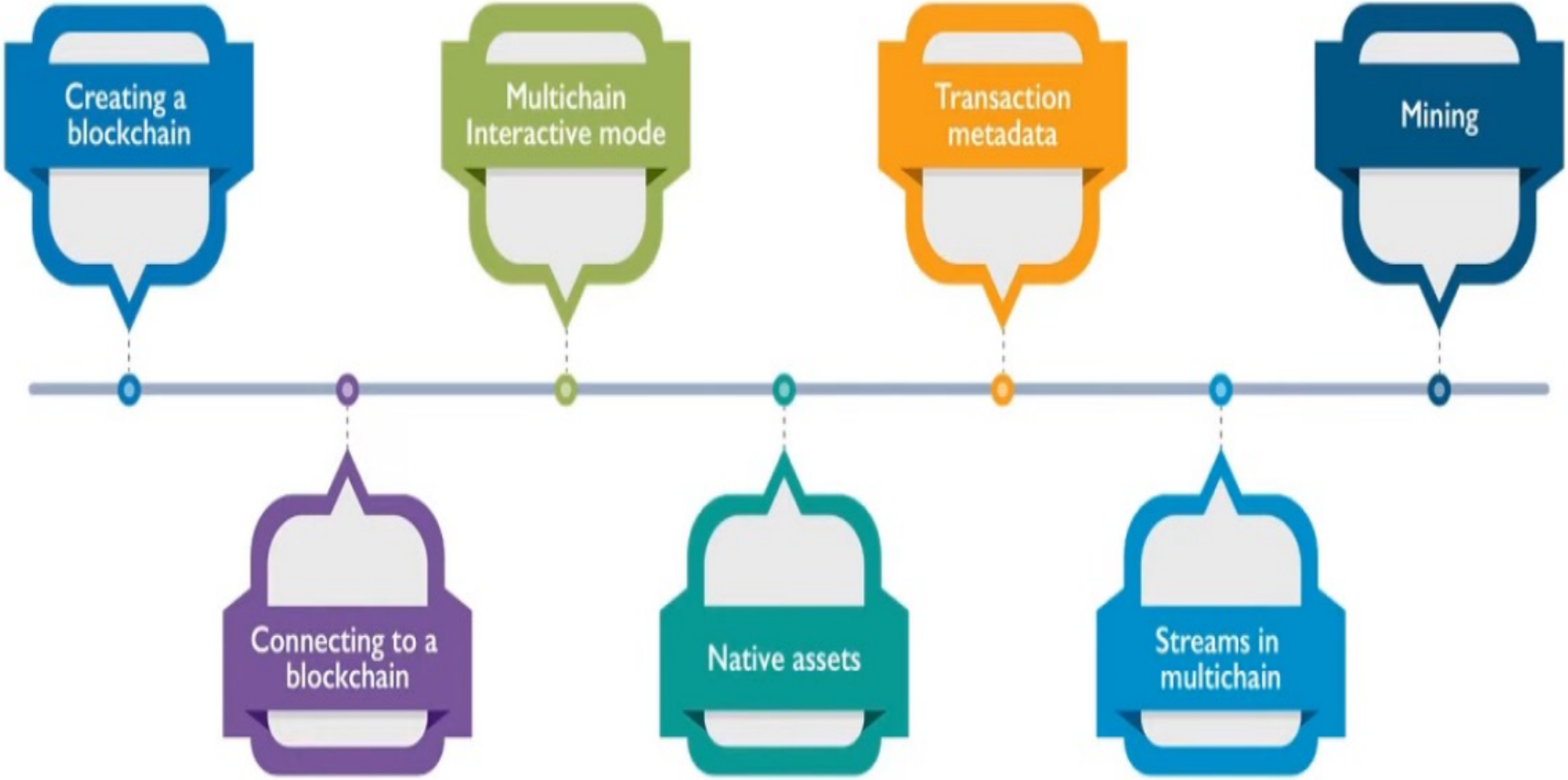


Security In MultiChain

- Forked from Bitcoin Core
  - $40B+ and over 8 years exposure to internet
- Full multi-signature support
- External key management
  - Bitcoin hardware security modules
- Consensus over data not execution
  - Avoids Ethereum-style hard forks
  - No need for fees to stop runaway

Steps To Create And Deploy Private Blockchain

Following are the steps to create & deploy private Blockchain:



This tutorial requires two server nodes. If you have not done so already, please download and install MultiChain Community or Enterprise Demo on each server. Note that one section of this guide applies to MultiChain Enterprise only.

1. Creating a blockchain

First we will create a new blockchain named `chain1`. On the first server, run this command:

```
multichain-util create chain1
```

If you are using Windows, first open a DOS command line in the directory where you installed the MultiChain executables. You can do this by navigating to that directory in the Windows Explorer, then typing `cmd` in the address bar at the top.

There are many blockchain parameters that can be modified, but we'll leave them on their default settings. Now initialize the blockchain, including creating the genesis block:

```
multichaind chain1 -daemon
```

You should be told that the server has started and then after a few seconds, that the genesis block was found. You should also be given the node address that others can use to connect to this chain.

Copy and paste the node address here: **chain1@[ip-address]:[port]**

2. Connecting to a blockchain

Now we'll connect to this blockchain from elsewhere. On the *second server*, run the following:

```
multichaind chain1@[ip-address]:[port]
```

You should be told that the blockchain was successfully initialized, but you do not have permission to connect. You should also be shown a message containing an address in this node's wallet.

Copy and paste the wallet address here: **1...**

Back on the *first server*, add connection permissions for this address:

```
multichain-cli chain1 grant 1... connect
```

Now try reconnecting again from the *second server*:

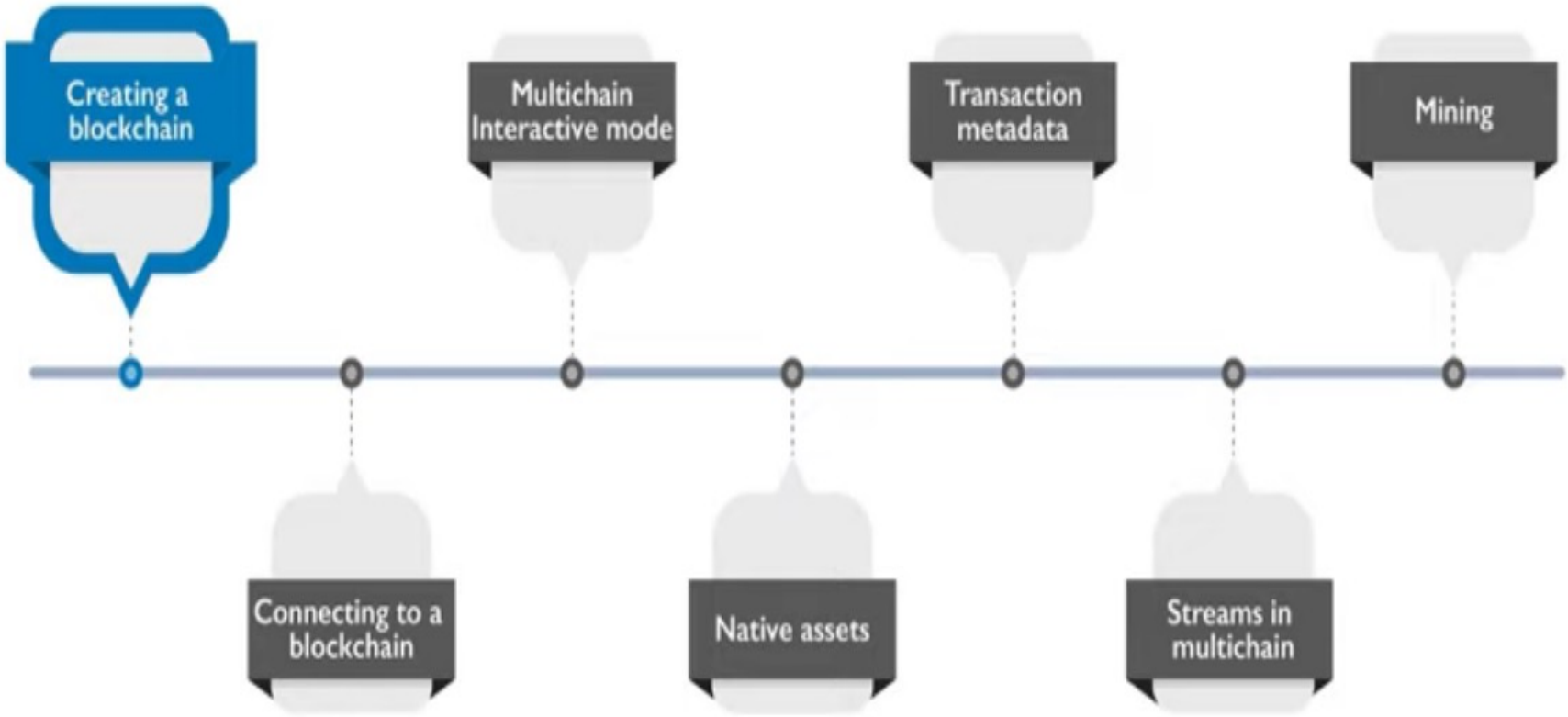
```
multichaind chain1 -daemon
```

You should be shown a message that the node was started, and it should display this second node's address.

3. Some commands in interactive mode

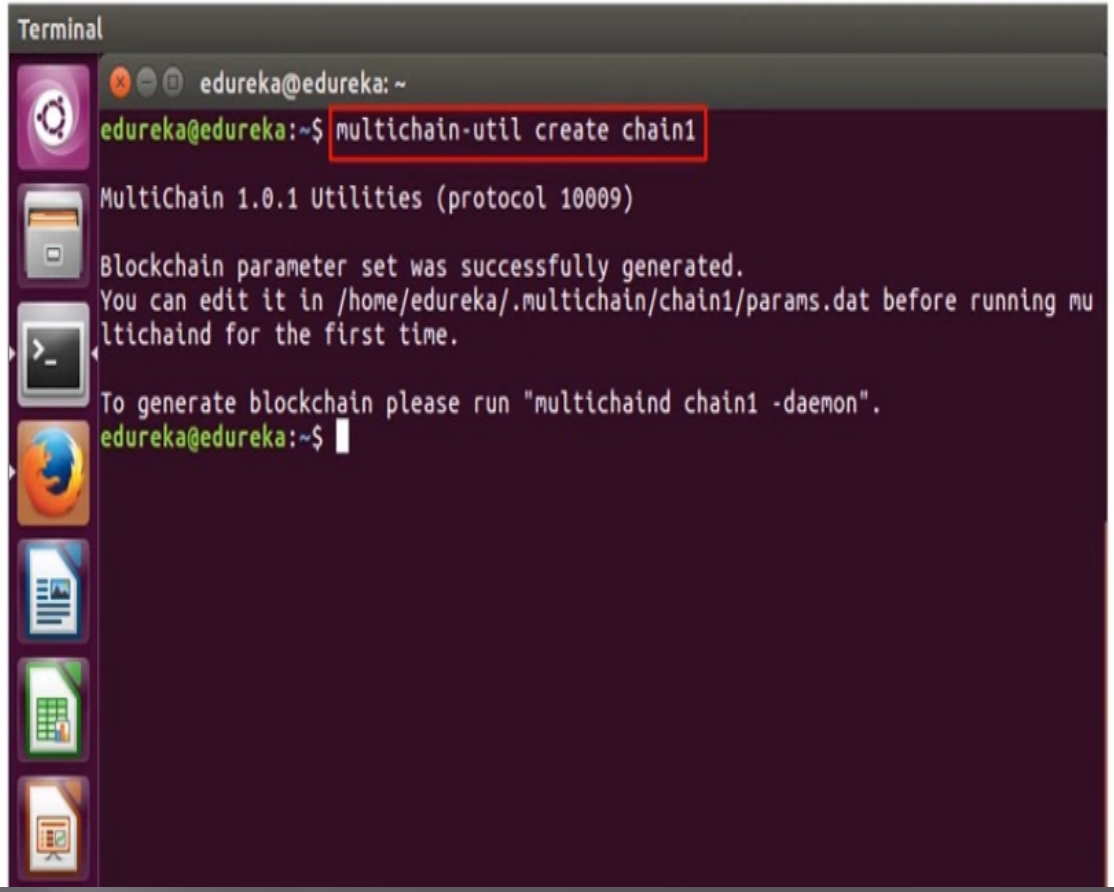
Before we proceed, let's enter interactive mode so we can issue commands without typing `multichain-cli chain1` every time. On *both servers*:

Step 1: Creating A Blockchain



To Create A Blockchain

- First we will create a new Blockchain named **chain1** (you can give the name of your choice).
- On the first server, run this command: `MultiChain-util create chain1`

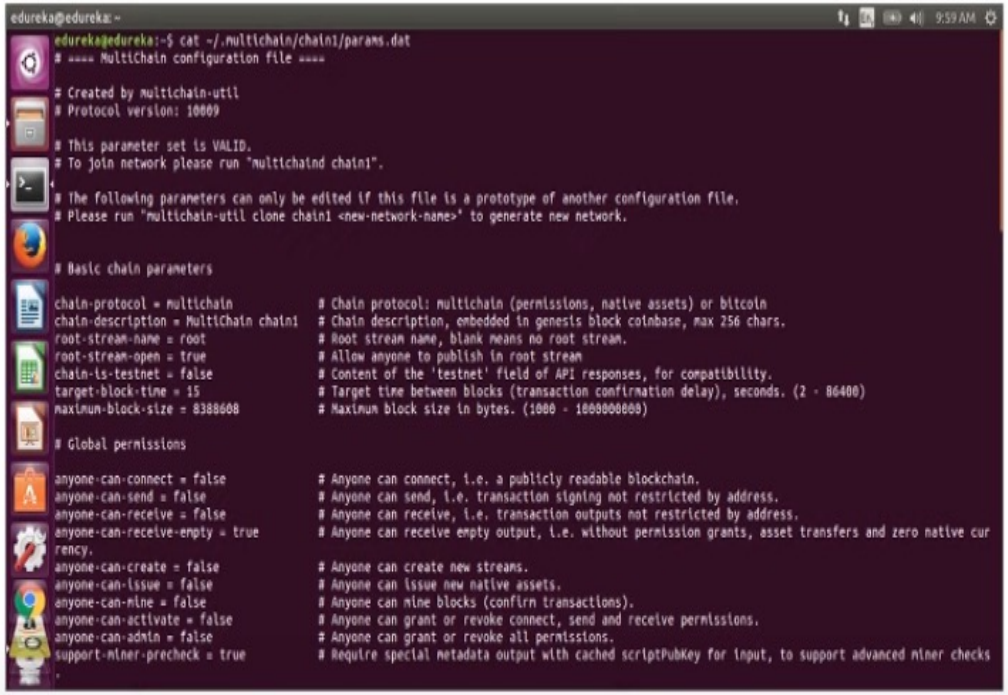


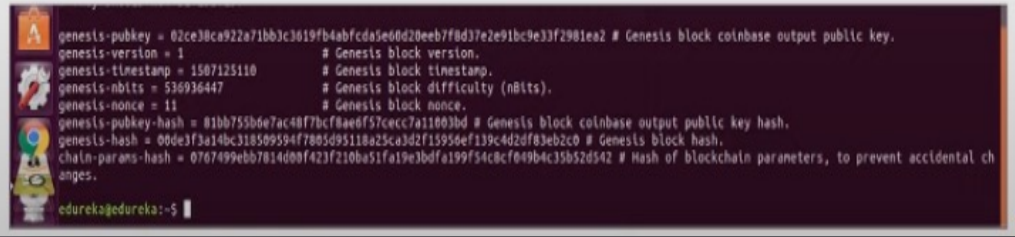
View Default Settings

- To view the Blockchain's default settings run the following commands:

```
cat ~/.MultiChain/chain1/params.dat
```

(Above command will list the parameters and settings of the Blockchain)





Note: List shown here is just a specimen. As the list was too long to display on the ppt

Adding Connection Permission For The Address

- Add connection permissions for this address:

```
MultiChain-cli chain1 grant 13aNooyp2w...hpBYd connect
```

Above command will grant permission to connect to the specified address

```
edureka@edureka:~$ multichain-cli chain1 grant 13aNooyp2wXPW5NzuxvzGEDtEG6Hxtq8L
hpBYd connect
{"method":"grant","params":["13aNooyp2wXPW5NzuxvzGEDtEG6Hxtq8LhpBYd","connect"],
"id":1,"chain_name":"chain1"}

bf58290f89543b61f498e09d48d2b509ea1516e4f8183c1cbd61aeb267fd8b78
edureka@edureka:~$
```

Note: The address here is unique for this particular Blockchain. It will be different when you run your Blockchain

Switching To Interactive Mode

- To enter the MultiChain interactive mode, on both servers run the following command:

```
MultiChain-cli chain1
```

Server 1 →

```
edureka@edureka:~$ multichain-cli chain1

MultiChain 1.0.1 RPC client


Interactive mode

chain1:
```

Server 2 →

```
edureka@edureka:~$ multichain-cli chain1

MultiChain 1.0.1 RPC client


Interactive mode

chain1:
```

Now that the Blockchain is working on two nodes, you can run the commands in this section on either or both