



BKU Blockchain Lab

Data Structures. APIs Usage

Presenters:

- **Huynh Sam Ha – 1610852**
- **Nguyen Van Tien – 1613535**

AGENDA

- 1. Factom Data Structures**
- 2. Installing Factom Environment**
- 3. How to use Factom Enterprise Wallet?**
- 4. Factom APIs Usage**
- 5. Summary**



Factom Data Structures

1. Chains - Grouping of Entries to an Application

- **Chains** in Factom are **sequences of Entries** that reflect the events relevant to ***an Application***.
- **Chain Name:**
 - A Chain Name is a value to ***uniquely identify a Chain***.
 - The **choice** of Chain Name is left up to the **user**.
 - It can be a random *number*, a *string* of text, a *public key*...

1. Chains - Grouping of Entries to an Application

- **ChainID:**

- A **ChainID** is a series of **SHA256** hashes of ***Chain Name segments***
- The ChainID is **32 bytes** long
- The algorithm hashes each segment of the Chain Name.
Those hashes are concatenated, and are hashed again into a single 32 byte value.
- ChainID = SHA256(SHA256(Name[0]) | SHA256(Name[1]) | ... | SHA256(Name[X]))

Chain

CHAIN ID

67a417820b1dc4edb6047602be06ced302635d5cf74a0a1512c45285137d617f

Chain

CHAIN ID

67a417820b1dc4edb6047602be06ced302635d5cf74a0a1512c45285137d617f

EXTERNAL IDS

hex ▼ d1aeb4dde7b9dddf7cd38dfb7fdf34f1de34e7cd9df39f3dddf76d3b7fde7a71adf7d37eb97fcef75aeb736dbdd74

hex ▼ 777e5ed7779c6b671e6f971ce9b6f4dfde1ae7471cf5eefaeb8eb8db7e9be5bd39e3de9ef5bf1be9e6ba75eeb97f5e78

hex ▼ d1ee357b475a6fcd6da6f9d5de3df777b477de9ef34d3bd366f8dbceda73675dfd71e77ddb66dbefdddf5fdbd79b

Entries 4

CREATED (UTC+0700) HASH

2018-09-27 15:05 3cd295501f9c46cf46ce19a342e17ec77ee736b05ddee8146335d90fe9fda5c0

2018-09-27 15:05 e38ca219d49416eef0c89e55148be81aad1c897e29644533a66c6264ee40792b

2018-09-27 15:09 4983923953206b9dae152a62e206a3cee1c82d0a2b447c7a46d4e54bc8297905

2018-09-27 15:09 fe6d163773bf46b93443de1b11f2e57cdfdb7f35bee0a4c0960218ed5e7cb75e

2. Entries - Contains an Application's raw data or a hash of its private data

- An **Entry** is the element which **carries** user data.
- **Entry Hash**
 - To calculate the Entry Hash, first the Entry is ***serialized*** and ***passed into*** a **SHA512** function.
 - The **64 bytes output** from the SHA512 function is **prepended** to the ***serialized Entry***.
 - The **Entry + Prependage** are then fed through a **SHA256** function, and the output of that is the ***Entry Hash***.

2. ENTRIES

| Entry Commit

Entry Commit

- An Entry Commit is a **payment** for a specific Entry
- They are **collected into** the ***Entry Credit chain*** as proof that a balance should be decremented
- **Cost is 1 EC per partial KiB. Empty Entries cost 1 EC.**

Entry

HASH

8f8d95c87d5ec28fb0e1ecde7611965aa9a6b0424b7b5409482af1fb047830da

CHAIN

9005bb7dd69fb9910ee0b0db7b8a01198f03623eab6dadf1eba01f9dbc207577

PARENT ENTRY BLOCK

KEYMR: bf04cdca0f666b871f4f5d96588758fb593ed34b0c9be7ff3cdc34a770694c30

CREATED (UTC+0700)

Thursday, September 27, 2018, 21:06

EXTERNAL IDS

raw

Hash

raw

a07317242f1bba9f0ac00bdd30ald8e154e4b2d1ale24cc824deb72cf39af218

CONTENT

raw

eHaqXjbInkpsYldSmexVfTO3FvRbluXDb+gofd9nvjY=

Entry Commit Structure

data	Field Name	Description
Header		
varInt_F	Version	starts at 0. Higher numbers are currently rejected. Can safely be coded using 1 byte for the first 127 versions.
6 bytes	milliTimestamp	This is a timestamp that is user defined. It is a unique value per payment. This is the number of milliseconds since 1970 epoch.
32 bytes	Entry Hash	This is the SHA512+256 descriptor of the Entry to be paid for.
1 byte	Number of Entry Credits	This is the number of Entry Credits which will be deducted from the balance of the public key. Any values above 10 are invalid.
32 bytes	Pubkey	This is the Entry Credit public key which will have the balance reduced. It is the ed25519 A value.
64 bytes	Signature	This is a signature of this Entry Commit by the pubkey. Parts ordered R then S. Signature covers from Version through 'Number of Entry Credits'

| Chain Commit

Chain Commit

- A Chain Commit is a **simultaneous payment** for a specific Entry and a payment to allow a **new Chain** to be ***created***
- They are collected into the **Entry Credit chain** as proof that a balance should be decremented

Chain Commit Structure

data	Field Name	Description
varInt_F	Version	starts at 0. Higher numbers are currently rejected. Can safely be coded using 1 byte for the first 127 versions.
6 bytes	milliTimestamp	This is a timestamp that is user defined. It is a unique value per payment. This is the number of milliseconds since 1970 epoch.
32 bytes	ChainID Hash	This is a double hash (SHA256d) of the ChainID which the Entry is in.
32 bytes	Commit Weld	SHA256(SHA256(Entry Hash ChainID)) This is the double hash (SHA256d) of the Entry Hash concatenated with the ChainID.
32 bytes	Entry Hash	This is the SHA512+256 descriptor of the Entry to be the first in the Chain.
1 byte	Number of Entry Credits	This is the number of Entry Credits which will be deducted from the balance of the public key. Any values above 20 or below 11 are invalid.
32 bytes	Pubkey	This is the Entry Credit public key which will have the balance reduced.
64 bytes	Signature	This is a signature of this Chain Commit by the pubkey. Parts ordered R then S. Signature covers from Version through 'Number of Entry Credits'

3. Entry Block Layer

Organizes references to Entries

Entry Credit Block

Factoid Block

Entry Block

Administrative Block

Blocks ⑥

(included in this directory block)

ADMIN BLOCK	1d16ca20b7d598c4fe08c8ece7ed31fb9781b6c3e0c6ad99ccaf592c9a92f8ba
-------------	--

ENTRY CREDIT BLOCK	3058c8e4e8541be1223b3bd867ead91bf6745c329bc133c7945d5fe51a16baea
--------------------	--

FACTOID BLOCK	55e760c3856fb3e2fff32c7fa70a53838333b6e5d923a95d894cde2572decfca
---------------	--

ENTRY BLOCK	72381b27de5b8123ad04ea90512b1e9328dc88041de1f5a7dc24a7e1fc7c9c52
-------------	--

CHAIN	9005bb7dd69fb9910ee0b0db7b8a01198f03623eab6dadf1eba01f9dbc207577
-------	--

ENTRY BLOCK	d99035eb3aea64d19bd1b590c25ba0a19a1eaf7530a2d35e8c970b64a46a50a2
-------------	--

CHAIN	1a4688541932aab08bec4c631b489d6cbc2389a28ad31df596da80c7a68c39a9
-------	--

ENTRY BLOCK	c7d94cfa2c4026bfa8e541e38cb5921bb1d16ed4351614360f6a34493c753bfc
-------------	--

CHAIN	046ed3414d901c71f0d0fa436fdcfbfe807001878ce1d6a936166b14339aff30
-------	--

Entry Credit Block:

- + An **Entry Credit (EC) Block** is a data structure which packages *Chain Commits*, *Entry Commits*, and *EC* balance increases over a **10 minute period**
- + The Entry Credit Block consists of a **header** and a **body**.

Entry credit block

HASH

3058c8e4e8541be1223b3bd867ead91bf6745c329bc133c7945d5fe51a16baea

PARENT DIRECTORY BLOCK

HEIGHT: 160169 KEYMR: d3ca63dd5233efa061d29eb7980ee48d056abf6db9b988c6c97a3f7a2c4317bb

Entries 285

CREATED (UTC+0700) HASH

2018-09-27 20:24 7344c6d964cb267660f968633d3eacac5cab51f6f629eabd1b8f2c04be44d207

2018-09-27 20:24 5cf9fe28618bec23312cb1cd11c48e61f9d27d1723878ee9479415e69daf06ce

2018-09-27 20:24 62adeb466545ea9877bb4e0ebd4385cdef8ec6d9a507502113d29f6c6210a974

Entry Credit Block

Header	
32 bytes	EC ChainID
32 bytes	BodyHash
32 bytes	PrevHeaderHash
32 bytes	PrevFullHash
4 bytes	DB Height

varInt_F	Header Expansion Size
Variable	Header Expansion Area
8 bytes	Object Count
8 bytes	Body Size
Body	
variable	All objects

Factoid Block

- + Factoid Block is a data structure which packages **Factoid transactions** over a **10 minute period**.
- + The Factoid Block consists of a ***header*** and a ***body***.

Factoid block

INPUTS

0 FCT

OUTPUTS

0 FCT

ECS CREATED

0 EC

KEYMR

55e760c3856fb3e2fff32c7fa70a53838333b6e5d923a95d894cde2572decfca

PARENT DIRECTORY BLOCK

HEIGHT: 160169 KEYMR: d3ca63dd5233efa061d29eb7980ee48d056abf6db9b988c6c97a3f7a2

Transactions ①

CREATED (UTC+0700) TRANSACTION ID

Factoid Block

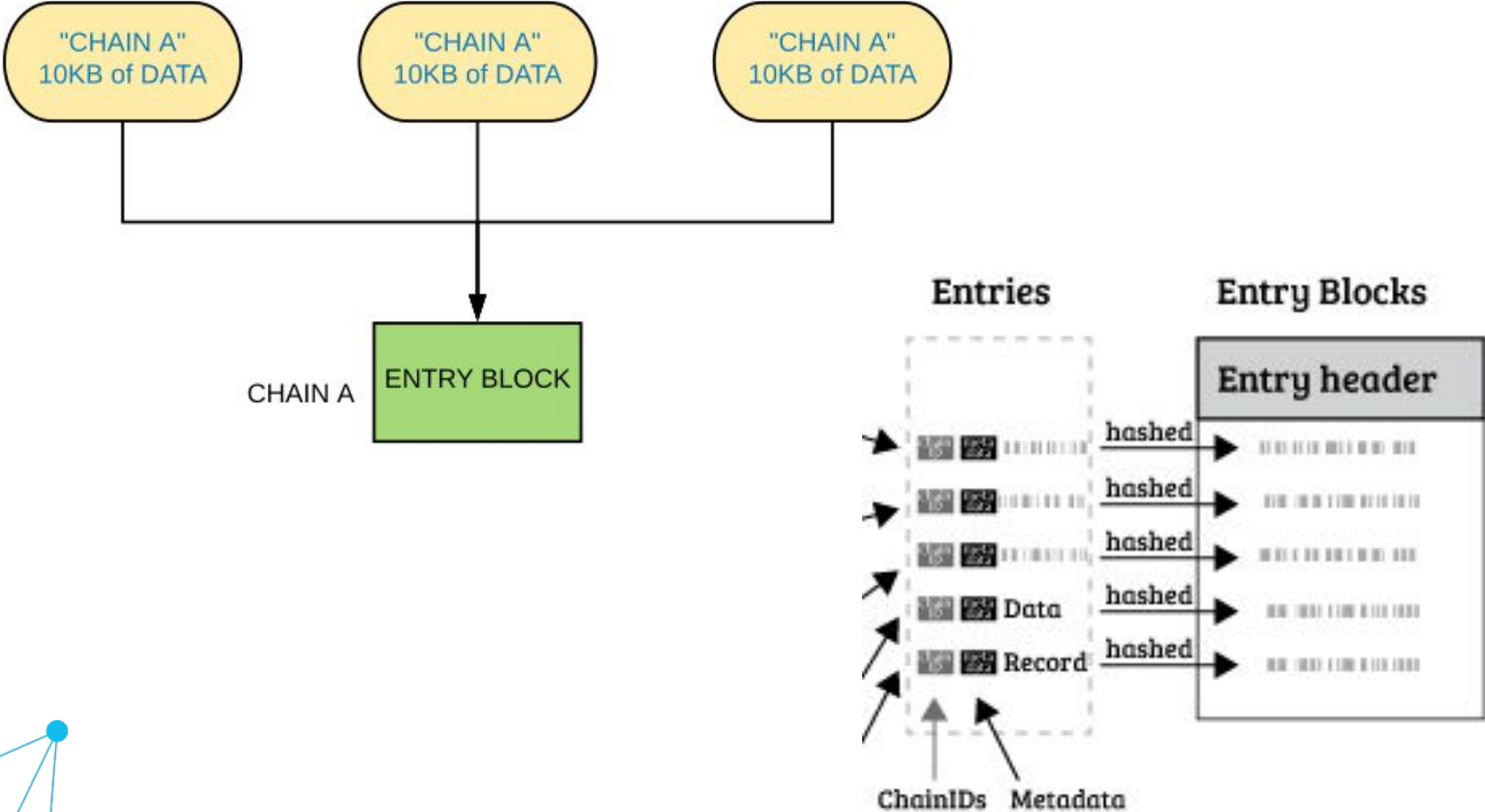
Header	
32 bytes	Factoid ChainID
32 bytes	BodyMR
32 bytes	PrevKeyMR
32 bytes	PrevLedgerKeyMR

8 bytes	EC Exchange Rate
4 bytes	DB Height
varInt_F	Header Expansion Size
Variable	Header Expansion Area
4 bytes	Transaction Count
4 bytes	Body Size
Body	
variable	All objects

| Entry Block

- + An Entry Block is a data structure which packages references to Entries all sharing a ChainID over a 10 minute period
- + The Entry Blocks form a blockchain for a specific ChainID.
- + The Entry Block consists of a header and a body.

Entry Block



Entry block

NEXT ENTRY BLOCK

549799aa02ce9ff04b4e6fa8c7993ff7055ea64b5efee24a4820914bd94d5fe9

KEYMR

72381b27de5b8123ad04ea90512b1e9328dc88041de1f5a7dc24a7e1fc7c9c52

PREVIOUS ENTRY BLOCK

5f66d7c3d90466c3e503c595f9b2d1fda088df9bb3c97d9d66af617c5ce61a00

CHAIN

9005bb7dd69fb9910ee0b0db7b8a01198f03623eab6dadf1eba01f9dbc207577

PARENT DIRECTORY BLOCK

HEIGHT: 160169 KEYMR: d3ca63dd5233efa061d29eb7980ee48d056abf6db9b988c6c97a3f7a2c4317bb

STARTED (UTC+0700)

Thursday, September 27, 2018, 20:15

Entries (283)

CREATED (UTC+0700) HASH

2018-09-27 20:16 212716ed408c388832e86985e32d2ac00b164bb4885362b18581df1b59b9afb3

2018-09-27 20:16 427db2ebd66f0e6412f4eb17a299b7fb0355a985fdeb9bf6d5d184428914d5f1

Entry Block

Header	
32 bytes	ChainID
32 bytes	BodyMR
32 bytes	PrevKeyMR
32 bytes	PrevFullHash

4 bytes	EB Sequence
4 bytes	DB Height
4 bytes	Entry Count
Body	
32 bytes	All objects

| Administrative Block

- + Administrative Block is a special block which accompanies this Directory Block.
- + It contains the ***signatures*** and ***organizational data*** needed to validate previous and future Directory Blocks.
- + It has a **LookupHash**, which is a SHA256 of the entire block. The **LookupHash** is included in the directory block body paired with the Admin ChainID.

Admin block

HASH

1d16ca20b7d598c4fe08c8ece7ed31fb9781b6c3e0c6ad99ccaf592c9a92f8ba

PARENT DIRECTORY BLOCK

HEIGHT: 160169 KEYMR: d3ca63dd5233efa061d29eb7980ee48d056abf6db9b988c6c97a3f7a2c4317bb

Entries (22)

CREATED (UTC+0700) CONTENT

2018-09-27 20:15

prevdbsig:

sig: 36115b8f152987586b796ca73cbecda110a1caf2a2df1378aaf05c490be09489654c4470c369366360055e:

pub: 3f4abe88347f304fddb377542af1ffae43ed935b6cadaf44486f92ff2be2a90b

identityadminchainid: 8888887f5125bfc597a05eca2db64298b88a9233dafdeb44bc0db7d55ee035aa

adminidtype: 1

Administrative Block

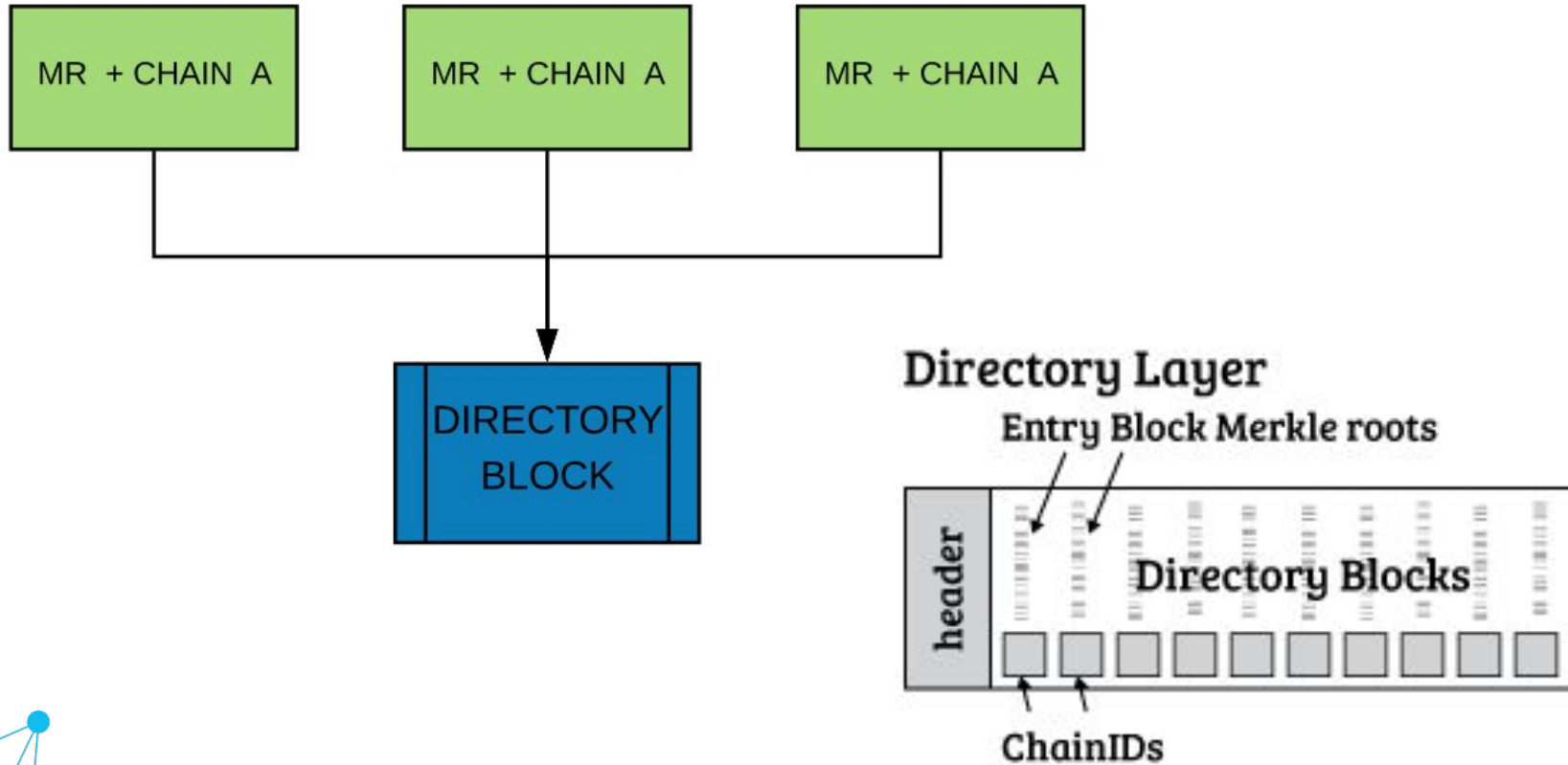
Header	
32 bytes	Admin ChainID
32 bytes	BackReferenceHash
4 bytes	DB Height
varInt_F	Header Expansion Size

Variable	Header Expansion Area
4 bytes	Message Count
4 bytes	Body Size
Body	
variable	All objects

Directory Layer - Organizes the Merkle Roots of Entry Blocks

- + A Directory Block consists of a header and a body.
- + The body is a series of pairs of ChainIDs and Entry Block Merkle Roots.

Directory Block



Directory block

HEIGHT

160169

START TIME (UTC+0700)

Thursday, September 27, 2018, 20:15

NEXT DIRECTORY BLOCK

HEIGHT: 160170 KEYMR: 9d947b2e13255cdb4fc59b46b60e08021f92f9dcc11

KEYMR

d3ca63dd5233efa061d29eb7980ee48d056abf6db9b988c6c97a3f7a2c4317bb

PREVIOUS DIRECTORY BLOCK

HEIGHT: 160168 KEYMR: d7e227c977116e9021288b3c84a0a239d3105f14416

Blocks ⑥

(included in this directory block)

ADMIN BLOCK 1d16ca20b7d598c4fe08c8ece7ed31fb9781b6c3e0c6ad99ccaf592c9a92f8ba

ENTRY CREDIT BLOCK 3058c8e4e8541be1223b3bd867ead91bf6745c329bc133c7945d5fe51a16baea

FACTOID BLOCK 55e760c3856fb3e2fff32c7fa70a53838333b6e5d923a95d894cde2572decfca

ENTRY BLOCK 72381b27de5b8123ad04ea90512b1e9328dc88041de1f5a7dc24a7e1fc7c9c52

CHAIN 9005bb7dd69fb9910ee0b0db7b8a01198f03623eab6dadf1eba01f9dbc207577

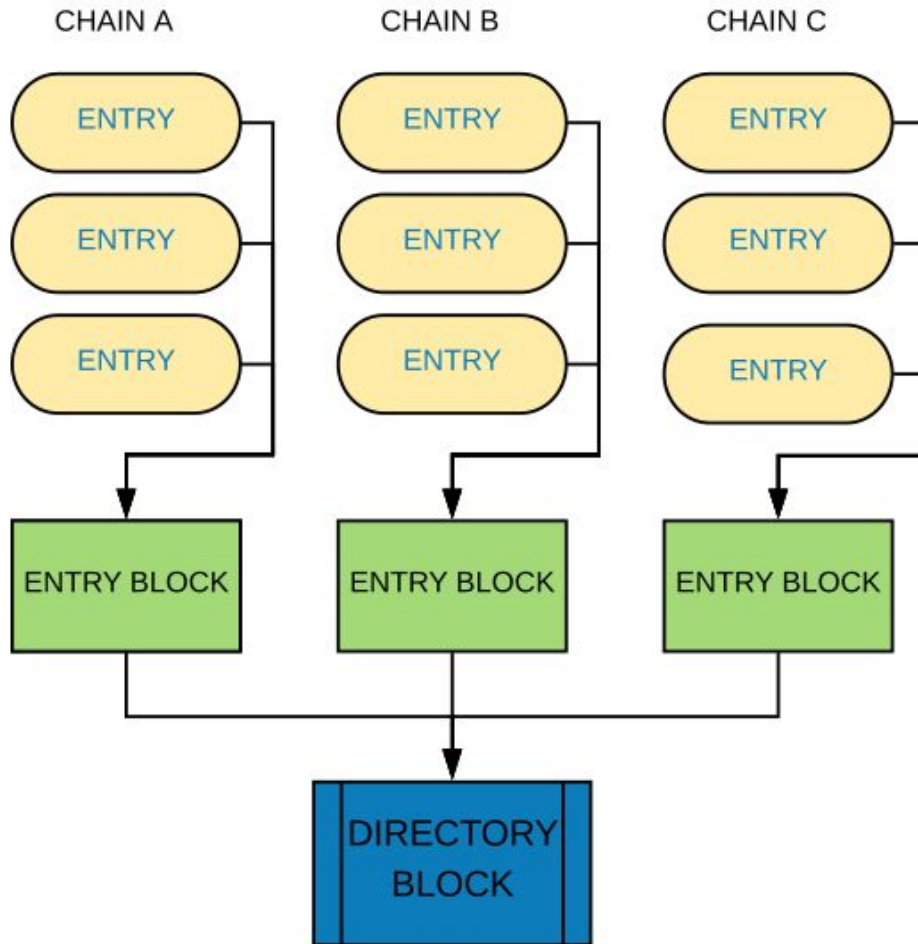
Directory Block

Header		
varInt_F	Version	Describes the protocol version that this block is made under. Only valid value is 0. Can safely be coded using 1 byte for the first 127 versions.
4 bytes	NetworkID	This is a magic number identifying the main Factom network. The value for MainNet Directory Blocks is 0xFA92E5A2. TestNet is 0xFA92E5A3.
32 bytes	BodyMR	This is the Merkle root of the body data which accompanies this block. It is calculated with SHA256.
32 bytes	PrevKeyMR	Key Merkle root of previous block. It is the value which is used as a key into databases holding the Directory Block. It is calculated with SHA256.
32 bytes	PrevFullHash	This is a SHA256 checksum of the previous Directory Block. It is calculated by hashing the serialized block from the beginning of the header through the end of the body. It is included to allow simplified client verification without building a Merkle tree and to doublecheck the previous block if SHA2 is weakened in the future.
4 bytes	Timestamp	This the time when the block is opened. Blocks start on 10 minute marks based on UTC (ie 12:00, 12:10, 12:20). The data in this field is POSIX time, counting the number of minutes since epoch in 1970.
4 bytes	DB Height	The Directory Block height is the sequence it appears in the blockchain. Starts at zero.
4 bytes	Block Count	This is the number of Entry Blocks that were updated in this block. It is a count of the ChainID:Key pairs. Inclusive of the special blocks. Big endian.

Directory Block

Body		
32 bytes	Admin Block ChainID	Indication the next item is the serial hash of the Admin Block.
32 bytes	Admin Block LookupHash	This is the LookupHash of the Admin Block generated during this time period.
32 bytes	Entry Credit Block ChainID	Indication the next item belongs to the Entry Credit Block.
32 bytes	Entry Credit Block HeaderHash	This is the serial hash of the Entry Credit Block Header generated during this time period.
32 bytes	Factoid Block ChainID	Indication the next item belongs to the Factoid Block.
32 bytes	Factoid Block KeyMR	This is the KeyMR of the Factoid Block generated during this time period.
32 bytes	ChainID 0	This is the ChainID of one Entry Block which was updated during this block time. These ChainID:KeyMR pairs are sorted numerically based on the ChainID.
32 bytes	KeyMR 0	This is the Key Merkle Root of the Entry Block with ChainID 0 which was created during this Directory Block.
32 bytes	ChainID N	Nth Entry Block ChainID.
32 bytes	KeyMR N	Nth Entry Block KeyMR.

Complete system





Installing Factom Environment



Install Factom Environment

Install Factom Federation

INTRODUCTION

- + Factom Federation (FF) is available for Mac, Windows, Linux and Docker.
- + Step by step guide on how to install FF binaries on Mac, Windows, and Linux.

Step 1 Download the installer for Mac, Windows, or Linux on [GitHub](#). There are various versions, make sure to select the one best suited for your system.

Step 2 Save it to your desktop or downloads folder on your local hard drive.

Step 3 Follow the instructions for your OS.

LINUX INSTALLATION

- + This installer is for commands *factomd*, *factom-walletd* and *factom-cli*.
- + The three binaries will be installed on your local drive together with a *.factom* folder in the root of the local user Home Folder “*~/.factom*”.
- + Download the “**factom-amd64.deb**” or “**factom-i386.deb**” installer that suits your system as per Step 1 then run the following command to install.

```
sudo dpkg -i ./factom-amd64.deb
```

```
sudo dpkg -i ./factom-i386.deb
```

LINUX INSTALLATION

OS	Factomd Installer
Windows 64bit	FactomInstall-amd64.msi
Windows 32bit	FactomInstall-i386.msi
Mac	Please install from source
Linux (Ubuntu/Debian) 64bit	factom-amd64.deb
Linux (Ubuntu/Debian) 32bit	factom-i386.deb
Linux (Redhat/Centos)	Please install from source

Manual download

1. Go to Github Repository here
<https://github.com/FactomProject/distribution#factom-command-line-interface-programs>
2. Download **factom-amd64.deb** for Ubuntu 64bit

INSTALL FACTOM FEDERATION

LINUX INSTALLATION

You also can run by the command using *wget*

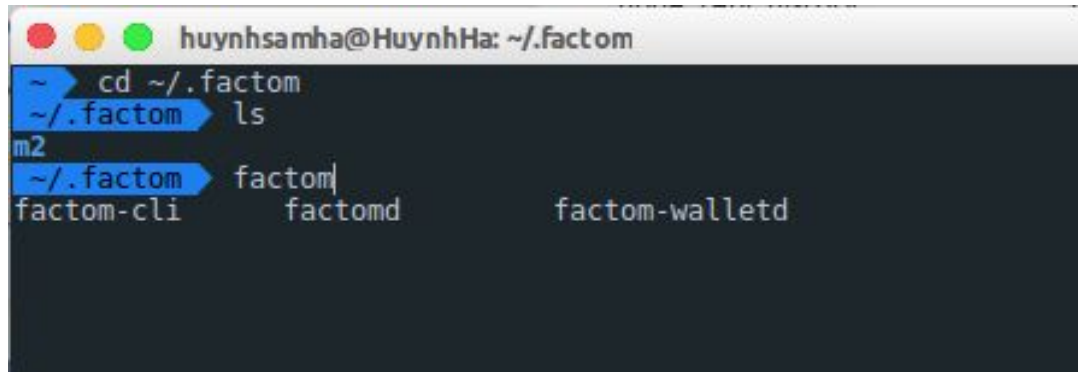
“*wget*

`https://github.com/FactomProject/distribution/releases/download/v5.4.2/factom-amd64.deb`”

LINUX INSTALLATION

Run “`sudo dpkg -i ./factom-amd64.deb`”

Open terminal and 3 factom commands are available, including **factom-cli**, **factomd** and **factom-walletd**



```
huynhsamha@HuynhHa: ~/.factom
~$ cd ~/.factom
~/.factom$ ls
factom  factom-cli  factomd  factom-walletd
```


FACTOM COMMANDS

- **factomd** is the main program. It manages the blockchain, connects to the public network, and enforces the network rules.
- **factom-walletd** is an application for holding private keys. It builds Factoid transactions and handles crypto related operations to add user data into Factom.
- **factom-cli** is a program for users to interface with *factomd* and *factom-walletd*. It may be used to create Chains, Entries, and Factoid transactions.



Install Factom Environment

Factom Control Panel?

INTRODUCTION

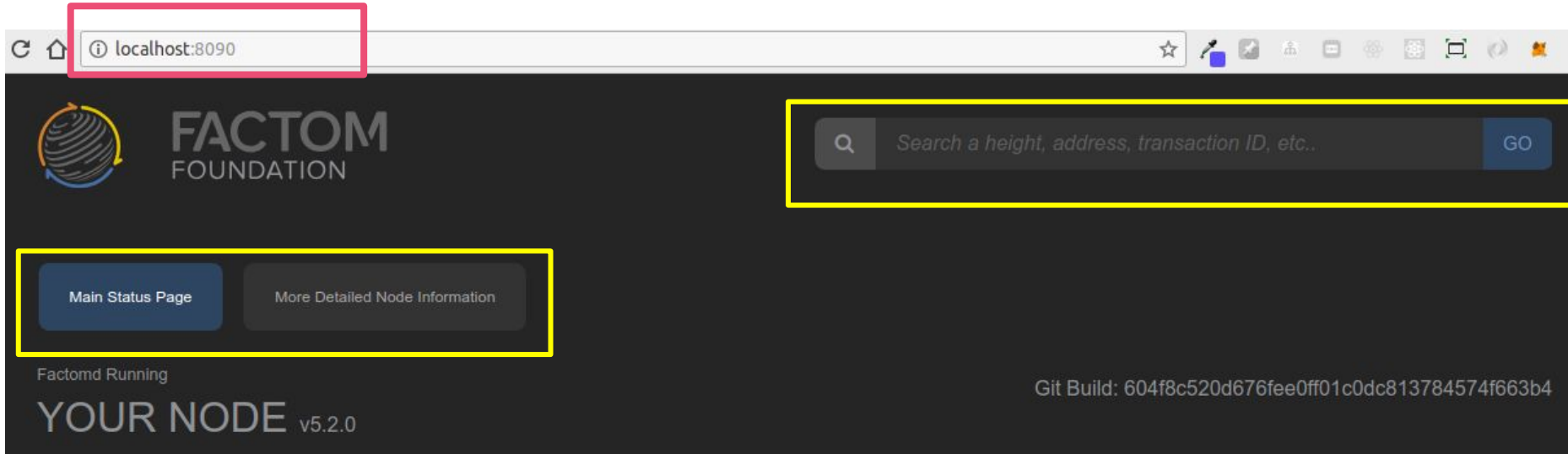
- + The Factomd Control Panel (FCP) is graphical interface for a factomd node.
- + It presents a lot of useful information such as the state of the node, blockchain sync status, connected federated servers, etc.
- + It also allows you to search for transactions' IDs, block height, factoid and entry credit addresses, or other data points similar to using the Factom Explorer.
- + A very useful tool to have for any Factom user or developer who wants to keep an eye on what's going on when running factomd on their local machines.

INSTALLING AND USAGE

- + After installing Factom Federation (FF), you can access Factomd Control Panel without installation.
- + The Control Panel is accessible after running “**factomd**” via command line at the following URL: <http://localhost:8090>.
- + The Control Panel has two main windows:
 - + The Main Status Page
 - + The More Detailed Node Information Page

FACTOM CONTROL PANEL

MAIN STATUS PAGE



FACTOM CONTROL PANEL

MAIN STATUS PAGE

Main Status Page

More Detailed Node Information

Factomd Running

YOUR NODE v5.2.0

Git Build: 604f8c520d676fee0ff01c0dc813784574f663b4

Your Block Height:

51144

Node Sync Status (1st pass):

32% Synced (51144 of 157121)

Node Sync Status (2nd pass):

3% Synced (4889 of 157121)

Federated

1

Audit

0

18 Connections:

IP	STATUS	DURATION	SENT	RECEIVED	ACTIONS
23.111.178.50	Online	8 mins	150(1.1 MB)	410(2.2 MB)	DISCONNECT
23.20.21.45	Online	8 mins	118(61.2 KB)	253(1.3 MB)	DISCONNECT
18.223.95.3	Online	8 mins	171(1.1 MB)	378(1.8 MB)	DISCONNECT
18.214.236.26	Online	8 mins	126(1.1 MB)	311(1.9 MB)	DISCONNECT
18.206.2.235	Online	8 mins	106(59.9 KB)	249(1.3 MB)	DISCONNECT
18.197.200.61	Online	8 mins	114(47.5 KB)	235(1.3 MB)	DISCONNECT
MSGs(DATA)			2.4 K(10.5 MB)	6.1 K(28.7 MB)	

MAIN STATUS PAGE

TRANSACTIONS & ENTRIES

Factoids(100)

Chain/Entry Commits (100)

Last Directory Block - 51664

TRANSACTION ID

TOTAL INPUT

INPUT
ADDRESSES# OUTPUT
ADDRESSES

fa8c0d3eb76d0a0bbc25e3e7edda3ff1467f9d8392426deaccce11d4cbe132

0.110200

1

1

a14068a9fc09ca99c642dea5ec0245b69646541661c9c39438903c10a3b29f5a

99.510200

1

1

6dab59c833462a188dab80df7648e930b50be46c38ec52a8d9f6f2fd1b13f5dc

8.510200

1

1

68ee9f3c118c0f13b6d7479ffd3125d33cff49fc53ac7f776eff264153050624

199.010200

1

1

2ed11d2508fcc310b36c8c35df2d14ef97baf02668d9cecc20077d0a0fa99c206

494.127200

1

1

450a0277349b98b73899418914a65fce3bf9807cf0cb3fdc183513f5e40c3dd3

14.606800

1

1

eed0581e09ba27421f7acff748de1ede0eb293af60916cb873cb4c86c68a62ed

466.665200

1

1

a459150bfae9faad10cf61aca238c43f2e25a1f9e633c4ef75becc1b38bf5a14

298.510200

1

1

MAIN STATUS PAGE

Factoids(100)

Chain/Entry Commits (100)

Last Directory Block - 51823

ENTRY HASH

CHAIN ID

EC COST

ca2c6920a88c2caf052c876eb59f5c76b34cfedd7468bce651d4adf67ee15bab

MAIN STATUS PAGE

TRANSACTIONS & ENTRIES

Factoids(100)

Chain/Entry Commits (100)

Last Directory Block - 52078

LAST DIRECTORY BLOCK INFORMATION

KeyMR:	f498edba1f7540fe98bc1173f55550d8e99418ad87e52462318454a6d9fedf5a
Body KeyMR:	4c48d69f38b4af4701f15f19f7f261c48c00c529a4e6e68754e853e6a8958478
Full Hash:	d7ee069a9de17e2c5e94290c6cad7f32cd4bc2c6c534c0604aa41d4c50544090
Timestamp:	2016-08-29 14:50:00
Block Height:	52078

MORE DETAIL NODE INFORMATION PAGE

Main Status Page More Detailed Node Information

TERMINAL DATA IMPORT


Summary Process List PrintMap Servers Elections Connections

Raw Syncing

```
===SummaryStart===
  FNode0[455b7b]  _____  0/ 0  0.0%  0.000 52929[058116] 52927/52930/157124  -/ 0      69/0/0/2126
152/0/271/0      0      0      0/0/0      0/0/0  0.00/0.00 52930-:-0 - 73c047


                                0
                                Review  0
                                Holding  47
                                Acks     0

                                MsgQueue  0
                                InMsgQueue 0
                                APIQueue  0
                                AckQueue  0
```



50

SEARCH HEIGHT, ADDRESS, TRANSACTION, ETC

 **FACTOM**
FOUNDATION

DIRECTORY BLOCK

KeyMR:	8a7611f9d49ed89b8ff7849f5e11b1bc1945bc5cce080253a4a2ef2bd87899a8
Body KeyMR:	b085b785be2eead177a4902a46b0a2dc29843ccdee58fcc60bd422fc010f3b25
Full Hash:	2c16d35c03d897df24285e47dc32238204dd454eb56f4bbb646d0bbd3f7112a0
TimeStamp:	2015-09-02 06:10:00
Block Height:	20
Previous Directory Block:	Previous KeyMR : 5e1d504123f8d743b53ef574a48fe73d3ab96a4d921cd98bb6aea6a641233608 Previous Full Hash : d69f6ad1f4f846637d195c48c653ce3e0dc8bf5f7efdbad16a33b1f429898f91

Blocks Contained in Block 4 Blocks

Type:	Admin Block
KeyMR:	f9f7fcee7afade8d462896c8e976c0f56fc52de1a8eb8711522768e189db75fa



Install Factom Environment

Install Enterprise Wallet

INTRODUCTION

- + Desktop Application for Windows, Mac, or Linux, uses the same file locations as the binary version (*factom-walletd*)
- + Step by step guide on how to install Enterprise Wallet on Mac, Windows, and Linux.

Step 1 Download the installer for Mac, Windows, or Linux on [GitHub](#).

- **Mac:** *enterprise-wallet-setup.dmg*
- **Windows 64bit:** *enterprise-wallet-setup-amd64.exe*
- **Linux (Ubuntu/Debian) 64bit:** *enterprise-wallet-setup-amd64.deb*
- **Linux (Redhat/Centos):** *enterprise-wallet-linux.zip*

Step 2 Save it to your desktop or downloads folder on your local hard drive.

Step 3 Follow the instructions for your OS.

LINUX INSTALLATION

1. Manual download [enterprise-wallet-setup-amd64.deb](https://github.com/FactomProject/distribution/releases/download/v0.4.2.11/enterprise-wallet-setup-amd64.deb) on Github.

Or use the following command for Ubuntu 64bit

“wget

<https://github.com/FactomProject/distribution/releases/download/v0.4.2.11/enterprise-wallet-setup-amd64.deb>”

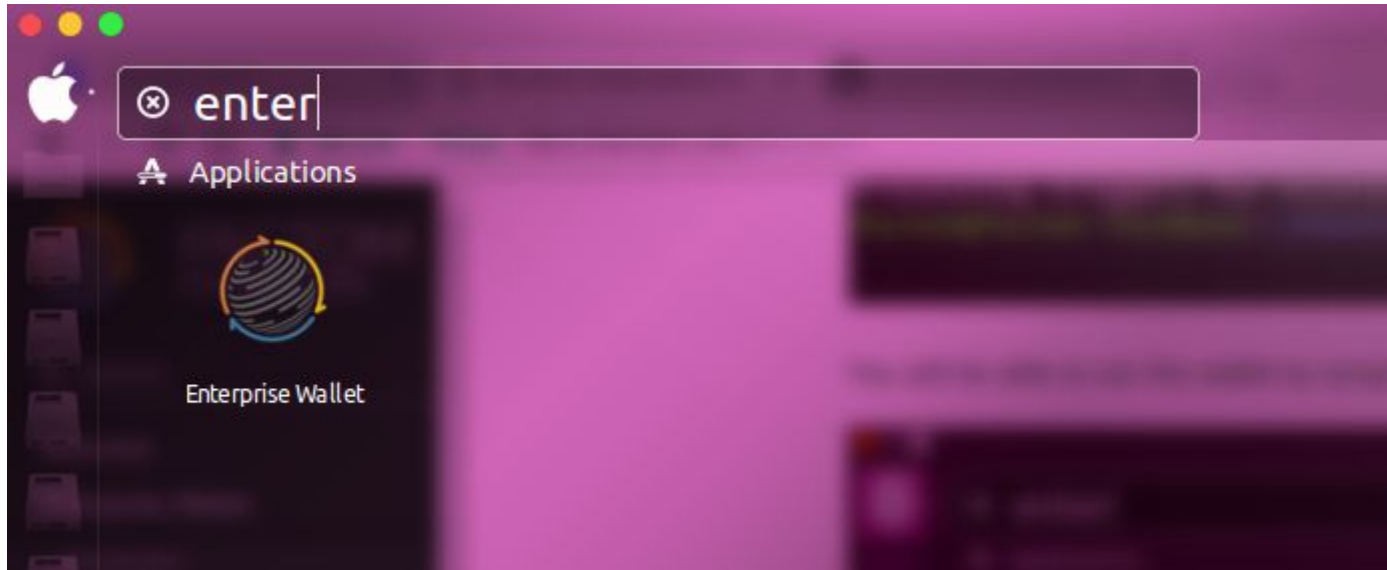
2. Run command for installing:

```
sudo dpkg -i enterprise-wallet-setup-amd64.deb
```

INSTALL ENTERPRISE WALLET

LINUX INSTALLATION

3. Open Enterprise Wallet





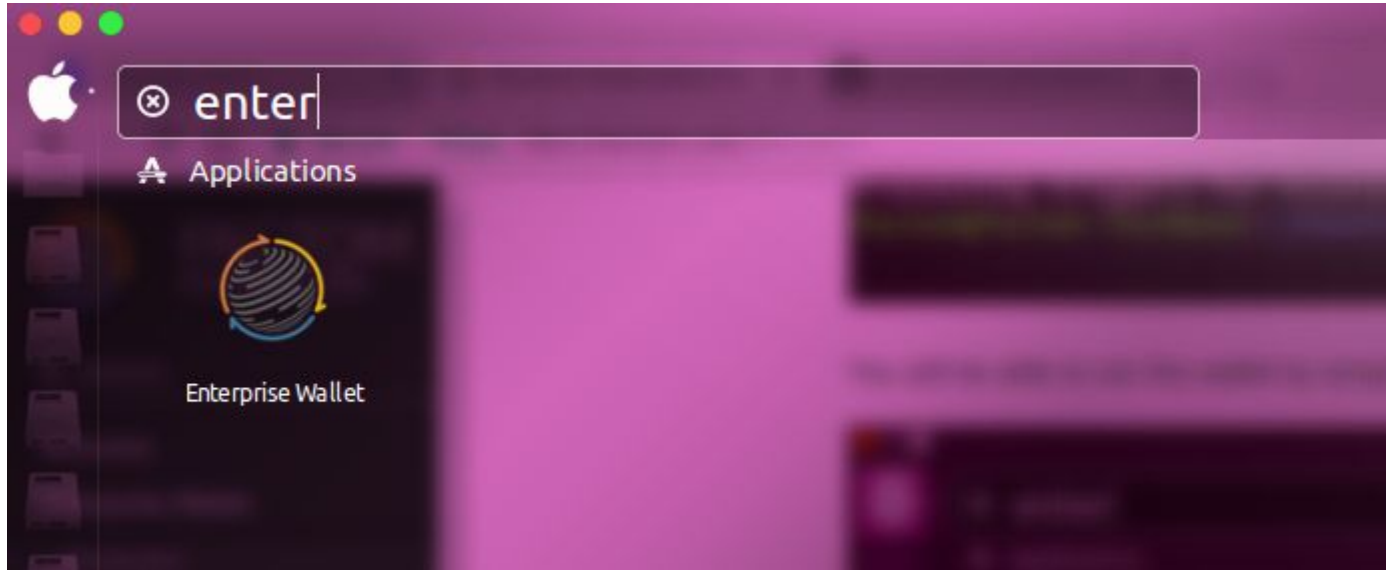
How to use Enterprise Wallet?

How to use Enterprise Wallet?

Choose Your Wallet Type

CHOOSE YOUR WALLET TYPE

OPEN ENTERPRISE WALLET



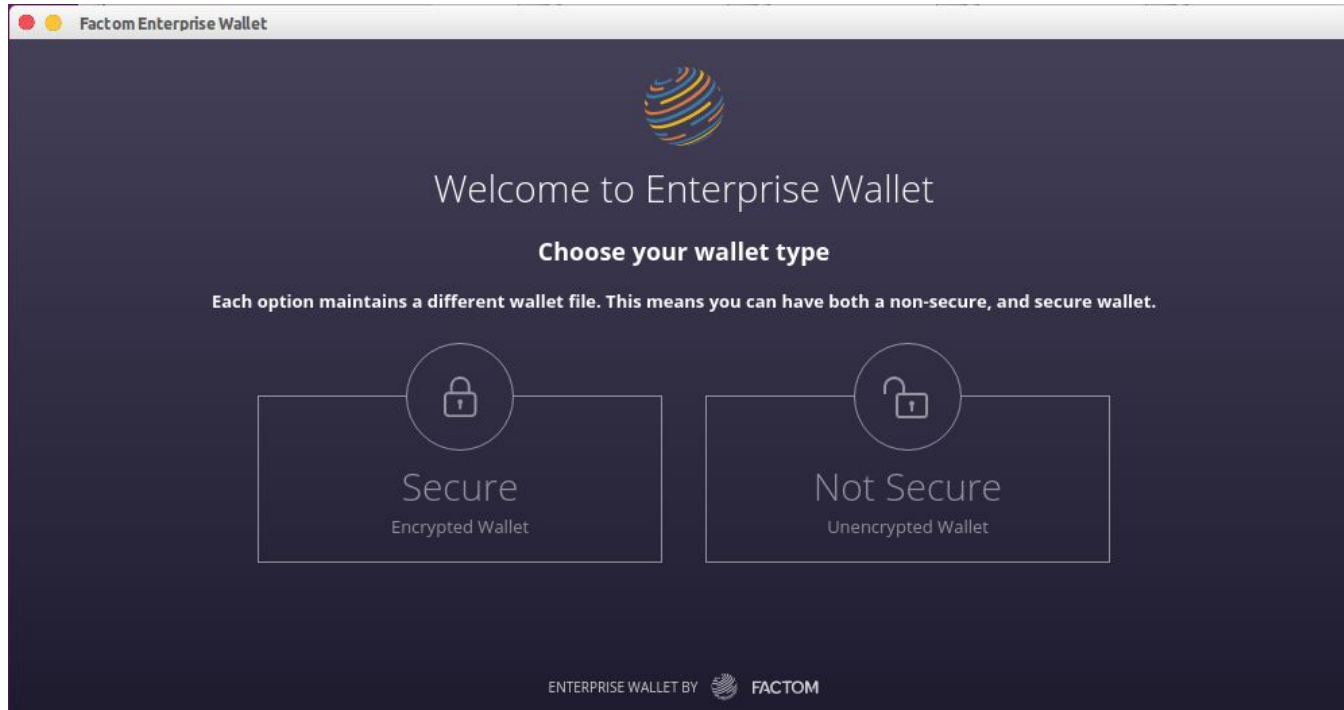
CHOOSE YOUR WALLET TYPE

CHOOSE WALLET TYPE

- + When you launch Enterprise Wallet you have the option to choose a “*secure*” or a “*not secure*” wallet, the former is encrypted the latter is unencrypted.
- + To open either one, just close and relaunch Enterprise Wallet, selecting the one you wish to open.

CHOOSE YOUR WALLET TYPE



CHOOSE WALLET TYPE




CHOOSE YOUR WALLET TYPE

SECURE TYPE

Factom Enterprise Wallet



Secure
Encrypted Wallet

 Enterprise Wallet

Opening a secure wallet will encrypt all private keys and the seed.

Keep in mind this means that **if you forget your password and seed, there is no recovery option.**

Be careful selecting this option.

When using an encrypted wallet, be sure to immediately back up your seed. This allows you to regenerate your wallet if you lose the password or the database gets corrupted.

This option will not convert a v1 wallet if found on disk. This is intentional, as converting any wallet to an encrypted wallet means the original wallet still exists, and still insecure. Addresses and seeds can still be backed up and restored.


For directions on how to transfer your Factoids to a secure wallet, [read our blog post here.](#)

Input the password used to encrypt your wallet.

.....

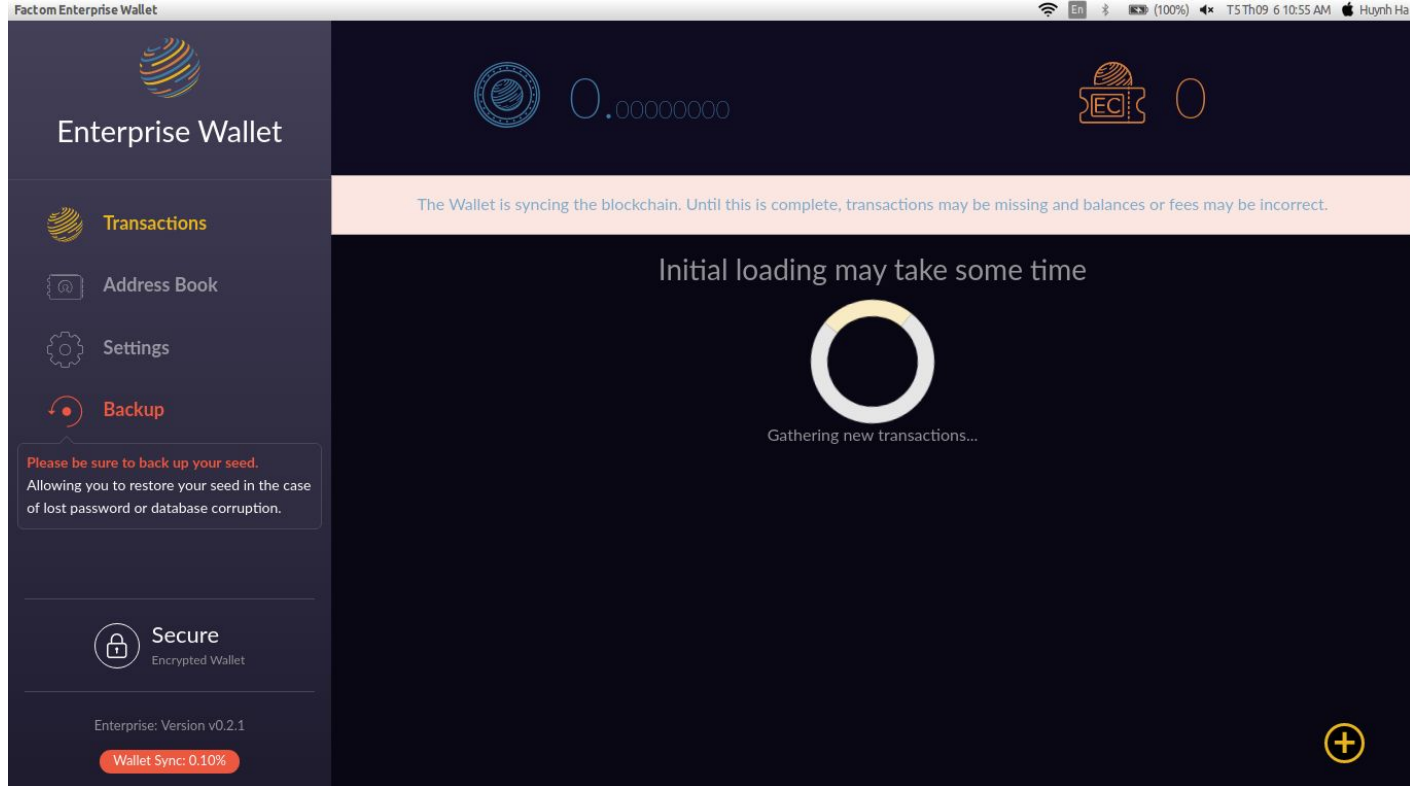
☒ I acknowledge that if I lose this password and my seed I will lose access to the wallet and its funds.

Proceed to Encrypted Wallet

ENTERPRISE WALLET BY  FACTOM

CHOOSE YOUR WALLET TYPE

SECURE TYPE

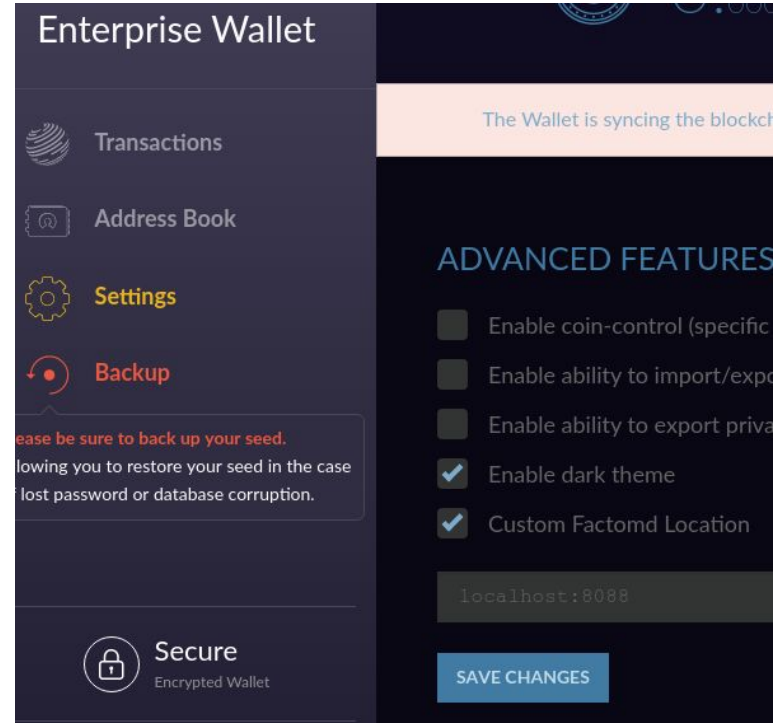


How to use Enterprise Wallet?

Enterprise Wallet Mode

RUN ONLINE

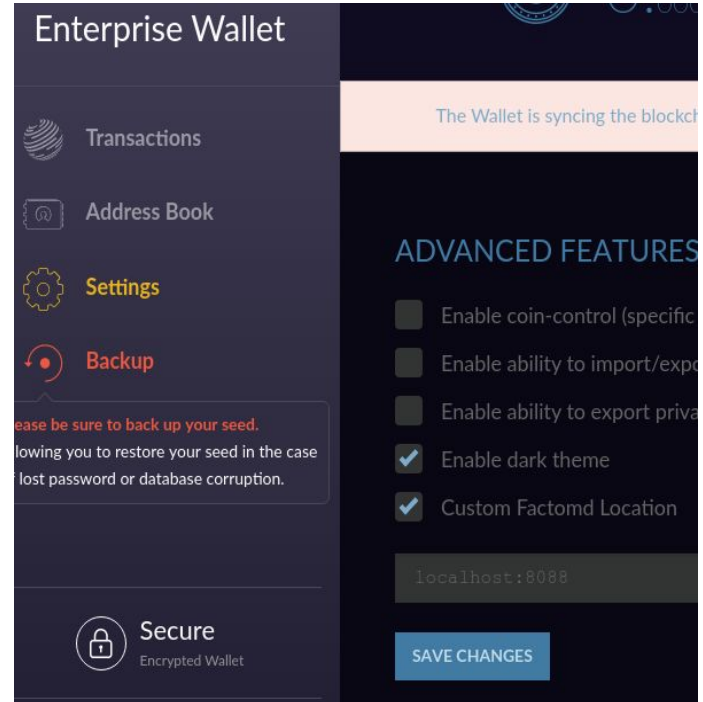
1. Go to “Settings.”
2. Now click “Custom Factomd Location” and enter in a valid factomd instance. Use: **courtesy-node.factom.com**.
3. Then click “Save Changes.”
4. You are now using an online version of factomd! No need to sync the whole blockchain, just the transactions.



ENTERPRISE WALLET MODE

RUN LOCALLY

1. Go to “Settings.”
2. Now click “Custom Factomd Location” and enter in a valid factomd instance. Use: **localhost:8088**.
3. Then click “Save Changes.”
4. Open terminal and Start your factom by run: ***“factomd”***
5. Now your Enterprise Wallet is syncing with your local database



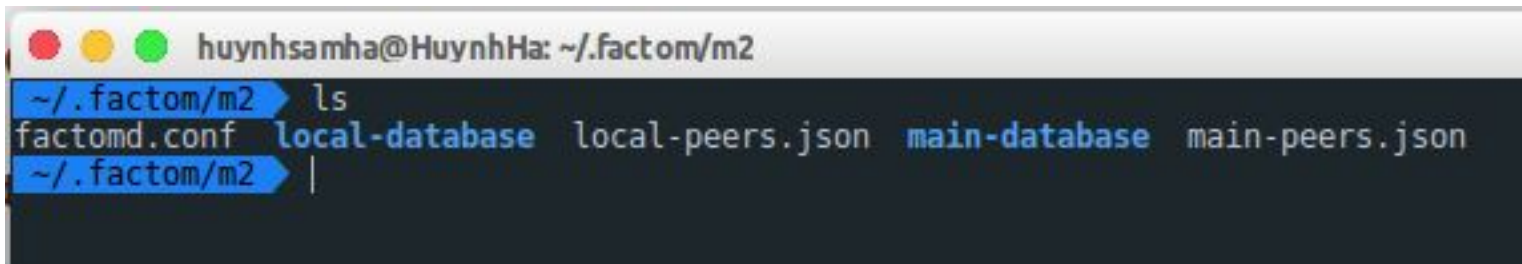
CHANGE NETWORK TO LOCAL

By default, **factomd** will run on **MAIN** net, that will sync your database on local machine with MAIN net, about 12GB.

To change to **LOCAL** net, run **factomd** command with param **-network=LOCAL**.

factomd -network=LOCAL

Now your local run on database at local, which not sync with MAIN net.

A terminal window with a title bar showing 'huynhsamha@HuynhHa: ~/.factom/m2'. The terminal has a dark background with blue and white text. The user has entered 'ls' and the output shows 'factomd.conf', 'local-database', 'local-peers.json', 'main-database', and 'main-peers.json'.

```
huynhsamha@HuynhHa: ~/.factom/m2  
~/.factom/m2 ➔ ls  
factomd.conf  local-database  local-peers.json  main-database  main-peers.json  
~/.factom/m2 ➔ |
```

How to use Enterprise Wallet?

How to use the features?

HOW TO USE THE FEATURES?

CREATE A FACTOID ADDRESS

NEW ADDRESS

Generate address from:

Random new factoid address



Private key:

An new factoid address will be created

Nickname:

FA_001

ADD TO ADDRESS BOOK

HOW TO USE THE FEATURES?

CREATE AN ENTRY CREDIT ADDRESS

NEW ADDRESS

Generate address from:

Random new entry credit address



Private key:

A new entry credit address will be created

Nickname:

ECA_001

ADD TO ADDRESS BOOK

HOW TO USE THE FEATURES?

IMPORT FROM PRIVATE KEY

In local, to use Factoids, please use with private key

Fs3E9gV6DXsYzf7Fqx1fVBQPQXV695eP3k5XbmHEZVRLkMdD9qCK

The screenshot shows a dark-themed interface for creating a new address. The title 'NEW ADDRESS' is at the top left. Below it, the 'Generate address from:' section has a dropdown menu with 'Import from private key' selected. To the right of this dropdown is a blue button with a white downward arrow. Below the dropdown, the 'Private key:' field contains the text 'Fs3E9gV6DXsYzf7Fqx1fVBQPQXV695eP3k5XbmHEZVRLkMdD9qCK'. To the right of the private key field is a 'Nickname:' field containing 'FA_100'. At the bottom right, there is a blue button labeled 'ADD TO ADDRESS BOOK'.

NEW ADDRESS

Generate address from:

Import from private key

Private key:


Nickname:


FA_100

ADD TO ADDRESS BOOK

HOW TO USE THE FEATURES?

SEND AND RECEIVE FCT

 20000.00000000

 0

SELECT ADDRESS

- ☐ (0.00000000 FCT) FA_001
- ☒ (0.00000000 FCT) FA_002
- ☐ (0.00000000 FCT) FA_003
- ☐ (0.00000000 FCT) FA_004
- ☐ (0.00000000 FCT) FA_005
- ☐ (20000.00000000 FCT) FA_100

ADD ADDRESS

100 FCT +

Amount of factori FCT -


Fee: FCT


Total: FCT

VIEW TRANSACTION

HOW TO USE THE FEATURES?

SEND AND RECEIVE FCT



 20000.000000000

 0

SEND FACTOIDS

Send to:

OUTPUTS

FA2LYPd6yuEtwK8jCSmb9lJsEH4qhJyYb2QlHAu4v7HzEbTLFFn2		100	FCT	+
FA3cd7hv8FWzrgbCNqQYEhdanDWqD8Ato3hrXFYNswmzQMDqY4Cb		200	FCT	-

Fee:

FCT



Total:

FCT

VIEW TRANSACTION

HOW TO USE THE FEATURES?



SEND AND RECEIVE FCT

 20000.000000000 0

SEND FACTOIDS

Send to:

OUTPUTS

FA2LYPd6yuEtwK8jCSmb91JsEH4qhJyYb2Q1HAu4v7HzEbTLPfN2		100	FCT	+
FA3cd7hv8FWzrgbCNqQYEHdanDWqD8Ato3hrXFYNswmzQMDqY4Cb		200	FCT	-

Fee:

0.00022

FCT

SEND TRANSACTION

EDIT TRANSACTION

Total:

300.00022






FCT

Click "Send Transaction" to send, or go back to editing it

VIEW TRANSACTION

HOW TO USE THE FEATURES?

SEND AND RECEIVE FCT

	19999.99978000		0
	09/08/2018 : Sent FA_100 (FA2jK2HcLnRdS94dEcU27rF3meoJfpUcZPSinpb7AwQvPRY6RL1Q)	-300.0002 FCT	
	09/08/2018 : Received FA_001 (FA2LYPd6yuEtwK8jCSmb91JsEH4qhJyYb2Q1HAu4v7HzEbTLPfN2) FA_002 (FA3cd7hv8FWzrgbCNqQYEhdanDWqD8Ato3hrXFYNswmzQMDqY4Cb)	300 FCT	
	09/01/2015 : Received FA_100 (FA2jK2HcLnRdS94dEcU27rF3meoJfpUcZPSinpb7AwQvPRY6RL1Q)	20000 FCT	

HOW TO USE THE FEATURES?

SEND AND RECEIVE FCT

On Factom Control Panel

FACTOID TRANSACTION

Inputs

FA2jK2HcLnRdS94dEcU27rF3meoJfpUcZPSinpb7AwQvPRY6RL1Q

Outputs

FA2LYPd6yuEtwK8jCSmb91JsEH4qhJyYb2Q1HAu4v7HzEbTLPfN2
FA3cd7hv8FWzrgbCNqQYEhdanDWqD8Ato3hrXFYNswmzQMDqY4Cb

Transaction ID (txid): 40bf397e0977a471236c40ae31a9c9e81c8369d3e90c8ec97152bb9ffd127fef

Transaction Hash: 5107b99f946c9676f68976ce2181f67a2d1aa4fc924b411ae883aaba96d3716f

Total Input: 300.00022000 Factoids

Total Factoid Output: 300.00000000 Factoids

Total EC Output: 0.00000000 Factoids

Transaction Status: DBlockConfirmed



HOW TO USE THE FEATURES?

CONVERT FCT TO ENTRY CREDIT

BUY ENTRY CREDITS

Send to:

OUTPUTS

EC3dhgwyDBVd2jecS5in5qp5uuhehVzRz9tvkxxvVmd9o8hqG2vk		100	EC	+
EC34e6FTeVHShAZbeYqbSs5l3w74iHLhWLdcZaYZ3jYBy5rK2Gvw		200	EC	-

Fee:

0.00022

FCT

SEND TRANSACTION

EDIT TRANSACTION

Total:

0.00322

FCT

Click "Send Transaction" to send, or go back to editing it

VIEW TRANSACTION

HOW TO USE THE FEATURES?

CONVERT FCT TO ENTRY CREDIT



19999.99656000



300



09/08/2018 : Sent

FA_100 (FA2jK2HcLnRdS94dEcU27rF3meoJfpUcZPSinpb7AwQvPRY6RL1Q)

-0.0032 FCT



09/08/2018 : Converted

ECA_001 (EC3dhgwyDBVd2jecS5in5qp5uuhehVzRz9tvkxxvVmd9o8hqG2vk)

0.003 FCT

ECA_002 (EC34e6FTeVHShAZbeYqbSs513w74iHLhWLdcZaYZ3jYBy5rK2Gvw)

HOW TO USE THE FEATURES?

VERIFY FCT AND EC BALANCES

Factom provided 2 way to verify transactions:

- + Use Factom Control Panel (<http://localhost:8090/>)
- + Use Factom Explorer (<https://explorer.factom.org/>)

Factom APIs

| Usage

Factom APIs Usage

Overview APIs In Factom Blockchain

factomd
API

factom
walletd
API

Debug API

Overview APIs In Factom Blockchain

factomd API

- + Use factomd -network=LOCAL
- + Run on <http://localhost:8088/v2>
- + The API is designed for outside application to process transactions and interact with the Factom federated servers.

factom-walletd API

- + Use factomd-walletd
- + Run on <http://localhost:8089/v2>
- + The API is designed for managing wallet, create, factoid account, ec account, chain, transaction and entry.

All these APIs use JSON-RPC, which is a remote procedure call protocol encoded in JSON



Factom APIs Usage

Setup Environment and API Testing Tool

Install Factom Federated for Setup Environment

factomd

Run on

<http://localhost:8088>

Use APIs for transactions

factom-walletd

Run on

<http://localhost:8089>

Use APIs for chains,
entries, accounts and
transactions

Factom Control Panel

Run on

<http://localhost:8090>

Open browser for
verification

Install Postman for API Testing



Postman Makes API Development Simple



chrome web store



Postman

Offered by: www.getpostman.com

★★★★★ 9,132

[Extensions](#)

 3,871,517 users

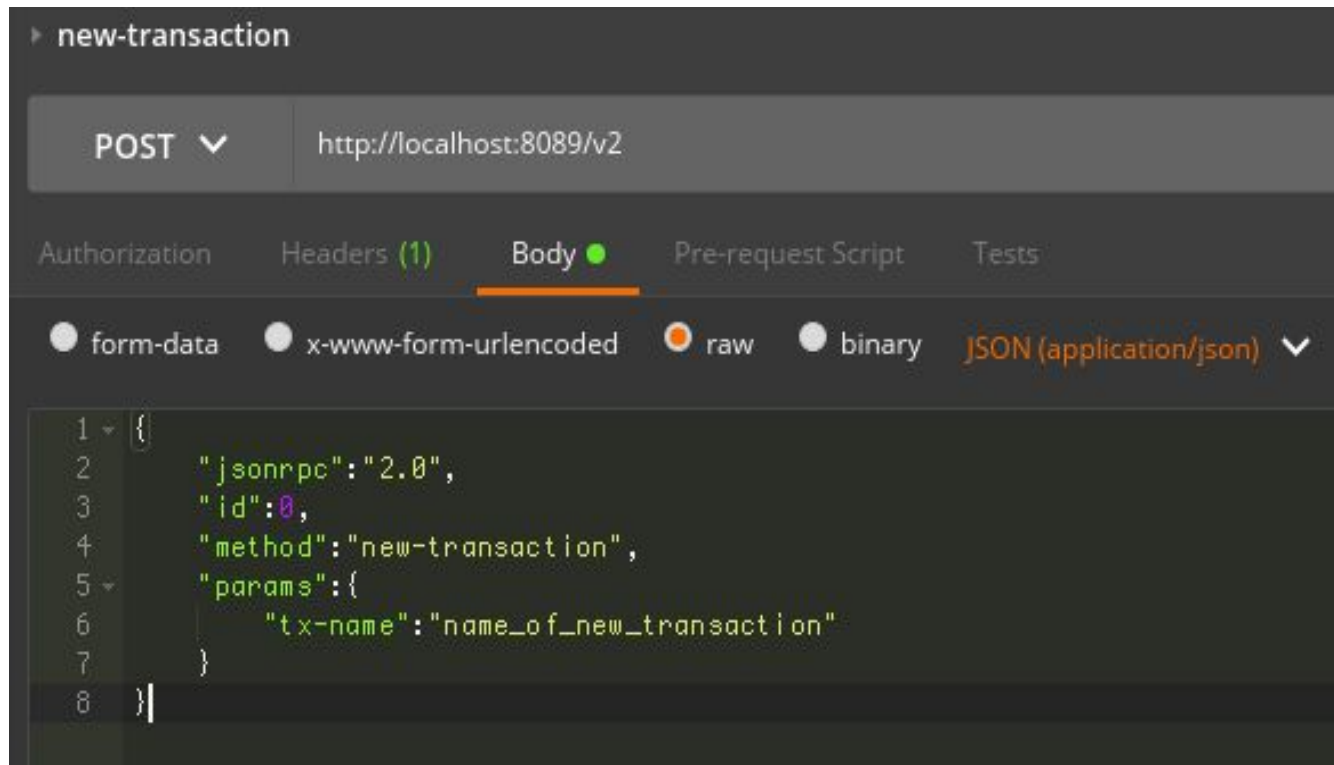
Factom APIs Usage

Quickstart APIs Testing with Postman

Go to my setup Postman for using Postman easier:

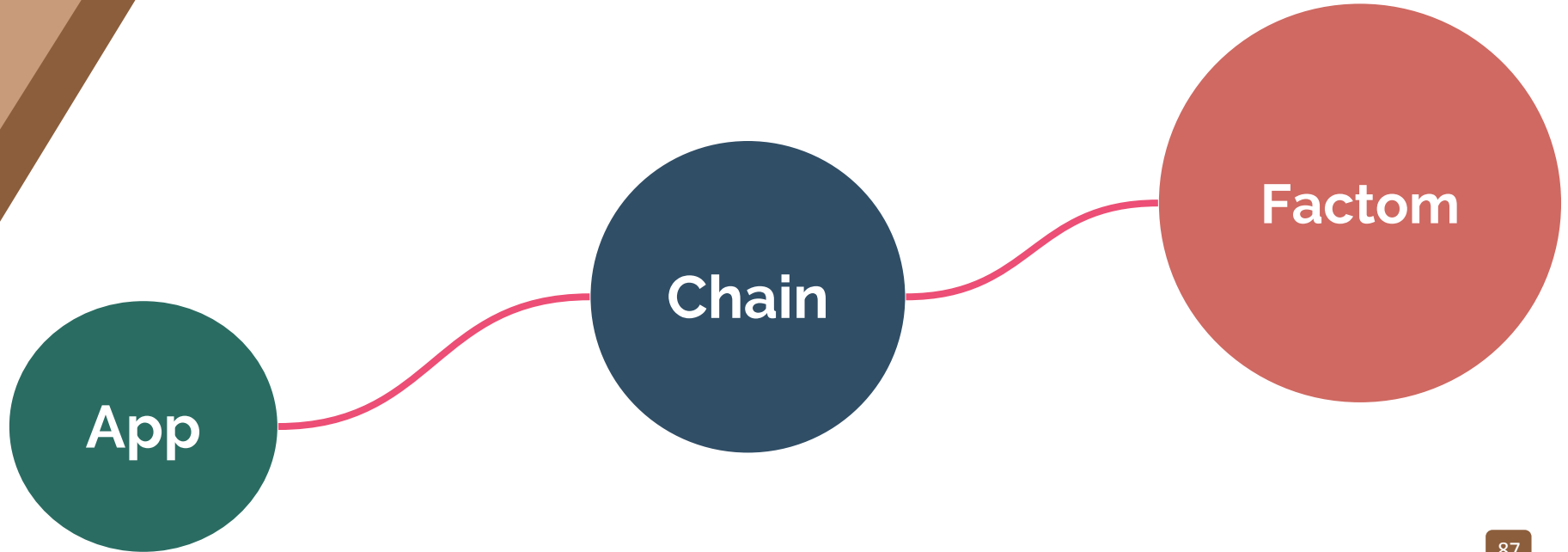
<https://documenter.getpostman.com/view/2583401>

First example creating new transaction

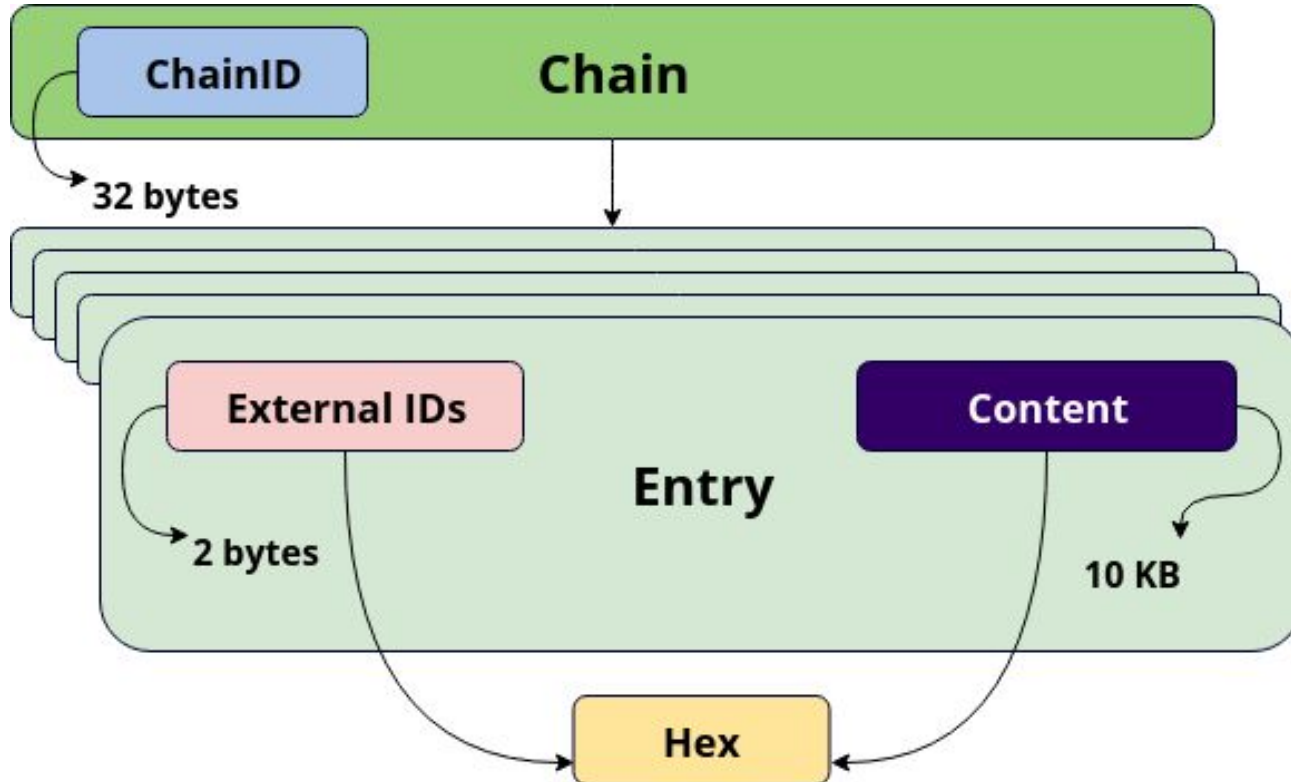


Factom APIs Usage

Life of Chain In Factom



Chain and Entries



3 steps for Creating new Chain

compose-chain

Use factom-walletd (8089)
Compose transaction
marshals the transaction
into a hex encoded string

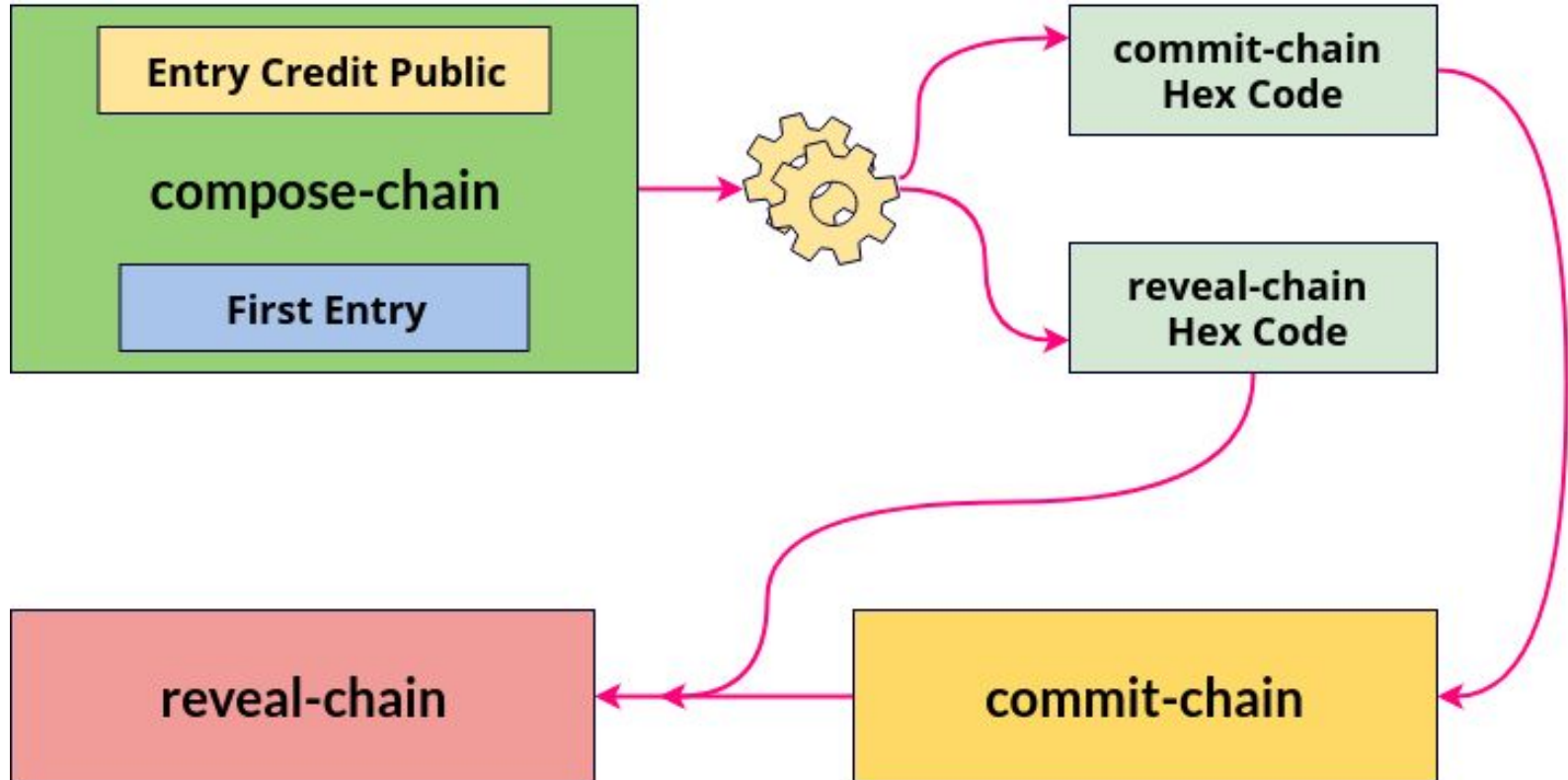
commit-chain

Use factomd (8088)
Send a Chain Commit
Message to factomd to
create a new Chain. The
commit chain hex encoded
string

reveal-chain

Use factomd (8088)
Reveal the First Entry in a
Chain to factomd after the
Commit to complete the
Chain creation. The
reveal-chain hex encoded
string

3 steps for Creating new Chain



3 steps for Creating new Entry In a Chain

compose-entry

Use factom-walletd (8089)
Compose entry into a hex encoded string for a Chain

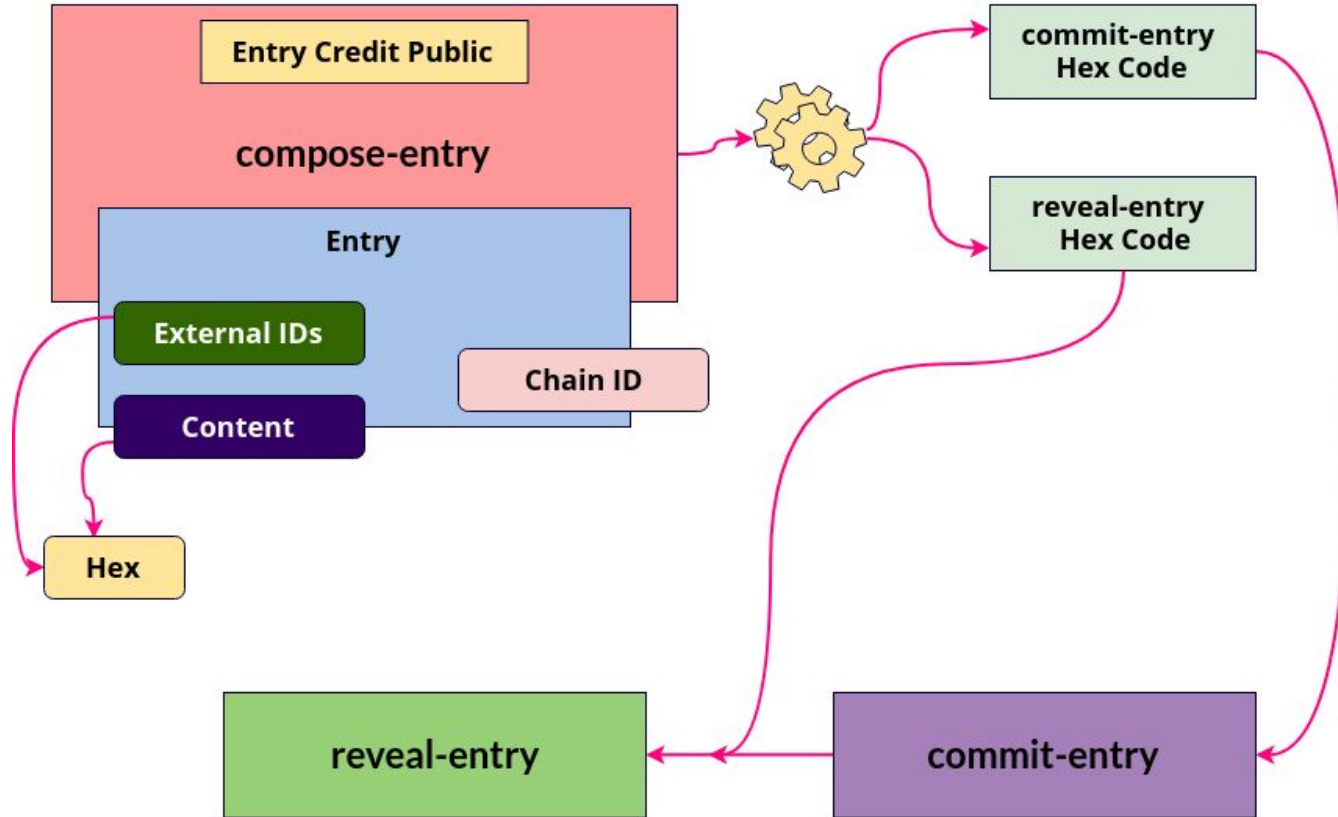
commit-entry

Use factomd (8088)
Send an Entry Commit Message to factom to create a new Entry. The entry commit hex encoded string

reveal-entry

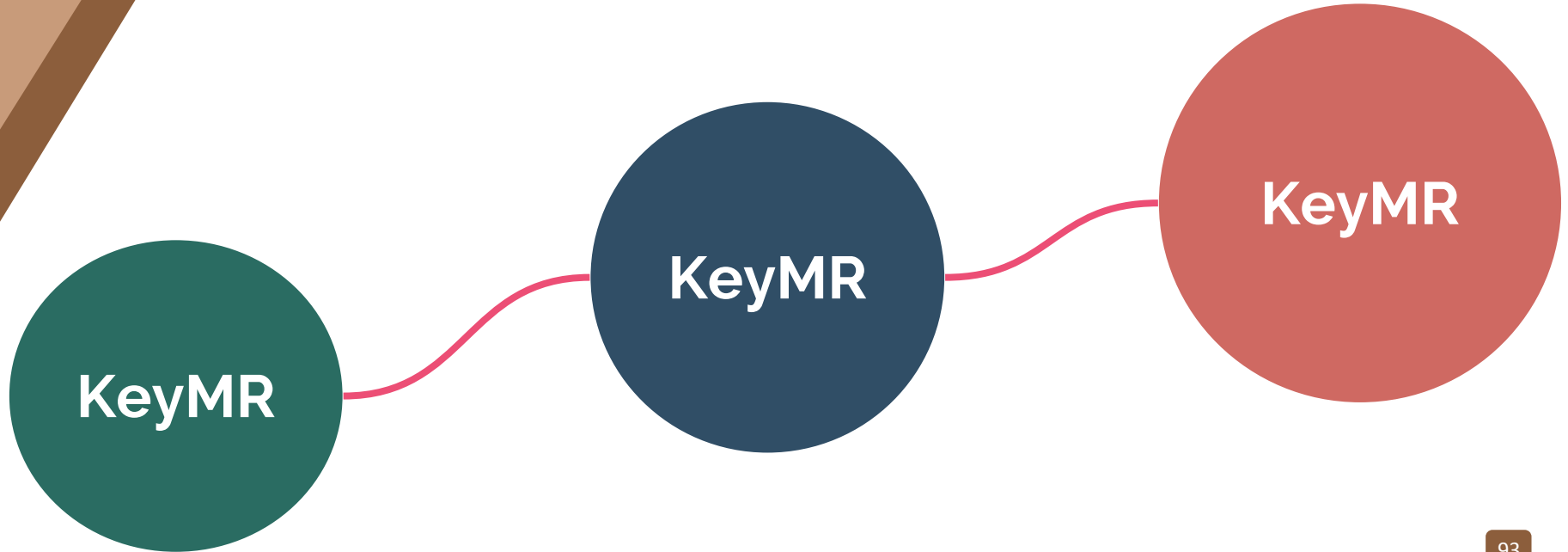
Use factomd (8088)
Reveal an Entry to factomd after the Commit to complete the Entry creation. The reveal-entry hex encoded string

3 steps for Creating new Entry In a Chain

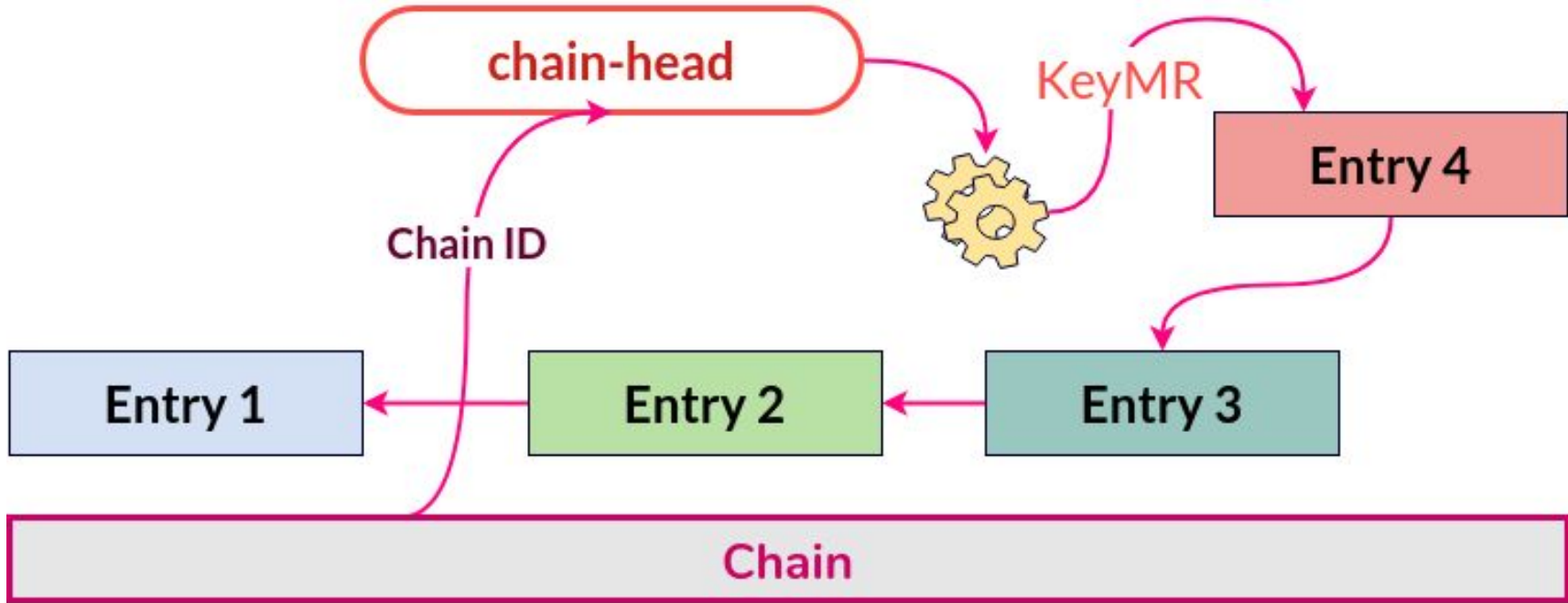


Factom APIs Usage

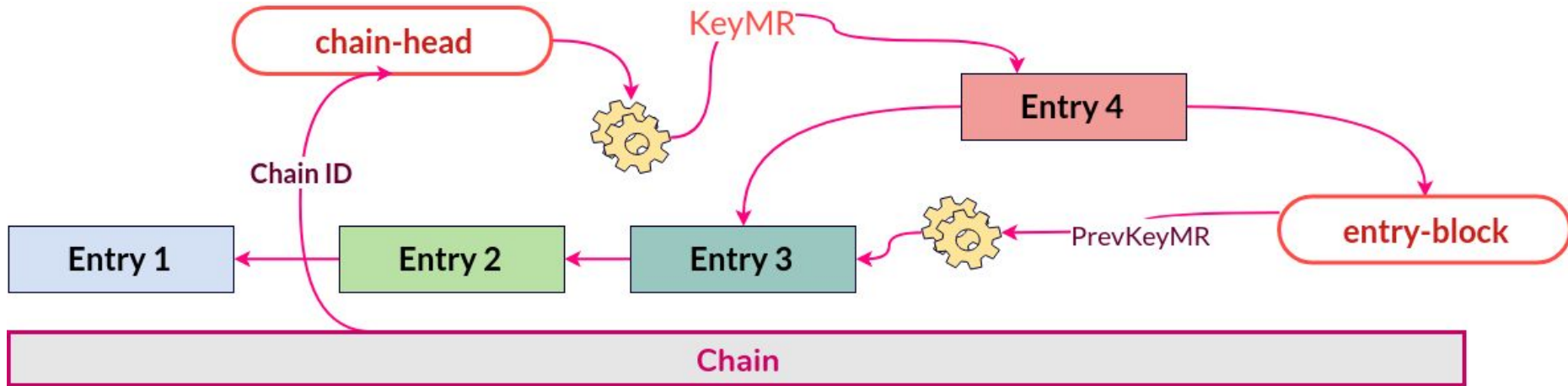
How to retrieve a Chain?



Use chain-head to retrieve a last KeyMR



Use entry-block to recursive all chain

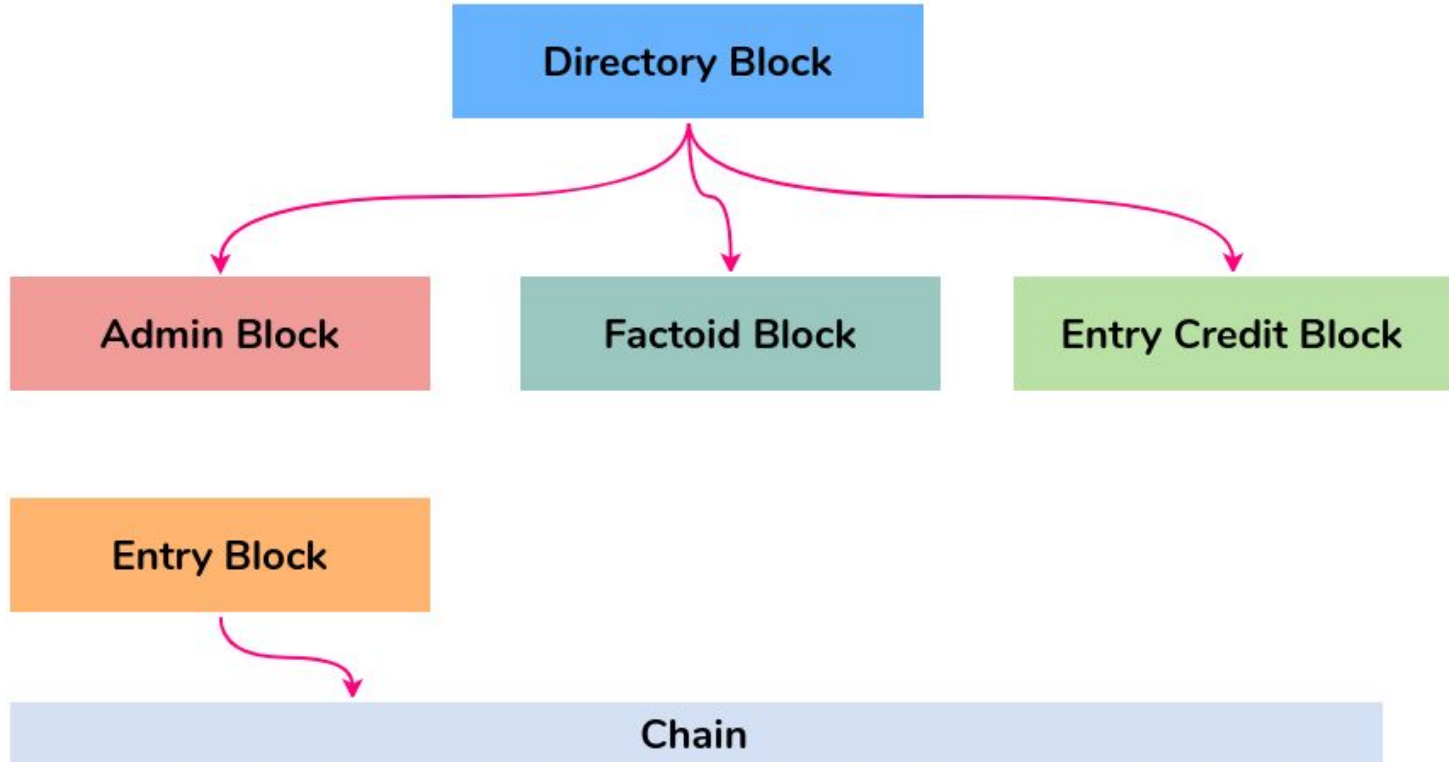




Factom APIs Usage

Retrieve Blocks

Types of Blocks



How to retrieve?

Merkel Root Key

directory-block
admin-block
factoid-block
entrycredit-block
entry-block

Height

ablock-by-height
dblock-by-height
ecblock-by-height
fblock-by-height

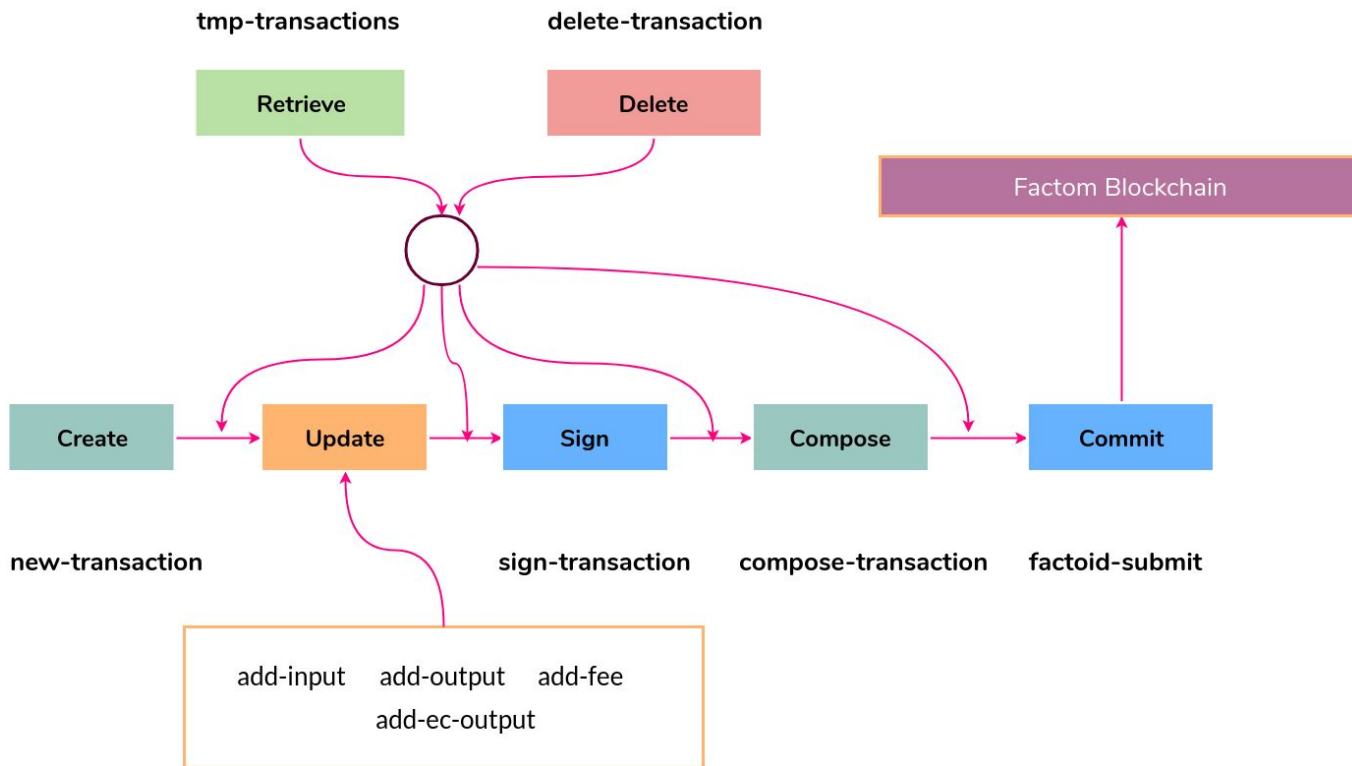


Factom APIs Usage

Life of Transaction

What Is a Transaction In Factom?

How to do it with APIs?



Summary

REFERENCES

- + Factom Data Structures:
<https://github.com/FactomProject/FactomDocs/blob/master/factomDataStructureDetails.md>
- + Enterprise Wallet: <https://docs.factom.com/#enterprise-wallet>
- + Factom Control Panel:
<https://docs.factom.com/#the-factomd-control-panel>
- + Factom Developer Sandbox Setup:
<https://github.com/FactomProject/FactomDocs/blob/master/developerSandboxSetup.md>
- + Factomd APIs: <https://docs.factom.com/api>