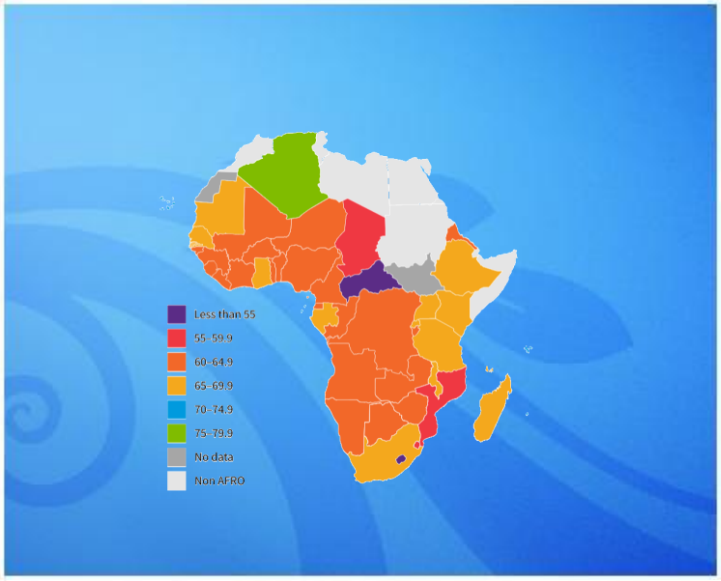
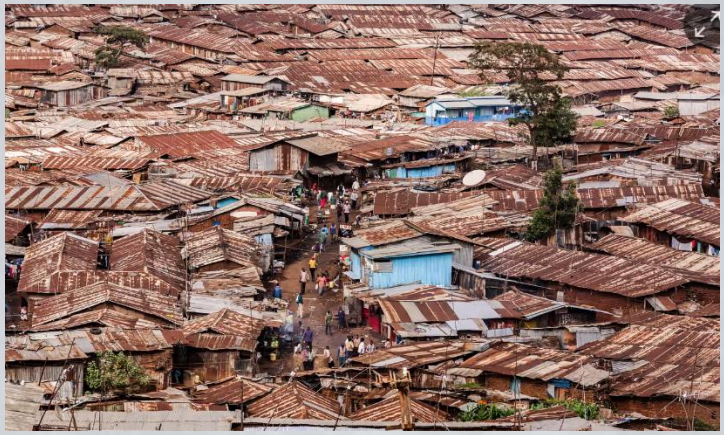
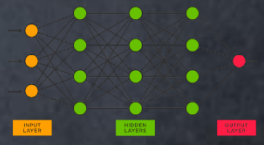




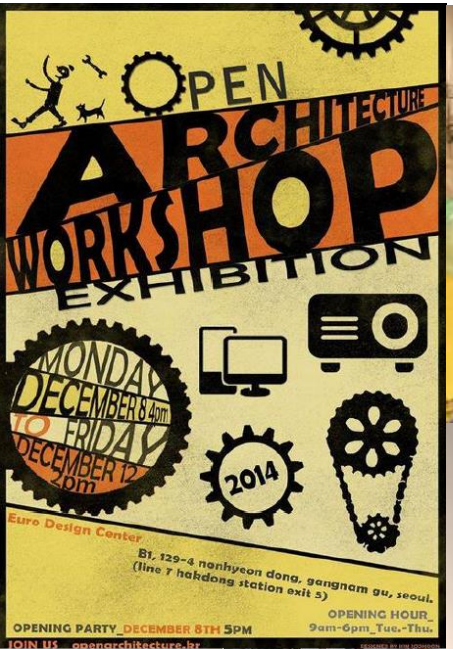
AHPB platform

Ph.D Taewook Kang
laputa99999@gmail.com
daddynkidsmakers.blogspot.com



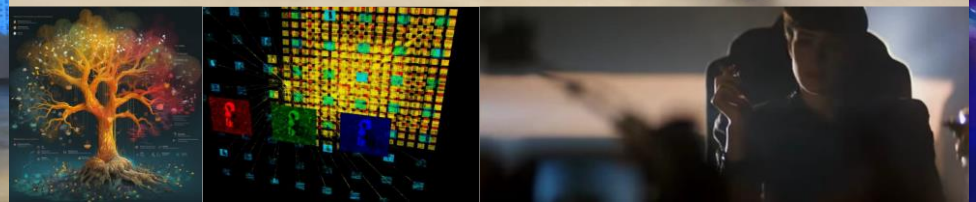
Hobby

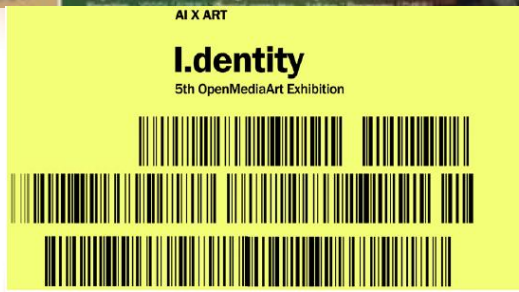
animation, SF, media art

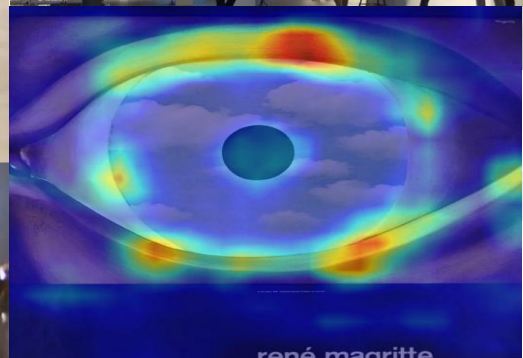


좋아요 · 댓글 달기 · 공유하기

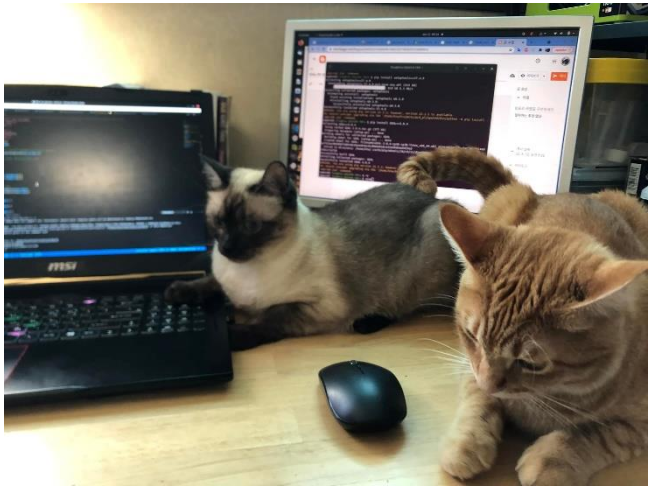
Jintaek Lim, 김주훈, 김현수님 외 5명이 좋아합니다. 114명이 읽음







집에서는 집사



Bitcoin-Based Low-Income Housing Project in Devloping Countries

Survey - Affordable Housing in Africa

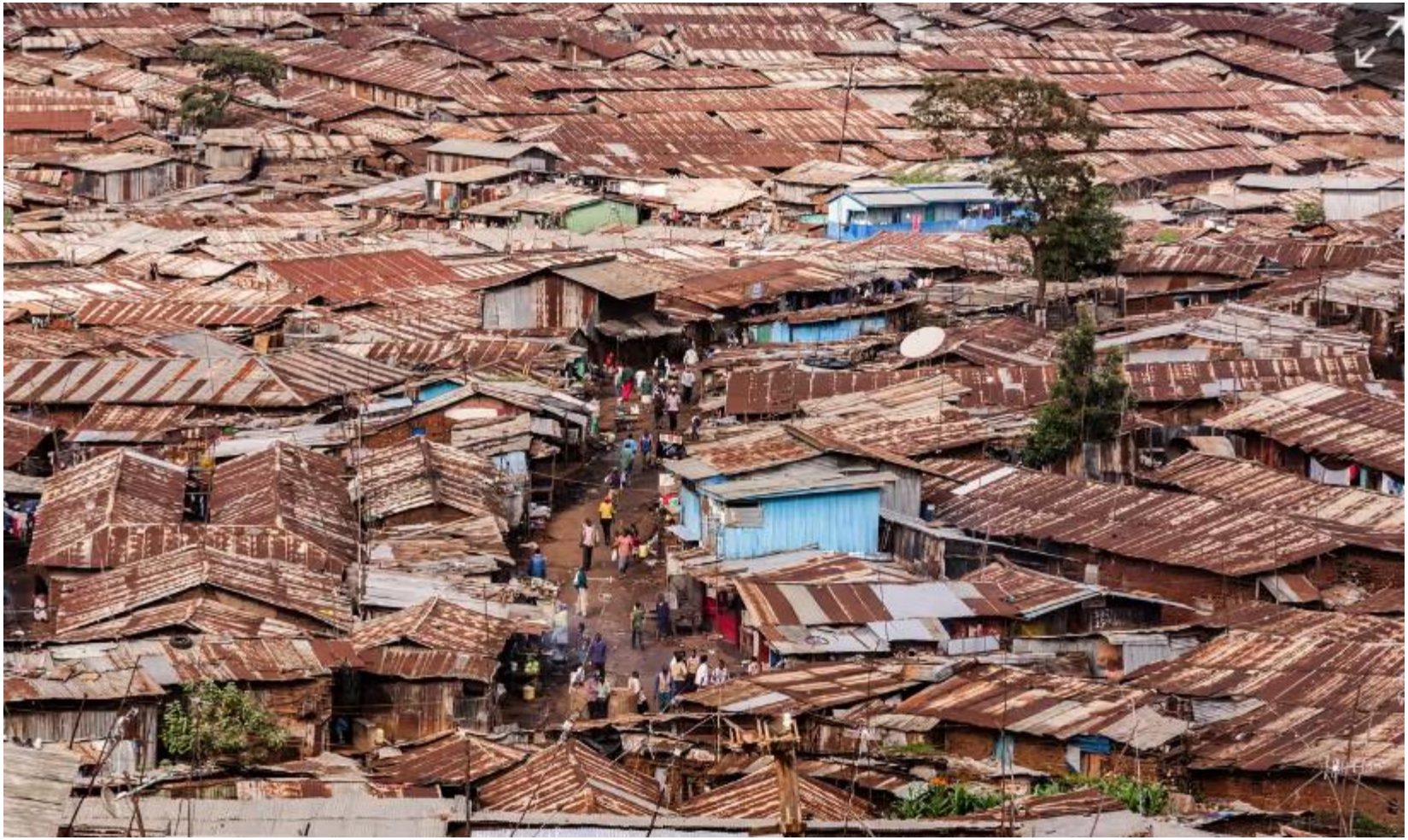
The magnitude of this challenge is evident in the staggering statistics surrounding global wealth distribution and poverty. A staggering 97% of the world's wealth is held by just 30% of the global population, leaving 70% of the planet economically disempowered.

Over the past 50 years, foreign aid amounting to more than $2.3 trillion has been dispersed to poor countries. However, research indicates that this aid can not only be ineffective but possibly damaging to recipient communities and countries.

This failure can be attributed to various factors, including corruption, overheads, audit costs, and money not reaching its intended recipients.

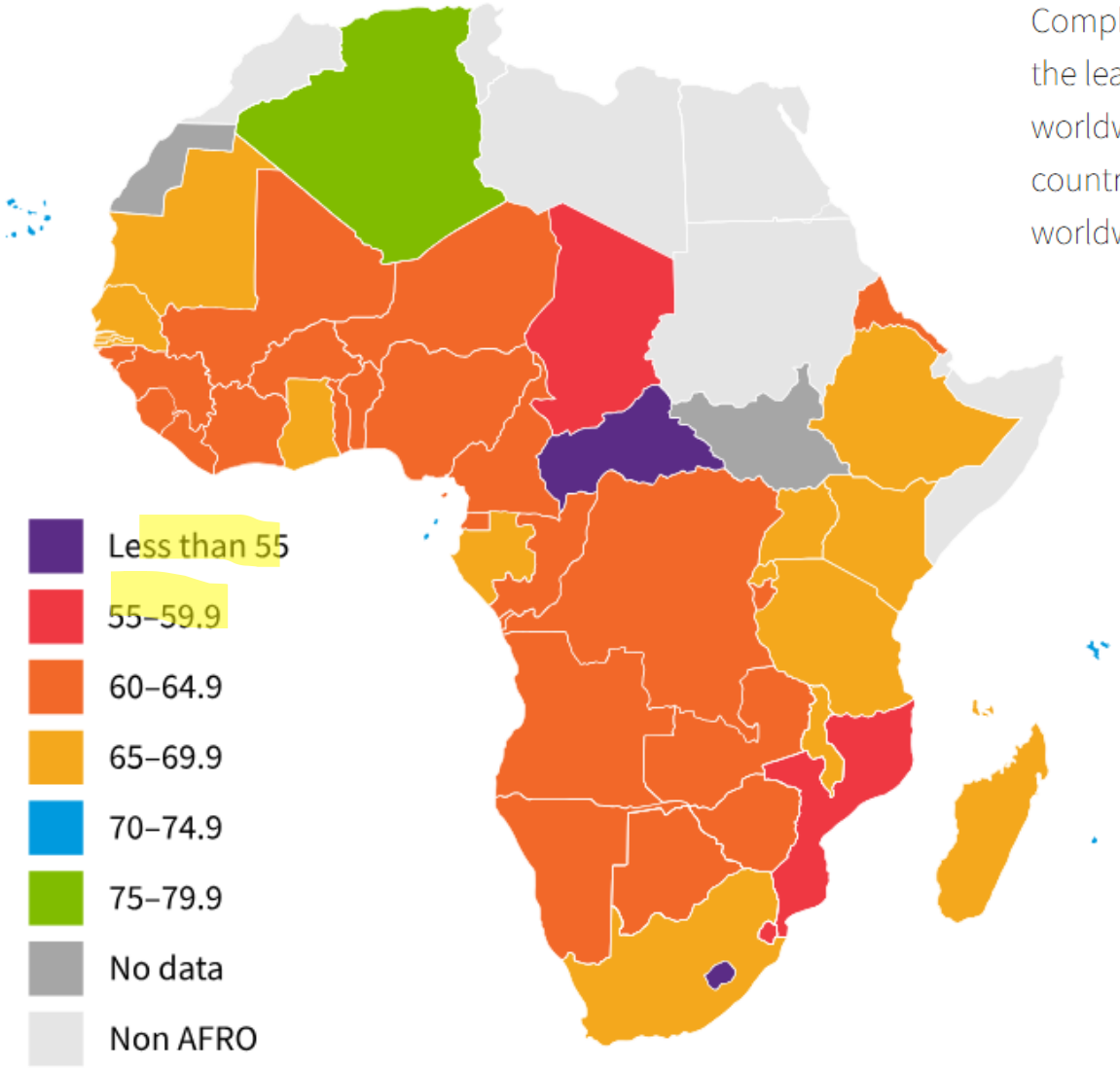
Empowering Communities: DeFi Enabled Affordable Housing in Africa | by Mike Borman | Coinmonks | Medium

Survey - Affordable Housing in Africa



A view of Soweto and Lindi, suburbs of Nairobi belonging to Kibera, the largest urban slum in Africa. Photograph: David Levene/The Guardian

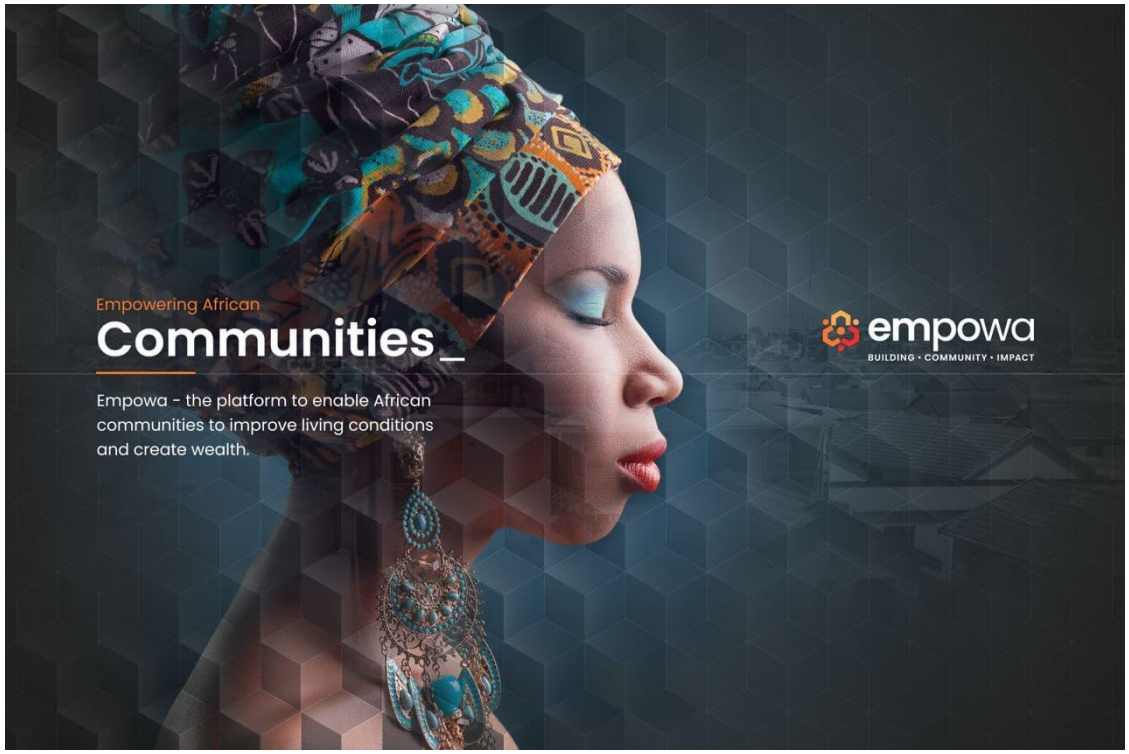
Survey - Affordable Housing in Africa



Complications of pregnancy and childbirth are the leading cause of death for girls aged 15–19 worldwide, and low- and middle-income countries account for 99% of maternal deaths worldwide among women aged 15–49. In

WHO AFRO iAHO | - Section IV - Health Impact

Survey - Affordable Housing in Africa



Empowa | Unlocking Africa's affordable, climate-smart housing opportunity

Survey - Affordable Housing in Africa

Empowa | Unlocking Africa's affordable, climate-smart housing opportunity

Survey - Affordable Housing in Africa

Mary works with Betta Build, an Empowa Project Partner in South Africa, to build rental units on her land. Here's a summary of the process:

1. Betta Build assesses Mary's land and suggests modular, relocatable units due to Mary's lack of formal land title.
2. They agree on building 4 units; an agreement is signed and recorded on the Empowa Platform.
3. Betta Build engages a local supplier, who quotes R450,000 for construction.
4. The rental property is valued at R750,000 and listed on the Empowa Platform for 50,000 EMP tokens.
5. Empowerment Cards are generated, distributed among packs, and sold to fund the project.
6. 30,000 EMP tokens are held in a smart contract for payment; the remaining 20,000 EMP tokens are transferred to Yield Generators.
7. Betta Build helps Mary find a Property Manager, who collects rent, manages tenants, and maintains the property.
8. Rental agreements are recorded on the Empowa Platform, and rent is collected in ZAR before being converted to EMP tokens.
9. Rent is distributed among the Collector, Property Manager, and Mary according to a smart contract.
10. Mary's share of rent unlocks the 20,000 EMP tokens over the ownership transition period.
11. An Insurance fund is maintained to cover potential relocation costs.
12. After the initial ownership transition period, the Insurance fund becomes a tenant Rewards fund.
13. Collectors can sell their Empowerment Cards on the Empowa Platform at any time.
14. At the end of the ownership transition period, Mary takes full ownership, Collectors are paid in EMP tokens, and tenants receive rewards.

Empowering Communities: DeFi Enabled Affordable Housing in Africa | by Mike Borman | Coinmonks | Medium

Survey - HouseAfrica

Survey - Affordable Housing in Africa

"Verification is a key part [of] what we're doing. ... Traditionally, if you're a wholesale provider of capital and you want to send it to somebody in order to provide home loans, you have no view of what's happening on the ground. And that's effectively what we're doing; we're creating that mechanism by which there can be a direct view of exactly what's happening. And that's one of the ways we mitigate risk."

Jordan described the verification element of the Empowa platform by highlighting that payments for funding home loans and leases are recorded on-chain. That, in theory, brings more transparency to the process.

Altogether, time would prove the feasibility of Empowa's vision. A few have tried to build blockchain-based real estate solutions for Africa. One of them, HouseAfrica, launched in 2019, seeking to decentralize access to real estate investment. It has since pivoted away from investing to running a blockchain-based land registry service called Sytemap. HouseAfrica recently raised a $400,000 venture investment.

Fresh off a $3.75 million token sale, African blockchain company Empowa is looking to fix Africa's broken home loan market (mariblock.com)

Survey - HouseAfrica

Our Offer

Satellite Map

Our smooth and secure satellite platform streamlines the process of verifying land locations and distances for property buyers.

Blockchain & NFT

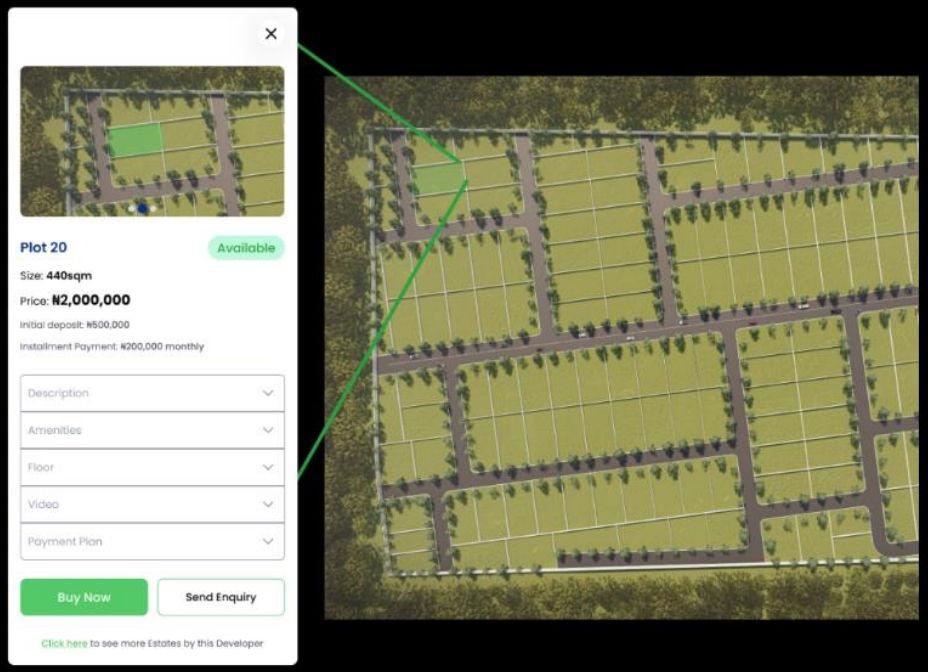
We leverages on blockchain & NFT in helping real estate companies issue verifiable certificate of land allocation with ease.

Payments & Payouts

We makes it easier to real estate companies to accept payments from anywhere, anytime and also automates their commission

API Tools (coming soon)

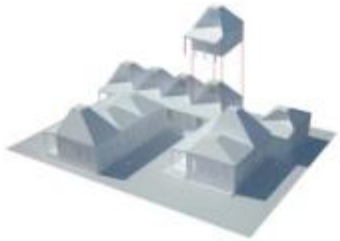
With our, well-documented APIs, you can build Proptech products (Verification tool, Payment collections and tokenization)



Sytemap uses satellite map to help first-time buyers and diasporas to easily verify, choose better lands, make payments and get digital title powered on blockchain & NFT.

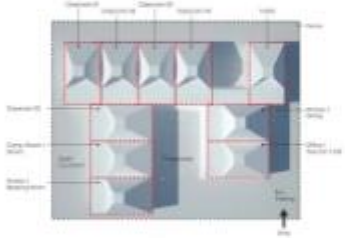
Explore sytemap

BIM building information modeling & Modular Africa







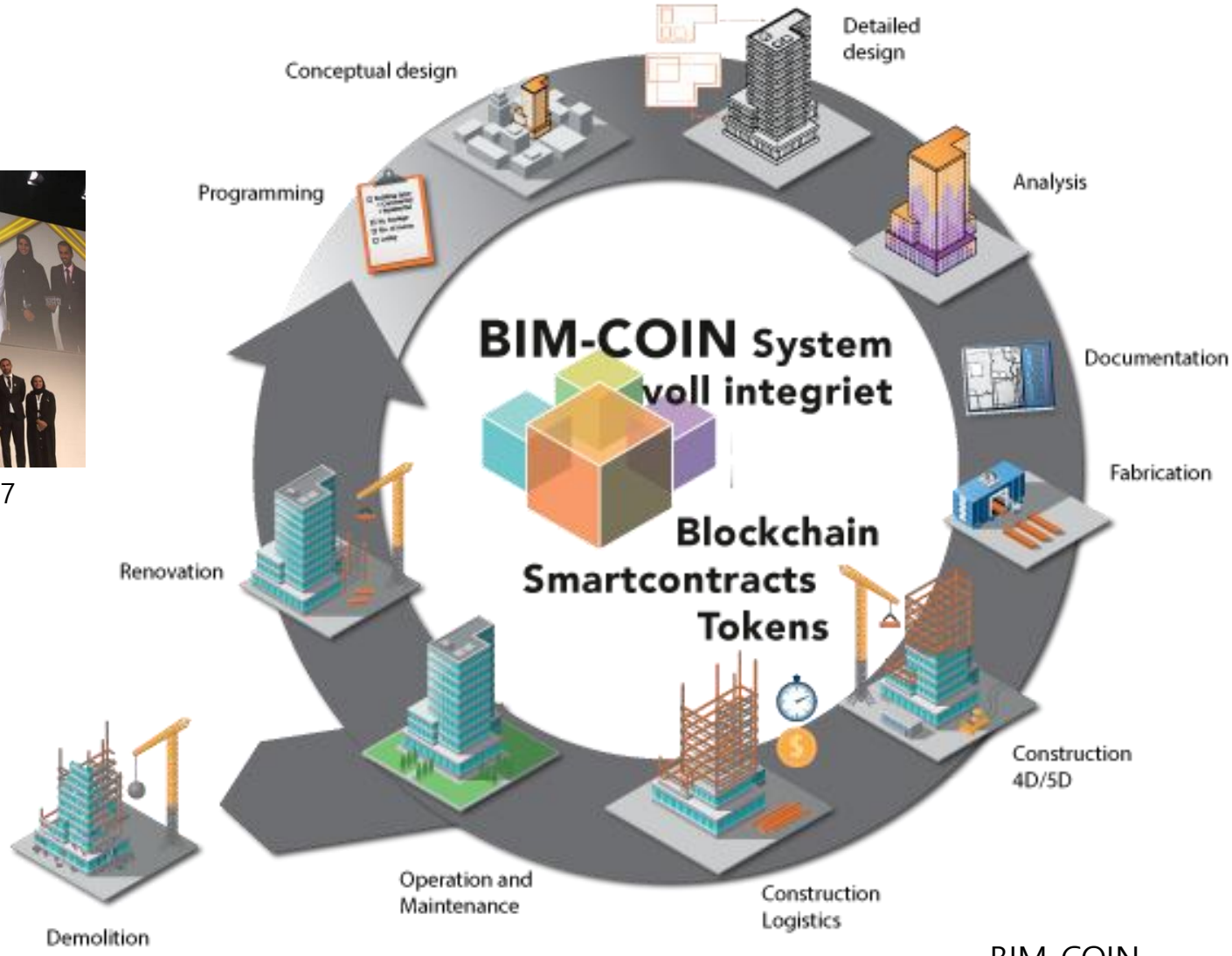



Modular Africa | Nottoscale | Archello


modular dwelling proposal could provide solution to the rural sub-saharan africa housing crisis (designboom.com)

BIM building information modeling & Modular Africa

Albawaba business, 2017

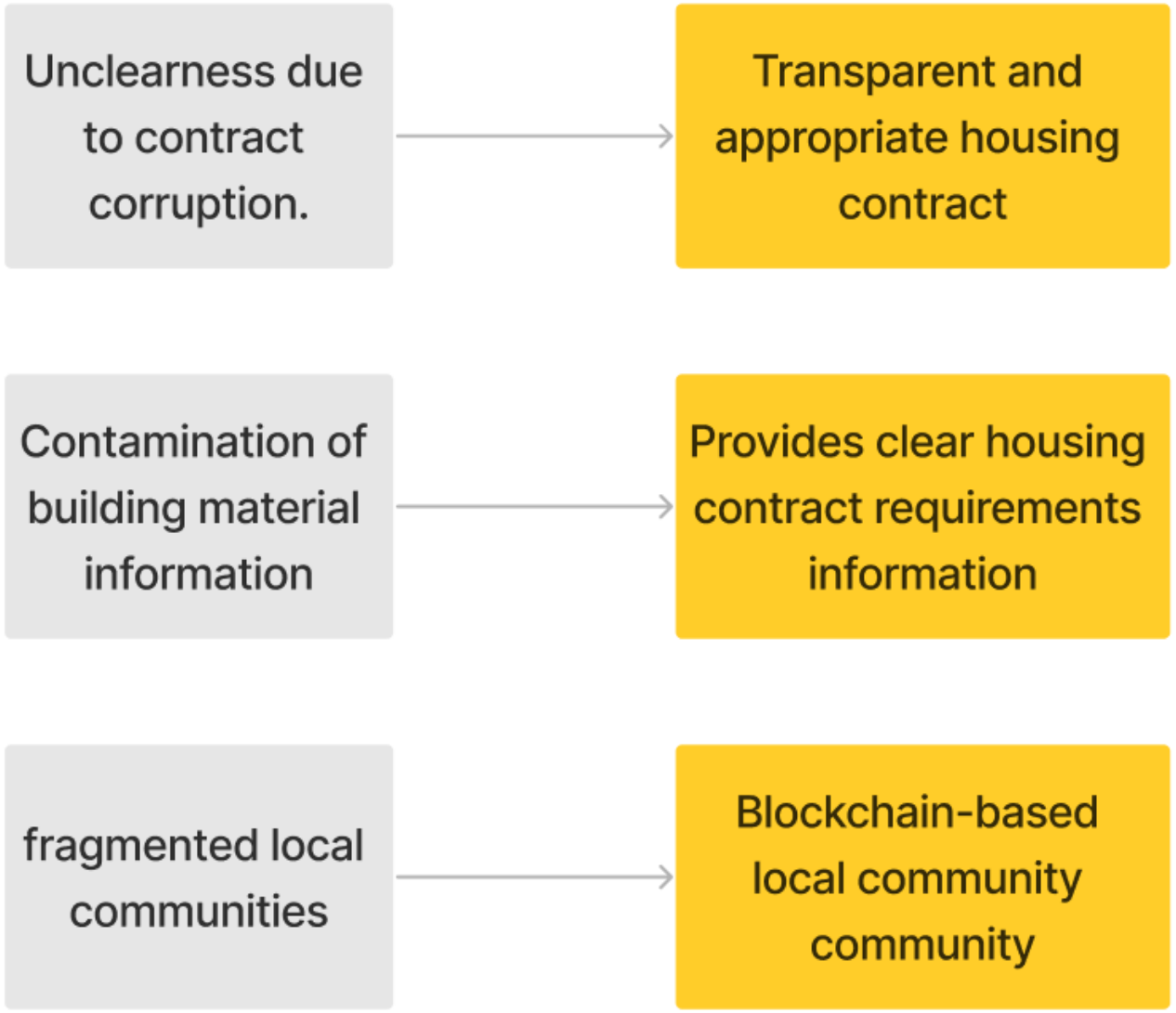


BIM principle, 2018.1

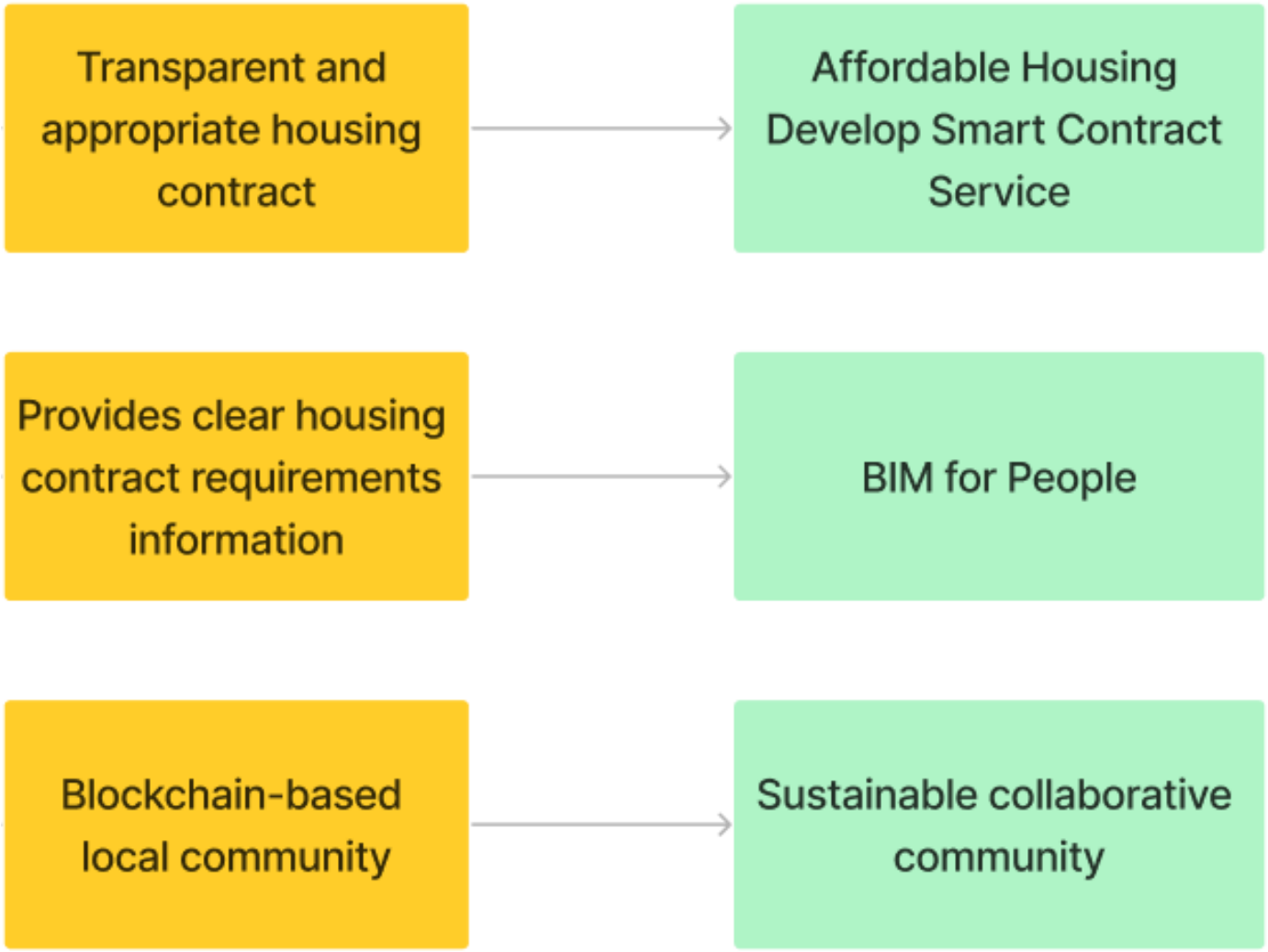
BIM-COIN
DUBAI CITY

AHPB solution

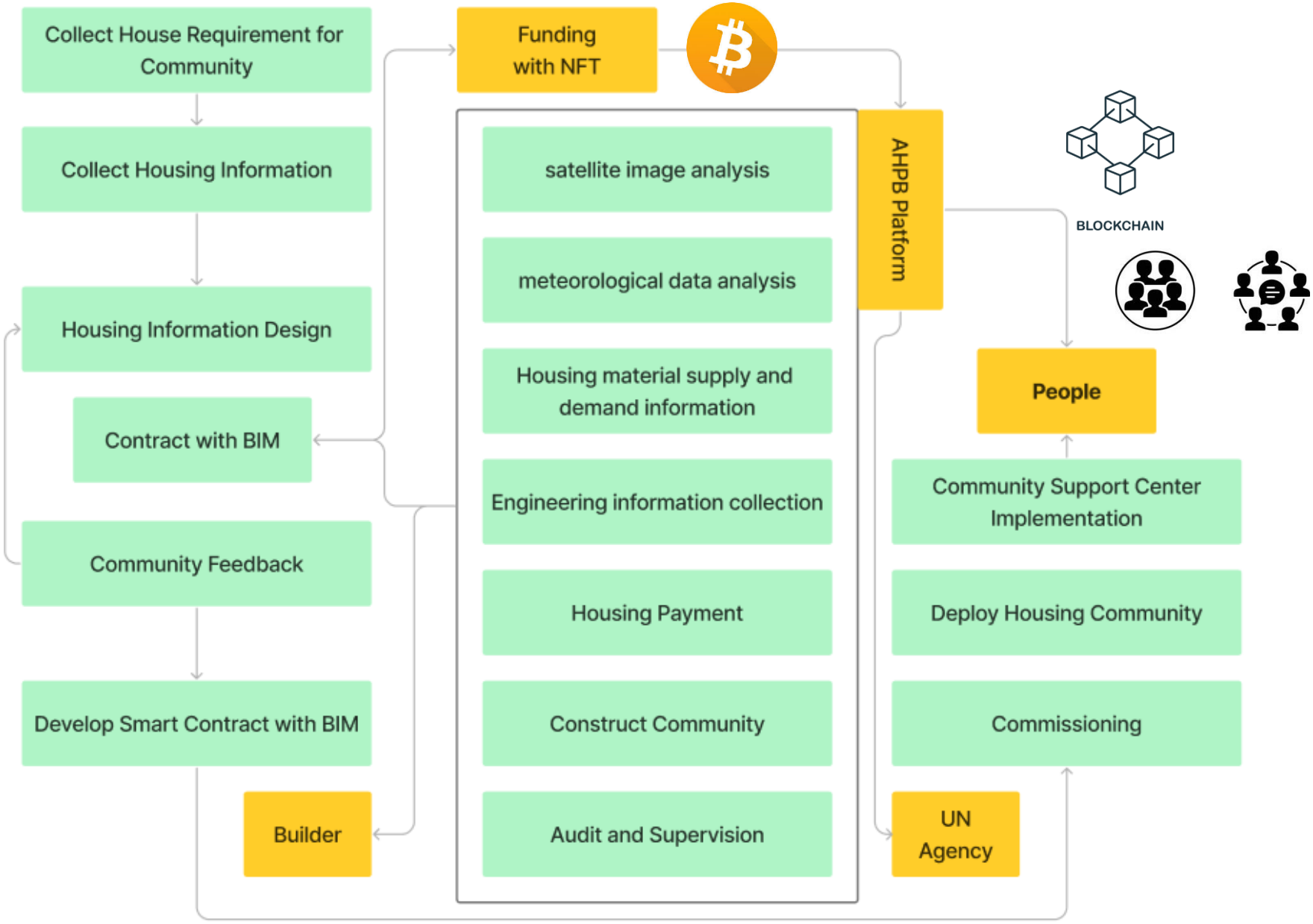
Affordable Housing for People powered by Blockchain - AHPB



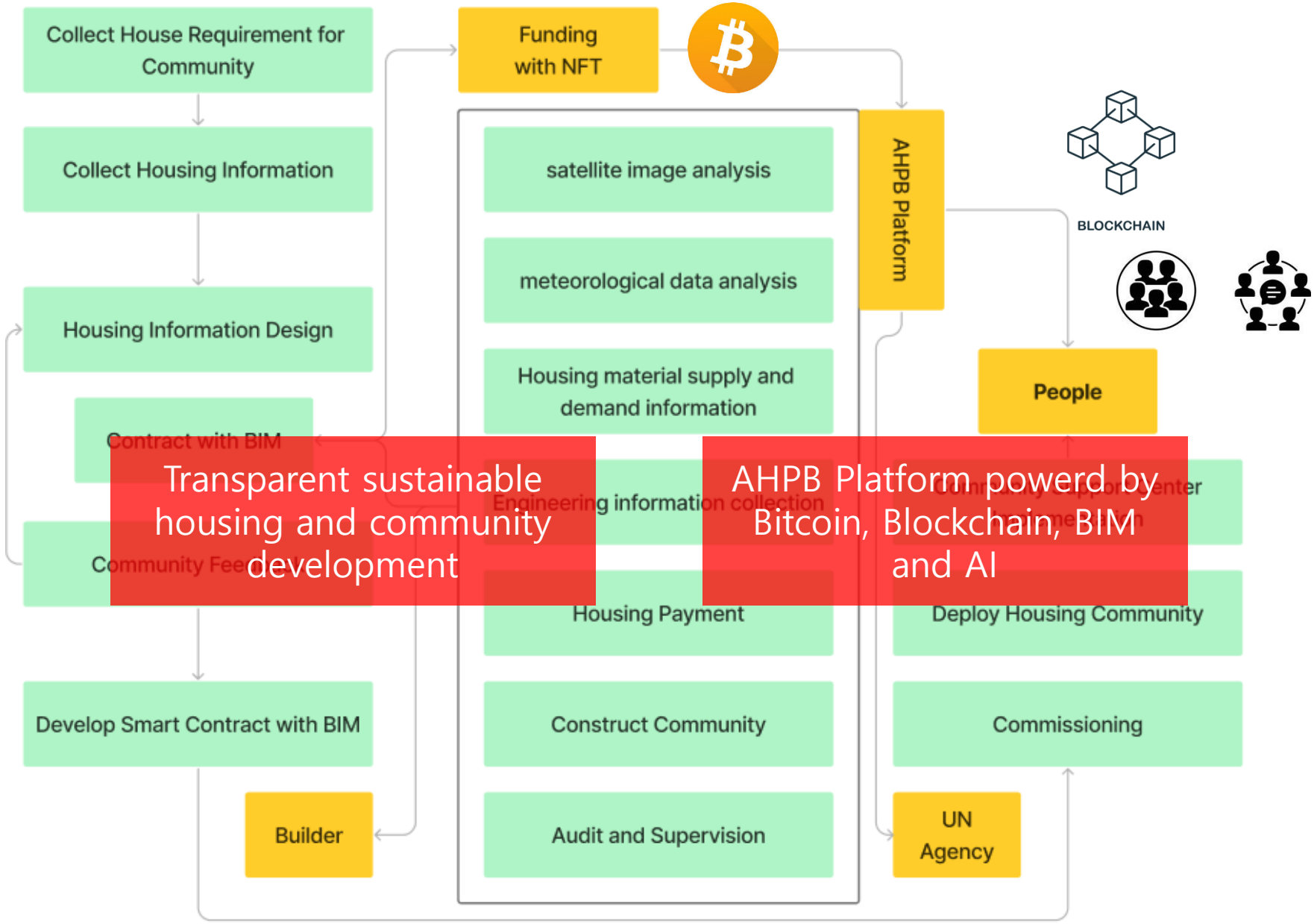
Affordable Housing for People powered by Blockchain - AHPB



Affordable Housing for People powered by Blockchain - AHPB



Affordable Housing for People powered by Blockchain - AHPB



Affordable Housing for People powered by Blockchain - AHPB

Affordable Housing for People powered by Blockchain - AHPB


STRONG AM I
WITH THE BLOCKCHAIN

Thanks

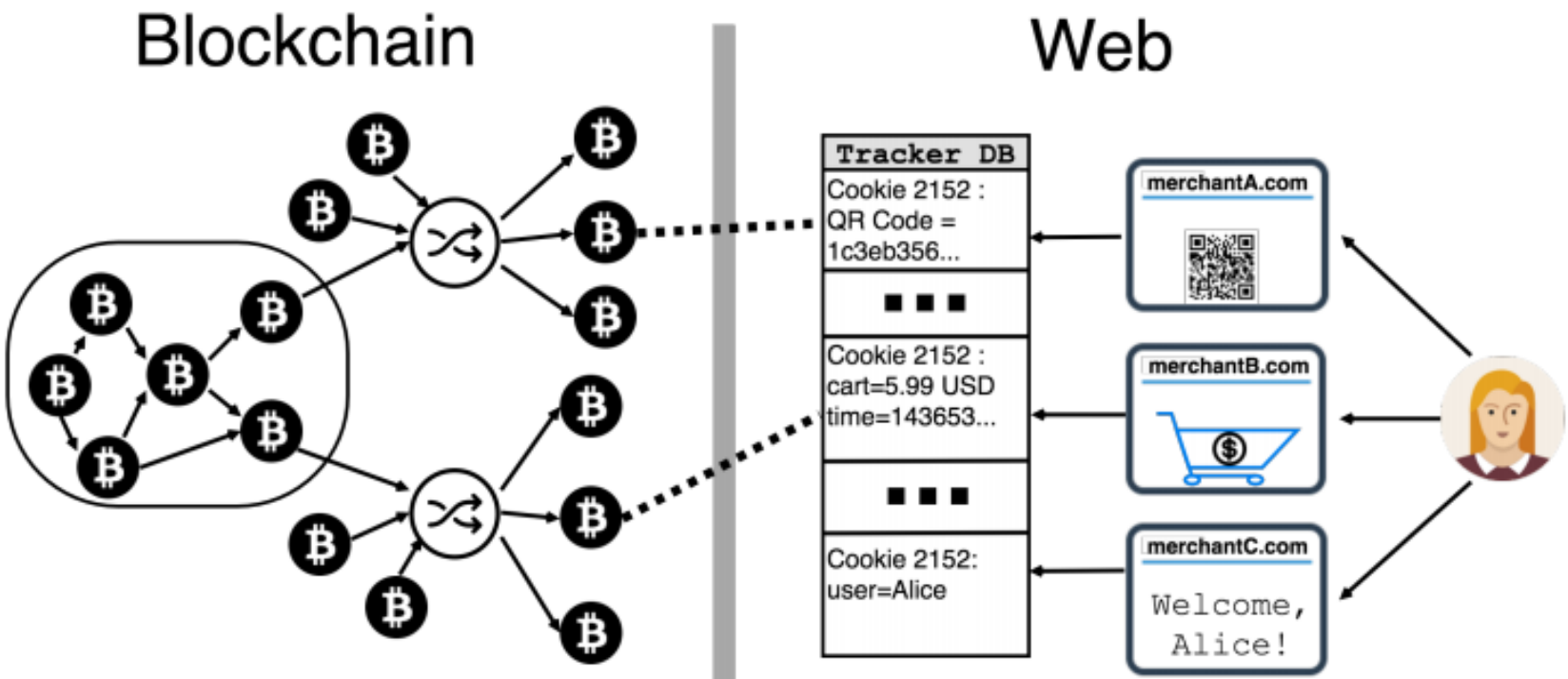
Daddy Makers
(daddynkidsmakers.blogspot.com)

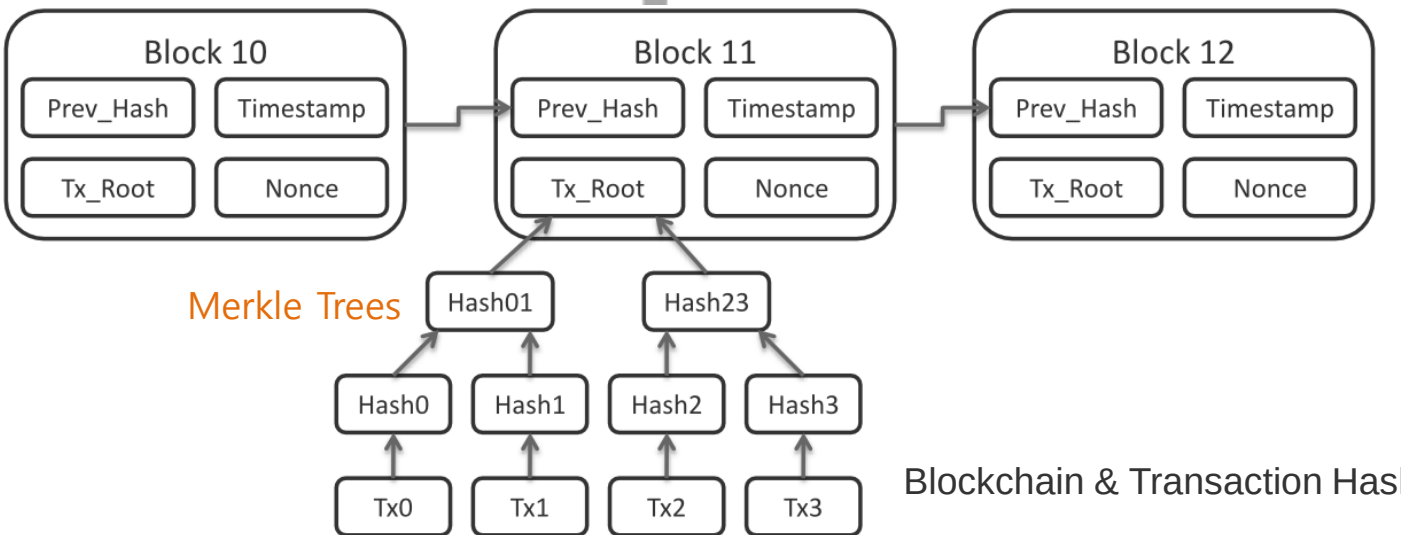
Computer graphics digest on Apple Podcasts
Apple Podcasts － 《BIM digest》
Software engineering digest on Apple Podcasts

laputa99999@gmail.com

Bitcoin Research

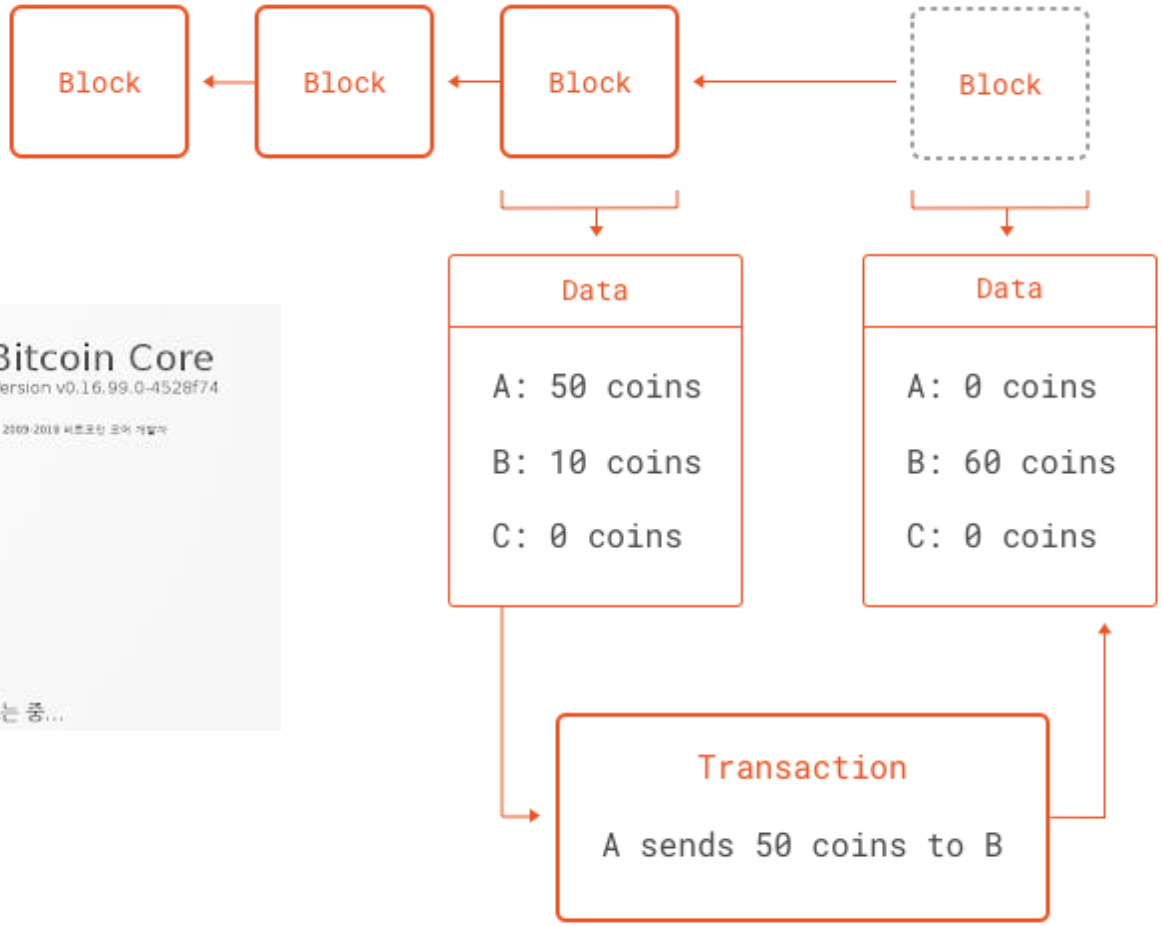
Bitcoin architecture





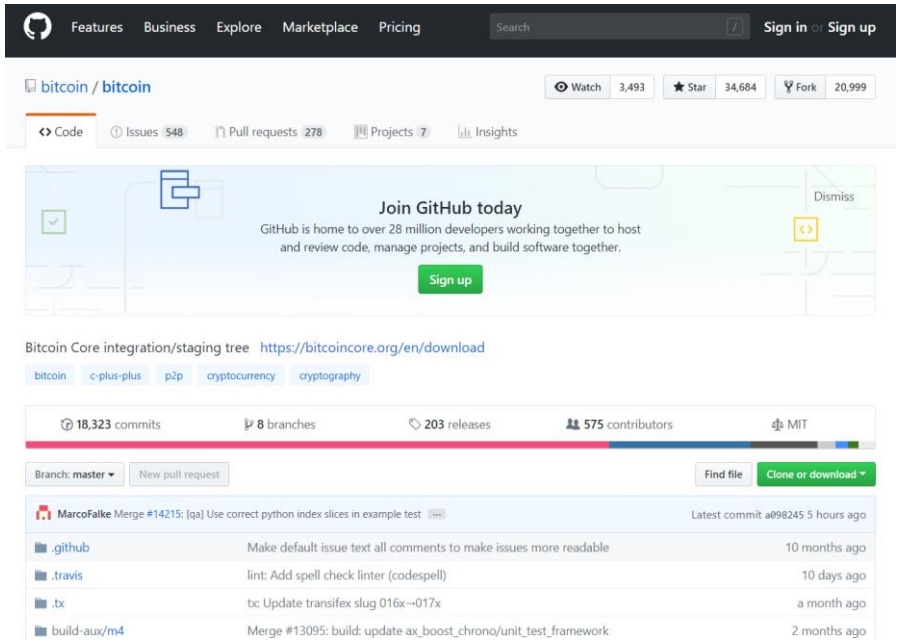
Blockchain & Transaction Hash (Wikipedia)

Bitcoin architecture



Bitcoin wiki

Bitcoin architecture



Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

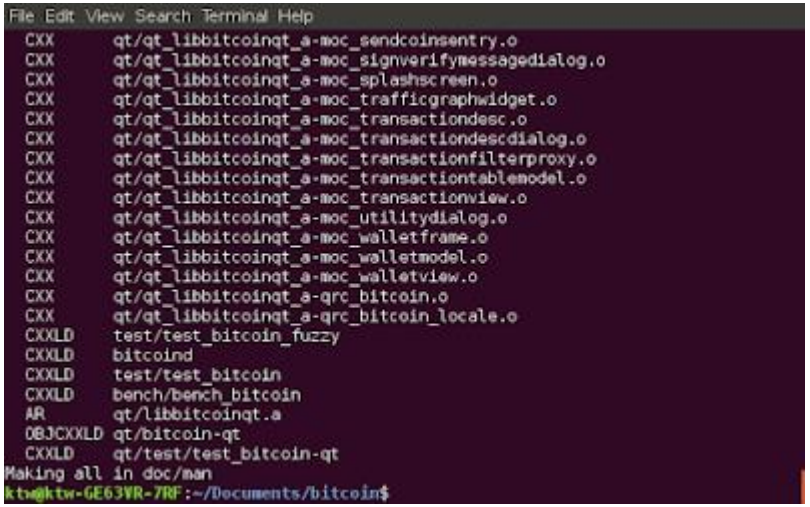
Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions,

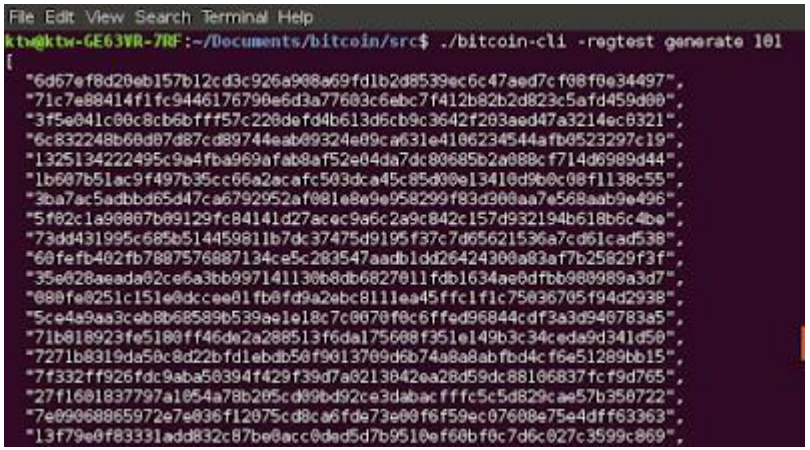
Bitcoin architecture

[Bitcoin core analysis]

sudo apt-get update
sudo apt-get upgrade

git clone https://github.com/bitcoin/bitcoin.git
make -s -j5



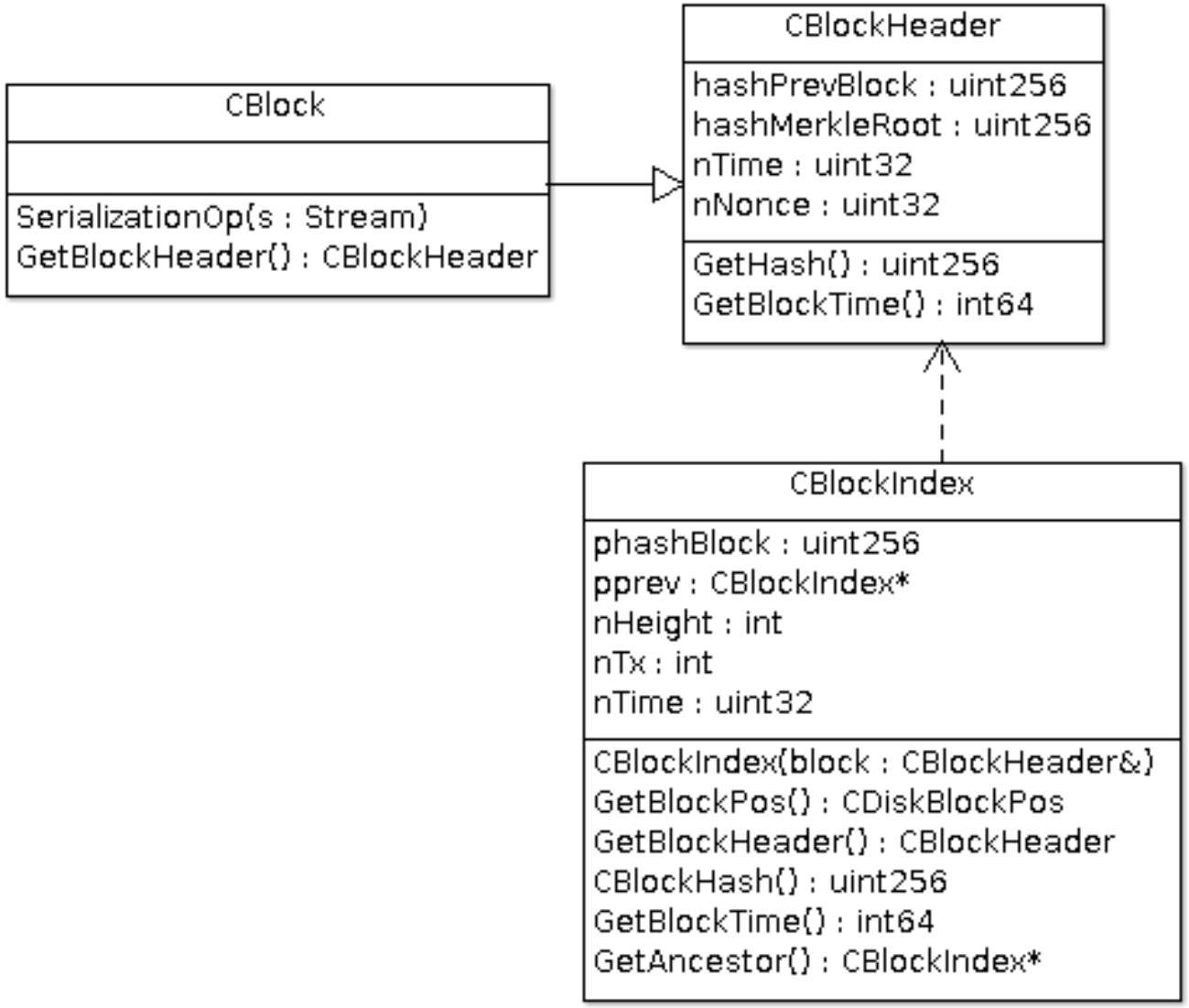


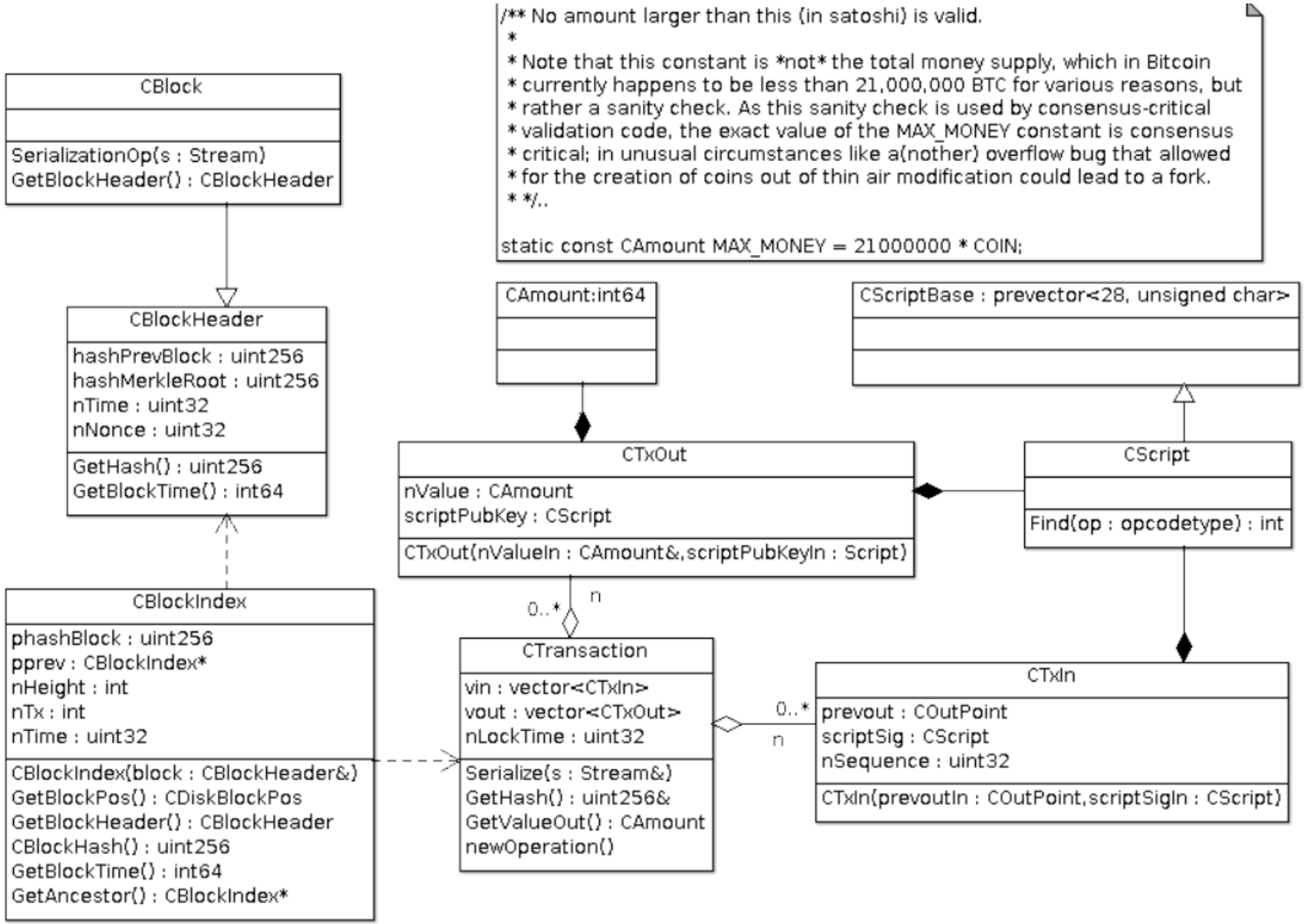
Bitcoin server start
./bitcoin-cli -regtest generate 101
./bitcoin-cli -regtest getblockcount

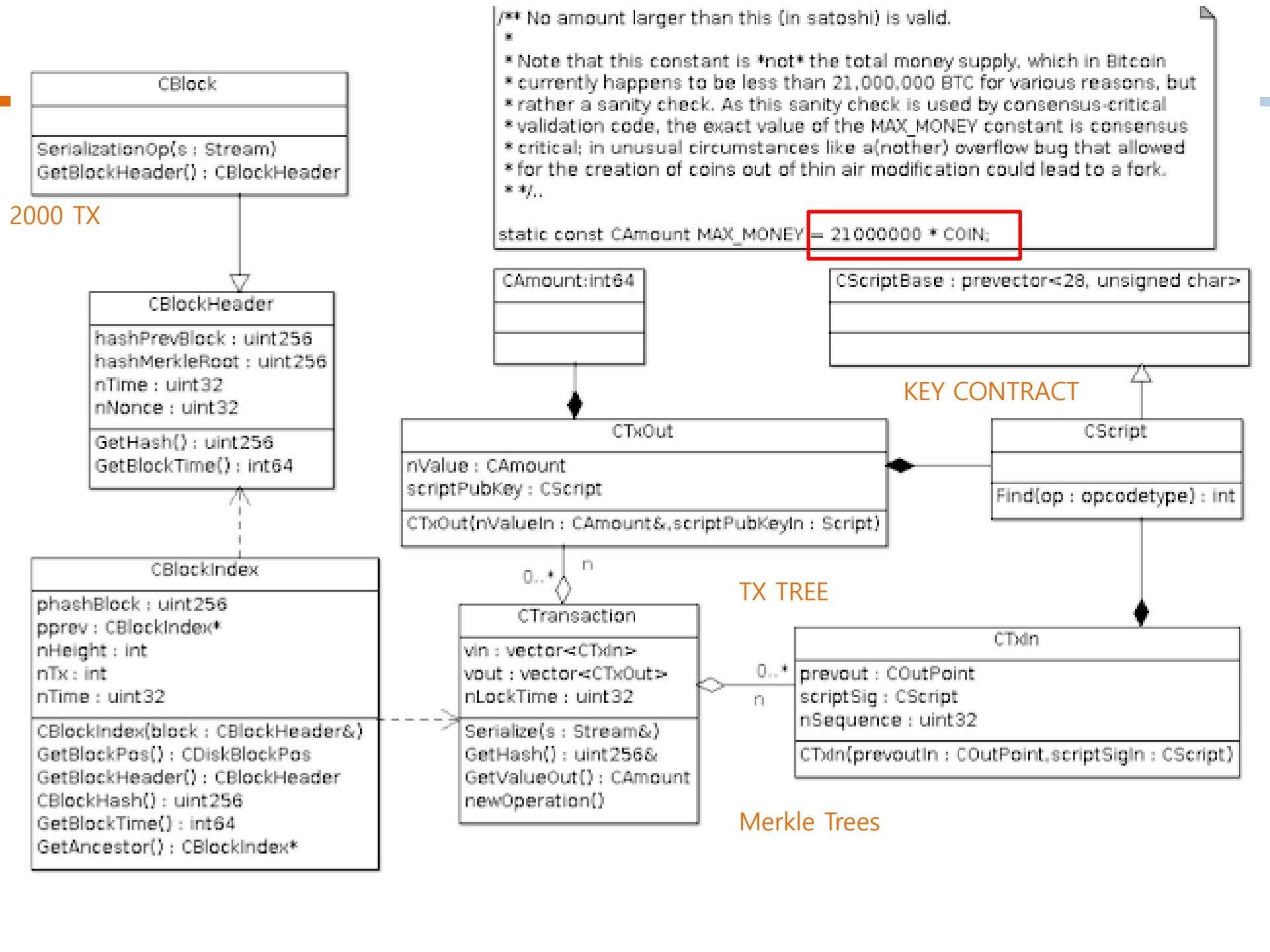
./bitcoin-cli -regtest getnewaddress ktw
./bitcoin-cli -regtest getbalance
50.00000000

./bitcoin-cli -regtest sendtoaddress [앞에서 생성한 계좌번호] 10
./bitcoin-cli -regtest generate 1
[
"36254b11d6c28434b0e14a2a84d633d38e46177d9298a56e132346a3d340be0c"
]

Bitcoin architecture




/** No amount larger than this (in satoshi) is valid.
*
* Note that this constant is *not* the total money supply, which in Bitcoin
* currently happens to be less than 21,000,000 BTC for various reasons, but
* rather a sanity check. As this sanity check is used by consensus-critical
* validation code, the exact value of the MAX_MONEY constant is consensus
* critical; in unusual circumstances like a(nother) overflow bug that allowed
* for the creation of coins out of thin air modification could lead to a fork.
* */..
static const CAmount MAX_MONEY = 21000000 * COIN;
CBlock
SerializationOp(s : Stream)
GetBlockHeader() : CBlockHeader
CBlockHeader
hashPrevBlock : uint256
hashMerkleRoot : uint256
nTime : uint32
nNonce : uint32
GetHash() : uint256
GetBlockTime() : int64
CAmount:int64
CScriptBase : prevector<28, unsigned char>
CTxOut
nValue : CAmount
scriptPubKey : CScript
CTxOut(nValueIn : CAmount&,scriptPubKeyIn : Script)
CScript
Find(op : opcodetype) : int
CBlockIndex
phashBlock : uint256
pprev : CBlockIndex*
nHeight : int
nTx : int
nTime : uint32
CBlockIndex(block : CBlockHeader&)
GetBlockPos() : CDiskBlockPos
GetBlockHeader() : CBlockHeader
CBlockHash() : uint256
GetBlockTime() : int64
GetAncestor() : CBlockIndex*
0..*
n
CTransaction
vin : vector<CTxIn>
vout : vector<CTxOut>
nLockTime : uint32
Serialize(s : Stream&)
GetHash() : uint256&
GetValueOut() : CAmount
newOperation()
0..*
n
CTxIn
prevout : COutPoint
scriptSig : CScript
nSequence : uint32
CTxIn(prevoutIn : COutPoint,scriptSigIn : CScript)


/** No amount larger than this (in satoshi) is valid.
*
* Note that this constant is *not* the total money supply, which in Bitcoin
* currently happens to be less than 21,000,000 BTC for various reasons, but
* rather a sanity check. As this sanity check is used by consensus-critical
* validation code, the exact value of the MAX_MONEY constant is consensus
* critical; in unusual circumstances like a(nother) overflow bug that allowed
* for the creation of coins out of thin air modification could lead to a fork.
* */
static const CAmount MAX_MONEY = 21000000 * COIN;
CBlock
SerializationOp(s : Stream)
GetBlockHeader() : CBlockHeader
2000 TX
CBlockHeader
hashPrevBlock : uint256
hashMerkleRoot : uint256
nTime : uint32
nNonce : uint32
GetHash() : uint256
GetBlockTime() : int64
CAmount:int64
CScriptBase : prevector<28, unsigned char>
KEY CONTRACT
CTxOut
nValue : CAmount
scriptPubKey : CScript
CTxOut(nValueIn : CAmount&,scriptPubKeyIn : Script)
CScript
Find(op : opcodetype) : int
CBlockIndex
phashBlock : uint256
pprev : CBlockIndex*
nHeight : int
nTx : int
nTime : uint32
CBlockIndex(block : CBlockHeader&)
GetBlockPos() : CDiskBlockPos
GetBlockHeader() : CBlockHeader
CBlockHash() : uint256
GetBlockTime() : int64
GetAncestor() : CBlockIndex*
0..*
n
TX TREE
CTransaction
vin : vector<CTxIn>
vout : vector<CTxOut>
nLockTime : uint32
Serialize(s : Stream&)
GetHash() : uint256&
GetValueOut() : CAmount
newOperation()
0..*
n
CTxIn
prevout : COutPoint
scriptSig : CScript
nSequence : uint32
CTxIn(prevoutIn : COutPoint,scriptSigIn : CScript)
Merkle Trees