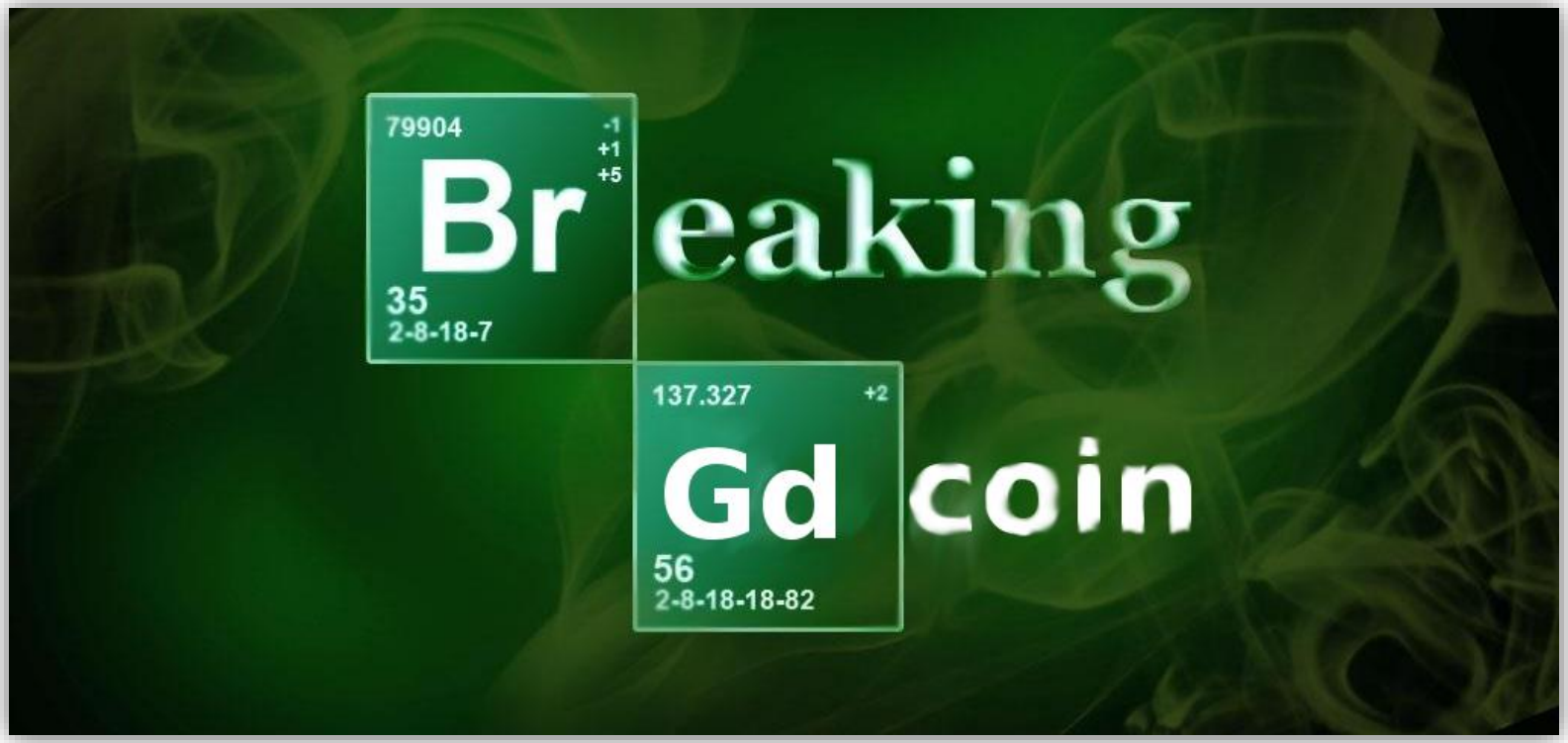


RUHR-UNIVERSITÄT BOCHUM

Breaking and Fixing Gridcoin

Martin Grothe, Tobias Niemann, Juraj Somorovsky and Jörg Schwenk

Source: http://gaincryptocurrency.altervista.org/alterpages/10710353_865938750143391_6543553170888387417_o.jpg

Bitcoin

Bitcoin

Block chain - details

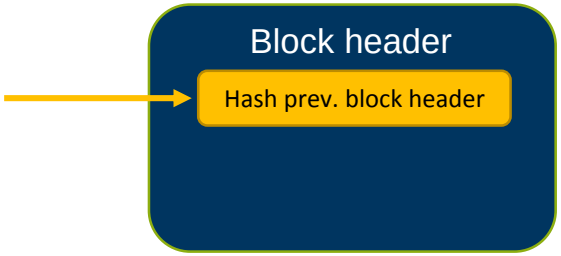
Bitcoin

Block chain - details



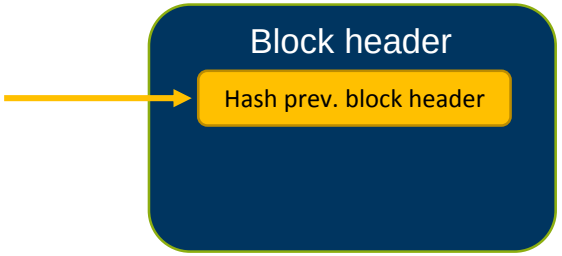
Bitcoin

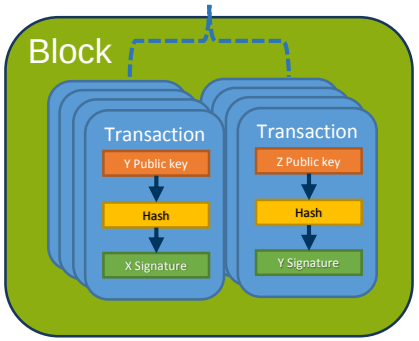
Block chain - details



Bitcoin

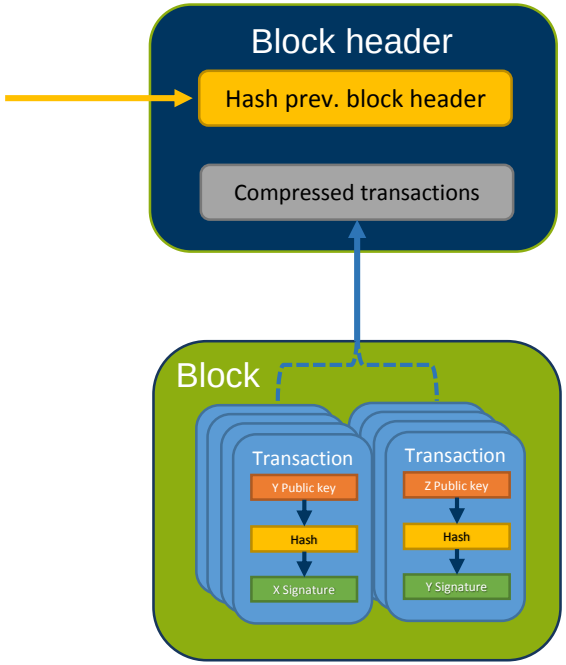
Block chain - details





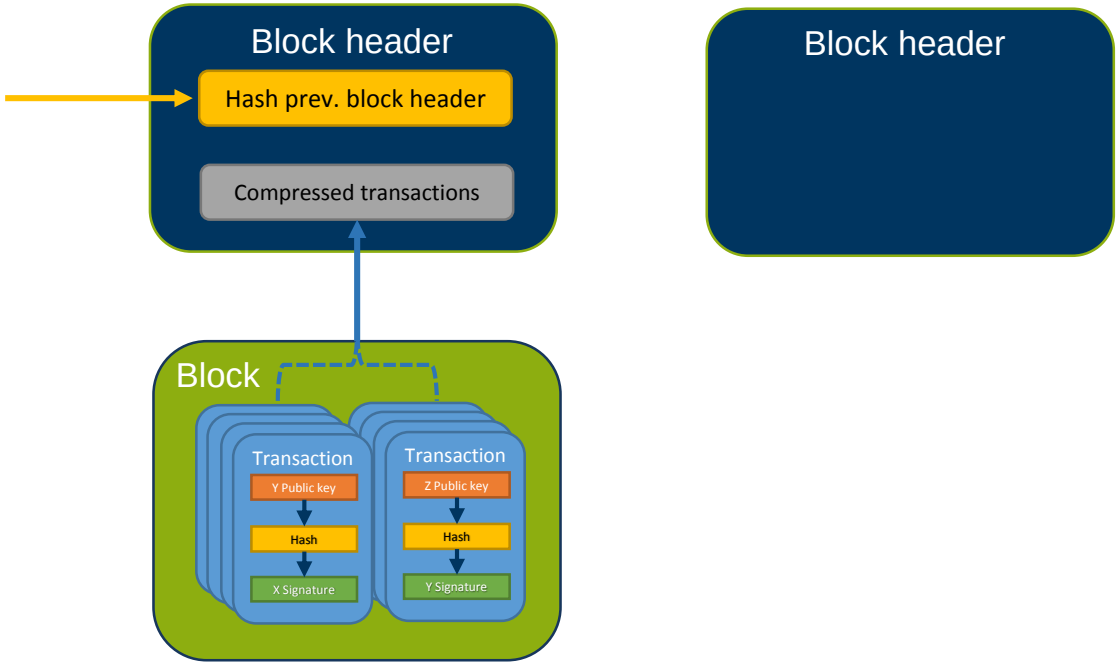
Bitcoin

Block chain - details



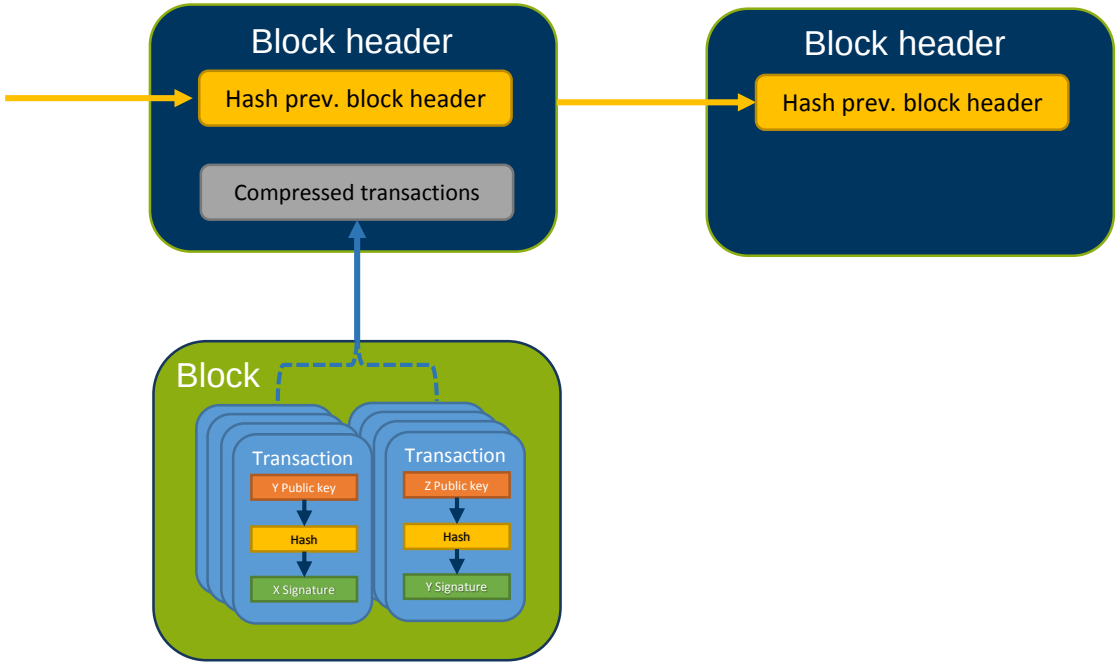
Bitcoin

Block chain - details



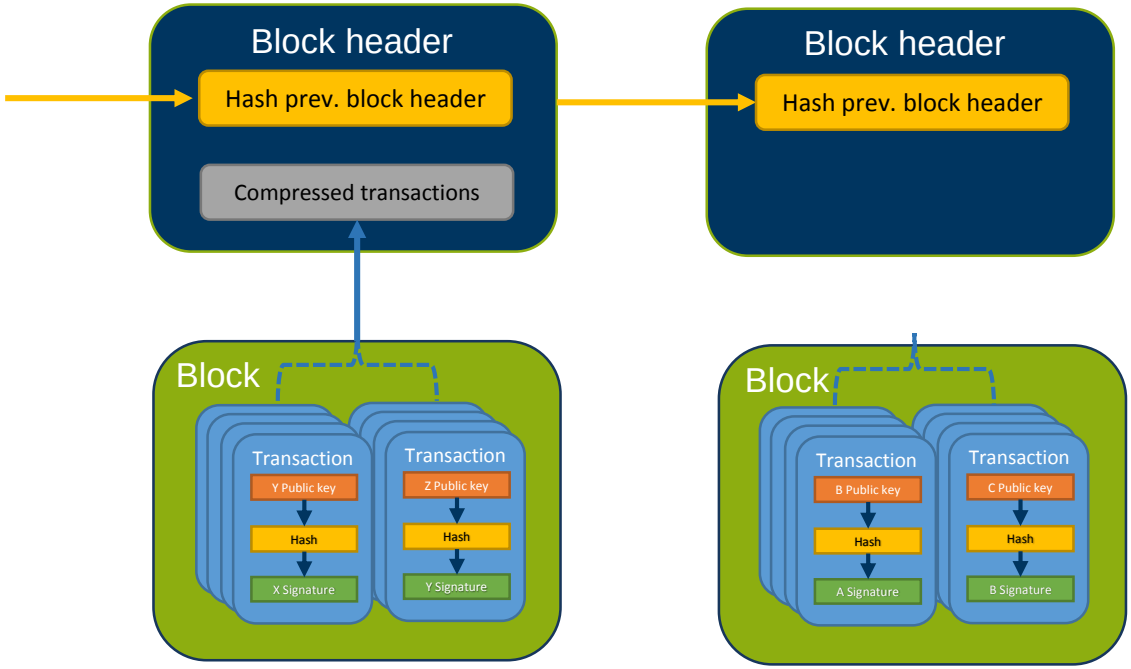
Bitcoin

Block chain - details



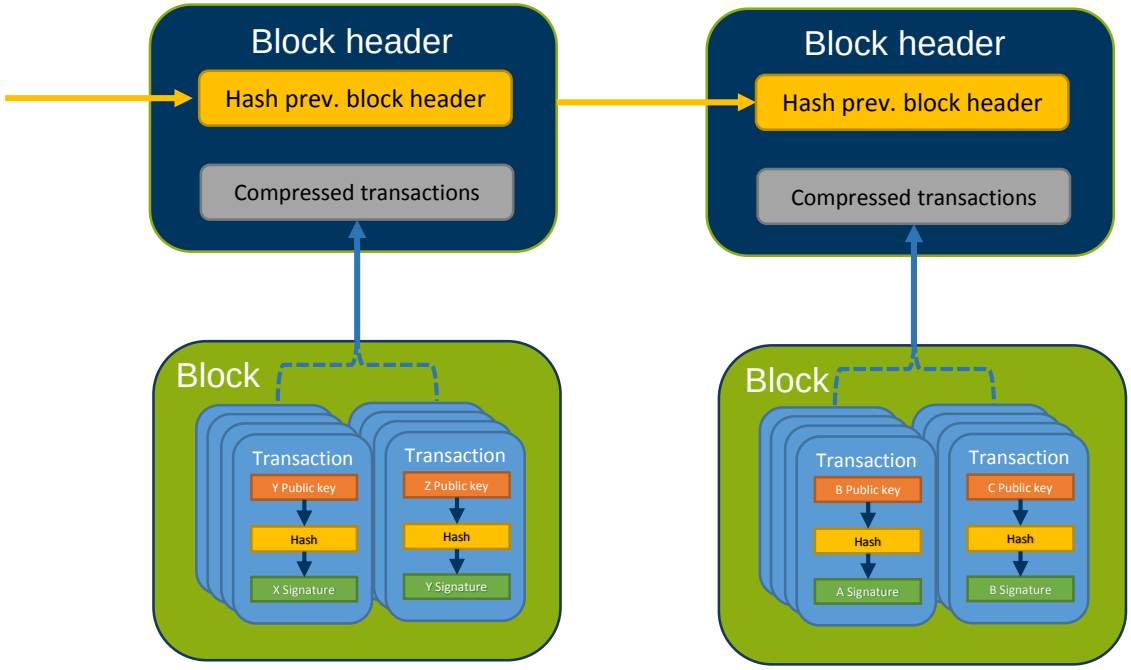
Bitcoin

Block chain - details



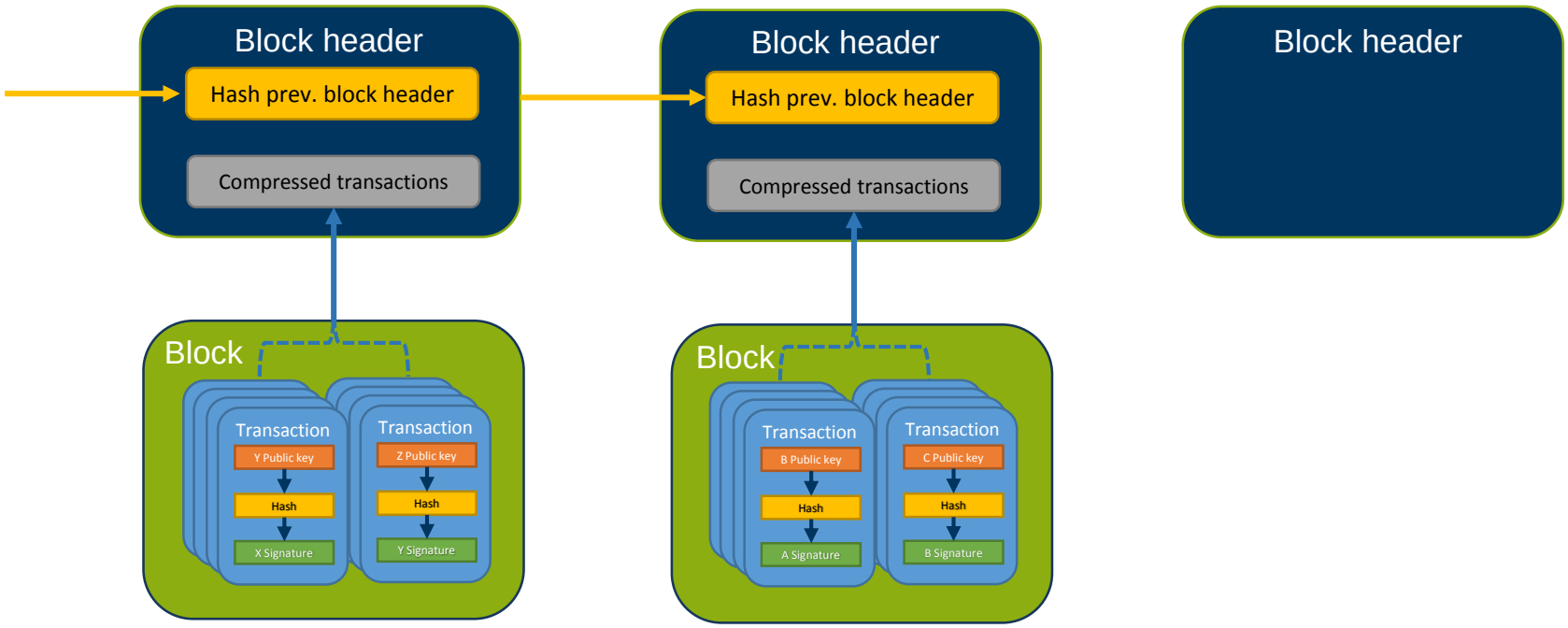
Bitcoin

Block chain - details



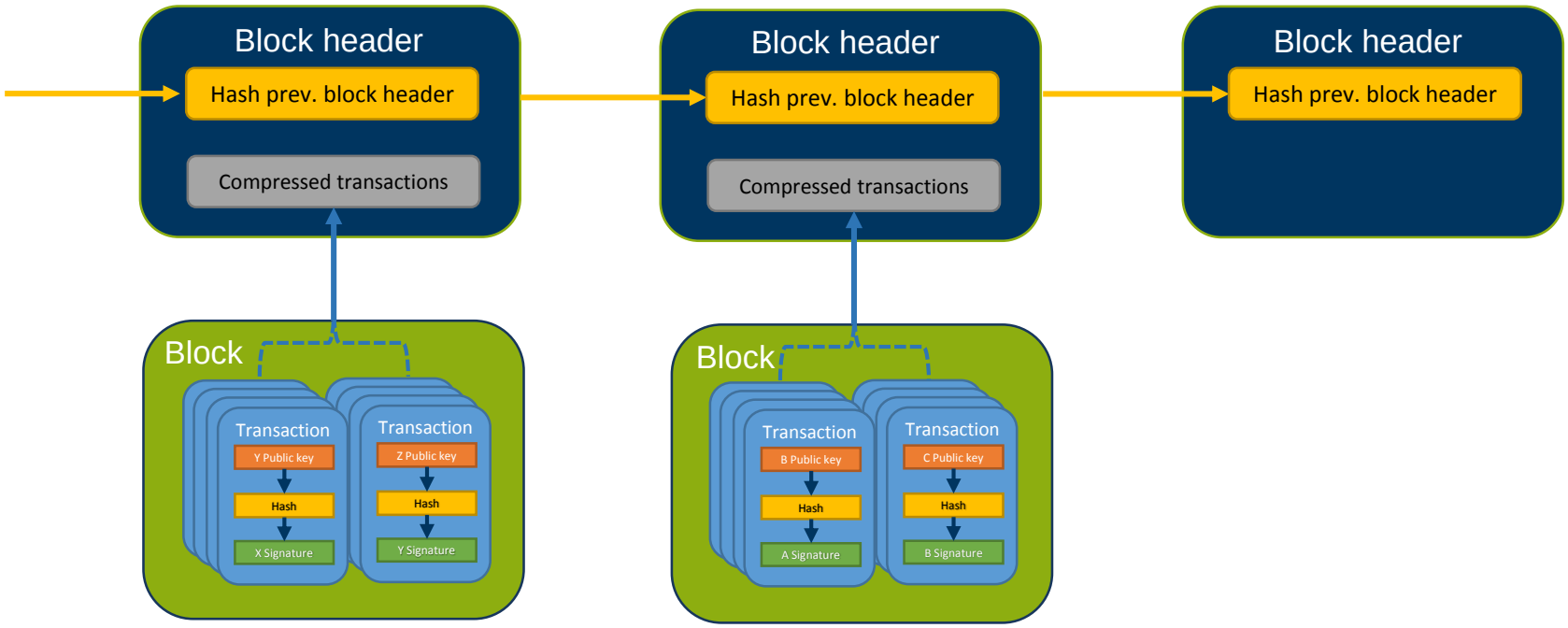
Bitcoin

Block chain - details



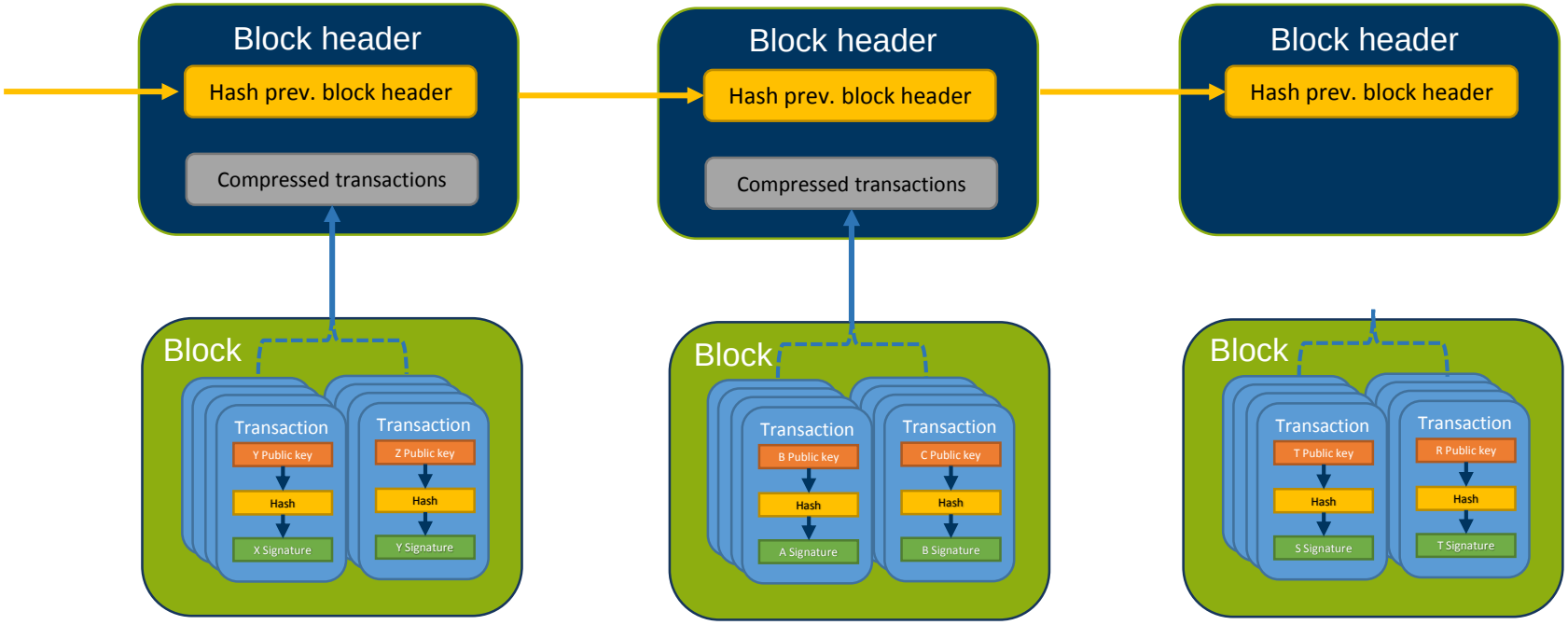
Bitcoin

Block chain - details



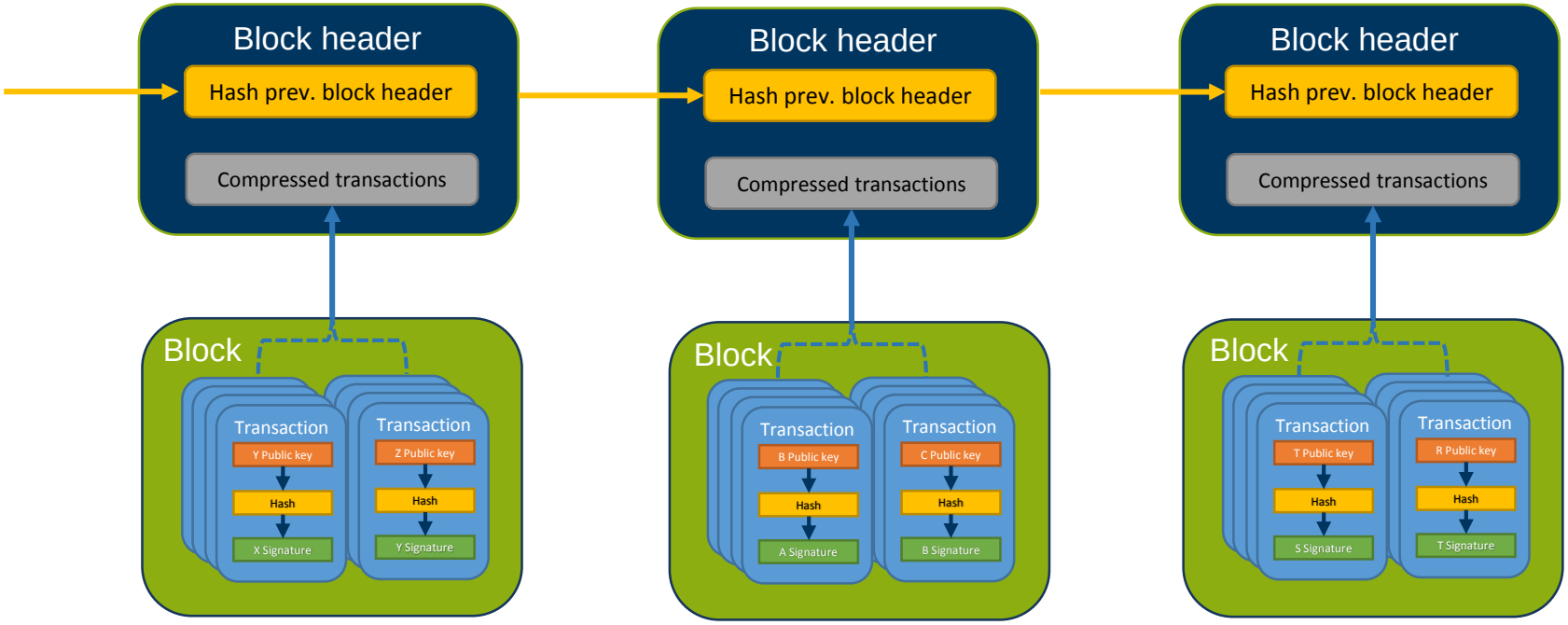
Bitcoin

Block chain - details



Bitcoin

Block chain - details



Bitcoin

Proof-of-Work

Bitcoin

Proof-of-Work

- Solving a computational moderately hard problem
- Solution allows to append a block to chain

Bitcoin

Proof-of-Work

- Solving a computational moderately hard problem
- Solution allows to append a block to chain
- $Hash^2$(MerkleTreeHash||…||nonce) < 0x00000000||FE32…9AB

Bitcoin

Proof-of-Work

- Solving a computational moderately hard problem
- Solution allows to append a block to chain
- $Hash^2$(MerkleTreeHash||…||nonce) < 0x00000000||FE32…9AB
- Genesis block: 0000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
- Recent block: 00000000000000000217084a9362676393abd073f4123a67b0a54b9cc86c0a03

Bitcoin

Proof-of-Work

- Solving a computational moderately hard problem
- Solution allows to append a block to chain
- $Hash^2$(MerkleTreeHash||…||nonce) < 0x00000000||FE32…9AB
- Genesis block: 0000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
- Recent block: 00000000000000000217084a9362676393abd073f4123a67b0a54b9cc86c0a03
- Reward for new block
- Creates a race

Bitcoin

Limitations

Bitcoin

Limitations

- Developer regulates block size

Bitcoin

Limitations

- Developer regulates block size
  - Limiting the number of transactions per block

Bitcoin vs Bitcoin cash

Bitcoin

Limitations

- Developer regulates block size
  - Limiting the number of transactions per block
- What if you want to change bitcoin's protocol
- Different hash algorithm

Bitcoin vs Bitcoin cash

Bitcoin

Limitations

- Developer regulates block size
  - Limiting the number of transactions per block
- What if you want to change bitcoin's protocol
- Different hash algorithm
- Etc.

Bitcoin vs Bitcoin cash

Bitcoin

Limitations

- Developer regulates block size
  - Limiting the number of transactions per block
- What if you want to change bitcoin's protocol
- Different hash algorithm
- Etc.

Bitcoin vs Bitcoin cash

Source code is public, let's modify it!

Altcoin

Altcoins

Overview

Altcoins

Overview

- Are modified versions of bitcoins source code

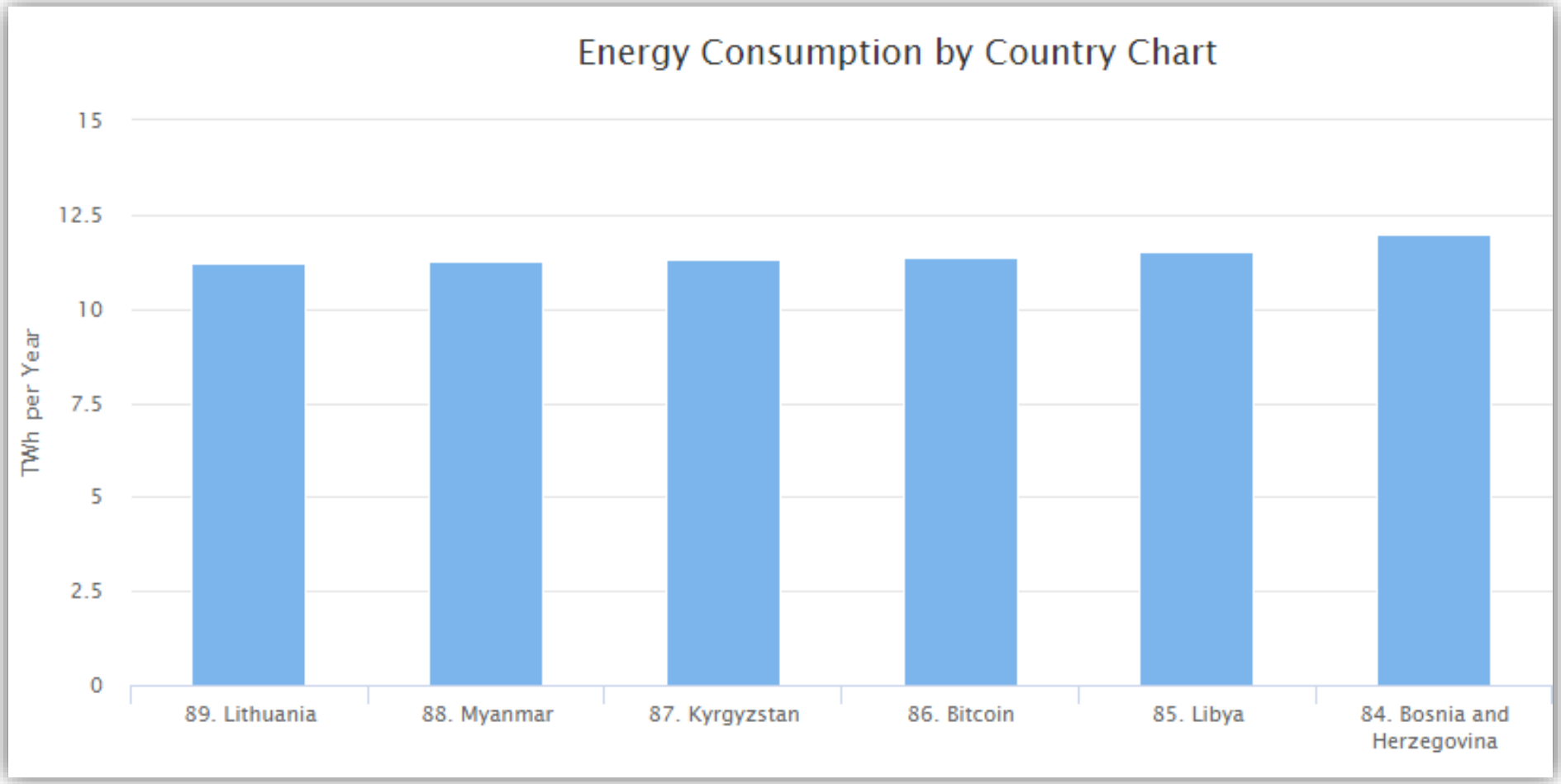


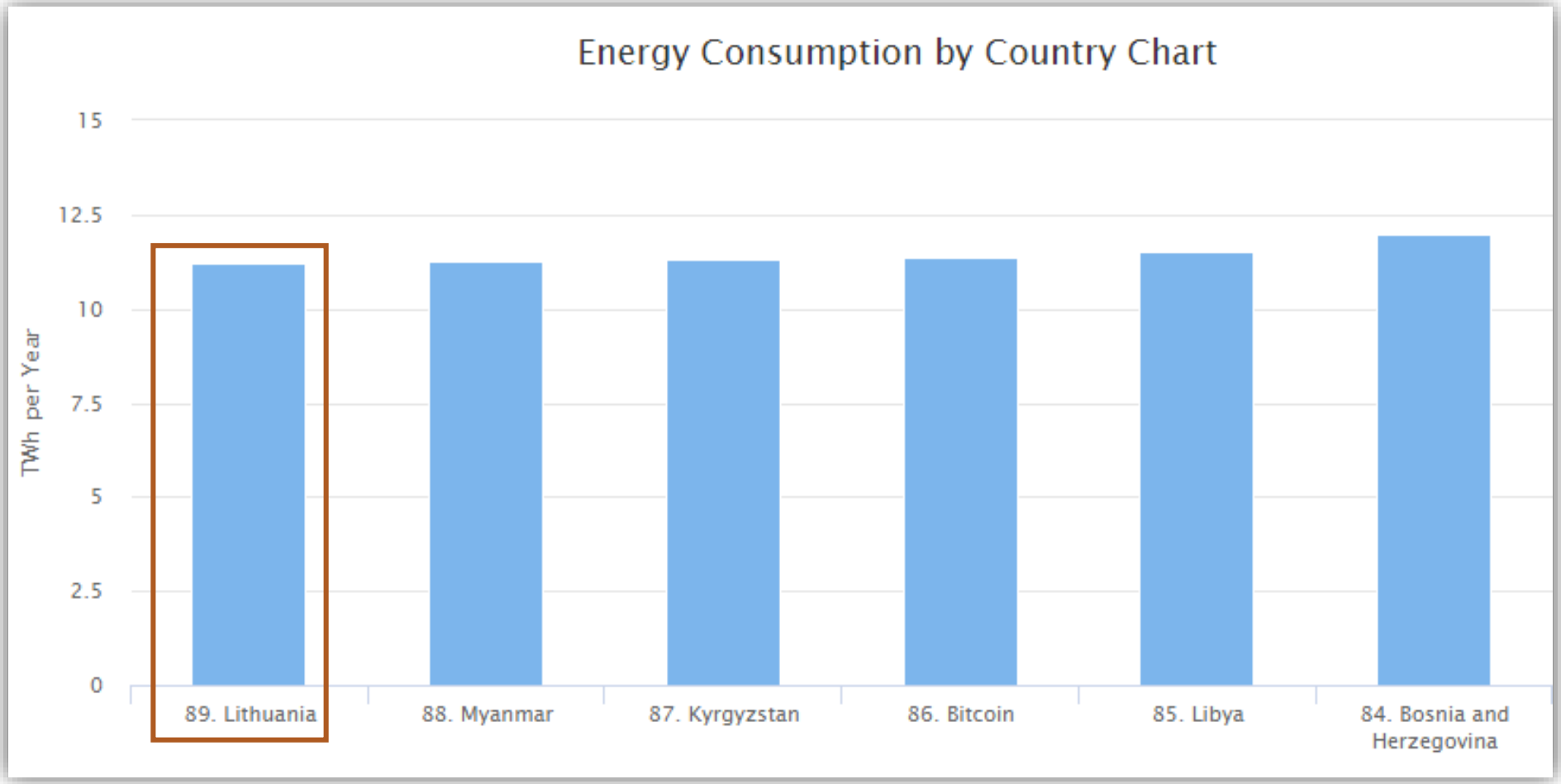
Altcoins

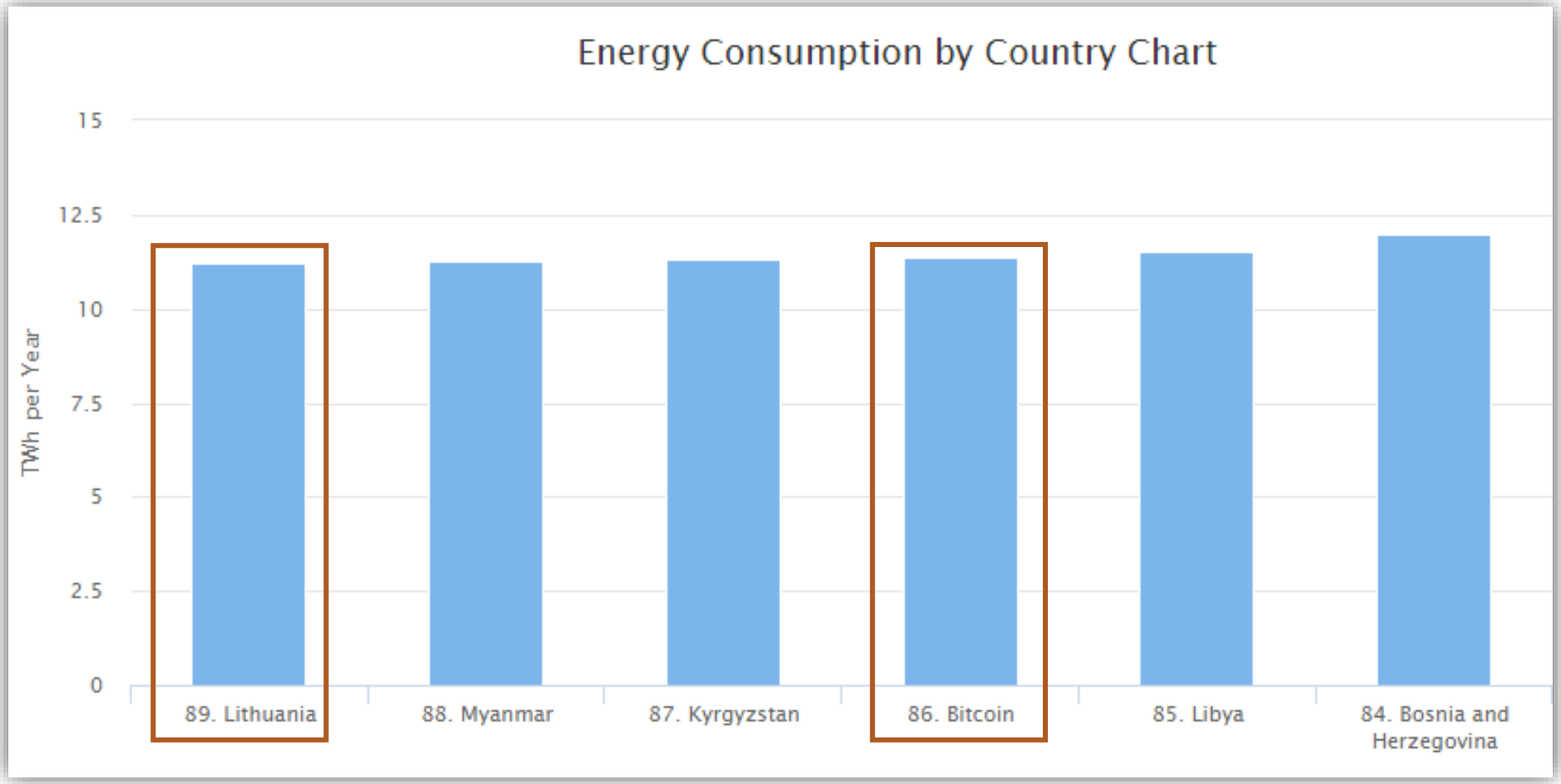
Overview

- Are modified versions of bitcoins source code
- All proof-of-work schemes waste energy




Energy Consumption by Country Chart
TWh per Year
15
12.5
10
7.5
5
2.5
0
89. Lithuania
88. Myanmar
87. Kyrgyzstan
86. Bitcoin
85. Libya
84. Bosnia and Herzegovina


Energy Consumption by Country Chart
TWh per Year
15
12.5
10
7.5
5
2.5
0
89. Lithuania
88. Myanmar
87. Kyrgyzstan
86. Bitcoin
85. Libya
84. Bosnia and Herzegovina


Energy Consumption by Country Chart
TWh per Year
15
12.5
10
7.5
5
2.5
0
89. Lithuania
88. Myanmar
87. Kyrgyzstan
86. Bitcoin
85. Libya
84. Bosnia and Herzegovina

Altcoins

Overview

- Are modified versions of bitcoins source code
- All proof-of-work schemes waste energy



Altcoins

Overview

- Are modified versions of bitcoins source code
- All proof-of-work schemes waste energy
- There are alternatives like coin age or activity



Altcoins

Overview

- Are modified versions of bitcoins source code
- All proof-of-work schemes waste energy
- There are alternatives like coin age or activity
- But can we make computations meaningful?
- First approaches:
  - Burstcoin – Storage space
  - DDosCoin – DoS = PoW (WOOT'16)
  - Namecoin – distributed DNS
  - Gridcoin – Utilize BOINC for meaningful computations



Gridcoin

Gridcoin

Overview

Gridcoin

Overview

- Actively developed since 2013

Gridcoin

Overview

- Actively developed since 2013
- In the TOP 100 of valuable coins
- Market cap. > 12 million euros

Gridcoin

Overview

- Actively developed since 2013
- In the TOP 100 of valuable coins
- Market cap. > 12 million euros
- Combines different existing techniques

Gridcoin

Overview

- Actively developed since 2013
- In the TOP 100 of valuable coins
- Market cap. > 12 million euros
- Combines different existing techniques
  - Peercoin – Coin possession instead PoW
  - BOINC – Meaningful computation
- Has computing power of 1.6 PetaFLOPS





Gridcoin

Overview

- Actively developed since 2013
- In the TOP 100 of valuable coins
- Market cap. > 12 million euros
- Combines different existing techniques
  - Peercoin – Coin possession instead PoW
  - BOINC – Meaningful computation
- Has computing power of 1.6 PetaFLOPS
- Ca. 80k lines of code (c++)
- No documentation ☹





Gridcoin

BOINC

Gridcoin

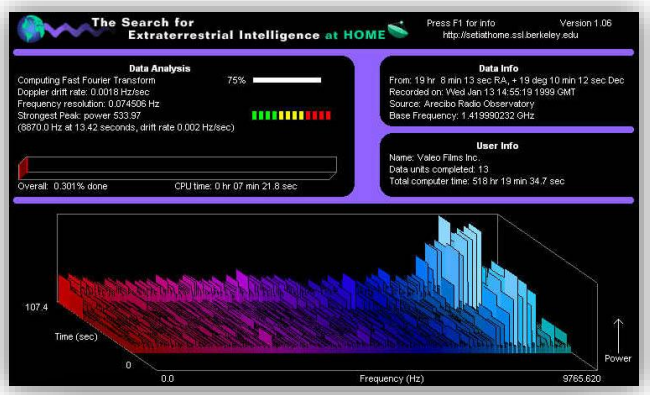
BOINC

- Software system
- Volunteer participate to computational projects

Gridcoin

BOINC

- Software system
- Volunteer participate to computational projects
- Solving chunks of scientific problems, e.g.:
  - Searching for extraterrestrial intelligence in radio signals (Seti@Home)

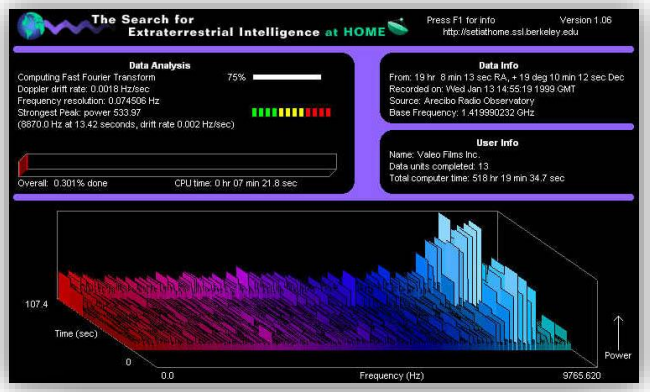




Gridcoin

BOINC

- Software system
- Volunteer participate to computational projects
- Solving chunks of scientific problems, e.g.:
  - Searching for extraterrestrial intelligence in radio signals (Seti@Home)
  - Protein folding (Folding@Home)
  - Find prime numbers (PrimeGrid)

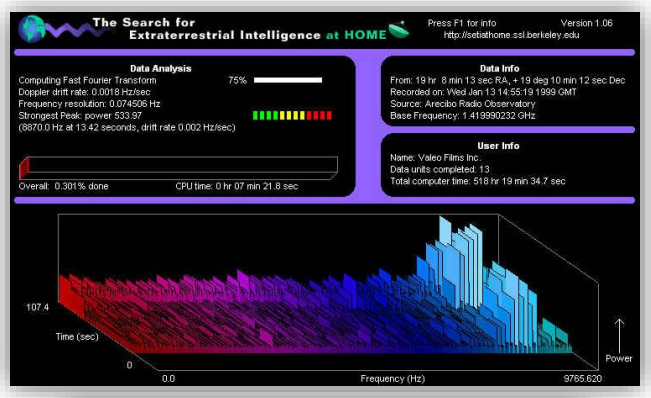




Gridcoin

BOINC

- Software system
- Volunteer participate to computational projects
- Solving chunks of scientific problems, e.g.:
  - Searching for extraterrestrial intelligence in radio signals (Seti@Home)
  - Protein folding (Folding@Home)
  - Find prime numbers (PrimeGrid)





Open problem, how to efficient add meaningful computation to the block chain.

Gridcoin

BOINC - Details

- Correct solution
  - $PID = MD_5(rand||email)$
- Statistical websites list #credit for every CPID

Gridcoin

BOINC - Details

- Correct solution →
  - $PID = MD_5(rand||email)$
- Statistical websites list #credit for every CPID

Gridcoin

BOINC - Details

- Correct solution → volunteer gets reward
  - $PID = MD_5(rand||email)$
- Statistical websites list #credit for every CPID

Gridcoin

BOINC - Details

- Correct solution → volunteer gets reward
- Reward = *credits* (imaginary)
- Increases competition
  - $PID = MD_5(rand||email)$
- Statistical websites list #credit for every CPID

Gridcoin

MD **5** $MMDD$ MD **5** $5\ MD\ 5\ (rraannddeemmaaiill)$

$BOINC$ **- Details** → volunteer gets reward

- Correct solution
- Reward = *credits* (imaginary)
- Increases competition
- Every volunteer has a unique ID
  - CPID (cross project id)
  - $CPID = MD_5(rand||email)$
- Statistical websites list #credit for every CPID

Gridcoin

***MD* 5 *MMDD MD* 5** *5 MD 5 (rraannddllee mmaaiill)*

BOINC **- Details** → volunteer gets reward

- Correct solution
- Reward = *credits* (imaginary)
- Increases competition
- Every volunteer has a unique ID
  - CPID (cross project id)
- Statistical websites list #credit for every CPIDStatistical websites list #credit for every CPID

Gridcoin

***MD* 5 *MMDD MD* 5** *5 MD 5 (rraannddllemmaaiill)*

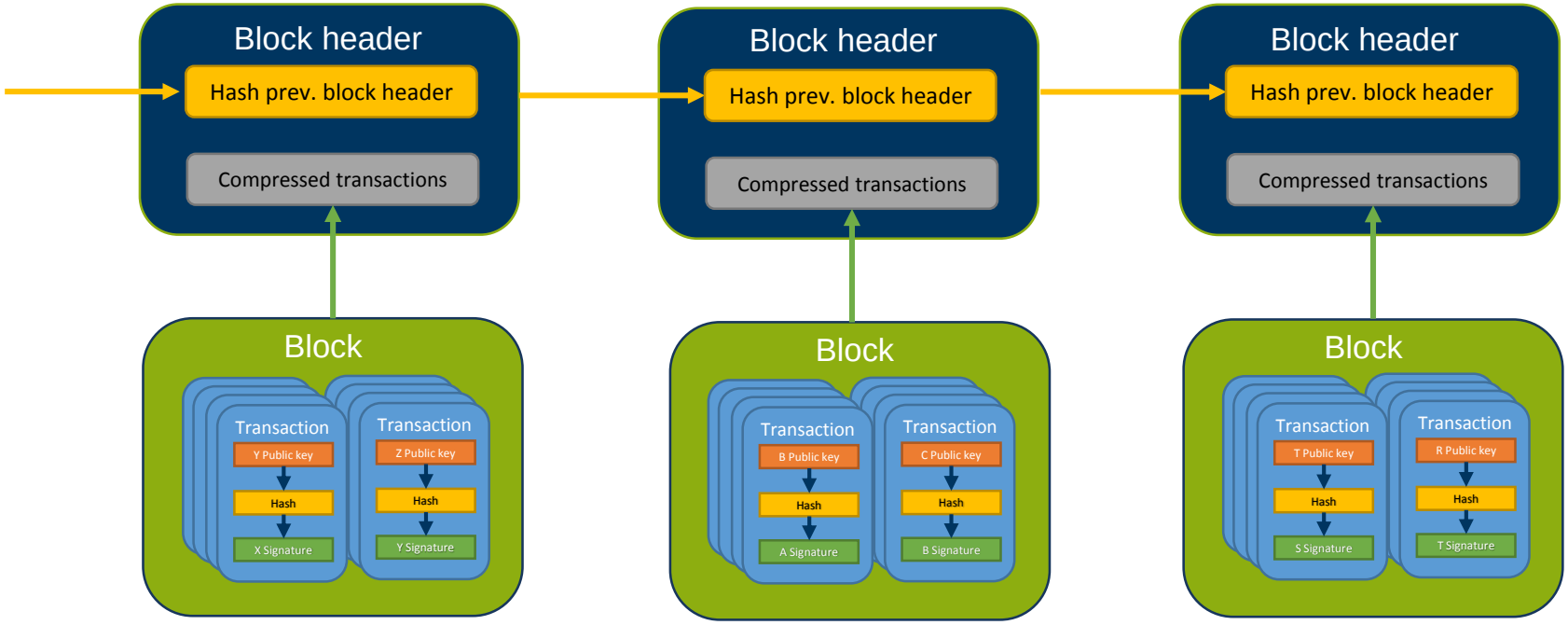
BOINC **- Details** → volunteer gets reward

- Correct solution
- Reward = *credits* (imaginary)
- Increases competition
- Every volunteer has a unique ID
  - CPID (cross project id)
- Statistical websites list #credit for every CPIDStatistical websites list #credit for every CPID

Why not transform *credits* to coins?

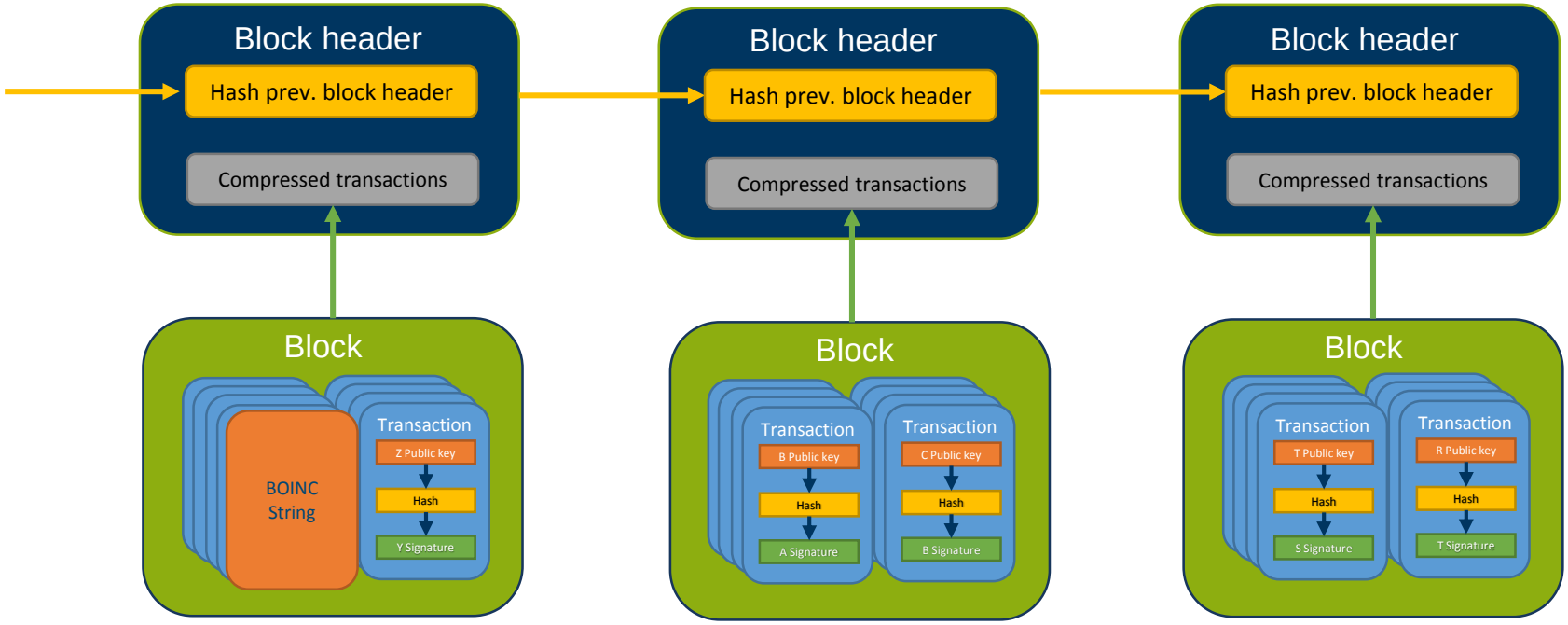
Gridcoin

Block chain – BOINC integration



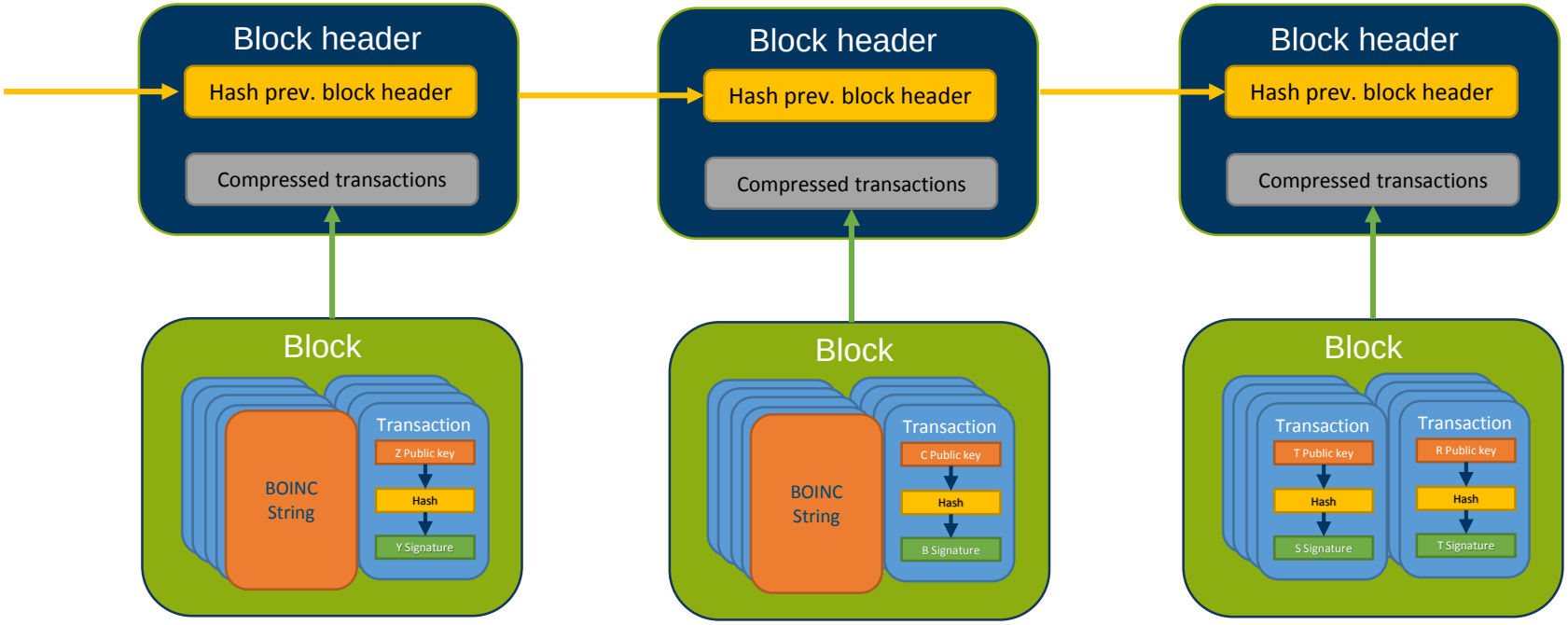
Gridcoin

Block chain – BOINC integration



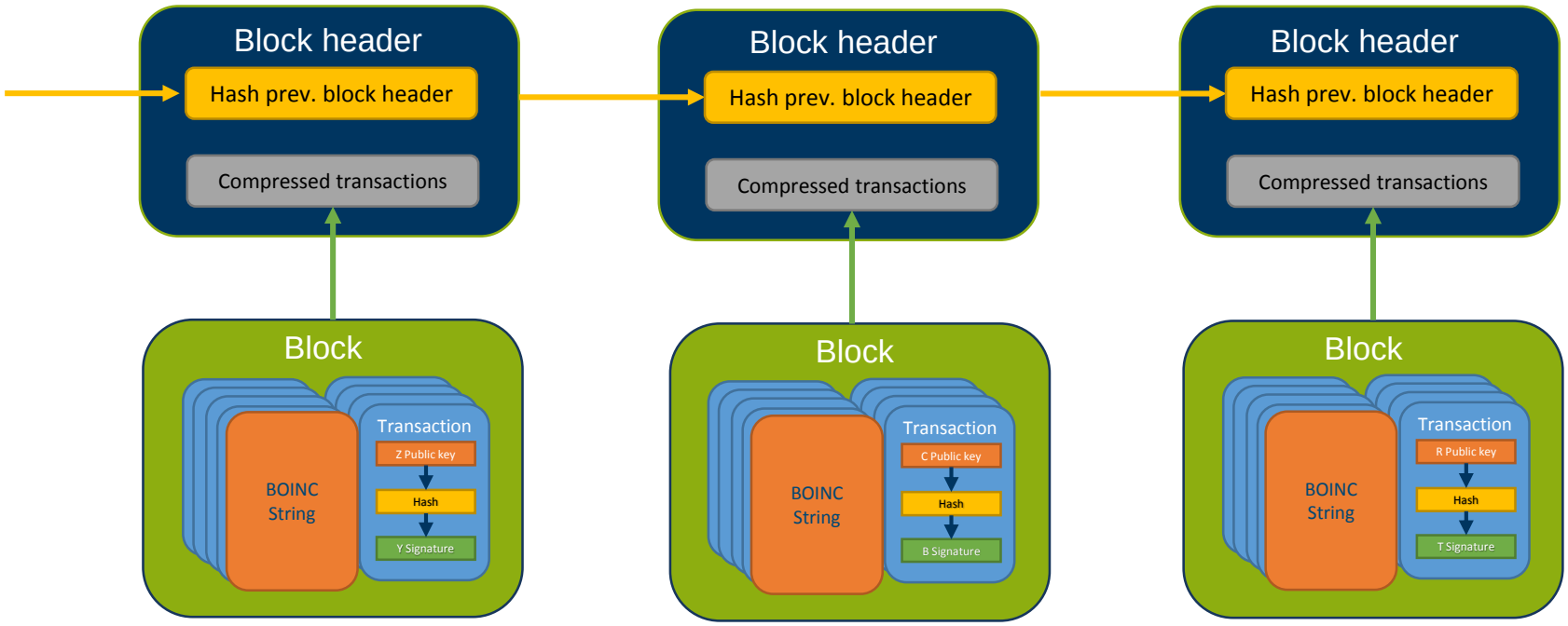
Gridcoin

Block chain – BOINC integration



Gridcoin

Block chain – BOINC integration

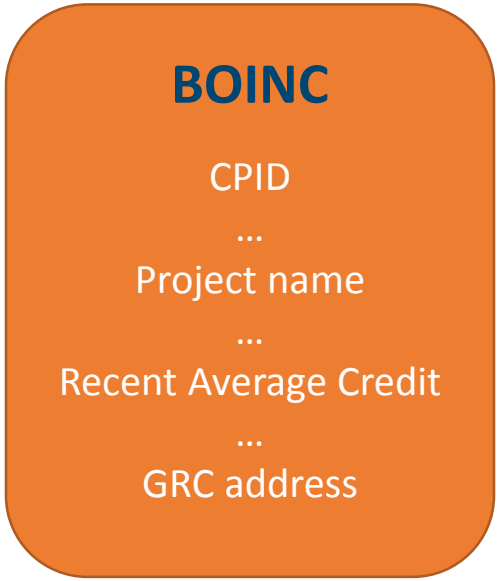


Gridcoin

Block chain – BOINC integration

Gridcoin

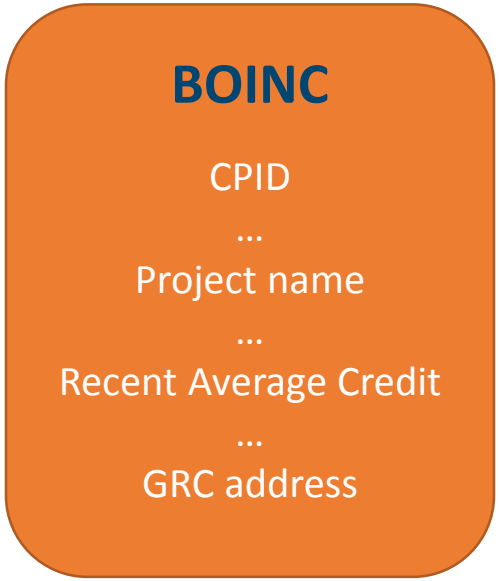
Block chain – BOINC integration



Gridcoin

Block chain – BOINC integration

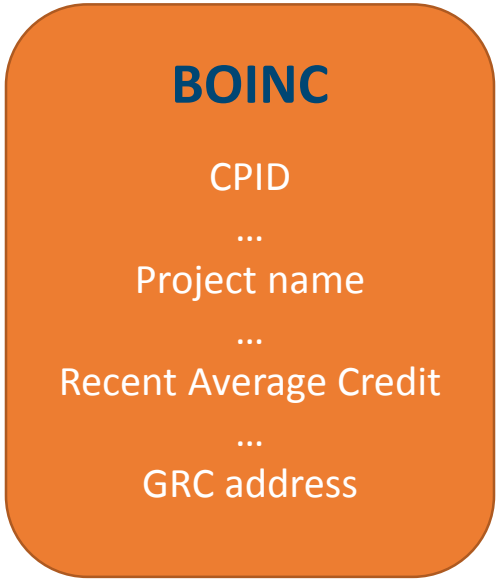
- BOINC volunteers = *researchers*
- Without BOINC = *investor*
- 2 possibilities to gain reward



Gridcoin

Block chain – BOINC integration

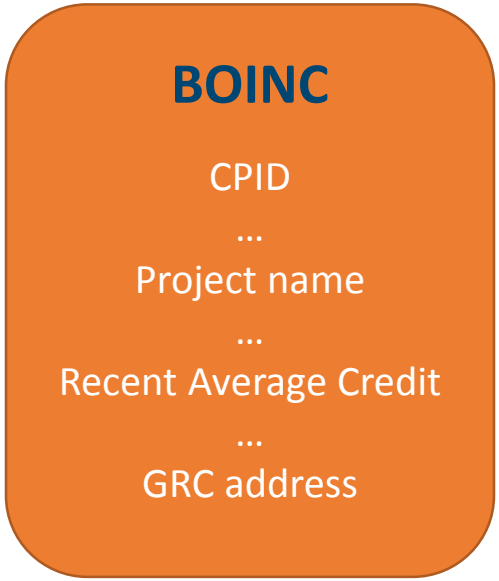
- BOINC volunteers = *researchers*
- Without BOINC = *investor*
- 2 possibilities to gain reward
  - Proof-of-Stake (coin possession)
    - Reward without BOINC = 16,68 coins



Gridcoin

Block chain – BOINC integration

- BOINC volunteers = *researchers*
- Without BOINC = *investor*
- 2 possibilities to gain reward
  - Proof-of-Stake (coin possession)
    - Reward without BOINC = 16,68 coins
  - Proof-of-Research
    - Reward with BOINC = 16,68 + X coins
      - 0 < X < 50 based on BOINC work done



Gridcoin

BOINC / Gridcoin infrastructure

Gridcoin block chain

Gridcoin users

Gridcoin

BOINC / Gridcoin infrastructure

Gridcoin block chain

Gridcoin users





Gridcoin

BOINC / Gridcoin infrastructure

BOINC project A

Statistical website

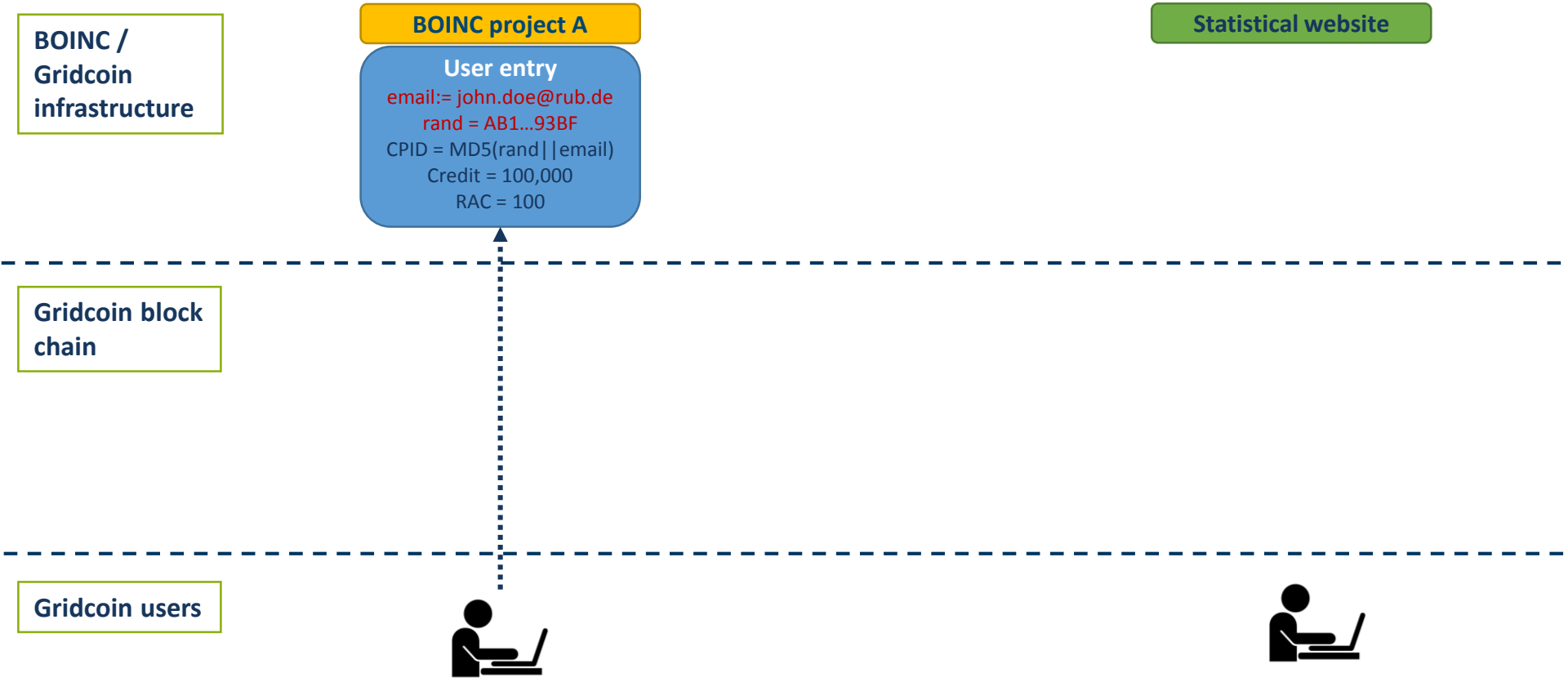
Gridcoin block chain

Gridcoin users

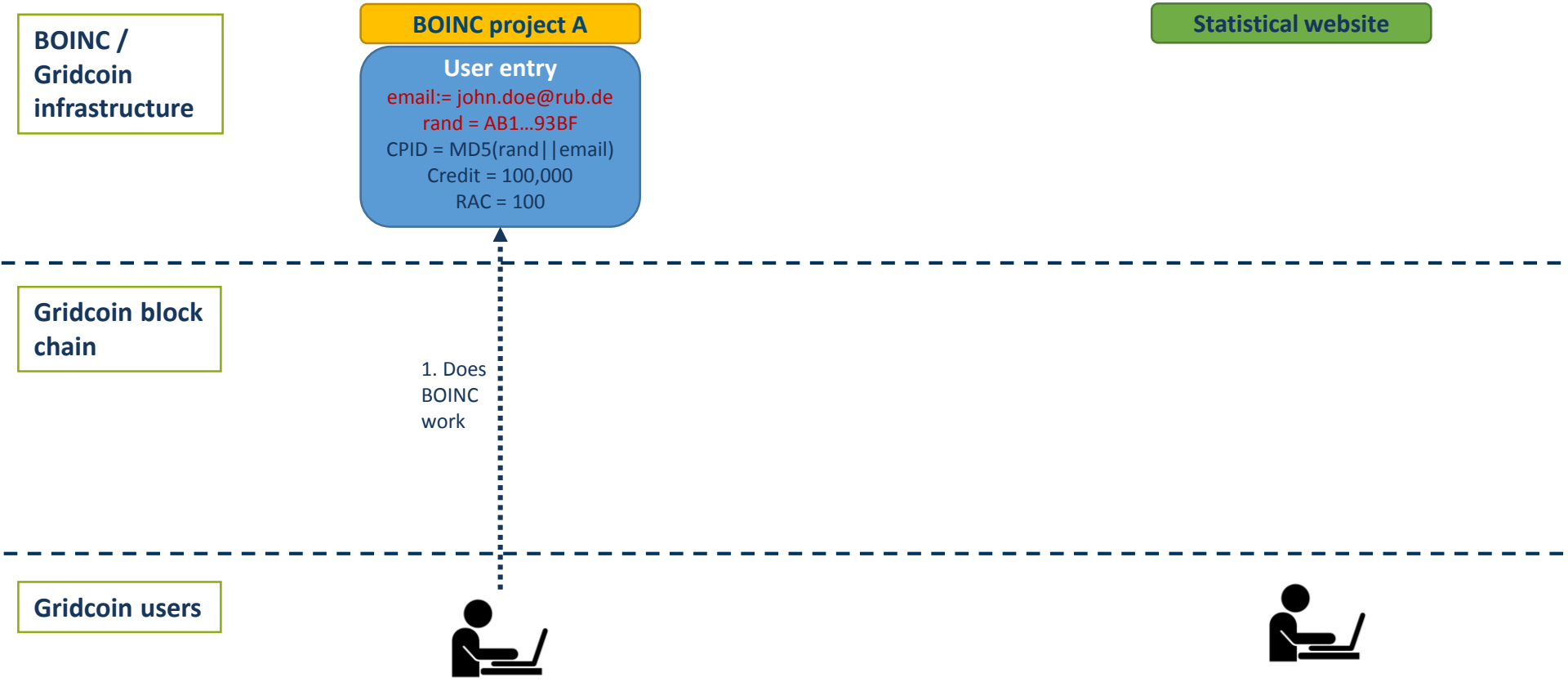




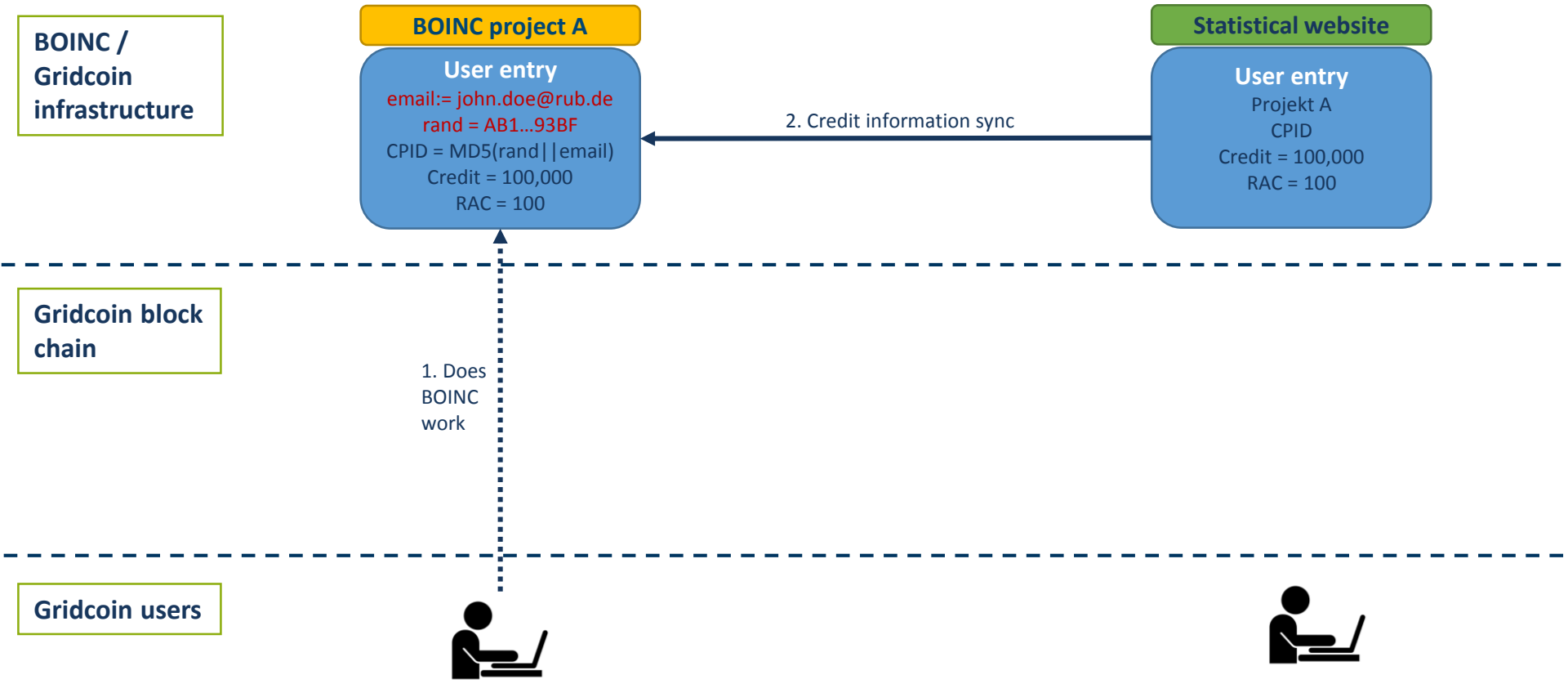
Gridcoin


BOINC / Gridcoin infrastructure
BOINC project A
User entry
email:= john.doe@rub.de
rand = AB1…93BF
CPID = MD5(rand||email)
Credit = 100,000
RAC = 100
Statistical website
Gridcoin block chain
Gridcoin users

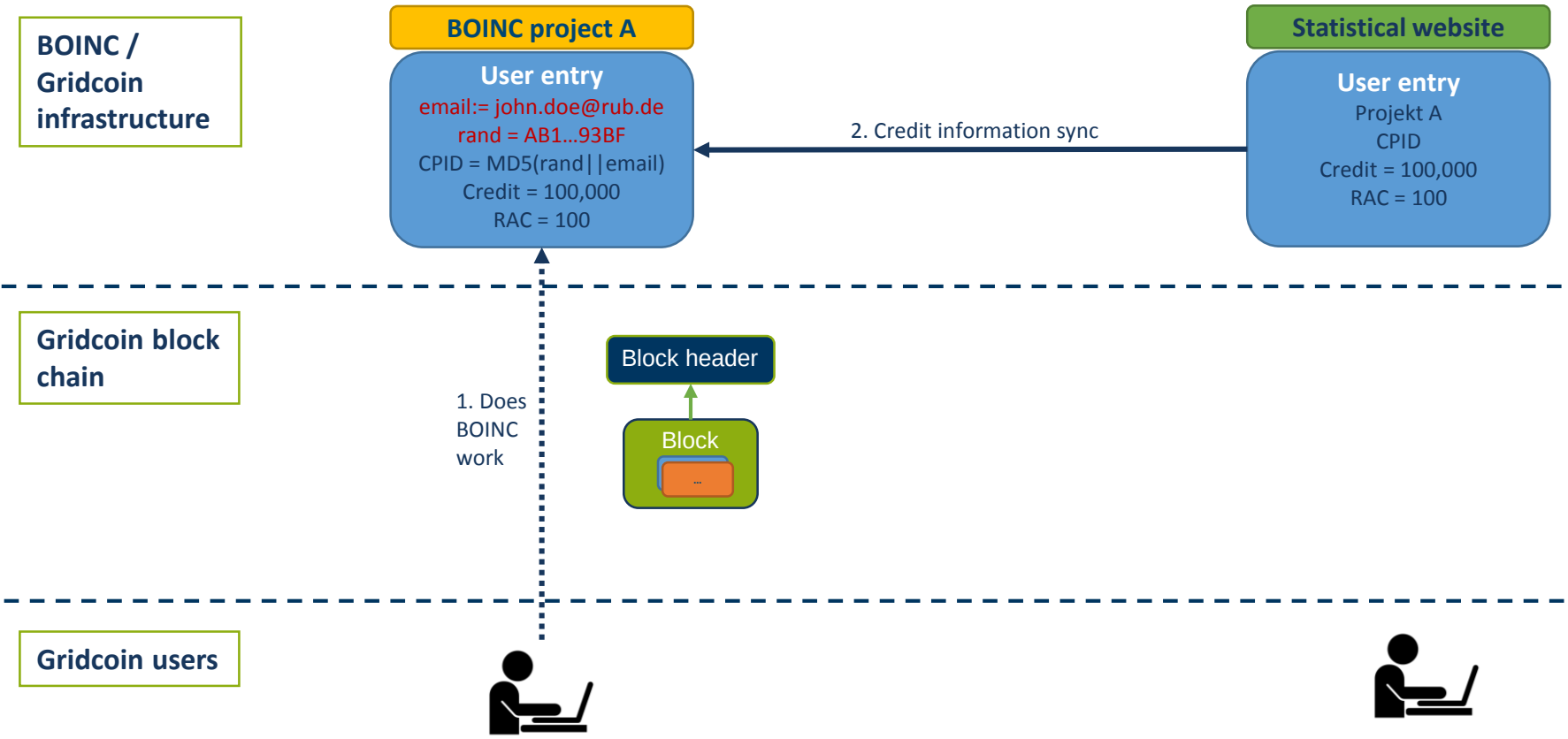
Gridcoin


BOINC / Gridcoin infrastructure
BOINC project A
User entry
email:= john.doe@rub.de
rand = AB1…93BF
CPID = MD5(rand||email)
Credit = 100,000
RAC = 100
Statistical website
Gridcoin block chain
1. Does BOINC work
Gridcoin users

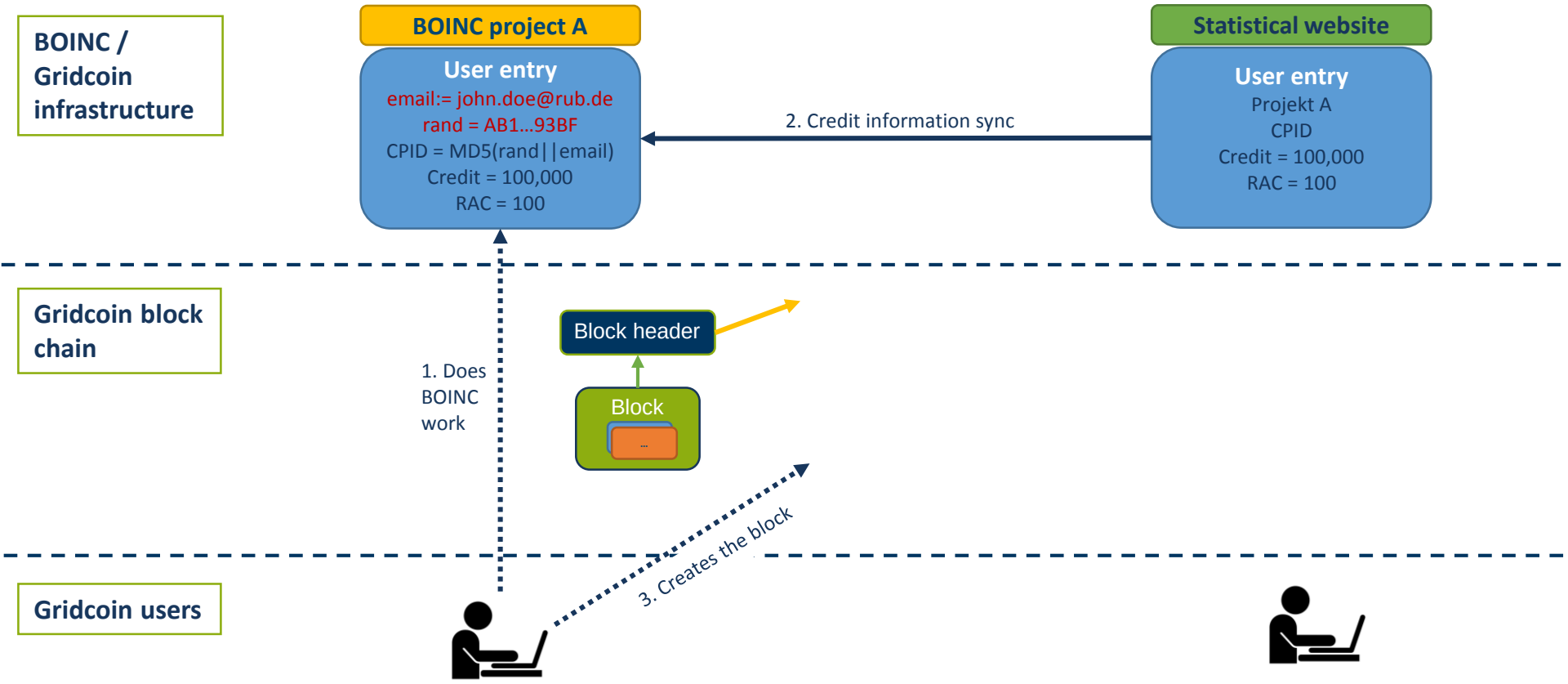
Gridcoin


BOINC / Gridcoin infrastructure
BOINC project A
User entry
email:= john.doe@rub.de
rand = AB1…93BF
CPID = MD5(rand||email)
Credit = 100,000
RAC = 100
2. Credit information sync
Statistical website
User entry
Projekt A
CPID
Credit = 100,000
RAC = 100
Gridcoin block chain
1. Does BOINC work
Gridcoin users

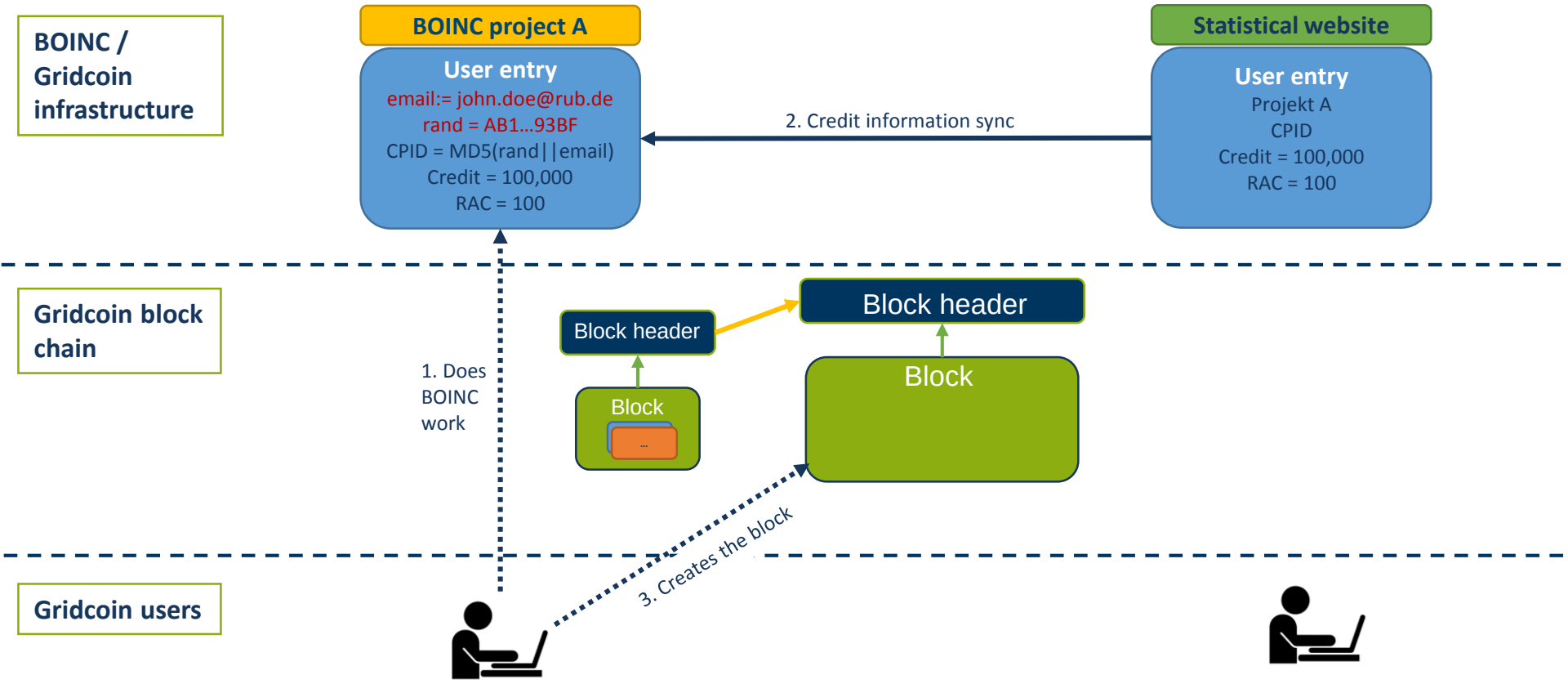
Gridcoin


BOINC / Gridcoin infrastructure
BOINC project A
User entry
email:= john.doe@rub.de
rand = AB1…93BF
CPID = MD5(rand||email)
Credit = 100,000
RAC = 100
2. Credit information sync
Statistical website
User entry
Projekt A
CPID
Credit = 100,000
RAC = 100
Gridcoin block chain
1. Does BOINC work
Block header
Block
Gridcoin users

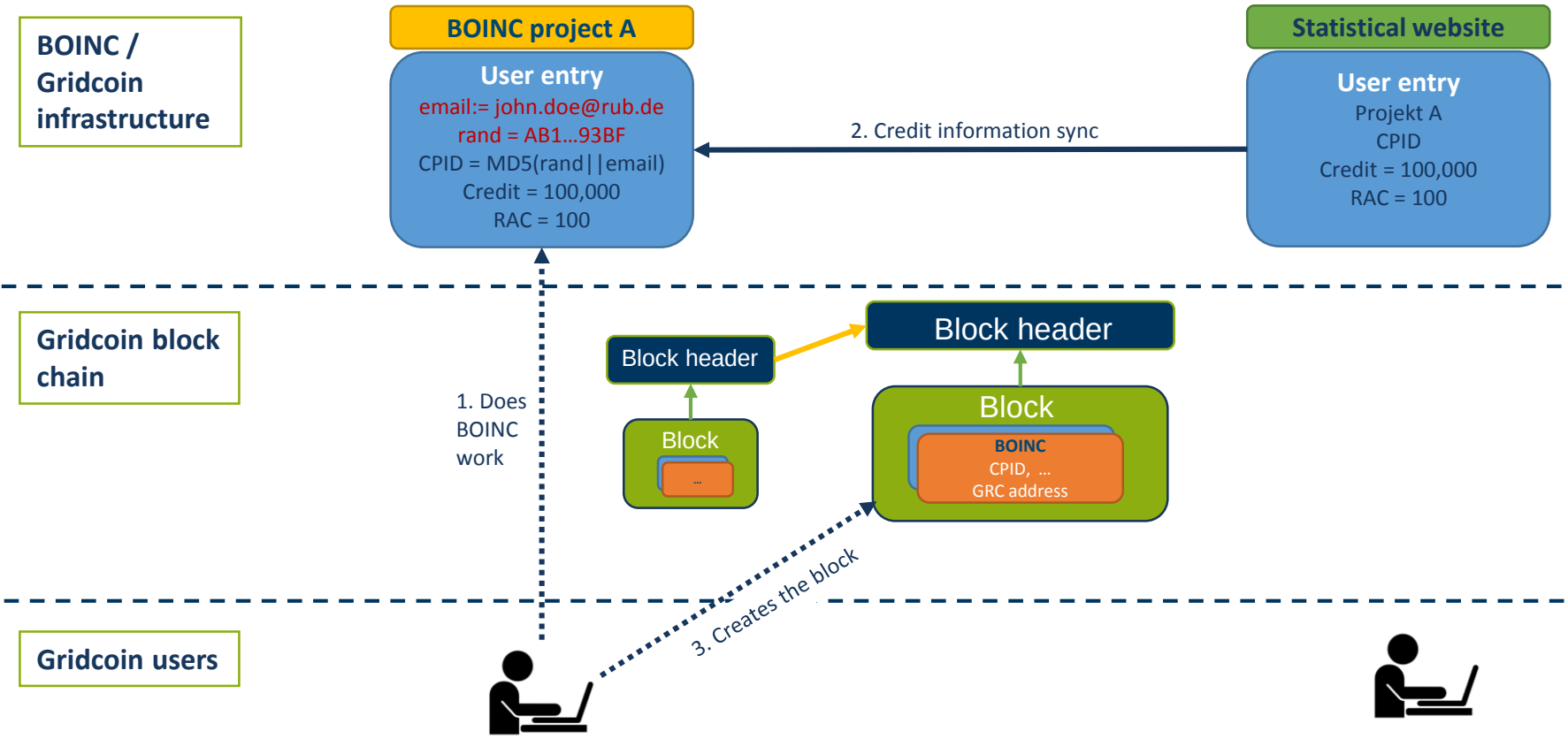
Gridcoin


BOINC / Gridcoin infrastructure
BOINC project A
User entry
email:= john.doe@rub.de
rand = AB1…93BF
CPID = MD5(rand||email)
Credit = 100,000
RAC = 100
2. Credit information sync
Statistical website
User entry
Projekt A
CPID
Credit = 100,000
RAC = 100
Gridcoin block chain
1. Does BOINC work
Block header
Block
...
3. Creates the block
Gridcoin users

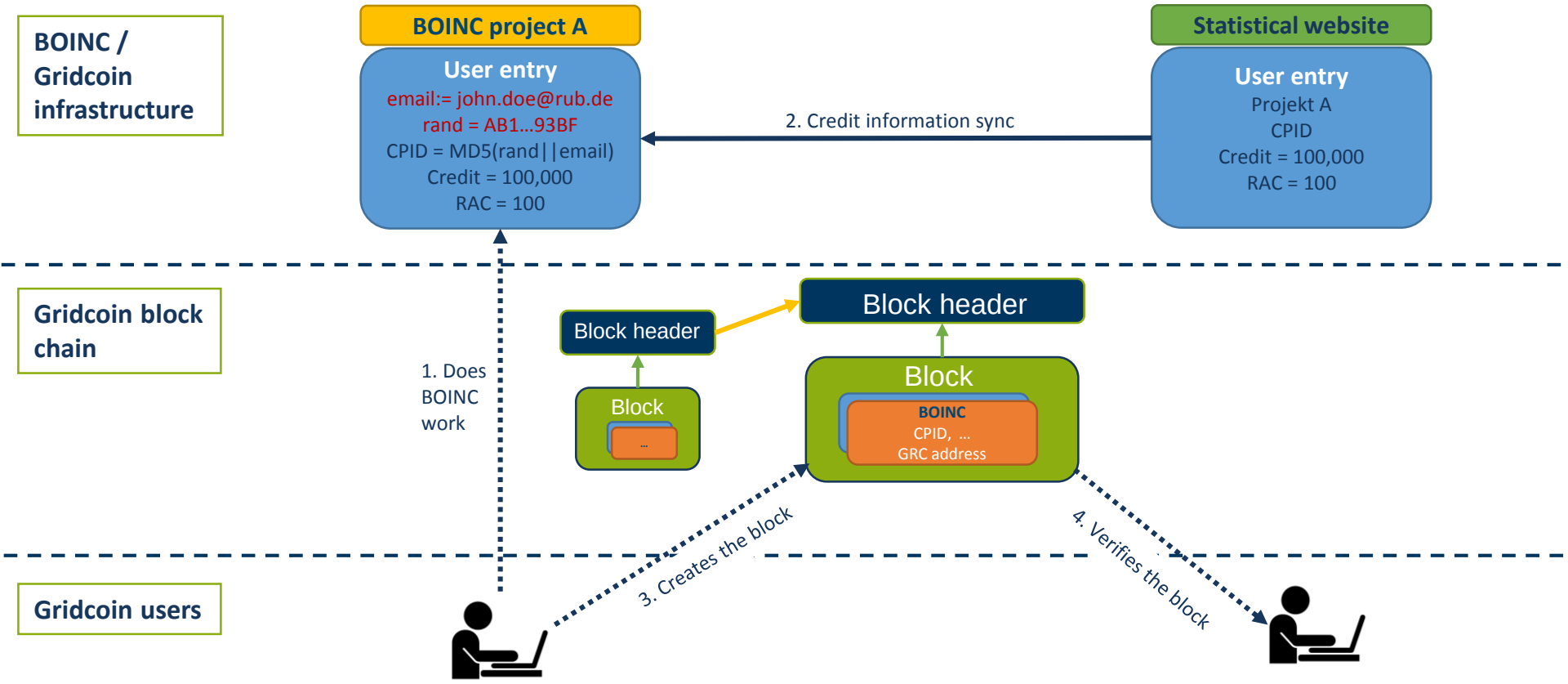
Gridcoin


BOINC / Gridcoin infrastructure
BOINC project A
User entry
email:= john.doe@rub.de
rand = AB1…93BF
CPID = MD5(rand||email)
Credit = 100,000
RAC = 100
2. Credit information sync
Statistical website
User entry
Projekt A
CPID
Credit = 100,000
RAC = 100
Gridcoin block chain
1. Does BOINC work
Block header
Block
Block header
Block
3. Creates the block
Gridcoin users

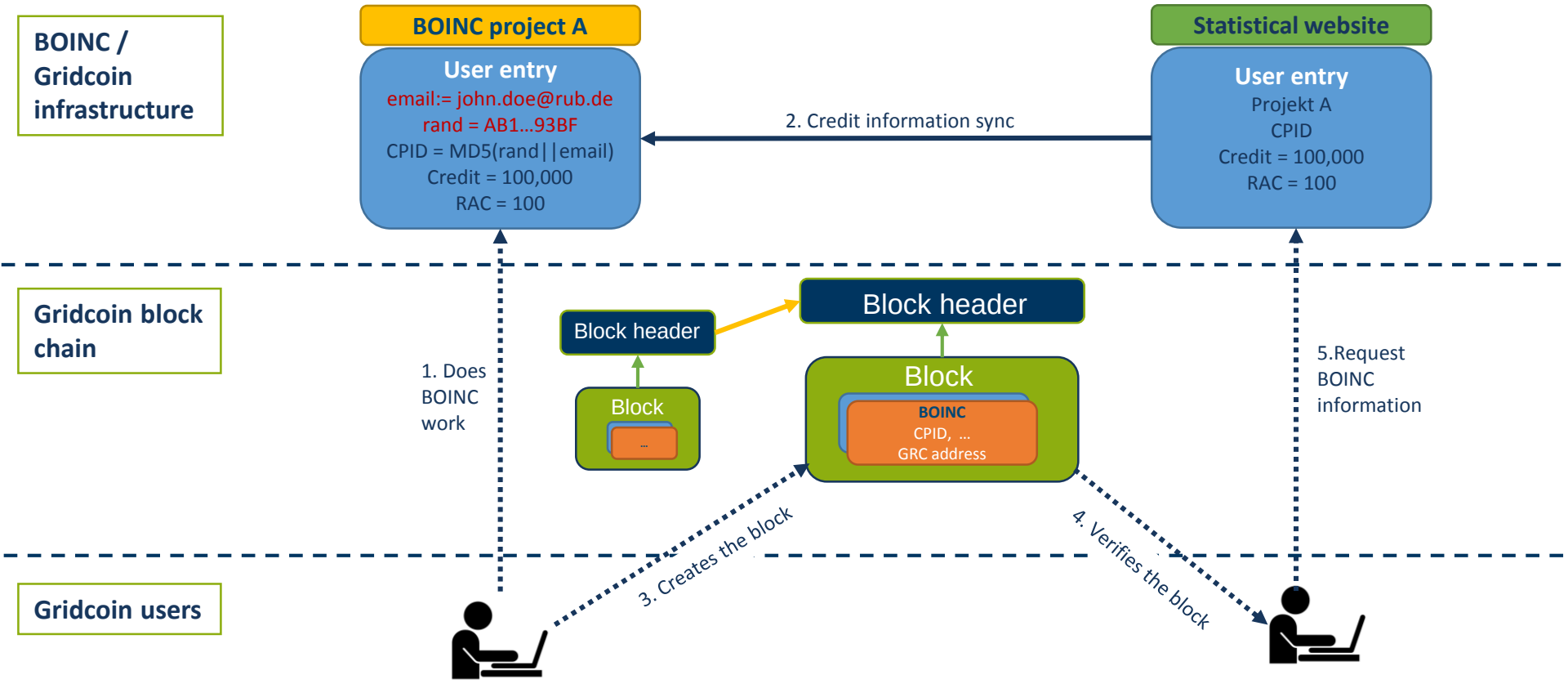
Gridcoin


BOINC / Gridcoin infrastructure
BOINC project A
User entry
email:= john.doe@rub.de
rand = AB1…93BF
CPID = MD5(rand||email)
Credit = 100,000
RAC = 100
2. Credit information sync
Statistical website
User entry
Projekt A
CPID
Credit = 100,000
RAC = 100
Gridcoin block chain
1. Does BOINC work
Block header
Block
...
Block header
Block
BOINC
CPID, …
GRC address
3. Creates the block
Gridcoin users

Gridcoin


BOINC / Gridcoin infrastructure
BOINC project A
User entry
email:= john.doe@rub.de
rand = AB1…93BF
CPID = MD5(rand||email)
Credit = 100,000
RAC = 100
2. Credit information sync
Statistical website
User entry
Projekt A
CPID
Credit = 100,000
RAC = 100
Gridcoin block chain
1. Does BOINC work
Block header
Block
...
Block header
Block
BOINC
CPID, …
GRC address
3. Creates the block
4. Verifies the block
Gridcoin users

Gridcoin


BOINC / Gridcoin infrastructure
BOINC project A
User entry
email:= john.doe@rub.de
rand = AB1…93BF
CPID = MD5(rand||email)
Credit = 100,000
RAC = 100
2. Credit information sync
Statistical website
User entry
Projekt A
CPID
Credit = 100,000
RAC = 100
Gridcoin block chain
1. Does BOINC work
Block header
Block
...
Block header
Block
BOINC
CPID, …
GRC address
5.Request BOINC information
3. Creates the block
4. Verifies the block
Gridcoin users

The Challange

Identity proof

The Challange

Identity proof

- Problem:
  - Copy CPID from block chain

The Challange

Identity proof

- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers

The Challange

Identity proof

- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
- How to prove miner = researcher?

The Challange

Identity proof

- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
- How to prove miner = researcher?
- Gridcoins solution:

The Challange

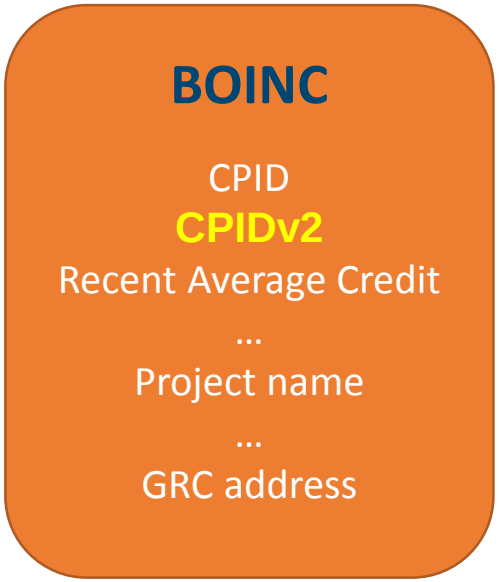
Identity proof

- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
- How to prove miner = researcher?
- Gridcoins solution: CPIDv2

The Challange

Identity proof

- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
- How to prove miner = researcher?
- Gridcoins solution: CPIDv2



The Challange

Identity proof

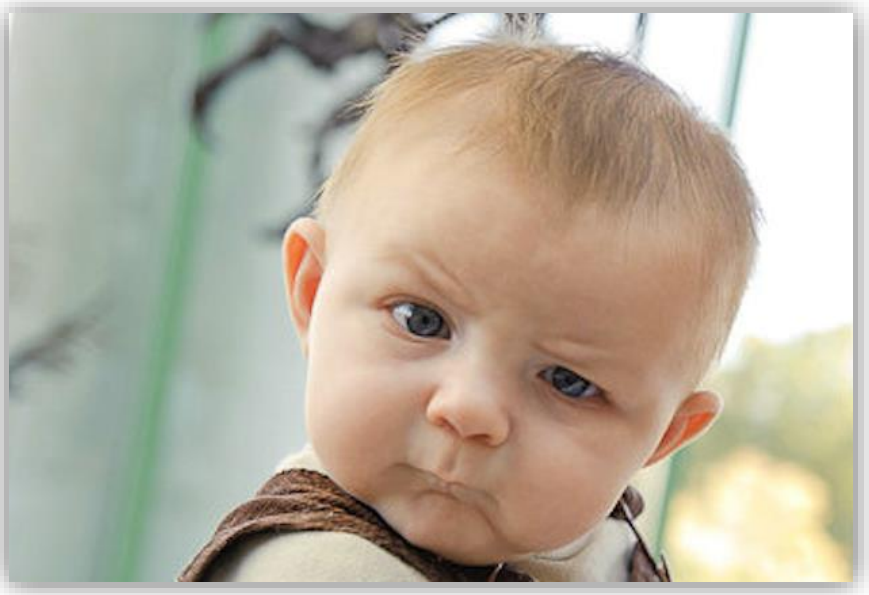
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
- How to prove miner = researcher?
- Gridcoins solution: CPIDv2

```
(0x1bb+1236-0x68d)]+=c;state[(0x92f+6920-0x2434)]+=d;
memset(x,(0x123d+2605-0x1c6a),sizeof x);}


void CPID::update(const unsigned char input[],size_type length){
size_type index=count[(0xbaf+535-0xdc6)]/(0x1609+3405-0x234e)%blocksize;
if((count[(0x12f6+3141-0x1f3b)]+=(length<<(0x1894+3130-0x24cb)))<(length<<
(0x902+5767-0x1f86))count[(0xed4+4990-0x2251)]++;count[(0x1410+3546-0x21e9)]+=(
length>>(0x1325+1183-0x17a7));
size_type firstpart=(0xcf2+5399-0x21c9)-index;size_type i;
if(length>=firstpart){
memcpy(&buffer[index],input,firstpart);transform(buffer);
for(i=firstpart;i+blocksize<=length;i+=blocksize)transform(&input[i]);index=
(0x786+6933-0x229b);}else i=(0x3c9+6892-0x1eb5);
memcpy(&buffer[index],&input[i],length-i);}

void CPID::update(const char input[],size_type length){update((const unsigned
char*)input,length);}int HexToByte(std::string hex){int x=(0x4a+4863-0x1349);std
::stringstream ss;ss<<std::hex<<hex;ss>>x;return x;}int ROL(std::string
blockhash,int iPos,std::string hash,int hexpos){std::string cpid3="";if(iPos<=(
int)hash.length()-(0x1c97+497-0x1e87)){std::string hex=hash.substr(iPos,
(0xa5d+6424-0x2373));int rorcount=BitwiseCount(blockhash,hexpos);int b=HexToByte
(hex)-rorcount;if(b>=(0x5b2+1768-0xc9a)){return b;}}return HexToByte("\x30\x30")
;}std::string CPID::Update6(std::string non_finalized,uint256 block_hash){std::
string shash=HashHex(block_hash);for(int i=(0x1258+3278-0x1f26);i<(int)
boinc_hash_new.length();i++){non_finalized+=ROR(shash,i,boinc_hash_new);}return
non_finalized;}std::string Update7(std::string longcpid,uint256 hash_block){std
::string shash=HashHex(hash_block);int hexpos=(0x632+1664-0xcb2);std::string
non_finalized="";for(int i1=(0xbd+8943-0x23ac);i1<(int)longcpid.length();i1=i1+
(0x1ac0+2812-0x25ba)){non_finalized+=ROL(shash,i1,longcpid,hexpos);hexpos++;}
CPID c7=CPID(non_finalized);std::string hexstring=c7.hexdigest();return
hexstring;}void CPID::update5(std::string longcpid,uint256 hash_block){std::
string shash=HashHex(hash_block);int hexpos=(0x43b+6491-0x1d96);unsigned char*
input=new unsigned char[(longcpid.length()/(0x534+8432-0x2622))+
(0xa19+5289-0x1ec1)];for(int i1=(0x1348+1245-0x1825);i1<(int)longcpid.length();
i1=i1+(0x478+5707-0x1ac1)){input[hexpos]=ROL(shash,i1,longcpid,hexpos);hexpos++;
}input[longcpid.length()/(0x7ac+7506-0x24fc)+(0xba+3605-0xece)]=
(0x3c3+5976-0x1b1b);size_type length=longcpid.length()/(0x1186+4729-0x23fd);
size_type index=count[(0xf02+2743-0x19b9)]/(0x174+6416-0x1a7c)%blocksize;
if((count[(0x1eb3+1439-0x2452)]+=(length<<(0x89c+7639-0x2670)))<(length<<
(0x1f07+935-0x22ab)))count[(0x488+7340-0x2133)]++;count[(0xb5+5597-0x1691)]+=(
length>>(0x374+3554-0x1139));
size_type firstpart=(0x195+7412-0x1e49)-index;size_type i;
if(length>=firstpart){
memcpy(&buffer[index],input,firstpart);transform(buffer);
for(i=firstpart;i+blocksize<=length;i+=blocksize)transform(&input[i]);index=
(0x144f+4713-0x26b8);}else i=(0x1053+3145-0x1c9c);
memcpy(&buffer[index],&input[i],length-i);}


CPID&CPID::finalize(){static unsigned char padding[(0xbf3+5821-0x2270)]={
(0x14b4+2295-0x1d2b),(0xd2+9568-0x2632),(0xa60+6685-0x247d),(0x1749+2154-0x1fb3)
,(0xb80+3428-0x18e4),(0x9d+9832-0x2705),(0x6d5+1494-0xcab),(0x714+3-0x717),
(0x13bf+4470-0x2535),(0x1a98+1182-0x1f36),(0x1398+3220-0x202c),
(0x50f+1352-0xa57),(0x97c+2239-0x123b),(0xa12+2843-0x152d),(0x934+3404-0x1680),
(0xb50+4854-0x1e46),(0x1166+928-0x1506),(0x1def+2302-0x26ed),(0xe84+1817-0x159d)
,(0x205+8670-0x23e3),(0x1311+2117-0x1b56),(0x23fa+91-0x2455),(0xec0+2817-0x19c1)
,(0x12b6+5050-0x2670),(0x141c+2938-0x1f96),(0x1c2b+389-0x1db0),
(0x8b6+3730-0x1748),(0x1274+509-0x1471),(0xc82+2063-0x1491),(0x15e6+916-0x197a),
(0x9b7+6733-0x2404),(0x1463+4638-0x2681),(0x48c+4237-0x1519),(0x6d3+8197-0x26d8)
,(0x1531+3021-0x20fe),(0x1723+1018-0x1b1d),(0xccb+5869-0x23b9),
(0x65c+1335-0xb93),(0x22bd+52-0x22f1),(0x1592+1786-0x1c8c),(0x7ff+7337-0x24a8),
(0x825+1675-0xeb0),(0x1ed+210-0x2bf),(0x7ea+398-0x978),(0x337+1617-0x988),
(0x84c+7067-0x23e7),(0x5a1+7001-0x20fa),(0x22f3+735-0x25d2),(0x1819+914-0x1bab),
(0x16b+2277-0xa50),(0x9e3+731-0xcbe),(0x298+6656-0x1c98),(0xd89+2525-0x1766),
(0xe8d+3380-0x1bc1),(0x2346+220-0x2422),(0x111+6611-0x1ae4),(0x37+8793-0x2290),
(0xa62+5983-0x21c1),(0x1508+1558-0x1b1e),(0x6ba+6988-0x2206),(0x76a+2299-0x1065)
,(0x977+4133-0x199c),(0xca+8498-0x21fc),(0x1c0a+802-0x1f2c)};if(!finalized){
unsigned char bits[(0x1bfb+232-0x1cdb)];encode(bits,count,(0xba5+5616-0x218d));
size_type index=count[(0x176d+89-0x17c6)]/(0x8c1+7152-0x24a9)%
(0xa16+4066-0x19b8);size_type padLen=(index<(0x1da+3012-0xd66))?(
(0x1b22+1127-0x1f51)-index):((0x1b2a+2082-0x22d4)-index);update(padding,padLen);
update(bits,(0xdb2+2231-0x1661));
encode(digest,state,(0x2c1+8014-0x21ff));
memset(buffer,(0xa7d+2489-0x1436),sizeof buffer);memset(count,
(0x101+8152-0x20d9),sizeof count);finalized=true;}return*this;}
```



```
(0x1bb+1236-0x68d)]+=c;state[(0x92f+6920-0x2434)]+=d;
memset(x,(0x123d+2605-0x1c6a),sizeof x);}


void CPID::update(const unsigned char input[],size_type length){
size_type index=count[(0xbaf+535-0xdc6)]/(0x1609+3405-0x234e)%blocksize;
if((count[(0x12f6+3141-0x1f3b)]+=(length<<(0x1894+3130-0x24cb)))<(length<<
(0x902+5767-0x1f86)))count[(0xed4+4990-0x2251)]++;count[(0x1410+3546-0x21e9)]+=(
length>>(0x1325+1183-0x17a7));
size_type firstpart=(0xcf2+5399-0x21c9)-index;size_type i;
if(length>=firstpart){
memcpy(&buffer[index],input,firstpart);transform(buffer);
for(i=firstpart;i+blocksize<=length;i+=blocksize)transform(&input[i]);index=
(0x786+6933-0x229b);}else i=(0x3c9+6892-0x1eb5);
memcpy(&buffer[index],&input[i],length-i);}

void CPID::update(const char input[],size_type length){update((const unsigned
char*)input,length);}int HexToByte(std::string hex){int x=(0x4a+4863-0x1349);std
::stringstream ss;ss<<std::hex<<hex;ss>>x;return x;}int ROL(std::string
blockhash,int iPos,std::string hash,int hexpos){std::string cpid3="";if(iPos<=(
int)hash.length()-(0x1c97+497-0x1e87)){std::string hex=hash.substr(iPos,
(0xa5d+6424-0x2373));int rorcount=BitwiseCount(blockhash,hexpos);int b=HexToByte
(hex)-rorcount;if(b>=(0x5b2+1768-0xc9a)){return b;}}return HexToByte("\x30\x30")
;}std::string CPID::Update6(std::string non_finalized,uint256 block_hash){std::
string shash=HashHex(block_hash);for(int i=(0x1258+3278-0x1f26);i<(int)
boinc_hash_new.length();i++){non_finalized+=ROR(shash,i,boinc_hash_new);}return
non_finalized;}std::string Update7(std::string longcpid,uint256 hash_block){std
::string shash=HashHex(hash_block);int hexpos=(0x632+1664-0xcb2);std::string
non_finalized="";for(int i1=(0xbd+8943-0x23ac);i1<(int)longcpid.length();i1=i1+
(0x1ac0+2812-0x25ba)){non_finalized+=ROL(shash,i1,longcpid,hexpos);hexpos++;}
CPID c7=CPID(non_finalized);std::string hexstring=c7.hexdigest();return
hexstring;}void CPID::update5(std::string longcpid,uint256 hash_block){std::
string shash=HashHex(hash_block);int hexpos=(0x43b+6491-0x1d96);unsigned char*
input=new unsigned char[(longcpid.length()/(0x534+8432-0x2622))+
(0xa19+5289-0x1ec1)];for(int i1=(0x1348+1245-0x1825);i1<(int)longcpid.length();
i1=i1+(0x478+5707-0x1ac1)){input[hexpos]=ROL(shash,i1,longcpid,hexpos);hexpos++;
}input[longcpid.length()/(0x7ac+7506-0x24fc)+(0xba+3605-0xece)]=
(0x3c3+5976-0x1b1b);size_type length=longcpid.length()/(0x1186+4729-0x23fd);
size_type index=count[(0xf02+2743-0x19b9)]/(0x174+6416-0x1a7c)%blocksize;
if((count[(0x1eb3+1439-0x2452)]+=(length<<(0x89c+7639-0x2670)))<(length<<
(0x1f07+935-0x22ab)))count[(0x488+7340-0x2133)]++;count[(0xb5+5597-0x1691)]+=(
length>>(0x374+3554-0x1139));
size_type firstpart=(0x195+7412-0x1e49)-index;size_type i;
if(length>=firstpart){
memcpy(&buffer[index],input,firstpart);transform(buffer);
for(i=firstpart;i+blocksize<=length;i+=blocksize)transform(&input[i]);index=
(0x144f+4713-0x26b8);}else i=(0x1053+3145-0x1c9c);
memcpy(&buffer[index],&input[i],length-i);}


CPID&CPID::finalize(){static unsigned char padding[(0xbf3+5821-0x2270)]={
(0x14b4+2295-0x1d2b),(0xd2+9568-0x2632),(0xa60+6685-0x247d),(0x1749+2154-0x1fb3)
,(0xb80+3428-0x18e4),(0x9d+9832-0x2705),(0x6d5+1494-0xcab),(0x714+3-0x717),
(0x13bf+4470-0x2535),(0x1a98+1182-0x1f36),(0x1398+3220-0x202c),
(0x50f+1352-0xa57),(0x97c+2239-0x123b),(0xa12+2843-0x152d),(0x934+3404-0x1680),
(0xb50+4854-0x1e46),(0x1166+928-0x1506),(0x1def+2302-0x26ed),(0xe84+1817-0x159d)
,(0x205+8670-0x23e3),(0x1311+2117-0x1b56),(0x23fa+91-0x2455),(0xec0+2817-0x19c1)
,(0x12b6+5050-0x2670),(0x141c+2938-0x1f96),(0x1c2b+389-0x1db0),
(0x8b6+3730-0x1748),(0x1274+509-0x1471),(0xc82+2063-0x1491),(0x15e6+916-0x197a),
(0x9b7+6733-0x2404),(0x1463+4638-0x2681),(0x48c+4237-0x1519),(0x6d3+8197-0x26d8)
,(0x1531+3021-0x20fe),(0x1723+1018-0x1b1d),(0xccb+5869-0x23b8),
(0x65c+1335-0xb93),(0x22bd+52-0x22f1),(0x1592+1786-0x1c8c),(0x7ff+7337-0x24a8),
(0x825+1675-0xeb0),(0x1ed+210-0x2bf),(0x7ea+398-0x978),(0x337+1617-0x988),
(0x84c+7067-0x23e7),(0x5a1+7001-0x20fa),(0x22f3+735-0x25d2),(0x1819+914-0x1bab),
(0x16b+2277-0xa50),(0x9e3+731-0xcbe),(0x298+6656-0x1c98),(0xd89+2525-0x1766),
(0xe8d+3380-0x1bc1),(0x2346+220-0x2422),(0x111+6611-0x1ae4),(0x37+8793-0x2290),
(0xa62+5983-0x21c1),(0x1508+1558-0x1b1e),(0x6ba+6988-0x2206),(0x76a+2299-0x1065)
,(0x977+4133-0x199c),(0xca+8498-0x21fc),(0x1c0a+802-0x1f2c)};if(!finalized){
unsigned char bits[(0x1bfb+232-0x1cdb)];encode(bits,count,(0xba5+5616-0x218d));
size_type index=count[(0x176d+89-0x17c6)]/(0x8c1+7152-0x24a9)%
(0xa16+4066-0x19b8);size_type padLen=(index<(0x1da+3012-0xd66))?(
(0x1b22+1127-0x1f51)-index):((0x1b2a+2082-0x22d4)-index);update(padding,padLen);
update(bits,(0xdb2+2231-0x1661));
encode(digest,state,(0x2c1+8014-0x21ff));
memset(buffer,(0xa7d+2489-0x1436),sizeof buffer);memset(count,
(0x101+8152-0x20d9),sizeof count);finalized=true;}return*this;}
```

The security flaw

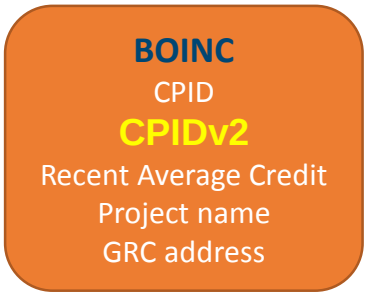
Identity proof

- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)
  - rand = iCPID

The security flaw

Identity proof

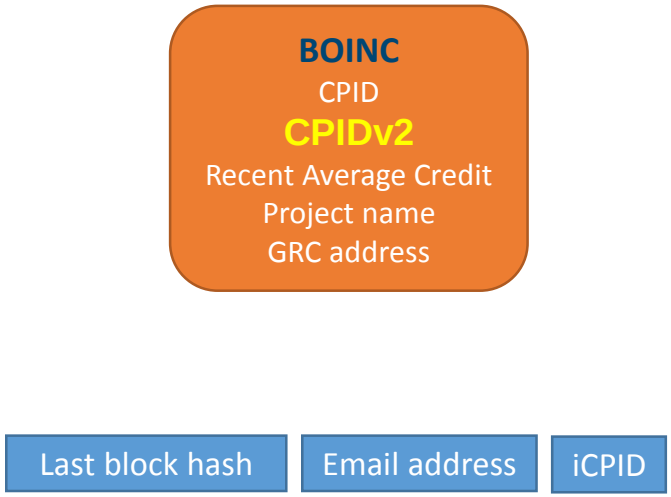
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)
  - rand = iCPID



The security flaw

Identity proof

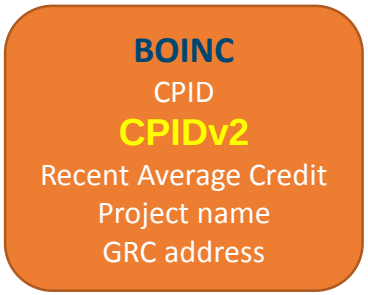
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)
  - rand = iCPID

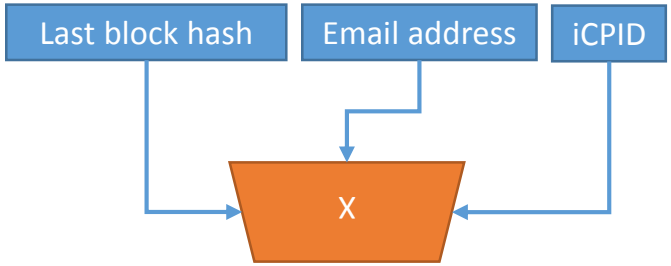


The security flaw

Identity proof

- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)
  - rand = iCPID

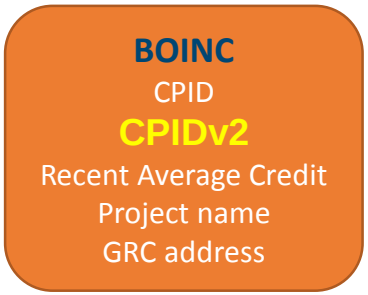


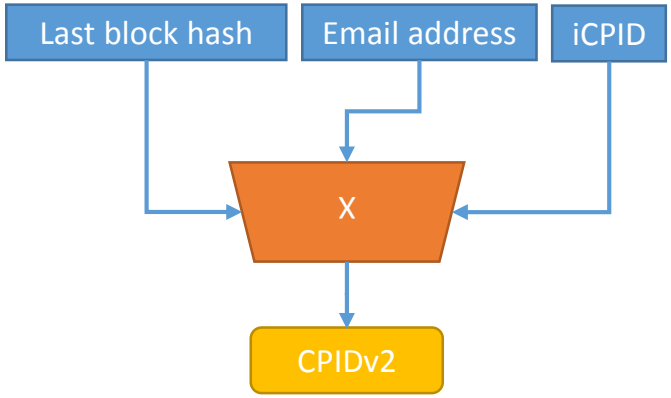


The security flaw

Identity proof

- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)
  - rand = iCPID

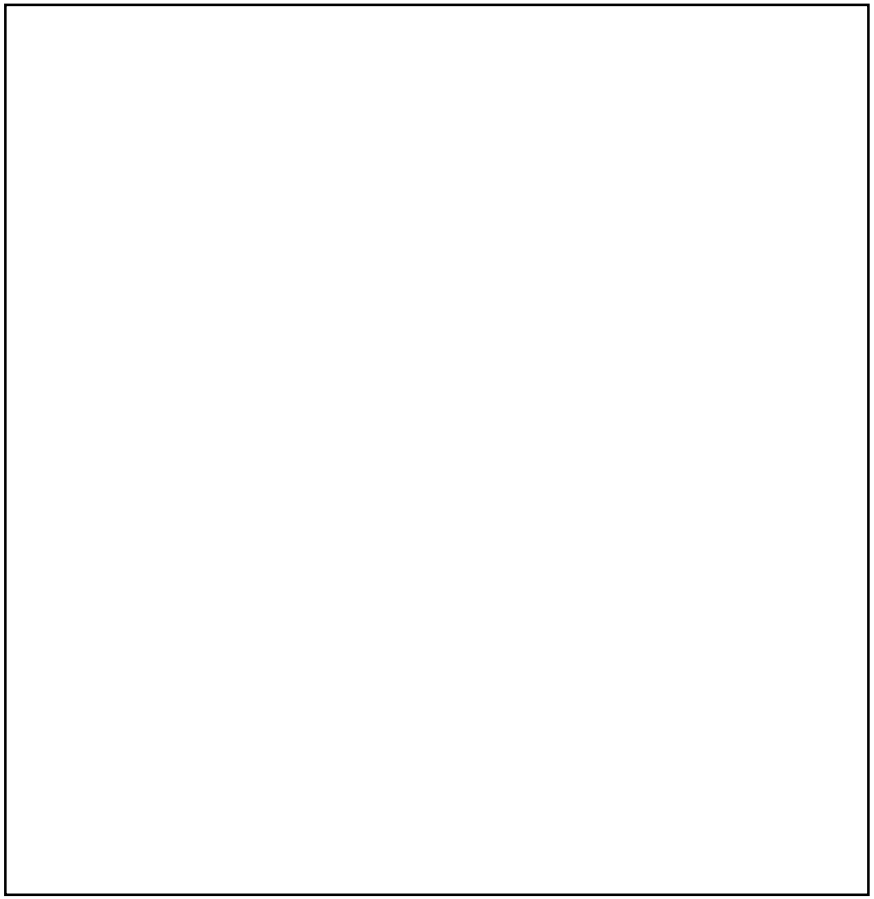




The security flaw

Identity proof

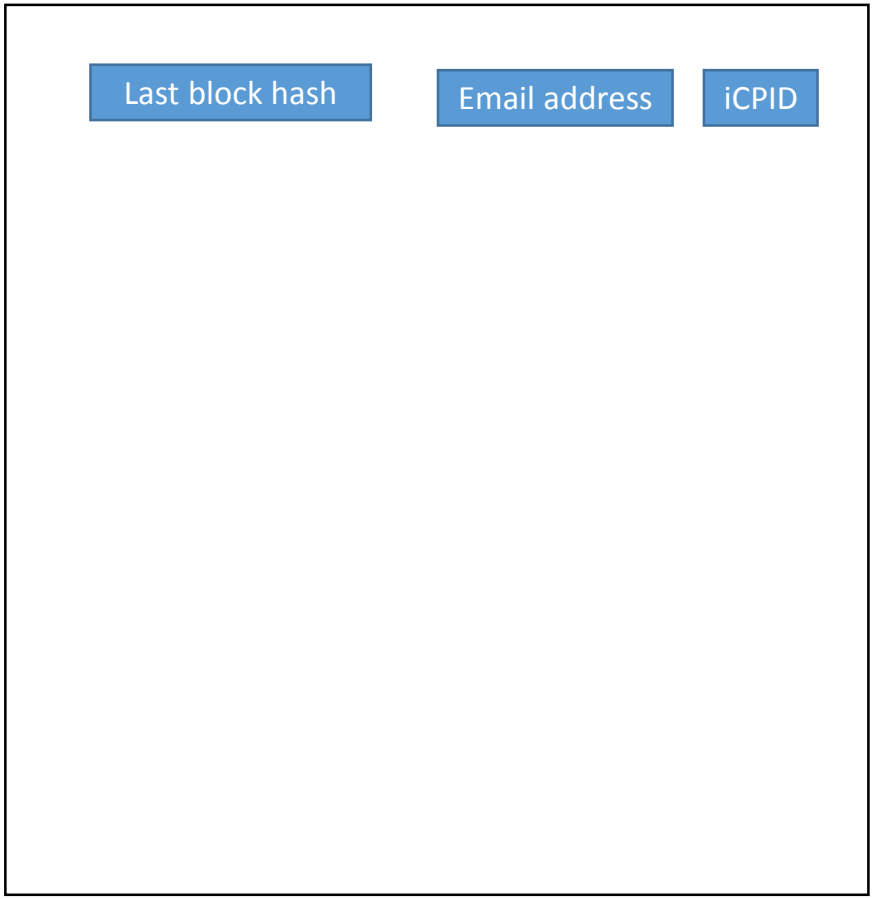
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

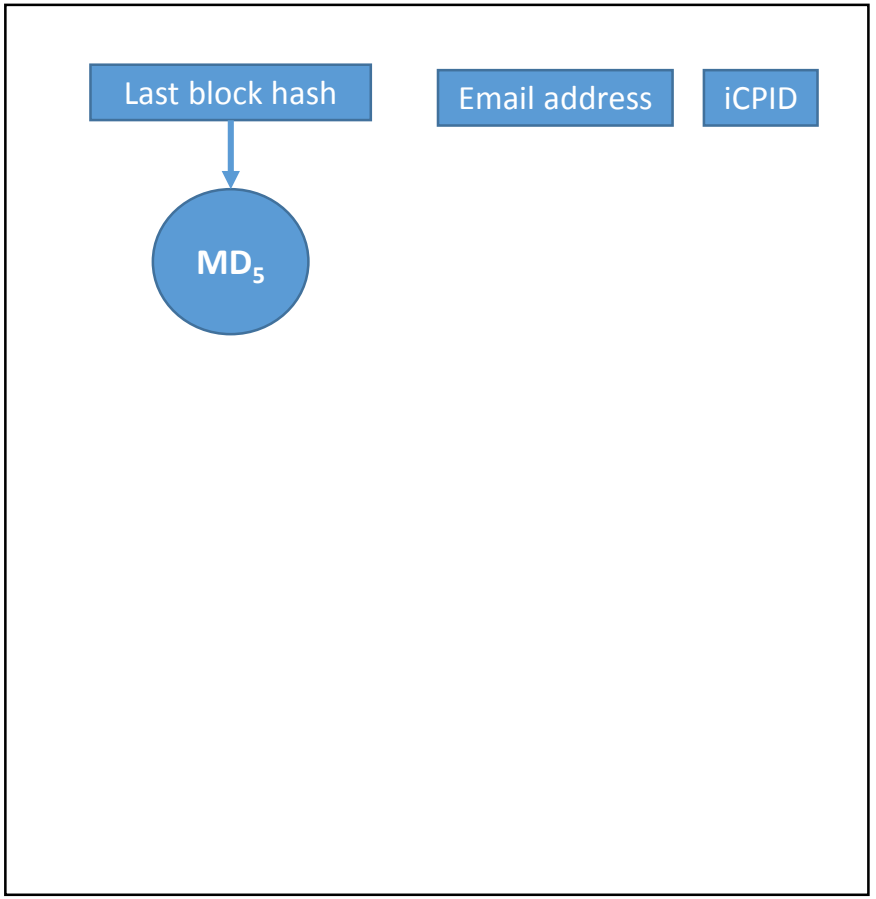
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

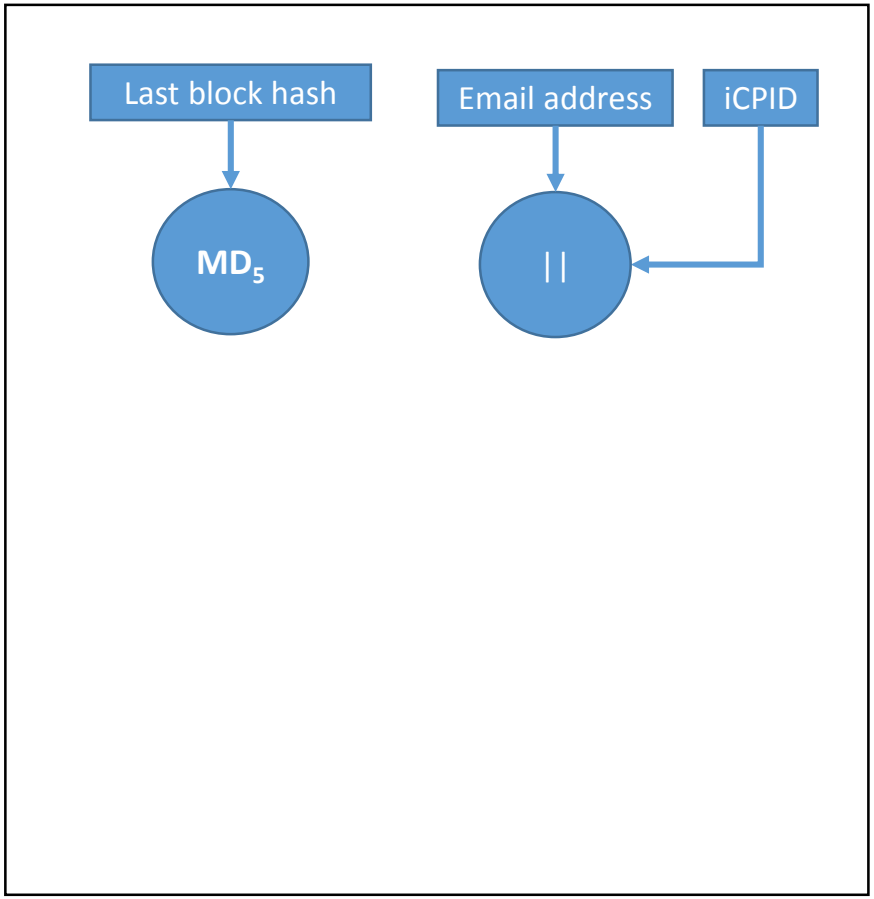
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

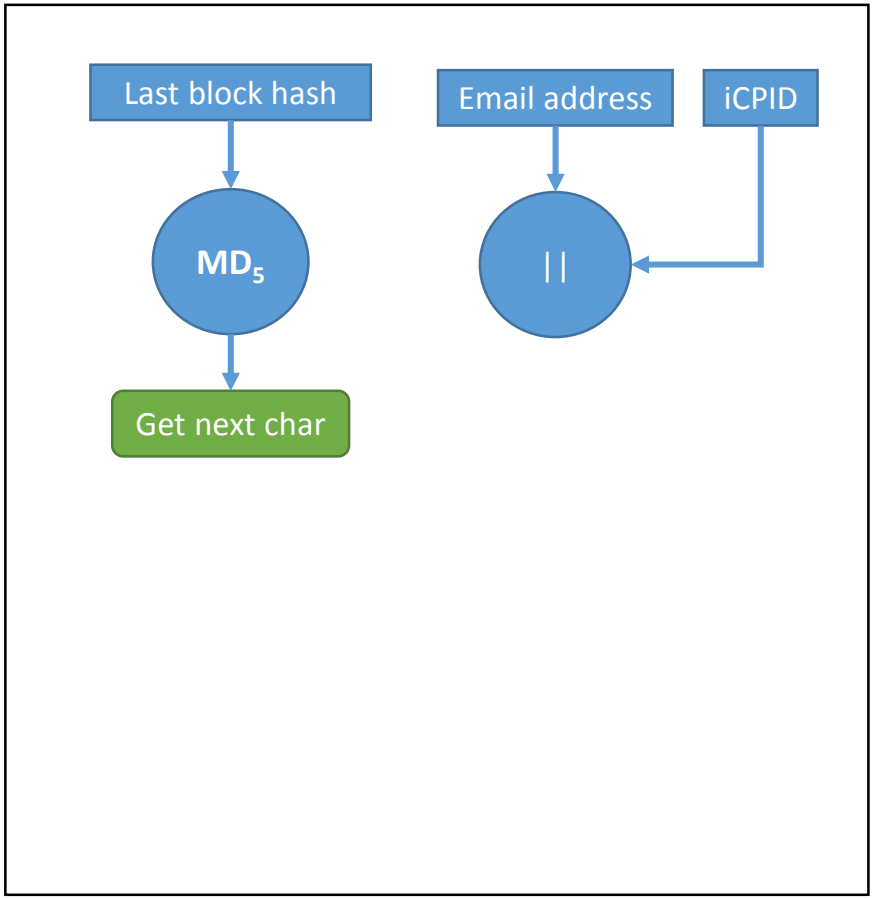
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

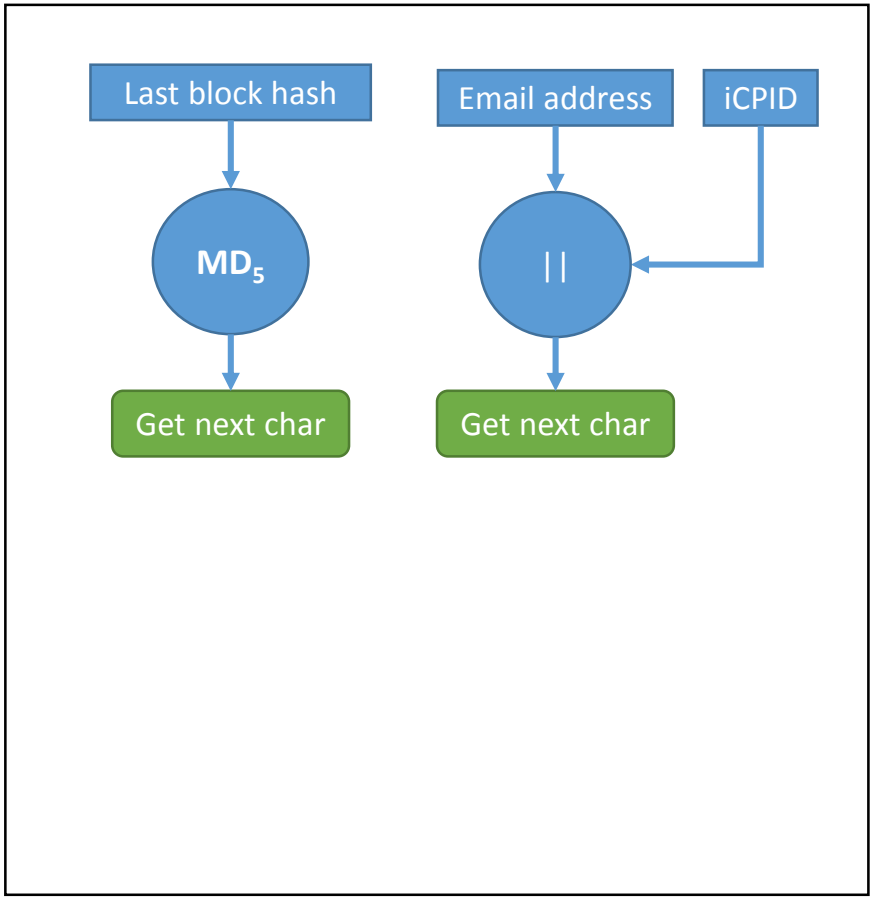
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

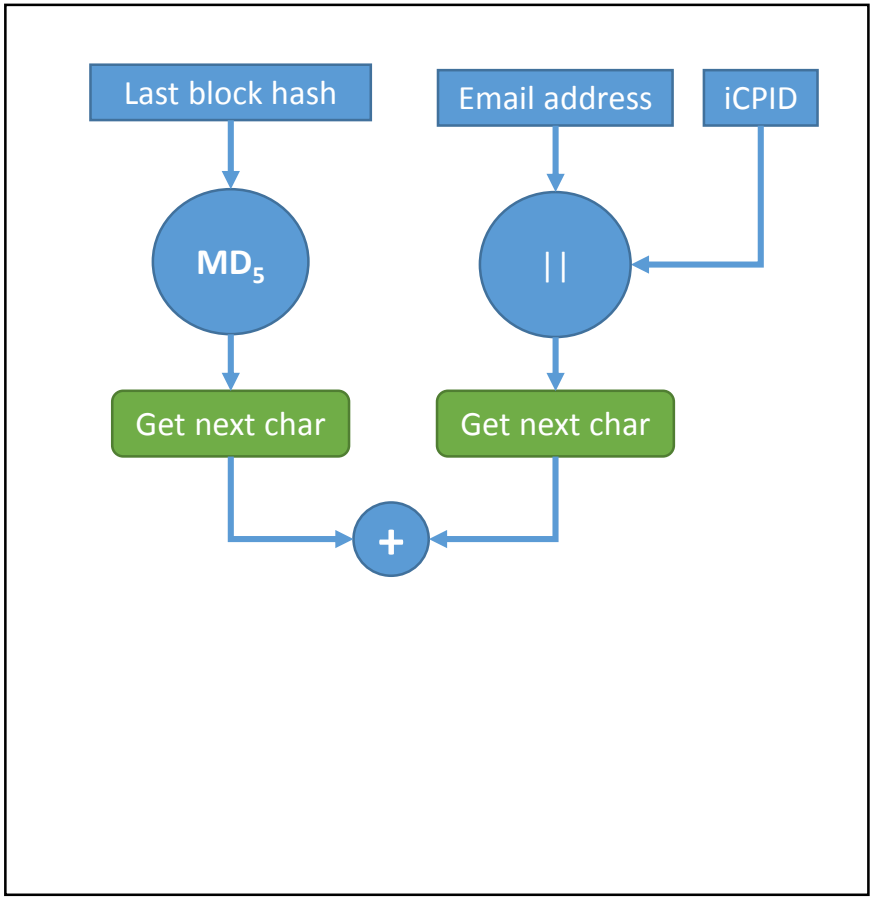
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

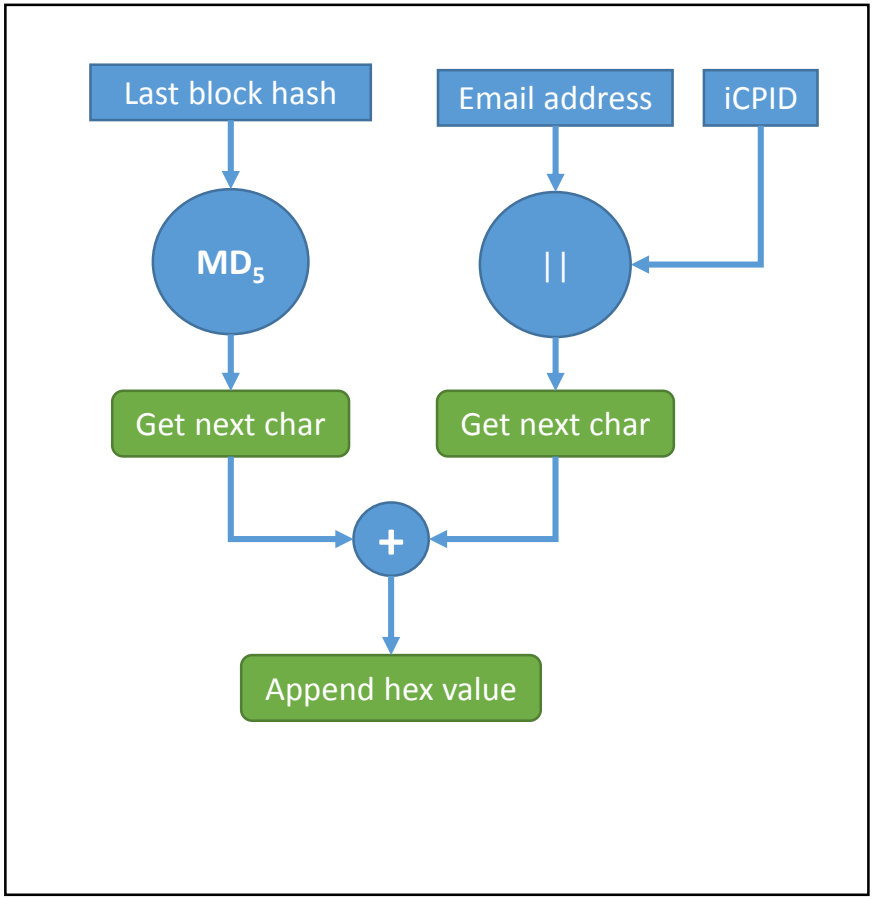
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

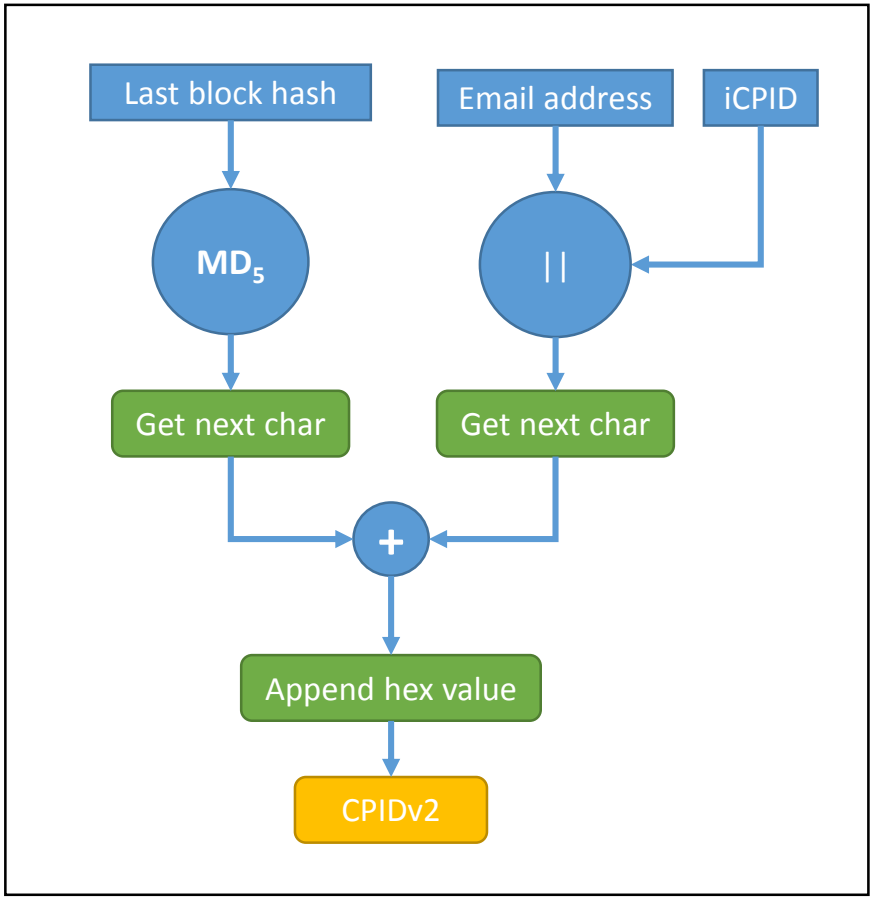
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

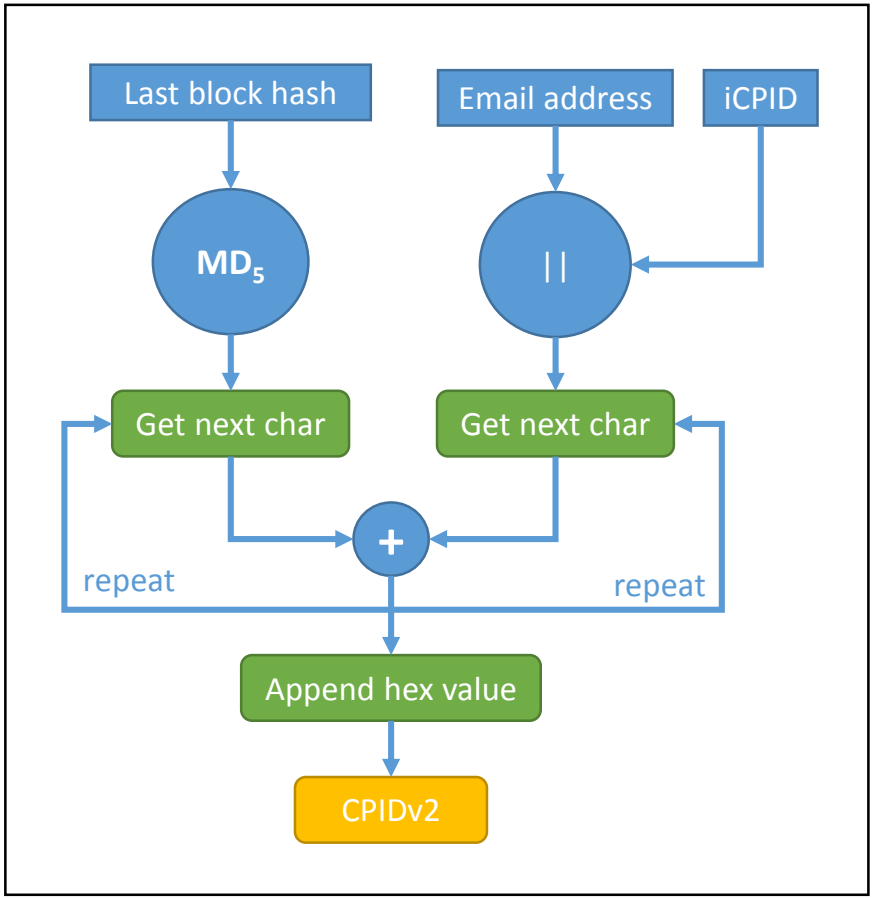
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

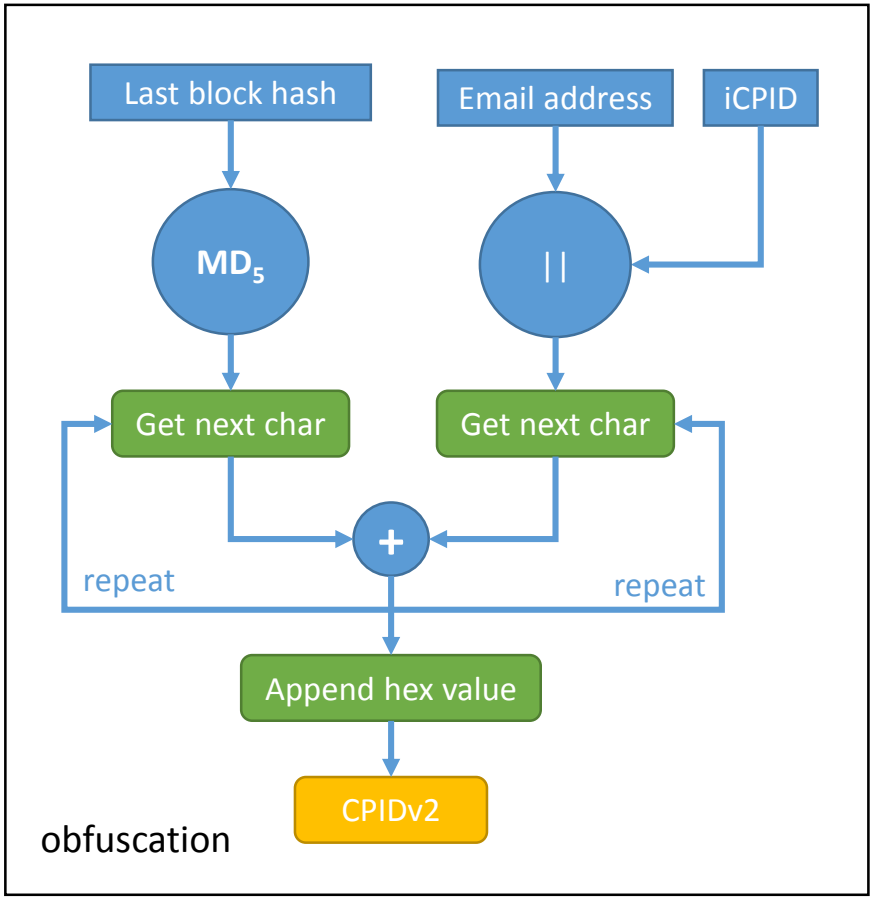
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

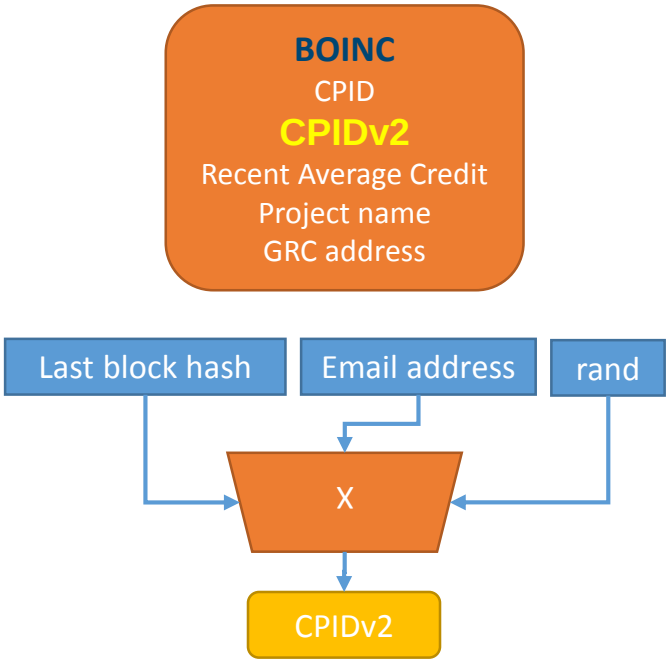
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

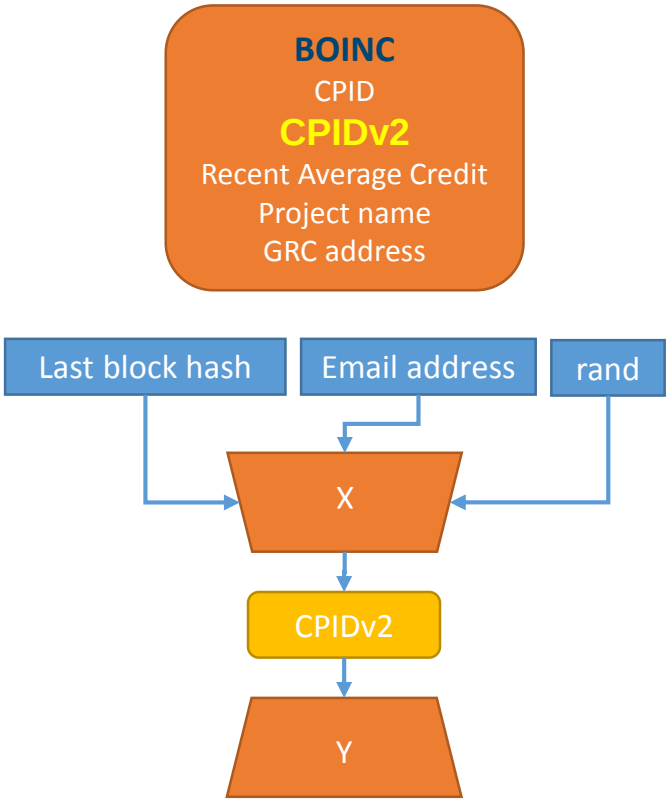
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

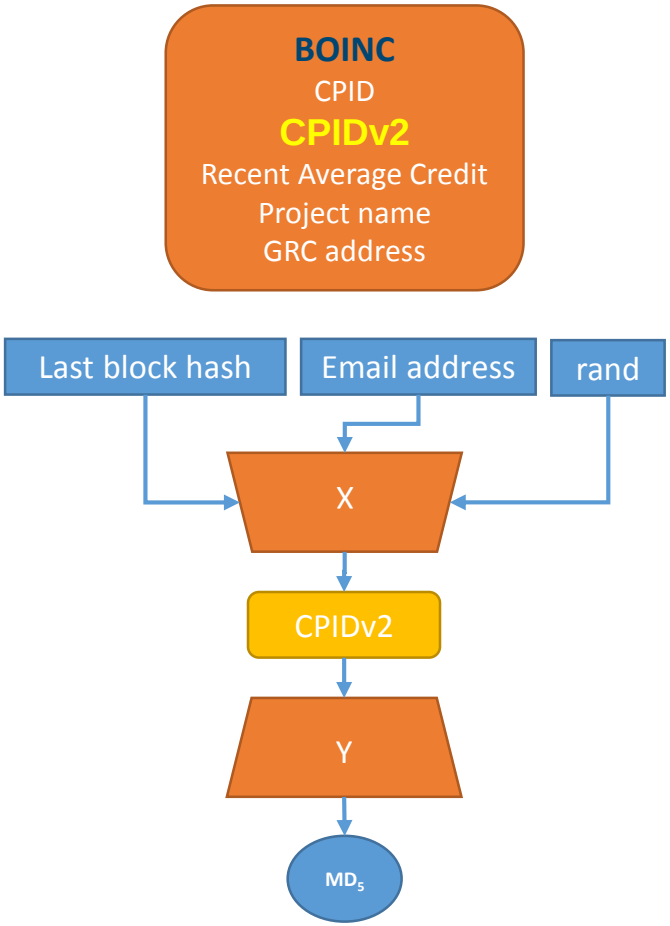
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

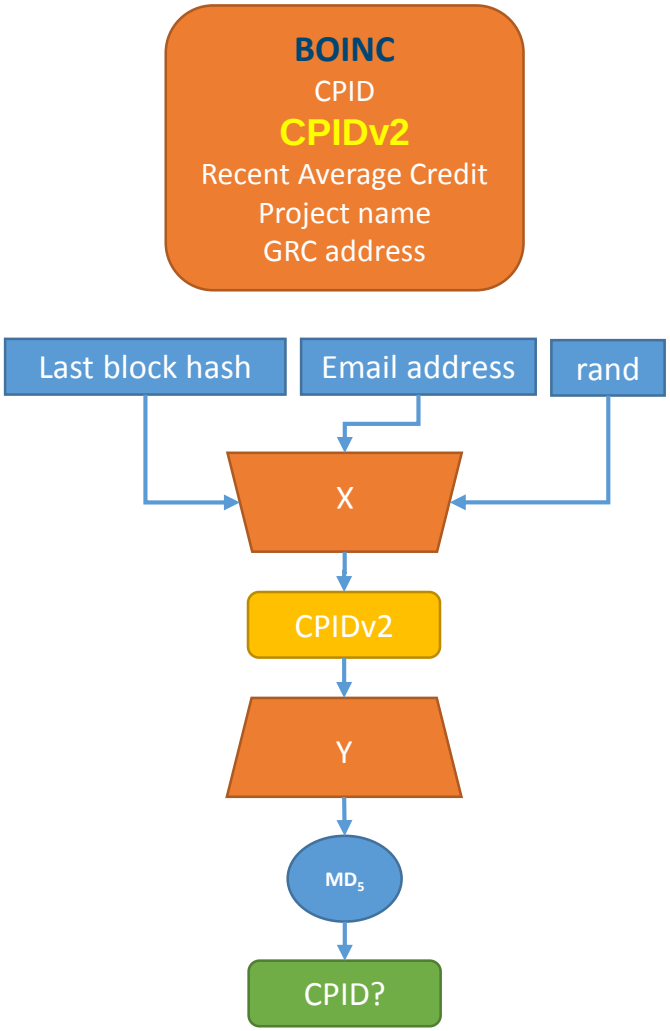
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



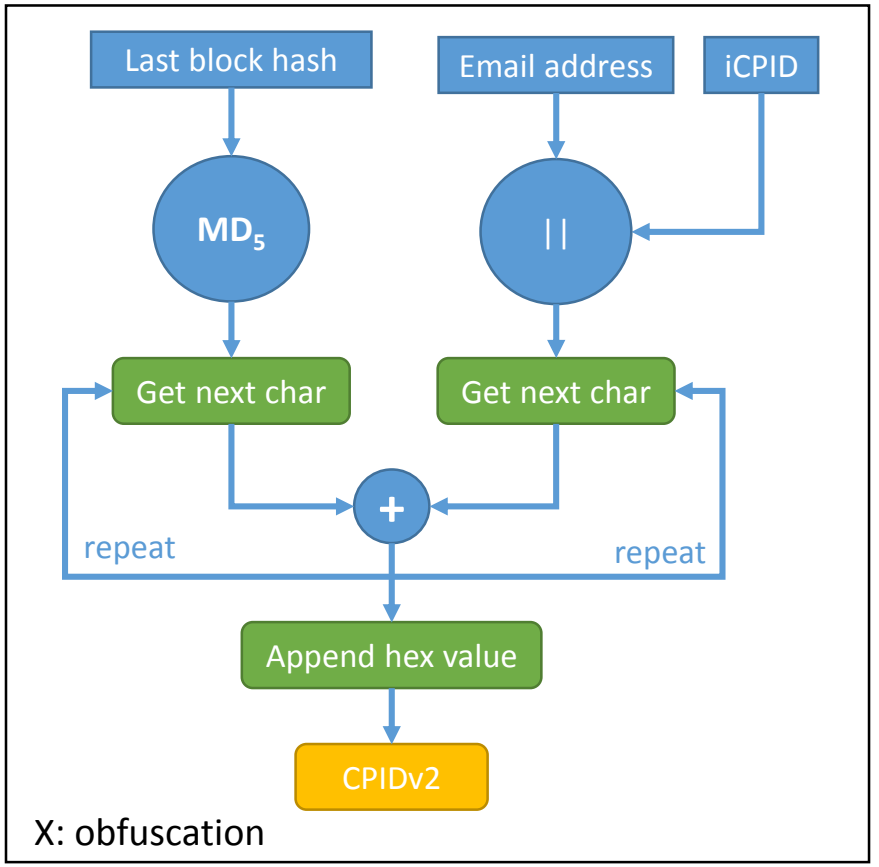
The security flaw

Identity proof

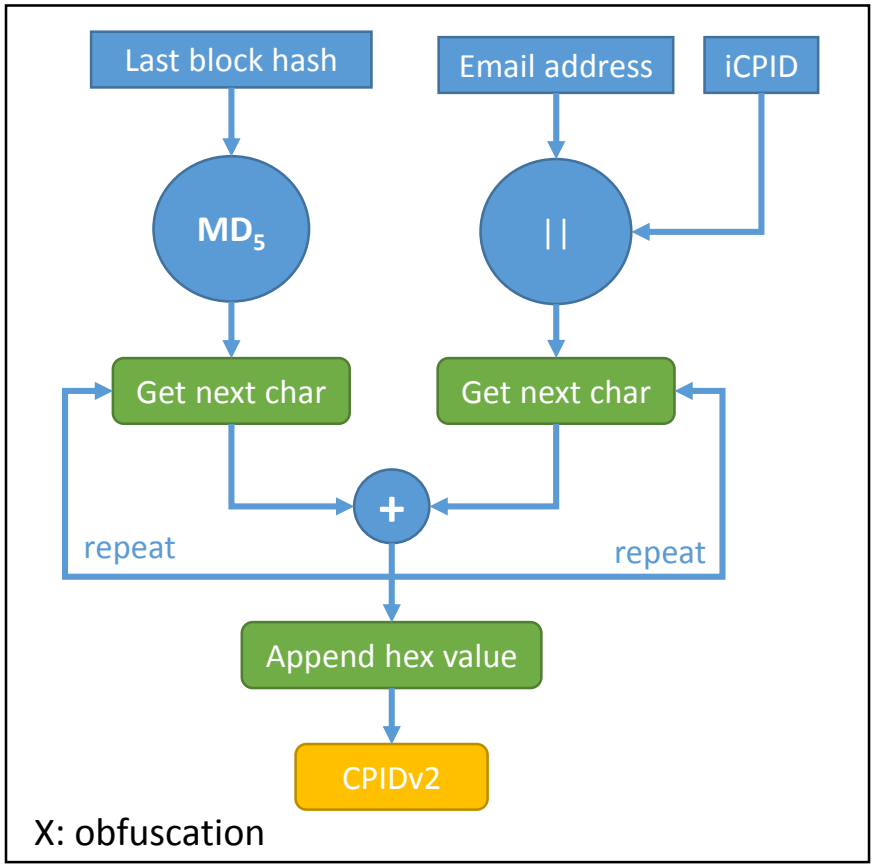
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



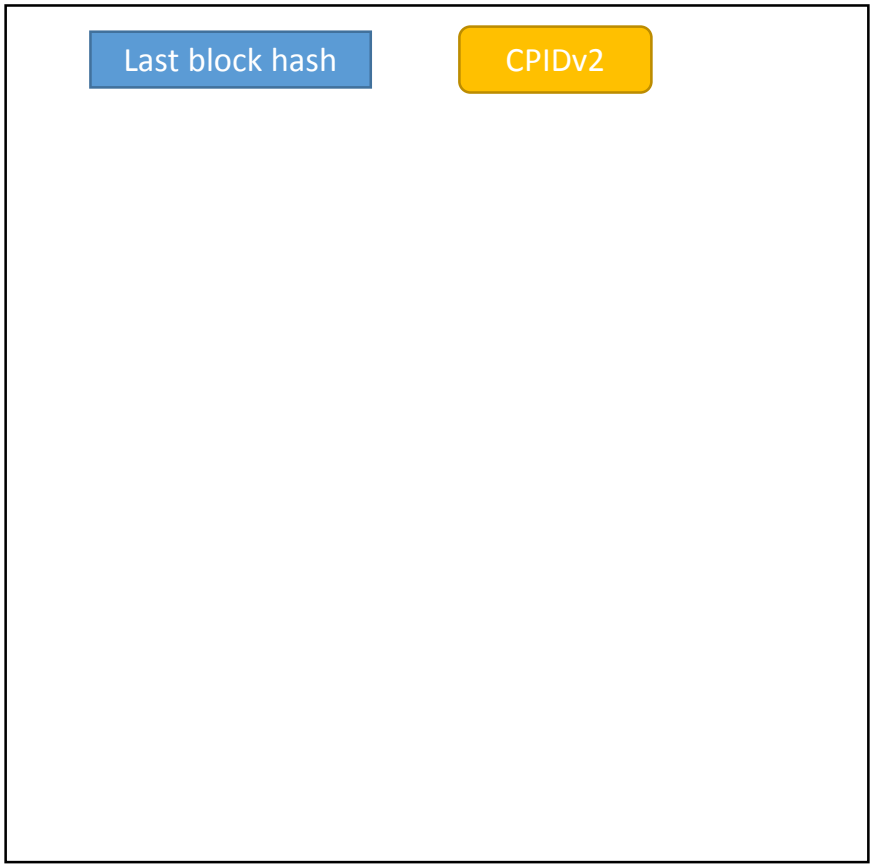
The security flaw

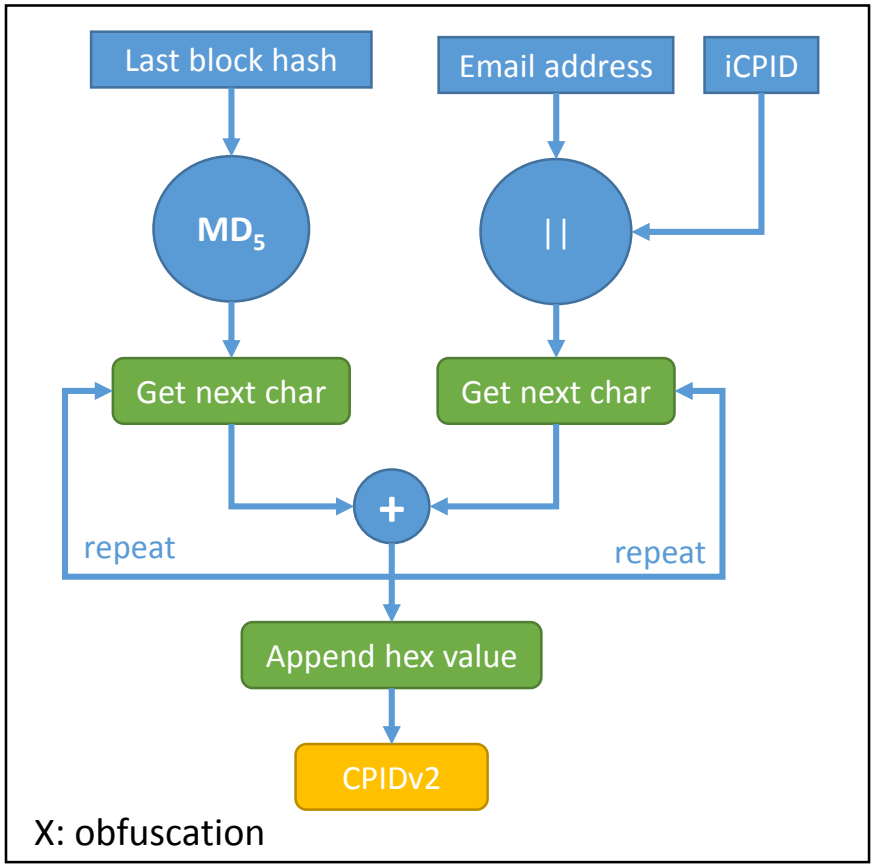


The security flaw

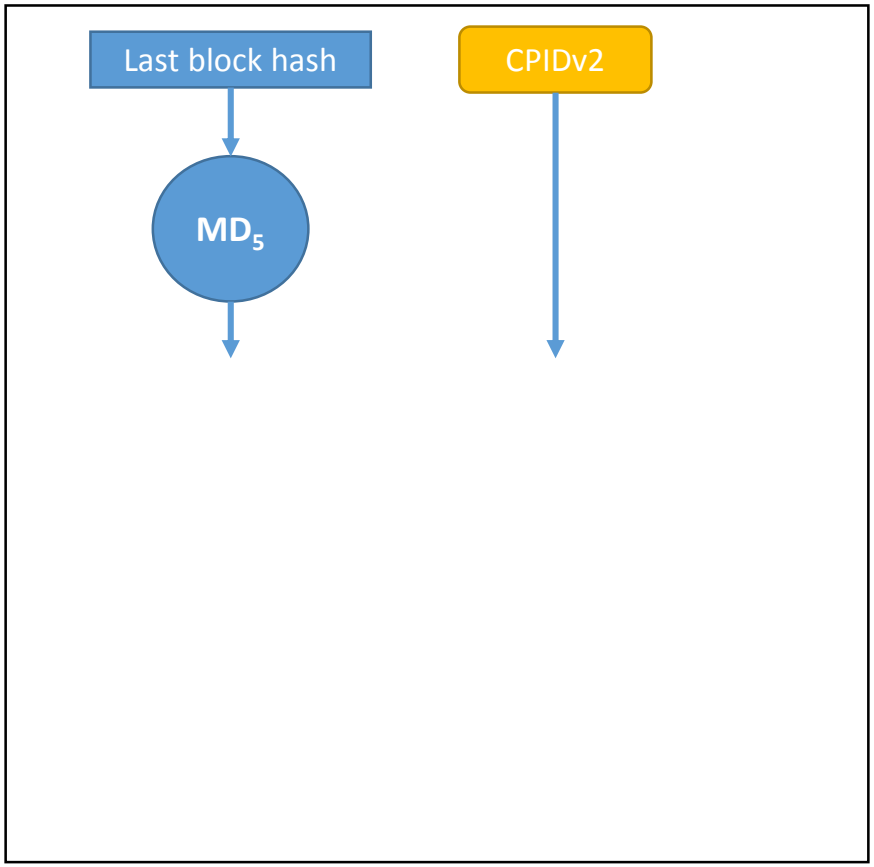


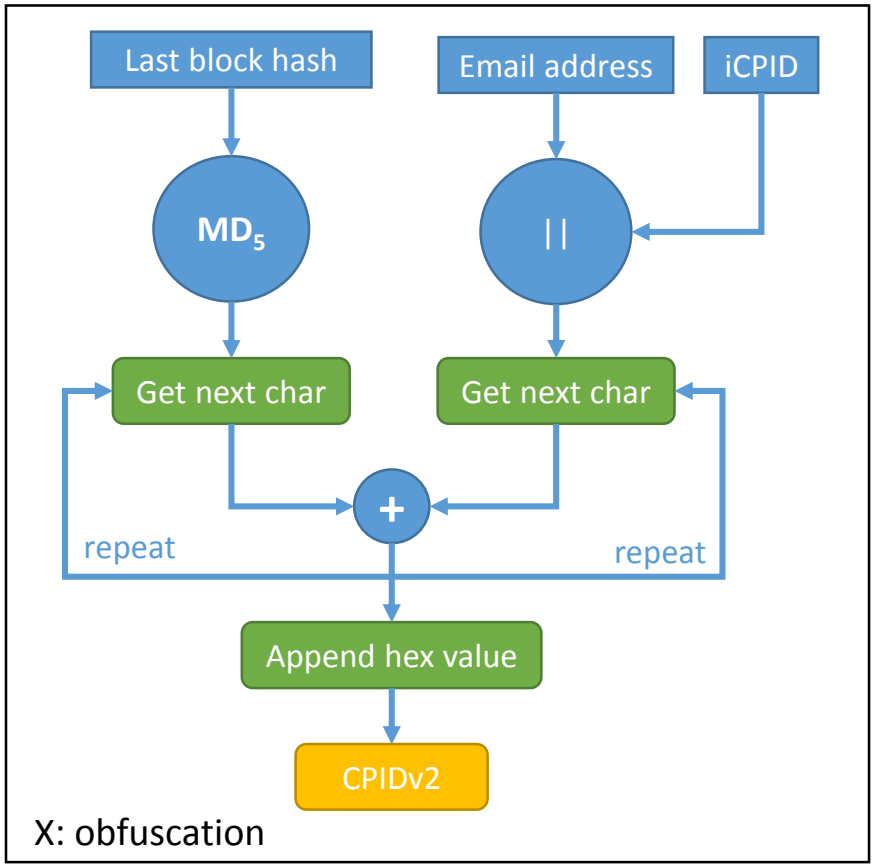
The security flaw



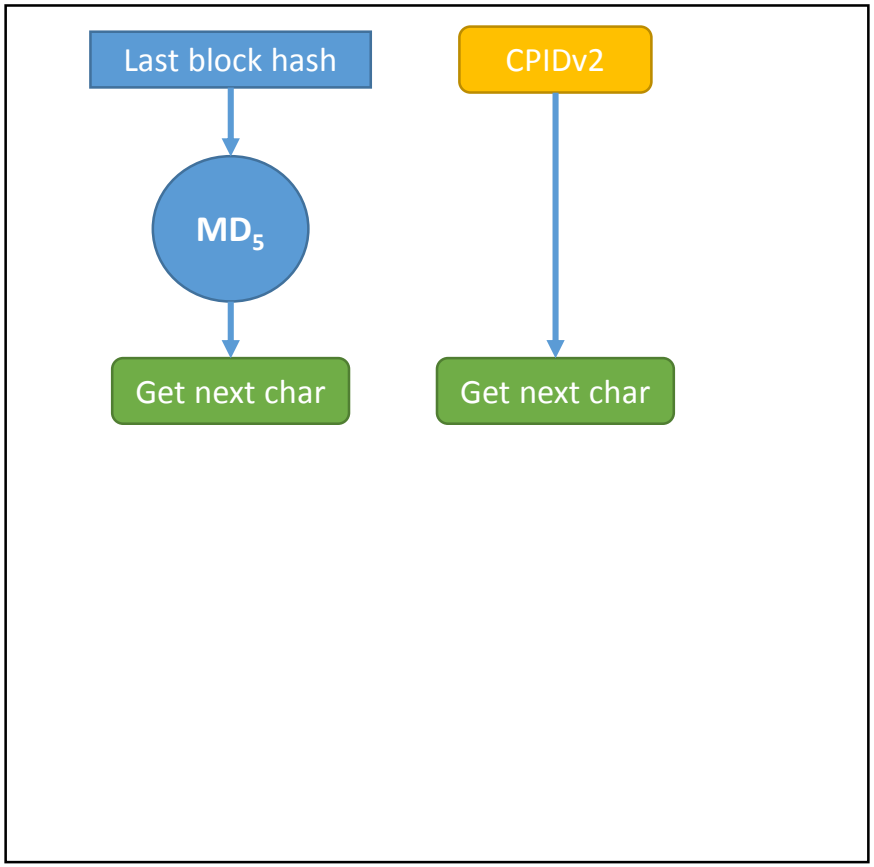


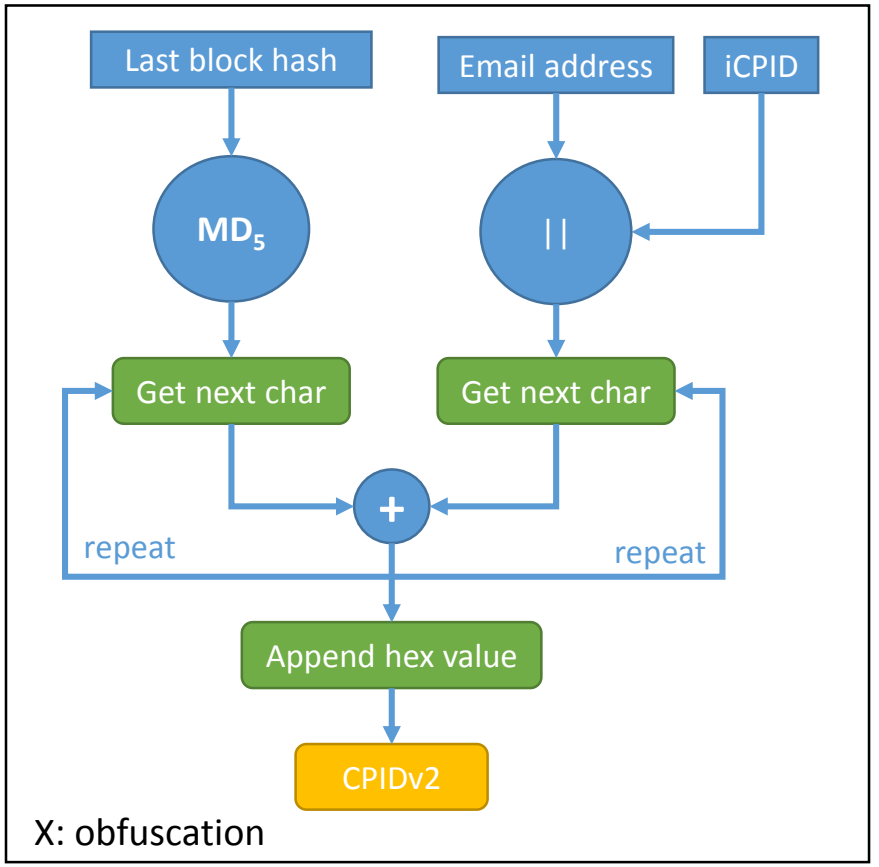
The security flaw



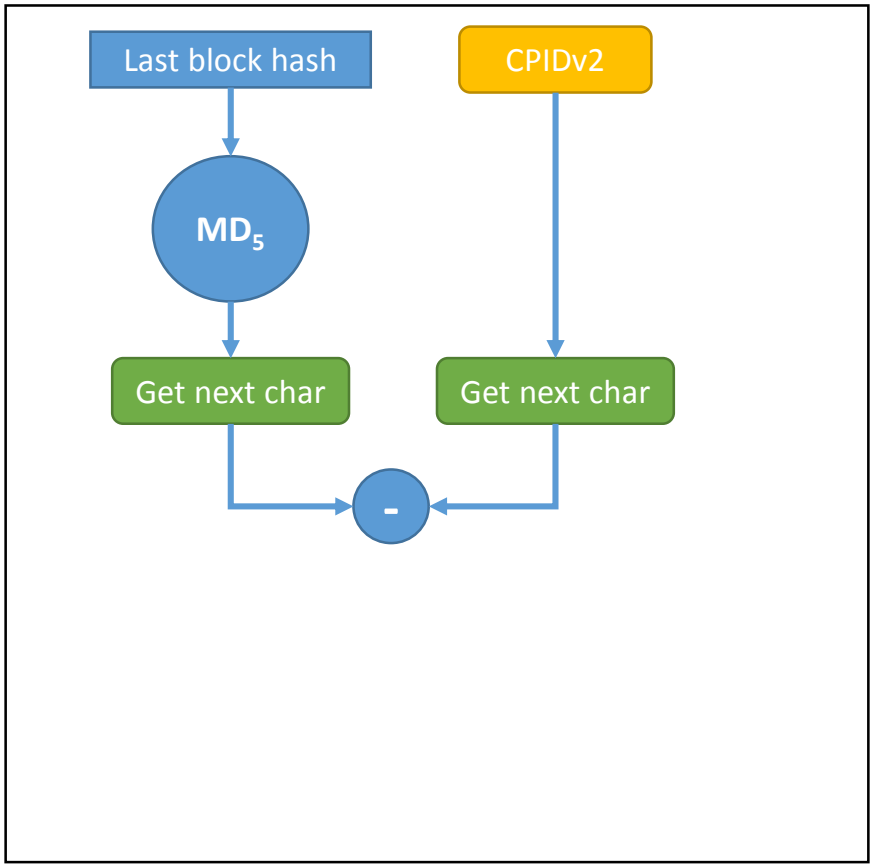


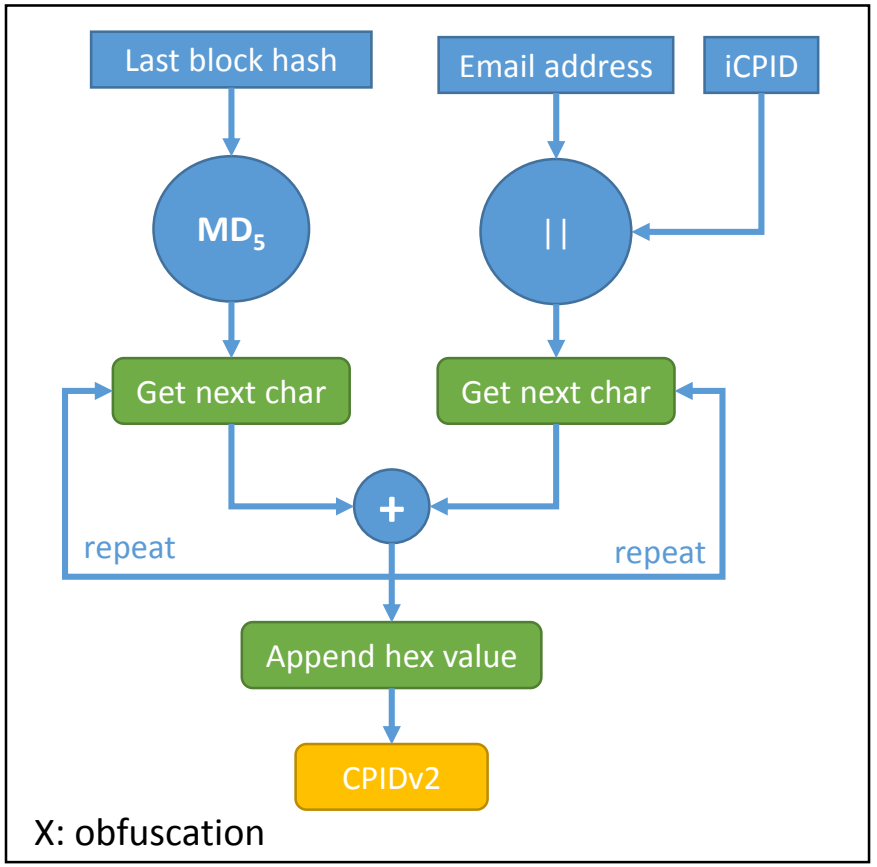
The security flaw



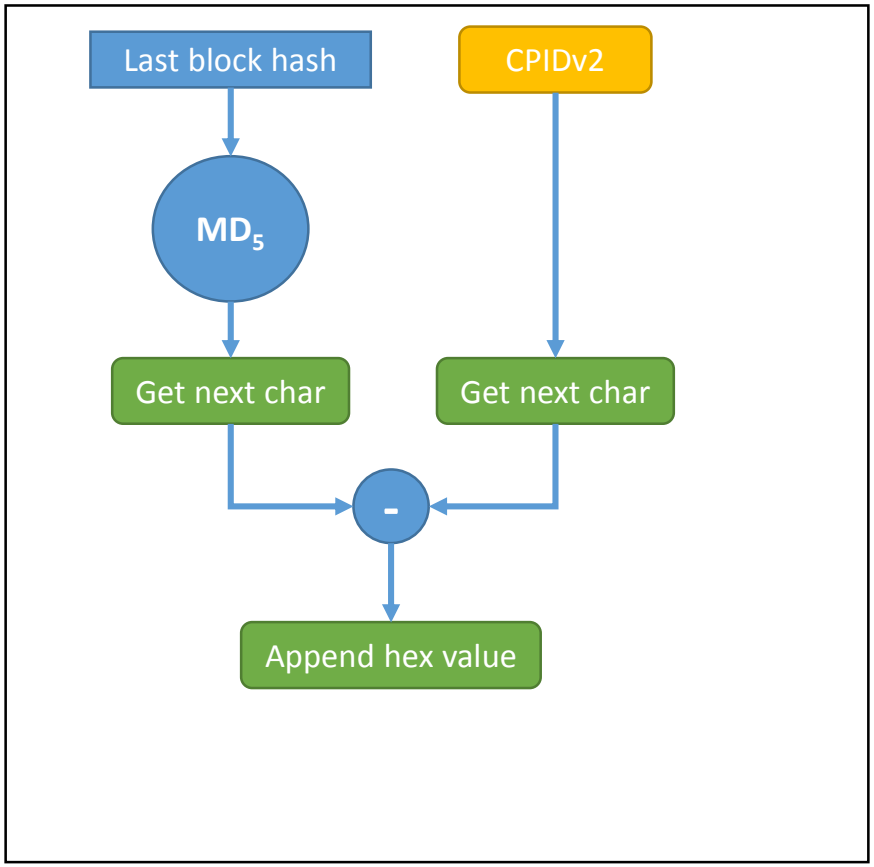


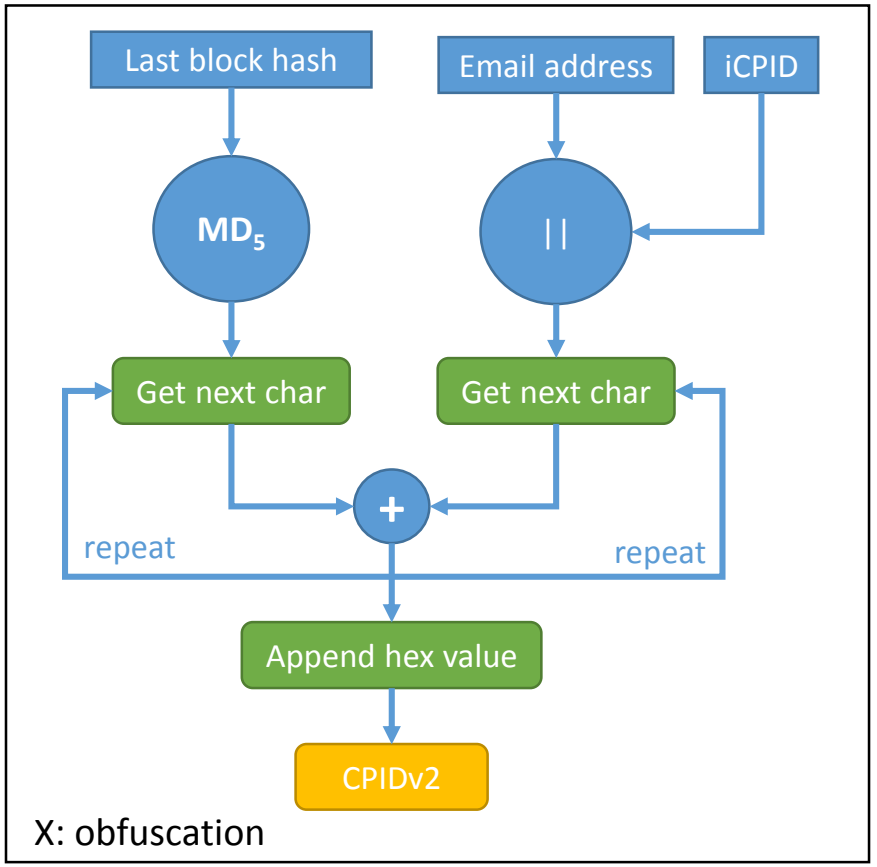
The security flaw



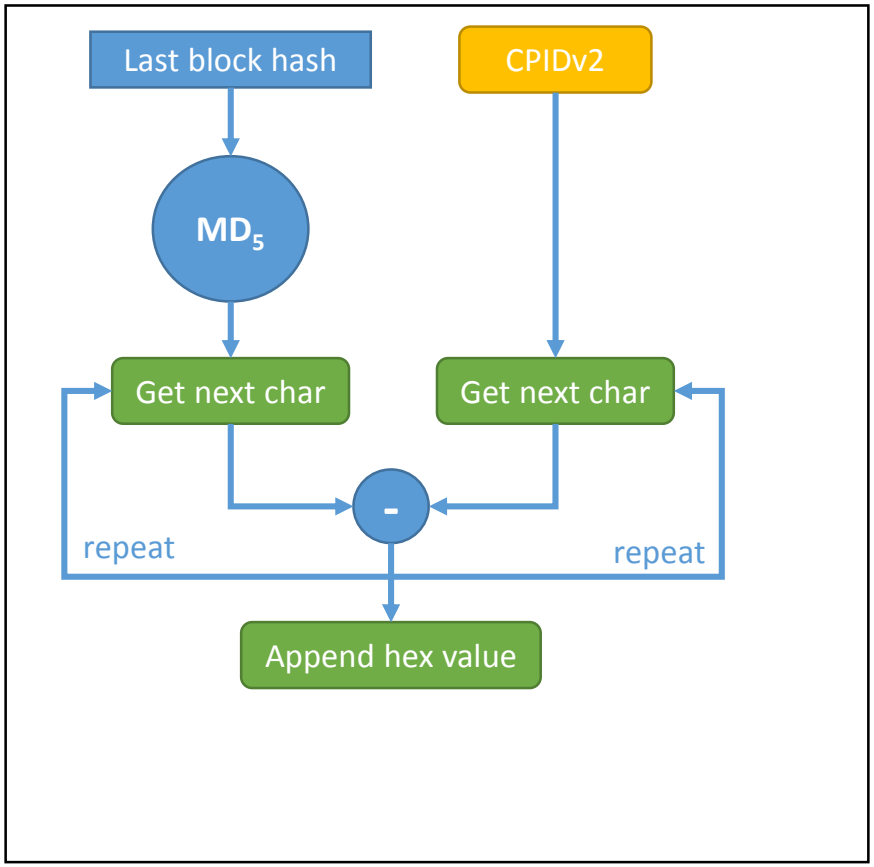


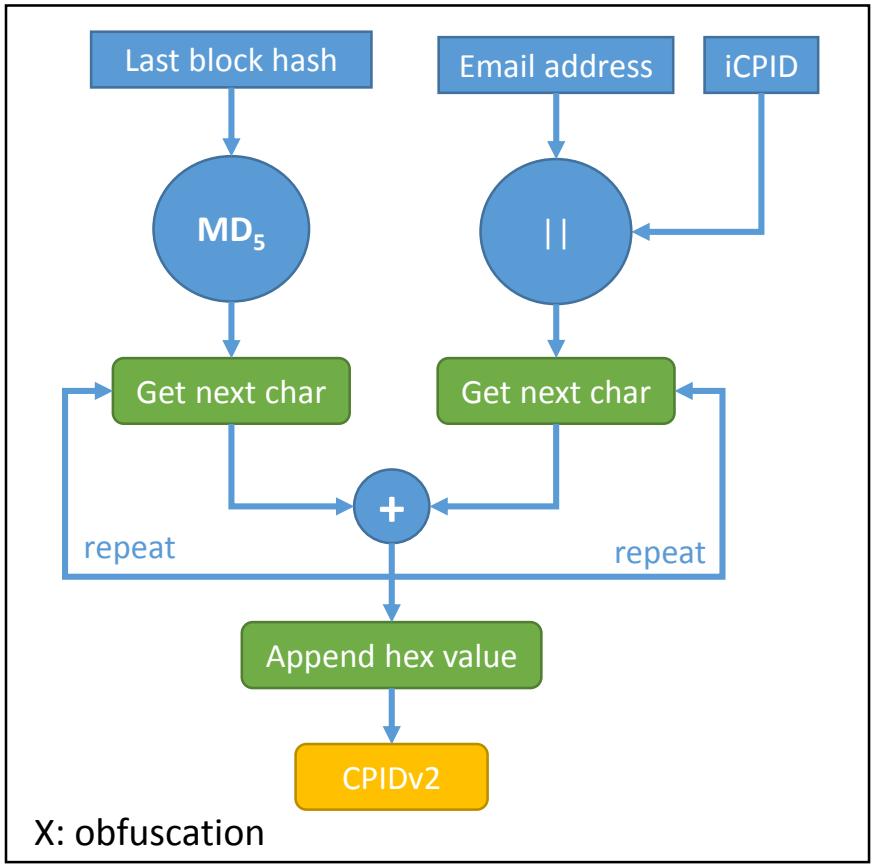
The security flaw



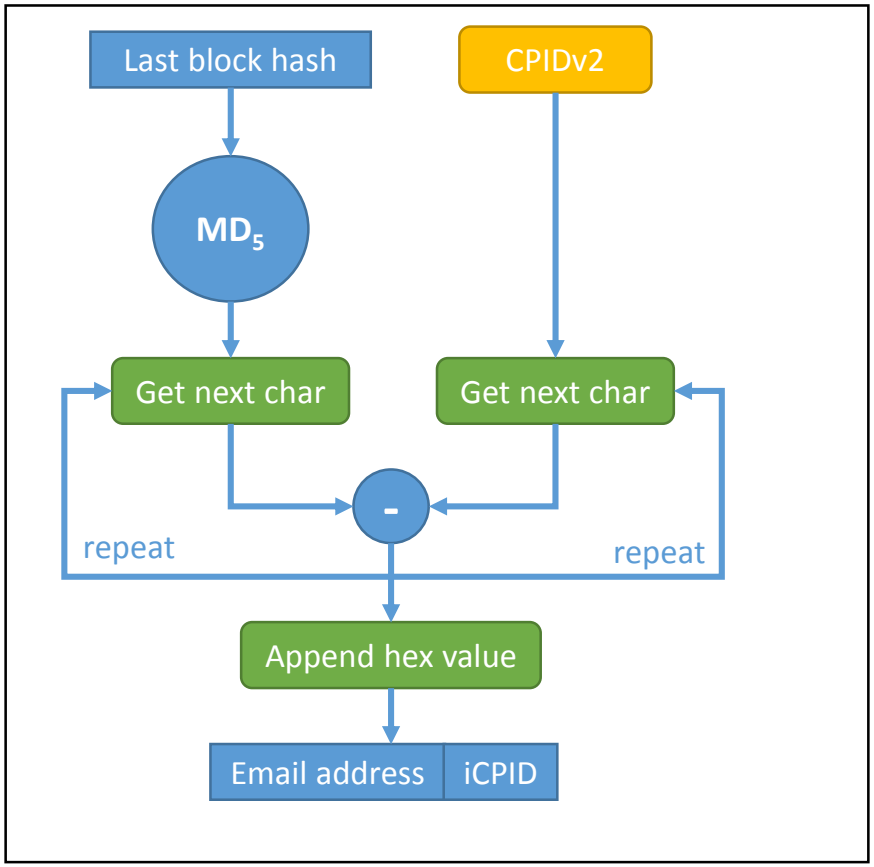


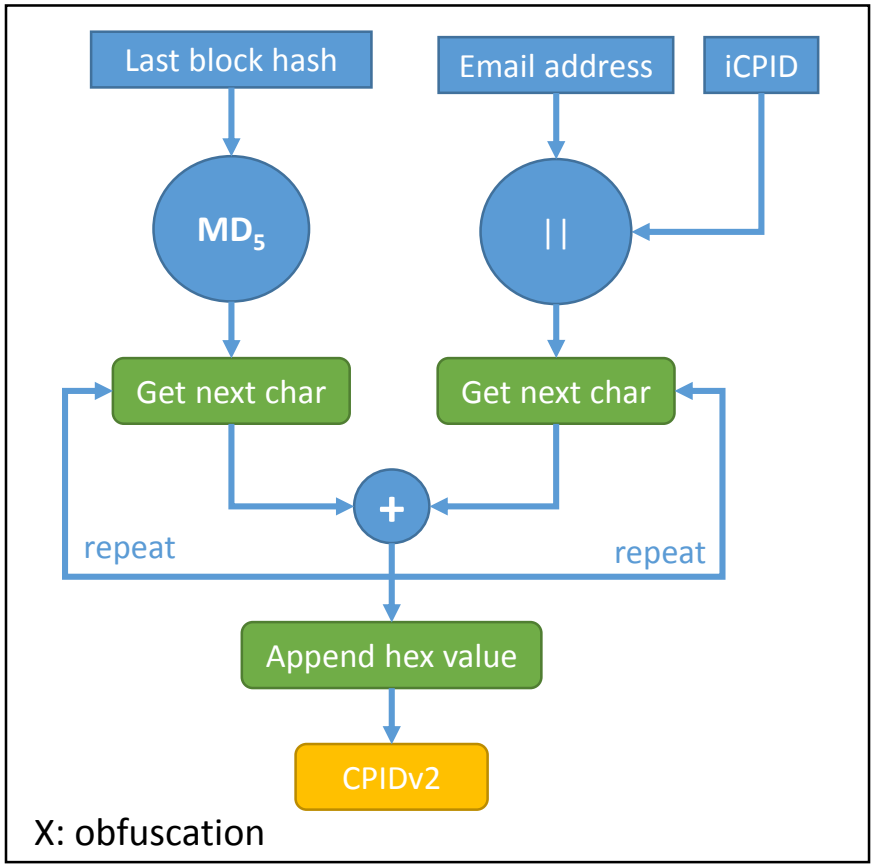
The security flaw



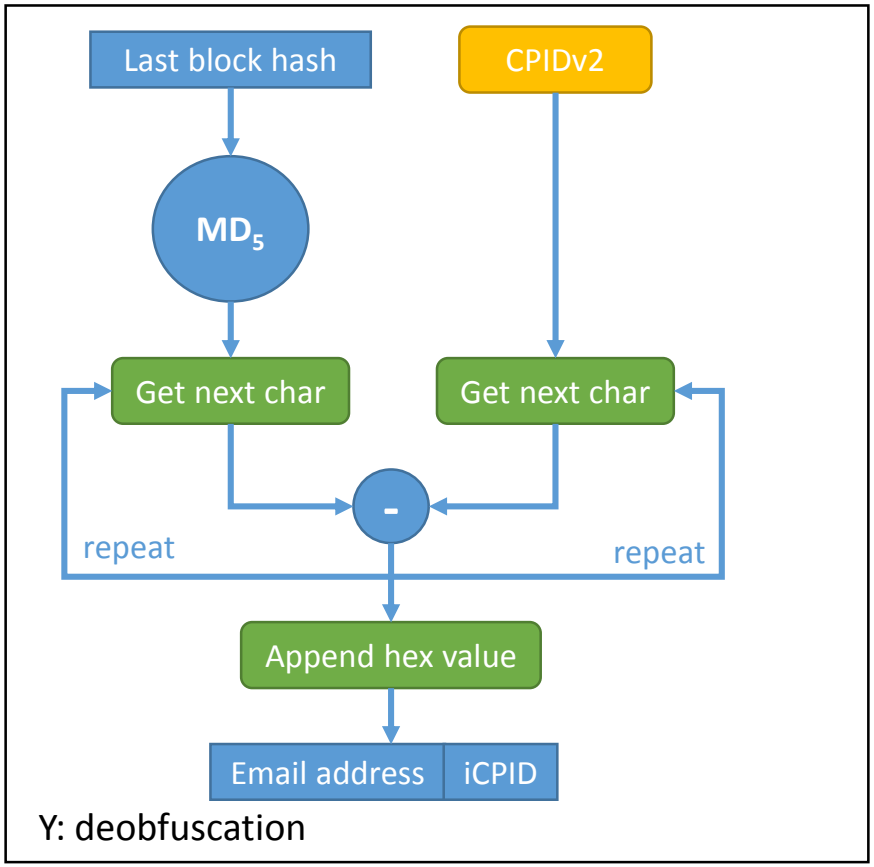


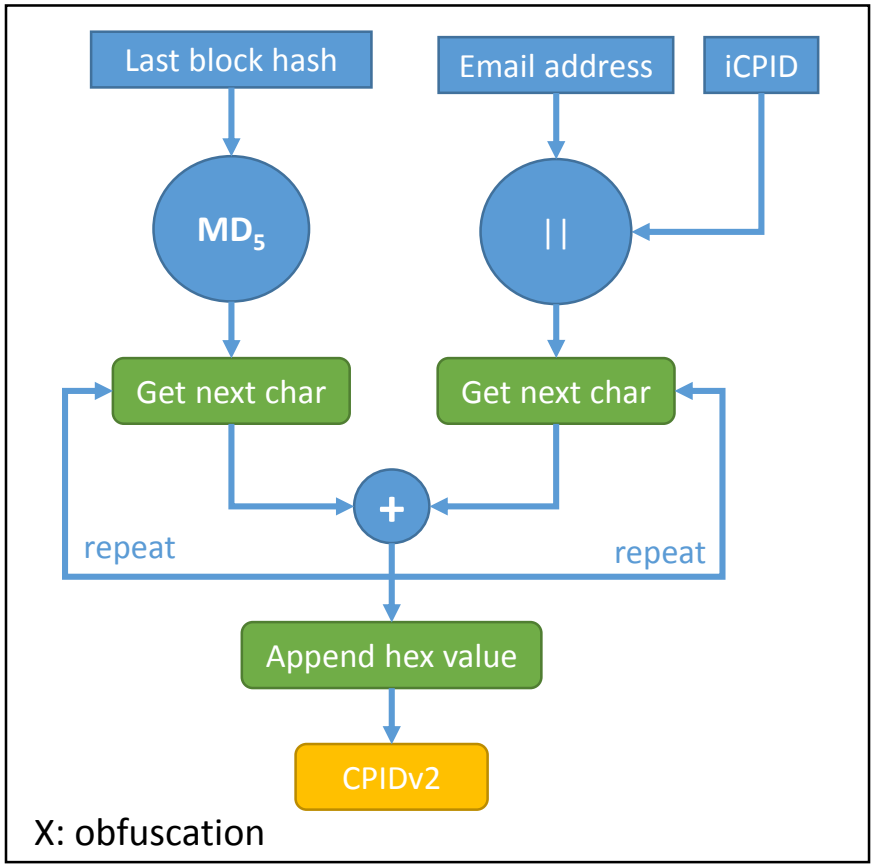
The security flaw



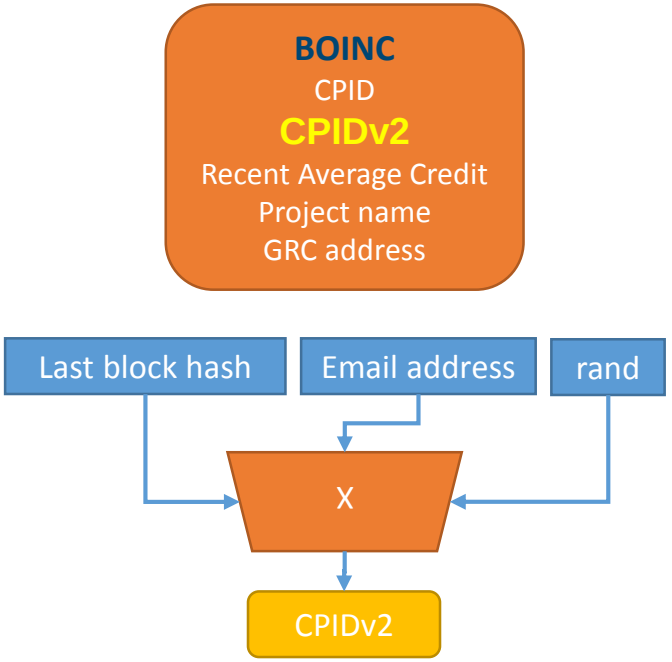


The security flaw





The security flaw



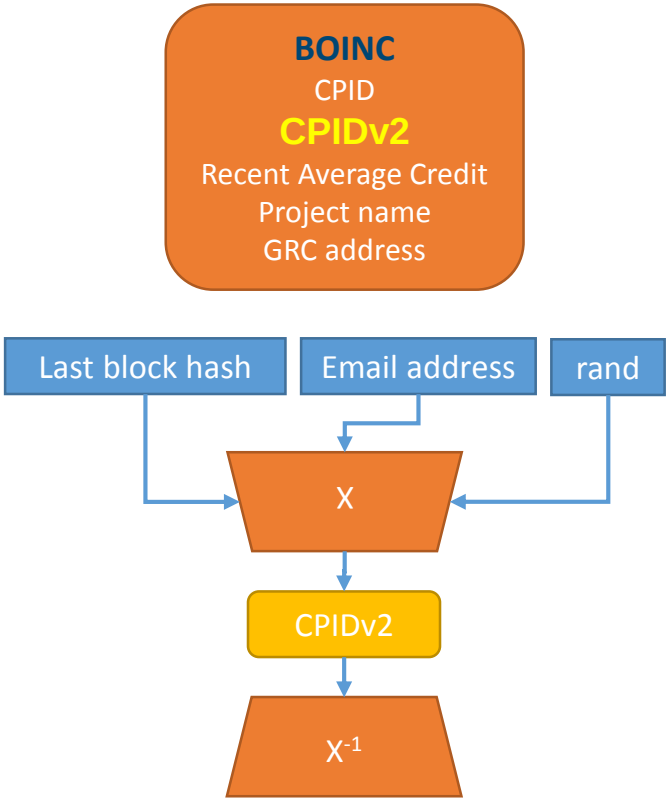
Identity proof

- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)

The security flaw

Identity proof

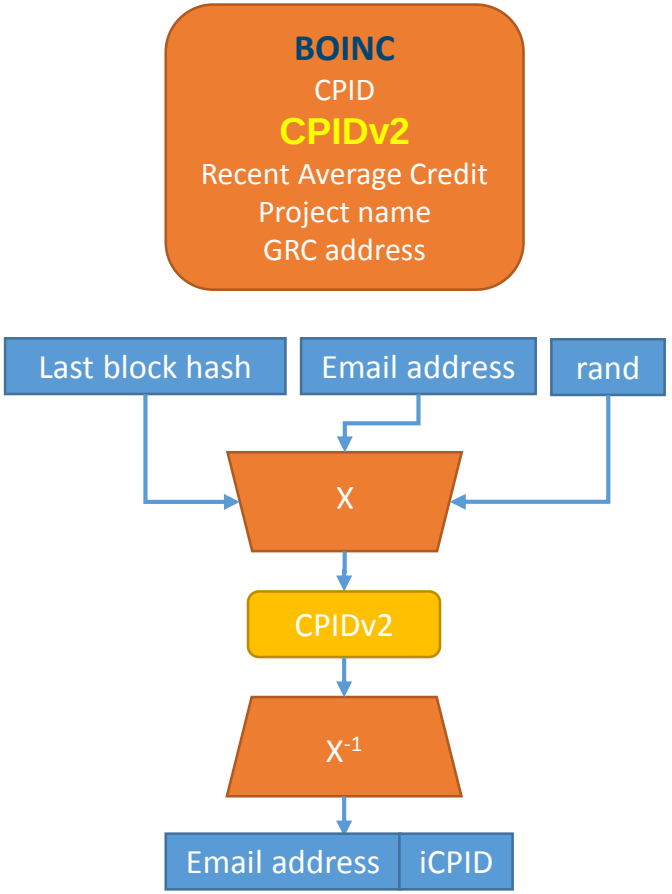
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

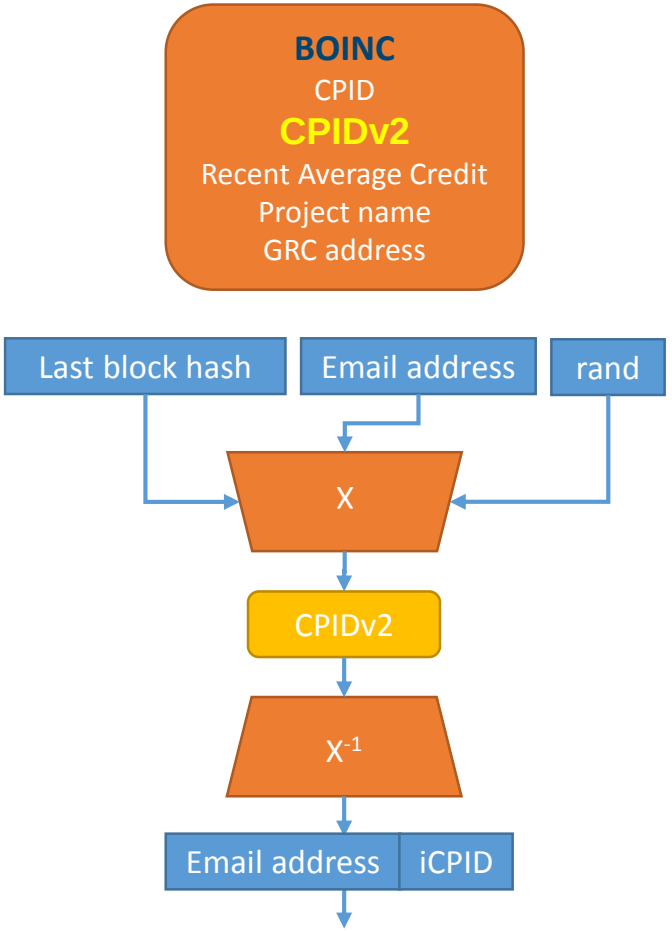
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

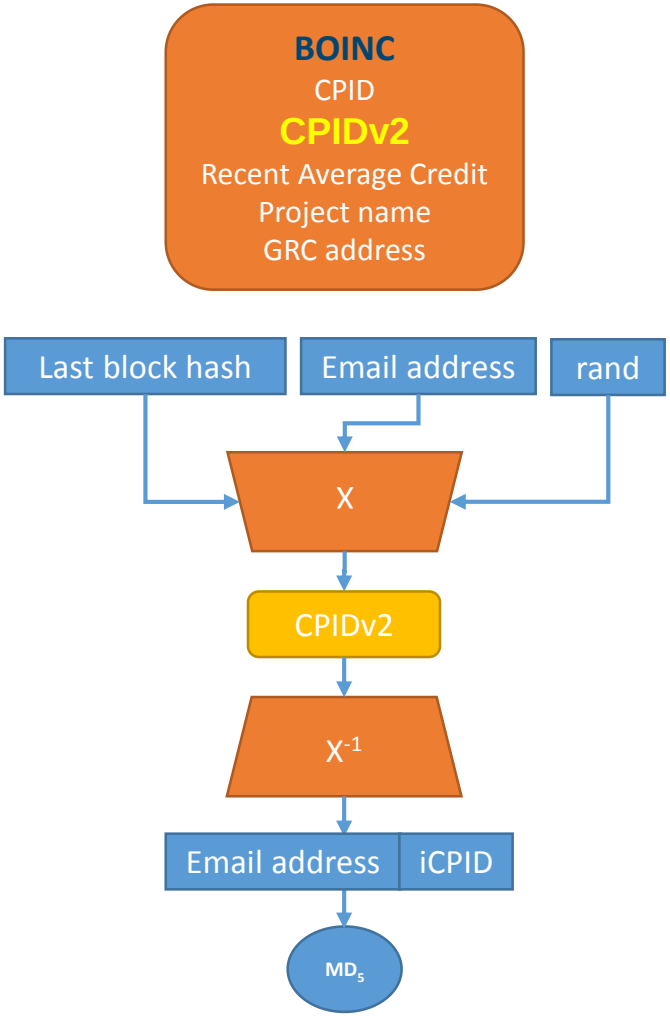
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

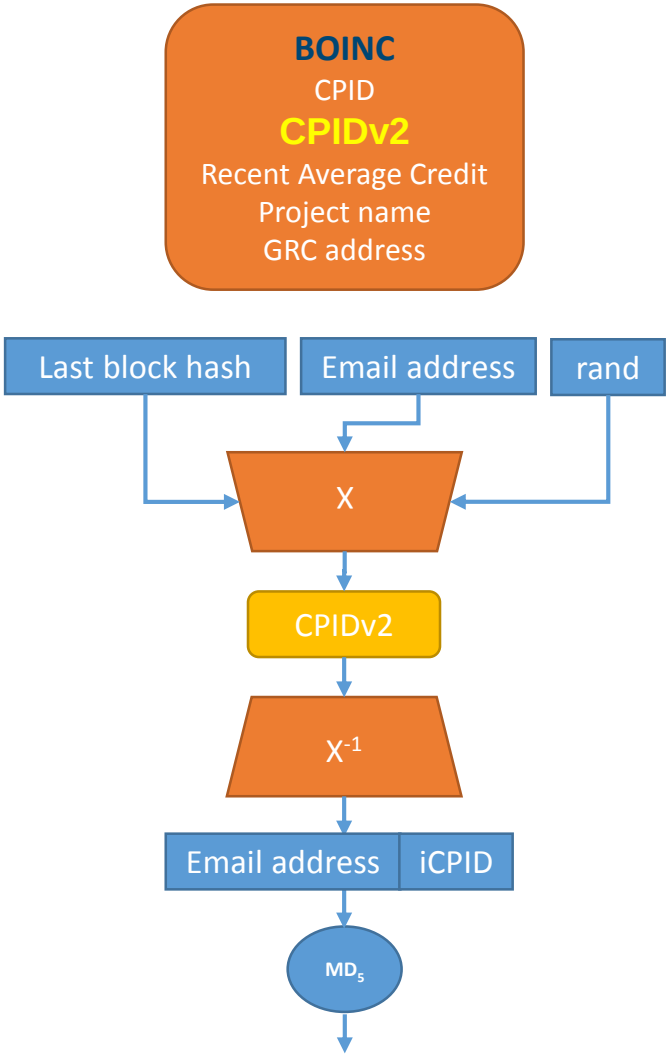
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

Identity proof

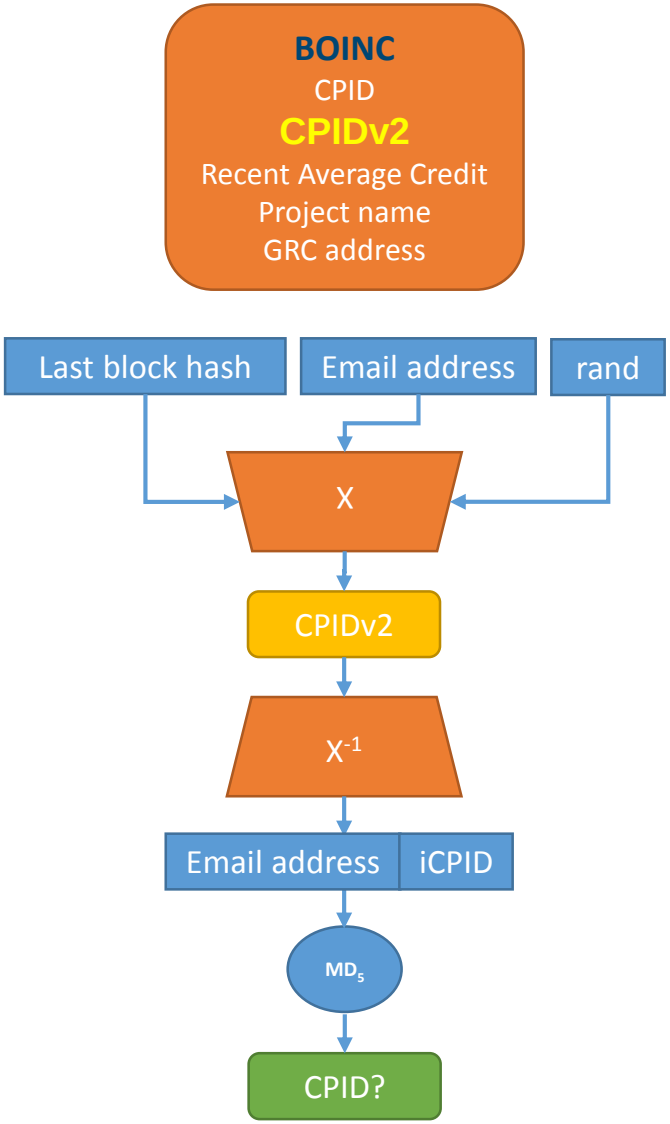
- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw

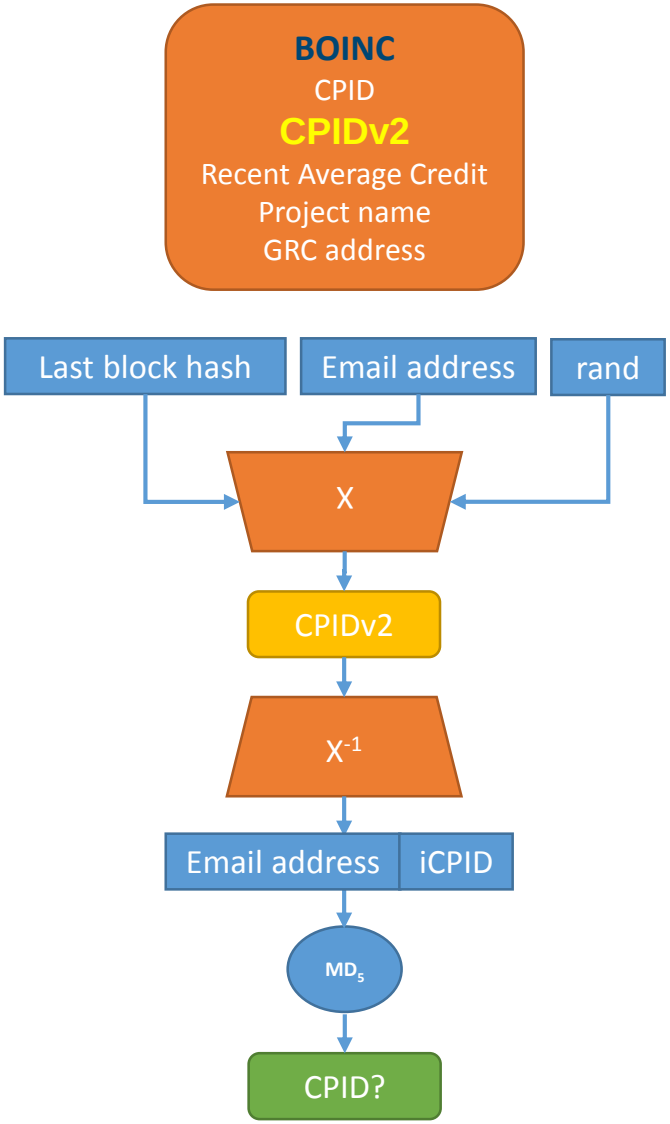
Identity proof

- Problem:
  - Copy CPID from block chain
  - Steal rewards from researchers
  - How to proof miner = researcher?
- Gridcoins solution: CPIDv2
- Reminder:
  - CPID := MD_5(rand||email address)



The security flaw





Implications

- No security at all:
  1. Now we can steal email addresses
  2. We can steal BOINC work from others
     - Get higher reward for free

Steal the block reward

BOINC / Gridcoin infrastructure

Gridcoin block chain

Gridcoin users

Steal the block reward

BOINC / Gridcoin infrastructure

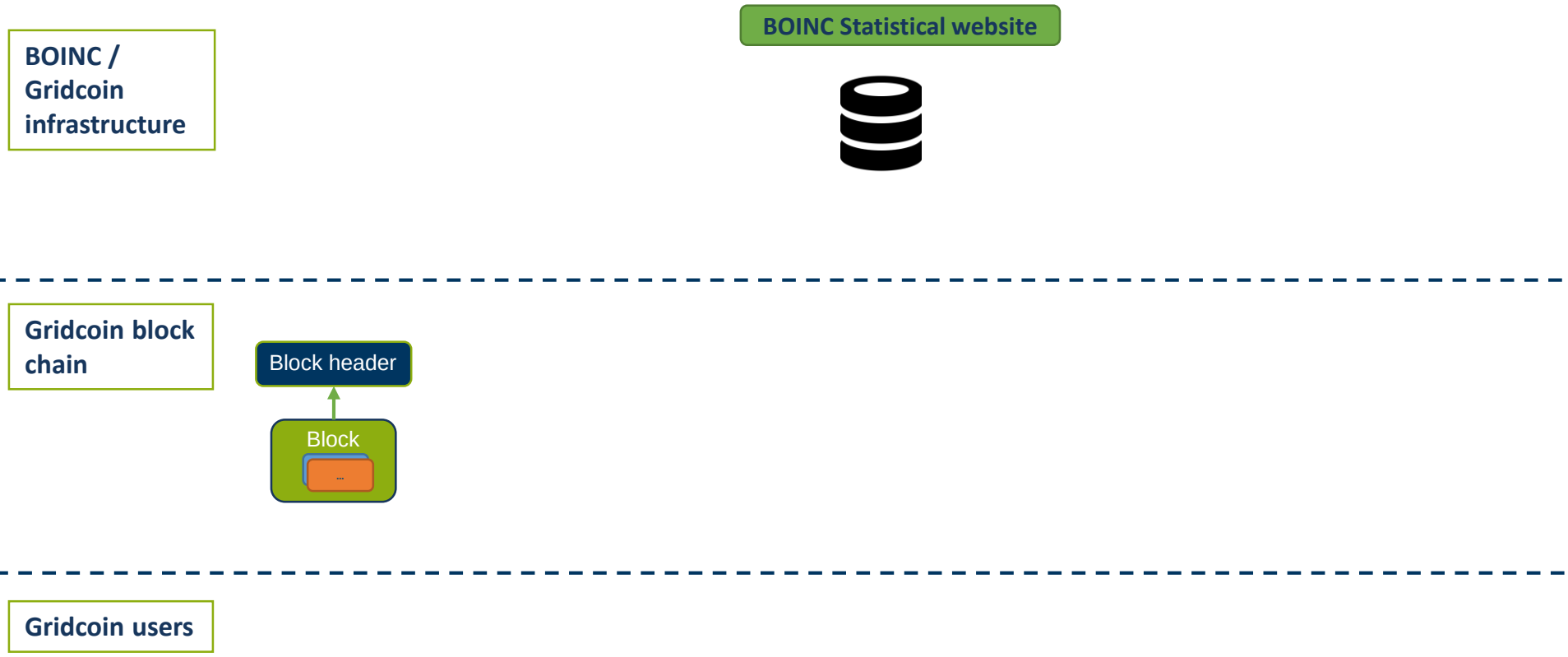
BOINC Statistical website



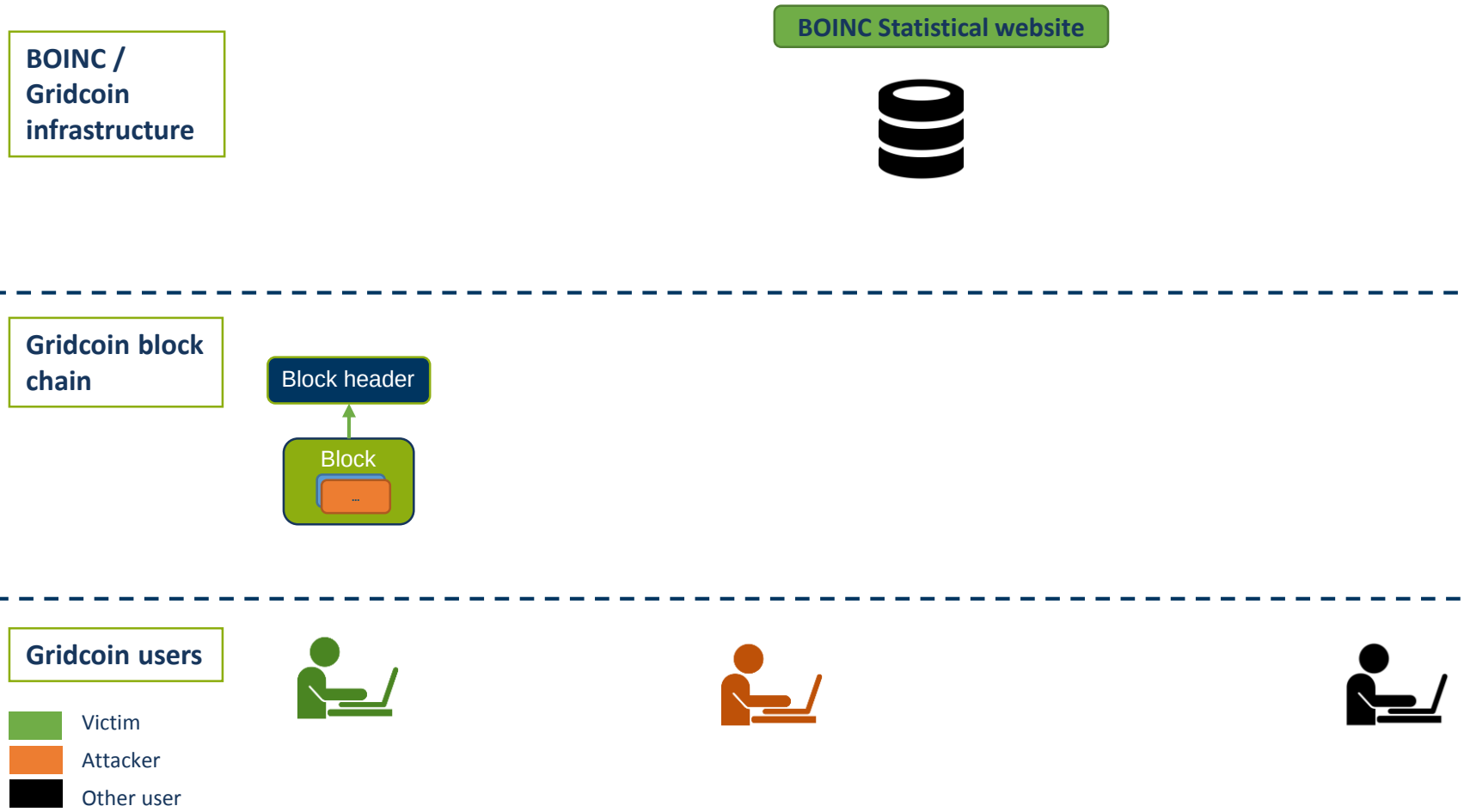
Gridcoin block chain

Gridcoin users

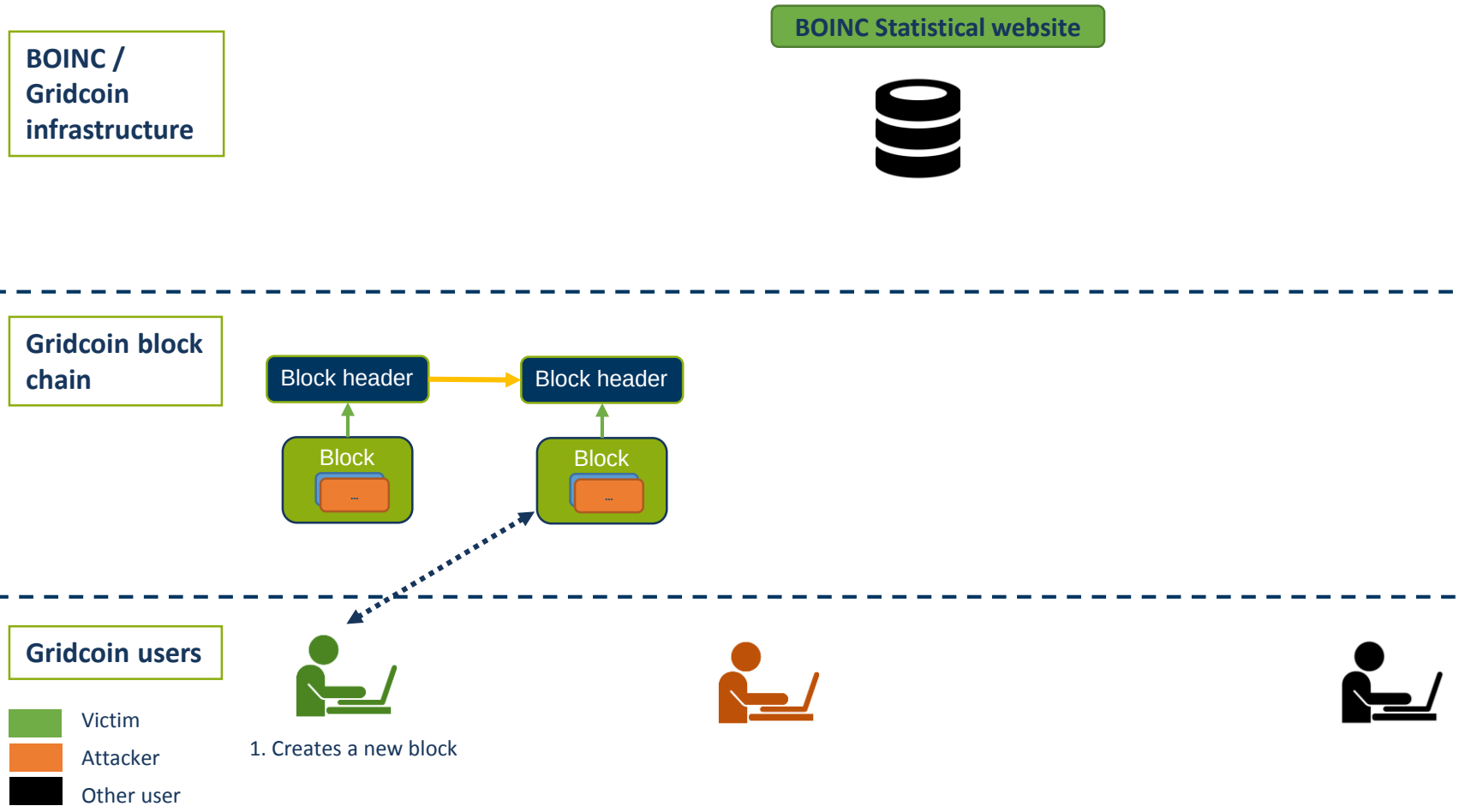
Steal the block reward



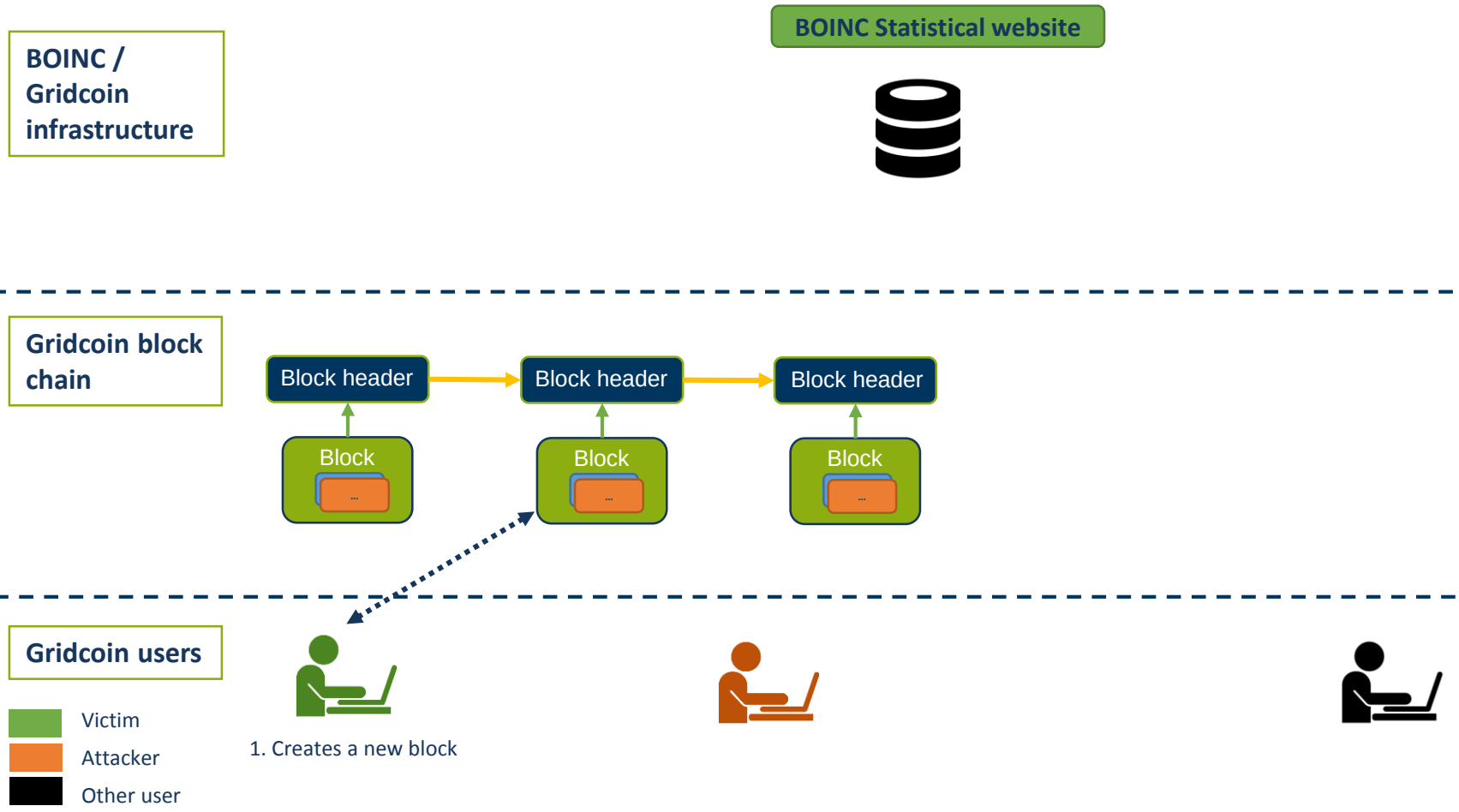
Steal the block reward



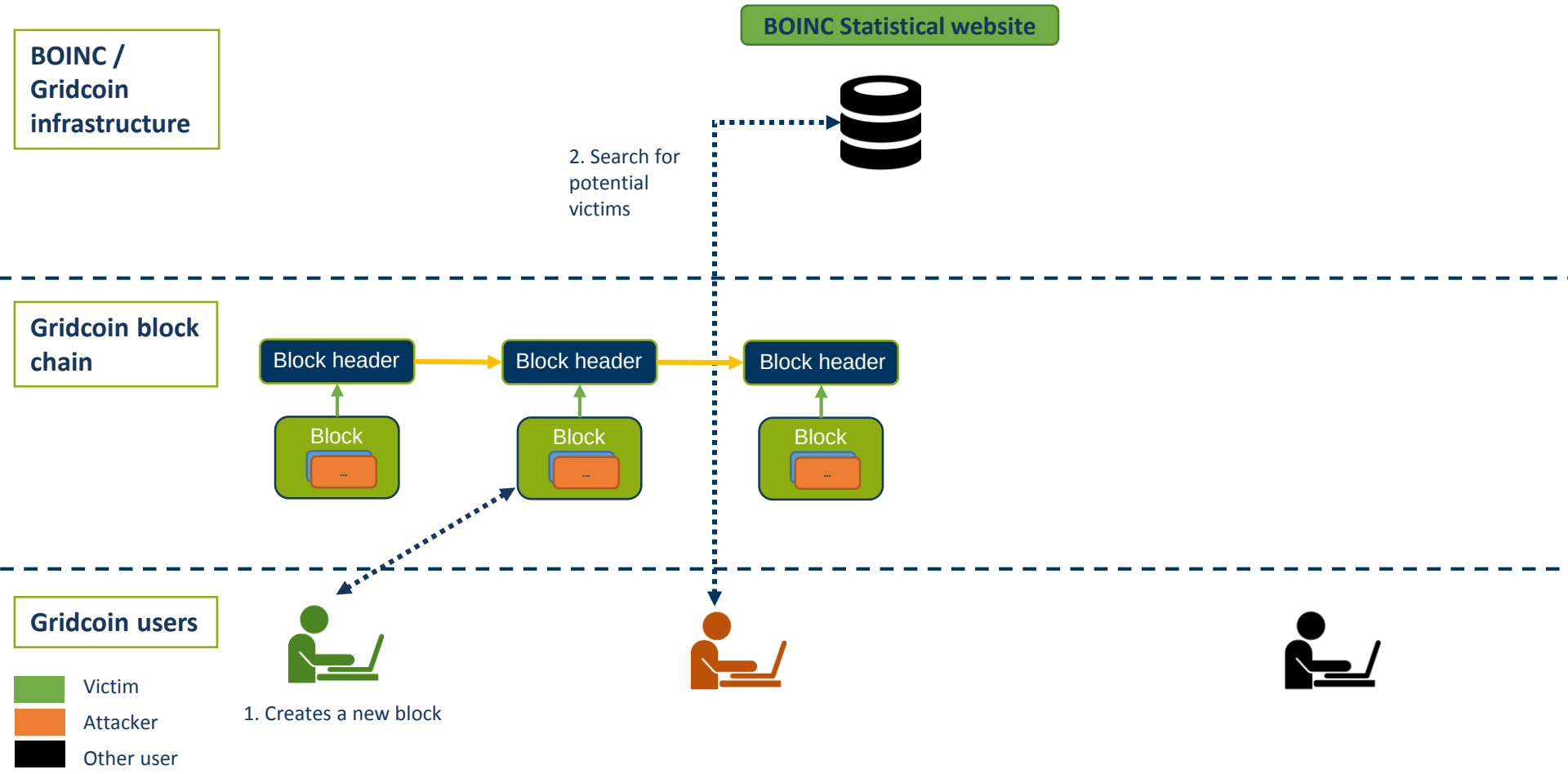
Steal the block reward



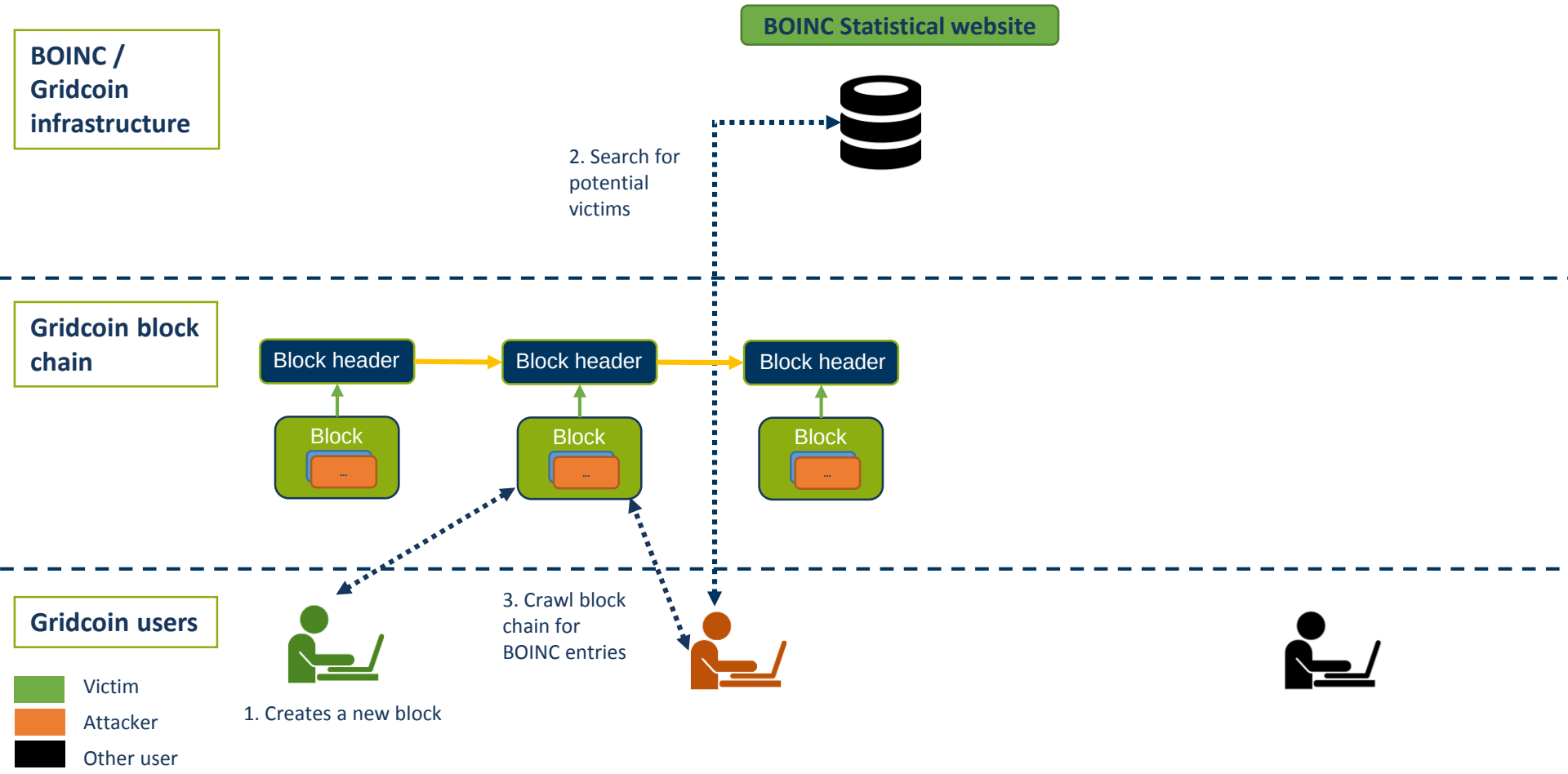
Steal the block reward



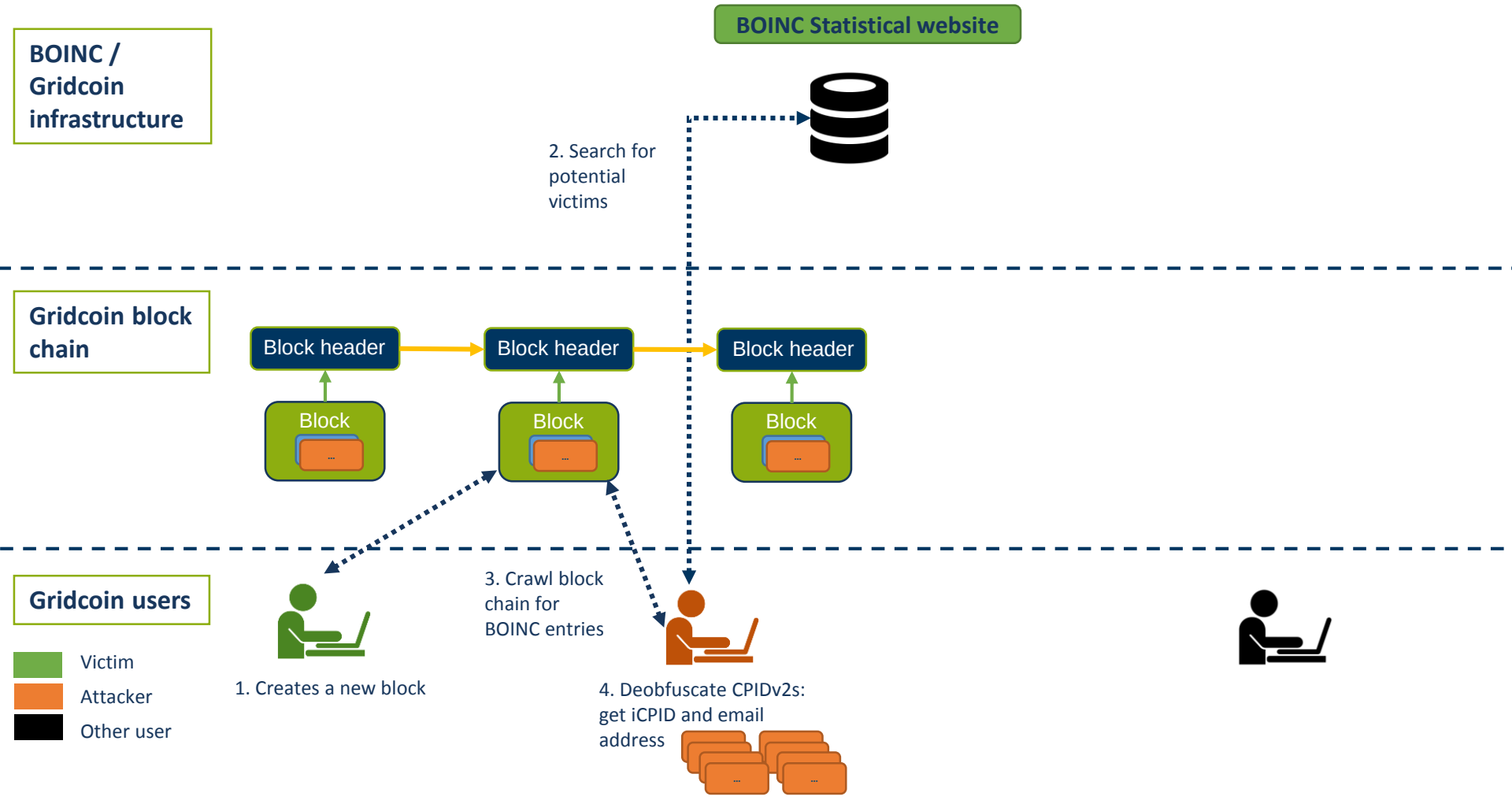
Steal the block reward



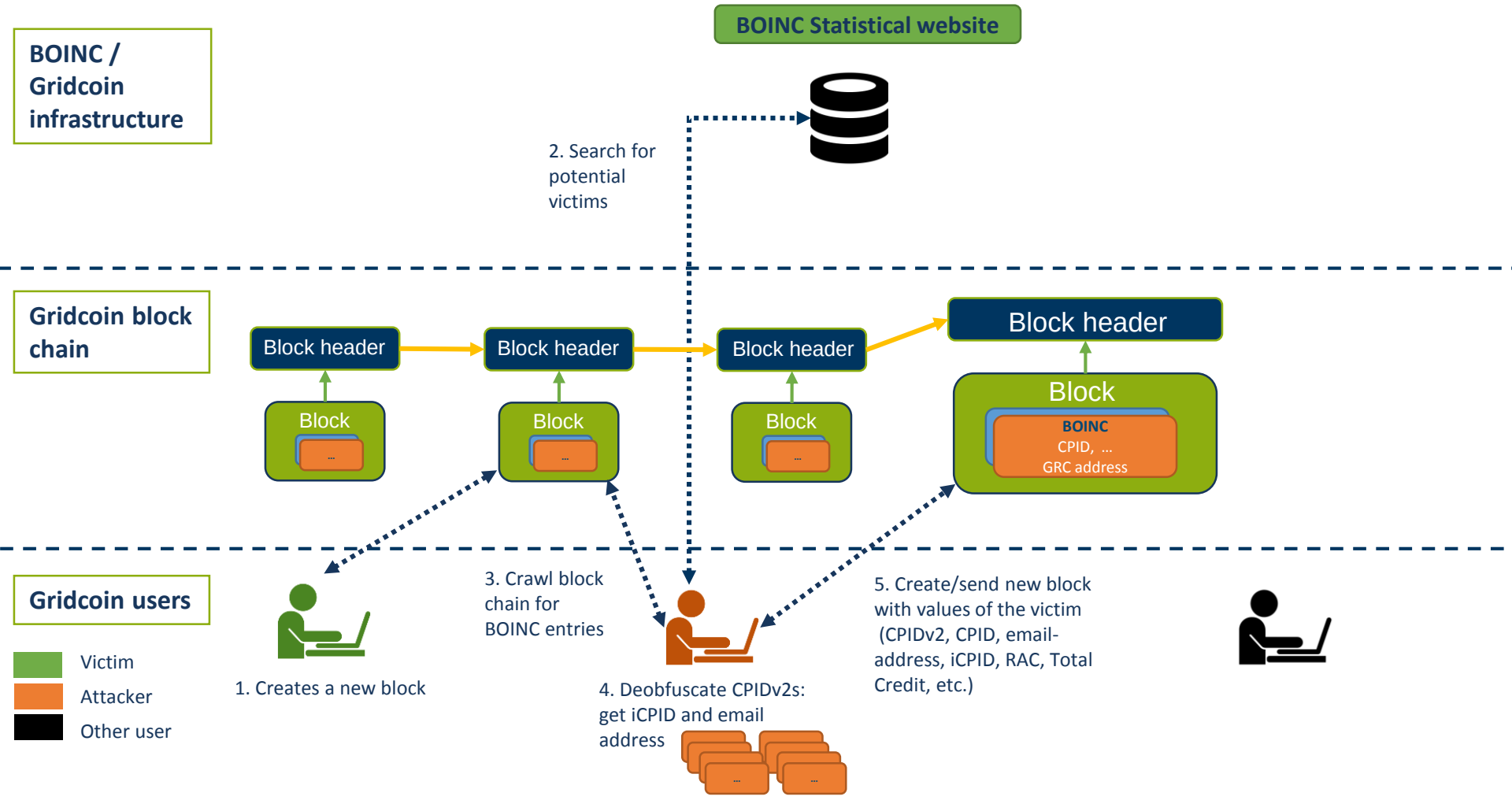
Steal the block reward



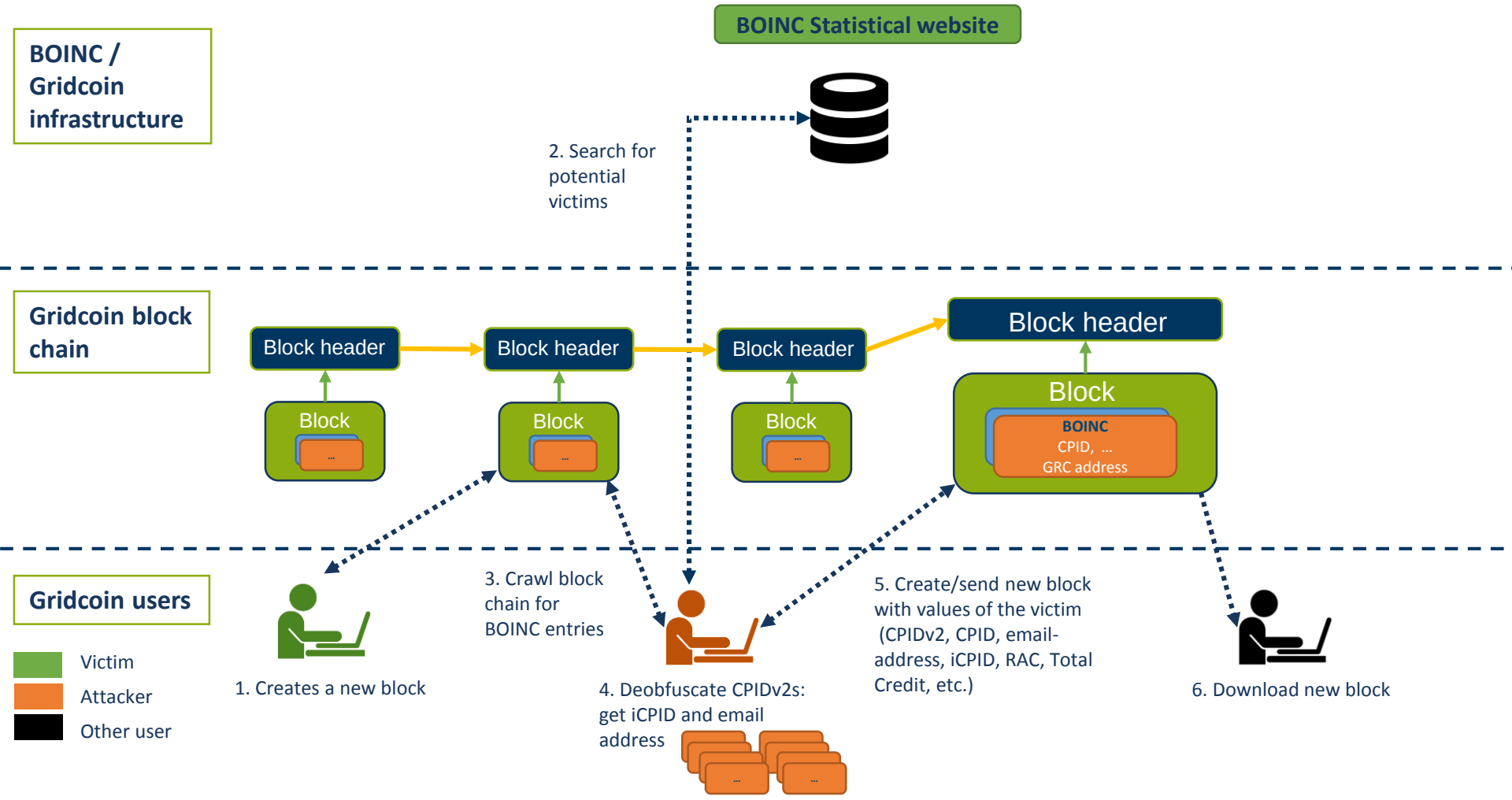
Steal the block reward



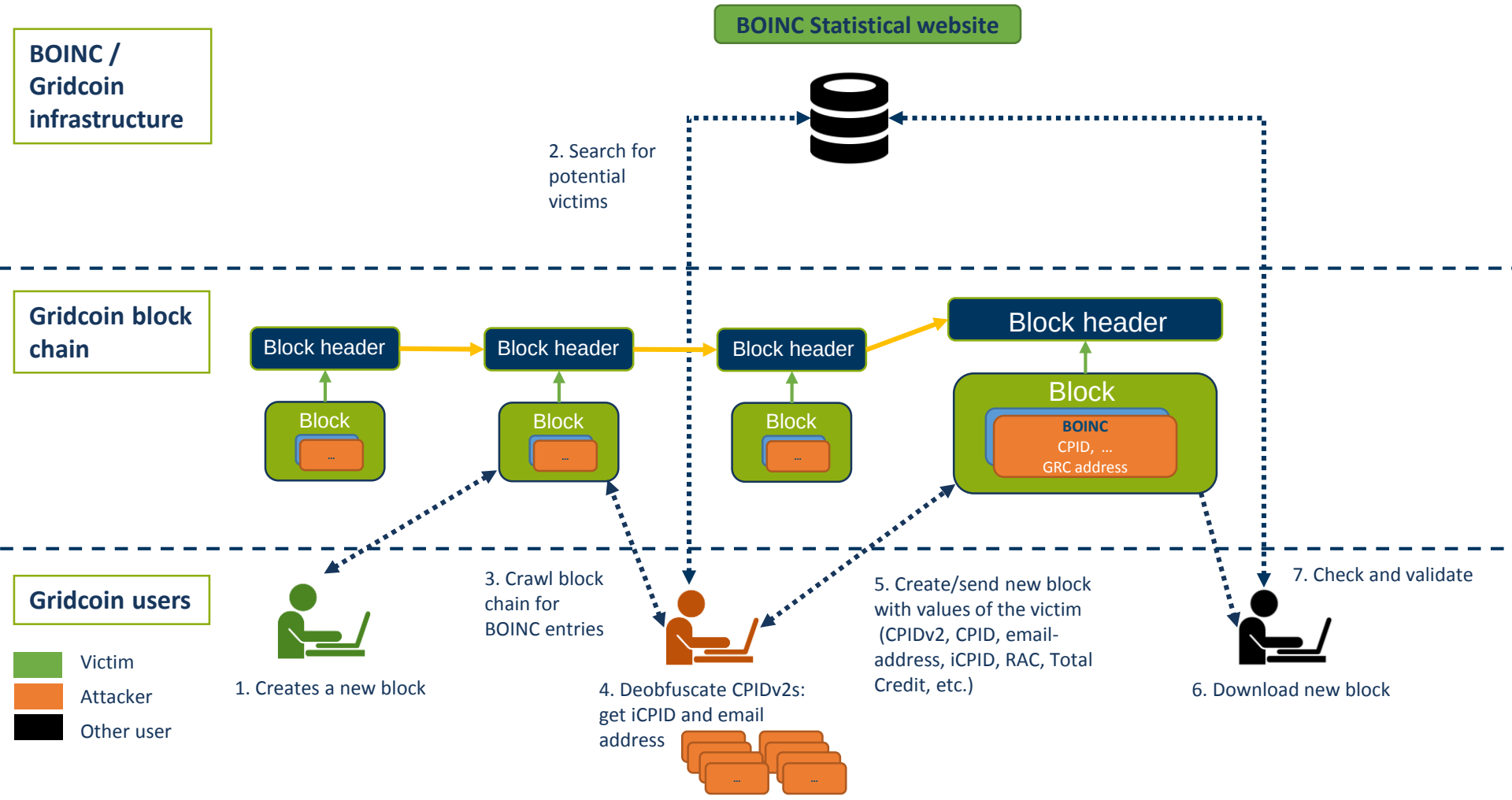
Steal the block reward



Steal the block reward



Steal the block reward



The security flaw

Proof of Concept

- Verified our assumptions
- Created a victim
  - Participate in BOINC (SETI@home)
- We implemented an attacker
- Stole 72.4 Gridcoins from our victim
- Block Numbers:
  - 614249
  - 615189
  - …
  - 622658

Block	Time	PoR	PoS
632862	2016-08-11 21:09:52	4,84	0
631002	2016-08-09 20:45:20	2,91	1,48
630000	2016-08-08 18:50:56	0,87	0
629677	2016-08-08 10:49:52	0,43	0,01
629521	2016-08-08 07:05:04	18,95	0,19
622658	2016-08-01 00:07:12	11	0,7
618716	2016-07-27 18:34:08	5,6	0,33
616844	2016-07-25 18:33:52	3,22	0,31
615810	2016-07-24 14:54:08	0,75	0
615570	2016-07-24 08:30:08	0,34	0
615461	2016-07-24 05:39:28	0,9	0
615189	2016-07-23 22:26:08	1,58	0,18
614715	2016-07-23 09:45:52	1,58	0,01
614249	2016-07-22 21:05:20	41,66	1,86
601680	2016-07-08 23:47:12	0,57	0
601504	2016-07-08 19:13:36	0,33	0,51
601408	2016-07-08 16:33:04	2,05	0
600769	2016-07-08 00:03:28	3,33	0
599717	2016-07-06 20:27:44	3,22	0
598773	2016-07-05 19:54:56	0,14	0,21

The Fix

For the 2. Attack

The Fix

For the 2. Attack

- Allows special transactions (beacons)
- We use this beacons

The Fix

For the 2. Attack

- Allows special transactions (beacons)
- We use this beacons
- Tie CPID to public key

The Fix

For the 2. Attack

- Allows special transactions (beacons)
- We use this beacons
- Tie CPID to public key
- Mining requires advertising the CPID beforehand

The Fix

For the 2. Attack

- Allows special transactions (beacons)
- We use this beacons
- Tie CPID to public key
- Mining requires advertising the CPID beforehand
- Reward requires signature from corr. private key

The Fix

For the 2. Attack

- Allows special transactions (beacons)
- We use this beacons
- Tie CPID to public key
- Mining requires advertising the CPID beforehand
- Reward requires signature from corr. private key
- Beacon is valid for 6 months

The Fix

For the 2. Attack

- Allows special transactions (beacons)
- We use this beacons
- Tie CPID to public key
- Mining requires advertising the CPID beforehand
- Reward requires signature from corr. private key
- Beacon is valid for 6 months

$$Nonce \xleftarrow{\$} \{0,1\}^{128}$$

$$Input := \mathsf{H}(eCPID) \;||\; PK_1 \;||\; Nonce$$

$$FirstBeacon := \mathsf{Sign}_{SK_1}(Input) \;||\; Input$$

The Fix

For the 2. Attack

- Allows special transactions (beacons)
- We use this beacons
- Tie CPID to public key
- Mining requires advertising the CPID beforehand
- Reward requires signature from corr. private key
- Beacon is valid for 6 months

$$Nonce \xleftarrow{\$} \{0,1\}^{128}$$
$$Input := \mathsf{H}(eCPID) \;||\; PK_1 \;||\; Nonce$$
$$FirstBeacon := \mathsf{Sign}_{SK_1}(Input) \;||\; Input$$

$$Nonce \xleftarrow{\$} \{0,1\}^{128}$$
$$Input := \mathsf{H}(eCPID) \;||\; PK_2 \;||\; Nonce$$
$$UpdateBeacon := \mathsf{Sign}_{SK_1}(Input) \;||\; Input$$

Responsible disclosure

- 1. responsible disclosure (design vuln)
  - Informed developer on 09/14/2016
  - Supplied our attacks and fixes (see the paper)
  - Quickly got a responded (1 day)
    - First, they did not understand the fix
  - In the end they fixed it in version 3.5.8.7
    - released on 03/26/2017

Responsible disclosure

- Gridcoin developer != secure coder or protocol security experts
- They included new implementation vulnerabilities with their new version
  - We tried to fix the new vulnerabilities
  - Developer never answered to our 2. responsible disclosure
  - See gridcoin-attacks.org for these 0-days
  - Result: do not use Gridoin anymore, it is insecure!

Conclusion

Conclusion

1. We still can steal reward and get the email addresses of many Gridcoin BOINC user
2. Security researcher look @ Altcoins, there are plenty of them
   - Altcoins TOP 11 to 847 are worth > $16 billion
3. Still hard challenge:
   - Create cryptocurrency with generic meaningful computations as PoW
4. Gridcoin is totally vulnerable and probably will never achieve better security
   - See gridcoin-attacks.org

Contact

Martin Grothe | Twitter: @ashitaka007 | gridcoin-attacks.org

Chair for Network and Data Security
Horst-Görtz Institut
Ruhr-Universität Bochum

www.nds.rub.de/mgrothe
www.nds.rub.de/jsomorovsky
www.nds.rub.de/joerg-schwenk

Fixing Gridcoin with the beacon system

BOINC / Gridcoin infrastructure

Gridcoin block chain

Gridcoin users

Fixing Gridcoin with the beacon system

BOINC / Gridcoin infrastructure

Gridcoin block chain

Gridcoin users





Fixing Gridcoin with the beacon system

BOINC / Gridcoin infrastructure

BOINC project A

Statistical website

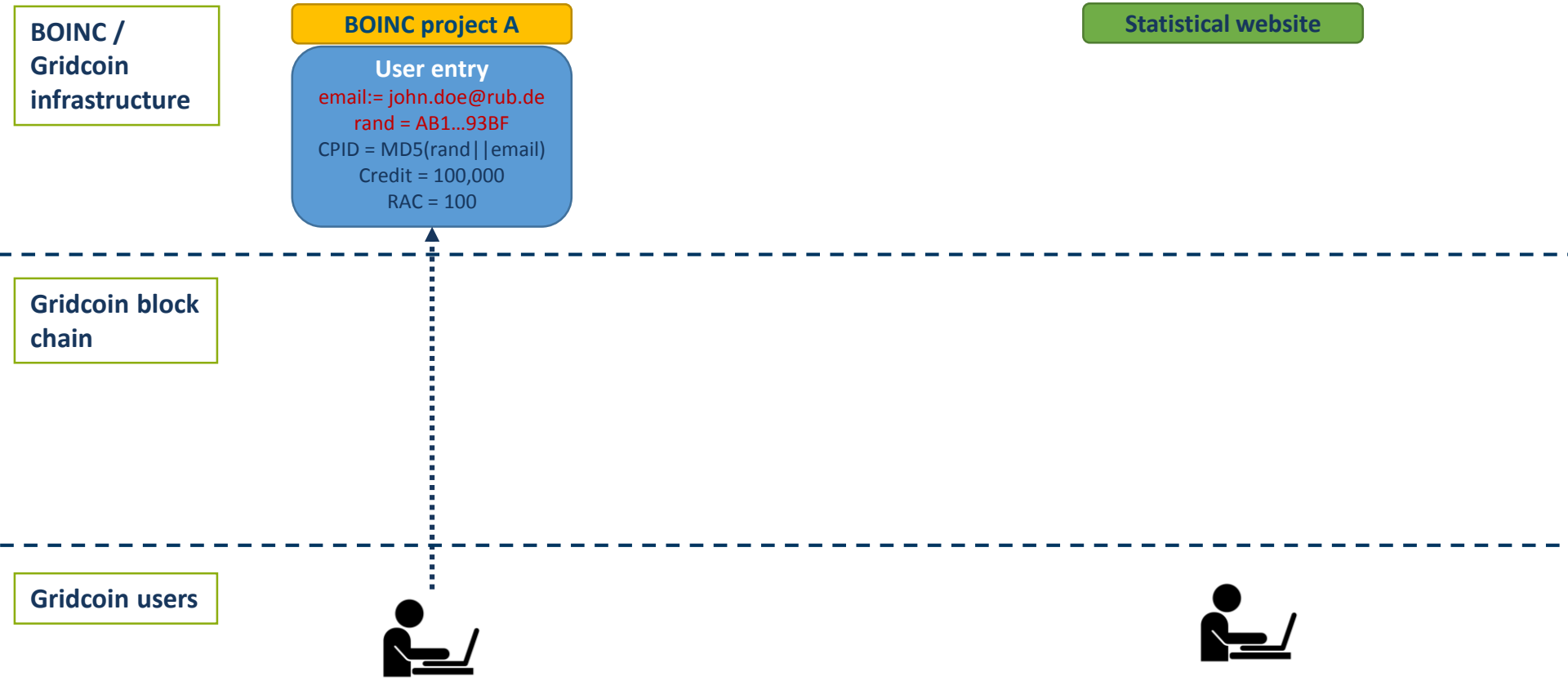
Gridcoin block chain

Gridcoin users

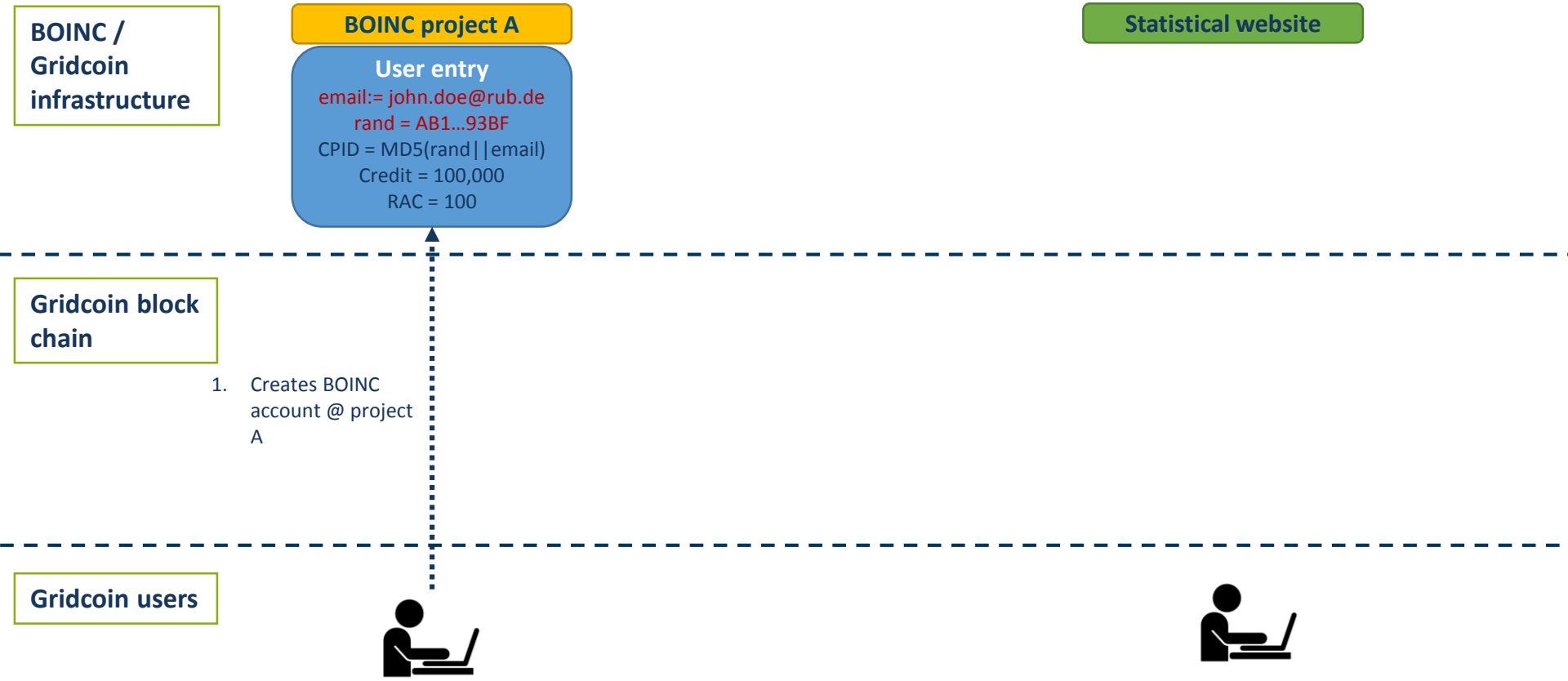




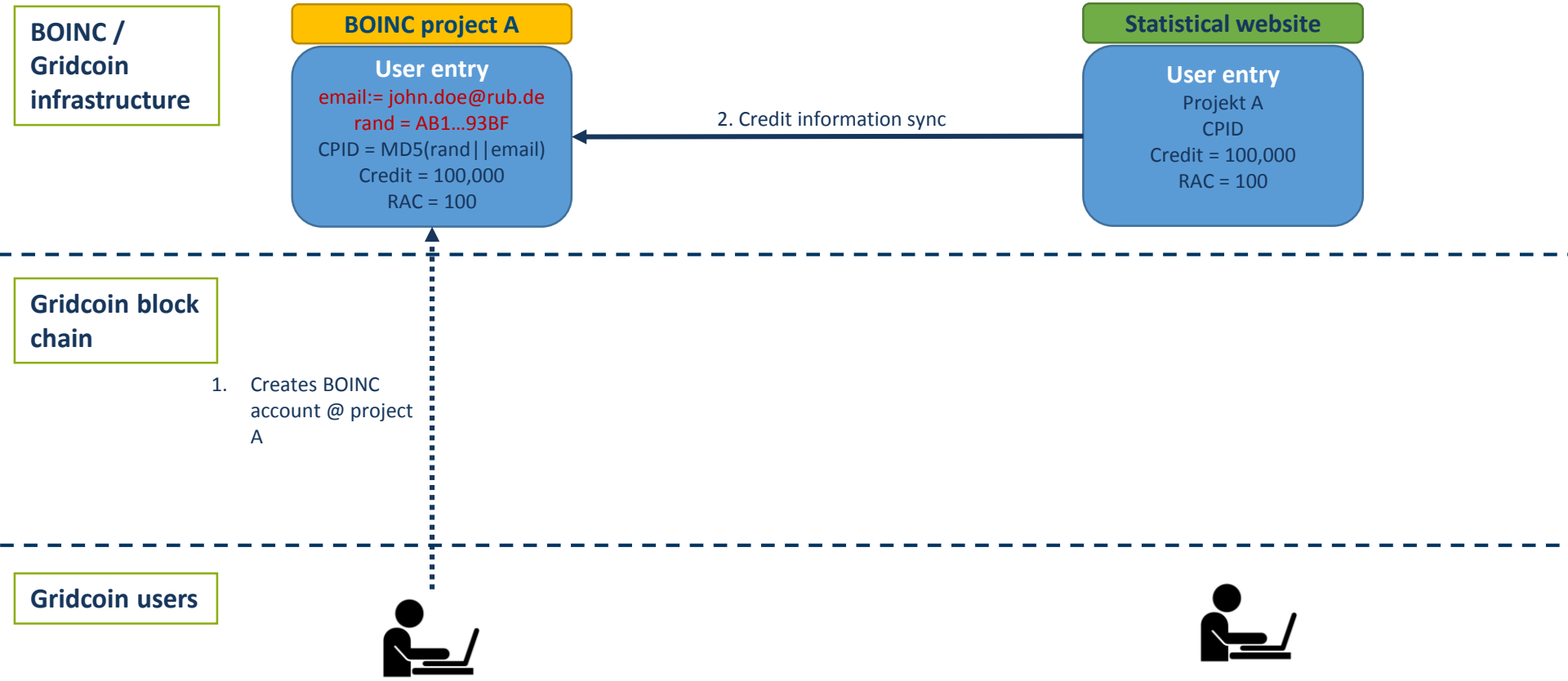
Fixing Gridcoin with the beacon system



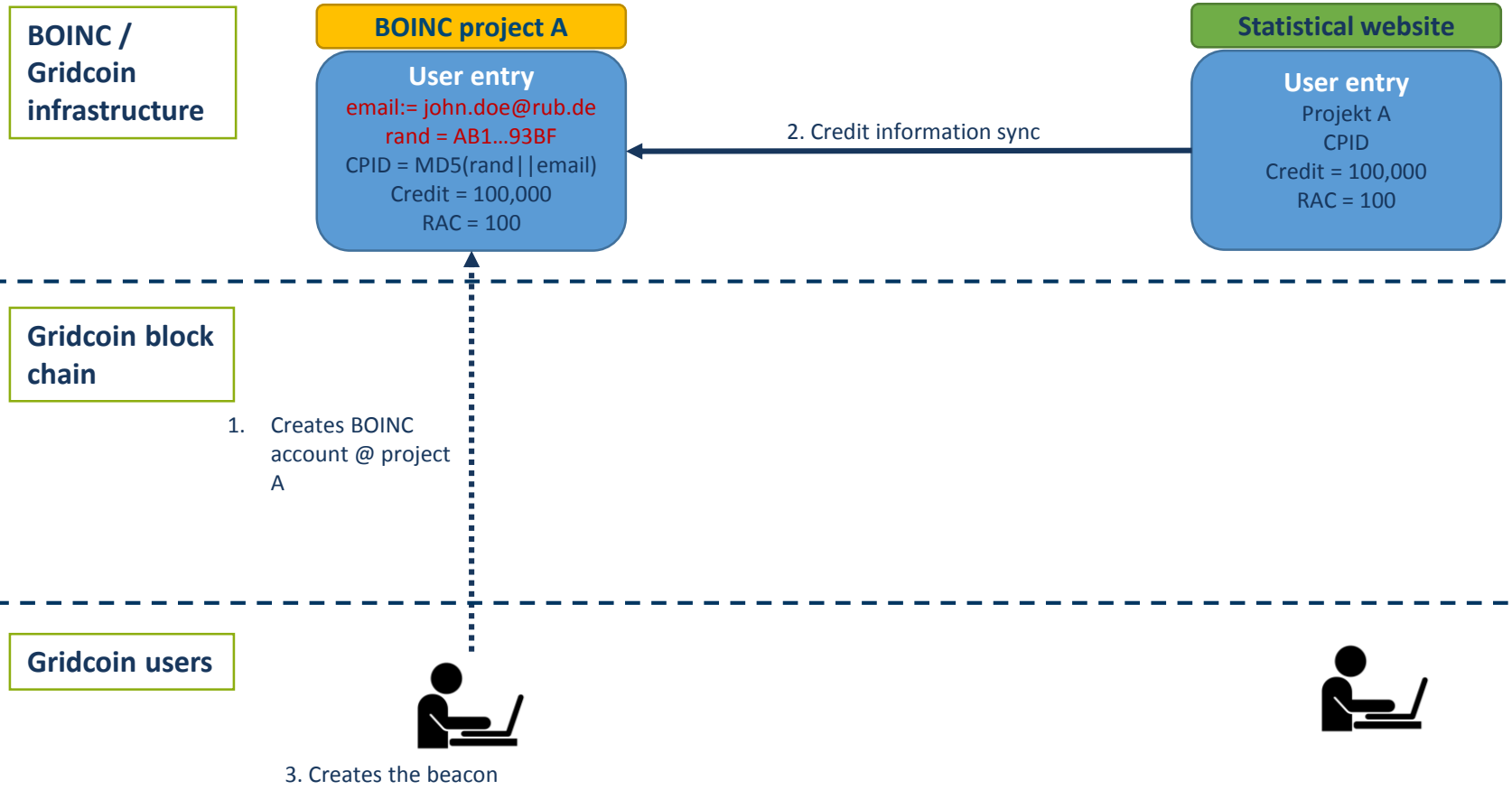
Fixing Gridcoin with the beacon system



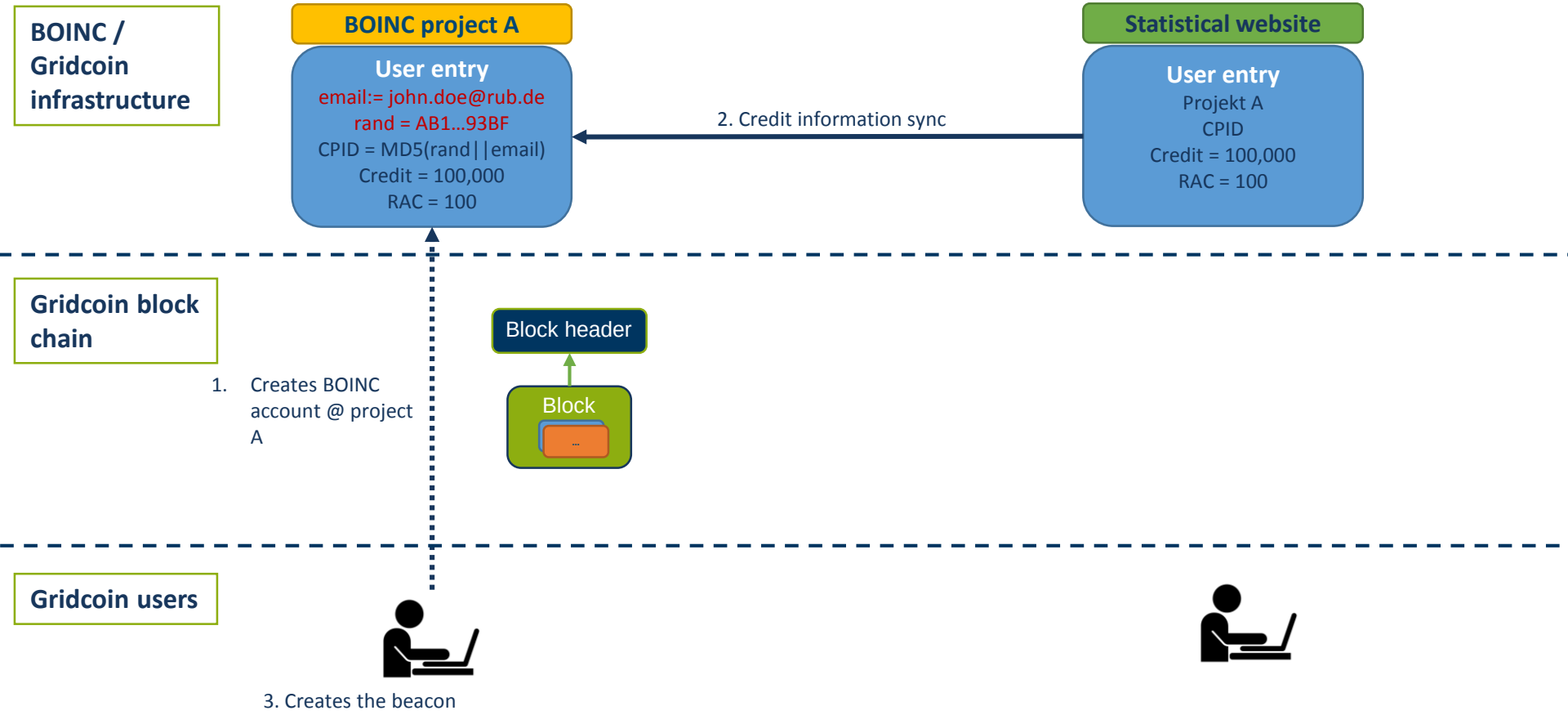
Fixing Gridcoin with the beacon system



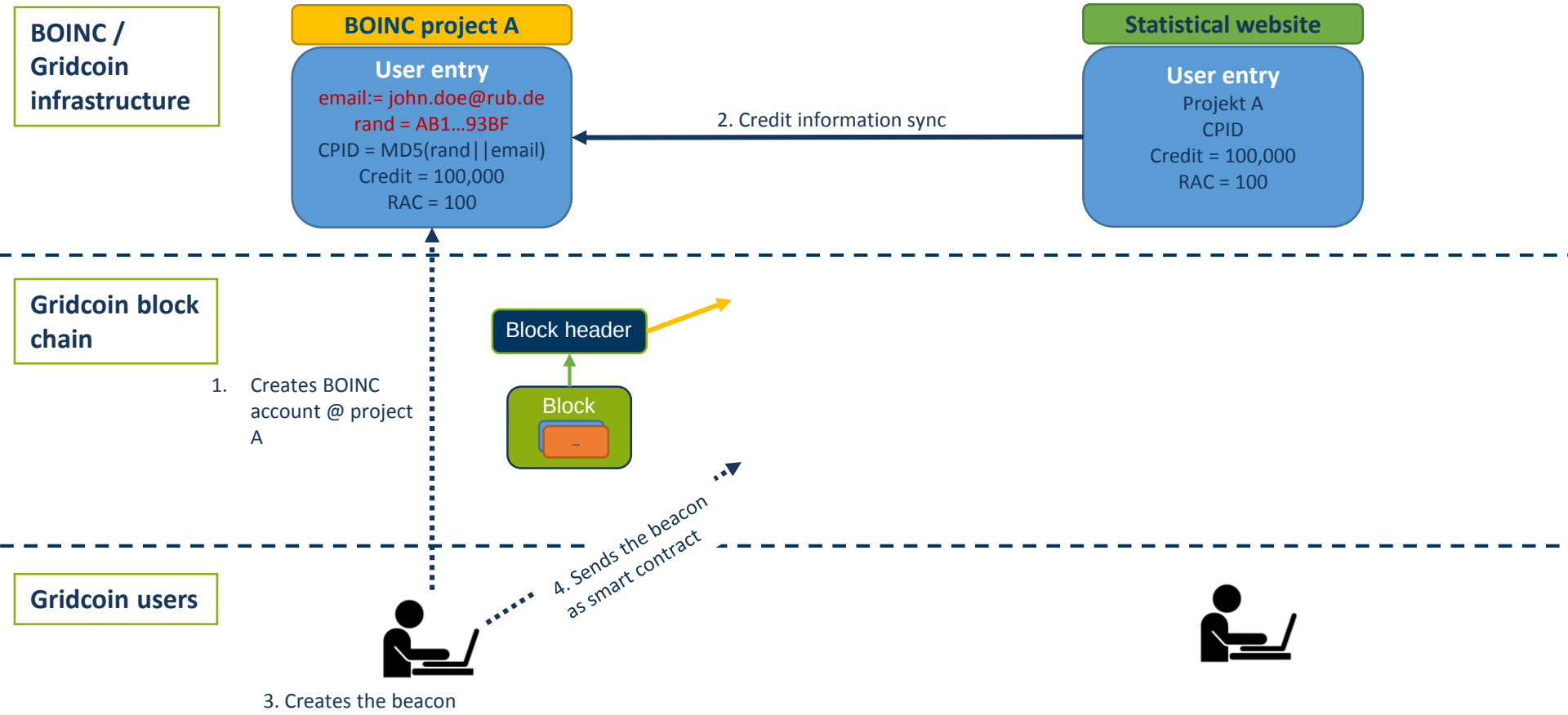
Fixing Gridcoin with the beacon system



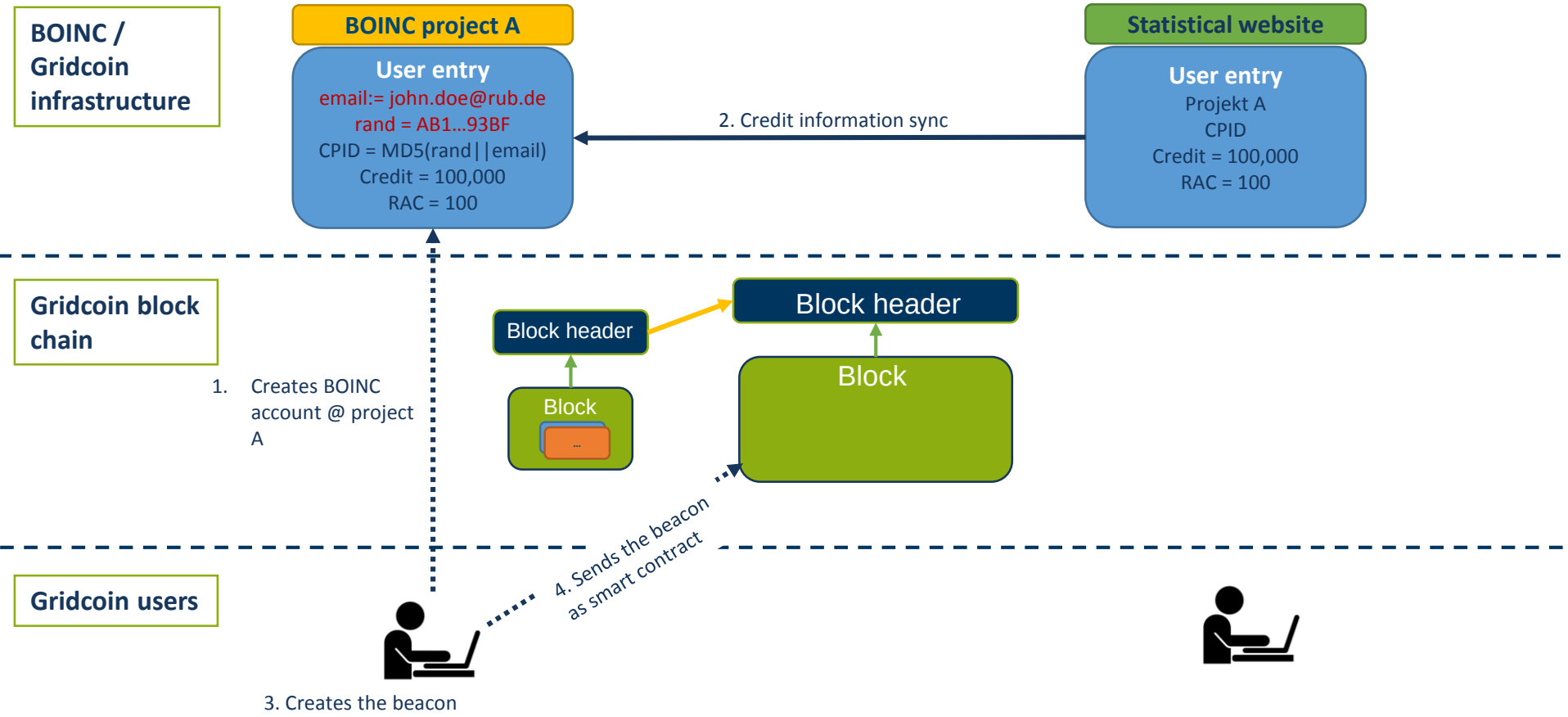
Fixing Gridcoin with the beacon system



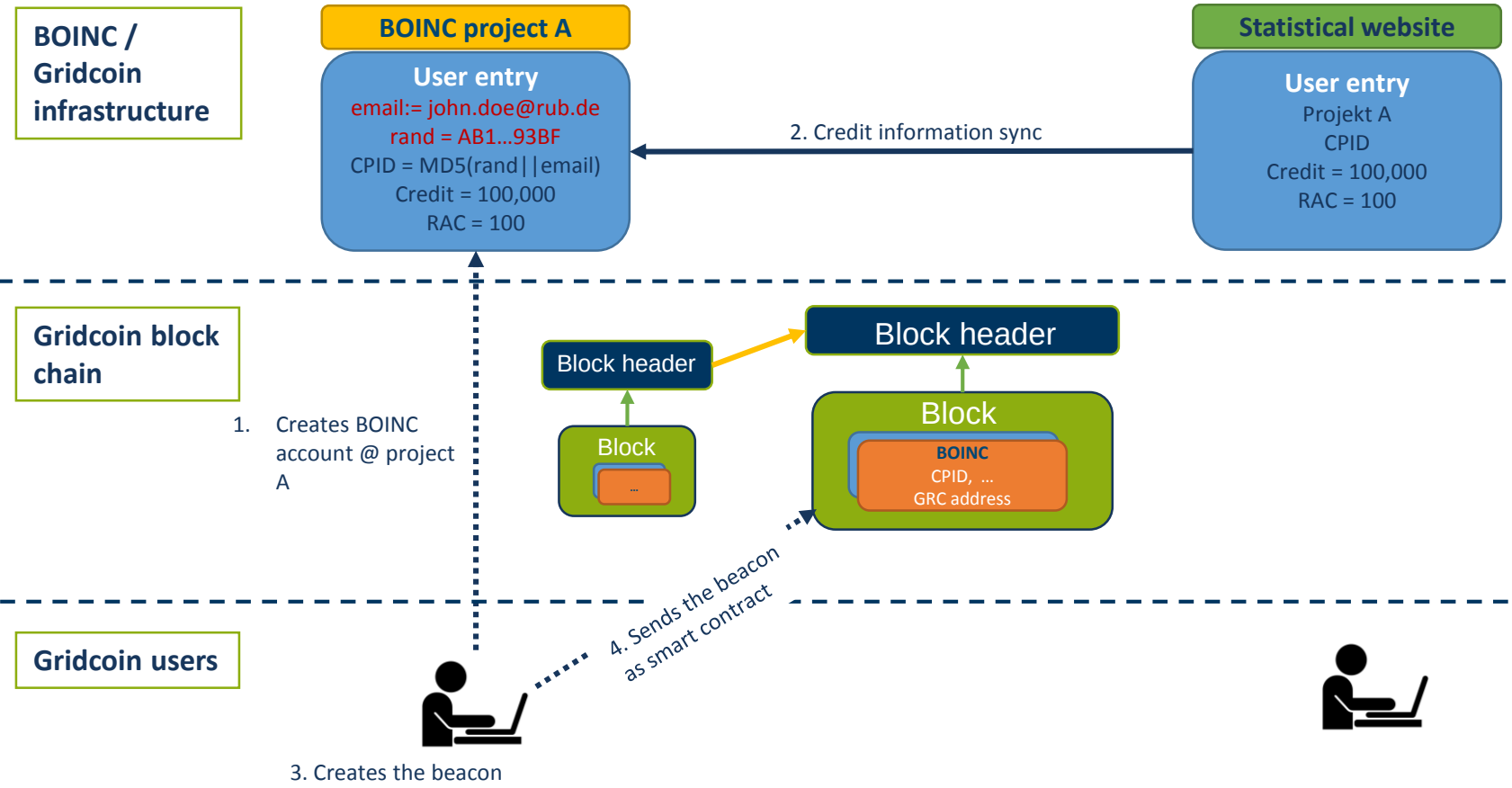
Fixing Gridcoin with the beacon system



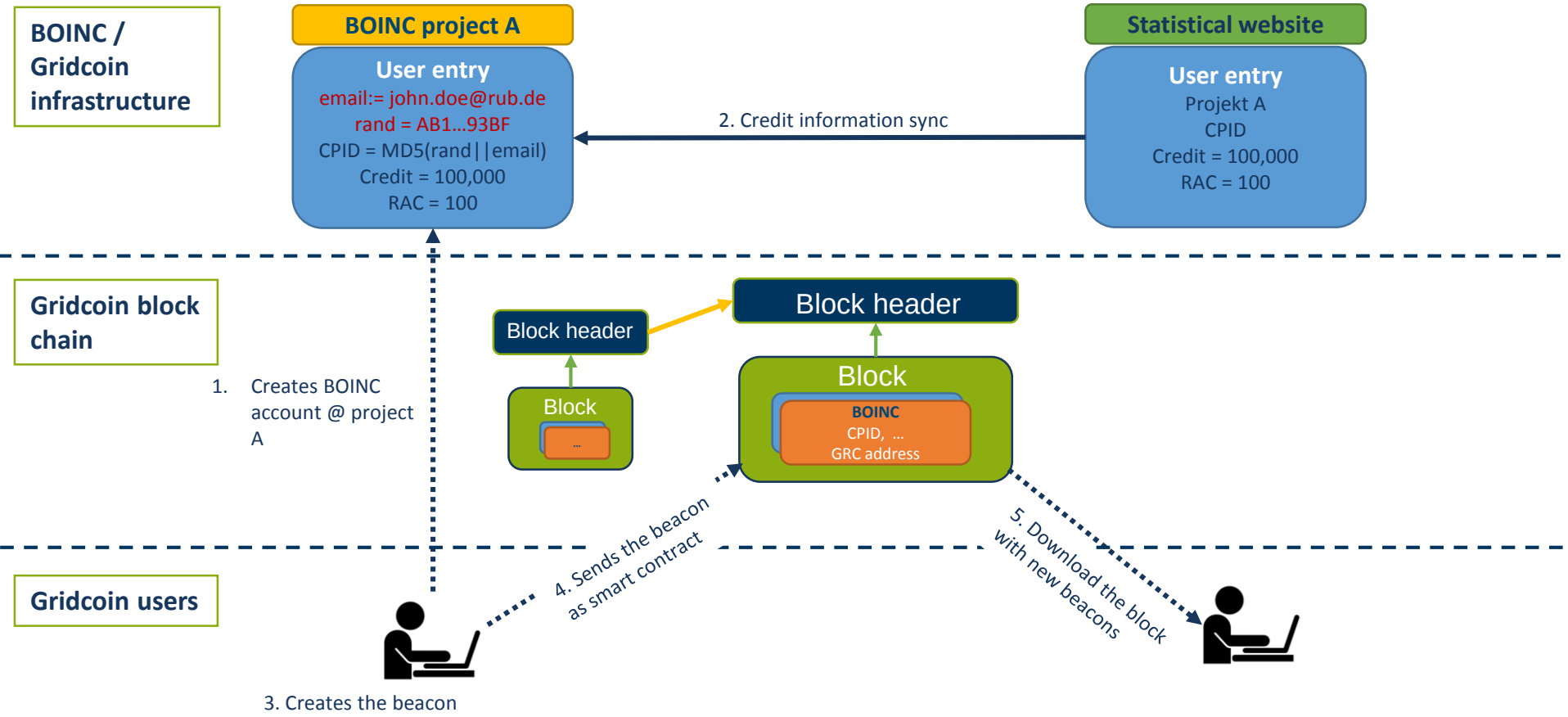
Fixing Gridcoin with the beacon system



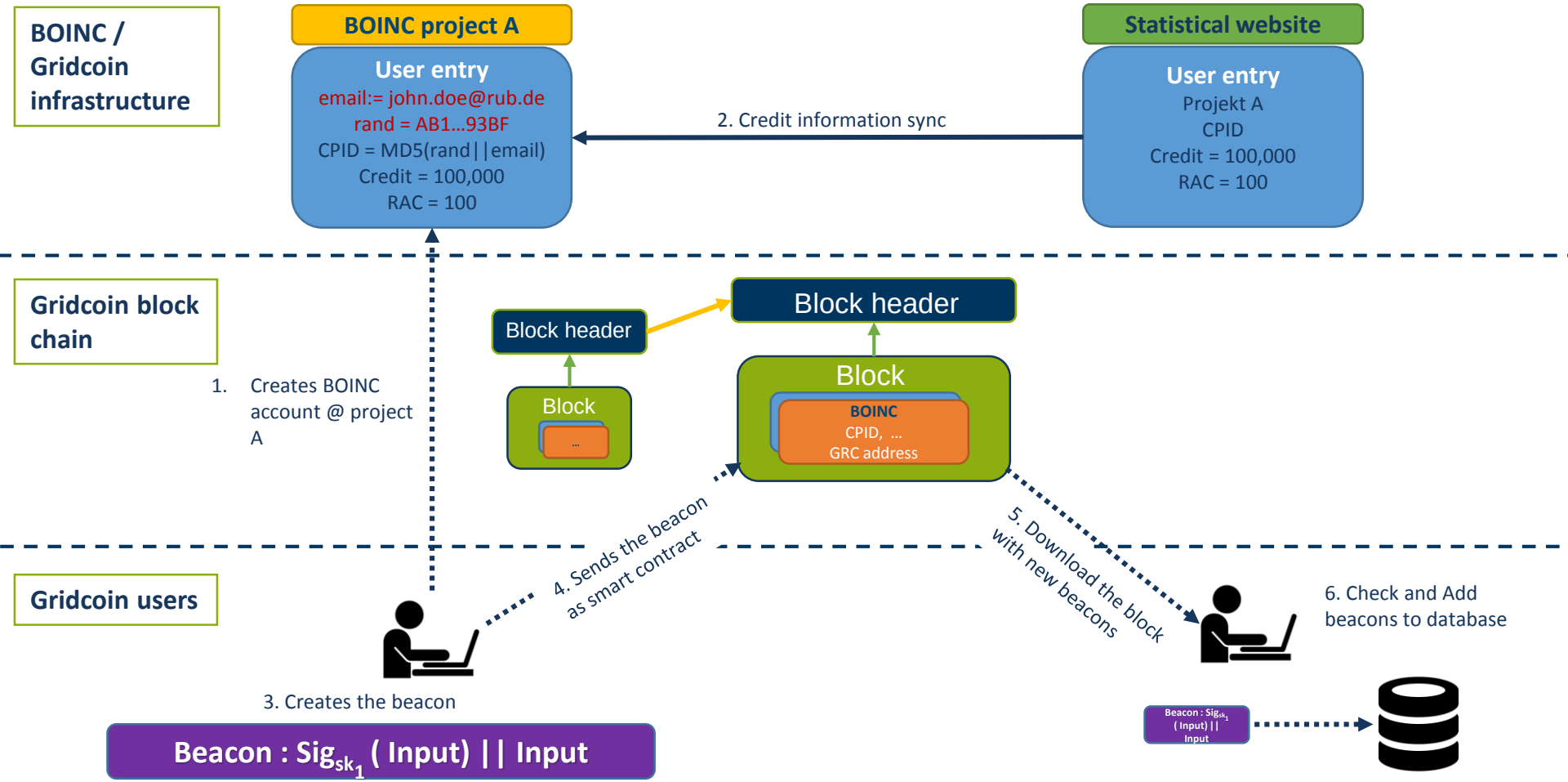
Fixing Gridcoin with the beacon system



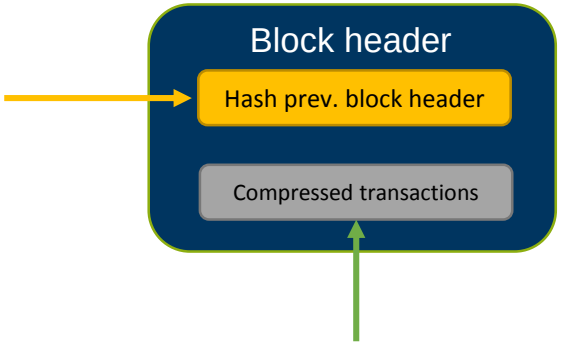
Fixing Gridcoin with the beacon system



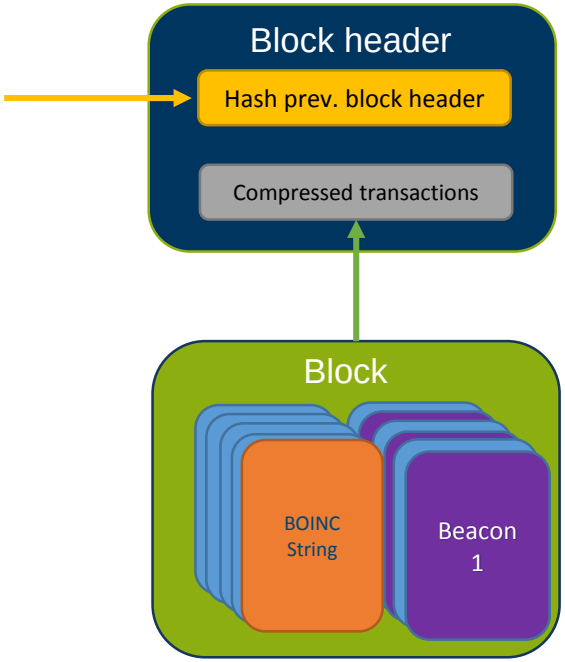
Fixing Gridcoin with the beacon system



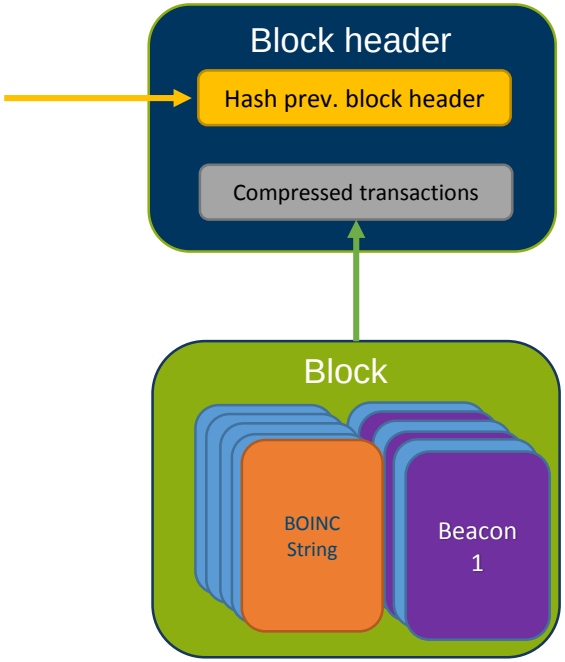
Fixing Gridcoin with the beacon system



Fixing Gridcoin with the beacon system



Fixing Gridcoin with the beacon system

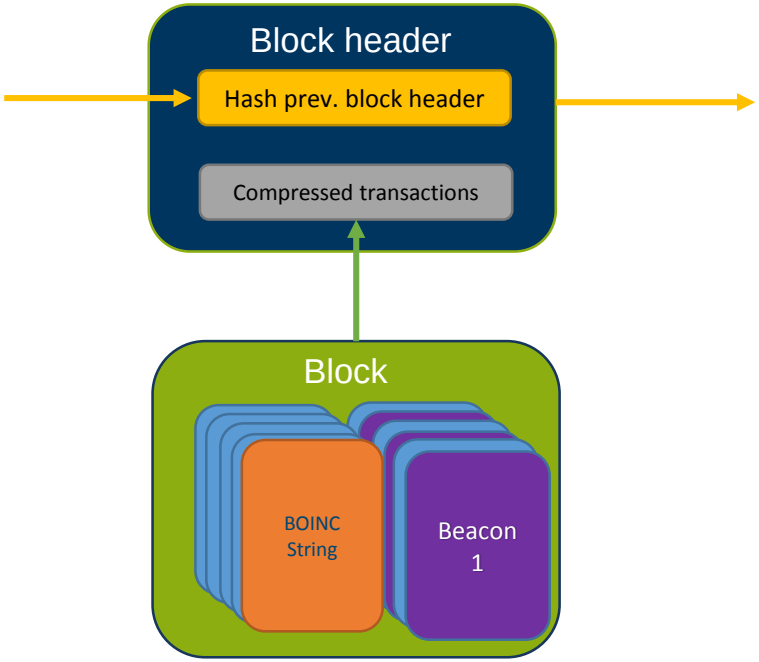


Check/Add Beacon 1 to Database
- SK_1 and PK_1 to $H(CPID_X)$

Check/Add Beacon 2 to Database
- SK_2 and PK_2 to $H(CPID_Y)$

Check/Add Beacon 3 to Database
- SK_3 and PK_3 to $H(CPID_Z)$

Fixing Gridcoin with the beacon system

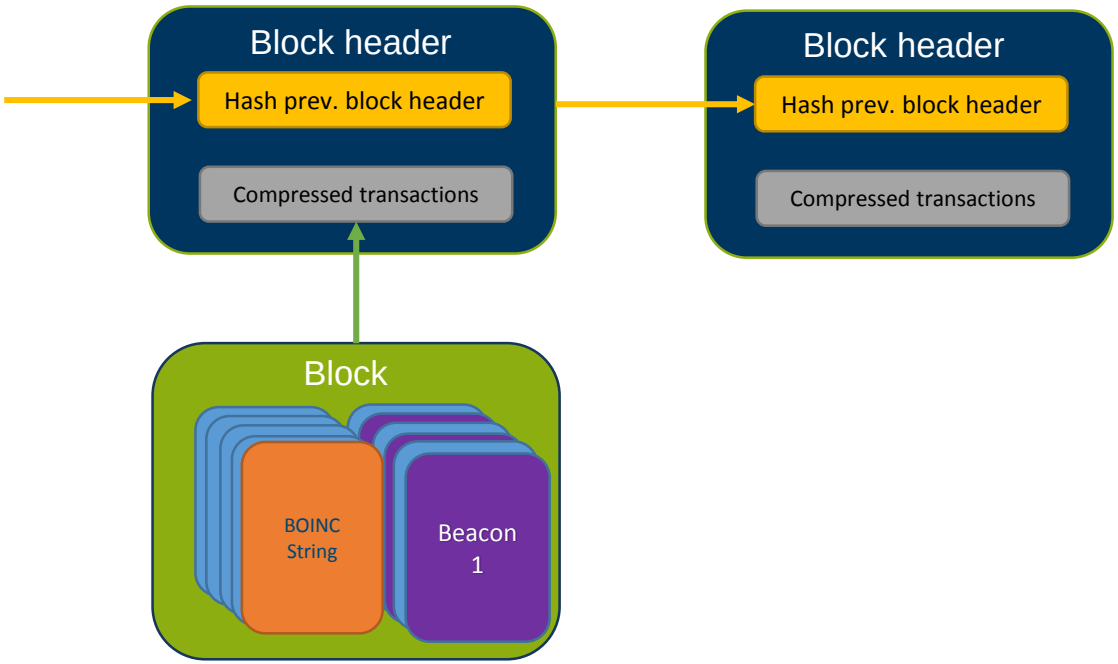


Check/Add Beacon 1 to Database
- SK_1 and PK_1 to $H(CPID_X)$

Check/Add Beacon 2 to Database
- SK_2 and PK_2 to $H(CPID_Y)$

Check/Add Beacon 3 to Database
- SK_3 and PK_3 to $H(CPID_Z)$

Fixing Gridcoin with the beacon system

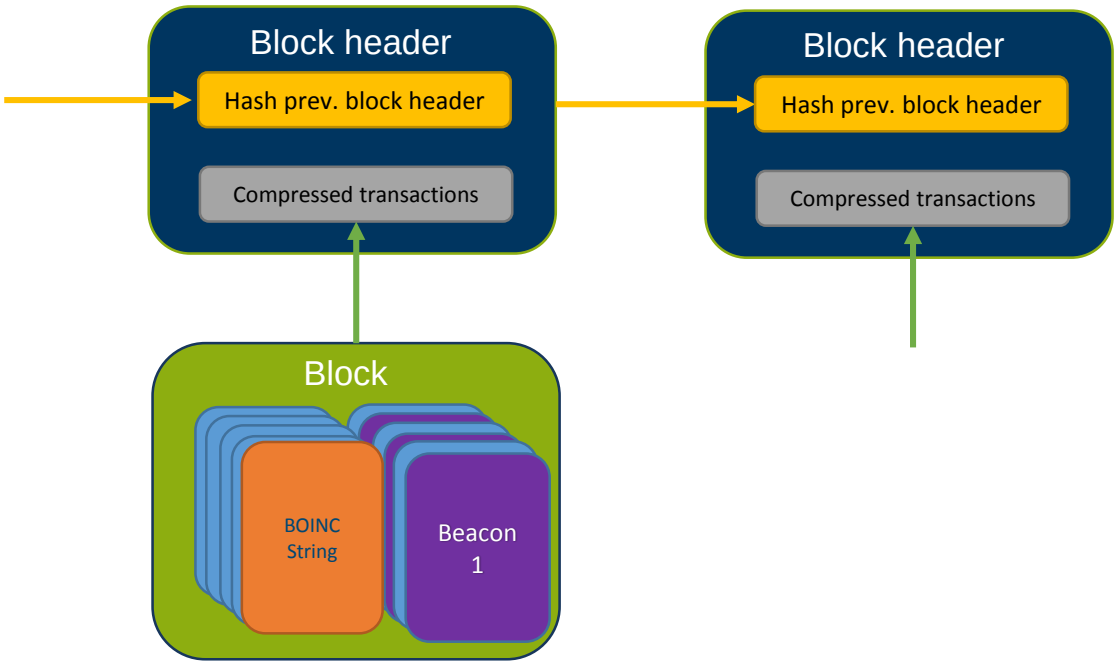


Check/Add Beacon 1 to Database
- SK_1 and PK_1 to $H(CPID_X)$

Check/Add Beacon 2 to Database
- SK_2 and PK_2 to $H(CPID_Y)$

Check/Add Beacon 3 to Database
- SK_3 and PK_3 to $H(CPID_Z)$

Fixing Gridcoin with the beacon system

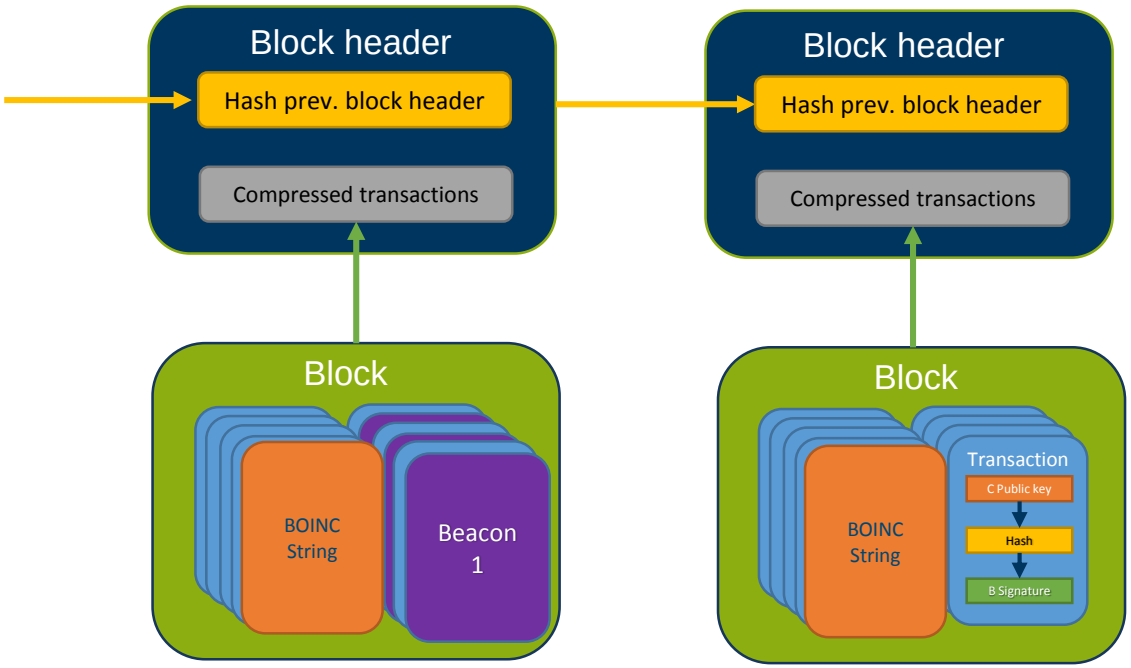


Check/Add Beacon 1 to Database
- SK_1 and PK_1 to $H(CPID_X)$

Check/Add Beacon 2 to Database
- SK_2 and PK_2 to $H(CPID_Y)$

Check/Add Beacon 3 to Database
- SK_3 and PK_3 to $H(CPID_Z)$

Fixing Gridcoin with the beacon system

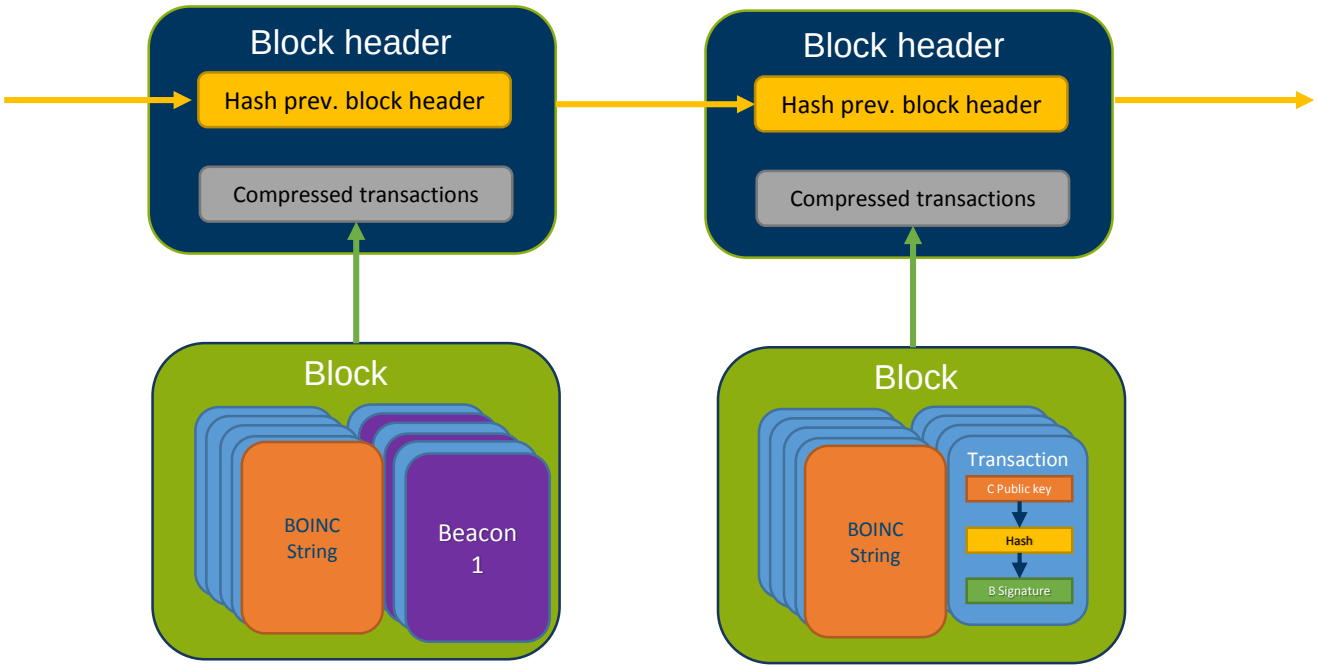


Check/Add Beacon 1 to Database
- SK_1 and PK_1 to $H(CPID_X)$

Check/Add Beacon 2 to Database
- SK_2 and PK_2 to $H(CPID_Y)$

Check/Add Beacon 3 to Database
- SK_3 and PK_3 to $H(CPID_Z)$

Fixing Gridcoin with the beacon system

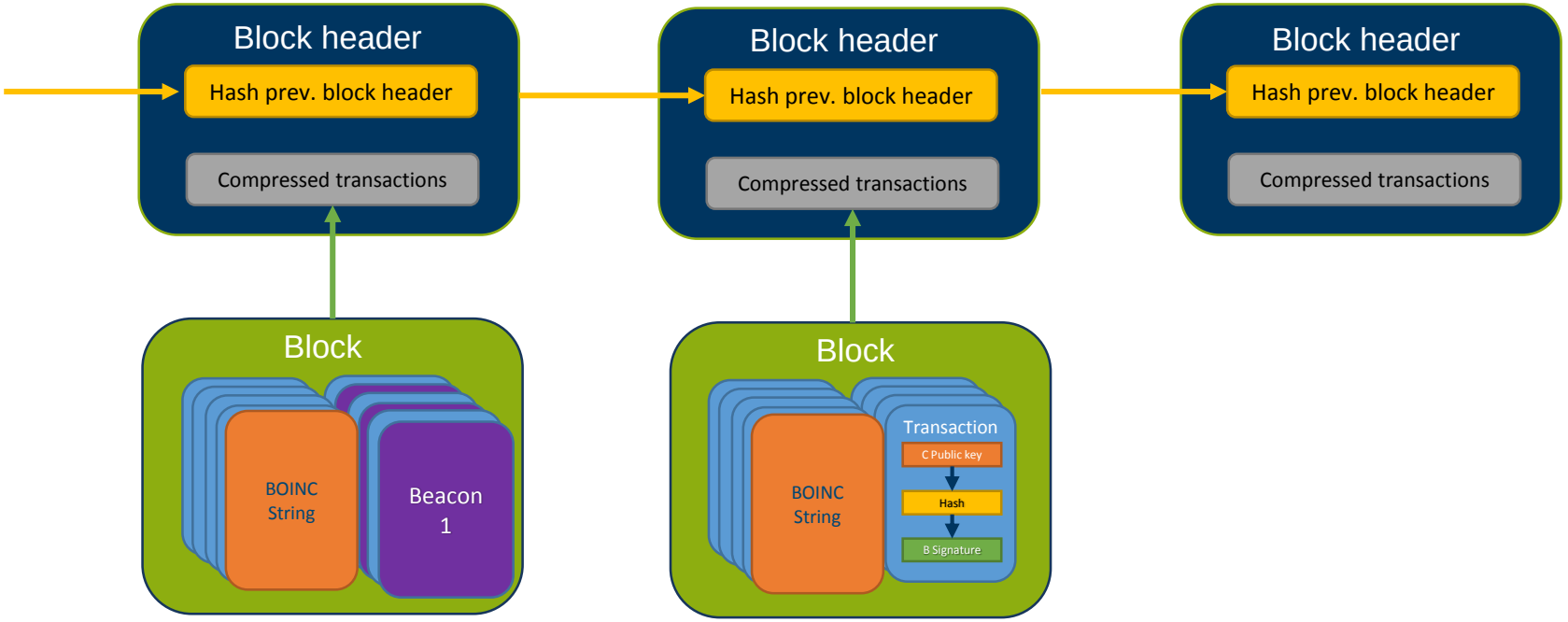


Check/Add Beacon 1 to Database
- SK_1 and PK_1 to $H(CPID_X)$

Check/Add Beacon 2 to Database
- SK_2 and PK_2 to $H(CPID_Y)$

Check/Add Beacon 3 to Database
- SK_3 and PK_3 to $H(CPID_Z)$

Fixing Gridcoin with the beacon system

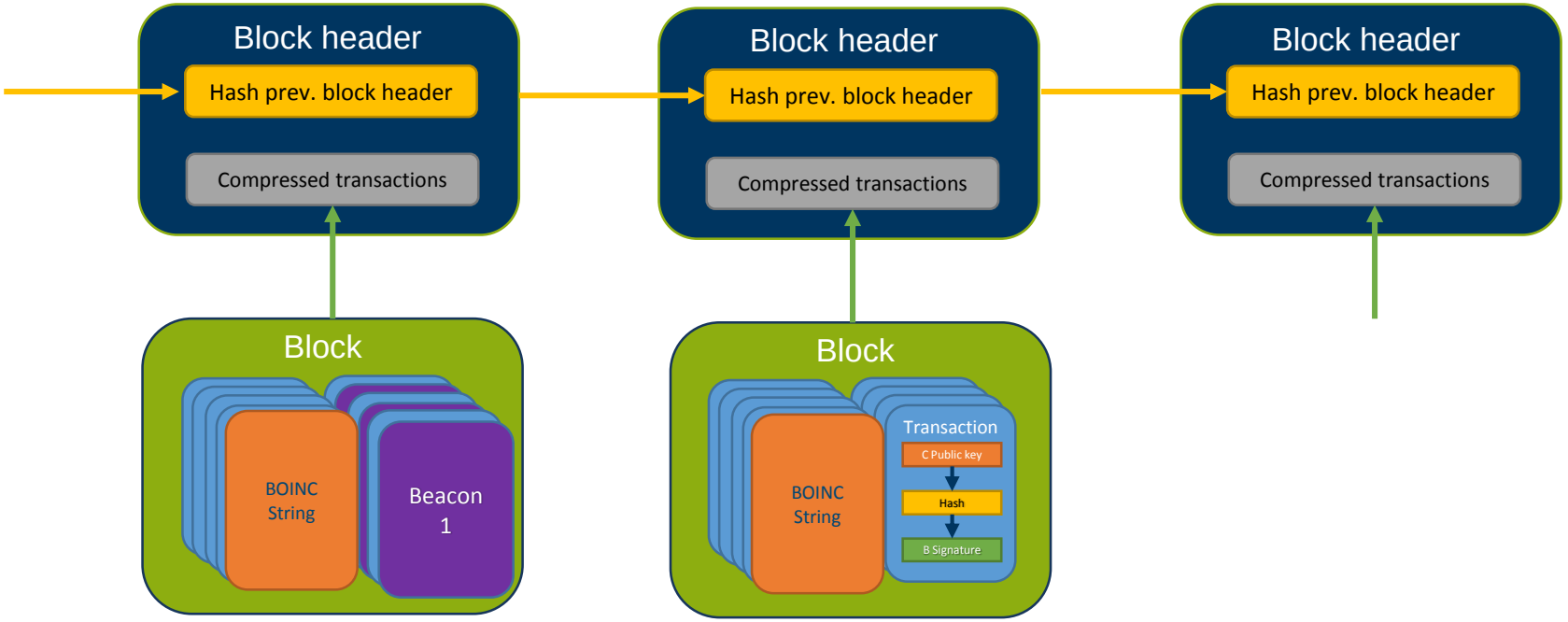


Check/Add Beacon 1 to Database
- SK_1 and PK_1 to $H(CPID_X)$

Check/Add Beacon 2 to Database
- SK_2 and PK_2 to $H(CPID_Y)$

Check/Add Beacon 3 to Database
- SK_3 and PK_3 to $H(CPID_Z)$

Fixing Gridcoin with the beacon system

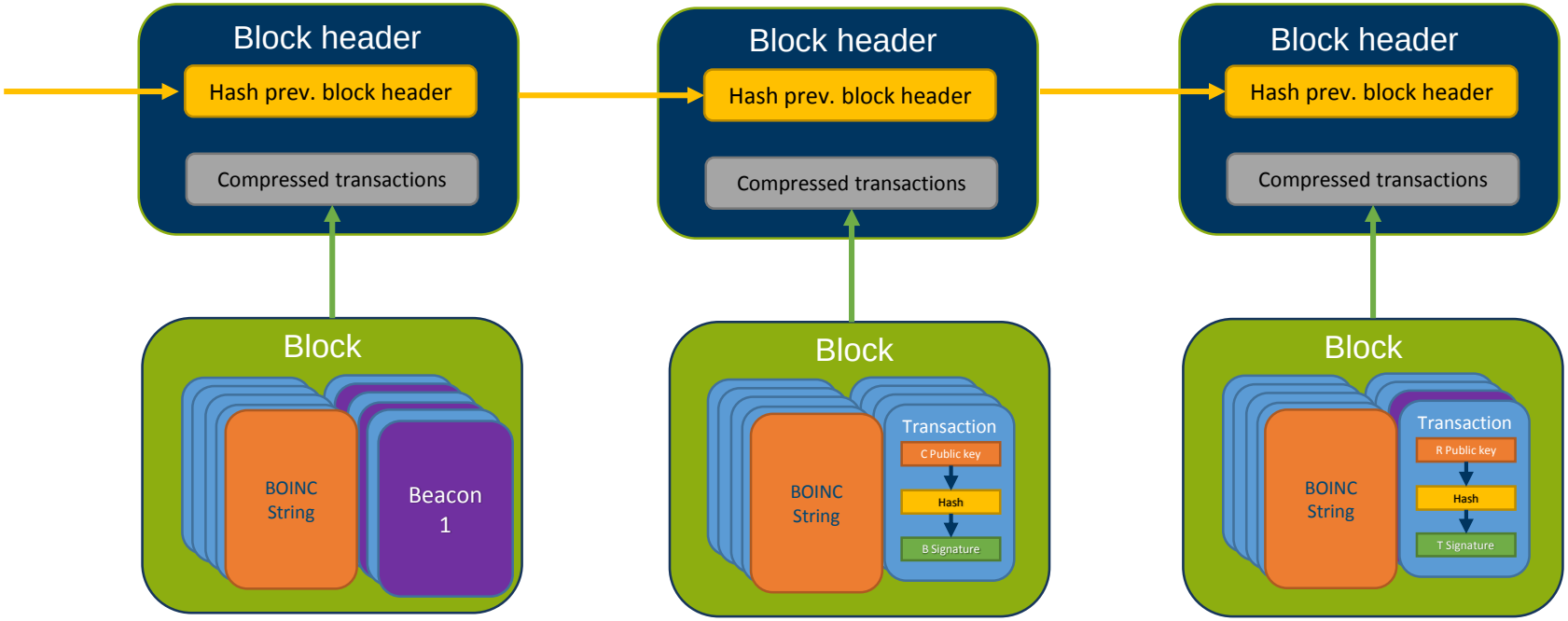


Check/Add Beacon 1 to Database
- SK_1 and PK_1 to $H(CPID_X)$

Check/Add Beacon 2 to Database
- SK_2 and PK_2 to $H(CPID_Y)$

Check/Add Beacon 3 to Database
- SK_3 and PK_3 to $H(CPID_Z)$

Fixing Gridcoin with the beacon system

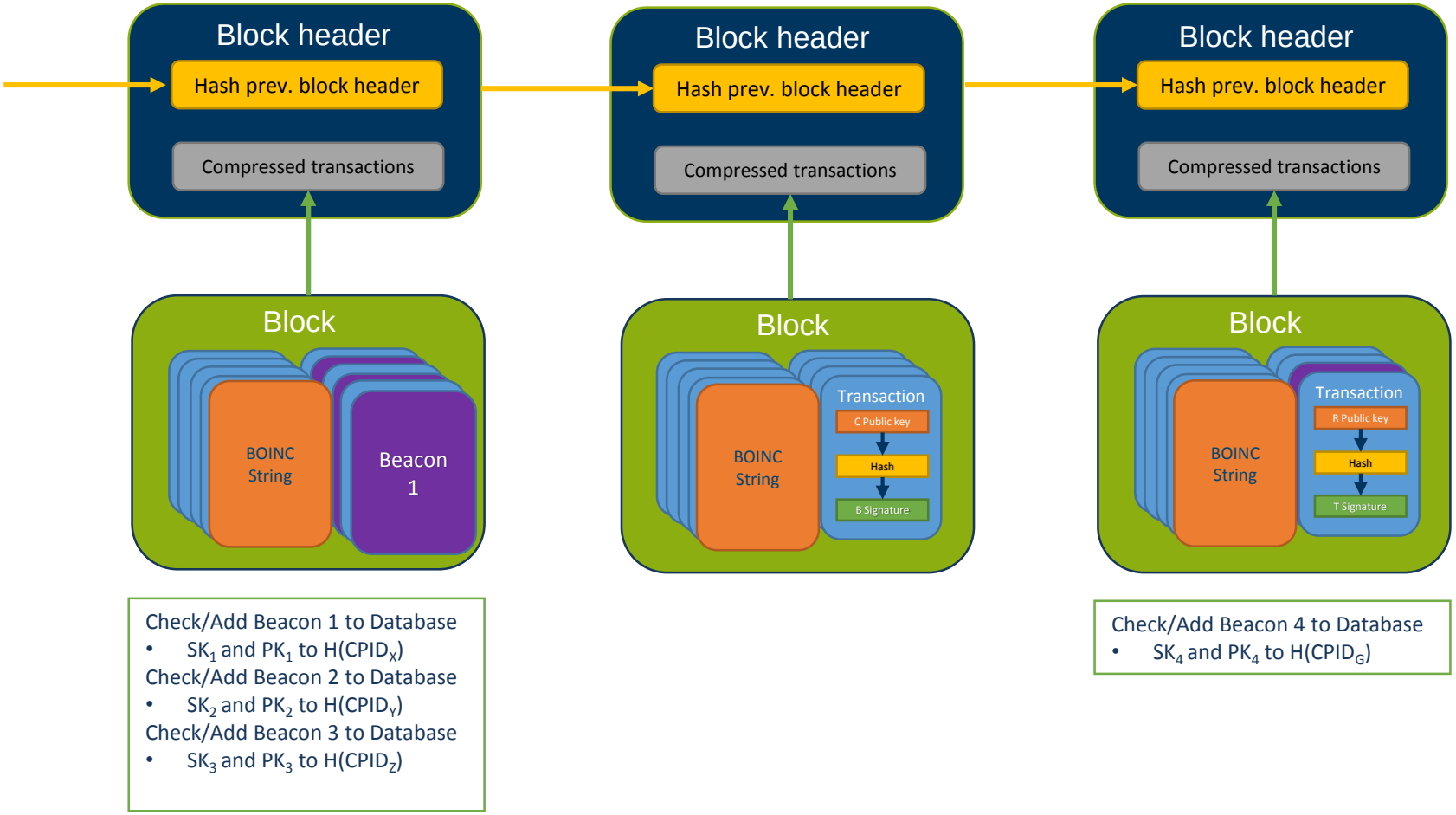


Check/Add Beacon 1 to Database
- SK_1 and PK_1 to $H(CPID_X)$

Check/Add Beacon 2 to Database
- SK_2 and PK_2 to $H(CPID_Y)$

Check/Add Beacon 3 to Database
- SK_3 and PK_3 to $H(CPID_Z)$

Fixing Gridcoin with the beacon system



Check/Add Beacon 1 to Database
- SK_1 and PK_1 to $H(CPID_X)$

Check/Add Beacon 2 to Database
- SK_2 and PK_2 to $H(CPID_Y)$

Check/Add Beacon 3 to Database
- SK_3 and PK_3 to $H(CPID_Z)$

Check/Add Beacon 4 to Database
- SK_4 and PK_4 to $H(CPID_G)$

Fixing Gridcoin with the beacon system

BOINC / Gridcoin infrastructure

Gridcoin block chain

Gridcoin users

Fixing Gridcoin with the beacon system

BOINC / Gridcoin infrastructure

Gridcoin block chain

Gridcoin users





Fixing Gridcoin with the beacon system

BOINC / Gridcoin infrastructure

BOINC project A

Statistical website

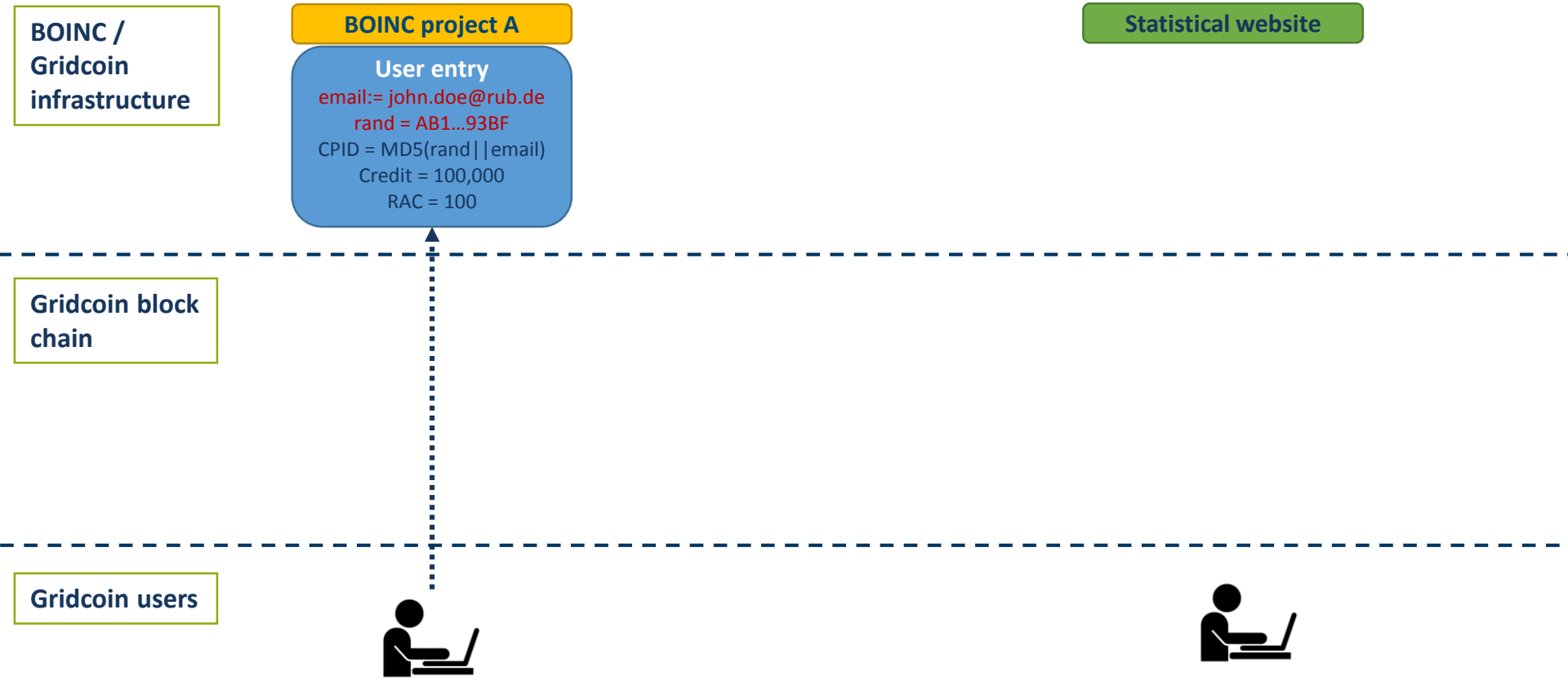
Gridcoin block chain

Gridcoin users

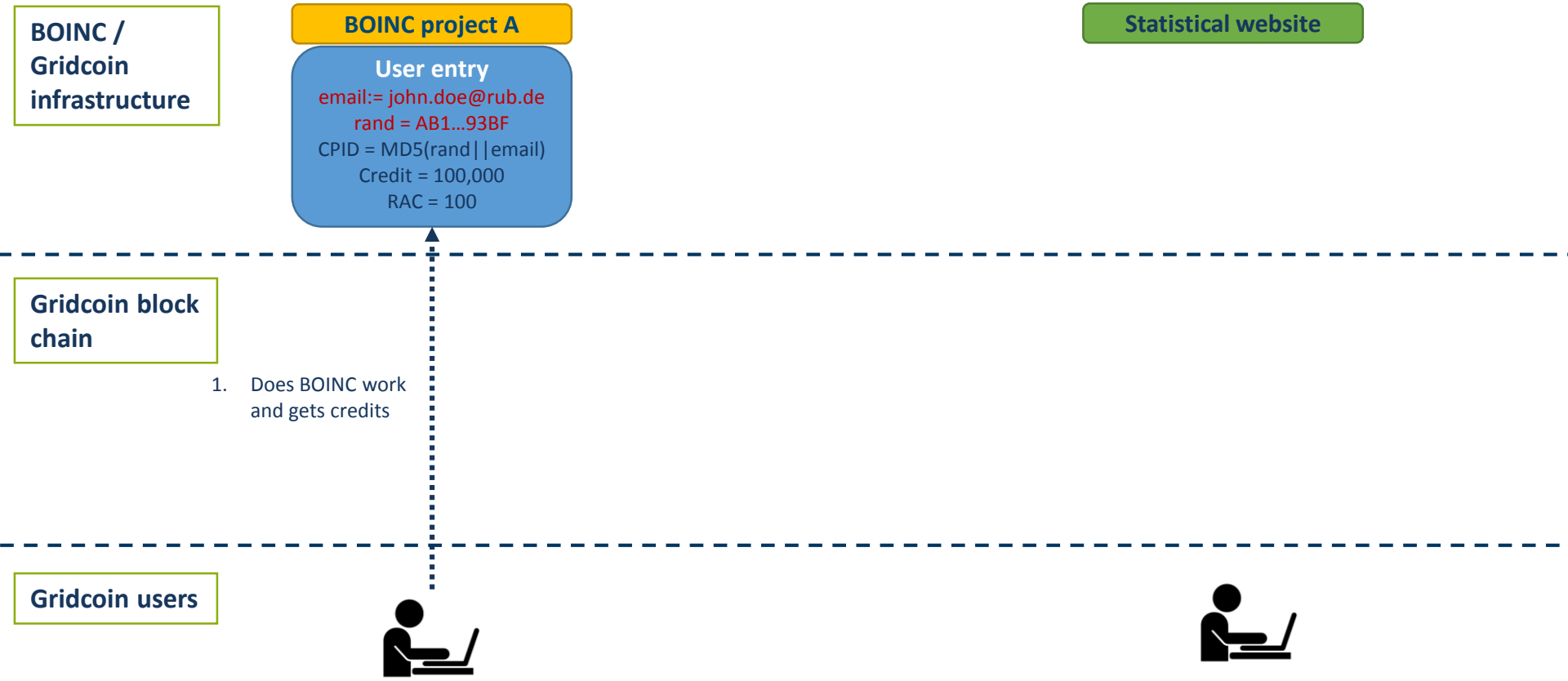




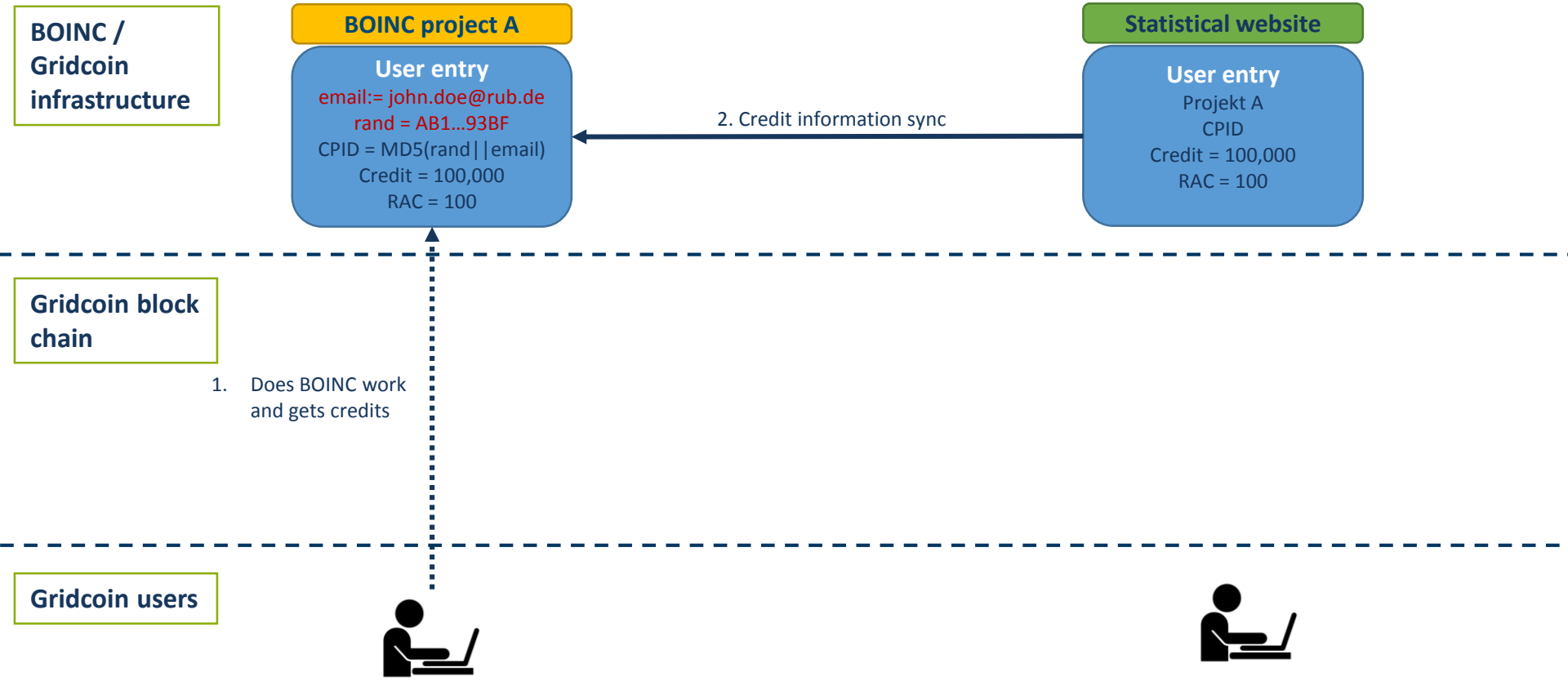
Fixing Gridcoin with the beacon system



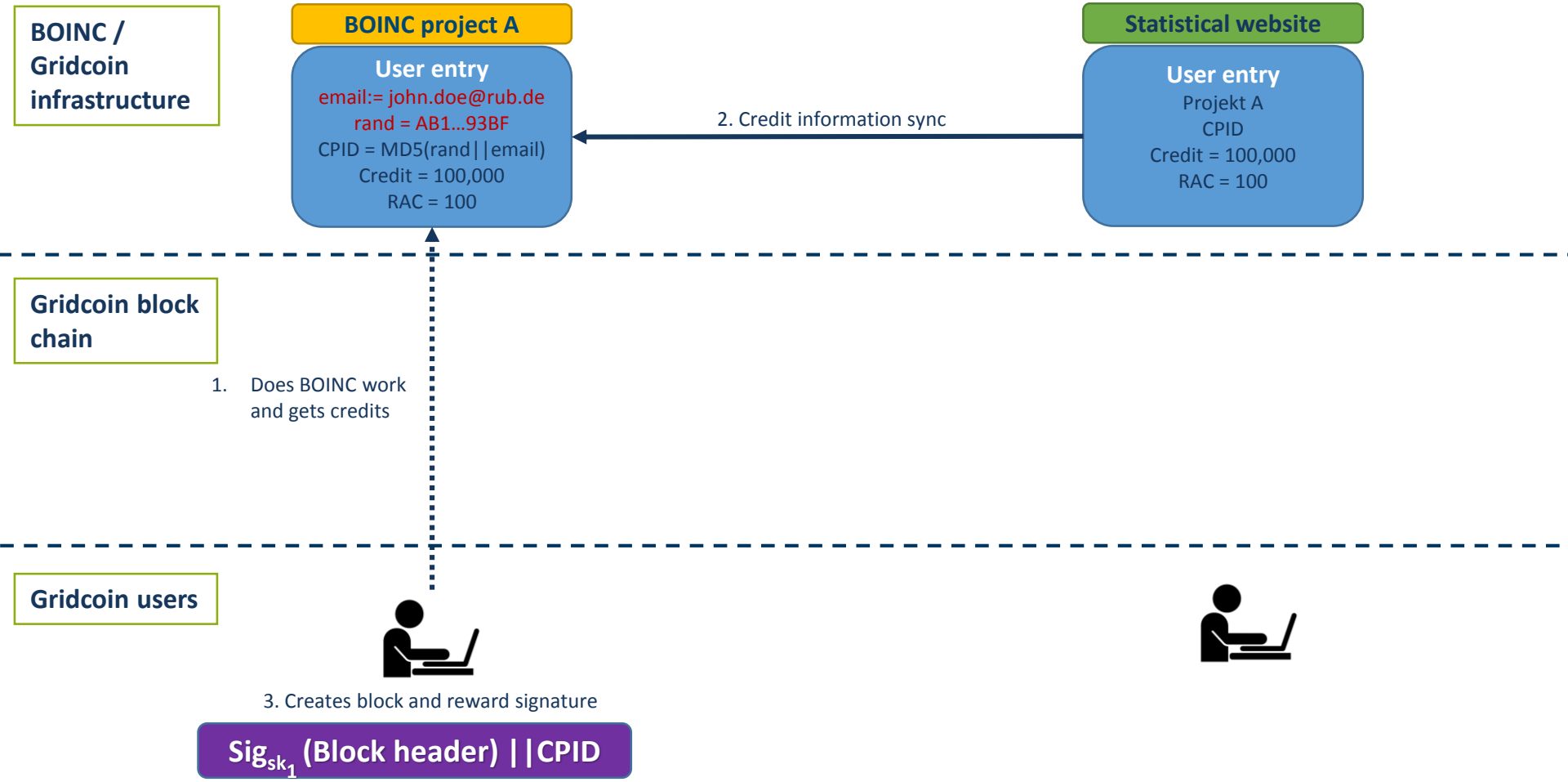
Fixing Gridcoin with the beacon system



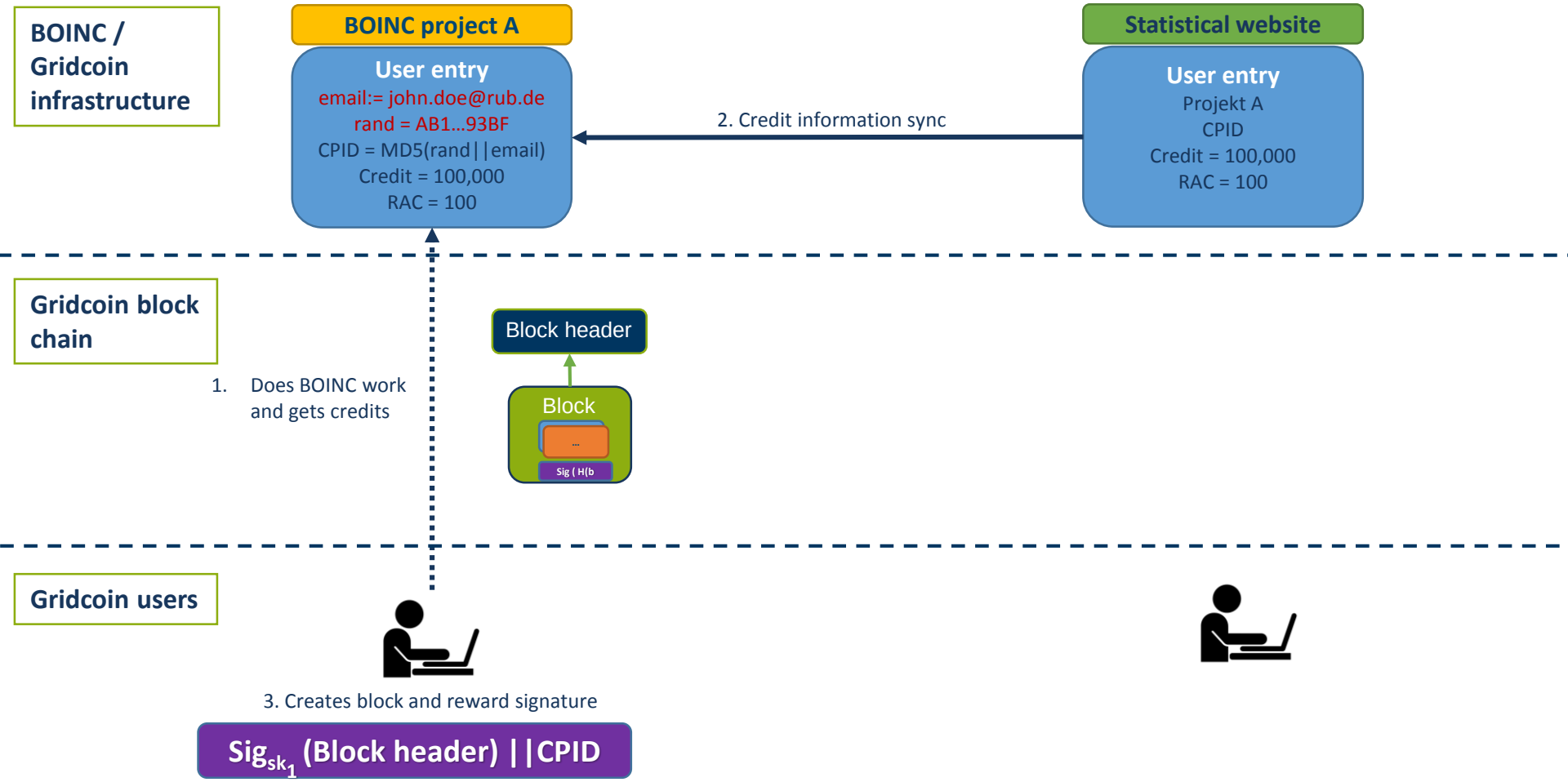
Fixing Gridcoin with the beacon system



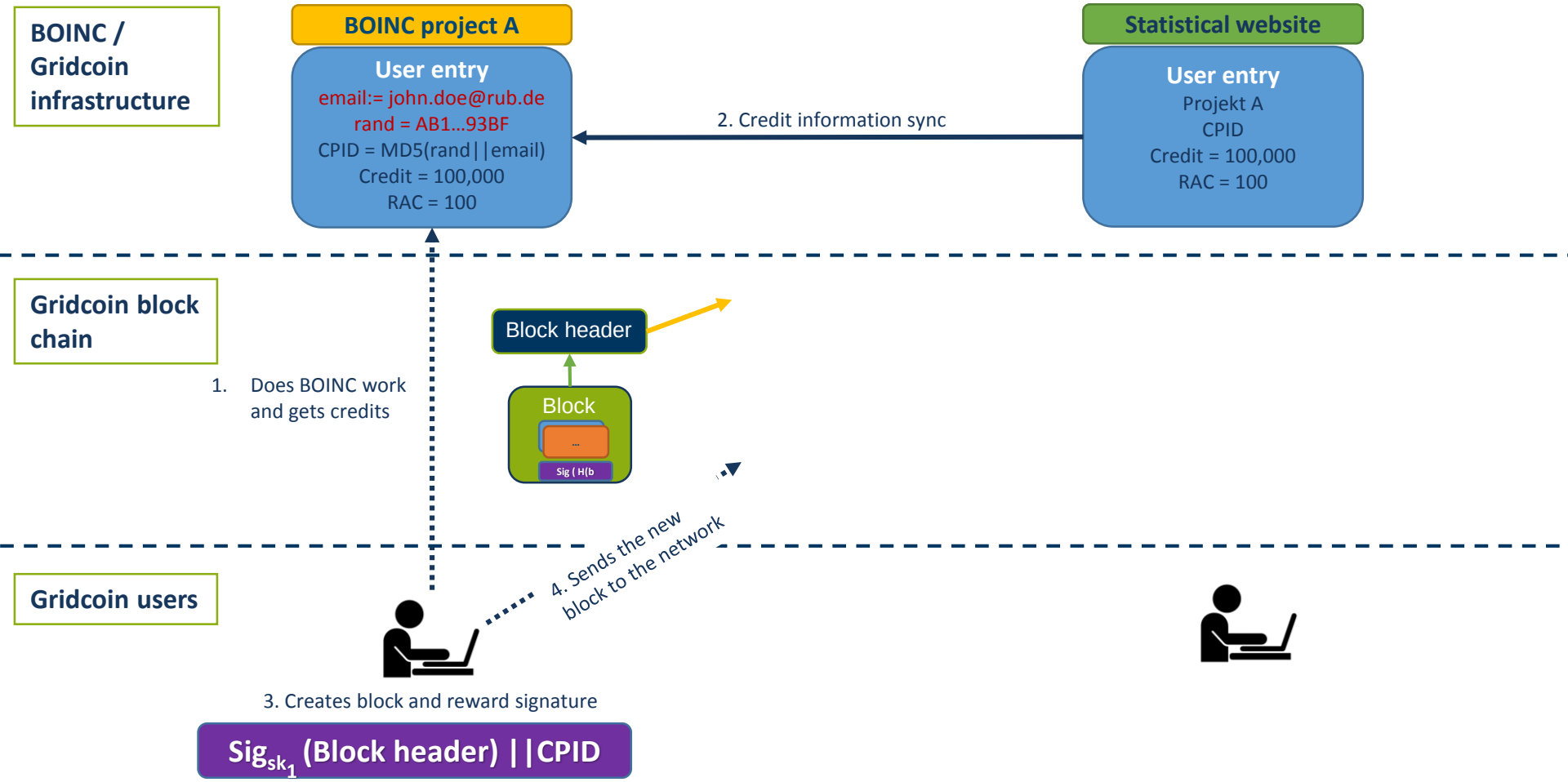
Fixing Gridcoin with the beacon system



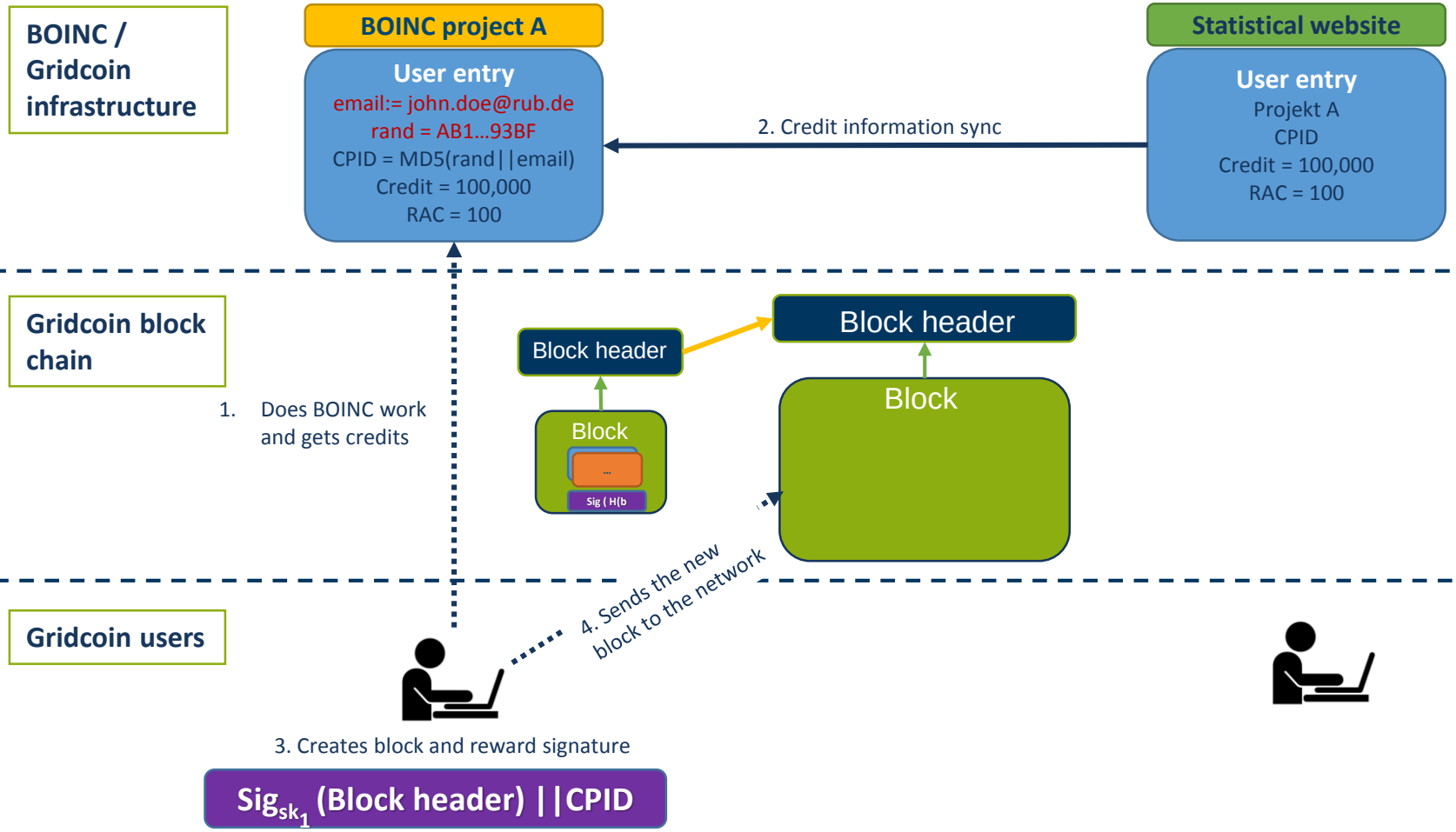
Fixing Gridcoin with the beacon system



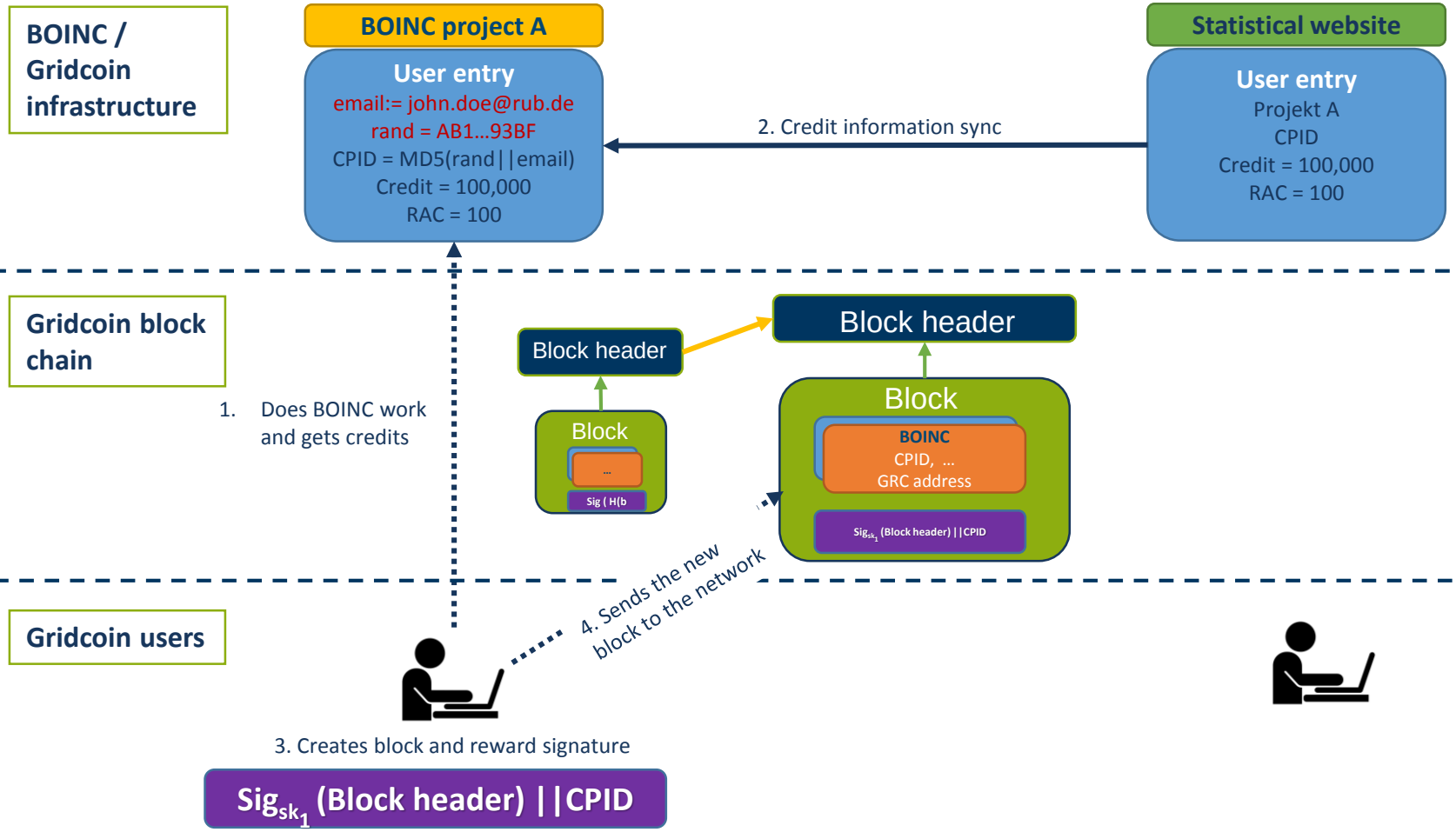
Fixing Gridcoin with the beacon system



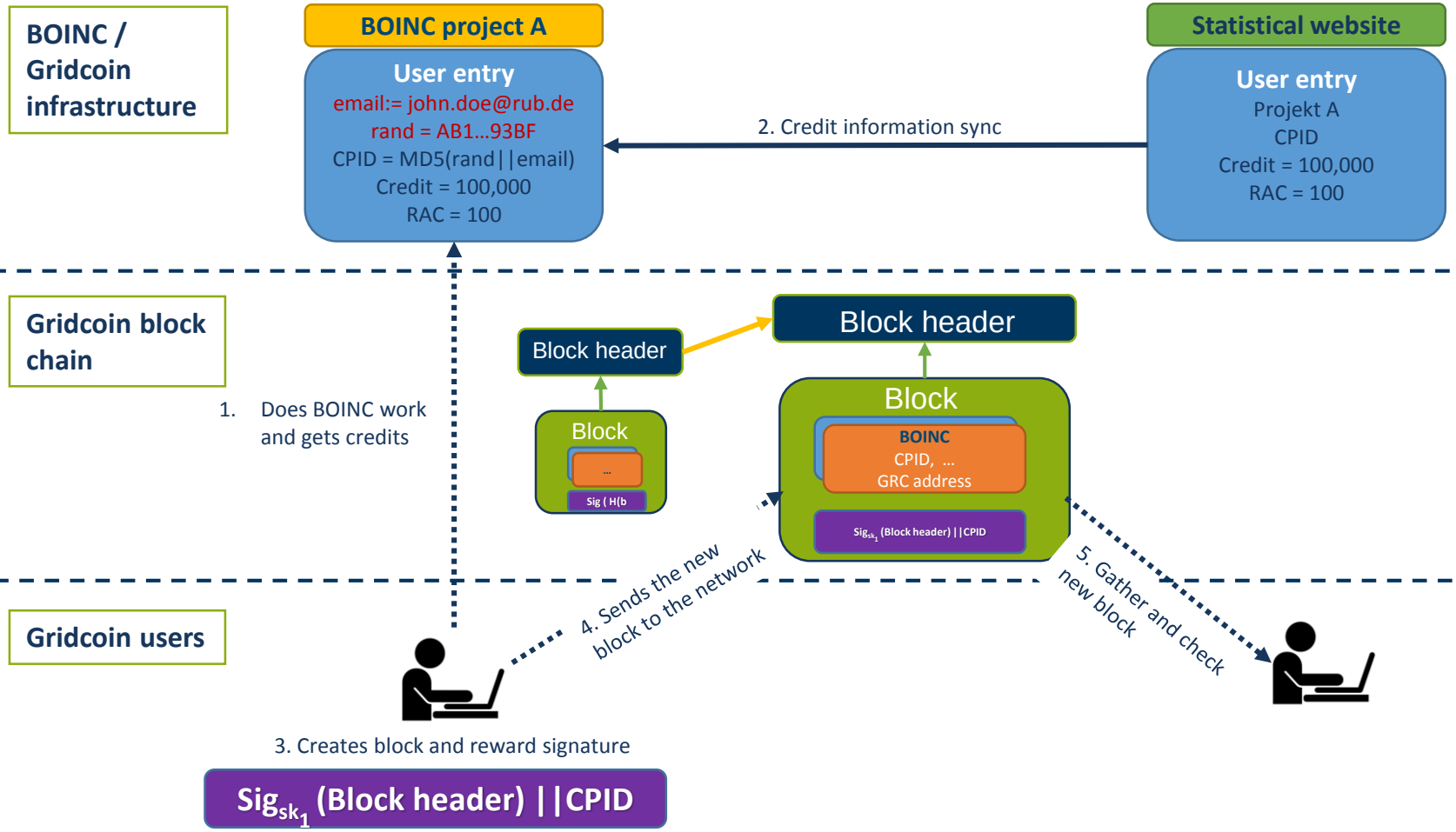
Fixing Gridcoin with the beacon system



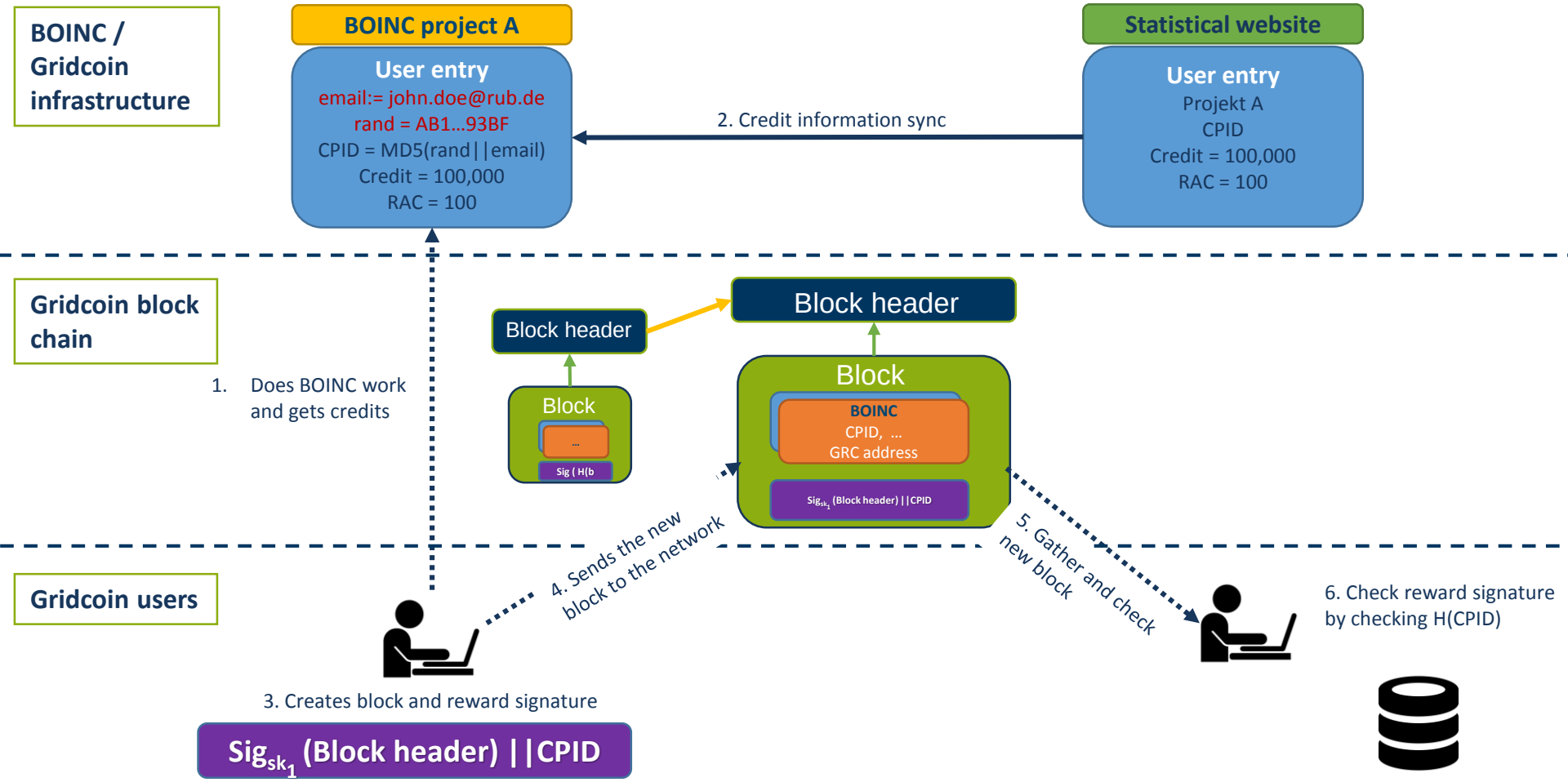
Fixing Gridcoin with the beacon system



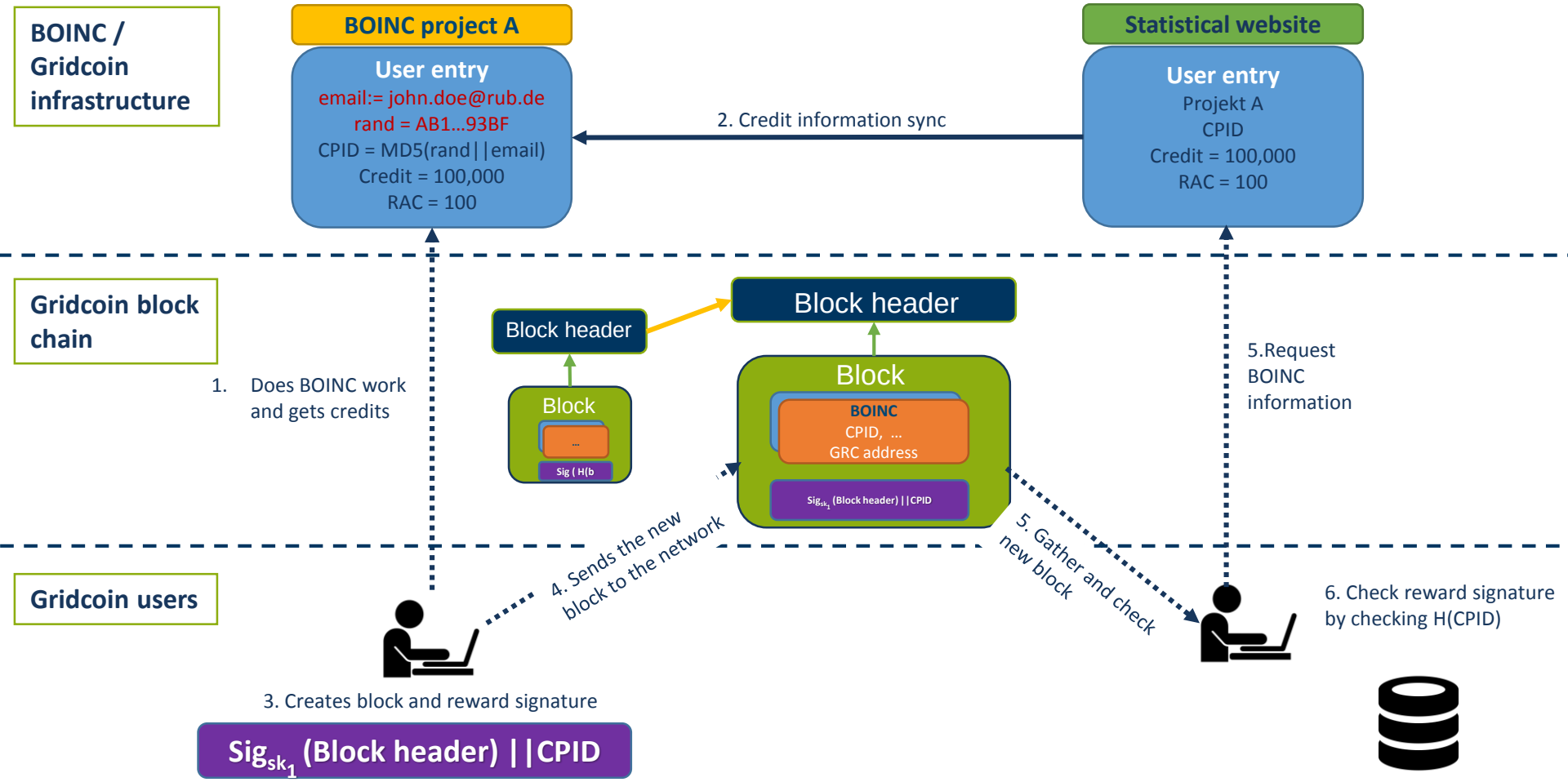
Fixing Gridcoin with the beacon system



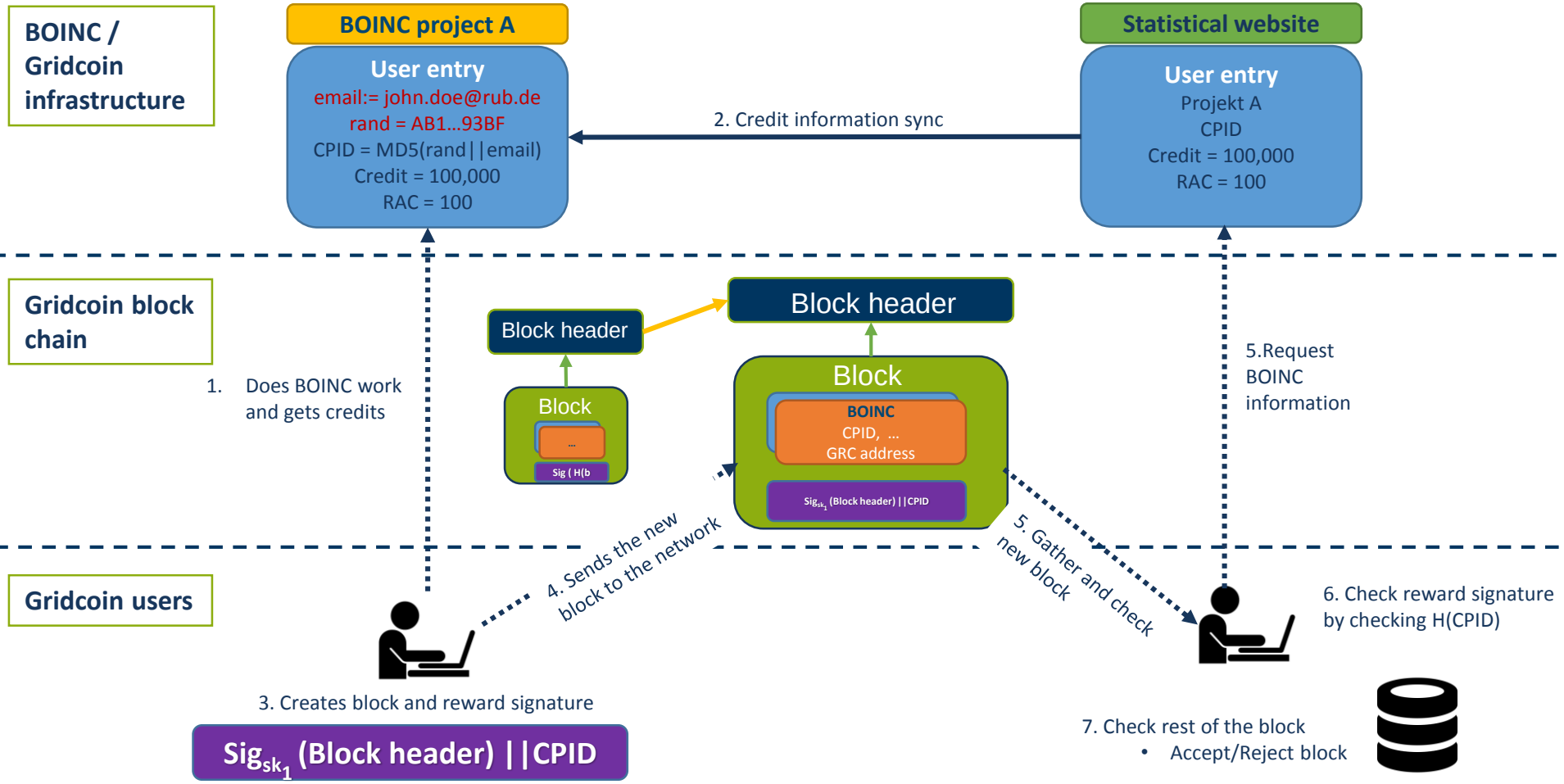
Fixing Gridcoin with the beacon system



Fixing Gridcoin with the beacon system



Fixing Gridcoin with the beacon system



Bitcoin

Bitcoin

History



Bitcoin

History

- 2008 paper published on cryptography mailinglist
- By Satoshi Nakamoto (?)
- Jan. 2009 source code released
- Soon used for trading



Bitcoin

History

- 2008 paper published on cryptography mailinglist
- By Satoshi Nakamoto (?)
- Jan. 2009 source code released
- Soon used for trading



laszlo
Full Member

Activity: 199

Trust: **0**: -0 / +0(0)
Ignore

Re: Pizza for bitcoins?
May 22, 2010, 07:17:26 PM

I just want to report that I successfully traded 10,000 bitcoins for pizza.

Pictures: http://heliacal.net/~solar/bitcoin/pizza/

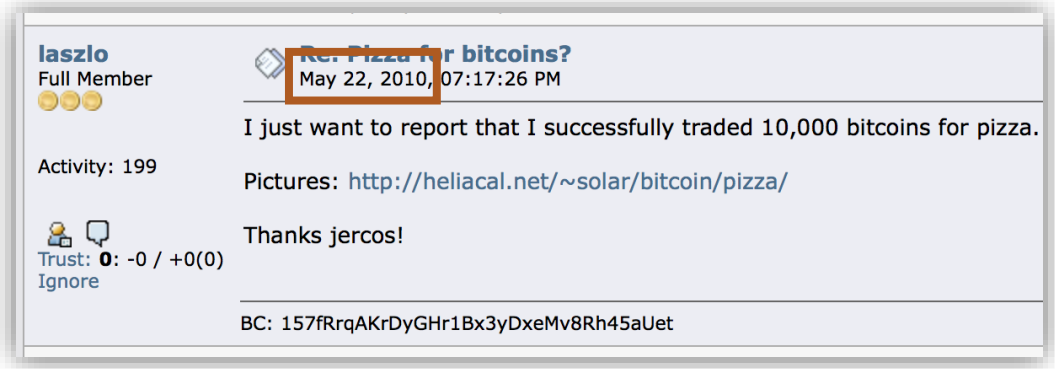
Thanks jercos!

BC: 157fRrqAKrDyGHr1Bx3yDxeMv8Rh45aUet

Bitcoin

History

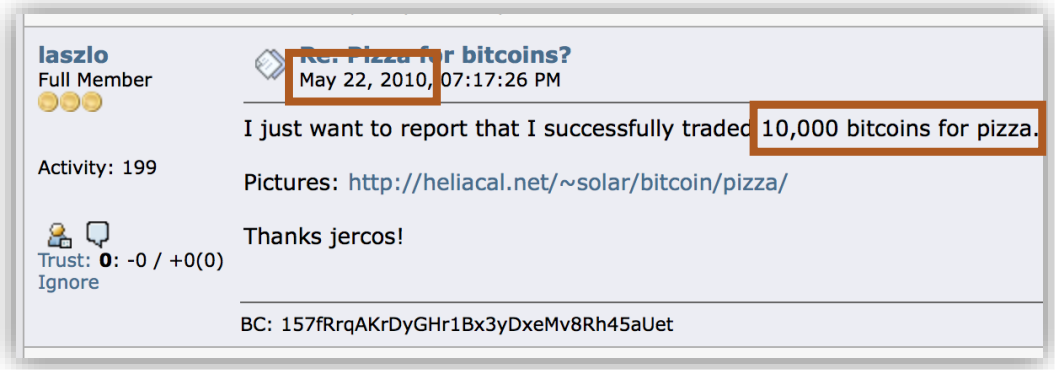
- 2008 paper published on cryptography mailinglist
- By Satoshi Nakamoto (?)
- Jan. 2009 source code released
- Soon used for trading



Bitcoin

History

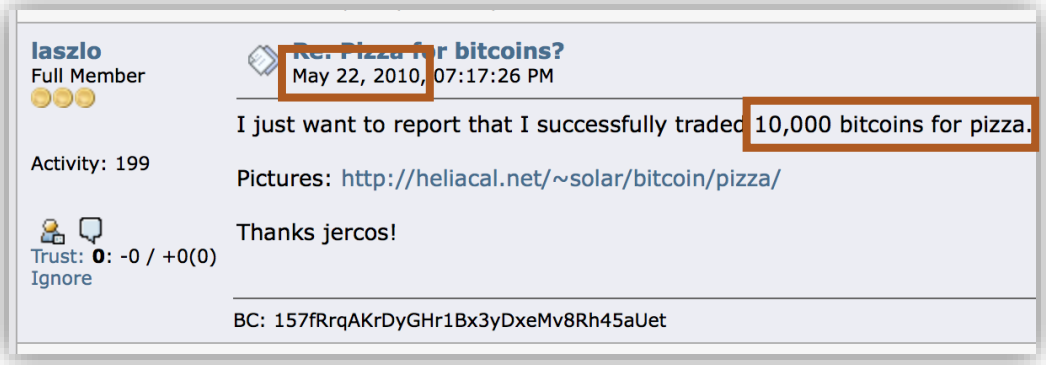
- 2008 paper published on cryptography mailinglist
- By Satoshi Nakamoto (?)
- Jan. 2009 source code released
- Soon used for trading



Bitcoin

History

- 2008 paper published on cryptography mailinglist
- By Satoshi Nakamoto (?)
- Jan. 2009 source code released
- Soon used for trading

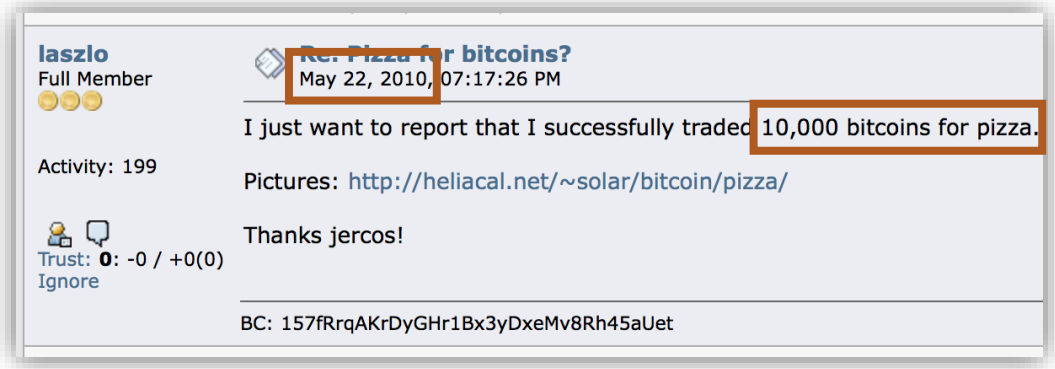




Bitcoin

History

- 2008 paper published on cryptography mailinglist
- By Satoshi Nakamoto (?)
- Jan. 2009 source code released
- Soon used for trading



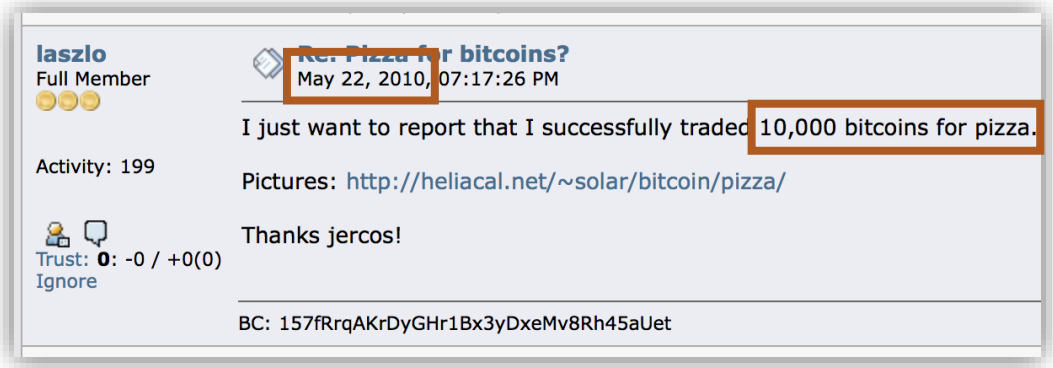
10,000 bitcoin = 41$ (22.10.2010)



Bitcoin

History

- 2008 paper published on cryptography mailinglist
- By Satoshi Nakamoto (?)
- Jan. 2009 source code released
- Soon used for trading





10,000 bitcoin = 41$ (22.10.2010)

10,000 bitcoin = 11,35 million € (27.04.2017)

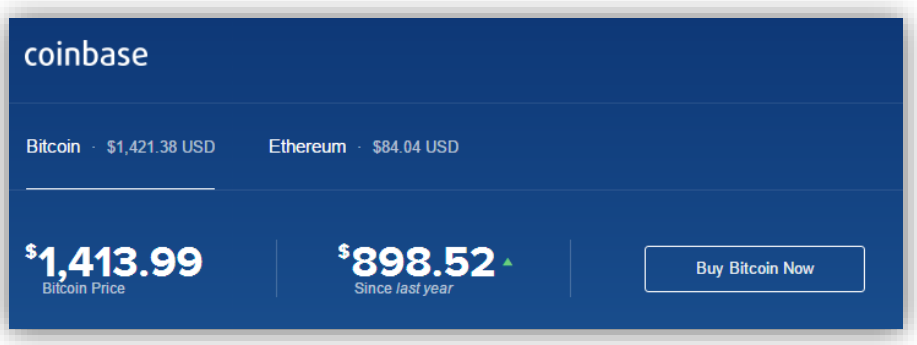
Bitcoin

How to get Bitcoins?

Bitcoin

How to get Bitcoins?

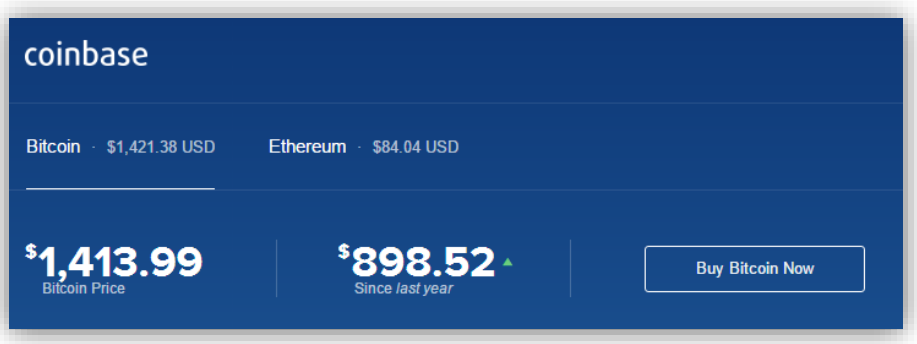
- Exchange places



Bitcoin

How to get Bitcoins?

- Exchange places
- Person to Person

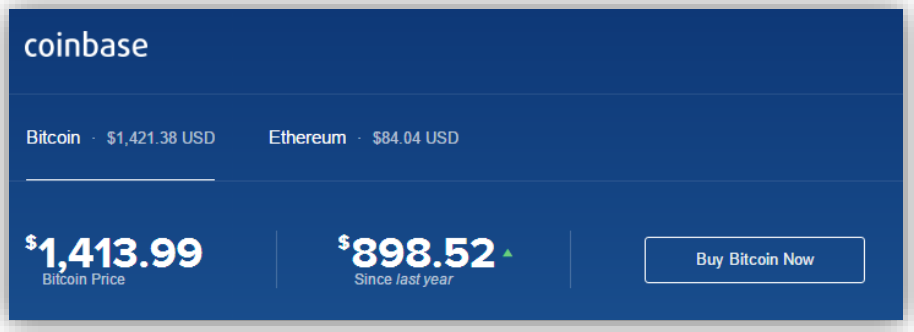




Bitcoin

How to get Bitcoins?

- Exchange places
- Person to Person
- Bitcoin ATMs







Bitcoin

Transactions

Source/Very good Howto: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

- Imagine:
  - Money is a string of bits
  - “Alice pay Bob 1 bitcoin”

Source/Very good Howto: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

- Imagine:
  - Money is a string of bits
  - “Alice pay Bob 1 bitcoin”

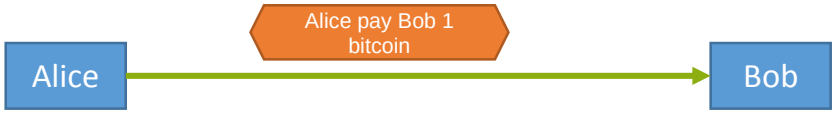


Source/Very good Howto: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

- Imagine:
  - Money is a string of bits
  - “Alice pay Bob 1 bitcoin”

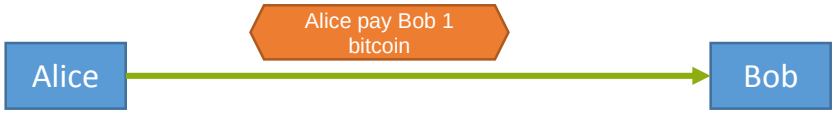


Source/Very good Howto: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

- Imagine:
  - Money is a string of bits
  - “Alice pay Bob 1 bitcoin”
- Problem: Eavesdrop (steal)



Source/Very good Howto: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

- Imagine:
  - Money is a string of bits
  - “Alice pay Bob 1 bitcoin”
- Problem: Eavesdrop (steal)
  - Solution: Signature

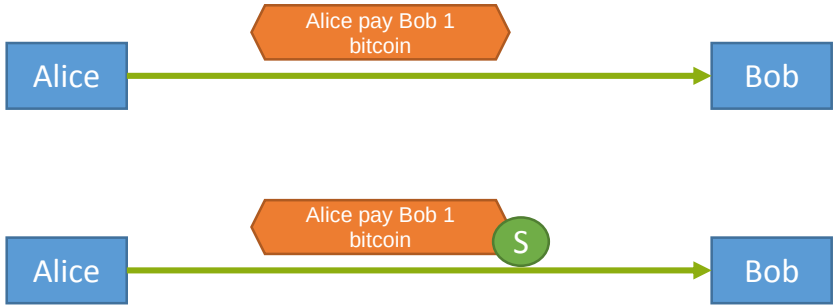


Source/Very good Howto: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

- Imagine:
  - Money is a string of bits
  - “Alice pay Bob 1 bitcoin”
- Problem: Eavesdrop (steal)
  - Solution: Signature

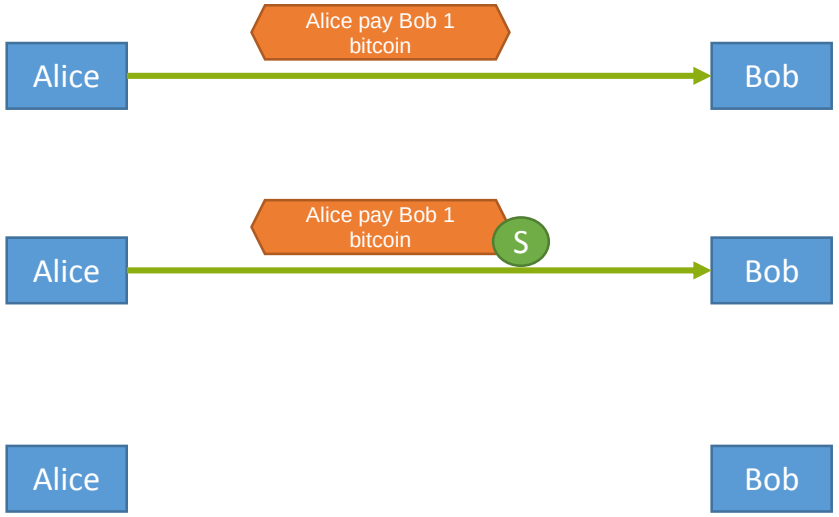


Source/Very good Howto: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

- Imagine:
  - Money is a string of bits
  - “Alice pay Bob 1 bitcoin”
- Problem: Eavesdrop (steal)
  - Solution: Signature
- Problem: Repeating

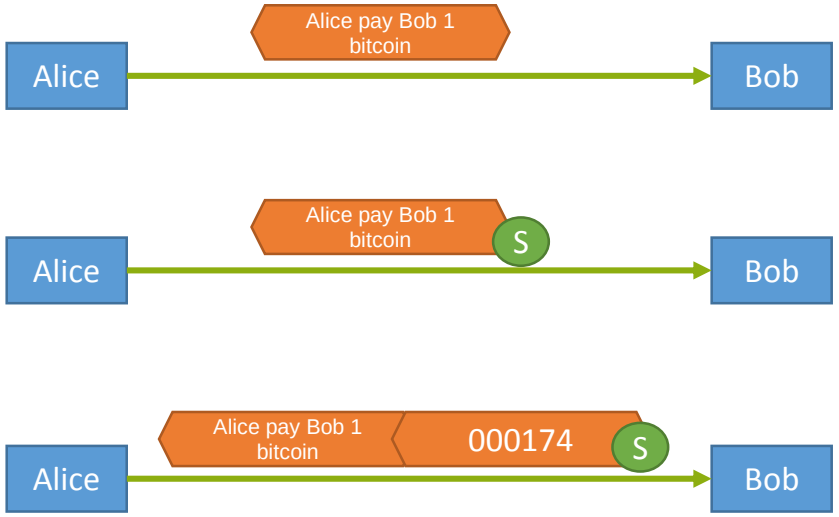


Source/Very good Howto: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

- Imagine:
  - Money is a string of bits
  - “Alice pay Bob 1 bitcoin”
- Problem: Eavesdrop (steal)
  - Solution: Signature
- Problem: Repeating
  - Solution: Serial number

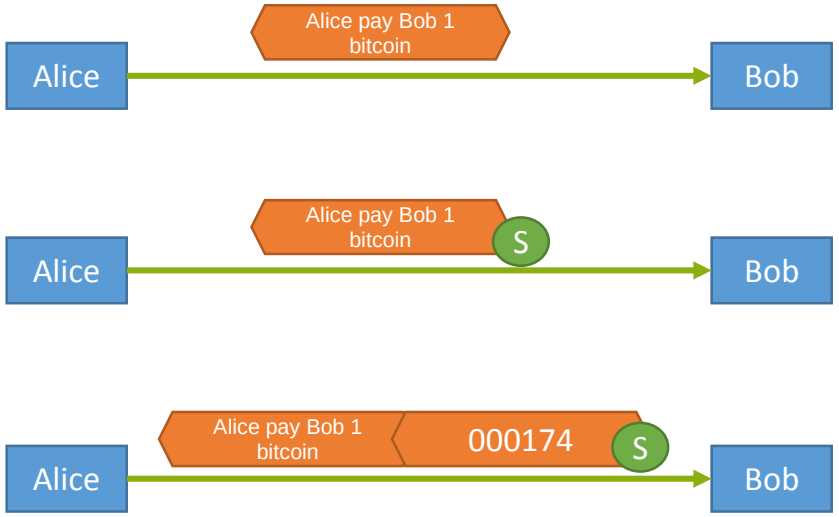


Source/Very good Howto: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

- Imagine:
  - Money is a string of bits
  - “Alice pay Bob 1 bitcoin”
- Problem: Eavesdrop (steal)
  - Solution: Signature
- Problem: Repeating
  - Solution: Serial number
- Problem:
  - Uniqueness of serial number

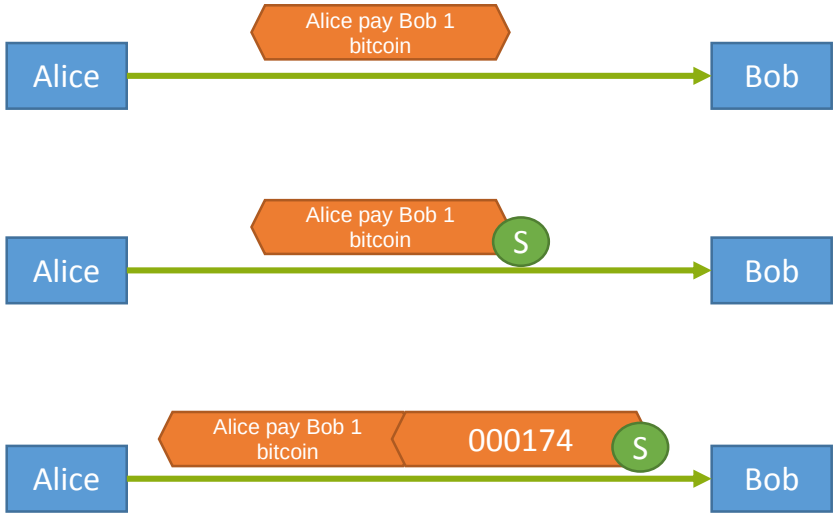


Source/Very good Howto: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

- Imagine:
  - Money is a string of bits
  - “Alice pay Bob 1 bitcoin”
- Problem: Eavesdrop (steal)
  - Solution: Signature
- Problem: Repeating
  - Solution: Serial number
- Problem:
  - Uniqueness of serial number
  - Solution: Public ledger



Source/Very good Howto: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

Source: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

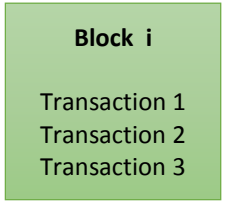
- Every user keeps track of all transactions
- Collect them in blocks
- All blocks are chained together

Source: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

- Every user keeps track of all transactions
- Collect them in blocks
- All blocks are chained together

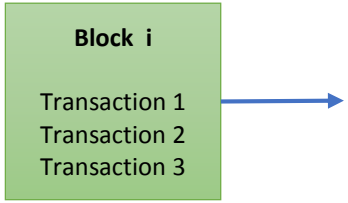


Source: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

- Every user keeps track of all transactions
- Collect them in blocks
- All blocks are chained together

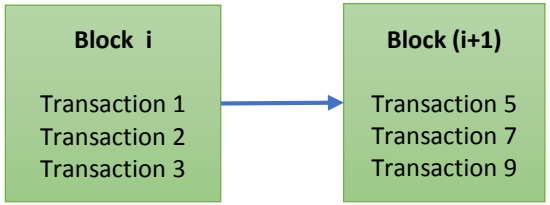


Source: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

- Every user keeps track of all transactions
- Collect them in blocks
- All blocks are chained together

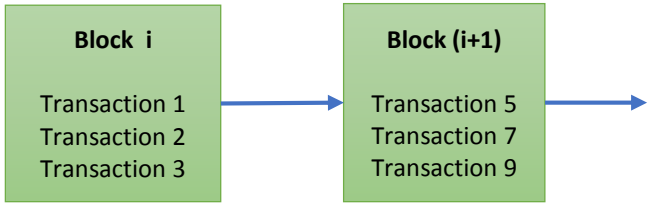


Source: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

- Every user keeps track of all transactions
- Collect them in blocks
- All blocks are chained together

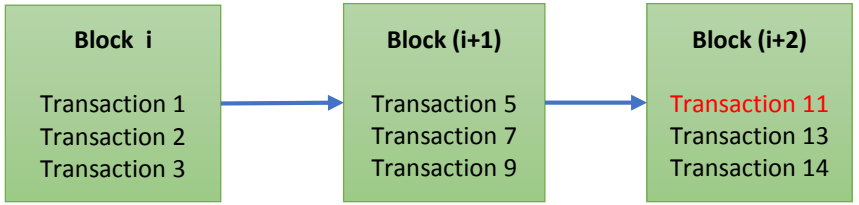


Source: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

- Every user keeps track of all transactions
- Collect them in blocks
- All blocks are chained together

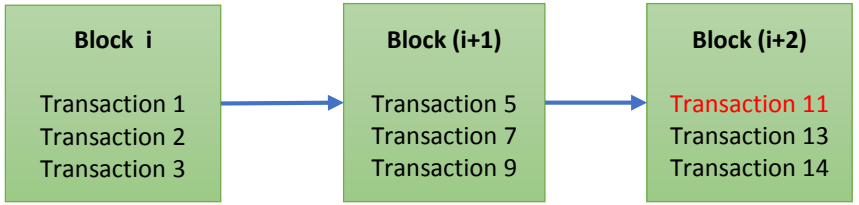


Source: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Transactions

- Every user keeps track of all transactions
- Collect them in blocks
- All blocks are chained together



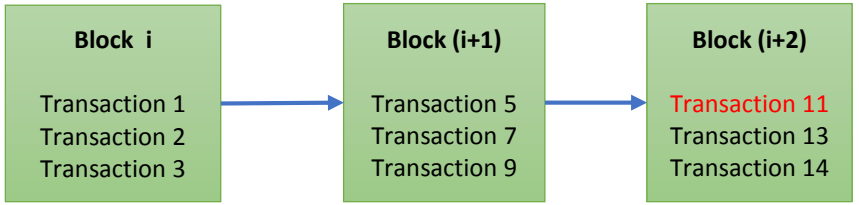
- Problem: Double spending

Source: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

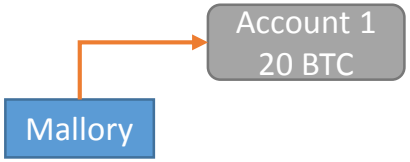
Bitcoin

Transactions

- Every user keeps track of all transactions
- Collect them in blocks
- All blocks are chained together



- Problem: Double spending

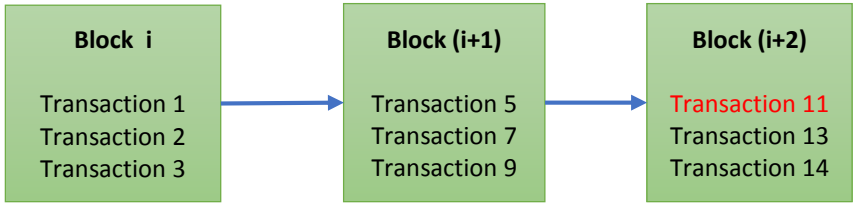


Source: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

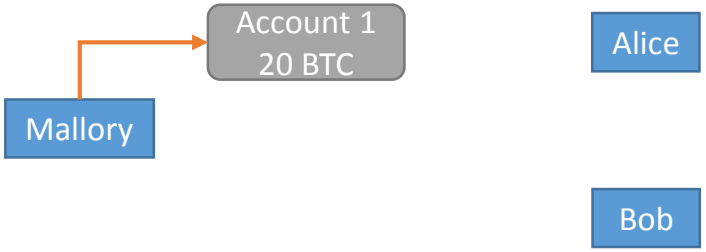
Bitcoin

Transactions

- Every user keeps track of all transactions
- Collect them in blocks
- All blocks are chained together



- Problem: Double spending

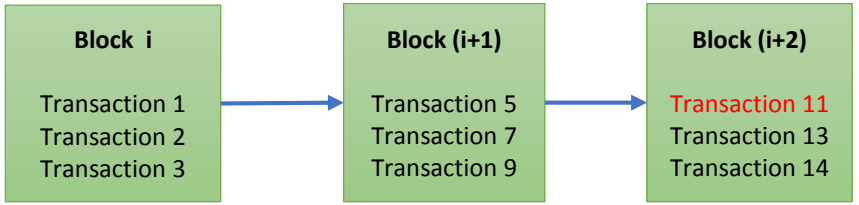


Source: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

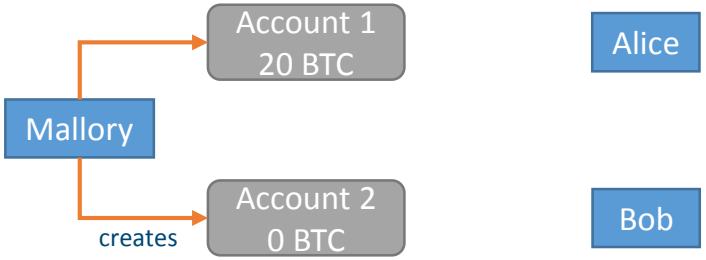
Bitcoin

Transactions

- Every user keeps track of all transactions
- Collect them in blocks
- All blocks are chained together



- Problem: Double spending

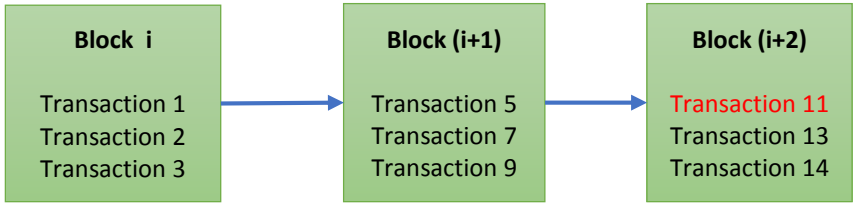


Source: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

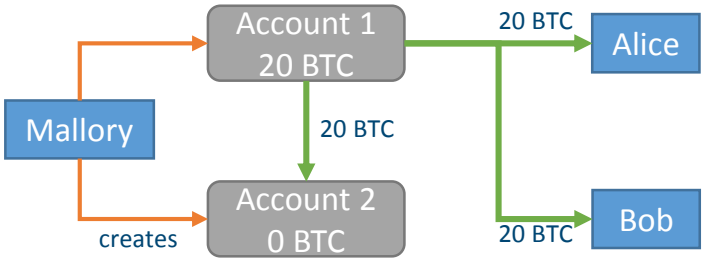
Bitcoin

Transactions

- Every user keeps track of all transactions
- Collect them in blocks
- All blocks are chained together



- Problem: Double spending

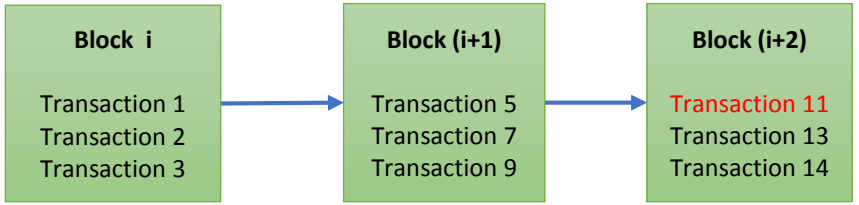


Source: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

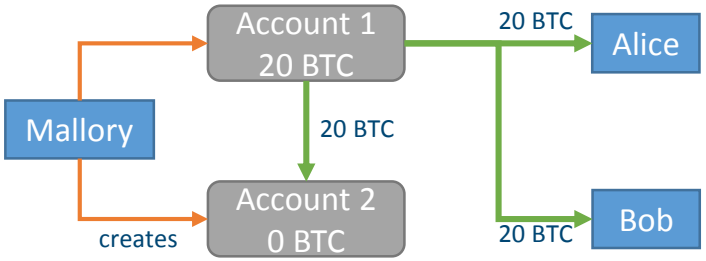
Bitcoin

Transactions

- Every user keeps track of all transactions
- Collect them in blocks
- All blocks are chained together



- Problem: Double spending
- Solution: Proof-of-Work



Source: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Bitcoin

Proof-of-Work

Bitcoin

Proof-of-Work

- Solving a computational moderately hard problem
- Solution allows to append a block to chain

Bitcoin

Proof-of-Work

- Solving a computational moderately hard problem
- Solution allows to append a block to chain
- Hash(transactions||…||nonce) < 0x0000000||FE32…9AB

Bitcoin

Proof-of-Work

- Solving a computational moderately hard problem
- Solution allows to append a block to chain
- Hash(transactions||…||nonce) < 0x0000000||FE32…9AB
- Genesis block: 0000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
- Current block: 00000000000000000217084a9362676393abd073f4123a67b0a54b9cc86c0a03

Bitcoin

Proof-of-Work

- Solving a computational moderately hard problem
- Solution allows to append a block to chain
- Hash(transactions||…||nonce) < 0x0000000||FE32…9AB
- Genesis block: 0000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
- Current block: 00000000000000000217084a9362676393abd073f4123a67b0a54b9cc86c0a03
- Reward for new block
- Creates a race

Bitcoin

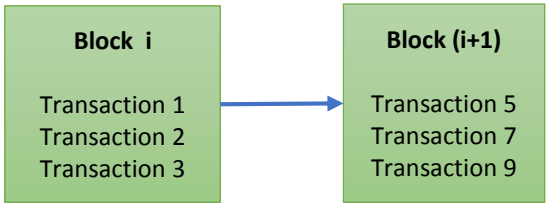
Block chain fork

- Block get's broadcasted into the bitcoin network
- What, if two blocks are created at the same time?

Bitcoin

Block chain fork

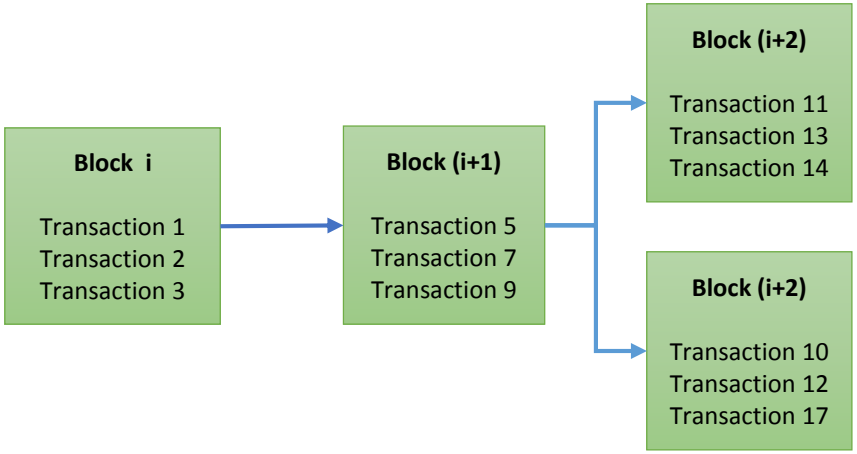
- Block get's broadcasted into the bitcoin network
- What, if two blocks are created at the same time?



Bitcoin

Block chain fork

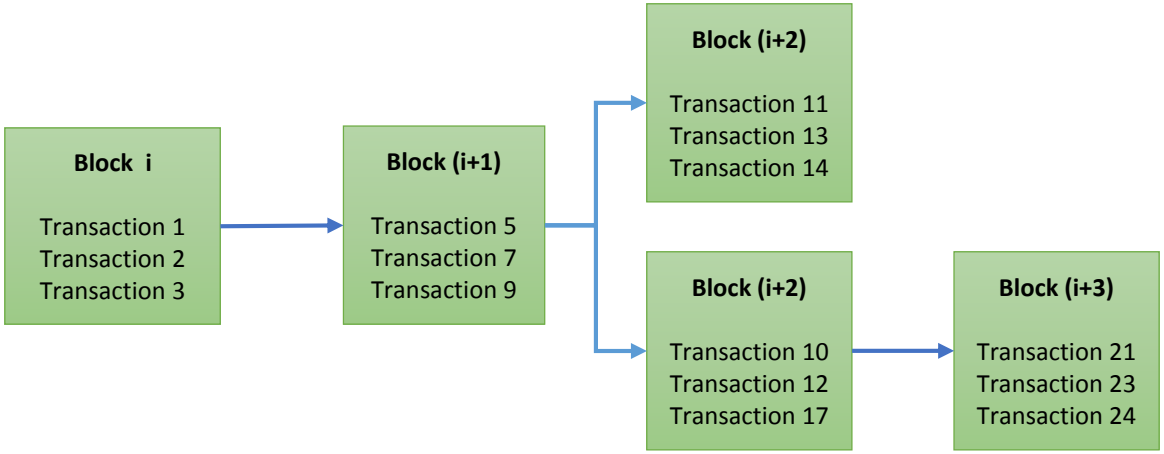
- Block get's broadcasted into the bitcoin network
- What, if two blocks are created at the same time?



Bitcoin

Block chain fork

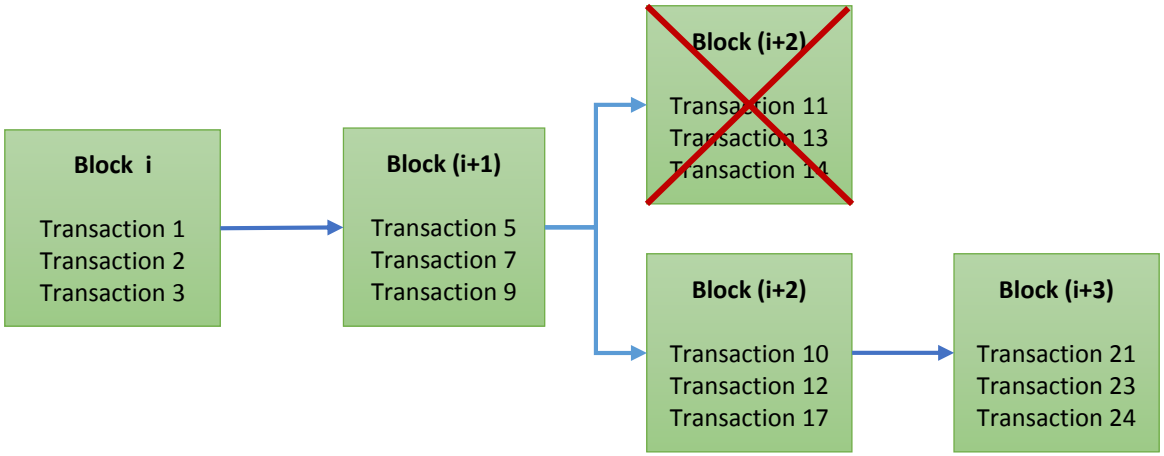
- Block get's broadcasted into the bitcoin network
- What, if two blocks are created at the same time?



Bitcoin

Block chain fork

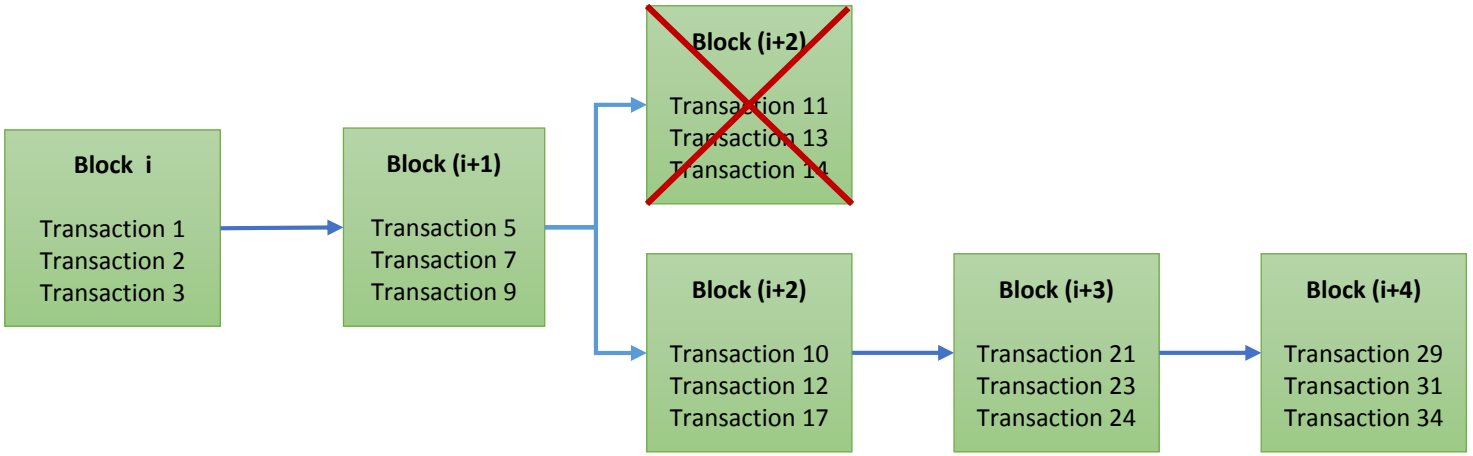
- Block get's broadcasted into the bitcoin network
- What, if two blocks are created at the same time?



Bitcoin

Block chain fork

- Block get's broadcasted into the bitcoin network
- What, if two blocks are created at the same time?



Bitcoin

Confirmed Transactions

Bitcoin

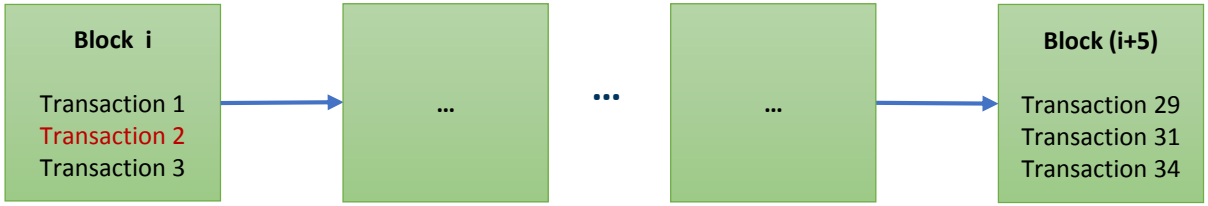
Confirmed Transactions

- Transactions are confirmed
  - If they were included 5 blocks before the current one

Bitcoin

Confirmed Transactions

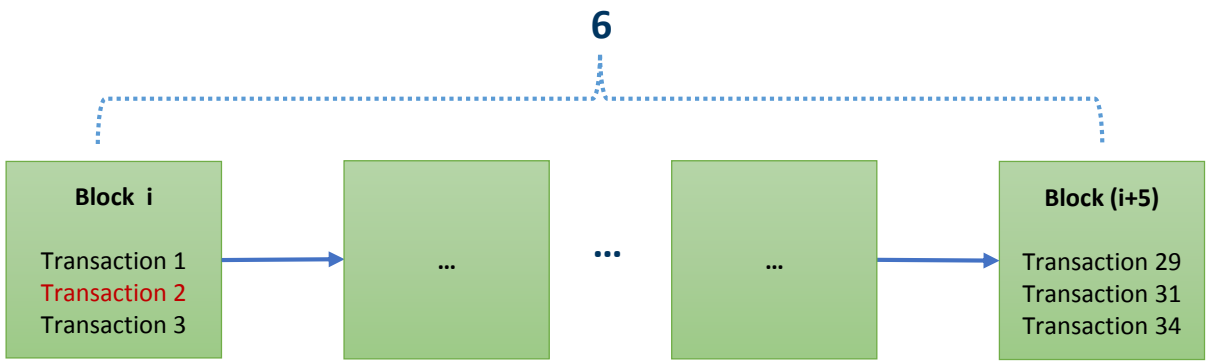
- Transactions are confirmed
  - If they were included 5 blocks before the current one



Bitcoin

Confirmed Transactions

- Transactions are confirmed
  - If they were included 5 blocks before the current one

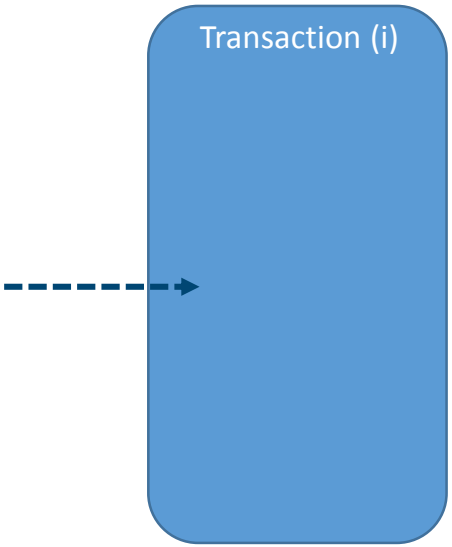


Gridcoin

Transaction - details

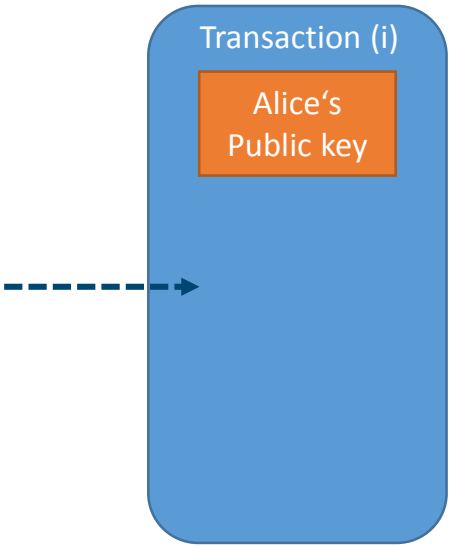
Gridcoin

Transaction - details



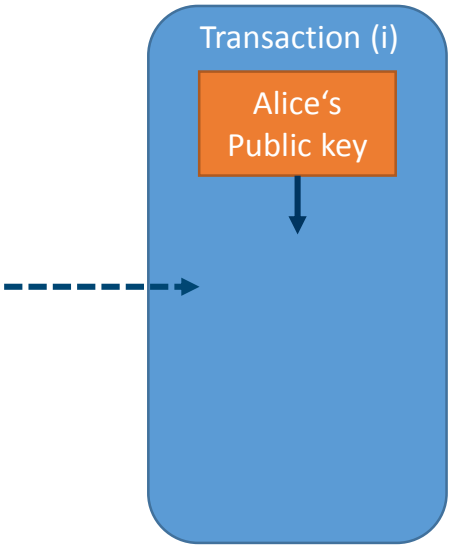
Gridcoin

Transaction - details



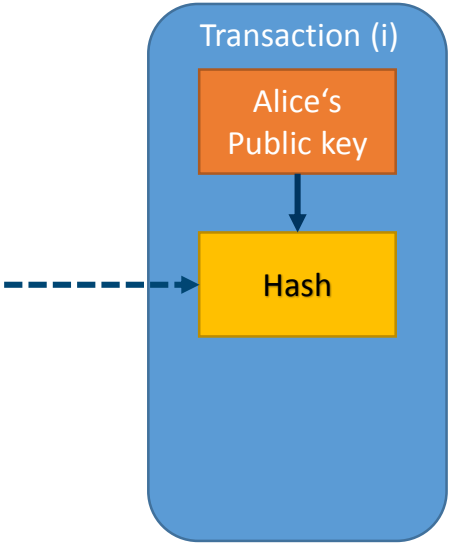
Gridcoin

Transaction - details



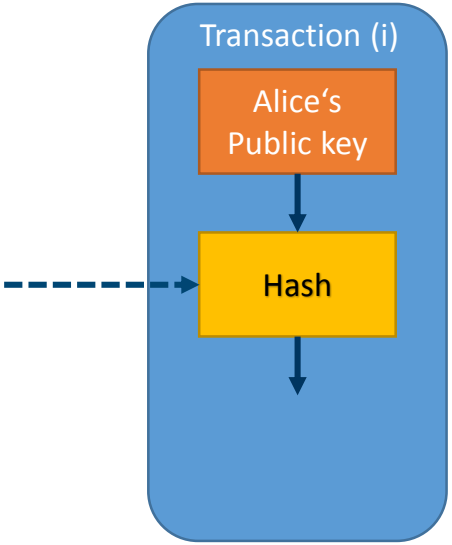
Gridcoin

Transaction - details



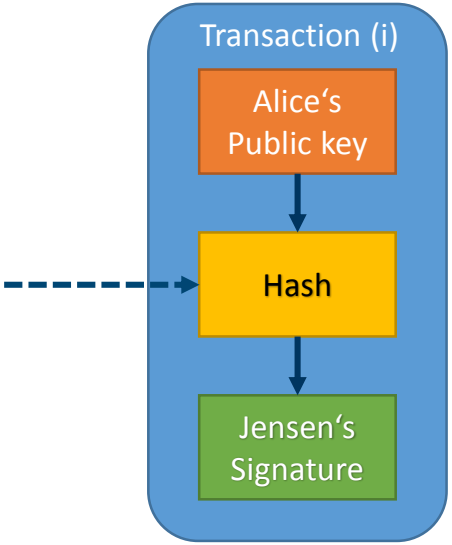
Gridcoin

Transaction - details



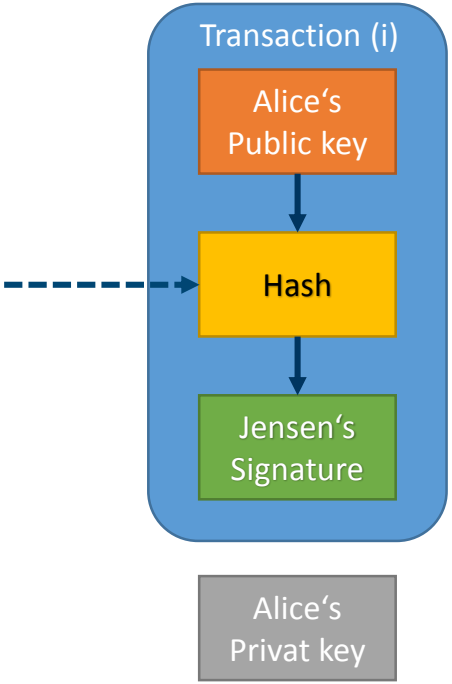
Gridcoin

Transaction - details



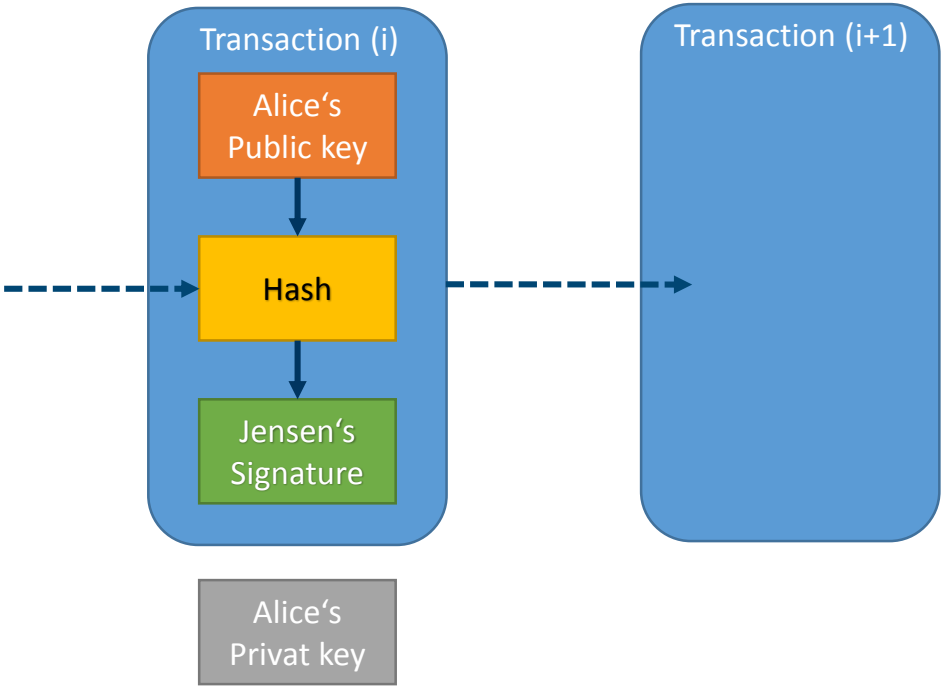
Gridcoin

Transaction - details



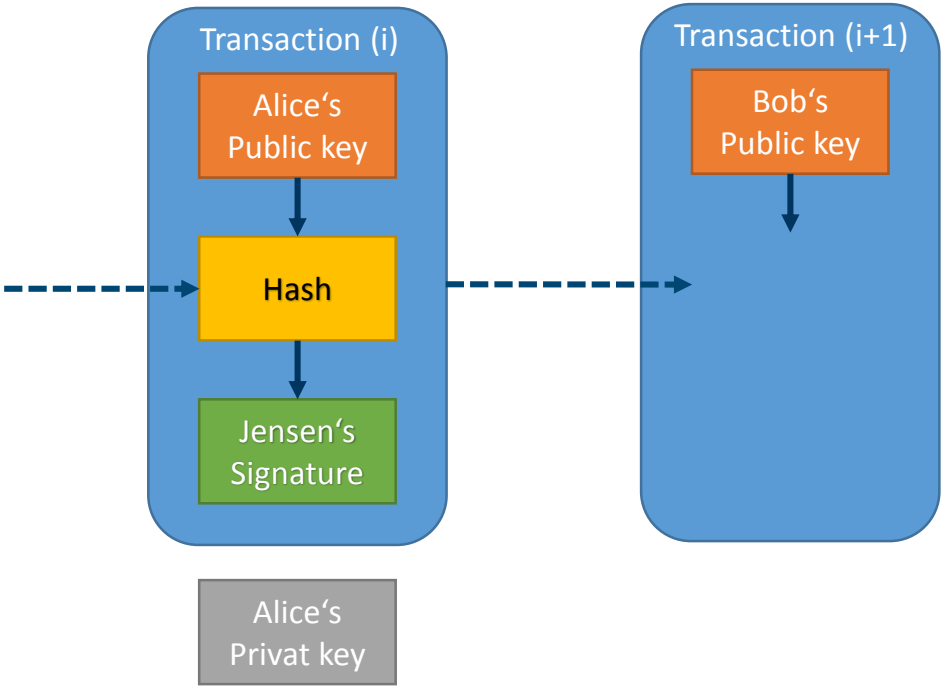
Gridcoin

Transaction - details



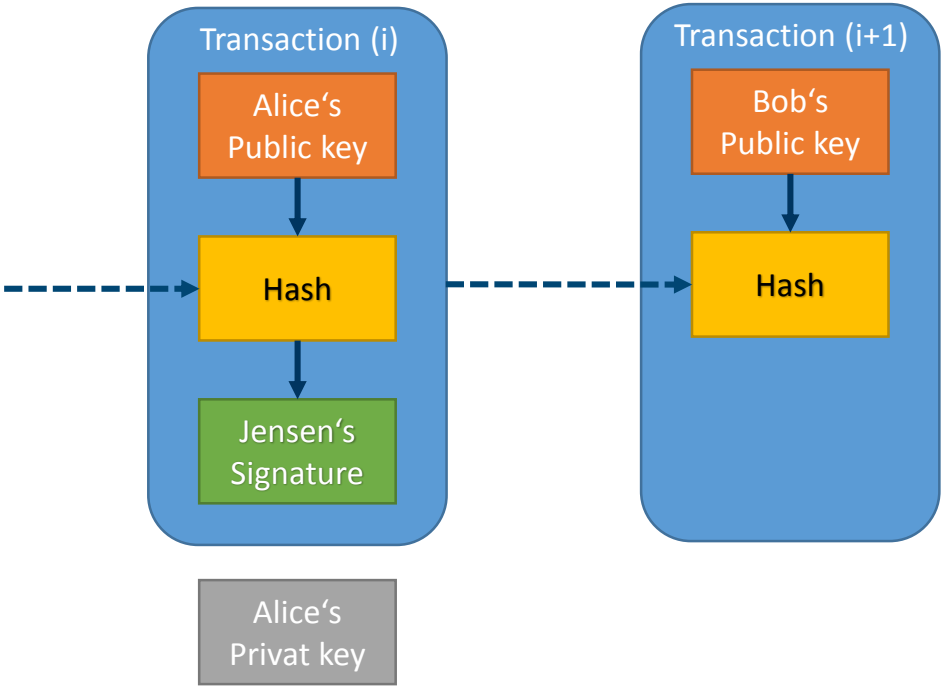
Gridcoin

Transaction - details



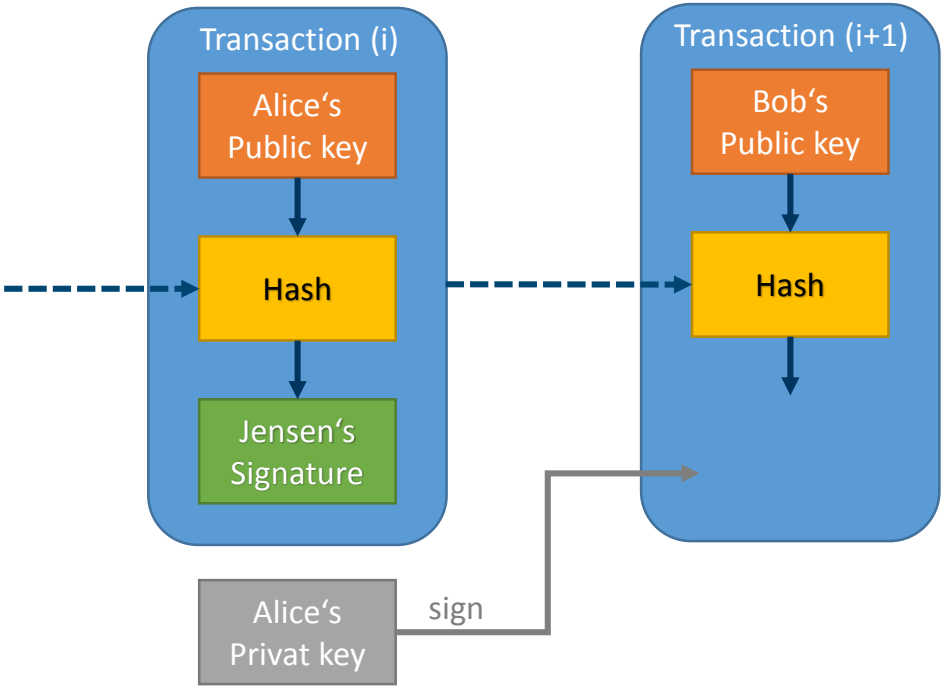
Gridcoin

Transaction - details



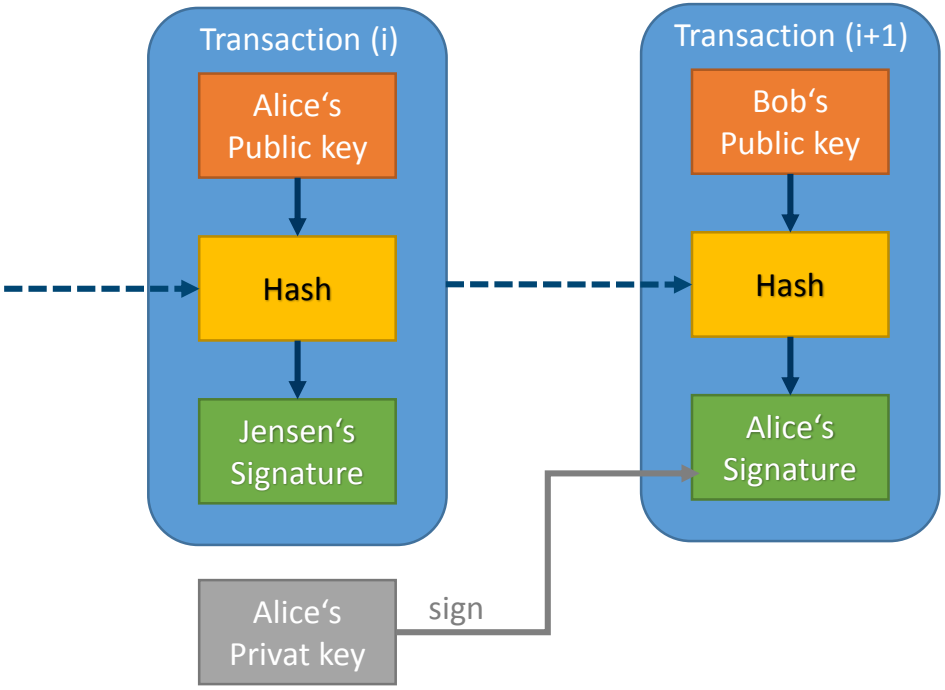
Gridcoin

Transaction - details



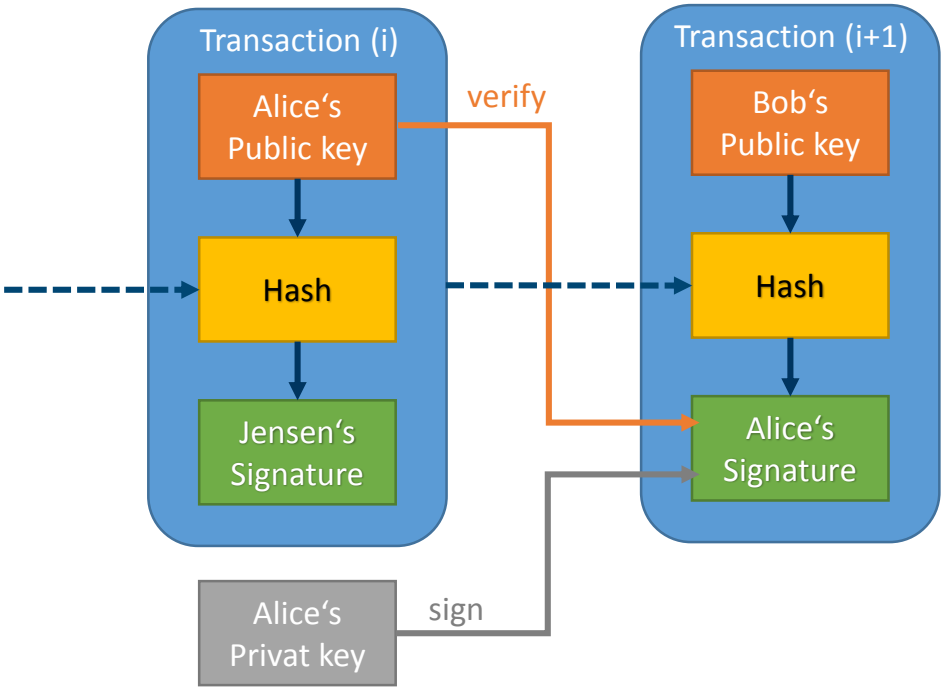
Gridcoin

Transaction - details



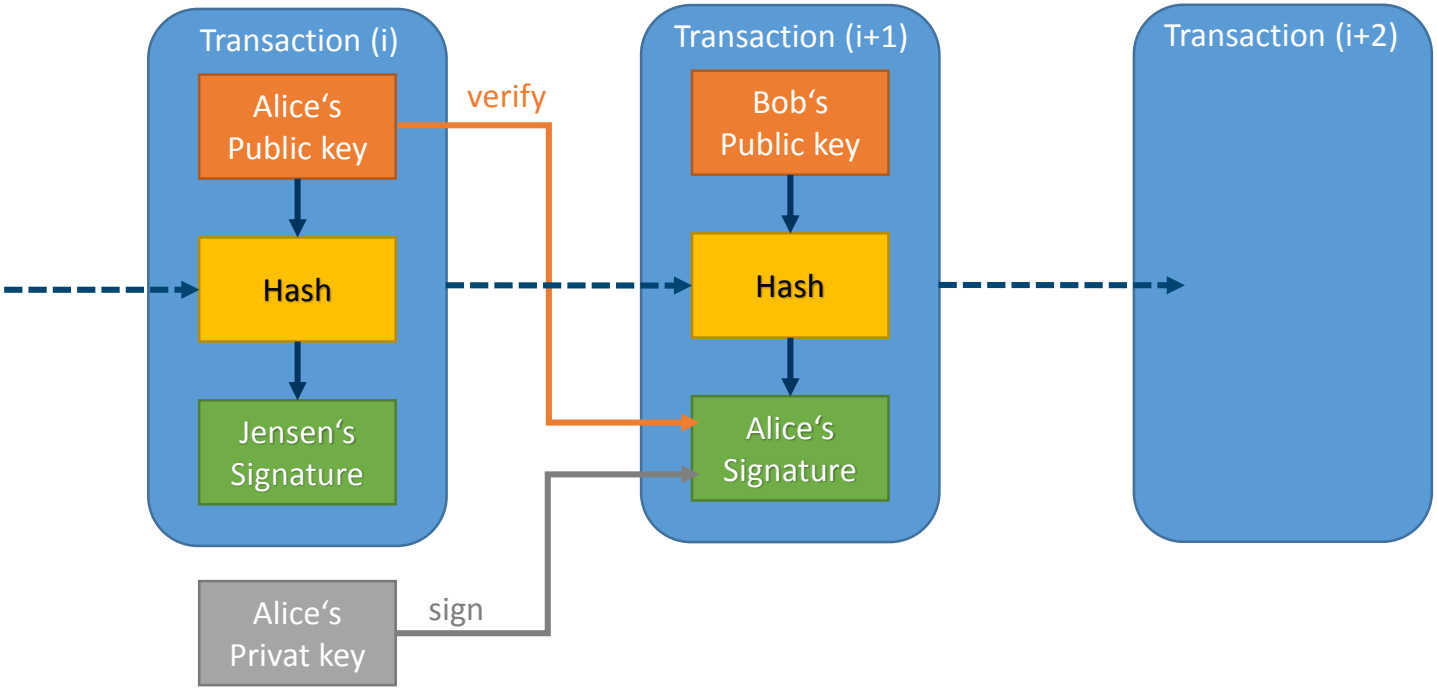
Gridcoin

Transaction - details



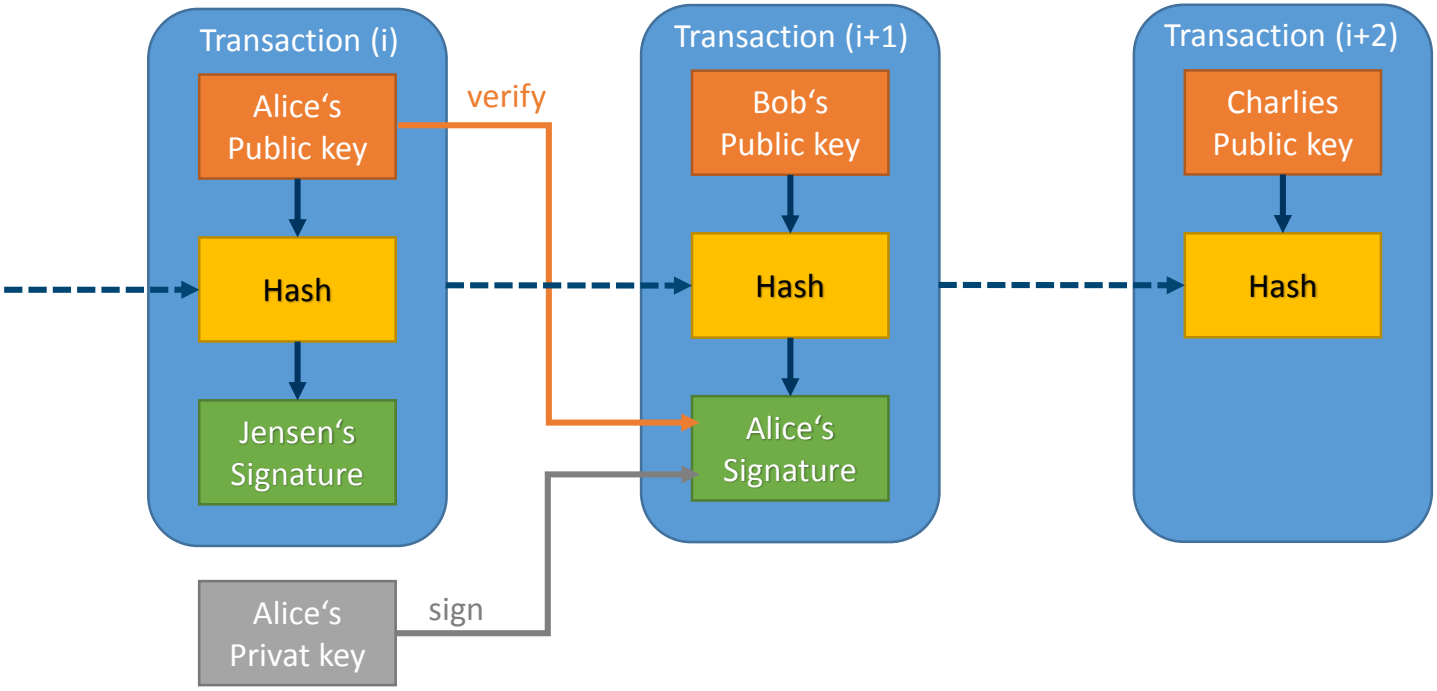
Gridcoin

Transaction - details



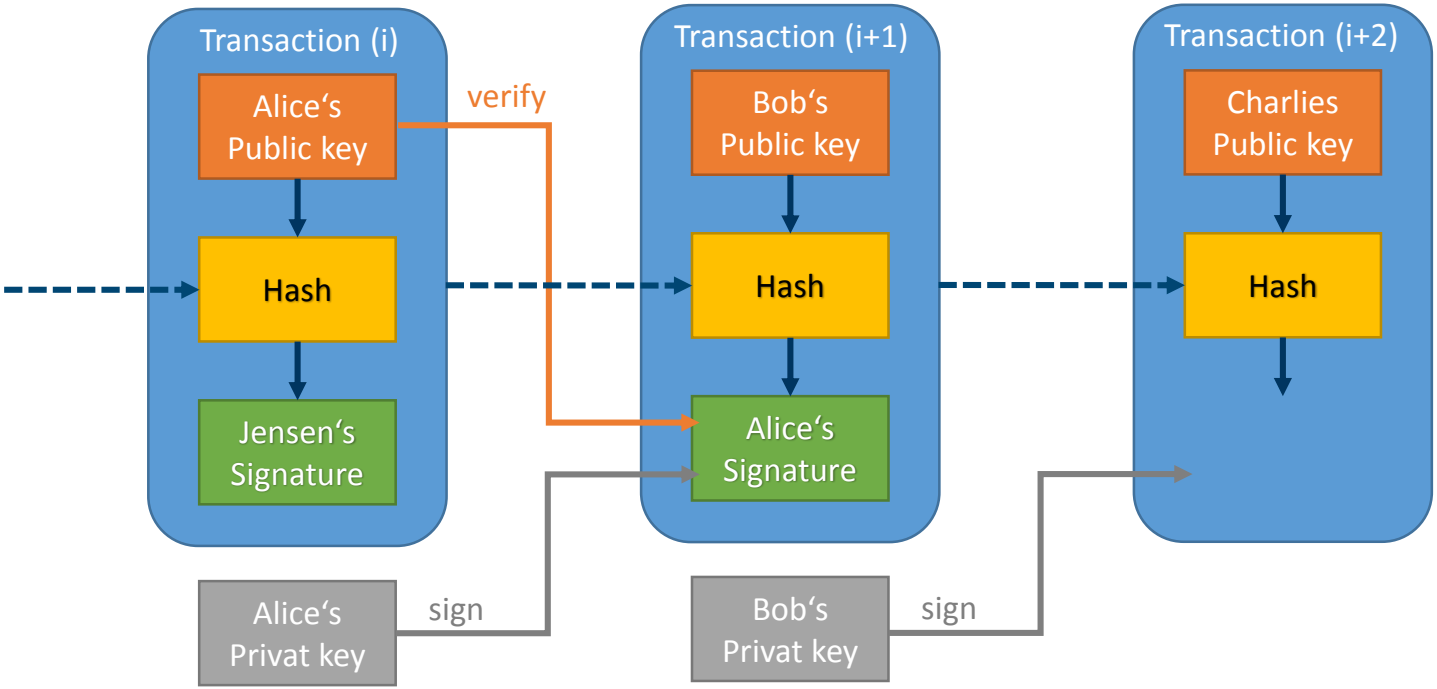
Gridcoin

Transaction - details



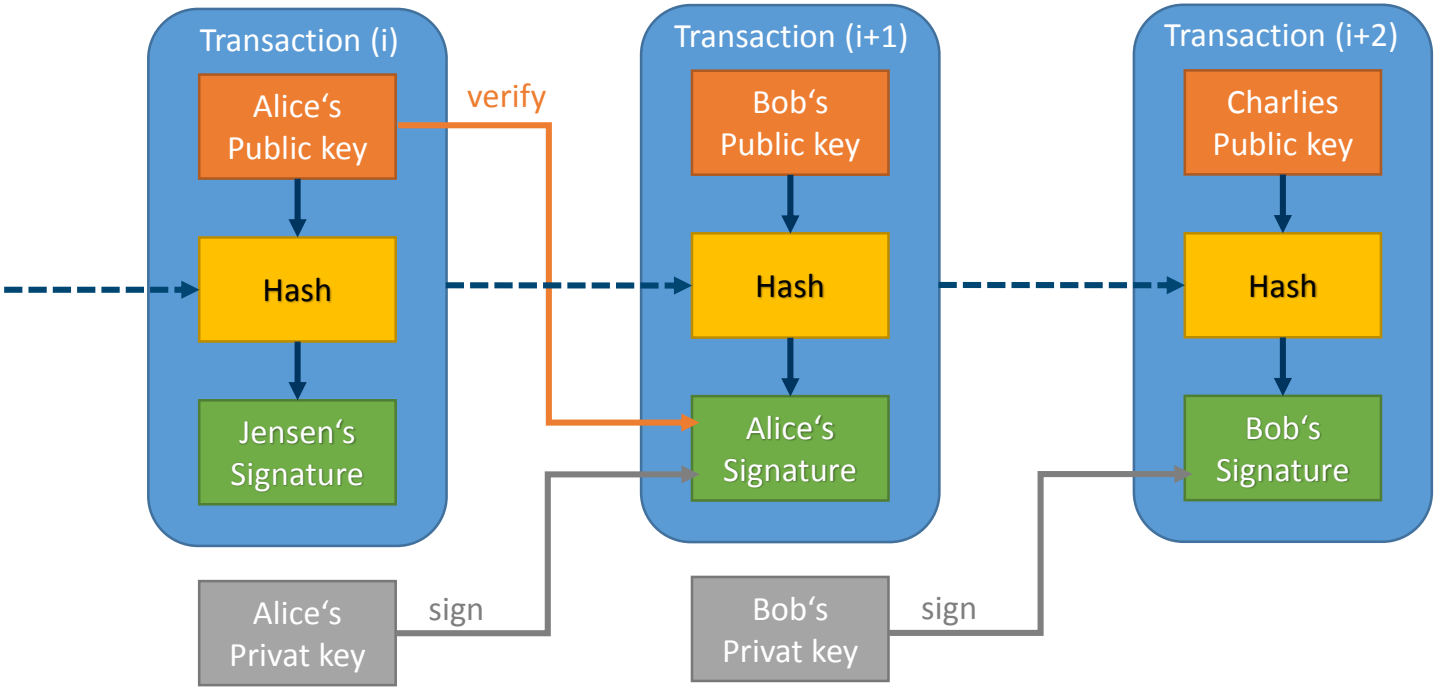
Gridcoin

Transaction - details



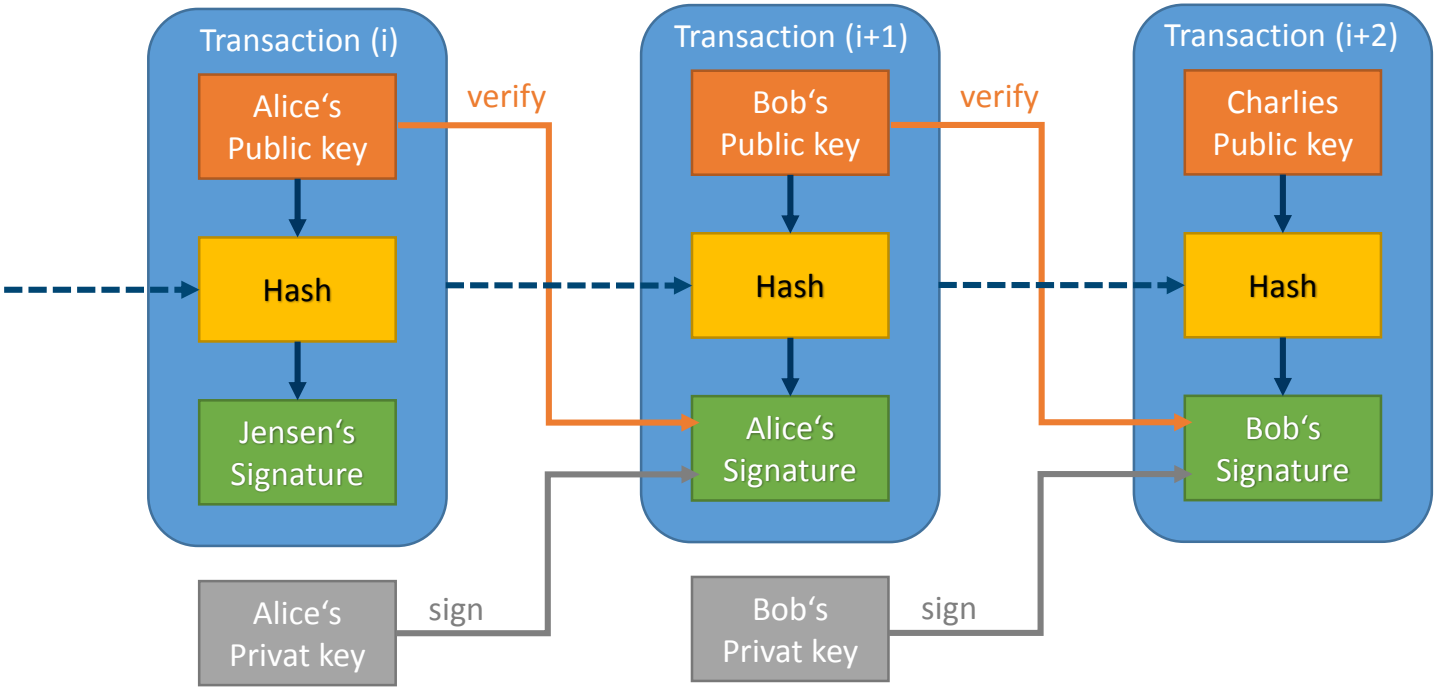
Gridcoin

Transaction - details



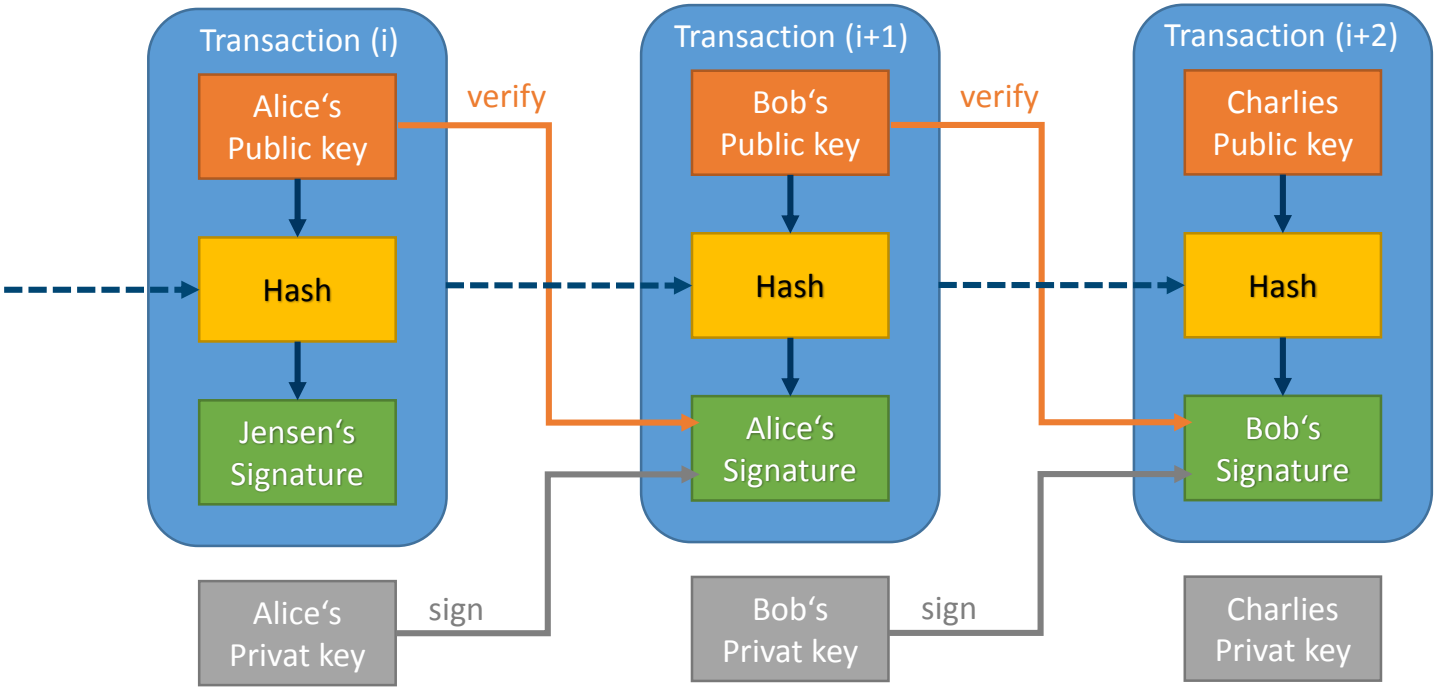
Gridcoin

Transaction - details



Gridcoin

Transaction - details



Bitcoin

Double Spending

- Security of bitcoin PoW
  - total computational power of the honest user
- Double spending still possible
  - Attack has 15% success rate with 30% hash rate of network
  - Attack has 100% success rate with 51% hash rate of network

Bitcoin

Mining

Bitcoin

Mining

- Every time a block is mined
  - Miner get's bitcoin reward

Bitcoin

Mining

- Every time a block is mined
  - Miner get's bitcoin reward
  - Via special transaction (coinbase transaction)
  - Current 12 BTC
  - Halved approx. every 4 years

Bitcoin

Mining

- Every time a block is mined
  - Miner get's bitcoin reward
  - Via special transaction (coinbase transaction)
  - Current 12 BTC
  - Halved approx. every 4 years



Bitcoin

Mining

- Every time a block is mined
  - Miner get's bitcoin reward
  - Via special transaction (coinbase transaction)
  - Current 12 BTC
  - Halved approx. every 4 years





Bitcoin

Mining

- Every time a block is mined
  - Miner get's bitcoin reward
  - Via special transaction (coinbase transaction)
  - Current 12 BTC
  - Halved approx. every 4 years







Bitcoin

Mining

- Every time a block is mined
  - Miner get's bitcoin reward
  - Via special transaction (coinbase transaction)
  - Current 12 BTC
  - Halved approx. every 4 years





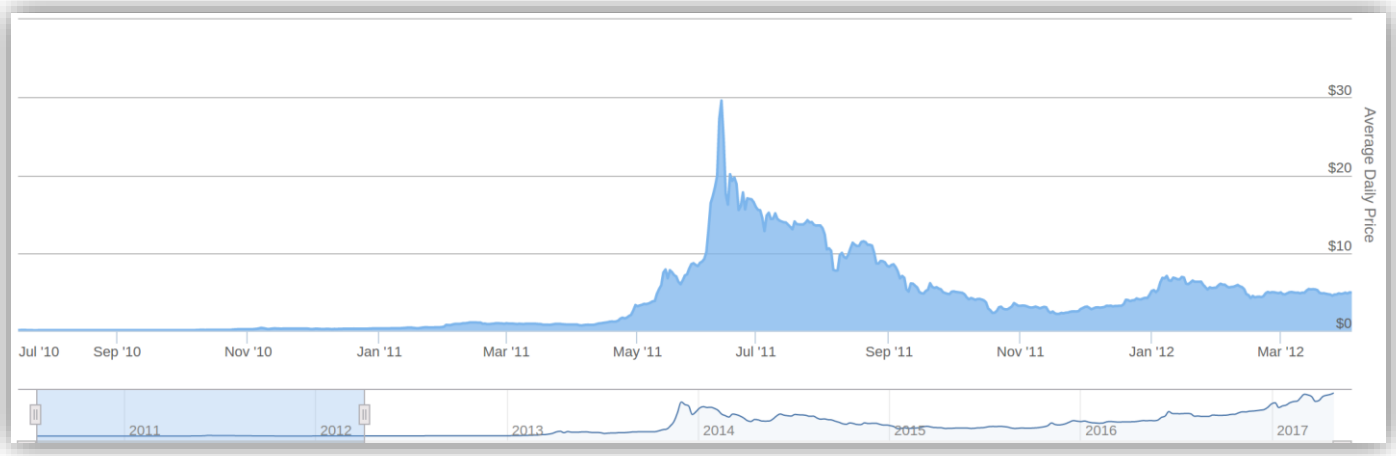




Bitcoin

Bitcoin value over time

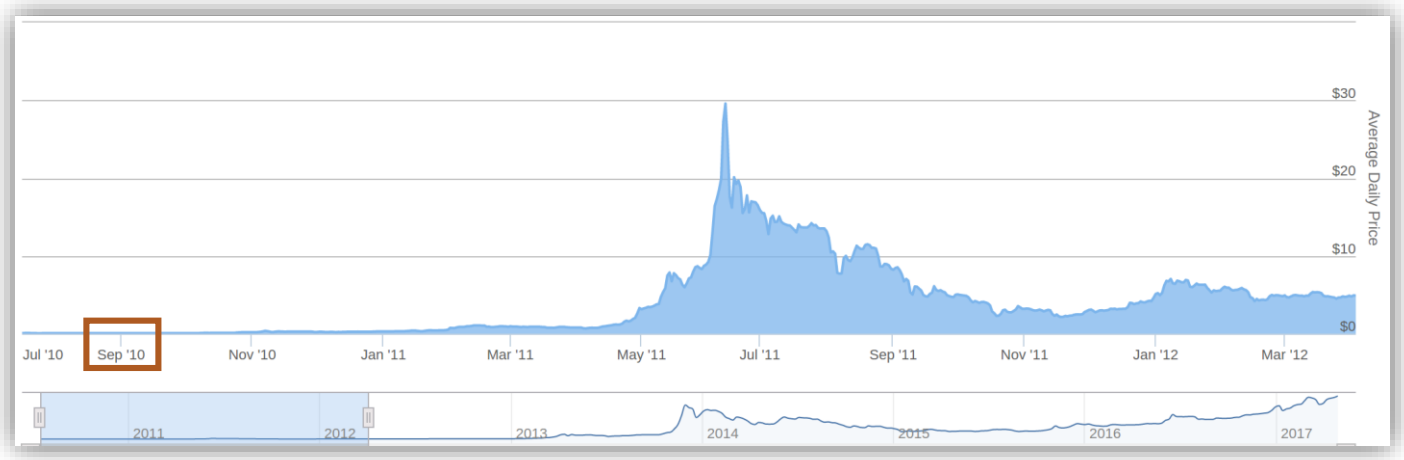
- Huge success



Bitcoin

Bitcoin value over time

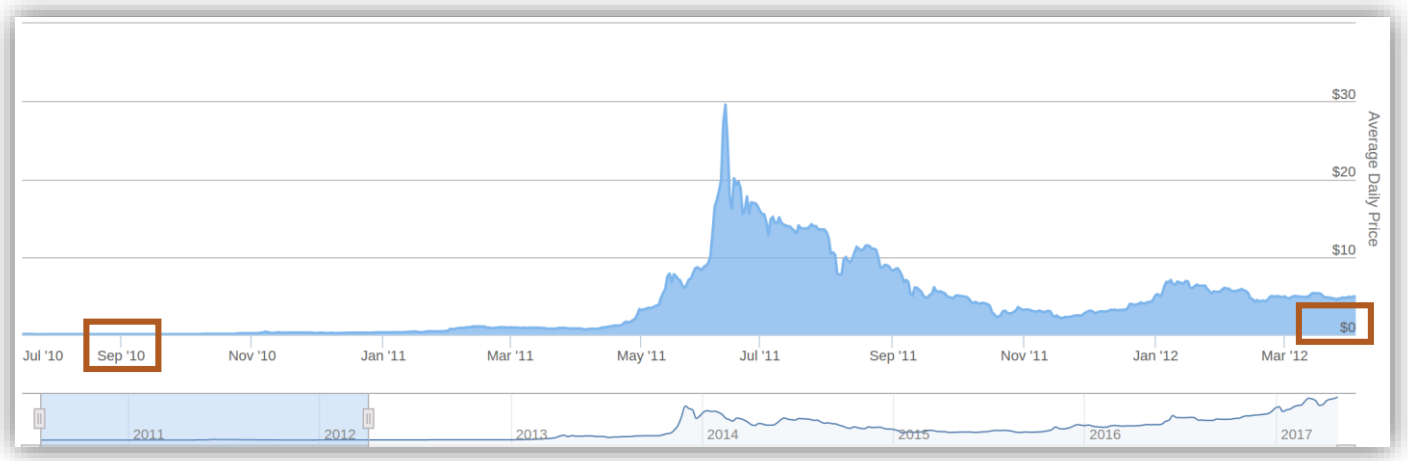
- Huge success



Bitcoin

Bitcoin value over time

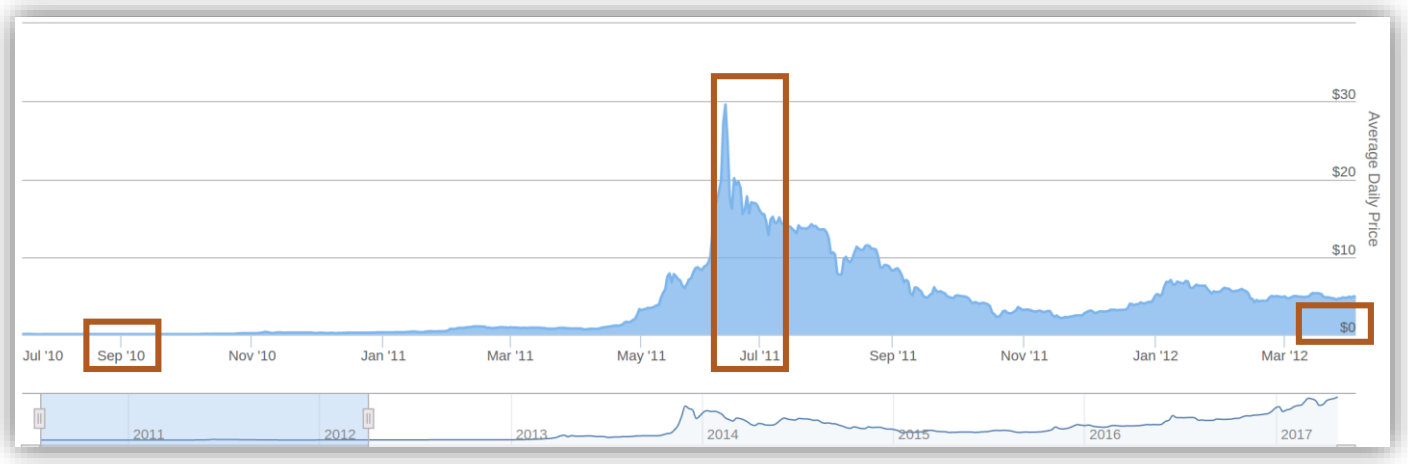
- Huge success



Bitcoin

Bitcoin value over time

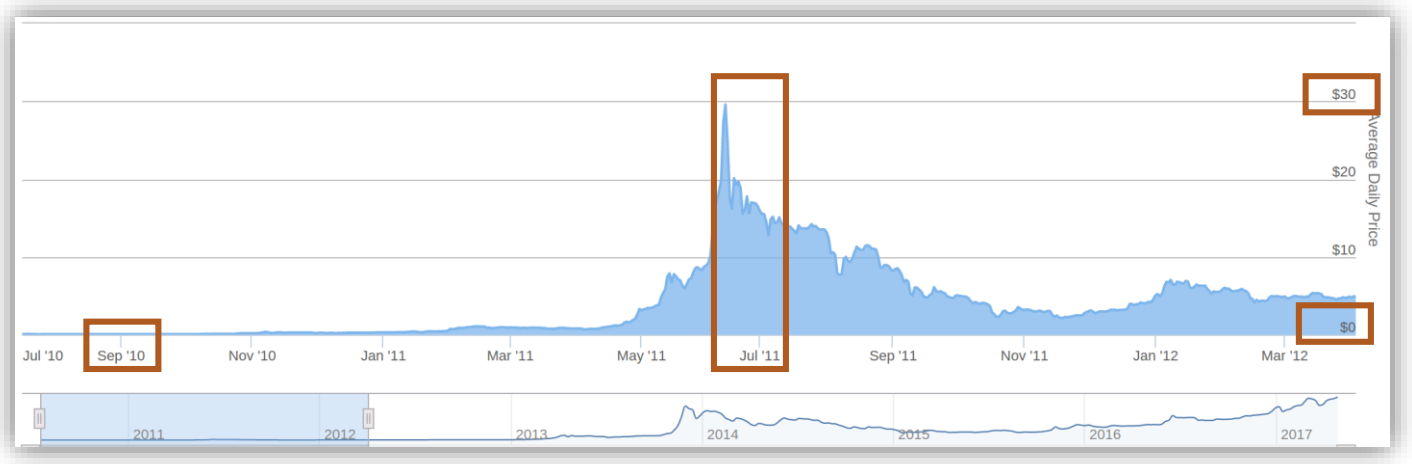
- Huge success



Bitcoin

Bitcoin value over time

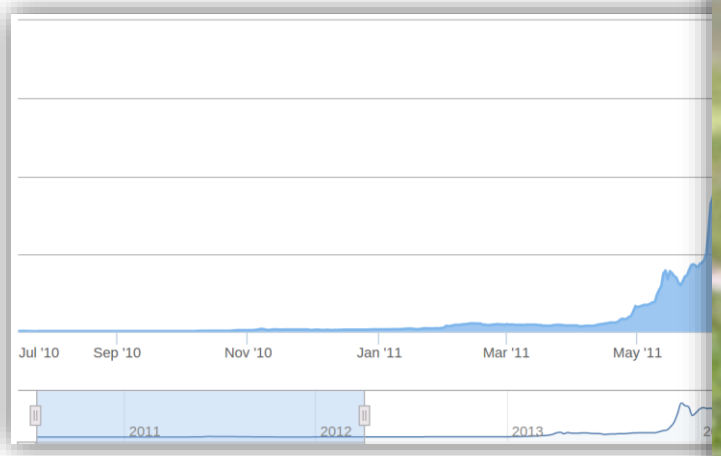
- Huge success



Bitcoin

Bitcoin value over time

- Huge success





Bitcoin

Bitcoin value over time

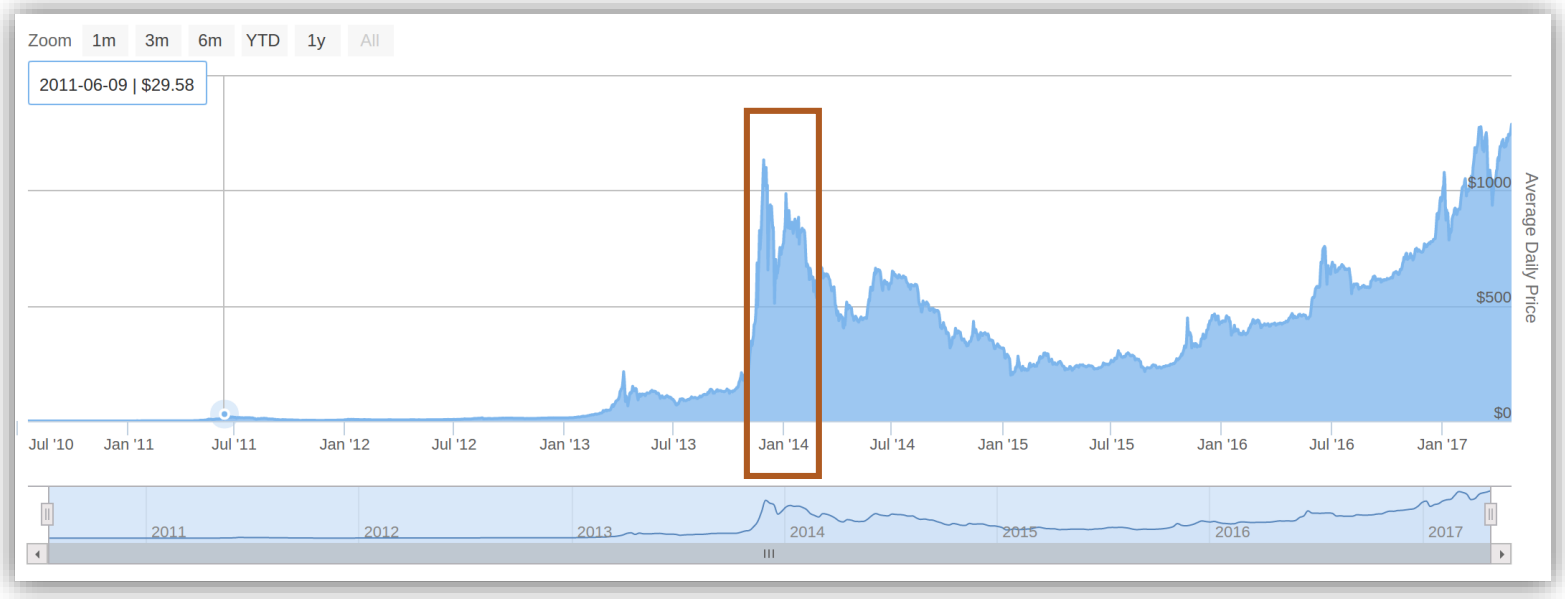
- Huge success



Bitcoin

Bitcoin value over time

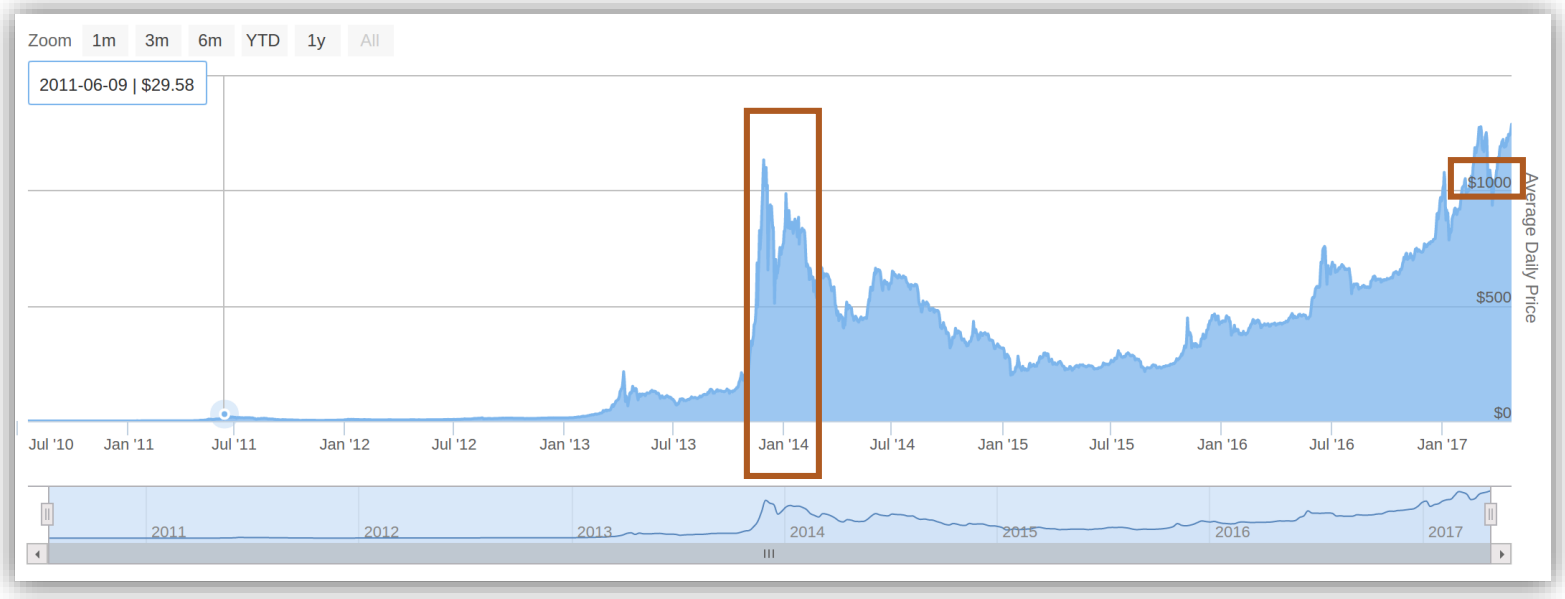
- Huge success



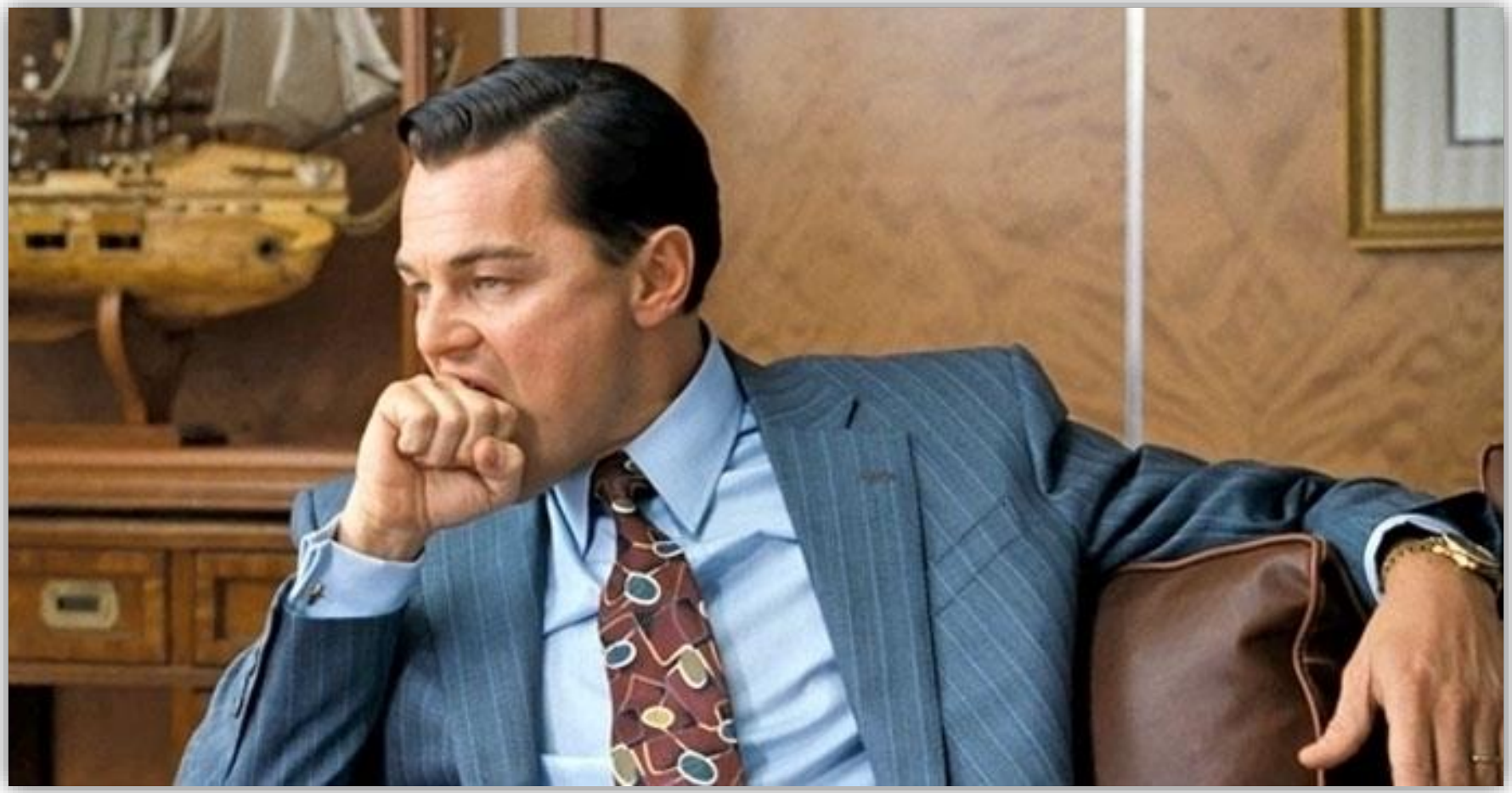
Bitcoin

Bitcoin value over time

- Huge success



Bitcoin



Bitcoin



Bitcoin

Limitations

Bitcoin

Limitations

- Developer regulates block size
  - Limiting the number of transactions per block

Bitcoin

Limitations

- Developer regulates block size
  - Limiting the number of transactions per block
- What if you want to change bitcoin's protocol
- Different hash algorithm

Bitcoin

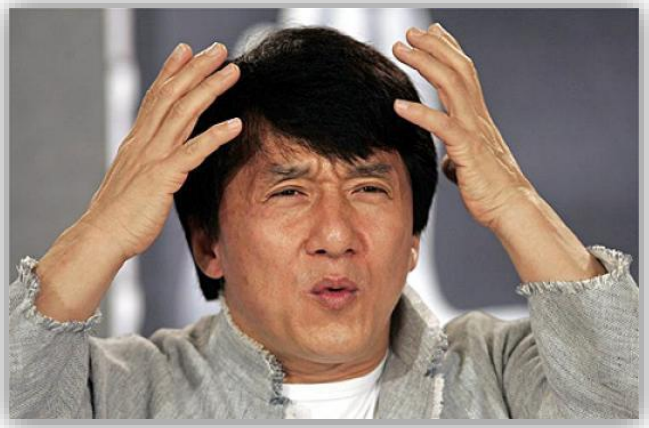
Limitations

- Developer regulates block size
  - Limiting the number of transactions per block
- What if you want to change bitcoin's protocol
- Different hash algorithm
- Etc.

Bitcoin

Limitations

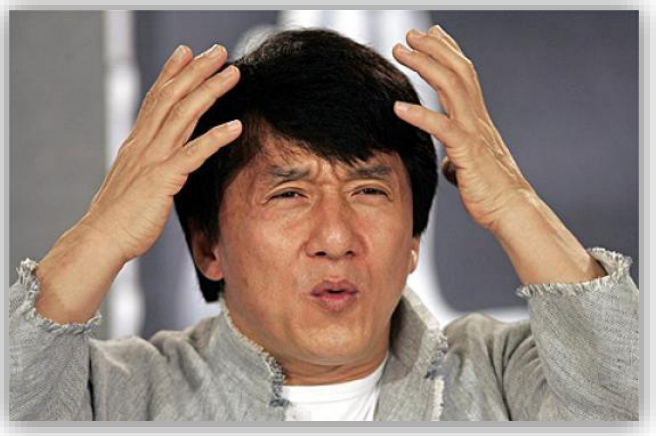
- Developer regulates block size
  - Limiting the number of transactions per block
- What if you want to change bitcoin's protocol
- Different hash algorithm
- Etc.



Bitcoin

Limitations

- Developer regulates block size
  - Limiting the number of transactions per block
- What if you want to change bitcoin's protocol
- Different hash algorithm
- Etc.



Source code is public, let's modify it!

Sources

Slide 1:

- startin-slide-picture-cryptocurrency.jpg **http://gaincryptocurrency.altervista.org/alterpages/10710353_865938750143391_6543553170888387417_o.jpg**

Slide 4:

- fiat-logo.png **https://en.wikipedia.org/wiki/File:Fiat_Logo.svg**
- euro.jpg **https://www.bpb.de/cache/images/5/220135-3x2-article620.jpg?FF353**

Slide 5:

- altcoin-chips.png **https://i2.wp.com/www.altcointoday.com/wp-content/uploads/2015/10/coins.png**

Slide 6:

- Examples for trading **https://www.slideshare.net/jonwaller0/what-isbitcoinen (Slide 13)**
- Benefits of Cryptocurrencies **http://www.huffingtonpost.com/ameer-rosic-/7-incredible-benefits-of-_1_b_13160110.html**
- skeptical-kid-blank.jpeg **https://img.memesuper.com/fc693ae7e695c4eea4034dc354bc356d_-kid-meme-blank-skeptical-meme-skeptical_305-305.jpeg**

Slide 8:

- Bitcoin Pizzas forum entry **https://bitcointalk.org/index.php?topic=137**
- BC_Logo.png **https://en.bitcoin.it/wiki/File:BC_Logo_.png**
- Pizza.jpg **http://heliacal.net/~solar/bitcoin/pizza/**

Slide 9:

- Bitcoin_exchange.jpg **https://www.coinbase.com/**
- bitcoin-atm.jpg **https://cache.caricoin.com/wp-content/uploads/38937_01_los_angeles_receives_first_bitcoin_atm_machines_in_the_united_states_full-1080x675.jpg**
- withdraw-bitcoin-to-usd-1024x502.jpg **http://cashinbitcoin.com/wp-content/uploads/2017/02/withdraw-bitcoin-to-usd-1024x502.jpg**

Sources

Slide 10-13:

- Idea/approach adapted from **http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/**

Slide 14:

- Success for double spending based on hash rate **https://arxiv.org/pdf/1402.2009.pdf**

Slide 16:

- bitcoin-mining-asics.jpg **https://i.ytimg.com/vi/SNB11HPv_lo/maxresdefault.jpg**
- bitcoin-mining-fpga.jpg **http://cdn5.thinkcomputers.org/wp-content/uploads/2014/02/mining-rig-26.jpg**
- bitcoin-mining-farm.jpg **https://cdn.static-economist.com/sites/default/files/images/print-edition/20150110_WBP001_0.jpg**
- bitcoin-cpu-cluster.jpg **http://bedroombitcoinbeast.blogin.com.au/tag/bitcoin-mining/**

Slide 17:

- bitcoin-price-rate-2010-2012.png **https://99bitcoins.com/price-chart-history/**

Slide 18:

- bitcoin-price-rate-2010-2012.png **https://99bitcoins.com/price-chart-history/**
- happy-kid.jpg **http://www.relatably.com/m/img/skeptical-african-kid-meme-9gag/c9acfd4d17a9ba06203d833739b482a1.jpg**

Slide 19:

- bitcoin-price-rate-chart-all-time.png **https://99bitcoins.com/price-chart-history/**

Slide 20:

- wolf-of-wallstreet-fist-bite.jpeg **http://img.pandawhale.com/159212-Leo-Dicaprio-biting-knucle-mem-29ZB.jpeg**

Slide 21:

- wolf-of-wall-street-laugh.jpg **https://ewedit.files.wordpress.com/2015/01/wolf-of-wall-street-leonardo-dicaprio1.jpg?w=612**

Sources

Slide 22:

- jackie-chan-why-blank.jpg **https://houseofgeekery.files.wordpress.com/2012/08/jackie-chan-whut.jpg**

Slide 24:

- altcoin-chips.png **https://i2.wp.com/www.altcointoday.com/wp-content/uploads/2015/10/coins.png**

Slide 25:

- bitcoin-energy-consumption.png **https://www.iea.org/publications/freepublications/publication/KeyWorld2016.pdf**
- Ddoscoin **https://github.com/ewust/DDoSCoin**
- Burstcoin **https://www.burst-team.us/**
- Namecoin **https://namecoin.org/**

Slide 27:

- boinc_600.jpg **https://boinc.berkeley.edu/logo/boinc_600.jpg**
- peercoin-logo.svg **https://peercoin.net/assets/img/logos/peerunity.svg**

Slide 28:

- broadband_rfi1.jpg **https://setiathome.berkeley.edu/sah_glossary/images/broadband_rfi1.jpg**
- folding-at-home.jpg **http://images.bit-tech.net/content_images/2009/06/what-is-folding-and-why-does-it-matter/fold1.jpg**

Slide 35:

- lock-hit-by-bullet.jpg **http://media.coindesk.com/uploads/2016/08/lock-broken-728x485.jpg**

Slide 37:

- breaking-gridcoin.jpg **http://cdn23.us1.fansshare.com/photos/breakingbad/ndqzmjm-mzuz-disea-la-intro-de-tus-series-favoritas-breaking-bad-logo-642337318.jpg**

Slide 39:

- skeptical-baby.png **http://i0.kym-cdn.com/entries/icons/original/000/008/402/Screen_shot_2012-01-18_at_5.28.23_PM.png**

Sources

Slide 45:

- Steve-Carell-Facepalm.gif **http://gif-finder.com/wp-content/uploads/2015/02/Steve-Carell-Facepalm.gif**
- fence-fails.jpg **https://s-media-cache-ak0.pinimg.com/564x/4b/c8/af/4bc8afa3a8bb0735fc703736aa7c986f.jpg**

Slide 46:

- make-money-memes-money-meme_300-300.jpeg **https://img.memesuper.com/9bc3ddad6db0f1757f2a02d7ffad7500_make-money-memes-money-meme_300-300.jpeg**

Cryptocurrency

In general

- Only digital



Cryptocurrency

In general

- Only digital
- Based on cryptography
  - Asym. cryptography
  - Hash functions



Cryptocurrency

In general

- Only digital
- Based on cryptography
  - Asym. cryptography
  - Hash functions
- Decentralized (Peer-to-Peer network)



Cryptocurrency

In general

- Only digital
- Based on cryptography
  - Asym. cryptography
  - Hash functions
- Decentralized (Peer-to-Peer network)
- Finite supply

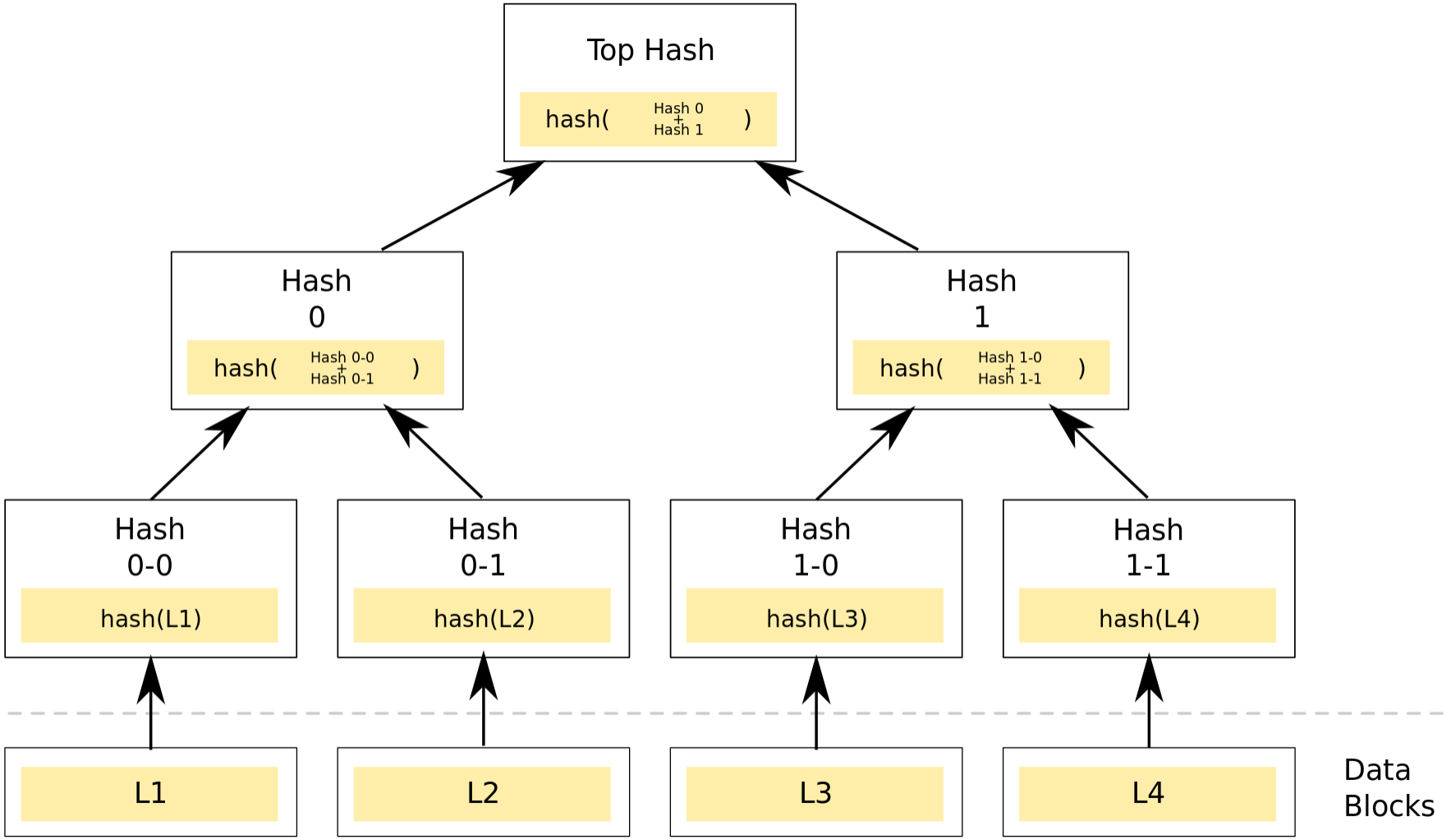


Cryptocurrency

In general

- Only digital
- Based on cryptography
  - Asym. cryptography
  - Hash functions
- Decentralized (Peer-to-Peer network)
- Finite supply
- Pseudonym/anonym




Top Hash
hash(Hash 0 + Hash 1)
Hash 0
hash(Hash 0-0 + Hash 0-1)
Hash 1
hash(Hash 1-0 + Hash 1-1)
Hash 0-0
hash(L1)
Hash 0-1
hash(L2)
Hash 1-0
hash(L3)
Hash 1-1
hash(L4)
L1
L2
L3
L4
Data Blocks