# A History of Bitcoin Security

(in 5 minutes)

Dec 4th, 2020

John Newbery

brink

# The Satoshi Era

*“It would help if people stop generating”*

**Date:** *2010-08-15*

**BTC Price:** *~$0*

**Total BTC value:** *~$0*

# Example: Buffer overflow bug

- Overflow bug allows infinite Bitcoin to be created by anyone
- Fix is a consensus change
- Fix rolled out as new executable “by fiat”
- Essentially no public review process

**satoshi**
Founder
Sr. Member

Activity: 364
Merit: 2599

## Re: overflow bug SERIOUS
August 15, 2010, 09:06:45 PM #7

It would help if people stop generating. We will probably need to re-do a branch around the current one, and the less you generate the faster that will be.

A first patch will be in SVN rev 132. It's not uploaded yet. I'm pushing some other misc changes out of the way first, then I'll upload the patch for this.

Author | Topic: Please upgrade to 0.3.8! (Read 11325 times)

**satoshi**
Founder
Sr. Member

## Please upgrade to 0.3.8!
August 03, 2010, 11:40:18 PM #1

Version 0.3.8 adds an important security improvement. Everyone should upgrade to get this change.

The new safety feature displays a warning message in the status bar and locks down RPC if it detects a problem

Author | Topic: *** ALERT *** Upgrade to 0.3.6 (Read 25712 times)

**satoshi**
Founder
Sr. Member

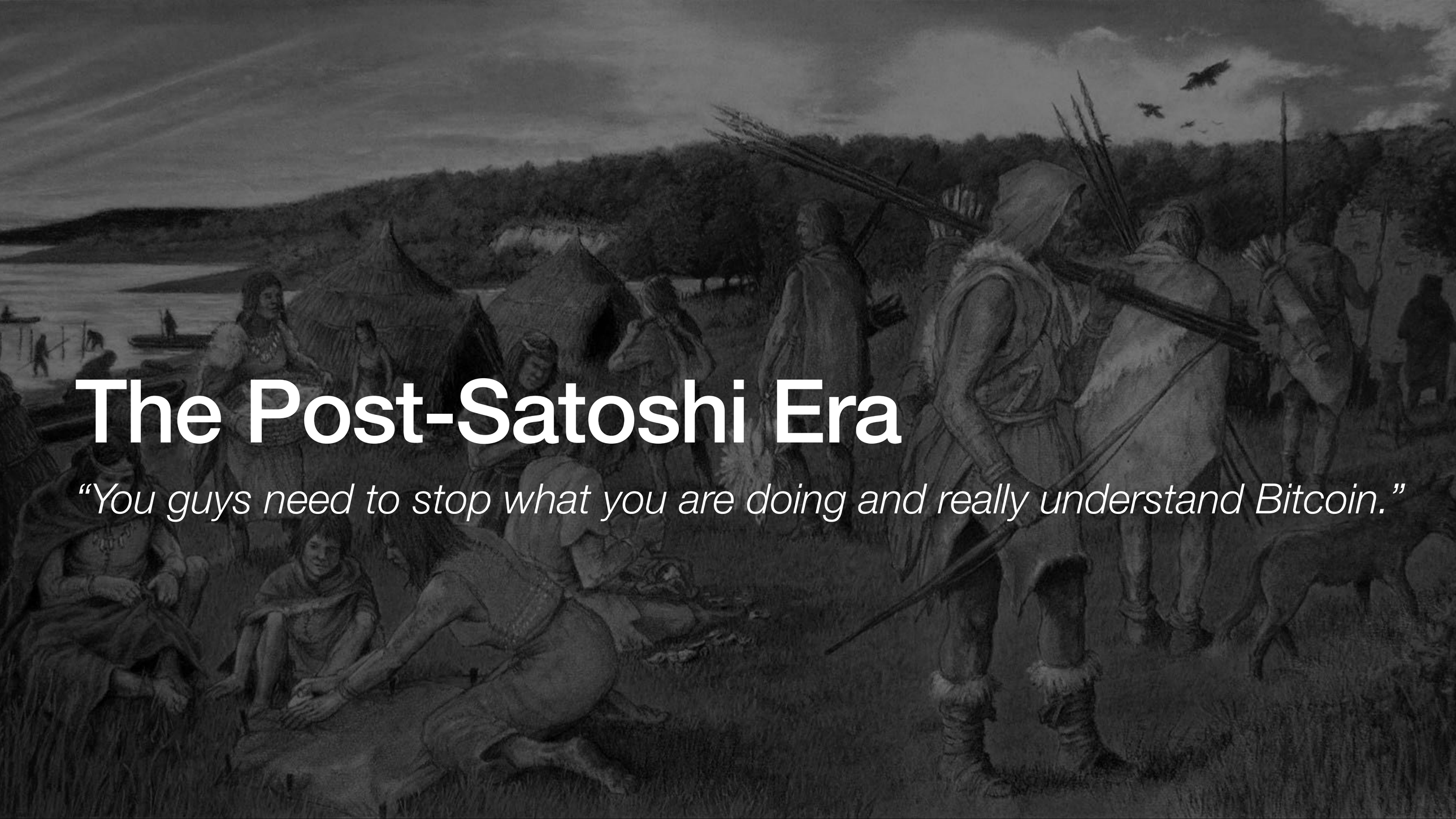
Activity: 364
Merit: 2599

## *** ALERT *** Upgrade to 0.3.6
July 29, 2010, 07:13:06 PM #1

Please upgrade to 0.3.6 ASAP! We fixed an implementation bug where it was possible that bogus transactions could be displayed as accepted. Do not accept Bitcoin transactions as payment until you upgrade to version 0.3.6!

If you can't upgrade to 0.3.6 right away, it's best to shut down your Bitcoin node until you do.

Also in 0.3.6, faster hashing:
- midstate cache optimisation thanks to tcatm
- Crypto++ ASM SHA-256 thanks to BlackEye



# The Post-Satoshi Era

*“You guys need to stop what you are doing and really understand Bitcoin.”*

**Date:** *2011-12-27*

**BTC Price:** *~$4*

**Total BTC value:** *~$30,000,000*

# Example: OP_EVAL

- OP_EVAL (alternative proposal to P2SH) merged with minimal review
- After merge, Russell O’Connor points out fatal flaw
- OP_EVAL code reverted

gavin

**roconnor** on Dec 27, 2011

Currently in the OP_EVAL processing code you have:

```
if (!EvalScriptInner(stack, subscript, txTo, nIn, nHashType, pbegincodehash, pendcodehash, nOpCount, nSigOpCount
```

The postfix ++ operator returns the unincremented value of the variable.

So my understanding is that (1) this doesn't limit the depth of recursive calls and (2) this does limit the number of non-recursive calls you OP_EVAL you have in one script.

In particular (1) implies that that Gavin's example (why wasn't this tested) of "OP_PUSHDATA {OP_DUP OP_EVAL} OP_DUP OP_EVAL" should run in an infinite loop (though I haven't tested this).

<rant>

**laanwj** on Dec 27, 2011 Member

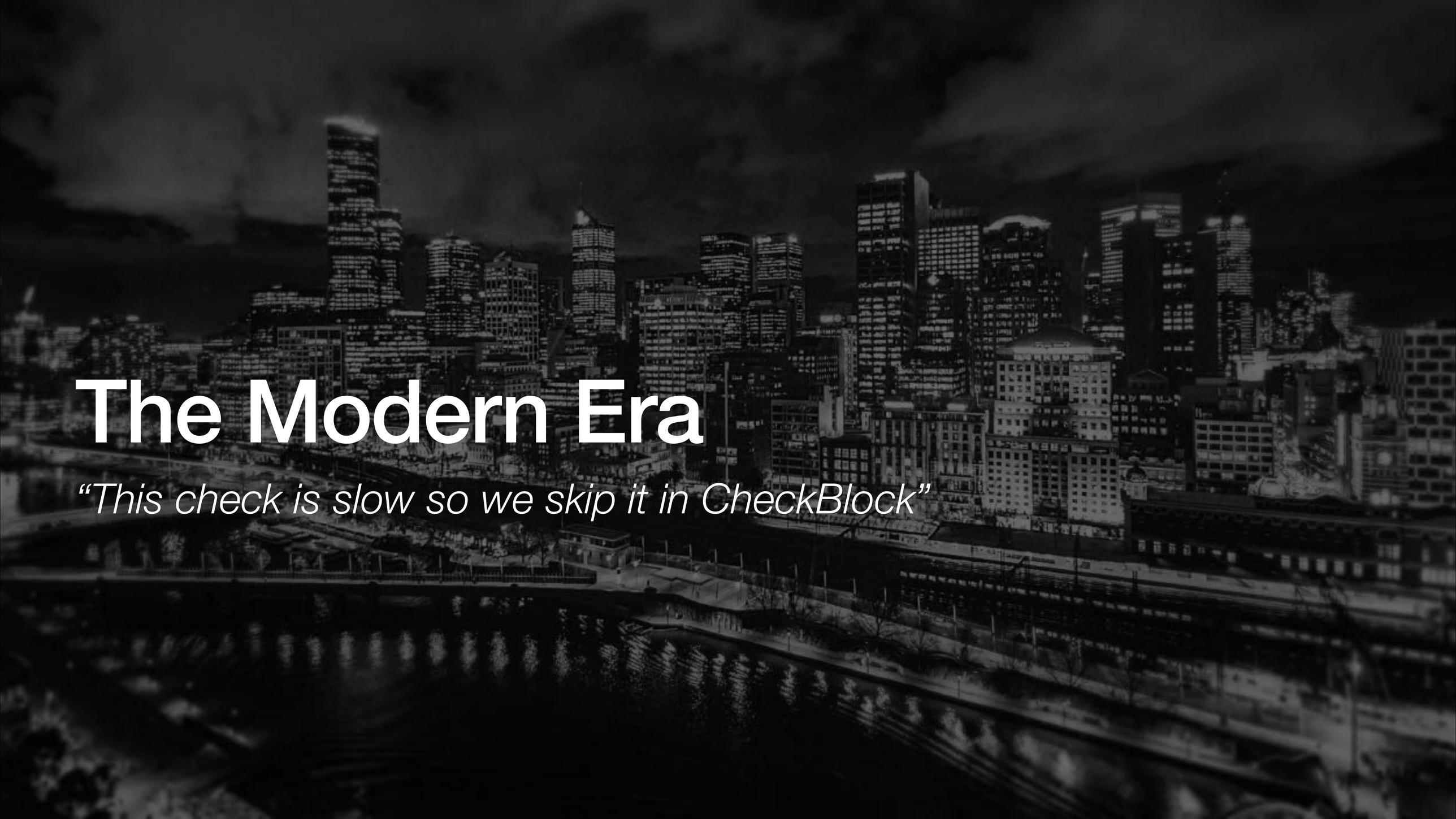
Agreed **@gavinandresen** I don't think simply delaying the change would have won much. People hardly look at "non-official" branches.

Somehow we should encourage more people to take a look at the source carefully, and attack it from any angle possible. Thanks a lot **@roconnor**.

</rant>

70 minutes
plete
ly held in

ecification
on the
proving that
ll have the


The Modern Era
“This check is slow so we skip it in CheckBlock”

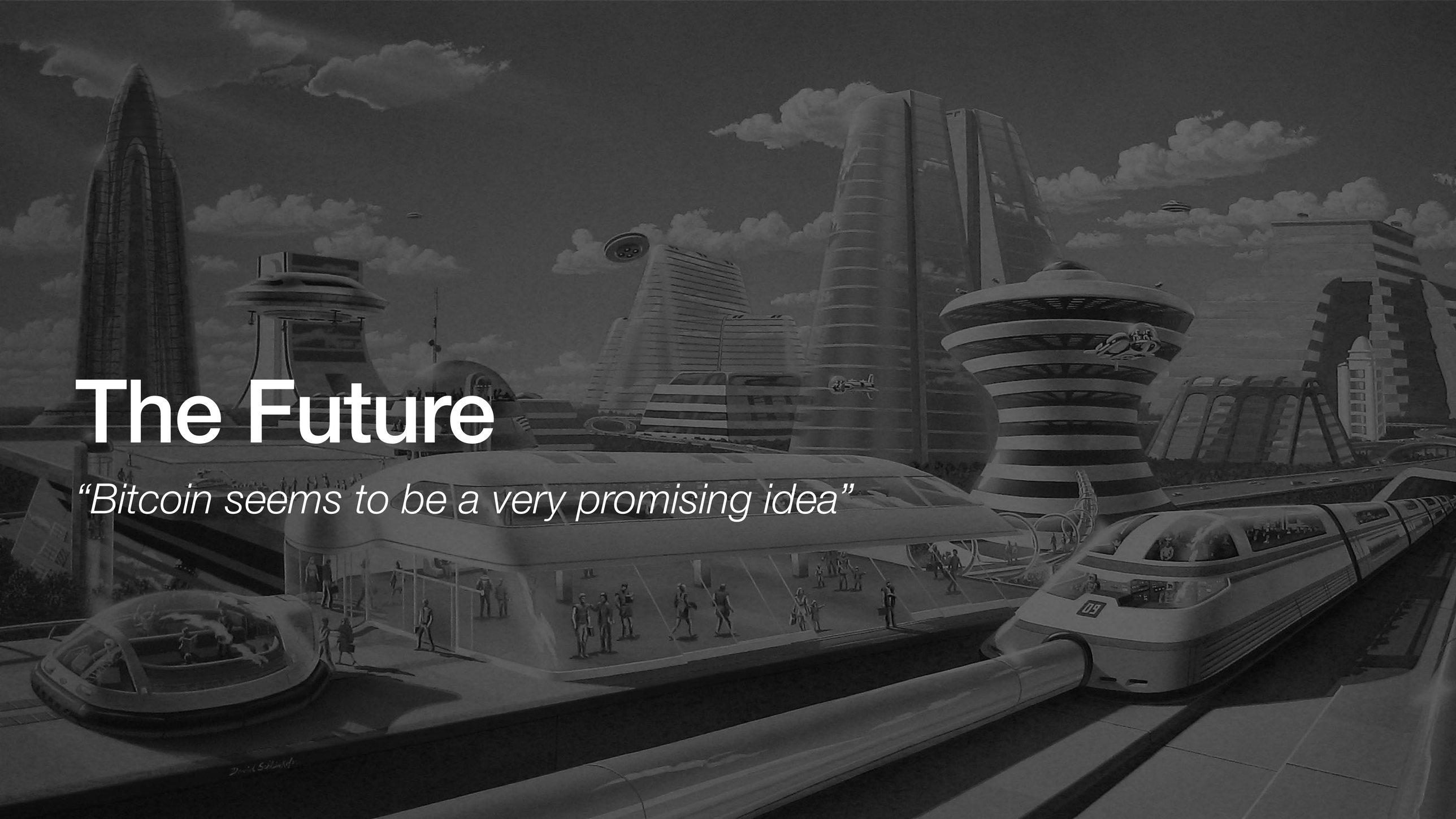
**Date:** *2018-09-17*

**BTC Price:** *~$6,500*

**Total BTC value:** *~$110,000,000,000*

# Example: CVE-2018-17144

- A change was made in Nov 2016 to speed up block propogation
- A separate change was made in June 2017 in the way unspend outputs are stored
- Both were reviewed by experienced developers
- Individually fine, but taken together they allow unlimited inflation!
- Responsibly disclosed, fix pushed out within a day



# The Future

*“Bitcoin seems to be a very promising idea”*

**Date:** *2020–12-03*

**BTC Price:** *~$19,500*

**Total BTC value:** *~$350,000,000,000*

# The Good News

- Code quality is improving
- Process is improving
- Test coverage is improving
- Review process and qualified reviewers are improving
- Code modularization is improving

# The Challenge

- We find bugs *every month*
- The next one could be “the big one”
- Bitcoin has grown up very quickly
- Improving the culture takes time
- We don’t have enough skilled developers to secure a trillion dollar project

# What can be done?

1. Fund developers!

2. Invest in the development culture - mentoring, coaching, development

3. Scale out - specialization, modularization

# Thank you!

John Newbery

brink