

Verification of (Ethereum) Smart Contracts

Lars Hupel
Blockchain Technology Conference
2019-11-13

INOQ



blockchain
technology
conference

Conclusion

- Smart Contracts are code.

Conclusion

- Smart Contracts are code.
- Code has bugs.

Conclusion

- Smart Contracts are code.
- Code has bugs.
- Smart Contracts have bugs.

Conclusion

- Smart Contracts are code.
- Code has bugs.
- Smart Contracts have bugs.

The end.

Conclusion

- Smart Contracts are code.
- Code has bugs.
- Smart Contracts have bugs.

The end?

**We are barely beginning to
understand smart contracts,
their engineering and their
implications.**



Smart Contracts



Smart Contracts

- digitized contracts with conditions and protocols
- fully automated
- immutable
- decentralized
- without intermediary

Bugs

Classic bug mitigation techniques don't work on (public) blockchains:

9

0800 Antan started
1000 " stopped - antan ✓
13⁰⁰ (033) MP-MC 1.982
(033) PRO 2 2.130
conv 2.130
Relays 6-2 in 033 failed
in Relay
Relays changed
1100 Started Cosine Tape (Si
1525 Started Multi Adder T
1545 Relays (moth)

First actual case of bug
~~1630~~ 1630 Antan started.
1700 closed down.

Bugs

Classic bug mitigation techniques don't work on (public) blockchains:

- updating the program

9

0800 Antan started
1000 " stopped - antan ✓
13⁰⁰ (033) MP-MC 1.982
(033) PRO 2 2.130
conv 2.130
Relays 6-2 in 033 fail
in Relay
Relays changed
1100 Started Cosine Tape (Si
1525 Started Multi Adder T
1545 Relan
(moth)

First actual case of bu
16⁰⁰ Antan started.
1700 closed down.

Bugs

Classic bug mitigation techniques don't work on (public) blockchains:

- updating the program
- rolling back the database

9

0800 Antan started
1000 " stopped - antan ✓
13⁰⁰ (033) MP-MC 1.982
(033) PRO 2 2.130
conv 2.130
Relays 6-2 in 033 fail
in Relay
Relays changed
1100 Started Cosine Tape (Si
1525 Started Multi Adder To
1545 Relays (moth)

First actual case of bug
1615/1630 Antan started.
1700 closed down.

Bugs

Classic bug mitigation techniques don't work on (public) blockchains:

- updating the program
- rolling back the database
- putting the system in lockdown

9

0800 Antan started
1000 " stopped - antan ✓
13⁰⁰ (033) MP-MC 1.982
(033) PRO 2 2.130
conv 2.130
Relays 6-2 in 033 fail
in Relay
Relays changed
1100 Started Cosine Tape (Si
1525 Started Multi Adder To
1545 Relan
(moth)
First actual case of bu
1630 Antan started.
1700 closed down.

Bugs

Classic bug mitigation techniques don't work on (public) blockchains:

- updating the program
- rolling back the database
- putting the system in lockdown
- sending apology letters

9

0800 Antan started
1000 " stopped - antan ✓
13⁰⁰ (033) MP-MC 1.582
(033) PRO 2 2.130
conv 2.130
Relays 6-2 in 033 fail
in Relay
Relays changed
1100 Started Cosine Tape (Si
1525 Started Multi Adder To
1545 Relan
(moth)
First actual case of bu
1615/1630 Antan started.
1700 closed down.

Bugs

Classic bug mitigation techniques don't work on (public) blockchains:

- updating the program
- rolling back the database
- putting the system in lockdown
- sending apology letters
- ...

9

0800 Antan started
1000 " stopped - antan ✓
13⁰⁰ (033) MP-MC 1.582
(033) PRO 2 2.130
conv 2.130
Relays 6-2 in 033 fail
in Relay
Relays changed
1100 Started Cosine Tape (Si
1525 Started Multi Adder To
1545 Relays (moth)

First actual case of bu
1615/1630 Antan started.
1700 closed down.

Bugs happen



The DAO (2016)

- *Decentralized Autonomous Organization*
- kind of a digital VC
- idea: investors vote on proposals to spend money
- reentrancy bug exploited to "steal" \$ 50 M

Bugs happen



Parity Wallet (2017)

- online wallet for Ether and other tokens
- bug exploited to "steal" \$ 30 M

Bugs happen



Parity Wallet (2017)

- online wallet for Ether and other tokens
- bug exploited to "steal" \$ 30 M
- another bug led to complete loss of \$ 360 M

Ethereum's smart contracts are full of holes

Blockchain-powered computer programs promise to revolutionize the digital economy, but new research suggests they're far from secure.

by **Mike Orcutt**

Mar 1, 2018

Computer programs that run on blockchains are shaking up the financial system. But much of the hype around what are called smart contracts is just that. It's a brand-new field. Technologists are just beginning to figure out how to design them so they can be relied on not to lose people's money, and—as a new survey of Ethereum smart contracts illustrates—security researchers are only now coming to terms with what a smart-contract vulnerability even looks like.



Preventing bugs

- bugs may have catastrophic effects
- we can't mitigate bugs

Preventing bugs

- bugs may have catastrophic effects
- we can't mitigate bugs
- we need to prevent bugs from happening

A new problem?

Finance is already strongly dependent on algorithms.

Knight Capital Says Trading Glitch Cost It \$440 Million

BY NATHANIEL POPPER AUGUST 2, 2012 9:07 AM 356

Runaway Trades Spread Turmoil Across Wall St.



Errant trades from the Knight Capital Group began hitting the New York Stock Exchange almost as soon as the opening bell rang on Wednesday. *Brendan McDermid/Reuters*

1 of 4

4:01 p.m. | Updated

\$10 million a minute.

That's about how much the trading problem that set off turmoil on the stock market on Wednesday morning is already costing the trading firm.

The [Knight Capital Group](#) announced on Thursday that it lost \$440 million when it sold all the stocks it accidentally bought Wednesday morning because a computer glitch.

'My savings are missing': technical glitch reduces Barclays customers' cash to zero



One reader's Isa – worth several thousand pounds – was shown as £0 on the Barclays banking app CREDIT: AFP

Follow

By **Dominic Webb**

21 FEBRUARY 2019 • 7:06PM

Barclays' online banking customers are reporting that their savings accounts are being displayed as having a balance of zero.



Software Engineering



Development workflow

Development workflow

1. develop contract with Solidity

Development workflow

1. develop contract with Solidity
2. deploy with Remix/Metamask (or other tools)

Development workflow

1. develop contract with Solidity
2. deploy with Remix/Metamask (or other tools)
3. ???

Testing



Truffle: Framework covering the entire development lifecycle

- testing (Solidity and JavaScript)
- deployment
- migrations

Testing



Truffle: Framework covering the entire development lifecycle

- testing (Solidity and JavaScript)
- deployment
- migrations

The end.

Testing



Truffle: Framework covering the entire development lifecycle

- testing (Solidity and JavaScript)
- deployment
- migrations

The end?

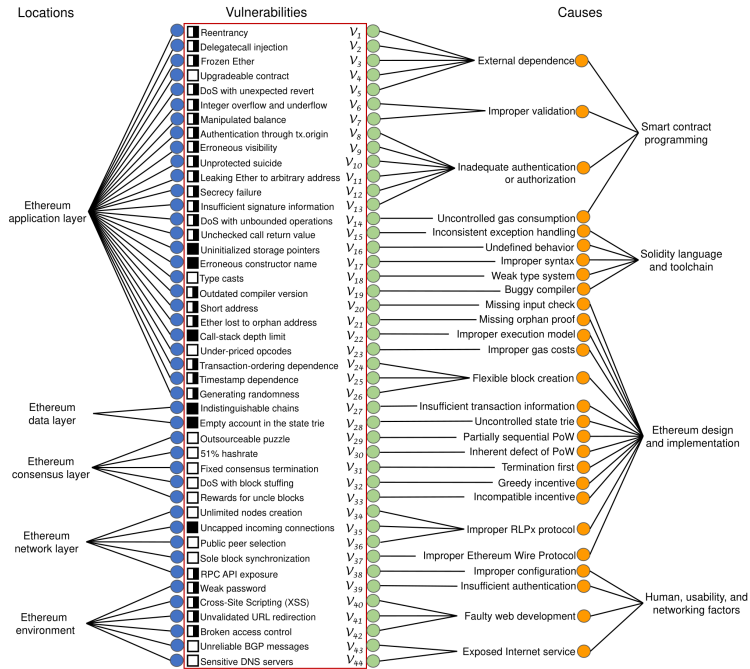
Testing is not enough

“ Program testing can be a very effective way to show the presence of bugs, but is hopelessly inadequate for showing their absence. ”

Edsger W. Dijkstra

**How can we test for
vulnerabilities?**

**What even are our
vulnerabilities?**



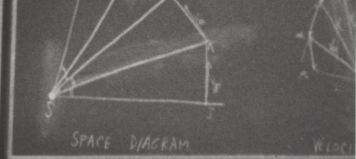
Vulnerability causes

Cause	total	eliminated	avoidable	open
Programming	14	0	13	1
Solidity	5	2	2	1
Ethereum	18	4	5	9
Human/usability	7	0	5	2
total	44	6	25	13

Huashan Chen, Marcus Pendleton, Laurent Njilla, and Shouhuai Xu: "A Survey on Ethereum Systems Security: Vulnerabilities, Attacks and Defenses" (2019)

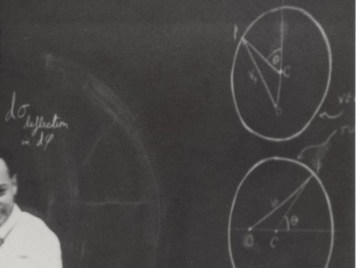
$= FG'$
 $FQ + F'Q = FQ + G'Q$
 $> FG'$
 $\therefore Q$ lies outside
 (on or at) the ellipse

$\alpha = \text{Area swept/sec}$
 $R^2 \Delta\theta / \alpha = \Delta t$
 $\Delta V = \frac{3}{R^2} (\Delta t) = \frac{3}{R^2} R^2 \frac{\Delta\theta}{\alpha} = \frac{3}{\alpha} \Delta\theta$
 $= V_R \Delta\theta$



(a) + (b), then (c)
 ce toward Sun
 Force as $1/r^2$

$V_R = \alpha$
 $b = \frac{3}{V_R \tan \frac{\phi}{2}}$
 $\tan \frac{\phi}{2} = \frac{V_R}{V_\infty} = \frac{3}{\alpha V_\infty} = \frac{3}{b V_\infty}$
 x-sect area for deflection $> \phi$
 $\propto \pi b^2$ or πz^2



Formal Methods

Formal Methods?

“ *'Formal Methods' refers to mathematically rigorous techniques and tools for the specification, design and verification of software and hardware systems.* ”

César Muñoz (NASA)

Code contracts

```
@Requires("x >= 0")  
@Ensures("result >= 0")  
static double sqrt(double x);
```

What is verification?

Verification consists of three components:

1. formal language for specification
2. formal semantics for implementation
3. static analysis

subsection "Preservation of semantics"

lemma ccomp_bigstep:

"(c,s) $\Rightarrow$ t $\implies$ ccomp c $\vdash$ (0,s,stk) $\rightarrow^*$ (size(ccomp c),t,stk)"

proof(induction arbitrary: stk rule: big_step_induct)

case (Assign x a s)

show ?case by (fastforce simp:fun_upd_def cong: if_cong)

next

case (Seq c1 s1 s2 c2 s3)

let ?cc1 = "ccomp c1" let ?cc2 = "ccomp c2"

have "?cc1 @ ?cc2 $\vdash$ (0,s1,stk) $\rightarrow^*$ (size ?cc1,s2,stk)"

using Seq.IH(1) by fastforce

moreover

have "?cc1 @ ?cc2 $\vdash$ (size ?cc1,s2,stk) $\rightarrow^*$ (size(?cc1 @ ?cc2),s3,stk)"

using Seq.IH(2) by fastforce

ultimately show ?case by simp (blast intro: star_trans)

next

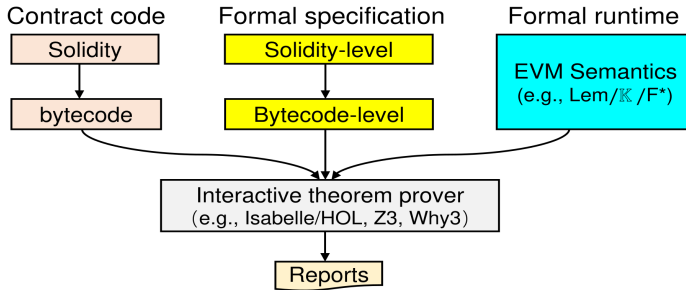
case (WhileTrue b s1 c s2 s3)

let ?cc = "ccomp c"

let ?cb = "bcomp b False (size ?cc + 1)"

let ?cw = "ccomp(WHILE b DO c)"

Verification in Ethereum



Huashan Chen, Marcus Pendleton, Laurent Njilla, and Shouhuai Xu: "A Survey on Ethereum Systems Security: Vulnerabilities, Attacks and Defenses" (2019)

Full stack verification

Component	Availability
Specification of EVM	✓
Specification of Solidity	?
Verified Solidity compiler	✗
Verified EVM runtime	✓
Solidity VCG	✗
Formal business case	✗

Verification works

“The HackerGold (HKG) token is an ERC20 token written in Solidity [...] We found that the [...] implementation does not conform to the ERC20 standard.”

Formally Verified Smart Contracts by Runtime Verification (2017–2019)

Verification works

“The Vyper ERC20 token is shipped as part of the official Vyper language compiler distribution.

Vyper, currently being developed by the Ethereum Foundation, is an experimental smart contract language with syntax similar to Python, designed with the goal of being a simpler and more secure alternative to Solidity.

The Vyper ERC20 token was successfully verified against the ERC20-EVM specification, implying its full conformance to the ERC20 standard.

”

Example

```
Theorem can_claim_back id b d st bc:
  (* (a) The backer b has donated d, so the contract holds
      that record in its state *)
  donated b d st →
  (* (b) The campaign has not been funded. *)
  ¬ funded (state st) →
  (* (c) Balance is small: not reached the goal. *)
  balance st < (get_goal (state st)) →
  (* (d) Block number exceeds the deadline. *)
  get_max_block (state st) < block_num bc →
  (* (conclusion) Backer b can get their donation back. *)
  ∃ (m : message),
    sender m == b ∧
    out (step_prot c st bc m) = Some (Msg d id b 0 ok_msg).
```

Figure 5. A backer can claim back her funds if the campaign fails.

Research

- *Workshop on Trusted Smart Contracts (2017–)*
- *Workshop Formal Methods for Blockchains (2019–)*



Legal Force & Effect

Code is Law

“ However, these more complex agreements, with greater engagement with real world goods and services, highlight the necessity of effective dispute resolution, as well as indicate a necessary interrelation with territorial legal systems.

Code is Law

“ However, these more complex agreements, with greater engagement with real world goods and services, highlight the necessity of effective dispute resolution, as well as indicate a necessary interrelation with territorial legal systems.

Contestation mechanisms are necessary to 'soften' the effects of self-executing 'smart contracts', and make transactions reversible, allowing the outcomes of dispute resolution to be enforced.

”

Jake Goldenfein, Andrea Leiter: "Legal Engineering on the Blockchain: 'Smart Contracts' as Legal Conduct" (2018)

Legalese

- "Legalese" is (English) writing that is inscrutable to laypeople
- Are smart contracts an improvement?
- Does verification help?

Smart Contracts?

“ *There are only two things you need to know about smart contracts:*

- 1. They're not smart.*
- 2. They're not contracts.*

”

Philip Wadler



Q & A

INNOQ
www.innoq.com

Lars Hupel

 lars.hupel@innoq.com

 [@larsr_h](https://twitter.com/larsr_h)

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim a. Rh.
Germany
+49 2173 3366-0

Ohlauer Str. 43
10999 Berlin
Germany

Ludwigstr. 180 E
63067 Offenbach
Germany

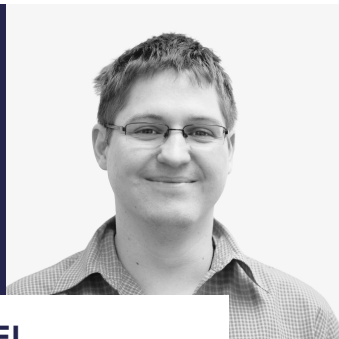
Kreuzstr. 16
80331 München
Germany

c/o WeWork
Hermannstrasse 13
20095 Hamburg
Germany

innoQ Schweiz GmbH

Gewerbestr. 11
CH-6330 Cham
Switzerland
+41 41 743 01 11

Albulastr. 55
8048 Zürich
Switzerland



LARS HUPEL

Consultant
innoQ Deutschland GmbH

Lars enjoys programming in a variety of languages, including Scala, Haskell, and Rust. He is known as a frequent conference speaker and one of the founders of the Typelevel initiative which is dedicated to providing principled, type-driven Scala libraries.

Die Blockchain

Eine Kette der Möglichkeiten



Sources

- Treaty of Rome: https://en.wikipedia.org/wiki/File:Treaty_of_Rome.jpg, author unknown
- Lightning: <https://pixabay.com/photos/lightning-storm-weather-sky-399853/>
- Lady Justice: <https://pixabay.com/photos/justice-statue-lady-justice-2060093/>
- Feynman with blackboard: [https://commons.wikimedia.org/wiki/File:HD.3A.053_\(10481714045\).jpg](https://commons.wikimedia.org/wiki/File:HD.3A.053_(10481714045).jpg)
- Interchange: <https://unsplash.com/photos/nDfEFYiGrAY>
- Mike Orcutt: "Ethereum's smart contracts are full of holes", <https://www.technologyreview.com/s/610392/ethereums-smart-contracts-are-full-of-holes/>
- Huashan Chen, Marcus Pendleton, Laurent Njilla, and Shouhuai Xu: "A Survey on Ethereum Systems Security: Vulnerabilities, Attacks and Defenses", <https://arxiv.org/pdf/1908.04507.pdf>
- Jake Goldenfein, Andrea Leiter: "Legal Engineering on the Blockchain: 'Smart Contracts' as Legal Conduct", https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3176363
- Nathaniel Popper: "Knight Capital Says Trading Glitch Cost It \$440 Million", <https://dealbook.nytimes.com/2012/08/02/knight-capital-says-trading-mishap-cost-it-440-million/> (2012)
- Domic Webb: "'My savings are missing': technical glitch reduces Barclays customers' cash to zero", <https://www.telegraph.co.uk/personal-banking/savings/savings-missing-technical-glitch-reduces-barclays-customers/> (2019)
- Ilya Sergey, Amrit Kumar, Aquinas Hobor: "Scilla: a Smart Contract Intermediate-Level Language", <https://arxiv.org/pdf/1801.00687.pdf>
- Solidity Semantics: <https://github.com/kframework/solidity-semantics>