WIKIPEDIA

Solidity

Solidity is an object-oriented programming language for implementing smart contracts[2][3] on various blockchain platforms, most notably, Ethereum.[4] It was developed by Christian Reitwiessner, Alex Beregszaszi, and several former Ethereum core contributors.[5] Programs in Solidity run on Ethereum Virtual Machine.

Solidity

The Solidity language logo

Paradigm	Object-orientated
Stable release	0.4.21[1] / 8 March 2018
Website	github.com/ethereum/solidity (https://github.com/ethereum/solidity)
Influenced by	
JavaScript, C++, Python	

Contents

History

Solidity was proposed in August 2014 by Gavin Wood;[6] the language was later developed by the Ethereum project's Solidity team, led by Christian Reitwiessner.

Solidity is the primary language on Ethereum[7] as well as on other private blockchains on platforms that compete with Ethereum, such as Monax and its Hyperledger Burrow blockchain, which uses Tendermint for consensus. SWIFT deployed a proof of concept using Solidity running on Burrow.[3][8]

Description

Solidity is a statically typed programming language designed for developing smart contracts that run on the Ethereum Virtual Machine (EVM).[9]

Solidity uses ECMAScript-like syntax which makes it familiar for existing web developers; however unlike ECMAScript it has static typing and variadic return types. Solidity is different from other EVM-targeting languages such as Serpent and Mutan in some important ways. It supports complex member variables for contracts, including arbitrarily hierarchical mappings and structs. Solidity contracts support inheritance, including multiple inheritance with C3 linearization. Solidity introduces an application binary interface (ABI) that facilitates multiple type-safe functions within a single contract (this was also later supported by Serpent). The Solidity proposal also includes "Natural Language Specification", a documentation system for specifying user-centric descriptions of the ramifications of method-calls.[10][11]

Example of a Solidity program:[12][13]

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity ^0.8.4;

contract Coin {
    // The keyword "public" makes variables
    // accessible from other contracts
    address public minter;
    mapping (address => uint) public balances;

    // Events allow clients to react to specific
    // contract changes you declare
    event Sent(address from, address to, uint amount);

    // Constructor code is only run when the contract
    // is created
    constructor() {
        minter = msg.sender;
    }

    // Sends an amount of newly created coins to an address
    // Can only be called by the contract creator
    function mint(address receiver, uint amount) public {
        require(msg.sender == minter);
        balances[receiver] += amount;
    }

    // Errors allow you to provide information about
    // why an operation failed. They are returned
    // to the caller of the function.
    error InsufficientBalance(uint requested, uint available);

    // Sends an amount of existing coins
    // from any caller to an address
    function send(address receiver, uint amount) public {
        if (amount > balances[msg.sender])
            revert InsufficientBalance({
                requested: amount,
                available: balances[msg.sender]
            });

        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        emit Sent(msg.sender, receiver, amount);
    }
}
```

Development platform availability

- ErisDB by AWS
- Hardhat
- Microsoft Visual Studio[14]
- Microsoft Visual Studio Code[15]
- Tendermint on Microsoft Azure
- Remix by Ethereum[16]

Blockchain platforms

Solidity is available on:

- Ethereum

- Binance Smart Chain[17]
- Ethereum Classic
- Avalanche C-Chain
- Counterparty (which runs on Bitcoin)[18][19]
- Tron
- Hedera Hashgraph

Criticism

Many security properties of smart contracts are inherently difficult to reason about directly, and the Turing-completeness of Solidity means that verification of arbitrary properties cannot be decidably automated. Current automated solutions for smart contract security analysis can miss critical violations, produce false positives, and fail to achieve sufficient code coverage on realistic contracts.[20] Solidity has been blamed for the error-prone implementation of Ethereum smart contracts due to its counterintuitive nature, its lack of constructs to deal with blockchain domain-specific aspects, and its lack of centralized documentation of known vulnerabilities.[21]

In 2016, a Cornell University researcher stated that Solidity was partially to blame for The DAO hack that took place that year. He stated: "this was actually not a flaw or exploit in the DAO contract itself: technically the Ethereum Virtual Machine (EVM) was operating as intended, but Solidity was introducing security flaws into contracts that were not only missed by the community, but missed by the designers of the language themselves."[22]

Limitations of Solidity

Unlike programs in traditional programming languages, which can be debugged, in Solidity contracts mistakes cannot be edited or fixed; transactions cannot be reversed. Solidity follows the "Code is Law" mantra, which means any smart contract must be flawlessly coded when it comes into effect.

There have been some hacking cases such as the aforementioned 2016 DAO hack in which US$60 million was stolen, and a 2021 hack that caused a fork in the Ethereum system.

To prevent technical errors and mistakes, Coinbase, the largest cryptocurrency exchange in the US, introduced a new tool named Solidify. This tool is an AI auditing system that detects and classifies smart contract risks.

References

1. "Release 0.4.21" (https://github.com/ethereum/solidity/releases/tag/v0.4.21). 8 March 2018. Retrieved 15 March 2018.
2. Afshar, Vala; Evangelist, ContributorChief Digital; Salesforce (17 July 2017). "Ethereum Is The Second Most Valuable Digital Currency, Behind Bitcoin" (https://www.huffpost.com/entry/ethereum-is-the-second-most-valuable-digital-currency_b_596bc5c7e4b022bb9372b2b2). *HuffPost*. Retrieved 10 April 2019. {{cite web}}: |first2= has generic name (help)
3. "SOFE Berlin: Swift unveils blockchain proof-of-concept" (https://www.finextra.com/newsarticle/29813/sofe-berlin-swift-unveils-blockchain-proof-of-concept). *Finextra* (News). 24 November 2016. Retrieved 24 November 2016.

4. Finley, Klint. "Someone Just Stole $50 Million from the Biggest Crowdfunded Project Ever. (Humans Can't Be Trusted)" (https://www.wired.com/2016/06/50-million-hack-just-showed-dao-human/). *Wired*.
5. "List of contributors" (https://github.com/ethereum/solidity/graphs/contributors). *GitHub*.
6. Benoit Schweblin. "StackEdit Viewer" (https://stackedit.io/viewer#!url=https://gist.githubusercontent.com/gavofyork/31b35cd2252a00d0d057/raw/16de06189d2175d2e31b300f1f8531e20c927635/solidity-original). *stackedit.io*.
7. Nikolic, Ivica; Kolluri, Aashish; Sergey, Ilya; Saxena, Prateek; Hobor, Aquinas (14 March 2018). "Finding The Greedy, Prodigal, and Suicidal Contracts at Scale". arXiv:1802.06038 (https://arxiv.org/abs/1802.06038) [cs.CR (https://arxiv.org/archive/cs.CR)]. "Different source languages compile to the EVM semantics, the predominant of them being Solidity"
8. KENTOURIS, CHRIS (13 December 2016). "Blockchain's Smart Contracts: What's Smart, What's Not" (http://finops.co/operations/blockchains-smart-contracts-what-smart-whats-not/). *Finops* (News). Retrieved 14 December 2016.
9. "Hyperledger Fabric Tutorial - Create a blockchain app for loyalty points" (https://developer.ibm.com/patterns/loyalty-points-fabric-evm/). *IBM Developer*. Retrieved 10 April 2019.
10. Kapetanios-2008-06-27, p. 309.
11. ethereum. "Ethereum Natural Specification Format" (https://github.com/ethereum/wiki/wiki/Ethereum-Natural-Specification-Format). *GitHub*.
12. "Subcurrency Example from the Solidity documentation" (https://solidity.readthedocs.io/en/v0.5.14/introduction-to-smart-contracts.html#subcurrency-example).
13. Schneier, Karthikeyan; Schneier, Antoine; Bhargavan, Cedric; Delignat-Lavaud, Anitha; Fournet, Gollamudi; Schneier, Bruce; Rastogi, Nadim; Sibut-Pinote, Aseem; Rastogi1, Thomas; Swamy, Nikhil; Zanella-Beguelin, Santiago (27 August 2016). "Short Paper: Formal Verification of Smart Contracts" (http://research.microsoft.com/en-us/um/people/nswamy/papers/solidether.pdf) (PDF). *Microsoft Research, French Institute for Research in Computer Science and Automation, Harvard University*. Archived (https://web.archive.org/web/20160827092146/http://research.microsoft.com/en-us/um/people/nswamy/papers/solidether.pdf) (PDF) from the original on 27 August 2016.
14. Teeter, Cale (1 April 2016). "Solidity Integration with Visual Studio" (https://medium.com/@ConsenSys/solidity-integration-with-visual-studio-8bdab2ff8a74). *Medium*. Archived (https://web.archive.org/web/20161127081428/https://medium.com/@ConsenSys/solidity-integration-with-visual-studio-8bdab2ff8a74) from the original on 27 November 2016. Retrieved 10 June 2021.
15. PatAltimore. "Use Visual Studio Code to connect to Azure Blockchain Service - Azure Blockchain" (https://docs.microsoft.com/en-us/azure/blockchain/service/connect-vscode). *docs.microsoft.com*. Retrieved 27 March 2020.
16. "Welcome to Remix's documentation!" (https://remix-ide.readthedocs.io/en/latest/index.html). Retrieved 16 June 2022.
17. "Binance Smart Chain" (https://github.com/binance-chain/bsc). *GitHub*. 26 October 2021.
18. Vigna, Michael J. Casey and Paul (12 November 2014). "BitBeat: Bitcoin 2.0 Firm Counterparty Adopts Ethereum's Software" (https://blogs.wsj.com/moneybeat/2014/11/12/bitbeat-bitcoin-2-0-firm-counterparty-adopts-ethereums-software/). *Wall Street Journal*. ISSN 0099-9660 (https://www.worldcat.org/issn/0099-9660). Retrieved 16 April 2021.
19. Swan, Melanie (2015). *Blockchain : blueprint for a new economy* (https://www.worldcat.org/oclc/900781291) (1st. ed.). [Sebastopol, Calif.] ISBN 978-1-4919-2047-3. OCLC 900781291 (https://www.worldcat.org/oclc/900781291).

20. Tsankov, Petar; Dan, Andrei; Drachsler-Cohen, Dana; Gervais, Arthur; Bünzli, Florian; Vechev, Martin (15 October 2018). "Securify: Practical Security Analysis of Smart Contracts" (https://arxiv.org/pdf/1806.01143.pdf) (PDF). *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery: 67–82. arXiv:1806.01143 (https://arxiv.org/abs/1806.01143). doi:10.1145/3243734.3243780 (https://doi.org/10.1145%2F3243734.3243780). hdl:10044/1/87935 (https://hdl.handle.net/10044%2F1%2F87935). S2CID 46936025 (https://api.semanticscholar.org/CorpusID:46936025).
21. Atzei, Nicola; Bartoletti, M.; Cimoli, Tiziana (2017). "A Survey of Attacks on Ethereum Smart Contracts (SoK)" (https://www.semanticscholar.org/paper/A-Survey-of-Attacks-on-Ethereum-Smart-Contracts-Atzei-Bartoletti/aec843c0f38aff6c7901391a75ec10114a3d60f8). *POST*. Lecture Notes in Computer Science. **10204**: 164–186. doi:10.1007/978-3-662-54455-6_8 (https://doi.org/10.1007%2F978-3-662-54455-6_8). ISBN 978-3-662-54454-9. S2CID 15494854 (https://api.semanticscholar.org/CorpusID:15494854).
22. Finley, Klint (18 June 2016). "A $50 Million Hack Just Showed That the DAO Was All Too Human" (https://www.wired.com/2016/06/50-million-hack-just-showed-dao-human/). *Wired* (News). Retrieved 18 February 2017.
