Analysis of the Solidity Compiler for Smart Contract Redundancy Detection

Jonas Gebele, June 22, 2020, Kick-Off Presentation

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

Outline

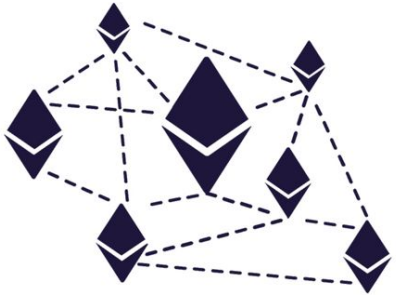
1. Motivation
2. Background Information
3. Problem Statement and Goal of Thesis
4. Research Questions
5. Timeline

Motivation

EVM Bytecode Analysis for ...

Security Analysis

- Bug Hunting
- Vulnerability Research
- Security Audits



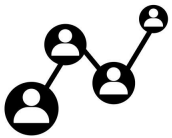
June 2019 - **14,5 million contracts** created and 60 million accounts stored

Usage Analytics

- Analysis of smart contract interactions
- Gas-Cost inspection

Quantitative Analysis

- Transaction tracking

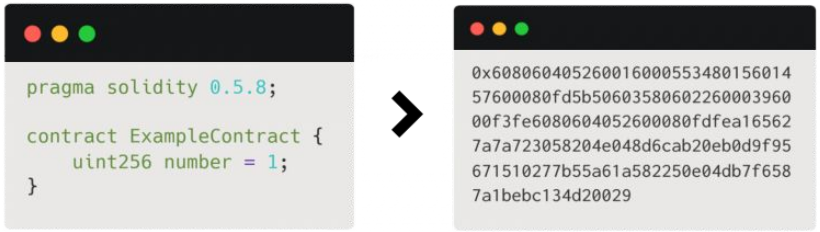


Outline

1. Motivation
2. Background Information
3. Problem Statement and Goal of Thesis
4. Research Questions
5. Timeline

Background Information

Solidity Code
(sourceFile.sol)



EVM (Deployment)
Bytecode

solc - Solidity Compiler

```
$ solc --optimize --bin sourceFile.sol
```

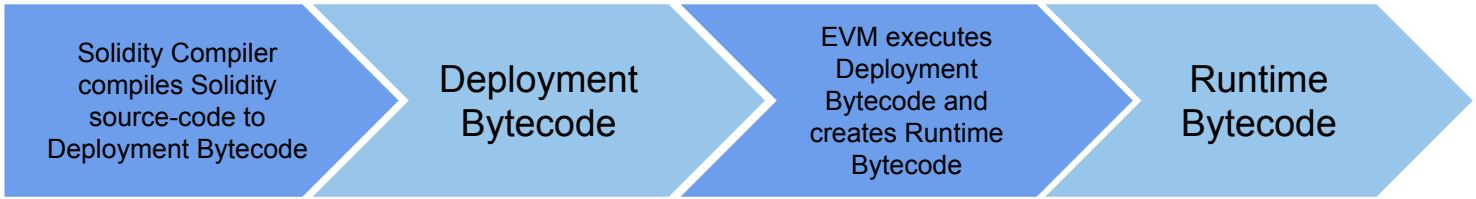
Background Information

Contract creation transaction

```
> src =  web3.eth.accounts[0];
> ourContractDeploymentBytecode = "0x6080604052600160005534801560014…"
```

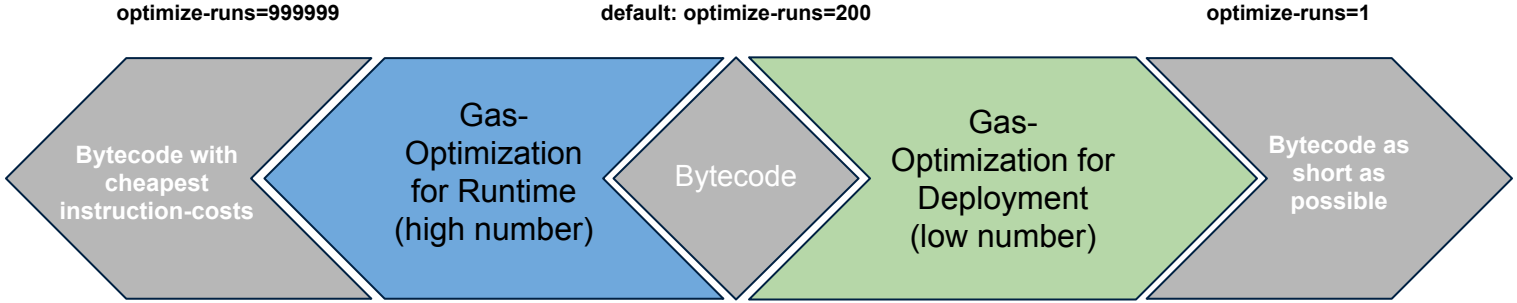
```
> web3.eth.sendTransaction ({
      from: src,
      data: ourContractDeploymentBytecode,
      gas: 113558,
      gasPrice: 200000000000
})
```

Deployment workflow of a smart contract



Background Information


```
$ solc --optimize-runs=200  --bin sourceFile.solv
```



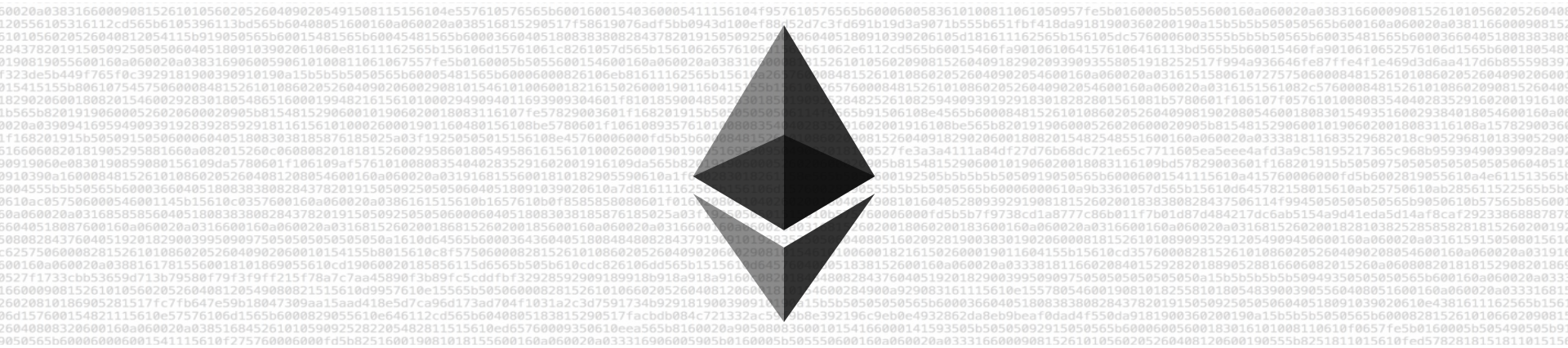
Outline

1. Motivation
2. Background Information
3. Problem Statement and Goal of Thesis
4. Research Questions
5. Timeline

Problem Statement and Goal of Thesis

Many studies in bytecode-analysis work with sets of unique smart contracts

Missing inclusion of the optimization process of the compiler



How many EVM bytecodes are redundant due to different or missing optimization?

Outline

1. Motivation
2. Background Information
3. Problem Statement and Goal of Thesis
4. Research Questions
5. Timeline

Research Questions

R1: How does the bytecode-optimizer work in detail?

R2: Which metrics and properties should be used to determine the uniqueness of smart contracts?

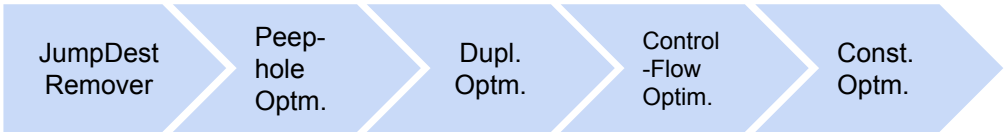
R3: How many bytecodes and therefore smart contracts are redundant regarding their functionality due to different or missing optimization?

How does the bytecode-optimizer work in detail?

Research question 1

1.1 Internals of the bytecode-optimizer

- Of which internals does the bytecode-optimizers consist and what do they do?



1.2 Optimization in the compilation-process

- Which optimization-steps are done per default?
- In which compilation-context is the optimization done?

1.3 How do the compilation parameters affect the optimization-process?

1.4 Which sections of the bytecode get optimized?

Which metrics and properties should be used to determine the uniqueness of smart contracts?

Research question 2

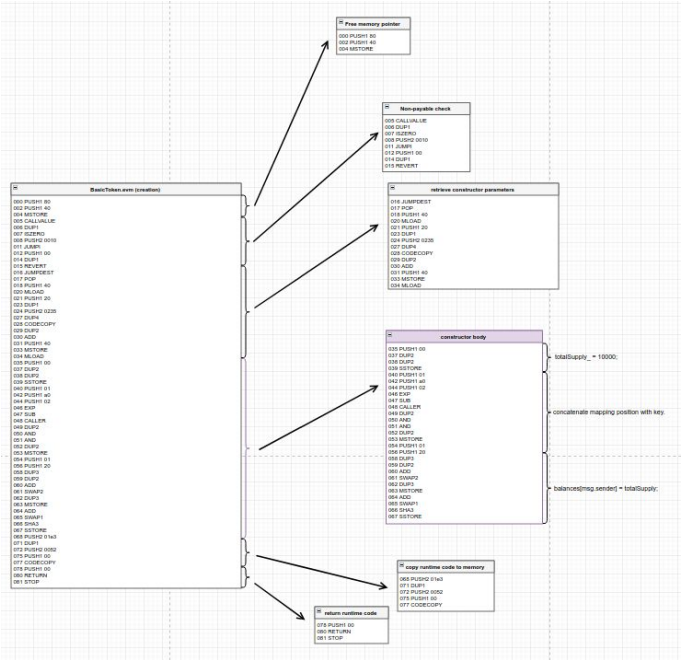
2.1 Generic structure of the deployment bytecode

- How to extract the function-signatures from the function selectors?
- How to separate the function-wrappers with with function bodies?

2.2 Useful metrics to compare bytecodes

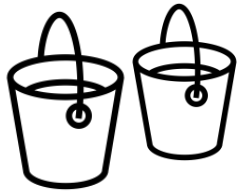
- String-comparison algorithms like the levenshtein-distance, opcode-metrics, function signatures, control-flow graphs…?

2.3 When are bytecodes considered to originate from the same source-code?



How many bytecodes and therefore smart-contracts are redundant regarding their functionality due to different or missing optimization?

Research question 3



3.1 What are processes to optimize the comparison of bytecodes and to make it more efficient?

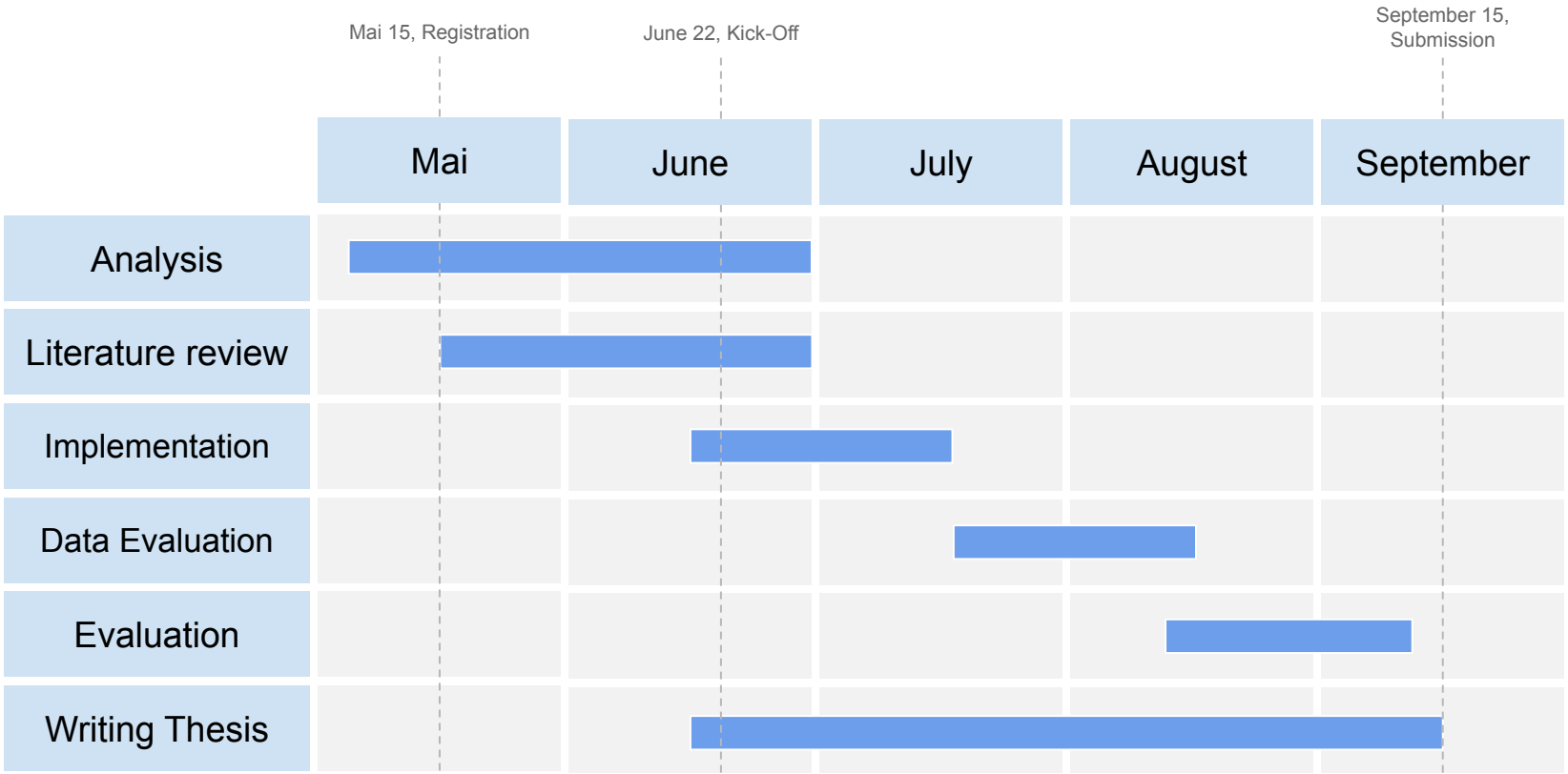
3.2 What are the results of the analysis on how many smart contracts are redundant due to different or missing optimization?

- How do the results differ from those of other studies?

Outline

1. Motivation
2. Background Information
3. Problem Statement and Goal of Thesis
4. Research Questions
5. Timeline

Timeline


Mai 15, Registration
June 22, Kick-Off
September 15, Submission
Mai
June
July
August
September
Analysis
Literature review
Implementation
Data Evaluation
Evaluation
Writing Thesis

Jonas Gebele

jonas.gebele@in.tum.de

Technische Universität München
Faculty of Informatics
Chair of Software Engineering for
Business Information Systems

Boltzmannstraße 3
85748 Garching bei München