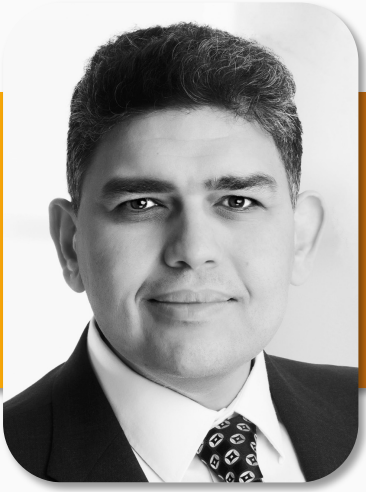
A Large-Scale Exploratory Study on the Proxy Design Pattern in Ethereum Blockchain

Amir M. Ebrahimi

Bram Adams

Gustavo A. Oliva

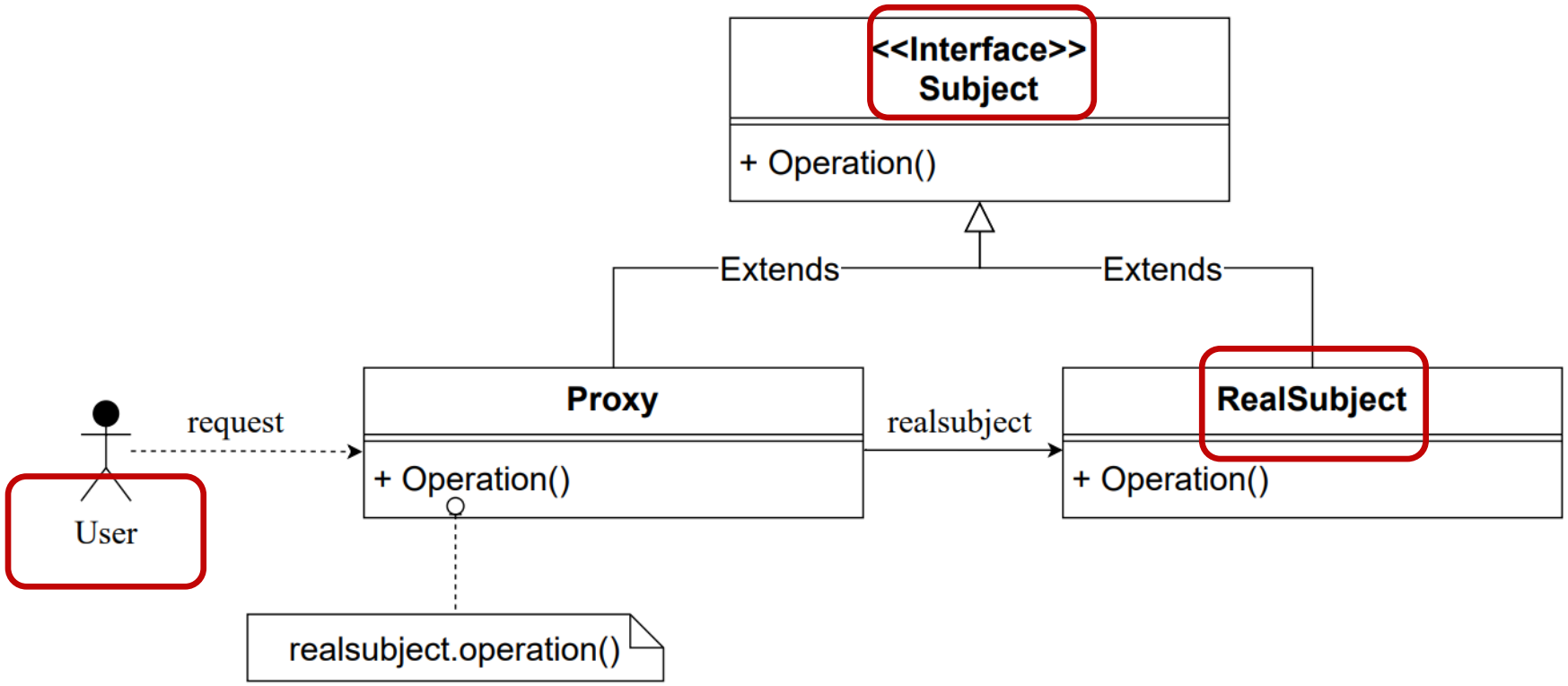


Ahmed E. Hassan

Queen's University

Software Analysis & Intelligence Lab

The Proxy Design Pattern: A cornerstone of conventional software design



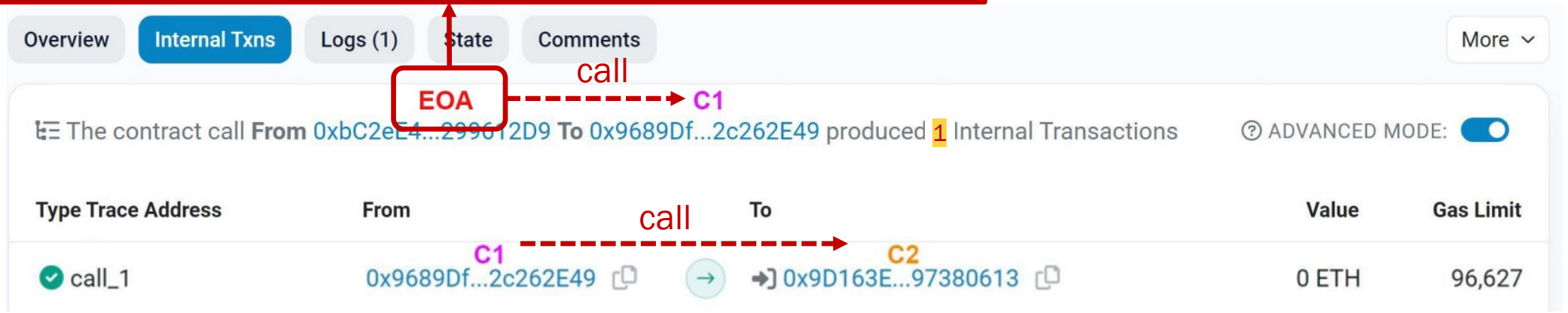
The GoF Proxy Design Pattern

Programable blockchains offer unique features to application development

- Ethereum introduced **smart contracts**, a software that runs on Ethereum blockchain
- **Transactions** is the way to interact with a contract

01. Decentralization
02. Tamper-proof transactions
03. Traceability
04. Transparency
05. Security

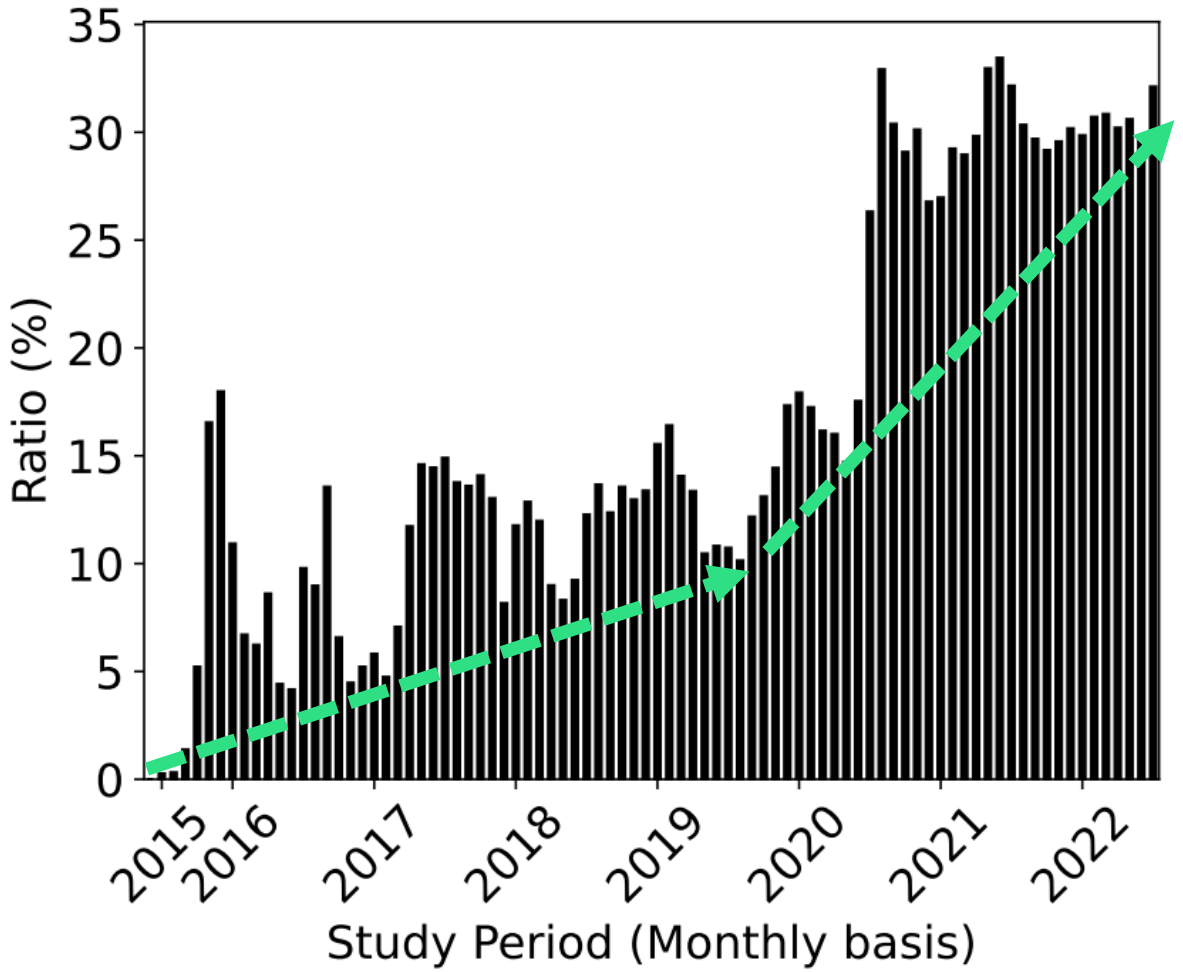
An Externally owned account represents either a user or developer



A sample transaction

Smart contracts are not monolithic applications

1. How prevalent are proxy contracts?
2. How are proxy contracts integrated into applications?



The monthly ratio of multi-contract transactions

Proxy contracts: a dual-blade, enabling maintenance while fracturing blockchain immutability.

- How do developers upgrade smart contracts despite **immutability**?
  - Solution: Upgradeability proxy contracts

1. How prevalent are different types of proxies?
2. Tracking proxies is critical for security reasons
   - Lack of techniques that effectively detect proxies **accurately**, at **scale** and in a **timely manner**

Dataset & research questions

Dataset	Ethereum's dataset [Aug. 2015 to Sep. 2022] **50M** smart contracts **1.6B** transactions	
Research Questions		RQ1: How prevalent is the proxy mechanism in the Ethereum ecosystem?
		RQ2: What are different creational patterns for deploying proxy contracts?
		RQ3: What are the different types and properties of proxy contracts?

Proxy Detection method: Our method efficiently identifies all proxy contracts in **under 15 minutes**

- A proxy has two signatures:
  i. It shall use the delegatecall
  ii. The proxy contract shall have a similar interface to the actual serving contract's ones.

- Our method matches the **behavior** of over 50M smart contracts against the two proxy signatures

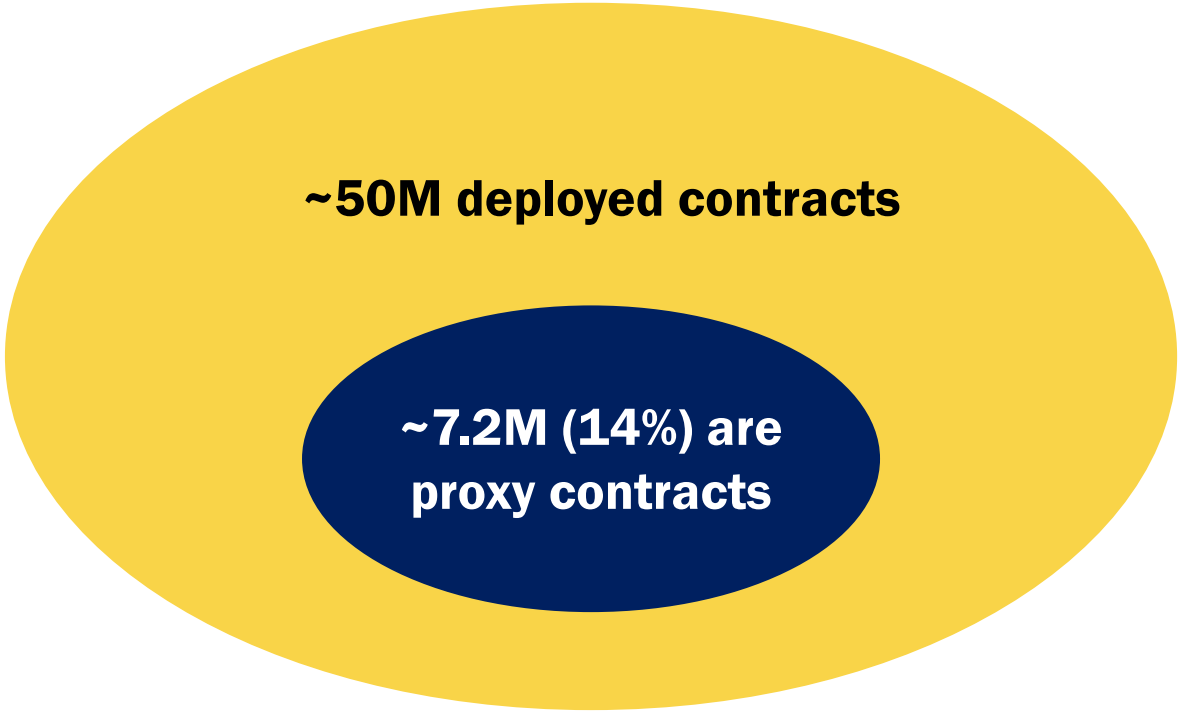
Evaluation: Our method exhibits **perfect** precision and recall and **outperformed** a previous study

- Ground truth dataset (385 randomly labeled contracts)
  - Achieve 100% precision and recall
- Compared our performance with Salehi et. al's work.
  - We detected 300K more proxy contracts

RQ1: How prevalent is the proxy mechanism in the Ethereum ecosystem?

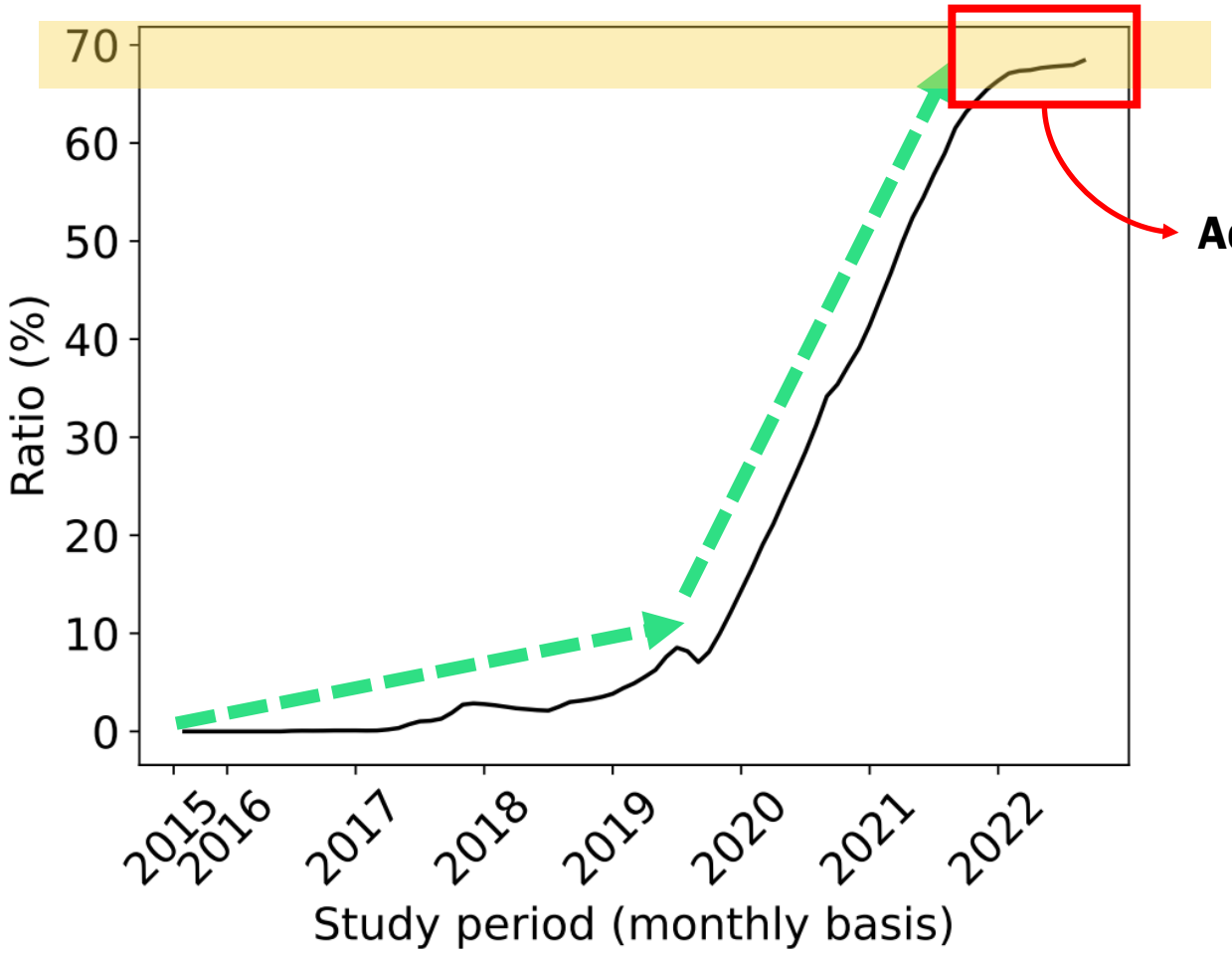
- Prevalence metrics
  1. General ratio of proxy contracts
  2. Stakeholder adoption
  3. Smart contracts design
  4. Usage context

General ratio: 14% of all deployed contracts are proxies.



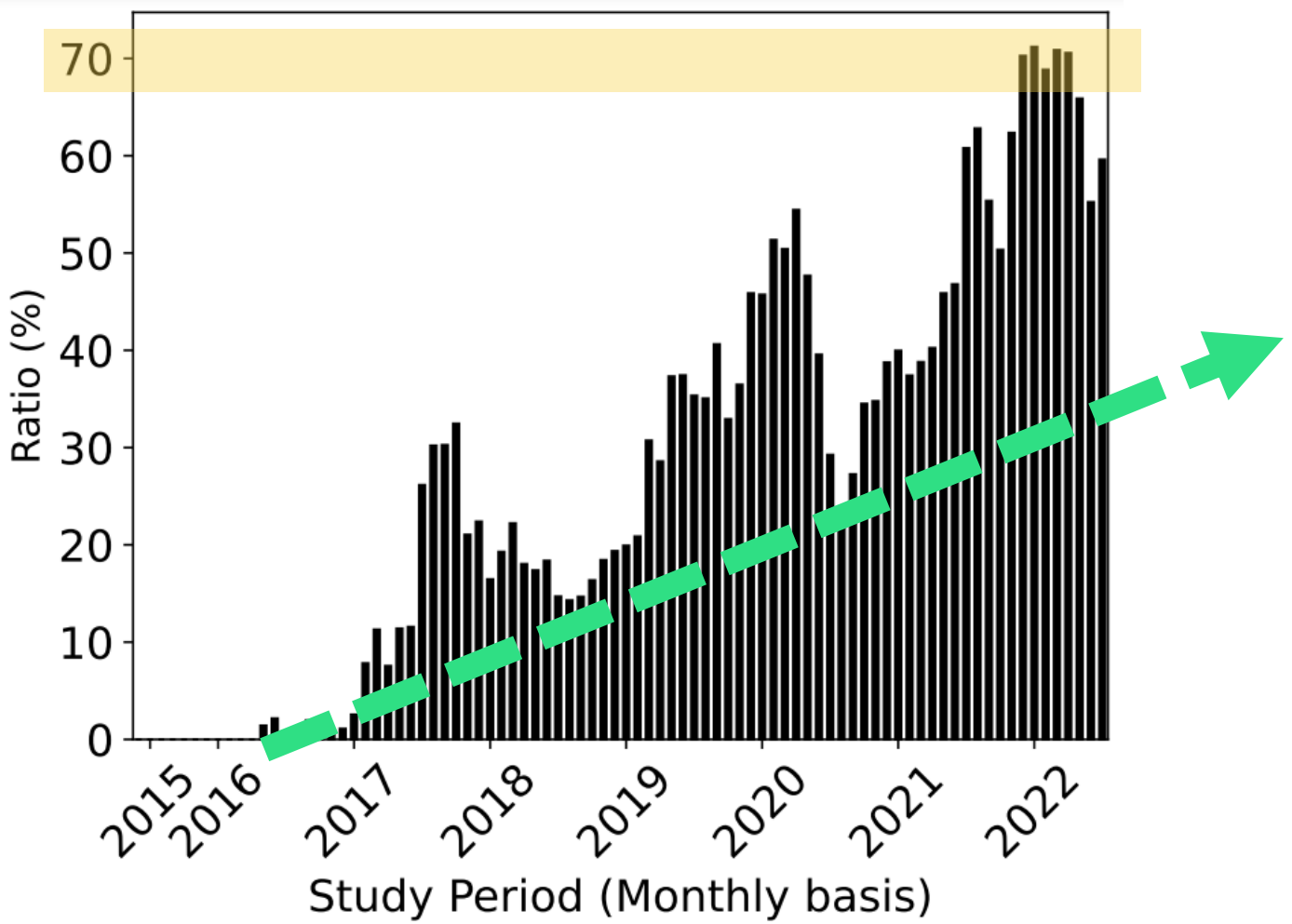
Proxy contracts share of all deployed contracts

Stakeholder adoption: Over **two-thirds** of all EOAs who deployed a contract, at least deployed one proxy contract too



The monthly cumulative ratio of EOAs who deployed a proxy contract

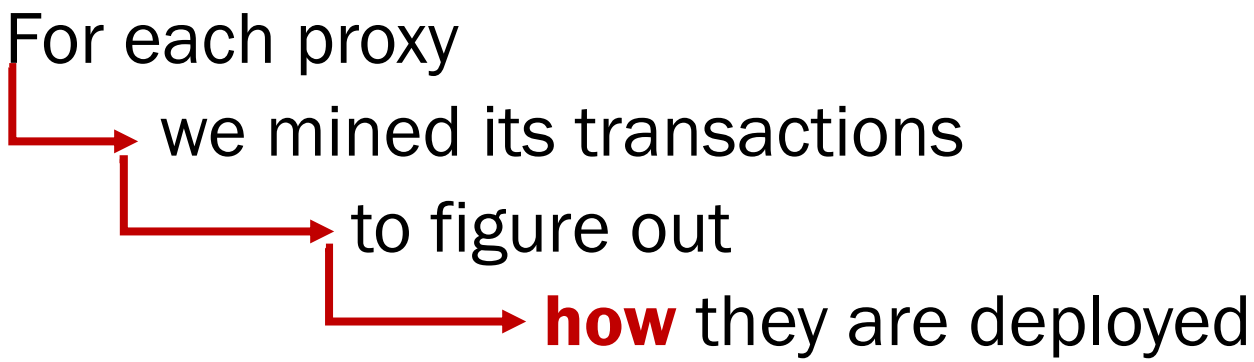
Design: Proxy contracts are increasingly being used in the design of modular applications.



The monthly ratio of multi-contract transactions that involve at least one proxy contract.

RQ2: What are the different creational patterns for deploying proxy contracts?

Method



We found 12 creational patterns for deploying proxy contracts.

Off-chain is the most popular deployment style among practitioners.

Id	Creational Pattern	Deployment style	Prac. count	Proxy instance count
1	EOA > P	Off-chain	20,210 (94.66%)	50,174 (0.69%)
2	EOA > FA > P	On-chain	1,385 (6.49%)	6,618,012 (91.39%)

Most proxy contracts are created using an on-chain style

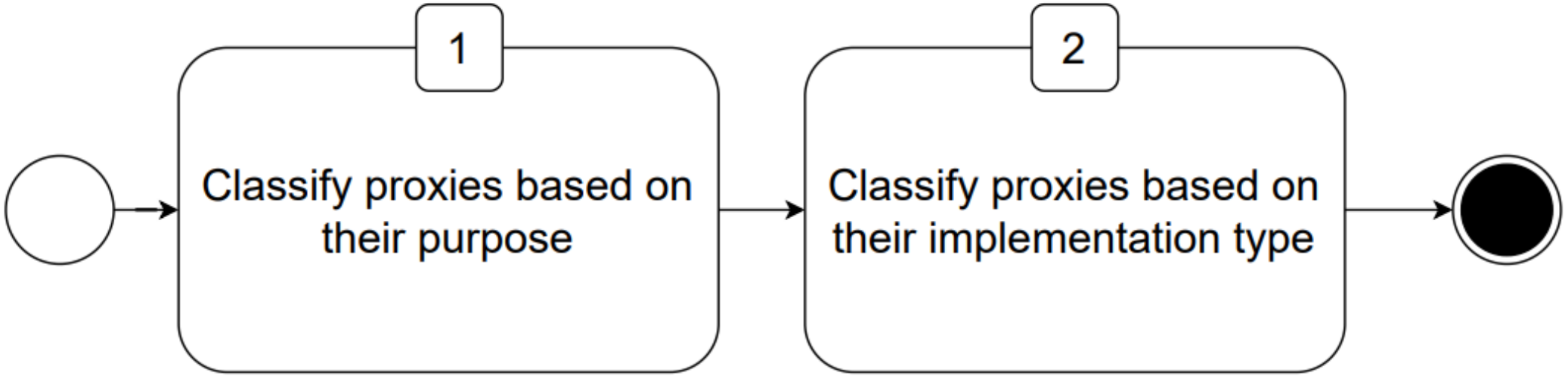
Item	Stands for
>	Deployment operator
EOA	Practitioner/Developer
P	Proxy smart contract
FA	Factory smart contract

Reference for reading creational patterns

On-chain vs Off-chain deployment styles

Off-chain	On-chain
Smart contracts is deployed on blockchain	Smart contracts is deployed on blockchain
Deployment Infrastructure operates outside of blockchain	Deployment Infrastructure operates on blockchain
Deployment scripts are written in Web3, JavaScript	A smart contract instantiates another smart contract
Flexible	Less flexible
Less transparent & secure	Transparent & secure
Lower interoperability among contracts	Higher interoperability among contracts
Less likely for deploying proxy clones	More likely for deploying proxy clones
Often when a lower number of proxy contracts are deployed	Often when a lower number of proxy contracts are deployed

RQ3: What are the different types and properties of proxy contracts?



Most proxies (86%) are forwarders whereas 14% are for upgradeability purposes.

Labels

- Upgradeability proxy
- Forwarder proxy

Sample

- 385 random proxy contracts

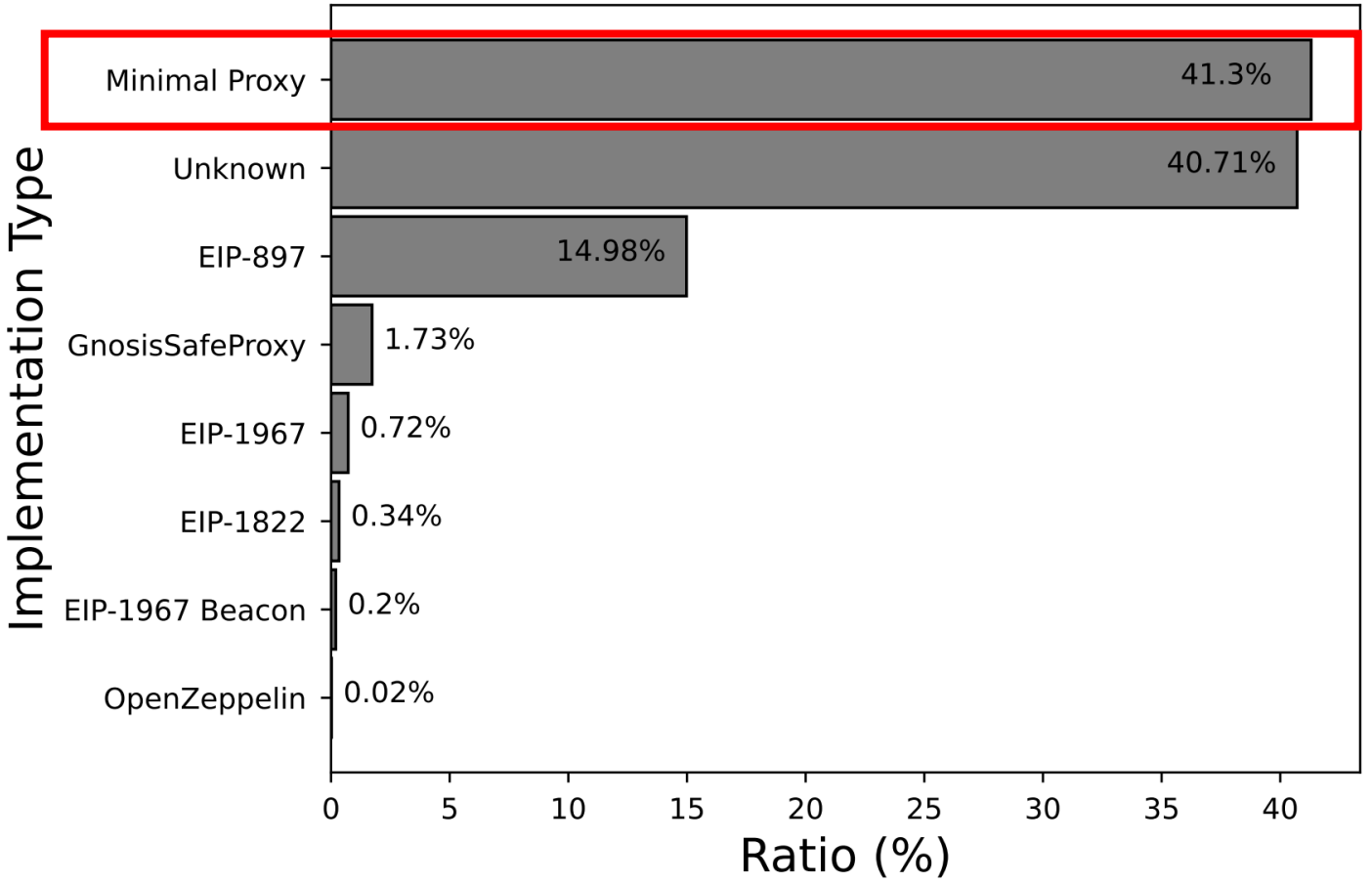
Method

- Qualitative Study

Research gap: Automatic approaches that detect upgradeability proxy contracts and monitor releases

41.3% of proxies are minimal proxies, while 40.7% are unknown implementations.

- Tool
  - Evm-proxy-identification
- Reference implementations
  - 7 known proxy reference implementations
- Sample
  - 16,602 random proxy contracts



Implications to practice

- Proxies reduce transparency in marketplaces.

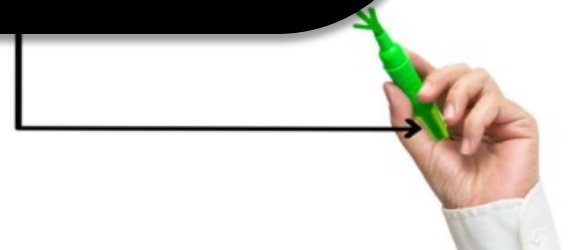
- Future studies should aim to study smart contracts release engineering via upgradeability proxies.

Effective and **efficient** method for detecting proxy contracts.

RQ1: How prevalent is the proxy mechanism in the Ethereum ecosystem?

22

RQ2: What are the different ... ifferent types and properties of deploying proxy contracts?



23

Read more

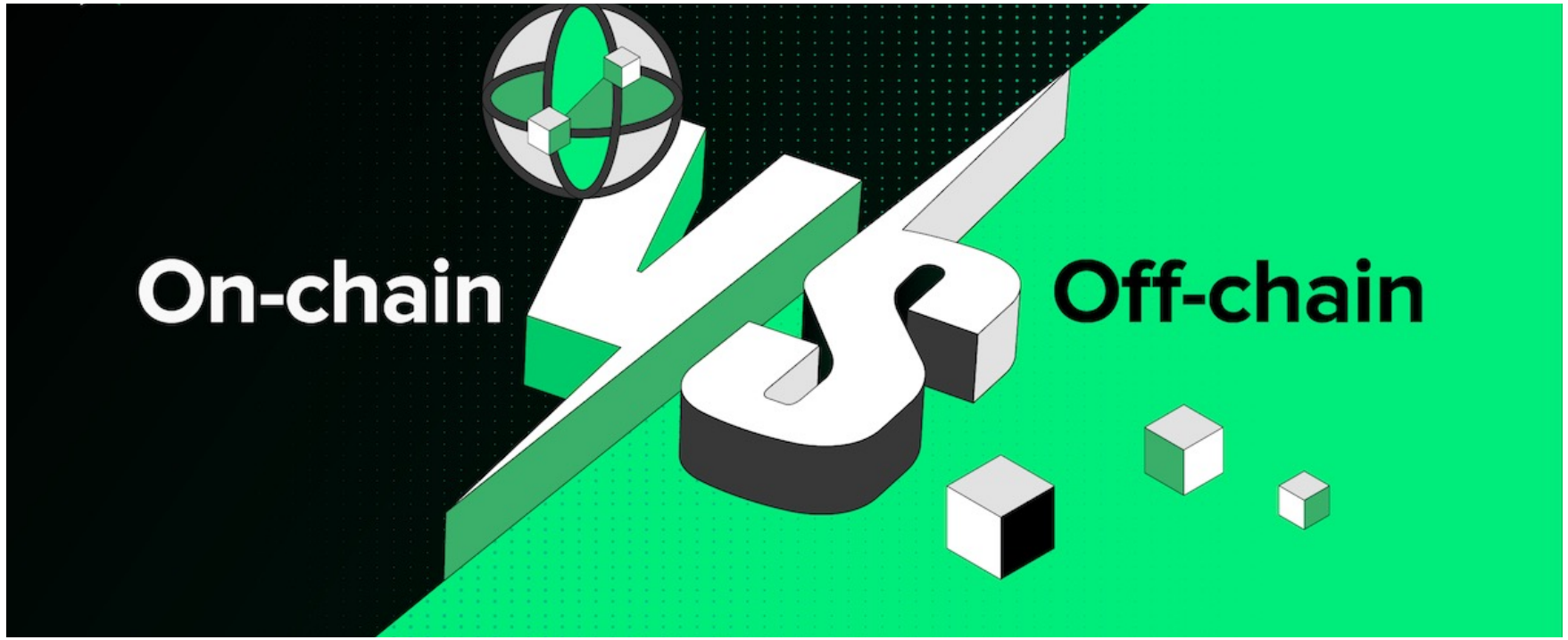
Edu.Ebrahimi@gmail.com

RQ3: What are the different types and properties of proxy contracts?

Effective and efficient method for detecting proxy contracts.

RQ1: How prevalent is the proxy mechanism in the Ethereum ecosystem?

RQ2: What are the different creational patterns for deploying proxy contracts?



RQ3: What are the different types and properties of proxy contracts?

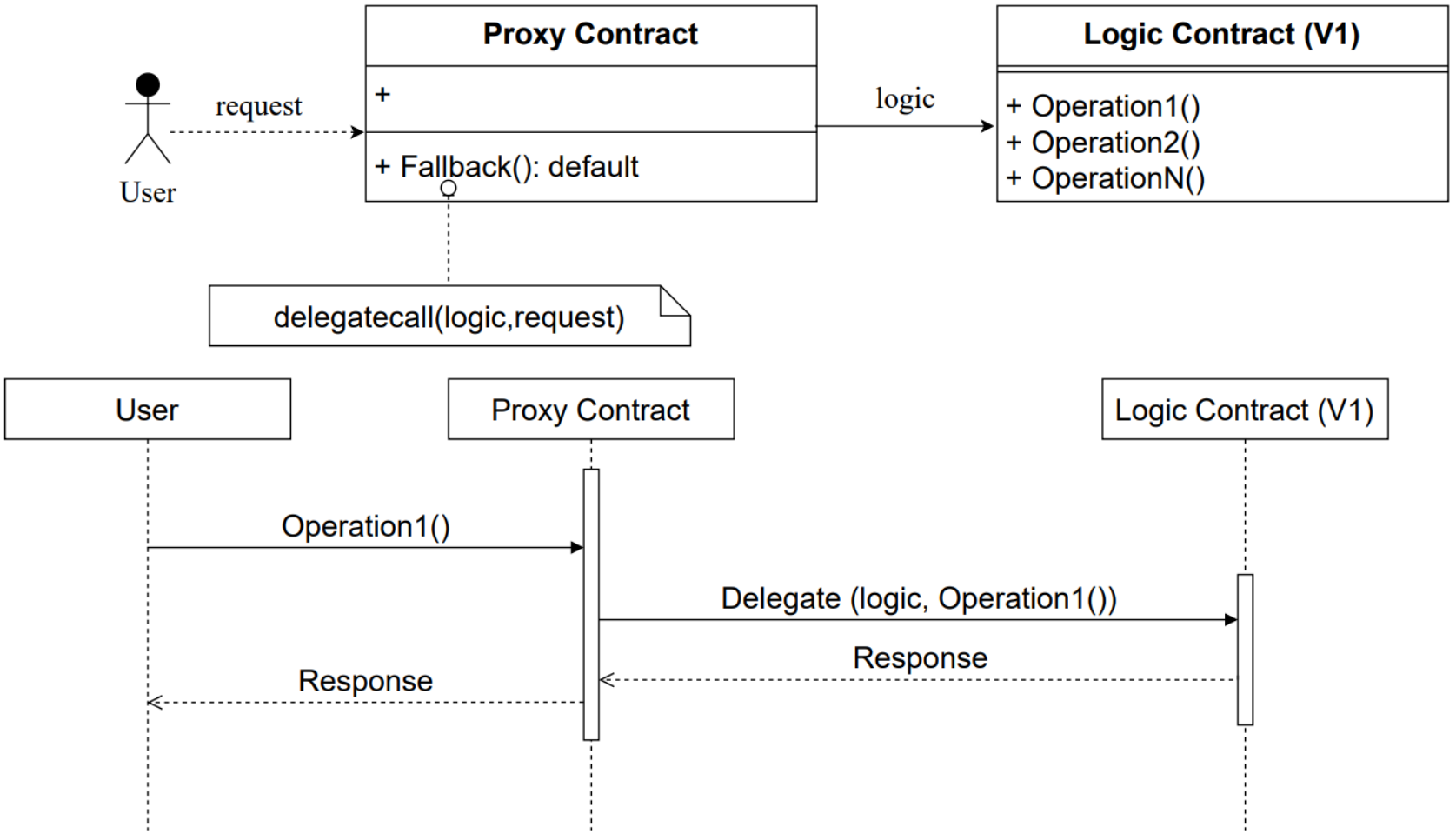
Summary

A. 86% of proxies are forwarders, whereas 14% enable upgradeability

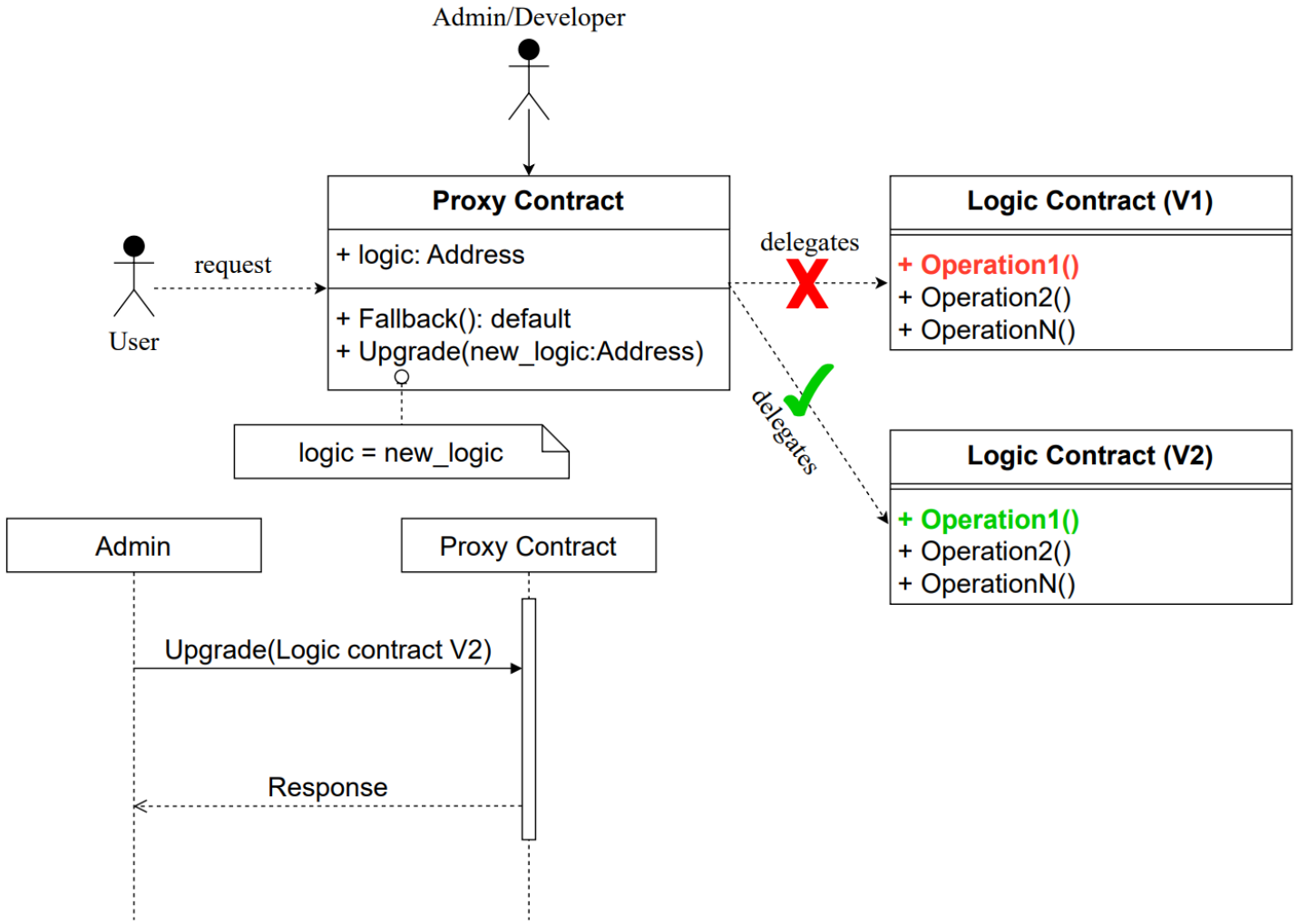
B. The majority of proxies (60%) are implemented according to one of seven reference implementations.

C. 41% of all proxies follow the Minimal Proxy (EIP-1167) reference implementation.

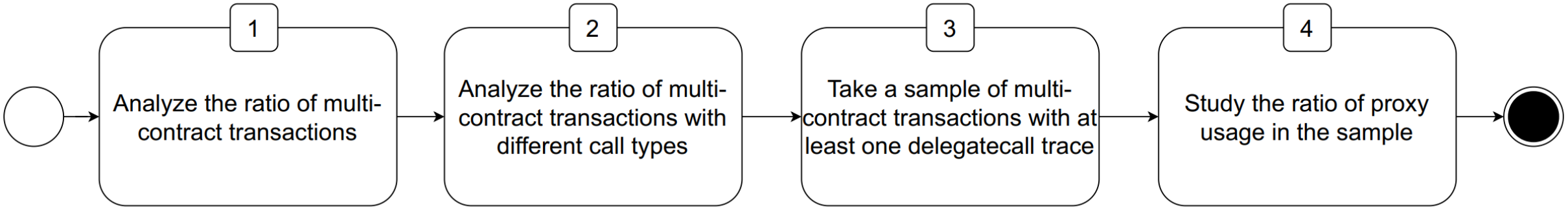
The proxy design pattern in smart contracts

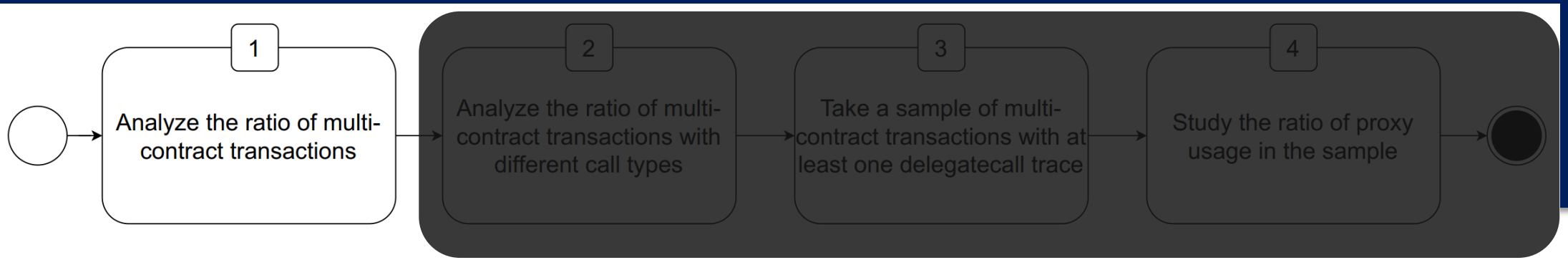


Upgradeability proxy contracts

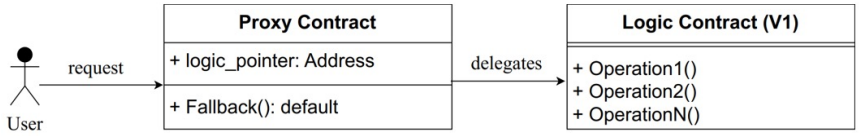


Prem-study: Is the proxy pattern a relevant practice in the domain of smart contracts?

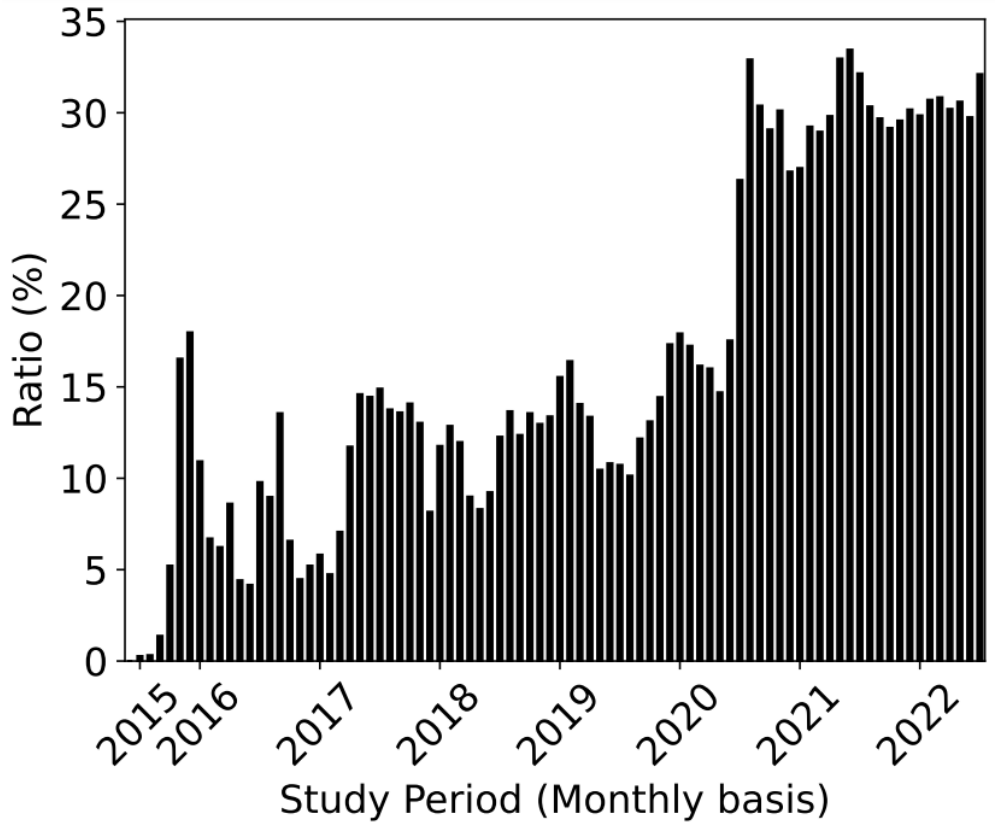




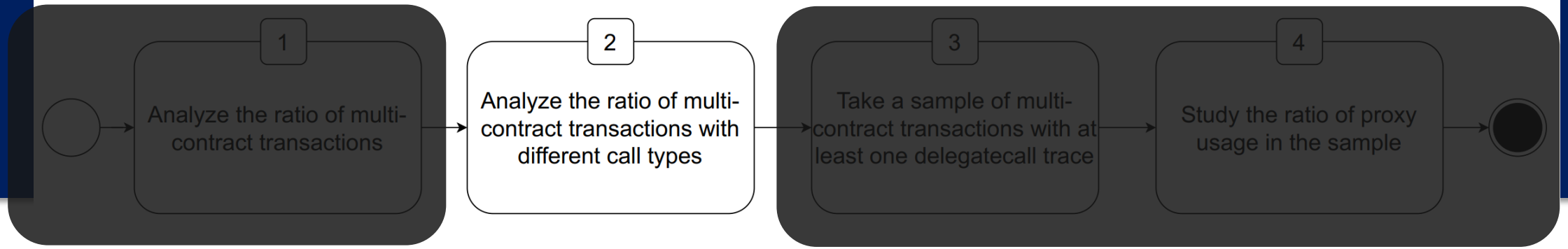
- A proxy increases
  - modularity and encapsulation



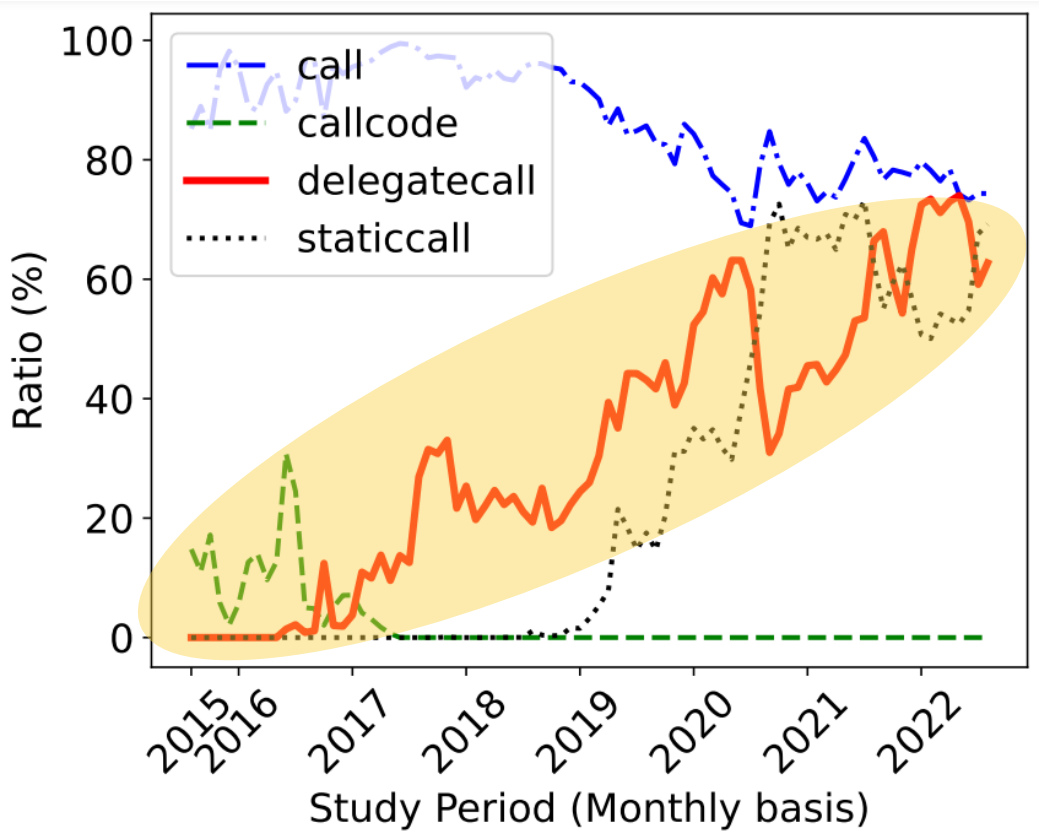
  - If the proxy pattern is employed, then there must be some transactions in which different contracts interact with each.



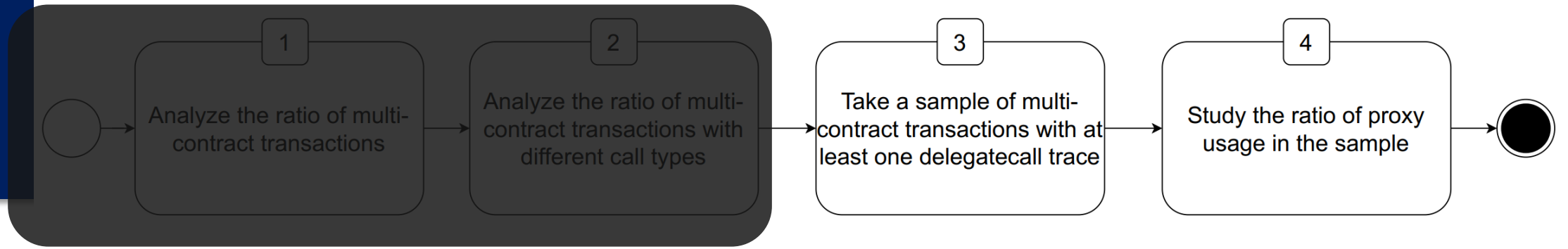
The monthly ratio of multi-contract transactions



- A proxy uses
  - Delegatecall type of call to interact with the actual contract
  - If the proxy pattern is employed, the multi-contract transaction must use delegatecall

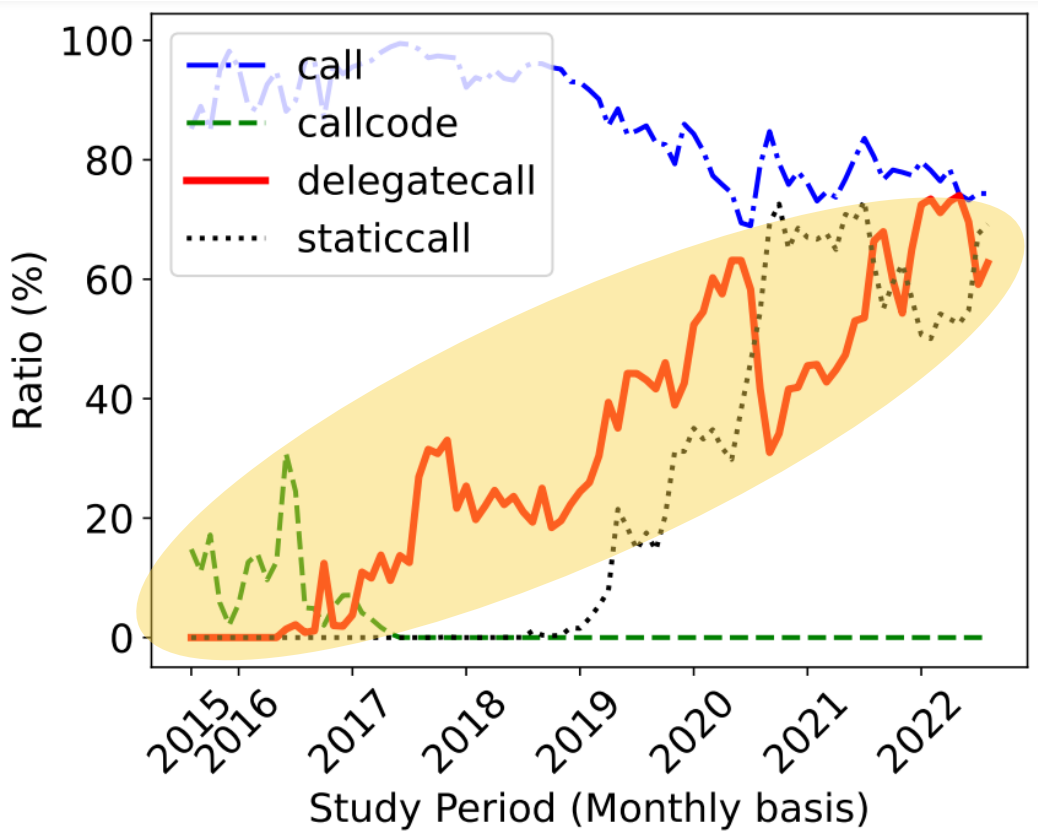


The monthly ratio of multi-contract transactions that use various types of calls.

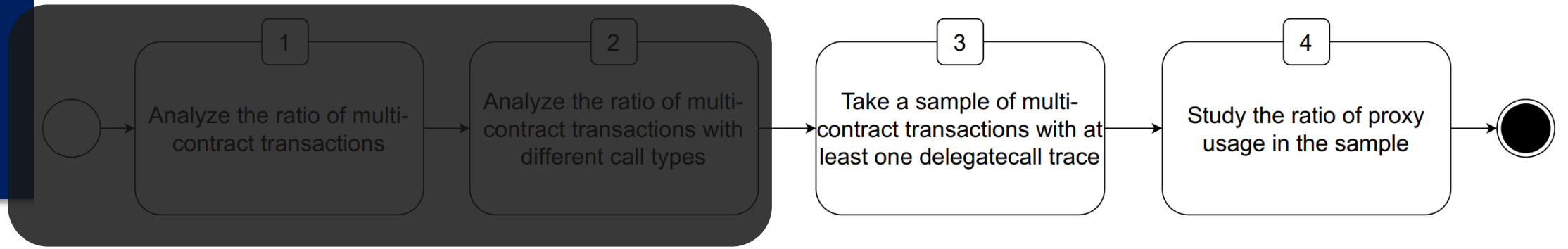


- Not every delegatecall is a sign of using proxy contracts, e.g.,
  - Library calls use delegatecall
  - Etc.

- Use Etherscan to analyze
  - 385 multi-contract transactions with at least one delegatecall operation

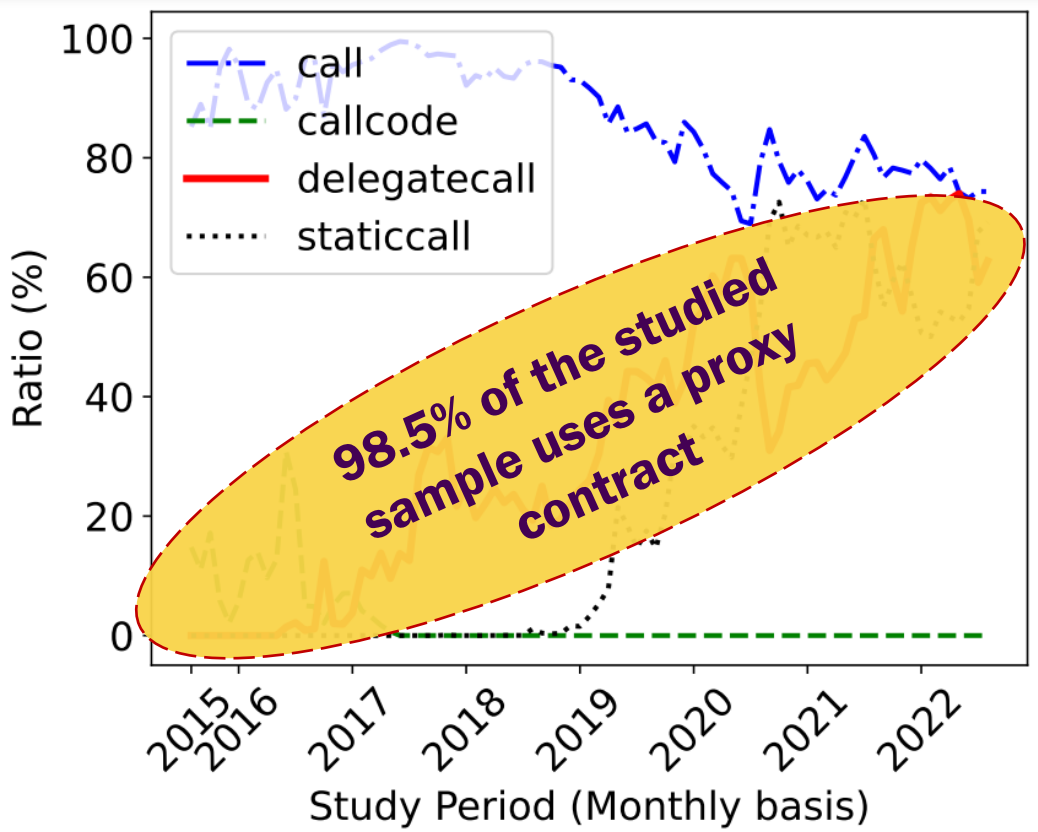


The monthly ratio of multi-contract transactions that use various types of calls.



- Not every delegatecall is a sign of using proxy contracts, e.g.,
  - Library calls use delegatecall
  - Etc.

- Use Etherscan to analyze
  - 385 multi-contract transactions with at least one delegatecall operation



The monthly ratio of multi-contract transactions that use various types of calls.

Prem-study: Is the proxy pattern a relevant practice in the domain of smart contracts?

Summary

1. An increasing trend of up to 33% monthly in the ratio of transactions involving multiple contracts.

2. Most importantly, 98.5% of multi-contract transactions with delegatecalls involve a proxy contract.

RQ2: What are the different creational patterns for deploying proxy contracts?

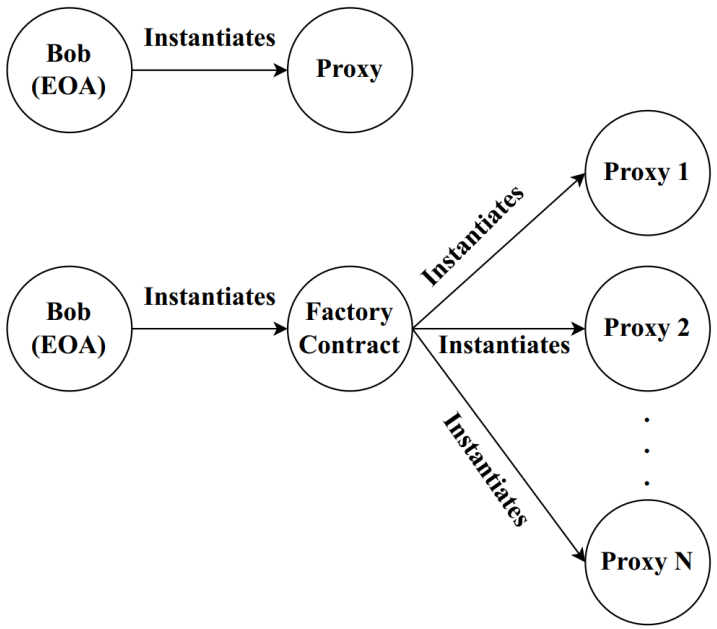
Summary

1. We found 12 creational patterns that are categorized into two major styles: i) on-chain and ii) off-chain deployment styles.

2. While the off-chain deployment of proxies is the most frequently chosen (94.6%) style, the majority (99.3%) of proxies are deployed automatically using the on-chain style..

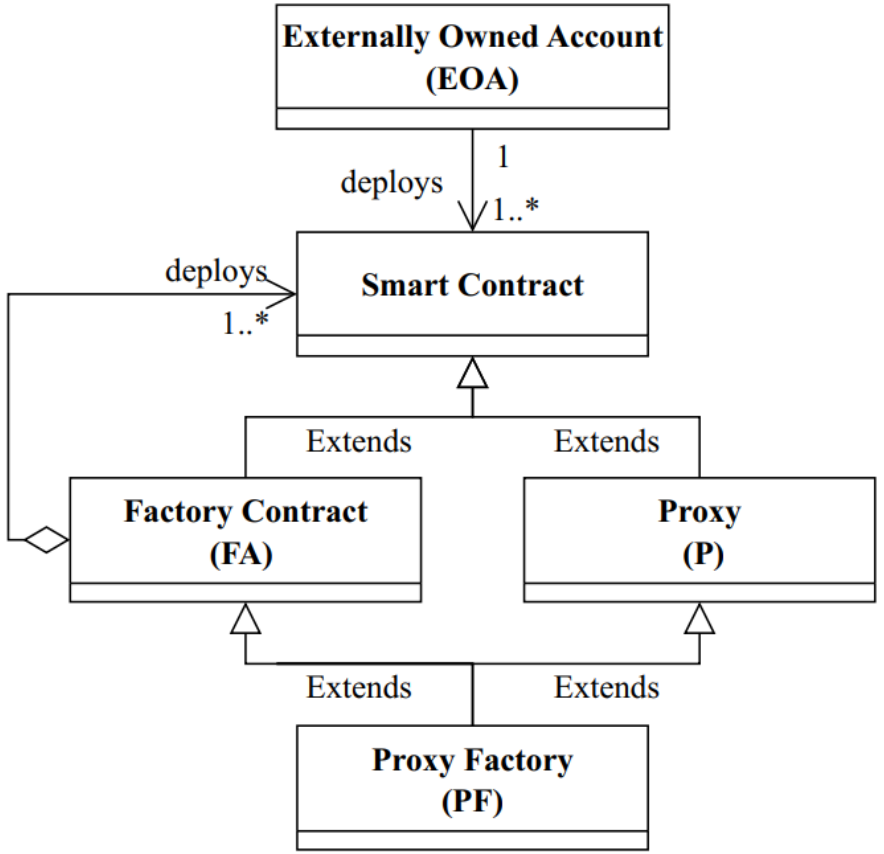
Obs #1: We found 12 different creational patterns for deploying proxy contracts.

Id	Creational pattern
1	EOA > P
2	EOA > FA > P



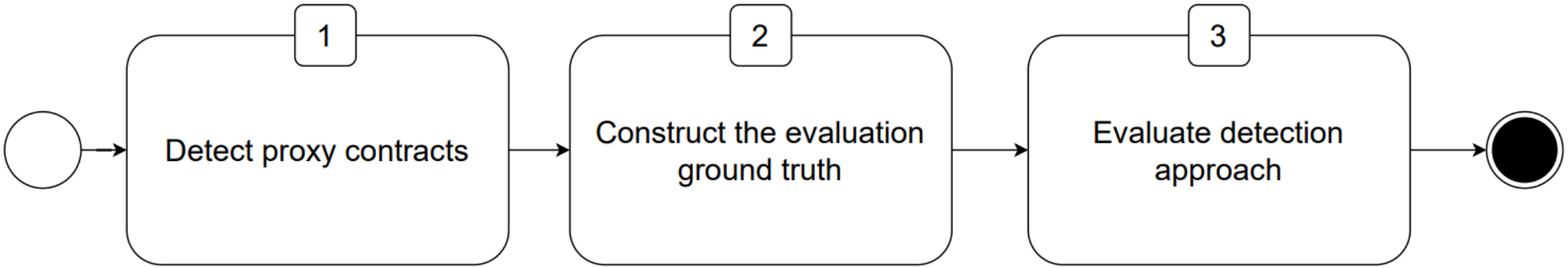
Obs #1: We found 12 different creational patterns for deploying proxy contracts.

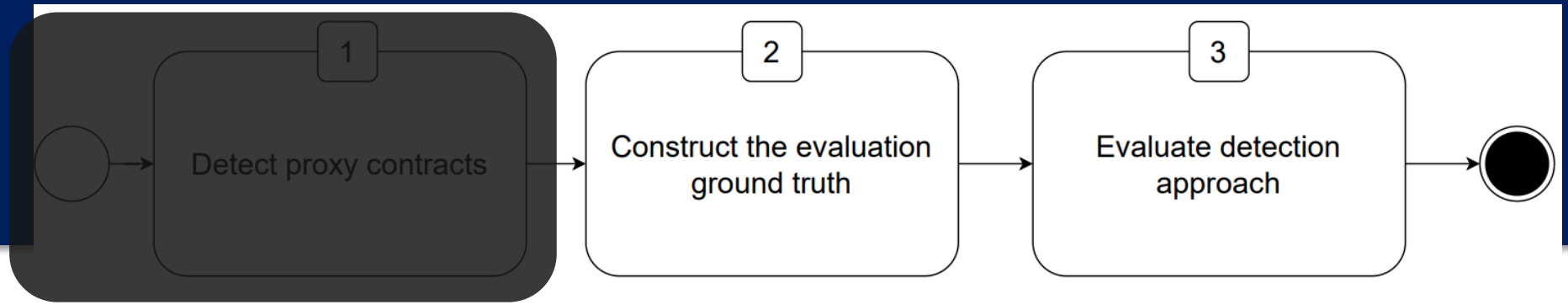
Id	Creational pattern
1	EOA > P
2	EOA > FA > P
3	EOA > PF > P
4	EOA > FA > PF > P
5	EOA > FA > FA > P
6	EOA > PF > PF > P
7	EOA > FA > PF > PF > P
8	EOA > FA > FA > FA > P
9	EOA > FA > FA > PF > P
10	EOA > FA > FA > FA > FA > P
11	EOA > FA > FA > PF > PF > P
12	EOA > FA > FA > PF > PF > PF > P



A metamodel that summarizes proxy creational patterns

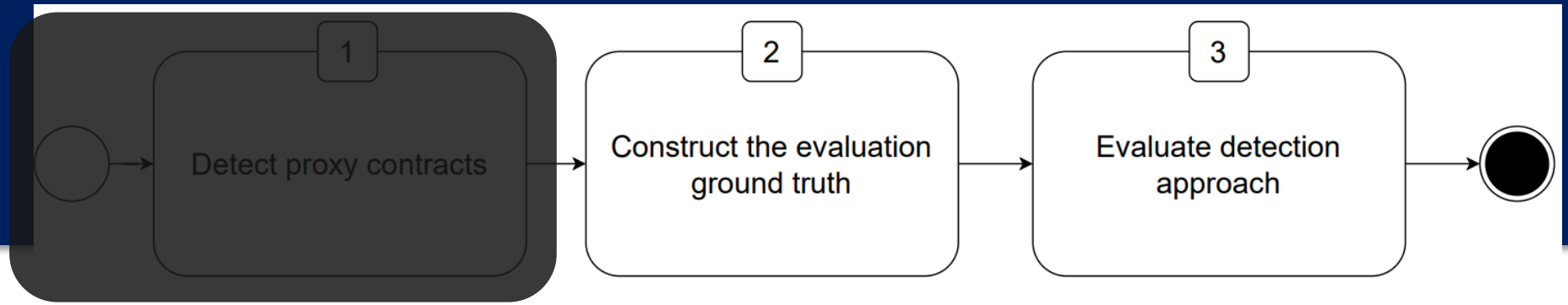
Proxy Detection Approach & Evaluation





i. Ground truth dataset (385 random contracts)
   - 90 proxy contracts
   - 295 others

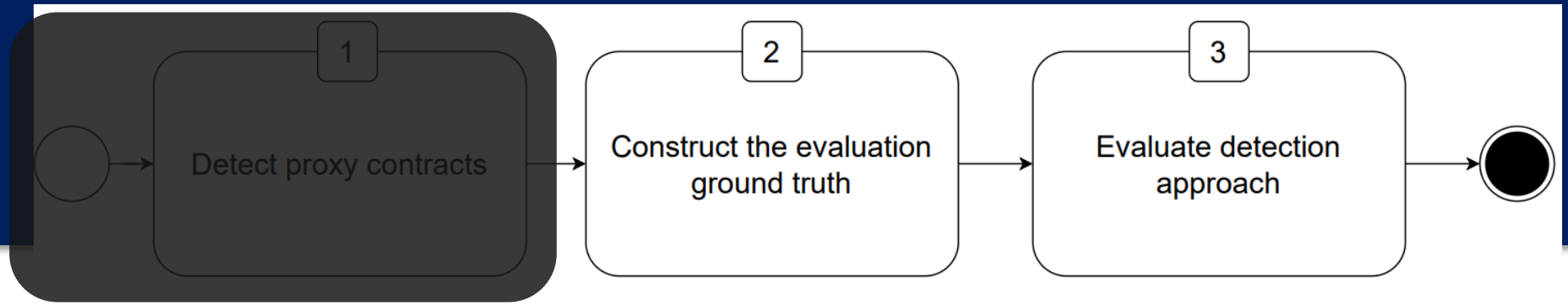
ii. Compared our performance with Salehi et. al's work.



- Ground truth dataset (385 random contracts)
  - 90 proxy contracts
  - 295 others

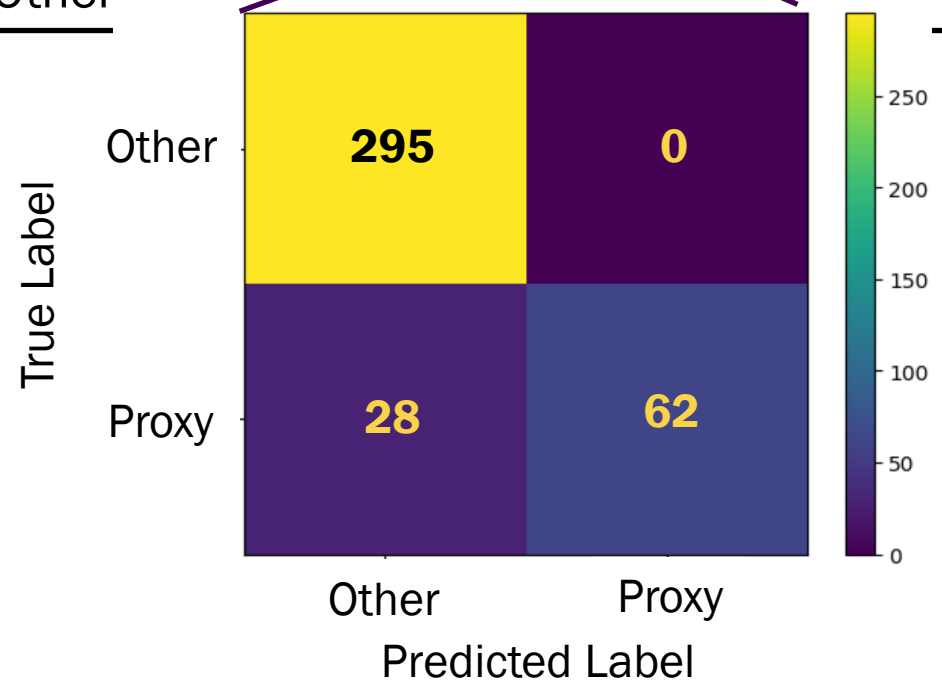
Label	Precision	Recall	F1-measure
Proxy	100%	68.9%	81.6%
Other	91.3%	100%	95.5%

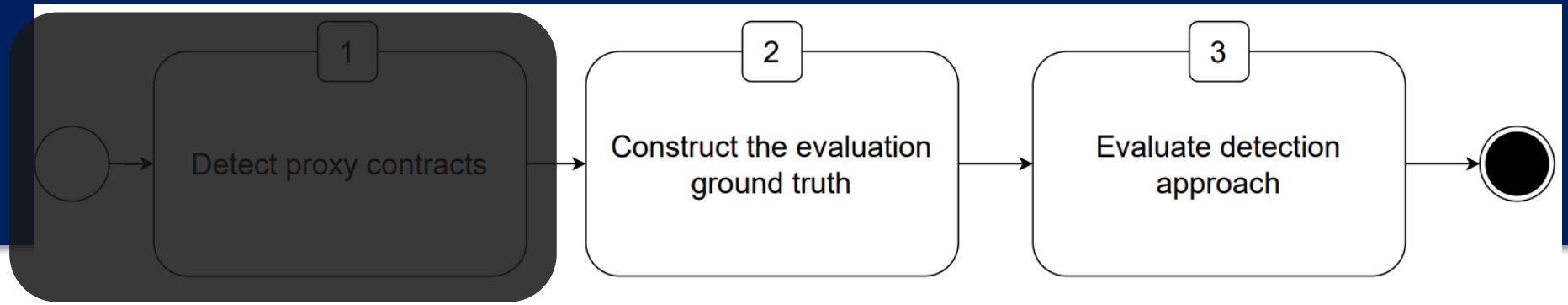
The performance of our proxy detection approach.



- Ground truth dataset (385 random contracts)
  - 90 proxy contracts
  - 295 others

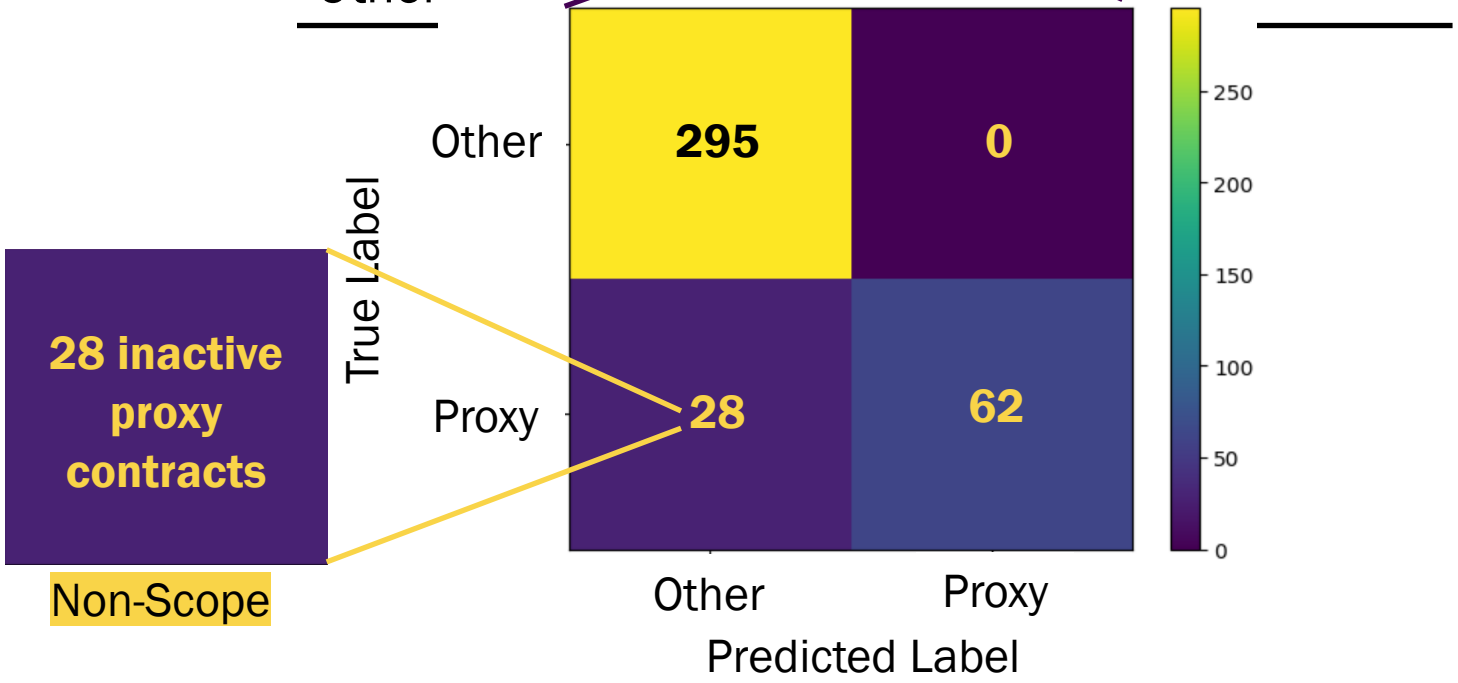
Label	Precision	Recall	F1-measure
Proxy	100%	68.9%	81.6%
Other	91.3%	100%	95.5%

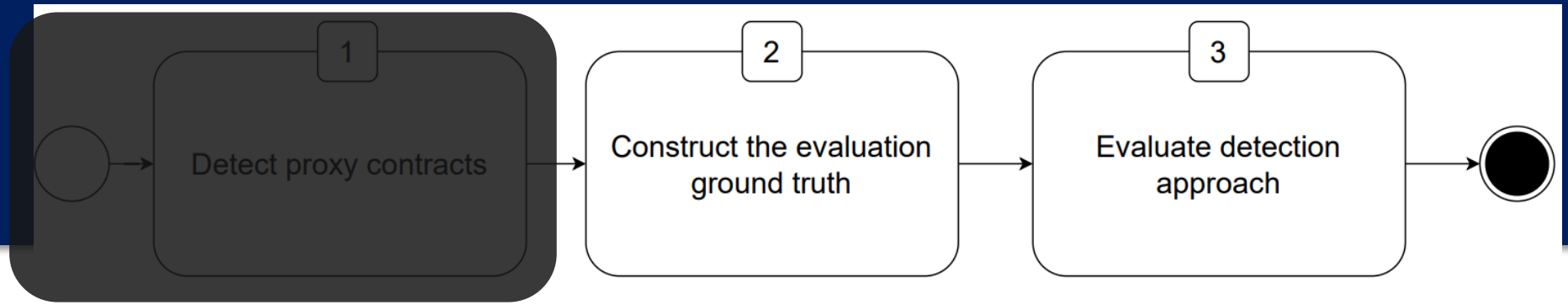




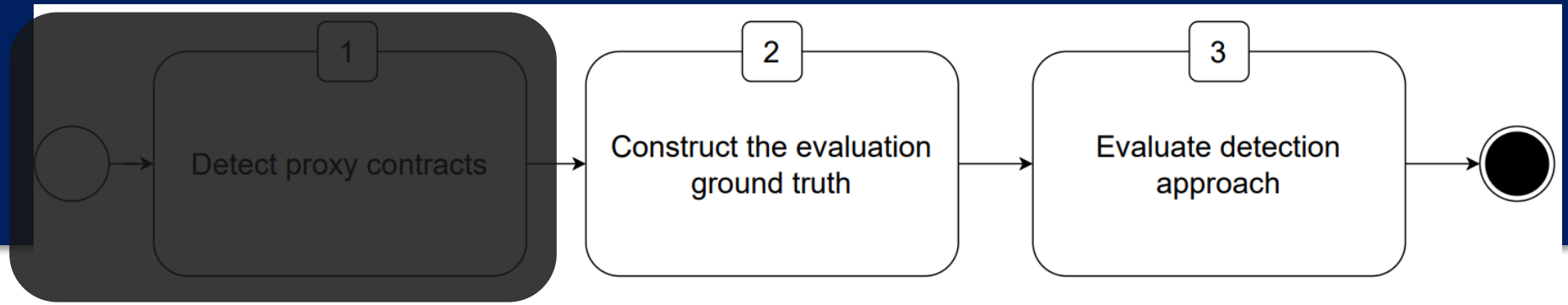
- Ground truth dataset (385 random contracts)
  - 90 proxy contracts
  - 295 others

Label	Precision	Recall	F1-measure
Proxy	100%	68.9%	81.6%
Other	91.3%	100%	95.5%





- Compared our performance with Salehi et. al's work.
  - Both approaches are behavioral
  - Sep-05-2020 to Jul-20-2021



- Compared our performance with Salehi et. al's work.
  - Both approaches are behavioral
  - Sep-05-2020 to Jul-20-2021

Approach	Number of proxy contracts
Salehi et. al. approach	1,427,215
Our approach	1,723,309

We detected
~ 300K more proxies

RQ1: How prevalent is the proxy mechanism in the Ethereum ecosystem?

Summary

A. Our approach
   i. Efficiently mines active proxy contracts (in under 20 min)
   ii. Achieve perfect precision and recall
   iii. Improves upon previous studies
B. Over 14% of contracts are proxies.
C. Our assessment from the three viewpoints of usage context, stakeholder adoption, and smart contracts design shows that the tendency for using proxy contracts is growing.