A Blockchain based Witness Model for Trustworthy Cloud Service Level Agreement *Enforcement*

Huan Zhou, Xue Ouyang, Zhijie Ren, Jinshu Su, Cees de Laat, Zhiming Zhao

UNIVERSITY OF AMSTERDAM

IEEE INFOCOM

Paris 1/May/2019

Outline

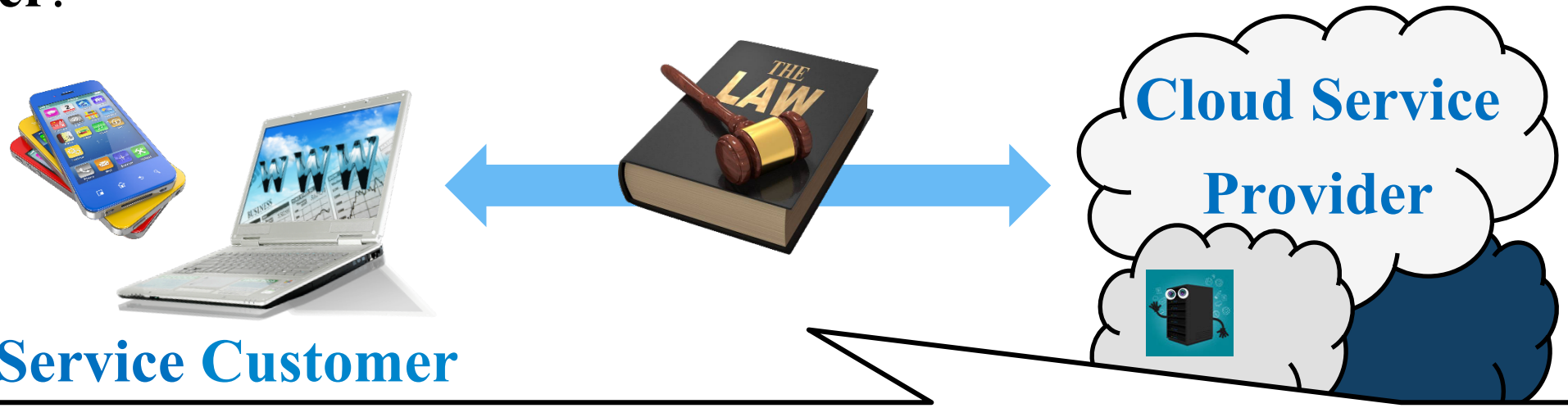
- Cloud SLA / Blockchain: background and challenges
- Witness Model Design and Key Techniques to Ensure Trustworthiness
  - Witness Management and Unbiased Random Selection
  - Payoff Function Design and Nash Equilibrium
  - Witness Audit Mechanism
- Experimental Study
- Conclusions

What is Cloud SLA?

Cloud SLA (Service Level Agreement) is a **business** concept which defines the contractual and financial agreements between the Cloud **customer** and **provider**.

What is Cloud SLA?

Cloud SLA (Service Level Agreement) is a **business** concept which defines the contractual and financial agreements between the Cloud **customer** and **provider**.



Example:
A Cloud customer, ***C***, buys a VM (Virtual Machine), ***X***, from an IaaS Cloud provider, ***P***, for one hour. They make an agreement: in this one hour,

- If the VM, ***X***, does not crash, ***C*** → ***P*** 1000 credits. (payment)
- If the VM, ***X***, crashes, ***C*** → ***P*** 500 credits. (compensation)

What is Cloud SLA?

Cloud SLA (Service Level Agreement) is a **business** concept which defines the contractual and financial agreements between the Cloud **customer** and **provider**.

Cloud Service

Provider is in a centralized and dominating position:

- **Less fair;**
- **Lack of violation proof;**
- **Manual enforcement.**

Clo

Ex
A
ho

- If the VM, ***X***, does not crash, ***C*** → ***P*** 1000 credits. (payment)
- If the VM, ***X***, crashes, ***C*** → ***P*** 500 credits. (compensation)

What is Cloud SLA?

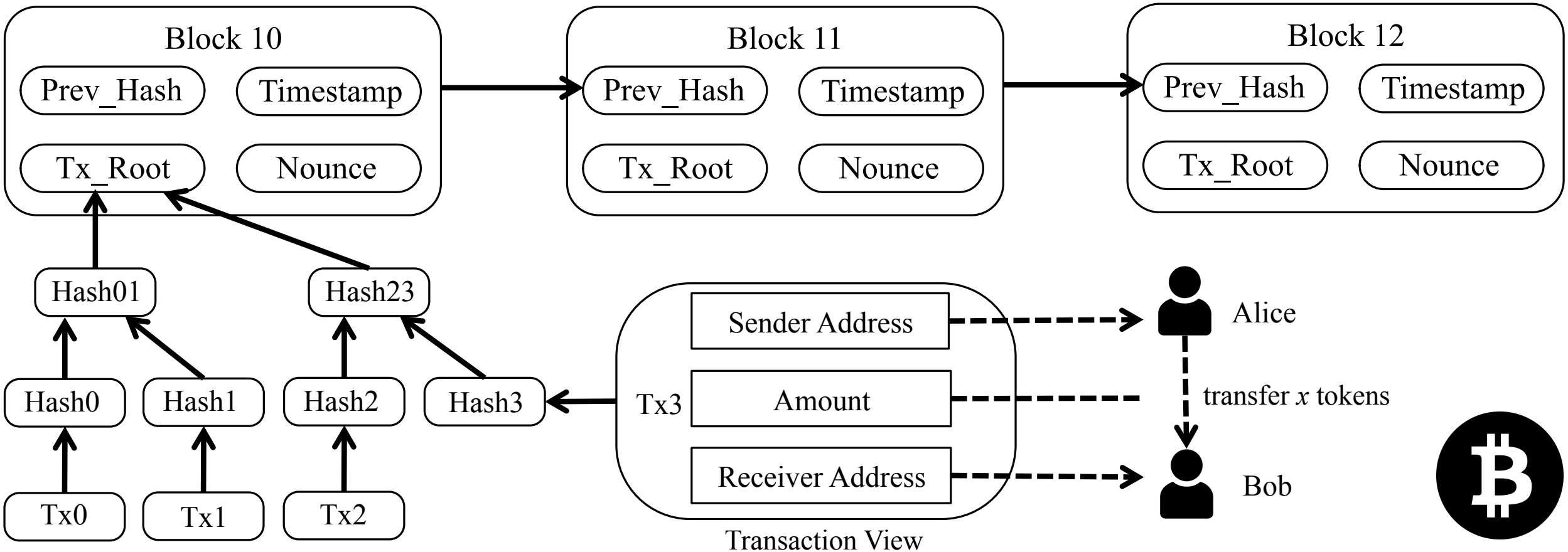
Cloud SLA (Service Level Agreement) is a **business** concept which defines the contractual and financial agreements between the Cloud **customer** and **provider**.

1. **How to ensure** the fairness of the provider and customer in the agreement?
2. **Who and how to detect** and **prove** the SLA violation?
3. **How to automate** the process of payment and especially the compensation?

- If the VM, ***X***, crashes, ***C*** → ***P*** 500 credits. (compensation)

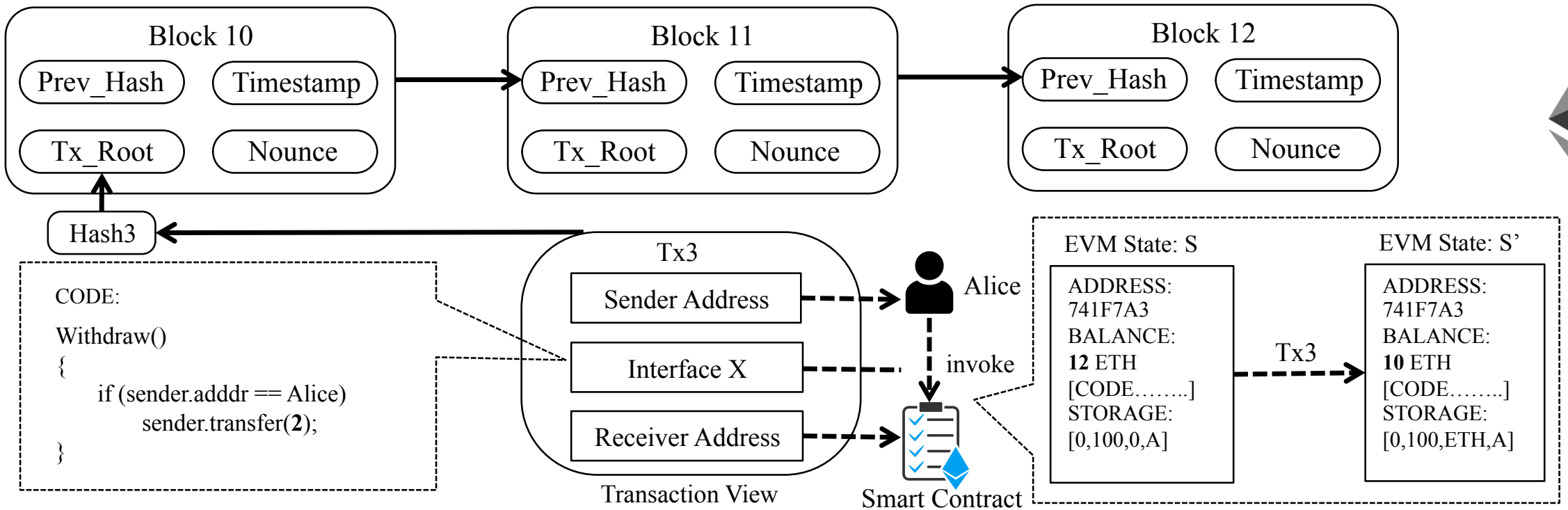
Blockchain: **decentralized** and **immutable** ledger

- Blockchain is a technique, which makes every participant having consensus on a decentralized ledger, e.g., through PoW (Proof of Work).
- Bitcoin is the first generation application of blockchain, from 2009.



Blockchain: **smart contract**

- Ethereum is the second generation blockchain, from 2015.
- It proposes EVM (Ethereum Virtual Machine), which is a set of byte values to represent a virtual machine state.
- Ethereum works as a world-wide computer. The program running on this computer is named as **Smart Contract**.



Opportunities: **automation**

A Cloud customer, ***C***, buys a VM (Virtual Machine), ***X***, from an IaaS Cloud provider, ***P***, for one hour. They make an agreement: in this one hour,

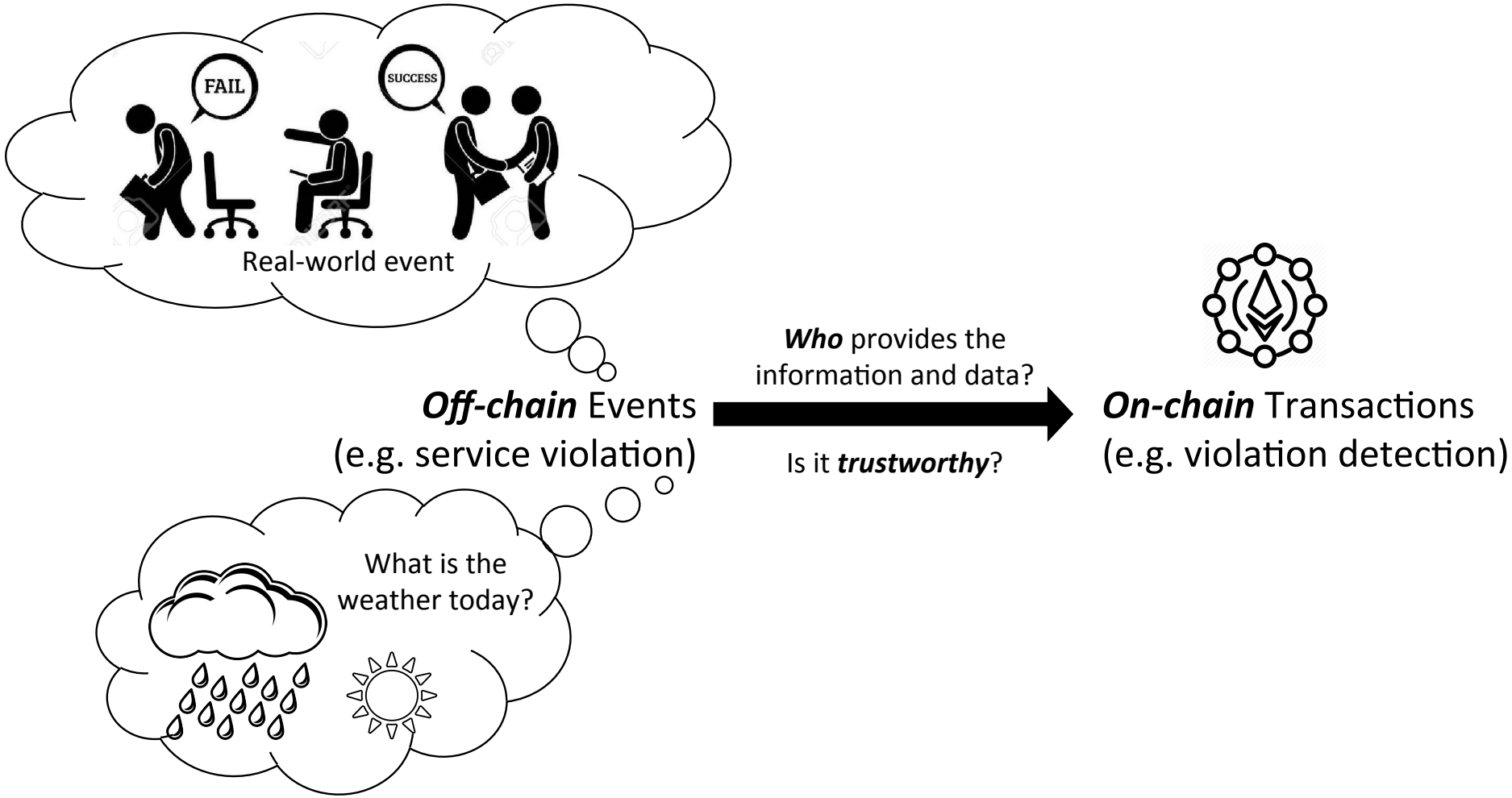
- If the VM, ***X***, does not crash, ***C*** → ***P*** 1000 credits. (payment)
- If the VM, ***X***, crashes, ***C*** → ***P*** 500 credits. (compensation)

Customer invokes this Payment interface

Smart Contract CODE:

```
Payment()
{
    if( !X.violated )
        C.transfer(P, 1000)
    else
        C.transfer(P, 500)
}
```

Challenges: **violation detection?**



In the context of Cloud SLA, **who** can be the **judge** to convince both, the provider and customer, that the service violation really happens? **How**?

Current solution: **oracle**

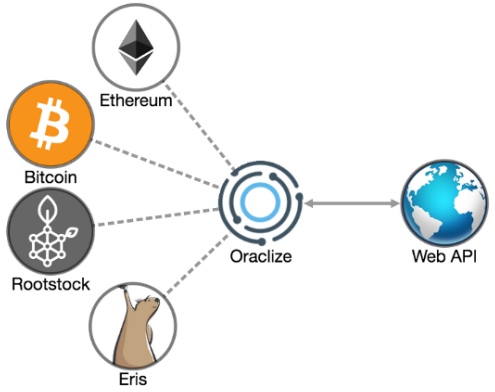
Who?
Oracle: Perform as "Data Carrier"

Off-chain Events → ***On-chain*** Transactions

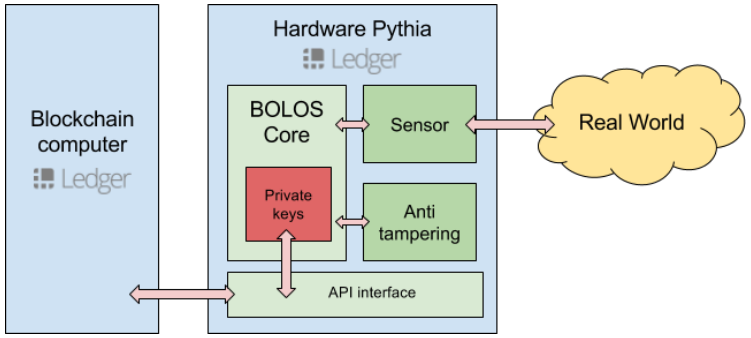
Is it ***trustworthy***?

Third trusted party

Software Oracle



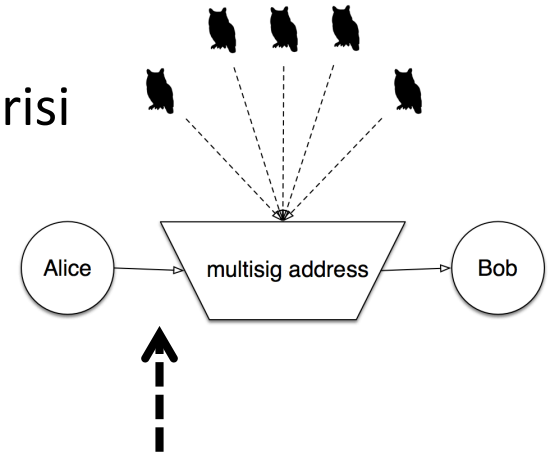
Hardware Oracle



- Must trust the third party;
- Single point of failure;

Distributed oracles

Orisi

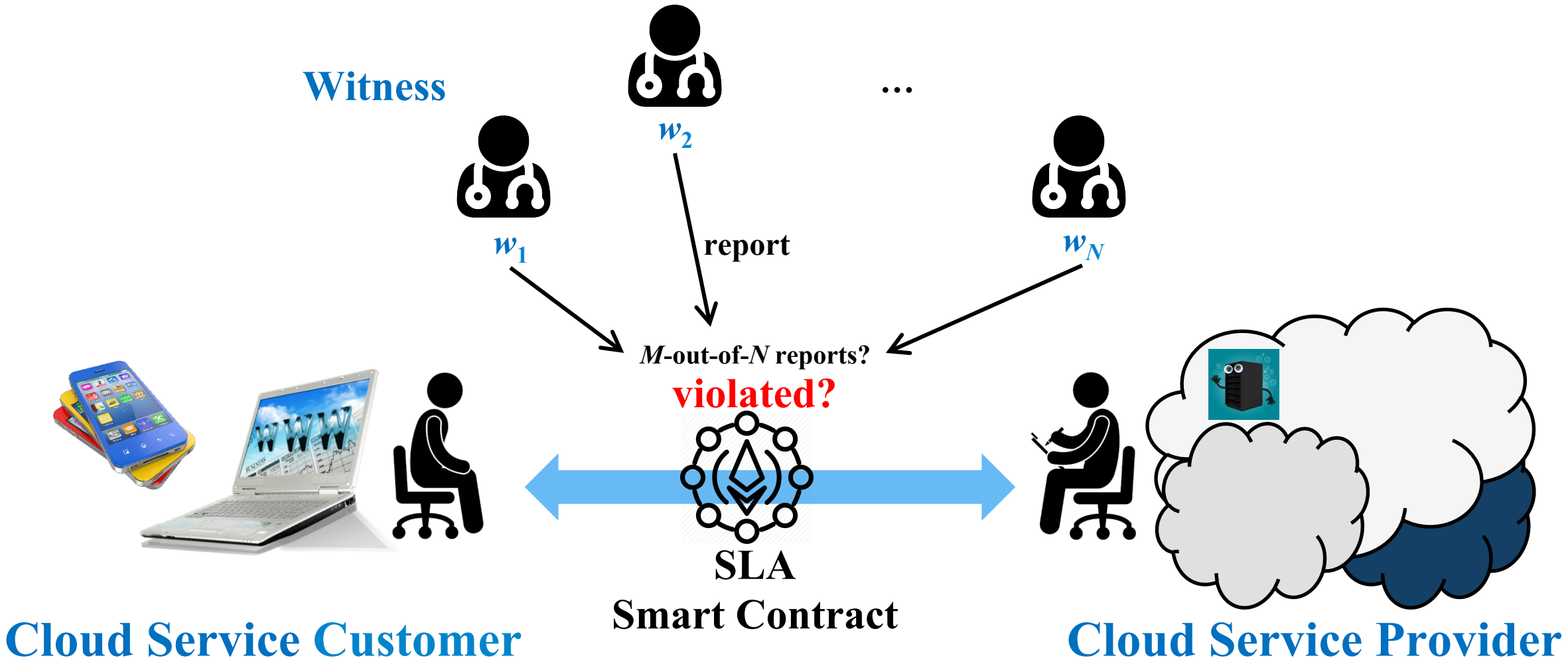


For oracles:

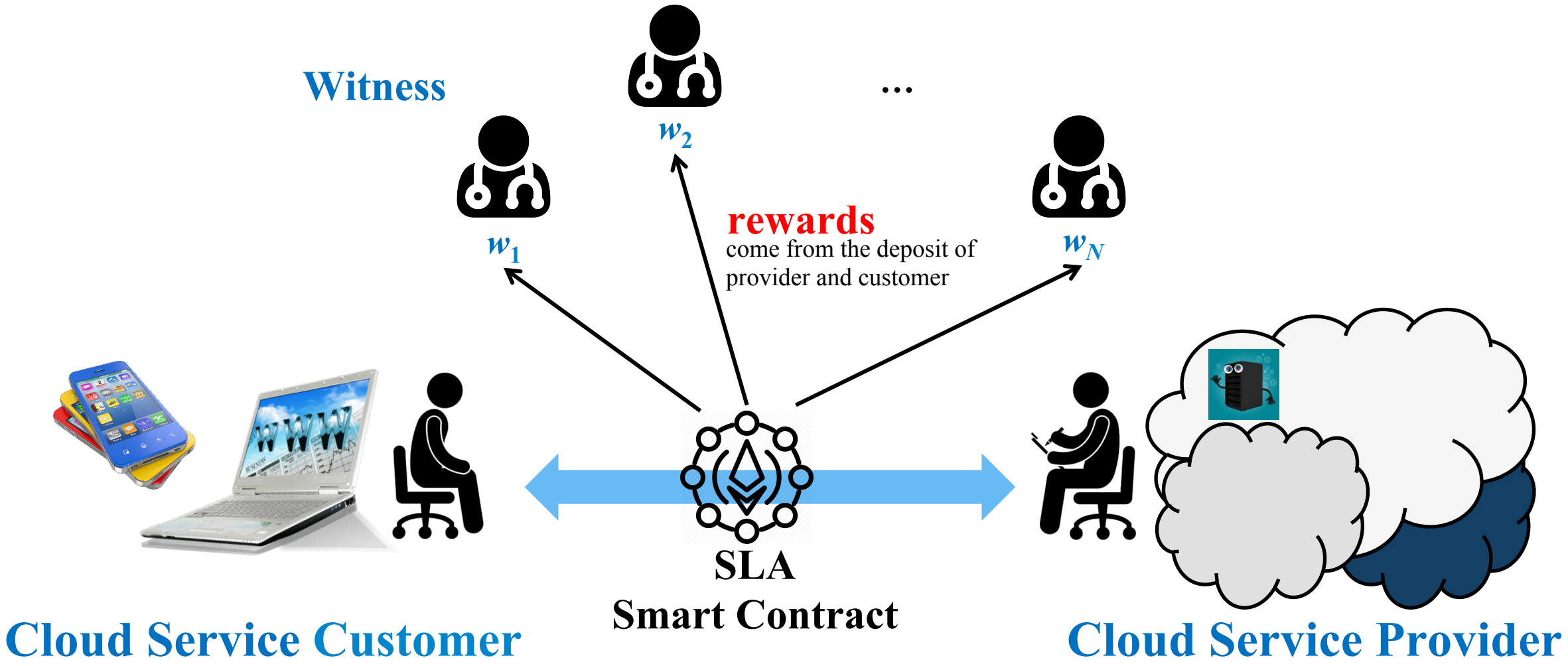
- Require them independent and trustworthy;
- No incentive;
- Consensus issue;

Our proposal:
decentralized witness model

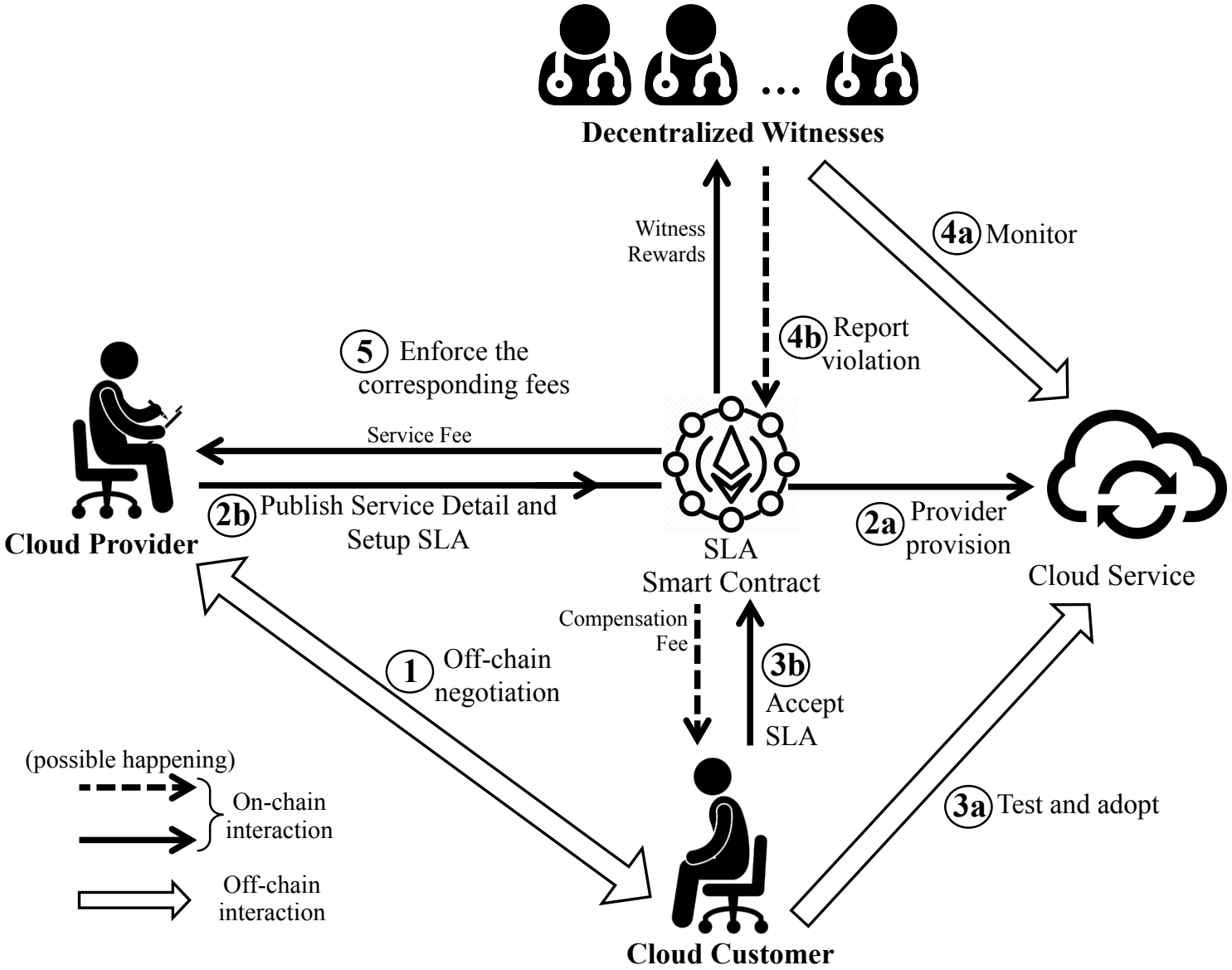
Our proposal: **decentralized witness model**



Our proposal: **decentralized witness model**



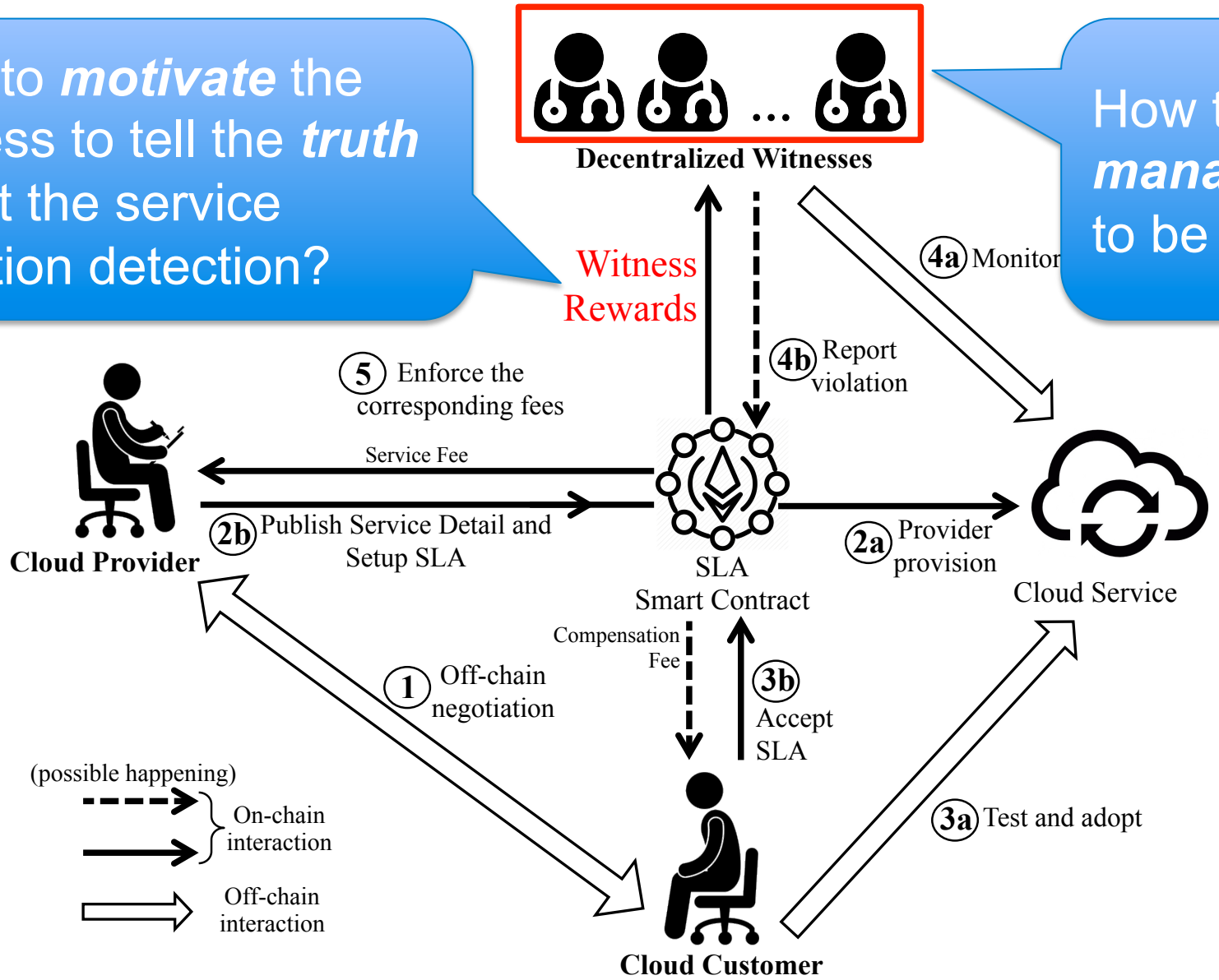
How does it work in Cloud SLA



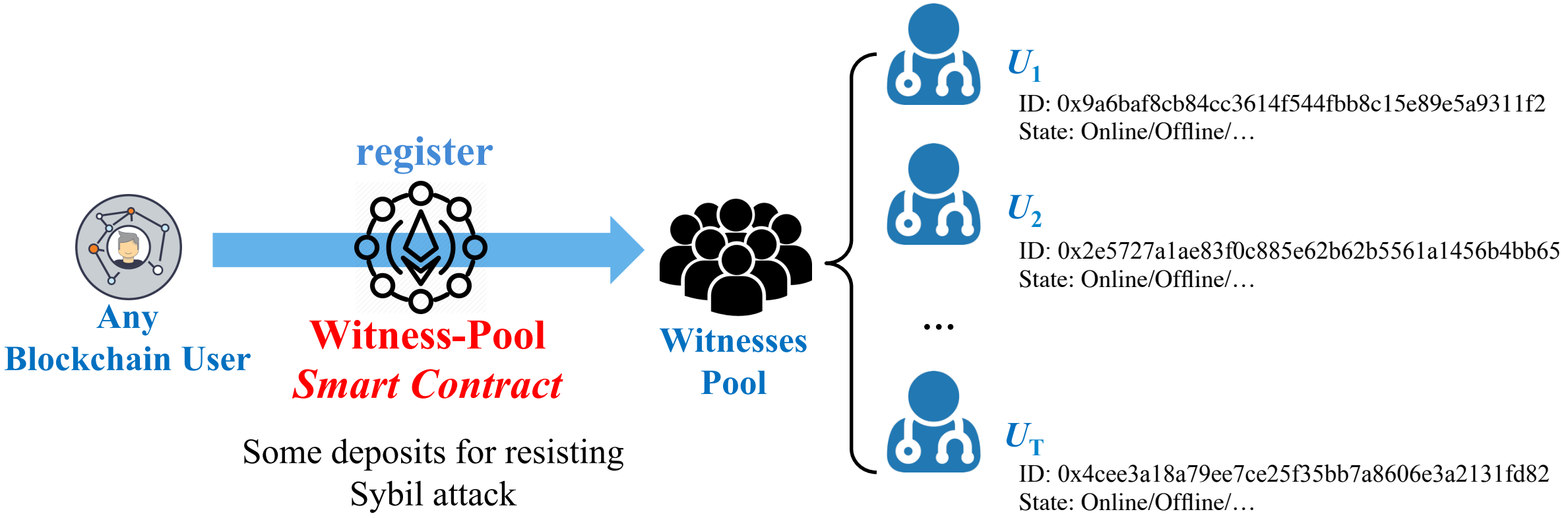
How does it work in Cloud SLA

How to ***motivate*** the witness to tell the ***truth*** about the service violation detection?

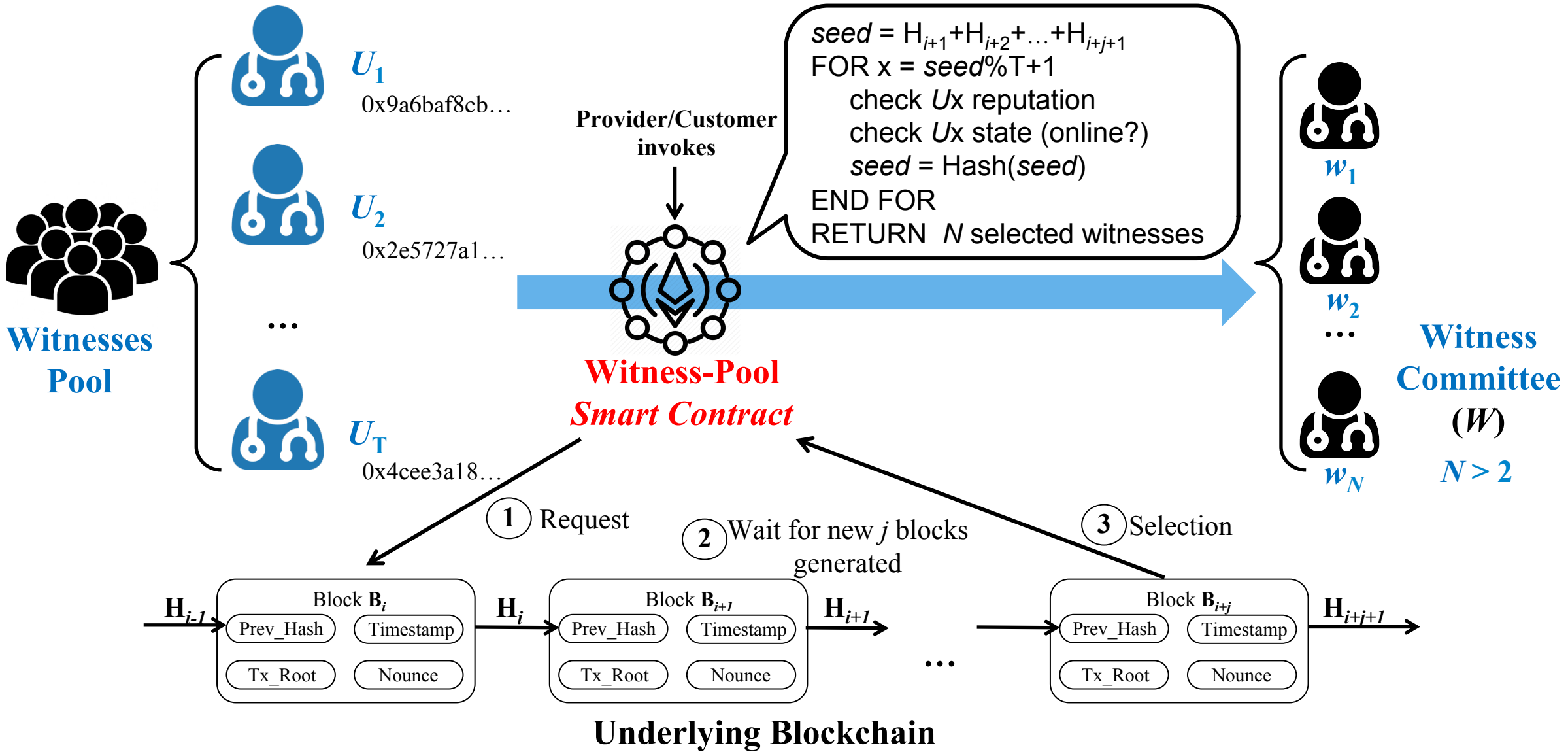
How the witnesses are ***managed*** and ***selected*** to be independent?



Decentralized Witnesses Pool



Unbiased Random Selection Procedure



Witness-as-a-Game

Strategic Form Game with Complete Information

Witness Committee (W)

Players: w_1 w_2 … w_N

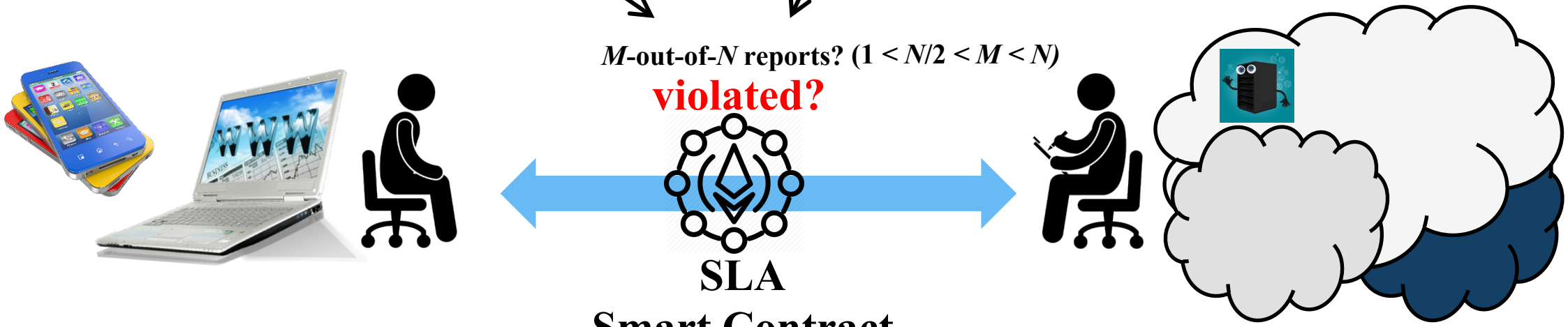
Actions:
- $\sigma_k^{(r)}$: w_k **R**eport the service violation to the smart contract
- $\sigma_k^{(s)}$: w_k do not report and keep **S**ilence to the smart contract

report

Payoff function $\pi_k(\sigma_k, \sigma_{-k})$: rewards of w_k in this strategy profile

***M*-out-of-*N* reports? ($1 < N/2 < M < N$)**

violated?



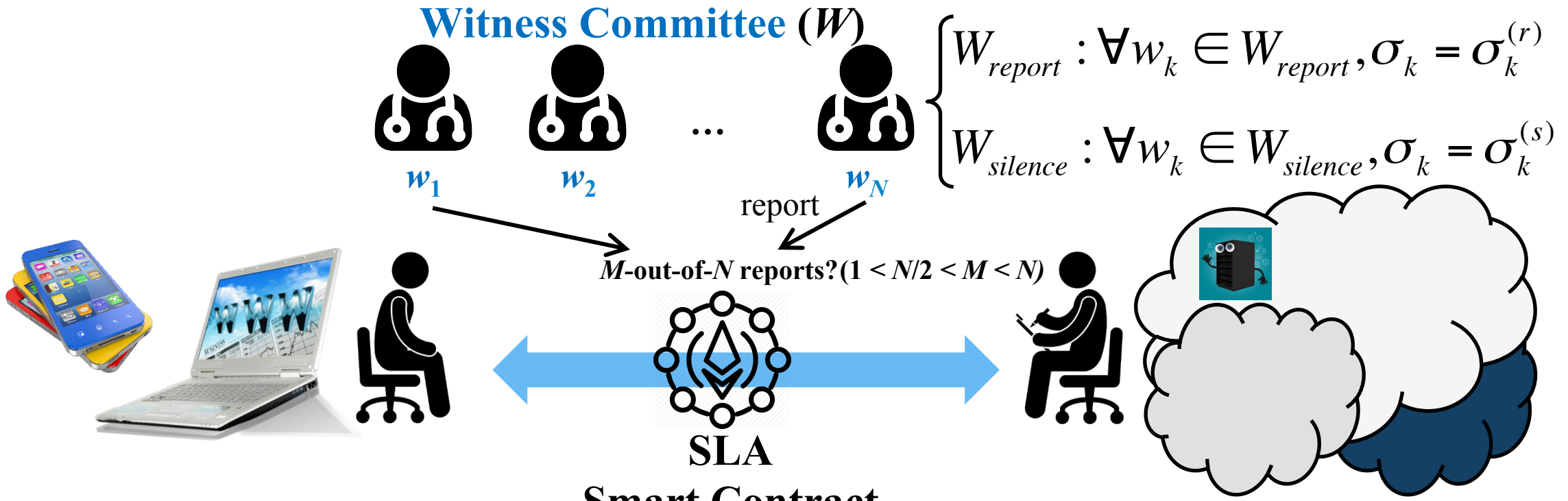
SLA

Smart Contract

Cloud Service Customer

Cloud Service Provider

Payoff: **witness incentive model**



$$\begin{cases} W_{report} : \forall w_k \in W_{report}, \sigma_k = \sigma_k^{(r)} \\ W_{silence} : \forall w_k \in W_{silence}, \sigma_k = \sigma_k^{(s)} \end{cases}$$

$\|W_{report}\| \geq M$?

yes → **violated**

no → **not violated**

Payoff function

$$\forall w_k \in W_{report}, \pi_k(\sigma_k^{(r)}, \sigma_{-k}) = 10$$

$$\forall w_k \in W_{silence}, \pi_k(\sigma_k^{(s)}, \sigma_{-k}) = 0$$

$$\forall w_k \in W_{report}, \pi_k(\sigma_k^{(r)}, \sigma_{-k}) = -1$$

$$\forall w_k \in W_{silence}, \pi_k(\sigma_k^{(s)}, \sigma_{-k}) = 1$$

Proof: **Nash Equilibrium**

In the witness game, there are two and only two Nash equilibrium points:

- $\forall w_k \in W, \sigma_k = \sigma_k^{(r)}$
- $\forall w_k \in W, \sigma_k = \sigma_k^{(r)}$

Take the example of three-witness game $(N = 3, M = 2)$

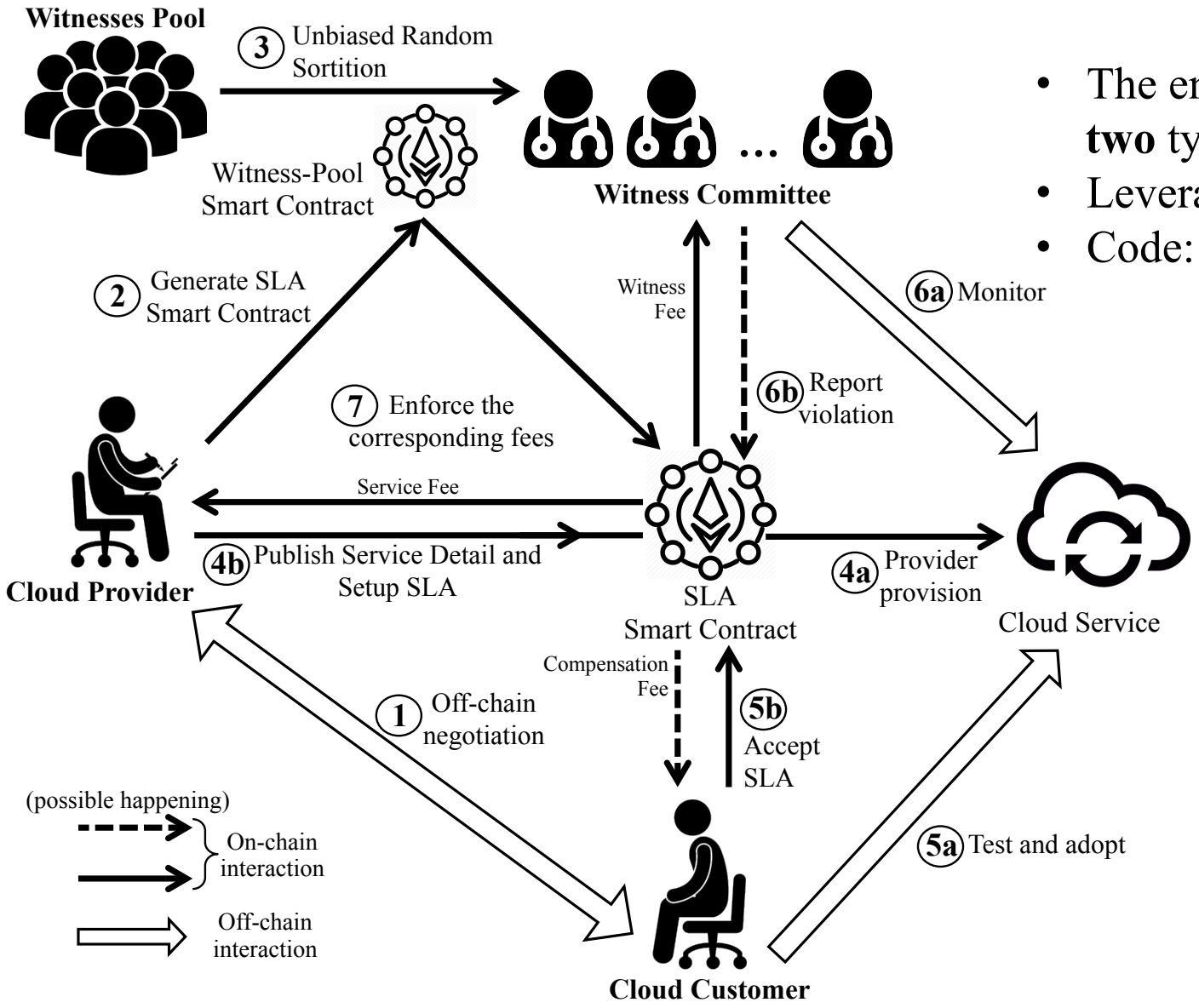
w_1 : *Alice*	w_3 : *Candy*			
	$\sigma_3^{(r)}$: *Report*		$\sigma_3^{(s)}$: *Silence*	
	w_2 : *Bob*		w_2 : *Bob*	
	$\sigma_2^{(r)}$: *Report*	$\sigma_2^{(s)}$: *Silence*	$\sigma_2^{(r)}$: *Report*	$\sigma_2^{(s)}$: *Silence*
$\sigma_1^{(r)}$: *Report*	(10, 10, 10)	(10, 0, 10)	(10, 10, 0)	(-1, 1, 1)
$\sigma_1^{(s)}$: *Silence*	(0, 10, 10)	(1, 1, -1)	(1, -1, 1)	(1, 1, 1)

Reputation: **Witness Audit**

- All behaviors of a witness are recorded in the blockchain, which are trackable and immutable.
- The audit mechanism is leveraged to calculate the reputation value of a witness.
- If the reputation of a witness is too low, it would not be selected by the selection algorithm.
- The malicious or unrational witnesses can be:

  Lazy witness → someone prefers *not to report* the violation.

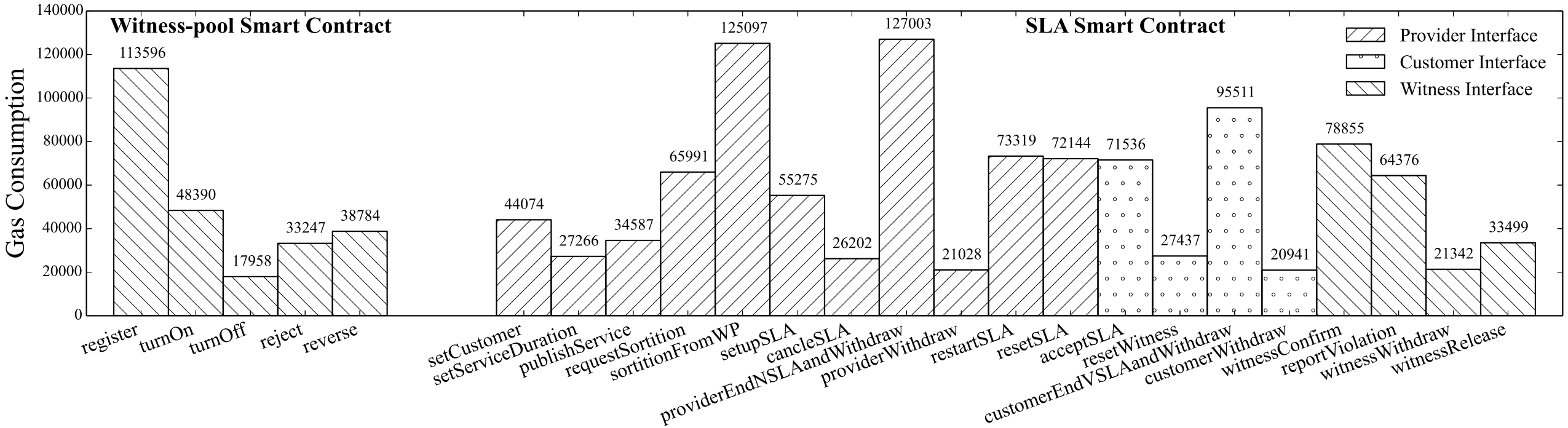
Implementation: **Ethereum**



- The entire system is implemented based on the **two** types of smart contracts
- Leverage **Solidity** to program smart contracts
- Code: https://github.com/zh9314/SmartContract4SLA

Experimental Study: **performance characteristics**

- We deploy the implemented smart contracts on the test net of Ethereum blockchain, ***Rinkeby***.
- We test all possible scenarios to exploit and validate the functionality of different interfaces.
- The **gas consumption** of an interface determines the **transaction fee** needed to pay the miner in Ethereum, when invoking that interface.
- The **more complex** of the interface is, the **more transaction fee** required when it is invoked.



Conclusions

1. A **decentralized witness model** is proposed for Cloud SLA enforcement;
2. **Witness-as-a-Game** for incentive: in order to maximize the rewards, a witness always **has to** offer honest monitoring service;
3. The trustworthiness is proved through **game theory**;
4. A **prototype system** is fully implemented based on Ethereum blockchain.

Future Work

- **Break the limitation** of this work: the witness can only provide Boolean value, “TRUE” or “FALSE”;
- Further **optimize** the interface implementation to reduce the gas consumption;
- Consider some **more** application **scenarios**, not only Cloud SLA;
- Develop **user-friendly tools** to interact with the smart contract.

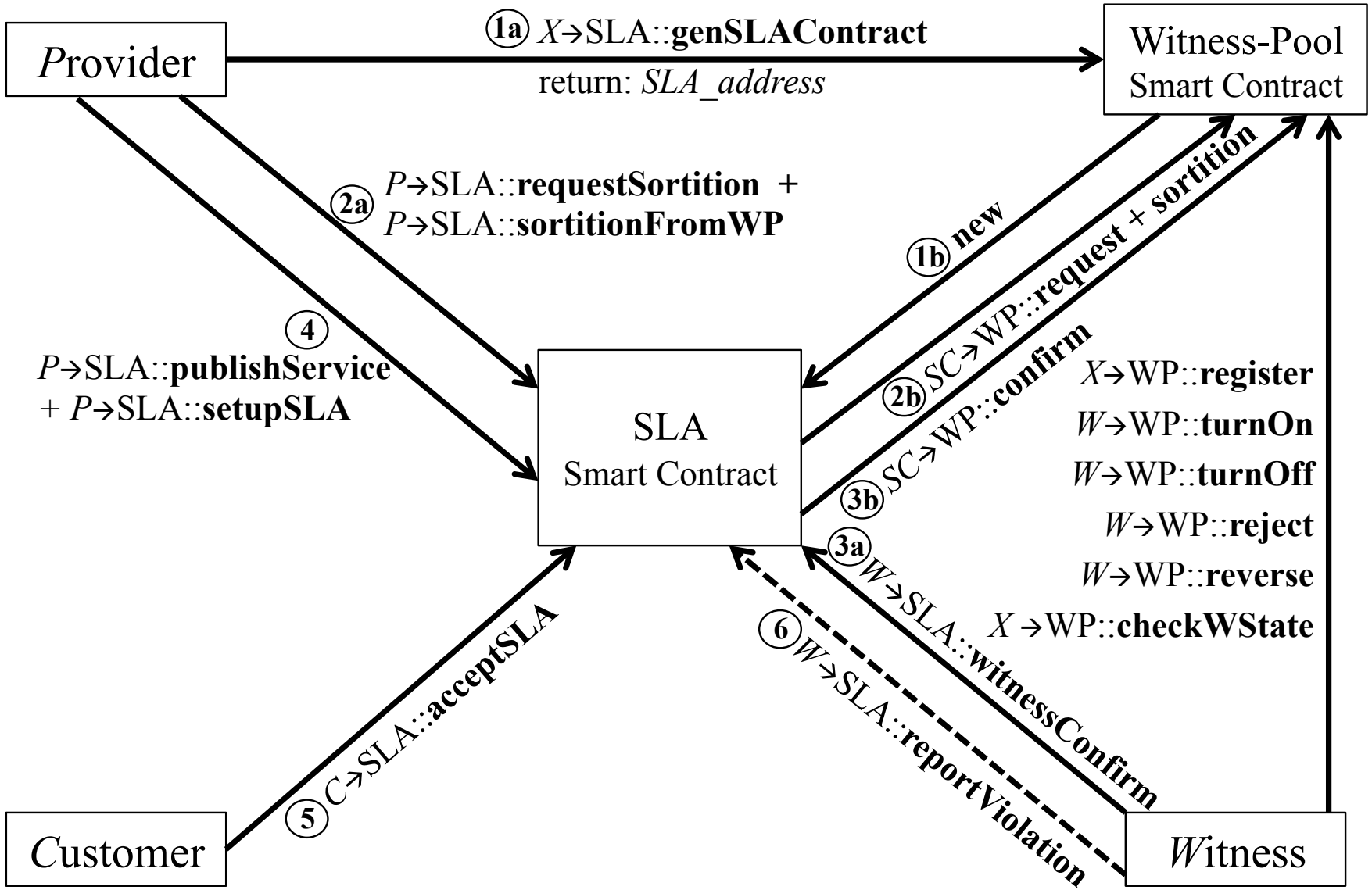
To the best of our knowledge, this is the first work of **trustworthy decentralized** oracles based on **economic principles (game theory).**

Thank you!

Questions?

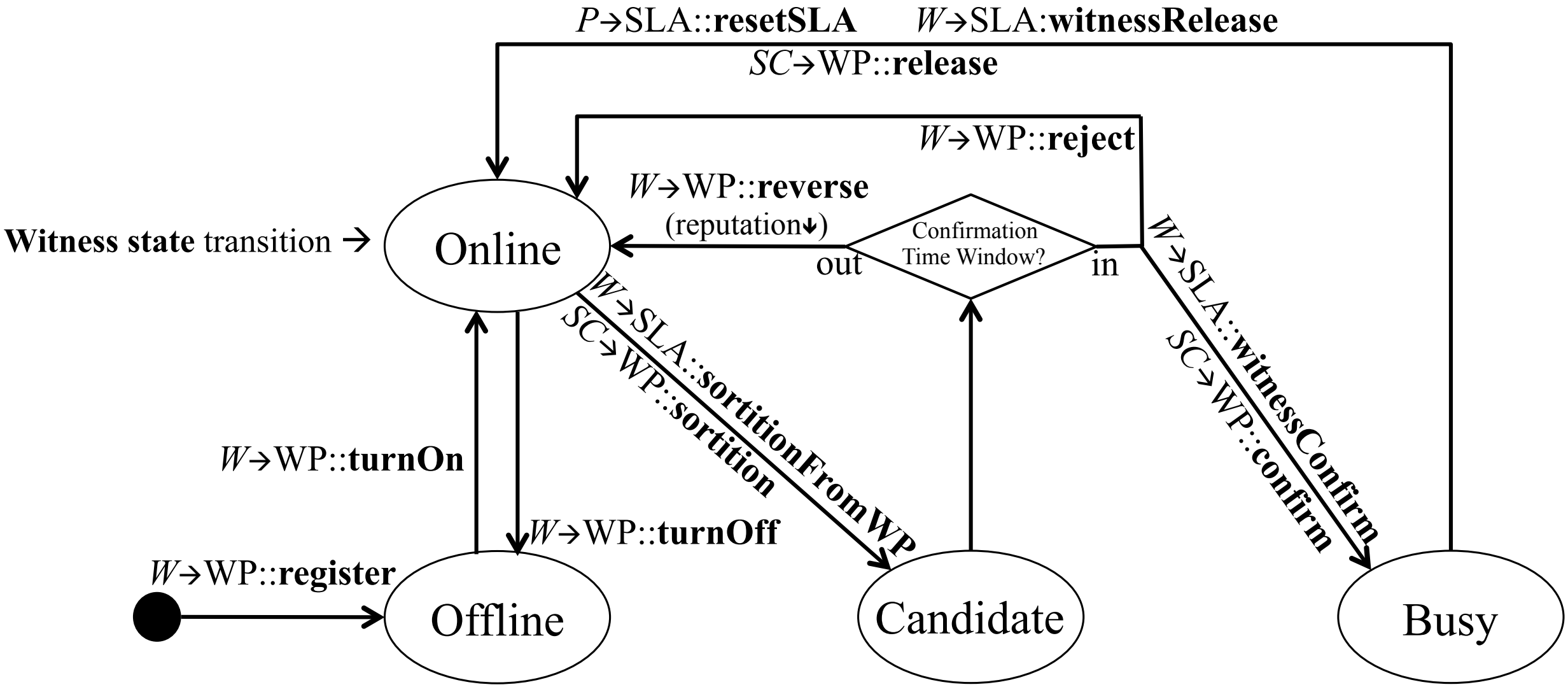
Backup Slides

Overall Relationship



Backup Slides

Witness-pool Smart Contract Implementation



Backup Slides

A Specific SLA Smart Contract Implementation