Welcome to ASCrypto 2025!

Opening Remarks

Organizers

Javier Verbel

Arantxa Zapico

Speakers

Sophia Yakoubov

Alan Szepieniec

Benedikt Bünz

Monday

Now-10:30: Introduction to Proof Systems

Coffee Break

11:00-12:30: Folding and Accumulation Schemes

Lunch break

14:00-15:30: Introduction to zk-STARKs

Coffee Break

16:00-17:30: Secure MPC and applications to zk proofs

Tuesday

9:00-10:30: Folding and Accumulation Schemes

Coffee Break

11:00-12:30: Introduction to zk-STARKs

Lunch break

14:00-15:30: Secure MPC and applications to zk proofs

Coffee Break

16:00-17:30: Q&A Practical Session

Funded Students

50 participants
65 applications for funding
30 students
11 countries

Thanks!

ecosystem support program

Microsoft

International Association for Cryptologic Research

CROSSING

Introduction to Proof Systems

Arantxa Zapico
Ethereum Foundation

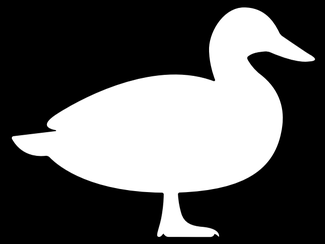
ASCRYPTO. Medellin, September 2025

This talk:

How to build SNARKs (Succinct Non-Interactive Arguments of Knowledge) or just SNARGs (without knowledge) from Interactive Proofs (what are Interactive Proofs?)

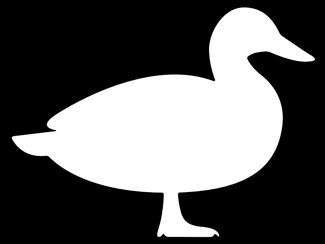
Interactive Proofs

Interactive Proofs

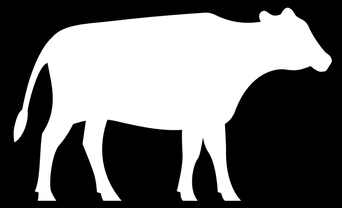


Prover

Interactive Proofs

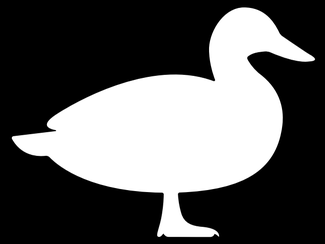


Prover

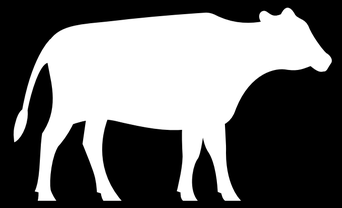


Verifier

Interactive Proofs

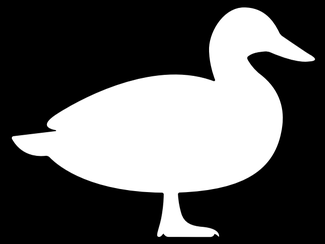


Peggy

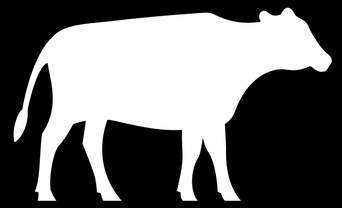


Victor

Interactive Proofs



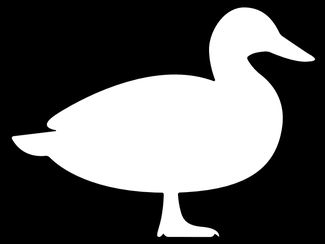
Pedrinho



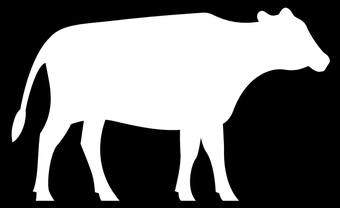
Valeria

Interactive Proofs

Something is true

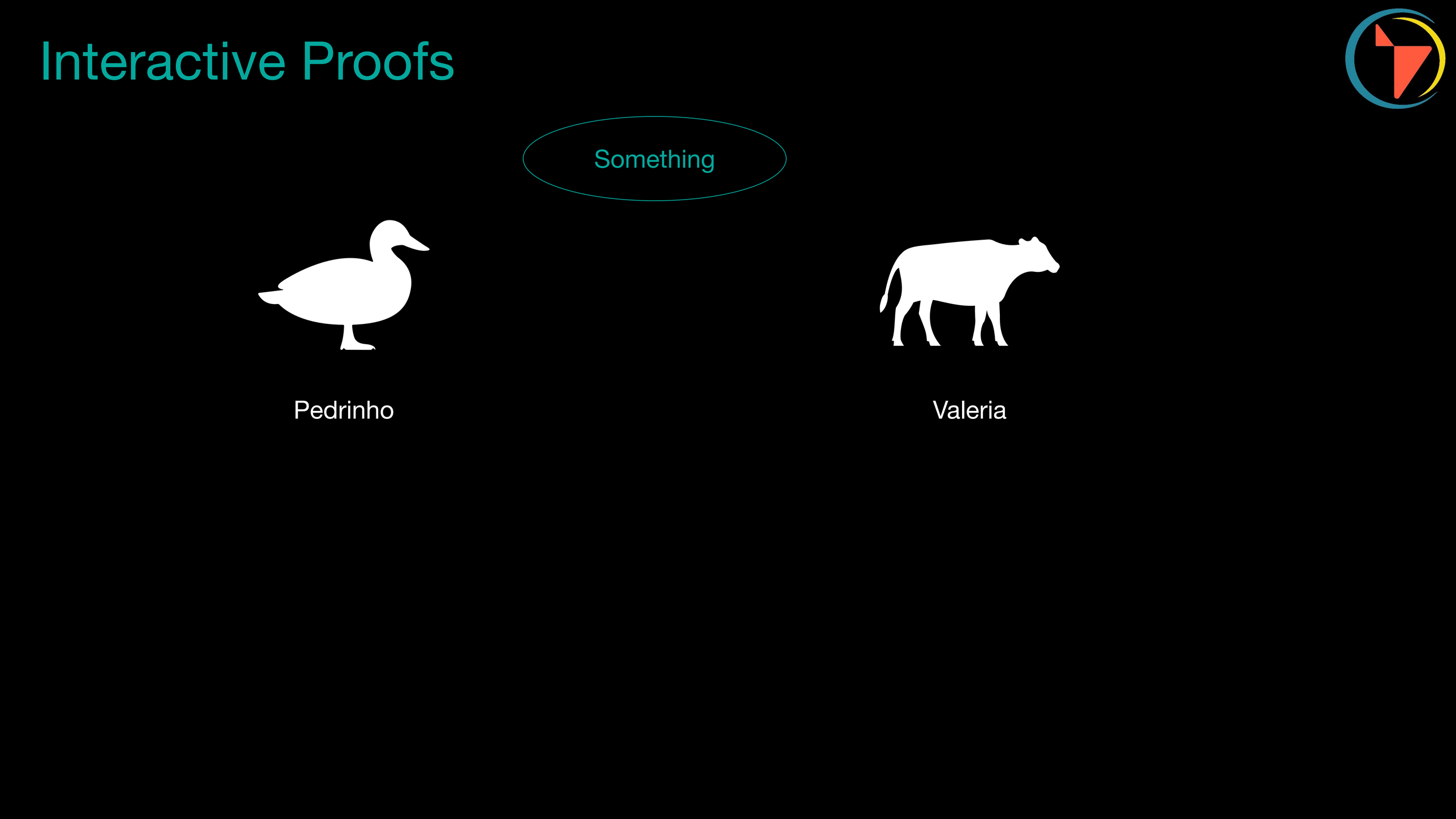


Pedrinho

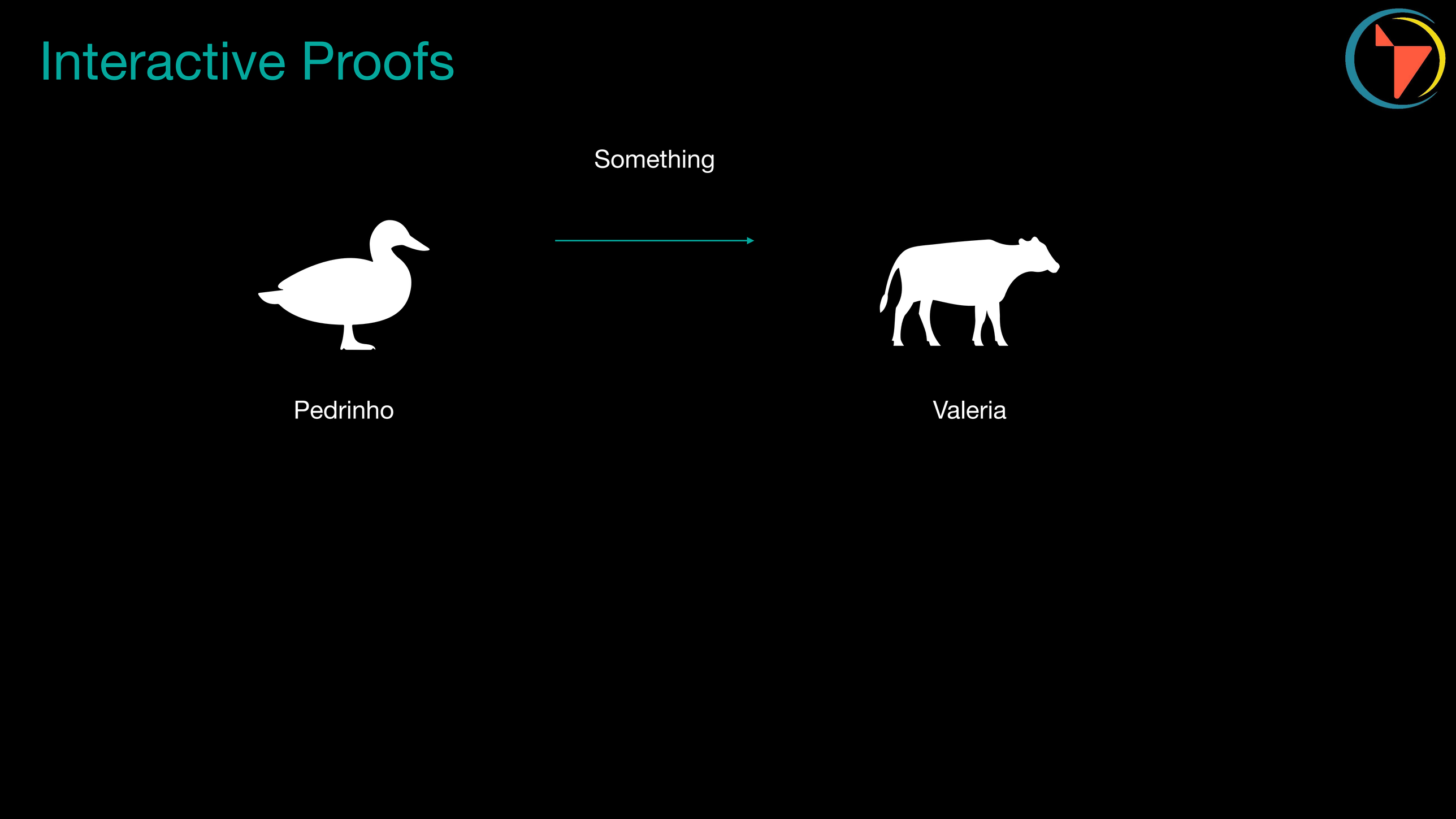


Valeria

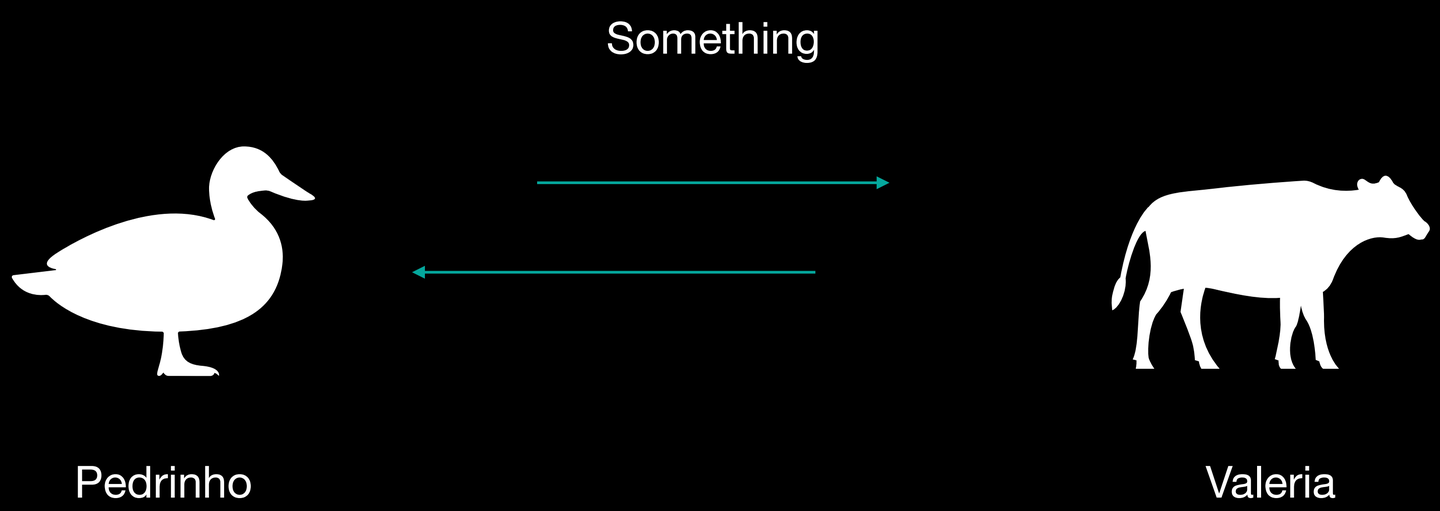
Interactive Proofs



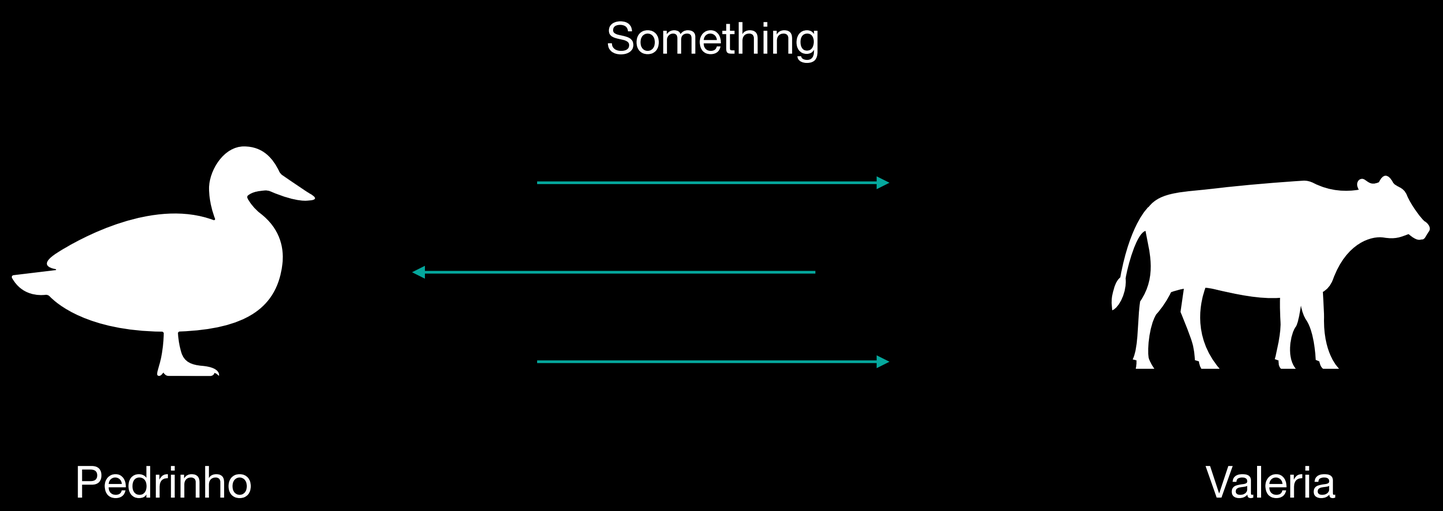
Interactive Proofs



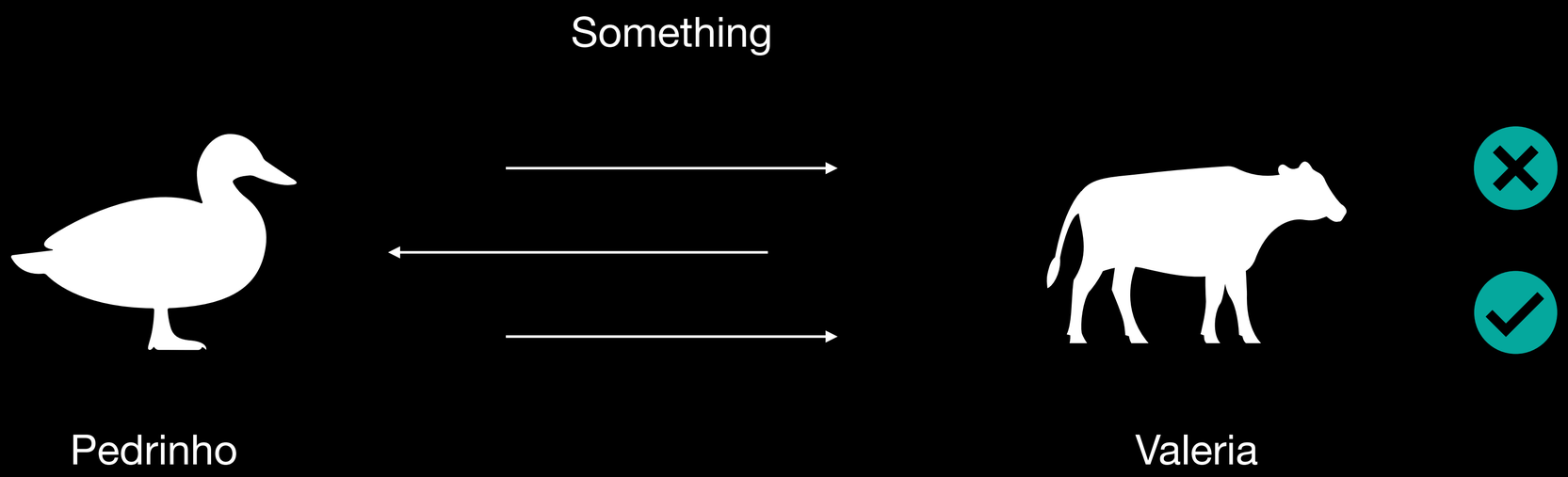
Interactive Proofs



Interactive Proofs

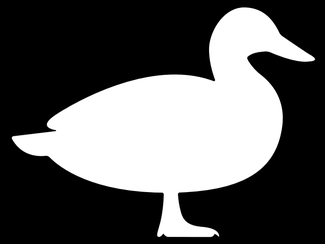


Interactive Proofs

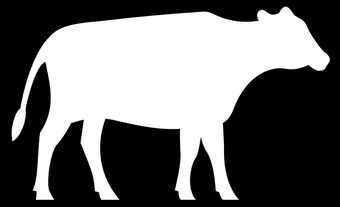


Examples of provers and verifiers

Examples of provers and verifiers

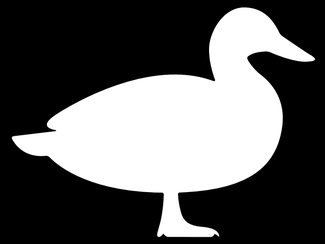


Me

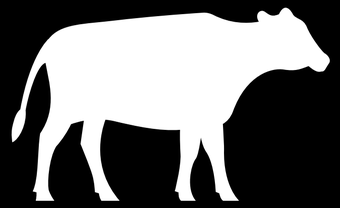


Gmail

Examples of provers and verifiers

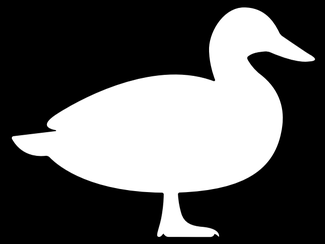


Google Cloud

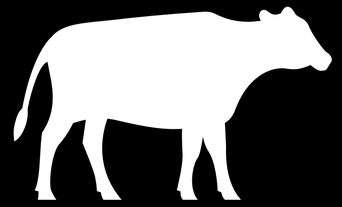


Mobil Phone

Examples of provers and verifiers

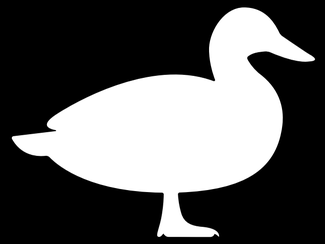


You

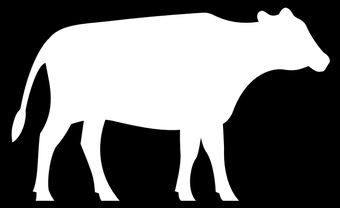


Security at Club

Examples of provers and verifiers

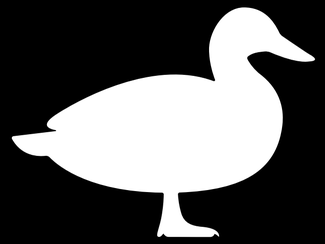


Cryptocurrency user

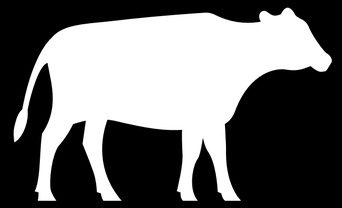


Block Builder

Examples of provers and verifiers

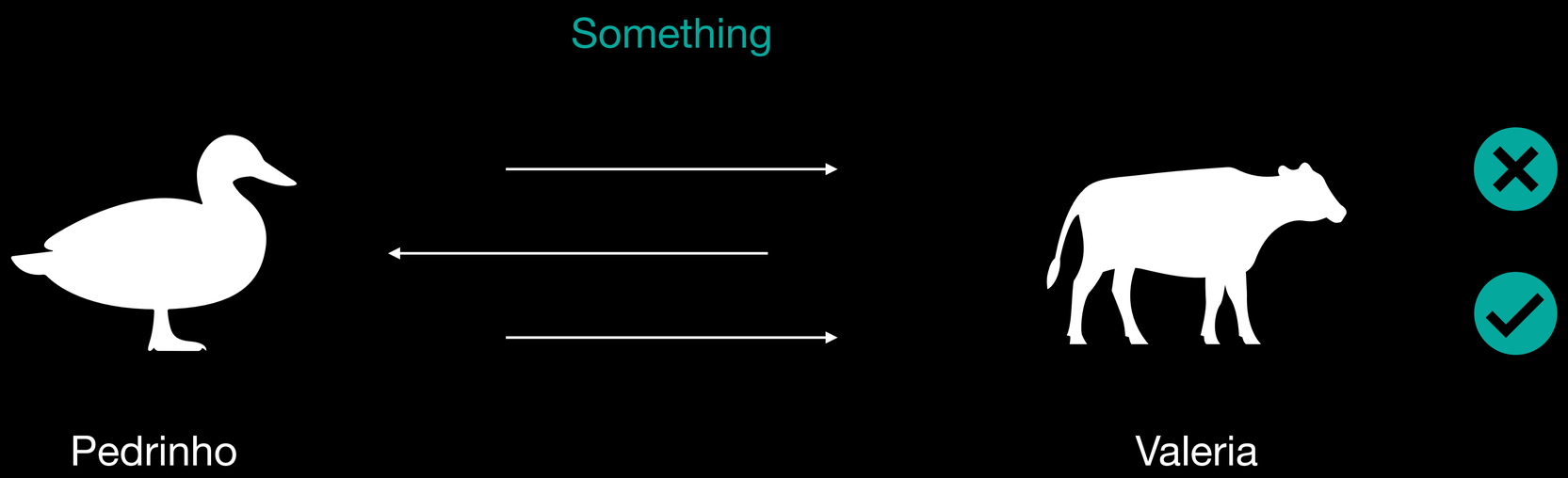


ZkVM

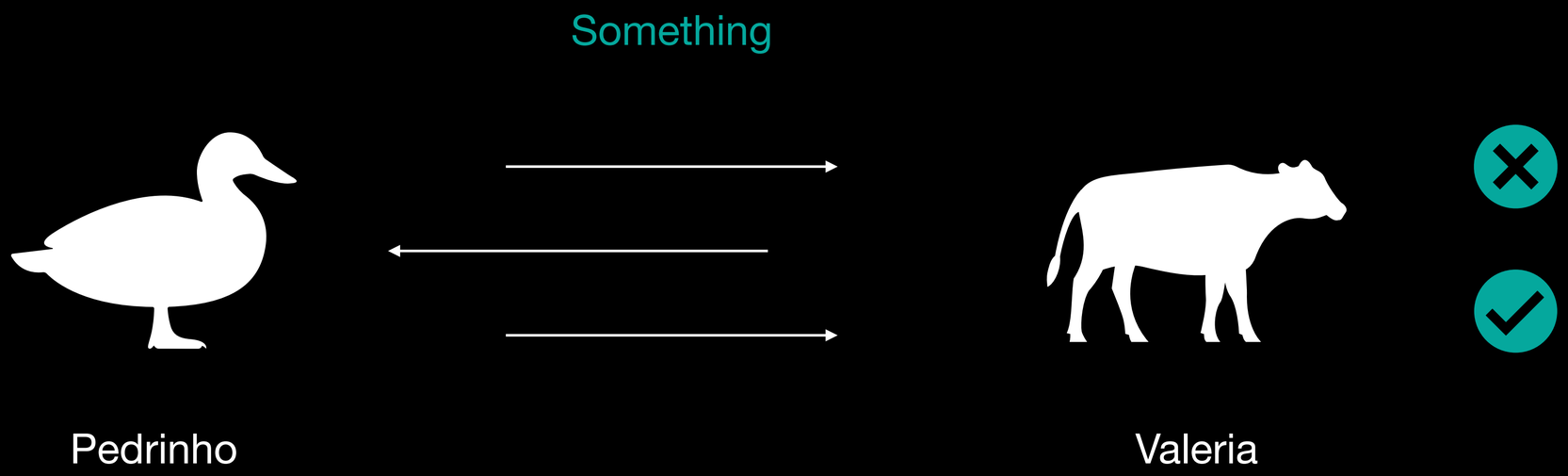


Smart Contract

Interactive Proofs

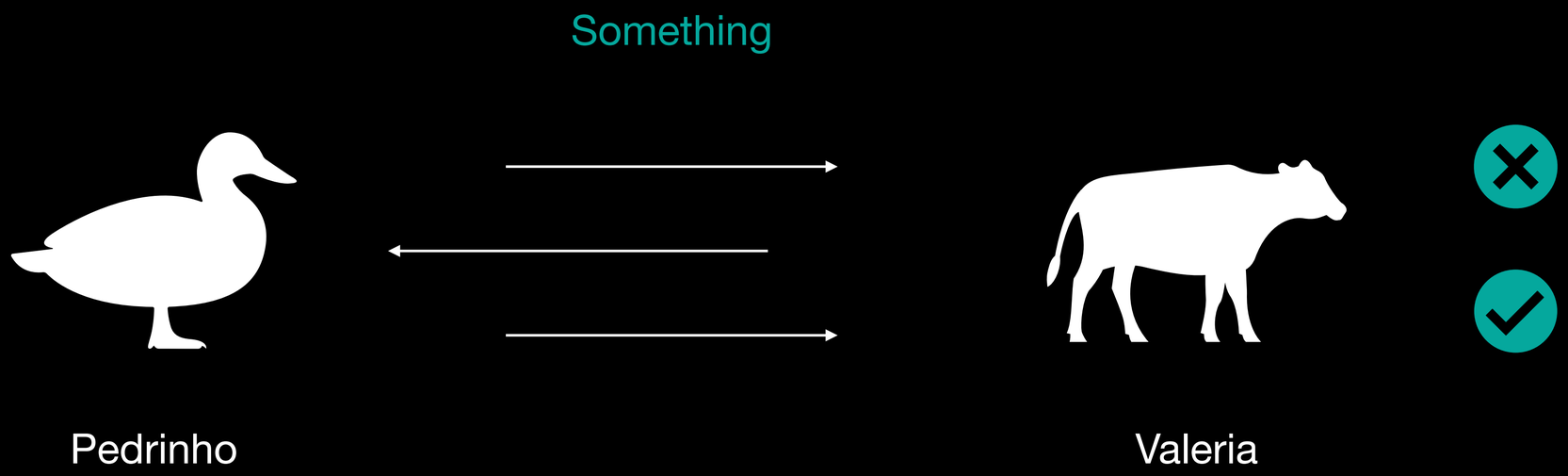


Interactive Proofs



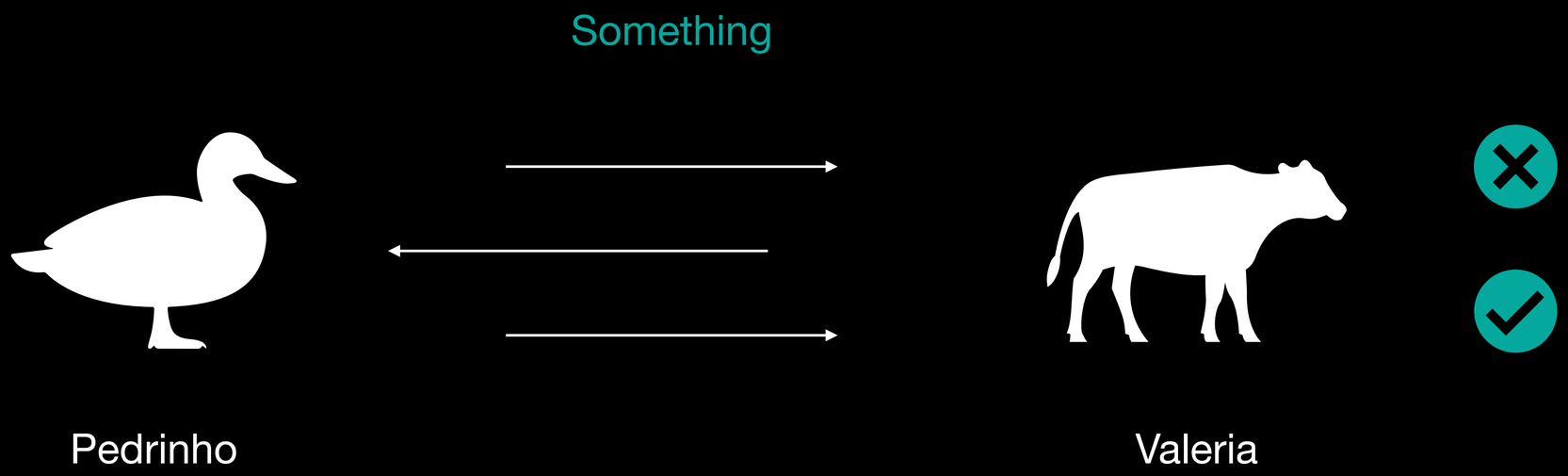
Completeness

Interactive Proofs



Completeness If Something is indeed true and both, Prover and Verifier, follow the procedure, Verifier accepts

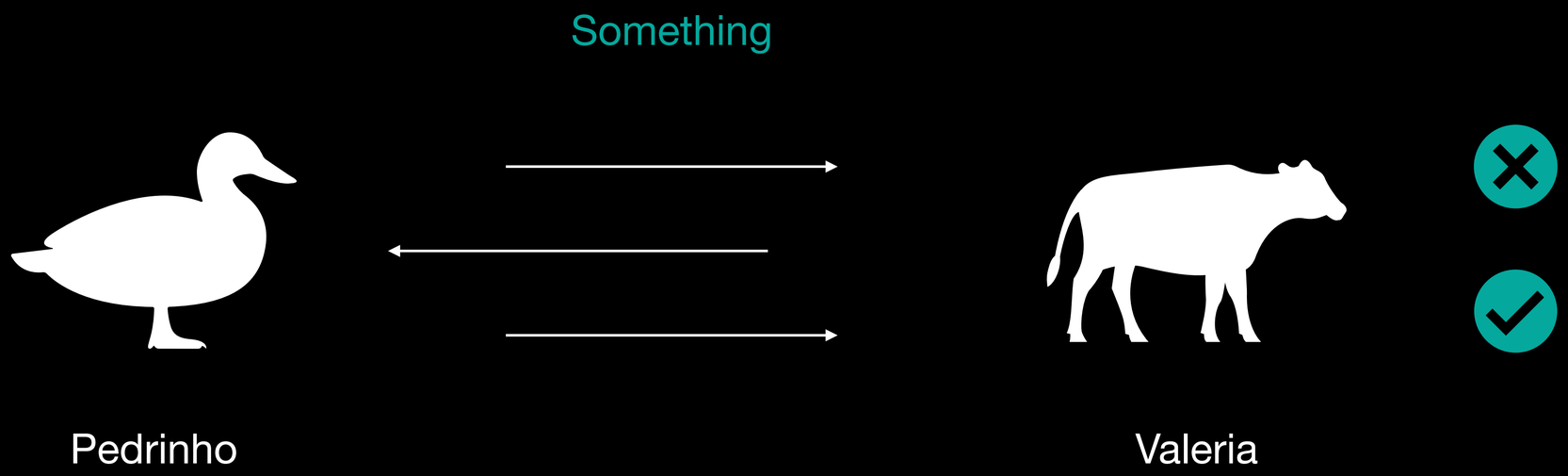
Interactive Proofs



Completeness If Something is indeed true and both, Prover and Verifier, follow the procedure, Verifier accepts

Soundness

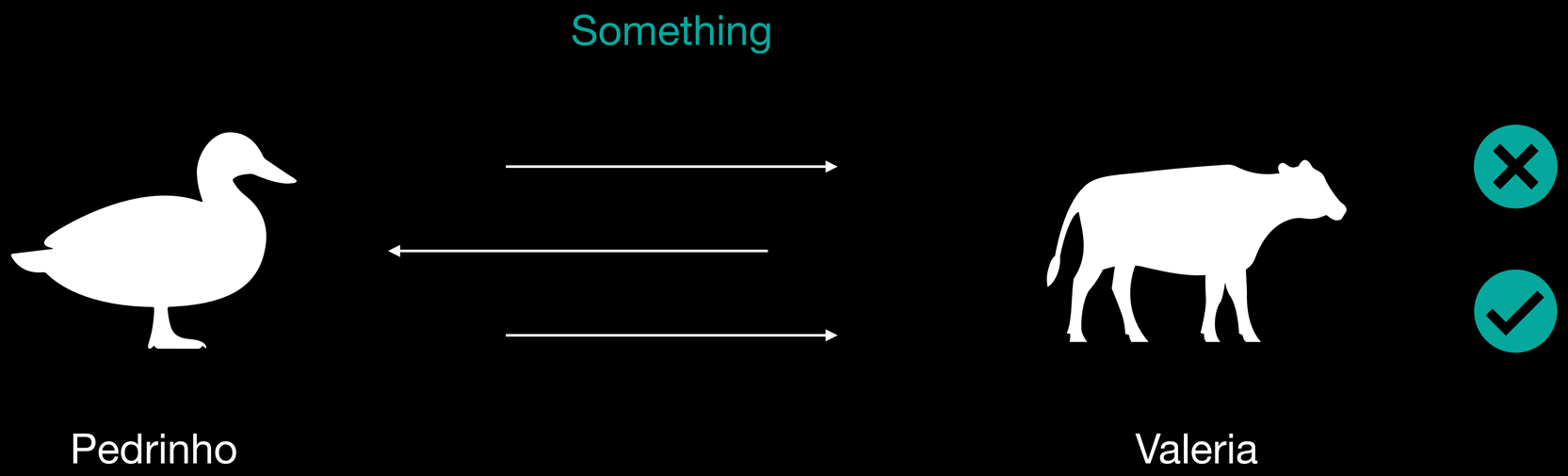
Interactive Proofs



Completeness If Something is indeed true and both, Prover and Verifier, follow the procedure, Verifier accepts

Soundness If *something* is false, then Verifier rejects with overwhelming probability

Interactive Proofs

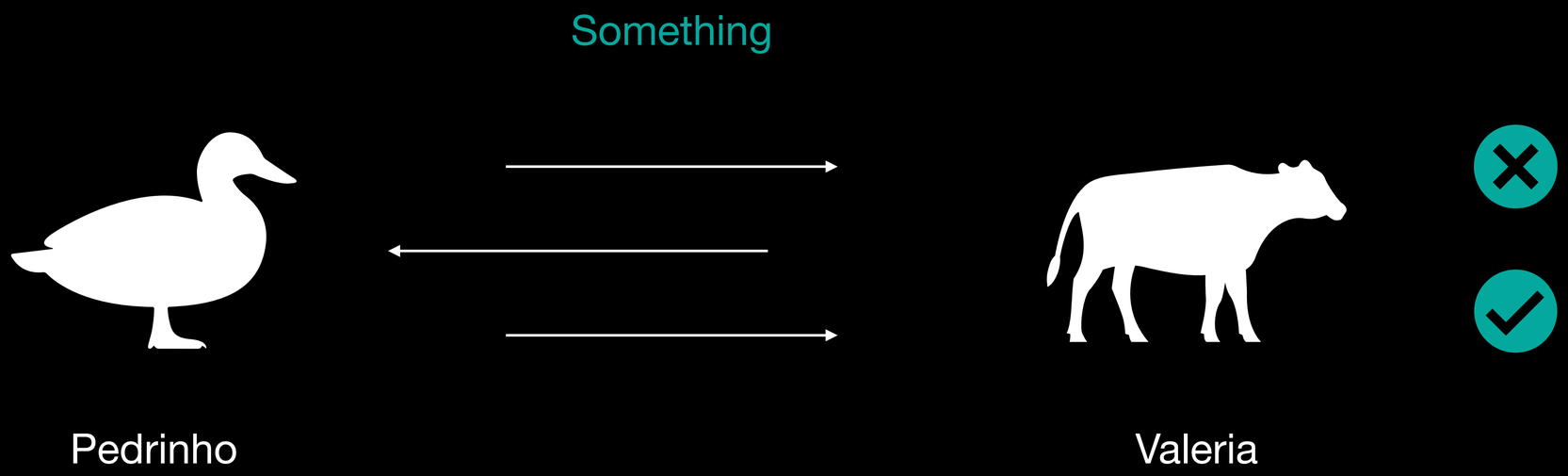


Completeness If Something is indeed true and both, Prover and Verifier, follow the procedure, Verifier accepts

Soundness If *something* is false, then Verifier rejects with overwhelming probability

Zero-Knowledge

Interactive Proofs



Completeness If Something is indeed true and both, Prover and Verifier, follow the procedure, Verifier accepts

Soundness If *something* is false, then Verifier rejects with overwhelming probability

Zero-Knowledge The Verifier does not learn anything but the truth of *Something*

Something

Something

R is a PT decidable relation

$R = \{(x, w) : \ldots\}$ is a PT decidable relation

$R = \{(x, w) : \ldots\}$ is a PT decidable relation

Something is true

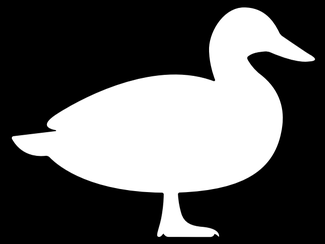
$R = \{(x, w) : \ldots\}$ is a PT decidable relation

$$x \in \mathscr{L}_R$$

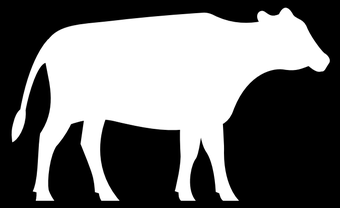
$R = \{(x, w) : \ldots\}$ is a PT decidable relation

$$x \in \mathscr{L}_R$$

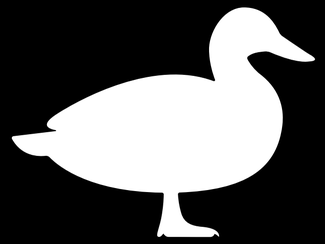
$$\mathscr{L}_R = \{x \ s \,.\, t \,.\, \exists w \ s \,.\, t \,.\, (x, w) \in R\}$$



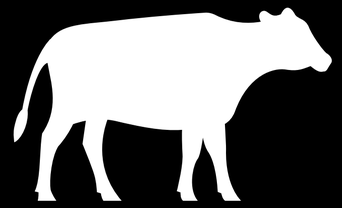
You



Security at Club

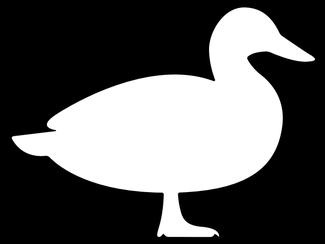


You

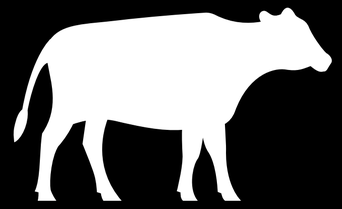


Security at Club

$$R = \{(x, w) : x \text{ is a name and } w \text{ an age above 18}\}$$



You



Security at Club

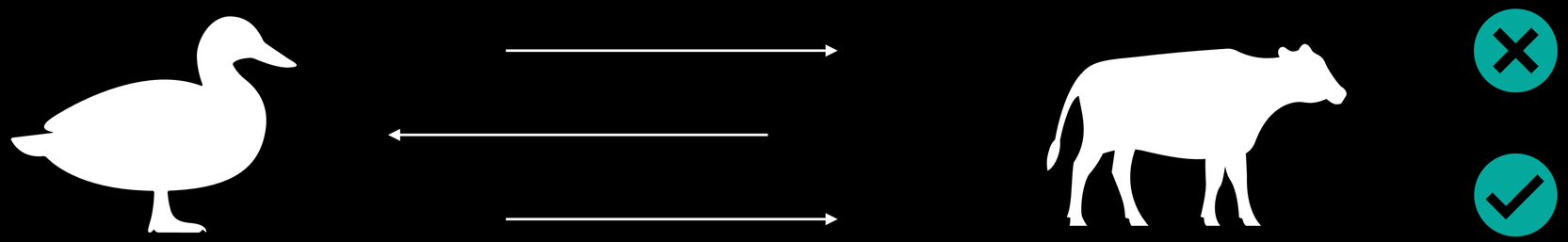
$$R = \{(x, w) : x \text{ is a name and } w \text{ an age above 18}\}$$

"I am in $\mathscr{L}_R$": there exists a w (my age) such that

(me, w) $\in R$

Something is true

Something



Pedrinho

Valeria

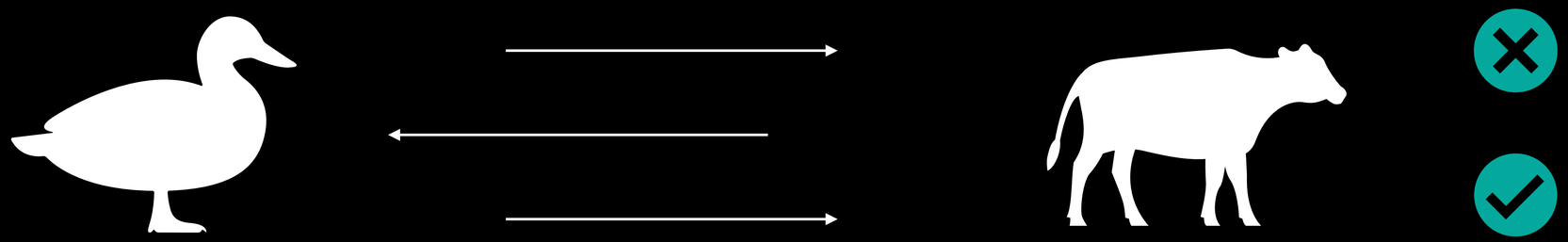
Completeness If Something is indeed true and both, Prover and Verifier, follow the procedure, Verifier accepts

Soundness If *something* is false, then Verifier rejects with overwhelming probability

Zero-Knowledge The Verifier does not learn anything but the truth of *Something*

$$R = \{(x, w) : something\}$$

Something



Pedrinho

Valeria

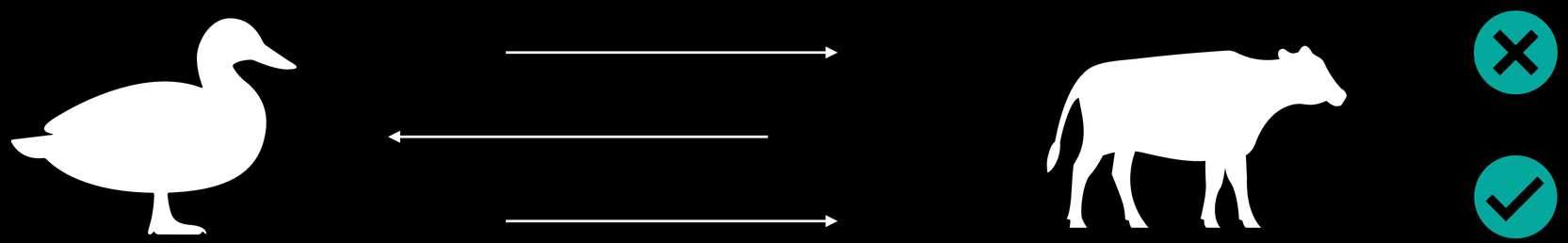
Completeness If Something is indeed true and both, Prover and Verifier, follow the procedure, Verifier accepts

Soundness If *something* is false, then Verifier rejects with overwhelming probability

Zero-Knowledge The Verifier does not learn anything but the truth of *Something*

$$R = \{(x, w) : something\}$$

R, x



Pedrinho

Valeria

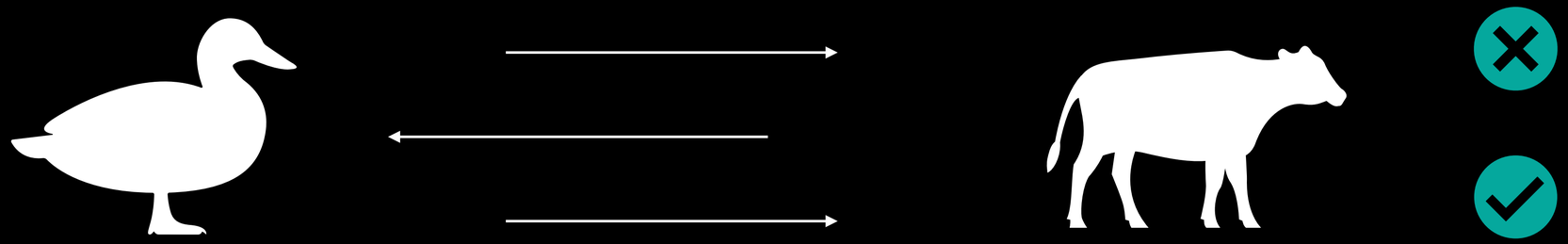
Completeness If Something is indeed true and both, Prover and Verifier, follow the procedure, Verifier accepts

Soundness If *something* is false, then Verifier rejects with overwhelming probability

Zero-Knowledge The Verifier does not learn anything but the truth of *Something*

$$R = \{(x, w) : something\}$$

$$pp \leftarrow \mathcal{K}, x$$



Pedrinho$(pp, (x, w))$

Valeria (pp, x)

Completeness If Something is indeed true and both, Prover and Verifier, follow the procedure, Verifier accepts

Soundness If *something* is false, then Verifier rejects with overwhelming probability

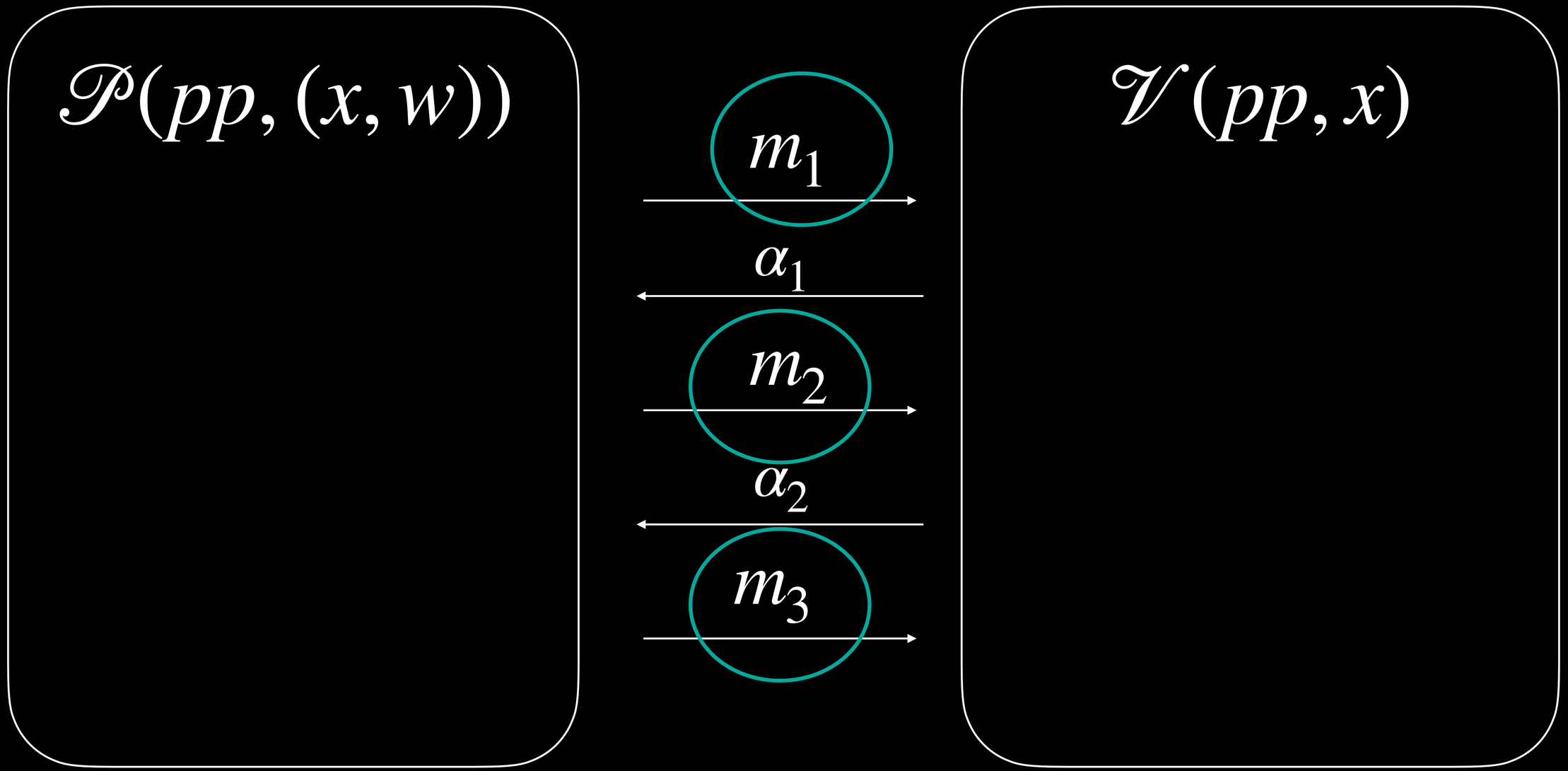
Zero-Knowledge The Verifier does not learn anything but the truth of *Something*

Probabilistic Polynomial Time Algorithms

$\mathcal{P}(pp, (x, w))$

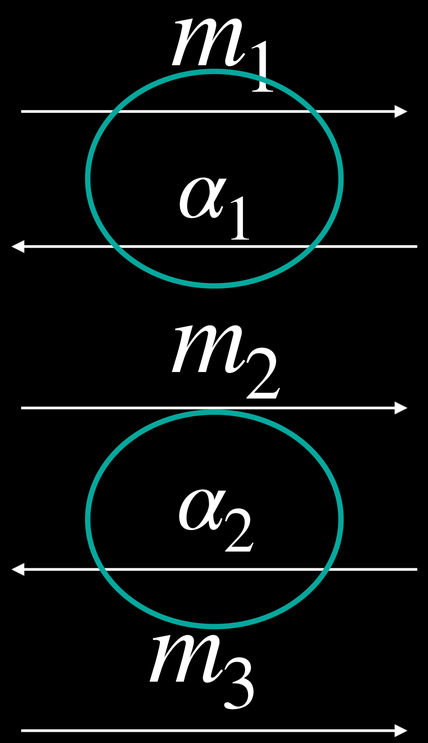
$\mathcal{V}(pp, x)$

Probabilistic Polynomial Time Algorithms



Probabilistic Polynomial Time Algorithms

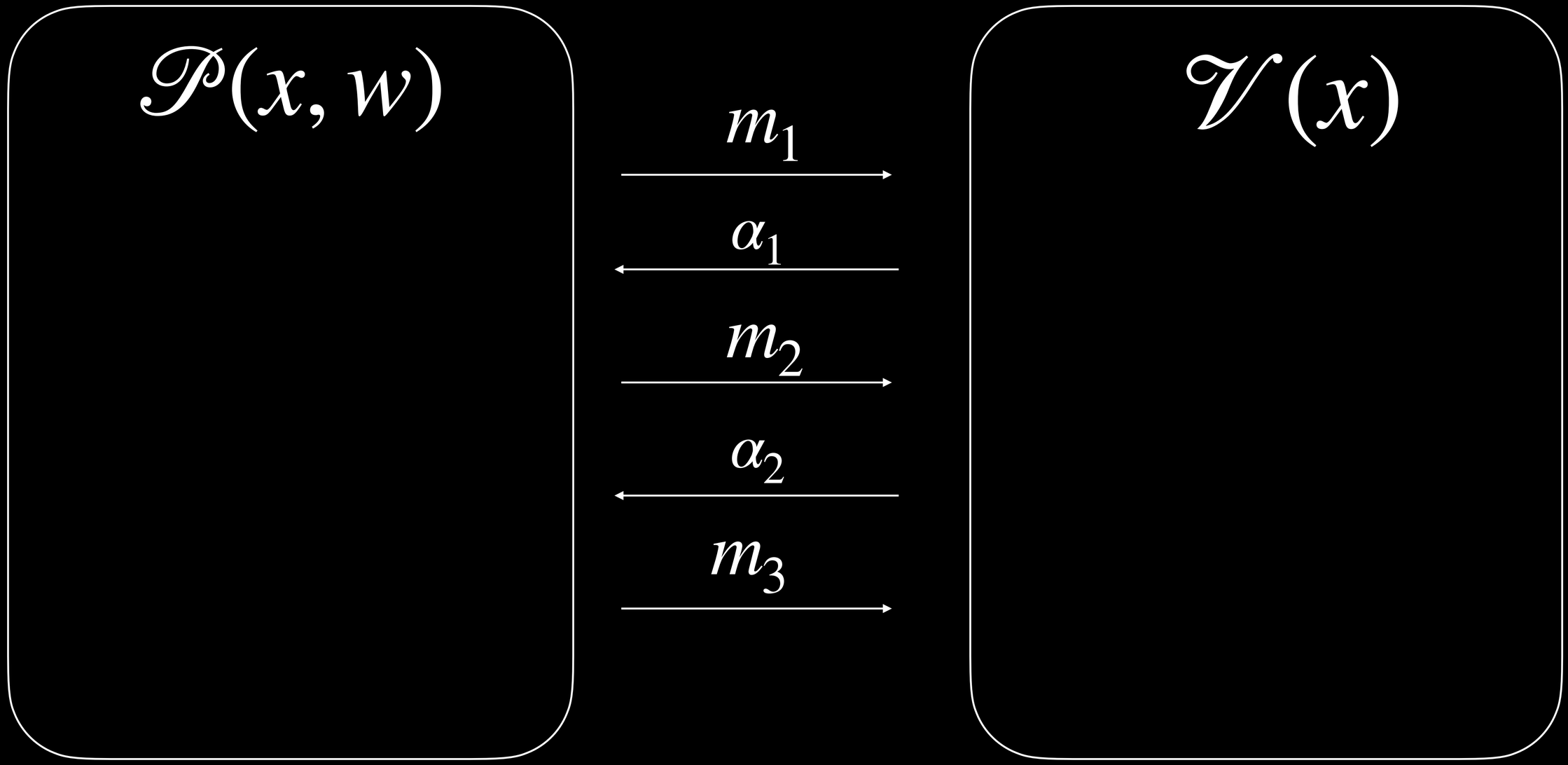
$\mathcal{P}(pp, (x, w))$



$\mathcal{V}(pp, x)$

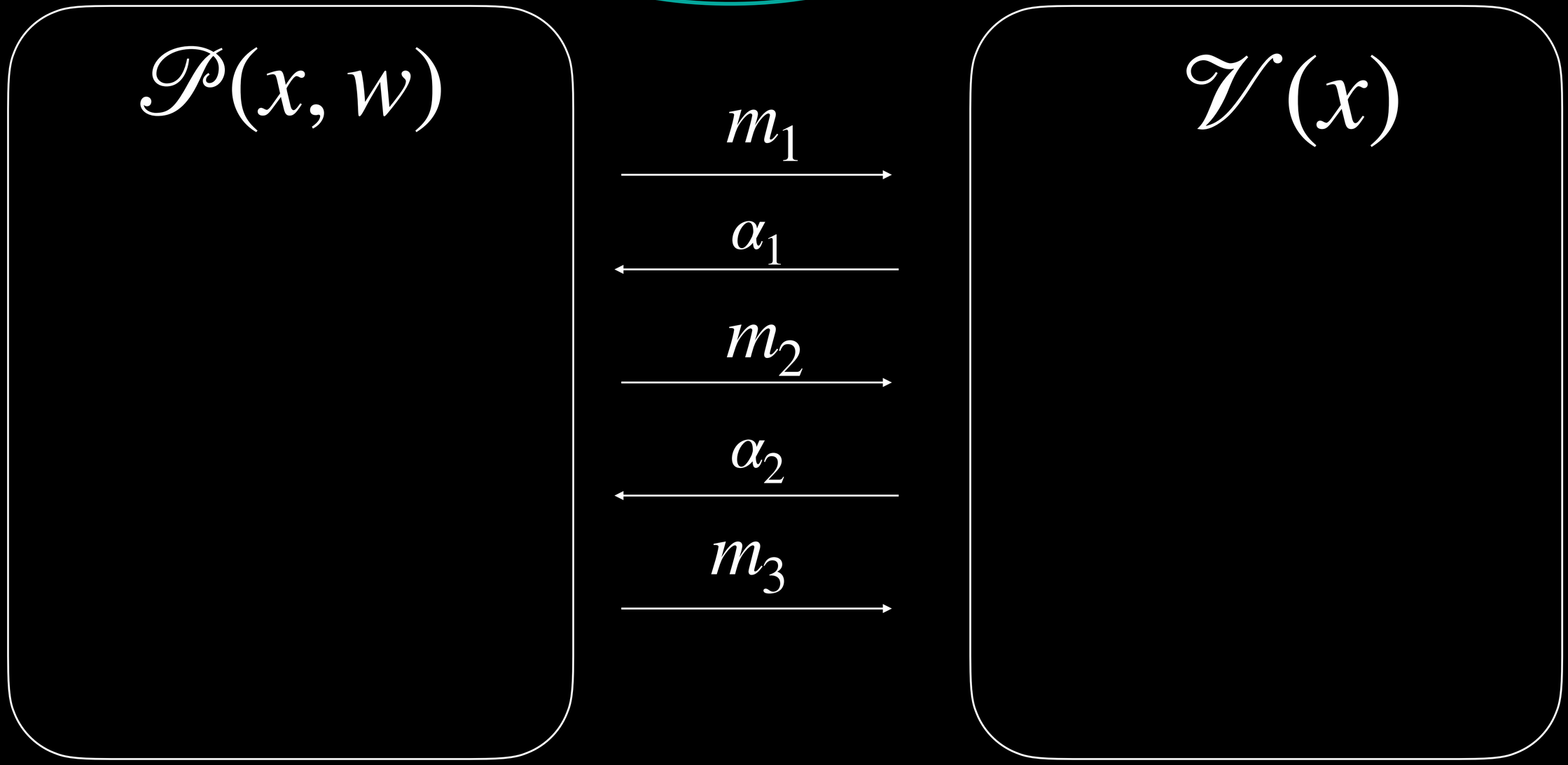
Verifier is public coin

Efficiency

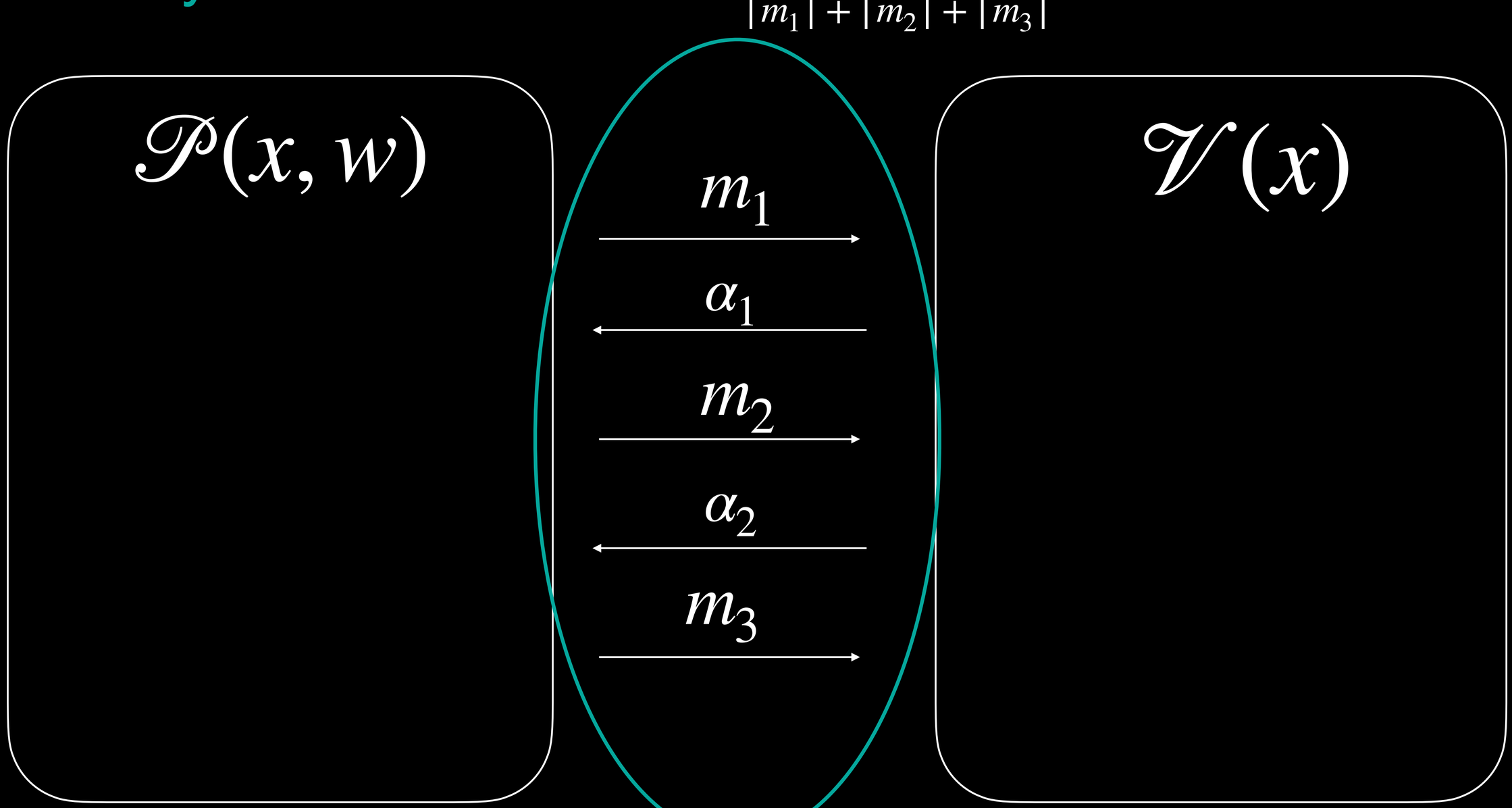


Efficiency

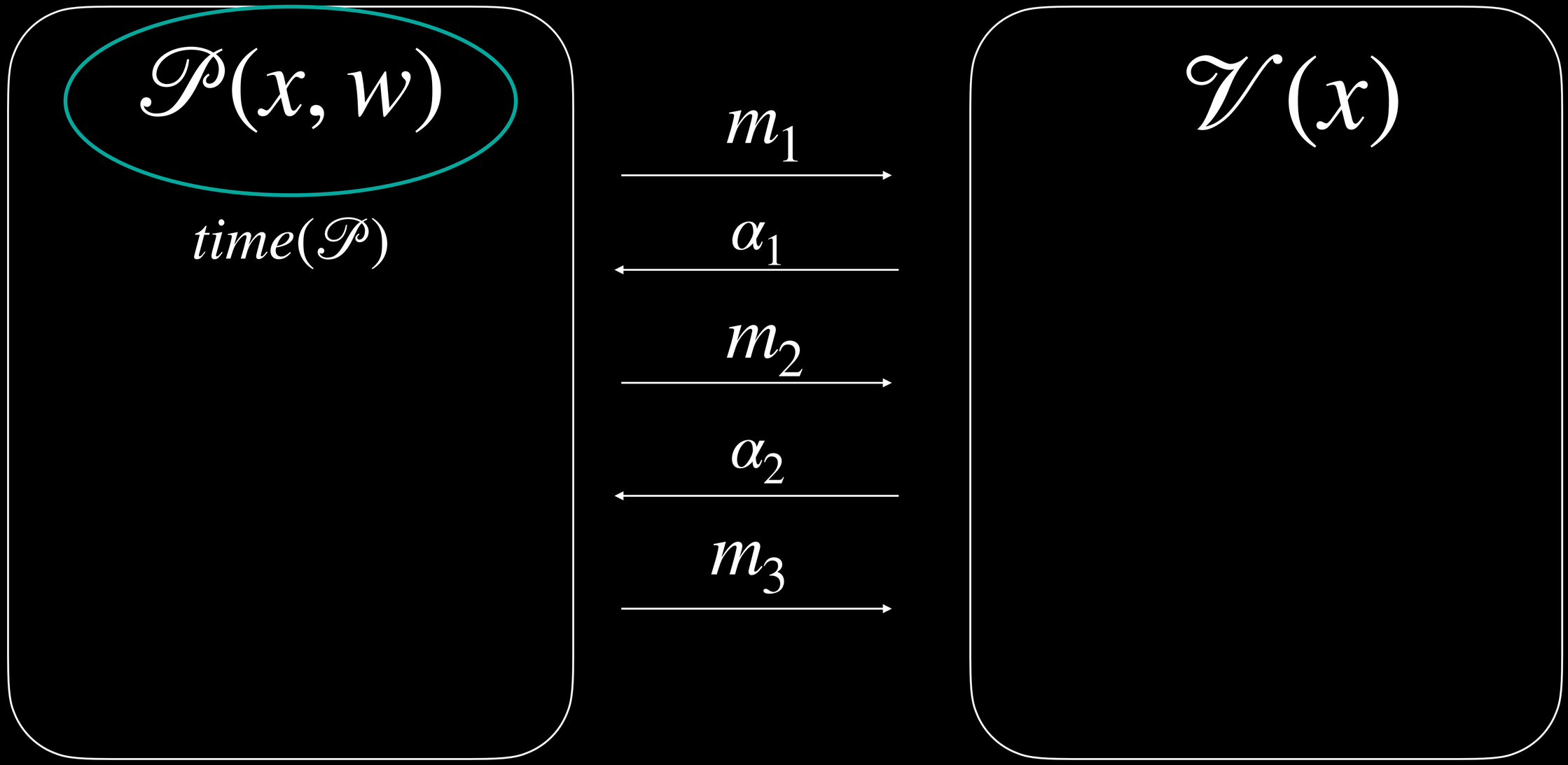
$pp \leftarrow \mathcal{K}$



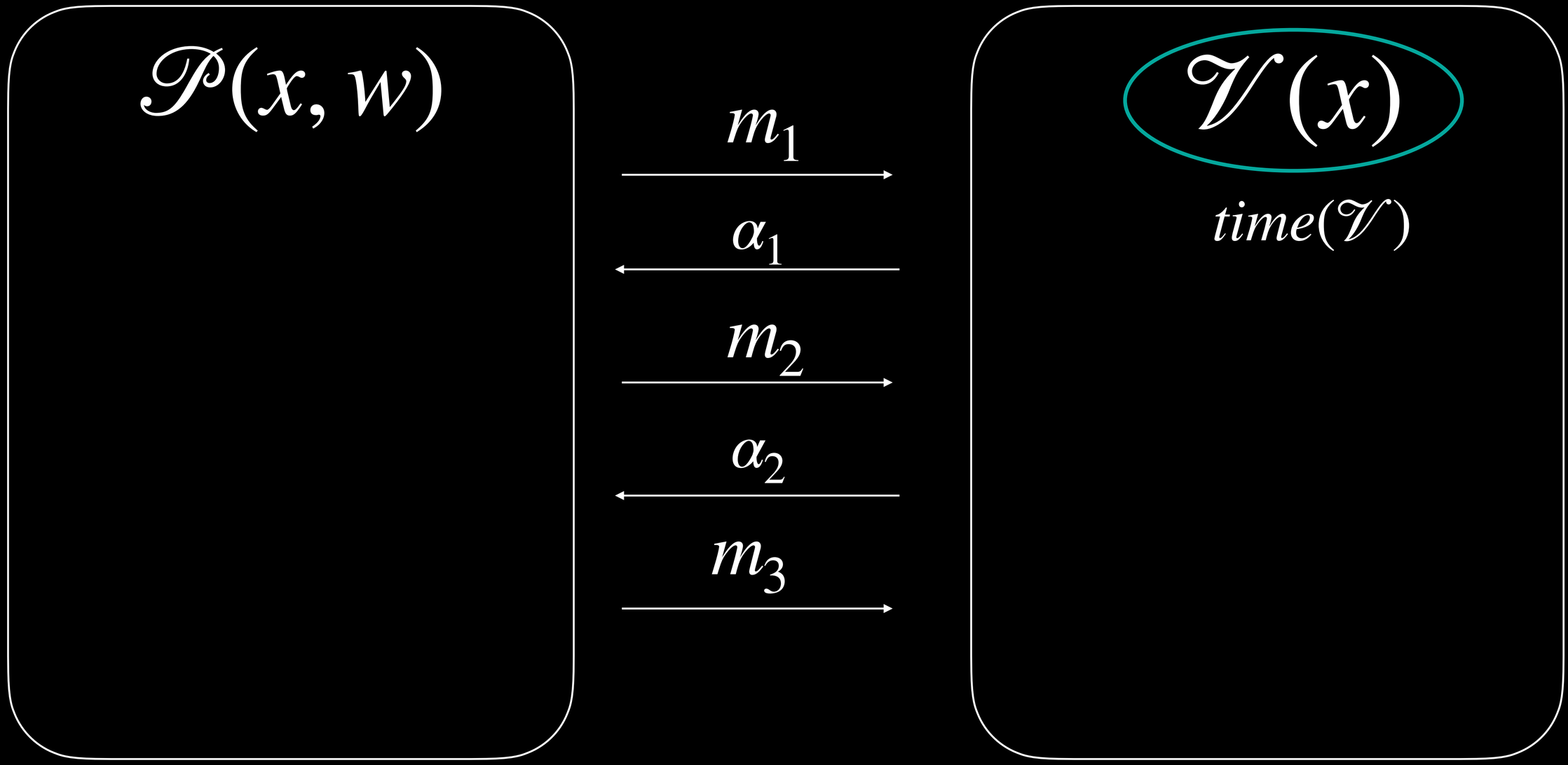
Efficiency: Proof Size



Efficiency: Prover time



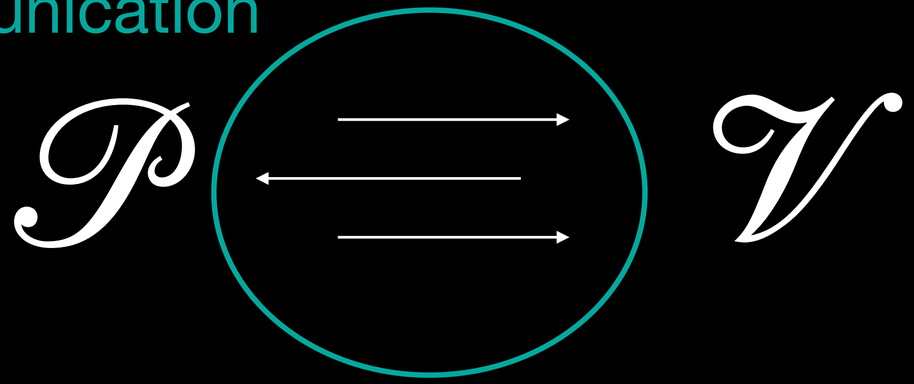
Efficiency: Verifier Time



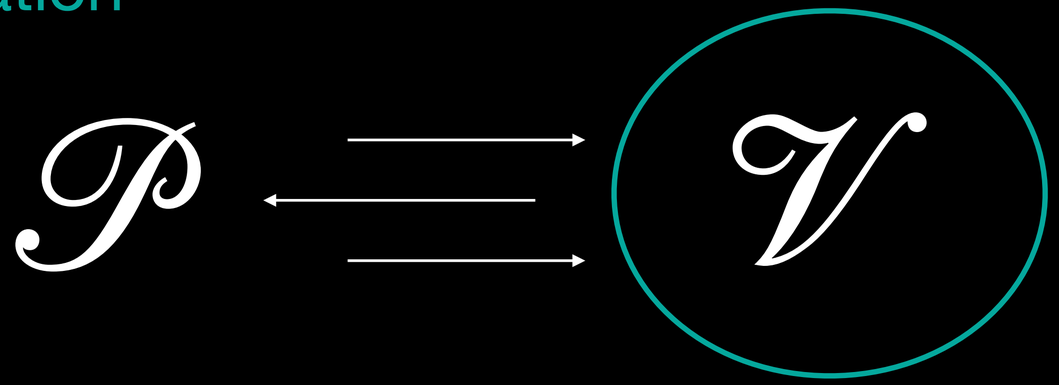
Succinctness

Succinctness

In communication



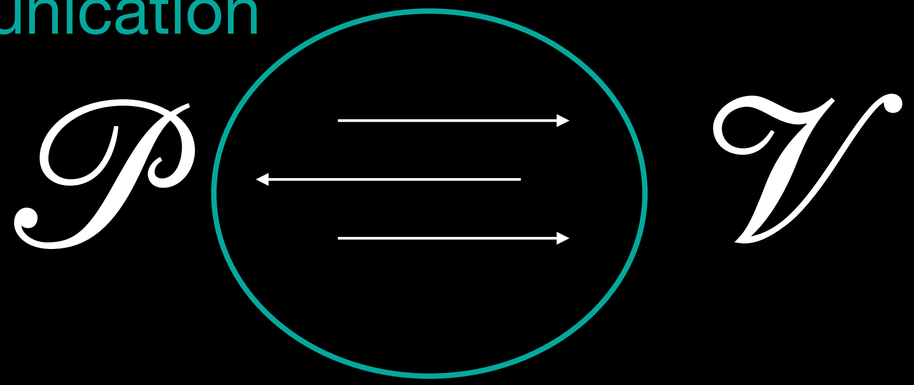
In verification



Succinctness

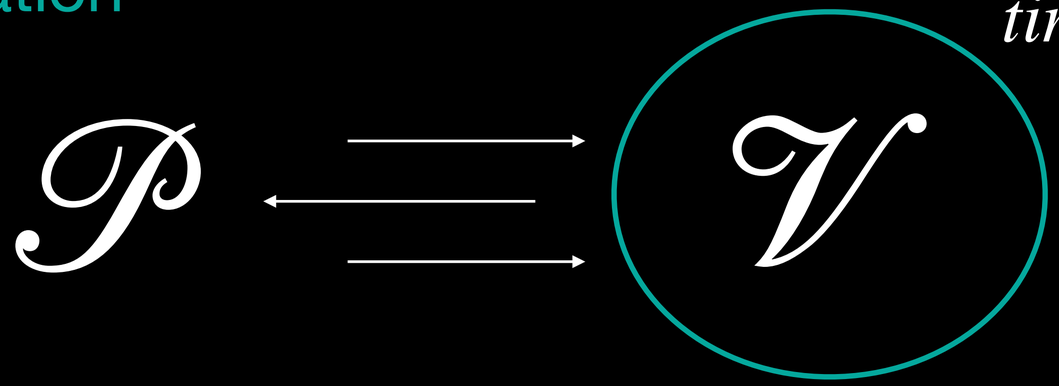
$$|m_1| + |m_2| + |m_3| << |w|$$

In communication



In verification

$$time(\mathscr{V}) << time_R(x, w)$$



(Perfect) Completeness

(Perfect) Completeness

If Something is indeed true and both, Prover and Verifier, follow the procedure, Verifier accepts

(Perfect) Completeness

If $x \in \mathscr{L}_R$ and both, Prover and Verifier, follow the procedure, Verifier accepts

(Perfect) Completeness

If $x \in \mathscr{L}_R$ and both, Prover and Verifier, follow the procedure, Verifier accepts

$$Pr\left[\langle \mathscr{P}(pp,(x,w)), \mathscr{V}(pp,x))\rangle\right] = 1$$

(Perfect) Completeness

If $x \in \mathscr{L}_R$ and both, Prover and Verifier, follow the procedure, Verifier accepts

$$Pr\left[\langle \mathscr{P}(pp,(x,w)), \mathscr{V}(pp,x))\rangle\right] = 1$$

Completeness If Something is indeed true and both, Prover and Verifier, follow the procedure, Verifier accepts

Soundness If *something* is false, then Verifier rejects with overwhelming probability

Zero-Knowledge The Verifier does not learn anything but the truth of *Something*

Completeness $$Pr\left[\langle \mathscr{P}(pp,(x,w),\mathscr{V}(pp,x))\rangle\right] = 1$$

Soundness If *something* is false, then Verifier rejects with overwhelming probability

Zero-Knowledge The Verifier does not learn anything but the truth of *Something*

(Computational) Soundness

If *something* is false, then Verifier rejects with overwhelming probability

(Computational) Soundness

If $x \notin \mathscr{L}_R$, then Verifier rejects with overwhelming probability

(Computational) Soundness

If $x \notin \mathcal{L}_R$, then Verifier rejects with overwhelming probability

If $\nexists\ w\ s.t.\ (x, w) \in R$, then Verifier rejects with overwhelming probability

(Computational) Soundness

If $x \notin \mathcal{L}_R$, then Verifier rejects with overwhelming probability

If $\nexists\ w\ s.t.\ (x, w) \in R$, then Verifier rejects with overwhelming probability

$$Pr\left[\langle \mathcal{P}^*(pp, x), \mathcal{V}(pp, x)\rangle\right] \leq negl(\lambda)$$

(Computational) Soundness

If $x \notin \mathscr{L}_R$, then Verifier rejects with overwhelming probability

If $\nexists\ w\ s.t.\ (x,w) \in R$, then Verifier rejects with overwhelming probability

$$Pr\left[\langle \mathscr{P}^*(pp,x), \mathscr{V}(pp,x)\rangle\right] \leq negl(\lambda)$$

Arguments

(Computational) Soundness

If $x \notin \mathcal{L}_R$, then Verifier rejects with overwhelming probability

If $\nexists\ w\ s.t.\ (x, w) \in R$, then Verifier rejects with overwhelming probability

$$Pr\left[\langle \mathcal{P}^*(pp, x), \mathcal{V}(pp, x)\rangle\right] \leq negl(\lambda)$$

We are actually talking about arguments

Completeness $$Pr\left[\langle \mathscr{P}(pp,(x,w),\mathscr{V}(pp,x))\rangle\right] = 1$$

Soundness If *something* is false, then Verifier rejects with overwhelming probability

Zero-Knowledge The Verifier does not learn anything but the truth of *Something*

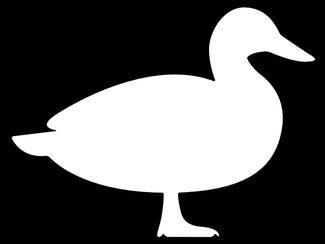
Completeness $$Pr\left[\langle \mathscr{P}(pp,(x,w),\mathscr{V}(pp,x))\rangle\right] = 1$$

Soundness $$Pr\left[\langle \mathscr{P}^*(pp,x),\mathscr{V}(pp,x)\rangle\right] \leq negl(\lambda)$$

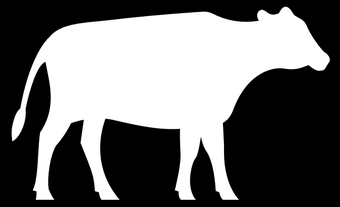
Zero-Knowledge

The Verifier does not learn anything but the truth of *Something*

Examples of provers and verifiers

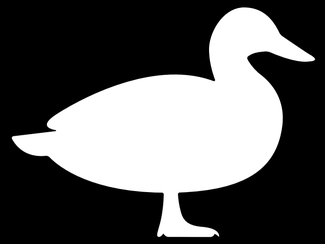


Me

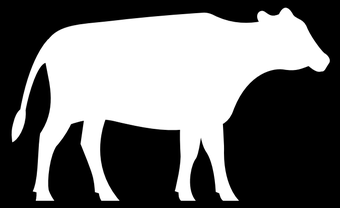


Gmail

Examples of provers and verifiers



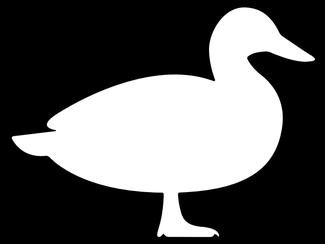
Me



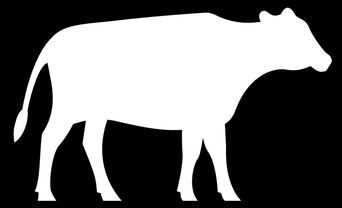
Gmail

There exists a password for this email address

Examples of provers and verifiers



Me



Gmail

There exists a password for this email address

Not enough!!!
I should *know* it

Arguments of Knowledge

Knowledge-soundness

There exists a PT algorithm $\mathscr{E}$, the extractor, such that for every malicious prover $\mathscr{P}^*$:

Knowledge-soundness

There exists a PT algorithm $\mathscr{E}$, the extractor, such that for every malicious prover $\mathscr{P}^*$:

$$Pr\left[(x,w) \in R : w \leftarrow \mathscr{E}^{\mathscr{P}^*}(x)\right] - Pr\left[\langle \mathscr{P}^*(x), \mathscr{V}(x)\rangle = 1\right] \leq negl(\lambda)$$

Knowledge-soundness

There exists a PT algorithm $\mathscr{E}$, the extractor, such that for every malicious prover $\mathscr{P}^*$:

$$Pr\left[(x,w) \in R : w \leftarrow \mathscr{E}^{\mathscr{P}^*}(x)\right] - Pr\left[\langle \mathscr{P}^*(x), \mathscr{V}(x)\rangle = 1\right] \leq negl(\lambda)$$

Knowledge-soundness

There exists a PT algorithm $\mathscr{E}$, the extractor, such that for every malicious prover $\mathscr{P}^*$:

$$Pr\left[(x,w) \in R : w \leftarrow \mathscr{E}^{\mathscr{P}^*}(x)\right] - Pr\left[\langle \mathscr{P}^*(x), \mathscr{V}(x)\rangle = 1\right] \leq negl(\lambda)$$

An argument that satisfies knowledge-soundness is an argument of knowledge

Completeness $$Pr\left[\langle \mathscr{P}(pp,(x,w),\mathscr{V}(pp,x))\rangle\right] = 1$$

Knowledge-Soundness

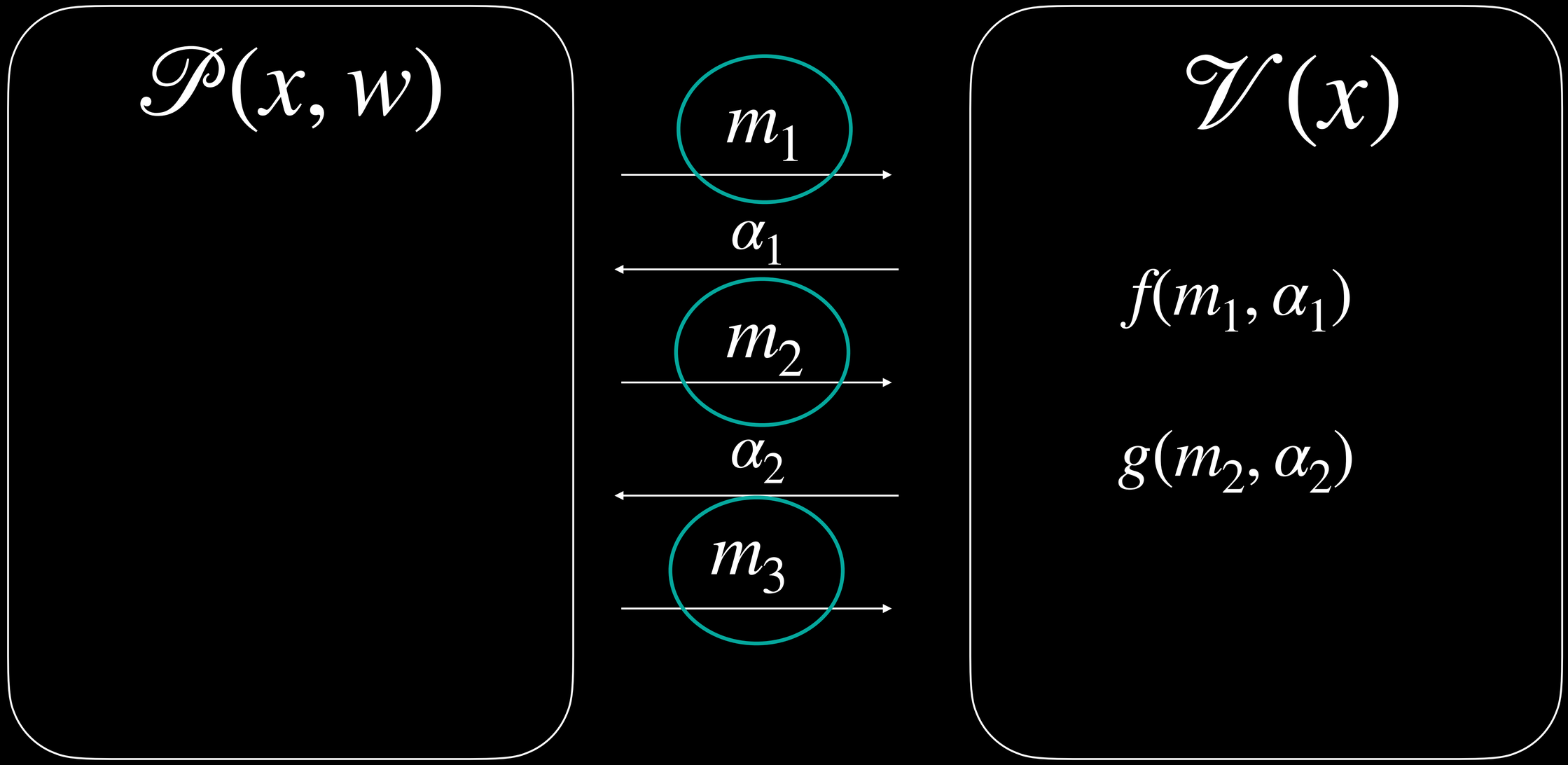
$$Pr\left[(x,w) \in R : w \leftarrow \mathscr{E}^{\mathscr{P}*}(x)\right] - Pr\left[\langle \mathscr{P}*(x),\mathscr{V}(x)\rangle = 1\right] \leq negl(\lambda)$$

Zero-Knowledge

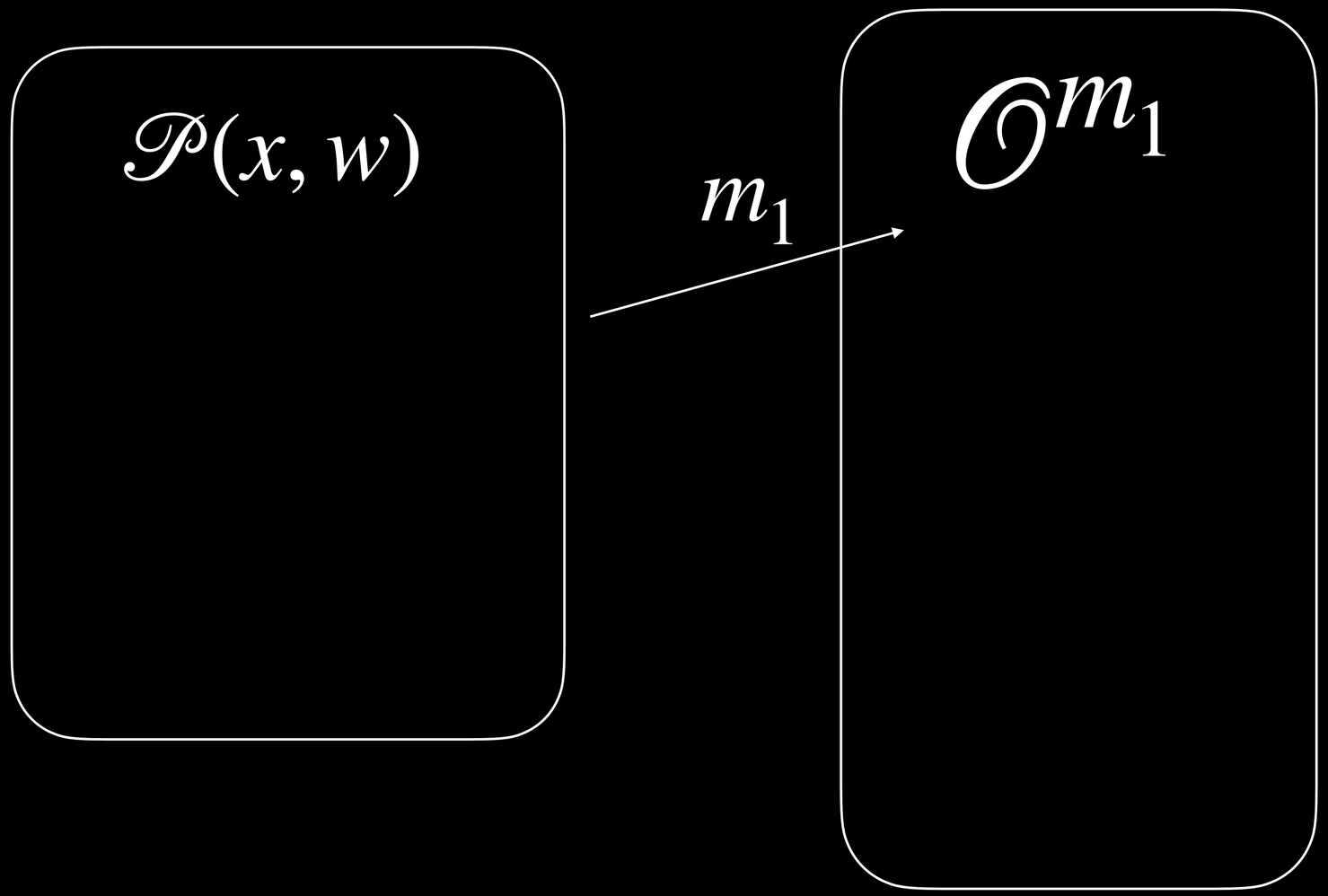
The Verifier does not learn anything but the truth of *Something*

How to build SNARK(G)s?

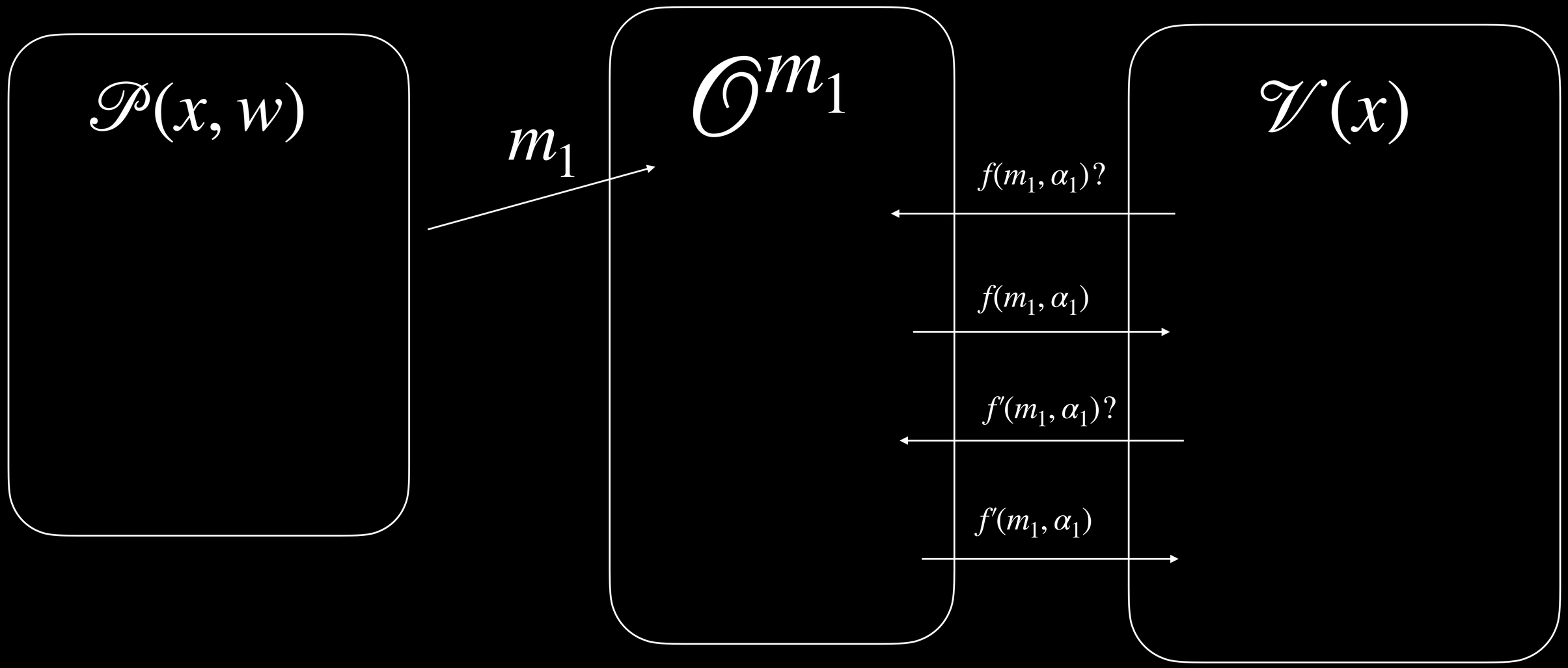
Tool 1: Interactive Oracle Proof



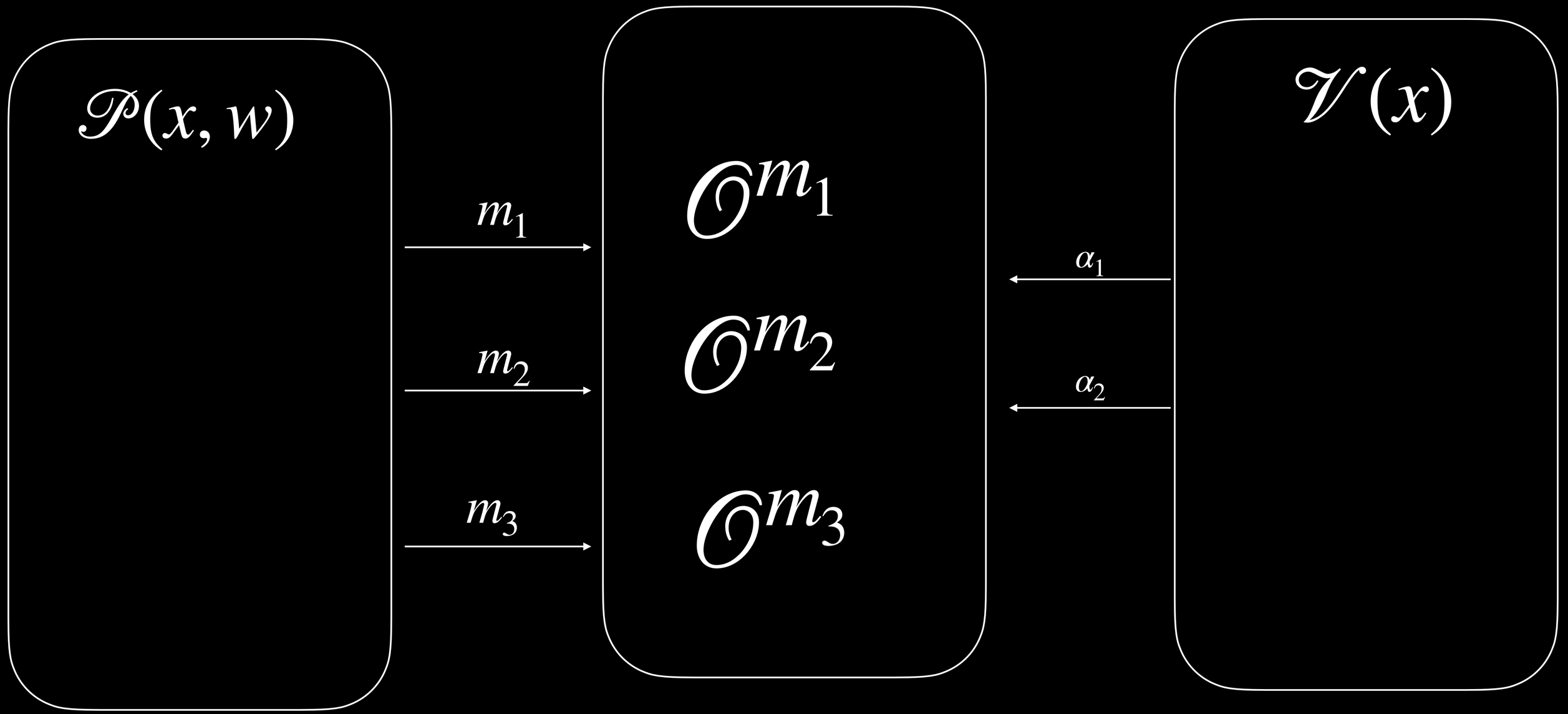
Tool 1: Interactive Oracle Proof



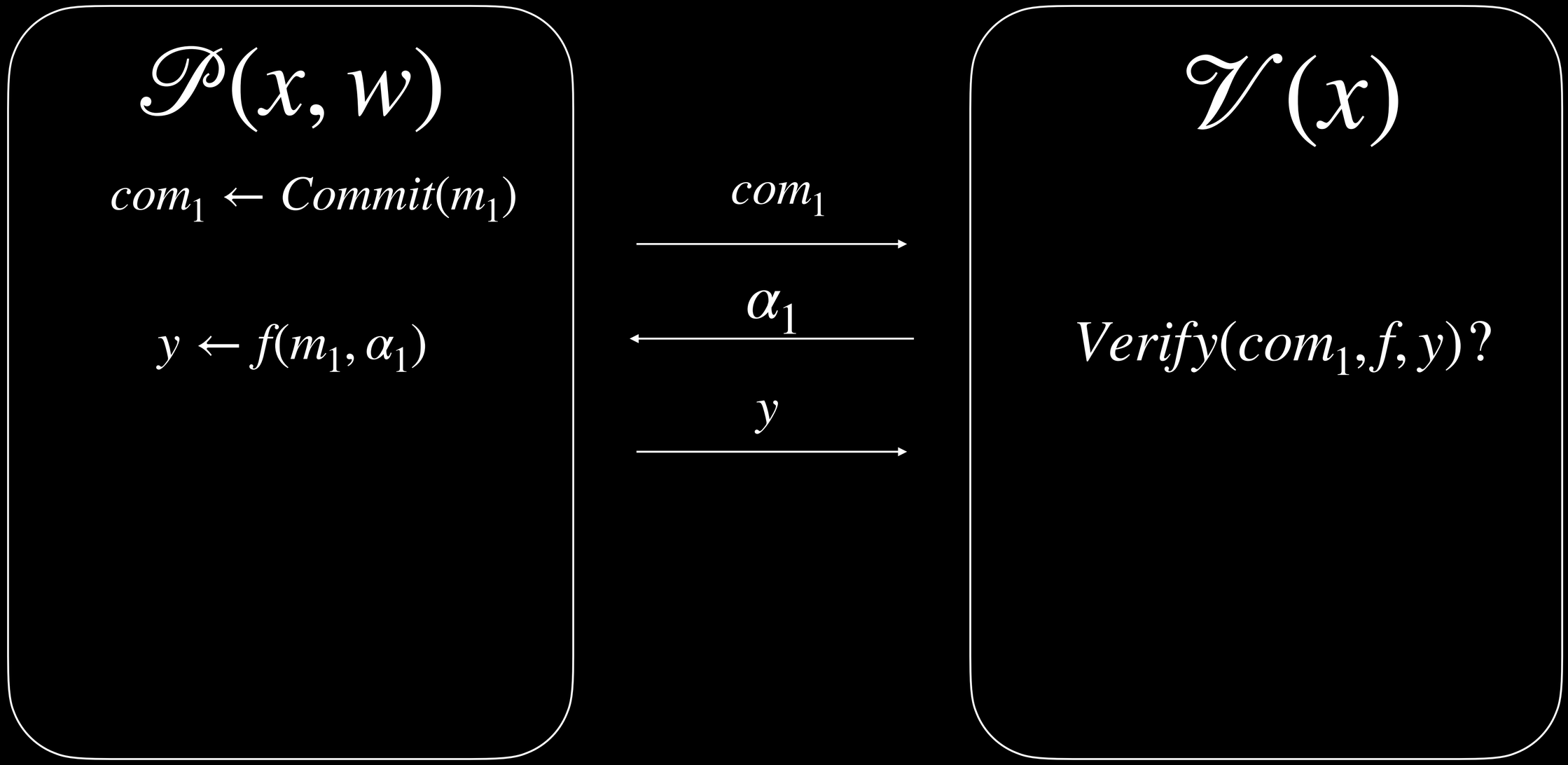
Tool 1: Interactive Oracle Proof



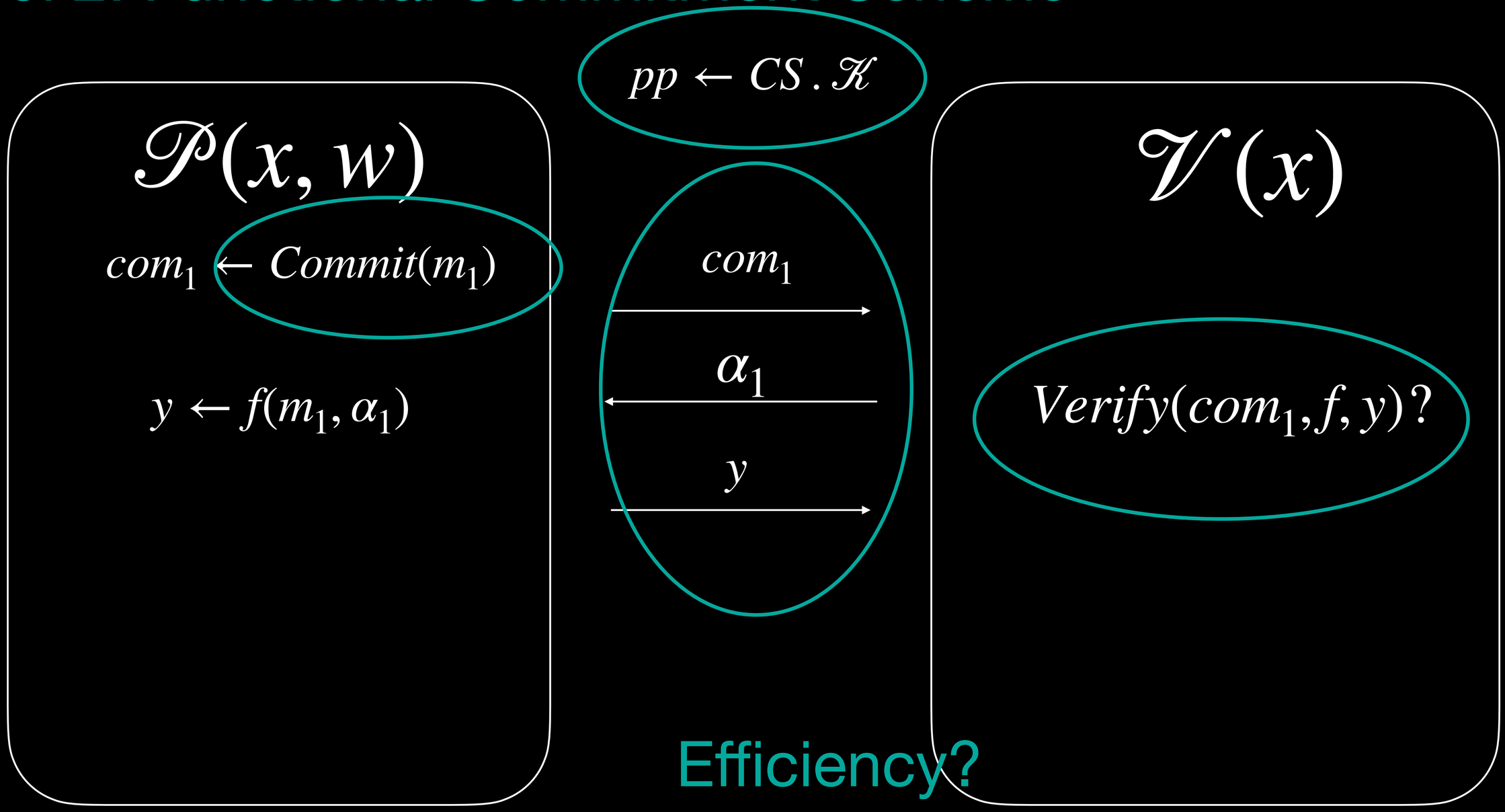
Tool 2: Functional Commitment Scheme



Tool 2: Functional Commitment Scheme

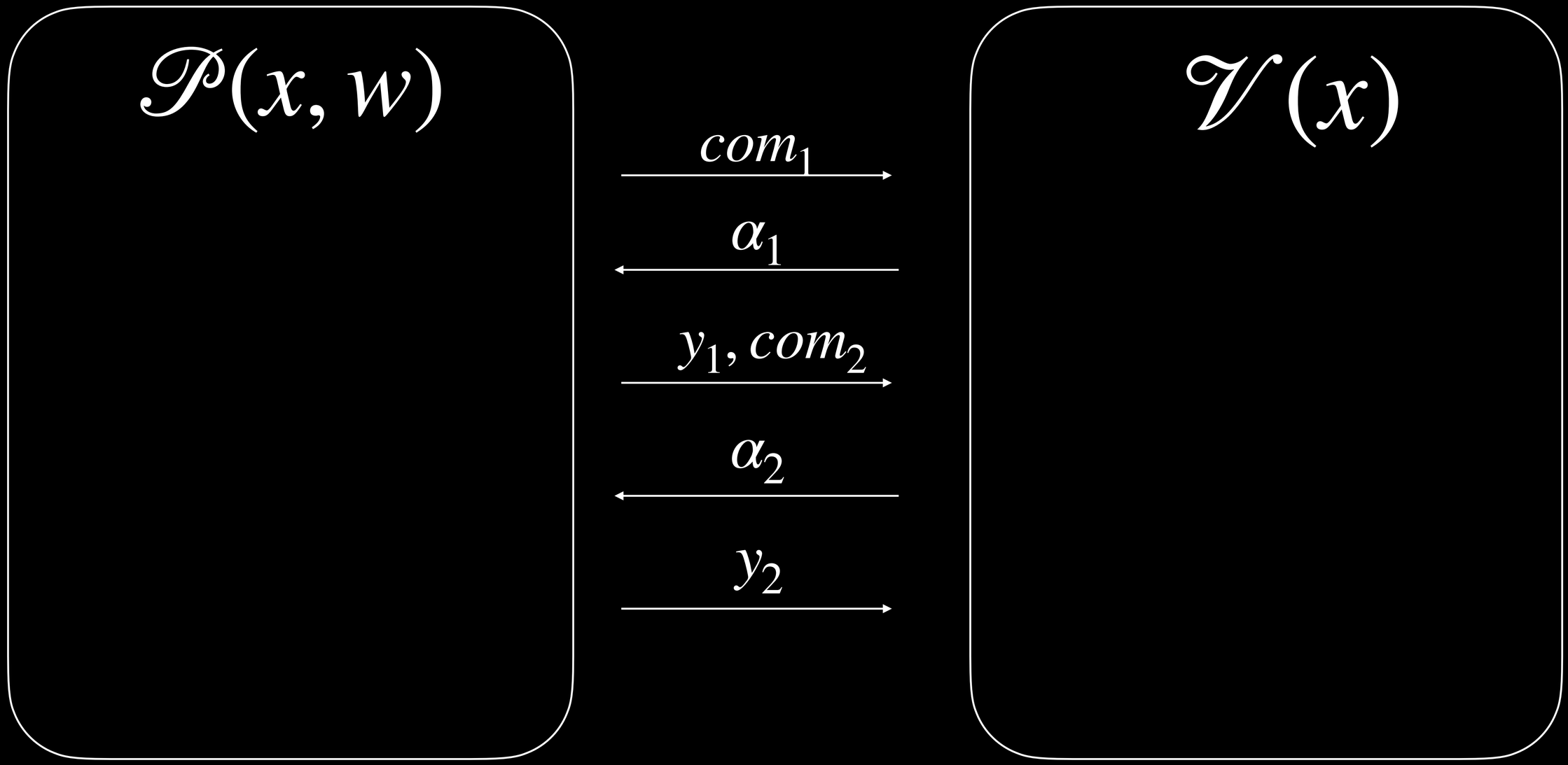


Tool 2: Functional Commitment Scheme

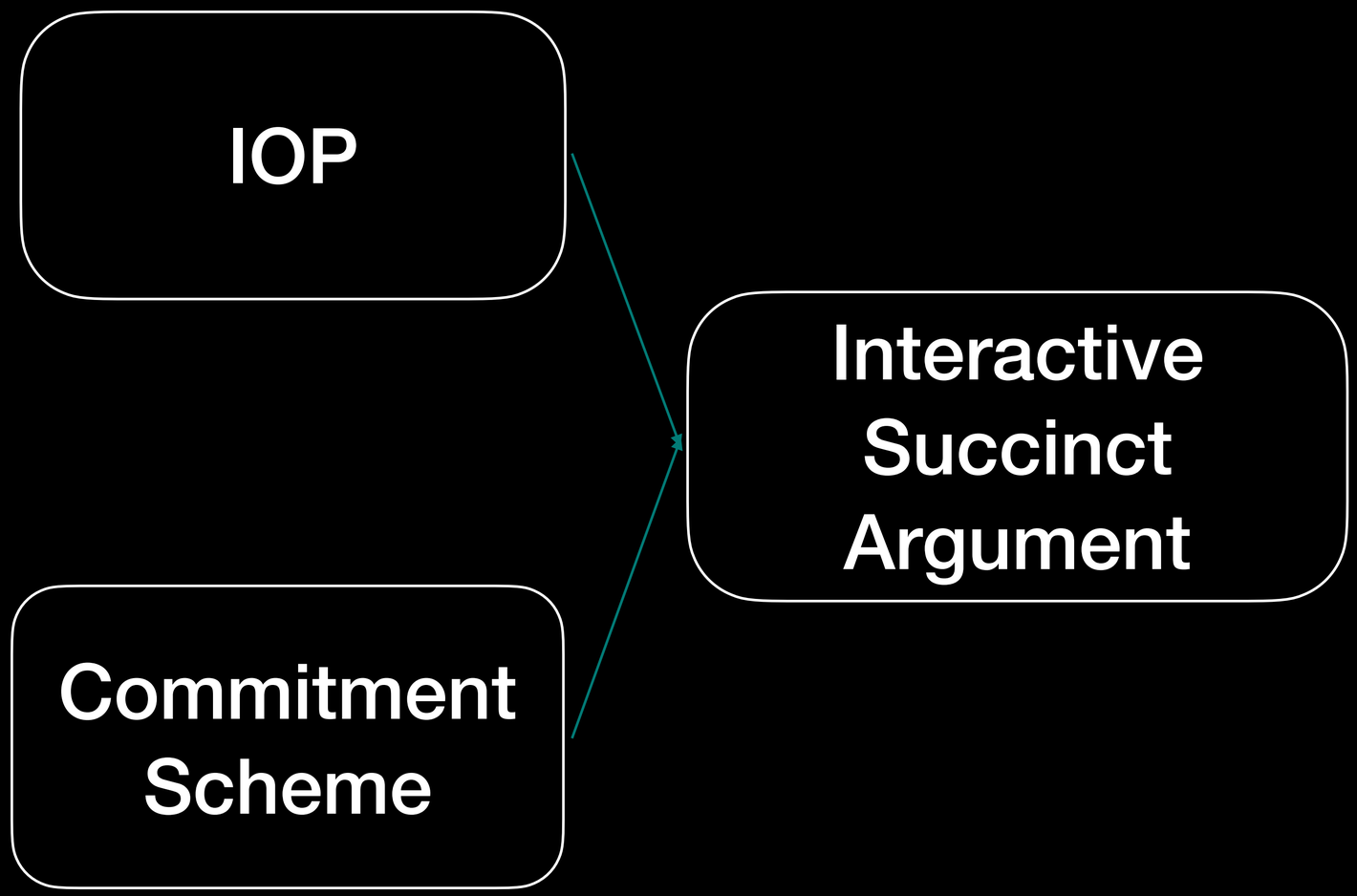


Efficiency?

Efficiency

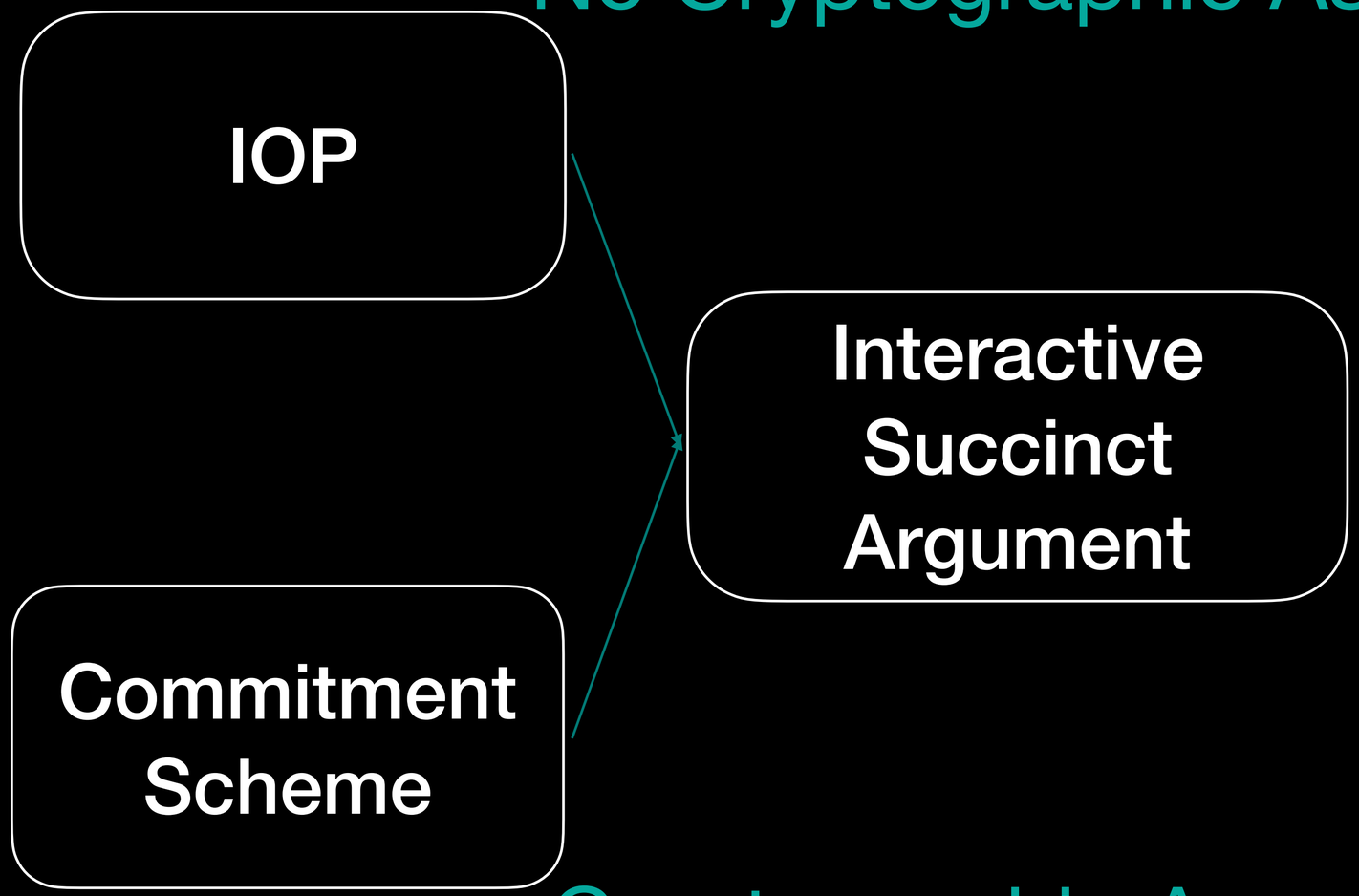


From Interactive to Non-interactive Proofs



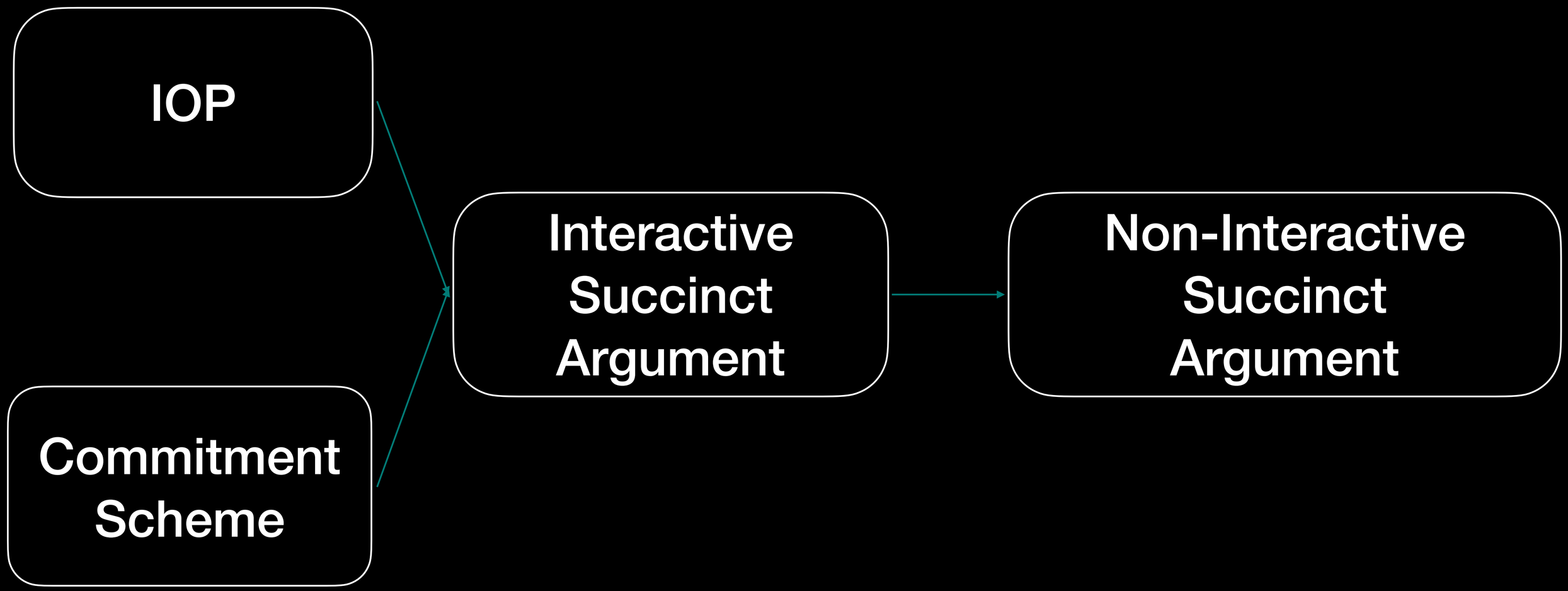
From Interactive to Non-interactive Proofs

No Cryptographic Assumptions



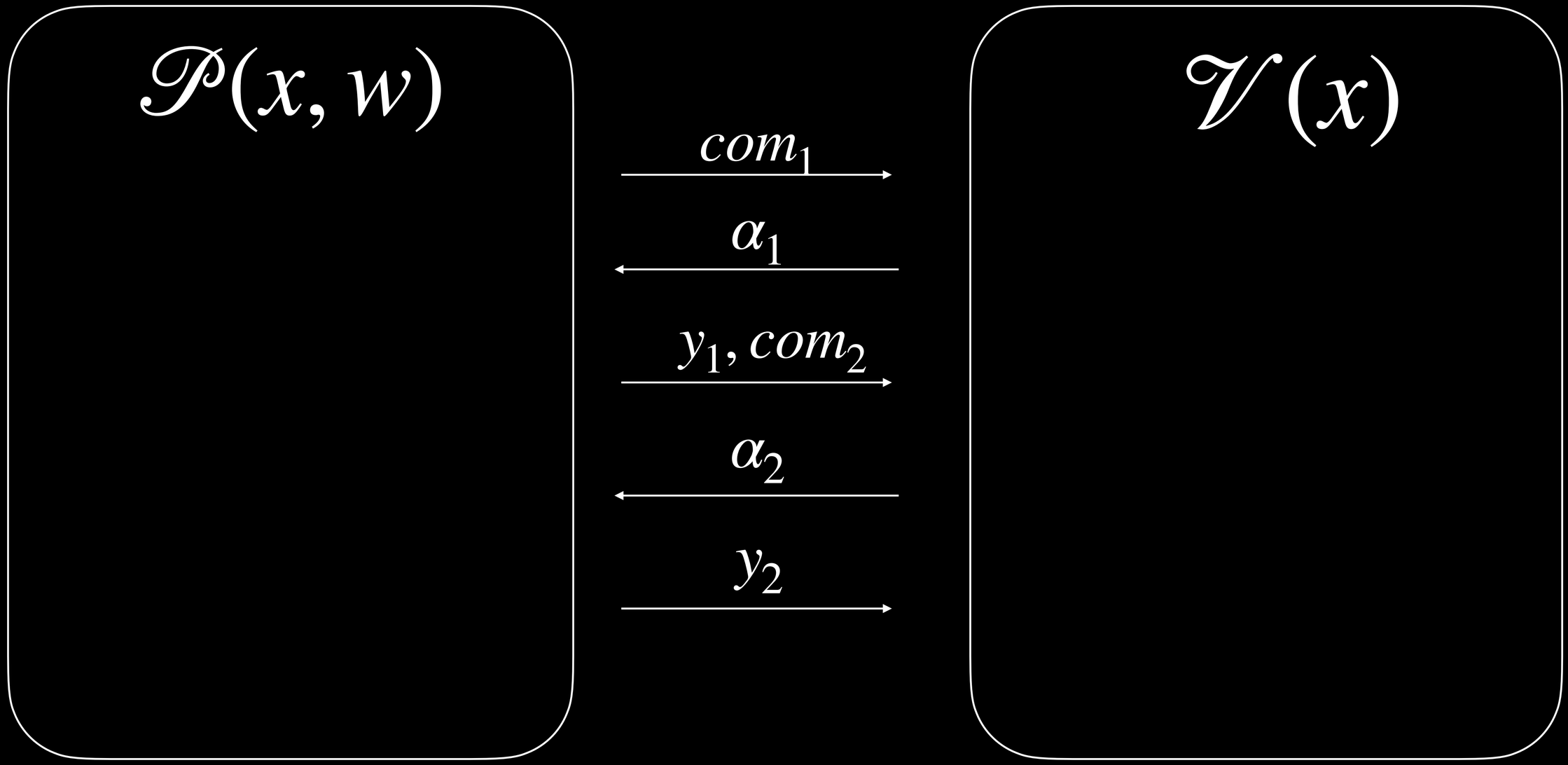
Cryptographic Assumptions here!

From Interactive to Non-interactive Proofs

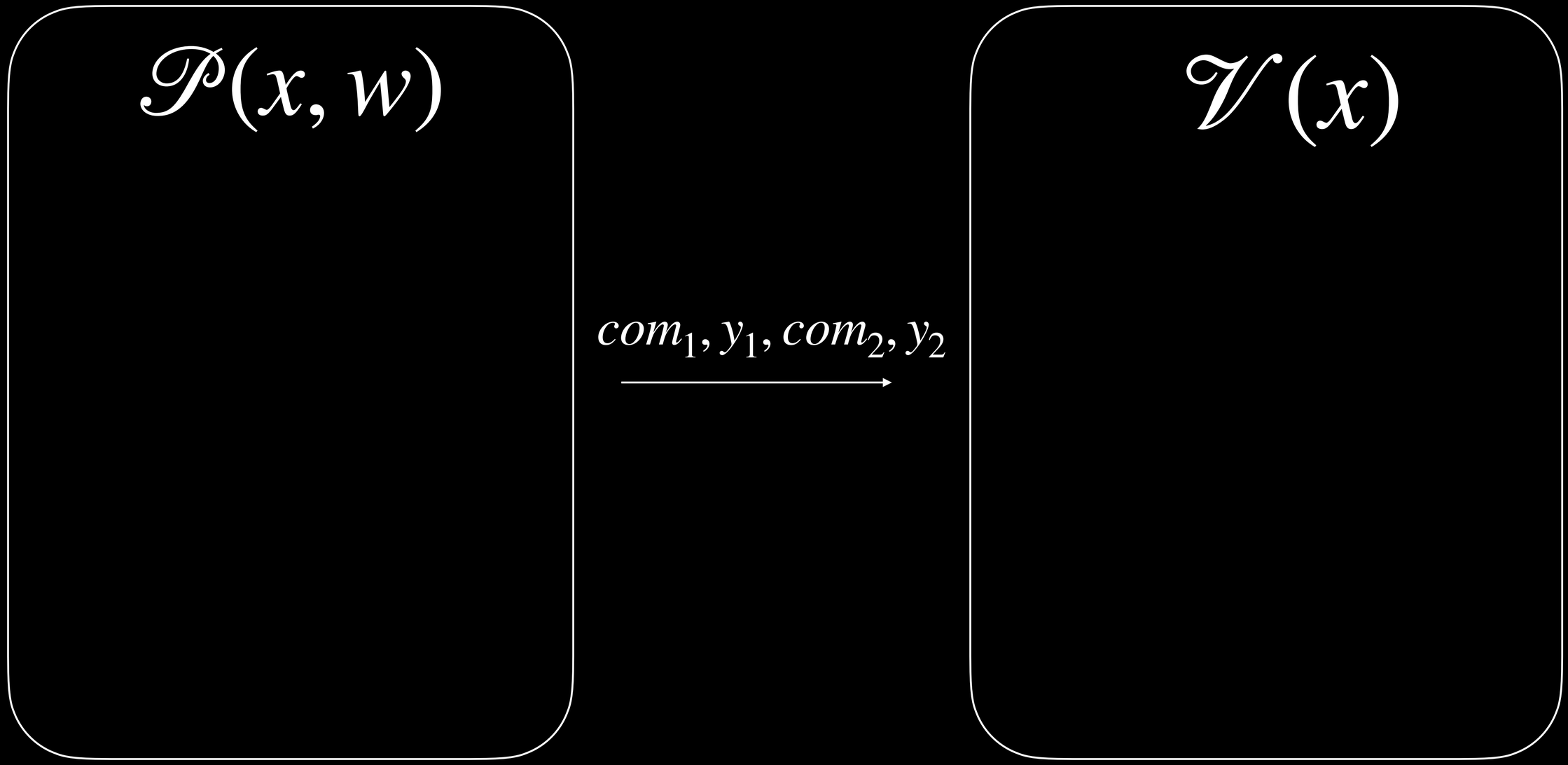


Tool 3: Hash function

Tool 3: Hash function



Tool 3: Hash function



Tool 3: Hash function

$$H : \{0,1\}^* \rightarrow \{0,1\}^{256}$$

Tool 3: Hash function

$$H : \{0,1\}^* \rightarrow \{0,1\}^{256}$$

Collison resistant:

Find x, y such that $H(x) = H(y)$

Tool 3: Hash function

$$H : \{0,1\}^* \rightarrow \{0,1\}^{256}$$

Collison resistant:

Find x, y such that $H(x) = H(y)$

Pre-image resistant:

Given z, find x such that $H(x) = z$

Tool 3: Hash function

$$H : \{0,1\}^* \rightarrow \{0,1\}^{256}$$

Collison resistant:

Find x, y such that $H(x) = H(y)$

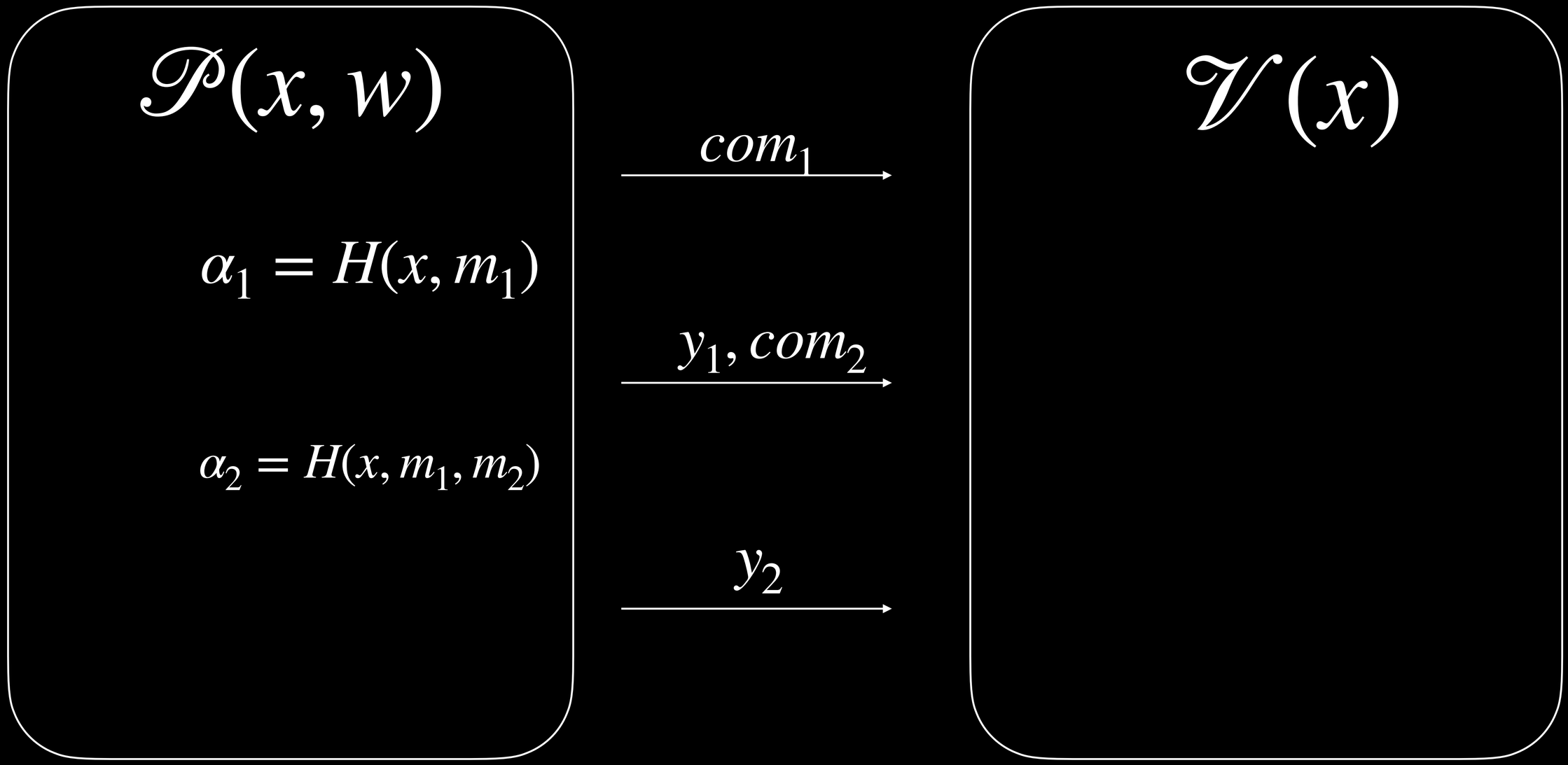
Pre-image resistant:

Given z, find x such that $H(x) = z$

Second pre-image resistant:

Given x find y such that $H(x) = H(y)$

The Fiat-Shamir Heuristic



The Fiat-Shamir Heuristic

$\mathscr{P}(x, w)$

com_1

$\alpha_1 = H(x, m_1)$

y_1, com_2

$\alpha_2 = H(x, m_1, m_2)$

y_2

$\pi = com_1, y_1, com_2, y_2$

$\mathscr{V}(x)$

The Fiat-Shamir Heuristic

$\mathscr{P}(x, w)$

com_1

$\alpha_1 = H(x, m_1)$

y_1, com_2

$\alpha_2 = H(x, m_1, m_2)$

y_2

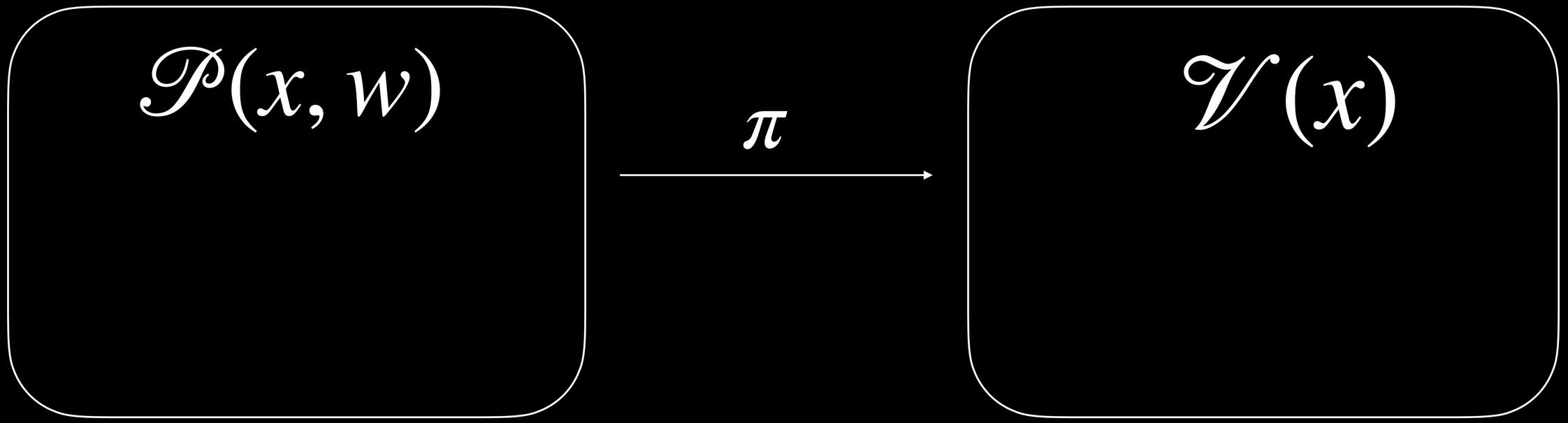
$\pi = com_1, y_1, com_2, y_2$

$\longrightarrow$

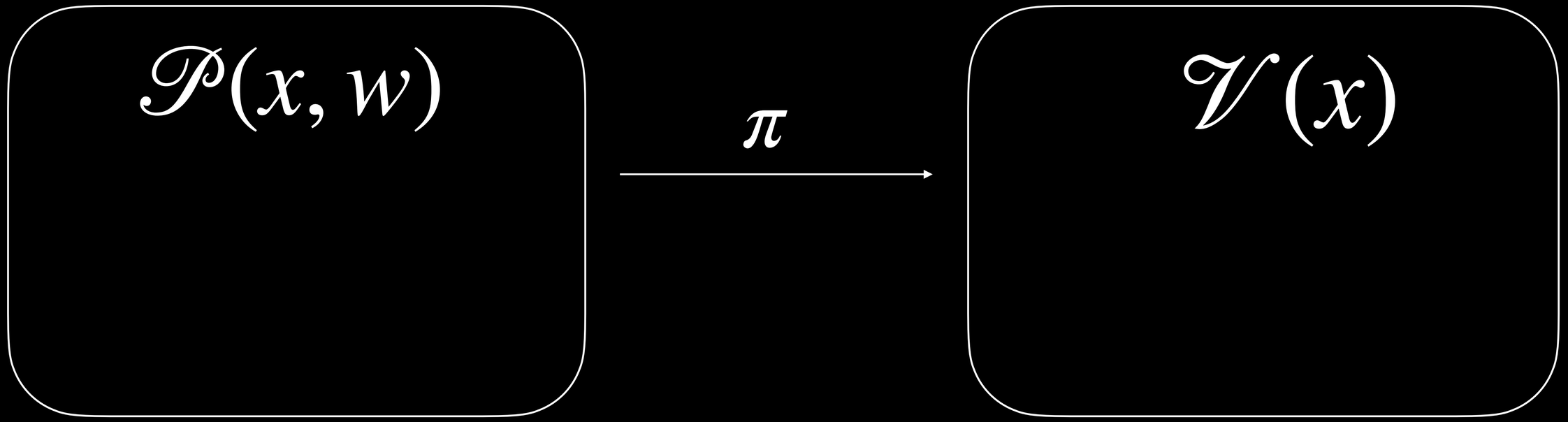
$\mathscr{V}(x)$

Secure under the Random Oracle Model!!!!

The Fiat-Shamir Heuristic



The Fiat-Shamir Heuristic



Knowledge soundness:

$$Pr\left[\begin{array}{c} (x,w) \notin R \wedge \\ \mathscr{V}(pp,x,\pi) = 1 \end{array} ; \begin{array}{c} pp \leftarrow \mathscr{K} \\ (x,\pi) \leftarrow \mathscr{P}^*(pp) \\ w \leftarrow \mathscr{E}(pp,x,\pi) \end{array}\right] \leq negl(\lambda)$$

Take aways:

SNARK: Succinct Non-interactive Argument of Knowledge

SNARG: Succinct Non-interactive Argument

Efficiency: Prover time, verifier time, proof-size, pp-size

Security: setup (trusted/transparent), model (ROM) and assumptions (discrete log)

Most of it depends on the commitment scheme!

¡¡¡Gracias!!!

Obrigado!!

arantxa@ethereum.org

@criptolatinoOrg