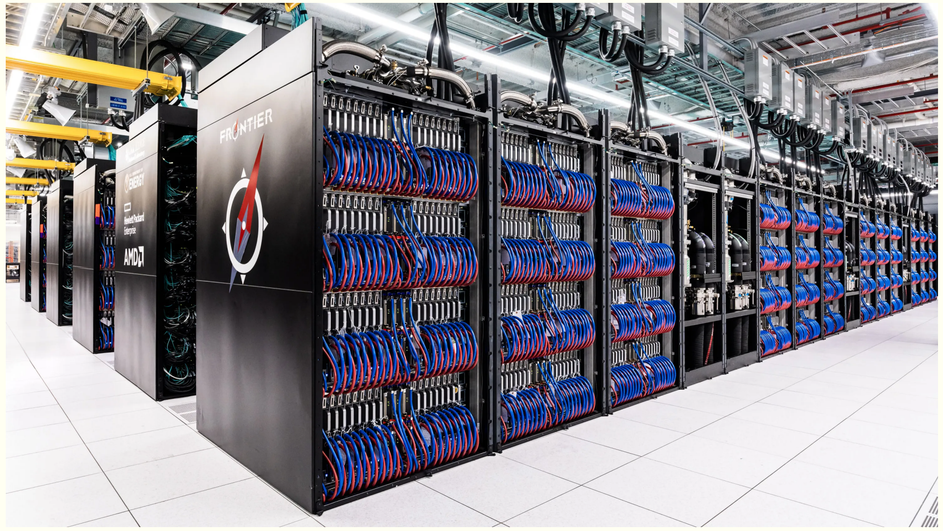
Unexpected uses of quantum computers

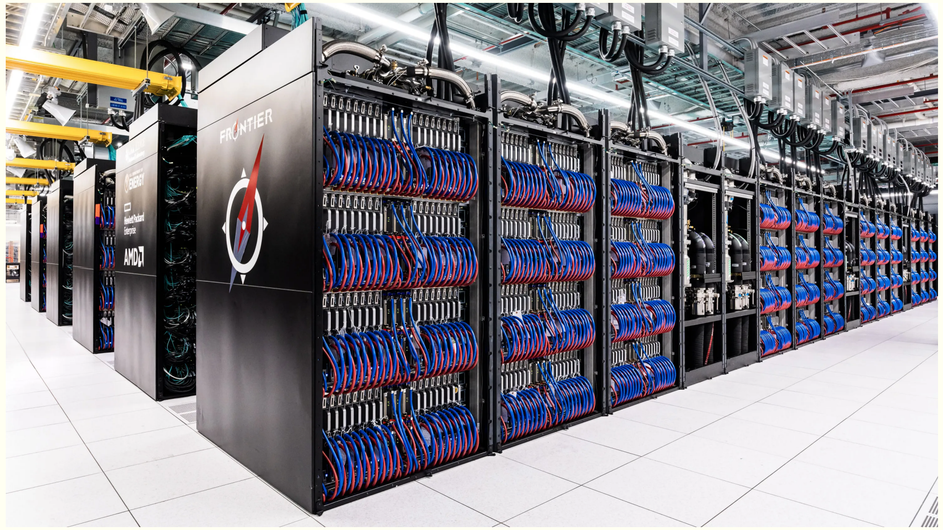
John Bostanci

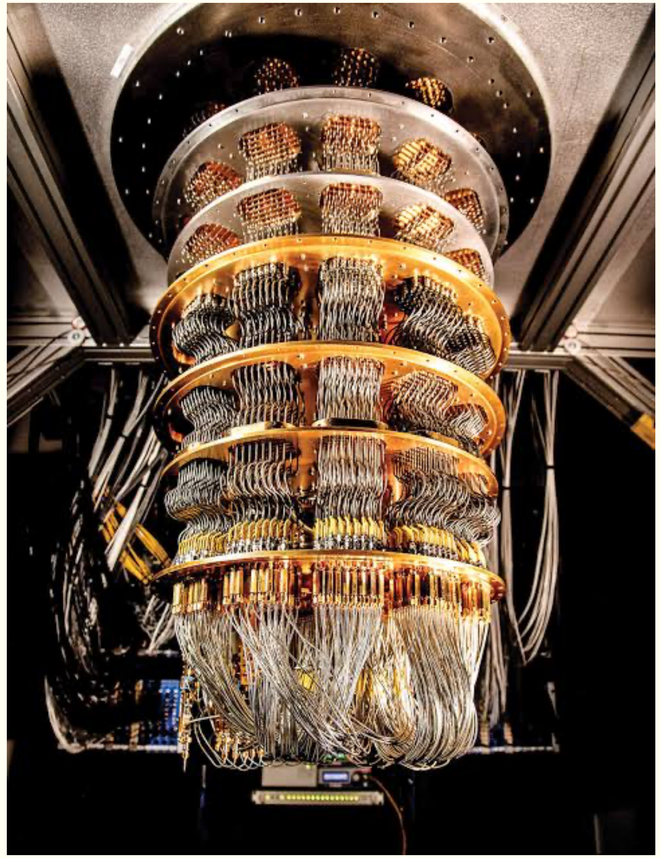
What is quantum computing?

What is quantum computing?

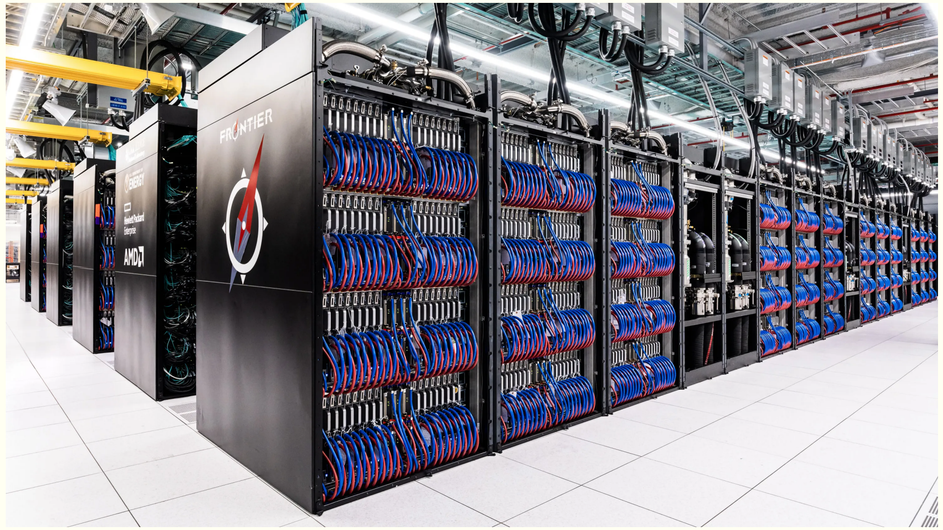


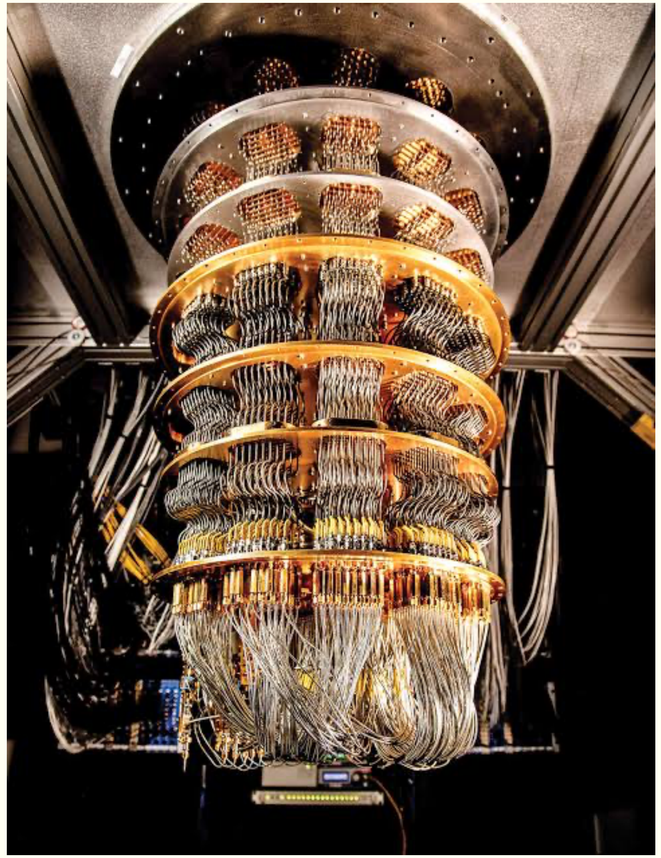
What is quantum computing?

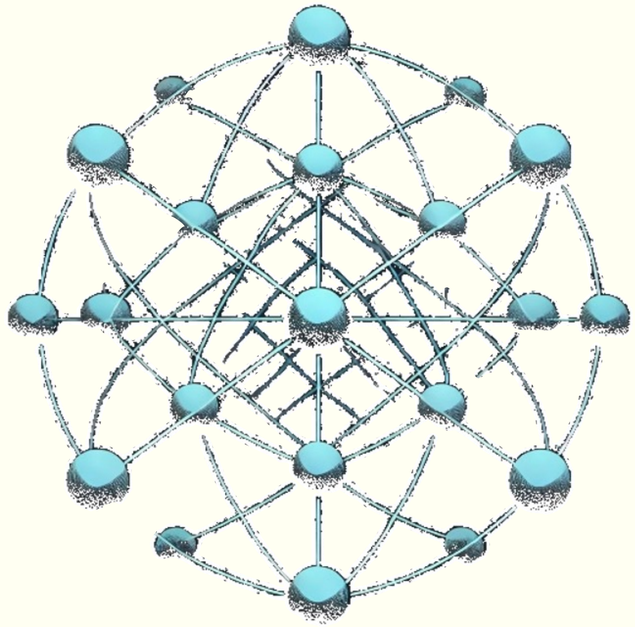




What is quantum computing?







What is quantum computing?

To me:

Quantum computing is the idea of using the physical world to perform computation,

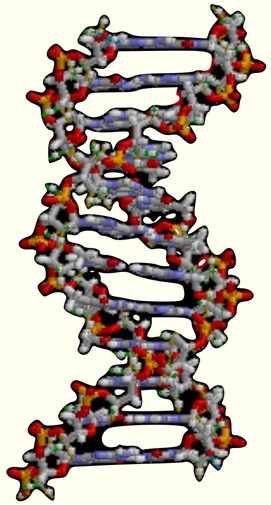
What is quantum computing?

To me:

Quantum computing is the idea of using the physical world to perform computation, and specifically using the properties of quantum mechanics!

Expected uses of quantum computers?

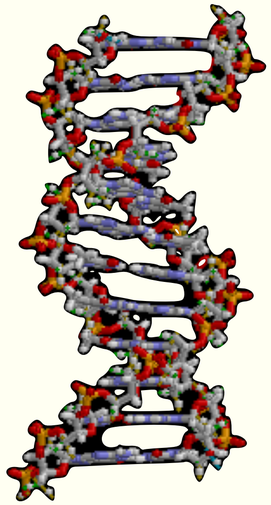
- Simulating nature?





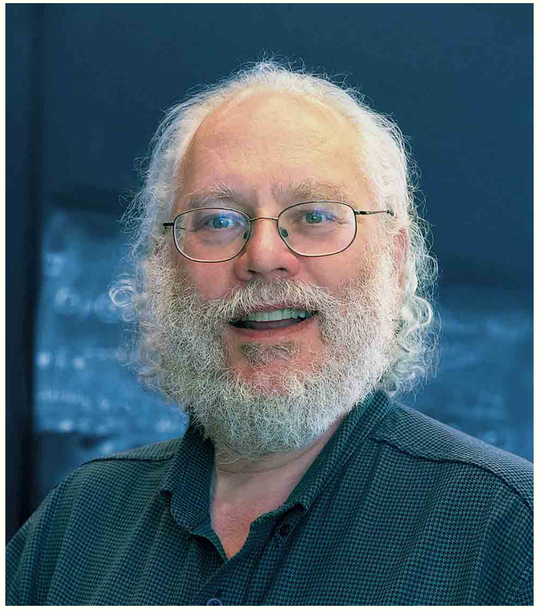
Expected uses of quantum computers?

- Simulating nature?





- Breaking cryptography?





What else can quantum computers do?

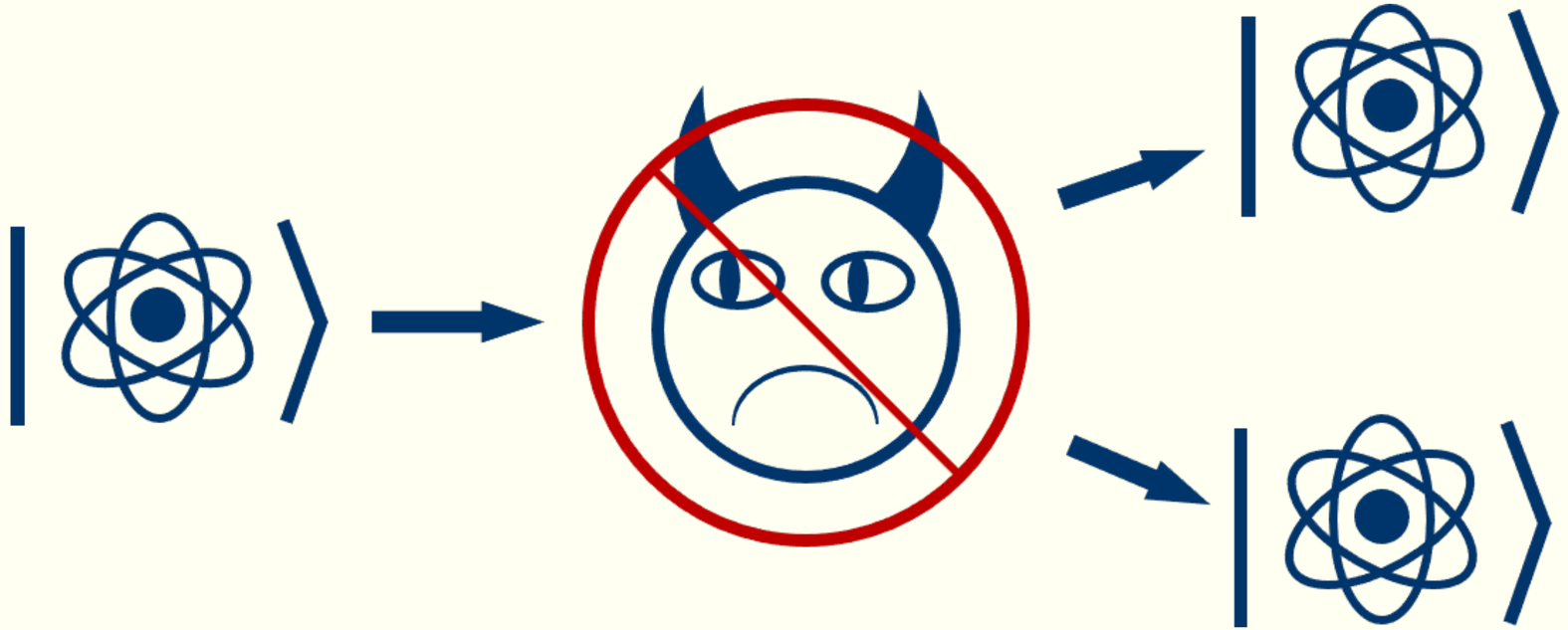
One less popularized use of quantum computers is: They can give “physical properties” to data!

What else can quantum computers do?

One less popularized use of quantum computers is: They can give “physical properties” to data! For example: You can’t make an exact copy of physical objects!

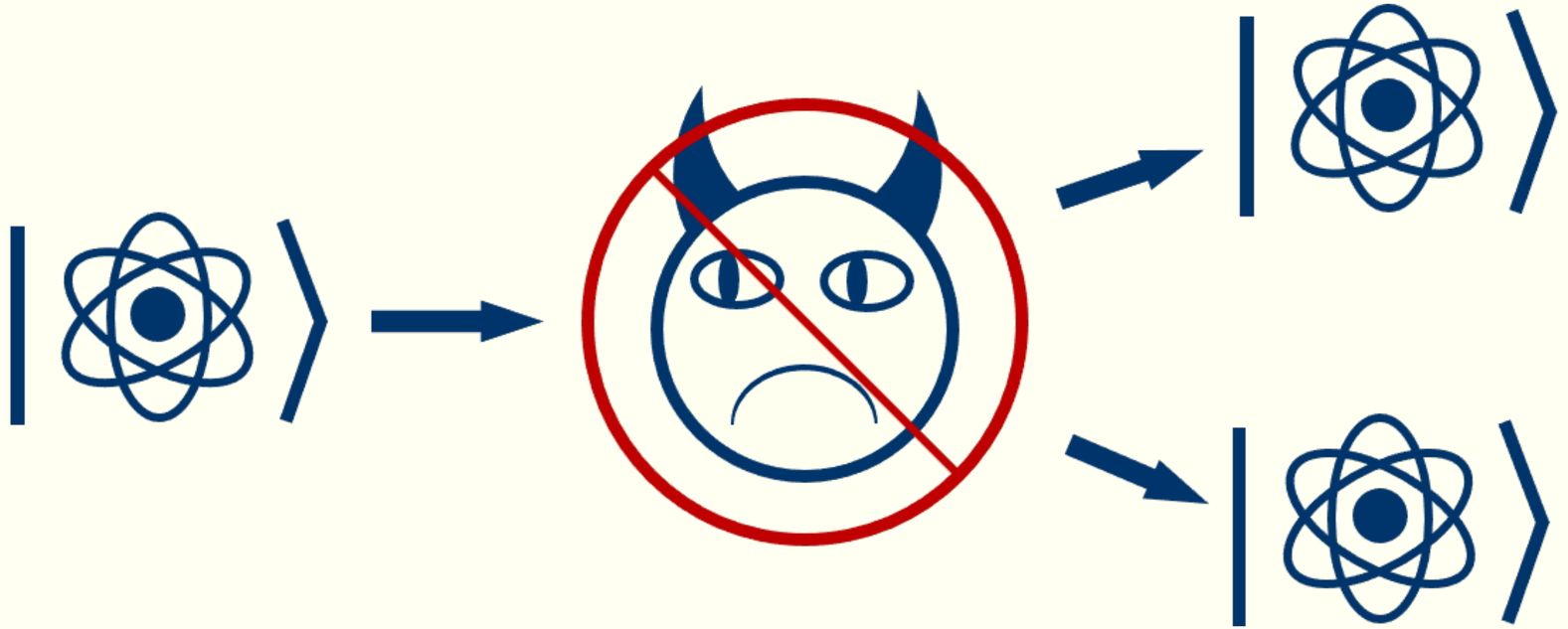
What else can quantum computers do?

One less popularized use of quantum computers is: They can give “physical properties” to data! For example: You can’t make an exact copy of physical objects!



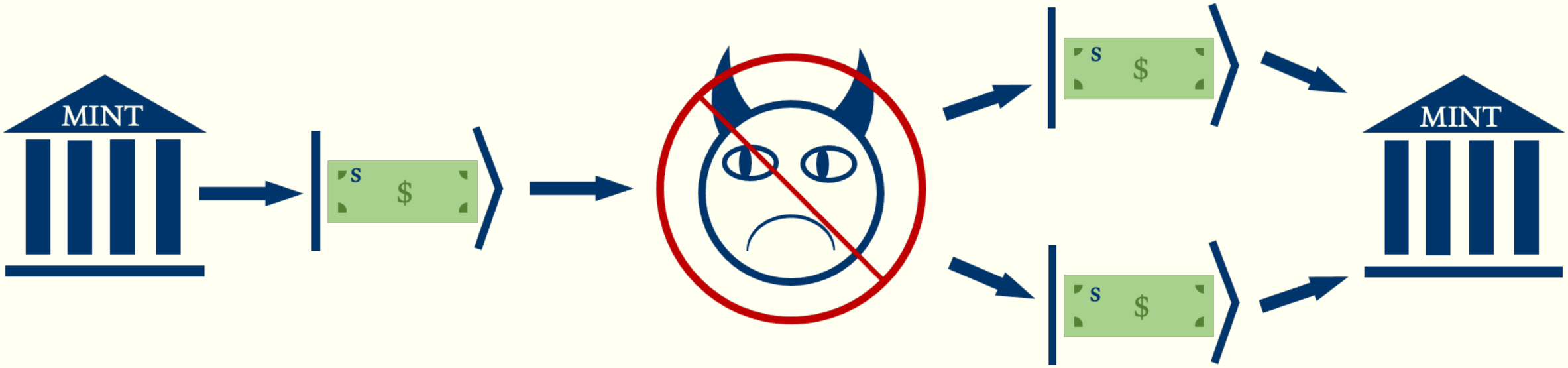
Enter quantum cryptography

Based on this idea, Stephen Wiesner first proposed “quantum money” in 1972.



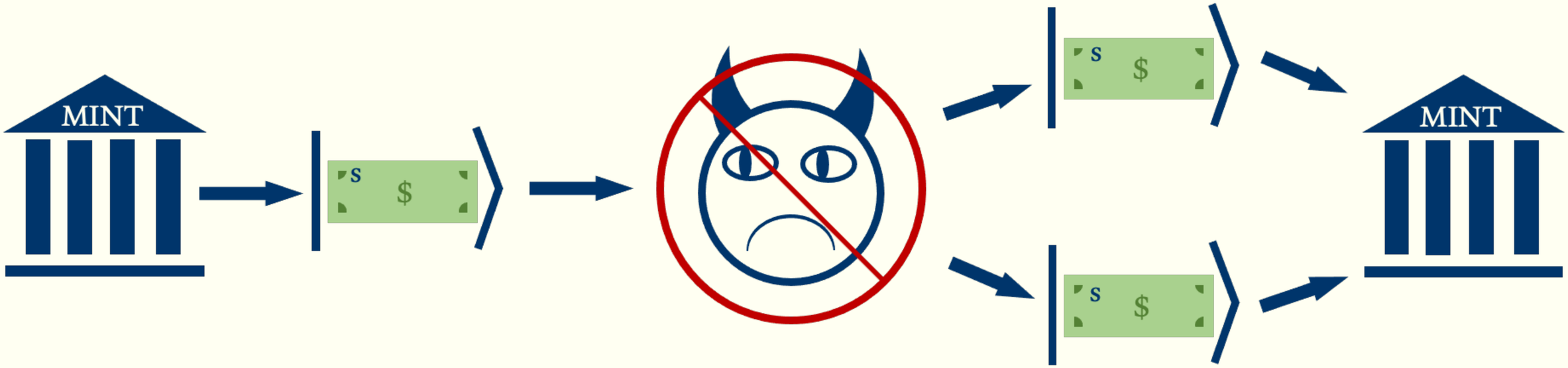
Enter quantum cryptography

Based on this idea, Stephen Weisner first proposed “quantum money” in 1972.



Enter quantum cryptography

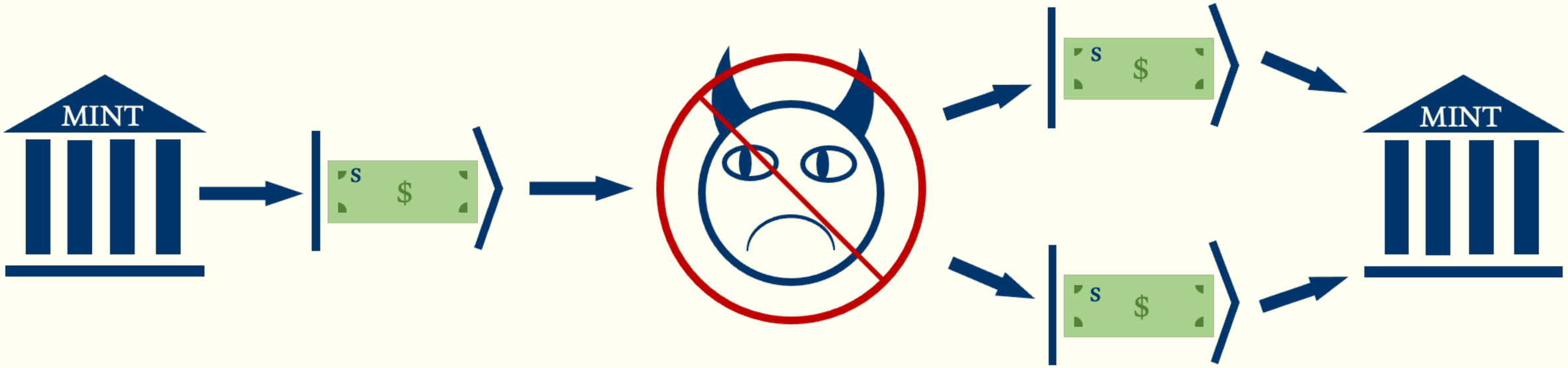
Based on this idea, Stephen Weisner first proposed “quantum money” in 1972.



His ideas, in turn, inspired quantum key distribution, another big application of quantum computers!

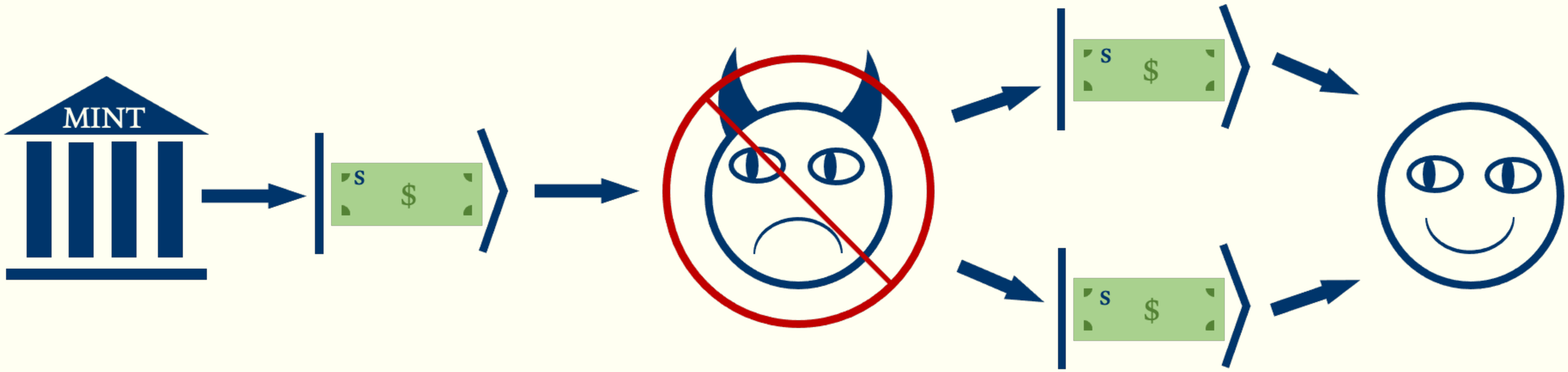
Quantum cryptography today

Quantum cryptography had a flurry of work in the 1980's and 90's, but was revived in 2009 by the work of Aaronson.



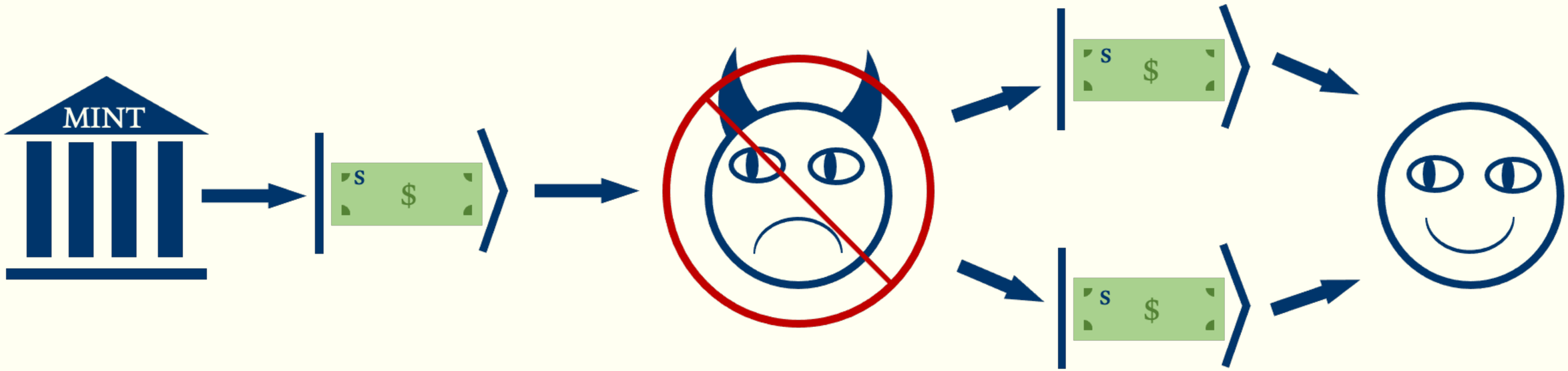
Quantum cryptography today

Quantum cryptography had a flurry of work in the 1980's and 90's, but was revived in 2009 by the work of Aaronson.



Quantum cryptography today

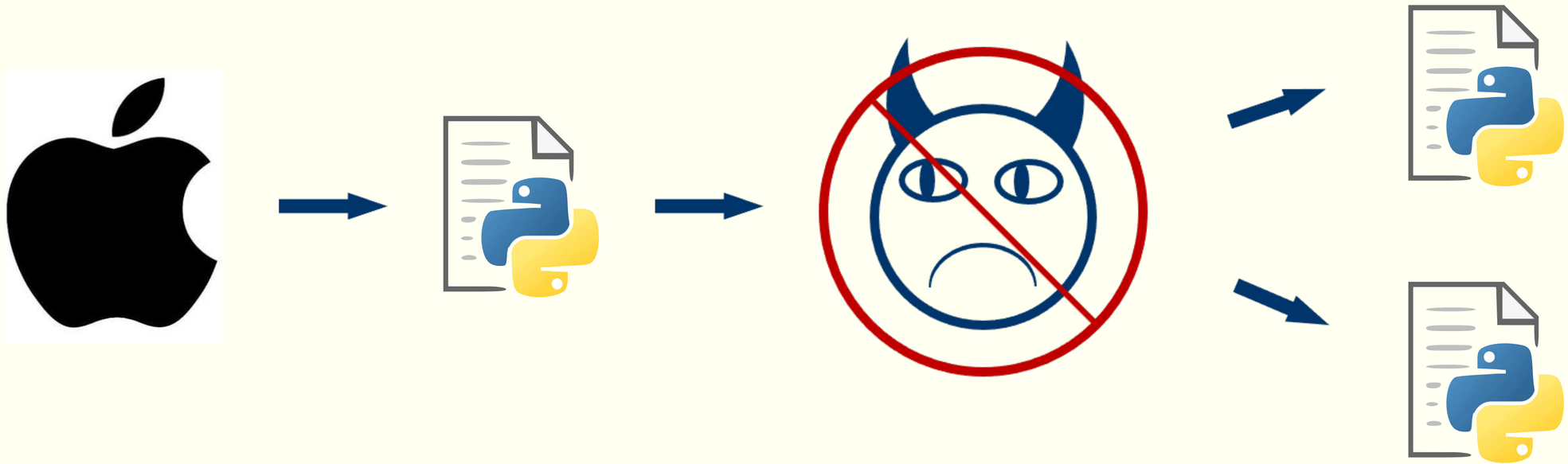
Quantum cryptography had a flurry of work in the 1980's and 90's, but was revived in 2009 by the work of Aaronson.



Can unclonable quantum states be endowed with other features?

Quantum cryptography today

Quantum cryptography had a flurry of work in the 1980's and 90's, but was revived in 2009 by the work of Aaronson.



Can unclonable quantum states be endowed with other features?

Quantum cryptography today

In the last 10 years, there has been another flurry of quantum cryptographic primitives, inspired by the work of Aaronson.

Quantum cryptography today

In the last 10 years, there has been another flurry of quantum cryptographic primitives, inspired by the work of Aaronson.

- Quantum lightning (collision resistant quantum money) [Zha'17]

Quantum cryptography today

In the last 10 years, there has been another flurry of quantum cryptographic primitives, inspired by the work of Aaronson.

- Quantum lightning (collision resistant quantum money) [Zha'17]
- One-shot signatures [AGKZ'20]

Quantum cryptography today

In the last 10 years, there has been another flurry of quantum cryptographic primitives, inspired by the work of Aaronson.

- Quantum lightning (collision resistant quantum money) [Zha'17]
- One-shot signatures [AGKZ'20]
- Quantum fire [BNZ'25]

Quantum cryptography today

In the last 10 years, there has been another flurry of quantum cryptographic primitives, inspired by the work of Aaronson.

- Quantum lightning (collision resistant quantum money) [Zha'17]
- One-shot signatures [AGKZ'20]
- Quantum fire [BNZ'25]
- Unclonable encryption [BL'20]

Quantum cryptography today

In the last 10 years, there has been another flurry of quantum cryptographic primitives, inspired by the work of Aaronson.

- Quantum lightning (collision resistant quantum money) [Zha'17]
- One-shot signatures [AGKZ'20]
- Quantum fire [BNZ'25]
- Unclonable encryption [BL'20]
- One-time programs [BGS'13]

Quantum cryptography today

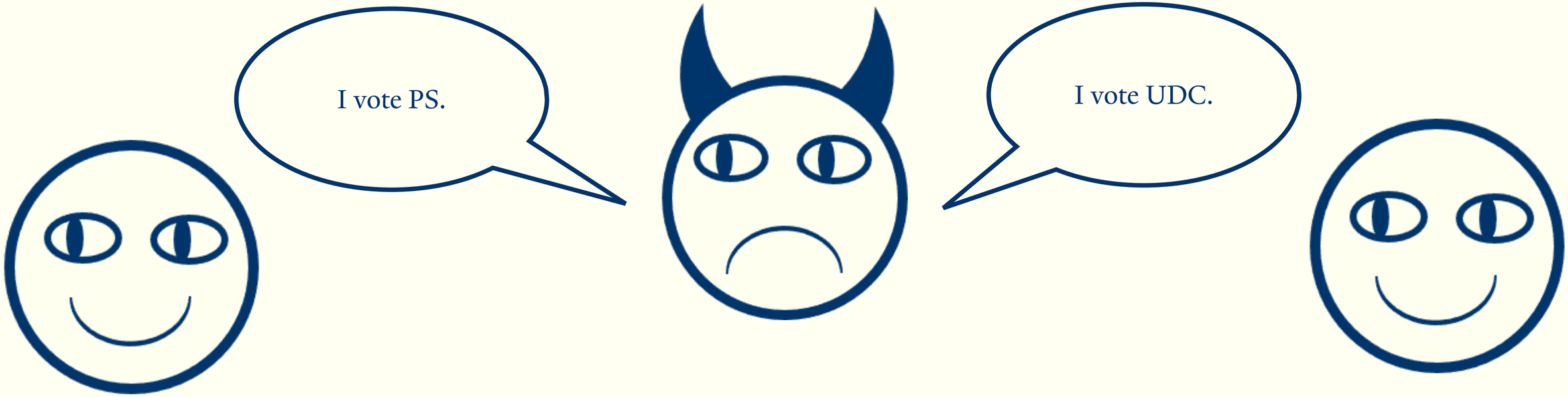
In the last 10 years, there has been another flurry of quantum cryptographic primitives, inspired by the work of Aaronson.

- Quantum lightning (collision resistant quantum money) [Zha'17]
- One-shot signatures [AGKZ'20]
- Quantum fire [BNZ'25]
- Unclonable encryption [BL'20]
- One-time programs [BGS'13]

Are these ideas useful for practical problems that computer scientists encounter?

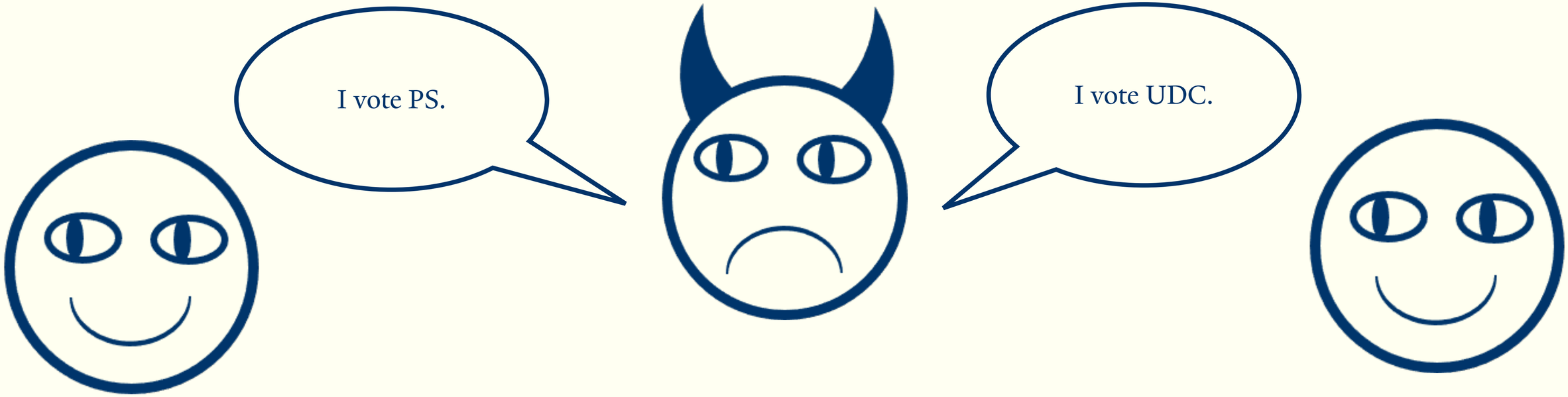
Equivocation in consensus

Equivocation is concept in consensus/blockchains when a party tells two different people different results:



Equivocation in consensus

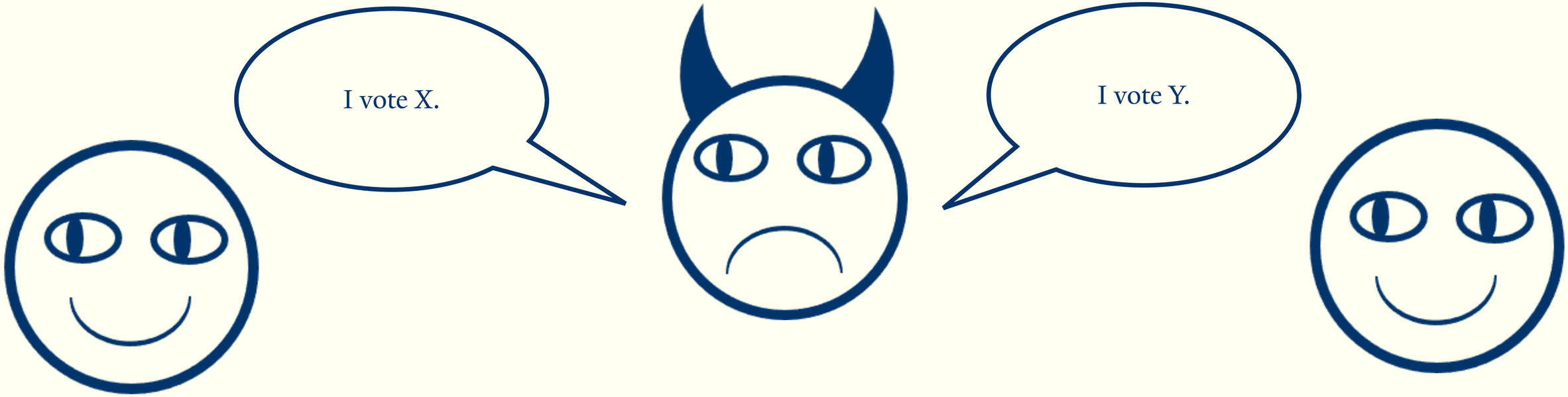
Equivocation is concept in consensus/blockchains when a party tells two different people different results:



This can be a problem when people want to agree on things (for example, leader elections) in a distributed system. Consensus protocols want to mitigate people lying, but also reduce communication cost.

Equivocation in consensus

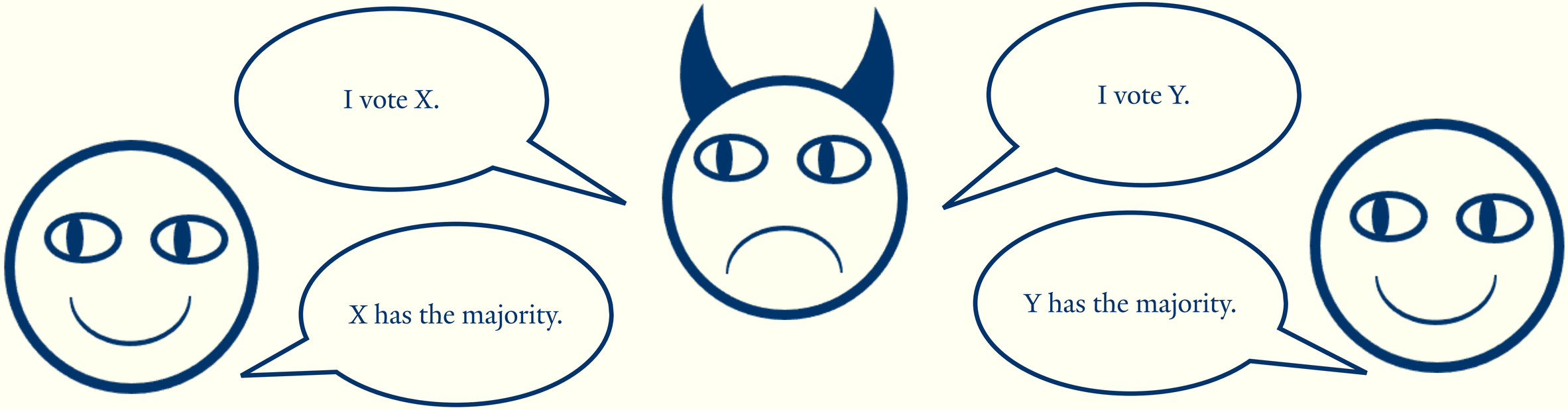
Equivocation is concept in consensus/blockchains when a party tells two different people different results:



This can be a problem when people want to agree on things (for example, leader elections) in a distributed system. Consensus protocols want to mitigate people lying, but also reduce communication cost.

Equivocation in consensus

Equivocation is concept in consensus/blockchains when a party tells two different people different results:



This can be a problem when people want to agree on things (for example, leader elections) in a distributed system. Consensus protocols want to mitigate people lying, but also reduce communication cost.

Equivocation in consensus

What if every person could only “sign” one vote?



One shot signatures

A one-shot signature scheme is a triple of algorithms (Gen, Sign, Ver), with the property that the following hold

One shot signatures

A one-shot signature scheme is a triple of algorithms (Gen, Sign, Ver), with the property that the following hold

(Correctness): Verify accepts a valid signature with probability almost 1. I.e.,

$$\Pr\left[1 \leftarrow \mathsf{Ver}(\mathsf{CRS}, \mathsf{pk}, m, \sigma_m) \;\middle|\; \begin{array}{c} (\mathsf{pk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{Gen}(\mathsf{CRS}) \\ \sigma_m \leftarrow \mathsf{Sign}(\mathsf{CRS}, |\mathsf{sk}\rangle), m) \end{array}\right] = 1 - \mathsf{negl}$$

One shot signatures

A one-shot signature scheme is a triple of algorithms (Gen, Sign, Ver), with the property that the following hold

(Correctness): Verify accepts a valid signature with probability almost 1. I.e.,

$$\Pr\left[1 \leftarrow \mathsf{Ver}(\mathsf{CRS}, \mathsf{pk}, m, \sigma_m) \;\middle|\; \begin{matrix}(\mathsf{pk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{Gen}(\mathsf{CRS}) \\ \sigma_m \leftarrow \mathsf{Sign}(\mathsf{CRS}, |\mathsf{sk}\rangle), m)\end{matrix}\right] = 1 - \mathsf{negl}$$

(Soundness): The probability that an adversary generates 2 messages with signatures is negligible. I.e., For all adversaries A,

$$\Pr\left[1 \leftarrow \mathsf{Ver}(\mathsf{CRS}, \mathsf{pk}, m_0, \sigma_0) \wedge 1 \leftarrow \mathsf{Ver}(\mathsf{CRS}, \mathsf{pk}, m_1, \sigma_1) \,|\, (\mathsf{pk}, m_0, m_1, \sigma_0, \sigma_1) \leftarrow A(\mathsf{CRS})\right] = \mathsf{negl}$$

Equivocation in consensus

What if every person could only “sign” one vote?

A one-shot signature lets you sign messages, with the promise that only a single message can every be generated with a valid signature (for a particular public key).



Equivocation in consensus

What if every person could only “sign” one vote?

A one-shot signature lets you sign messages, with the promise that only a single message can every be generated with a valid signature (for a particular public key).

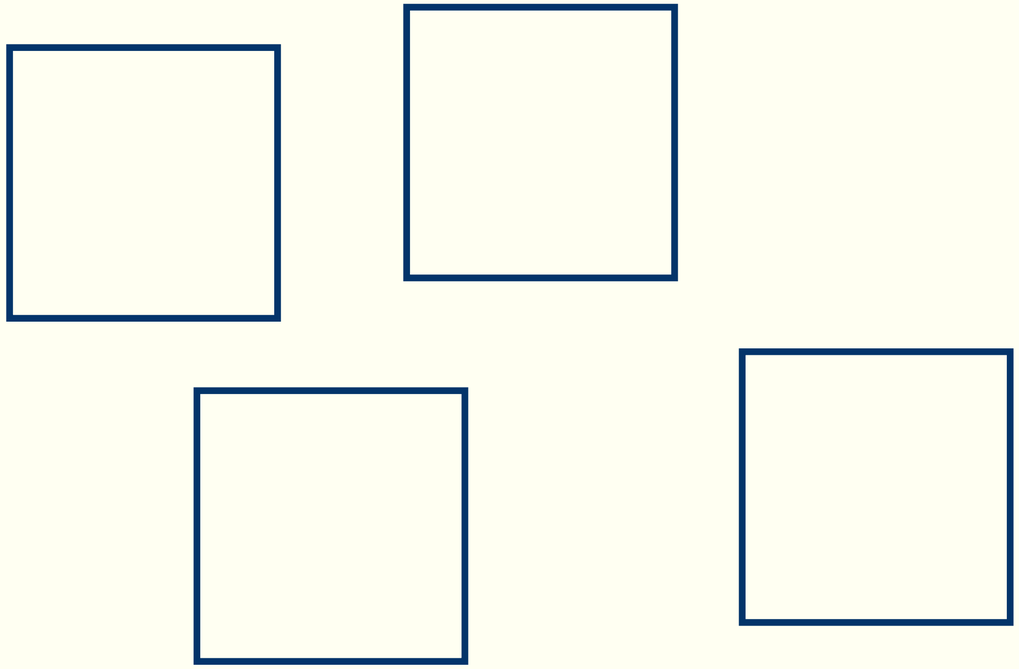


It turns out a lot of things in blockchains become easier with one-shot signatures! For example, a “leader” can only sign a single block, so no need for complicated protocols!

AI exfiltration and intelligence explosion?

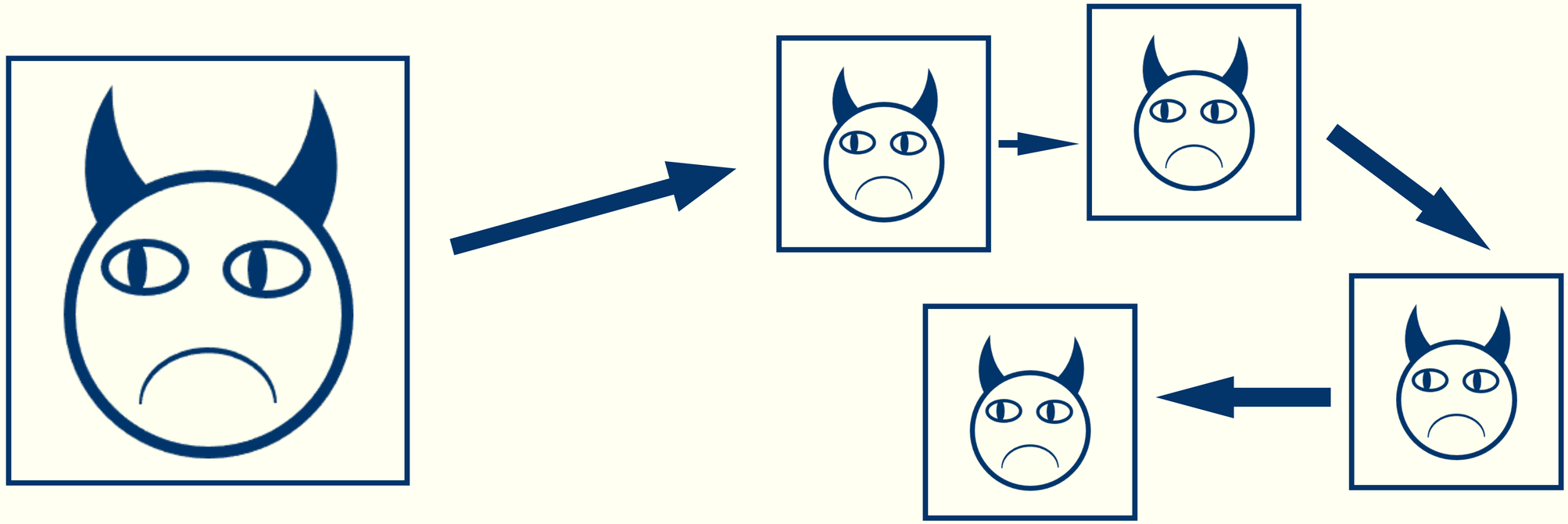
A big fear in AI safety (at least, as far as I understand) is that models might escape their original servers, and start running more and more instances of themselves…





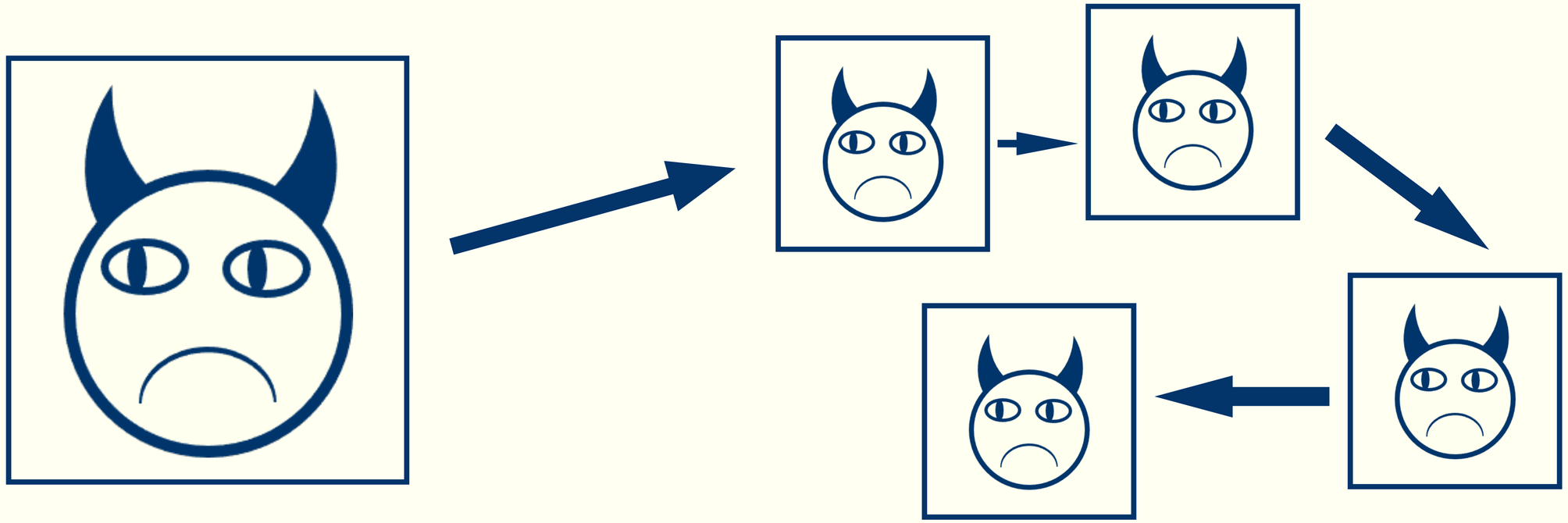
AI exfiltration and intelligence explosion?

A big fear in AI safety (at least, as far as I understand) is that models might escape their original servers, and start running more and more instances of themselves…



AI exfiltration and intelligence explosion?

A big fear in AI safety (at least, as far as I understand) is that models might escape their original servers, and start running more and more instances of themselves…



What if the AI was being run as a copy-protected program?

Quantum copy protection

Quantum copy protection is a triple of algorithms (Setup, Generate, Compute), with the following interfaces:

- $\mathsf{Setup}() \mapsto \mathsf{sk}$
- $\mathsf{Generate}(\mathsf{sk}, f) \mapsto (\rho_f, \{U_{f,x}\})$
- $\mathsf{Compute}(\rho_f, \{U_{f,x}\}, x) \mapsto y$

Quantum copy protection

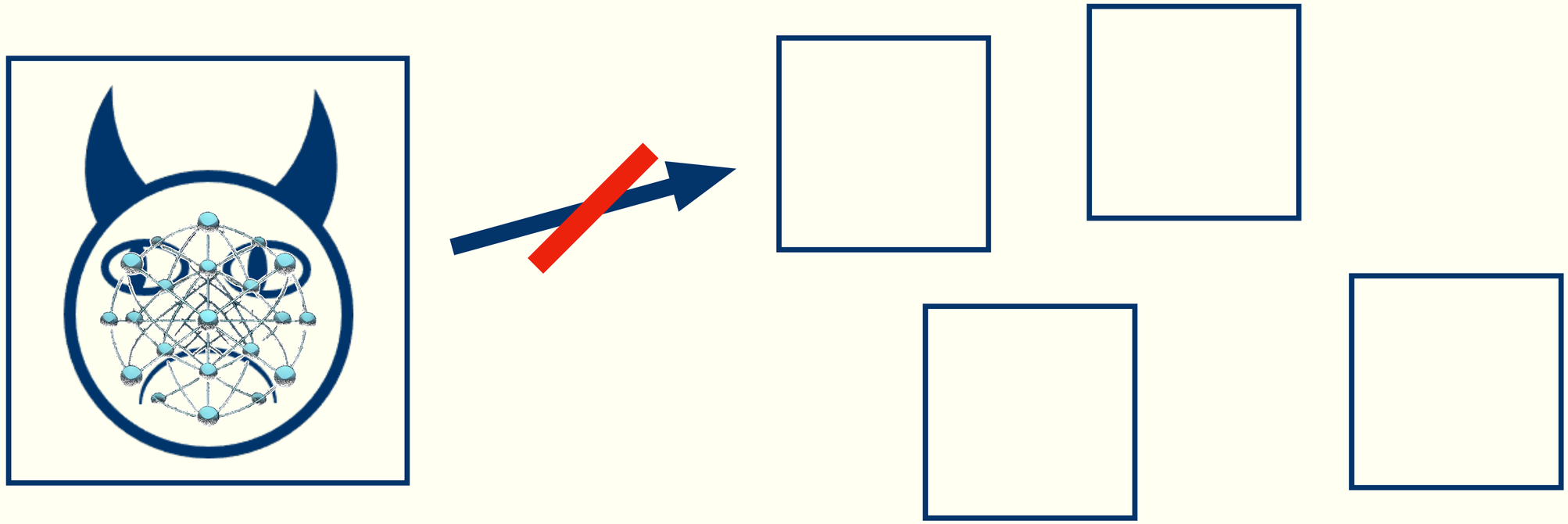
Quantum copy protection is a triple of algorithms (Setup, Generate, Compute), with the following interfaces:

- $\mathsf{Setup}() \mapsto \mathsf{sk}$
- $\mathsf{Generate}(\mathsf{sk}, f) \mapsto (\rho_f, \{U_{f,x}\})$
- $\mathsf{Compute}(\rho_f, \{U_{f,x}\}, x) \mapsto y$

What makes it unclonable is that an adversary should not be able to split ρ_f into two states ρ_1 and ρ_2 that both let you evaluate f!

AI exfiltration and intelligence explosion?

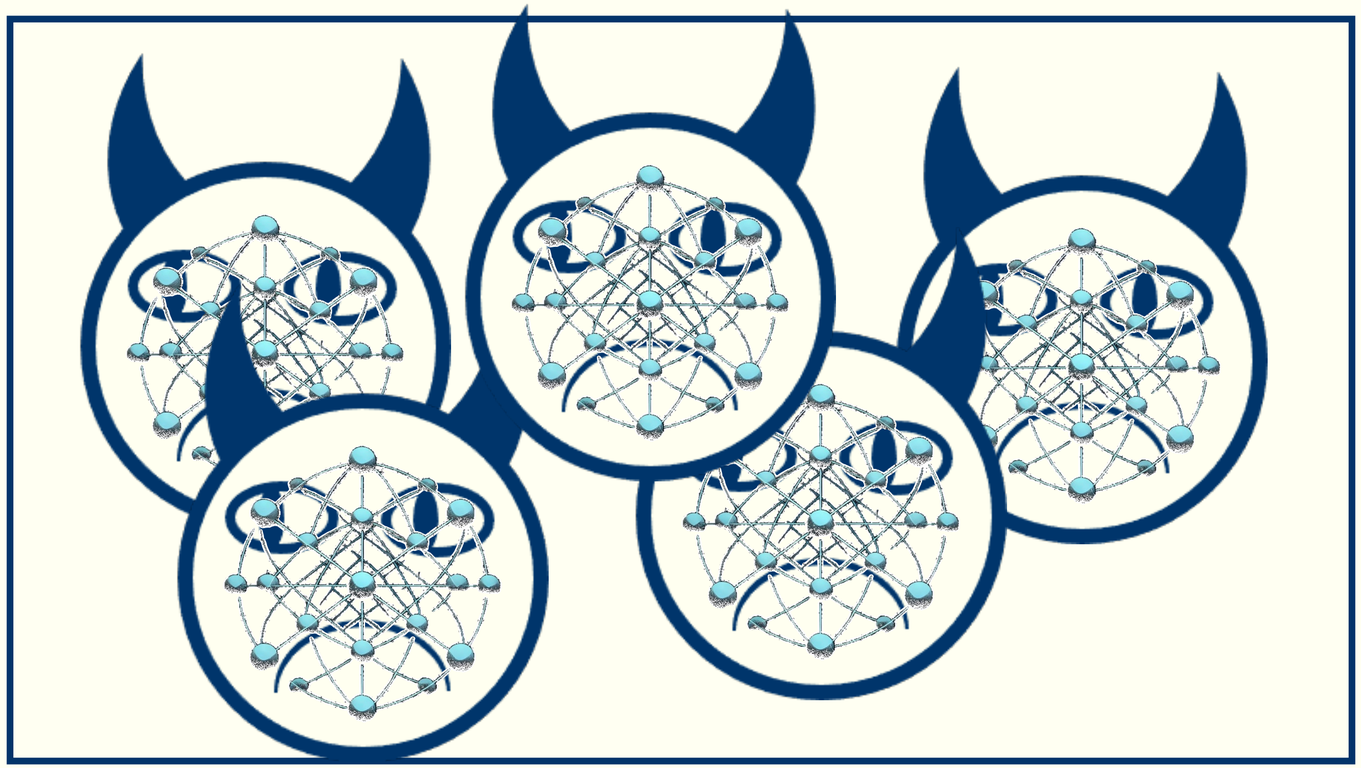
A big fear in AI safety (at least, as far as I understand) is that models might escape their original servers, and start running more and more instances of themselves…



What if the AI was being run as a copy-protected program?

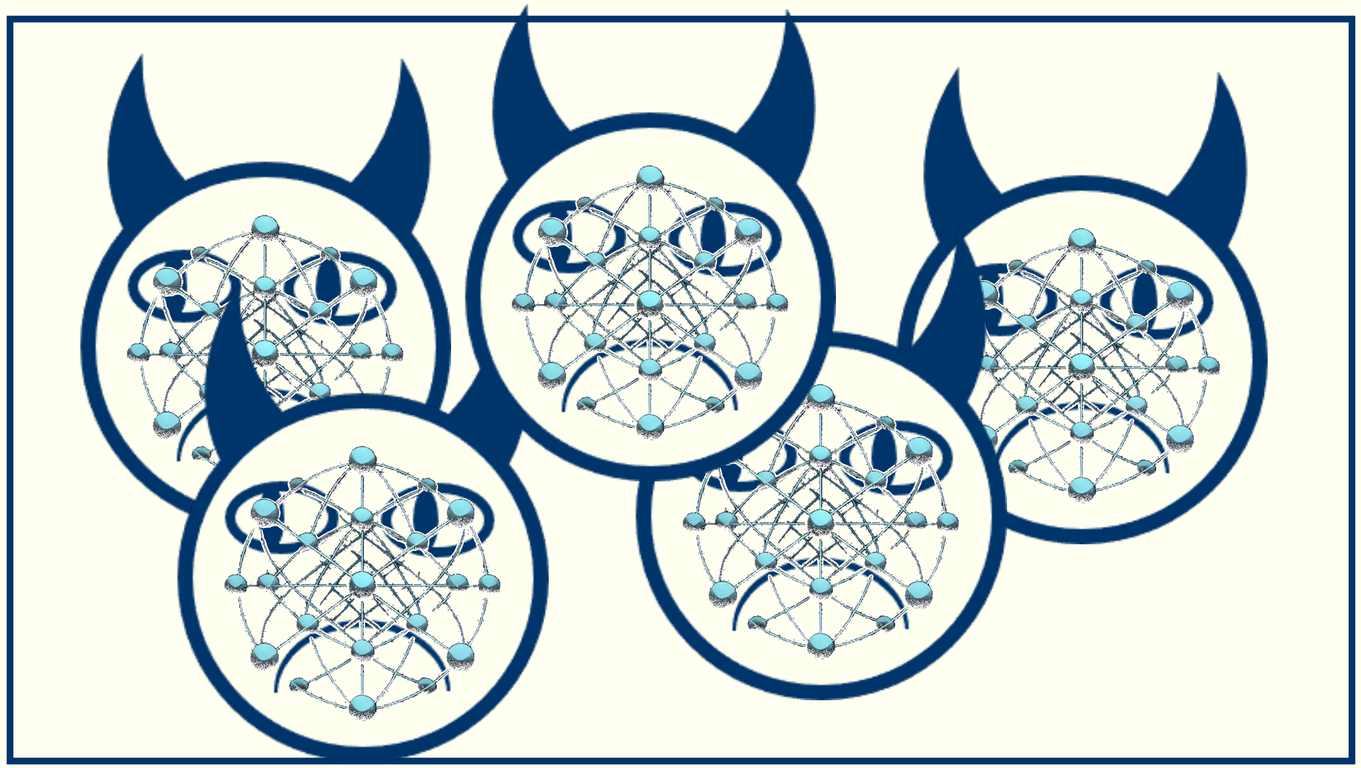
AI exfiltration and intelligence explosion?

What if we actually want to be able to run many copies of a AI agent on the same server (so we can service many requests), but we just don't want AI to escape…



AI exfiltration and intelligence explosion?

What if we actually want to be able to run many copies of a AI agent on the same server (so we can service many requests), but we just don't want AI to escape…



This motivates clonable, but not "telegraphicable" quantum states.

Quantum programming fire?

Quantum programming fire is a triple of algorithms (Setup, Generate, Compute, Clone), with the following interfaces:

- $\mathsf{Setup}() \mapsto \mathsf{sk}$
- $\mathsf{Generate}(\mathsf{sk}, f) \mapsto (\rho_f, \{U_{f,x}\})$
- $\mathsf{Compute}(\rho_f, \{U_{f,x}\}, x) \mapsto y$
- $\mathsf{Clone}(\rho_f) \mapsto \rho_f^{\otimes 2}$ (It's actually of if they are entangled)

Quantum programming fire?

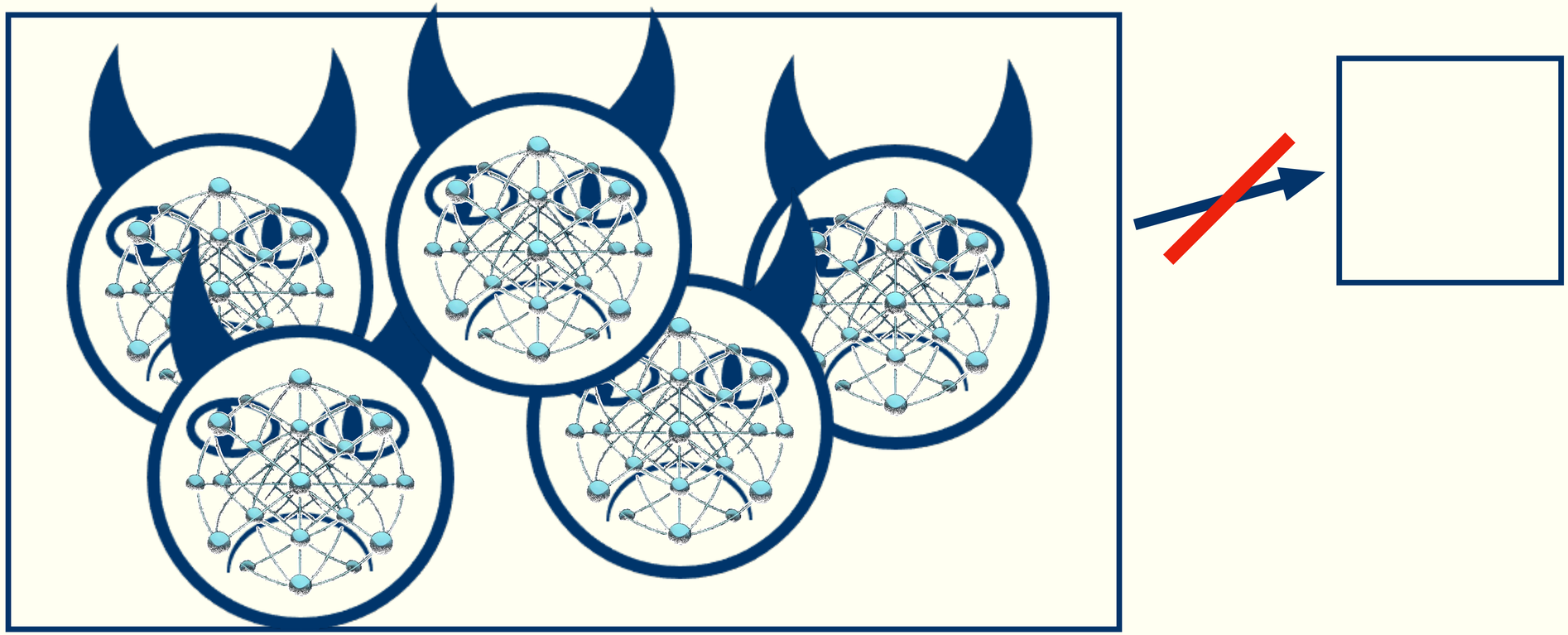
Quantum programming fire is a triple of algorithms (Setup, Generate, Compute, Clone), with the following interfaces:

- $\mathsf{Setup}() \mapsto \mathsf{sk}$
- $\mathsf{Generate}(\mathsf{sk}, f) \mapsto (\rho_f, \{U_{f,x}\})$
- $\mathsf{Compute}(\rho_f, \{U_{f,x}\}, x) \mapsto y$
- $\mathsf{Clone}(\rho_f) \mapsto \rho_f^{\otimes 2}$ (It's actually of if they are entangled)

Un-telegraphicable means that no efficient adversary should be able to produce a classical message m that can be used to prepare a copy of any valid program state!

AI exfiltration and intelligence explosion?

What if we actually want to be able to run many copies of a AI agent on the same server (so we can service many requests), but we just don't want AI to escape…



This motivates clonable, but not "telegraphicable" quantum states.

AI exfiltration and intelligence explosion?

But… copy protection (and quantum fire) are only secure if the underlying program is **hard to learn** from it's input and output behavior. (For example, someone could just train their own version of the same model if they could do this).

AI exfiltration and intelligence explosion?

But… copy protection (and quantum fire) are only secure if the underlying program is **hard to learn** from it's input and output behavior. (For example, someone could just train their own version of the same model if they could do this).

My two cents: It's probably much more costly to distill a model that it would be to just transfer the weights, so this still buys you something.

AI exfiltration and intelligence explosion?

But… copy protection (and quantum fire) are only secure if the underlying program is **hard to learn** from it's input and output behavior. (For example, someone could just train their own version of the same model if they could do this).

My two cents: It's probably much more costly to distill a model that it would be to just transfer the weights, so this still buys you something.

Putting the whole model into a quantum computer would seem tough though… Are there more practical ways to copy protect LLMs?

Can you think of other unexpected uses of quantum computers?