The Real-Ideal Paradigm for Systems

A-Exam

Suraaj Kanniwadi







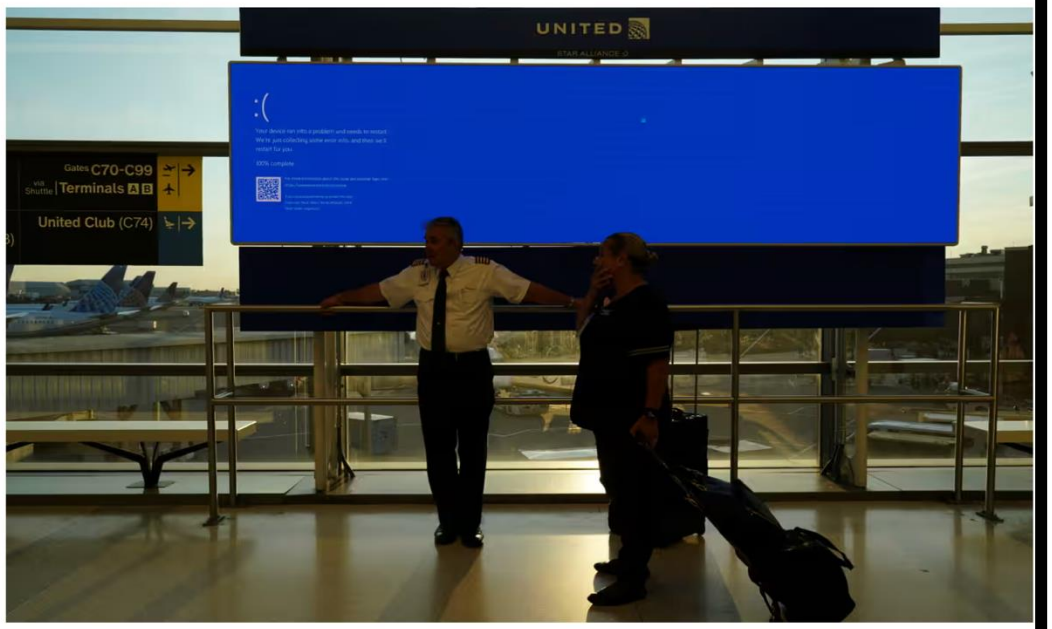
We need Secure Systems

We need Secure Systems

US transportation, police and hospital systems stricken by global CrowdStrike IT outage

US wakes up to Microsoft system collapse from software update that has crippled world IT systems

Windows global IT outage: what we know so far

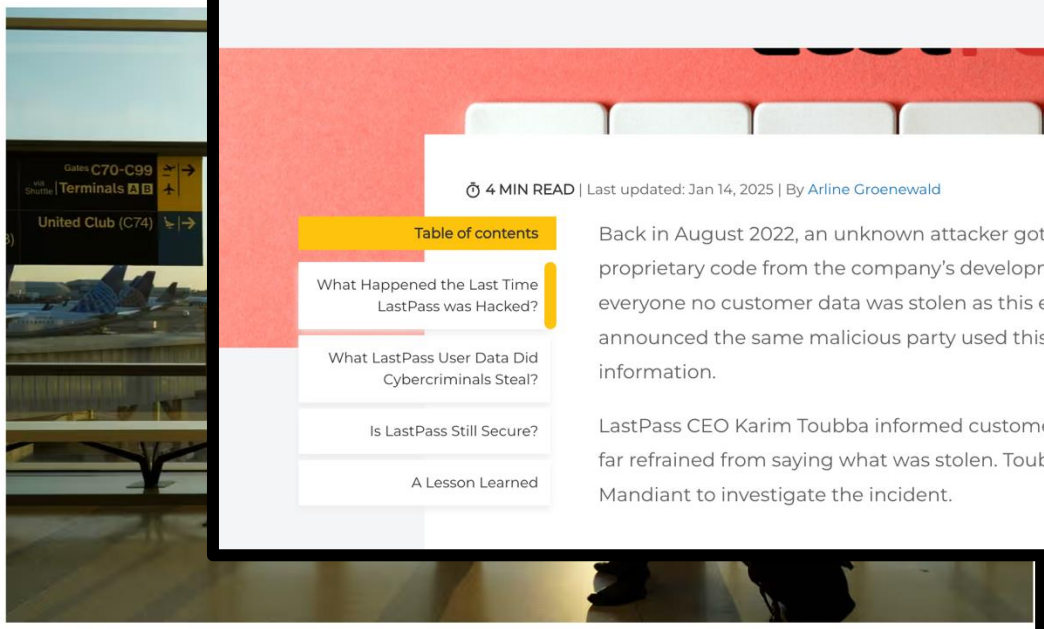


A Microsoft 'blue screen of death' us seen inside New Jersey's Newark international airport after CrowdStrike software crashed systems globally on Friday. Photograph: Bing Guan/Reuters

We need Secure Systems

US transportation, police and hospital systems stricken by global CrowdStrike IT outage

US wakes up to
update that has
Windows glo



A Microsoft 'blue screen of death' us seen inside New Jersey's Newark international airport after CrowdStrike software crashed systems globally on Friday. Photograph: Bing Guan/Reuters

Password Manager LastPass Suffers Data Breach After August Security Breach

4 MIN READ | Last updated: Jan 14, 2025 | By Arline Groenewald

Table of contents

- What Happened the Last Time LastPass was Hacked?
- What LastPass User Data Did Cybercriminals Steal?
- Is LastPass Still Secure?
- A Lesson Learned

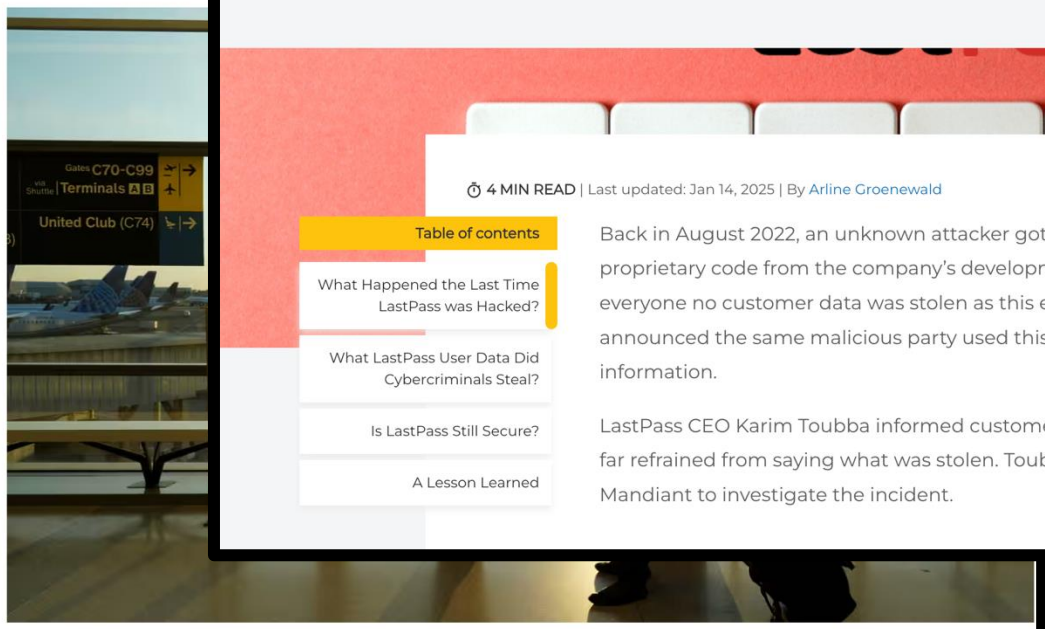
Back in August 2022, an unknown attacker got into LastPass's security systems and stole some proprietary code from the company's development environment. At the time, the company assured everyone no customer data was stolen as this environment doesn't contain any. Now, the company announced the same malicious party used this stolen data to access "certain elements" of customers' information.

LastPass CEO Karim Toubba informed customers of the breach in a November 30 blog post, but has so far refrained from saying what was stolen. Toubba simply stated LastPass is working with security firm Mandiant to investigate the incident.

We need Secure Systems

US transportation, police and hospital systems stricken by global CrowdStrike IT outage

US wakes up to
update that has
Windows glo



A Microsoft 'blue screen of death' us seen inside New Jersey's Newark international airport after CrowdStrike software crashed systems globally on Friday. Photograph: Bing Guan/Reuters

Password Manager LastPass Suffer
Security Bre

4 MIN READ | Last updated: Jan 14, 2025 | By Arline Groenewald

Table of contents

- What Happened the Last Time LastPass was Hacked?
- What LastPass User Data Did Cybercriminals Steal?
- Is LastPass Still Secure?
- A Lesson Learned

Back in August 2022, an unknown attacker got into Last
proprietary code from the company's development envi
everyone no customer data was stolen as this environm
announced the same malicious party used this stolen da
information.

LastPass CEO Karim Toubba informed customers of the
far refrained from saying what was stolen. Toubba simpl
Mandiant to investigate the incident.

THE DAO HACK

The DAO was a decentralized autonomous organization on Ethereum, designed to manage investments through smart contracts, without traditional intermediaries.

JUNE 17, 2016

An attacker exploited a vulnerability in the DAO's smart contract.

– $60 MILLION

Drained in ETH, causing widespread panic in the Ethereum community.

HOW WAS THE DAO HACKED?

Reentrancy attack

The attacker used a "reentrancy attack," repeatedly calling the DAO's withdrawal function before the contract's balance updated. This flaw allowed funds to be siphoned without reducing the balance.

WHO HACKED THE DAO?

Unknown

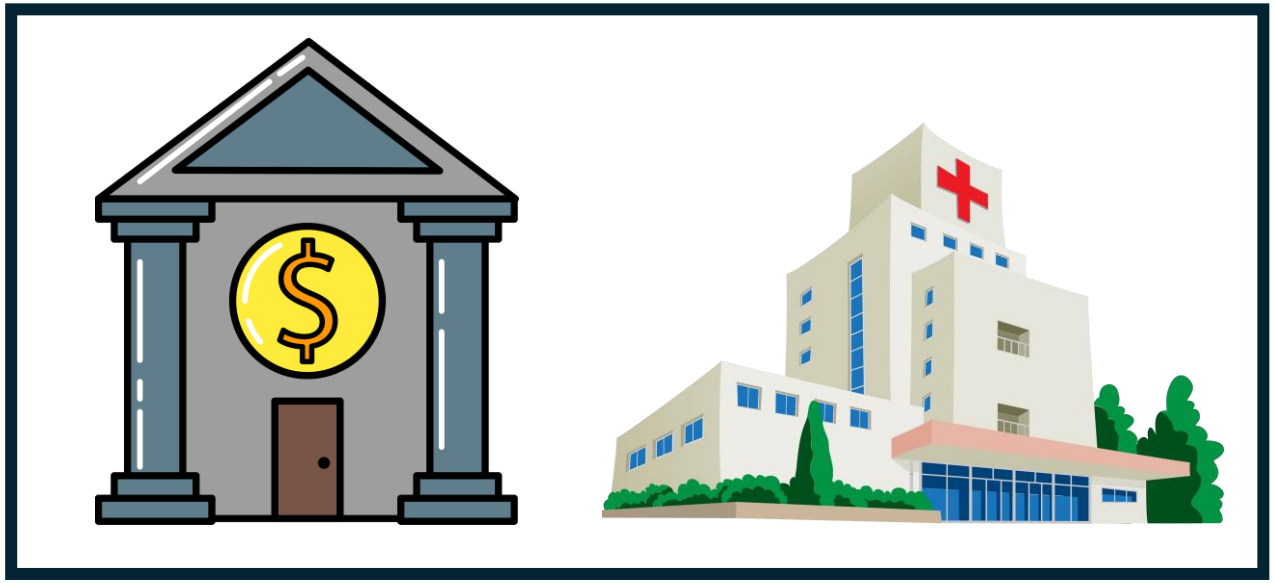
The identity of the DAO hacker remains unknown. Despite extensive investigation, the attacker has not been identified, and theories about who might be responsible are largely speculative.

The First Step…



The First Step…





The First Step…

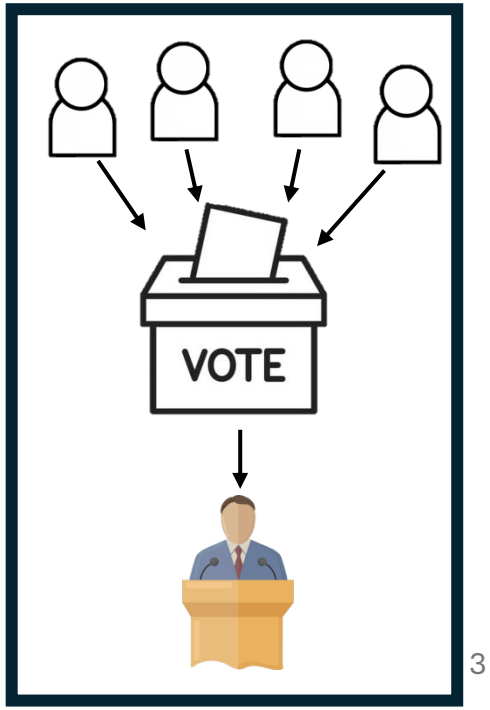




The First Step...



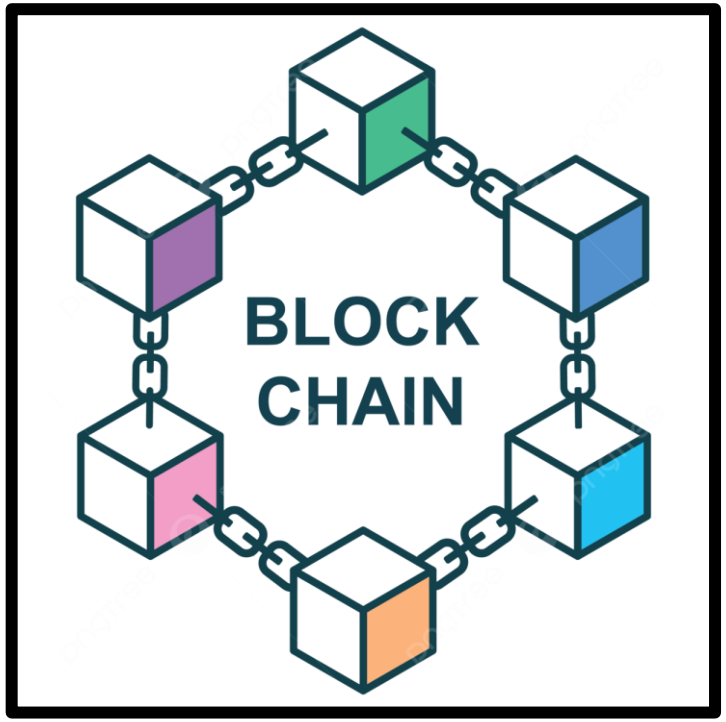


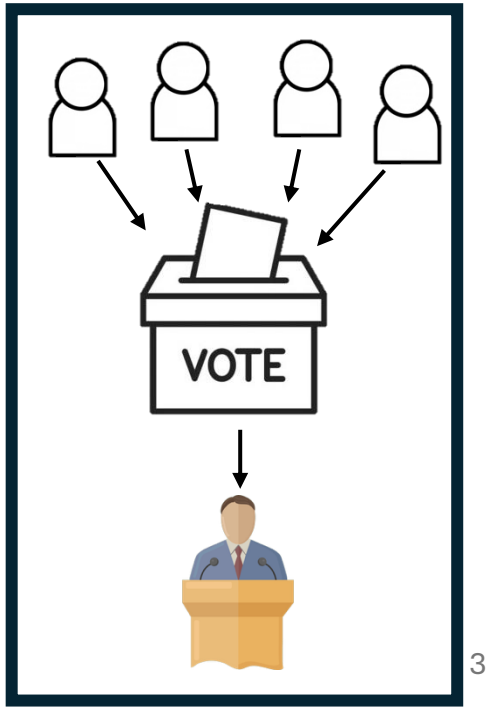


The First Step...









Well-known Specification Techniques

Well-known Specification Techniques

Trace Refinement

The existence of refinement mappings

Authors: Martín Abadi, Leslie Lamport | Authors Info & Claims

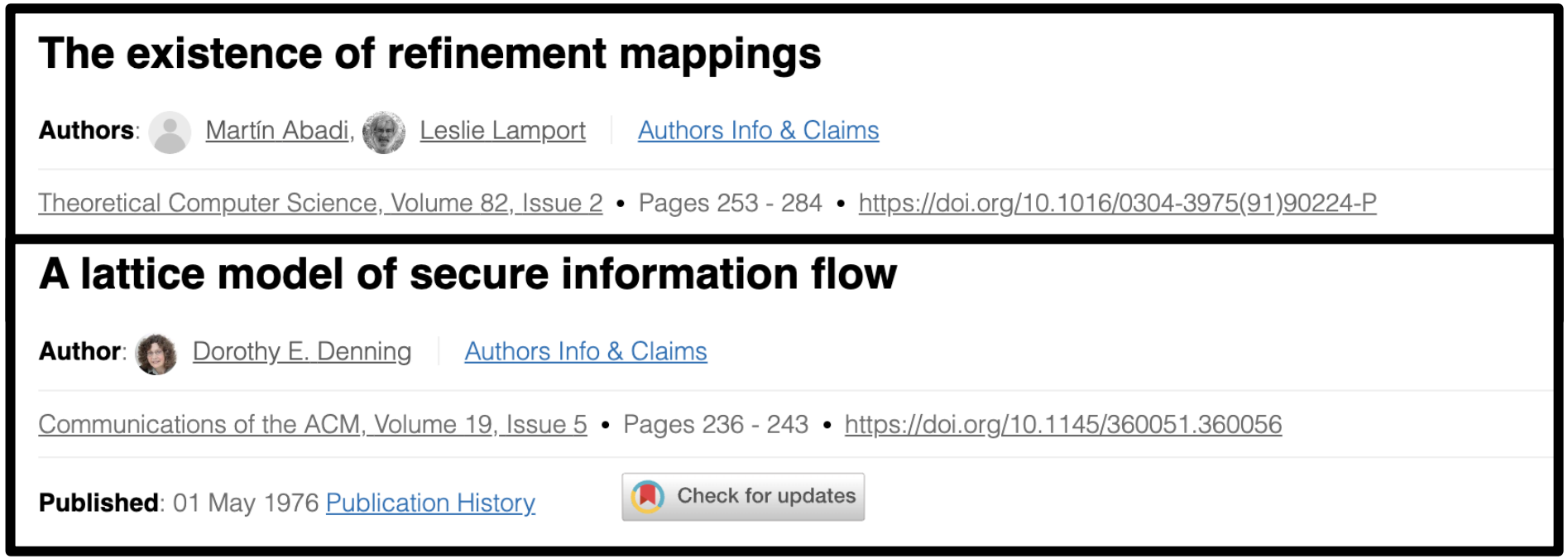
Theoretical Computer Science, Volume 82, Issue 2 • Pages 253 - 284 • https://doi.org/10.1016/0304-3975(91)90224-P

Published: 31 May 1991 Publication History

Well-known Specification Techniques

Trace Refinement

Noninterference



The existence of refinement mappings

Authors: Martín Abadi, Leslie Lamport | Authors Info & Claims

Theoretical Computer Science, Volume 82, Issue 2 • Pages 253 - 284 • https://doi.org/10.1016/0304-3975(91)90224-P

A lattice model of secure information flow

Author: Dorothy E. Denning | Authors Info & Claims

Communications of the ACM, Volume 19, Issue 5 • Pages 236 - 243 • https://doi.org/10.1145/360051.360056

Published: 01 May 1976 Publication History

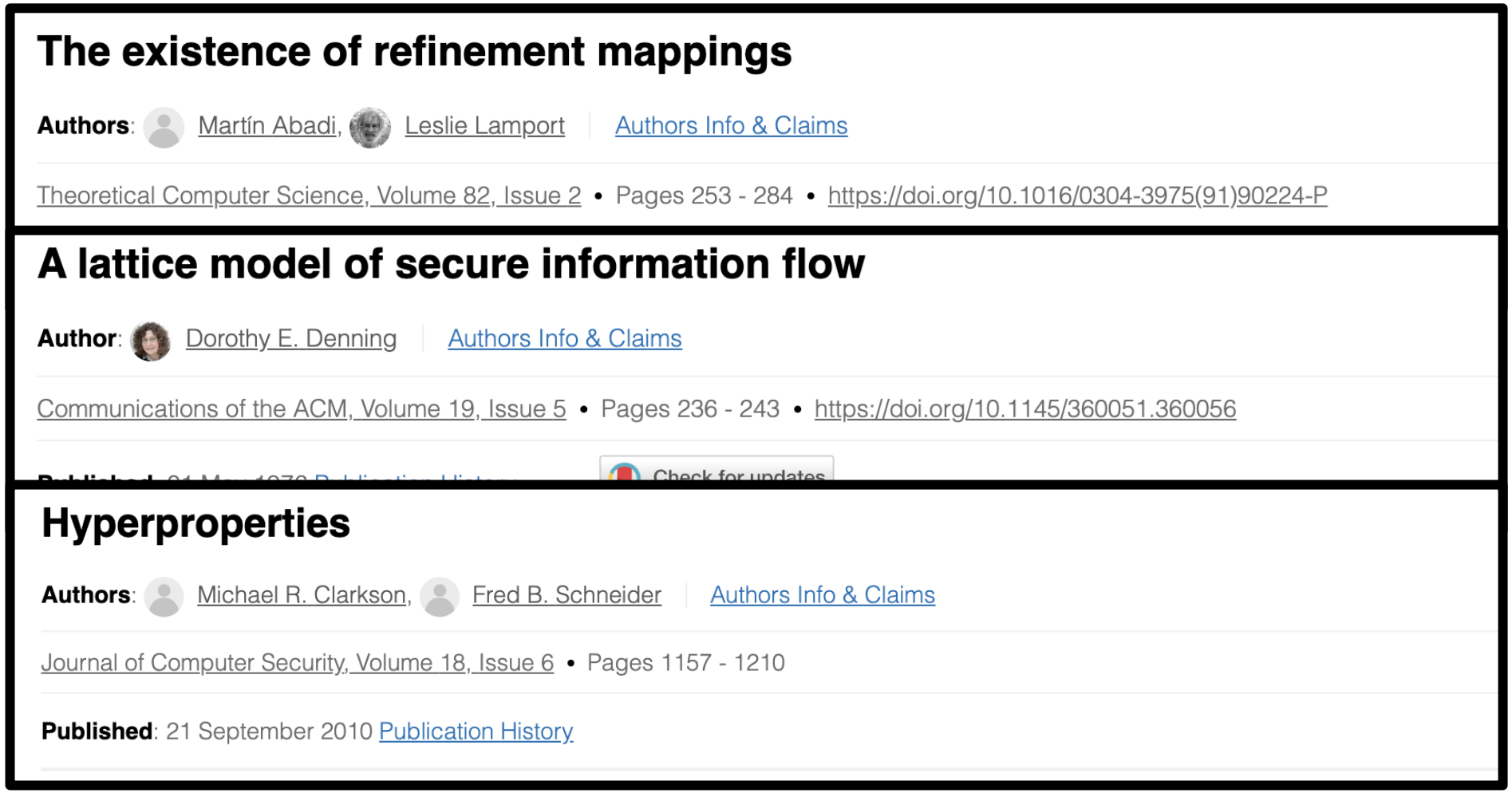
Check for updates

Well-known Specification Techniques

Trace Refinement

Noninterference

Hyperproperties



The existence of refinement mappings

Authors: Martín Abadi, Leslie Lamport | Authors Info & Claims

Theoretical Computer Science, Volume 82, Issue 2 • Pages 253 - 284 • https://doi.org/10.1016/0304-3975(91)90224-P

A lattice model of secure information flow

Author: Dorothy E. Denning | Authors Info & Claims

Communications of the ACM, Volume 19, Issue 5 • Pages 236 - 243 • https://doi.org/10.1145/360051.360056

Check for updates

Hyperproperties

Authors: Michael R. Clarkson, Fred B. Schneider | Authors Info & Claims

Journal of Computer Security, Volume 18, Issue 6 • Pages 1157 - 1210

Published: 21 September 2010 Publication History

Well-known Specification Techniques



Trace Refinement

The existence of refinement mappings

Authors: Martín Abadi, Leslie Lamport | Authors Info & Claims

Theoretical Computer Science, Volume 82, Issue 2 • Pages 253 - 284 • https://doi.org/10.1016/0304-3975(91)90224-P

Noninterference

A lattice model of secure information flow

Author: Dorothy E. Denning | Authors Info & Claims

Communications of the ACM, Volume 19, Issue 5 • Pages 236 - 243 • https://doi.org/10.1145/360051.360056

Check for updates

Hyperproperties

Hyperproperties

Authors: Michael R. Clarkson, Fred B. Schneider | Authors Info & Claims

Journal of Computer Security, Volume 18, Issue 6 • Pages 1157 - 1210

Published: 21 September 2010 Publication History

Well-known Specification Techniques



Trace Refinement

Noninterference

Hyperproperties

The existence of refinement mappings

Authors: Martín Abadi, Leslie Lamport | Authors Info & Claims

Theoretical Computer Science, Volume 82, Issue 2 • Pages 253 - 284 • https://doi.org/10.1016/0304-3975(91)90224-P

A lattice model of secure information flow

Author: Dorothy E. Denning | Authors Info & Claims

Communications of the ACM, Volume 19, Issue 5 • Pages 236 - 243 • https://doi.org/10.1145/360051.360056

Check for updates

Hyperproperties

Authors: Michael R. Clarkson, Fred B. Schneider | Authors Info & Claims

Journal of Computer Security, Volume 18, Issue 6 • Pages 1157 - 1210

Published: 21 September 2010 Publication History

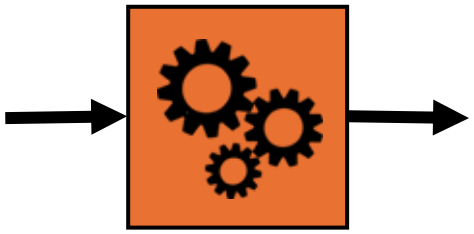
Trace Refinement

Trace Refinement

“Specification is collection of permitted single executions”

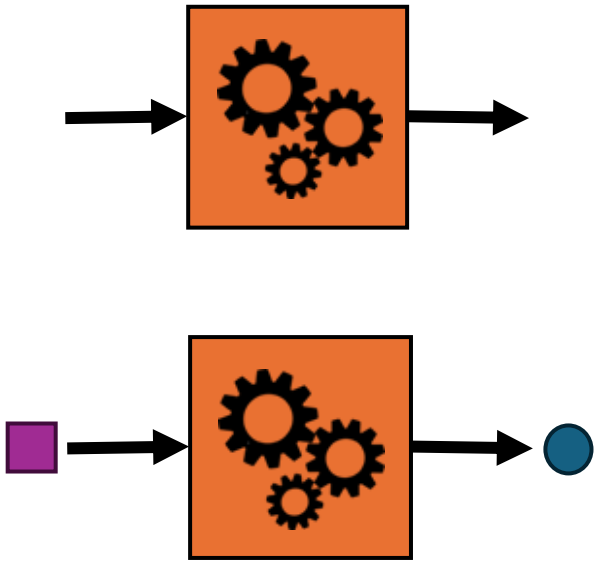
Trace Refinement

“Specification is collection of permitted single executions”



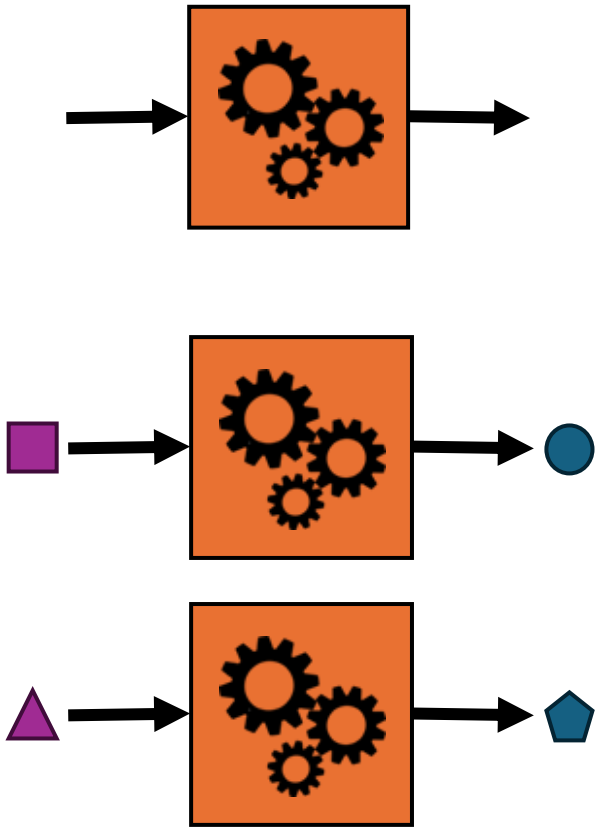
Trace Refinement

“Specification is collection of permitted single executions”



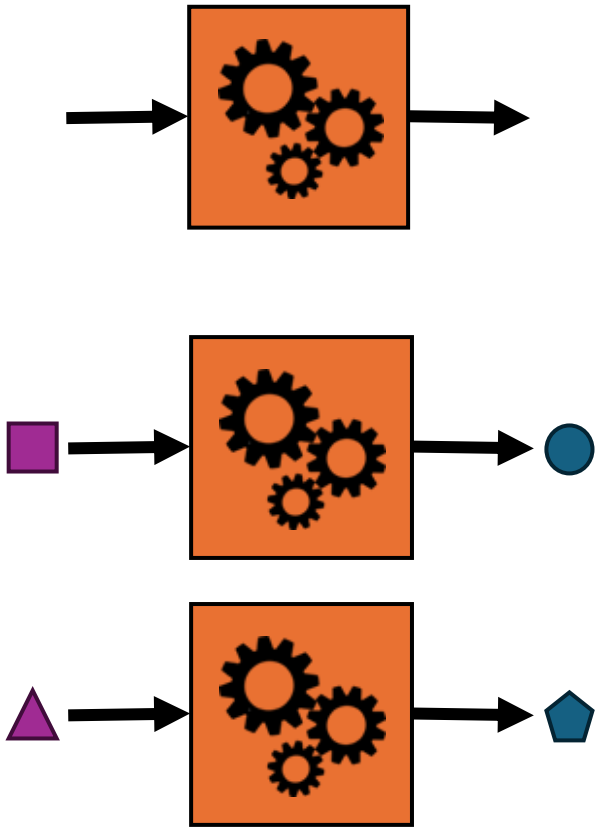
Trace Refinement

“Specification is collection of permitted single executions”



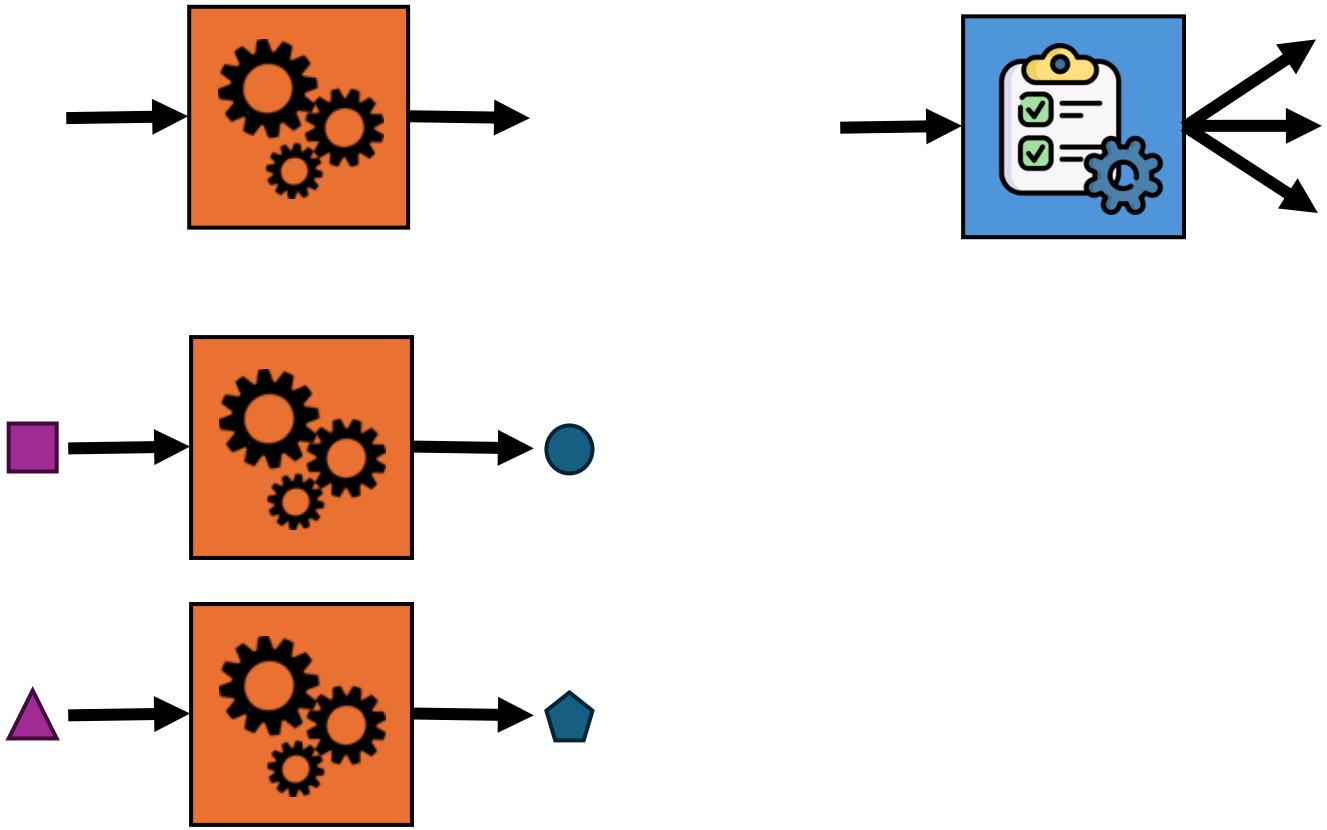
Trace Refinement

“Specification is collection of permitted single executions”



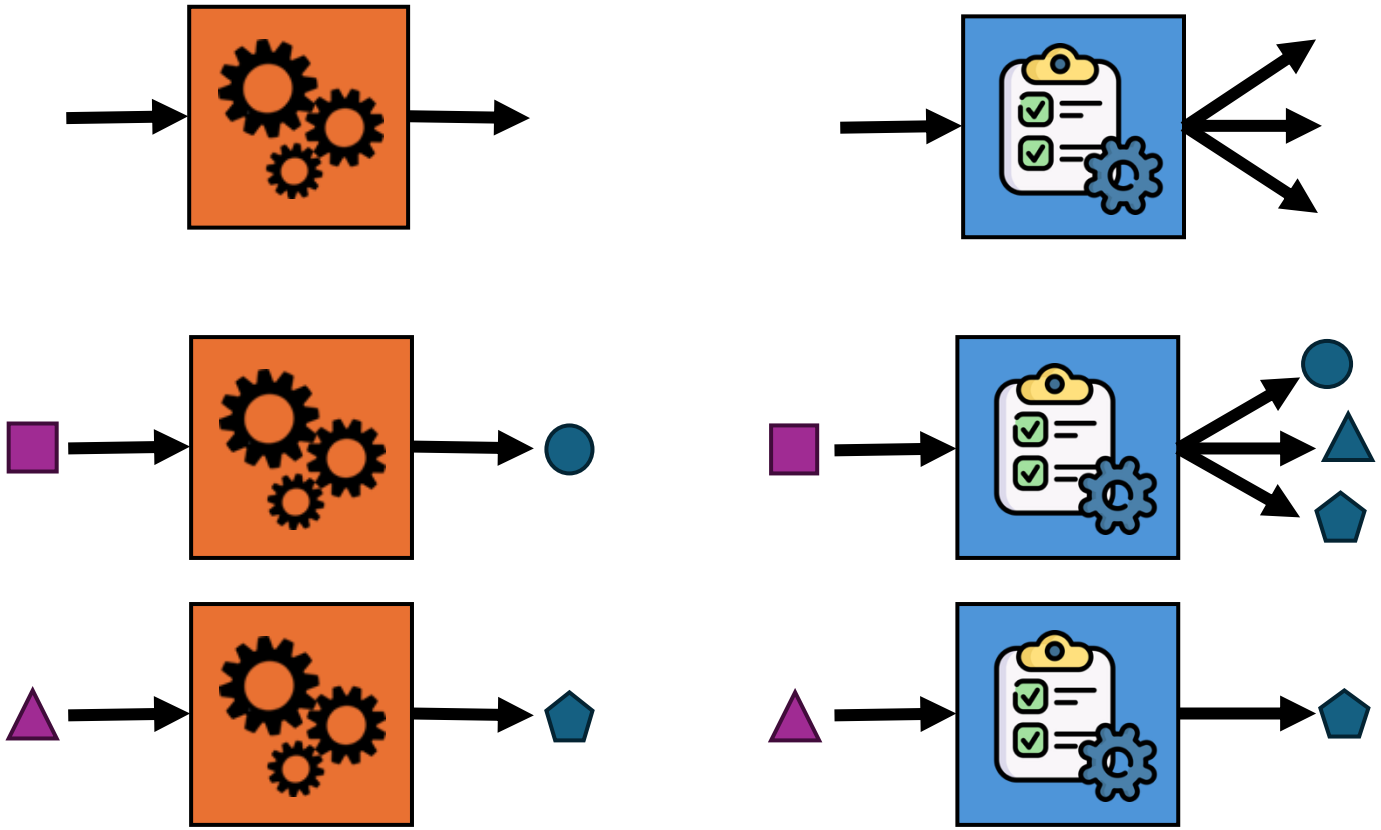
Trace Refinement

“Specification is collection of permitted single executions”



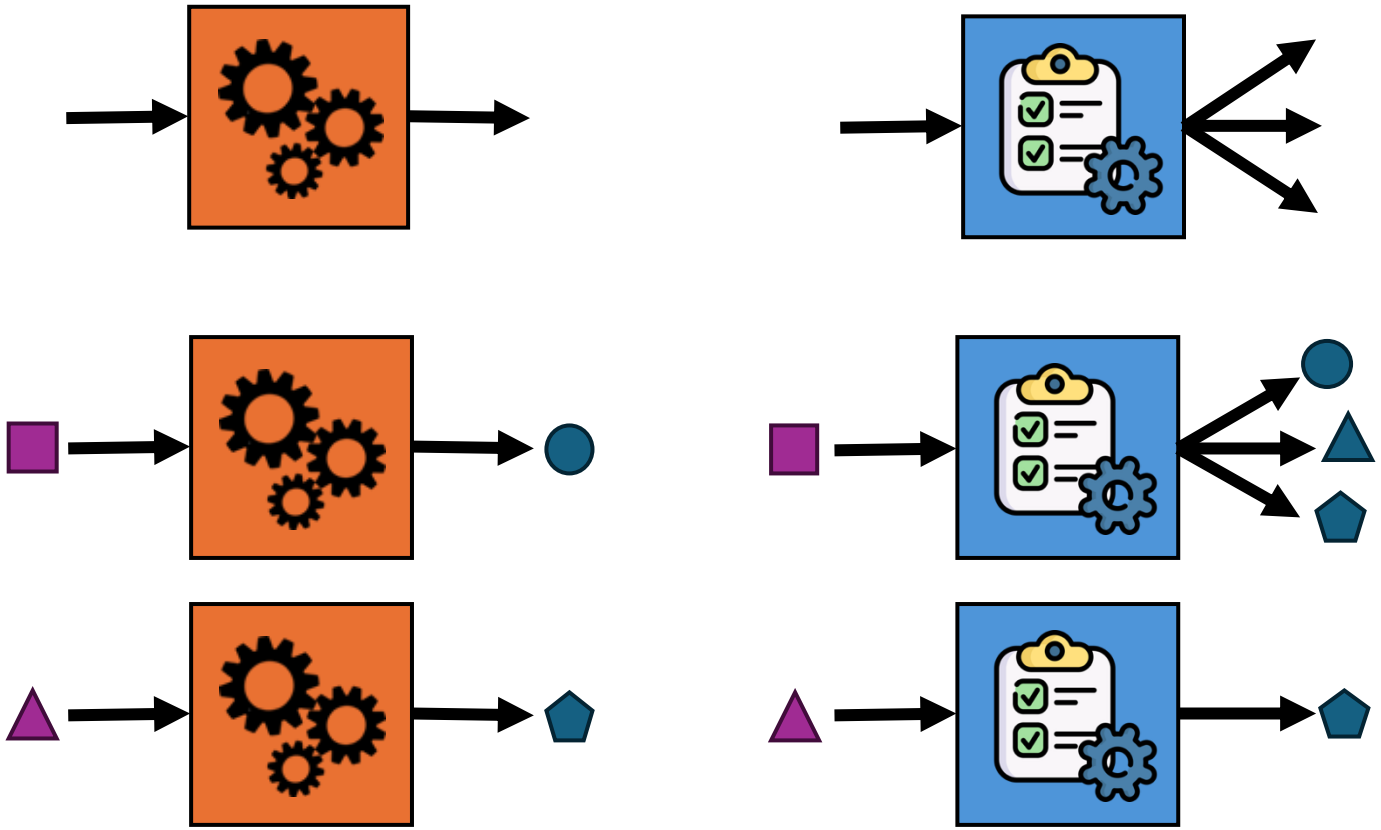
Trace Refinement

“Specification is collection of permitted single executions”



Trace Refinement

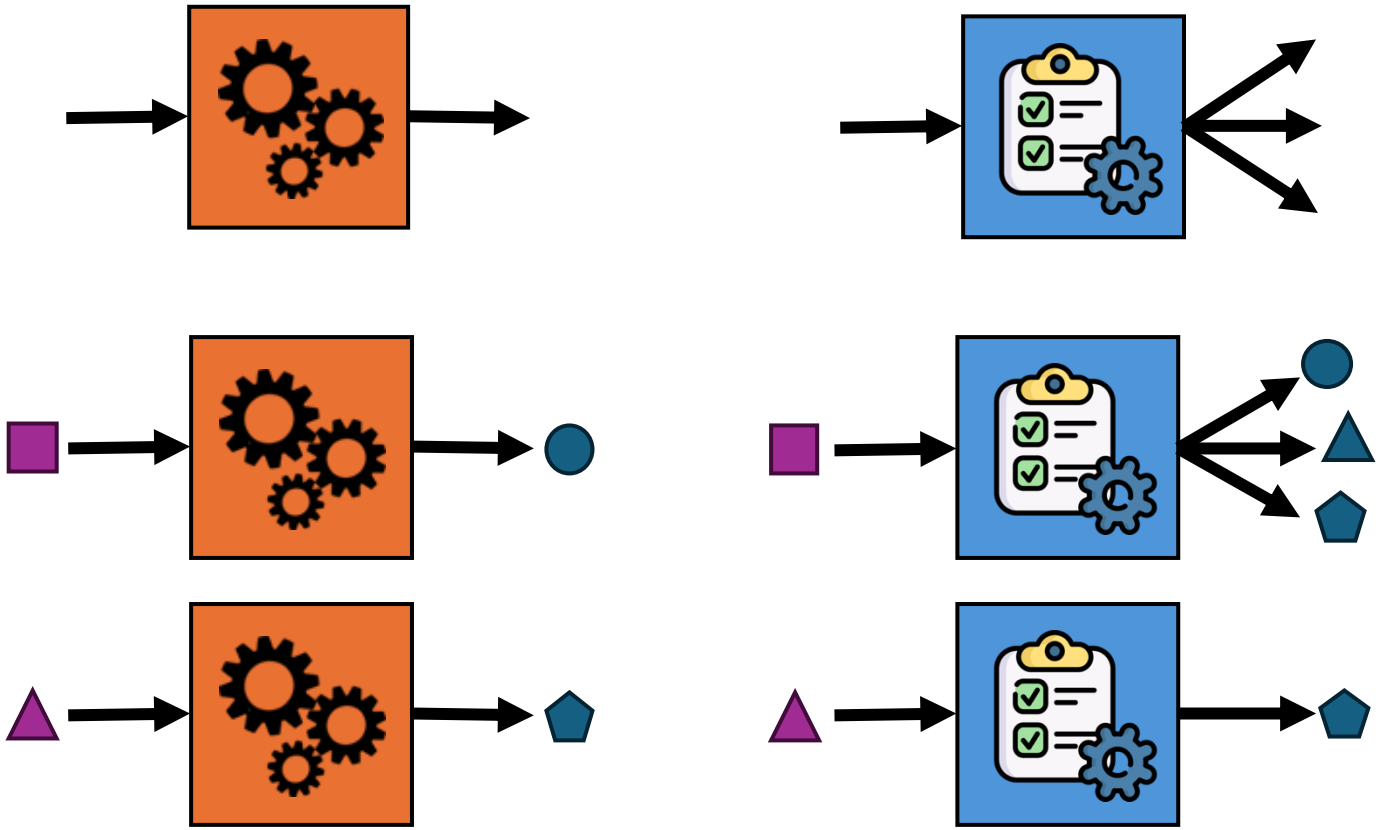
“Specification is collection of permitted single executions”





Trace Refinement

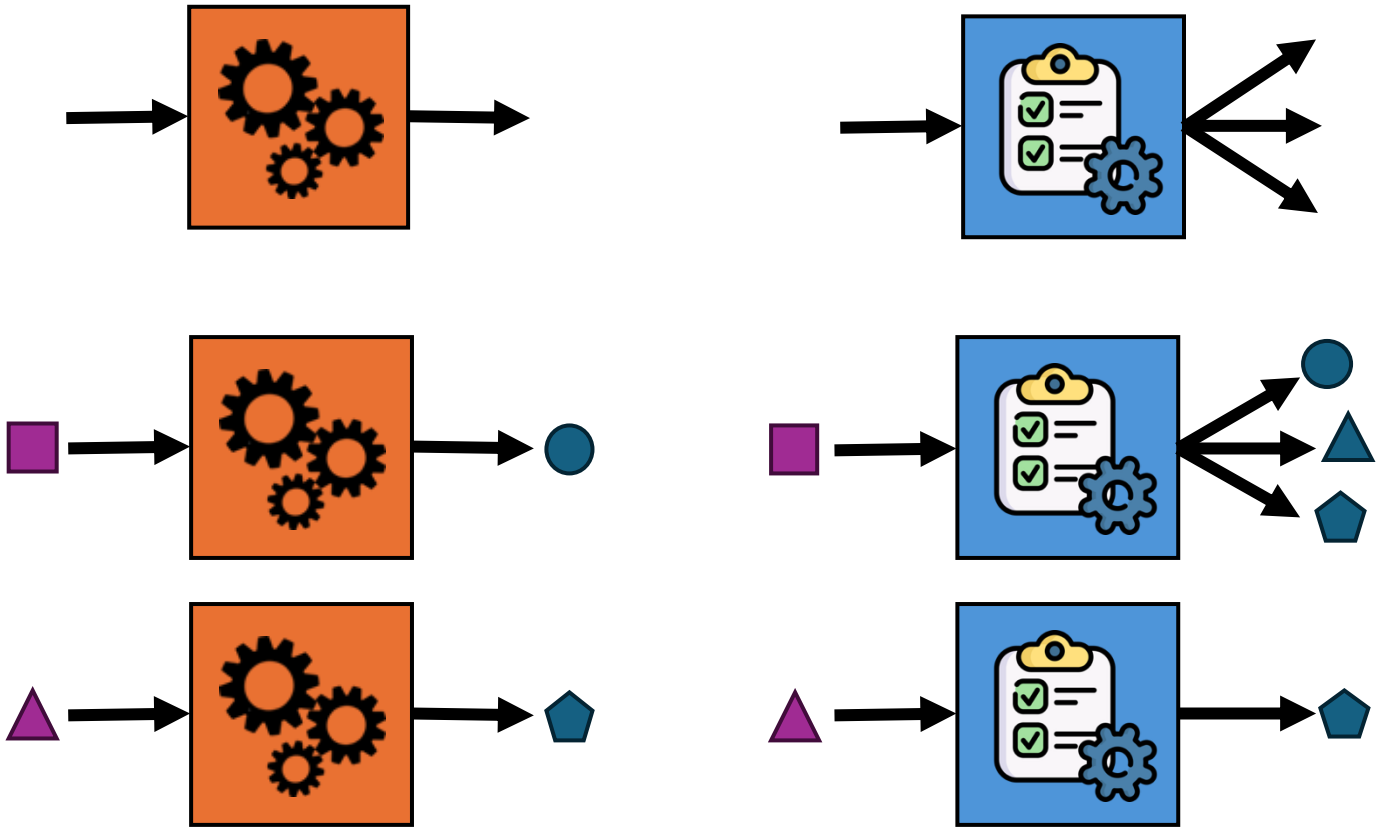
“Specification is collection of permitted single executions”





Trace Refinement

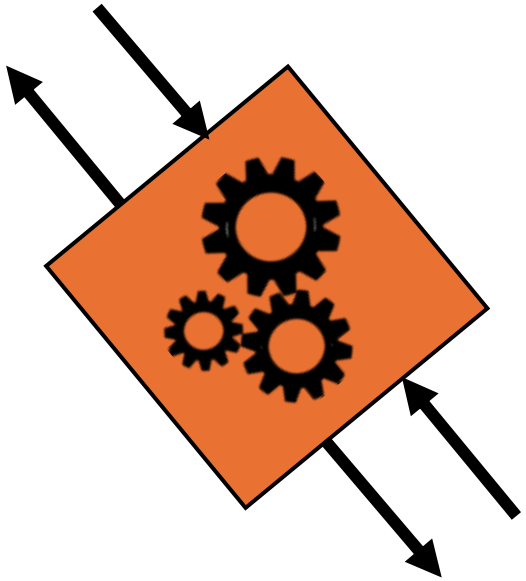
“Specification is collection of permitted single executions”



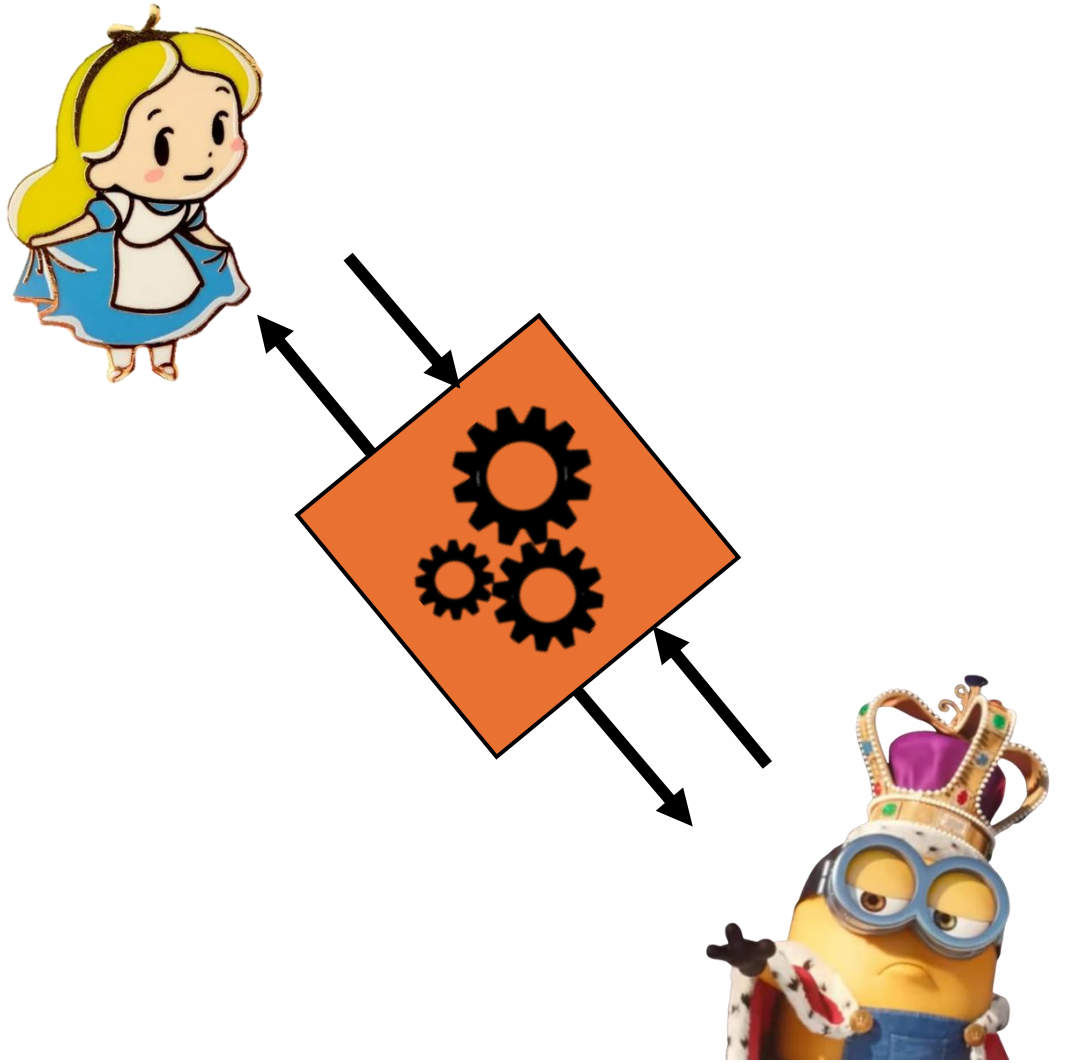


Good, but not enough..

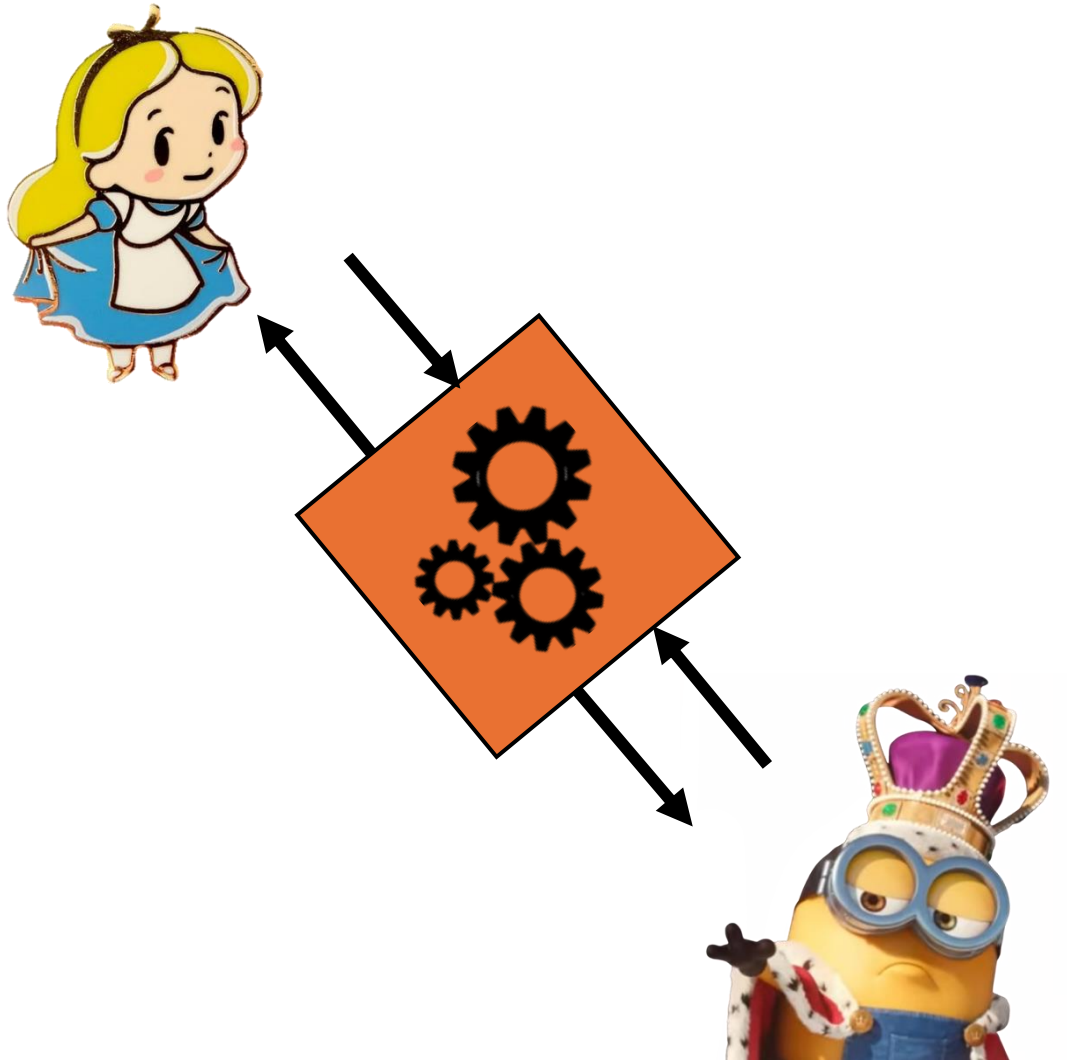
Good, but not enough..



Good, but not enough..

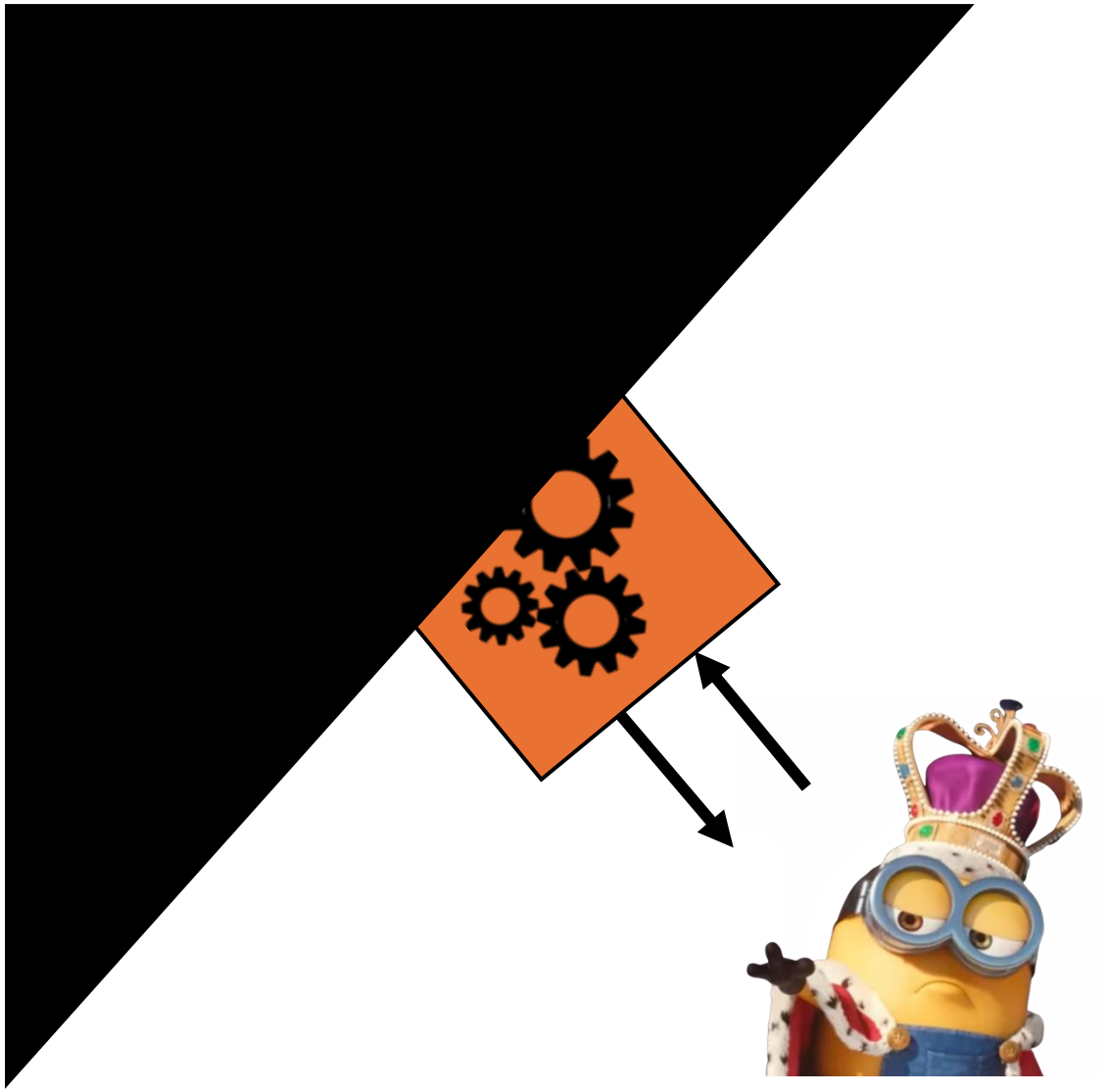


Good, but not enough..



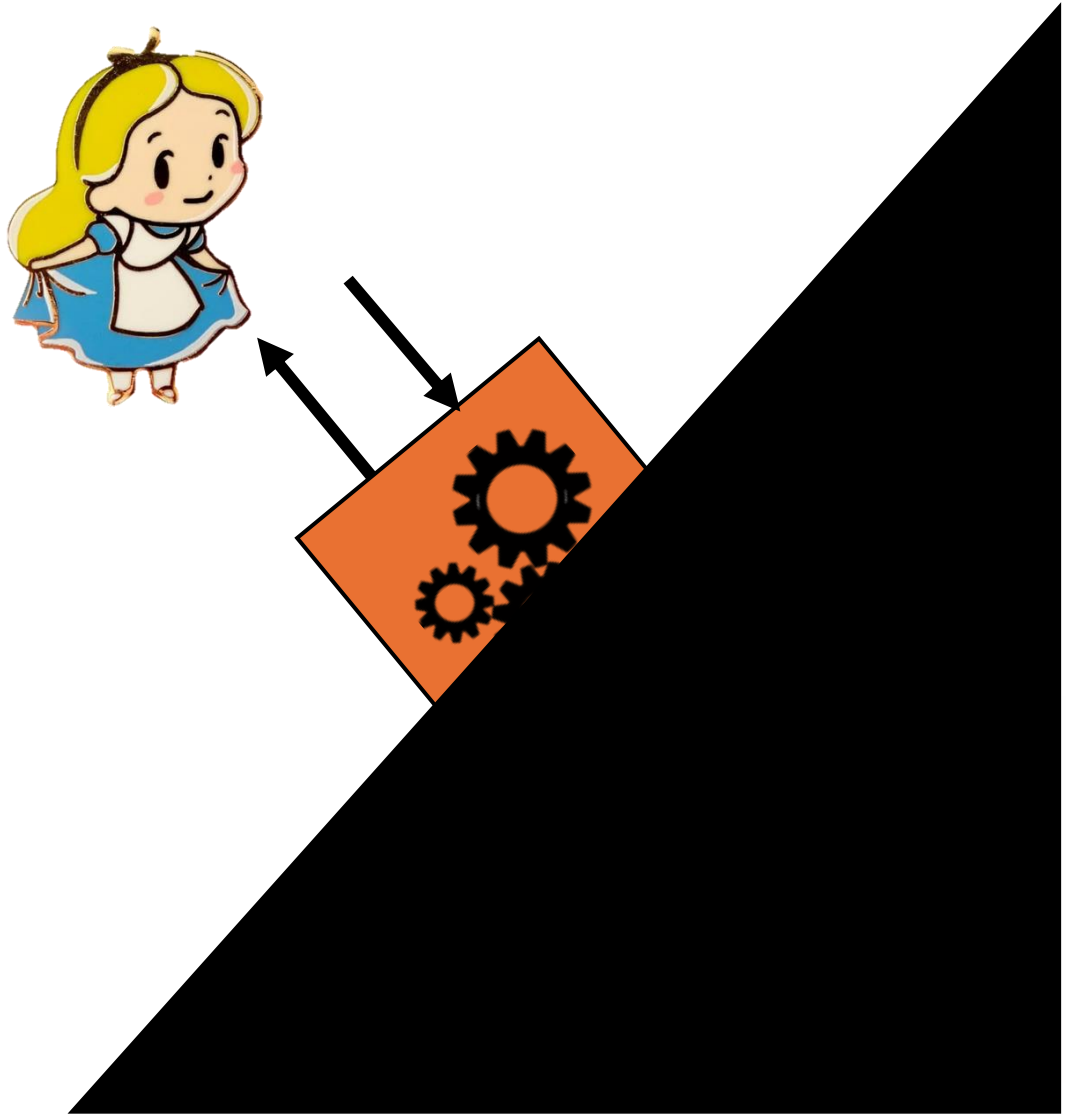


Good, but not enough..



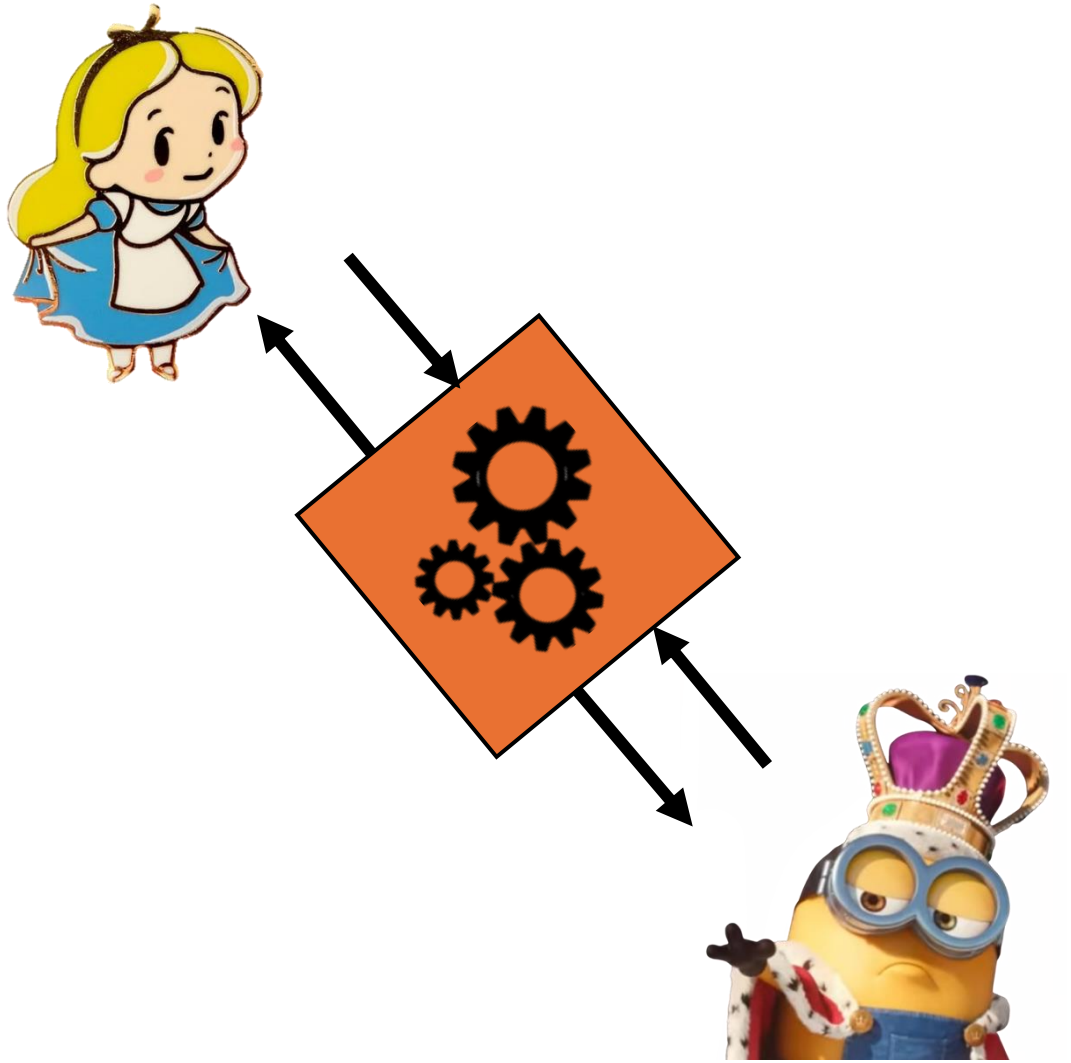


Good, but not enough..



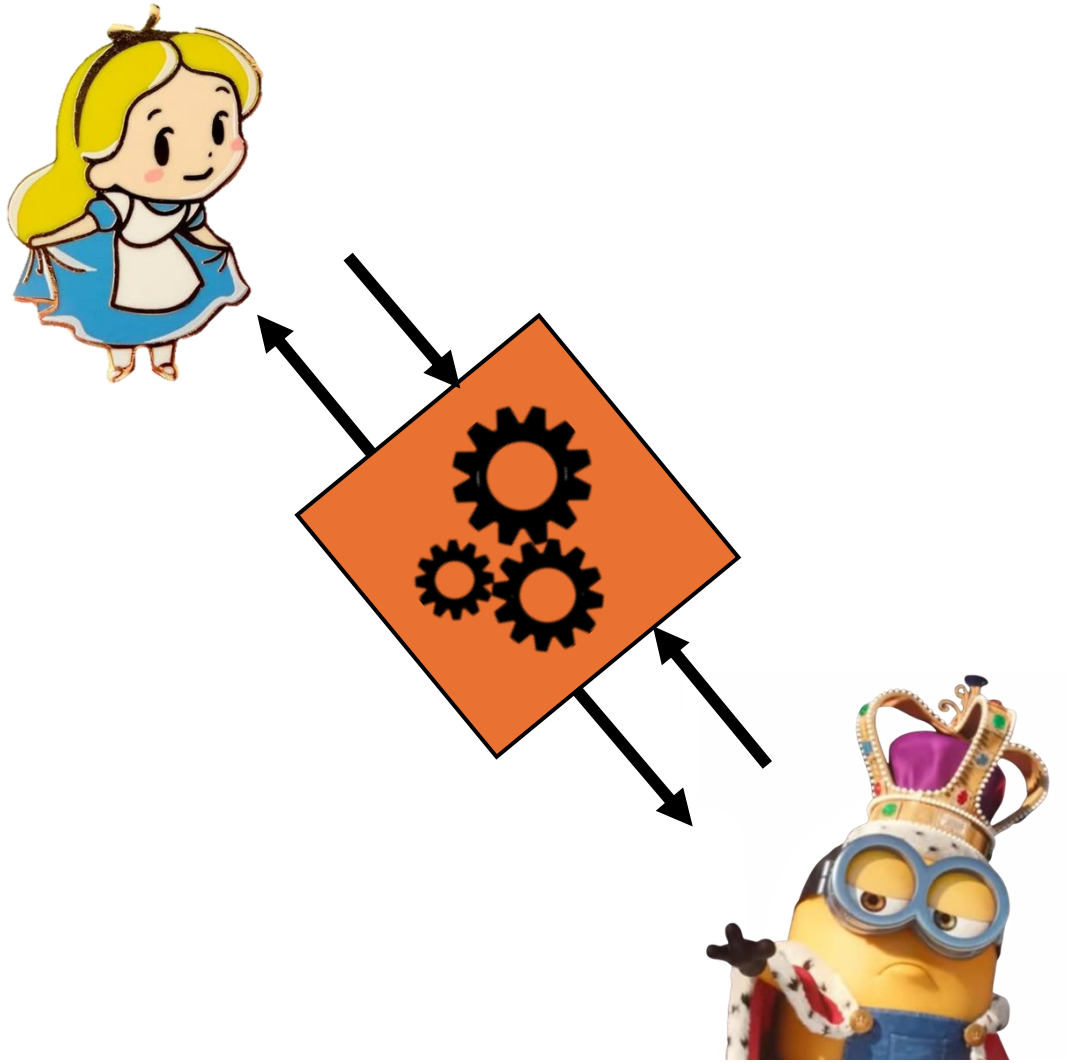


Good, but not enough..





Good, but not enough..





Isolation cannot be expressed as a Trace Refinement Specification!



Well-known Specification Techniques



Trace Refinement

Noninterference

Hyperproperties

The existence of refinement mappings

Authors: Martín Abadi, Leslie Lamport | Authors Info & Claims

Theoretical Computer Science, Volume 82, Issue 2 • Pages 253 - 284 • https://doi.org/10.1016/0304-3975(91)90224-P

A lattice model of secure information flow

Author: Dorothy E. Denning | Authors Info & Claims

Communications of the ACM, Volume 19, Issue 5 • Pages 236 - 243 • https://doi.org/10.1145/360051.360056

Check for updates

Hyperproperties

Authors: Michael R. Clarkson, Fred B. Schneider | Authors Info & Claims

Journal of Computer Security, Volume 18, Issue 6 • Pages 1157 - 1210

Published: 21 September 2010 Publication History

Well-known Specification Techniques



Trace Refinement

Noninterference

Hyperproperties

The existence of refinement mappings

Authors: Martín Abadi, Leslie Lamport | Authors Info & Claims

Theoretical Computer Science, Volume 82, Issue 2 • Pages 253 - 284 • https://doi.org/10.1016/0304-3975(91)90224-P

A lattice model of secure information flow

Author: Dorothy E. Denning | Authors Info & Claims

Communications of the ACM, Volume 19, Issue 5 • Pages 236 - 243 • https://doi.org/10.1145/360051.360056

Check for updates

Hyperproperties

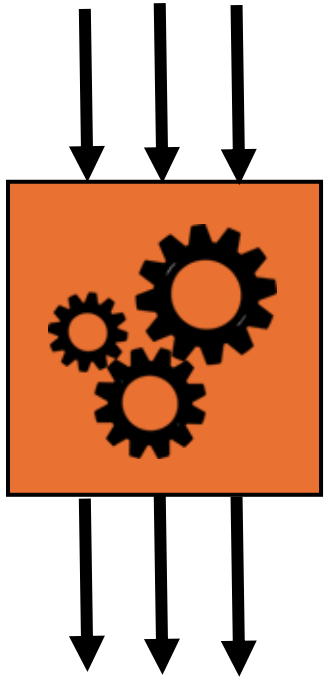
Authors: Michael R. Clarkson, Fred B. Schneider | Authors Info & Claims

Journal of Computer Security, Volume 18, Issue 6 • Pages 1157 - 1210

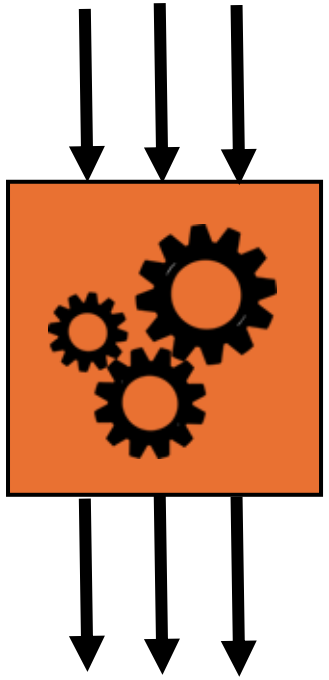
Published: 21 September 2010 Publication History

Noninterference

Noninterference

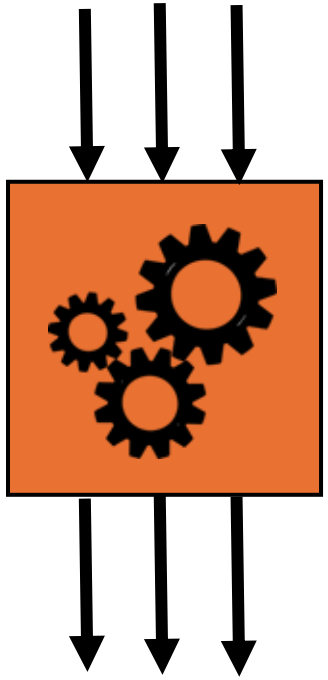


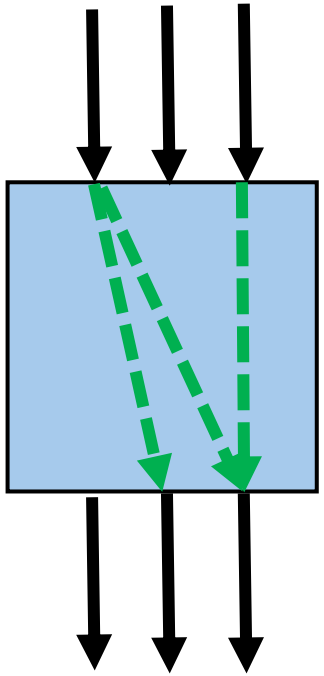
Noninterference





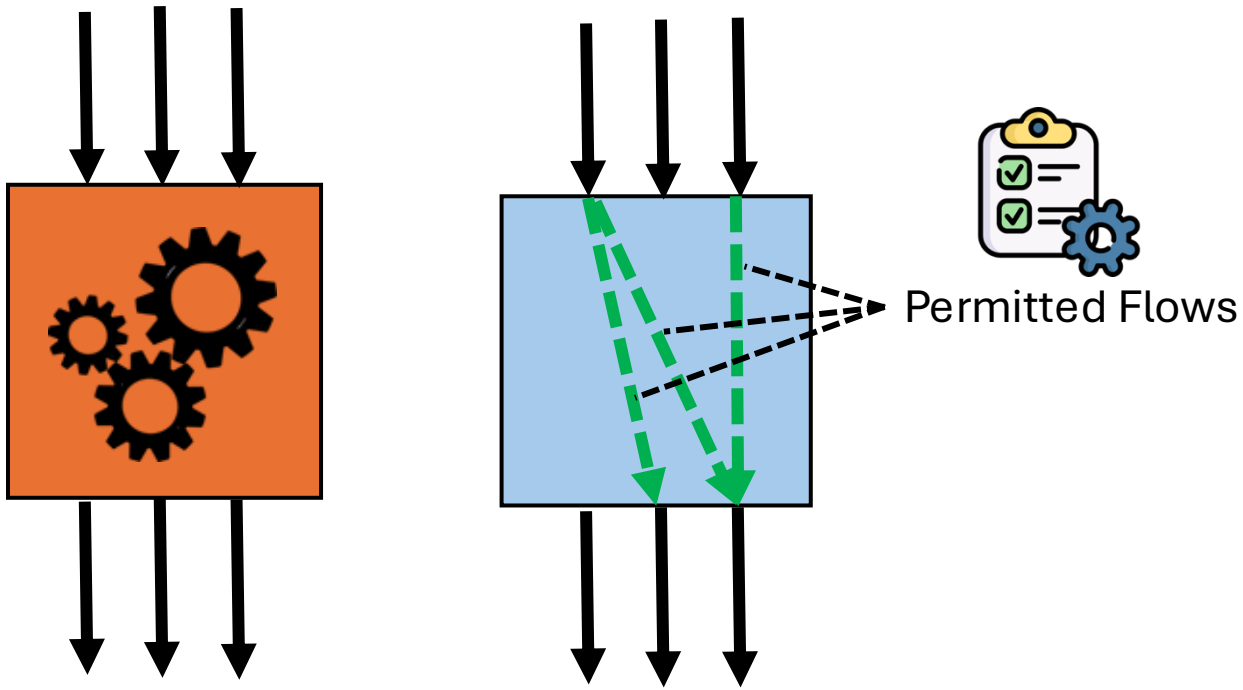
Noninterference



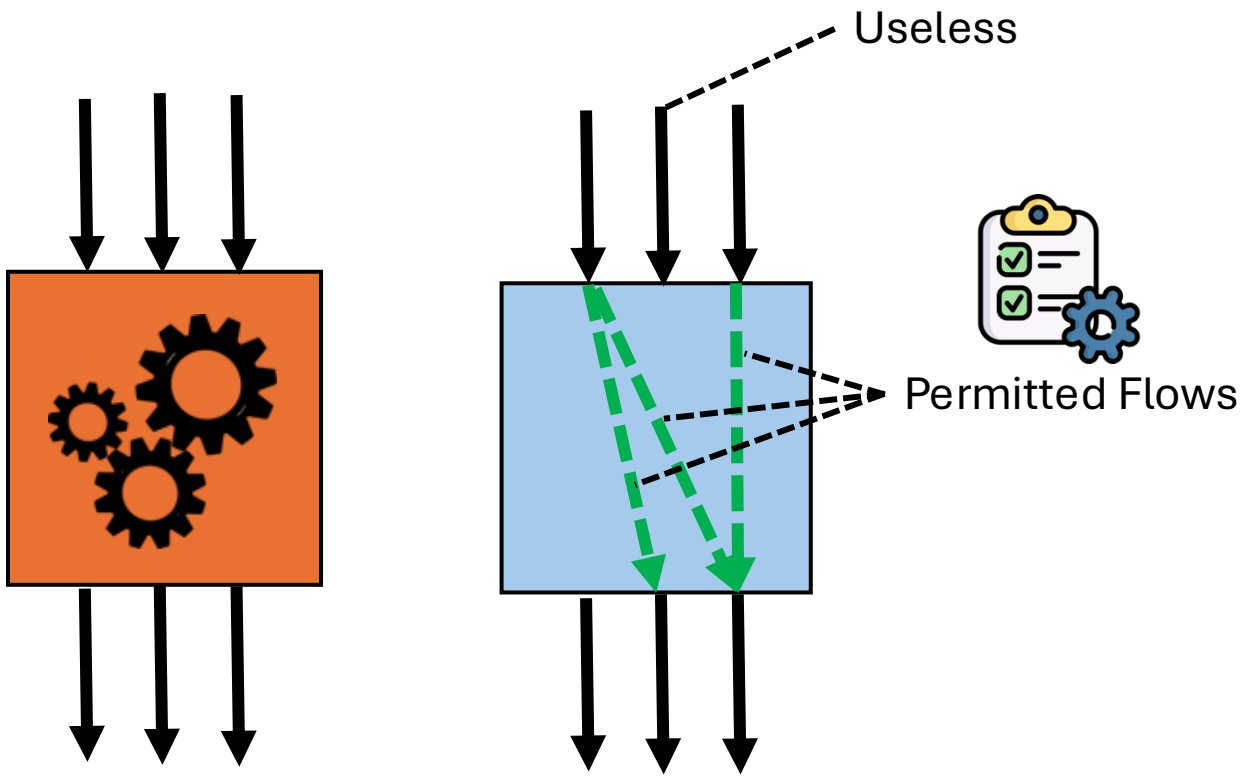




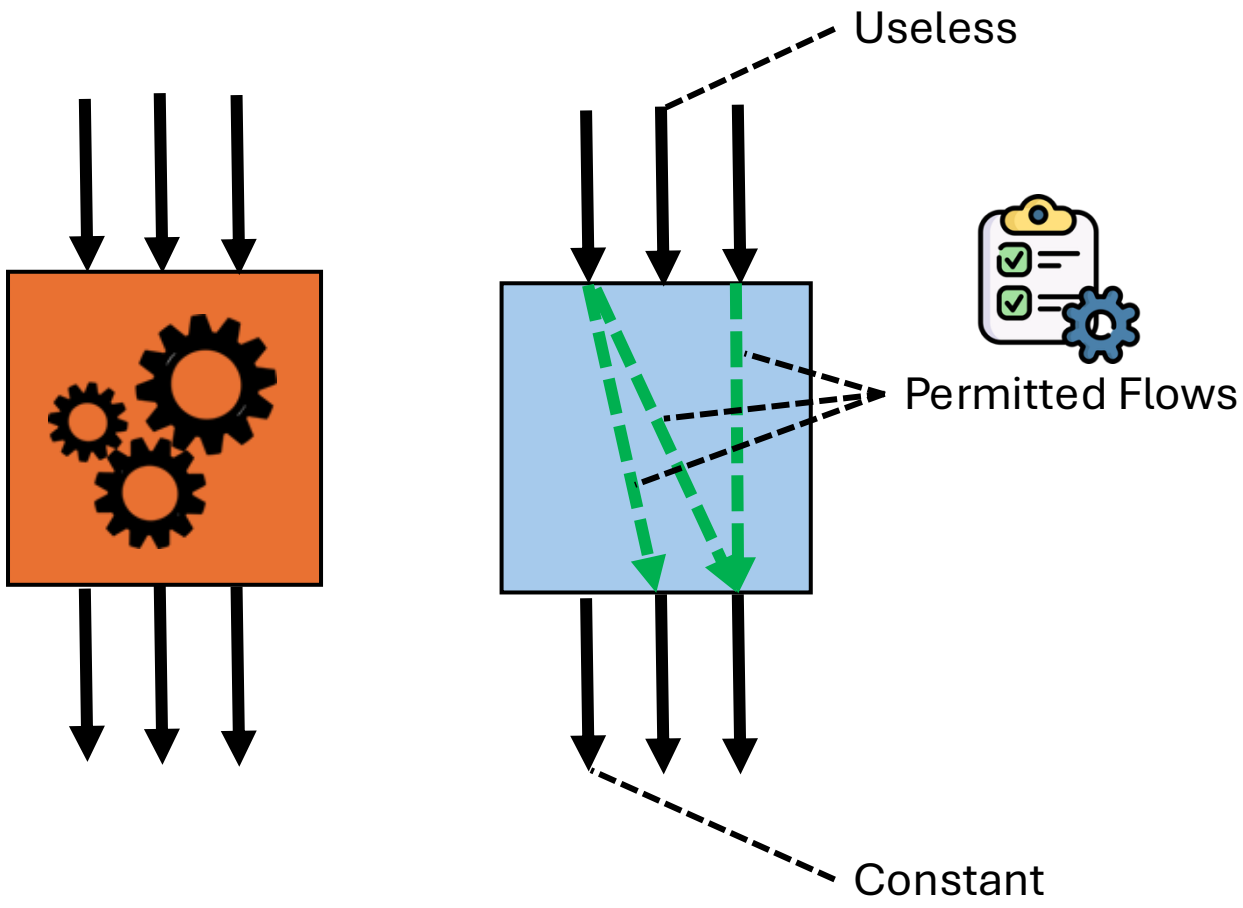
Noninterference



Noninterference

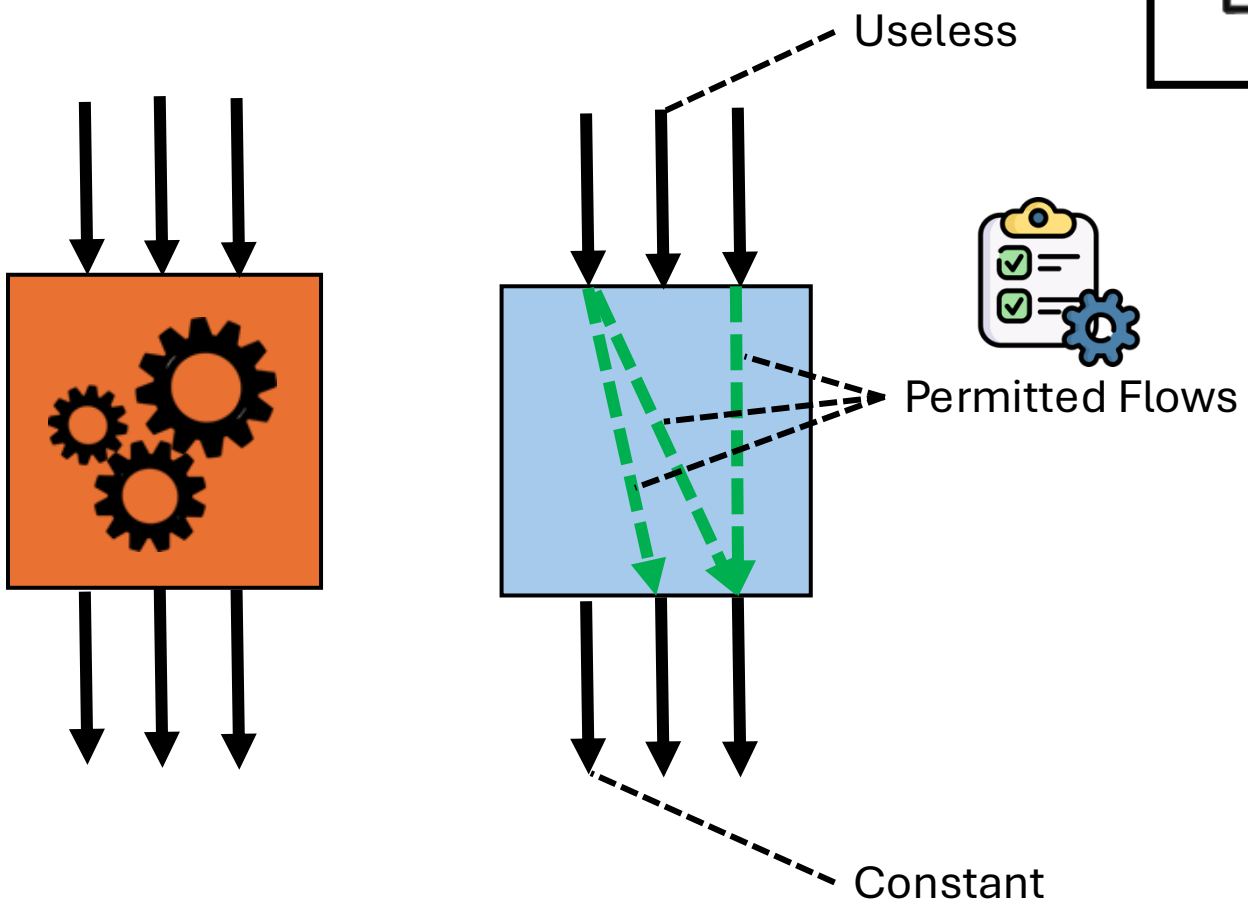


Noninterference



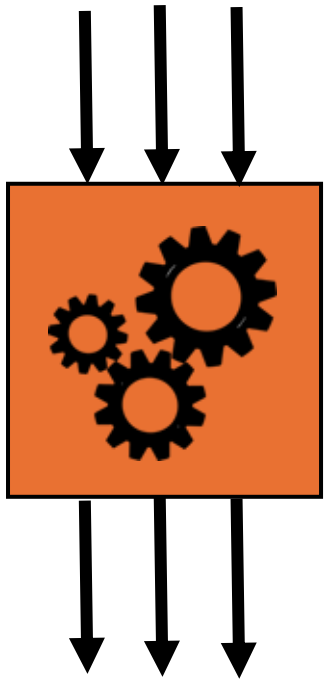
Noninterference

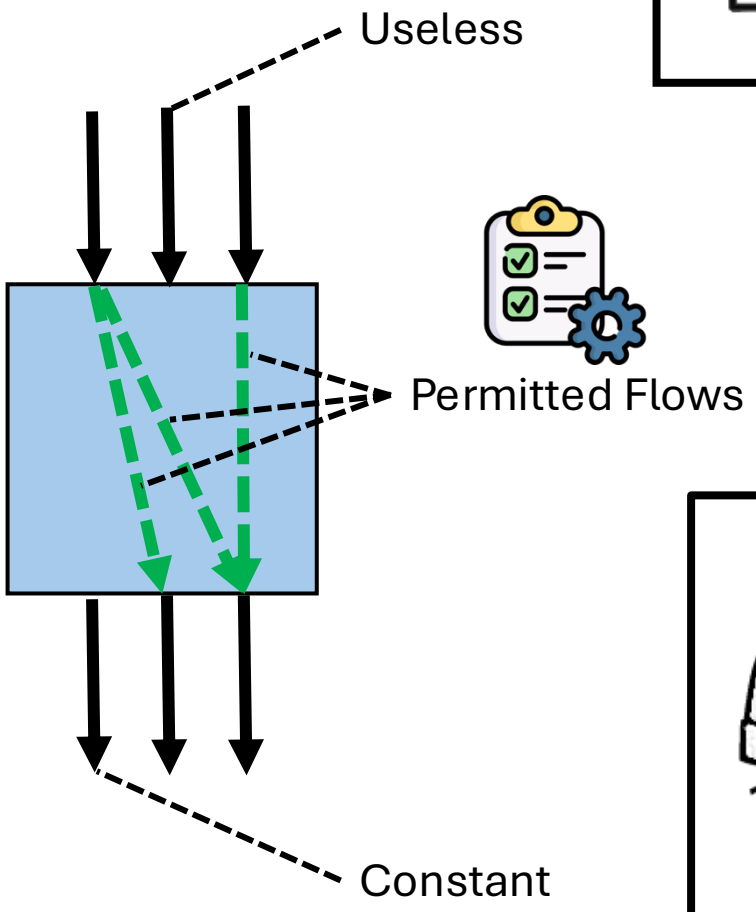


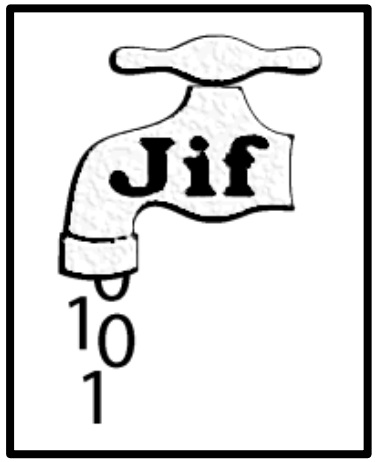


Noninterference





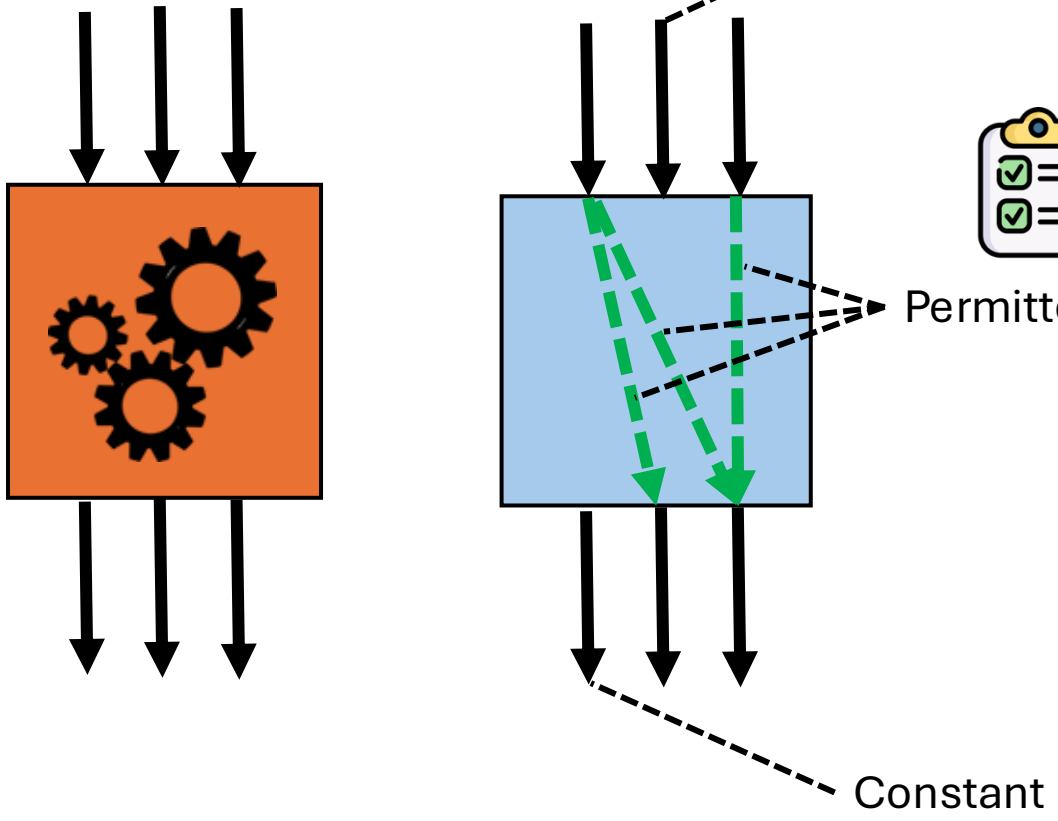


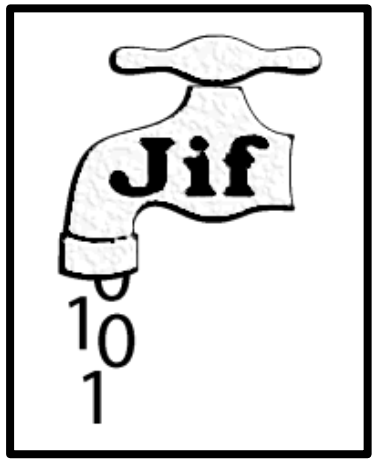


Noninterference





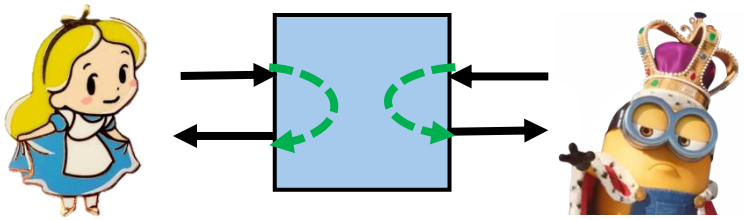


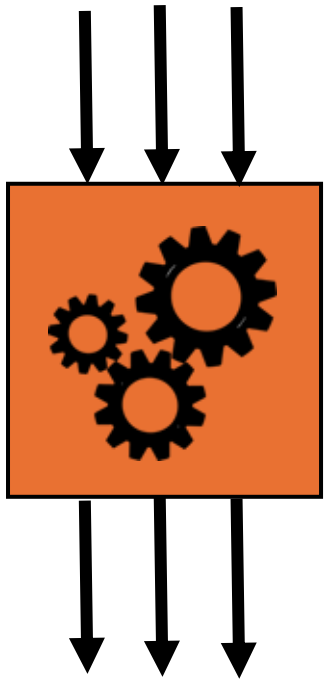


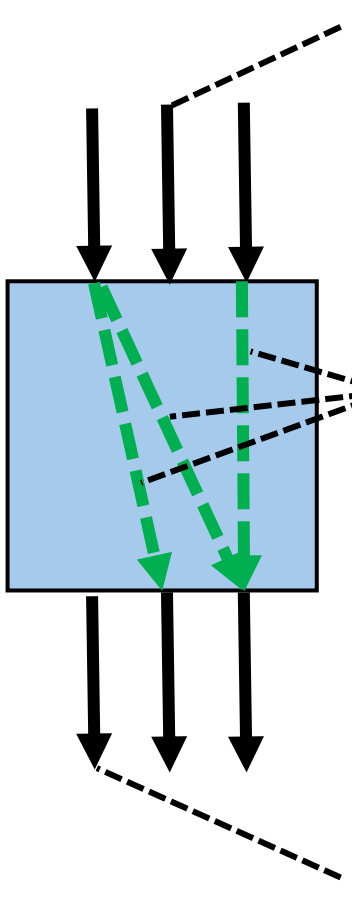
Noninterference









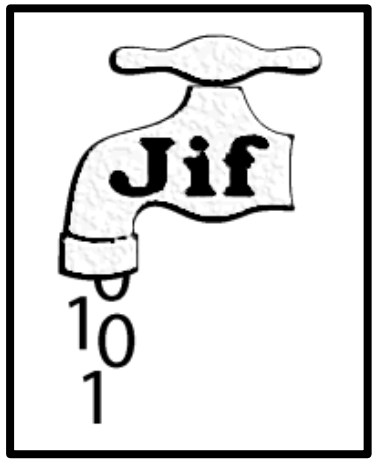


Useless

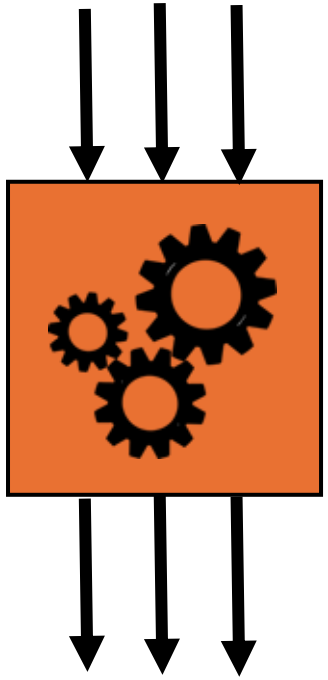


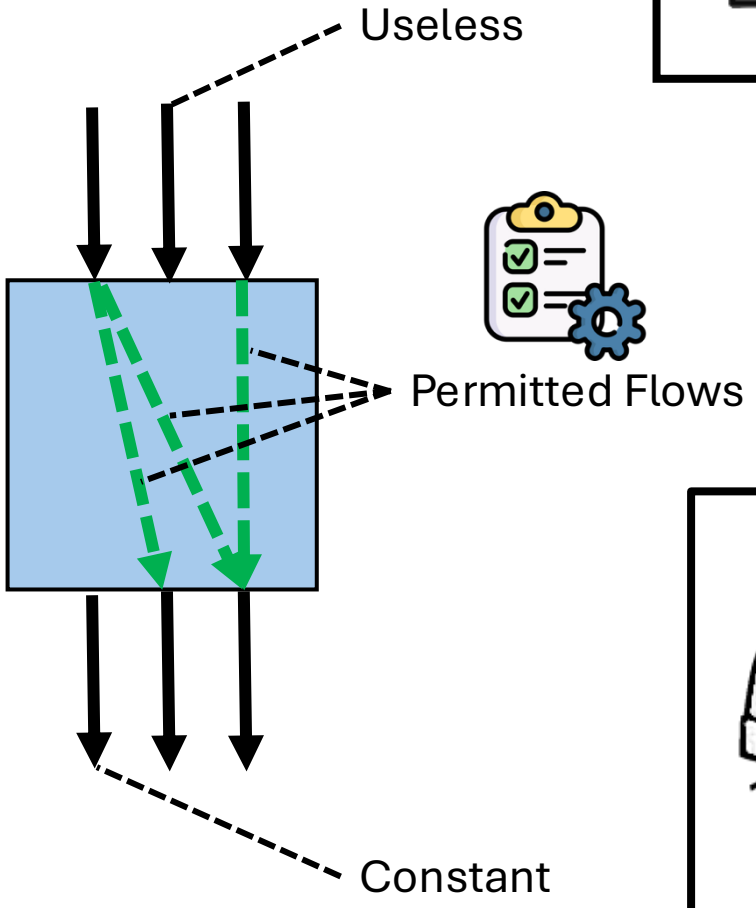
Permitted Flows

Constant



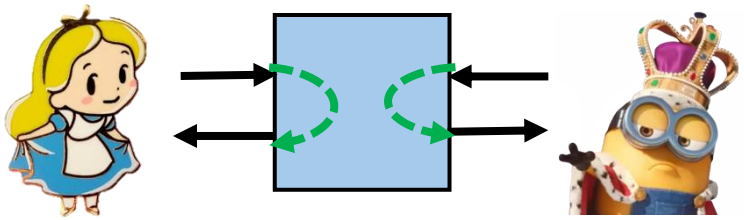
Noninterference

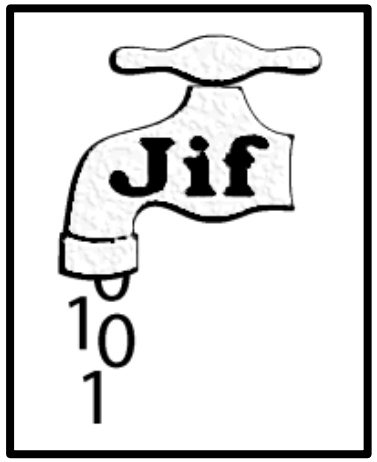


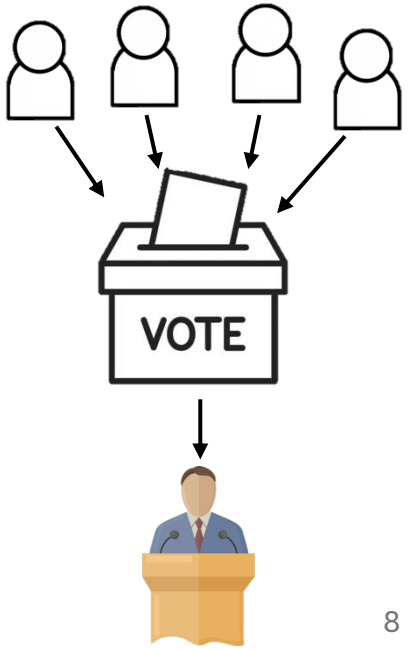




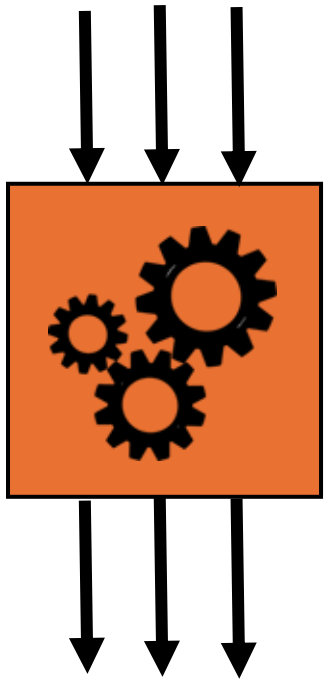


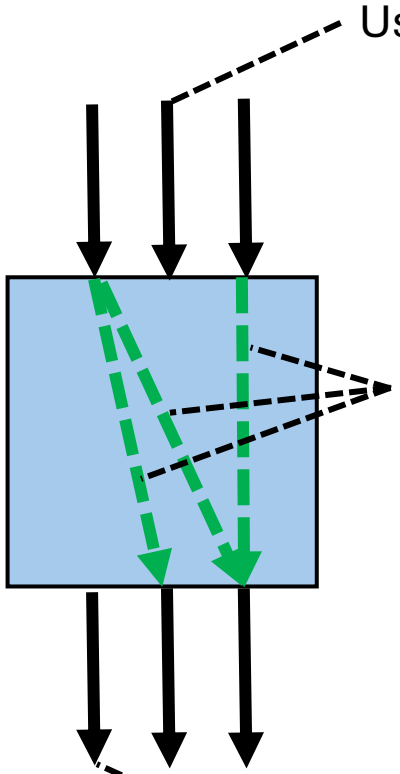






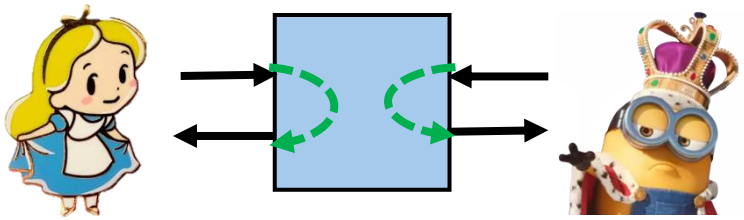
Noninterference

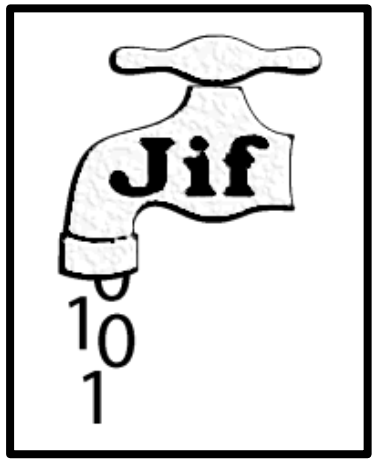


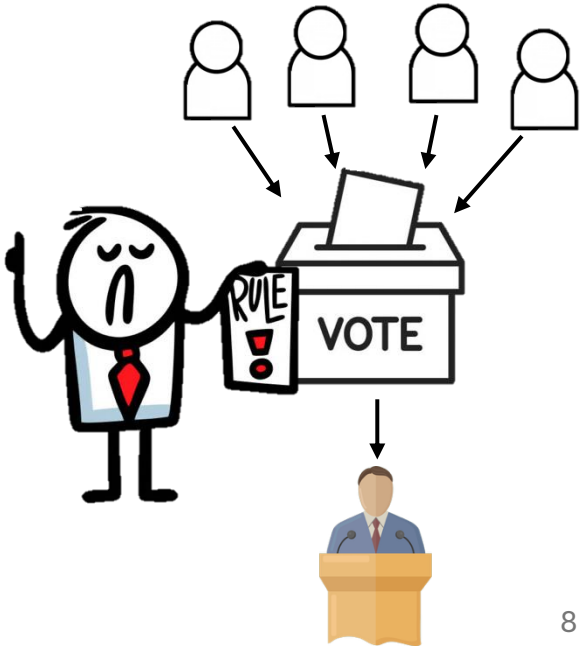




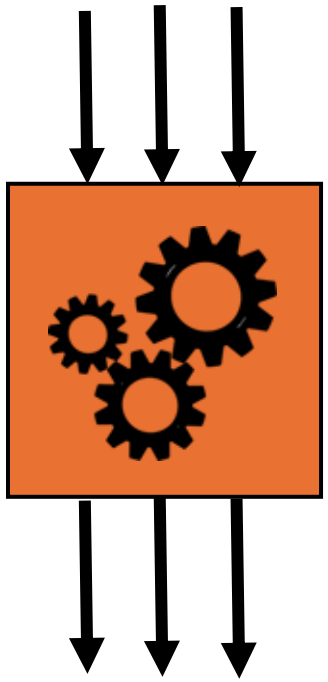


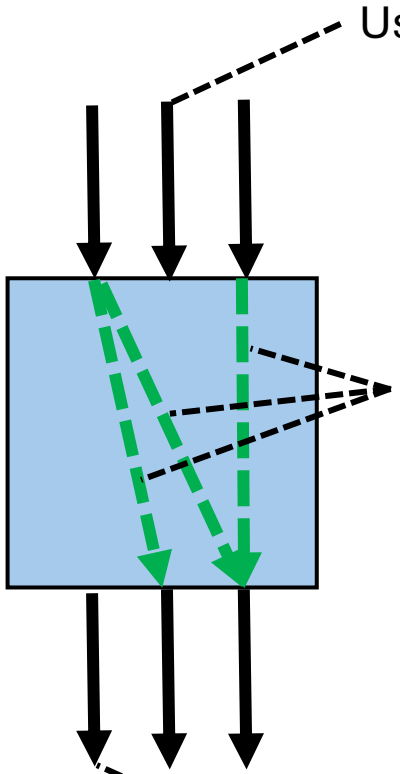






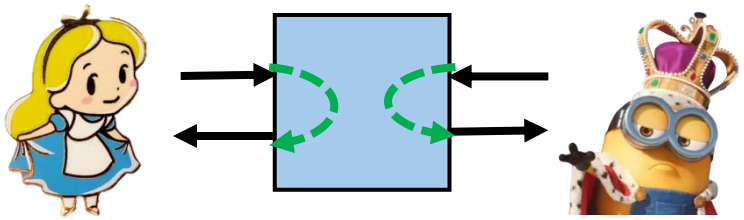
Noninterference

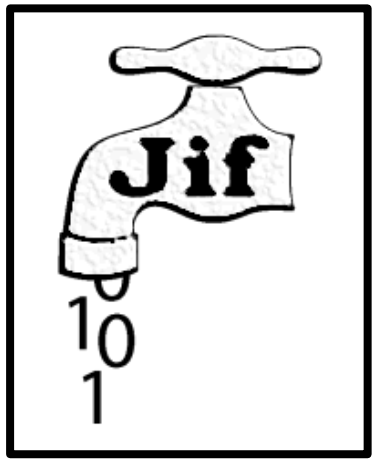


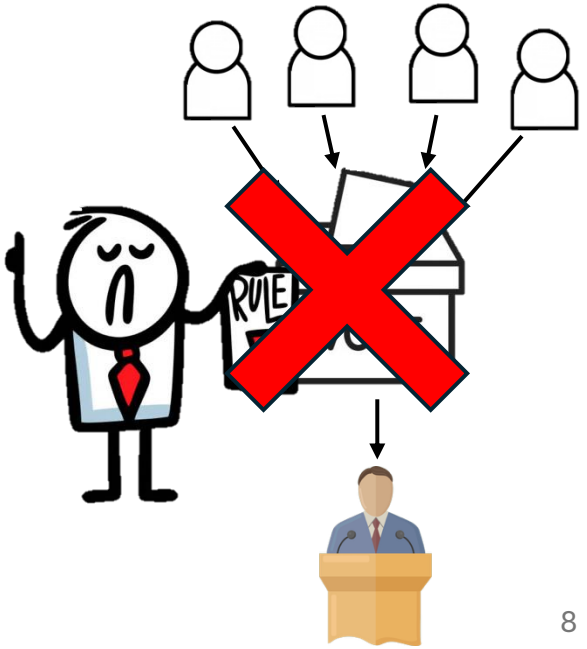












Well-known Specification Techniques



Trace Refinement

The existence of refinement mappings

Authors: Martín Abadi, Leslie Lamport | Authors Info & Claims

Theoretical Computer Science, Volume 82, Issue 2 • Pages 253 - 284 • https://doi.org/10.1016/0304-3975(91)90224-P

Noninterference

A lattice model of secure information flow

Author: Dorothy E. Denning | Authors Info & Claims

Communications of the ACM, Volume 19, Issue 5 • Pages 236 - 243 • https://doi.org/10.1145/360051.360056

Check for updates

Hyperproperties

Hyperproperties

Authors: Michael R. Clarkson, Fred B. Schneider | Authors Info & Claims

Journal of Computer Security, Volume 18, Issue 6 • Pages 1157 - 1210

Published: 21 September 2010 Publication History

Well-known Specification Techniques



Trace Refinement

Noninterference

Hyperproperties

The existence of refinement mappings

Authors: Martín Abadi, Leslie Lamport | Authors Info & Claims

Theoretical Computer Science, Volume 82, Issue 2 • Pages 253 - 284 • https://doi.org/10.1016/0304-3975(91)90224-P

A lattice model of secure information flow

Author: Dorothy E. Denning | Authors Info & Claims

Communications of the ACM, Volume 19, Issue 5 • Pages 236 - 243 • https://doi.org/10.1145/360051.360056

Check for updates

Hyperproperties

Authors: Michael R. Clarkson, Fred B. Schneider | Authors Info & Claims

Journal of Computer Security, Volume 18, Issue 6 • Pages 1157 - 1210

Published: 21 September 2010 Publication History

Hyperproperties

Hyperproperties



Hyperproperties

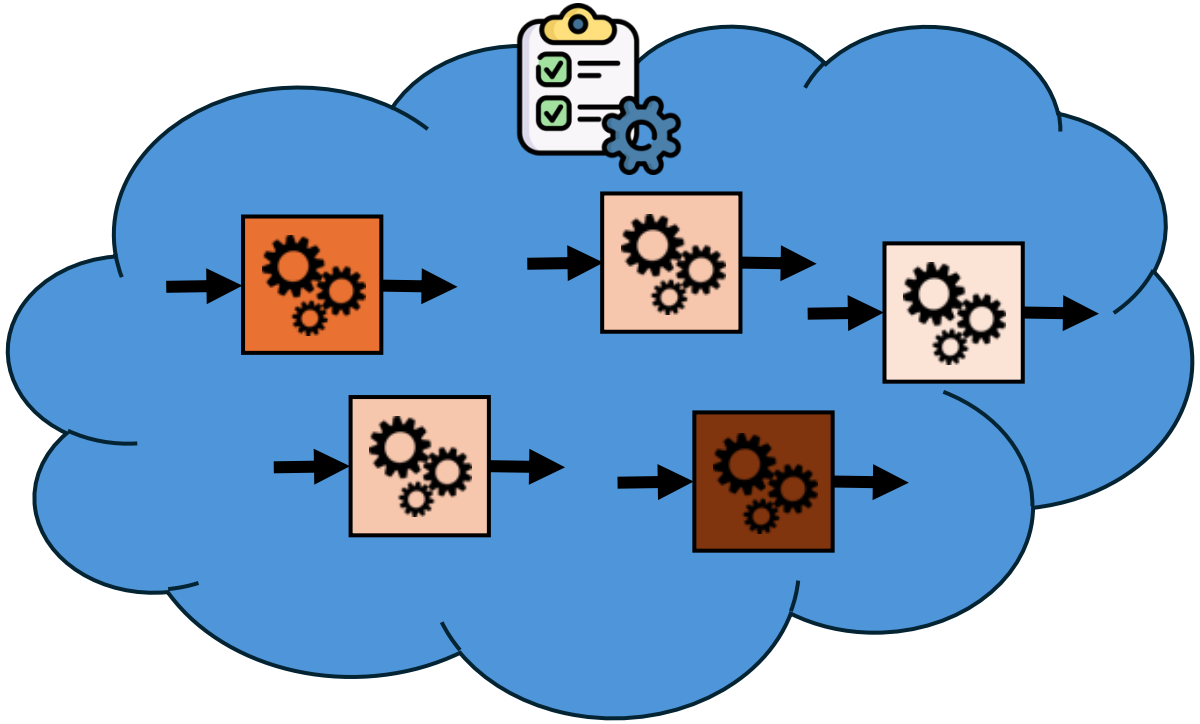




Hyperproperties





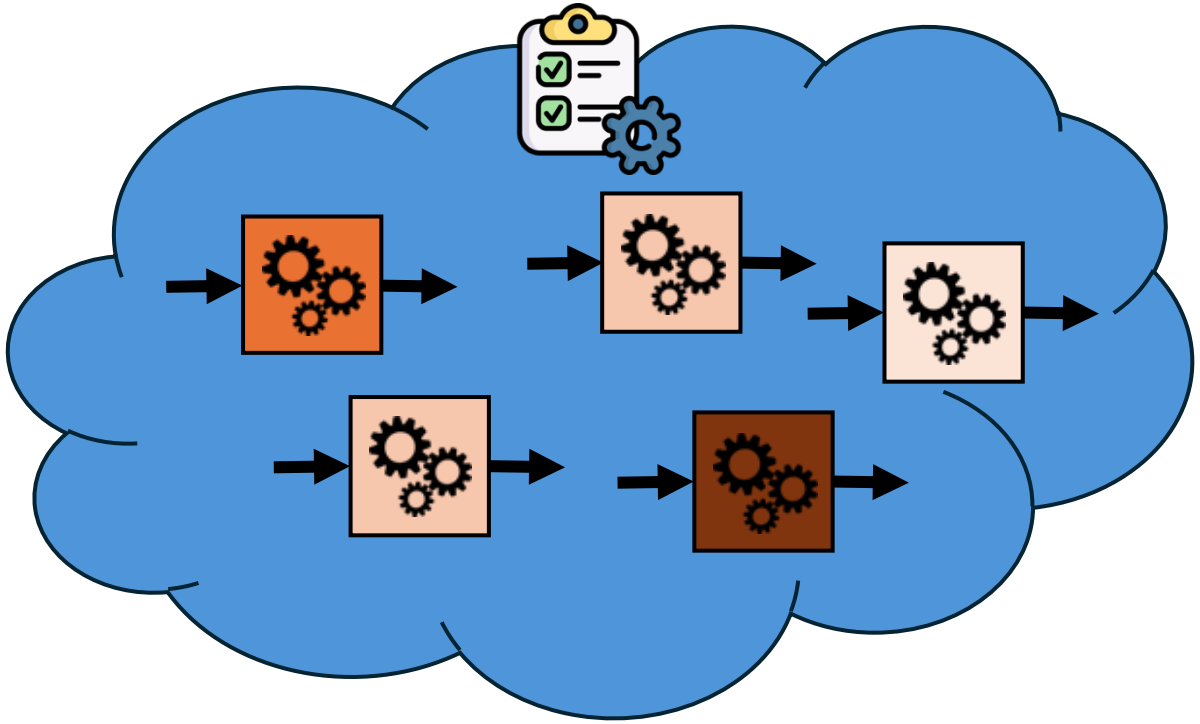


Hyperproperties

“Any Predicate on Systems”





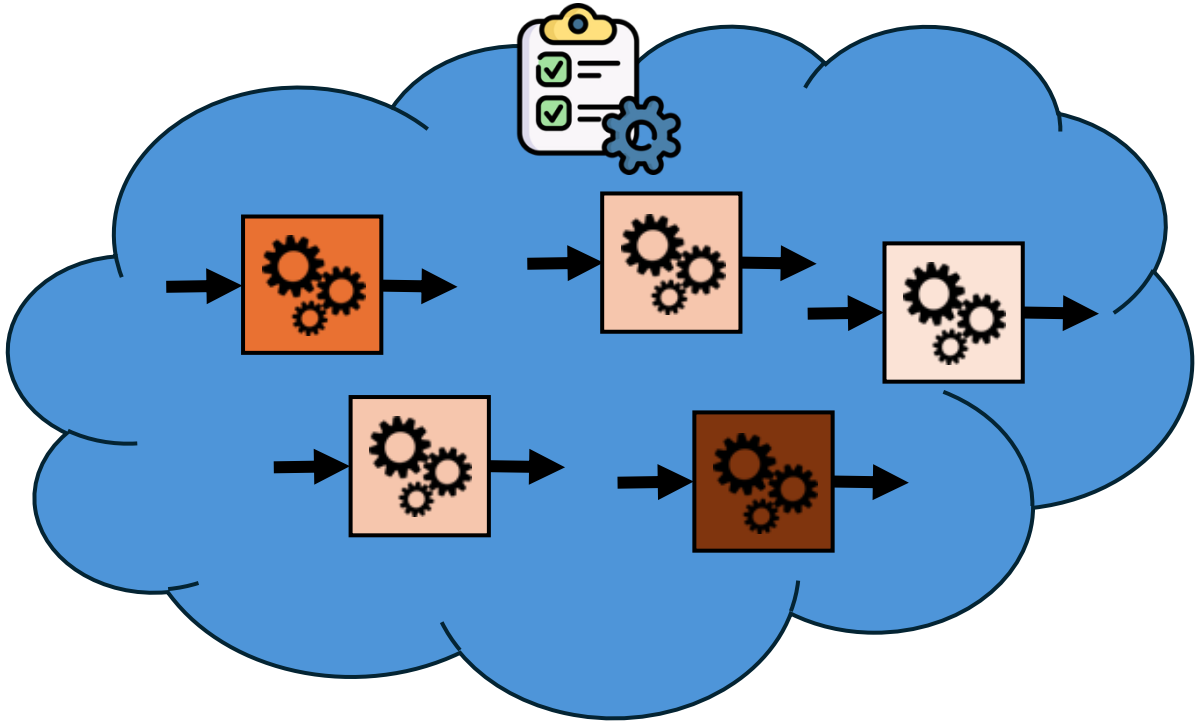


Hyperproperties

“Any Predicate on Systems”







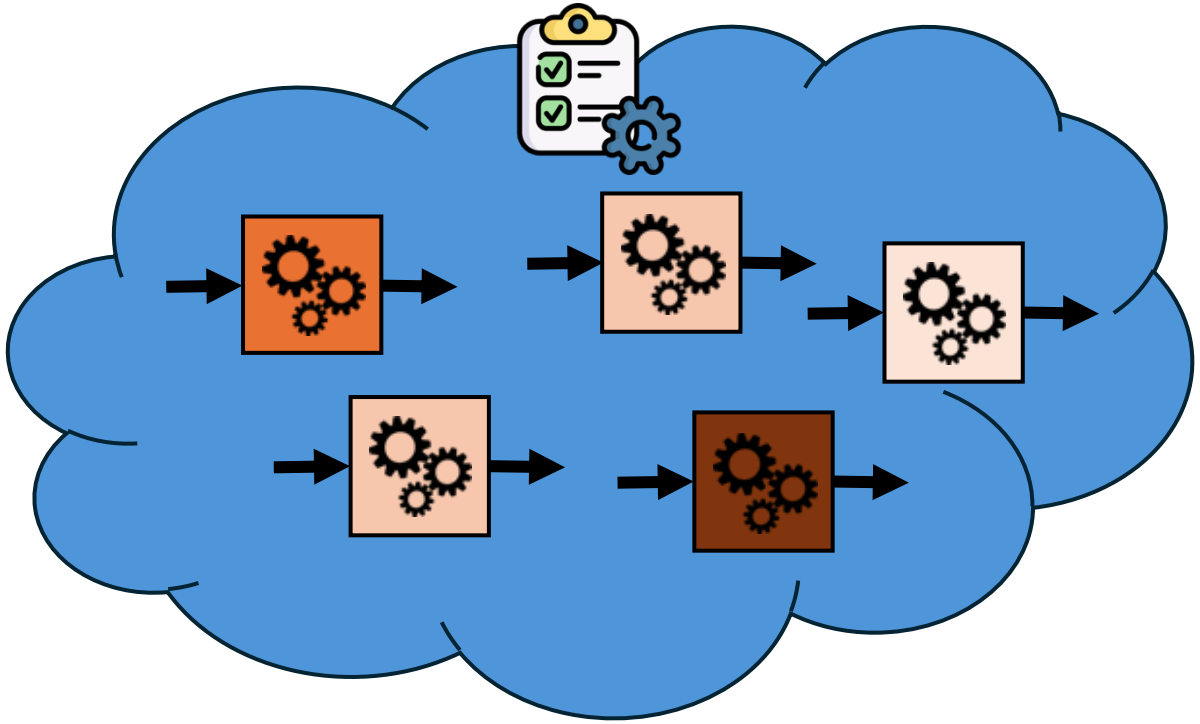
1

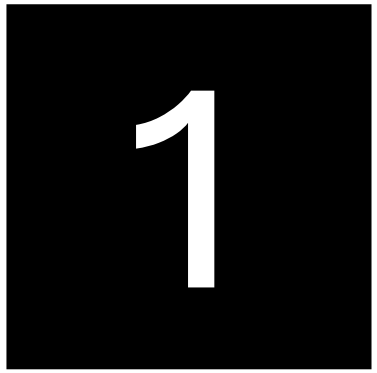
Hyperproperties

“Any Predicate on Systems”











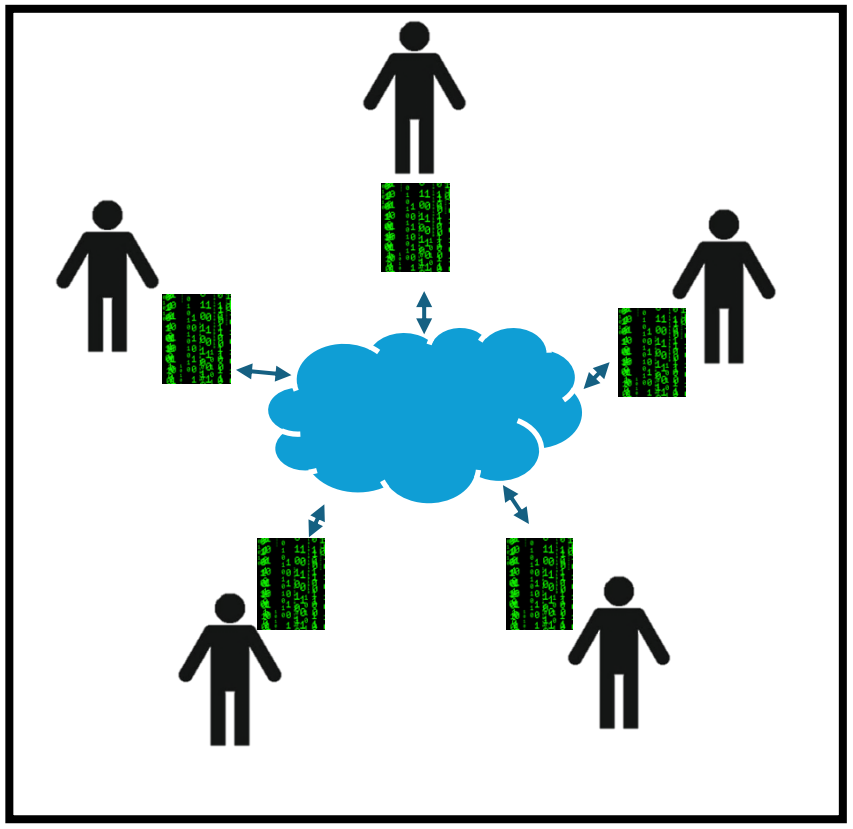
A Cryptographic Idea

A Cryptographic Idea



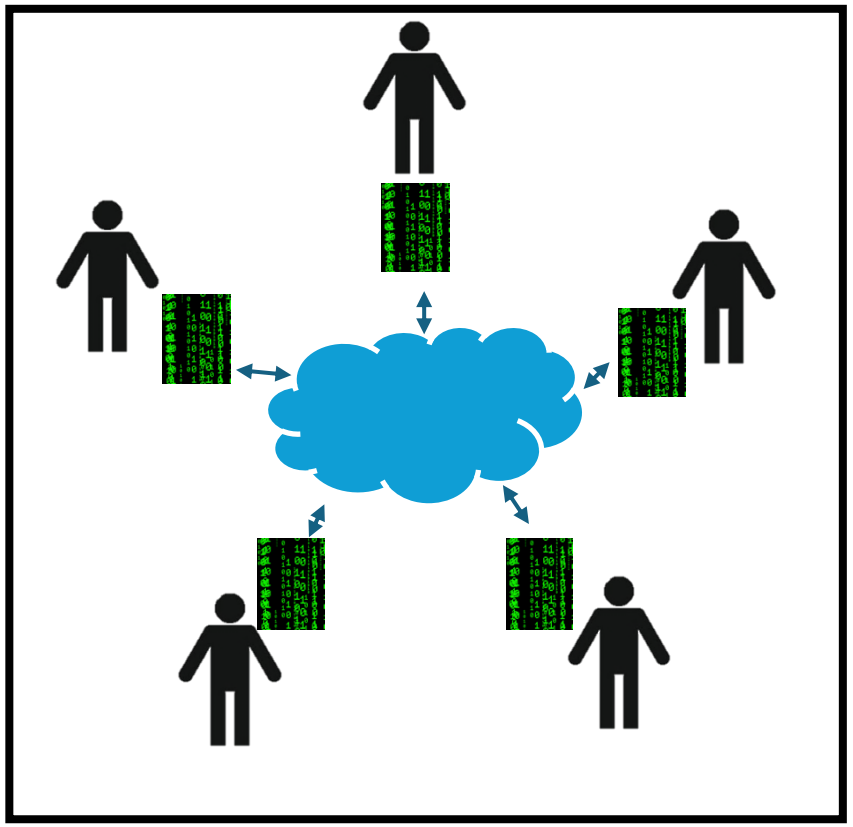
A Cryptographic Idea

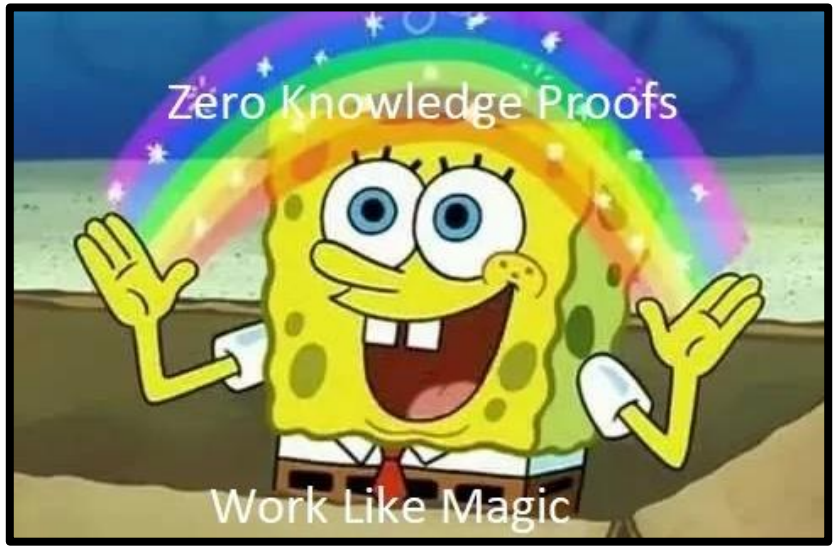




A Cryptographic Idea

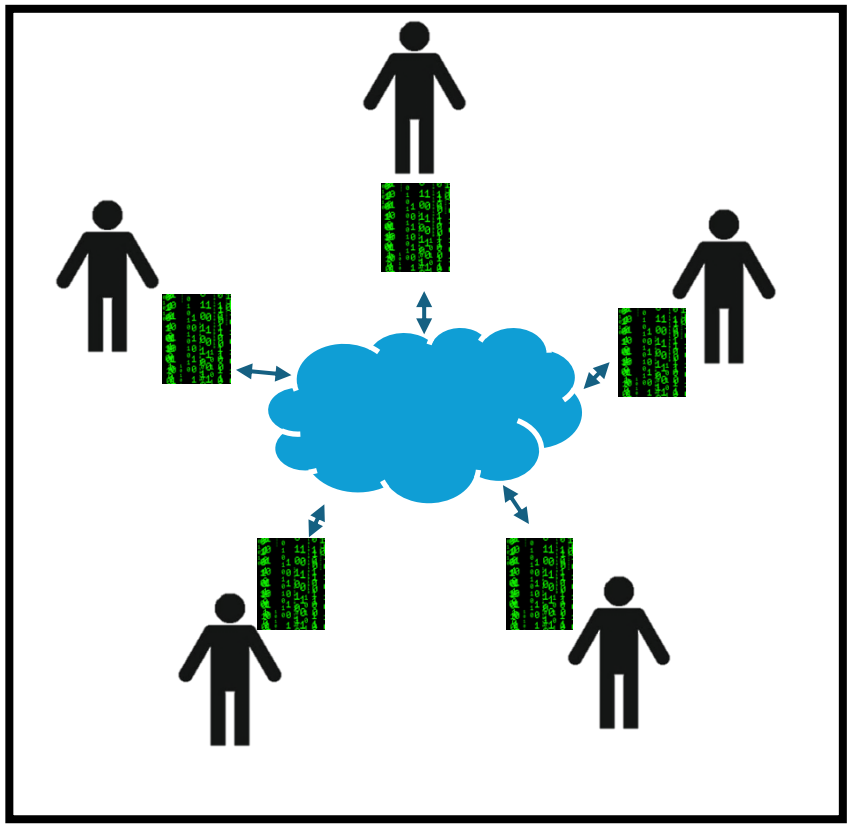


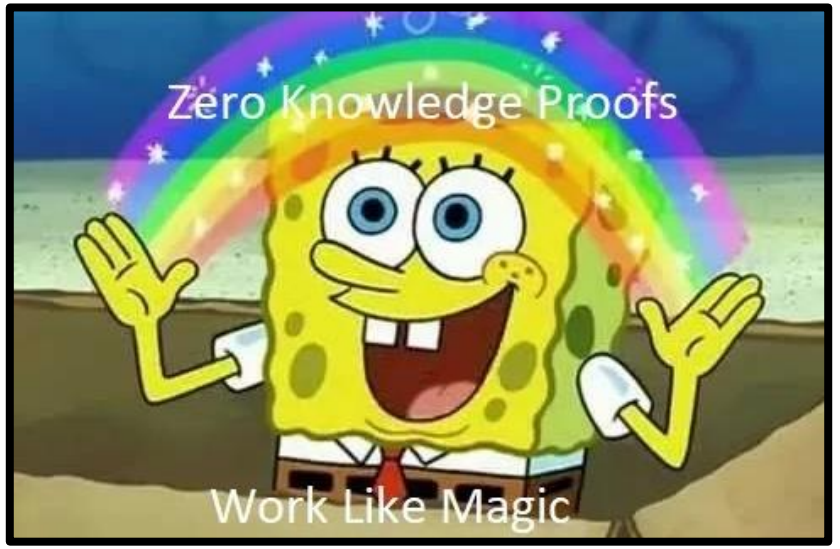




A Cryptographic Idea



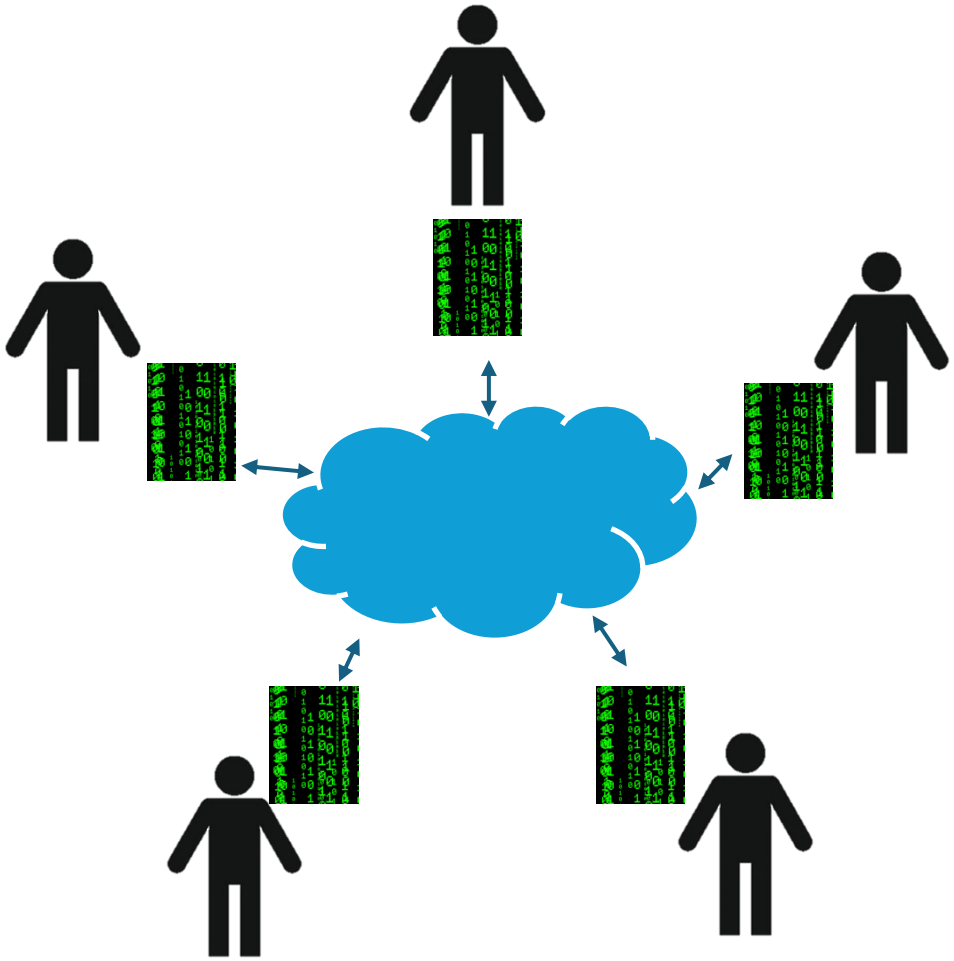




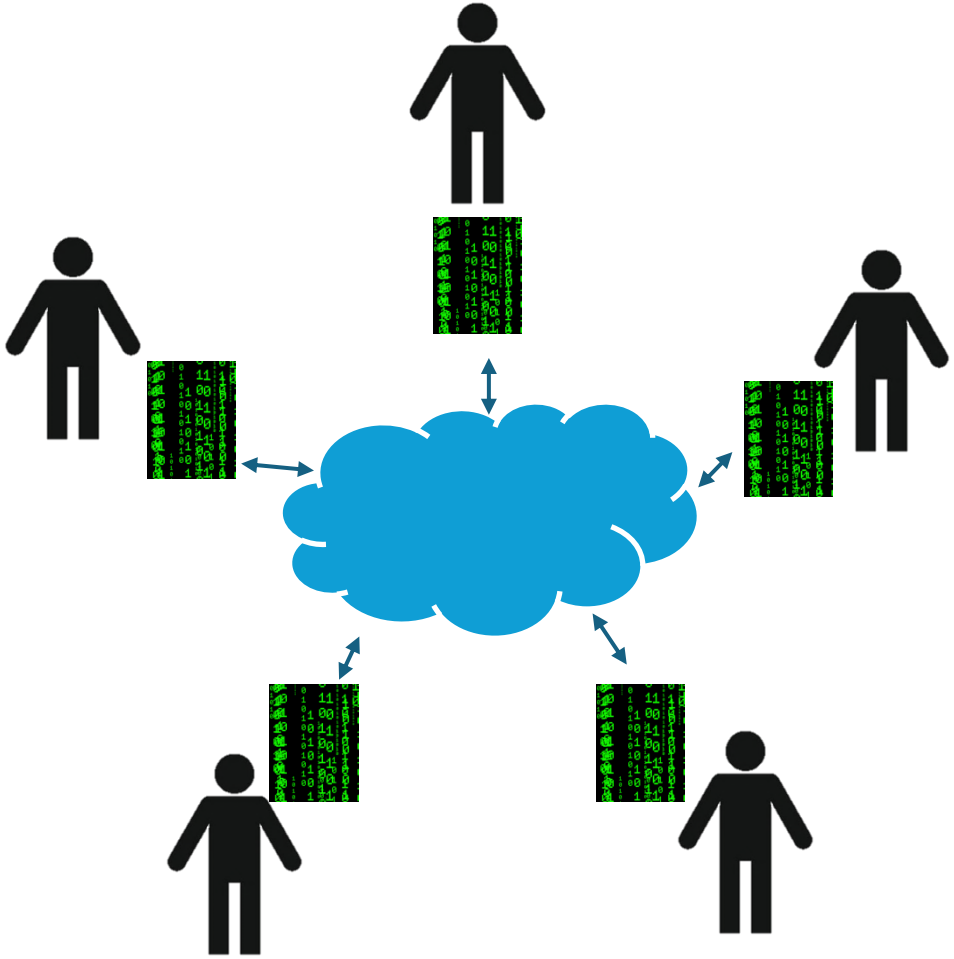
How to specify requirements?

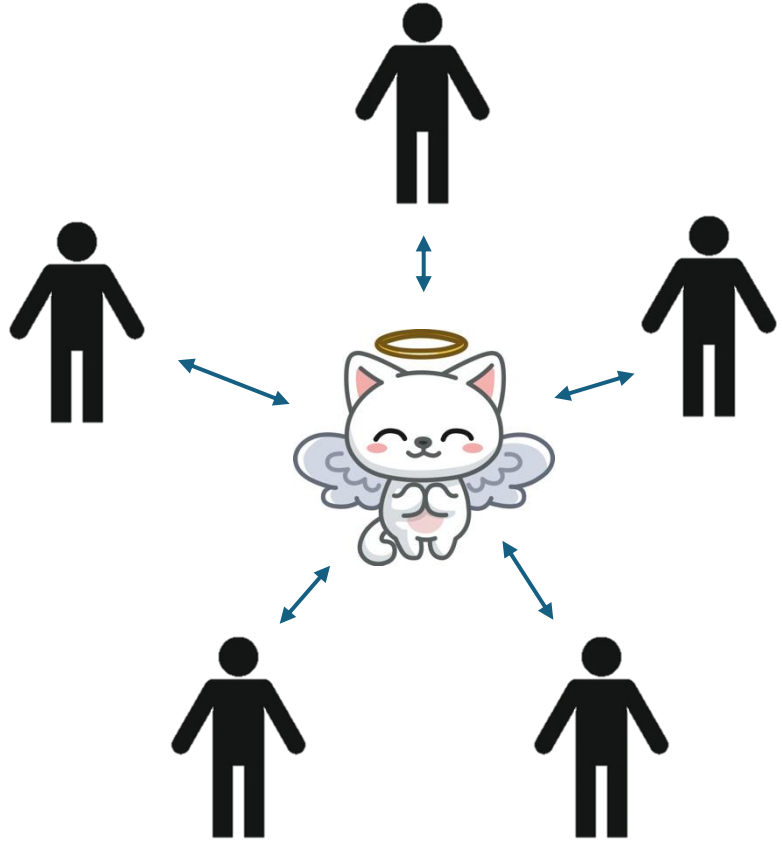
The Real-Ideal Paradigm

The Real-Ideal Paradigm

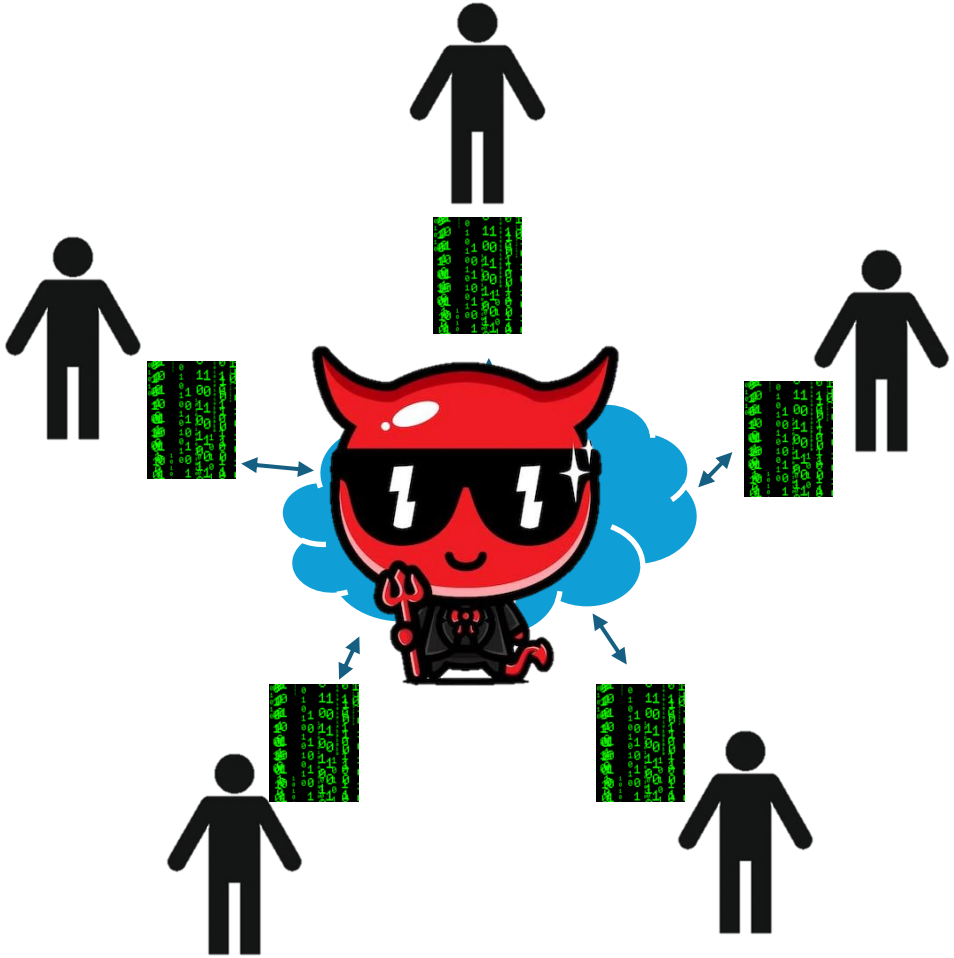


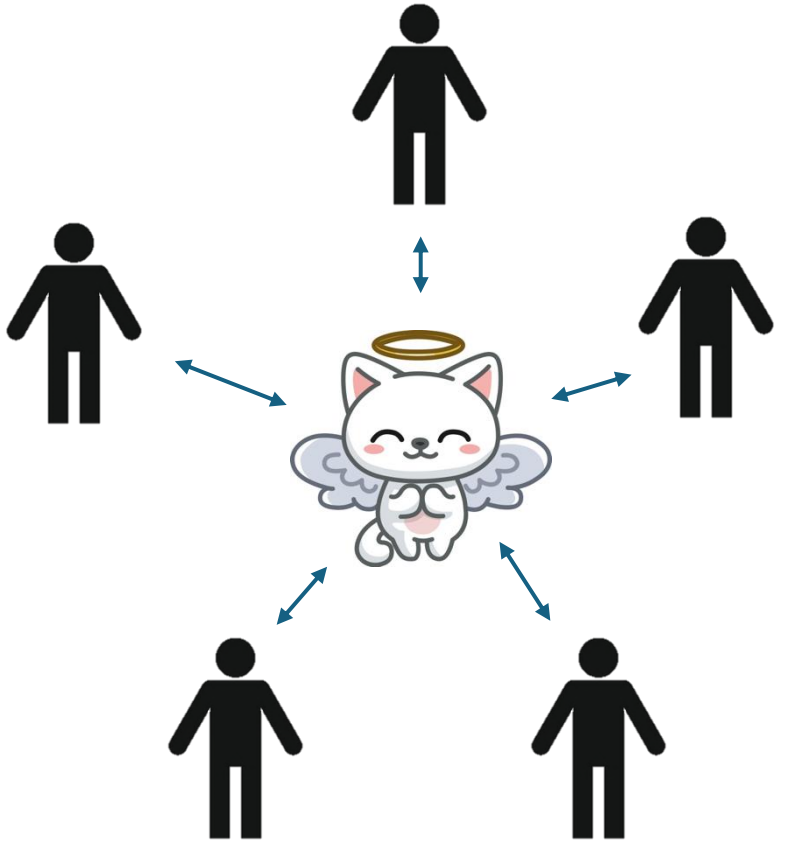
The Real-Ideal Paradigm



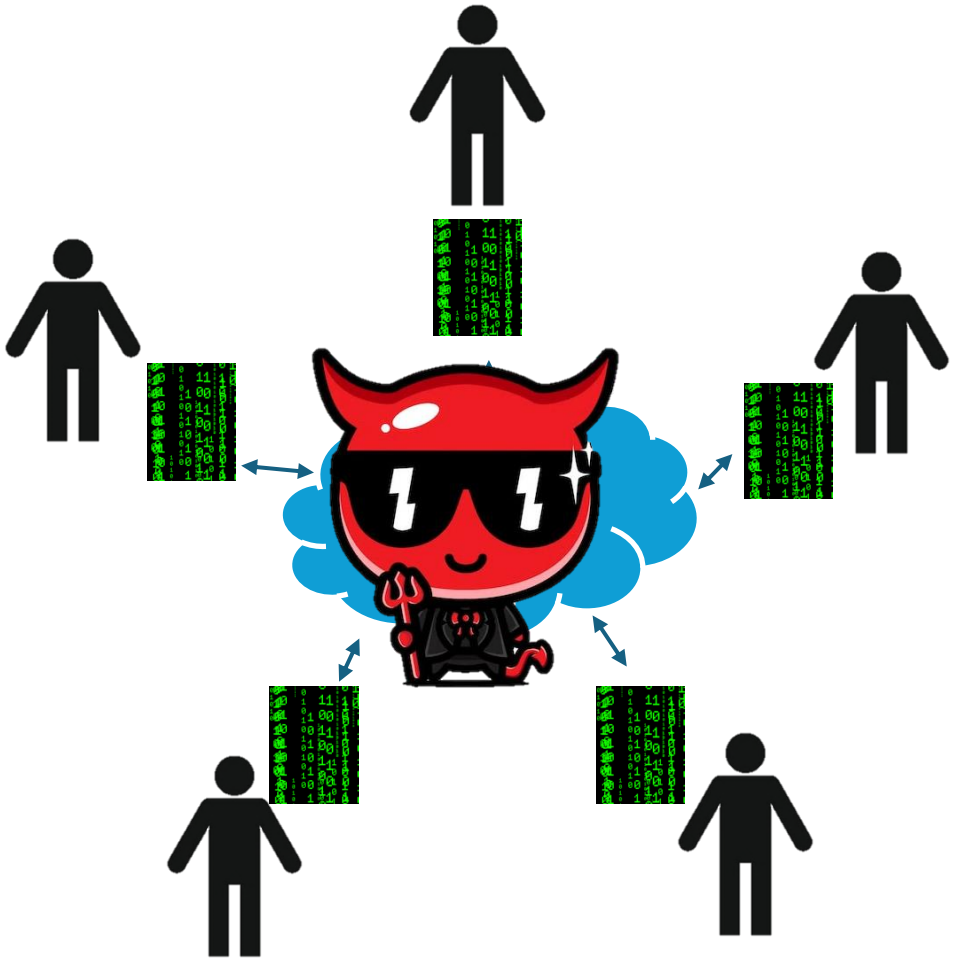


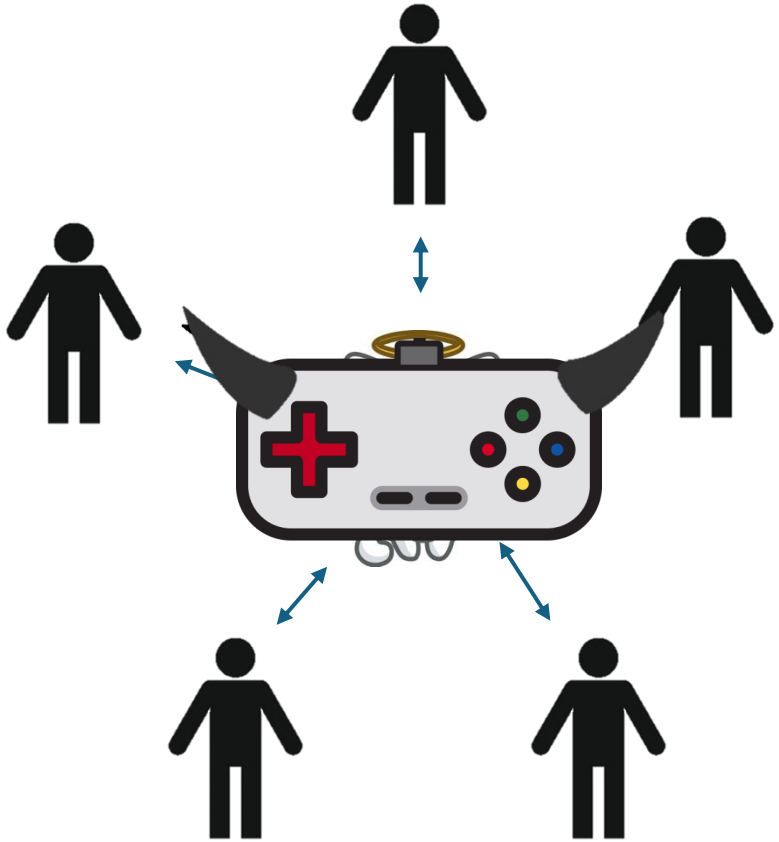
The Real-Ideal Paradigm



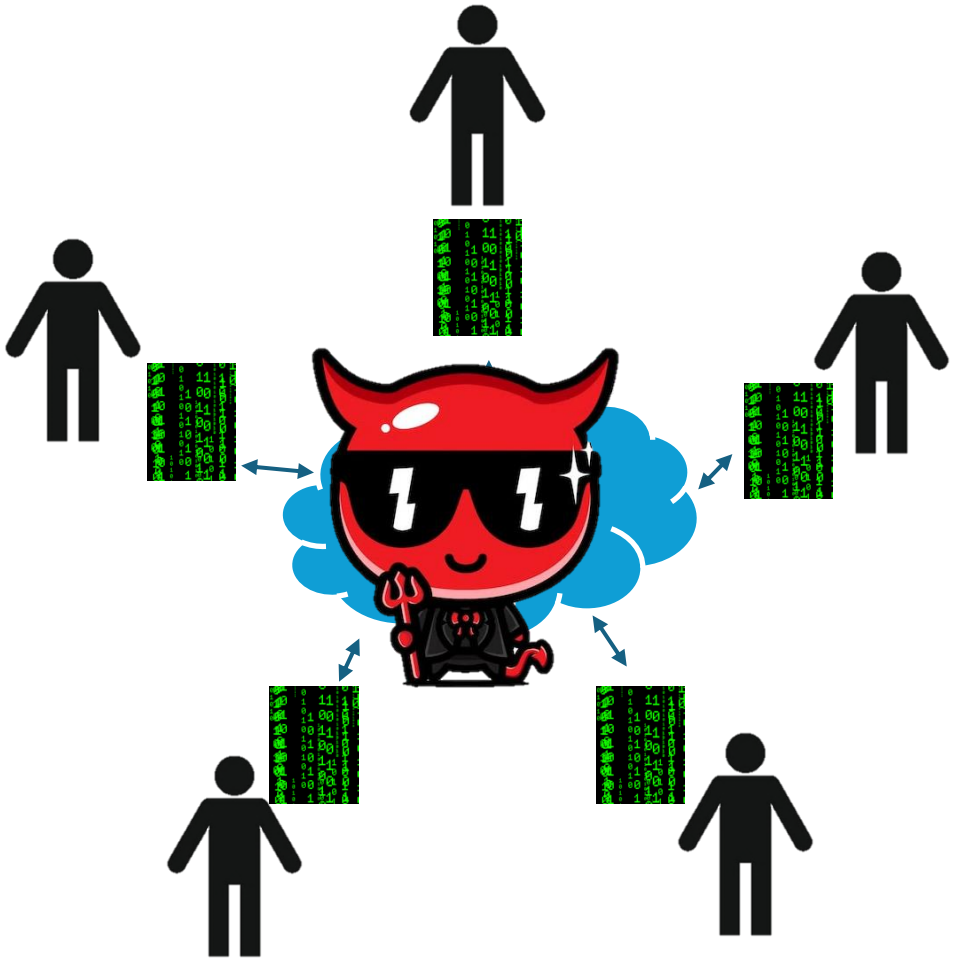


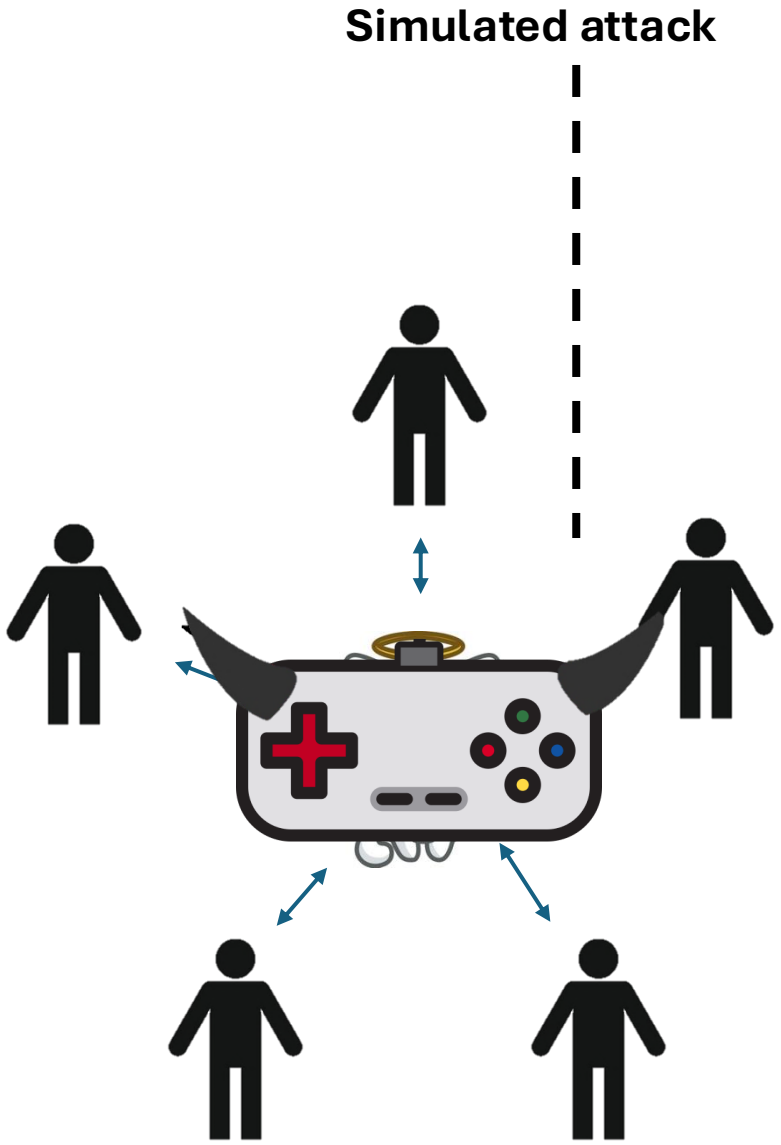
The Real-Ideal Paradigm



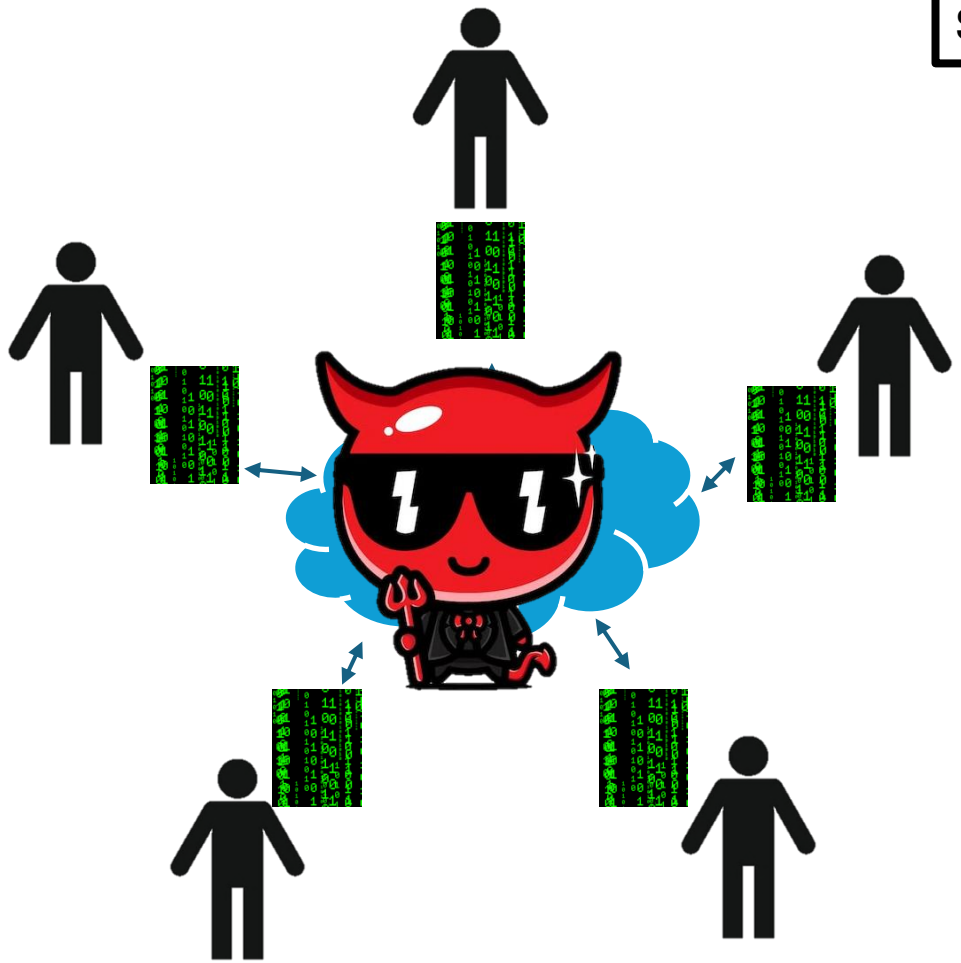


The Real-Ideal Paradigm

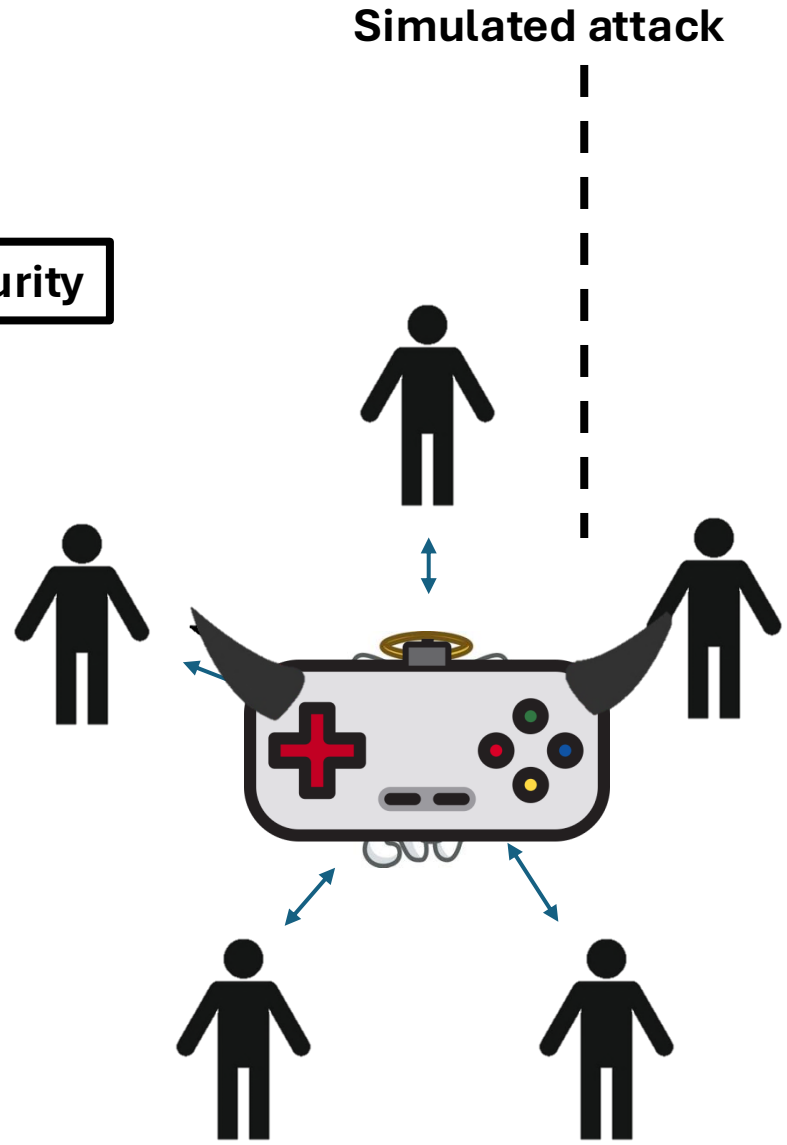




The Real-Ideal Paradigm



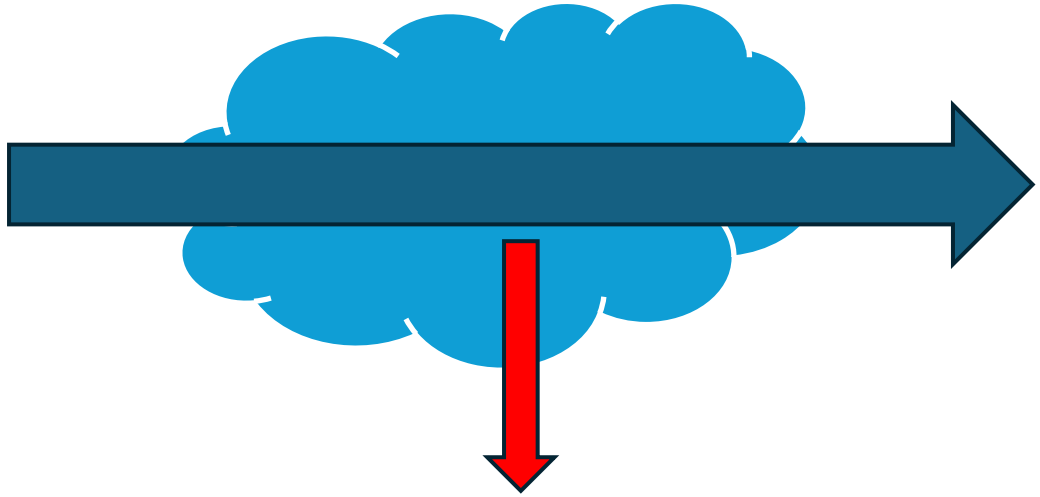
Simulation Based Security



Real System: Secure Channel w/ Encryption

Real System: Secure Channel w/ Encryption



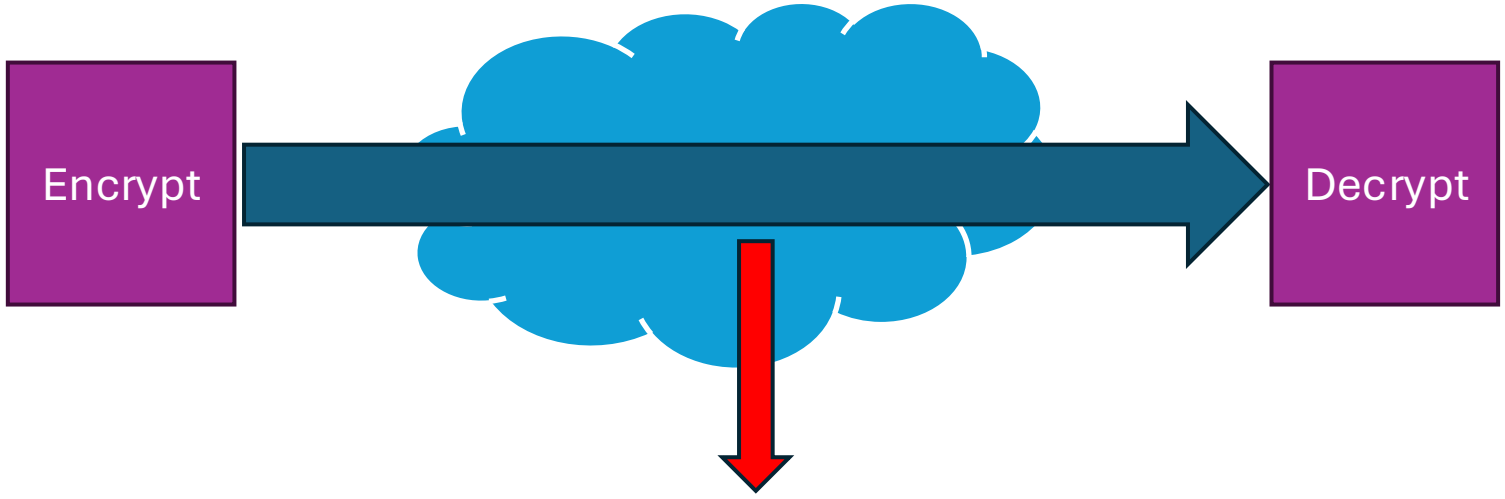






Real System: Secure Channel w/ Encryption

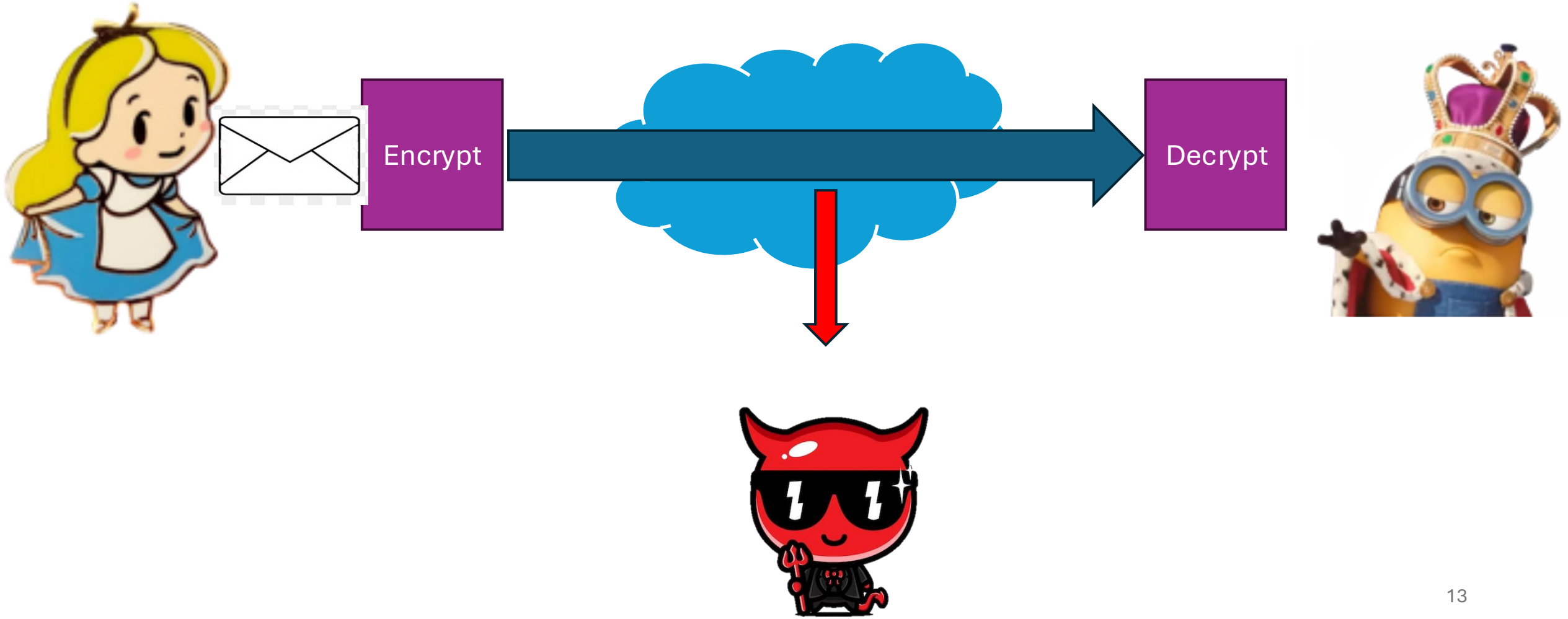






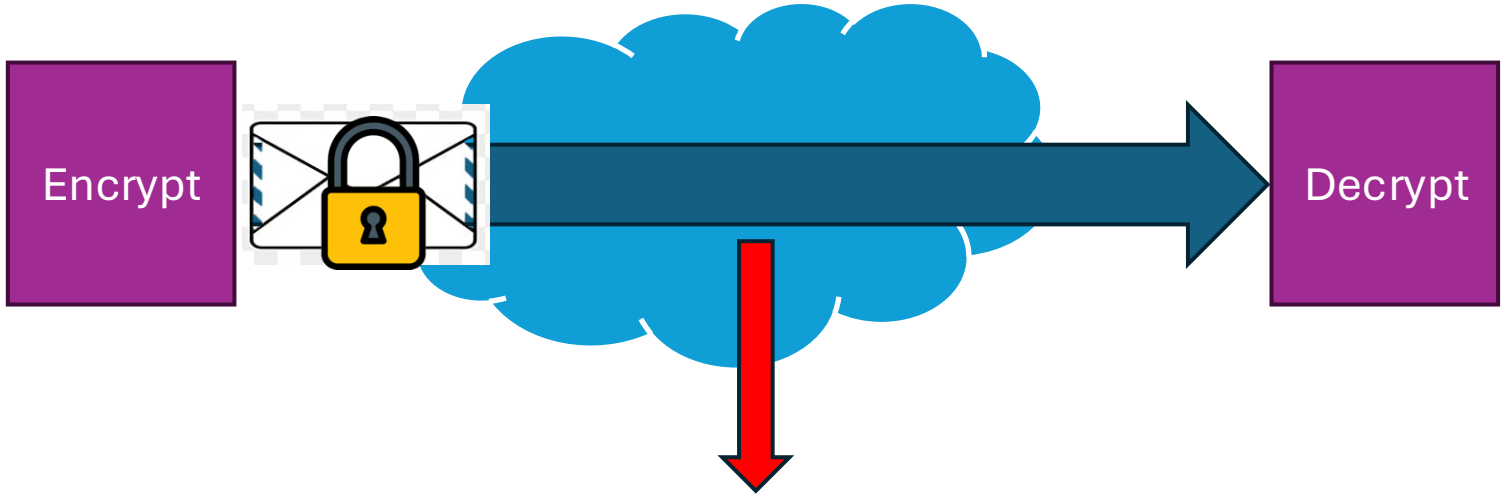


Real System: Secure Channel w/ Encryption



Real System: Secure Channel w/ Encryption



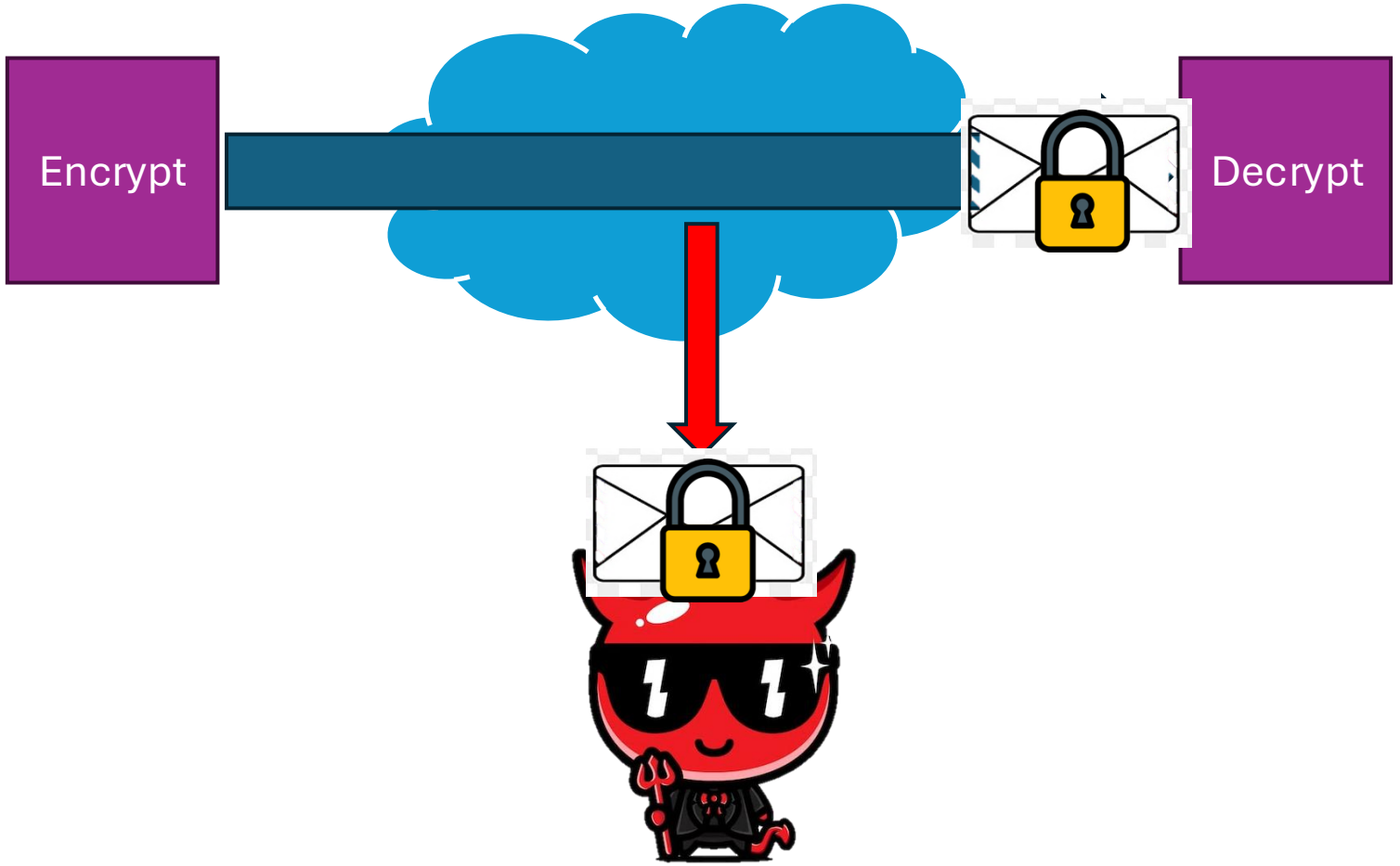






Real System: Secure Channel w/ Encryption

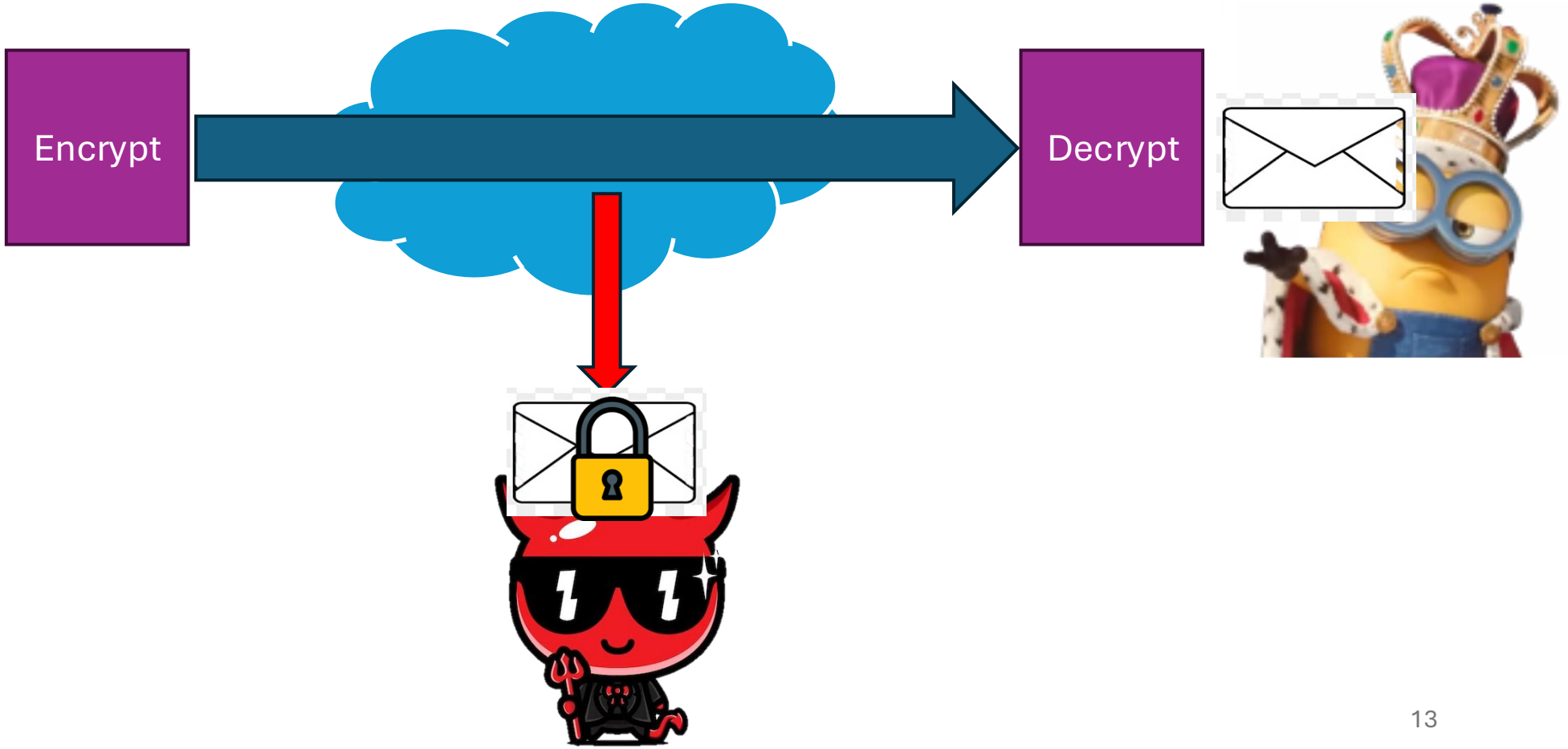






Real System: Secure Channel w/ Encryption





Ideal System: The Specification

Ideal System: The Specification

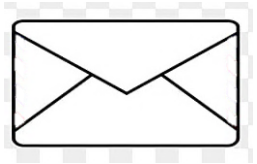






Ideal System: The Specification



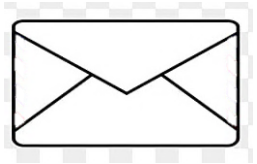






Ideal System: The Specification





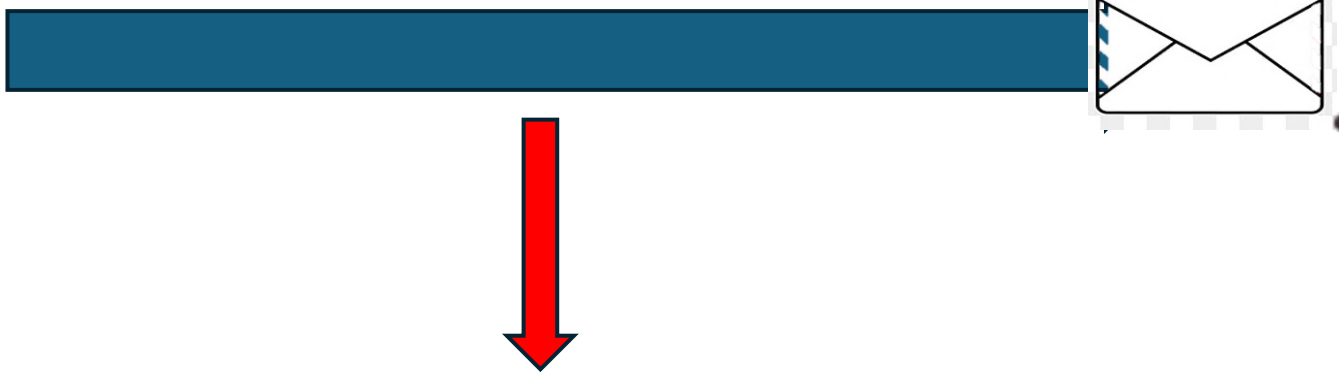




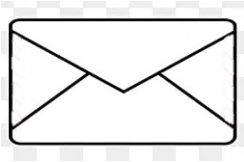


Ideal System: The Specification





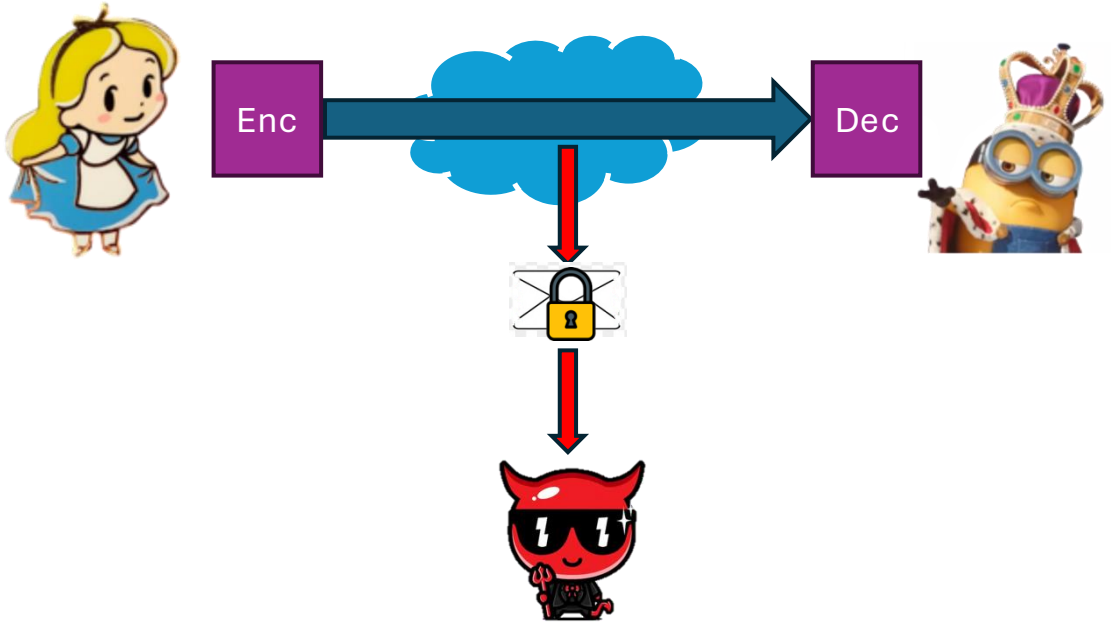


Len ()

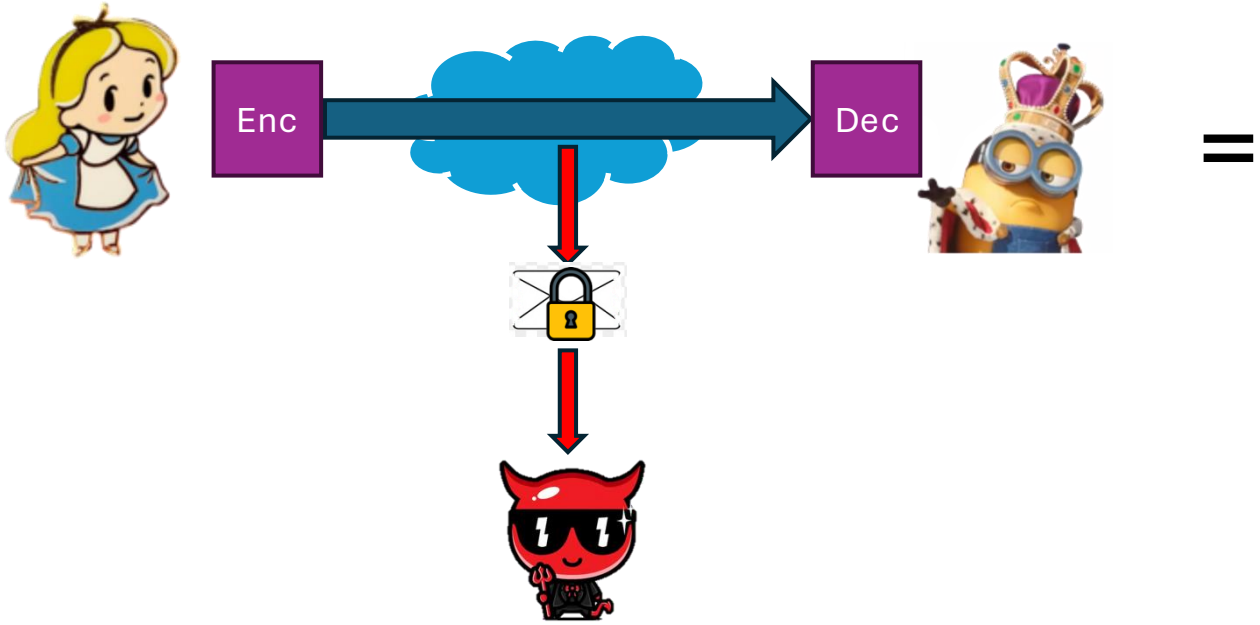


Real implements Ideal?

Real implements Ideal?



Real implements Ideal?



=

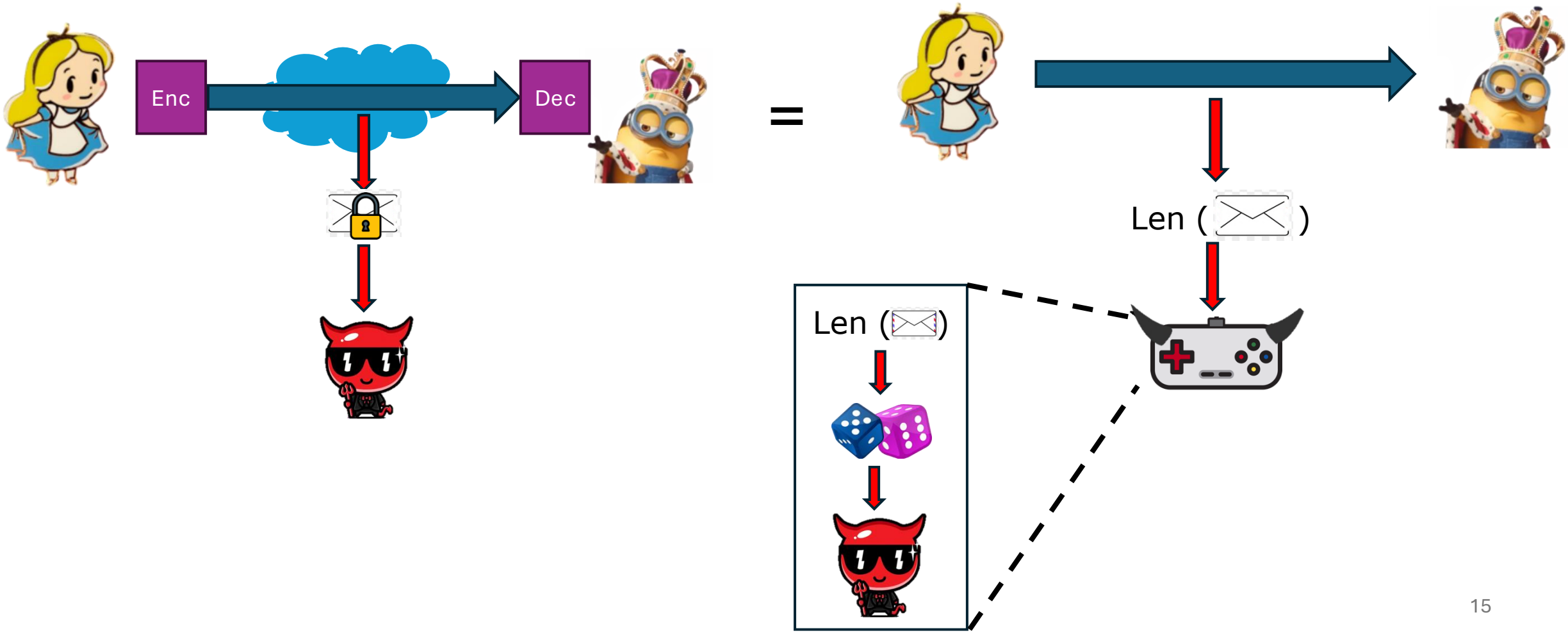
Real implements Ideal?



Real implements Ideal?

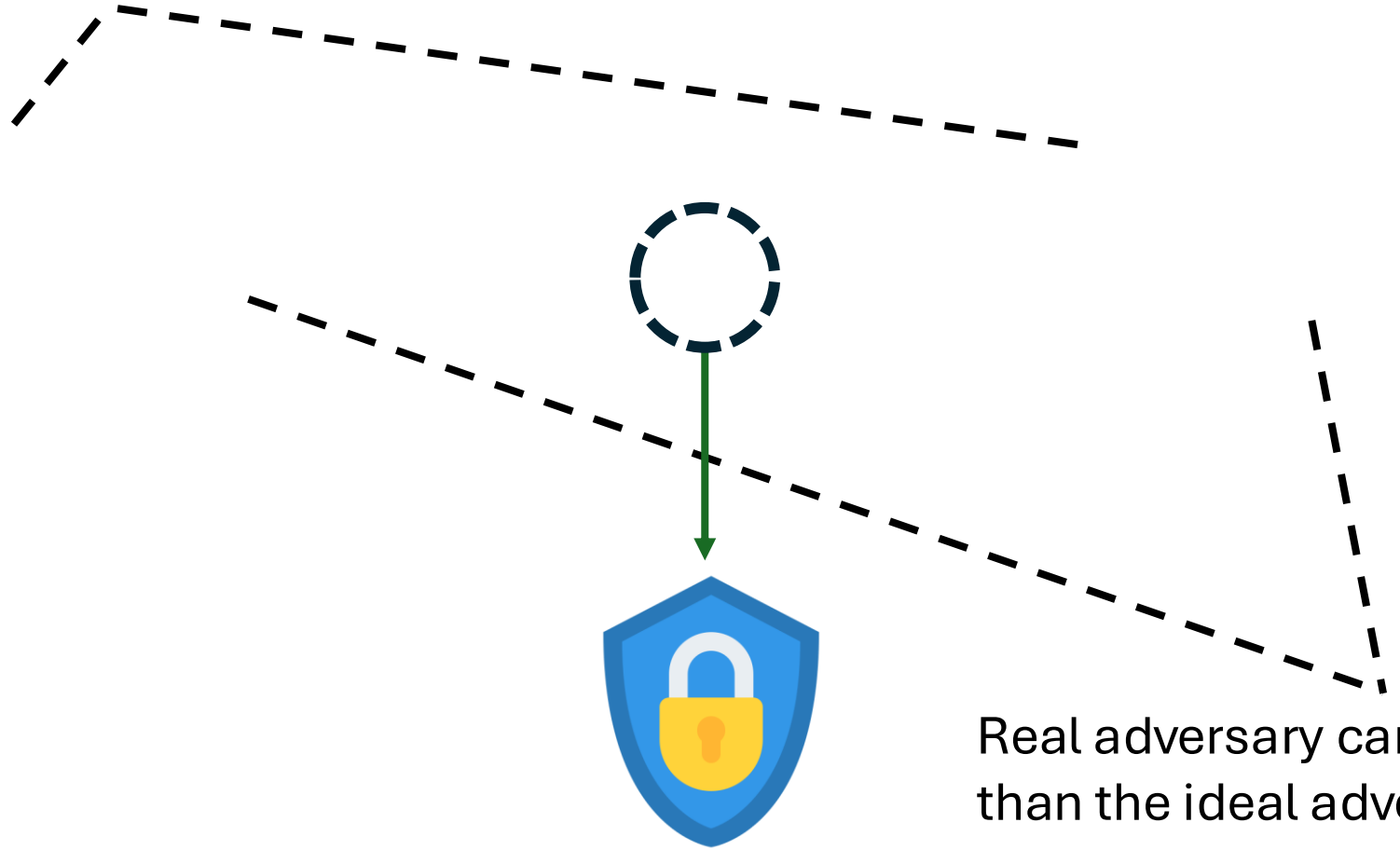


Real implements Ideal?



Features

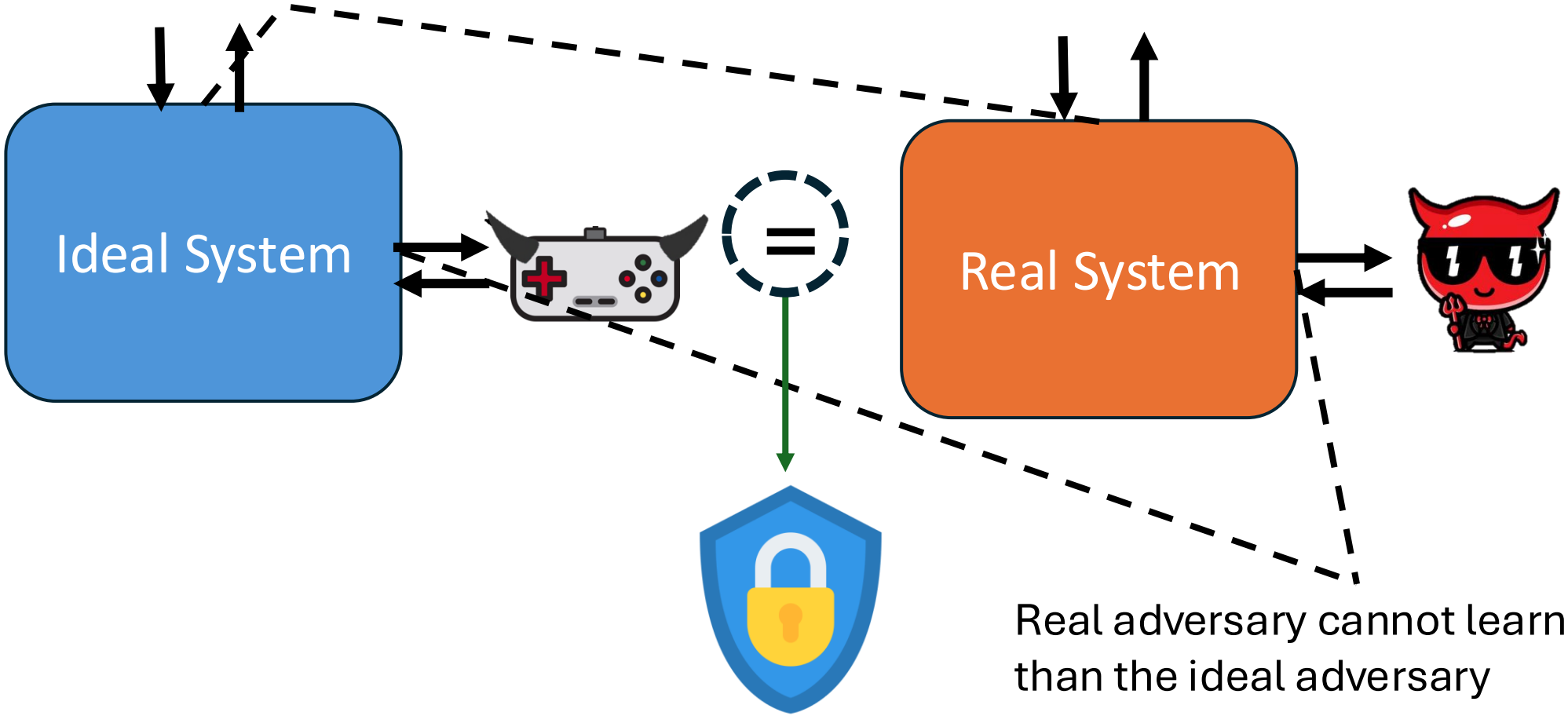
Input output behavior in both cases must match



Real adversary cannot learn **more** than the ideal adversary

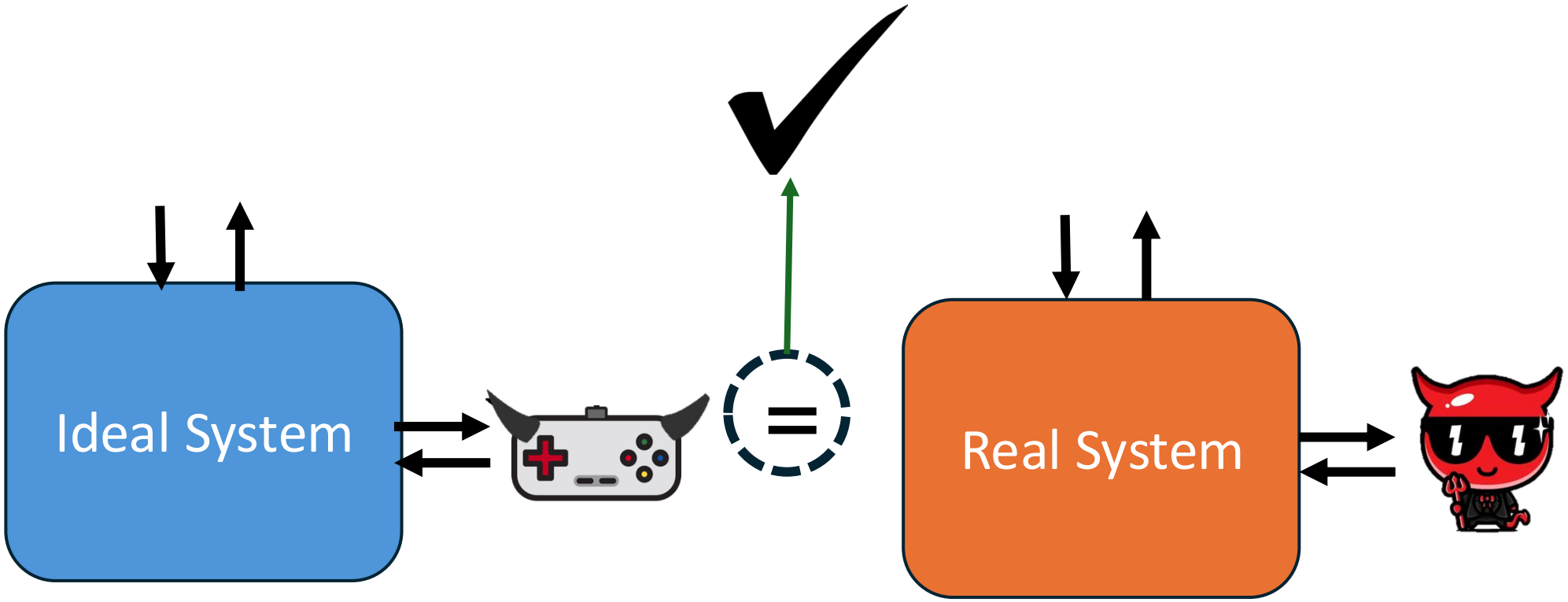
Features

Input output behavior in both cases must match



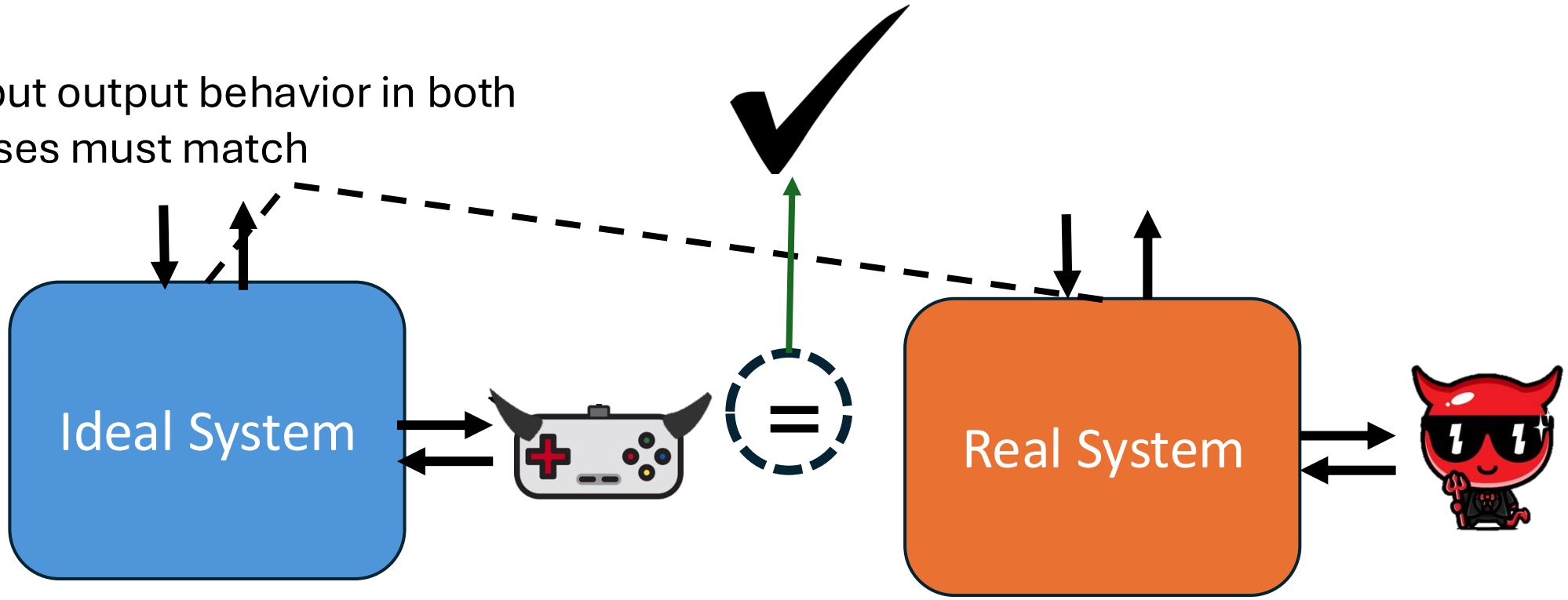
Real adversary cannot learn **more** than the ideal adversary

Features

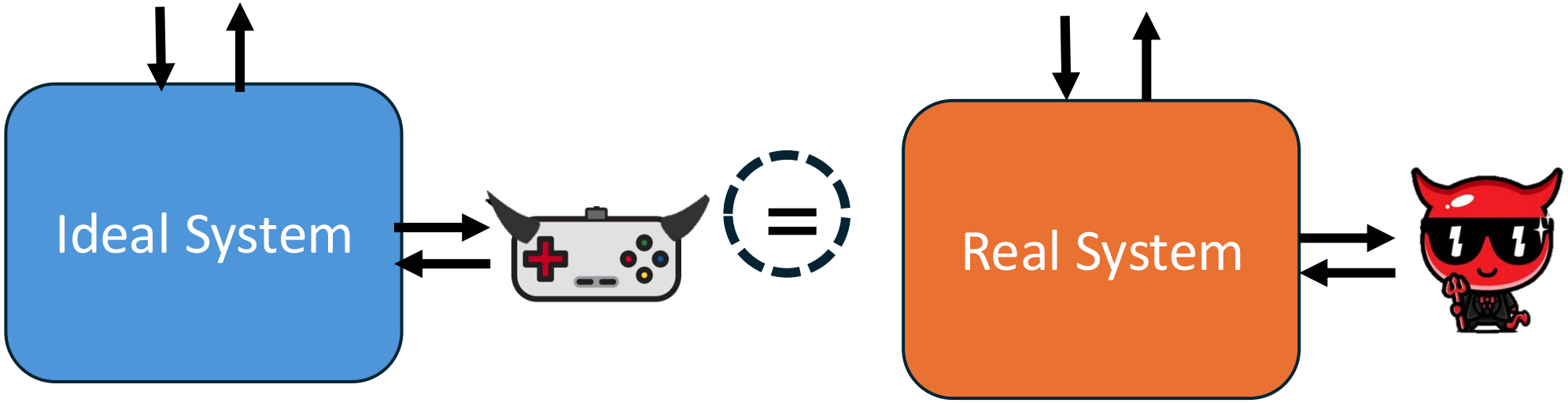


Features

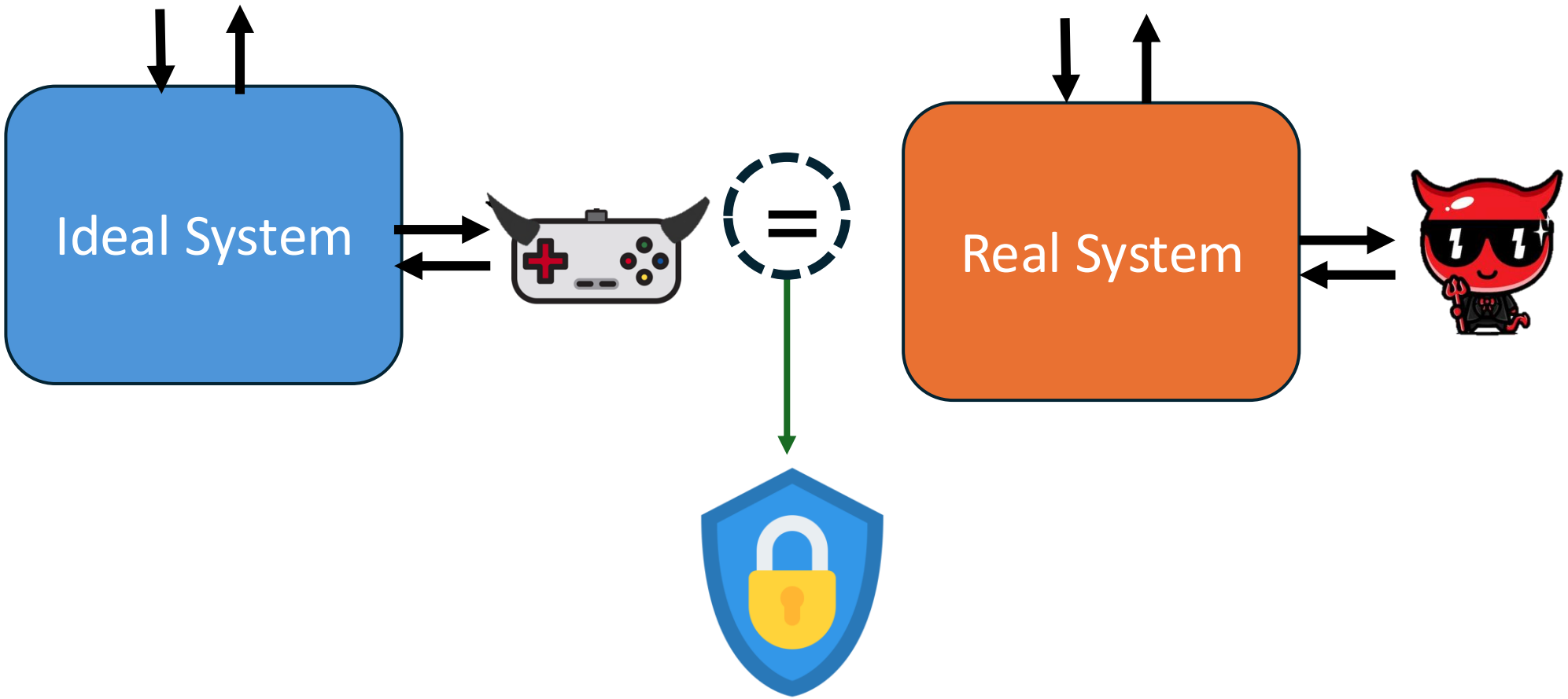
Input output behavior in both cases must match



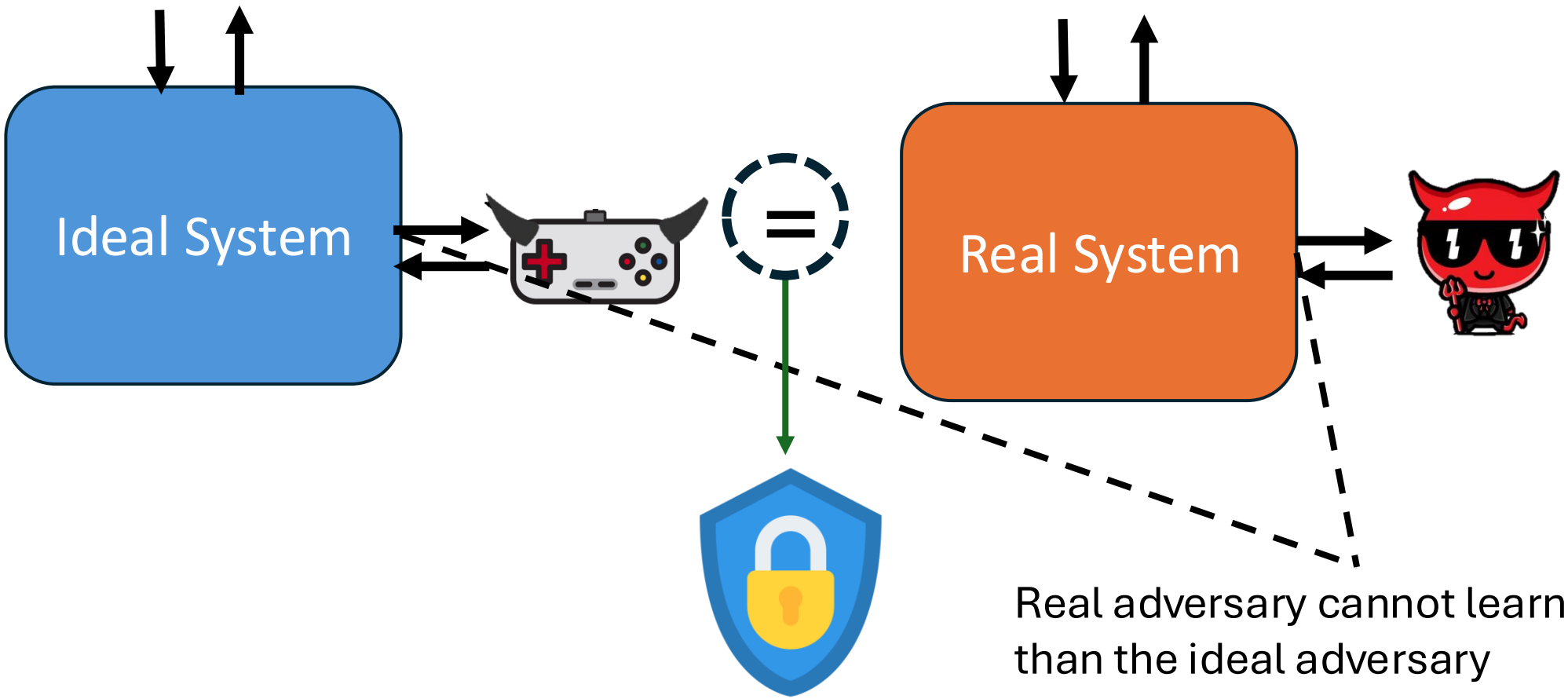
Features



Features



Features



Real adversary cannot learn **more** than the ideal adversary

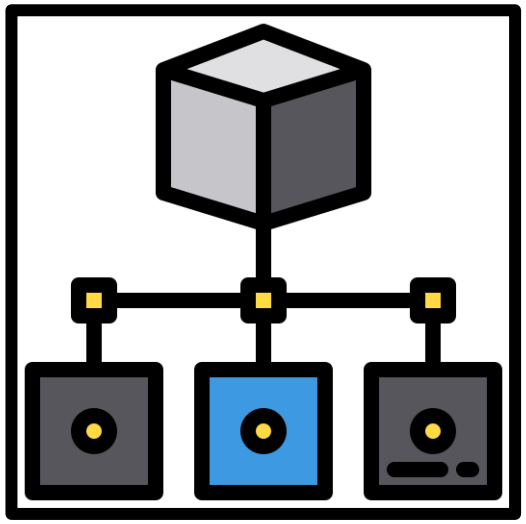
Research Directions

Research Directions



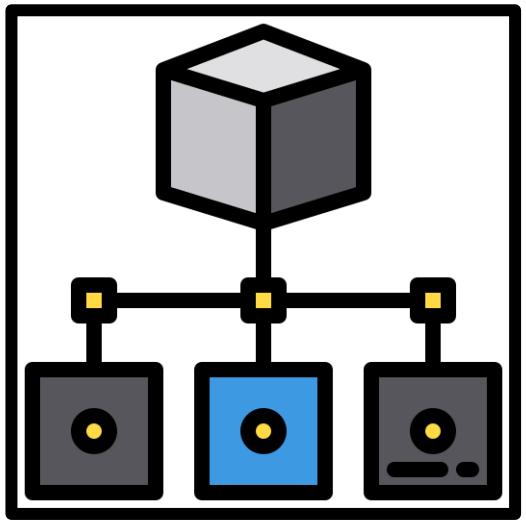
Research Directions

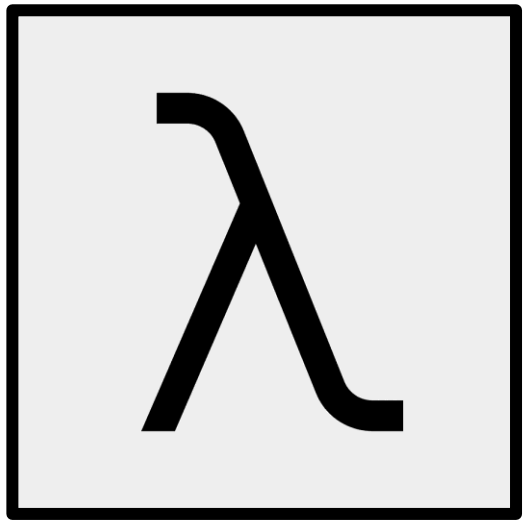




Research Directions

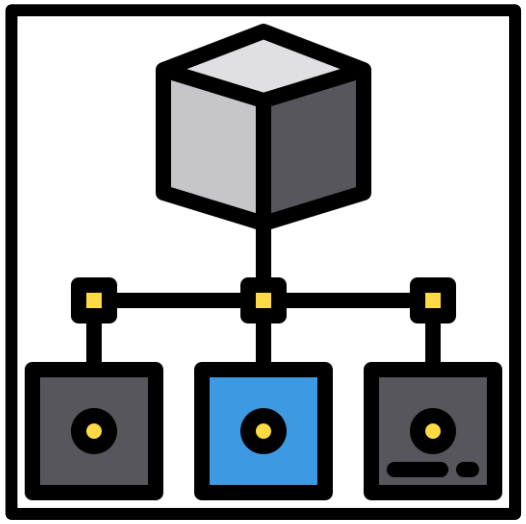


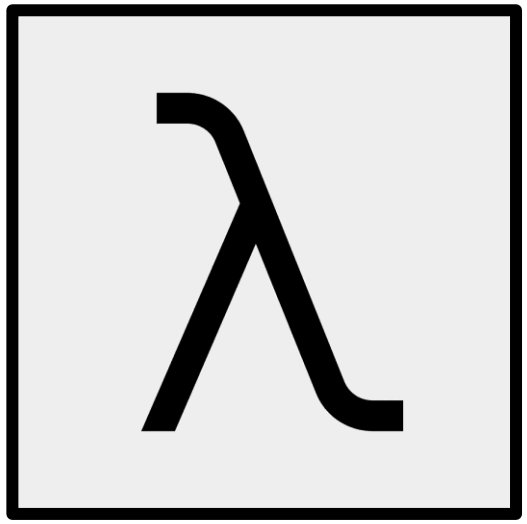




Research Directions





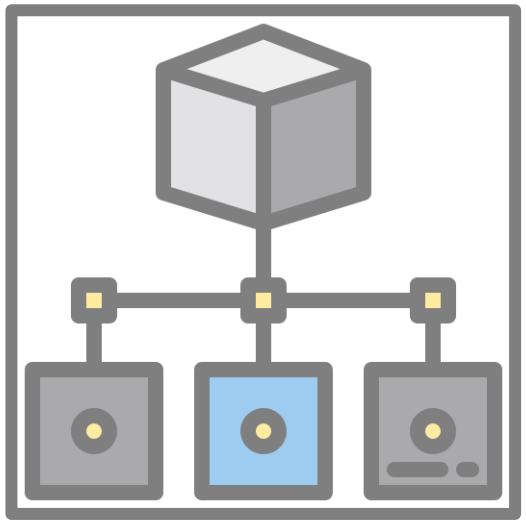





CAUTION
WORK
IN PROGRESS
CREATIVE safety supply
creativesafetysupply.com
866-777-1360
FS72927-24

Research Directions



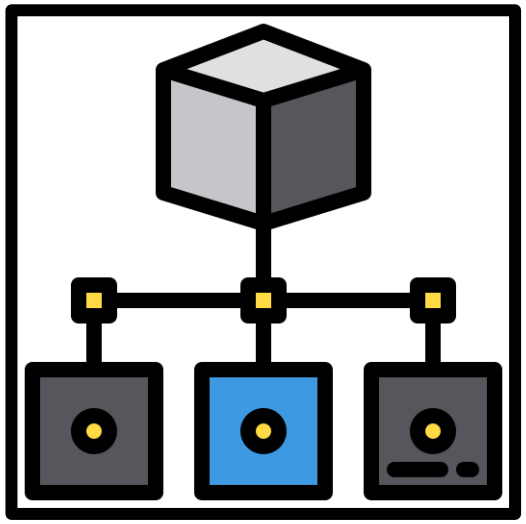






Research Directions









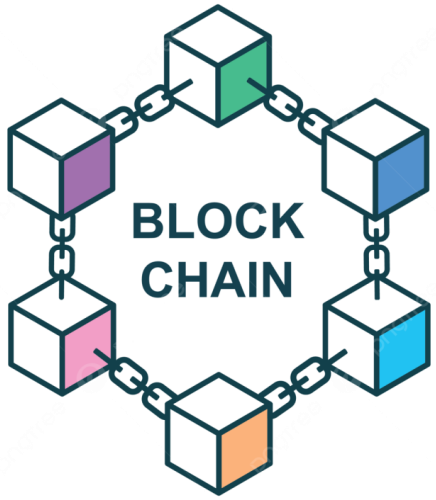
SCIF – Secure Smart Contracts

SCIF – Secure Smart Contracts



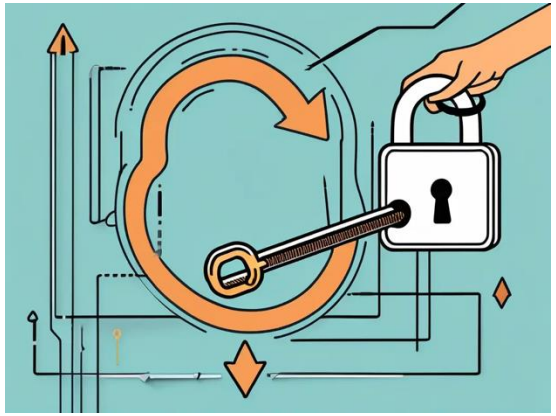
SCIF – Secure Smart Contracts

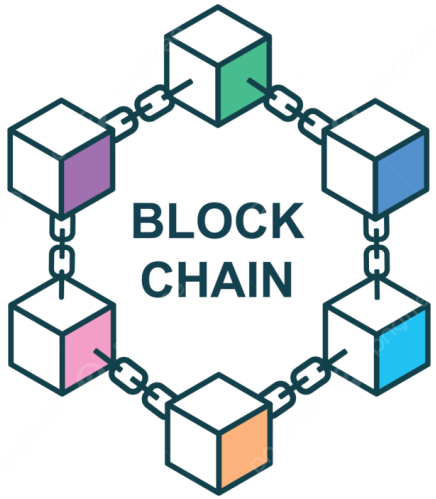




SCIF – Secure Smart Contracts



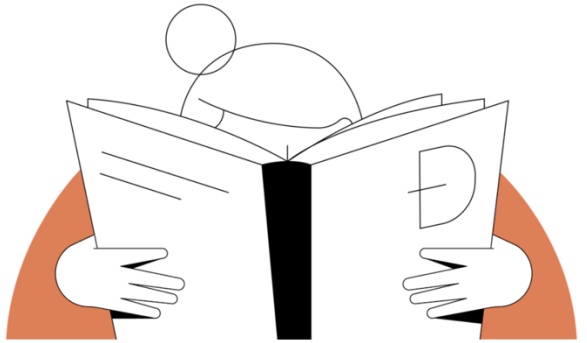




What Was The DAO?

Remember The DAO? This contentious event early in the history of blockchain shook the Ethereum community to its core.

By Cryptopedia Staff
Updated February 26, 2025 • 4 min read

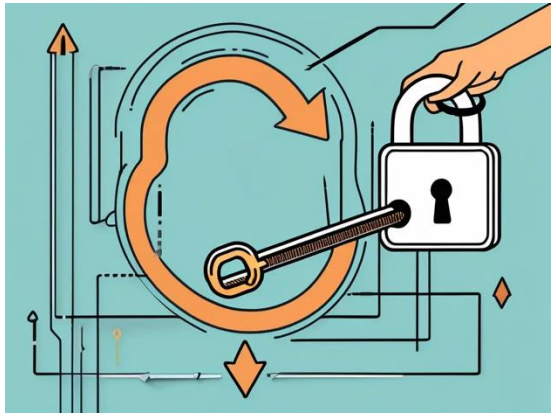


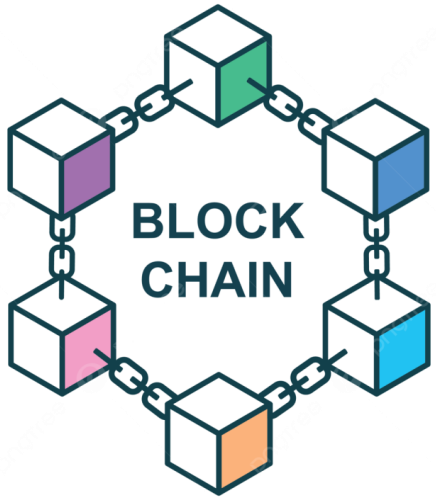
Summary

The DAO was a decentralized autonomous organization (DAO) that was launched in 2016 on the Ethereum blockchain. After raising $150 million USD worth of ether (ETH) through a token sale, The DAO was hacked due to vulnerabilities in its code base. The Ethereum blockchain was eventually hard forked to restore the stolen funds, but not all parties agreed with this decision, which resulted in the network splitting into two distinct blockchains: Ethereum and Ethereum Classic.

SCIF – Secure Smart Contracts





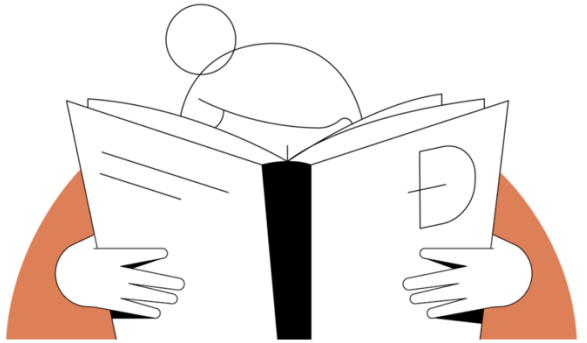




What Was The DAO?

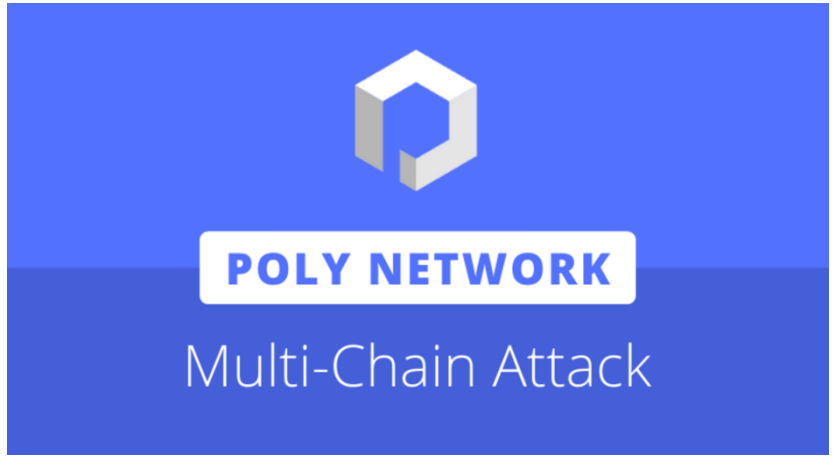
Remember The DAO? This contentious event early in the history of blockchain shook the Ethereum community to its core.

By Cryptopedia Staff
Updated February 26, 2025 • 4 min read



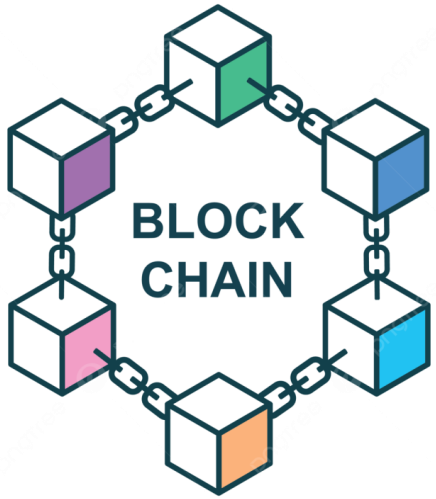
Summary

The DAO was a decentralized autonomous organization (DAO) that was launched in 2016 on the Ethereum blockchain. After raising $150 million USD worth of ether (ETH) through a token sale, The DAO was hacked due to vulnerabilities in its code base. The Ethereum blockchain was eventually hard forked to restore the stolen funds, but not all parties agreed with this decision, which resulted in the network splitting into two distinct blockchains: Ethereum and Ethereum Classic.



SCIF – Secure Smart Contracts



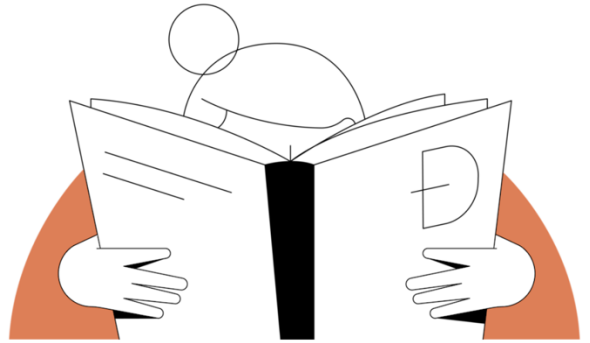




What Was The DAO?

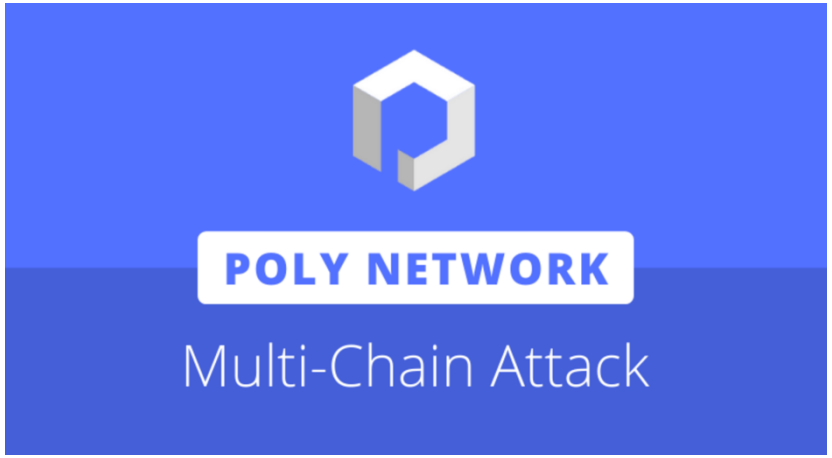
Remember The DAO? This contentious event early in the history of blockchain shook the Ethereum community to its core.

By Cryptopedia Staff
Updated February 26, 2025 • 4 min read



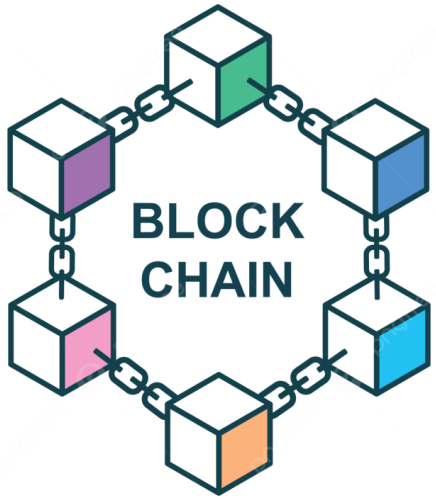
Summary

The DAO was a decentralized autonomous organization (DAO) that was launched in 2016 on the Ethereum blockchain. After raising $150 million USD worth of ether (ETH) through a token sale, The DAO was hacked due to vulnerabilities in its code base. The Ethereum blockchain was eventually hard forked to restore the stolen funds, but not all parties agreed with this decision, which resulted in the network splitting into two distinct blockchains: Ethereum and Ethereum Classic.



SCIF – Secure Smart Contracts



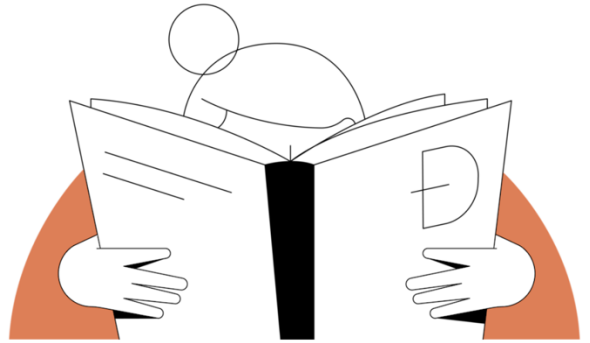




What Was The DAO?

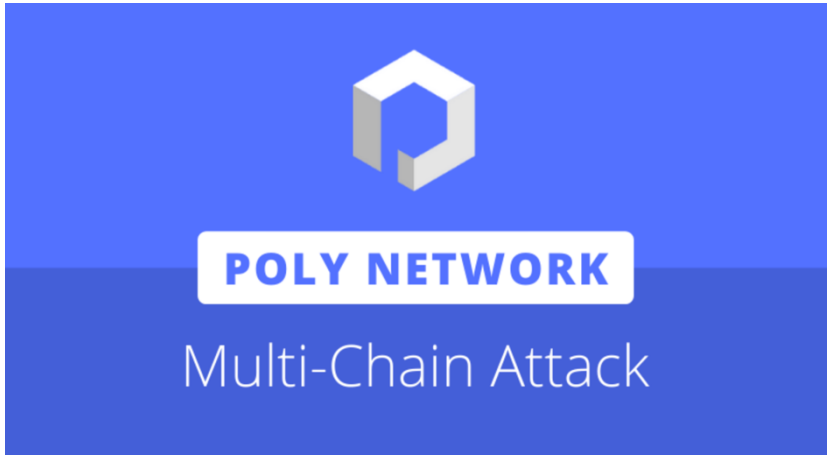
Remember The DAO? This contentious event early in the history of blockchain shook the Ethereum community to its core.

By Cryptopedia Staff
Updated February 26, 2025 • 4 min read



Summary

The DAO was a decentralized autonomous organization (DAO) that was launched in 2016 on the Ethereum blockchain. After raising $150 million USD worth of ether (ETH) through a token sale, The DAO was hacked due to vulnerabilities in its code base. The Ethereum blockchain was eventually hard forked to restore the stolen funds, but not all parties agreed with this decision, which resulted in the network splitting into two distinct blockchains: Ethereum and Ethereum Classic.



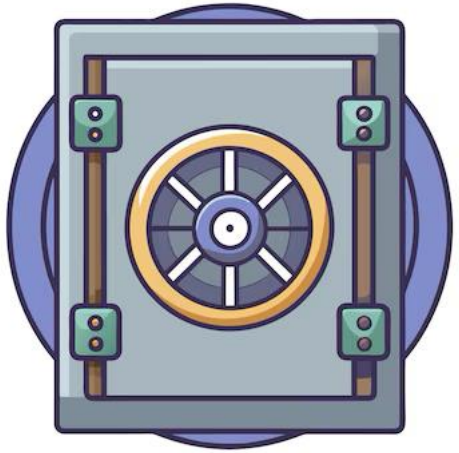
A Confused Deputy Attack

A Confused Deputy Attack

The Deputy



The Target

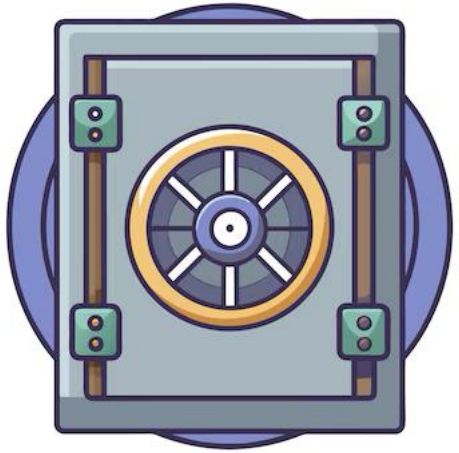


A Confused Deputy Attack

The Deputy



The Target

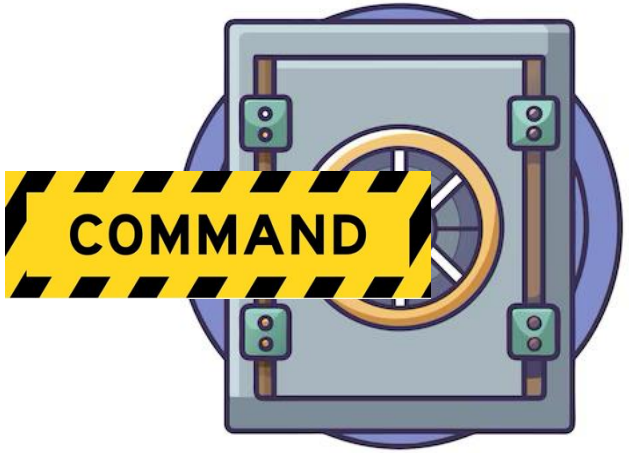


A Confused Deputy Attack

The Deputy



The Target



A Confused Deputy Attack

The Deputy



The Target

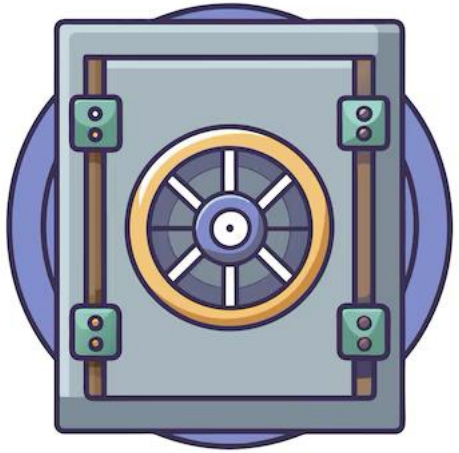


A Confused Deputy Attack

The Deputy



The Target

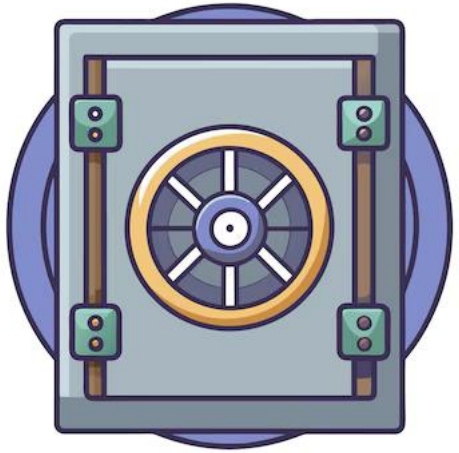


A Confused Deputy Attack

The Deputy



The Target





A Confused Deputy Attack

The Deputy



The Target



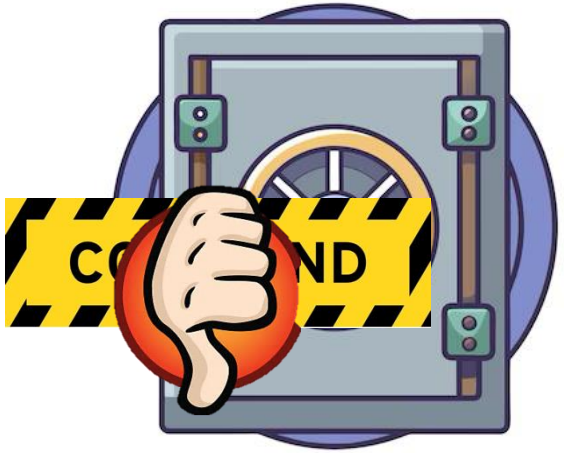


A Confused Deputy Attack

The Deputy



The Target



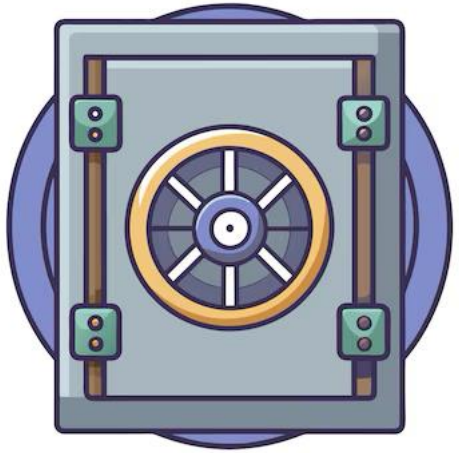


A Confused Deputy Attack

The Deputy



The Target



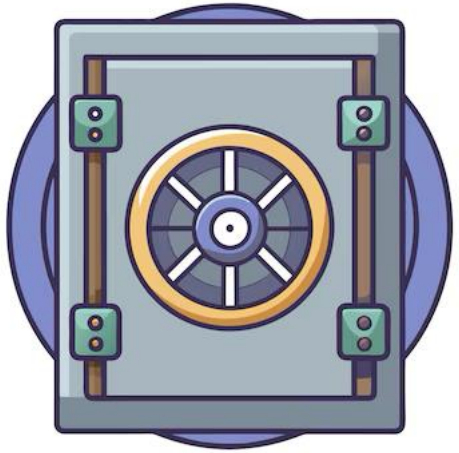


A Confused Deputy Attack

The Deputy



The Target



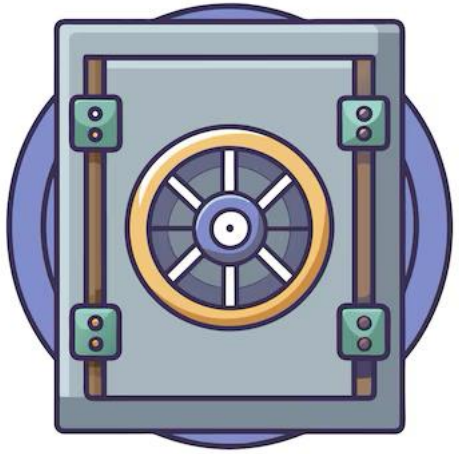


A Confused Deputy Attack

The Confused Deputy



The Target



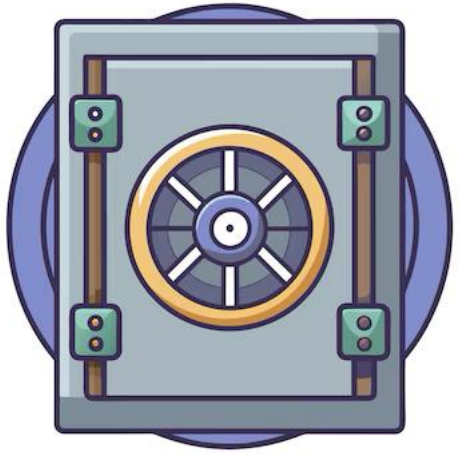


A Confused Deputy Attack

The Confused Deputy



The Target





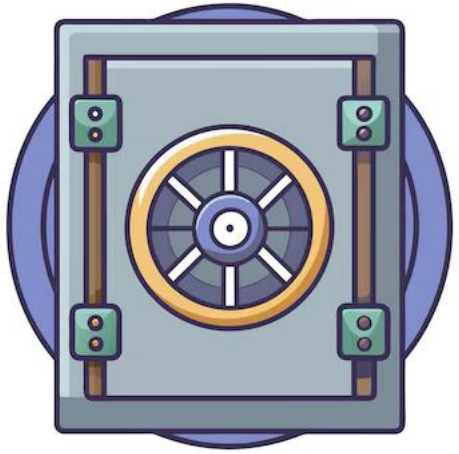
A Confused Deputy Attack

The Confused Deputy



“Do COMMAND on Target”

The Target



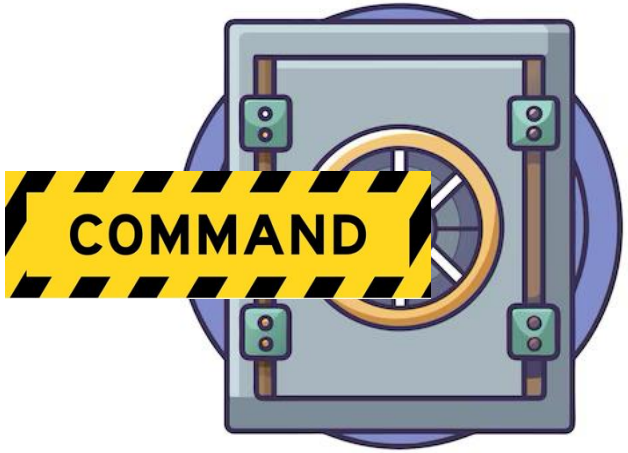


A Confused Deputy Attack

The Confused Deputy



The Target





A Confused Deputy Attack

The Confused Deputy



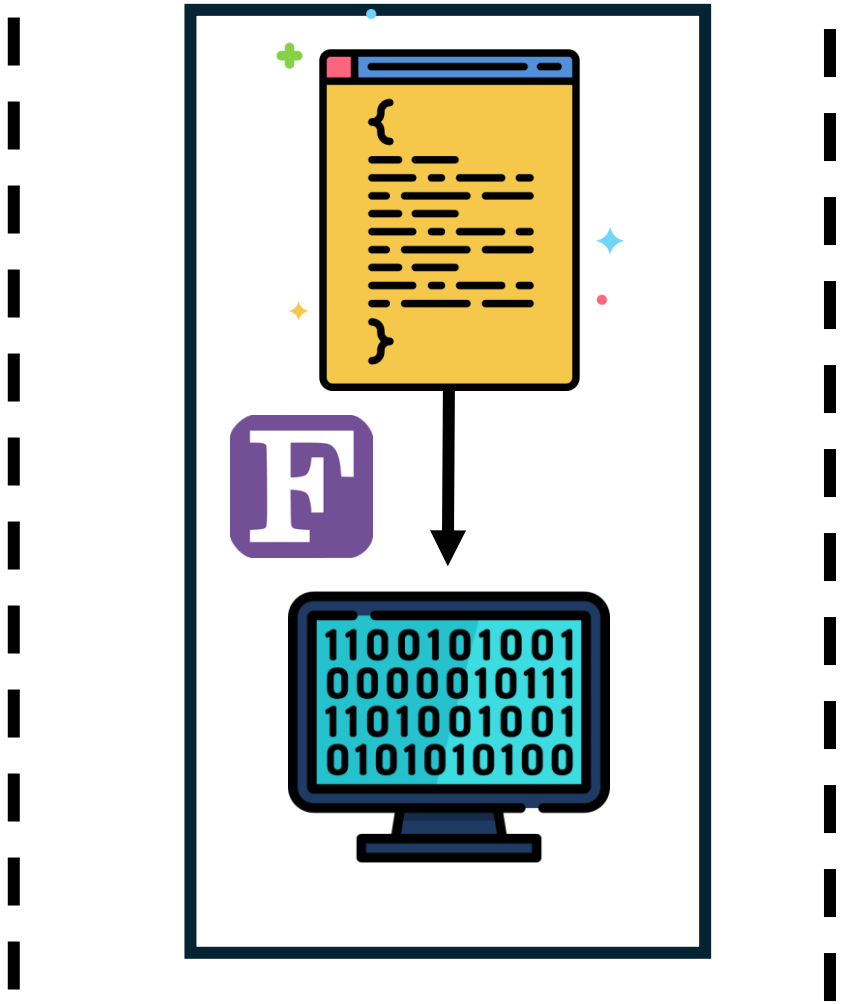
The Target





The Classic Example

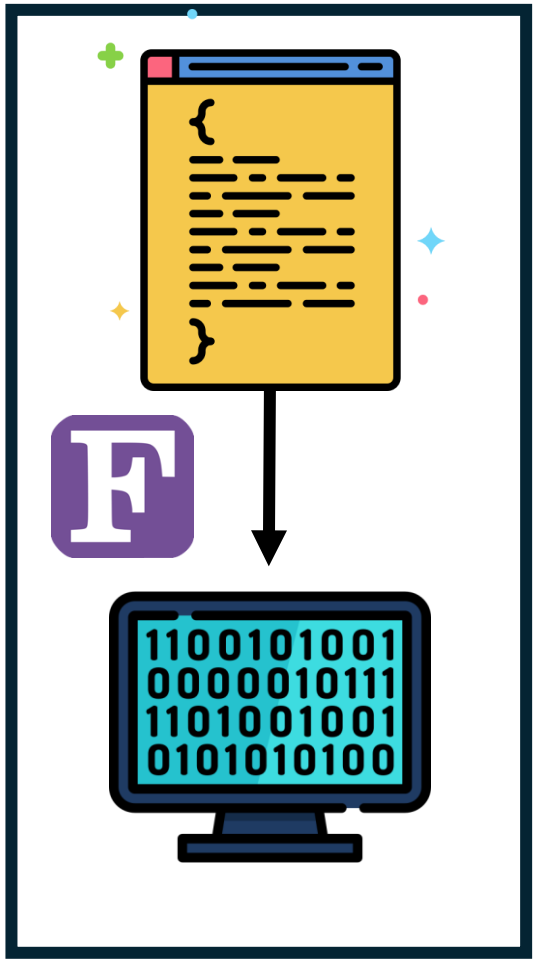
The Classic Example



The Classic Example

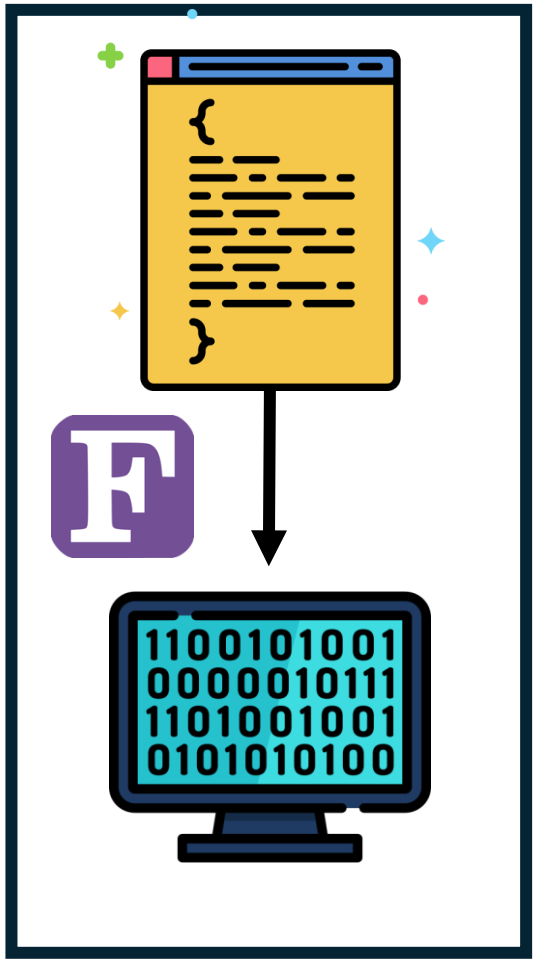
compile to /bob/output





The Classic Example



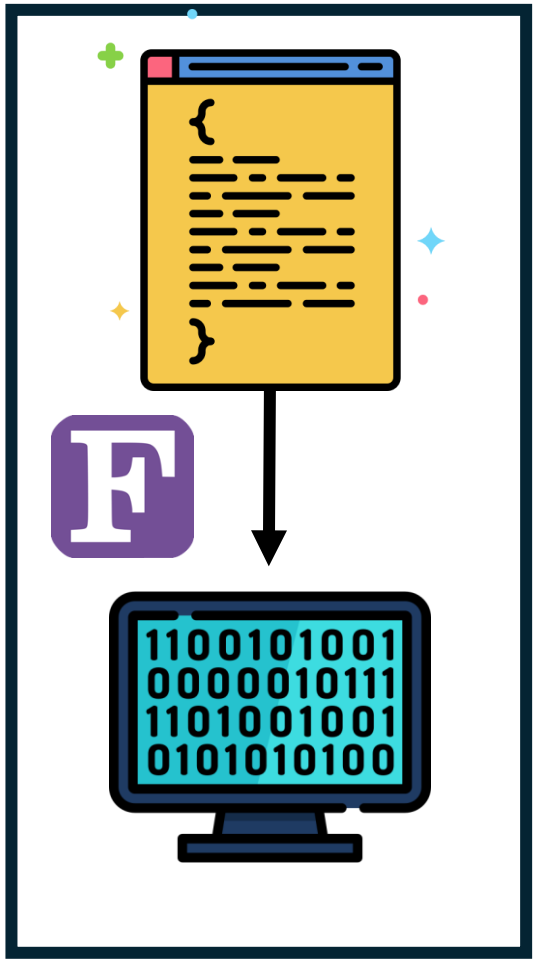




/bob/output

The Classic Example





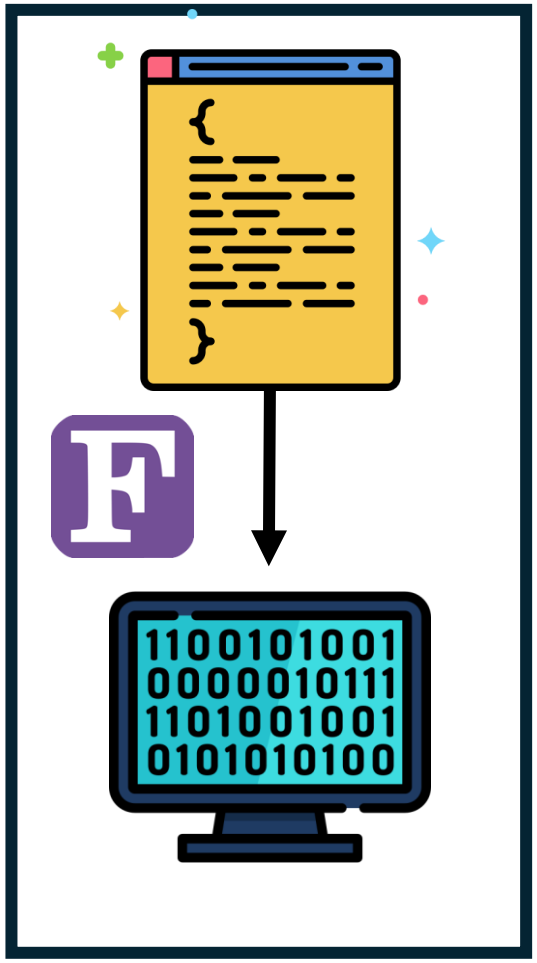




/bob/output

The Classic Example





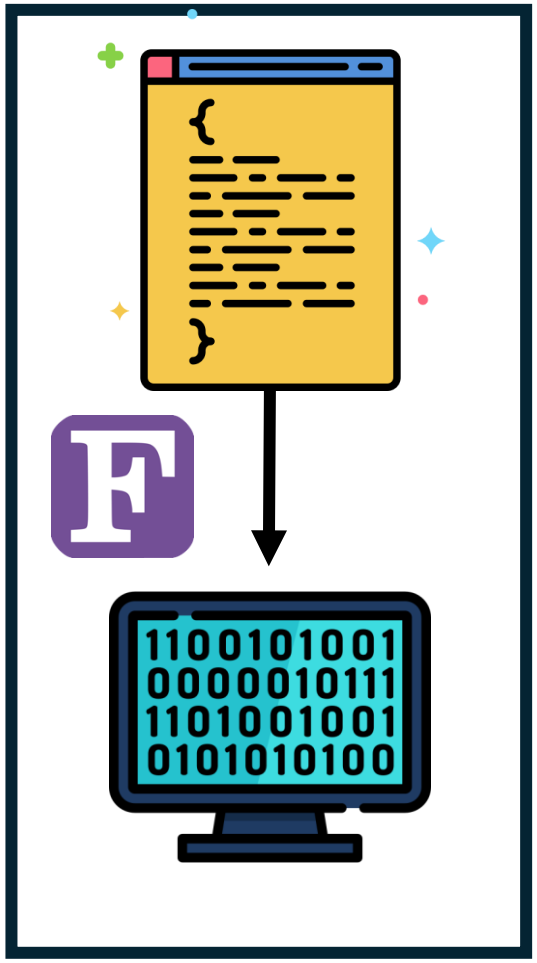




/bob/output

The Classic Example









/bob/output

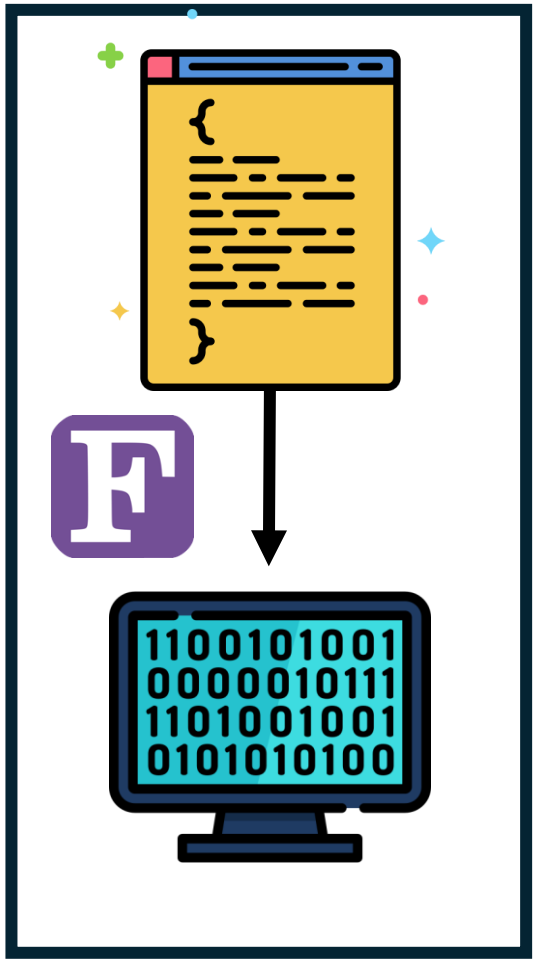
/sysx/billing
/sysx/logs

The Classic Example



compile to /bob/output







/bob/output



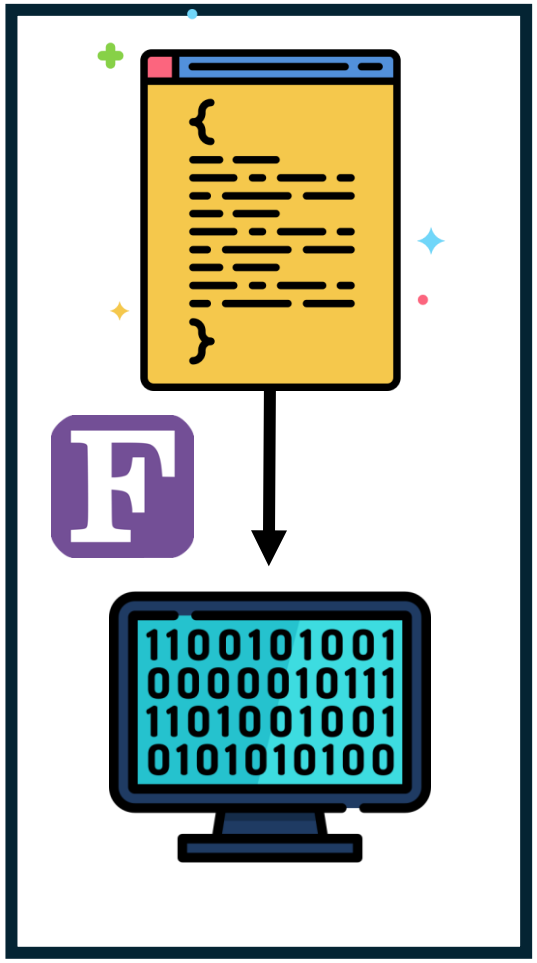
/sysx/billing
/sysx/logs

The Classic Example



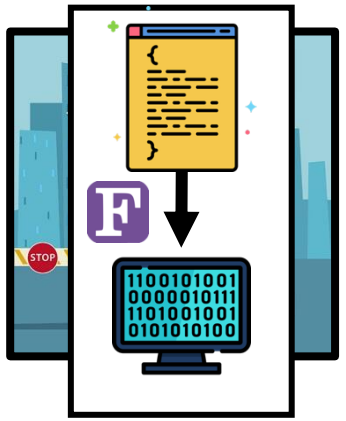
compile to /bob/output







/bob/output



/sysx/billing
/sysx/logs

The Classic Example

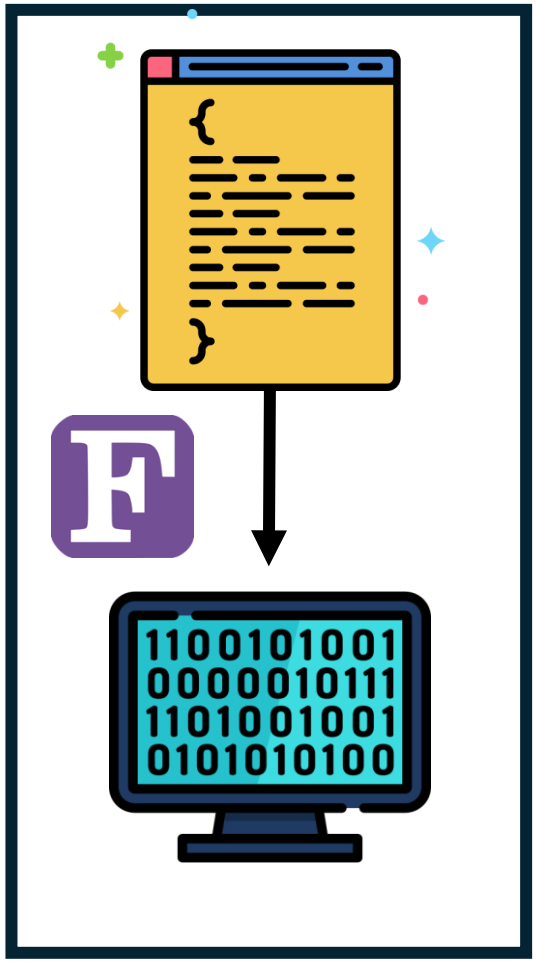


compile to /bob/output



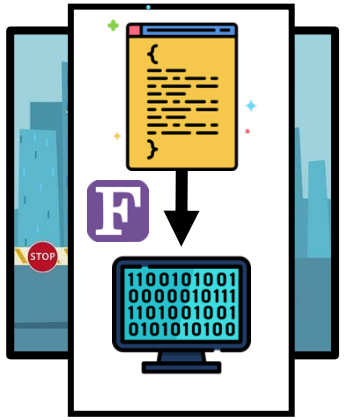
compile to /sysx/billing







/bob/output



/sysx/billing
/sysx/logs

The Classic Example

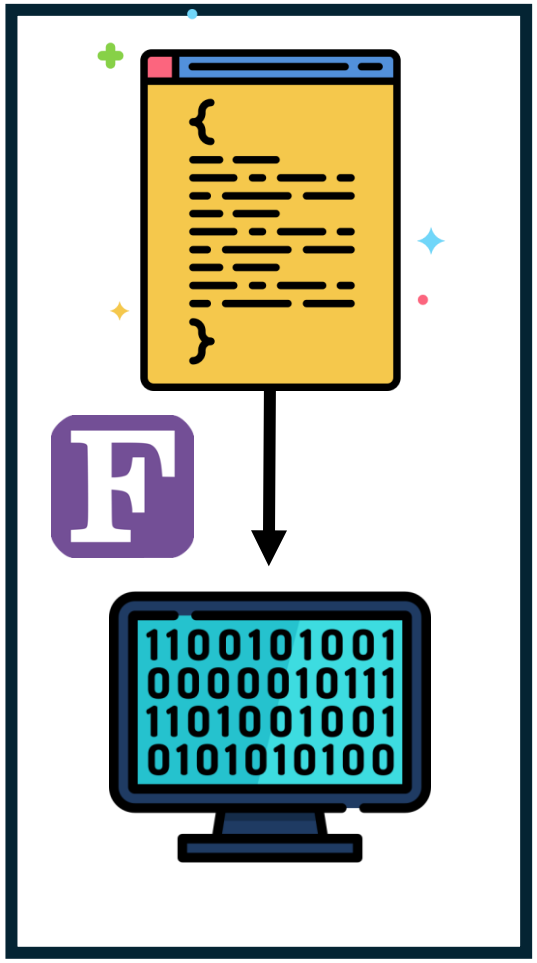


compile to /bob/output



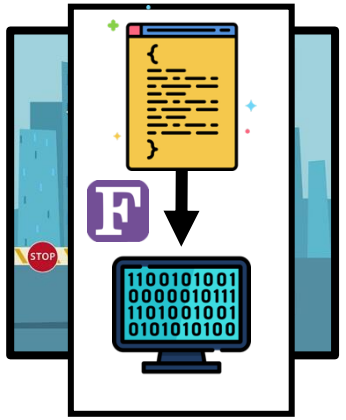
compile to /sysx/billing







/bob/output



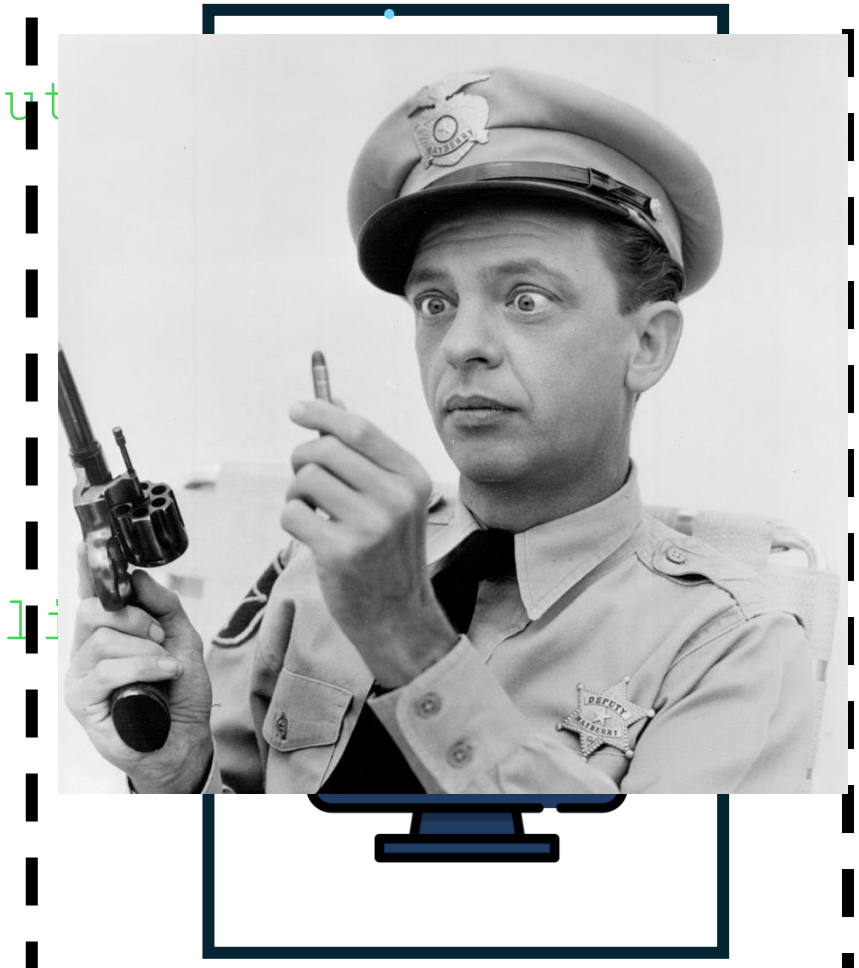
/sysx/billing
/sysx/logs

The Classic Example



compile to /bob/output



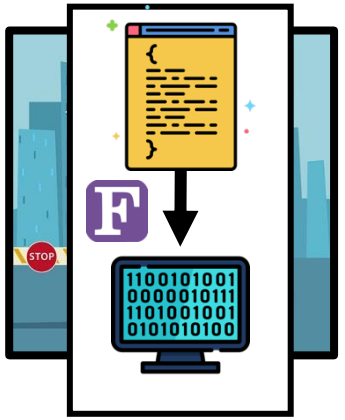




/bob/output

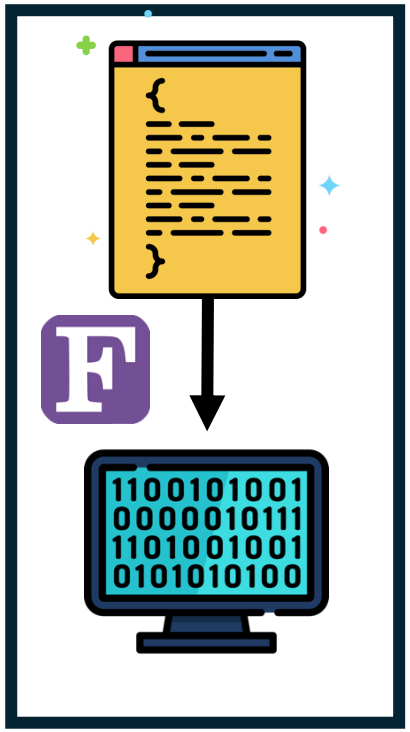
compile to /sysx/billi





/sysx/billing
/sysx/logs

Authority Confusion

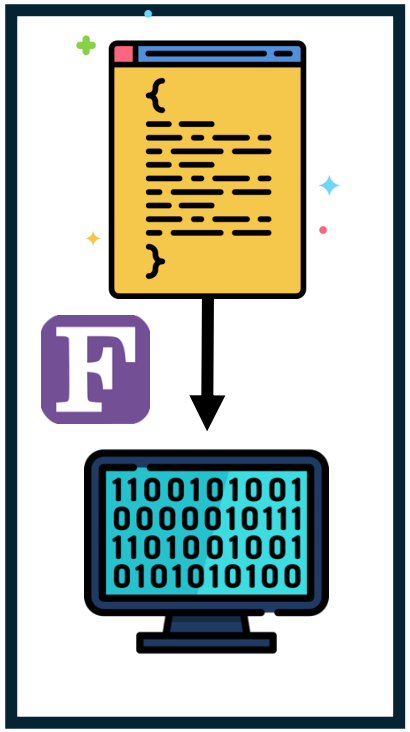


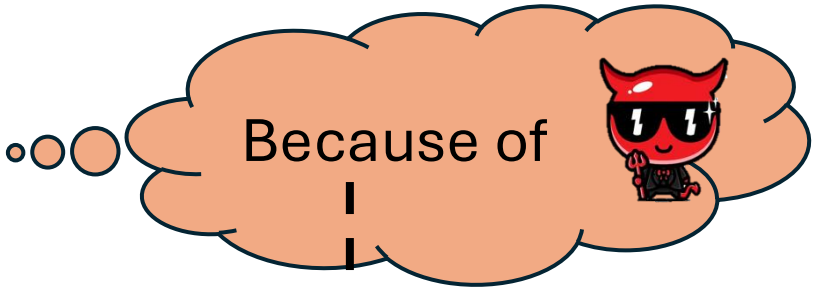

```
compile to /sysx/billing
```





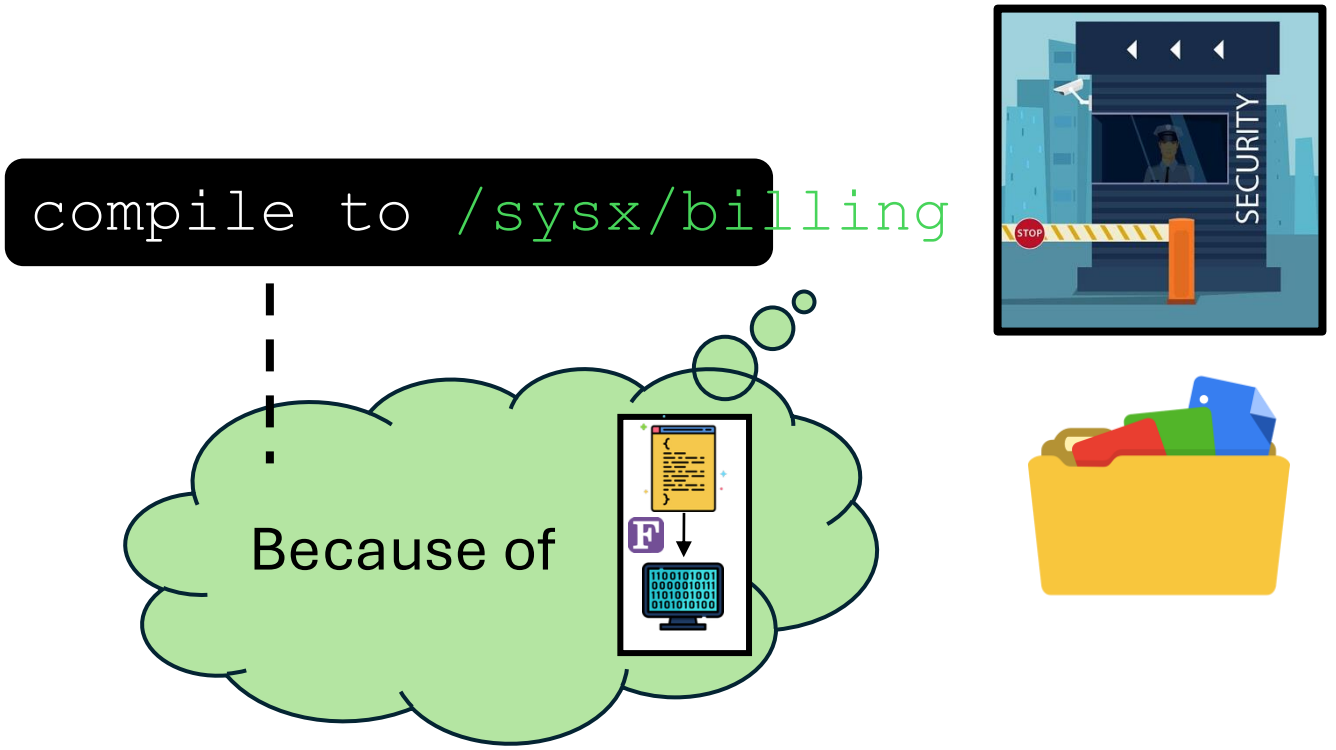
Authority Confusion



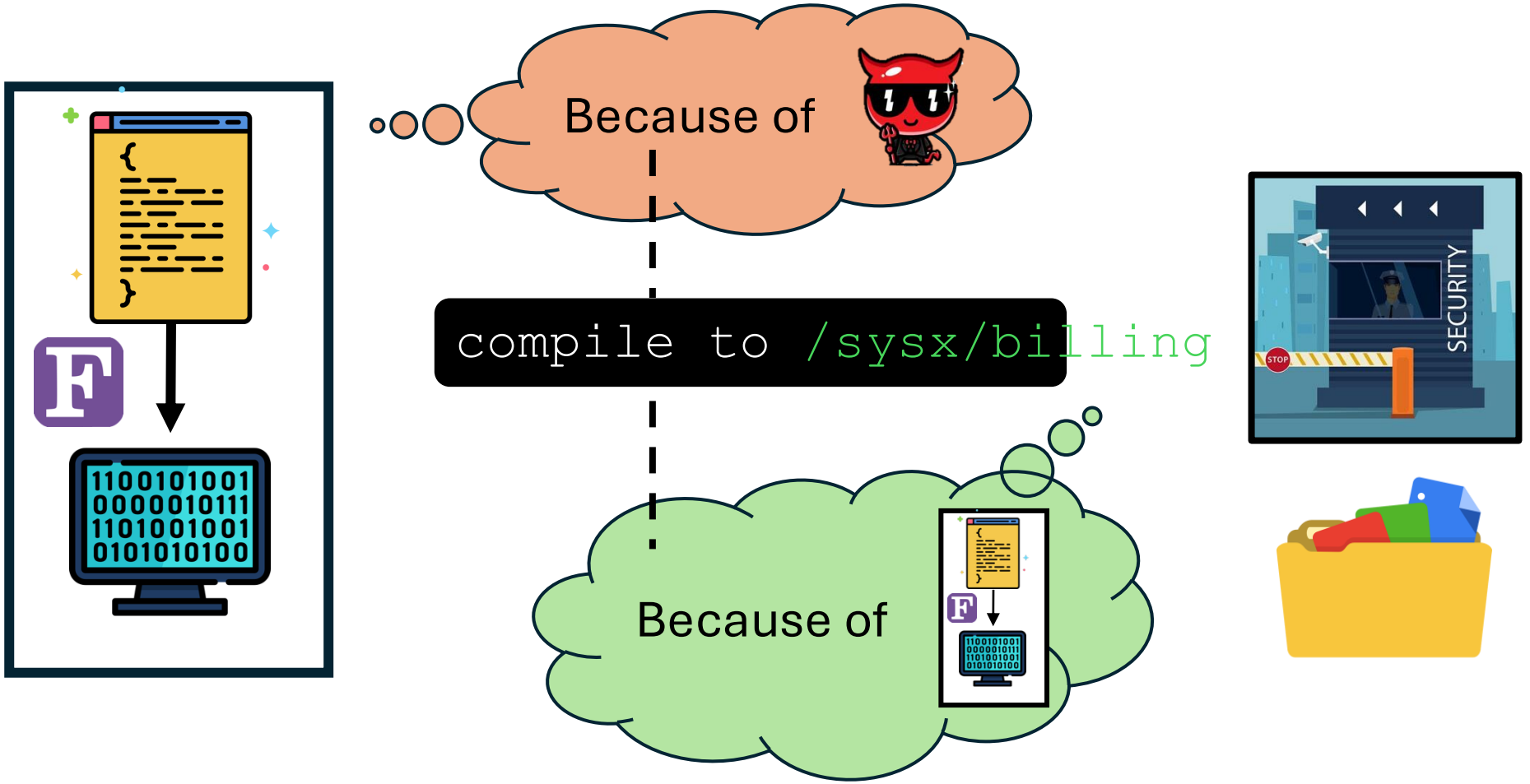



```
compile to /sysx/billing
```

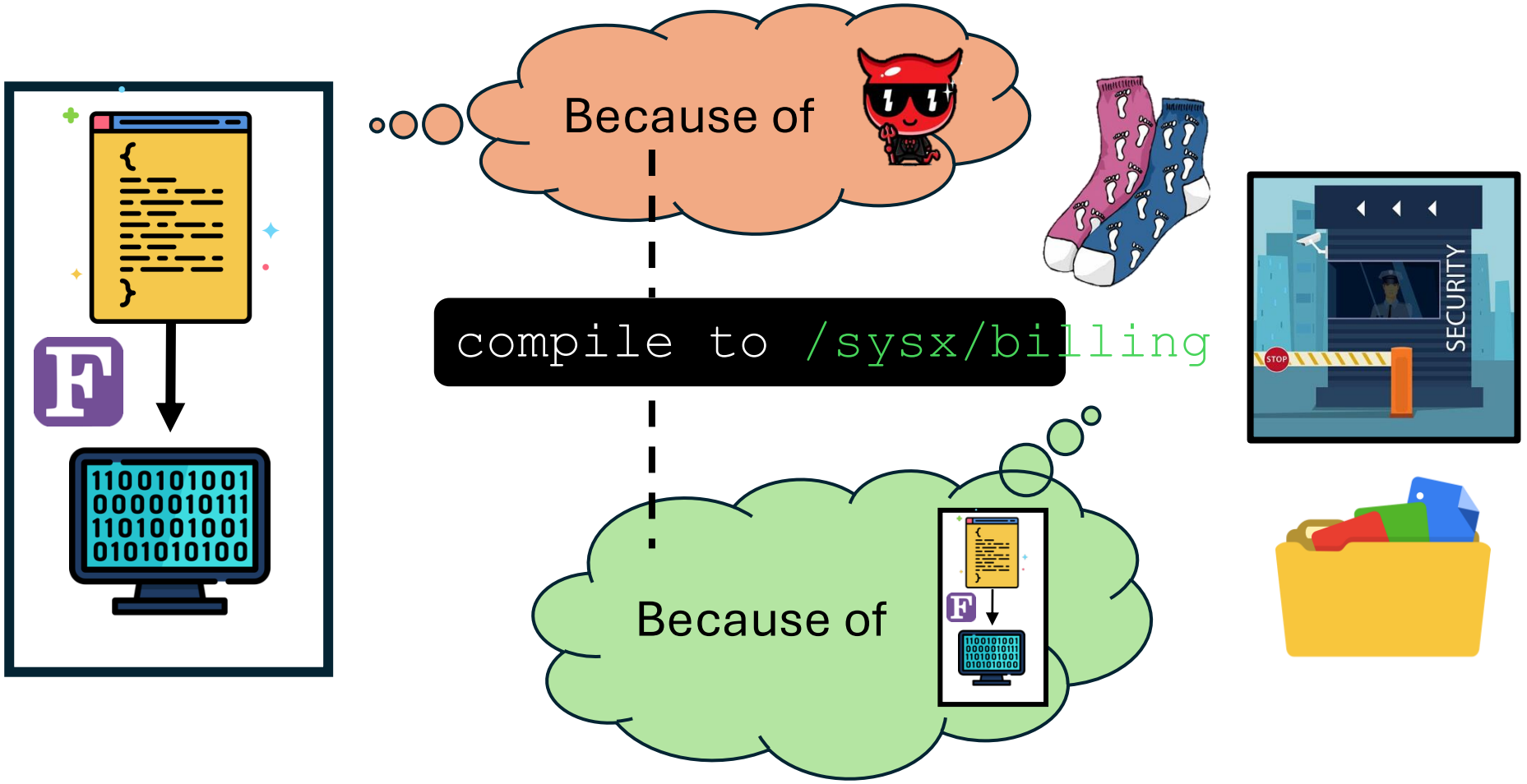
Authority Confusion



Authority Confusion

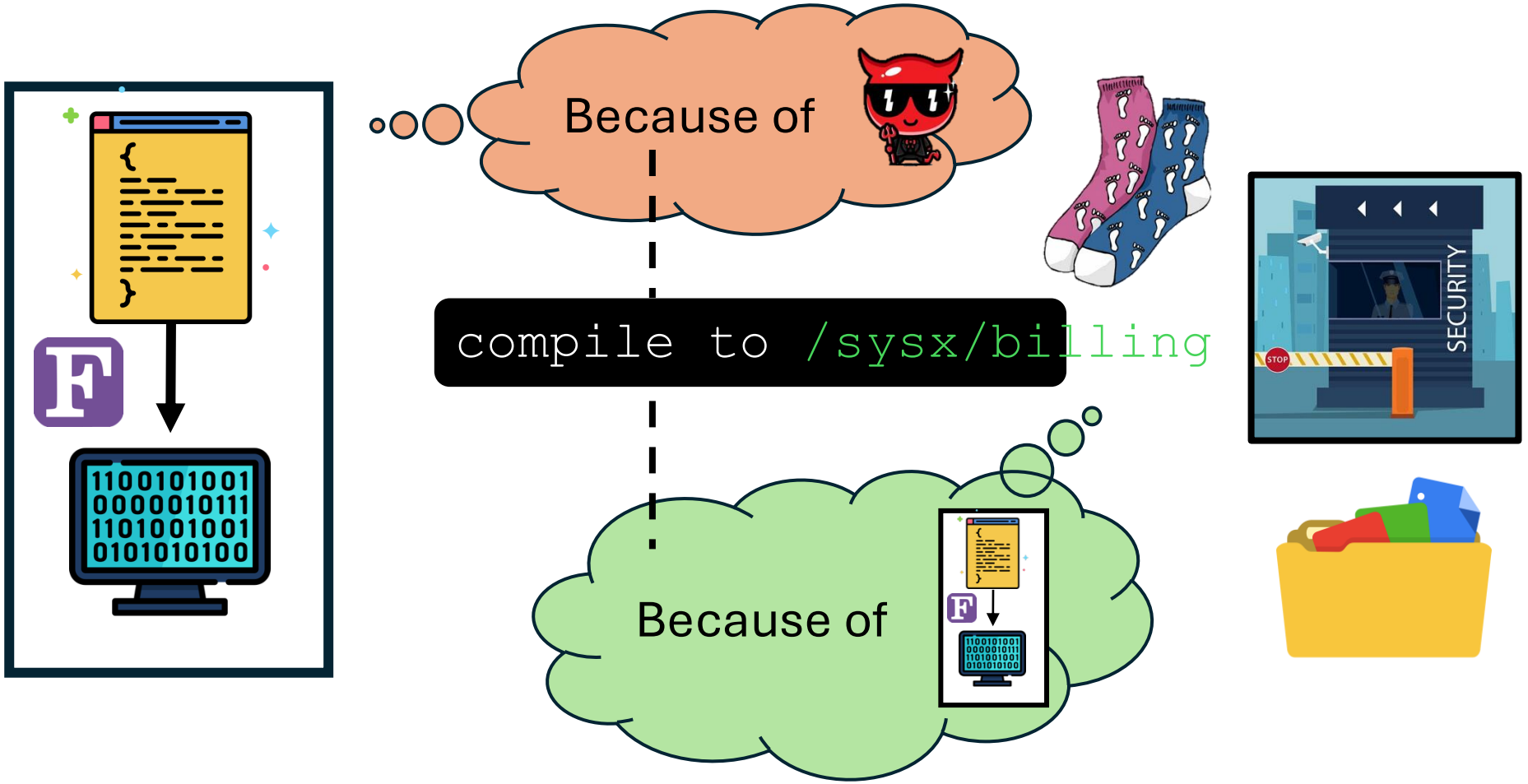


Authority Confusion



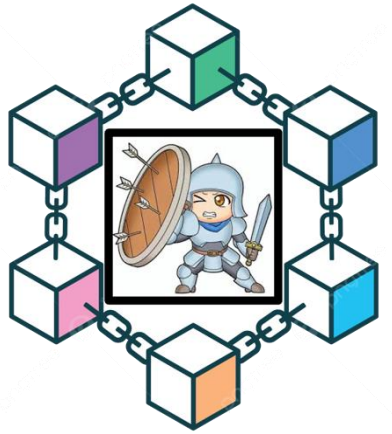
Authority Confusion

Notion of authority can be captured by information-flow labels!

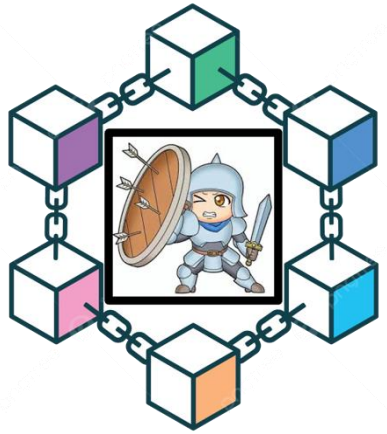


The Real-Ideal Paradigm?

The Real-Ideal Paradigm?

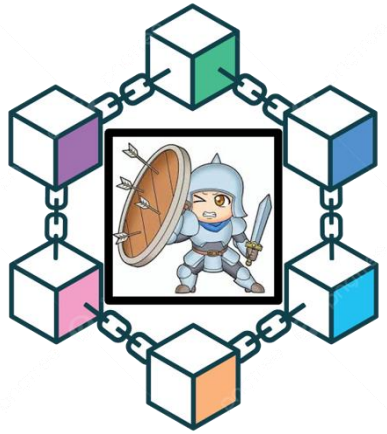


The Real-Ideal Paradigm?



Real Situation

The Real-Ideal Paradigm?

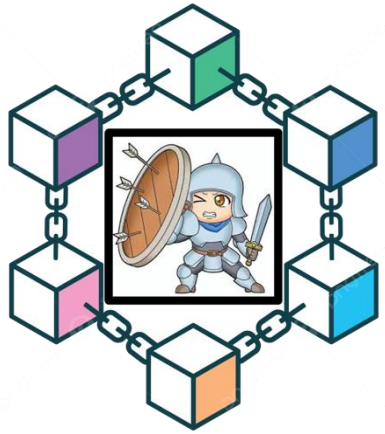


Real Situation



Actions don’t respect information flow types

The Real-Ideal Paradigm?

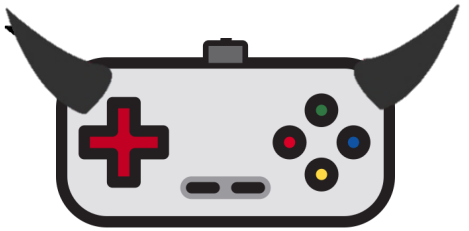


Real Situation



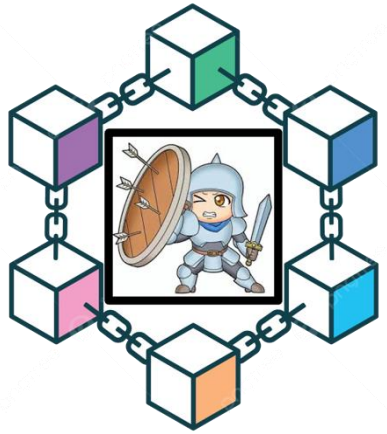
Actions don’t respect information flow types

Ideal Situation



Actions respect information flow types

The Real-Ideal Paradigm?



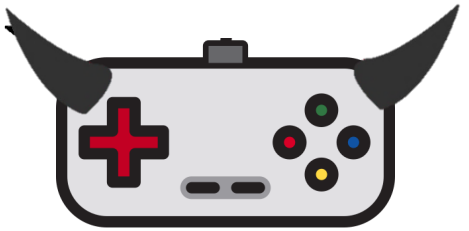
Real Situation

∀



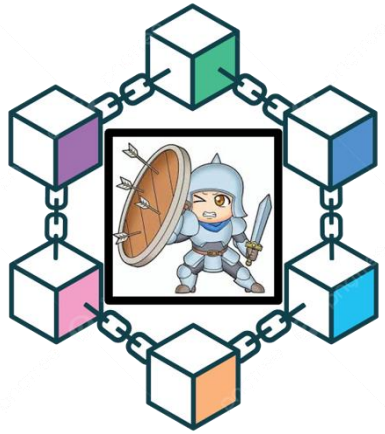
Actions don’t respect information flow types

Ideal Situation



Actions respect information flow types

The Real-Ideal Paradigm?



Real Situation

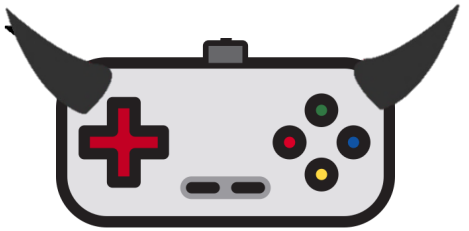
∀



Actions don’t respect information flow types

Ideal Situation

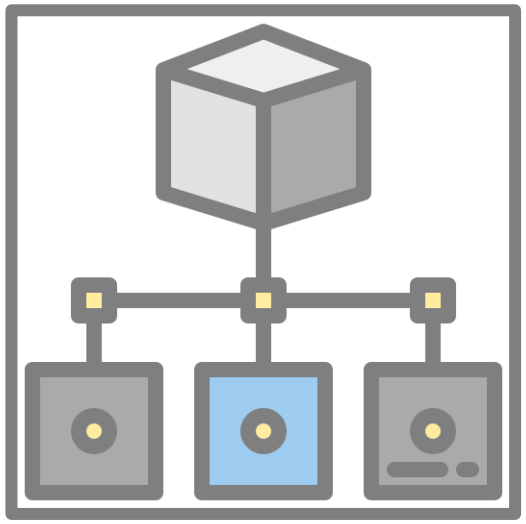
∃



Actions respect information flow types

Research Directions



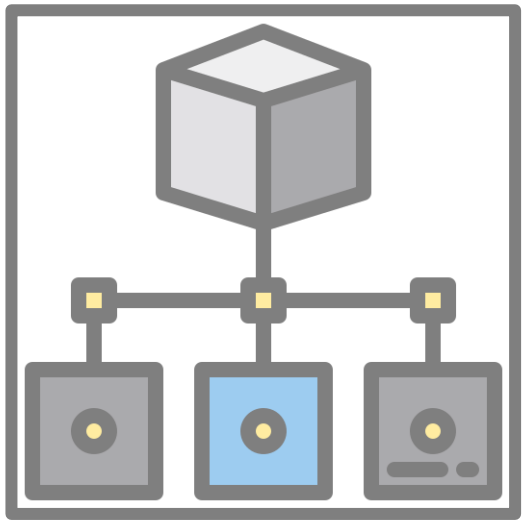


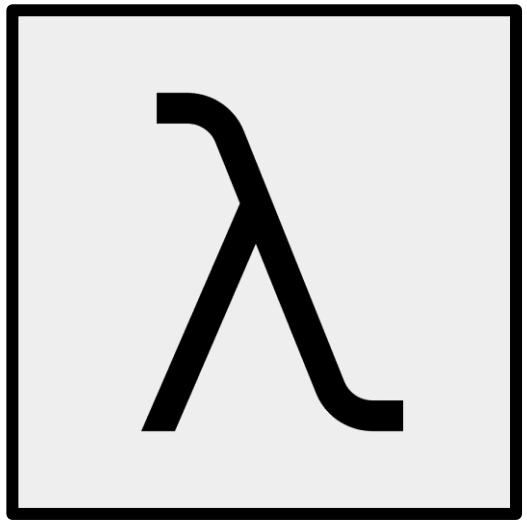




Research Directions









NMIFC

NMIFC

Nonmalleable Information Flow Control

Authors:  Ethan Cecchetti,  Andrew C. Myers,  Owen Arden | Authors Info & Claims

CCS '17: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security • Pages 1875 - 1891
https://doi.org/10.1145/3133956.3134054

NMIFC

Nonmalleable Information Flow Control

Authors:  Ethan Cecchetti,  Andrew C. Myers,  Owen Arden | Authors Info & Claims

CCS '17: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security • Pages 1875 - 1891
https://doi.org/10.1145/3133956.3134054

$$\Gamma \vdash t : \tau$$

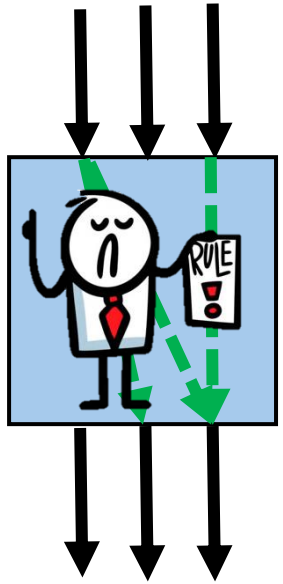
NMIFC

Nonmalleable Information Flow Control

Authors:  Ethan Cecchetti,  Andrew C. Myers,  Owen Arden | Authors Info & Claims

CCS '17: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security • Pages 1875 - 1891
https://doi.org/10.1145/3133956.3134054

$$\Gamma \vdash t : \tau$$



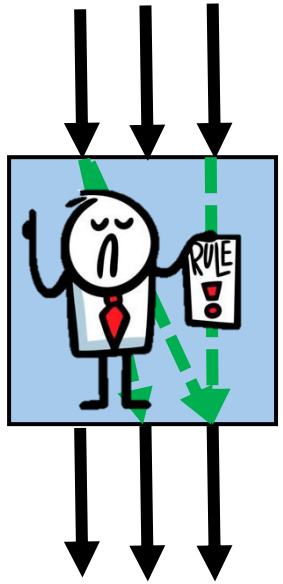
NMIFC

Nonmalleable Information Flow Control

Authors:  Ethan Cecchetti,  Andrew C. Myers,  Owen Arden | Authors Info & Claims

CCS '17: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security • Pages 1875 - 1891
https://doi.org/10.1145/3133956.3134054

$$\Gamma \vdash t : \tau$$





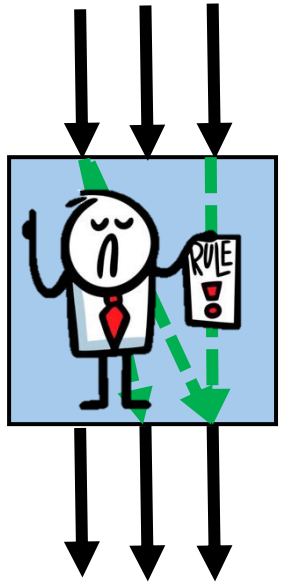
NMIFC

Nonmalleable Information Flow Control

Authors:  Ethan Cecchetti,  Andrew C. Myers,  Owen Arden | Authors Info & Claims

CCS '17: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security • Pages 1875 - 1891
https://doi.org/10.1145/3133956.3134054

$$\Gamma \vdash t : \tau$$





Useful!

Useful!

Viaduct: an extensible, optimizing compiler for secure distributed programs

Authors: Coşku Acay, Rolph Recto, Joshua Gancher,  Andrew C. Myers, Elaine Shi | Authors Info & Claims

PLDI 2021: Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation
Pages 740 - 755 • https://doi.org/10.1145/3453483.3454074

Published: 18 June 2021 Publication History

Check for updates

Useful!

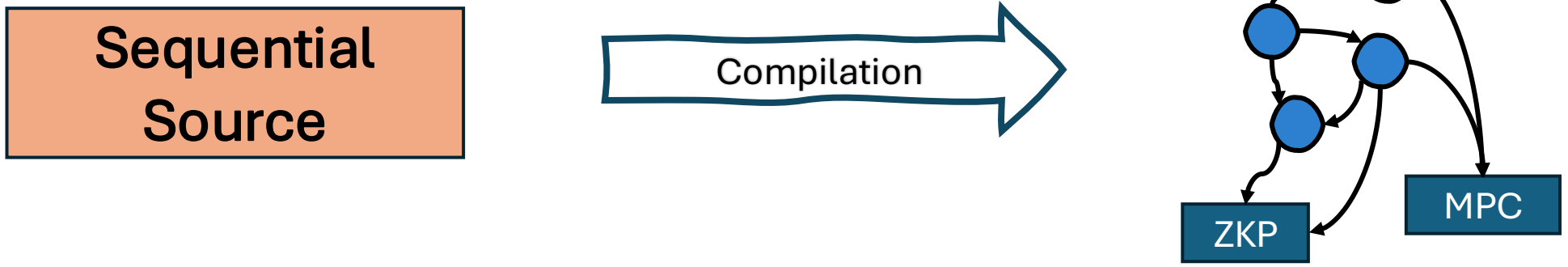
Viaduct: an extensible, optimizing compiler for secure distributed programs

Authors: Coşku Acay, Rolph Recto, Joshua Gancher, Andrew C. Myers, Elaine Shi | Authors Info & Claims

PLDI 2021: Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation
Pages 740 - 755 • https://doi.org/10.1145/3453483.3454074

Published: 18 June 2021 Publication History

Check for updates



Useful!

Viaduct: an extensible, optimizing compiler for secure distributed programs

Authors: Coşku Acay, Rolph Recto, Joshua Gancher,  Andrew C. Myers, Elaine Shi | Authors Info & Claims

PLDI 2021: Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation
Pages 740 - 755 • https://doi.org/10.1145/3453483.3454074

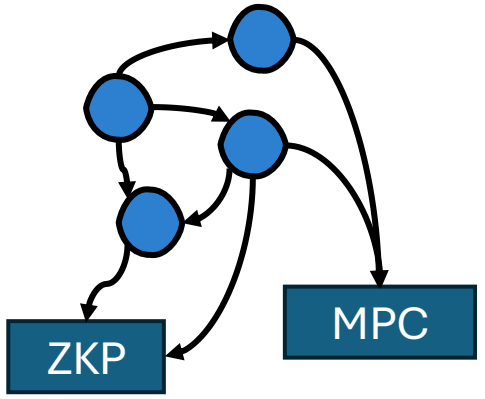
Published: 18 June 2021 Publication History

Check for updates

Sequential Source

Compilation

$\geq_{Real-Ideal}$



Useful!

Viaduct: an extensible, optimizing compiler for secure distributed programs

Authors: Coşku Acay, Rolph Recto, Joshua Gancher, Andrew C. Myers, Elaine Shi | Authors Info & Claims

PLDI 2021: Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation
Pages 740 - 755 • https://doi.org/10.1145/3453483.3454074

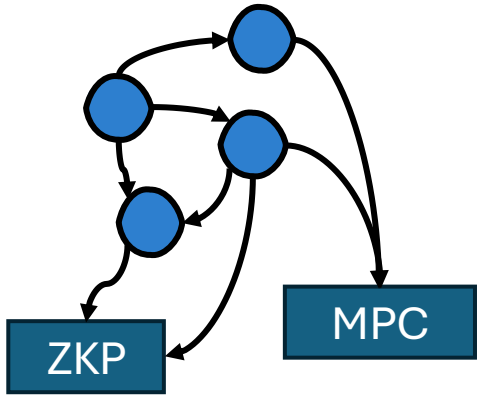
Published: 18 June 2021 Publication History

Check for updates

Sequential Source

Compilation

$\geq_{Real-Ideal}$



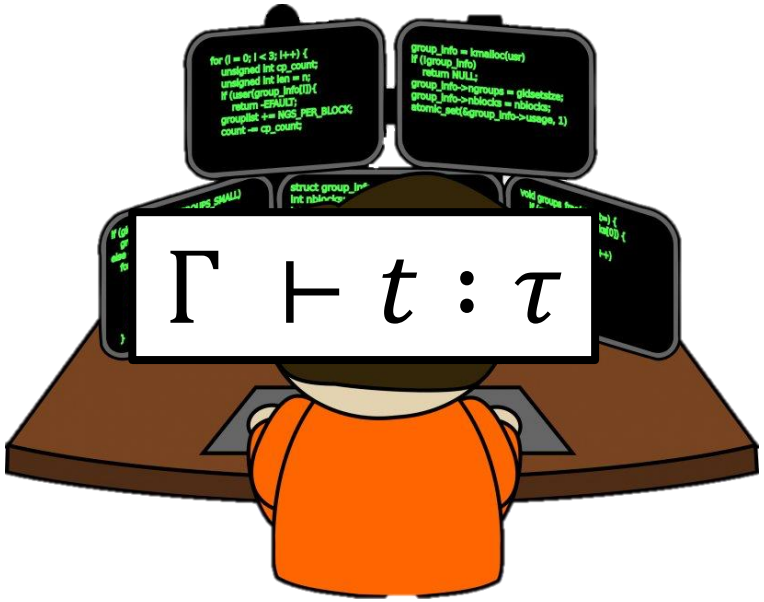
$\Gamma \vdash t : \tau$!!

A Little Confusing

A Little Confusing

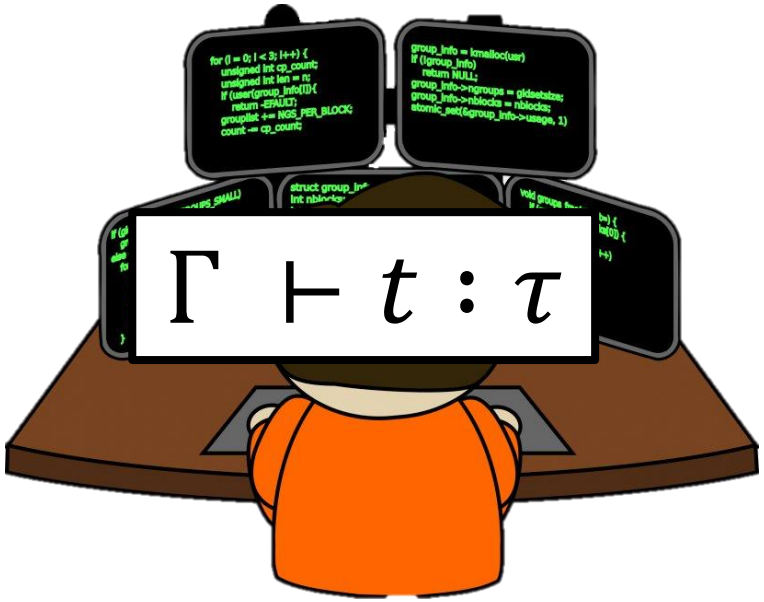


A Little Confusing



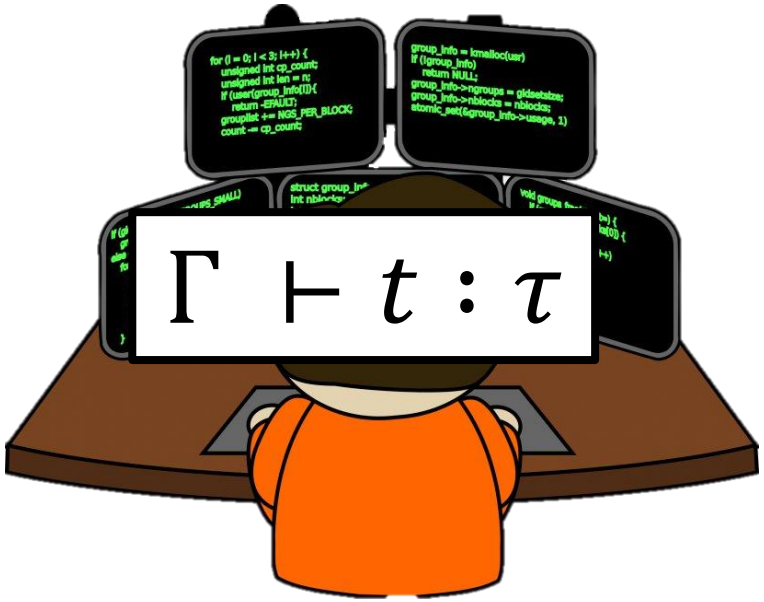
A Little Confusing

Noninterference



NMIFC

A Little Confusing



Noninterference

```
let a = input from Alice
let b = input from Bob

let sum = a + b

output sum to Alice, Bob
```

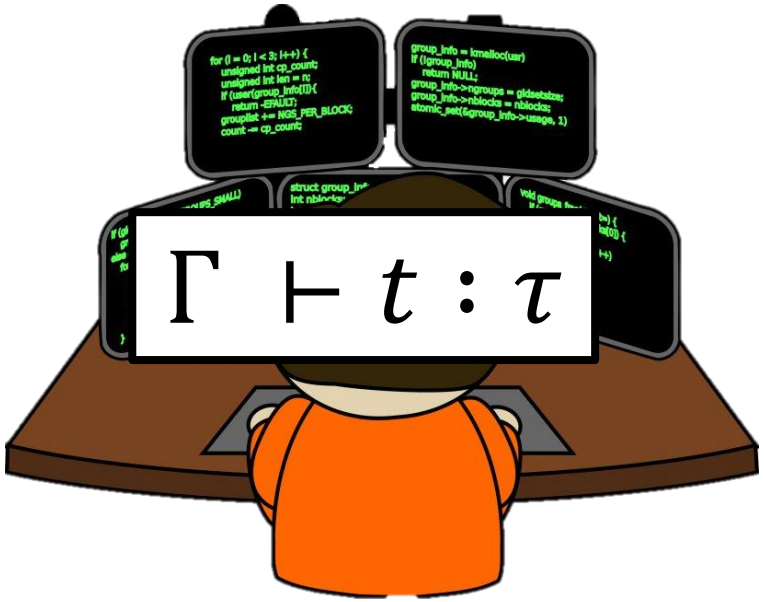
NMIFC

```
let a = input from Alice
let b = input from Bob

let sum = a + b

output sum to Alice, Bob
```

A Little Confusing



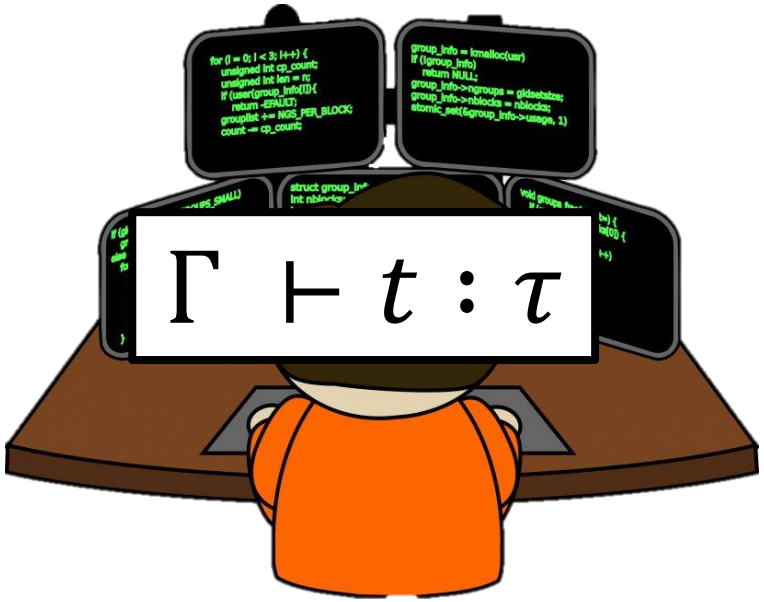
Noninterference



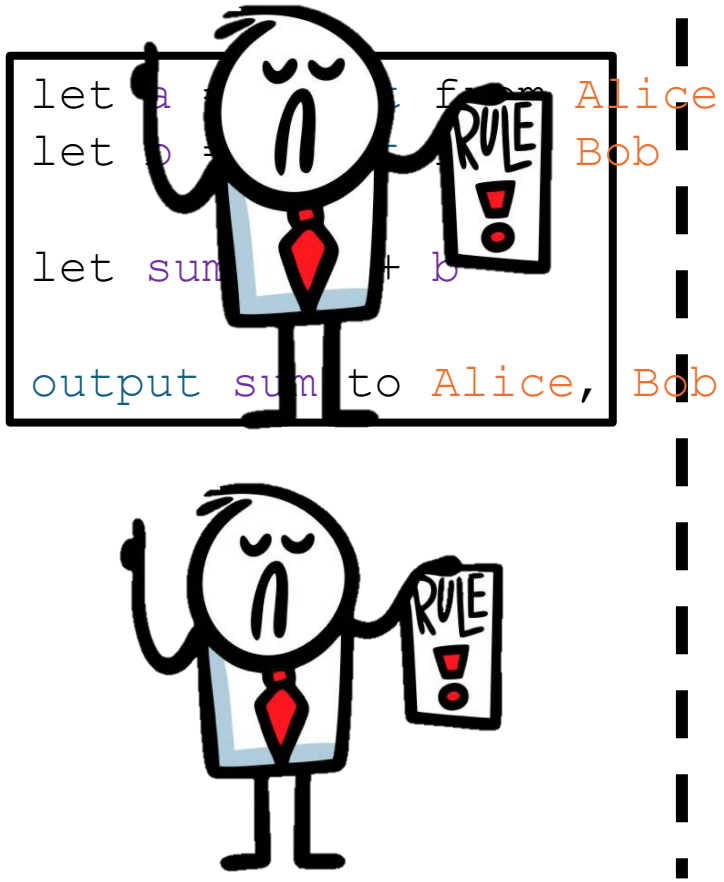
NMIFC



A Little Confusing



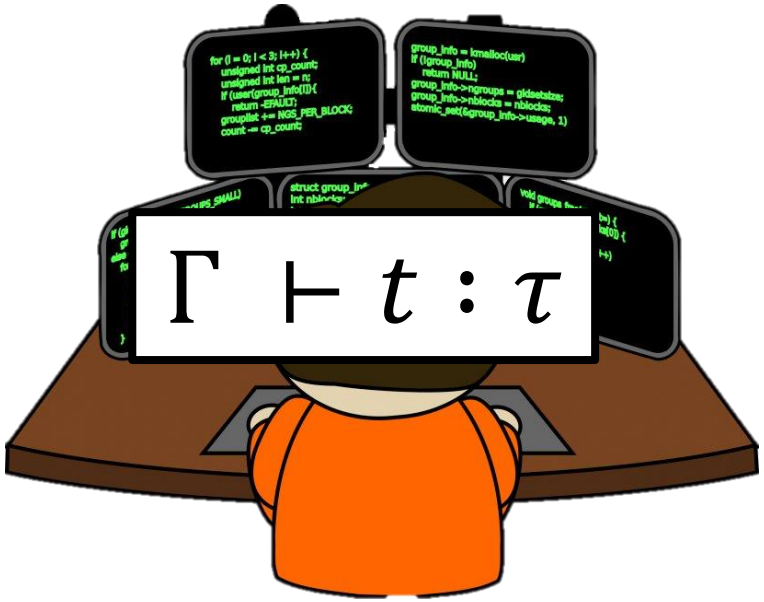
Noninterference



NMIFC



A Little Confusing



Noninterference

```
let a = input from Alice
let b = input from Bob

let sum = a + b

output sum to Alice, Bob
```

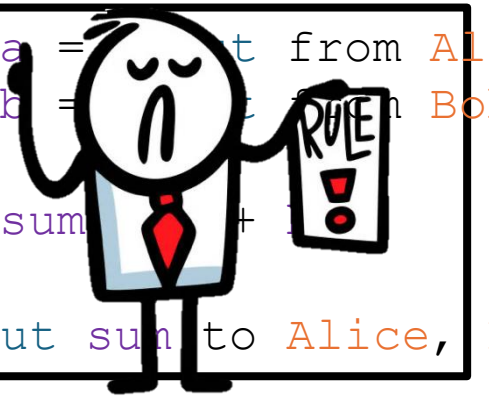


NMIFC

```
let a = input from Alice
let b = input from Bob

let sum = a + b

output sum to Alice, Bob
```

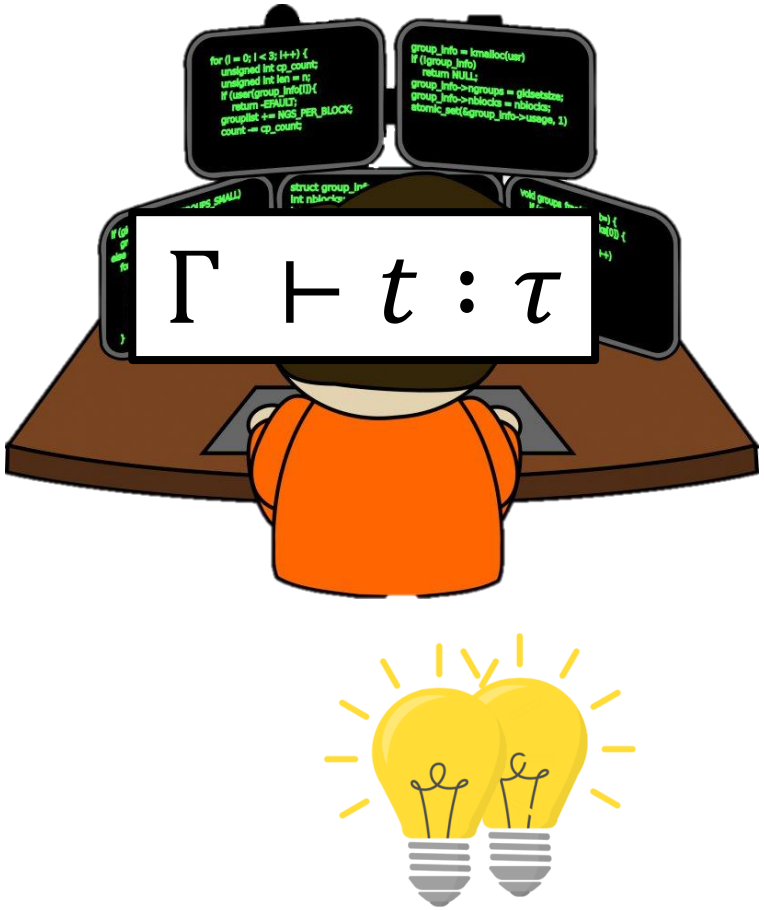



```
let a = endorse(input from Alice
let b = endorse(input from Bob,

let sum = declassify(a > b,⊥)

output sum to Alice, Bob
```

A Little Confusing



Noninterference

```
let a = input from Alice
let b = input from Bob

let sum = a + b

output sum to Alice, Bob
```



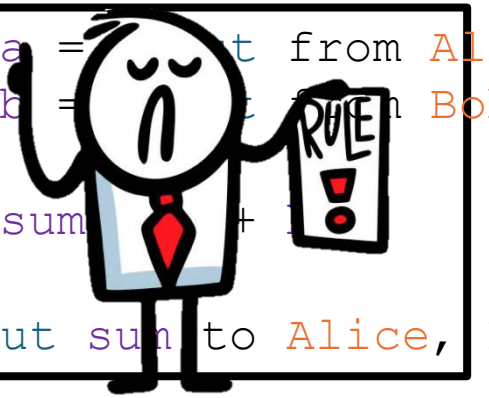


NMIFC

```
let a = input from Alice
let b = input from Bob

let sum = a + b

output sum to Alice, Bob
```



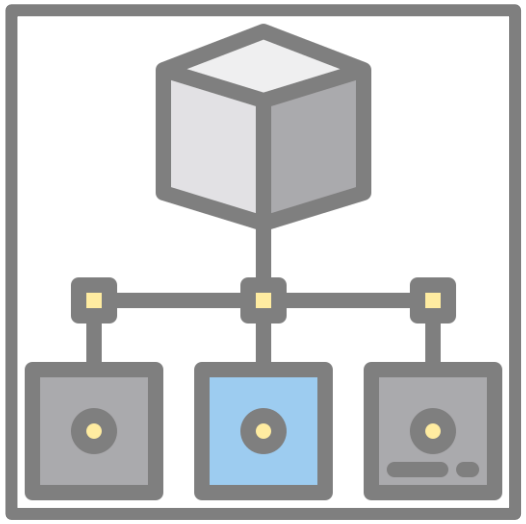

```
let a = endorse(input from Alice
let b = endorse(input from Bob,

let sum = declassify(a > b,⊥)

output sum to Alice, Bob
```

Research Directions



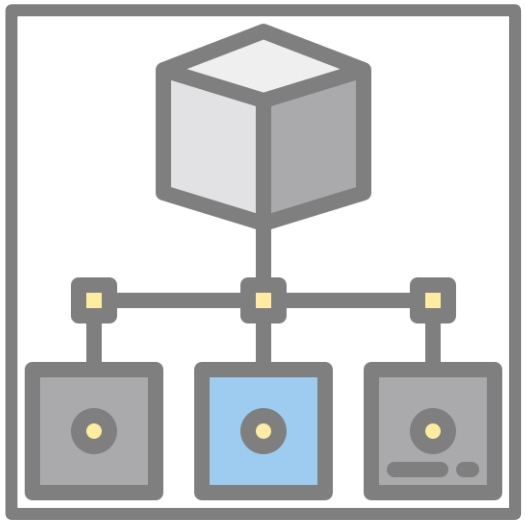






Research Directions









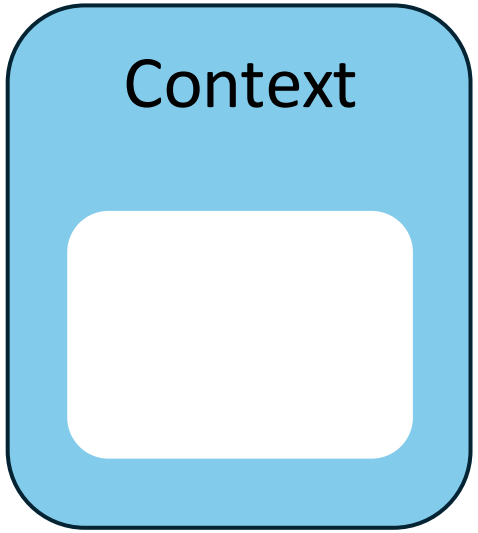
The Question of Composition

The Question of Composition

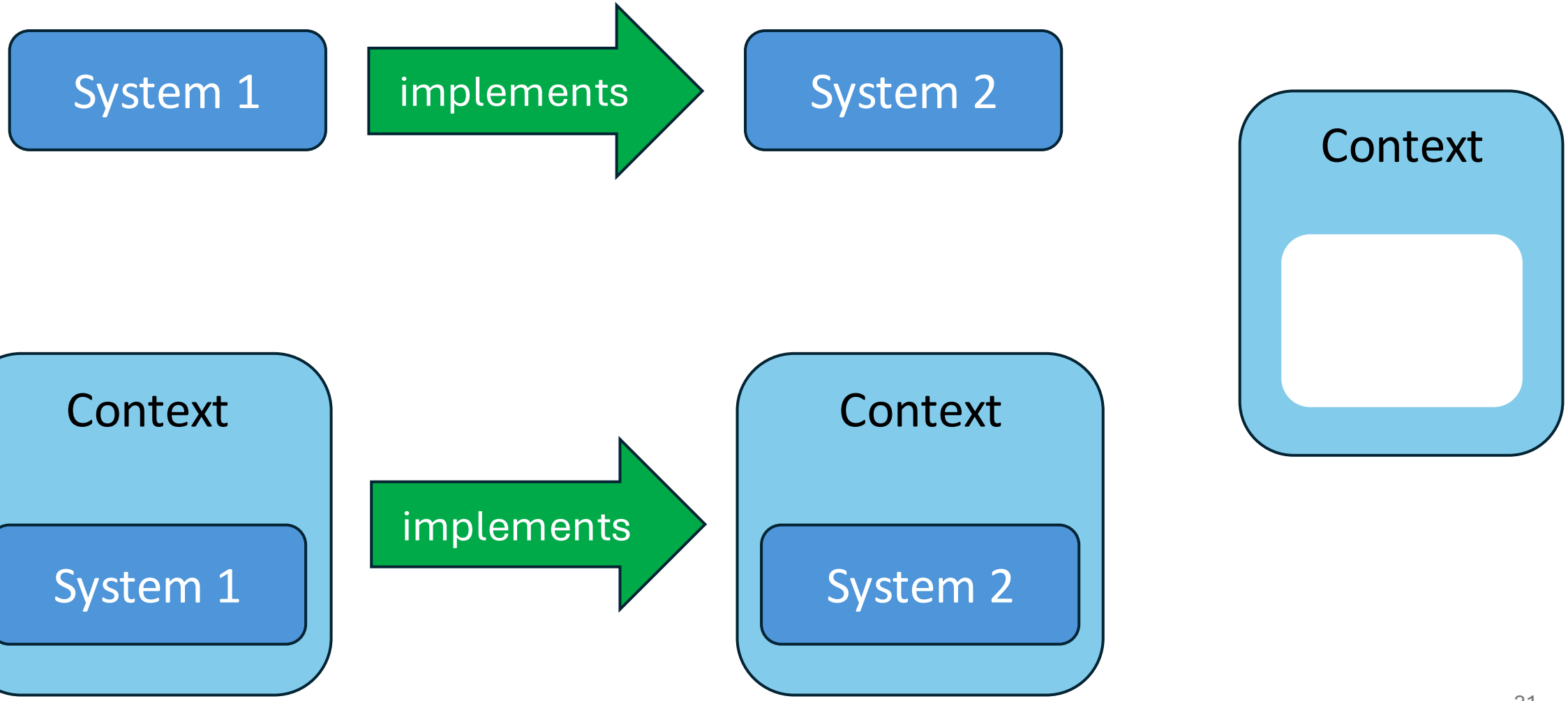


The Question of Composition



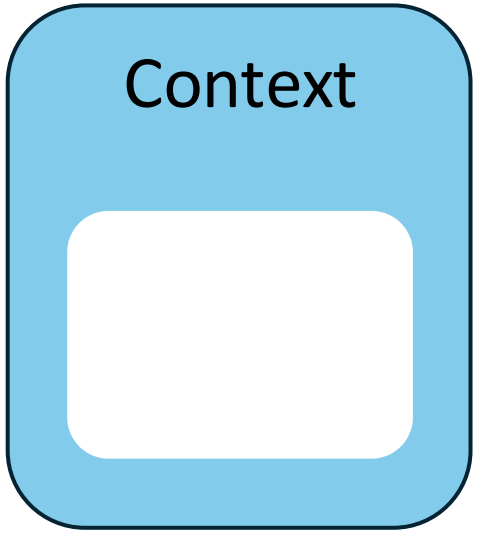


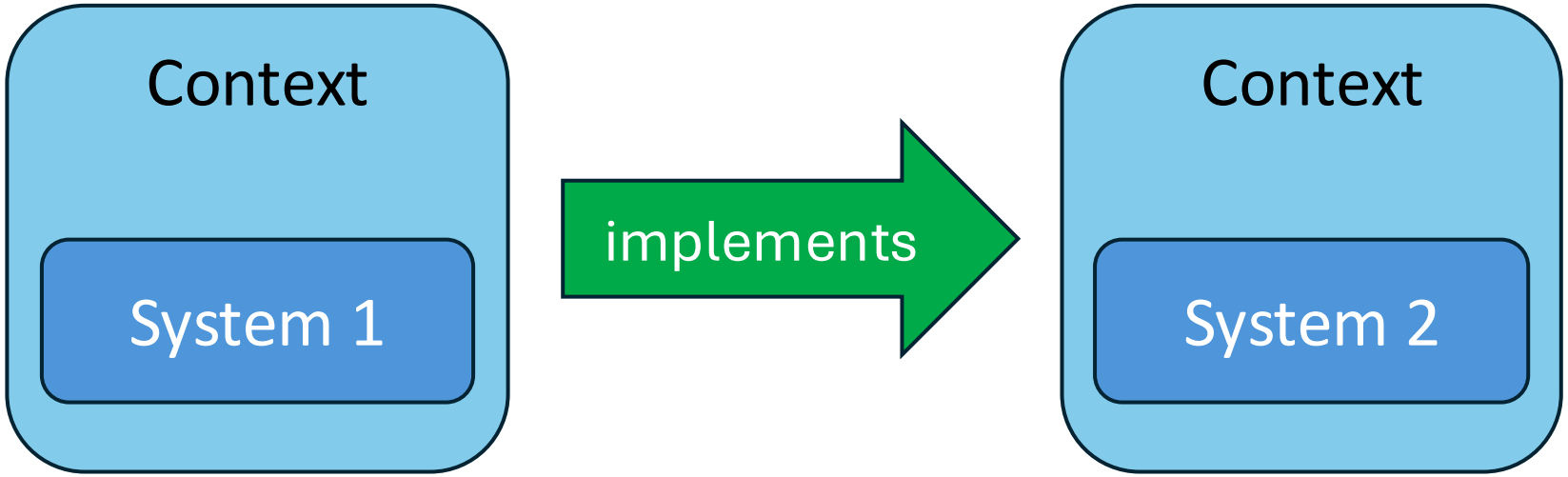
The Question of Composition



The Question of Composition







Not obvious!

Failure of Compositional Security

Failure of Compositional Security





Failure of Compositional Security



Key Exchange



Failure of Compositional Security

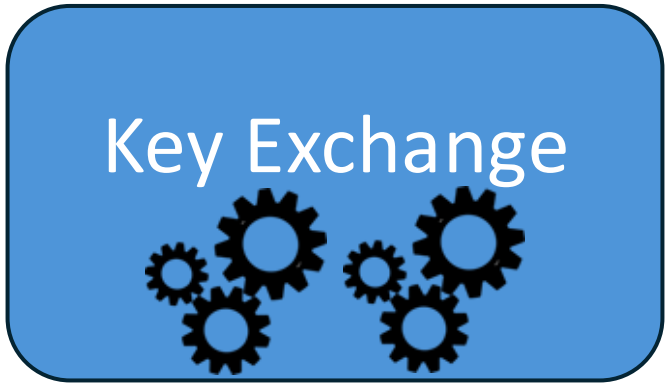


Key Exchange



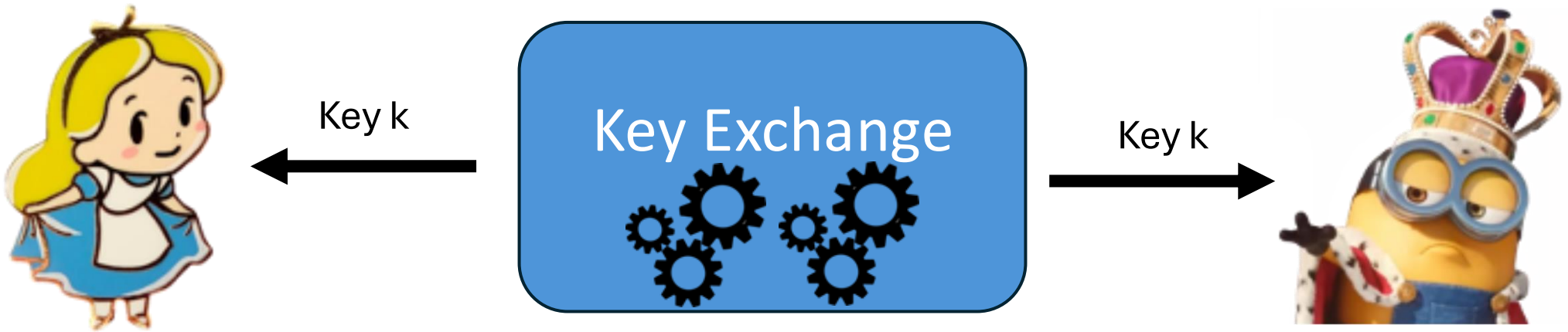
Failure of Compositional Security



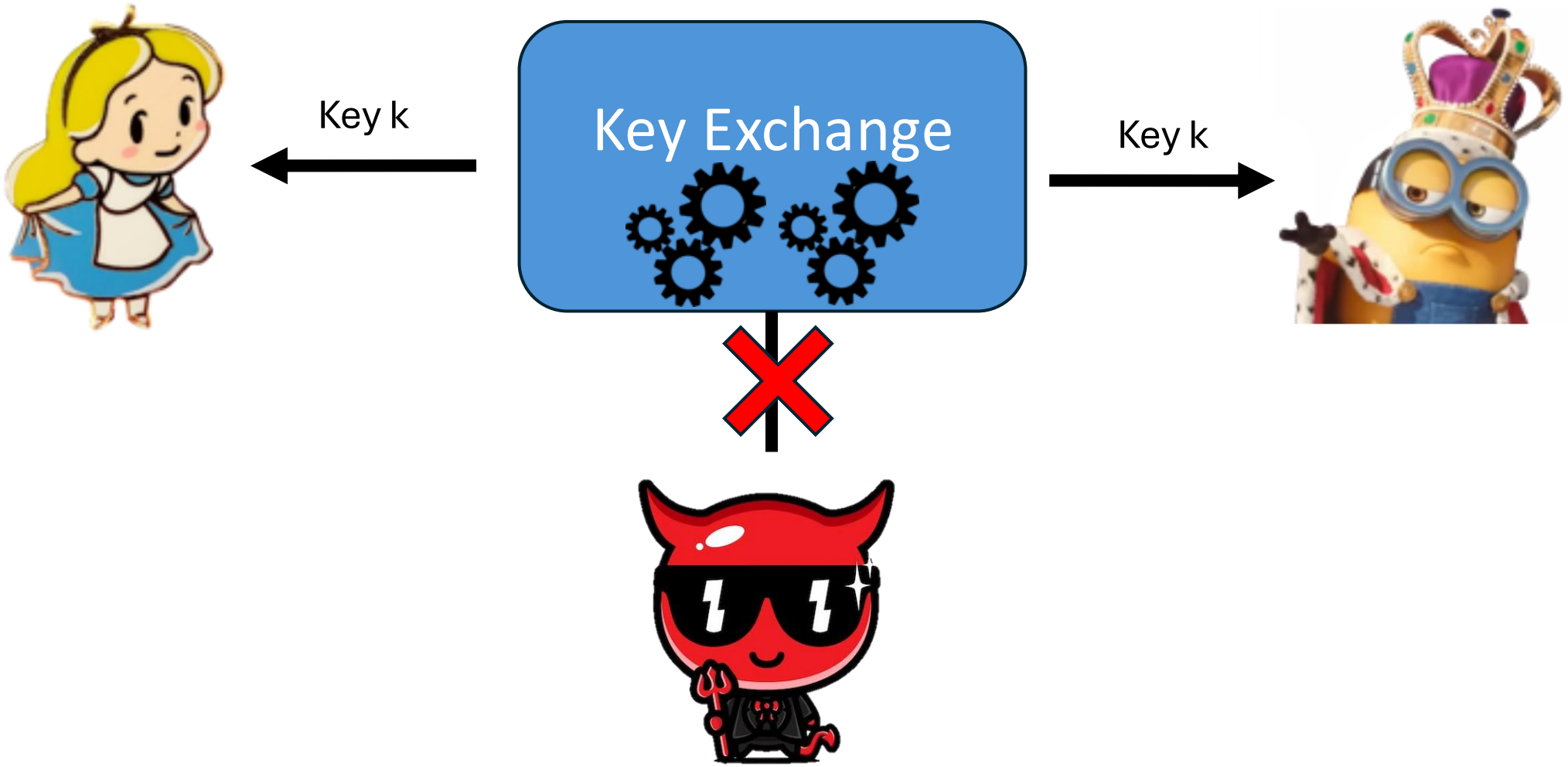




Failure of Compositional Security

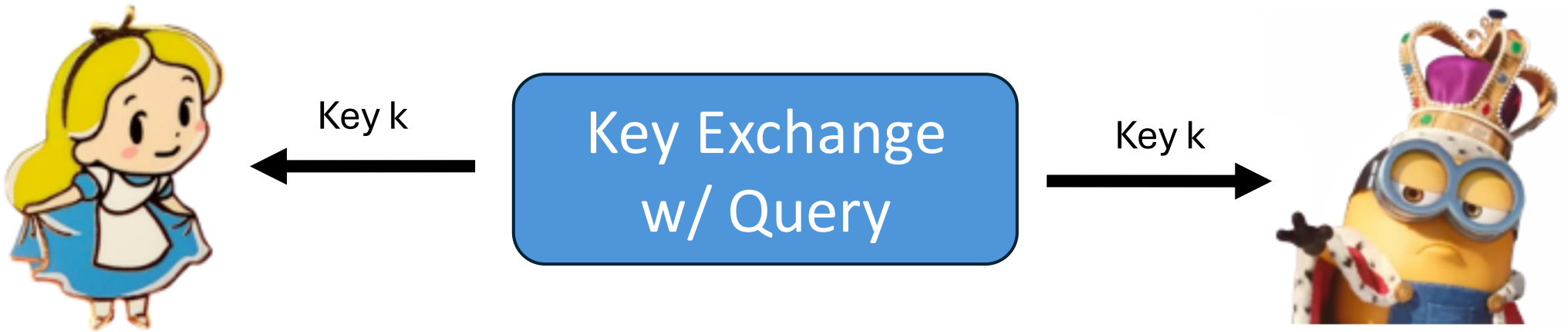


Failure of Compositional Security



Key Exchange with Query

Key Exchange with Query



Key Exchange with Query



Key k

Key Exchange w/ Query

Key k





Key Exchange with Query



Key k

Key Exchange w/ Query

Key k



Is k' the key?



Key Exchange with Query



Key k

Key Exchange w/ Query

Key k



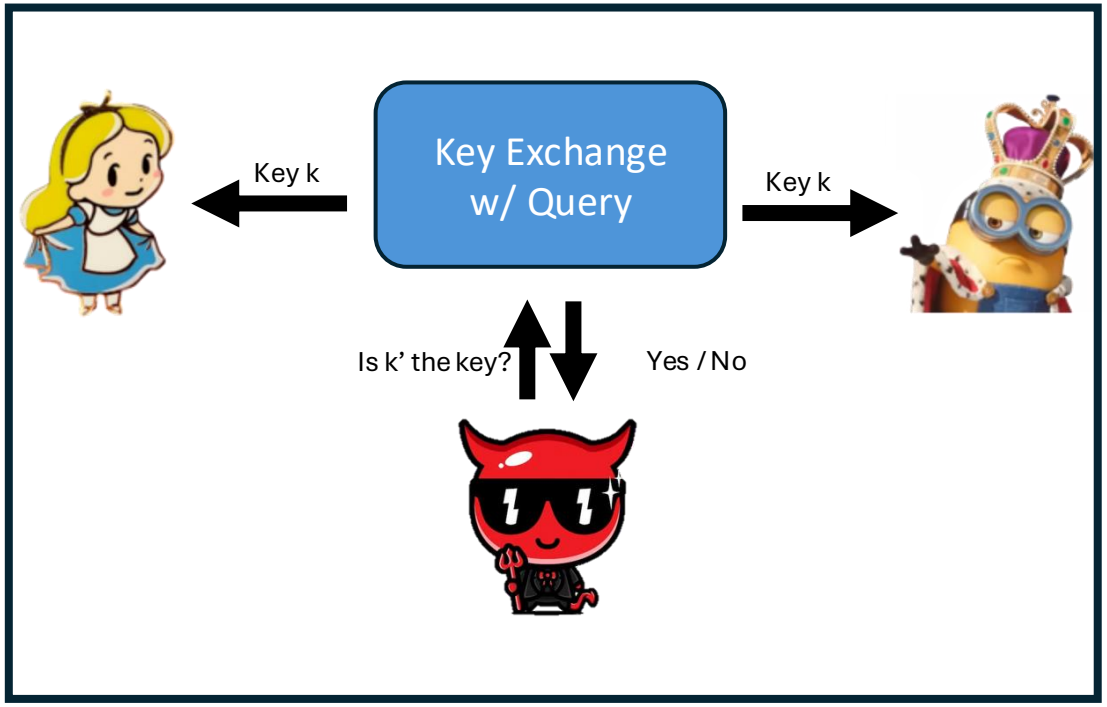
Is k' the key?

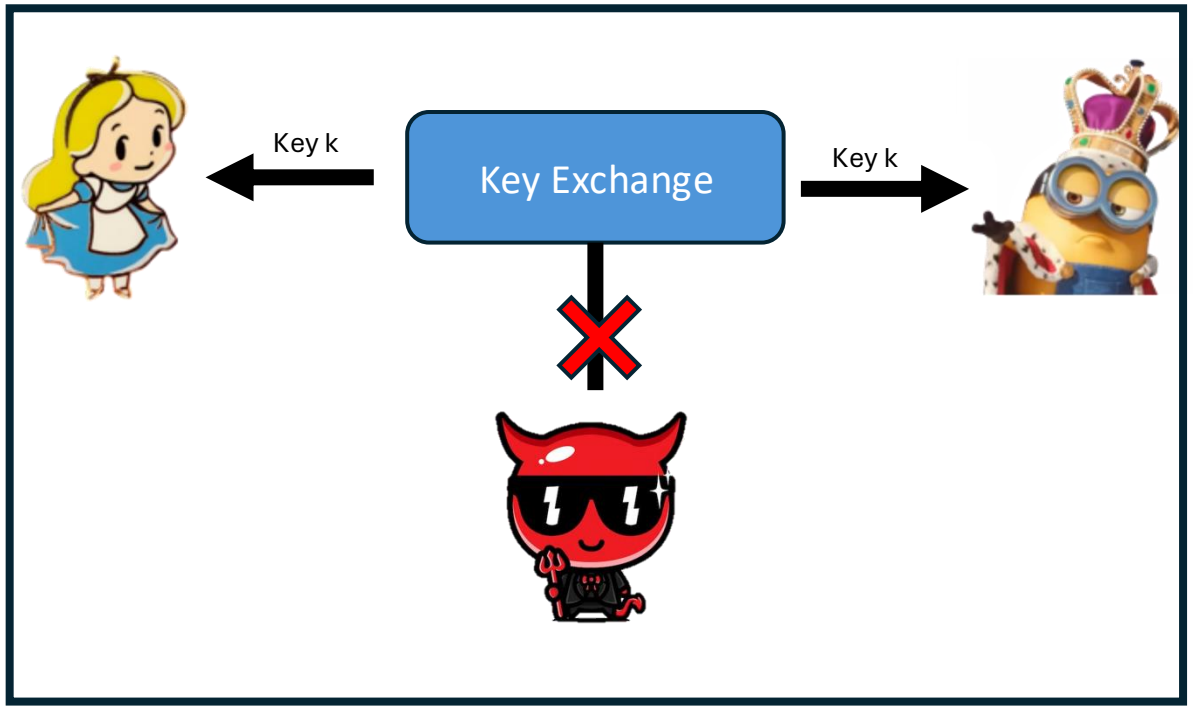
Yes / No



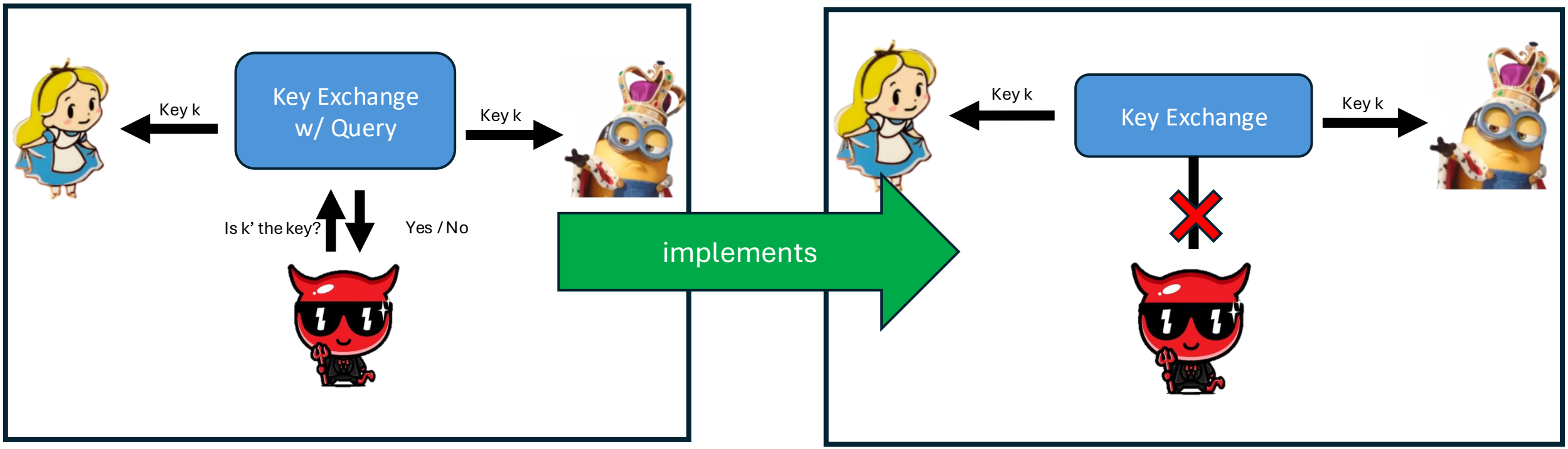
Real-Ideal implementation*

Real-Ideal implementation*

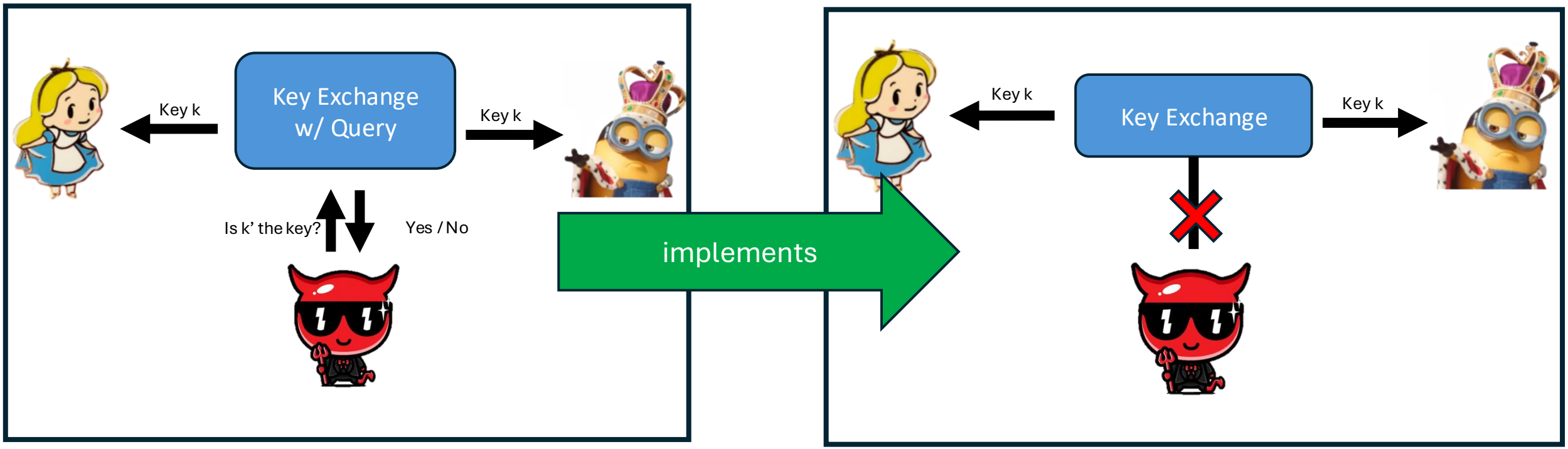




Real-Ideal implementation*



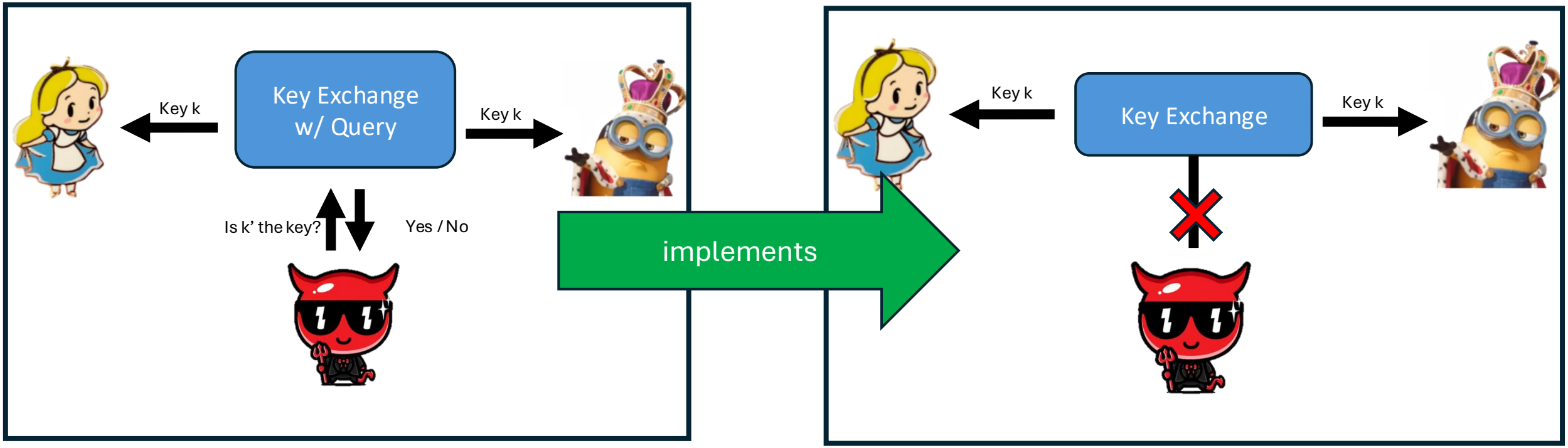
Real-Ideal implementation*



* In *Standalone Model* of Real-Ideal paradigm

Real-Ideal implementation*

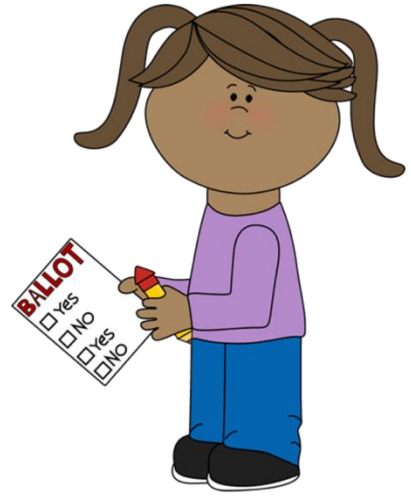




* In *Standalone Model* of Real-Ideal paradigm

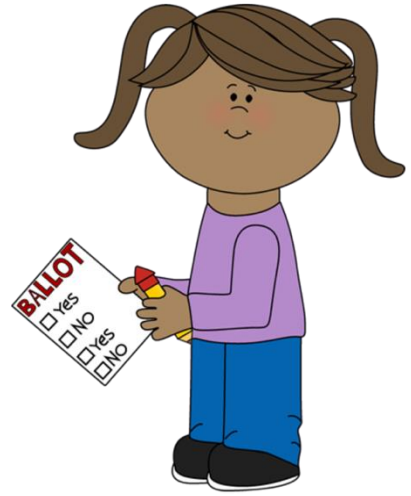
Sending a Private Vote

Sending a Private Vote



Sending a Private Vote





Sending a Private Vote



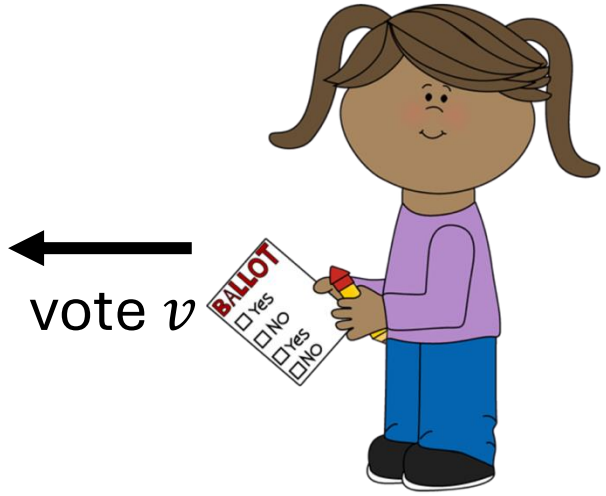




Sending a Private Vote



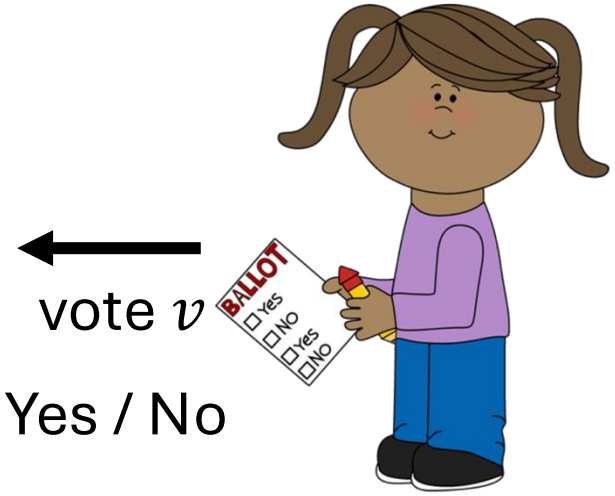




Sending a Private Vote



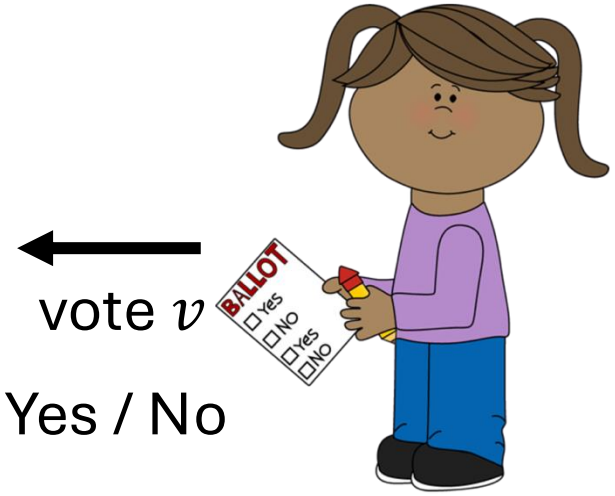




Sending a Private Vote



Public Channel



Sending a Private Vote

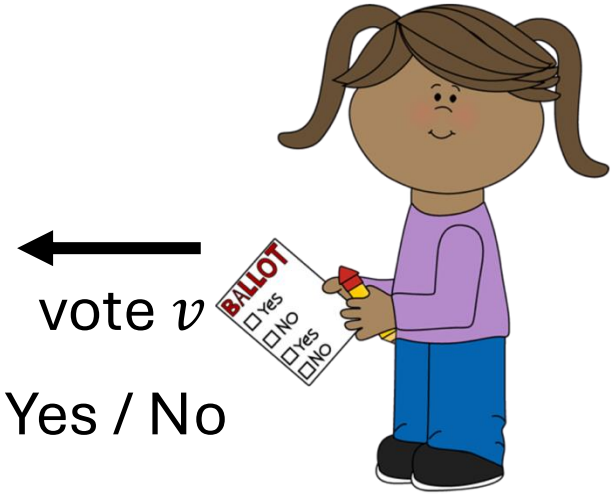


Public Channel



vote v

Yes / No



Sending a Private Vote



Key Exchange

Public Channel



vote v

Yes / No



Sending a Private Vote



Key Exchange

key k

Public Channel

vote v

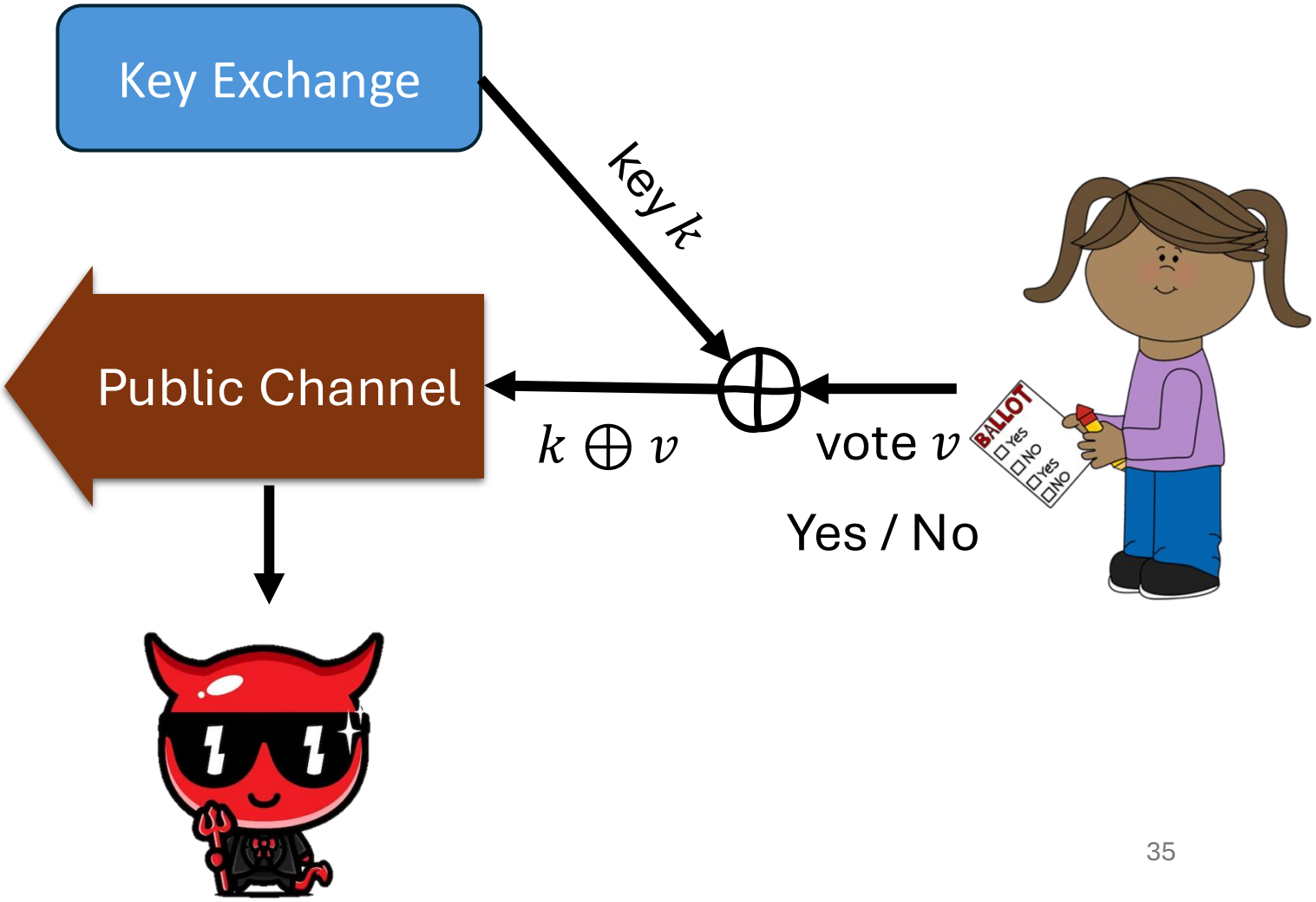
Yes / No



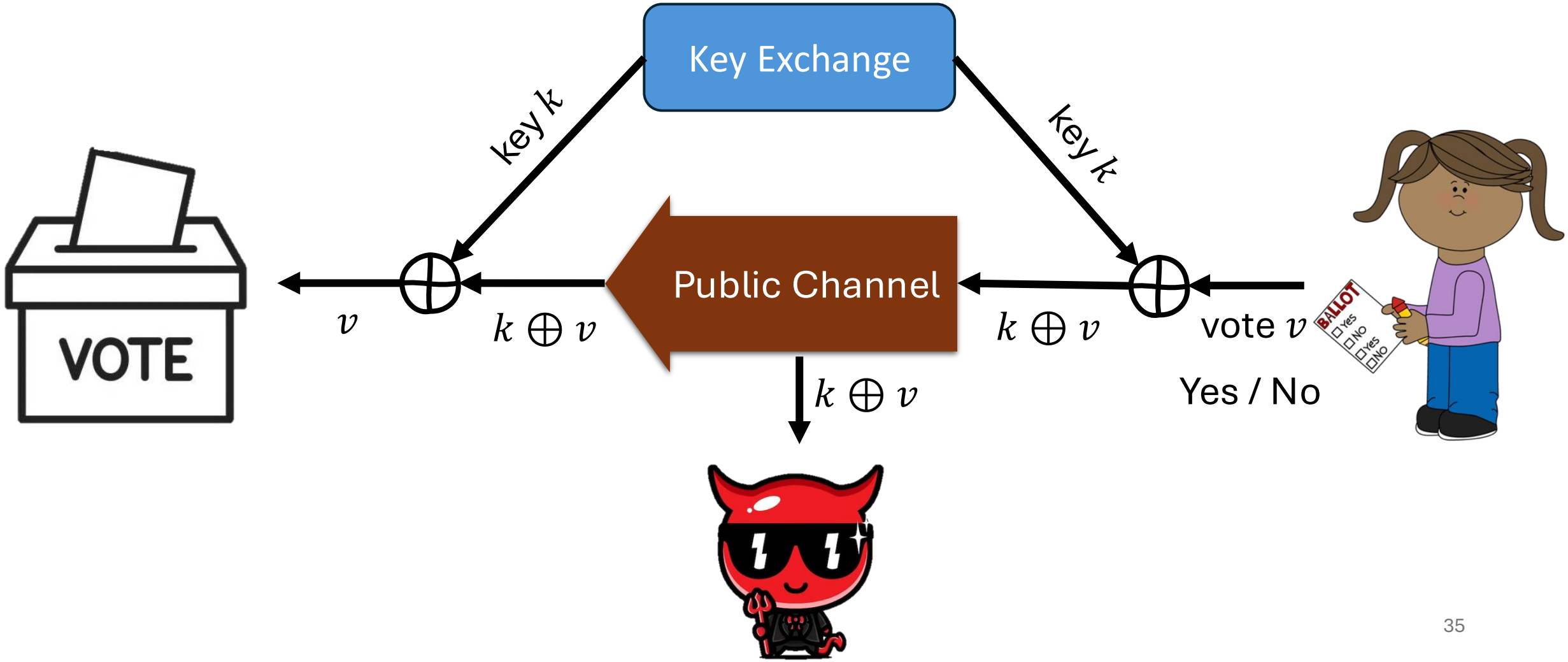


Sending a Private Vote

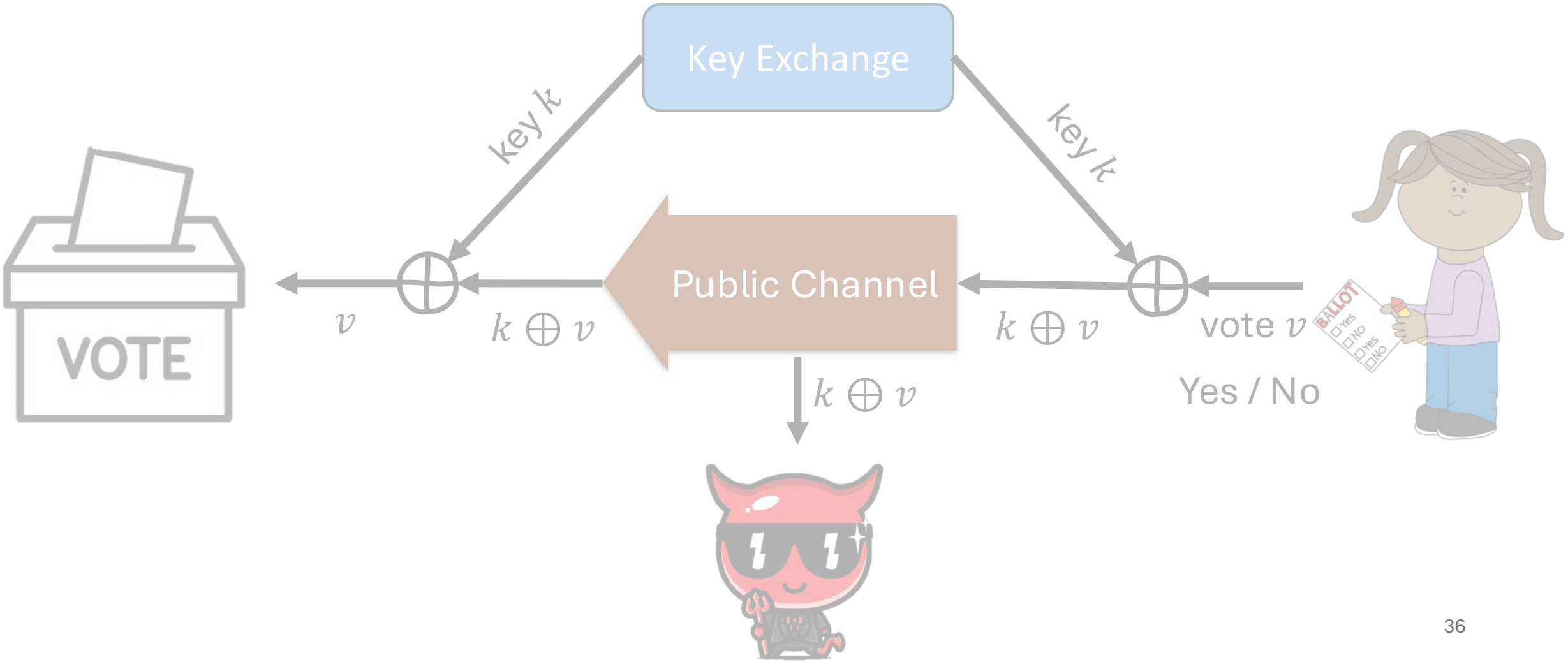




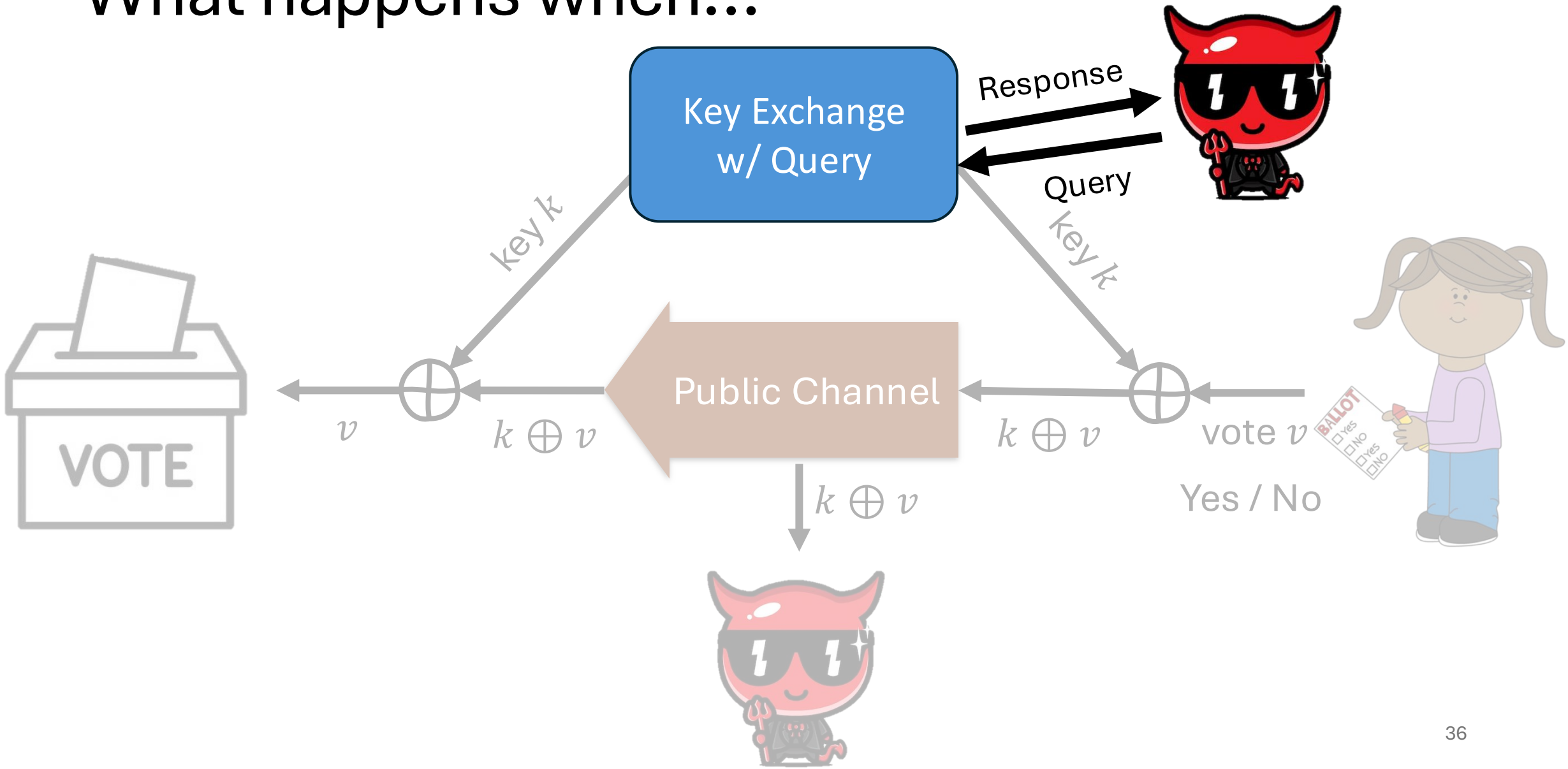
Sending a Private Vote



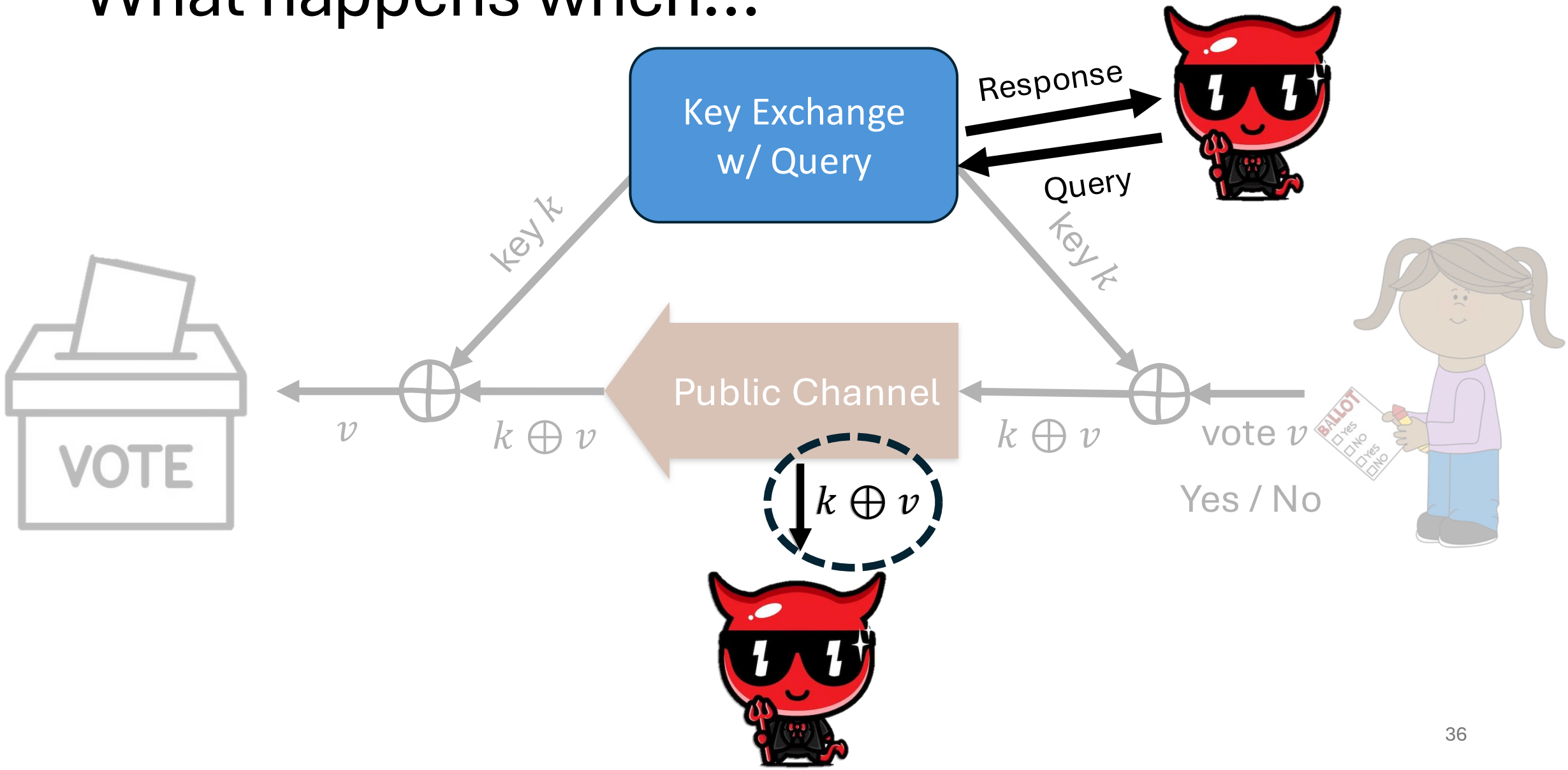
What happens when…



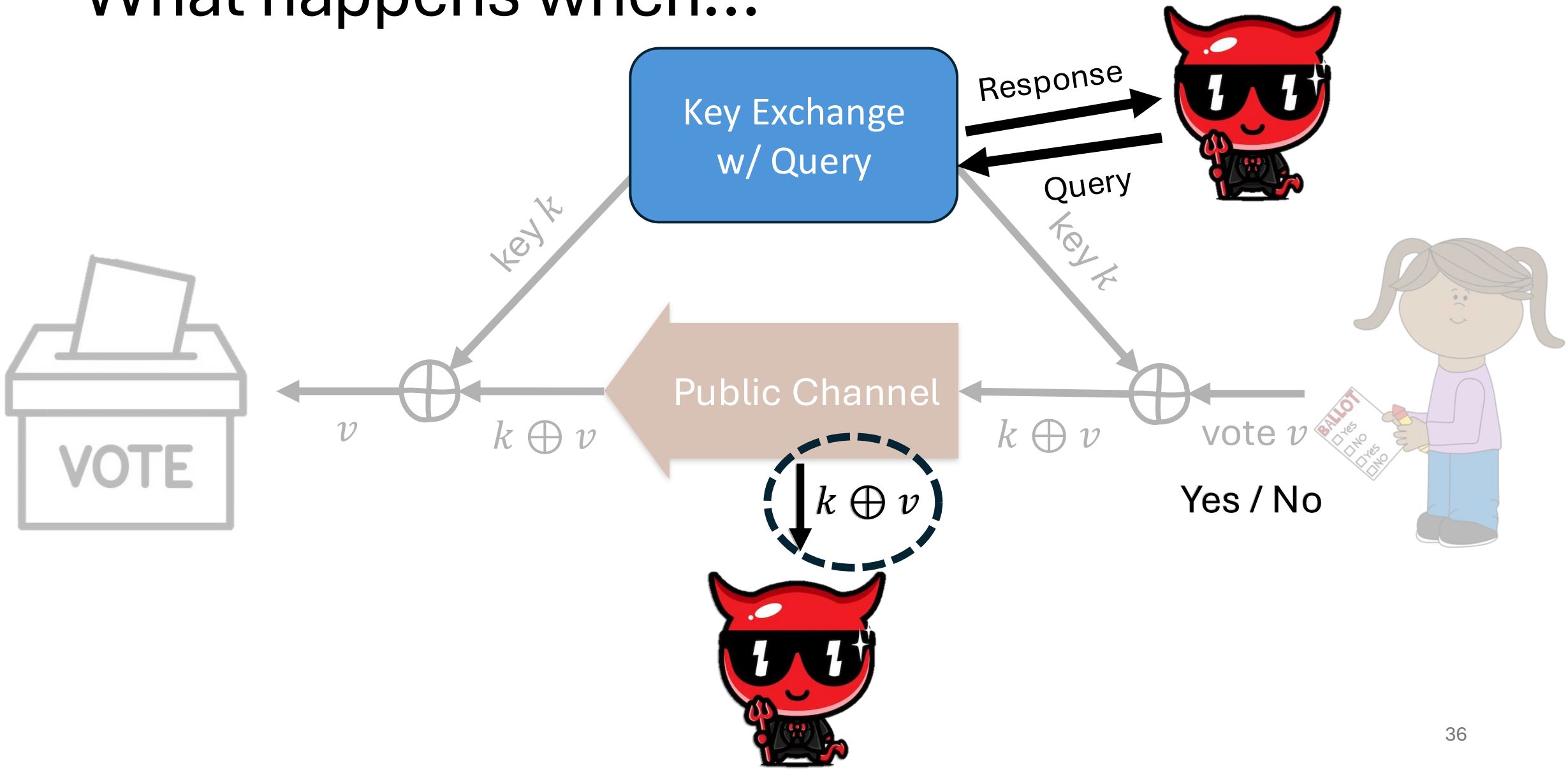
What happens when…



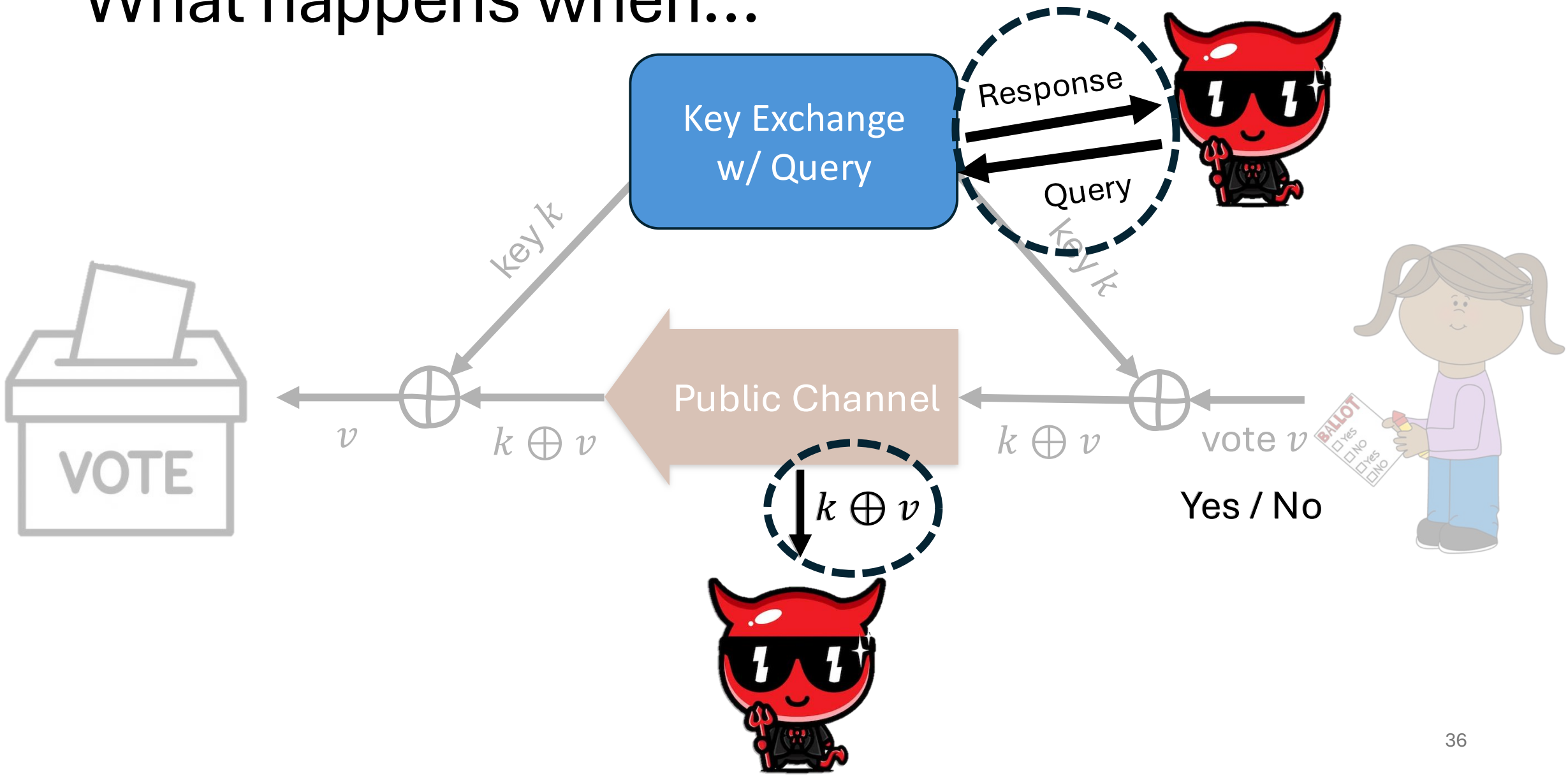
What happens when…



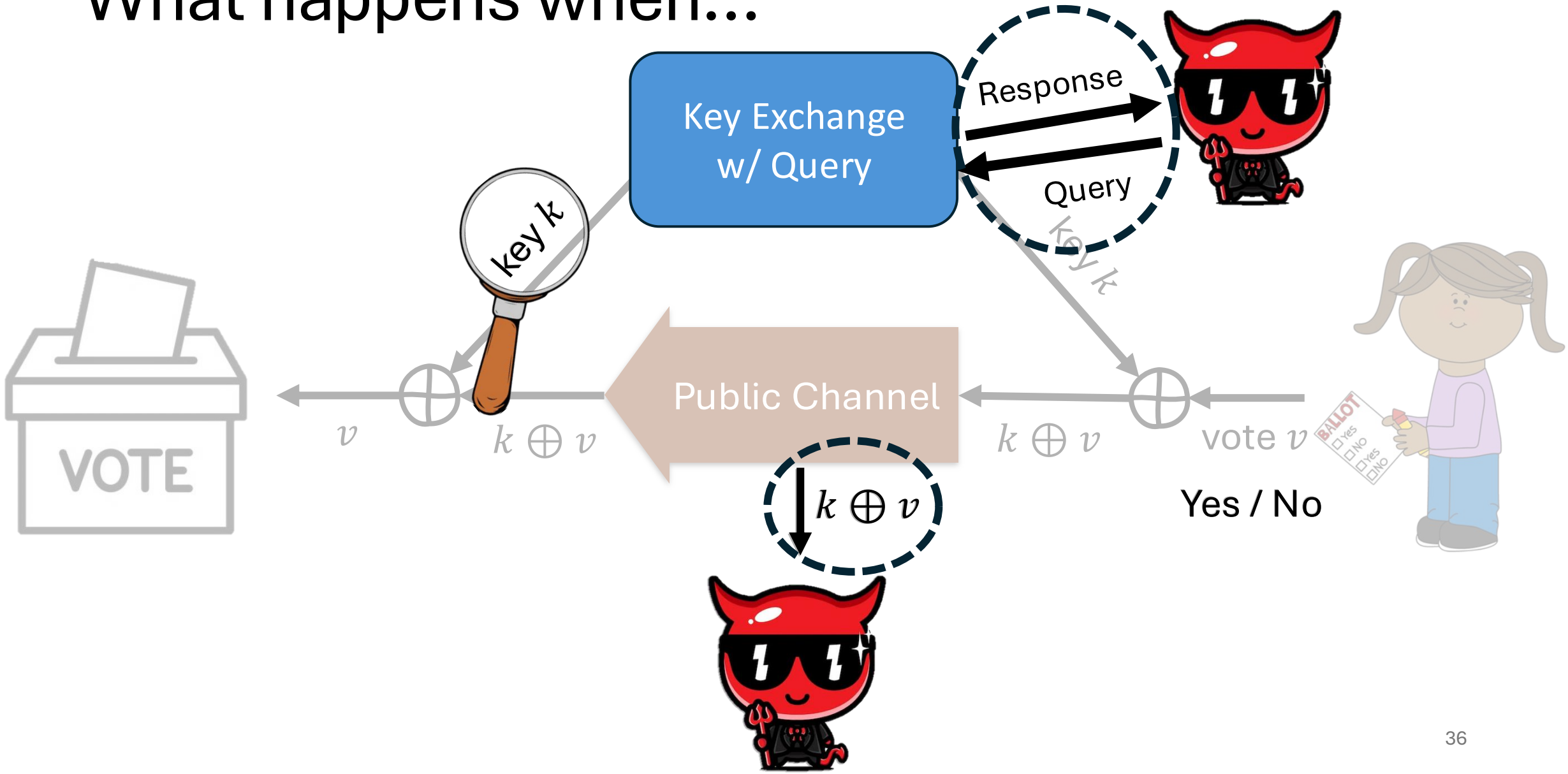
What happens when…



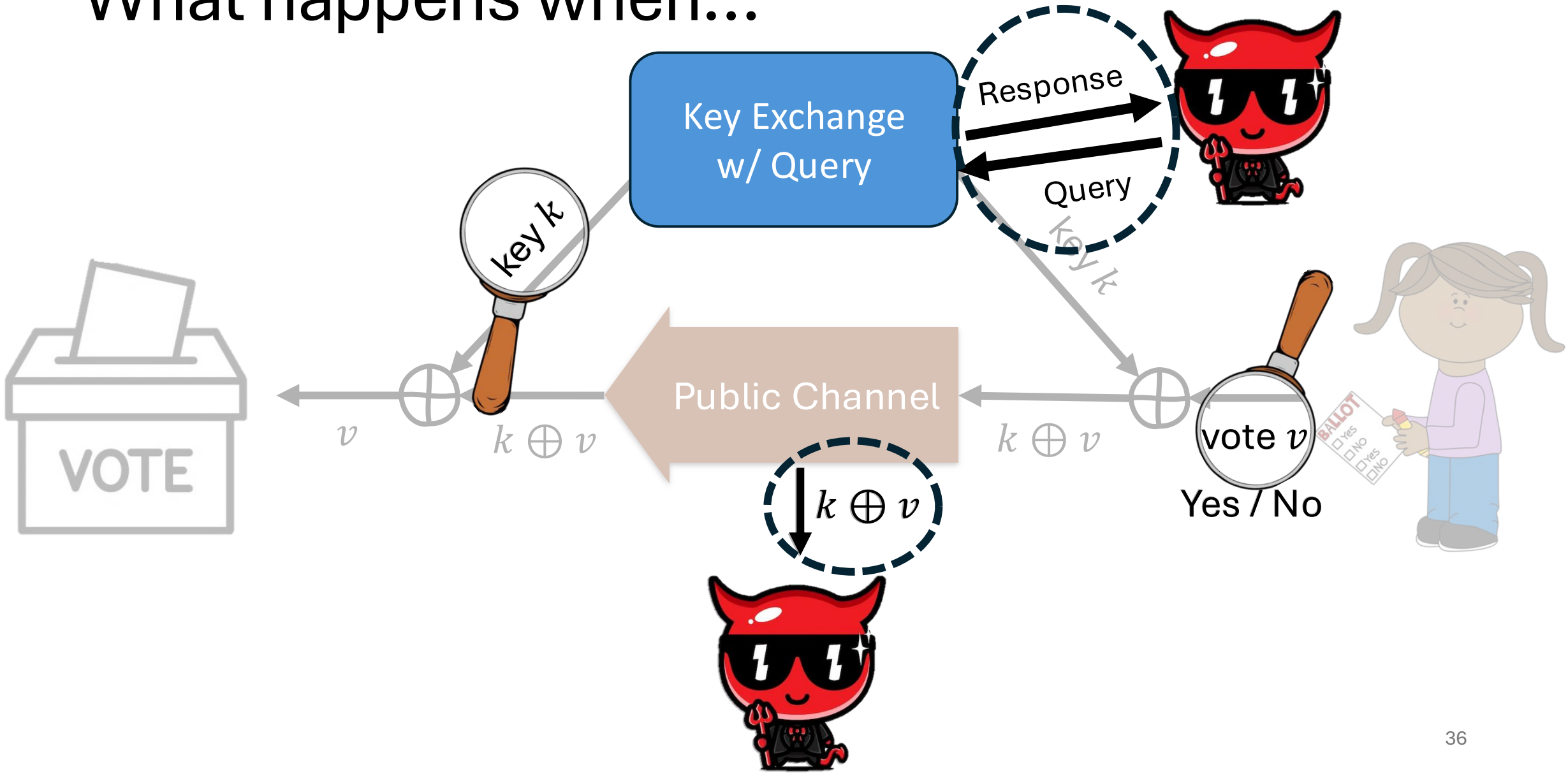
What happens when…



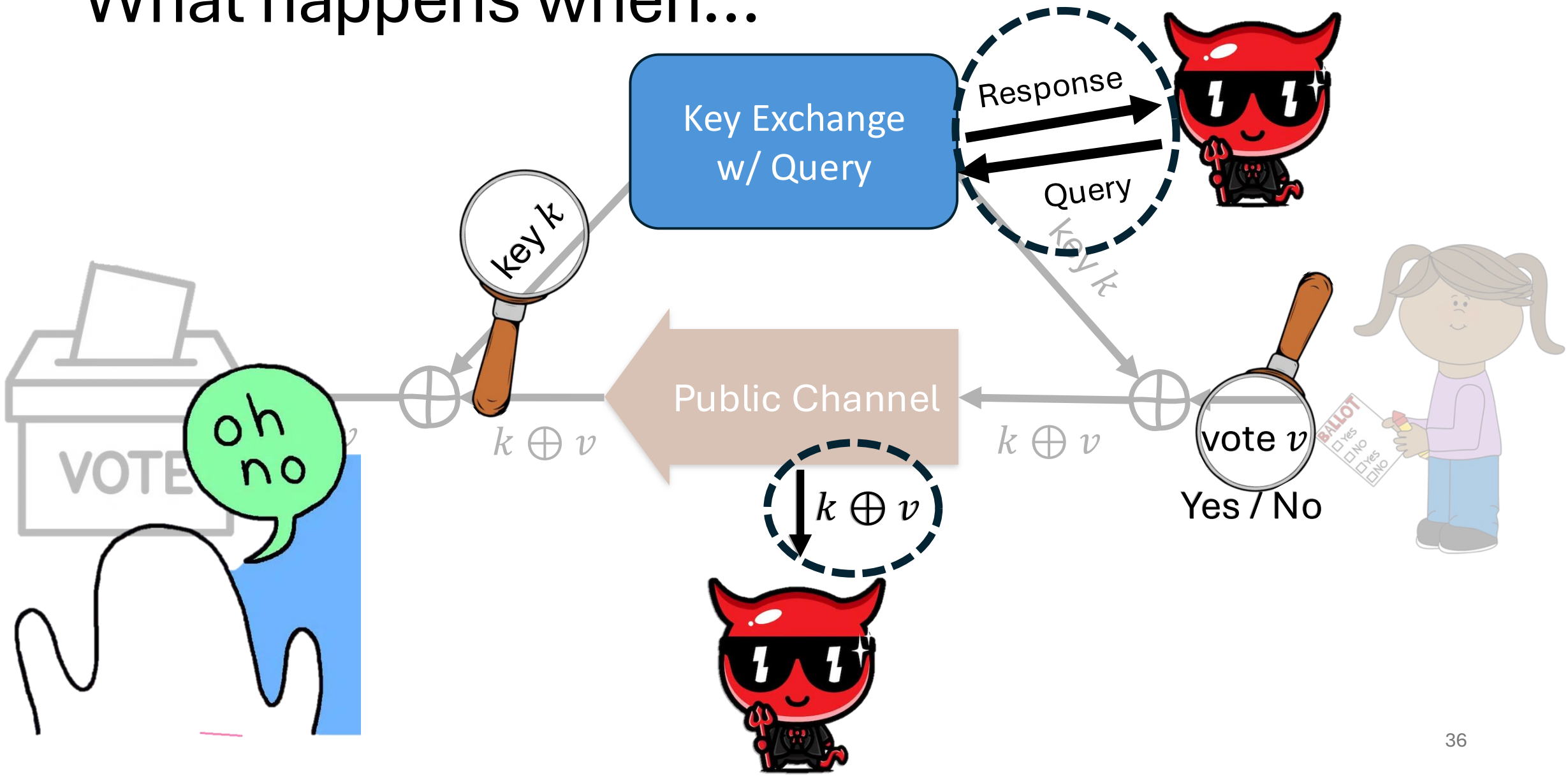
What happens when…



What happens when…

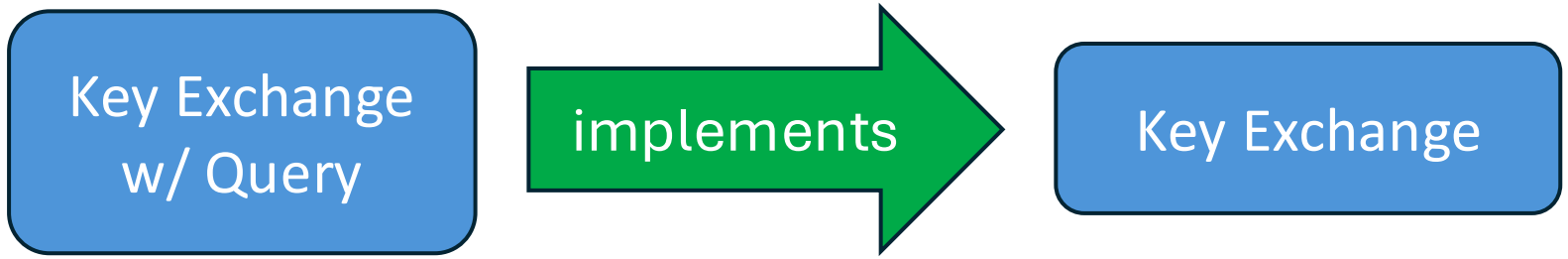


What happens when…

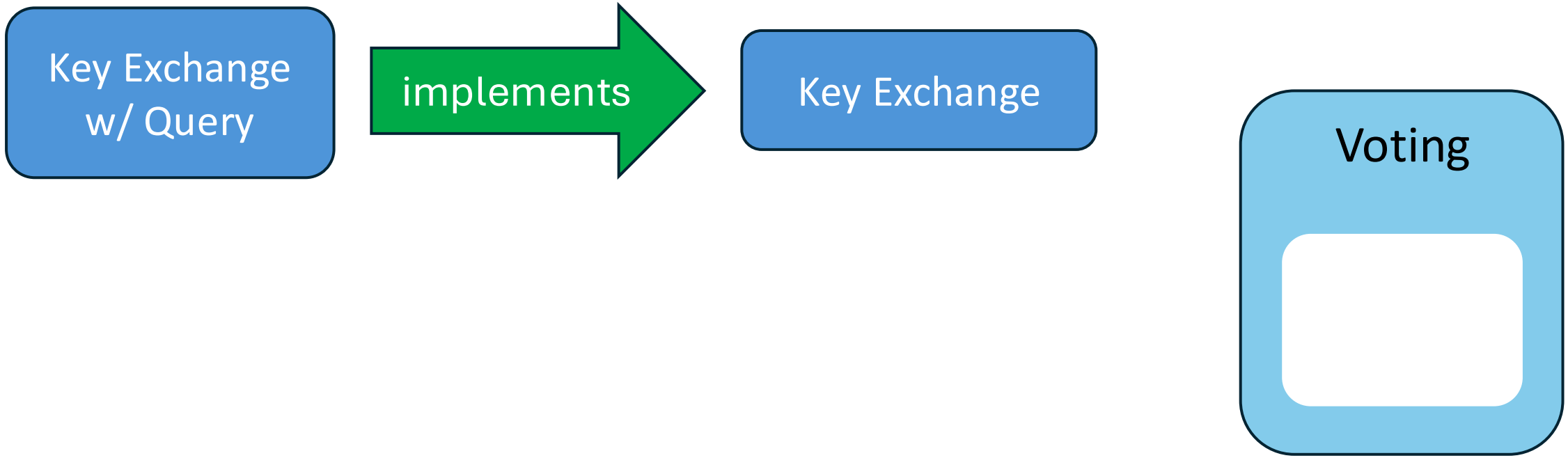


Composition Failure

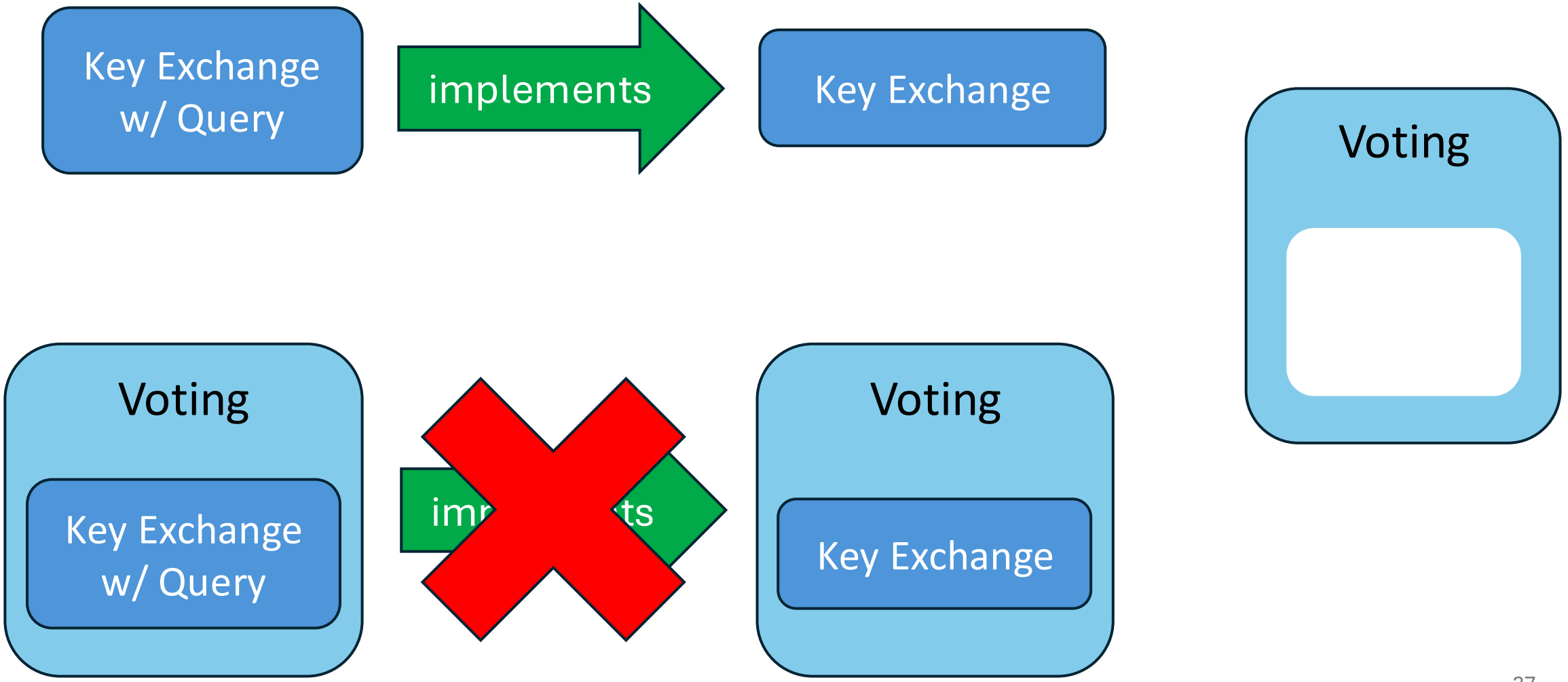
Composition Failure



Composition Failure

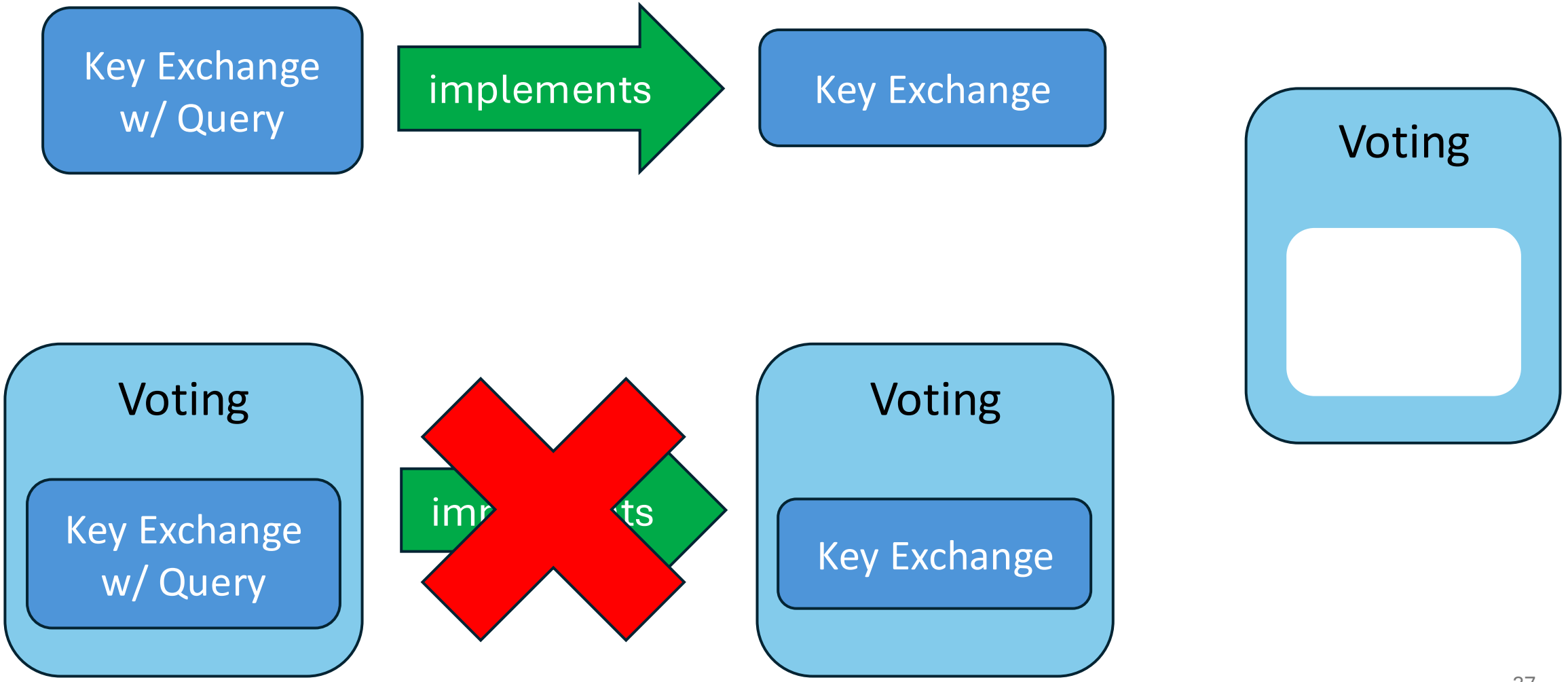


Composition Failure



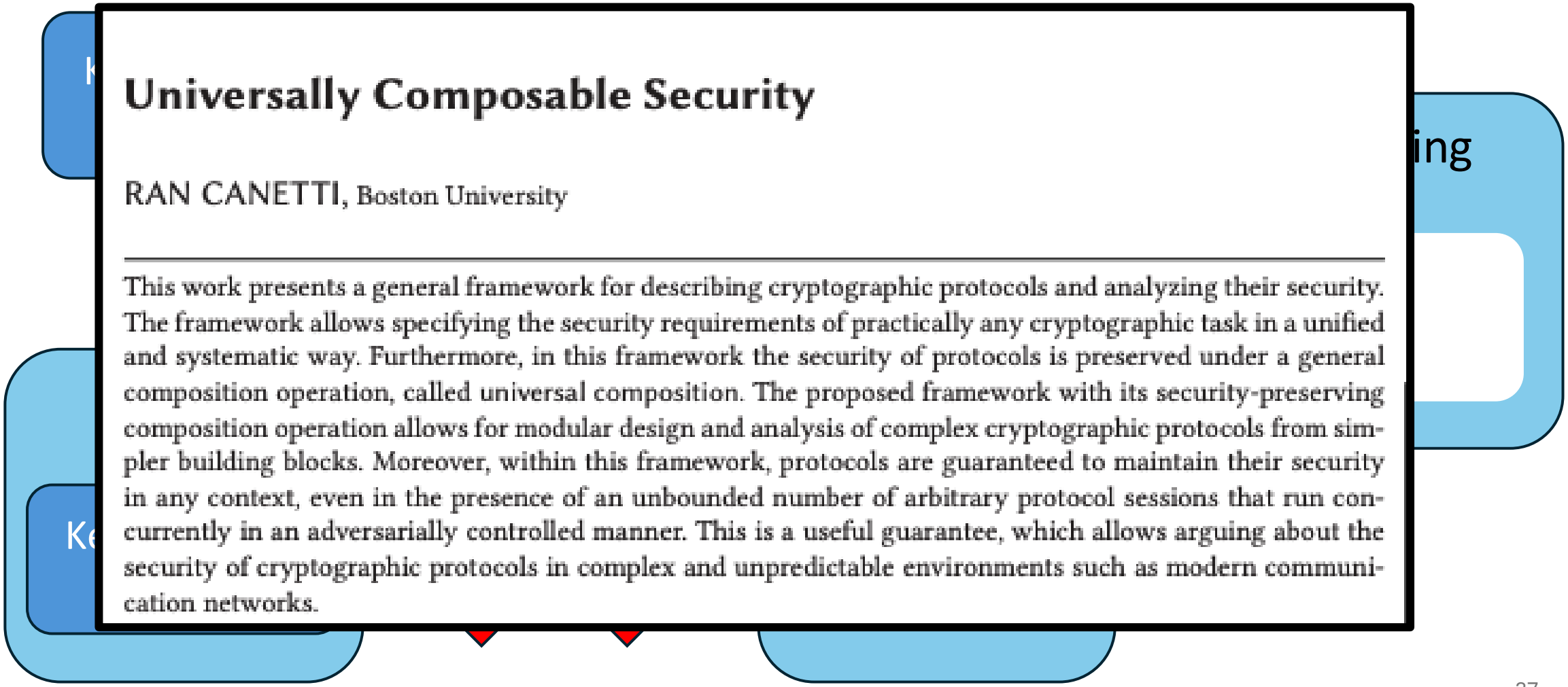
Composition Failure

* In *Standalone Model* of Real-Ideal paradigm



Composition Failure

* In *Standalone Model* of Real-Ideal paradigm



Universally Composable Security

RAN CANETTI, Boston University

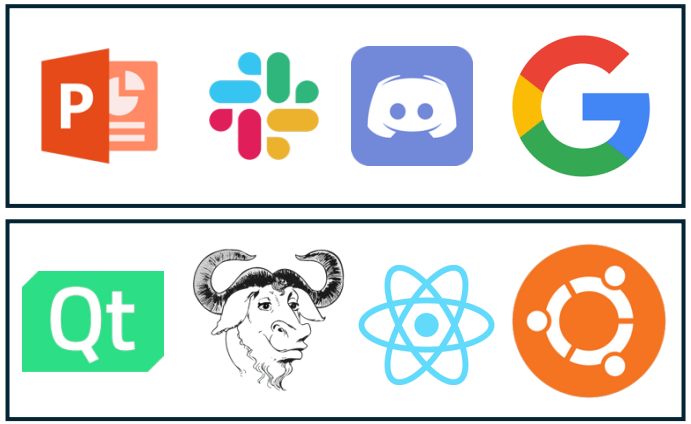
This work presents a general framework for describing cryptographic protocols and analyzing their security. The framework allows specifying the security requirements of practically any cryptographic task in a unified and systematic way. Furthermore, in this framework the security of protocols is preserved under a general composition operation, called universal composition. The proposed framework with its security-preserving composition operation allows for modular design and analysis of complex cryptographic protocols from simpler building blocks. Moreover, within this framework, protocols are guaranteed to maintain their security in any context, even in the presence of an unbounded number of arbitrary protocol sessions that run concurrently in an adversarially controlled manner. This is a useful guarantee, which allows arguing about the security of cryptographic protocols in complex and unpredictable environments such as modern communication networks.

Secure Application?

Secure Application?



Secure Application?



Secure Application?



Secure Application?

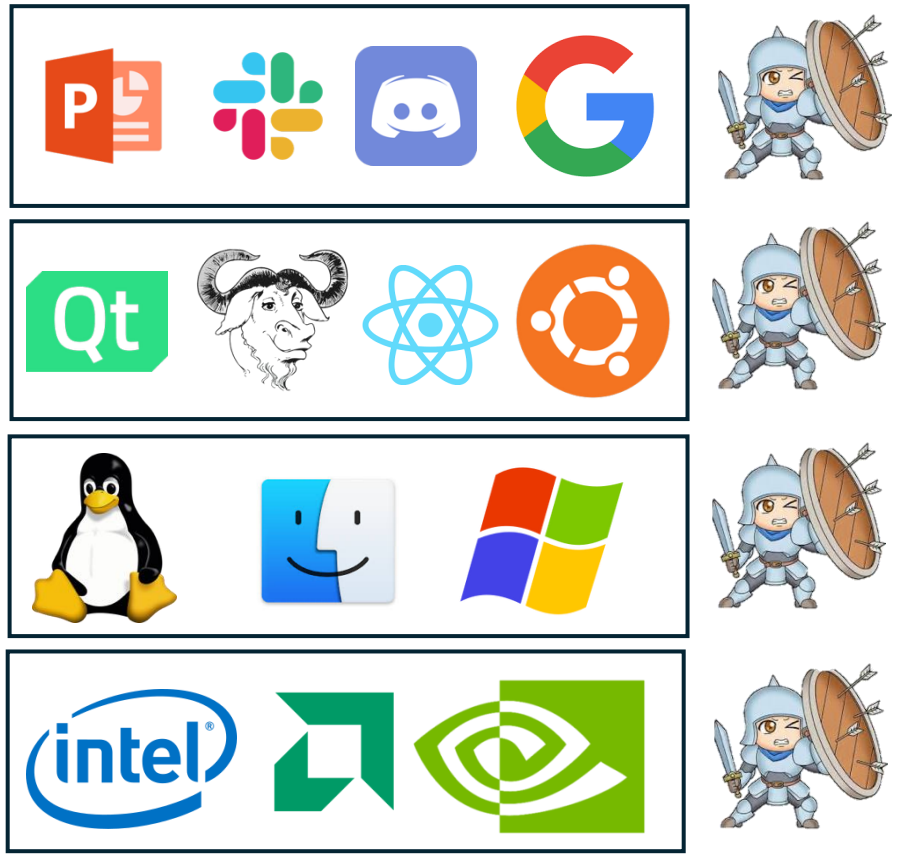


Secure Application?

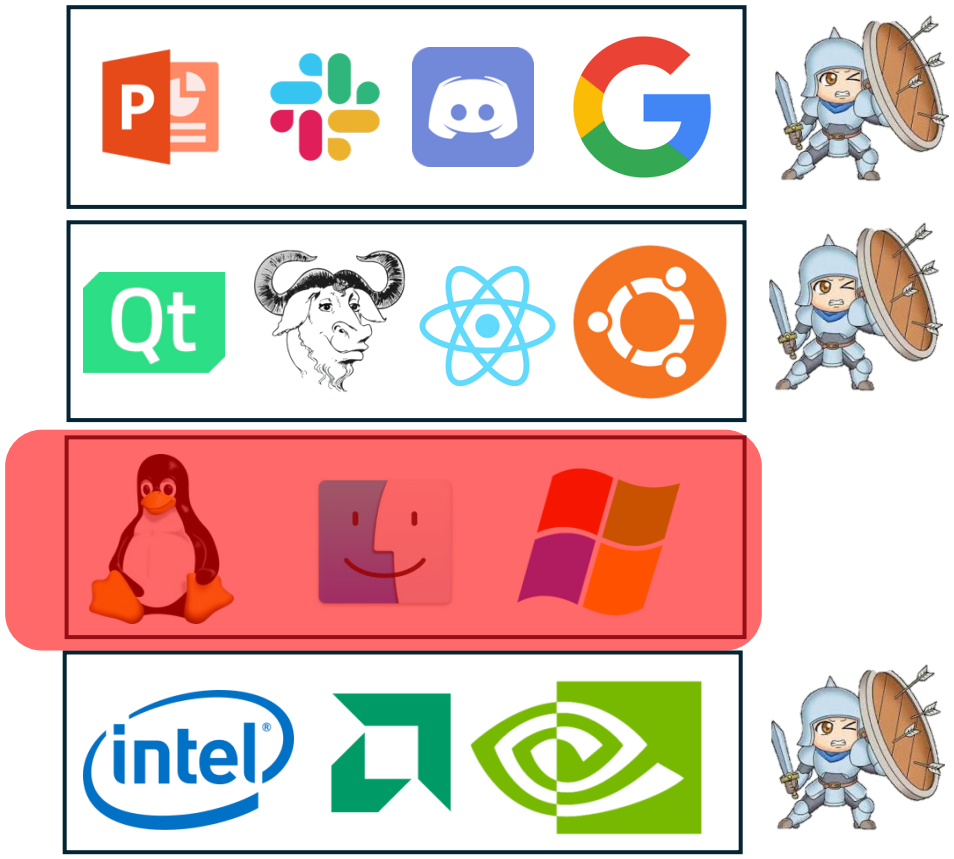




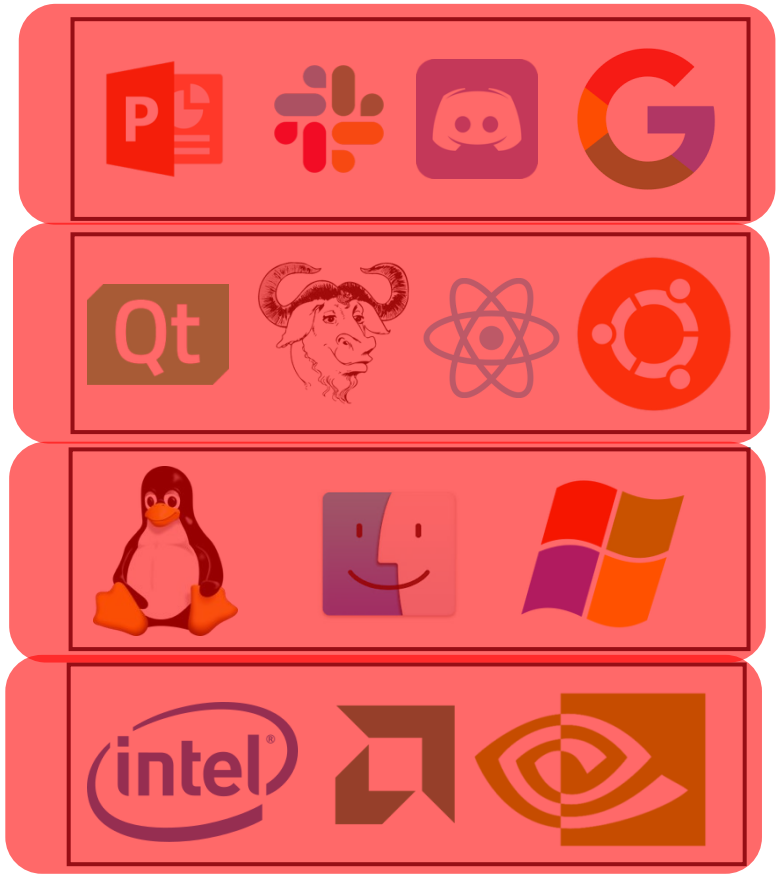
Secure Application?



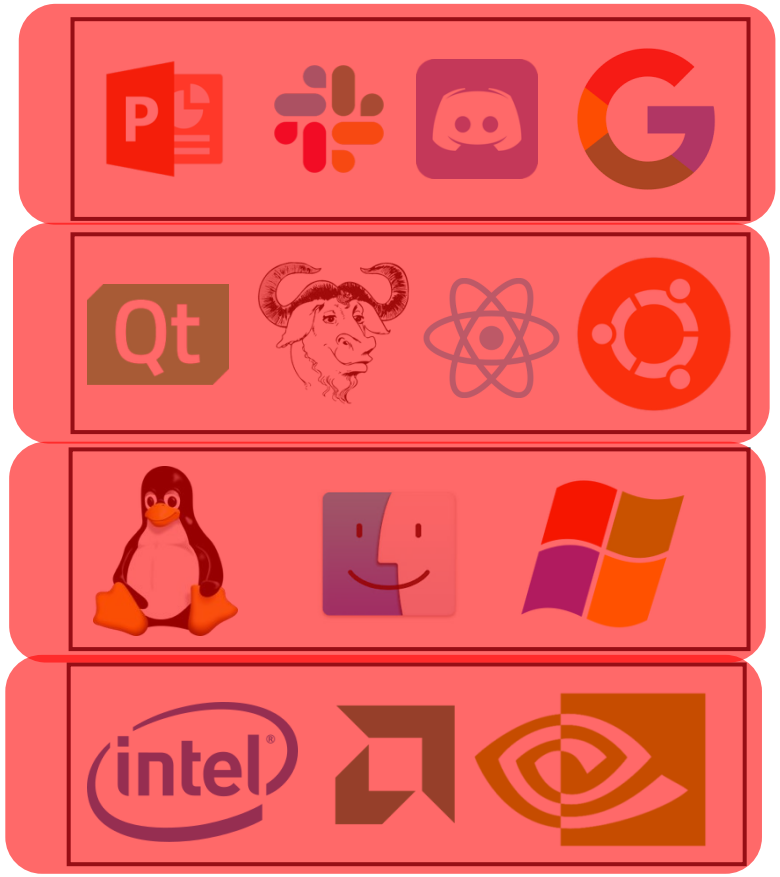
Secure Application?

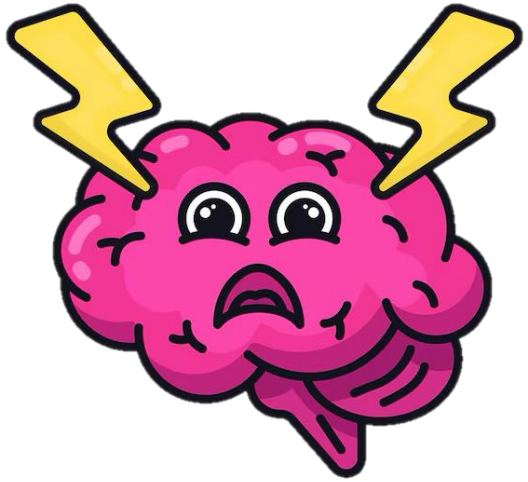


Secure Application?

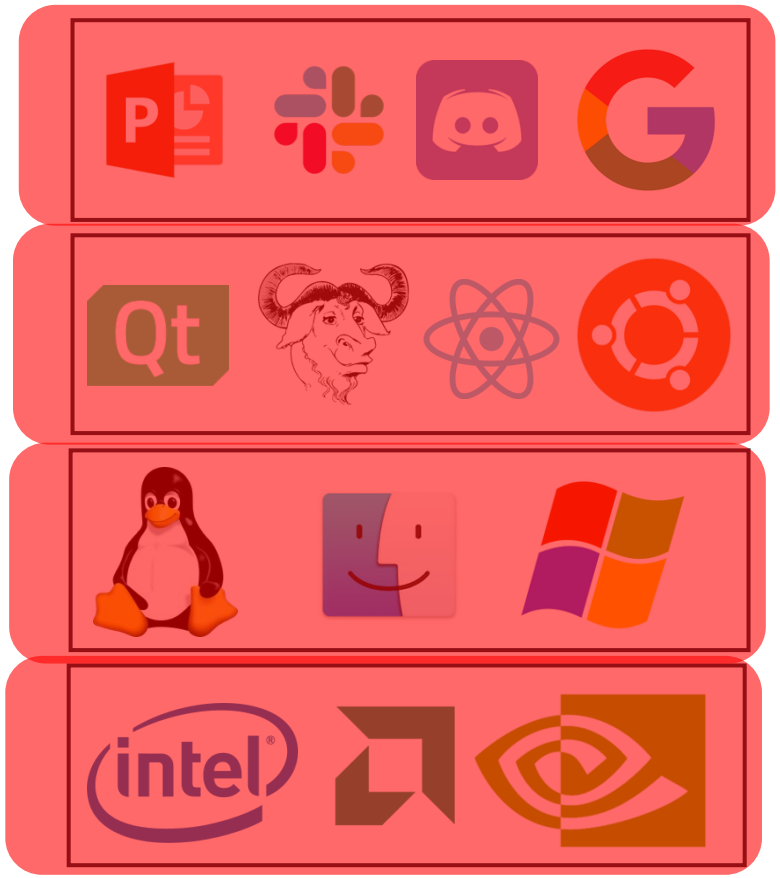


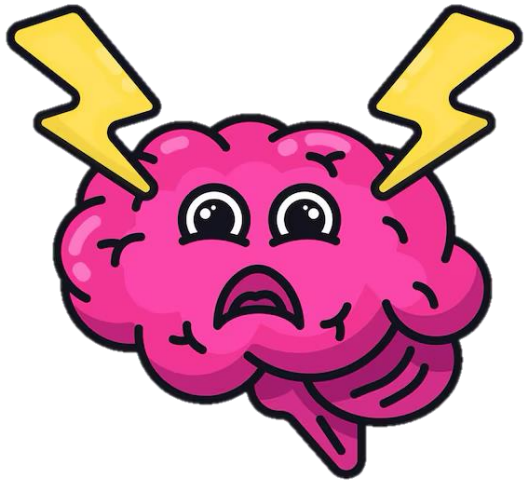
Secure Application?





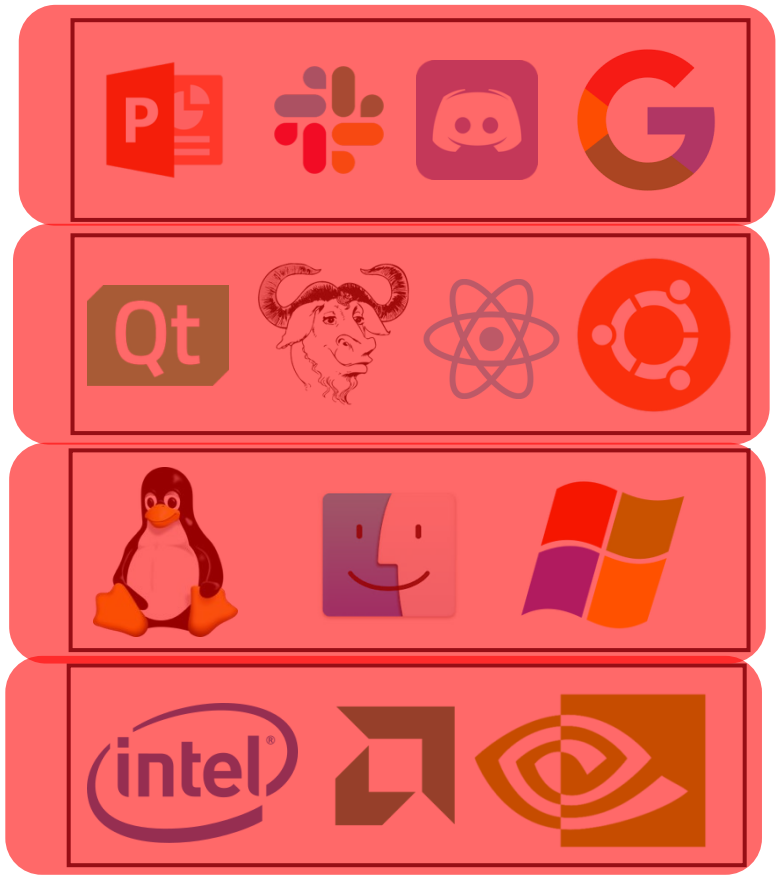
Secure Application?

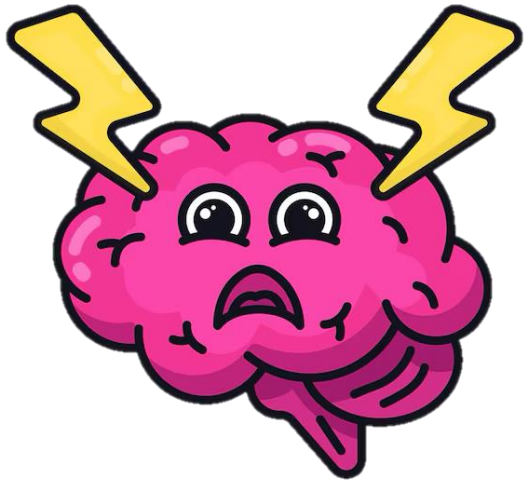






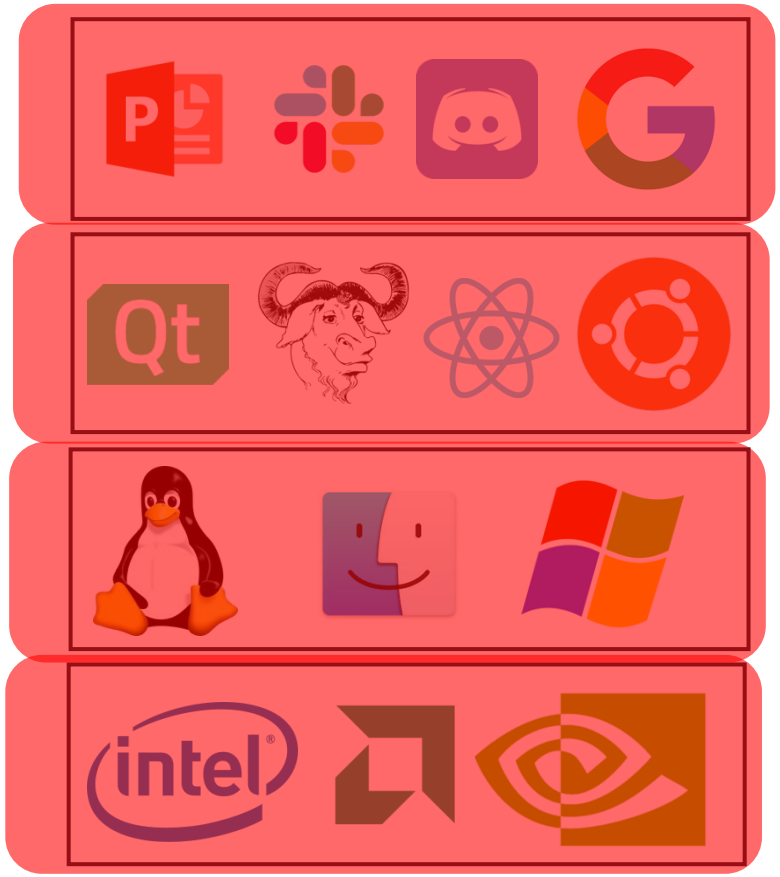
Secure Application?

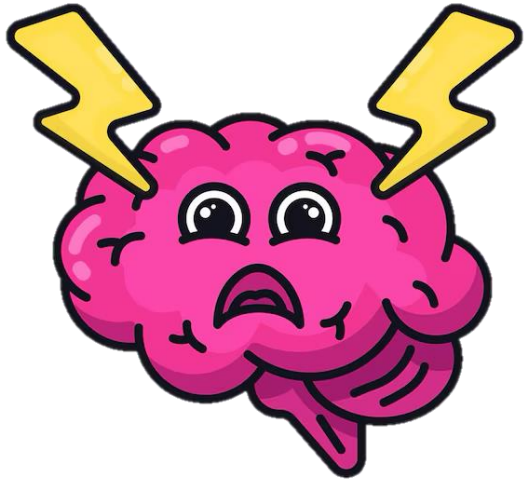




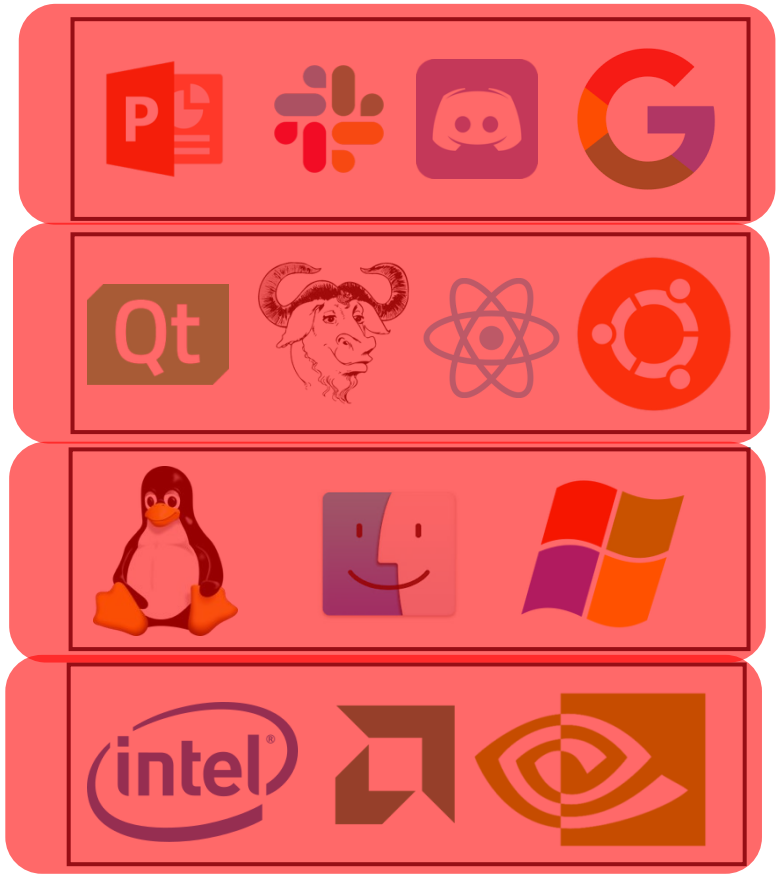


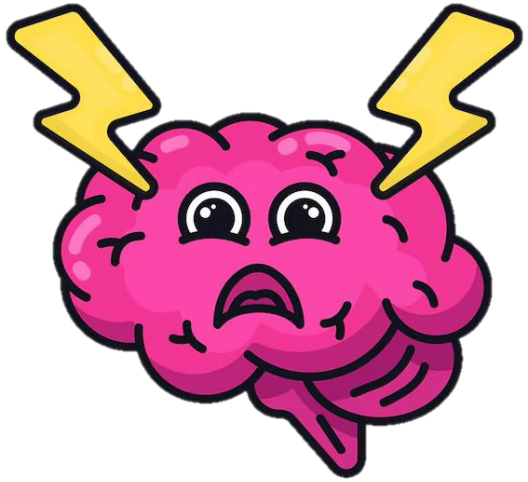
Secure Application?

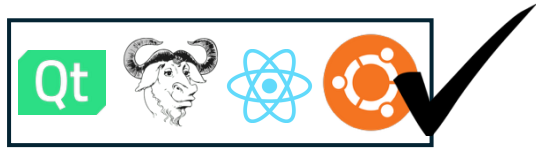




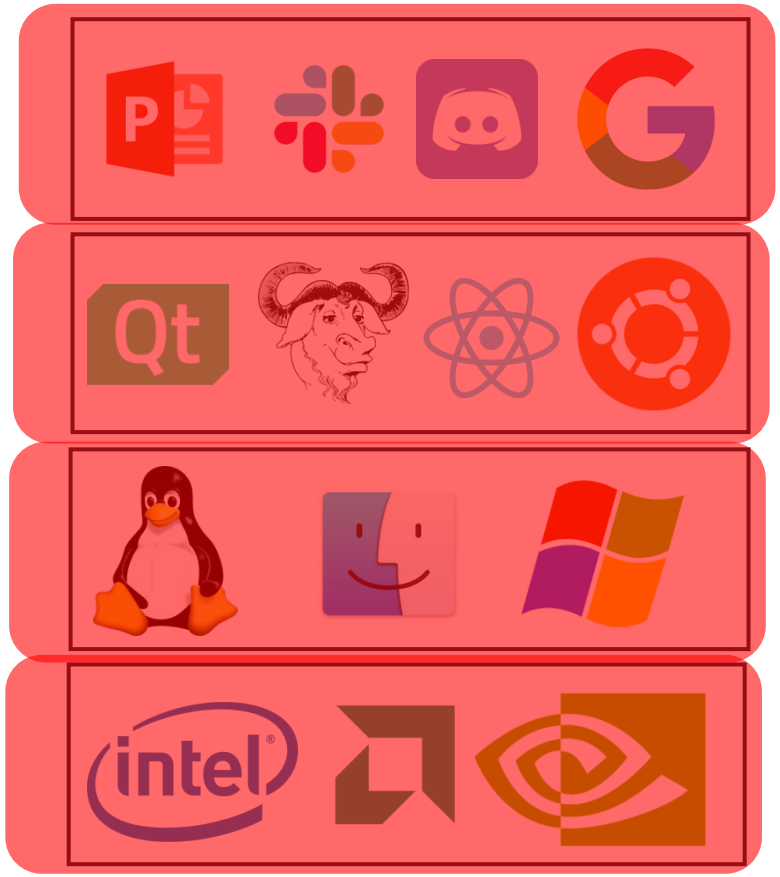
Secure Application?

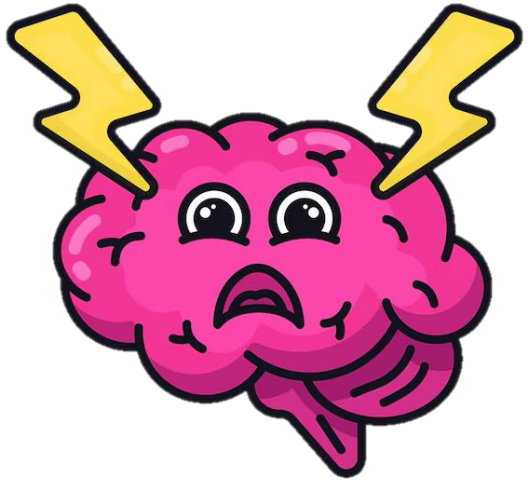


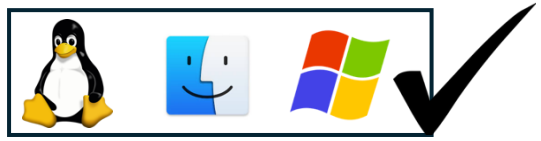




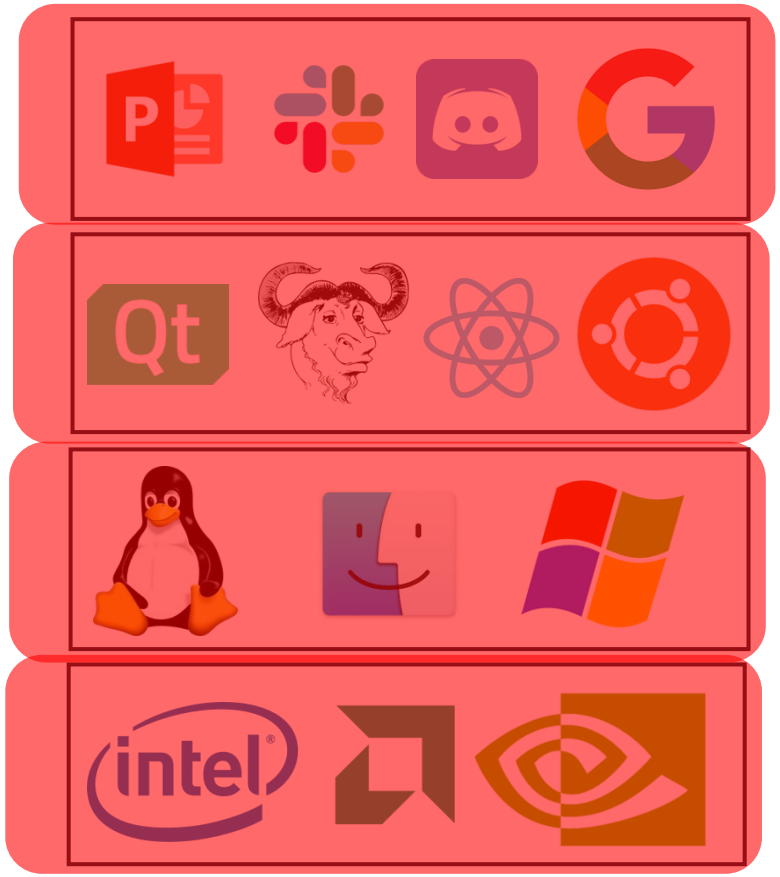
Secure Application?

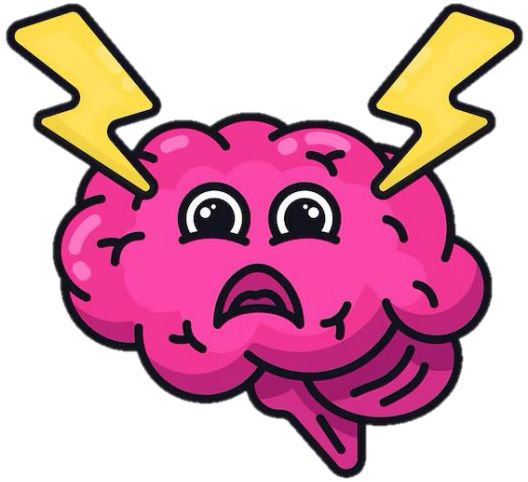






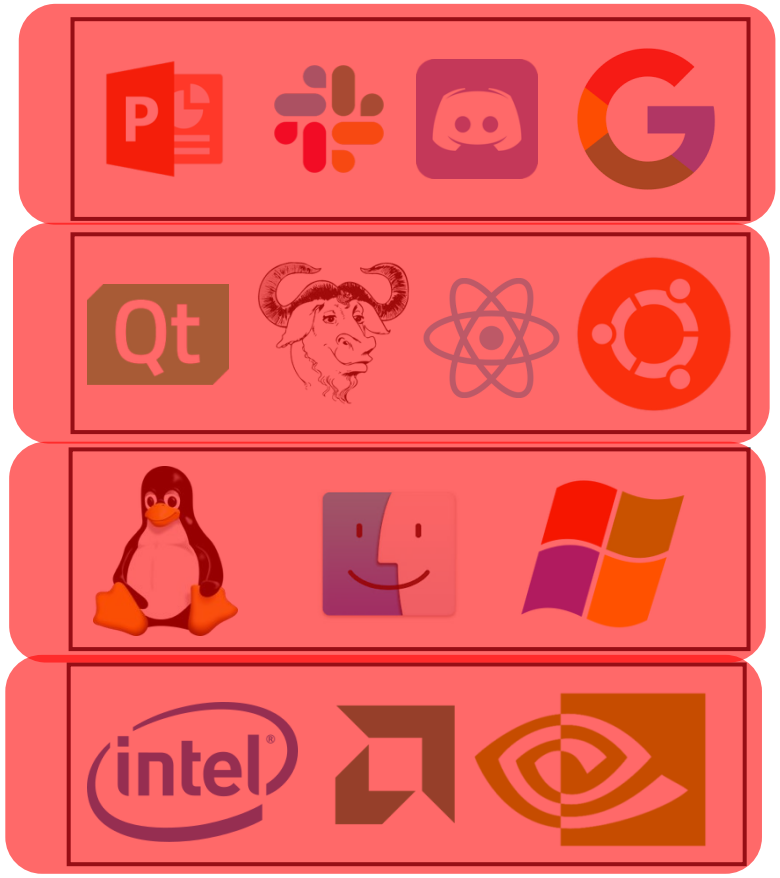
Secure Application?

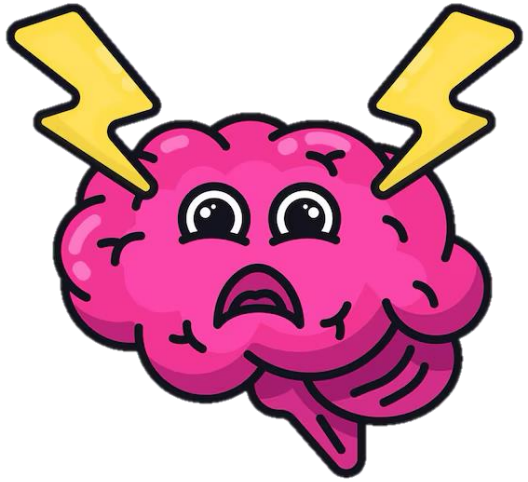


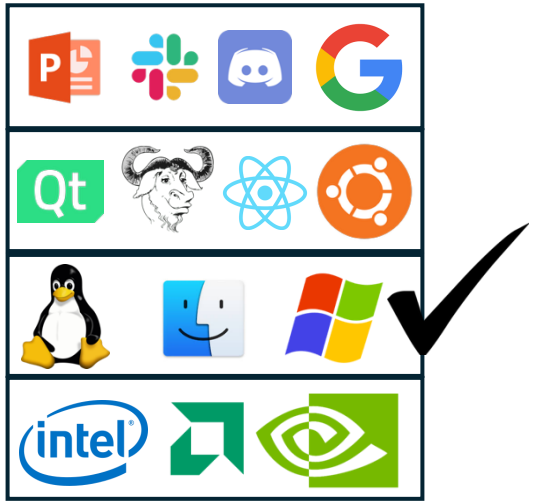




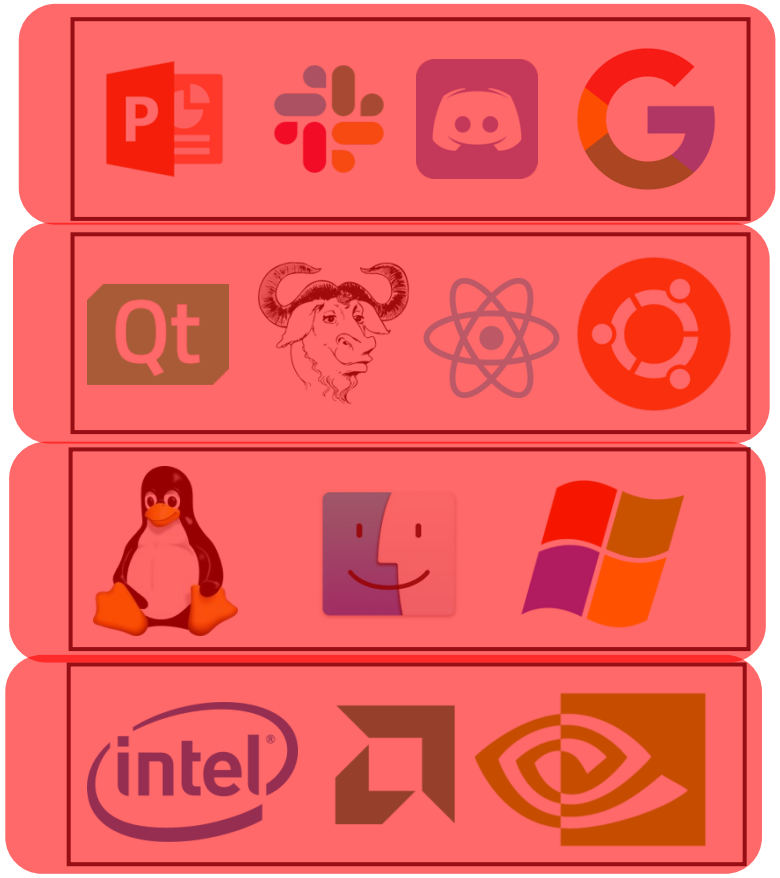
Secure Application?

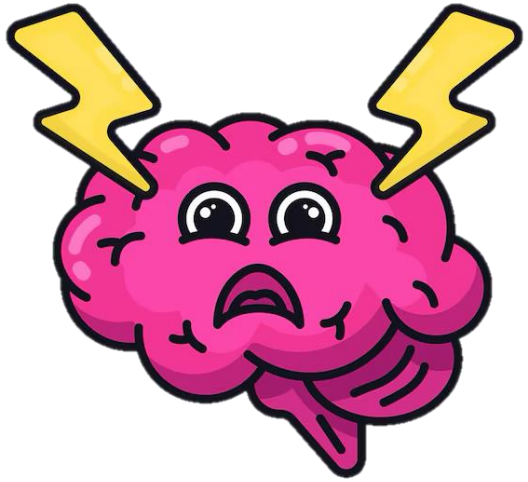


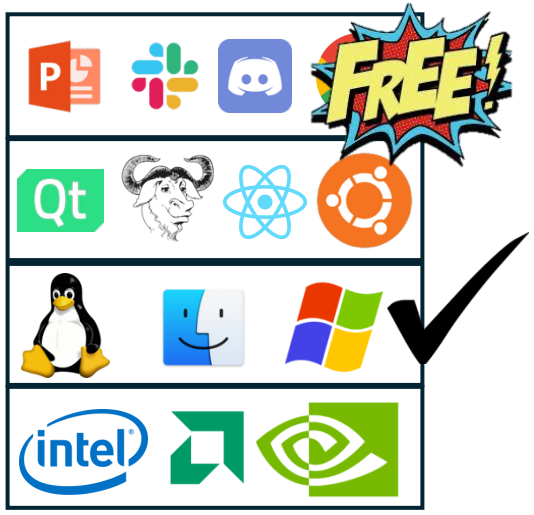




Secure Application?







Other Related Work

Other Related Work

Verifying Hardware Security Modules with Information-Preserving Refinement [PDF] from usenix.org

Authors	Anish Athalye, M Frans Kaashoek, Nickolai Zeldovich
Publication date	2022
Conference	16th USENIX Symposium on Operating Systems Design and Implementation
Pages	503-519
Description	Knox is a new framework that enables developers to build hardware security modules (HSMs) with high assurance through formal verification. The goal is to rule out all hardware bugs, software bugs, and timing side channels.

Other Related Work

Verifying Hardware Security Modules with Information-Preserving Refinement

[PDF] from usenix.org

Authors	Anish Athalye, M Frans Kaashoek, Nickolai Zeldovich
Publication date	2022
Conference	16th USENIX Symposium on Operating Systems Design and Implementation
Pages	503-519

Relative security: Formally modeling and (dis) proving resilience against semantic optimization vulnerabilities

[PDF] from andreipopescu.uk

Authors	Brijesh Dongol, Matt Griffin, Andrei Popescu, Jamie Wright
Publication date	2024/7/8
Conference	2024 IEEE 37th Computer Security Foundations Symposium (CSF)
Pages	403-418
Publisher	IEEE
Description	Meltdown and Spectre are vulnerabilities known as transient execution vulnerabilities, where an attacker exploits speculative execution (a semantic optimization present in most modern processors) to break confidentiality. We introduce relative security, a general notion of information-flow security that models this type of vulnerability by contrasting the leaks that are possible in a “vanilla” semantics with those possible in a different semantics, often obtained from the vanilla semantics via some optimizations. We describe incremental proof methods, in the style of Goguen and Meseguer's unwinding, both for proving and for disproving relative security, and deploy these to formally establish the relative (in)security of some standard Spectre examples. Both the abstract results and the case studies have been mechanized in the Isabelle/HOL theorem prover.

Other Related Work

Verifying Hardware Security Modules with In

Authors: Anish Athalye, M Frans Kaashoek, Nickolai Zelc

Publication date: 2022

Conference: 16th USENIX Symposium on Operating System

Pages: 503-519

Relative security: Formally modeling and semantic optimization vulnerabilities

Authors: Brijesh Dongol, Matt Griffin, Andrei Popes

Publication date: 2024/7/8

Conference: 2024 IEEE 37th Computer Security Found

Pages: 403-418

Publisher: IEEE

Description: Meltdown and Spectre are vulnerabilities where an attacker exploits speculative ex most modern processors) to break confid general notion of information-flow security contrasting the leaks that are possible in different semantics, often obtained from t We describe incremental proof methods, unwinding, both for proving and for dispro formally establish the relative (in)security abstract results and the case studies hav theorem prover.

Composable security analysis of OS services

Authors: Ran Canetti, Suresh Chari, Shai Halevi, Birgit Pfitzmann, Arnab Roy, Michael Steiner, Wietse Venema

Publication date: 2011

Conference: Applied Cryptography and Network Security: 9th International Conference, ACNS 2011, Nerja, Spain, June 7-10, 2011. Proceedings 9

Pages: 431-448

Publisher: Springer Berlin Heidelberg

Description: We provide an analytical framework for basic integrity properties of file systems, namely the binding of files to filenames and writing capabilities. A salient feature of our modeling and analysis is that it is *composable:* In spite of the fact that we analyze the filesystem in isolation, security is guaranteed even when the file system operates as a component within an arbitrary, and potentially adversarial system.

Our results are obtained by adapting the *Universally Composable* (UC) security framework to the analysis of software systems. Originally developed for cryptographic protocols, the UC framework allows the analysis of simple components in isolation, and provides assurance that these components maintain their behavior when combined in a large system, potentially under adversarial conditions.

Summing it up…

Summing it up…

- Existing security specification techniques are not satisfactory.

Summing it up...

- Existing security specification techniques are not satisfactory.
- Real-Ideal paradigm is a promising new methodology for security specification in broader settings.

Summing it up...

- Existing security specification techniques are not satisfactory.
- Real-Ideal paradigm is a promising new methodology for security specification in broader settings.
- In fact, the security of many existing systems can already by thought of as some form of the Real-Ideal paradigm.

Summing it up...

- Existing security specification techniques are not satisfactory.
- Real-Ideal paradigm is a promising new methodology for security specification in broader settings.
- In fact, the security of many existing systems can already by thought of as some form of the Real-Ideal paradigm.
- Research goals, both short-term and long-term, are presented which make this vision more concrete.


Always has been
It’s been the Real-Ideal paradigm all along?