

Verificación en Dinámica de Sistemas Distribuidos (y Posibles Conexiones con la IA)

Armando Castañeda

Instituto de Matemáticas, UNAM

Trabajo conjunto con [Gilde Valeria Rodríguez](#)

WIKIPEDIA 25 years of the free encyclopedia

Search Wikipedia

List of software bugs

Contents hide

Article Talk

From Wikipedia, the free encyclopedia

This is a *dynamic list* and may never be able to satisfy particular standards for completeness. You can add missing items, with references to *reliable sources*.

Many **software bugs** are merely annoying or inconvenient, but some can have extremely serious consequences as a threat to human well-being.^[1] The following is a **list of software bugs** with significant consequences

Administration
Blockchain
Electric power transmission
Encryption
Financial
Media
Military
Space
Telecommunications
Transportation
Video
See
References
External links

Medium Search

Get app Write

Member-only story

Famous Software Bugs Since 2000: Simple Lessons for Beginner Developers

www.youtube.com · watch

25 crazy software bugs explained

Find all the best dev content at <https://daily.dev/fireship> Let's explore 25 crazy **software bugs** that changed the world.

YouTube · Fireship · Nov 1, 2024

24 key moments in this video

www.youtube.com · watch

The 5 Software Bugs

Heisenbug: A **bug** that alters its behavior when being investigated. Bohrbug: A repeatable **bug** that remains consistent across executions.

YouTube · Cam Pedersen · Oct 31, 2023

www.youtube.com · watch

The Most Bizarre Software Bugs in History - Mia Bajić

We've all heard that we should test our **software**, but what happens when we don't? Sometimes, it leads to strange and unexplainable events.

Gleap Product Resources Pricing Login Sign up

Engineering

5 Software Bugs That Made History

July 29, 2025





Large language models in law: A survey

Jinqi Lai ^a, Wensheng Gan ^a, Jiayang Wu ^a, Zhenlian Qi ^b, Philip S. Yu ^c

Show more

+ Add to Mendeley Share Cite

<https://doi.org/10.1016/j.aiopen.2024.09.002>

Under a Creative Commons [license](#)

Abstract

The advent of artificial intelligence (AI) has significantly impacted the traditional industry. Moreover, recently, with the development of AI-generated content (AIGC) and law have found applications in various domains, including image recognition, automatic text generation, and interactive chat. With the rapid emergence and popularity of large models, it is evident that AI will drive transformation in the traditional judicial industry. However, the application of legal large language models (LLMs) is still in its nascent stage. Several challenges need to be addressed. In this survey, we aim to provide a comprehensive survey of legal LLMs. We not only conduct

Member-only story

Famous Software Bugs Since 2000: Simple Lessons for Beginner Developers

www.youtube.com · watch

25 crazy software bugs explained

[nature](#) > [news](#) > article

NEWS | 05 March 2026 | Clarification [06 March 2026](#)

How AI is shaping the war in Iran – and what’s next for future conflicts

Rapid technological development is prompting urgent discussions on regulating the use and procurement of artificial intelligence for military use.

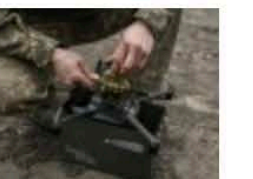
By [Nicola Jones](#)



You have full access to this article via **Universidad Nacional Autónoma de México.**

Related Articles

[Lethal AI weapons are here: how can we control them?](#)



¿Cómo garantizar corrección?



- “Program testing can be used to show the presence of bugs, but never to show their absence”
- “One can never guarantee that a proof is correct, the best one can say is: I have not discovered any mistakes”



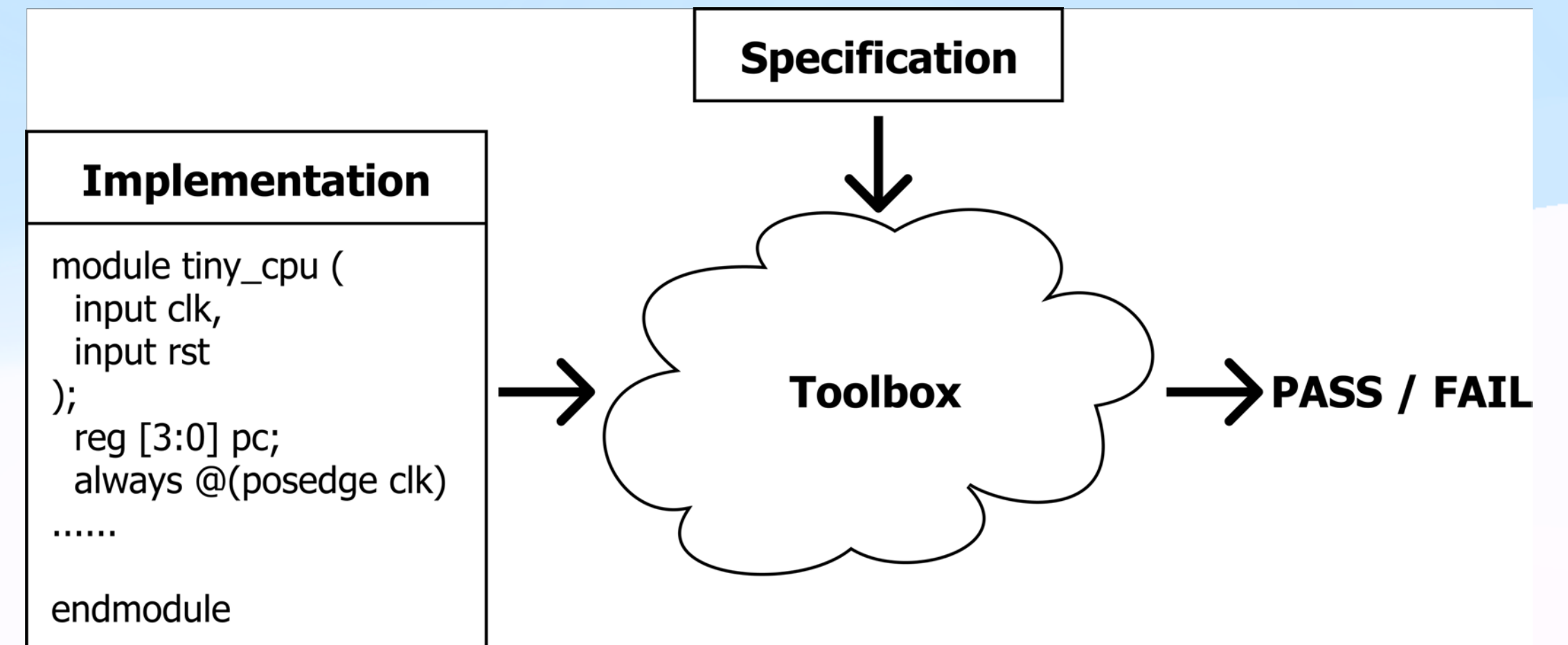
- No existe programa que decide si otro programa se detendrá con una entrada dada (indecidibilidad del problema de la detención).



- Determinar computacionalmente si una formula propositional es satisfacible es un problema NP-completo (Teorema de Cook-Levin).

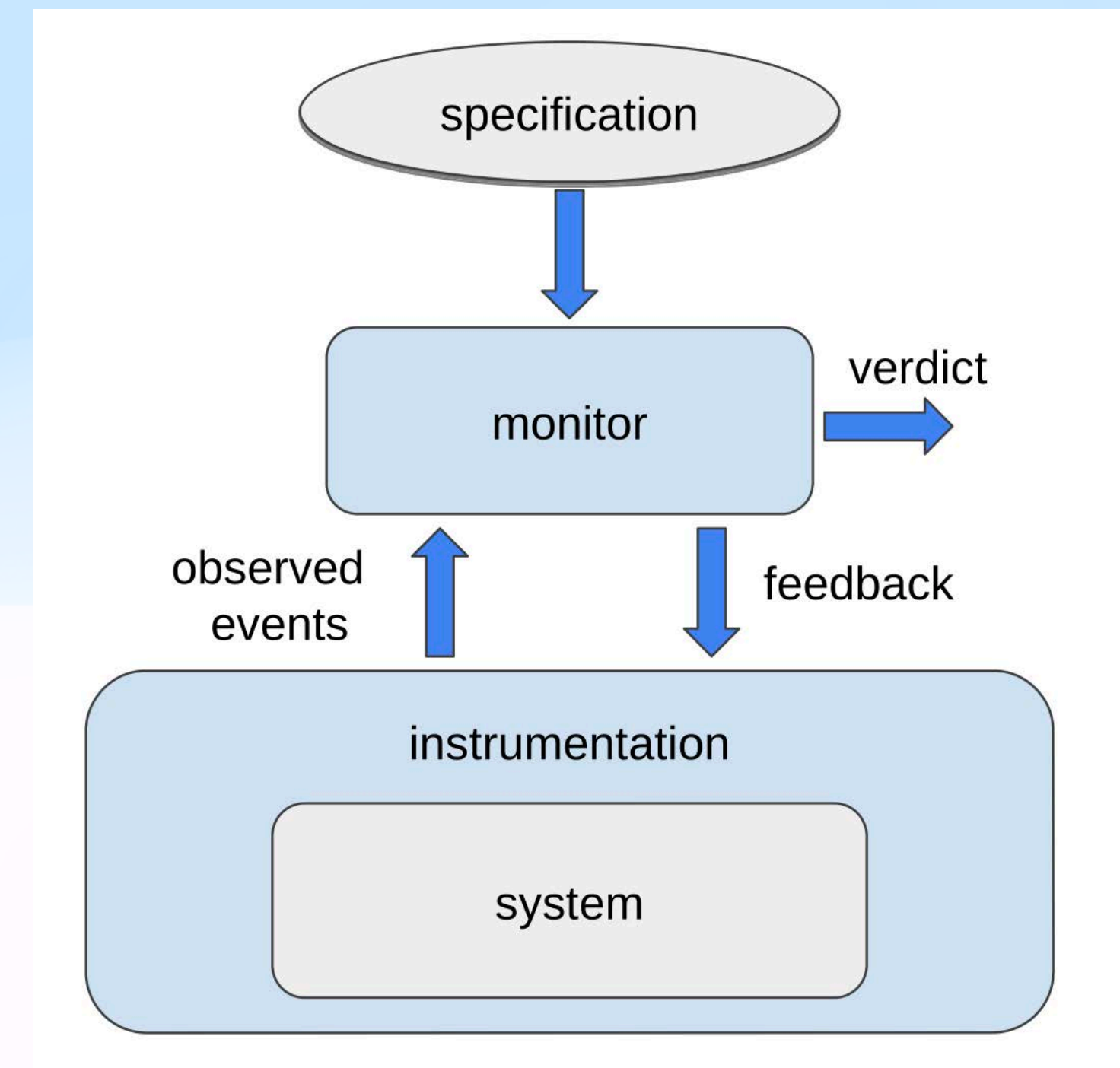
Verificación formal

- Pruebas formales (formal testing).
- Verificación de modelos (model checking).
- Verificación deductiva (theorem proving).
- Verificación en tiempo de ejecución (runtime verification).



Verificación en tiempo de ejecución (VTE)

- Técnica dinámica.
- Verificar la **ejecución actual** del sistema.
- Idealmente:
 - Hacer cumplir corrección (enforce correctness).
 - Rendición de cuentas (accountability).



Sistemas distribuidos

Sistemas compuestos de agentes con capacidades de cómputo y un medio de comunicación a través del cual los agentes se comunican y coordinan con el objetivo de solucionar un problema de forma cooperativa.

Sistemas distribuidos

Sistemas compuestos de agentes con capacidades de cómputo y un medio de comunicación a través del cual los agentes se comunican y coordinan con el objetivo de solucionar un problema de forma cooperativa.



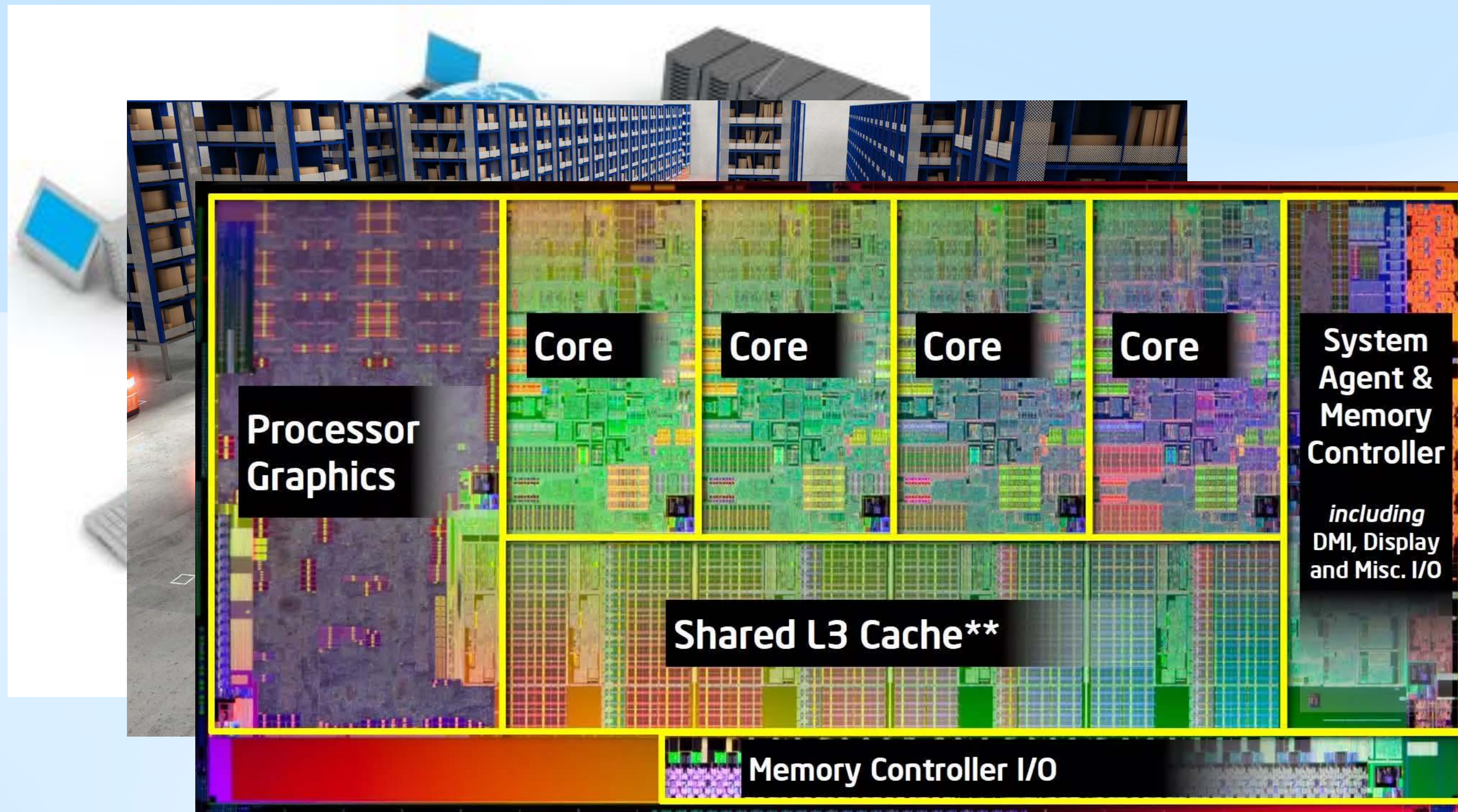
Sistemas distribuidos

Sistemas compuestos de agentes con capacidades de cómputo y un medio de comunicación a través del cual los agentes se comunican y coordinan con el objetivo de solucionar un problema de forma cooperativa.



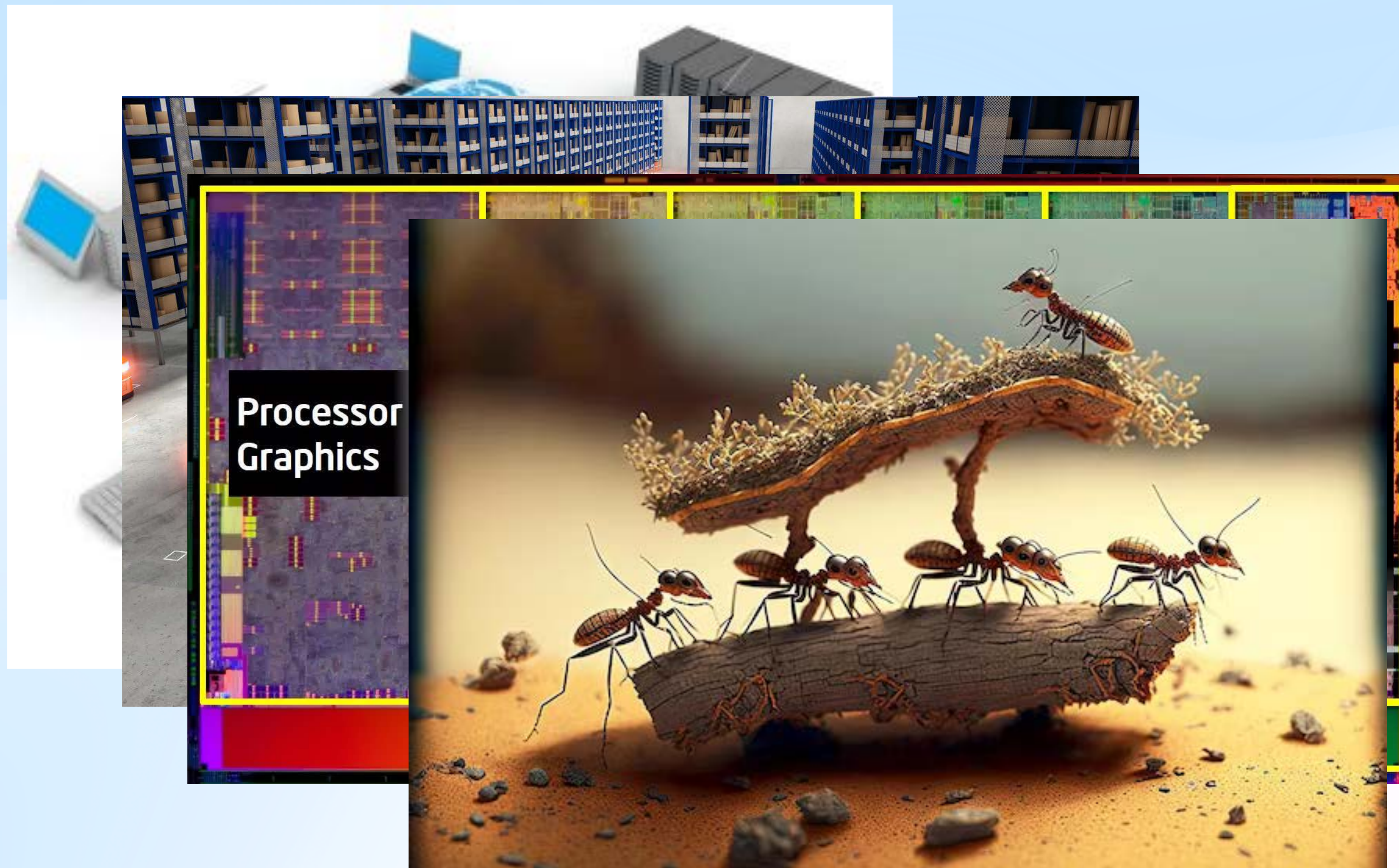
Sistemas distribuidos

Sistemas compuestos de agentes con capacidades de cómputo y un medio de comunicación a través del cual los agentes se comunican y coordinan con el objetivo de solucionar un problema de forma cooperativa.



Sistemas distribuidos

Sistemas compuestos de agentes con capacidades de cómputo y un medio de comunicación a través del cual los agentes se comunican y coordinan con el objetivo de solucionar un problema de forma cooperativa.

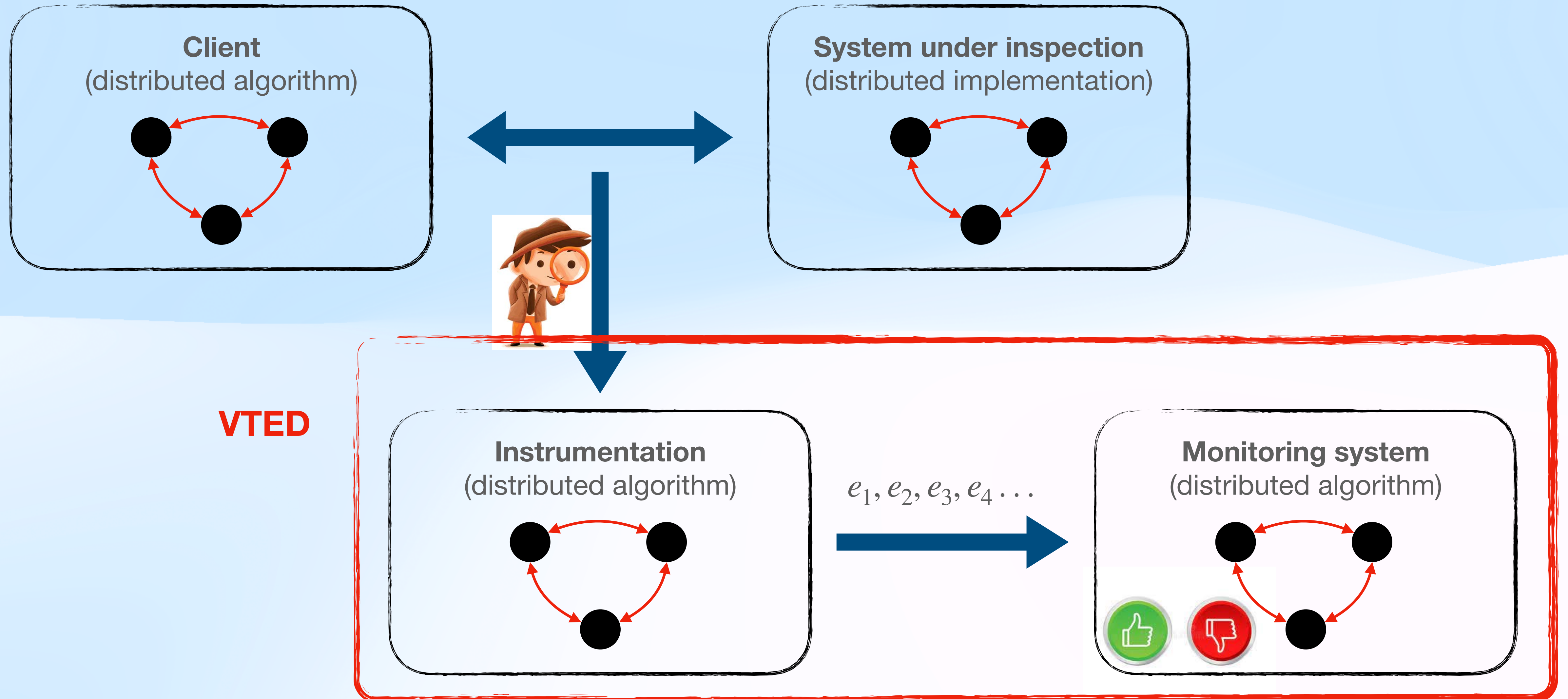


Sistemas distribuidos

Sistemas compuestos de agentes con capacidades de cómputo y un medio de comunicación a través del cual los agentes se comunican y coordinan con el objetivo de solucionar un problema de forma cooperativa.

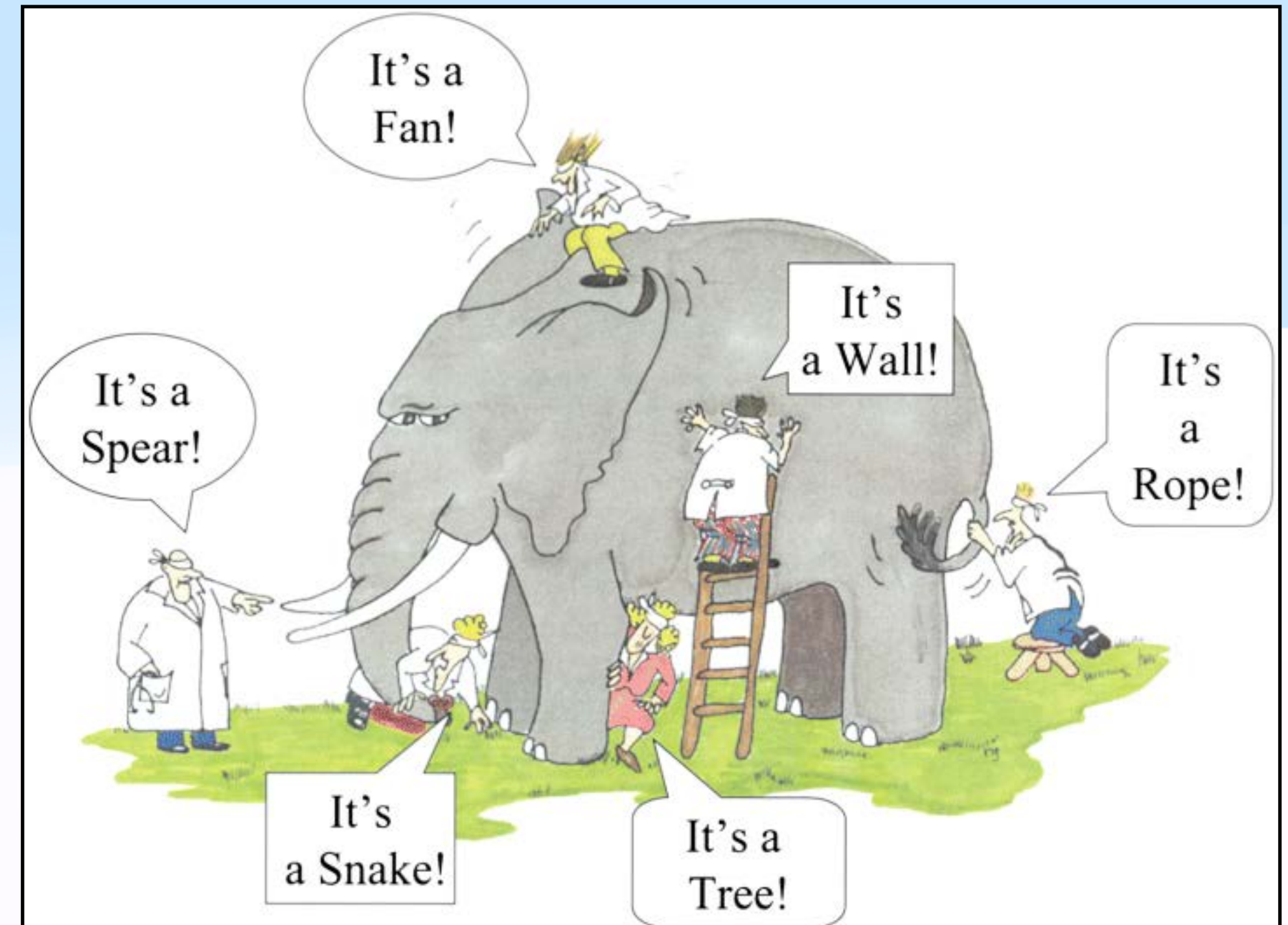


VTE distribuida (VTED)



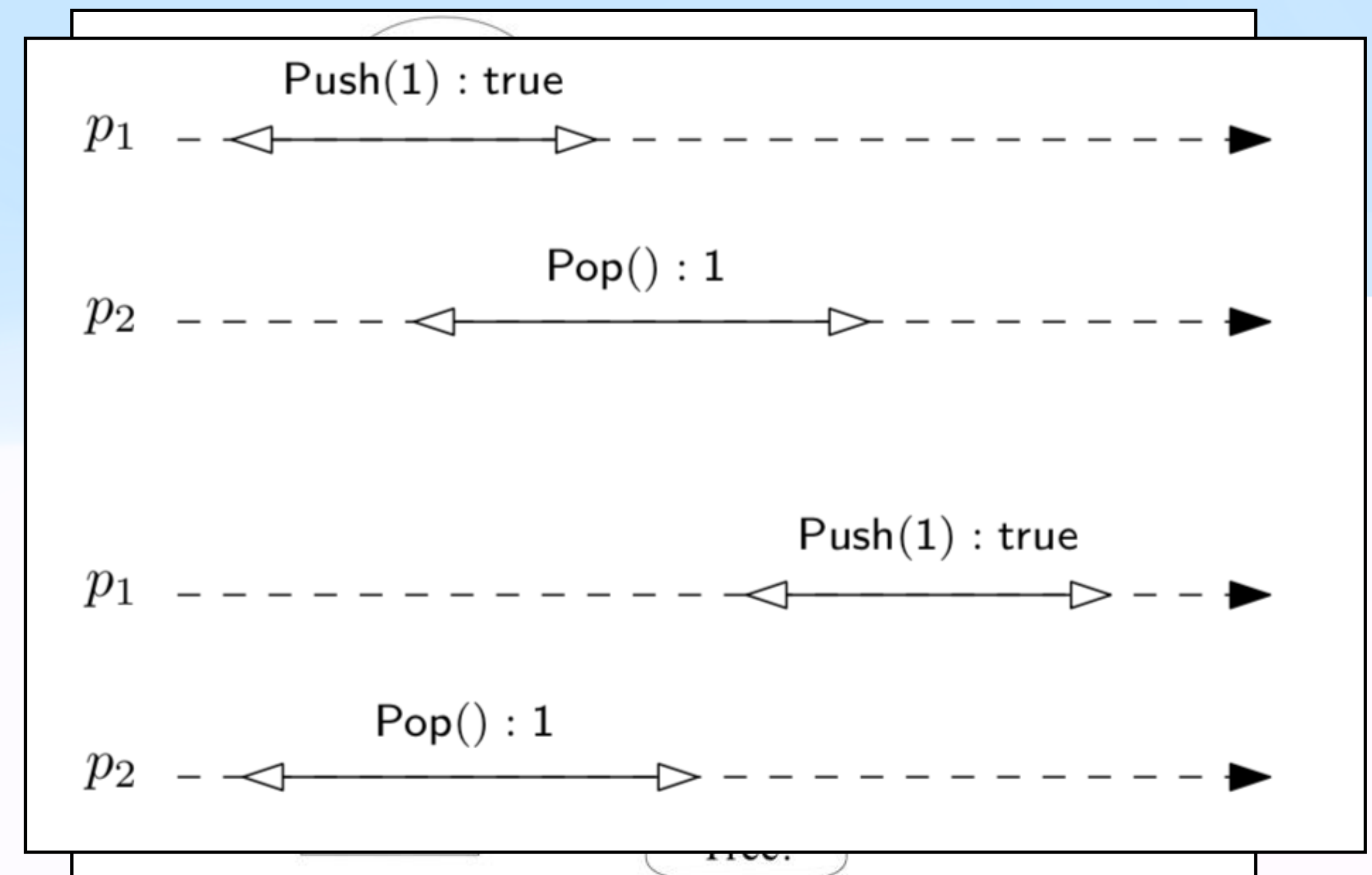
Reto principal: indistinguibilidad

- Ningún agente/proceso/computadora “conoce” la ejecución actual.
- Vistas locales de la ejecución total.
- ¡Indistinguibilidad!
- VTED es un problema distribuido.



Reto principal: indistinguibilidad

- Ningún agente/proceso/computadora “conoce” la ejecución actual.
- Vistas locales de la ejecución total.
- ¡Indistinguibilidad!
- VTED es un problema distribuido.



Nuestra propuesta

- Estudio de VTED desde un perspectiva algorítmica; teoría del cómputo distribuido.
- Modelo teórico de “caja negra”:

decibilidad de lenguajes + contexto distribuido + adversarial

- Objetivo: desarrollar una teoría de VTED para entender los límites de lo realizable, teoría con potenciales implementaciones en la práctica.

Nuestra propuesta

Gilde Valeria Rodríguez:

Verificación de la linealizabilidad en tiempo de ejecución.

Tesis de Maestría, PCIC UNAM (2022).

Armando Castañeda, Gilde Valeria Rodríguez:

Asynchronous Wait-Free Runtime Verification and Enforcement of Linearizability.

Journal of the ACM 73(1): 3:1-3:37 (2026).

Versión de congreso en **PODC 2023**

Gilde Valeria Rodríguez, Armando Castañeda:

Towards Efficient Runtime Verified Linearizable Algorithms.

RV 2024: 262-281 (Runner-Up Best Paper Award)

Version extendida enviada por invitación a **International Journal on Software Tools for Technology Transfer, Special Issue RV 2024**

Nuestra propuesta

Armando Castañeda, Gilde Valeria Rodríguez:
Asynchronous Fault-Tolerant Language Decidability for Runtime Verification of Distributed Systems.
PODC 2025: 218-229

Gilde Valeria Rodríguez, Miguel Angel Piña, Armando Castañeda:
A Reliable Non-Intrusive Runtime Verification Framework for Linearizability.
Aceptado en **Science of Computer Programming**.

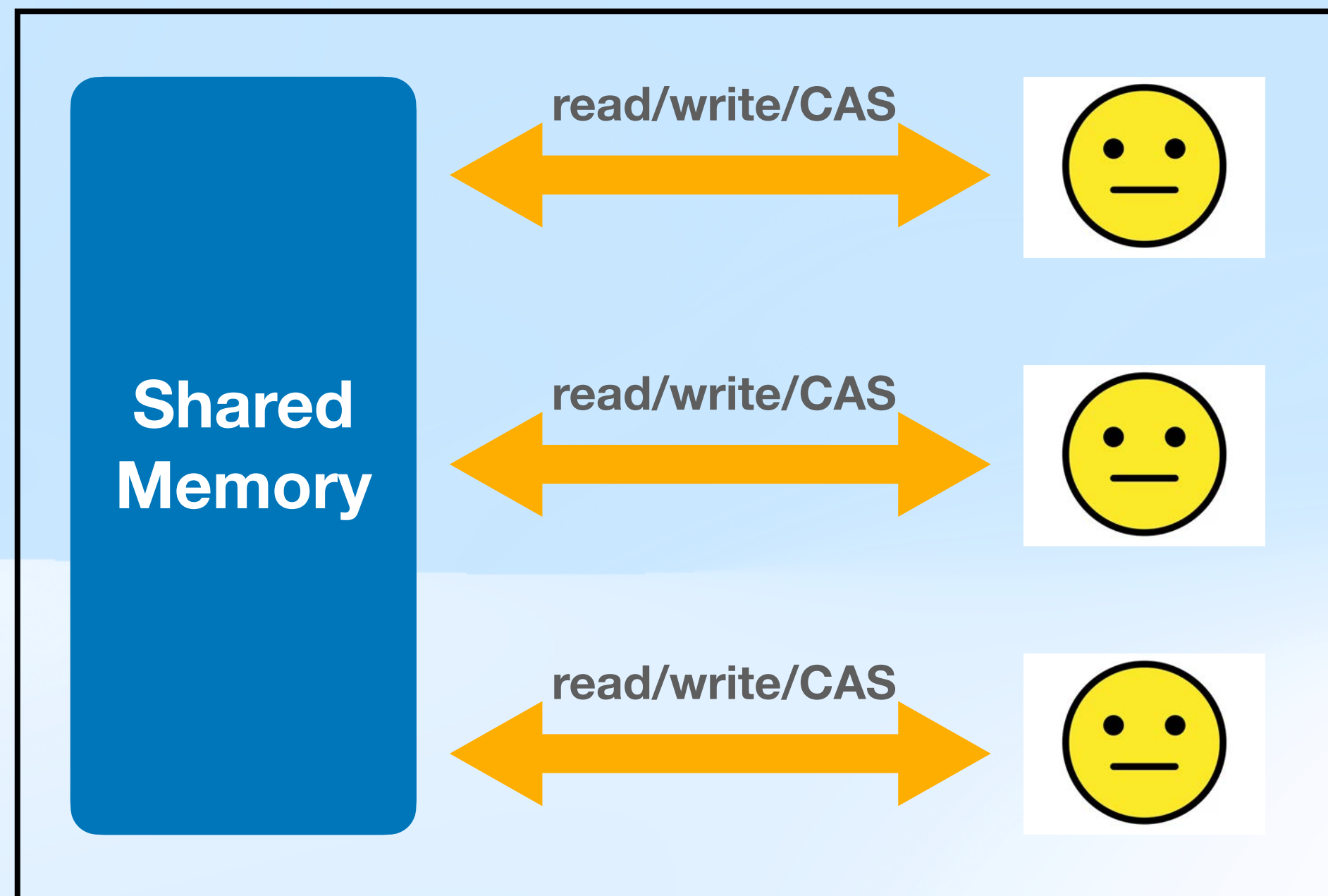
Gilde Valeria Rodríguez, Miguel Angel Piña:
Prototipo funcional para programas multi-hilo linealizables en Java.
<https://github.com/PRISM-Concurrent/efficient-distributed-rv>

Trabajo previo

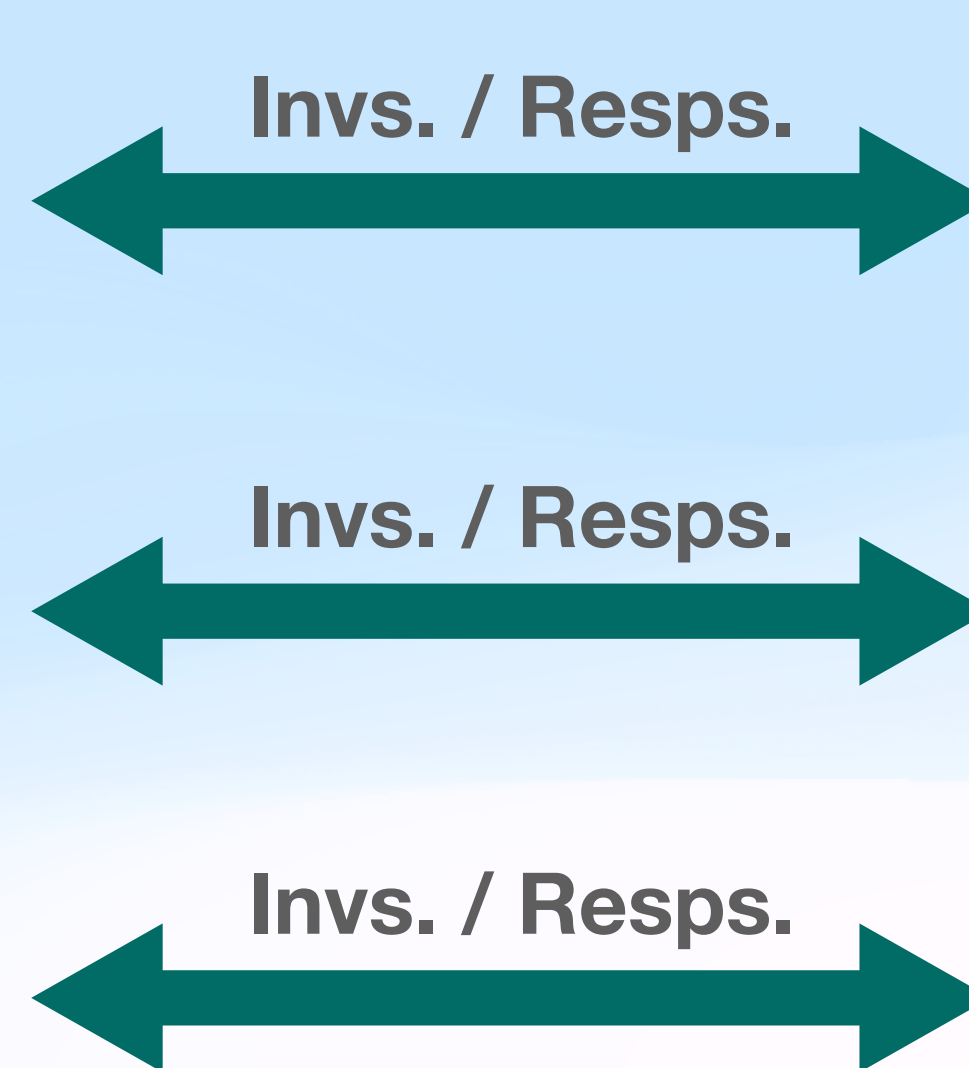
- VTED es un área de estudio aún emergente, con fuertes motivaciones prácticas.
- Propuestas para tipos diversos sistemas y propiedades.
- Todas con fuertes suposiciones del sistema bajo inspección (principalmente de tiempo).
- Algunas con implementaciones prácticas.

Modelo adversarial para VDTE

Verificador



∞



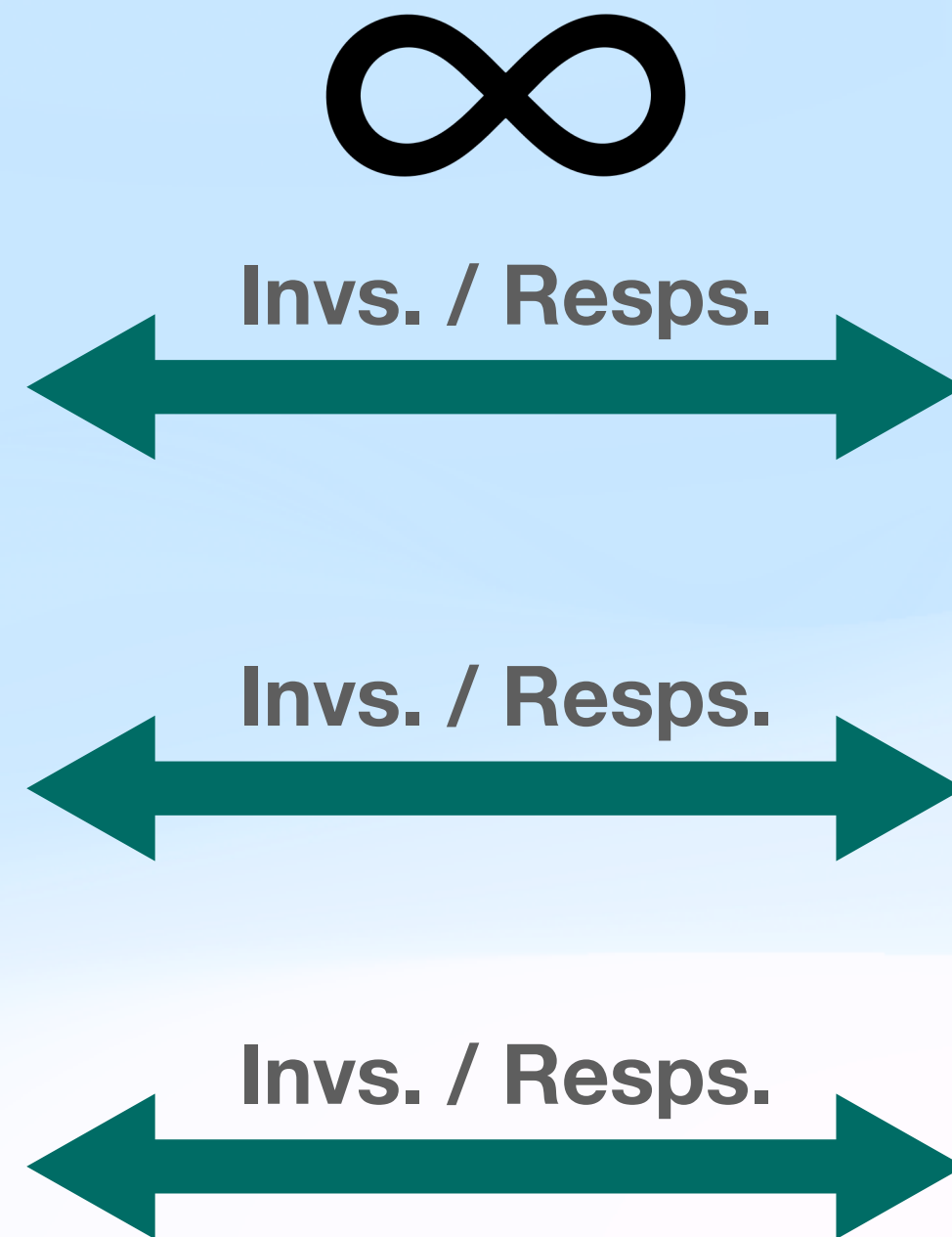
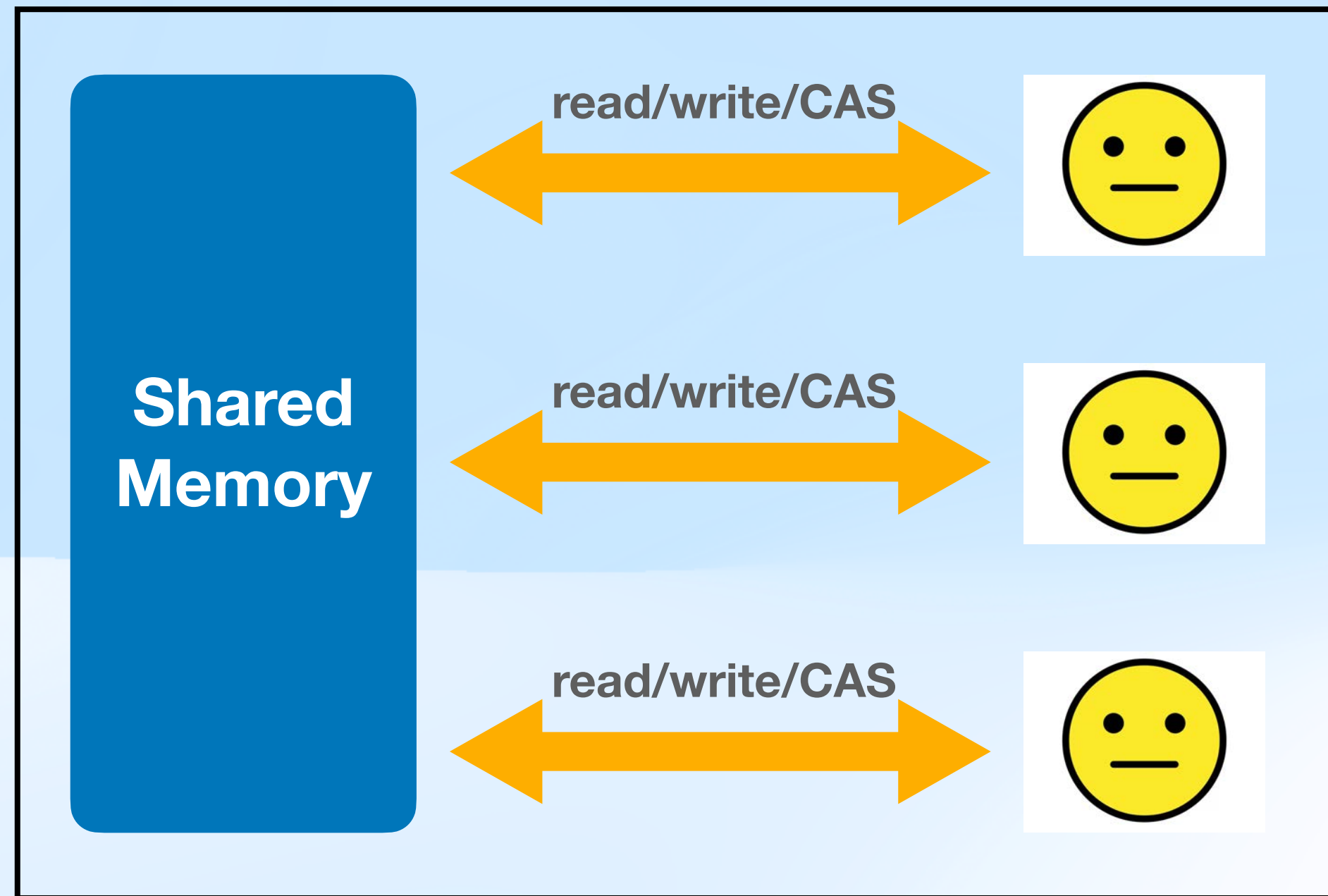
Adversario A



Modelo adversarial para VDTE

Verificador

Adversario A



$$\begin{aligned}
 x &= \langle \rangle [] \{ \} \\
 x_1 &= \langle \rangle \\
 x_2 &= [] \\
 x_3 &= \{ \}
 \end{aligned}$$

$$\begin{aligned}
 x' &= [] \langle \rangle' \{ \} \\
 x'_1 &= \langle \rangle' \\
 x'_2 &= [] \\
 x'_3 &= \{ \}
 \end{aligned}$$

$$\begin{aligned}
 x'' &= \langle [] \rangle' \{ \}' \\
 x''_1 &= \langle \rangle' \\
 x''_2 &= [] \\
 x''_3 &= \{ \}'
 \end{aligned}$$

Resultados

Verificación fuerte: todo agente reporta **NO** si y solo si el comportamiento del adversario es incorrecto.

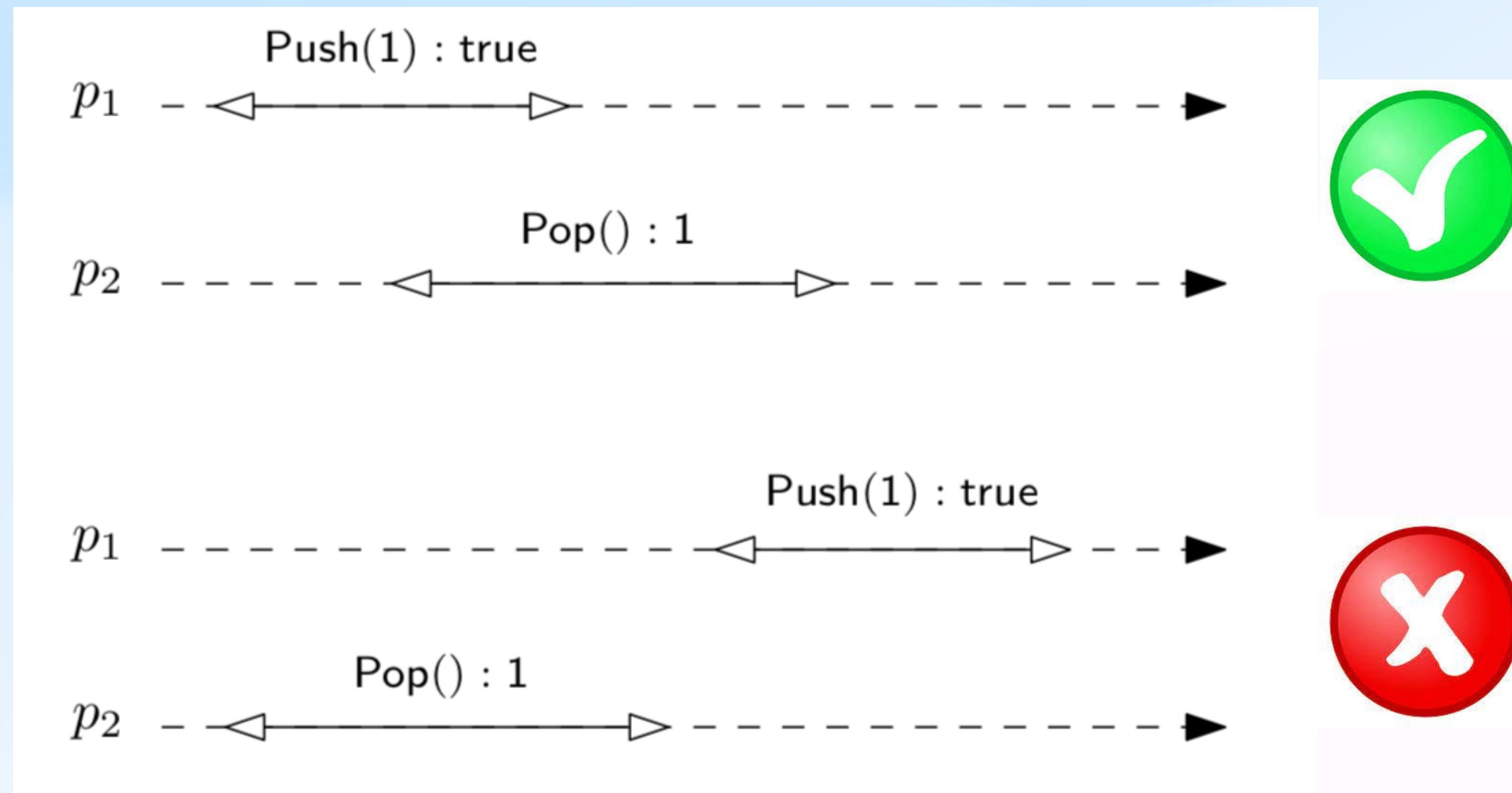
Verificación débil: todo agente reporta **NO** un número infinito de veces si y solo si el comportamiento del adversario es incorrecto.

✘ impossible
✔ possible

D. Language \ Model	Adversary A	
	Strong	Weak
<i>Linearizable register</i>	✘	✘
<i>Seq. consistent register</i>	✘	✘
<i>Linearizable ledger</i>	✘	✘
<i>Seq. consistent ledger</i>	✘	✘
<i>Event. consistent ledger</i>	✘	✘
<i>Weak event. counter</i>	✘	✔
<i>Strong event. counter</i>	✘	✘

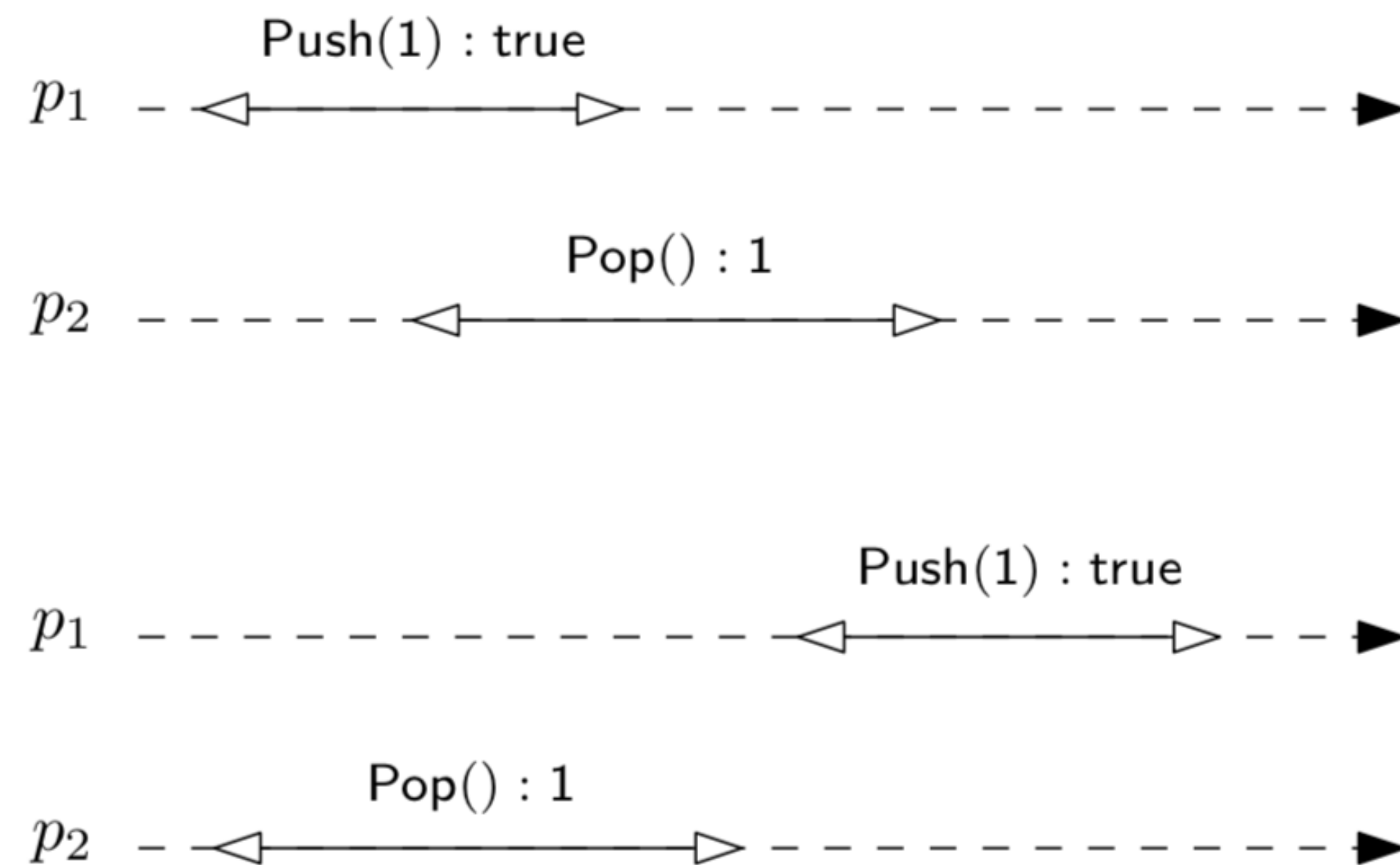
Resultados

Propiedades sensibles al orden: el orden en que ocurren las cosas es determinante para la corrección.



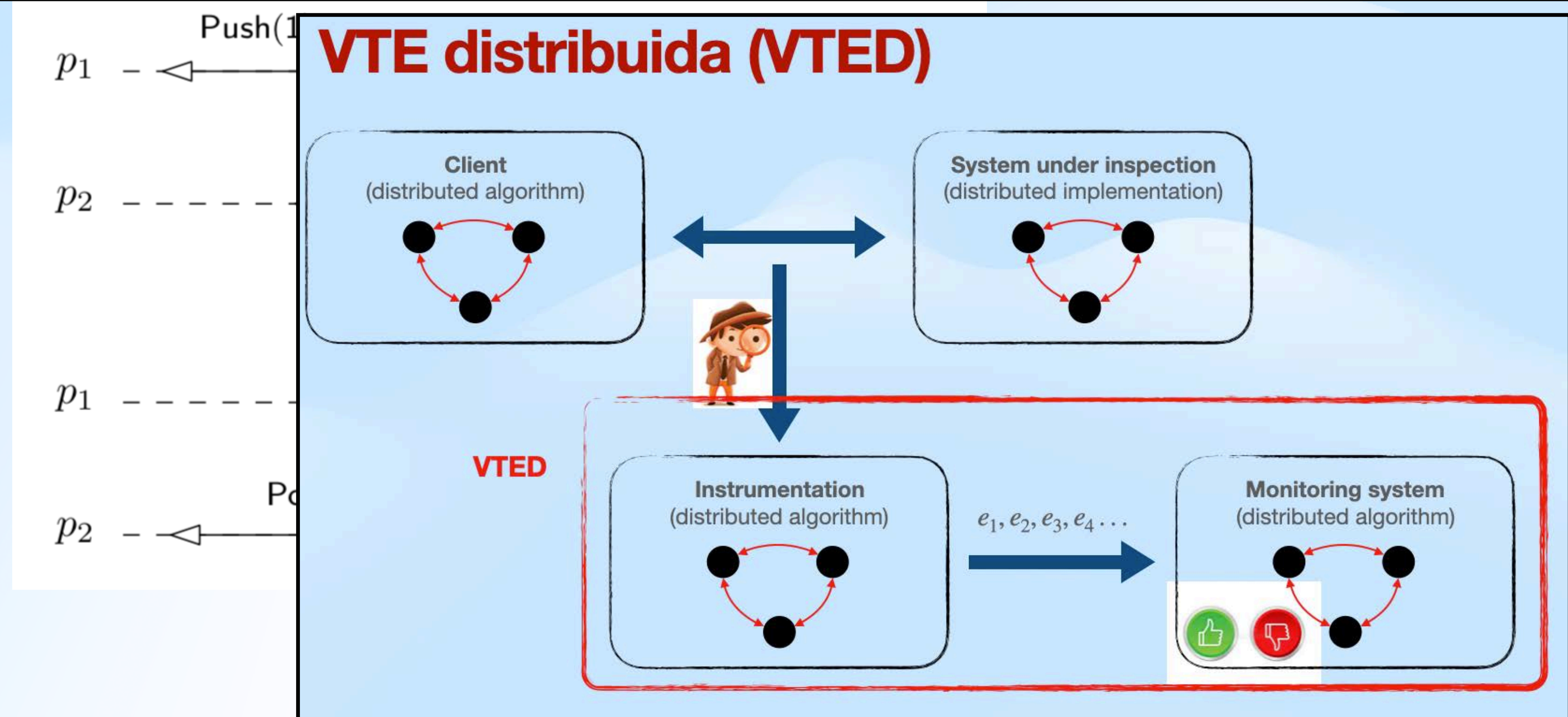
Resultados

Teorema: Si una propiedad es sensible al orden, entonces *no existe* definición de verificación bajo la cual la propiedad es verificable.



Resultados

Teorema: Si una propiedad es sensible al orden, entonces *no existe* definición de verificación bajo la cual la propiedad es verificable.



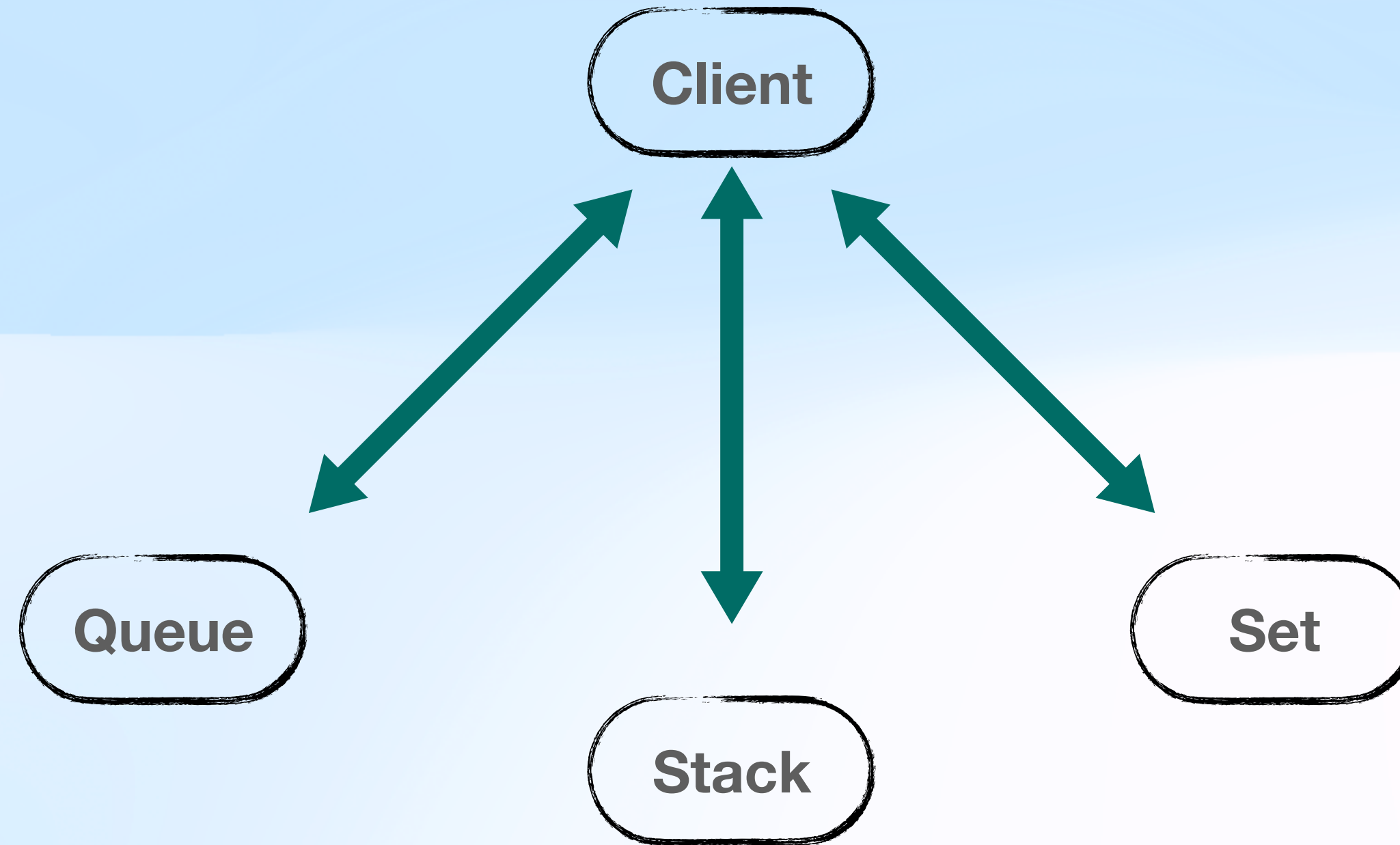
Resultados

Verificación predictiva:

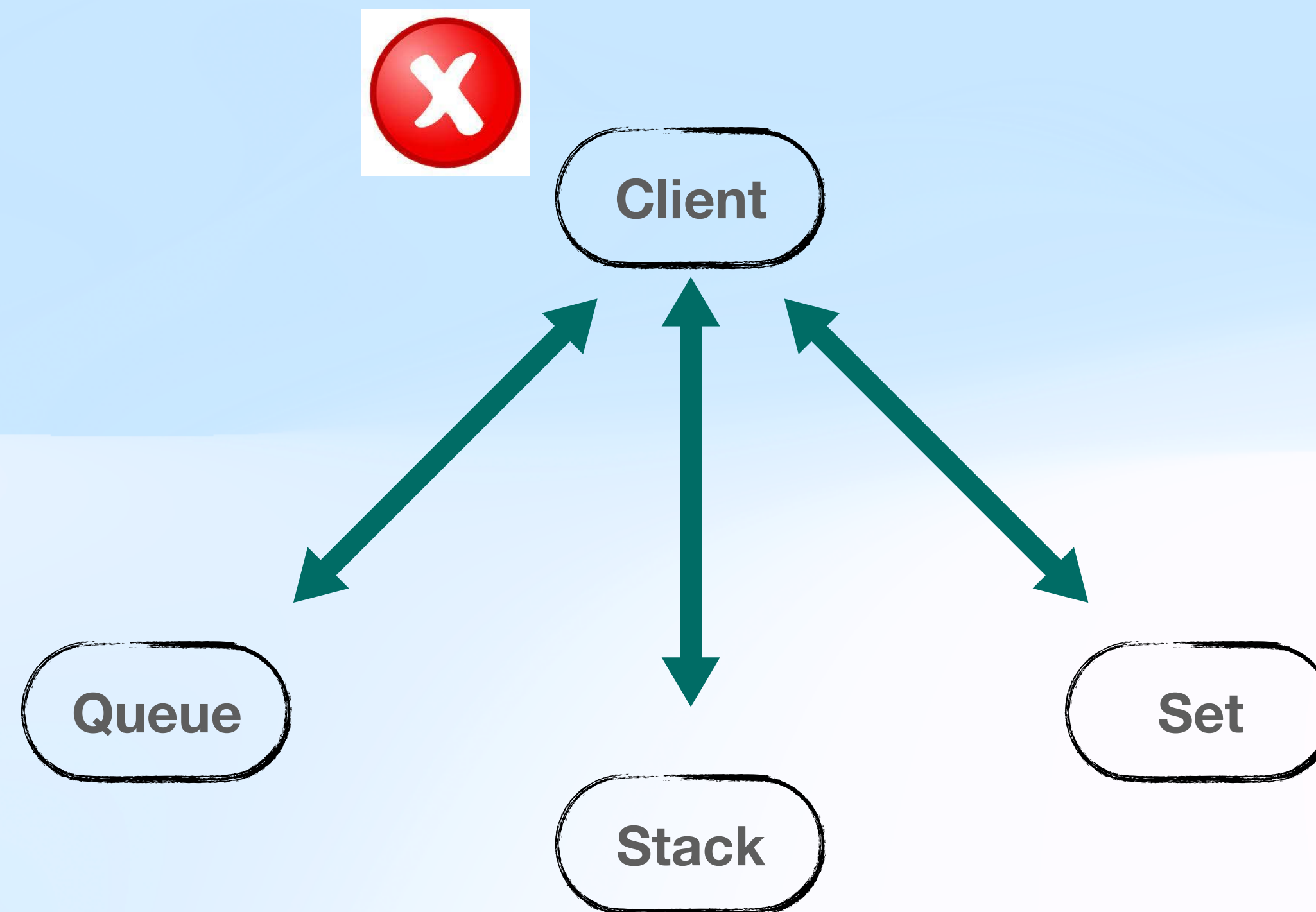
- Todo comportamiento incorrecto es detectado, es decir, **no hay falsos positivos**.
- **Permite falsos negativos.**
Debe poder concluirse que el sistema bajo inspección es incorrecto (análisis predictivo).

D. Language \ Model	Adversary A^τ	
	Decidability	
	P. Strong	P. Weak
<i>Linearizable register</i>	✓	✓
<i>Seq. consistent register</i>	✓	✓
<i>Linearizable ledger</i>	✓	✓
<i>Seq. consistent ledger</i>	✓	✓
<i>Event. consistent ledger</i>	✗	✗
<i>Weak event. counter</i>	✗	✓
<i>Strong event. counter</i>	✗	✓

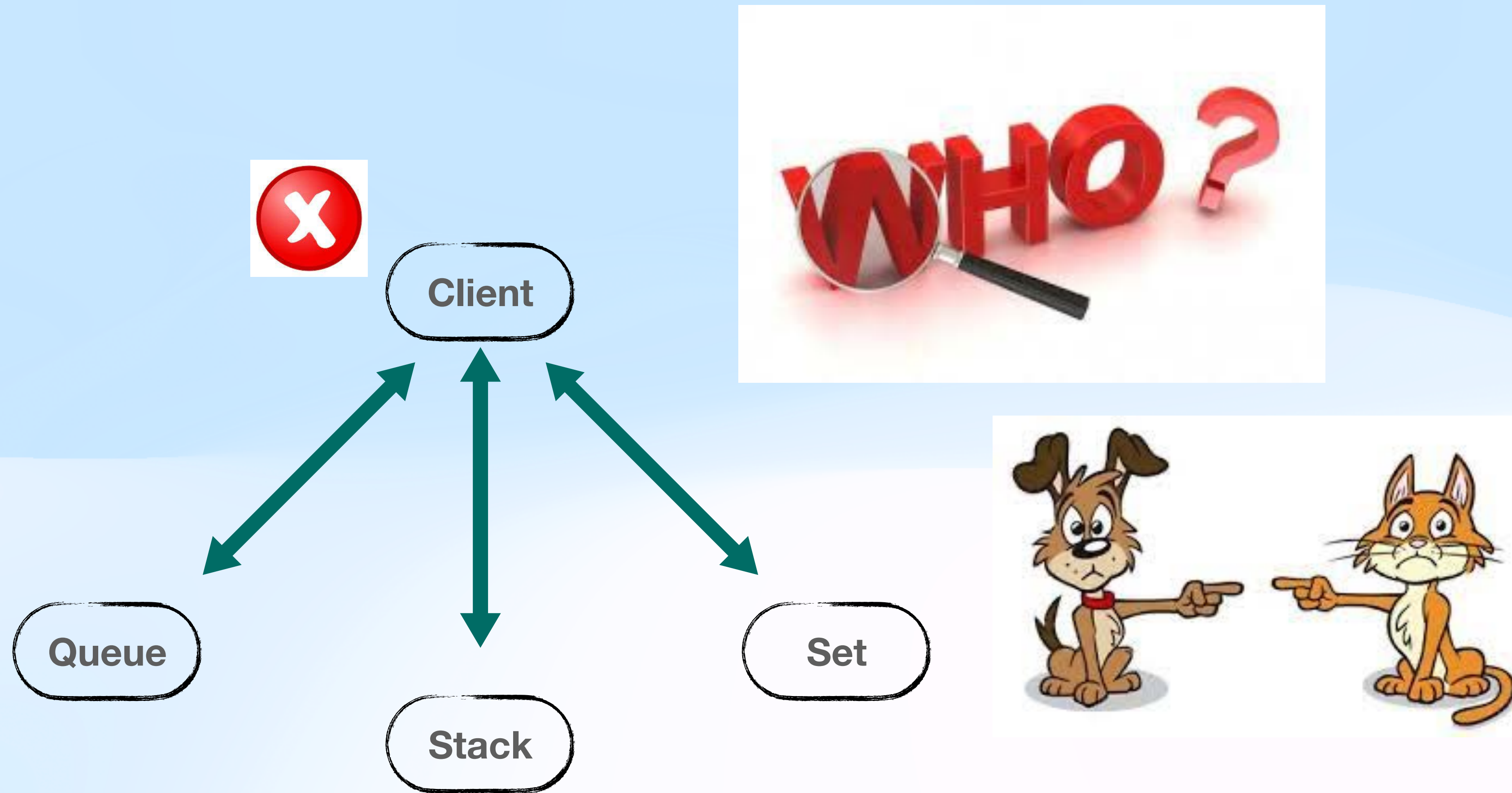
Rendición de cuentas (accountability)



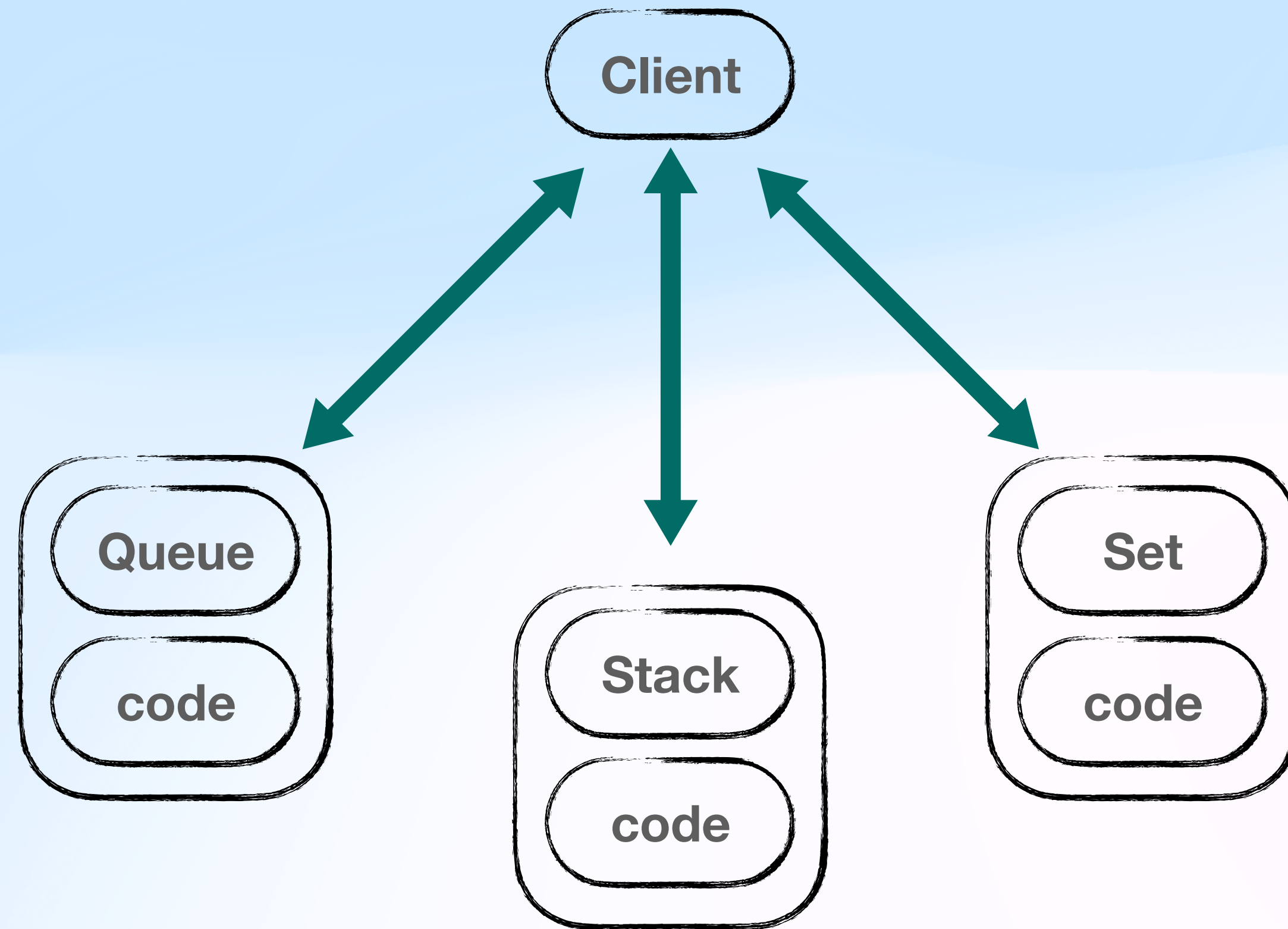
Rendición de cuentas (accountability)



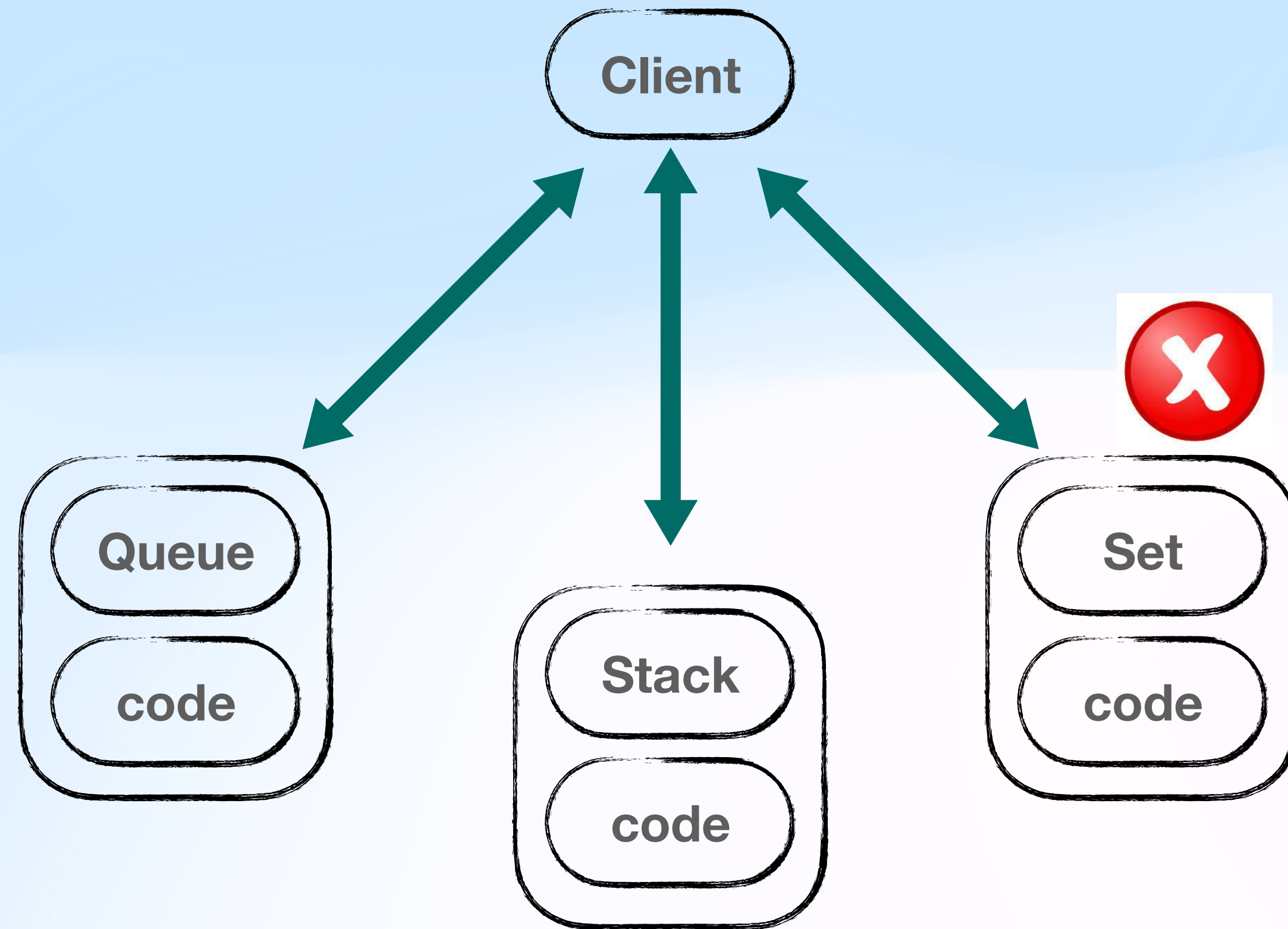
Rendición de cuentas (accountability)



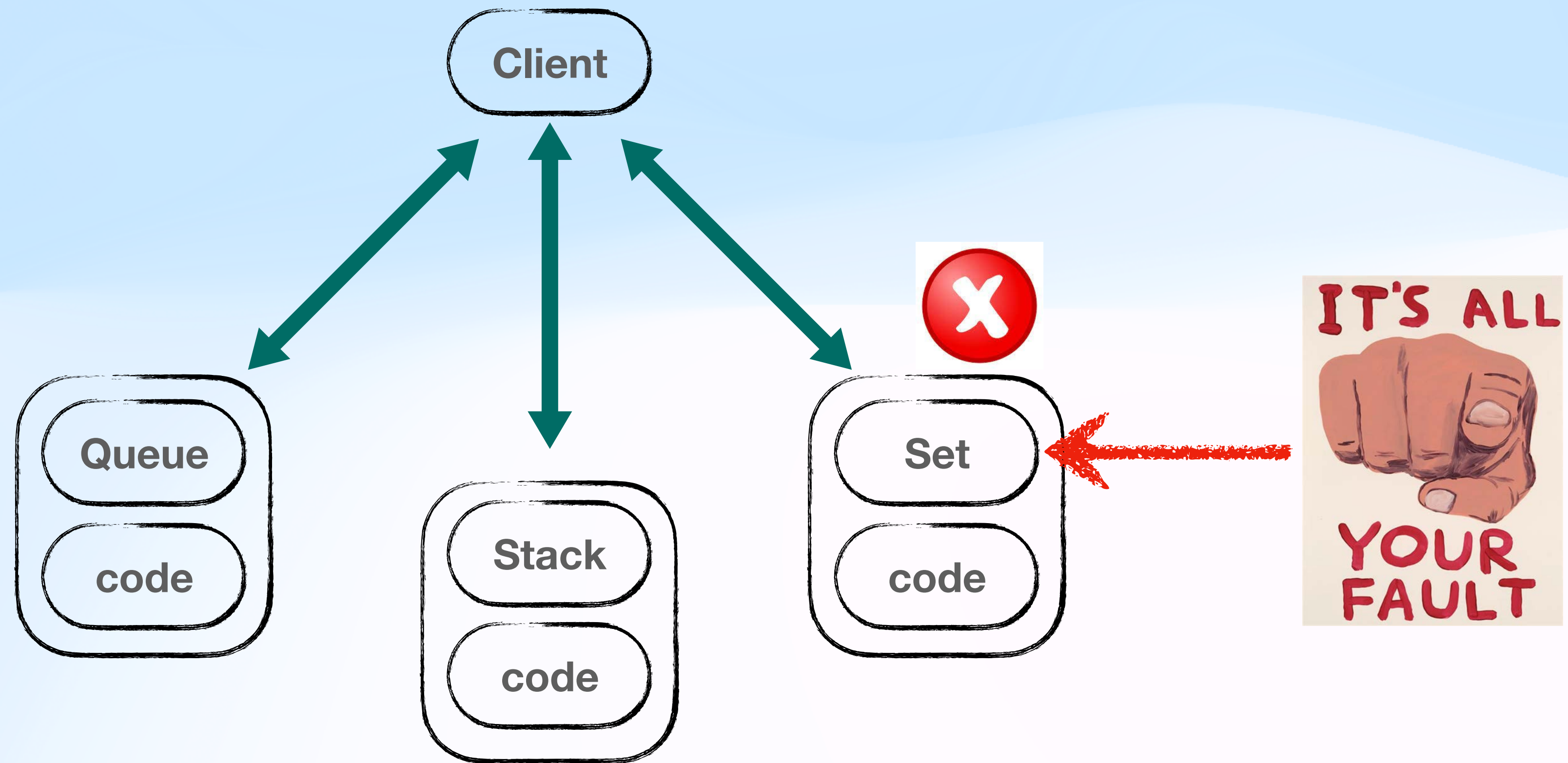
Rendición de cuentas (accountability)



Rendición de cuentas (accountability)



Rendición de cuentas (accountability)



AI Accountability Review

LLMs Can't Provide Faithful Explanations Needed for AI Accountability

NICK DIAKOPOULOS
MAR 24, 2026

Traceability and Accountability in Role-Specialized Multi-Agent LLM Pipelines

Amine Barrak


Department of Computer Science and Engineering
Oakland University, Rochester, MI, USA
aminebarrak@oakland.edu

CAMBRIDGE UNIVERSITY PRESS

Discover Content Products and Services

Cambridge Core | Browse Services Open research


Home Journals Data & Policy Volume 6 AI as a constituted system: accountability lessons...



AI as a constituted system: accountability lessons from an LLM experiment

Part of: Data for Policy Proceedings 2024

Published online by Cambridge University Press: 25 November 2024

[Kelsie Nabben](#)  [Show author details](#)

Article Figures Comments Metrics

[Save PDF](#) [Share](#) [Cite](#)

Abstract


This study focuses on the practicalities of establishing and maintaining AI infrastructure, as well as the considerations for responsible governance by investigating the integration of a pre-trained large language model (LLM) with an organisation's knowledge management system via a chat interface. The research adopts the concept of "AI as a constituted system" to emphasise the social, technical, and institutional factors that contribute to AI's governance and accountability. Through an ethnographic approach, this article details the iterative processes of negotiation, decision-making, and reflection among organisational stakeholders as they

Abstract
Policy Significance Statement
Introduction
Part 1: KMS-GPT: a "well constituted knowledge object"
Part 2: governance considerations
Part 3: Resurrection

npj | artificial intelligence


Explore content About the journal Publish with us

nature > npj artificial intelligence > comment > article

 npj Artificial Intelligence has APC waivers available that can be allocated upon acceptance on an *ad-hoc* basis. For more information, contact the Publisher, [Ronghua Guo](#).

Comment | [Open access](#) | Published: 05 November 2025

We need accountability in human–AI agent relationships

[Benjamin Lange](#) , [Geoff Keeling](#), [Arianna Manzini](#) & [Amanda McCroskery](#)

[npj Artificial Intelligence](#) 1, Article number: 38 (2025) | [Cite this article](#)

5790 Accesses | 3 Citations | 5 Altmetric | [Metrics](#)

We argue that accountability mechanisms are needed in human-AI agent relationships to ensure alignment with user and societal interests. We propose a framework according to which AI agents' engagement is conditional on appropriate user behaviour. The framework incorporates design-strategies such as distancing, disengaging, and discouraging.

Responsabilidad de la IA

Definición e introducción

La responsabilidad de la IA se refiere a la idea de que la inteligencia artificial debe desarrollarse, desplegarse y utilizarse de manera que la responsabilidad de los malos resultados pueda asignarse a las partes responsables. La tecnología basada en IA suele plantear problemas de rendición de cuentas debido a la opacidad y complejidad de los sistemas de aprendizaje automático y profundo, al número de partes interesadas que suelen participar en la creación e implementación de productos de IA y al potencial de aprendizaje dinámico de la tecnología.

A menudo se critica a los sistemas de IA por ser una "caja negra", lo que significa que los usuarios no pueden explicar o interpretar completamente el proceso que hay detrás de un resultado. Si la toma de decisiones de la IA no puede explicarse o entenderse, resulta muy difícil asignar responsabilidades y hacer que las partes rindan cuentas por los resultados perjudiciales.

Para más información sobre la responsabilidad de la IA, consulte los siguientes recursos.

Pérdida de transparencia, trazabilidad y responsabilidad



behaviour. The framework incorporates design-strategies such as distancing, disengaging, and discouraging.

Responsabilidad de la IA

Definición e introducción

La responsabilidad de la IA se refiere a la idea de que la inteligencia artificial debe desarrollarse, desplegarse y utilizarse de manera que la responsabilidad de los malos resultados pueda asignarse a las partes responsables. La tecnología basada en IA suele plantear problemas de rendición de cuentas debido a la opacidad y complejidad de los sistemas de aprendizaje automático y profundo, al número de partes interesadas que suelen participar en la creación e implementación de productos de IA y al potencial de aprendizaje dinámico de la tecnología.

A menudo se critica a los sistemas de IA por ser una "caja negra", lo que significa que los usuarios no pueden explicar o interpretar completamente el proceso que hay detrás de un resultado. Si la toma de decisiones de la IA no puede explicarse o entenderse, resulta muy difícil asignar responsabilidades y hacer que las partes rindan cuentas por los resultados perjudiciales.

Para más información sobre la responsabilidad de la IA, consulte los siguientes recursos.

Pérdida de transparencia, trazabilidad y responsabilidad



behaviour. The framework incorporates design-strategies such as distancing, disengaging, and discouraging.

Responsabilidad de la IA

Definición e introducción

La responsabilidad de la IA se refiere a la idea de que la inteligencia artificial debe desarrollarse, desplegarse y utilizarse de manera que la responsabilidad de los malos resultados pueda asignarse a las partes responsables. La tecnología basada en IA suele plantear problemas de rendición de cuentas debido a la opacidad y complejidad de los sistemas de aprendizaje automático y profundo, al número de partes interesadas que suelen participar en la creación e implementación de productos de IA y al potencial de aprendizaje dinámico de la tecnología.

A menudo se critica a los sistemas de IA por ser una "caja negra", lo que significa que los usuarios no pueden explicar o interpretar completamente el proceso que hay detrás de un resultado. Si la toma de decisiones de la IA no puede explicarse o entenderse, resulta muy difícil asignar responsabilidades y hacer que las partes rindan cuentas por los resultados perjudiciales.

Para más información sobre la responsabilidad de la IA, consulte los siguientes recursos.

Pérdida de transparencia, trazabilidad y responsabilidad



behaviour. The framework incorporates design-strategies such as distancing, disengaging, and discouraging.

Responsabilidad de la IA

Definición e introducción

La responsabilidad de la IA se refiere a la idea de que la inteligencia artificial debe desarrollarse, desplegarse y utilizarse de manera que la responsabilidad de los malos resultados pueda asignarse a las partes responsables. La tecnología basada en IA suele plantear problemas de rendición de cuentas debido a la opacidad y complejidad de los sistemas de aprendizaje automático y profundo, al número de partes interesadas que suelen participar en la creación e implementación de productos de IA y al potencial de aprendizaje dinámico de la tecnología.

A menudo se critica a los sistemas de IA por ser una "caja negra", lo que significa que los usuarios no pueden explicar o interpretar completamente el proceso que hay detrás de un resultado. Si la toma de decisiones de la IA no puede explicarse o entenderse, resulta muy difícil asignar responsabilidades y hacer que las partes rindan cuentas por los resultados perjudiciales.

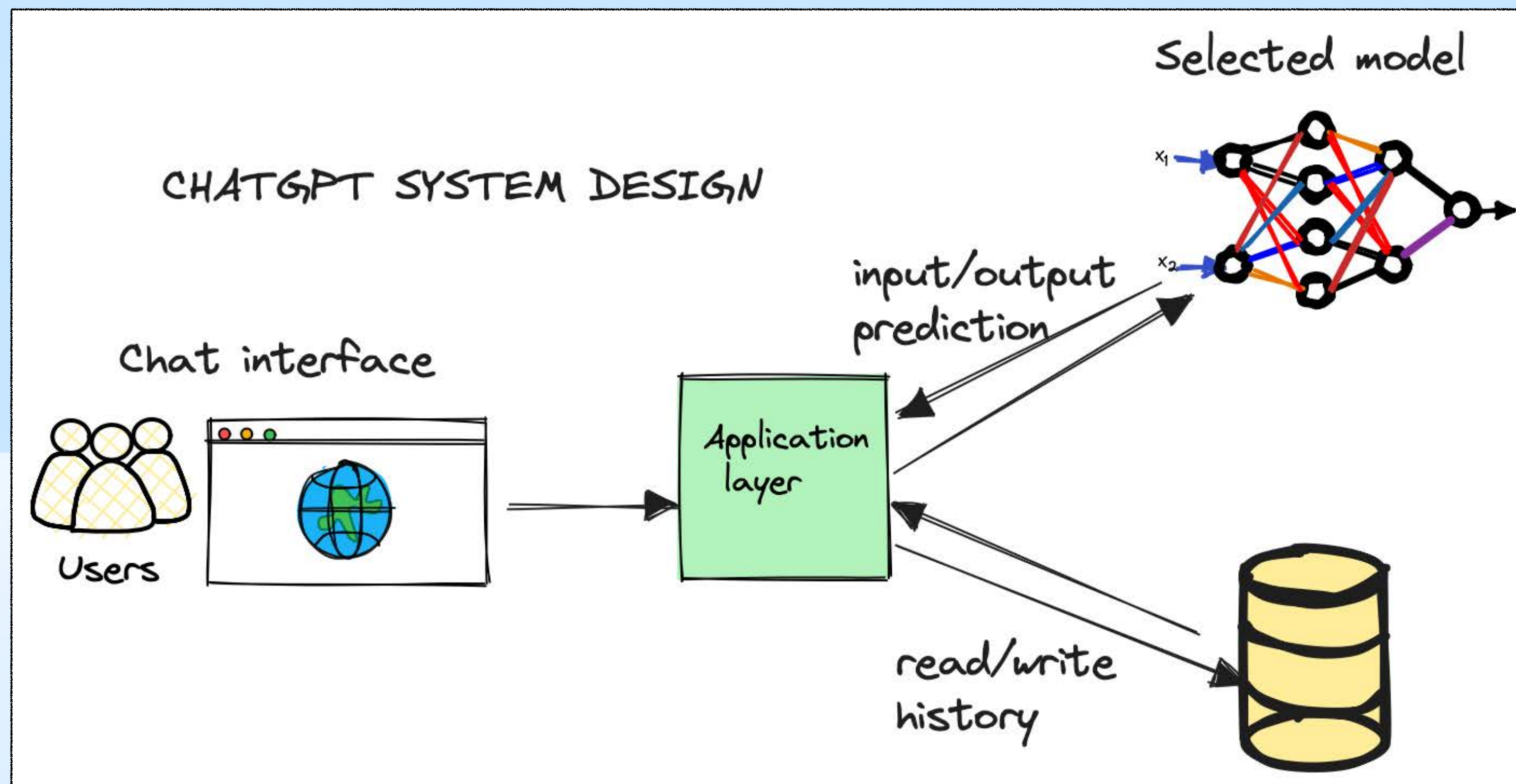
Para más información sobre la responsabilidad de la IA, consulte los siguientes recursos.

Pérdida de transparencia, trazabilidad y responsabilidad

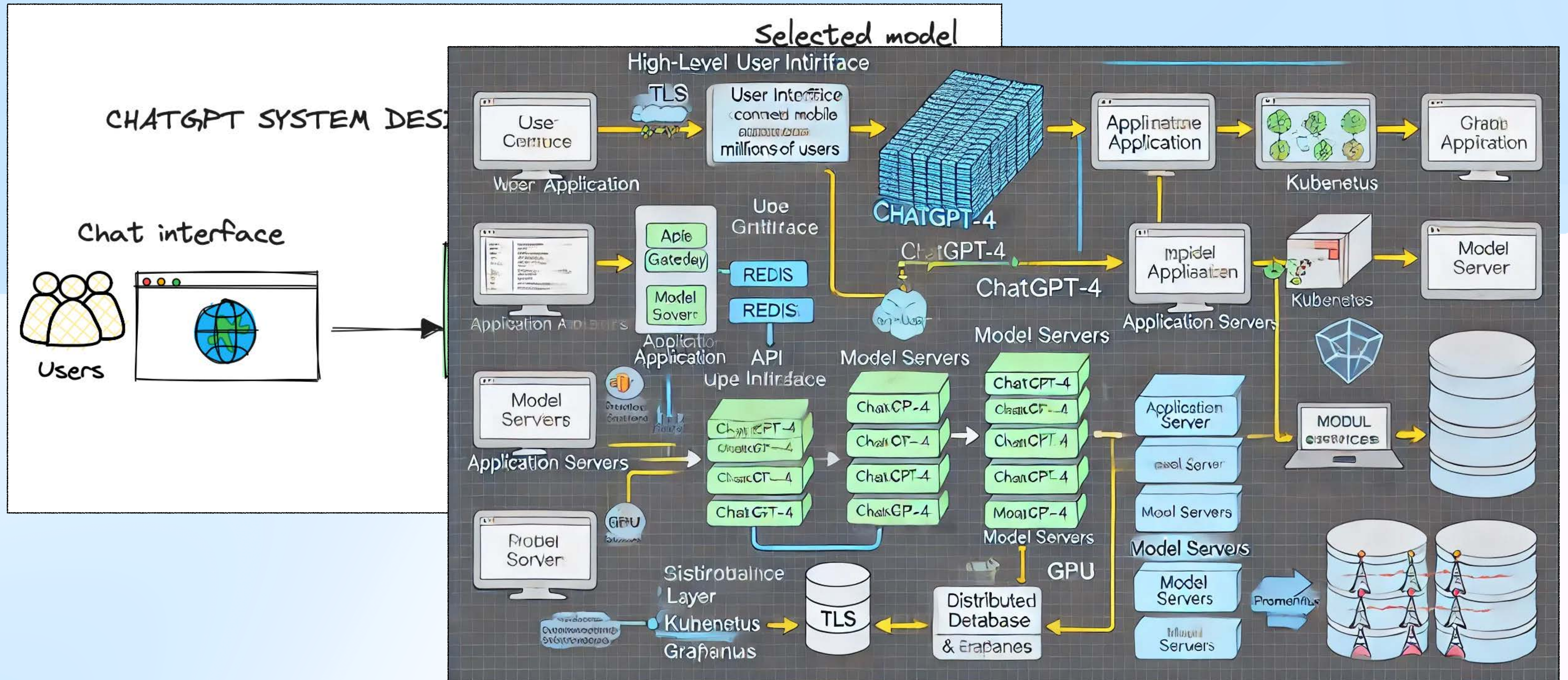


behaviour. The framework incorporates design-strategies such as distancing, disengaging, and discouraging.

LLM son sistemas distribuidos



LLM son sistemas distribuidos



Implicaciones?

- El modelo propuesto captura (tal vez parcialmente) sistemas IA actuales?
- Resultados implican que la responsabilidad/rendición de cuentas en general es imposible en IA?
- Las técnicas propuestas son aplicables a la IA actuales? Tal vez en componentes específicos?

