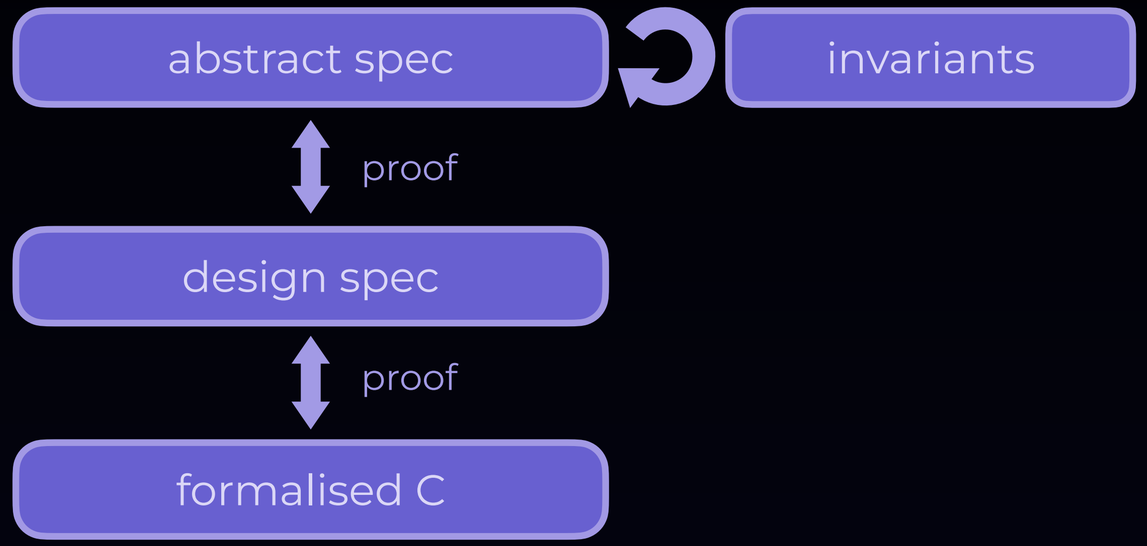

# seL4's Security Proofs

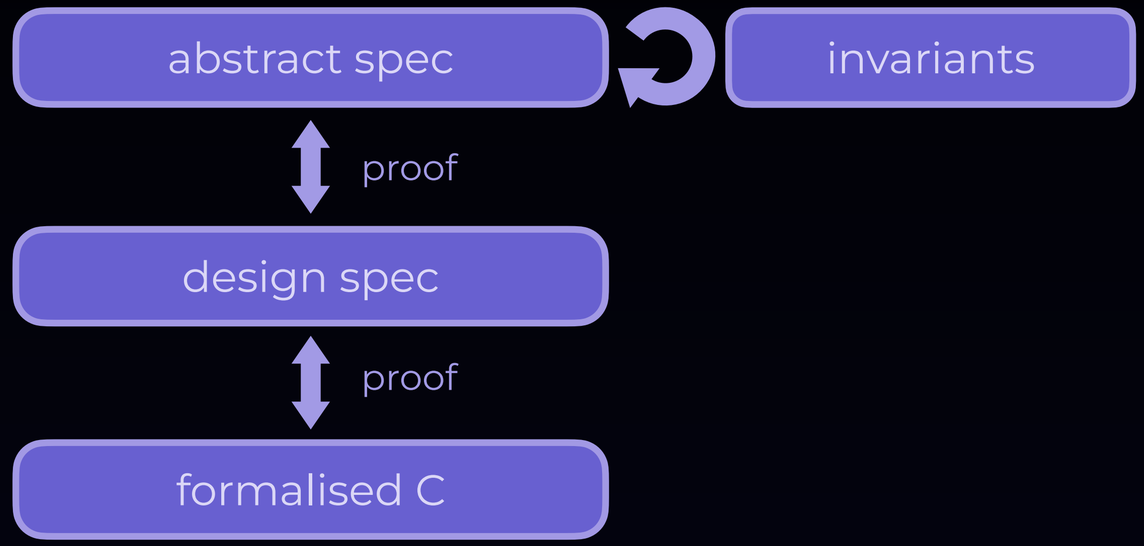
Ryan Barry | seL4 summit 2026, Vancouver, Canada

# What is “correctness”?

# Verification stack

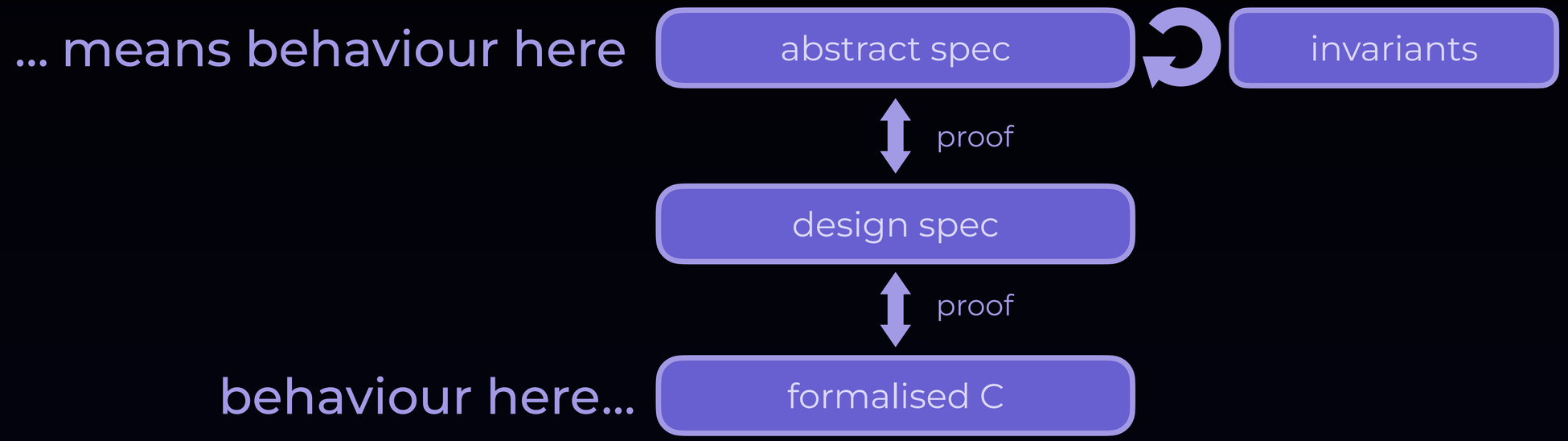


# Verification stack




behaviour here...

# Verification stack



# Assumptions

# Assumptions

- Every proof makes assumptions

# Assumptions

- Every proof makes assumptions
- Among others, seL4's assumptions include:

# Assumptions

- Every proof makes assumptions
- Among others, seL4's assumptions include:

**Hardware Assumptions**

We assume the hardware works as expected

# Assumptions

- Every proof makes assumptions
- Among others, seL4's assumptions include:

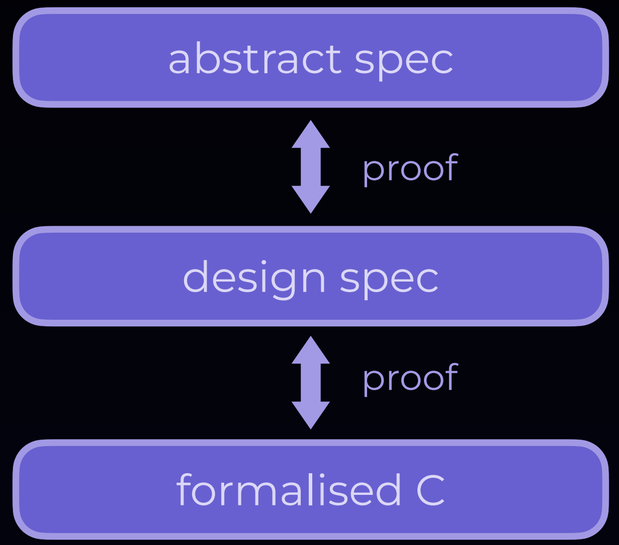
**Hardware Assumptions**

We assume the hardware works as expected

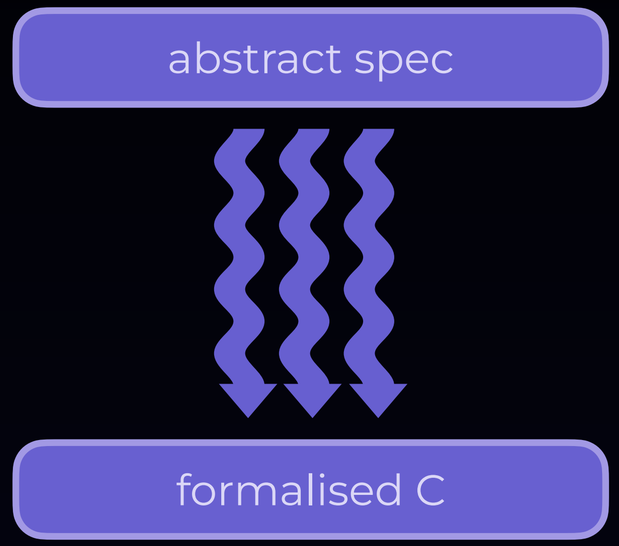
**Specification Assumptions**

We assume the specification says what we want it to

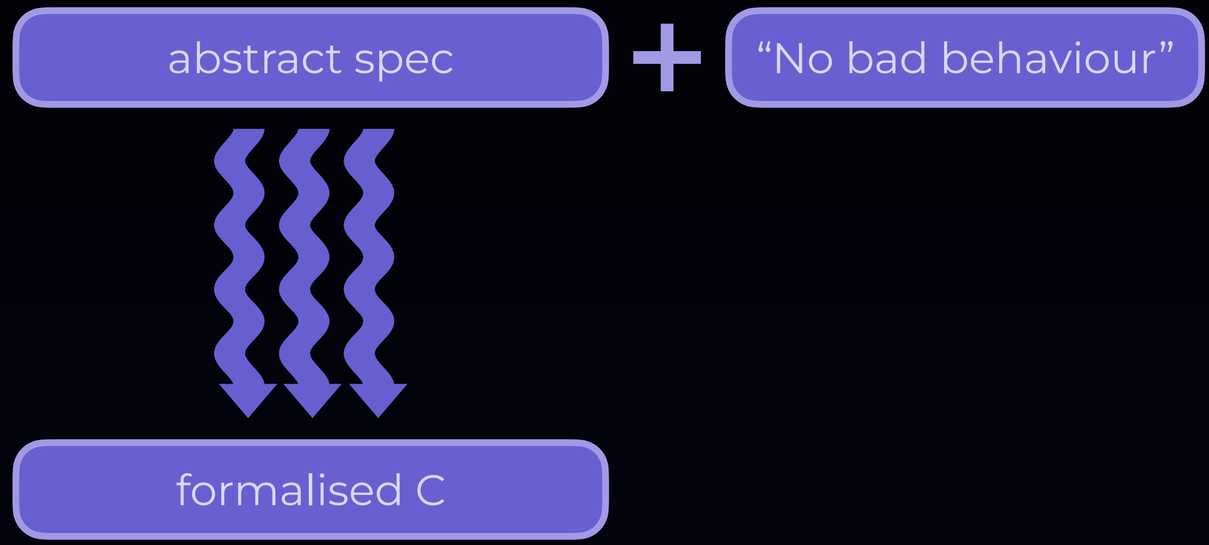
# Verification stack



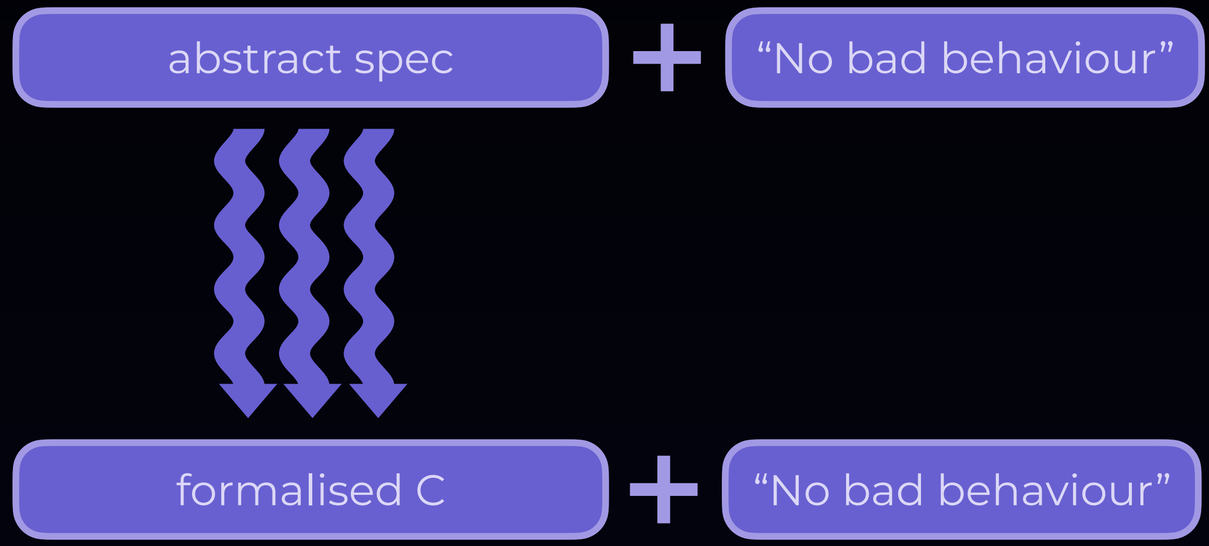
# Verification stack



# Verification stack



# Verification stack



# Security Properties

# CIA Triad

# CIA Triad

# CIA Triad

- Confidentiality
  - Can’t learn information without authorisation

# CIA Triad

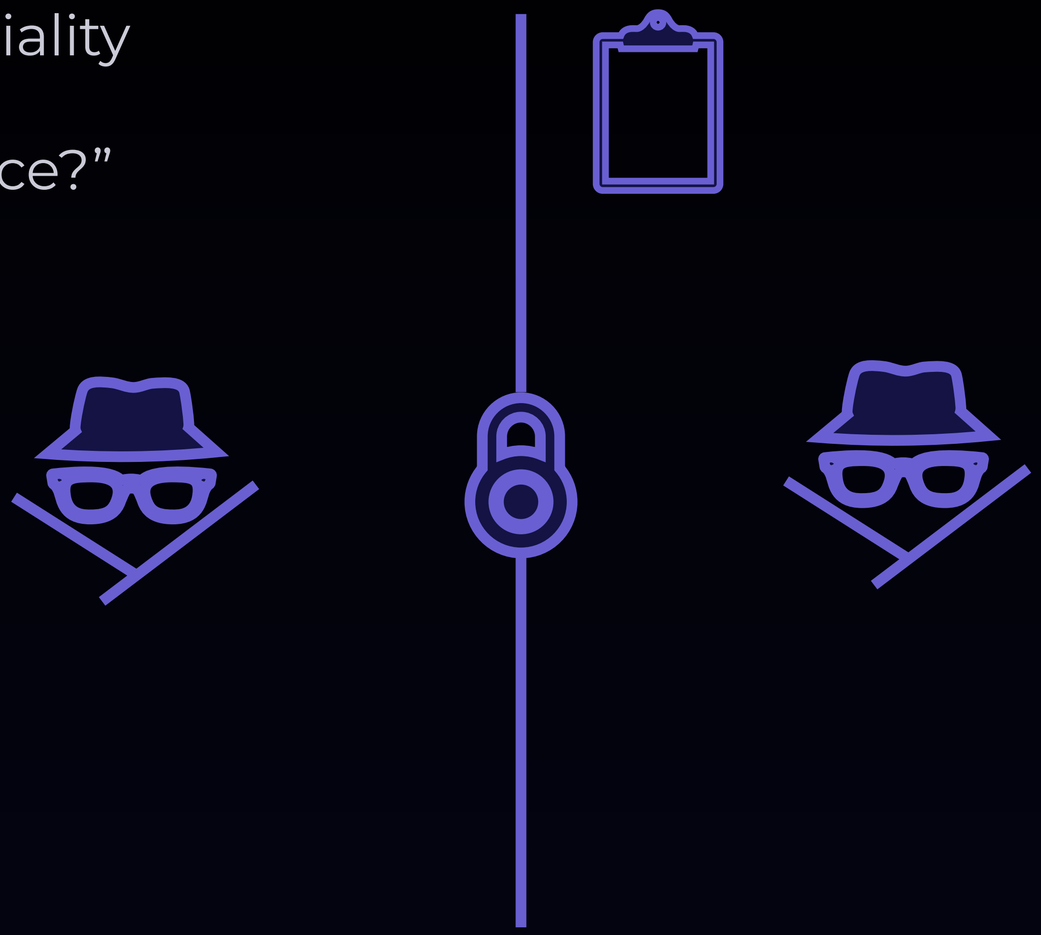
- Confidentiality
  - Can’t learn information without authorisation
- Integrity
  - Can’t modify data without authorisation

# CIA Triad

- Confidentiality
  - Can’t learn information without authorisation
- Integrity
  - Can’t modify data without authorisation
- Authority Confinement
  - Can’t escalate authority

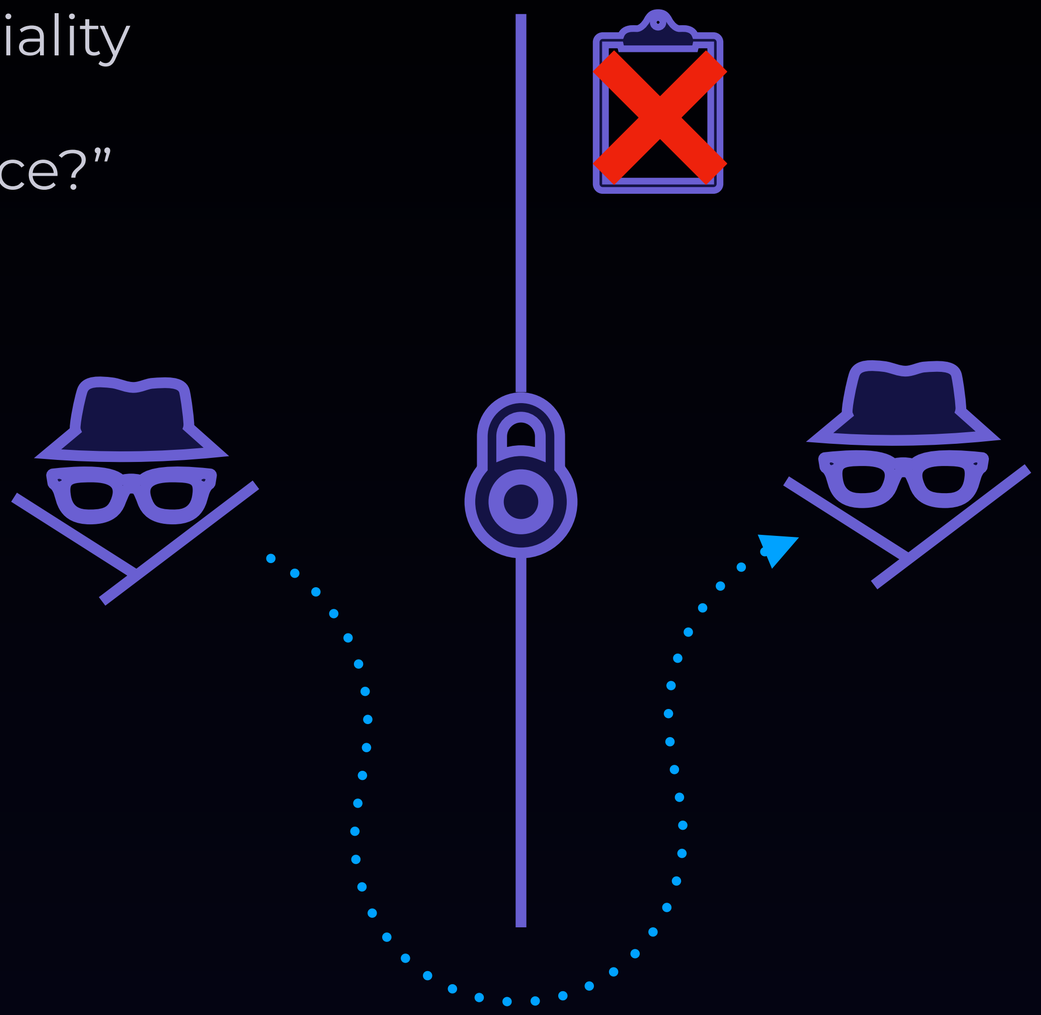
# Confidentiality

Breaking confidentiality
=
“Who can I influence?”



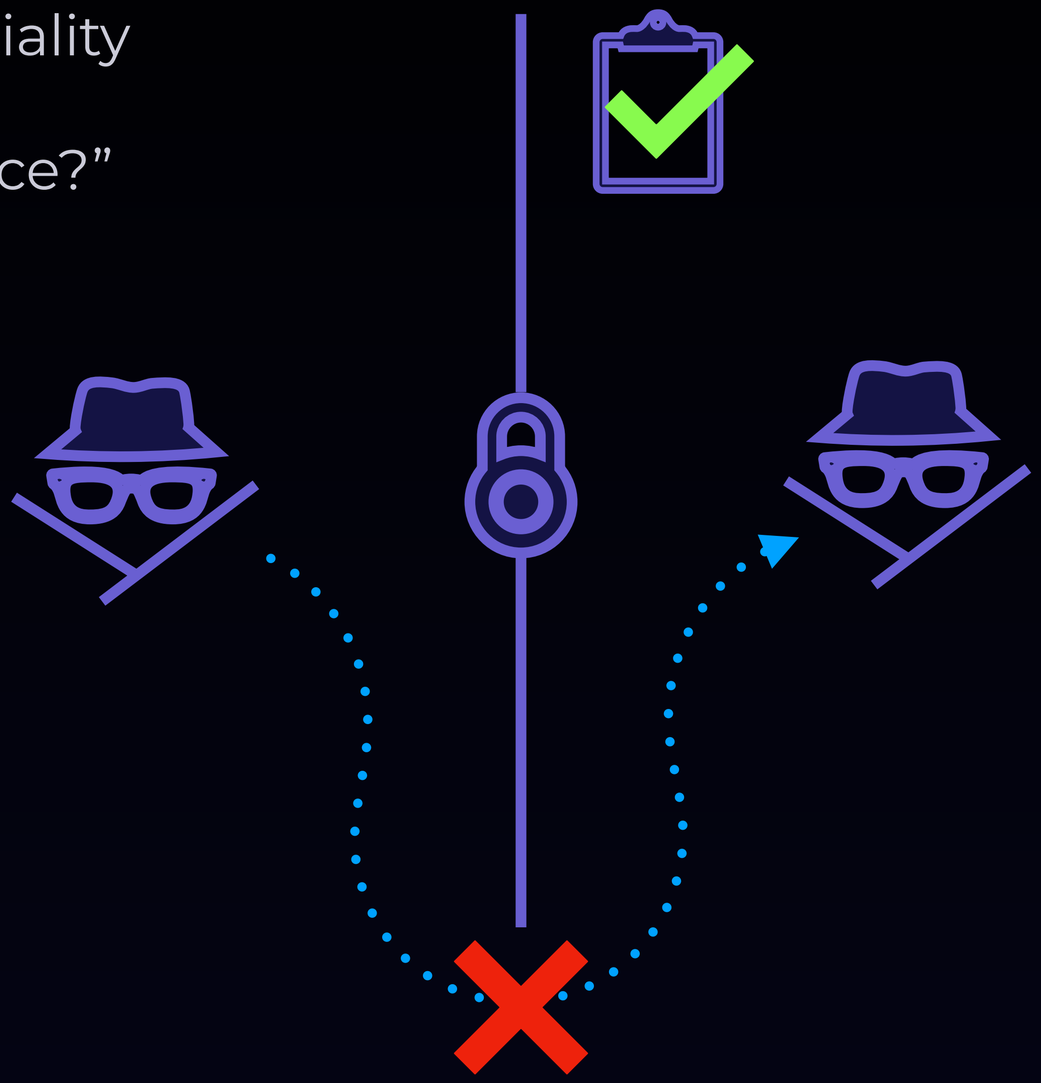
# Confidentiality

Breaking confidentiality
=
“Who can I influence?”



# Confidentiality

Breaking confidentiality
=
“Who can I influence?”

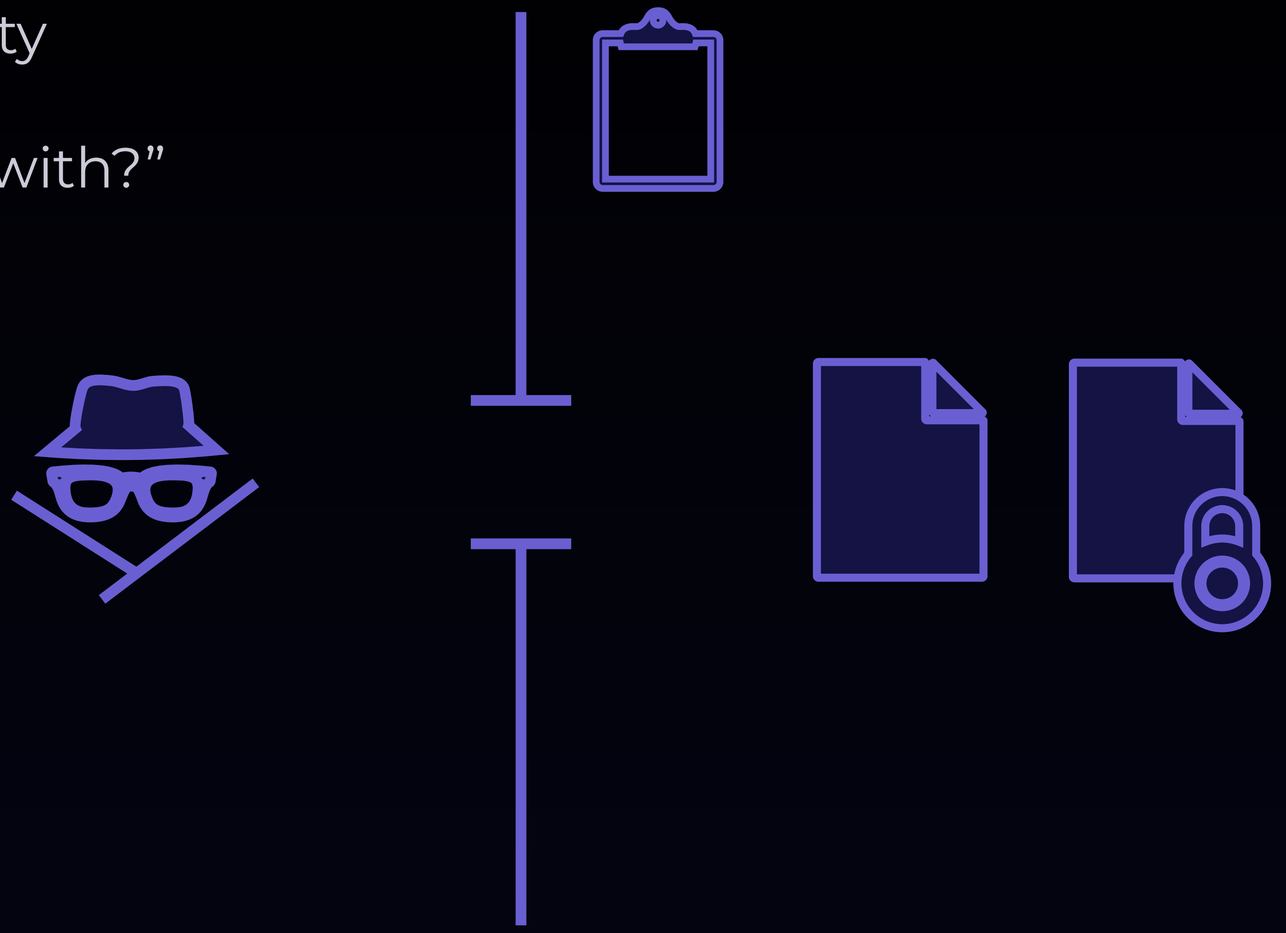


# Integrity

Breaking integrity
=
“What can I tamper with?”

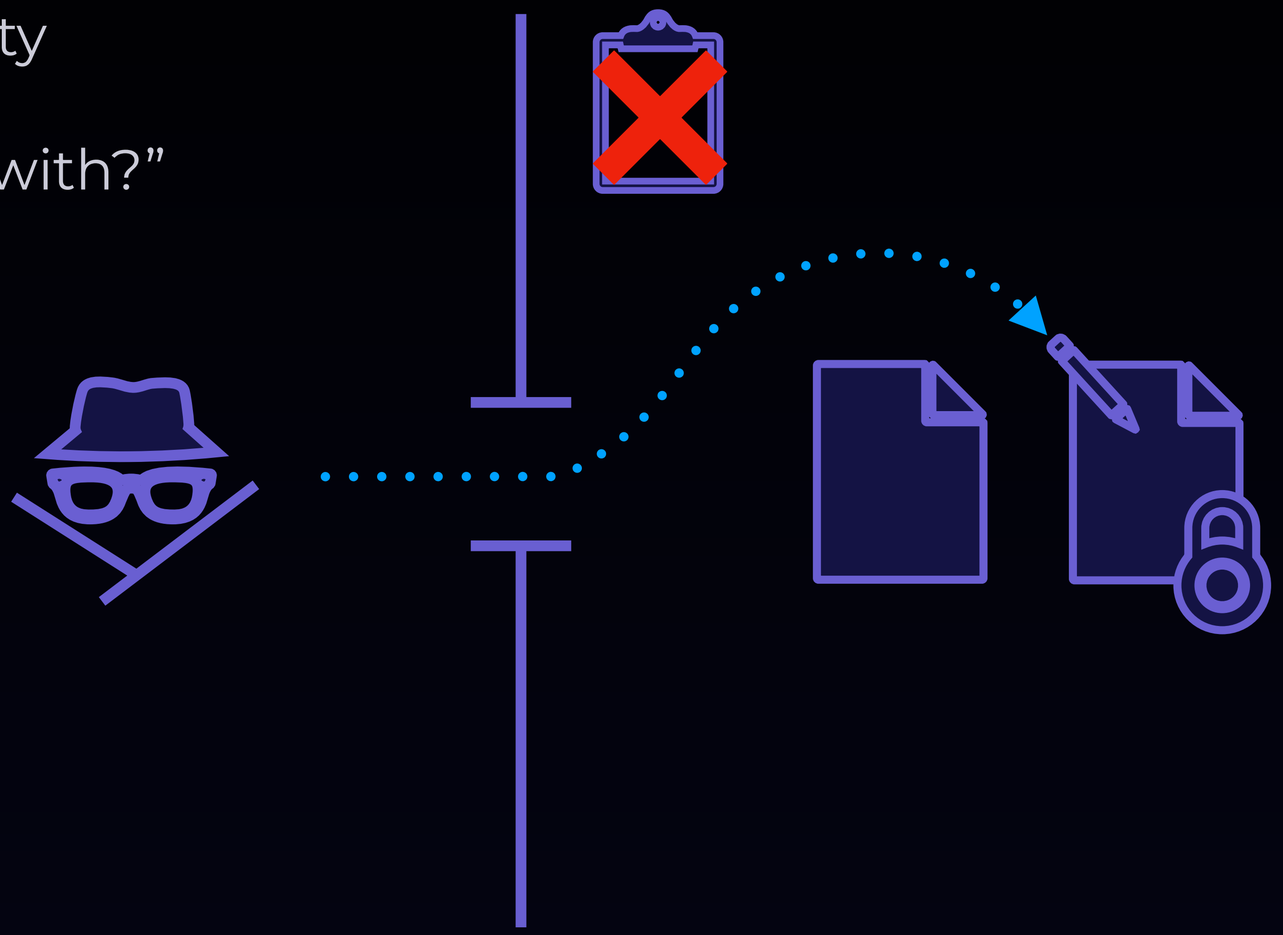
# Integrity

Breaking integrity
=
“What can I tamper with?”



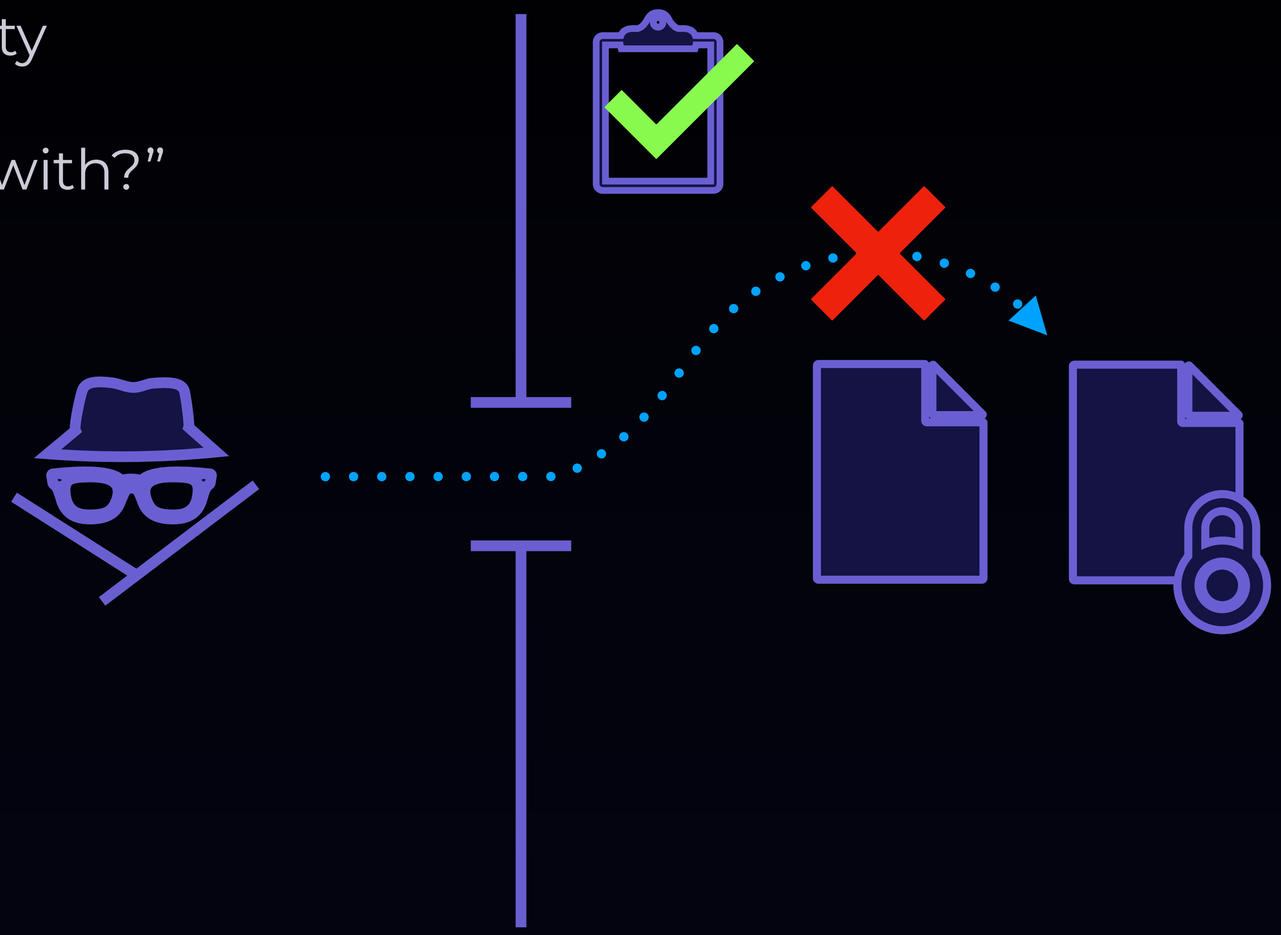
# Integrity

Breaking integrity
=
“What can I tamper with?”



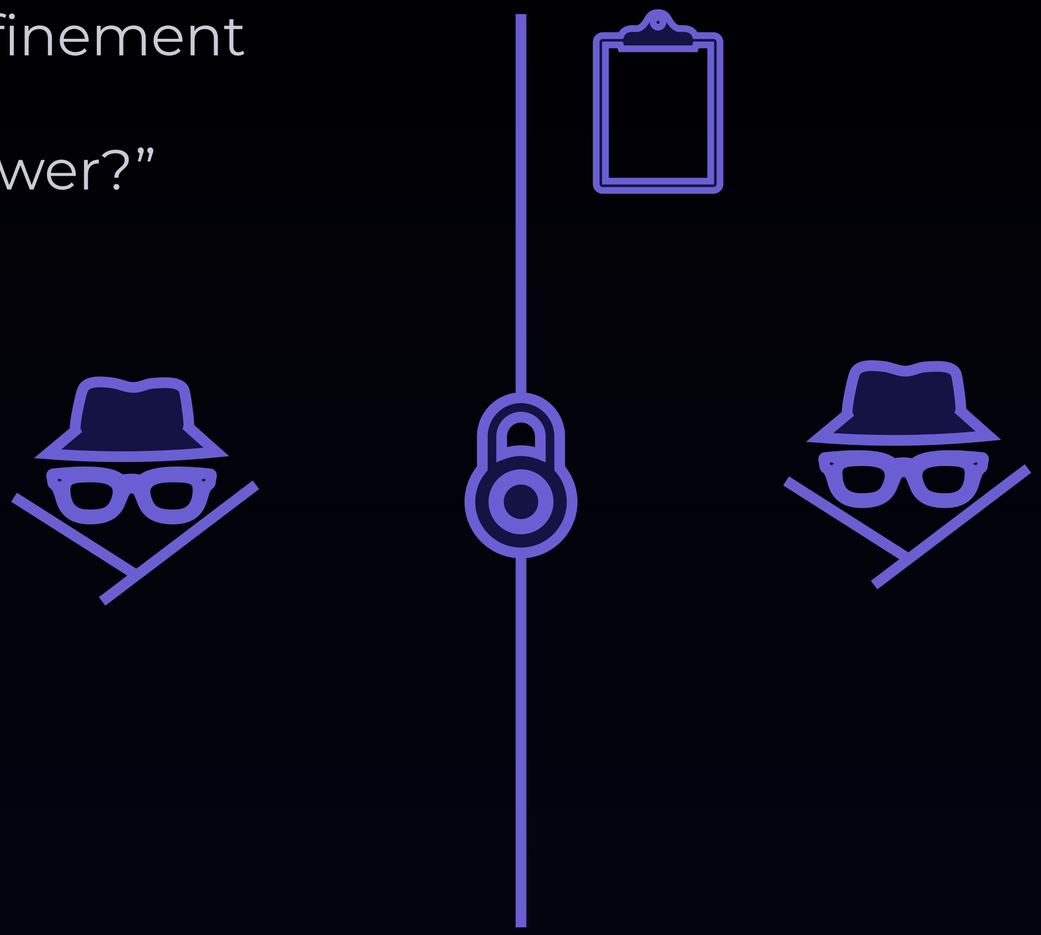
# Integrity

Breaking integrity
=
“What can I tamper with?”



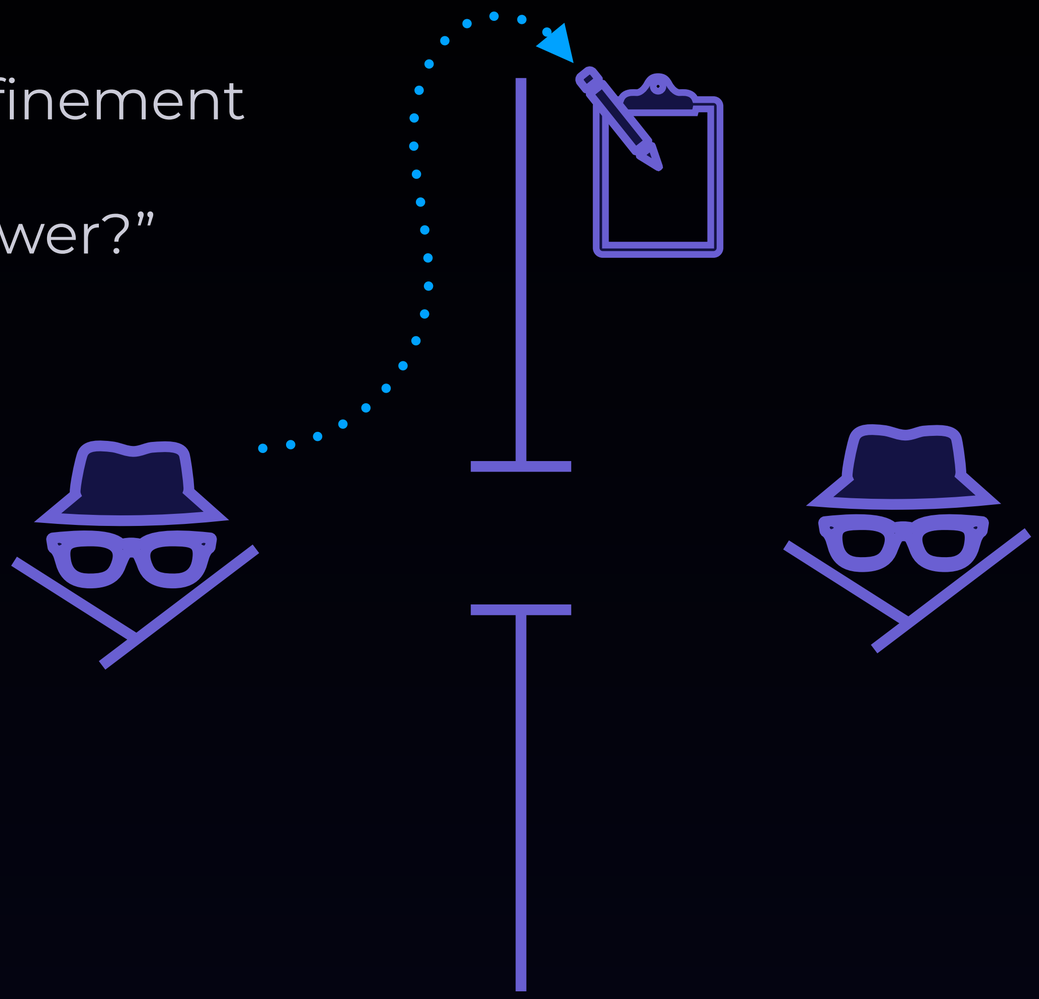
# Authority Confinement

Breaking authority confinement
=
“How can I usurp power?”



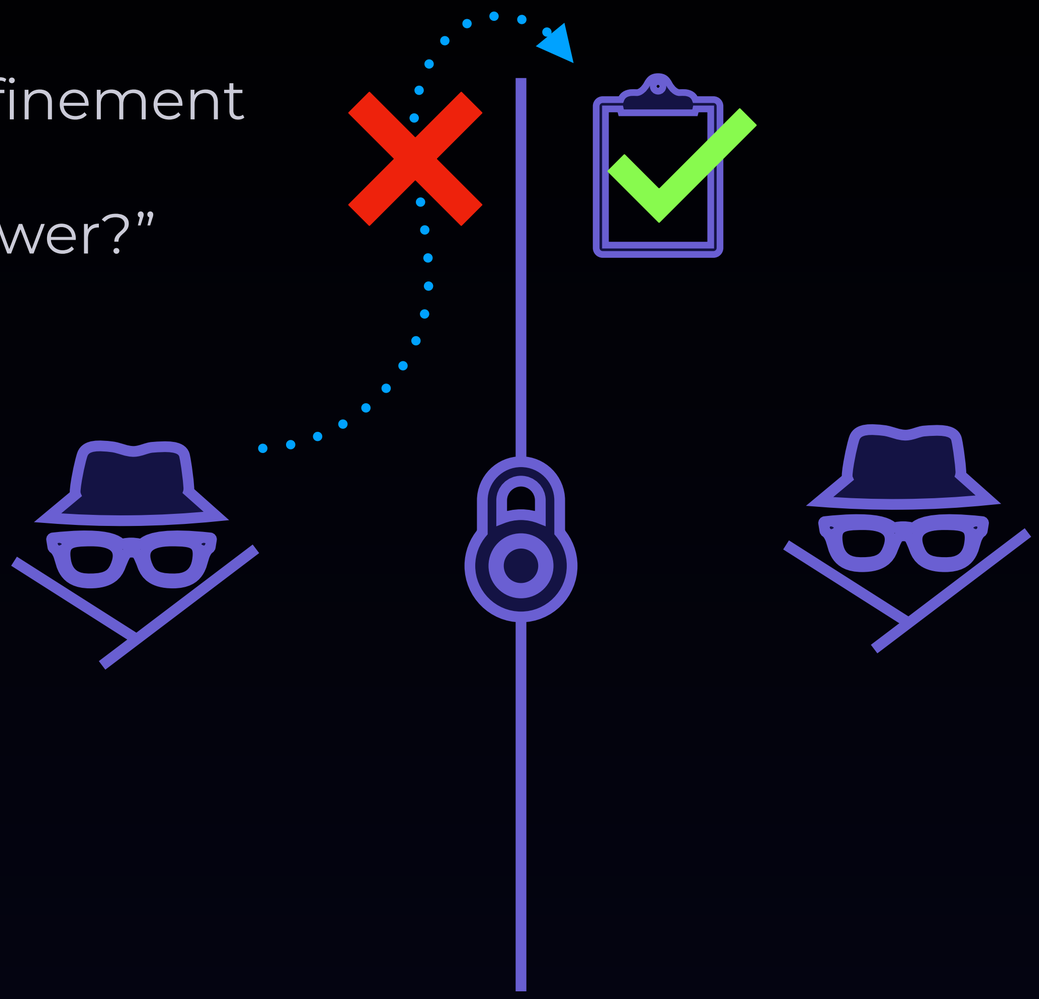
# Authority Confinement

Breaking authority confinement
=
“How can I usurp power?”



# Authority Confinement

Breaking authority confinement
=
“How can I usurp power?”



# PAS

Policy, Abstraction, Subject

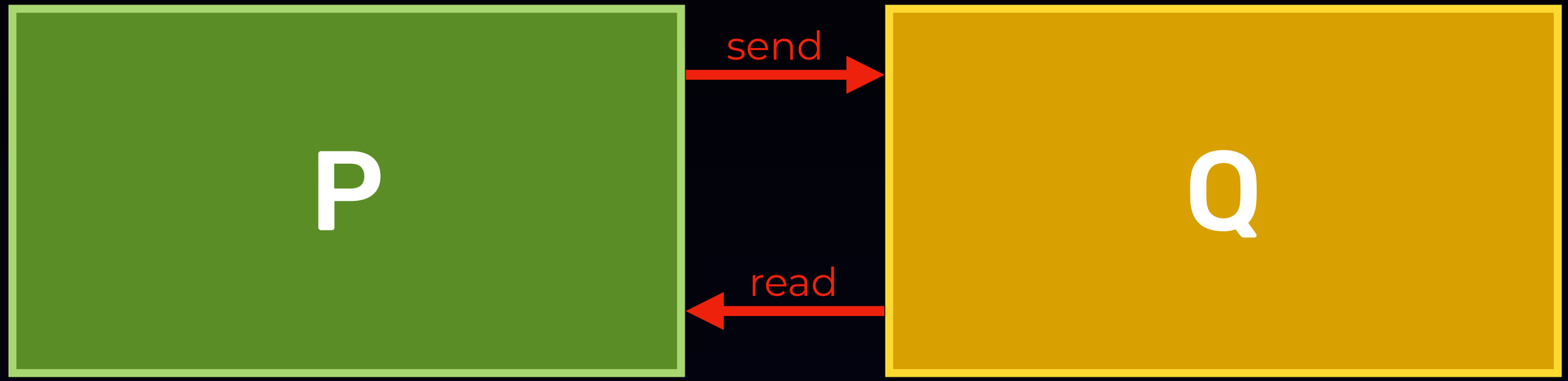
# Policy

# Policy

- Agent Authority Graph
  - Nodes are agents
  - Edges are authorities

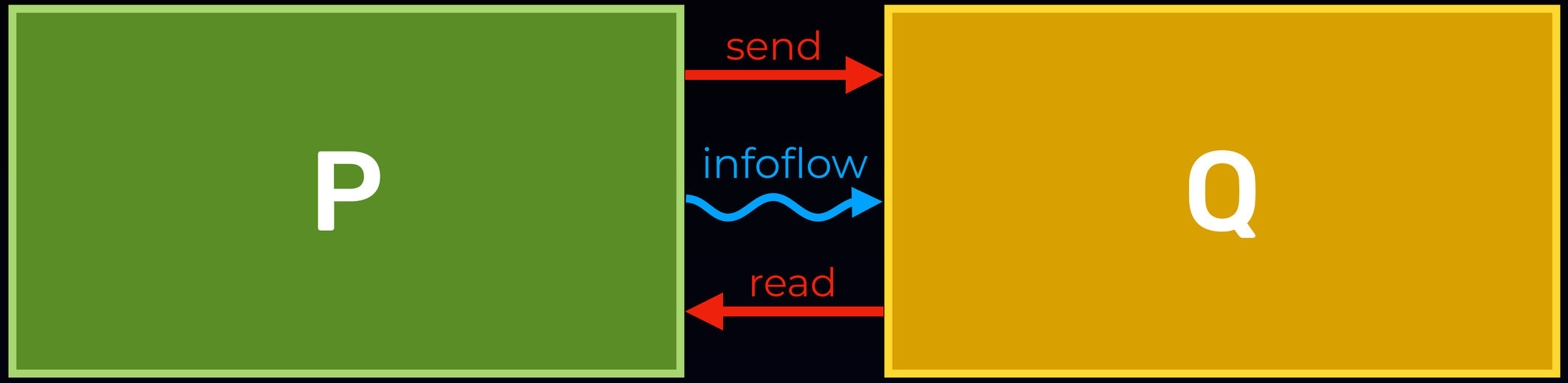
# Policy

- Agent Authority Graph
  - Nodes are agents
  - Edges are authorities

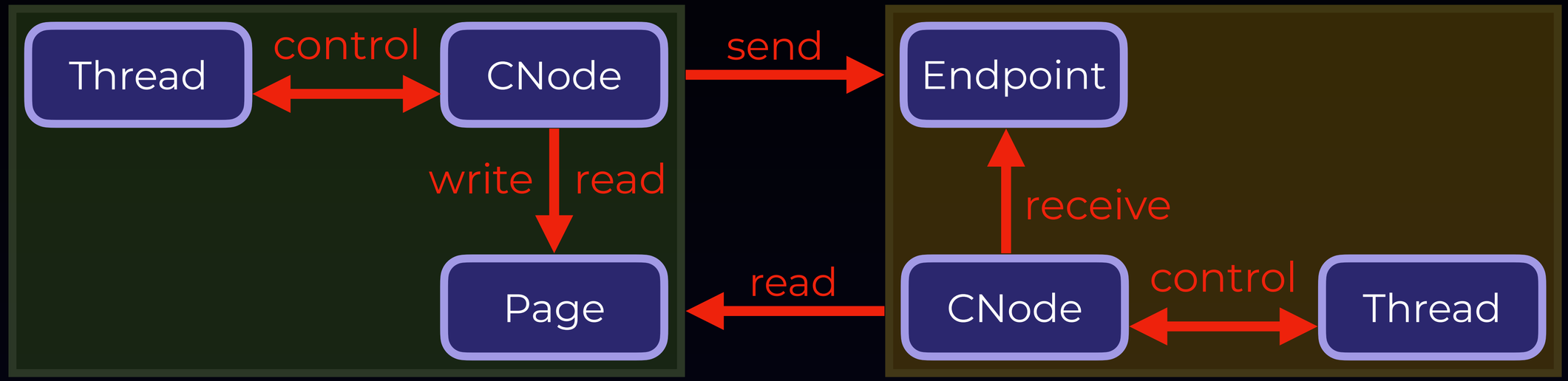


# Policy

- Agent Authority Graph
  - Nodes are agents
  - Edges are authorities

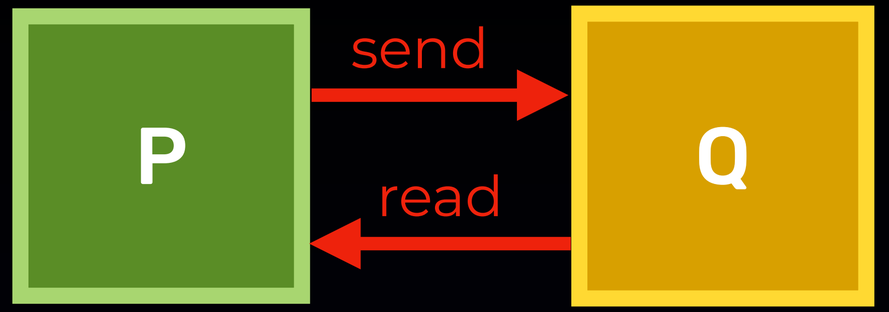


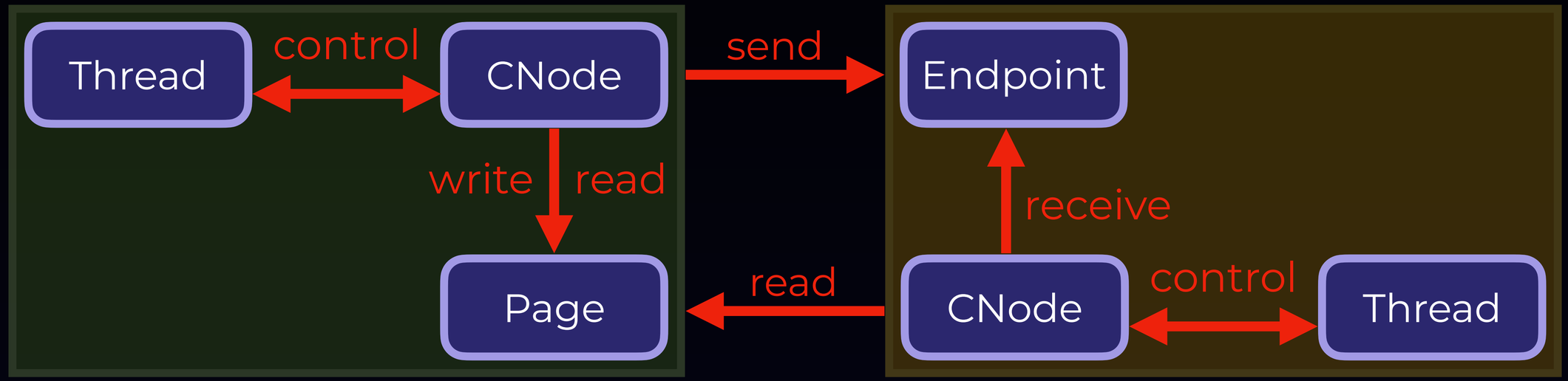
# Abstraction



# Abstraction

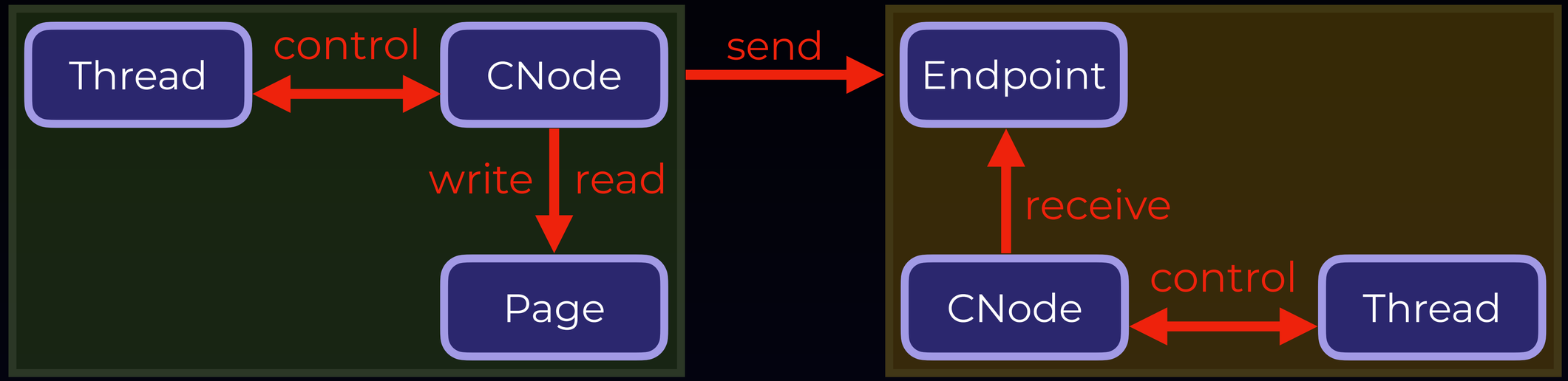
- Mapping memory to agents
  - Green maps to P
  - Yellow maps to Q



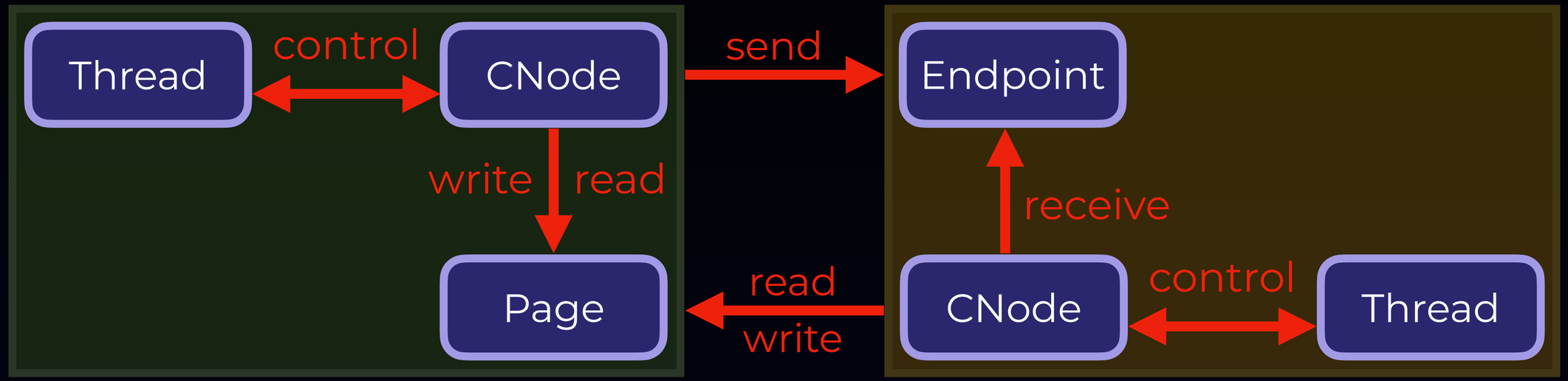
# Abstraction

- Mapping memory to agents
  - Green maps to P
  - Yellow maps to Q

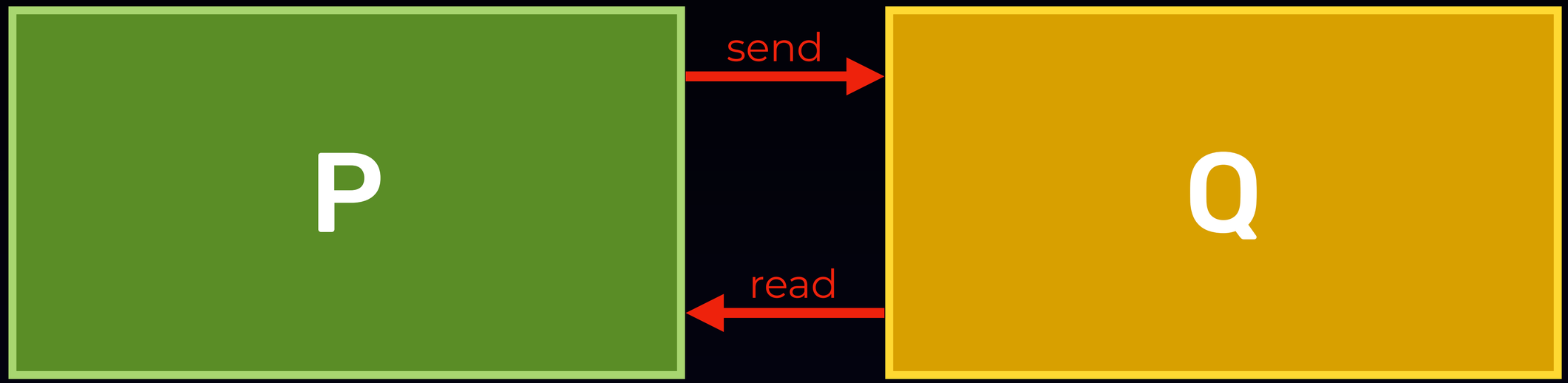



# Abstraction

- Mapping memory to agents
  - Green maps to P
  - Yellow maps to Q

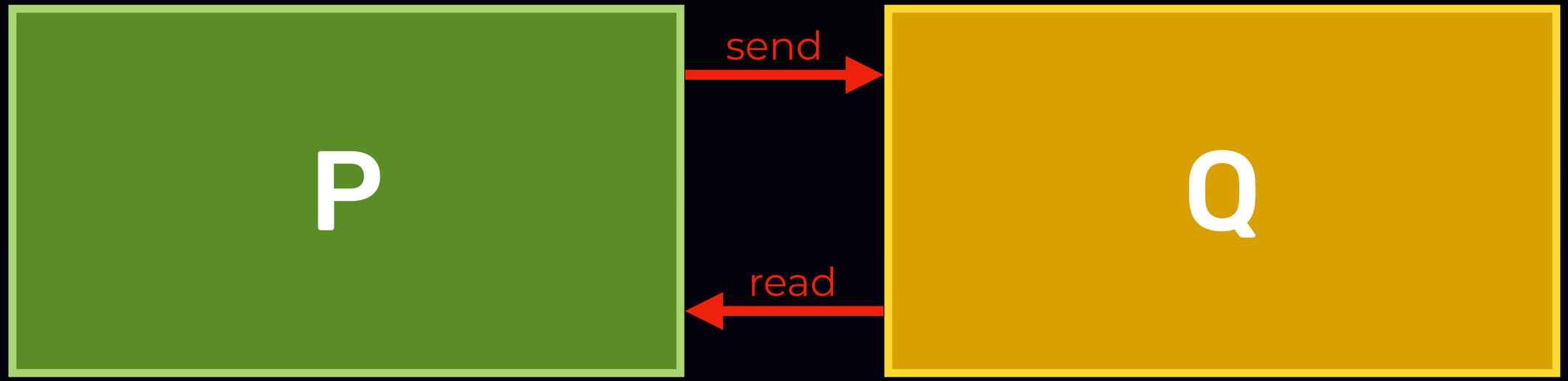



# Subject



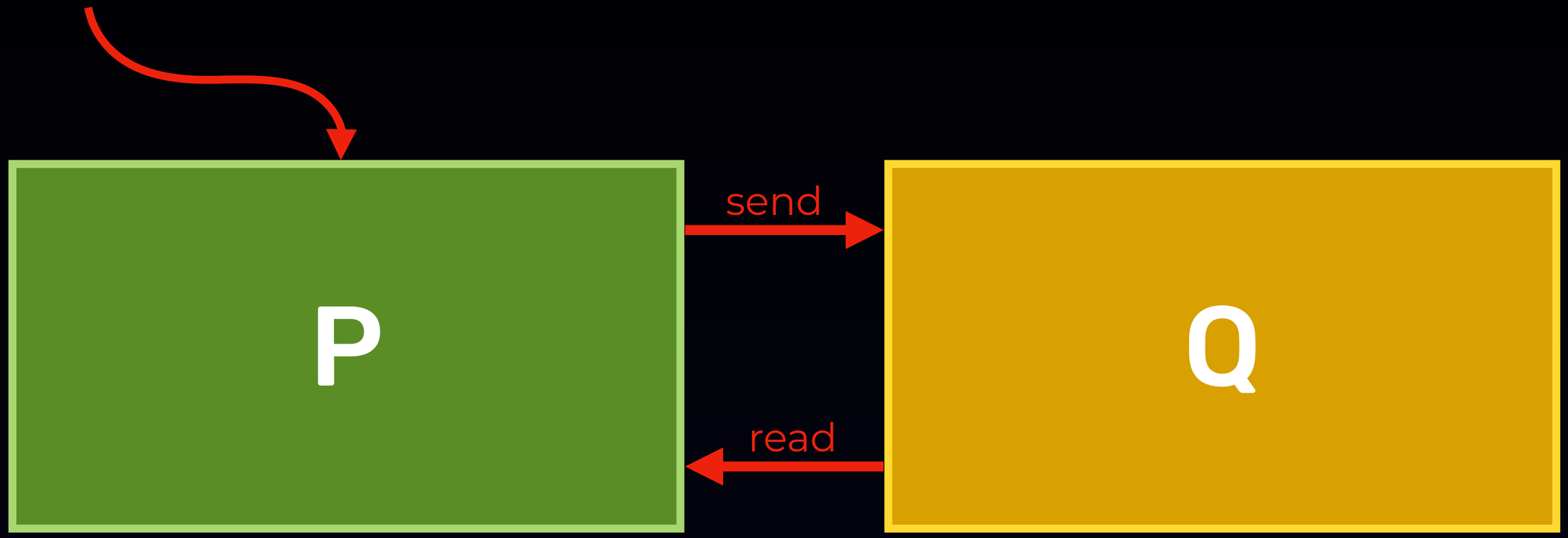
# Subject

- Agent in focus



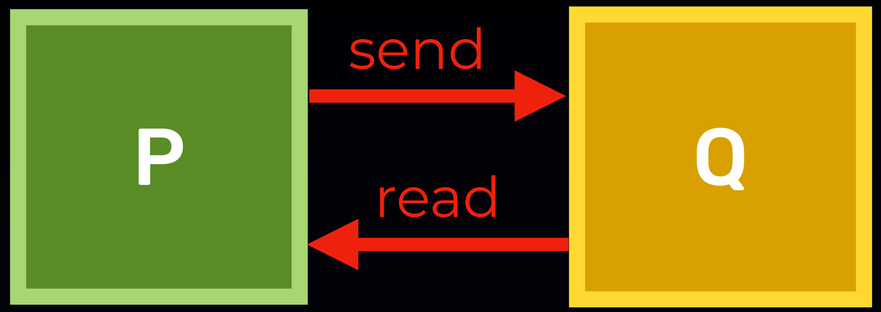
# Subject

- Agent in focus

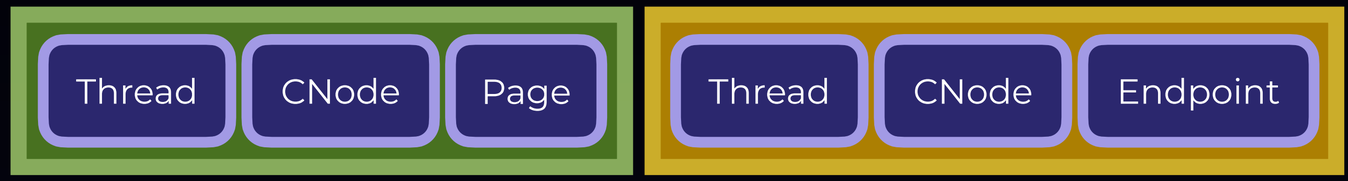


# Summary

**P**olicy




**A**bstraction



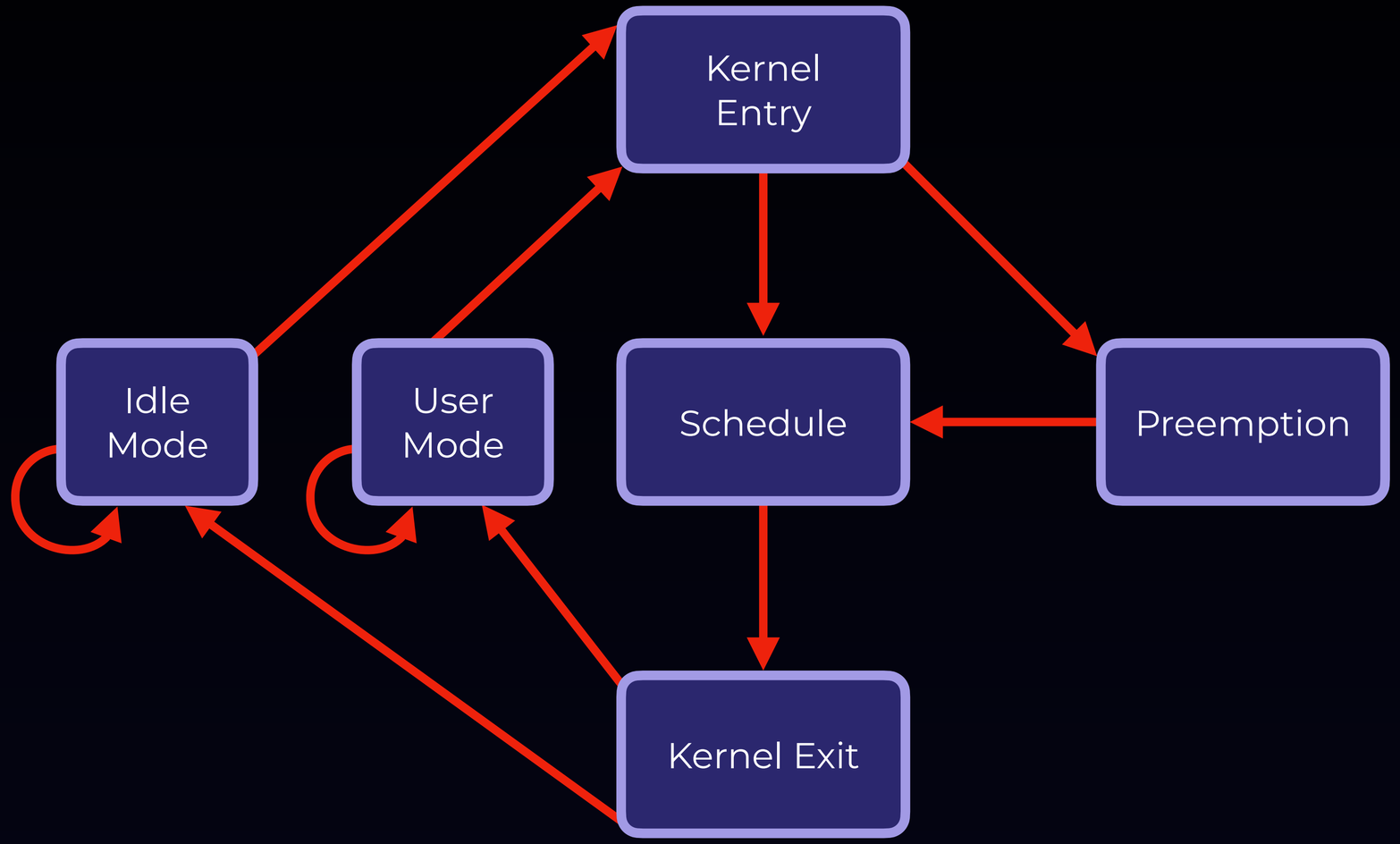

**S**ubject

# Proofs

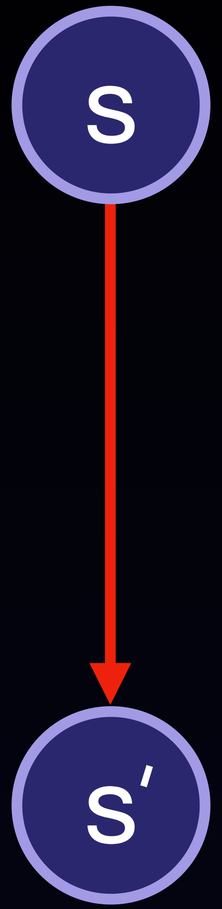
# Proofs

(Warning: Technical)

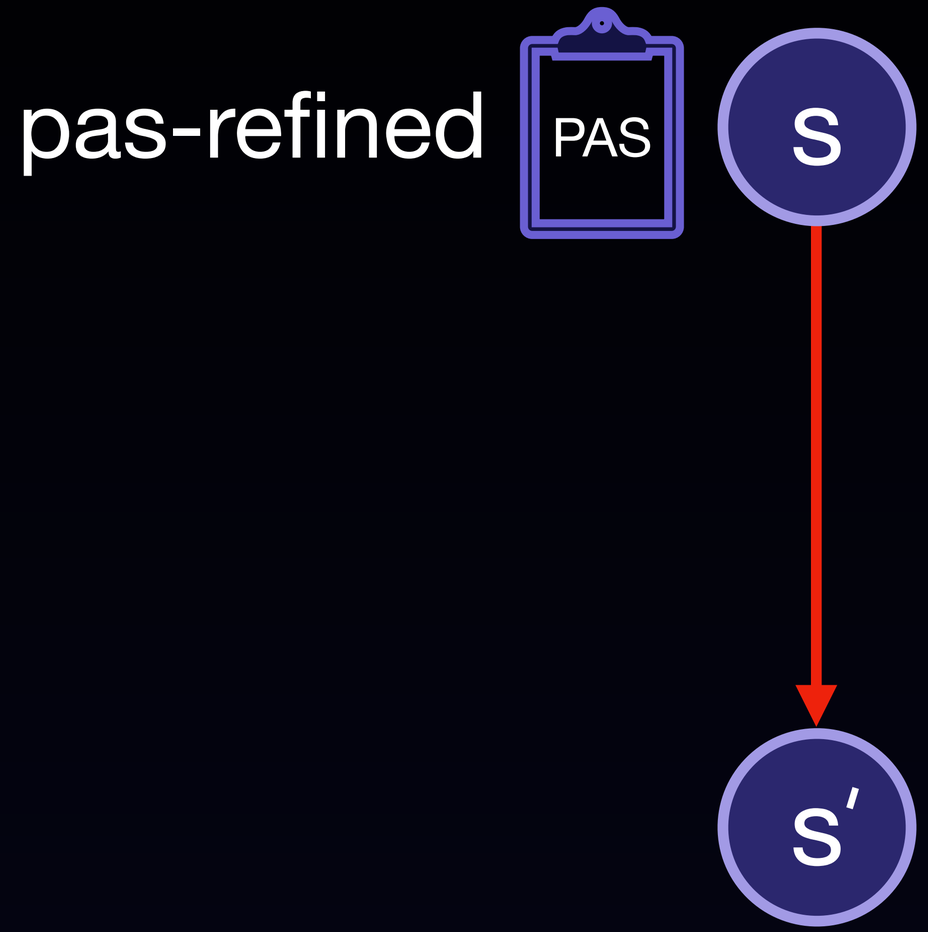
# Transition System



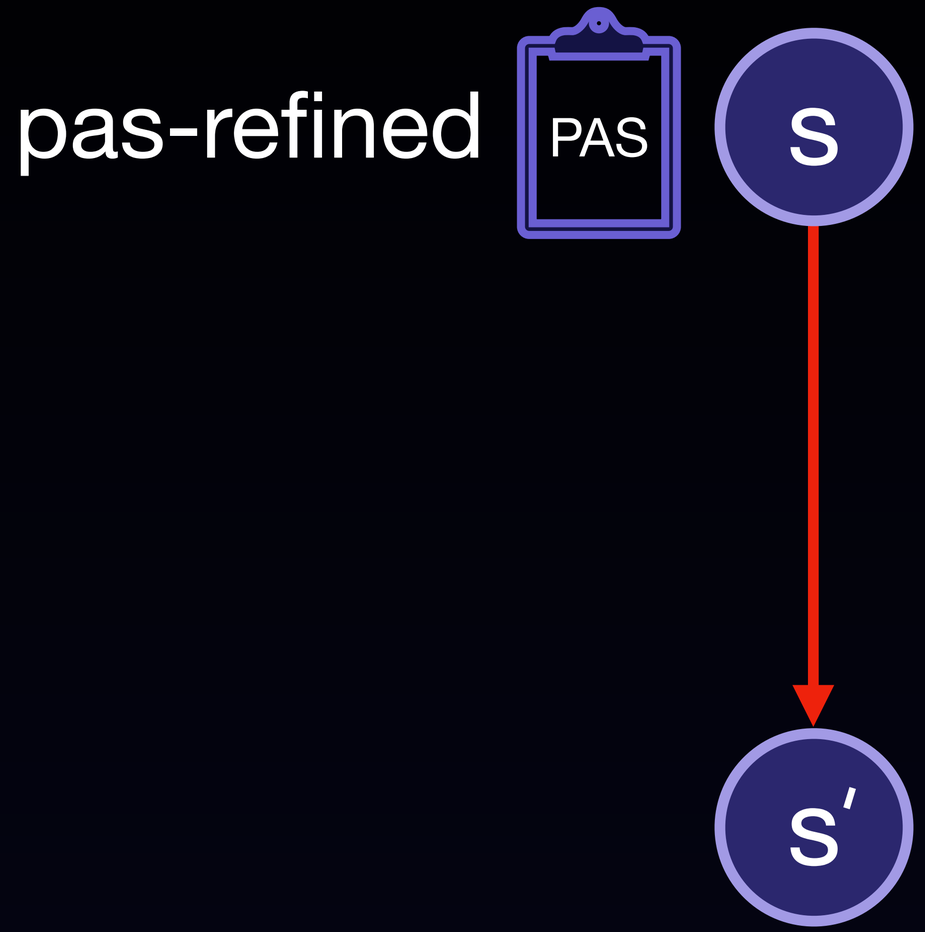
# Authority Confinement



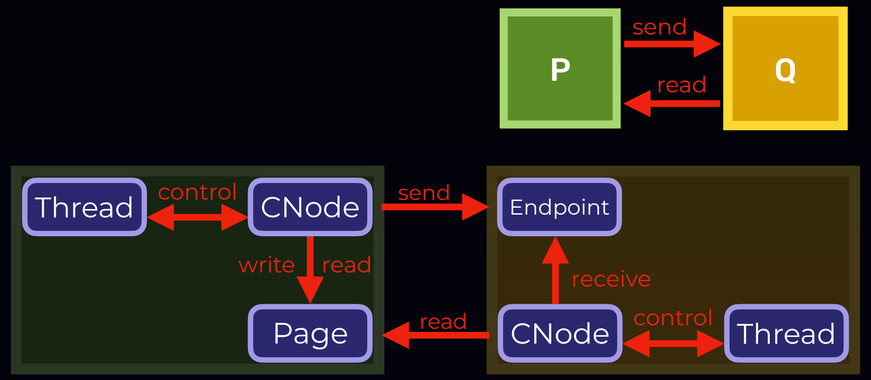
# Authority Confinement



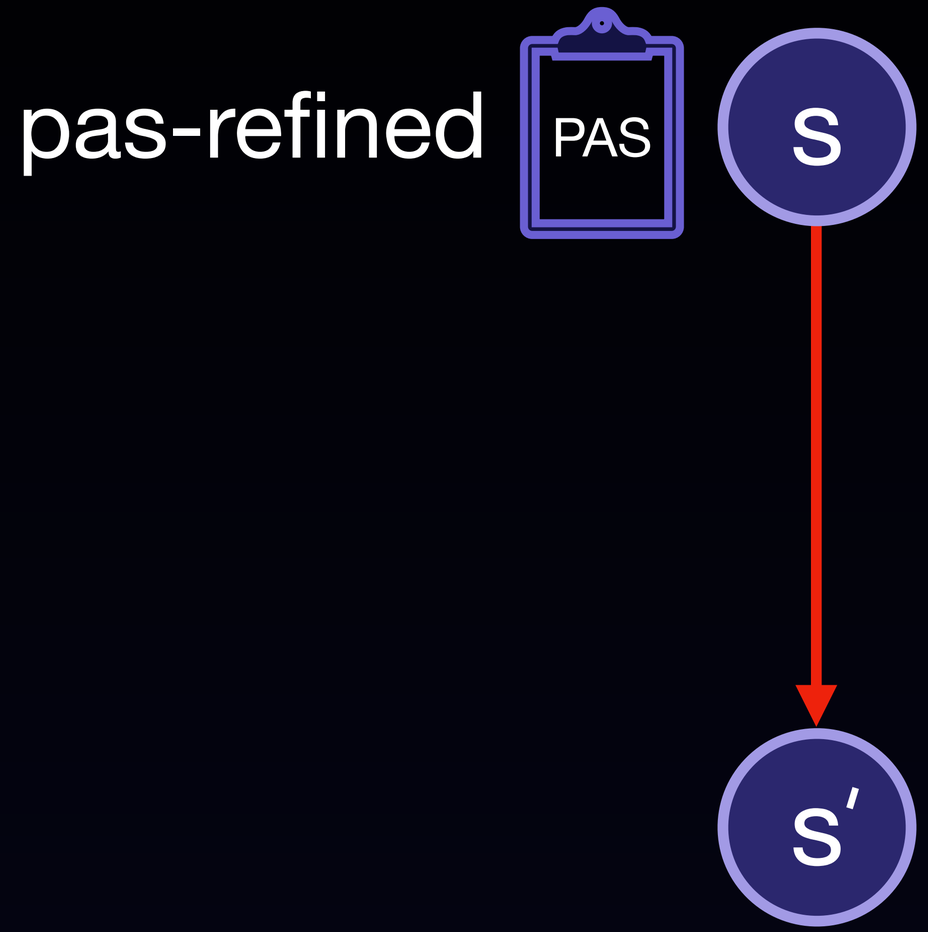
# Authority Confinement



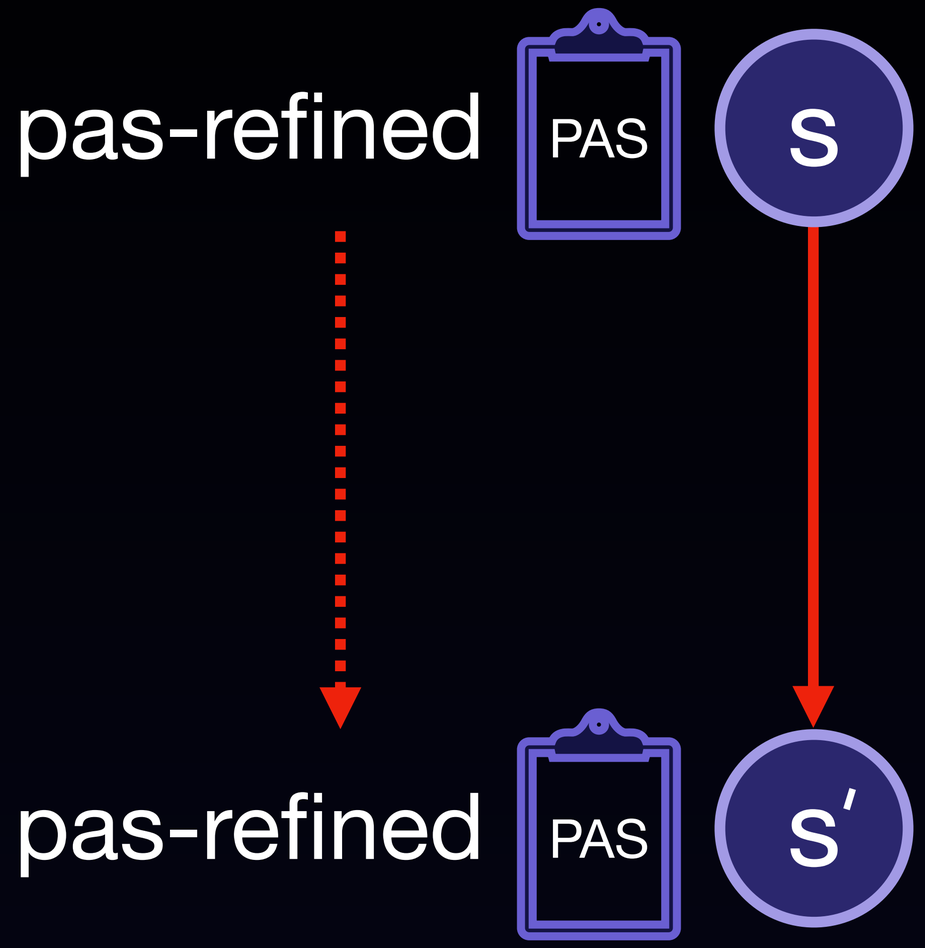

Recall



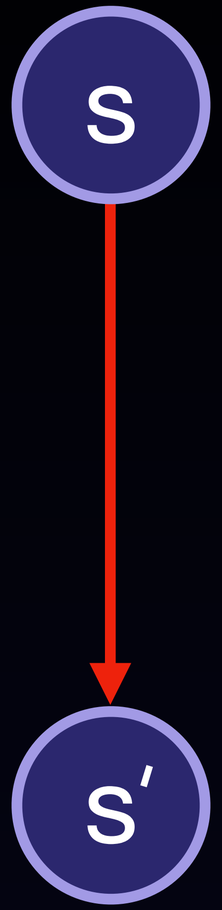
# Authority Confinement



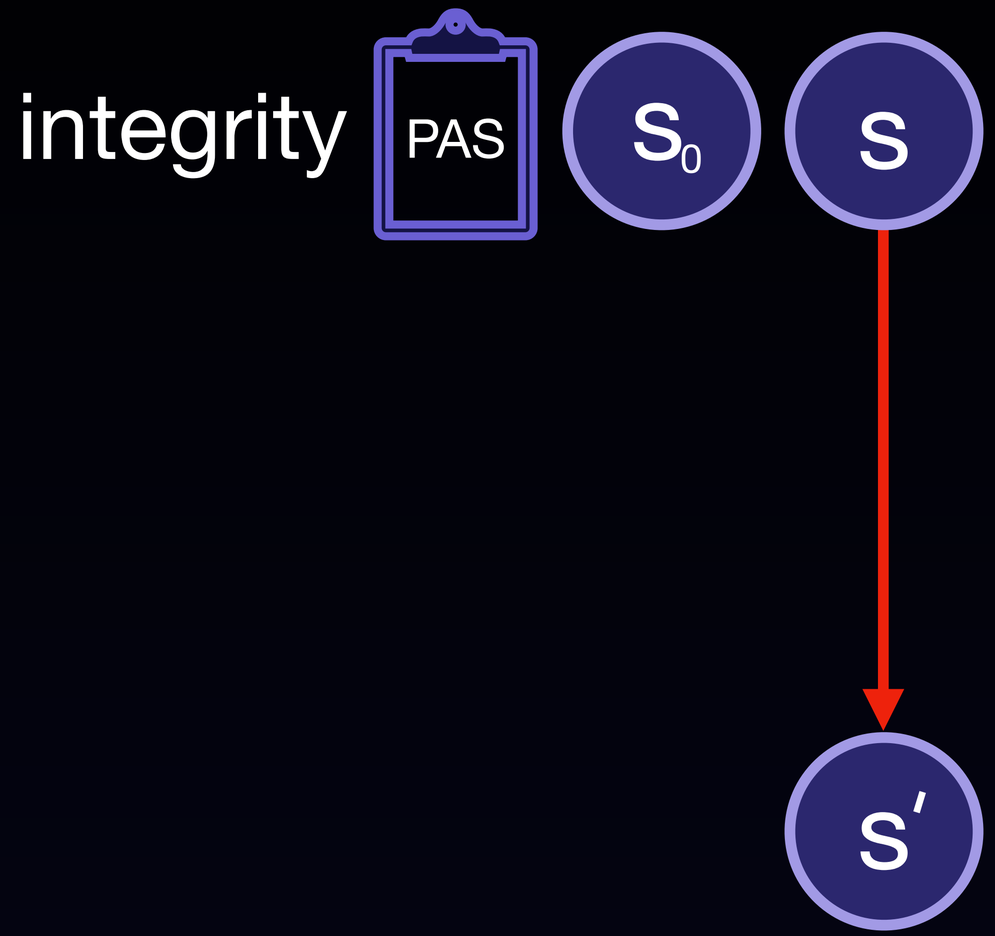
# Authority Confinement



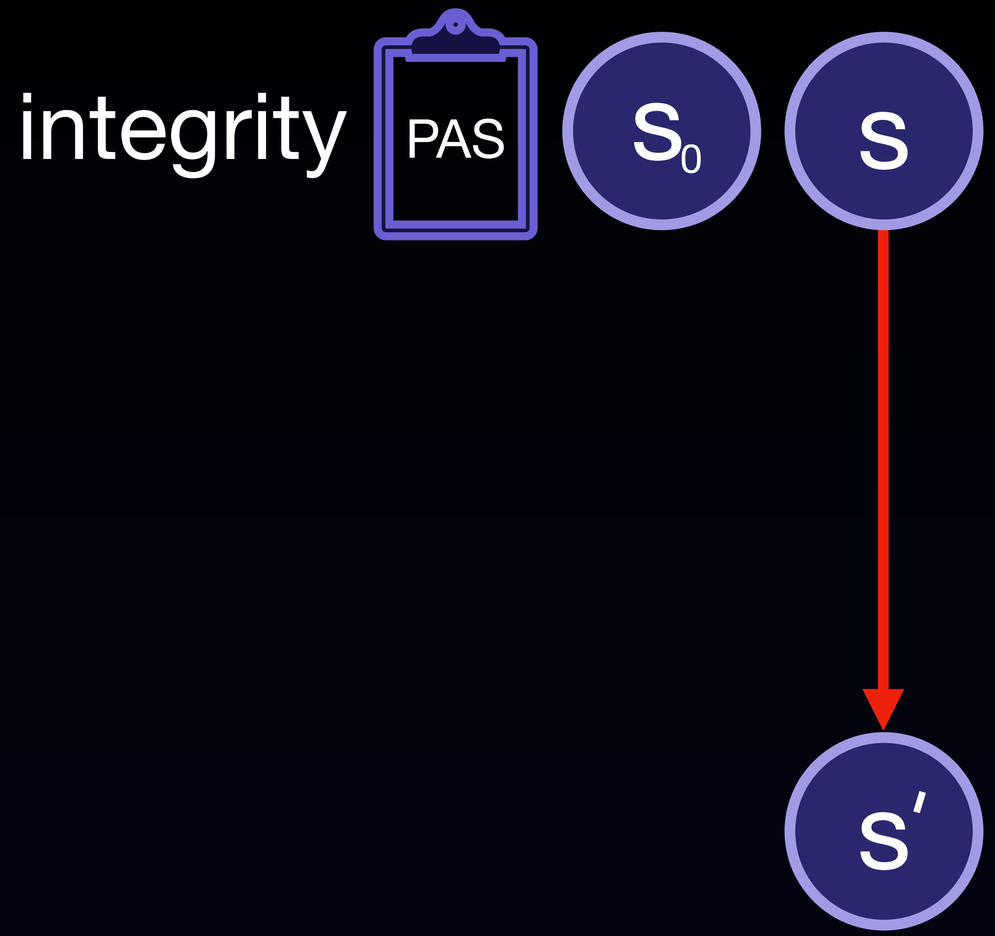
# Integrity

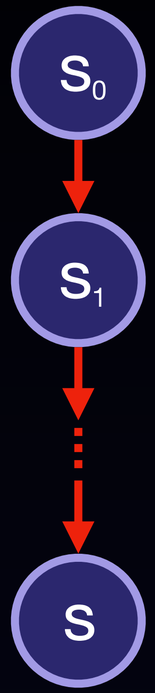


# Integrity

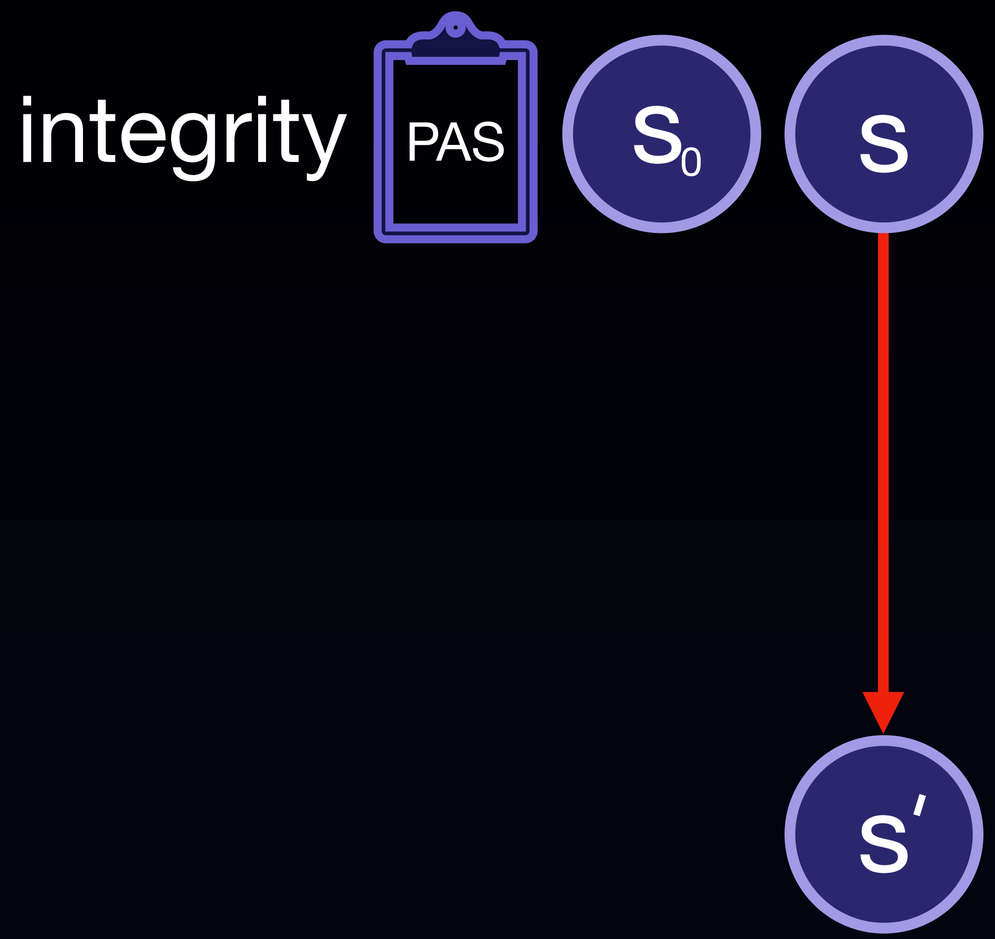


# Integrity

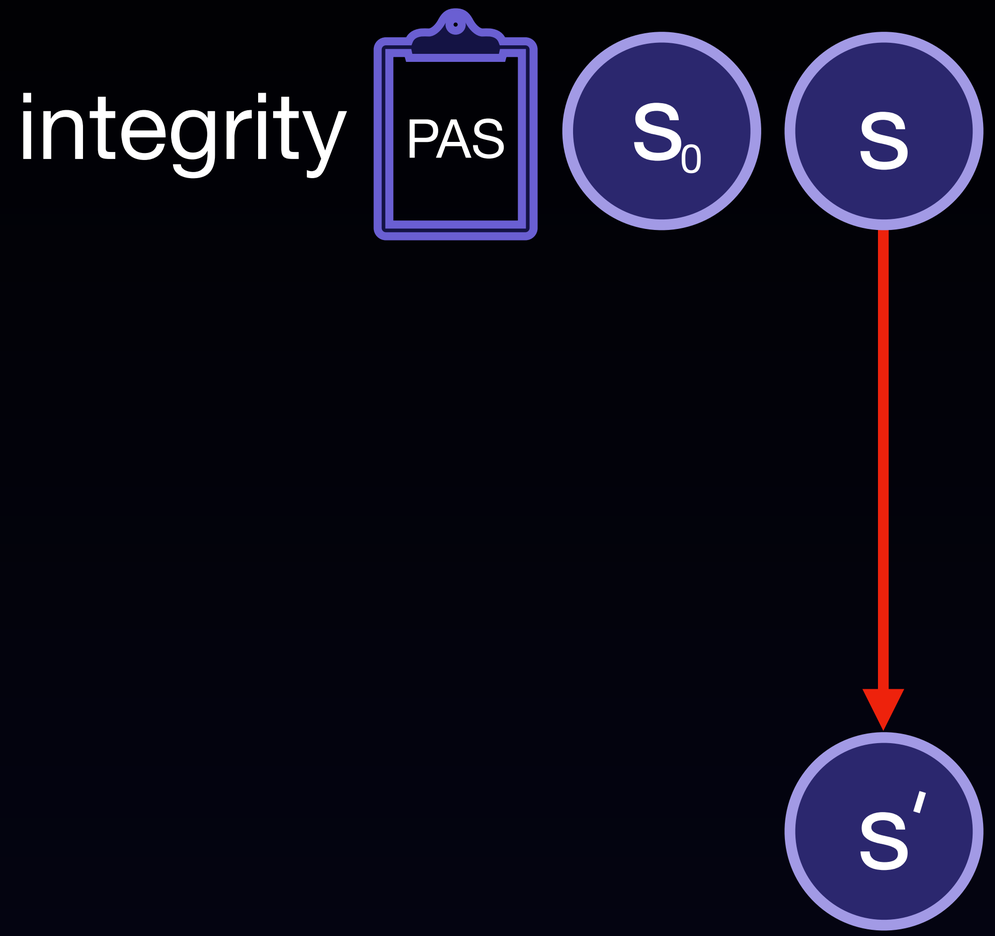


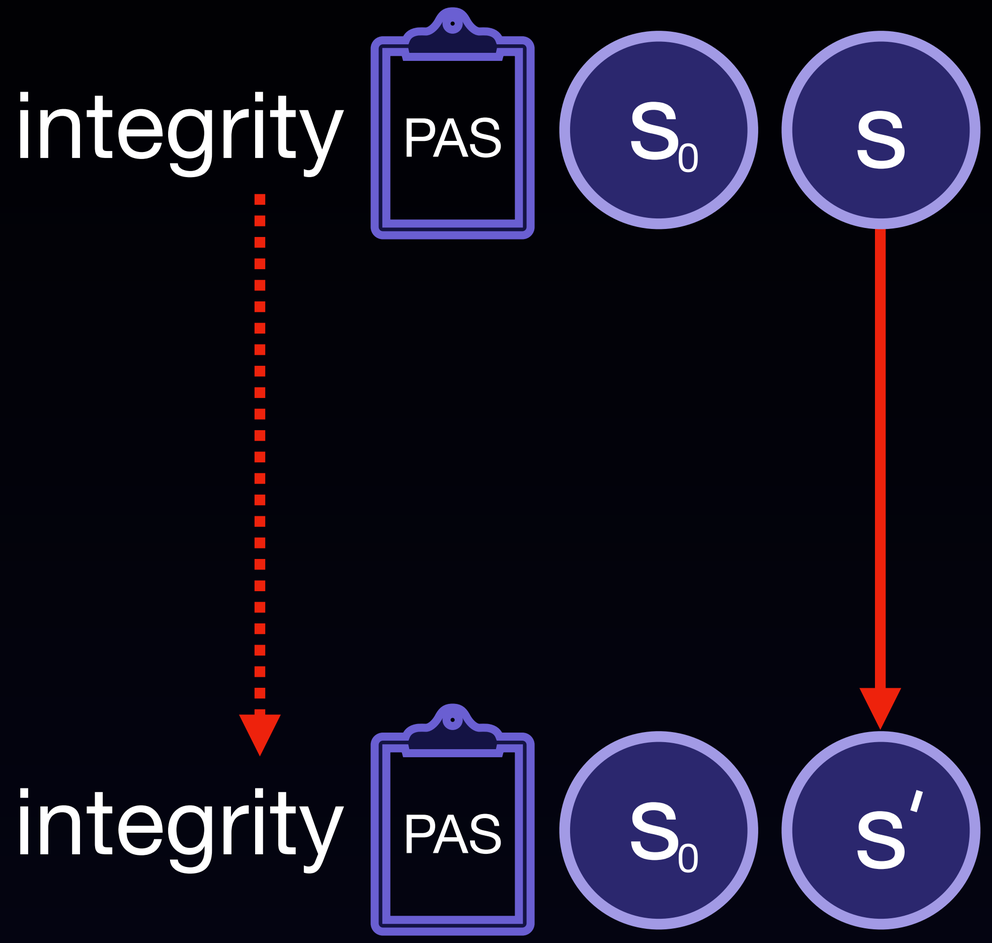
# Integrity



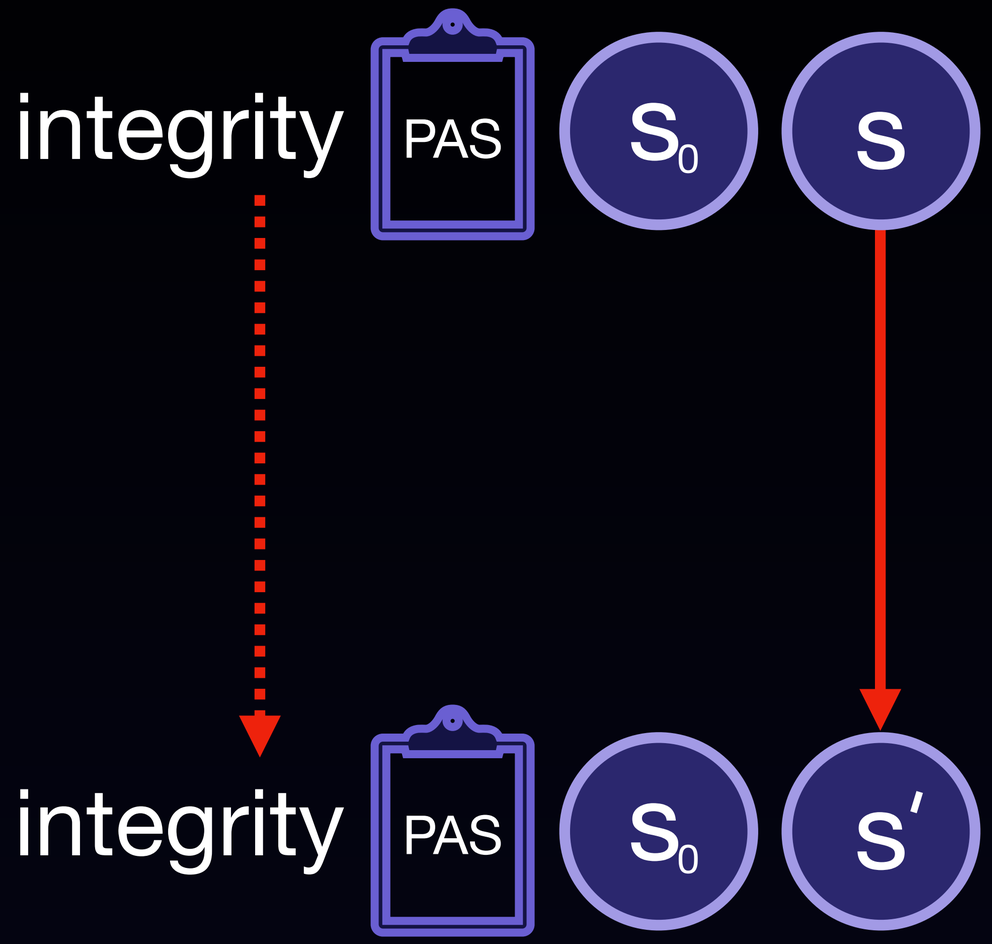

# Integrity

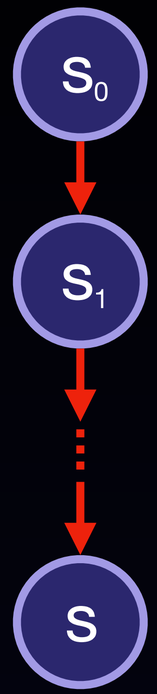


# Integrity

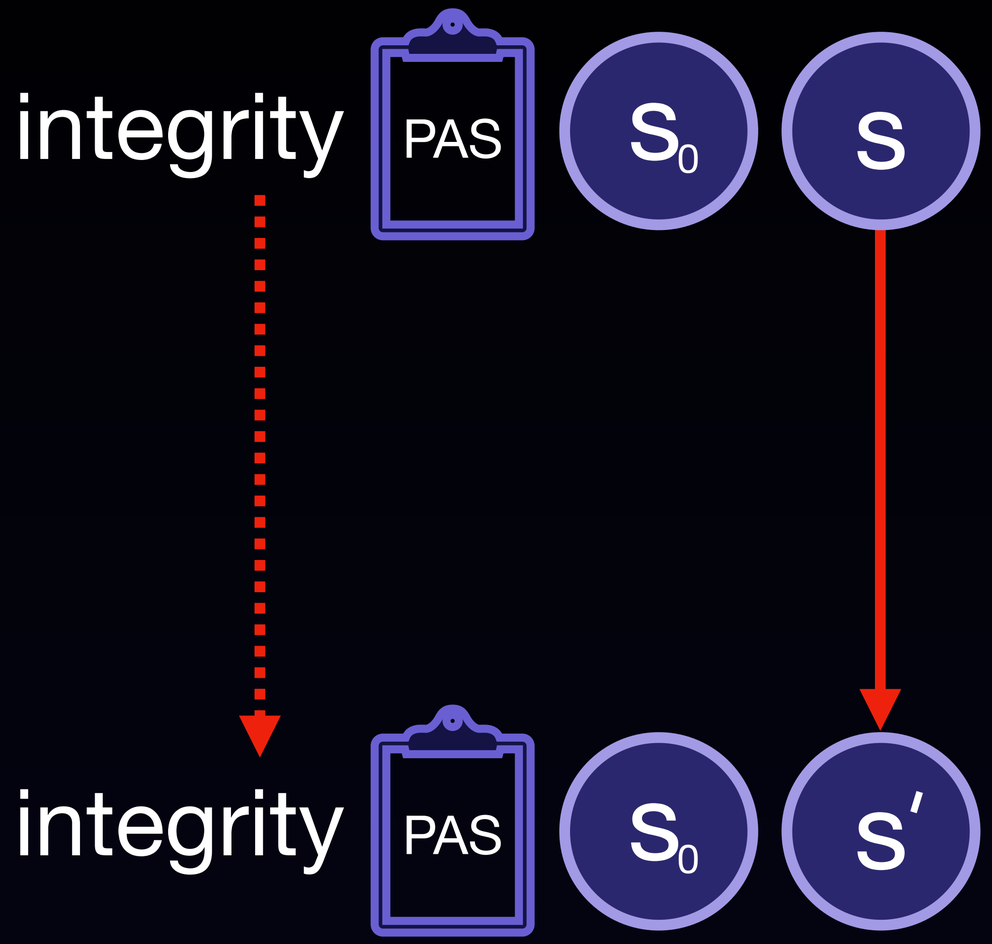


# Integrity

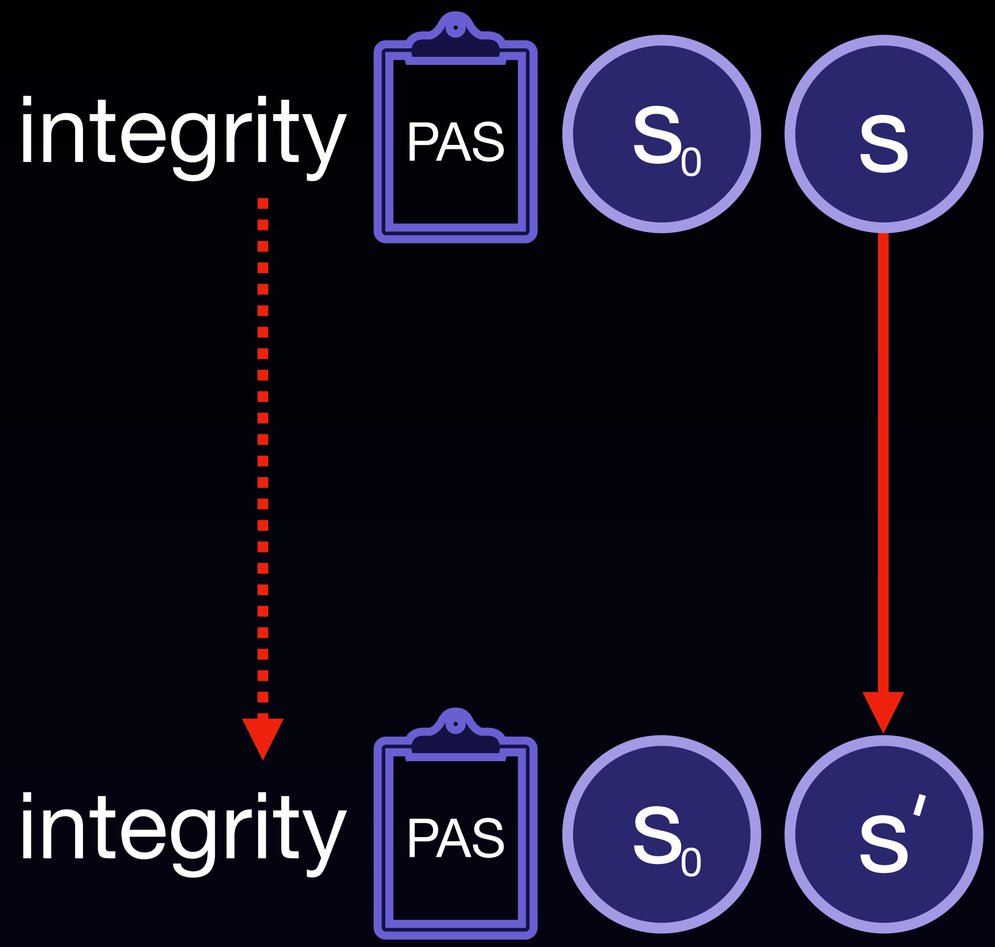


# Integrity

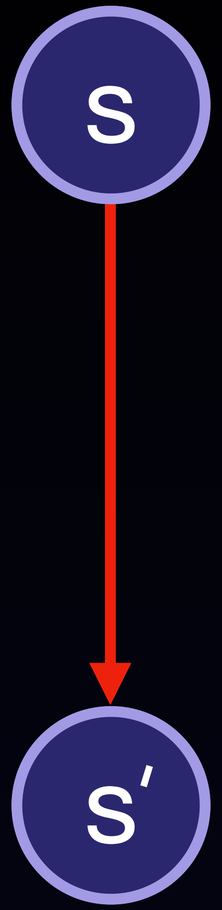



# Integrity

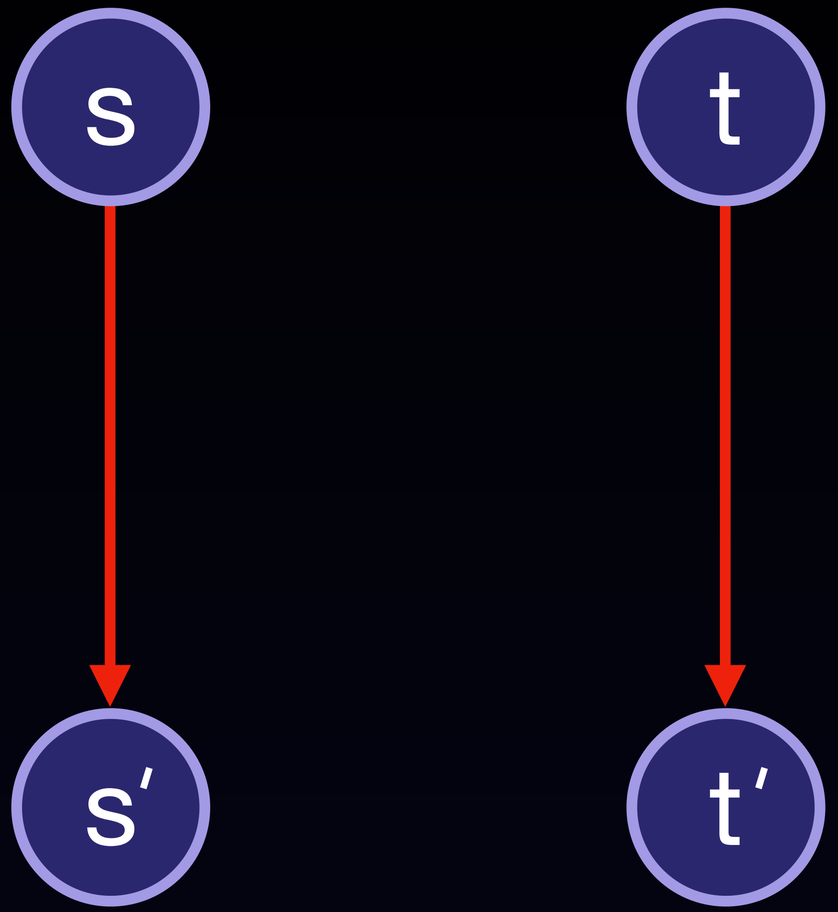



integrity PAS $s_0$ s′

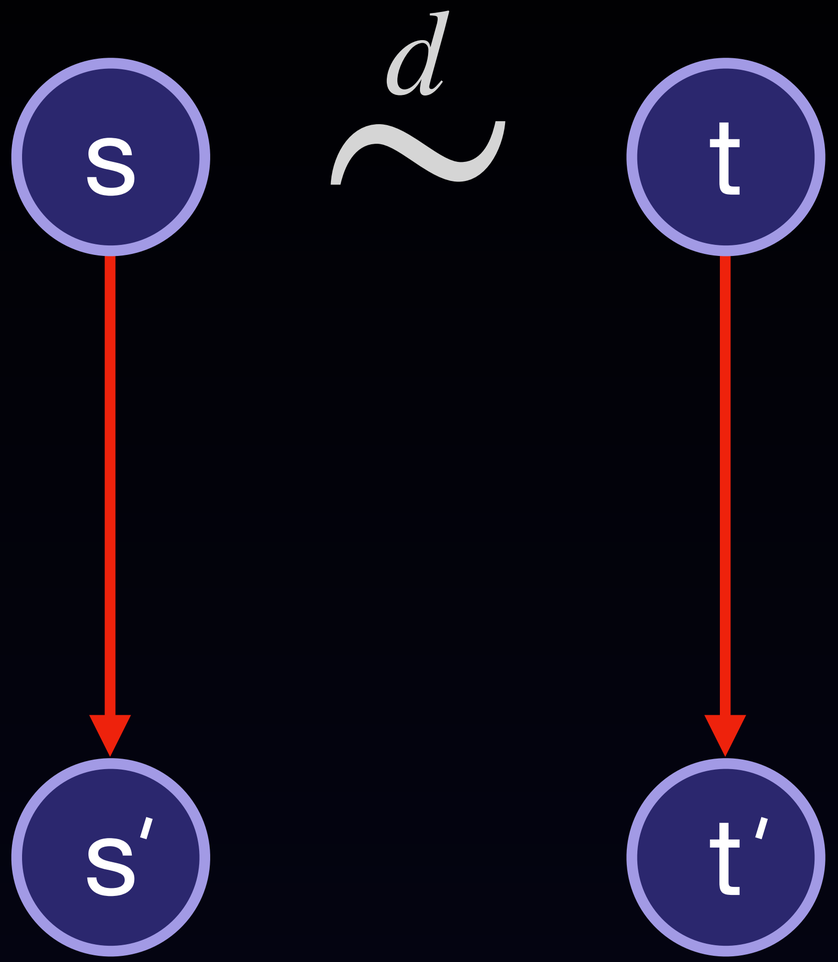
# Confidentiality



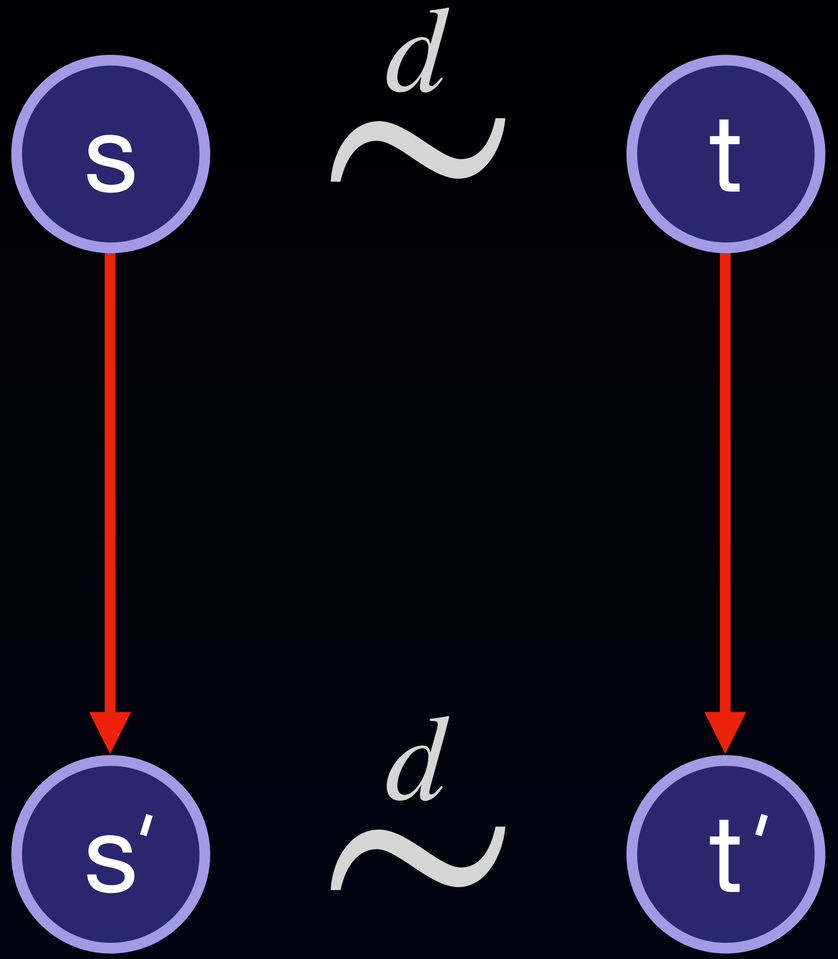
# Confidentiality



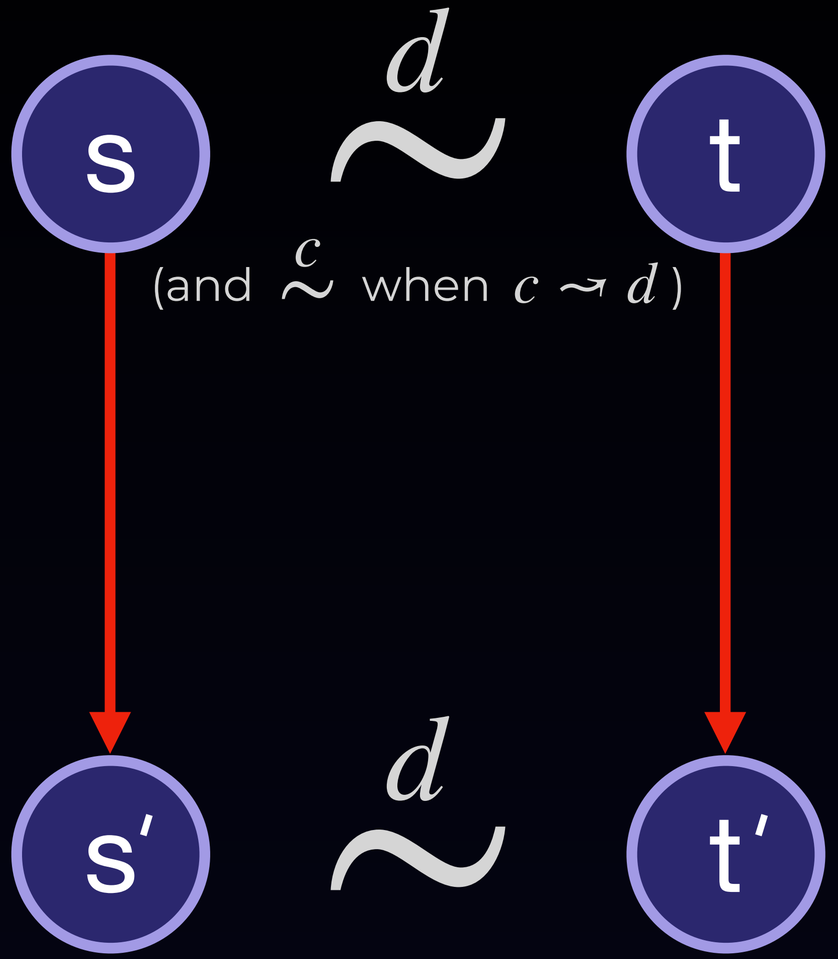
# Confidentiality



# Confidentiality



# Confidentiality



# Assumptions

- seL4's security assumptions include:

# Assumptions

- seL4's security assumptions include:

**Hardware Assumptions**

We assume the hardware works as expected

# Assumptions

- seL4’s security assumptions include:

**Hardware Assumptions**

We assume the hardware works as expected

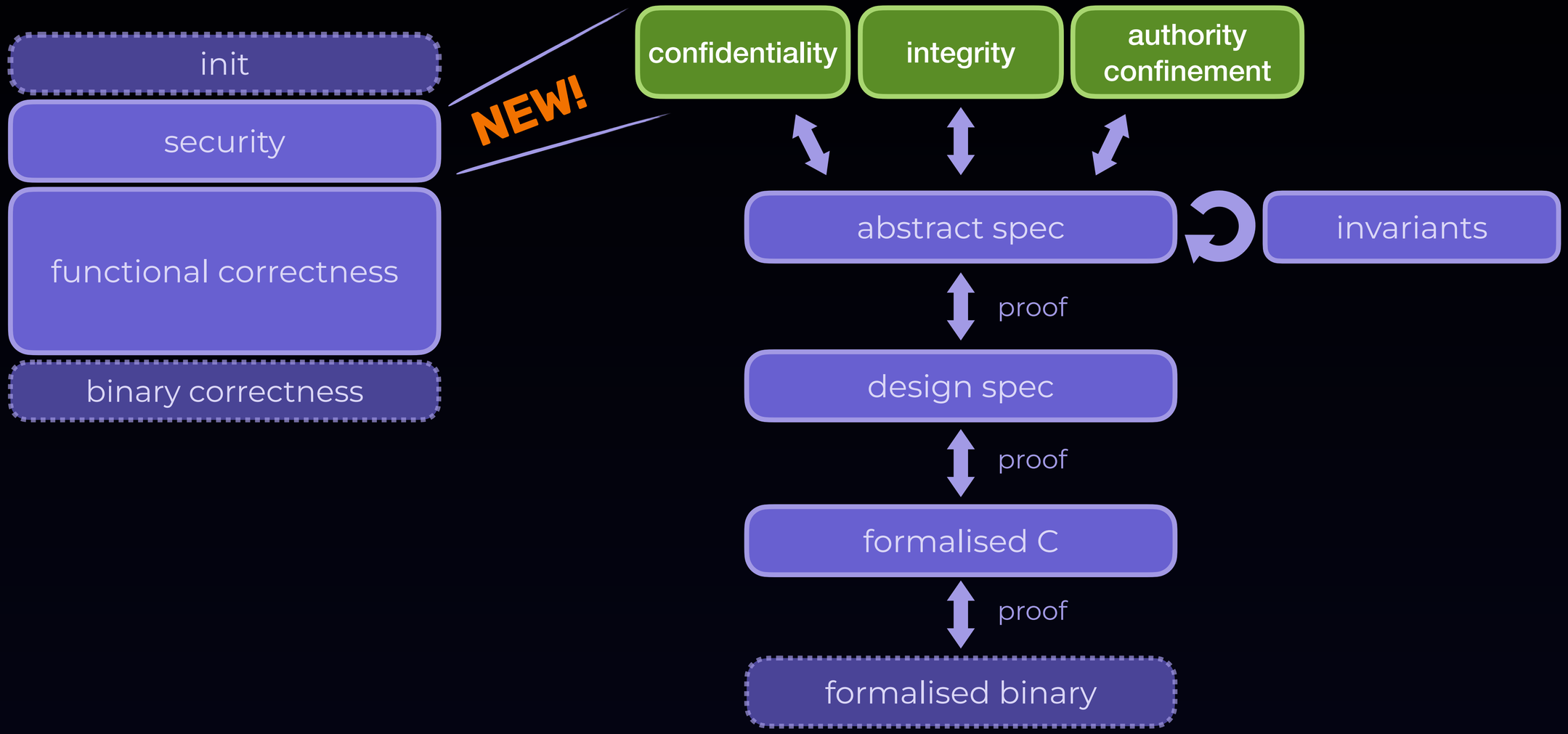
**Side-channel Assumptions**

We assume our model captures all relevant information channels

# AArch64 Security

Thanks to support from NCSC

# AArch64 Verification Stack

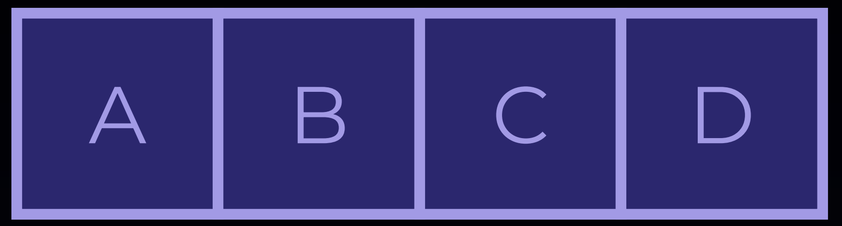


# AArch64 Additions

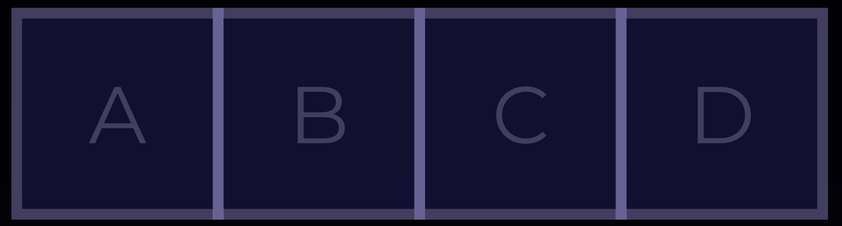
**New to the security proofs:**

- Virtual CPUs (VCPU)
- Floating-point units (FPUs)

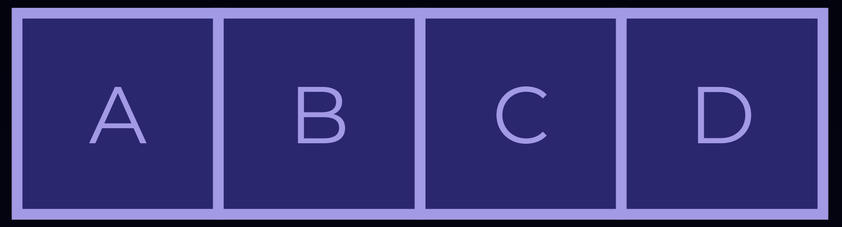
# Example



# Example

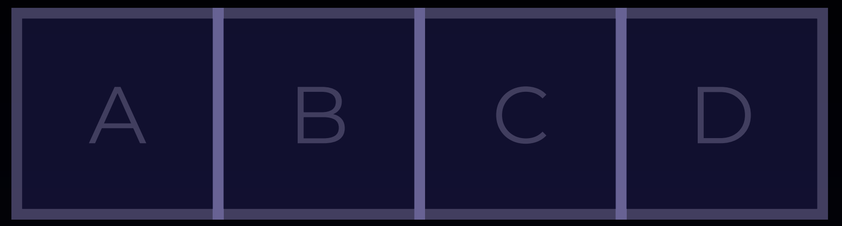



Heap

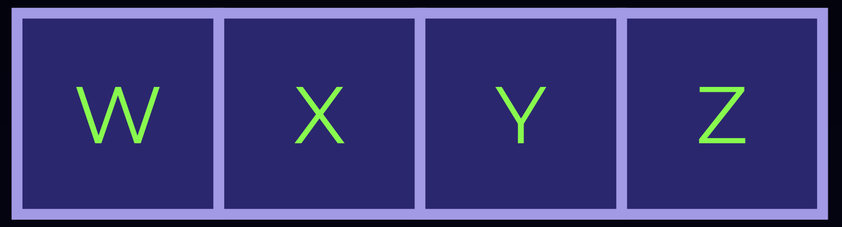



Registers

# Example




Heap




Registers

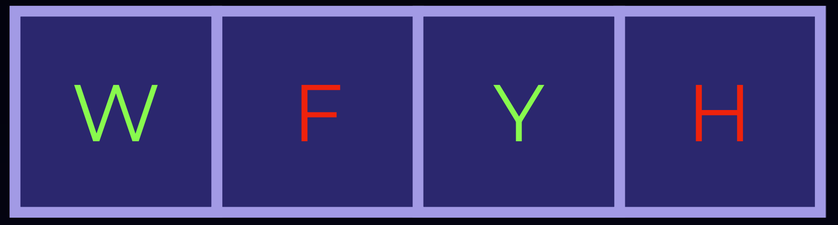
# Example


Heap


Registers

# Example

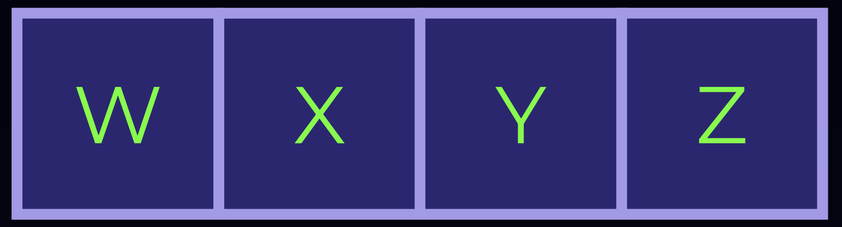

Heap




Registers

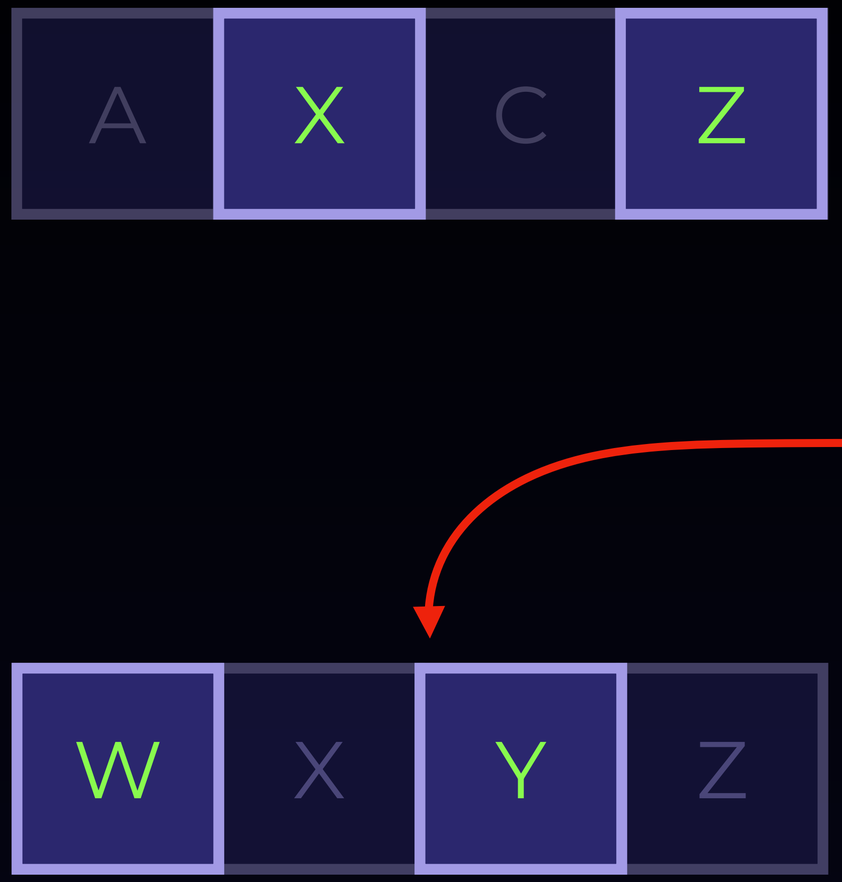
# Example


Heap




Registers

# Example



# Example

Constrain how this changes


Heap

+

Constrain how this changes


Registers

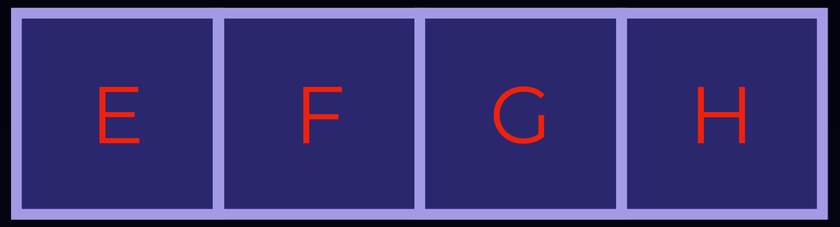
# Example

Constrain how this changes


Heap

+

Constrain how this changes




Registers

# Example

Constrain how this changes


Heap

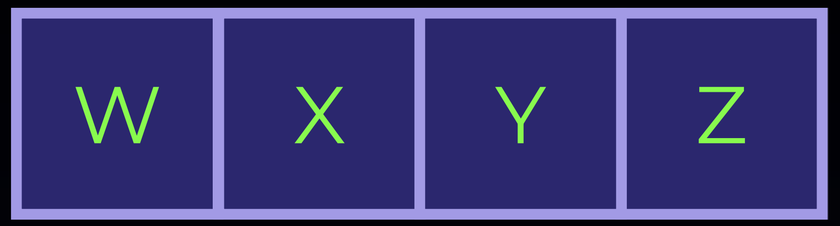
+

Constrain how this changes


Registers

# Example

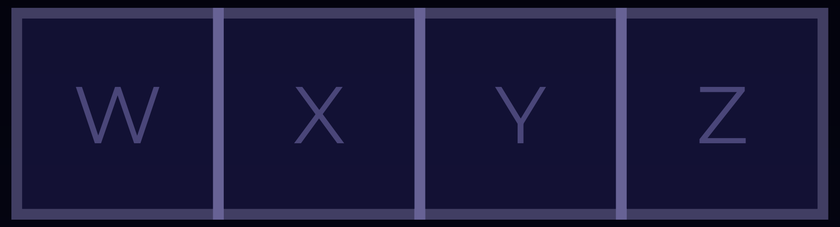
Constrain how this changes




Heap

+

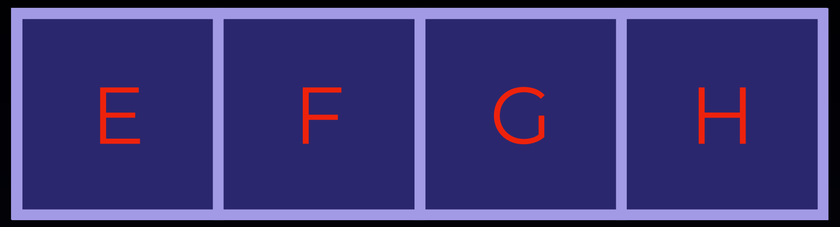
Constrain how this changes




Registers

# Example

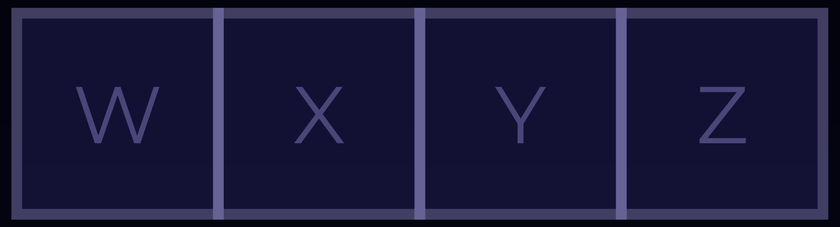
Constrain how this changes




Heap

+

Constrain how this changes



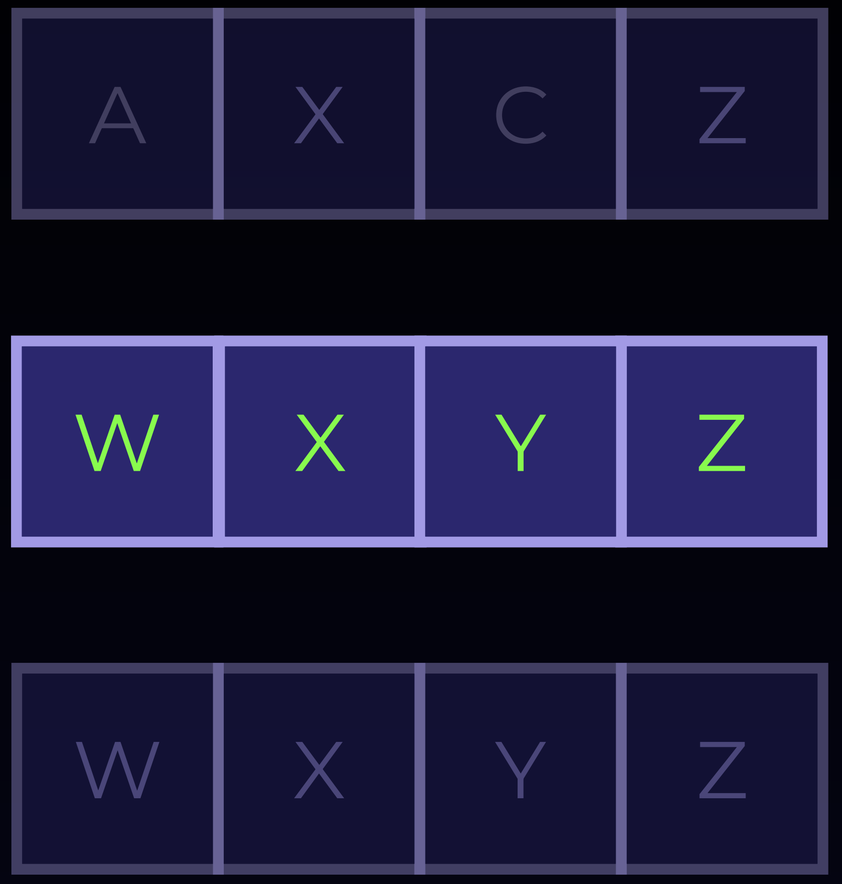

Registers

# Example

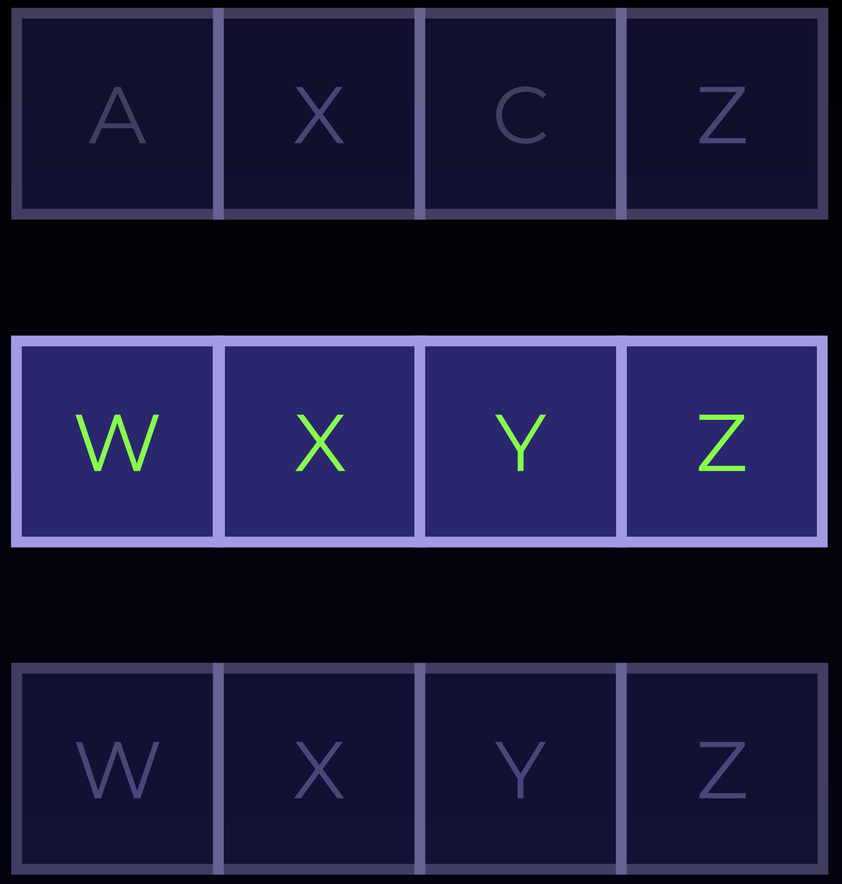

Heap


Registers

# Example

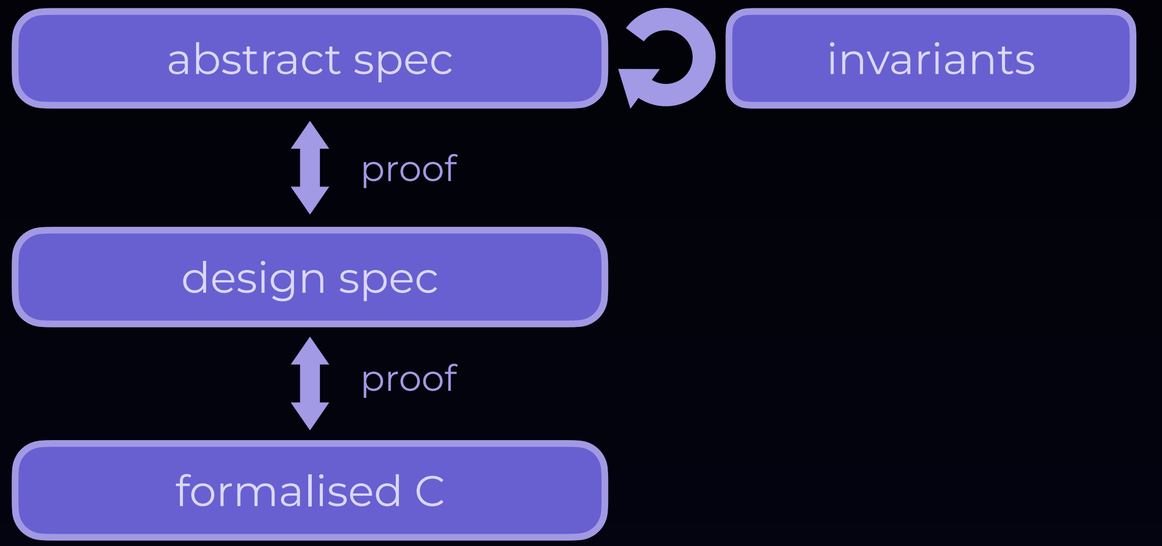


# Example

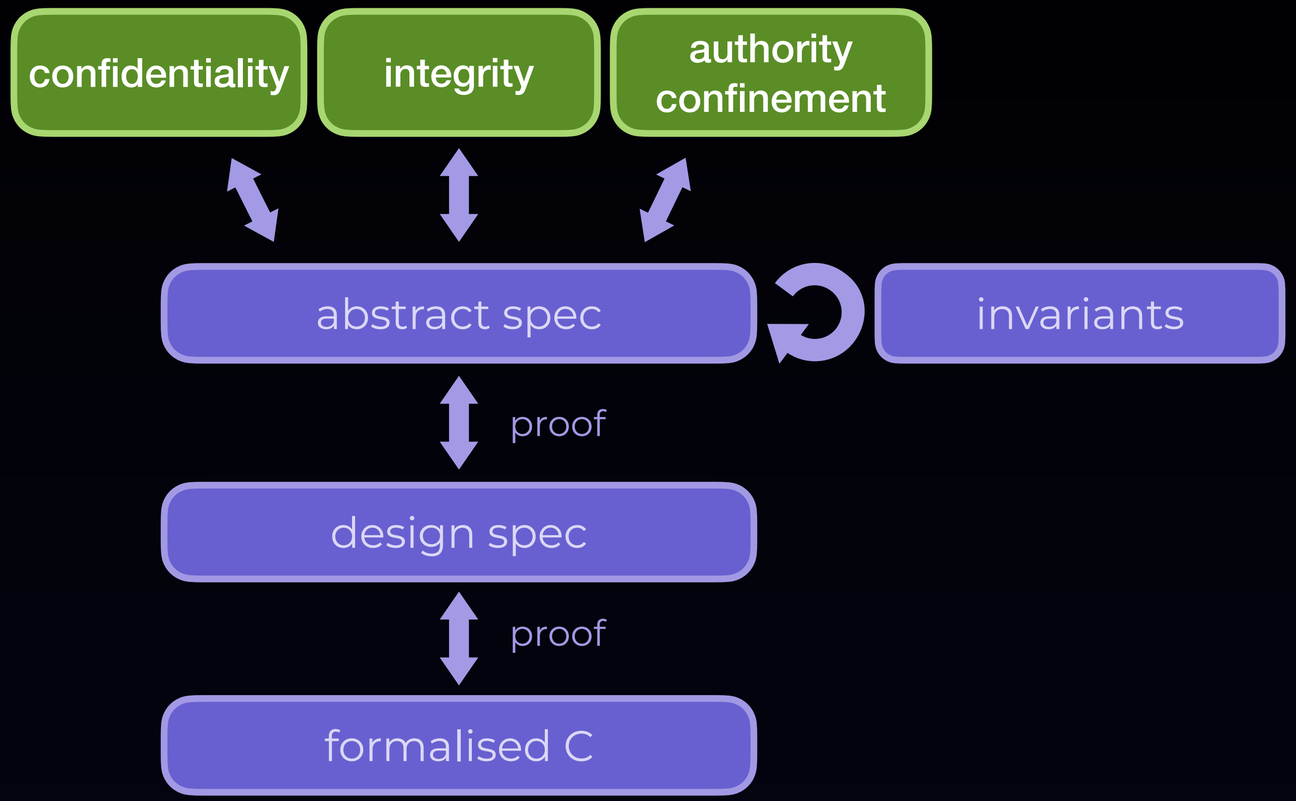


# Summary

# Summary

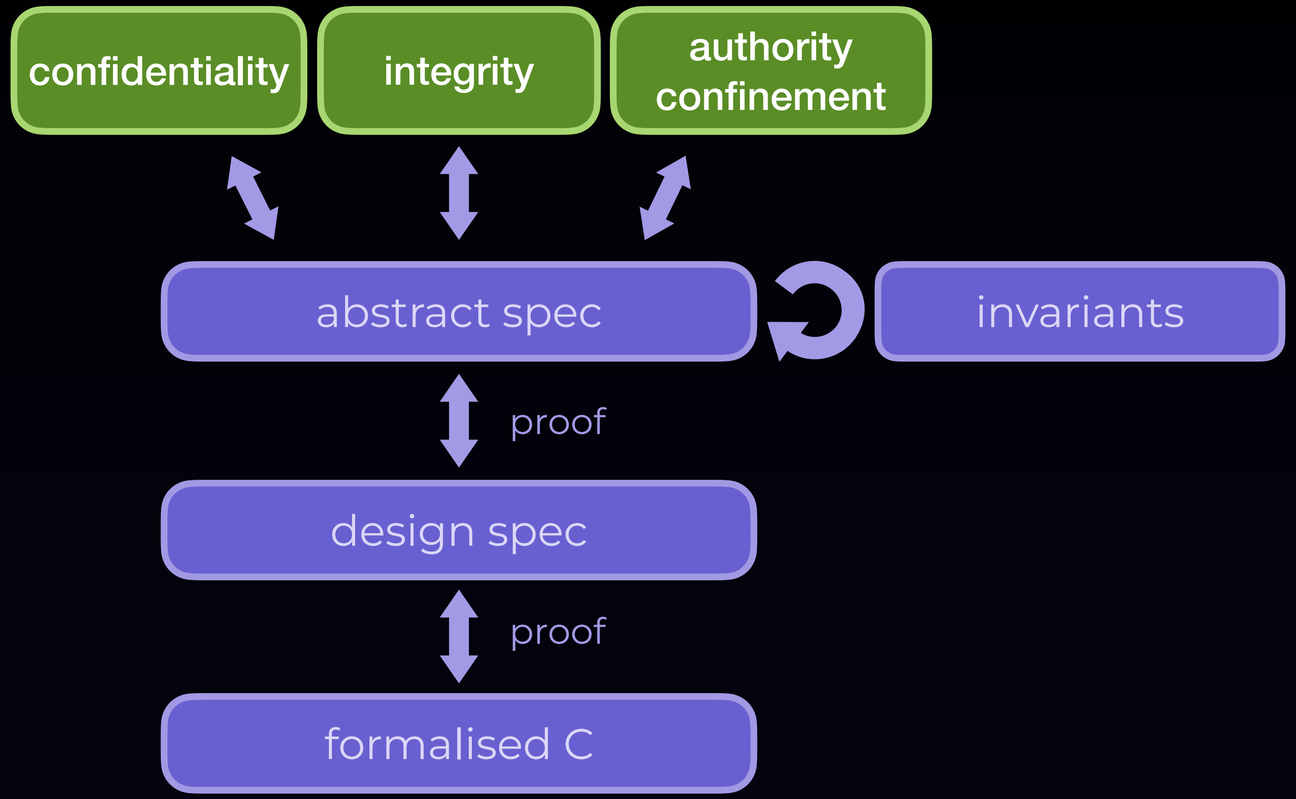

abstract spec
invariants
proof
design spec
proof
formalised C

# Summary


confidentiality
integrity
authority confinement
abstract spec
invariants
proof
design spec
proof
formalised C

# Summary

**Now on AArch64**



# What else to expect

**Also happening in seL4 verification**

- **MCS** proofs completed for RISC-V
- **RISC-V Platforms** all supported
- **Arch-split Refine** completed
- **Multikernel** proofs in progress

# Thank You