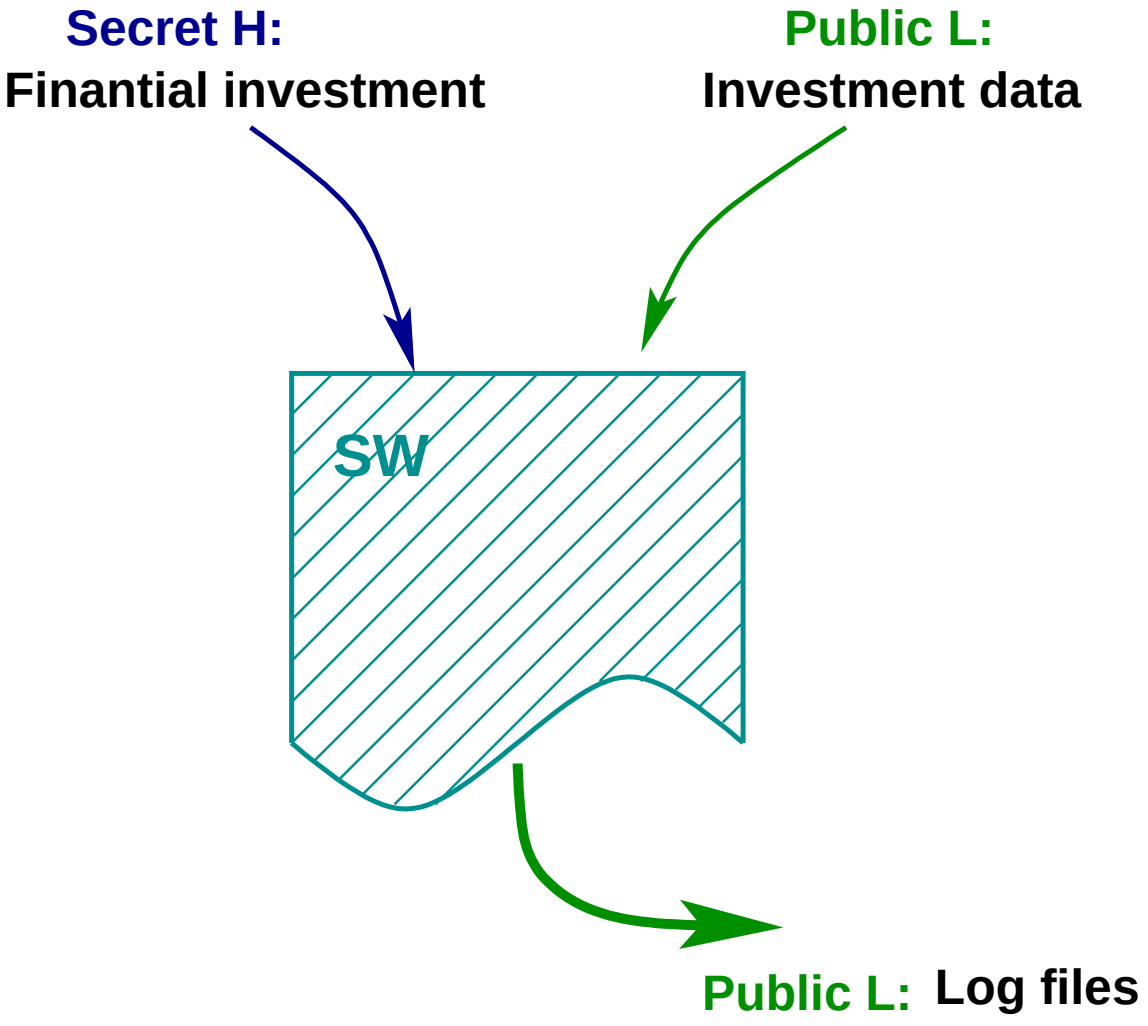
On the rôle of Abstract Non-Interference in Language-based Security

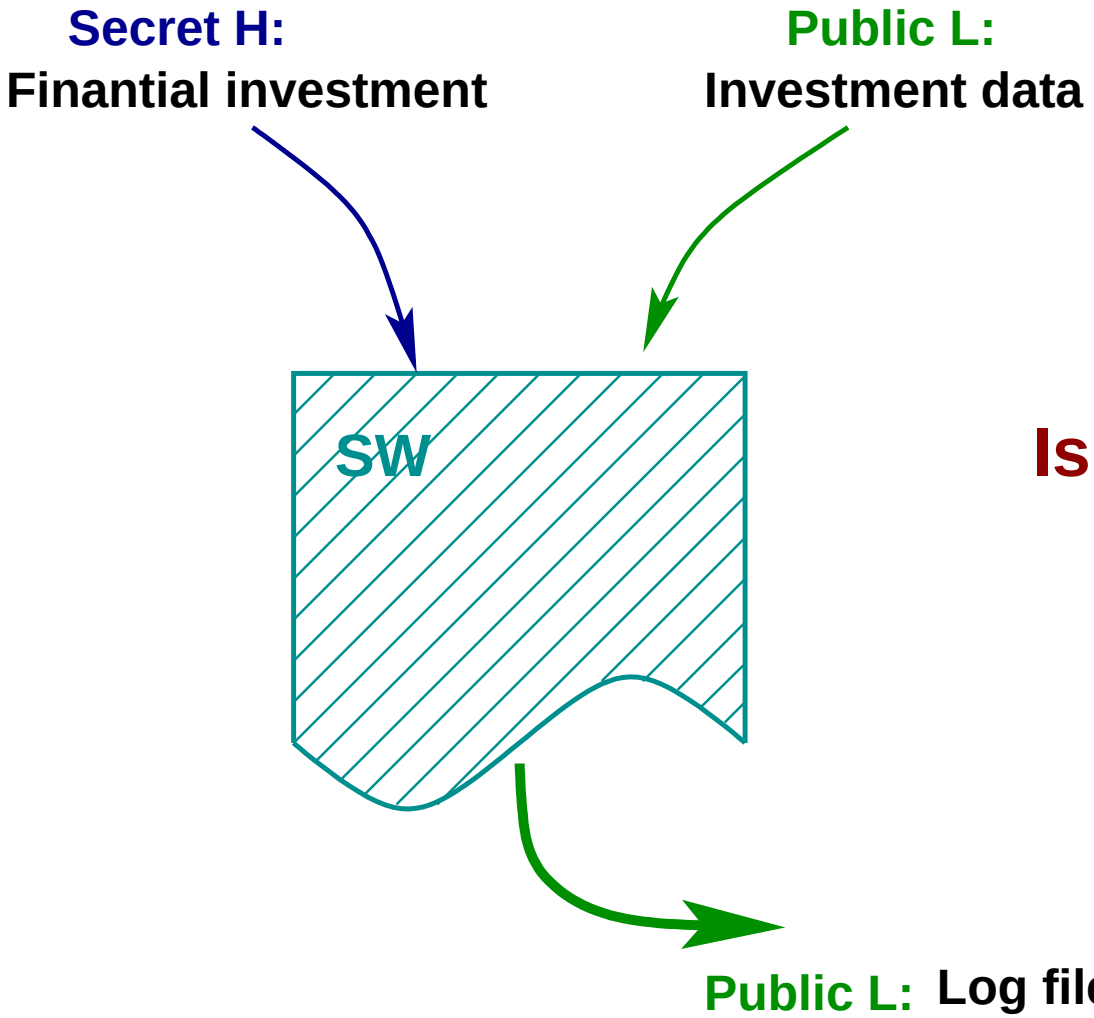
Isabella Mastroeni

Tsukuba, November 3rd, 2005

The Problem: Non-Interference

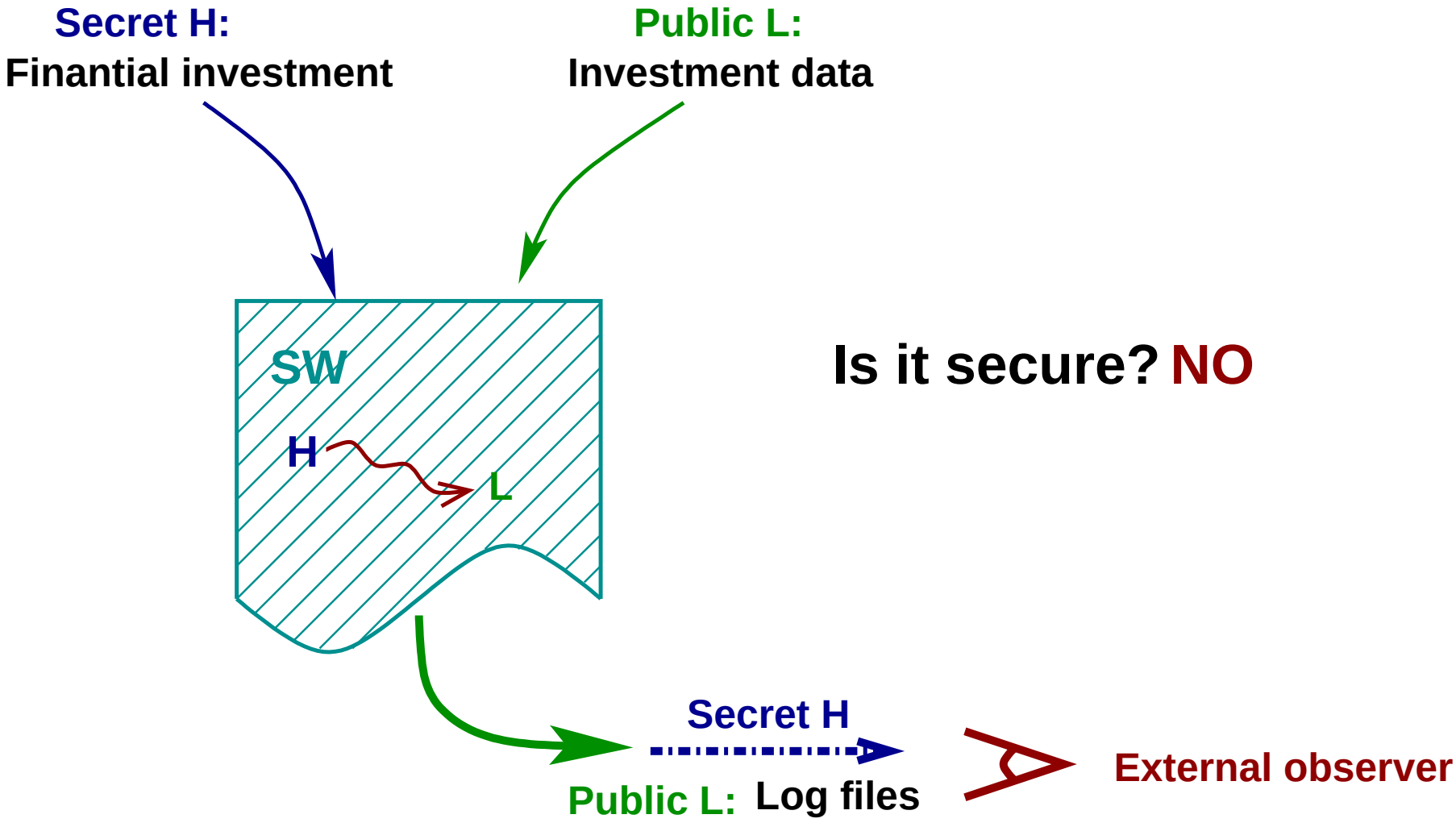


The Problem: Non-Interference



Is it secure?

The Problem: Non-Interference



Background

SECURITY PROPERTY: States which classes have not to interfere with other classes of objects.

Background

SECURITY PROPERTY: States which classes have not to interfere with other classes of objects.

Confinement problem [Lampson'73]: *Preventing the results of computations leaking even partial information about the confidential inputs.*

Background

SECURITY PROPERTY: States which classes have not to interfere with other classes of objects.

Confinement problem [Lampson'73]: *Preventing the results of computations leaking even partial information about the confidential inputs.*

Non-interference policies require that any *change upon confidential data has not to be revealed through the observation of public data.*

- Many real systems are intended to leak some kind of information
- Even if a system satisfies non-interference, some kind of tests could reject it as insecure

Background

SECURITY PROPERTY: States which classes have not to interfere with other classes of objects.

⇓

Confinement problem [Lampson'73]: *Preventing the results of computations leaking even partial information about the confidential inputs.*

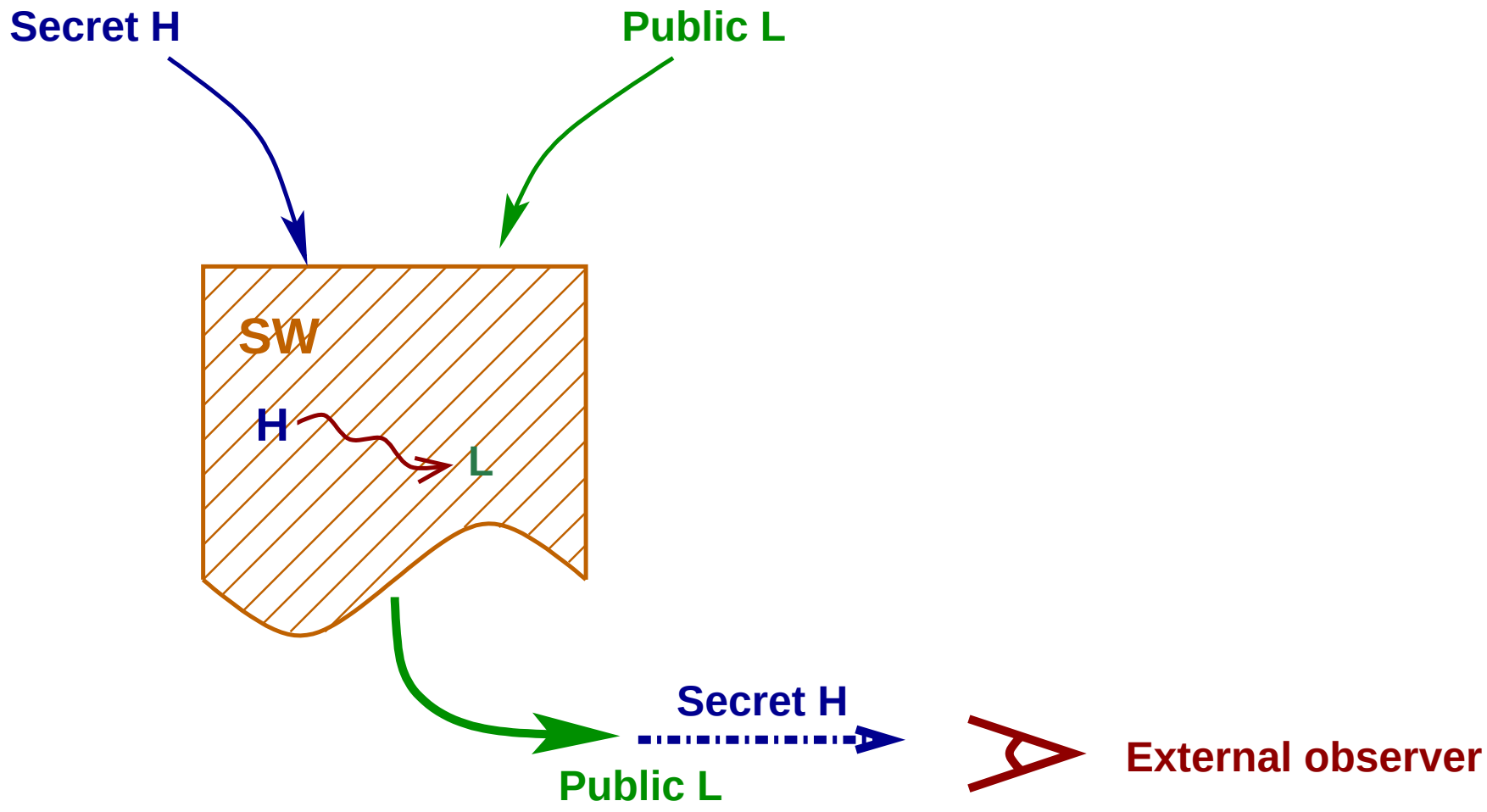
⇓

Non-interference policies require that any *change upon confidential data has not to be revealed through the observation of public data.*

- Characterizing released information: [Cohen'77], [Zdancewic & Myers'01], [Clark et al.'04], [Lowe'02];
- Constraining attackers: [Di Pierro et al.'02], [Laud'01].

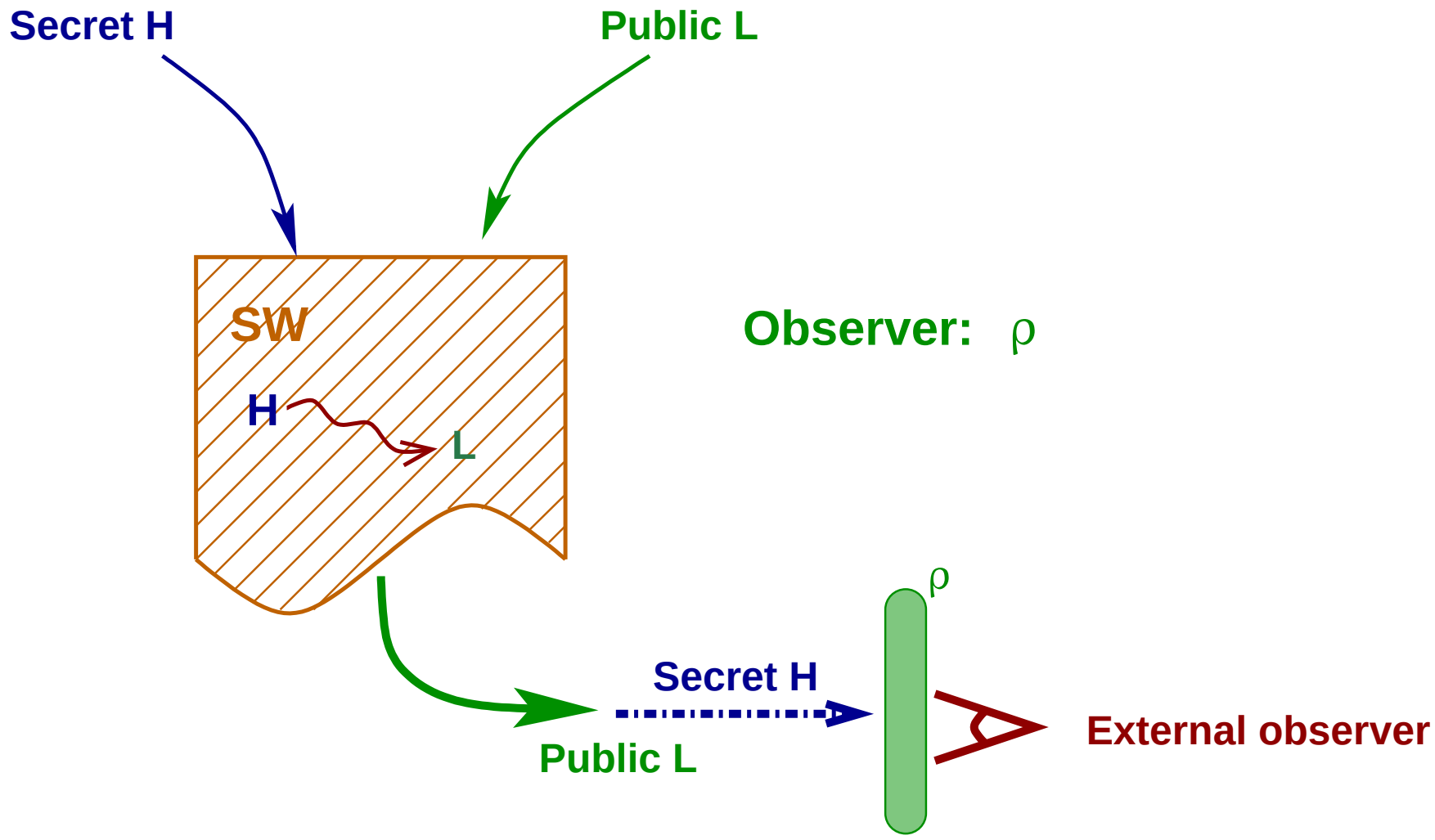
Abstract Non-Interference

[Giacobazzi & Mastroeni '04]



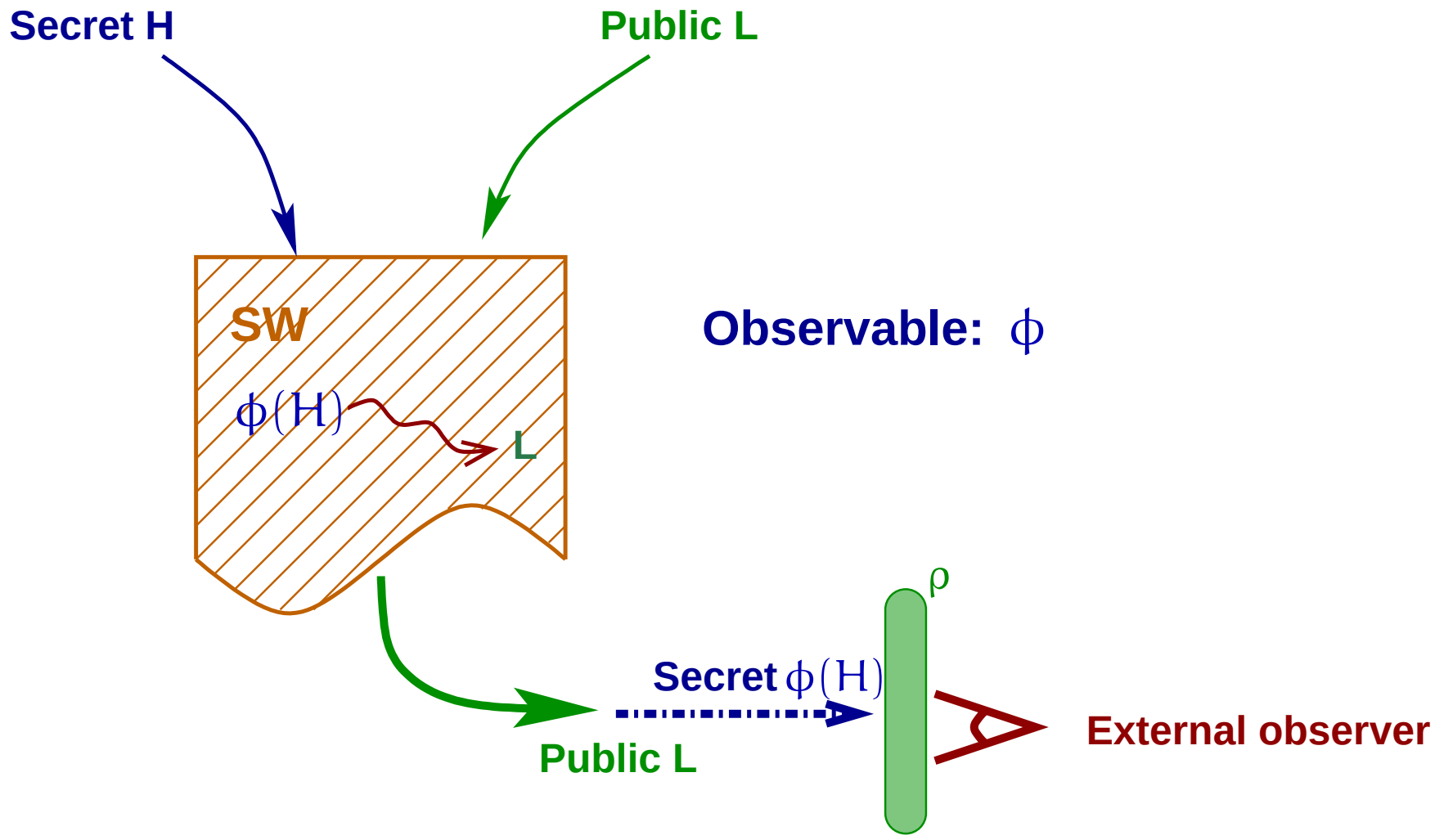
Abstract Non-Interference

[Giacobazzi & Mastroeni '04]



Abstract Non-Interference

[Giacobazzi & Mastroeni '04]



AI: Lattice of Abstractions

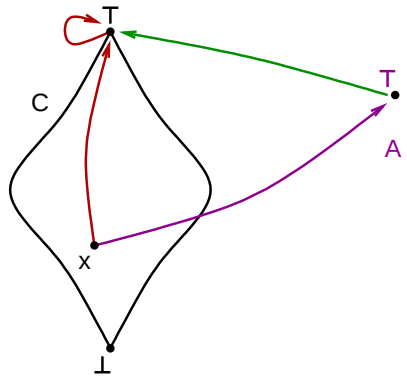
The concrete domain $< C, \leq, \wedge, \vee, \bot, \top >$ [Cousot & Cousot '79]

Lattice of abstract domains $\equiv Abs(C)$

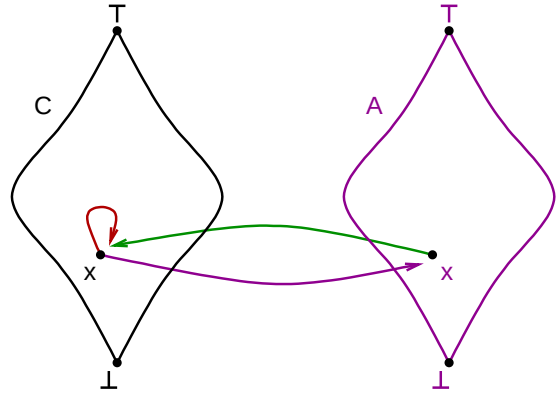
$< Abs(C), \sqsubseteq, \sqcap, \sqcup, \top, C >$

$A_1 \sqsubseteq A_2 \quad \Leftrightarrow \quad A_2 \subseteq A_1$ (A_1 more precise than A_2)

Top:



Bottom:



The dimensions of Non-Interference

We distinguish two points of view:

- WHO observes the information flows?
- WHAT information flows?

The dimensions of Non-Interference

We distinguish two points of view:

- WHO observes the information flows?

  How can we weaken non-interference by characterizing the observational capability of *who* observes?

  - By means of *Equivalence relations*: PER MODEL, ROBUST DECLASSIFICATION;
  - By means of *Abstract domains*: ABSTRACT NON-INTERFERENCE;

- WHAT information flows?

The dimensions of Non-Interference

We distinguish two points of view:

- WHO observes the information flows?

  How can we weaken non-interference by characterizing the observational capability of *who* observes?

- WHAT information flows?

  How can we weaken non-interference by characterizing *what* of the private information flows?

The dimensions of Non-Interference

We distinguish two points of view:

- WHO observes the information flows?

  How can we weaken non-interference by characterizing the observational capability of *who* observes?

- WHAT information flows?

  How can we weaken non-interference by characterizing *what* of the private information flows?

  - *Declassifying* what *can* flow: SELECTIVE DEPENDENCY, ENFORCING ROBUST DECLASSIFICATION, ABSTRACT NON-INTERFERENCE, DELIMITED RELEASE, RELAXED NONINTERFERENCE;

The dimensions of Non-Interference

We distinguish two points of view:

- WHO observes the information flows?

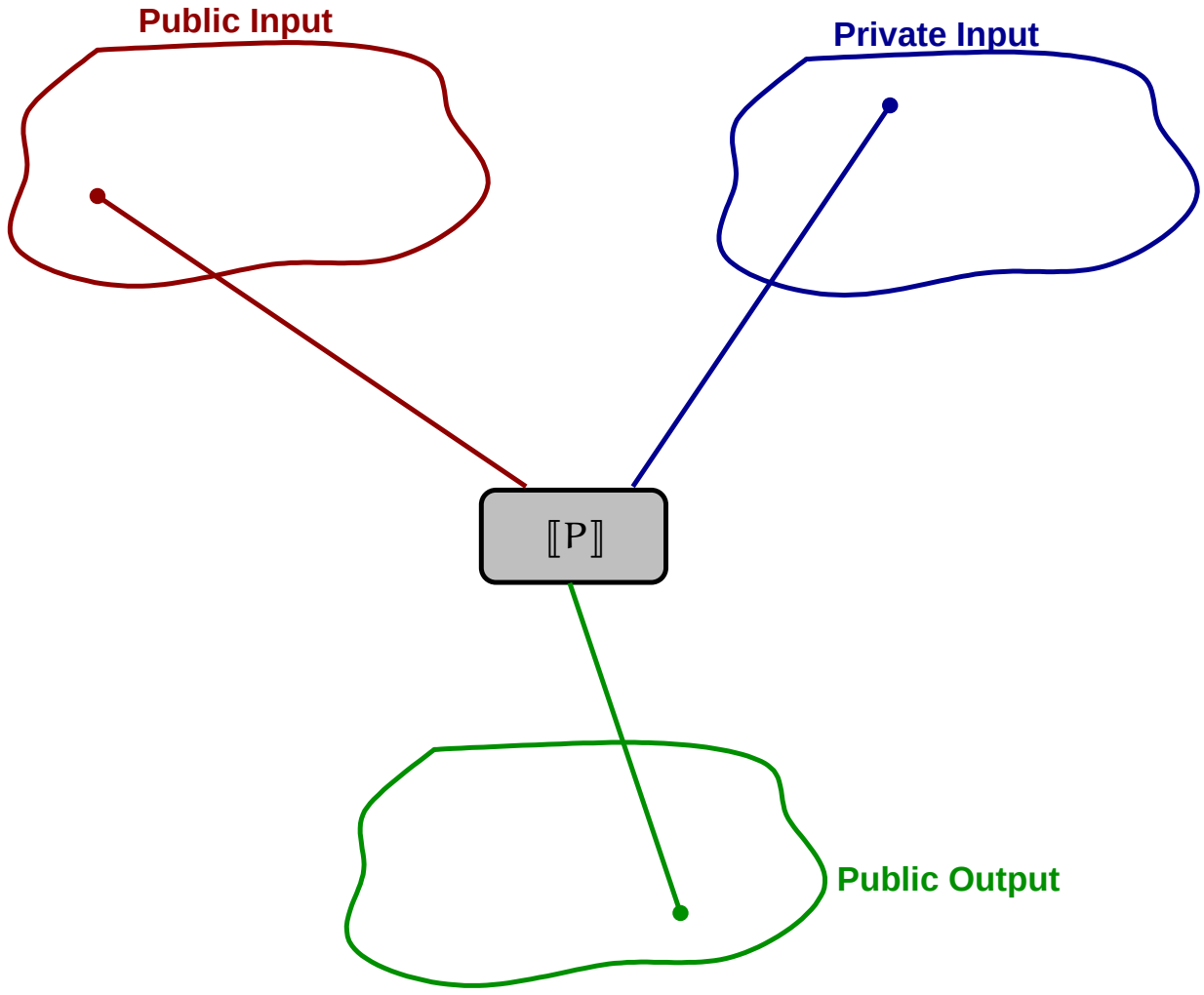
  How can we weaken non-interference by characterizing the observational capability of *who* observes?

- WHAT information flows?

  How can we weaken non-interference by characterizing *what* of the private information flows?

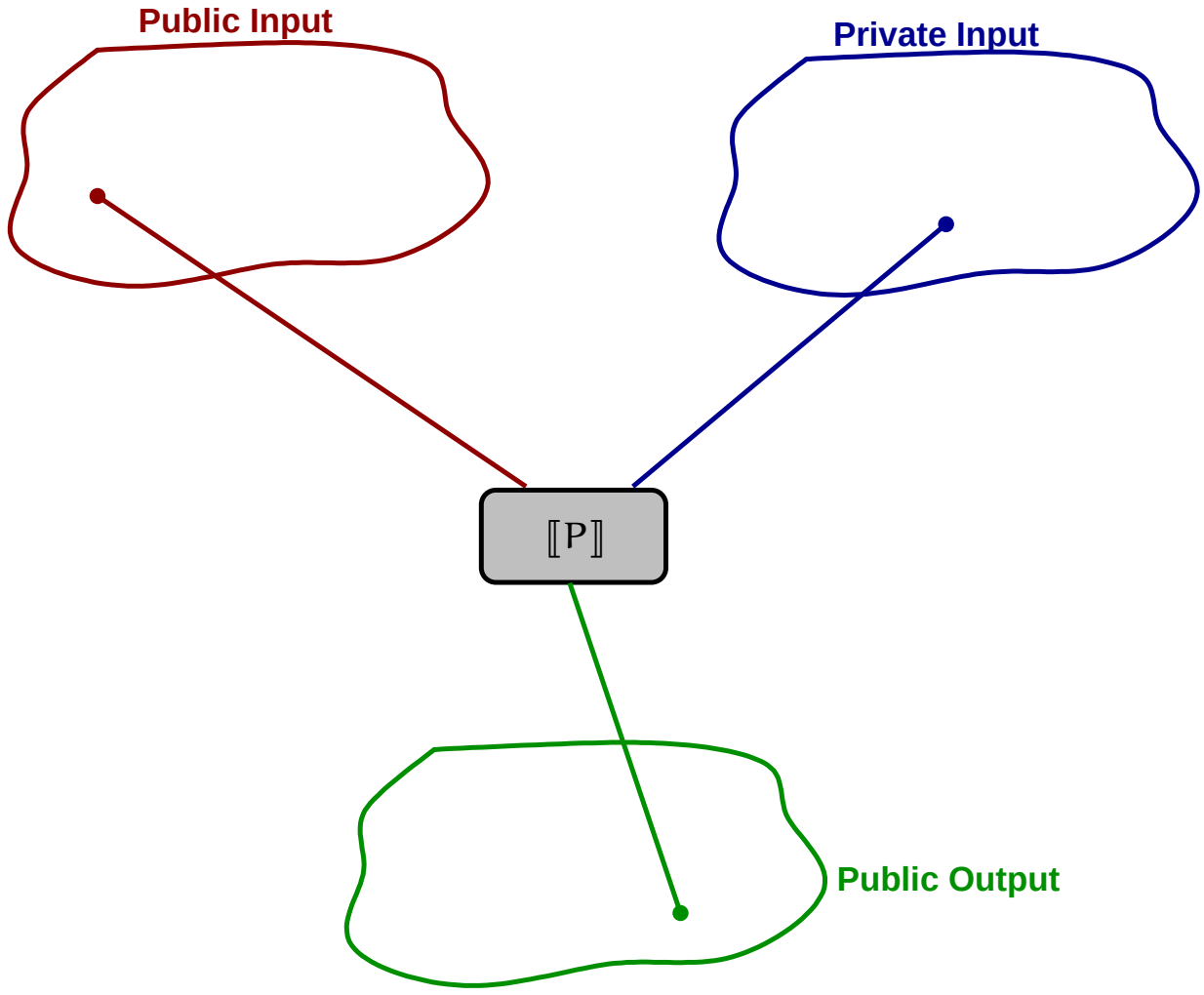
  - *Declassifying* what *can* flow: SELECTIVE DEPENDENCY, ENFORCING ROBUST DECLASSIFICATION, ABSTRACT NON-INTERFERENCE, DELIMITED RELEASE, RELAXED NONINTERFERENCE;
  - *Classifying* what *cannot* flow: ABSTRACT NON-INTERFERENCE.

Standard non-interference



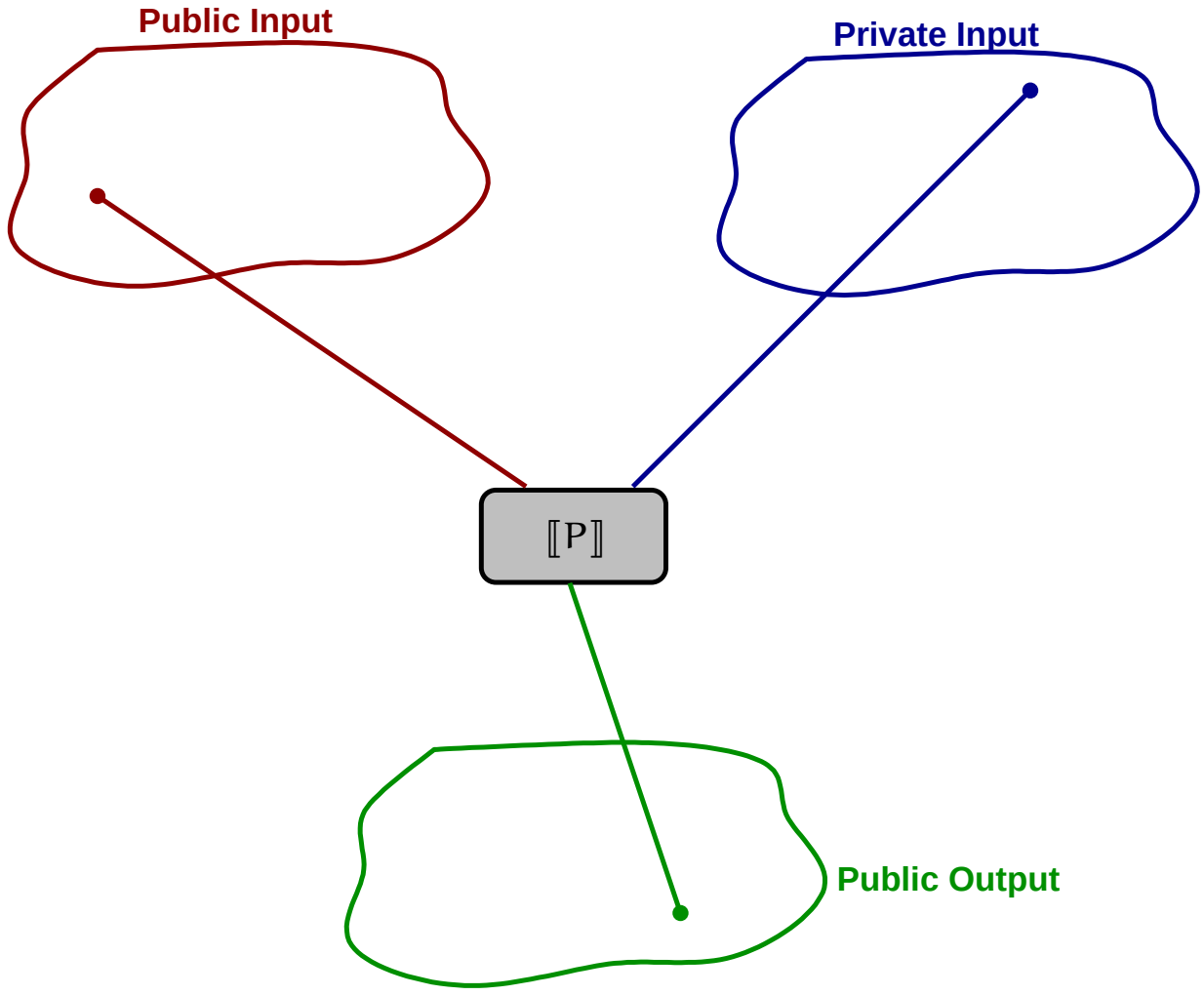
$$\forall l : \mathsf{L}, \forall h_1, h_2 : \mathsf{H}.\ [\![P]\!](h_1, l)^{\mathsf{L}} = [\![P]\!](h_2, l)^{\mathsf{L}}$$

Standard non-interference



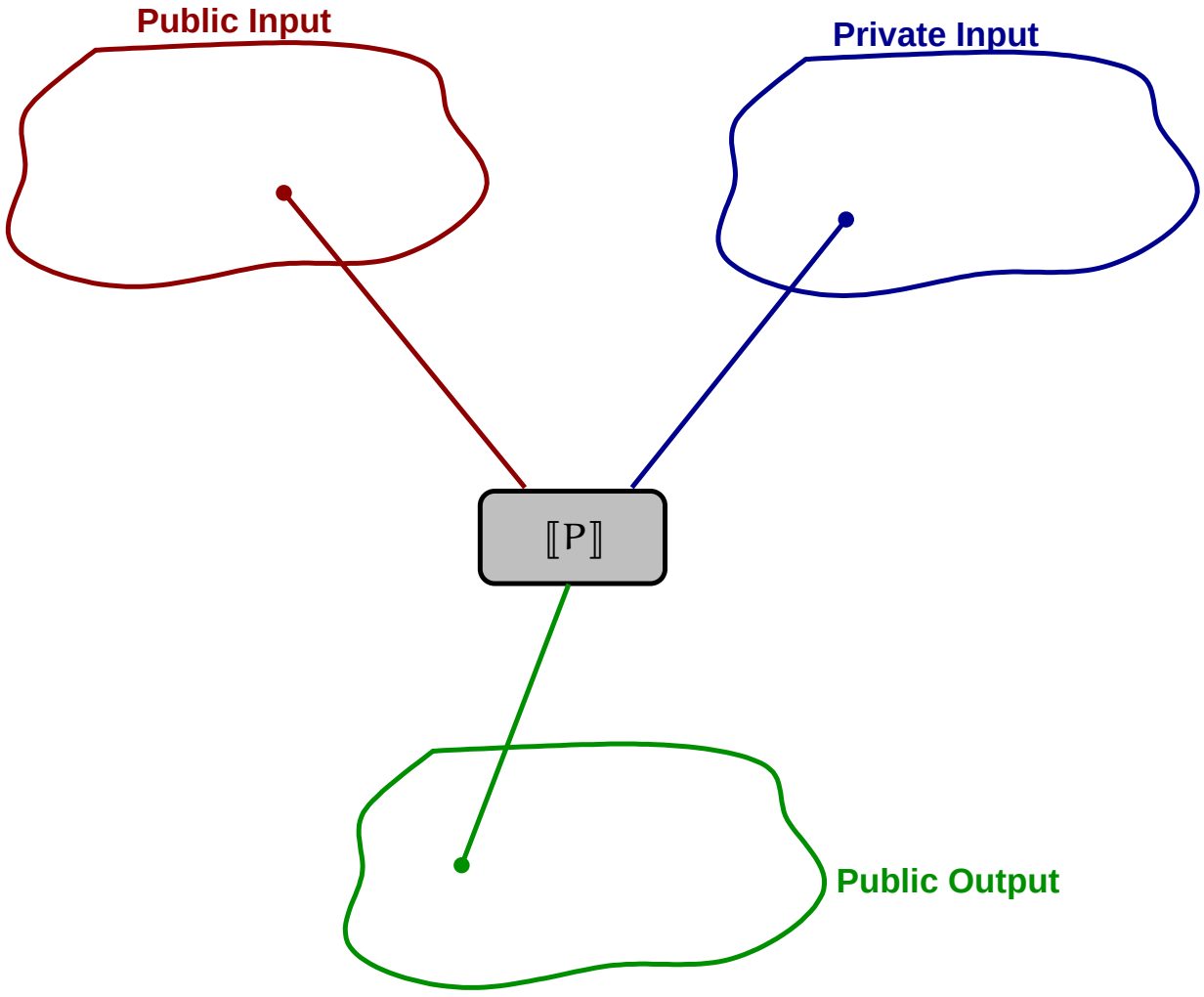
$$\forall l : \mathsf{L}, \forall h_1, h_2 : \mathsf{H}.\ [\![P]\!](h_1, l)^{\mathsf{L}} = [\![P]\!](h_2, l)^{\mathsf{L}}$$

Standard non-interference



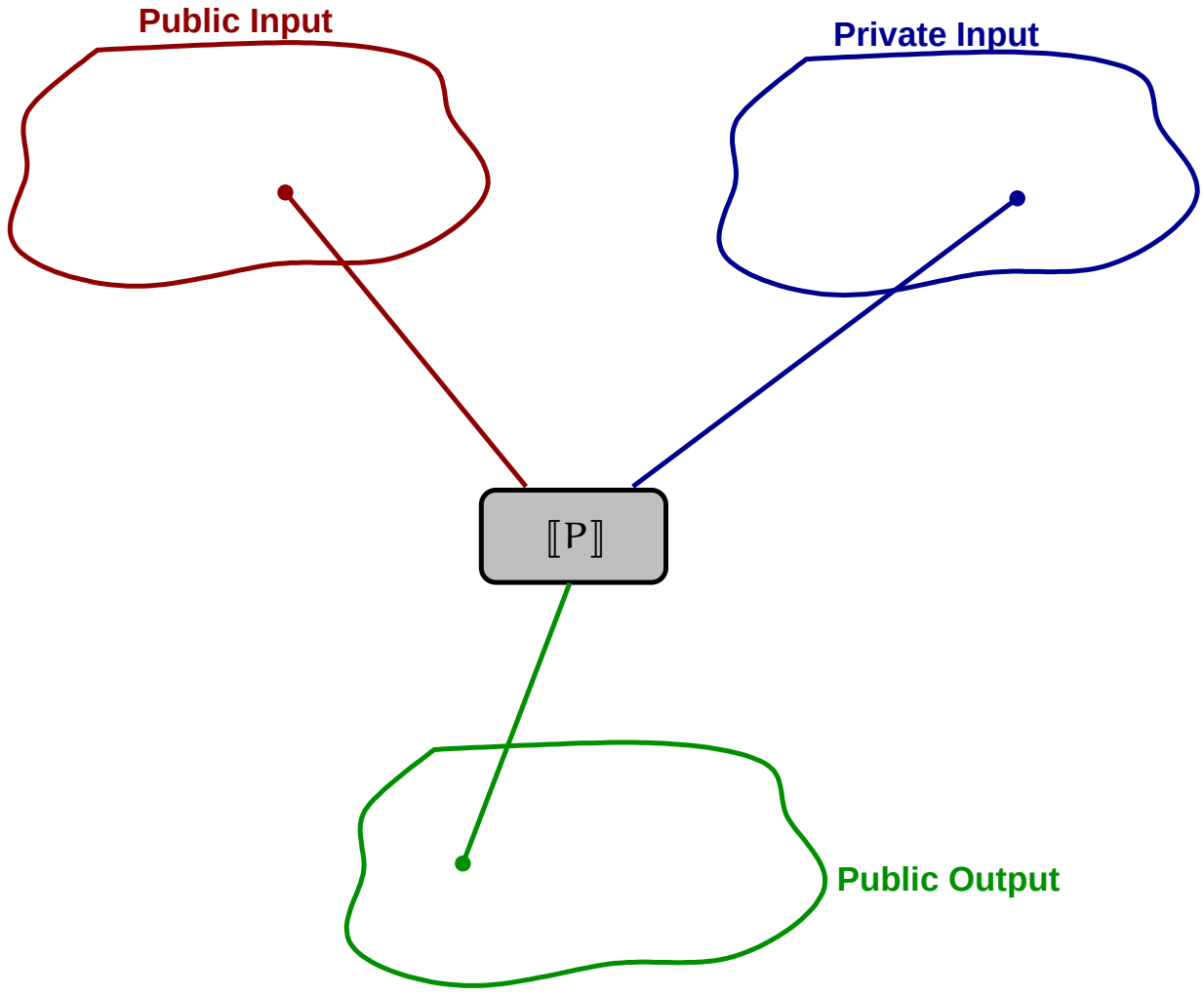
$$\forall l : L, \forall h_1, h_2 : H.\ [\![P]\!](h_1, l)^L = [\![P]\!](h_2, l)^L$$

Standard non-interference



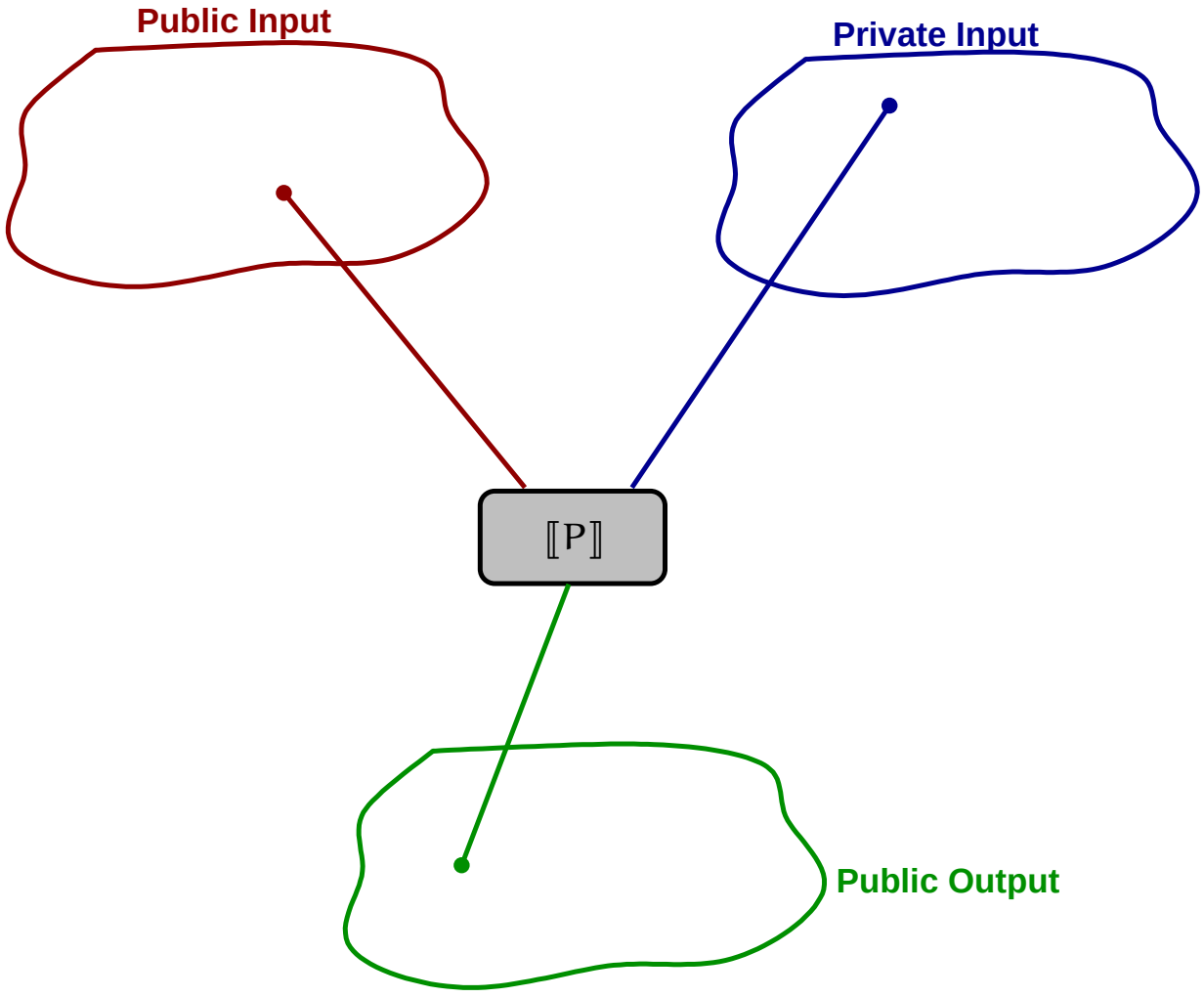
$$\forall l : \mathsf{L}, \forall h_1, h_2 : \mathsf{H}.\ [\![P]\!](h_1, l)^{\mathsf{L}} = [\![P]\!](h_2, l)^{\mathsf{L}}$$

Standard non-interference



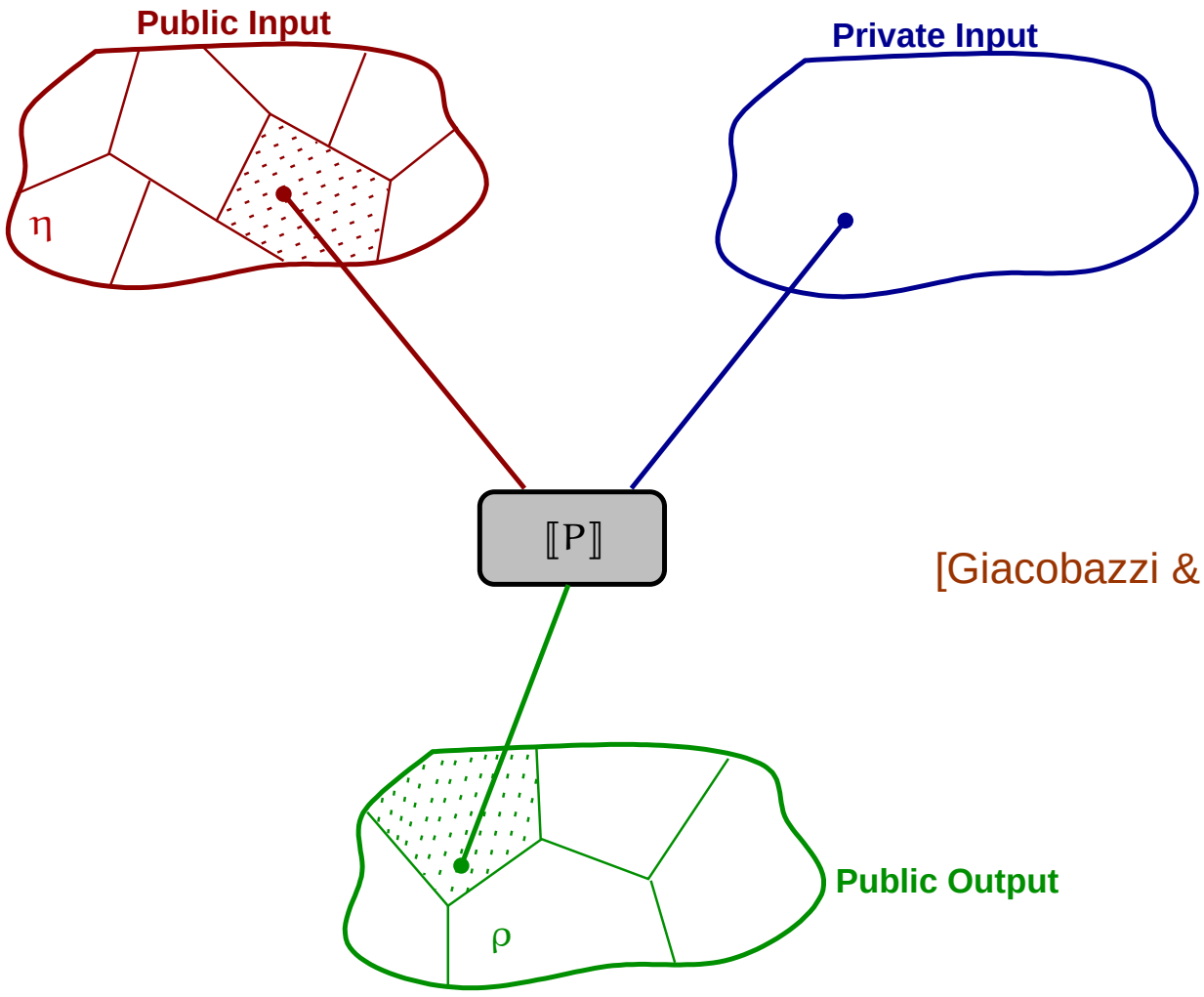
$$\forall l : \mathsf{L}, \forall h_1, h_2 : \mathsf{H}.\ [\![P]\!](h_1, l)^{\mathsf{L}} = [\![P]\!](h_2, l)^{\mathsf{L}}$$

Standard non-interference



$$\forall l : \mathsf{L}, \forall h_1, h_2 : \mathsf{H}.\ [\![P]\!](h_1, l)^{\mathsf{L}} = [\![P]\!](h_2, l)^{\mathsf{L}}$$

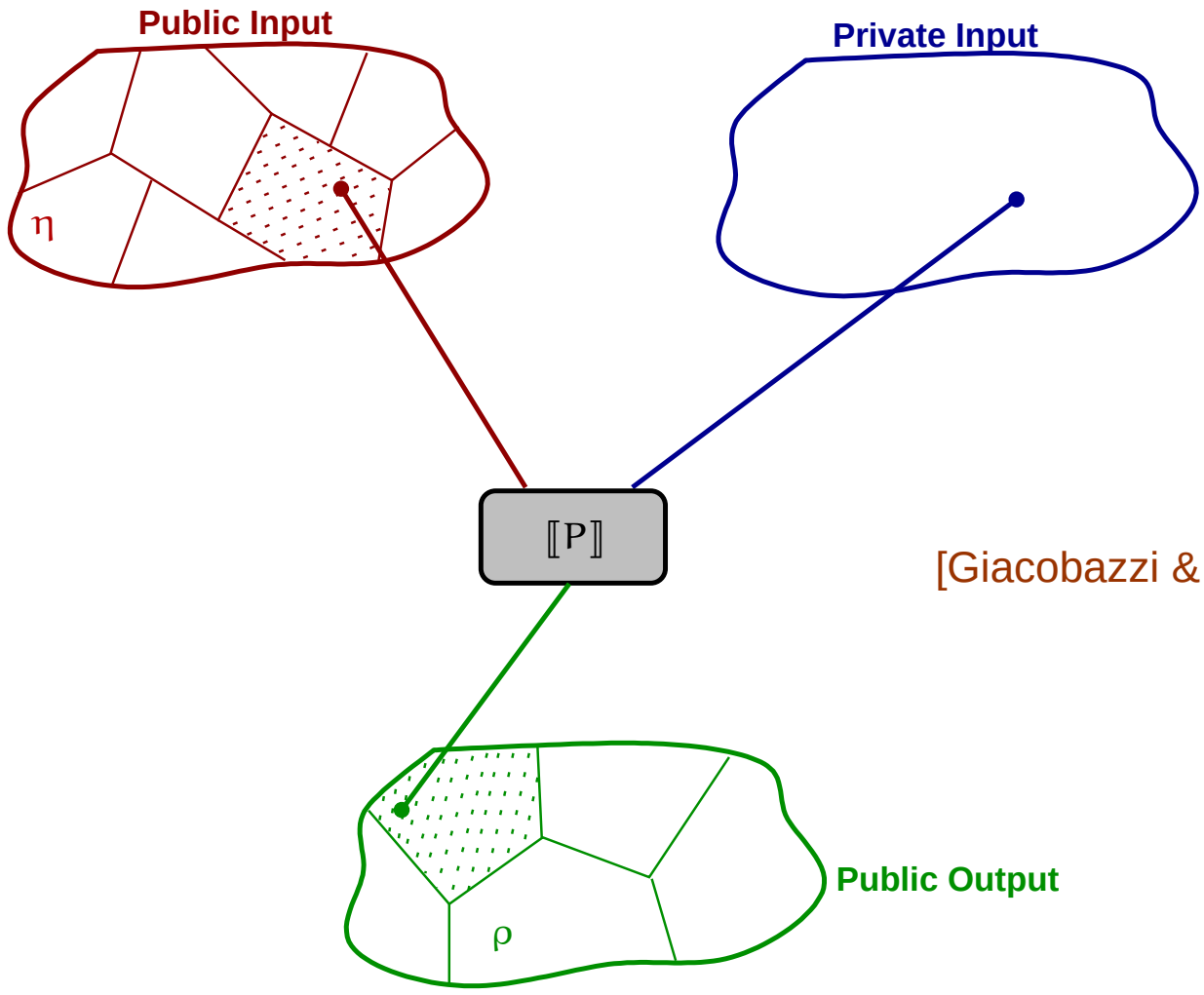
(Narrow) Abstract Non-Interference



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})): [\eta]\mathrm{P}(\rho): \eta(l_1) = \eta(l_2) \Rightarrow \rho([\![\mathrm{P}]\!](h_1, l_1)^{\mathsf{L}}) = \rho([\![\mathrm{P}]\!](h_2, l_2)^{\mathsf{L}})$$

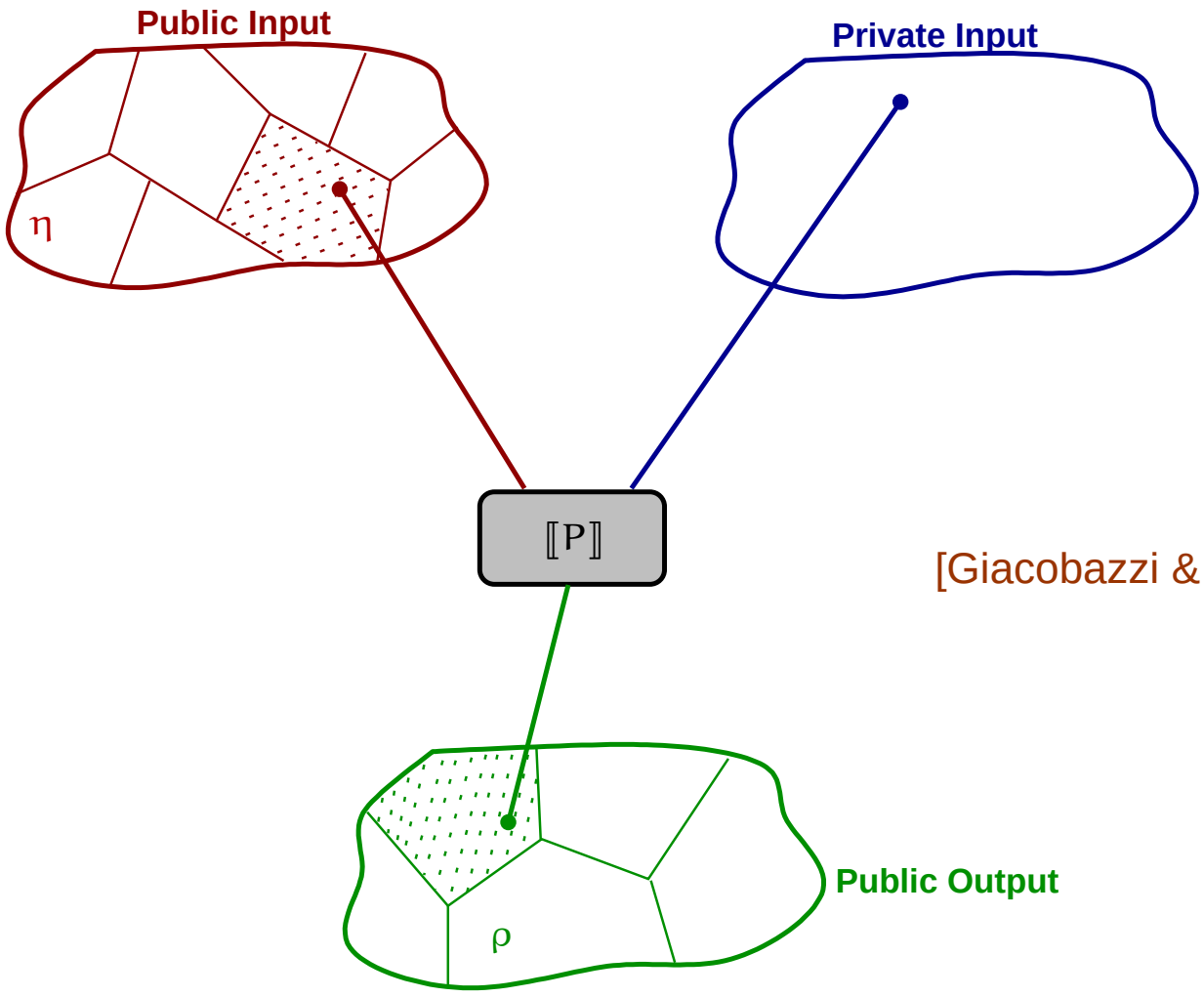
(Narrow) Abstract Non-Interference



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})): [\eta]\mathrm{P}(\rho): \eta(l_1) = \eta(l_2) \Rightarrow \rho([\![\mathrm{P}]\!](h_1, l_1)^{\mathsf{L}}) = \rho([\![\mathrm{P}]\!](h_2, l_2)^{\mathsf{L}})$$

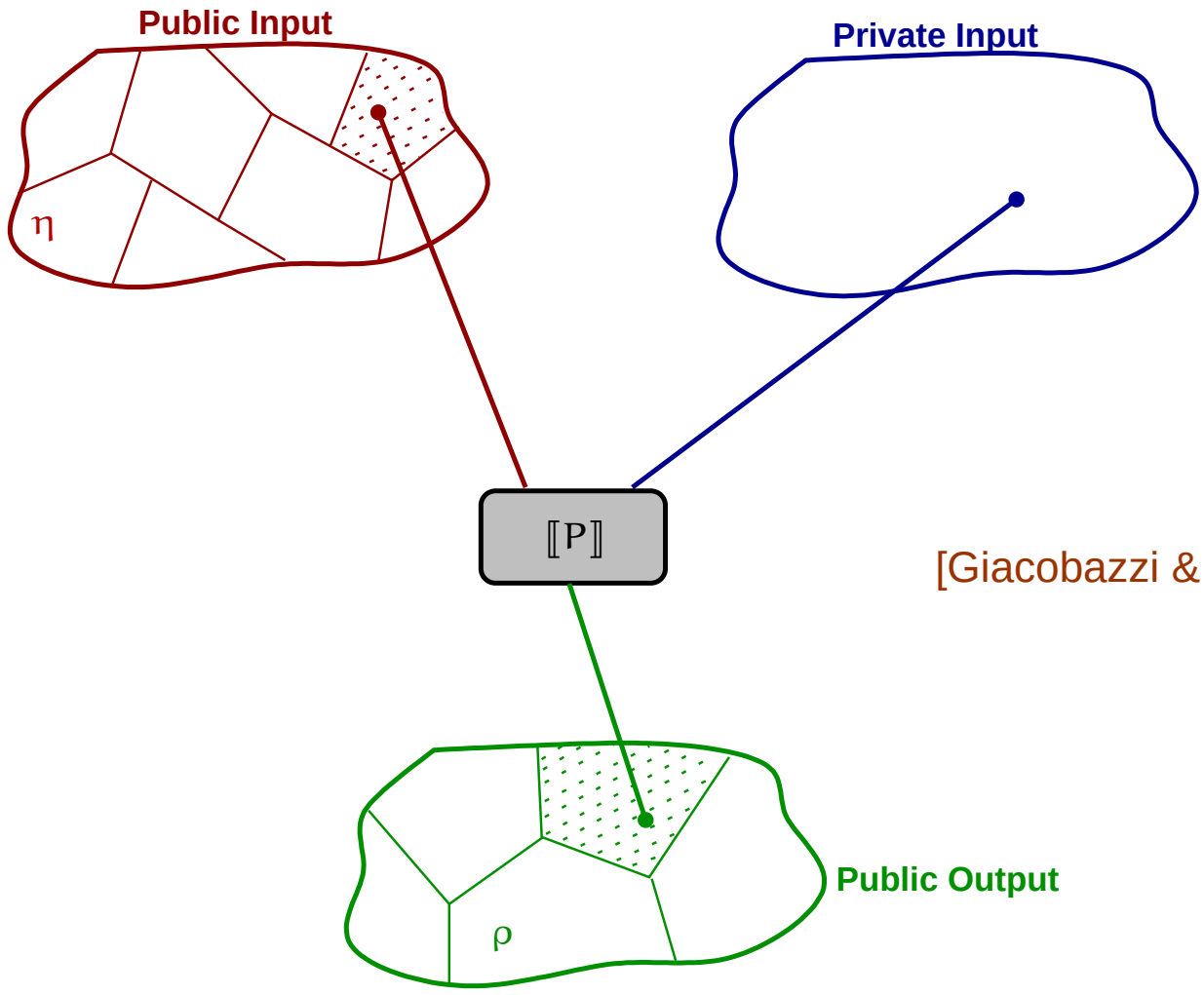
(Narrow) Abstract Non-Interference



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})): [\eta]\mathrm{P}(\rho): \eta(l_1) = \eta(l_2) \Rightarrow \rho([\![\mathrm{P}]\!](h_1, l_1)^{\mathsf{L}}) = \rho([\![\mathrm{P}]\!](h_2, l_2)^{\mathsf{L}})$$

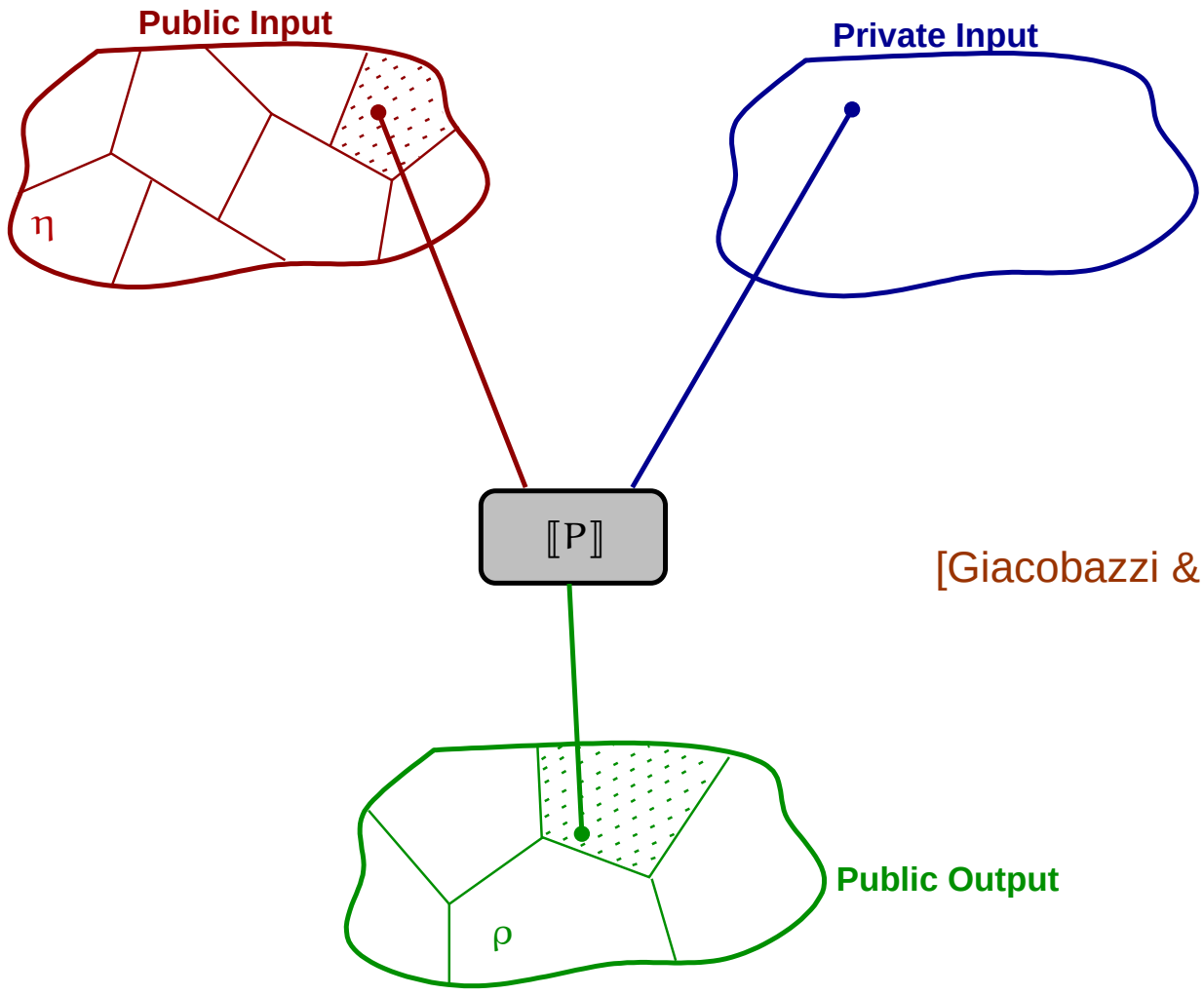
(Narrow) Abstract Non-Interference



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})) \colon [\eta]\mathrm{P}(\rho) \colon \eta(\mathrm{l}_1) = \eta(\mathrm{l}_2) \Rightarrow \rho(\llbracket \mathrm{P} \rrbracket(\mathrm{h}_1, \mathrm{l}_1)^{\mathsf{L}}) = \rho(\llbracket \mathrm{P} \rrbracket(\mathrm{h}_2, \mathrm{l}_2)^{\mathsf{L}})$$

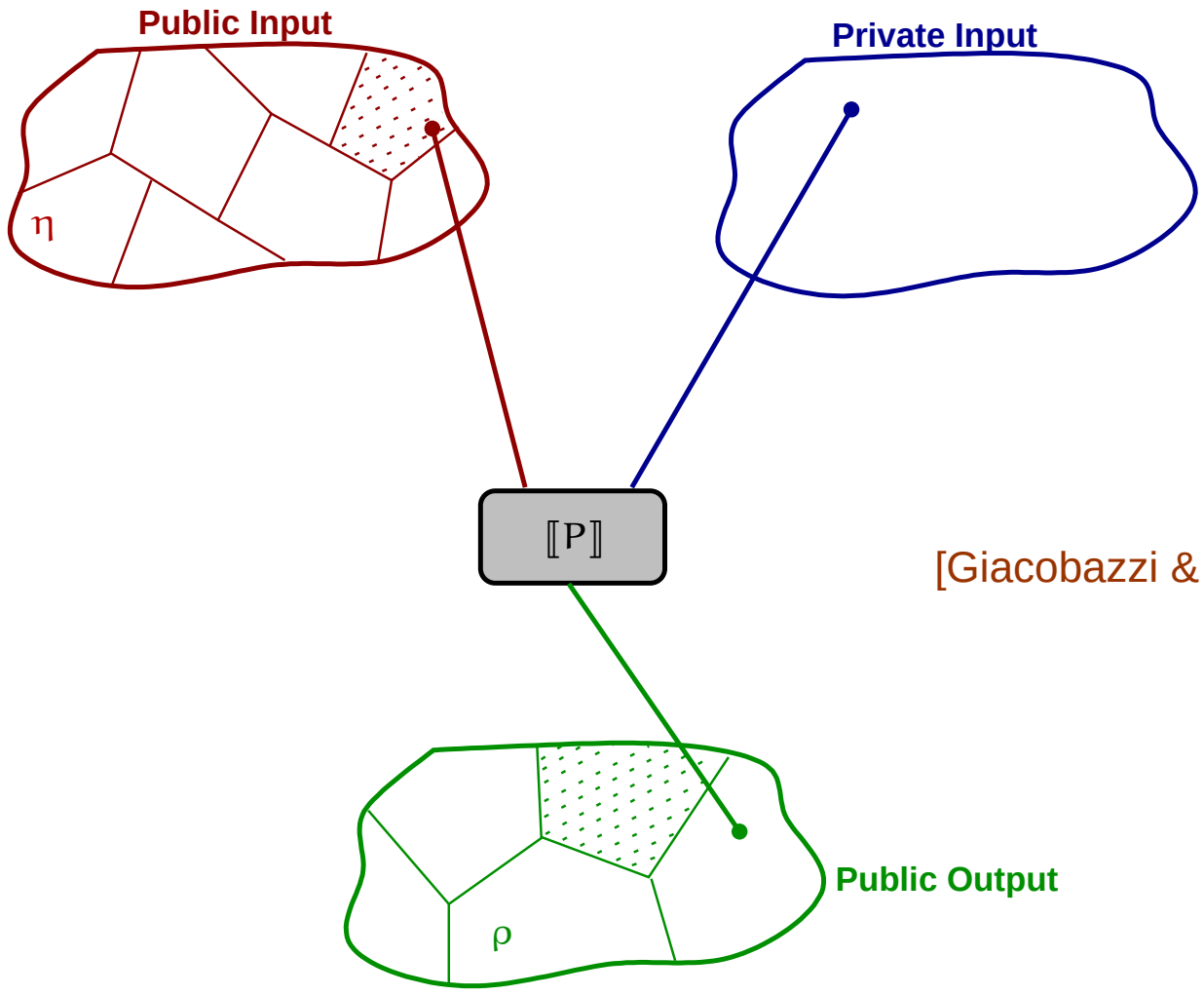
(Narrow) Abstract Non-Interference



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})): [\eta]\mathrm{P}(\rho): \eta(l_1) = \eta(l_2) \Rightarrow \rho(\llbracket \mathrm{P} \rrbracket(h_1, l_1)^{\mathsf{L}}) = \rho(\llbracket \mathrm{P} \rrbracket(h_2, l_2)^{\mathsf{L}})$$

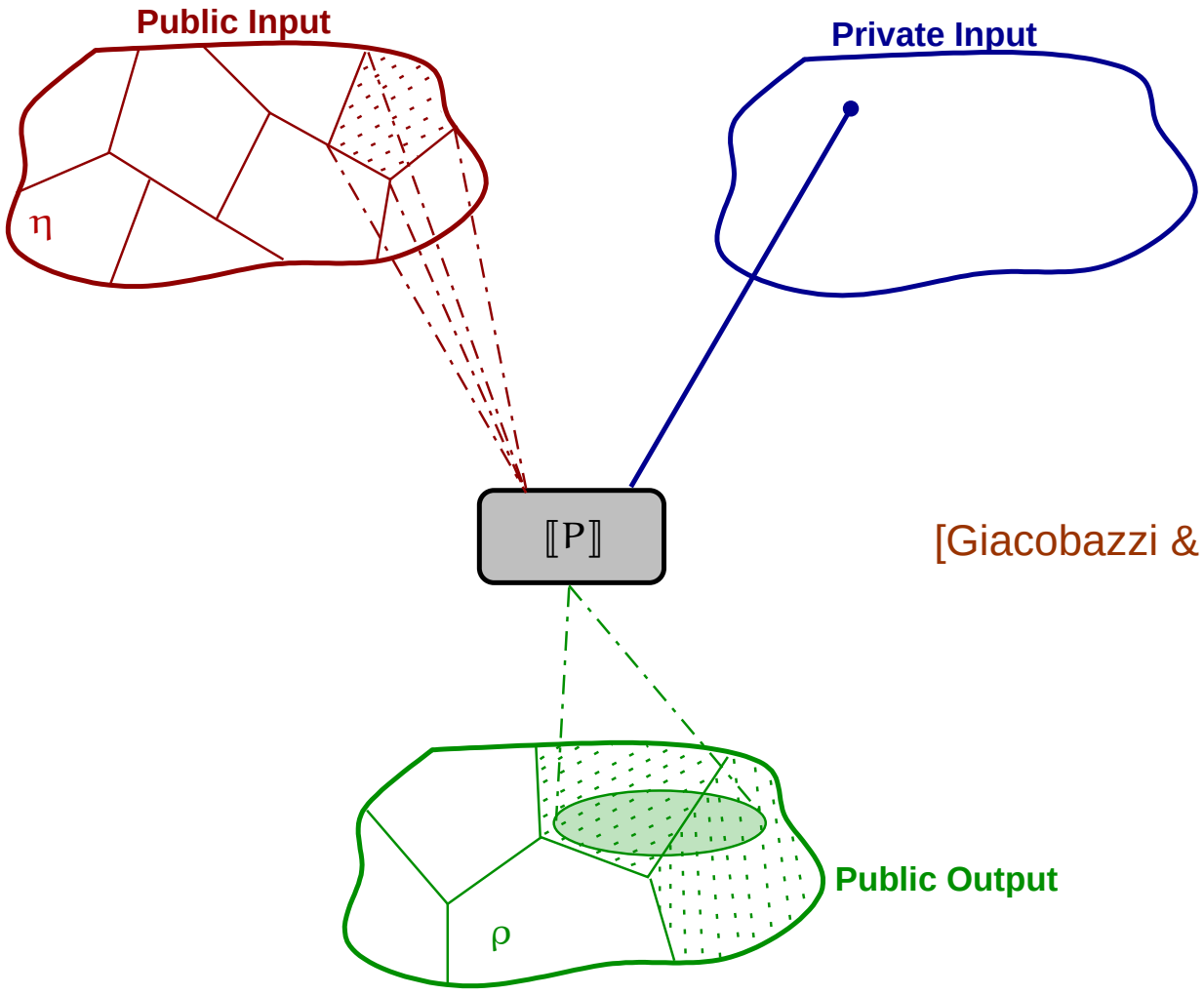
(Narrow) Abstract Non-Interference



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})): [\eta]\mathrm{P}(\rho): \eta(l_1) = \eta(l_2) \Rightarrow \rho([\![\mathrm{P}]\!](h_1, l_1)^{\mathsf{L}}) = \rho([\![\mathrm{P}]\!](h_2, l_2)^{\mathsf{L}})$$

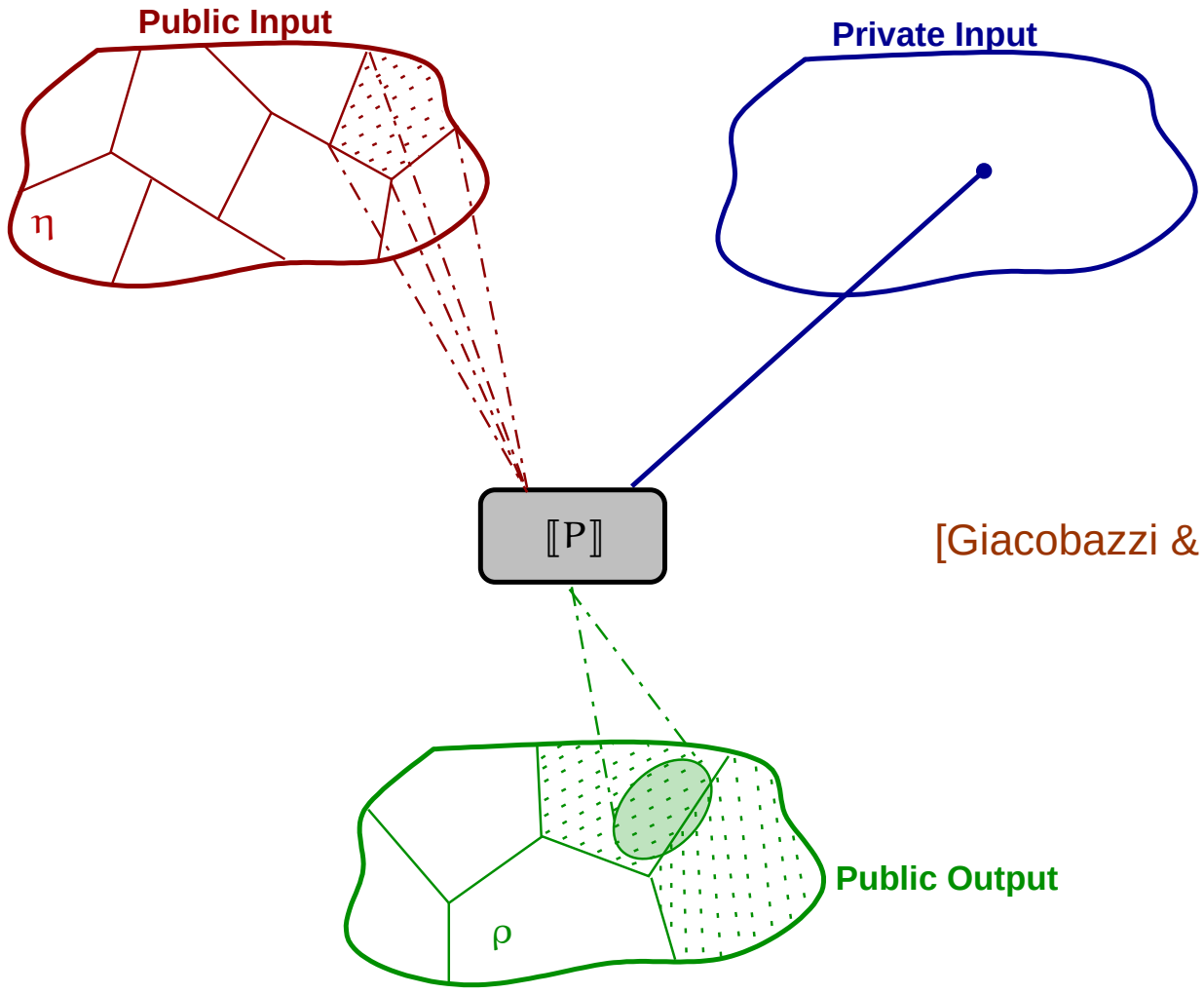
Abstract Non-Interference (ANI)



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})) \colon (\eta)\mathrm{P}(\rho) \colon$$
$$\eta(l_1) = \eta(l_2) \Rightarrow \rho([\![\mathrm{P}]\!](h_1, \eta(l_1))^{\mathsf{L}}) = \rho([\![\mathrm{P}]\!](h_2, \eta(l_2))^{\mathsf{L}})$$

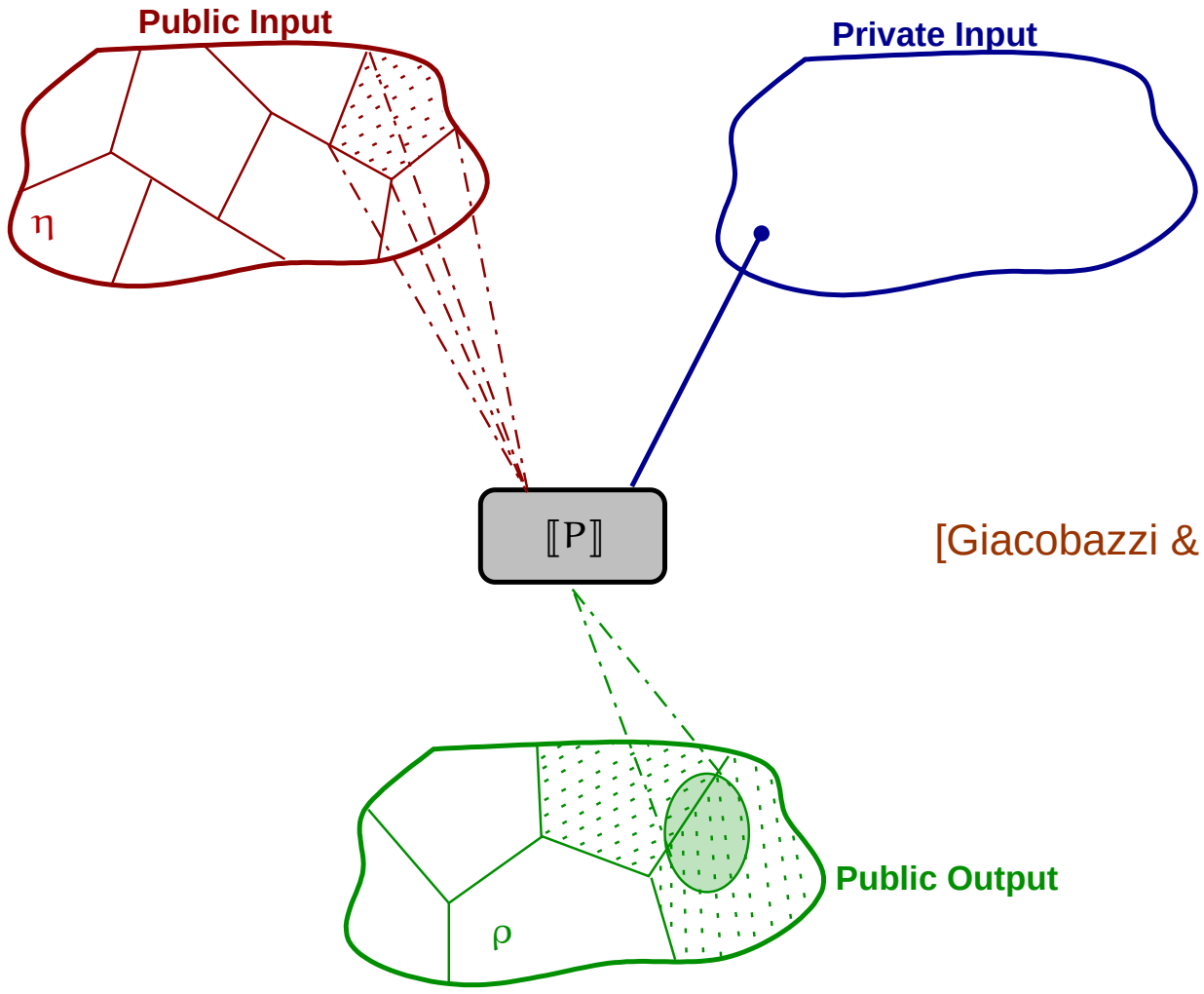
Abstract Non-Interference (ANI)



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})) \colon (\eta)\mathrm{P}(\rho) \colon$$
$$\eta(l_1) = \eta(l_2) \Rightarrow \rho([\![\mathrm{P}]\!](h_1, \eta(l_1))^{\mathsf{L}}) = \rho([\![\mathrm{P}]\!](h_2, \eta(l_2))^{\mathsf{L}})$$

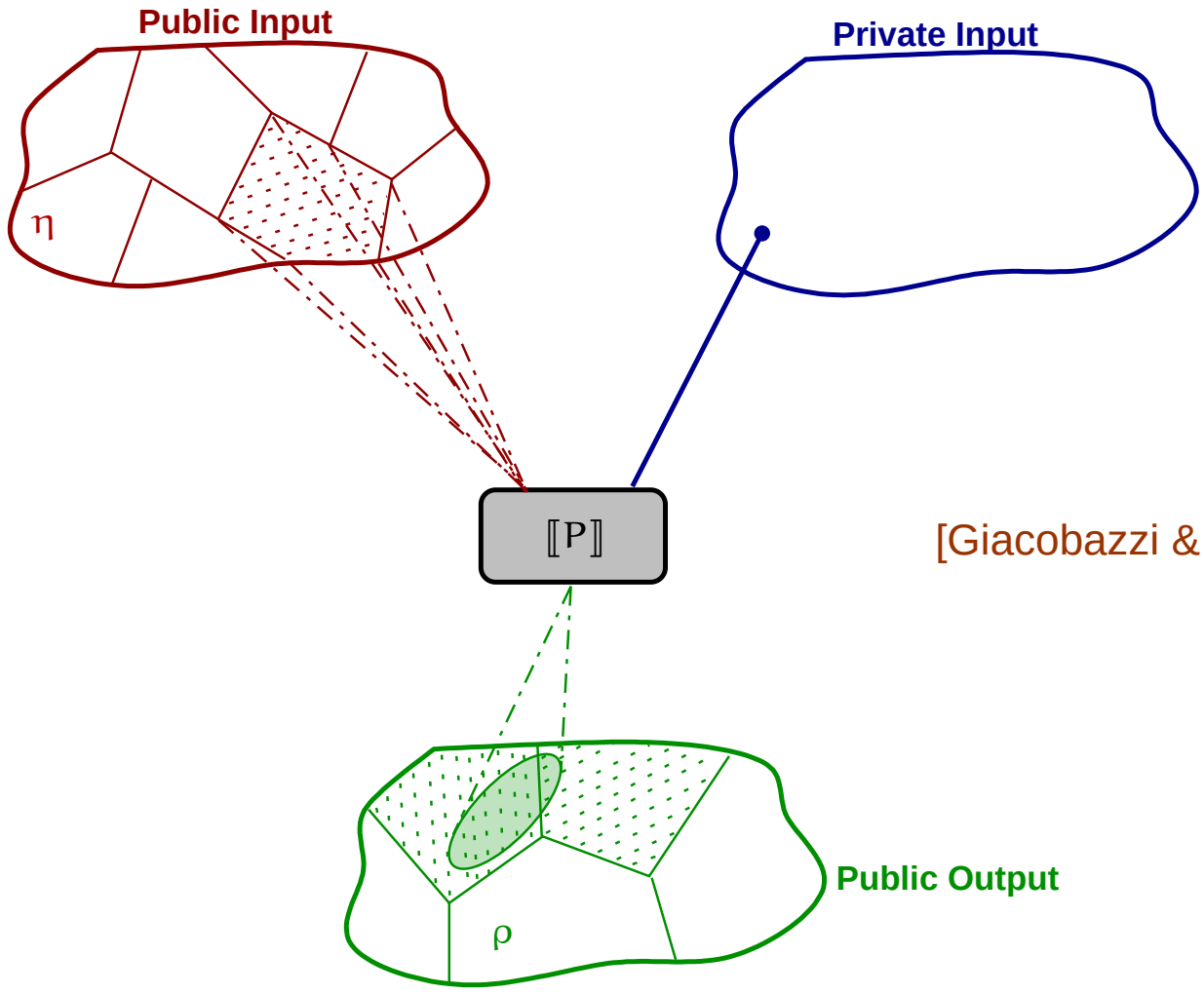
Abstract Non-Interference (ANI)



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})) \colon (\eta)\mathrm{P}(\rho) \colon$$
$$\eta(l_1) = \eta(l_2) \Rightarrow \rho([\![\mathrm{P}]\!](h_1, \eta(l_1))^{\mathsf{L}}) = \rho([\![\mathrm{P}]\!](h_2, \eta(l_2))^{\mathsf{L}})$$

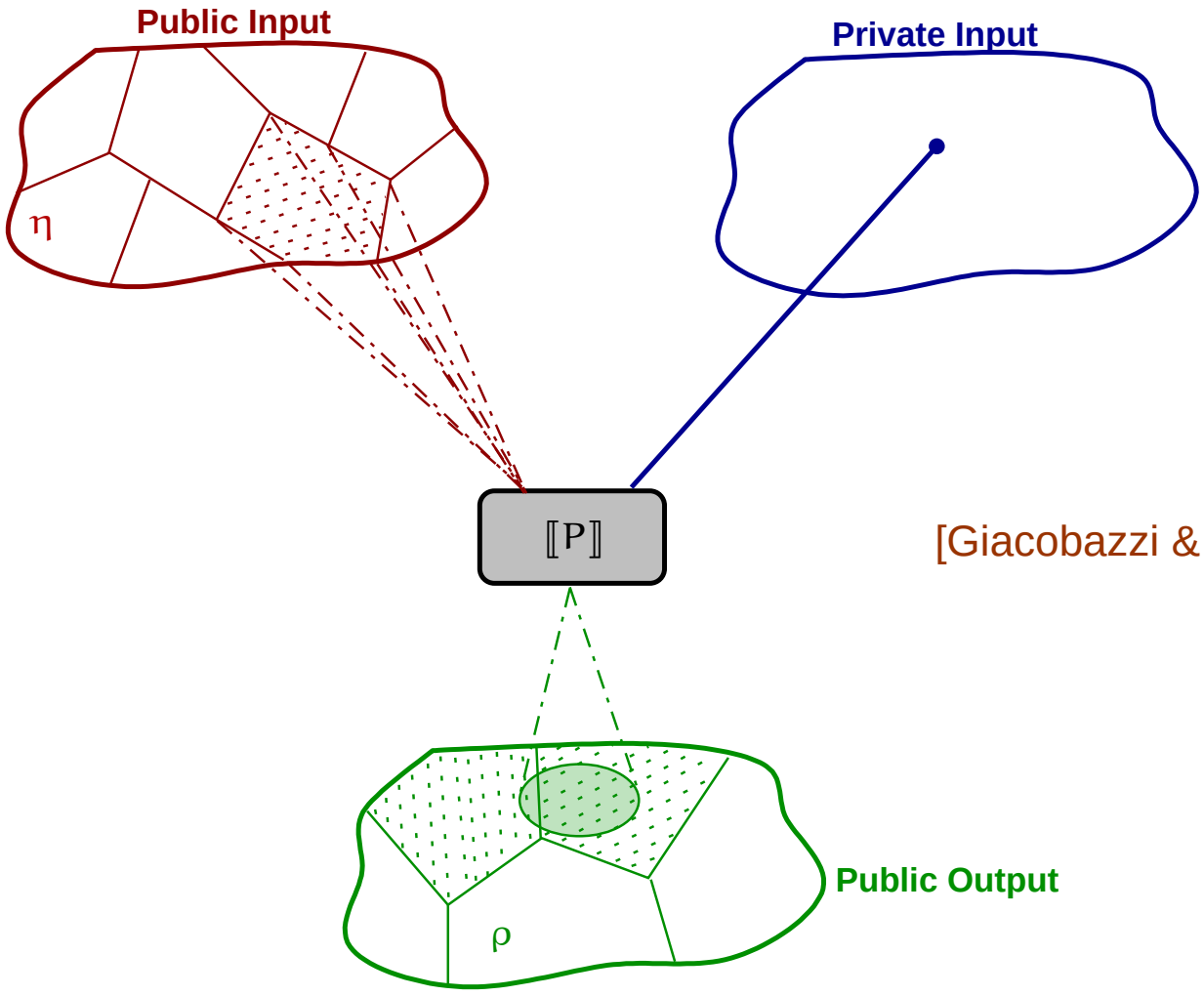
Abstract Non-Interference (ANI)



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})) \colon (\eta)\mathrm{P}(\rho) \colon$$
$$\eta(l_1) = \eta(l_2) \Rightarrow \rho([\![\mathrm{P}]\!](h_1, \eta(l_1))^{\mathsf{L}}) = \rho([\![\mathrm{P}]\!](h_2, \eta(l_2))^{\mathsf{L}})$$

Abstract Non-Interference (ANI)



[Giacobazzi & Mastroeni '04]

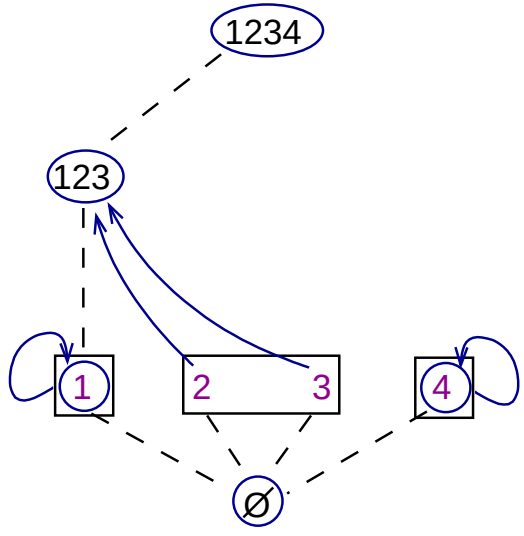
$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})) \colon (\eta)\mathrm{P}(\rho) \colon$$
$$\eta(l_1) = \eta(l_2) \Rightarrow \rho([\![\mathrm{P}]\!](h_1, \eta(l_1))^{\mathsf{L}}) = \rho([\![\mathrm{P}]\!](h_2, \eta(l_2))^{\mathsf{L}})$$

ANI vs Other Approaches

- PER MODEL OF ANI: [Hunt & Mastroeni ’05]

Partitioning closures

[Ranzato and Tapparo ’04]

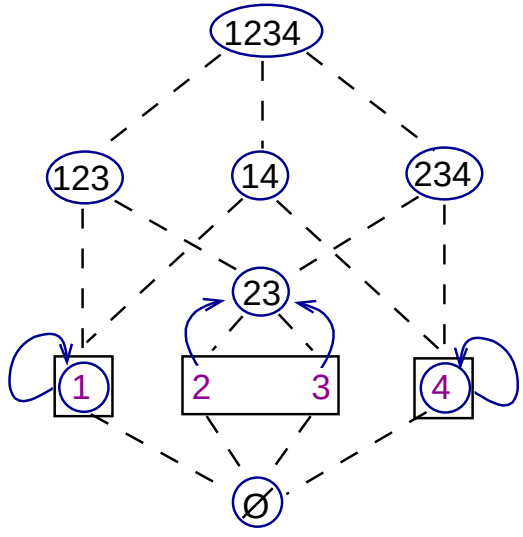


ANI vs Other Approaches

- PER MODEL OF ANI: [Hunt & Mastroeni '05]

Partitioning closures

[Ranzato and Tapparo '04]



$\Pi(\eta)$ is the most concrete partitioning closure containing η!

ANI vs Other Approaches

- PER MODEL OF ANI: [Hunt & Mastroeni '05]

$\Pi(\eta)$ is the most concrete partitioning closure containing η!

$$[\eta]\mathrm{P}(\rho) \quad \text{iff} \quad [\![\mathrm{P}]\!] : \mathit{All} \times \mathit{Rel}^{\eta} \rightarrowtriangle \mathit{All} \times \mathit{Rel}^{\rho}$$
$$\text{iff} \quad [\Pi(\eta)]\mathrm{P}(\Pi(\rho))$$

ANI vs Other Approaches

- PER MODEL OF ANI: [Hunt & Mastroeni '05]

$\Pi(\eta)$ is the most concrete partitioning closure containing η!

$$[\eta]\mathrm{P}(\rho) \quad \text{iff} \quad [\![\mathrm{P}]\!] : \mathit{All} \times \mathit{Rel}^{\eta} \rightarrowtriangle \mathit{All} \times \mathit{Rel}^{\rho}$$
$$\text{iff} \quad [\Pi(\eta)]\mathrm{P}(\Pi(\rho))$$

$$\Downarrow$$

$$[\mathrm{R}]\mathrm{P}(\mathrm{S}) \text{ iff } [\![\mathrm{P}]\!] : \mathit{All} \times \mathrm{R} \rightarrowtriangle \mathit{All} \times \mathrm{S}$$

ANI vs Other Approaches

- PER MODEL OF ANI: [Hunt & Mastroeni '05]

$\Pi(\eta)$ is the most concrete partitioning closure containing η!

$$[\eta]P(\rho) \quad \text{iff} \quad [\![P]\!] : \mathit{All} \times \mathit{Rel}^{\eta} \rightarrowtriangle \mathit{All} \times \mathit{Rel}^{\rho}$$
$$\text{iff} \quad [\Pi(\eta)]P(\Pi(\rho))$$

$$[R]P(S) \text{ iff } [\![P]\!] : \mathit{All} \times R \rightarrowtriangle \mathit{All} \times S$$

- ANI VS SECURITY FOR ROBUST DECLASSIFICATION: [Zdancewic & Myers '01]

The PER model of Abstract Non-Interference on the *trace semantics* IS EQUIVALENT to the security property introduced for Robust Declassification!

Declassification: Selective dependency

[Cohen '78]

WE MAY NOT CARE IF OUTPUT VARIABLE b REFLECTS WHETHER INPUT VARIABLE a IS ODD OR EVEN. HOWEVER, WE MIGHT LIKE TO SHOW THAT b DEPENDS UPON a IN NO OTHER WAY.

- STRONG DEPENDENCY: $\exists s_1, s_2$ such that

$$\forall x \neq a. s_1(x) = s_2(x) \ \wedge \ [\![P]\!](s_1)(b) \neq [\![P]\!](s_2)(b)$$

- SELECTIVE DEPENDENCY: $\exists s_1, s_2$ such that

$$\forall x \neq a. s_1(x) = s_2(x) \wedge \phi(s_1) \ \wedge \ \phi(s_2) \ \wedge \ [\![P]\!](s_1)(b) \neq [\![P]\!](s_2)(b)$$

Declassification: Selective dependency

[Cohen '78]

WE MAY NOT CARE IF OUTPUT VARIABLE b REFLECTS WHETHER INPUT VARIABLE a IS ODD OR EVEN. HOWEVER, WE MIGHT LIKE TO SHOW THAT b DEPENDS UPON a IN NO OTHER WAY.

- STRONG DEPENDENCY: $\exists s_1, s_2$ such that

$$\forall x \neq a. s_1(x) = s_2(x) \ \wedge \ [\![P]\!](s_1)(b) \neq [\![P]\!](s_2)(b)$$

- SELECTIVE DEPENDENCY: $\exists s_1, s_2$ such that

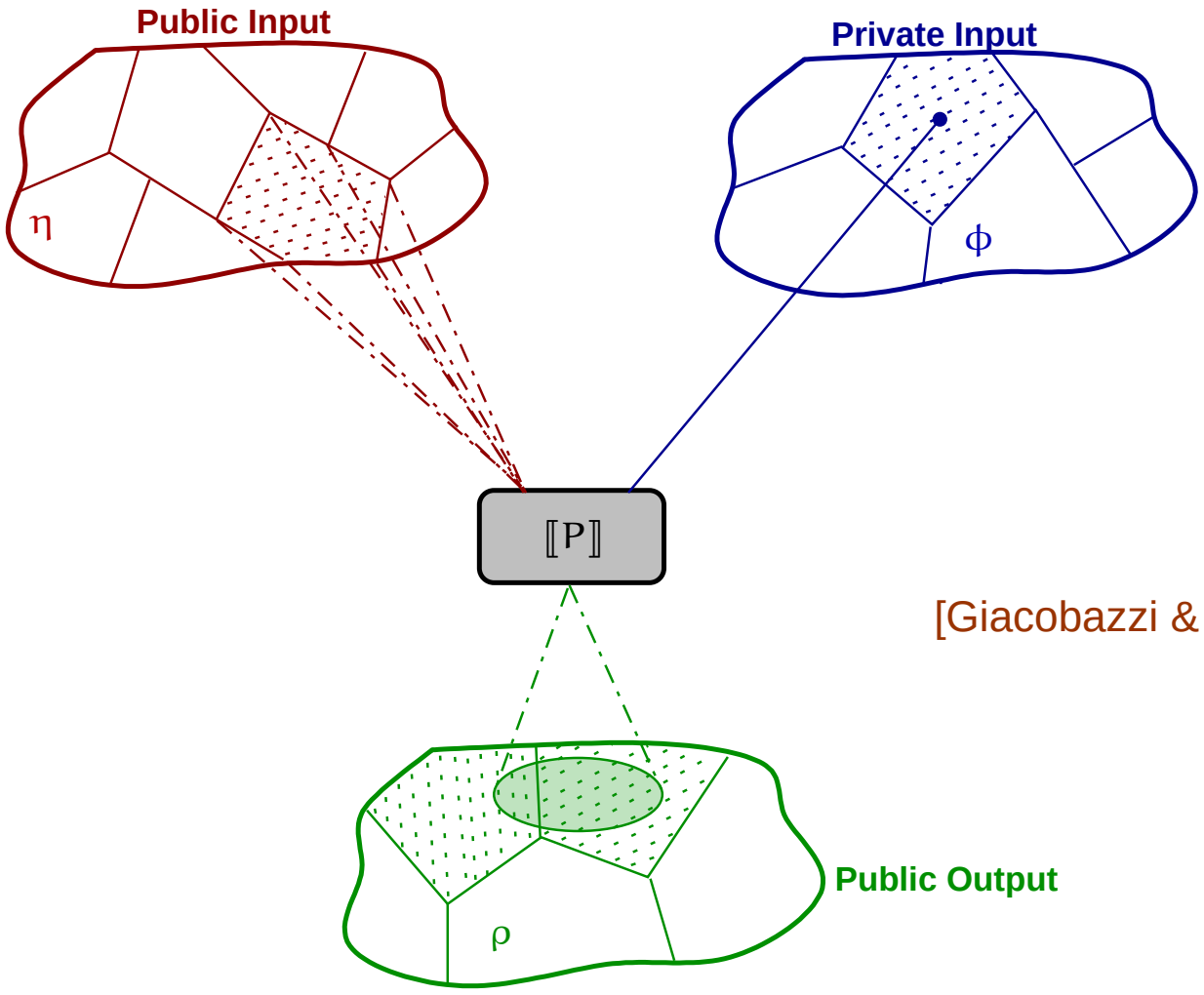
$$\forall x \neq a. s_1(x) = s_2(x) \wedge \phi(s_1) \ \wedge \ \phi(s_2) \ \wedge \ [\![P]\!](s_1)(b) \neq [\![P]\!](s_2)(b)$$

EXAMPLE:

Let $l := x + (h \ \mathit{mod}\ 4)$ and $\phi = h \ \mathit{mod}\ 4 = 3$

$\Rightarrow \phi$ eliminates all the variety that is conveyed.

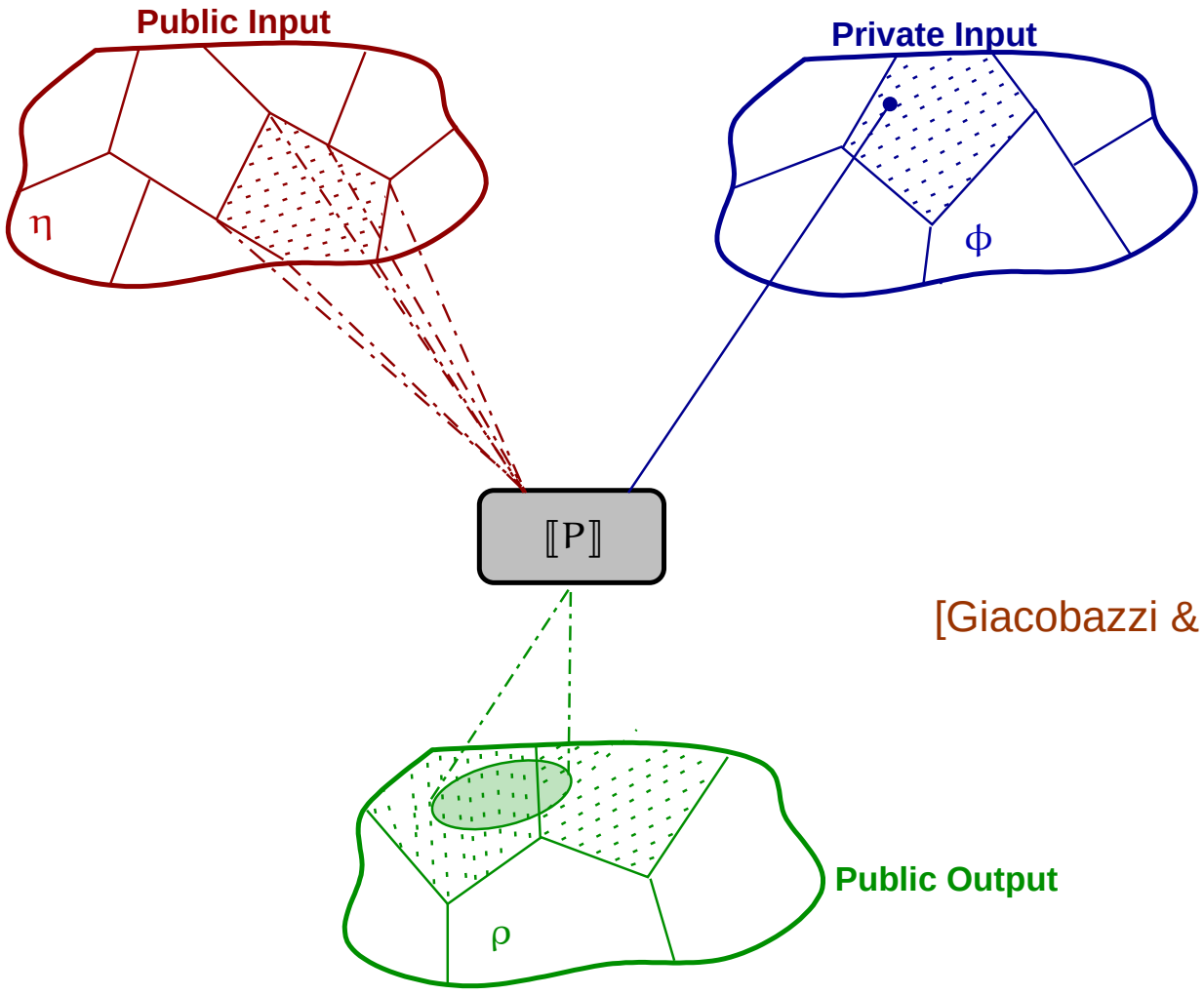
Declassified ANI (via allowing)



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})), \phi \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{H}})) \colon (\eta)\mathrm{P}(\phi \Rightarrow \rho) \colon$$
$$\eta(l_1) = \eta(l_2) \text{ and } \phi(h_1) = \phi(h_2) \Rightarrow \rho(\llbracket \mathrm{P} \rrbracket(h_1, \eta(l_1))^{\mathsf{L}}) = \rho(\llbracket \mathrm{P} \rrbracket(h_2, \eta(l_2))^{\mathsf{L}})$$

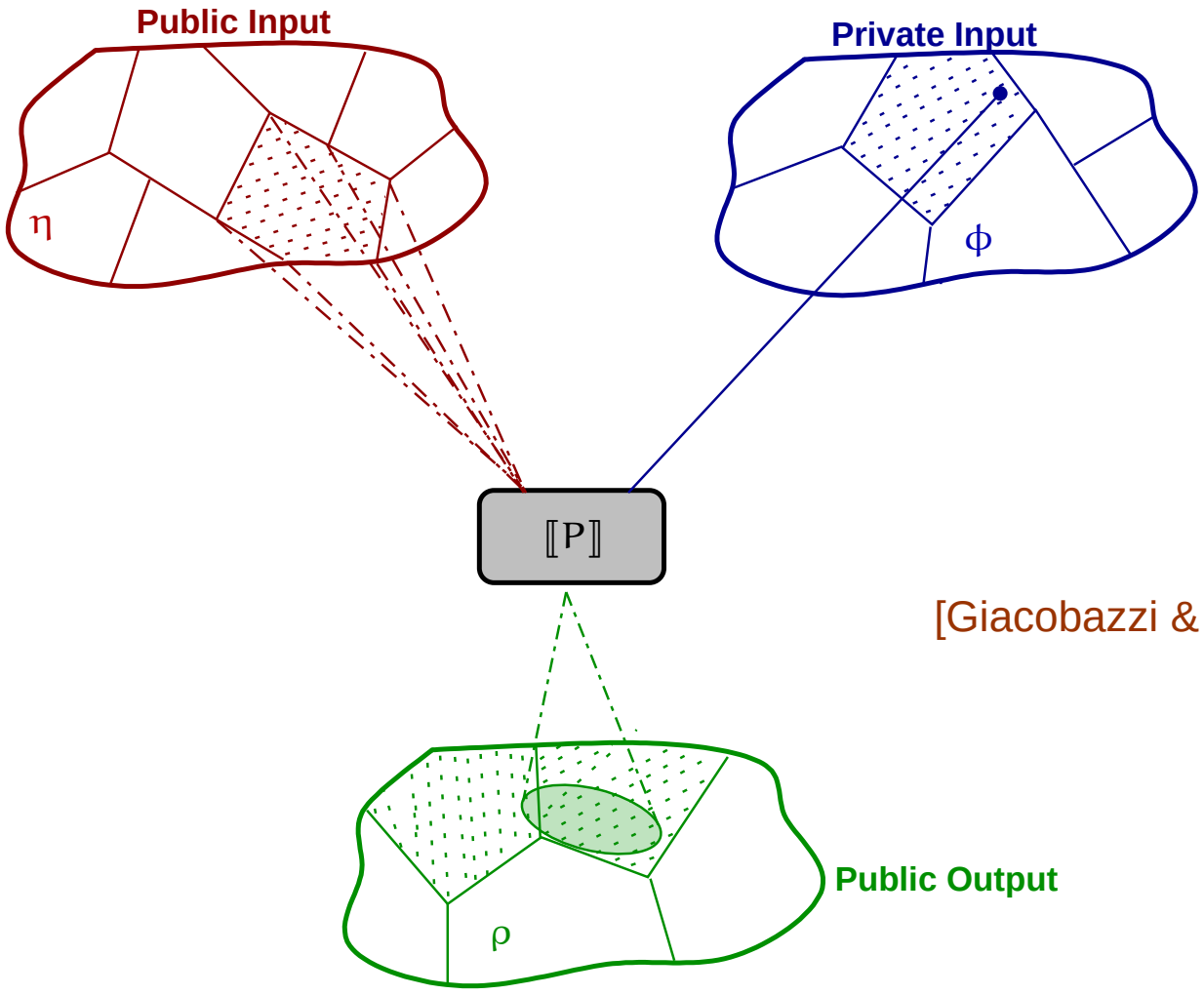
Declassified ANI (via allowing)



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})), \phi \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{H}})) \colon (\eta)\mathrm{P}(\phi \Rightarrow \rho) \colon$$
$$\eta(l_1) = \eta(l_2) \text{ and } \phi(h_1) = \phi(h_2) \Rightarrow \rho(\llbracket \mathrm{P} \rrbracket(h_1, \eta(l_1))^{\mathsf{L}}) = \rho(\llbracket \mathrm{P} \rrbracket(h_2, \eta(l_2))^{\mathsf{L}})$$

Declassified ANI (via allowing)



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})), \phi \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{H}})) \colon (\eta)\mathrm{P}(\phi \Rightarrow \rho) \colon$$
$$\eta(l_1) = \eta(l_2) \text{ and } \phi(h_1) = \phi(h_2) \Rightarrow \rho(\llbracket \mathrm{P} \rrbracket(h_1, \eta(l_1))^{\mathsf{L}}) = \rho(\llbracket \mathrm{P} \rrbracket(h_2, \eta(l_2))^{\mathsf{L}})$$

ANI vs Enforcing Robust Declassification

[Myers et al. ’04]

Language=IMP+*declassify*(e)+$[\bullet]$

$\mathrm{P}[\mathrm{a}]$ is the program P under the attack a!

ANI vs Enforcing Robust Declassification

[Myers et al. '04]

Language=IMP+*declassify*(e)+$[\bullet]$

$P[a]$ is the program P under the attack a!

$P[\bullet]$ is ROBUST if

$$\forall s_1, s_2 \in \Sigma. \forall a, a' : \ [\![P[a]]\!](s_1)^L = [\![P[a]]\!](s_2)^L \ \Rightarrow \ [\![P[a']]\!](s_1)^L = [\![P[a']]\!](s_2)^L$$

ANI vs Enforcing Robust Declassification

[Myers et al. '04]

Language=IMP+*declassify*(e)+$[\bullet]$

$P[a]$ is the program P under the attack a!

$P[\bullet]$ is ROBUST if

$$\forall s_1, s_2 \in \Sigma. \forall a, a' : [\![P[a]]\!](s_1)^L = [\![P[a]]\!](s_2)^L \Rightarrow [\![P[a']]\!](s_1)^L = [\![P[a']]\!](s_2)^L$$

If a *can control only the public inputs* then this is ANI!

ANI vs Enforcing Robust Declassification

[Myers et al. '04]

Language=IMP+*declassify*(e)+$[\bullet]$

$P[a]$ is the program P under the attack a!

$P[\bullet]$ is ROBUST if

$$\forall s_1, s_2 \in \Sigma. \forall a, a' : \ [\![P[a]]\!](s_1)^L = [\![P[a]]\!](s_2)^L \ \Rightarrow \ [\![P[a']]\!](s_1)^L = [\![P[a']]\!](s_2)^L$$

If a *can control only the public inputs* then this is ANI!

EXAMPLE:

$$P \stackrel{\text{def}}{=} [\bullet]; l := l + \textit{declassify}(h \ \textit{mod}\ 3)$$

P satisfies *robust declassification*!

ANI vs Enforcing Robust Declassification

[Myers et al. '04]

Language=IMP+*declassify*(e)+$[\bullet]$

$P[a]$ is the program P under the attack a!

$P[\bullet]$ is ROBUST if

$$\forall s_1, s_2 \in \Sigma. \forall a, a' : \ [\![P[a]]\!](s_1)^{\mathsf{L}} = [\![P[a]]\!](s_2)^{\mathsf{L}} \ \Rightarrow \ [\![P[a']]\!](s_1)^{\mathsf{L}} = [\![P[a']]\!](s_2)^{\mathsf{L}}$$

If a *can control only the public inputs* then this is ANI!

EXAMPLE:

$$P \stackrel{\text{def}}{=} [\bullet]; l := l + \textit{declassify}(h \ \textit{mod}\ 3)$$

P satisfies *robust declassification*!

Declassified ANI proves that P satisfies *robust declassification* by declassifying the LEAST amount of information necessary!

ANI vs Enforcing Robust Declassification

[Myers et al. '04]

Language=IMP+*declassify*(e)+$[\bullet]$

$P[a]$ is the program P under the attack a!

$P[\bullet]$ is ROBUST if

$$\forall s_1, s_2 \in \Sigma. \forall a, a' : \ [\![P[a]]\!](s_1)^L = [\![P[a]]\!](s_2)^L \ \Rightarrow \ [\![P[a']]\!](s_1)^L = [\![P[a']]\!](s_2)^L$$

If a *can control only the public inputs* then this is ANI!

EXAMPLE: Consider

$$P = \ l := l + (h \ \mathit{mod}\ 3)$$

The MAXIMAL amount of information disclosed is

$\phi = \{\top, 3\mathbb{Z}, 3\mathbb{Z}+1, 3\mathbb{Z}+2, \varnothing\}$.

ANI vs Delimited release

[Sabelfeld & Myers '04]

Language=IMP+*declassify*(e)

$s_1 \approx_E s_2$ iff $\forall e \in E.\ [\![e]\!](s_1) = [\![e]\!](s_2)$

ANI vs Delimited release

[Sabelfeld & Myers '04]

Language=IMP+*declassify*(e)

$s_1 \approx_E s_2$ iff $\forall e \in E.\ [\![e]\!](s_1) = [\![e]\!](s_2)$

P does satisfy DELIMITED RELEASE, $E = \left\{ e \,\middle|\, \textit{declassify}(e) \text{ in P} \right\}$

$$\forall s_1, s_2 \in \Sigma.\ s_1^L = s_2^L \wedge s_1 \approx_E s_2 \ \Rightarrow\ [\![P]\!](s_1)^L = [\![P]\!](s_2)^L$$

ANI vs Delimited release

[Sabelfeld & Myers '04]

Language=IMP+*declassify*(e)

$s_1 \approx_E s_2$ iff $\forall e \in E.\ [\![e]\!](s_1) = [\![e]\!](s_2)$

P does satisfy DELIMITED RELEASE, $E = \left\{ e \;\middle|\; \textit{declassify}(e) \text{ in } P \right\}$

$$\forall s_1, s_2 \in \Sigma.\ s_1^L = s_2^L \wedge s_1 \approx_E s_2 \Rightarrow [\![P]\!](s_1)^L = [\![P]\!](s_2)^L$$

EXAMPLE:

The program P satisfies delimited release and P′ does not:

$$P \stackrel{\text{def}}{=} \textbf{if } \textit{declassify}(h \geq k) \textbf{ then } (h := h - k; l := l + k) \textbf{ else nil}$$

$$P' \stackrel{\text{def}}{=} \left[\begin{array}{ll} l := 0; & \\ \textbf{while } n \geq 0 \textbf{ do} & k := 2^{n+1} \\ & \textbf{if } \textit{declassify}(h \geq k) \textbf{ then } (h := h - k; l := l + k) \textbf{ else nil} \\ & n := n - 1 \end{array}\right.$$

ANI vs Delimited release

[Sabelfeld & Myers '04]

Language=IMP+*declassify*(e)

$s_1 \approx_E s_2$ iff $\forall e \in E.\ [\![e]\!](s_1) = [\![e]\!](s_2)$

P does satisfy DELIMITED RELEASE, $E = \left\{ e \;\middle|\; \textit{declassify}(e) \text{ in } P \right\}$

$$\boxed{\forall s_1, s_2 \in \Sigma.\ s_1^L = s_2^L \wedge s_1 \approx_E s_2 \ \Rightarrow\ [\![P]\!](s_1)^L = [\![P]\!](s_2)^L}$$

EXAMPLE:

The program P satisfies delimited release and P′ does not:

$$P \stackrel{\text{def}}{=} \textbf{if } \textit{declassify}(h \geq k) \textbf{ then } (h := h - k; l := l + k) \textbf{ else nil}$$

$$\phi_k = \{\mathbb{V}^H, \left\{ h \;\middle|\; h \geq k \right\}, \left\{ h \;\middle|\; h < k \right\}, \varnothing\}$$

$$\phi = \bigsqcap_k \phi_k = \textit{id}$$

ANI vs Delimited release

[Sabelfeld & Myers '04]

Language=IMP+*declassify*(e)

$s_1 \approx_E s_2$ iff $\forall e \in E.\ [\![e]\!](s_1) = [\![e]\!](s_2)$

P does satisfy DELIMITED RELEASE, $E = \left\{ e \,\middle|\, \textit{declassify}(e) \text{ in } P \right\}$

$$\forall s_1, s_2 \in \Sigma.\ s_1^L = s_2^L \wedge s_1 \approx_E s_2 \ \Rightarrow\ [\![P]\!](s_1)^L = [\![P]\!](s_2)^L$$

EXAMPLE:

The program P satisfies delimited release and P' does not:

$$P' \stackrel{\text{def}}{=} \left[\begin{array}{ll} l := 0; & \\ \textbf{while } n \geq 0 \textbf{ do} & k := 2^{n+1} \\ & \textbf{if } \textit{declassify}(h \geq k) \textbf{ then } (h := h - k; l := l + k) \textbf{ else nil} \\ & n := n - 1 \end{array}\right.$$

$\phi = id$

ANI vs Relaxed non-interference

[Li & Zdancewic '05]

Language=λ-calculus (no explicit declassification)

$\Downarrow$

P does satisfy RELAXED NONINTERFERENCE, if

$$\boxed{P \equiv f(n_1\,\sigma_1)(n_2\,\sigma_2)\ldots(n_k\,\sigma_k)}$$

where

f is a λ-term without any secret variable

σ_i are all the private variables

n_i are λ-terms denoting downgrading policies such that $(n_i\,\sigma_i)$ are all public.

ANI vs Relaxed non-interference

[Li & Zdancewic '05]

Language=λ-calculus (no explicit declassification)

$\Downarrow$

P does satisfy RELAXED NONINTERFERENCE, if

$$\boxed{P \equiv f(n_1\,\sigma_1)(n_2\,\sigma_2)\ldots(n_k\,\sigma_k)}$$

EXAMPLE:

$$P_1 \stackrel{\text{def}}{=} \left[\begin{array}{l} x := \mathit{hash}(secret); \\ y := x \ \mathit{mod}\ 2^{64}; \\ \textbf{if}\ y = input\ \textbf{then}\ output = 1\ \textbf{else}\ output := 0; \end{array}\right.$$

The corresponding λ-program does satisfy *relaxed noninterference* with the downgrading policy:

$$\lambda secret.\,\lambda input.\,(\mathit{hash}(secret)\ \mathit{mod}\ 2^{64}) = input$$

ANI vs Relaxed non-interference

[Li & Zdancewic '05]

Language=λ-calculus (no explicit declassification)

$\Downarrow$

P does satisfy RELAXED NONINTERFERENCE, if

$$\boxed{P \equiv f(n_1\sigma_1)(n_2\sigma_2)\ldots(n_k\sigma_k)}$$

EXAMPLE:

$$P \stackrel{\text{def}}{=} \left[\begin{array}{l} x := \mathit{hash}(sec); y := x \ \mathit{mod}\ 2^{64}; \\ \textbf{if } y = in \textbf{ then } out = 1 \textbf{ else } out := 0; \end{array}\right.$$

$\Downarrow$

$$\phi_{in} = \left\{ \begin{array}{l} \mathbb{V}^H, \left\{ sec \;\middle|\; \mathit{hash}(sec) \ \mathit{mod}\ 2^{64} = in \right\}, \\ \left\{ sec \;\middle|\; \mathit{hash}(sec) \ \mathit{mod}\ 2^{64} \neq in \right\}, \varnothing \end{array} \right\}$$

$$\phi = \bigsqcap_{in} \phi_{in} = \{\mathbb{V}^H, \varnothing\} \cup \left\{ 2^{64}\mathbb{Z} + n \;\middle|\; 0 \leq n < 2^{64} \right\}$$

ANI vs Relaxed non-interference

[Li & Zdancewic '05]

Language=λ-calculus (no explicit declassification)

$\Downarrow$

P does satisfy RELAXED NONINTERFERENCE, if

$$\boxed{P \equiv f(n_1\sigma_1)(n_2\sigma_2)\ldots(n_k\sigma_k)}$$

EXAMPLE:

Password checking:

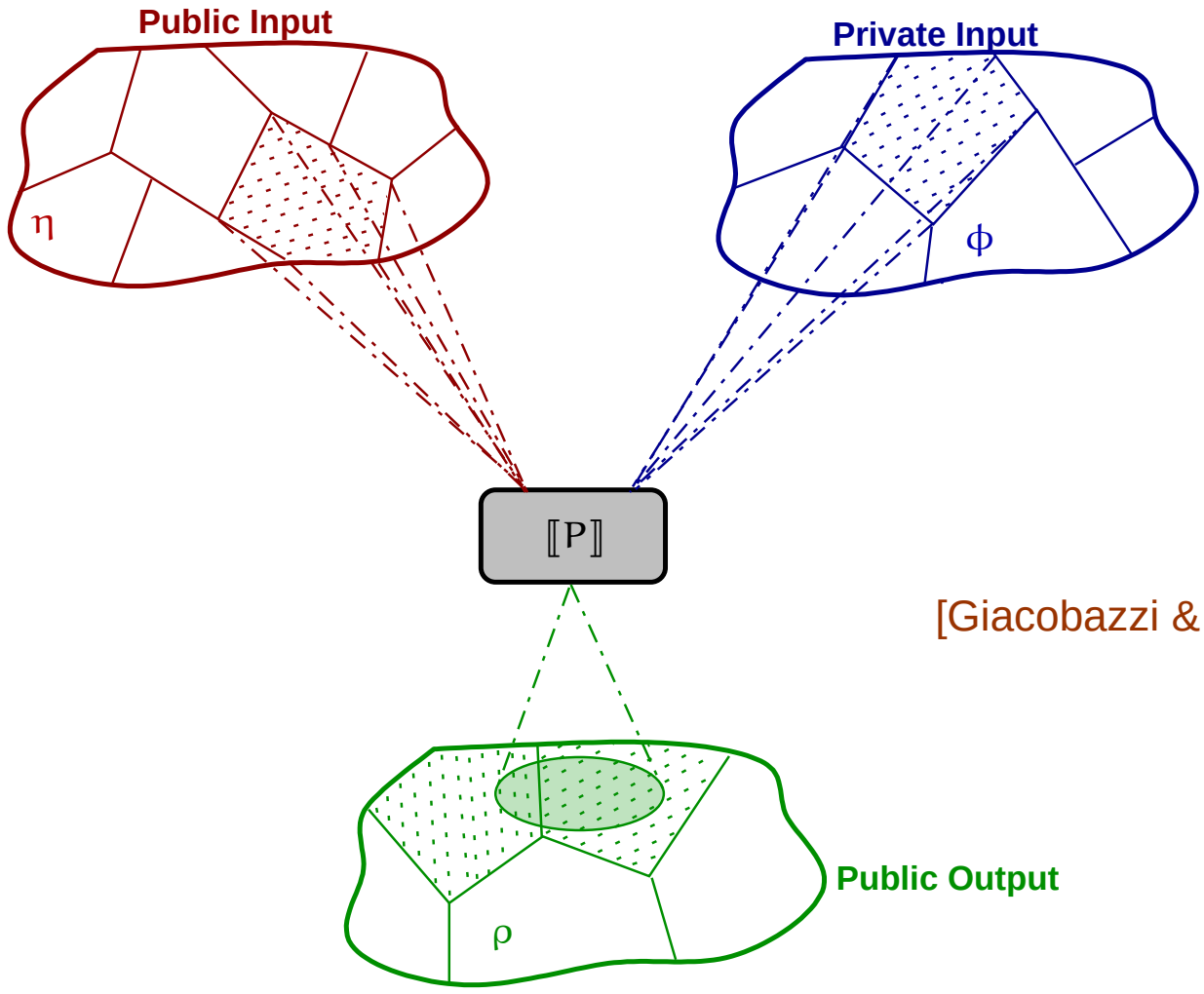
$$P = \lambda in.\textbf{if } in = \sigma_{pw} \textbf{ then } out := 1 \textbf{ else } out := 0$$

it does satisfy *relaxed non-interference* since we can transform in:

$$\lambda x.\lambda g : \mathbb{Z} \longrightarrow \mathbb{Z}.(\textbf{if } g(x) \textbf{ then } out := 1 \textbf{ else } out := 0)in((\lambda x.\lambda y.x = y)\sigma_{pw})$$

$\Rightarrow$ For *Declassified ANI* the program is not secret: $\phi = id$.

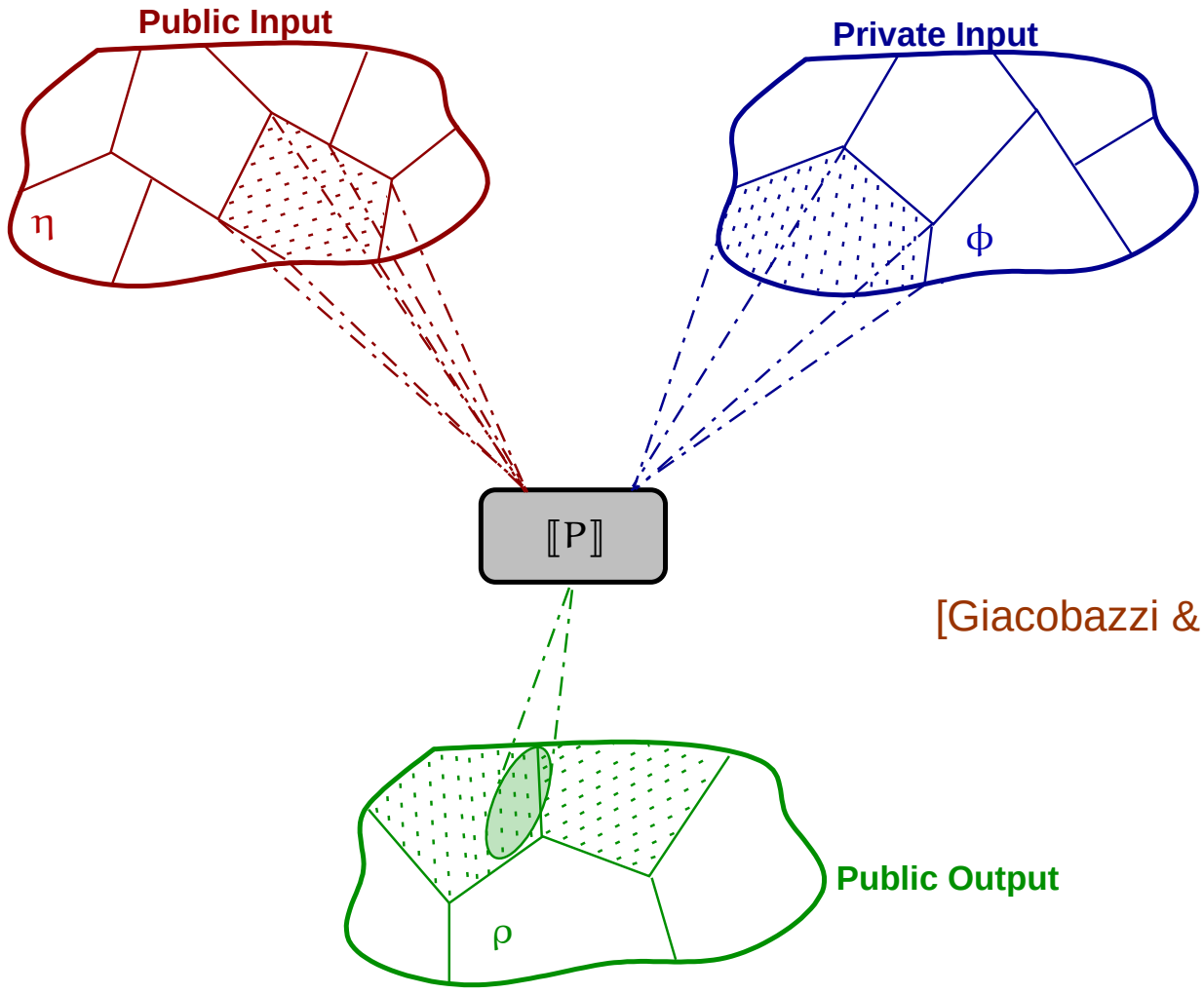
A new viewpoint: Declassified ANI via blocking



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})), \phi \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{H}})) \colon (\eta)\mathrm{P}(\phi \rightsquigarrow \rho) \colon$$
$$\eta(l_1) = \eta(l_2) \Rightarrow \rho(\llbracket \mathrm{P} \rrbracket(\phi(h_1), \eta(l_1))^{\mathsf{L}}) = \rho(\llbracket \mathrm{P} \rrbracket(\phi(h_2), \eta(l_2))^{\mathsf{L}})$$

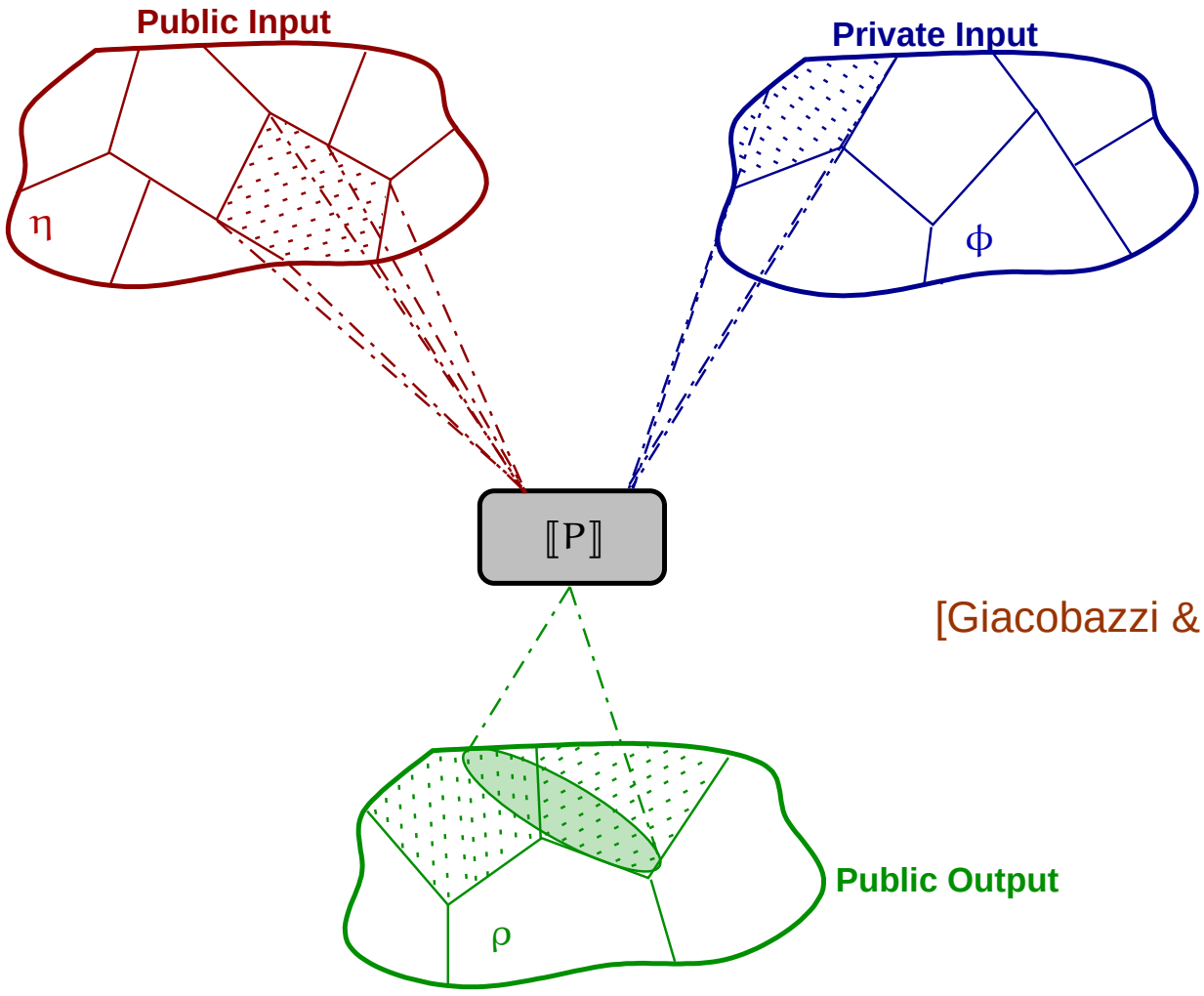
A new viewpoint: Declassified ANI via blocking



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})), \phi \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{H}})) \colon (\eta)\mathrm{P}(\phi \rightsquigarrow \rho) \colon$$
$$\eta(l_1) = \eta(l_2) \Rightarrow \rho(\llbracket \mathrm{P} \rrbracket(\phi(h_1), \eta(l_1))^{\mathsf{L}}) = \rho(\llbracket \mathrm{P} \rrbracket(\phi(h_2), \eta(l_2))^{\mathsf{L}})$$

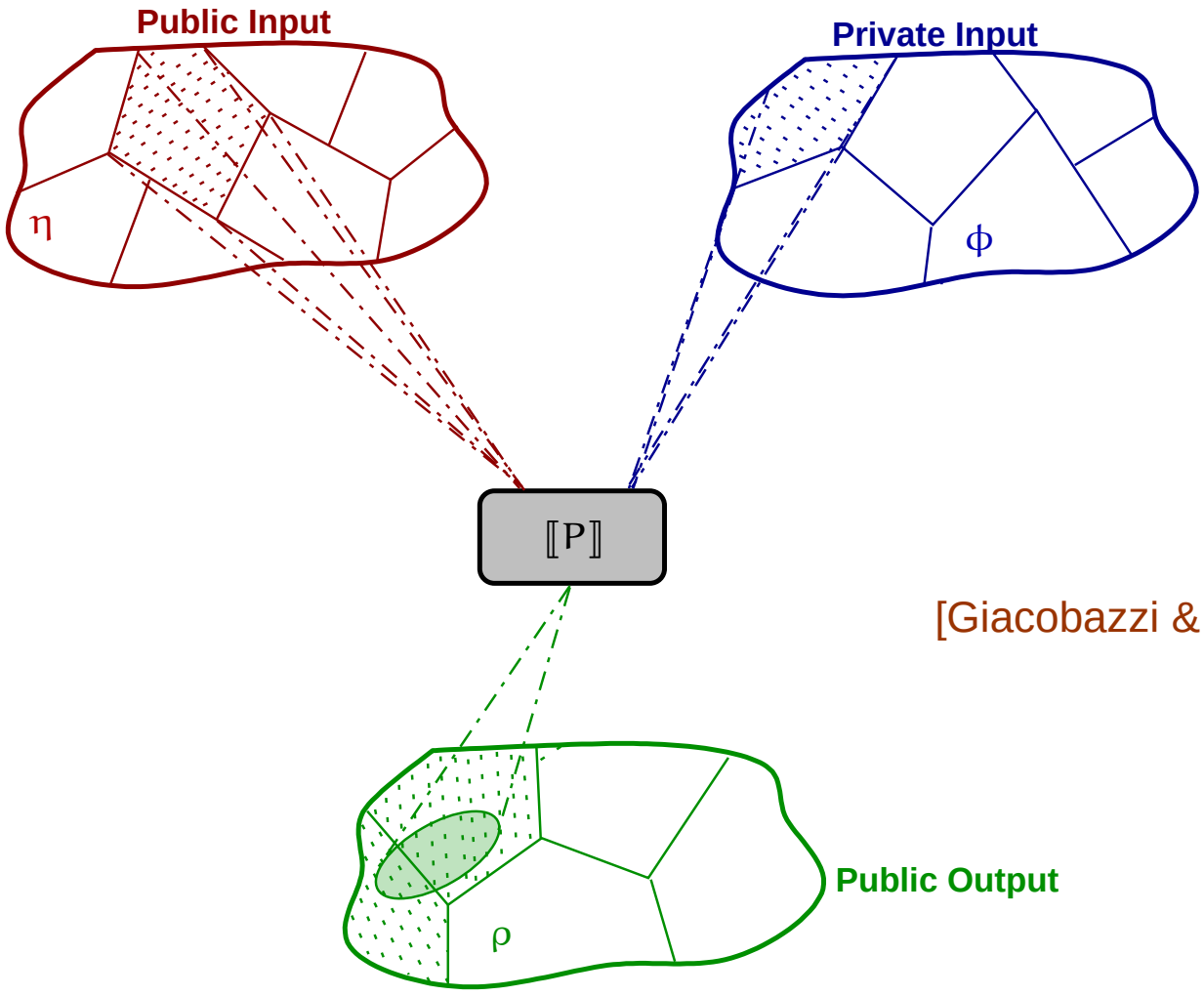
A new viewpoint: Declassified ANI via blocking



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})), \phi \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{H}})) \colon (\eta)\mathrm{P}(\phi \rightsquigarrow \rho) \colon$$
$$\eta(l_1) = \eta(l_2) \Rightarrow \rho(\llbracket \mathrm{P} \rrbracket(\phi(h_1), \eta(l_1))^{\mathsf{L}}) = \rho(\llbracket \mathrm{P} \rrbracket(\phi(h_2), \eta(l_2))^{\mathsf{L}})$$

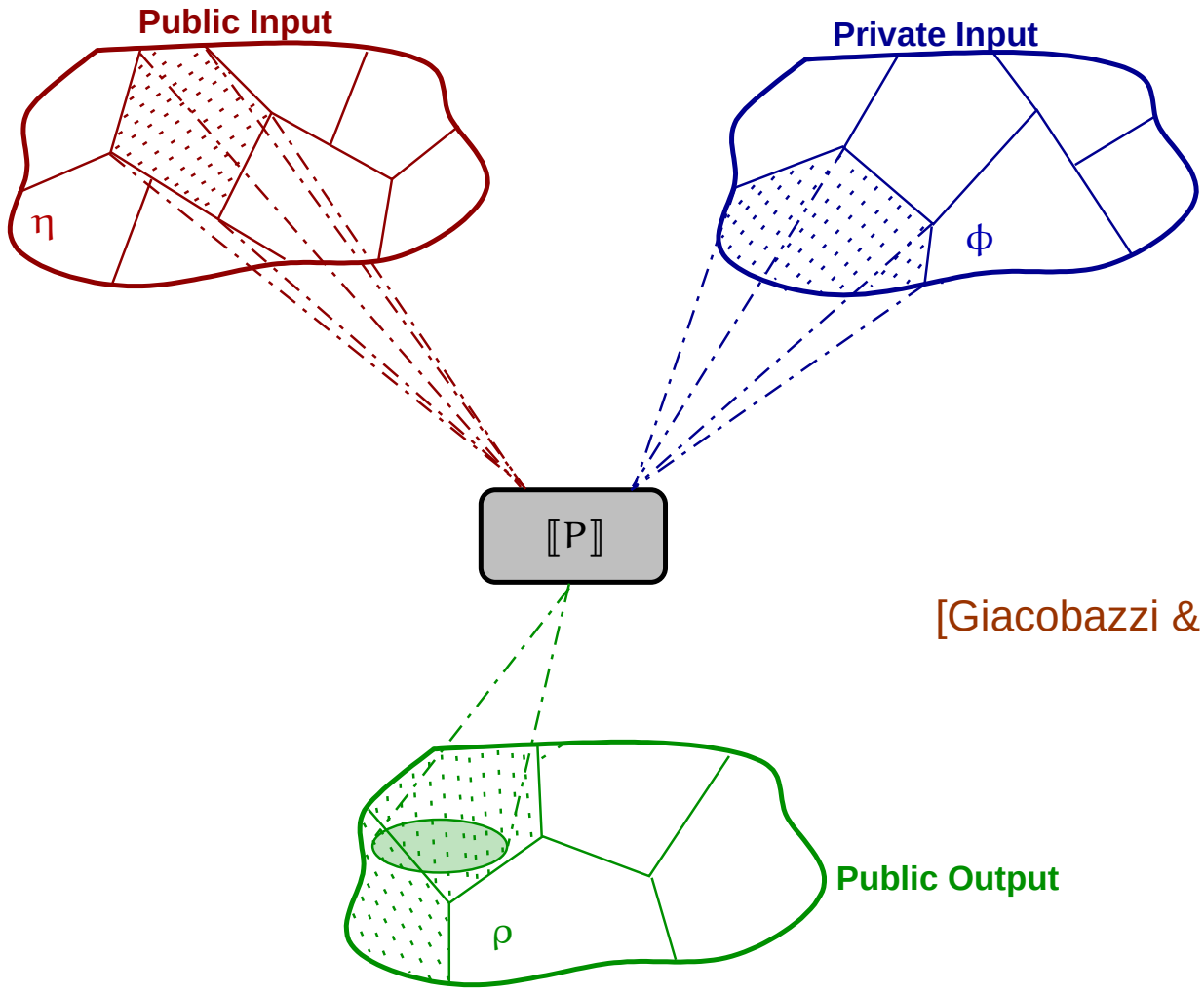
A new viewpoint: Declassified ANI via blocking



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})), \phi \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{H}})) \colon (\eta)\mathrm{P}(\phi \rightsquigarrow \rho) \colon$$
$$\eta(l_1) = \eta(l_2) \Rightarrow \rho(\llbracket \mathrm{P} \rrbracket(\phi(h_1), \eta(l_1))^{\mathsf{L}}) = \rho(\llbracket \mathrm{P} \rrbracket(\phi(h_2), \eta(l_2))^{\mathsf{L}})$$

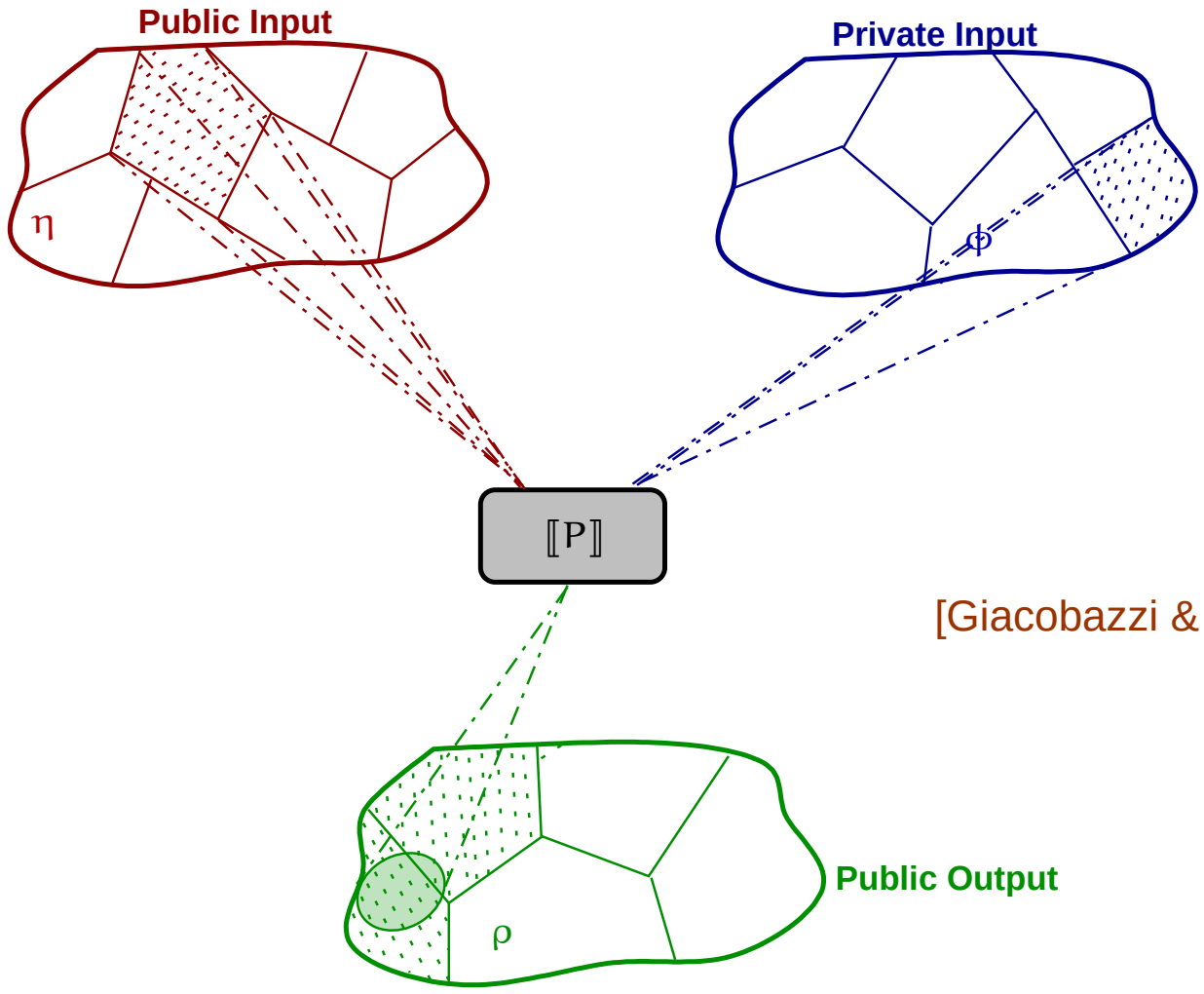
A new viewpoint: Declassified ANI via blocking



[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})), \phi \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{H}})) \colon (\eta)\mathrm{P}(\phi \rightsquigarrow \rho) \colon$$
$$\eta(l_1) = \eta(l_2) \Rightarrow \rho(\llbracket \mathrm{P} \rrbracket(\phi(h_1), \eta(l_1))^{\mathsf{L}}) = \rho(\llbracket \mathrm{P} \rrbracket(\phi(h_2), \eta(l_2))^{\mathsf{L}})$$

A new viewpoint: Declassified ANI via blocking



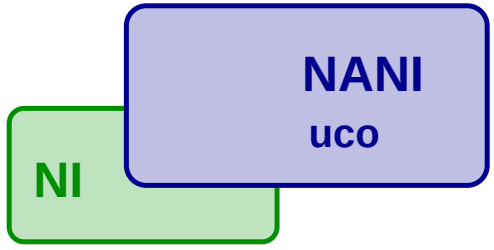
[Giacobazzi & Mastroeni '04]

$$\rho, \eta \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{L}})), \phi \in \mathit{Abs}(\wp(\mathbb{V}^{\mathsf{H}})) \colon (\eta)\mathrm{P}(\phi \rightsquigarrow \rho) \colon$$
$$\eta(l_1) = \eta(l_2) \Rightarrow \rho(\llbracket \mathrm{P} \rrbracket(\phi(h_1), \eta(l_1))^{\mathsf{L}}) = \rho(\llbracket \mathrm{P} \rrbracket(\phi(h_2), \eta(l_2))^{\mathsf{L}})$$

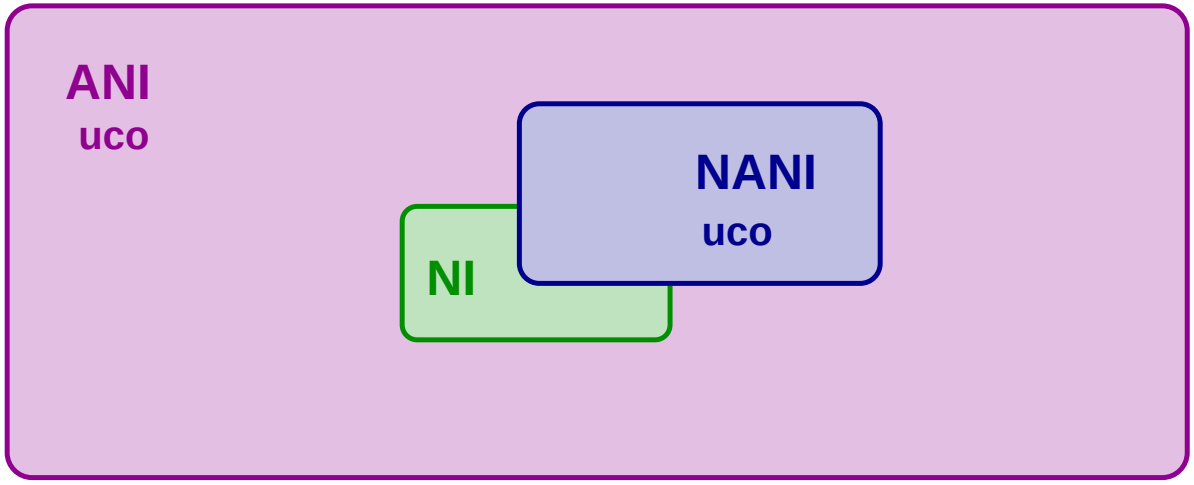
Conclusions

NI

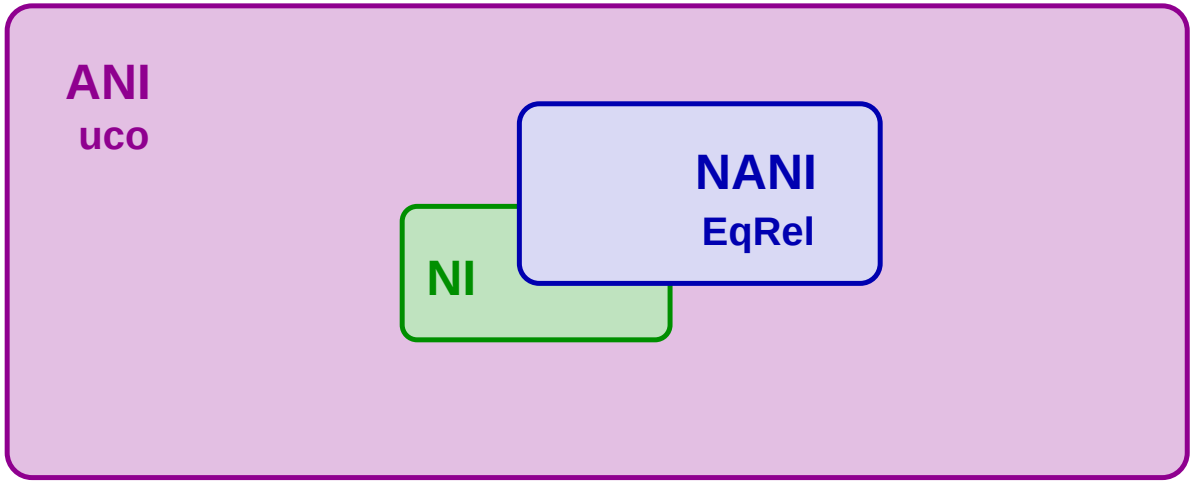
Conclusions



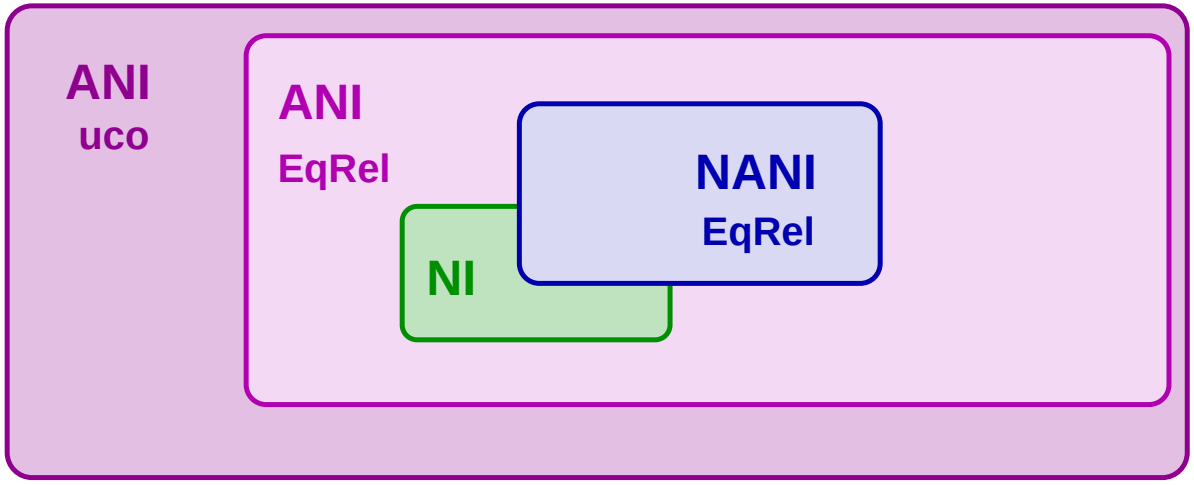
Conclusions



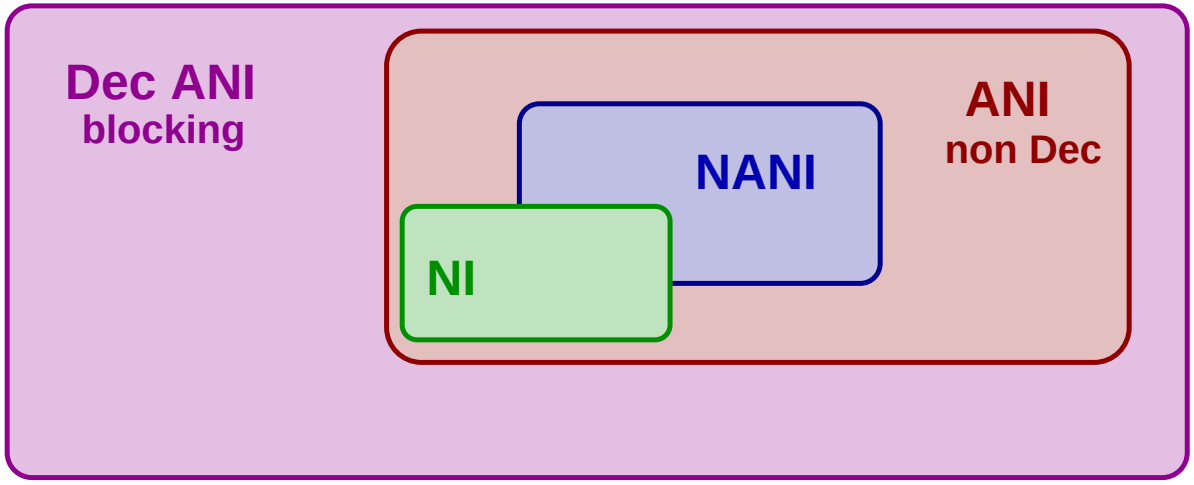
Conclusions



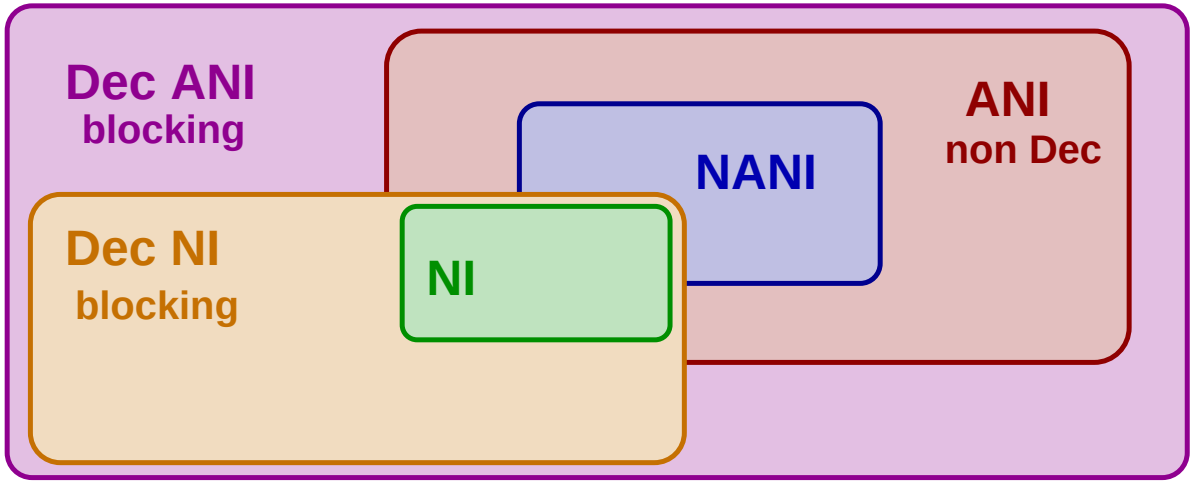
Conclusions



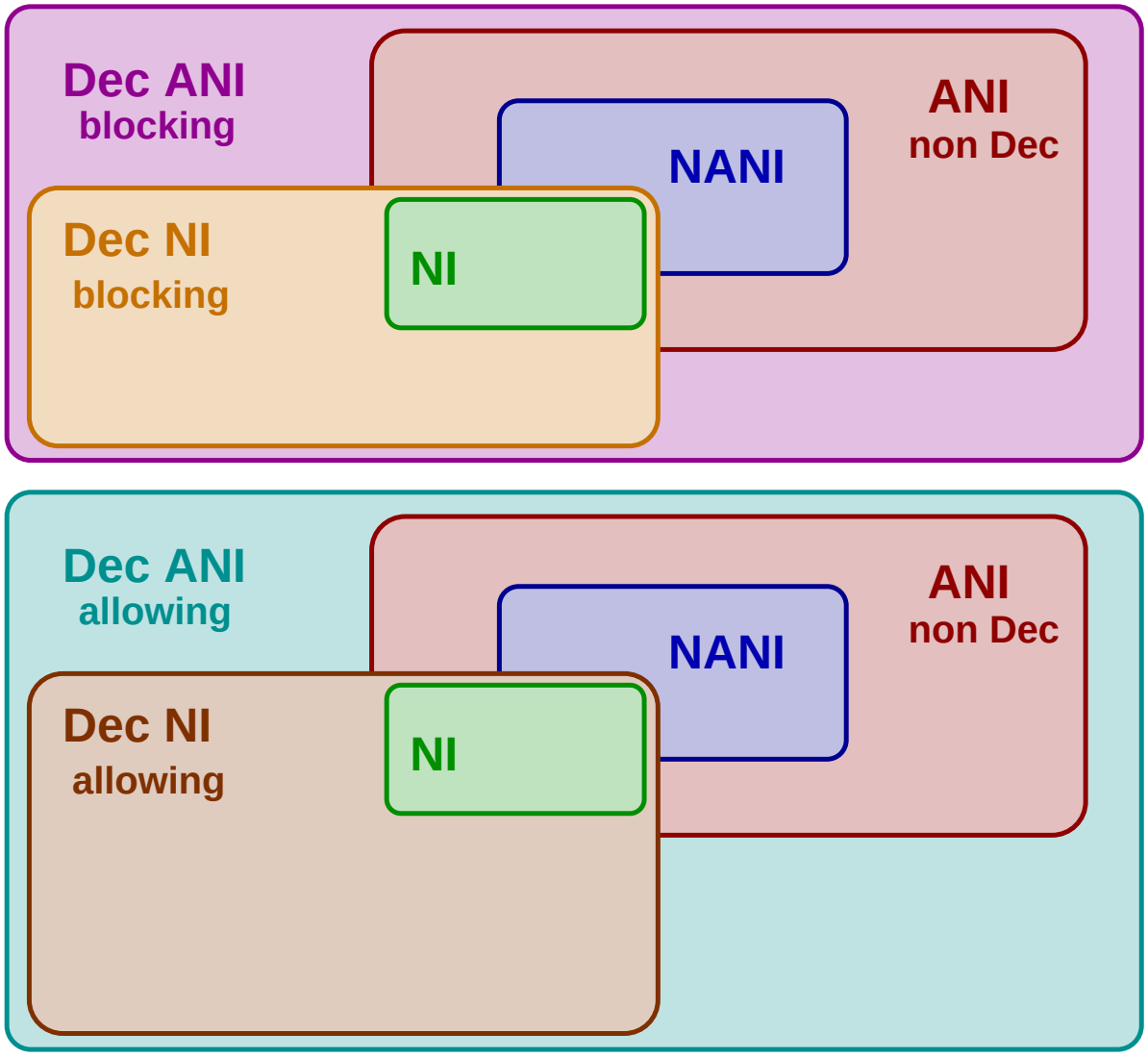
Conclusions



Conclusions



Conclusions



Conclusions