A dependently typed ambient calculus

C. Lhoussaine V. Sassone

University of Sussex, UK

Global Computing 2004 (9.03.04)

The Objective

In general:

Resource access control in global computing systems

Specifically:

Boundary control in mobile agent systems.

The Objective

In general:

Resource access control in global computing systems

Specifically:

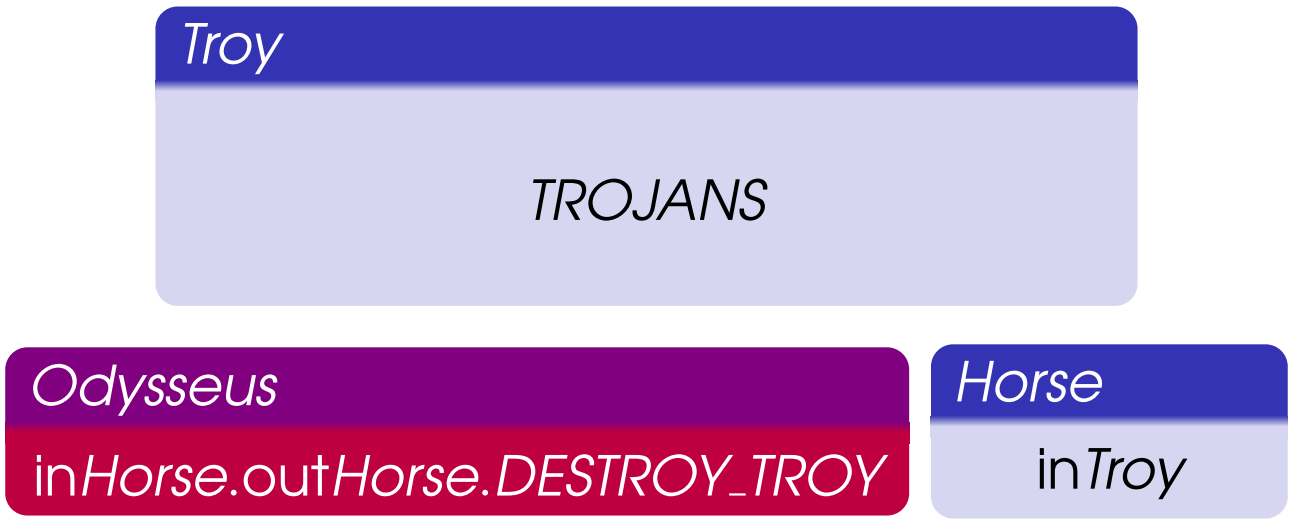
Boundary control in mobile agent systems.

Ambient calculus

- complete, yet small set of primitives for mobility
- hierarchical structure
- local communication

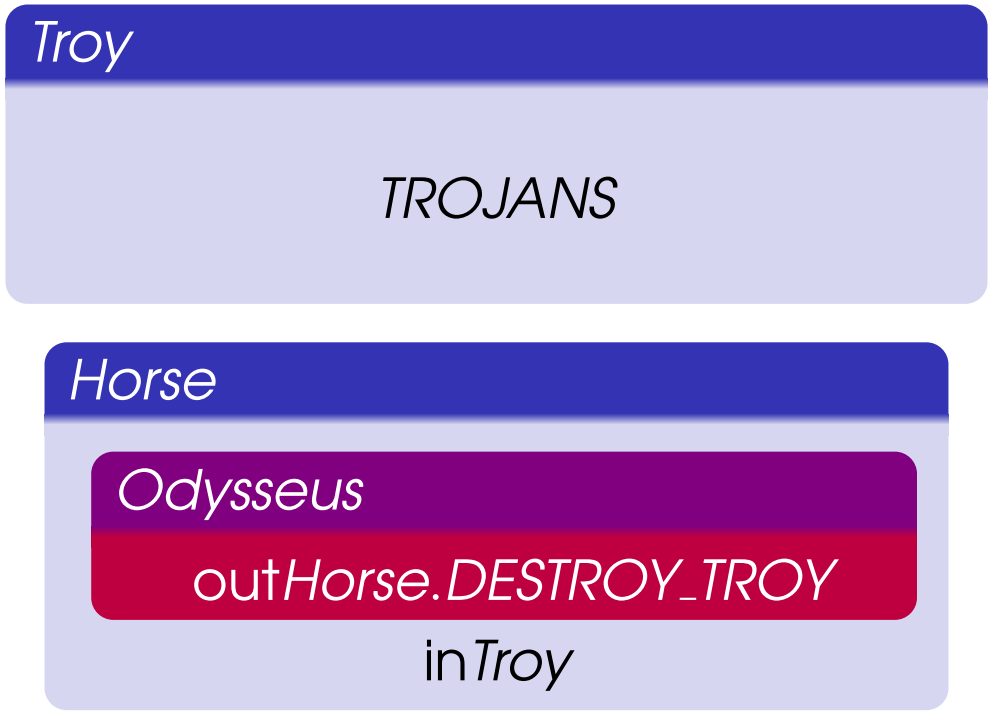
Troy Horse

Required Policy: No `Achaean` inside the `City` walls.



Troy Horse

Required Policy: No `Achaean` inside the `City` walls.

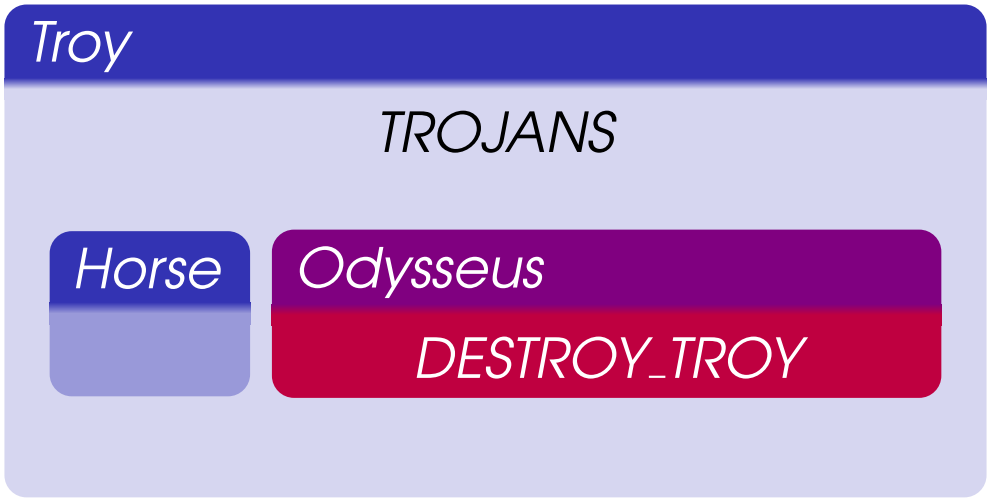


Troy Horse

Required Policy: No `Achaean` inside the `City` walls.

Troy Horse

Required Policy: No `Achaean` inside the `City` walls.

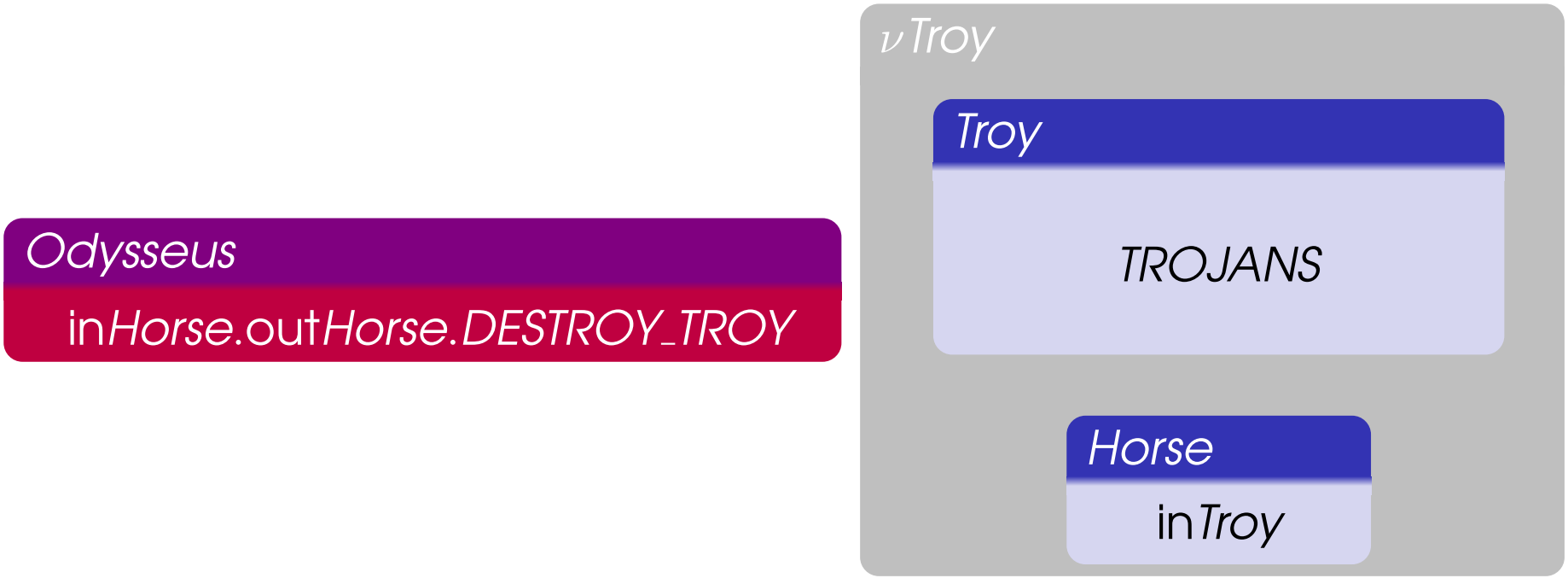


Troy Horse

Required Policy: No `Achaean` inside the `City` walls.

Observe:

Scope restriction does not prevent *Odysseus* ending up in *Troy*



Type based approach

- The past: Groups. Types specify allowed parents' groups.

$$\mathit{Odysseus} : \texttt{Achean}[\mathsf{mob}\{\texttt{Ground}, \texttt{Toy}, \texttt{City}\}]$$
$$\mathit{Horse} : \texttt{Toy}[\mathsf{mob}\{\texttt{Ground}, \texttt{City}\}]$$
$$\mathit{Troy} : \texttt{City}[\_]$$

Type based approach

- The past: Groups. Types specify allowed parents' groups.

$$\textit{Odysseus} : \texttt{Achean}[\mathsf{mob}\{\texttt{Ground}, \texttt{Toy}, \texttt{City}\}]$$
$$\textit{Horse} : \texttt{Toy}[\mathsf{mob}\{\texttt{Ground}, \texttt{City}\}]$$
$$\textit{Troy} : \texttt{City}[\_]$$

Ill-typed unless *Odysseus* declares parent group `City`.

Type based approach

- The past: Groups. Types specify allowed parents' groups.

$$\begin{aligned} \mathit{Odysseus} &: \texttt{Achean}[\mathsf{mob}\{\texttt{Ground}, \texttt{Toy}, \texttt{City}\}] \\ \mathit{Horse} &: \texttt{Toy}[\mathsf{mob}\{\texttt{Ground}, \texttt{City}\}] \\ \mathit{Troy} &: \texttt{City}[\_] \end{aligned}$$

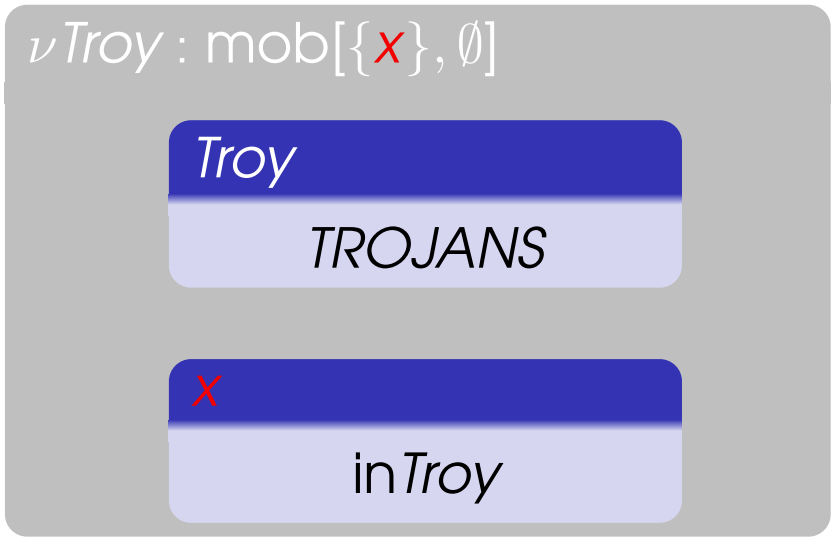
- The present: Dependent types. A simpler, more flexible and fine-grained approach.

$$\begin{aligned} \mathit{Odysseus} &: \mathsf{mob}[\emptyset, \{\mathit{Horse}, \mathit{Troy}\}] \\ \mathit{Horse} &: \mathsf{mob}[\{\mathit{Odysseus}\}, \{\mathit{Troy}\}] \\ \mathit{Troy} &: \mathsf{mob}[\{\mathit{Horse}, \mathit{Odysseus}\}, \emptyset] \end{aligned}$$

Dynamic Types

- Via communication can express dynamic, ad-hoc, personalised services and security policies.

$\langle Horse \rangle \mid (x).$



The Types

- Ambient types:

  a :mob$[\mathcal{P}, \mathcal{C}]$

  $\mathcal{P}$: set of possible *parents* for a

  $\mathcal{C}$: set of possible *children* for a

- Capability types:

  in$a$:cap$[\mathcal{P}]$

  $\mathcal{P}$: set of ambients where in$a$ might be exercised

Typing Contexts and Coherence

- Typing contexts:

$$\Gamma ::= a_1 : \mathsf{mob}[\mathcal{P}_1, \mathcal{C}_1], \ldots, a_n : \mathsf{mob}[\mathcal{P}_n, \mathcal{C}_n]$$

- Coherence:

$$a : \mathsf{mob}[\{b\}, \emptyset], b : \mathsf{mob}[\mathcal{P}, \mathcal{C}] \Rightarrow a \in \mathcal{C}$$
$$a : \mathsf{mob}[\emptyset, \{b\}], b : \mathsf{mob}[\mathcal{P}, \mathcal{C}] \Rightarrow a \in \mathcal{P}$$

Typing Contexts and Coherence

- Typing contexts:

$$\Gamma ::= a_1 : \mathrm{mob}[\mathcal{P}_1, \mathcal{C}_1], \ldots, a_n : \mathrm{mob}[\mathcal{P}_n, \mathcal{C}_n]$$

- Coherence:

$$a : \mathrm{mob}[\{b\}, \emptyset], b : \mathrm{mob}[\mathcal{P}, \mathcal{C}] \Rightarrow a \in \mathcal{C}$$
$$a : \mathrm{mob}[\emptyset, \{b\}], b : \mathrm{mob}[\mathcal{P}, \mathcal{C}] \Rightarrow a \in \mathcal{P}$$

- Need accurate context updating

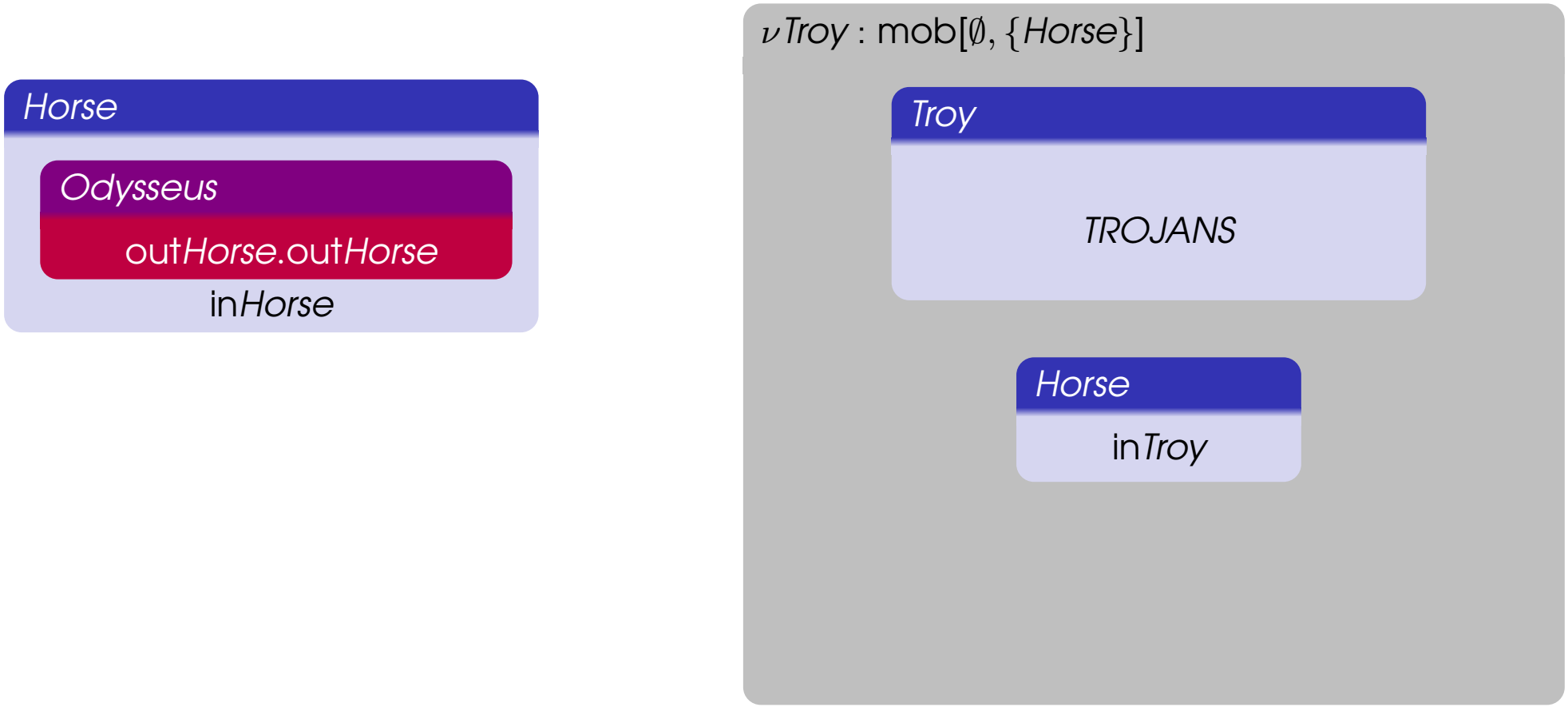
$\nu\, Troy : \mathrm{mob}[\emptyset, \{Horse\}]$

$Troy : \mathrm{mob}[\emptyset, \{Horse\}]$
$Horse : \mathrm{mob}[\mathcal{P} \cup \{Troy\}, \mathcal{C}]$

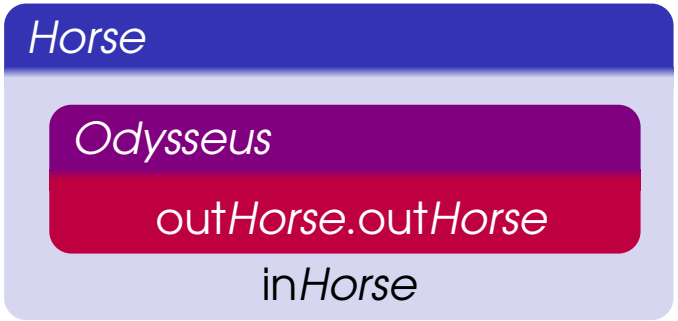
$Horse : \mathrm{mob}[\mathcal{P}, \mathcal{C}]$

$\Gamma^{(b:\mathrm{mob}[\mathcal{P},\mathcal{C}])}, b : \mathrm{mob}[\mathcal{P}, \mathcal{C}]$: the coherent updating of Γ wrt. the new assignment $b : \mathrm{mob}[\mathcal{P}, \mathcal{C}]$.

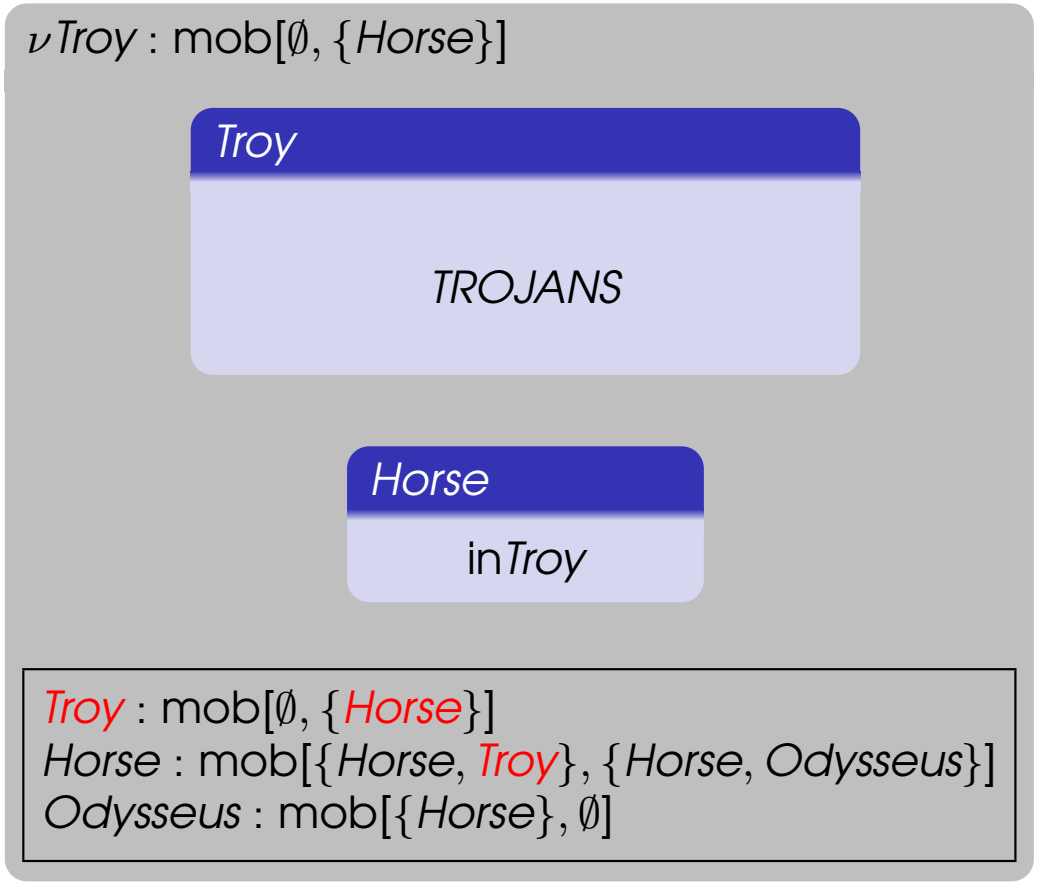
Matrioshka (nesting) Horses



Matrioshka (nesting) Horses

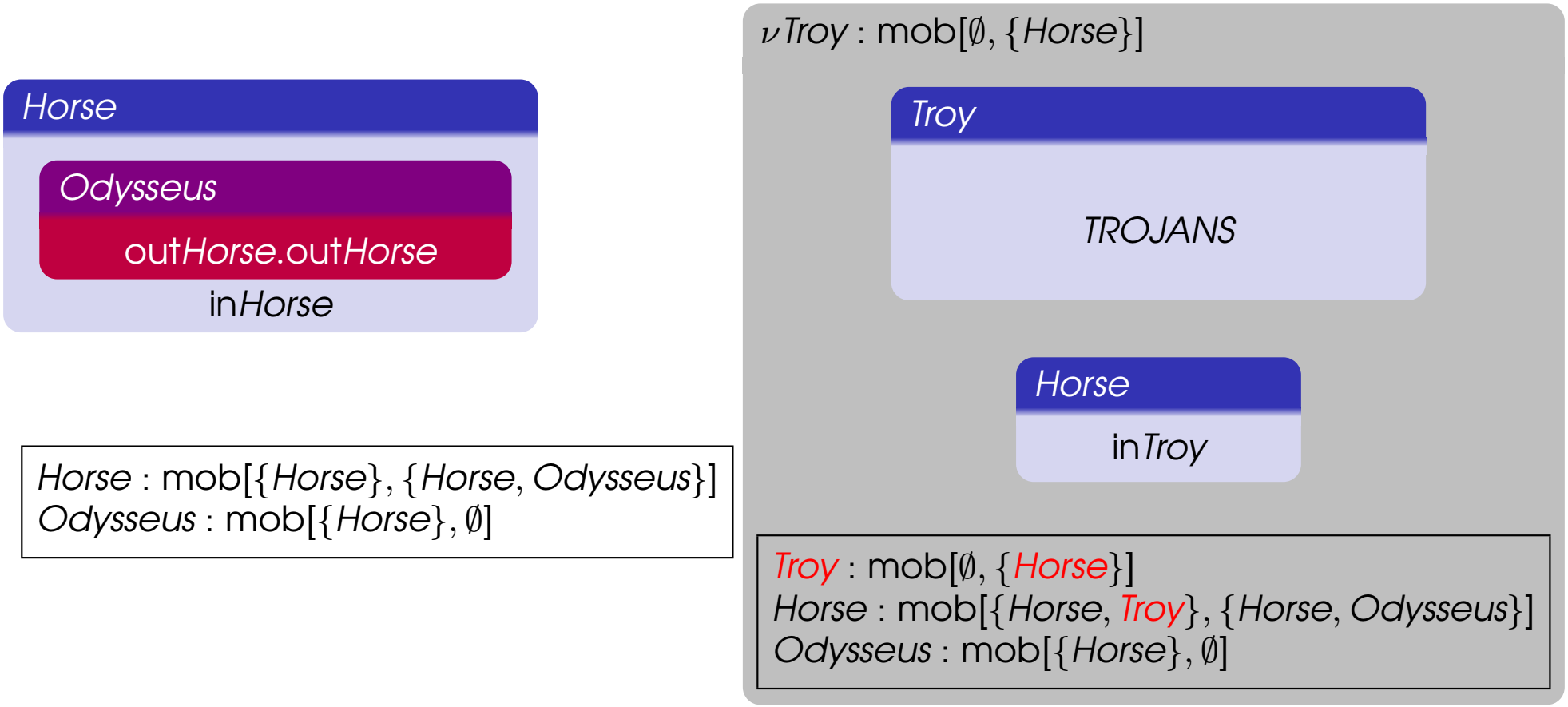


$Horse : \mathrm{mob}[\{Horse\}, \{Horse, Odysseus\}]$
$Odysseus : \mathrm{mob}[\{Horse\}, \emptyset]$



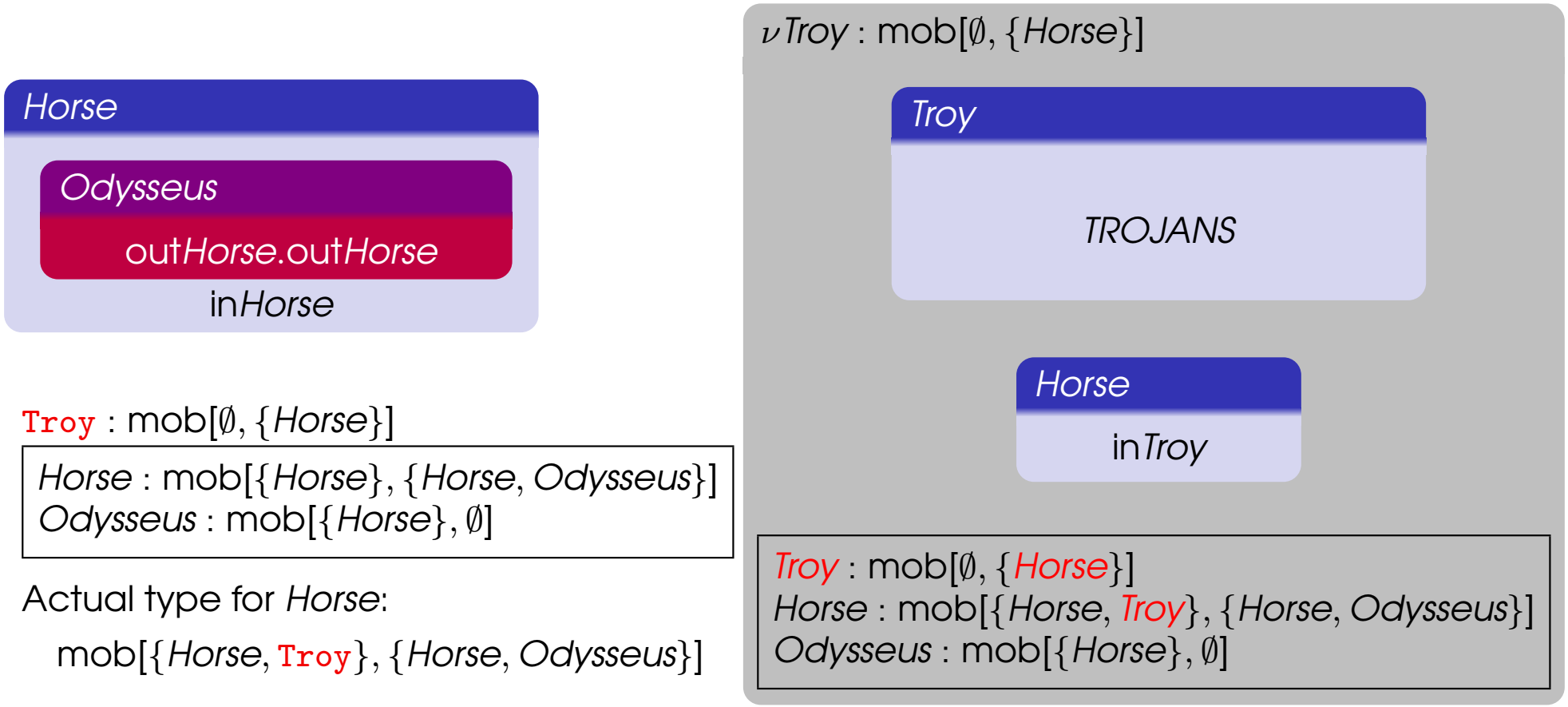
$Troy : \mathrm{mob}[\emptyset, \{Horse\}]$
$Horse : \mathrm{mob}[\{Horse, Troy\}, \{Horse, Odysseus\}]$
$Odysseus : \mathrm{mob}[\{Horse\}, \emptyset]$

Matrioshka (nesting) Horses



- Need to account for the potential new capabilities acquired during *Odysseus*'s execution: Abstract names

Matrioshka (nesting) Horses



Troy : mob[∅, {Horse}]

Horse : mob[{Horse}, {Horse, Odysseus}]
Odysseus : mob[{Horse}, ∅]

Actual type for *Horse*:

mob[{Horse, Troy}, {Horse, Odysseus}]

Troy : mob[∅, {Horse}]
Horse : mob[{Horse, Troy}, {Horse, Odysseus}]
Odysseus : mob[{Horse}, ∅]

- Outside "ν *Troy*," *Odysseus* is ill-typed.

Abstract Contexts

Abstract Context:

$$\Theta, \Xi ::= a_1 : \mathrm{mob}[\mathcal{P}_1, \mathcal{C}_1], \ldots, a_n : \mathrm{mob}[\mathcal{P}_n, \mathcal{C}_n]$$

Typing judgements:

$$\Gamma \vdash^{\Theta} a : \mathrm{mob}[\mathcal{P}, \mathcal{C}] \qquad \Gamma \vdash_a^{\Theta;\Xi} P$$

Γ : "concrete" typing context

Θ : local abstract context

Ξ : external abstract context

a : current location

Actual ambient type

$$\frac{\mathsf{mob}[\mathcal{P},\mathcal{C}] = (\Gamma;\Theta,\Xi)[b]}{\Gamma \vdash_a^{\Theta;\Xi} b : \mathsf{mob}[\mathcal{P},\mathcal{C}]}$$

$$\ldots, \mathit{Horse} : \mathsf{mob}\begin{bmatrix}\{\mathit{Horse}\}\\ \{\mathit{Horse}, \mathit{Odysseus}\}\end{bmatrix} \vdash^{\emptyset;\mathtt{Troy}:\mathsf{mob}[\emptyset,\{\mathit{Horse}\}]} \mathit{Horse} : \mathsf{mob}\begin{bmatrix}\{\mathit{Horse}, \mathtt{Troy}\}\\ \{\mathit{Horse}, \mathit{Odysseus}\}\end{bmatrix}$$

Typing in capability

$$\frac{\Gamma \vdash^{\Theta} a : \mathsf{mob}[\mathcal{P}, \mathcal{C}] \qquad \mathcal{P}' \subseteq \mathcal{C}}{\Gamma \vdash^{\Theta} \mathsf{in}a : \mathsf{cap}[\mathcal{P}']}$$

$$\ldots, \mathit{Troy} : \mathsf{mob}[\emptyset, \{\mathit{Horse}\}] \vdash^{\emptyset} \mathsf{in}\mathit{Troy} : \mathsf{cap}[\{\mathit{Horse}\}]$$

Typing out capability

$$\frac{\Gamma \vdash^{\Theta} a : \mathsf{mob}[\mathcal{P}, \mathcal{C}], a_i : \mathsf{mob}[\mathcal{P}_i, \mathcal{C}_i] \qquad \mathcal{P} \subseteq \mathcal{P}_i}{\Gamma \vdash^{\Theta} \mathsf{out}a : \mathsf{cap}[\{a_1, \ldots, a_n\}]}$$

$$\frac{\ldots \vdash^{\texttt{Troy}:\mathsf{mob}[\emptyset,\{\mathit{Horse}\}]} \mathit{Horse} : \mathsf{mob}[\{\mathit{Horse}, \texttt{Troy}\}, \{\mathit{Horse}, \mathit{Odysseus}\}] \quad \ldots \vdash^{\texttt{Troy}:\mathsf{mob}[\emptyset,\{\mathit{Horse}\}]} \mathit{Odysseus} : \mathsf{mob}[\{\mathit{Horse}\}, \emptyset]}{\ldots \not\vdash^{\texttt{Troy}:\mathsf{mob}[\emptyset,\{\mathit{Horse}\}]} \mathsf{out}\mathit{Horse} : \mathsf{cap}[\{\mathit{Odysseus}\}]}$$

Scope restriction

$$\frac{\Gamma^{(a:\mathsf{mob}[\mathcal{P},\mathcal{C}])}, a : \mathsf{mob}[\mathcal{P},\mathcal{C}] \vdash_b^{\Xi;\Theta} P}{\Gamma \vdash_b^{\mathsf{a}:\mathsf{mob}[\mathcal{P},\mathcal{C}],\Xi;\Theta} (\nu a : \mathsf{mob}[\mathcal{P},\mathcal{C}])P}$$

$$\frac{\begin{array}{l}\mathit{Horse} : \mathsf{mob}[\{\mathit{Horse}, \mathit{Troy}\}, \{\mathit{Horse}, \mathit{Odysseus}\}] \\ \mathit{Odysseus} : \mathsf{mob}[\mathit{Horse}, \emptyset] \\ \mathit{Troy} : \mathsf{mob}[\emptyset, \{\mathit{Horse}\}]\end{array} \quad \vdash^{\emptyset;\emptyset} (\cdots)}{\begin{array}{l}\mathit{Horse} : \mathsf{mob}[\{\mathit{Horse}\}, \{\mathit{Horse}, \mathit{Odysseus}\}] \\ \mathit{Odysseus} : \mathsf{mob}[\mathit{Horse}, \emptyset]\end{array} \quad \vdash^{\mathtt{Troy}:\mathsf{mob}[\emptyset,\{\mathit{Horse}\}];\emptyset} (\nu \mathit{Troy} : \mathsf{mob}[\emptyset, \{\mathit{Horse}\}])(\cdots)}$$

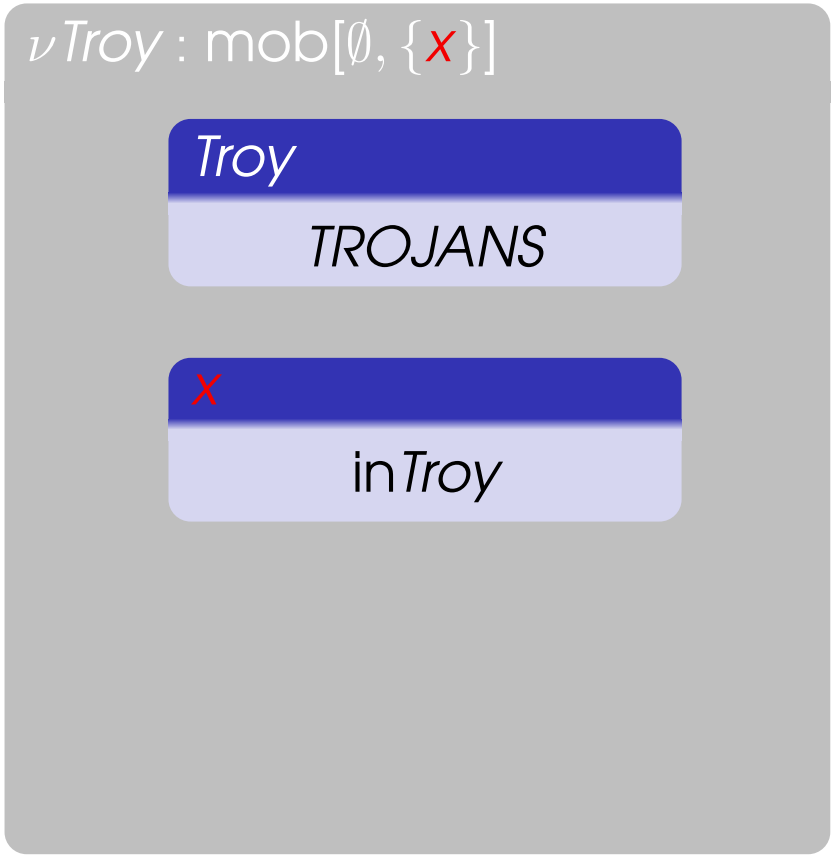
Parallel composition

$$\frac{\Gamma \vdash^{\Theta_1;\Xi,\Theta_2} P \qquad \Gamma \vdash^{\Theta_2;\Xi,\Theta_1} Q}{\Gamma \vdash^{\Theta_1,\Theta_2;\Xi} P \mid Q}$$

$$\frac{\Gamma \vdash^{\emptyset;\texttt{Troy}:\mathsf{mob}[\emptyset,\{Horse\}]} Horse[\cdots] \qquad \Gamma \vdash^{\texttt{Troy}:\mathsf{mob}[\emptyset,\{Horse\}];\emptyset} (\nu\, Troy)(\cdots)}{\Gamma \vdash^{\texttt{Troy}:\mathsf{mob}[\emptyset,\{Horse\}];\emptyset} Horse[\cdots] \mid (\nu\, Troy)(\cdots)}$$

The Effect of Communication: Dynamic Types

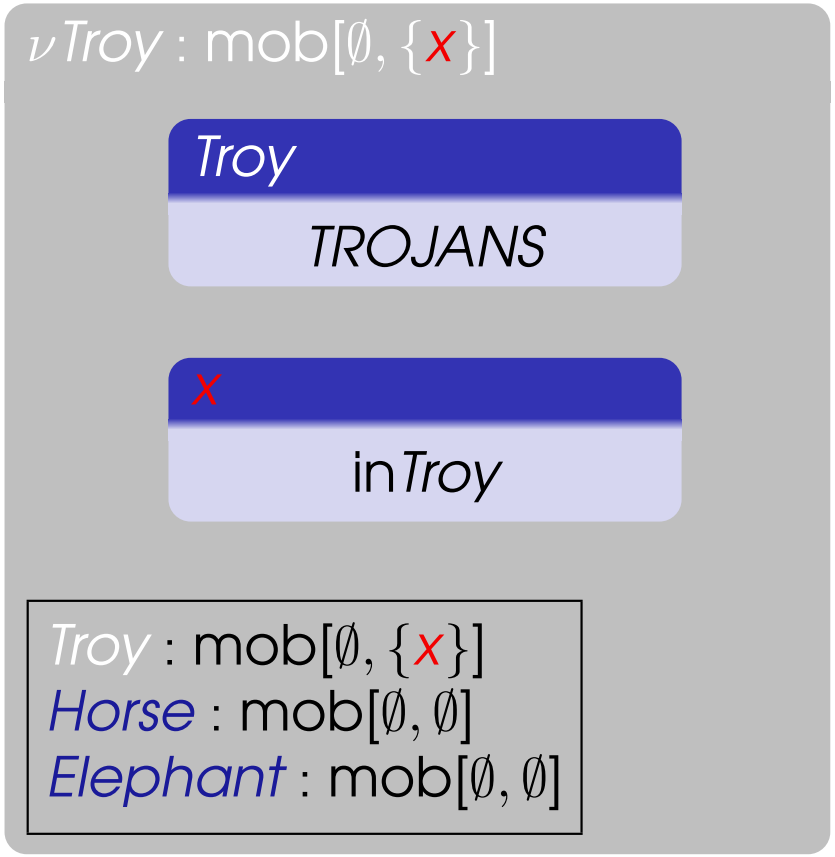
$\langle Elephant \rangle \mid \langle Horse \rangle \mid (x).$

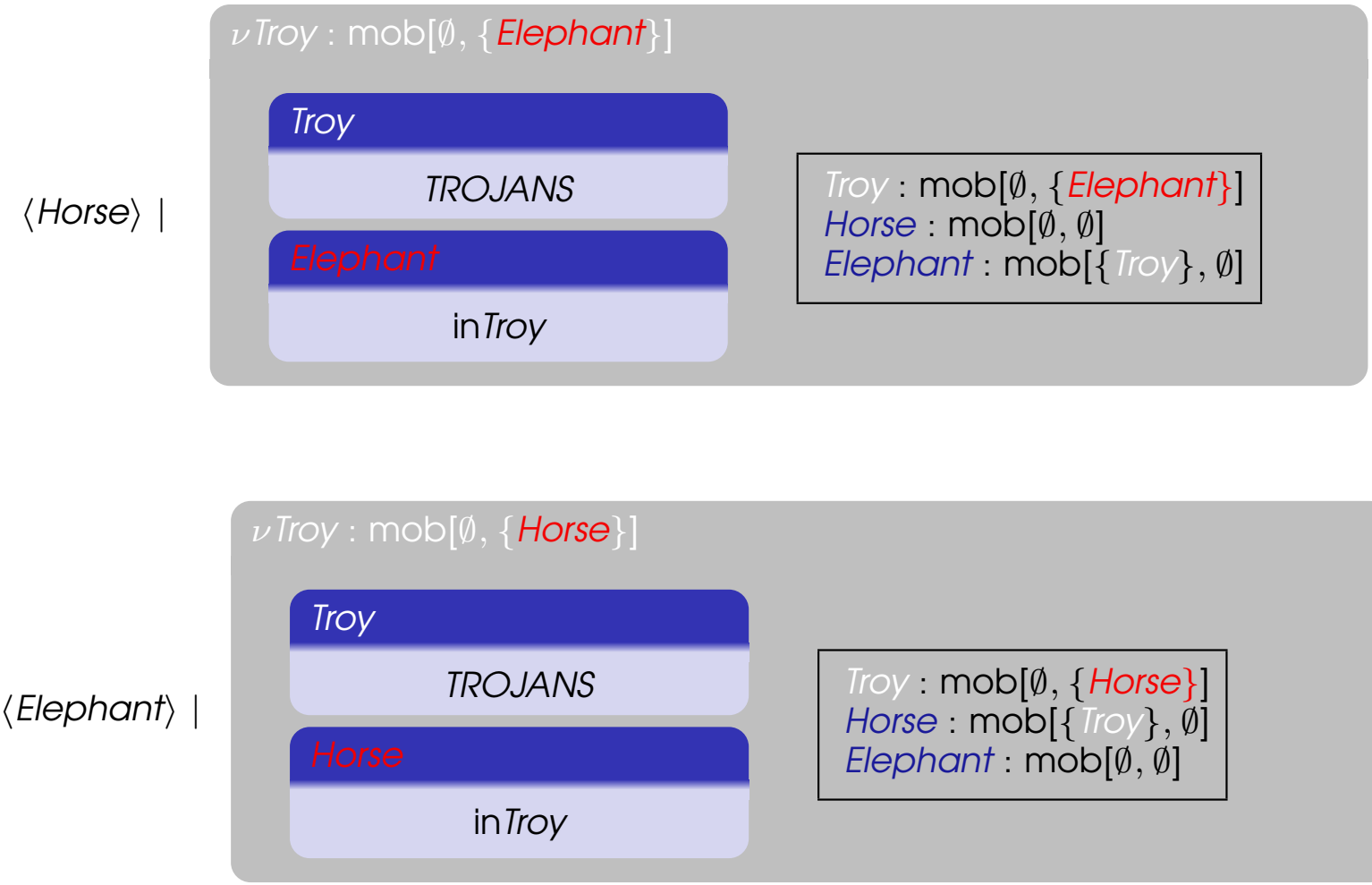


The Effect of Communication: Dynamic Types

$\langle Elephant \rangle \mid \langle Horse \rangle \mid (x).$

$Horse : \mathrm{mob}[\emptyset, \emptyset]$
$Elephant : \mathrm{mob}[\emptyset, \emptyset]$





- Orthogonal policies depending on possible communications: need to track of all possible types

New types

- Ambient types

$$a : \mathrm{amb}[\mathrm{mob}[\mathcal{P}, \mathcal{C}], \mathrm{com}[\mathcal{E}, \mathcal{L}]]$$

  $\mathcal{E}$: set of ambient names where a might be communicated
  $\mathcal{L}$: set of ambient names that might be communicated inside a

- Variable types

$$x : \mathrm{var}[\mathcal{B}]$$

  $\mathcal{B}$: set of ambient names that might be bound to x

- Multiple types

$$\mathit{Horse} : \begin{cases} \mathrm{amb}[\mathrm{mob}[\emptyset, \emptyset], \mathrm{com}[\{\mathit{top}, \emptyset\}]], \\ \mathrm{amb}[\mathrm{mob}[\{\mathit{Troy}\}, \emptyset], \mathrm{com}[\{\mathit{top}, \emptyset\}]] \end{cases}$$

Conclusion

- Name dependent typing:
  - simple and intuitive types against "nasty" typing rules;
  - still relatively easy, yet more flexible and expressive than groups;
  - sensible application: access control for personalised, dynamic services.

Conclusion

- Name dependent typing:
  - simple and intuitive types against "nasty" typing rules;
  - still relatively easy, yet more flexible and expressive than groups;
  - sensible application: access control for personalised, dynamic services.

- Central technical notion: abstract names and contexts. Keep track of capabilities "acquirable" dynamically by "crossing" names' scopes.

Conclusion

- Name dependent typing:
  - simple and intuitive types against "nasty" typing rules;
  - still relatively easy, yet more flexible and expressive than groups;
  - sensible application: access control for personalised, dynamic services.
- Central technical notion: abstract names and contexts. Keep track of capabilities "acquirable" dynamically by "crossing" names' scopes.
- Related work in the literature:
  - MIKADO's dynamic types for DPI (Hennessy et al.).
  - Yoshida's existential dependent types for DPI.
  - DART's dependent types for the Ambient Calculus (Amtoft and Wells).