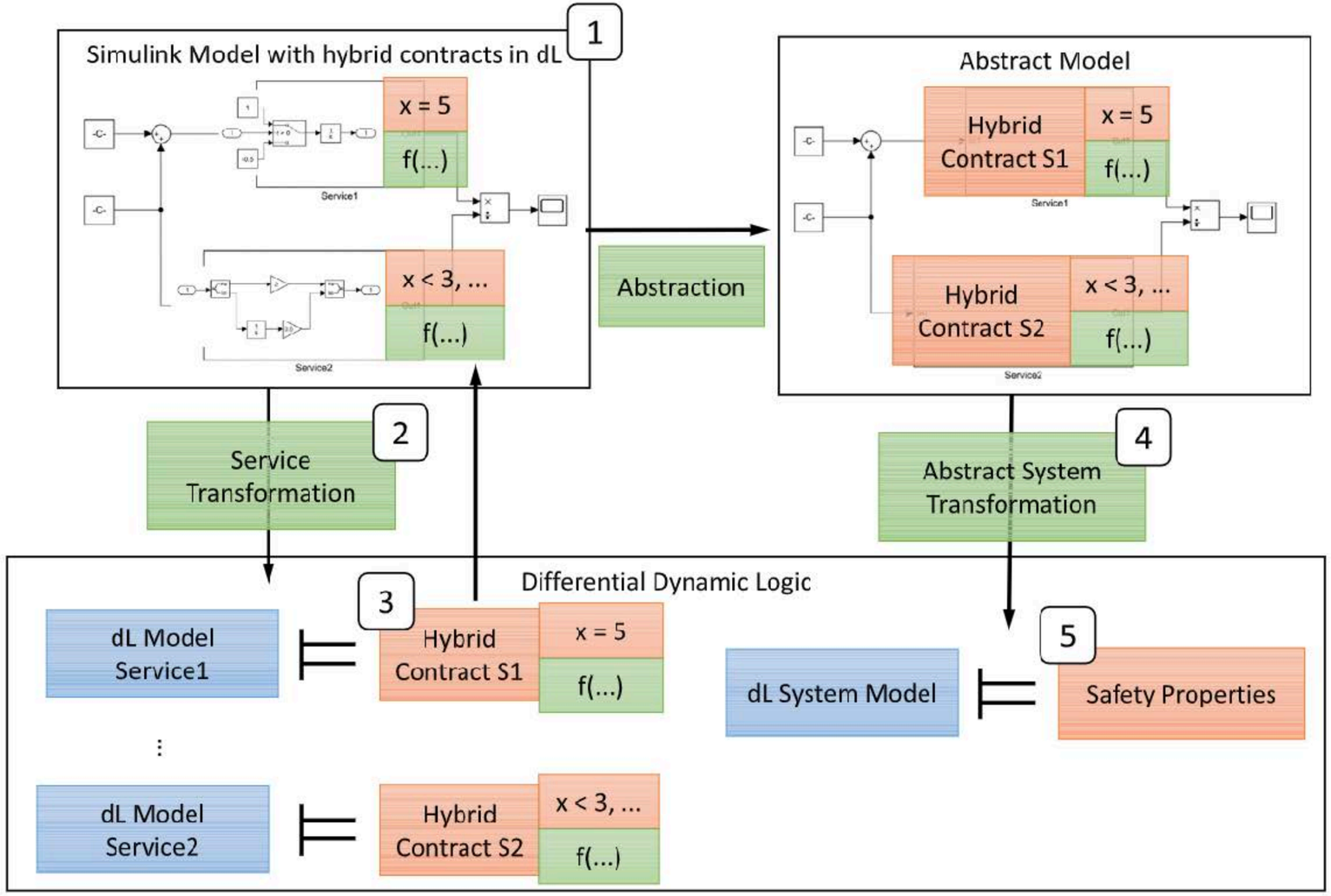
Applications of KeYmaera X

Semantics of and compositionality in Simulink



Science of Computer Programming 211 (2021) 102694

Contents lists available at ScienceDirect

Science of Computer Programming

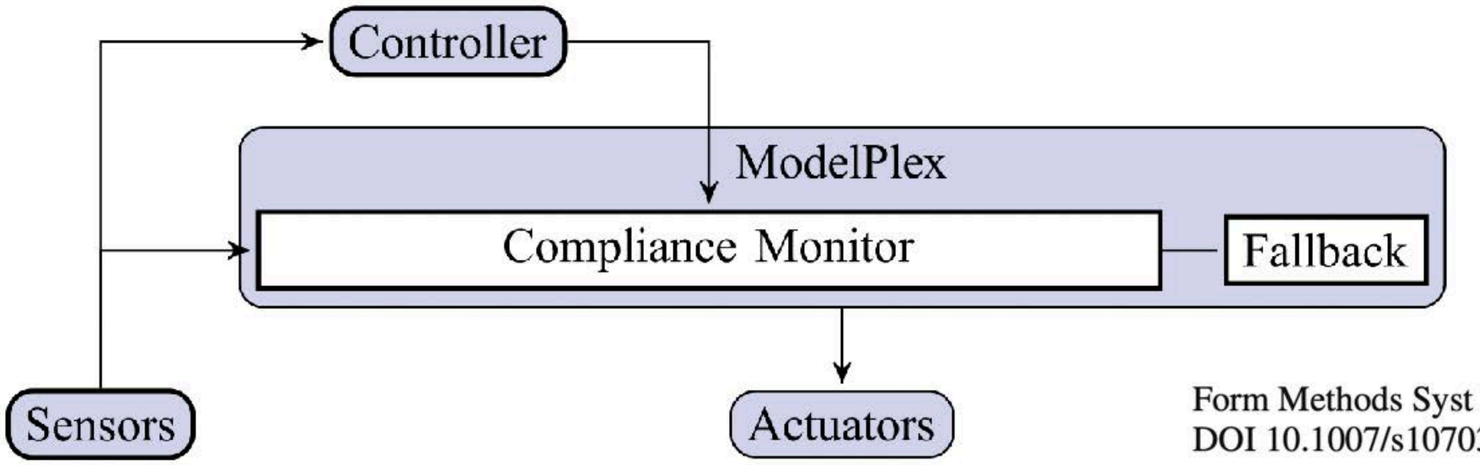
www.elsevier.com/locate/scico

Service-oriented decomposition and verification of hybrid system models using feature models and contracts ☆

Timm Liebrenz [a,*], Paula Herber [a,*], Sabine Glesner [b]

[a] Embedded Systems Group, University of Münster, Schlossplatz 2, 48149 Münster, Germany
[b] Software and Embedded Systems Engineering Group, Technische Universität Berlin, Straße des 17. Juni 135, 10623 Berlin, Germany

ModelPlex: models as monitors

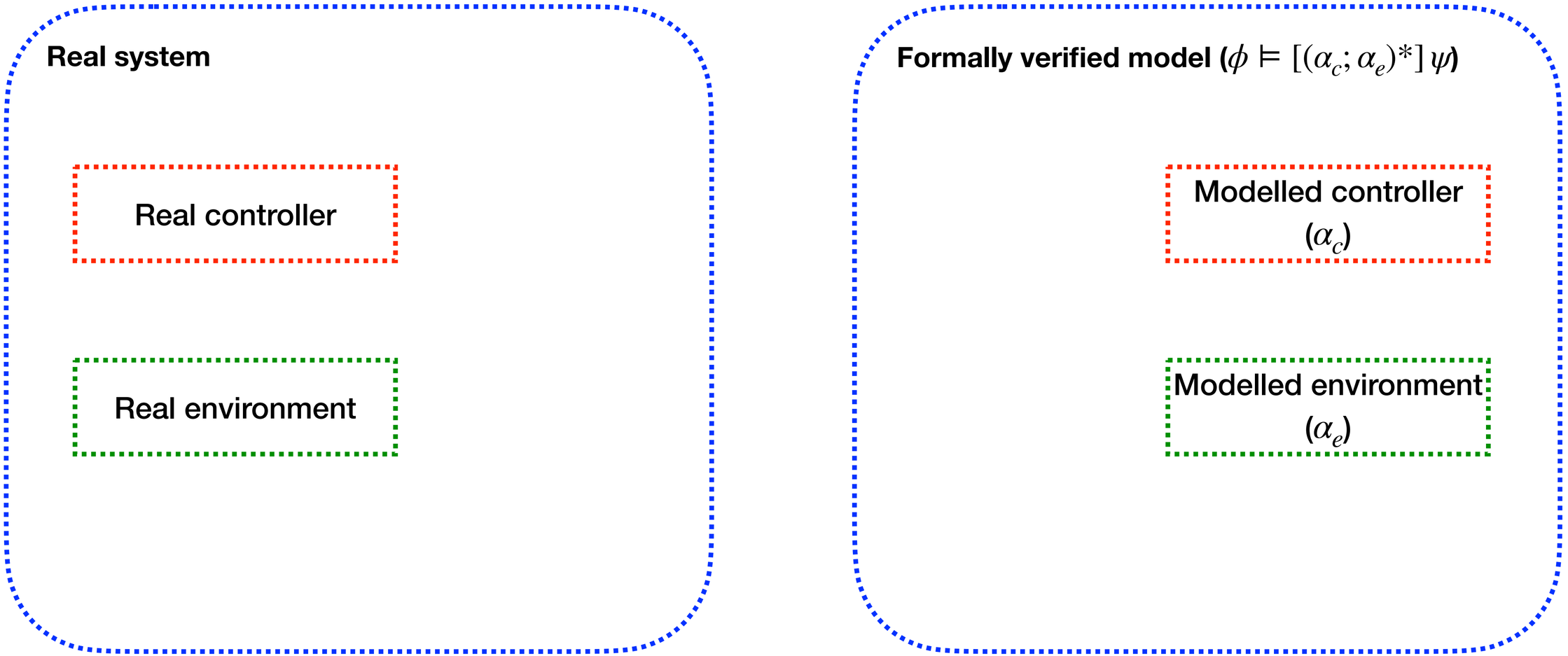


Form Methods Syst Des (2016) 49:33–74
DOI 10.1007/s10703-016-0241-z

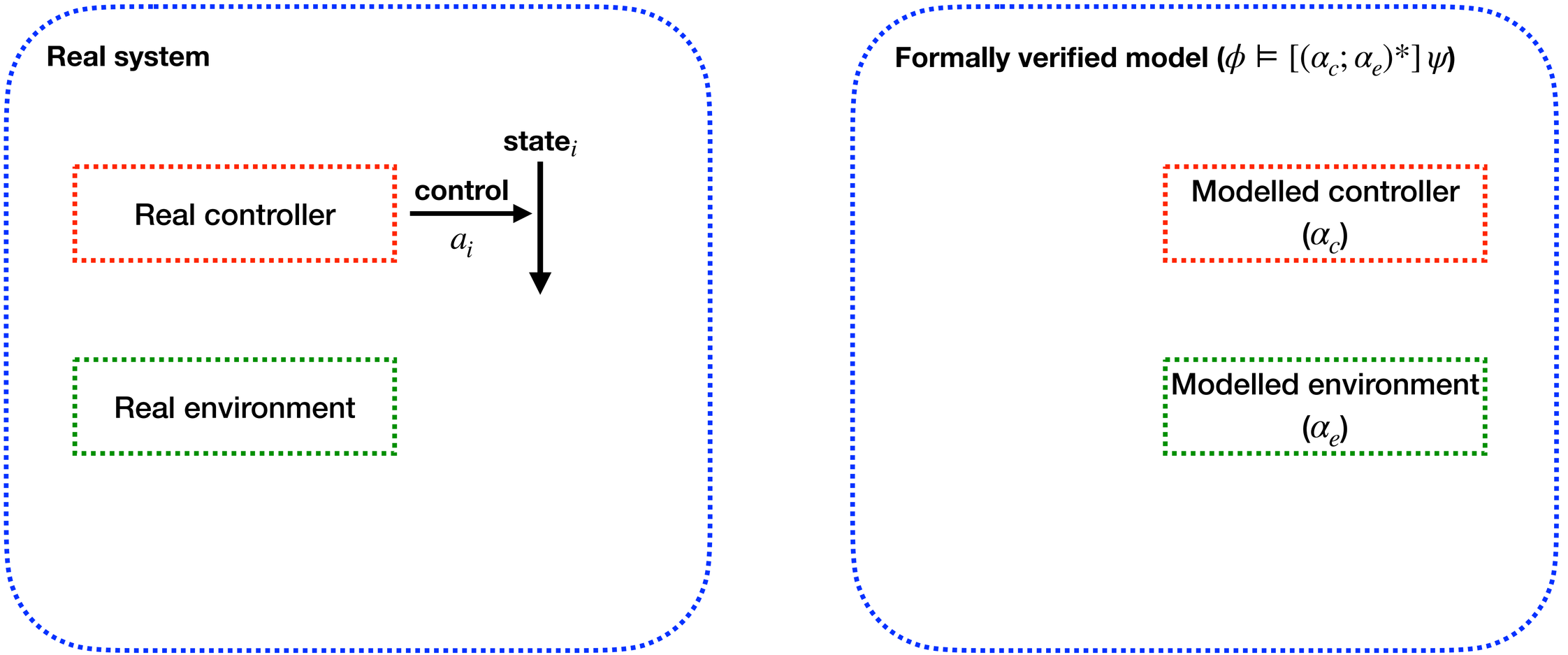
ModelPlex: verified runtime validation of verified cyber-physical system models

Stefan Mitsch[1,2] · André Platzer[1]

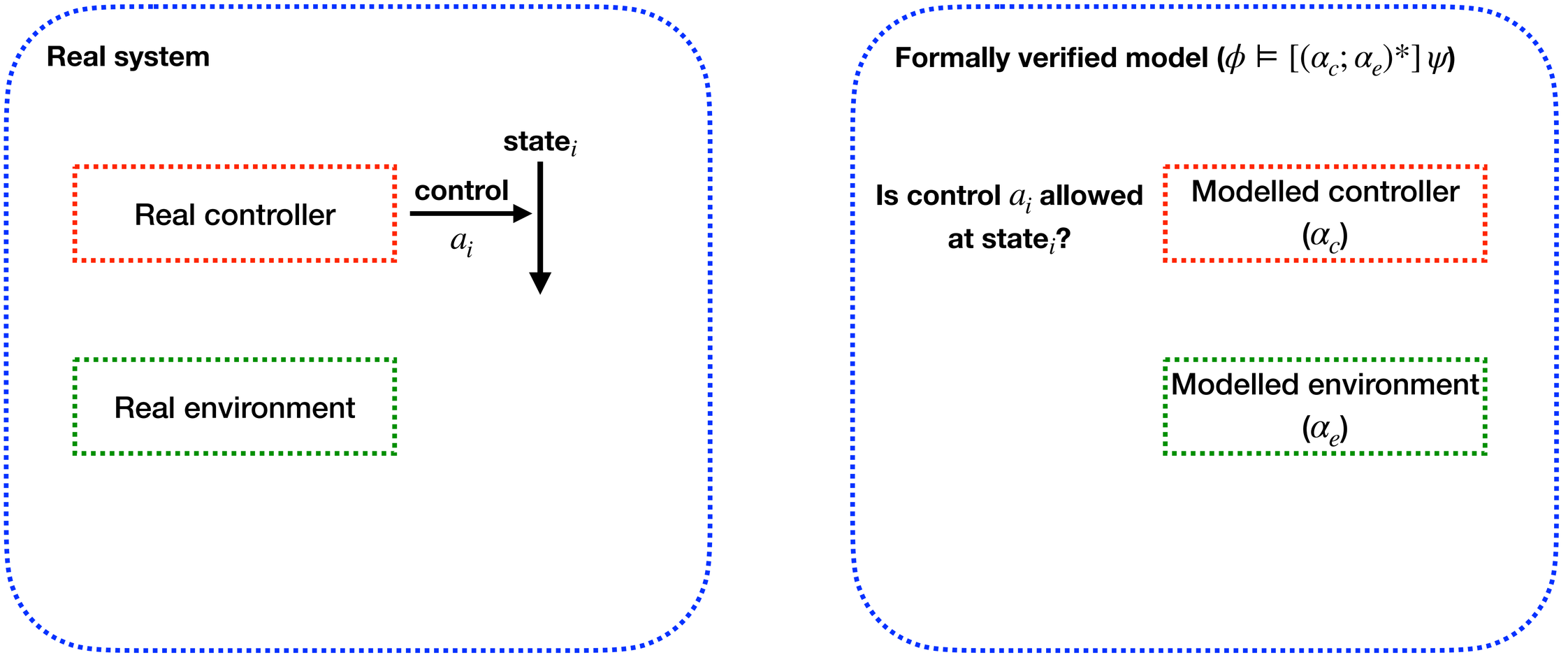
ModelPlex: models as monitors



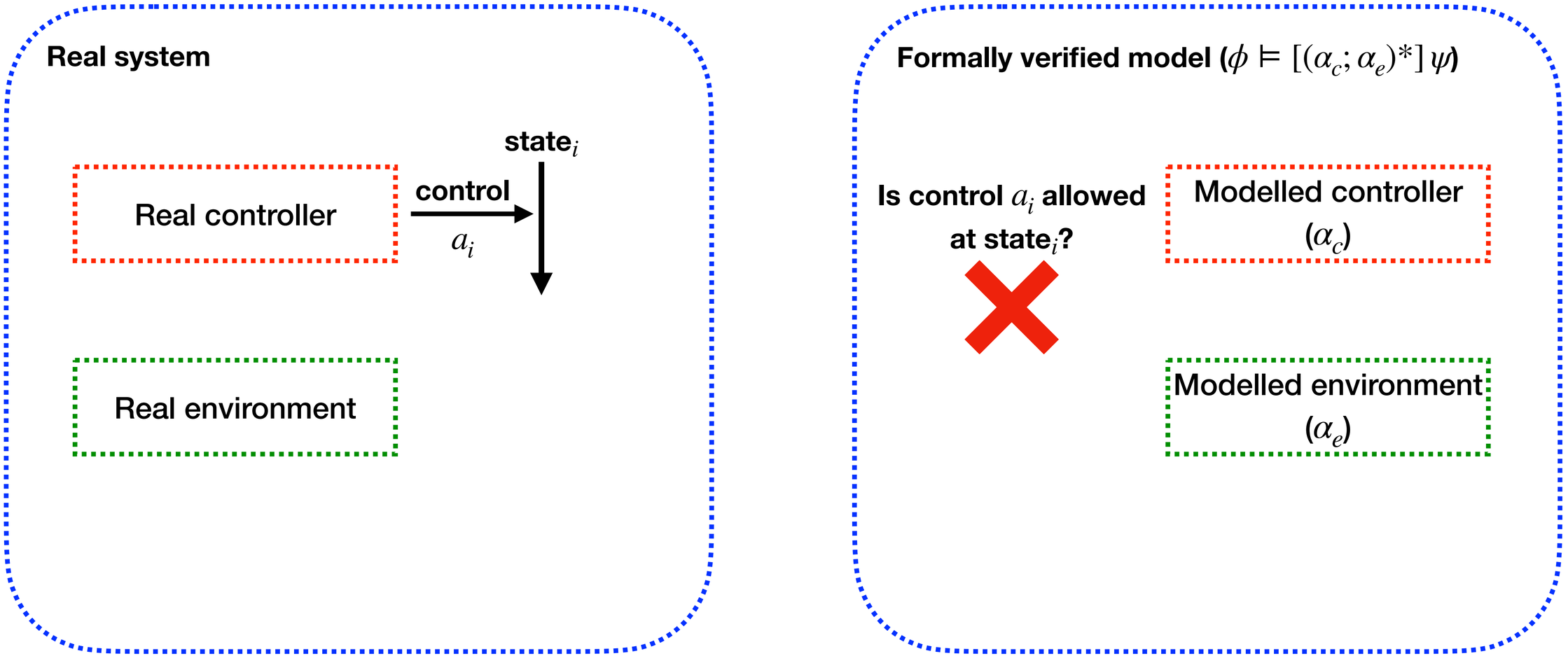
ModelPlex: models as monitors



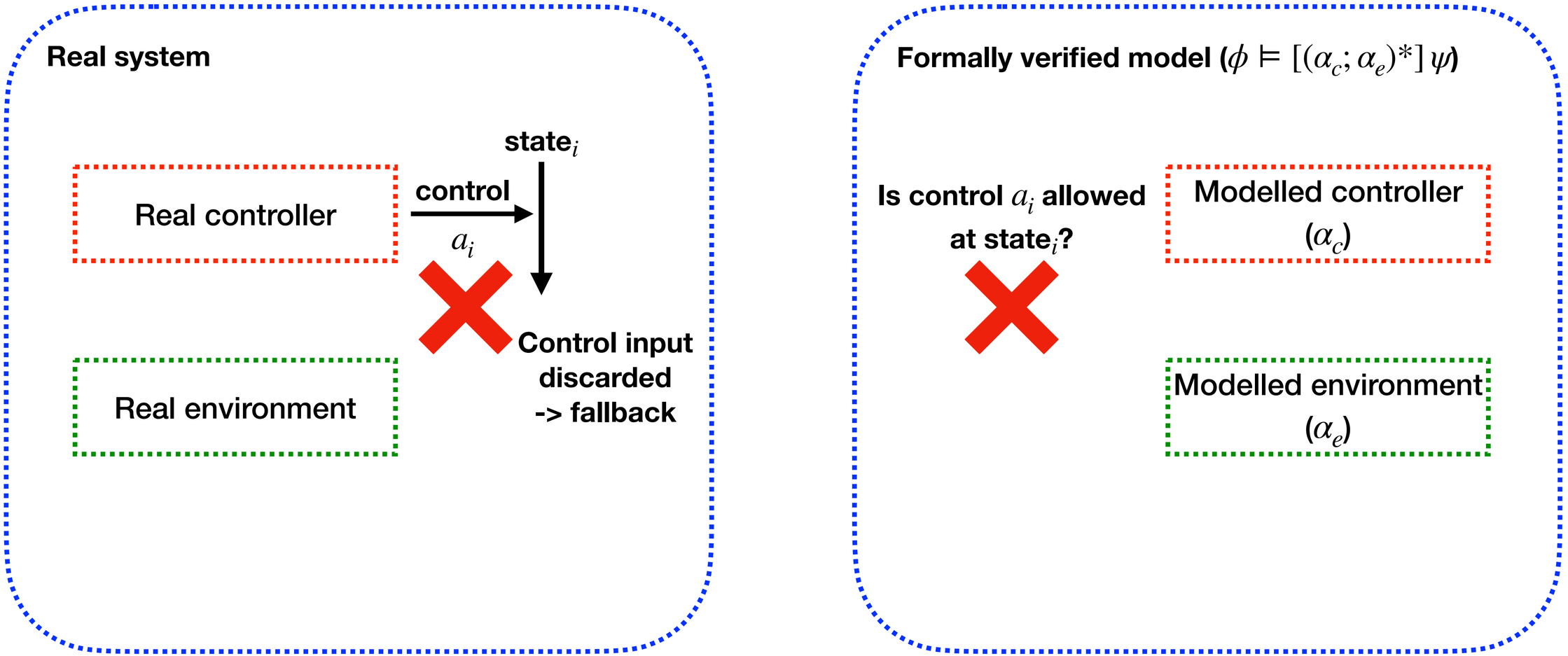
ModelPlex: models as monitors



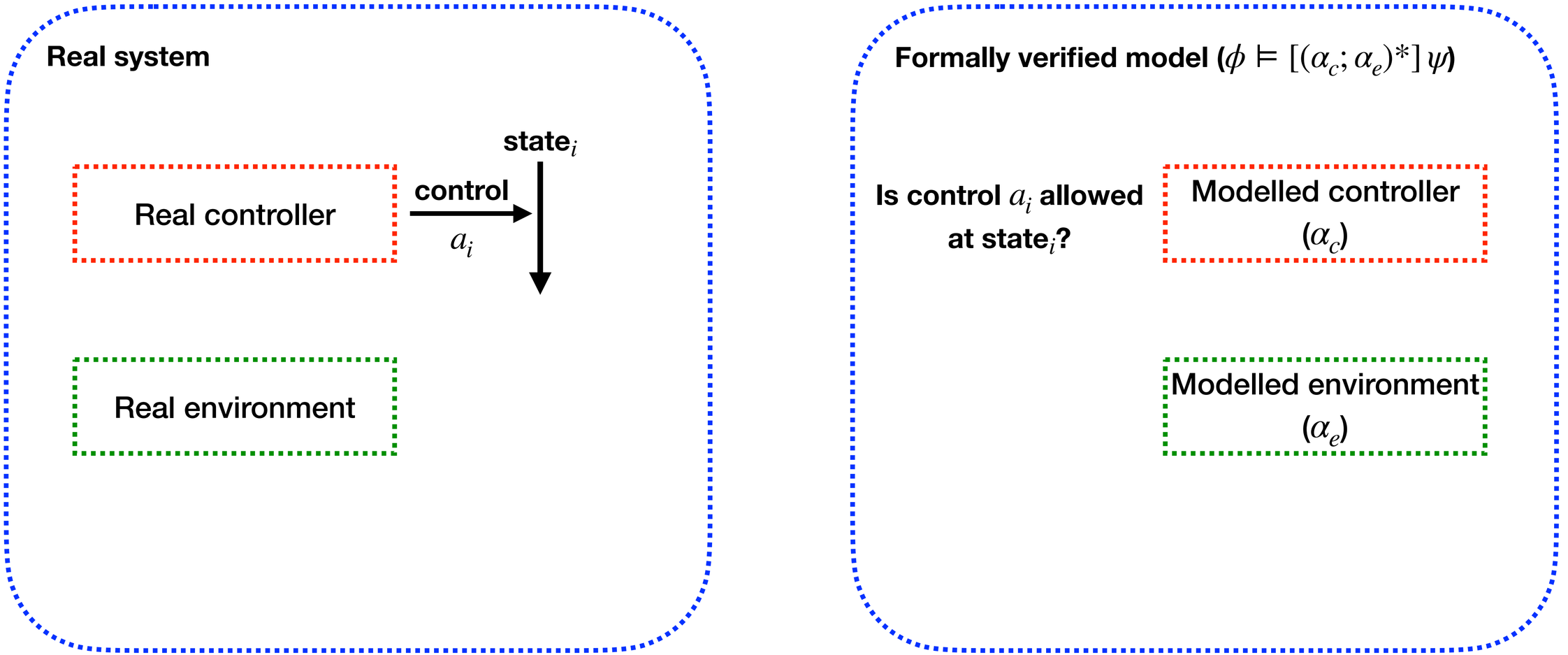
ModelPlex: models as monitors



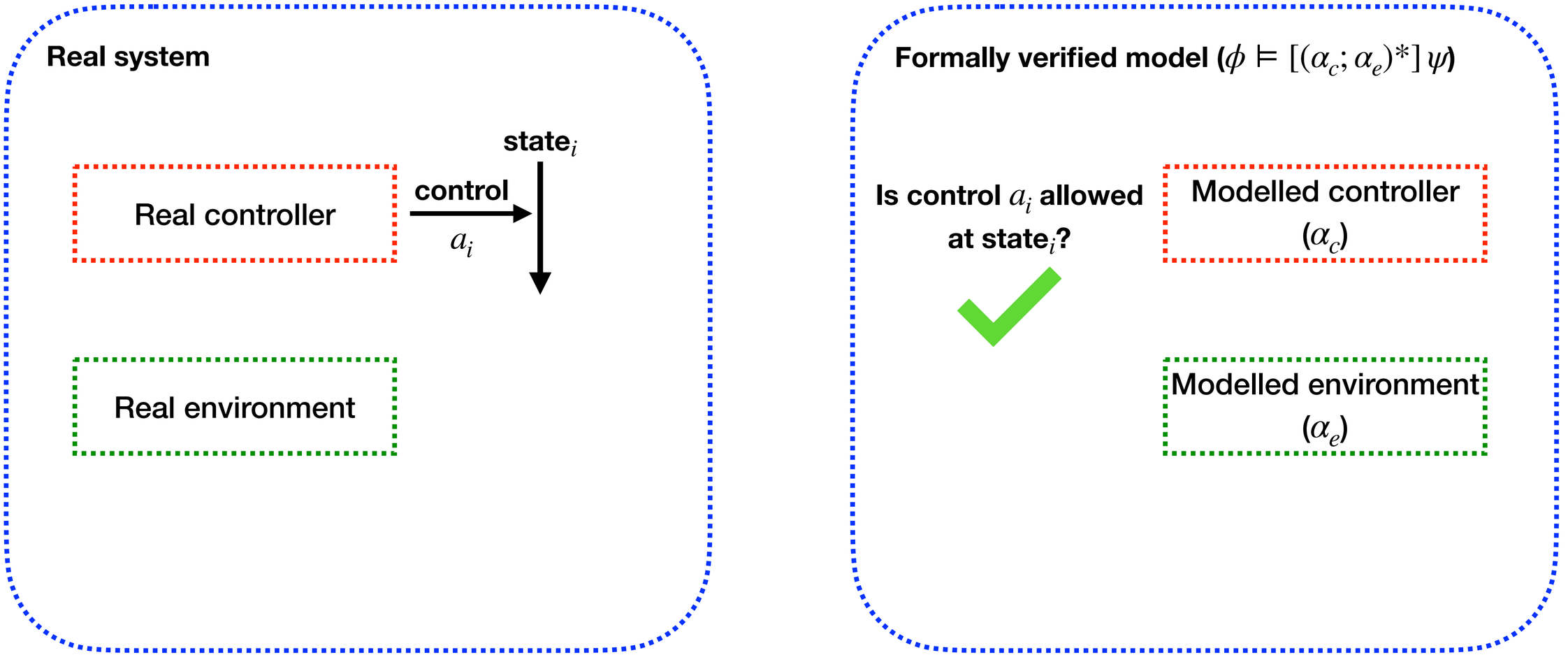
ModelPlex: models as monitors



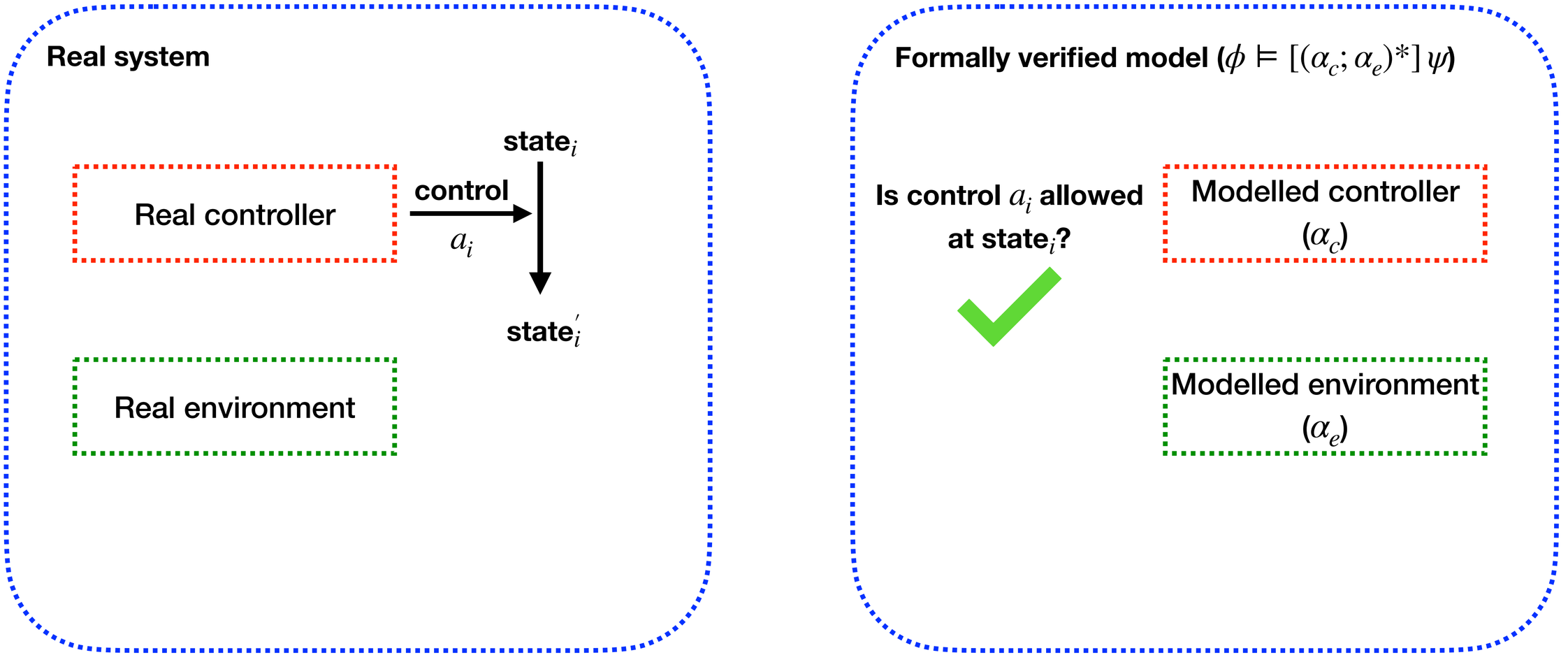
ModelPlex: models as monitors



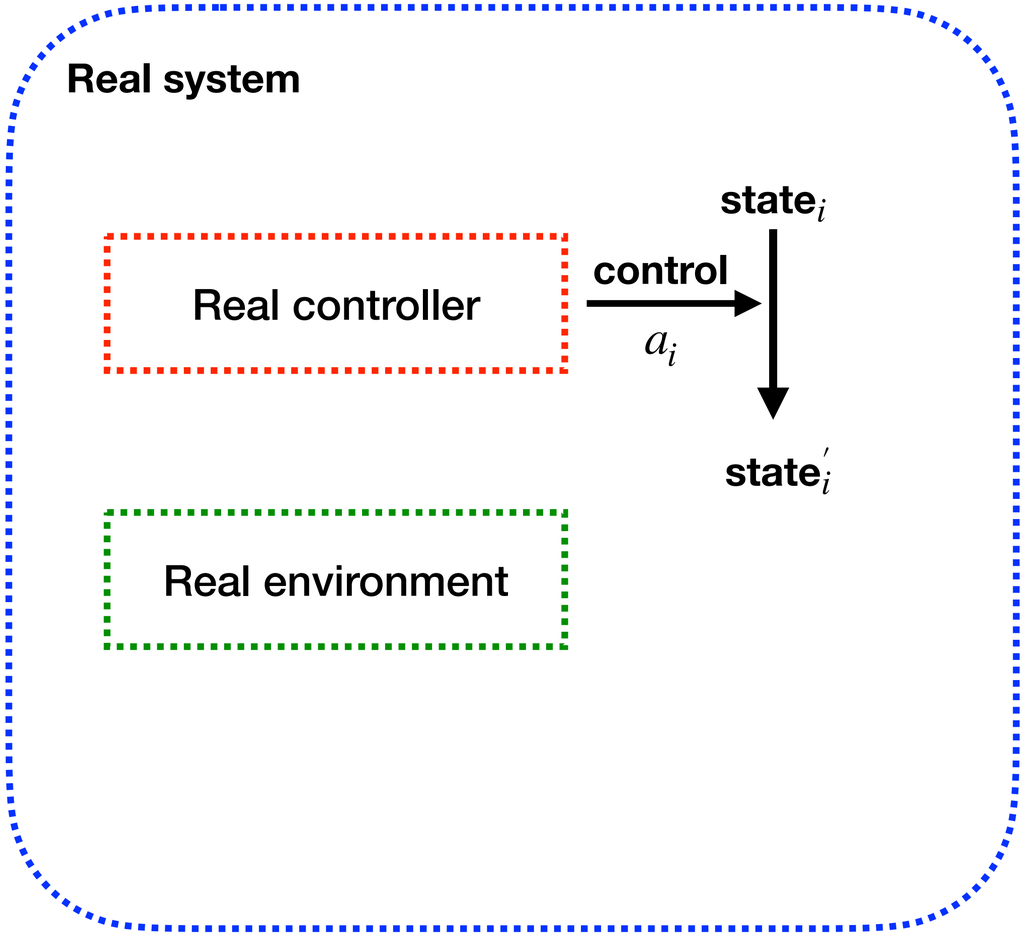
ModelPlex: models as monitors

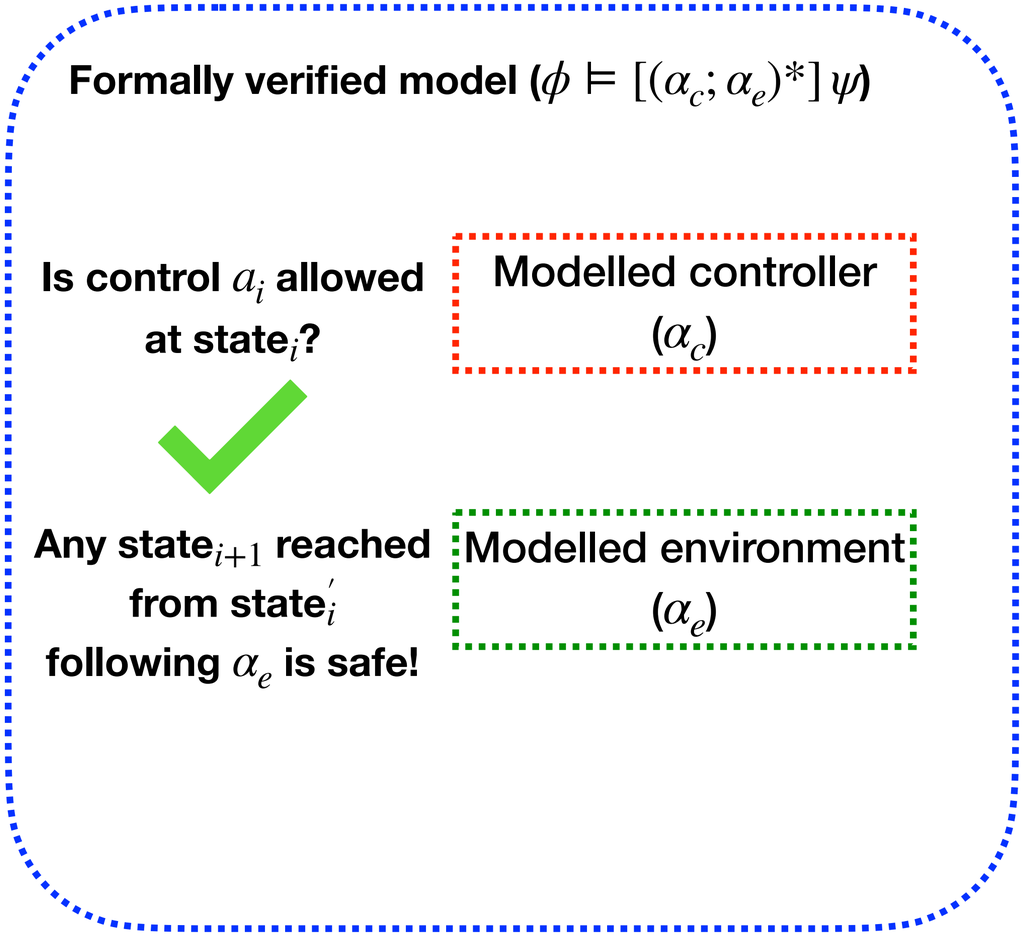


ModelPlex: models as monitors

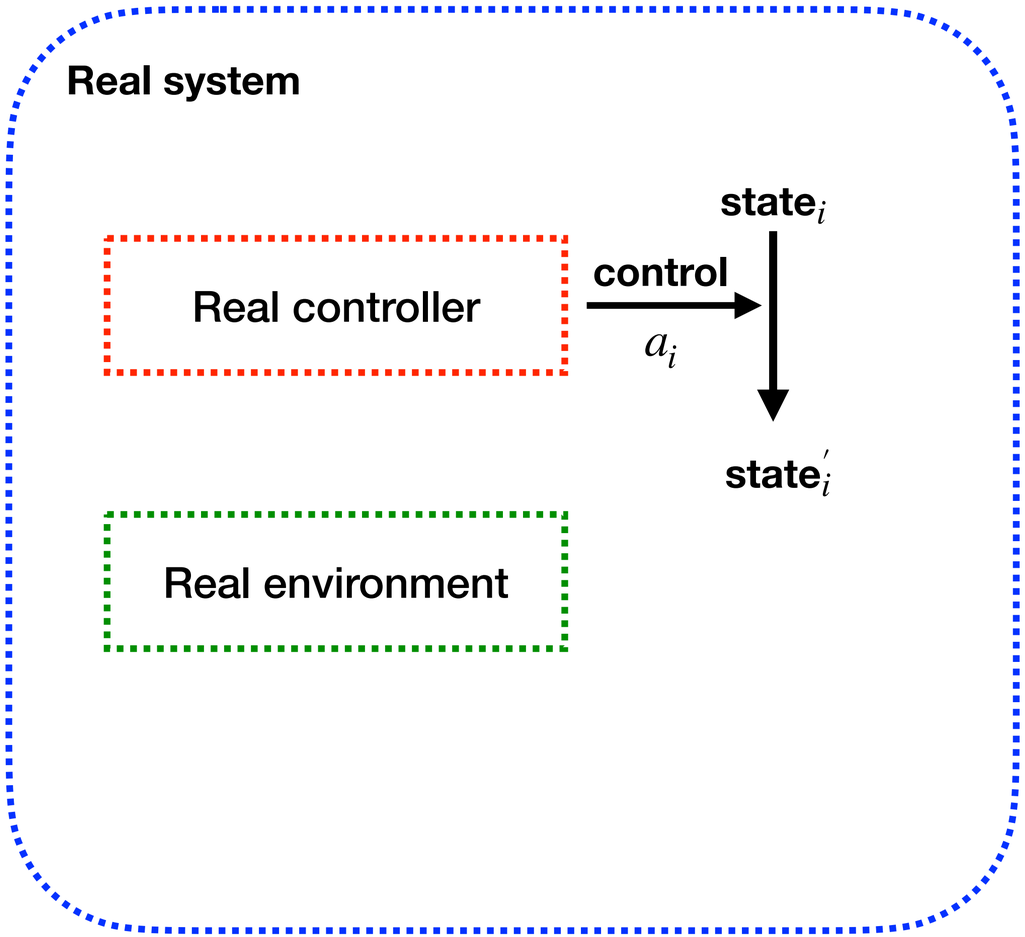


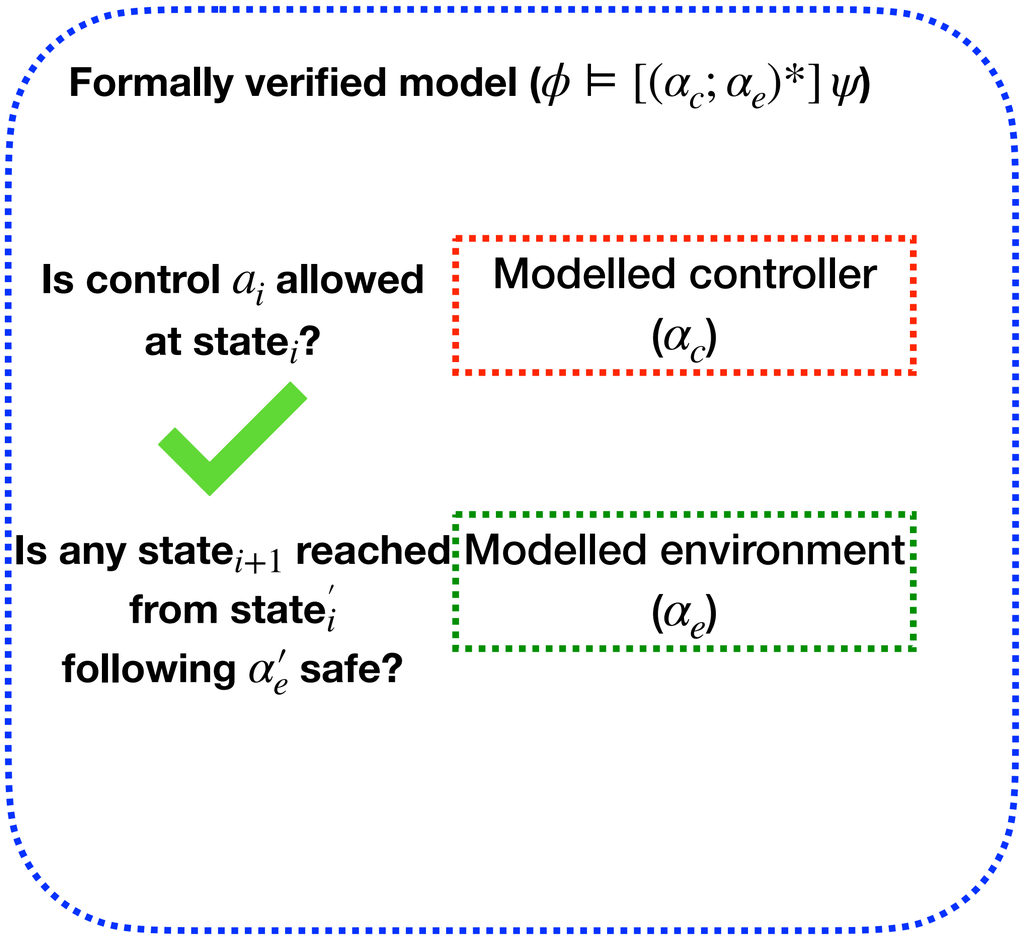
ModelPlex: models as monitors



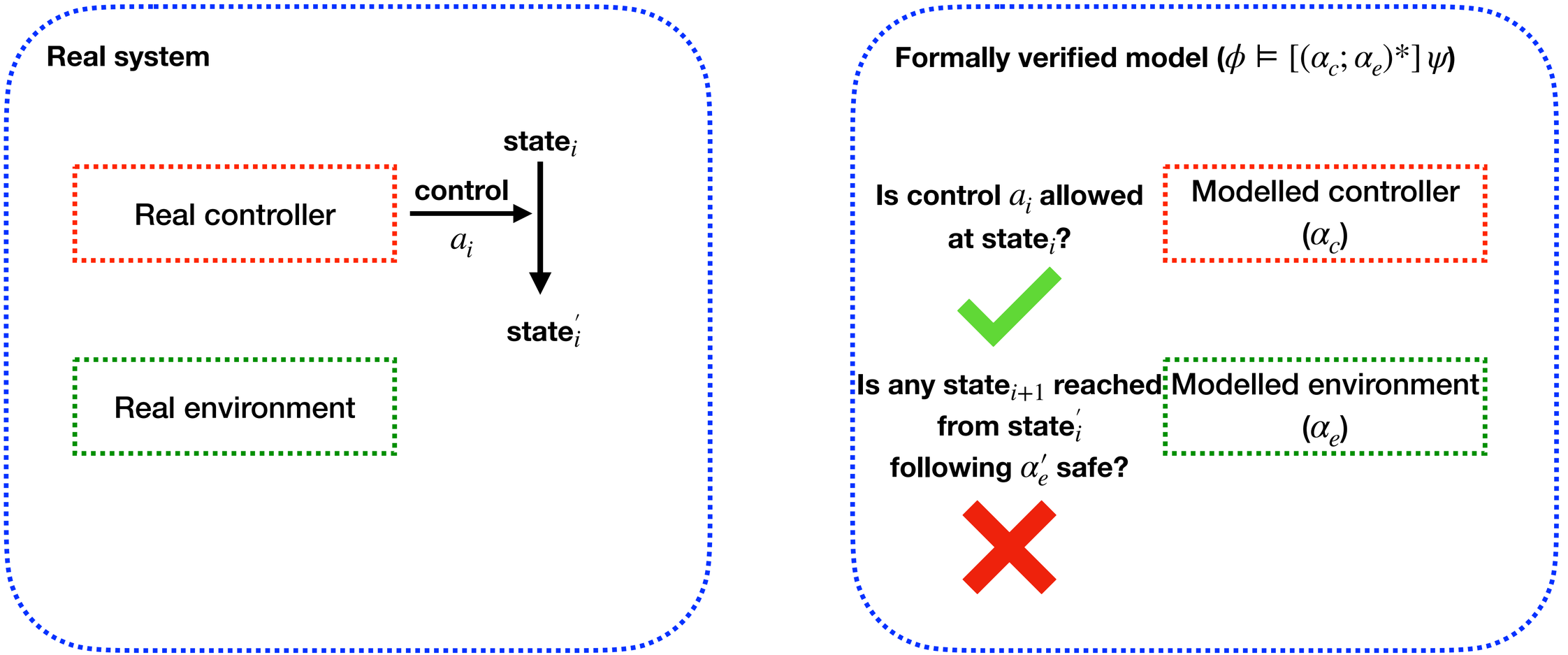


ModelPlex: models as monitors

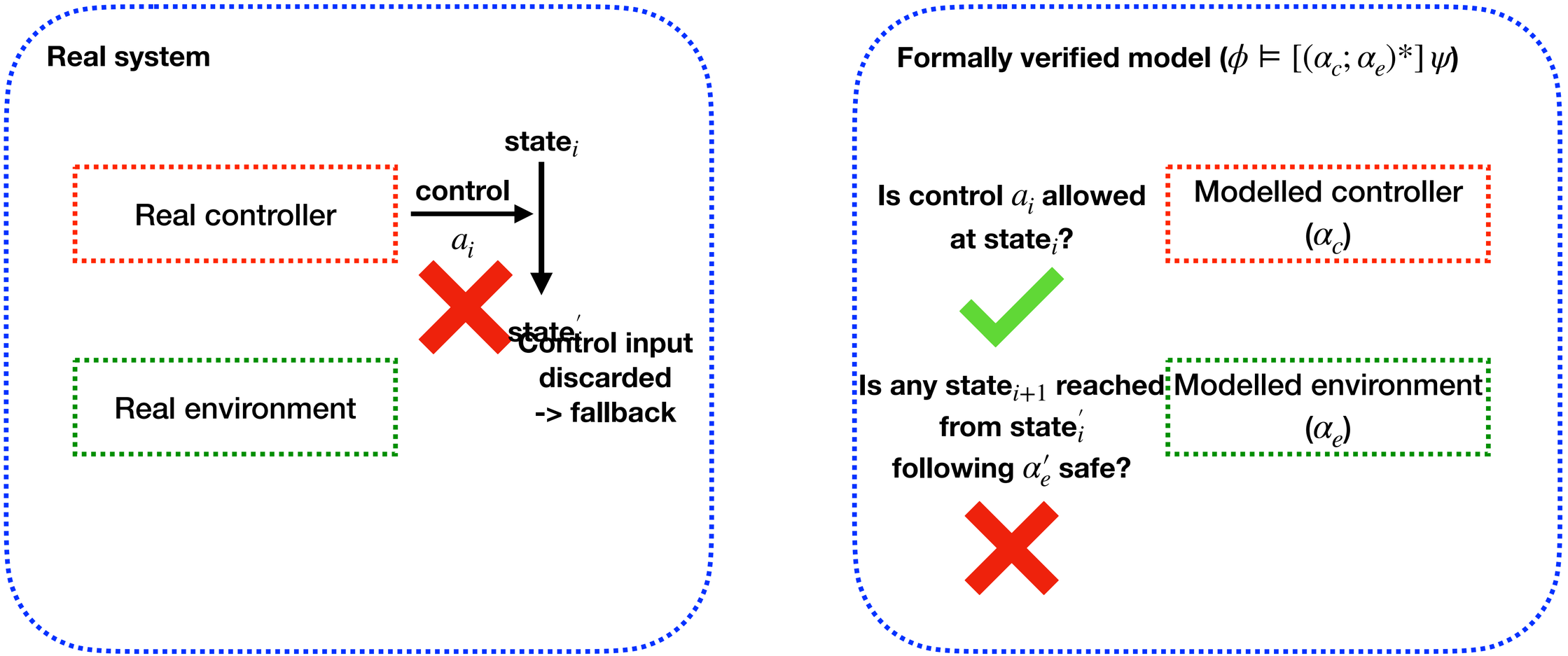




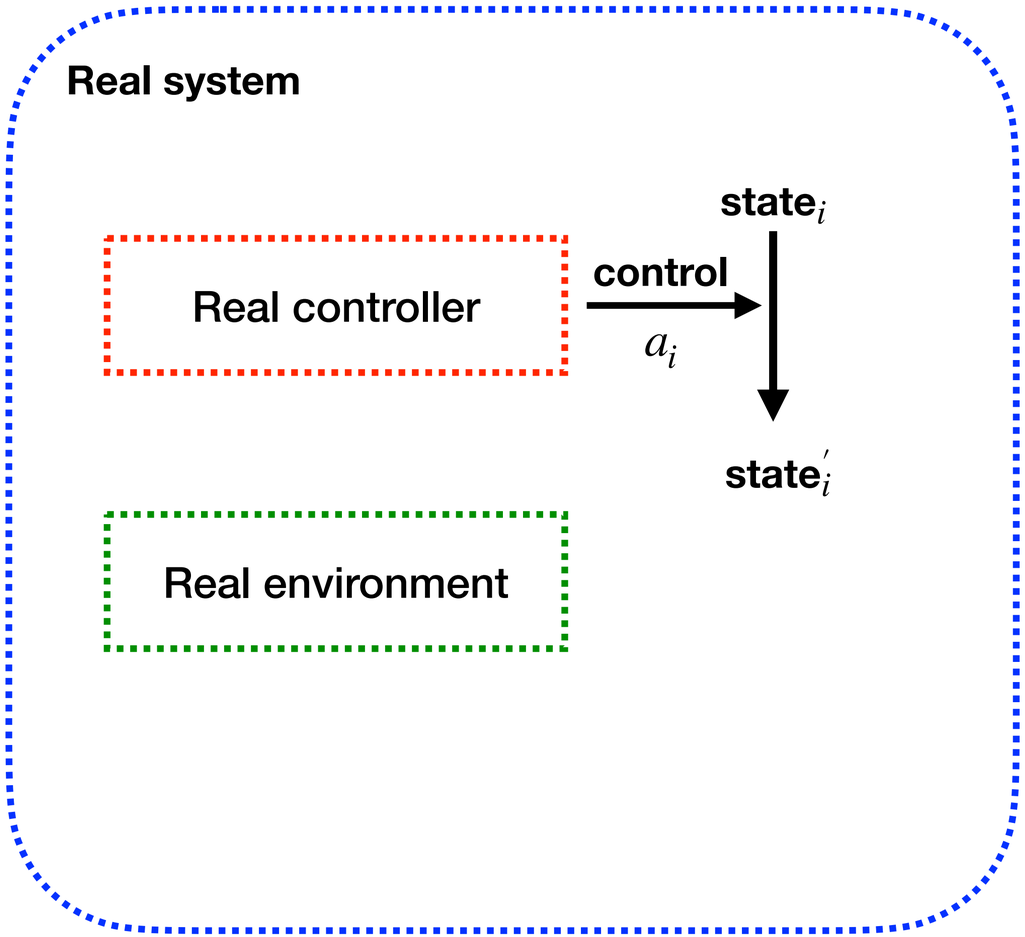
ModelPlex: models as monitors

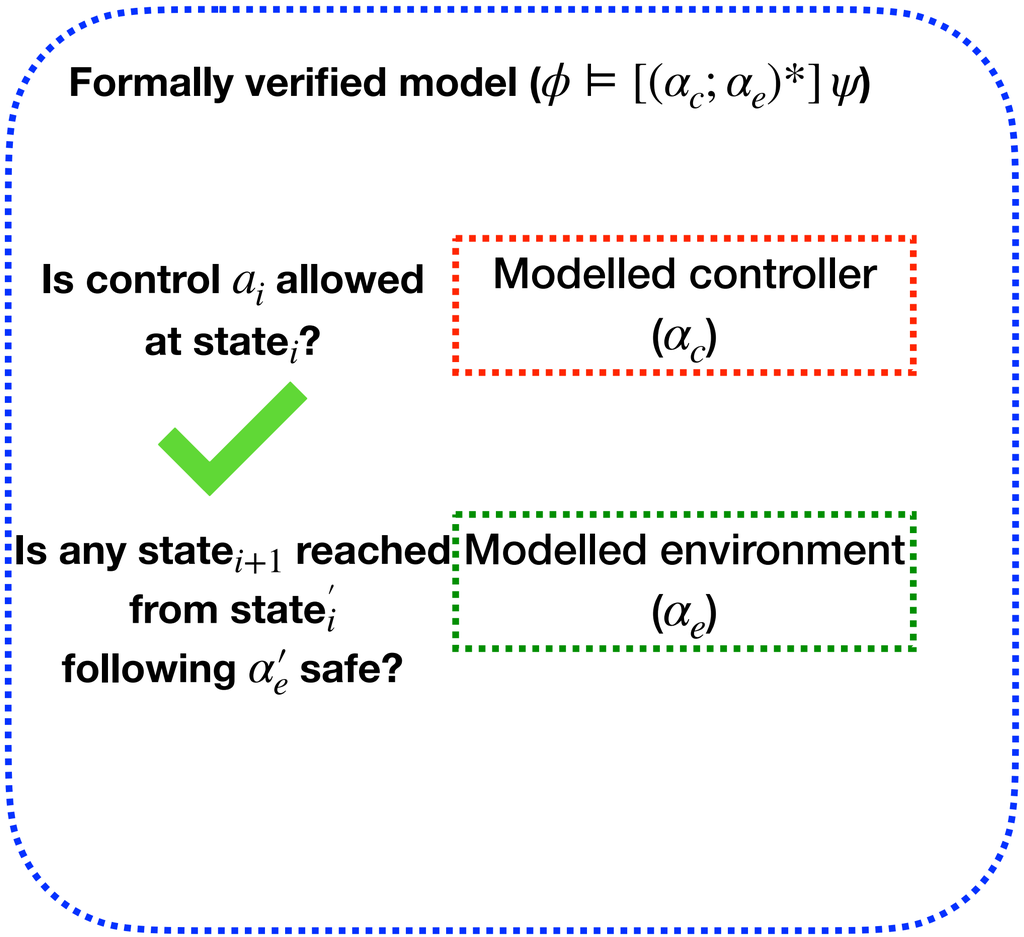


ModelPlex: models as monitors

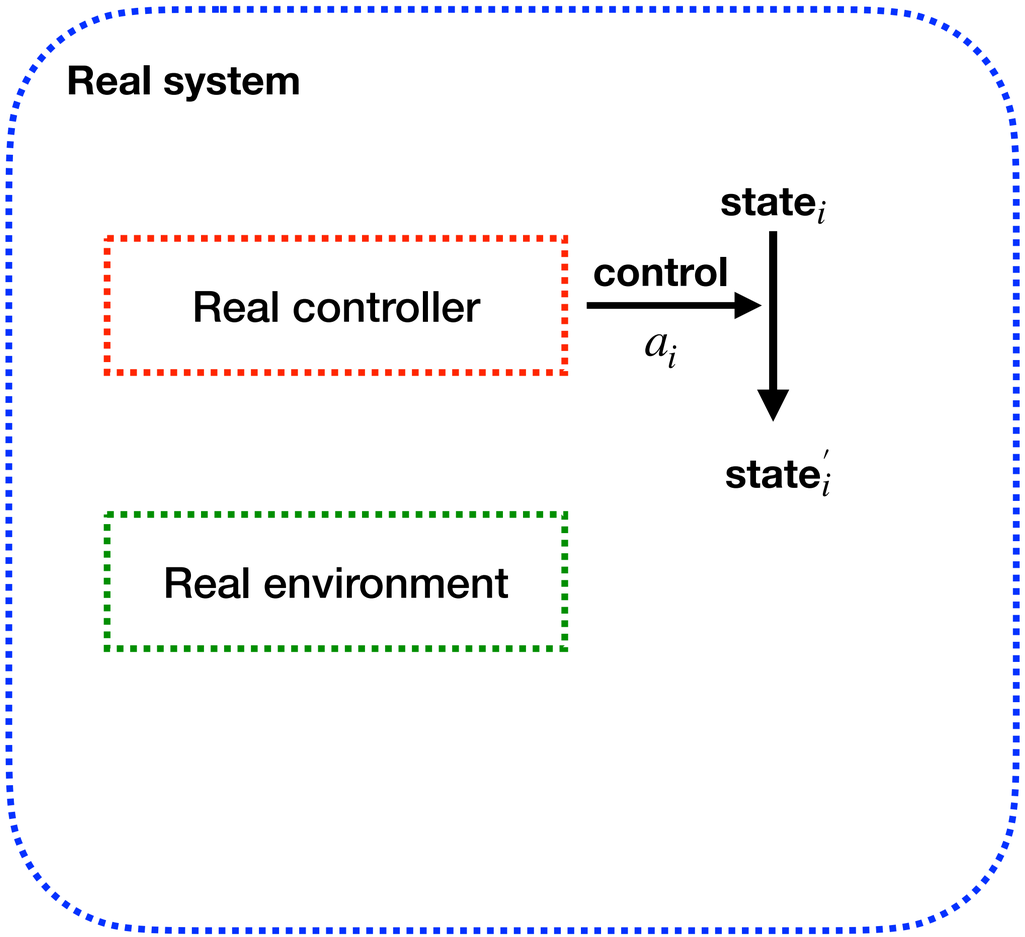


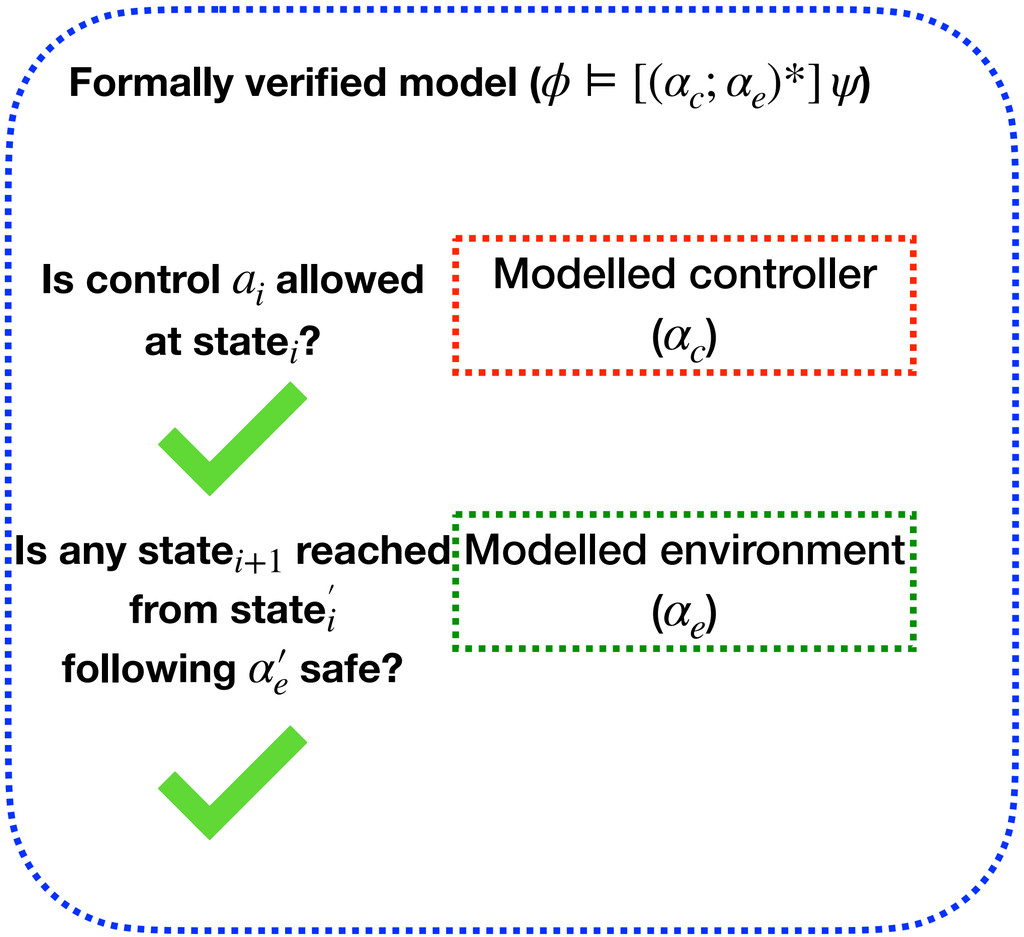
ModelPlex: models as monitors



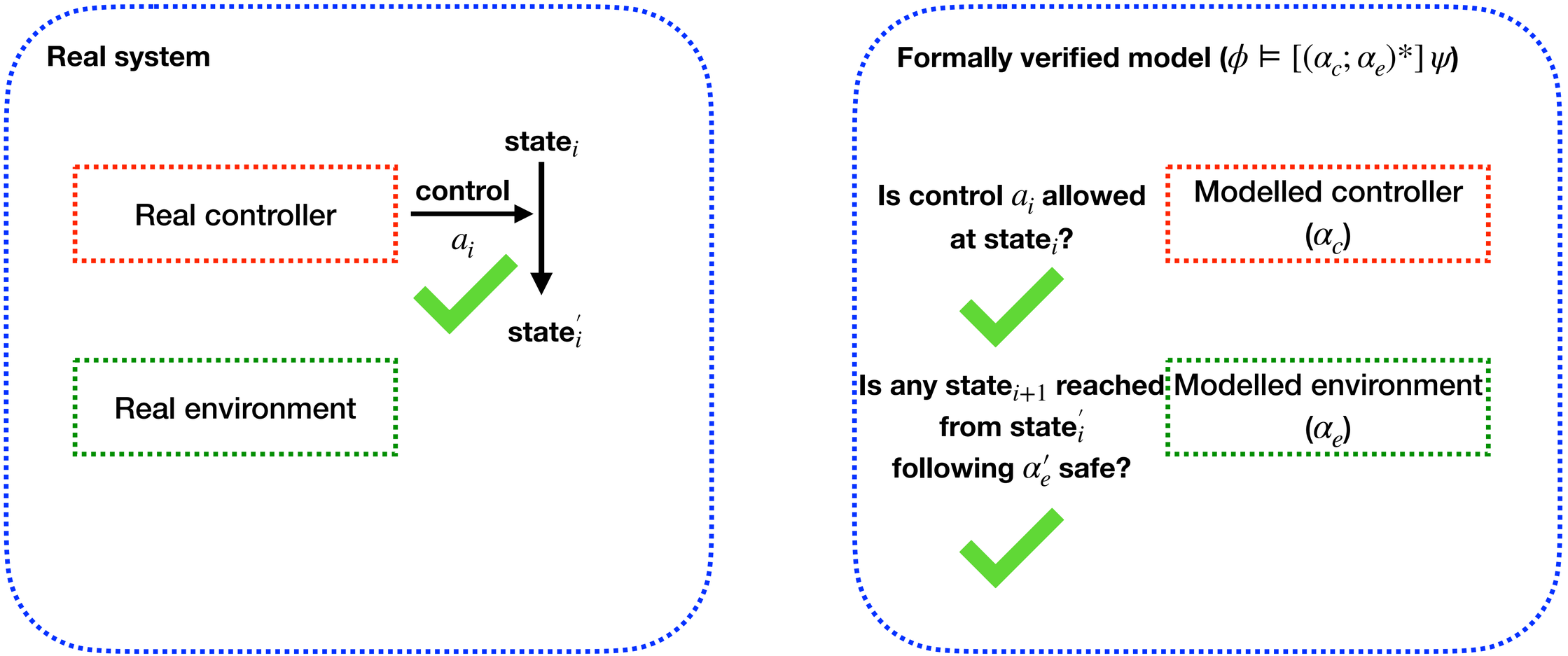


ModelPlex: models as monitors

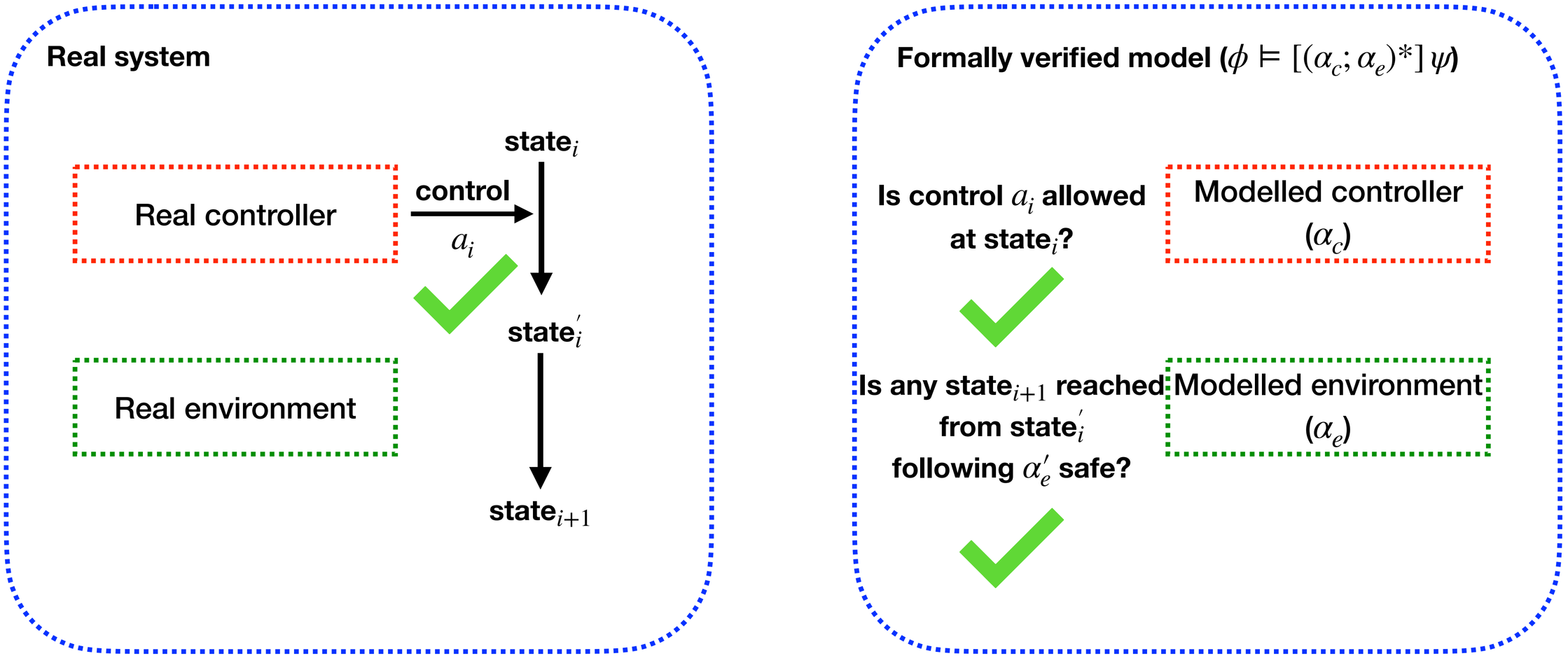




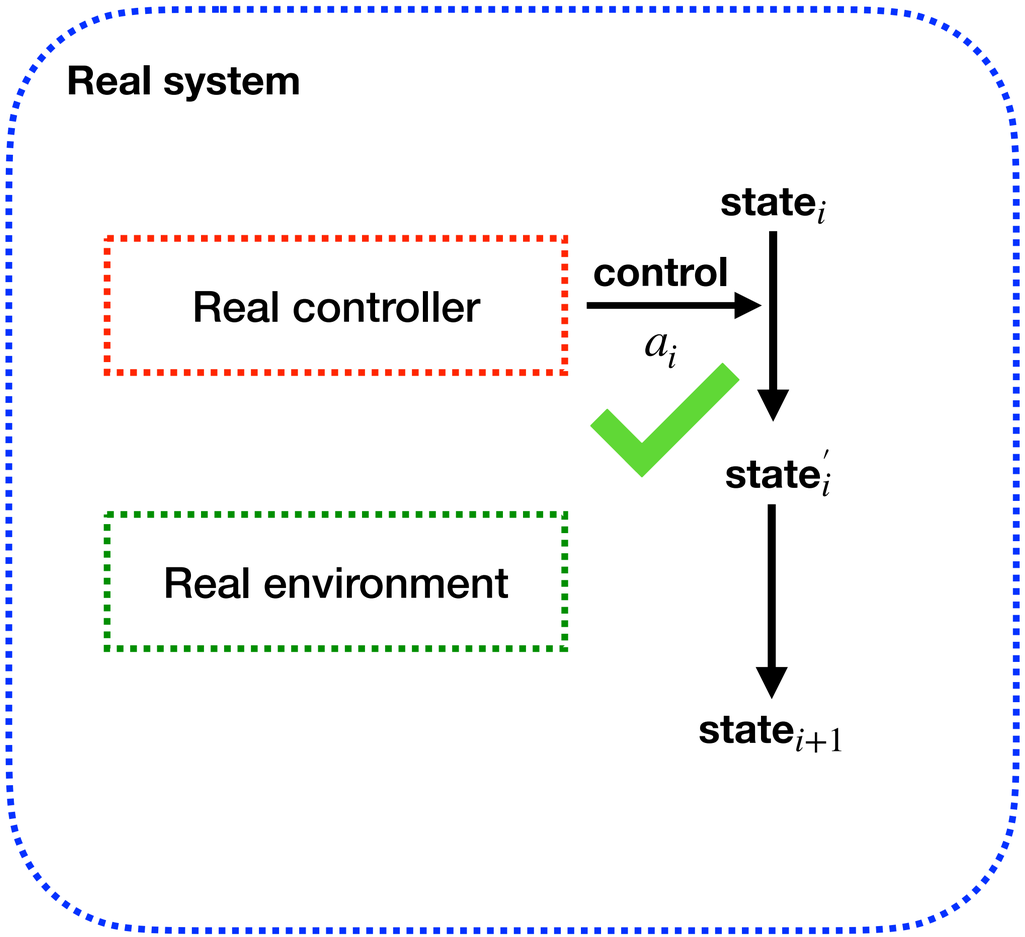
ModelPlex: models as monitors

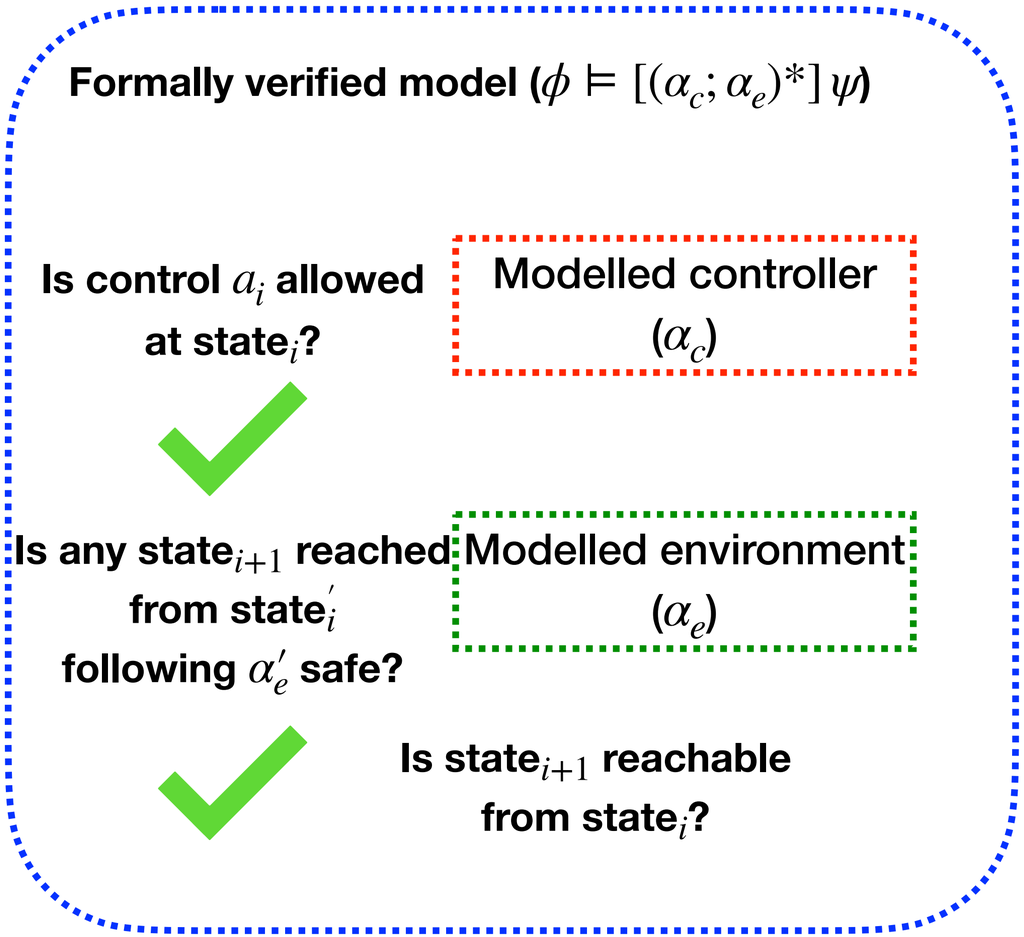


ModelPlex: models as monitors

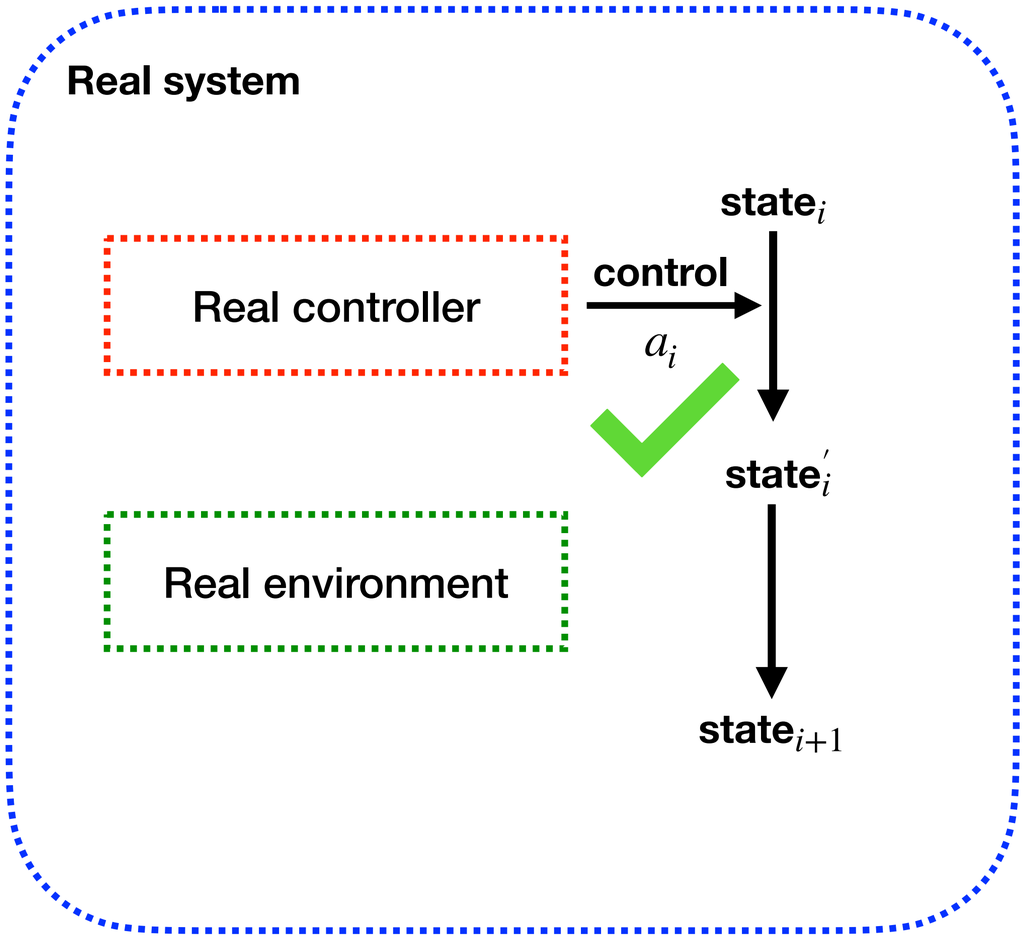


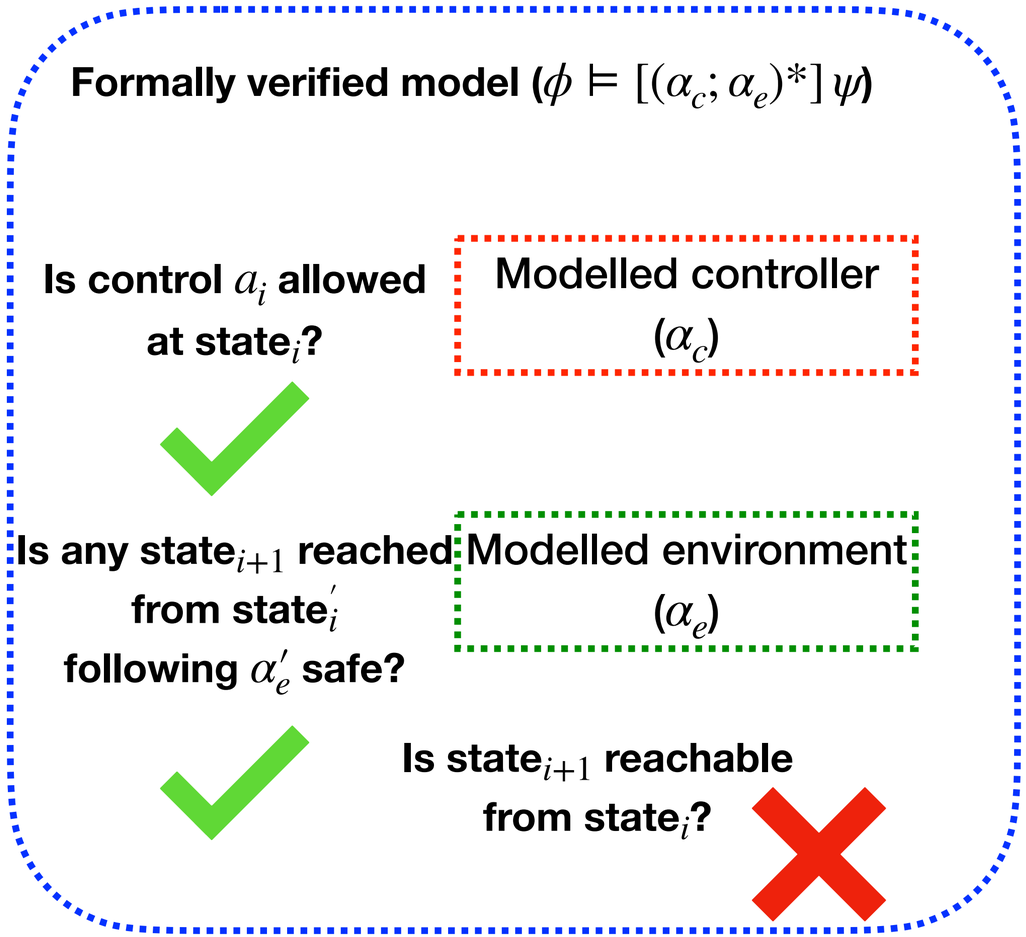
ModelPlex: models as monitors



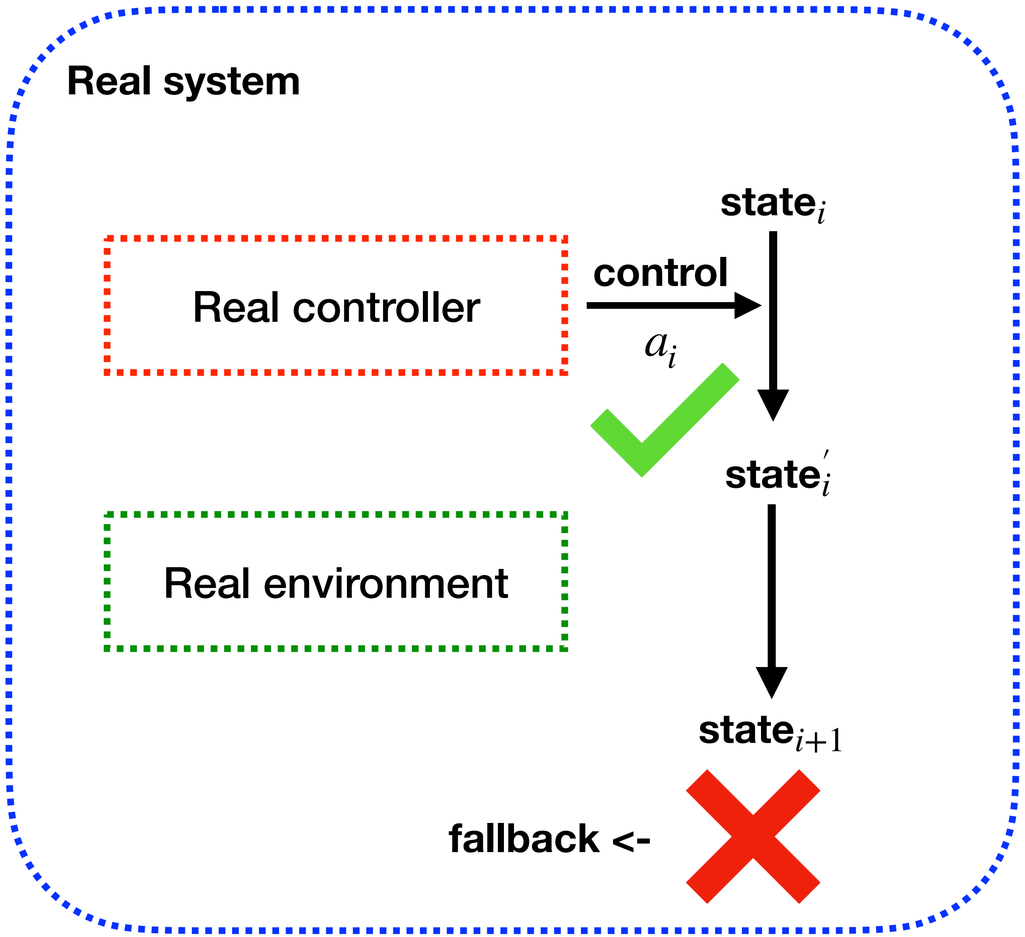


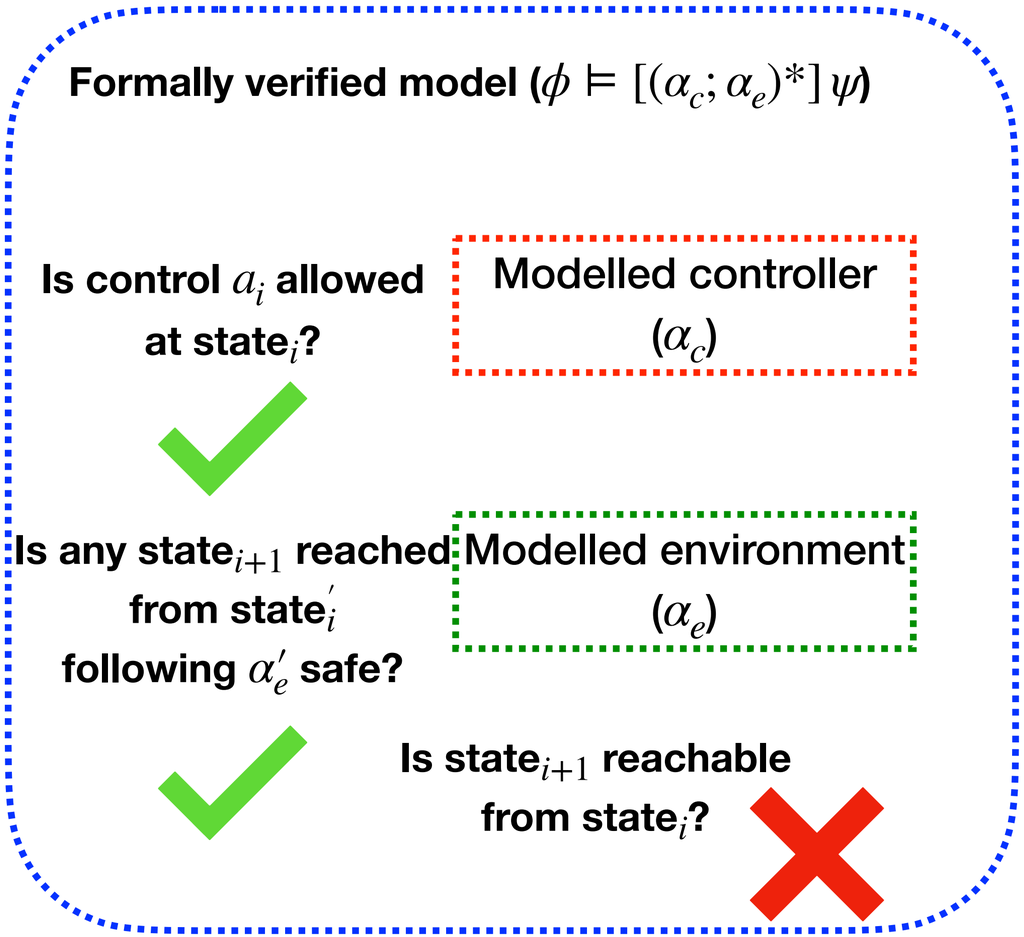
ModelPlex: models as monitors



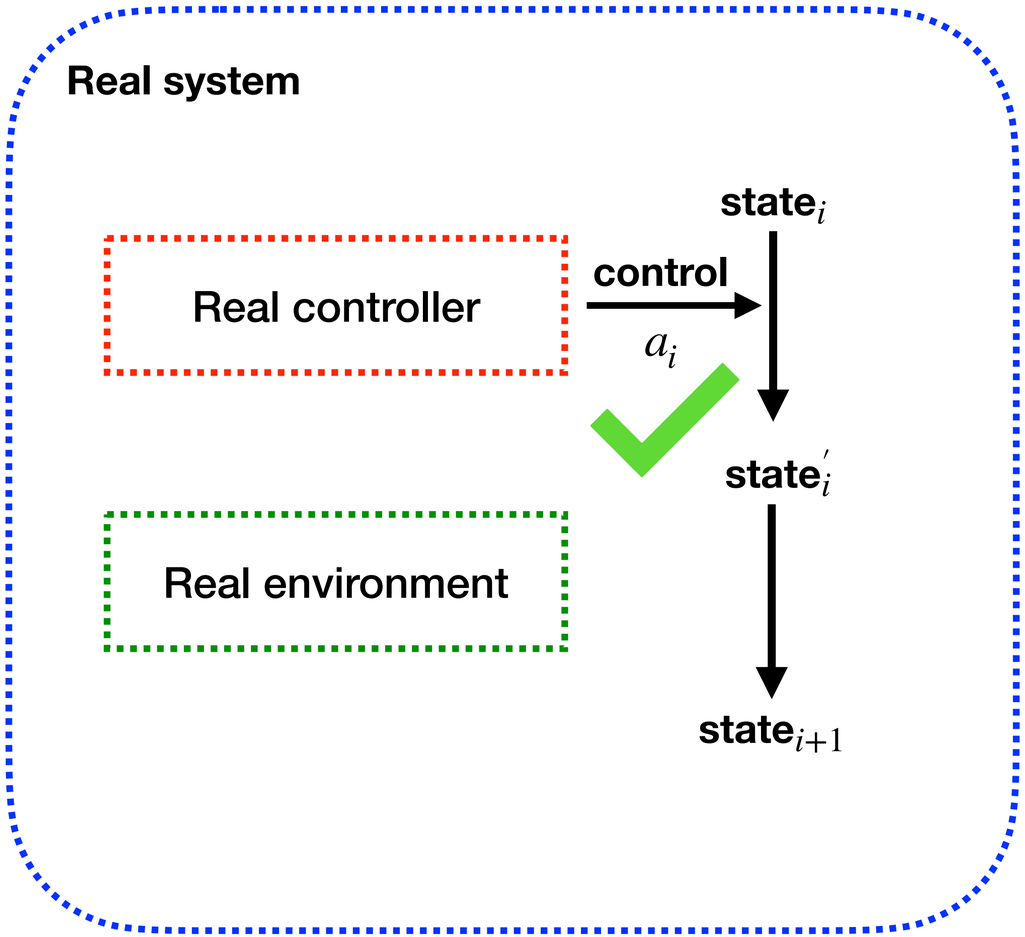


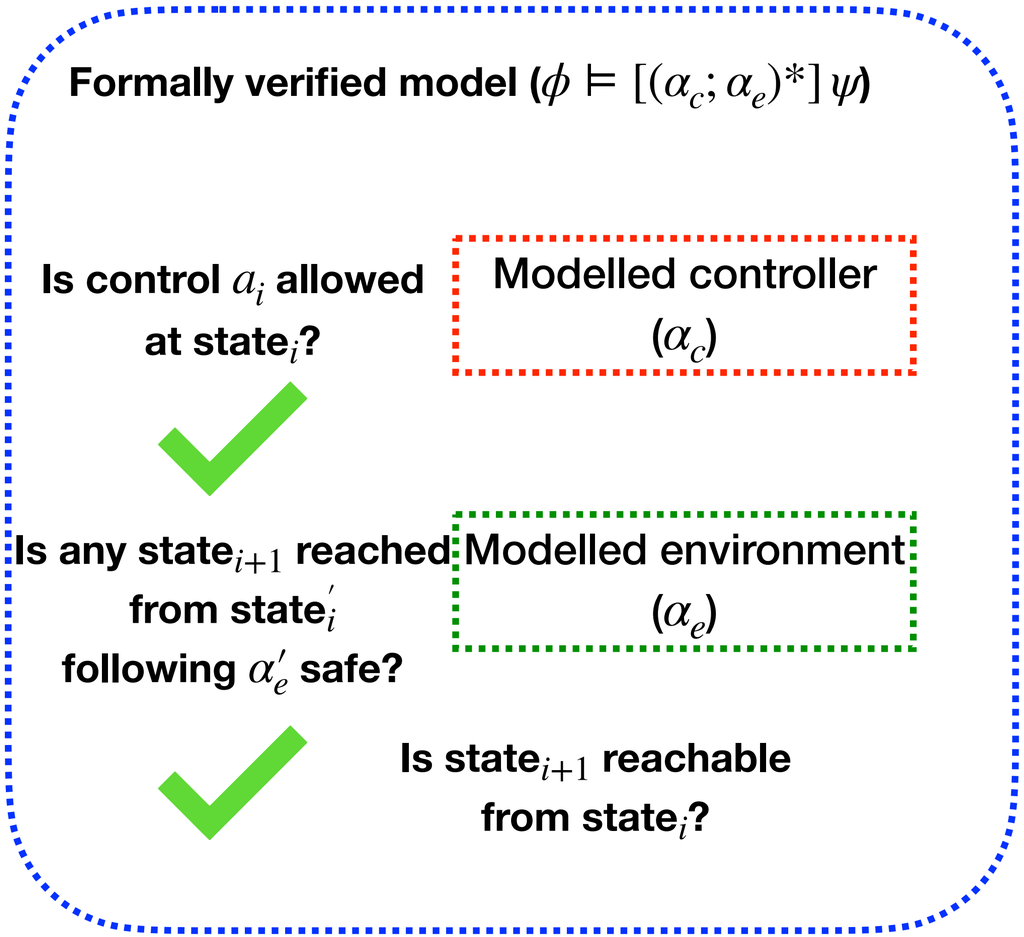
ModelPlex: models as monitors



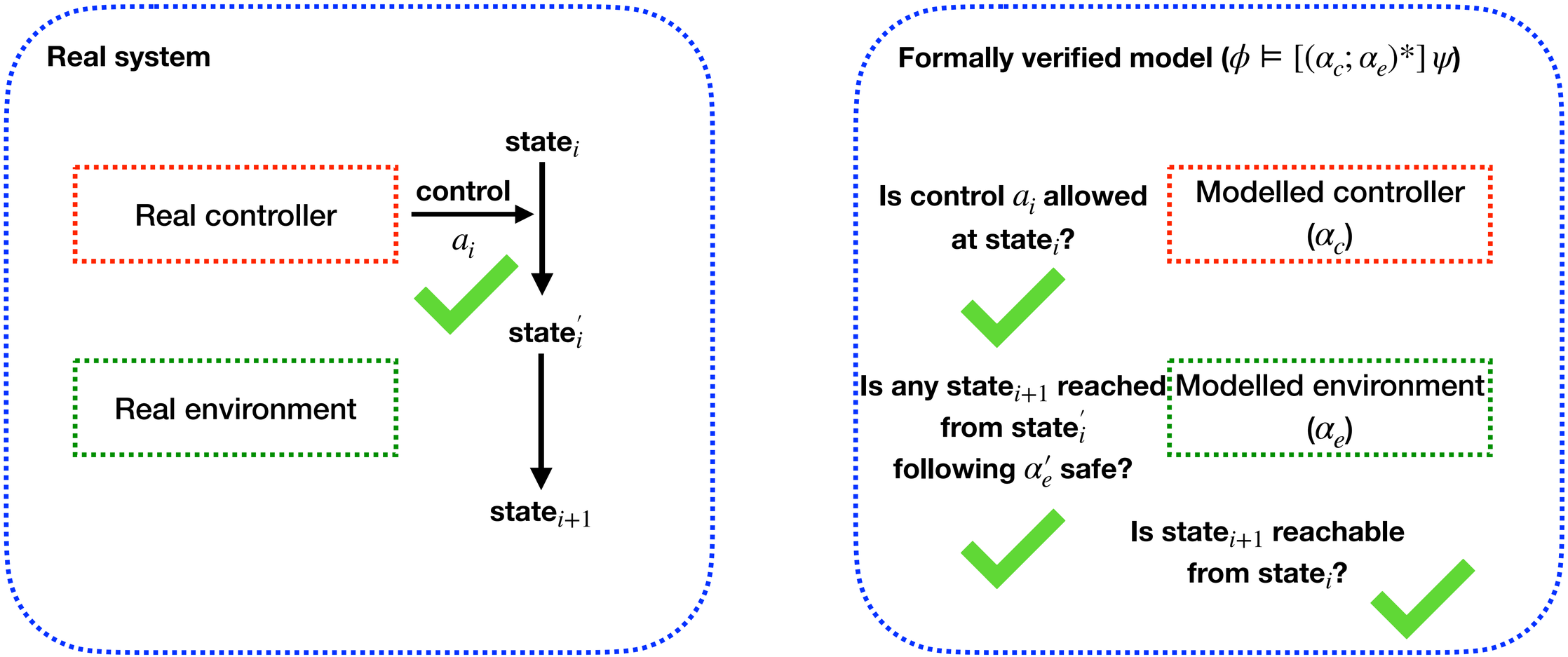


ModelPlex: models as monitors

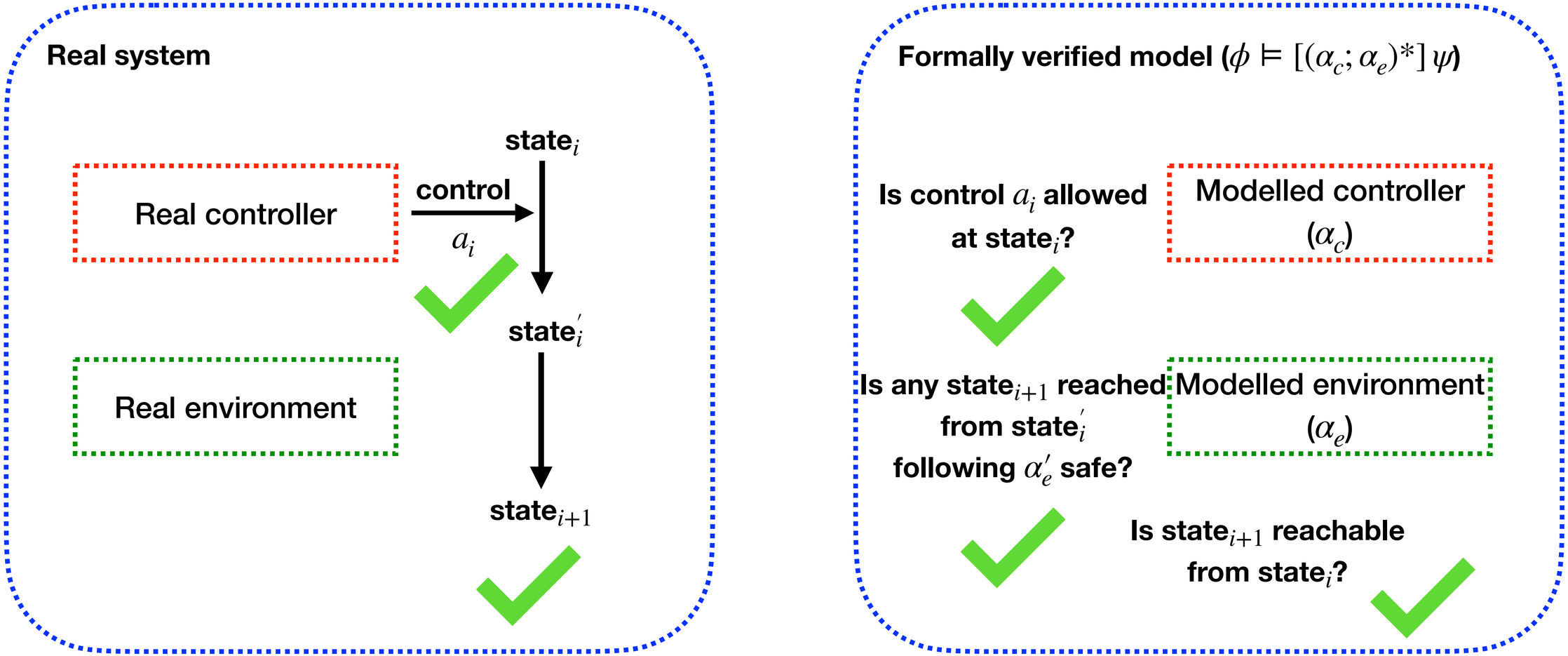




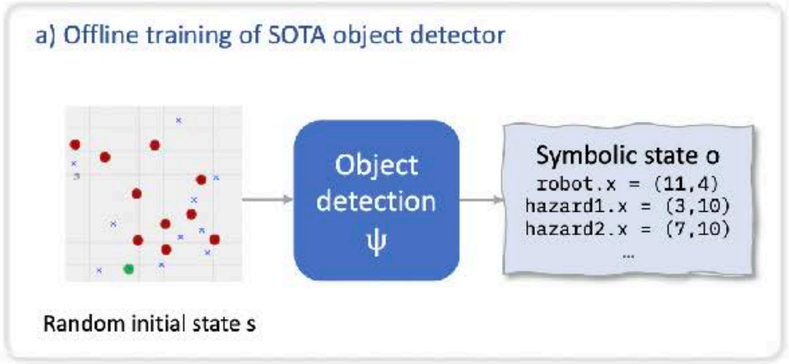
ModelPlex: models as monitors

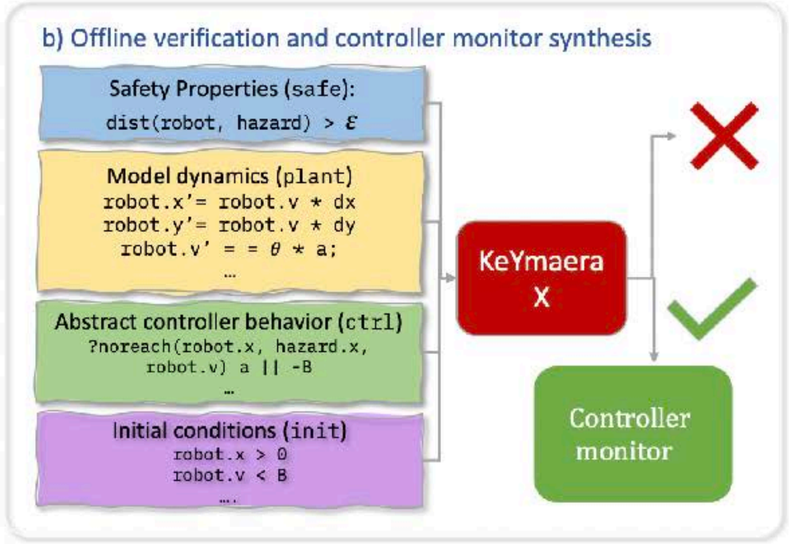


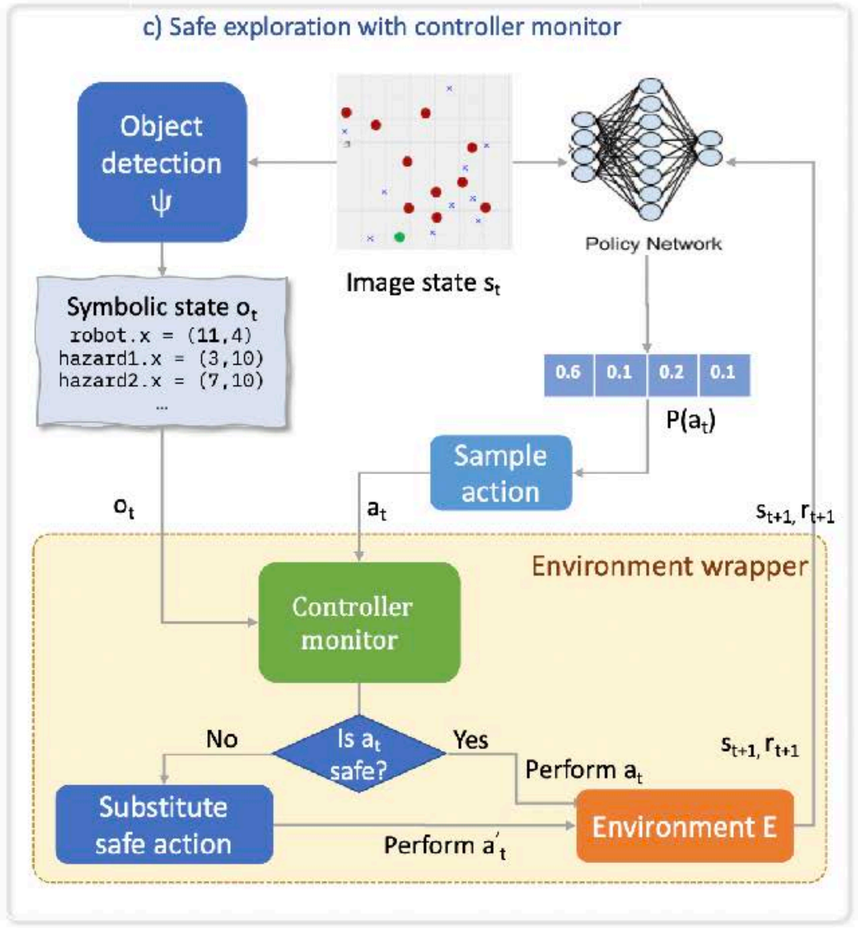
ModelPlex: models as monitors

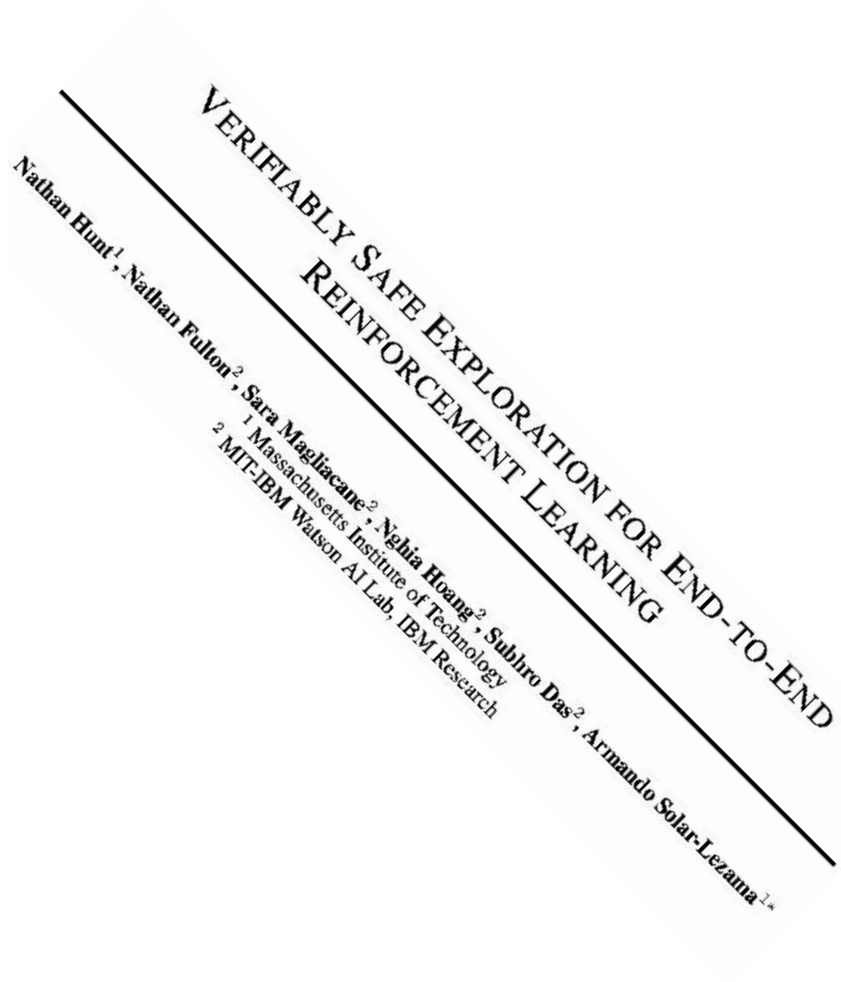


ModelPlex: models as monitors

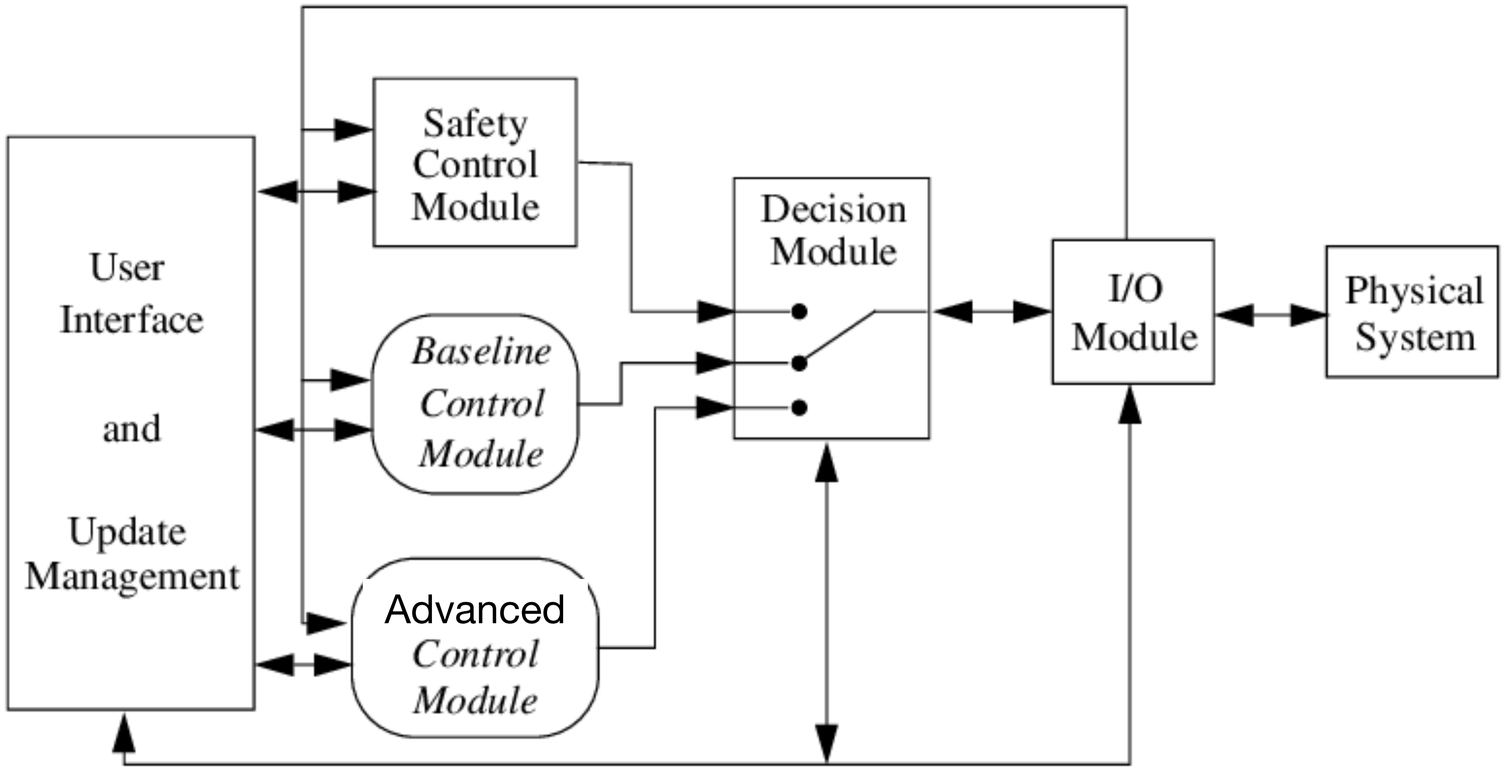




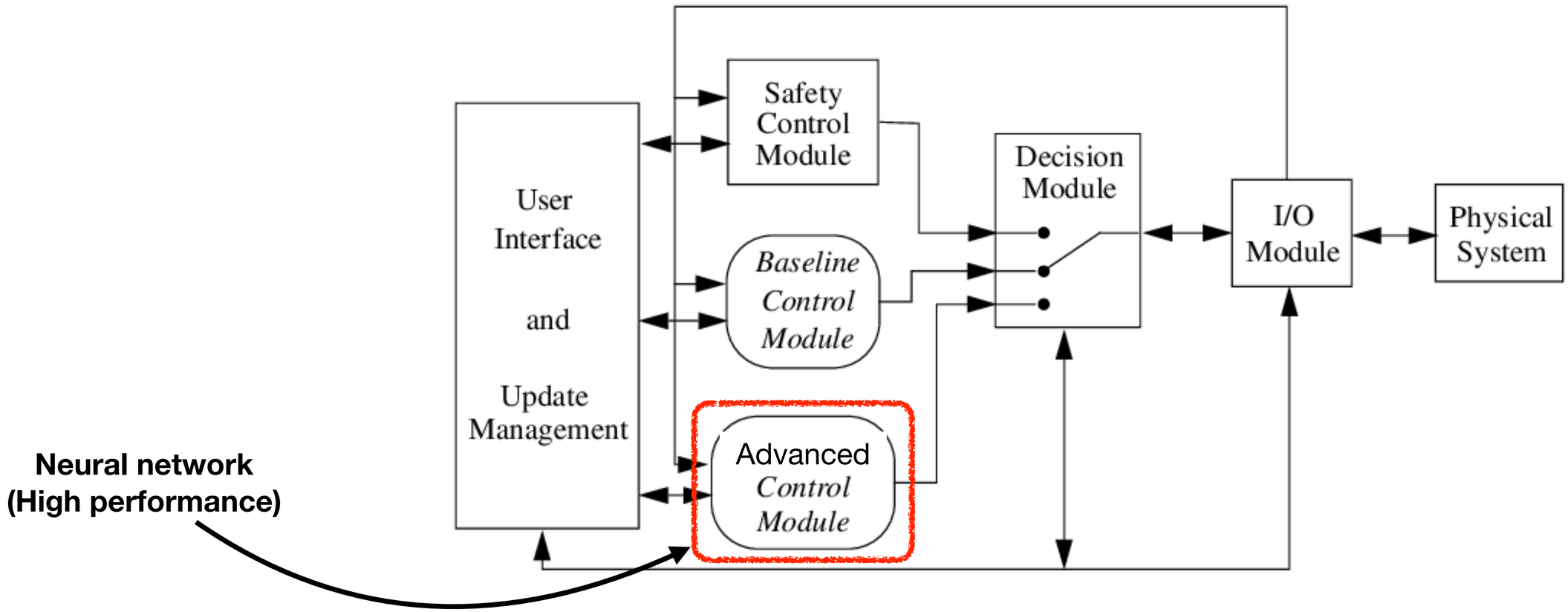




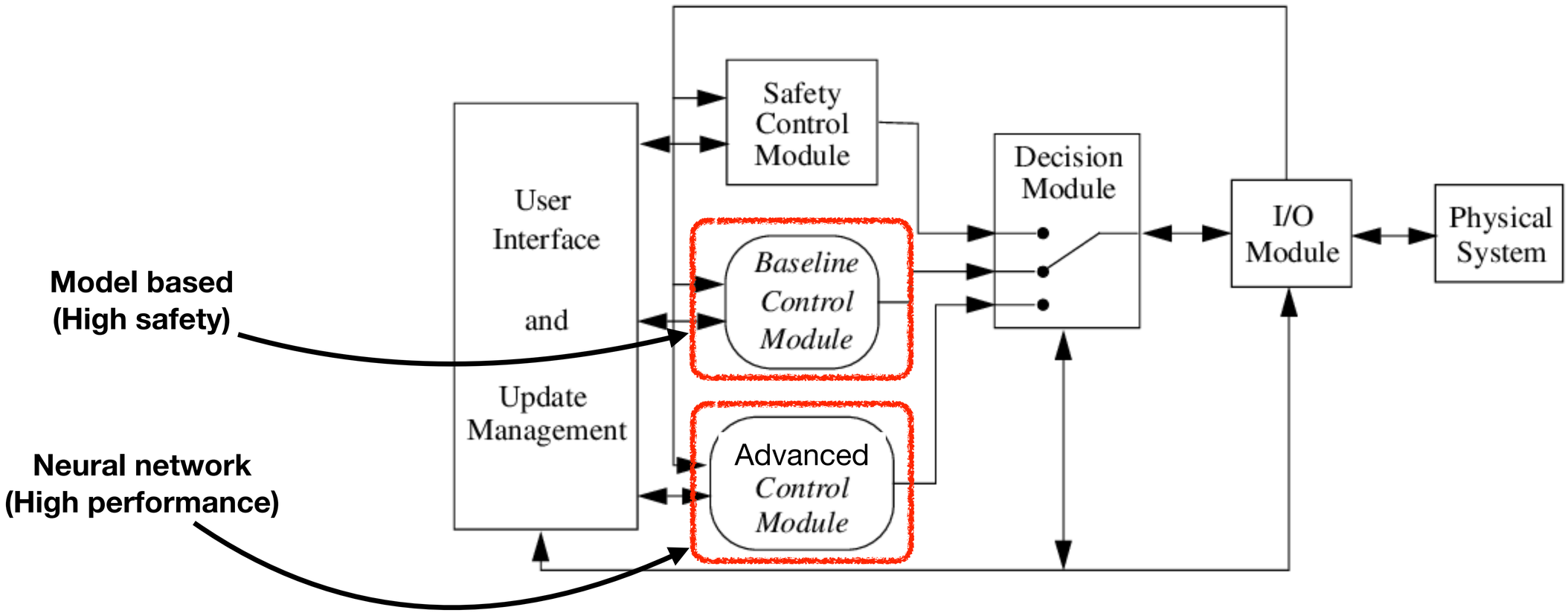
Simplex architecture



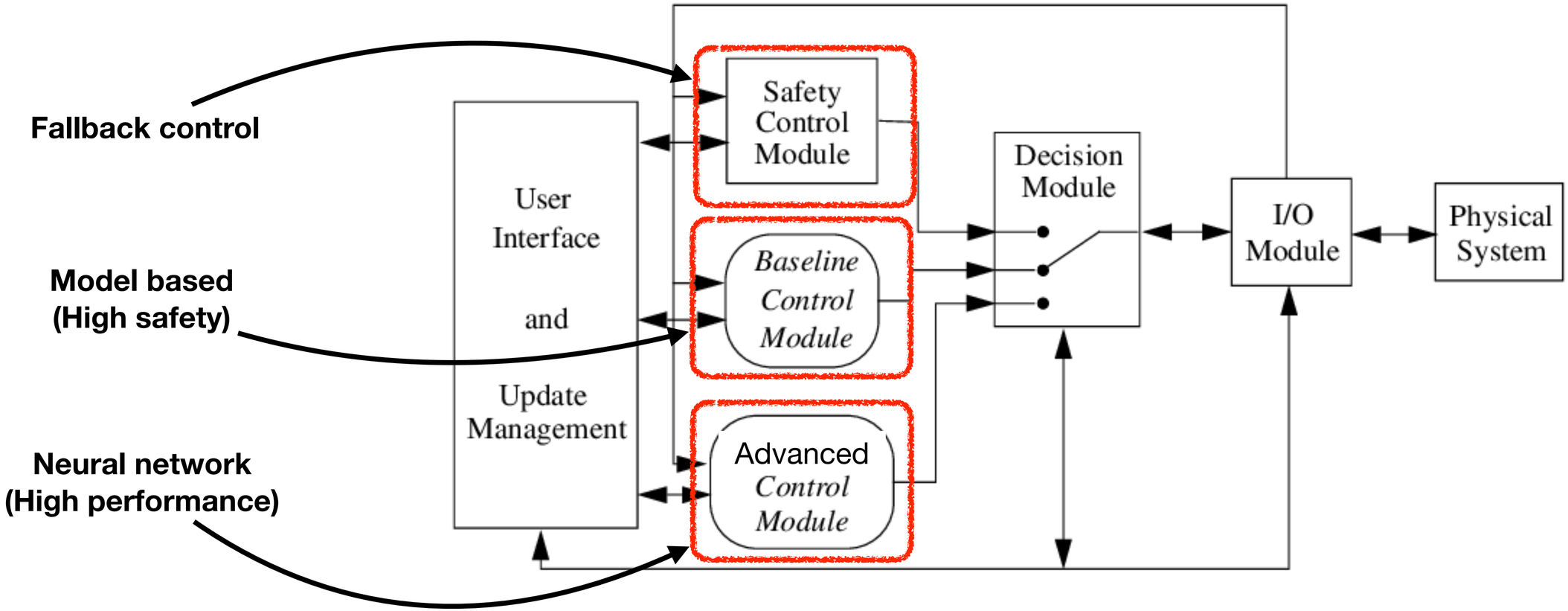
Simplex architecture



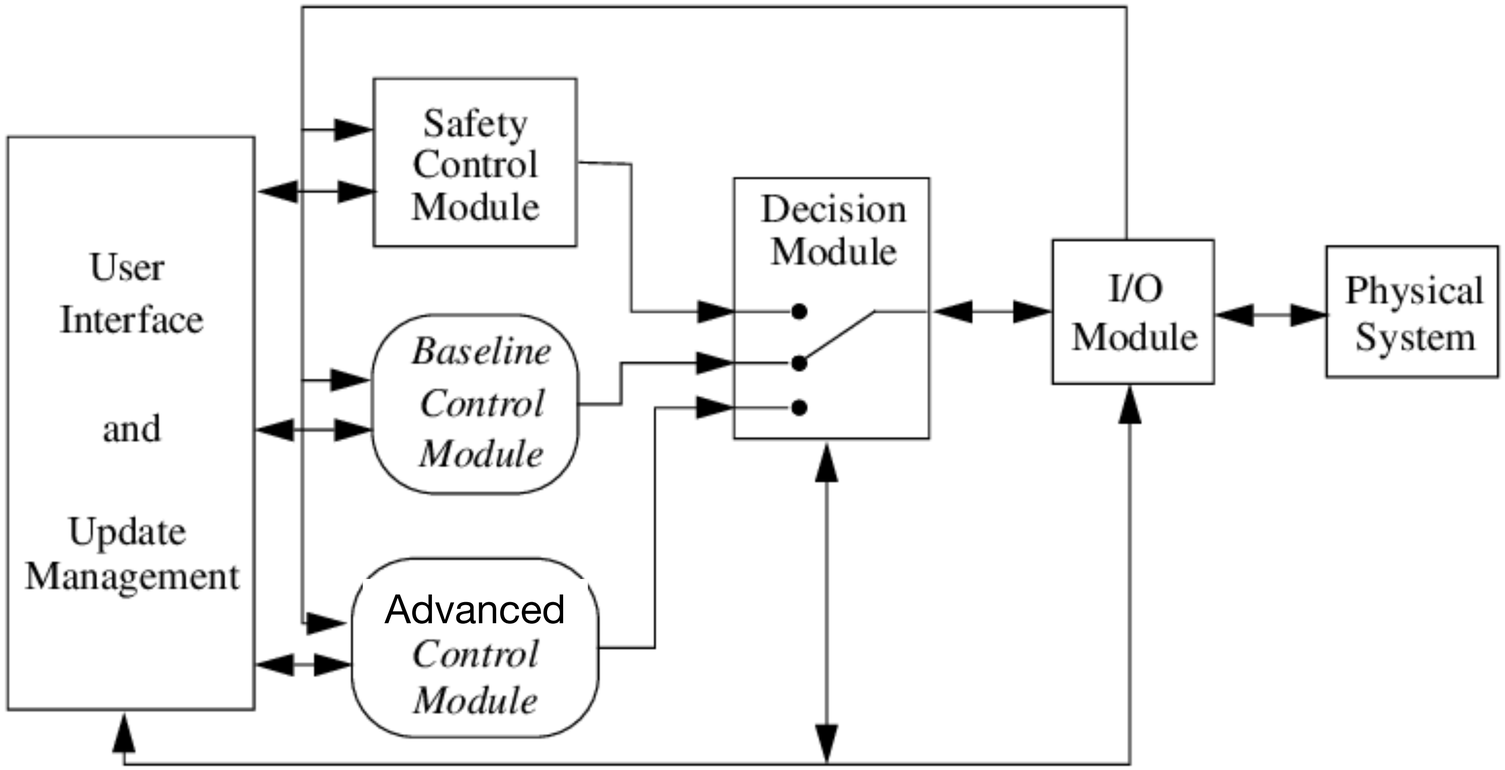
Simplex architecture



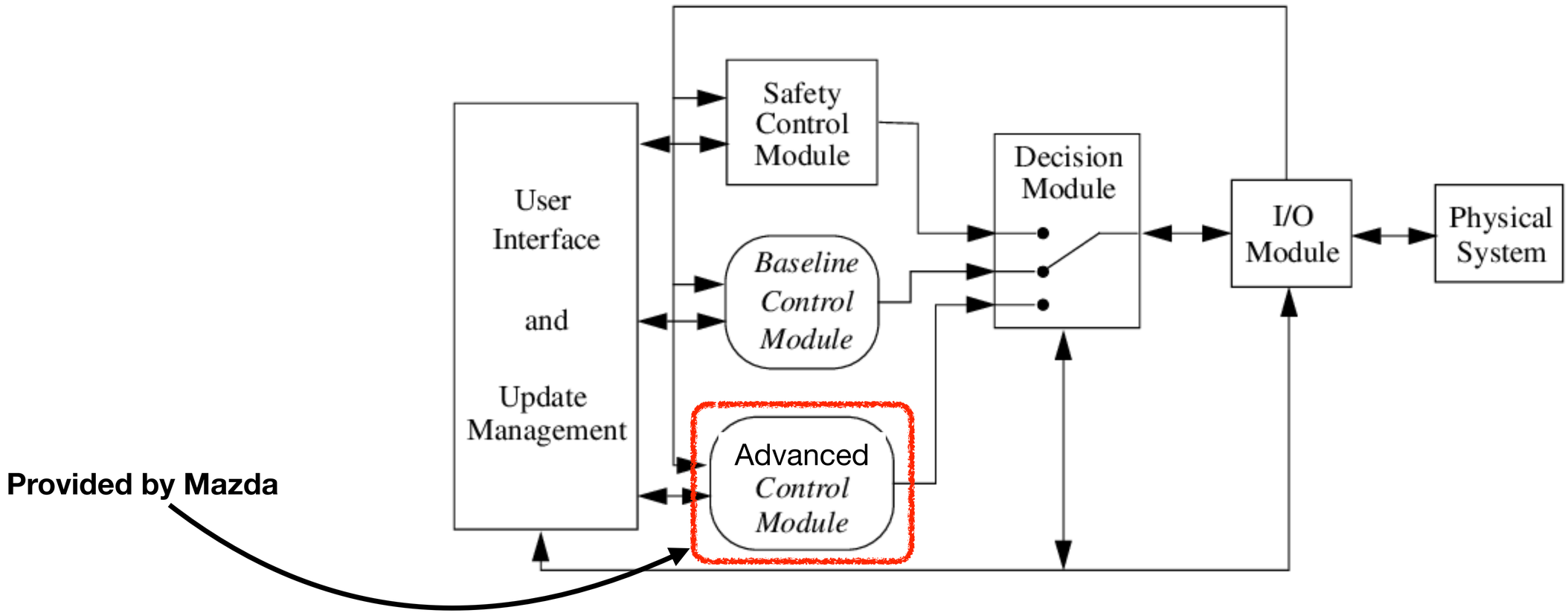
Simplex architecture



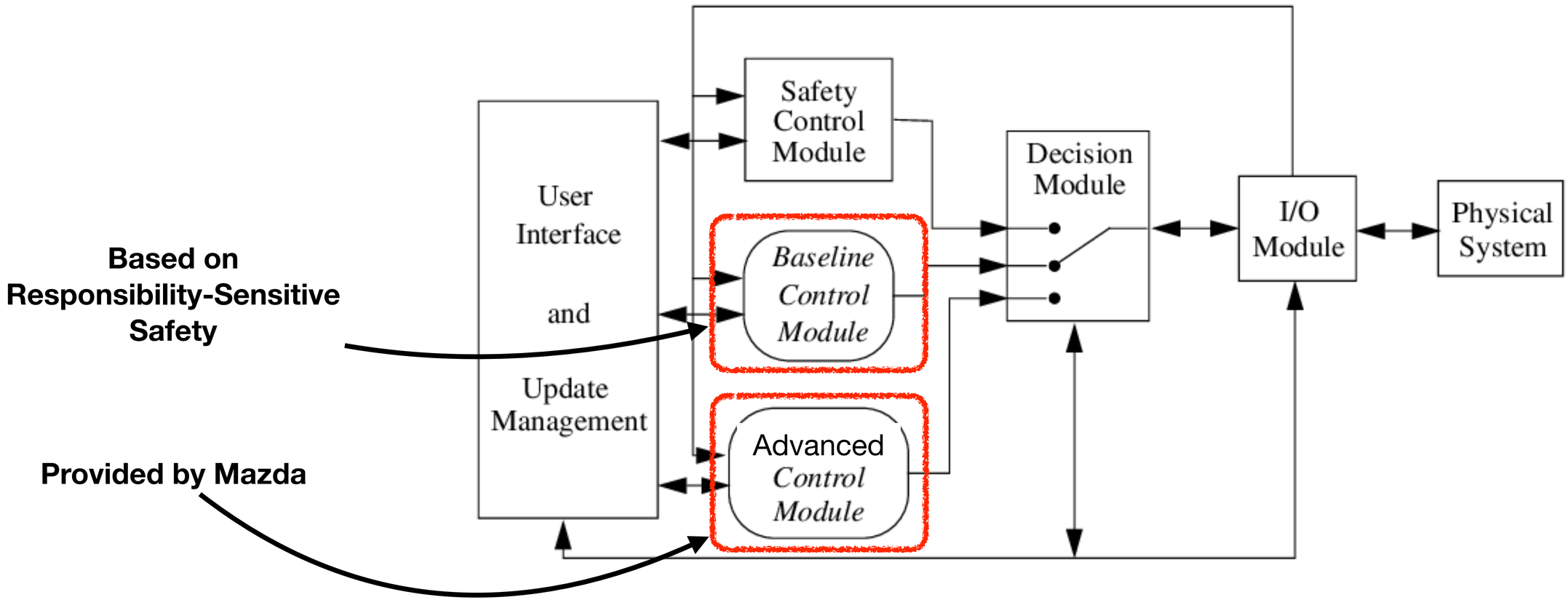
Simplex architecture



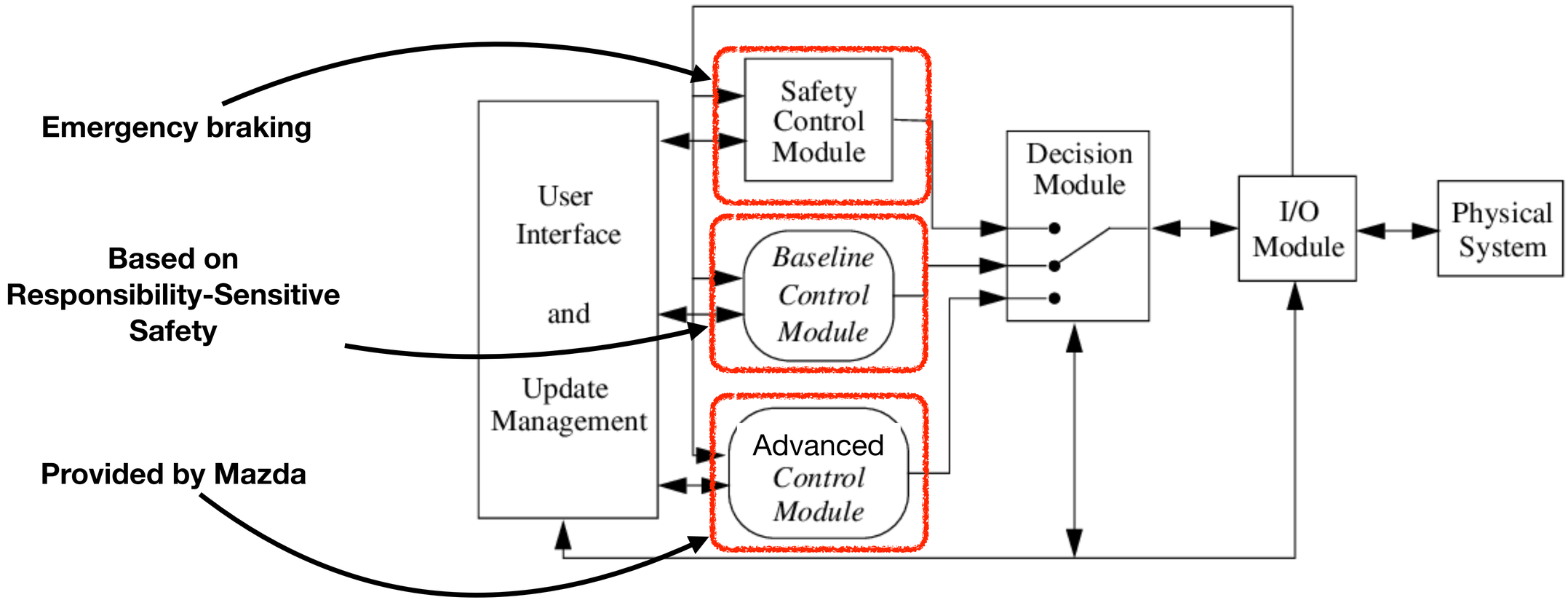
Simplex architecture



Simplex architecture



Simplex architecture



Intel's Mobileye

Example: RSS distance

Example: RSS distance

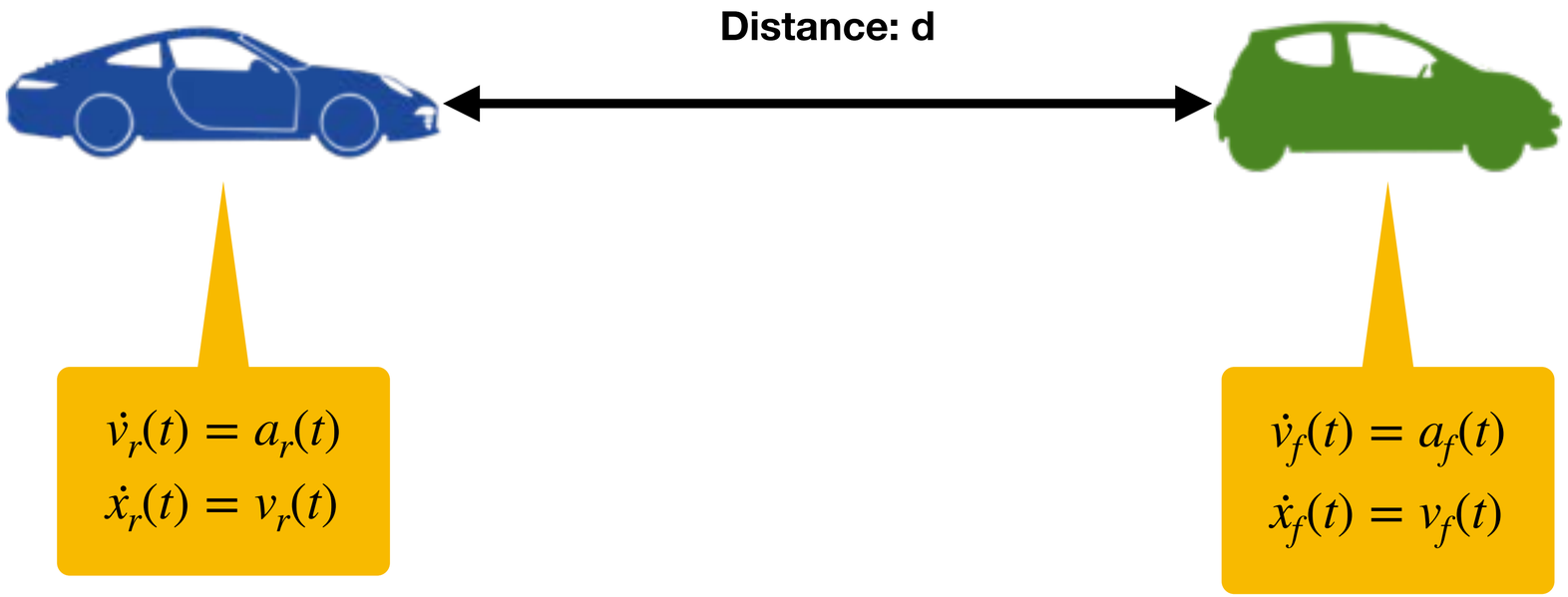
$$\dot{v}_r(t) = a_r(t)$$
$$\dot{x}_r(t) = v_r(t)$$

Example: RSS distance

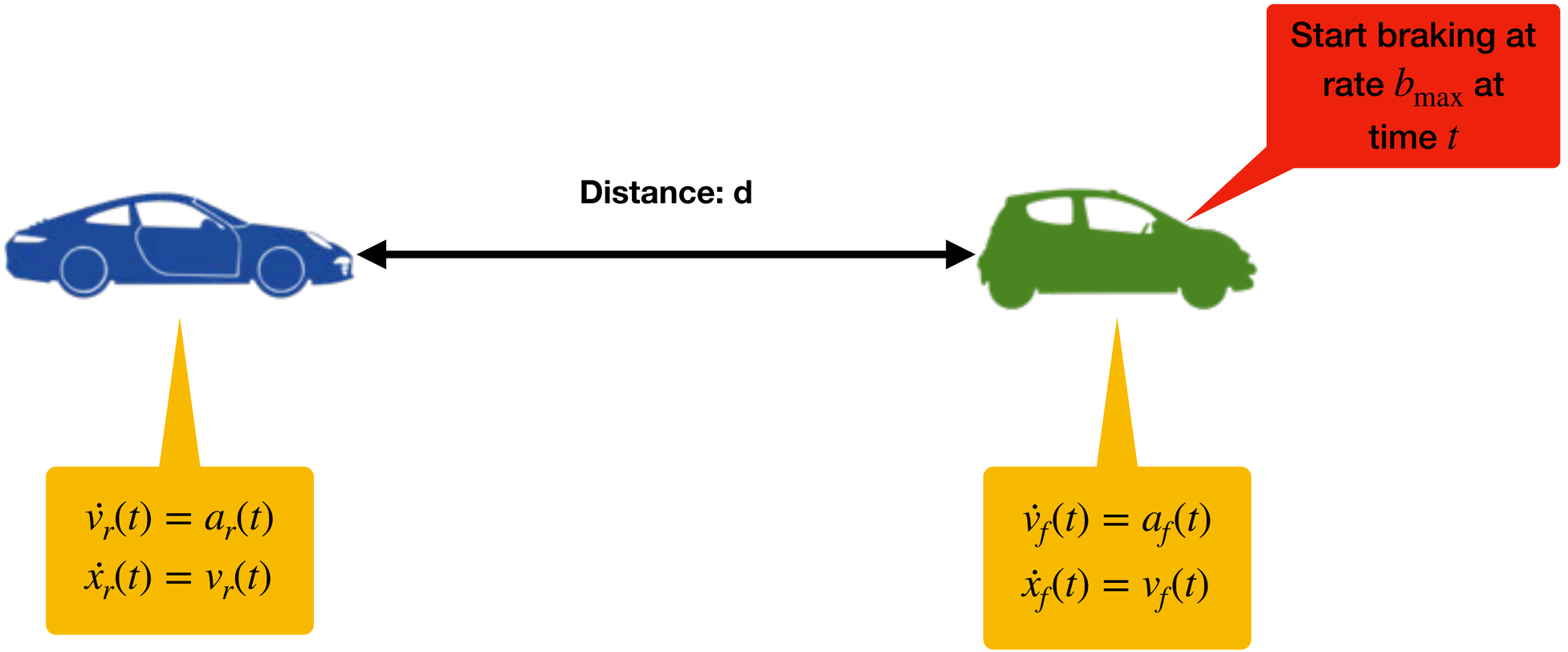
$$\dot{v}_r(t) = a_r(t)$$
$$\dot{x}_r(t) = v_r(t)$$

$$\dot{v}_f(t) = a_f(t)$$
$$\dot{x}_f(t) = v_f(t)$$

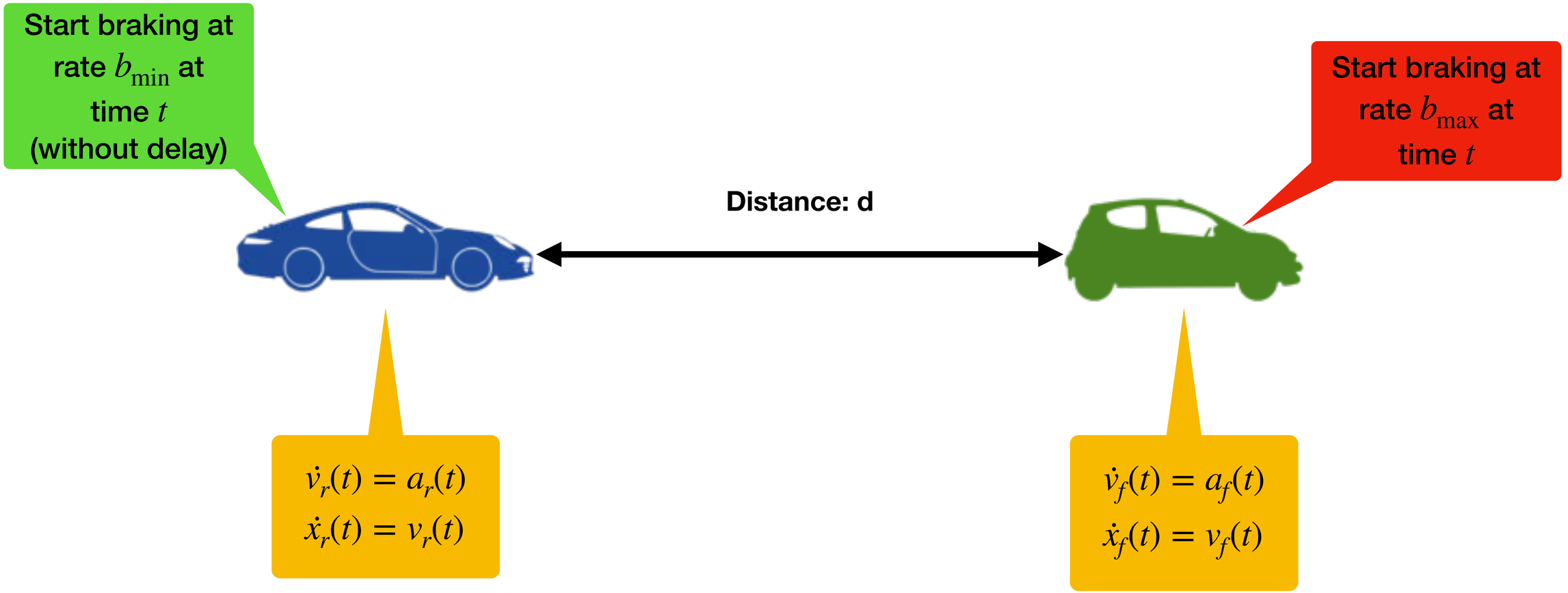
Example: RSS distance



Example: RSS distance



Example: RSS distance



Example: RSS distance

Start braking at rate $b_{\min}$ at time t (without delay)

Start braking at rate $b_{\max}$ at time t

Distance: d

$\dot{v}_r(t) = a_r(t)$

$\dot{x}_r(t) = v_r(t)$

There is a potential crash iff

$$d < \frac{v_r(t)^2}{b_{\min}} - \frac{v_f(t)^2}{b_{\max}}$$

$\dot{v}_f(t) = a_f(t)$

$\dot{x}_f(t) = v_f(t)$

Example: RSS distance

Start braking at rate $b_{\min}$ at time $t + \rho$ (with delay)

Start braking at rate $b_{\max}$ at time t

Distance: d

$$\dot{v}_r(t) = a_r(t)$$
$$\dot{x}_r(t) = v_r(t)$$

$$\dot{v}_f(t) = a_f(t)$$
$$\dot{x}_f(t) = v_f(t)$$

Example: RSS distance

Start braking at rate $b_{\min}$ at time $t + \rho$ (with delay)

Start braking at rate $b_{\max}$ at time t

Distance: d

$$\dot{v}_r(t) = a_r(t)$$
$$\dot{x}_r(t) = v_r(t)$$

There is a crash iff

$$d < \frac{(v_r(t) + a_m a x \rho)^2}{b_{\min}} - \frac{v_f(t)^2}{b_{\max}} + a_{max}\rho$$

$$\dot{v}_f(t) = a_f(t)$$
$$\dot{x}_f(t) = v_f(t)$$

RSS distance, statement

Assuming that the front car is braking has speed v_r and that its maximum braking rate is $b_{\max}$.
Assuming that the rear car has response time ρ and speed v_f, and that the maximum comfortable braking rate is $b_{\min}$ and maximum acceleration rate is $a_{\max}$.
Then the following is a controller for the rear car to comfortably response to any braking:

- If the rear car detect that the distance is $< \frac{(v_r(t) + a_m a x \rho)^2}{b_{\min}} - \frac{v_f(t)^2}{b_{\max}} + a_{max}\rho$ then it brakes at rate $b_{\min}$.
- Otherwise, do whatever you want.

Furthermore, this controller is optimal.

RSS distance, statement

Assuming that the front car is braking has speed v_r and that its maximum braking rate is $b_{\max}$.
Assuming that the rear car has response time ρ and speed v_f, and that the maximum comfortable braking rate is $b_{\min}$ and maximum acceleration rate is $a_{\max}$.
Then the following is a controller for the rear car to comfortably response to any braking:

- If the rear car detect that the distance is $< \frac{(v_r(t) + a_m a x \rho)^2}{b_{\min}} - \frac{v_f(t)^2}{b_{\max}} + a_{max}\rho$ then it brakes at rate $b_{\min}$.
- Otherwise, do whatever you want.

Furthermore, this controller is optimal.

This statement has been formalised in KeYmaera X!

On-going work I

- RSS's definition of "comfortable" is not reasonable
- We are formalising a similar statement for a more reasonable one, which requires:
  - A new RSS distance and
  - A new proper response.
- On-going work with Rose Bohrer (originally from the group developing KeYmaera X)

On-going work II

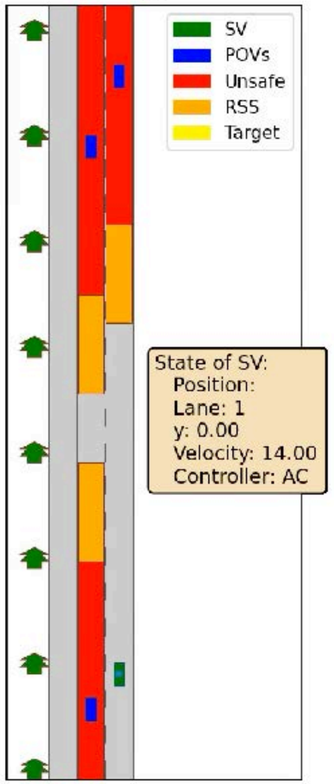
- Write a fully formalised baseline controller using RSS principles

- Basic idea: the goal will be split in subtask for which formalised RSS-like rules describe proper answers

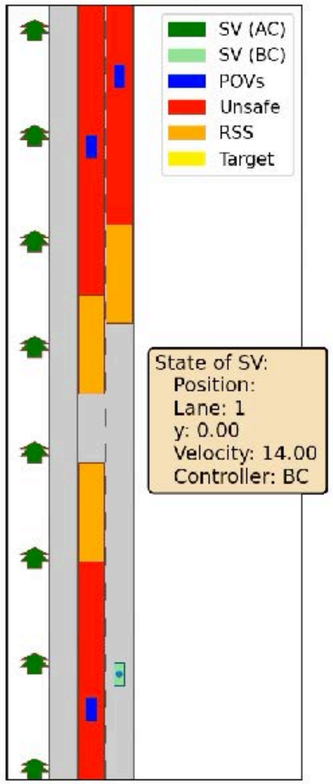
- On-going work with Hasuo-sensei's group:
  - A proof-of-concept about an emergency stop scenario
  - Rules formalised in dL-like logic on paper, partially formalised in KeYmaera X
  - Implemented our own proof checker and partially formalised there
  - A paper (published), a patent (??), press releases (on-going)
  - Future: make the process subscenarios -> rule design -> proof fully automatised
  - Future: apply our method on an automatic bus line in Odaiba

Illustration I: a safer stop

AC



AC+RSS



AC+*RSS*^{GA}

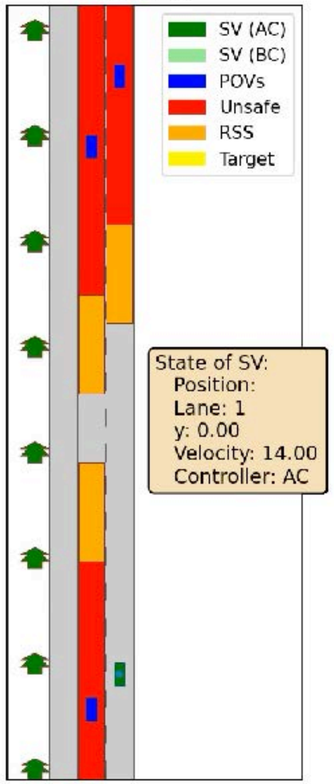
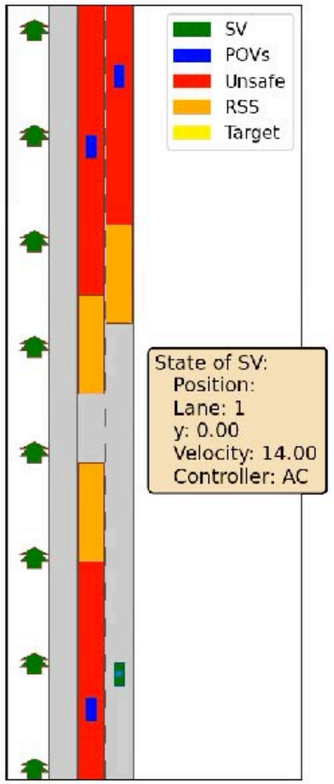
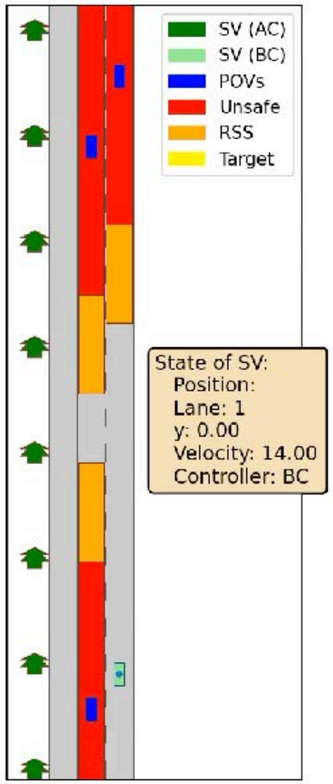


Illustration I: a safer stop

AC



AC+RSS



AC+*RSS*^{GA}

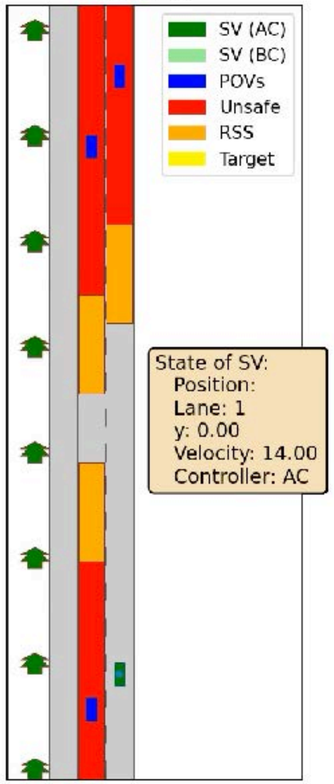
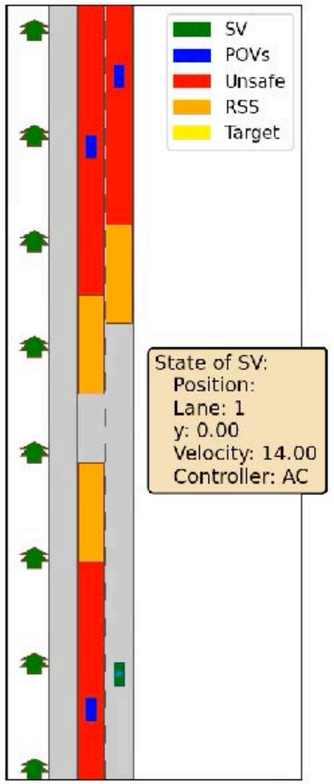
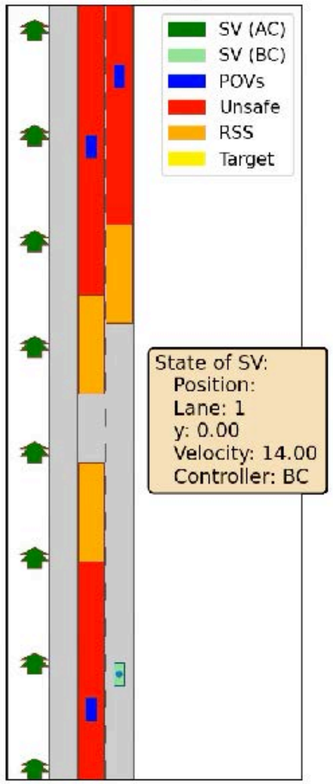


Illustration I: a safer stop

AC



AC+RSS



AC+*RSS*^{GA}

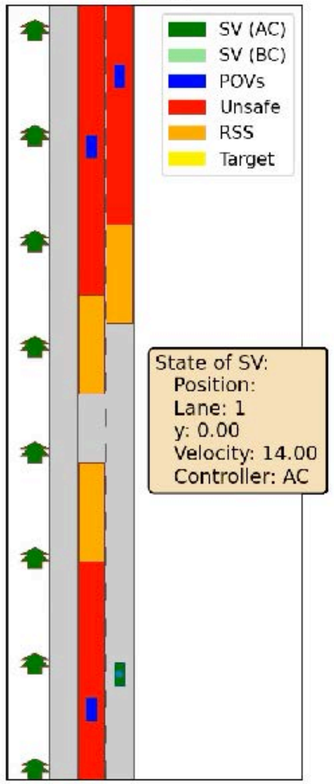
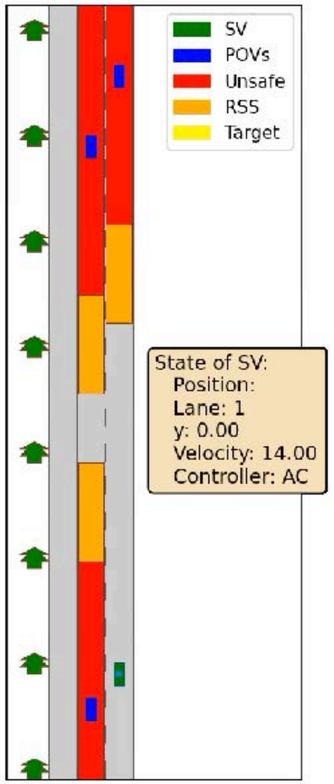
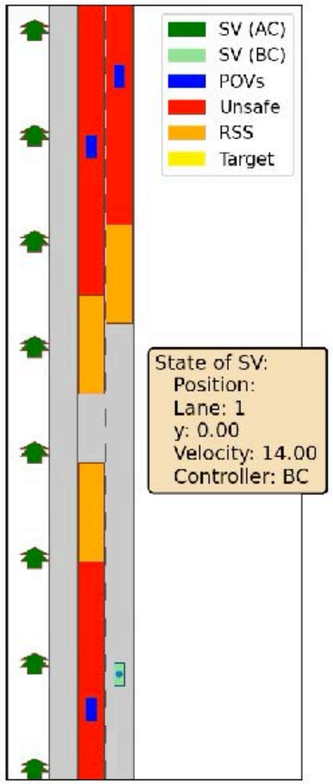


Illustration I: a safer stop

AC



AC+RSS



AC+*RSS*^{GA}

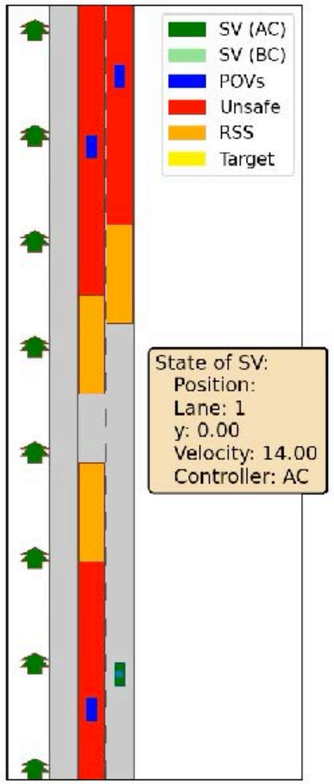
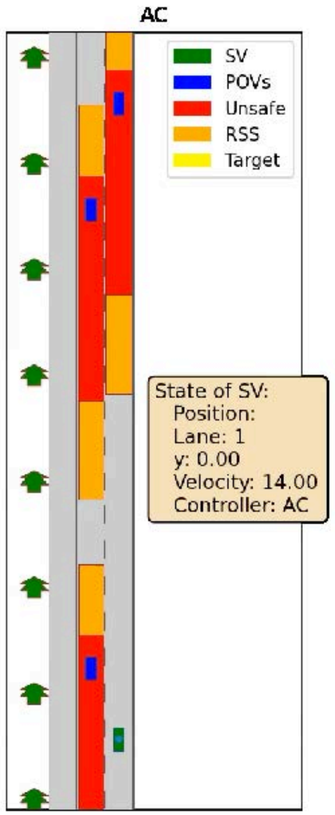
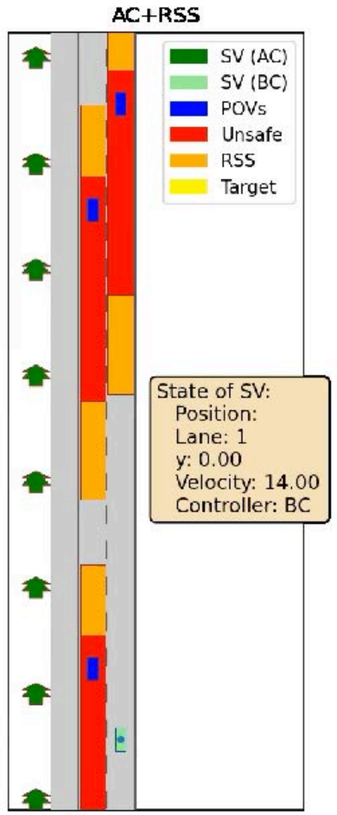


Illustration II: a bold but safe stop





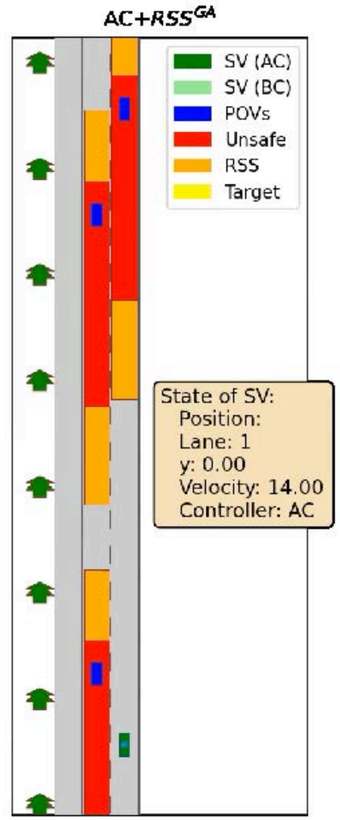
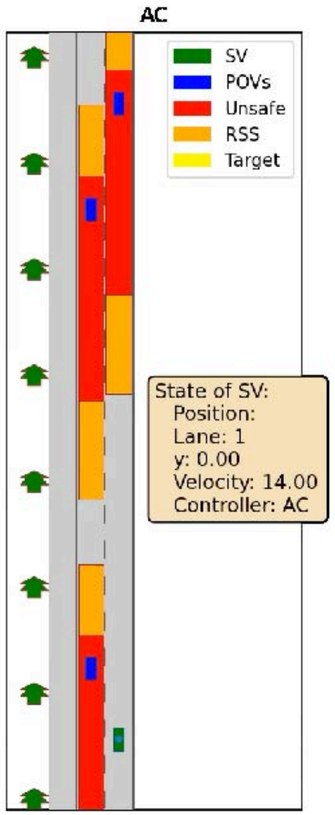
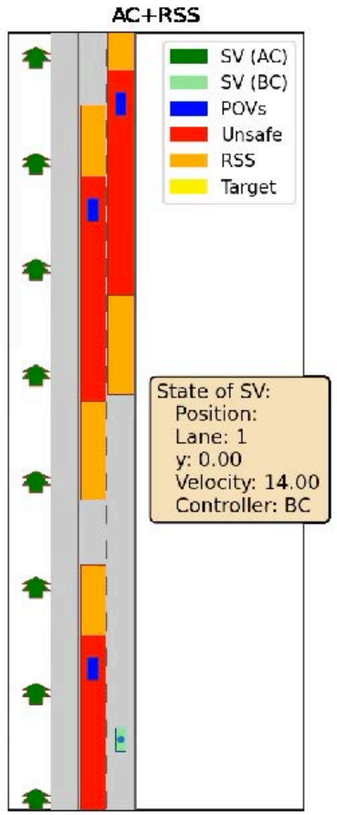


Illustration II: a bold but safe stop





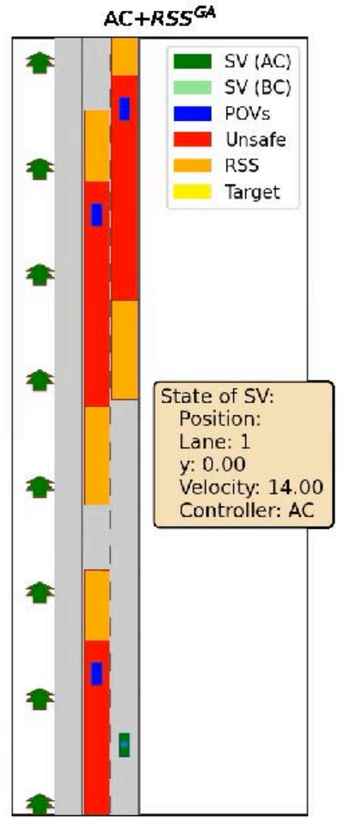
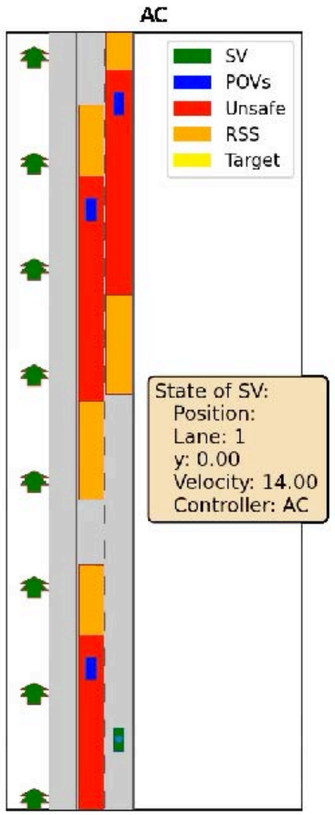
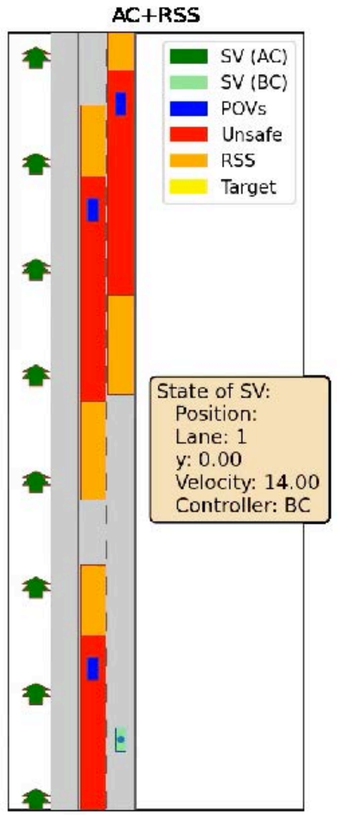


Illustration II: a bold but safe stop





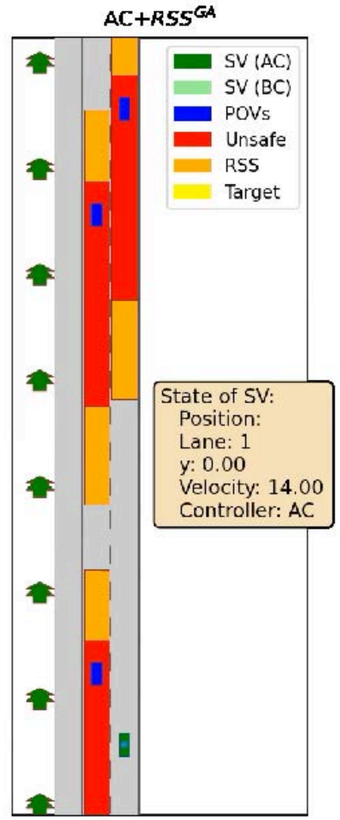


Illustration II: a bold but safe stop