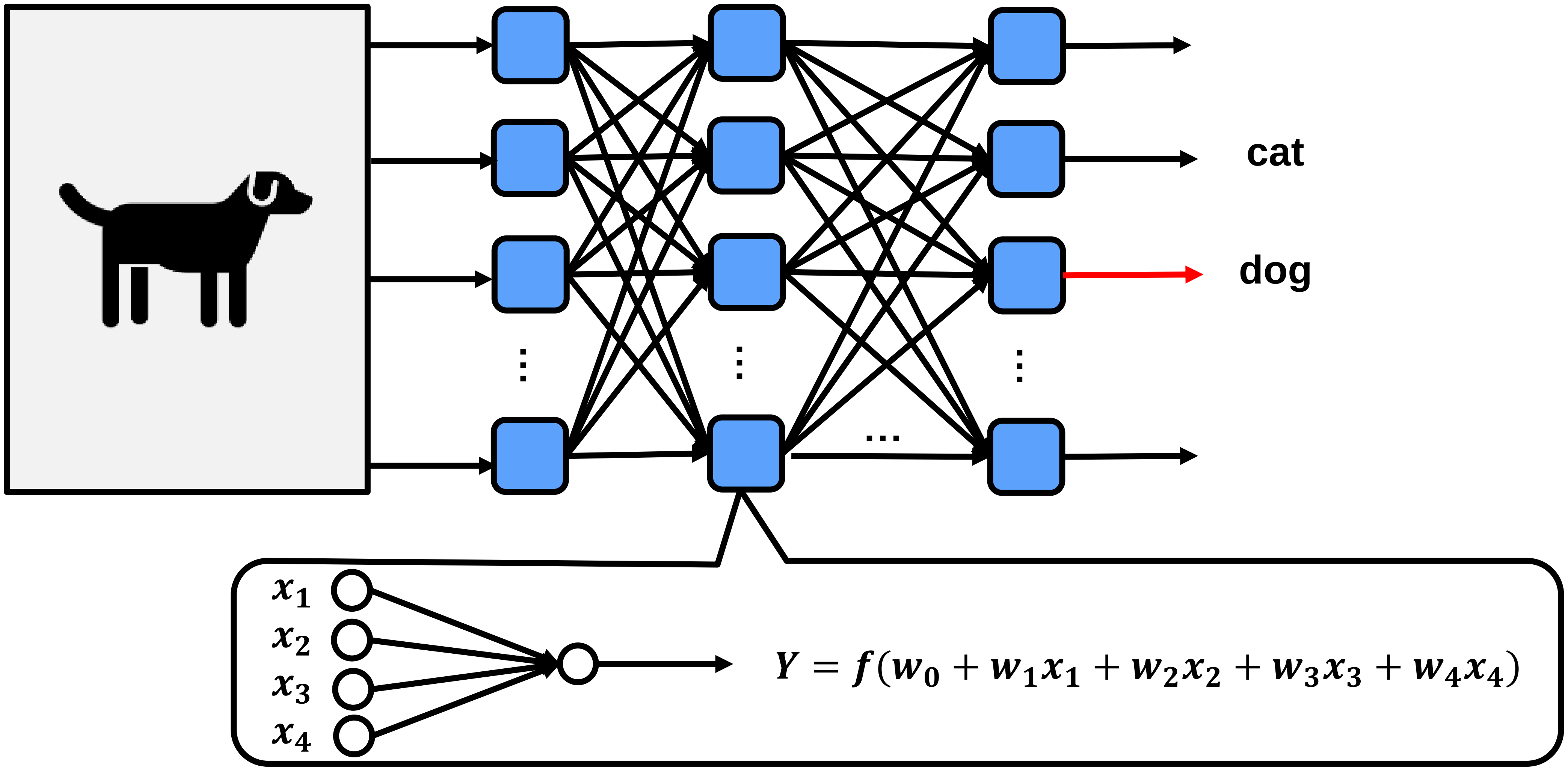
Runtime-Assurance for AI

Dionisio de Niz, Ph.D.

Principal Researcher & Technical Director
Assuring Cyber-Physical Systems

V&V of AI/ML: Machine Learning Simplified View

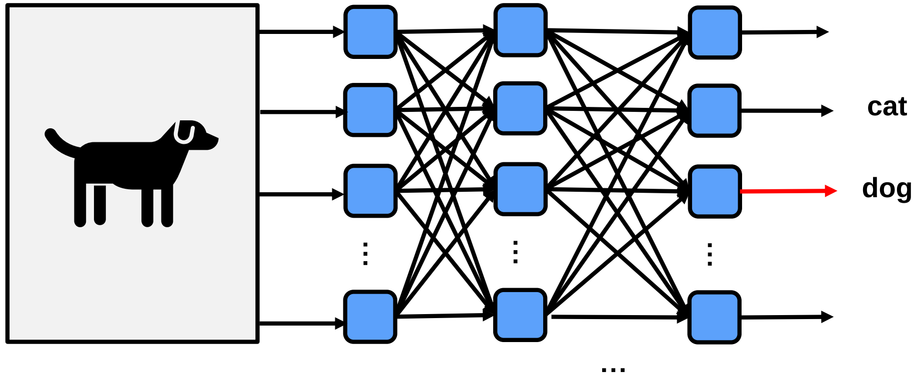


$$Y = f(w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4)$$

V&V Machine Learning Challenges

Computation not based on application logic (Trained ML)

- Neurons triggered by combined weighted input
- Weights “trained” in learning face



Computation changes at runtime (Evolving ML)

- As neural network learns it changes computation
- Behavior / Computation not known at design time

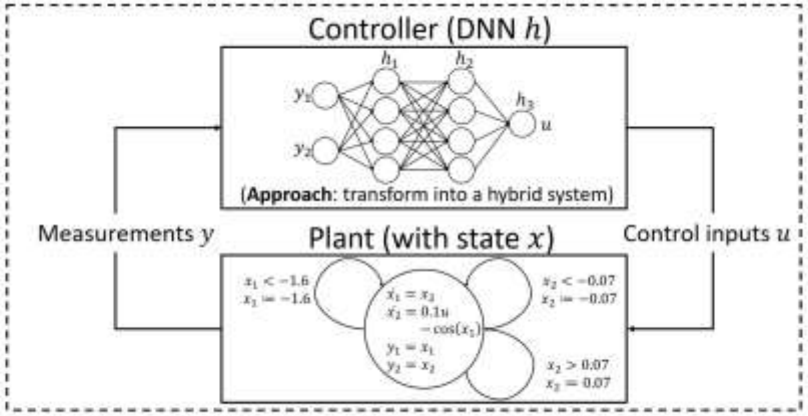
V&V Approaches for Trained ML

Reluplex[1]

- Applied to the ReLu activation function:
  - Y = max(0,x)
- Extends Simplex LP solver
- Encode the equations connecting input to outputs as constrains in LP
- Create new variables for active (non-zero) and inactive (zero) outputs
- Explore assignments that satisfy constrains
- Applied to ACAS

Verisig[2]

- For closed-loop Cyber-Physical System
- Transforms NN into hybrid system
  - Transform sigmoid activation into quadratic equation
- Use reachability tools (e.g. dReach) to verify output



Problem: verify property ϕ about the closed-loop system

[1] Katz, Guy, Barrett, Clark, Dill, David L., Julian, Kyle, Kochenderfer, Mykel J." Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks". Computer Aided Verification. 2017.

[2] Radoslav Ivanov, James Weimer, Rajeev Alur, George J. Pappas, Insup Lee. "Verisig: verifying safety properties of hybrid systems with neural network controllers" HSCC '19

V&V Approach for Evolving ML

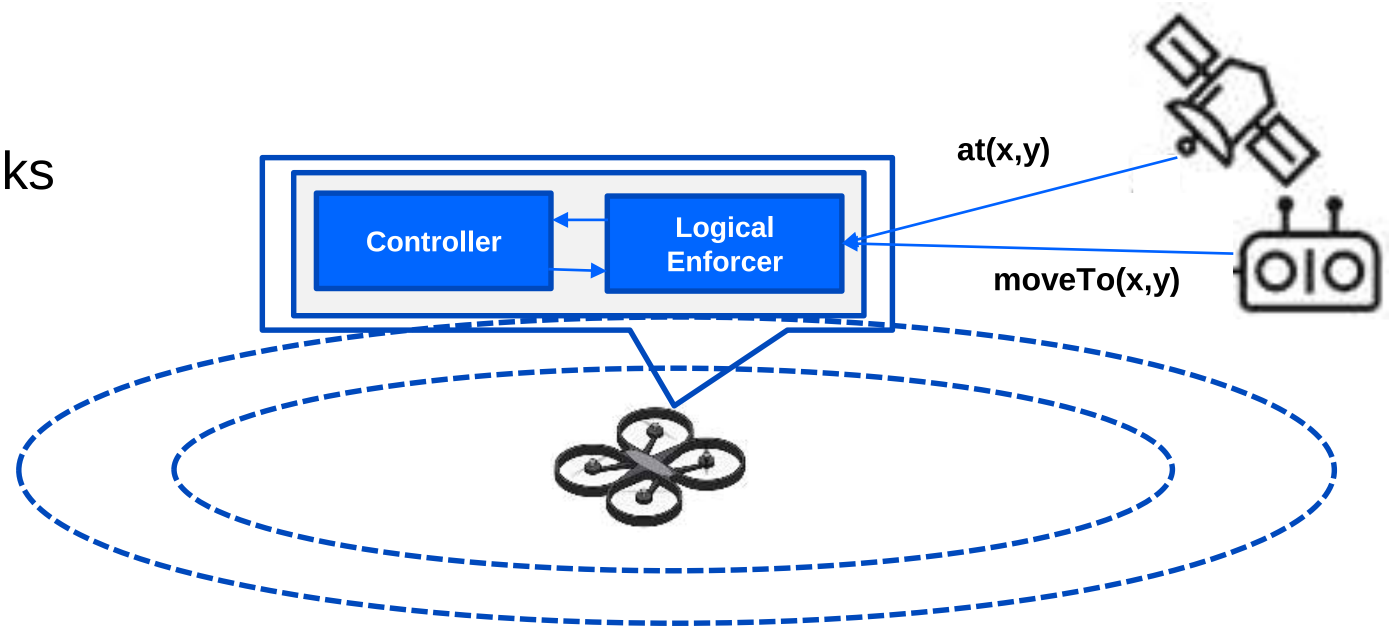
Add Enforcer

- Watch for safety property ϕ
- Replace unsafe actions

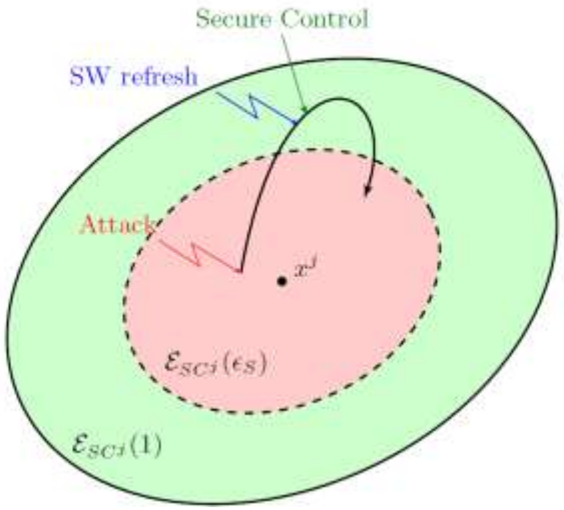
Formally: specify, verify, and compose multiple enforcers

- Logic: Enforcer intercepts/replaces unsafe action
- Timing: at right time
- Physics: verified physical effects

Protect enforcers against failures/attacks



Verifying Physics (Control Theory)



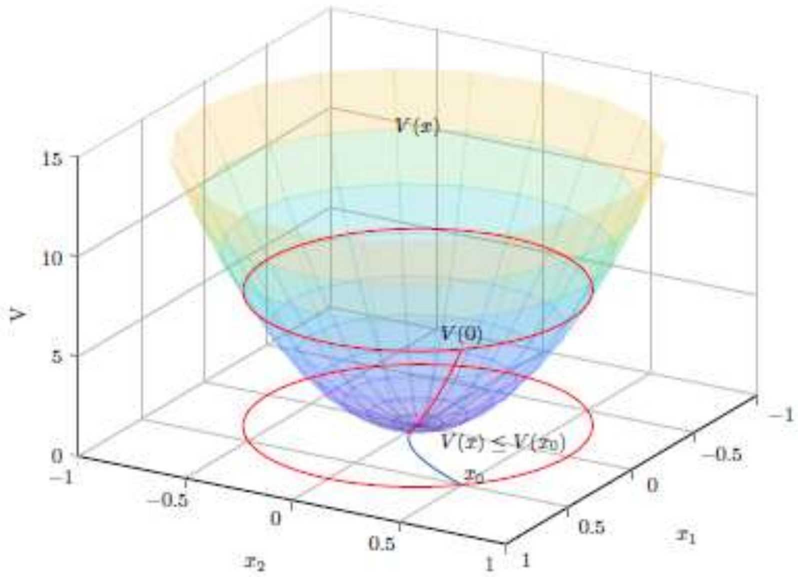
Model physics using control theory

System includes

- Physical vehicle and environment
- Software controlling airplane

Evaluate if combined system

- Behaves like a “cone” with setpoint at bottom
  – Theoretically known as Lyapunov function
- Enforcer periodically “samples” (monitors) for misbehavior
  – If between enforcer sample potential misbehavior
    - Theory verifies that system still in “cone” after misbehavior
  – Model also evaluates if enforcer recovery keeps system in cone



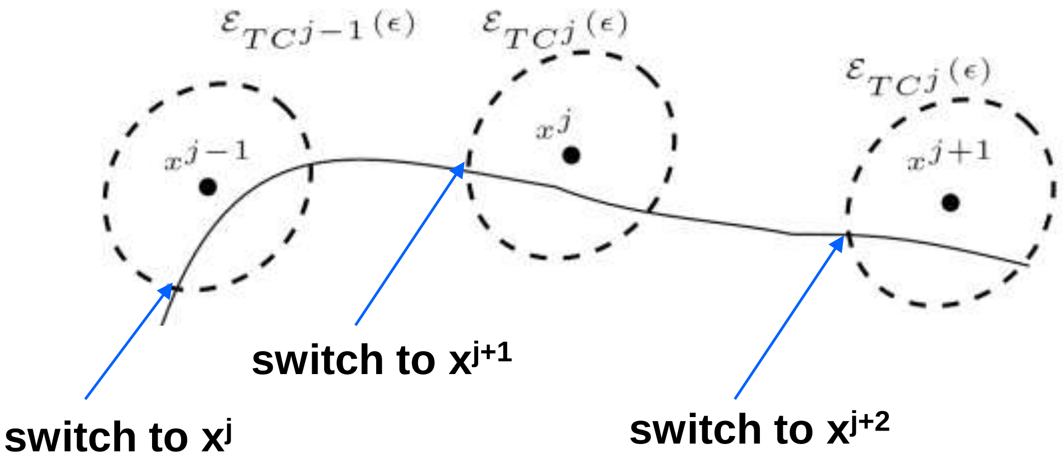
Analysis of Mission Progress

Idea:

Provide a sequence of waypoints that represent a sequence of equilibrium points around which we define the Safe Set.

Goal:

- Safety transition from one waypoint to the next one.
- Liveness (in the case of no errors)

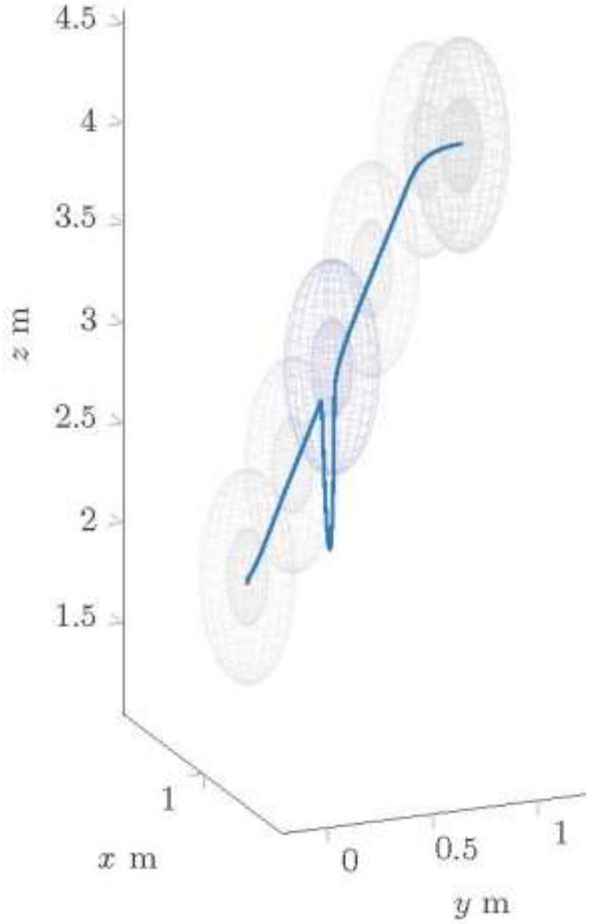


Analysis of Mission Progress Enforcing Unsafe Behavior

System model of a drone across mission

Safety “cone” in 3-D is a sphere

Evaluate misbehavior and enforcer recover



Drone Experiment

Are We Done Yet?

Scalable Verification

- Only verify safety-critical components
- Guarding unverified one

Trust

- Protect verified components
- Against attacks or bugs from unverified components

Enforcing Unverified Components

Enforcing Unverified Components

Ant picture attribution in : https://commons.wikimedia.org/wiki/File:Ant_illustration.jpg

Enforcing Unverified Components

But enforcer can be corrupted (bug or cyber attack)

Ant picture attribution in : https://commons.wikimedia.org/wiki/File:Ant_illustration.jpg

Add Memory Protection

Trusted = Verified & Protected

Ant picture attribution in : https://commons.wikimedia.org/wiki/File:Ant_illustration.jpg

Are We Done Yet?

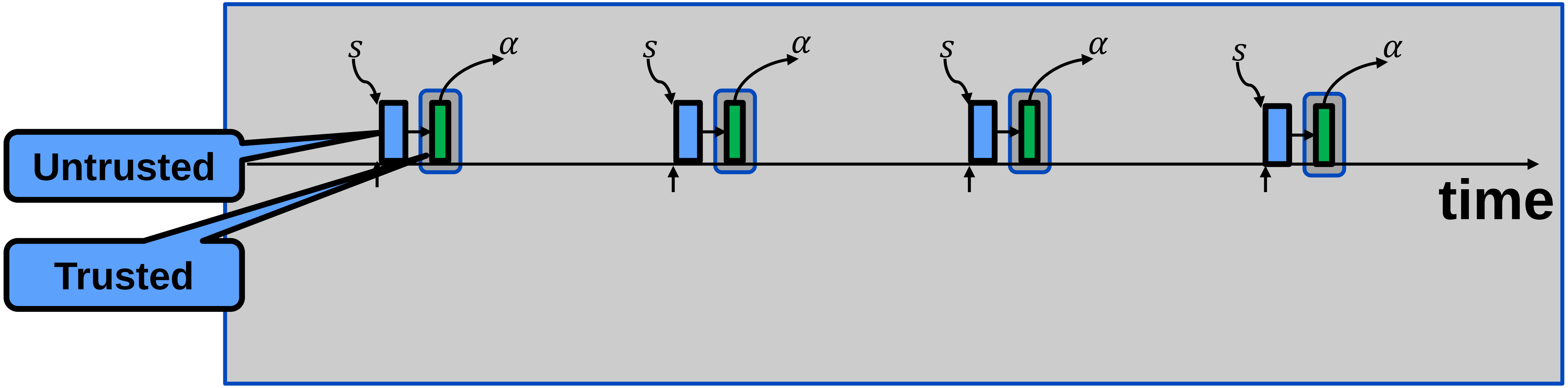
Timing can still be corrupted

- Guaranteed correct value
- BUT potentially at wrong time

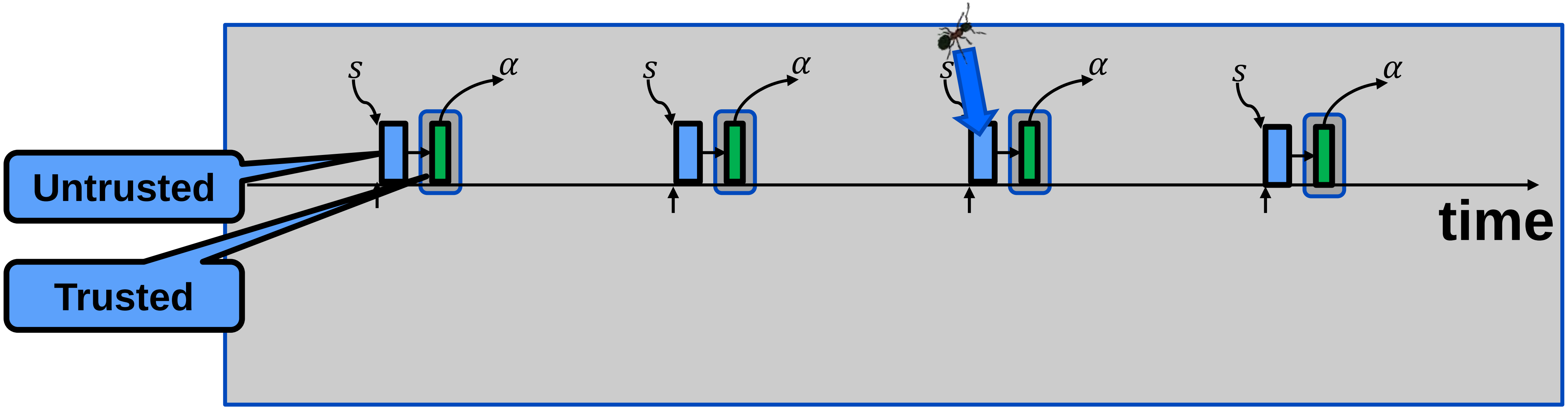
Trusted timely actuation

- Tamper-proof time-triggering mechanism
- In sync with periodic controller
- In sync with expected untrusted

Periodic Execution Must Finish by Deadline

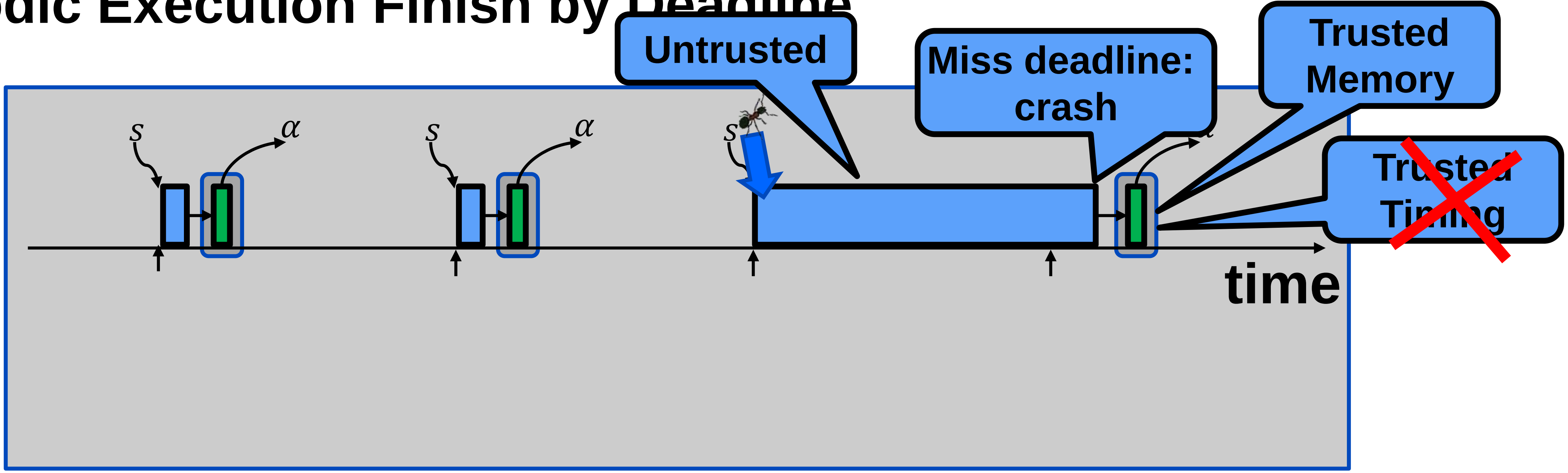


Periodic Execution Must Finish by Deadline



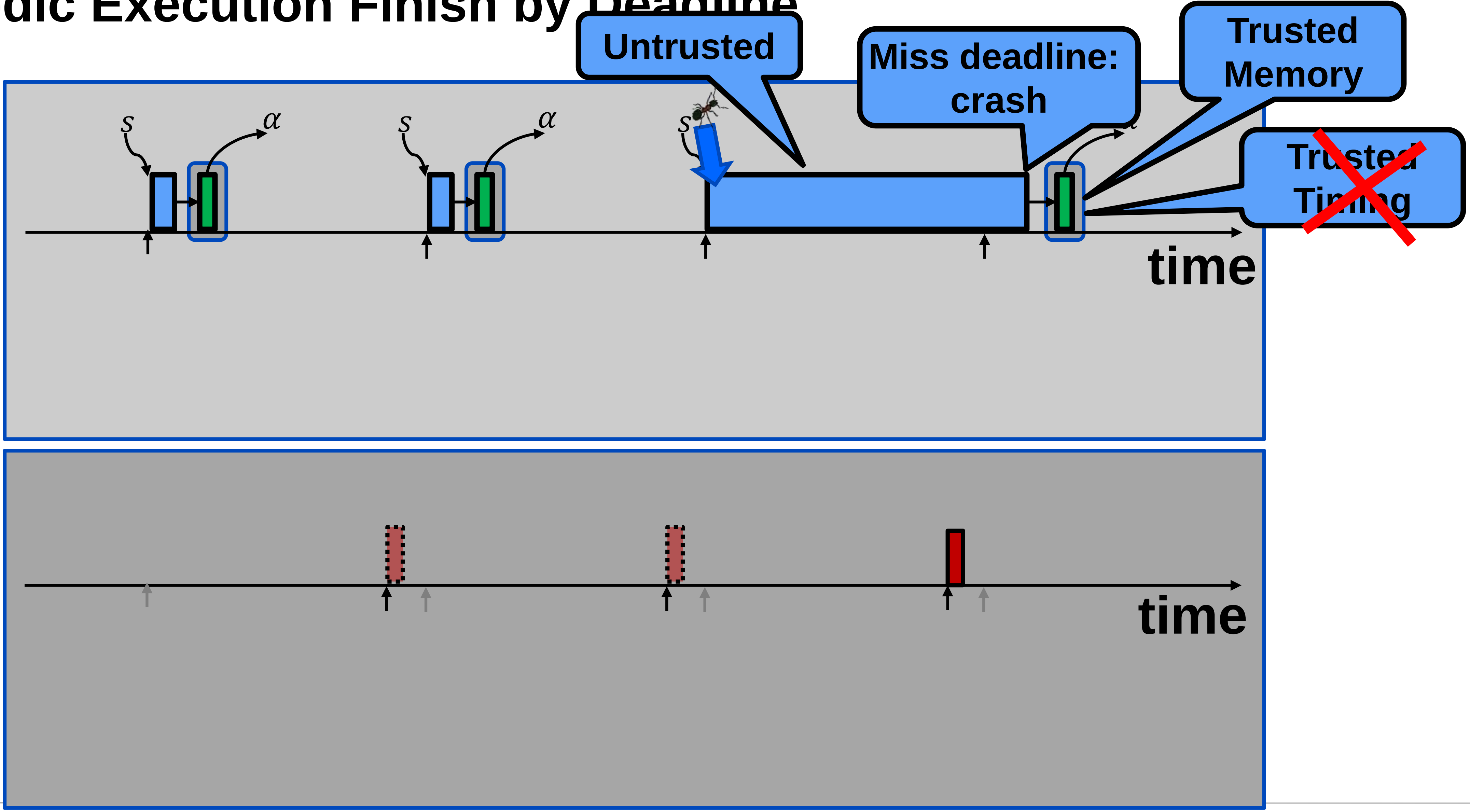
Ant picture attribution in : https://commons.wikimedia.org/wiki/File:Ant_illustration.jpg

Periodic Execution Finish by Deadline

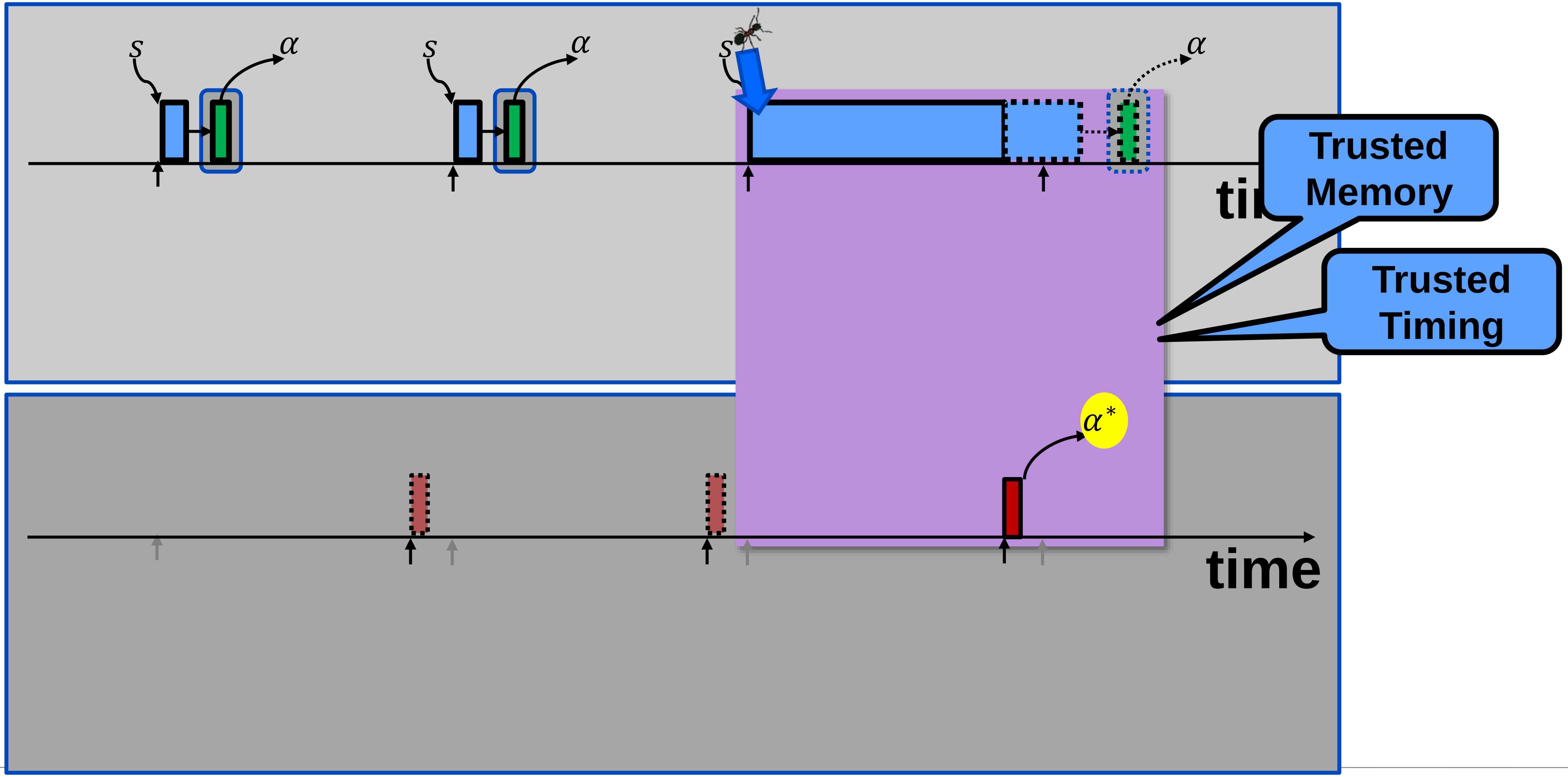


Ant picture attribution in : https://commons.wikimedia.org/wiki/File:Ant_illustration.jpg

Periodic Execution Finish by Deadline

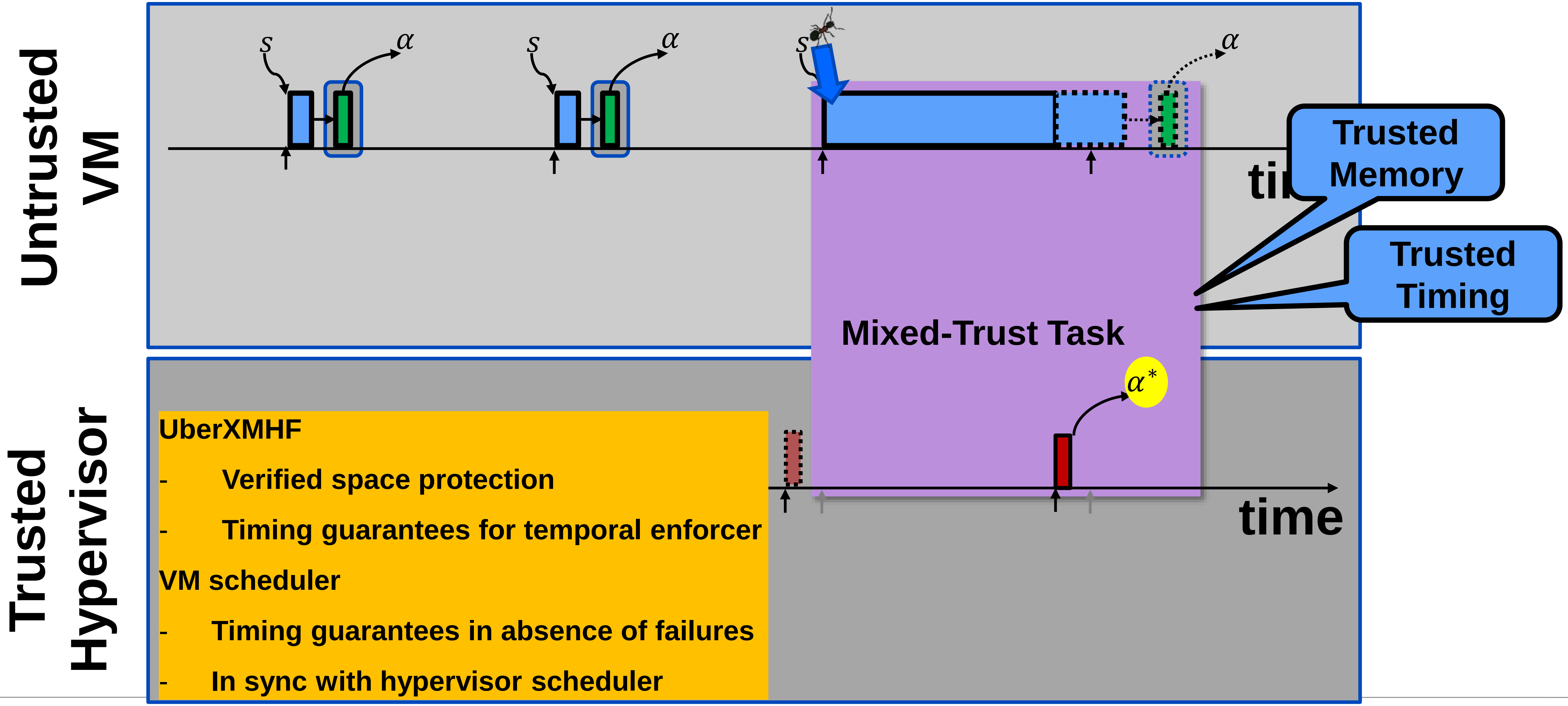


Periodic Execution Finish by Deadline



Ant picture attribution in : https://commons.wikimedia.org/wiki/File:Ant_illustration.jpg

Real-Time Mixed-Trust Computation



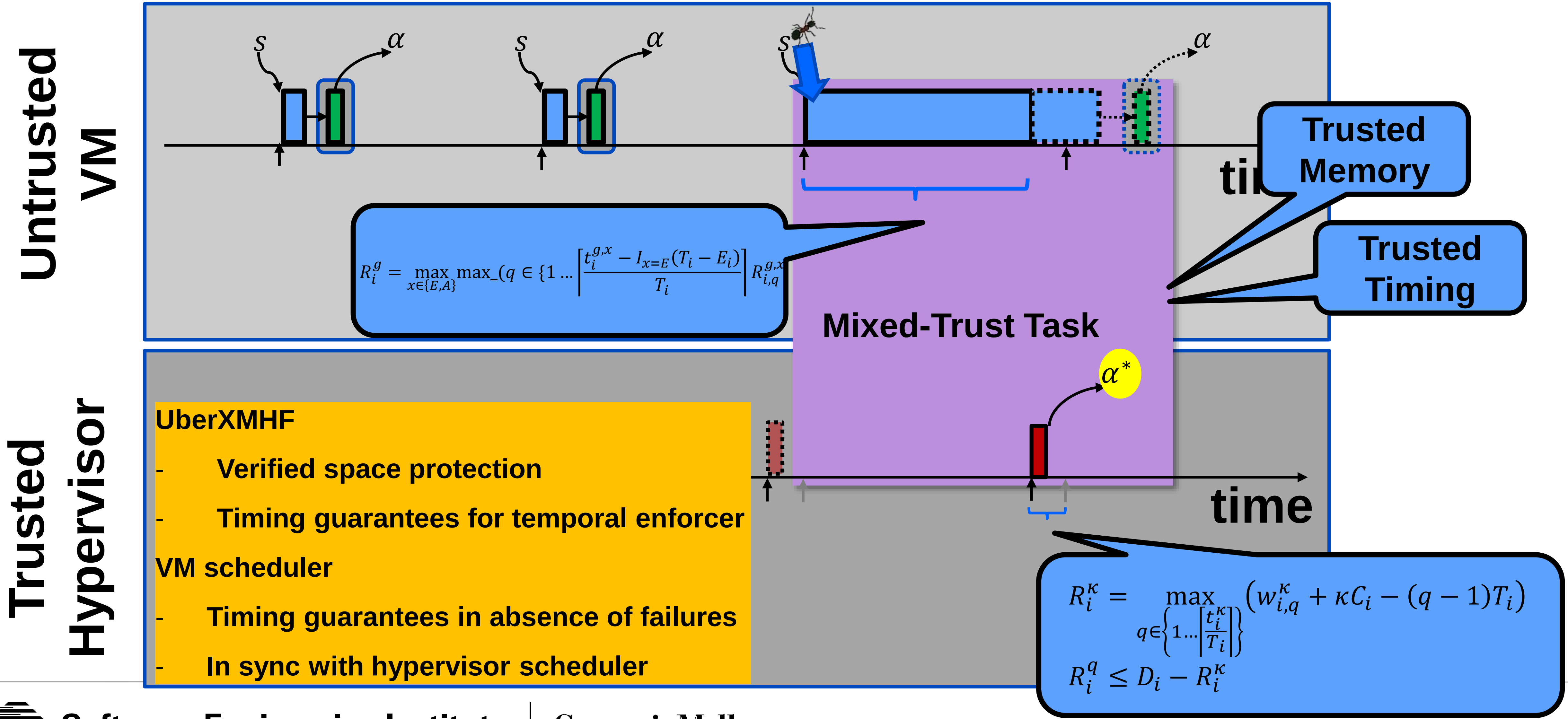
UberXMHF

- Verified space protection
- Timing guarantees for temporal enforcer

VM scheduler

- Timing guarantees in absence of failures
- In sync with hypervisor scheduler

Real-Time Mixed-Trust Computation



Concluding Remarks

ML Verification for Trained ML

- LP-Based
- Hybrid rechability for CPS

ML Verification for Evolving ML

- Enforcers to
  - Monitor and
  - Correct unsafe actions

Focus on key properties:

- Safety
- Security

Combined Relevant Scientific Domains

- Timing
- Logic
- Physics (Control)

Verification only effective if protected!

- Verified Protection: Hypervisor