



Runtime-Assurance for AI

Dionisio de Niz, Ph.D.

Principal Researcher & Technical Director
Assuring Cyber-Physical Systems



Copyright 2020 Carnegie Mellon University.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

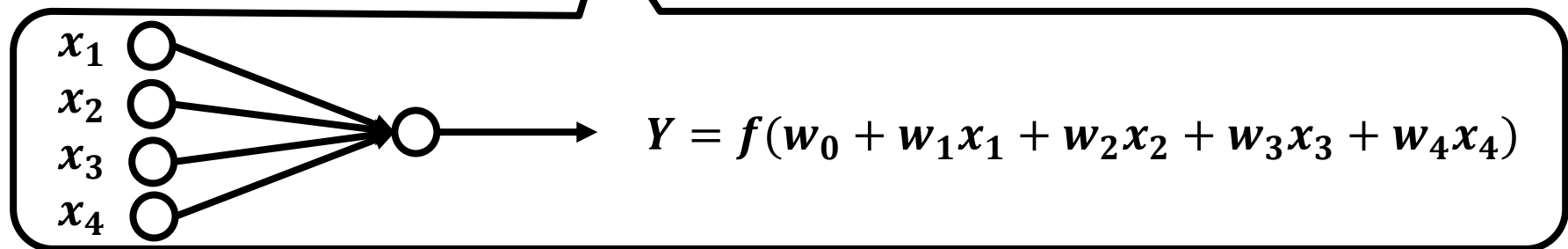
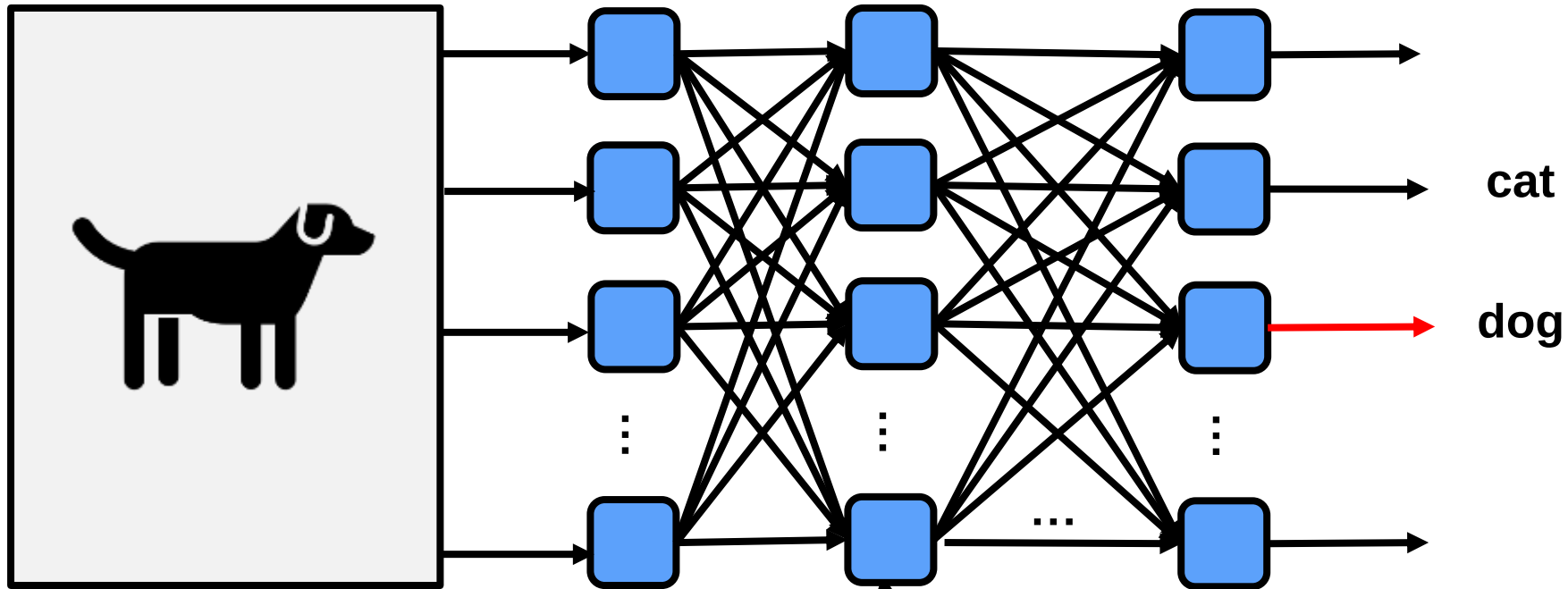
This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM20-0993



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



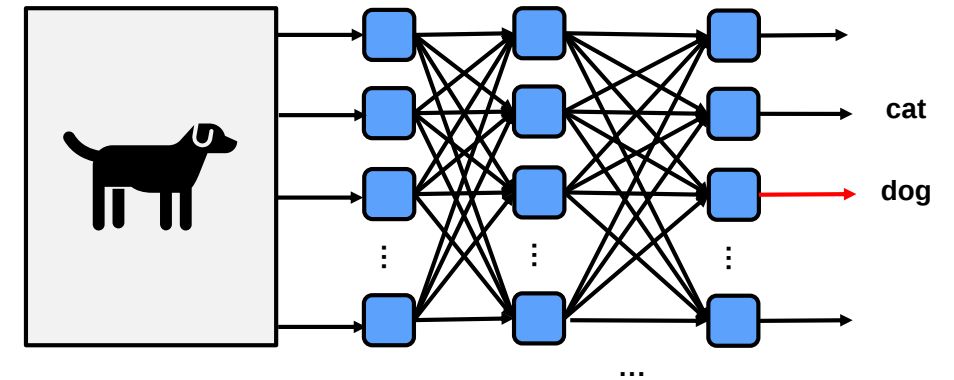
V&V Machine Learning Challenges

Computation not based on application logic (Trained ML)

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

Computation changes at runtime (Evolving ML)

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



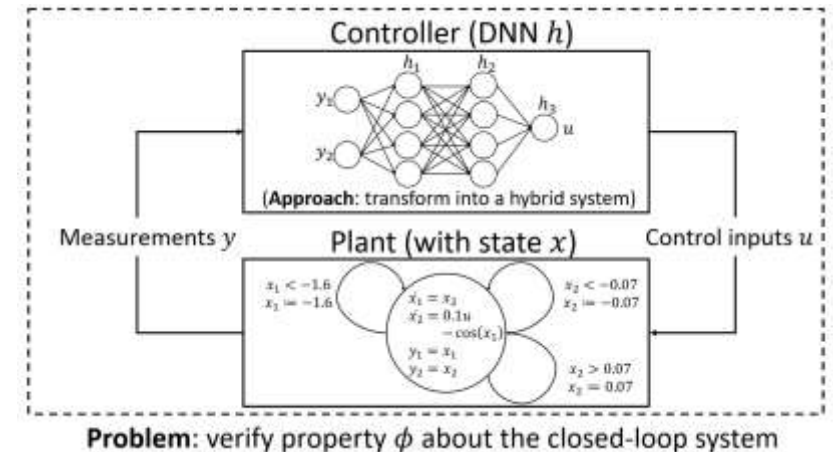
V&V Approaches for Trained ML

Reluplex¹

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

Verisig²

- 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



¹Katz, 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.

²Radoslav 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

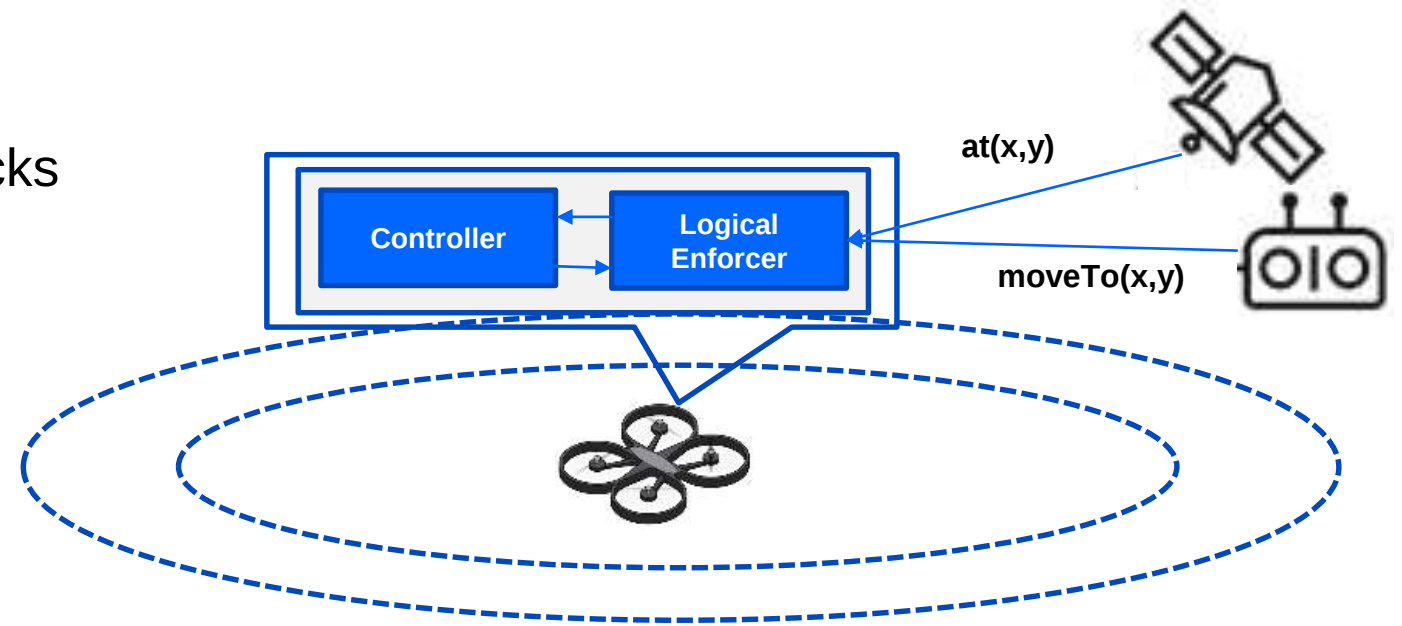
Add Enforcer

- 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)

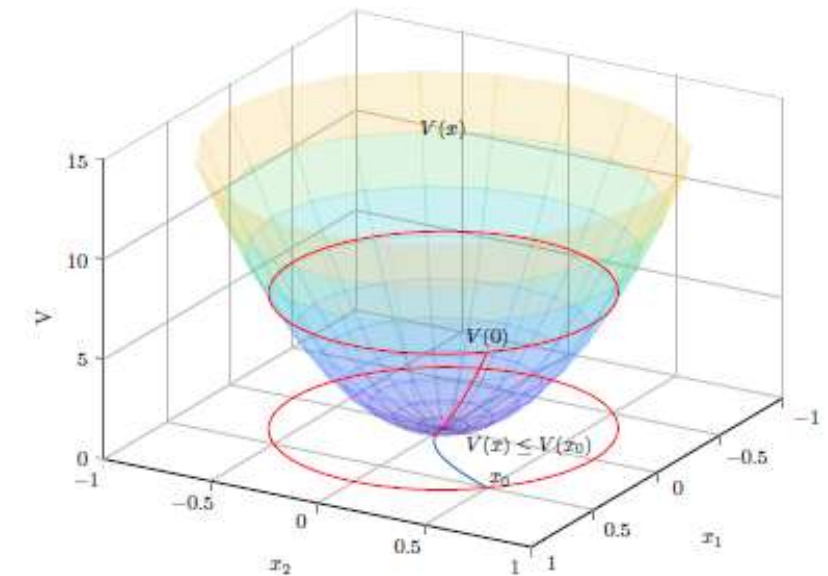
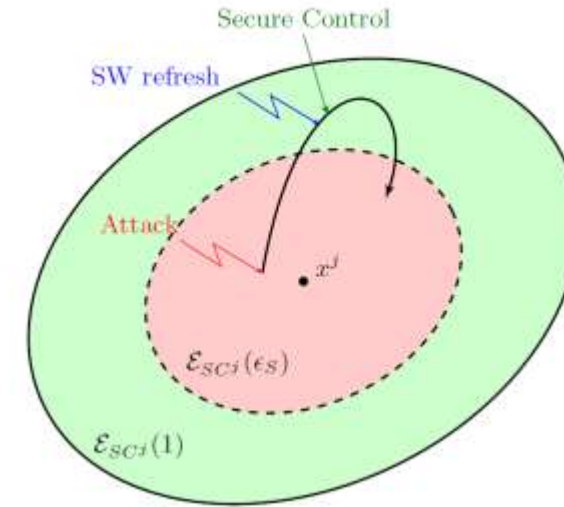
Model physics using control theory

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



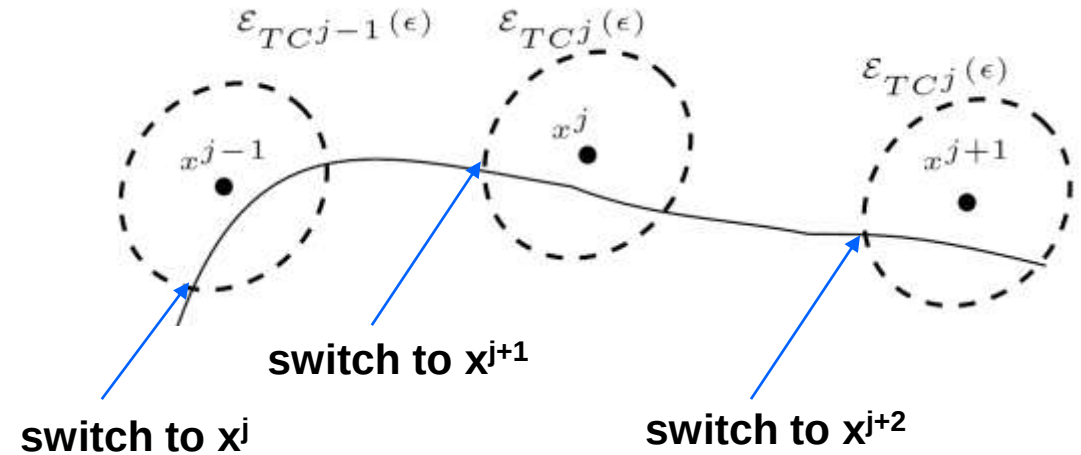
Analysis of Mission Progress

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)

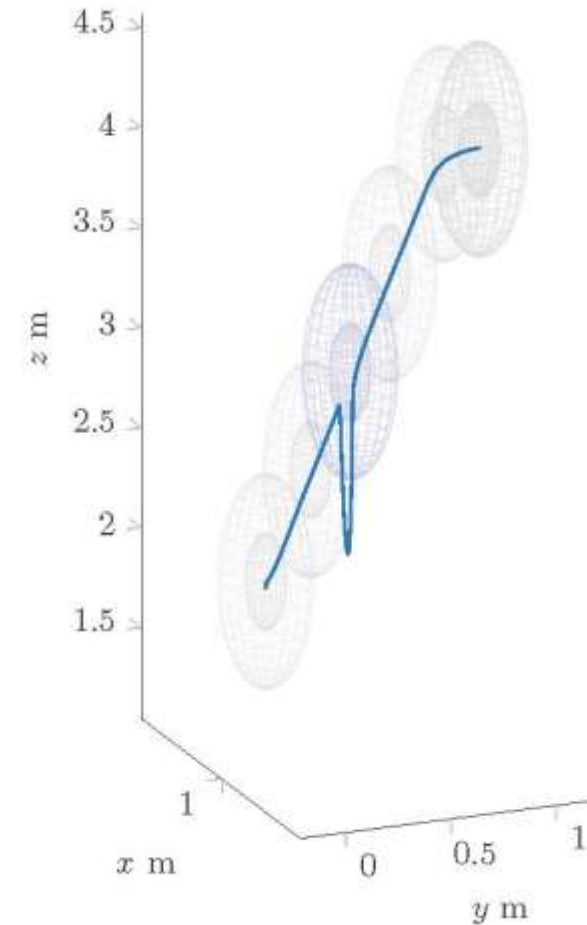


Analysis of Mission Progress Enforcing Unsafe Behavior

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

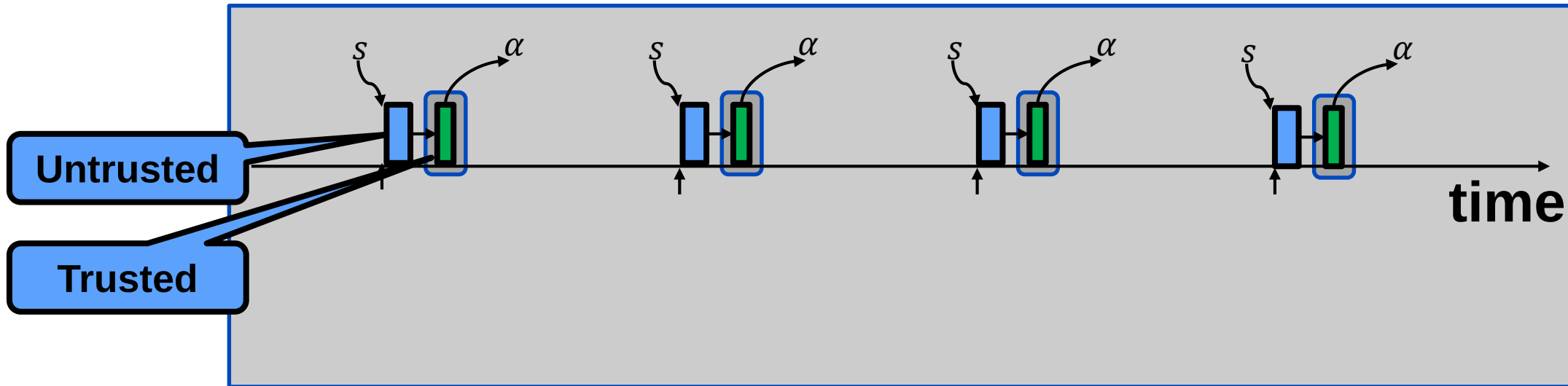
- Guaranteed correct value
- BUT potentially at wrong time

Trusted timely actuation

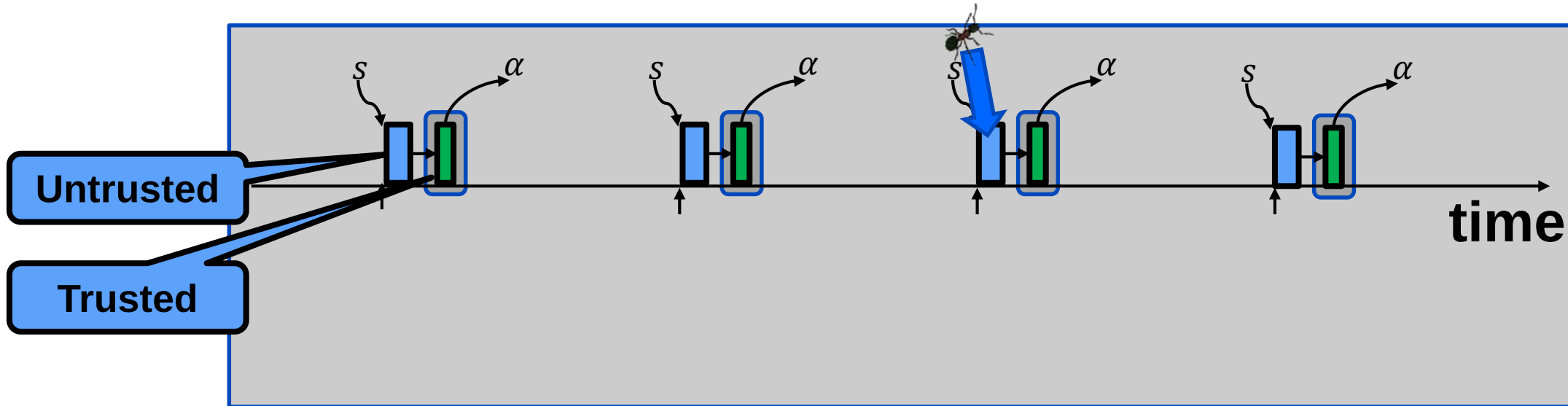
- Tamper-proof time-triggering mechanism
- In sync with periodic controller
- In sync with expected untrusted



Periodic Execution Must Finish by Deadline



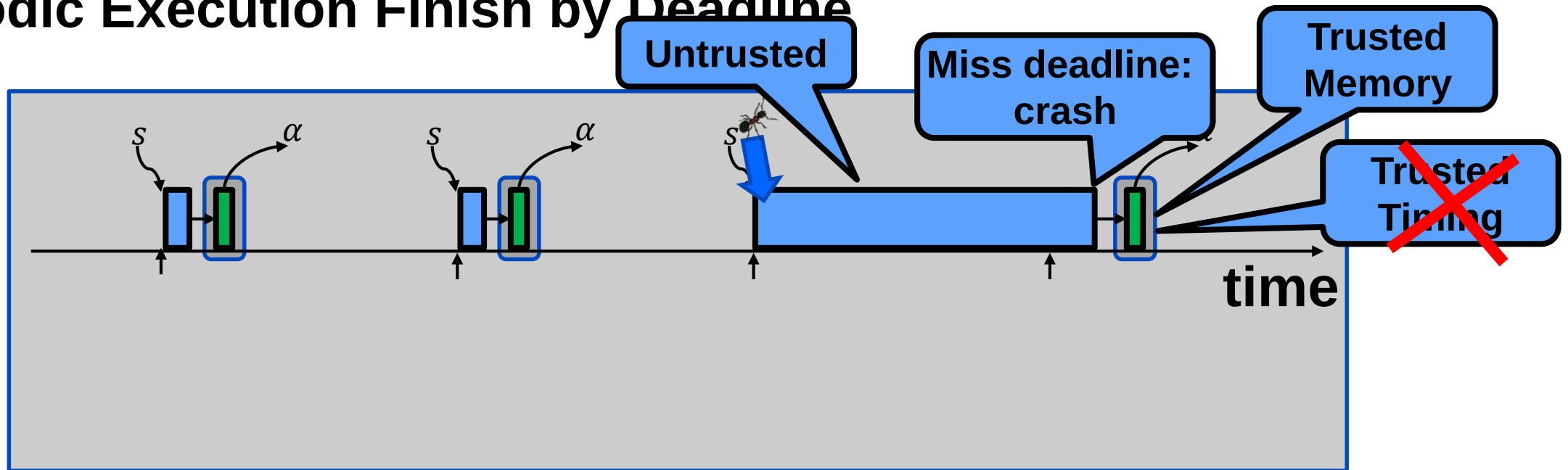
Periodic Execution Must Finish by Deadline



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



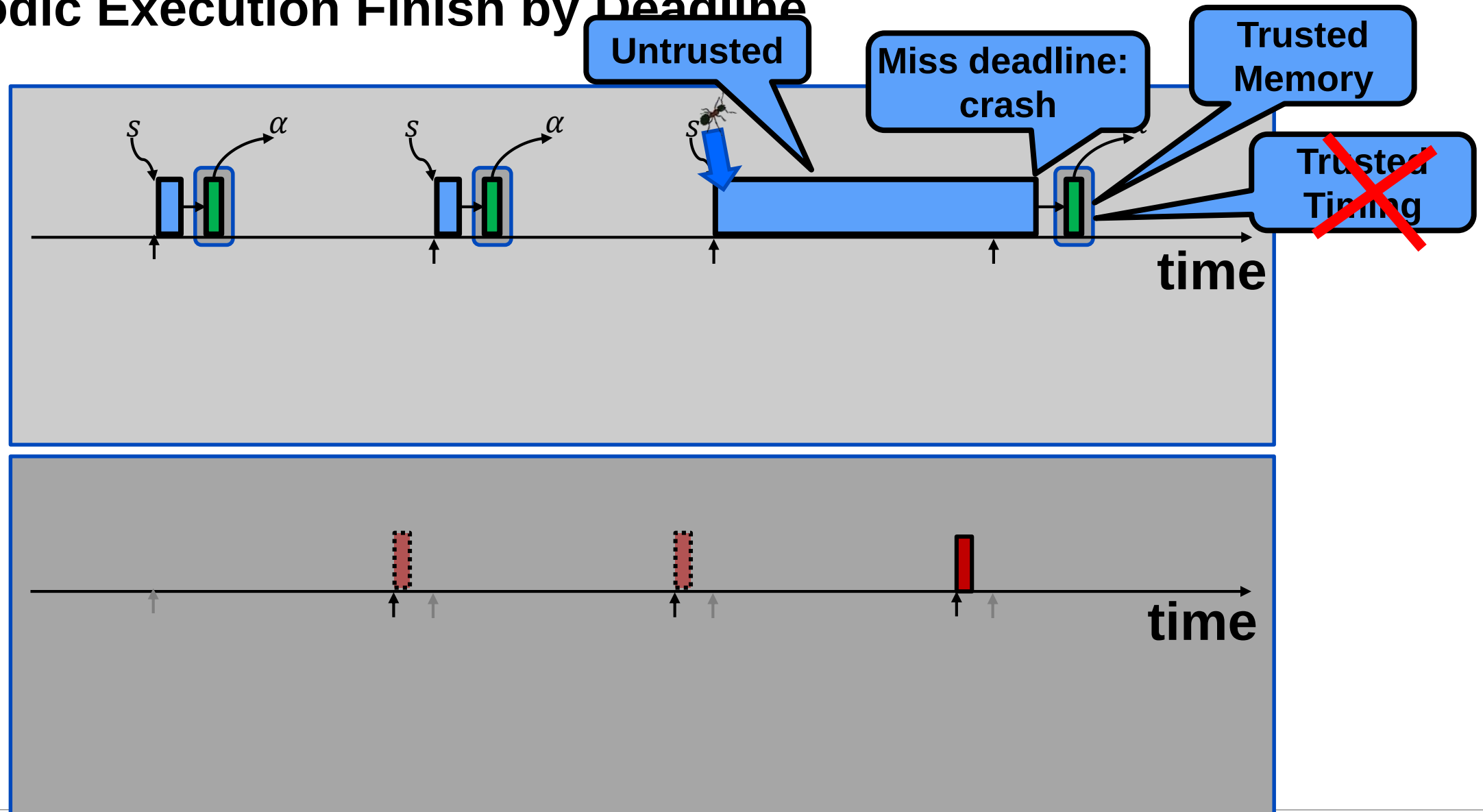
Periodic Execution Finish by Deadline



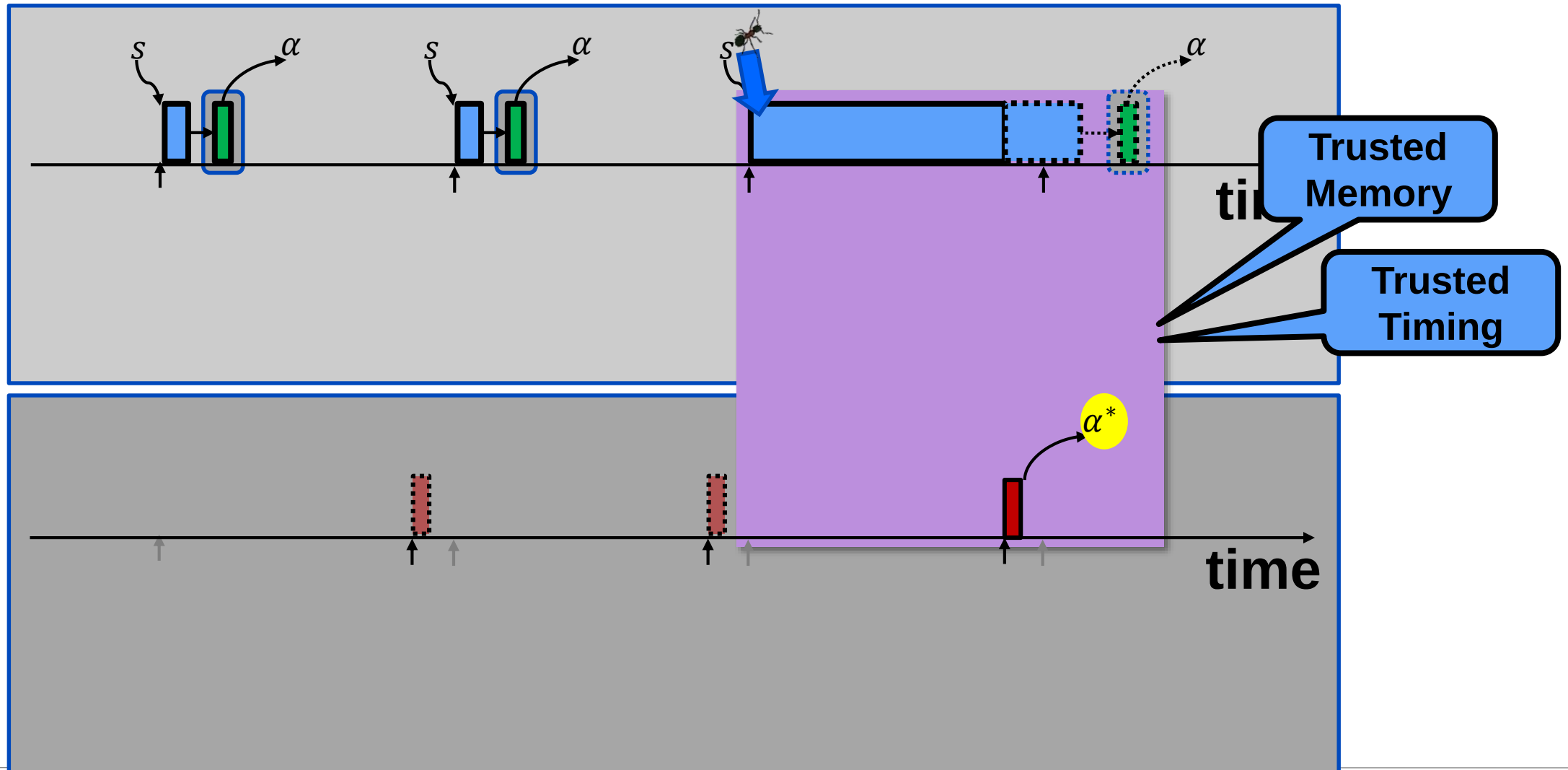
Ant picture attribution in : https://commons.wikimedia.org/wiki/File:Ant_illustration.jpg



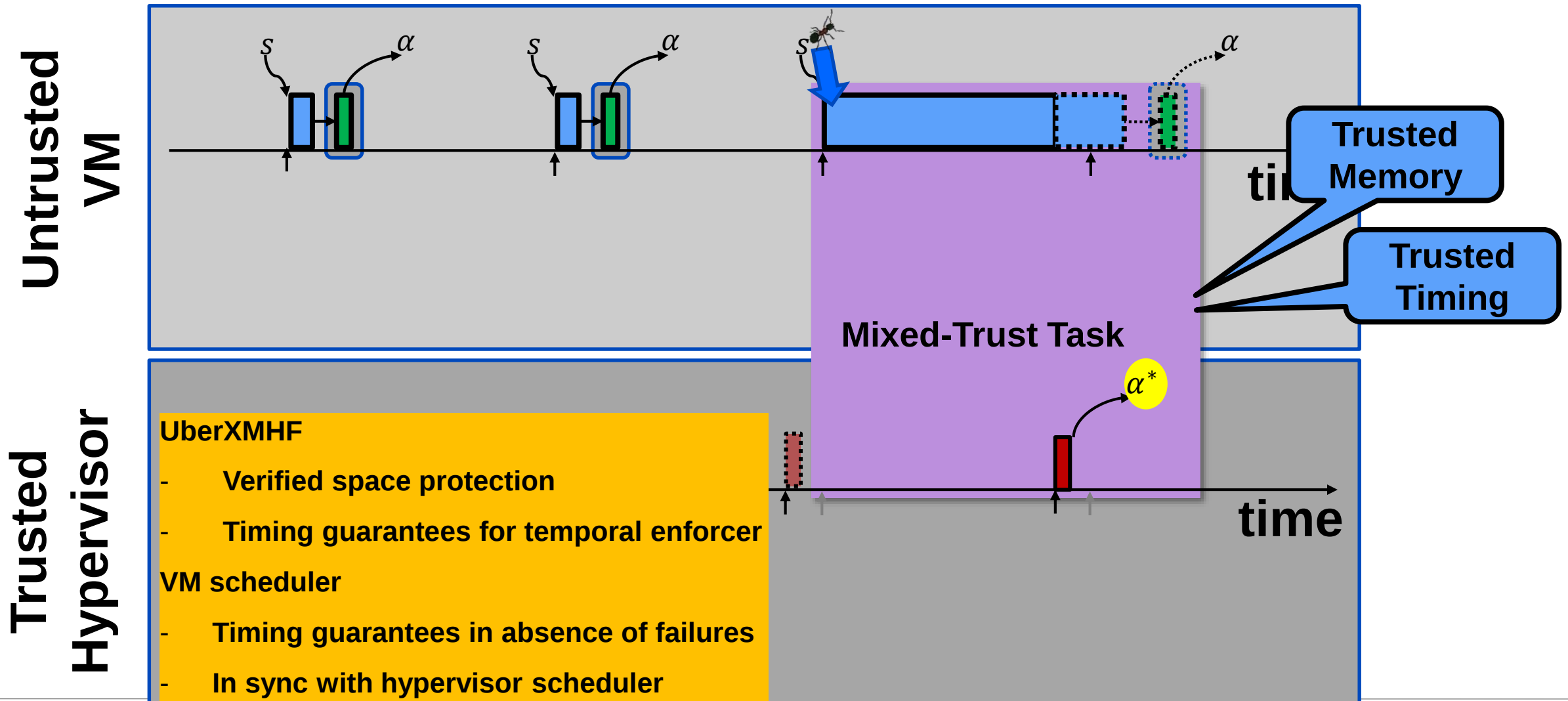
Periodic Execution Finish by Deadline



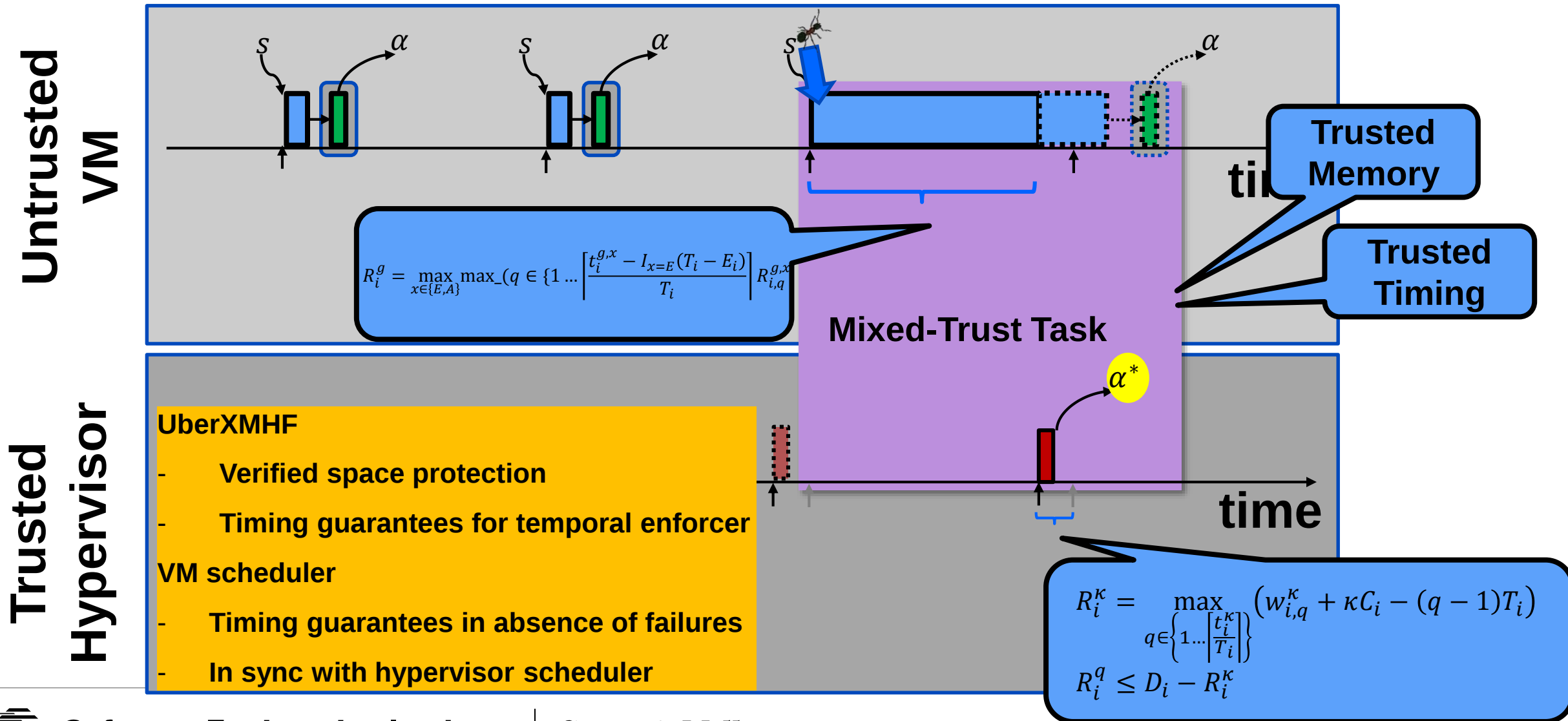
Periodic Execution Finish by Deadline



Real-Time Mixed-Trust Computation



Real-Time Mixed-Trust Computation



Concluding Remarks

ML Verification for Trained ML

- LP-Based
- Hybrid reachability 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

