

# Rapid Certifiable Trust

## Introduction

Fielding **new technologies** is essential to **preserve superiority**. However, this is only possible if these technologies are **validated for safety**.

## Challenges for validation

- Growing system complexity
- Changing behavior at runtime (e.g., machine learning)
- Interactions with physical world (e.g., vehicles)
  - Correct value
  - At right time (before crash)

## Methods

**Formal** automatic verification

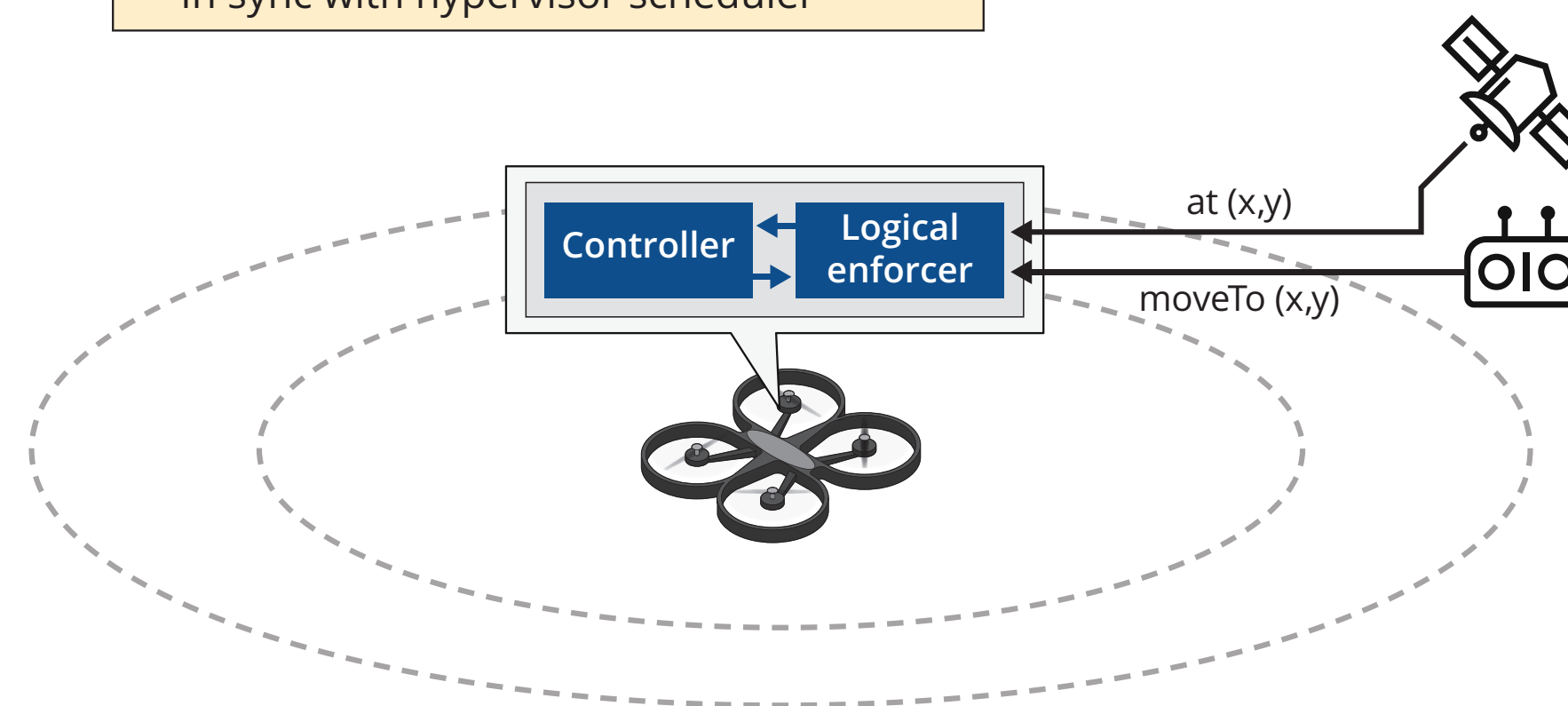
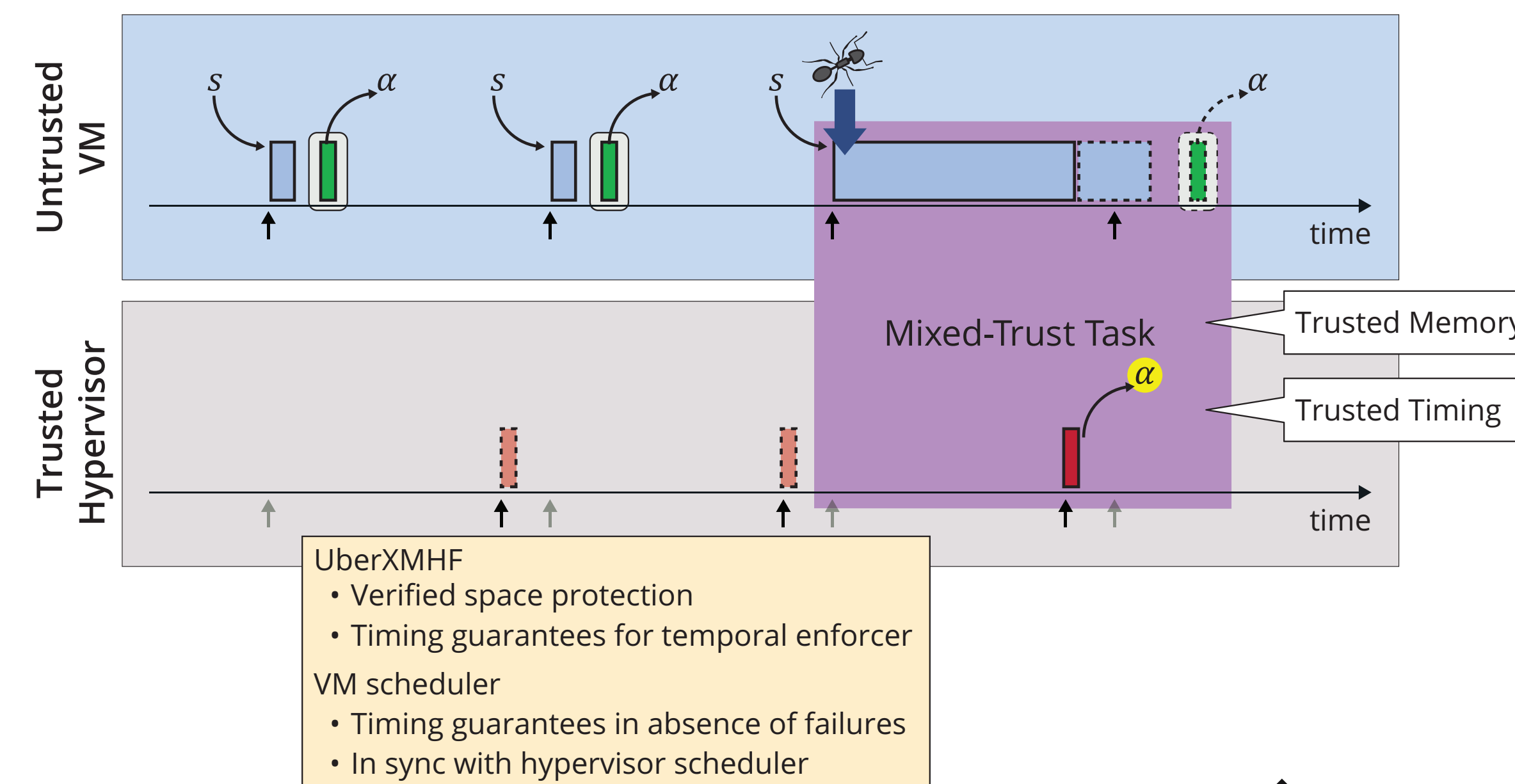
- Scalable
  - Unverified components
  - Monitored and **enforced** by verified components
  - Protected from unverified components
- Verified from
  - Physics: verify reaction of physical model (e.g., physical vehicle)
  - Logic: correct value, with correct protection
  - Timing: At the right time
- Verified protection

## Results

- Real-Time **Mixed-Trust** Computation
- Verified protection mechanism (micro-hypervisor: UberXMHF)
- Timing verification of combined trusted/untrusted (mixed-trust)
- Physics verification of enforcement

**Preserve safety** by verifying only a small part of the system. **Assure trust** by protecting verified parts.

## Trust = Verified + Protected



## Verifying Physics

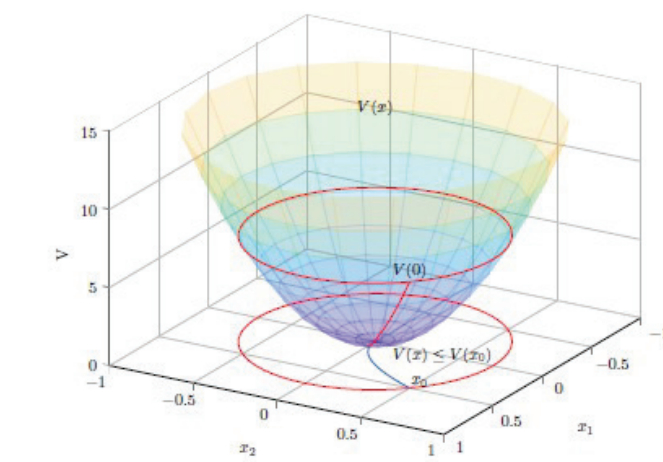
Ensure that an unverified controller cannot violate safety bounds

**Controlled System:**  $\dot{x} = f_\varphi(x) \triangleq f(x, \varphi(x))$   
**Lyapunov Function:**  $V_\varphi : \mathbb{R}^n \rightarrow \mathbb{R}, \mathcal{N}_{V_\varphi}(x_{eq}) \subseteq \mathcal{N}_\varphi(x_{eq}), V_\varphi(x_{eq}) = 0$  and  $\forall x \in \mathcal{N}_{V_\varphi}(x_{eq}) - \{x_{eq}\} : (i) V_\varphi(x) > 0,$

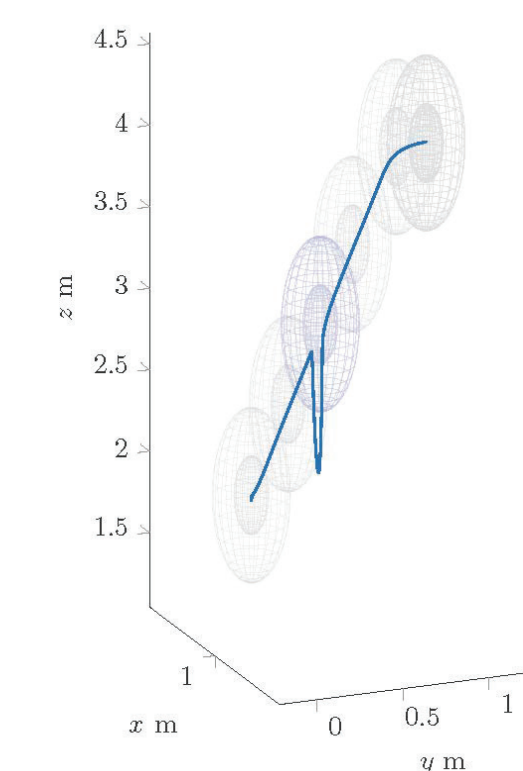
$$\dot{V}_\varphi(x) = \frac{\partial V}{\partial x} \cdot f_\varphi(x) < 0$$

**Lyapunov level set:** For  $\epsilon > 0,$

$$\mathcal{E}_\varphi(\epsilon) = \{x \in \mathcal{N}_{V_\varphi}(x_{eq}) \mid V_\varphi(x) \leq \epsilon\}, \quad \epsilon \leq 1$$

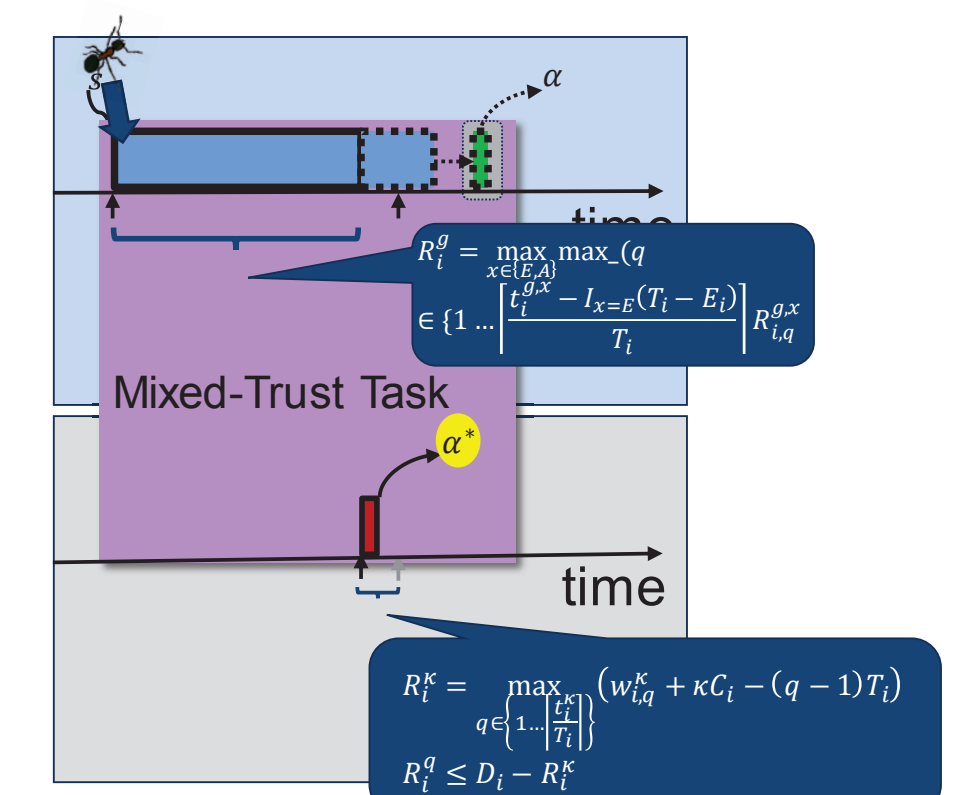


## Mission: sequence of set points



## Verifying Timing

Response time  $\leq$  Deadline



Copyright 2019 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.

Internal use:\* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use:\* 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 external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

\* These restrictions do not apply to U.S. government entities.

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

DM19-1044