

Rapid Certifiable Trust

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

Challenges for Validation

- Increasingly complex systems
- Changing behavior at runtime (e.g., machine learning)
- Interactions with physical world (e.g., vehicles)
 - Must have correct value
 - Occur at right time (i.e., before crash)

Methods

Formal automatic verification

• Scalable

- Unverified components
- Monitored and enforced by verified components
- Verified components protected from unverified components

• Verified

- Physics: verify reaction of physical model (e.g., physical vehicle)
- Logic: correct value with correct protection
- Timing: occurs at the right time

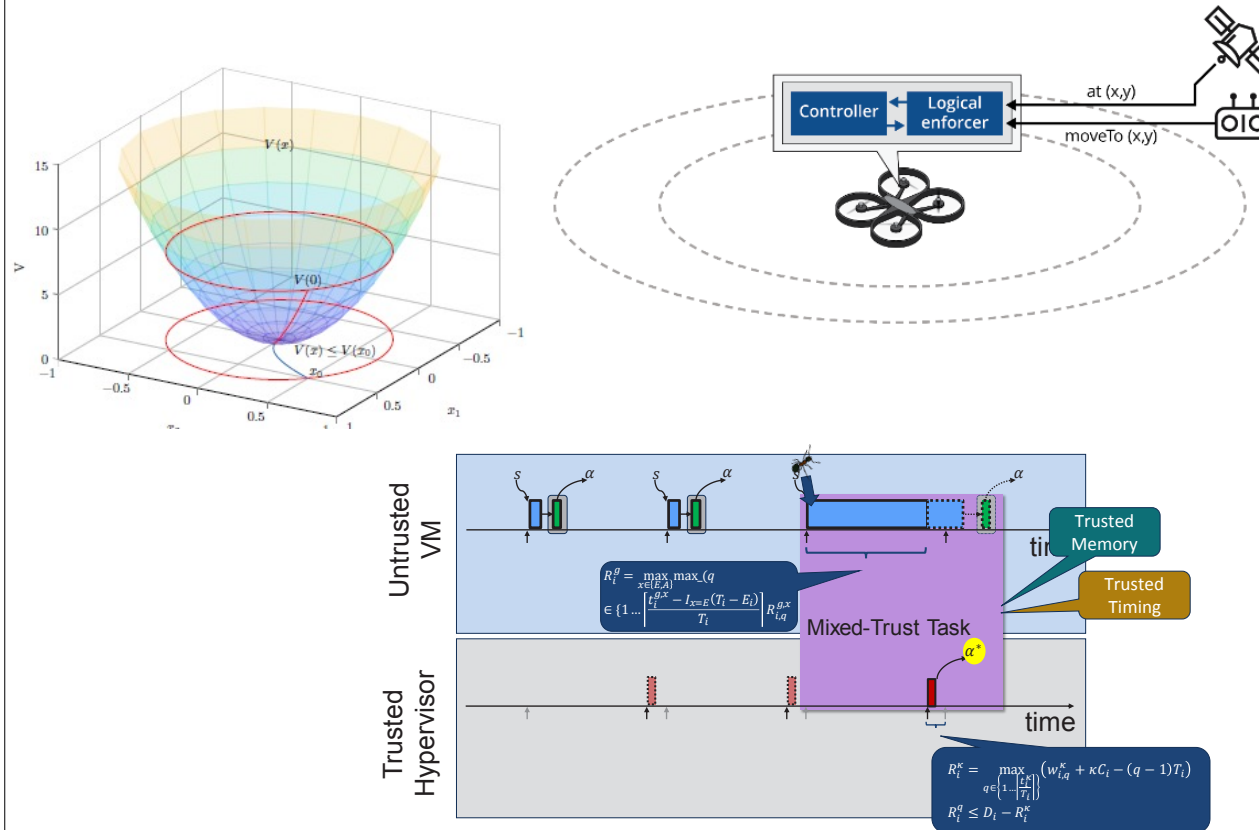
• Protect verified components

Results

Real-time Mixed-Trust Computation

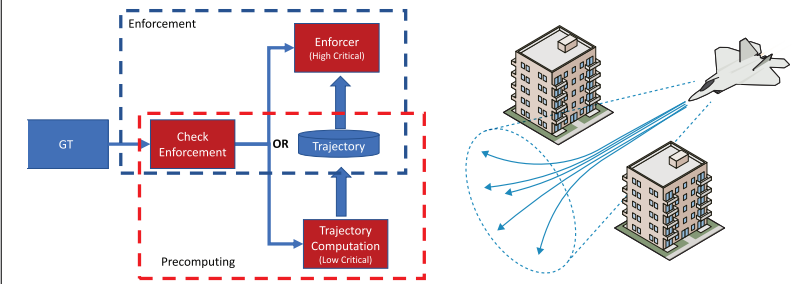
- Verified protection mechanism (micro-hypervisor: uber XMHF)
- Timing verification of combined trusted/untrusted (mixed-trust)
- Physics verification of enforcement

Preserve safety by verifying only a small part of the system.
Assure trust by protecting the verified part.
Trust = Verified + Protected



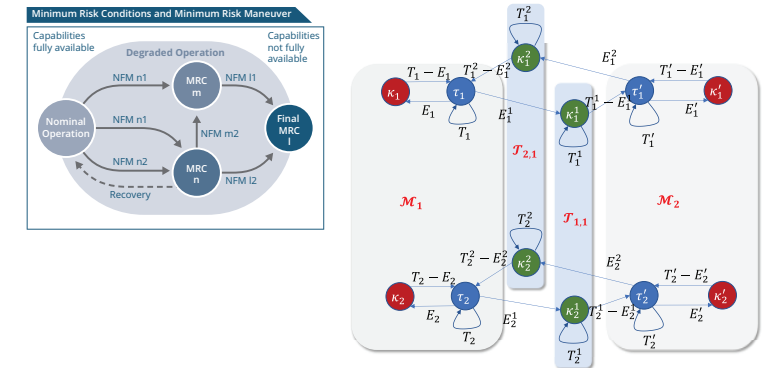
NEW RESULTS

Predictive Mixed-Trust Scheduling



Balance trajectory production/consumption: $G_i^d(I_i - 1) - S_i^e \geq 0$
 + Response: $R_i^p = \kappa C_i^p + \sum \left[\frac{R_i^p}{T_j} \right] \kappa C_j^p - \left[\frac{R_i^p}{J_j T_j} \right] (\kappa C_j^p - \kappa C_j^e)$

Resilient Mixed-Trust Autonomy Scheduling



$$J(g_j) = \max_{v_{j,q} \in V_j} (D_{j,q} - C_{j,q})$$

$$rf_{\pi_j}^{v_{i,k}}(t) := \max\{e(\pi_j) | \pi_j \text{ is prefix of } \pi_j \text{ and } \text{end}(\pi_j, v_{i,k})\}$$

$$\text{end}(\pi, v_{i,k}) = \begin{cases} p(\pi) \leq t & \text{if } v_{i,k} \text{ is non-preemptive} \\ p(\pi) < t & \text{otherwise} \end{cases}$$

$$MI(v_{i,k}) = P(v_{i,k}) + \sum_{g_j \in hp(i)} rf_{\pi_j}^{v_{i,k}}(MI(v_{i,k}) + J(g_j))$$

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.

DM20-0907