Certifiable Distributed Runtime Assurance

Challenges

- Assure safety of distributed cyber-physical systems
- Unpredictable algorithms (machine learning)
- Multi-vehicle (distributed) coordinating to achieve mission

Solutions

- Add simpler (verifiable) runtime enforcer to make algorithms predictable
- Formally: specify, verify, and compose multiple enforcers
- Enforcer intercepts/replaces unsafe action at right time

Formalization (time-aware logic)

State of system: Variable Values

Statespace & Actions

- $S = \{s\}$, Safe states: $\phi \subseteq S$
- $R_P(\alpha) \subseteq S \times S$; $R_P(\alpha, s) = \{s' \mid (s, s') \in R_P(\alpha)\}$

Enforceable states

- $C_\phi = \{s \mid \exists \alpha: R_P(\alpha, s) \in C_\phi\}$

Safe actions:

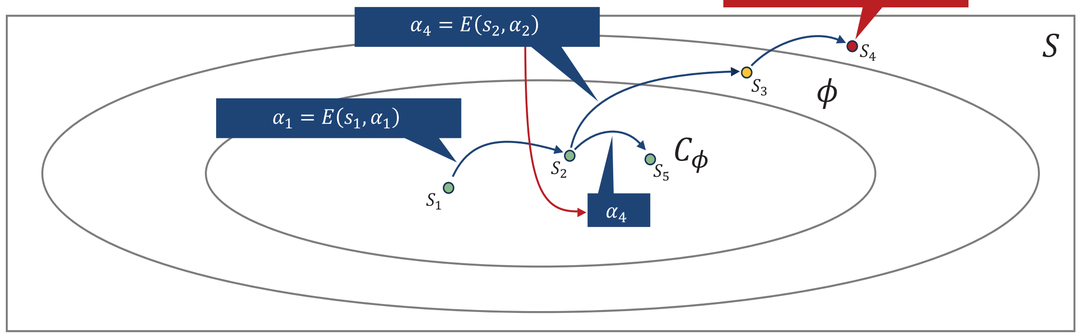
- $SafeAct(s) = \{\alpha \mid R_P(\alpha, s) \in C_\phi\}$

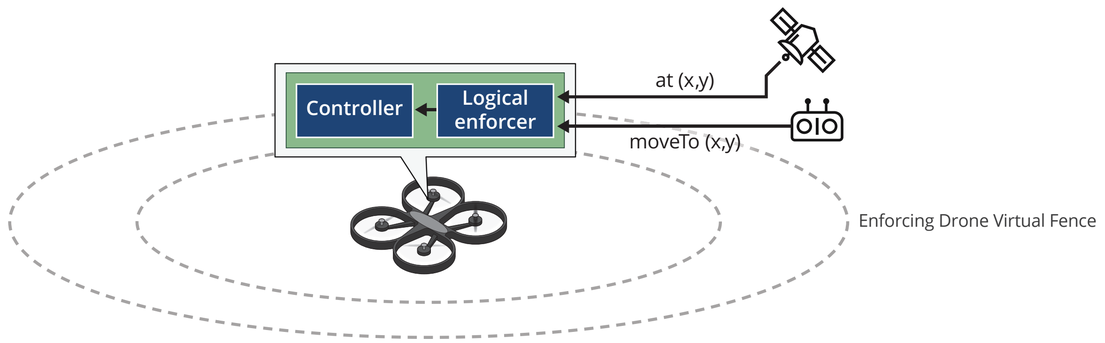
Logical Enforcer: $E = (P, C_\phi, \mu)$

- Set of safe actions: $\mu(s) \subseteq SafeAct(s)$
- Monitor and enforce safe action:

$$(\alpha =) \begin{cases} \alpha, & \alpha \in \mu(s) \\ pick(\mu(s)), & otherwise \end{cases}$$

Enforcer $\mathrm{E}(s, \alpha): \alpha \in SafeAct(s)\ ?\ \alpha : \alpha' \in SafeAct(s)$





Timing Enforcement

- Unverified software may never finish!
- => No action produced to be enforced!

Temporal Enforcer

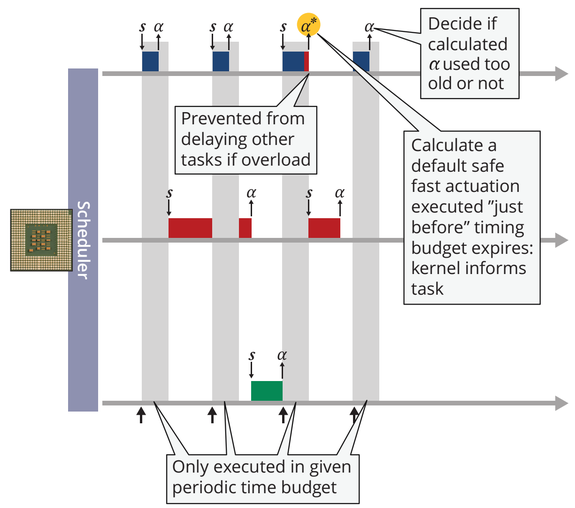
- Protect other tasks from bogus never-ending (or large) executions
- Produce default safe actuation if task takes too long

How

- Each task gets a CPU budgetstop task if budget exceeded
- If task about to exceed budget executesafe action

Timing Guarantees

- Never allow task to exceed budget
- Always execute actuation

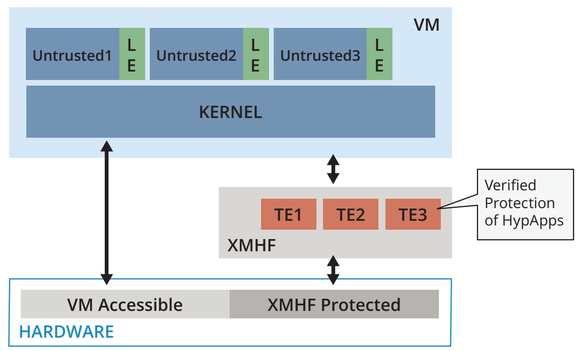


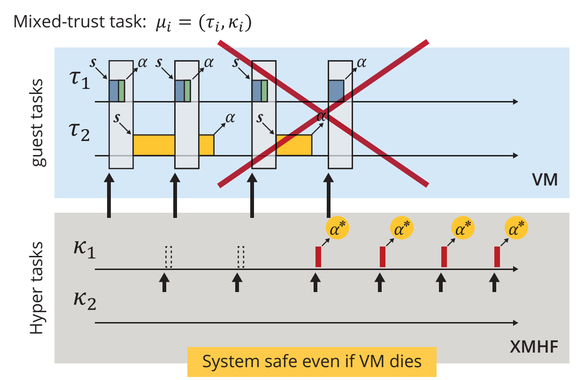
Protecting Enforcer from Untrusted Components

- Unverified software faults may corrupt enforcers

Mixed Trust Computing and Scheduling

- Untrusted/unverified software for quick/ cheap fielding
- Trusted enforcer to guarantee safety properties
- Micro-Hypervisor protects enforcers
- Coordinated VM+Hypervisor schedulers guarantees timing





Enforcers Allows Verification of Complex CPS: Autonomous Vehicles

- Limit misbehavior with verifiable enforcers
- Result: Verified whole system

Verified: Logic, Timing, Physics!

KEY for Assured Autonomy

Dr. Dio de Niz | dionisio@sei.cmu.edu