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

Timing Enforcer
- Unverified software may never finish!
- \(\Rightarrow\) 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 budget
- Stop task if budget exceeded
- If task about to exceed budget, execute safe action

Timing Guarantees
- Never allow task to exceed budget
- Always execute actuation

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