ReSecure: A Restart-Based Security Protocol for Tightly Actuated Hard Real-Time Systems

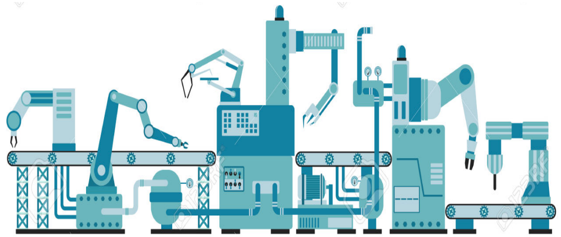
Fardin Abdi, **Monowar Hasan**, Sibin Mohan,
Disha Agarwal and Marco Caccamo

November 29, 2016

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Embedded Real-Time Systems

- **Time** and **Safety** critical!

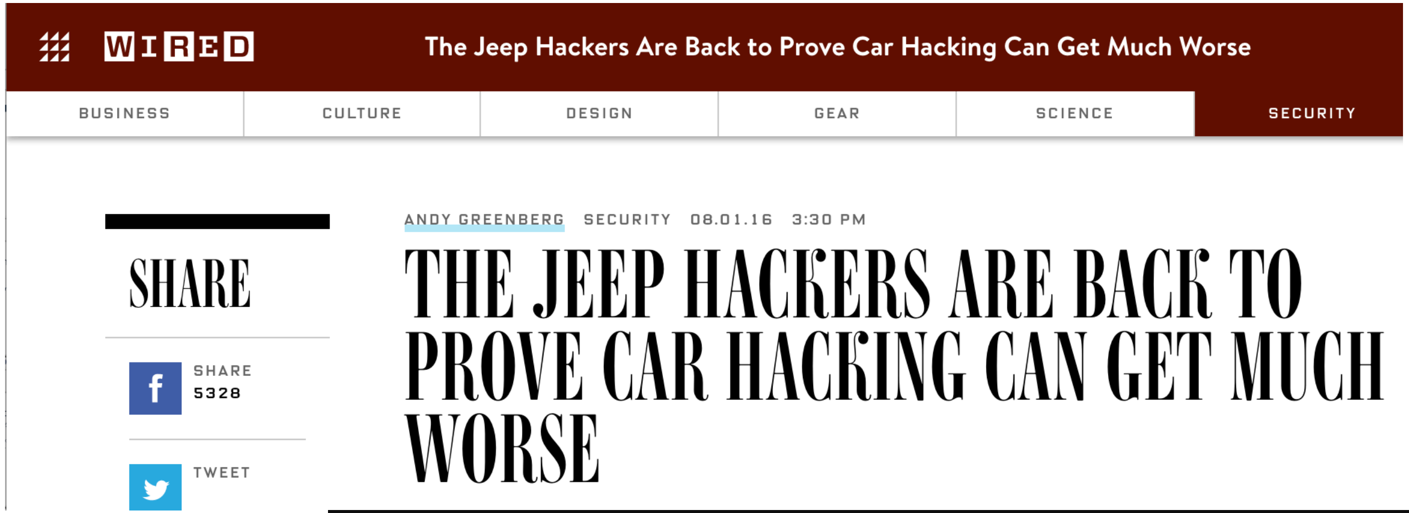


Is Cyber-Security an Issue for RTS Design?

- A decade ago:
  - *Perhaps NO*

Is Cyber-Security an Issue for RTS Design?

- A decade ago:
  - *Perhaps NO*
- Now:
  - ***YES!***



WIRED The Jeep Hackers Are Back to Prove Car Hacking Can Get Much Worse

BUSINESS CULTURE DESIGN GEAR SCIENCE SECURITY

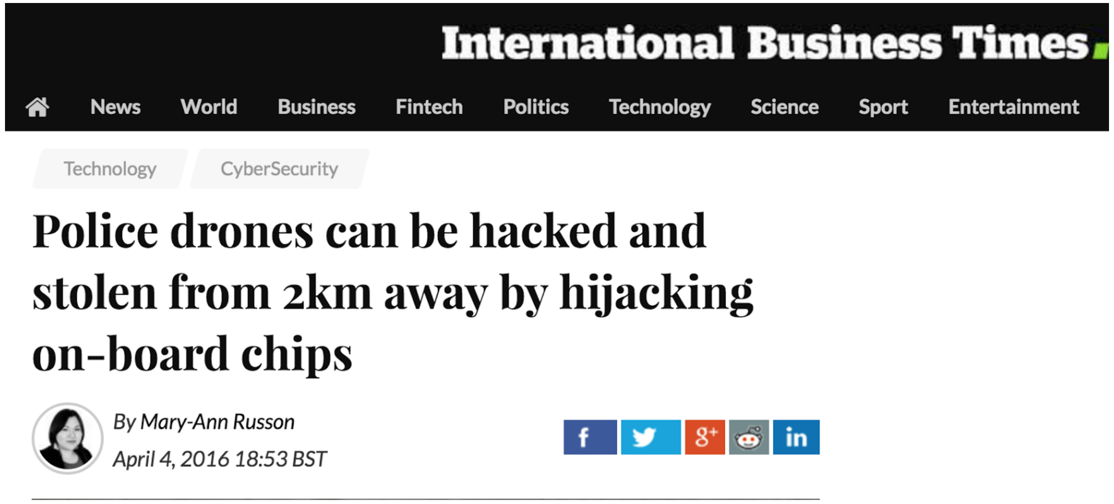
SHARE

SHARE 5328

TWEET

ANDY GREENBERG SECURITY 08.01.16 3:30 PM

THE JEEP HACKERS ARE BACK TO PROVE CAR HACKING CAN GET MUCH WORSE



International Business Times

News World Business Fintech Politics Technology Science Sport Entertainment

Technology CyberSecurity

Police drones can be hacked and stolen from 2km away by hijacking on-board chips

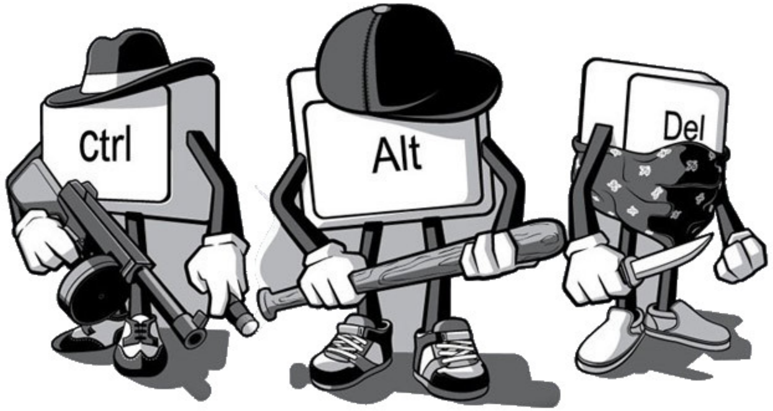
By Mary-Ann Russon

April 4, 2016 18:53 BST

Our Approach

Security through *Restarts*

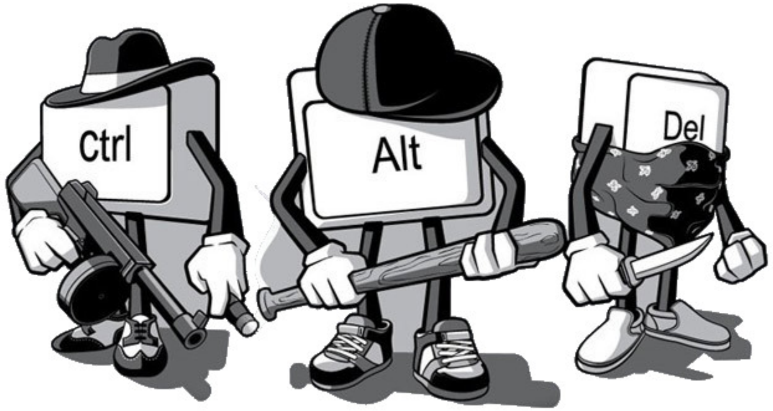
and fresh Reload!



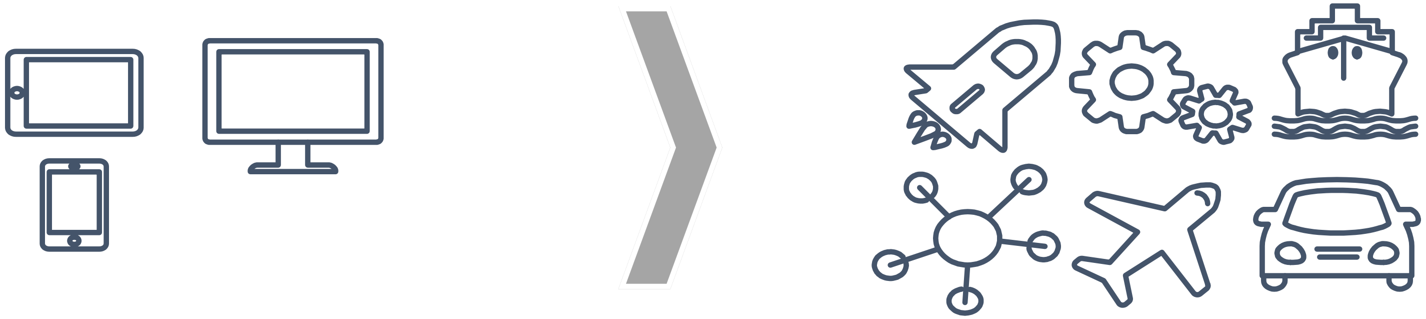
Our Approach : ReSecure

Security through ***Restarts*** and fresh Reload!

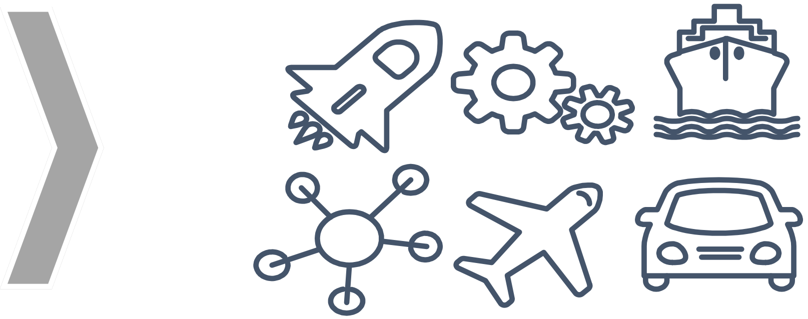
- Why Restart?
  - *Reliably remove malicious components*



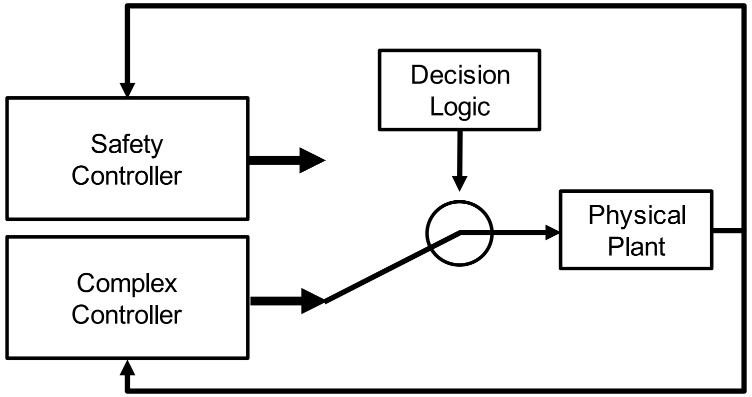
Restarting in RTS Domain



Restarting in RTS Domain



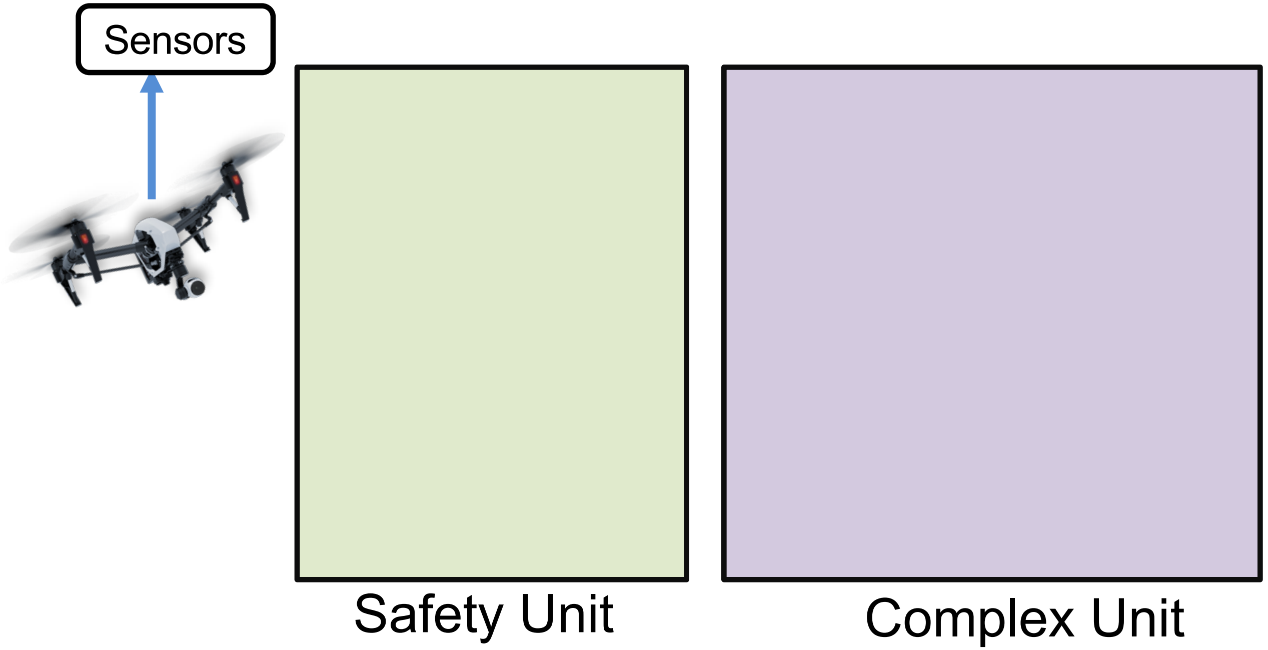
- How to enable Restart in RTS?
  - *Use Simplex! [Sha01]*



[Sha01] Lui Sha, Using Simplicity to Control Complexity, *IEEE Software*, 2001

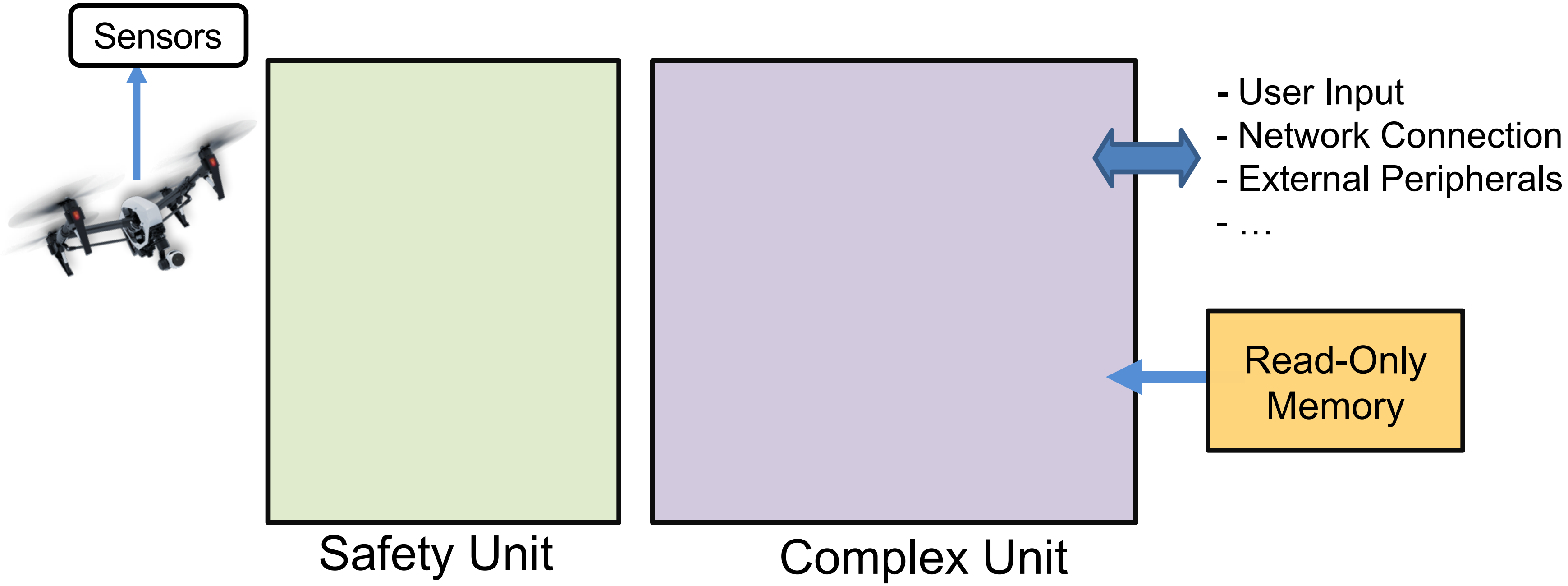
ReSecure

Architecture



ReSecure

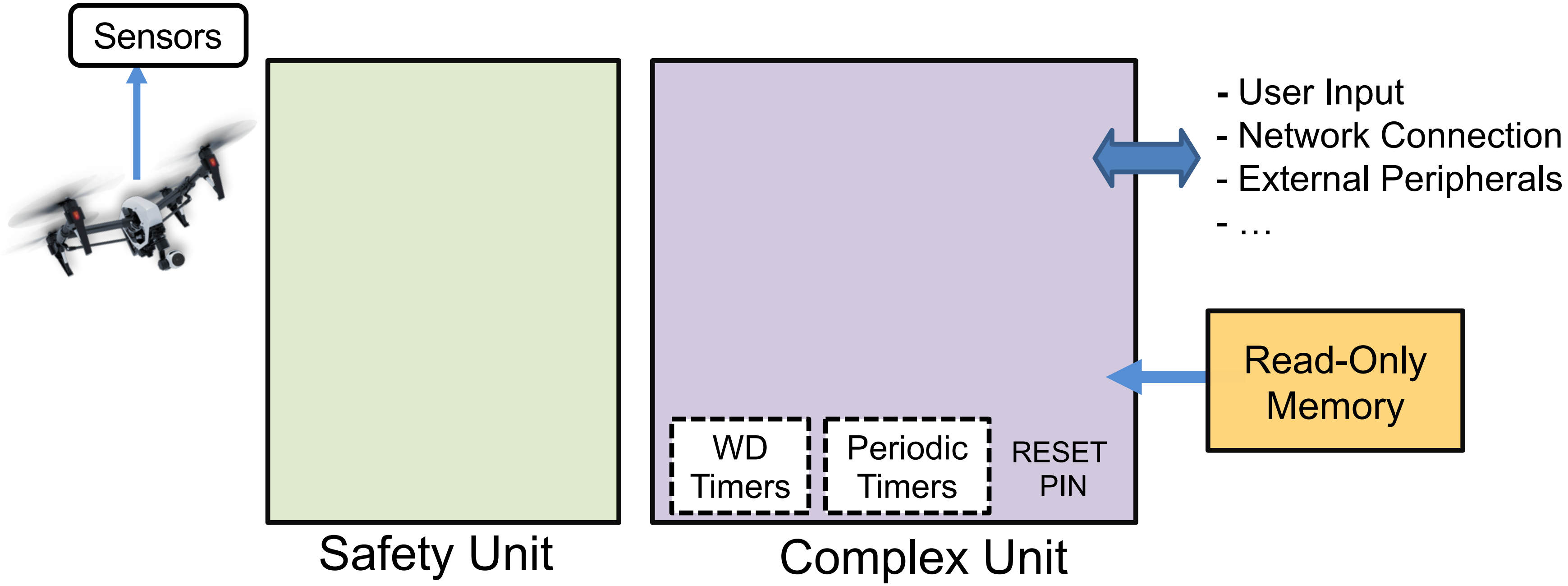
Architecture



- Safety Unit: Bare-metal, verified
- Complex Unit: OS/Firmwire can fail

ReSecure

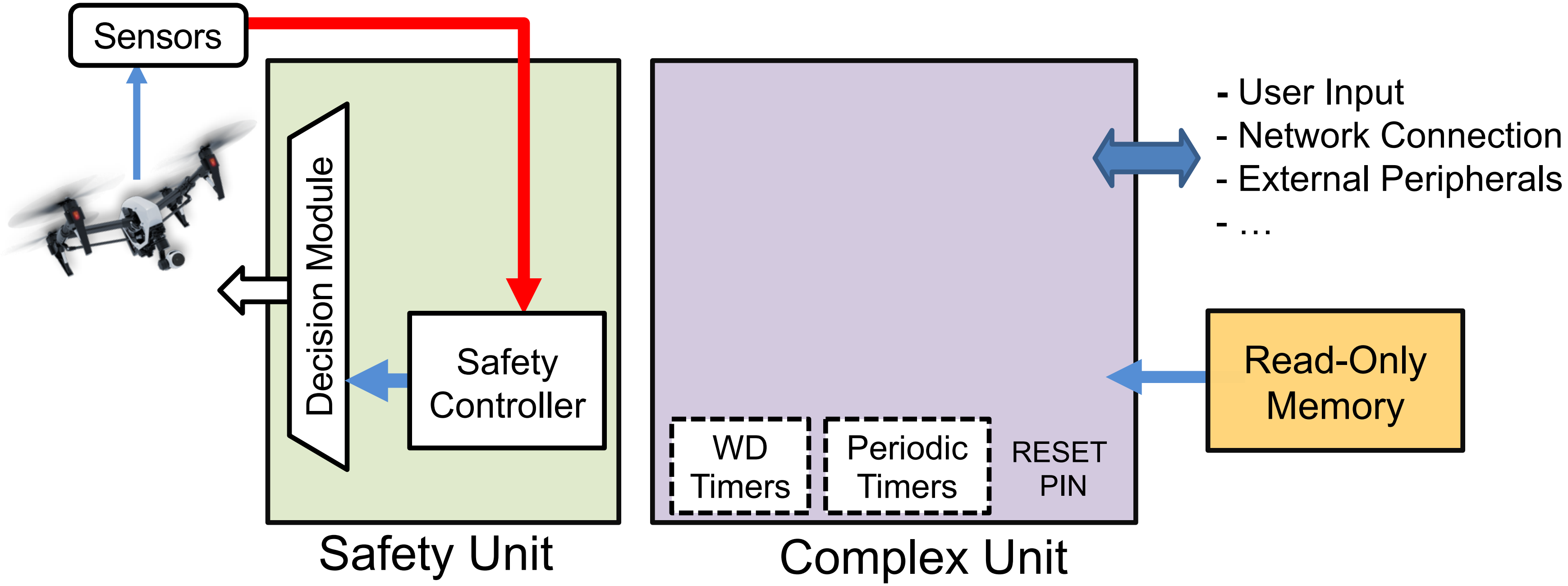
Architecture



- WD times: restart the Complex Unit upon fail-stop

ReSecure

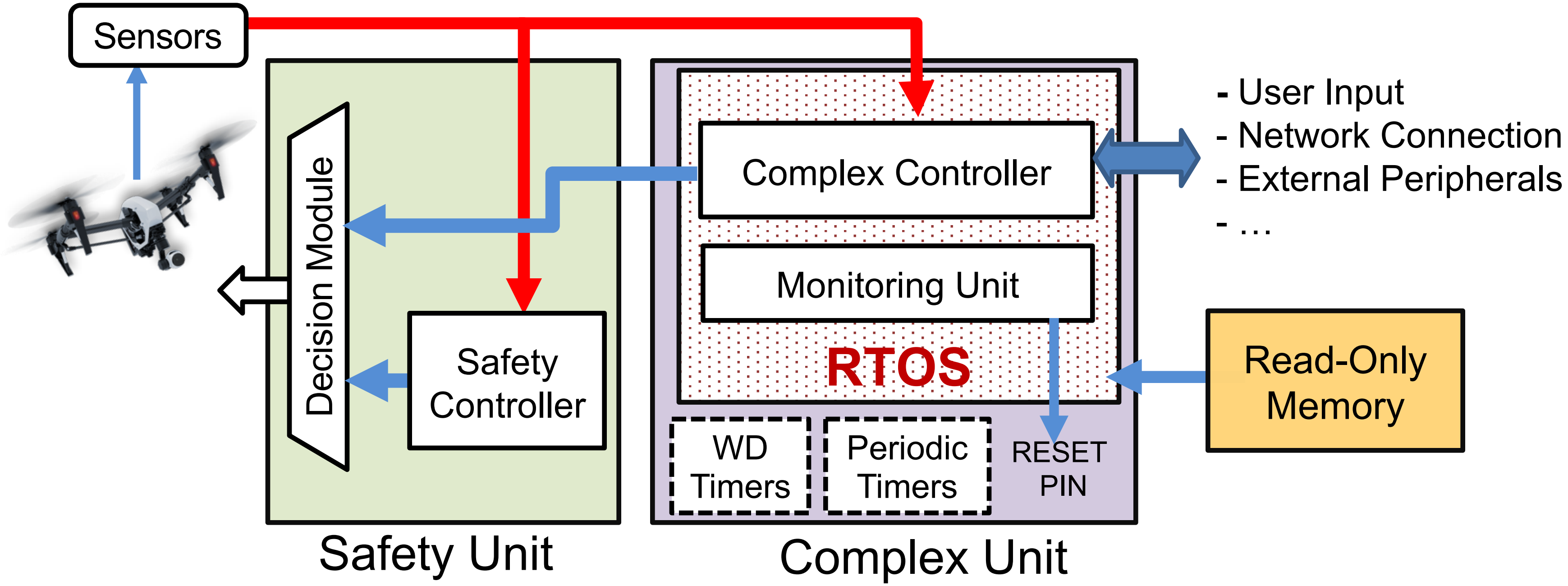
Architecture



- Safety Unit: can always keep the system safe!
- Decision Module: predicts if the future states are safe

ReSecure

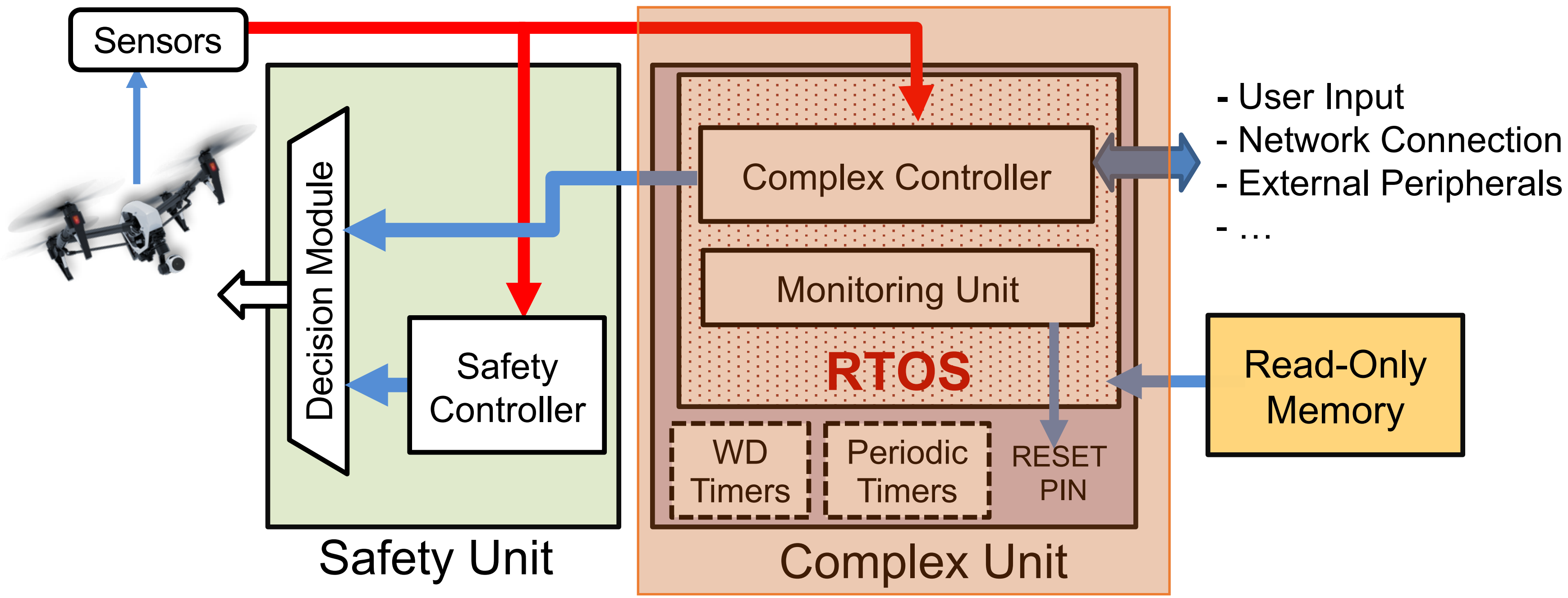
Architecture



- Complex Controller: not verified, can create unsafe command!

ReSecure

Architecture

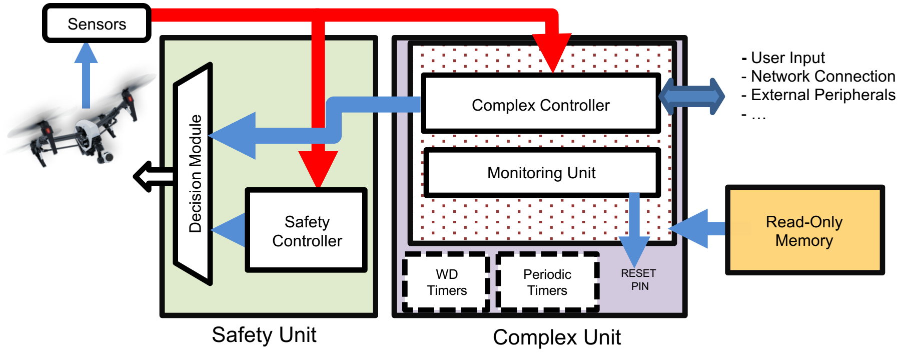


Complex Unit may get compromised!

Physical System remains safe!

Adversary Model

- Can compromise entire Complex Unit
  - *Includes real-time OS (RTOS) and the real-time applications*
- Denial of Service (DoS) attacks
  - *System and Network-level resource exhaustion*
- Information leakage through side-channels



Triggering Restart

1) Monitoring Unit
2) Watchdog Timers

Triggering Restart

1) Monitoring Unit
   - *Any technique can be implemented*
   - *Not 100% reliable!*

Triggering Restart

1) Monitoring Unit
   - *Any technique can be implemented*
   - *Not 100% reliable!*
2) Watchdog Timers
   - *Can recover the system when the monitoring unit is compromised*

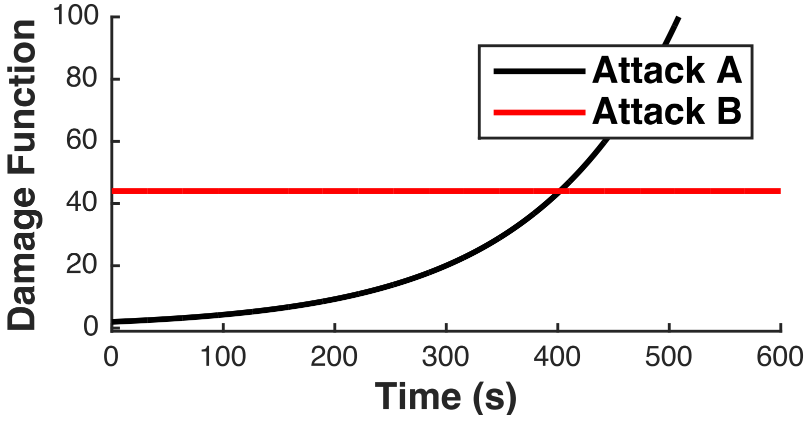
Restart timer frequency

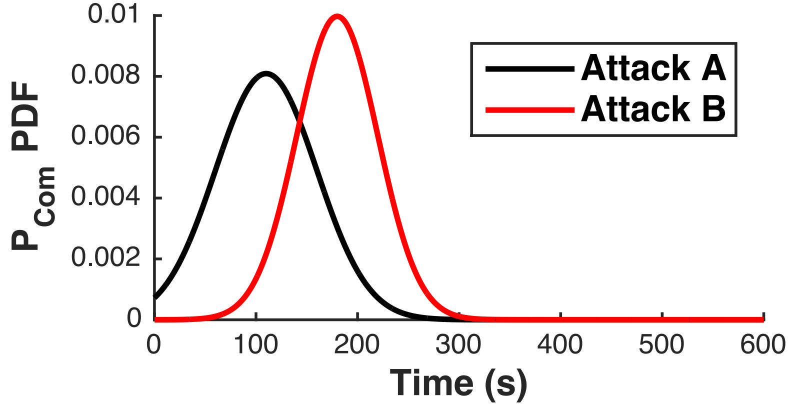
Lower Security
Higher Performance

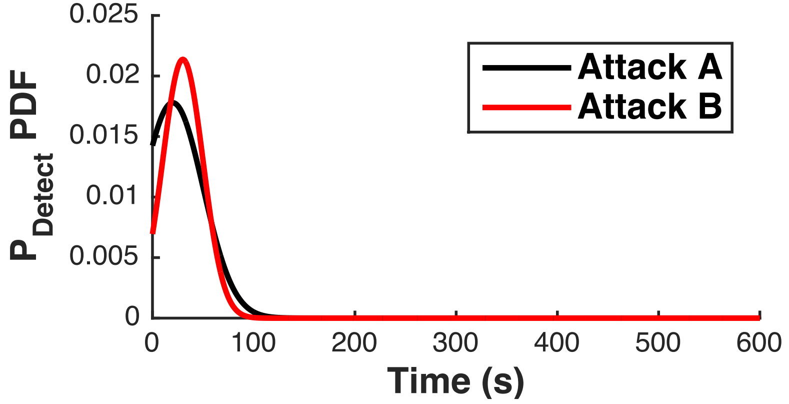
Higher Security
Lower Performance

Impact of Restart

Properties of Attacks



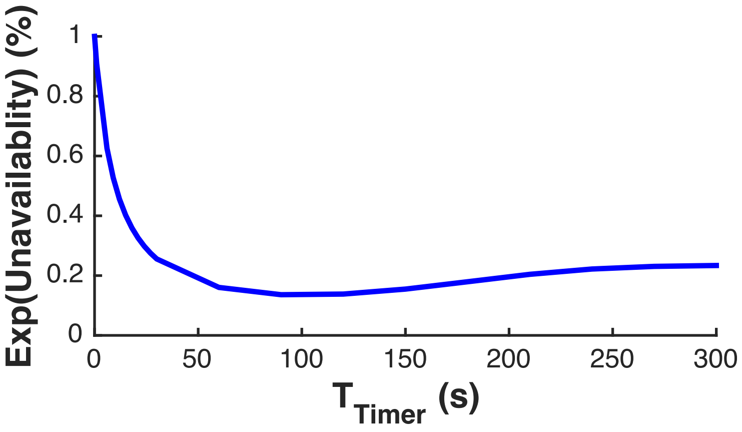


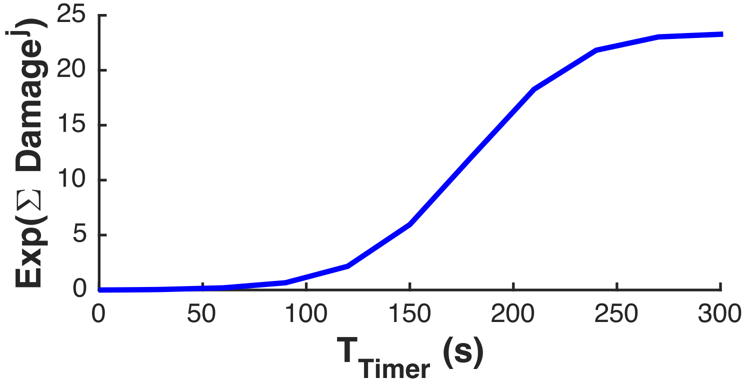


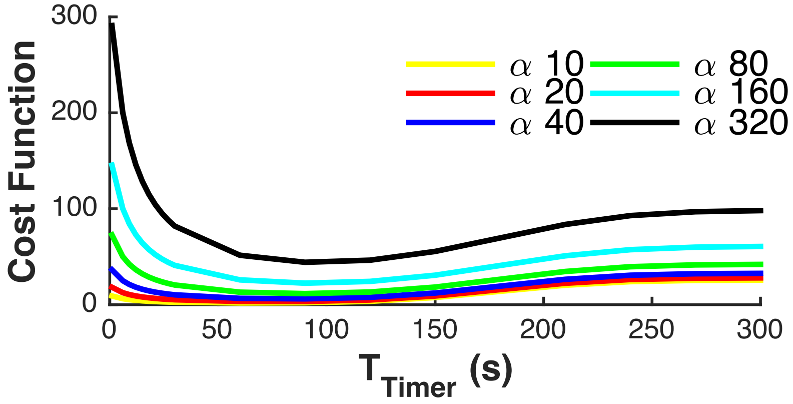
Impact of Restart

Performance-Security Tradeoff

$$Cost(T_{Timer}) = Exp(\text{Total Damage}) + \alpha \cdot Exp(\text{Unavailability}).$$

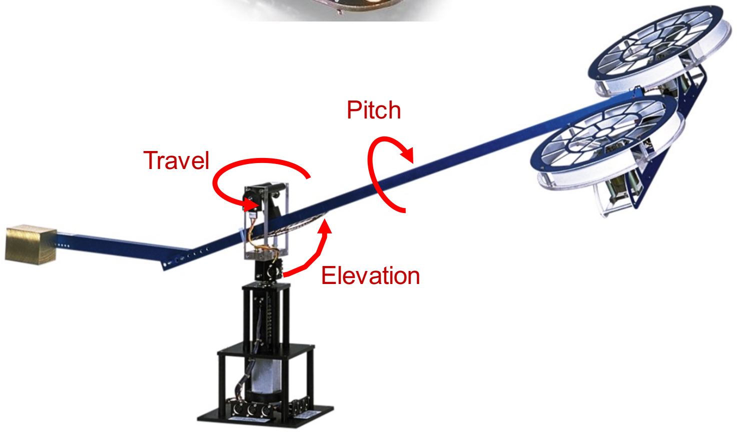






Implementation

- **Complex Unit:**
  - *BeagleBone Black (ARM Cortex-A8)*
- **OS:**
  - *Linux with RT-PREEMPT patch*
- **Real-time system:**
  - *Hardware-in-Loop simulation of 3-DOF Helicopter*



○ Source-codes: https://github.com/mnwrhsn/restart_n_secure_cps

Experience & Evaluation

- **Recovery by Watchdog Timeout:**
  - *Launch "fork bomb" attack*
  - *Recovered from the attack by ~14 second*
- **Recovery by Monitoring Unit:**
  - *Inject a kernel-level malware*
  - *Intercepts every* `read()` *system call*
  - *Recovers from the intrusion within* $T_R + T_{MU} - t = 13 + 2 - t \approx 15\ sec.$

Conclusion & Future Work

- Illustrate restart as a viable mechanism to ensure security in safety-critical systems
- Designers of the system can now evaluate the necessary trade-offs between control system performance degradation and increased security guarantees
- Future work
  - *Domain Specific Analysis of Attacks*
  - *Randomization and Restarts*

THANK YOU!

Questions?

SUPPLIMENTARY SLIDES

Safety Controller Design

- Goals:
  - *To keep system within the Linear constraints*
  - *To stay within the limits of actuators*
- Strategy:
  - *To find a region where all the above are always satisfied*
  - *To design a state feedback controller that keeps the system within that region*