Eliminating Inter-Domain Vulnerabilities in Cyber-Physical Systems: **An Analysis Contracts Approach**

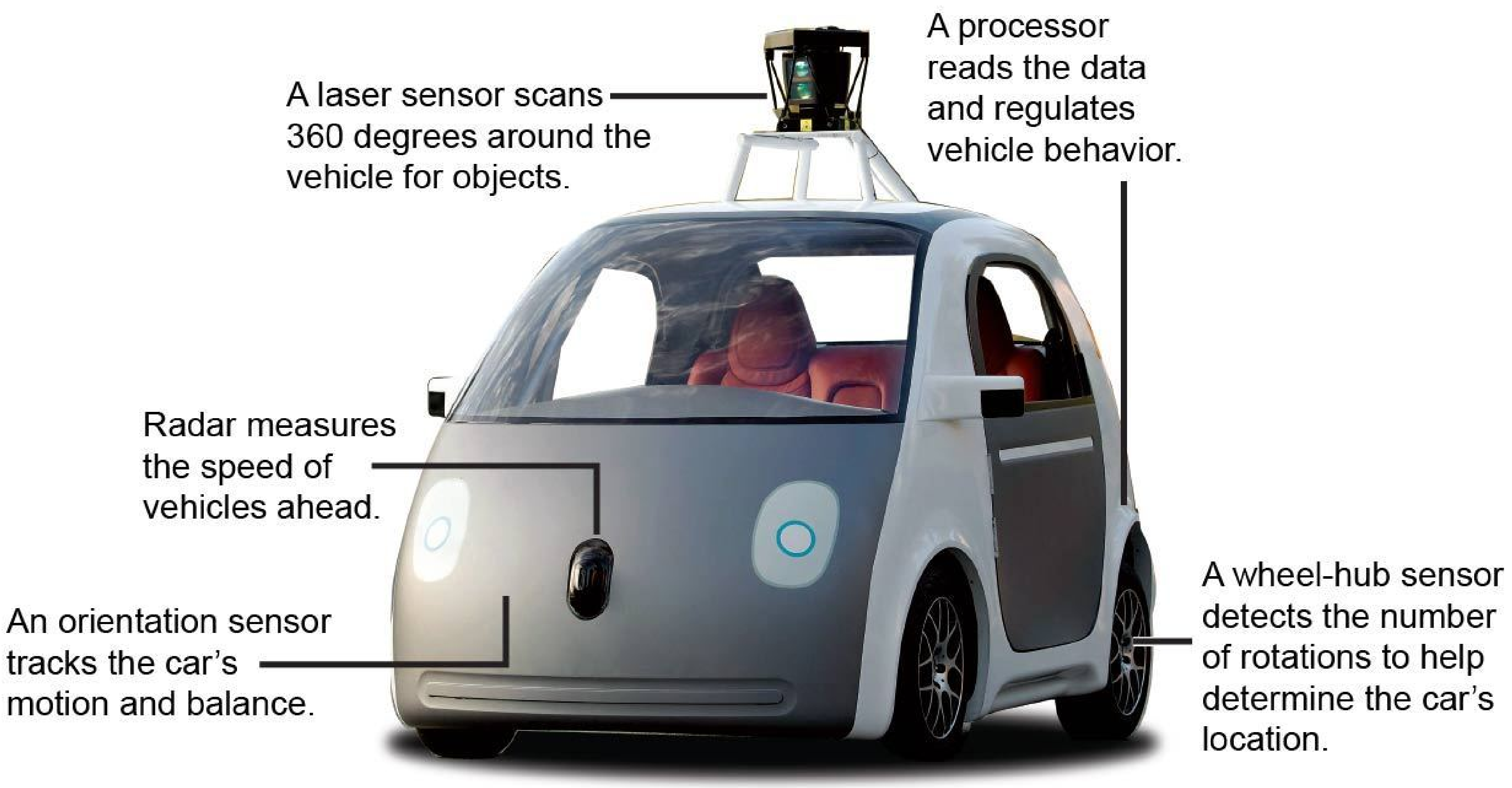
Ivan Ruchkin
Ashwini Rao
Dionisio de Niz
Sagar Chaki
David Garlan

Carnegie Mellon

isr institute for SOFTWARE RESEARCH

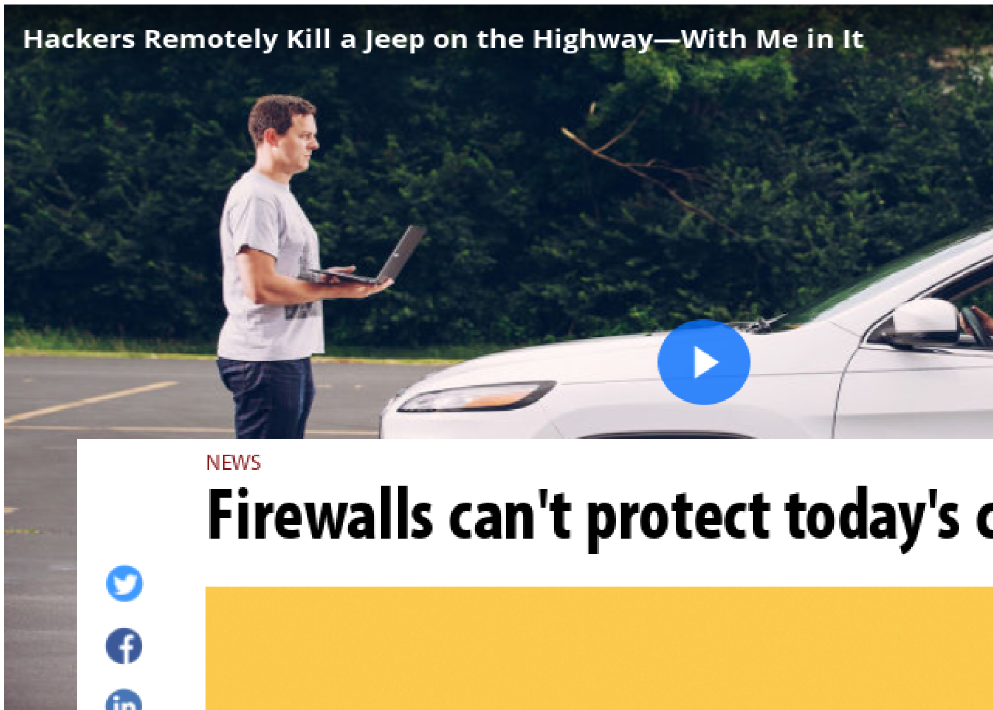
Software Engineering Institute

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.



- Safety, efficiency, fault-tolerance
  - Formal verification, control theory, reliability engineering, ...

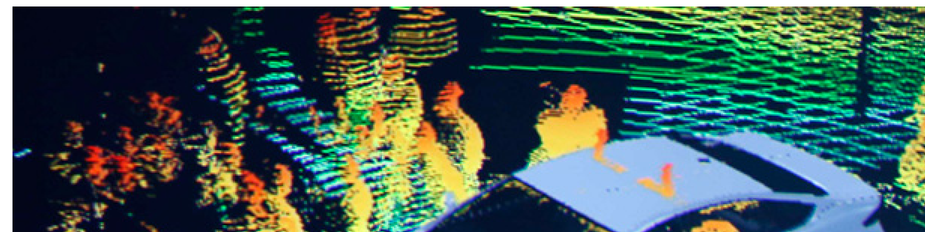
HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT



Researcher Hacks Self-driving Car Sensors

By Mark Harris
Posted 4 Sep 2015 | 19:00 GMT

Share | Email | Print | Reprint



NEWS

Firewalls can't protect today's connected cars

MORE LIKE THIS

Hacker: 'Hundreds of thousands' of vehicles are at risk of attack

Update: Chrysler recalls 1.4M vehicles after Jeep hack

Senators call for investigation of potential safety, security threats from...

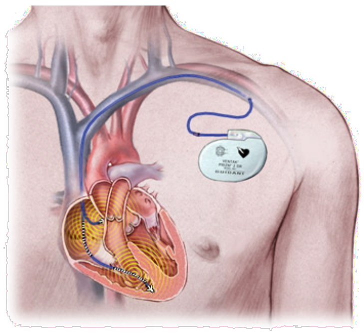
on IDG Answers

If I buy a Chromebook and can't get to grips with OS can I convert to windows?

Cyber-Physical Systems and Vulnerabilities

- Software-controlled distributed autonomy
- Complex physical behavior



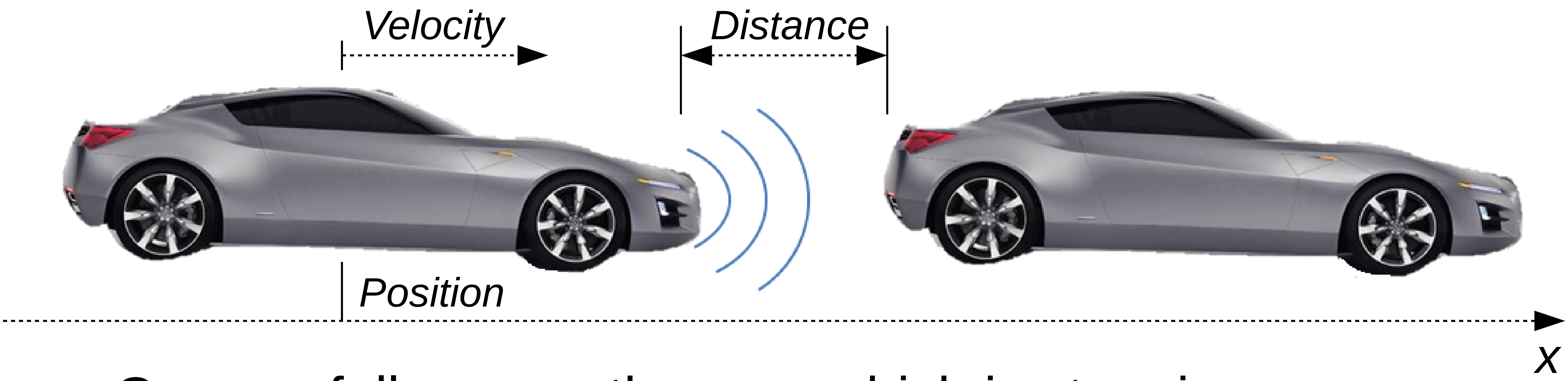



- Diverse interactions: networks, physics, …
  - Potentially malicious
- Diverse attack surfaces and vulnerabilities

Outline

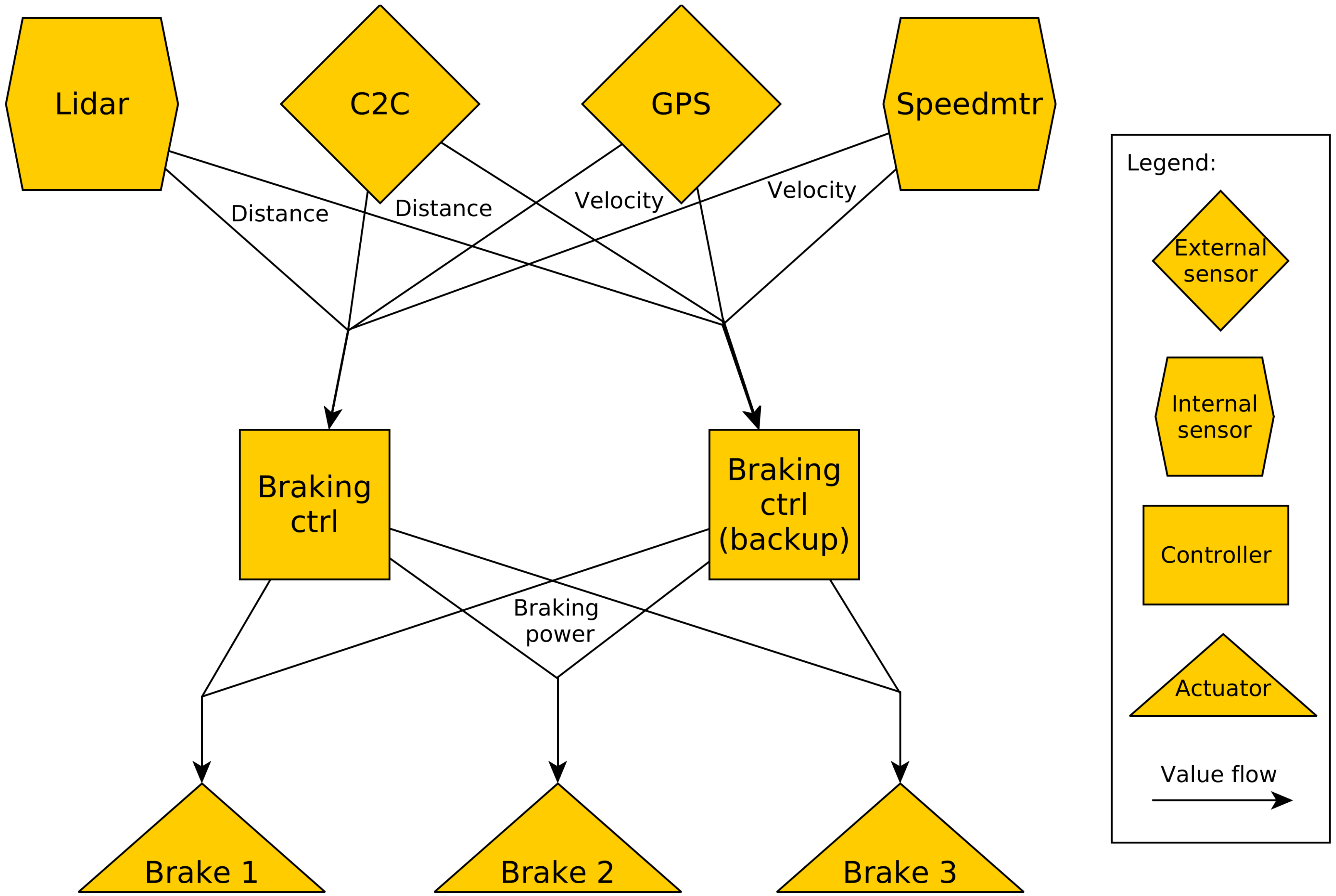
- Security in cyber-physical systems
- **Inter-domain vulnerabilities**
- Analysis contracts approach
- Discussion

Scenario



- One car follows another car, which is stopping.
- Senses position, distance, and velocity.
- *Safety:* must brake and stop without crashing.
  - Depends on effective *control:* slows down smoothly (esp. on ice)
  - Depends on *reliability:* stops even if a sensor malfunctions
  - Depends on *sensor security:* stops even if a sensor is spoofed

Braking Subsystem Architecture



Full model: *github.com/bisc/collision_detection_aadl*

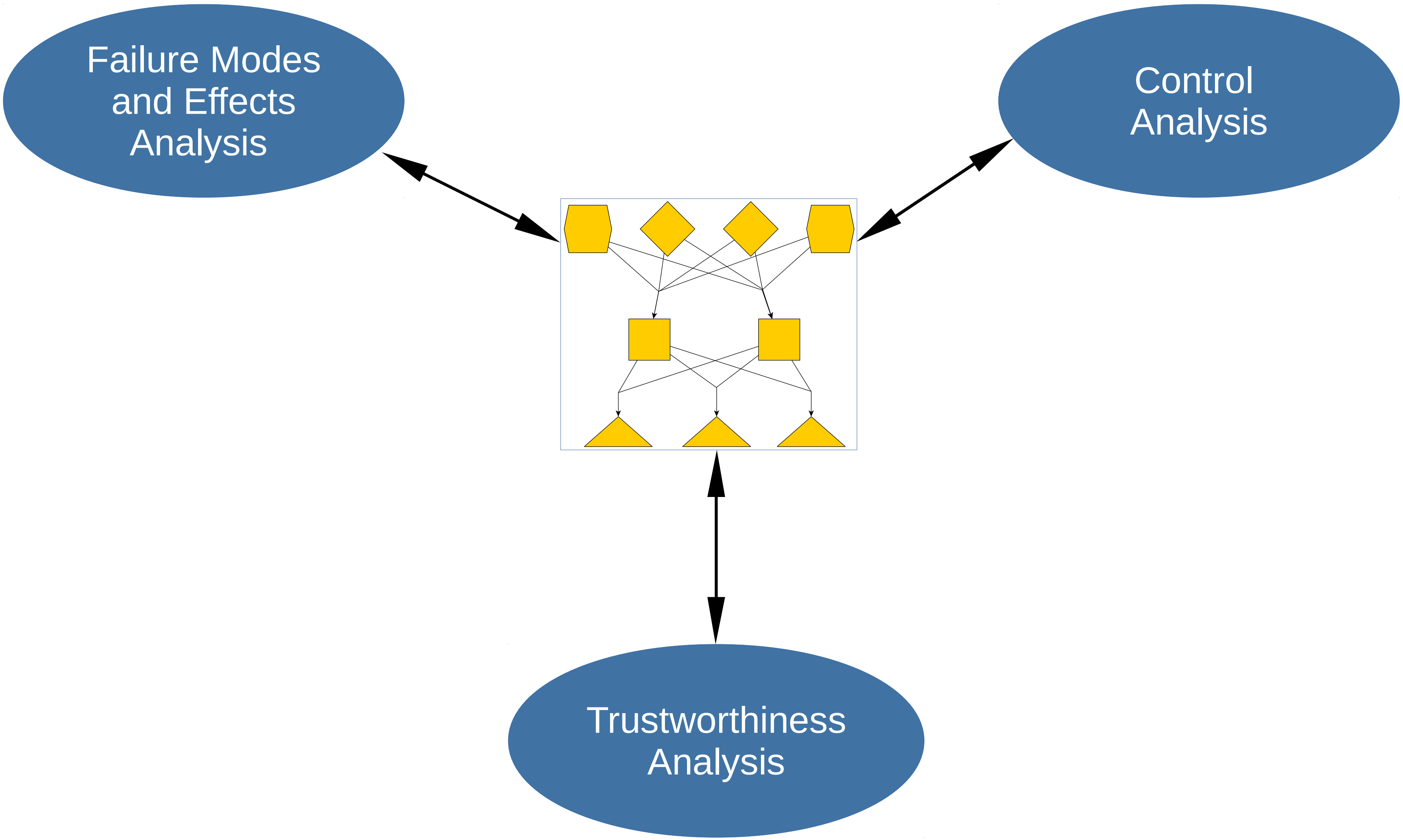
Exploiting Sensors

- Adversary models:
  - Knows the system's architecture
  - Internal or external (not all-powerful)
  - Spoofs data for respective sensor type
- Attack steps (online):

  1. Find a vulnerable set of sensors in a car
  2. Spoof all of the sensors in the set

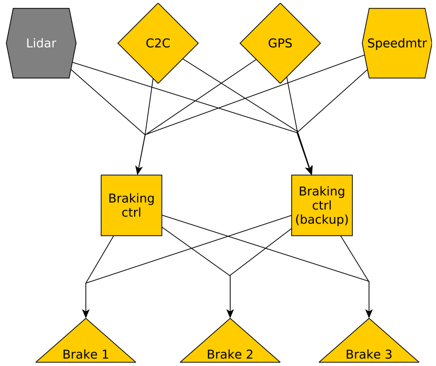
  Impact: the control is misled and possibly crashes

Analyses (offline)

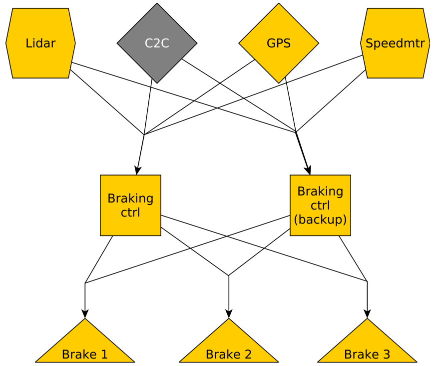


Analysis 1: FMEA

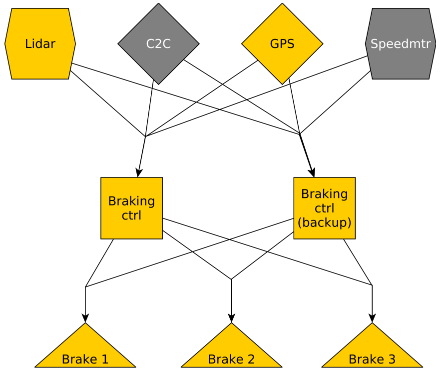
- *Failure Modes and Effects Analysis* [Schneider1996]
  - Mature and common in reliability engineering
- Goals:
  1. Determine most likely “failure modes”
     - Configurations where some components failed



P = 0.1



P = 0.05



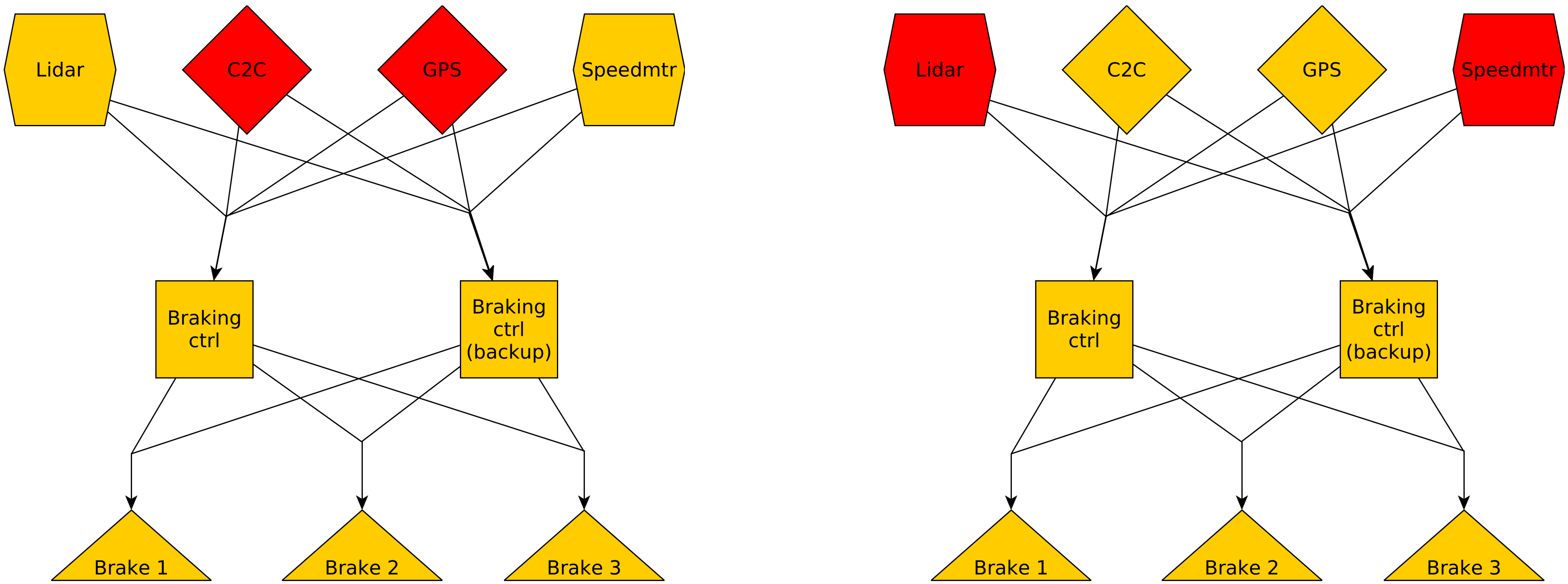
P = 0.01

  2. Augment the system to reduce failure likelihood

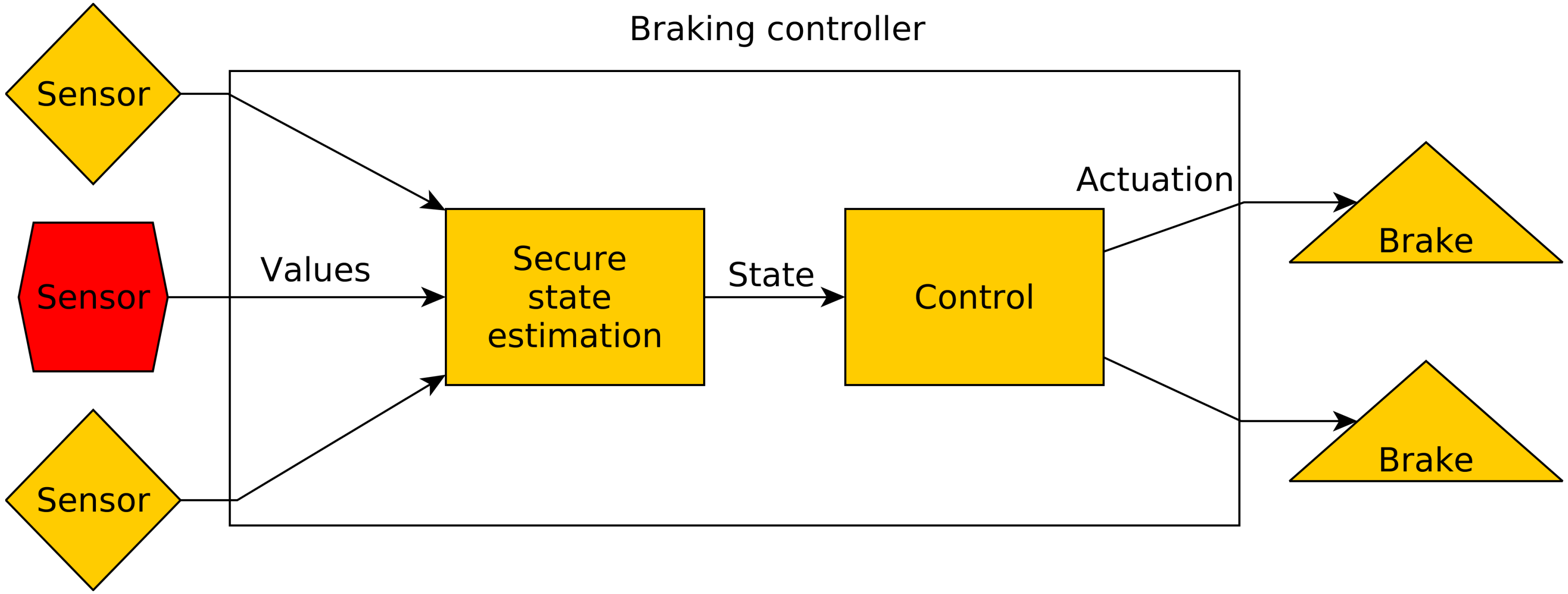
Analysis 2: Sensor Trustworthiness

- Goal: determine trustworthiness of each sensor
  - Given an attacker model [Miao2013]

Internal attacker *External attacker*



Analysis 3: Secure Control



- Goals: [Fawzi2014]

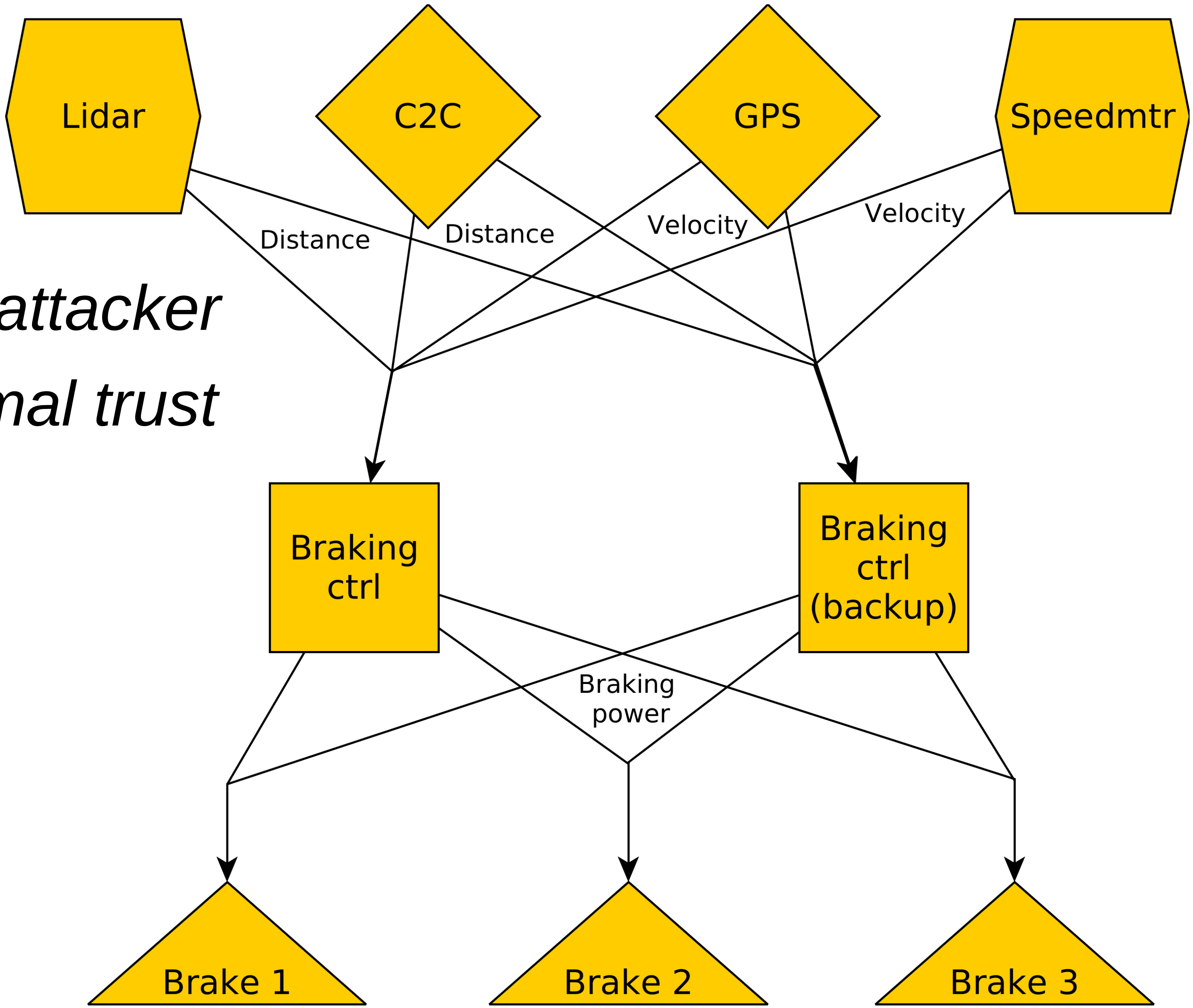
1. Tune controllers and state estimators
2. Determine if control is safe and smooth

- **Minimal sensor trust assumption:** at least 50% sensors are providing trustworthy data (for each sensed variable)

Exploiting Vulnerability

Internal attacker

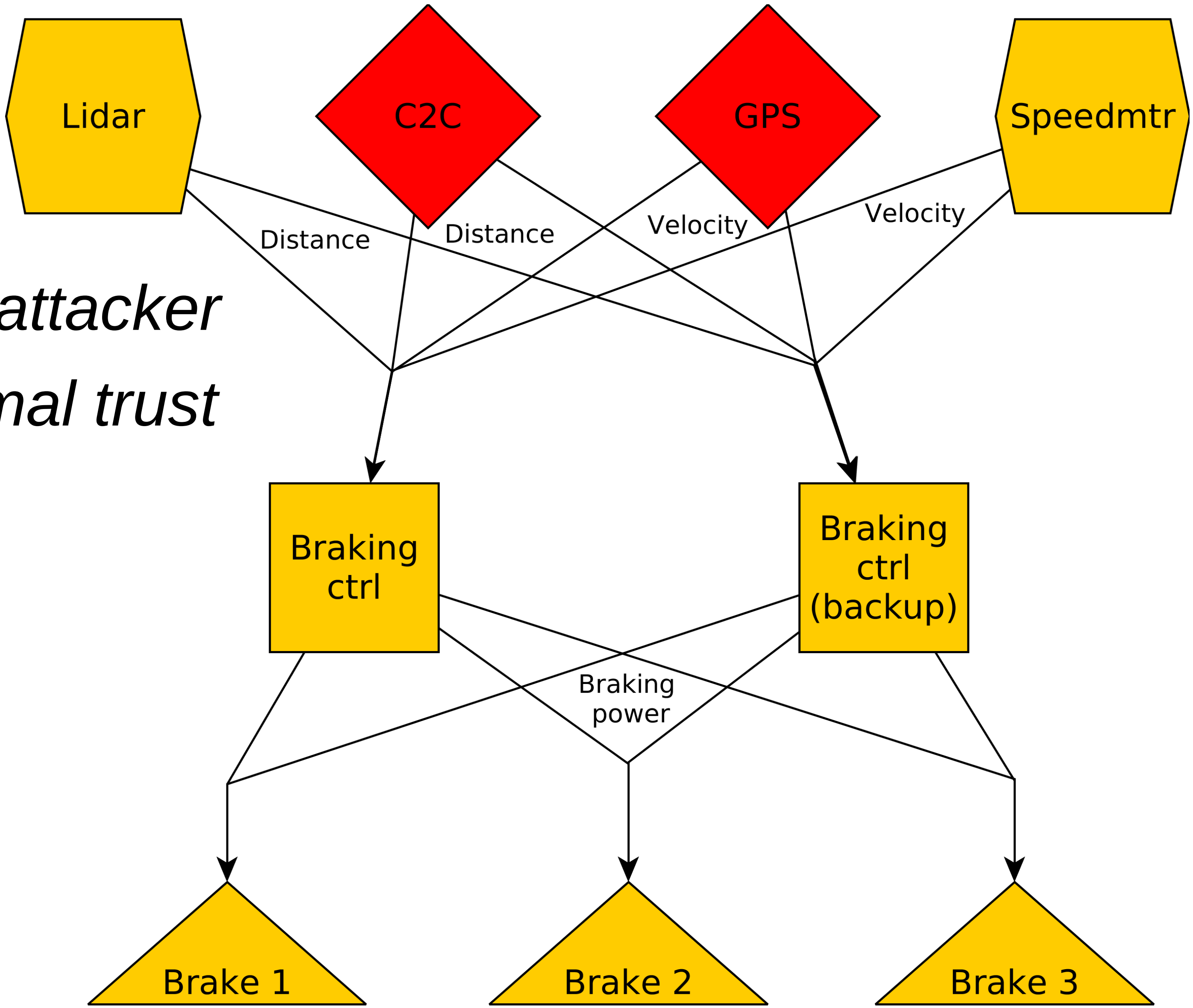
✔ *minimal trust*



Exploiting Vulnerability

Internal attacker

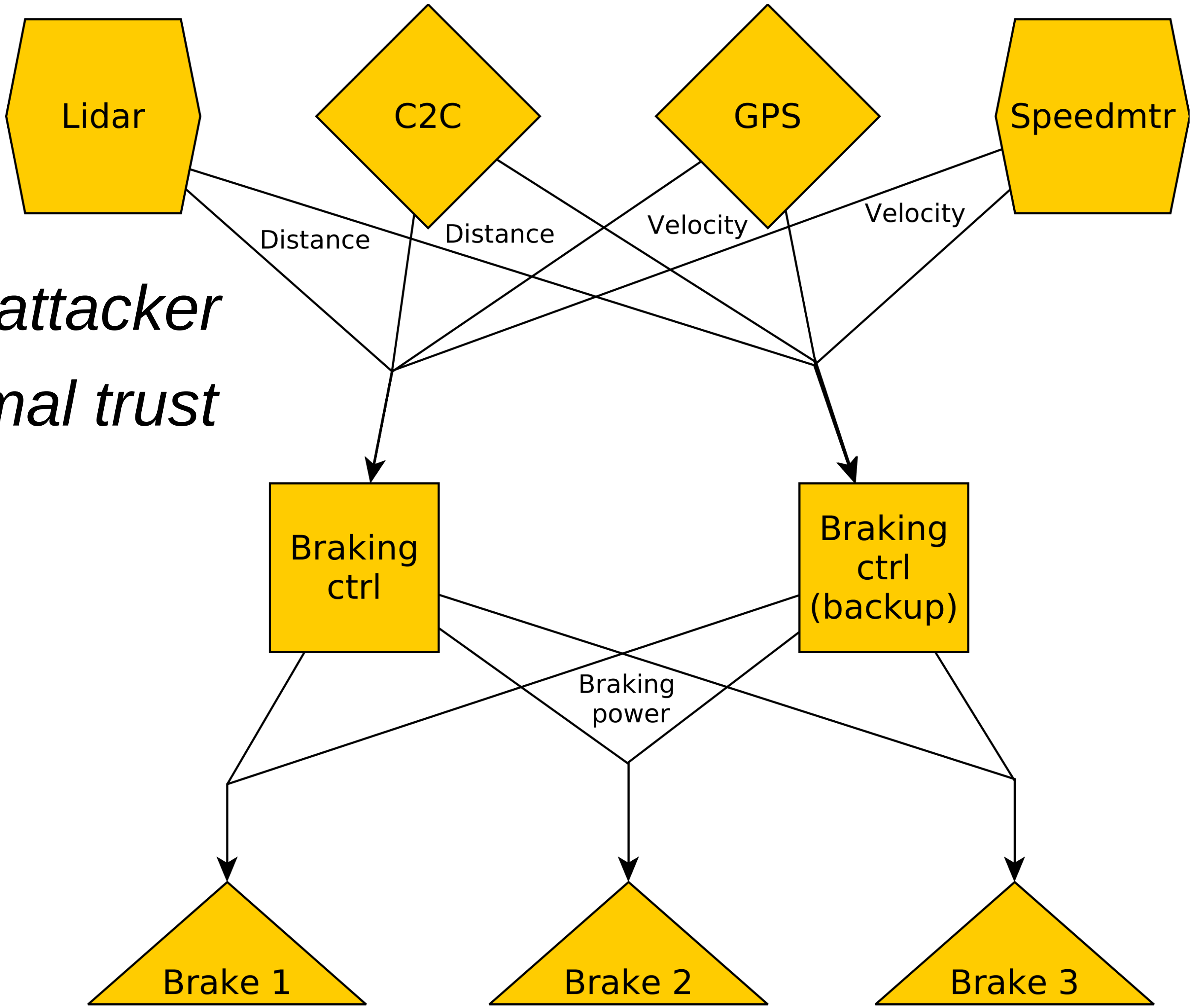
✓ *minimal trust*



Exploiting Vulnerability

Internal attacker

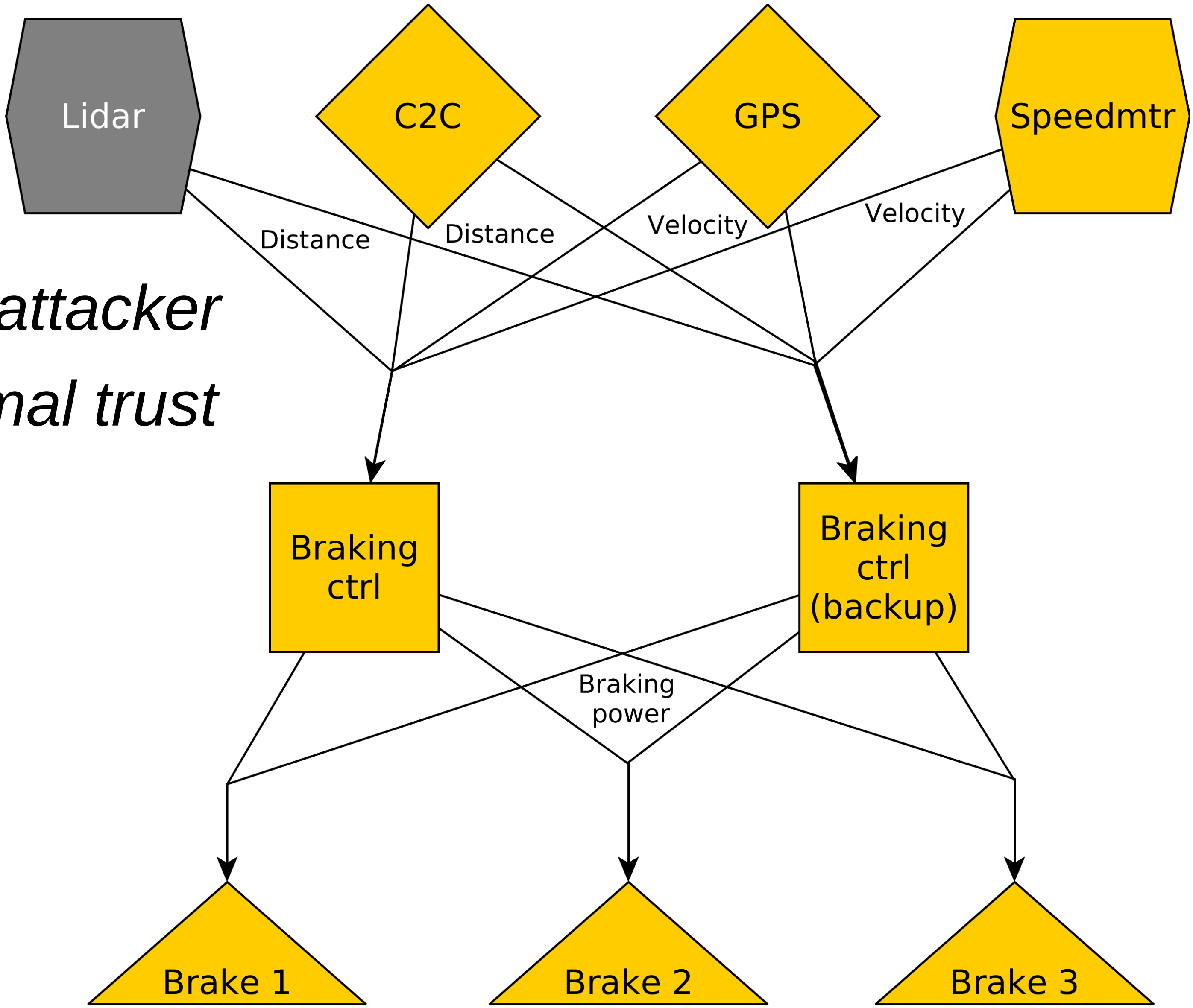
✔ *minimal trust*



Exploiting Vulnerability

Internal attacker

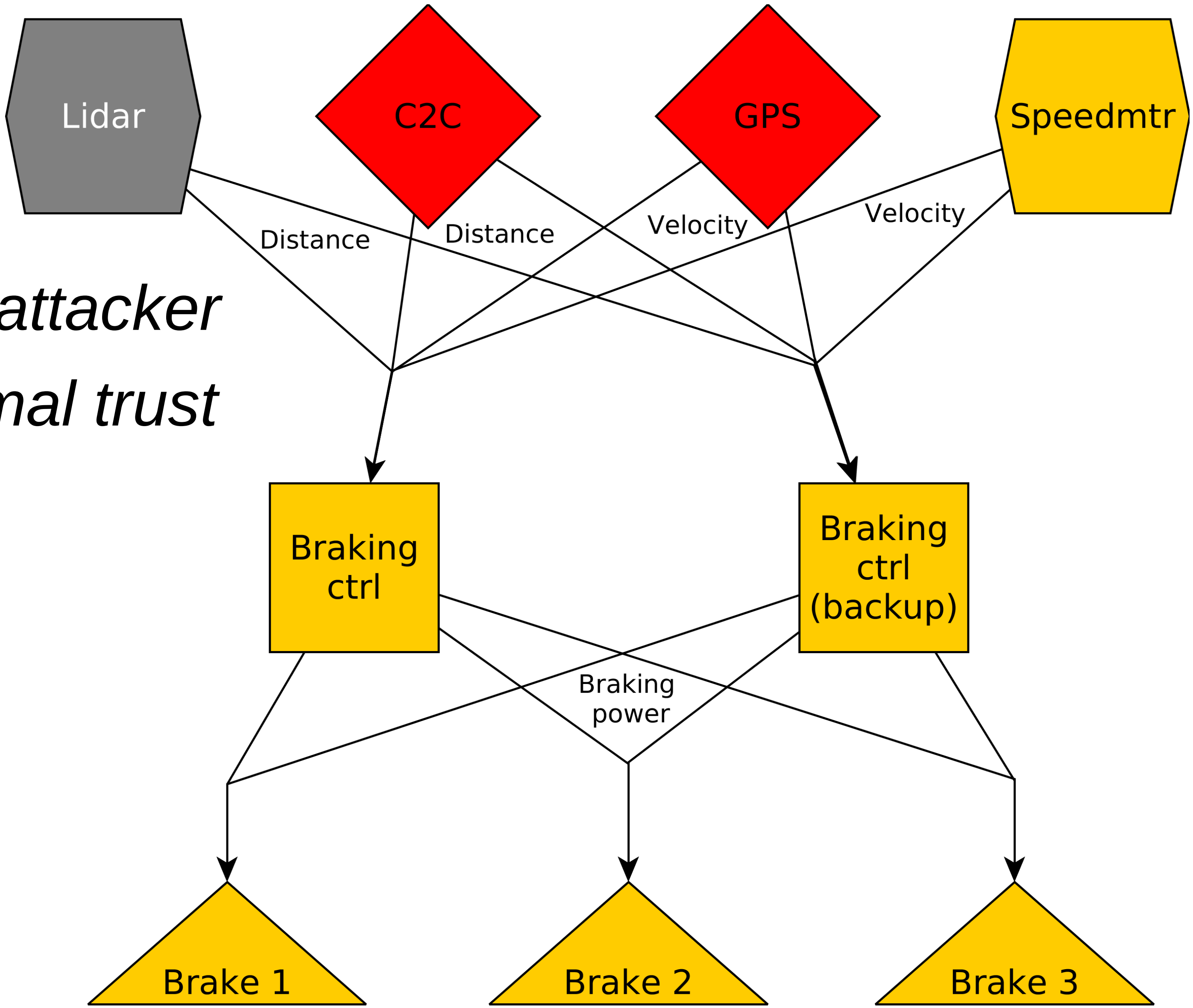
✔ *minimal trust*



Exploiting Vulnerability

Internal attacker

✖ *minimal trust*



Problem: Inter-Domain Vulnerabilities

- Uncontrolled *analysis interactions* may lead to introduction of vulnerabilities into CPS.
- *Cause:* unsatisfied dependencies and assumptions.
- Introduced offline, exploited online.

Outline

- Security in cyber-physical systems
- Inter-domain vulnerabilities
- **Analysis contracts approach**
- Discussion

Possible Solutions

- Cybersecurity **online**: IDS, firewalls
  - Oblivious of diverse engineering analyses
- Cybersecurity **offline**: encryption, secure protocols, secure-by-design
  - May not work with physical world
- **Control-theoretic** CPS security *[Fawzi2014]*
  - Does not consider fault-tolerance and other factors
- **Component** modeling, interface theories
  - Focuses on system parts, not quality concerns

Analysis Contracts Approach

1. Model the system's architecture
2. Formalize *contracts* for analyses *[Ruchkin2014]*
   - Inputs, outputs, assumptions, guarantees
3. Execute analyses correctly (offline)
   - Dependencies met
   - Assumptions satisfied

- *Expectation:* inter-domain vulnerabilities are detected and prevented

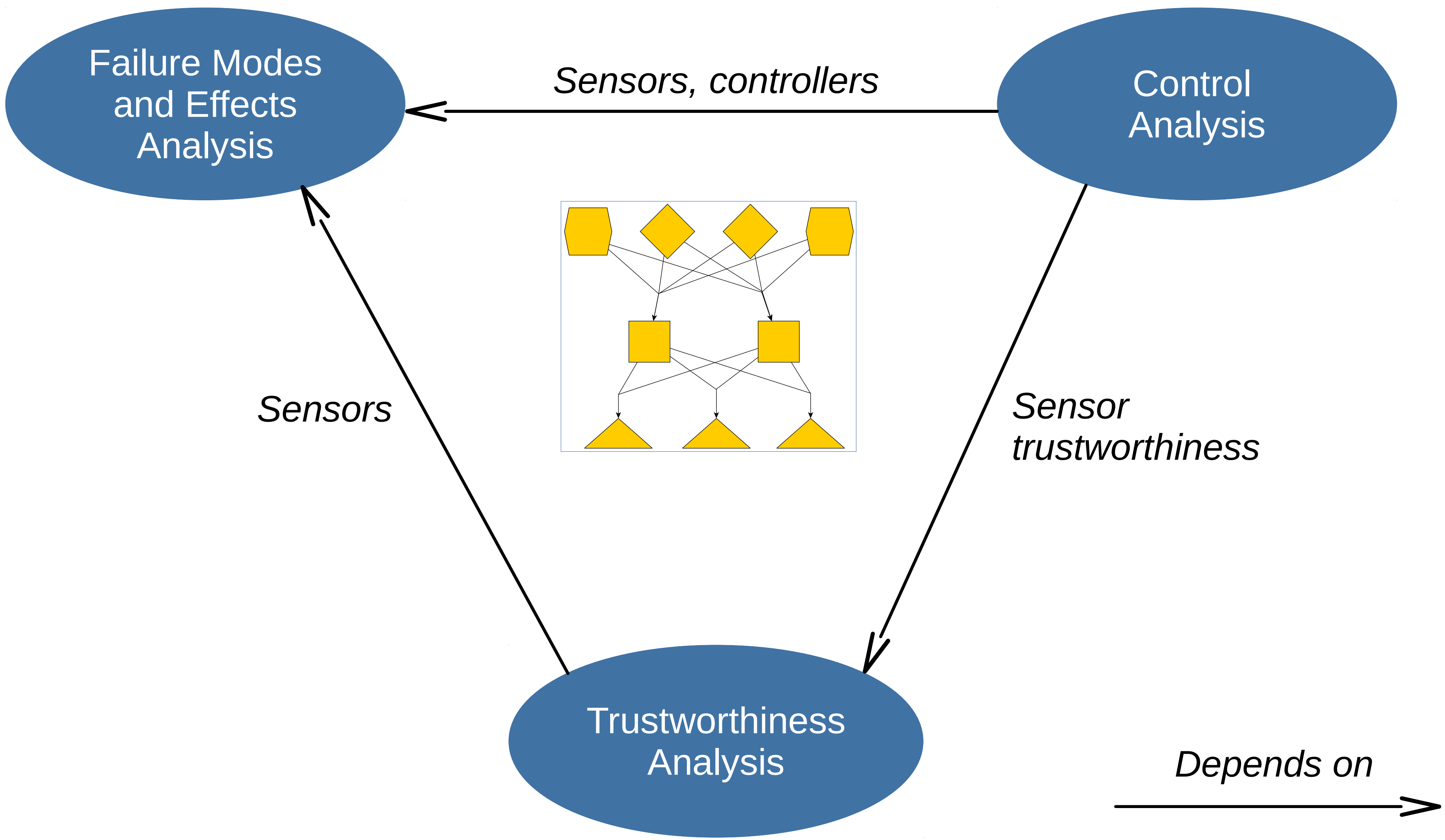
Step 1: Architecture Modeling

- AADL – *Architecture Analysis and Design Language* [Feiler2005]
- Provides standardized high-level vocabulary
  - *Components and connectors:* sensors, controllers, actuators, …
  - *Properties:* sensor variables, trustworthiness, attacker model, ...
  - *Modes:* configurations of components, connectors, and their properties

Step 2: Analysis Contract Specification

Analysis	Input	Output
FMEA	Fault-tolerance requirements	**Sensors**, **controllers**, modes
Trustworthiness	**Sensors**, attacker model	Sensor trustworthiness
Control	**Sensors**, **controllers**	Control safety

Analytic Dependencies



Assumptions and Guarantees

- Logically specify for each analysis
- Ctrl analysis assumption (minimal sensor trust):

$$\forall m \in \mathbb{M} \cdot |m.S_{trustworthy}| / |m.\mathbb{S}| \geq 0.5$$

- Actual second-order encoding in SMTv2:

$$\begin{aligned} &\forall m \in \mathbb{M}\ \forall c \in m.\mathbb{R}, v \in c.\mathsf{VarsR} \cdot \\ &\quad \exists f : \mathbb{S} \to \mathbb{S} \cdot \forall s_u \in m.\mathbb{S} \cdot \\ &\qquad v \in s_u.\mathsf{VarsS} \wedge s_u.\mathsf{Trust} = \bot \implies \\ &\qquad \exists s_t \in m.\mathbb{S} \cdot v \in s_t.\mathsf{VarsS} \wedge s_t.\mathsf{Trust} = \top \wedge f(s_t) = s_u \end{aligned}$$

Step 3: Contract Verification

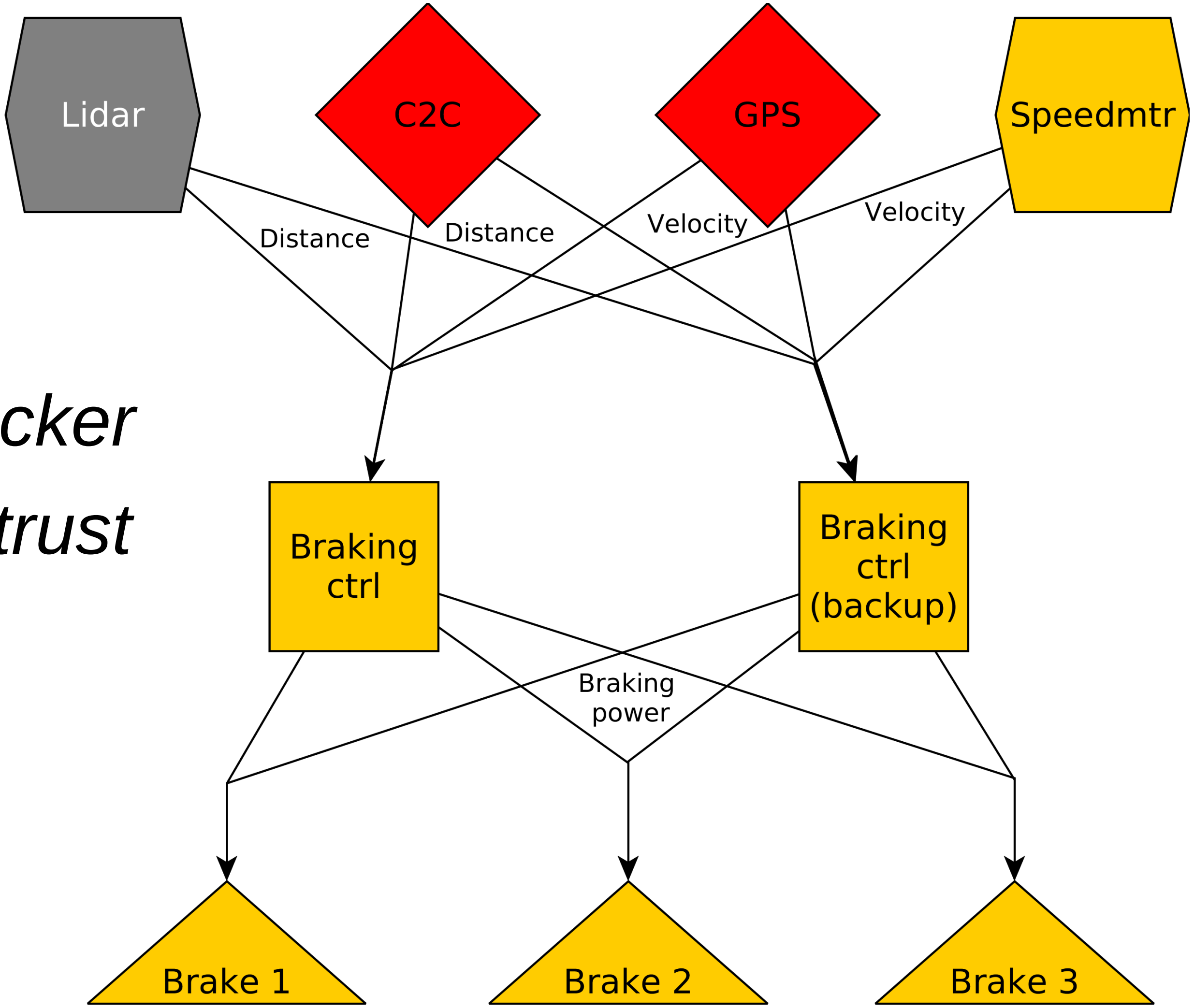
- Deterministic: first-order predicate logic
  - Implemented in the ACTIVE tool *[Ruchkin2014]* using the Z3 solver
  - Doesn't support second-order yet
- Probabilistic
  - Not fully designed, or implemented
  - Plan to:
    - Incorporate Probabilistic Computation Tree Logic (PCTL) in the language
    - Use probabilistic model checking tools: PRISM or MRMC

Detecting Vulnerability

$$\forall m \in \mathbb{M} \cdot |m.S_{trustworthy}|/|m.\mathbb{S}| \geq 0.5$$

Internal attacker

✖ *minimal trust*



Outline

- Security in cyber-physical systems
- Inter-domain vulnerabilities
- Analysis contracts approach
- **Discussion**

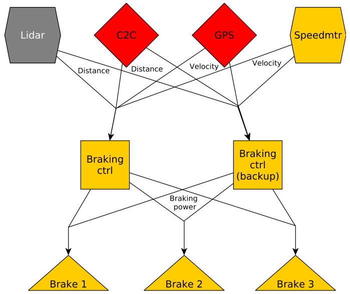
Limitations

- Generality
  - Approach applicable to other domains?
- Scalability & expressiveness
  - Will verification be feasible in other cases?
- Practicality
  - Is the up-front formal effort worth it?

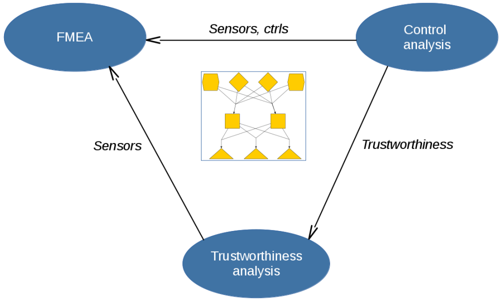
Future Work

- Richer contracts
  - Behavioral models for security
  - Probabilistic statements
  - Something else?
- Incorporating relevant domains
  - Suggestions?
- Validation
  - NOT building a self-driving car from scratch
  - Ideas?

Summary



- Described inter-domain vulnerabilities
- Demonstrated the analysis contracts approach
  - Specified analysis contracts
  - Determined dependencies
  - Verified deterministic assumptions



- *Future work:* more models and analyses, richer contracts, and validation

Email me: iruchkin@cs.cmu.edu
ACTIVE tool: github.com/bisc/active
Car model: github.com/bisc/collision_detection_aadl

References

- H. Schneider. Failure Mode and Effect Analysis: FMEA From Theory to Execution. Technometrics, 38(1), 1996.
- C. Miao, L. Huang, W. Guo, and H. Xu. A Trustworthiness Evaluation Method for Wireless Sensor Nodes Based on D-S Evidence Theory. In Wireless Algorithms, Systems, and Applications, Springer, 2013.
- H. Fawzi, P. Tabuada, and S. Diggavi. Secure Estimation and Control for Cyber-Physical Systems Under Adversarial Attacks. IEEE Transactions on Automatic Control, 59(6), 2014.

References (continued)

- I. Ruchkin, D. D. Niz, D. Garlan, and S. Chaki. Contract-based integration of cyber-physical analyses. In Proceedings of the 14th International Conference on Embedded Software. ACM Press, 2014.
- I. Ruchkin, D. De Niz, S. Chaki, and D. Garlan. ACTIVE: A Tool for Integrating Analysis Contracts. In The 5th Analytic Virtual Integration of Cyber-Physical Systems Workshop, Rome, Italy, 2014.
- P. H. Feiler, B. Lewis, S. Vestal, and E. Colbert. An Overview of the SAE Architecture Analysis & Design Language (AADL) Standard: A Basis for Model-Based Architecture-Driven Embedded Systems Engineering. In Architecture Description Languages. Springer Science, 2005.
- R. Nieuwenhuis, A. Oliveras, C. Tinelli. Solving SAT and SAT Modulo Theories: From an Abstract Davis–Putnam–Logemann–Loveland Procedure to DPLL(T). In Journal of the ACM, 2006.
- L. de Moura and N. Bjrner. Z3: An Efficient SMT Solver. In Lecture Notes in Computer Science, pages 337{340. Springer, 2008.

AADL Example

```
system implementation avoidance_subsystem.impl
    subcomponents
        avoidance_process_A: process collision_threat_handler.A;
        avoidance_process_B: process collision_threat_handler.B;
        watchdog_process: process watchdog_proc.impl;
        vehicle_processor: processor basic_computing::real_time.one_ghz;
        vehicle_memory: memory basic_computing::ram.standard;
        vehicle_bus: bus basic_computing::basic_bus.standard;
        bus_driver: device basic_devices::bus_driver.standard;
        event_distributor: device basic_devices::event_distributor.standard;
    modes
        -- sensor failure modes
        nominal: initial mode;
        fail_mode_1: mode;
        fail_mode_2: mode;
        fail_mode_3: mode;

        nominal-[condition_1]->fail_mode_1;
        nominal-[condition_2]->fail_mode_2;
        nominal-[condition_3]->fail_mode_3;

        fail_mode_1-[condition_nominal]->nominal;
        fail_mode_2-[condition_nominal]->nominal;
        fail_mode_3-[condition_nominal]->nominal;
```

Probabilistic Contracts

- *Reliability assumption:* “probabilities of sensors not working are independent.”

$$\forall c_1, c_2 \in \mathbb{S} \cdot P(\neg c_1.\text{Avail} \mid \neg c_2.\text{Avail}) \leq P(\neg c_1.\text{Avail}) + \epsilon_{fail}$$

- *Security assumption:* “probabilities of sensors not working are dependent.”

$$\exists c_1, c_2 \in \mathbb{S} \colon P(\neg c_1.\text{Avail} \mid \neg c_2.\text{Avail}) \geq P(\neg c_1.\text{Avail}) - \epsilon_{trust}$$